

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA



**PROTOCOLO DE COMUNICAÇÃO CAN E SUAS
APLICAÇÕES NA INDÚSTRIA AUTOMOBILÍSTICA.**

Área de Eletrônica da Universidade São Francisco de Itatiba - SP

por

Lucas de Camargo Nascimento

Prof. André Luís Faustino da Silva
Orientador

Itatiba (SP), Junho de 2006

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA



**PROTOCOLO DE COMUNICAÇÃO CAN E SUAS
APLICAÇÕES NA INDÚSTRIA AUTOMOBILÍSTICA.**

Área de Eletrônica da Universidade São Francisco de Itatiba - SP

por

Lucas de Camargo Nascimento

Relatório apresentado à Banca Examinadora do
Trabalho de Conclusão do Curso de Engenharia
Elétrica para análise e aprovação.
Orientador: Prof. André Luís Faustino da Silva

Itatiba (SP), Junho de 2006

SUMÁRIO

LISTA DE ABREVIATURAS	iii
LISTA DE FIGURAS	iv
LISTA DE TABELAS.....	v
LISTA DE EQUAÇÕES	vi
RESUMO	vii
ABSTRACT	viii
1. INTRODUÇÃO.....	1
1.1. OBJETIVOS	2
1.1.1. Objetivo Geral	2
1.1.2. Objetivos Específicos.....	2
1.2. METODOLOGIA.....	2
1.3. ESTRUTURA DO TRABALHO	2
2. CONTROLLER AREA NETWORK (CAN)	4
2.1. CARACTERÍSTICAS GERAIS	4
2.2. ARQUITETURA DO BARRAMENTO	5
2.3. ENDEREÇAMENTO E CARACTERÍSTICAS DO BARRAMENTO	7
2.4. ESTRUTURA DO BARRAMENTO E MEIOS DE TRANSMISSÃO.....	8
2.5. APLICAÇÕES DO BARRAMENTO	10
2.5.1. Can High Speed	11
2.5.2. Can Low Speed.....	13
2.6. IMPLEMENTAÇÕES CAN.....	13
2.7. QUADROS DE MENSAGEM.....	15
2.7.1. Quadro de Dados.....	15
2.7.2. Quadro Remoto	18
2.7.3. Quadro de Erro	19
2.7.4. Quadro de Sobrecarga	20
2.7.5. Espaço entre Quadros.....	21
2.8. RECONHECIMENTO E MANIPULAÇÃO DE ERROS.....	22
2.8.1. Verificação de Redundância Cíclica (CRC - Cyclic Redundancy Check)	22
2.8.2. Verificação de reconhecimento	23
2.8.3. Verificação de Quadro	24
2.8.4. Monitoramento do Barramento.....	24
2.8.5. Bit Stuffing.....	25
2.8.6. Estados de Erro	25
2.9. CONCEITOS NRZ E BIT STUFFING	27
2.10. SINCRONIZAÇÃO DO BARRAMENTO.....	28
3. ESTUDO COMPARATIVO	29
3.1. BARRAMENTOS UTILIZADOS.....	29
3.1.1. RS-232	30
3.1.2. RS-485	31
3.2. COMPARATIVO	31
4. FUNCIONAMENTO TÍPICO DE UMA APLICAÇÃO AUTOMOBILÍSTICA COM PROTOCOLO CAN	37
5. CONSIDERAÇÕES FINAIS.....	41
REFERÊNCIAS BIBLIOGRÁFICAS	42

LISTA DE ABREVIATURAS

TCC	Trabalho de Conclusão de Curso
USF	Universidade São Francisco
CAN	Controller Area Network
ISO	International Standard Organization
SAE	Society of Automotive Engineers
OSI	Open System Interconnection
NRZ	Non Return to Zero
CSMA/CD	Carrier Sense Multiple Acces with Collision Detection
RTR	Remote Transmission Request
CRC	Cyclic Redundancy Check

LISTA DE FIGURAS

Figura 1. Barramento serial CAN.[8].....	5
Figura 2. Camadas do modelo OSI.[4].....	5
Figura 3. Endereçamento de mensagens. [8].....	7
Figura 4. Estados do Barramento [8]	8
Figura 5. Meios de transmissão. [8]	9
Figura 6. Relação Taxa de transmissão / distância. [10].....	9
Figura 7. Classes de Aplicações do Barramento CAN. [8].....	11
Figura 8. Padrão CAN High Speed. [8]	12
Figura 9. Nível de tensão no barramento. [8].....	12
Figura 10. CAN padrão e CAN estendido. [8].....	14
Figura 11. Diferenças entre versões do Protocolo CAN. [8]	15
Figura 12. Quadros de dados existentes no Protocolo CAN. [8]	15
Figura 13. Quadro de dados Padrão. [8].....	17
Figura 14. Quadro de dados Extendido. [8]	18
Figura 15. Quadro de dados Extendido. [8]	19
Figura 16. Quadro de Erro. [8].....	20
Figura 17. Quadro de Sobrecarga. [8].....	21
Figura 18. Quadro de Espaço entre quadros. [8].....	22
Figura 19. Mecanismos de manipulação de erros no barramento. [8]	22
Figura 20.CRC error. [8].....	23
Figura 21.Cheque de reconhecimento. [8]	23
Figura 22.Verificação de Quadro. [8]	24
Figura 23.Monitoramento do Barramento. [8]	25
Figura 24.Estados de Erro. [8]	26
Figura 25. Conceito NRZ. [8]	27
Figura 26. Conceito Bit Stuffing. [8]	28
Figura 27. Exemplo de barramento em um automóvel. [11]	32
Figura 28. (a) Painel de Instrumentos; (b) Central de controle do Motor; (c) Simulador para o Painel de Instrumentos.	38
Figura 29. Janela principal do software Canalyzer.	39
Figura 30. Janela do Interactive Generator Block do software Canalyzer.....	40

LISTA DE TABELAS

Tabela 1. Comparativo entre barramentos aplicados à classe A. [11]	33
Tabela 2. Comparativo entre barramentos aplicados à classe B. [11].....	35
Tabela 3. Comparativo entre barramentos aplicados à classe C. [11].....	36

LISTA DE EQUAÇÕES

RESUMO

NASCIMENTO, Lucas C. **Protocolo de comunicação CAN e suas aplicações na indústria automobilística**. Itatiba, 2006. 51f. Trabalho de Conclusão de Curso, Universidade São Francisco, Itatiba, 2006.

A utilização de redes de comunicação para sistemas embarcados deixou de ser uma ferramenta opcional para tornar-se uma necessidade.

CAN, Controller Area Network, é uma especificação de interconexão e protocolo para comunicação que está sendo utilizada em larga escala pela indústria automobilística. Devido a queda no custos dos circuitos eletrônicos e aprimoramento nos conhecimentos das aplicações CAN, tivemos um crescimento significativo de 20 milhões de nós CAN em uso mundial, no ano de 1997, para cerca de 120 milhões no ano 2000. O protocolo CAN é, certamente, o principal protocolo de barramento em veículos na Europa. [8]

Neste trabalho serão abordados as principais características, princípios de funcionamento, os aspectos elétricos, funcionais e quais as vantagens por se optar pelo protocolo CAN em sistemas embarcados.

Palavras-chave: CAN 1. Protocolo CAN 2. Barramento 3.

ABSTRACT

The utilization of network communication in embedded systems is not more an optional tool but a real necessity.

CAN, controller Area Network, is a specification of interconnection and protocol to communication that is being used in large scale by automotive industry. Due the fall in the costs of electronics circuits and improvement in the knowledge of CAN applications. In 1997 there were a significant increasement from 20 millions of CAN nodes around the world, to about 120 millions in 2000. The CAN protocol is, certainty, the main protocol of bus in vehicles in Europe.

In this work, it will be showed the main characteristics, functionality concepts, the electrical aspects, functional and which are the advantages to use CAN protocol in embedded systems.

Keywords: *CAN 1. CAN Protocol 2. Bus 3.*

1. INTRODUÇÃO

O desenvolvimento do CAN começou quando mais e mais dispositivos eletrônicos foram implementados nos modernos veículos. Exemplos de tais dispositivos incluem sistemas de gerenciamento do motor, suspensão ativa, ABS, controle de câmbio, controle de luzes, ar condicionando, airbag e central de trava elétrica. Tudo isso significa mais segurança e mais conforto para o motorista e, é claro, uma redução de consumo de combustível, gases de emissão e custos para a montadora do automóvel.

Melhorar as características do veículo era necessário para que os diferentes sistemas de controle e seus sensores pudessem trocar informações. Isto era normalmente feito por conexão discreta entre os diferentes sistemas, ou seja, ligação ponto a ponto.

A exigência para a troca de informações cresceu de tal forma que foram necessários cabos com comprimentos de até vários quilômetros e muitos conectores. Isto acarretou problemas crescentes relativos à custo de material, tempo de produção e confiabilidade.

A solução para este problema foi a conexão dos sistemas de controle por um barramento serial. Este barramento teve que cumprir algumas exigências especiais devido ao seu uso em um veículo. Para cumprir tais exigências criou-se o protocolo CAN.

O CAN ou Controller Area Network é um sistema de barramento avançado que suporta de forma eficaz sistemas de gerenciamento distribuído. Foi desenvolvido inicialmente para o uso em veículos automotivos pela Robert Bosch GmbH, Alemanha, no fim dos anos 80. Logo em seguida foi padronizado internacionalmente pela ISO (International Organization for Standardization) e pela SAE (Society of Automotive Engineers).

1.1. OBJETIVOS

1.1.1. Objetivo Geral

Apresentar uma tecnologia de comunicação serial aplicada em larga escala na indústria automobilística.

1.1.2. Objetivos Específicos

Realizar um estudo do protocolo CAN referente às suas características elétricas e funcionais comparando com as tecnologias existentes no mercado, visando atender os seguintes objetivos

- Detalhar tecnicamente o barramento CAN em relação aos aspectos de endereçamento, estrutura, aplicação e implementação;
- Comparar os atuais protocolos utilizados na indústria automobilística;
- Demonstrar o funcionamento de uma aplicação automobilística com Protocolo CAN através da utilização de simuladores e módulos comerciais;

1.2. METODOLOGIA

O trabalho foi realizado através de estudo e pesquisas em normas específicas, pesquisas webgráficas e também consultas a profissionais da área.

A demonstração do funcionamento será realizada através de recursos utilizados na indústria automobilística atualmente, como módulos eletrônicos do veículo com interface CAN, simuladores e software de gerenciamento do barramento CAN (CANalyzer).

1.3. ESTRUTURA DO TRABALHO

No capítulo 1 faz-se uma introdução ao trabalho, apresentação dos objetivos e a metodologia utilizada.

No capítulo 2 faz-se um estudo técnico do protocolo de comunicação CAN, abrangendo suas características elétricas e funcionais.

No capítulo 3 faz-se uma descrição comparativa com os outros protocolos de comunicação utilizados na indústria automobilística.

No capítulo 4 faz-se a demonstração do funcionamento de uma aplicação automobilística com protocolo CAN.

E pôr fim, no capítulo 5, faz-se a conclusão e considerações finais.

2. CONTROLLER AREA NETWORK (CAN)

O protocolo CAN começou a ser desenvolvido em 1983 pela empresa BOSCH na Alemanha, e foi anunciado oficialmente em 1986, sendo aplicado em unidades de controle eletrônico nos automóveis produzidos pela Mercedes.

Em 1987 surgiram os primeiros circuitos integrados para CAN, fabricados pela Intel e pela Philips.

2.1. CARACTERÍSTICAS GERAIS

CAN é um barramento serial utilizado para interligar dispositivos em rede. Hoje, apesar de também ser utilizado em aplicações industriais, este barramento é aplicado principalmente em sistemas embarcados por possuir características desejadas na área da comunicação de dados, como tolerância a interferência eletromagnética, prioridade de mensagens, recuperação de falhas, entre outras.

Uma rede CAN pode interligar até 2032 dispositivos, sendo que o limite prático é de aproximadamente 110 dispositivos, sendo cada dispositivo tratado como um nó da rede. No nível físico, o link serial mais usado é composto de dois fios, o sinal tem característica diferencial, é capaz de operar até 1 Mbps, tendo restrições de velocidade em virtude da distância entre os nós. Para uma rede com extensão de 1 km, a velocidade por ser reduzida até 50Kbps. Cada nó ligado a este link serial é capaz de “ouvir”, simultaneamente a outros nós, os dados transmitidos na rede. A escrita, porém, é uma operação permitida somente para um dispositivo por vez.

O protocolo CAN 2.0A tornou-se o padrão ISO 11898-1 em 1993. A última versão do protocolo é a 2.0B. A maior diferença entre as duas versões é a quantidade de bits no identificador, sendo 11 bits para o 2.0A e 29 bits para o 2.0B. Os padrões ISO para CAN, ou com origem em redes do deste tipo, são os seguintes: ISO 11898-2 (alta velocidade), ISO 11898-3 (tolerante a falhas / baixa velocidade).

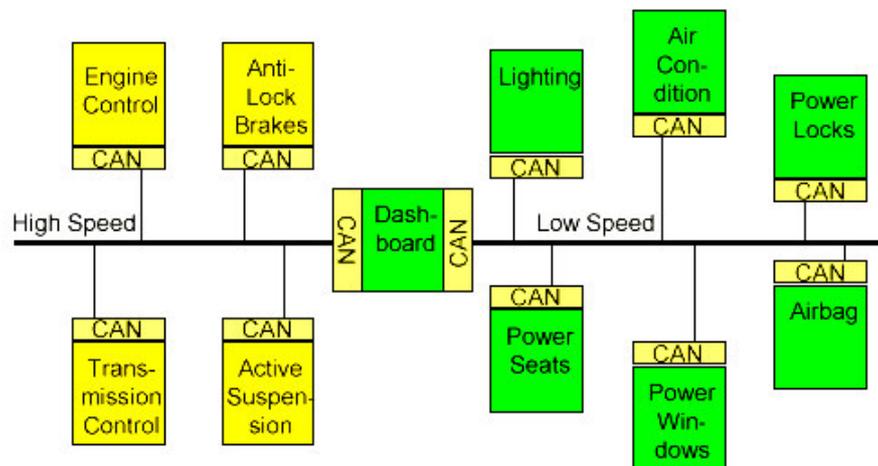


Figura 1. Barramento serial CAN.[8]

2.2. ARQUITETURA DO BARRAMENTO

Tendo como referência o modelo ISO-OSI, o protocolo CAN utiliza duas camadas, a Camada Física e a Camada de Enlace. A figura 2 apresenta todas as camadas do modelo ISO-OSI e destaca as camadas utilizadas pelo protocolo CAN.

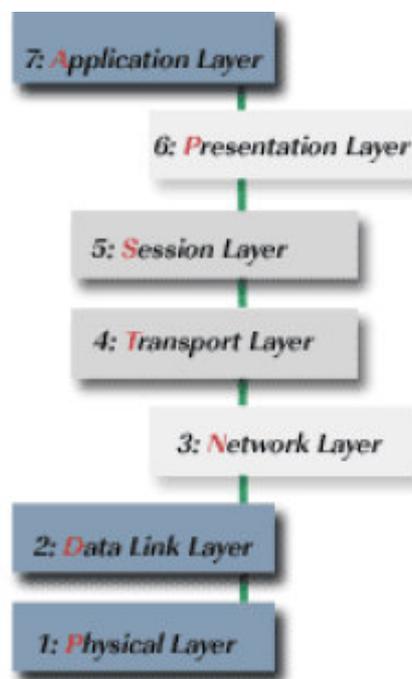


Figura 2. Camadas do modelo OSI.[4]

- **Camada Física:**

Realiza a codificação, a decodificação, a temporização e a sincronização do sinal. O padrão de codificação de linha utilizada é o NRZ (Non Return to Zero) para que o valor médio de ocorrência de bits recessivos (nível lógico 1) e dominantes (nível lógico 0) seja equilibrado. As características desta camada não são definidas pela especificação da BOSCH [3], porém, existe uma norma ISO que define as características do modelo OSI.

A camada física também é responsável pelo confinamento de falhas (juntamente com a camada de enlace) e tratamento de falhas provenientes do barramento.

- **Camada de Enlace:**

A Camada de Enlace divide-se em duas subcamadas, sendo elas:

- LLC: responsável pela filtragem de mensagens, notificação de sobrecarga e controle de recuperação.

- MAC: encapsula e desencapsula dados, realiza a codificação dos quadros (Bit Stuffing: caso aconteça 5 bits consecutivos apresentando o mesmo nível, insere um bit com valor inverso), controle de acesso ao meio, detecção e sinalização de erros.

Estas duas subcamadas são responsáveis ainda pelo confinamento de falhas, ou seja, um nó que estiver com muitos erros de transmissão ou recepção poderá ser automaticamente desligado da rede. O controlador CAN é responsável por lidar automaticamente com estes serviços de forma transparente ao software.

2.3. ENDEREÇAMENTO E CARACTERÍSTICAS DO BARRAMENTO

É um barramento multi-master, ou seja, com uma estrutura aberta, linear e uma linha de barramento lógica com nós de direitos iguais. O número de nós não está limitado pelo protocolo, podendo ser mudado dinamicamente (inserção ou retirada) sem perturbar a comunicação dos outros nós.

No protocolo CAN, os nós do barramento não têm um endereço específico. Ao invés disso, a informação de endereço é contida no identificador das mensagens transmitidas que indica o conteúdo da mensagem e a prioridade da mesma (quanto menor for o valor do identificador maior será a prioridade). O identificador informa qual é a mensagem (temperatura do motor, velocidade, rotação, etc.). Desta forma, uma mensagem não é enviada por um transmissor somente a um receptor, mas sim, cada unidade que se mostrar interessada pela mensagem pode recebê-la (processo Multicasting e Broadcasting).

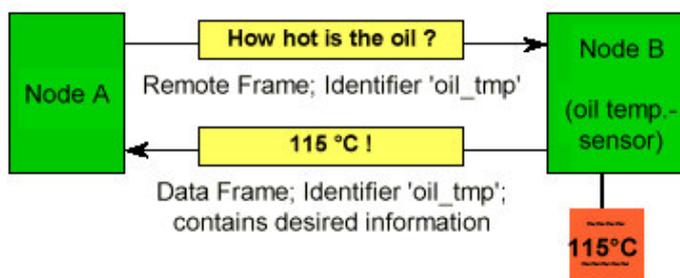


Figura 3. Endereçamento de mensagens. [8]

O protocolo CAN provê um sofisticado mecanismo de detecção e manipulação de erro, como CRC, além de alta imunidade contra interferência eletromagnética. Mensagens errôneas são automaticamente retransmitidas, erros temporários são recuperados e erros permanentes são seguidos por sistema automático de desligamento de nós defeituosos. Com isso, tem-se uma consistência de dados garantida em todo o sistema.

2.4. ESTRUTURA DO BARRAMENTO E MEIOS DE TRANSMISSÃO

O acesso ao barramento é manipulado através do avançado protocolo de comunicação serial Carrier Sense Multiple Access / Collision Detection (CSMA/CD). Isto significa que a colisão de mensagens é evitada através de arbitragem bit a bit, sem perda de tempo.

Há dois estados do barramento: "dominante" e "recessivo". A lógica do barramento usa o mecanismo "Wired-AND", que quer dizer, os "bits dominantes" (equivalente ao nível lógico "Zero") sobrepõem os "bits recessivos" (equivalente ao nível lógico "Um").

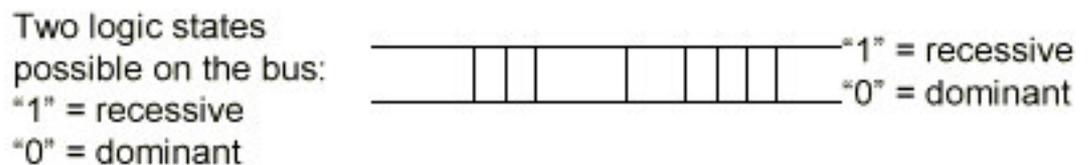


Figura 4. Estados do Barramento [8]

Só se todos os nós transmitirem bits recessivos ("Um"), o barramento estará no estado recessivo, assim que um nó transmite um bit dominante ("Zero"), o barramento passa ao estado dominante.

Assim sendo, para a linha de barramento do CAN, deve ser escolhido um meio físico que possa transmitir o dois possíveis estados da linha: "dominante" e "recessivo". Um dos modos mais comuns e mais baratos é usar um par de fios trançado.

Também um meio óptico poderia ser usado. Neste caso, o estado recessivo seria representado pela "ausência de luz" e o estado dominante pela "presença de luz".

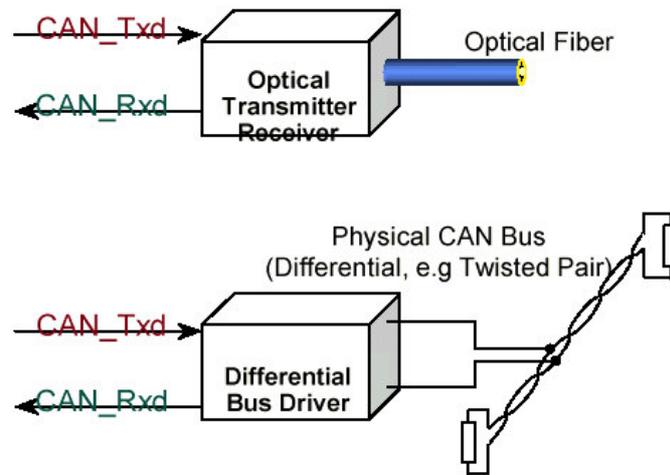


Figura 5. Meios de transmissão. [8]

Com o par de fios trançado, a taxa de transmissão máxima no barramento é 1 Mbaud para um cabo de até 40 metros. Para comprimentos maiores que 40 metros a taxa de transmissão deve ser reduzida. Para um barramento de 1000 metros a taxa de transmissão ainda pode chegar a 50 Kbaud. Comprimentos maiores que 1000 metros devem usar drivers especiais. A figura 6 mostra a relação entre a taxa de transmissão e a distância ente os nós.

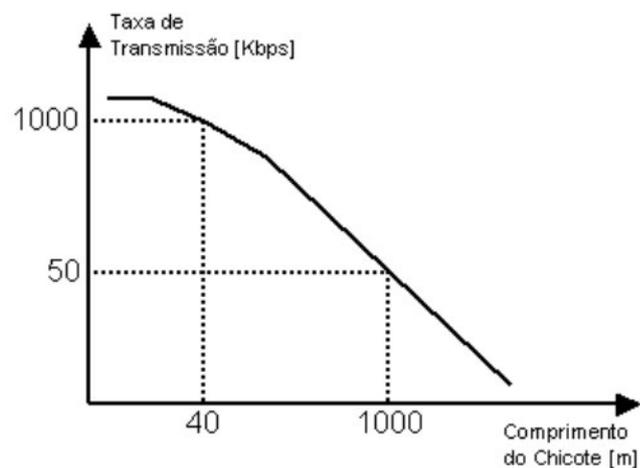


Figura 6. Relação Taxa de transmissão / distância. [10]

As linhas do barramento são chamadas "CAN_H" e "CAN_L". As duas linhas do barramento CAN_H e CAN_L são controladas pelos nós com um sinal diferencial. O par de fios trançado é terminado com um resistor tipicamente de 120 Ohms em cada ponta da linha. Devido à transmissão operar de forma diferencial, o CAN é resistente à interferência eletromagnética, porque ambas as linhas do barramento são afetadas da mesma maneira, o que deixa o sinal diferencial inalterado. Para reduzir ainda mais a sensibilidade contra a interferência eletromagnética, as linhas podem ser protegidas adicionalmente por uma malha blindada. Isto também reduz a irradiação eletromagnética do próprio barramento, especialmente em altas taxas de transmissão.

Se o nó A quer enviar uma mensagem a um ou mais nós, ele deve testar antes de tudo se o barramento está livre, ou seja, no estado inativo (recessivo). É o que chamamos de Carrier Sense. Neste caso o nó A torna-se o Master do barramento, todos os outros nós vão automaticamente para o estado de recepção e o nó A envia sua mensagem. Após a recepção correta da mensagem, todas as estações testam (via identificador da mensagem) se os dados recebidos tem significado para elas ou não. Em caso afirmativo os dados são aproveitados, senão são desprezados.

2.5. APLICAÇÕES DO BARRAMENTO

Aplicações do barramento em veículos podem ser separadas em três categorias diferentes, de acordo com as suas capacidades de tempo real:

- Classe A: Para um barramento de velocidade baixa, com taxas de até 10 Kbps, como por exemplo, aplicações de gerenciamento de sistemas no chassi;
- Classe B: Para um barramento de velocidade intermediária, com taxas de 10 Kbps a 125 Kbps, como por exemplo, painel e diagnósticos;
- Classe C: Para um barramento de velocidade alta, com taxas de 125 kbps a 1 Mbps para aplicações de tempo real, como gerenciamento do motor, caixa de câmbio, ABS, etc.

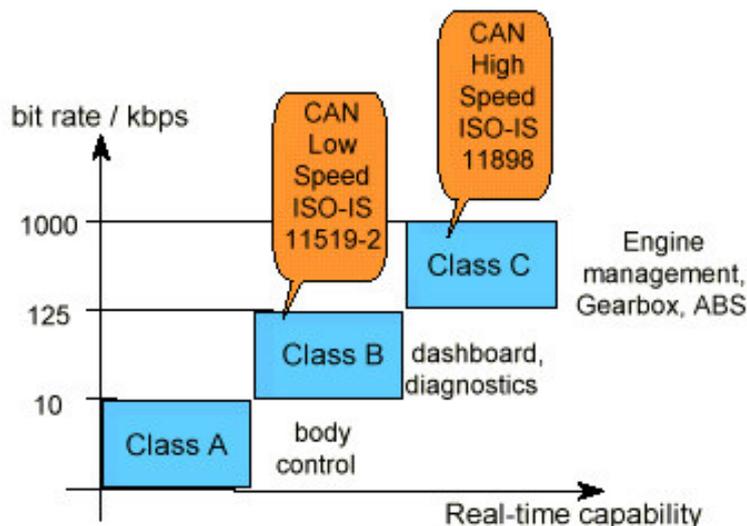


Figura 7. Classes de Aplicações do Barramento CAN. [8]

Para o uso do CAN em veículos, foram definidos dois padrões para a interface do barramento: CAN High Speed (Alta Velocidade) de acordo com a ISO-IS 11898 com taxas entre 125 kbps e 1 Mbps; e CAN Low Speed (Baixa Velocidade) de acordo com a ISO-IS 11519-2 com taxas de até 125 kbps.

2.5.1. Can High Speed

As linhas do barramento podem ter até 40 m de comprimento à taxa máxima de 1 Mbps e é terminado por um resistor de 120 Ohms. Podem ser mais longas se diminuída a taxa de transferência. Até 30 nós podem ser conectados com drivers de CAN. Para a conexão de mais nós, drivers mais fortes ou repetidores devem ser usados. Para evitar reflexões, a conexão das linhas do barramento com os nós não deve exceder 0.3 m a 1 Mbps.

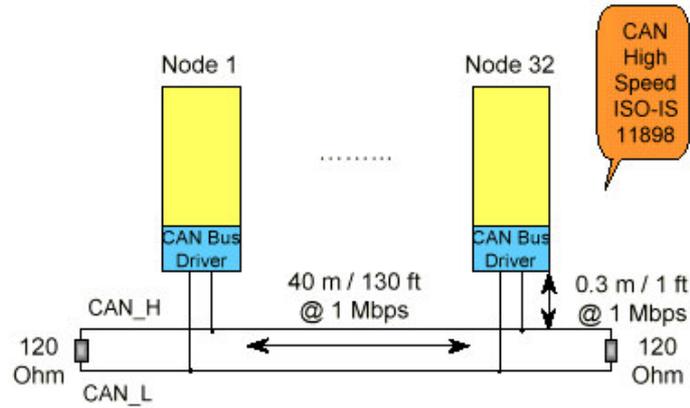


Figura 8. Padrão CAN High Speed. [8]

Um bit recessivo é representado através das duas linhas do barramento com um nível de tensão de aproximadamente 2.5 V, de modo que a diferença de potencial entre CAN_H e CAN_L seja de 0 V.

Um bit dominante é representado colocando-se CAN_H em 3.5 V e CAN_L em 1.5 V. Isto resulta uma diferença de potencial para um bit dominante de cerca de 2.0 V.

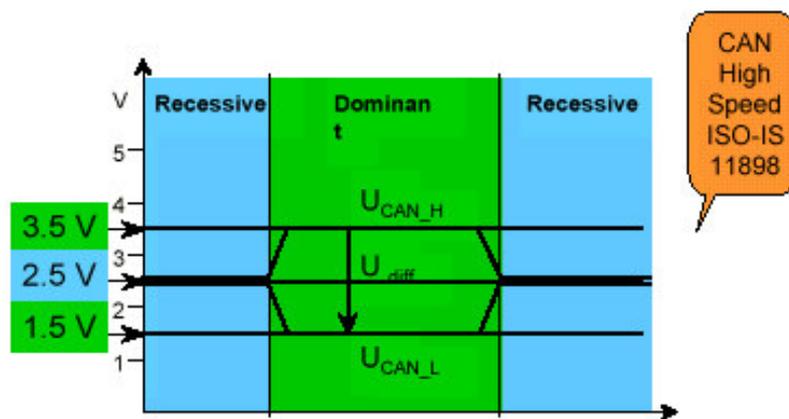


Figura 9. Nível de tensão no barramento. [8]

2.5.2. Can Low Speed

As linhas do barramento podem atingir uma velocidade máxima de 150 Kbps. Esse limite de taxa de transmissão deixa o sistema mais imune à interferência, não sendo mais necessário o resistor de terminação, e até 20 nós podem ser conectados.

Um bit recessivo é representado colocando-se CAN_H em 1.75 V e CAN_L em 3.25 V. Isto resulta uma diferença de potencial para um bit dominante de cerca de 1.5 V.

Um bit dominante é representado colocando-se CAN_H em 4.0 V e CAN_L em 1.0 V. Isto resulta uma diferença de potencial para um bit dominante de cerca de 3.0 V.

2.6. IMPLEMENTAÇÕES CAN

As especificações originais do CAN (Versões 1.0, 1.2 e 2.0A) especificam um identificador de mensagem de 11 bits. Isto é conhecido como "CAN Padrão". Aqueles Quadros de Dados e Quadros Remotos que contêm um identificador de 11 bits são por este motivo chamados de Quadros Padrões. Com estes quadros, 2048 mensagens diferentes podem ser identificadas. A especificação V2.0A foi atualizada (versão 2.0B) para remover esta possível limitação de número de mensagem e, com isto, atender ao padrão SAE J1939 para o uso do CAN em caminhões. A versão 2.0B do CAN é chamada "CAN Estendido".

Quadros estendidos, de acordo com especificação V2.0 parte B, contêm um identificador de 29 bits permitindo mais de 536 milhões de identificadores de mensagem. O identificador de 29 bits é composto do identificador de 11 bits (ID Básico) e o Identificador Estendido de 18 bits (ID Estendido).

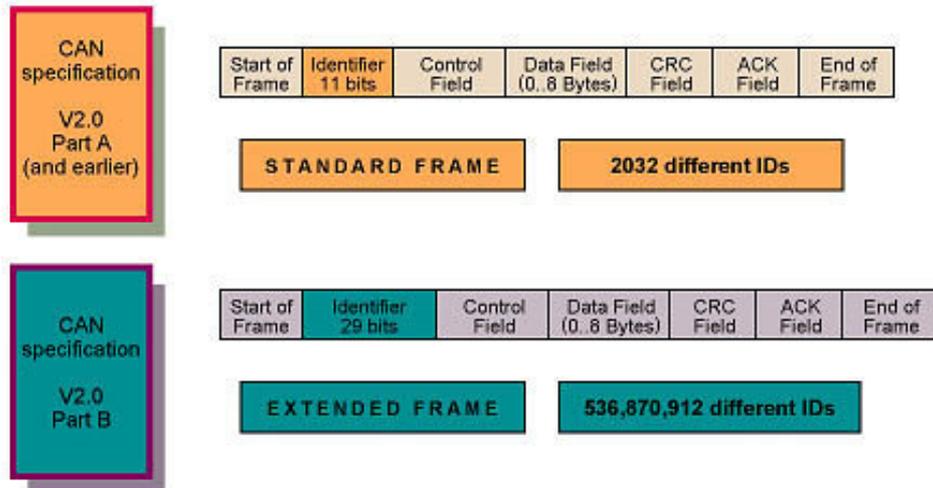


Figura 10. CAN padrão e CAN estendido. [8]

Há três tipos diferentes de módulos CAN disponíveis:

- 2.0A - considera identificadores de 29 bits como um erro.
- 2.0B Passivo - ignora identificadores de mensagem de 29 bits
- 2.0B Ativo - aceita ambos identificadores de mensagem de 11 e 29 bits.

Módulos CAN especificados conforme CAN V2.0A podem transmitir e receber somente Quadros Padrões de acordo com o protocolo de CAN Padrão. Mensagens com identificador de 29 bits enviadas a um módulo CAN Padrão gera erros no barramento.

Se um dispositivo é especificado conforme CAN V2.0B, há mais de uma distinção. Módulos nomeados "V2.0B Passivo" só podem transmitir e podem receber Quadros Padrões mas toleram Quadros Estendidos sem gerar Quadros de Erro. Dispositivos "V2.0B Ativos" podem transmitir e receber Quadros Padrão e Estendidos.

A figura 11 demonstra as diferenças entre os módulos 2.0A, 2.0B passivo e 2.0B ativo.

	Frame with 11 bit ID	Frame with 29 bit ID
V2.0B Active CAN Module	Tx/Rx OK	Tx/Rx OK
V2.0B Passive CAN Module	Tx/Rx OK	Tolerated
V2.0A CAN Module	Tx/Rx OK	<u>Bus ERROR</u>

Figura 11. Diferenças entre versões do Protocolo CAN. [8]

2.7. QUADROS DE MENSAGEM

Veremos agora os quadros de mensagem existentes no protocolo CAN.

2.7.1. Quadro de Dados

Um Quadro de Dados é gerado por um nó do CAN quando o nó deseja transmitir dados. É formado por vários campos. E está dividido em Quadro de Dados Padrão e Estendido.

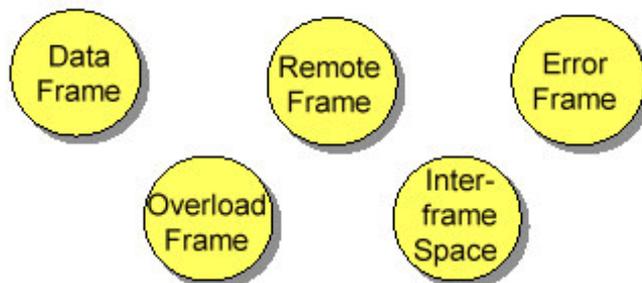


Figura 12. Quadros de dados existentes no Protocolo CAN. [8]

2.7.1.1. Quadro de Dados Padrão

O quadro começa com um bit dominante de “Início de Quadro” (Start of Frame) para sincronização de todos os nós receptores. O bit “Início de Quadro” é seguido pelo Campo de Arbitragem (Arbitration Field) que consiste em 12 bits: os 11 bits do identificador que refletem o conteúdo e prioridade da mensagem mais o bit de “Pedido de Transmissão Remota” (Remote Transmission Request, ou RTR) usado para distinguir um Quadro de Dados (RTR = dominante) de um Quadro Remoto (RTR = recessivo).

O segundo campo é o Campo de Controle (Control Field) e consiste em 6 bits. O primeiro bit deste campo é chamado de bit IDE (Identificador de Extensão) e está em estado dominante para especificar que o quadro é um quadro padrão. O bit seguinte é reservado e definido como um bit dominante. Os 4 bits restantes do Campo de Controle são o Código de Tamanho de Dados (Data Length Code, ou DLC) e especifica o número de bytes de dados contido na mensagem (0 - 8 bytes).

O dados enviados seguem no Campo de Dados (Data Field) cujo tamanho é definido pelo DLC acima (0, 8, 16,....., 56 ou 64 bits). Em seguida vem o Campo de Redundância Cíclica (Cyclic Redundancy Field, ou CRC Field), usado para descobrir possíveis erros de transmissão. O Campo CRC consiste em uma sucessão de 15 bits CRC, completada por um bit delimitador recessivo.

O terceiro campo é o Campo de Reconhecimento (Acknowledge Field). Durante esta janela o nó que está transmitindo a mensagem envia um bit recessivo. Qualquer nó que tenha recebido um quadro sem erro reconhece a correta recepção do quadro mandando de volta um bit dominante (independente do nó estar configurado para aceitar aquela mensagem específica ou não). Com isto, pode-se concluir que o CAN pertence ao grupo de protocolos “in-bit-response”. Em seguida vem o bit delimitador que é recessivo e não pode ser reescrito por um bit dominante.

Sete bits recessivos (Fim de Quadro) finalizam o Quadro de Dados. Entre dois quadros o barramento deve permanecer no mínimo três bits recessivos. Se nenhum nó quiser mais transmitir, o barramento permanece no estado de repouso (Bus Idle).

A figura 13 mostra o Quadro de Dados Padrão e o posicionamento dos bits e campos.

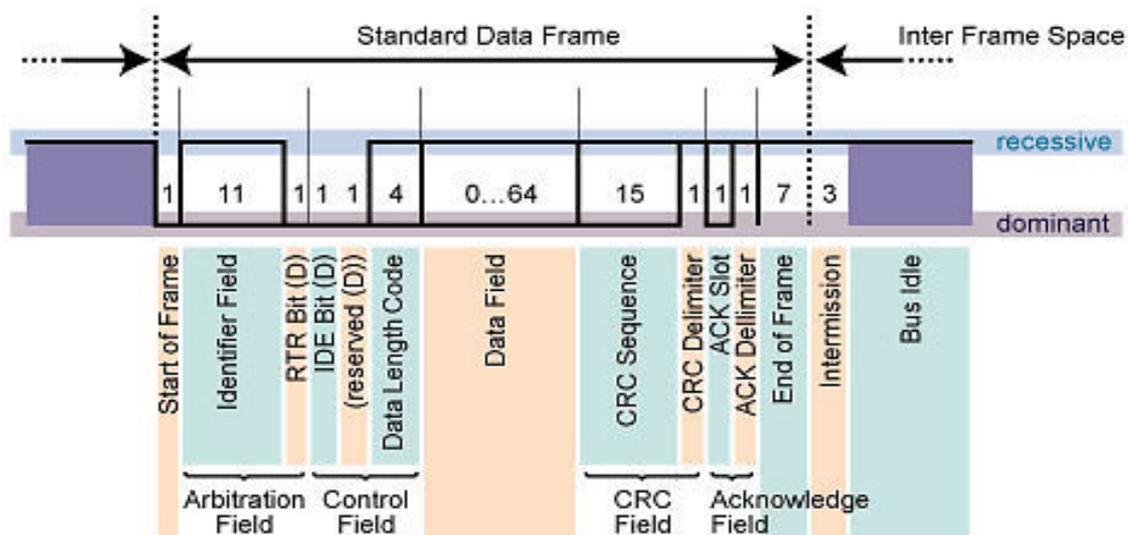


Figura 13. Quadro de dados Padrão. [8]

2.7.1.2. Quadro de Dados Estendido

O quadro começa com um bit dominante de “Início de Quadro” (Start of Frame) para sincronização de todos os nós receptores. O bit “Início de Quadro” é seguido pelo Campo de Arbitragem (Arbitration Field) que consiste em 32 bits: os 11 bits do identificador que refletem o conteúdo e prioridade da mensagem, mais um bit de “Requisição Remota Substituta” (Substitutes Remote Request, ou SRR) que é sempre recessivo, mais um bit IDE (Identificador de Extensão) sempre recessivo para especificar que o quadro é um quadro estendido. Em seguida vêm os 18 bits de extensão do identificador que, somados aos 11 formam um identificador de 29 bits. O último bit deste campo é o bit RTR que têm a mesma função que no quadro de dados padrão, ou seja, verificar se o quadro é um quadro de dados ou um quadro remoto.

O segundo campo é o Campo de Controle (Control Field) e consiste em 6 bits. Os dois primeiros são sempre dominantes e são reservados. Os 4 bits restantes do Campo de Controle são o Código de Tamanho de Dados (Data Length Code, ou DLC) e especifica o número de bytes de dados contido na mensagem (0 - 8 bytes).

O restante dos campos é idêntico ao quadro de dados padrão.

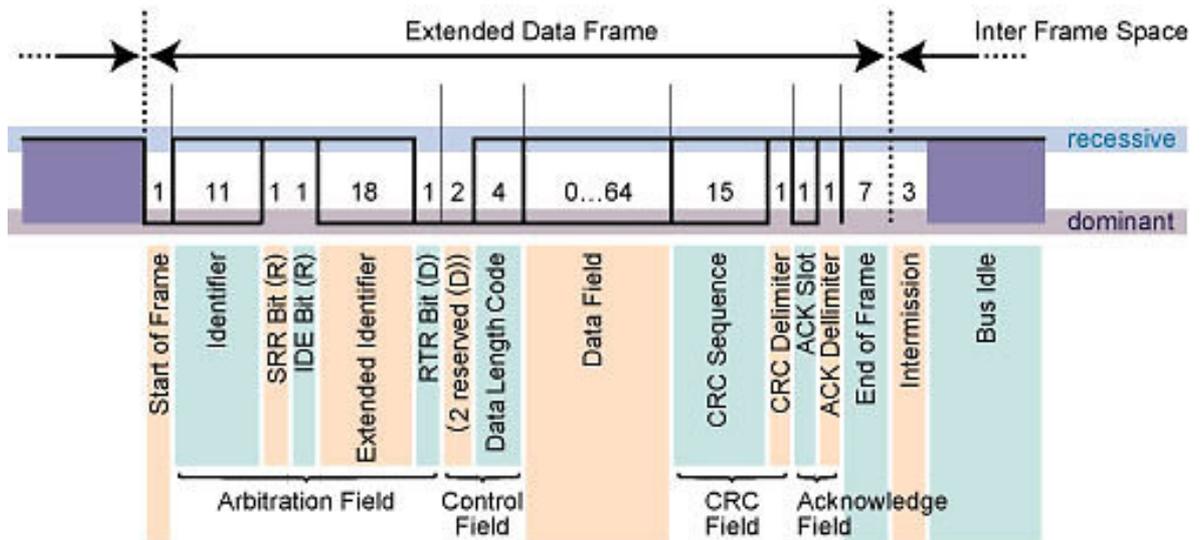


Figura 14. Quadro de dados Estendido. [8]

2.7.2. Quadro Remoto

Geralmente a transmissão de um dado é executada por uma base autônoma com o nó de fonte de dados (por exemplo um sensor) enviando um Quadro de Dados. Também é possível, porém, pedir para um nó de destino os dados da fonte enviando um Quadro Remoto. Pode ser também “padrão” ou “estendido”, como no quadro de dados, sendo que há duas diferenças entre um Quadro de Dados e um Quadro Remoto.

Primeiramente o bit RTR é transmitido como um bit dominante no Quadro de Dados e depois, no Quadro Remoto, não há nenhum Campo de Dados. Num caso muito pouco provável de um Quadro de Dados e um Quadro Remoto com o mesmo identificador serem transmitido ao mesmo tempo, o Quadro de Dados ganha a arbitragem devido ao bit RTR dominante seguindo o identificador. Deste modo, o nó que transmitiu o Quadro Remoto recebe os dados desejados imediatamente.

Se um nó deseja pedir os dados da fonte, este envia um Quadro Remoto com um identificador que corresponda com o identificador do Quadro de Dados exigido. O nó da fonte de dados apropriado enviará então um Quadro de Dados como resposta para este pedido remoto com o mesmo identificador do Quadro Remoto.

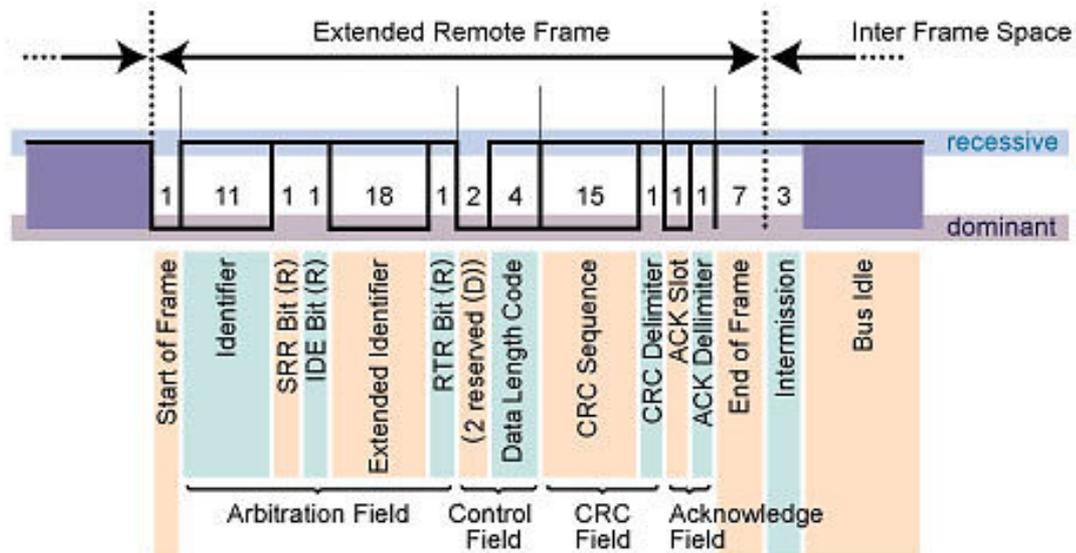


Figura 15. Quadro de dados Estendido. [8]

2.7.3. Quadro de Erro

O Quadro de Erro é gerado por qualquer nó que detecta um erro no barramento. Consiste de dois campos, um campo Flag de Erro seguido de um campo Delimitador de Erro. O Delimitador de Erro tem em 8 bits recessivos que encerram o Quadro de Erros e permite aos nós do barramento reiniciar as comunicações de forma limpa depois de um erro.

Existem duas formas para o campo Flag de Erro, dependendo do status do erro do nó que detectou o mesmo:

Primeira: Se o nó que descobre um erro no barramento é do tipo “erro-ativo”, então o nó interrompe a transmissão da mensagem atual gerando um campo Flag de Erro ativo composto de seis bits dominantes sucessivos. Esta seqüência de bits viola a regra “Bit-Stuffing”. Todas as outras estações reconhecem o erro resultante de “Bit-Stuffing” e em troca geram seu próprio Quadro de Erro. O campo Flag de Erro consiste então entre seis e doze bits dominantes sucessivos (gerado por um ou mais nós). O campo Delimitador de Erro completa o Quadro de Erro. Depois da conclusão do Quadro de Erro a atividade no barramento volta ao normal e o nó interrompido tenta retransmitir a mensagem abortada.

Segunda: Se o nó que descobre um erro no barramento é do tipo “erro-passivo”, então o nó transmite um campo Flag de Erro passivo seguido, novamente, pelo campo Delimitador de Erro. O campo Flag de Erro passivo consiste de seis bits recessivos sucessivos, e, portanto, o Quadro de Erro (para um nó de “erro-passivo”) consiste em 14 bits recessivos (isto é, nenhum bit dominante). Com isso podemos concluir que, a menos que o erro no barramento seja descoberto pelo nó que está transmitindo de fato (mestre do barramento), a transmissão de uma Quadro de Erro por um nó do tipo “erro-passivo” não afeta nenhum outro nó na cadeia. Se o nó mestre do barramento gera um Flag de Erro passivo, isto pode fazer com que outros nós gerem Quadros de Erro devido à violação do “Bit Stuffing” resultante.

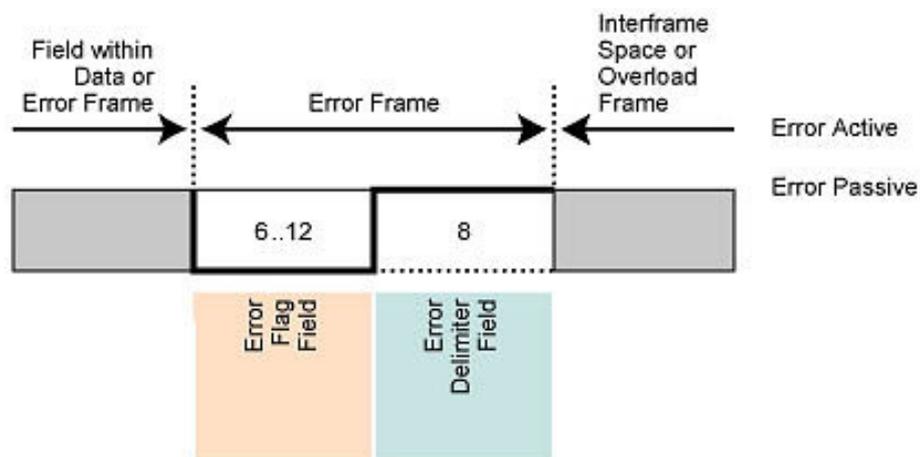


Figura 16. Quadro de Erro. [8]

2.7.4. Quadro de Sobrecarga

Um Quadro de Sobrecarga tem o mesmo formato de um Quadro de Erro ativo, porém só pode ser gerado durante Espaço entre Quadros. Este é então o modo que um Quadro de Sobrecarga pode ser diferenciado de um Quadro de Erro (enviado durante a transmissão de uma mensagem). Consiste de 2 campos, um Flag de Sobrecarga seguido por um Delimitador de Sobrecarga. O Flag de Sobrecarga tem seis bits dominantes seguidos por Flags de Sobrecarga gerados por outros nós (como no Flag de Erro ativo, dando um máximo de doze bits dominantes novamente).

O Delimitador de Sobrecarga consiste em oito bits recessivos. Um Quadro de Sobrecarga pode ser gerado por um nó se, devido às condições internas, ele não esteja apto a começar a recepção da próxima mensagem ou se durante um Espaço entre Quadros for reconhecido um bit dominante. Um nó pode gerar um máximo de 2 Quadros de Sobrecarga seguidos, a fim de retardar o início da próxima mensagem. Um Quadro de Sobrecarga só pode ser transmitido do primeiro bit do Espaço entre Quadros em diante, para poder distinguir do Quadro de Erro.

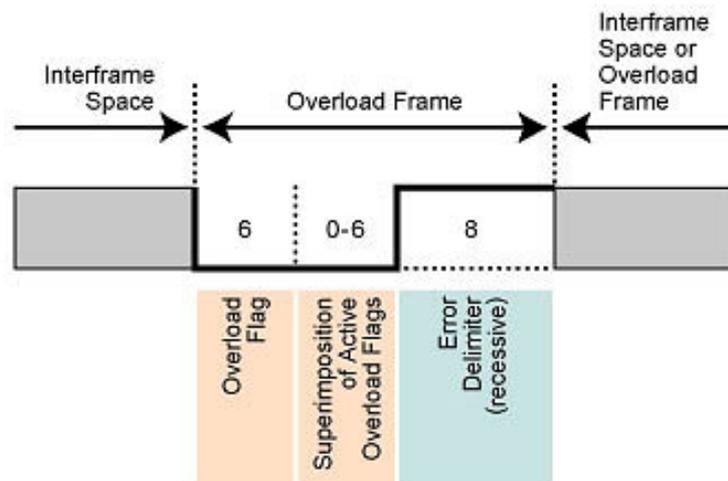


Figura 17. Quadro de Sobrecarga. [8]

2.7.5. Espaço entre Quadros

Espaço entre Quadros separa um quadro predecessor (de qualquer tipo) de um Quadro de Dados Remoto. É composto de pelo menos 3 bits recessivos, denominados de Intermissão. Este tempo é provido para permitir aos nós um tempo de processo interno antes do início do próximo quadro de mensagem. Depois da Intermissão, para nós do tipo “erro-ativo” as linhas do barramento permanecem no estado recessivo (Barramento Inativo) até o próximo início de transmissão.

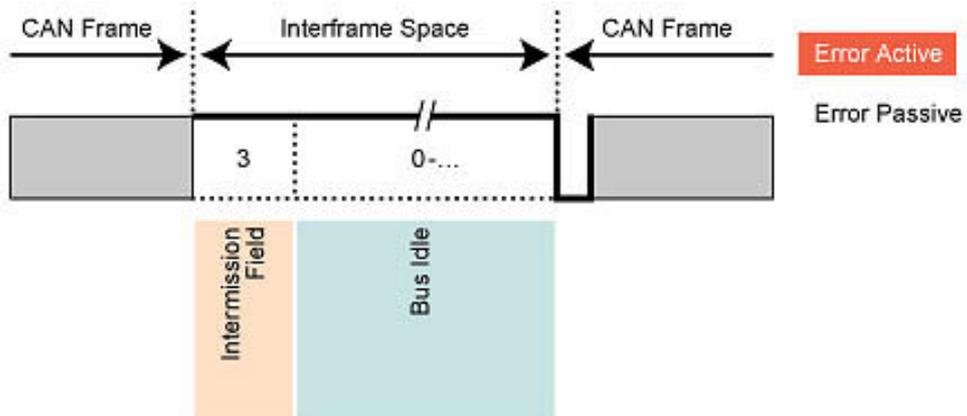


Figura 18. Quadro de Espaço entre quadros. [8]

2.8. RECONHECIMENTO E MANIPULAÇÃO DE ERROS

O protocolo CAN provê os seguintes mecanismos de reconhecimento de erro:

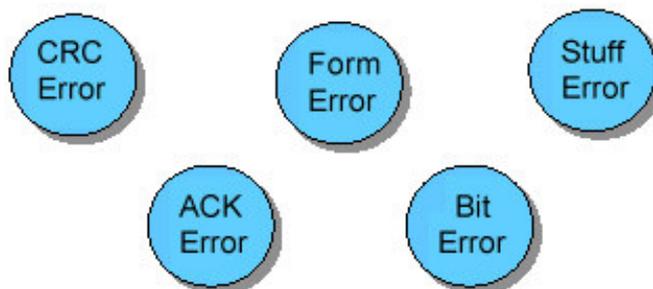


Figura 19. Mecanismos de manipulação de erros no barramento. [8]

2.8.1. Verificação de Redundância Cíclica (CRC - Cyclic Redundancy Check)

O transmissor calcula o “check sum” da seqüência de bits desde o bit de Início de Quadro” até o fim do Campo de Dados, e transmite esse valor no Campo CRC dos Quadros de Dados e Quadros Remotos do CAN. O nó receptor também calcula a seqüência CRC usando a mesma fórmula e executa uma comparação com o valor recebido no campo CRC. Se o nó B descobre um valor diferente entre o calculado e o recebido, então um erro CRC aconteceu. O nó B descarta a mensagem e transmite uma Quadro de Erro para pedir a retransmissão do quadro adulterado.

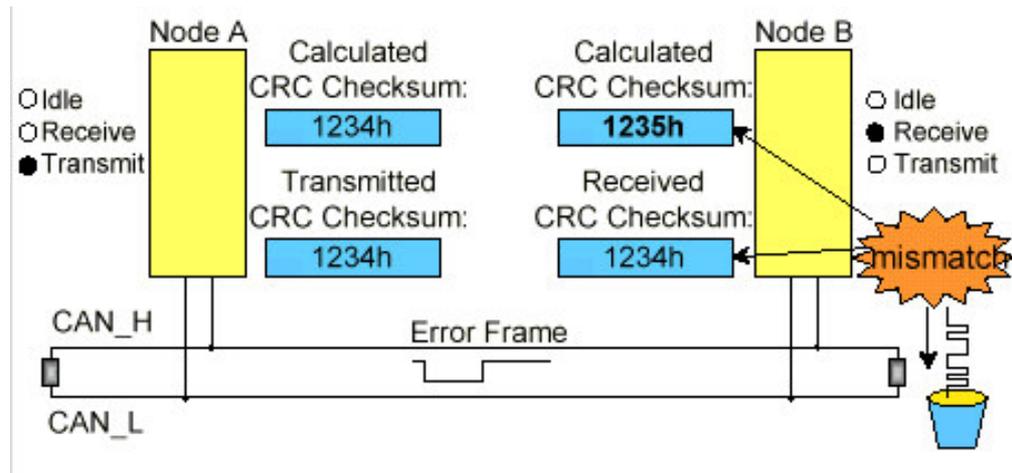


Figura 20.CRC error. [8]

2.8.2. Verificação de reconhecimento

Com o Cheque de Reconhecimento o transmissor verifica o Campo de Reconhecimento de uma mensagem para determinar se o bit de Abertura de Reconhecimento (ACK Slot) que é enviado como um bit recessivo contém um bit dominante. Se este for o caso, pelo menos um outro nó recebeu o quadro corretamente. Se não, um erro de Reconhecimento ocorre e a mensagem tem que ser repetida. Entretanto, nenhum Quadro de Erro é gerado.

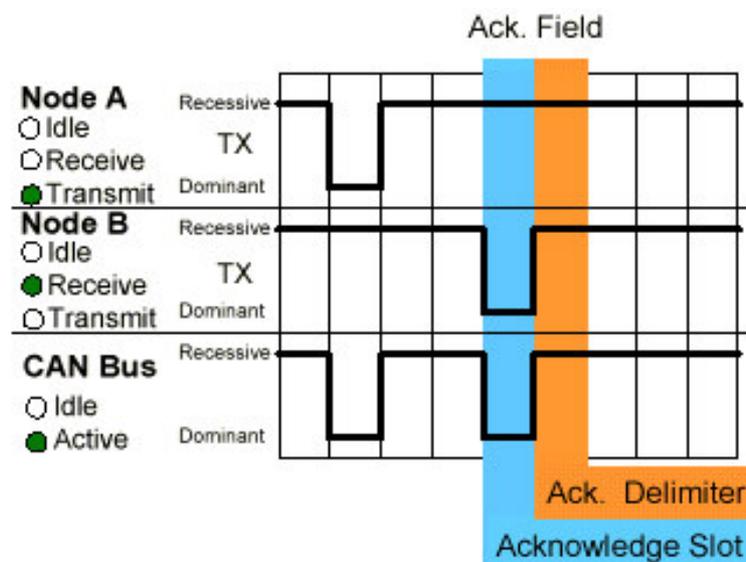


Figura 21.Cheque de reconhecimento. [8]

2.8.3. Verificação de Quadro

Outro mecanismo de reconhecimento de erro é a Verificação de Quadro. Se um transmissor descobre um bit dominante em um dos quatro segmentos: Delimitador CRC, Delimitador de Reconhecimento, Fim de Quadro ou Espaço entre Quadros, então um erro de Quadro ocorre e um Quadro de Erro é gerado. A mensagem será então repetida.

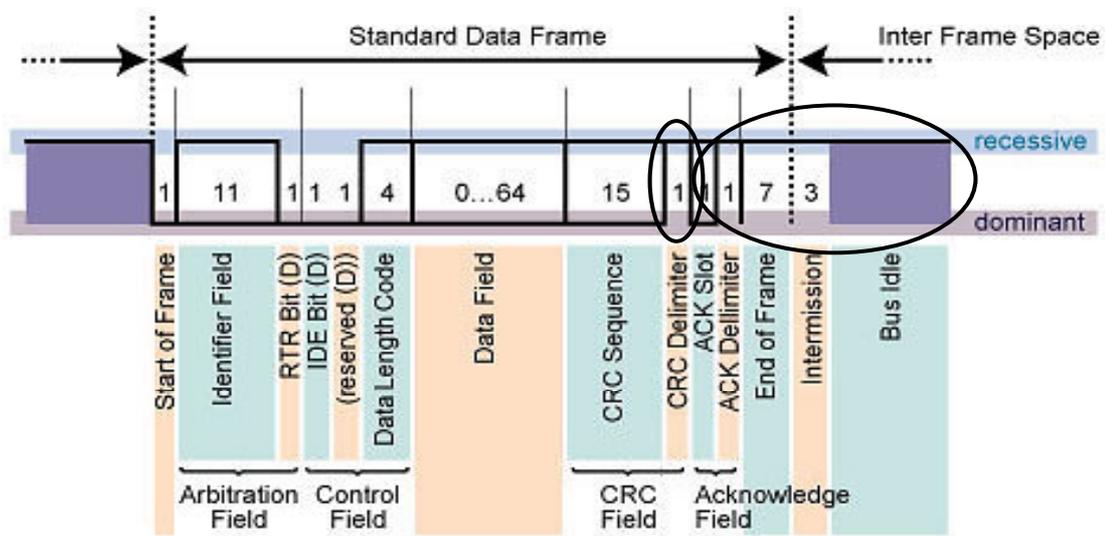


Figura 22.Verificação de Quadro. [8]

2.8.4. Monitoramento do Barramento

Todos os nós executam um monitoramento de bit. Um erro de bit acontece se um transmissor envia um bit dominante mas detecta um bit recessivo na linha do barramento ou, envia um bit recessivo mas detecta um bit dominante. Neste caso é gerado um Quadro de Erro e a mensagem é repetida.

Quando um bit dominante é detectado em vez de um bit recessivo, nenhum erro ocorre durante o Campo de Arbitragem ou no bit Abertura de Reconhecimento pois estes campos podem ser reescritos por um bit dominante para se obter a arbitragem e as funcionalidades do reconhecimento.

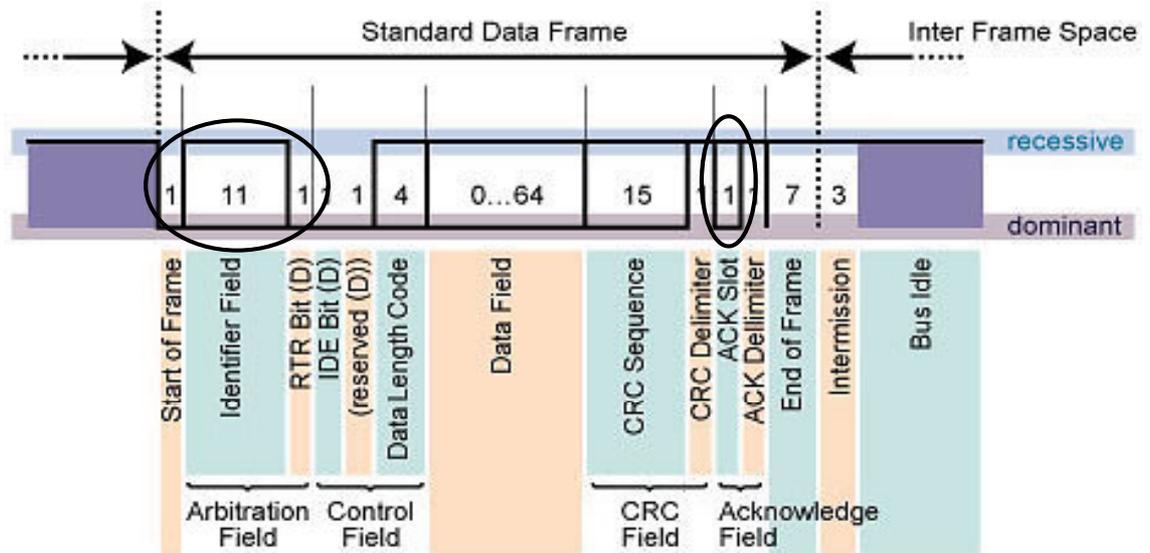


Figura 23. Monitoramento do Barramento. [8]

2.8.5. Bit Stuffing

Se forem detectados seis bits consecutivos com a mesma polaridade entre o bit de Início de Quadro e o Delimitador CRC, a regra de “Bit Stuffing” é violada. Acontece, então, um erro de Stuff gerando um Quadro de Erro. A mensagem é repetida.

2.8.6. Estados de Erro

Cada nó do CAN pode estar em um dos três estados de erro, conhecidos como "erro-ativo", "erro-passivo" ou "fora do barramento" de acordo com o valor de seus contadores internos de erro.

O estado erro-ativo é o estado normal logo após um reset. O nó do barramento pode então receber e transmitir mensagens e pode transmitir Quadro de Erro ativo (composto por bits dominantes) sem qualquer restrição. Durante a comunicação do CAN, os contadores de erro são atualizados de acordo com regras bastante complexas. Para cada erro na recepção ou transmissão os contadores de erro são incrementados em um certo valor. Para cada transação bem sucedida os contadores de erro são decrementados em um certo valor. O estado erro-ativo é válido desde que ambos os contadores de erro (recepção e transmissão) sejam menor ou igual a 127.

Se um dos contadores de erro alcançar o valor de 128, o nó passa ao estado erro-passivo. No estado erro-passivo as mensagens podem ser recebidas e transmitidas, embora, após transmissão de uma mensagem, o nó tem que suspender a transmissão, esperando 8 bits a mais que nós no estado erro-ativo antes de poder transmitir outra mensagem. Em termos de sinalização de erro, somente Quadros de Erro passivo (composto por bits recessivos) podem ser transmitidos por um nó no estado erro-passivo.

Se ambos os contadores de erro abaixam de 128 novamente devido a uma comunicação próspera no barramento, o nó passa ao estado erro-ativo.

Uma característica do protocolo de CAN é fazer com que nós defeituosos sejam retirados automaticamente do barramento. O estado “fora do barramento” é atingido se o contador de erro de transmissão excede o valor de 255. Neste caso, todas as atividades no barramento são interrompidas, o que torna temporariamente impossível a estação participar na comunicação do barramento. Durante este estado, mensagens não podem ser recebidas e nem transmitidas. Para voltar ao estado “erro-ativo” e zerar o contador de erro, o nó do CAN tem que ser reinicializado.

A figura 24 mostra o fluxo de transição entre os possíveis estados de erro.

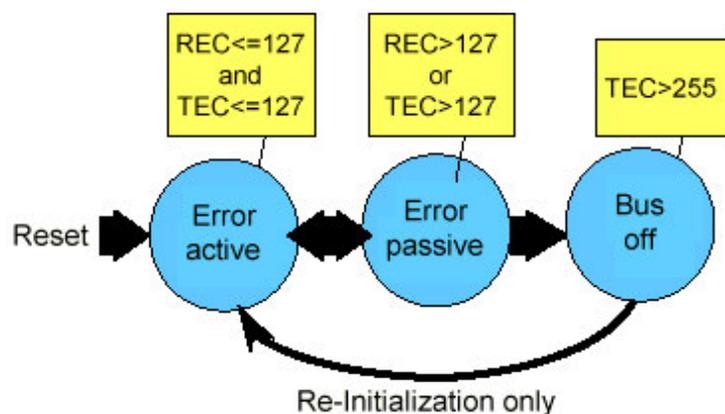


Figura 24.Estados de Erro. [8]

2.9. CONCEITOS NRZ E BIT STUFFING

O protocolo CAN usa codificação de bit NRZ (Non-Return to Zero). Isto significa que o sinal é constante durante um tempo de bit inteiro e só um intervalo de tempo é necessário para representar um bit.

Normalmente, mas não sempre, um bit "zero" corresponde a um bit dominante deixando o barramento no estado dominante, e um bit "um" corresponde a um bit recessivo colocando o barramento no estado recessivo.

Uma característica da codificação NRZ é que o sinal não gera flancos (que poderiam ser usados para re-sincronização do sinal) quando a quantidade de bits sucessivos transmitidos com a mesma polaridade for grande.

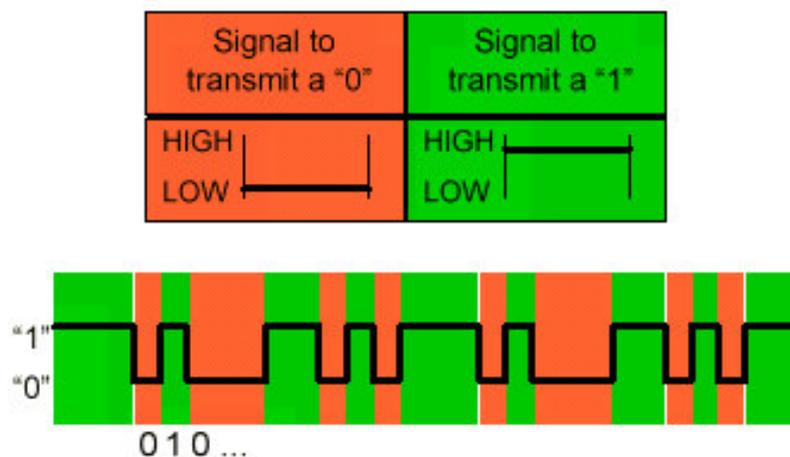


Figura 25. Conceito NRZ. [8]

Portanto, o conceito de Bit-Stuffing é usado para assegurar a sincronização de todos os nós do barramento. Isto significa que durante a transmissão de uma mensagem, um máximo de cinco bits consecutivos podem ter a mesma polaridade. Sempre que foram transmitidos cinco bits consecutivos de mesma polaridade, o transmissor inserirá um bit adicional (Stuffing) de polaridade oposta ao fluxo de bits, antes de transmitir bits adicionais. O receptor também confere o número de

bits com a mesma polaridade e remove os bits inseridos artificialmente ao fluxo de bit. Isto é chamado "de-stuffing".

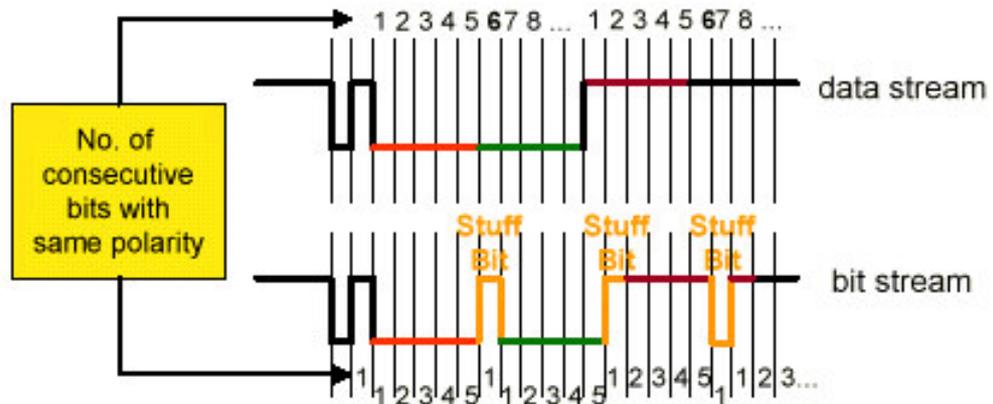


Figura 26. Conceito Bit Stuffing. [8]

2.10. SINCRONIZAÇÃO DO BARRAMENTO

Em contraste a muitos outros barramentos, o CAN manipula a transferência de mensagens de forma síncrona. Todos os nós são sincronizados no começo de cada mensagem com o primeiro flanco de descida do quadro correspondente ao bit de Início de Quadro. Isto é chamado de “Sincronização Dura”.

Para assegurar uma leitura correta até o último bit, os nós do CAN precisam re-sincronizar ao longo de todo o quadro. Isto é obtido em cada transição de recessivo para dominante.

3. ESTUDO COMPARATIVO

Realizaremos agora um comparativo entre o protocolo CAN e outros protocolos seriais utilizados pela indústria automobilística.

3.1. BARRAMENTOS UTILIZADOS

Entre os barramentos seriais existentes, podemos destacar alguns como os mais utilizados pela indústria automobilística. Entre as características desejáveis em um protocolo para aplicações embarcadas, podemos destacar:

- Possibilitar a redução do cabeamento da rede;
- 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 responsável pelo gerenciamento da rede;
- Robustez suficiente para garantir seu funcionamento em ambientes móveis e nocivos;
- Capacidade para detectar e tratar eventuais falhas geradas por problemas em hardware e software ou interferências eletromagnéticas;

Analisando-se as aplicações existentes, consegue-se identificar os protocolos que foram utilizados no desenvolvimento de algumas redes embarcadas. Dentre eles destacam-se o RS-232, o RS-485 e o barramento CAN.

3.1.1. RS-232

Pode ser considerado um dos protocolos mais populares de todos os tempos, existindo em praticamente todos os computadores pessoais e de grande porte. Foi concebido para a comunicação bidirecional de dados entre dois dispositivos, a uma distância máxima variando de 150m a 300m, dependendo da taxa de transmissão e do tipo de cabo utilizado. É caracterizado por utilizar linhas desbalanceadas, onde o sinal é aplicado a um dos fios e referenciado ao outro, conectado ao terra comum do sistema.

Como vantagens do RS-232, podemos destacar:

- Compatível com a grande maioria dos micro-controladores existentes;
- Protocolo de simples implementação;

Sobre as desvantagens deste protocolo, podemos destacar:

- Dificuldade de expansão e restrições à implementação de um controle distribuído, especialmente por se trabalhar com duas ECUs (várias ECUs poderiam ser utilizadas caso fosse aplicado um sistema multiseria);
- Grande quantidade de cabeamento, principalmente pela dificuldade de instalação de várias ECUs (passa a ser necessário levar os sinais dos sensores a um dos dois nós existentes, nem sempre localizados em suas proximidades);
- Problemas com interferências eletromagnéticas devido a não ser uma rede de linhas balanceadas ou diferenciais;

3.1.2. RS-485

Este protocolo possibilita a troca de dados entre uma quantidade maior de nós e a uma distância maior entre eles se comparado com o RS-232. A quantidade de pontos conectados à rede, dependendo do comprimento da mesma e da taxa de transmissão, pode chegar a 256. Apesar disso, existe somente um nó capaz de conversar com todos os demais, sendo conhecido como mestre.

Como vantagens do RS-485, podemos destacar:

- Possibilidade de expansão da rede a até 256 nós;
- Protocolo de simples implementação;
- Menor comprimento total dos cabos, uma vez que os nós podem ser melhor distribuídos na rede e aproximados dos sensores;
- Maior imunidade a ruídos elétricos, uma vez que adota par trançado e loop de corrente em sua linha de comunicação;

Sobre as desvantagens, podemos destacar o fato de não ser um controle, verdadeiramente, distribuído, caracterizando na uma comunicação mestre-escravo.

3.2. COMPARATIVO

Como vimos no item 2.5, existem três classes para aplicações de barramentos em sistemas embarcados.

- Classe A: Para um barramento de velocidade baixa, com taxas de até 10 Kbps, como por exemplo, aplicações de gerenciamento de sistemas no chassi;
- Classe B: Para um barramento de velocidade intermediária, com taxas de 10 Kbps a 125 Kbps, como por exemplo, painel e diagnósticos;

- Classe C: Para um barramento de velocidade alta, com taxas de 125 kbps a 1 Mbps para aplicações de tempo real, como gerenciamento do motor, caixa de câmbio, ABS, etc.

A figura 27 mostra a aplicação de cada classe sendo utilizada em um automóvel, sendo que a classe A é mencionada como “REDE 3”, a classe B como “REDE 2” e a classe C como “REDE 1”.

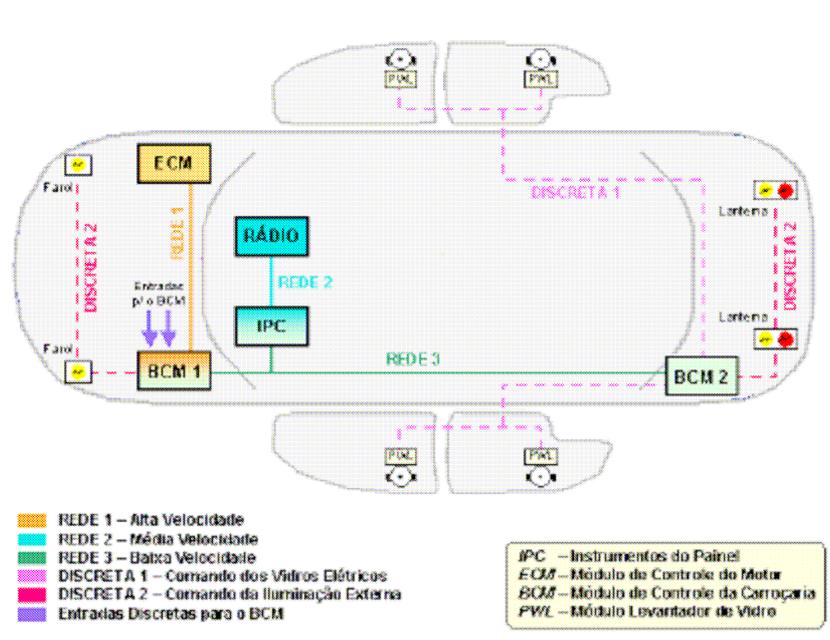


Figura 27. Exemplo de barramento em um automóvel. [11]

A seguir são apresentadas algumas tabelas que relacionam os diversos tipos de protocolos existentes divididos em suas classes de aplicação.

Classe A: Protocolos com Taxa de Transmissão de até 10Kbps. Geralmente relacionados às funções de conforto de um veículo.

	SINEBUS	PC	SAE J1708	CCD	ACP	BEAN	LIN
Instituição diret. relacionada	Delco Electronics	Philips	SAE / TMC	Chrysler	Ford	Toyota	Motorola
Aplicação principal	Sistemas de Áudio	Comunicação entre Displays e Rádios	Controle e Diagnóstico	Controle e Diagnóstico	Sistemas de Áudio	Controle e Diagnóstico	Sensores e Atuadores Inteligentes
Tipo de barramento	Fio Único	Par Trançado	Par Trançado	Fio Único	Par Trançado	Fio Único	Fio Único
Codificação dos sinais	SAM	AM	NRZ (Non Return to Zero)				
Deteção de erros	Não Disponível	Bit ACK	Check SUM de 8 Bits	Check SUM de 8 Bits	Check SUM de 8 Bits	CRC de 8 Bits	Check SUM de 8 Bits
Quantidade de dados	10 - 18 bits	--	--	5 Bytes	6 - 12 Bytes	1 - 11 Bytes	8 Bytes
Taxa de transmissão	66,6 Kbps - 200 Kbps	1 Kbps - 100 Kbps	9.600 bps	7.812,5 bps	9.600 bps	10 Kbps	20 Kbps
Comprimento máximo do barramento	10 metros	Não especificado	40 metros	--	40 metros	--	40 metros
Quantidade máxima de nós na rede	--	--	20	10	20	20	16

Tabela 1. Comparativo entre barramentos aplicados à classe A. [11]

Podemos observar que o protocolo CAN não está nesta classe de protocolos.

Classe B: Protocolos com Taxa de Transmissão de 10Kbps a 125Kbps. Geralmente relacionados ao controle dos sistemas de entretenimento de um veículo.

	CAN 2.0 ISO11898	CAN 2.0 SAE J1939	J1850 Class 2	J1850 SCP	J1850 PCI
Instituição diret. relacionada	SAE & ISO	SAE	GM	Ford	Chrysler
Aplicação principal	Controle e Diagnóstico	Controle e Diagnóstico	Controle e Diagnóstico	Controle e Diagnóstico	Controle e Diagnóstico
Tipo de barramento	Par Trançado	Par Trançado	Fio Único	Par Trançado	Fio Único
Codificação dos sinais	NRZ (Non Return to Zero)	NRZ (Non Return to Zero)	VPW	PWM	VPW
Deteção de erros	CRC	CRC	CRC	CRC	CRC
Quantidade de dados	0 - 8 Bytes	8 Bytes	0 - 8 Bytes	0 - 8 Bytes	0 - 10 Bytes
Taxa de transmissão	10 Kbps - 1 Mbps	250 Kbps	10,4 Kbps	41,6 Kbps	10,4 Kbps
Comprimento máximo do barramento	40 metros (p/ 1 Mbps)	40 metros (p/ 1 Mbps)	35 metros	35 metros	35 metros
Quantidade máxima de nós na rede	32	32	32	32	32

Tabela 2. Comparativo entre barramentos aplicados à classe B. [11]

Classe C: Protocolos com Taxa de Transmissão de 125Kbps a 1Mbps. Geralmente relacionados ao controle dos sistemas de segurança de um veículo.

	CAN 2.0 ISO 11898 & ISO11519-2	CAN 2.0 SAE J1939
Instituição diretamente relacionada	SAE & ISO	SAE
Aplicação Principal	Controle e Diagnóstico	Controle e Diagnóstico
Tipo de Barramento	Par Trançado	Par Trançado
Codificação dos Sinais	NRZ (Non-Return to Zero)	NRZ (Non-Return to Zero)
Detecção de Erros	CRC	CRC
Quantidade de Dados	0 - 8 Bytes	8 Bytes
Taxa de Transmissão	10 Kbps - 1 Mbps	250 Kbps
Comprimento Máximo do Barramento	40 metros (p/ 1Mbps)	40 metros (p/ 1Mbps)
Quantidade Máxima de Nós na Rede	32	32

Tabela 3. Comparativo entre barramentos aplicados à classe C. [11]

Percebe-se que, para cada aplicação, existe uma série de protocolos disponíveis, validados e com tecnologia conhecida. Ainda assim, o CAN é o protocolo com a maior variedade de configurações possíveis, sendo capaz de cobrir a grande maioria das necessidades relacionadas a troca de dados dos veículos automotivos atualmente existentes e em desenvolvimento.

4. FUNCIONAMENTO TÍPICO DE UMA APLICAÇÃO AUTOMOBILÍSTICA COM PROTOCOLO CAN

Vamos agora demonstrar o funcionamento do protocolo CAN sendo aplicado em um automóvel.

Tendo como base a figura 27, veremos a comunicação entre o Módulo IPC (instrumentos do painel) e o Módulo ECM (módulo de controle do motor). Os módulos utilizados são produzidos pela Magneti Marelli Sistemas Automotivos e implementados no veículo Palio, da montadora Fiat.

Também serão utilizados, um software de diagnose chamado Canalyzer e um simulador específico para o Painel de Instrumentos que irá possibilitar a comunicação do software Canalyzer com o barramento, este simulador também foi desenvolvido pela Magneti Marelli Sistemas Automotivos.

O Canalyzer é desenvolvido pela Vector e possibilita a simulação dos outros módulos da rede sem que estes estejam conectados fisicamente ao barramento. A figura 28 mostra todos os módulos que serão utilizados na demonstração.



(a)



(b)



(c)

Figura 28. (a) Painel de Instrumentos; (b) Central de controle do Motor; (c) Simulador para o Painel de Instrumentos.

Com o Canalyzer conectado ao Painel, pode-se comunicar com todos os outros módulos do barramento, a figura 29 mostra a janela principal do Canalyzer, esta janela está dividida em outras 3 janelas, sendo elas:

- Write: Mensagens enviadas especificamente ao Painel de Instrumentos, pode-se ler ou escrever dados na memória do Painel.
- Trace: Pode-se observar as mensagens sendo trocadas entre todos os módulos do barramento em tempo real.
- Measurement Setup: Configura as aplicações específicas do Canalyzer, como gráficos, estatística do barramento, filtros e outras funções.

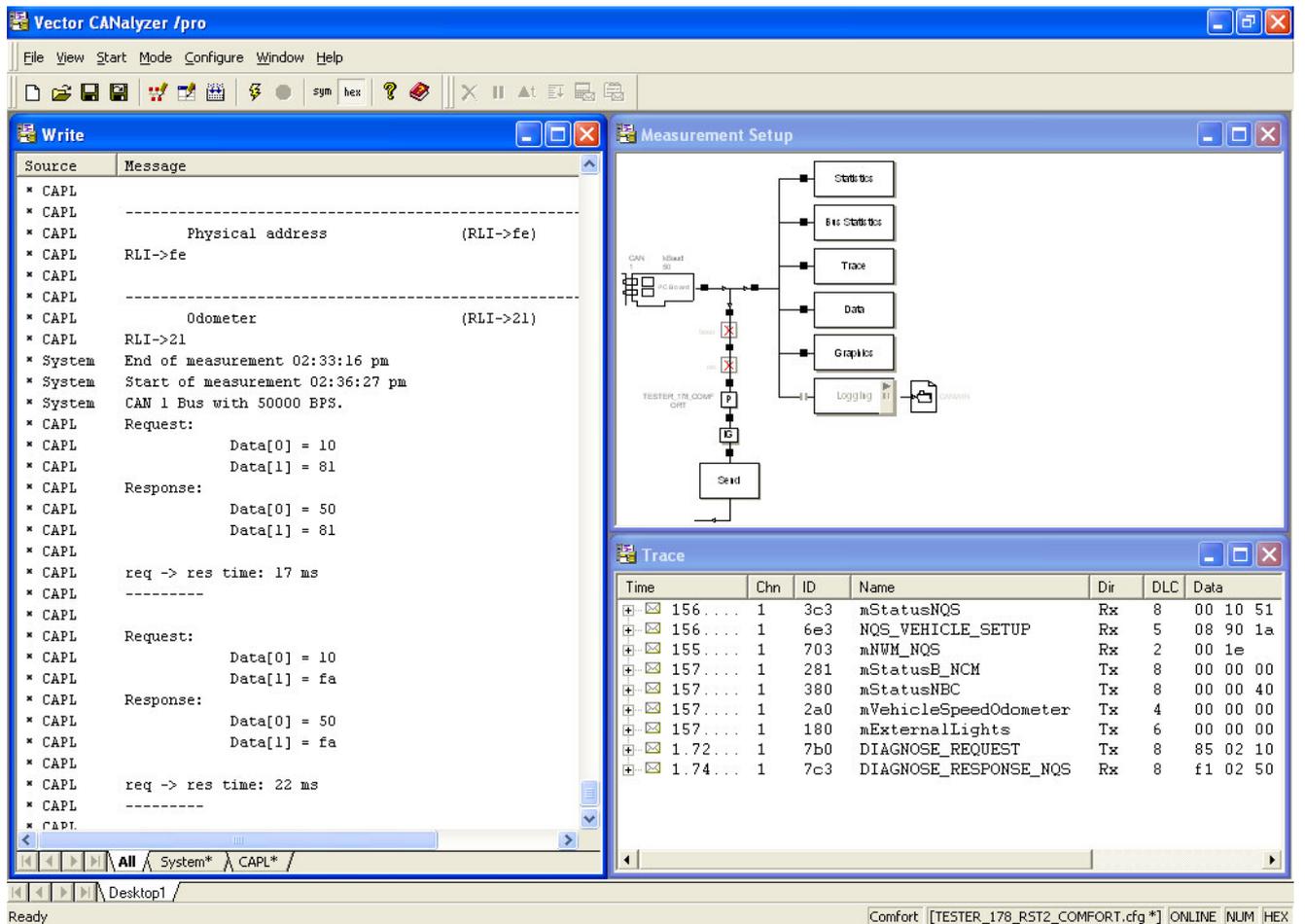


Figura 29. Janela principal do software Canalyzer.

Para simular os outros módulos no barramento temos outra janela do Canalyzer, chamada de Interactive Generator Block, que possibilita a simulação e configuração dos módulos. A figura 30 mostra esta janela.

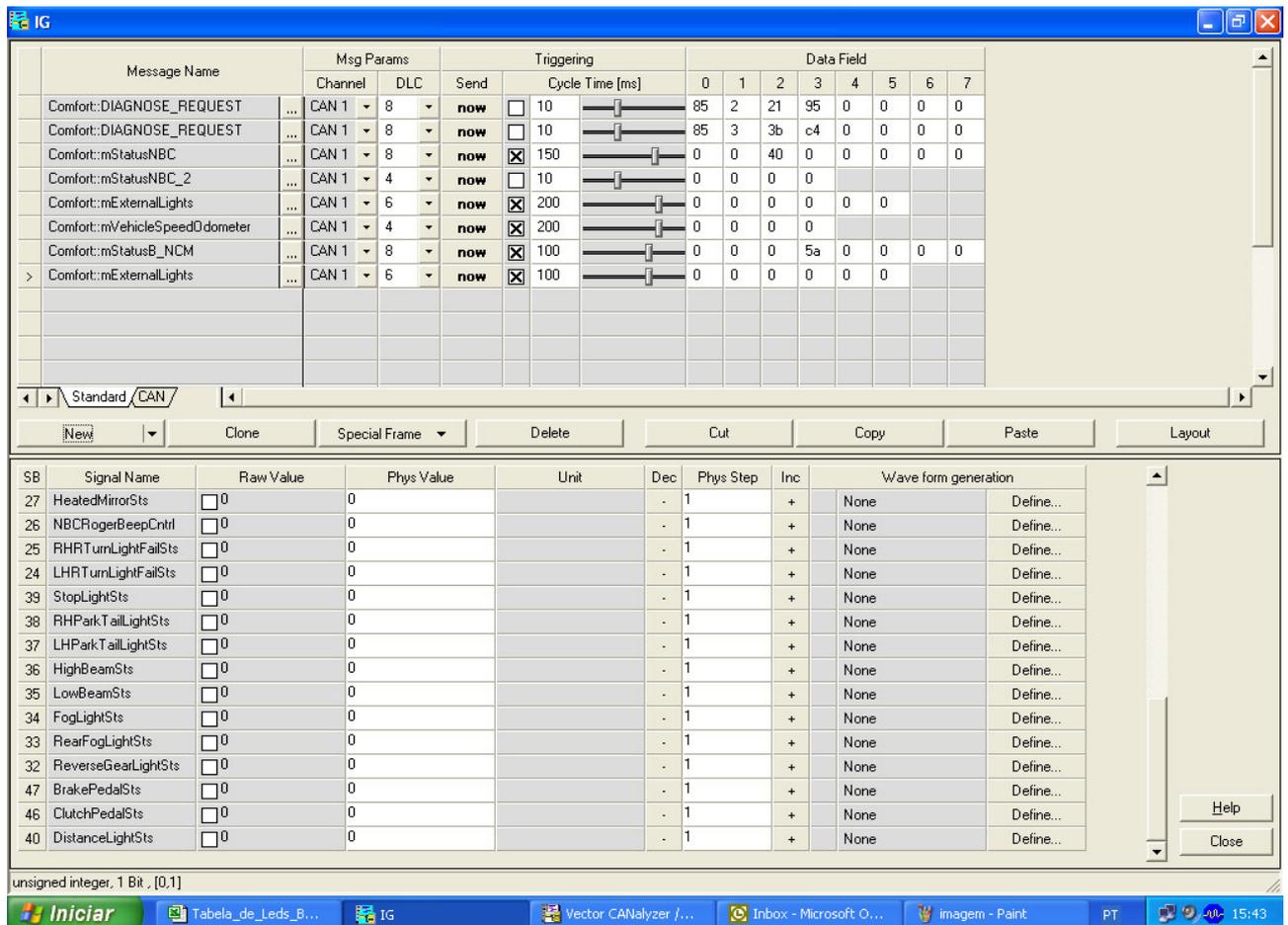


Figura 30. Janela do Interactive Generator Block do software Analyzer.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho permitiu o aprofundamento de conhecimentos no domínio da concepção, e aplicação de sistemas de comunicação serial implementadas pela indústria automobilística, em específico o Protocolo de Comunicação CAN.

As empresas, de um modo geral, e em específico as automotivas, estão encarando um mercado cada vez mais competitivo, onde a velocidade do avanço tecnológico e seu acompanhamento tornam-se fatores vitais, e sem dúvidas o não acompanhamento desse desenvolvimento tecnológico significa em pouco tempo estar obsoleto no mercado.

Estudos como este permitem a disseminação de informações facilitando o acompanhamento do desenvolvimento tecnológico que presenciamos nos dias de hoje.

A nós estudantes que buscamos sucesso profissional e pessoal, fica a necessidade de acompanhamento do desenvolvimento da nossa área de atuação e a consciência de que a vida é um aprendizado constante.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ISO 11898. *Road Vehicles – Interchange of Digital information - Controller Area Network (CAN)* for high speed communication, 1993.
- [2] ISO 11898-3. *Road Vehicles – Low speed, fault tolerant, medium dependent information*. ISO, 2003.
- [3] Bosch, Robert. *Can Specification*, version 2.0, Stuttgart, 1991
- [4] Katho. In *Vehicle Networks*, Cap. 3a, CAN: OSI Model. Bélgica, 2005.
- [5] Katho. In *Vehicle Networks*, Cap. 3b, CAN: Transceivers. Bélgica, 2005.
- [6] Katho. In *Vehicle Networks*, Cap. 3c, CAN: Data Link Layer. Bélgica, 2005
- [7] CiA. *Controller Area Network : an overview*. Disponível em: <<http://www.can-cia.de>>, acessado em 19/05/06.
- [8] Bosch, Robert. *Curso Can*, Apresentação Power Point
- [9] Luppe, Marcelo. *Can Technical Overview*, Motorola Semiconductor Products Sector, 2002.
- [10] Material retirado do site Can Bus, no endereço:
www.pcs.usp.br/~laa/Grupos/EEM/CAN_Bus_Parte_2.html, em 19/05/06.
- [11] Guimarães, Alexandre de A. *Protocolos de Comunicação em Sistemas Embarcados Automotivos, Aeroespaciais e Agrícolas*, 06/2004.