CANopen safety development solutions

Dr. Frank Jungandreas, Klaus Rupprecht, SYS TEC electronic GmbH

Influenced or driven by Industry 4.0, we can see that digitalisation also has a strong impact on sensors and measurement systems. CANopen was established to distribute digitalised measurement data. Using a controller inside a sensor with CANopen communication features is generally a part of a system with distributed intelligence. This also means distributed safety liability whenever the measurement data is part of the monitoring chain of the system. We present the following:

- **Short introduction to CANopen safety**
- **Development of CANopen Safety environments**
- **The process of certification.**

Introduction

This functional safety feature protects against risks arising from defective and faulty equipment. Functionally safe devices are used wherever people and the environment must be protected. The development of safe systems is carried out in compliance with appropriate standards, such as the basic safety standard IEC61508 and/or standards in the various fields of application.
According to the sector the safety device is operating the development has to follow also special safety regulation beside the mentioned IEC61508. Due to that the development needs a very good understanding of the application to recognize the essential safety needs. Bus communication systems for process data exchange, even in safety relevant application, gets more and more in the focus of the development.
The following article describes some aspects of the realisation of safe devices with CANopen Safety per IEC61508. IEC61508 contains all the tasks that are required for development of electrical, electronic and programmable electronic systems and operation thereof (abbreviated as E/E/PE).
It contains a comprehensive safety life cycle. For each phase of the safety life cycle, measures are defined in order to ensure safety. These include the concept phase, the risk analysis, the implementation phase (hardware and software development with verification and validation), commissioning and operation of the system. Also the usage of the device till end of life including servicing and maintenance is in the safety focus. For each safety function, a Safety Integrity Level (SIL) is determined. This is a measure of the remaining residual risk. To evaluate this, there is a calculation of the probability of failure of all components within the safety function. This must be less than/equal to the failure limit that is defined by the SIL.

CANopen safety

CANopen Safety is a protocol extension to the proven CANopen Standard (EN50325-4). At the physical layer and the lowest level of communication according to the OSI model, the internationally standardised CAN bus (ISO 11898-1/2) is used.
CANopen Safety was specified by the international users‘ and manufacturers association CAN in Automation (CiA) as DS304 and transferred into EN50325-5. Therefore, it provides the user with a standardised protocol. CANopen Safety allows the user to transfer functionally safe information or process data.
This takes place using so called SRDOs (Safety Relevant Data Objects). With the SRDO definition, it is possible to transmit both safe and non-safe information via the same CAN medium. Therefore, safety functions can be integrated into existing systems. In Figure 1, this is shown schematically. An emergency button is connected via CAN bus to a circuit

breaker function located on a motor drive. Other non-safe devices communicate via the same CAN bus segment as well.

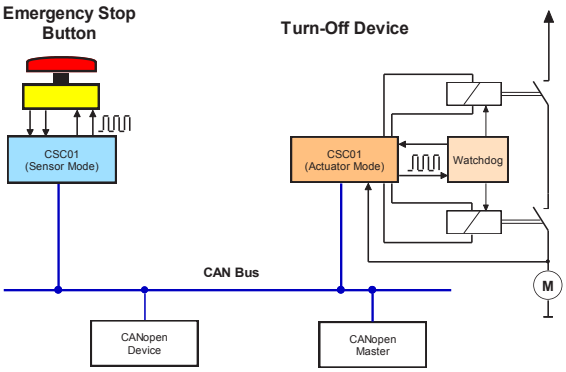

Figure 1: CANopen safety devices within a CANopen network

For safe transmission of data over bus systems, the so-called “Black Channel” is principle used. This means that the transfer layer is not considered to be safe. It is possible to use such a Black Channel in safety applications with adding a so- called Safety Communication Layer (SCL). SCL stands above the Black Channel and achieves reliable transmission. There are a number of communication errors (see [6]) which must be controlled by the Safety Communication Layer. These are:

- Falsification
- Unintentional repetition
- Incorrect sequence
- Loss
- Unacceptable delay
- Insertion
- Masquerade
- Addressing

To enable safe transmission via CAN, a SRDO has the following properties:

- A SRDO consists of 2 CAN messages.
- The safety-related information is transmitted redundantly, with the data being inverted in the second CAN message.
- The SRDO CAN identifiers discriminates themselves in 2 bits.
- The CAN identifier of 1st CAN message is always odd, the CAN identifier of the 2nd CAN message is always even.
- A SRDO is sent cyclically. The refresh time determines the period.
- The receiver monitors the cyclical reception using the parameter SCT (Safeguard Cycle Time).
- The distance between the 2 CAN messages of an SRDO is monitored by the recipient and must be ≤ SRVT (SR Validation Time).

The receiver checks the validity of the SRDO. An invalid SRDO leads the receiver to initiate the safe state for the associated safety function.

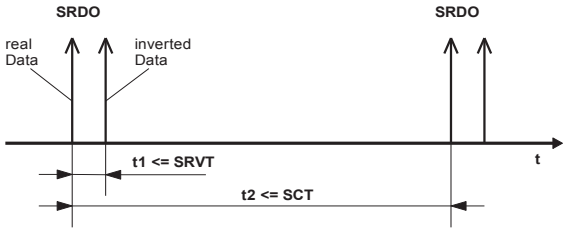

Figure 2: Timing exanple of an SRDO

Implementation models for safe CANopen devices

Figures 3 and 4 show two implementation variants. Model II shows an implementation with a redundant structure of SCL, DLL (Data Link Layer) and PhL (Physical Layer). Each channel sends a CAN message from an SRDO. SRDO reception takes place via both channels.

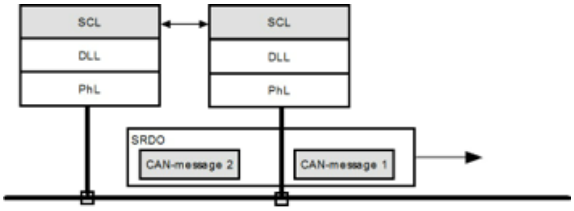

Figure 3: Model II

In Model III, a joint PhL is used.

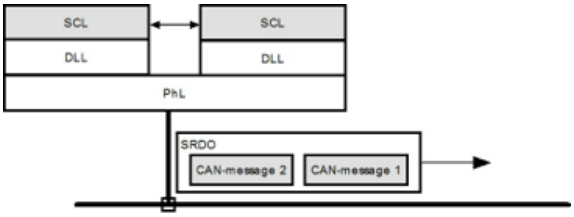

Figure 4: Model III

To assess the safety function, the residual error probability must be determined. The residual error probability of CAN (PCAN) is according to [2]:

$P_{CAN} = 7*10^{-9}$
With a redundant structure, according to models II or III, this is then squared.

$R_{LS}(P) = R(P_{CAN})^2 = 4.9 * 10\text{-}17$
The residual error rate per hour is crucial for the evaluation. The following applies:

$\Lambda = 3600 * R_{LS} * v(m-1) * 100$
(v: number of messages per second)
(m: number of participants)

The factor 100 above indicates that the transmission of the data contributes only to about 1 % of the whole safety functionality of the device application.

Example:

Requirement SIL3 level is set to $\Lambda < 10^{-7}$
The application uses 64 safe devices (m) and 44 SRDOs shall be communicated per second (v).

With this example the calculated refresh time is ≥ 23 ms.

Error types

Systematic errors may occur at any time during the development of the device.
Due to that systematic errors include errors in the specification, translation errors, errors in the dimensioning, manufacturing defects, software bugs, etc. Such errors must be avoided via Quality Assurance (QA) measures. Errors that may still arise despite QA measures must be brought under control.
Random failures include hardware failures (e.g. faulty solder joint, component failures, etc.) cannot be avoided.
These errors must be brought under control as well, they may not lead the device to a safety critical situation.

Failures are divided into safe (λs) and dangerous (λd) failures. The dangerous failures can be further divided into dangerous recognised (λdd) and unrecognised (λdu) failures.
The aim must be to minimise the number of dangerous undetected failures. This takes place with diagnostic measures.

The degree of diagnostic coverage (DC) is a measure of the effectiveness of the diagnoses.

$$DC = \frac{\lambda_{dd}}{\lambda_d}$$

The DC is dependent on the safety structure and the SIL which is to be achieved.

The proportion of safety failures SFF (Safe Failure Fraction) is defined as follows:

$$SFF = \frac{\Sigma \lambda_s + \Sigma \lambda_{dd}}{\Sigma \lambda_s + \Sigma \lambda_d}$$

Table 1: Depending on the SFF of the safety structure

SFF	**Safety Structure (Type B according to EN61508-2)**		
	HFT=0 (1oo1)	HFT=1 (1oo2)	HFT=2 (1oo3)
<60%	not permitted	SIL 1	SIL 2
60%-<90%	SIL 1	SIL 2	SIL 3
90%-<99%	SIL 2	SIL 3	SIL 4
≥99%	SIL 3	SIL 4	SIL 4

SFF depends on the HFT (Hardware Fault Tolerance), see Table above.
HFT = N: N+1 failures can lead to loss of the safety function.
PFH is the probability of failure during demand (probability of dangerous failure per hour) for systems with high demand rates (high-demand systems). The following applies:

$PFH = \lambda_{du}$ (for 1oo1 systems)

Determining factors are:

- λ_{du}: Number of dangerous failures not recognised by the diagnosis
- λ_{dd}: Amount of dangerous failures recognised by the diagnosis in the period TD between emergence and recognition of the failure
- Number of common causes (CC) for multichannel systems, i.e. Failures that impact equally dangerously in all channels (β factor).

From this, the requirements for the diagnostic test interval are derived.
Continuous and periodic tests must be carried out. The diagnostic test interval is dependent on the safety structure.

Example:

HFT = 0 (1oo1) and SIL3 within the safety cycle time
HFT = 1 (1oo2) and SIL2 within the time of the occurrence of a second failure, e.g. 1/h

$t_{test} = \frac{V}{\lambda_{dd}}$

V maximum acceptable failure rate per hour

Example:

$t_{test} = \frac{10^{-7}}{6.9FIT} = 14.5h$

Choice of hardware architecture

From the relationships described above, it can be concluded that the chosen hardware architecture has a crucial influence on safety requirements.

Therefore, including a 1oo1 architecture means more effort in the diagnosis. And all diagnoses must be made within the safety cycle time. This will cause a high CPU load as a result of the diagnostic routines. To achieve a SIL 2 or PL d (Performance Level), the degree of diagnostic coverage achieved must be “medium.” The diagnostic measures include:

- Calculation of an 8-bit checksum via the programme code
- RAM test “walk-path”
- Register test
- OP-Code test, including flags
- Testing the address calculation
- Testing the programme counter and stack pointer
- Measures against soft errors
- Diagnostics of peripherals used

To be able to carry out all required diagnoses in the event of a safety cycle time of 20 ms, care must be taken in choosing the CPU that the CPU-specific diagnostic capabilities have already been implemented in hardware. These include also an CRC generator in hardware with DMA.

To meet the requirements for the calculation of the residual error probability of the CANopen Safety Protocol, a CPU must be used with two separated CAN controllers. This corresponds to Model III, see Figure 4 as per [2].

The use of a so-called Lockstep CPU is one way to achieve a 1oo1 architecture. Lockstep CPUs are available pre-certified to SIL 3 according to [4]. The special feature of the Lockstep CPU is that the programme is run transparently for the user, on 2 CPUs (separated CPU cores) at the same time, with the results then being compared. If an error occurs on one CPU while programme execution the device is lead to a determined corresponding error response. Flash ECC and RAM interfaces are further measures for fault detection inside the Lockstep CPUs.

The advantage of 1oo1 architecture is that the software – e.g. the CANopen stack – only has to be run on one CPU. Furthermore, space requirements and the costs can be kept low.

When using a 1oo2 architecture and the same safety requirements, a “low” degree of diagnostic coverage must be reached. This reduces the effort involved in development. Furthermore, the diagnosis does not have to be carried out within the safety cycle time, but in the time of a second failure occurring. The leads to a considerable reduction of the CPU load via the diagnostic routines. The dual-channel architecture corresponds to Model II, see Figure 3 as per [2] and meets the requirements for the calculating the residual error probability of the CANopen Safety Protocol. A disadvantage of the 1oo2 architecture is the increased software overheads in porting the CANopen stack and the application software. A communication channel between the two CPUs of a device is required. The device must behave externally like a CANopen device.

Table 2: Hardware solution comparison

	1 CPU	2 CPUs
Safety structure	1oo1	1oo2
Probability of default CPU according to the manufacturer	16FIT	11.6 FIT
λ_{du}	0.08 FIT	4.6 FIT
PFH	8*10-11	1.2*10-8 β =2% βD = 2 %
SFF	> 99 %	> 60 %
Requirements for effectiveness of diagnostic methods	high	low
Diagnostic test interval	17 ms	1 h
CPU usage during the diagnosis	> 70 %	<< 1 %
Development effort	high	low

Certification process

One of the most important task during a safety development is to involve the notified body and their experts as soon as possible. This shall be done with the scope to the application the device is invented to run. It is important to counterweight the safety needs against the necessary efforts, at least the device shall be safe on a cost-level the market will accept. This first step is based on a concept paper wherein a block-diagram and a software concept is described.

Special care has to be taken for the implementing of the documentation, there is a demand for a so-called safety manual.

Beside the normal testing efforts for hard- and software several test steps and procedures are added if we are talking about safety. To insure the proper and safe functionality of the device you may have:

- Unit–Tests with Tessy
- Black-Box Test
- White-Box Test
- Timing and abnormal usage Test

Last but not least the toolchain for the development of the software functions is minimum improved for safety implementation or in the best case, but not necessarily needed, it is certificated for safety.

Conclusion

Selecting the CPU must be made on the basis of application and feature requirements. Since there are a variety of influencing factors that determine the architecture, a project typically begins with designing a basic concept to develop the foundations. In a subsequent approach phase, the results are refined in such a way that they can be discussed with the registration authorities. These steps are followed by further development steps.

It must be assumed that safety development requires higher costs and a greater amount of time. This should also be taken into account in the schedules.

References

[1] [1] CiA DS 301, CANopen application layer and communication profile

[2] EN 50325-5, Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces - Part 5: Functional safety communication based on EN 50325-4

[3] Untersuchungsbericht Nr. 2000 22949-01 zum CANopen-Safety Protokoll, BIA Berufsgenossenschaftliches Institut für Arbeitssicherheit, Fachbereich Maschinenschutz – Steuerungstechnik, St. Augustin, 16. 10. 2000

[4] IEC 61508:2010 „Functional safety of electrical/electronic/programmable electronic safety-related systems“, Part 1-7,

[5] F. Jungandreas, Entwurf funktional sicherer Mikrocontrollersysteme - Realisierungsmöglichkeiten in CANopen Safety Netzen, Journal of the University of Applied Sciences Mittweida, 3; 14-17; IWKM, Internationale Wissenschaftliche Konferenz Mittweida, 23,; 2014

[6] DIN EN 61784-3:2010: Industrielle Kommunikationsnetze – Profile - Teil 3: Funktional sichere Übertragung bei Feldbussen – Allgemeine Regeln und Profilfestlegungen, (IEC 61784-3:2010); Deutsche Fassung EN 61784-3:2010

Dipl.-Ing. Klaus Rupprecht
SYS TEC electronic GmbH
Am Windrad 2
DE-08468 Heinsdorfergrund
Tel.: + 49-3765-38600-0
Fax: +49-376-38600-4100
klaus.rupprecht@systec-electronic.com
www.systec-electronic.com

Dr. Frank Jungandreas
SYS TEC electronic GmbH
Am Windrad 2
DE-08468 Heinsdorfergrund
Tel.: +49 3765-38600-0
Fax: +49-3765-38600-4100
frank.jungandreas@systec electronic.com
www.systec-electronic.com