

# EMBEDDED CONTROLLERS, FIELD BUS AND A MODULAR IO CONCEPT: CENTRAL ELEMENTS OF BESSY II CONTROLS \*

J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller, G. Pfeiffer, J. Rahn, H. Rüdiger  
BESSY, Berlin, Germany

## Abstract

Wherever applicable IO nodes connected by a field bus (FB) have apparent advantages: Simple and flexible installation and cabling, easy signal conditioning as well as data filtering and process security provided by intelligent actuators and sensors. The cost of a complex 3rd control system layer may be a high development effort. At BESSY several design principles of this layer control the risk: A few multi purpose IO cards form a modular set. Multiple IO bus connectivity provides high installation flexibility. 'Controller Area Network' (CAN) is the FB connected to the VME system of the standard model control system. Communication is provided by a small set of software modules designed to be easily adapted to hardware changes. The communication protocol with BESSY IO modules is a small and efficient subset of the 'CAN Application Layer' (CAL) standard.

## 1 ADVANTAGES OF A FIELD BUS LAYER

A reliable automated and flexible control system is essential for the effective operation of complex installations like a 3rd generation light source. A wide variety of solutions with emphasis on different properties can make up such a control system. Today the distributed architecture of console workstations on the same fast network as local process computers is widely accepted as *standard model*.

The amount of device control directly handled by local process computer IO cards compared to the data transmission via additional field bus networks connecting intelligent sensors and actuators is less settled. The advantages of a predominantly field bus based architecture obviously overcompensates the drawbacks with respect to the requirements at BESSY:

### *Installation is Simplified.*

- + *Short signal cables:* The DAC can be placed as close as possible to the power supply. Combined with properly specified components analog signals with 16 bit resolution become meaningful.
- + *Secure and cheap connection:* The field bus protocol guarantees reliable data transmission via shielded twisted pair cable even in noisy environments.
- + *Electrical independency:* Cross talk via common ground or bus backplane does not exist.

### *Intelligent Actuators and Sensors can be Utilized.*

\* Funded by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie and by the Land Berlin

- + *Data preprocessing:* Raw measured values are converted to relevant physical quantities.
- + *Asynchronous data collection:* Autonomously acquired data can satisfy requests immediately.
- + *Data sharing:* Field bus multicast facility allows task synchronisation and interlocks.
- + *Autonomous control applications:* The embedded controller may generate ramp functions, perform regulation loops and monitor device status.

### *Process Security is Improved.*

- + *Separate CPU per device:* Equipment specific software and configuration is very localized and simple.
- + *Stateful recovery:* Service is not interrupted by higher level system crashes, actual IO status and setpoints are provided to the rebooting systems.
- + *Independent trouble shooting:* Replacement of faulty or suspicious hardware does not affect other devices.

Nevertheless increased complexity and development effort should not be underestimated: A field bus communication has to be developed that matches the standard network protocol (at BESSY EPICS/Channel Access). Real time databases and program configurations of local process computer and embedded controller require additional data and link definitions. The intelligent subsystem makes task distribution as well as identification of failures more difficult.

## 2 SMART IO HARDWARE SET

The modular set of hardware (Fig. 1) satisfying most of the requirements is remarkable small. Thus the software development, configuration and maintenance effort becomes calculable.

### 2.1 Highly Integrated IO Components

- + *Multipurpose Analog IO Card:* This versatile module with high quality components is optimized to control power supplies with high stability. Features are 16 Bit DAC, 4 mux 15+sign Bit ADC, 8 Bit digital in, 8 Bit digital out, low thermal drift components combined with fully isolated analog and digital stages as well as two bus and one device connectors.
- + *Flexible Digital IO:* Card has 16 Bit digital in, 16 Bit, digital out and the same set of connectors. Bus interface unit, programmable array logic, input/output stage are fully separated segments on board.
- + *Embedded Controller:* The Piggyback Module equipped with i386EX CPU and CAN Controller can be attached to the IO Cards via ISA Bus Connector.

- + Bus Adapters/Connectors and Carrier Boards allow to use the basic components in a variety of VME, ISA 96 and CAN bus configurations.

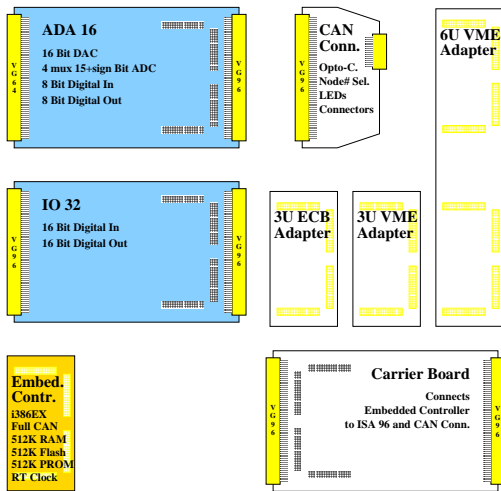


Figure 1: Components of the Modular IO Set

### 2.2 Configuration Flexibility

Wherever possible devices have been specified with slot, connector and power supply for the interface. Therefore the most common configuration is a standard IO card equipped with embedded controller, plugged into the device and attached to the CAN via bus connector (Fig. 2, top).

Unexpensive ISA 96 crates provide access to special devices exclusively equipped with analog signals or requiring more than one IO card. The embedded controller is connected to the ISA bus by a carrier board. Software is identical to the previous case. The interfaces are plugged into the ISA bus with their bus connectors (Fig. 2, middle). For a more efficient use of the ISA 96 crate the backplane can be divided into 4 independent segments of 5 slots.

6U and 3U VME bus adapters allow to use the interfaces also at the local process computer level directly (Fig. 2, bottom).

### 2.3 Network Setup

The CAN field bus has a mature protocol on an inexpensive chip available from multiple sources. Industrial support as well as increasing connectivity to industrial devices [1] make CAN a reasonable choice. The multicast facility is required for synchronisation of corrective actions [2] and used for device protection interlocks.

On the embedded controller board the Intel 82527 Full CAN Chip is used to free CPU and development effort from CAN object management. The selected VME CAN card (VCAN2 from *esd*) is equipped with a MC 68000 16 MHz CPU and two Philips PCA 82C200 Basic CAN chips capable of handling 2048 CAN objects each and providing gateway functionality between two CAN segments [2].

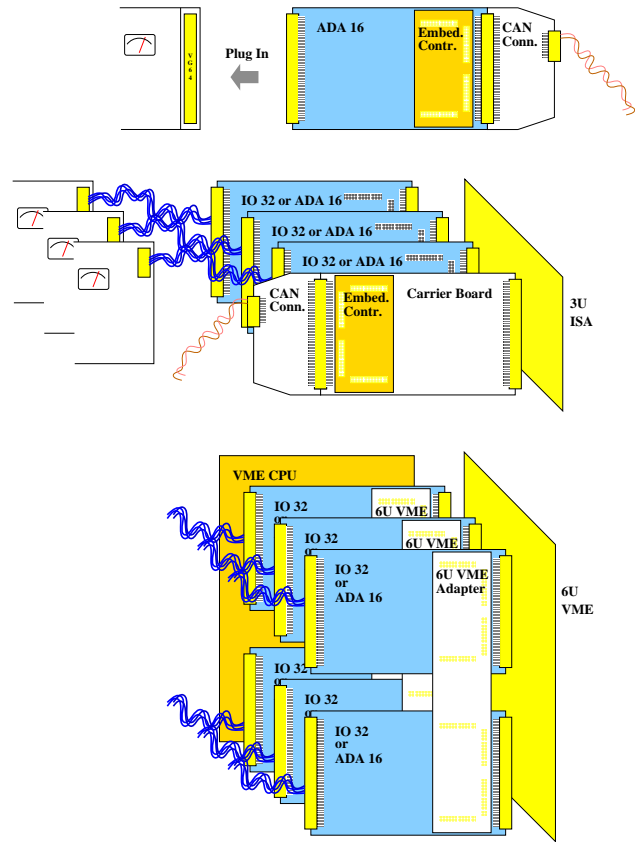


Figure 2: Basic Configuration Options

## 3 ADAPTIVE SOFTWARE DESIGN

Using CAN as major building block of the control system a CAN communication software is required that allows for hardware changes as well as for different protocols.

### 3.1 Uniform Data Link Layer

A simple programmers interface (SCI - *Simple CAN Interface*) has been defined hiding specifics of the installed hardware and providing a protocol independent data link layer. SCI reduces the effort of data transmission from a CPU to a certain CAN segment to function calls like 'Create CAN Object' assigned to a port #, 'Read' and 'Write' with different synchronisation behaviour.

On the VME side SCI is a library portable between different operating systems and capable of multithreading. Presently the VCAN2 card of *esd* is supported. The portable SCI board support driver utilizes the vendor supplied memory mapping of CAN objects. Integration of e.g. CAN *Industry Pack* modules could be done by embedding the appropriate driver (e.g. from [3]) into SCI.

SCI used by the embedded controller is a C-library of Intel specific code. It features the transparent handling of several Full CAN chips that may be used on the same CAN segment to increase the number of available CAN objects.

### 3.2 Multiple CAN Protocols Supported

The asynchronous communication between VME master and IO node requires read, write, buffering and busy/retry mechanisms for identifiable chunks of data.

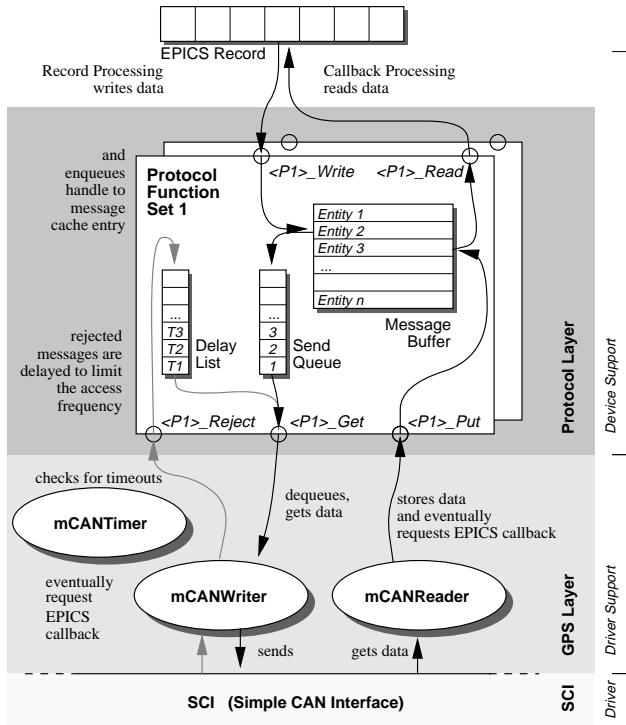


Figure 3: Sketch of the Multi Protocol Handler

In addition to the transmission engine the software has to support different communication protocols (Fig. 3). In the CAL [1] standard a *CAN Message Specification* (CMS) is given. Any protocol can be identified as conformal to the CANopen CAL/CMS implementation conventions or not.

Base protocol for data exchange between VME and embedded controller is a minimal and efficient subset of CAL/CMS called *lowCAL*. Protocol support and communication handler run symmetric both on VME motherboard and embedded controller CPU (Fig. 4).

The communication protocol for the CAN interface of the RF power amplifier Siemens S5 PLC has been specified to deliver *lowCAL* conformal messages. Other commercial CAN modules following the CAL/CMS specification (e.g. PT 100) have been integrated with minimal effort.

A PC CAN card in combination with the equivalent development environment for PCs and the embedded controller allow to interface PC based (otherwise standalone) control applications with the same software bundle.

The CAN interface to a commercial motor controller (MOCON) used for the insertion device gap drives does not fully comply with CAL/CMS. Consequently a variant of *lowCAL* has been implemented. The software would also be able to support proprietary protocols like SDS (*Smart Distributed Systems*) or *DeviceNet*.

Development and maintainance of autonomous control

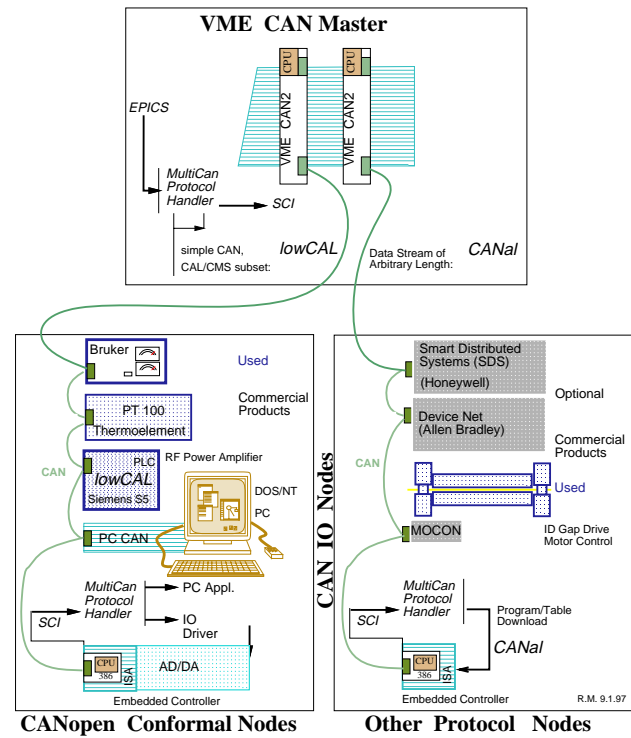


Figure 4: Supported CAN Communication Protocols

tasks performed by the embedded controller is drastically simplified if code and configuration data can be downloaded from a host via the CAN connection. It is foreseen to implement the appropriate protocol supporting data streams of arbitrary length (*CANal*) (Fig. 4).

## 4 EXPERIENCES AND SUMMARY

Today 30% of the planned control system installation is operational and fulfills the specified requirements (speed, robustness etc.). The installation process is rapid – partly due to the amount of about 80% CAN based IO nodes mostly in ‘Plug In’ configuration. The CAN communication software design revealed its power while connecting commercial CANopen conformal sensor devices. Portable implementation has been proven by an independent installation at another laboratory (SLS).

## 5 REFERENCES

- [1] CAN in Automation International Users and Manufacturers Group e.V.: *CAN Application Layer for Industrial Applications (CAL)*, CiA e.V., Nürnberg, 1995.
- [2] 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, p. 1017
- [3] See <http://www.ast.cam.ac.uk/~anj/epics/>, section ‘UKIRT/Gemini CANbus driver’