

SOFTEMA Cookbook 2

Project planning guide for SOFTEMA

SOFTEMA - Software of control systems on machines v1.3.0 - SOFTEMA_Example.xlsx (Logged out: Read mode)

File Edit Print View Extras Role:Read mode Phase: All phases Help

Last used ▾ Open... Save Save as... Close... Project Team Login/Logout Manage documents Previous page

Project A1 Safety functions A2.4 IO list A3 Measures A4 Requirements B3 Module architecture B4 Matrix C+E B4 Matrix compact C1 Codereview D1 Validation Changes

Update table Hide columns Inputs Outputs Activate selection Show selection

No	Operating mode	Test	SF.No	SFID	Prio	SF name	O1	O3	O4	O2	Lock	Verification	Validation	Cor
							OS_M1 [A24.0]	OS_M2_STO [A32.0]	OS_M2_SLS [A32.4]	OS_M3 [A24.2]				
C0						ALLOK	ON	ON	ON	ON	x			
C1	B0: All	C0	SF1	-SF10.1		1 If emergency stop EMST, then Motor M1 switch off, Motor M2 in STO, Motor M3 switch off, with acknowledge button ACK acknowledge.	OFF (*IM1*)	OFF (*IM1*)	NOP	OFF (*IM1*)	x			
C2	B1: Automatic	C0	SF2	-SF11.1.1		2 If guard door SG1, then Motor M1 switch off, with acknowledge button ACK acknowledge.	OFF (*IM2*)	NOP	NOP	NOP	x			
C3	B1: Automatic	C0	SF3	-SF11.2.2		2 If guard door SG2, then Motor M2 in STO, with acknowledge button ACK acknowledge.	NOP	OFF (*IM3*) AND	NOP	NOP	x			
C4	B1: Automatic	C0	SF4	-SF11.3.1		2 If guard doors SG2 & SG3, then Motor M1 switch off, with acknowledge button ACK acknowledge.	OFF (*IM3*) OR	NOP	NOP	NOP	x			
C5	B1: Automatic	C0	SF5	-SF11.4.3		2 If edge protection sensor fast-moving gate SL_SG2, then Motor M3 switch off, with acknowledge button ACK acknowledge.	NOP	NOP	NOP	OFF (*IM6*)	x			
C6	B2: Setup mode	C8	SF6	-SF14.1.2		2 If link SG2 & /SG3 & 3S1, then Motor M2 in SLS, with acknowledge button ACK acknowledge.	NOP	OFF not	OFF (*IM5*)	NOP	x			
C7	B2: Setup mode	C8	SF7	-SF14.2.2		2 If link SG2 & /SG3 & 3S2, then Motor M2 in SLS, with acknowledge button ACK acknowledge.	NOP	OFF not	OFF (*IM5*)	NOP	x			
C8	B2: Setup mode	C0	TF1			2 SG2 open, SG3 closed, IS_TIP1, 2 not operated	NOP	OFF	ON	NOP	x			
C9	B2: Setup mode	C8	TF2			2 SG2 open, SG3 closed, IS_TIP_1, 2 operated	NOP	OFF	ON	NOP	x			
€€€€											x	0 %	0 %	

C:\SOFTEMA EN PROJECTS\ Table changed Ini file: SOFTEMA.ini CB:<empty>

Author: Albert Bohlscheid, Martin Ulrich, Andy Lungfiel, Michael Huelke
Institute for Occupational Safety and Health (IFA) of the
German Social Accident Insurance (DGUV)
Alte Heerstr. 111
53757 Sankt Augustin
Phone: +49 30 13001-0
Fax: +49 30 13001-38001
Internet: www.dguv.de/ifa

Published by: German Social Accident Insurance (DGUV)
Glinkastrasse 40
10117 Berlin

- January 2025 -

Table of contents

1	Introduction	6
2	About this guide	7
3	SOFTEMA at a glance	9
3.1	What can SOFTEMA do?.....	9
3.2	How is SOFTEMA used? - A brief tour through the program.....	9
3.3	Where is SOFTEMA available?.....	11
3.4	Installation and versioning.....	11
3.5	Interfaces to SOFTEMA.....	11
3.6	The role concept	11
3.7	The user administration.....	12
4	Preparation of project planning with SOFTEMA	13
4.1	Specification of the safety requirements.....	13
4.2	List of signals and function blocks	13
4.3	Error-preventing/error-controlling measures.....	13
5	Setting up a project file	14
5.1	Determine project organization	14
5.2	Selecting and setting up the project directory.....	14
5.3	Setting options for SOFTEMA.....	14
5.4	Selecting a suitable project template.....	14
5.5	Customizing the project template	15
6	Entering the project information and dates.....	19
6.1	Entering project information	19
6.2	Defining safety functions	20
6.3	Entering input and output signals	20
6.4	Configuring tables for measures and requirements	21
6.5	Entering modular architecture	21
6.6	Adding to the "Persons" table via user administration	21
6.7	Adding to the "Documents" table.....	21
7	Software design	22
7.1	Updating and adding the "B4 Matrix C&E" table.....	22

7.2	Updating the "B4 Matrix compact" table	25
7.3	Verification and validation plans.....	26
8	Coding of the application program	27
8.1	Measures as part of the tool qualification	27
9	Verification of the coded program	28
9.1	Measures as part of the tool qualification	28
9.2	Verification in the "A2.4 IO list" table	28
9.3	Verification in the "A3 Measures" table	28
9.4	Verification in the "B3 Modular architecture" table.....	28
9.5	Verification in the "B4 Matrix C&E" table	29
9.6	Verification in the "B4 Matrix compact" table	29
9.7	Verification in the "Code review" table.....	29
10	Validation of the application program	30
10.1	Measures as part of the tool qualification	30
10.2	Validation in the "A2.4 IO list" table	30
10.3	Validation in the "B4 Matrix C&E" table	30
10.4	Validation in the "B4 Matrix compact" table	30
10.5	Validation in the "A1 Safety functions" table.....	30
10.6	Validation in the "A4 Requirements" table	31
10.7	Validation in the "Validation" table.....	31
11	Checking the project file.....	32
11.1	Visual inspection in the tables.....	32
11.2	Comments in the tables	32
11.3	Protocol fields	32
12	Print functions	33
12.1	Printer setup	33
12.2	Printing tables	33
12.3	Creating a summary.....	33
13	Application program documentation	37
14	Modification of the application program.....	38
14.1	Modification of the project data	38

14.2 Updating the specification tables.....38

14.3 Verification, validation and testing of the modifications39

14.4 Documentation of the modifications39

Appendix A : Literature40

Appendix B : List of abbreviations41

1 Introduction

Machine manufacturers are increasingly implementing safety functions by programming safety programmable logic controllers. The standards DIN EN ISO 13849-1 [1] and DIN EN 62061 [2] define, among other things, requirements for the software development of safety functions. This is intended to prevent dangerous systematic errors in the application software for a machine.

In DGUV project FF-FP0319 "Norm compliant development and documentation of safety related application software in manufacturing system engineering" [3] (2011 to 2013), the Bonn-Rhein-Sieg University of Applied Sciences devised a specific procedure for implementation of the requirements contained in the new standards for the software development of safety functions for machines. The method was evaluated and documented using industrial examples. The IFA subsequently published the project results as part of IFA Report 2/2016 "Safety-related application software for machinery - The IFA matrix method -" [4]. To implement and simply apply this matrix method, the IFA is developing the SOFTEMA software [5], which, like the IFA tool SISTEMA [6], is available to download free of charge.

SOFTEMA Cookbook 1 is available in SOFTEMA as program documentation and reference manual via the menu command HELP → SOFTEMA COOKBOOK 1. This Cookbook 2, on the other hand, describes the steps for using SOFTEMA throughout the development process (menu command HELP → SOFTEMA COOKBOOK 2).

Note: It is strongly recommended that you read the IFA Report 2/2016 and the SOFTEMA Cookbook 1 that goes with the version before using SOFTEMA.

An important prerequisite for working with SOFTEMA and the project files is an understanding of the application of the IFA matrix method and the EN ISO 13849 series of standards in general. The IFA supports you with free publications:

- The IFA provides information on the EN ISO 13849 series of standards at <https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/sicherheit-von-maschinensteuerungen/index.jsp>
- and information about SOFTEMA on the page <https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-softema/index.jsp>.
- Further information, instructions and examples on application programming using the IFA matrix method can be found in the IFA Report 2/2016 [4] and in the SOFTEMA examples for download: <https://www.dguv.de/ifa/fachinfos/arbeiten-4-0/industrie-4-0/sicherheitsbezogene-maschinensoftware/index-2.jsp>
- The definition of safety functions is described in SISTEMA Cookbook 6 [7], see <https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-sistema/sistema-kochbuecher/index.jsp>.

2 About this guide

Safety-related application software (SRASW) for machinery can be specified, validated and documented in accordance with standards using the IFA matrix method. To implement the IFA matrix method, the IFA software SOFTEMA can be used to edit Excel project files.

Note: SOFTEMA - like any software tool for the development and verification of safety-related control systems - must be qualified for use. SOFTEMA falls into the category of "offline support tools". Therefore, when using SOFTEMA, the effects of potential tool errors and the required risk reduction of the developed safety functions must be evaluated in order to define appropriate error-preventing measures (e.g. review of the SOFTEMA results; test of the software modules developed with the SOFTEMA results; etc.).

It is therefore essential that you observe the specifications in SOFTEMA Cookbook 1 regarding tool qualification and the necessary error-preventing measures to be taken by SOFTEMA users. These measures are specifically named in the project phases in the following chapters.

This guide describes the application of SOFTEMA along the development process (simplified V-model: Figure 1) in the following chapters:

- Brief description of SOFTEMA in Chapter 3
- Preparatory project work in the Chapters 4, 5 and 6
- Constructive activities with software design and coding in Chapters 7 and 8
- Testing activities with verification, validation and testing in Chapters 9, 10 and 11
- Final work in the Chapters 12 and 13
- Modification of application programs in Chapter 14

The appendices and provide references and useful information.

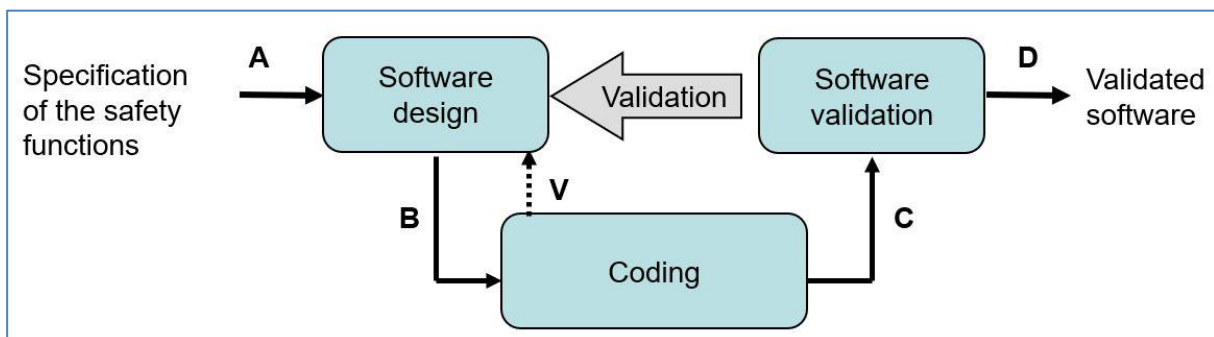


Figure 1: V-model for the software development of safety functions

Change history

The cookbook matches the SOFTEMA version if the version number matches. Changes compared to the previous cookbook version are marked in **yellow** in the text.

1.0.0: Cookbook 2 for SOFTEMA Version 1.0.0

1.1.0: Cookbook 2 for SOFTEMA Version 1.1.0

- None changes

Layout conventions

The following formats are used in this cookbook:

Italics

is used for file names and extensions, new terms and highlighting.

SMALL CAPS

indicate elements of the user interface such as menu names and buttons.

Table font

used in tables for column and row identifiers (while the *table* names are enclosed in quotation marks).

Boxes

highlight notes and warnings.

3 SOFTEMA at a glance

This chapter provides an overview of the basic features and functions of this tool. Further information and user aids (e.g. other SOFTEMA cookbooks) can be found on the SOFTEMA download page (<https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-softema/index.jsp>).

3.1 What can SOFTEMA do?

SOFTEMA can only open and edit one project file at a time for the specification and documentation of an application program. However, the software can be run multiple times to view and edit different projects and programs in parallel. This means that project data can be copied and pasted between several SOFTEMA instances (or Excel instances) via the clipboard.

SOFTEMA project files use the file type of a Microsoft Excel workbook (*.xlsx). The project files can be edited either with SOFTEMA or directly with Microsoft Excel. In Excel, all tables in the worksheets are freely editable. In SOFTEMA, the contents are protected by the user administration. The specialized functions described below are naturally only available under SOFTEMA. The project files do not contain macros and SOFTEMA cannot open Microsoft Excel workbooks with macros. All SOFTEMA functions are integrated and protected in the software. However, additional worksheets can be inserted under Excel and used for development and documentation, e.g. for documentation of the control hardware. SOFTEMA can load and display these additional worksheets, but cannot edit them.

3.2 How is SOFTEMA used? – A brief tour through the program

SOFTEMA manages the tables required for the IFA matrix method [4] and also the information required for project management, such as project description, user administration, change logs, document management, etc. Figure 2 shows, for example, the Cause&Effect (C&E) matrix for the software specification of a project in SOFTEMA.

No.	Operating mode	Test	I7	I5	I6	I3	I4	I1	I2	I8	I9	I10	SF-No.	SFID	Prio	SF name	O1	O3	O4	O2	Lock	Verification	Validation	Comment
C0			1	1	1	1	1	1	1	1	1	0				ALLOC	ON	ON	ON	ON	x			
C1	B0-All	C0	1	1	1	1	1	1	1	1	0	0	SF1	-SF10.1	1	If emergency stop EMST, then Motor M1 switch off, Motor M2 in STO, Motor M3 switch off, with acknowledge button ACK acknowledge.	OFF ("IM1")	OFF ("IM1")	NOP	OFF ("IM1")	x			
C2	B1-Automatic	C0	1	0	1	1	1	1	1	1	0	0	SF2	-SF11.1	2	If guard door SG1, then Motor M1 switch off, with acknowledge button ACK acknowledge.	OFF ("IM2")	NOP	NOP	NOP	x			
C3	B1-Automatic	C0	1	1	1	0	1	1	1	1	0	0	SF3	-SF11.2	2	If guard door SG2, then Motor M2 in STO, with acknowledge button ACK acknowledge.	NOP	OFF ("IM2") and	NOP	NOP	x			
C4	B1-Automatic	C0	1	1	1	0	0	1	1	1	0	0	SF4	-SF11.3	2	If guard doors SG2 & SG3, then Motor M1 switch off, with acknowledge button ACK acknowledge.	OFF ("IM3") and	NOP	NOP	NOP	x			
C5	B1-Automatic	C0	1	1	1	0	1	1	1	1	0	0	SF5	-SF11.4	3	If edge protection sensor or fast-moving gate SL_SG2, then Motor M3 switch off, with acknowledge button ACK acknowledge.	NOP	NOP	NOP	OFF ("IM5")	x			
C6	B2-Setup mode	C8	1	1	1	0	1	1	1	1	0	0	SF6	-SF14.1	2	If link SG2 & /SG3 & 3S1, then Motor M2 in SLS, with acknowledge button ACK acknowledge.	NOP	OFF not	OFF ("IM5")	NOP	x			
C7	B2-Setup mode	C8	1	1	1	0	1	1	1	1	0	0	SF7	-SF14.2	2	If link SG2 & /SG3 & 3S2, then Motor M2 in SLS, with acknowledge button ACK acknowledge.	NOP	OFF not	OFF ("IM5")	NOP	x			
C8	B2-Setup mode	C0	1	1	1	0	1	1	1	1	0	0	TF1		2	SG2 open, SG3 closed, IS_TIP1, 2 not operated	NOP	OFF	ON	NOP	x			
C9	B2-Setup mode	C8	1	1	1	0	1	1	1	1	1	1	TF2		2	SG2 open, SG3 closed, IS_TIP1, 2 operated	NOP	OFF	ON	NOP	x			
=====																					0 %	0 %		
																					Date	20.03.2015	27.03.2015	
																					Name	Johanna Dietz	Marcel Benus	

Figure 2: C&E matrix in SOFTEMA

For a new project, the user opens an empty but already pre-formatted project template. After completing the project description ("Project" table), the safety functions with their properties such as PL_r, operating type, priority etc. are entered in table "A1 Safety functions" (see SISTEMA Cookbook 6 [7]). The input and output signals are entered in table "A2.4 IO list", each with variable names and hardware/network addresses. External content can also be copied and pasted into all tables via the clipboard.

The catalog of error-preventing measures and the programming rules can be selected and adapted in table "A3 Measures". The tables "A3 Measures" and "A4 Requirements" should already be pre-assigned in the project template. The list of required function blocks for inputs stage and outputs stage is based on the safety functions, the peripheral hardware and the I/O list. These should be managed in table "B3 Modular architecture".

With these preparations, the table "B4 Matrix C+E" can be filled in. This is done using the buttons for automatic updating for I/O signals, modules and safety functions. The actual software specification is then carried out by:

- Assigning input signals to the individual safety functions and
- enter the logical link between the input signals and the switching operations of the output signals.

The second point is required for coding the processing stage. A specialized editor helps with this connections. The compact representation in table "B4 Matrix compact" helps with large projects. You create this table simply by using the update function, which automatically converts the "B4 Matrix C+E" table. At this point at the latest, all available functions for formal

verification of the tables mentioned should have been used to detect and correct omissions, duplicates and contradictions.

After verification of all input documents and the specification described above, the program can be coded in the language selected for the control system. The code must also be verified. This process is documented in detail in various tables and summarized in "C1 Codereview". The program is then validated, which is also documented individually in various tables and summarized in table "D1 Validation". The questions in tables C1 and D1 can be adapted and supplemented as required. Persons who subsequently check the project can also document and comment on their activities.

If the safety functions, I/O signals or function blocks are modified, the changes from tables A1, A2.4 and B3 are automatically updated in the specification tables and revised manually. All modifications are initially marked in color (yellow). The markings are deleted manually once the new coding, verification and validation of these modifications has been completed.

3.3 Where is SOFTEMA available?

The SOFTEMA tool is offered as freeware for use free of charge on the IFA website. Current information on the development status, beta versions and the link to the download are available at <https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-softema/index.jsp>.

3.4 Installation and versioning

SOFTEMA is installed using the installation program supplied. SOFTEMA project files are provided with a four-digit version number x.y.e.g.. They originate from the SOFTEMA program with which they were created or modified. Further information on these topics can be found in the SOFTEMA Cookbook 1, the information "SOFTEMA - Getting Started" and in the SOFTEMA FAQ.

3.5 Interfaces to SOFTEMA

In the current version of SOFTEMA, no interfaces are implemented. It makes more sense to implement interfaces directly to the Excel file in order to be able to import and export dates.

3.6 The role concept

SOFTEMA defines various roles for the persons involved in the project. Each person involved should select the appropriate role for the task. Each role has authorizations defined in the tables. This prevents unauthorized and unintended changes. For example, a person in the Check1 role can only read and none of the table contents can be changed. Only the `_Comment_Check` column and its protocol fields (Check1, Date) can be edited in this role. The current definition of roles is described in SOFTEMA Cookbook 1, Section 4.10.

3.7 The user administration

The current description of user administration can be found in SOFTEMA Cookbook 1, Section 4.11.

4 Preparation of project planning with SOFTEMA

This chapter describes which documents and information must be available before project planning with SOFTEMA can begin.

4.1 Specification of the safety requirements

A very basic prerequisite for programming safety functions is the existence of the safety requirements for a projected machine. A general outline for a specification of the safety requirements is proposed in IFA Report 2/2017 [8], Box 6.1. These safety requirements also include the list of safety functions introduced in Chapter 5 of the same report. The definition of safety functions is described in detail in SISTEMA Cookbook 6 [7]. All safety functions, including their properties, must be available so that they can be entered in the "A1 Safety functions" table in the project file.

Individual texts or a selection of several cells can be inserted into the table via the Windows clipboard. Importing a complete list via a data interface is not currently implemented in SOFTEMA.

4.2 List of signals and function blocks

The control hardware is specified and projected on the basis of the functional and safety requirements. This also provides important information for project planning with SOFTEMA.

The safety-related input and output signals must also be available for the specification of safety functions. These must be entered in table "A2.4 IO list".

In addition, the function blocks used for the inputs stage and outputs stage, whether developed by the manufacturer or in-house, can be entered in the table "B3 Modular architecture".

Individual texts or a selection of several cells can be inserted into the table via the Windows clipboard. The import of these lists via a data interface is currently not implemented in SOFTEMA.

4.3 Error-preventing/error-controlling measures

Prior to project planning, it is necessary to determine which fault avoidance and fault control measures are to be applied. Further information on important measures can be found in IFA Report 2/2016, Chapter 5. These measures, other project planning rules and technical features must be entered in the project template file in the table "A3 Measures" (Section 5.5.5).

5 Setting up a project file

This chapter describes how to configure SOFTEMA and the project template file used.

5.1 Determine project organization

The following must be clarified before setting up a project file: Which persons work on which projects or applications and in which role? Project leads play an important role in the preparation of a SOFTEMA project, as they are generally responsible for project quality and the implementation of measures to prevent software errors. This includes defining requirements and error-preventing measures (Section 5.5.5) and the design of suitable test protocols (Sections 5.5.10 and 5.5.11).

5.2 Selecting and setting up the project directory

The project directory must first be defined. It can be located on a local or networked drive. Full read and write permissions must be available. One or more project files for one or more control program(s) (e.g. for a machine) are edited in this directory. All these project files are configured in the same way by the INI-files in the project directory (see Section 5.3). Documents should also be saved in a subdirectory of the project directory. In this way, the project directory can be archived with all associated files using an archiving tool.

5.3 Setting options for SOFTEMA

Possible editing options are described in SOFTEMA Cookbook 1, Chapter 9 "Options". SOFTEMA can read in "project-related" options via various initialization files (INI-files for short). These INI-files affect all project files in the same directory.

There are also "SOFTEMA-related" options: They affect the general behavior of SOFTEMA for all project files (regardless of their location) and are set via the menu bar with the command EXTRAS → OPTIONS. These global settings are stored by SOFTEMA in the Windows registry (of the user) - i.e. there are none initialization files for these options!

5.4 Selecting a suitable project template

SOFTEMA cannot create new project files. Instead, a pre prepared project file is used as a template and opened in SOFTEMA. To test SOFTEMA, you can start directly with the installed standard template file `__SOFTEMA_Template__.xlxsx`. Based on this, it will be necessary and useful to extend this standard template file in order to obtain a company or project specific project template. The regular maintenance and use of such a customized project template will certainly increase the acceptance and efficiency of the matrix method with SOFTEMA. In the following Section 5.5 the steps for customizing the template are described. If a customized project template already exists, this section can be skipped.

5.5 Customizing the project template

Starting from the standard template file `__SOFTEMA_Template__.xlsx`, some tables can be edited directly in the SOFTEMA user interface, although these tables may be easier to customise with Microsoft Excel. Both methods are possible.

Other tables in the project template can only be edited using Microsoft Excel. This applies in particular to the tables with measures and requirements as well as the log tables.

Note: Individual columns and rows can also be added to all tables (see SOFTEMA Cookbook 1, Chapter 5 "Description of the data format for SOFTEMA"). This is usually only possible with Microsoft Excel. In addition, notes, logos etc. can be entered in the table areas above the start control character or below the end control character. However, after editing with Excel, the checksum of the project file no longer matches and an error message appears when opening with SOFTEMA. You can still continue working and the checksum will be set correctly the next time you save.

5.5.1 Customizing the "Persons" table or creating the user administration

First of all, the person responsible for customizing the SOFTEMA template should create a user administration, as otherwise it will not be possible to edit various tables in SOFTEMA. This step must be carried out in SOFTEMA, as "sheet protection" is active for the table to protect it from manipulation.

Note: The "Persons" worksheet is used by SOFTEMA for user administration and must not be edited with Excel!

The SOFTEMA template initially only has one user, the "Admin". Its default password at the beginning is "admin" and must be changed after the first login!

A detailed description of user administration can be found in SOFTEMA Cookbook 1, Chapter 4.11 "User administration".

To customize the SOFTEMA template, it is recommended to create a user with superuser rights. This user can edit all tables and, if necessary, delete them at the end of editing.

5.5.2 Customizing the "Project" table

In Microsoft Excel, additional rows P17 to P20 and P24 to Pxy can be added with their own descriptions (see SOFTEMA Cookbook 1, Section "Project tab"). Additional comment rows (without a row identifier for the `_No` column) are also possible.

5.5.3 Customizing the "A1 Safety functions" table

With Microsoft Excel, comment rows (without the row identifier of the `_No.` column) can already be added in the project template to provide a clear structuring, e.g. "Automatic

mode", "Setup mode" or similar. However, this can also be done later via the context menu in SOFTEMA.

5.5.4 Customizing the "A2.4 IO list" table

With Microsoft Excel, comment rows can already be added to the project template (without the rows identifier of the `_Nr` column), e.g. "Input signals", "Output signals" or similar, to provide a clear structure. However, this can also be done later via the context menu in SOFTEMA.

5.5.5 Customizing the "A3 Measures" table

This table documents all measures taken (programming rules, tools, conventions, error detection and control, etc.) for this programming project. Microsoft Excel can be used to add, edit or delete rows with the individual measures. Each measure must then be uniquely named by a row identifier of the form `Rx` in the column `_No`. Additional comment rows (without a row identifier in the `_No` column) help to structure the measures and provide them with headings. The table in the standard template file serves only as an example and must not be used unchanged.

During subsequent project planning with SOFTEMA, the "Project lead" role can deactivate (`_Active` column) and change individual measures.

5.5.6 Customizing the "A4 Requirements" table

This table documents which normative requirements are relevant for the project. Microsoft Excel can be used to add, edit or delete rows with the individual requirements. Each requirement must then be uniquely named by a row identifier of the form `Ax` in the `_No` column. Additional comment rows (without row identifiers in the `_No` column) help to structure the requirements and provide them with headings. The table of the standard template file can serve as an example, as it contains typical requirements of DIN EN ISO 13849-1:2016 [1], Section 4.6, for application software.

During subsequent project planning with SOFTEMA, individual requirements can be deactivated (`_Active` column), but not changed.

5.5.7 Customizing the "B3 Modular architecture" table

With Microsoft Excel, comment rows (without the row identifier of the `_No`. column) can already be added in the project template, e.g. "Input modules", "Output modules" or similar. However, this can also be done later via the context menu in SOFTEMA. Furthermore, typical, frequently used function blocks can be entered in the project template in advance or read in via the initialization file `SOFTEMA_FB.INI` (see SOFTEMA Cookbook 1, Chapter 9 "Options").

5.5.8 Customizing the "Persons" table or creating new team members

SOFTEMA's user management function allows all internal and external persons typically involved to be entered with their contact details in the template file. Users can be edited/added/deleted at any time via the user administration. Which of these people in which role involved in a particular project is also administered via the user administration of SOFTEMA.

5.5.9 Customizing the "Documents" table

Documents that may be important for each specific project (guidelines, standards, instructions, programming rules, etc.) can already be entered in this table in the template file. Each document must then be uniquely named by a line identifier of the form Dx in the _No column. During the project, the table can still be edited via the context menu in SOFTEMA and extended to include project specific documents.

5.5.10 Customizing the "C1 Codereview" log table

This table contains questions for the code review. The contents of the standard template file can be used as an example, but can also be changed in the _Description column and supplemented with your own specific questions or aspects. Further details can be found in SOFTEMA Cookbook 1, Section 8.1.10 "Tab: C1 Codereview". Each protocol line must then be uniquely named by a line identifier of the form Rx in the _No column. During the subsequent project planning with SOFTEMA, individual protocol questions can no longer be changed via the context menu. This can only be done in Excel.

5.5.11 Customizing the "D1 Validation" log table

This table contains questions for the final validation, whereby these are presented in two tables areas (above for validation steps and below for project documentation). The contents of the standard template file can be used as an example, but can also be changed in the _Description column and supplemented with your own specific questions or aspects. For further details, please refer to SOFTEMA Cookbook 1, Section 8.1.11 "Tab: D1 Validation". Each protocol line must then be uniquely named by a line identifier of the form Vx (for questions/aspects of validation) or Dx (for questions/aspects of documentation) in the _No column. During subsequent project planning with SOFTEMA, individual protocol questions can no longer be changed via the context menu, but only in Excel.

5.5.12 Customizing the other tables

There are other tables that are not adjusted in the template file. Nevertheless, the control characters ("\$\$\$\$") and the column identifiers must be entered in all tables, otherwise the file cannot be opened with SOFTEMA. In detail, these are:

- The "B4 Matrix C+E" table is generated automatically by SOFTEMA based on the project specifications and is updated by the SOFTEMA users. Content lines with the line identifier Cx can therefore not be entered in advance. The table is empty in the project template.
- The "Changes" table is empty in the project template and is filled by the SOFTEMA users during the project.
- The tables "B4 Matrix compact" and "Protocol" are completely generated and updated by SOFTEMA. Therefore, these tables cannot and must not be customized beforehand and do not contain any content lines with the respective line identifier in the template file.

5.5.13 Optional project tables

When preparing a project-specific template file, further optional tables can be added as Excel worksheets. These tables with texts and graphics can be loaded and displayed by SOFTEMA in a separate tab (see SOFTEMA Cookbook 1, Section 8.1.16 "Tab: Loading table"), but cannot be edited there.

6 Entering the project information and dates

Before SOFTEMA project planning begins, there is typically already a range of information and data is usually already available from the risk assessment, project and hardware design. This chapter describes how this information and data should or must be entered in the project template file used (Section 5). This should usually be done via the SOFTEMA user interface. However, for larger amounts of data, the project file can be edited using Microsoft Excel, taking care to format the tables correctly.

6.1 Entering project information

In the SOFTEMA Cookbook 1, the Chapter 8 "Structure and functions of the tabs and tables" describes the PROJECT tab with the column and row identifiers. Some rows are filled by SOFTEMA. The following rows (Table 1) should be entered manually at the start of the project:

Table 1: Project information

Designation	Description
project name project status project version project number customer contractor	The project name and project number will typically remain constant over the course of the project, while the project status should be adjusted depending on progress. The selection for the project status is configurable. The project version may need to be adjusted according to the progress of the project and after modifications.
project lead Projecting commissioning validate Check1 Check2	The names of the persons with these roles can be edited manually or selected from the drop-down list. The list evaluates the "Persons" table, which should have been filled in beforehand. This allows the persons involved and their degrees of independence [4] to be documented.
plant/machine documentation document	These lines can be used to describe the projected plant/machine, e.g. using a linked document such as a project specification.
Project-specific lines	Additional project-specific rows can be added to the "Project" table (see SOFTEMA Cookbook 1). The descriptions must also be filled in accordingly.

6.2 Defining safety functions

The "Tab: A1 Safety functions" is described in section 8.1.3 in the SOFTEMA Cookbook 1. The safety functions are usually defined as part of the risk assessment [7]. These should then be entered as completely as possible in the tab and serve as a guideline for the software specification to be carried out with SOFTEMA. There are two ways of doing this, which are described in detail in SOFTEMA Cookbook 1:

- The function name can be entered directly in the `_Description` column and supplemented by the columns `_PLr` to `_Operating mode` or
- It is generated after configuring the functions in the columns `_Protection` to `Bx` and is also supplemented by the columns `_PLr` to `_Operating mode`.

To enable clear traceability throughout the entire machine/plant project planning, each safety function should be assigned an Safety Function Identifier (SFK) and referenced in other project documents if necessary.

The commands in the context menu can be used to create new table rows (safety functions) or copy and customize existing rows. The line identifiers of deleted lines are not reused.

6.3 Entering input and output signals

The "Tab: A2.4 IO list" is described in the SOFTEMA Cookbook 1 (Section 8.1.4). The input and output signals of the programmed controller with their addresses, variable symbols and designations are usually defined as part of the hardware configuration. They should then be entered as completely as possible in this tab - as a default for the software specification to be carried out with SOFTEMA.

There are two ways to do this in the current SOFTEMA version:

- Direct entry of the signals in the columns `_Description` to `_Module` or
- Copy these column contents from an existing external signal list to the Windows clipboard and paste them into previously created empty signal rows in this table.

Import functionalities for common data formats are planned for future versions.

The "`_Module`" column is a special feature. If an input signal or the output signal is processed via a function block or module, this module can be specified here. Only function blocks or modules that have already been created in the "B3 Modular architecture" table can be selected here.

If a module is defined, the function module name, e.g. `[DOOR_SG1]`, is displayed in the square brackets for this signal in table "B4 Matrix C+E" instead of the address, e.g. `[E8.4]`.

The "`_Active in C+E`" column can help to keep an overview in the "B4 Matrix C+E" table. When the "B4 Matrix C+E" table is updated, all input and output signals are automatically added to the matrix. If there are signals that are not required for the matrix - e.g. because

they are only processed by a function block itself - they can be deactivated here. Deactivated signals are now (during an update) not added to the C&E matrix or marked as removed.

6.4 Configuring tables for measures and requirements

The "Tab: A3 Measures" and the "Tab: A4 Requirements" are described in the SOFTEMA Cookbook 1 in the sections 8.1.5 and 8.1.6. The contents of these two tables are already defined in the project template (Section 5.5). Table A4 cannot be edited in SOFTEMA. However, individual measures and requirements can be deactivated (column `_Active`), as a result of which they then remain unconsidered during project planning and validation. In such cases, a reason for deactivation should be entered in the comment column.

6.5 Entering modular architecture

The "Tab: B3 Modular architecture" is described in the SOFTEMA Cookbook 1 (Section 8.1.7). In order to document the project completely, the function blocks used should be entered with all their instances and supplementary information. This is part of configuration management, among other things. For later editing of the logic cells in table "B4 Matrix C+E", it is at least necessary to enter the instance name and the safety-related enable signal (only one output!) of these function blocks (columns `_Instance name` and `_Outputs`).

6.6 Adding to the "Persons" table via user administration

The PROJECT TEAM form is described in SOFTEMA Cookbook 1 (Section 8.1.14). The persons already entered in the template file can be updated and added at the start of the project or later via the user administration.

6.7 Adding to the "Documents" table

The "Documents tab" is described in the SOFTEMA Cookbook 1. The "Standard documents" already entered in the template file can be supplemented with project-specific documents, descriptions, literature, etc. at the start of the project or later.

7 Software design

Once all the information and dates available for project planning have been entered into the prepared project template, SOFTEMA project planning can begin. This chapter describes the steps for the software design and the resulting software specification as well as the test and validation plans. In order to be able to detect specification errors before these steps, all previously completed tables should already be checked for correctness and completeness using the "Formal checks" tab function (available in "A1 Safety functions" and "2.4 IO list"). Otherwise, the software specification would have to be updated again later.

Note: The functions "Update/Create new table" and "Formal checks", which can be called up in several tabs, are not carried out automatically in the current SOFTEMA version when the input data is changed, but must always be triggered manually by the SOFTEMA user.

7.1 Updating and adding the "B4 Matrix C&E" table

The software specification is presented in the form of the tables "B4 Matrix C&E" and "B4 Matrix compact", the latter of which is generated entirely by SOFTEMA (Section 7.2). The "Tab: B4 Matrix C+E" is described in detail in SOFTEMA Cookbook 1 (Section 8.1.8).

The "Update table" button is used to transfer all input data from tables A1, A2.4 and B3 or to update new or changed input data in the "B4 Matrix C&E". Safety functions are entered as rows C_x and IO signals as columns I_x or O_x.

If a row has been added or updated, the cell in the `_Test` column must be customized. This cell indicates the state/cause of the system from which this row is to be tested. This is usually the cause C0, in which all safety functions are either deactivated or not requested. A safety function or a test case is usually requested from this state/cause.

If input data has been updated or new data has been created, all affected rows/columns are marked *yellow* in the table. The user must then check these manually, add to them and delete the markings themselves.

If, on the other hand, input data has been deleted, the corresponding row/column headers are marked *red* for deletion after the table has been updated. The user must then delete these manually.

Changes that affect the contents of the logic cells (e.g. deletion of input data) are indicated in the logic cells by note texts. As a remedy, these cells must be edited again.

7.1.1 Specifying software for safety functions

In the right-hand section of the table "B4 Matrix C&E", in the O_x columns, users can specify the boolean logic operations for the logic module (processing stage) to implement the safety

functions. These specifications are then the template for the subsequent coding of the processing stage.

Each intersection of a safety function or a test case with a column Ox is identified as a logic cell. Each logic cell is edited individually using the logic editor (see SOFTEMA Cookbook 1, section 8.2 "The logic editor"). In the first row of the logic cells, you enter how this output variable Ox is to be controlled when the safety function/test case in question is required.

There are three control alternatives, which also determine the background color of the cell and the logic editor:

- OFF (red background) means: Output variable is controlled with 0/False (typical for wired contactor or valve controls)
- NOP (white) means: This output variable is none activated.
- ON (green) means: Output variable is controlled with 1/True (**can only be selected for test functions**)

The switching conditions are entered in the lines of the logic cell below, according to which the output variable Ox is to be controlled.

7.1.2 How to build the software specification

The steps for the software specification are:

Step 1: For each individual safety function that triggers a switching operation for an actuator, enter the logical link of the input variables from the logic module that triggers the switching operation in the corresponding logic cell of the table. The switching operation is indicated by "OFF". If a safety function of an actuator triggers none of the switching operations, enter "NOP".

An example of a logic cell shows Figure 3 with the safety function SF1 and how it acts on the output QS_M1:

The first line of the cell contains "OFF" and the input variable "EMST_OK" below it. Due to the negative logic of the input variables, this entry should be read as follows: "If the variable EMST_OK = FALSE, then output QS_M1 should be FALSE", i.e. the output is switched OFF.

In the instructions of the Figure 3 the input variable "EMST_OK" has been supplemented: by a comment with the line identifier "IM1" and the instance name of the corresponding input function block/module "Not_Halt_S1" (from "B3 Modular architecture" table). These additions to the instructions are configurable (see SOFTEMA Cookbook 1, Chapter 9 "Options").

O1: QS_M1 [A24.0] Contactors Motor M1 (1K1, 1K2) OFF (*IM1*) S_ESTOP_S1_OK
SF1 (1): -SF10.1 If emergency stop EMST, then Motor M1 switch off, Motor M2 in STO, Motor M3 switch off, with acknowledge button ACK acknowledge.

Figure 3: Example 1 for the specification of a logic cell

In Figure 4 shows an example of a logic cell for the safety function SF4, which again acts on the output QS_M1:

O1: QS_M1 [A24.0] Contactors Motor M1 (1K1, 1K2) OFF (*IM3*) S_D00R_SG2_OK OR (*IM4*) S_D00R_SG3_OK
SF4 (2): -SF11.3.1 If guard doors SG2_SG3, then Motor M1 switch off, with acknowledge button ACK acknowledge.

Figure 4: Example 2 for the specification of a logic cell

For safety function SF4, the OFF-switching operation is triggered precisely when both safety gates SG2 *and* SG3 are open. Therefore, due to the negative logic of the input variables, the OR link "SG2_OK or SG3_OK" must be entered. This reads: "If the expression (SG2_OK OR SG3_OK) = FALSE, then output QS_M1 should be FALSE."

In the instructions of the Figure 4 the input variables have been supplemented: by a comment with the line identifiers "IM3" or "IM4" and the instance names of the corresponding input function blocks "Fast moving gate_SG2" or "Vertical guard door_SG3".

Step 2: The complete logic operation per actuator (i.e. across all safety functions) is then derived from the AND operation of the logic cell instructions in the actuator column.

Figure 5 shows a section of the resulting specification for the exemplary logic module as a combination of the switching instructions from Figure 3 and Figure 4.


```

(* Contactors Motor M1 (1K1, 1K2) [A24.0] *)
QS_M1 := (*Operating mode B0: All*)
  (*PRI01*)
  (*IM1*) S_ESTOP_S1_OK
  AND
  (
    (*Operating mode B1: Automatic*)
    (*PRI02*)
    (
      (*IM3*) S_DOOR_SG2_OK
      OR
      (*IM4*) S_DOOR_SG3_OK
    )
  );

```

Figure 5: Example 3 for the resulting specification for an actuator

Note: The generic structure of the specification for the logic module is described in IFA Report 2/2016 [4], section 6.6 and in Figure 16. The Boolean linking of several switching conditions of different priorities and modes of operation is shown there. However, if switching conditions of an operating mode are combined with an AND function (such as the blue blocks in Figure 16), the output of this AND function must also be deactivated (= FALSE) if the mode of operation is inactive. For this purpose, the enable signal of the mode of operation (in positive logic) is linked to the AND function, for example.

7.1.3 Adding test cases

While the safety functions are only updated by SOFTEMA in the matrix, the test cases must be entered manually in a suitable position via the context menu INSERT ROW → INSERT TEST ROW BELOW and then edited. In such a test row, all cells must be edited manually, including a unique description TF_n in the _SF-No column (n = consecutive number of test cases) and the priority of the test case in the _Prio column. A constellation of the input signals can be set in columns I_x and the expected switching behavior OFF/ON/NOP (without further switching conditions!) is entered in columns O_x in the logic editor. This test case must then be described in the _SF name column.

7.2 Updating the "B4 Matrix compact" table

This table represents a compact, quasi transposed form of the "B4 Matrix C+E", i.e. the roles of the rows and columns are reversed. The rows each describe an actuator/output and its control by inputs and safety functions in compact form. This is a practical form of display for testing the application program or for very large amounts of data. The table can be PRINTED individually via the PRINT menu.

The "Tab: B4 Matrix compact" is described in detail in SOFTEMA Cookbook 1 (Section 8.1.9). The user can use the buttons to delete, create new or update the table (e.g. for

modifications). Updated cells are marked. However, this marking may be written/deleted again during the subsequent update.

7.3 Verification and validation plans

When the aforementioned tables for the specification of the logic module were designed, the cells for the verification and validation (V&V for short) required after coding were also created in the `_Verification` and `_Validation` columns. In these columns, the V&V performed is indicated by selecting the texts "not OK" or "OK".

Note: Spreadsheets that have the `_Lock` column can only be verified and validated if the lock has been set.

If you also want to enter a V&V information/criterion for each of the rows in the tables "B4 Matrix ...", columns with "project-specific column identifiers using predefined prefixes" (see SOFTEMA Cookbook 1 Section 5.4) such as a `#CO_ValiCriterion` column to the right of the `_Validation` column should already have been added in the project template file.

The following therefore applies to the application of the matrix:

- Programming persons read the corresponding column of the C&E matrix for each actuator with regard to the entries of the logic cells in order to code the logic.
- Testing persons read the rows of the matrix to test individual safety functions.

8 Coding of the application program

In the next step of project planning, the specification of the table "B4 Matrix C+E" must be converted into the code of the logic module (processing stage) for the application program. Several approved programming languages, which are offered by the development environments of the control systems, are possible. "Structured text (ST)" is used as the example language in this guide. This is also due to the fact that the logic editor in the specification generates a specification text corresponding to ST (see also 7.1.2 How to build the software specification)

Note: With the SOFTEMA Code Generator it is possible to automatically generate the code of the logic module (processing stage) from the logic cells.

8.1 Measures as part of the tool qualification

In SOFTEMA, users must take measures to detect errors that are likely to be contained and thus potentially incorrect specifications. This includes

- Review of the specifications generated with SOFTEMA, in particular in the tables "B3 Modular architecture", "B4 Matrix C&E" and "B4 Matrix compact". The specifications are compared with the specified safety requirements.
- Module test of the program parts coded with these specifications to ensure that the specified safety requirements are met.
- (Extended) functional test of the complete application program to ensure that the specified safety requirements are met.

As part of the verifications (Chapter 9) and validations (Chapter 10), the aforementioned measures are carried out and confirmed.

9 Verification of the coded program

This chapter describes the steps for verification of the coded program against the specifications and project data. After verification, the protocol fields must be filled in or updated with the date, name and signature of the verified program version.

The persons who carry out verifications should be independent of the persons who created the verified dates. An orientation on which degrees of independence should be observed can be found in IFA Report 2/2016 [4], Section 5.15.

The verifications described below are generally carried out as visual checks and each contain two questions that must be confirmed by an "OK" in the verification column:

- Have the contents of the table - before coding - been created correctly and completely?
- Have the contents of the table been implemented correctly and completely in the code?

9.1 Measures as part of the tool qualification

In SOFTEMA, users must take measures to detect the errors likely to be contained in SOFTEMA and thus a potentially incorrect specification. This includes the verifications described below.

9.2 Verification in the "A2.4 IO list" table

The `_SW-Verif.` column confirms that the signals are correctly entered with variable symbol, address and description and that the variables are also correctly connected in the program or with the associated function blocks.

The `_DIAG` test column confirms that state and error information about input/output modules, for example, is correctly connected to the associated function blocks.

9.3 Verification in the "A3 Measures" table

The `_Verification` column confirms that the measures required for the specification and coding of the program have been implemented. Measures that only affect later activities in the project (commissioning, testing, etc.) can only be verified and confirmed later.

9.4 Verification in the "B3 Modular architecture" table

The `_Verification` column confirms that all entries for the module in the row are present and correct. This includes the content required for the specification as well as the additional content requested by the project lead (see Section 6.5).

9.5 Verification in the "B4 Matrix C&E" table

The `_Verification` column confirms that all entries in the row are present and that the logic cells are filled in correctly and completely. The switching instructions must also be compared with the definition of the associated safety function or test case.

9.6 Verification in the "B4 Matrix compact" table

The `_Verification` column confirms that all entries in the row are present and that the cells are filled in correctly and completely. The totals field at the bottom of the table is not automatically mirrored in the "Code review" table in the current version of SOFTEMA. However, it could be entered there as an additional manual verification point (see Section 5.5.10).

9.7 Verification in the "Code review" table

In this table, the sum cells of all previously mentioned verifications (exception: verification in "B4 Matrix compact") are summarized. These cells in the `_Verification` column cannot be entered manually. The degree of fulfillment of these verifications is given as a percentage.

In addition, further verification points may have been added to the project template (see Section 5.5.10). The corresponding cells in the `_Verification` column must be filled manually ("OK", "not OK") in order to assess these verification points.

10 Validation of the application program

This chapter describes the steps for validation of the coded program and all previous steps and measures in the V-model. After a validation, the protocol fields must be filled in or updated with the date, name and signature of the validated program version.

The persons carrying out validations should be independent of the persons who created the validated dates. An orientation on which degrees of independence should be observed can be found in IFA Report 2/2016 [4], section 5.15.

10.1 Measures as part of the tool qualification

SOFTEMA users must take measures to detect errors likely to be contained in SOFTEMA and thus a potentially incorrect specification. This includes the validations described below, in particular of the specifications, as these serve as a coding template.

10.2 Validation in the "A2.4 IO list" table

In the `_Validation` column, an IO check is used to confirm that the signals are wired correctly or that communication is configured correctly for data exchange via the network.

10.3 Validation in the "B4 Matrix C&E" table

In the `_Validation` column, a black box test of the logic module and the program (additional simulation for higher PL) is used to confirm that the specified switching instructions for the safety functions are executed correctly.

10.4 Validation in the "B4 Matrix compact" table

In the `_Validation` column, a black box test of the logic module and the program (additional simulation for higher PL) confirms that the specified switching instructions are executed correctly with effect on the actuators. In the current version of SOFTEMA, the totals field at the bottom of the table is not automatically mirrored in the "Validate ration" table. However, it could be entered there as an additional manual validation point (see Section 5.5.11).

10.5 Validation in the "A1 Safety functions" table

In the `_Validation` column, function tests (extended function test for higher PL) are used to confirm that the program in the control system executes the safety functions correctly - as specified in all parameters.

10.6 Validation in the "A4 Requirements" table

In the _Validation column, it is confirmed that the normative requirements that are required for software development have been implemented. Requirements that affect later activities in the project (documentation, testing, etc.) can only be validated and confirmed in this table at a later date.

10.7 Validation in the "Validation" table

This table summarizes the sum cells of all previously mentioned validations (except : validation in "B4 Matrix compact"). These cells in the _Validation column cannot be entered manually. The degree of fulfillment of these validations is given as a percentage.

In addition, further validation points may have been added to the project template (see Section 5.5.11). The corresponding cells in the _Validation column must be filled manually ("OK", "not OK") in order to assess these validation points.

11 Checking the project file

This chapter describes how persons with the roles Check1/Check2 can check/test a project in SOFTEMA. The test can be carried out by internal and/or external persons, which is why two roles and corresponding input cells are provided.

The persons conducting the tests should be independent of the persons who prepared the tested dates. An orientation on the level of independence that should be maintained can be found in IFA Report 2/2016 [4], Section 5.15.

11.1 Visual inspection in the tables

In the Check1/Check2 roles, all tables can be viewed, but most of them cannot be edited. This visual test can therefore be used to determine the completeness, consistency and plausibility of the tables. You can get a quick overview of these activities in the "C1 Codereview" and "D1 Validation" tables. By double-clicking on the table names in the _Reference sheet column and then back using the PREVIOUS PAGE toolbar command, you can quickly navigate through the verification or validation of the tables.

The following functions that are important for a test can also be carried out by the Check1/Check2 roles:

- the FORMAL CHECKS function in various tabs
- Print functions including summary (Chapter 12)
- Open linked documents, e.g. in the "Documents" table

11.2 Comments in the tables

The _Comment_Check column is one of the fields that can be edited by the Check1/Check2 roles. In the current version of SOFTEMA, only this column is provided for both roles together. These comments can only be read by all other roles.

11.3 Protocol fields

The protocol fields for Check1/Check2 below the verification/validation columns can also be edited exclusively by these roles. A date and the name of the person performing the test can be selected there to document the test process.

12 Print functions

This chapter describes the various options for printing the table contents in SOFTEMA or exporting them to a PDF file.

12.1 Printer setup

The menu command FILE → PRINT SETUP opens a dialog, in which the printer active for SOFTEMA can be selected and configured.

12.2 Printing tables

The menu command PRINT → PRINT TABLE is used to print the table currently displayed . A dialog for the print settings opens first (Figure 6). After confirming the print settings with the OK button, a print preview is displayed. You can then print in this preview.

The `IniFile_PrintSettings` option (see INI-file) assigns the name of the file in which the print settings (SETTINGS tab in Figure 6) can be saved. The print settings are automatically loaded from this file when the dialog is opened.

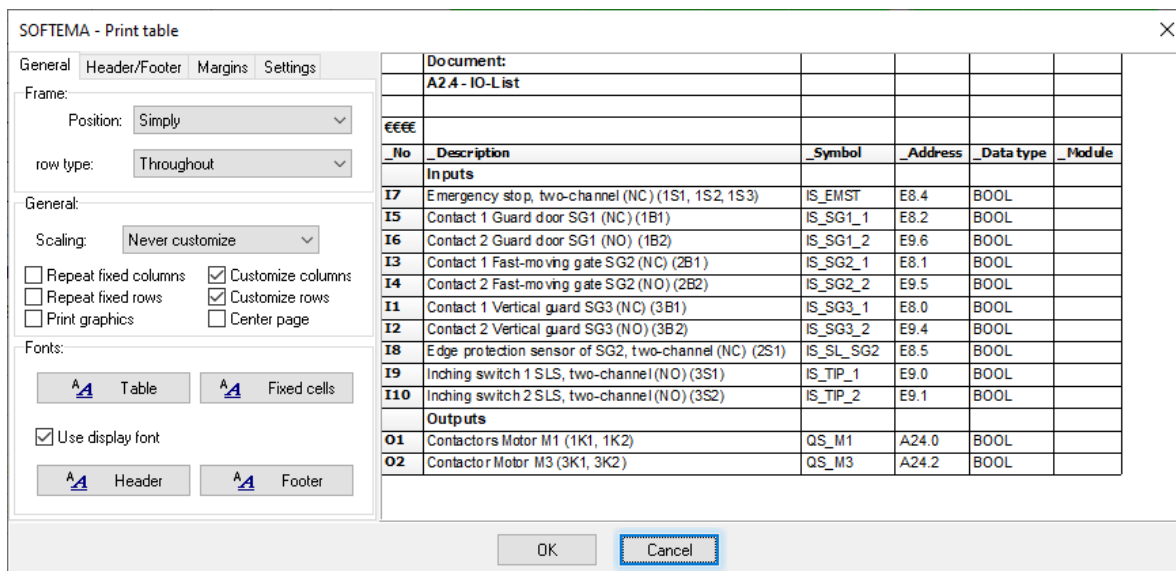


Figure 6: Dialog for print settings

12.3 Creating a summary

The menu command PRINT → SUMMARY is used to create a fully permanent project summary with the contents of *all* tables in the project. First, a dialog opens for the options (Figure 7). Here you can control the output directly to the printer, to a file or to a preview (default setting).

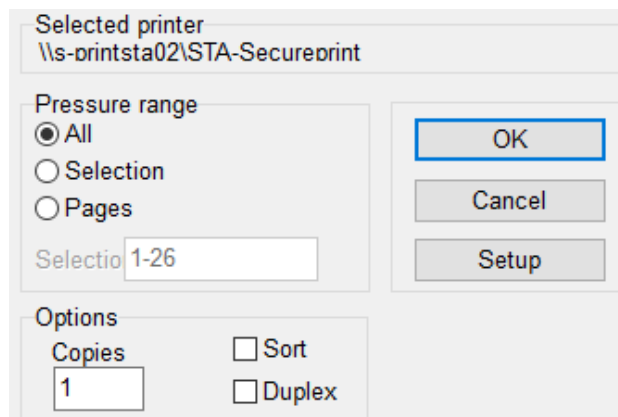


Figure 7: Options for the summary to be created

The summary is then generated based on the current table contents. In the preview (Figure 8), the summary can then be viewed, printed or saved as a PDF.

The project must be saved beforehand, whereby the checksum is recalculated. This checksum is displayed in the header on every page of the summary. This checksum and the date of the last change can be used to check whether an existing summary and a loaded project match.

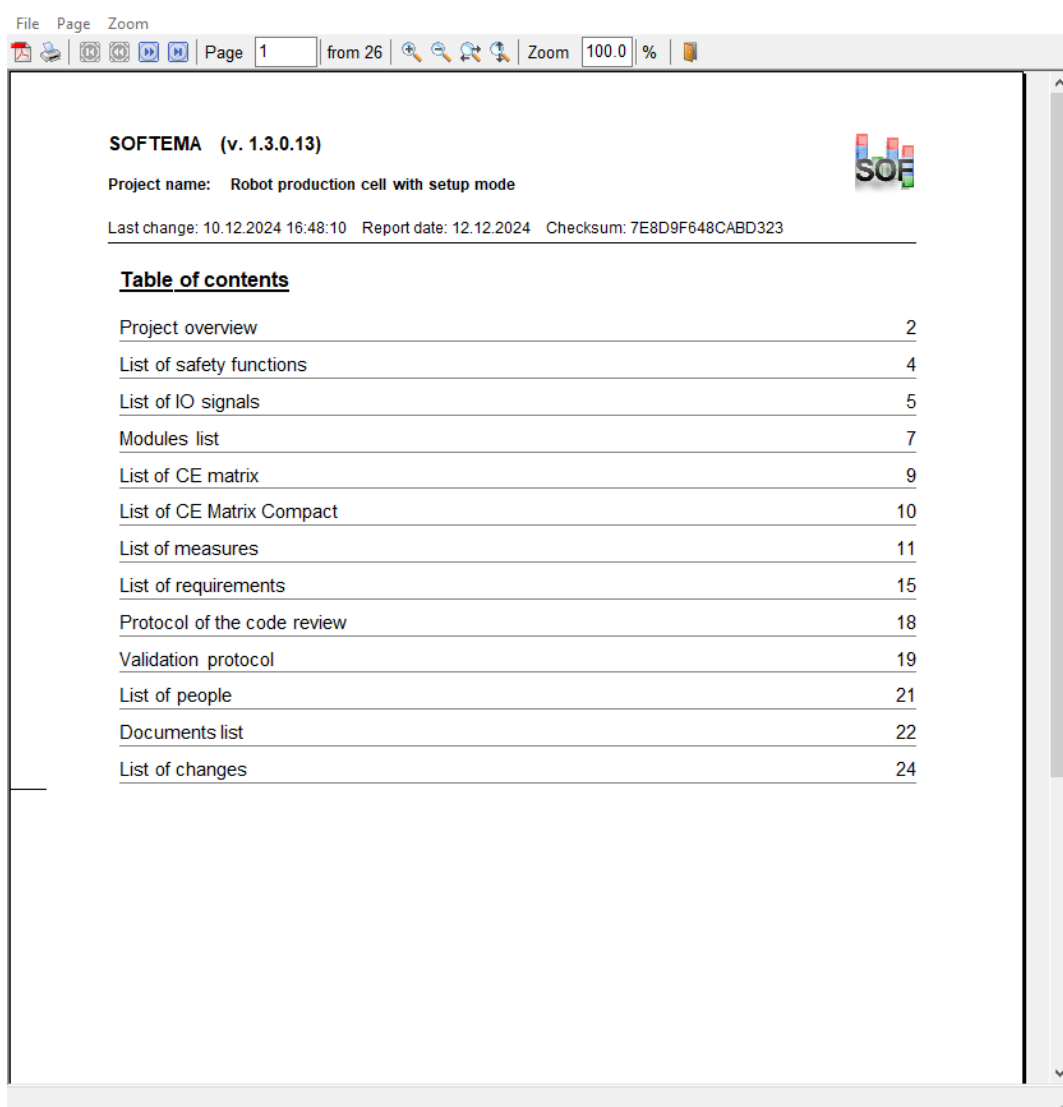


Figure 8: Preview of the summary (1st page: table of contents)

From the second page of the summary, the information of the "Project" table is shown above and the overview of the tables below (Figure 9) is displayed. For each table, this overview shows the quantity of content rows (e.g. quantity of safety functions or quantity of measures) and the degree of fulfillment of the verifications/validations from 0 % to 100 %.

<u>Tables overview</u>			
Safety functions:	Quantity: 7		0 % validated
IO signals:	Quantity: 14	100 % verified	100 % validated
Measures:	Quantity: 44	100 % verified	
Requirements:	Quantity: 36		100 % validated

Figure 9: Preview of the summary (from 2nd page: overview of the tables)

On the following pages of the summary, all tables are presented with an overview and below that the details of all content lines, here as an example the beginning part of the "List of IO

signals" (Figure 10).

List of IO signals			
Number of rows: 14	Number verified: 14 (100 %)		
Verify Date/Name: 04.03.2014 / Johanna Dietz			
Veri.Check1 Date/Name: <empty> / <empty>			
Veri.Check2 Date/Name: <empty> / <empty>			
Veri.Signature Program: 1272993002			
I7	Emergency stop, two-channel (NC) (1S1, 1S2, 1S3)		
	Symbol: IS_EMST	Address: E8.4	
	Status: Active		Verification: OK
I5	Contact 1 Guard door SG1 (NC) (1B1)		
	Symbol: IS_SG1_1	Address: E8.2	
	Status: Active		Verification: OK
I6	Contact 2 Guard door SG1 (NO) (1B2)		
	Symbol: IS_SG1_2	Address: E9.6	
	Status: Active		Verification: OK

Figure 10: Preview of the summary (list of IO signals)

The overview shows:

- The number of all rows and the number of verified rows and below
- date and names of all persons involved in the verification or validation process.

In the details, the identifiers are shown as follows:

- In red color if the verification or validation is missing or "not OK" for a content line (last input in Figure 10).
- Crossed out if a content line is "not active" (last measure in Figure 11).

Comment Development of own function blocks:

ME R19: Own reusable function blocks are developed and documented separately according to the V-model.		
Status: Active		Verification: OK
ME R20: The function blocks are tested completely in the simulation.		
Status: Active		Verification: OK
ME R21: Library management must be maintained for your own function blocks.		
Status: Active		Verification: OK
ME R22: Provide your own function blocks with code protection.		
Status: Nicht Aktiv		Verification: <empty>

Figure 11: Preview of the summary (list of measures)

13 Application program documentation

The documentation for an application program projected with SOFTEMA should consist of the following parts:

- the SOFTEMA project file together with the associated INI-files and other configuration files
- the summary and the table printouts or PDF files (see Chapter 12)
- the documents that were linked in the project file (especially in the "Documents" table)
- the documents named in the "Validation" table

If the above files and documents have been stored in a project directory with sub directories, then archiving this project directory with an external tool in an archive file format (e.g. ZIP, RAR, TAR) is quite simple. Such an archive file can then be passed on to different persons who want to check this project, for example.

None of the built-in functions for archiving are available in the current SOFTEMA version.

In addition, all documents that can be generated by the development environment of the application program (e.g. listing of the code, symbol tables, cross-reference lists) must be archived.

It must be possible to assign the archived documents to a version of the executable application program.

14 Modification of the application program

This chapter is dedicated to the question of how modifications to the application program can be supported in SOFTEMA. Depending on the type of modification, some or all of the phases of the V-model must be run through again. The project version and project status should also be adjusted in the "Project" table. Notes on the modifications can be stored in the comment fields, in project-specific columns or in the "Changes" table. The project group should have defined a process for handling modifications.

14.1 Modification of the project data

If safety requirements, safety functions or the control hardware change or if program errors are corrected, there will be changed or new dates in the rows of tables A1, A2.4 and A3:

- Rows to be changed must be unlocked, whereupon the verification/validation cells are deleted. Changed cells should be marked manually to draw attention to these changes (Figure 12, input I13, column `_Address`).
- New rows are automatically marked and the verification/validation cells are initially empty (Figure 12, input I15).


No	Description	Symbol 	Address	Data type	Module	Active in C+E	Active	Lock	SW-Verif.	IO test	_DIAG test
	Inputs										
I7	Emergency stop, two-channel (NC) (1S1, 1S2, 1S3)	IS_EMST	E8.4	BOOL		Active	Active	x	OK	OK	OK
I5	Contact 1 Guard door SG1 (NC) (1B1)	IS_SG1_1	E8.2	BOOL		Active	Active	x	OK	OK	OK
I6	Contact 2 Guard door SG1 (NO) (1B2)	IS_SG1_2	E9.6	BOOL		Active	Active	x	OK	OK	OK
I3	Contact 1 Fast-moving gate SG2 (NC) (2B1)	IS_SG2_1	E8.1	BOOL		Active	Active	x	OK	OK	OK

Figure 12: Changed/new rows for modifications

This indicates in the tables which rows need to be verified/validated again after a modification.

All modifications should be documented in the "Modifications" table.

14.2 Updating the specification tables

After the project data has been updated, the specifications in the "B4 Matrix C+E" and "B4 Matrix compact" tables are no longer up-to-date. Therefore, the update of these tables must now be triggered manually (Sections 7.1 and 7.2). The updated cells are marked and the logic cells may have to be changed. This is only possible if the affected rows are unlocked, which also deletes the verification/validation cells.

Additional test cases may be required due to the modifications.

All adjustments should be documented in the "Changes" table.

14.3 Verification, validation and testing of the modifications

After coding the modifications, the necessary verifications are carried out in places (Chapter 9), validations (Chapter 10) and tests (Chapter 11). The protocol fields must also be updated with the date, name and signature of the modified program version.

Once the modification has been successfully completed, the edited cells must be unmarked.

14.4 Documentation of the modifications

Because the application program has changed, a complete documentation of this new program version must be created and archived (Chapter 12 and 13).

Appendix A : Literature

- [1] EN ISO 13849-1: Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design (11/2006); and Amendment 1 to EN ISO 13849-1:2015. DIN Media, Berlin 2006/2015
- [2] DIN EN 62061: Safety of machinery - Functional safety of electrical, electronic and programmable electronic safety-related control systems (09/2013). DIN Media, Berlin 2013
- [3] Becker, N.; Eggeling, M.: Final report on DGUV project no. FF-FP0319: Norm compliant development and documentation of safety related application software in manufacturing system engineering. German Social Accident Insurance (DGUV), Berlin 2013.
https://www.dguv.de/ifa/forschung/projektverzeichnis/ff-fp_0319-2.jsp
- [4] Huelke, M.; Becker, N.; Eggeling, M.: Safety-related application software for machinery - The IFA matrix method - (IFA Report 2/2016). German Social Accident Insurance (DGUV), Berlin 2016.
<https://www.dguv.de/ifa/fachinfos/arbeiten-4-0/industrie-4-0/sicherheitsbezogene-maschinensoftware/index-2.jsp>
- [5] SOFTEMA software assistant. German Social Accident Insurance (DGUV), Berlin.
<https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-softema/index.jsp>
- [6] Software-Assistant SISTEMA. German Social Accident Insurance (DGUV), Berlin.
<https://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-sistema/index.jsp>
- [7] Definition of safety functions - What is important? (SISTEMA Cookbook 6). German Social Accident Insurance (DGUV), Berlin 2015.
https://www.dguv.de/medien/ifa/en/prasoftwa/sistema/kochbuch/sistema_cookbook6_en.pdf
- [8] Hauke, M.; Schaefer, M.; Apfeld, R.; Bömer, T.; Huelke, M. et al.: IFA-Report 2/2017e Functional safety of machine controls - Application of EN ISO 13849 -. German Social Accident Insurance (DGUV), Berlin 2017.
<https://www.dguv.de/ifa/publikationen/reports-download/reports-2017/ifa-report-2-2017/index-2.jsp>

Appendix B: List of abbreviations

Table 2 contains the abbreviations used in this cookbook.

Table 2: Abbreviations used in this cookbook

Abbreviation	Description
C&E matrix	Cause and Effect matrix; synonym: cause and effect table; cause-effect diagram
INI-file	Initialization file; normal text file with the file extension <i>.ini</i>
I/O	Input/Output
NOP	No OPeration: is a command in the C&E matrix that does nothing.
OFF	Off; a command in the C&E matrix that controls the output variable with 0/False
ON	On; a command in the C&E matrix that controls the output variable with 1/True
PL	Performance Level
PL _r	Required Performance Level
SF	Safety Function
SFK	Safety Function identifier: unique description of the SF in the project
SISTEMA	Software assistant of the IFA "Safety Integrity Software Tool for the Evaluation of Machine Applications"
SOFTEMA	Software assistant of the IFA "Safety of software on machines"
PLC(s)	Programmable logic controller(s)
F-PLC(s)	Fail-safe Programmable Logic Controller(s)
SRASW	Safety-related application software
SRESW	Safety-related embedded software
ST	PLC language: Structured Text
TF	Test case