Centro de Sistemas Telemáticos e Computacionais

Instituto Superior Técnico

NavIST Group
Fault-Tolerant Real-Time Distributed
Systems and Industrial Automation

CAN Dual-Media Redundancy: Prototype Development

CSTC Technical Report RT-01-05

J. Rufino, G. Arroz

September 2001

Centro de Sistemas Telemáticos e Computacionais - NavIST Group
Fault-Tolerant Real-Time Distributed Systems and Industrial Automation

CAN Dual-Media Redundancy: Prototype Development

Project PRAXIS/P/EEI/14187/1998 - DEAR-COTS
Distributed Embedded ARchitecture using Commercial Off-The-Shelf components

Technical Report: CSTC RT-01-05
Authors: J. Rufino, G. Arroz
Date: September 2001

Avenida Rovisco Pais, 1049-001 Lisboa - PORTUGAL, Tel: +351-218418397/99, Fax: +351-218417499.
NavIST WWW Page URL - http://pandora.ist.utl.pt.

CAN Dual-Media Redundancy: Prototype Development*

José Rufino
ruf@digitais.ist.utl.pt
IST-UTL†

Guilherme Arroz
egsa@alfa.ist.utl.pt
IST-UTL

Abstract

Standard fieldbuses are today a cost-effective solution for distributed computer control systems. However, the efficient implementation of fault-tolerance mechanisms on the simple fieldbus environment presents non-negligible problems. In particular, the network infrastructure has to be resilient to partitions, offering high levels of reliability against temporary medium faults and availability in the presence of permanent faults. This can be achieved through network media redundancy. This report analyzes the design and implementation of dual-media mechanisms in the Controller Area Network (CAN) fieldbus.

1 Introduction

Continuity of service and determinism in message transmission delays are two fundamental requirements of fault-tolerant real-time applications, that must be fulfilled through the use of some form of network-level redundancy.

Fieldbuses are in essence a technology whose area of application requires continuity of service. Fieldbuses are widely used in systems intended for real-world interfacing (i.e. integrating sensors and/or actuators) which are specially sensitive to the availability of the network infrastructure, a problem that we address in the context of CAN, the Controller Area Network [3]. CAN is a low-cost fieldbus, with widespread acceptance in control applications, able to extremely robust operation. However, resilience to medium partitioning has to be provided as an extension to the standard specification.

Arguing with the high costs of other solutions, an existing commercial redundant CAN design uses a self-healing ring/bus architecture [5]. Though such a scheme provides resilience to open/short-circuits in the physical wiring, it does not solve the problem of CAN continuity of service efficiently: ring reconfiguration takes time (it can last as long as $100ms$) and meanwhile the network is partitioned. Bus media redundancy does not exhibit those shortcomings and do represent a natural design choice, but efficient mechanisms for its implementation were not published at the time [8].

In [9], we have presented a *Columbus' egg* idea that opens room for a simple implementation of bus media redundancy in CAN. However, to be of practical use such scheme should be enhanced with mechanisms handling not only medium partitions but also stuck-at and omission failures, as discussed in [9]. The present report addresses how those mechanisms can be efficiently implemented and integrated in a CAN media selection unity, using standard off-the-shelf components [2].

*This work was partially supported by FCT, through Project PRAXIS/P/EEI/14187/1998 (DEAR-COTS).

†Instituto Superior Técnico - Universidade Técnica de Lisboa, Avenida Rovisco Pais, 1049-001 Lisboa, Portugal. Tel: +351-218418397 - Fax: +351-218417499. NavIST Group CAN WWW Page - http://pandora.ist.utl.pt/CAN.

2 CAN Media Selection Unity

The architecture of the *CAN media selection unity*, to be inserted between the redundant media interfaces and a standard CAN controller, is depicted in Figure 1. Apart from replication, commercial CAN components are used. No assumption is made concerning the use of a particular CAN controller and multiple solutions are allowed for the physical layer: inexpensive two-wire differential cabling used together with classical (non fault-tolerant) transceivers [6] or fiber optics technology [7].

All provisions for media replication management are made as extensions to the standard. Resilience to Medium crash (stuck-at/broken) and omission failures is achieved through the mechanisms described in [9].

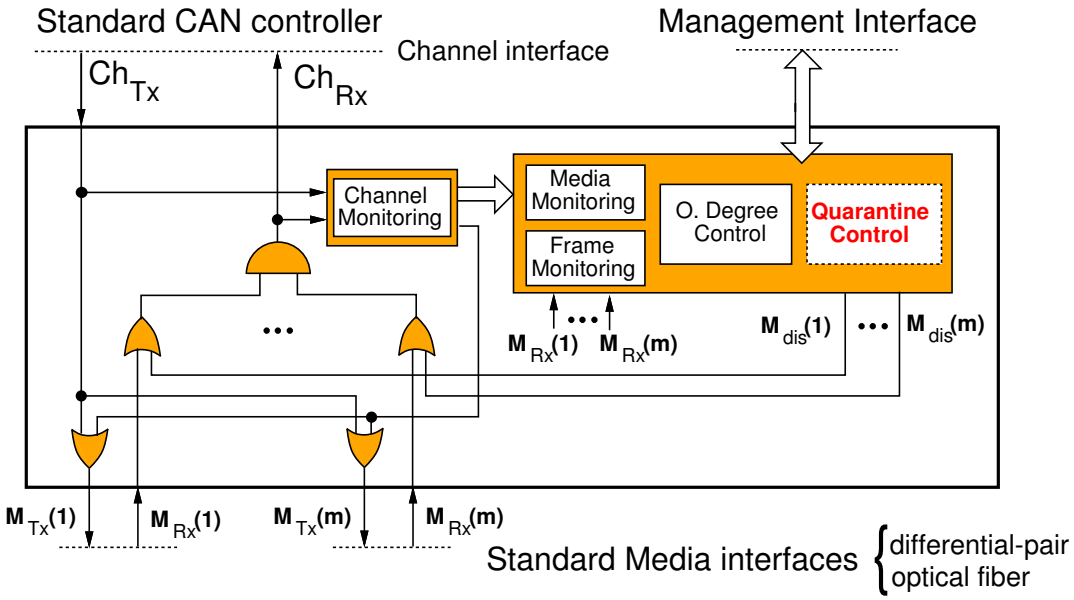

Figure 1: Architecture of the CAN media selection unity

The central component of the CAN media selection unity (Figure 1) is the AND function specified by the *Columbus' egg* strategy. The remaining functions are structured in a modular way [10], as follows:

- **Channel monitoring** – analyzes Channel activity with regard to the occurrence of permanent node failures and checks whether or not a frame transfer succeeds;
- **media monitoring** – analyzes the state of each Medium interface with regard to the occurrence of permanent failures;
- **frame monitoring** – compares Channel activity with the behavior of each Medium interface, thus allowing the detection of omission errors;
- **omission degree control** – this module uses the indications provided by Channel and frame monitoring modules to keep track of the omission degree of each Medium;
- **quarantine control** – aims to control the effects of single-medium errors that nevertheless affect the behavior of media monitoring functions at all media interfaces.

The attributes of the CAN protocol can and should be exploited in the design of media redundancy management modules, in order to keep complexity low [10]. For example, stuck-at dominant and recessive failures are mutually exclusive events, and this may be used to reduce the complexity

of the *media monitoring* circuitry. A similar approach can be followed in the design of the pattern recognition machinery that identifies the error signaling and frame termination sequences.

At a slightly different level, the assignment of pre-defined constant values to a relevant set of protocol-related parameters allows to simplify the implementation of CAN monitoring mechanisms, at the cost of a lower flexibility in the configuration of those parameters.

Complexity issues are of major relevance to the integration of the media selection unity in a single, inexpensive, medium capacity programmable logic device. A prototype of the media selection unity, aiming its integration in a *field programmable gate array*, is currently being specified in VHDL[1]. A preliminary specification of such a design is described in [4].

Assuming a fundamental role to the integration of the CAN media selection unity in the system architecture, the set of functions defined for the corresponding management interface are summarized in Table 1.

Invocation Primitives (can-msu.req)			
Description		*Parameters*	
Initialize		*baud*	bit rate signaling parameters
		k_m	media omission degree bound
Notification Primitives (can-msu.nty)			
Description	*Condition*	*Parameters*	
Omission degree exceeded	$M_{Od} > k_m$	m	failed Medium
Stuck-at-dominant Medium	M_{stk-d}	m	failed Medium
Stuck-at-recessive Medium	$M_{idle} \wedge \neg M_{ds}$	m	failed Medium
		mid	message identifier
Medium partition	$M_{idle} \wedge M_{ds}$	m	failed Medium
		mid	message identifier
Stuck-at-dominant Channel	Ch_{stk-Tx}	node transmitter failure	

Table 1: CAN media selection unity management primitives

The operation of the media selection unity is started upon the issuing of the layer management action specified by the *initialize primitive* (cf. Table 1), that defines the rate of data signaling in the bus and the allowed media omission degree bound.

The notification primitives of Table 1 specify how the failures detected by the CAN media selection unity are signaled to higher layers. Those layer management notifications allow the implementation of high-level applications, providing diagnose functions extremely helpful to maintenance activities. For example: the signaling that a given Medium has failed calls for an action to repair the network infrastructure. Furthermore, the functionality provided by the CAN media selection unity allows to distinguish a stuck-at-recessive from a medium partition failure and the set of parameters signaled upon failure detection permits a high-level diagnose application to establish a node connectivity matrix, useful to pinpoint the location of the failure in the network cabling.

Crucial for such a kind of applications, the correctness of every message identifier captured at the PHY-MAC interface and signaled upon the detection of a given Medium failure, is secured. Such notifications are issued only in the absence of Channel errors.

[1]Very High-Speed Integrated Circuits (VHSIC) Hardware Description Language. A fairly comprehensive introduction to VHDL can be found in [1].

References

[1] P. Ashenden. The VHDL cookbook. Department Computer Science, University of Adelaide, South Australia, July 1990. Available from the following URL: ftp://chook.cs.adelaide.edu.au/pub/VHDL-Cookbook.

[2] DEAR-COTS Consortium. DEAR-COTS: a Distributed Embedded ARchitecture using Commercial Off-The-Shelf components - Technical Annex. PRAXIS XXI Program - Project Proposal, July 1998.

[3] ISO. *International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network for high-speed communication*, November 1993.

[4] A. Matias. Design and implementation of can media redundancy mechanisms. IST Graduation Project Report, Advisors: J. Rufino and G.Arroz, Instituto Superior Técnico, Lisboa, Portugal, February 2000. (preliminary version - in portuguese).

[5] Red-can a fully redundant can-system. NOB Elektronik AB Product Note - Sweden, 1998. http://www.nob.se.

[6] Philips Semiconductors. *PCA82C250 - CAN Controller Interface*, April 1994.

[7] M. Rucks. Optical layer for CAN. In *Proceedings of the 1st International CAN Conference*, pages 2.11–2.18, Mainz, Germany, September 1994. CiA.

[8] J. Rufino. Dual-media redundancy mechanisms for CAN. Technical Report CSTC RT-97-01, Centro de Sistemas Telemáticos e Computacionais do Instituto Superior Técnico, Lisboa, Portugal, January 1997.

[9] J. Rufino, P. Veríssimo, and G. Arroz. A Columbus' egg idea for CAN media redundancy. In *Digest of Papers, The 29th International Symposium on Fault-Tolerant Computing Systems*, pages 286–293, Madison, Wisconsin - USA, June 1999. IEEE.

[10] J. Rufino, P. Veríssimo, and G. Arroz. Design of bus media redundancy in CAN. In D. Dietrich, P. Neumann, and H. Schweinzer, editors, *Fieldbus Technology - System Integration, Networking and Engineering*, pages 375–380. Springer, September 1999. (Proceedings of the Fieldbus Conference FeT'99, Magdeburg, Germany).