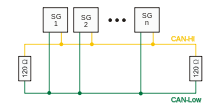
**Overview of CAN bus in Embedded Systems**

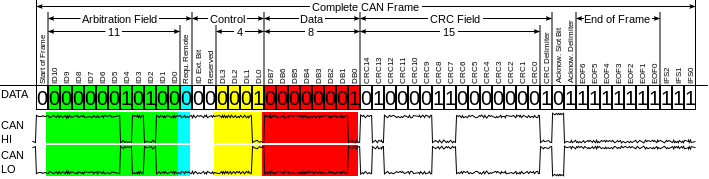
The CAN(Controller Area Network) bus is a multi-master, multi-cast serial bus capable of up to 1 Mbps transfer speeds at a bus length of 40 meters. The network part of CAN means that other nodes can exist on the bus as in the following topology:



where SG1,SG2 .. SGn represent the nodes or CAN-enabled devices on the bus. The 120 ohm resistors represent terminators to the bus and CAN-HI and CAN-LO represent the different lines of the bus representing different voltage levels. This design helps reduce electrical interference. This bus topology is not unlike the bus networking topology used in TCP/IP networks [1]. In fact the CAN bus implements the Physical Layer and Data Link Layer of the OSI networking model [2]. However, unlike computer networks, one can not route the data through the CAN bus using switches or routers. It is not a point-to-point communication. Data is multi-cast or broadcast to all nodes on the bus. There is no unique address associated with a node on the bus. Multi-master means that any node at any given time can gain control of the bus and transmit data to other nodes.

What are the advantages of a CAN implementation? CAN was founded in 1983 by Robert Bosch corporation for use in automotive applications. The first CAN controller chips were produced for market by Intel and Phillips in 1987. In 1993 the International Organization for Standardization released the CAN standard ISO 11898. The Bosch CAN 2.0 specification does dictate the physical layer of the bus- only the data link layer. ISO 11898-2 and ISO 11898-3 cover the different physical layers used for high-speed and low-speed CAN. Therefore, CAN physical implementations vary from 1 wire to 4 wire implementations with the extra 2 wires used for power and ground, for example. As automobiles advanced in efficiency, convenience, and pollution reduction, ECUs or microcontrollers used implement these systems became more complicated. Different areas of vehicle such as the engine, brakes, turn signals, air bags, and air conditioning needed more fine tuning. With the additional controls came additional wiring and the embedded system became overly complicated and a single point of failure. With CAN, the wiring can be reduced significantly and 1 ECU can be divided into multiple ECUS each dedicated to a particular system of the vehicle. CAN allows multiple microcontrollers to communicate with each other and yet remain distributed. For example, if the brake system malfunctions this does not have to affect the air bag system. They are loosely coupled, allowing the heterogeneous systems to communicate with each other but remain physically independent. ISO 11898-2 and ISO 11898-3 define a physical wiring standard. Today’s vehicles may contain an average of 30 microcontrollers. Without the CAN specification, the implementation and coordination of safety-critical systems would not be possible. Other advantages of CAN include a built-in robust error-detection system, node fault protection, and a non-destructive arbitration system by CSMA/CD.[3] Today, CAN has moved to other applications besides automobiles such as agriculture equipment, industrial controls, entertainment, and robotic systems.

CSMA/CD or Carrier Sense Multiple Access / Collison Detection mechanism allows an arbitration and node priority to be implemented. A typical CAN frame in the data link layer is shown in the diagram below:



The basic fundamentals of the frame are the composition of the signal which represents a logic ‘1’ or a logic ‘0’. In terms of arbitration ‘1’ represents a recessive level and ‘0’ represents a dominant level. The default state of the signal is recessive level. In the physical implementation in the figure a dominant value is represented by the CAN HI line going high, and the CAN LO line going low. The opposite or recessive level is represented by CAN HI going low, and CAN LO going high. When a frame is sent out by CAN control module on a microcontroller, it first detects other frames on the other nodes. The frames are compared bit by bit starting at the arbitration field. The first dominant bit or ‘1’ bit in the frame is the frame and therefore the node that wins the bus and can send. The other frames and their respective nodes must back off and try to re-send as specified by CAN baud rate. In this way, a priority mechanism can be implemented whereby the lower the value of the arbitration field the higher the priority of the frame. If two frames have the same priority and thus the same arbitration field then a Bit error will occur. Thus, the Arbitration field must have unique value compared to other frames read on the bus. If the frame is successfully sent without errors the microcontroller will see a dominant bit in the Acknow. Slot in the receiving frame.

Fault modes make up three different error types- active errors, passive errors, and bus off. Internal error counters in the CAN module determine the error type. Both a transmit error counter and receive error counter exist. The counters can be used to determine which state the node is in. For example if the both the transmit error count and receiver error count are above a certain threshold then the node enters a passive error state. Active error mode means that a node can send all frame types including error frames. Passive error mode can send frame types except for error frames. Bus off state means the node cannot send anything. Error frames consist of two different fields. The first field is a combination of ERROR FLAGS (6-12 dominant/recessive bits) contributed from different nodes. The following second field is the error delimiter (8 recessive bits).[4] The CRC is calculated over the bits of the frame before it is sent by the node. The receiver recalculates the CRC of received frame and compares it to the CRC field. If it differs the frame is discarded and the receiving node sends a error frame. Of course all nodes on the bus receive the error frame just as they would a normal frame as shown in picture above. Each receiver can decide to ignore or process the frame. The Arbitration field can also act as an ID of the node whereby other nodes can process a message based on it’s ID or sender node. The only limit on the amount of nodes that exist on the bus is determined by electrical constraints or amount of load on the bus.

The CAN standard does not define security mechanisms such as authentication. These can be provided by application layer software such as OpenCAN which operates on the higher levels of the OSI model. It is conceivable that a message on the bus could be intercepted by a rogue node and retransmitted causing harm to the system.

In conclusion, CAN knowledge in the field can be a valuable asset. It has grown to many other uses since it’s start in automobiles. CAN bus interaction with embedded systems is an important development and has lower overhead than traditional Ethernet communication, but a more robust and concise implementation than RS-232 for example regarding communication between multiple embedded systems.

Author: Kerry Murphy

**REFERENCES**

*1****.***[*http://en.wikipedia.org/wiki/Network\_topology#Bus*](http://en.wikipedia.org/wiki/Network_topology#Bus)

*2.* [*http://en.wikipedia.org/wiki/OSI\_model*](http://en.wikipedia.org/wiki/OSI_model)

*3.* [*http://www.atmel.com/Images/doc32152.pdf*](http://www.atmel.com/Images/doc32152.pdf)

*4.* [*http://en.wikipedia.org/wiki/CAN\_bus#ACK\_slot*](http://en.wikipedia.org/wiki/CAN_bus#ACK_slot)