International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 3, Issue 5(May 2014), PP.24-29

Real-Time System Bus Architecture for Small Aircraft Using Can-Protocol

E.Giri Prasad[1], S.Arunvinthan[2], Niladri Shekhar Das[3]
(Avionics, AISST- Amity University, India)

Abstract:- In the Industrial and Automotive appliances, distributed systems are more frequently used. To facilitate and guarantee real-time performance in an exceedingly distributed system using CAN Protocol within the presence of infinite network errors. The time triggered protocol (TTP) is added onto a CAN bus configuration to improve the system reliability and stability. TTP-CAN may become an acceptable data bus system design standard to meet aviation design standard requirements. This paper provides an idea for the implementation of this network protocol between safety-critical Instruments with Reliability and Availability.

Keyword: Distributed systems, CAN Bus, Time-Triggered Protocol (TTP)

I. INTRODUCTION

Controller area Network, CAN, is a communication protocol developed in „80's by Bosch [1] with the aim of providing an appropriate serial communication by means of data exchange between the units on-board the vehicles and that was originally developed for automotive applications.

Arbitration and error detection are basic ideas of CAN bus standard. Arbitration is enforced by having a unique symbol field in a CAN message format [2]. Highest priority symbol takes the bus management. The identifier with low binary variety has the very best priority. Bus access conflicts are resolved by bit-wise arbitration, of the identifiers concerned, by every station observing the bus level bit by bit. If 2 nodes begin along the transmission, of recessive and/or dominant bits, the dominant transmitter will win. All different nodes become a beholder and don't re-attempt transmission till the bus is accessible once more. CAN networks are simple to install and utilize inexpensive circuitry so that almost every actuating and sensing unit in the automotive and industry appliances is equipped with a CAN interface [4].

II. CAN NETWORK STRUCTURE

The CAN specifies only the physical and data link layers of the ISO/OSI model. Each node connected to a CAN network has the structure of Fig.1. It is composed by the Bus Driver (BD), the Communication Controller (CC) and the Host Controller (HC). The bus is a pair of wires, usually denoted with CANL and CANH, which transmits the bits encoded in terms of differential voltage values. BD implements the physical layer and operates in a way so that a BD transmitting a low logical value (bit 0) overwrites a high logical level (bit 1) transmitted by any number of BDs. This means that the bit 0 is dominant over the bus and the bit 1 is recessive. CC implements the data link layer that manages the access of the node to the bus, checks the correctness of the received messages, performs the bit-stuffing and processes the transmission errors. CC sends or receives the bit stream of a message through BD and is connected to it by means of two wires, conventionally denoted as Tx and Rx. HC is usually a microprocessor or a microcontroller that implements the application layer, if any, and the appliance tasks generating the data to be sent to the other nodes and consuming the data coming from them. HC exchanges the data with CC by means of a serial or parallel bus. CC encapsulates the data coming from HC in the frame of the messages to be transmitted and extracts the data from the received messages to forward them to HC.

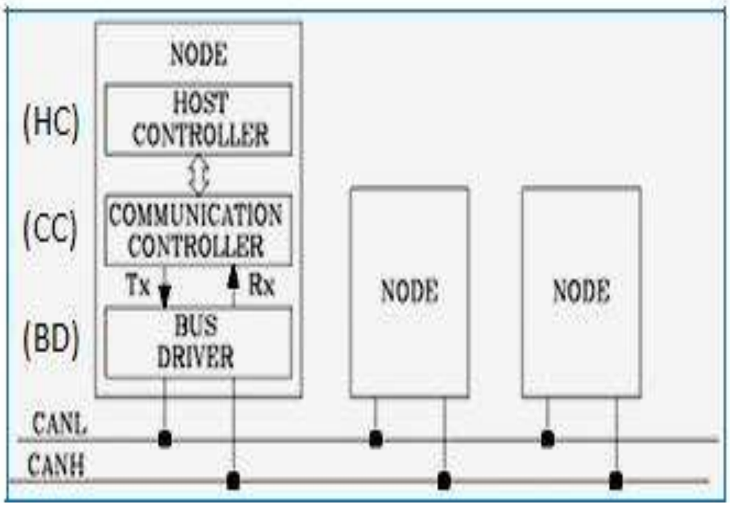

Fig-1 CAN Network Node Format

The CAN protocol does not have any application layer and the user must work out suitable routines to support the data exchange between data link layer and appliance. A CAN controller detects errors within a message and sends an error flag. Consequently, other CAN controllers, on the same bus, will take appropriate action e.g. discard the current message. The CAN standard defines five different error detection methods; two work on bit level, while three methods work on message level.

Bit Monitoring: Each transmitter on the CAN bus compares the bit currently on the bus with the bit originally transmitted, if found different then a bit error is signalled.

Bit Stuffing: No occurrence of more than five consecutive bits of the same level (except in case of error flag); else a stuff error is signalled.

Frame Check: Special fields in CAN frame must maintain certain logical level (i.e. the start of frame SOF) otherwise a frame error is signalled.

Acknowledgement Check: All nodes on the bus, that correctly receives a message, are expected to send a dominant level in the so-called Acknowledgement Slot in the message. If the transmitter didn't detect an Acknowledgement then Acknowledgement error is signalled.

Cyclic Redundancy Check: Each message features a 15-bit Cyclic Redundancy Checksum (CRC). Discrepancies between received and calculated checksums result in CRC Error.

The above-mentioned errors of CAN are overcome by the TT protocols [5]. TTP/C has been developed by the University of Vienna for the data exchange in real-time distributed systems.CC autonomously transmits a message at fixed time instants. The nodes access the bus according to the Time Division Multiple Access (TDMA) strategy, which assigns a time window to each node for the data transmission. In the TT protocols, the communication tasks are arranged in a periodical way. Each period is divided into time intervals, called time slots, during which only one node is entitled to access the bus. Consequently, there is no message collision and the transmissions are carried out without delays, thus resulting in a time-deterministic behaviour of the network. Moreover, the TT protocols are state-deterministic because in any time instant the network activities are known as, for instance, the node that occupies the bus. Time Trigger Protocol (TTP) was originally developed for high-dependability hard real-time applications, where timing error detection and fault-tolerance must be provided [5]. Software and hardware technology to merge TTP into CAN architecture for aircraft avionics system applications is studied.

For the Ethernet, each station waits in silence and then starts to transmit. If more than one station tries to transmit simultaneously, they are all detected to wait for a randomly assigned time period, and to try again in the next bus idle time. Ethernet is a carrier sense broadcast bus, since each station waits until the bus is idle, and monitors its own traffic to avoid collisions. CAN is also a carrier-sense broadcast bus in application.

The mass use of digital avionics, has been a standard for modern aircraft today [3]. The concept of the Integrated Modular Avionics (IMA) makes the structure of system more flexibly simplified shown in figure 2. In air transport system, the ARINC 629 data bus is organized to handle huge data exchange, however, the complicate system configuration in very expensive installation and maintenance is not suitable for small aircraft implementations. For general small aircraft, the data bus configuration and performance should be less expensive and simple. Among mature technologies, the controller area network (CAN) bus is a suitable choice. CAN bus is designed for ground transportation mobiles, the reliability and stability index is not as critical as that for aviation system. The reliability and stability concerns on CAN implementation into aviation system should be investigated with system redundancy considerations.

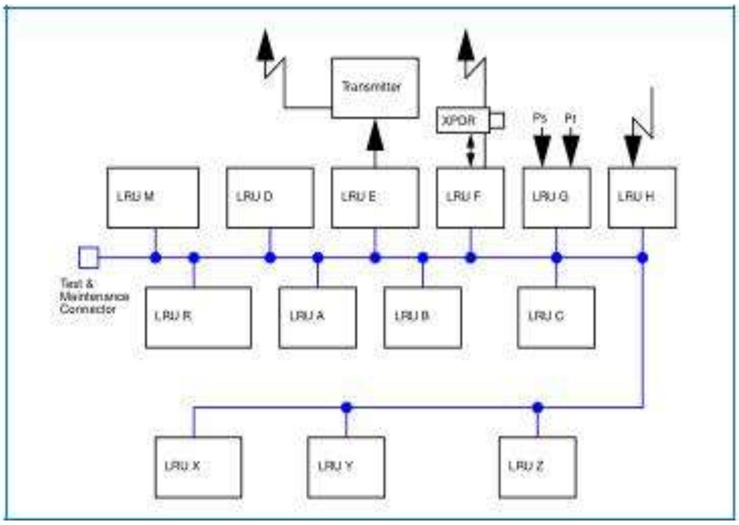

Fig-2 IMA Structure

CAN Message Format

Figure 3 shows the message format of CAN protocol [8]. A message in the CAN standard frame format begins with the start bit called the “Start of Frame” (SOF), and is followed by the Arbitration field to consist of the identifier and the “Remote Transmission Request” (RTR) bit used to distinguish the data frame and the data request frame, called remote frame. The following “Control field” contains the Identifier Extension (IDE) bit to distinguish the CAN standard frame and the CAN extended frame, as well as the Data Length Code (DLC) used to indicate the number of following data bytes in the “Data field”. If the message is used as a remote frame, the DLC contains the number of requested data byte. The following “Data field” is able to hold up to 8 data bytes. The integrity of the frame is guaranteed by the following Cyclic Redundant Check (CRC) sum. The bit in the ACK slot is sent as a recessive bit and is overwritten as a dominant bit by those receivers that have received the data correctly at this time.

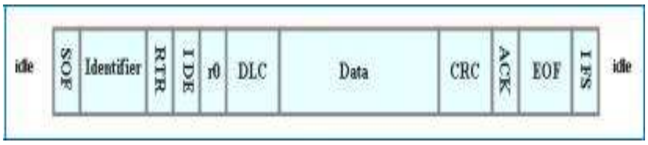

Fig-3 CAN Message Format

III. IMPLEMENTATION OF CAN BASED PROTOCOL IN AVIONICS

CAN offers significant advantages for highly reliable data communication in mission and safety critical applications that has made it attractive to aviation [3]. CAN network components are both well-tested and inexpensive due to incredibly high production volumes. CAN is a two-wire, multi-master broadcast serial bus standard that efficiently supports real-time control in distributed embedded systems [3]. Figure 4 CAN topology is a straight line with the LRUs connected to this line using small stub lengths. It has the following characteristics that can used to implement in Avionics

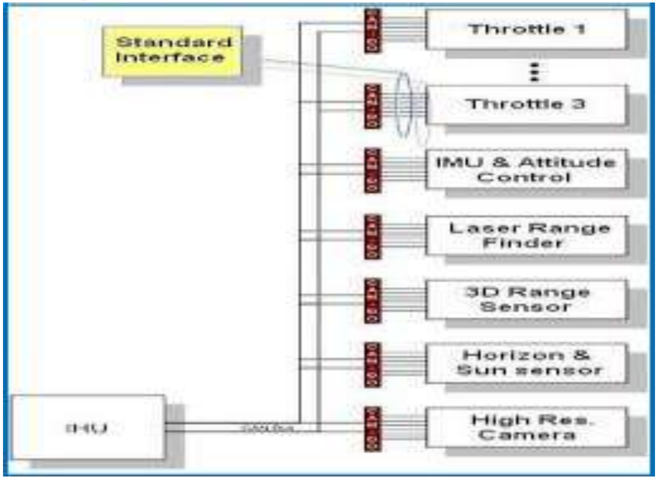

Fig-4 CAN Bus Architecture

Democratic Network: No master/slave relationship is required for normal operation. Every node on the bus has the same rights for participation in the bus traffic.

Self-Identifying Message Format: Information about the type of the data and the transmitting node is included in each message.

Message Numbering: Continuous numbering of transmitted messages supports coherent data processing in redundant systems.

Message Status Code: Information about the integrity of the data source is transported with each message.

Emergency Event Signalling Mechanism: Information about failures detected by built-in-test functions is transmitted by the affected node.

Node Service Mechanism: Addressing of specific nodes for integrity monitoring, data download, time synchronisation or interrogation using connection-oriented and connectionless services is supported.

Identifier Assignment: CAN protocol offers a predefined identifier distribution for normal operation data. More than one identifier distribution scheme is supported.

Ease of Implementation: The amount of code to integrate CAN protocol into safety or mission critical software is very small to minimize the effort for testing and certification.

Openness to Extensions: All definitions are extendable to provide flexibility for future enhancements and requirements of specific applications.

Free Availability: Absolutely no cost or royalties apply for use of CAN protocol or its specification.

CAN belong to the class of event-triggered protocols, where the control signals are derived primarily from non-time events occurring outside or inside the computer system. But TTP belongs to the class of the time-triggered protocols [7], where the control signals are solely derived from the progression of time CAN Bus shown figure 5. CAN allows messages to be assigned in priorities. High-priority messages will always be sent prior to low-priority messages, minimizing latency of high-priority messages. This event-driven approach is excellent for allowing separately developed processing elements on a shared bus.

Several papers had discussed to perform time-triggered communication [7] over CAN to achieve computability with respect to the timing behaviour of the communication system. In this implementation of centralized scheduling in CAN, the master node sends a special control message with a particular identifier, the master message, in the beginning of each cycle. This message indicates which data must be transmitted during that particular cycle.

In practical, the node with high frequency has higher priority. A message with a smaller identifier value is a higher priority message. In the experiment, if the information length is specially extended in cooperation with the change in message cycle to make a focus on conflict phenomenon.

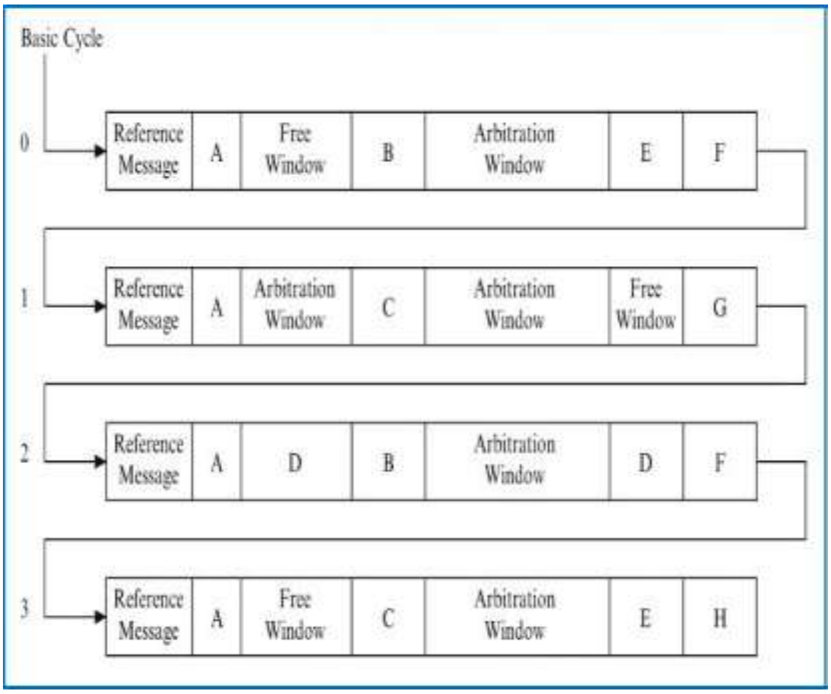

Fig-5Time Triggered CAN system

The reliability of a data bus system is a time dependent function describing by the probability that the lifetime of the system exceeds a predefined time. Reliability is also defined as the probability of no failures in an operation interval. Availability is the probability that the system is operational at the given point in time.

Reliability and availability are used as measures of performance and “yardstick” for quantitatively comparing the effectiveness of various fault-tolerant methods [6].

To demonstrate the reliability improvements using redundant channel, a hypothetical situation is used where the failure rate is identical for each channel. In this study, a failure rate ( ) of 0.0001/hour is chosen to generate quantitative information for comparison. The reliability of single bus is found from Equation (1).

$$_1 \quad = 1 - \quad 1 - \qquad (1)$$

Assuming that $= ^-$ then the Reliability equation becomes

$$_1 = ^- \qquad (2)$$

The graph is plotted for reliability of single bus from equation (2). From this graph, it is possible to observe that the reliabilities of each case remain almost the same value within the 10 hours of work, and in an acceptable range until the end of 100 hours of work.

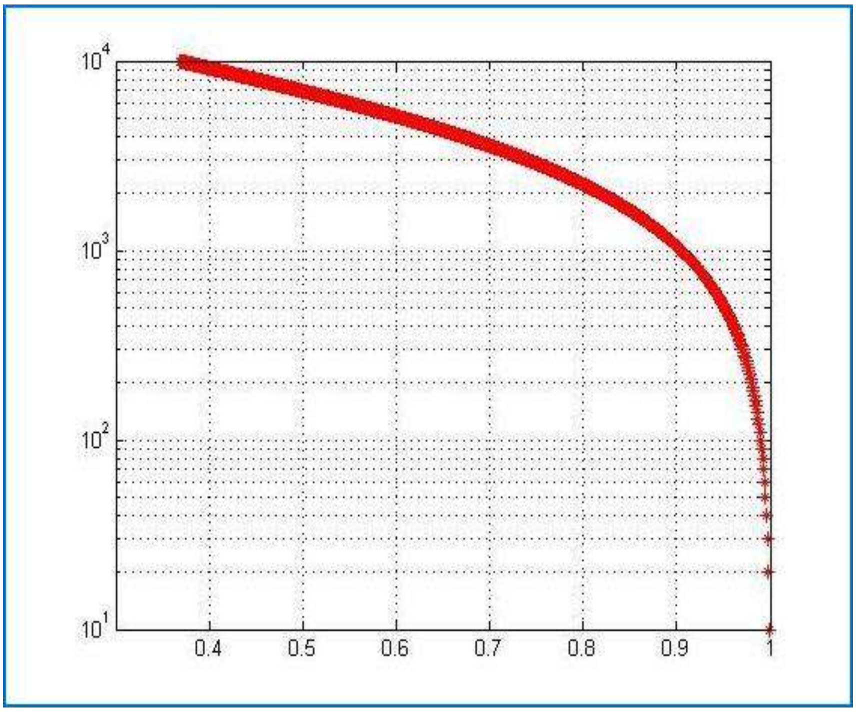

Fig-6 Reliability of CAN Bus

Availability is another measure of system performance on a system being operational at the instant of time t. It can be measured during operation by recording the downtimes and operating times for several failure and repair cycles. The availability is given by the ratio of the sum of the uptimes for the system divided by the sum of the uptimes and the downtimes.

The steady state availability is given by

$$= \frac{\quad}{\quad + \quad} = \frac{1}{1+}$$

Where MTTF- Mean Time to Failure and MTTR – Mean Time to Repair are the concerned parameters. If a system fails on an average of once a year, its MTTF is 8,000 hours, and is the repair rate of the system. For a constant failure and repair rates $= \frac{1}{\overline{\quad}}$ and $= \frac{1}{\overline{\quad}}$.

In the case of the system goes down for about 1 hour whenever it fails, and the MTTF is 8,000 hours $= 0.000125/h$, the availability would be $= 0.9999$

In the CAN based avionics network design, the complexity is not a great deal in fabrication and implementation.

IV. CONCLUSION

Thus the CAN protocol provides the efficient handling of time-triggered traffic because of the deterministic collision resolution mechanism for the requirements of small aircraft with reliable medium of communication, broadcast ability and multi-master, multi-receiver. To enhance the function of CAN bus for improving its stability and reliability, this paper provides an idea for the redundant architecture implementation.

The redundancy can be used in the firmware to detect the error and switch to backup. CAN testing approach is proposed to comply with industrial bit-level testing requirements.

Reliability and availability are used as measure of performance for the effectiveness in the single bus architecture. From the simulations, this bus architecture indeed enhances the fault tolerances of system and also satisfies requirement of safety, reliability, stability and availability.

ACKNOWLEDGEMENT

The authors are grateful to Amity Institute of Space Science and Technology [AISST] for its extensive support on this project. The authors are grateful to Mr.Mani Shankar Prasad for his great support to his research and his dedicated assistance to the study of Real Time System Bus Architecture for Aircrafts.

REFERENCES

[1]. Robert Bosch GmbH, "CAN Specification Version 2.0", September 1991.
[2]. C_CAN User"s Manual, Robert Bosch GmbH, 2000.
[3]. NASA Langley Research Centre: System Standard for the AGATE Airplane Avionics Data Bus, AGATE-WP01-001-DBSTD.
[4]. CAN - Controller Area Network CAN, "An In-vehicle Serial Communication Protocol", SAE Handbook 1992, SAE Press, pp. 20341–20355, 1990.
[5]. TTTech, "Time Triggered Protocol TTP, High level Specification Document", 2002.
[6]. M.Short and M.J.Pont, "Fault-Tolerant Time-Triggered Communication Using CAN", IEEE Transactions on Industrial Informatics, vol. 3, n.2, pp.131-142, May 2007.
[7]. T. Fuhrer, B. Muller, W. Dieterle, F. Hartwich, R. Hugel, and M. Walther. Time triggered communication on CAN. Techreport, Robert Bosch GmbH, 2000.
[8]. Marco Di Natale, Haibo Zeng, Paolo Giusto, Arkadeb Ghosal, "Understanding and Using the Controller Area Network Communication Protocol" Theory and Practice 2012.