INTEGRATION OF CANopen-BASED CONTROLLERS WITH TINE CONTROL SYSTEM FOR PETRA III

P.K. Bartkiewicz, T. Delfs, S.W. Herb, B. Pawlowski
DESY, Hamburg, Germany

Abstract

For PETRA III, the high brilliance third-generation light source being built now at DESY in Hamburg, Germany, we have established a new hardware development standard for controller designs.

It includes communication on the fieldbus level, hardware interfacing to fieldbuses, and a communication application software layer for device firmware. The CAN bus and CANopen protocol were chosen as a primary fieldbus standard, and three branches of generic, CANopen compliant interfacing modules were designed for rapid controller hardware development. For fieldbus management, configuration, and integration with the TINE [1] control system, the generic TICOM (TINE-Based CANopen Manager) software was written.

This document gives an overview of our fieldbus hardware development standard and of the key features of TICOM. It also describes the first applications built on top of the standard.

INTRODUCTION

At the end of June 2007 HERA, the hadron-electron ring accelerator at DESY, was shut down. The last stage of the preaccelerator chain, PETRA II, is now being rebuilt and from 2009 on will serve as PETRA III, a high brilliance storage-ring-based X-ray source. In addition to reconstruction of the storage ring most of the components of PETRA's control system, including fieldbuses, controllers, servers, and operator applications, will be replaced.

We decided to replace the successful, but more than 25 year old SEDAC, a DESY-proprietary master-slave fieldbus system, by a modern, industrial standard. After a selection process [2] the CAN (Controller Area Network) [3] was chosen as a fieldbus with CANopen [4, 5] as the application layer protocol. Although one important selection criteria was the availability of commercial controllers compatible with the chosen fieldbus standard, most of controllers used for PETRA III require specialized functionality and therefore will be designed by our engineers.

In order to simplify the designs and make maintenance easier, we established controller development standards which cover hardware interfacing to fieldbuses, communication application software layer for device firmware, and fieldbus management and configuration tools.

Two generic boards with CANopen communication firmware are already available to the hardware designers, with a third still under construction.

The device servers of PETRA II, mostly PCs equipped with SEDAC interfaces and running Visual Basic code, will be replaced by rack-mounted Linux PCs or Windows PCs running C/C++ and Java server applications. Although CAN is the primary fieldbus standard, there will be still some SEDAC lines to permit to re-use of existing equipment. Programmers writing server to fieldbus interfacing code are offered the Common Device Interface (CDI) [6], which hides bus-specific implementation details and provides a unified set of fieldbus access APIs.

The direct connection of CAN fieldbus nodes to the TINE control system is offered by TICOM (TINE based CANopen Manager), a generic TINE server providing direct access to all process variables of CANopen devices as well as some functionality of CANopen management, including the automatic configuration of the devices.

The TICOMview client application, playing the role of a user interface, offers a comfortable means of manipulating the TICOM properties. Since Java is our language of choice for coding operator applications, TICOMview is written in Java, providing operating system independent access to the TICOM servers, and thereby to the CANopen nodes and their process variables.

The following sections of this paper describe the above topics in more detail.

THE CONTROLLER TO CAN BUS INTERFACING STANDARD

Engineers at DESY developing specialized controllers for PETRA III can build now their applications using two generic CAN bus interfacing solutions.

The first is based on the Freescale 16-bit HCS12 microcontroller. The 3U Eurocard format board contains a Freescale HCS12 microcontroller with necessary support chips, an optically isolated CAN interface, and a backplane connector with standardized signal to pin allocation where 24 I/O lines as well as 16 ADC input lines are located. The In System Programming feature makes firmware development and future maintenance very convenient. So far these boards have been programmed with C and assembler language, using the Metrowerks [7] CodeWarrior IDE, with support of the

USB debugging/programming device produced by P&E Microcomputer Systems [8].

The second CAN bus interface solution is a 120x70 mm mezzanine card with the a 32 bit Freescale ColdFire 5282 microcontroller and associated support chips as well as an Altera FPGA dedicated to custom logic and external memory interfacing. More than 2000 logic cells are available for the designer. Two connectors of the mezzanine card encapsulate 80 individually programmed I/O lines with optionally alterable functionality such as interrupts, PWM, clock inputs etc. The flash memory of the board is pre-programmed with the firmware loader, which provides the means of uploading firmware over the CAN bus using the CANopen segmented transfer. This feature allows us to remotely reprogram all boards connected to the CAN bus. These boards have also been programmed using the Metrowerks CodeWarrior IDE.

For both microcontrollers the same commercially available CANopen stack source code (delivered by the Vector Informatik GmbH [9]) was ported and the same set of APIs was defined. This lets firmware programmers concentrate on device tasks, always using the same means of interfacing their firmware to the transmission protocol.

OVERVIEW OF ALREADY AVAILABLE CANopen CONTROLLERS FOR PETRA III

DESY Designs

Several specialized controllers have been already built on top of the previously described generic development platforms

Using the ColdFire 5282 generic board we have built magnet power supply controllers providing specialized functionality such as current ramping and transient- and histogram recorders. In order to re-use some existing controllers a SEDAC crate backplane to CANopen protocol adapter has also been built.

Designs using the HCS12 generic boards include sublimation pump and valve position controllers for the Vacuum system, septum and kicker controllers, and general I/O modules used for various interlock properties.

Commercial Designs

The non-evaporable getter vacuum pumps will use commercial Energy 3000 power supplies produced by the Kniel GmbH Company. In order to satisfy our requirement of having a CANopen compatible interface the company has modified their widely sold product, equipped so far with the RS232 interface. The first set of 16 devices, grouped in two electronic racks, has already been successfully tested.

CONNECTION TO THE CONTROL SYSTEM

We have also provided a standardized solution for integrating the CAN bus with the server level of our control system. The standardization covers both hardware and software.

Hardware Platform

As a hardware platform for the servers we will use PC104, a well known and popular industrial standard for which a wide choice of CAN extension cards is available. Currently we use PC104 units produced by the Kontron AG [10] and dual channel CAN cards produced by the PEAK-System Technik GmbH [11]. The PC104 processor card stacked with the CAN card is adapted for the 3U Eurocard standard, and occupies 2 slots. The solution is very robust; there are no mechanical hard drives and we use only a small 32 MB flash disk. Since the units must work reliably at remote locations the hardware watchdog on the PC104 board is activated. In order to avoid risk of flash disk data inconsistency in case of accidental reboots, the root disk partitions are configured read-only.

As an operating system we are using ELinOS, an embedded version of Linux distributed by SYSGO AG [12]. ELinOS comes with a package of configuration tools which enable scaling and quick setup of the kernel components and easy transfer to the flash disk.

A significant advantage of using PC104 and the Linux operating system is the possibility of using desktop machines for code writing and debugging. The PEAK-System company also produces USB to CAN interfaces. In this case both interfaces, the PC104 card and the USB to CAN module are handled by the same driver, therefore the entire development of our software as well as first tests were done on the desktop machine. The code was then ported without modification to the PC104 system.

Server Software

TINE, the most popular control system at DESY, is also being used for PETRA III. The very small footprint of TINE servers makes possible to store them on the small flash disks of the PC104 boards. We have written TICOM (TINE Based CANopen Manager), a generic server which standardizes the information flow between the CANopen devices and the higher levels of the TINE control system. All CANopen devices present on the CAN bus are accessible as TINE devices, and all process variables of CANopen nodes are directly addressed as TINE properties of the devices. CANopen specific details (such as bit representation of data and assignment of data to a CAN message) are hidden at this level. Client applications contacting servers deal only with the names identifying the data. Such an approach not only frees the client application programmers from learning the

CANopen protocol, but also makes possible future hardware replacement with other designs for which the process variables may be represented in another way.

The entire configuration of the TICOM bus interface layer and the mapping of CANopen data to TINE are accomplished automatically at start-up by reading in a CANopen DCF (Device Configuration File) or EDS (Electronic Data Sheet) for each bus node and using the CANopen object dictionary descriptions to bind memory locations to the bus process variables and their names. This demonstrates the utility of the formal device description standards defined by CANopen.

TICOM also implements the functionality of the CANopen Manager (defined by the DS302 [13] CANopen specification), providing mechanisms for node configuration management, Network Management Master (NMT), and SYNC and Time message generation.

Although TICOM is a complete stand-alone server providing a connection between the CANopen protocol and TINE, it offers also the possibility of adding user specific code. There is an empty "user code" function which is periodically called every few milliseconds and may be filled with additional code to perform application specific tasks. API calls allowing access to the CANopen protocol methods and mechanisms are available to the user.

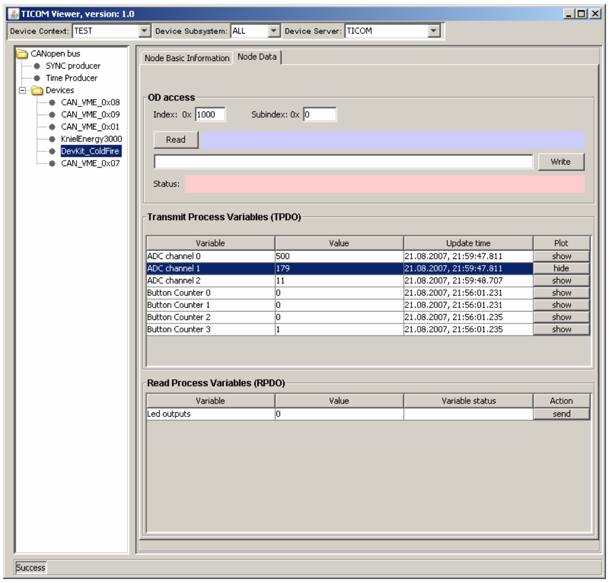
The implementation of the CDI interface is planned in the near future in order to ensure communication with middle layer servers using that technology.

CLIENT APPLICATIONS

Since TICOM is a TINE server, all of the programming guidelines familiar to our developers are valid. Programmers writing client applications requesting data transfer over a properly configured CANopen bus will not need any knowledge of technical issues related to the CAN bus and CANopen protocol.

There is also a TICOM viewer, a TICOM dedicated client application written in Java, which gives an overview of the CAN bus managed by the selected TICOM server and permits issuing remote NMT commands. It also provides very detailed information about each CANopen node connected to the bus and direct access to the node's object dictionary and to its all process variables (fig. 1). This application should be a convenient and generic configuration and maintenance tool.

In addition the Instant Client, a generic TINE client application, can be used for querying and sending commands to the TICOM server.



Device Context: TEST
Device Subsystem: ALL
Device Server: TICOM

OD access
Index: 0x 1000
Subindex: 0x 0
Status:

Transmit Process Variables (TPDO)

Variable	Value	Update time	Plot
ADC channel 0	500	21.08.2007, 21:59:47.811	show
ADC channel 1	179	21.08.2007, 21:59:47.811	hide
ADC channel 2	11	21.08.2007, 21:59:48.707	show
Button Counter 0	0	21.08.2007, 21:56:01.231	show
Button Counter 1	0	21.08.2007, 21:56:01.231	show
Button Counter 2	0	21.08.2007, 21:56:01.235	show
Button Counter 3	1	21.08.2007, 21:56:01.235	show

Read Process Variables (RPDO)

Variable	Value	Variable status	Action
Led outputs	0		send

Success

Figure 1: Screen shot of the TICOM viewer.

REFERENCES

[1] http://tine.desy.de

[2] P.K. Bartkiewicz, S.W Herb "The Common Firmware Programming Interface For Fieldbus Related Projects At Petra III" Proceedings of PCaPAC2006, Jefferson Lab, Newport News, Virginia, USA.

[3] Wolfhard Lawrenz "CAN System Engineering. From Theory to Practical Applications" Springer-Verlag New York Inc., 1997.

[4] Olaf Pfeiffer, Andrew Ayre, Christian Keydel "Embedded Networking with CAN and CANopen", RTC Books 2003.

[5] http://www.can-cia.org

[6] Reinhard Bacher, Philip Duval, Honggong Wu "Common Device Access For Accelerator Controls" Proceedings of PCaPAC2005, Hayama, Japan.

[7] http://www.metrowerks.com

[8] http://www.pemicro.com

[9] http://www.Vector-informatik.com

[10] http://www.kontron.com

[11] http://www.peak-system.com

[12] http://www.sysgo.com

[13] CAN in Automation Standard Proposal 302 "Framework for CANopen Managers and Programmable CANopen Devices".