iCC 1996

3rd international CAN Conference

in Paris (France)

Sponsored by

Motorola Semiconductor
National Semiconductor
Philips Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

Paavo Kärkkäinen*, Jorma Kahila and Pasi Pulkkinen****

*** Oulu Institute of Technology, Oulu Finland**
**** Orion Corp. Normet, Finland**

FAULT DIAGNOSTIC UTILIZING CAN COMMUNICATION

Abstract

Machine control software are in general highly embedded in target systems. Distributed processors and standard CAN-buses have provided suitable tools to design machine diagnosis and fault tolerance based on user requirements or safety aspects.

In this paper some general and on target machine designed diagnosing elements and monitoring functions are explained. The surveyd target machine is just developed for loading explosive into holes in mining applications. Because of highly dangerous work environment machine fault detection and diagnostic maintenance are of crucial importance.

Both continuous operation and rapid responses are peculiar to the developed machine. On the basis of real time maintenance the error sources are divided in critical requiring immediate user operations and in slowly developing faults caused mainly by component wearing.

The distributed computer architecture provides means to implement fault tolerating systems and especially CAN contollers facilitate to embed diagnosis messages both in measuring data and in special safety functions. The distributed control makes it possible to diagnose the functions of the machine more effectively, but, however, on the other hand the distributed control is a new and a serious risk factor. Without sufficient checking intelligent subsystems may behave unexpectedly: erroneous interpretation of system state or fault sensor can trigger the actuator on at wrong time with dangerous consequences.

Introduction

Computer chips with integrated communication facilities such as CAN-controllers have made it possible to design distributed systems both from economical aspects as well as from the basis of system safety and fault-free operation. Concerning the machine safety and usability the distributed control has a benefit to supervise machine functions more effectively, but at the same time distributed software form a new and difficult manageable risk factor. Without continuos and sufficient checkings the machine with distributed computer controlled operations may under fault conditions behave unexpectedly: subsystem can trigger the actuator on or off at wrong time which may disturb whole system behavior thoroughly.

The aim of fault detection and diagnosis is in machine to detect faults which might lead to

failures. Fault detection is the process of checking a system for erroneous states. Fault detection is based on the principles of redundancy (mainly to detect hardware faults) and diversity (software faults).
Special methods applicable are assertion progrmming, N-version programming and the safety bag technique and on hardware level by introducing sensors.

Continuous operation and rapid response are typical requirements to machine controls. Faults can be generated at different rate. From the point of real time maintanance and system diagnosis the faults emerging rapidly are critical. In particular fault forecasting (calculating trends), fault correction, maintenance and supervisory actions may be supported by Artificial Intelligence (AI) based systems in a very efficient way in diverse channels of a system, since the rules might be derived directly from the specifications and checked against these. In practice the main bottle neck is the defect of sufficient diagnostic knowledge in order to forecast especially the less frequently emerging faults.

The usability of the diagnosis functions mainly depends on the amount and quality of diagnostic knowledge. This can be divided into design and empirical knowledge.

There are no special safety requirements concerning distributed computer controls. Also there are no generally accepted requirements on safety operations in various risk levels of system functions. As a basic design rule can be stated that the machine control system has to be designed and implemented so that under normal working conditions the system malfunction does not cause dangerous situations.

Standards concerning serial communication and field buses are still under development. The satndard draft IEC 1491 handles serial data link for real-time communication between controls and drives. Basic principle in failure management is that actuators are equipped with supervising functions which quarantee that system powers down in situations where correct response to master commands is prohibited.

CAN is a serial multimaster communication protocol primarily intended for real-time control. The CAN protocol specification mainly covers the data link layer (DLL) of a communication system. CAN protocol has proven to be very reliable and error free communication form. However, there are needed some services above CAN-protocol which are not implemented in CAN-spesification. Higer level services are e.g. block transfer, message scheduling and determination of message identifications. An essential safety requirement is that in the system it is not allowed two nodes having same identification code. Another basic service principle applied in the application described in this paper is that receiving node always acknowledges the message by sending back another message.

CAN-protocol does not detect situation where some node has detached from the bus. These kind of failures can be detected checking nodes by regular message inquiries.

Diagnostic functions in CAN-based machine control

Intelligent CAN-nodes can maintain statistic about work hours and operation times. During

of machine. Faults which arise during normal operations can be detected by continuous state checking of intelligent nodes. If fault is detected machine control system allows only safe and necesary machine operations.

Fault forecasting is as an aid in machine maintanance and it is based on satistical analysis of certain measurements and on empirical evaluations. The most critical data for later analyzis of machine condition can be divided into

- control parameter values and variations,
- inconsistency in machine operations,
- evaluation of sensor information.

Basic diagnostic components in a mining machine consists of four CAN-nodes one being as a master (and user interface unit). Diagnostic system comprises diagnostic properties, diagnostic data collection tasks, a service terminal and an interface to service computer for on line data collection. A special analyzis software is needed for interpretation of collected parameter data. During services it is possible to add intellgent sensors into system which are not necessary in machine control.

Diagnostic properties

Diagnostic functions can be embedded into machine control system and these can be called during normal operations or invoke as a special service and measurement task. A difficult task is to rank the criticality of faulted components which could result in injury, damage, or system degradiation. Certain operations need special attention and these can't be started without suitable sensor states and system response to diagnostic measures. Next some possible parameters for a special mining machine maintanance are represented:

- water pressure
- emulsion pressure
- water percent
- emulsion temperatures in tank in pump
- voltage levels in system
- charging time per hole
- actuator control values
- motion coordination errors during charging.

These parameters together with statistics of normal machine operations can serve as a part of machine maintanance in evaluation of the condition actuators, valves, mechanical components or electric system.

Run-time diagnosis is stored in the flash memory of diagnostic node together with process parameters and these can be examined via user interface or analyze later in PC-application program.

The tasks of diagnostic node are:

- fault evaluation and assertation of error messages
- system halting
- reinitialization of node operations after error acknoledgement
- storing diagnostic data
- maintain memory operations (programming, erasing, and memory size limits)

Each node gathering sensor data checks and filters sensor information based on expected limits. Fault sensor states may be achieved in the value domain or in the time domain on different levels. Critical sensors can be wired to different nodes and diagnostic node checks the possible discrepancy in sensor interpretations.

Also CAN communication bus needs diagnosis as it contains mechanically easily corrupted components. The most important diagnosis methods are failure detection of CAN physical layer and node checking.

CAN bus supports especially well off-board diagnostics, because it can utilize available message structures . These can be used to command valves, read memory areas, sensor states, AD- and DA -conversions etc.

Case Example: **Emulsion Charging Mining Machine**

This machine is intended to be used in mines to fill holes drilled in the rock with explosive emulsion (see figure 1). Working in mines is very tough and handling explosive is an additional risk factor which together set very strict requirements for safety and reliability of the machine.The most critical component in the machine is the emulsion pump and its operation is examined with several sensors connected to CAN-system and parallel this is an additional hard-wired logic to detect main fault conditions.

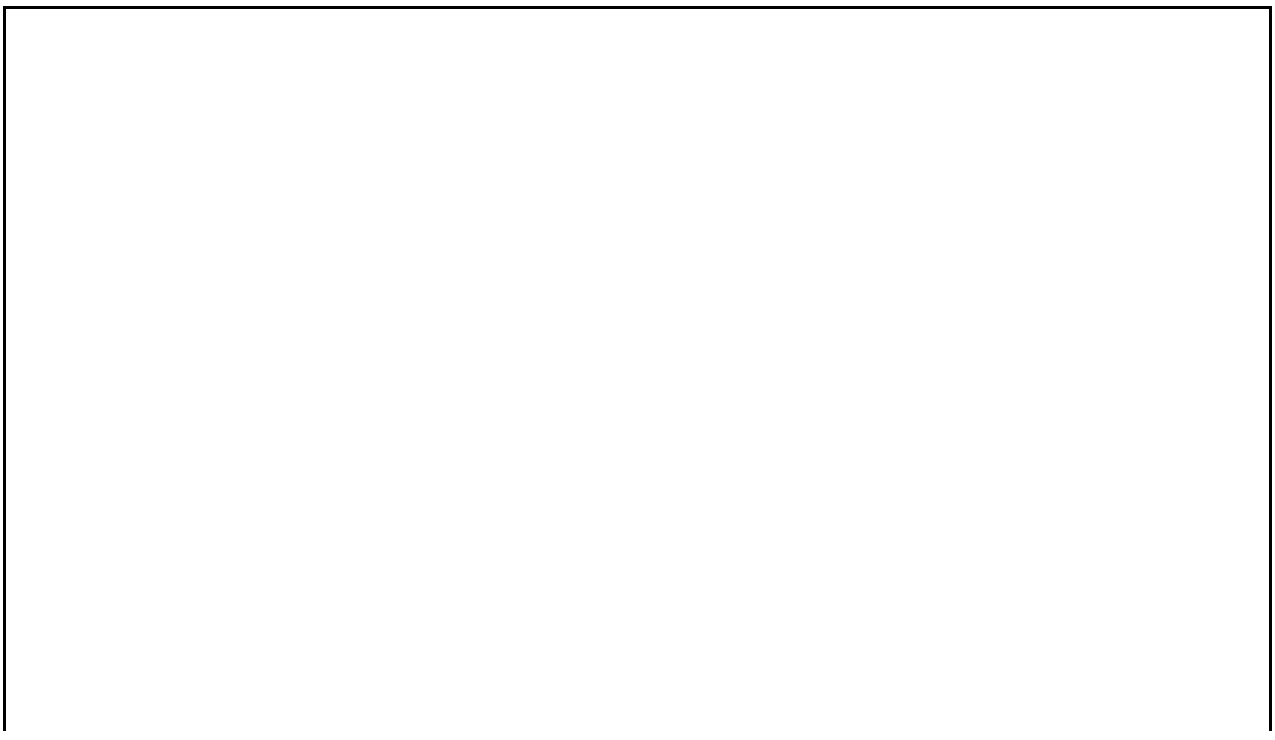

Figure 1. A mining machine for emulsion charging.

A typical work cycle of the machine is first to position the hose to the start of the hole and in automatic operation the operator selects the hole number with preprogrammed data. The needed hole parameters are hole length, hole diameter, and the length of the primer and based on these parameters the control system feeds the hose and controls the emulsion amount into the hole. There is also semiautomatic operation mode available for such hole lengths which are unknown beforehand. Unknown disturbances in programmed situations are always possible and under these conditions the operator intervention must be easily selectable. E.g. the hole may be blocked or some big cave opening to the hole drains emulsion over calculated amount. The diagnosis system monitors machine internal sensors and if some node notifies error machine operations are stopped and user must do certain error recovery operations before system can continue interrupted work .

CAN Control Nodes

Controlling machine operations are divided to four CAN nodes one being master and user interface unit. One node is termed as supervisor, because it uses redundant sensor information with two other nodes. Each node is built around 80C592 controller. The keypad unit has additional RS-232 interface to communicate with a PC or a service terminal. PC interface is used both for programming and testing and to diagnose collected information from machine operations and sensor states.
In Figure 2 is depicted the block diagram of nodes interfaced with sensors and actuators. The nodes are named based on their functional activity as ”charger”, ”feeder” ,”supervisor, and ”keypad”. Charger node takes care of emulsion pump and water pump . Feeder node controls hose movements which are realized by two actuators. Supervisor node checks the states of various sensors. The validation of sensor state is interpreted according to system state which is maintained in master unit.

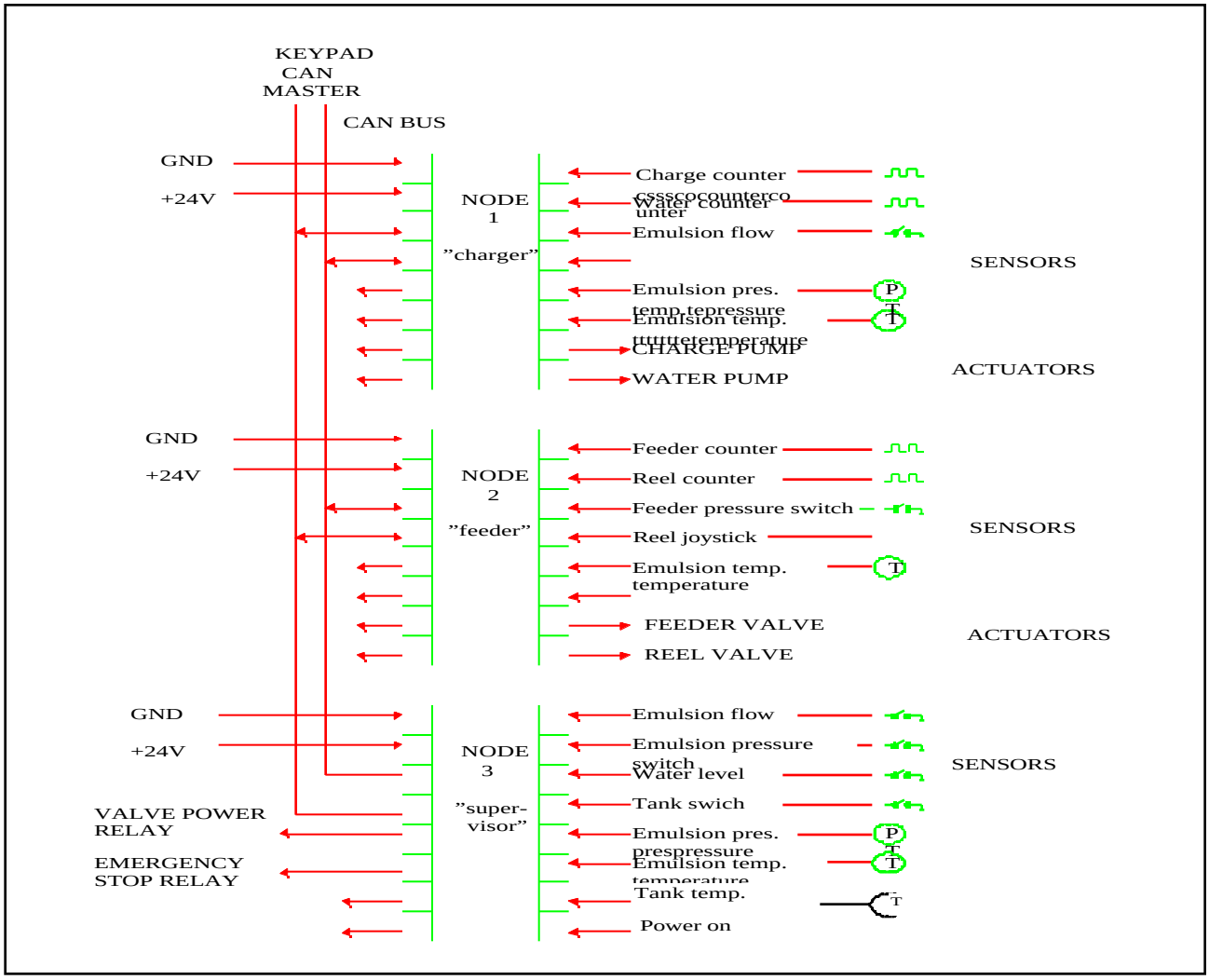

Figure 2. Sensor/actuator interface diagram in a mining machine.

Message Structures

The keypad coordinates the message transfer in the CAN-bus. Each node has been assigned a special address (id-code), which determines the message receiver. There are three types of messages used: control messages , measurement messages and error state inquiries. Control messages are sent only when required but measurements are monitored at regular intervals from each node (about 200 ms intervals). The message exchange is part of the diagnosis: the node has to respond to each message at least by sending back status information. Also if the node does not receive any message within certain time it stops active tasks and transits to wait state. If master receives error code it requests the node to send once more the message before accepting the situation.

a. Control message

A typical control message contains actuator position/volume value, speed value and actuator mode and initial state settings. One byte is reserved for special actuator parameter. An

example of control message CAN_Com(Set_Emu) is given to command emulsion pump to start in automatic mode.

Master → Charger node

Id-Code	DLC	message id	data 1	data 2	data 3	data 4	data 5	data 6
1	7	21	12	34	40	1	3	0
			Volume count	Speed	Mode	Status	Extra parameter	

Charger node → Master

Id-Code	DLC	message id	status

b. Measuremet message

The measurement request contains only the node address and source id of desired measurement. Athe structure of measurement request CAN_Com (Read_Emu) and response from node are below.

Master → Charger node

Id-Code	DLC	message id

Charger node → Master

Id-Code	DLC	message id	data 1	data 2	data 3	data 4	data 5	data 6	data 7
1	8	16	0	89	35	45	3	1	0
			Volume count	Speed	Pressure	Mode	State	Status	

c. Error State Inquiry

As a response to control or measurement messages master receives also status information which the node sets based on sensor state analysis. The errors have been divided into serious faults and recoverable failures. An example of error codes which are consided critical are listed in Table 1.

Table 1. Some critical error codes

Code	State
1, 2, 3	Node 1,2,3 does not respond

5	master CAN transmit error
6	Emulsion pressure overlimit
7	Emulsion temperature high
8	Water level low

If master obtains in status code a bit indicating e.g. that pressure overlimit has happened it commands the charger node to switch off the power from the valves and all other nodes to transfer to error state. If desired the master node can ask additional measurement values, which the error state initiating node stored during error interpretation.

In error state 6 either the supervisor node or charger node or both have recorded four succeeding pressure values which were above the set overlimit and time value since the emulsion pumping was started. A message structure from the node to master indicating error source and related information is following : (below is as example the field values in case of pressure sensor)

Id-Code	DLC	message id	Error source	data 1	data 2	data 3	data 4	data 5	data 6
1	8	31	6	110	100	102	96	0	25

In this message as error source is emulsion pressure sensor and in data fields are pressure values (11.0, 10.0, 10, 2, 9.6 bars) and situation arose 25 seconds after pumping started.

Conclusions

Some general principles how to utilize distibuted CAN controllers in machine diagnosis are outlined in this paper**.** CAN has certain benefits in flexibility and it provides for maintanance equipments basic services for data reading and following. Complex systems can be divided into subsystems which are more easily controllable and interactions among subsystems reveals effectively possible error sources.

On-line machine diagnosis is achieved by checks of individual nodes according to system state and sensor data available. The results of these checks are stored and associated machine operational data for later analyzis.

In a mining machine message transfer and node responses are checked in every message exchange. This might not be suitable in applications where communication delays due to message acknoledgements cant’t be tolerated.