2008 IEEE International Workshop on Factory Communication Systems

WFCS 2008 Proceedings

May 21 - 23, 2008
Steigenberger Hotel de Saxe
Dresden, Germany

Edited by:
Gianluca Cena and Françoise Simonot-Lion

IEEE
Industrial Electronics Society

TU Dresden
Institute of Applied
Computer Science

IEEE Catalog Number: CFP08WFC-PRT

Table of Contents

Wireless Networks II

Session Chair: Andreas Willig, TU Berlin, Germany

Industrial Communications

Session Chair: Max Felser, Berne Univ. of Applied Sciences, Switzerland

Keynote

Session Chair: Gianluca Cena, IEIIT-CNR, Italy

Safety & Security

Session Chair: Julian Proenza, University of the Balearic Islands, Spain

WiP Session 2: Building Automation, Industrial Communications, Applications

Session Chair: Nicolas Navet, INRIA, France

Dependable Networks

Session Chair: Ye-Qiong Song, INPL-LORIA, France

State-of-the-Art

Session Chair: Wolfgang Kastner, TU Vienna, Austria

Middleware I

Session Chair: Klaus Kabitzsch, TU Dresden, Germany

Middleware II

Session Chair: Leon Urbas, TU Dresden, Germany

Middleware III

Session Chair: Martin Wollschlaeger, TU Dresden, Germany

Safe Commissioning and Maintenance Process for a Safe Fieldbus

Thomas Novak[1], Peter Fischer[2], Michael Holz[2], Michael Kieviet[3], Thomas Tamandl[4]

[1)] Vienna University of Technology,
Institute of Computer Technology
Gusshausstrasse 27-29
1040 Vienna, Austria
novakt@ict.tuwien.ac.at

[2)] University of Applied Sciences and Arts
in Dortmund
Sonnenstrasse 96
44139 Dortmund, Germany
{fischer, michael.holz}@fh-dortmund.de

[3)] Innotec GmbH.
Heinrich Wildung-Weg 3
21224 Rosengarten, Germany
michael.kieviet@innotecsafety.de

[4)] LOYTEC Electronics GmbH
Blumengasse 35
1170 Vienna, Austria
ttamandl@loytec.com

Abstract

In the last recent years some fieldbus systems have been equipped with additional safety related features according to the international standard IEC 61508. Thus these systems can be used in new fields of application. Typically, nodes are enhanced with hardware and safety related software.

Moreover it is required to specify a safe commissioning and maintenance process for the fieldbus systems outlined in this paper for LON. It has to cover all aspects during installation, operation and maintenance of a fieldbus. It includes the process of safe binding as well as the process of replacement and modification.

1. Introduction

In general, fieldbus systems are used to monitor and control processes in building automation or industrial environments. Typical applications are heat and climate controls or light control.

Because of their excellent properties fieldbus systems are becoming important in other fields of application, e.g. emergency door control system, energy controls or fire damper control. As a result the need for safety is constantly growing. Fieldbus systems should be designed in a way that they meet specific requirements regarding safety.

In the international standard IEC 61508 a high amount of requirements to receive a safe system are specified. The standard covers all parts of the product life cycle from the definition to the maintenance phase. It gives requirements for the device itself and for the development and maintenance process.

Generally speaking, there are two options to make a fieldbus system safe: design a new one or integrate additional features into an existing one. The first choice has two major drawbacks: It is more expensive than the second choice. Secondly, the new safe devices cannot communicate with existing non-safe devices. As a consequence a lot of fieldbus systems have been added safety features such as a safe protocol and further hardware.

A safe fieldbus is CAN (Control Area Network) with CANopen safety [2]. It meets requirements of safety integrity level (SIL) 3 defined in IEC 61508 (see section 2 for details) by using a redundant hardware structure with two microcontrollers on every safe node. Another solution is Safety-over-EtherCAT [1]. It also specifies a redundant hardware structure and a safe protocol including data duplication and CRCs to fulfil requirements of SIL 3. Other technical solutions are mentioned in [13].

However, from the safety point of view it is not sufficient to just redesign the hardware structure of the nodes and add safety related software being executed on the node. Additionally, the commissioning and maintenance process of a fieldbus system must be adapted to the safety requirements. A safe commissioning and maintenance process must cover all aspects during the installation, operation and maintenance phase of a fieldbus system.

The remainder of the document is structured as follows: Section 2 gives an overview of safety in fieldbus systems in general. Additionally, the project SafetyLon is introduced where LON (Local Operating Network) standardized in EN 14908 [6] is made safe. The safety related life cycle model is mentioned in section 3. Section 4 outlines the safety related software architecture on PC and node side required to handle the safe commissioning and maintenance process. Finally,

978-1-4244-2350-7/08/$25.00 ©2008 IEEE.

section 5 discusses the details of the commissioning and maintenance process.

2. Safety in Fieldbus

The international standard IEC 61508 - Functional Safety of electrical/electronic/programmable electronic safety related systems, introduces measures for safety related systems. Within the standard [10] safety in general is defined as "The absence of unacceptable risk of physical injury or damage to health of people […]".

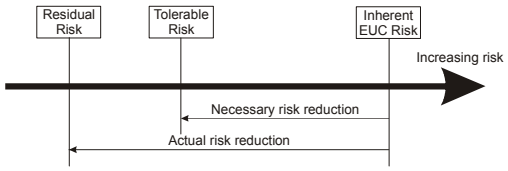

Figure 1 Tolerable risk

IEC 61508 introduces a life cycle for planning, realising, maintaining and decommissioning of a safety related system. Following the life cycle, it supports the developers in avoiding systematic and handling stochastic failures. The probability of systematic failures can be reduced during the whole project by documenting all activities as listed in [14]; during implementation phase by specifying coding guidelines for programming source code or testing with redundant hardware; during operation by monitoring the program flow. Stochastic failures can be detected by running online hardware self tests or by specifying a safe protocol for message exchange.

The introduced measures and methodologies lead to a reduction of the inherit risk of an equipment under control such as a node of a fieldbus system below the maximum tolerable risk (see figure 1.)

Identification of failures and risk assessment is done with a hazard and risk analysis. Safety requirements and performance requirements are derived from the hazard and risk analysis. In the end safety functions are determined from the aforementioned requirements.

The likelihood for a successfully performing of the safety functions is categorized in four safety integrity levels (SIL). According to IEC 61508 SIL 1 is the lowest level and SIL 4 the highest. Each safety integrity level corresponds with a specified residual error probability. Refer to table 1 for the error probability of each safety integrity level. The lower the residual error probability the higher the performance of the safety functions must be, i.e. the more failures must be avoided or detected during operation.

With regard to fieldbus systems hazardous events can be identified as shown in table 2 [13] left column. The right column lists the safety requirements that are the basis for designing the safety related software.

Table 1 Safety Integrity Level according to IEC 61508

Safety integrity level (SIL)	High demand or continuous mode (Error probability per hour)
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Within the European collective research project SafetyLon funded by the European Union within the "6th Framework Programme" the LON is made safe according to the requirements of SIL 3 given by IEC 61508. The major goal of the project is

- to develop hard- and software for a safe node,
- to design development tools for creating a safe application,
- to create management tools for handling the safe network,
- to allow safe and non-safe devices to operate on the same network.

According to the requirements of SIL 3 compliant systems and taking the requirements for the safe node hardware and the hazards in fieldbus systems into account, a special safe node hardware structure is required (figure 2).

The safe node hardware is based on a 1oo2 (1 out of 2) hardware structure. This kind of hardware structure offers a hardware fault tolerance of 1. That is, a single fault does not compromise the safe behaviour of the safe node. It is achieved by joint actions of both microcontrollers involved, called Safety Chip 1 and Safety Chip 2. They control the inputs and outputs and perform the required actions for LON communication. Nonetheless, the functionality of the hardware must be tested frequently to ensure a high hardware integrity. Therefore online self tests for CPU (central processing unit), the volatile and the non volatile program and data storage are integrated. All online self tests are designed in order to meet the requirements for SIL 3 compliant systems using a 1oo2 hardware structure. Refer to [15] for details.

To overcome the hazards caused by the fieldbus system a safe protocol on the top of the LonTalk protocol, according to IEC 61508, is implemented. By means of the safe communication protocol all typical hazards in fieldbus systems (see table 2) can be detected with a probability required by SIL 3 compliant systems. Therefore the safe protocol consists of:

- Message ID
- Safe address
- Duplicated payload for cross checking
- CRC (cyclic redundancy check)
- Timestamp

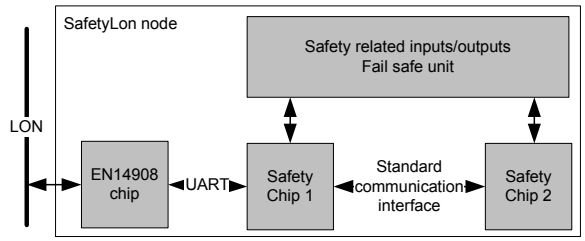

Figure 2 Hardware architecture of a safe node

For a detailed explanation of the safe protocol refer to [12].

Due to these extensions within the standard LonTalk protocol the LON and the EN 14908 chip are treated as an unsafe or called "grey" channel [14]. I.e., the 1oo2 channel architecture need not to be used in the "grey" channel. As a result only one connection to LON, connected to Safety Chip 1, is sufficient to guarantee the safe behaviour of the system. The received data, or data to send, are processed from both safety chips. Only one safety chip is neither able to verify the correctness of the received data nor to set up data to send.

Table 2 Typical hazards in fieldbus systems

Hazard	Safety requirements
Corruption of data	CRC, duplication of message
Loss of a message	Use of a watchdog
Insertion of a message	Use of safe source addresses
Repetition of a message	Use of a timestamp
Wrong sequence of messages	Use of a timestamp
Delay of a message	Use of a timestamp
Non safety related message	Use of a specific header, safe source addresses

3. Safety Related Life Cycle Model

SafetyLon supports the whole application during several phases of its life cycle concerning IEC 61508. The safety related communication is only one part of the complete safety concept of machinery, building automation and plants. Moreover processes have to be defined that specify how to program safety functions, download safety related node software and safe node user application to the node; finally how to configure and maintain the network.

3.1. Safety Related Data

For designing the application layout, the characteristics of SafetyLon must be taken into account. The possibilities and restrictions concerning

- Separation and protection of SafetyLon domains
- SafetyLon addresses
- Single installation PC

must be observed.

Coding and compilation can be done using regular, non-safety related tools. The safety functions, i.e. safety related software and safe node user application, are coded with safety programming guidelines as mentioned in [9] for C programming language by the use of an editor. After finishing the coding of the safe node user application, the compiler generates a specific, application related binary file. In addition to this the safety related node software file has to be attached. This software file contains all safety related routines for internal tests, communication among the controllers, timers, etc [12]. Note that custom coding and compilation, e.g. in the field, is explicitly not allowed when using regular, non-safety related tools. In this case it is required to use safe tools for coding and compilation.

Beside the safety related node software, other parameters are required to start the operation of the system:

1. Each node is assigned a unique identifier, called a safe address (SADR). The safe address prevents a node from masquerade. It must be avoided that the safe tool talks with the wrong device. If so, it is possible that wrong parameters will be downloaded to the safe node.
2. Communication among nodes is done by means of network variables (NVs). Generally, a network variable is a data item that an application on Node A expects to get from Node B on a network (an input network variable) or expects to make available to Node B on a network (an output network variable). The logical connection among NVs is called "binding".
3. Different safe node user applications get different identifiers. As each device can have different applications with different communication relations, it is necessary to address the right application.

Safety related data (address, binding and application information) to be exchanged with the safe node is called "device system file" (DSF). After the DSF has been verified by the node, the DSF is accessed in a diverse way and uploaded to the PC to get a valid document file. Once uploaded the DSF is compared with the already existing DSF on the PC. If the verification shows no differences, a valid document file will be generated. After automatic comparison by the PC the user also has to confirm manually at least the safe address, safe node user application information and binding parameters.

Such a way of exchanging safety related data is required because is must be granted to the user that the same DSF as stored on the PC is available on the node to avoid any data inconsistency. Therefore the DSF is downloaded to the device and uploaded again. The PC only supports the user in comparing the DSFs, but the

final commitment to the DSF must be given by the user. After that an acknowledgement is sent to the node and data on the node is valid. In short, the user must be sure that the right DFS is on the node. Therefore the user has to acknowledge the DSF manually.

3.2. Safety Related Life Cycle State Diagram

A requirement coming from IEC 61508 is that every part of the "life" of a device must be considered. As a consequence a model and corresponding functions must be specified that cover the complete life cycle of a device. Typically, the device is in idle state waiting for an external event to get started. Most of its time it is in operation. Sometimes the device will be modified and in case of a failure will switch to safe state.

To meet the aforementioned requirement, a state diagram is specified. The safe life cycle is implemented in the form of a state machine derived from the state diagram within the safe operating system of the safe node. A safe node has seven different operation modes as shown in figure 3 and elaborated during the project.

After generation and before the comparison of the DSF (device system file) the node runs in idle mode where it does not communicate nor starts to operate. All safe outputs are switched off.

After the comparison of the DSF, during the installation, the safe node is in test mode where it does not communicate nor starts to operate. To test the function of the node a PC has to initiate the function test. All files, which are stored in the safe node are produced by a non-safety related tool. Therefore the operator has to test all programmed functions to ensure that the node operates identical to the safe address, safety related node software file and binding parameters. If the operator agrees to all addresses and to all safety functions, he confirms to the safe node that everything has been tested. This sets the mode from test to pre-run.

In the pre-run mode it is able to operate and to communicate. In difference to run mode the node generates a warning signal to inform the operator of its mode. To get in the run mode, the node must receive the confirmation from the operator that everything is tested. If the device is reset in pre-run mode, it switches to test mode again because pre-run mode can only be entered after operator confirmation.

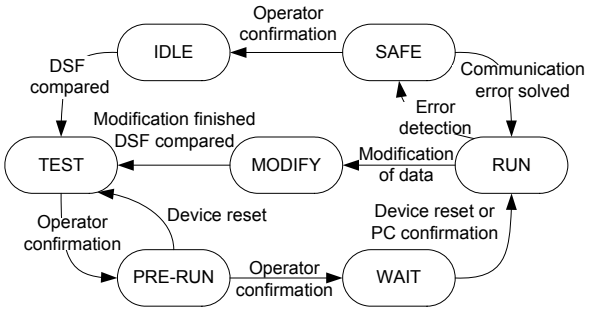

Figure 3 Safety related life cycle diagram

In the wait mode the node waits for a confirmation by a PC tool to go into the run mode. This mode is required if the system design is based on a coordinated start-up procedure that does not allow to switch over all safe nodes synchronously by reset.

In the run mode it is able to operate and to communicate. To go into this mode the node must have been in the pre-run mode first. There are two options to get into this mode: after (power-up) reset or after explicitly enabling this mode by a PC.

In operation the safety related data transfer is handled by SafetyLon. The under laying network is responsible for the data transport among the nodes within a defined timing quality. Loss of timing or transport quality within the under laying communication layer is detected by the SafetyLon layer and will result in loss of availability not in loss of safety.

In case of an error the safe node switches to safe mode. Two classes of error are specified: recoverable (the node goes to run mode automatically after the error is solved) and non-recoverable error. The first one is e.g. a communication problem due to congestion on the network; the second one e.g. a hardware failure in the RAM or CPU. In the mode the safe outputs are switched off.

The modification mode is reached after a modification of data has been initiated from external, e.g. over the network. A modification of data is done while transferring binding information. While the safe node is in modification mode, it behaves like in idle mode. Therefore in this mode the safe node does not communicate with other safe nodes and never starts to operate. When the modification has been finished the new DSF is compared and – if the verification is successful – the safe node enters test mode. Then a new test must take place.

4. System Architecture

The implementation of the safety related life cycle model requires additional software components running on a PC and the safe nodes. Moreover, a safe

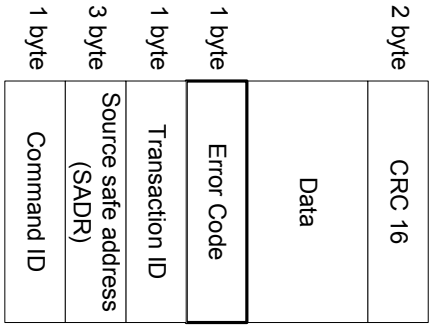

Figure 4 Safe network management massage format

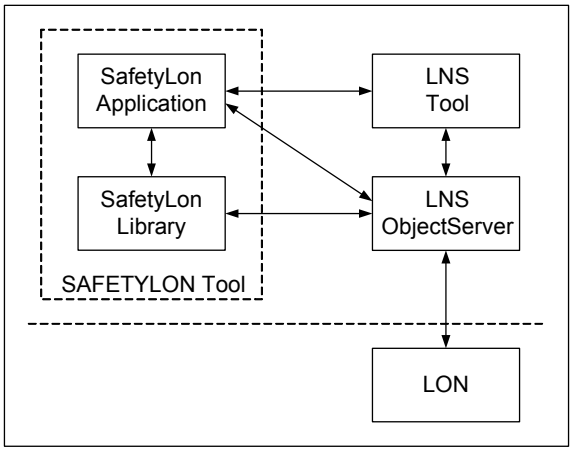

Figure 5 SafetyLon Tool with the SafetyLon Application, the SafetyLon Library and the communication interfaces to LNS and the network.

management protocol has to be specified to ensure safe communication for commissioning and maintenance between a PC and safe nodes exclusively.

In addition to the safe communication protocol, the safe management protocol is embedded into the payload field of the LonTalk protocol [7]. It specifies a request and a response message structure [8] with at maximum 42 byte. Messages only differ in byte 6 of the message structure as shown in figure 4. Only the response message includes an one byte error code field. The error code signals the software on PC side if the request message was processed successfully or failed due to some reason.

The command ID specifies what the node should do. Destination SADR equals the safe address of the node. Transactions ID is an unique identifier that characterizes the transaction between a PC and a single node. The other fields a self-explanatory and therefore not discussed.

Communication is always triggered by the PC that sends a request to the safe node. It processes the request and responds by sending a message back. Every commissioning and maintenance process needs three request/response messages according to [8]. The first message exchange is required to ensure that the correct safe node is addressed. The second one is to write data to or read data from the node. The third to acknowledge data as outlined in subsection 3.1. Consequently the system can be in three different states.

1. Open an transaction PC-node – the PC challenges the node to send a transaction ID unique for the designated transaction.
2. Send commissioning/maintenance command – the PC sends the actual command that is going to be executed at node side.
3. Commit/cancel transaction PC-node – the PC sends a commit/cancel message that signals the node to execute or not execute the command now.

The following subsections present the software architecture on PC and safe node side to handle the safe management communication.

4.1. PC side

In the SafetyLon project it was decided not to use commissioning tools available on the market for configuring and administrating the LON, but to use the LNS (LonWorks Network Operating System) database. See figure 5 for a schematic overview of the SafetyLon Tool and its interfaces to LNS. The PC tools are divided into SafetyLon Library (SLL) and the SafetyLon Application (SLA). That is for being independent from the LNS Tool. Developers of LNS Tools integrate the SLA in the LNS Tool and only use the SLL. Developers without their own LNS Tool can develop their own SLA in dependencies of existing LNS Tools on the market. This way of structuring the software allows two options to develop an individual commissioning tool for several devices. The following section explains the functionality of the SLL and the SLA within the project SafetyLon.

4.1.1. SafetyLon Application

SafetyLon Application (SLA) (see figure 6) provides services to configure one or several safe nodes, depending on the network system design. The application contains an interface to the SLL and it is instantiating the SLL within the initialization process. With functions used from this library, the application configures the safe nodes. The User-Interface shows the operator the resources of every safe node that are changeable. To obtain information from devices, the application uses the interface to the LNS Object Server API. It is required because the SLA is a standard LNS plug-in.

The SLA instantiates the LNS Object Server and then it is passed through the SAFETY API to the SLL together with at least LNS network and system object during initialization. As a consequence both software parts interact with the same instance of LNS Object Server. The event-sink for handling LNS events is implemented in the SLA. Thus the changes in the LNS

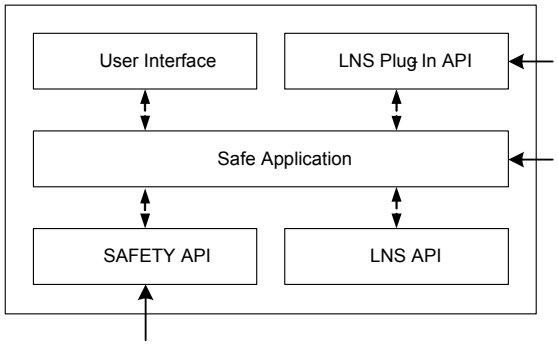

Figure 6 Software architecture of SafetyLon application

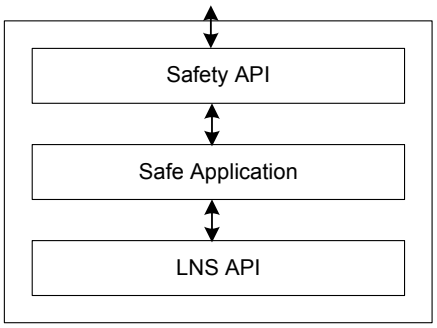

Figure 7 Software architecture of SafetyLon library

Object Server event handling should not influence the functionality of the SLL.

4.1.2. SafetyLon Library

The SafetyLon Library (SLL) software architecture (see figure 7) consists of two interfaces and one application. The LNS API handles the connection to LNS Object Server. The SAFETY API is the interface for the SLA to the SLL. The handling of LNS Object Server is described in the Echelon guidelines of the LNS Application Developer's Kit [5]. The SAFETY API must be created and defined in this library.

The SLL is a COM server as an in-proc server (a dynamic link library) and the implemented interface is the dual interface. For supporting other languages the IDL (Interface Definition Language) describes the properties and methods of the SLL. So it can be avoided to use Microsoft Foundation Classes and the SLL is independent from new Microsoft releases. Changes in the LNS Object Server should not influence the functionality of the SLL.

From the implementation point of view the SafeLon Library is divided into 3 DLLs [8]. The SafetyLon Library DLL is the main DLL that is used by the client, which is a COM EXE (LNS Plug In). The DLL provides functions to configure a safe node. This DLL uses the SafetyLon Frame DLL to construct and deconstruct the management and diagnostic communication frame (see figure 4). The frame is sent by the SafetyLon Library DLL to the safe node. The current configuration of a safe node including the safe address is stored in the SafetyLon Database DLL. It manages and stores the safe addresses. Additionally it stores all configuration information in the database.

4.2. Node Side

As already shown in figure 2 every safe node includes two safety chips. Safety related software on both safety chips is structured in three layers as shown in figure 8 [12]. Since only Safety Chip 1 is connected to the LON, the software part to access the network is omitted on Safety Chip 2.

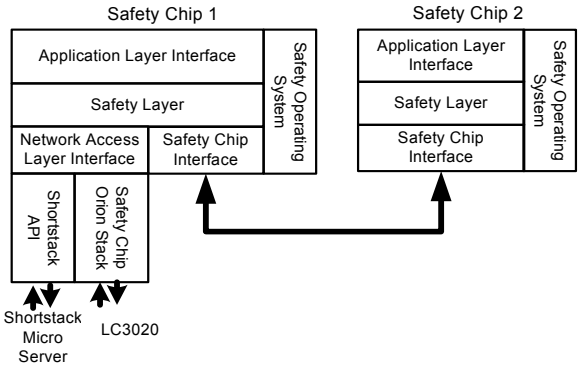

Figure 8 Safe node software

The underlying part of the safety related software is a third party EN 14908 software. It is used to communicate with the EN 14908 network interface: ShortStack Micro Server from Echelon [4] or LC 3020 from LoyTec [11]. The lower layer offers services to communicate over the LON and to exchange messages between the safety chips. The middle layer incorporates the safe network management protocol stack; moreover online hardware self tests [15], interfaces to the safe inputs and outputs. The upper layer is the application programming interface for the user. The safety operating systems runs the software on each safety chip. It is beyond the scope of this document and not outlined here.

Safe network management messages are exchanged by means of LonTalk [7] explicit message service [3]. Safety Chip 1 receives a safe network management request via the EN 14908 network interface from the PC (see figure 9). It forwards the complete request to Safety Chip 2 using the Safety Chip Interface. Both safety chips are processing the request in the SafetyLon Communication module and must agree on the same result. They inform each other by exchanging acknowledgments. Next both are building the response message with the expected data. Safety Chip 2 sends its message to the other safety chip. It compares the response message with its own one. In case of identity it

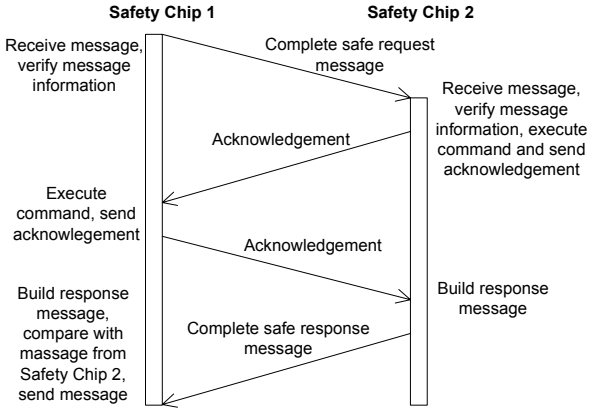

Figure 9 Control flow of processing safe management messages

sends the response as it is otherwise it sets a defined error code first.

5. Management Process

It is obvious that a safe fieldbus as any other non-safe fieldbus system needs mechanism to manage the system. Since thorough requirements are given by IEC 61508, standard tools on PC side must be enhanced (see subsection 4.1) as well as safe node software (see subsection 4.2). The SafetyLon tool providing safe mechanism to manage the system safely supports the commissioning and maintenance process.

5.1. Commissioning

The application layer communication is based on network variables (NVs) which are sent and received by nodes. The logical connection between an input NV and an output NV is called "binding". The binding parameters consist of two parts, the LON binding parameters and the SafetyLon binding parameters. The first ones are irrelevant for SafetyLon. A binding table has to be built by the operator. This can be done by ordinary network management software. The SafetyLon binding parameters are defined as a parameter table for each node "consuming" the information (a so-called consumer). The content of this table contains the expected safe addresses of connected nodes "producing" the information (so-called producers) and the time expectations (timing parameters) for receiving messages. The commissioning of a node is basically separated into three main steps:

5.1.1. Identification with addressing

The safe node has an non-safe unique network identifier (NID) The NID resides in the EN 14908 chip and is not over-writable. Using standard network management software the NID is stored on the PC. Each safe node with its unique NID is assigned a safe address by the SafetyLon tool triggered from PC side. Moreover the file containing the address information will be generated with this data afterwards on the PC.

5.1.2. Binding Parameters and Verification

Each producer and consumer even on the same node has its own unique safe address which will be stored in the non volatile memory on both safety chips. The safe address is going to be transferred to a producer or a consumer by using the safe address of the safe node and a CRC. Then both safety chips check whether the received safe address for the producer or consumer is unique and data integrity by verifying CRC is granted.

The identification of safe node is done by using the NID stored in a network management database. This information will be sent together with the safe address to the safe node. After receiving the node verifies the received NID with the NID in the EN 14908 chip. During the storing operation the data is checked and the CRC is verified. So any failure from transfer or from the storing function will be detected by the device. After the identification and verification process only the safe address is used to address the node or a producer or consumer inside of a node.

5.1.3. Application Identification Parameters and Verification

Each safe node user application gets a unique ID during development. It is used to identify different applications and also make it possible to distinguish between various versions of the same application. For that reason the application identifier is read from the non volatile memory of both safety chips by the safe management tool. It is transferred to the PC and must be approved manually by the operator.

5.2. Maintenance

Maintenance process comprises gathering of diagnostic information, replacing of safe nodes and modification such as new safe bindings. Details are outlined in the following.

5.2.1. Diagnostic Information

Such information is stored on every safe node. It can be retrieved by the SafetyLon tool on PC side as mentioned in section 4. Typical diagnostic information are the error log of a node or the current consumers bound to the safe node.

5.2.2. Replacement of a Safe Node

If there is the need of replacing a safe node, the new safe node has a new NID. If the old (defective) safe node is replaced by a new one the old DSF (device system file) has to be loaded into the new safe node. As the safe node checks the stored NID, it is not agreeing to the new DSF. The safe node shows this failure by flashing a LED. Now the operator has to push the push button again and therefore agree to a device replacement. If this is the case, the safe node does the following function:

- Check whether all device information from the DSF will fit to the SLN (e.g. number of inputs, safe node user application).
- If the check is done with a positive result the safe node is going to create a new DSF with a new NID and the same safe address of the safe node and each producer and consumer.
- The new DSF is being transferred to the PC, verified by the tool on PC side and stored as a new document file.

5.2.3. Modification

This process means for example adding or deleting a new binding. In this case the same procedure is required as outlined in subsection 5.1.2. In that process the DSF of very safe node can be used to verify the functionality

of the safety system more efficiently. The DSF of a safe node represents a tested and verified status of the safe network. After modifications they can be used to verify the differences between the state before and after the modification. To do so the PC software sends a DSF including NID and a CRC to the corresponding safe node. The safe node verifies the DSF and sends it back to PC. The PC then verifies the received file with the original document file. If the verification succeeds the safe node has not to be tested again. If verification fails, this safe node needs to be tested again.

6. Conclusion

The SafetyLon project aims at extending the standard LON to a SafetyLon by enhancing the existing hardware on safe node side, PC and node software. Additionally, a safe management process is specified that guarantees safe commissioning and maintenance of the SafetyLon. Hard-, software and management process is designed so that requirements according to IEC 61508 SIL 3 (safety integrity level 3) are met.

The safety related life cycle model is the basis for the safe management process. It defines different states and covers all parts of the SafetyLon life cycle.

To carry out the safe management process safety related software on PC side (SafetyLon tool) and on safe node side is mandatory. Moreover a, safe management protocol is required to grant message exchange between PC and safe node only.

A management process is the safe commissioning of the node. It is separated into three steps, namely assigning a safe address to each node, make the safe binding among safe nodes and check the application identifier on each node. Such an approach guarantees that only safe nodes with defined applications can communicate safely with each other.

To maintain SafetyLon, functionality to retrieve diagnostic information, to replace a safe node and to modify the binding among safe nodes is implemented. This kind of features ensures a suitable management of a flexible and changing system.

In the end it must be mentioned that enhancing standard tools with safety features does not result in getting a safe tool. The presented approach still relies on non-safe tools. The advantage is that the effort to create the enhanced tool is lower compared to the development of safe tools. However, the disadvantage is that changing the safe node user application during operation of the system by simply uploading a new one to the safe node is strictly forbidden. Therefore safe tools such as a safe editor and a safe program to download a safe node user application to a node would be necessary.

References

[1] Beckmann, G., "Die EtherCAT-Sicherheitslösung", *AUTlook,* 05/2007, pp. 71-73, 2007.

[2] "CANopen Framework for Safety-Related Communication", *CiA work Draft 304*, CAN in Automation e.V., 2000.

[3] Dietrich, D., Loy, D., Schweinzer H.J., *LON-Technologie*, Hüthig Verlag, Heidelberg, 1998.

[4] Echelon Corporation, "ShortStack User's Guide v2", *Echelon,* 2000.

[5] Echelon Corporation, "LNS Programmer's Guide, Turbo Edition", *Echelon*, 2004.

[6] EN 14908, "European Norm. Open data communication in building automation, controls and building management – control network protocol", *EN 14908*, 2006.

[7] EN 14908-1, "Open data communication in building automation, controls and building management – control network protocol – Part 1: Protocol Stack", *EN 14908*, 2006.

[8] Fischer, P., Holz, M., Mentzel, M., "Network Management for a Safe Communication in an Unsafe Environment", in *Proceedings of the 5th IEEE conference on industrial informatics*. IEEE, Vienna, Austria, 2007.

[9] Hatton, L., *Safer C – Developing Software for High-Integrity and Safety-critical Systems*, McGraw-Hill Book Company Europe, England, 1995.

[10] IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 4: Definitions and abbreviations", *IEC 61508*, 1999.

[11] LOYTEC, " LC3k Controller Family v1.2", *LOYTEC,* 2005.

[12] Novak, T., Tamandl, T., "Architecture of a Safe Node for a Fieldbus System", in *Proceedings of the 5th IEEE conference on industrial informatics*. IEEE, Vienna, Austria, 2007.

[13] Reinert, D, Schaefer, M., *Sichere Bussysteme in der Automation*, Hüthig Verlag, Heidelberg, 2001.

[14] Wratil, P., Kieviet, M., *Sicherheitstechnik für Komponenten und Systeme*, Hüthig Verlag, Heidelberg, 2006.

[15] Tamandl, T., Preininger, P., "Online Self Tests for Microcontrollers in Safety Related Systems", in *Proceedings of the 5th IEEE conference on industrial informatics*. IEEE, Vienna, Austria, 2007.

IEEE Industrial Electronics Society

WFCS 2008 Main Page

IEEE WFCS08 Manuscript Submission System

Comments from Reviewers Regarding Your Manuscript

DD-000485

Manuscript Information

View the submitted manuscript

Contact Person

Name (first, last): **Mr. Thomas Novak**
Affiliation: **Vienna University of Computer Technology, Institute of Comput**
E-mail: **novakt@ict.tuwien.ac.at**

Address: **Mr. Thomas Novak**
Vienna University of Computer Technology, Institute of Comput
Gusshausstrasse 27-29/E384
1040 Vienna
Austria
Telephone: **+43-1-58801-38427**
Facsimile:

Manuscript

Title: **Safe Commissioning and Maintenance Process for a Safe Fieldbu**
Authors: **Thomas Novak, Peter Fischer, Michael Holz, Michael Kieviet, T**
Technical Track: **TT Technical Papers**
Main Keywords: **II-40. Security and safety applications**

Author Database

1st author: **Mr. Thomas Novak, Vienna University of Computer Technology, I**
2nd author: **Prof. Peter Fischer, University of Applied Sciences and Arts,**
3rd author: **Mr. Michael Holz, University of Applied Sciences and Arts, Ge**
4th author: **Mr. Michael Kieviet, Innotec GmbH, Germany <michael.kieviet@i**
5th author: **Mr. Thomas Tamandl, LOYTEC Electronics GmbH, Austria <ttamand**

Technical Details

Transaction Number: **DD-000485**

Data Type: **pdf**
Date: **12:06:04 12/20/07**
Origin: **M815P026.adsl.highway.telekom.at/62.47.133.218**
Status: **OK - submission completed**

Feedback from the Reviewers

- A. Contribution and Clarity:
  importance of scope ---------- [4 - Significant]
  original result ---------- [3 - Minor]
  application oriented ---------- [4 - Significant]

clear and concise ---------- [4 - Significant]

B. Recommendations to TPC ---------- [4 - Should be included]

C. Suggested form of presentation -- [O - Oral presentation]

Comments:
Paper describes implementation of a safe fieldbus using LON, in particular commissioning and maintenance procedures. Original research contribution is low, but the value of the paper is in describing practical procedures.

Comments on manuscript:
Section 1 and 2: The description of IEC61508 could be shortened. Why does the safe protocol guarantee SIL3 ?
Section 2: The "duplicated payload" does not give much protection. Paragraph including "As a result only on connection to LON, connected to Safety Chip 1, is sufficient ..." is unclear.
Section 3.1: What are "Network Variables", what makes them safe ?
Section 3.1, 4, 5.1.1: Why are safe addresses considered safe - just because they are manually double-checked ?
Figure 3: Useful diagram, give reference, or is this your contribution ?

- A. Contribution and Clarity:
  importance of scope ---------- [4 - Significant]
  original result ---------- [3 - Minor]
  application oriented ---------- [4 - Significant]
  clear and concise ---------- [3 - Minor]

  B. Recommendations to TPC ---------- [2 - Last resort]

  C. Suggested form of presentation -- [O - Oral presentation]

  Comments:
  This paper deals with the adaptation of the IEC 61508 standard on functional safety for electronic systems to the specific needs and features of the LON protocol in order to create a safe version of it (the so-called SafetyLon). In particular, it is presented here how the safe commissioning and maintenance processes are to be performed in compliance with IEC 61508.

  Said safety standard indicates that the targeted level of safety should be provided in all phases of the system life cycle. Therefore, in the paper, a model for the life cycle of the SafetyLon is presented. Then the system architecture is described. Later on the paper focuses on the specific management process.

  Many different issues are covered but the paper lacks an accurate indication of which parts of the description are new an which are covered in previous papers. In fact in all sections there are references to previous work generating doubts of the actual novelty of the work. I guess the new contribution is the life cycle model and the management process, whereas the architecture corresponds to previous work.

  Moreover there is no evaluation section. The document is more a kind of general overview of the systems with more detailed description of some of its parts, rather than a focused research paper providing information of all the

typical phases of a design process (including implementation, test and other forms of verification). Due to the size of the system and the amount of different interacting features, this evaluation should be mandatory. Besides this evaluation, some references to the lessons learnt during the process of using the IEC 61508 guidelines (is the standard clear enough, useful enough, etc.) would for sure be interesting for the WFCS audience.

Furthermore, the comparison with other approaches is not very detailed. Obviously, the complexity of these sorts of developments is an impairment for a thorough comparison with other authors' work. But in this case there is no reference to how other similar systems deal with the specific subject of the paper (the safe commissioning). Other approaches are only mentioned as alternatives for the general structure of the system, not for the commissioning.

Some parts of the text should be more carefully written in order to prevent misunderstandings. Some of the aspects that should be clarified are:

- The abstract is maybe not enough specific (describing the actual contents of the paper).

- Why it is enough to connect the LON chip to only one of the Safety Chips to have a safe behaviour. Aren't there any faults of the LON chip that could be not covered by the error detection mechanisms of SafetyLon?

- First paragraph of second column in page 3 is quite confusing.

- The transition from "PRE-RUN" to "TEST" in Figure 3 is not discussed in the text.

- How communication error recovery is performed (from SAFE to RUN in Figure 3).

- The Management Process described in Section 5 should be explicitly related to specific modes among those appearing in Figure 3.

- In Figure 7, an arrow connecting the LNS API with the LNS Object Server should be included.

- In Page 6, column 1, it should be "SafetyLon Library" instead of "Safe Layer Library"?

- Last paragraph of column 1 in Page 6 should be rewritten.

- Trusting Safe Chip 1 for making the "final" comparison among redundant response messages would not be risky, instead of trusting an independent, simpler comparator?

- Where does the replacement of a Safe Node (Section 5.2.2) fit into the previously described life cycle?

Some typos and minor writing issues:

- Page 2. Column 1. Last but one paragraph. It should be "Each safety integrity level corresponds..."

- Page 3. Column 1. Paragraph 2. It should be "As a result only one..."

- Page 3. Column 2. Paragraph 2. It should be "parameters are required to start the operation of the system..."

```
- Page 3. Column 2. Paragraph 3. It should be "It must be
avoided..."

- Page 4. Column 1. Paragraph 5. It would be better "...to
inform the operator of its mode. To get in the run mode,
the node has to..."

- "Response" is used as a verb in Page 5, column 1.
```

- ```
  A. Contribution and Clarity:
       importance of scope ------------ [5 - Exceptional]
       original result ---------------- [4 - Significant]
       application oriented ----------- [5 - Exceptional]
       clear and concise -------------- [4 - Significant]

  B. Recommendations to TPC ---------- [4 - Should be included]

  C. Suggested form of presentation -- [O - Oral presentation]

  ------------------------------------------------------------------------

  Comments:
  Very interesting paper, describing the SafetyLON approach to
  deal with safety in fieldbus systems.

  It provides very useful information for the reader, namely
  on how to design a IEC61508 compliant system.

  Some improvements that can be made to the paper:
  - References [13] and [14] should be replaced by
  English-language references, if available;
  - Some details about the status of the SafetyLON project
  should be included, such as: available products as an output
  of this project.
  ```

Accepted or Rejected?

- The manuscript is on the list of accepted manuscripts.
- The manuscript is not on the list of rejected manuscripts.
- The manuscript is not on the list of withdrawn manuscripts.
- The manuscript is not on the list of registered manuscripts.

Done Viewing


  ```