MICHAEL BACHMAN & JOHN ZALABAK, Curtis Instruments, USA

PURE LOGIC

Smarter vehicles require smarter components. But generic parts need customizing, which can be a difficult undertaking. Distributed logic could be the answer

Industrial EV designers are faced with a triple challenge: reducing vehicle design costs and time-to-market, while creating increasingly sophisticated vehicles. Ever since intelligent components became an integral part of vehicular design and marketing, OEMs have turned to leading industry electronic suppliers, such as Curtis Instruments, to help address this challenge. In response, a new era of design has been ushered in, providing designers with new tools and an innovative approach to increase design efficiency and produce a better vehicle.

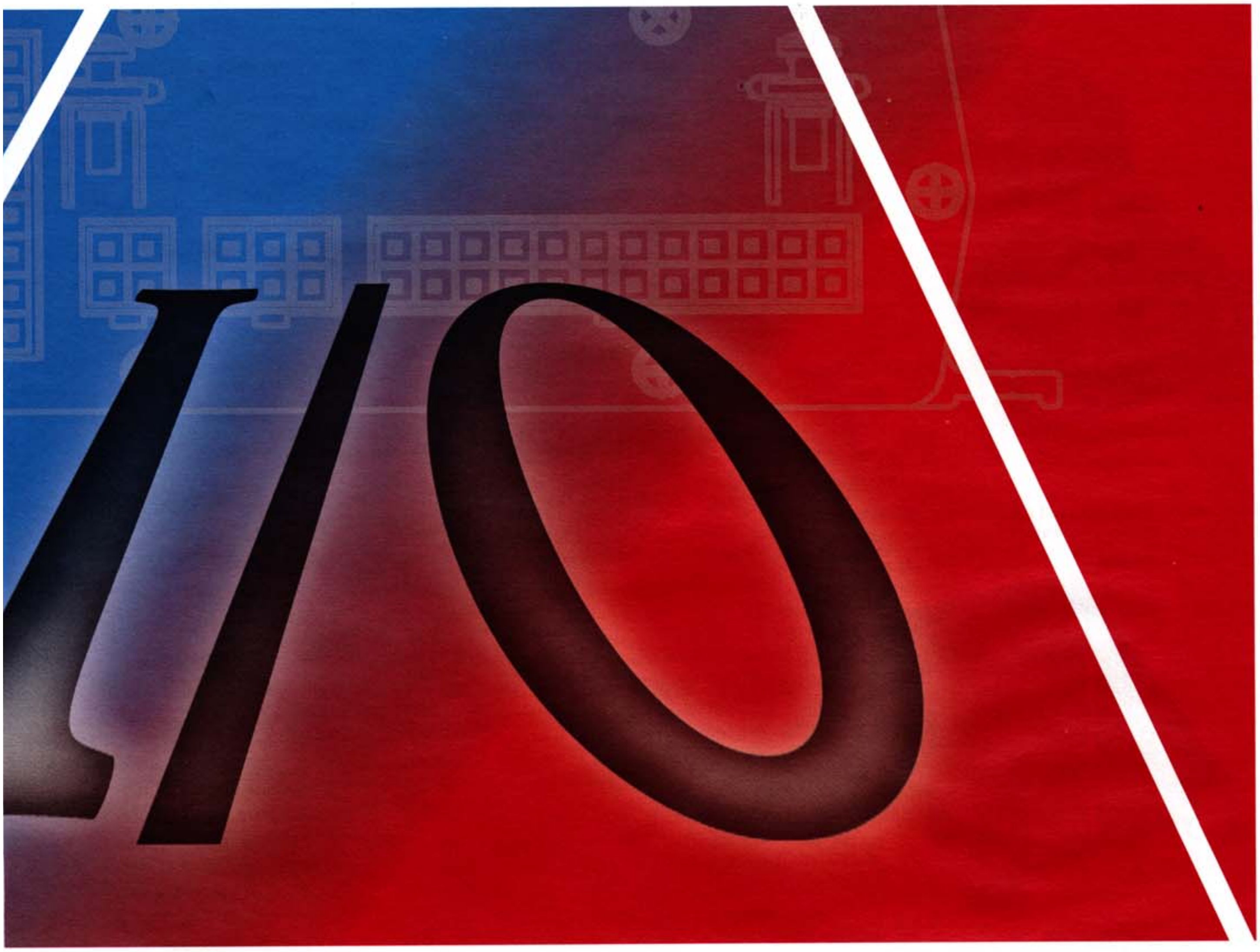
Centralized design or distributed logic and I/O?

As vehicles become increasingly intelligent, there is a need for a more systematic approach to design. Traditionally, vehicles are developed using a centralized design. Some vehicle applications encompass an 'all-in-one' microprocessor component encumbered with multiple jobs, including the task of managing the overall system. Other vehicles have a number of microprocessor components, e.g. one or more controllers, a display and, usually, a custom-made 'main interface unit' that manages the vehicle system.

In a centralized approach, generic components present serious limitations because they are fitted with predetermined features – I/O for tilt switches, inclinometers, accelerometers and photocells. These are just a select few of the many features that can be made available. Many predetermined features – sometimes up to 75% of the I/O on generic components – cannot be used by a particular application. In order to include the features that will give the vehicle the unique 'feel' they want, designers must either customize generic components, add custom components to the vehicle system, or both. Either way, this type of customization complicates the design process and, in most cases, makes vehicle maintenance and upgrades more difficult.

Although generic components are cost-effective in that OEMs can buy off-the-shelf products, in most centralized designs, generic components are not truly efficient – OEMs, in effect, buy more power and features than they are able to use. Customizing components can be a difficult undertaking. Often, designers must rely on suppliers for the custom work. Properly educating and training suppliers on equipment, and the continual discussions between the various experts, can substantially increase the vehicle's development costs and time-to-market.

A vehicle design based on distributed logic and I/O unifies the vehicle resource, making it available to the entire vehicle system. Two key components make up this type of design. One is the CANbus, which physically connects the system and allows a virtual network of I/O; the second is Vehicle Control Language (VCL). Curtis developed VCL, a powerful programming language, specifically for the industrial



EV industry. With this high-level language, which is user-friendly and easy to use, designers can dynamically configure the system, implement custom functions and control the vehicle.

Efficiency and flexibility

Efficiency and flexibility can be achieved in a design that allows for distributed vehicle resource (logic and I/O) and the programming of that resource at the vehicle system level. With the combination of CAN and VCL, designers can now develop this type of system.

In effect, microprocessors and I/O on motor controllers, displays and system software are universal resources, so designers can therefore utilize them completely when configuring the system. VCL provides the flexibility to apply the resource to the vehicle system as the designer sees fit.

An example of designing a vehicle with distributed logic and I/O would be as follows – the designer decides what the vehicle system needs, and tallies up the amount of resource – such as controllers, memory, signals, pot switches, or solenoids – that will be required to accommodate the system needs. The system is then designed to operate at two levels: there is local processing to control – the motor, for example; and processing on a system level, integrating the operation of all components, and essentially controlling the system at large. Generic components stay generic, because customization, configuration, and control of the vehicle are programmed at the system level.

With this approach, components – such as a traction controller, for example – can be used specifically for the special operation it was designed to do. The designer can apply any additional power and I/O from CAN-based components to where it makes sense. Features can be placed to where they are most effective, whether for wiring purposes, to share software, or for proximity (to a temperature module, for example). In many cases, custom components – like semi-custom displays and custom-built main interface units – can be eliminated. All of this is possible because VCL enables the system level considerations to be applied via the network.

The CAN-compatible Curtis Model 1311 programmer for setting parameters and performing diagnostic functions

Curtis Models 1236 and 1238 are the new generation of CAN-compatible AC controllers

The Model 1310 system controller provides single-point 'master control' of multiple vehicle functions

VCL gives designers the capability to develop an integrated vehicle, and the power to customize and differentiate it

Creating that 'differentiating feel'

VCL is the tool that eases the task of setting up and managing the vehicle system. It incorporates over 40 years of Curtis's industry experience and knowledge. The program provides designers with predefined functions common to the industry and an extensive library of samples that demonstrate how to solve common problems. Features with regard to traction lift, steering, display, diagnostics, fault handling and maintenance can be seamlessly combined.

Now that designers can effectively utilize the resource they have, there is much more room for creativity. This creativity gives way to eliminating or incorporating components; combining or sharing features; creating additional functionality, or whatever the designer envisions when designing the vehicle. There is now the freedom to design systems, mix and match system resources and create an integrated vehicle.

Differentiation is also possible with VCL, a key advantage of the program. In the past, designers have had to educate their suppliers on the critical details of an application so that their vehicle can get the 'differentiating feel' they desire. Designers have often felt hampered by technology, or their resources, in their attempts to develop exclusive features. This new design approach overcomes these limitations.

Curtis can develop a program for an OEM's own particular application, or, because the program is easy to learn, VCL can be placed in the hands of the expert for the application. Instead of having to educate a software engineer on the concept of tightness connectivity (between the feel of the steering and the grip of the ground), the vibration engineer can more easily learn VCL and, apply his field expertise to tweak the system for the desired tightness. It is how these subtleties are applied that makes the difference between an average vehicle and an excellent one.

Making the job easier

VCL has been designed to be robust. In comparison to other languages, it enables rapid development of real-time control programs. It is relatively simple; safe in terms of resource allocation, and efficient. Yet its real strength lies in its ability to express solutions to real-time programming problems.

Because it is tightly integrated with the CANbus, VCL takes the pain out of system configuration and control. It liberates designers from operational details, while at the same time providing flexibility in establishing connections. CANbus is essentially a set of mailboxes, so with VCL it is easier to allocate or reallocate mailboxes to a particular node on the system. Messages can be sent, either on demand, or automatically, to the extent that it enables a continuous stream of information to flow from one node to another. There is predictability as to the time and/or rate at which messages can be sent.

Traditionally, programming of this type is conducted in Assembler, which requires numerous instructions to complete a simple task. VCL covers two layers – it wraps up common operations, such as 'set up a CAN mailbox', and sequencing operations, such as 'send this message every 20 milliseconds', into one macro-operation. One VCL command can save dozens to hundreds of assembler instructions. The language is loaded with abstractions that simplify the programming.

Yet, VCL goes one step further. Built into the program is the notion of the importance of time. Events can be synchronized and sequenced, and their timing is guaranteed. For example, signal changes can be set up to run automatically, and passed off to another function, such as when the signal goes off and the vehicle stops within 50 milliseconds. This structure is seamlessly integrated into the language to allow design efficiency.

Distributed logic is here

Designing vehicles using distributed logic and I/O has many advantages. Cost savings are practically guaranteed – CAN solutions help reduce the expenses associated with wiring and maintenance. Add VCL to the picture and the savings can be significant. OEMs purchase only the resource they need, not more. Customization of components can be eliminated, while vehicle integration features can be gained. Time-to-market and product change cycles benefit considerably, and vehicle maintenance and warranty costs are reduced.

Another advantage to consider is flexibility. VCL gives designers the capability to develop an integrated vehicle, and the power to customize and differentiate the vehicle can be placed directly in their hands. VCL also works with Flash programmable memory, which allows designers to instantly make modifications to the vehicle at any development phase, from prototyping to final field-testing.

For the last 10 years, the industrial EV industry has needed a better design approach. The time has come – distributed logic and I/O is here today. It is practical – every designer can take advantage of the approach and every vehicle can benefit from it. Curtis presents an effective, efficient and affordable solution. It gives vehicle designers the capability to streamline the design process, together with the design and application flexibility to build intelligent, custom vehicles. ***iVT***