A Business Introduction To CAN

Version 1.1
2008-08-14
Application Note AN-AND-1-100

Author(s)	Bruce Emaus
Restrictions	Public Document
Abstract	This application note is an introduction to the business side of the CAN Protocol.

Table of Contents

1.0 Overview

Controller Area Network, which is almost always referred to as CAN, is a widely used communication system used to interconnect electronic modules into a functioning distributed embedded product. Such distributed product architectures are rapidly replacing the old world of centralized architectures. Why? For business reasons.

2.0 CAN – A Worldwide Standard

Developed at Bosch during the mid '80s for automotive applications, the CAN protocol has emerged as a worldwide standard, both in the automotive industry and in several other industries, including agriculture, heavy truck, bus, construction equipment, marine, industrial automation, and medical.

For the automotive industry, the CAN protocol has essentially replaced J1850 and all earlier proprietary protocols which were individually developed and supported by each car company. CAN has been adopted by many other companies which use distributed product architectures to replace their internally developed proprietary protocol -- in most cases a UART-based protocol.

2.1 CAN Makes Business Sense

The basic reason for this widespread standardization is quite simple -- CAN is a lower-cost solution than any other serial communication method. CAN is more robust than a UART-based protocol and less expensive than using Ethernet.

It is true that the cost of CAN silicon is perhaps ten times higher than the cost of UART silicon. But how much does the UART cost today? (Below one penny). The cost differential is no longer business relevant. However, when the accountants factor in the additional business cost of the internal supporting infrastructure necessary to take care of a proprietary protocol, the case for selecting CAN becomes much easier to justify. The wide availability of CAN Controllers, CAN transceivers, CAN software, CAN knowledge, CAN tools, and CAN engineering services is a low cost substitute compared to using internal resources.

2.2 Saving Resources By Abandoning Proprietary Protocols

Many companies began with UART-based protocols as the basis of their distributed product architectures. For these pioneers, the creation of a proprietary protocol has produced economic value, but it also consumed resources. In most cases, the CAN protocol may not have been available at the start or was too costly to consider at the time.

Creating a proprietary protocol takes considerable time, and requires the company to create its own tools. The continued engineering effort to support a proprietary protocol constantly consumes human resources. Is it worth continuing with a proprietary protocol? This question is easy to answer when the supporting engineering staff leaves the company: a change is absolutely required. In almost all cases the replacement protocol is CAN. So from a practical point of view, your company will most likely change from what you have today to the CAN Protocol sometime in the future.

Is the CAN Protocol better than your company's current proprietary solution? While there may be a few exceptions, in almost every case the answer is yes. CAN is a superior technical and business alternative.

2.3 Reasons To Use CAN

From the business perspective, the key reasons to use CAN include:

- Low-cost CAN hardware components – widely available
- Low-cost CAN tools – in comparison to making your own tools
- Low-cost CAN software components – in comparison to writing your own software
- Low-cost CAN training – in comparison to making your own training classes
- Low-cost CAN protocol improvements – compared to upgrading your own proprietary protocol

On the technical side, CAN is better than most UART-based proprietary protocols for many reasons, including:

- CAN operates at the message level – the UART functions at the byte level, and message handling must be implemented by additional communication software
- CAN has automatic error handling – because the UART essentially has no error handling, this activity must be implemented by additional software
- CAN is more reliable than the UART – the academic community has reported an undetected error rate of one message for CAN every 1000 years – the undetected error rate of most UART-based protocols is relatively unknown.

2.4 CAN Cost Savings

Transitioning your distributed products from your proprietary protocol to the use of CAN should produce savings and reduce the level of resources that were previously used. The major savings include:

- Zero resource expenditures for developing and maintaining protocol analysis tools
- Zero resource expenditures for fixing and maintaining the protocol – the industry handles this
- Near zero resource expenditures for CAN communication software development – if off-the-shelf components are used
- Zero resource expenditures for documenting the communication hardware – widely available

- Zero resource expenditures for documenting the communication software – if off-the-shelf components are used

Essentially a large portion of your company resources that have been devoted to managing and maintaining your own proprietary protocol can be moved to new engineering activities – perhaps more product development. This is exactly why the car companies ended the creation of their own protocols – they realized that their business is making cars, not protocols.

Technical staff previously involved in developing tools for proprietary protocols may be moved to a new area.

3.0 Key Business Considerations

3.1 CAN Tools

While most CAN developers, especially in the automotive industry, use the higher-powered Vector tool CANoe, the beginning engineer who is new to CAN is perhaps better off using the more economical CANalyzer. True, other lower-cost tools are available for getting started, but most of these tools fail at one very important point – the ability to detect Error Frames. This somewhat obscure subtlety of the CAN Protocol, which usually has no importance during the beginning portion of the learning curve, means the encounter requires a certain amount of self-discovery by the engineer. Once it is discovered that Error Frame detection is extremely important, many make the transition to CANalyzer.

During one visit to a customer, the question was asked, “Why would the engineers at our parent company suggest using Vector tools instead of following the corporate directive to use the internally generated tools?” The answer was simple – the word was spreading throughout the engineering community that Vector tools were superior to what was being created internally.

While some companies always choose the low-cost path, others recognize the significant time savings when using the right tools. Few engineers use low-cost DVMs to solve difficult problems, but instead they use the oscilloscope or logic analyzer.

4.0 Using Off-The-Shelf CAN Communication Software

Many companies use third-party off-the-shelf CAN communication software components rather than devote internal resources to develop their own. One key advantage of using off-the-shelf CAN communication software is the significant reduction in overall software development. Why spread your resources across both the communication software and the application software activities? Instead, using prepackaged communication software lets your company concentrate on its core product software development.

Others have chosen off-the-shelf software simply to reduce the time to market.

While the off-the-shelf software concept may be somewhat unfamiliar to you, its practice is widespread in the automotive industry. The reason for this is simple. Virtually all car companies realize that off-the-shelf communication software offers a consistency level well beyond that of the software, which is developed by suppliers individually. This is one of the major lessons the auto industry has learned – misinterpreted communication specifications equals delayed programs.

Vector is a major off-the-shelf software component supplier for the auto industry. Please contact us for more information about Vector software components. We have several application notes to introduce the general concepts, both from the business and technical viewpoints.

4.1 Vector Training And Consulting Services

In addition to tools and off-the-shelf CAN communication software components, Vector also offers training, technical consulting, and business consulting services.

Vector training classes and consulting services will make you proficient in the following areas:

- CAN Protocol
- CANalyzer
- CANoe
- CAN communication software
- CAN Physical Layer Development – an on-site technical workshop
- Developing Distributed Embedded Systems – an on-site technical and business workshop

5.0 Other Potentially Relevant Business Considerations

5.1 Using A CAN-Based Supplier Model

If a company standardizes its CAN implementation for its entire distributed product architecture, then the CAN hardware and CAN communication software can be reused for all of the company's products. Every implementation is the same – only the application is different. While such corporate standardization does require a certain amount of systems engineering effort, the overall development savings is quite significant. If a company chooses to use a group of outside suppliers to simultaneously co-develop the various modules that make up the company's distributed product line, then distributing the reusable communication components to the supplier base can be a big advantage. Once the network portion of the product has been developed and placed on the shelf, the engineering development cost for it stays near zero.

This is exactly the model used by the automotive industry. All of the vehicle OEM's Tier 1 suppliers use the same CAN hardware design and use identical CAN communication software. At the product or vehicle level, the CAN-based network is consistent across the entire enterprise (the OEM and its suppliers).

5.2 Cost To Upgrade To CAN From A Proprietary Protocol

There are, of course, expenses involved to transition your distributed products from your existing proprietary protocol to the use of CAN. These expenditures include:

- Redesigning the communications interface circuitry to use CAN transceivers
- Selecting a suitable CAN Controller or a microcontroller with an integrated CAN Controller
- Redesigning the product's printed circuit board
- Developing or purchasing CAN communication software
- Purchasing CAN tools

While some of the existing staff, especially the UART-based protocol experts, will find the transition relatively easy and perhaps a simple academic exercise, you may consider sending one or more individuals to training classes to learn the CAN protocol and to learn how to use some of the more powerful CAN tools.

5.3 Getting Engineering To Buy Into The Transition To CAN

If your company is currently not using a distributed product architecture, the adoption of the CAN Protocol is relatively simple. Engineering finds the transition both stimulating and quite challenging. New processes are required and systems engineering becomes much more important to the success of the new CAN-based design.

However, if your company already uses a proprietary protocol, there may be significant resistance in transitioning your products' communication system to the CAN Protocol. Will your technical staff embrace CAN? The answer to this depends on the type of individuals that have been a part of your internal infrastructure that supports your current proprietary protocol. If these individuals are the architects of all of your company's proprietary protocol, these individuals will tend to oppose change. "Pulling the plug on your own infrastructure" is not very characteristic of most engineering departments.

At one company, when management discovered that the last staff members who supported the company's proprietary protocol had left the company, the rapid transition to the CAN protocol was of high strategic importance. The business situation required immediate attention to redesign the CAN Protocol into the product.

6.0 Contacts

Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
Germany
Tel.: +49 711-80670-0
Fax: +49 711-80670-111
Email: info@vector-informatik.de

Vector CANtech, Inc.
39500 Orchard Hill Pl., Ste 550
Novi, MI 48375
USA
Tel: +1-248-449-9290
Fax: +1-248-449-9704
Email: info@vector-cantech.com

VecScan AB
Theres Svenssons Gata 9
41755 Göteborg
Sweden
Tel: +46 (0)31 764 76 00
Fax: +46 (0)31 764 76 19
Email: info@vecscan.com

Vector France SAS
168 Boulevard Camélinat
92240 Malakoff
France
Tel: +33 (0)1 42 31 40 00
Fax: +33 (0)1 42 31 40 09
Email: information@vector-france.fr

Vector Japan Co. Ltd.
Seafort Square Center Bld. 18F
2-3-12, Higashi-shinagawa,
Shinagawa-ku
Tokyo 140-0002
Japan
Tel.: +81 3 5769 7800
Fax: +81 3 5769 6975
Email: info@vector-japan.co.jp

Vector Korea IT Inc.
Daerung Post Tower III, 508
182-4 Guro-dong, Guro-gu
Seoul, 152-790
Republic of Korea
Tel.: +82-2-2028-0600
Fax: +82-2-2028-0604
Email: info@vector-korea.com