International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011
Proceedings published by International Journal of Computer Applications® (IJCA)

Control Area Network for Reliable Car Communication

K.Punitha
Assistant Professor
Kalasalingam Universirty
Anand Nagar,Krishnankoil

S. Arun Kumar
PG Student
Kalasalingam University
Anand Nagar,Krishnankoil

N.Vijay Ganesh
PG Student
Kalasalingam University
Anand Nagar,Krishnankoil

ABSTRACT:

Automotive industry uses CAN as the in-vehicle network (IVN) for the engine management, the body electronics like door and roof control, air conditioning, and lightning, as well as for the entertainment control. The American and the Far East passenger car manufacturers have also started implementing CAN-based car automation. CAN networks used in engine management connect several electronic control units (ECUs). Most of the passenger cars are equipped with CAN-based multiplex systems connecting body electronic ECUs. These multiplex networks link door and roof control units as well as lighting control units and seat control units. The different CAN-based IVNs are connected via gateways1. In many system designs, the gateway functionality is implemented in the dashboard. The dashboard itself may equipped with a local CAN network connecting the different displays and instruments.

Keywords: CAN, Embedded Controllers, Sensors, Labview

1. INTRODUCTION

Today vehicle contains large number of electronic control systems. Bus- multiplexed structure of electronic system in electric vehicle has many advantages. The most important is the connection reduction (size and length of wires) that reduces the electromagnetic interferences, which agreed with electromagnetic compatibility. CAN is expected to be used in sensor networks for vehicle control, and industrial automation. CAN is a multi master network. It uses CSMA/CD+AMP (Carrier Sense Multiple Access/Collision Detection with Arbitration on Message Priority). Before sending a message the CAN node checks if the bus is busy. It also uses collision detection.

2. ABOUT CAN

Controller Area Network (CAN) is a broadcast, differential serial bus standard, originally developed in the 1980s by Robert Bosch GmbH, for connecting electronic control units (ECUs). CAN was specifically designed to be robust in electromagnetically noisy environments. It can be even more robust against noise if twisted pair wire is used. Although initially created for automotive purposes (as a vehicle bus), nowadays it is used in many embedded control applications (e.g., industrial) that may be subject to noise.

The CAN bus is primarily used in embedded systems, and as its name implies, is the network established among micro controllers. It is a two-wire, half duplex, high-speed network system and is well suited for high-speed applications using short messages. Its robustness, reliability and the large following from the semiconductor industry are some of the benefits with CAN. The messages it sends are (8 data bytes max) protected by a CRC-15.

CAN theoretically link up to 2032 devices on a single network. However, due to the practical limitation of the hardware (transceivers), it can only link up to110 nodes on a single network. It offers high-speed communication rate up to 1 Mbits/sec thus allows real-time control. In addition, the error confinement and the error detection feature make it more reliable in noise critical environment.

2.1 CAN Provides

1. A multi-master hierarchy, which allows building intelligent and systems. If one network node is defect the remaining network node is still able to operate.

2. Broadcast Communication: A sender of information transmits to devices on the bus. All receiving devices read the message and decide if it is relevant to them. This guarantees data integrity as all devices in the system which uses the same information.

3. Sophisticated error detecting mechanisms and re-transmission of faulty messages also guarantees data integrity.

2.2 CAN standards

The original specification is the Bosch specification. Version 2.0 of this specification is divided into two parts

- ❖ Standard CAN (Version 2.0A). Uses 11 bit identifiers
- ❖ Extended CAN (Version 2.0B). Uses 29 bit identifiers

The two parts define different formats of the message frame, with the main difference being the identifier length. There are two ISO standards for CAN. The difference is in the physical layer, where

- ISO 11898 has an upper limit of 1Mbit/second.
- ISO 11519 has an upper limit of 125kbit/second.

2.3 Principle

Data messages transmitted from any node on a CAN bus does not contain addresses of either the transmitting node, or of any intended receiving node. Instead, the content of the message is labeled by an identifier that is unique throughout the network. All other nodes on the network receive the message and each performs an acceptance test on the identifier to determine if the message, and thus its content, is relevant to that particular node. If the message is relevant, it will be processed; otherwise it is ignored.

2.4 Identifiers & arbitration

The unique identifier also determines the priority of the message. The lower the numerical value of the identifier, the higher the priority. This allows arbitration if two (or more) nodes compete for access to the bus at the same time. The higher priority message is guaranteed to gain bus access as if it were the only message being transmitted. Lower priority messages are automatically re-transmitted in the next bus cycle, or in a subsequent bus cycle if there are still other, higher priority messages waiting to be sent.

Each CAN message has an identifier which is 11 bits (CAN 2.0A) or 29 bits (CAN 2.0B). This identifier is the principle part of the CAN arbitration field, which is loaded in the beginning of CAN message. The identifier identifies the type of message, but is also the message priority. The bits in a CAN message can be sent as either high or low. The low bits are always dominant, which means that if one node tries to send a low and another node tries to send a high, the result on the bus will be a low. A transmitting node always listens on the bus while transmitting. A node that sends a high in the arbitration field and detects a low knows that it has lost arbitration. It stops transmitting; letting the other node with a higher priority message, continue uninterrupted.

Two nodes on the network are not allowed to send messages with the same id. If two nodes try to send a message with the same id at the same time arbitration will not work. Instead, one of the transmitting nodes will detect that his message is distorted outside of the arbitration field. The nodes will then use the error handling of CAN, which in this case ultimately will lead to one of the transmitting node being switched off (bus-off mode).

3. ARM MICROCONROLLER

The ARM (Advanced RISC Machine) architecture is a 32-bit RISC processor architecture developed by ARM Limited, that is widely used in a number of embedded designs. Because of their power saving features, ARM CPUs are dominant in the mobile electronics market, where low power consumption is a critical design goal. Today, the ARM family accounts for approximately 75% of all embedded 32-bit RISC CPUs, making it one of the most widely used 32-bit architectures in the world. ARM CPUs are found in all corners of consumer electronics, from portable devices to computer peripherals. The Fig.1 shows the 32-bit microcontroller growth.

The use of RISC processors over CISC processors in embedded systems today is wide spread and seems to be the trend of the future. This is because when it comes to embedded systems, RISC has many advantages over CISC both in hardware and software implementation of these embedded systems. Some of these benefits of RISC are:

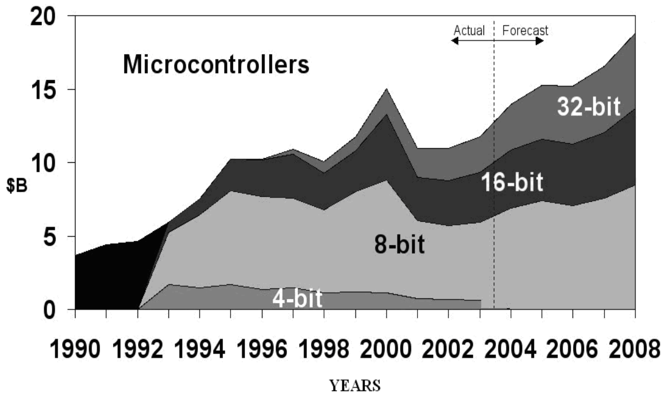


Fig.1- 32 bit Microcontroller Market

- Very fast responses to non-deterministic events
- Simpler assembler coding
- High throughput
- Low power consumption

ARM architecture is developed to utilize the benefits of CISC and RISC by improving the code density and reducing the power consumption. ARM Limited has incorporated a novel mechanism, called the Thumb architecture. The Thumb instruction set is a 16-bit compressed form of the original 32-bit ARM instruction set, and employs dynamic decompression hardware in the instruction pipeline. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles. The RISC instruction set and related decode mechanism are much simpler than those of CISC designs.

ARM architecture has a unique combination of features that makes ARM very popular embedded architecture today:

- ARM cores are simple compared to most other general-purpose processors.
- A typical ARM chip contains several peripheral controllers, a DSP and some amount of on-chip memory.
- ARM ISA and pipeline designs are aimed to minimizing energy consumption.
- ARM architecture is highly modular: the only mandatory component of an ARM processor is the integer pipeline. All other components, including caches, MMU, FP unit and other co-processors are optional.
- ARM architecture provides a high performance for embedded applications.

4. 89C51 MICROCONTROLLER

The microcontroller used here is P89C51RD2BN. The expansion of the part number of this microcontroller is given in Fig.2

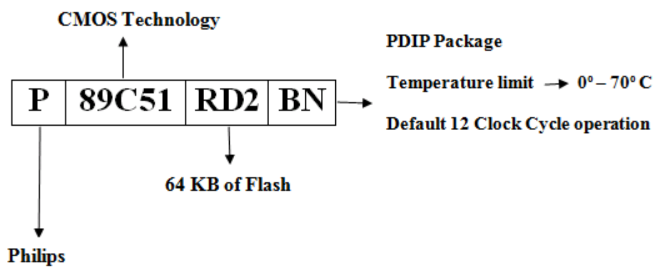

CMOS Technology
PDIP Package
Temperature limit → 0° – 70° C
Default 12 Clock Cycle operation
P
89C51
RD2
BN
64 KB of Flash
Philips

Fig.2 89C51 part number expansion

The P89C51RD2BN contains a non-volatile 64KB Flash program memory that is both parallel programmable and serial In-System and In-Application Programmable. In-System Programming (ISP) allows the user to download new code while the microcontroller sits in the application. In-Application Programming (IAP) means that the microcontroller fetches new program code and reprograms itself while in the system. This allows for remote programming over a modem link. A default serial loader (boot loader) program in ROM allows serial In-System programming of the Flash memory via the UART without the need for a loader in the Flash code. For In-Application Programming, the user program erases and reprograms the Flash memory by use of standard routines contained in ROM.

The device supports 6-clock/12-clock mode selection by programming a Flash bit using parallel programming or In-System Programming. In addition, an SFR bit (X2) in the clock control register (CKCON) also selects between 6-clock/12-clock mode. Additionally, when in 6-clock mode, peripherals may use either 6 clocks per machine cycle or 12 clocks per machine cycle. This choice is available individually for each peripheral and is selected by bits in the CKCON register. This device is a Single-Chip 8-Bit Microcontroller manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The instruction set is 100% compatible with the 80C51 instruction set. The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. The added features of the P89C51RD2BN make it a powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.

5. FEATURES OF 89C51

- 80C51 Central Processing Unit
- On-chip Flash Program Memory with In-System Programming (ISP) and In-Application Programming (IAP) capability
- Boot ROM contains low level Flash programming routines for downloading via the UART
- Supports 6-clock/12-clock mode via parallel programmer (default clock mode after Chip Erase is 12-clock)
- 6-clock/12-clock mode Flash bit erasable and programmable via ISP
- 6-clock/12-clock mode programmable “on-the-fly” by SFR bit
- Peripherals (PCA, timers, UART) may use either 6-clock or 12-clock mode while the CPU is in 6-clock mode
- Speed up to 20 MHz with 6-clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clocks per machine cycle
- Fully static operation
- RAM expandable externally to 64 kilo bytes
- Seven interrupt sources
- Four 8-bit I/O ports
- Full-duplex enhanced UART
    - ➢ Framing error detection
    - ➢ Automatic address recognition
- Programmable clock-out pin
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programmable Counter Array (PCA)
    - ➢ PWM
    - ➢ Capture/compare

6. PROJECT DESCRIPTION

CAN Based vehicle automation, is implemented using two nodes. Each node contains embedded microcontroller, CAN controller, CAN transceiver and some electronic control devices. Since the microcontroller has the ability to interface with number of electronic control devices we can connect many devices with the embedded microcontroller. Here Electronic control units like Tyre Pressure, Engine temperature and Vibration, etc., are interfaced with the embedded microcontroller. The two nodes are interconnected using CAN cable. CAN uses only short message, so the maximum utility of the load is 64 bits.

To implement this project 8-bit Microcontroller is used P89C51 and firmware for this project is developed using Embedded ‘C’. The Receiver side uses a ARM LPC2148 micro controller. The LPC2148 is a 32-bit RISC micro controller based on ARM7TDMI architecture.

Temperature, pressure, vibration,ultrasonic sensors are interfaced with Embedded microcontroller.The sensors are the transducers that converts non-electrical quantity into electrical quantity. The output of the sensing device is qualified by using Signal Conditioning circuit. This analog signal is converted into digital using Analog to digital Converter (ADC).The proposed implementation of the project is shown in Fig.3.

The embedded microcontrollers in both the nodes are programmed to collect the information from the electronic control units or to control the units as per the commands from the other node. The information from the temperature and pressure sensor interfaced with node1 and is collected by the microcontroller and it is transferred to the CAN controller1 through Serial Peripheral Interface (SPI). The information from the vibration and ultrasonic sensor interfaced with node1 and is collected by the microcontroller and it is transferred to the CAN controller2 through Serial Peripheral Interface (SPI). The CAN Controller converts the data from the format of SPI to the format of CAN. The CAN controller is the stand-alone device performs the functions of CAN protocol. The CAN transceiver splits the data from the CAN controller to CAN_LOW and CAN_HIGH. These transceivers are specially designed for high-speed differential data transmission between the CAN controllers and the physical differential bus lines.

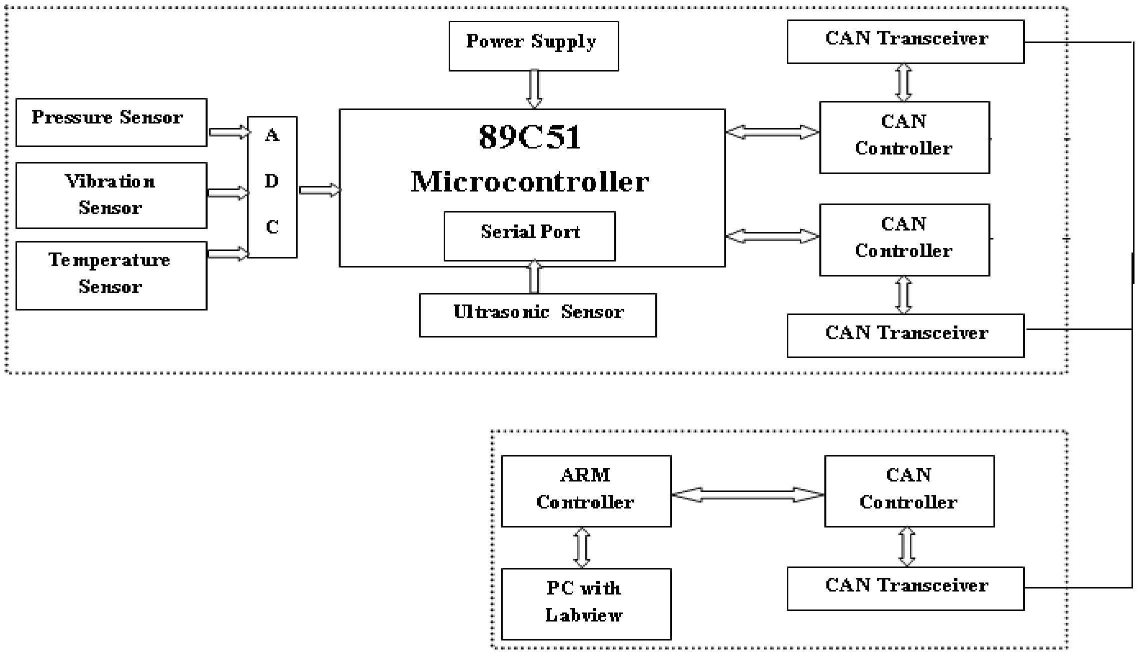


Fig.3 Block Diagram

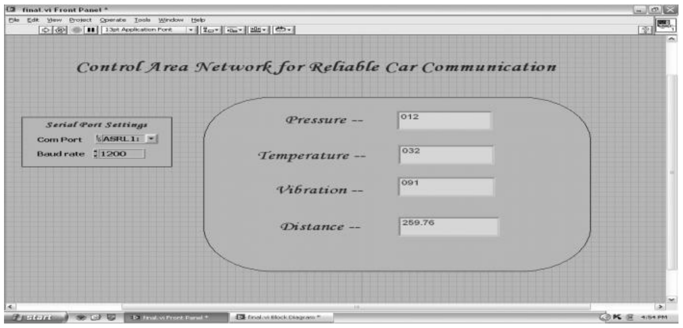
The CAN transceiver connected in the node2 receives the data from both CAN controllers using CAN protocol. The application software is developed with Lab view, which is a Visual Engineering Environment tool used for Virtual instrumentation. The readings from the sensors are displayed in the PC. The error confinement and the error detection feature make CAN more reliable in noise critical environments. Since CAN is a multi-master network we can use this for real-time applications in factory environment.

7. ABOUT LABVIEW

LabVIEW, short for laboratory Virtual Instrument Engineering Workbench, is a programming environment in which we create programs using a graphical notation. It offers more flexibility than standard laboratory instruments because it is software based. Because of LabVIEW's graphical nature, it is inherently a data presentation package. Output appears in any form Charts, Graphs and user-defined graphics comprise just a fraction of available output options.

LabVIEW program consists of one or more Virtual Instrumentations (VIs). Virtual Instruments are called such because their appearance and operation often imitate actual physical instruments.

8. RESULT



Control Area Network for Reliable Car Communication

Serial Port Settings
Com Port ASRL1::
Baud rate 1200

Pressure -- 012
Temperature -- 032
Vibration -- 091
Distance -- 259.76

Fig.4 Labview Simulation Results

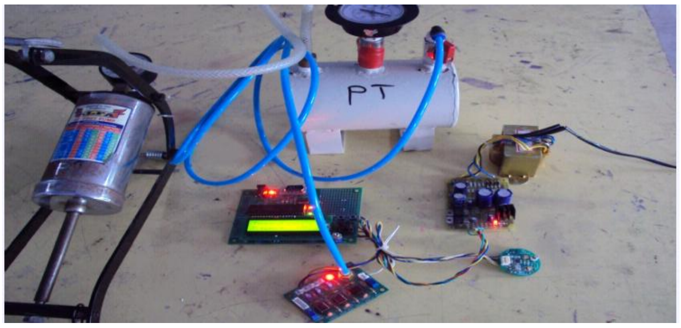


Fig.5 Hardware Module

9. CONCLUSION

The considerable system design challenges have been discussed, reasons for the choice of system architecture have been justified and aspects of the detailed design have been covered.

In future we are planned to transmit the unit cost and the last date to the consumer’s mobile phone by interfacing the GSM module with this kit.

10. REFERENCES

[1] Huaqun Guo, Jun Jie Ang and Yongdong Wu, “Extracting Controller Area Network Data for Reliable Car Communications”,I Proc.IEEE, 2009,pp.1027-1032.

[2] Jing Xu, Tao Lu, Lingling Gao, “Design and Application of In-Vehicle Terminal for Car Network System Based on ARM9”, IEEE international workshop on Education Technology and Training, 2008, pp.324-327.

[3] Lamia Chaari, Nouri Masmoudi,Lotfi Kamoun, “Electronic control in Electric Vehicle Based on CAN Network”, IEEE Trans. 2002.

[4] Jadsonlee da Silva Sa,Jaidilson Jo da Silva, Miguel Goncalves Wanzeller and Jose Sergio da Rocha Neto, “Monitoring of Temperature Using Smart Sensors Based on CAN Architecture”, IEEE Proceedings of the 15th International conference on Electronics, Communication and Computers, 2005.

[5] Zhou Yongqin, Wang Xudong, Zhou Meilan, “The Research and Realization for Passenger Car CAN Bus”, IEEE Trans, 2006, pp.244-247.

[6] LI Gangyan, Xu Jun, “An information Acquisition Method of City Bus Integrated Control Network”, IEEE Computer Society, 2008, pp.722-725.