# icc 1996

3rd international CAN Conference

in Paris (France)

Sponsored by

**Motorola Semiconductor**
**National Semiconductor**
**Philips Semiconductors**

Organized by

**CAN in Automation (CiA)**
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

Tony Moon

# Control in Automotive Body System Networks

Electronics Engineering Manager - New Technology
AB Automotive Electronics Ltd.
Forest Farm,
Cardiff CF4 7YS, U.K.

## Abstract

The multiplexing systems currently being implemented are essentially information sharing systems. In the body system in particular, it is possible to reduce the cost of implementing features by growing one of the micros into a 16 bit device and slimming down the others. The bus will carry commands from the 16 bit master device as well as information. Many of the tasks in the body system do not need to run often. The 16 bit device can make use of large (economical) flash memory for reprogrammability, but in order to benefit from reprogrammability, the whole bus needs to run under a communications manager that will guarantee the real time performance of the bus when program updates occur. This type of development will further divorce electronics hardware from software, and will require organisational changes in companies to implement it effectively.

## 1. Background

We are now seeing the first generation of multiplexed systems in volume production, and the multiplexing system has largely been grafted on to traditional control and circuit configurations. The layout and partitionning of the control modules owes much to the traditions and structures in the car companies, and the thesis of this paper is that it is time for a fresh look, armed now as we are with tools that unify all the vehicle's control sub-systems in a way that enables them to be considered as a whole.

Multiplexing has followed the normal practice of the motor industry in making progress in a series of small steps rather than technological leaps. The industry has had plenty of experience of leaps too far, and prefers to make changes in steps as this is perceived as offering lower risk. The failure of some of the more spectacular leaps is for quite unforeseen reasons. Multiplexing on current vehicles has therefore largely been grafted on to existing systems as a means of reducing harness congestion and sharing data from individual sensors or sources of information. Thus the car companies have retained their traditional departmental structure in which each department has ownership of a particular vehicle function (such as engine, transmission, suspension, brakes, heavac, etc.). As electronic control has grown on to vehicles, each department has grown its own particular controller (ECU), supplied by its chosen component manufacturer. It owns that ECU, and a data network can be grafted on with little difficulty. The data network merely serves to supply some of the input information necessary to carry out the control function of that sub-system, and provide a route by which diagnostics can be

Road Speed), in which case that ECU will be the supplier of that information to the rest of the vehicle. The traditional departmental organisation in the car companies remains intact.

## 2. Architecture

The availability of reliable high speed data networks such as CAN is enabling a different approach to be considered. If all ECUs put all the information that they see on to the CAN bus, then the bus becomes the single information source for the whole vehicle. It then becomes possible to see all the ECUs that are connected on the CAN network as a single sub-system in its own right, and the processing of inputs to create outputs can be done by any convenient processor on that bus. Some ECUs have high speed control functions to perform, but apart from these, it is possible to carry out all the remaining control functions in a single ECU. Clearly there are all manner of considerations that make such a possibility rather impractical, such as the desire to carry parts over from one model to the next, but the availability of a high speed reliable means to exchange data should lead to a fresh look at the architecture of the control systems. For guidance on the issues involved, see refs. 1 & 2.

This fresh look at architecture needs to be carried out against a background of the massively increasing complexity of software. Automatic climate control systems illustrate this well. One of the first ones with software control appeared in 1982 and had 1.8K bytes of code. A system to be launched soon has a lot more than 64K bytes in it. I cannot give an exact figure as we are still making changes. Our company launched one four years ago with 16K. If this growth is exponential, then the code size has doubled every 2.75 years. Of course, much of the growth has been in diagnostics and other non-functional operations, but it has greatly outstripped the growth in complexity of inputs and outputs. Meanwhile, the hardware is growing in complexity relatively slowly, partly due to the impact of CAN reducing the number of inputs required into to each individual module.

An analysis of inter-ECU traffic shows that it can fall into two categories:

- Information co-ordinating real time control aspects of the vehicle such as:

  road speed
  engine status
  transmission status
  steering status
  collision avoidance / intelligent cruise control information

- Information relating to needs of the occupants such as:

  journey information: navigation, traffic warnings & road tolling
  visibility: lighting, wipers & reversing aid
  comfort: climate control + seat and mirror & steering wheel positions
  security: access control
  entertainment & communications: radio, CD changer, integrated telephone
  vehicle status monitoring

Ref. 4 describes many systems in development that will improve driving and require information flows in addition to those above. The real time control traffic has to be delivered fast, and it generally consists of short messages broadcast in specific time slots. The information relating to the needs of the occupants is more varied and can consist of anything from short switch status messages to long text strings.

For this reason, a popular architecture on the more complex vehicles is to have two separate buses, one for high speed control which we shall call the "vehicle dynamics bus" and one for other functions which we shall call the "body system bus". CAN is almost universally used as the vehicle dynamics bus in Europe, but there is some diversity in the bus used for the body system. We end up with a figure of 8 bus structure. There is a gateway module that exchanges information between the two, and it is a convenient point to couple in a diagnostic port.

The vehicle dynamics ECUs generally have to work as real time controllers with a reaction time ranging from a few tens of microseconds to a few milliseconds. Given these deadlines, it makes little sense to try to combine all the control functions into a single module.

## 3. Body System

The body network is used for a quite diverse set of functions, many of which require quite

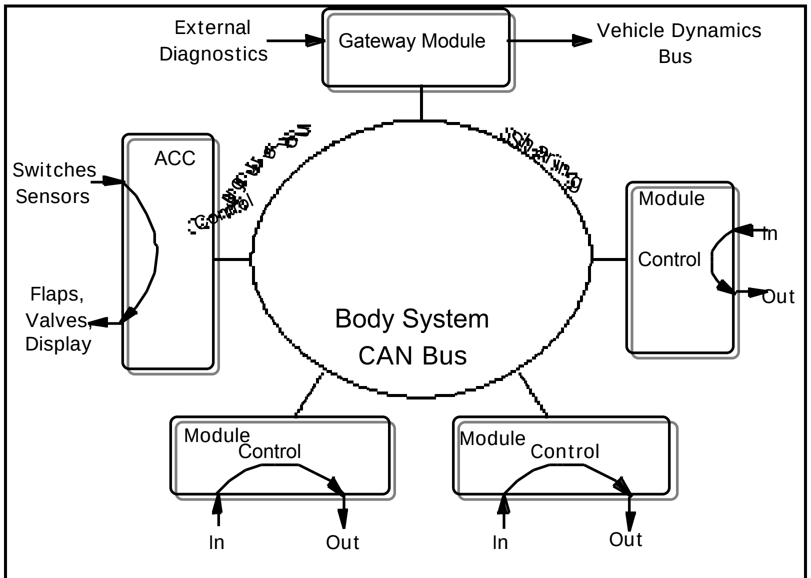

Fig. 1: Conventional Body System Bus - Information Sharing

complex processing of tasks that do not actually have to run very often, and the vehicle can occasionally tolerate delays in them. Good examples of this are automatic climate control, where changes in temperature take place so slowly there is little point in running the main control algorithm faster than once every 4 seconds. Access control algorithms need priority at specific times, but should not be run otherwise. One of the controllers on the bus usually drives a general-purpose secondary information display and possibly a speech generator. Both have to present information that is language dependent.

Some of the computing tasks within the body system area are better run on a powerful 16 bit micro than the traditional 8 bit devices in current use, because the 16 bit micros are much better at handling arithmetic with larger numbers. Many of the micros are idle or doing trivial tasks for most of the time (reversing aid, electronic phone book, road pricing, seat & mirror memory, e tc.).

Related to this growth in complexity of code is a trend towards the use of re-programmable memory, so that the code can be finalised much later in the vehicle development programme and corrections can even be made later via diagnostics without the need for the physical replacement of any parts. This currently poses problems. The range of micros available with flash memory on board is rather limited (none with CAN + flash according to information presented to this conference last year ref. 3), although moves are afoot in this direction. This means that currently flash memory has to be implemented in a separate chip external to the micro. This gives rise to two difficulties:

- an external bus is required to carry instructions and data between flash EPROM and micro. This high speed bus is a major source of radiated emissions.
- the flash memory required is usually at the bottom end of the sizes available so it has a high cost per bit.

Flash memory is mainly designed for use in the telecomms and computer industries where the demand is for rather bigger devices.

The central thesis of this paper is that one large 16 bit micro sitting on the body system bus is highly effective as a means of implementing many of the more complex control algorithms and offers other significant advantages. The best position for this "master module" is probably as the gateway module between the two buses as it has to make use of a lot of information from both.

The other micros on the body system bus can be slimmed right down and run code that basically makes them read inputs on to the CAN bus, then read outputs commands from the bus and implement them. There are some closed loop control functions that are better left in the local processors, such as the control of the temperature in the air ducts on the climate control system. Output commands can require some local implementation, such as generating the pulses to move a stepper motor to a specific position. The body system bus thus becomes one of both information sharing and control. The local processors can carry out relatively simple functions that are not so complex to validate, and generally masked processors can be used.

What if the bus fails? Then all the local processors resort to a mode of local control that provides a lower level of control, but keeps the system safe and legal. For example, the heavac will go into a manual control only mode. Clearly this design criterion is a key aspect of the

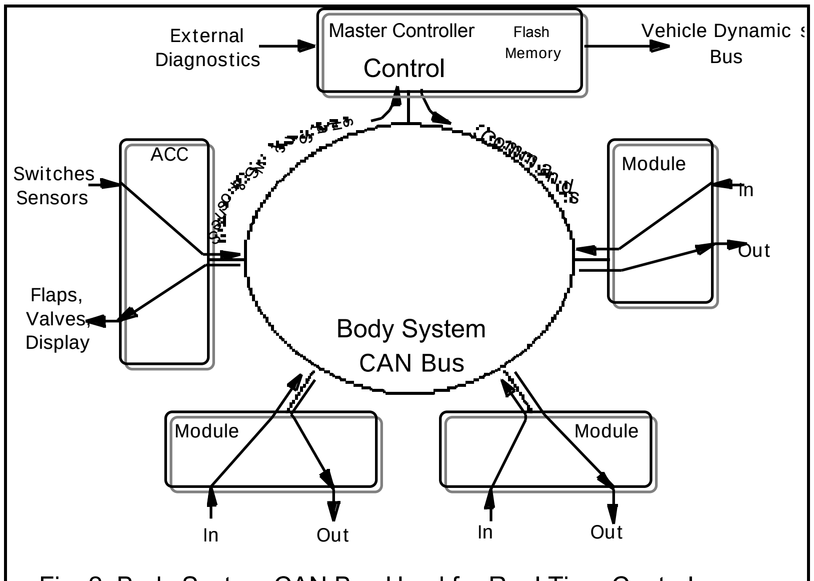

Fig. 2: Body System CAN Bus Used for Real Time Control

partitionning of the system (see refs. 1 & 2).

By using masked processors for local control, the only device that will need to run with external throw a bit more money at it in order to ensure that its external bus does not emit too much noise. For instance:

- It can be located away from the antenna and radio
- It can be enclosed in a fully screened box
- It can use a multi-layer pcb with an interior ground plane
- By keeping its I/O count low, then few harness wires will need filtering

The cost of these measures on every module would be considerable.

The flash memory used by the master module will need to be quite large, of the size that is more in demands in the computer & telecomms industries and therefore more economical. The processor will be need to be multi-tasking, and as it has no hard disk, it will need a lot more ram than conventional micros in order to save its state when switching between tasks. Here again, the computer industry can help as 1 Mbit X 16 static rams are not expensive. As far as the main processor goes, both Motorola and Intel offer basic 16 bit CMOS microprocessors as mature low cost devices, much cheaper than most 8 bit microcontrollers, and sold in large volumes into the telecomms market. All the support peripherals (I/O, watchdog, timers, CAN interfaces, diagnostic interfaces, DMA, etc.) can be mopped up into a single asic. The density of submicron logic makes this highly cost-effective at reasonable volumes.

Our examination of typical architectures shows that there are major savings in the cost of electronics with this strategy.

One criticism of over-centralisation is that it makes the cost of replacing any one module too high which gives the vehicle a reputation of being expensive to maintain. This strategy actually avoids this trap because the cost of replacing the master module is relatively low. It contains a single pcb, no expensive peripherals or power electronics. It needs to be sold with an empty flash memory, so that the cost of the software is not charged to the customer when the unit has to be replaced due to a hardware failure. After all, if you replace the hard disk of your computer, you do not expect to have to buy a new set of software for it.

The master module would be able to enhance vehicle security by implementing a challenge-response program to all the other modules on the bus to confirm their correct identity.

## 4. Software Structure

The software needs to run under an operating system, or at least under the control of a task scheduler. The aim is to ensure that overall system performance can be guaranteed. The conditions can change with the need to upgrade the software, and as such, it is important that all the bus scheduling continues to meet real time deadlines. The system will therefore benefit from running under a communications manager such as Volcano (ref. 5). This is a software kernel developed by NRTT that manages all the communications over the CAN bus. It defines the message format and priority of all the traffic. Volcano arranges the packing of messages and the assignment of priorities to each to guarantee that each message meets the real time delivery requirements defined for it.

The implementation of such a master controller will require organisational changes in the car companies, so that each department will lose ownership of its individual control module. In its place, an electronic system team will develop, which will have ever more responsibility for the management and control of diverse specifications and software modules.

In the same way as development and ownership of engine management software has returned to the car makers from component suppliers, so control of the body system software is likely to return to the car makers. This trend will see a further divorcing of software design from hardware design in the electronic component suppliers as part of the software for a particular control function may actually run in a different module made by a different supplier possibly located on a different continent!

1.  The Motor Industry Software Reliability Association:  "Development Guidelines for Vehicle Based Software", available from MIRA, Watling Street, Nunneaton CV10 0TU, UK (ISBN 0 9524156 0 7).

2.  Motor Industry Software Reliability Association: "Report 1: Diagnostics and Integrated Vehicle Systems", available from MIRA, Watling Street, Nunneaton CV10 0TU, UK.

3.  Lawrenz, Wolfhard:  "Worldwide Status of CAN - Present and Future", in Proceedings of the 2nd. International CAN Conference (CIA).

4.  Parkes, A.M. & Franzen, S. "Driving Future Vehicles",  published by Taylor & Francis Ltd. 4 John Street, London WC1N 2ET, ISBN 0 7484 0042 7.

5.  Tindell, K. & Hansson, H.:  "Babbling Idiots, Dual Priority Protocol and Smart CAN Controllers" in  Proceedings of the  2nd. International CAN Conference (CIA).