Features and application fields for the CANopen Safety Chip CSC01

R. Sieber, F. Jungandreas, SYS TEC electronic

This article describes the properties and possible applications of the „CANopen Safety Chip (CSC)“ in safety related devices. The object being described is a chip, partially certified by TÜV, which has met all the requirements of the IEC61508 standard up to SIL3 (safety integrity level). With the implementation of the CSC in safety related systems, a considerable reduction in costs for development and certification is achieved. Time to market is reduced noticeably. The structures of the CSC and the safety related permanent firmware are described here as well. Examples are shown to demonstrate possible applications. For example, the CSC assumes all control and monitoring functions in simple fail safe sensor/actuator systems. In complex devices (such as safety light curtains, drives) it can be integrated as a communication interface. There is a function and data interface available to serve as an interface to the application. This allows the user to access the CANopen functions, safety relevant data and the Object Dictionary. The user-oriented programmability of the CSC facilitates flexible implementation in various application environments.

Introduction

The use of bus systems in safety relevant systems is steadily increasing. The international standard IEC61508 defines measures and methods for developing and evaluating processor based safety systems, thus establishing an internationally recognized basis for modern control systems in the safety technology field.

The CSC provides manufacturers of safety relevant devices with a certified chip. It enables the development of safety relevant sensors and actuators which are networked over a standardized field bus.

The time and costs for development and certification of safety relevant devices are reduced considerably with the implementation of a partially certified CSC. Certification can be obtained through the TÜV (Technischer Überwachungsverein-German Association for Technical Inspection) or the BIA (Berufs-genossenschaftliches Institut für Arbeits-sicherheit - Institute for occupational safety of accident insurance institutions).

The CSC can be integrated as a partially preprogrammed chip in various safety relevant devices. Such systems could include safety sensors (e.g. emergency stop button, safety light curtains, safety mat, two-hand controls etc.) and actors (safety relay, drives etc.). The CAN field bus (international standardized in ISO11898-1/2) is used for networking. The "CANopen Framework for Safety Relevant Communication" [1], specified by CiA (CAN in Automation international users ans manufacturers group) in DSP 304 is used as a network protocol.

Figure 1 depicts the concept of the CSC with corresponding peripherals. The connection of sensors and actuators must be dual-channel.

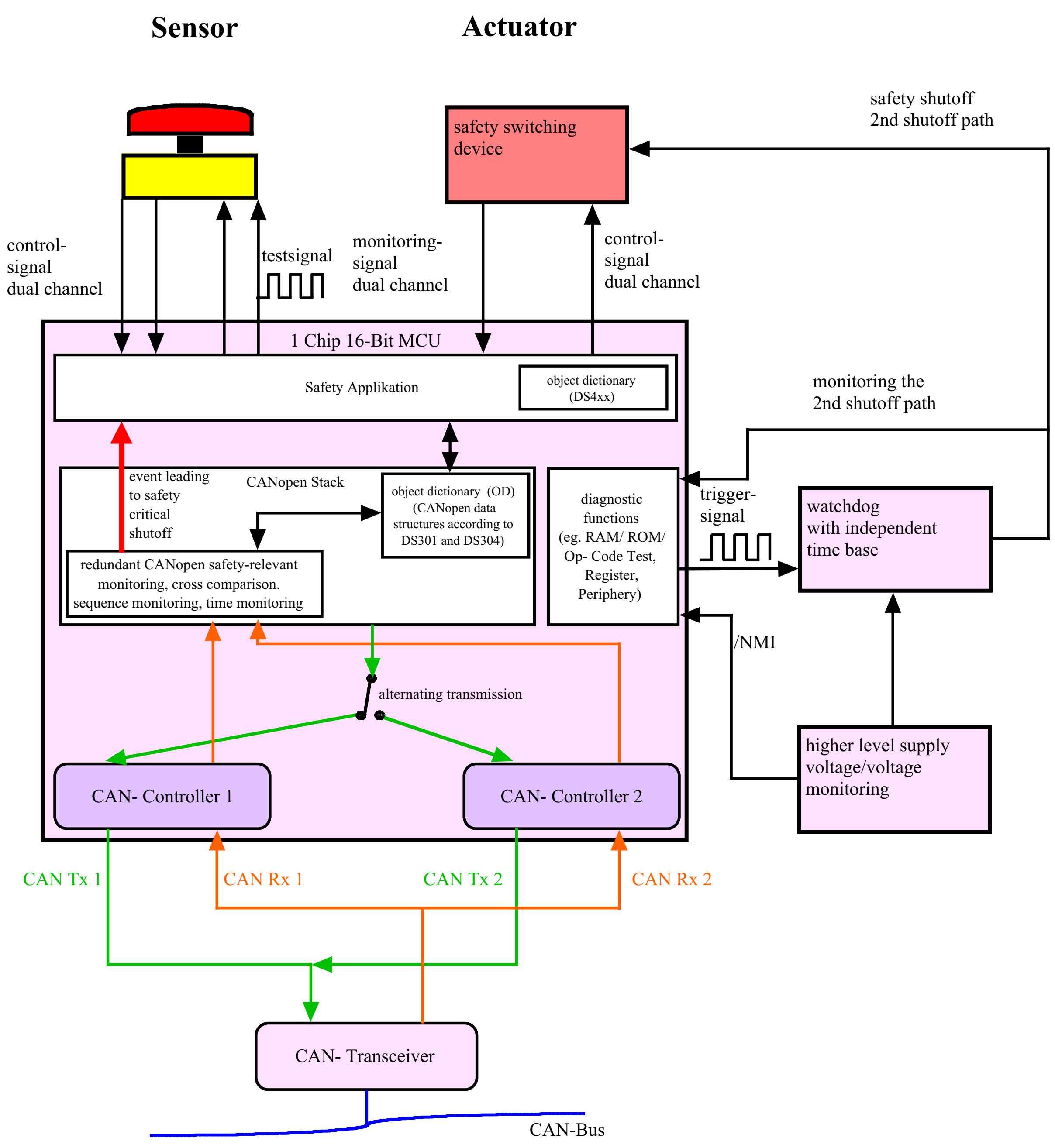
Sensor
Actuator
safety switching device
safety shutoff 2nd shutoff path
control-signal dual channel
testsignal
monitoring-signal dual channel
control-signal dual channel
1 Chip 16-Bit MCU
Safety Applikation
object dictionary (DS4xx)
monitoring the 2nd shutoff path
event leading to safety critical shutoff
CANopen Stack
object dictionary (OD) (CANopen data structures according to DS301 and DS304)
redundant CANopen safety-relevant monitoring, cross comparison. sequence monitoring, time monitoring
diagnostic functions (eg. RAM/ ROM/ Op- Code Test, Register, Periphery)
trigger-signal
watchdog with independent time base
/NMI
alternating transmission
higher level supply voltage/voltage monitoring
CAN- Controller 1
CAN- Controller 2
CAN Tx 1
CAN Rx 1
CAN Tx 2
CAN Rx 2
CAN- Transceiver
CAN-Bus

Figure 1: Concept of the CANopen Safety Chip

CANopen Safety Protocol

The CiA Draft Standard Proposal 304 „CANopen Framework for Safety Relevant Communication“ defines the CANopen protocol expansions for the integration of safety relevant devices in CANopen networks. The protocol enables safety relevant devices to operate along with non-safety relevant devices in a CANopen network. The safety functions are realized via special communication objects, SRDOs (safety relevant data object).

An SRDO consists of two CAN messages. The following rules apply for construction of an SRDO:

1. The CAN identifiers for both CAN messages vary in at least two bit positions.
2. The data contents of both CAN messages is redundant. However the data of the second CAN telegram is inverted bit for bit.
3. An SRDO is transferred periodically, whereby the period between two SRDOs is determined by the SCT (safeguard cycle time).

4. The period between both CAN messages assigned to an SRDO must not exceed the SRVT (safety relevant object validation time).

The sequence of both CAN messages assigned to an SRDO must be maintained. First the real data is transferred and then the inverted data.

The recepient (data sink) checks the validity of an SRDO. The temporal and logical succession of both CAN messages assigned to an SRDO is compared with an expected value. Subsequently the user data undergoes verification. Recognized errors will result in a change into the secure state of the assigned actuators. The secure state is to be defined by the device manufacturer and/or user, based on the application requirements.

The properties of the SRDO (CAN identifier, SCT, SRVT, Mapping) are stored in the Object Dictionary and checked for validity by a CRC (16-Bit cyclic redundant check).

In order to reduce the reaction time in safety relevant systems, there is a "global failsafe command" (GFC) defined in DSP304. It consists of two high-priority CAN messages (CAN identifier 1 and 2). Upon receipt of one of the two CAN messages the GFC is valid. The GFC contains no data and can therefore be sent by all networked nodes. Whoever sends the GFC must inform the network of the reason for this GFC transmission via SRDO.

The CANopen Safety Protocol allows safety relevant sensors and actors to be connected directly with one another. A safety relevant control unit (e.g. PLC, safety monitor) is not required. Therefore logically comparable safety chains, like those found in standard wired technology, can be realized (e.g. emergency stop button effects the safety monitoring relay directly).

Structure and Functionality of the CSC

The CSC is populated with a 16-bit microcontroller featuring two internal CAN controllers that are used redundantly. Physical connection to the CAN bus is implemented by routing signals from both on-chip CAN controllers to a single CAN transceiver and applicable EMI protection circuitry. The microcontroller operates in single-chip mode and runs with a 16 MHz internal system clock. It provides 10 kByte internal SRAM and 256 kByte Flash memory. A wide selection of on-chip peripheral modules (e.g. ADC, DAC, DMA, Timer, ports, serial synchronous and asynchronous interfaces) is available to the user. The controller is available in a QFP-100 housing and is specified for operation in a temperature range from –40 to +85°C.

The software of the CSC consists of two main parts, the permanent firmware and the variable safety application. The permanent firmware contains the CANopen safety protocol stack with all safety relevant field bus functions, the diagnostic functions for RAM, register, stack, Flash (permanent software and safety application) and the watchdog functions with monitoring. At the time of purchase, the chip already contains this part of the firmware in the Flash. It cannot be modified by the user (basic functionality of the actual CSC).

The user’s safety application is located in the second, variable Flash sector. It is loaded into the Flash by the user at a later time.

The permanent firmware controls all processes in the CSC. Thus the conditions for transfer of an SRDO are verified among others. The test controls whether the SCT has elapsed and whether the data and the configuration are valid. The SRDO is then assembled and transferred. Upon receipt of an SRDO the received CAN message is examined as well. If the SRDO is valid, the safety application will be notified of the receipt in two ways (call of an event function and status in the so called safety relevant RAM). If a faulty SRDO was received or if no valid SRDO could be received in the expected time, the safety application will be notified as well. The safety application has to initiate the corresponding error response. Error responses to the SRDO enable targeted action to certain parts of the device which can then be rendered into a secure state.

Other functional units not effected by the local error are not influenced.

The CSC provides 4 SRDOs to ensure secure data transfer. Two SRDOs are reserved for the transmission of safety relevant data and two are reserved for receiving data. There are 128 bits of data (organized bytewise) available in both send and receive directions. The data structure in the SRDO is static and can be set by the user during programming. The entry of safety relevant variables in the Object Dictionary can be performed by the user during program development. The classification into the appropriate device profile likewise took place thereby.

Beside the CANopen safety services, the following CANopen standard functions are integrated in the CSC:

- 2 transmit and 2 receive PDOs (process data object), PDO linking, static PDO mapping, synchronous and asynchronous transfer
- 1 SDO Server (service data object), expedited and segmented transfer
- NMT Slave (network management)
- Heartbeat Producer
- Emergency Producer

The CANopen functions enable configuration via the CANopen Master or configuration tools, as well as the transfer of non-safety relevant data via PDO or SDO.

The CSC is specified and certified for use in devices according to IEC61508 up to SIL3 (safety integrity level).

The CSC consists of a microcontroller with redundant safety structures. A second shut-off path is required to set the device into a secure state if the microcontroller fails. According to IEC61508 such field bus devices are ranged in the class of highly available devices.

Such highly available devices require a degree of diagnostic coverage >=99% (for SLI3). The diagnostic test interval must be correspondingly smaller than the safety cycle time.

The safety cycle time is the maximum required time between the recognition of a safety relevant event ("danger discovered", "emergency shut-off activated", diagnostic error discovered) and the electrical initialization of the corresponding safety response ("Shut off", "Render drive into secure state", "Utilize safety valve"). For the CSC the safety cycle time t_{SP} was set to <=20ms.

The secure and timely discovery of errors therefore takes on special importance.

The diagnostic routines of the CSC are an important part of the permanent firmware. They determine directly the time performance and resource requirements of the chip. Time intensive diagnostics include the calculation of a 16-bit CRC across program memory as well as the diagnosis of the RAM. The algorithms used for it determine crucially the usable size of the Flash and RAM. The permanent firmware checks the 5kByte RAM via a transparent GALPAT test and the 42kByte program memory (permanent firmware and safety application) via the 16-bit CRC.

Diagnostic routines for the Op-Code, the system stack, the register and the internal periphery are also part of the permanent firmware.

Errors recognized by the diagnostic function are treated as severe safety related errors. For a sensor device this means transfer of SRDOs is stopped immediately. The CSC is now in an intrinsically secure state that cannot be exited. In an actuator application this will lead to release of the external Watchdog, which sets the actuator into the secure state via the secondary shut-off path.

The CSC contains a logical program execution monitor, which is tested by the permanent firmware. The safety application is also integrated in the program process monitoring. Errors that are recognized by the program execution monitor, are considered the same as diagnostic errors and cause a change into an intrinsically secure state.

The additional temporal program execution monitor function comes into use when the CSC is implemented in actuators. For this a Watchdog with an external time base and a time window is used. Errors in the temporal program execution effect the actors via the secondary shut-off path.

The user is solely responsible for the safety application. It serves as the general function and data interfaces of the permanent software and is used for additional data processing according to the required functionality.

Parameters for initialization of the CANopen stack (CAN bitrate and node address) are handed over in the application.

The applied periphery, which is not submitted to the diagnosis of the permanent firmware, has to be diagnosed in the safety application. If there is no safety application running on the CSC, then this will cause an error in the monitoring of the logical program execution and result in a change to an intrinsically secure state.

A variable user software enables direct integration of simple applications (e.g. emergency stop device, emergency monitoring relay, safety valve) in the CSC, whereby costs can be minimized. In more complex applications (such as safety related drives, safety light curtains, laser scanners) the CSC can function as dedicated bus interface and communicate with other hosts or microcontrollers. In such cases the safety application provides secure transfer of safety related data to and from the superordinate modules. The interface can be configured freely by the device developer.

The entire free CSC periphery is available for the design of the interface. This includes the synchronous (I_C) and asynchronous (UART) serial interfaces or parallel interfaces via the freely available ports. A total of 70 free port pins are available to the user, some of them with alternative functions.

If the chip is used as a bus interface, safety critical errors are sent from the safety application to the superordinate unit as a binary shut-off signal.

The superordinate CPU performs the shut-off itself. The CPU can, for example, transfer the drive to a secure location with a predefined function and thereby bring the entire system into an intrinsically secure state.

Structure of the function interface between the permanent firmware and the safety application

Configuration of the permanent firmware (e.g. the CANopen stack), signalization of events (e.g. SRDO received, diagnostic error recognized, send GFC, GFC received etc.) and the call of the safety application all occur via a program and data interface.

Since both software parts are developed separately, they can not be linked with eachother via the development environment's linker. Interfaces are required for an alternating call. The interface is defined by a jump table for the function call and data memory for parameter transmission. Data access occurs with fixed addresses that are known by both software parts. This call mechanism via fixed data structures is identified as "Callgate" in the following paragraphs.

A close functional connection between the safety application and the permanent firmware exists.

„Callgate“ makes functions available for the following mechanisms:

1. The permanent firmware calls the safety application functions.
2. The safety application calls permanent firmware functions or CANopen stack functions respectively.
3. Data exchange between the application and the permanent firmware that are components of the Object Dictionary.
4. Data exchange between the application and the permanent firmware that are not components of the Object Dictionary.

„Callgate“ encapsules the call mechanism and makes a normal C function interface available.

In the „Callgate“ „marshaling“ for the parameter transmission is required. Whereby wrapper functions are used in the permanent firmware as well as in the safety application, that hide the marshaling from the other software components.

The user receives the wrapper functions of the CSC as C source code. Thus he is able to easily use the functions of the permanent firmware.

The transmission of safety related data in redundant memory occurs over fixed structures that are recognized by both parts of the software. This mechanism enables an efficient use of data and minimization of resource requirements. Recopying the data during the transmissioin between modules is not required.

„Callgate“ functions:

1. Functions that have to be made available by the safety application and that are called in the permanent firmware:

Appinitialisation() : This function is called by the permanent firmware in order to initialize the application. Within the function, the safety application initializes its own global and local variables, sets the CANopen node number and defines and registers its own sections of the Object Dictionary.

AppProcess(): This function is called cyclically by the permanent firmware; the application executes its own cyclical processes in this function.

AppPdoEvent(): signals the transmission or receipt of a PDO.

AppSrdoEvent(): This function is called if an SRDO was sent or received without error.

AppGfcEvent(): signals the receipt of a GFC

AppSrdoError(): This function is called if an error occurred during the receipt of an SRDO, or if a send-SRDO could not be sent within the refresh time. The safety application confirms the successful processing of the event. If the processing is not confirmed then the permanent firmware will recognize this as a safety error and change to an intrinsically secure state.

AppStopWatchdog(): This function is called by the permanent firmware if the application’s Watchdog can no longer be used (e.g. due to a CAN error or diagnostic error). This function also provides a confirmation and monitoring mechanism, which causes the device to change to its secure state in the event of an error.

2. Permanent firmware functions that can be used by the safety application:

CscSetNodeId(): configures the node number to be used

CscDefineVariable(): defines a variable (for object entries of variable length, e.g. manufacture device name, index 0x1008 according to DS 301); is used for variables that are initialized in the safety application, whereby the object entry is located in the CANopen stack‘s permanent Object Dictionary.

CscDefineVarTab(): defines a variable block for representation in the Object Dictionary (for objects of a fixed length, e.g process variables)

CscRegisterOdPart(): registers the user specific part of the Object Dictionary in the CANopen stack of the permanent firmware.

CscWriteObject(): writes a value in an Object Dictionary entry

CSCReadObject(): Reads a value from an Object Dictionary entry

CSCSendEmergency(): sends an emergency message to the CAN bus

CSCSendGfc(): sends a GFC

The microcontroller provides a series of interrupt sources. All interrupt sources that are not used by the permanent firmware can be used by the safety application. The function interface makes a mechanism available that allows interrupt service routines to be assigned to corresponding interrupt vectors. All unused interrupts are processed by a standard interrupt handler.

Resource Requirements

The CSC is designed to maintain a security cycle time of 20ms with the resources used. Dure to the time intensive diagnostic routines (see above), not all of the microcontroller 's RAM and Flash is available for the CSC. The permanent firmware requires 32 kByte Flash and approximately 2 kByte RAM. Furthermore, 512 Byte RAM are required for the system stack. Approximately 10 kByte Flash and 2.5 kByte RAM are available for the safety application. All safety relevant data that is transferred per SRDO is a component of the permanent firmware and is used by the appilcation as shared memory. The system stack is also used equally by both software parts.

Certification Hints

The safety concept was evaluated and certified by TÜV Rheinland [9]. A partial certification of the CSC is underway and will be complete by the end of 2003. This certification covers the internal functionality, the diagnoses of the microcontroller (RAM, Flash, register, system stack, CAN controller, the timers and DMA channels used), the data transfer via CANopen Safety, as well as the logical and temporal programm execution monitoring. Thus the various possible uses as sensors or actuators or combined use (sensor and actuator in one device) are considered. This guarantees that the CSC can be implemented in devices certified up to SIL3 according to the IEC61508 norm.

The implementation of the CSC in secure devices also requires certification of the entire device. In order to achiece certification all steps prescribed in the IEC61508 must be adhered to. In the corresponding user documentation for the CSC, all partial steps and requirements are listed that need to be considered when utilizing the CSC. This guarantees that the partial certification for the CANopen Safety Software and the corresponding diagnoses is recoginzed.

The following has to be certified by the device manufacturer:

- Linking the safety application to the permanent software,
- Utilizing of the interfaces to the periphery,
- Diagnostic routines that are not already covered by the CSC effect the periphery used by the safety application such as timers, ports, ADC, DAC, I_C, UART…,
- The entire hardware implementation of the CSC; suggestions for the creation of an external Watchdog with an independent time basis and the CAN bus circuitry are available. Topics to be specified are the structure of the supply voltage and voltage monitoring, as well as the connection of sensors and actors.

Application Example

The CSC is integrated into an emergency stop device. Thus the device can be connected directly to a field bus. The contacts are connected to free port pins, enabling a dual channel connection with dynamic signal injection. The complete safety application fits into the chip. The classification as a sensor requires no external Watchdog. Thus external component costs are minimized. The device can be integrated in existing CANopen systems.

A safety monitoring relay, which also contains the CSC and is therefore suitable for a field bus, can function as an emergency shut-off device.

The logical link of the safety function via the configuration of the SRDOs in the Object Dictionary renders additional safety technology devices (e.g. PLC) superfluous. With this concept the safety function can be easily integrated in existing systems and the existing field bus can be used.

Development Board

A Development Board is available for developing devices. It contains the CSC, an external Watchdog with an independent time base, a CAN bus connection (transceiver) and a programming interface.

Firmware examples for the safety application demonstrate the function interfaces. All available port pins are routed to pin header connectors.

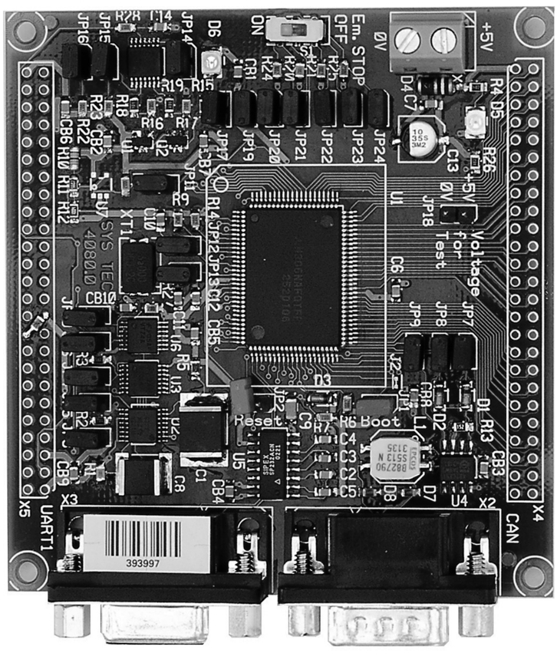

Figure 2: CSC Developmentboard

References

[1] CiA Draft Standard Proposal 304; „CANopen Framework for Safety Relevant Communication“ Version 1.0, 01.01.2002

[2] IEC61508 „Functional safety of electrical/ electronic/ programmable electronic safety- related systems“, Part 1-7, First edition 2000-05

[3] Reinert, D., Schaefer, M.: „Sichere Bussysteme für die Automatisierung“, ISBN 3-7785-2797-5, Hüthig- Verlag, 2001

[4] Klug, J., Schaefer, M.: „Fehler-erkennende Maßnahmen in Mikro-prozessoren“, Sicherheitstechnisches Informations- und Arbeitsblatt 330 225, BIA- Handbuch 29, VII/97

[5] Untersuchungsbericht Nr. 2000 22949-01 zum CANopen-Safety Protokoll, BIA Berufsgenossen-schaftliches Institut für Arbeits-sicherheit, Fachbereich Maschinen-schutz – Steuerungstechnik, St. Augustin, 16. 10. 2000

[6] „Single-Chip 16-Bit CMOS Microcomputer M16C/6N Group“ Preliminary Specifications REV.B, Mitsubishi Semiconductor Data Sheet; 2nd Edition, March 1999

[7] Reliability Report No.: MSR-01-1206, Mitsubishi Semiconductor Device Single Chip 16-Bit Microcomputer Type M306NAFGTFP, Mitsubishi Electric Corporation, Japan; December 2001

[8] F. Jungandreas: Produktanforderungen, CANopen Safety Chip CSC01, Dokument-Version/Nr 1.01/ L-1027_1 vom 28.02.2003, SYS TEC electronic GmbH, CAN in Automation

[9] Bericht über die Konzeptprüfung des CANopen Safety Chip CSC01 der Firma SYS TEC electronic GmbH; TÜV Rheinland Berlin Brandenburg, Bericht-Nr.: 968/EL215.00/03 vom 29.04.2003

Dipl.- Ing. R. Sieber
SYS TEC electronic GmbH
August- Bebel- Str. 29
D- 07973 Greiz
Phone: +49 3661 6279-0
Fax: +49 3661 63248
E-mail: R.Sieber@systec-electronic.de
Web: www.systec-electronic.de

Dr.- Ing. F. Jungandreas
SYS TEC electronic GmbH
August- Bebel- Str. 29
D- 07973 Greiz
Phone: +49 3661 6279-0
Fax: +49 3661 63248
E-mail: F.Jungandresa@systec-electronic.de
Web: www.systec-electronic.de