CERTIFIKAT

No. 46 24 02

PLUS+1 Development tools, version 6.0.X

Holder / Issued for / Manufacturer
Sauer-Danfoss (Älmhult) AB, Teknikgatan 1, SE-34334 ÄLMHULT, Sweden

Product name and description
The PLUS+1 development tools, version 6.0.X are support tools used for code generation, where high level graphical symbols are transferred to source code and in addition supports the assembling, linking and downloading of the source code to the hardware and reading/writing of information from/to the hardware. The PLUS+1 Development tools, version 6.0.X, consists of two main parts, namely:

- GUIDE tool, that handles the translation of graphical symbols to executable code
- Service Tool, that handles the download of executable code to the hardware and reading/writing of information from/to the hardware

Certification
The product described above fulfils the requirements placed on support tools for use in the development of application software according to the standard IEC 61508:2010 *Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 1 and 3* within the following limitations:

- GUIDE tool, requirements are only fulfilled for graphic code and not for Structured Text code and not for externally developed C-code.
- Service Tool, requirements are fulfilled as long as it is not used as a *software on-line support tool* as defined by IEC 61508-4:2010.

The certification is based on a functional safety assessment according to IEC 61508:2010 described in SP report PX18354:A supplemented with a separate evaluation of the PLUS+1 Development tools described in SP report PX18354:B and the user documentation in the currently valid revision.

Marking
Each sample that conforms in all respects with the original item certified may display the text "Type-examined by SP". In addition, manuals and marking shall also show the number of the standard, the reached SIL (Safety Integrity Level) of the item, the number of this certificate and the serial number or equivalent of the item.

Validity
This certificate is valid until not later than 28th February 2018.

Miscellaneous
Other terms and conditions are set out in SP's certification rules for type-examination, SPCR 123. This is the first issue of this certificate.

Borås, 28 February 2013

SP Technical Research Institute of Sweden
Certification

Lennart Månsson
Certification Manager

Lennart Aronsson
Certification Officer

Certificate issued by SP

SP Technical Research Institute of Sweden

Postal address	Phone / Fax	Reg.number	E-mail / Internet
SP Box 857 SE-501 15 Borås SWEDEN	+46 10 516 50 00 +46 33 13 55 02	556464-6874	info@sp.se www.sp.se

This certificate may not be reproduced other than in full, except with the prior written approval by SP.

REPORT

Contact person
Lars Strandén
Electronics
+46 10 516 55 52
lars.stranden@sp.se

Date 2013-03-19 Reference PX18354:C

Sauer-Danfoss (Älmhult) AB
Teknikgatan 1
S-343 34 ÄLMHULT

PLUS+1 6.0 Safety Evaluation Overview

Summary

This document is public and focusses on the Sauer-Danfoss AB PLUS+1 tools from a safety perspective according to IEC 61508:2010. This document is an extract of "SP report PLUS+1 6.0 Evaluation, PX18354:B, 2013-02-28" and is intended for users and as initial information when considering Sauer-Danfoss AB PLUS+1 tools.

The PLUS+1 Development tools are two, namely:

- GUIDE tool, that handles the translation of graphic symbols to executable code.
- Service tool, that handles the download of the executable code to hardware.

These are certified according to IEC 61508:2010 under certain conditions.

This document may not be reproduced other than in full, except with the prior written approval of SP.

1 Commission

This document is public and gives an overview of the Sauer-Danfoss AB PLUS+1 tools from a safety perspective according to IEC 61508:2010. The document is assumed to match PLUS+1 version 6.0.x (where x is an integer ≥ 1) however there could be coming versions that may have effects on this document. This has to be checked by Sauer-Danfoss AB.

The purpose is to give an overview of the possible quality of a product developed using the PLUS+1 environment. In this document a product is a safety function or a part of a safety function according to IEC 61508:2010.

The work was performed by Lars Strandén, SP Electronics, from February 2013 to March 2013.

2 Client

Sauer-Danfoss (Älmhult) AB
Teknikgatan 1
S-343 34 ÄLMHULT

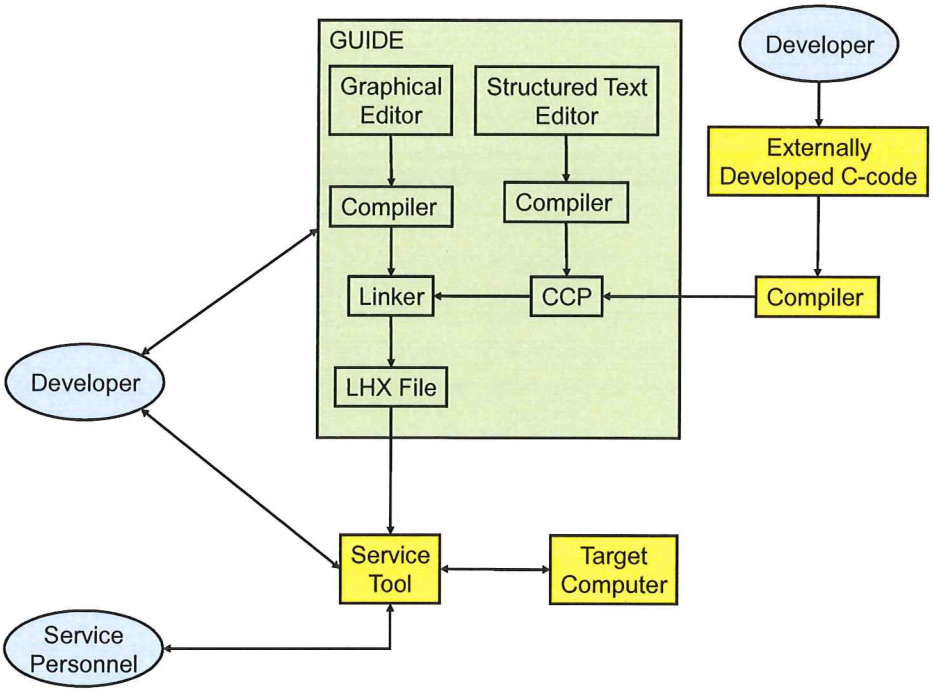
Contact person: Johan Karlsson

3 Terminology

Term	Description
CCP	Compiled Code Package used for making it possible to merge external C-code, Structured Text code and graphic code.
HWD file	HardWare Description file
IEC 61508	IEC 61508:2010 part 1-7
LHX file	Downloadable protected file
NV memory	Non Volatile memory
SD	Sauer-Danfoss AB in Älmhult. SD is here considered as a development environment provider not as a safety function provider.
SD graphic language	The SD language including graphic symbols and connections between them.
SIL application	An application according to IEC 61508 SIL i.e. a safety critical application (safety function or part of it).

4 PLUS+1 Overview

The PLUS+1 development environment is shown in the figure below.



GUIDE supports the SD graphic language and a subset of Structured Text (see IEC 61131). Both can be edited, connected and compiled in the GUIDE environment. The purpose of including Structured Text is primarily for including already written code. However, there could be cases where adaptation is needed in order to incorporate Structured Text code into GUIDE. A call is always made from the graphic GUIDE component to subroutine defined using Structured Text. Since IEC 61508 considers a single programming language at a time each of the languages is handled separately. For example, each needs a coding guidelines. Special conditions must be fulfilled if part of the application is implemented in Structured Text. In order to use both languages each is compiled and linked together.

It is also possible for a developer to handle externally developed C-code and to create a CCP to be linked with other code. However, in this case the responsibility is completely up to the developer and SD does not give any support.

Generally, responsibility is shared between SD and the safety function developer when creating safety related applications. In case of doubt, however, the final responsibility always rests with the safety function developer.

5 Safety measure examples

5.1 Parameters

Parameters can be used for supporting variants and also allowing different behaviour. Special care is needed in order to not create inconsistent sets of parameters.

In order to improve safety, read only parameters are recommended but this is up to the application developer to handle. A graphic symbol is used for specifying read only parameters. The application decides how to handle valid and non-valid read only parameters. Read only parameters are stored in NV memory and they are downloaded to the target computer.

The application is able to limit changes of read / write parameters by using the signal "ServiceTool.DisableWrite". To fulfil the requirements of off-line support tool (see below) this signal has to be used when the machine is in operation.

Notice that, due to HW, the number of writes for NV memory is limited. After the limit has been passed the memory content may become corrupted.

5.2 Protection from unauthorized use

The means for protection against unauthorized use are licenses, passwords and application controlled access to the application at runtime.

License is checked for verifying information at start of GUIDE, Service Tool and internal GUIDE executables. The data of the license file is both checksummed and encrypted.

Password and expire date can be set for GUIDE and Service Tool. Thus it is possible to block obsolete versions.

The table below shows intellectual property protection that can also be used for improving safety related aspects.

Type of IP	Protect program download	Disables access to Checkpoints and parameters	Disables access to program code
Application Type	Yes	No	No
Keyed software (connects SW to HW)	Yes	No	No
TOOLKEY, prevent unauthorized diagnostic access	No	Yes, to all.	No
Access level 0-9 (9 is the most restrictive)	No	Yes, depending on Access rights setting	No
Locked pages (view disabled, read only, normal)	No	No	Yes

The application itself can decide access to the application at runtime and handle its own password protection.

Since user identity is stored for several actions it is possible to trace responsible person.

5.3 SIL application

For the scope of this report the following applies for a SIL application:

- In the HWD file a flag is set indicating support of SIL application.
- Activity Log is included.
- Only a SIL version of Service Tool can be used for SIL application.

A SIL flag is set in the LHX file if all conditions are fulfilled for a SIL application i.e.

- SIL version of HWD file. Connection is made to specific hardware since information in system file specifies hardware with specific part number.
- SIL version of GUIDE
- COMPILE SIL2 has been used
- Manual acknowledge via dialogues. Application developer user identity is included in LHX file.

6 Programming languages

6.1 Graphic code

The graphic code has the following advantages:

- Non-programmers can understand and create programs
- Clear separation of application and infrastructure/hardware.
- Exhaustive test of graphic symbols is possible since there is a limited number of possible code blocks (one per graphic symbol). This makes it also possible to fully verify an external C-compiler with respect to the code generated by the graphic symbols.

Each application has a top level graphic page. A page may consist of modules, sub-pages and symbols. At each cycle, the execution takes place as a strict sequence from left to right.

Coding guidelines exist for use of graphic code.

The application can check its own safety level (SIL or not SIL). It shall be used for stopping the whole SIL application if not all conditions for SIL are fulfilled.

There are many different error messages that could be set as a result from erroneous graphic input.

If the result of a calculation made by a symbol is outside the range of the output type, the output value is undefined. If capped symbols are used the output value is limited to the range of the output type and an overflow output is set active. In any case no warning or error is given. Thus special care is needed.

Before application is started the first time outputs are set to predefined values. Special care is then needed in order to not create a dangerous situation at start up. This is up to the application to handle.

6.2 Structured Text

Since the SD handling of Structured Text does not fulfil all requirements in IEC 61508 it means that Structured Text shall not be used in SIL applications unless the application developer is able to prove that the implementation fulfills the requirements of IEC 61508.

A graphic component is used for accessing Structured Text code.

A Structured Text subroutine can generate different types of error. Checks are made automatically (by GUIDE at compile time and by runtime support at runtime).

In case deadline is passed when executing Structured Text code the error is propagated back to first called Structured Text subroutine and then to the graphic code where it is handled in the normal way.

6.3 C (externally developed)

Since the SD handling of externally developed C-code does not fulfil all requirements in IEC 61508 it means that externally developed C-code shall not be used in SIL applications unless the application developer is able to prove that the implementation fulfills the requirements of IEC 61508.

A graphic component is used for accessing externally developed C-code.

SD does not take any responsibility for including externally generated C-code, this is completely up to the application developer to handle. No guidance is given by SD.

7 Tools

7.1 Classification

In IEC 61508 tools are classified as T1, T2 or T3. T1 concerns e.g. administrative tools, T2 concerns test tools while T3 concerns tools that could affect the behaviour at runtime. The following is a possible classification of SD tools:

1. Software off-line support tool, T2 (Service Tool). Service Tool can be used as a test tool during application development.
2. Software off-line support tool, T3 (GUIDE). GUIDE is a compiler/translator and used as a development tool during application development.
3. Software on-line support tool - Not affecting behaviour (Service Tool). Even though not explicitly stated in IEC 61508 this case could be seen as matching T2. It is up to the developer to guarantee that the behaviour is not affected when using Service Tool at runtime (this is set up in the application).
4. Software on-line support tool - Software element according to IEC 61508 (Service Tool). In this case the Service Tool can affect behaviour at runtime (decided by the application) and is thus considered a part of the safety function (like any other Software Element). This case is complicated since much effort is necessary and SD details are needed.

Compatibility and suitability of other tools used together with SD tools is up to the application developer. SD is responsible for GUIDE and Service Tool working together.

7.2 GUIDE

There are several internal checks e.g. detection of corrupt file.

An error, by definition, stops the creation of an executable.

For a SIL application all included application parts are treated as SIL. Everything is compiled from graphics to linkable objects files since the directive COMPILE SIL2 is used. Thus there is no mixture of parts with different SILs. The following conditions apply for fulfilling COMPILE SIL2:

- A market released and certified according to IEC 61508 version of PLUS+1 GUIDE (there are built-in checks if this is the case).
- A released version of an HWD that is certified according to IEC 61508. Such HWD tells the PLUS+1 GUIDE if it supports the "COMPILE SIL2".
- Manual acknowledge via dialogues.

In the HWD file a flag is set indicating support of SIL application.

7.3 Service Tool

As described above the Service Tool can have different roles. Especially important is to be aware of the roles at runtime. For not affecting behaviour at runtime an analysis has to be made concerning the application handling this situation.

There are several internal checks e.g. detection of corrupt file.

To download a "Compile SIL2" application the following conditions apply:

- A market released and certified according to IEC 61508 version of Service Tool (there are built-in checks if this is the case).
- Only a SIL version of Service Tool can be used for SIL application.

The Service Tool reads back a checksum of the downloaded data, calculated by the boot loader of the hardware after the download is completed.

Service Tool blocks download if error is detected.

8 References

[1] IEC 61508:2010, parts 1-7

[2] PLUS+1 6.0 Evaluation, PX18354:B, 2013-02-28

SP Technical Research Institute of Sweden
Electronics - Software

Performed by

Lars Strandén

Examined by

Johan Hedberg