Amber Arizpe

Applications of Controller Area Network Protocol

Group 6
Remote Monitoring System
Course: ECE 442 Senior Design
Date: 25 JAN, 2002
Web Page: **http://www.orst.edu/~johnsom8/ece44x/whitepaper.htm**

Applications of Controller Area Network Protocol

Robert Bosch conceived the Controller Area Network Protocol in a German company for the car industry. As cars became more complex in ways of controlling parts of the car through automation, the wiring for these complex systems became extensive and cumbersome. It was the intent of the protocol to provide a low-cost electronic alternatives through combining the multitude of wires into a few wires and a bus system. This paper shall explore how the CAN protocol achieved this and how this protocol is being adapted for other fields, other than automotive.

The need for this protocol arose from the increasing complexity of the automobile. Car manufactures were being asked by the customer for the car to be more convenient, comfortable and still be safe. The manufactures also had to appease government safety regulations in order to sell their cars. These interesting problems were resolved with electronic control systems. Such systems include anti-locking brakes, environment control, engine management systems, remote door locking, some devices were being used to reduce pollution and fuel consumption. With the increase of an electronic presence in the automobile, it became important to come up with a centralized system to tie all these different electronic control systems. Unfortunately, it also meant that wiring of all the control systems increased as well. And if a sensor's information needed to be duplicated to different controllers in the car, wiring for such connections went up exponentially. This introduced delay to the system and space became even scarcer. There had to be an electronic solution to keep communication real-time, wiring was to be reduced to an acceptable level and there could be no or very little error in signals.

The CAN protocol was conceived and implemented in 1986 as a solution to relay information from sensors to a centralized controller or controllers. It was later standardized internationally under ISO 11898(CAN2.0A) and ISO 11519(CAN2.0B). The elegance of the solution was that it reduced the number of wires needed, the protocol could reach up to speeds of 1 Mbit/s, and it could send and receive commands from a controller.

The protocol is structured for the needs of the car. Imagine this situation, when starting up the car, it runs through some basic diagnostic checks of the car system. The central computer, which controls all the electronic systems, sends out a message to all the systems for a “GO” signal. This message is given a unique ID number. Each system would know that it only listens to a specific list of ID numbers. In turn, the systems would send either a “GO” or “NOGO” message and the central computer would read it.

Since this protocol is message based, one message need only be sent to reach all the necessary systems. Each system can choose to acknowledge or ignore a message based on a list of authorized numbers. With this kind of system, it decreases the number of wires to about two wires. And like all systems the limit on number of unique messages depends on which CAN is in use, if it is Basic CAN (CAN2.0A), it will allow up to 2^{11} possible messages, while extended CAN (CAN2.0B) allows 2^{29}.

Another kind of system that could use such a message system perhaps would be a packaging system. CAN would send messages to the controller as an alarm or pass on messages to tell the next stage of the packaging to start. Another example might be a photocopier or basic medical systems or even some complex toy, such as robotics. There are many products today that use the message system of CAN to accomplish various tasks at the same time, in sequence, or as a warning system.

With the wiring issue reduced to an acceptable and optimum level, the question of error handling comes up. If there are several messages being sent at once, how is priority given and how are message failures taken care of or averted? CAN solve these problems by following a few rules.

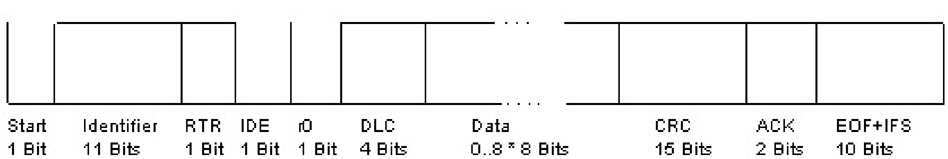
The lower the ID number, the higher the priority. And if there is an error anywhere in the message, an error frame will be generated which warns all the other systems to ignore this particular message. The system that sent the message makes note of the error frame and attempts to resend the message.

In the case of the car, it would not be good practice for the gas gauge to mis-send a warning to the central computer about running on empty. According to the gas gauge it sent several warnings but the message kept getting errors and therefore ignored. The CAN protocol has five separate methods of checking for errors: three at the message level and two at the bit level of the code. If any one of these methods finds the slightest error, the message is flagged as an error and is later resent. Such a redundant form or error checking is very attractive to systems that require accuracy and little time loss due to errors.

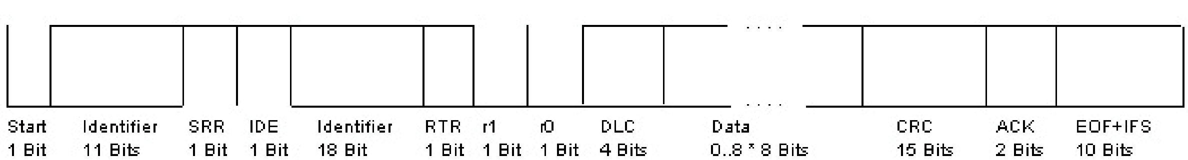
Systems such as navigation systems, elevator control systems, and even all the other systems mentioned earlier. All these kinds of systems require some sort of error handling mechanism. If the message is damaged en route to other nodes, pertinent information or the timing of a system can be thrown off. Also, such error handling may be perfect for a communication system that needs to only send small bits of information.

When the CAN messages are broken down to its component level, the data component is

Dataframe CAN 2.0 A (11 Bit Identifier)



Dataframe CAN 2.0 B (29 Bit Identifier)



actually quite small in relation to other components. Each message allows for 8 bytes of data to be sent, while this seems small it actually is quite enough to send data to and from the controller to its nodes. Also, having small messages allows faster transmission, which allows more messages to be sent over a period of time.

In the case of the car, messages being sent to the environment controls need only hold information to either turn on/off the heat/air-conditioner, open certain air vents or turn on the defroster. These aren't complicated messages. And it is interesting to note that when complicated systems are broken down to smaller systems, the complexity of the message's content to control these smaller ones decrease as well.

Systems that could benefit from this small, fast and accurate protocol could be medicinal in nature, such as an X-ray machine or patient monitoring devices. Such possibilities could extend to tools in space, environmental controls and robotics, or to security systems in small businesses. The Controller Area Network Protocol has the potential to be used in so many different fields for systems that require minimal wiring, excellent error handling and fast response times.

<u>References</u>

Bagschik, Peter. *An Introduction to CAN*. I + ME. 22 Nov. 2001 http://www.ime-actia.com/can_intro.htm

Bosch, CAN specification, Version 2.0, 1991, Robert Bosch GmbH

Etschberger, Konrad. *Controller Area Network, Basics, Protocols, Chips and Applications.* Weingarten, Germany University of Applied Sciences. IXXAT Automation GmbH. 2001.

Schofield, M. J. Controller Area Network(CANbus). 12 Jan. 2002 http://www.mjschofield.com/