

O Protocolo CAN: Entendendo e Implementando uma Rede de Comunicação Serial de Dados baseada no Barramento “Controller Area Network”.

Alexandre de Almeida Guimarães

General Motors do Brasil Ltda

Universidade de São Paulo

Antônio Mauro Saraiva

Universidade de São Paulo

Copyright © 2002 Society of Automotive Engineers, Inc

RESUMO

Os objetivos deste artigo são explicar os conceitos do protocolo CAN e demonstrar como iniciar a implementação de uma rede de dados nele baseada. Destacam-se, entre outras coisas, seus conceitos fundamentais, os formatos disponíveis das mensagens, as principais normas existentes, as formas de detecção de falhas e os fundamentos de um Dicionário de Dados. São abordados os conceitos principais que devem ser observados no momento de se desenvolver um módulo e um barramento CAN, além dos aspectos de hardware e software em relação aos módulos e à geometria do barramento.

ABSTRACT

The objectives of this paper are to explain CAN bus main concepts and presents how to initiate a development of a CAN based network. Highlights, among other things, its primary concepts, available message formats, main existing norms, failure detection mechanisms and primary Data Dictionary concepts. Presents also main concepts should be considered when developing a CAN based module, its Hardware and Software aspects and network geometry.

PALAVRAS-CHAVE

CAN, Controller Area Network, Multiplexing, ISO11898, SAE J1939.

INTRODUÇÃO [1]

Atualmente, percebe-se um crescimento exponencial dos módulos eletrônicos disponibilizados nos veículos automotivos. Estes módulos passam, cada vez mais, a compartilhar informações entre si; muitas vezes em tempo-real.

Uma das formas de se conectar estes módulos, uns aos outros, é através de uma **Arquitetura Centralizada**. A figura #1 ilustra este tipo de arquitetura.

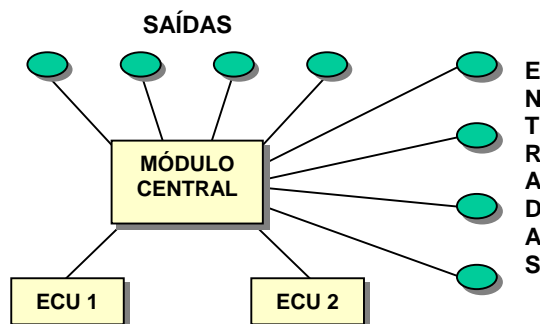


Figura #1: Diagrama Esquemático de uma Arquitetura Centralizada.

Como vantagens desta arquitetura, pode-se destacar:

- Simplicidade do hardware utilizado, sendo constituído basicamente pelos sensores, o cabeamento e uma placa eletrônica de gerenciamento do sistema.
- Todos os dados estarão disponíveis à Unidade Central durante toda a operação do sistema.

Como desvantagens, pode-se destacar:

- Grande quantidade de cabeamento requerido.
- Limitação das possibilidades de expansão do sistema.

Outra forma de conectá-los é através de uma **Arquitetura Distribuída**. Neste caso, existiriam vários módulos inteligentes espalhados pela aplicação, recebendo apenas uma parcela dos dados, geralmente aqueles gerados próximo a ele mesmo, e enviando-os aos módulos que necessitem de tais informações para seu próprio processamento.

A figura #2 ilustra este tipo de arquitetura.

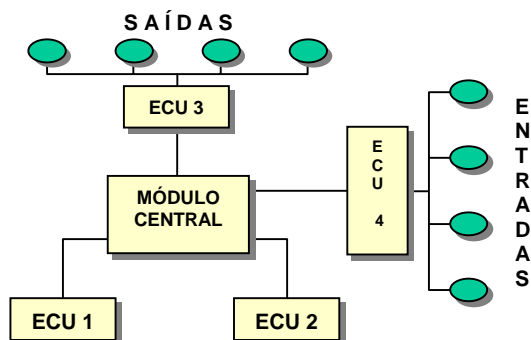


Figura #2: Diagrama Esquemático de uma Arquitetura Distribuída.

Como vantagens desta arquitetura, pode-se destacar [2], [3]:

- Reduzida quantidade de cabeamento da rede, por estarmos instalando módulos remotos bem próximos aos sensores.
- Permite a ampliação do sistema com significativa facilidade.
- Permite a aplicação de protocolos de comunicação mais inteligentes, o que aumentaria ainda mais a confiabilidade do sistema.
- Facilita a criação do software de aplicação, uma vez que possibilita a sua modularização e distribuição pelos vários módulos remotos.
- Possibilita também a modularização do projeto do sistema e da execução dos testes de validação, aumentando a sua confiabilidade e reduzindo os prazos envolvidos no desenvolvimento.

Como desvantagens desta arquitetura, pode-se destacar:

- Difícil implementação do software de controle da rede, o que depende diretamente do protocolo escolhido.
- Difícil determinação das prioridades de transmissão dos dados.
- Difícil determinação da taxa de transmissão ideal para uma dada aplicação [4].

Do ponto de vista de uma aplicação qualquer, alguns requisitos mínimos devem ser observados no momento de se desenvolver uma rede embarcada de comunicação de dados. Pode-se destacar os seguintes [5], [6]:

- Estar preparada para trabalhar em ambientes móveis, sendo resistente especialmente a elevada vibração dos equipamentos.

- Ter cabeamento reduzido para facilitar a sua instalação e reduzir os problemas com manutenção.
- Ser imune as interferências eletro-magnéticas.

Considerando a Arquitetura Distribuída a melhor solução para aplicações onde vários módulos estarão responsáveis pelo controle de sistemas específicos, porém, compartilhando dados, a forma como estes dados serão transmitidos de um módulo para o outro passa a ser o principal desafio a partir de agora.

Dentre as características desejáveis em um protocolo para aplicações veiculares, pode-se destacar [7]:

- Cabeamento total da rede reduzido.
- Ser capaz de transmitir altas taxas de informação, uma vez que os sistemas operam com informações em tempo-real.
- Boa flexibilidade das linhas de comunicação para facilitar a instalação do chicote no veículo.
- Capacidade de interligar diversos nós, garantindo futuras expansões do sistema.
- Trabalhar dentro do conceito multi-mestre, eliminando a existência de um módulo principal e responsável pelo gerenciamento da rede.
- Robustez suficiente para garantir seu funcionamento em ambientes nocivos e móveis.
- Capacidade para detectar e tratar eventuais falhas geradas por problemas em hardware e software ou, interferências externas como as eletro-magnéticas.

O protocolo que cobre todos estes requisitos é o CAN (Controller Area Network).

O CAN BUS (BARRAMENTO CAN)

CONCEITUAÇÃO GERAL [8]:

O CAN é um **protocolo de comunicação serial síncrono**. O sincronismo entre os módulos conectados a rede é feito em relação ao início de cada mensagem lançada ao barramento (evento que ocorre em intervalos de tempo conhecidos e regulares).

Possui características interessantes, como o fato de ser **multi-mestre**, onde todos os módulos podem se tornar mestre num determinado momento e escravo num outro, e trabalhar com mensagens **multicast**, onde todas as mensagens são recebidas por todos os módulos da rede.

Outro ponto forte deste protocolo é o fato de ser fundamentado no conceito **CSMA/CD with NDA (Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration)**. Isto significa que todos os

módulos verificam o estado do barramento, analisando se outro módulo não está enviando mensagens com maior prioridade. Caso isto seja percebido, o módulo cuja mensagem tiver menor prioridade cessará sua transmissão e o de maior prioridade continuará enviando sua mensagem deste ponto, sem ter que reiniciá-la.

Outro conceito bastante interessante é o **NRZ (Non Return to Zero)**, onde cada bit transmitido representa efetivamente um dado e não, por exemplo, o barramento em *idle* (sem operação efetiva).

A velocidade de transmissão dos dados é proporcional ao comprimento do barramento. A maior taxa de transmissão especificada é de 1Mbps considerando um barramento de 40 metros. A figura #3 representa a relação entre o comprimento da rede (barramento) e a taxa de transmissão dos dados.

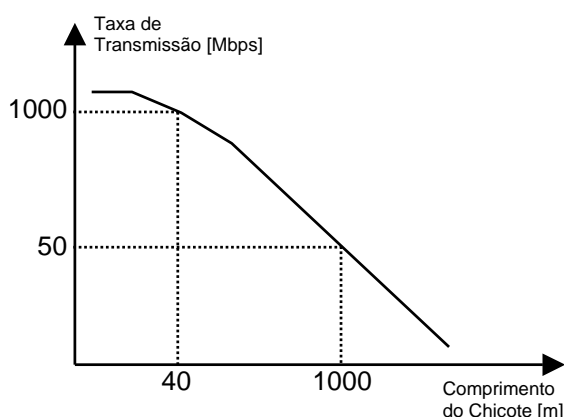


Figura #3: Relação entre a Taxa de Transmissão e o Comprimento da rede.

Considerando fios elétricos como o meio de transmissão dos dados, existem 3 formas de se constituir um barramento; dependentes diretamente da quantidade de fios utilizados. Existem redes baseadas em 1, 2 e 4 fios. As redes com 2 e 4 fios trabalham com os sinais **CAN H (High)** e **CAN L (Low)**. No caso dos barramentos com 4 fios, além dos sinais de dados, um fio com o VCC e outro com o GND fazem parte do barramento, levando a alimentação aos módulos subsequentes da rede. As redes com apenas 1 fio têm este, o fio de dados, chamado de **CAN**.

Considerando o CAN fundamentado em 2 e 4 fios, estes devem ser trançados, não necessariamente blindados e os dados interpretados pela análise da diferença de potencial entre os fios CAN H e CAN L. Por isso, o barramento CAN é classificado como **Par Trançado Diferencial**. Este conceito elimina os efeitos causados por interferências eletro-magnéticas, uma vez que qualquer ação sobre um dos fios será sentida também pelo outro, causando flutuação nos sinais de ambos para o mesmo sentido e com a mesma intensidade. Como o que vale para os módulos que recebem as mensagens é a diferença de

potencial entre os fios (e esta permanecerá inalterada), a comunicação não é prejudicada.

No CAN, os dados não são representados por bits em nível "0" ou nível "1". São representados por bits **Dominantes** e bits **Recessivos**, criados em função da condição presente nos fios CAN H e CAN L. A figura #4 ilustra os níveis de tensão numa rede CAN, assim como os bits Dominantes e Recessivos.

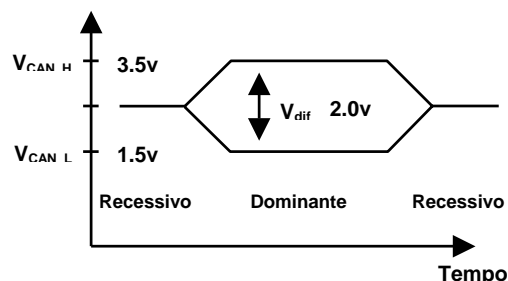


Figura #4: Níveis de Tensão no CAN e os bits Dominantes e Recessivos.

Como mencionado no início deste item, todos os módulos podem ser mestre e enviar suas mensagens. Da mesma forma, o protocolo é robusto a ponto de evitar a colisão entre mensagens através de uma **arbitragem** bit a bit não destrutiva. Imagine 2 módulos enviando, bit a bit, ao mesmo tempo, mensagens diferentes. Após enviar um bit, cada módulo analisa o barramento e verifica se outro módulo na rede o sobrescreveu. Vale lembrar que um bit Dominante sobrescreve eletricamente um Recessivo. Um módulo interromperá imediatamente sua transmissão caso perceba que existe outro módulo transmitindo uma mensagem com prioridade maior. Este, com maior prioridade, continuará normalmente sua transmissão.

FORMATOS [8]:

Existem 2 formatos de mensagens no protocolo CAN:

CAN 2.0A: Mensagens com identificador de 11 bits. É possível ter até 2048 mensagens numa rede constituída sob este formato, o que pode caracterizar uma limitação em determinadas aplicações. A figura #5 apresenta o quadro de mensagem do CAN 2.0A.



Figura #5: Quadro de Mensagem - CAN 2.0A.

CAN 2.0B: Mensagens com identificador de 29 bits. É possível ter até 5,3 milhões de mensagens numa rede constituída sob este formato. Percebe-se que a limitação em virtude da quantidade de mensagens não mais existe. O que poderá ser observado em alguns casos é que, os 18 bits adicionais no identificador aumentarão o tempo de transmissão de cada mensagem, o que poderá caracterizar

um problema em determinadas aplicações que trabalhem em tempo-real. A figura #6 apresenta o quadro de mensagem do formato CAN 2.0B.



Figura #6: Quadro de Mensagem - CAN 2.0B.

PADRÕES [1]:

Os fundamentos do CAN são especificados por duas normas, a ISO11898 e a ISO1519-2. A primeira, **ISO11898**, determina as características de uma rede trabalhando com alta velocidade de transmissão de dados (de 125Kbps à 1Mbps). A segunda, **ISO1519-2**, determina as características de uma rede trabalhando com baixa velocidade (de 10Kbps à 125Kbps).

Ambos os padrões especificam as camadas física e de dados, respectivamente #1 e #2, se considerado o **padrão de comunicação OSI de 7 camadas (ISO7498)**. As demais camadas são especificadas por outros padrões, cada qual relacionado a uma aplicação específica.

Existem diversos padrões fundamentados no CAN, dentre os quais pode-se destacar:

- NMEA 2000: Baseado no CAN 2.0B, este padrão é utilizado em aplicações navais e aéreas.
- DIN 9684 – LBS: Baseado no CAN 2.0A, este padrão é utilizado em aplicações agrícolas.
- ISO 11783: Baseado no CAN 2.0B, este padrão também é utilizado em aplicações agrícolas.
- SAE J1939: Baseado no CAN 2.0B, este padrão é utilizado em aplicações automotivas, especialmente ônibus e caminhões.

Cada um dos padrões supracitados especifica as camadas de Rede (#3), Transporte (#4), Sessão (#5), Apresentação (#6) e Aplicação (#7), inclusive as mensagens pertinentes ao seu dicionário de dados.

Por diversas vezes, quando se analisa um protocolo de comunicação [7], este se encontra associado a uma determinada classe, que poderá ser “A”, “B” ou “C”. A **Classe “A”** abrange os protocolos que trabalham em baixa velocidade (até 10Kbps). A **Classe “B”** abrange os protocolos que trabalham em média velocidade (de 10Kbps à 125Kbps). A **Classe “C”**, por fim, abrange os protocolos que trabalham em alta velocidade (de 125Kbps à 1Mbps). O CAN está classificado dentro das classes “B” e “C”, dependendo diretamente da aplicação analisada.

DETECÇÃO DE FALHAS [7] [8]:

Uma das maiores vantagens do CAN é a sua robustez e a capacidade de se adaptar a condições de falhas, temporárias e permanentes. Podem-se classificar as falhas em 3 categorias ou níveis:

Nível de Bit: Dentre esta categoria, 2 tipos de erro são possíveis:

Bit Monitoring: Após a escrita de um bit dominante, o módulo transmissor verifica o estado do barramento. Se o bit lido for recessivo, significará que existe um erro no barramento.

Bit Stuffing: Apenas 5 bits consecutivos podem ter o mesmo valor (dominante ou recessivo). Caso seja necessário transmitir 6 ou mais bits iguais, o módulo transmissor inserirá um bit contrário após cada grupo de 5 bits iguais e consecutivos. O módulo receptor ficará encarregado de, durante a leitura, retirar automaticamente este bit, chamado de *Stuff Bit*. Caso uma mensagem seja recebida com pelo menos 6 bits consecutivos iguais, algo de errado estará ocorrendo no barramento.

Nível de Mensagem: Dentre esta categoria, 3 tipos de erro são possíveis:

CRC ou Cyclic Redundancy Checks: Funciona como um *checksum*. O módulo transmissor calcula um valor em função dos bits da mensagem e o transmite juntamente com ela. Os módulos receptores recalculam este *CRC* e verificam se este é igual ao enviado.

Frame Check: Os módulos receptores analisam alguns bits da mensagem recebida. Estes bits não mudam de mensagem para mensagem e são determinados pelo padrão CAN.

Acknowledgment Error Check: Os módulos receptores respondem a cada mensagem boa recebida, escrevendo um bit dominante no campo *ACK* de uma mensagem resposta que é enviada ao módulo transmissor. Caso esta mensagem resposta não for recebida (pelo transmissor da mensagem de dados original), significará que, ou a mensagem de dados transmitida estava corrompida, ou nenhum módulo a recebeu.

Toda e qualquer falha acima mencionada, quando detectada por um ou mais módulos receptores, fará com que estes coloquem uma mensagem de erro no barramento, avisando toda a rede de que aquela mensagem continha um erro e que o transmissor deverá reenviá-la.

Além disso, a cada mensagem erroneamente transmitida/recebida, um contador de erros é incrementado em “1” unidade nos módulos receptores, e em “8” unidades no transmissor. Módulos com este contador igual a zero são considerados **Normais**. Para o contador contendo valores entre 1 e 127, o módulo é considerado **Error Active**.

Contadores contendo valores entre 128 e 255 colocam o módulo em condição de **Error Passive**. Finalmente, contadores contendo valores superiores a 255, o módulo é considerado em **BusOff** e passa a não mais atuar no barramento. Estes contadores também são decrementados a medida que mensagens corretas são recebidas, reduzindo o grau de incerteza em relação a atividade dos módulos ora com contadores contendo valores diferentes de zero.

Nível Físico: Para os barramentos com 2 e 4 fios, caso algo de errado venha a ocorrer com os fios de dados CAN H e CAN L, a rede continuará operando numa espécie de modo de segurança. Seguem abaixo as condições que permitem a continuidade da comunicação:

- Curto do CAN H (ou CAN L) para GND (ou VCC);
- Curto entre os fios de dados CAN H e CAN L;
- Ruptura do CAN H (ou CAN L);

DICIONÁRIO DE DADOS [9]:

É a parte mais dedicada a aplicação quando se trabalha com um protocolo como o CAN. O Dicionário de Dados (ou *Data Dictionary*) é o conjunto de mensagens que podem ser transmitidas naquela determinada rede.

A forma mais interessante de se organizar um dicionário de dados é criando uma matrix com todos os módulos da rede. Esta matrix mostrará cada mensagem sob a responsabilidade de cada módulo e também quem a transmite e quem precisará recebê-la. Outros dados importantes desta matrix são: o tempo de atualização dos valores da mensagem, o intervalo de transmissão da mesma e o valor relativo ao seu identificador. Além desta matrix, a documentação referente ao Dicionário de Dados deverá conter uma descrição detalhada de cada mensagem, bit a bit.

O Dicionário de Dados é implementado numa rede CAN via software e deverá ser o mesmo em todos os módulos conectados a rede. Isto garantirá total compatibilidade entre os participantes do barramento.

EXEMPLO DE UMA REDE CAN [8]:

Uma rede CAN, dependendo da sua aplicação, poderá ter até centenas de módulos conectados. O valor máximo para a conexão de módulos num barramento depende do padrão que se utiliza naquela dada aplicação.

Toda rede CAN possui 2 **Terminadores**. Estes terminadores nada mais são que resistores entre 120 e 124 ohms, conectados a rede para garantir a perfeita propagação dos sinais elétricos pelos fios da mesma. Estes resistores, um em cada ponta da rede, dão a impressão aos módulos de que o barramento é infinito.

Outra característica de determinadas aplicações fundamentadas no CAN é que estas poderão ter duas ou mais sub-redes CAN trabalhando, cada qual, em uma velocidade diferente. Os dados são transferidos de uma sub-rede para a outra através de módulos que atuam nas duas sub-redes. Estes módulos são chamados de **Gateways**.

A figura #7 ilustra uma rede CAN genérica, com duas sub-redes e os 2 terminadores ilustrados. O *Gateway* desta aplicação é o Painel de Instrumentos.



Figura #7: Exemplo de Rede CAN – Destaque para os Terminadores.

O PROJETO DE UMA REDE CAN EXPERIMENTAL

Os próximos itens procuram ilustrar os pontos principais que devem ser considerados durante o desenvolvimento de um módulo e uma rede baseados no protocolo CAN.

HARDWARE DO MÓDULO [10]:

Os módulos eletrônicos são geralmente chamados de *ECUs* (*Electronic Control Unit*). Se analisarmos uma ECU CAN, encontraremos dois componentes básicos: O Microcontrolador e o Transceiver.

Uma ECU pode ser desenvolvida considerando dois microcontroladores; um deles para o processamento do programa, gerenciamento da memória e controle das entradas e saídas (I/Os) e outro para o gerenciamento do protocolo propriamente dito. Por outro lado, pode ser utilizado um microcontrolador que agregue todas estas funções num único CI.

O **Microcontrolador** deve ser escolhido em função da aplicação a que se destina a ECU. Existem microcontroladores com dezenas de I/Os digitais e analógicos, portas de comunicação baseadas em UART, I²C e Paralela e também saídas PWM. Além da capacidade de controlar o CAN.

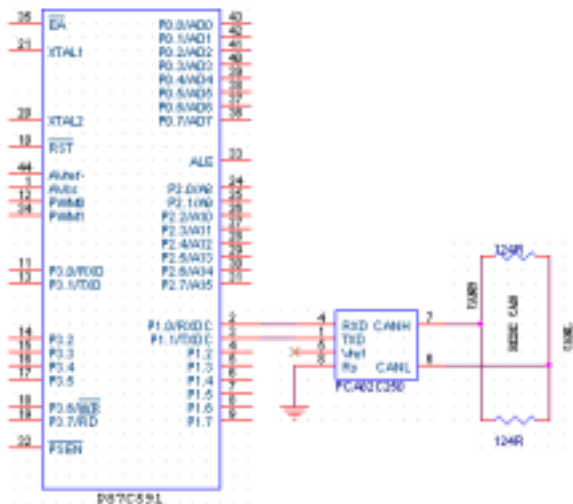
Outro ponto que deve ser observado é a memória interna do microcontrolador. Além da RAM, existem microcontroladores com memória interna EPROM, EEPROM, OTP e Flash EPROM. Uma alternativa durante o desenvolvimento, poderá ser utilizar uma memória RAM e uma memória EPROM externas, onde será executado o programa e armazenados os dados gerados. Neste caso, o microcontrolador deverá ser informado de que o programa

está alocado na memória externa. Isto é feito através de um pino disponível em todos os microcontroladores, geralmente chamado de *EA (External Access)*.

Em relação ao gerenciamento do protocolo, existem microcontroladores que trabalham somente com o formato 2.0A e microcontroladores que trabalham com ambos os formatos, 2.0A e 2.0B.

O **Transceiver** nada mais é que um CI que controla a Recepção e a Transmissão de um protocolo. Ele está conectado diretamente ao microcontrolador CAN e é responsável pela geração dos sinais que serão transmitidos por um meio específico. O CAN é geralmente utilizado para comunicação via fios elétricos, mas poderia ser implementado para comunicação via *MOST* (fibra ótica), RF ou até mesmo *Bluetooth*. Esta determinação do meio físico de transmissão é feita pelo Transceiver. Da mesma forma, existem transceivers que estabelecem barramentos com 1 ou 2 fios de dados, como também para determinadas velocidades de comunicação.

A figura #8 ilustra a conexão entre um microcontrolador com CAN incorporado e um transceiver CAN. Esta dupla em particular é formada por um microcontrolador com memória OTP e barramento com 2 fios de dados. Observe que o microcontrolador possui diversas outras funções (saídas PWM e comunicação via UART entre outras).



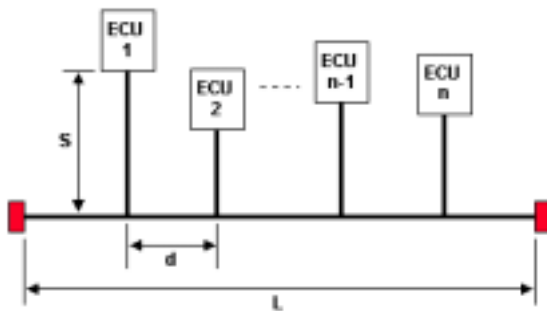
MONTAGEM DA REDE:

Após o desenvolvimento das ECUs, será preciso conectá-las através do chamado barramento. Vale lembrar que o termo “Bus”, do nome CAN Bus, significa exatamente barramento. Apesar de parecer simples, o ato de interligar os módulos requer bastante atenção.

Sobre o cabeamento necessário [8], considerando uma aplicação CAN de 2 fios, deve-se utilizar um par trançado, onde a secção transversal de cada um dos fios deve ser de no mínimo 0,35mm².

As 2 terminações (resistores de aproximadamente 120 ohms), do ponto de vista teórico, poderiam ser instaladas nas extremidades do chicote; diretamente nos fios de dados CAN H e CAN L. Do ponto de vista prático isto é extremamente complexo. O que deve ser feito é adicionar as terminações nas duas ECUs conectadas aos extremos da rede. Se a rede for embarcada em veículos e as ECUs forem montadas dependendo dos opcionais do mesmo, deve-se procurar instalar as terminações nas ECUs que estarão sempre presentes no veículo. As terminações são mandatórias numa rede CAN !

No momento de se projetar o roteamento do barramento [12], algumas regras em relação ao comprimento dos chicotes devem ser observadas. O sincronismo das operações das ECUs no CAN é fundamentado no tempo de propagação física dos elétrons no barramento. Assim, a relação do comprimento de determinados intervalos do chicote no barramento são fundamentais ao bom funcionamento da rede. A figura #10 mostra um diagrama que ilustra as medidas que devem ser observadas no desenvolvimento do chicote.



Onde: S (máximo comprimento da ramificação) = 0,3m
d (mínima distância entre ramificações) = 0,1m
L (máximo comprimento da rede a 1Mbps) = 40m
Obs: O valor da distância “d” deve ser aleatório.

Figura #10: Geometria de uma Rede CAN.

Após montado o barramento, caso seja necessário qualquer retrabalho no mesmo, é aconselhável a troca do chicote danificado. Emendas poderão alterar a impedância característica da rede e com isso afetar o seu funcionamento.

CONCLUSÕES

O CAN Bus mostra-se o protocolo de comunicação mais adequado às aplicações embarcadas em veículos. Sejam estas aplicações automotivas, navais ou agrícolas, o CAN é capaz de atender seus requisitos de robustez e gerenciamento da comunicação (taxas de transmissão compatíveis e detecção de falhas).

Ao se trabalhar com o CAN, deve-se observar qual a norma diretamente relacionada a aplicação em questão. Esta norma terá informações sobre a taxa de transmissão da rede e também sobre o dicionário de dados da aplicação, entre outras.

Durante o desenvolvimento do hardware de uma ECU CAN, os pontos principais a serem observados estão relacionados ao microcontrolador e ao transceiver. A determinação destes componentes está diretamente ligada a aplicação efetiva da ECU.

O desenvolvimento do firmware da ECU é uma das etapas mais importantes na criação do sistema. É nele que serão declaradas as mensagens da aplicação e também o funcionamento da ECU em relação aos sensores e atuadores do sistema.

A montagem efetiva da rede (barramento), apesar de parecer simples, envolve diversos conceitos relacionados a Física. Certamente, o projetista não precisará conhecer totalmente o comportamento dos elétrons através dos fios das linhas de dado, mas a observação de alguns conceitos básicos relacionados é primordial.

LISTA DE ABREVIações

ACK: Acknowledgement.

CAN Bus: Barramento Controller Area Network.

CI: Circuito Integrado.

CD: Collision Detection.

CRC: Cyclic Redundant Check.

CSMA/CD: Carrier Sense Multiple Access.

EA: External Access.

ECU: Electronic Control Unit.

EEPROM: Electrical Erasable and Programmable ROM.

EPROM: Electrical Programmable ROM.

I/O: Input/Output.

ISO: International Organization for Standardization.

Kbps: Kilo bits por segundo.

Mbps: Mega bits por segundo.

NDA: Non-Destructive Arbitration.

NRZ: Non-Return to Zero.

OSI: Open System Interconnection.

OTP: One Time Programmable.

PC: Personal Computer.

RAM: Random Access Memory.

ROM: Ready Only Memory.

SAE: Society of Automotive Engineers.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Guimarães, A.A. **O Protocolo CAN Bus nas Aplicações Off-Road: Uma Análise Comparativa entre os Padrões Existentes**, SAE paper nº 2001-01-3853. São Paulo, 2001.
2. Fredriksson, L.B. **Controller Area Networks and the protocol CAN for machine control systems**, Mechatronics, v.4 n.2, p.159-92, 1994.
3. Zuberi K.M.; Shin K.G. **Real-time decentralized control with CAN**. In: IEEE Conference on Emerging Technologies and Factory Automation, 1996. Proceedings. IEEE, 1996. p.93-9.
4. Hofstee, J.W.; Goense, D. **Simulation of a Controller Area Network-based Tractor-Implement Data Bus according to ISO 11783**, J. Agric. Engng Res. (1999) 73, 383-394.
5. Sigrimis, N.; Hashimoto, Y.; Munack, A.; De Baerdemaeker, J. **Prospects in Agricultural Engineering in the Information Age**, CIGR-Ejournal, invited paper, 1998.
6. Strauss, C.; Cugnasca, C.E.; Saraiva, A.M. **Protocolos de Comunicação para Equipamentos Agrícolas**, CONAI, 1998.
7. **Controller Area Network (CAN) - Module 6.**, BOSCH - Powertrain University, 2000.
8. **ISO 11898** - Road vehicles-Interchange of digital information-Controller area network (CAN) for high-speed communication, 1993.
9. **Data Dictionary for V-Bus applications**, Issue 1.3, 2001, GM proprietary documentation.
10. **Phillips Web Site**, www.semiconductors.philips.com, consultado em Março de 2002.
11. Strauss, C. **Implementação e Avaliação de uma Rede Experimental baseada em CAN para Aplicações Agrícolas**, Dissertação de Mestrado, EPUSP, 2001.

12. **ISO11783 Workshop**, Agrithecnica 99, <http://isotc.iso.ch/livelink/livelink>, 1999.

REFERÊNCIAS ADICIONAIS

Fredriksson, L.B. **Distributed Embedded Control Systems in Robotics – Ten Years of Development**, KVASER AB, 1997.

Stone, M.L. **Dynamic Address Configuration in SAE J1939**, Biosystems and Agricultural Engineering, Oklahoma State University.

Stone, M.L. **High Speed Networking in Construction and Agricultural Equipments.**, Department of Biosystems and Agricultural Eng'g, 1994, www.agen.okstate.edu/home/mstone/.

Axelsson, J. **Serial Port Complete**, ISBN 0-9650819-2-3, Lakeview Research, 1998.

Why CAN, KVASER Web Site, www.kvaser.se/can/index.htm, 2000.

HLPs: SAE J1939, KVASER AB, 2000.

Stone, M.L.; Zachos, M. **Application of J1939 Networks in Agricultural Equipments**, Oklahoma State University Dearborn Group.

Controller Area Network – Background Information, How CAN works, Implementation, Application Layer. www.omegas.co.uk/CAN/, 2000.

Guimarães, A.A.; Saraiva, A.M. **As Aplicações Agrícolas e o Protocolo CAN: Uma Aplicação a um Monitor de Semeadora**. In: 3º Congresso da SBI-Agro, Foz do Iguaçu. Anais. SBI Agro. Lavras. 2002. p.35-42.

Stepper, M. R. **J1939 High Speed Serial Communications, The Next Generation Network for Heavy Duty Vehicles**, SAE paper nº 931809, 1993.

Alvarenga, C. **Multiplexing in Automobiles – An Application Example of the CAN Protocol**, SAE paper, Multiplexing and Networking pag. 503 – 514.

Dietmayer, K.; Overberg, K. W. **CAN Bit Timing Requirements**, SAE paper nº 970295, 1997.

Buehring, P. **CAN Controller Architecture Optimized for SAE-J1939 Applications**, SAE paper nº 940361, 1994.

Samuel, J. **Developing Diagnostics on KWP 2000 and CAN**, SAE paper nº 981112, 1998.

SOBRE O AUTOR – Alexandre de Almeida Guimarães trabalha na GM do Brasil desde 1993. Iniciou suas atividades na área de automação industrial, onde permaneceu por cerca de 4 anos. Trabalha atualmente como Engenheiro de Produto, sendo responsável por Módulos de Controle de Carroçaria, Barramentos de Dados e Sistemas de Entretenimento (Rádio, Alto-falante e Antena) em veículos de passageiros. Formou-se em Engenharia Elétrica pela Pontifícia Universidade Católica de São Paulo e cursa Mestrado em Sistemas Digitais na Escola Politécnica da Universidade de São Paulo. Sua dissertação é baseada no protocolo CAN (Controller Area Network) aplicado aos veículos agrícolas.



