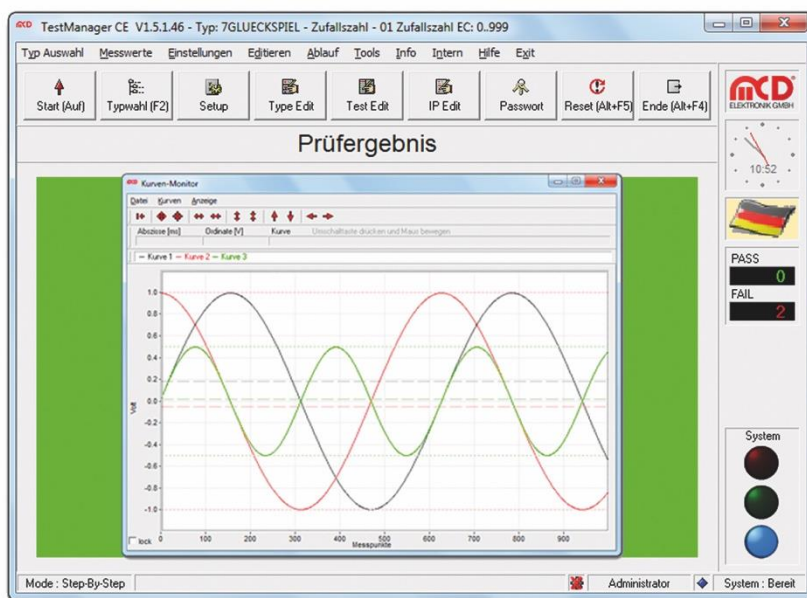# Manual

TestManager CE

GET IN **touch**
WITH SENSITIVE TESTING

Softline
Modline
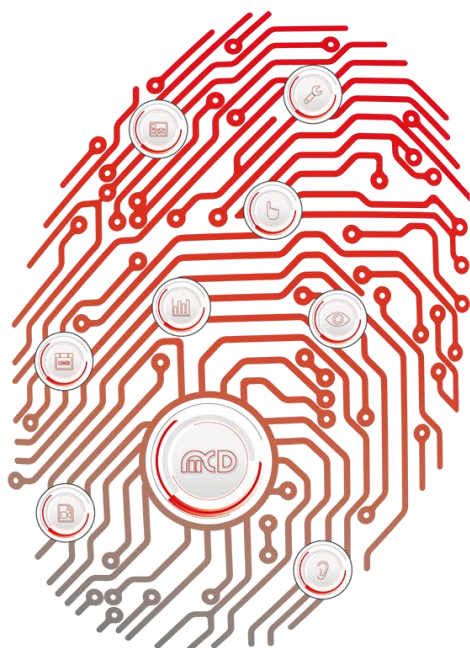Conline
Boardline
Avidline
Pixline

Application

MCD Elektronik GmbH	Date: 2018-04-12	Page 2 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Table of Contents

MCD Elektronik GmbH	Date: 2018-04-12	Page 4 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 6 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 7 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

MCD Elektronik GmbH	Date: 2018-04-12	Page 8 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			MCD ELEKTRONIK GMBH
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

MCD Elektronik GmbH	Date: 2018-04-12	Page 10 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

MCD Elektronik GmbH	Date: 2018-04-12	Page 11 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

1 Getting started

1.1 System requirements

- Windows NT, 2000 or XP
- Program directory on a local drive
- Files are allowed to be copied in the program directory
- 32 MB RAM
- VGA Monitor (640*480 Pixel)
- Pentium® PC or compatible
- Installed hardware according to configuration
- Optional: Installed network, if needed
- Optional: Installed Borland Database Engine (BDE)

This configuration is the minimum requirement, the computer can always be configured higher.

The program itself does not create any data files except it's program directory. The only exceptions are the files in paths which are explicitly specified (e.g. for saving measured values on the network)

The program can also be started on normal (office-) PCs, when the access to the non-available hardware is deactivated in the basic settings of the program.

The program supports Window XP Stile, when it is activated in the operating system and it should also be activated on the desktop option in the basic settings.

1.2 Installation

The. TestManager-software requires no special installations. All the required information is saved within the directory structure. A backup of the TestManager -Software is thus possible by making a copy of the entire directory. On the other hand it is also possible to copy the TestManager -Software (including all the required sub-directories) on the test computer. It may also be required that the copied test program should be configured as per the test computer.

To update the TestManager program it is only required to copy the necessary files. To be on the safer side it is suggested to create backups of the files that are being copied. If for an update some other files are required to be copied in addition to the TestManager Program, then this may be mentioned in the update information.

The TestManager program may need a special directory structure to function properly. (see details in chapter 16.2)

1.3 License

A program license is required for to access the system ports using TestManager program. This access depends on the connected hardware (Victor measuring system) or it can be available as a system depending license from MCD. Without a license it is unable to access the system ports (only an editing mode is available).

More information on the topic license is available in chapter 16.1.

1.4 BDE (optional)

A Borland database engine (BDE) is required if the measured values are to be stored in a DBF format. If the BDE is absent or corrupted on the test computer it can be installed from the provided installation CD. After the installation following settings are required.

MCD Elektronik GmbH	Date: 2018-04-12	Page 13 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Open the system configuration (control panel), click on the BDE administration option, select the option card 'Configuration', select the path 'Configuration/Driver/Native/Dbase'. Change the value of 'Level' to 5. To make these changes administrator rights are required.

2   Start of the program

The TestManager program will be started on executing the .EXE file. By fully automatic testing units it is recommended to make a shortcut of the program in the auto start directory.

Following parameters can be given:

/EDITONLY

inhibits the program to access the hardware. The current license is then ignored and the program goes in an edit only mode. If the F7 button is pressed when the TestManager program is starting a similar edit only mode can be achieved.

/APPLICATIONNAME=<VALUE>

displays the window title with the name in <VALUE>. If <VALUE> consists of many words, then the entire parameter must be set in double inverted commas, Example:
```
"/APPLICATIONNAME=This is a complete titel."
```

/FULLEXCEPTION

Normally the threads present in the wait status are shown as exception messages. This option gets all the threads in the exception message.

3   Used notations

In this chapter the different notations used in the program are described.

Type – Type family – Type variants

The program allows the test of many different DUT (Device under test) types. Each DUT type requires an individual set of data for its test for e.g. test sequence, complexity of test, limiting values, stimuli etc. The program manages the data separately for each type family. Each type family consists of one ore more type variants.

One type variant is a unique type of DUT. On the basis of the data of a type variant the exact test sequence is determined by the TestManager program. Two different type variants should be present for DUTs having different test sequences.

If two types of DUTs (and so their type variants) are so similar that they can be managed together, they can be managed in one type family.

If the DUTs are entirely different, then each type variant forms an own type family.

Type list

All type families and type variants defined in system are listed in the type list.

Column

A column is a mechanical unit, i.e. listed in the program in order to test the DUTs. One column consists of 1..n rows or contact positions. For only one DUT the column contains only one contact position. At the beginning and at the end of a DUT test, special actions can be initiated in the program.

The rows of a column will be contacted one after the other. All DUTs of a column should belong to the same type variant.

Row – Contact position

A single column can contain a number of contact positions, i.e. will be contacted. Each row consists of 1..n DUTs or test positions. To test a single DUT the row contains only one test position. At the beginning and at the end of a test row, special actions can be initiated in the program. The DUTs in a row will be contacted one after the other.

DUT (Device under test)– Test position

One test position represents a DUT, which will be tested by the program and will get a test result.

Test sequence

The testing sequence defines the course of tests for a type variant. Thus the test steps are orderly defined in the sequence list. The test sequence is defined before the start of the test and cannot be changed within the test. However, conditional jumps within the test sequence are possible. Thus a test sequence is copy of test steps of that family with the selected type variant and with the selected sequence/ system number. During the loading of test sequence the program also loads the data and the parameters for the corresponding test steps.

Sequence list

For each type family a sequence list will be managed which contains all possible test sequences for all variants of the family. The sequence list consists of sequence steps, comments and filter entries for the sequence steps. Depending on the filters (system number, sequence number, type variant) the required test sequence for the selected system will be constructed on selecting the specific type variant.

Sequence step

A sequence step is the smallest unit of a sequence list. It contains the following information: about the system number which should be activated, the sequence number which it belongs to, the number of the sequence step which shall be started and also information for the conditional jumps. A sequence step only represents only the test step to be executed.

Sequence number

Several test sequences can be saved in the sequence list for each type variant. These test sequences can be different with respect to the complexity of the test. They are defined by the sequence number which has to be activated for each type variant saved in the sequence list.

System number

If there are several TestManager test systems in a production line, a joint sequence list can be made for all the systems. Then each TestManager system has its own system number. This is used to control only that sequence list i.e. loaded into the test sequence when the type variant is selected.

Test step – Test step parameter

A test step normally defines a test instruction. It gets a measured value and a measured result (e.g. pass / fail) after its execution.

A test step also can be used to execute some system settings (set step). In this case the saving of the measured value can be deactivated. The test step data contains additionally the number and the name of the test step, the limiting value, the interpreted step to execute and the parameter for the interpreter step. A test step can be executed so many times as it is repeated in the sequence list.

Interpreter step – IP step

A interpreter step is a user defined procedure, which is executed by a test step. The interpreter step implements the necessary actions to acquire a measured value and then returns it back to the system. A interpreter step gets the parameters from the test step. Thus a interpreter step can be used by several test steps (e.g. current measurement). The necessary information about the measuring channel, the measuring range etc. are supplied to the interpreter step by the parameter database.

Interpreter steps are implemented in a specific MCD programming language which can also be edited by the user.

System steps – System interpreter steps

In addition to the interpreter test steps, some system interpreter steps are also present. These interpreter steps are not used in the test sequence, but only by the system in special cases like program start, program end, test start, test end, etc. Through this an adaptation from the TestManager system to different environments is possible.

Measuring value – Measuring result

In each interpreter step called from a test step a measured value is created and returned to the system. The system compares the measured value with the limits of the test step and determines the measured result of the step (whether pass/fail/invalid). In special cases the result can also be set in the interpreter step or a alphanumeric measured value as text can be set.

Monitor

A monitor is a window in which information about system components are shown and different actions can be initialised. The monitor displays the output of the interpreter or the system automatically. The user can manipulate the system components independently from the interpreter.

MCD Elektronik GmbH	Date: 2018-04-12	Page 17 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

4   Program sequence

4.1   Base program

Directly after program start the cards and address definitions are read (UMSBusses.dat, UMSCards.dat, and UMSAdres.dat see details in attachment). These definitions become a part of the interpreter syntax. Any error occurring during this stage can not be resolved by a program reset. In this case, stop the program, correct the problematic data and restart the program to solve the error.

MCD Elektronik GmbH	Date: 2018-04-12	Page 18 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

During the program start only those program modules which are not switched of in the 'Switch of Modules' section of the basic setup, will be initialised. Modules, deactivated in the 'Switch of Modules' section are absent everywhere in the running system. Thus to reactivate these modules a new start of the entire program is required.

After the initial start-up the program executes an internal reset. Here the modules are initialised and the previous type variant data are loaded. If an error occurs during this reset the program will go into the FAIL state and it is not possible to start a test. The reason for the error must be corrected and the reset must be executed again. The program remains in an error state till the user executes a reset-action or terminates the program.

After a successful reset the entire system is initialised and ready to test with the current values in basic settings. After the first successful reset (or when especially requested to do so in the basic setup) the program executes the interpreter step BOOT. After successful reset, the program executes the interpreter step RESET. Now the system enters the WAIT state. If the interpreter step LOOP is activated, it is executed in a cyclic manner. The system remains in this state until the program is ended, a test is started or any other action is executed by the user. While the user executes an action, the program remains in that state USERDIALOG unless the action is ended by the user. The opening of monitors is not considered as a special user action since the program can continue to work normally.

If a test is executed then the programs enters the state TESTING. The DUTs will be tested in the given order. After the last DUT the measured values are saved ant the program returns to the WAIT state.

If an program exit is called then the interpreter step LEAVE will be executed. All the active program modules will be reset enter a reset and the program is ended.

4.2   Testing

Depending on the selected simultaneous test mode, the test sequence will be executed one time or more than one times, while a single DUT or multiple DUTS might/may be active. Overview: Simultaneous test modes
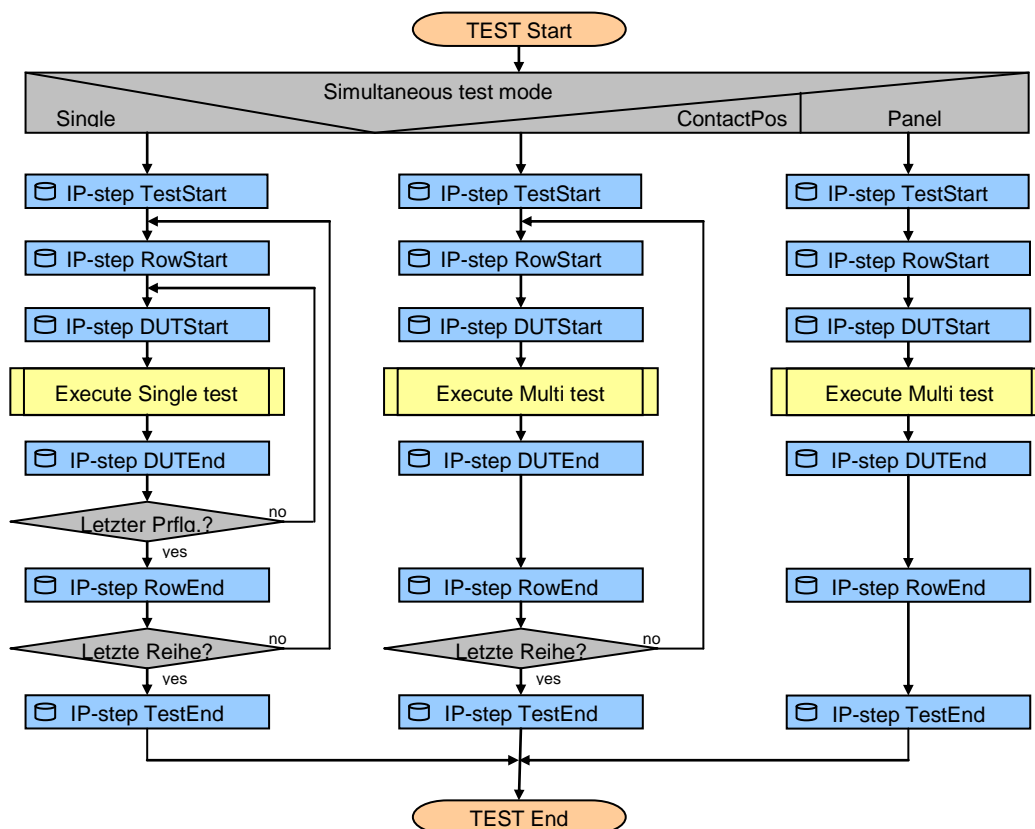
Simultaneous test mode	Number of executions of the test sequence	Number of active DUTs per pass
Single	once per activated DUT	only one DUT
ContactPos	once per contact position	all active DUTs of the contact position
Panel	once	all active DUTs of the panel

Special, new interpreter commands have to be used in the simultaneous test modes ContactPos and Panel to transport the measured values and step results to the system. (see `SetValueEx` etc.)

4.3   Execution of a test sequence

Schematic execution of one pass through the test sequence. Depending on the simultaneous test mode, one or more DUTs will be active.

MCD Elektronik GmbH	Date: 2018-04-12	Page 20 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			
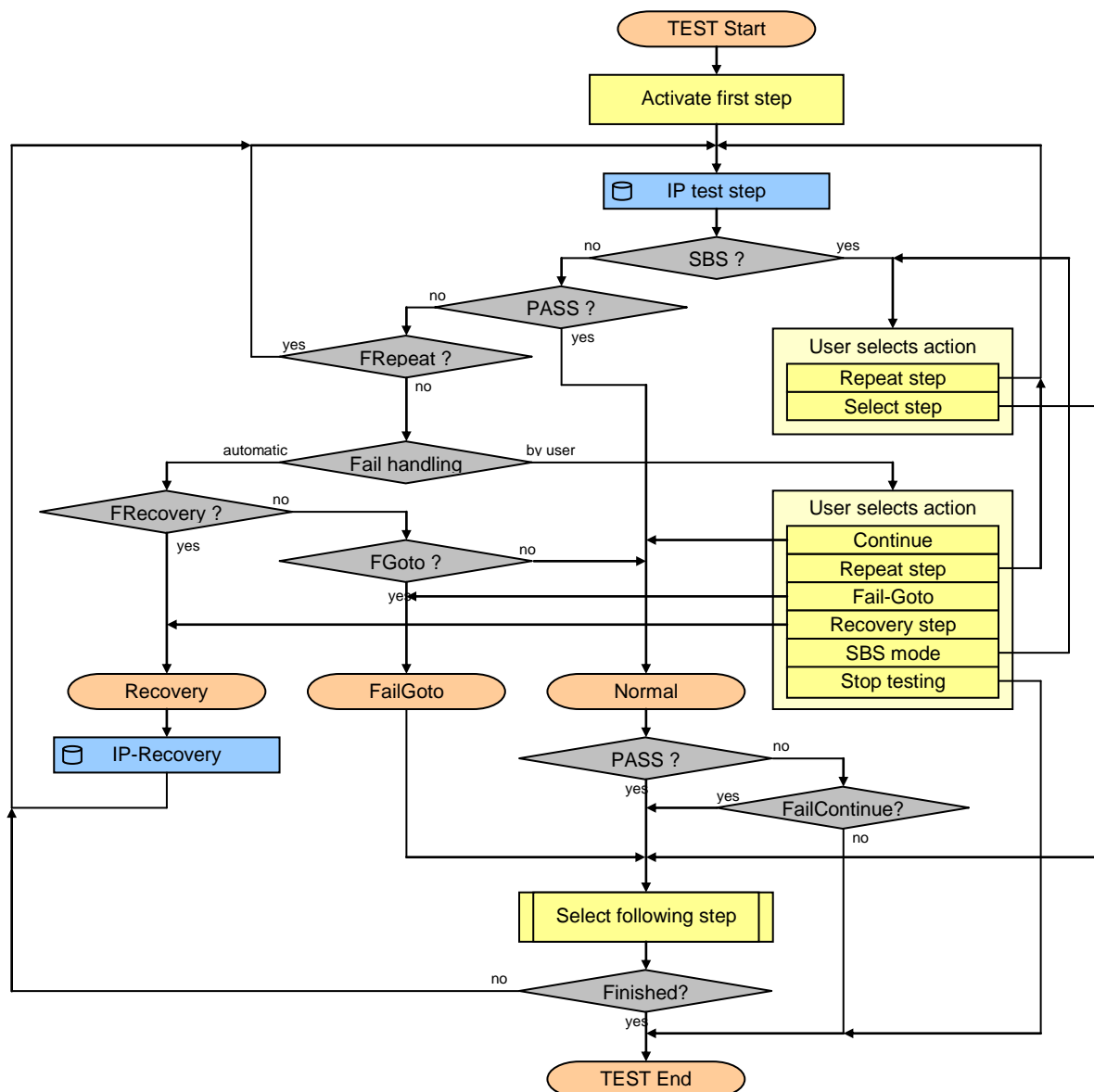
4.4 Selecting the following test step

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 21 of 181	**mcD**® ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

5   System states

5.1   System modes Automatic/ Step by step (Auto / SBS)

Normally the system state is in the automatic mode. The test is executed exactly as described in the test sequence. After having the entire test step results a test result is formed by which the DUT is rated as pass or fail.

From the password level Master the single step mode or step by step mode (SBS) can be initiated. In this mode the system waits after each test step and the operator can choose the next test step within the test sequence. The change over can be done before the test starts as well as during the test.

If any one of the test step is executed in the SBS mode, no test result will be generated for this DUT.

If the system is left in the SBS mode or if the password level is switched back to SBS without the correct user rights, then the system automatically changes to automatic mode.

5.2   System state

The system state represents the entire state of the TestManager program. The different states will be shown as text at the bottom right and using a indicator light at the bottom of the info-column. The following different states are shown:

Waiting

The system is in resting state. A test can be started anytime. Indicator light colour: blue.

Testing active

A test is running. Indicator light colour: green

Error

An error has occurred, which requires intervention of the user as further testing is impossible. At least a reset must be initiated to set the system back to the waiting state. Indicator light colour: blinking red

User dialog

When the user opens a window, e.g. to edit the test sequence or to choose a type variant then no test start is possible and all the background actions are stopped. Only after closing the window the system goes back to its previous state i.e. waiting or error state. Indicator light colour: none, as it further signalises the previous state.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 22 of 181	**MCD** ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

6 General

6.1 Dialog window

6.1.1 Confirmation window

A confirmation window is showed when at a particular point of time the decision of the user is required for e.g. a safety check before deleting a file.

A confirmation window doesn't indicate any error but is a part of the program.
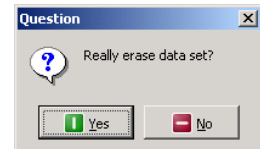
6.1.2 Warning message

A warning window is showed when at a particular point of time the decision of the user is required. They are shown when the user for e.g. exits a procedure without saving the data. A warning window doesn't indicate an error condition and no error status is developed.
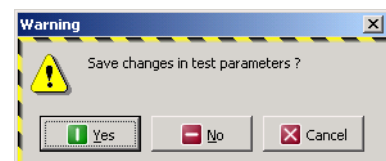
6.1.3 Error message

An error message is shown when an error condition has been generated. For Service purpose, the error number on the window title and the error address along with the error text are important. The error number describes the type of error and the error address tells about its location in the program.

A backup of the error is also saved in the error log file.

After an error message the program returns to an error condition.
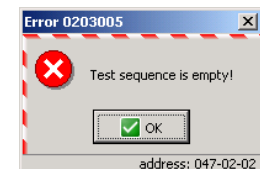
6.1.4 Exception error

An Exception error occurs when the resources required by the program are not provided from the operating system. In the normal program routine such exceptions must not occur. In case they occur they are internally processed by the program. If at all they are unable to be processed, then the error has occurred at an unexpected position (e.g. a DLL). Such errors posses the address 000-00-0 because of their unexpected position.

Such exception error are unable to be solved by the user, thus MCD should be informed and the error log file should be sent along with the error details to MCD. In this log file entire details of the error is described which is highly relevant for correcting the errors.

After exception error an exit program should be executed, because a correct data structure of the program cannot be guaranteed any more.

After an exception error the program returns to an error condition.

7   Main window desktop

The desktop window is always active. When no test is executed or in the fail state the program waits for an action to occur.

The elements of the main window in the normal state are:

Caption (A)

The caption consist of:

Program name and design (here: TestManager)

Program version, having the format: 'Main version, sub version, revision, build'

Details to the actual loaded type variant: Type family, type description, type code.

Menu bar (B)

Main menu of the program to access all the functions. After the program start, many elements are deactivated, because the program starts with password level 'User'. See menu 'Intern / Password' for further information.

MCD Elektronik GmbH	Date: 2018-04-12	Page 24 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			MCD ELEKTRONIK GMBH
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Symbol bar (C)

Contains buttons for direct access to important function as a shortcut to the menu. The number and layout of the buttons can be edited in the basic setting option desktop. The symbol bar can also be removed completely.

Info bar (D)

Shows information about the system. The required elements by the system can be switched on or off. The status lamps which show the actual system status are also present on it (see details in chapter 0).

Main area (E)

In the main area the test result will be shown when activated. Alternatively this area is used by the system interpreter steps to show information.

Status bar (F)

Shows information about the system state. Especially about the actual mode, warnings, hints, the actual password level and status.

The different fields are:

1. Field: Test mode Automatic/ Step by step (Auto / SBS)

2. Field: Information or warnings

3. Field: When the small gear symbol (⚙), is as shown, it means that the program is licensed and can access the hardware ports. But when a read cross (🐞), is present on the gear symbol is as show, it means that the program is in edit only mode

4. Field: Password level

5. Field: Shows whether a system step or a user IP step is active

   ◆   The loop step is activated but it is inactive at the moment

   ◆   The loop step is active at the moment

   ◇   The loop step is inactive at the moment

   ◆   The system is in a error state

   ◆   A user IP step is active at the moment (initiated for e.g. from user tools menu)

6. Field: System status as text

7.1   Menu

7.1.1   Menu type selection

7.1.2   Type selection / Type select

This menu item can also be selected from the symbol bar or by using the F2 key. A type variant must be manually selected if it is not automatically selected in loop step.

The data in the window has an hierarchy structure, with first level as type family, second as type variant and the third as the variants.

When the user having an editing authority is allowed to edit the type lists, then a small checkbox is shown at the right

corner which also enables to display the inactive variants. These variants can be loaded and edited but cannot be started. For the user having no editing authority the checkbox is absent. The inactive variants are not displayed here.

If the selection is confirmed by pressing OK, the chosen sequence number is then listed as default for the type variant. Thus a type change to the chosen sequence is then initiated.

The default sequence number (i.e. the last manually loaded sequence) will be selected automatically as the sequence to be tested. This can however be change by the IP command 'System.SelectTestSequence'

7.1.3   Menu - Measured values

7.1.3.1   Measured values / Show

The menu item Measured values / Show can only be selected, only if the measured values from the previous test are present in the memory. An access is only possible to the measured values of the previous test. Earlier test values cannot be accessed. It is also accessible from the symbol bar or using the F4 key. In the upper part of the window the results of the different DUTs are shown and the required DUT can be selected.

For the selected DUT the different measured values are shown in the lower part. The measured values appears in this list and whether it was pass or fail is indicated by the measured result.

The measured values in the table can be sequentially sorted on double clicking the heading of the column. Another double click on the heading reverses the sorted order. By double clicking the first empty column the original sequence is achieved back.

The duration column shows the time consumed by each step using a bad windows time resolution (approx.60ms). Smaller values other than the system clock is shown as '<1Tick'.

A small triangle (◣)on the upper right corner of the limit values shows that these limits origin from the IP step and not from the parameter list.

A small light blue star (✳) on the left corner of the measured values, shows that a comment is attached to this measured values. The comment is displayed as soon as the mouse is positioned over this star symbol.

When the button 'Report' is clicked the files can be printed. See Chapter 14

7.1.3.2   Measured values / Local statistics

Shows the local statistic of the testing unit. For further information see chapter 13.15.

MCD Elektronik GmbH	Date: 2018-04-12	Page 26 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

7.1.3.3 Measured values / Enumerated data report

Enumerated data is used to transfer system, test, measured values and reports. The data can be called using a number or the respective text. This Enumerated data report shows all the data used in the system.

7.1.3.4 Measured values / other

The menu measured values can be extended by copying e new module the original program. For further information see chapter 13.
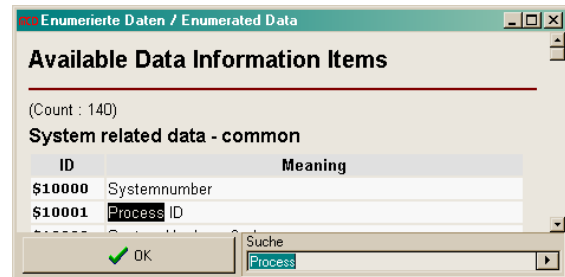
7.1.4 Menu Setup

7.1.4.1 Setup / Registration

Shows the registration form which allows the user to give some information to the system (e.g. his personal

number). The text fields and the keyboard can be defined in the basic settings for the registration form. Shortcut for the registration window is F5.

For system without keyboard, but only mouse or for touch screen systems a monitor keyboard is present to enter the user data.

For further information see module registration chapter 13.3.
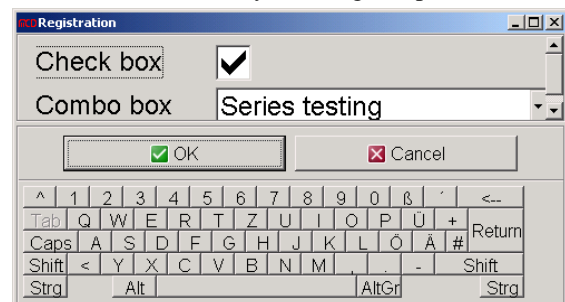
7.1.4.2 Setup / Basic setup

The whole TestManager system can be parameterised with the basic setup configurations. The left side of the window shows the different components in a tree structure. The right side shows the adjustable values for the selected object on the left side.

The layout and complexity of the tree on the left side depends on the requirements of the installed modules. The layout of the right side is defined by the modules.

The 'Configuration' window i.e. shown above has a exceptional position. It acts as a main switch for the system components. Thus the components can be switched off quickly, e.g. when the test sequence should be tested on a office computer. The individual switches are also found on their respective sites.

Shortcut for the basic setup window is F5.
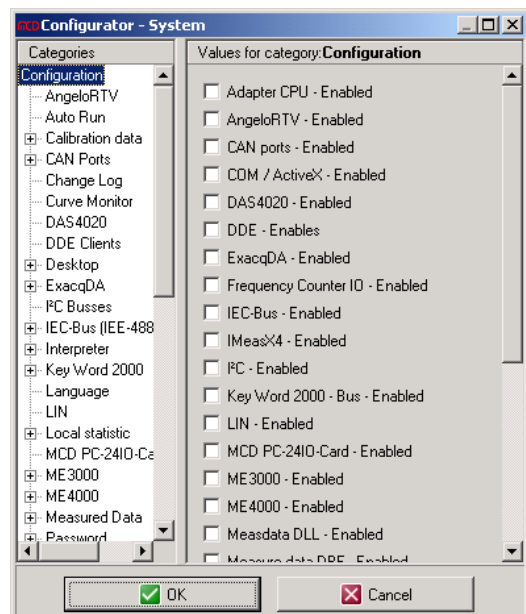
Parameterization of system:

System number
Number of the system. This number is required by the active sequence step.

Process ID
Identification for the testing process. It can be entered in measured value table. It can be asked as an enumerated data item. It is not required anywhere else in the program.

System name

Description of the system. It can be asked as an enumerated data item. It is not required anywhere else in the program.

Company name

Description of the company name. It can be asked as an enumerated data item. It is not required anywhere else in the program.

Area number

7.1.5 Menu Edit

7.1.5.1 Edit / Type list edit

This menu item calls the type list editor. With the help of this editor one can change the existing type variants and create a new variant. Due to its complexity this topic is explained in a separate chapter. See details in chapter 8.

7.1.5.2 Edit / Test program edit

This menu item calls the test sequence editor. With the help of this editor one can change the actually loaded test sequence or the whole sequence list along with the parameter table. Due to its complexity this topic is explained in a separate chapter. See details in chapter 9.

7.1.5.3 Edit / Interpreter steps edit

This menu item calls the interpreter step editor, these editor is also explained later with the interpreter. It is explained in the chapters later how with this editors help one can change existing type variants and create new variants. Due to its complexity this topic is explained in a separate chapter. See details in chapter 12.2

7.1.5.4 Edit / [miscellaneous]

The menu Edit can be expanded by adding new modules to the basis program. Further information is found in the documentation of the modules.

7.1.6 Menu Start

7.1.6.1 Start / Auto – SBS

It acts as a switch between the automatic mode and the single step mode (SBS – step by step). The same function can be accessed over the F3 key or the symbol bar as well. The button in the symbol bar shows the actual state.

7.1.6.2 Start / Active position

It permits the user to select the active DUTs. Normally all the DUTs are set to active and are ready to test. If at all a DUT receives a Fail result in the previous test then it can be deactivated for further tests using this dialogue. This can be done manually using this dialogue window or an interpreter command. Shortcut for this dialogue window is F5. By clicking on the checkbox on the left corner for each DUT they can be activated. Active DUTs are indicated with green and inactive DUTs are indicated in red. By clicking on the row or column header the entire column or row can be selected.

MCD Elektronik GmbH	Date: 2018-04-12	Page 28 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			
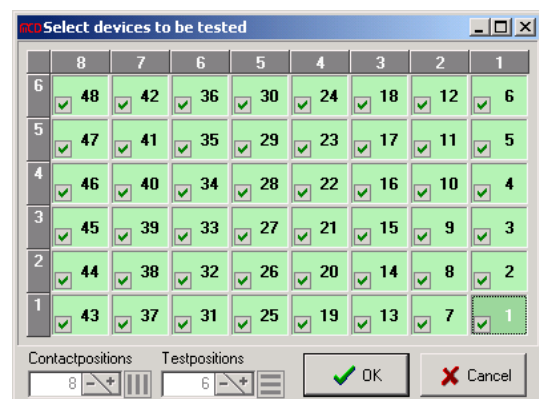
7.1.6.3   Start / Start test

It starts the test manually. If the loop step is inactive then the system can be directly started using this menu. But if the loop step is active then only a flag is set, which has to be regularly checked by the loop step. Thus it is possible to start the test in a the system is in a pre-defined condition. The same function can be accessed over the symbol bar button having an 'up arrow symbol'.

7.1.6.4   Start / Auto run
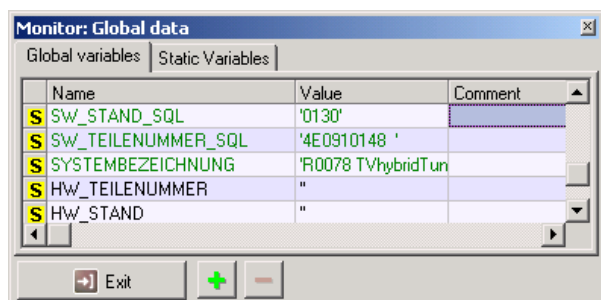
The auto run function helps to repeat the starting of a test automatically. For further information see chapter 13.6

7.1.7   Menu Tools

The menu tools consists of all monitors, showing the current system status and which also can be manipulated. The activity to display the monitors here is parameterised in the basic settings.

7.1.7.1   Tools / Global data

This monitor can be accessed and edited by the interpreter.
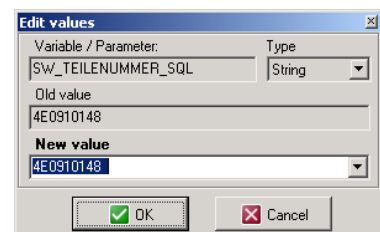
7.1.7.1.1   Global variable

It shows the global variables defined in the interpreter and permits changes to their values. It is also possible to delete or add new variables. The variables are displayed in grid column format.
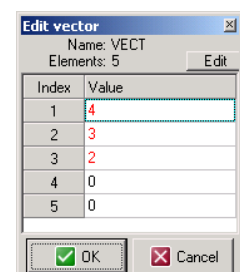
The first column shows the type of global variable R = Real, S = String and V = Vector. The second column shows the name of the global variable. The third column shows the current value of the global variable, while in the fourth column a comment or description can be given for that global variable. By clicking on any of the rows a window is opened which allows to change the values of the global variable. Additionally for a real and string variable the second column can be selected using the cursor button and edited. Invalid values will be rejected. The information in the last column can also be selected by the cursor button to edit any text in it. This text is irrelevant for the program.

The real and string variables can be directly changed but vectors have to be changed using a different dialogue.

This dialogue shows a list of individual values along with the vector name and its number of elements. Die individual values can directly be edited here and are indicated in red on changing them.
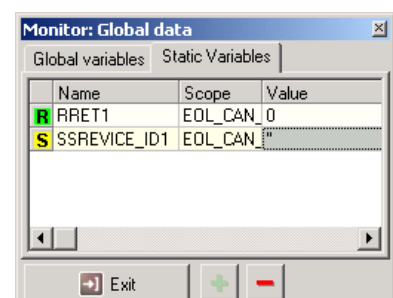
The number of elements of a vector can be edited using the button edit. A small window is opened showing the number of vector elements . If the number is reduced (minimum 0) then the respective rows are reduced in the dialogue. If the number is reduced (maximum 65535) then the respective rows are increased with 0 as initial value in the dialogue.

7.1.7.1.2   Statistic variables

Shows the statistical values defined in the interpreter. These statistical values cannot be edited. However this is the only position where the statistical values can be deleted.

The first column shows the type of variable R = Real, S = String and V = Vector. The second column. The second column consist the name of the variable. The third columns consists the validity of the variable or the

name of the source data, interpreter step, library, procedure or function where the variable is defined.

The last column shows the actual value of the variable. The actual value of the statistical variable can only be changed by editing the interpreter step and executing it.

The statistical variable to be deleted is remembered and marked by a red cross. Any unused statistical variable is immediately deleted.

7.1.7.1.3 COM-Server

The page ‚COM-Server' shows the names that have been used with the interpreter command `COM.Store` to store COM objects into the global COM object table. It is planned to show more information about the stored COM objects in future releases of the program.

7.1.7.1.4 DLLs

The page ‚DLLs' show all DLLs opened by the interpreter together with the registered procedures and functions of this DLLs and the parameter definitions used in the registration command.

7.1.7.2 Tools / IP Monitor

The IP monitor allows the debugging and control of system IP steps. The current IP step is shown as the monitor heading along with the actual program status. The white area below shows the actual debug field, where system messages of interpreter and the output of the IP command 'Debug.Show' are displayed. The first checkbox allows to start the next IP step in the debug mode. The second checkbox stops the program after executing the current IP step and waits for a user action before executing the next IP step. The buttons below help to repeat an IP step,, editing the source of a previous IP step or to start a new IP step.
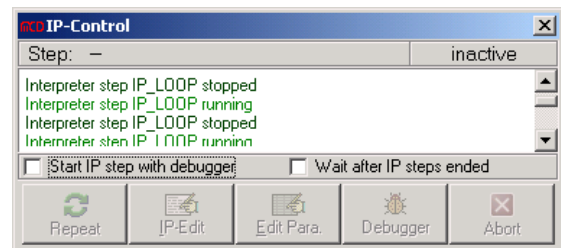
The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.

7.1.7.3 Tools/ Others

If a monitor is not frequently used, but may be still required. This monitor can be set in the basic settings to be shown in the user tools extension.

7.1.7.4 Tools / [miscellaneous]

The menu Tools can be expanded by adding new modules to the main program. For further information see the documentation of modules.

7.1.8 Menu User tools

These monitors are the user defined monitors defined in the desktop option of the basic settings. These monitors are displayed here.

7.1.9 Menu Info

7.1.9.1 Info/ Service

It is an information window showing where and how you can get support when you have problems with the TestManager system.

7.1.9.2   Info/ Log

Consists of a window showing the individual system log files. Any text can be searched for within the log file. The length of the log file is controlled by the program and thus an overflow is avoided.

In all the log files the latest information is always at the bottom. The tabs at the bottom of the window help to switch over between the different log files. The buttons at the bottom help for navigation without a keyboard, changing of text size, searching a text, copying to clipboard and deleting the current log file.
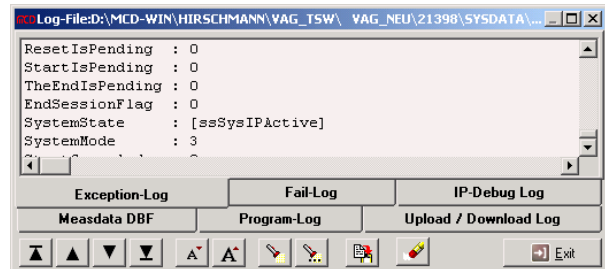
The log file window is not modal, thus it doesn't block the execution of a running program when it is simultaneously open. It is then automatically updated and visualised.

7.1.9.2.1   Info/ Log/ Program log

Contains information about important program events of a normal program sequence. In each row the first position is a time stamp which describes the exact time of the event in the norm ISO-8601 format.

7.1.9.2.2   Info/ Log / Fail log

The occurred problems and errors will be recorded in the fail log file. These events have a ISO-8601 timestamp, too, but can be split across several lines. Almost all the error messages occurred in the program will be recorded in the fail log file.

7.1.9.2.3   Info/ Log / Exception log

Consists of all the exceptional errors occurred in the program (for e.g. the commonly occurred protection error). Similar to the Fail log file and the error can be split across several lines. When an exceptional errors occurs it is important to send this log file along with error description.

7.1.9.2.4   Info/ Log / IP debug log

This file is filled with debug data from the interpreter.

7.1.9.3   Info/ PC Info

Shows information about the PC. With the button 'Report' these information can be saved in a ASCII file. This file can be sent per e-mail.

7.1.9.4   Info/ Update info

This window shows the previous updates made to the TestManager program. The changes in the updated versions can be clearly observed here.
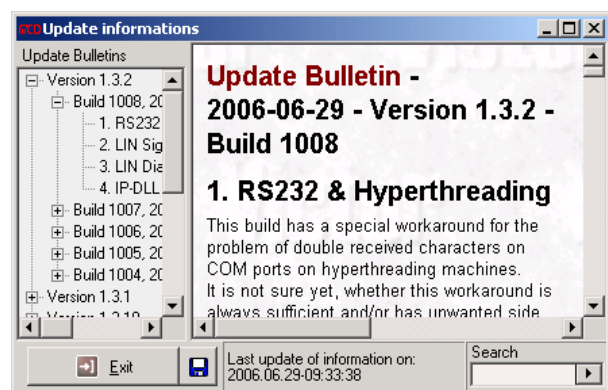
On the left side a tree structure showing the various versions and build numbers is shown. By clicking on a build the respective text is shown on the right side.

The button with the floppy symbol at the bottom helps to save the update info in HTML format. By using the text field 'search' the required text can be searched for from top to bottom. For a new search you must scroll back to the top or by pressing the 'Pos 1' button on your keyboard.

7.1.9.5   Info/ Others

The menu Info can be expanded by adding new modules to the main program. For further information see the documentation of modules.

7.1.10   Menu Intern

7.1.10.1   Intern/ Password

For most of the complex actions other than simply executing a test, you have to the register to the system with a password.

There are 5 different password levels in the system, which can be chosen with F1 – F5. The exact rights for each level can be activated in Menu/ Intern/ Access levels. Normally each of the level is superior to the other and the higher levels include the rights of the lower ones.

Example for access level structure

- User : Execution of tests, starting stage

- Master : Activating / Deactivating of tests

- Service : Creating and maintenance of tests

- Administrator : Comprehensive system maintenance including all rights.

- MCD : Special level only for MCD purposes

The user level does not need a password, for all other levels an appropriate password has to be entered. The input can be displayed in clear text instead of the star symbol. By clicking on the keyboard button a virtual keyboard is displayed or vanished. The input is independent of the upper and lower cases.

If the correct password is entered then the dialogue closes after activating the new password level. If a incorrect password is entered then the dialogue remains open and en error message is displayed.

Hint: Earlier versions had a fixed password for the Administrator and MCD levels. Since these passwords became commonly known it was necessary to remove this fixed passwords, which no more exist. If at all you were using such fixed password and you are now unable to access your passwords, then please contact MCD for service. You shall then receive a password from MCD have a on day validity. This helps you to select a password which can be used for further testing.

A user with administrator rights is able to change the passwords of the master and administrator level.

On the left hand side, the changeable passwords levels are shown (User and MCD password are not changeable).
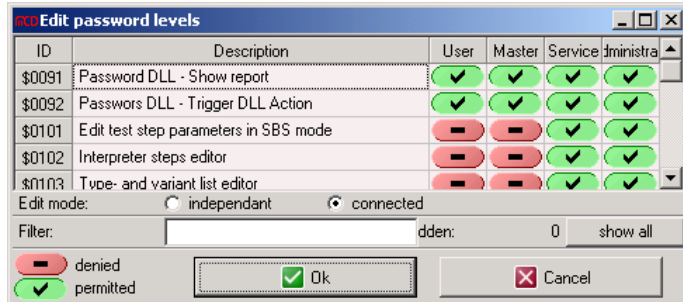
The symbols show the status of the password.

⊗   These passwords cannot be changed because for editing the user should have at least the same password level

🔒   The password has not yet been edited but can be edited

🔧   The password is changed and is valid

❗   The password is changed and is invalid (e.g. too small)

MCD Elektronik GmbH	Date: 2018-04-12	Page 32 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			
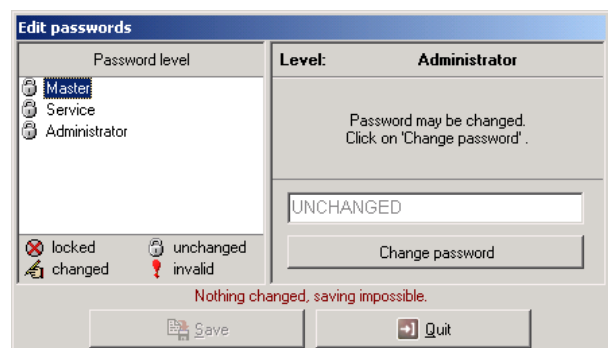
If the password level has to be changed, click on the button 'Change password' and type in the new password. By clicking on the button 'Dismiss changes' the changes are reversed back. Also by clicking on 'Quit' the changes are reversed back. On clicking on 'Save' the passwords is encrypted and saved in a INI–file.

7.1.10.2 Intern / Access levels

As previously described in Intern/ Password, each password level can be defined with different access levels. The individual actions for each of the password level are described in the rows on the left side. In the 4 columns on the right side the access levels to these actions can be granted. The authority for the different access levels can be changed using a mouse click on the symbol. The grey coloured rows are an exception. These field are not changeable by the current access level (for e.g. it is not permitted to cancel the administrator rights to access the different levels). All other actions require permission while some of the actions are reserved for the MCD level. At the bottom in the edit mode if 'connected' is activated then the rights are collectively marked. If an lower level is permitted then the access is also granted to the upper levels, and vice-versa if an action is denied for a upper level all the lower levels are respectively denied. Thus it can be guaranteed that the user for e.g. has no superior rights than an administrator. If the option independent is selected each action can be individually marked. If a text is entered in the edit field of 'Filter' all the action containing this text are marked in red. Thus it helps to find the required action easily.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 33 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

7.1.10.3   Intern / License

7.1.10.3.1   Actual license tab

This window gives information about the actual status of the program license. The display 'Operating mode' shows whether the program is licensed with access to the entire system hardware or whether it has an edit-only license with no access to the system hardware.

The following information is shown:

- whether the program is executed on a Victor hardware platform
- whether a MCD license server is used
- and also the base of the existing license
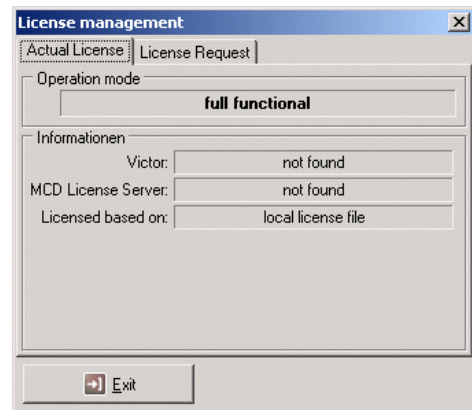
7.1.10.3.2   License request tab

Using this second tab one or more licenses for the same system can be ordered.

Thus the displayed hardware code is received from the running system, while the software code is generated by the program itself.

In the area for number of licenses the quantity (excluding the one present for Victor hardware) of licenses should be entered. If a license already exists for the current system it will be automatically considered by MCD. Thus if for a Victor hardware 4 licenses are required, then a 3 should be entered for the number of licenses.

Important Hint:

The hardware code i.e. to be delivered depends on the access level of the user. Thus under certain circumstances a different hardware code is delivered when the administrator is logged on, as that when a normal user is logged on. Thus it is advisable to generate the license file with those user rights, which will be later on used for the testing system.

In very rare cases the hardware code may not be provided from the system, in such cases please contact MCD for support.
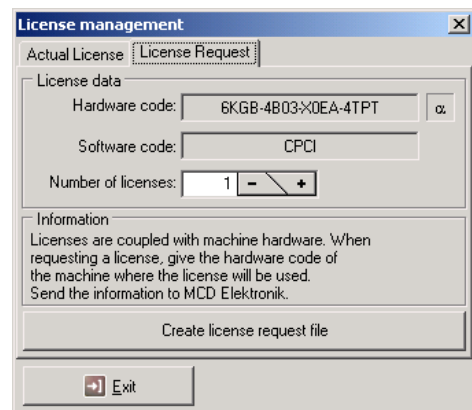
7.1.10.3.3   Short time license tab

The tab short time license is only visible when the program has no valid license.

In order to activate the program for 8 hours, please enter the colour coded numbers on the left to the edit field on right and then press the button 'activate license'.

In the desktop window it will be shown the short time license expires.

7.1.10.4   Intern / Language

The program is able to display all of its outputs, either window labelling or error messages in any desired language. This is possible if the installed windows version supports the language. Unicode character sets like Chinese are not supported.

The user can install any number of additional languages. See appendix. Inside the program one can choose between the installed languages. This language change is also for user defined messages, like user messages during test runs.

7.1.10.4.1   Select language window

In this window there is a button with a flag and description for each installed language. When there are more then 5 languages installed, you can scroll them with the cursor keys. After the selection of a language all labels of the desktop and the comments are changed.

7.1.10.4.2   Parameter-page language

Language
All the languages found by the program in the language directory are shown in the basic settings. The actual language is marked. If here another language is marked here then it is selected during reset.

Language-path
This edit field in the basic settings sets the path for the program for searching the language files. If the edit field is empty the default settings (%EXE%\SysData\Language) are taken. If a path is present then the respective language files are loaded. The same path can be used for loading the UserMsg.Ini file which is used for different actions of the program.

7.1.10.4.3   Automatic translation of user messages

The language files consists only of those text originally required by the main program. The most commonly occurred problem is that the user defined messages from the interpreter steps are not translated.

Thus the user must edit the interpreter step for every new language required. To avoid this confusion the UserMsg.Ini file is present in the same directory of language files. It has the following structure.

```
[TEXTE]
0001011=Funktional tester not ready
0001012=Funktional tester is testing
```

The heading 'TEXTE' must not be changed. After that a set of numbers is present followed by corresponding texts. The numbers should have seven digits and beginning with zeroes. The numbers are the text ID for the following text after the '=' symbol.

Thus the most commonly used messages can be used in the UserMsg.Ini file.

MCD Elektronik GmbH	Date: 2018-04-12	Page 35 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

7.1.11   Menu Help

7.1.11.1   Help / Help

Online help is not available yet

7.1.11.2   Help / About

Shows information about the program. The build number is usually interesting when an error occur. The 'build' number is incremented with minor changes in the program. The 'date' is the day of the program generation.
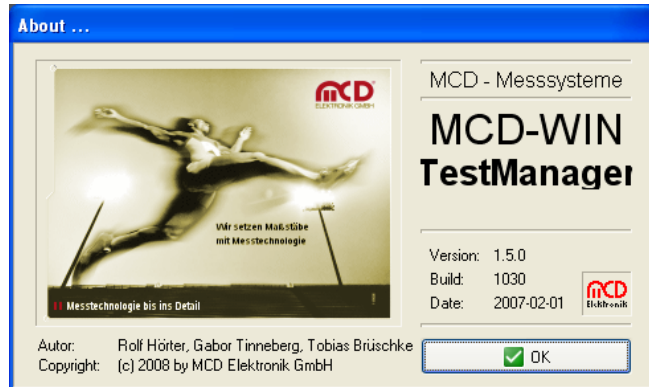
7.1.12   Exit

7.1.12.1   Exit / Reset program

Performs a complete program reset and thus resetting all modules. The type data will be loaded again, all interfaces will be closed and re-opened and the reset step will be executed.

A fail state will be deleted by a reset. The system enters a wait state afterwards.

Exceptions:

- there was a failure during reset itself
- there was a failure during the loading of the program which cannot be cleared by the reset

If an exit to the basic settings is made with 'OK' the reset will be initiated automatically

A rest can also be initiated by the shortcut Alt+F5 or by the symbol bar button.

7.1.12.2   Exit / Leave program

The program is closed only after a confirmation by the user. In the menu Intern/ Access levels the user access level authority can be assigned for safety check before leaving the program.

7.2   Main area of the desktop window

As per the used basic setting different information's can be displayed on the main area of the desktop window.

7.2.1   Waiting status before any previous test

If the program is in an idle position where no object is supposed to be displayed in the main area (no test results or interpreter objects to be shown), then the program attempts to display a background picture in the main area.

Thus it searches in the actual type directory for a graphic file with the name 'Back' with data type .BMP or .JPG. If no such file is available in TypeData then the SysData is searched for this file. Thus a type depending background picture can be selected.

In the absence of any such file a coloured background is shown.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 36 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

7.2.2 Displaying the test results

The test results is automatically displayed after a test in the main area, if the corresponding option is selected in the basic settings. If there are a number of fail steps for a test, then each individual failures can be observed by scrolling the buttons below. Similarly if there are many DUTs in rows and columns each DUT result can be observed by scrolling the buttons below.

7.2.3 Display of the interpreter

The screen object of the interpreter occupies the whole main area of the desktop window. It overlaps all other outputs.

7.3 Parameter page of desktop

This basic setting option decides for the display of the desktop.

Info-column

It provides the visibility for the elements on the info column. It is the right hand column of the desktop window which cannot be changed.

Show info-column
Decides for the visibility of the entire info column. If the info column is not activated then all its elements remain invisible.

Logo (User defined or MCD)
Decides the visibility of the logo. The type of logo is described further below.

Analog clock

Decides the visibility of the clock on the info column.

Flag for language selection

Decides the visibility of the flag for language selection. Along with the menu the language can also be changed by clicking on this Flag symbol.

Type –Picture (%TYP%\TypePic.*)
Activates or deactivates the type picture. Thus it searches in the actual type directory for a graphic file with the name 'TypePic' with file type .BMP or .JPG. The picture if found is displayed or else a red cross is shown.

Local statistics
Decides the visibility of the Local statistics on the info column. The content of statistic to be displayed is decided further below.

Show state lights
Decides the visibility of the state lights which represent the system status. When the state lights are deactivated the program status is still visible on the status bar.

Show

Show DUT result

If it is activated in the configurator, then the test result is displayed in the main window after the test. For further information see chapter 7.2.

Logo window

Show user defined logo

If it is activated in the configurator, then a user defined logo is displayed instead of an MCD logo.

File name

The path of the user defined logo with file type .BMP or .JPG can be defined here.

Transparent

If it is activated, then a user defined logo is displayed in its original colour and invisible to the background.. Thus the logo is shown transparent with the same background of the desktop window. This function should be used only when the logo is a BMP file, because JPG files do not allows possess the original colours.

Mode

If at all the logo has a different size as that of the logo window (70x39 pixel), the following options are available:

- Centred   The logo is shown in its original size in the centre of the logo box.
- Fitting   The logo is expanded or shrunk independent of the sides to the size of the logo box.. The borders are removed.
- Single   The top left corner of the logo in its original size is placed on the top left corner of the logo box.
- Stretched   The logo is expanded or shrunk depending on the sides to the size of the logo box.. The borders may remain.
- Tiled   The logo is tiled or partitioned to an extent that so that it fits in the logo box.
- Adjust to logo   The breadth and height of the logo is scaled and positioned to the breadth and height of the logo box.

The best result is obtained when the logo is transformed with a picture editing program to the size as that of the logo window. Then it should be scaled with the mode 'Adjust to logo'.

Application Window

Window size

This option allows the selection between the screen size for old used touch screen victor monitors with size (800x600 Pixel) and to any other changeable screen size (limited to 400x300 to the present monitor size). Option for background bitmap (%TYP%\BACK.BMP resp. %SYS%\BACK.BMP)

Mode

Shows the different modes to display the background bitmap (see chapter 7.2). For further information of the different information see the description above in 'Logo/ mode'. However the mode 'Adjust to logo' is absent here because the size for the background bitmap depend on the size of the application window.

Miscellaneous

Colour of the clocks second hand
Helps to change the colour of the clocks second hand, perhaps to that of the logo.

Show image in menu
Helps to activate the small images in the left corner.

Type picture

Transparent
Shows the type picture in transparent mode on the info column. For further information see 'logo/ transparent'.

Mode
Shows the type pictures display mode on the info column. For further information see 'logo/ mode'.

XP themes

Support XP themes
Allows to select the program to use the XP themes. Microsoft has made it possible through the Windows XP® that the user can define the programs overview features to his own choice.. The program must also support this feature. The victor program should have a common overview in all the operating systems. And hence this option is present for the user to change the program overview. It occurs only when:

- The operating system is through Windows XP®

- In the basic setup/ desktop properties Support XP themes is selected

Otherwise or when the option is deactivated the program is viewed in a classical overview. When the option is activated, an executable file 'TestManager.Manifest.exe' is created and then deleted in the program directory. This file supports the XP operating system, to executes the new overview. After activation of this option the victor program must be restarted again.

Old view(classical): ✅ OK   New view (XP): ✅ OK

7.3.1 Parameter page of desktop/ Speed tool bar

Shows the overview of the buttons on the desktop symbol list.

Execution:

Type of speed tool bar:

- standard – is a fixed defined tool bar. It ignores the settings made blow.
- user defined.-.Activates the settings made blow.
- none.-.No tool bar is shown

Buttons: Set user defined bar to standard buttons
Redefines the tool bar with the standard tool bar settings. These buttons can further be edited. The buttons of the standard tool bar are: Start test, type select, registration, local statistic, automatic/ SBS mode, show measured value, reset, end program.

Buttons: Set user defined bar to editor buttons
Redefines the tool bar with the editor bar settings. These buttons can further be edited. The buttons of the editor tool bar are: Start test, type select , set-up, type list editor, test sequence editor, interpreter editor, show measured value, reset, end program.

User defined speed tool bar
Shows the current over view of the tool bar. By clicking on a button, it is activated and ready for editing using the options below.. The active button is also light coloured.

Select buttons

Edit button
Helps to select the active button. By clicking on a button, it is activated and ready for editing using the options below. The active button is also light coloured.

Move left
Moves the active button on the tool bar to the left by one position.

MCD Elektronik GmbH	Date: 2018-04-12	Page 39 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Move right
Moves the active button on the tool bar to the right by one position.

Delete
Deletes the active button on the tool bar (after a safety check). It is possible to delete all the buttons.

Add
Adds a new button at the end of the tool bar. This button can then be moved to the left or right. The maximum number of buttons which are correctly fit on the tool bar are twelve.

Edit buttons
Shows details of the selected button and helps to edit it.

Action
Selects the action to be executed by that button. The list to select the actions depends upon the active modules (see chapter 15) selected in the basic setup.

Parameter
Parameter for the action. Only the data permitted by this action parameter can be entered here.

Parameter Inf.
Information for the action parameter can be entered here.

Text ID
Defines where the button ID text should be obtained from, USERMSG.INI or default text (see chapter 7.1.10.4.3). If the value entered is 0 then USERMSG.INI is ignored and default text is selected. If the value entered is -1 then the internal program values are used.

Default text
It is the text to be used as the button ID when the USERMSG.INI is not found.

Bitmap
Selects the path for a bitmap, and uses the default bitmap when the path is empty.

7.3.2 Parameter page of desktop/ Menu

Show shut down in exit menu
It activates the shut down option in exit menu. This executes the system action 16 (see chapter 15).

Parameter for shut down in exit menu

Parameter for shut down system action 16 (see chapter 15).

7.3.3 Parameter page of desktop/ User tools

The user tools menu is managed here.

Show user tools menu
It activates the user tools in user tools menu.

User defined menu

Items in Menu and in Overview – (Right click shows menu as a popup menu)
Shows an overview of the elements of the user tools menu. An element can be selected and activated using a mouse click.

Select menu item

Edit menu item
Helps to select the active element. By clicking on an element, it is activated and ready for editing using the options below. The active button is also highlighted.

Move up

Moves the active element on the tool bar upwards by one position.

Move down

Moves the active element on the tool bar downwards by one position.

Delete

Deletes the active element on the tool bar (after a safety check). It is possible to delete all the element.

Add

Adds a new element at the end of the tool bar. This button can then be moved to the left or right. Any number of elements can be fitted to this menu.

Edit menu item

Shows details of the selected element and helps to edit it.

Action

Selects the action to be executed by that element. The list to select the actions depends upon the active modules (see chapter 15) selected in the basic setup.

Parameter

Parameter for the action. Only the data permitted by this action parameter can be entered here.

Parameter Inf.

Information for the action parameter can be entered here.

Text ID

Defines where the button ID text should be obtained from, USERMSG.INI or default text (see chapter 7.1.10.4.3). If the value entered is 0 then USERMSG.INI is ignored and default text is selected. If the value entered is -1 then the internal program values are used.

Default text

It is the text to be used as the element ID when the USERMSG.INI is not found.

Bitmap

Selects the path for a bitmap, and uses the default bitmap when the path is empty.

8 Type list editor

Type variants and type families can be created, modified and deleted using the type editor.

The changes occur only in memory (only with an exception of rewriting of type families, see below) and take place only when the save button is pressed.

The editor is in beta state and hence is explained further in detail.

Please see the module Upload/Download for further information.
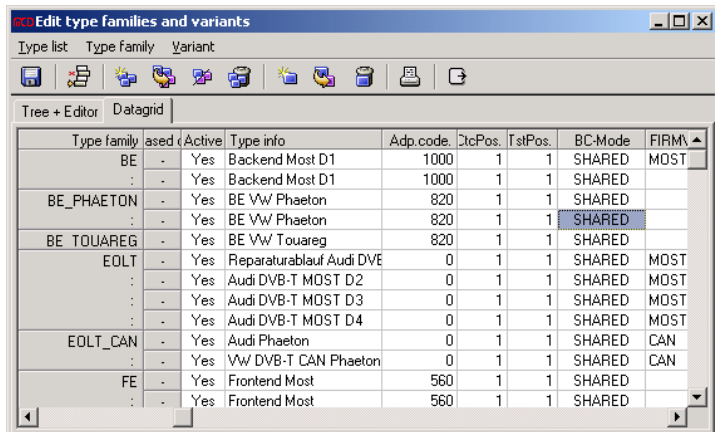
8.1 Tree editor

In the tree editor view all the defined families and variants are shown on the left side of this window. The highlighted variant on the left is shown in detail on the right and can be edited. If a family is selected on the left no user defined field is shown on the right. The program defined fields are shown on the top while the user defined fields are shown at the bottom.
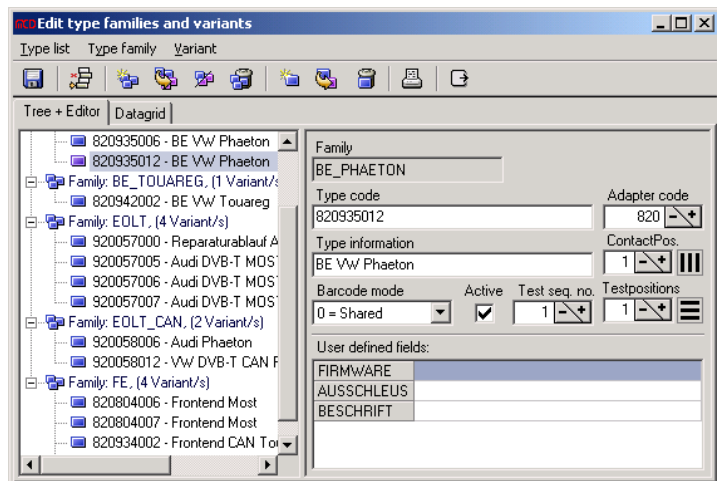
8.2 Data grid

Similar to the tree structure in the data grid also the values can be edited. Here the first column consists of the type family, while the second column (based on) shows the belonging of the type variant when a new one is made. The values of a type variant in the grid can be edited.

8.3 Fields

8.3.1 Program internally used fields

Family/ Type family	Shows the name of the type family to which the variant belongs to. As the family name is used in the file and as directory names, the name of the type family cannot be changed directly. The type family name can have indefinite number of characters and can contain alpha numeric characters. But only for the data name valid characters are required. All variants with the same family name belonging to a single type family.
Type code	The type code is an unique identity for that type variant. In complicated processes, with the help of type code the appropriate type variant and test sequence can be chosen automatically. If a particular type code is repeated then its colour changes to red. In the tree editor the entire edit field turns red.
Sequence Nr.	It shows the sequence number for the test sequence, which is loaded automatically on selection of the variant.
Active	Allows the disabling of a variant, when it should not be used. In the data grid the values '0' or 'N' for 'No' or '1','Y' or 'J' for 'Yes' can be entered.

MCD Elektronik GmbH	Date: 2018-04-12	Page 42 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Information/ Type info	Description of the type variant is shown as information, but is not used anywhere.
Adapter code	It helps to crosscheck the correct DUT adapter for the selected type variant. This analysis should be done by an interpreter step and not by the system. The adapter code is a number in the range 0..999
Contact positions	It is the number of rows in the type variant panel. For a single DUT tests a '1' must be entered
Test positions	It is the number of DUTs in a contact position. For a single DUT tests a '1' must be entered
Barcode mode	Describes how the barcode will be managed. If 'SHARED' is used (value 0) then the whole column receives a single barcode. Each individual DUTs is then identified with that barcode plus an incrementing number. . If 'SEPARATE is used (value 1) each DUT has its own barcode. For single DUTs this option is irrelevant.

8.3.2   User defined fields

In addition to the program internally used fields, use defined field can be added (see Menu 'Type list/ Change structure'). Values from this field are not used by the program but can be used by the interpreter or as enumerated values.

In the data grid these fields are shown in separate columns after the program internally used columns. In the tree editor these fields are shown in separate rows in the part of the window.

8.4   Menu Type list

8.4.1   Save type list

Saves the actual status of the program in the file %TDR%\TYPELIST.CSV.

Due to the function of backward compatibility the files can be additionally saved as %TDR%\TYPELIST.DBF. This helps the older software versions to read the program files. However this option may not be present in future.

The old type list files are rewritten as TYPELIST.~CS and are not deleted.

8.4.2   Menu Type list/ Print report

Opens the report generator to print the files of a single variant or all the variants.

8.4.3   Menu Type list/ User defined fields

Opens the window to edit the structure of the user defined fields.

The defined fields are shown in a grid. Using the up down arrows on the tool bar the user field can be moved up or down.

Using the '+ New' symbol on the tool bar a new field can be added.

Using the '- Delete' symbol on the tool bar the highlighted field can be deleted. It is not immediately deleted but a '-' minus symbol is shown on this field in the centre (Action). Thus this field has been marked to be deleted

Using the U⇗X Symbol (Name) selects and marks the field whose name is to be edited. Also when a text is edited in the 'new name' column this field is marked.

MCD Elektronik GmbH	Date: 2018-04-12	Page 43 of 181	MCD ELEKTRONIK GMBH
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

To Using the '=' Symbol (No change) the marked areas can be assigned with their old names.

Using the 'OK' button the window can be closed after saving the data. Only if no errors are present the 'OK' button is visible. E.g. for errors are: Repeated field name, an empty field name.

The number of field are unlimited.

The contents of the deleted fields are lost in the type list and the new fields created are initially empty. Edited fields receive the new names.

8.4.4 Menu Type list/ Quit

Exits the type list editor. If any logical errors are present (for e.g. is same type codes are assigned) then a safety check appears, whether the user really wants to exit the editor.

If the changes are not saved, then the safety check appears, whether the user really wants to exit the editor without saving.

If the data is saved a reset is executed in order to load the new type lists.

8.5 Menu Type family

8.5.1 Menu Type family/ Create new type family

It helps to create a new type family. A dummy field is created for the new family, which can be edited later. At the top a list of the existing field are shown. It is not required to mark any of them., but the new family name must be entered in the text field at the bottom.

The new family name can consist of alpha-numerical characters and special symbols as used in the file names.

The 'OK' button is only active when a valid family name is assigned. Empty values or existing names are not permitted.
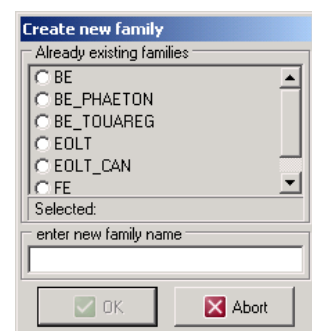
8.5.2 Menu Type family/ Derivate type family

To deviate a new type family the existing family is copied and a new family name is assigned to it. Thus a new type family is created along with all its sub-components. Normally the type codes are also copied, which must be edited later on.

At the top a list of the existing field are shown. It is required to mark any one of them., and also the new family name must be entered at the bottom.

The new family name can consist of alpha-numerical characters and special symbols as used in the file names.

The 'OK' button is only active when a valid family name is assigned. Empty values or existing names are not permitted.
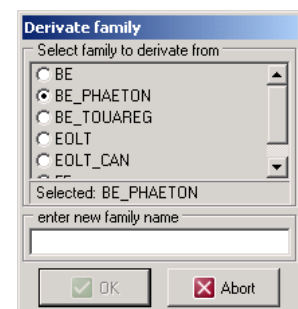
8.5.3 Menu Type family/ Rename type family

For renaming a type family the entire directory structure on the hard disk should be manipulated.

Thus all the previous changes made in the type list editor should be saved.

Since family name is present within this directory and file names, on renaming the file names they are automatically edited.

At the top a list of the existing field are shown. It is required to mark any one of them., and also the new family name must be entered at the bottom.

The new family name can consist of alpha-numerical characters and special symbols as used in the file names.

The 'OK' button is only active when a valid family name is assigned. Empty values or existing names are not permitted.

8.5.4 Menu Type family/ Delete whole type family

Deletes the entire type family along with all its sub components after a safety check.

The last type family cannot be deleted.

The type-data directory remains at the end and cannot be deleted. These is to avoid a loss of the entire data due to any accidental deleting. If the directory is really not needed anymore it can be deleted using the explorer.

8.6 Menu variant

8.6.1 Menu variant/ Create a new variant

Creates a new variant as a member of the active family. The values of this new variant can then be edited.

8.6.2 Menu variant/ Copy a variant

Helps to copy the selected variant within the active family.

8.6.3 Menu variant/ Delete variant

Helps to delete the selected type variant. If the last member of the type family is deleted, then the entire type family is deleted. See chapter 8.5.4.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12 Revision: V1.4	Page 45 of 181 Author: SW department	MCD ELEKTRONIK GMBH
Title: MCD TestManager CE - Software Manual			
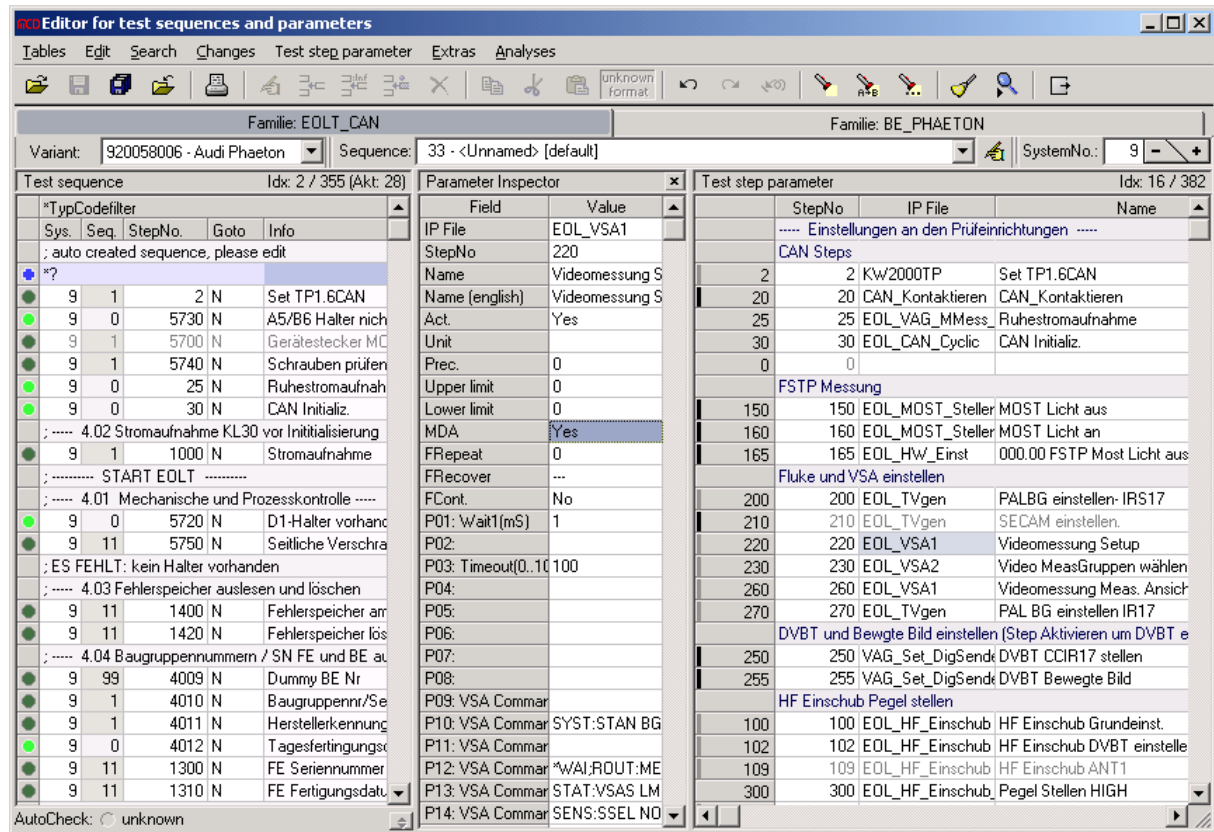
9 Test sequence editor

In the test sequence editor the data for the selected type family is shown which can be then edited. If the data of another family is to be edited, then a variant of this family or the new family has to be selected. Multiple number of families can be activated and edited simultaneously.

The window of the test sequence consists of the menu bar, the tool bar and the active editing field. In this editing field the active families are shown with individual tabs with the respective family names. The user can switch between the families by clicking on the tabs or by using the shortcut Ctrl-Tab. However the selected family is distinguished from the rest by a coloured tab.

The tab for each family consists of the activation bar with the selection field for variants. The editing field below is further divided to three sections the test sequence list, the parameter inspector and the test step parameters.

Important: When editing the sequence list or the test steps these changes take place in all variants of that type family, because all variants share these data. This is however useful, because with this a change concerns all variants and therefore has to be done only once. However unintended changes can also be made to other variants.

With the start of the test sequence editor the data is loaded in the memory. All the changes made take place only in the memory. Changes which are not be saved are hence lost.

For further information see the module Upload/Download, chapter 13.36.

Each family has its own setting for the editing field. These setting are saved as far as possible and are restored back next time when this family is opened..

The test sequence list and the test step parameter list have an unlimited 'Undo/ Redo' function. Thus the changed steps can be recovered back till almost the last memory position.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany	Date: 2018-04-12	Page 46 of 181	
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Edit fields which are edited since the previous opening of the test sequence editor have a coloured background. The changes of an edit field can be recovered back till almost the last memory position. This action is independent of the 'Undo/ Redo' function and thus can also be reversed.

9.1 Mouse commands

Along with the usual windows mouse functions this editor supports some additional functions.

Changing the column width

By stretching the column divider at the top of the sequence list or the step parameter list the width of the columns can be adjusted. By double clicking on the right column divider the program automatically sets the column width such that all the text is properly readable.

Changing the column sequence

The sequence of the column can be changed by dragging that column with the mouse.

Changing the row sequence

The sequence of the rows can be changed by dragging that row with the mouse. It is also applicable if many rows are selected to be moved.

Extending the selection to the entire row

By clicking on the first column of the sequence list or the step parameter list the entire row is marked.

Dragging of steps

The test steps can be dragged from the step parameter list to the sequence list. The entire row should be marked first and then be dragged sequence list.

Double clicking a field

When the selected field possesses a logical value or an incrementing value, its value can be altered by double clicking on it. The logical value is negated or toggled and the incrementing value is incremented by double clicking.

All other fields show no changes on double clicking.

9.2 Keyboard commands

Most of the keyboard commands are shown in the menu. The additionally used keyboard commands for the users convenience are:

Cursor key

Helps to navigate to select the required field

Shift +Cursor

Extends the selection by adding the next block

Ctrl + Enter

Opens the test step editor for that step

Ctrl + R

Extends the selection to the entire row. The operation is then executed on the entire row and not on the rows contents

Ctrl + Del

Deletes all the selected rows. It functions like the command Ctrl + R+ delete. The basic principle to ignore the invalid values entered, and the old value are shown again.

Ctrl + Space

When the selected field possesses a logical value or an incrementing value, its value can be altered by this command. The logical value is negated or toggled and the incrementing value is incremented by this command.

All other fields show no changes on using this command.

F2/ Enter/ any alpha key

Activates the field editor to edit the contents of this field. If the editor is activated by any other key other than F2/ Enter then the field contents are replaced by the key character. Using the ESC key an active field editor can be discontinued without changing its contents. By leaving the edit field after pressing the Enter key or any other character, the original contents are replaced by the new. When the selected field possesses a logical value or an incrementing value, then instead of the normal edit field a selection field having the required options is displayed.

9.3   Toolbar, context menu

All the buttons in the toolbar and context menu function similar to the options of the menu of the test sequence editor (see chapter 9.9). The 'save' button is an exception, and it saves either the sequence list or the test parameter list whichever is active.

9.4   Activation bar

In a test sequence editor the entire type family can be edited and all the test steps are visible. On an activating a test sequence the test steps are however filtered with respect to the type code, sequence and system number filtered (see details below).

In order to display the active steps in the editor the information regarding the type code, sequence and system number should be provided. These can be done by selecting the correct option from the activation bar. The variant selects the type variant with respect to the type code i.e. then compared with the variant filter. The sequence selects the sequence number i.e. then compared with the values of the test step number. Similarly the system number is also compared.

The selected values influences only the view in the editor and has no influence on the currently loaded variant and system number.

The default values for any loaded type variant is the standard sequence number and the system number as given in the basic settings.

Due to the activation bar it is possible to view the activated steps for a particular combination of typecode, sequence and system number.

The 'hand with pen' symbol near the sequence edit field helps to edit the sequence names (see chapter 9.16).

9.5   Sequence list

The sequence list describes the sequence of the test steps to be executed in a test. Hence the sequence list can contain more then one test sequence or sequences for more then one type variants. The used test sequence is extracted from the test sequence list of the variant only after the type selection and the loading of the type data.

The selection of the steps in the actual test sequence list depends on the set filters. These filters are: Type code of the type variant, sequence number and system number. The functions of the filters are described below. Whether

a test step row in the sequence list is active for the chosen type variant is shown by a marker in the first left column. If this marker is bright, then this row is active for this variant, otherwise it will be ignored during the loading of the test sequence.

Values in the sequence list can be edited directly in the data grid directly, by typing the new value or by changing to the edit mode using the Enter key to edit the entry. Alternative a edit window can be initiated with the shortcut Ctrl + Enter.

If for the selected step in the sequence list a test step exists then this step is automatically highlighted in the test step parameter list.
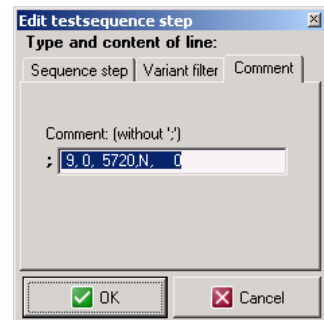
A row in the sequence list can contain either a comment, a code filter or a sequence step.

The different possibilities of a value in the sequence list is explained below:

9.5.1   Comments

Comments are not noticed by the program and can be used to describe the steps in the sequence list. The background of the comment rows is grey. The length of the comments is not limited. For comment rows no symbol will be shown in the first column, because they don't have a effect on the test sequence. Comments can be easily edited in the edit window. If you edit the comment directly, you have to ensure that there a semicolon is present as the first character on the row, or this may lead to an error.
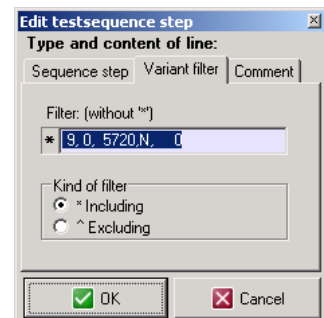
9.5.2   Type code filter

Type code filters activate or deactivate the following sequence steps. These filters are used to decide which sequence steps should be executed for which variants.

There are two types of variant filter including and excluding filter. The including filters have a light blue background, start with a star(*)symbol and have a (+)symbol in the first column. While the excluding filter have a light red background, start with a roof (^) symbol and have a (-) symbol in the first column.
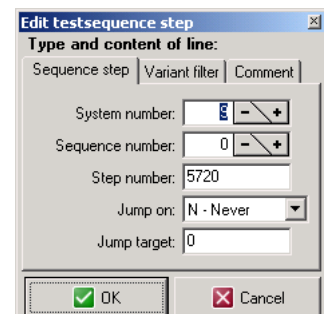
On using the edit window the start and roof symbol are automatically assigned. But if you edit the filter directly, you have to ensure that there a start and roof symbol is present as the first character on the row.

Type code filters have a marker in the first column. This colour of the marker is light if the filters suits the type code of the active variant and dark if the filter does not suit the type code.

A filter matches a code only if the characters of the filter and the code are the same. If a code has more characters as a filter, then this will be ignored. The following symbols perform special functions:

?   represents to any other character in the type code

*   represents to zero or any other character in the type code

[..]   represents to the characters present in the brackets [] in the type code.
         All the characters can be written in [123456] or in short using '-'symbol
         [1-6]. The first character should not be a '^'.

[^..]   represents to the characters in the type code, when it is not present in
         brackets. Values can be written as earlier one accept that a '^' should be
         present at the beginning.

The shortest possible filter is '*?'. This filter matches all type codes and should always stands on the beginning of the sequence list.

9.5.3 Sequence steps

●	1	1	1300	N	Stromaufna

The actual test sequence consists of the sequence steps. A sequence step consists of 7 fields (The jump information is shown together in the test step number field of the grid):

Activity

The activity of the step is indicated by the colour of the symbol in the first column. If it is light coloured the step is active and if it is dark the step is inactive and will not be loaded.

System number

The system number is for filtering and decides on which system the step will be executed. It has to be the same as the system number entered in the basic settings. The system number '0' in the sequence step implies that the step will be executed on all systems. A system number which is not matching will be displayed with a grey background, the number '0' with a light grey background.

Example to use the system number: A test sequence for a product is created, tested and installed on a system in the production line. Later on a second test system is introduced because the testing time has to be reduced to increase the production. It can use the same data (sequence list) as the first system, only the sequence steps will be distributed over the test systems by adjusting the system number to achieve a short testing time. Thus it is sufficient to maintain the data only on one place, because both systems work with a identical list.

Sequence number

The sequence list allows to save several test sequences in one sequence list. The sequence number present in the type list is used to decide which sequence will be loaded. Only those sequence steps are loaded in the test sequence, whose number is identical with the number in the type list or whose sequence number is '0'.

Also a part of the sequence can be omitted out for test purposes by assigning these sequence steps a sequence number i.e. not in use.

A sequence number which is not matching will be displayed with a grey background and the number '0' with a light grey background.

Test step number

The test step number makes a connection to the actual test steps. In the test sequence each test step is called by its number and is respectively loaded and executed during the test. If there is no test step with the specified number, so this number will be displayed in red. If the sequence list is loaded later, then a missing test step generates an error while loading.

Jump on – Jump target

The 'jump on' defines if and when after the execution of a test step the normal test execution should be stopped and when a jump should be executed.

Types of jumps:

N Never   Never jump, jump targets listed will be ignored

P Pass   Jump on pass

F Fail   Jump on fail or invalid results

A Always   Always jump, don't care the result

C Calculated   Interpreter step calculates whether a jump is to be made

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 50 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

A test step number is specified for the jump target. The system searches the next occurrence of this test step number in the sequence list. Then for a 'Always' and 'Calculated' jump the step is searched first in the forward and then in the backward direction. For 'Fail' the step is searched first in the backward and then in the forward direction. For a 'Pass' jump is only the backward direction possible.

In the sequence list jump type and jump target is displayed together in one row, separated with a dash ('P – 1000'). By directly entering the value the jump type and jump target should be entered without any separation ('P1000').

In case of a 'Calculated' jump the jump is manipulated by the interpreter step.

When the jump target is absent the field is marked in red.

In case of a 'Fail' jump which cannot be executed because the called step is currently occupied by a fail recovery action, then the field is marked in orange.

Info

In the last column the name (i.e. not editable) describing the respective step number is shown.

### 9.5.4	Fail row

A fail row is a sequence list row, whose contents can't be interpreted by the program. It can arise by writing wrong values (for e.g. the semicolon of a comment is deleted) or due to a corrupted data in the type files. A fail row is displayed with a red background and a light red marker 'E' in the first column. Sequence lists with fail rows cannot be loaded by the program.

## 9.6	Test steps

Test steps define only how the required test action should be performed, but not when they should occur. Test steps can be used once or more often in a sequence. The test steps on the right side of the editor thus form a pool, from which the sequence on the left can be built.

Each test step consists of several fields. The test step number thus ha s a high importance. It is the key forming the link to the sequence list.

Along with test steps the parameter list can also contain a comment row. They can be used to summarise and describe the test blocks logically. They are not considered by the program.

Fields of the test steps:

Hint: The sequence of the fields can be altered on dragging with the mouse or by the corresponding menu option.

Step Nr

Test step number is the key and link to the sequence list. It should be unique. If a number is repeated several times, then only the first appearance is considered and the remaining is marked in red. The number should be maximum 8 digits without a decimal point and commas.

IP Step

Name of the interpreter step to be executed but without '.IPS'. The step is then searched in the interpreter step directory. The name can contain a directory, which should then be a sub directory of the interpreter step directory.

Name / Name (English)

Description of the test step in the language of the country and in English (or any second language). In the basic settings the user can choose which language field should be used in the display. The description will be displayed in the test window and is used for saving the measured values

Active

Activity of the test step. Inactive test steps (as value '0' for NO) will be loaded, but not executed. Active test steps (as value '1' for YES) will be loaded and executed. Inactive test steps will be displayed in the system colour as inactive texts.

Unit

Unit of the measuring value. Will only be used for display and the saving of measured values. It has no effect on the scaling of the measured values.

Precision

Digits after the decimal point in result. Describes on how much digits after the decimal point of the measured value should be rounded to. Minimum 0, maximum 9 positions after the decimal. It should be corresponding to the accuracy of the measurement.

Upper Limit / Lower Limit

Limits for the measured value. Numerical measured values will be tested against this limits automatically by the system. When the lower limit is greater than the upper limit both fields are marked in red.

MDE

Measuring data acquisition – Data will be saved if this is set to yes value as '1' or if the measured value is outside the limits.

F Repeat (Fail repeats)

Here the test step is repeated F Repeat times before declaring the result as fail and performing any described error state instructions.

F Recover (Fail recovery)

After the failed test step is repeated F Repeat times, the action described in F Recover will be executed:

0	---	No action
1	RCVR step	Execute recovery step, afterwards repeat test step. Repeat as many times as stated in the basic setup
2	FAIL go-to	If a jump was defined for fail case, execute jump. . Repeat as many times as stated in the basic setup

F Continue (Fail continue)

decides (if in spite of all fail measures an error still exists), whether the test should continue in fail case '1' for yes or aborted '0' for no.

Parameter name x Parameter value x   (x ∈ {1..32})
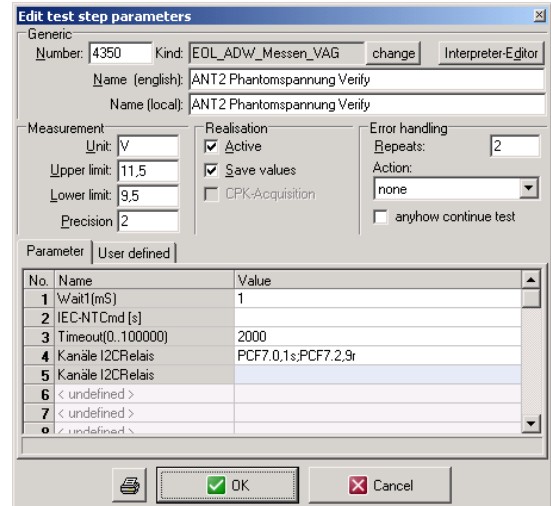
Pair of parameter name and value. The parameter values will be transferred to the executed interpreter step. The name of the parameter normally comes from the interpreter step, and is required only for the transfer of the value. The parameter name helps the user to understand the value on observing the sequence table. The value of the parameter thus has to be within the limits defined in the interpreter steps. A limit value check is however not performed while editing in the parameter sequence table. Hence it is easier and safer to make changes individually in the test steps edit window.

9.7 Edit window test step

The edit window can be initiated with the short cut Ctrl + Enter, context menu or the symbol bar. It provides the option to change the data of the actual selected test step in a comfortable and secure way. The meaning of the edit fields are described by the fixed field nearby in the grid. One can for e.g. choose a fail measures directly from a combo box. The interpreter step can also be chosen directly from a file dialog. Also you have the option to start the interpreter step editor directly for that particular step. On opening the window the program tries to open the interpreter step to read the parameter declarations. If an error occurs here the option to open the test step editor will be shown immediately. The defined parameters are displayed and the inputs are tested against the defined limits. Parameters out of these limits will be displayed with a red background. On pressing the 'OK' button all values are tested against their limits and the found errors are displayed.

The editing of the parameters should be preferred directly in the test step edit window editing, to avoid any possible errors in the test sequence.

The user defined field can be edited by activating the second tab 'User defined'.

9.8 Parameter inspector

Usually due to the big length of the row of a test step in the parameter list, it is inconvenient for editing. For this reason the parameter inspector was introduced. It displays the parameters of the selected test step from parameter list in a column form. The parameter names are displayed near the corresponding values, where only the values are editable along with those in the parameter list.

If instead of a test step a comment is selected, then the parameter inspector is empty.

9.9 Menu tables

9.9.1 Menu tables/ Save test sequence

Saves the test sequence.

On saving the test sequence the undo buffer is emptied and a recovery of the changes is not possible any more.

9.9.2 Menu tables/ Save test step parameter

Saves the test step parameter list.

The list is saved in text format as 'Comma-Separated-Values' short form CSV. For older program versions the list is saved as DBF data. The column width is automatically calculated as per the length of the data. Since the entire length for the Dbase field cannot exceed 2048 characters, the exceeded parameter values can get lost.. Only the beginning part of comments are saved.

The saving in DBF format may not exist in future.

On saving the test sequence the undo buffer is emptied and a recovery of the changes is not possible any more.

9.9.3 Menu tables/ Save all

Saves the test sequence list and also the test step parameter list.

On saving the test sequence the undo buffer is emptied and a recovery of the changes is not possible any more.

9.9.4   Menu tables/ Open type family

Opens a dialog window to open the new type family. The already open type family is shown in grey with an open directory symbol. Since every type family can only be opened once, the previously opened tab is activated again.
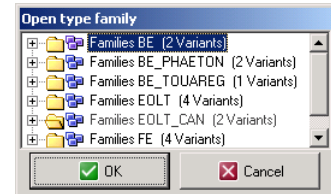
9.9.5   Menu tables/ Close type family

Closes the selected type family. If changes in this type family have to be saved, the user is asked by a dialog to save the test sequence and the parameter list separately.

The editor is automatically closed when the last type family is closed.

9.9.6   Menu tables/ Exit

Closes the editor. All the open type families are closed and then the entire data is saved. Before closing the editor a reset is executed in order to load the new changes made.

9.10   Menu edit

9.10.1   Menu edit/ Undo

It reverses any previous changes made in the sequence list or the parameter list. The undo buffer is extremely vast and is emptied only during saving or changing the parameter structure.

9.10.2   Menu edit/ Redo

It executes back the previously reversed value. Can be executed directly only after an undo if no other changes are made.

9.10.3   Menu edit/ Cut

Moves the actual selection in the clipboard. Functions in a similar manner like copying to clipboard and then deleting it.

9.10.4   Menu edit/ Copy

Copies the actual selection in the clipboard. Depending on the selected area the clipboard can contain the following data:

- Pure text   (shown as text, length in characters)
- A block of test step cells   ('ParCell', as dimension)
- A block of sequence list cells   ('SeqCell', as dimension)
- A block of test step rows   ('ParStep, numbers)
- A block of sequence list rows   ('SeqStep', numbers)

The type of clipboard content is shown on the right end near the area for clip board operations on the tool bar. The display is only normal coloured when the data from the clipboard is copied back to the original area.

If incomplete rows or columns are marked, then only the content of these cells are saved, excluding the comments or filters present.

If at all the comments or filters are to be copied then the complete rows or columns should be marked.

9.10.5   Menu edit/ Insert

Only those values from the clipboard can be copied to those areas which are compatible to the clipboard values. Thus a pure text can be copied only in an activated field editor.

To insert the cell blocks some special functions are present:

The inserted data is only copied in the cells of sequence list or test step list, thus ignoring the comments or filters present.

If a cell is marked (no block), then an equivalent block selected by the clip board is used to save this cell.

If a block is marked (minimum 1x2 or 2x1 cells), then the marked cells are inserted with the data from the clipboard.. If required the block from the clipboard is repeated till the entire marked target area is filled.

9.10.6   Menu edit/ Delete

It deletes the entire marked area. If an entire row is marked then the whole row is deleted. If only cells are marked then the cells are filled with their empty values, for e.g. text fields are filled with empty values, numerical values with 0, logical values with 'no' and for combo-box the first value.

9.10.7   Menu edit/ New test step

Inserts a new test step in the test step list, or in the sequence list before the marked cell.

9.10.8   Menu edit/ New variant filter

Inserts a new variant filter before the marked cell.

9.10.9   Menu edit/ New comment

Inserts a new comment in the test step list, or in the sequence list before the marked cell.

9.10.10   Menu edit/ Edit

Opens the test step editor window for that marked step. Similar to Ctrl-Enter.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 55 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

9.11   Menu search

9.11.1   Menu Search/ Search

Shows the dialog to search a text. In the edit field above the text to be searched for can be given.

As search area the list selected is usually the area currently marked i.e. the sequence list or the test step list.

Only those fields are searched which are selected by the 'Tick' mark. The rest are ignored. The area in the test table has a context menu (with right mouse click). This helps to sort or mark the fields alphabetically. However it has no effects on actual table.

Using this context menu the fields can also be activated or deactivated.

The area to search the whole table, after the selection (when a cell is selected) or the entire marked area (when an entire block is marked) can be selected using the options below.

Also options are present to set the search direction i.e. important for repeating a search. Additionally case sensitive searches can be made and also those cells be searched which contain the whole text.

With 'OK' the search for the given text is started. This search can be repeated after the marked area when required.
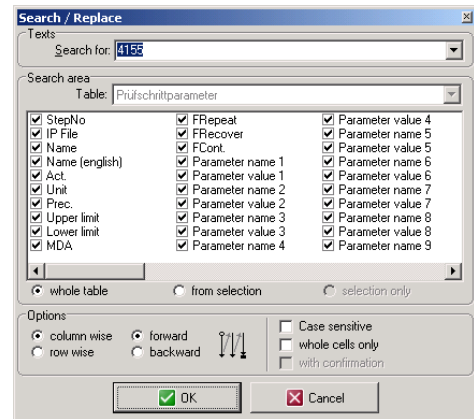
9.11.2   Menu Search/ Replace

Shows the dialog to replace a text. It is quite similar to the search dialog.

In a n extra text field the text to be replaced with the found text can be given.

Addition an option is present whether each replacement should be confirmed by the user.

With 'OK' the replacement for the given text is started. With 'All' the replacement is repeated till the search string is not found anymore in the selected area. If the option of 'With confirmation' is not selected all the replacements are automatically done.

If the option of 'With confirmation' is selected the replacement is done on confirmation with the user. If 'Yes' is pressed the found text is replaced. If 'All' is pressed it act as if 'With confirmation' is not selected and all the replacements are automatically done. If 'No' is pressed the found text is not replaced and if 'Cancel' is pressed the search is interrupted.
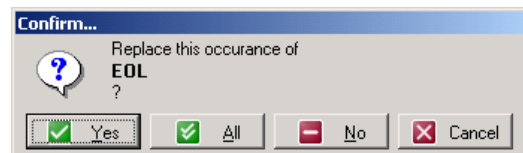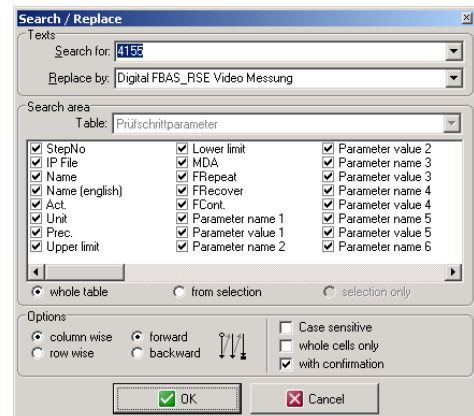
9.11.3   Menu Search/ Repeat

Repeats the previous search or replace action with the previously made settings from the actual position.

MCD Elektronik GmbH	Date: 2018-04-12	Page 56 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

9.12 Menu changes

9.12.1 Menu changes/ Highlight changes

If this option is activated (default) all the previous changes made since opening of this editor window or any previous save action (only those values which are currently changed) are shown with a light yellow background. This helps to easily indicate the changes.

9.12.2 Menu changes/ Undo this change

This menu can only be selected for the field i.e. changed since opening of this editor window or any previous save action. This particular field is then set back to its previous value. This acts like a conditional undo function, and this action can be reset by an undo. This action can be executed using the symbol list or the keyboard shortcut Alt-Ctrl-Backspace action.

9.13 Menu test step parameter

9.13.1 Menu test step parameter/ Edit view

Opens a dialog to edit the column width and sequence of the parameters in the test parameter list.

In the dialog the parameter names and values are shown in an sequential order similar to that as shown from left to right in the test parameter list. The marked field can be moved up or down (left to right in the test parameter list) in the dialog using the arrow buttons below. In the right column the width (monitor pixel) of the column in the test parameter list can be given. This affects only the view of test parameter list and does not affect the actual data. If the width is entered as zero then the default program values are used.
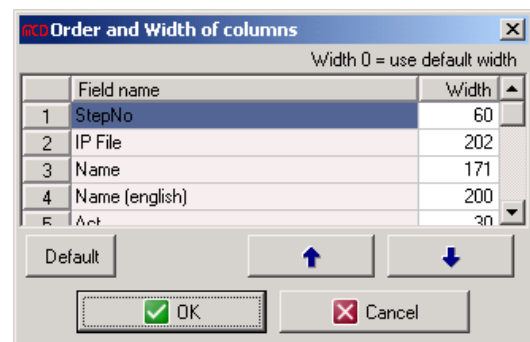
The button 'Default' sets all the column width and sequence of the parameters to the standard values i.e. first the program intern fields then the parameters each with name and value and then finally the user defined fields.

9.13.2 Menu test step parameter/ Edit data structure

Opens a dialog to edit the structure of the user defined fields. The defined fields are shown in the grid window and can be moved up or down using the arrow symbols.

Using the '+ New' symbol on the tool bar a new field can be added.

Using the '- Delete' symbol on the tool bar the highlighted field can be deleted. It is not immediately deleted but a '-' minus symbol is shown on this field in the centre (Action). Thus this field has been marked to be deleted

Using the U⇗X Symbol (Name) selects and marks the field whose name is to be edited. Also when a text is edited in the 'new name' column this field is marked. Invalid names (i.e. non permitted symbols, double characters or names which are internally used).are displayed in red.

To Using the '=' Symbol (No change) the marked areas can be assigned with their old names.

Using the 'ok' button the window can be closed after saving the data. Only if no errors are present the 'ok' button is visible. E.g. for errors are: Repeated field name, an empty field name.

The number of field are not limited.

The contents of the deleted fields are lost in the type list and the new fields created are initially empty. Edited fields receive the new names.

Earlier versions had a data grid with default width to save data in DBF format.. As in the new program the width is flexible having minimum default values this option is not needed any more.

9.13.3   Menu test step parameter/ Short column head titles

If this option field is marked then the columns are shown using short titled names.. Thus the fields having less values can be displayed using smaller views.

9.13.4   Menu test step parameter/ Adjust column width to data

It adjusts the column width to that of the most longer content present in that column. Has the same effect as, when double clicked with the mouse sequentially on each column title.

Warning: Column with long contents are displayed with wide columns. Changes in the column width are not saved in the undo puffer.

9.13.5   Menu test step parameter/ Show parameter inspector

Activates the parameter inspector.

9.14   Menu extras

9.14.1   Menu extras/ Start auto check

Starts the auto check using the test conditions activated under auto check manually. The test can last for a couple of seconds and a window is displayed below during this test. After the test ends the result is displayed at the lower left corner of the editor. When no errors are found 'Auto check : passed' is shown otherwise its 'Auto check : failed' and the found errors are shown in a list. On selecting any one of these errors the respective error is indicated with a red box as hint for the error. If after an auto check the entire rows or columns are copied or deleted then the result of auto check will not be valid anymore. It is advisable to restart the auto check in such a case.
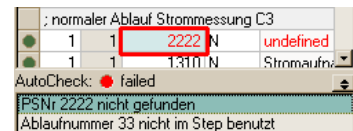
9.14.2   Menu extras/ Define auto check

By using auto check the program searches automatically for errors. The type of errors to be corrected can be defined in this dialog.

Each row represents a test condition. The two checkbox columns when activated show how the auto check should be conducted. The left column for test automatically before saving while the right one is used to start the auto check manually. Auto check is performed automatically while saving if the corresponding menu is activated.

The individual test conditions are:

Check for all steps existing

All the steps used in the sequence list must also be present in the test parameter list. This is independent whether the step could be activated (steps can be deactivated using the variant filter, system or sequence number) in the sequence list. If such a error is present and the sequence step is active it results to an error in sequence while loading.

Find double step numbers on parameter side

Finds the similarly assigned step numbers in the parameter list side. Such an error would not lead to an error condition because the second occurrence of the same number is however ignored. But it is difficult for the user to locate second occurrence in the entire database.

Check for goto targets present

Checks whether all the go to targets used are present in the sequence list. Such a mistake would lead to an error condition when the go to step wants to execute a missing target step. The editor only checks if the required target is present, but doesn't verify whether the go to and the target are activated. Thus a risk still exists for a go to without target.

Find missing interpreter source codes

Checks whether all the interpreter steps present in the column step type exist in the directory. This check is independent whether the steps are activated or whether the steps exist in the sequence list.

+ Check syntactical correctness of IP sources

Along with checking the presence of interpreter steps the syntax checks is also done. The same test algorithm as in interpreter editor is used, which finds the type conflicts in the interpreter step.

+ Check correctness of parameters

Along with checking the presence of interpreter steps the parameter fields are also checked for, whether they are matching to the requirements of that interpreter step.

All variant codes are mentioned at least one time as a variant filter

Requires that each of the type code should be used as a variant filter at least once in the sequence list.

All sequence numbers are mentioned at least one time in a step (no zero only sequences)

Requires that each of the sequence number should be used as a variant filter criteria at least once in the sequence list. This means that the entire sequence list must not only contain sequence number as zero.

Find steps where upper limit < lower limit

For all the steps in the parameter list the upper limit must be greater than lower limit.

9.14.3   Menu extras/ Do auto check before saving

Performs an auto check every time before saving the data. Thus only those the test conditions activated under auto check are executed. If no error is found the data is saved and if an error is found a dialog is opened whether to save the data in spite of these errors.

9.14.4   Menu extras/ Clean up parameters

The 32 pairs of parameters fields (both name and value) for each step are interconnected with the parameter definitions of the interpreter step. Each interpreter step can possess up to 32 parameters from a test step. The values originate from the parameter value field of the test steps. To easily understand the meaning of these parameters the interpreter provides a name, which describes parameter without viewing into the interpreter. This description is found in parameter name fields. If the step type in changed or the parameters of the interpreter step are changed, then the parameter list of the interpreter and parameter values of the test step parameter list do not match any more. This does not affect the program as the parameter name field are ignored. The visibility of the program is however limited.
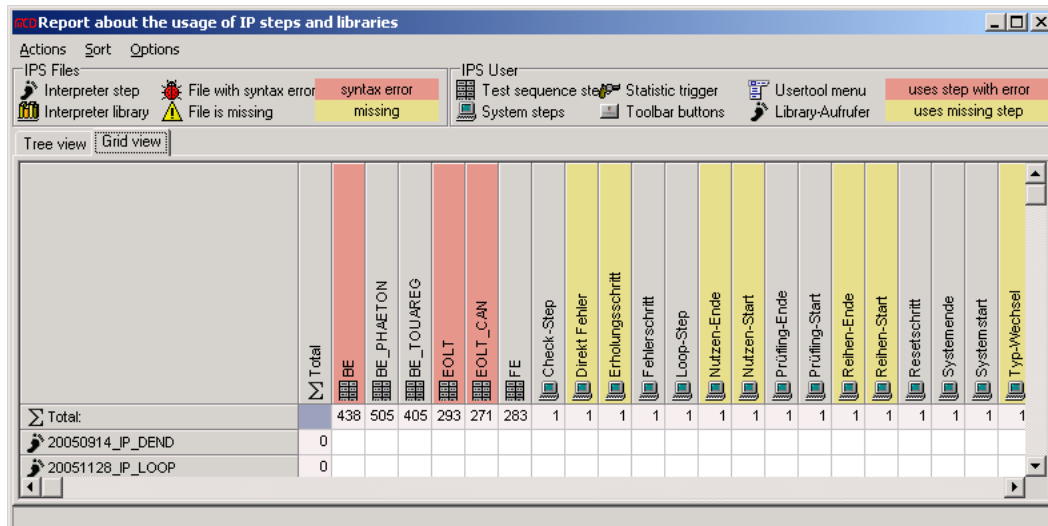
On selecting this option the parameter list of the interpreter and parameter values of the test step parameter list are compared. On selecting it a dialog is opened with first option 'correct names' (always active). This is the actual purpose of the menu and thus must be always active. The second option 'remove unused parameters' must be carefully used. It removes all the values unwanted by the interpreter. This proves to be problematic if at all these empty field are used to save any values or comments for the user. Such data is lost on activating this option.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 60 of 181	**MCD** ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

9.15   Menu analyse

9.15.1   Menu analyse/ Usage of IP steps

Presents an analysis of all the interpreter steps for all the type families.

All existing interpreter steps are checked for any possible syntax errors. Also the interpreter steps are analysed for all the libraries i.e. used. An analysis for all the type families is made to detect the number of times each interpreter step is used. Finally all the possible sources for the usage of interpreter steps are analysed (System steps or actions called by a menu or action button).

The analysed values are displayed in a tabular format. Each row represents an interpreter step or library. Each column represents an user of the interpreter step or library. Thus some of the interpreter step occur in the rows (are called) as well as in the columns (used libraries). It is shown in the following cross fields how often a step (or library) is called by a user (Type family, system step, interpreter step etc).

The rows having coloured background are especially interesting. They show whether the step/ library is absent or whether they have syntax error (see legends).

Using the first menu 'action' the data can be analysed again. This is important when in the third menu 'options' the setting have been changed.

The first menu 'action/ remove unused steps' helps to remove all the unused steps and libraries from IP step directory. These unused steps are then shifted to a new directory called 'Unused' thus cleaning unwanted source codes.

The second menu option 'sort' helps to sort the data using different criteria's.

The settings in the third menu 'option' changes the view. After changing the menu 'option/ Pool XXXX' the data must be scanned again to activate the selected pooling. The menu 'option/ ignore inactive steps' hides the system steps which are deactivated in the basic settings.

MCD Elektronik GmbH	Date: 2018-04-12	Page 61 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

9.15.2   Menu analyse/ Usage of IP steps

Analyses the current type family for all the combinations of type variants, system numbers and sequence numbers. It analyses the total number of steps for each combination.

By selecting the first option 'order' the order of the display can be changed. The options for 'system number' and 'test sequences' allow the selection of the required combination. In the option 'this' any required single number or if many then separated by semicolon (1;2;3;4), for ranges separation using hyphen (1-4). Following combinations are also possible (1;4-6;10).
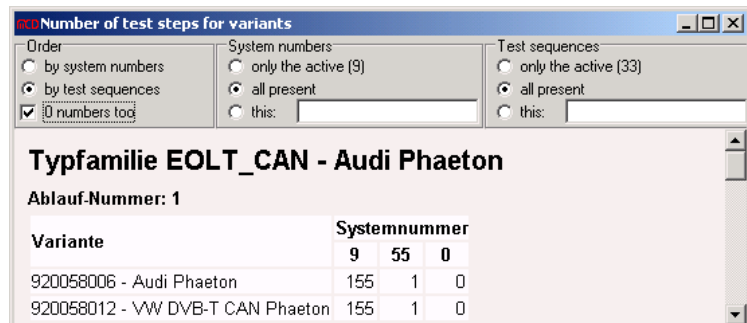
9.16   Editor sequence names

This editor allots names for the used sequence numbers.

A sequence is defined as soon as a number is used in its column in the test sequence list. Thus there is no special procedure to define a sequence number X. The used number X is added to the sequence list however without any name. The name can be edited only using the sequence name editor. The 'hand with pen' symbol near the sequence edit field helps to open the window for the sequence name editor. It shown the names and numbers for all the existing sequence numbers.
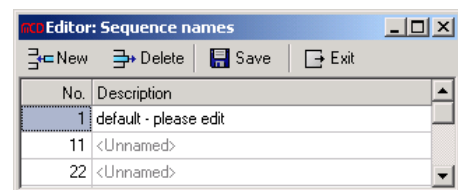
The existing sequence numbers with no names are shown as 'unnamed'. The function of the options in the tool bar can be self explained. It is possible to name the sequence numbers which are marked as unused. If at all a sequence number is deleted unknowingly although it is present in the test sequence list. Then on opening the editor the number appears again without a name. The list is always sorted with ascending order of sequence numbers.

9.17   Parameter page Test step parameter

Description

Used value in field
The test parameter list consist of two description fields for each step. One column is for the domestic country language and the other column is for the English language description for the interpreter step. Using this option the required language to display the test steps can be selected. This language selection influence the test step display as well as the language text used to be saved in the measured value database.

User defined fields
As the number and the names of the user defined fields cannot be tested by the program, it is quite possible to forget such fields in type family. When the interpreter tries to search such user defined field a run time error is caused. By using the following edit field such run time errors can be avoided.

Fields
Provides these fields to the program as user defined fields. Each row can consist one field name only. The database can contain more number of user defined fields.

When field not in database

Controls the case if any one of the field defined above is absent in the database. Selectively the program provides an option to trigger an error or it can do so as the field possesses an empty value.

10   Test sequence

10.1   Sequence of a test

A test can be started only when a) the loop step is active and the step contains the command 'StartTest' or b) the loop step is inactive and the start command is executed using the menu.

At first it is checked whether the loop step sets a type code for the device under test DUT and whether a type change should be made. If a type change is required but it is not possible due to the basic settings then a system error is triggered. If type change is permitted then the respective variant of that type family is loaded. If any problem occurs here a system error is triggered.

Before the actual test start the check step will be executed. With this it can be checked if a test for this DUT in this particular DUT position is currently possible for e.g. problems like mismatches between the test adapter and the DUT type or any superior control can forbid to test this DUT. Within the check step the test can be interrupted without any test result assigned to the DUT. Also no results are entered in the data base or statistics.

After the check step the start steps will be executed according to the DUT position and the basic settings. Thus the start steps will be executed for each DUT start in the test sequence.

The test sequence will be executed for each DUT as long as the end is reached. It is also ended when an error occurs at a test step whose value for 'FContinue' is set to no.

10.2   Parameter page Test Sequence

Test mode

Allows the test to start in the automatic or in step by step mode. It is equivalent to the selection in Menu or the function key F3.

DUT Management

Permit automatic type change

If this option is selected an automatic type change can be executed in the loop step i.e. different to the current type code. If this option is not selected and the loop step tries to make a type change, then a system error is triggered. If the option is inactive the required type code must be manually set.

Barcode exists

This has to be selected when a bar-code or similar code exists which helps the loop step to identify the type of the DUT. When this option is selected and the loop step must select a type code. If it is not selected no type code is selected and the previous one is used.

Combination using these options:

Permit type change	No		Yes	
Barcode exists	No	Yes	No	Yes
Loop-step selects no type code	Starts test with the previous type variant	Error, no type code selected	Starts test with the previous type variant	Error, no type code selected
Loop-step selects previous type code	Starts test with the previous type variant	Starts test with the previous type variant	Starts test with the previous type variant	Starts test with the previous type variant
Loop-step selects another type code	Error, type change not allowed	Error, type change not allowed	Loads the new type variant and starts the test	Loads the new type variant and starts the test

Fail handling

Fail handling by operator
When this option is activated, instead of the normal fail handling options a user dialog is triggered. It provides the user with options to decide when a fail result has occured. The options are repeat test step, execute recovery step (if defined), or execute goto (if defined), continue or stop the test in step by step mode (if permited by the password level).

Maximum execution of recovery step
Decides the number of times the failed test step should repeat the recovery step, when the automatic fail handling is activated.

Maximum execution of fail jump each step
Decides the number of times the failed test step should repeat the fail jump, when the automatic fail handling is activated.

Delays

Delay before test

Delay after test
Decides the delay time before or after the test. It originates from the older program versions and can be realised by the corresponding delay times in start and end steps.

Start Number Management

The start number is an internal counter, which increments the counter independent on the DUT type after each test with one. Thus for each test a distinct serial number can be generated. The maximal possible count is 999999999 (1 million-1).

Reset start number
Decides when the start number can be reset. The options are never (set back to 1 only after overflow) or once a day, week, month, or year.

Maximum start number

Decides the overflow limit for the start number after which the counter springs back to 1.

Save start number
Decides when the preciously used start number can be saved. The first possibility (reset or end of program) uses the hard disk seldom and thus avoids the risk of data loss.. the second possibility (before every test) is safer, but makes the performance poor.

Current (last used) start number
Shows the previous start number.

Last reset start number
Shows the date of the previous reset of start number.

MCD Elektronik GmbH	Date: 2018-04-12	Page 65 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

10.2.1   Sub page Panel Layout

Simultaneous test mode

This option selects how a test of a multi DUT panel is executed.

- Single – each DUT separately (default, standard way of operation)

This is the option selected by default and will select the way of operation that was the only possible way in program version prior to version 1.3.3. The test sequence will be executed one time for each active DUT separately. A test step will return only a result for one single DUT. This option should be chosen for single DUT panels.

- ContactPos – all DUTs of one contact position together

The test sequence will be started once for each contact. All DUTs of the current position will be active together. A test step has to return a step result for each active DUT. Special new interpreter commands have been introduced for this purpose (see `ActiveDUTs`, `SetValueEx` etc.)

- Panel – all DUTs of the whole panel together

As above, only will all DUTs of the panel be active and the test sequence is started only one time to test all the DUTs. A test step has to return a step result for every active (that is every) DUT.

Test panel structure (numbering)

These options allow to adopt the representation of the DUTs on screen to the real layout of the panel. They will only manipulate the representation of the panel but not influence with the program operation itself.

Display of contact and test positions
Select whether the contact positions will be shown horizontally (default) or vertically. The test positions are always perpendicular to the contact positions.

Position of the first DUT
Defines the corner position of DUT number 1.

Scheme
Shows (for a 3x3 panel) the selected layout of the panel.

Sequence with inactive test positions

Also start when all DUTs inactive
Decides whether the column start and column end steps should be executed in spite all the DUTs of that column are inactive. This can be important when in the steps for e.g. the DUTs must be move from their position or a communication with a superior system has to be made.

Also start completely if inactive positions present
Decides whether the row start and row end steps should be executed in spite all the DUTs of that row are inactive. This can be important when in the steps for e.g. when the mechanics has to remain activated for the next step.

Reactivate test positions after test automatically
Reactivate the inactive test positions after test automatically.

10.2.2   Sub page System Steps

System steps – Activity and filenames
The following option always consist of with a checkbox and a data name field. The checkbox decides whether the step is active or inactive. If the checkbox is not selected then the data field has no influence. Using the buttons behind the data field a selection box can be opened. Behind the buttons the default names are given

For certain steps addition field are present, which are described below. For information on system steps see chapter 12.4.

Execute boot step after next successful reset
Forces the (under circumstances again) execution of the boot step after the next successful reset.

Pause between calls (loop step)
Assigns a pause between the execution of 2 loop steps. The time is calculated between the end of the first step to the start of the next step. Large time makes the system lagging and slow reactions to external events. For small times the system also requires small times for e.g. reactions on user defined action in the menu.

Execute fail step also for invalid results
It decides whether to execute the fails step for such tests which have invalid results (e.g. no result) instead of fail.

11   Test window

As soon as a test starts, it will be switched from the normal program window to the test window.

As compared to the normal program window the test window has the following qualities. Its size can be altered but at the end of test the original size and position as that of the program window is taken back. It cannot be also closed by normally by the upper right cross on the task bar. However it is assured that on cancellation of the test the end steps will be executed. This avoids for e.g. a mechanic to remain opened or unlocked.

The big area in the upper left part of the window i.e. the output area shows the tabs which can be used by the interpreter. It can show the history of the executed test steps or an output field which can be used for text outputs of the test steps.

The step info or the debug info and the display of the actual contact position and the active DUT overview are shown in the upper right area. If the test panel has only one contact position, then the contact positions will not be shown.

The middle area contains an info-column, which shows the description of the current test step, the currently measured value and the unit.

In the lower area a logo, an animated bitmap and the button area are present.

MCD Elektronik GmbH	Date: 2018-04-12	Page 68 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

11.1 Output area

The output area can either be controlled by the interpreter (see details above or the interpreter object 'screen') managed site, a history of the executed steps (see left fig.) or an output field for debug purposes(see right fig.).

11.1.1 Test step history

It shows information about the currently executed test steps. The actual executed test step is not included in the list. The display area in its first left column shows the number of DUTs in the panel, the number and description of the test step, the measured value, the unit and the duration of the test step. The text colour depends on the result of the test step and are as follows: black = good, red = fail, olive = invalid. The colours and the number of entries to be saved can be selected in the basic setup.

It is possible that a test step is displayed in red although the test result was pass. This can be because the results are temporary. It is also possible that a fail step is showed in red now but is later on overwritten to pass due to the fail handling steps.
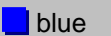
11.1.2 Output field for debug purposes

This site is only visible if the test sequence is executed in the single step mode. Start and end of an interpreter step are in light green, dark green and error messages in red. The messages generated by the interpreter step commands can have their defined colours. With this debug commands for e.g. the temporary results can be shown.

11.2 DUT overview

The DUT overview shows the state of the DUTs in the actual contact position. The first set represents the sequences of the test and the second set the results of the test.

	Test ended		Test running
	Waiting for test		Position inactive

teal	Not tested yet	blue	Just testing
green	Good	red	Fail
white	Without result (SBS)	yellow	Cancelled

11.3 Tool bar

The button area consists of three parts. The two rightmost buttons are the 'help' and 'direct fail' button.

The 'help' button is not yet supported. The 'direct fail' button is activated only when it is selected in the basic setup. On clicking this button the normal test sequence is broken, the direct help step is then executed and the user can define the known error. Thus the DUT has a fail result.

The eight buttons on the left side are freely definable in the basic settings and can be activated using the interpreter.

The order of the 'help' and 'direct fail' and the freely definable buttons can be selected in the basic settings.

The other buttons will be managed by the system.

11.3.1 Button Automatic / SBS (step by step)

These buttons enter different modes depending on the current password level. If the current password level doesn't permit a step by step mode,

then the button 'password' is activated. By clicking on this button the password dialog is opened. After entering a successful password the step by step mode is switched ON. Thus the step by step mode can be entered after starting the test.

As soon as the step by step mode is started the user can switch back to the automatic mode using the same button. The button always shows the current mode on it.

If during the test the step by step mode is activated at least once, then no test result will be generated for that DUT. The shortcut for this button is the F3 key.

11.3.2 Button Abort

With the cancel button the actual test can be stopped. The current DUT gets the test result 'Break'. The end steps (like DUT end, row end and column end) will be executed. Following DUTs in the row or columns will not be tested anymore. The function of this button can also be reached with the End key.

If (in the SBS mode) the Abort button is pressed while the shift key is down, only the active step is stopped. This important when the step enters an endless cycle and must be interrupted. It is similar when the 'debug' button is pressed and then 'cancel' button in the debug dialog is pressed, only that this is more faster.

11.3.3 Buttons Step + and Step –

These buttons are only active in the single step mode. With this the user can go forward or backward in the test sequence. The effect of these buttons sometimes seems to be delayed, because the interpreter has to process the whole step to be able to jump to the next or previous step. The functions of this buttons can also be initiated with the 'Up or Down' keys.

11.3.4 Button Repeat

This button is only activated in single step mode. With this the actual step can be repeated. The effect of this button sometimes seems to be delayed, because the interpreter has to process the whole step to repeat it.

11.3.5 Button Select

This button is only activated in step by step mode. It opens a selection window, where the user can select any step to jump to, manage breakpoints or run till a selected step.

The selection window shows the test steps of the test sequence. The current step will be displayed in light blue , the selected step in light green. With the button Jump the system jumps directly to the selected step. With the button 'Goto' the system switches to automatic mode and executes all steps until the selected step and then switches back to step by step mode.

The breakpoints can also be set and deleted here. In step by step mode the breakpoints will not be noticed, because the system stops after each step in any case. In automatic mode the system switches to single step mode at a test step with a breakpoint.
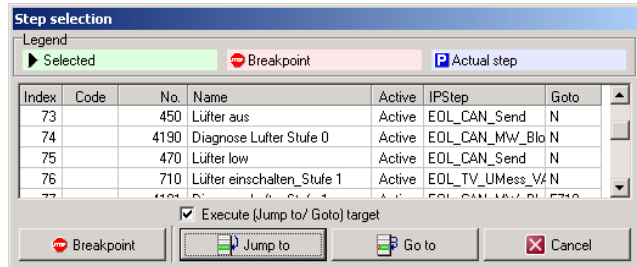
11.3.6   Button IP-edit

This button is only activated in single step mode. It opens the interpreter step editor with the current interpreter step. The effect of this button sometimes seems to be delayed, because the interpreter has to process the whole step first. After exiting the interpreter step editor the current test step will be executed again.
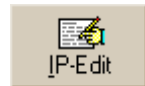
11.3.7   Button Edit Para.

This button is only activated in single step mode. It opens the parameter editor for the current test step. The effect of this button sometimes seems to be delayed, because the interpreter has to process the whole step first. After exiting the parameter editor the actual test step will be executed again.

11.3.8   Button Tools

The button Tools opens a menu, from which the tools can be started which can be used in the test window. For the function of the several tools see the according modules.

11.3.9   Button Debugger

This button is only activated in single step mode when a interpreter step is running. It stops the interpreter after the execution of the current instruction (also long delay instructions will be executed completely). It then allows the execution of the interpreter step by step and allows the user to observe and change the data. For the functioning of the debugger see the respective chapter in the interpreter description.

If no interpreter step is active when pressing the button 'Debugger', then the checkbox 'Start in debugger' will be marked.

11.3.10   Checkbox - start in debugger

If the next starting interpreter step should by stopped by the debugger directly on the first instruction, then checkbox 'Start in debugger' must be marked. Then the next interpreter step will be started with its first instruction in the debugger. The checkbox is then deleted.

11.3.11   Select 'error'

When an eligible password level user is in step by step mode, then the normal error handling is overwritten by the automatic mode. The following options can be selected in the error handling part of each step parameter window:

Handle-   the normal error handling
In SBS mode-   change to SBS mode when error occurs

Ignore   continue test when error occurs, as if no error has occurred

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 72 of 181	**MCD** ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

11.4   Parameter page – Test window

Display

Show information section (right hand side)
If deactivated then, it hides the entire right part of the test window.

Show debug info
Usually the 'debug info' is shown only in the SBS mode. If this option is activated it is also shown in the automatic mode.

Show test step info
Usually the 'step info' is shown only in the SBS mode. If this option is activated it is also shown in the automatic mode.

Show test time
Activates the display of the test time duration (specific customer requirement).

Show DUT info even for only one DUT
Shows the DUT info area below even for only one DUT. It is important when the 'debug info' commands are used for individual tests.

Step history

Show history
Hides the step history tab in the test window when deactivated. Makes the system a little faster.

Depth of history
Decides for the depth of the history. More longer the history more is the time required for maintenance.
Font colour 'no result'
Font colour 'PASS
Font colour 'FAIL
Font colour 'Invalid'
Allows to select the colour for different step results in the step history tab in the test window. (e.g. A customer doesn't want red colour on the display in the production)

Data of actual test step

Show bar with data of actual step
Hides the bar with data of actual step and measured value in the test window when deactivated. Makes the system a little faster.

Clear measured value panel when starting a new step
Prevents from displaying any old measured values, when a new test step is running. Makes the system a little faster and may slightly blink.

Debug show--

Number of lines in history
Decides for the depth of the debug show history. If a zero is entered then the previous values are not available any more.

Elements of lower bar

Show bitmap area(logo and moved bitmap)
Hides the bitmap in the lower left part of the test window when deactivated. Makes the system a little faster. The logo is taken over from the desktop settings.

Show animated bitmap
Hides only the animated bitmap in the lower left part of the test window when deactivated. Makes the system a little faster.

Filename   %SYS%\MCD_FLY36.BMP
Gives the file location of the animated bitmap data. The file must be like a half cut film strip. It should consist of single slides of the same picture. The final picture is then constructed from these slides by the program. Each individual slide must have length and breadth of 36 pixel.

Interval
Provides the display duration for the picture. At the end of the time interval the picture is displayed again.

Show user defined buttons
Hides only the 8 freely defined buttons in the lower left part of the test window when deactivated.

Show help button and direct fail button
Hides the help button and direct fail button in the lower left part of the test window when deactivated.

Stretch mode for buttons
Selects the mode, which influences the buttons according to the width of the test window.

Buttons

Show user defined buttons
Is an image of the button bar from the previous page.

Position
Helps to interchange the position between the freely defined buttons and the help/ direct help buttons.

Preview
Shows how the actual view of the buttons later. By clicking on any of the button it can be selected for edition. Below each button the return value in brackets and the keyboard shortcuts are given. These information are not visible later.

Select button
Activates the selected button and provides its data for edition.

Edit button

Active
Decides the activity of the button bar. Inactive buttons cannot be pressed.

Type
The program supports 2 different types of button on the button bar. In the first type the buttons are allowed to posses a bitmap with a single lined text. While in the second type a multi-lined text along with a background colour is permitted.

Layout
Decides for the position of the graphic on the button. Is activated only if a graphic button is selected above in 'type'.

Message ID
Selects an ID to search the name in the USERMSG.INI (see chapter 7.1.10.4.3). If the value '0' is selected then the USERMSG.INI is skipped and the default text is selected.

Default text
It is the text which is used as default or when no value is found in the USERMSG.INI file.

Bitmap
Gives the file location of the buttons bitmap data. It must be a BMP file. It is useful only if show bitmap is selected above in the option 'type'.

Character set
Selects the font type for the text used for e.g. 'Arial' or 'Times new roman'. If the field is empty a standard font is used.

Character height

Selects the height of the text font in pixel. If the field has '0' a standard font is used.

Return code

They are the codes returned to the interpreter command Screen.LastButton. The codes of the individual buttons should be opposite paired.

Short cut

These are the keyboard shortcuts used to press each button without using the mouse. The codes of the individual buttons should be opposite paired.

Character colour

Colour of the text font to be used.

Use standard background colour

Decides whether to use the standard background colour or the colour selected below.

Background colour

It selects an alternative for the background colour. The colour is used only after mixing it with 50 % the normal background colour. It is useful only if 'multi-lined text' is selected above in the option 'type'.

Reset button

Button: Reset all buttons to grey, inactive, caption = 'reserve' and other default values.

It sets back all the present values to the default values.

12   Interpreter

The interpreter executes the interpreter steps. Interpreter steps are ASCII files, which can be created and edited by the user himself. With the help of the interpreter the whole system can be accessed. Thus it is possible to configure the TestManager system as per the user requirement or any hardware requirement, without changes in the basis program.

12.1   Language

The interpreter language is similarly constructed like the programming languages PASCAL or BASIC. This language consists of procedures, functions, libraries. The data types real, string and vector are also present. For further description about the language see the related documentation.

12.2   Interpreter editor

The interpreter steps are pure ASCII files and thus they can be edited using any available editor or NOTEPAD. Within the TestManager program there exist an own editor which helps to edit the interpreter steps.

The editor can be opened using the menu 'Edit / Edit interpreter steps' or directly from the test sequence editor. The editor can be also be opened from the test window.

If the editor is started out of the menu, then the previously used interpreter step is shown first. On starting the editor directly from the test sequence editor, the actual interpreter step will be loaded automatically.

The display of the interpreter steps is shows a highlighted syntax. Thus the reserved words are displayed with a different colour automatically. The editor can open several files at the same time on its different tabs. But only one of the files is the main file, and its tab is shown with a flag. During syntax check (see details below) only this flagged file will be checked, in spite any other file being in the front tab.

The editor provides an automatic suggestion to complete the interpreter commands. If a part of a interpreter command and point is typed for e.g. 'Curve. ' then all the possible extensions for this command allowed by the interpreter are shown after a short time. If the cursor is already within the second or next part of the command this function can be triggered using the keyboard shortcut 'Ctrl-Space'.
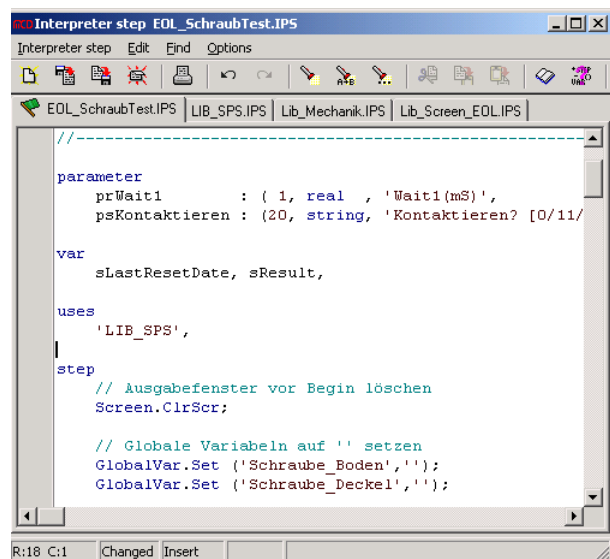
12.2.1   Menu Interpreter step

New

Creates a new page. The page will be filled with the content of the template file.

Load

Shows the open dialog box and allows to load another file.

Save

Saves the actual file with its name. If the file does not have a name, then 'save as' will be executed automatically.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 76 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Save as

Saves the file with a new given filename.

Close

Closes the file. If there are unsaved changes, so a confirmation request will be shown. If the last file is closed, then the entire editor will be closed.

Main file

Describes the file, which will be used during the syntax check. The tab of this file is highlighted by a flag on the tab. The main file can also be chosen with a double click on the tab.

Template file

Chooses the template file, whose content will be copied into new interpreter steps.

List of previously loaded files

This list show the previously edited files. In addition to the filenames the editor also memorises the cursor position.

Exit

Closes all files and exits the editor. If there are unsaved changes, a confirmation window will be shown.

12.2.2   Menu Edit

Undo

Undoes the last change. This undo can be done for several number of times.

Redo

Undoes the action of the undo command.

Cut

Moves the selected text to the clipboard.

Copy

Copies the selected text to the clipboard.

Insert

Inserts the text from the clipboard.

12.2.3   Menu Search

Search

Allows the search for a text. It is possible to search all the open files, entire directories and the sub directories.

Replace

Helps to replace a searched text with the given text.

Continue

Repeats the last search / replace action.

Syntax help

Opens the window for syntax help. The user can then copy commands from the syntax help into the clipboard and paste it into the interpreter step.

MCD Elektronik GmbH	Date: 2018-04-12	Page 77 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Syntax check

Performs a syntax check for the main file. Errors caused due to the improper parameter limits and data types (for e.g. a string instead of a real) cannot be checked. If an error is found the cursor is positioned at it.

Check and Run

Performs a syntax check. If no errors were found, the editor will be closed and the step can be tested immediately.

12.2.4   Menu Options

Font

Allows the selection of the font in the editor window. Only the font type and font size can be selected. Colour and text width (width, cursive) are ignored.

Toolbar

Allows to switch the symbol bar on or off.

Colour coding

Activates the syntax highlighting.

12.2.5   Keyboard shortcuts

Edit	
Del	deletes the character right to the cursor
Backspace	deletes the character left to the cursor
Ctrl + Backspace	deletes the word left to the cursor
Ctrl + Y	deletes the whole line
Ctrl + Shift + Y	deletes till line end
Ctrl + N	inserts a new line
Shift + Ctrl + I	Move front the marked block
Shift + Ctrl + U	Move back the marked block
Cursor moves	
Arrow keys	cursor in arrow direction
Pos1	to the beginning of the line
End	to the end of the line
Ctrl + Page down	to the last visible line
Ctrl + Page up	to the first visible line
Page down	one screen page down
Page up	one screen page up
Ctrl + Pos1	to the beginning of the file
Ctrl + End	to the end of the file
Ctrl + Left	to the beginning of the word left
Ctrl + Right	to the beginning of the word right
Select	
Shift + Arrow keys	enlarge selection
Ctrl + A	select all
Selection mode	
Shift + Ctrl + N	normal selection
Shift + Ctrl + C	selection column by column

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 78 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Shift + Ctrl + L	selection line by line
Jump positions	
Ctrl + 1..9	moves the cursor to the positions 1..9
Shift + Ctrl + 1..9	defines positions 1..9 for the cursor
Undo / Redo	
Alt + Backspace	Undo
Ctrl + Z	
Shift + Alt + Backspace	Redo
Shift + Ctrl + Z	
Clipboard	
Ctrl + Ins	copy selection to clipboard
Ctrl + C	
Shift + Ins	pastes the content of the clipboard
Ctrl + V	
Ctrl + Del	moves selection to clipboard
Ctrl + X	
Miscellaneous	
Ctrl + Up	moves the visible area upwards
Ctrl + Down	moves the visible area downwards
Ins	switches between insert and overwrite mode

12.2.6   Open dialog box

The open dialog box comfortably allows to open a interpreter step in the editor.

On the right side an overview of the selected step on the left is displayed.

Using the following tabs on left the following options to select an interpreter step are available: the file system, history of previous steps (the history tab posses a context menu for edition), the system steps and the sequence steps.
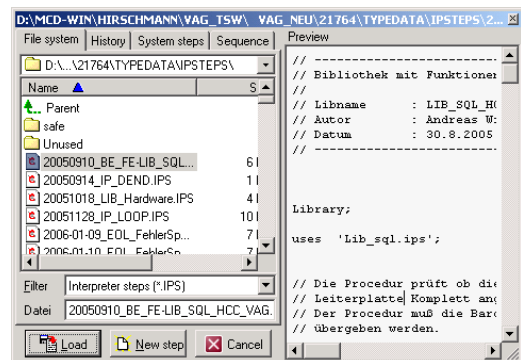
12.2.7   Interpreter help

The help for the interpreter can be activated using 'F1' or 'Ctr+F1'. On using 'F1' the previously used help page will be displayed. On using 'Ctr+F1' the interpreter editor tries to identify the command at which the cursor is present and tries to display an interpreter help for it.

12.3   Interpreter test steps

During the test sequence interpreter steps are called and executed from the sequence table. The test sequence refers to the name of the interpreter step i.e. to be executed. An interpreter step can be used from different test steps. The test steps can transfer parameters to the interpreter step.

12.4   System steps

System steps are special interpreter steps, which are not called out of the test sequence, but at several system actions. The interpreter step to be used by the system action is defined in the basic settings (test sequence).

The system steps can't get parameters and are not type dependant.

The following system steps are defined:

Program start – Boot step

Will be executed once the program is loaded. It is executed after the internal reset and before the reset interpreter step. All modules are already initialised and can be accessed.

Reset – Reset step

Will be executed during each reset after the modules have performed their reset. Is also executed directly after the boot step at system start.

Program end – Leave step

Will be executed directly before the program is closed.

Wait state – Loop step

If the system is in wait state this step will be executed cyclic (for e.g. wait for a sensor to start the test automatically). The repetition duration of the loop step is defined in the basic settings. The loop step can provide the system information regarding the type of DUT for e.g. the type code of the new DUT.

The loop step must always be so programmed that it is executed in a short time period. One should never wait for an event to occur or an user action in the loop step. If at all it is necessary to wait for an event in loop step, then it should be programmed as a state machine. Thus if an event has to be waited for a flag can be set. Then when the flag is set the event is waited for otherwise the loop step is executed. If during the next loop step execution the event occurs then the flag is set back and the reaction is implemented. The wait period influences the cyclic repetition of the loop step and not due to any cycle within the loop step.

Background for this is that the program cannot enter any menu when a system step is active. Thus the user for e.g. can make changes in the basic settings only then when the loop step has been completely executed.

Check step

The check step will be executed when a test is started. In the check step for e.g. you can check if the right adapter for this DUT is connected. Or it can be waited for the superior system to enable the DUT. Inside the check step the test can be aborted without making entries in the measured value tables. When the test is not aborted, the actual test sequence starts and entries in the measured value tables for the DUT(s) will be done at any event. Within the check step the test can be interrupted without any test result assigned to the DUT. Also no results are entered in the data base or statistics. If the test is further continued with the actual test sequence then the results is entered in the data base.

Test start step

Will be executed at the beginning of each contact column.

Row start step

Will be executed at the beginning of each contact row.

DUT start step

Will be executed with the start of each DUT. Afterwards the test sequence follows.

DUT end step

Will be executed after the test sequence for each DUT, independent of the DUT result.

Row end step

Will be executed for each contact row, after the last DUT in this row has been tested.

Test end step

Will be executed for each contact column, after the last DUT in this column has been tested.

Fail step

Will be executed, if a DUT after all fail handling actions has the result 'fail'. In the basic settings the fail step can also be activated for the result 'invalid'.

Recovery step

For the steps whose fail handling is set in such a manner that, the recovery step should be executed if a measured value is out of the limits. Thus for e.g. a communication link can be established again, and failed step can then be repeated.

Direct fail step Dfailstep

This step provides the user an option to interrupt the current test sequence at anytime. The user can thus inform the system with a known error for e.g. on comparing the DUTs the user observes a scratch on the cover. Thus he can interrupt the current test sequence and thus inform the system with a known error code in the step.

Type change

Should be done before starting the test. The type change is only possible when it is activated in the basic settings and triggered in the loop step.

The system interpreter steps can be switched individually, if they are not needed by the system.

For system performing single tests (no multiple DUTs), it is irrelevant whether any of the 3 start or end steps are executed. Multiple steps can also be used.

12.5   Interpreter-Debugger

An integrated debugger for the interpreter can be started from the test window or the interpreter centre (see below). It allows a step by step or block by block execution of the step. It also permits the changes of the variables and parameters in the step.
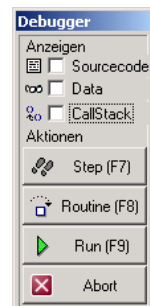
12.5.1   Debugger window

When the debugger window is visible, the debugger is active and the execution of the interpreter step is stopped.

With the buttons 'step', 'routine' and 'run the execution of the interpreter step can be continued further.

'Step'	It executes only a single interpreter instruction. Also the self defined procedures or functions occurring in the path are executed by the debugger. These subroutines are also executed step by step. Shortcut for this is 'F7' key.
'Routine'	executes a single interpreter instruction like a 'Step', but also the self defined procedures or functions are executed. These calls will be treated as one instruction and the complete subroutine is executed. Shortcut for this is 'F8' key.
'Run	switches the debugger back to the normal interpreter mode. Shortcut for this is 'F9' key.

The check boxes 'Source code' 'Data' and 'Call stack' switches the corresponding debugger displays on or off.

The debugger is automatically activated if the interpreter comes to a breakpoint during source code processing. Breakpoint will be managed in the debug window 'Source code'. Breakpoints are automatically deleted if another test step is chosen from the test sequence.

12.5.2   Debugger Source code window

The source code of the current interpreter step and the used libraries are shown in the window. The line with the blue background shows the instruction which will be executed next. Breakpoints will be shown with red background. The content of the window can be scrolled to see the whole source code.

If in the call stack window the step name is clicked twice then source position is shown with a yellow background. Thus the sequential execution of instructions can be analysed later (thus the position of where this function has been called can be found).

Breakpoints can be set and deleted with a double click with the left mouse button. If the desired line is previously selected, then the breakpoints can be set or deleted with the shortcut Ctrl+F8.

If the mouse cursor is positioned over a variable or a parameter, then its type and value is shown in the status bar of the window. With a double right click with the mouse the variable or parameter can be observed in the debugger watch list.

12.5.3 Debugger Watch list

The watch list shows the content of variables and parameters. The variables or parameters can be added either with a double right click in the source code window or with the first 3 buttons. The first button adds all variables, the second one all the parameters and the third combines both. With the last button you can delete rows from the watch list.

The rows with red font colour indicate that the value of the variable has changed since the last display.

The symbols at the beginning of the row shows the kind and type of the data:

Kind :	**V**	Variable	**P**	Parameter	**A**	UMS-Address
Type :	**R**	Real	**S**	String	**V**	Vector

With a double click on a variable the value of this variable can be changed. This also applicable to the parameters and is the only possibility to change parameters after loading, as the step itself is not able to do this.

If a real or string value is to be changed then the upper left window will be displayed and the new value can be entered. With vector values the upper right window will be opened and the values of the vector can be changed. The number of elements inside the vector can be changed with the 'Edit' button.

12.5.4 Debugger Call stack window

The call stack window shows the source on an called function or procedure. At the bottom the step name is shown and then the respective libraries along with their variables are shown. If in this list the step name is clicked twice then source position is shown with a yellow background.

12.6 Parameters of the interpreter

Basic setup page interpreter

Syntax options

Allow text after 'End'
If this option is activated comments are allowed after the 'End' of an interpreter step.

Allow unused variables
If this option is activated unused variables are allowed in an interpreter step.

Allow unused parameters
If this option is activated unused parameters are allowed in an interpreter step.

Source buffer

Keep interpreter source code files in main memory

Normally after execution of an interpreter step, the step and its libraries are deleted from the memory and reloaded again when required. If this option is activated the steps are saved in the memory.

Maximum number of files in memory

It gives the maximum number of steps and libraries to be saved in the memory. For '0' it is set to unlimited. Only in emergency the limits should be set. For insufficient memory it is better to completely deactivate the source buffer.

Maximum total size of files in memory

It gives the maximum space for the steps and libraries to be saved in the memory. The allotted space should be sufficient to execute a complete test sequence. For insufficient memory it is better to completely deactivate the source buffer.

If either the limits for number of files or total size are reached, then the most oldest unused files will be lost first. This can cause a continuous deleting and loading of data during a test. Hence the limits should be so big that the entire data for a test sequence fits in the memory. Otherwise it is better to completely deactivate the source buffer.

The buffer is automatically emptied when a reset is executed, or when an interpreter step is saved. If an interpreter step is edited outside the program, then a reset should be executed. Thus the data will be freshly loaded by the program, and the old data is replaced with new in the buffer.

Debugger

Write Debug.Show information to IPDebug.log file

If this option is activated the test outputs of Debug.Show command will also be saved in the IPDebug.log file additionally. It additionally functions like the command Debug.Log. Colours are however not allowed.
The basic setup page for interpreter has the following parameters:

12.6.1   IP Alias

Interpreter object: Alias

It is used when the same interpreter step performs different functions on two different systems. For e.g. consider a manual and a fully automatic testing unit which perform the same test. Thus individual system steps should be written for both the systems. An alternative is to write these steps different libraries called by the same system steps. In this case only the respective library names should be changed in the step. To avoid this confusion the alias is present. The user should just mention the library name i.e. to be replaced.
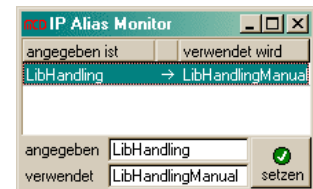
For e.g. consider a system step which calls the library 'LibHandling'. Thus on the manual testing system an alias is created with the name 'LibHandlingManual' to replace the 'LibHandling'. Similarly on the automatic testing system an alias is created with the name 'LibHandlingAutomatic'. Thus both system can have the system steps with only different handling steps.

Thus alias provide the library name to replace the dummy library name in the step. Along with the basic settings the aliases can be assigned using interpreter commands and IP alias monitor.

Thus the IP alias monitor allows to observe and edit their names. For new entries enter the new file names and press the button 'set'. To edit the name change it in the second edit field and press 'set'. The procedure for deleting an alias is similar to editing but only the second field should be left empty.

Enter aliases like XXX=YYY, one alias per line
All the aliases used since the beginning of the system are entered here. One alias per row. The library name before the equal to sign will be replaced by the library name after the equal to sign.

Show aliases in debug window
If selected shows the replacing aliases explicitly in the debug window.

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

12.6.2   COM/ ActiveX

Interpreter object: COM

The Interpreter object: COM provides access to the COM (Common Object Model) Server. This COM server must be installed on the system.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 84 of 181	**mCD**® ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

COM/ ActiveX Enabled
Activates the entire access to the COM server

Log communication to log file
It allows to record the entire communication with the COM server

12.6.3   Global variable

Interpreter object: Global variable

Global variables initialised in the interpreter steps can be created and deleted during the test. Additionally everlasting global variables are present which are also known as persistent global variables. They are created in the basic setup and posses these values throughout the end of the program.

Persistent global variables

Each row in the grid represents a persistent global variable. Within the grid the name, value and the comment can be edited.

The name is used to access these variables using the interpreter. It must be distinct. Persistent global variables can be transferred to Measured value-DLLs and the report generators. Since the enumerated should have maximum 19 characters, the persistent global variables should not have more than 19 characters to be used as enumerated data.

The value should correspond to its variable. The type is indicated by the first alphabet in the column. By double clicking on any of the rows a dialog to edit values is triggered (see chapter 7.1.7.1).

The comments can be freely selected as they do not influence the program.

New variable
A new persistent global variable is created. The same dialog like global variable monitor is used (see chapter 7.1.7.1).

Remove variable
The persistent global variable is deleted. It is the only possible way to delete a persistent global variable.

12.6.4   Power supply

Interpreter object: Power supply

Enable access to the calibration data
On opening the basic settings an access to the calibration data is disabled to avoid any changes caused carelessly. By activating this option access to the calibration data is enabled.

Uofs
Voltage supplied by power supply when it controlled with 0Volts.

Umax
Voltage supplied by power supply when it controlled with 10Volts.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 85 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13   Module

Modules are program components which are not necessary for running the program but they provide additional features. For e.g. the CAN ports, these ports are only needed if the DUT or the test environment has a CAN interface.

In future it is planned to construct these modules like 'Plug-Ins' which allow to install new modules without a change of the basic program.

Modules usually contain the following components:

Basic functionality –explains what the module does, reason and aim

Parameters – configuration of the basic setup

Window – display monitor used for the access of the module

Interpreter objects – expanding the interpreter syntax

Not every module consists of all components.

The interpreter objects is described in the special interpreter language documentation (online help).

13.1   Change log

It helps to identify and track the changes made in the test sequence. If the change log is activated, then while closing the test sequence editor a dialog window is opened. The user can enter the reason for editing here. The changed area, date time and computer name are entered by the program. The user name is taken from the windows registration name, but however can be edited. The description can be long enough and multiple lined.
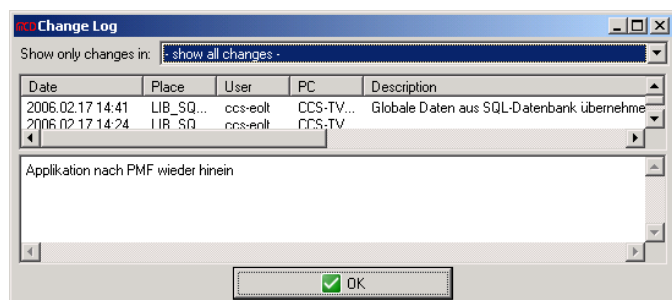
Using the double arrow button in the left corner below any previously made description can be used again. If the checkbox nearby is activated then the same change descriptions are automatically entered till the next type change.

The changed data is saved in different data according to the places of changes. Changes in the type list are saved in the type data root directory (%TDR%). Changes in the type family are saved in the each of the respective type family. Changes in the interpreter steps are saved in the interpreter step directory. The name of the file is always 'Changes.log'.
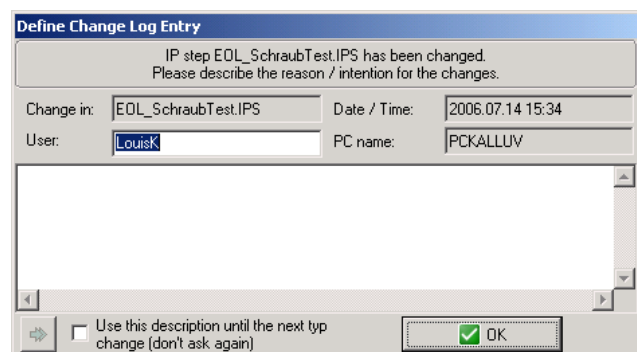
13.1.1   Parameters

The change log can be activated in the basic setup.

13.1.2   Overview

If the change log is activated, then in the menu info a tracking window for the changes made can be opened (Info/ Change log).

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 86 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

The tracking window collects the changes made for displaying it to the user.

In the window a filter for sorting the data can be set in the combo box at the top. All the changes done are shown in the list, which can also be sorted by double clicking on the column description.

For each selected change the entire description is shown below.

Using the button 'manual entry' the user can enter a new change information anytime.

13.2 AngeloRTV Frame grabber

AngeloRTV Frame grabber can read and edit up to 16 video channels. In order to access this card the respective driver and the card must be installed. If the module is deactivated then no driver or card is required.

Exact card description: ADLINK RTV Series.

Interpreter object: AngeloRTV

13.2.1 Parameters

General

AngeloRTV enabled
Activates the complete access to the frame grabber card. For systems where the card driver is absent the card must be deactivated.

Item in tools menu

Show in
Selects whether and where to display the monitor in the tool menu.

13.2.2 AngeloRTV monitor

The AngeloRTV frame grabber monitor is currently in beta stage and will be explained here in detail later.
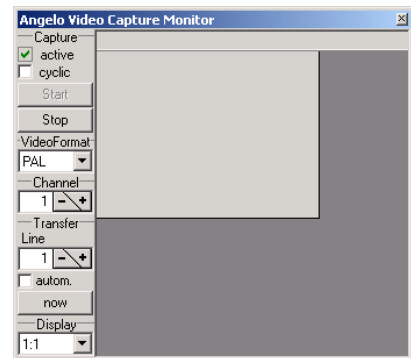
13.3 Registration form

The registration form is a simple window like form consisting of several input fields in which the user can enter information. The user can edit this information during the program. These information can be read and processed with interpreter instructions. For e.g. the user can enter his user ID or tester ID and this number can be printed on a test protocol. It can also be saved on an EEPROM inside the DUT.

Interpreter object: `RegForm`

13.3.1 Parameters

The different fields of the registration form are defined on this page of the basic settings. Each row in the grid preview shows one input field of the form. The order in the grid describes the order in the form.

Entries

Buttons: New edit field, new combo box, new checkbox
Each button creates a new field of its type in the registration form.

Virtual keyboard

Layout file
This file is responsible for the function and number of keys defined on the keyboard.

Preview (approx. half size)
It shows the preview of the final registration form but almost in half size.

For every new field in the form a new page is generated in the tree structure below. On this page the individual characters of the field can be defined. These pages consist of common parameter elements to all 3 field types in the upper half. While the lower half of the page consists of field specific parameters.

Common elements

Common values

IP Name
The interpreter uses this name to access this field. Although all the characters are allowed, only valid interpreter symbols should be used.

Message ID
It defines the ID which will be used to search in the USERMSG.INI for describing the button (see chapter 7.1.10.4.3). A 0 can be entered, to ignore the USERMSG.INI and replace it by the default text.

Default-Message
This text will be used if the Message ID is not found in the USERMSG.INI.

Password level
Defines the minimum password level required to edit the field.

Program reset after change
If this field is activated, any changes made in the field is followed by an system reset on closing the window.

Default value

Default-value
Is the value which is displayed as default when the window is opened.

Field specific elements

Edit field parameter

Maximum length
Decides the maximum length and width of the input in the field. If extra characters are entered they are ignored. A zero means unlimited characters can be entered.

Indent to position
Normally the field name (message ID or default text) is shown to the left and the edit field to the right. In order to adjust all the fields to begin from a common starting line in the left this field is present. The edit field is then aligned to the left according to the position selected by the user even though the text on the right is shorter. The value entered is scaled.

Allowed characters
Shows of the allowed input characters. All other characters are ignored. All the characters can be entered for e.g. '123456789' or in short with '..' for e.g.: '0..9' = all digits, '0..9'. if the field is empty all characters are allowed.

Mandatory
When this is activated the form cannot be closed with 'OK' on leaving the edit field empty.

Convert to UPPER CASE
Converts the lower case automatically to upper case.

Hide characters (******)
For every character entered a '*' is shown instead of that character.

Locked edit – needs to be unlocked
Shows a locked field, having a lock symbol on the right. The field must be first unlocked to edit by double clicking the lock. These can protect against any accidental changes made.

MCD Elektronik GmbH	Date: 2018-04-12	Page 88 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Check box parameter

As default values for a check box '0' = not marked, all others stand for = marked.

Indent to position

This option is only activated if the option below 'Check box after text' is selected.

Normally the field name (message ID or default text) is shown to the left and the edit field to the right. In order to adjust all the fields to begin from a common starting line in the left this field is present. The edit field is then aligned to the left according to the position selected by the user even though the text on the right is shorter. The value entered is scaled.

Check box after text

Normally the box is shown left of the text. But in order to align all the edit fields this option is present. It shows the check box after text.

Combo box parameter

Indent to position
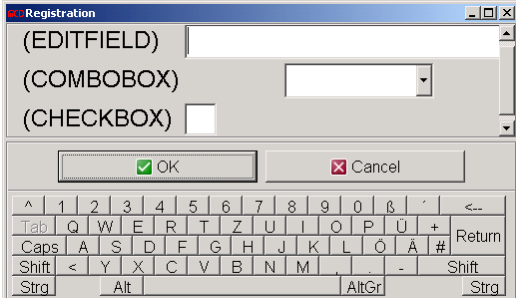
Normally the field name (message ID or default text) is shown to the left and the edit field to the right. In order to adjust all the fields to begin from a common starting line in the left this field is present. The edit field is then aligned to the left according to the position selected by the user even though the text on the right is shorter. The value entered is scaled.

Items

The options to be selected in the combo box should be listed here. Each item on each row.

13.3.2   Registration form monitor

The registration form can be accessed with the shortcut F5 key, the menu or the respective button in the symbol bar. It memorises the last screen position and the size which is defined by the number and kind of the entries. If any compulsory field is present then it will be displayed with a red background if it is empty. Closing the form with Ok is not possible in this case. If the form is exited with Cancel, then all the changes made are reversed. On closing the window with Ok, all changes are saved and can be called using the interpreter. If a field connected with a reset is edited then the OK button triggers a system reset after closing the window.

The changes are also saved permanently, so that the new inputs are available on the next program start. The value of the input field can also be changed using the interpreter instructions.

If 'Must' field is empty and the user tries to start a test then the registration form is displayed automatically. If the input is cancelled, the system goes to fail state.

13.4   CAN Ports (via Vector CANCardX)

The CAN ports allow the communication with DUT or test environment using the CAN bus. As hardware for CAN the following cards can be used:

- CANCardX/ CANCardXL belonging to the company Vector is available in a PCMCIA format. Additional to the card a suitable physical driver and adapter cable (CABs) are required. For the operation of the PCMCIA carrier under Windows NT a special driver called as the Card wizard is needed.

    Driver library: Vector

- CAN-AC2-PCI card belongs to the company Softing

    For this card the suitable driver must be installed (Piggy banks).

    Driver library: Vector

MCD Elektronik GmbH	Date: 2018-04-12	Page 89 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

- PEAK CAN card belongs to the company Peak

  Optionally an external converter for the driver layers can be inserted.

  Driver library: Peak evaluation or Peak light

The cards must be correctly installed.

To access these cards different driver library are required depending on the card manufacturer.

- Vector library

  It is used for the CANCardX(XL) and CAN-AC2. Allows to access the virtual channels for each of the installed card. Additionally the vector driver possess 2 virtual drivers and an access to the CANPara dongle for the parallel port. The access to CANPara is not yet tested.

- Peak evaluation library

  In spite of the name this library is not a test product. It should be additionally purchased from Peak. It allows a multiple access to Peak can cards.

- Peak light library

  It allows access to a single Peak can card (maximum 2 CAN channels). (This library is not yet implemented).

It is possible to use both Vector and Peak CAN ports simultaneously and thus mix the 2 ports. This is not yet tested.

The driver libraries and their corresponding DLLs are only accessed when the CAN module and the respective libraries are active. On deactivating the CAN module in the program there is no requirement of a driver installation or any DLL.

The program can manage multiple number of CAN parts. In order have a clear overview and performance, the unused ports should be removed.

Interpreter object: CAN

13.4.1   Parameters

General

CAN ports enabled
The entire access to CAN might be completely switched on or off on this page. All the CAN libraries are affected.

Vector library (CANCardX, CAN-AC2 etc.) enabled
Activates the access to the respective library.

Peak PCAN evaluation library enabled
Activates the access to the Peak PCAN evaluation library.

Peak light library enabled
Activates the access to the Peak light library. As this library not yet tested this function cannot be selected.

Overview (actual setup)
Shows the actual configuration. If CAN driver were earlier activated then the connection between the physical channels and the defined canals are shown. If CAN is inactive then only the channels defined by the program are shown.

Item in tools menu

Show in
Selects whether or where the monitor should be shown in the tools menu.

MCD Elektronik GmbH	Date: 2018-04-12	Page 90 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Button: Create a new port

Creates a new CAN port (naturally only a logical one within the program). This is further added by an extra page below where it can be set up.

For each possible CAN channel a seperate page exists:

Can port

Name

Defines the name under which the interpreter can access the channel. Although the name can consist of any characters, only valid interpreter characters should be used. The name has to be unique.

Active

If the port is deactivated the entire access to the port is blocked.

Driver

Selects the driver library to connect the channel. The selection 'CAN channel' below depends on this selected driver option.

CAN Channel

Connects the CAN port with the physically available CAN channel. The number of options depends on the selected driver option above.
The Vector library provides 32 channels. If this library is selected then the 32 channels are shown along with their corresponding existing channels.
The Peak library provides 16 channels. If this library is selected then the 16 channels are shown along with their corresponding existing channels.

Open automatically

If activated then the CAN port is opened automatically after a reset. Otherwise the port has to be opened using an interpreter command before access.

Data format

In CAN protocol a bit duration is not the smallest dividable time unit. Mostly the unit consists of multiple segments and this in turn consists of multiple basic units (time quanta). Thus the baud rate is calculated as the reciprocal on the number of basic time units per bit multiplied by the duration of one basic time unit. The duration of a basic time units found by the pre division of a quartz clock. The program is user friendly and so allows the user to enter the required baud rate and the length of the bit segment. The pre divider is not calculated. It is not possible to calculate all the combinations of baud rate and segment lengths.

Baud rate

Sets the baud rate, for CAN communication in symbols.

SJW

Synchronisation Jump Width, describes how the time unit should be synchronised to the data on clock deviation.

Time Seg 1

Length of the first part of the basic time units.

Time Seg 2

Length of the second part of the basic time units.

Number of samples

Number of samples required by the CAN controller in order to calculate the value of the bits.
When the combination of baud rate and segment length do not match an error message is triggered. If the combination is valid then a small graphic representing the bit structure is shown.

Acceptance Filter

The acceptance filter describes which CAN frames should be received at all.

Mask + Code
It forms the input filter for the small identifier (11 bit). A frame is only then permitted when all the 1 bits of the masked value and that of the received frame are same. Positions of 0 bit in the mask value are ignored. For mask=0 all frames are permitted and for mask=7FF only frames with similar identifiers are permitted.

Mask 29 bit

Code 29 bit
It forms the input filter for the long identifier (29 bit). It has similar function like above.

Receive buffer size

Receive buffer size

Frames
Describes limit for CAN frames in the receive buffer. When the buffer overflows, then the incoming frames are lost.

Buttons: Copy port
Creates a new port using the similar settings of the current port. The name is entirely changed.

Buttons: Delete port
Deletes the current port.

13.4.2   CAN-Monitor

The CAN monitor can be called using the menu tools in the main window or with the button tools in the test window.

The monitor shows a tab for each CAN channel. The following details can be observed from top to bottom:

- the state of the current parameters of the channel-this area depends on the driver library
- filters for events of the channel
- event area for this channel-events, messages and error messages along with a time stamp
- symbol for cyclic messages, the receive buffer status and a button to open/ close the channels
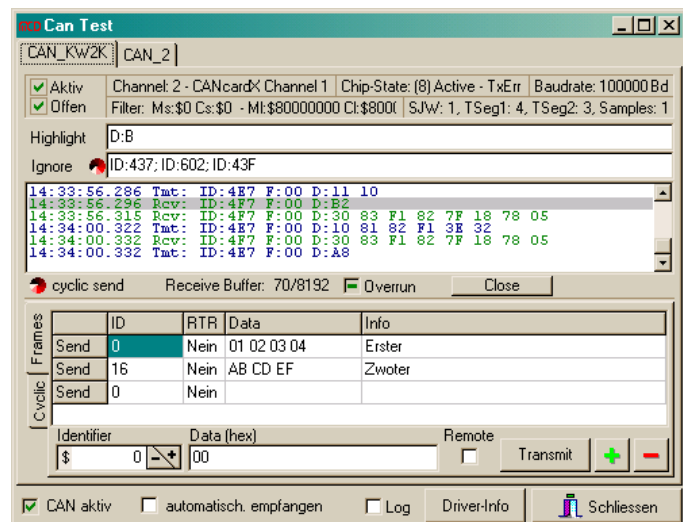
The filters influence whether and how the events should be displayed. The first filter 'Highlight' allows to distinguish the required text using a highlighted background (see the second row in the message area). The second filter 'Ignore' ignores the selected messages completely. Instead of showing these messages the cursor is moved further to the next message. With the help of this filter uninterested and cyclic messages can be completely ignored. The filter consists of a number of texts which are separated by a semicolon. Consider the e.g. filter 'ignore' : 'ID:437' 'ID:602 'ID:43F' where all these IDs will be ignored and all the messages with the first byte 'B' will be highlighted.

The lower part of the site consists of two sub pages, one for normal messages which are stored here in the grid and can be sent with a click on 'Send'. New messages can be defined in the fields Identifier, Data and Remote. They can then be sent direct with the button 'transmit' or be stored in the grid with the plus button. The minus button deletes the actual row from the grid. The grid data are stored in the 'ToolData' directory. The second sub page shows the cyclic sent messages. These can be defined in the monitor or created by the interpreter. As soon as a cyclic message is sent, the symbol for this messages moves one step ahead.

In the lower left corner it is shown whether the entire CAN module is activated. If the CAN module is deactivated or if any single channel is not activated, then a hint is displayed in the event area.

Sent messages will be executed immediately, independent of whether it is sent from the interpreter or from the monitor (exceptions are cyclic messages). All the received frames are displayed in the event area only after they are read from the receive buffer. This can be executed by the interpreter or from the monitor itself. To execute this checkbox below 'Automatic receive' must be activated. It should be noted that the monitor deletes the frames from the receive buffer for displaying it and thus they can't be received from the interpreter anymore. For the normal operation the 'automatic receive' must be switched off.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.

When the 'Log' checkbox is selected, all the message in the event area is saved in the log file '%LOG%\CAN.LOG'. The 'ignore' filter influences this log file also.

With the button 'Driver-Info' the status of the CAN drivers can be requested. This shown information consists of the internal data of the driver.

13.5   DAS4020 high speed AD card

It is a high speed data acquisition card which acquires fast analog signals.

Because of errors in the card driver this card will not be used in further projects.. The normal calling of the driver can cause the system to crash. On crashing the system is rebooted automatically without properly shutting down. Thus in worst case the hard disk may be damaged causing loss of data. In the program precautions are taken to react over the critical conditions (e.g. monitored timing of function calls).

The card must be properly installed with all its drivers. Memory should be allotted to the card in the control panel.

13.5.1   Parameters

Common

DAS4020 enabled
Activates the DAS4020 completely. For systems without installed driver the module must be deactivated.

External drivers for analog inputs

Channel 0

Channel 1

Channel 2

Channel 3
The conversion factor for each channel can be entered. The total input value will be multiplied with the factor before passing the value to the program. Thus external divider can be calculated.
The DAS4020 card is not yet included in the calibration scheme with gain and offset values.

Items in tool monitor

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

13.5.2   DAS4020-Monitor

The monitor allows to access the card and to show the measured values. The status of the card is show on the info tab. The second tab (Channel view) shows the previously measured values of the 4 channels and the current value set.

If the card is ready-to-operate, then by activating 'cyclic read' the individual channels can be measured in a cyclic manner. The channels having a green tick mark on their side will only be read. By double clicking this green symbol the cyclic measurement can be activated. By clicking the arrow symbol to the right the range of the channel can be changed.
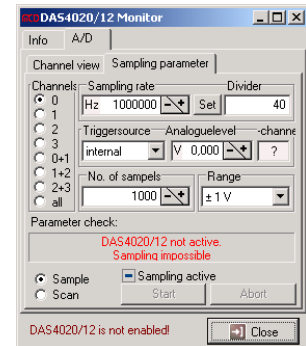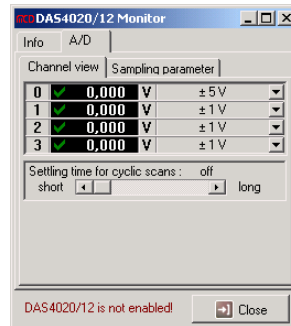
The last tab (sampling parameter) allows to control the changes of the sampling parameter. A sample is possible only when no error is present in the field 'parameter check'.

13.6   Auto run

An auto run can serve the following purposes of, testing a test sequence, stability of the test sequence, gathering measured values for the testing instrument capability.

The program is started new as many times as the value entered. A start is equivalent to pressing the start button in the menu or setting a MenuStart flag.

Interpreter object: CAN

13.6.1   Parameter page

Auto run

Number of automatic starts
Number of starts required can be entered here. After these starts, the auto run is closed automatically.

Delay between two starts
Sets the time delay between the end of a test and the start of the next test.

13.6.2   Auto run overview

The auto run can be started using the menu 'Start/ Auto run/ Start auto run'. If the auto run is currently executed then at the top left hand corner a small control field is shown.

Here the number of test i.e. to be still conducts and the delay between two starts is shown. The time is display in minutes and seconds, and parts of a second is shown as a progress bar.

The three symbols below show the current status of the auto run and also serve as buttons to switch the status. When the first symbol is pressed it means that the auto run is active and the time till the next start is shown. By pressing the middle symbol the auto run can be paused. Thus the time and number of starts are frozen. The auto run can be restarted again by pressing the left symbol again. Any user action for triggering a menu automatically pauses the auto run.

By pressing the rightmost symbol the auto run can be stopped. The menu 'Start/ Auto run/ Start auto run' has the same function.

13.7   DDE Clients

Using the DDE clients a Dynamic data exchange connection to the DDE servers can be established. The program can manage up to eight DDE clients. The DDE server can either work on the same local system or on any other computer (only with Windows NT or 2000). Before exchanging data over the DDE the connection between the client and the server must be established. This can be done using the interpreter or in the tool monitor. An automatic connection is not possible.

Interpreter object: DDEClient

13.7.1   Parameter page

General

DDE Enable
All the DDE clients can be completely disabled here.

Item in tool menu

Show in

Selects whether and where the alias monitor should be shown in the tools menu.

DDE client X

Active
Sets the DDE client active. The entire access to this client is denied if it is deactivated.

Service

Topic
Gives the service name for the opposite sided connection and the topic name for the communication between them. If the server is on another computer (Net-DDE), then in the service of the computer the following value has to be specified. Example: '\\CPCI1\NDDE$'

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 95 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.7.2   DDE-Monitor

The history shows the previous actions occurred in the DDE system. Thus the actions of all channel are displayed together. The number at the beginning indicates the DDE channel number.

On the different tabs the status of each channel are shown. The user can execute a connect or a disconnect and can also send macros and requests.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.
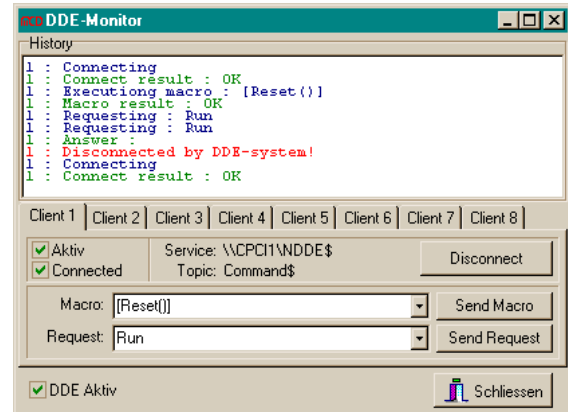
13.8   Exacq high speed AD card

The Exacq XH3240 High-speed data acquisition card helps to capture fast analog signals. The card must be installed completely with all the required drivers. Thus the DLLs required by the card will also be installed.

Interpreter object: ExacqDA

13.8.1   Parameter page

General

DDE Enable
All the DDE clients can be completely disabled here.

Item in tool menu

 Show in
Selects whether and where the alias monitor should be shown in the tools menu.

Page-calibration data

Button: Allow access on calibration data
Usually on opening the basic setup the access to the calibration data is denied and not shown. A click on this button shows calibration data on the sub pages below.

Page-Analog inputs

This page contains a sub-page for each type of analog coupling.

Page-DC

Consists for all the input ranges the offset and gain values. These values are required for the software calibration of the measured values.

The values here are only applicable for the DC coupling. The result is calculated by using the formula:

Result = (Input value-Offset) x Gain

Thus result stands for the value passed on to the program, either in the tool monitor or for the interpreter. Input value is the direct value measured by the card.

Page-AC

Similar to the DC coupling page, but is only applicable for the AC coupling.

MCD Elektronik GmbH	Date: 2018-04-12	Page 96 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

Page-Term

Similar to the DC coupling page, but is only applicable for a terminated DC coupling.

13.8.2   ExacqDA monitor

The first tab 'info' of the monitor show general information about the card and its status.

The second tab shows the individual measured values. The values are refreshed when accessed by the interpreter or when the cyclic checkbox is activated. During the sampling no cyclic measurement is done. By using the progress bar a sedation time between channel switching and measurement can be applied.

The third tab shows the parameters of the sampling and allows to start it.
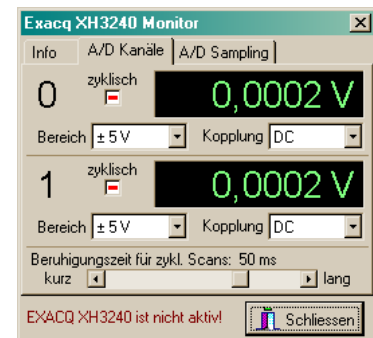
13.9   I²C Bus

The program can communicate in different ways with the I²C Bus according to the type of hardware used. The program reads the data required for I²C Bus from I2CBusses.Dat on program start. See chapter 16.6.

Interpreter object**:** `I2C, PCF8574`

13.9.1   Parameter page I²C Bus

Common

I2C – Enabled
Activates the access to all the I²C Busses.

Item in tool menu

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

13.9.2   I²C Monitor

The I²C Monitor provides a low level access for a defined bus and also to defined I²C modules.

I²C Bus access

The first tab allows a direct accuses to the bus. For each bus an independent tab is shown. For each bus a tab is shown. This tab shows the communication over the bus. On the bottom half a grid is present where frequently used commands can be saved. Below this grid is an edit field where new commands can be entered and saved.

Below this a tool bar with buttons present for low level access. This previous tool bar is present only for those I²C Bus which permit such a low level access.

The commands in the grid can be executed by pressing the 'Exec' button in the right column and can be deleted by pressing the 'minus' sign below. New commands from the edit field can be executed by pressing the 'Exec' button in the right side and can be saved by pressing the 'plus' sign below.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.

All these commands can be sent using the interpreter object. When the cursor is moved over the 'question' mark to the right of the edit field a small help for the command is shown. Further help is provided in the interpreter help for this object.

The functions of the 'low level' buttons in the last row below are:

Start - generates a start condition.

Stop - starts a stop condition.

Send - sends a data byte with value from the nearby edit field to the bus.

Read - reads a data byte from the bus and sends an 'ack'.

RdNAK- reads a data byte from the bus and sends a 'no ack'.

Test toggles the SDAOut connection and checks whether the SDAIn communicates with the signal.

PCF8574

The second main tab of the I²C monitors manages the PCF8574 device. Each device possesses an individual column in the grid. The name of the device is centred on top as heading. On the top right the bus address and right below the device address byte is shown. Below it the current status of the device is shown, once in numbers and also displayed as divided by 8 port pins (P1..P7).

If an error occurs during communication with the device (no Acknowledge error in a SerIO module) then a red 'question' mark is shown to the left.

For an input two bytes are shown, first the output value with 'O' and then the read input value 'I'.
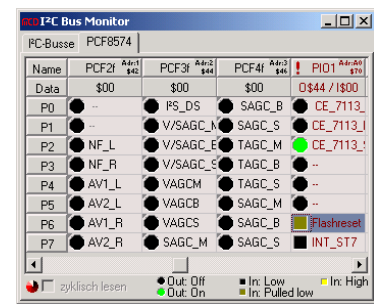
Each cells of the grid represents a pin of the port. The display is configured according to the definition in the I2CBusses.Dat file. If a pin is configured as unused then its name is displayed in grey and the functions of the pins have no influence.

If the pin is an output, then a round symbol (due to pixel formation, but actually it is octagonal) is shown and when its an input a square symbol is shown. For inverted signals a small triangle is shown on the top left corner of that cell. Outputs can be switched over by clicking on the cells. Inputs can be refreshed by using interpreter or by marking checkbox 'read cyclic'. Also an input state can be switched over by a mouse click. But it should be noted that an input must always give a high signal so that it can be pulled down to ground.

The pin description can be changed using the interpreter command (`PCF8574.PIN.INFO.SET`).

13.10 IEC bus

The IEC bus allows the communication with IEC bus / GBIB / IEEE-488 devices.

As hardware a TestManager IEC bus card from manufacturers like ICS, Keithley or National instruments can be used. The cards are available in different variants for different bus PC systems (cPCI, PCI, USB). The correct drivers depending on the operating system must be installed.

To access the card the program uses a DLL supplied by the card manufacturer. This DLL is normally copied during the hardware installation of the installations program of the card in the Windows/System32 directory. If the DLL is absent the program triggers an error message. The cards from ICS and NI both use the DLL 'GPIB-32.DLL'. Both these DLLs are not compatible. Thus if a wrong DLL is used or a wrong manufacturer is selected the program triggers an error message saying 'routine not found in the DLL'.

The drivers and the related DLLs are only accessed if the IEC bus module is activated.

Interpreter object: `IEC`

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 98 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.10.1   Parameter page

The parameters of the IEC bus consists of the of the PCs parameters and the parameters of the devices to communicate with (IEC ports). Multiple number of IEC ports can be defined.

Values on the main page:

General

IEC bus enabled

Activates the access to IEC ports.

Item in tool menu

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

Type of card

The used PC hardware can be selected here

Manufacturer: ICS electronics
This is the standard card used in a Victor system. This card doesn't support an unlimited timeout, thus the program selects the next available timeout.

Manufacturer: Keithley (CEC)
This card is mostly used in PCI systems.

Manufacturer: National instruments (NI)
National instruments NI-488.2 card used in PCI systems.

Addressing – Card

Primary address
Primary Address of the IEC-Card.

Secondary address
Secondary Address of the card. This address is only used if the option below is activated.

Use secondary address
Activates the secondary address and is normally not required.

Overview

Shows a list of the currently defined IEC devices. The colours signify the device status. The red colour signifies an address conflict between 2 devices. Grey signifies inactive devices while green stands for active devices without conflict.

Button: Create new Port
A new IEC port can be created.

Sub page for the Port

For each IEC-Port a sub page is created to set up the port.

IEC bus port

Name
Describes the name of the port to access it from the interpreter. Although the name can have any type of characters, only valid interpreter characters should be used. The name should be unique.

Comment
Any text, reason of use, or a more detailed description of the device. Has only a comment function.

Active
Describes either if the port can be accessed or all access is blocked.

Addressing

Primary address
Primary Address to access the device. This address should also be set up at the device.

Secondary address
Secondary Address to access the device. This address should also be set up at the device. This address is only used if the option below is activated.

Use secondary address
Activates the secondary address should be used (Default off).

Availability test

For activated ports the system can execute an automatic test to ensure that the device is present, switched on and ready to communicate. If an IEC device does not respond as expected, a system error is triggered.

Kind of the tests
Chooses from the 5 types of availability test to be performed:

- none — no test is executed
- only by address — the device is addressed and must only serve the handshake connections.
- send with any reply — the *Test string send* will be sent followed by any reply from the device.
- send with expected reply — the *Test string send* will be sent and it should be followed by the *Expected reply*
- send with similar reply — similar to send with expected reply, but here the reply can be longer than the *Expected reply* and the *Expected reply* is allowed to contain joker symbols like '?'.

Send test string
Decides for the text, which is to be sent at the last 3 types of availability tests. The device must reply to this text.

Expected reply
Describes the reply which the device has to send back for the selected type of availability test.

Communication sequence

Timeout
Limits the time for a device to reply at the sent request. The values of the list are obtained from the driver and no other values can be used.

Receive parameter

EOS-Byte (read)
Describes the byte which is used as EOS byte (End-of-string-byte) from the card. The input is hexadecimal.

Abort data read at EOS-Byte
If activated, the card aborts at the defined EOS byte and provides the text received till then.

End-Of-String
Describes the character sequence for the system to recognise the end of text during communication. The character sequence should be entered as a sequence of hexadecimal bytes. The End-Of-String is deleted automatically during communication.

Send parameter

Set EOI with last sent byte
If activated the EOI signal is set with the last sent byte.

End-Of-String

Describes the character sequence the system automatically adds at the end of a text i.e. to be sent. The character sequence is entered as a sequence of hexadecimal bytes.

Buttons: Copy port

This button copies the data of the current port to a new port.

Buttons: Copy port

This button deletes the current port.

MCD Elektronik GmbH	Date: 2018-04-12	Page 101 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

13.10.2   IEC bus Monitor

The IEC bus monitor can be accessed from the menu tools in the main window or with the button tools from the test window. The menu is inactive if no IEC ports are defined.

The monitor shows a tab for each IEC port. At the top of the tab the status of the port is show. The events on the port are shown in the area below it.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.
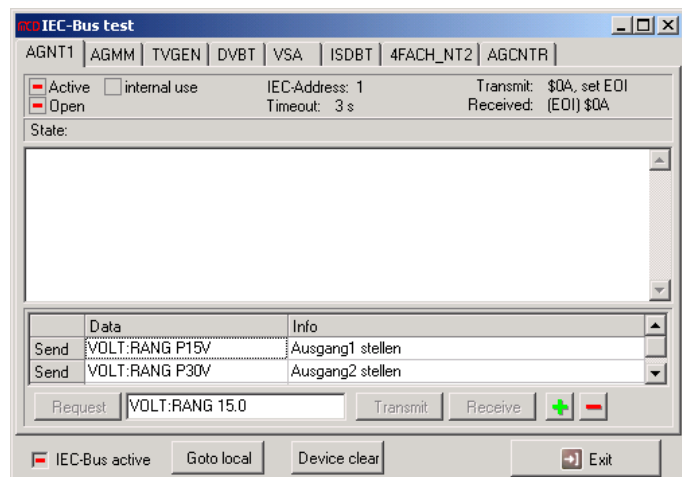
The lower part of the card contains a grid where the frequently used messages are saved and can be sent with a click on 'Request' in the first column. Additionally a new message can be entered in the edit field. With the button 'Send' the message is sent to the IEC device. With the button 'Request' the message is sent and a reply is expected. With the button 'Receive' nothing is sent, only tried to receive a text. The message in the edit field can be saved in the grid by using the plus button. The minus button deletes the current row from the grid. With the buttons 'Goto local' and 'Device clear' the respective commands will be sent t o the IEC device.

For the whole IEC module is shown in the lower left corner. If the IEC module is not activated, a hint is displayed in the event area.

13.11   Calibration data

Although the entry to show the calibration data is present in the individual modules, but for the managing the data (memory location, access from the interpreter) is done by a separate module.

When a calibrations value or the type of saving i.e. the memory location is changed. Then while closing the basic setup the program asks the user whether to save the data or to load the data from the new memory location. Finally it is useful when the data is loaded from a wrong location and is invalid. In this case the data should not be saved in a new location, since the valid files there may be overwritten.

13.11.1   Parameter page

Storage of calibration data

Save in
Selects the memory location for the data. It is from here that the files will be loaded again. The available options are:

- Common basic setup: This selection can lead to problems if the System.ini data is copied to another computer. Thus the calibration data are also copied although it does not belong to this new computer. In the worst case the original calibration data is overwritten by the wrong files and are lost.
- Device depending file: The program uses the hardware code of the system to build a system specific file name for safety.
- User defined calibration file

User defined calibration file
Gives the name for the calibration file if the last option is selected.

MCD Elektronik GmbH	Date: 2018-04-12	Page 102 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Sub page calibration system

This page is devoted for the development of the hardware calibration module for the Victor testing system.

Use calibration data if existing

If the calibration hardware and valid calibration files are present they can be used for the software calibration when activated.

Warn, if calibration data missing

If calibration hardware is present but no valid calibration data then a system error is triggered.

State

Gives information regarding status of the calibration system.

13.12   Key Word 2000

The Key Word 2000 module implements an equal diagnosis protocol. The communication can be made either using K-Line or any other transport protocol (currently TP2.0, TP1.6 also in future).

For communication with K-Line a serial interface with a KW2000 converter (converts RS232 to a one wire bus) is used. The transport protocol is sent using an Optolyzer for MOST or a CAN port according to the requirement.

The Key Word 2000 module automatically creates for every KW2000 port a RS232 port (KW2000, KW2000_2) in the module for serial ports. Here the parameter for this port can be set. For a proper functioning of KW2000 module the module itself, the serial ports and the special port KW2000 have to be activated.

Interpreter object: KW2000

13.12.1   Parameter page

Main page

Item in tool menu

Key Word 2000 - Bus - Enabled
It activates the entire access to the Key Word 2000 module. For a proper functioning of KW2000 module the module itself, the serial ports and the special port KW2000 along with the related RS232 port have to be activated.

Item in tool menu

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

Sub page for each port

Decides the type of protocol for the ports and that part of parameter i.e. independent of the protocol type.

COM Port

Active
Activates the port. An inactive port cannot access the connected hardware of any type.

Kind of port
Selects the type of related transport medium and the transport protocol.
It is also possible to switch the kind of port from interpreter when for e.g. the CAN port is used with different protocols.

Internal handling of negative response codes 7F SID XX

Code 21 - "Busy-RepeatRequest" – handle internally
If activated then the negative replies with Code $21 will be handled internally, and the request is then repeated again as per the time given below.

Code 78 - "ReqCorrectlyRcvd-RspPending" - handle internally
If activated then the negative replies with Code $78 will be handled internally, till a reply is received.

Task Complete Timeout
Gives the maximum time-out time for the request handling of the negative replies.

Delay before repeated request
Gives the maximum delay time before the request handling of the negative reply Code $21 and the internal handling.

13.12.1.1   Protocol type Kline via RS232

Parameter for Kline via RS232

Used COM port
Shows the used RS232 port which cannot be edited. The port specific settings like (baud rate, data format etc.) can be changed at the corresponding port in the of module serial ports. Using the interpreter these values can be temporarily altered.

Timing

Inter-Byte-Time TG (P4/P5)
The minimal time between the sending of two bytes (Protocol time P4). Due to the hardware structure of the PC this time limit is not guaranteed. In the Key Word 2000 protocol this is the time limit required for the reply from the DUT to the tester (Protocol time P5). The system cannot measure this time and accepts the data also without Inter-Byte-Time.

Timeout Receive (P2)
Describes the time the DUT should reply to a request by the tester. (Protocol time P2) If the DUT does not reply, then this results to a timeout.

Line Idle Time Min (P3)
Describes the time the tester has to wait to send another request, after the DUT has replied (Protocol time P3)

Line Idle Time Max (P3)
Describes the maximum time limit within which the tester should send a new request to keep the DUT alive (Protocol time P3 max)

Addressing

Addressing mode
Describes how the first message byte should be structured. The CARB mode can be selected, but is not supported because it is required only by high level functions. Low level functions can build their messages completely by themselves.

Tester-(PC)-Address
Describes the address of the PC i.e. used in the high level functions. Depending on the addressing mode a physical or a functional address can be interpreted. The addressing mode 'no address' is without any function.

DUT-(ECU)-Address
Describes the address of the DUT (Electronic Control Unit) i.e. used in the high level functions. Depending on the addressing mode a physical or a functional address can be interpreted. The addressing mode 'no address' is without any function.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany	Date: 2018-04-12	Page 104 of 181	
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Title: MCD TestManager CE - Software Manual			

Physical connection

Connected by
Depending on the realisation of the Kline a 1-wire port or as a pseudo Kline a 2-wire port the program must ignore (1-wire) or accept (2-wire) its echoes respectively

13.12.1.2   Protocol type TP1.6 via CAN

Parameter for TP1.6 via CAN

CAN port
Shows the CAN port used for the communication. A CAN port with the same name must be defined under CAN modules. Using the interpreter these values can be temporarily altered.

Addressing

Tester KW2000 address
Address of the tester in KW2000 protocol. This will be used as the sender address for service IDs in the KW2000 frame (Default: $F1).

Tester TP1.6 address
Address word of the tester to open KW2000 using the TP1.6 protocol. The documentation for transport protocol (Appendix A) can be used (Default : $F0).

Tester-opening channel
CAN ID used by the tester to send the opening message. The documentation for transport protocol (Appendix A) can be used (Default : $2D0).

ECU KW2000 address
Address of the DUT in KW2000 protocol. This address will be used as target address in KW2000 Frame by the Service IDs (no default).

ECU TP1.6 TP address
Address word of the DUT to open the KW2000 using TP1.6 protocol. The documentation for transport protocol (Appendix A) can be used (no default).

ECU TP1.6 CAN address word
Address word of the DUT to open the TP1.6 protocol. The documentation for transport protocol (Appendix A) can be used (no default).

ECU-opening channel
CAN ID used by the DUT to send the opening message. The documentation for transport protocol (Appendix A) can be used (no default).

ECU-request channel
CAN ID used by the tester to send the TP messages to the DUT. The documentation for transport protocol (Appendix A) can be used (no default).

ECU-response channel
CAN ID used by the DUT to reply to the TP messages of the tester. The documentation for transport protocol (Appendix A) can be used (no default).

Timing

T1 Timeout acknowledge
Time within which the DUT must reply to a request. Affects only the direct replies for a message.

Receive timeout
Time within which the DUT must reply to a request completely.

MCD Elektronik GmbH	Date: 2018-04-12	Page 105 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Miscellaneous

TP1.6 block size
Number of messages, after which the DUT will be requested for an acknowledge. This value will be compared with the block size of the DUT and the smaller value will be used. After the last message of a frame an acknowledge is always requested.

Leave CAN open when closing KW2000 port
Leaves the CAN open even after the KW2000 port is closed.

Keep TP1.6 channel alive?
If the transport protocol is to be kept alive, then the KW2000 module starts a cyclic communication with tester-present messages. This keeps the channel open. The following communication need not reconstruct the communication and thus it is faster.

Example for entering a Diagnose communication (from the testers point of view)

Tmt: = Transmit, Rcv: = Receive, The values having grey background can be influence by the basic setup, other values are fixed or calculated dynamically.

Tmt: ID:2D0 D:37 C0 E7   **Opening the connection by the tester**
    2D0 – *Tester-Opening channel*
        37 – *ECU TP1.6 CAN Address word*
            C0 – Opcode Connection-Request
                E7 – lower 8 Bit von *ECU-Request channel*

Rcv: ID:4D7 D:F0 D0 F7   **Reply from the DUT**
    4D7 – *ECU-Opening channel*
        F0 – *Tester TP1.6 Address*
            D0 – Opcode Positive Connection-Reply
                F7 – lower 8 Bit von *ECU- Response channel*

Tmt: ID:4E7 D:A0 05 6D 8A 45 CA   **Connection Setup by the tester**
    4E7 – *ECU-Request channel*
        A0 – Opcode Connection Setup
            05 – *TP1.6 Block size*
                6D – *T1 Timeout Acknowledge* (Timing value is coded in OSEK)
                    8A 45 CA – the remaining timing parameter of the tester (T2 .. T4)

Rcv: ID:4F7 D:A1 05 94 54 4A F2   **Connection Acknowledge by the DUT**
    4F7 – *ECU- Response channel*
        A1 – Opcode Connection Acknowledge
            05 – Block size of the DUT
                94 54 4A F2 – the remaining timing parameter of the DUT (T1 .. T4)

At this point the TP1.6 Transport channel has been constructed. Only the diagnose must be started now. From here onwards the data will be exchanged in TP1.6 protocol. This increase the overheads.

Tmt: ID:4E7 D:10 57   **Request of Key-Words**
    4E7 – *ECU-Request channel*
        10 – Transport Protocol Information Byte TPCI – Data followed, Frame-number 0, last frame, Acknowledge is requested
            57 – Reference data: *ECU TP1.6 TP Address* K-Line address word of the DUT

Rcv: ID:4F7 D:B1   **Acknowledge of the DUT to the tester message**
    4F7 – *ECU- Response channel*
        B1 – Transport Protocol Information Byte TPCI – Acknowledge for frame 0

Rcv: ID:4F7 D:10 55 E9 8F   **Synchronisation byte and Key-Word from DUT**

    4F7 – *ECU- Response channel*

        10 – Transport Protocol Information Byte TPCI – Data followed, Frame number 0, last frame, Acknowledge is requested

           55 E9 – Reference data: Synchronisation byte, 1. Key-Byte (will not be used further)

              8F – Reference data: 2. Key-Byte

Tmt: ID:4E7 D:B1   **Acknowledge of the tester to the DUT message**

Tmt: ID:4E7 D:10 70   **Return the inverted 2. Key-Byte to the DUT**

    4E7 – *ECU-Request channel*

        10 – Transport Protocol Information Byte TPCI – Data followed, Frame-number 0, last frame, Acknowledge is requested

           70 – Reference data: inverted 2. Key-Byte

Rcv: ID:4F7 D:B1   **Acknowledge of the DUT to the tester message**

Rcv: ID:4F7 D:10 A8   **Return the inverted address words**

    4F7 – *ECU- Response channel*

        10 – Transport Protocol Information Byte TPCI – Data followed, Frame-number 0, last frame, Acknowledge is requested

           A8 – Reference data: inverted *ECU TP1.6 TP Address* K-Line address word

Tmt: ID:4E7 D:B1   **Acknowledge of the tester to the DUT message**

At this point the Diagnose communication is established and KW2000 communication can be continued. From here on the reference data will be packed in KW2000 messages and further on to TP1.6 messages. These naturally increases the overheads more.

Tmt: ID:4E7 D:10 82 82 F1 10 89 8E   **KW2000 command to the DUT**

    4E7 – *ECU-Request channel*

        10 – Transport Protocol Information Byte TPCI – Data followed, Frame-number 0, last frame, Acknowledge is requested

           82 – KW2000 Header: Format byte, Physical address, 2 Byte Reference data

              82 – KW2000 Header: Target address: *ECU KW2000 Address*

                  F1 – KW2000 Header: Source address: *Tester KW2000 Address*

                    10 89 – Reference data, SID: 10, Parameter 89

                        8E – KW2000 Check sum

Rcv: ID:4F7 D:B1   **Acknowledge of the DUT to the tester message**

Rcv: ID:4F7 D:10 82 F1 82 50 89 CE   **Response of the DUT to the KW2000 command**

    4F7 – *ECU- Response channel*

        10 – Transport Protocol Information Byte TPCI – Data followed, Frame-number 0, last frame, Acknowledge is requested

           82 – KW2000 Header: Format byte, Physical address, 2 Byte Reference data

           F1 KW2000 Header: Target address: *Tester KW2000 Address*

           82 KW2000 Header: Source address: *ECU KW2000 Address*

           50 89 –Reference data Positive Response to SID 10 and Data byte

           CE – KW2000 Check sum

Tmt: ID:4E7 D:B1   **Acknowledge of the tester to the DUT message**

Till here 2 Bytes of reference data are sent and 2 Bytes are received!

MCD Elektronik GmbH	Date: 2018-04-12	Page 107 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

13.12.1.3 Protocol type TP2.0 via CAN

Parameter for TP2.0 via CAN

CAN port
Shows the CAN port used for the communication. A CAN port with the same name must be defined under CAN modules. Using the interpreter these values can be temporarily altered.

Addressing

Tester TP2.0 address
Address word of the tester to open KW2000 using the TP2.0 protocol. The documentation for transport protocol (Appendix A) can be used (Default : $F0).

ECU TP2.0 TP address
Address word of the DUT to open the KW2000 using TP2.0 protocol. The documentation for transport protocol (Appendix A) can be used (no default).

CAN RX identifier
It is the CAN receive identifier which the tester uses to send its reply.

Timing

T1 Timeout acknowledge
Time within which the DUT must reply to a request. Affects only the direct replies for a message.

Receive timeout
Time within which the DUT must reply to a request completely.

Miscellaneous

TP2.0 block size
Number of messages, after which the DUT will be requested for an acknowledge. This value will be compared with the block size of the DUT and the smaller value will be used. After the last message of a frame an acknowledge is always requested.

Leave CAN open when closing KW2000 port
Leaves the CAN port open even after the KW2000 port is closed.

Keep TP2.0 channel alive?
If the transport protocol is to be kept alive, then the KW2000 module starts a cyclic communication with tester-present messages. This keeps the channel open. The following communication need not reconstruct the communication and thus it is faster.

13.12.1.4 Protocol type TP2.0 via MOST

Parameter for TP2.0 via MOST+-

Used COM port
Shows the used RS232 port which cannot be edited here. It is here that the 'Optolyzer for MOST' should be connected. The port specific settings like (baud rate, data format etc.) can be changed at the corresponding port in the of module serial ports.

Addressing

Tester TP2.0 address
Target address of the tester to open KW2000 using the TP2.0 protocol. The documentation for transport protocol (Appendix A) can be used.

ECU TP2.0 TP address
Target address of the DUT to open the KW2000 using TP2.0 protocol. The documentation for transport protocol (Appendix A) can be used.

MCD Elektronik GmbH	Date: 2018-04-12	Page 108 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

CAN RX identifier

TP2.0 for MOST constructs the frames in a capsule of TP2.0 for CAN. Thus a CAN identifier is required, for the tester to receive its reply.

It is the CAN receive identifier which the tester uses to send its reply.

Tester MOST address

Address of the tester in the MOST ring.

ECU MOST address

Address of the DUT in the MOST ring.

Timing

T1 Timeout acknowledge
Time within which a confirmation should be received, when its is required by the transport protocol.

Receive timeout
Time within which the DUT must reply to a service request.

Miscellaneous

TP2.0 block size
Number of messages, after which the DUT will be requested for an acknowledge. This value will be compared with the block size of the DUT and the smaller value will be used.

Leave RS232 open when closing KW2000 port
Leaves the RS232 port open even after the KW2000 port is closed.

Keep TP2.0 channel alive?
If the transport protocol is to be kept alive, then the KW2000 module starts a cyclic communication with tester-present messages. This keeps the channel open. The following communication need not reconstruct the communication and thus it is faster.

13.12.2   Key Word 2000 - Monitor

The Key Word 2000 monitor can be accessed from the menu tools in the main window or with the button tools out of the test window.

The monitor shows a separate tab for each port. In the top half the events occurring on the bus are shown.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.
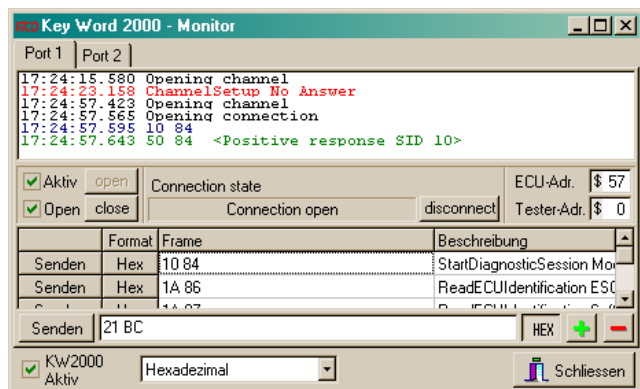
The lower part contains a grid where the frequently used messages are saved and can be sent with a click on 'Send' in the first column. Additionally there you can enter a new message in the edit field. With the button 'Send' the message is sent to the DUT and a header is created according to the settings of the module. The message in the edit field can be saved in the grid with the plus button. The minus button deletes the actual row from the grid.

Using the button 'HEX/ ASC' the format in the edit field can be interpreted and switched. The same switching function is possible for the commands in the grid by clicking on the second left column in the grid.

The part of the monitor below the event area depends on the transport medium and protocol. It is possible to open or close the port here. Also the status of the port is shown here.

MCD Elektronik GmbH	Date: 2018-04-12	Page 109 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

For the entire KW2000 module the activity is shown in the lower left corner. If the module is inactive, a hint is displayed in the event area.

13.13 Curves module

The curves module is a tool for visualising any analogue curve. It also possesses functions for analysing such curves. Curves extracted from any source, e.g. sampling with the ME3000 card, reading from an oscilloscope or reading any previously saved curve can be edited and analysed in a homogenous way. A curve is a collection of many analog value, which are related to the X axis with respect to time. At a moment only one curve can be managed.

Interpreter object: Curve

MCD Elektronik GmbH	Date: 2018-04-12	Page 110 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

13.13.1   Curves monitor

The Curves monitor can be accessed from the menu Tools in the main window or with the button Tools out of the test window.

The actual curve set is shown. With a click and drag on the curve using the left mouse button you can zoom in that area. With a click using the right mouse button the system zooms completely out and the entire curve is displayed in the window.

Additionally for every curve the average (arithmetic mean), the minimum and the maximum values are shown in doted lines. The colour of each line is according to the colour of the curve. Display of the curves can be switched on or off in the menu.
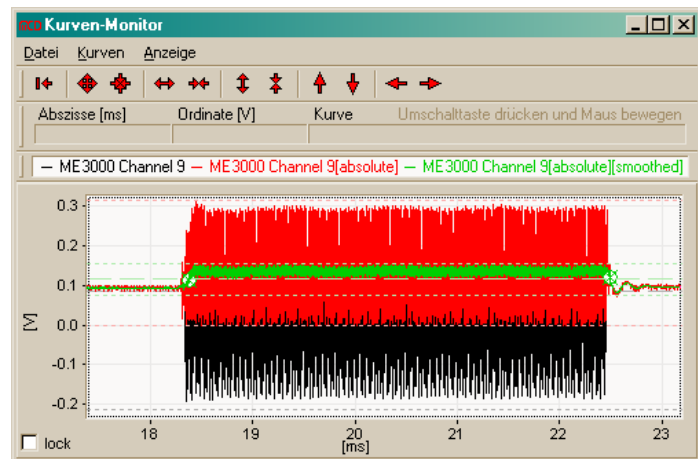
The display can also be edited with the buttons on the tool bar or with the following keyboard shortcuts:

	Zoom completely out	(Pos1)			
	Zoom in	(PgUp)		Zoom out	(PgDn)
	Zoom in vertically	(Ctrl-Down)		Zoom out vertically	(Ctrl-Up)
	Zoom in horizontally	(Ctrl-Left)		Zoom out horizontally	(Ctrl-Right)
	Move window up	(Up)		Move window down	(Down)
	Move window left	(Left)		Move window right	(Right)

If the Shift key is pressed when moving the mouse over the curve, then the time(X) and the value(Y) of the curve when the cursor is currently positioned on the curve will be shown at the top in the measuring bar.

With the menu file the actual curve set can be saved or a previously saved curve set can be loaded. With the menu curves the curves can be added or removed. With the menu display the tool bar and the measuring bar can be switched off.

Normally the curve monitor scales itself such that the curve is entirely visible in the monitor. If the same scale is required for the next new curve then the check box lock must be selected.

13.13.2   Parameter page

Item in tool menu ───────────────

Show in

Selects whether and where the alias monitor should be shown in the tools menu.

13.14   LIN

Hint: This part of documentation is only an expansion of the LIN function to the slave task. This expansion is not yet complete and will be completed till the version 1.3.

In the earlier versions the LIN communication was implemented only as an I/O communication and was entirely controlled by the interpreter. The new version helps to use the LIN description files

(LDF) to internally build a complete master and slave task. The communication is developed by a thread in the background. The option for selecting the mode of communication whether in old I/O mode or in the new task mode can be made using the basic setup. Depending on the selection the respective monitor is displayed.

The program uses for the LIN bus a RS232 port and generates a port called 'LIN' automatically in RS232 module. On the port an external K-Line converter should be connected to realise the physical layers. The LIN timing can only generated by software and cannot be guaranteed due to Windows. No problems have been found till date.

Interpreter object: LIN

13.14.1   LIN in I/O mode

13.14.1.1   Function of the I/O mode

In I/O mode the program can work only as the master task in a LIN system. The Slave task is not realised. The program 'replies' only to self sent Slave tasks to send the messages. Each communication is performed by the interpreter itself. A definite master task can be sent cyclic to keep the DUT alive.

13.14.1.2   LIN Monitor in I/O mode

The events on the LIN bus are shown in the top part of the monitor.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.

Below it the state of the bus is shown. This can be opened and closed using the left buttons. A cyclic message (from right edit field) can be sent in the background when during this time no other communication is done using the bus.

Using the edit field at the bottom a master task with or without data can be sent and a reply can be waited for.

If the checkbox 'Log' is selected all the events shown in the monitor above will be saved in '%LOG%\LIN.LOG'. Each row will have the structure: Timestamp, port, action (S = Send, R = Receive, C = Control), data where the master task is shown in round brackets.

Example:
```
10:02:02.85 | LIN1 | S | (32) 12 34 56
10:02:09.47 | LIN1 | S | (31)
10:02:09.69 | LIN1 | R |  <timeout error>
```
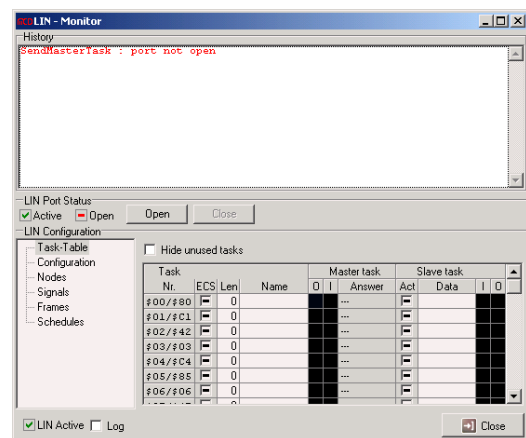
13.14.2   LIN in task mode

13.14.2.1   Function of the task mode

In the task mode the communication is not controlled by the interpreter, but from a task table. The program processes the table in the background. Thus a master or a slave function can be performed. The task table can be constructed using the 'LIN definition files' or by using an interpreter command.

The implementation of task mode is not entirely complete and so is the documentation for it.

13.14.2.2   LIN Monitor in task mode

The implementation of task mode is not entirely complete and so is the documentation for it.

13.14.3   Parameter page

Common

LIN – Enabled
Activates the LIN port. For the proper functioning of the LIN bus an RS232 port should also be activated.(also in Task mode the title Automotive interface will be used)!

Open automatically
Opens the LIN port automatically if activated. Otherwise it must be opened using the interpreter.

COM-Port
Gives a hint about the RS232 port.

Operating mode

Mode
Selects between the LIN I/O mode or the task mode for operation.

Hardware
Selects whether the protocol should be generated by the program using the RS232 port for execution or by using an Automotive interface from the manufacturer Kopf. The using of the USB interface is only possible in the task mode.

Kopf Automotive interface serial number
If in the task mode the Kopf Automotive interface is used, then the serial number of the interface can be entered here. Thus a specific interface can be connected when multiple interfaces are connected to the system. If no number is entered then the first interface is connected as default. Thus it is necessary to enter some value here when many programs would like to access a specific interface.

Normal baud rate
Decides for the communication baud rate. It can be overwritten by a baud rate value from an LDF.

Sync. baud rate
Decides for the baud rate which will be used to develop the synchronisation-breaks. If 0 is entered the baud rate will be automatically calculated. This is not required if the Kopf Automotive interface is used.

Timing

Values here will only be considered if LIN is used in the I/O Mode.

Keep Alive Time
It is the time after which the Keep-Alive Frame should be sent on a silent bus (a bus without communication).

Additional Sync delimiter delay
This option was introduced due to the customer requirement in order to increase the SyncDelimiter as compared to the default LIN protocol value. This value should be very carefully changed (if required only). Default value is 1080µs.

LIN Definition

Values here will only be considered if LIN is used in the task mode.

Load this LDF file after program reset

LIN Description File (LDF)
Describes the LDF file which should be automatically loaded.

Item in tool menu

Show in
Selects whether and where the alias monitor should be shown in the tools menu.

13.15   Local statistics

Local statistics shows the events occurring in the system. It is not foreseen to make statistical valuation for e.g. process efficiency or testing instrument efficiency.

The data management of the local statistics is controlled automatically by the program and cannot be deactivated.

The values of local statistics can be accessed using the interpreter.

The value of the local statistics is saved in a file at the end of the program (%SYS%LoclStat.ini and %TYP%\TypeStat.ini) and is loaded again at the next program start.

Interpreter object: `LocalStat`

13.15.1   Elements of local statistic

13.15.1.1   Statistical values

The statistical values of the local statistics possess a distinct serial number (SVID Statistical Value IDentification). These values are concluded to groups which have a distinct serial number (SGID Statistical Group IDentification).

Group SGID	Value SVID	Menmonic	Symbolizes
1000		Program actions	Program actions
	1001	Prog starts	Number of program starts
	1002	Prog resets	Number of resets done
	1003	Type changes	Number of Type changes
1100		Starts	Start events for all Type variants
	1101	Test starts	All the column starts
	1102	Row starts	All the row starts
	1103	DUT starts	DUT starts
2000		Test starts	Start events for all Type variants
	2001	Type Test	Column starts
	2002	Type Row	Row starts
	2003	Type DUT	DUT starts
2100		Test results	Test results for current Type variants
	2101	Type PASS	good DUT
	2102	Type FAIL	fail DUT
	2103	Type Abort	Stopped
	2104	Type Total	Total
2200		Previous error	Information about previous error
	2201	Type Last	Step-Nr
	2202	Type Cont Count	Count of sequential errors
2300		Top error	Error statistic for current Type variants
	2301	Type Top	Top-error
2400		Contact column unit	Contact column unit
	2400+x	Contacts Group x	Contact column counter group x (x = 1..4)
	2410+x	Fails Group x	Error group x (x = 1..4)
	2420+x	Cont.Fails x	Consecutive error group x (x = 1..4)
	2430+x	Same Fails x	Self consecutive error group x (x = 1..4)
9000		UserManaged Values	User managed values
	9000+x	UMV#x	User managed values x (x = 1..20)

MCD Elektronik GmbH	Date: 2018-04-12	Page 114 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

13.15.1.2   Info-column entries

The values of the local statistics can be shown on the left info-column of the desktop window.

13.15.1.3   Trigger

On changes of the local statistic values, they can be compared with their limits. If an over limit occurs an event can be triggered. The trigger should be defined in the basic setup and cannot be defined from the interpreter.

13.15.1.4   Contact unit

An important topic for the management of a test system is the counting of the number of mechanical systems used for e.g. the needle-bed, DUT holder, mechanics etc.. As the program should be able to meet all possible requirements the contact unit was introduced. A contact unit is generally a element which is useful whiles testing a DUT. It is only then useful because for e.g. a DUT holder is only then useful when it is present in the testing system.

A contact unit can be defined in the local statistics window (see below) and posses a distinct name. It belongs to one of the 4 groups and is either active or inactive during the test. If it is active then its use and error counters are influenced from the test results.

There are 3 types of contact units

'Column' – this unit can be manipulated, when it is active and a column test is started.

'Row' – this unit can be manipulated, when it is active and a row test is started, which belongs to a particular column.

'DUT' - this unit can be manipulated, when it is active and a DUT test is started, which belongs to a particular column.

13.15.2   Parameter page

Management of type data counters ────────────

When to save data
Decides when the change values should be entered into the permanent program data. Normally for good performance this must always be done at program resets or at program ends. If any external program accesses these values then the data can be saved after every test end.

Continuous failure management ────────────

This failure counts the continuously occurring errors of the same type. Thus it can be found for e.g. that the error is not caused because of the DUT but due to the testing method. Normally the error counters are not set to zero on PASS but retain their present value. But they are incremented by one when the same error occurs and set to one if another error occurs. The error must not occur consecutively. Alternatively the counters can also be set to 0 on PASS. In this case the same error must occur consecutively.

Reset type continuous failure counter on PASS
Decides the management of continuous failure counter for the type variants.

Reset DUT continuous failure counter on PASS
Decides the management of the individual continuous failure counter for the DUT testing units depending on the contact columns. For each contact columns of a testing unit a continuous failure counter will be managed.

User Managed Values ────────────

User Managed Values was introduced to show any required value in the statistical area of the info-column. These value is further irrelevant for the program. However it may be interesting because these values can be altered by the interpreter.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 115 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.15.2.1 Sub page info-column

The details on the Info bar is observed on the right hand side of the desktop window.

Button: Create new entry in info-column
It Create new entry in info-column. For every new entry in info-column a separate sub-page is created.

13.15.2.2 Info-column entry:

Statistical value
Selects the fixed statistical value or the value A to be calculated with the formula below can be selected from the combo box options This is shown in the info-column.

Statistical value B
Selects the statistical value B required in the formula. It is only considered if a percent value has to calculated between A and B.

Display
Selects whether the statistical value or the percent value should be displayed.

Labels
Text, to show the value in the info-column.

Comment
A comment can be entered here. This is irrelevant for the program.

Unit
Unit, which may be required for that value.

Show unit
Selects the position to display the unit between:
- Above the box, with the label: above the text to the right side.
- Right hand side of the box, after the value: in the same row of the text at the right end.

Precision
The position of decimal point in order to format the value.

Standard colour
Colour of the value, when it lies within limits and is valid.

Fail colour
Colour of the value, when it lies outside the limits and is valid.

Invalid colour
Colour of the value, when it is valid for e.g. division by 0.

Tested limits
Selects from with which of the limits the value should be compared. It influences the display of the value.

Lower limit
Lower limit for the value, is only applicable if in 'tested limits' the option lower limits or both limits is selected.

Upper limit
Upper limit for the value, is only applicable if in 'tested limits' the option upper limits or both limits is selected.

Position ─────────────────────

Button: up
Moves the entry upwards by one position

Button: down
Moves the entry downwards by one position

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 116 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Button: Copy entry
Creates a new entry in the info-column and copies the values of the current entry.

Button: Delete entry
Deletes the entry from the info-column.

13.15.2.3   Sub page Trigger

Button: new statistic trigger
Creates a new statistic trigger. For every new statistic trigger a separate sub-page is created

13.15.2.4   Sup page for each statistic trigger

Trigger

Description
Description of the trigger as text. It is ignored by the program but however when triggered an entry in the program log book is made.

Active
Activates the trigger.

Statistical value A
Selects the statistical value to be watched

Check on
Decides the point of time, to check the values against the limits.

Checked limits
Selects from with which of the limits the value should be compared. It influences the display of the value.

Lower limit
Lower limit for the value, is only applicable if in 'tested limits' the option lower limits or both limits is selected. If the limit is active and the value is below the limit then the action is trigger.

Upper limit
Upper limit for the value, is only applicable if in 'tested limits' the option upper limits or both limits is selected. If the limit is active and the value is above the limit then the action is trigger.

Action
Decides for the action to be executed as soon as the limit has been crossed.

Parameter
Parameter for the selected action.

Button: Copy trigger
Creates a new trigger and copies the values of the current entry.

Button: Delete trigger
Deletes the current statistical trigger.

13.15.3   Local statistics monitor

The local statistics monitor can be accessed from the menu, or with the shortcut 'F6'.

The values of local statistics are shown and can be manipulated on a number of tabs in the tool monitor.

The first tab 'program' shows the values independent to the type variants. The individual groups can be deleted using the 'reset' button here.

The second 'actual type' tab shows the values depending on the type variants. Along with the number of starts and test results the fails are arranged according to their frequency on a second tab here.

The third tab 'User managed values' corresponds to the local statistics page in the basic setup.

The fourth tab manages the 'contact units'.
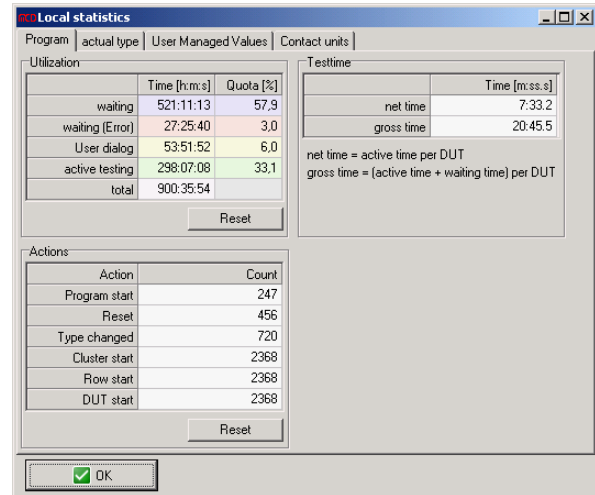
The defined contact units are shown in a grid. The individual columns are:

Active: shows whether the contact unit is activated and thus manipulating the start.

Error: Shows whether an error had occurred on the previous execution of the contact position. At the foot of the table a hint for the error will be shown.

Name: Identifier, which can be activated by the interpreter. Should follow the interpreter character rules and must be distinct.

Group: Multiple contact units can be fused into a group, for this group a statistic trigger can be assigned, which controls all the members of this group. At the moment 4 groups are assigned.

Contacts: Number of test this contact position has performed.

Error: Number of times the test was a failure.

Continuous error: Shows the number of times the DUT contact unit had consecutive fails (also with different errors). It is unequal to 0 only when the previous test was a fail.

Same error: Shows the number of times the DUT contact unit had consecutive fails with the same error. It is unequal to 0 only when the previous test was a fail.

Using the button in each table the individual values can be set back to 0. If the checkbox 'All' is selected then all the values of the contact unit are set to 0.

The plus button creates a new contact unit, the button with the hand edits the selected contact unit and the minus button deletes a contact unit.

13.16   ME3000

The ME3000 module establishes a connection to the Meilhaus ME3000 multifunction card. To access the card, the card and the required drivers have to be correctly installed. If the module is switched off, no card or driver is needed.

Care should be taken while installing a new card, which should be generally ordered through MCD. The cards from MCD distinguish from the other cards as they have special serial numbers which allow an easier driver installation (especially changing two cards without new installation) and allowing the expanded functions enabled.

Interpreter object: ME3000

13.16.1 Parameter page ME3000

General

ME3000 – Enabled
Activates hardware access to the ME3000 card.

Digital inputs

Execute time measurement with real time priority
If this option is activated then the command ME3000.DIn.Meas of the interpreter is executed with real time priority. The measurements are done with an accuracy in µs. However other real time applications running simultaneously on the computer may receive less computing time from the system.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Page for calibrations data

Button: Enable access to calibrations data
On opening the basic setup initially the access to the calibration data is disabled and hidden to avoid accidental changes. By clicking this button the calibration data is shown on a sub page for edition.

Page for analog outputs

This page consist of the calibration data for the 4 analog outputs and different output ranges. For each channel/range combination a range and offset value can be programmed. The calculation on the card for the given value is done according to the formula:

Displayed value = (Raw value - Offset) x Gain

Raw value is the value i.e. to appear on the card or the testing unit. Displayed value is the value i.e. sent to the card.

The calculation procedure is described in the ME3000 object of the interpreter help.

Page analog inputs

This page consist of individual pages for the different ranges of the 4 analog inputs. The sub page consists of all range and offset values for all the channels. These can be calibrated by software.

The calculation is done according to the formula:

Output value = (Set value - Offset) x Gain

Thus the output value will be further used in the program to display the result or by the interpreter. Set value is the value read by the card.

The calculation procedure is described in the ME3000 object of the interpreter help.

13.16.2 ME3000 – Monitor

13.16.2.1 Info tab

The info tab gives information about the status of the card and the drivers. It is shown here whether the ME3000 module is enabled and ready. The checkbox 'Cyclic requests' activates a cyclic reading for the tabs which read information from the card. If this checkbox is not selected, then the values change only if the interpreter reads a channel. The read value will also be shown in the monitor.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 119 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

The cyclic requests do not affect the ME3000 measurements except for the timing. (The ME3000 card will be accessed more frequently, so the interpreter action may be delayed). However due to the active samplings no measurements using the AD converter can be made.

13.16.2.2   Digital IO tab

This tab shows the 16 digital inputs and 16 digital outputs of the ME3000 card.

The inputs are read and displayed in a cyclic manner, if the checkbox is marked. Otherwise the ports are only refreshed and displayed if the interpreter reads the inputs. The channels C-F are system inputs and cannot be used by the interpreter. A yellow circle stands for high signal at the input, a dark circle means low signal.

The outputs will be refreshed when their state changes. A green circle stands for high signal and a dark circle means a low signal. The channels 0-7 can be switched with a double click or by using the space key. The key plus switches to on, the key minus switches to off. The channels 8-F are used from the system and cannot be changed by the interpreter or manually. As the outputs to the UMS (see UMS module) are done using the ME3000 card internally, each action on the serial bus is seen on the digital IO tab. This highly slows down the digital I/O page of the UMS. This can be important or unimportant according to the problem situation.

13.16.2.3   A/D channel view tab

This tab shows the 16 analog inputs of the ME3000 card.

The channels are read in a cyclic manner and displayed if the checkbox is marked. Otherwise the display is only refreshed if the interpreter reads the respective channel. The last read value and the measuring range is displayed on each row. The measuring range can be set for the individual channels using the buttons on the right with down arrow symbol.

Using the scroll bar at the bottom end of the window the settling time before the cyclic scans can be adjusted. If the settling time is too less and thus the impedance of the source signal is too high, then the measuring channel has very less time to swing at the right value. It can thus lead to a wrong measurement. However the load on the system is reduced when shorter settling time is used. Only those channels marked with green tick symbol cane read back in a cyclic manner. By clicking on this green symbol the activity of the channel can be changed.

13.16.2.4   A/D Sampling-parameter tab

On this tab the parameter for the sampling action can be set and displayed (Sampling a given number of samples for one or more channels with a given sample frequency).

All parameter of the sampling can be set. Since all the parameters are interrelated with each other thus the field 'Parameter check' displays if a sampling is possible (e.g. the maximum sample frequency depends on the number of channels).

Using the buttons 'Start' and 'Abort' the sampling can be started or a running sampling can be stopped.

The buttons 'Start' is active only when the sampling parameters are valid.
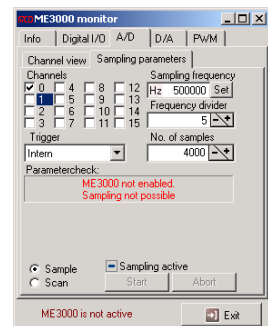
13.16.2.5   D/A channels tab

The tab D/A channels shows the status of the 4 analog output channels. A new value can be entered in the white edit fields and is activated on pressing the 'Set' buttons. The output range can be set for each single channel with the buttons with the down arrow symbol.

Attention:

Concerned only for Victor test-system

The Victor test-system possesses after the DA converters of the ME3000 power amplifiers which increase the current carrying capacity of the outputs. The voltage output is also increased by factor 1.5. If the voltage amplification values are set in the calibration factors, then the output voltage of the test system are identical or lower to the amplification factor.

Concerned only for Victor test-system (only for Variant 1)

In systems with a internal regulated linear power supply the fourth DA channel is used for controlling the power supply (only for Variant 1). The voltage supplied by the DA converters is not the actual output voltage of the power supply. Its voltage is amplified depending on the calibration factor of the ME3000. Thus the power supply produces a voltage output equivalent to the program display by calculating the voltage amplification values.

13.16.2.6 D/A Curve tab

The ME3000 card can output a arbitrary function on a DA converter. Using this tab the defined curve can be displayed. The output can be controlled using the buttons 'Start' and 'Stop'. Also in the area 'curve-synthesise' a square, sine, or saw tooth function can be defined and given out. The amplitude is thus the difference from the highest signal point to the offset voltage (Vss/2).

The offset ± amplitude must be within the output range of the chosen channel, otherwise the curves will cut at the borders of the output range.

The curves are calculated by clicking on the buttons 'Sin' (sine), 'Sq' (Square), or 'Saw' (triangle). For square wave signal output the edit field parameter describes the relation between high and low phase. For saw tooth signals the edit field parameter describes the relation between rise time and fall time. For a sine wave this parameter has no function.

The topics discussed above regarding the displayed and actual voltage values are also applicable here.
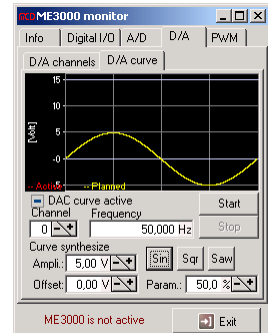
13.16.2.7 PWM tab

Displays the status of the PWM channel of the ME3000 card and allows manual changes.
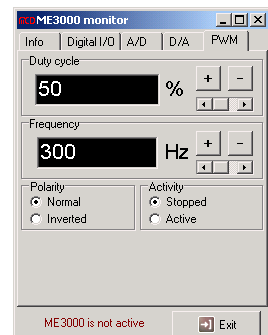
13.17 ME4000

The ME4000 module establishes a connection to the Meilhaus ME4000 multifunction card. To access the card, the card and the required drivers have to be correctly installed. If the module is switched off, no card or driver is needed.

Interpreter object: `ME4000`

13.17.1 Parameter page ME4000

General

ME4000 – Enabled
Activates hardware access to the ME4000 card.

Digital inputs

Execute time measurement with real time priority
If this option is activated then the command ME4000.DIn.Meas of the interpreter is executed with real time priority. The measurements are done with an accuracy in µs. However other real time applications running simultaneously on the computer may receive less computing time from the system.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Page for calibrations data

Button: Enable access to calibrations data
On opening the basic setup initially the access to the calibration data is disabled and hidden to avoid accidental changes. By clicking this button the calibration data is shown on a sub page for edition.

MCD Elektronik GmbH	Date: 2018-04-12	Page 122 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Page for analog outputs

This page consist of the calibration data for the 4 analog outputs and different output ranges. For each channel/ range combination a range and offset value can be programmed. The calculation on the card for the given value is done according to the formula:

Displayed value = (Raw value - Offset) x Gain

Raw value is the value i.e. to appear on the card or the testing unit. Displayed value is the value i.e. sent to the card.

The calculation procedure is described in the ME4000 object of the interpreter help.

Page analog inputs

This page consist of individual pages for the different ranges of the 4 analog inputs. The sub page consists of all range and offset values for all the channels. These can be calibrated by software.

The calculation is done according to the formula:

Output value = (Set value - Offset) x Gain

Thus the output value will be further used in the program to display the result or by the interpreter. Set value is the value read by the card.

The calculation procedure is described in the ME4000 object of the interpreter help.

13.17.2   ME4000 – Monitor

13.17.2.1   Info tab

The info tab gives information about the status of the card and the drivers. It is shown here whether the ME4000 module is enabled and ready.

It is important that the driver version should be minimum 4.02. An error for displaying the curves of the DA converter is present in the older version.

The checkbox 'Cyclic requests' activates a cyclic reading for the tabs which read information from the card. If this checkbox is not selected, then the values change only if the interpreter reads a channel. The read value will also be shown in the monitor.

The cyclic requests do not affect the ME4000 measurements except for the timing (The ME4000 card will be accessed more frequently, so the interpreter action may be delayed). However due to the active samplings no measurements using the AD converter can be made.
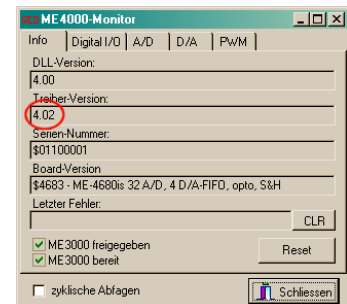
13.17.2.2   Digital IO tab

This tab shows the 16 digital inputs and 16 digital outputs of the ME4000 card.

The inputs are read and displayed in a cyclic manner, if the checkbox is marked. Otherwise the ports are only refreshed and displayed if the interpreter reads the inputs. The channels C-F are system inputs and cannot be used by the interpreter. A yellow circle stands for high signal at the input, a dark circle means low signal.

The outputs will be refreshed when their state changes. A green circle stands for high signal and a dark circle means a low signal. The channels 0-7 can be switched with a double click or by using the space key. The key plus switches to on, the key minus switches to off. The channels 8-F are used from the system and cannot be changed by the interpreter or manually. As the outputs to the UMS (see UMS module) are done using the ME4000 card internally, each action on the serial bus is seen on the digital IO tab. This highly

slows down the digital I/O page of the UMS. This can be important or unimportant according to the problem situation.
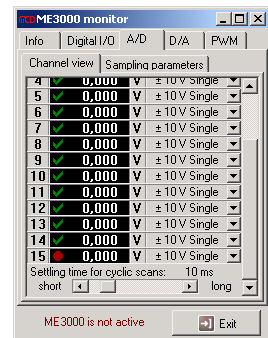
Although the ME4000 card has 4-8bit ports (different than the ME3000 card which has 2-16bit ports) the software considers it internally as 2-16bit ports. The relation of the ME4000 ports to the logical ports of the software is as follows:

Outputs

Port A	#	Output high
A7	15	UMS Address A4
A6	14	UMS Address A3
A5	13	UMS Address A2
A4	12	UMS Address A1
A3	11	UMS Address A0
A2	10	UMS DataLatchEnable DLEN
A1	9	UMS Clock
A0	8	UMS Data

Port C	#	Output low
C7	7	DOut Bit 7 or UMS A5
C6	6	DOut Bit 6
C5	5	DOut Bit 5
C4	4	DOut Bit 4
C3	3	DOut Bit 3
C2	2	DOut Bit 2
C1	1	DOut Bit 1
C0	0	DOut Bit 0

Inputs

Port B	#	Input high
B7	15	Alarm connection
B6	14	Return channel UMS
B5	13	I²C 1 SDA In
B4	12	I²C 2 SDA In
B3	11	DIn Bit B
B2	10	DIn Bit A
B1	9	DIn Bit 9
B0	8	DIn Bit 8

Port D	#	Input low
D7	7	DIn Bit 7
D6	6	DIn Bit 6
D5	5	DIn Bit 5
D4	4	DIn Bit 4
D3	3	DIn Bit 3
D2	2	DIn Bit 2
D1	1	DIn Bit 1
D0	0	DIn Bit 0

13.17.2.3   A/D channel view tab

This tab shows the 16 analog inputs of the ME4000 card.

The channels are read in a cyclic manner and displayed if the checkbox is marked. Otherwise the display is only refreshed if the interpreter reads the respective channel. The last read value and the measuring range is displayed on each row. The measuring range can be set for the individual channels using the buttons on the right with down arrow symbol.

Using the scroll bar at the bottom end of the window the settling time before the cyclic scans can be adjusted. If the settling time is too less and thus the impedance of the source signal is too high, then the measuring channel has very less time to swing at the right value. It can thus lead to a wrong measurement. However the load on the system is reduced when shorter settling time is used. Only those channels marked with green tick symbol cane read back in a cyclic manner. By clicking on this green symbol the activity of the channel can be changed.

If any of the 16 channels is activated for differential measurement, then the corresponding channel above it will be automatically hidden.

MCD Elektronik GmbH	Date: 2018-04-12	Page 124 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

13.17.2.4 A/D Sampling-parameter tab

On this tab the parameter for the sampling action can be set and displayed (Sampling a given number of samples for one or more channels with a given sample frequency).

All parameter of the sampling can be set. Since all the parameters are interrelated with each other thus the field 'Parameter check' displays if a sampling is possible (e.g. the maximum sample frequency depends on the number of channels).

Using the buttons 'Start' and 'Abort' the sampling can be started or a running sampling can be stopped.

The buttons 'Start' is active only when the sampling parameters are valid.

13.17.2.5 D/A channels tab

The tab D/A channels shows the status of the 4 analog output channels. A new value can be entered in the white edit fields and is activated on pressing the 'Set' buttons.

Changing the output ranges is not possible because the ME4000 is only available with fixed ranges. By moving the mouse over the symbol '' to the left below the channel number, the real calibrated output range will be shown.

Attention:

Concerned only for Victor test-system

The Victor test-system possesses after the DA converters of the ME4000 power amplifiers which increase the current carrying capacity of the outputs. The voltage output is also increased by factor 1.5. If the voltage amplification values are set in the calibration factors, then the output voltage of the test system are identical or lower to the amplification factor.

Concerned only for Victor test-system (only for Variant 1)

In systems with a internal regulated linear power supply the fourth DA channel is used for controlling the power supply (only for Variant 1). The voltage supplied by the DA converters is not the actual output voltage of the power supply. Its voltage is amplified depending on the calibration factor of the ME4000. Thus the power supply produces a voltage output equivalent to the program display by calculating the voltage amplification values.
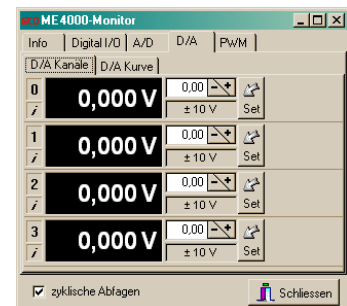
13.17.2.6 D/A curve tab

The ME3000 card can output a arbitrary function on a DA converter. Using this tab the defined curve can be displayed. The active curve (i.e. currently on the output) is shown in red and additionally the frequency is displayed. The frequency depends on the entire curve and not on the individual points.

If the curve is only planned (i.e. constructed only inside the internal buffer) then it is shown in yellow.

A planned curve (yellow) can be activated by the button 'Start' (red curve), which can be stopped by the button 'Stop'. Also in the area 'curve-synthesise' a square, sine, or saw tooth function can be defined and given out. It can then be restarted again.

Using the 'Edit' button a curve editor can be opened. The planned curve is then transferred to this editor.
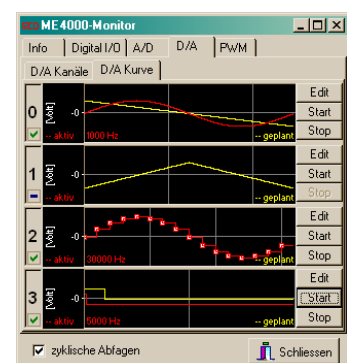
13.17.2.7   Curve editor

In the curve editor standard signal forms can be generated. They are waveforms with square, sine, or saw tooth functions. The required waveform can be selected by 'Curve form'. Parameters will be set under 'Curve parameters' which consist of amplitude (the difference from the highest signal point to the offset voltage (Vss/2), offset voltage, symmetry parameter and the frequency.

For square wave signal output the edit field parameter describes the relation between high and low phase. For saw tooth signals the edit field parameter describes the relation between rise time and fall time. For a sine wave this parameter has no function.

The offset ± amplitude must be within the output range of the chosen channel, otherwise the curves will cut at the borders of the output range.
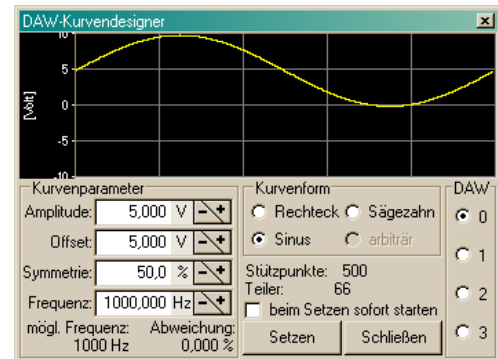
The topics discussed above regarding the displayed and actual voltage values are also applicable here.

Depending on the frequency the curve designer tries to find the number of interpolation points required to construct the selected frequency. The achieved frequency and the deviation in % will be shown.

If the 'set ' button is pressed the planned curve is activated on the selected DAC channel. If the checkbox 'Start on set' is selected then the waveform is immediately started.

13.17.2.8   PWM tab

The PWM Function from the ME4000 card is not yet supported. But it is however possible to create a very precise PWM signal using the ME4000.

13.18   Measure data/ User defined measured data fields

Each test step produces a set of data along with the actual measured value and the step result. Also a set of meta-data is produced for e.g. time of test, duration of the step and data which depends on the type of step for e.g. the number of the step, units etc.

User defined measured data fields help the user to define extra field for each test step i.e. to be filled by the interpreter. These field are available for all the measured data and the number is not limited.

Internally the user defined measured value field are called with a new name. As enumerated data items they have a common identification number and their name is realised as a distinctive feature.

User defined measured data fields ─────────────────────────

Button: new field
Creates a new field

Button: Delete field
Deletes the selected field

Button: Move up
Moves the selected field upwards by one position. The order of fields does not affect the program. Thus related fields can be grouped together.

Button: Move down
Moves the selected field downwards by one position.

The defined field is shown in a grid. The individual columns are:

Name – Identification for the field. It must be distinct and follow the rules for interpreter characters.

Value – is the previous set value for that field (from interpreter or in the basic setup). The value of such a field is generally of type string.

Will be cleared on – shows when the system automatically deletes the field. If undeleted the value is retained till the program end. Possible values:

> 1 – Program start
> 2 – on start of a column test
> 3 – on start of a row test
> 4 – on start of a DUT test
> 5 – on start of each test step.

13.19   Measured data DBF

The saving of the measured values in data bases with DBF (dBase) format is standard way to save data. An installed BDE (Borland data base) engine is required.

The data can be saved in standard format or in freely defined tables.

Due to continuous saving of these measured values the hard disk had usually a space shortage because the tables saved were generally large sized. To avoid this problem a data maintenance was introduced.

Interpreter object: MsdDBF

13.19.1   Parameter page Measured data DBF

General

Measured data DBF – Enabled
Activates the entire DBF module. When the module is deactivated, no BDE is needed.

Destination

Local path

Network path
Defines the two paths where the measured values tables are to be saved. From the technical aspect both the paths are similar i.e. they can lie either on the network or any local path. The basic idea behind it was first to try saving the data on the network. If not possible then save to local.

Save to
Selects the actual path from the above options to save data.

Save to other path, if selected path is not available
Activates the automatic mode to save the data on the optional path when saving in the actual path has failed.

Method

Storage
If 'Standard' is selected the tables for all the types will be created and saved together directly in the selected path. If 'Type oriented' is selected then for each type family a sub directory is created 'Type XXX' i.e. similar to the type data structure and saved in the given path. The tables are then saved in these sub directories.

Save measured data
Normally all the measured values are saved after a complete panel test in order to reduce the load on the network. In exceptional cases (for long test sequences or many DUTs in a panel) it may be logical to save the data after every DUT test.

13.19.1.1 Parameter page Measured data DBF – Data maintenance

The data maintenance performs 2 important functions: The tables are maintained as small as possible because larger tables need more time to access. Secondly it should be avoided that the hard disk is entirely filled with unwanted data.

Data maintenance ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Start when
Decides when the data maintenance should be started. During data maintenance the directories will be searched for the selected criteria and will be the respective tables will be edited for e.g. deleted. It must also be confirmed that the selected option also occurs. Thus the option 'program start' when selected has little meaning if the program is never newly started.

Data maintenance in local path
Selects the directory under 'local path' for data maintenance.

Data maintenance in network path
Selects the directory under 'network path' for data maintenance.

Show info window while data maintenance
Selects whether a info window should be opened during data maintenance.

Files
If only the fixed table format will be used then the option 'only standard files' can be selected. This is faster as compared to the option 'all *.DBF Data' which also considers the file names changed by the interpreter commands.

Data maintenance – Backup ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

Backup analysis the size of the table and moves the large sized tables to a sub directory. It is actually no backup but a move.

Backup of the DBF files time controlled

after __ Hours

after __ Days

without time trigger
Decides whether the table backup should take place independent of its size

Backup of DBF files if they

Are larger than __ kB
Decides whether the table backup should take place depending on the entire size of all the tables.

Backup Path:
Gives the path where the tables should be moved as backup. The path can contain a drive name and a path. The actual measured data directory can begin with '.\' for e.g. : '.\BACKUPS'.

File name:
Selects the pattern to create a new name for the table. The file name can contain the following characters %Y - Year, %M - Month, %D - Tag, %H - Hour, %N - Minute (all with double digits)^* will be replaced by the previous file name. If the same name is used earlier then an index will be attached.

Delete backup records ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

This second part of data maintenance deletes the oldest table as soon as the criteria is fulfilled.

If more than __ kB data exists
If activated, then as soon as the files exceeds this maximum size they will be deleted.

MCD Elektronik GmbH	Date: 2018-04-12	Page 128 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

Older than __ days
If activated, then as soon as the files exceeds this time limit they will be deleted.

13.19.1.2 Parameter page Measured data DBF – Table structure

The format in which the DBF tables should be saved is selected.

Fixed table structure

The fixed table structure are the old fixed formats up to version V1.2.6. They are however faster as compared to the new free definable formats.

Save fixed tables
Activates the entire fixed tables. To actually save the files, the main option of 'Measured values in DBF format' should also be activated.

Save measured results (PASSFAIL.DBF)

Is valid if only the fixed table format is active. It saves the test results in the table PASSFAIL.DBF. For each DUT one row of measured data is used.

Tabular structure of PASSFAIL.DBF

Field name	Format	Meaning
SERIALNR	C24	Serial number of the DUT
SYSTEM	C4	System number of the system on which the test has taken place.
PROCESSID	N6:0	ProcessID from the basic setup
TYPECODE	C16	Type code of the Type variant
TYPEFAMILY	C25	Type family of Type variant
TYPEINFO	C30	Description of the Type variant
STARTNR	N6:0	A distinct start number for the present test
FIXTURE	N6:0	not used
MODULNR	N6:0	Position of the DUT in the panel, order of the numbering can differ from fixture.
TESTMODE	C4	System mode : SBS or AUTO
TESTRESULT	C7	Test result : PASS, FAIL, INVALID or BREAK
DURATION	N20:4	Duration of the test
DATE	C	Day of the test
TIME	C	Time of the test

Save measured values (MEASDATA.DBF)
Is valid if only the fixed table format is active. It saves the test results in the table MEASDATA.DBF. In the table only those measured values of a DUT are saved whose save value is enabled in the parameter table or if the result is beyond the limits.

Tabular structure of MEASDATA.DBF and FAILDATA.DBF

Field name	Format	Meaning
SERIALNR	C24	Serial number of the DUT
SYSTEM	C4	System number of the system, on which the test takes place
PROCESSID	N6:0	ProcessID from the basic setup

MCD Elektronik GmbH	Date: 2018-04-12	Page 129 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

TYPECODE	C16	Type code of the Type variant
TYPEFAMILY	C25	Type family of the Type variant
STARTNR	N6:0	A distinct start number for the present test
FIXTURE	N6:0	not used
MODULNR	N6:0	Position of the DUT in the panel, order of the numbering can differ from fixture.
TESTMODE	C4	System mode : SBS or AUTO
TESTNR	N6:0	Sequence number of the test sequence
STEPNR	N11:0	Number of the test step
STEPART	C20	Name of the Interpreter step
STEPNAME	C30	Name of the test step
VALUE	C30	Measuring value
UNIT	C5	Unit
UPPERLIMIT	N20:4	Upper limit
LOWERLIMIT	N20:4	Lower limit
STEPRESULT	C7	Step result
DURATION	N20:4	Duration of the step
DATE	C10	Day of the test
TIME	C10	Time of the test

Save measured values (FAILDATA.DBF)
Is valid if only the fixed table format is active. It saves the test results in the table FAILDATA.DBF. In the table only those measured values of a DUT are saved whose results are beyond the limits or when the next option is activated.

Save failure data with all values (also good)
Saves all the measured values along with the good tests for failed DUTs in the FAILDATA.DBF.

Format

Decimal separator for measured values
Selects the decimal separator as a decimal or a comma or from the windows setup as selected.

User defined table structures

User defined table structures can be freely defined, with contents also from the Enumerated Data Items. They are slower than the fixed tables.

Save user defined tables structures
Activates to save the user defined tables. To actually save the files, the main option of 'Measured values in DBF format' should also be activated.

Structure file:
Selects the file structure as per the format of the saved table..

Button: Edit table structures
Opens the editor to edit table structures.
--- Documentation is absent ---

MCD Elektronik GmbH	Date: 2018-04-12	Page 130 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

Button: Rescan structure files
Rescans the structure files and shows the interpreted content.

Additionally the interpreted content of the structure files swill be shown. If the user defined tables are not activated, then nothing is displayed.

13.20   Measured data DLL

Additional than saving the data in DBF format the measured values can be transferred to a DLL. This DLL should possess a port for the program to access it. The data will be transferred to the DLL through this port.

The port should be described in its own document.

13.20.1   Parameter page

General

Measdata DLL – Enabled
Activates the access to the DLL

File name
File name of the DLL.

Current state

Gives information about the current status of the measured values DLL.

Button: Configure DLL
This button is only visible when the DLL is active and a routine for configuration is available. This is activated by pressing this button. The further actions depend on the DLL.
If the DLL executes a routine enquiring the status, then the requested results are displayed.

In case the MeasDataDLL generates an error

Save data to %PRG%\MEASDATA\MsdDLL.Dat
If an error occurs in the DLL during data transfer, then the measured values etc. will be saved in the following data path. For formatting see the documentation for DLL.

Create file even if there is no error (for test purposes will be switched off when the program is left)
Saves the data even when no error occurs.

Debugging

Always store the data sent to the DLL in %PRG%\MEASDATA\MsdDLL.Dbg
Saves the data transferred to the DLL additionally to this given location. For formatting see the documentation for DLL. This option is not deactivated at the program end. Caution: the data increases very fast and unlimited!

13.20.2   MeasData DLL Monitor

Using the menu 'Values / MeasData DLL Report' a monitor can be opened which shows the current status of the port.

The first tab shows the status from the program point of view. It is especially shown whether the DLL is active and error free and the data required by the DLL from the program.

The second tab shows the information from the DLL point of view. It corresponds to the information page i.e. also shown in the basic setup.

The third tab shows the Enumerated Data Report.

The fourth tab shows the timing required to transfer and process the required data to the DLL.

The text entered in the edit field 'search' will be searched on pressing the arrow symbol.

13.21   Switch off modules

In the program development cycle the integration of new ports and modules keeps on increasing. Hardly any system requires all the modules.

The non-required modules require space in the basic setup, in the menus or during the program initialisation and also increasing the interpreter check. Thus to avoid this and activate only the modules required, the switching of modules was introduced.

The switching of modules provides something similar like winter sleep for the deactivated modules. These modules are not visible any more in the basic setup, in the menus or as interpreter commands and help. They are visible. They are only visible in the 'Switch off modules' in the basic setup where they can be reactivated. As the module was almost entirely removed from the program, while reactivating a module the program must be restarted. (Similar is applicable while deactivating the module)

13.21.1   Parameter page

A table with all the modules that can be activated is shown here. The first column shows the module names, the second one shows the actual state. The third one shows the future state and permits edition. The fourth and the fifth show dependency and their inter-relation.

Some modules depend on the other modules for e.g. the LIN module depends on the RS232 module. If the interrelated module is deactivated then the depended module is automatically deactivated.

The third column showing the future status has 3 possible states:

☑ active – the module is activated

▬ switched off – the module is switched off

☑ blocked – this module requires another module for its functioning which is blocked or switched off at the moment.

After every change the program must be restarted again to activate the changes done. If the modules are deactivated then the interpreter help and commands are also hidden. If these commands are used in the interpreter, then a syntax error is triggered.

The modules are only identified if a new program version is being used, the older version did not have the module feature. Information to the modules in the older versions are shown in a special area and cannot be edited. Thus the modules cannot be activated here as they are not implemented.

13.22   Passwords

To avoid that a user having high password level for e.g. administrator logs in and then leaves the place without logging out, an automatic mechanism is present to set back the password level after a definite time.

Secondly a low level user having no rights, can with an active administrator password misuse the rights later.

The password itself will not be invalid and however the active user level will set back to a lower level. The administrator can thus immediately log in.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 132 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.22.1 Parameter page

Limited period of validity ————————————————

Here for all the password levels it will be decided, when and whether after how many minutes the password level should be invalid.

13.23 Printer

This module provides access to the printers defined under the windows printer control. The output can be given as printer files, through RS232 or printer ports. In the program 4 printer ports are available with individual numbers which can be accessed by the interpreter.

Interpreter object: `Printer`

13.23.1 Parameter page

The set up of the Printer describes the parameters required for the set up of each individual printer. The parameters which can be configured in the basic set up are listed below.

General ————————————————

Printer enabled:

It activates the access to all the connected printers.

Printer X (X = 1..4) ————————————————

Active

It activates the access to this printer.

Name

Gives the target name for the printer data. The name can be:
- To print on an installed windows printer:
  name should be similar to the printer name in windows control panel for printers and faxes.:
  Example: `HP DesignJet 20PS by HP`
- To print on a standard windows printer:
  Example: `(DEFAULT)`
- To print as a data '(FILE)' before the file name enter (possible path)
  Example: `(FILE)%LOG%\Dateiname.prn`
- To print on a COM-Port: '(FILE)COM' and the number of the COM Port
  Example: `(FILE)COM1`
- To print on a LPT Port '(FILE)LPT' and the number of the LPT Ports
  Example: `(FILE)LPT1`

Alternative timeout

Permits to change the standard timeout to access the COM and LPT ports. Normally this timeout is 5 minutes and it is waited after every printed line. As smallest value 5 seconds can be entered. Some versions of windows ignore smaller values and wait for 5 minutes.

13.24 Serial ports

The serial ports are used as a medium for communication between all types of RS232 ports. It is independent of the card, hardware or the PC i.e. used. When the COM port is detected by the Windows system administration then it can be used by the program.

MCD Elektronik GmbH	Date: 2018-04-12	Page 133 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Title: MCD TestManager CE - Software Manual			

Additionally it provides ports for communications with other modules (for e.g. LIN, KW2000). It uses the port parameters for its administration.

Interpreter object: `RS232`

13.24.1   Parameter page

The parameters consist of a main page for all the ports and sub pages for each port.

General

Serial ports-enabled:

It activates the entire access to all the serial ports.

Overview

Shows the status of the currently configured ports. There can be multiple ports shown in green which share a physical com port. However it must be confirmed that at any instance only one of the COM ports is opened (using interpreter command). Thus only one of these ports should possess the attribute 'Autoopen'. If 2 of the same ports are active and auto-open simultaneously, then they are shown in red in the overview.

Button: Create new port

A new serial port can be created.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Monitor

File Masks

Provides a data mask, to be used in the monitor for the data send dialog. The format is:

<Filter>|<description>;

Editor

The type of editor like notepad, WordPad etc. required to edit the text files can be selected here.

Hide disabled ports

Hides the unused part in the tool monitor for more clarity. Another alternative is to delete the unused ports.

13.24.2   Parameter page of Individual serial ports

Com Port

Name

A name can be given to the COM port in this edit field. The new port will always be addressed by the interpreter using this name and thus must possess only interpreter characters. For internally used serial ports the names are fixed and cannot be edited.

Comment

A comment for the port above can be entered in this field. It is ignored by the program.

Active
Activates the port.

MCD Elektronik GmbH	Date: 2018-04-12	Page 134 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Port number

Number of the windows COM port, i.e. related with opening of the port.

Open automatically

If activated then the port is automatically opened. Caution should be taken that for each physical port only one logical port is opened.

Data Format

Baud rate

It determines the speed of the data transfer. It can also contain odd values. The values which can be realised however depends on the hardware.

Data bits

Number of data bits in each block.

Parity

Parity of data transfer. While receiving however the parity check is not performed.

Stopbits

Number of stop bits after each block. Can be used for slow communication for e.g. micro controllers. To send data slowly to the controller so that it has enough time for processing it.

Receive parameter

Timeout

Time limit to receive a reply.

End-Of-String

Character set at the end of a string to detect the end. It is only used by a High-Level-Routine READ. As any character can be entered here (even control characters), a character set using HEX bytes is used foe e.g. '0D 0A' stand for Carriage Return + Line Feed.

Remove end-Of-String

Decides whether while receiving a string a set of know characters should be separated.

Send parameter

Is used for High-Level-Routine SEND.

Start-Of-String

Attaches this character set at the start of the string before it is sent. It should be entered as HEX bytes.

End-Of-String

Attaches this character set at the end of the string before it is sent. It should be entered as HEX bytes.

Hardware-Handshake

Transmission of data

CTS controlled (send only if CTS is active)

When activated the data is sent only if the CTS connection is active.

DSR controlled (send only if DSR is active)

When activated the data is sent only if the DSR connection is active.

Reception of data

RTS activated

Status of the RTS connection while opening the port

DTR activated

Status of the DTR connection while opening the port

Software-Handshake (XOn/XOff) ─────────

XOn/XOff protocol used for transmission
When activated uses the RS232 drivers XOn/XOff protocol for sending. The driver stops sending data when it receives an XOff and starts sending data when it receives a XOn. While sending however the XOn/XOff protocol is not used.

Character for XOn
The ASCII Code, to be used as XOn Character.

Character for XOff
The ASCII Code, to be used as XOff Character.

Only for display in monitor, at automatic receive: ─────────

If automatic receive is activated then the monitor waits for the End-Of-String before it shows the character.

Display after at most __ characters also without EOS
It shows the character also without EOS, when the given number of characters are available.

Display after at most __ ms, also without EOS
It shows the character also without EOS, when the last characters has crossed the given time limit.

Display format
The initial display format for the monitor.

Show vertical lines
Show vertical lines in monitor, which is helpful to compare the binary data.

Show time
Show a time-stamp before the data in the tool monitor.

Button: Copy port
Creates a new port similar to the current port (inapplicable for internal ports)

Button: Delete port
Deletes the current port (inapplicable for internal ports)

13.24.3 RS232 Monitor

The monitor shows a window for each serial port. According to the settings in the basic setup the inactive ports are hidden. At the top of each individual port window the current status and handshake status of the port is shown. The handshake status of RTS and DTR, and the port status opened or closed can be activated using the check boxes above. Below it the current status events of each port are shown. The list can contain 2500 lines of history and thus can be scrolled back.

The contents of the output window can be deleted using Ctrl-E shortcut. By using the mouse and the shift key the contents can be selected. It can be then copied to the clipboard using Ctrl-C. The whole area can be marked using Ctrl-A. These entire actions are also available using the context menu.

The lower part of the window contains a grid where the frequently used messages are saved. They can be sent with a click on 'Send' in the first column. Also there is an option in the third column 'Form' for sending the

commands either in ASCII code or HEX code. In the last column 'Info' description for the commands can be written. Additionally one can enter a new message in the edit field. By pressing the button 'Transmit' the message is sent. The message in the edit field can be saved in the grid using the plus button. The minus button deletes the actual row from the grid.

Using the AddFile option any text file to be sent can be added to the grid. By clicking on the first column the file is sent serially and by clicking on the third column the text file is opened.

Normally the monitor shows data only when it is requested from the input buffer by the interpreter. If no interpreter is active then the data is displayed in the monitor by activating the check boxes 'automatic receive'. The monitor accesses the input buffer now and the data is thus unavailable to the interpreter. Thus for normal program testing please deactivate 'automatic receive'.

13.25   Serial IO bus (SerIO)

The serial IO Bus system uses a serial port and communicates with more modules on one or more SerIO busses.

Some of the members belonging to this module are the old MechIO Module, the new 4fach Current measure module (IMeasX4) and the AdapterCPU (AdapCPU). Many of these modules also in combinations can be added to the same port provided each of it has a unique address. The port and the module thus build a bus. The new class 'Unknown' helps to integrate new unknown modules which were not yet supported by the Program monitor.

The program can contain many of these buses.

All modules on the same bus must have the same baud rate!

The parameters of the serial port used by the serial IO Bus system will be fixed in the basic setup of the module serial ports.

The modules will be called using their module name and module number. The module number is always related only to the module class. Thus there can be MechIO module with module number 1and there can also be an AdapCPU with module number 1. The module number are created due to a sequential definition. The module names can be freely created.

13.25.1   Parameter page Serial IO bus

Overview

Currently configured Serial IO buses and modules
Shows an overview of the bus and modules defined on this bus. Red texts point out to errors (caused mostly due to address or name conflicts).

Allow fundamental changes (will be realised with next reset)
It permits changes in the fundamental parameters of the SerIO Buses. These changes will be realised with the next reset.

Number of SerIO-Bus
Changes the number of SerIO buses. For very bus a serial port will be created in the module serial ports.

Naming system
During the initial development of SerIO modules only the Mechanic Module was available, thus the serial ports were named as 'MechIO, MechIO2,...'. As the new SerIO modules came into picture the name was changed to 'SerIO, SerIO2, ...'. Thus for the new systems the second option should be selected.

Log communication in log file
Allows to save the communication in log file (all the sent queries and replies) %LOG%\SERIO.LOG.

The set up of the COM ports is done in the module serial ports. The following parameters should be properly checked and set:

MCD Elektronik GmbH	Date: 2018-04-12	Page 137 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

Baud rate : depending on the module
Data bits : 8
Parity : none
Stop bits : 1
Receive –Timeout : 200
Receive -End-Of-String : 0D
Receive -... separate : on
Send -Start-Of-String : 12
Send -End-Of-String : 0D
Handshake connections : both off

13.26   SerIO – AdapterCPU

The AdapterCPU allows the simple realisation of an I²C port direct into the test adapter using RS232 communication.

13.26.1   Parameter page AdapterCPU

Common

Adapter CPU – Enabled
Activates the entire AdapterCPU Module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset
If this checkbox is active (default) a firmware request is sent to the module. In case of an AdapterCPU any response is accepted. An error is triggered if no response is received. If the checkbox is deactivate the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Button: Delete module
Deletes that module and the module number of the last one is reduced by one.

MCD Elektronik GmbH	Date: 2018-04-12	Page 138 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Button: up
Moves the module upwards by one position. Thus it exchanges the module number with the module above.

Button: down
Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

13.27   SerIO - IMeas x 4 I/O

The IMeas x 4 I/O module consists of 4 current measuring channels and one voltage measuring channel.

Interpreter object: `IMeasX4`

13.27.1   Parameter page IMeas x 4 I/O

Common ─────────────────────────

Adapter CPU – Enabled
Activates the entire IMeas x 4 I/O module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module

Item in tool menu ─────────────────

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset
If this checkbox is active (default) a firmware request is sent to the module. The response must consist the text 'IMeas x 4'. An error is triggered if a false or no response is received. If the checkbox is deactivated the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Button: Delete module
Deletes that module and the module number of the last one is reduced by one.

Button: up
Moves the module upwards by one position. Thus it exchanges the module number with the module above.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 139 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Button: down

Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

13.27.2 Monitor IMeas x 4 IO

The monitor shows a separate tab for each of the defined (and not hidden) IMeasX4 module. If their presence is not checked then that module is also hidden in the monitor.

The 4 current measuring channels and one voltage channel show their previously measured values. By clicking at the arrow symbol at the end of each row the ranges can be changed.

Below the measured values the current status of the communication is shown.
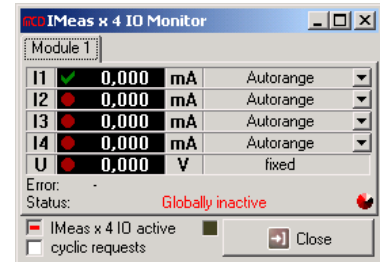
On activating the check box below all the channels marked with a green tick mark symbol is regularly updated. Thus the display is not only updated by interpreter requests. The green tick mark can be deactivated by clicking on it.

13.28 SerIO – Mechanics IO

The mechanical IO allows the access an external input/ output module, which can be controlled with a RS232 connection. This module has 16 digital inputs, 16 digital outputs and 4 analog inputs or 8 digital input/ output.

Interpreter object: MECHIO

13.28.1 Parameter page Mechanics IO

Common

Mechanics IO – Enabled
Activates the entire Mechanics IO module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset

If this checkbox is active (default) a firmware request is sent to the module. The response must consist of the following order for an 8 Bit module ‚I/O-8' and for 16 Bit module ‚I/O-16'. An error is triggered if a false or no response is received. If the checkbox is deactivate the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Trigger level

Decides for the trigger level for the digital inputs.

Kind of module

Selects the type of module (8 or 16 bit).

Button: Delete module

Deletes that module and the module number of the last one is reduced by one.

Button: up

Moves the module upwards by one position. Thus it exchanges the module number with the module above.

Button: down

Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 141 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.28.2   Mechanics IO monitor

The monitor shows a separate tab for each of the defined (and not hidden) Mechanics IO module. If their presence is not checked then that module is also hidden in the monitor.

If a name is given to the module, it is shown as title of the monitor.

The monitor displays the current state of the digital inputs, outputs and the 4 analog inputs. The inputs are read in cyclic manner and displayed if the checkbox 'Cyclic requests' is marked or when requested by the interpreter.

The analog inputs function similar like the digital inputs.

The names of the channels are read from the path '%SYS%\MechIO.Dat'.
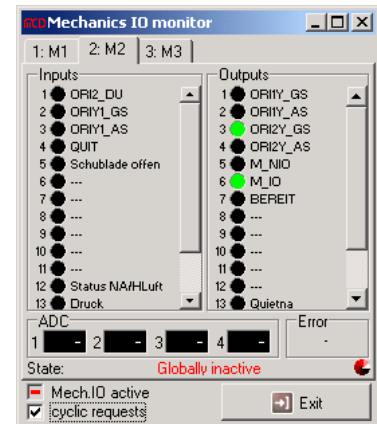
13.29   SerIO – RCMeas

The RCMeas module provides a measuring channel for measuring components. It is possible to measure resistances, capacitors, diodes and voltages using it.

Interpreter object: `RCMeas`

13.29.1   Parameter page RCMeas

Common ──────────────────────────

RCMeas – Enabled
Activates the entire RCMeas module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module

Item in tool menu ──────────────────

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset
If this checkbox is active (default) a firmware request is sent to the module. The response must consist the text 'RCMESS'. An error is triggered if a false or no response is received. If the checkbox is deactivate the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Button: Delete module
Deletes that module and the module number of the last one is reduced by one.

Button: up
Moves the module upwards by one position. Thus it exchanges the module number with the module above.

Button: down
Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

13.29.2   RCMeas Monitor

The monitor shows a separate tab for each of the defined (and not hidden) RCMeas module. If their presence is not checked then that module is also hidden in the monitor.

If a name is given to the module, it is shown as title of the monitor.

The required range and measuring quantity can be selected using the combo box. The measuring parameters can be set using the edit fields.

The monitor displays the current status in a cyclic manner if the checkbox 'Cyclic requests' is marked or when requested by the interpreter.

In the middle the status of the communication is shown
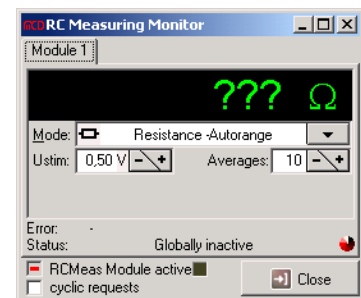
13.30   SerIO – Unknown

The unknown module class allows to define modules, which were not yet supported by the program with monitors or interpreter objects.

Interpreter object: `SerIO`

13.30.1   Parameter page Unknown

Common

SerIO Unknown – Enabled
Activates the entire SerIO Unknown module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset
If this checkbox is active (default) a firmware request is sent to the module. The response must consist the text as stated below. An error is triggered if a false or no response is received. If the checkbox is deactivate the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Answer of the version request contains
If the earlier checkbox is selected then the text i.e. to be checked for the firmware request should be entered in this edit field. It is independent of the character size. Thus a verification of the correct module can be done.

Button: Delete module
Deletes that module and the module number of the last one is reduced by one.

Button: up
Moves the module upwards by one position. Thus it exchanges the module number with the module above.

Button: down
Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

13.31   SerIO – FreqIO

The FreqIO module provides a frequency counter with 4 input channels.. Along with the frequency of a signal also its duration between high and low phase can be measured.

Interpreter object: `FreqIO`

13.31.1   Parameter page FreqIO

Common ───────────────

FreqIO – Enabled
Activates the entire FreqIO module.

Automatic retries on communication problem
If activated repeats the program communication error of that command automatically twice.

Button: Create new module
Creates a new module

Item in tool menu ───────────────

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sup page for each module

Module name
Creates a name for the module. The module can be later called using this name or the module number (which is sequentially achieved). The name must be unique (different to all the modules types). It is independent of the character size and must possess valid interpreter characters (first alphabet, then alphabet, numbers or underscore).

Module – Enabled
Activates the module.

on SerIO-Port
Selects the SerIO-Port on which the module will be connected.

Address
Address on the selected SerIO-Port

Check presence of module during reset
If this checkbox is active (default) a firmware request is sent to the module. The response must consist the text 'COUNTER_1'. An error is triggered if a false or no response is received. If the checkbox is deactivate the presence of the module is not checked. Thus it is not confirmed whether the module is connected to the SerIO bus.

Button: Delete module
Deletes that module and the module number of the last one is reduced by one.

Button: up
Moves the module upwards by one position. Thus it exchanges the module number with the module above.

Button: down
Moves the entry downwards by one position. Thus it exchanges the module number with the module below.

13.31.2   FreqIO Monitor

The monitor shows a separate tab for each of the defined (and not hidden) FreqIO module. If their presence is not checked then that module is also hidden in the monitor.

If a name is given to the module, it is shown as title of the monitor.

The required channel can be selected on the left side. The function and the measuring parameters can be set using the edit fields.

The monitor displays the current status in a cyclic manner when the channel is selected and the 'measure' button is pressed or when requested by the interpreter.

In the middle the status of the communication is shown.
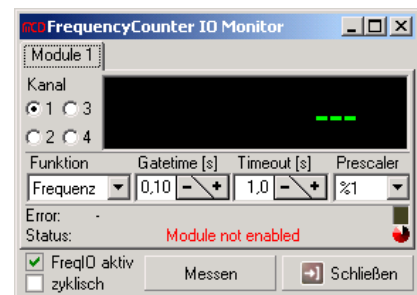
13.32   SM2044 Multimeter

The SM2044 module manages the SM2044 Multimeter card from the manufacturer Signametrics.

Please observe that initialisation of the card during reset last for long time (up to 20 seconds). This is correct and not an abnormal behaviour of the program.

Interpreter object: `SM2044`

13.32.1   Parameter page of SM2044

Common ────────────────

SM2044 – Enabled
Allows the entire access to the card. To access the card the card and all the required drivers should be properly installed. If the card is deactivated no card or drivers will be required.

Calibration data
The SM2044 manages for its own calibration and nothing is done by the program. Before starting the card driver it should be informed about the location of the Signametrics calibration values.

Item in tool menu ────────────────

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 145 of 181	**MCD** ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

13.32.2   SM2044 Monitor

The SM2044 monitor allows the selection of the measurement rate and range, starting of a measurement or cyclic measurements. By clicking the down arrow symbol near the options 'Function', 'Range' and 'Measrate' their respective parameters can be set.

If 'Autorange' is active then the card automatically selects the range. Not all combinations of 'Function', 'Range' and 'Measrate' are possible. Manual measurement is only possible if the cyclic measurement is deactivated. The cyclic measurements are possible only when the measurement rate is greater than 10 measurements per second.

13.33   Sockets (TCP/IP)

The socket module provides TCP/IP sockets, using which communication can be made with TCP/IP sockets of other programs.

Till Build 984 it was possible to implement client sockets only, which communicated with server sockets of other programs. But from Build 985 onwards server sockets have been implemented in the program itself, which can communicate with client sockets of other programs.

Interpreter object: `Socket`

13.33.1   Parameter page TCP/IP sockets

The Parameters consists of a general page for all sockets and sub page for every newly defined socket.

General

Sockets – Enabled
Activates all the sockets.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Button: Create new socket
Creates a new socket.

Sub page for induvidual sockets

Socket

Name
It is the name of the socket i.e. used by the interpreter and thus must possess valid interpreter characters.

Comments
It only describes the socket and is ignored by the program.

Active
Activates the socket.

Open automatically
If activated then the socket is automatically opened after reset. An open server socket can receive connections, while an open client sockets connects with the server on activation.

Port
TCP/IP port, for the client socket to access which opens the server socket.

Kind of socket
Selects the type of socket:

Client socket: It is a socket which communicates with server sockets of other program.

Server socket single: Opens a port i.e. available for client socket of other program for communication. The data from all the connected client sockets enter a common input buffer. While the data sent will be sent to all the client sockets.

Server socket multiple: Opens a port i.e. available for client sockets of other program for communication. The data from all the connected client sockets enter a separate input buffer. While the data sent will be sent to the client sockets individually. (These sockets are not yet implemented)

Client socket parameter

Host
Give the name of the host PC where the server socket is present. Either as TCP/IP Address (127.0.0.1) or as PC name (\\Tester0815) or if a DNS is present as URL (www.server.de)

Open-Timeout
Time after which the port should be opened.

Asynchronous connect
As per the server or connection the asynchronous connection can be advantageous over the synchronous connection. Thus it is advisable to try.

Client socket parameter (both types)

Maximum number of clients
Gives the number of client sockets i.e. to be connected to the server socket. Other sockets will be denied access. Due to technical problems a short connection occurs. This is however stopped later.

Receive parameter

Timeout
Time limit to execute a read data instruction.

End-Of-String
Character set at the end of a string to detect the end. It is only used by a High-Level-Routine READ. As any character can be entered here (even control characters), a character set using HEX bytes is used foe e.g. '0D 0A' stand for Carriage Return + Line Feed.

Append End-Of-String
Attaches this character set at the end of the string before it is sent. It should be entered as HEX bytes.

Delete End-Of-String
Decides whether while receiving a string a set of know characters should be separated.

Receive buffer
Size of the receive buffer in bytes.

Send parameter

Start-Of-String
Attaches this character set at the start of the string before it is sent. It should be entered as HEX bytes.

End-Of-String
Attaches this character set at the end of the string before it is sent. It should be entered as HEX bytes.

Button: Copy port
Creates a new port similar to the current port.

Button: Delete port
Deletes the current port.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 147 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.33.2   Socket Monitor

The socket monitor functions similar to the RS232 monitor.

13.34   SQL Database

The SQL module allows access to all the databases, which provide an ODBC port. This includes all the database formats of the BDE.

At the moment only a single database can be accessed. It is further planned to open multiple databases also.

The SQL module needs an installed BDE.

Interpreter object: `SQL`

13.34.1   Parameter page SQL database

General

SQL access to database – Enabled
Activates all the databases.

Item in tool menu

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

Sub page for each database

Database x

active
Activates the access to the current database.

Auto connect
Opens an active database automatically after reset.

Location of database

Access mode
The databank can be accessed in two ways:
- Use alias: This should be placed as DSN in the ODBC control panel. The alias then provides the important parameters like location of database, driver, access parameter etc.
- Use driver and parameter: This consist of the required additional information.

Alias or driver
Name of the alias driver depending on the type of access mode selected.

Parameter
Additional parameters to access the database.

Button: Use dialog to setup database
As entering the location and parameter is not trivial, a dialog window can be opened to enter the data easily. On opening the window the parameters from the basic setup are used.
If 'alias' or 'driver' is selected, then the system searches them and shows the found values in the edit field of the monitor.

The required parameter can be searched for in the list.

On pressing connect a connection is attempted to the database. If any tables are found then they are shown in the list nearby.

MCD Elektronik GmbH	Date: 2018-04-12	Page 148 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

The button 'Default-parameter' copies the default parameters of the selected database in the list nearby.

'OK' button accepts the changes made and enters it in the basic setup.

The easiest way for a database connection for an ODBC data administration is to arrange an alias.

13.34.2   SQL Monitor

The SQL monitor shows a separate tab for each database (At the moment only a single database can be accessed, but it is further planned to open multiple databases also)

Using the check box the connection status can be changed. The database itself can be changed using the button '...'. But this change is however not permanent (it is not saved in the basic setup). If the database is connected and the tables are shown, then the required SQL commands can be written in the edit field above and executed by pressing 'execute'. If the command has a result then it will be highlighted in the table below. The result is also shown below the grid field.
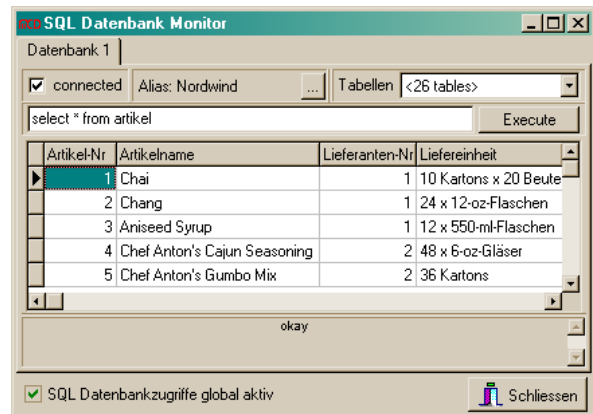
13.35   UMS

UMS allows the access to the cards of MCD Universal measuring system.

The hardware configuration for these card must be defined in an file externally (see chapter 16.5).

Interpreter object: UMS

13.35.1   Parameter page

The module can only be selected in the basic setup for, whether and where the tool monitor should be shown in the tools menu.

13.35.2   UMS Monitor

The monitor shows a window for each of the installed cards, as a tab in the card format and as a page in the tree format. By few cards the overview in card structure is advisable, as with many cards a lot of space is occupied by the tabs above. Thus for many cards the tree structure is advisable.
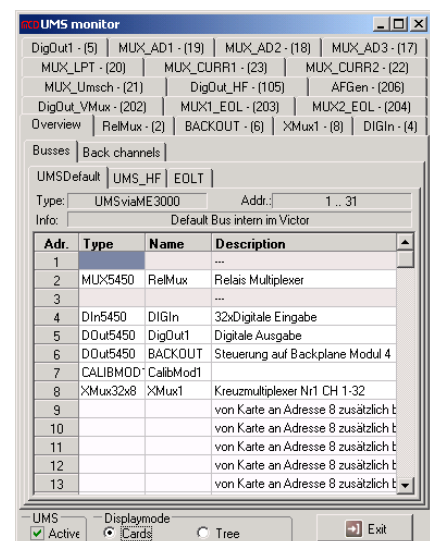
13.35.2.1   Overview

The overview has 2 sub pages, one for the busses and the other for the back channels.

The page for busses in turn shows individual tabs for each bus showing the important definitions and bus configurations of the cards.

The overview in the tree structure is unclear, because here the individual components of each bus are separately shown.

Changing the configurations is not possible here.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany	Date: 2018-04-12	Page 149 of 181	
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual
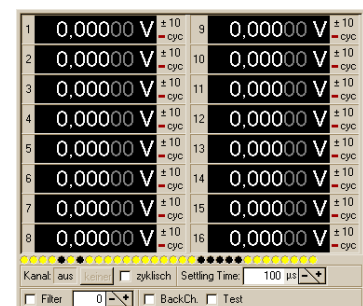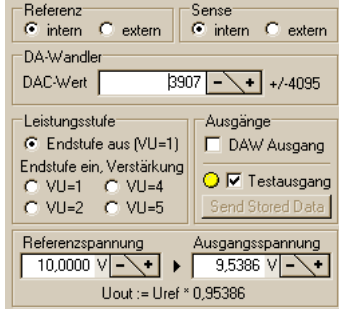
13.35.2.2 ADC16B

This page shows the previously made 16 channels. The values are updated only when they are accessed by the interpreter. If the check box 'cyclic' is activated then all those channels having green tick symbol before 'cyc' will be read in cyclic manner. Above 'cyc' the range for that channel can be selected. By just clicking on the range or by right clicking anywhere on the channel area a context menu to select the range and the activity of the channel can be called.

The yellow buttons below is present only for debug. This will be removed in the later program versions.

The check box 'test' activates a fixed voltage on the AD converter. Thus a self test of the input channels of the AD converter can be done.

13.35.2.3 DAC5450

It shows the status of the DA converter card. The individual settings can be controlled and manipulated. If the checkbox 'test output' is activated, then the current status of the digital comparator can be cyclic checked and is also displayed to the left. Thus the card function can be self tested. The comparator output is high with 9.5V.

The output voltage is only a theoretical value based on the selected reference voltage depending on the DAC value and the calculated final stage value.
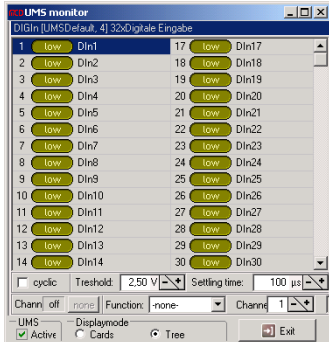
13.35.2.4 DIN5450

Shows the status of the 32 input channels of the card. The values are updated only when they are accessed by the interpreter. It is also updated if any action is done on the channel display or the checkbox 'cyclic' is activated.

The measured parameters can be influenced using the different settings on the toolbar below. If the checkbox 'test' is activated, then instead of the multiplexer output a fixed voltage 2.5V is activated on the comparator. Thus the comparator function and the input channels can be self tested.

If a function is selected then that measurement is performed cyclic on that channel. Its result is then shown on the right hand side of that channel.

If in the address files additional names are defined for each channel this address is used as a description for the channel in the monitor.
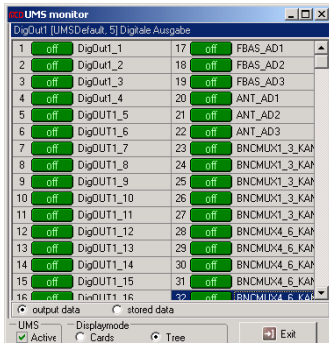
13.35.2.5 DOUT5450

Shows the status of the 32 output channels of the card. By clicking on the output status symbol or by double clicking in the channel area its status can be changed.

Using the radio buttons below the output for the previous status (output data) or the stored status i.e. not yet on the output (stored data) can be shown.

If in the address files additional names are defined for each channel this address is used as a description for the channel in the monitor.
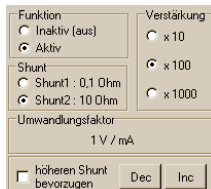
13.35.2.6 IUCONV

Shows the actual settings of the current/ voltage converter card.

Hint: 'Inactive (Off)' means that the current path through the shunt is completely closed and not that the shunt is just bypassed. The bypassing of the shunt i.e. of the entire current/ voltage converter must be done within the external circuitry.

Using the checkbox 'prefer higher shunt' it is decided by clicking of 'Inc' or 'Dec' ac

combination of higher or lower shunt should be preferred. If both the combination of shunts and PGA are available then the next higher or lower conversion factor is selected.

13.35.2.7   MUX5450

Shows the status of the 16 relays. By clicking on the relay symbol its state can be toggled.

Using the radio buttons below the output for the previous status (output data) or the stored status i.e. not yet on the output (stored data) can be shown.

If in the address files additional names are defined for each channel this address is used as a description for the channel in the monitor.

Relays supposed to be inactive in the address files are shown crossed in grey colour and thus cannot be accessed anymore.
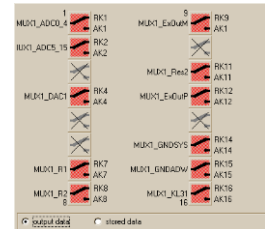
13.35.2.8   NfGEN16

Shows the current settings of the Audio-sinus-generator card

The reference frequency and reference voltage presently are fixed at 4,194304 MHz and 5 V and cannot be edited.

The further processing of the parameter is performed based on this reference.

13.35.2.9   RCMUX48X2

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed. The central part with white and black dots is present only for debug purposes. This will be removed in the later program versions.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.
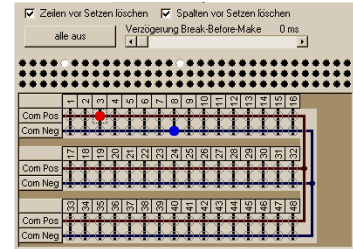
13.35.2.10   RCMUX192X2

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed. The central part with white and black dots is present only for debug purposes. This will be removed in the later program versions.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.
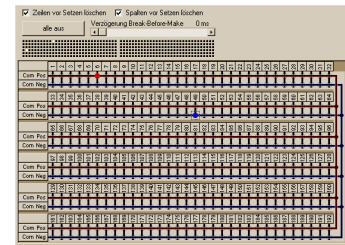
13.35.2.11   RNET2X15

This tab shows 2 identical areas for both the channels. The currently realised channel, each activated resistance R(on) and the status of the output relays is displayed here.

The resistors are activated when the 'Output on' check box is selected.

 The resistors can be changed in 3 different ways:

- Set new value in the edit field 'new' and press button 'set'.
- Double or halve the values using the buttons 'x2' and '1/2'.
- Increment the value using the progress bar and then do minute edition using the buttons '+' and '-'.

The output value will however be rounded to the next possible value.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.12   XMU X32X4

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed. The central part with white and black dots is present only for debug purposes. This will be removed in the later program versions.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent

short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.13   XMUX32X8

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.14   XMUX32X12

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.15   XMUX32X16

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.16   XMUX64X4

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column

connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.

The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.17   XMUX64X8

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed. The central part with white and black dots is present only for debug purposes. This will be removed in the later program versions.

The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.
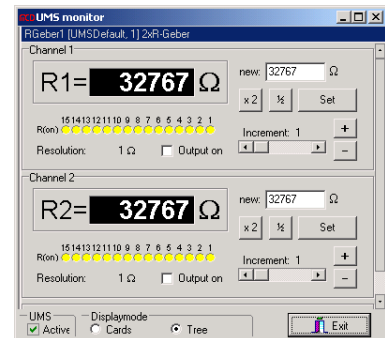
The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.35.2.18   XMUX96X8

Shows the status of the switching matrix. By clicking on any of the crossing point in tool monitor the status can be changed. For giving an entire overview of the whole matrix (even without a 19" Monitor with 1600x1200 pixel resolution) a small view of the entire matrix is shown below.
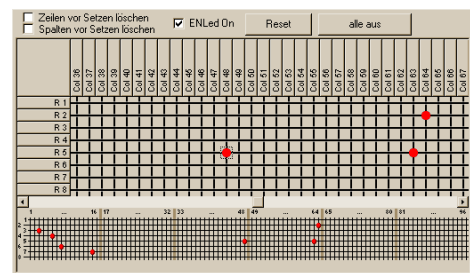
The options at the top help to manipulate the monitor and is similar to that of the interpreter commands. The two check boxes 'Clear rows before connecting' and boxes 'Clear columns before connecting' helps to prevent short circuits. The short circuit can occur by a wrong connection due to any previously made row or column connection. The delay time set by the progress bar 'Break-before-make delay' waits for the set time before activating the relays. The button 'Clear all' deactivates all connections.
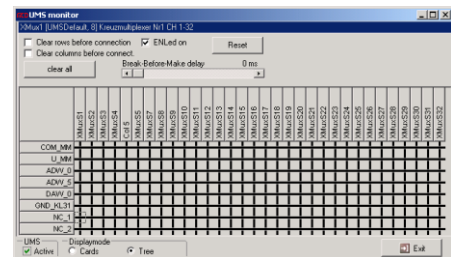
The check box 'ENLed on' activates the Enable LED of the card i.e. important for debug purposes. The button 'Reset' sets back the error logic of the card.

13.36   Upload / Download

The Upload / Download module assures the presence of the same test program files on multiple testing units. Thus all the testing units compare their test program files with a single master program folder.

Upload stand for coping the test program files to the master program folder. The upload can be done only manually and it requires editing rights of the master folder.

If the automatic download is activate, all the changes made in the current program without uploading will get overwritten as soon the download occurs from the master folder. This may be wished but care should be takes as it may lead to data loss if the changes done were important. For this reason when the automatic download is activate, a window is opened as soon as an editor (Type list, test sequence or interpreter) is opened with the following possibilities:

* Stop automatic download – The currently made changes is not damaged by downloading, but this program in not synchronised with the master folder any more. The automatic download must again be activated in the basic setup if required.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 155 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

- Continue automatic download - The currently made changes is damaged by downloading, if the program is restarted without uploading the currently made changes. By restarting the program next time the current program is synchronised with the master folder.

The above editor query appears only for the first time when the editor is called.

13.36.1 Parameter page Upload/ Download

General

Upload / Download -enabled
It activate the entire access to Upload/ Download.

Master path
Here the path of the main server can be entered where the master program folder can be accessed. The path can be on the local PC or any other server PC.

Following files will be ignored at Upload/ Download

Automatically:

User defined:
Exchanging all the files with the master path doesn't make sense for e.g. files of the local statistics, backup data. These files which are to be ignored during the Upload / Download process can be entered in the user defined options separated by a semicolon (the field automatic is shown as an example here). The file names can posses the usual wild cards.

Upload

The upload function can only be chosen from the menu only with higher password levels. Thus it is not required to be parameterised.

Download

Modus
Decides whether the download should be automatically done during the starting of the program.

Show warning message if download is disabled
This displays a warning message if the downloading is disabled. Thus it reminds the user to reactivate the automatic download if at all it has been deactivated during previous editor changes.

Show info window during downloads
If activated an information window is opened during an upload/ download process.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 156 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

13.36.2   Upload / Download Monitor
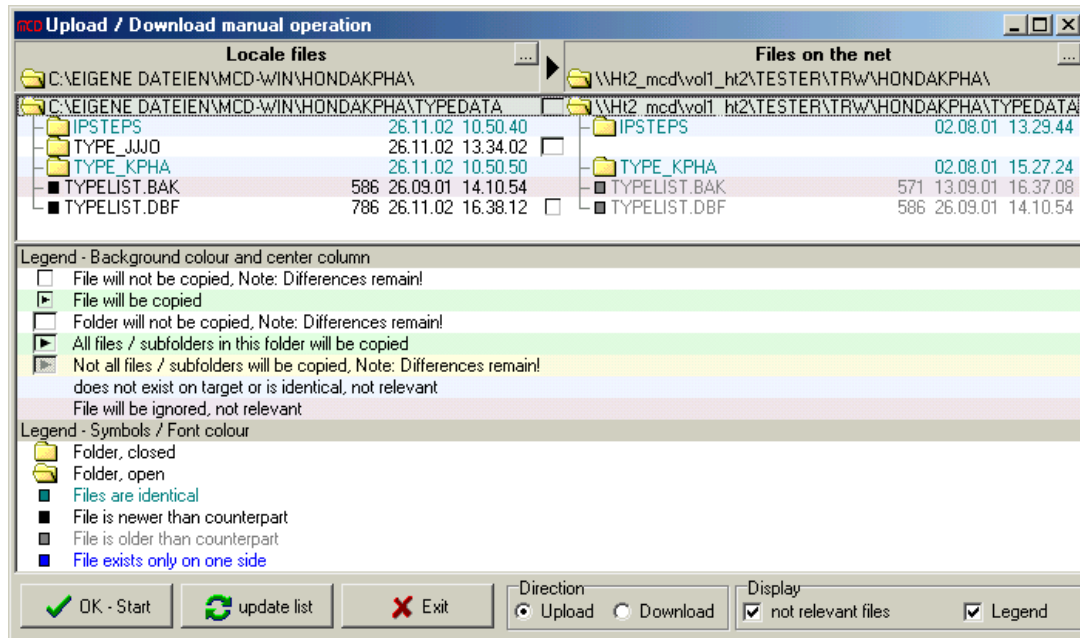
The Upload / Download monitor can be accessed from the menu edit in the main window.

In the topmost part the contents of the local program and the master server of the MCD program are displayed. This page is quite similar to the tree structure of folders in windows explorer. If the check box in display 'not relevant files' is selected then those files ignored by upload/ download are also shown. . If the check box 'legends' is selected then information to the different legends is provided below.

The symbols near the file names shows the direction of coping (see legends). By clicking on these arrow symbols they are reversed. Directories are opened on clicking.

The direction for uploading and downloading of files can be selected by the radio buttons in 'direction'. The direction of the arrow in the check box indicates the direction of the file transfer i.e. from local program and the master server. The coping process is started by pressing the button 'OK-Start'. If the files have been changed externally the directories should be reread again.

A click on the buttons '...' at the top allows to freely select the data i.e. to be compared. Thus it also possible compare two different directories for their contents.

13.37   Virtual Keyboard

The virtual keyboard is a keyboard window on the screen which simulates an actual keyboard. It can always be used for entering data only within the program text fields.

13.37.1   Parameter page virtual keyboard

General

Show in
Selects whether and where the tool monitor should be shown in the tools menu.

14   Reports (Print-out)

To adjust the print outs as per user requirement, the program is supported with an efficient report generator. The overview of the print outs is not fixed and can be edited by the user with help of the report editor. The report musters is saved as sub directories in %SYS%Reports separately according to the report type.

If a report is to be printed then the available musters are shown to the left, and details of the selected muster is shown to the right.

Some of the report musters shown in pointed brackets are integrated in the program and cannot be edited by the user. They can be however edited and saved with a new name.

The button 'Preview' generates a preview of the selected muster. It is possible to print a copy direct from the preview. By pressing the button 'Print' it can be printed excluding the preview. The printer options can be selected directly before printing.

The button 'Edit' opens the report editor and allows the user to define a new muster.

The report muster editor possesses its own documentation.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 158 of 181	**MCD** ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

15 Actions

A action is a function, provided by a program module which can be called using an identification code. Some actions allow or expect parameters before execution.

A action is only possible when that respective module is present in the program and is activated in the basic setup.

Actions can be interrelated with user defined elements of the program overview (e.g. Menu entry, Buttons) and can be called using the interpreter command System.Action.Trigger.

Some actions are executed only after execution of the test step and not all of them are useful from the interpreter. For some special inetreter commands are present.

15.1 Action codes for menu functions

Code	Description
0001	Open type selection window
0002	Show measured values from previous test
0003	Open the local statistics
0004	Open the registration window
0005	Open the basic setup
0006	Open the type and variant editor
0007	Open the test sequence editor
0008	Open the interpreter editor
0009	Switch between SBS and Automatic mode
0010	DUT activity dialog
0011	Start test
0012	Password dialog
0013	Language select dialog
0014	Program reset
0015	Program end
0016	Shutdown Windows - for parameter see Specials
0018	Manage access levels
0019	License information
0020	Start measdata dbf data care
0021	Start auto run - similar to Autorun.Mode (1)
0022	Stop auto run - similar to Autorun.Mode (0)
0101	IP-Help
2010	Show Upload/ Download dialog
2220	Show Change Log
2223	Show Enumerated Data – Report

15.2 Action codes for tool monitors

All monitor actions support the following parameters:

[SHOW|HIDE|TOGGLE]

SHOW - Show the monitor (this is the default action, if no parameter is given)

HIDE - Hide the monitor

`TOGGLE` - Toggle visibility of the monitor

Code	Description
2020	Curve - Monitor Parameter: (additional) `LOAD=` Filename - Load file into curve monitor `SAVE=` Filename - Save file from curve monitor
2030	Global Data Monitor
2040	UMS - Monitor
2050	IP - Monitor
2060	IP Aliases
2070	DDE - Monitor
2080	CAN - Monitor
2090	I²C - Monitor
2100	Virtual Keyboard
2110	KeyWord2000 - Monitor
2120	LIN - Monitor
2130	ME3000 - Monitor
2140	Mechanics IO - Monitor
2150	IEC Bus - Monitor
2160	RS232 - Monitor
2170	Sockets - Monitor
2210	DAS4020/12 - Monitor
2221	SQL - Monitor
2222	IMeas x4 - Monitor
2224	RCMeas - Monitor
2225	ExacqDA High-Speed AD Card Monitor
2226	AngeloRTV Monitor
2227	ME4000 Monitor
2228	SM2044 - Monitor
2229	Frequency counter IO - Monitor

15.3   Action codes for log files

Code	Description
3000	Upload/ Download Log
3010	Error Log
3020	Program Log
3030	Measdata DBF Log
3040	IP-Debug Log
3050	Exception Log
3060	SerIO Log
3080	Socket-Log

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 160 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

15.4   Action codes for special tasks

Code	Description
0014	Program reset
0015	Program end
0016	Shut down Windows Parameter: [SHUTDOWN|REBOOT|POWEROFF] [FORCE] [DIALOG] [NOCONFIRM] SHUTDOWN – Shut down Windows REBOOT – Reboot system POWEROFF - shutdown and power off FORCE - shutdown, even if programs rejects it DIALOG - show user dialog NOCONFIRM - don't ask for user confirmation
0017	Set the system into error state Parameter: Comment
0020	Start measdata dbf data care

15.5   Action codes to start IP steps

Helps to start an IP step.

IP steps can only be started outside of a running test sequence.

The called IP step will start after an actually running one is finished, not parallel.

Code	Description
1000	Start IP-step (everyone)
1001	Start IP-step (master or higher)
1002	Start IP-step (Service or higher)
1003	Start IP-step (Administrator or higher)

Parameter (for all IP actions):

IP-Step

The difference between the action codes is only the password level that is needed to activate the action.

15.6   Action codes to start a shell action

Shell actions can be used to start an external program or to execute a special task with a file (e.g. task 'print' with the file 'xyz.txt').

Code	Description
1100	Shell Action (everyone)
1101	Shell Action (Master or higher)
1102	Shell Action (Service or higher)
1103	Shell Action (Administrator or higher)

Parameter (for all shell actions):

[Action] File[; Parameter[; Path]]

Action is the task to be executed, default is 'open'

File is the file to be executed

Parameter are optional parameters (e.g. that will be given to an executed program)

`Path` is the folder where the action will be started

The difference between the action codes is only the password level that is needed to activate the action.

15.7   Action codes for external DLLs

Code	Description
0050	Show Password DLL - Report
0051	Send command to Password DLL Type and syntax of the commands depend on the DLL. The DLL must implement one of the routines TriggerAction or TriggerActionEx. If the routine TriggerActionEx is implemented, the DLL may return values.
2000	Show MeasData DLL - Report
2001	Send command to MeasData DLL Type and syntax of the commands depend on the DLL. The DLL must implement one of the routines TriggerAction or TriggerActionEx. If the routine TriggerActionEx is implemented, the DLL may return values.

16   Appendix 1

16.1   Licenses

The MCD program can be installed on any system without license and allows an unrestricted edition of data. But to access the hardware the program must be licensed. A program can obtain a license in four different ways:

- If a Victor hardware is found, then the program is automatically licensed (one hardware-one license).
- A short time license allows a hardware testing for about 8 hours. Menu below Intern/ License/ Short time license.
- A single or multiple license using the MCD-License-Server. The MCD-License-Server manages and allots the multiple licenses for a single system i.e. described more detailed in 'Order a license key'. Multiple licenses are only possible using an license-server.
- A single license without MCD-License-Server. For single licenses no server is required, only the license file (*.MLK) having the 'License Key' must be copied in the MCD program directory or windows directory (C:\WINDOWS 1.e. C:\WINNT).

16.1.1   Order a license key

The license is interrelated with the software and the PC hardware. Thus to calculate the key particular data is required:

Method 1, using the test program

- Start the test program
- Select the menu 'Intern/License'
- Select the tab 'Order license'
- Select the number licenses required
- Press the button 'Create license request file' and select a location for saving the file. Please edit the file name when there is no other option.

Method 2, using the program 'RequestLicense'

- Start the program RequestLicense
- Select the Software i.e. used.
- Select the number licenses required
- Press the button 'Create .MLR file' and select a location for saving the file. Please edit the file name when there is no other option.

In both cases

- Please send this file to MCD per email. It consists of all the information required to calculate the license key. Using an editor (e.g. Notepad) this file can be read, in order to cross-check the information sent to MCD.
- Using this file the license key is calculated by MCD. It is then sent back along with the information of the key. Please keep a backup of this file in a safe location.

16.1.2   A single license without MCD-License-Server

The MCD-License-Server description for installation below is not required for program versions Build 910 or higher for a single license. Copy the license file (*.MLK) in the windows directory, where it will be located by the program for the available licenses.

16.1.3　　Single or multiple licenses with MCD-License-Server

The following steps are only important if you require the MCD-License-Server for multiple licenses or when you have a program version Build 910 or older.

16.1.3.1 Installation of the MCD-License-Server

- Create a new directory with the name **`<program files>\MCD\Licenses`** where **`<program files>`** stands for the standard Windows directory to install programs. Normally it is called **`C:\Programme`**, but it may vary according to the Windows-version, language or installation type. You may also use another directory but it should however be one of the local drives.

- Copy the files belonging to this program in the newly created folder. Following files should be present:
`McdLicenser.exe`
`Register Licenseserver.bat`
`UnRegister Licenseserver.bat`

- Start the batch file '**`Register Licenseserver.bat`**' in the newly created – **Hint**: Please do not start the batch file present in the installation files! A DOS window should occur for a short time and then vanish.

16.1.3.2 Updates for the MCD-License-Server

When u receive an update, rename the old one just for backup and copy the new one in its place. A new installation is not required. Avoid keeping many versions of the server on the same system.

16.1.3.3 Installation of the license key

Thus to use the key it must be entered in the registry. This is then used by the license server:

- Close all the test programs

- Start the MCD-License-Server manually

- Select the tab 'Add'

- Press the button 'browse…'

- Open the 'MCD License Key' file received rom MCD

- Press the button 'add licenses'

- Close the MCD-License-Server (Alt-F4)

16.1.3.4 Remove the MCD License Server

Start the batch file '**`UnRegister Licenseserver.bat`**'. After that the entire directory can be deleted. If License key has been installed, there exists a key in the registry:
'HKEY_LOCAL_MACHINE\SOFTWARE\MCD\Licenser'. You can delete it if required.

16.2　　Folders/ Files

The program normally uses the folders and files only within the program directory. All files having the starting symbol as a wave (~) are usually backup files. The backup file having file name DATEI.XYZ is DATEI.~XY.

In some cases the information is saved in the new (mostly CSV) as well as old format (DBF). The program always tries to read the data from the new format. If not available then the old format is used. However while saving the data is saved in both the formats. In some cases the old format is neglected.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 164 of 181	MCD® ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

16.2.1   Program folders

Location: .

The executable files (.EXE) and the required DLLs are present in the program folder. The program itself doesn't require any DLLs, but DLLs can be accessed using the interpreter steps.

This directory is the base for the further folders and files.

While using Windows XP® if the new overview is activated then the program creates a file here called 'TestManager.Manifest.exe'. This allows the program to use the new Windows display features.

16.2.2   Type data folder

Location: .\TYPEDATA

Short form: %TDR%

In type data folder the type list is present. Here all the types which are to be tested are present. For each type family a sub folder is present TYPE_xxx where xxx stands for the name of the family.

Files:

CHANGES.LOG
It consists of a list of changes made in the type list

TYPELIST.CSV
The type list with all the type variants, which the program can test.

TYPELIST.DBF
Type list in the old dBase format. It is saved only due to the downward compatibility of the program.

16.2.3   Type specific files

Location: .\TYPEDATA\TYP_XXX

Short form: %TYP% (for the current type)

All the type family specific files like the sequence list and the test step list are present in the individual type family files. XXX is replaced with the respective type number.

Files:

BACK.___
It is the file with the background for the desktop window, when a member of that type family is loaded. It has a preference over the BACK picture from the system folder. The file ending must be BMP or JPG.

CHANGES.LOG
It consists of a list of changes made in the type list

PARAVIEW.INI
It consists of information regarding the layout of the test sequence editor of this family for e.g. column width, column order, preview of the parameter inspector.

PRM_XXX.CSV
Test step parameter table, it consists of the right side of the test sequence editor.

PRM_XXX.DBF
Test step parameter table in the old dBase format. It is saved only due to the downward compatibility of the program.

TBL_XXX.SEQ
Test sequence table. It consists of the left side of the test sequence editor.

MCD Elektronik GmbH	Date: 2018-04-12	Page 165 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH
Title: MCD TestManager CE - Software Manual			

TST_XXX.CSV
Name of the sequence. It consists of the data grid of the test sequence.

TYPEPIC.___
File containing a small picture of the DUT, which can be shown in the info bar. The file ending must be BMP or JPG.

TYPESTAT.INI
Data regarding the local statistic for this family.

16.2.4   Type template folder

Location: .\TYPEDATA\TYP_TMPL

This folder is not required any more and can be deleted.

16.2.5   Interpreter step folder

Location: .\TYPEDATA\IPSTEPS

Short form: %IPS%

It consists of all the interpreter steps.

16.2.6   System data folder

Location: .\SYSDATA

16.2.7   Localisation data

Location: .\SYSDATA\LANGUAGE

This folder consists of a sub directory for each supported language.

16.2.8   Localisation data for the language

Location: .\SYSDATA\LANGUAGE\xxx

Short form: %LNG%

In this folder many INI files are present, which consist of texts in that country language for various system messages.

Additional information for that language is also present (for e.g. country flag).

File:

FLAG.BMP
Flag for the language selection.

IP.INI
Text for interpreter messages

LABELS.INI
Text for the description of static elements on the program window.

LANGUAGE.INI
Information about the language for e.g. name, information of the flag bitmap etc.

STRINGS.INI
Text for the dynamic output messages of the program.

USERMSG.INI

Text, which can be managed by the user

16.2.9 IP-Hilfe Sprache xxxx

Location:.\SYSDATA\LANGUAGE\xxxx\IPHELP

It consist of many files for the interpreter help.

16.2.10 Log folder

Location: .\SYSDATA\LOGFILES

Short form: %LOG%

In this folder log files are present which provide information about the program events occurred so far.

It is especially interesting when problems occur. They are pure ASCII files and can be sent through email.

Files:

APPLICATION.LOG
Program log – send to MCD when problems occur.

CONFIG.LOG
Changes in the basic setup

ERROR.LOG
Error log – send to MCD when problems occur.

EXCEPTION.LOG
Exception log – send to MCD when problems occur.

IPDEBUG.LOG
Files, which can be created by interpreter command DEBUG.LOG.

LIN.LOG
Data saved by the LIN Monitor.

MSD_DBF.LOG
Events occurred during the file maintenance of the Measdata DBF Module.

RS232.LOG
Data saved by the RS232 Monitor.

UNITS.LOG
Information about the program, needed only for debug purposes.

UPDOWNLD.LOG
Events occurred during the upload or download of file to or from the master folder.

16.2.11 Tool data folder

Location: .\SYSDATA\TOOLDATA ()

It consists of INI files regarding the tools of the test system for e.g. the DDE macros used so far etc.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 167 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

16.2.12 Program setup-files

Location: .\SYSDATA\SYSTEM

Short form: %SYS%

In the INI-files in this folder the program memorizes the previous internal settings for the next test. A manual edition is not important.

Files:

BACK.___
It is the file with the background for the desktop window, when a member of that type family is loaded. It is shown when no BACK picture is present in the folder. The file ending must be BMP or JPG.

CAL_xxxxxxxxxxxxxxxx.INI
Instrument related calibration file for the system. xxxxxxxxxxxxxxxx is the hardware code of the system.

EDITTEST.INI
File containing the previous setting of the user in the test sequence editor.

FORMPOS.INI
File containing the previous position and size of the program windows.

I2CBUSSES.DAT
Definition of the I²C bus systems. See chapter 16.6.

IPEDITOR.INI
File containing the previous setting of the user in the interpreter editor.

LOCLSTAT.INI
System related files of the local statistics.

MECHIO.DAT
File containing the name and description of the channels of the Mechanic IO monitor

PASSWORD.INI
File containing the selected passwords and the access levels.

SYSTEM.INI
File containing all the important data regarding the basic setup.

UMSBUSSES.DAT, UMSCARDS.DAT, UMSADRES.DAT
Definition of the UMS. See chapter 16.5.

16.2.13 Report muster

Location: .\SYSDATA\REPORT

It consists of the report musters i.e. saved as sub directories separately according to the report type.

16.3 Simplifies paths of the program

Within the program the user can enter the entire path names in a simplified way using path templates. The templates replace the real path names. The following path templates are defined:

%TYP%\	current type data directory (of the current type)
%LOG%\	directory with the log files
%TDR%\	type data directory (with the type list)
%PRG%\	path of the program EXE

`%SYS%\`	Program set-up-Data
`%IPS%\`	Interpreter steps
`%LNG%\`	current language

The templates should be given only at the beginning of the file name.

16.4   Add a new language

A new language can be added simply by creating a new sub folder within the % SYS%\Language folder. The files of an existing language can be copied into it and translated into the required language. The program finds these files aotomatically.

While translating some special cases regarding the complexity of a software handbook should be considered. This information and the translation can also be provided by MCD if required.

16.5   UMS definitions

The UMS (Universal measurement system) is defined in 3 files outside the TestManager Program.

16.5.1   Buses

Program version after October 2002 are able to support multiple busses with UMS cards. The type of bus and the technology it uses are described in an ASCII-file called 'UMSBusses.DAT'.

16.5.1.1 File

The files must be present in the folder...\SYSDATA\SYSTEM.

The following conditions should be satisfied:

* Each bus should be defined in a separate row

* The sequence of the buses doesn't matter

* Each row consists a bus definition, a comment or is entirely empty

* Comments start in the usual manner '{' or '//' and continue till the end of row

16.5.1.2 Bus definition

Each row can be divided into 5 parts:

* The name of the bus
  The name of the bus must be unique and should not contain any of the reserved interpreter words. It must contain valid interpreter characters.

* The bus technology
  An identifier for the used bus technology.

* The basic address of the bus
  All the cards internally divide the address range from 0..1023 (allows 16 completely loaded buses, extension is possible). Each bus is allotted with 64 addresses from its basic address. The allotted address windows are not allowed to overlap, but gaps are allowed. As per hardware the cards in each bus are configured from 0..63. In the system the cards are managed using the basic address of the bus + individually set card address.

* The specific data for the bus
  Sub bus technology require additional data for the bus. This should be given in brackets. See below for a list of the generally required information.

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 169 of 181	**mCD**® ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	
Title: MCD TestManager CE - Software Manual			

- A description of the bus
  The description is shown in the UMS monitor and is present only as user information.

16.5.1.3 Bus-technology:

16.5.1.3.1 Bus-technology UMSviaME3000

Generally the internal bus in Victor is controlled using the ME3000 card. Currently the UMS bus is controlled using the ME3000 card and thus the UMSviaME3000 bus is present.

Data

BAW   (Bus address width) Range of the address bus, bit 5 or 6. Given in the form Form 'BAW=5' oder 'BAW=6' (Default=5).

Return connections
ME3000.0.A0 .. **ME3000.0.A15**, and in short form as **A0..A15**.
Default return connections (as connected in Victor) is **ME3000.0.A14**, or **A14**

16.5.1.3.2 Bus-technology UMSviaRS232

The bus is controlled using the RS232 and Atmel - Gateway –card. Many buses can be connected on a single RS232 Port. The software doesn't provide this possibility so far.

Data

Port   The RS232-port i.e. used for communication. The defaul value is 'UMS1'. Given in the format 'Port=XXXX'.

Address   Helps to distinguish between the multiple gateways on a RS232 Port. Given in the format 'ADDR=X' where X stands for the decimal value from 0..11, Default=0.
This value is not yet used internally and therefor should net be given or allowed as 0.

Example

( ), (Port=UMS1), (Addr=0), (Port=UMS1; Addr=0), (Addr=0; Port=UMS1)

Return connections
<Port>.*<Address>*.**DIn**
<Port> is the name of the RS232 Port, *<Address>* is the address of the gateway module.
Also **UMS1.0.DIn** when all the settings are set to default.

16.5.1.3.3 Bus-technology UMSviaRCMeas

The bus is controlled using the RS232, SerIO-Bus and RC-modules (from Mr. S. Mummert). Each RC-module can control its own bus. The modules must be defined in the basic set-up below SerIO-bus. They are then accessed just by calling the index in the module list.

Data

Module   The number of the RC module, (not its address on the SerIO-bus) Given in the format 'Modul=1'. No default value, 'Module' must be present.

Example

(Modul=1)

Return connections
<Name>.**DIn**   <Name> is the bus name. Example: **UMSRCMEAS.DIn**

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany	Date: 2018-04-12	Page 170 of 181	
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Revision: V1.4	Author: SW department	MCD ELEKTRONIK GMBH

Title: MCD TestManager CE - Software Manual

16.5.1.3.4 Bus-technology UMSviaAdapCPU

The bus is controlled using the RS232, SerIO-Bus and RC-modules (from Mr. S. Mummert). Each AdapterCPU -module can control its own bus. The modules must be defined in the basic set-up below SerIO-bus. They are then accessed just by calling the index in the module list. Caution: The TestManager Software doesn't check whether firmware version of the Adapter module can execute all the bus commands.

Data

Modul   The number of the AdapterCPU module, (not its address on the SerIO-bus) Given in the format 'Modul=1'. No default value, 'Module' must be present.

Example

(Modul=1)

Return connections

*<Name>***.DIn**   <Name> is the bus name. Example: **UMSADAPCPU.DIn**

16.5.1.3.5 Bus-technology UMSviaMCDPIO

The bus is controlled using the PIO (Parallel I/O) card. A PC can control multiple PIO cards. In this case, different basic addresses must be defined.

Data

Base   Definition of the basic address in the port range of the PC. Range $100 bis $3FC. The default is $310. The values can be decimal (784) or hexadecimal ($310), the hexadecimal has a better overview.

Example

(Base=$318)

Return connections

*<Name>***.C0** .. *<Name>***.C7**

<Name> is the bus name. Example: **UMSPIO.C1**. For the channels C0..C3 there exists alias names: C0=ADW, C1=DIn, C2=I2CIn, C3=CountIn, Example: **UMSPIO.DIn**. The default return connection is the xxx.DIn

16.5.1.3.6 Bus-technology UMSviaME4000

Generally the internal bus in Victor is controlled using the ME4000 card. Currently the UMS bus is controlled using the ME4000 card and thus the UMSviaME4000 bus is present.

Data

BAW   Range of the address bus, bit 5 or 6. Given in the form Form 'BAW=5' oder 'BAW=6' (Default=5).

SLOWDOWN   The access to the UMS bus is artificially prevented, when the bus is coupled with a card having slow opto-couplers (i.e. slow driving power). Then a numerical parmeter is needed to output the delay time in 'µs'.

Example

(BAW=6; SLOWDOWN=10)

Return connections

ME4000.0.A0 .. **ME4000.0.A15**, and in short form as **A0**..**A15**.

Default return connections (as connected in Victor) is **ME4000.0.A14**, bzw. **A14**

MCD Elektronik GmbH Hoheneichstr. 52 – 75217 Birkenfeld – Germany Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10 Info@mcd-elektronik.de – www.mcd-elektronik.de Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter Registergericht Mannheim - HRB 505692	Date: 2018-04-12	Page 171 of 181	MCD ELEKTRONIK GMBH
	Revision: V1.4	Author: SW department	

Title: MCD TestManager CE - Software Manual

16.5.1.3.7 Bustechnologie UMSviaToolmonitor

The internal bus is controlled via an external ToolMonitor, which is configured in the ToolManager.

Daten

MODUL   The name of the ToolMonitor, which controls the internal bus.

Beispiel

(MODUL=ULC)

Rückgabeleitungen

MSBackCh1.. **MSBackCh4**

Default return connections is **MSBackCh1**

16.5.1.4 Files missing

If the file UMSBusses.DAT is absent, then a default bus showing the following data is created:

```
{--------------------------------------------------------------------------}
{ BusName  Typ           Offs Data   Info
{--------------------------------------------------------------------------}
UMSDefault UMSviaME3000 0    (BAW=5) Default UMS Bus
```

16.5.1.5 Example

```
{--------------------------------------------------------------------------}
{ BusName  Typ           Offs Data   Info
{--------------------------------------------------------------------------}
UMSDefault UMSviaME3000 0    ()      Default Bus intern im Victor
UMSSubSys1 UMSviaRS232  100  (UMS1) degraded system using RS232 Port UMS1
```

16.5.2 Cards

All the cards present in the Universal measurement system are made familiar to the test program by the file UMSCards.DAT.

16.5.2.1 File

The file must be present in the folder ...\SYSDATA\SYSTEM.

The following conditions should be satisfied:

- Each card should be defined in a separate row

- The sequence of the card doesn't matter

- Each row consists a card definition, a comment or is entirely empty

- Comments start in the usual manner '{' or '//' and continue till the end of row

- The individual elements of the row are separated by a space or a tab (an exception is the optional data, everything within brackets is considered as data)

16.5.2.2 Card definition

Each row can be divided into 5 parts:

- The name of the defined card
  The name of the card must be unique and should not contain any of the reserved interpreter words. It must contain valid interpreter characters.

- The card address in the UMS
  The address range for each UMS bus ranges from 0 to 31 or 0 to 63 plus the basic (or offset) address of the bus. The address offset +0 and offset +32 are not used.
  Each address may be occupied by a single card only. Caution: Some addresses need an address i.e. divisible by 4(8) and they may need more than one address in the increasing order.
  For the buses having 5 address bits (e.g. UMSviaME3000) only 0..31 addresses are available. On the cards the address must always be added with +32 because the 6 address line is always high.
  The address of the cards can be allotted as sum of the offset address of the bus and card address(e.g.104) or in the form 'Busname.Address' (e.g. UMSSubSys1.4). The previous notation is available only after build 954.

- Identification code for the card type
  The identification code decides for the card type. The various identification codes for the card types are shown in the tables below:

- Optional data for initialisation
  Some card can be configured in different Victor variants. Using this data the system can be informed about the configuration. The data should be separated with semicolon in round brackets. If no data is to be entered an empty '()' can be present.

- A description of the card
  The description is shown in the UMS monitor and is present only as user information.

16.5.2.3   Type of cards

16.5.2.3.1   ADW16B

Card type	: Analog-digital-converter card 16 channels, 16 bit
Address	: 1 address at any position
Data	: 1 optional parameter:

BackCH=xx decides the return connections. The allowed values are xx:
all the return channels of the bus which the card uses.
If the BackChannel is not given, then the card connects
with the default return channels of the bus.

EDx=y Defines the external divider (External divider) y of the channel x. Condition: $1 \leq x \leq 16$
Formula: `Displayed value  = Measured value * External divider`

16.5.2.3.2   DAC5450

Card type	: Analoger Ausgabekarte mit einem Kanal (DAC-Karte)
Address	: 1 address at any position
Data	: 1 optional parameter:

BackCH=xx decides the return connections. The allowed values are xx:
all the return channels of the bus which the card uses.
If the BackChannel is not given, then the card connects
with the default return channels of the bus.

16.5.2.3.3   DIN5450

Card type	: Digitale input card 32 channels
Address	: 1 address at any position
Data	: 2 optional parameters:

MCD Elektronik GmbH	Date: 2018-04-12	Page 173 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

BackCH=xx   decides the return connections. The allowed values are xx:
    A14   (default)
    A12   (substitute connection)
further values are possible depending on the system set-up. If the BackChannel is not given, then the card connects with the default return channels of the bus.

PGA=xx   Decides for the type of PGAs. . The allowed values are xx:
    10   selects PGA with 1 / 10 / 100 / 1000 – pattern (default)
    205   like 10 (Part of the PGA Chip description)
    2   selects PGA with 1 / 2 / 4 / 8 – pattern
    204   like 2 (Part of the PGA Chip description)
    0   no PGA configured, Amplification is always 1
    none   like 0

16.5.2.3.4   DOUT5450

Card type   : Digitale output card 32 channels
Address   : 1 address at any position
Data   : 1 optional parameter:
Channels=$FF   A 32 Bit bit-pattern, which decides for the existing or non-existing channels to be displayed in the monitor. A set bit represents an activated channel. Range of the values: $0 .. $FFFFFFFF, Default: $FFFFFFFF

16.5.2.3.5   IUCONV

Card type   : Current-/Voltage converter card
Address   : 1 address at any position
Data   :   optional: 2 numerical values, which represent the resistance values of both the shunts in OHM. Default: 0.1 und 10

16.5.2.3.6   MUX5450

Card type   : Relay multiplexer 16 x toggle switch
Address   : 1 address at any position
Data   : 1 optional parameter:
Channels=$FF   A 16 Bit bit-pattern, which decides for the existing or non-existing channels to be displayed in the monitor. A set bit represents an activated channel. Range of the values: $0 .. $FFFF, Default: $FFFF

16.5.2.3.7   NfGEN16

Card type   : Sine-wave generator (low frequency) with 16x multiplexer
Address   : 2 Addresses after an even address
Data   : None

16.5.2.3.8   RCMUX48X2

Card type   : Bauteiltest multiplexer 48 x 2
Address   : 3 Address ab einer glatt durch 4 teilbaren Adresse
Data   : None

16.5.2.3.9   RCMUX192X2

Card type   : Component test multiplexer 192 x 2

MCD Elektronik GmbH	Date: 2018-04-12	Page 174 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

Address   : 15 Address ab einer glatt durch 4 teilbaren Adresse, Die Address Basis+3, Basis+7 und Basis+11 bleiben verwendbar!

Data   : keine

16.5.2.3.10   RNET2X15

Card type   : 2-fach R-Netz 15 Bit Auflösung

Address   : 1 Adresse an beliebiger Position

Data   : zwei numerische Werte, welche die Auflösung der Kanäle 1 und 2 in Ohm angeben

16.5.2.3.11   XMUX32X4

Card type   : Kreuzschienenmultiplexer 32 x 4

Address   : 4 addresses after an address i.e. completely divisible by 4

Data   : None or an entry 'PCB=x', where x stands for the layout version. (Version 0 had a problem, which is corrected by the software) The default is version 1.

16.5.2.3.12   XMUX32X8

Card type   : Cross-multiplexer 32 x 8

Address   : 8 addresses after an address i.e. completely divisible by 4

Data   : None or an entry 'PCB=x', where x stands for the layout version. (Version 0 had a problem, which is corrected by the software) The default is version 1.

16.5.2.3.13   XMUX64X8

Card type   : Cross-multiplexer 64 x 8

Address   : 16 addresses after an address i.e. completely divisible by 4

Data   : None

16.5.2.3.14   XMUX96X8

Card type   : Cross-multiplexer 96 x 8

Address   : 24 addresses after an address i.e. completely divisible by 4

Data   : None

16.5.2.4   Example

```
//----------------------------------------------------------------------
// Name     Adr.    Typ       opt.                  Description
// System   0..31   ID code   Data                  Card
//----------------------------------------------------------------------
RelMux1     1       MUX5450   ()                    Relay multiplexer1
RGeber1     3       RNET2x15  (1.0; 2.0)            2xR-generator
RGeber2     4       RNET2x15  (0.5; 4.0)            2xR-generator
XMux1       8       XMUX32x8  ()                    Cross multiplexer
XMux2       12      XMUX32x8  (PCB=0)               Cross multiplexer old
DigIn       17      DIN5450   (BackCh=A14; PGA=2)   Digital Input card
```

16.5.3   Addresses

For channels or group of channels the addresses can be defined, which allow these channels to be accessed from the interpreter. There are defined as ASCII-files with the name 'UMSAdres.Dat'.

16.5.3.1   File

The file must be present in the folder ...\SYSDATA\SYSTEM.

MCD Elektronik GmbH	Date: 2018-04-12	Page 175 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

The following conditions should be satisfied:

- Each address should be defined in a separate row

- The sequence of the address doesn't matter (except for the descriptions of the tool monitor which are written twice and the first row is used as the descriptions of the tool monitor)

- Each row consists a address definition, a comment or is entirely empty

- Comments start in the usual manner '{' or '//' and continue till the end of row

16.5.3.2   Card definition

Each row can be divided into 5 parts:

- The name of the address
  The name of the address must be unique and should not contain any of the reserved interpreter words. It must contain valid interpreter characters.
  For a single address many names can be given. Thus for a single address (same channel, same relays) many different names can be defined. Within the interpreter the definition is same, but for the monitor description only the first definition is used.

- The card address in the UMS
  An address is always related to a channel of the card in the UMS (or it represents a group of channels of a card). Using the card address the type of the address is decided, since the program can identify the type of the card.
  The card to which the address is related must be defined in the address file UMSCards.DAT. From version build 955 onwards instead of the numerical card address the name of the card can be given as an alternative.

- The data of the address
  Depending on the type of card used containing the addresses, they can contain different data. For e.g. for a multiplexer a bit pattern, for a R-Net a channel number. Partially using this data the type of address is further decided, because for some cards have different types of addresses (e.g. rows and columns by cross multiplexer). The different types of data is described below in detail. Numerical values can be given in decimal or hexadecimal (with a leading '$' symbol).

- A description of the card
  The description is shown in the UMS monitor and is present only as user information.

16.5.3.3   Address structure for the different cards

For the cards which are not described here, no further addresses can be defined (except the card addresses).

16.5.3.3.1   DIN5450

Card type   : Digital input card 32 channels
Data   : numerical Value from 1 to 32 for the channel numbers

16.5.3.3.2   DOUT5450

Card type   : Digital output card 32 channels
Data   : numerical Value from $0..2^{32}-1$, showing a bit pattern

16.5.3.3.3   MUX5450

Card type   : Relay multiplexer 16 x toggle switches
Data   : numerical Value from 0..65535, showing a bit pattern

MCD Elektronik GmbH	Date: 2018-04-12	Page 176 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany			
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10	Revision: V1.4	Author: SW department	
Info@mcd-elektronik.de – www.mcd-elektronik.de			
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter			
Registergericht Mannheim - HRB 505692			
Title: MCD TestManager CE - Software Manual			

16.5.3.3.4 RNET2X15

Card type   : 2-fach R-Netz 15 Bit resolution

Data   : numerical Value from 1 or 2, showing a bit pattern

16.5.3.3.5 XMUX..X.. and RCMUX..X..

Card type   : Cross multiplexer and Component-multiplexer

Data   : Identifier characters, following the row or column number

      ,R5'   - defines the row address (Row) for the 5th row

      ,C3'   - defines the column address (Column) for the 3rd column

      ,X5;3'   - definiert Kreuzungsadresse von Zeile 5 und Spalte 3

The values of the rows and columns must be within the rows and columns realised or present in the card. It is counted from 1.

16.5.3.4 Example

```
//---------------------------------------------------------------------
// Name in  Adr.   Data       Description
// System   0..31             The address
//---------------------------------------------------------------------
XMuxS1      8      C1         Column 1
XMuxS2      8      C2         Column 2
XMuxZ2      8      R2         Row 2
Xmux2X3     8      X2;3       Cross row 2 Column 3
Xmux7X13    8      X7;13      Cross row 7 Column 13

MUX1_OFF    1      $0000
MUX1_ALL    1      $FFFF

MUX1_1      1      $0001      Relay multiplexer 1 Channel 1
MUX1_2      1      $0002      Relay multiplexer 1 Channel 2

RX1         3      $1         Resistance Generator 1 Kanal 1
RX2         3      $2         Resistance Generator 1 Kanal 2
```

16.6 I2C Bus definition

The I²C bus data is defined in a separate file (%SYS%\I2CBusses.Dat). If the file is absent, then a default bus using the ME3000 card is defined.

Each row (i.e. not empty) in the file defines a I²C bus. The number of the defined buses is not limited.

Comments start in the usual manner '{' or '//' and continue till the end of row. The rows are constructed in a following manner:

BusName BusTyp (BusData) Info

The individual elements are separated by a space or a tab, but they must however lie in the same row.

BusName is the name used by the interpreter to access the bus. The name of the address must be unique and should not contain any of the reserved interpreter words. It must contain valid interpreter characters. The name is also independent of the small or big characters.

BusTyp decides for the type of bus (depending on the hardware). It must belong to any of the I²C Bus types defined in the program. See details below:

BusData decides for the used hardware and further options. The entries in the BusData part depend on the bus type used. If no data is present an empty () should be entered.

Info is only an optional description and is present only as user information.

16.6.1   BusTyp I2CviaME3000

The BusTyp **I2CviaME3000** realises the I²C Bus using the digital IO connections of the ME3000 card.

Two output connection for SCL and SDAOut and one input connection for SDAIn is needed. Using an external circuitry SDAOut and SDAIn must be interconnected.

Possible entries as data:

Name	Allowed	Default	Meaning
SDAIn	A0..A15	A13	Channel i.e. required to read back the SDA
SDAOut	B0..B15	B13	Channel i.e. required to output the SDA
SCLOut	B0..B15	B14	Channel i.e. required to output the SCL

The set default values are channels of the MCD bus, i.e. used twice for I²C purposes. This I²C bus helps to control the internal I²C functions. Thus for external applications in the test system an I²C bus having different channels should be defined.

A sensible range is

- SDAIn : A0..A11
- SDAOut : B0..B7
- SCLOut : B0..B7

Example of a bus definition:

```
//-----------------------------------------------------------------
// BusName     Type            Data                        Info
//-----------------------------------------------------------------
I2C_ME         I2CviaME3000 (SDAIn=A7;SDAOut=B7;SCLOut=B6)
```

16.6.2   BusTyp I2CviaRS232

The BusTyp **I2CviaRS232** realizes an I²C Bus using the handshake signals of an RS232 port.

The following pin order will be used:

RS232-Pin	**I²C Function**
RTS	SDA-out
CTS	SDA-in
DTR	SCL-out

It is not possible to change the pin order .

Values entered in **Data**:

Name	Allowed	Default	Meaning
Port	1, 2, 3...	-	Number of the used Com-Ports. It must be given, no default!
Invert	false, true	True	Set as True, when the channels are used externally with Max232 or when they are inverted

The port can also be used for communication without any problem, since the I²C-Bus influence only the handshake signals. The port must not be defined also as an SerPorts. It is irrelevant whether the port is opened by an RS232 object.

16.6.3   Bus types, used in a SerIO Module

Modules which are accessed using a serial port and a SerIO bus, for e.g.

- MechIO Module

- AdapterCPU Module
- IMeasX4 Module, etc.

Cab be used as a port for I²C Bus. To successfully access such an I²C Bus following systems should be ready and activated:

- The global serial port
- The serial port of that SerIO Bus module
- The module-class global
- The module, which is used as gateway

→ Or in other word the gateway module must be accessible.

According to the programming done please consider the information in the interpreter help.

16.6.3.1 Bus type I2CviaMechIO

The bus type **I2CviaMechIO** uses a Mechanic module as a gateway for the I²C Bus.

Possible entries in the data:

Name	Allowed	Default	Meaning
Module	1, 2, 3 ...	-	Number of the MechIO module in the basic setup, not its address or number of the used COM-Ports. It must be given, no default!

Example for a bus definition:

```
//----------------------------------------------------------------
// BusName    Type          Data                 Info
//----------------------------------------------------------------
I2C_MechIO   I2CviaMechIO (Modul=1)              I²C über Mechanic Modul 1
```

16.6.3.2 Bus type I2CViaAdapCpu

The bus type **I2CViaAdapCpu** uses a Adapter-CPU module as a gateway for the I²C Bus.

Possible entries in the data:

Name	Allowed	Default	Meaning
Modul	1, 2, 3 ...	-	Number of the Adapter-CPU module in the basic setup, not its address or number of the used COM-Ports. It must be given, no default!

Example for a bus definition:

```
//----------------------------------------------------------------
// BusName    Typ           Data         Info
//----------------------------------------------------------------
I2C_AdapCPU  I2CviaAdapCPU (Modul=1)     I²C using Adapter CPU Module 1
```

16.6.3.3 Bus type I2CViaIMeasX4

The bus type **I2CViaIMeasX4** uses a 4X current measurement module (IMeasX4) as a gateway for the I²C Bus.

Possible entries in the data:

Name	Allowed	Default	Meaning
Modul	1, 2, 3 ...	-	Number of the IMeasX4 module in the basic setup, not its address or number of the used COM-Ports. It must be given, no default!

Example for a bus definition:

```
//----------------------------------------------------------------
// BusName    Typ           Data         Info
//----------------------------------------------------------------
I2C_IMeasX4  I2CviaIMeasX4 (Modul=1)     I²C using 4X Current measurement
module 1
```

16.6.3.4 Bus type I2CViaRCMeas

The bus type **I2CViaIRCMeas** uses a 4X current measurement module (RCMeas) as a gateway for the I²C Bus.

Possible entries in the data:

Name	Allowed	Default	Meaning
Module	1, 2, 3 ...	-	Number of the RCMeas module in the basic setup, not its address or number of the used COM-Ports. It must be given, no default!

Example for a bus definition:

```
//----------------------------------------------------------------
```

```
// BusName    Type           Data        Info
//----------------------------------------------------------------
I2C_IMeasX4  I2CviaIMeasX4  (Modul=1)   I²C using an RCMeas module 1
```

16.6.3.5   I²C chip PCF8574

The IP object PCF8574 realizes the digital output using the chip PCF8574 on the I²C Bus.

The individual components must be defined in the I2CBusses.Dat and will be accessed using these names.

The definition is within the I2CBusses.Dat and is separately defined in a block which is followed by the title [PCF8574]. After this on each row a PCF chip will be defined.:

Example for a I2CBusses.DAT:

```
{----------------------------------------------------------------------------}
// BusName  Typ          Data                          Info
{----------------------------------------------------------------------------}
I2C_MCD     I2CviaME3000 (SDAIn=A13;SDAOut=B8;SCLOut=B9) Default Bus in Victor internally
I2C_Adapter I2CviaRS232  (Port=1; Invert=true)         a degraded system using a RS232 Port1
//----------------------------------------------------------------------------
// Name BusName    Adr. Data
//----------------------------------------------------------------------------
[PCF8574]
PCF0   I2C_Adapter 0  (Used=$3F;INVERT=$ff;PINS=Rel1,Rel2,Rel3,Rel4,Rel5,Rel6,--,--)
PCF1   I2C_Adapter 1  (Used=$3F;INVERT=$ff;PINS=Rel7,Rel8,Rel9,Rel10,Rel11,Rel12,--,--)
PCF2   I2C_Adapter 2  (Used=$3F;INVERT=$3f;PINS=Rel13,Rel14,Rel15,Rel16,Rel17,Rel18,--,--)
PCF3   I2C_Adapter 3  (Used=$3F;INVERT=$3f;In=$C0;PINS=Rel19,Rel20,Rel21,Rel22,Rel23,Rel24,--,--)
PCF4   I2C_Adapter 4  (Used=$1F;INVERT=$1F;IN=$E0;PINS=Rel25,Rel26,Rel27,Rel28,Rel29,IN,IN,IN)
{--Ende der Datei--}
```

The row with defines the [PCF8574].

Each row begins with the name of the PCF8574 chip, using which they can be accessed. It must contain valid interpreter characters.

The chip name is followed by the bus name on which the chip is connected. This bus name should be defined above in the file.

The third element is the address of the chip i.e. fixed by the three address pins. Thus the valid addresses are 0-7 for the PCF8574 **P** types. While for the PCF8574 **AP** types the addresses 8-15 can be used. Each bus may use a single address only once.

The fourth element is a list from optional additional data in brackets. The additional data are constructed like 'Name = Value'. The 'Name' gives the type of data and the allowed names are:

USED gives the number of bits that are used, followed by an 8 bit value. A set bit represents, an used bit (Default = $FF / 255). This value has influence only on the monitor and the unused pins are shown in grey.

INVERT shows the output signal i.e. to controlled in an inverted manner (active = low, inactive = high). It is followed by an 8 bit value where a set bit represents an inverted pin (Default=$00 / 0).

IN gives the number of bits that are used as inputs and is followed by an 8 bit value. A set bit represents, that the pin is used as an input. (Default =$00 / 0).

It should be known that a pin cannot be defined as an input as well as an inverted output. Thus a (IN & INVERT) must be always 0.

PINS it helps to define individual pin names in the monitor where each name is separated by comma. The name should itself not contain a comma or semicolon**.**

MCD Elektronik GmbH	Date:	Page 181 of 181	
Hoheneichstr. 52 – 75217 Birkenfeld – Germany	2018-04-12		
Tel. +49-7231 78405-0 – Fax: +49-7231 78405-10			
Info@mcd-elektronik.de – www.mcd-elektronik.de	Revision: V1.4	Author:	
Sitz : Birkenfeld - Geschäftsführer : Bruno Hörter		SW department	
Registergericht Mannheim - HRB 505692			

Title: MCD TestManager CE - Software Manual

17 Abbreviations

AUTO	automatic
ADC	analog to digital converter
BDE	Borland Database Engine
BAW	Bus address width
cPCI	compact peripheral component interconnect
CAN	Controller area network
DAC	digital to analog converter
DIN	digital input
DOUT	digital output
DUT	device under test
DLL	dynamic link library
ECU	electronic control unit
ID	identification
IP	interpreter
MUX	multiplexer
PGA	programmable gain amplifier
PC	personal computer
SBS	step by step
UMS	universal measurement system