INTELLIGENT MODULE FOR DATA EXCHANGE USING CAN INTERFACE

Ljudmila Roumenova Taneva

Technical University – Sofia,
Technological School „Electronic Systems“
Sofia, Mladost 1,
T.02 8751094; E. lusy_t@yahoo.com.

Abstract

The paper describes a CAN-bus based control system with changeable message priority. The system consists of microcontroller MSP430F149 (TI), sensors, CAN controller and several subsystems typical for the automotive industry and the industrial automation.

The CAN bus system is a multi-master system and the arbitration of the access to the bus is based on the transmitting devices' unique priority codes which are usually fixed. The priority code of the transmitting device determines the priority of the message transmitted by it.

In the proposed system the priority of the messages could be changed dynamically according the process stage and fixed priorities of the other nodes connected to the network.

The intelligent CAN controller increases the quality of the controlling complex processes and reduces the used hardware.

1. INTRODUCTION

The Controller-Area-Network (CAN) bus provides high level of noise immunity and fault tolerance [1]. For this reason it dominates the automotive industry and is widely used in industrial automation, military and many other areas characterized with harsh environment [2].

The CAN bus is a balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The bus is controlled by CAN controller and the devices interface with it via transmitters. The CAN controller also queues the incoming and outgoing messages [3]. These messages are usually broadcasted, i.e. sent to all bus participants simultaneously.

The transmission rate depends on the bus length. For buses shorter than 40 meters, the transmission rate is up to 1Mbits/s. Due to the practical limitation of the transceivers, in a single system could be linked up to 110 nodes (fig.1).

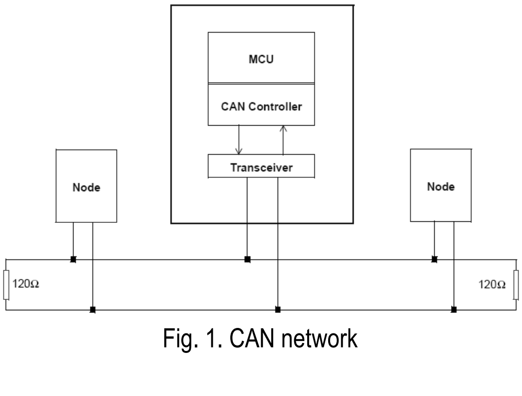

Fig. 1. CAN network

The messages with higher priority are always transmitted, but sometimes at the expense of the lower priority ones [4]. By loading the bus heavily it is possible for higher priority messages to interfere with the timely delivery of the low priority ones or even "block" them, not allowing transmission at all.

The developed intelligent CAN controller makes program changes of the message priority. The algorithm is preliminary defined and the microcontroller executes it. CAN protocol does not use physical addresses and sends messages with identifier that can be recognized by the various nodes. The identifier is used for message filtering and for determining message priority. The recipient node address in both transmit and receive modes is placed in the identifier as well.

2. PRINCIPLES OF DATA EXCHANGE

CAN is based on the "broadcast communication mechanism", which is based on a message-oriented transmission protocol. It defines message contents rather than stations and station addresses [2]. Every message has a message identifier, which is unique within the whole network since it defines content and also the priority of the message. The level of importance is used when several stations compete for bus access (bus arbitration) [1]. As a result of the content-oriented addressing scheme a high degree of system and configuration flexibility is achieved. It is easy to add stations to an existing CAN network without making any hardware or software modifications to the present stations as long as the new stations are purely receivers. This allows for a modular concept and also permits the reception of multiple data and the synchronization of distributed processes. Also, data transmission is not based on the availability of specific types of stations, which allows simple servicing and upgrading of the network.

In real-time processing the urgency of messages to be exchanged over the network can differ greatly: a rapidly changing dimension, e.g. engine load has to be transmitted more frequently and therefore with less delays than other dimensions, e.g. engine temperature. The priority, at which a message is transmitted compared to another less urgent message, is specified by the identifier of each message. The priorities are set during system design in the form of corresponding binary values and cannot be changed dynamically. The identifier with the lowest binary number has the highest priority.

Bus access conflicts are resolved by bit-wise arbitration of the identifiers involved by each station observing the bus level bit for bit. This happens in accordance with the wired-and-mechanism, by which the dominant state overwrites the recessive state. All those stations (nodes) with recessive transmission and dominant observation lose the competition for bus access. All those "losers" automatically become receivers of the message with the highest priority and do not re-attempt transmission until the bus is available again.

Transmission requests are handled in order of their importance for the system as a whole. This proves especially advantageous when system is overloaded. Since bus access is prioritized on the basis of the messages, it is possible to guarantee low individual latency times in real-time systems.

2.1. Message frame formats

The CAN protocol supports two message frame formats, the only essential difference being the length of the identifier (fig.2). The "CAN base frame" supports a length of 11 bits for the identifier (formerly known as CAN 1.0 A), and the "CAN extended frame" supports a length of 29 bits for the identifier (formerly known as CAN 2.0 B).

Fig. 2. Message frame format

2.1.1. CAN base frame format

A CAN base frame message begins with the start bit called "Start Of Frame (SOF)", this is followed by the "Arbitration field" which consist of the identifier and the "Remote Transmission Request (RTR)" bit used to distinguish between the data frame and the data request frame called remote frame. [2] The following "Control field" contains the "IDentifier Extension (IDE)" bit to distinguish between the CAN base 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 bytes. The "Data field" that follows is able to hold up to 8 data byte. The integrity of the frame is guaranteed by the following "Cyclic Redundant Check (CRC)" sum. The "ACKnowledge (ACK) field" compromises the ACK slot and the ACK delimiter. The bit in the ACK slot is sent as a recessive bit and is overwritten as a dominant bit by those receivers, which have at this time received the data correctly. Correct messages are acknowledged by the receivers regardless of the result of the acceptance test. The end of the message is indicated by "End Of Frame (EOF)". The "Intermission Frame Space (IFS)" is the minimum number of bits separating consecutive messages. Unless another station starts transmitting, the bus remains idle after this.

2.1.2. CAN extended frame format

The difference between an extended frame format message and a base frame format message is the length of the identifier used. The 29-bit identifier is made up of the 11-bit identifier ("base identifier") and an 18-bit extension ("identifier extension"). The distinction between CAN base frame format and CAN extended frame format is made by using the IDE bit, which is transmitted as dominant in case of an 11-bit frame, and transmitted as recessive in case of a 29-bit frame. As the two formats have to co-exist on one bus, it is set which message has higher priority on the bus in the case of bus access collision with different formats and the same identifier / base identifier: The 11-bit message always has priority over the 29-bit message. The extended format has some trade-offs: The bus latency time is longer (in minimum 20 bit-times), messages in extended format require more bandwidth (about 20

%), and the error detection performance is lower (because the chosen polynomial for the 15-bit CRC is optimized for frame length up to 112 bits). CAN controllers, which support extended frame format messages are also able to send and receive messages in CAN base frame format. CAN controllers that just cover the base frame format do not interpret extended frames correctly. However there are CAN controllers, which only support the base frame format, recognize extended messages and ignore them.

3. MODULE PRINCIPLE OF OPERATION

The developed intelligent CAN controller uses 12-channel 12-bit ADC to receive data from various sensors. The used microcontroller (MSP430F149) [1] receives information from the sensors, calculates data and transmits massage via CAN bus to the receiving nodes (fig.3). As transmitter is used CAN transceiver 82C250, Philips [3]. It operates in differential mode with inverted signals CAN_H and CAN_L.

For example, if the system consists of five nodes, the priority of each node is fixed from 0 to 8 and 0 is the highest priority.

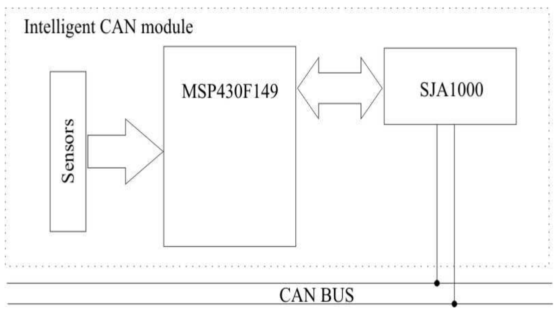

Fig. 3. Intelligent CAN controller

In the complex processes the importance of the parameters are different in the different stages of the process. To escape the loss of information, the sample rate and the message priority have to be changed depending on the algorithm of the program. At least two unique identifiers will have to be reserved for the intelligent CAN module (in this case four identifiers: 0, 2, 5, 8). The priority of the module will be changed according the algorithm when conditions for the parameters values are in the preliminary defined limits. If one of the conditions is available, the intelligent CAN controller changes the level of its own priority, righting 0 or 2, 5, 8 in the area of identifier.

The basic algorithm which the microcontroller executes is the follow: The microcontroller permanently receives the signals from the sensors, calculates data and store the result in the variable. When there is a request from the CAN Host node, the microcontroller sends the last actualization of this variable.

For example, if the measured signal is temperature, in this way the Host node receives the data in suitable type. It is not necessary to make additional calculations.

The algorithm calculates data in the measured unit, which is used by the Host node. If the Host node sends a command for continuous transmitting, transmits the value on every measurement. The program informs the Host node when the data is equal to the value, defined before,

4. CONCLUSION

The paper describes a CAN-bus based control system with non-fixed message priority. This intelligent CAN controller increases the quality, reliability and flexibility of the controlling complex processes and reduces the used hardware.

References

[1] Application Reports, TI, "MSP430 Family Mixed-Signal Microcontroller", 2003.

[2] R. Bosch, "CAN Specification", Robert Bosch GmbH, Stuttgard, 1991.

[3] Data Sheet, "PCA82C250 CAN controller interface", Philips, 2000.

[4] Docheva L., Al. Bekiarski ,I. Dochev, Analysis of Analog Neural Network Model With CMOS Multipliers, Proceedings of Czech and Slovak Technical Universities and URSI Committees, Journal Radioengineering, vol.16, №3, pp. 103-107, 2007.