

Vehicle-Bus Interface with GMLAN for Data Collection

David C. Pheanis
Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287-5406
David.Pheanis@asu.edu

Jeffrey A. Tenney
Western Microsystems
938 E. Ranch Road
Tempe, Arizona 85284-3225
Jeffrey.Tenney@gm.com

Abstract

Within a few years, nearly every vehicle that General Motors (GM) manufactures will have a microcontroller network known as GMLAN. A few GM vehicles already have the GMLAN network, and more vehicles with GMLAN will appear each year. GMLAN uses the Bosch-designed CAN (Controller Area Network) protocol, and it links all of a vehicle's various processors or *nodes* to form an in-vehicle data network. CAN uses a Non-Return-to-Zero protocol, NRZ-5, with bit stuffing, and the network implements Carrier Sense Multiple Access with Collision Resolution (CSMA/CR). GMLAN actually includes two linked CAN buses: a high-speed dual-wire CAN for high-speed nodes and a lower-speed single-wire CAN for other nodes.

The primary purpose of the GMLAN network is to improve reliability while simultaneously lowering cost by reducing the numbers of wires, connections, and special-purpose circuits in a vehicle. The GMLAN network also promotes synergism among the vehicle nodes to provide vehicle features that would simply not be feasible without the network.

We have developed a Vehicle-Bus Interface (VBI) that allows us to collect test data directly from a vehicle's nodes via the GMLAN network with a GM Standard Thermal Instrument (STI). The VBI facilitates vehicle testing by greatly reducing the need for external sensors and transducers.

1 Introduction

Microcontrollers are changing the automotive industry. Microcontrollers now control vehicle components such as the engine, the antilock braking system, the transmission, the instrument panel, and the entertainment center, to name just a few. A microcontroller does more than just make its own component work

correctly — it also participates in one of possibly several networks in the vehicle. Such a network, usually called a *vehicle bus*, allows the sharing of information among components and also reduces the numbers of wires and sensors required in the vehicle.

Until just recently, automotive testing by vehicle manufacturers has required the expensive and time-consuming installation of myriad sensors and transducers to collect data from the vehicle. In the late 1990's GM started using the Class-2 Bus [1] to network the nodes on GM vehicles, and engineers at GM's Desert Proving Ground developed a Vehicle-Bus Interface (VBI) that allowed them to collect test data directly from the vehicle nodes via the low-speed Class-2 Bus, which runs at approximately 10.4 KBS or 10,400 bits per second. Now we can interact with the much faster GMLAN network at 500 KBS to collect a wealth of data from sensors and controllers that already exist on the vehicle. This paper introduces a Vehicle-Bus Interface (VBI) that allows GM's Standard Thermal Instrument (STI) to collect data by interacting with the nodes on the GMLAN network that is already on some GM vehicles and will soon be on all GM vehicles.

2 CAN Protocol

Bosch pioneered the CAN serial data-link protocol in the early 1980's and subsequently licensed a number of semiconductor vendors to develop and produce controller chips for the CAN protocol. CAN applications today include not just automotive networks but also medical systems, home appliances, trains, ships, airplanes, factories, and even vending machines.

CAN specifies only the data-link layer of the network protocol, and it leaves all other layers open. For example, CAN does not specify any of the layers above the data-link layer, nor does it specify the physical layer,

which is below the data-link layer. CAN does, however, specify that the physical layer must support an *active* (i.e., dominant) state for a zero bit and a *passive* (i.e., recessive) state for a one bit. Many groups have developed differing physical layers that support CAN. GM, for example, has developed GMLAN, which defines two different physical layers along with the higher-level layers to provide two complete CAN implementations.

Note that CAN specifies the operation of every CAN controller chip, but the controller chip is separate from the transceiver chip, which interfaces with the physical layer and is not part of the CAN specification. Each different implementation of CAN is therefore independent with regard to its physical layer, and any CAN controller chip can support any physical CAN implementation if we merely pair it with the appropriate transceiver chip.

Since CAN does not specify the physical layer, designers are free to choose the bit rate as well as the physical signals and the bus media (e.g., single-wire or dual-wire copper, fiber, etc.). A designer might choose a slower bit rate to be able to have a longer physical bus length, or a designer might reduce the bus length to achieve a faster bit rate.

CAN uses a Non-Return-to-Zero protocol, NRZ-5, with bit stuffing. The idea behind bit stuffing is to provide a guaranteed edge on the signal so the receiver can resynchronize with the transmitter before minor clock discrepancies between the two nodes can cause a problem. With NRZ-5 the transmitter transmits at most five consecutive bits with the same value. After five bits with the same value (zero or one), the transmitter inserts a *stuff bit* with the opposite state. The next bit of the frame then follows the stuff bit, so the stuff bit is a wasted bit in the sense that it consumes bandwidth without transmitting information. However, the stuff bit guarantees an edge on the transmission signal after a maximum of five bit times, so the receiver can resynchronize with the sender on a one-to-zero edge after a maximum of only ten bit times. The stuff bit allows higher clock rates in spite of the fact that a frame can be as long as 108 bits.

The first eleven bits after the start bit in a standard CAN frame contain a frame identifier. The frame identifier and the following bit serve as an arbitration field. Frames with lower identifier values have higher priority because of the fact that zero bits are dominant. GMLAN (not CAN) specifies a particular identifier value for each type of GMLAN frame. For example, one particular identifier might indicate that the frame contains engine RPM and throttle position

while another identifier might indicate a frame with engine temperature.

Each node that transmits on a CAN bus also listens on the bus at the same time. If a node receives an active state when the node is transmitting a passive state in the arbitration field, that node loses arbitration and immediately ceases its transmission without affecting the dominant transmission that is on the bus. Eventually, one node wins arbitration as all others drop out, and the winning node's message is the only message that appears on the bus. This mechanism implements Carrier Sense Multiple Access with Collision Resolution (CSMA/CR). The appropriate message drives the bus, and we resolve collisions without damaging that message on the bus. Thus there is no need to retransmit the dominant message following a collision.

A standard CAN frame is a variable-length frame that contains from zero to as many as eight data bytes with eight bits each. A four-bit data-length field specifies the number of data bytes that follow in the frame. At a higher protocol level GMLAN (not CAN) defines the concept of a *message*. A message can be a single-frame message, or a message can span multiple frames with the frame identifier and various data fields indicating that the message is a multi-frame message. Each frame in a multi-frame message has the same frame identifier.

A node's CAN controller automatically appends a 15-bit CRC (Cyclical Redundancy Check) to each frame that it transmits, and the controller validates the CRC when receiving a message. Since each node receives all messages, including the messages that it transmits itself, a node's controller automatically retransmits any message that doesn't transmit properly the first time. Also, at least one recipient for each frame must send an acknowledgement bit after validating the CRC of the received frame. The transmitter verifies the remote assertion of the acknowledgement bit. This scheme makes the network reliable even in the electrically noisy environment of the vehicle.

3 GMLAN Implementation

In the late 1990's, GM started using the Class-2 Bus on its vehicles. The Class-2 Bus was an outgrowth of the proprietary Assembly-Line Data Link (ALDL) that GM used for activating and testing individual nodes of vehicles on the assembly line. The presence of the Class-2 Bus provided several important advantages after production, so the use of the bus quickly grew far beyond its original ALDL application.

GM soon decided to develop GMLAN with the CAN technology to replace the Class-2 Bus. Uses for the vehicle bus are quickly growing beyond the bandwidth of the Class-2 Bus, but CAN specifications allow a bus speed as high as one megabit per second, about a hundred times as fast as the Class-2 Bus. Also, CAN is potentially less expensive than the Class-2 Bus because many semiconductor vendors produce competing CAN chips whereas the Class-2 Bus uses custom-designed chips for its proprietary implementation.

Some vehicle nodes require high speed while others don't. For example, the Engine Control Module (ECM), the Powertrain Control Module (PCM), and the Antilock Braking System (ABS) all run at high speeds while modules such as seat controllers, window controllers, and door controllers can run at much lower speeds. Making the controllers for low-speed applications run at high speeds would not be cost effective, but a design with both a high-speed bus and a low-speed bus addresses this issue. A cheap 8-bit microcontroller can easily serve as a node on the low-speed bus while the high-speed bus requires faster, higher-cost processors.

GMLAN includes two CAN buses that connect to each other via a "gateway" node. One bus is a high-speed dual-wire CAN, and the other is a lower-speed single-wire CAN. Both buses use copper wire rather than fiber to reduce cost. GMLAN includes requirements to limit radiated emissions and to provide immunity to radiated emissions from other sources. The gateway node allows communication between the high-speed nodes and the low-speed nodes, so, for example, the low-speed radio controller can adjust the radio volume based on the engine RPM from the high-speed PCM.

The high-speed dual-wire CAN bus uses a differential signal for noise immunity at relatively high speeds. The bus has a maximum length (from the node at the head of the bus to the node at the tail of the bus) of 30 meters, and the bus can accommodate a maximum of 16 nodes. The bus operates at 500 KBS rather than the maximum CAN specification of one megabit per second since the lower speed allows a longer bus length, reduces radiated emissions, and is sufficient for the high-speed vehicle nodes.

The low-speed CAN bus operates over a single wire and uses the vehicle chassis as a ground reference. The use of a single wire reduces cost, and the low bus speed permits the use of inexpensive 8-bit microcontrollers as nodes. Because of the lower speed, 33.33 KBS, the bus has a maximum length of 60 meters and can accommodate as many as 32 nodes.

4 Bus Synergism

The GMLAN network connects all of the nodes in a vehicle with a maximum of only 90 meters of wire, so it provides a tremendous reduction in cost while greatly improving reliability. As a point of reference, the 1993 Corvette, a sophisticated vehicle without a vehicle bus, contains more than three *miles* of wire weighing over 200 pounds and has more than *fifty* wires running from the driver's door to the rest of the vehicle! Every wire is a wire that can break, and every connection is a connection that can fail. Wires add cost to the vehicle, and the sheer weight of the wire impacts fuel economy and performance.

Besides reducing cost and improving reliability, the vehicle bus also allows various vehicle nodes to work together to provide features that simply wouldn't be feasible without a bus. For example, the Powertrain Control Module (PCM) can broadcast the vehicle speed on the bus, and the node that controls the entertainment center can adjust the volume of the sound system automatically according to vehicle speed to compensate for increased or decreased road noise. The Antilock Braking System (ABS) can enhance vehicle control and safety by reacting to data values that the PCM broadcasts. The cell phone can broadcast its status, so an incoming or outgoing call on the cell phone can automatically mute the radio.

As another example, using the remote keyless entry system to unlock the doors can automatically activate the interior lights and adjust the power seat and entertainment center to the settings for the driver who unlocked the doors. The interior lights can go off automatically after the engine starts, and the vehicle can automatically lock its doors after reaching a speed of 15 MPH. Later, removing the key from the ignition after stopping the vehicle can automatically unlock the doors. Opening the door to leave the vehicle can then turn off the sound system, and the convenience lights can go off at a programmed interval after the driver and passengers leave the vehicle and close the doors.

We are only beginning to reap the advantages that are possible with the availability of communication among the various vehicle nodes. The vehicle bus will soon foster features that we can't even imagine today.

5 VBI Card for STI

General Motors uses the GM Standard Thermal Instrument (STI) to collect data from vehicles under

test, primarily for thermal testing. The STI is a real-time, embedded, distributed-processing system that collects, displays, and stores data to help engineers design better vehicles. The STI consists of a laptop computer known as the *controller*, a 16-slot chassis, and various types of plug-in cards.

The first plug-in card is the master communications (COMM) card, which always occupies the first slot in the chassis. The other plug-in cards are signal-conditioning cards that interface to transducers and other data sources, and the chassis can contain any suitable mix of as many as 15 signal-conditioning cards. Types of signal-conditioning cards include temperature/voltage cards, frequency cards, strain-gauge cards, and a vehicle-bus interface (VBI) card.

6 VBI Card

The VBI card is a new card that provides an interface to various vehicle buses, thus allowing the STI to obtain data from vehicle nodes and from sensors that are already part of the vehicle.

CARB, the California Air Resources Board, requires vehicle manufacturers to equip each vehicle with a port that allows an emissions tester to connect an instrument to the vehicle to obtain and validate the vehicle's emissions history and performance. GM meets this requirement by providing a connector to the various vehicle buses below the steering column. Serendipitously, we use this same connector to attach the STI's VBI card to the vehicle buses, so preparing to collect data from vehicle nodes is as simple as plugging a cable into an easily accessible connector. Compare this happy circumstance to traditional testing, which typically requires several days of preparation to install transducers and wiring on a vehicle, often damaging the vehicle and making it unsalable in the process.

The VBI card contains a Motorola MC68376 microcontroller along with Class-2 and CAN controllers. The card allows simultaneous communication with the Class-2 Bus, the low-speed single-wire CAN bus, and the high-speed dual-wire CAN bus. The VBI card acts as a test node on each vehicle bus. As a test node, the VBI card can request real-time data values from other nodes on the bus, and it can also command other nodes on the bus to perform various actions. Potentially, the VBI card could even flash new data tables or programs into vehicle nodes to change the characteristics of a vehicle for test purposes.

The Class-2 and CAN controllers occupy no memory space in the MC68376 microcontroller's memory map, and they require no external decode logic. They connect to the industry-standard Serial Peripheral Interface (SPI) under the MC68376's queued SPI controller.

The VBI card allows the STI user to record as many as 127 channels of information at sample rates as fast as 50 Hz. A *channel* is a single data value such as engine RPM, throttle position, coolant temperature, lateral acceleration, etc. We must limit the amount of information that a user can record from the vehicle buses because of the limited bus bandwidth. Remember that the vehicle itself uses some of the bus bandwidth (usually 20% or less) for its own operation, and we must take care to avoid affecting the very performance that we want to measure.

7 VBI Data-Collection Features

Collecting data from the GMLAN network presents a special challenge because of the nondeterministic nature of the bus. Bit stuffing makes the raw bus speed slightly data dependent at the bit level, but the bus is also nondeterministic in a much greater sense because a variable number of nodes with differing performance characteristics and unpredictable agendas share the bus.

The design of the VBI software carefully considers the fact that we can't depend on receiving data according to the delivery schedule that the test engineer requests. We must tolerate late responses and missing responses, and we must also accommodate bus traffic that isn't intended for us. We classify some problems as harmless problems and make them transparent to the STI user, and we report serious problems while taking special steps to recover from them automatically.

7.1 Eavesdropping

The VBI passively *eavesdrops* on vehicle nodes by watching bus traffic among the nodes for data values that apply to the current test. If any message on the bus contains any data value that we want, we make use of the data even if the VBI was not the intended recipient of the message. This feature allows us to collect test data without adding any traffic to the bus when desired data is already on the bus as a consequence of normal vehicle operation.

Active data collection, which allows the VBI to request specific values from specific vehicle nodes, is cur-

rently under development and will be available in the early part of 2003.

7.2 Data Extraction/Channel Mapping

Data channels for the VBI card don't map to specific physical resources on the card, or in the vehicle, for that matter. Therefore, our mapping of data values to data channels and our raw-data scaling must change dynamically and often dramatically with each new test configuration. Furthermore, when we receive data from a vehicle node, the node's message can contain data for just one channel or for as many as six channels. Additionally, we might need to request a data value each time we want it, or we might be able to program a vehicle node to report a particular data value at specified intervals.

In one test configuration, for example, we might have channel one reporting the coolant temperature in the lower radiator hose, and we might need to send a message to solicit that value each time we want another sample. The raw data might return in the third byte of the response message, and we might need to convert the raw data to degrees Celsius. In the next configuration, channel one might report the slip speed in the torque-converter clutch, and we might not need to send a message to solicit each value since we might be able to program the vehicle node to report the value at specified intervals. The raw data might return in the fourth and fifth bytes of the message, and we might need to divide by eight to convert the raw data to revolutions per minute.

The VBI software interprets the test configuration and handles each possible channel mapping, and the VBI software also extracts and converts each data value properly according to the configuration.

7.3 Persistent Acquisition

The VBI card keeps track of all nodes that we identify in the vehicle. We monitor each node's transmissions of State-of-Health (SOH) messages. We also detect a problem when a node has terminated diagnostic operation while we are using a collection scheme that requires the node to be in diagnostic mode. If we ever notice a loss of health or a premature termination of diagnostic mode, we immediately take corrective action. In many cases we can restore the node to the desired mode of operation automatically without user awareness. In other cases, we must report a loss of data due to the faulty node.

7.4 Post-Test Traffic Report

The VBI card stores all vehicle-bus traffic in a circular message buffer. This buffer contains the most recent traffic from the bus, storing as many messages as available memory allows. The buffer even contains bus traffic from before the beginning of the test if the VBI card was connected to the bus and was running.

Following a test, the user can request a traffic report. Each message in the report has an associated time stamp for reference. This traffic report is an invaluable tool for design engineers who are developing vehicle software and for test engineers who are troubleshooting problematic data. The traffic report allows us to investigate a vehicle's software and the behavior of nodes on the GMLAN network. We are concerned about the software in the vehicle because we rely on that software to generate accurate, properly formatted messages. The traffic report is also crucial to our further development of the VBI software because we can scrutinize our own behavior, both on the GMLAN network and in our real-time data processing.

8 Conclusion

The addition of the VBI card allows STI users to collect test data directly from a vehicle by simply connecting a cable to an easily accessible connector that is already on the vehicle. This convenience greatly reduces the need for the time-consuming and expensive installation of external transducers for data collection.

Additionally, the VBI card provides access to internal data values that simply aren't obtainable with external transducers. With the VBI card, the tester can determine not just what the vehicle is *doing* but also what the vehicle is *thinking*.

The VBI card has limitations, of course, since it can provide only data that is available through the vehicle buses, so the VBI card can't entirely replace the other STI signal-conditioning cards. However, the VBI card does supplement the other cards and makes the STI more versatile and more economical to use.

References

- [1] David C. Pheanis and Jeffrey A. Tenney, "Data Collection with Vehicle-Bus Interface," ISCA International Conference on Computers and Their Applications (CATA-2001), March 28-30, 2001, *Proceedings*, pp. 129-132.