CAN products

CAN bus systems

The message-oriented CAN bus (Controller Area Network), developed 1987 by Robert Bosch com-pany, is used in a wide range of applications to connect numerous system elements.

Originally developed for vehicle applications, CAN has growing acceptance in other industries, and has been enhanced to meet an increasing number of diverse requirements for field bus systems.

CAN and EUROS

The EUROS realtime operating system supports CAN networks in microcontroller applications. To achieve this, two software layers are available. The first one accesses the CAN hardware through drivers. It is mandatory. The second level - which is represented by a CAN protocol stack - is optional and provides logical data access.

CAN drivers

Control of a specific CAN controller on the hardware level is covered by a CAN driver, which must be compatible with the specific CAN controller type. In order to avoid the delay and expense of implementing an endless succession of drivers, after implementing more than 30 different CAN drivers, a new driver concept - the generic CAN driver - has been created.

A new application interface has been defined that covers the functionality of all available CAN con-trollers. This leads to a hardware abstraction layer over a standardized application programming interface (API). A common source code library now can be used to generate all existing CAN drivers. This approach allows that all CAN controllers are virtually identical - they have the same functions. If needed functionality is missing from a particular CAN controller, it is added/emulated by the driver's software.

Implementing new CAN controllers now means adding hardware-dependent code and connecting it with the existing source code library. This re-duces implementation and support efforts drastically and allows the re-use of tested source code. Additionally, if enhancements in the API are done, all drivers benefit automatically.

Each "generic CAN driver" is generated out of the source code library for each CAN controller type; such a driver provides the mechanisms of sending and receiving CAN messages as well as man-aging error conditions

Properties of a generic CAN driver

The following functionality is provided by generic CAN drivers:

- Initialisation of the CAN controller
- Configuration of communication parameters such as baud rate, sample point, etc.
- Transmission and receiption of both 11-bit and 29-bit messages on the same CAN bus segment
- Transmission of messages using timeout or asynchronous behaviour
- Configuration of callbacks (hooks) for message reception
- Configuration of message reception for single/ multiple message IDs and for single/multiple message ID/Mask pairs

CAN protocol stacks

CAN protocol stacks are situated on top of a CAN driver, on a second level, where the message content of CAN messages is evaluated and presented to an application using objects.

This enables the user to access data on a logical layer. It is not necessary to deal with message contents and format, and to keep timings and check timeouts; all of this is done by the CAN protocol stack itself.

Properties of CAN protocol stacks

CAN protocol stacks are implemented using approved CAN normatives, and provide the functionality described there. Hardware access to the CAN controller is done using a CAN driver. This allows the protocol stack to be totally hardware-independent because it uses the standardized driver functions.

Detailed data sheets are available

Supported CAN controllers

EUROS supports a wide range of CAN controllers from the following manufacturers: Atmel, Cirrus Logic, Fujitsu, Freescale, Intel, Infineon, OKI, Renesas, STMicroelectronics, TeleChips etc. Normally, when EUROS is ported to a new micro-controller architecture, a generic CAN driver is implemented using the existing source code base.

EUROS Embedded Systems GmbH
Campestraße 12 | D-90419 Nürnberg
Phone: +49 911 300 328 - 0 | Fax: +49 911 300 328 - 9
info@euros-embedded.com | www.euros-embedded.com