CAN: a Smart I/O System for Accelerator Controls - Chances and Perspectives

J. Bergl, B. Kuner, R. Lange, I. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger
BESSY GmbH, Lentzeallee 100, D-14195 Berlin, Germany

Abstract

Accelerator control systems should be flexible and open for future improvements. The introduction of a field bus based I/O system is a step in this direction.

At BESSY most of the accelerator devices are interfaced using the BESSY modular I/O system and the Controller Area Network (CAN), which provides excellent noise immunity and a sophisticated, reliable protocol together with low hardware costs. The rapid installation procedure of the BESSY II booster synchrotron and the experiences with the CAN installations show: CAN is a good choice to achieve an easy supportable I/O system without restricting future options.

Additional effort for installing a field bus based I/O level underneath a Standard Model Control System will be widely overcompensated by the robustness and flexibility of this approach. The growing market of CAN based industrial I/O with its increasing number of out-of-the-box interface solutions is yet another reason for choosing CAN for accelerator controls.

1 Introduction

The Controller Area Network (CAN) has been introduced in the late 80s by the Robert Bosch company. It was originally designed for automotive control, but meanwhile it is widely used for industrial applications. This results in a growing amount of available industrial I/O-modules and decreasing hardware prices.

2 CAN-facts and features

The CAN specification from Bosch [1] defines a serial communication protocol and covers basically the first two layers (i.e. Data Link Layer and Physical Layer) of the ISO Layer Model (Fig. 1).

Unlike other field buses (e.g. PROFIBUS) a CAN message contains no addresses, neither of the transmitter nor of the receiver. CAN messages (also called Communication Objects – COBs) are only specified by an unique COB-identifier with a length of 11 bits *(CAN 2.0A)* or 29 bits *(CAN 2.0B)*.

Only one node may write a certain COB onto the bus, but any attached node can read it synchronously. CAN is a multicast field bus with no distinct master or slave and *content-oriented* message exchange.

The priority of a COB is defined by its identifier where within the identifier's bitpattern a Zero is of higher priority than an One at the same digit.

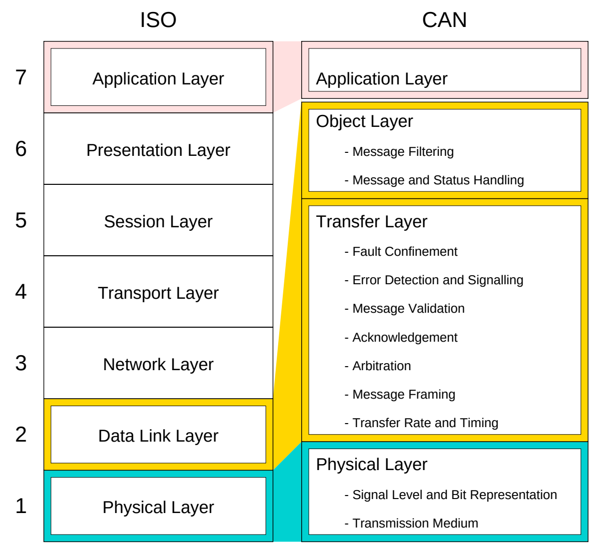

Figure. 1. ISO / CAN Layer Model

CAN uses an arbitration method called *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)*. When a node starts the arbitration it sends the identifier bitwise to the bus and simultaneously samples the bus state. If two nodes send in parallel a Zero will overwrite an One (wired-AND mechanism).

The node expecting an One at this point encounters an arbitration error and backs off from the bus, the node with a Zero at this digit continues sending without any effect to its arbitration. This mechanism leads to a non-destructive priority based bus arbitration. Therefore it is guaranteed that the higher priority message gains access to the bus – a fact obviously very important for implementing device control applications with deterministic behavior. The lower priority COB will be retransmitted in the following bus cycle if no other higher priority message is still waiting.

This arbitration scheme makes the bus length scale directly with the chosen baudrate. With 1MBit/s the maximum bus length should not exceed 25 meters (Tab. I) [4].

CAN is designed for very noisy environments and uses the *Non Return Zero (NRZ)* bitcoding. It is recommended to use only bus drivers following the *ISO11898 standard* [7]. The commonly used bus media is a shielded twisted pair cable. CAN buslines driven by ISO11898-transceivers are still working (but with reduced signal-to-noise ratio) if one of the two bus wires is broken or shortened to ground respectively

Table I
Recommended CAN Buslength vs. Baudrate

Baudrate	Buslength
1 MBit/s	25 m
800 kBit/s	50 m
500 kBit/s	100 m
250 kBit/s	250 m
125 kBit/s	500 m
50 kBit/s	1000 m
20 kBit/s	2500 m

to the supply voltage.

The CAN protocol embodies several error detection mechanisms like *Monitoring*, *Cyclic Redundancy Check*, *Bit Stuffing* and *Message Frame Check* to guarantee a proper message transfer. Important is the *Error Confinement*, i.e. CAN hardware distinguishes between disturbances and long term failures. CAN chips with long term failures automatically take themselves off the bus.

Last not least CAN chips are very inexpensive and there are several manufacturers selling CAN chips and microcontrollers with embedded CAN (Tab. II).

Table II
CAN Chip Manufacturer

Manufacturer	μP/CAN	CAN Chip
Hitachi	•	
Intel	•	•
Mitsubishi	•	
Motorola	•	
National Semiconductor	•	
NEC		•
Philips	•	•
Siemens	•	•
SGS Thomson	•	
Temic	•	•
Texas Instruments	•	

3 Flexible system design

The content-oriented multicast capability is a central element of CAN and supports a flexible I/O system design. The philosophy of CAN is similar to a producer/consumer model with several data sources. Data delivered by any node is available at the same time for every node interested. Therefore synchronous operations (e.g. correction schemes [9]) are as easy to implement as asynchronous operations (e.g. device control).

For low level applications the direct use of the CAN hardware is a reasonable way. COBs can be transferred writing and reading directly to/from the CAN chips without implementing higher level protocols. This may be adequate for applications only needing a limited functionality.

4 Interoperability and protocols

To achieve interoperability in field bus based implementations the definition of higher level protocols and standards is required.

For the industrial CAN environment there are mainly three different protocols available: *Can Application Layer (CAL)* [2] with the *CANopen* [3] Communication Profiles introduced by the *CAN in Automation Users Group (CiA)*, *DeviceNet* [6] introduced by Allen Bradley and now supported by the Open DeviceNet Vendor Association and the protocol *Smart Distributed System (SDS)* [5] from Honeywell.

In Europe and especially in Germany the non-proprietary *CAL/CANopen* is commonly used for CAN based applications and therefore most manufacturers selling CAN-I/O are supporting it.

CAL defines an implementation of an ISO/OSI Layer 7 including the specification rule for coding messages (*CMS*), dynamic distribution of identifiers (*DBT*) and the network and layer management (*NMT, LMT*) (Fig. 2).

The disadvantage of such a sophisticated protocol is the resulting implementation overhead. Therefore the CiA defined the CANopen communication profile as a subset of CAL. This reduces the protocol overhead dramatically. The main part of CAL used in CANopen is the *CAN Message Specification (CMS)*.

As a result for own applications the minimal requirement to achieve compatibility with CANopen hardware is to be CMS compliant.

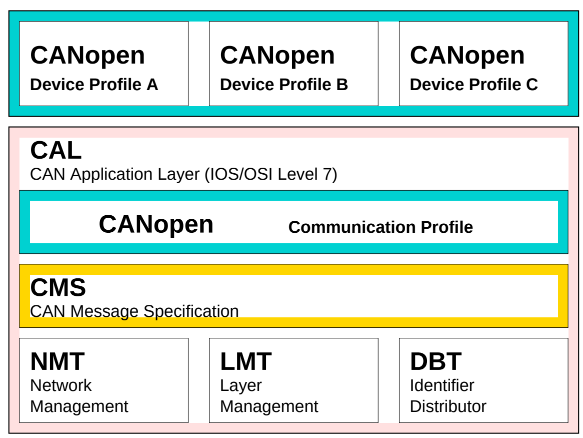

Figure. 2. CAL / CANopen Protocol

5 The BESSY approach

For the BESSY II light source an easy supportable I/O system with future enhancement options was needed. The system should be widely scalable. As far as possible non-proprietary standards should be used.

5.1 *Field bus and I/O system*

- **Requirements**

+ Field Bus:
  - usable in harsh environments
  - usable for low level applications
  - commercial available I/O
  - protocol standards available
  - embedded controller for own applications
  - minimal software effort for first steps
+ I/O Hardware Set:
  - as versatile as possible
  - minimal set of hardware variants
  - 16 bit analog resolution, low thermal drifts
  - digital I/O with insulation for each I/O-bit

- **Decisions**
  + Field Bus:
    - Controller Area Network
    - i80386 based embedded controller with on-board i82527 CAN chip
    - C programming language on embedded controller
  + I/O Hardware Set:
    - I/O combo-card with 16 bit analog, 8 bit digital I/O and integrated field bus interface
    - 16 bit digital I/O card with floating high insulated I/O lines (up to 4 kV) and field bus interface

The combination of the CAN field bus with a set of I/O hardware (Fig. 3) is the basic concept of the BESSY II I/O system. Various configuration options with only a few different hardware components (Fig. 4) are available and provide the flexibility needed to interface the majority of the BESSY II devices.

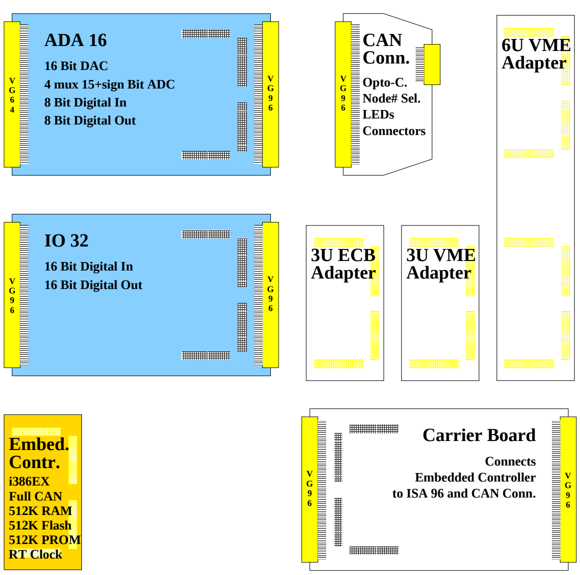

Figure. 3. Small Set of Multipurpose I/O Hardware

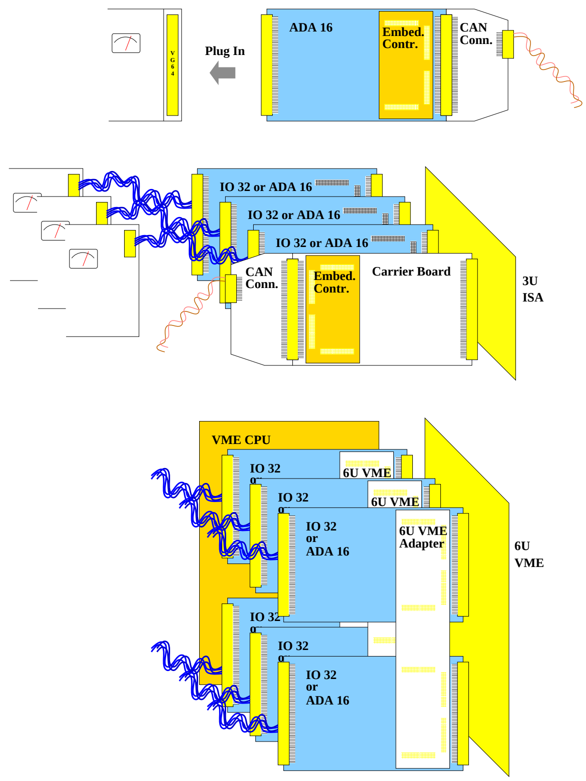

Figure. 4. Variety of Configuration Options

5.2 *Software design*

A Data Link Layer library called *Simple CAN Interface (SCI)* has been written to hide hardware specifics from the protocol layer on top of it. SCI and the protocol layer code are written in C and running on both platforms, i.e. VxWorks for the VME computers and generic i80386EX code for the embedded controllers (Fig. 5). This symmetric software design saved programming work and increases the code stability.

To achieve compatibility with the CAL/CANopen standards the protocol software is CMS compliant. Access to CANopen hardware has been successfully proven.

There is only one software module for all available I/O configuration options. The application on the embedded controller is generic and identifies the configuration (type and number of I/O cards) at bootup time. After initializing the hardware it reflects an image of the connected I/O cards over the CAN bus. There is no device specific information on this level.

6 Experiences

There are about 170 CAN nodes already installed and running at the synchrotron of BESSY II. The field bus works

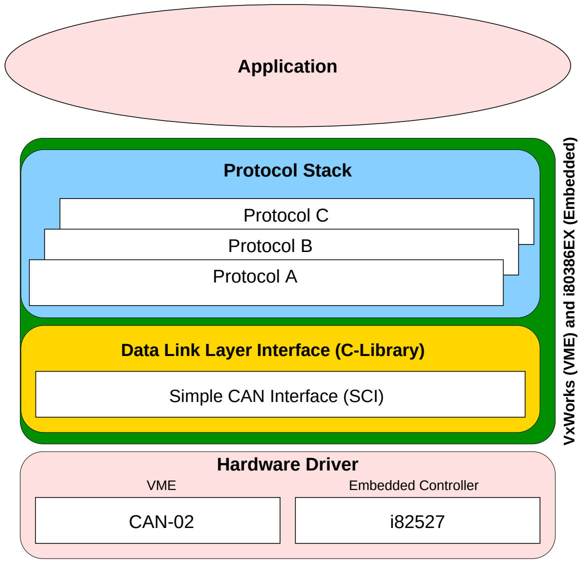

Figure. 5. Symmetric Software Design

successfully with 1MBit/s baudrate in all environments of the booster.

CAN shows its noise immunity at the modulator of the microtron where high current pulses induce large common mode distortions to the signal lines without effecting the reliability of the bus.

The typical CAN performance of the installed system (VME: MVME162-042, I/O system: i80386EX 25MHz) is as follows:

- typical CAN throughput on VME side approximately 4000 messages / s (250 μs / message)
- typical CAN transfer between an VME computer and embedded controller (request and reply / 2 CAN messages) including all software protocols needs about 800 μs

Because of the well defined interface slots of the accelerator devices (mainly power supplies) there was a very rapid plug and play (Fig. 4) installation with no cabling effort at the device side. For devices not following our interface definition the other configuration options of the I/O system are utilized.

As expected, the installation effort (e.g. cabling) was significantly decreased by choosing a field bus based solution. Using the embedded controllers adds the feature of a stateful recovery mechanism: the smart devices are storing setting and status information.

The I/O hardware and embedded controller software has been successfully used by an external manufacturer to implement the BESSY Personnel Safety Interlock. This was an important test to prove the versatility and adaptability of the modular I/O system.

7 Conclusion

Using CAN successfully for several years at the storage ring BESSY I [8] and the experiences with the installations at the booster of the third generation light source BESSY II show: CAN is a good choice for device control in accelerator and experimental physics environments.

The combination of CAN and a versatile I/O hardware set leads to a smart I/O system.

Using field bus based I/O systems for accelerator controls reveals the chance not only to share software but also hardware solutions.

References

[1] *CAN Specification Vers. 2.0*, Robert Bosch GmbH, Stuttgart 1991, Germany

[2] *CAN Application Layer*, CiA Draft Standard 201...207 Vers. 1.1, Erlangen, Germany

[3] *CANopen Communication Profile*, CiA Draft Standard 301 Vers. 3.0, Erlangen, Germany

[4] *Das offene CAN - Protokoll*, Jürgen Förster, Holger Zeltwanger, Elektronik 10/1997 p. 96, Franzis Verlag, Feldkirchen 1997, Germany

[5] *Smart Distributed System Vers. 2.0*, Honeywell Inc., Freeport IL 1996, USA

[6] *Device Net Specification Rel. 2.0*, Open DeviceNet Vendor Association, Coral Springs FL 1997, USA

[7] *Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication*, Document ISO/DIS 11898, International Standardization Organisation 1991

[8] G. v. Egan, T. Stein, J. Rahn, *Object Oriented Device Control Using the CAN Bus*, Proceedings of the 1993 ICALEPCS, Berlin 1993, Germany

[9] J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger, *Controller Area Network (CAN) — a Field Bus Gives Access to the Bulk of BESSY II Devices*, Proceedings of the 1995 ICALEPCS, Chicago 1995, USA

[10] J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger, *Embedded Controllers, Field Bus and a Modular IO Concept: Central elements of BESSY II Controls*, PAC97 , Vancouver 1997, Canada