Implementing CANopen Safety I/O modules

CANopen Safety is gaining acceptance in particular in mobile machinery. TTControl specializes in electronic control systems for this application field and has developed an I/O module featuring CANopen Safety.

Author
Marc Weissengruber
Martin Lampacher
TTControl - HYDAC International
Schoenbrunner Strasse 7
AT -1040 Vienna
Tel.: +43-1585-3434-0
Fax: +43-1585-3434-90
products@ttcontrol.com

Link
www.ttcontrol.com

CANopen I/O modules can reduce the wiring effort for sensors and actuators, which are located far from the processing device. By using an I/O module the wiring between logic and sensors/actuators is reduced to the CAN network and the power supply. Aside from CANopen functionalities, the logic and control functions implemented on the I/O module are reduced to a minimum, e.g. a PID controller for current-controlled PWM outputs. Thus, the actual logic (e. g. safety functions) is executed by the participants of the CANopen network, which process the I/O data of the CANopen NMT slave device, e.g. the CANopen host controller with NMT master functionality.

All of the company's CANopen I/O modules are implemented on the basis of their general-purpose control units. The HY-TTC 30 family, for instance, is the basis of the HY-TTC 30X family of CANopen I/O modules. Their general-purpose control units are delivered with a C-driver library, which can be used by a custom application to control the I/Os of the device.

The CANopen I/O module implementation is actually an application programmed in C, which uses the existing drivers of the underlying platform. Figure 1 depicts this concept: The CANopen application (outlined in blue) controls the underlying hardware exclusively via the existing drivers of the platform (outlined in green) basing upon the configuration stored in the object dictionary. The CAN communication is handled via a separated CANopen stack, which also uses the CAN driver of the platform drivers.

Figure 1: The CANopen Safety I/O modules are intended for outdoor use in mobile machinery; they are certified for PL-d safety-applications (Photo: TTControl)

Integration and operation

The CANopen I/O modules are delivered with ready-to-run software and an electronic datasheet (EDS file), which makes integration into a CANopen network without any additional development effort for the I/O module possible. The CANopen application of the I/O module is essentially an interface to the underlying general-purpose control unit which implements error detection and safety mechanisms. The I/O module is typically integrated into the system by loading the EDS file of the device into the application of the CANopen master, e.g. via Codesys. The EDS file contains the definition of the CANopen object dictionary of the device and thus the description of all available features and configuration options of the I/O module.

Once the EDS file is loaded by the CANopen host controller, the developer simply chooses the desired I/O setup and fixes the data exchange between the master and I/O module by configuring the corresponding CANopen communication object. The inputs and outputs of the I/O module can then be used as if they were I/Os of the host controller.

Configuration and operation

The host controller can configure the I/Os of the CANopen module by ▷

writing the required configuration parameters to the object dictionary of the device via SDO (Service Data Object) services. All configuration options of the device, including the configuration of a pin-functionality (e.g. configure the pin for use as digital output, PWM output, timer input or analog input), modes (measurement of voltage, resistance or current if configured as analog input) and safety parameters (e.g. upper and lower voltage limits of an analog input; the limits are periodically monitored by the device), are reflected by dedicated entries in the manufacturer specific area of the object dictionary.

Similarly, dedicated entries exist, which contain the pin values of the CANopen I/O module. The I/O device updates the entries of the input process data in the object dictionary periodically. The host controller or other connected CANopen devices can retrieve these values via the corresponding CANopen communication objects. Typically the I/O module is configured to periodically transmit input data via PDO (Process Data Object) or SRDO (Safety-Related Data Objects).

Analogously, outputs of the I/O module are controlled by writing the required set-points (such as the duty-cycle of a PWM output or the level of a digital/voltage output) to the object dictionary using CANopen communication objects. The I/O module periodically checks whether the corresponding entries have been updated and uses the set-points to control the output. Typically the I/O module is configured to receive these set-points via PDO or SRDO.

Diagnostics and error control

The I/O modules autonomously perform diagnostic tasks (e. g. RAM-tests of the CPU, signal-range checks of I/Os, monitoring of board temperature) and check the I/Os for errors (e. g. short-circuit or open-load detection) – the developer using this I/O module is relieved from this task. This reduces the development effort of the application on the CANopen host controller to a minimum.

It is only necessary to check and monitor the status of the used I/Os. Additionally, the NMT status is observed. Similar to the pin values used to read input data and set outputs, dedicated entries exist in the object dictionary, which contains the status of each pin (e.g. pin is o.k., short circuit has been detected or open load is present). The host controller or other connected CANopen devices can retrieve the status via CANopen communication objects. Typically the I/O module is configured to periodically transmit the status data via PDO or SRDO.

In addition to these manufacturer-specific entries the CANopen I/O modules also support the error handling mechanism of emergency messages as defined by the CANopen specifications. The devices generate EMCY (emergency) messages in case of critical errors such as short-circuits (as specified in CiA 401) or manufacturer-specific events. An example for a manufacturer-specific event of the CANopen I/O modules is the activation of a protection mechanism, which prevents the device from damage (input and output protection).

By implementing both – the manufacturer-specific monitoring approach (retrieving status information by reading dedicated entries in the object dictionary) as well as the error handling mechanism of EMCY messages – the user has the freedom to choose which approach to use.

▷

Implementing CANopen Safety

The HY-TTC 48X and HY-TTC 36X families are the company’s first CANopen Safety devices. The HY-TTC 48XS module meets the necessary safety requirements of ISO 13849-1. It has been certified for PL-d (performance level) according to ISO 13849-1 by TÜV North (Germany). The HY-TTC 30XSH and HY-TTC 30XSI safety variants are subject to certification according to PL c of ISO 13849-1. The CANopen Safety I/O modules are based on single-channel software architecture and support safe communication via CANopen according to EN 50325-5 via a single CAN channel.

A fundamental part of the safety concept of the CANopen I/O modules is the fact that they are developed as pure software applications on top of the matching, certified general-purpose ECUs. The majority of the safety requirements are implemented by this safety platform whose safety concept and parameters (e.g. assumed safety function, performance level and diagnostics as well as safety critical system components) match the ones of the CANopen safety I/O module.

The underlying certified safety platform executes all diagnostic measures such as the required fault-detection mechanisms to achieve the required diagnostic coverage as well as internal checks of the CPU – independent of the actual application. The safe execution of the actual application is also ensured by the safety platform by executing internal tests of the CPU (such as periodic tests of the CPU registers, stack checks, RAM tests and a CRC check of the flash) and by the ECU architecture itself (e.g. 1oo1D architecture with the corresponding test equipment such as a watchdog). The currently available CANopen I/O modules feature a microcontroller of the Infineon XC2000 series. The HY-TTC 48XS, for instance, uses the XC2287M microcontroller, which features an integrated ECC test to protect the RAM and therefore allows to assume that all data in the RAM is protected against bit-flips (the device activates the safe-state in case such a bit-flip is detected).

Thus the main responsibility of the CANopen application software is essentially only to safe the CAN communication channel as specified in EN 50325-5, e.g. by activating the safe-state in case of a loss of communication caused by a cable break on the CAN-lines. The application uses single-channel architecture, i.e. no software parts are executed redundantly (neither in time nor by software diversity).

Requirements for the system designer

A safety manual is provided to the system integrator, in order to ensure the correct integration of the CANopen safety I/O module into the overall functional safety concept of the safety system. These requirements have to be fulfilled to reach the specified level of safety integrity.

The safety manual also includes requirements for the safe configuration of manufacturer-specific device parameters such as pin-configurations and modes, which are not covered by EN 50325-5. For compatibility reasons between safe and non-safe CANopen I/O modules as well as third-party CANopen configuration tools the CANopen I/O modules do not require specific configuration mechanisms (such as the calculation of CRC values) for the safe configuration.

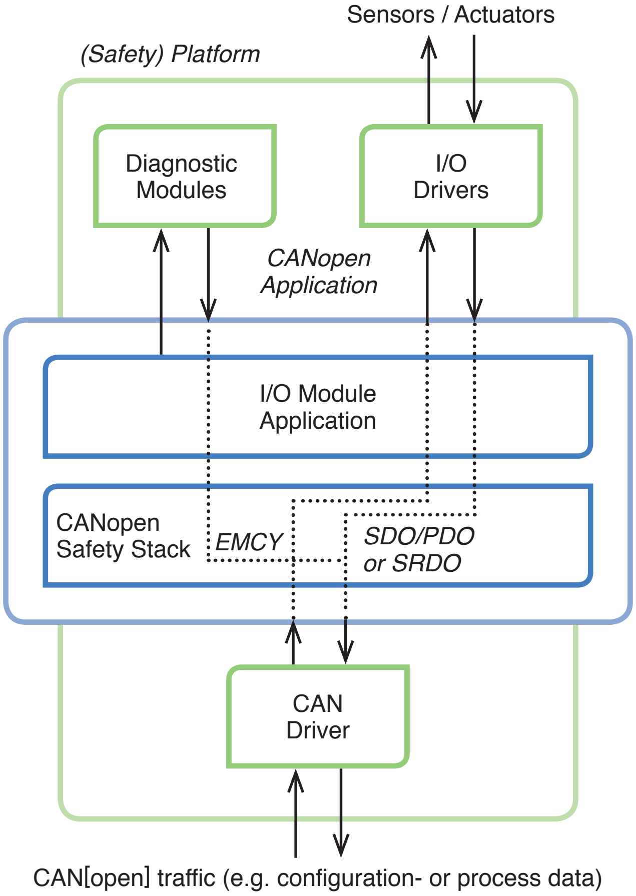

Figure 2: Basic outline of the concept used for the TTControl CANopen I/O modules (Source: TTControl)

Instead, the safety manual specifies a certain sequence, which has to be followed for the configuration if the device is used in safety-related systems:

- By default, all configurable pins of the CANopen devices are disabled (set to “not configured”). Thus the device has to be configured before use by writing the corresponding entries in the object dictionary.
- The device is configured by using the general CANopen configuration sequence (such as for PDOs), i.e. by using SDO services to write the required settings to the object dictionary.
- In order to ensure that the device has not received wrong values, e.g. due to bit-flips in the CAN frames containing the configuration data, all settings have to be read back and compared to the desired settings. The configuration has to be repeated if there are any errors.
- Once configured, the device has to be instructed to store the configuration in its non-volatile memory by writing the proper values to the CANopen parameter (1010h as specified in CiA 301).

The device stores the configuration redundantly and with CRC protection to provide the proper error detection. Upon start-up the device checks the consistency of the configuration; during run-time the configuration data is inherently protected by the underlying safety platform (amongst others by periodic tests of the RAM).

Specifying such a configuration sequence allows a certain degree of flexibility: In the prototyping phase of a system it is still easily possible to even manually configure the device. This would not be trivial if, for example, it was required to write ▷

a CRC calculated over the device configuration to the object dictionary. It also facilitates the transition from a normal CANopen I/O module to a CANopen Safety I/O module. Also, the implementation of such a configuration sequence in a configuration tool is extremely simple. Finally, the system designer may decide to use a different mechanism to verify that the device is correctly configured, e.g. by executing run-time tests via a CANopen host controller.

Compliance and device classification

The CANopen I/O modules comply with CiA 301 and CiA 305. The safety variants furthermore comply with EN 50325-5. All of the company's CANopen I/O modules are classified as "generic CANopen I/O modules" according to the CiA 401 CANopen device profile CiA for all standardized features such as digital I/Os. They provide the required entries in the object dictionary and also implement the specified error control mechanisms. Furthermore, CiA 401 specifies optional CANopen features such as interrupt-driven transmission of PDOs. Only a subset of I/O modules implements this feature.

CiA 401 also specifies an optional standardized PDO mapping, in order to provide plug-and-play functionality. This is advantageous for devices with limited configuration options. But the products feature a high-degree of configurability, which makes it very unlikely that a pre-defined PDO set-up matches the actual requirements of a system. This does not violate the CANopen conformity because in the device type parameter (object 1000h) the device-specific PDO mapping bit is set. At the moment, no pre-defined SRDO mapping is specified in the CiA 401 profile. Therefore, the safety variants of the CANopen I/O modules do not provide any pre-defined SRDOs, i.e. the devices do not send any SRDOs if not explicitly configured. ◀

References

[1] ISO 13849-1 (2006): Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design

[2] EN 50325-5 (2010): Industrial Communications Subsystem Based on ISO 11898 (CAN) for Controller-device Interfaces – Part 5: Functional safety communication based on EN 50325-4

[3] CiA 301 (2007): CANopen Application layer and communication profile

[4] CiA 305 (2008): CANopen Layer Setting Services (LSS) and protocols

[5] CiA 401 (2008): CANopen Device profile for generic I/O modules