

# EDUCATIONAL TOOLS FOR INDUSTRIAL COMMUNICATION NETWORKS DESIGN

P. Marino, H. Hernandez, M.A. Dominguez, F. Poza, F. Machado  
*DTE, University of Vigo, Galicia, Spain.*

{ pmarino, mdgomez, fpoza, fmachado } @uvigo.es, hhdez@nuyoo.utm.mx

**Abstract** In this paper it is presented the state-of-the-art in fieldbus technology, according with the authors' particular experience, giving diverse points of view about the necessary skills for using development tools and involved electronic devices, currently present in the training market. Given its growing impact all over automotive industry, an introduction about the CAN fieldbus is presented, along with the software and hardware tools that configure training environments for this protocol. Following the two different CAN nodes developed currently by the authors, using the aforementioned tools, are presented. Finally, future developments with great potential in the education of engineers, and fieldbus training applied to automotive industry and others, are also mentioned.

**Keywords:** Instructional design, Education of Information Technologies, Industrial Fieldbuses and Development.

## 1. Introduction

According to the definition of IEC (International Electrotechnical Commission) and ISA (Instrument Society of America) a fieldbus is a digital, serial, multidrop, data bus for communication with low-level industrial control and instrumentation devices such as transducers, actuators and local controllers. Fieldbuses are serial information transference buses used in the industry and orientated to data transmission in discrete and continuous processes. Before the appearance of fieldbuses, the connections between process units (PLCs, controllers, CNC machines, etc.) and field devices (sensors and actuators) were point-to-point links. It means complex (a lot of wires) and expensive (in time and money) installations. Also the maintenance and modification tasks were harder. These problems are solved with fieldbuses, because all devices are connected through a single physical medium (twisted pair, coaxial, two wires, optical fibre, etc.) extended over the whole area of the distributed control process system (cost reduction). So new devices can be added connecting it to the fieldbus without additional wiring (easy maintenance and modification). Other

advantage of fieldbuses is that allow the distribution of the intelligence of the control process. So the intelligence comes close to the sensor devices (smart sensors with capability of computation and communication) [Mariño, 2003].

Fieldbuses represent the lower level of the communications networks in a flexible manufacturing system. In this way, a fieldbus like a LAN (Local Area Network), fulfils the two first layers of the OSI model (physical and data link) and the last layer (application). Besides it has a layer for the management of the previous ones. But fieldbuses offer shorter messages (commands, events, measures, etc.), with answer times between 100ms and 5ms (operation in real-time), and high security in the communication (reliability), over distances from 200m up to 2km, with data rates lower than 1Mbps. Usually the physical medium is a pair wire with EIA RS-485 interfaces, although applications with coaxial cable, optical fibre, radio or infrareds can be found.

Nowadays the total standardisation of fieldbuses is not reached, [Marsh, 1999] but some organisations have indicated configuration profiles that are useful as a guide for the different manufacturers in the design of their field equipment.

## **2. Curriculum design for fieldbus skills**

The authors have been involved in research and development projects about fieldbus technology for seven years [Mariño et al., 1996]. Also, during that time, they have run curricula university programs and directed several Ph.D. thesis, about different topics in that area of knowledge [Mariño et al., 1999]. Their experience in the state-of-the-art has provided diverse points of view about the necessary skills for using development tools and involved electronic devices currently present in the training market. A curriculum about fieldbus knowledge was designed following several topics:

- Formal description languages in order to specify complex system requirements in an ambiguities free and complete way. Two languages were used: LOTOS and SDL.
- Complete specification of several fieldbus standards at the data link level of the OSI model, based on LOTOS and SDL, for making analysis and simulations of possible alternative solutions for a system. Thus it can be checked what alternative is more adequate and the right choice can be made at the design stage, which could bring important economic and time savings. For this specification two relevant European fieldbuses were selected: PROFIBUS and WorldFIP [Mariño et al., 1997].
- Design and implementation of nodes for the selected fieldbuses by means of development tools and prototype boards provided by manufacturers. An interesting line of research was opened when FPGA chips (Field

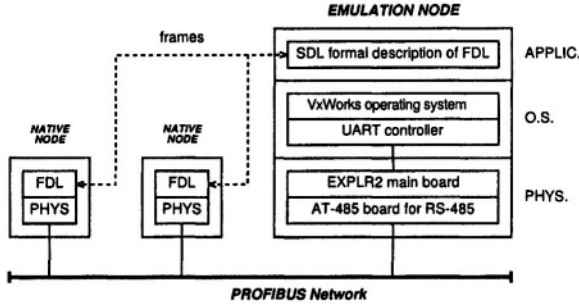


Figure 1. Profibus emulation node

Programmable Gate Array) were included for the design and implementation of that nodes [Lías et al., 2000].

- Design and implementation of nodes for the selected fieldbuses by means of RTOs (Real-Time Operating Systems). The growing interest in embedded systems, that can be reconfigured to use several fieldbus standards, gives another line of research in emulation nodes based on tools for developing environments (Fig. 1). For this purpose it was selected the Tornado tool which runs on the VxWorks real-time operating system, and is manufactured by Windriver Systems Inc.

In following sections it is made a short description about specification languages and fieldbuses used by the authors in their university lessons.

## 2.1 LOTOS language

LOTOS is a formal description technique (FDT) standardised by ISO to design services and protocols used in the communication of open systems (OSI). The system behaviour in LOTOS is specified as a succession of events or actions.

Using LOTOS the authors have specified the whole data link layer of the OSI model (level 2), for the two European fieldbuses PROFIBUS and WorldFIP.

## 2.2 SDL language

SDL is a standardised language used to specify and describe systems. It was developed by the ITU-T (International Telecommunications Union section Telecommunications), formerly CCITT (Comité Consultatif International Télégraphique et Téléphonique), and published as the Z.100 ITU-T Recommendation.

With SDL the authors have specified the whole data link layer of the European fieldbus PROFIBUS. Given that both FDTs were used to specify the same fieldbus (PROFIBUS), a comparative between both languages (LOTOS and SDL) based on achieved results was published by the authors [Mariño et al., 2001].

## **2.3 PROFIBUS reference**

In 1987 a group of manufacturers and end users, along with the German Government, defined a fieldbus based on the OSI/ISO model named PROFIBUS. This description was later published as the German standard DIN 19 245 and as the European CENELEC IN 50170 [CENELEC EN 50170, 1996a]. PROFIBUS uses only three of the OSI layers: the physical layer (PHY), the data link layer (FDL) and the application layer, FMS/LLI (Fieldbus Message Specification/Lower Layer Interface).

A PROFIBUS network was implemented by the authors to test the specifications made from LOTOS and SDL languages (sections 2.1 and 2.2), and in addition the designed and implemented nodes, making use of the development tools and prototype boards provided by manufacturers, and the included RTOs based emulators (section 1.2).

## **2.4 WorldFIP reference**

French manufacturers (Cegelec, Telemecanique, Efisystem, Gespac, etc.) promoted a protocol for industrial communications, that was standardised with the name of FIP (Factory Instrumentation Protocol). Its characteristics were recorded in the French standard NFC 46-601÷605. Later, this protocol was described by CENELEC into the standard EN 50170 vol. 3 with the name WorldFIP [CENELEC EN 50170, 1996b]. It was the same than the former French standard but with the adoption of the international standard IEC 1158-2 (now IEC 61158-2) for its physical layer.

A WorldFIP network was implemented by the authors to test both: the specification made from LOTOS language (section 2.1) and the nodes, which were designed and implemented using development tools and prototype boards provided by manufacturers, as well as the included FPGA chips (section 1.2).

## **3. CAN systems and devices classification**

Communication protocols applied to automotive applications are valued by semiconductor manufacturers, standardisation organisations and the Society of Automotive Engineers (SAE).

CAN systems and devices can be classified attending to several criteria (ISO, protocol version, mailbox structure and integration degree) specified next.

The ISO classification is the simplest: low speed (< 125 Kbps) and high speed (> 125 Kbps).

CAN is a message passing protocol and follows a producer-consumer model [Etschberger, 2001], thus, only when an important event occurs, the producer accesses the medium to communicate this information to the other nodes sending a message. The medium access is resolved by means of the message identifier priority, so that the highest priority message takes the medium and transmits the messages, and the transmission of the lower priority messages is postponed. The size of this identifier depends on the CAN specification version: 11 bits for version 2.0A and 29 bits for version 2.0B [bos, 1991].

The communication is basically carried out by means of diffusion. All the nodes receive the message and filter it according to the identifier configured in the buffers (known as mailboxes), which are responsible to establish the identifiers of the transmitted and/or received messages. When a message is received and its identifier has been previously configured on the mailbox, the CAN node verifies its integrity. Only if this message is error free, it would be transferred to the node controller, and in the opposite case it would be rejected.

Depending on the mailbox structure, two implementations are possible: BasicCAN and FullCAN structure.

CAN systems and devices can also be classified by its hardware integration degree in: Stand alone and SLIO (Slave-IO).

## 4. CAN educational tools

In this section the hardware and software tools, which comprise the educational environment for CAN fieldbus learning, are described.

### 4.1 esd development system

The development system manufactured by esd GmbH (electronic system design) comprises: a PC-CAN interface board (model *CAN-PCI/331*), a digital input/output module *CAN-CBM-DIO8* with 8 channels, I/O wiring including terminations and T-connectors, drivers, *CANscope* application software for Windows NT/95/98/2000, and CAN introduction and operating manuals. All system nodes are electrically insulated and comply with the CiA standards [esd, 2000].

The PC-CAN interface board uses a microcontroller of type 68331, which takes care of the local CAN data management, and a *Philips SJA1000* CAN controller, that complies with the ISO 11898 [ISO 11519-2, 1994, ISO 11898, 1993], allowing a data transfer rate of 1 Mbit/s. This controller is also compatible with CAN 2.0A and CAN 2.0B.

The 8 digital I/O module is configurable and comprises a *SAB 80C515C* microcontroller as a SLIO-CAN device, and a *PCA82C250T* CAN transceiver.

The *CANscope* application software is based on the CANopen standard, and allows the configuration of the development system and the monitoring of CAN events. In addition, the software development tools for application layer solutions includes the CAN device controllers and the NTCAN Application Programming Interface (API), available for Windows and UNIX environments, as well as for embedded and real-time systems.

## 4.2 IXXAT development system

This development system comprises the following IXXAT GmbH manufactured devices: 3 intelligent PC-CAN interface boards (*iPC-I 320/PCI*) for PCI bus systems, and licenses for *canAnalyser/32*, *CANopen Configuration Studio*, *CANopen Node Manager*, *CANopen Master API* and *Programming Client* software.

The PC-CAN interface board includes a *Dallas DS80C320* microcontroller, and *Philips SJA1000* and *Intel 82527* CAN controllers. This CAN adapter board enables a data transfer rate of 1Mbit/s, complies with the ISO 11898 and is compatible with CAN 2.0A and CAN 2.0B.

The IXXAT development system offers a very powerful software tool for the development, testing and service of CAN-based networks. Thus the *canAnalyser/32* software supports the functionality of layer 7 protocols (CAL, CANopen, DeviceNet and SDS), allowing the display and transmission of CAN objects with 11 and 29 bits identifiers. The *CANopen Configuration Studio* software is useful to configure and manage CAN networks (with the CANopen standard application layer implementation), and the other tools provide extended functions. Among the latter stand out, the protocol-specific representations of messages in CANopen based systems (*CANopen Client*), the transmission and reading of PDO- and SDO-objects (*CANopen Node Manager*), or individual programming of new functions with the aid of a programming module (*Programming Client* and *CANopen Master API*).

The development system also contains a modular I/O device manufactured by WAGO, with CANopen standard capabilities (*WAGO I/O System 750-307*), that works like a fieldbus node with CANopen standard support. This modular I/O device includes:

- A bus coupler, that constitutes the link between the fieldbus and the other modular devices with I/O capabilities.
- Special function modules, which capture and condition data to be processed. Particularly, they consist of two digital input modules (*750-402*) and two digital output modules (*750-504*), each one with 4 channels.
- Termination module, without I/O capabilities, but necessary to avoid failures.

## 5. Teaching environment for CAN networks

In this section two hardware projects are described: the design and development of CAN nodes (for general applications), and the development of an educational environment for CAN protocol teaching (including several devices communicated by means of this bus).

### 5.1 CAN nodes implementation

The CAN nodes are designed and developed as general application CAN devices, enabling the study of CAN and other derived protocols (i.e. TTCAN). The hardware implementation of these CAN nodes uses a *8751* microcontroller of the Intel MC-51 family to administrate the node, a *Philips SJA1000* CAN controller and a Texas Instruments CAN transceiver (*UC5350N*) to insulate the controller and bus voltages. The developed software presents a user interface that can directly establish connections to the microcontroller, and send messages to the CAN bus. This hardware implementation also allows the test and verification of the developed CAN hardware (section 1.5.2). On the top of figure 3 the CAN network based on the implemented nodes is depicted.

The educational environment for CAN protocol teaching consists of one monitoring station (protocol analyser) and four field devices that comply the CAN standard. The monitoring station captures frames going through the field-bus to perform further processing on them. For this purpose, a personal computer with a high speed CAN interface board manufactured by NSI is used, along with the CAN board controller software. This board is based on the *Intel 82527* CAN controller and *Philips 82C250* transceiver, which enables a data transfer rate of 1 Mbit/s, complies with the ISO 11898 and is compatible with CAN 2.0A and CAN 2.0B. The protocol analyser software was developed using the Borland C++ Builder programming environment, running on Windows 2000/NT/9x operating system.

The CAN devices are implemented on a printed circuit board that includes analog and digital inputs/outputs, a SLIO-CAN controller (*Philips P82C150*), and the required elements to configure the CAN controller and to attach the device to the bus.

### 5.2 Developed CAN hardware

The developed hardware is based on Programmable Logic Devices (PLDs) technology and implements a general purpose CAN device for control applications. Following the protocol description, the CAN functionality was implemented and executed on a Xilinx Field Programmable Gate Array (FPGA). This functionality was programmed using VHDL (VHSIC Hardware Description Language) by means of the Xilinx Foundation development kit, and im-

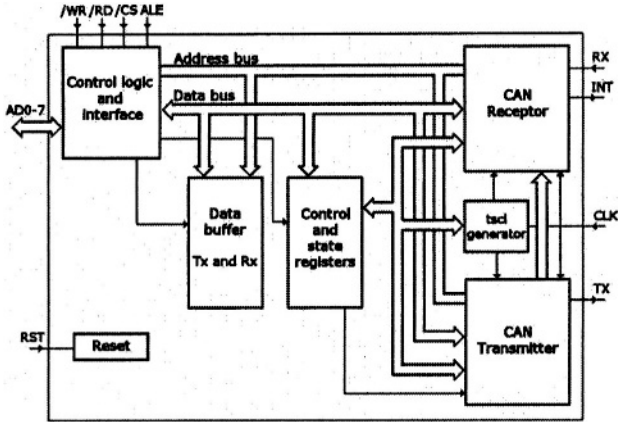


Figure 2. CAN protocol hardware description

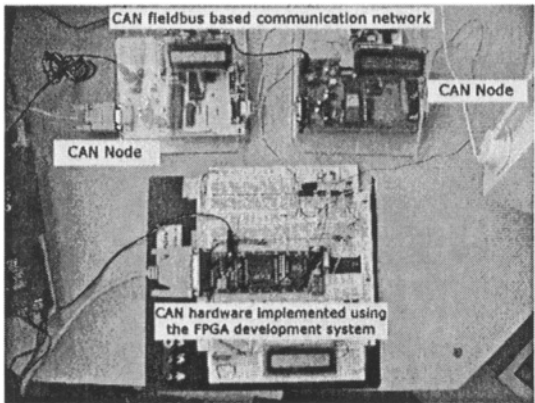


Figure 3. CAN fieldbus based communication network

plemented on a *XS40* prototyping board manufactured by Xess Corporation. This board includes a Xilinx 20,000-gate *XC4010XL* FPGA and an Intel *8051* microcontroller. Figure 2 shows the block diagram of the CAN protocol hardware description accomplished using the following Xilinx Foundation design tools: schematic design, LogiBLOX and VHDL. In addition, this hardware description of the CAN protocol can be easily adapted to other FPGAs.

The final system (Fig. 3) is formed by the CAN nodes (section 1.5) and the CAN hardware implemented using the FPGA development system.



## **6. Conclusions and future works**

In this paper the state-of-the-art in fieldbus technology is presented, based on the authors' experience working with engineering students of information technology at university grades. Proposed curriculum topics are: the introduction of FDTs in the formal description of industrial communication protocols, and the design and implementation of nodes for selected fieldbuses (i.e.: PROFIBUS, WorldFIP and CAN) by means of development tools and prototype boards provided by manufacturers. Particular lines of research, about the implementation of fieldbus nodes using FPGA chips and RTOs for emulators, are also pointed out.

Given its growing impact all over the automotive industry, an introduction about the CAN fieldbus is presented, along with the software and hardware tools that configure training environments for this protocol. The authors' more recent work on fieldbus technology is shown through the two different CAN nodes, developed by them employing the previously presented tools.

Future developments with great potential in IT university education include fieldbus training applied to automotive industry and others, such as:

- Application of formal description techniques (FDTs) to the development of integrated circuits for fieldbus networks, based on the generation of VHDL code (language for hardware description) directly from their FDT's specification [Gauer, 2002].
- Current demands for installing automotive electronic devices for communication and control have opened a trend towards the use of more electric power aboard cars in order to satisfy the future needs. Current technical reports forecast a move from 12V batteries to 36V ones, for loads with high-power (42V) and low-power (12V) demands [Frank et al., 2001].
- The software integration of FDTs and new packets that provide new functionalities of fieldbuses such as access networks, is an exciting area for new developments. Among them it is worth mentioning the work carried out by some European industrial groups such as IDA [Kaplan, 2001] and OSEK, that are devoted to integrate fieldbus technologies for Web/Internet and automotive software standards for embedded controls respectively.
- Another interesting area of research is the wireless fieldbus technology, not only in the fast moving sector of automotive industry where ITS (Intelligent Transportation Systems) programs are known worldwide, but also throughout every industrial sector [Pottie and Kaiser, 2000].

## Acknowledgments

This work has been sponsored by two R&D projects from the following entities: Research General Directorate of the MCYT, Ref. TIC2001-3701-C02-01, Central Government (Madrid, Spain); and Presidency Department, Ref. PGIDT01TIC30301PR, Autonomous Government (Galicia, Spain).

## References

- (1991). *CAN Specification Version 2.0*. Robert Bosch GmbH.
- (2000). *CAN-Starter-Kit for Windows NT/2000, Windows 95/98 Rev 1.2*. esd electronic system design GmbH.
- CENELEC EN 50170 (1996a). *General Purpose Field Communications System*, volume 2/3 (PROFIBUS).
- CENELEC EN 50170 (1996b). *General Purpose Field Communications System*, volume 3/3 (WorldFIP).
- Etschberger, K. (2001). *Controller Area Network: Basics, Protocols, Chips and Applications*. IXXAT Press.
- Frank, R., Kolanko, F., and Phillips, T. (2001). Higher-voltage vehicles generate new demands for power management. *Electronic Design*, pages 71–76.
- Gauer, R. (2002). The SoC design platform and its application in the automotive industry. *Information quarterly of ARM*, 1(1):30–32.
- ISO 11519-2(1994). *Road Vehicles - Low-speed serial data communication - Part 2: Low-speed controller area network (CAN)*. ISO 11519-2.
- ISO 11898 (1993). *Road Vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication*. ISO 11898.
- Kaplan, G. (2001). Ethernet's winning ways. *IEEE Spectrum*, pages 113–115.
- Lías, G., Valdés, M., Domínguez, M., and Moure, M. (2000). *Field Programmable Logic and Applications (FPL 2000)*, volume 1896, chapter Implementing a fieldbus interface using a FPGA, pages 175–180. Springer-Verlag.
- Mariño, P. (2003). *Enterprise Communications: Standards, Networks and Services*. RA-MA, 2nd edition.
- Mariño, P., Domínguez, M., Poza, F., and Nogueira, J. (1999). *Fieldbus Technology (FET'99)*, chapter Formal Description Software for WorldFIP Industrial Fieldbus, pages 157–163. Springer-Verlag.
- Mariño, P., Nogueira, J., Sigüenza, C., Poza, F., and Domínguez, M. (2001). The profibus formal specification: A comparison between two FDTs. *Computer Networks Journal, Elsevier Science*, 37:345–362.
- Mariño, P., Poza, F., Domínguez, M., and Nogueira, J. (1997). *Transformation-Based Reactive Systems Development (ARTS'97)*, chapter Specification and Verification of a Real-Time Field Bus with Formal Description Languages, pages 415–429. Springer-Verlag.
- Mariño, P., Poza, F., and Nogueira, J. (1996). *Balanced Automation Systems II (BASYS'96)*, chapter Information System Based on Industrial Standard Protocols, pages 201–208. Chapman & Hall.
- Marsh, D. (1999). Surviving the fieldbus wars. *Electronic Design*, pages 34–46.
- Pottie, G. and Kaiser, W. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43:51–58.