

Sistemas Automotivos Embarcados

UFSC - INE5355 – Sistemas Operacionais
Prof. Dr. Antônio Augusto Fröhlich
Alunos: Diego Tondo Souza
Diogo Bratti
Novembro de 2008

Índice

Sistemas Automotivos Embarcados.....	1
Índice.....	2
CAN - Controller Area Network.....	3
Introdução.....	3
Características.....	3
As mensagens.....	4
Detecção de erros.....	4
Em nível de mensagem:.....	4
Em nível de bit:.....	4
Confiabilidade dos dados.....	5
Exemplo.....	5
Utilização.....	5
Sync – Microsoft Operational System.....	6
Toyota's Operating System.....	6
Local Interconnect Network.....	7
Media Oriented Systems Transport (MOST).....	7
FlexRay.....	8
Estrutura do FlexRay ECU.....	9
Fontes.....	11

CAN - Controller Area Network

Introdução

O CAN é um padrão de barramento projetado para microcontroladores e dispositivos se comunicarem uns com os outros.

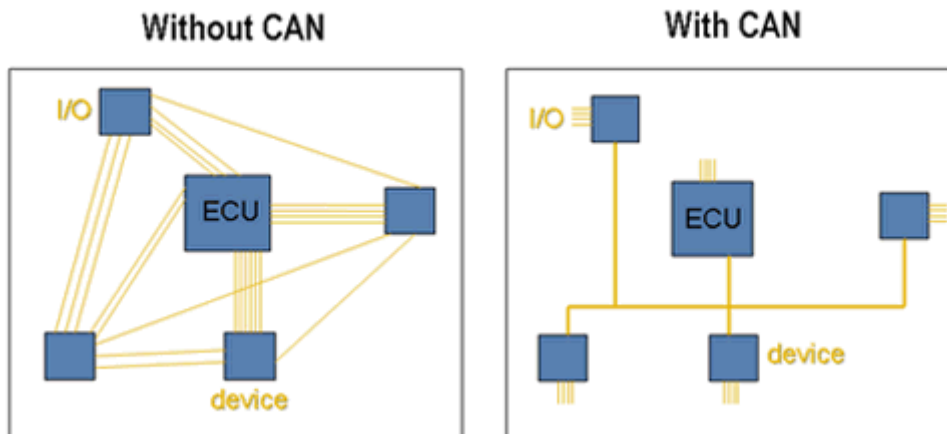
“Prevê identificadores de mensagens que facilitam o controle do fluxo de informação pela fiação. Como características de extrema relevância do CAN, podemos citar um controle de alto nível na detecção/correção de erros, grande flexibilidade na topologia e arranjo da rede intraveicular e baixa latência na comunicação entre os componentes.”

É baseado na norma ISO 11898 e ISO11519-2, desenvolvido inicialmente por Robert Bosch para utilização em redes de comunicação serial em veículos.

O Can consiste basicamente de um padrão de hardware com diferentes tipos de Frames, regras de decisão para a transmissão de mensagens e métodos para detecção e correção de erros.

Características

- Para um barramento de 50m, a velocidade máxima é de 1Mbit/s, para um barramento de 500m, 100kbit/s;
- Prioritização de mensagens definida pelo usuário, com latência máxima garantida para mensagens de maior prioridade;
- Característica de detecção e sinalização de erros construídas dentro do protocolo Can, com retransmissão automática de mensagens corrompidas;
- Requer um número bem menor de fios e conectores, diminuindo custos materiais e de instalação;
- Possibilita o compartilhamento de sensores disponíveis na rede em diferentes medidas;
- Flexibiliza o projeto, já que um novo sistema pode ser projetado apenas readaptando-se o software de controle.



Extraído de <http://zone.ni.com/devzone/cda/tut/p/id/2732>

As mensagens

O frame de mensagem pra transmissão no barramento tem 7 campos principais.

- Começa com o bit "start of frame"(SOF).
- Depois tem o "arbitration field", que tem o Identificador e o "remote transmission request"(RTR), que indica se aquele é um frame de dados ou de requisição, sem bytes de dados.
- O campo de controle tem o bit identificador de extensão (indica se o frame de mensagem é standard ou extended, porém a única diferença essencial entre esses tipos é o tamanho do Identificador, 11 ou 29 bits). Depois tem um bit reservado para futuras extensões, e mais 4 bits que indicam o número de bytes do campo de dados (0 a 8).
- Depois do campo de dados, o campo CRC que detecta erros de bit
- O campo ACK é usado pelo transmissor para receber de algum receptor um reconhecimento de um frame válido
- Fim da mensagem
- Intermission: é o número mínimo de bits separando mensagens consecutivas. Se não tem nenhum acesso seguinte ao barramento, o barramento permanece desocupado.

Detecção de erros

Em nível de mensagem:

- Cyclic Redundancy Check (CRC):

Salvaguarda a informação no frame adicionando bits de checagem no fim da transmissão. No receptor final esses bits são recomputados e testados com os bits recebidos. Gera "CRC error".

- Frame check:

Verifica a estrutura do frame transmitido checando os campos de bits com o formato fixado e o tamanho do frame. Gera "format errors"

- ACK errors:

Se nenhum acknowledgement for recebido pelo transmissor da mensagem, isso significa que houve erro de transmissão, ou que o campo ACK foi corrompido, ou que não há receptores.

Em nível de bit:

- Monitoração:

Monitora os sinais do barramento: cada nodo que transmite também observa o nível de sinal do barramento, e assim detecta diferenças entre o bit enviado e o recebido. Isso permite confiabilidade de detecção de todos os erros globais e erros locais ao transmissor.

- Bit stuffing:

Depois de ter transmitido cinco bits idênticos, um nodo vai sempre transmitir o bit oposto. Este bit extra é removido pelo receptor.

Se algum erro for descoberto por qualquer estação usando esses mecanismos, a transmissão corrente é abortada enviando um "error flag". Isto previne outras estações de aceitar a mensagem e assim assegura a consistência dos dados pela rede.

Depois que transmissão de uma mensagem com erro for abortada, o transmissor automaticamente retoma a transmissão. Haverá novamente competição pela alocação do barramento. Como regra, retransmissão começará dentro de 23 períodos de bits após a detecção de erro. Em casos especiais, o tempo de recuperação do sistema é de 31 períodos de bit. (31 micro-segundos, quando a taxa de transmissão é de 1Mbps).

Por mais efetivo e eficiente o método descrito seja, no caso de uma estação defeituosa, vai fazer com que todas as mensagens, inclusive as corretas seja abortadas, bloqueando assim o barramento se nenhuma medida de auto-monitoração for tomada.

O protocolo CAN provê, portanto um mecanismo para distinguir erros esporádicos de erros permanentes, e localizar falhas de estação. Isto é feito por avaliação estatística de situações de erro de estação, com a intenção de reconhecer os próprios defeitos da estação e possivelmente entrar num modo de operação onde o resto da rede CAN não é negativamente afetado.

A estação pode até se desligar para evitar que mensagens erroneamente reconhecidas como 'incorretas' sejam abortadas.

Confiabilidade dos dados

Exemplo

Taxa de transmissão=1Mbps

Média da utilização do barramento=50%

Total de vida de operação=4000h

Tamanho médio da mensagem=80bits

Então, total de mensagens transmitidas= 9×10^{10} .

Número estatístico de erros de transmissão não-detectados durante o tempo de vida é na ordem de menos de 10^{-2}

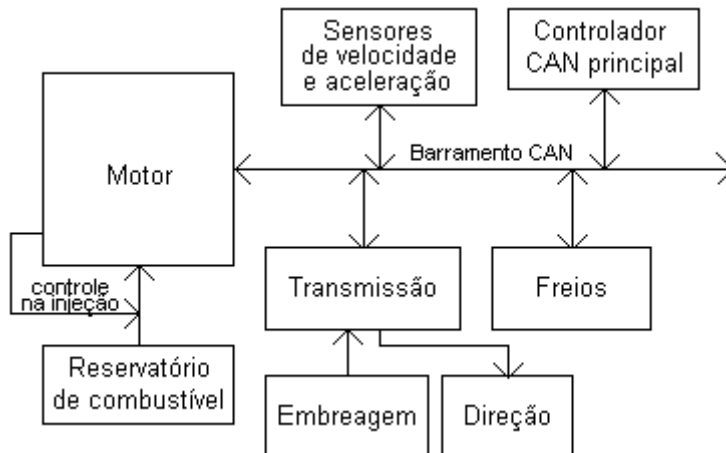
Ou seja,

Com um tempo de operação de 8h/dia, em 365 dias por ano, e uma taxa de erro de 0,7s, um erro não-detectado ocorreria a cada 100 anos.

Utilização

A figura a seguir ilustra uma típica aplicação do CAN usando um arranjo tradicional de componentes de um automóvel, incluindo o motor, a transmissão e reservatório de combustível. Tal aplicação engloba o controle de direção ou injeção de combustível. Podemos imaginar duas situações. Na primeira, o motor transmite uma

mensagem de informação de seu torque que, juntamente com informações de velocidade e aceleração, podem ser usadas pela transmissão para controle da direção. Na segunda, quando o motorista muda de marcha, a transmissão envia uma mensagem para que o motor possa regular a injeção de combustível.



Aplicação típica do protocolo CAN em uma rede embutida num automóvel.

Extraído de <http://www2.eletronica.org/artigos/eletronica-digital/introducao-as-redes-intraveiculares/>

Sync – Microsoft Operational System

Criado para facilitar a interação entre humano e acessórios do carro, especialmente dispositivos de áudio e telefone, ele pode ser utilizado por comandos de voz ou por um menu em um touchscreen. Concebido a partir da parceria entre Ford, Lincoln e Mercury, é utilizado também pela FIAT em uma variação do sistema original.

Pouca informação técnica está disponível, e no próprio no site da Microsoft não é fácil encontrar informações, mesmo não-técnicas do produto.

Apesar de suas limitações, é possível enviar mensagens, fazer ligações telefônicas, acessar dispositivos através da interface USB ou através de bluetooth e fazer chamadas de emergência e de socorro automotivo.

Site oficial : <http://www.syncmyride.com/>

Toyota's Operating System

Toyota prepara-se para lançar um OS, previsão de 2015, com expectativa de haver piloto automático, entretanto não há nada de concreto até então.

Local Interconnect Network

O LIN-Bus (Local Interconnect Network) é um padrão de comunicação veicular utilizada em arquitetura de redes inseridas em carros. A especificação do LIN é mantida pelo LIN-consortium, o qual teve sua versão 1.1, lançada em 1999.

O LIN bus é um sistema de rede pequeno e lento utilizado como uma opção barata para subredes CAN, utilizada para integrar sensores inteligentes ou atuadores em carros. Recentemente o LIN pode ser utilizado sobre a linha de força da bateria veicular com um transmissor especial, DC-LIN.

O protocolo LIN é composto por LIN-Mestre e LIN-Escravo. O LIN-Mestre utiliza um tabela de horários pré-definidos para enviar e receber dados no barramento LIN. As tabelas contém pelo menos o tempo relativo e onde a mensagem é iniciada. Um LIN Frame consiste de duas partes um cabeçalho e uma resposta. O cabeçalho sempre é enviado pelo mestre e a resposta sempre é enviada pelo escravo.

Os dados são transmitidos através do LIN serialmente, com oito bits de dados e com um bit de início e de parada, porém sem bit de paridade. A taxa de bit varia na faixa de 1kbaud até 20kbaud. Os dados no barramento são divididos em recessivos (logicamente altos) e dominantes (logicamente baixos). O tempo normal é considerado pela fonte estável do clock do LIN master, a menor entidade possível é um Bit Time (52 μ sec @ 19.2 kbaud).

O barramento pode atingir dois estados — Sleep-mode e ativo — utilizados pelo protocolo LIN. Enquanto o dado está no barramento todos os nodos LIN são requisitados para estarem no estado ativo. Depois de um timeout especificado, o nodo entra no Sleep-mode e será colocado em ativo quando receber um WAKEUP frame. Este frame pode ser enviado por qualquer nodo requisitando a utilização do barramento, o que pode acontecer tanto se o LIN Mestre seguir a programação interna, quanto se um dos LIN Escravos forem ativados por seu software interno. Depois de todos os nodos serem “acordados”, o Mestre continua enviando as mensagens programadas.

A LIN API (Application Programmers Interface) provê um conjunto de chamadas de funções (que tem base na linguagem C) que devem ser implementada por cada LIN driver. Utilizando as rotinas pré-definidas do driver, todos as funções LIN podem ser acessadas.

A utilização da API facilita a implementação de padrões na criação dos drivers. E também aumenta a velocidade dos testes.

Media Oriented Systems Transport (MOST)

Media Oriented Systems Transport (MOST) é um padrão de rede veicular com a intenção de interligar componentes multimídias em carros e outros veículos. Ele difere das outras tecnologias por poder transmitir uma quantidade imensa de dados, por fibra ótica, produzindo assim uma rede com uma transmissão de longe mais rápida se comparada com outras tecnologias atualmente existentes no setor.

A especificação do MOST define todas as sete camadas do Modelo de Referência de comunicação de dados ISO/OSI. As redes MOST podem suportar até sessenta e quatro dispositivos ou nodos. É possível utilizar um serviço plug and play que permite adicionar e remover dispositivos de maneira muito fácil e simples. Existe um temporizador mestre, que é um dos nodos, que constantemente alimenta o frame de dados. O preâmbulo ou cabeçalho do pacote sincroniza os outros nodos chamados de temporizadores escravos. A bandwidth total (incluindo streaming de dados e pacote de dados) é aproximadamente 23M Baud. Com sessenta canais e quinze canais MPEG1 disponíveis para configuração do usuário.

MOST é desenvolvido pela MOST Cooperation, um corpo de indústrias que fabricam automóveis (incluindo FORD, BMW, DaimlerChrysler, and General Motors), empresas que fornecem equipamento elétrico para automóveis (incluindo Infineon Technologies, Delphi E&S (former Delco), Denso (a division of Toyota), Bosch, and Hamamatsu), e empresas de audio-video (incluindo Sony, Philips, Linn Products, and Motorola). MOST é registrada pela SMSC. A MOST Cooperation tem autorização de utilizá-la sem cobrança de taxas.

FlexRay

FlexRay é um sistema de comunicação desenvolvido para aplicações de alto desempenho, com dois canais de transmissão para haver redundância e garantir robustez e segurança para eventuais problemas, como por exemplo um fio cortado. Com 10Mbps de throughput por canal pode ser utilizado em carros trabalhando em conjunto com sistemas CAN e LIN. FlexRay reduziria o preço pelo fato de diminuir o número de redes paralelas, o que diminuiria os gargalos de transmissão.

Os grandes benefícios do FlexRay são:

- arquitetura simplifica de rede para carros;
- desenvolvimento de controle inteligente;
- reduz a demanda de fios necessários;
- reduz a largura dos subsistemas de rede;
- computação distribuída através de um relógio global;
- sistemas eletromecânicos (X-by-wire) substituindo componentes hidráulicos.

Os participantes do consórcio que mantém o FlexRay compõem um núcleo, composto atualmente por sete integrantes: BMW, Bosch, Daimler, Freescale Semiconductor, General Motors, NXP Semiconductors, Volkswagen. Além disto existe duas classes de associados: associados premium e associados. Dentre os associados premium estão: austriamicrosystems AG, c&s group, Continental AG Hannover, Delphi Corporation, Denso, EADS, Elektrobit Automotive GmbH, Elmos Semiconductor AG, Fiat, Ford Motor Company, HONDA Motor Co., Infineon Technologies AG, Mentor Graphics, National Instruments, NEC Electronics Europe, NISSAN MOTOR CO., Ltd., PSA Peugeot Citroën, Renault, Renesas Technology Corp., Toyota Motor Corporation,

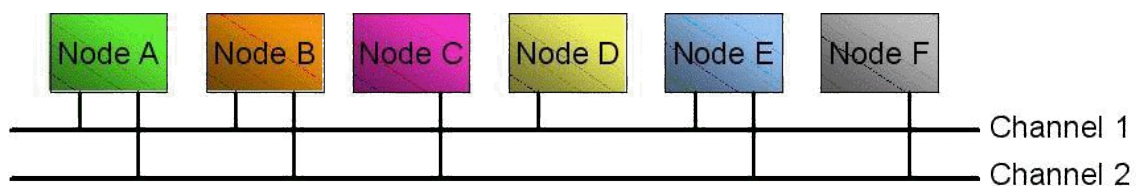
TTTech Automotive GmbH, TÜV NORD Mobilität GmbH & Co. KG, IFM, Tyco Electronics Corporation, TZM, Vector-Informatik.

O sistema FlexRay consiste de um barramento e processadores (electronic control units, or ECUs). Onde cada ECU tem um clock independente. A diferença entre o clock local e o clock de referencia não pode ser maior que 0,15%, com isto a diferença entre o dispositivo mais rápido e o mais lento do sistema pode ter uma diferença de apenas 0,3% no clock.

Isto significa que, se existe um ECU-s, transmissor, e um ECU-r, receptor, para cada trezentos ciclos do transmissor o receptor estará entre 299 e 301 ciclos. O clock é ressinchronizado frequentemente para assegurar que não hajam problemas.

FlexRay trabalha com o mesmo princípio do TDMA, onde os componentes ou mensagens tem um pedaço de tempo fixo alocado no qual eles tem acesso exclusivo ao barramento. Os pedaços de tempo são repetidos em um ciclo fixo. O tempo que cada mensagem está no barramento pode ser previamente descoberto e é determinístico.

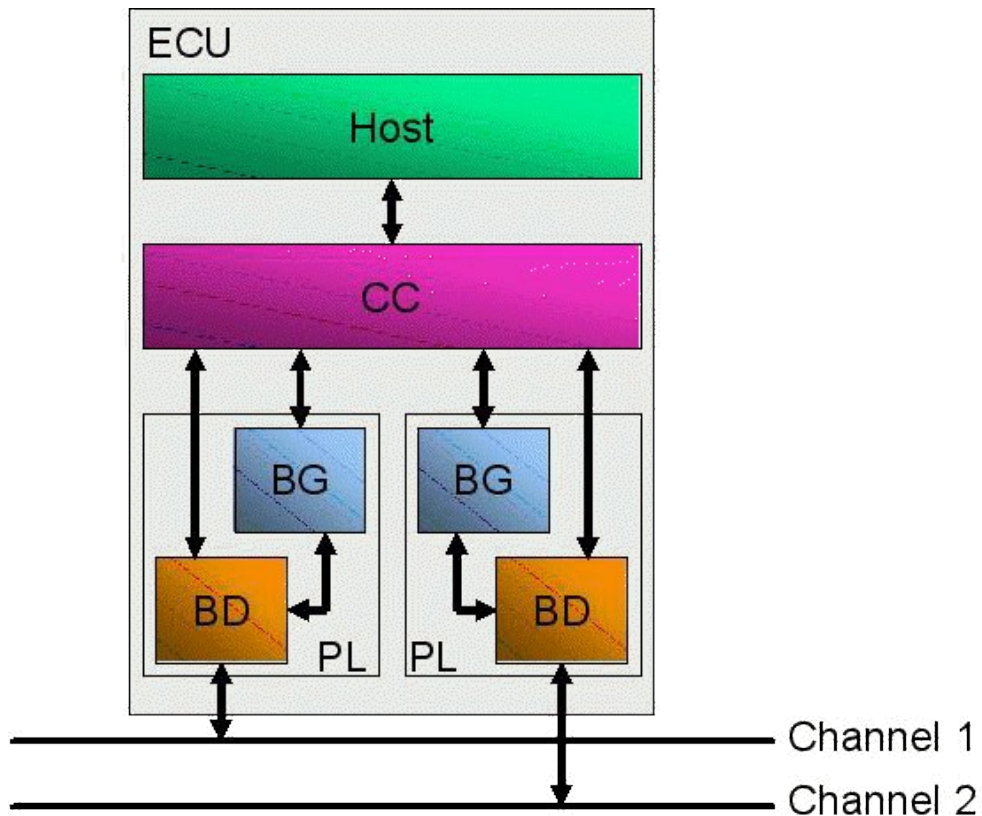
A figura a seguir demonstra a topologia do FlexRay, o qual possui dois canais de comunicação para haver redundância.



Estrutura do FlexRay ECU

O FlexRay ECU consiste de um processador hospedeiro (host processor), o FlexRay Communication Controller (CC) e o Bus Guardian (BG). O processador hospedeiro supre e processa os dados, os quais são transmitidos pelo CC.

O BG monitora o acesso ao barramento. O processador hospedeiro informa ao BG qual pedaço de tempo o FlexRay CC tem alocado. O BG então permite o FlexRay CC transmitir dados somente nestes pedaços de tempo e habilita o BD (Bus Driver). Dados podem ser recebidos a qualquer momento.



Apenas uma ECU escreve no barramento por vez. Cada bit é mantido no barramento até que pelo menos oito amostras (ciclos) do sinal. O receptor mantém um buffer das últimas cinco amostras e considera válida a amostra que mais se repetiu nas últimas cinco amostras.

Este tipo de procedimento pode afetar os bits que estão na fronteira das oito amostras mas não influenciarão na região central dos oito ciclos.

Fontes

http://en.wikipedia.org/wiki/Local_Interconnect_Network

http://en.wikipedia.org/wiki/Media_Oriented_Systems_Transport

<http://en.wikipedia.org/wiki/FlexRay>

http://www.tzm.de/en/FlexRay/FlexRay_Introduction.html

<http://www.flexray.com/>

<http://zone.ni.com/devzone/cda/tut/p/id/2732>

<http://www2.eletronica.org/artigos/eletronica-digital/introducao-as-redes-intraveiculares/>

http://www.gaii.unifei.edu.br/can/hp_can.htm