Controller Area Networks: Evolution and Applications

H. F. Othman, Y. R. Aji, F. T. Fakhreddin, A. R. Al-Ali

Computer Engineering Department
American University of Sharjah, UAE
aali@aus.edu

Abstract

This paper presents Controller Area Networks (CAN), their architecture, protocol, and standards. As a result, an overview of CAN applications, in both the industrial and non-industrial fields, is surveyed. We also propose the extension of CAN applications to home automation. The proposed system is a stand-alone single-chip embedded system equipped with 5 CAN ports to monitor and control home appliances locally. Home owners can also remotely access their homes via GPRS modem to control and monitor their home appliances.

Index Terms — Controller Area Networks (CAN), Home Automation, Home Appliances, Industrial Applications, Non-industrial Applications, GPRS Network.

I. Introduction

Nowadays, major communication networks can be divided into four types, namely: IP Core Network/Internet, Wireless LAN, 3G/4G Cellular Network and Ad-hoc PAN [1]. The common usage for these networks is to carry text, audio and video content. Recently, some of these networks have been utilized in industrial automation to monitor and control industrial plants [2-5].

Another type of networks is the Controller Area Network. CAN is intended as a communication network between the control units in vehicles. Nowadays, CAN applications are gaining ground and it is extending to industrial automation including marine and aircrafts electronics, factories, cars, trucks and many others [8].

The backbone of the Controller Area Network is a fast serial bus that is designed to provide a reliable, efficient, and a very economical link between sensors and actuators. CAN uses a twisted-pair cable to communicate at speeds up to 1 Mbits/sec, with up to 40 devices. CAN was originally developed to simplify the wiring in automobiles. In the past, automobile manufacturers used to connect devices in vehicles using point-to-point wiring systems. As more electronics and controllers are deployed to monitor and control vehicles, wiring started to become more complex, bulky, heavy and expensive. Automotive industry starts to reduce massive wires complexity with dedicated CAN link that provides low-cost, robust network, and multi-master communication system.

Figure 1 shows the efficiency and the wiring-reduction caused by implementing CAN among multiple devices.

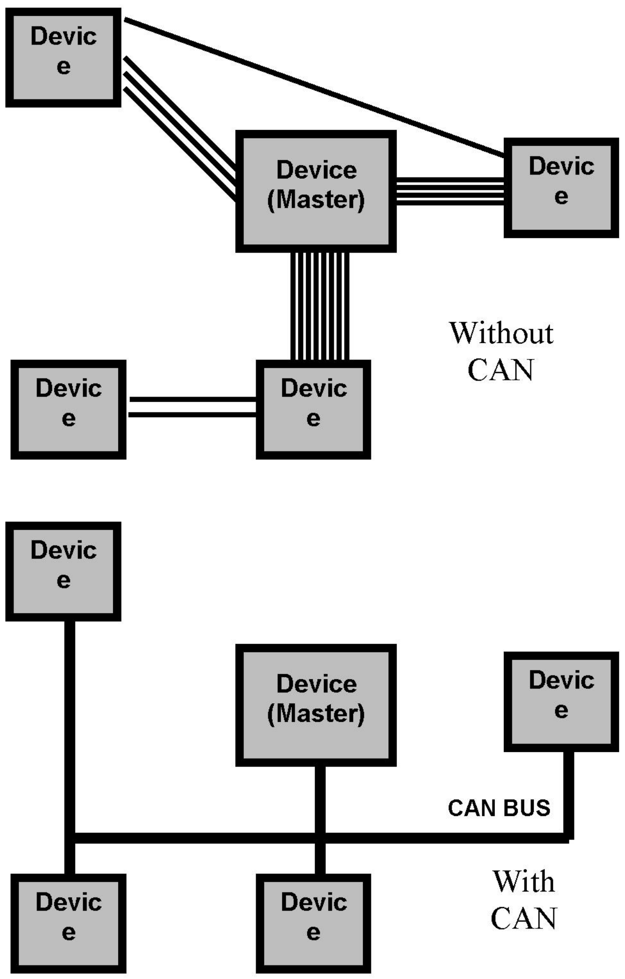

Fig.1. Wiring-reduction, Cost reduction with CAN

This paper presents a comprehensive overview of controller area networks, their architecture, protocol, and standards. Also, this paper gives an overview of CAN applications, in both the industrial and non-industrial fields. Due to CAN reliability, efficiency and robustness, we also propose the extension of CAN applications to home automation.

II. CAN Architecture, Standards and Protocol

CAN was originally developed by Robert Bosch (Germany, 1986) when Mercedes requested a communication system between three electronic control units in vehicles. Point to point communication was not suitable anymore, and the need of using a multi-master communication system became imperative. Although this origin can be traced to automotive industry, industrial automation rapidly showed the need of using such a popular bus system [6-7].
A CAN port is a two-wire, half duplex, high-speed network system that can reach a throughput up to 1 Mbits/sec [6]. Data, control commands and devices status can be transmitted and/or received in well-structured frames. Theoretically, CAN is capable of linking up to 2032 devices on a single network; due to hardware limitation, only 110 nodes can be linked-up to construct a single network. [6]

Similar to the traditional OSI model and IP model, CAN has a 4-layer protocol: physical layer, transfer layer, object layer, and application layer as shown in figure 2 [7].

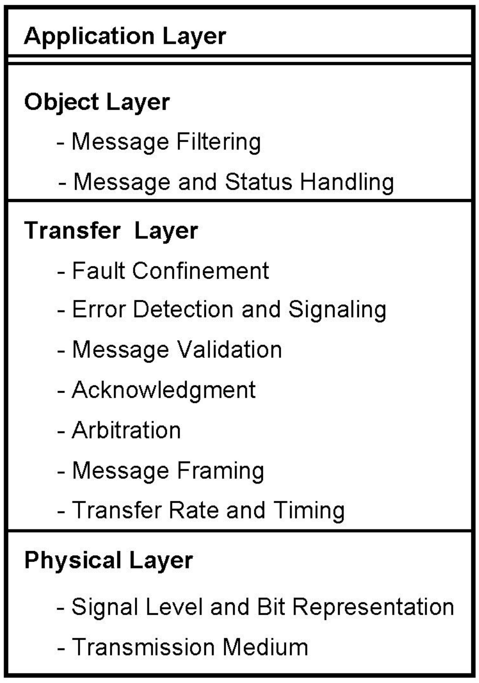

Fig.2. Layered Architecture of a CAN node [7]

The physical layer defines how the signals are transmitted. The transfer layer defines the Kernel of the CAN protocol. It is responsible for presenting and accepting the received/transmitted messages to/from the upper layer. These messages are sent and received using the following properties: bit timing, message framing and arbitration, acknowledgment, error detection and signaling, in addition to fault confinement. The object layer is responsible for message filtering and handling. The object layer and the transfer layer are both combined as the data link layer defined by the ISO/OSI model. The application layer is the upper layer, through which the user interferes. [7].

CAN (Version 2.0) has two different standards: CAN 2.0 A, standard CAN, using 11 bits for node identification; and the other standard is CAN 2.0 B or extended CAN, using 29 bits for node identification. With 11 bit, 2,048 unique messages are possible, whereas 536 million unique messages are possible with 29 bit identifier [6]. In a controller area network, all nodes – recipients – can see all messages and can accept or ignore any messages according to the system design.

There are two ISO standards classifying the two CAN's physical layer in terms of data rate; ISO 11898 which can handle speed up to 1 Mbit/sec and ISO 11519 that can handle speed up to 125 Kbit/sec, as shown in table 1.

	Standard	**Signal Rate**	**Identifier**
Low-Speed CAN	ISO 11519	125 kbps	11-bit
CAN 2.0 A	ISO 11898 (1993)	1 Mbps	11-bit
CAN 2.0 B	ISO 11898 (1995)	1 Mbps	29-bit

Table.1. CAN versions

The CAN protocol has been designed using four different frame types, namely: data, remote, overload and error frames [7].
Bits in a CAN protocol can be one of the following:

- "0": dominant bit
- "1": recessive bit

If a 0 and a 1 are sent simultaneously, in two different messages, then bit "0", (dominant) will be taken into consideration, discarding the message with the "1" bit.
The nodes in the network can have one of the following states:

a) Error active: a unit in this state can take part in the bus communication, by transmitting and receiving messages. If this node detects an error, then it sends an ACTIVE ERROR FLAG.
b) Error passive: if a unit has had several wrong transmitted or received messages, then it is not able to send an ACTIVE ERROR FLAG. It can send only a PASSIVE ERROR FLAG.
c) Bus off: whenever a unit has had several serious problems with the transmission of messages, it is temporarily out of service, i.e., it can't send nor receive any message.

Every node has two types of counts that determine in which state the unit is at a specific time:

a) Transmit error count
b) Receive error count.

These counts determine the state of the controller at any time. They are updated according to rules specified in the Bosch Specification for CAN. [7]

The data frame is used whenever the transmitter sends data to the receivers, with its own identifier. It is composed of the following, as shown in Figure 3 [7]:

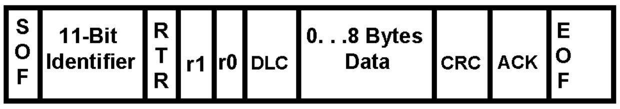

Fig.3. 11-Bit Identifier CAN standard data frame

- **SOF:** (Start Of Frame) this bits indicated the beginning of a DATA or REMOTE frame. It is a single dominant bit.
- **Identifier field** (either 11 or 29 bits):
  If CAN 2.0 A is used, the identifier is 11 bits.
  If CAN 2.0 B is used, the identifier is 29 bits, divided into 2 parts:
  - 11 bits, for compatibility with the 2.0 A version
  - The 18 remaining bits are located after the RTR bit.
- **RTR** (Remote Transmission Request)
  This bit is used to differentiate between a REMOTE and a DATA frame:
  - for a data frame, this bit has to have a dominant value (value of "0"); the identifier mentioned will represent the sending node, that publishes its data on the network
  - for a remote frame, this bit should be a recessive bit (value of "1"); the identifier mentioned will represent the node from which data is requested. A remote frame does not contain any Data Field.
- **r0** and **r1**: these two dominant bits are left for future expansion
- **DLC** (Data Length Code, 4 bits): this field indicates the number of bytes following the control field, i.e., the number of bytes in the Data Field.
- **Data Field:** it contains the data sent. The data can have 0 up to 8 byes maximum.
- **CRC Field:** it consists of 2 fields:
  a) CRC (Cyclic Redundancy Check) Sequence: this 15 bits field is a check code, that is used for error checking in the sent message
  b) CRC Delimiter: this is a recessive delimiter bit
- **ACK Field:** it consists of 2 fields:
  a) an ACK (Acknowledgment) Slot bit: it is transmitted as a recessive bit (value 1), and is over written (by a value of 0) as soon as a node, reading the message, acknowledges an error free message, i.e., a successfully received message.
  b) An ACK Delimiter: this is a recessive delimiter bit.
- **EOF Field:** This End Of Frame field consists of seven recessive bits, delimiting the end of a data or remote frame.

The remote frame is used whenever a *receiver* node requests data from any other source. A remote frame differs from the data frame in the following, as shown in figure 4 [7]:
- It does not contain any data field
- The RTR field should be a recessive bit.
- The Data Length is the value of the corresponding DATA frame
- The identifier is the one representing the node from which data is requested.

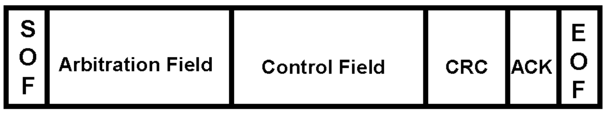

Fig.4. 11-Bit Identifier CAN remote frame

An overload frame is sent whenever a delay is needed in the network, as for example, whenever a *receiver* needs some time before receiving another *data* or *remote* frame. An overload frame consists of 2 fields, as shown in figure 5 [7]:
- Overload flag: it consists of six dominant bits
- Overload delimiter: it consists of eight recessive bits

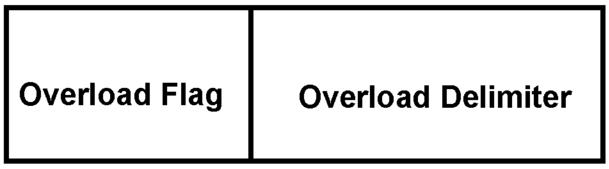

Fig.5. CAN overload frame

An error frame consists of 2 fields: Error flag and Error delimiter, as shown in figure 6 [7]. The error flag can be one of the following types:
- Active error flag: it consists of six consecutive dominant bits. It is used if the node transmitting this message is in active error state.
- Passive error flag: it consists of six consecutive recessive bits. It is used if the node transmitting this message is in passive error state.

The error delimiter consists of 8 recessive bits.

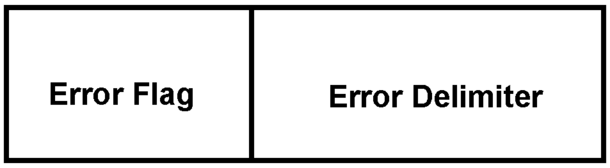

Fig.6. CAN error frame

III. CAN Applications

The main CAN application fields include passenger cars, trucks and buses, off-highway, cargo and passenger trains, marine electronics, aircraft and aerospace electronics , factory automation, industrial machine control, lifts and escalators, building automation, medical equipment and devices, non-industrial control and equipment [8].

- In passenger cars, the automation industry uses CAN for the vehicle engine management as the in-vehicle engine management network (IVN). This application includes managing the engine, the body electronics such as the door, the roof control, the AC, and the car lightning. The European and the American car manufacturers are using CAN-based IVNs [8].
- In the engine management, CAN networks connect several electronic control units (ECUs). Most of the European car manufactures have installed CAN high-speed networks (i.e. 500 kbit/sec) in their power engine systems [8].
- For power-train applications and for communication between the truck and the trailer, CAN is used as in-vehicle network. CAN is also used as an embedded control network in some of the truck-based superstructure control systems (i.e. firefighting equipment or concrete mixers) [8].
- In addition, several CAN-based IVNs are installed in some trucks and are interconnected via gateways. The reason why this is done is not to mix functions in one physical network in order to prevent interferences and disturbances [8].
- Medical devices and equipments, ranging from X-Ray machines to complete operating rooms, passing through patients beds, are using embedded CAN network to control and manage all required functions. Furthermore, visualization software, voltage control and many other types of equipment, which can be found in any complete hospital control systems, are being networked via CAN [8].
- Non-industrial machines, for example the vending machines (from which products may be sold independently of the presence of an actual vendor), the ATM machines (Automatic Teller Machines, from which money may be credited or debited independently of an actual bank employer), or copy machines and printers (examples of office equipments) may be networked using CAN [8].
- In marine electronics, CAN networks are widely used in ships and boats as embedded network in sub-systems. Dedicated marine devices with CAN connectivity are available. Also, CAN-based ship automation systems and marine sub-systems with CAN interface exist. In a ship automation system, several physical CAN networks may exist together [8].
- In factory automation, the demand for off-shelf plug-and-play is increasing; many applications are using CAN systems. These applications typically include conveyors and recoding of the produced data to be used by the end-user. In such applications, CAN is used to network machines and process control units together with the sub-systems of the factory [8].
- Also, CAN is used as an embedded network for controlling industrial machines ranging from textiles production, to printing, without forgetting the packaging machines. Single devices (I/O modules), and sub-systems are linked together using this kind of controlling networks. These applications can be grouped under what is called "motion control oriented" system [8]. In fact, these systems are based on a stand-alone intelligent drive, providing a compact solution and more flexibility, because of their built-in multi-tasking capability. Cereal manufacturers are models of "motion control oriented" system: such machines may be reconfigured depending on the size of the tray used for example. This configuration is in fact based on an independent operating system: a Programmable Logic Controller (PLC) controls the overall process and sends command signals to the machine under configuration. [8]

IV. Home Automation Proposed System

Home automations have been implemented using GSM-networks [9], power line communication, X10 [10] and many others.

In this work, we propose a stand-alone single-chip embedded system equipped with 5 CAN ports to monitor and control home appliances locally. An advantage of using CAN is that devices can send AND receive data (in contrast to previous applications, where devices may only receive messages from a master). Home owners can also remotely access their homes via GPRS modem to control and monitor their home appliances.

The proposed system consists of two parts. One part is installed at home, thereafter *home-controller*. The other part is the *remote-access* part. The *home-controller* is a stand-alone single-chip microcontroller equipped with 5 CAN ports and a GPRS modem [11].The *remote-access* part consists of a server equipped with a GPRS modem that can be accessed via the public wireless mobile network and the Internet by home owners.

- For demonstration purposes, only four appliances (Dish Washer, AC, Dryer and Heather), in addition to a lighting system are selected as shown in figures 7 and 8.

- The home-controller and remote-access software is divided into two parts. One part is developed using the embedded system's native language which can control the home appliances through the CAN ports, checking the devices status and transmitting them to the remote access sever through a GPRS modem. The other software part is to manage the process at the server side. It receives and transmits the home appliances status via the GPRS public network and the Internet to home owners. These ports are utilized to monitor and control home appliances locally such as dish-washer, washing machine, dryer, air-condition, heater, and lights. (Figure 8)
- The control methodology of the selected home appliances is translated into a *control driver* written in the micro-controller native language. (Figure 8) A temperature sensor is connected to an analog input of the micro-controller, with the necessary signal-conditioning and conversion. The control driver loop checks the temperature reading every 10 seconds and compares it with the thresh-hold value selected by the home owner. The micro-controller turns on or off the AC depending on the temperature value. The home lights are switched on when the control driver checks the micro-controller's clock and compares it with the selected time by the home-owner, which is the sunset time. The dish-washer, heater and dryer are switched on or turned off upon user's command via GPRS network. An important issue, in this context, is that the home appliances – devices – can talk together based on CAN standards. For instance, the AC might request the heater's status, whether ON or OFF, and adjust its cooling temperature accordingly.

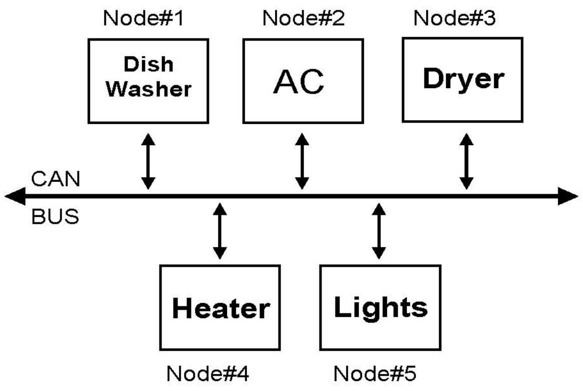

Fig.7. General System Architecture

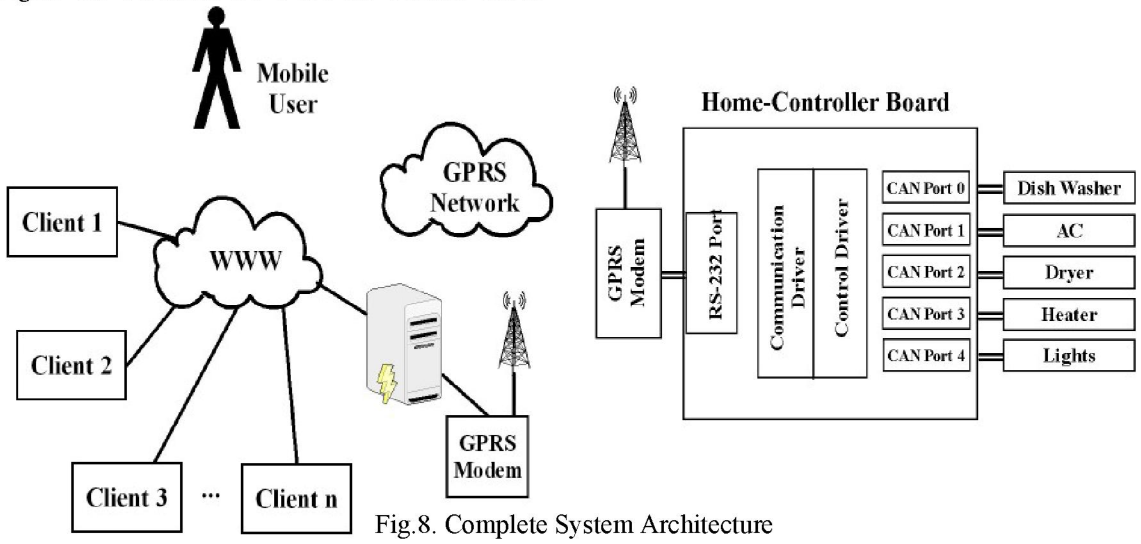

Fig.8. Complete System Architecture

VI. Conclusion

Controller Area Networks applications are emerging and gaining high ground in many applications from automobile industry to automation and factory industries. To mention a few, we can cite passenger cars, trucks and buses, marine electronics, factory automation, industrial machine control, and non-industrial control. In this paper, controller area networks' architecture, protocol and standards were presented. It also stressed on the key features of CAN systems that enabled the proposal of extending CAN applications towards home automation and control. The proposed system proves the ability of CAN to accommodate home automation into its scope.

References

[1] Th. Zahariadis," Evolution of the Wireless PAN and LAN standards", Computer Standards & Interfaces, Vol. 26, No. 3, May 2004, Pp 175-185.

[2] Qiu, B. and Gooi, H. (2000) 'Web-Based SCADA Display Systems for Access vi Internet', IEEE

Transaction on Power Systems, Vol., 15, No. 2, pp. 681-686.

[3] I. Lin, H. Broberg, " Internet-based monitoring and controls for HVAC applications", Industry Applications Magazine, IEEE, Vol. 8, No. 1, Jan.-Feb. 2002 Pp. 49 – 54.

[4] R. Fan, L. Cheded, O. Toker," Internet-based SCADA: a new approach using Java and XML", Journal of Computing & Control Engineering Journal Vol. 16, No. 5, Oct.-Nov. 2005 Pp. 22 – 26.

[5] A. Z. Alkar, U. Buhur, "An internet based wireless home automation system for multifunctional devices", IEEE Transactions on Consumer Electronics, Vol. 51, No. 4, Nov. 2005 pp. 1169 – 1174.

[6] "Controller Area Network - CAN Information," http://www.algonet.se/~staffann/developer/CAN.htm, 3 November 2005

[7] Bosch, R. "CAN SPEC*IFICATION* (Version 2.0)," Germany: Stuttgart, 1991

[8] "CAN Application Fields," http://www.can-cia.org/applications/, Dec. 2005

[9] A. Alheraish, "Design and Implementation of Home Automation System", IEEE Transactions on Consumer Electronics, Vol. 50, No. 4, Nov. 2004

[10] Wolfstone Group. "Intro to X-10," http://wolfstone.halloweenhost.com/TechBase/x10int_X10Intro.html#WHAT, Nov. 2005

[11] "Ten Flash-Based 16-Bit Microcontrollers," http://www.motorola.com/mediacenter/news/detail/0,,1075_773_23,00.html, Jan. 2006