



## **PMR3412 - Redes Industriais - 2023**

Aula 12 - RS-232/485 e protocolos

---

Prof. Dr. Marcos de Sales Guerra Tsuzuki

Prof. Dr. Newton Maruyama

16 de Novembro de 2021

PMR-EPUSP

Os slides que serão utilizados nesse ano são baseados no curso desenvolvido para os anos 2020, 2021 e 2022. Participaram da concepção do curso e desenvolvimento do material os seguintes professores:

- ▶ Prof. Dr. André Kubagawa Sato
- ▶ Prof. Dr. Marcos de Sales Guerra Tsuzuki
- ▶ Prof. Dr. Edson Kenji Ueda
- ▶ Prof. Dr. Agesinaldo Matos Silva Junior
- ▶ Prof. Dr. André César Martins Cavalheiro

1. RS-232
2. RS-485
3. RS-422
4. Protocolos de Comunicação
5. CAN - Control Area Network

**RS-232**

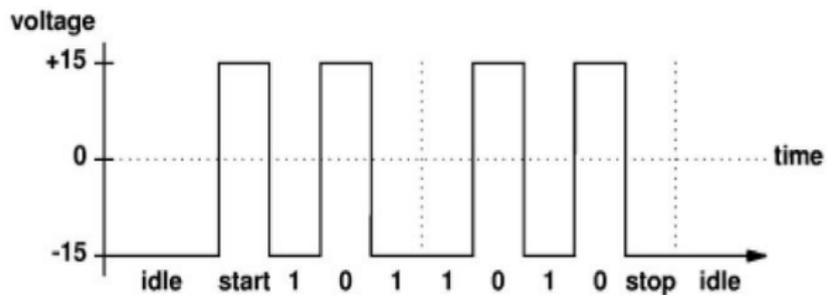
---

- ▶ A comunicação serial é a forma mais simples de comunicação entre dois dispositivos.
- ▶ É bem intuitiva, uma vez que você compreenda o padrão.
- ▶ Os sistemas em rede começaram com ela.
- ▶ RS-232 é um padrão pelo qual dois dispositivos se comunicam:
  - ▶ A conexão não pode ser mais longa que 15 metros;
  - ▶ Os potenciais de transmissão estão entre -15 V e + 15 V;
  - ▶ Ele é projetado para transmitir caracteres (comprimento de 7 bits).
  - ▶ É uma forma de comunicação assíncrona (está é uma forma eficiente de comunicação, se nenhum dado deve ser enviado, a conexão permanece “inativa” (ou *idle*)).

- ▶ Lógico 1 é -15 V;
- ▶ Lógico 0 é +15 V;
- ▶ Quando a conexão está inativa (ou *idle*), o hardware faz com que a conexão fique em lógico 1.
- ▶ A comunicação RS-232 é dependente da configuração de velocidade em que ambas as extremidades de hardware se comunicam. Em outras palavras, o hardware sabe quanto tempo um bit deve permanecer *high* ou *low*;
- ▶ O RS-232 também especifica o uso de *start* e *stop* bits.

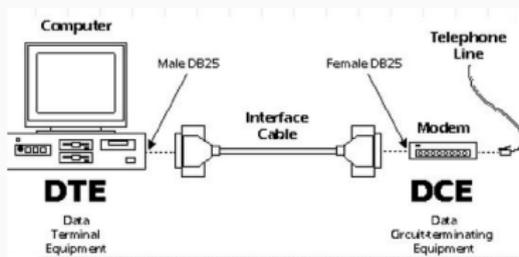
- ▶ Quando um caracter é enviado, a mesma sequência de comunicação deve ocorrer: 1. Envio de *start* bit; 2. Envio de 7 bits de dados; 3. Envio de *stop* bit.
- ▶ Esta comunicação é dependente do fato de que ambos os dispositivos estão amostrando os bits segundo a mesma taxa.
- ▶ O *start* bit é enviado como sinal lógico 0 pela linha, para informar o outro dispositivo a iniciar a amostragem;
- ▶ Lembre, que o sinal lógico 0 é + 15 V.
- ▶ O *stop* bit possui sinal lógico 1, que é - 15 V;
- ▶ o *stop* bit é sempre enviado (segundo o padrão RS-232).

## RS-232 - Enviando um Caractere

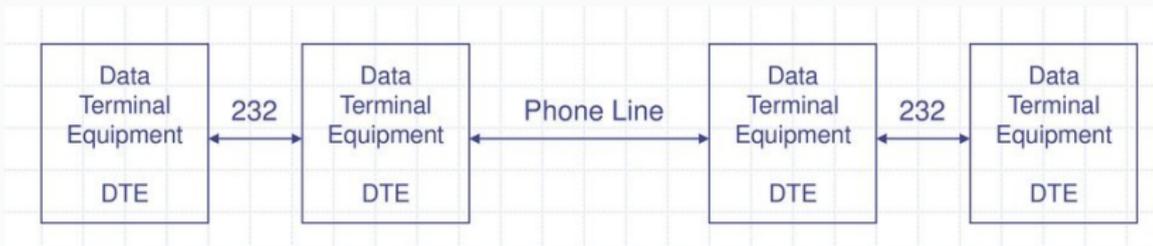


- ▶ A maioria das configurações são as seguintes: 1. bits por segundo; 2. número de bits; 3. paridade; 4. número de *stop* bits.
- ▶ A grande maioria dos dispositivos vinham com a configuração: 9.600, 8, none, 1.
- ▶ Os hardwares de RS-232 amostram a linha múltiplas vezes durante a transmissão de um único bit;
- ▶ Se as amostras não possuem todas o mesmo potencial, um erro de frame (*framing error*) ocorre;
- ▶ Um erro de frame pode ocorrer se um dispositivo está enviando dados mais rapidamente queo outro dispositivo consegue receber.

## RS-232 - Padrão de Interface Serial



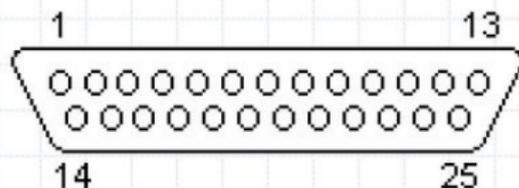
- ▶ EIA - Electronic Industry Associates, o primeiro padrão RS-232 foi criado em 1969 para:
  - ▶ DTE para DCE;
  - ▶ Inicialmente considerou linhas telefônicas utilizando modems em cada terminal.
- ▶ O último padrão foi EIA232E em 1991 (não possui RS, mas todo mundo continua utilizando RS).



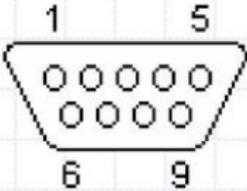
▶ Especificação básica:

- ▶ Máxima taxa de transmissão: 19.200 bps;
- ▶ Máximo comprimento de cabo: 15 metros;
- ▶ Máxima capacitância/metro: 15 pF;
- ▶  $\| V_{\max} \| = \pm 25 \text{ V}$ ;
- ▶ Lógica negativa: sinal lógico 1 = -12 V, sinal lógico 0 = +12 V;
- ▶ Possui dois padrões de conectores e pinagens: 1. DB9 um conector com 9 pinos; 2. DB25 um conector com 25 pinos.

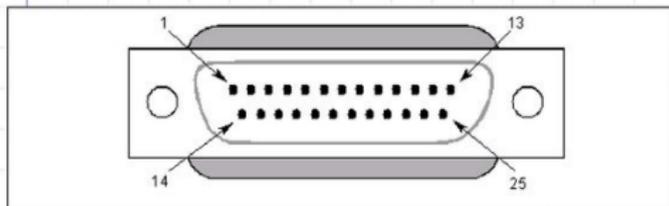
DB-25M	Function	Abbreviation
Pin #1	Chassis/Frame Ground	GND
Pin #2	Transmitted Data	TD
Pin #3	Receive Data	RD
Pin #4	Request To Send	RTS
Pin #5	Clear To Send	CTS
Pin #6	Data Set Ready	DSR
Pin #7	Signal Ground	GND
Pin #8	Data Carrier Detect	DCD or CD
Pin #9	Transmit + (Current Loop)	TD+
Pin #11	Transmit - (Current Loop)	TD-
Pin #18	Receive + (Current Loop)	RD+
Pin #20	Data Terminal Ready	DTR
Pin #22	Ring Indicator	RI
Pin #25	Receive - (Current Loop)	RD-



<b>DB-9M</b>	<b>Function</b>	<b>Abbreviation</b>	
Pin #1	Data Carrier Detect	CD	
Pin #2	Receive Data	RD or RX or RXD	
Pin #3	Transmitted Data	TD or TX or TXD	
Pin #4	Data Terminal Ready	DTR	
Pin #5	Signal Ground	GND	
Pin #6	Data Set Ready	DSR	
Pin #7	Request To Send	RTS	
Pin #8	Clear To Send	CTS	
Pin #9	Ring Indicator	RI	

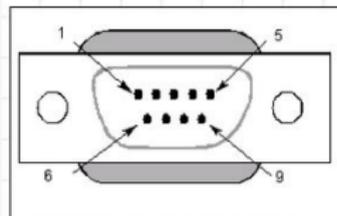


Connectors:  
Minimally, 3 wires: RxD, TxD, GND  
Could have 9-pin or 25-pin



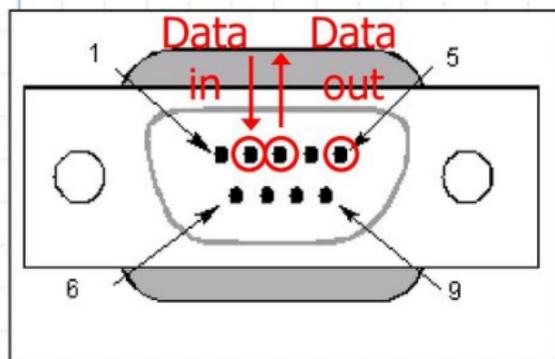
DB-25

**25-Pin Connector**



DB-9

**9-Pin Connector**

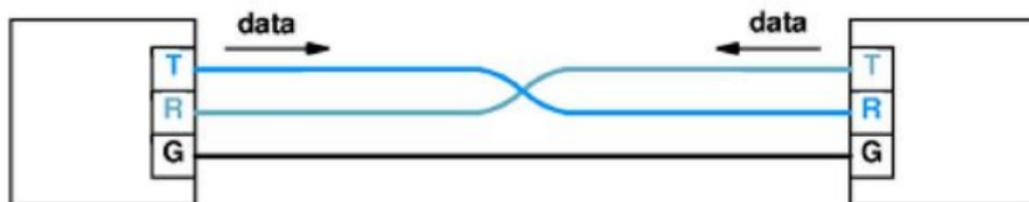


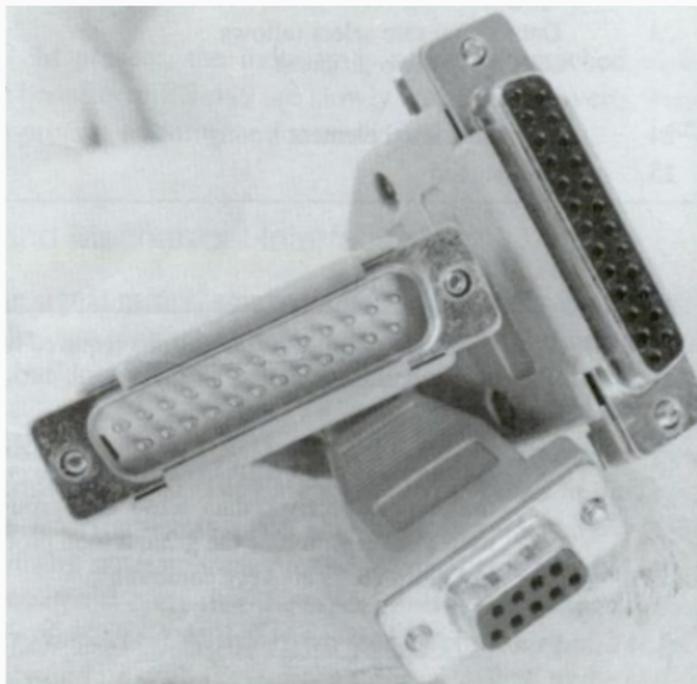
DB-9

### 9-Pin Connector

- Pin 1 – Data Carrier Detect (DCD)
- Pin 2 – Received Data (RxD)
- Pin 3 – Transmitted Data (TxD)
- Pin 4 – Data Terminal Ready (DTR)
- Pin 5 – Signal Ground (GND)
- Pin 6 – Data Set Ready (/DSR)
- Pin 7 – Request to Send (/RTS)
- Pin 8 – Clear to Send (/CTS)
- Pin 9 – Ring Indicator (RI)

A transmissão Full Duplex ocorre quando o dado é transmitido (ou pode ser transmitido) simultaneamente por ambos os dispositivos. Uma fiação especial é necessária para este fim.





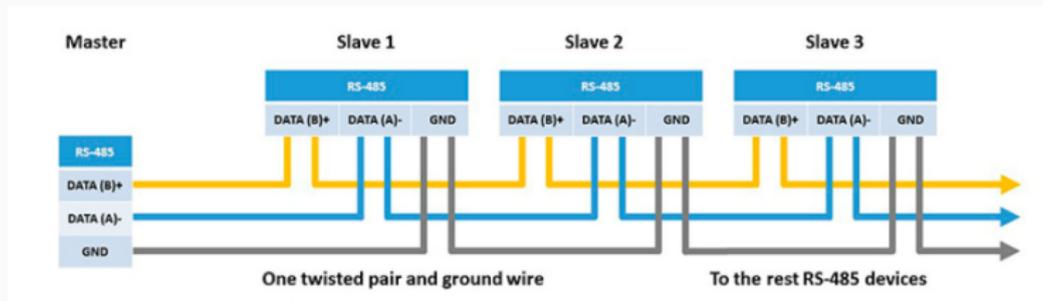
**RS-485**

---

- ▶ O que é o RS-485?
  - ▶ O RS-485 é um padrão de interface da EIA (Electronic Industry Associates);
  - ▶ o RS-485 suporta uma linha de transmissão balanceada que pode ser compartilhada;
  - ▶ Ele permite altas taxas de transmissão em longas distâncias no mundo real;
  - ▶ O padrão apenas especifica as características elétricas do driver e receptor. Ela não especifica e nem recomenda qualquer protocolo.
- ▶ Quão rápido pode ser o RS-485?
  - ▶ O RS-485 foi projetado para maiores distâncias e maiores taxas de transmissão em comparação ao RS-232;
  - ▶ De acordo com o padrão, 10 Mbits/s é a máxima velocidade e a distância pode chegar a até 3.000 metros.
  - ▶ Ele permite que múltiplos receptores estejam conectados de modo linear.

### ► Comunicação

- ▶ O RS-485 permite comunicação Half-Duplex e Multidrop com apenas um par de fios;
- ▶ Para a comunicação Half-Duplex é utilizada uma configuração servidor-escravo;
- ▶ A principal diferença com o RS-232 é que o RS-232 realiza uma conexão ponto a ponto bidirecional, enquanto que o RS-485 é um único bus de comunicação.
- ▶ Eletricamente, cada sinal no RS-232 utiliza um único fio com pontenciais simétricos sobre um fio terra comum. O RS-485 utiliza dois fios para transportar o único sinal diferencialmente.
- ▶ O sinal lógico "1" é representado por uma diferença de potencial entre as duas linhas +2V e +6V, e o sinal lógico "0" é representado pela diferença de potencial entre as duas linhas -6V e -2V.

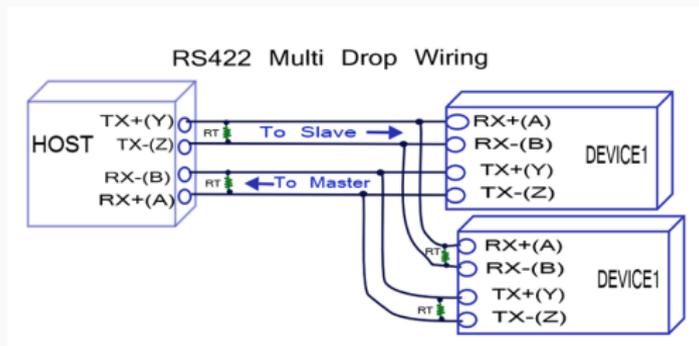


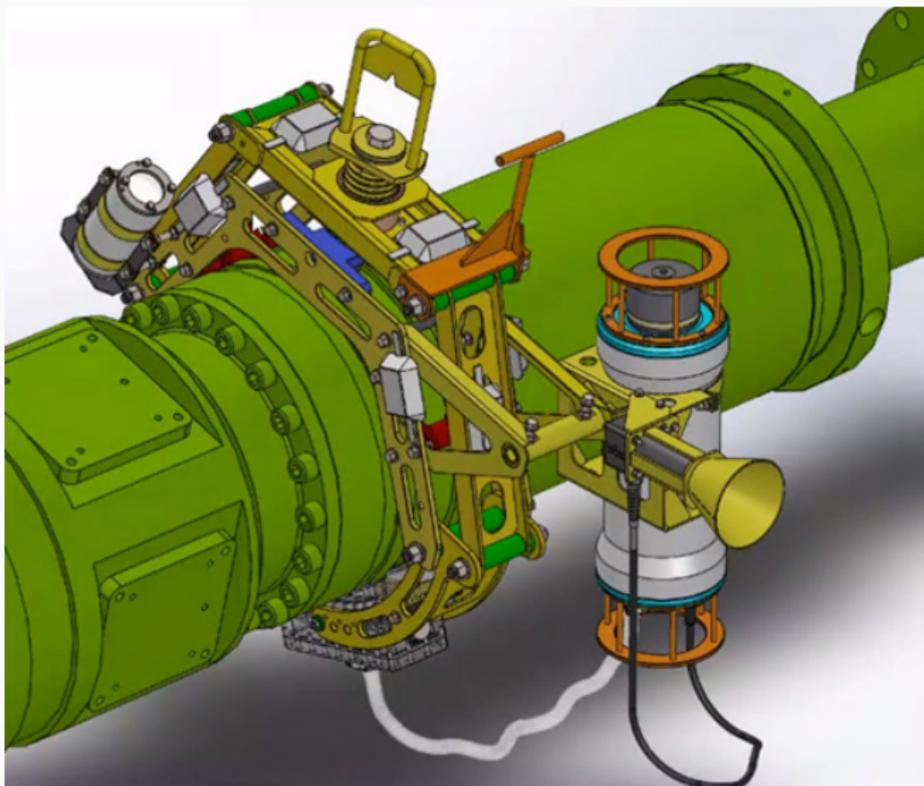
**RS-422**

---

### ► Características

- O RS-422 possui o mesmo princípio que o RS-485. Eles enviam e recebem sinais em modo diferencial e nenhum fio de terra é necessário. A operação diferencial é a razão fundamental para que ocorram transmissões em longa distância.
- o RS-422 consegue operar em modo full-duplex utilizando dois pares de fios. O RS-485 consegue operar apenas em modo half-duplex e não consegue transmitir e receber simultaneamente.



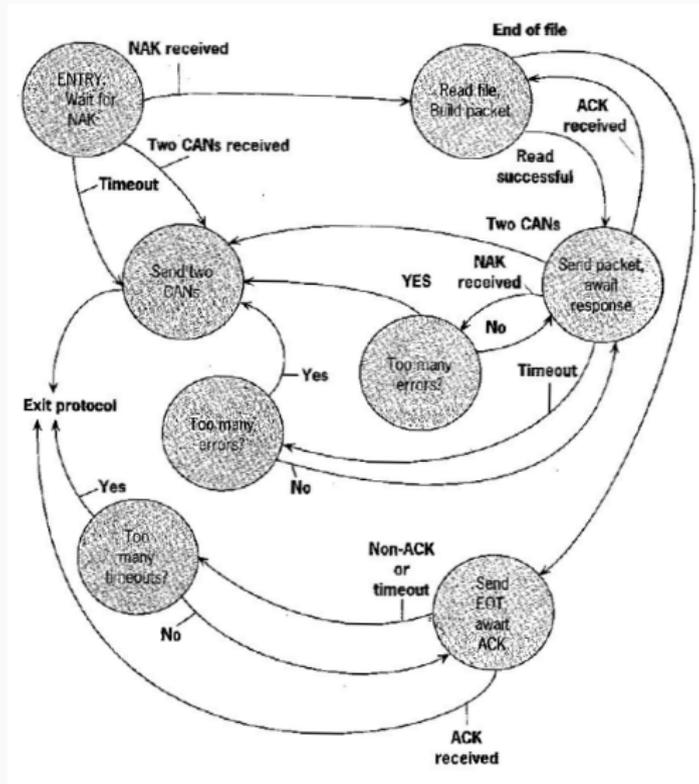


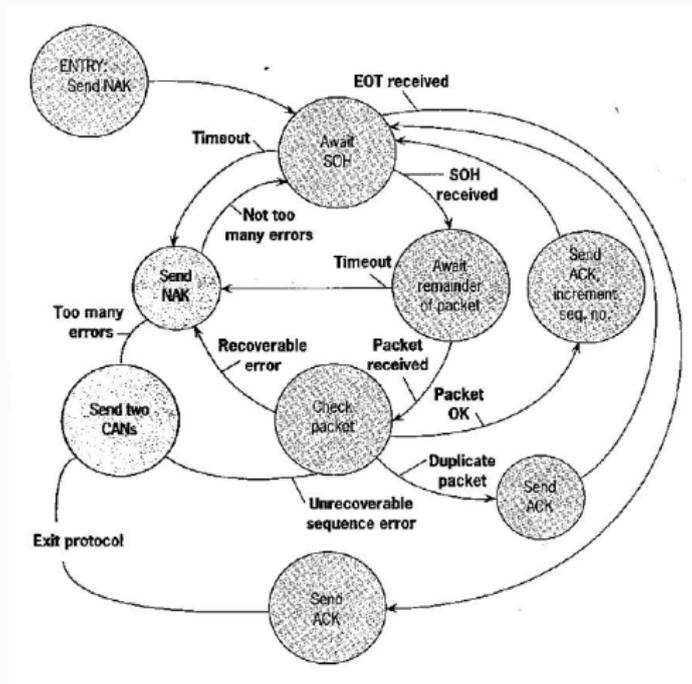
## **Protocolos de Comunicação**

---

## ► Comunicação Assíncrona

- É um protocolo half-duplex, com stop e wait;
- Foi extesivamente utilizado para cominuação entre dois PCs por linha telefônica;
- O Sender envia um frame ao receiver e espera por um frame de ACK;
- O Receiver pode enviar um sinal de Cancel (CAN) para abortar a transmissão;
- Caso o Receiver detecte que o pacote chegou intacto, um ACK é enviado. Em caso contrário um NACK é enviado, informando que ocorreu um erro. Neste caso, o último pacote é retransmitido pelo Sender;
- Este protocolo não é perfeito, pois verifica a paridade dos bits. É possível que dois erros ocorram, sendo que o segundo erro corrige o primeiro, e nenhum erro será detectado.
- Um frame possui: 1 byte SOH, 2 bytes de header, 128 bytes de dado e 1 byte de CRC.
- O SOH é o byte de início;
- O Header contém a sequência com 2 bytes;
- O Dado contém dos dados do pacote;
- O CRC é verificação cíclica de redundância. Este campo verifica se ocorreram erros no pacote.





- ▶ Y-MODEM é uma evolução em relação ao X-MODEM que permite a transmissão de vários arquivos.
- ▶ O Y-MODEM possui 16 bits para verificação de erro (o X-MODEM possui 8 bits).
- ▶ O Y-MODEM permitiu pacote de dados de até 1.024 bytes (eliminando o padding necessário no X-MODEM).
- ▶ O Z-MODEM é muito mais eficiente que o X-MODEM e suporta recuperação e erros.
- ▶ O Z-MODEM não espera por *acks* após o envio de cada bloco. Ele envia rapidamente os blocos em sucessão. Se a transferência for cancelada ou interrompida, ela pode ser recuperada posteriormente sem o re-envio dos pacotes já enviados.

## **CAN - Control Area Network**

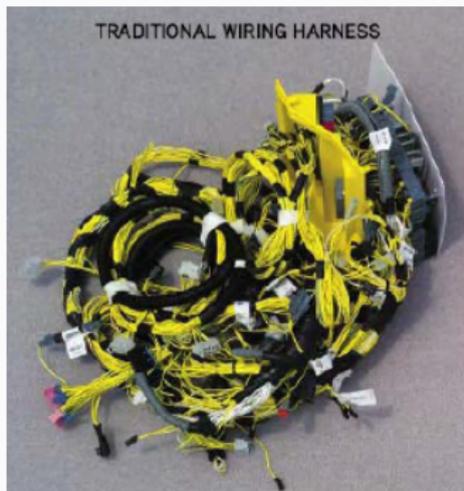
---

- ▶ Simples, robusto e eficiente bus de comunicação serial para redes embarcadas.
- ▶ Desenvolvido pela BOSCH (iniciado em 1983) e apresentado na SAE em 1986.
- ▶ Foi padronizado pela ISO em 1993.
- ▶ O primeiro chip com suporte a CAN foi o Intel 82526 e o Philips 82C200 apresentados em 1987.
- ▶ O primeiro carro a utilizar CAN foi o Mercedes classe S (W140 de 1991).

- ▶ Simples, robusto e eficiente bus de comunicação serial para redes embarcadas.
- ▶ Desenvolvido pela BOSCH (iniciado em 1983) e apresentado na SAE em 1986.
- ▶ Foi padronizado pela ISO em 1993.
- ▶ O primeiro chip com suporte a CAN foi o Intel 82526 e o Philips 82C200 apresentados em 1987.
- ▶ O primeiro carro a utilizar CAN foi o Mercedes classe S (W140 de 1991).

Cabeamento tradicional anterior aos anos 90 em carros de luxo tinham:

- ▶ 30 kg de fios
- ▶ 1 km de fios de cobre
- ▶ 300 conectores, 2.000 terminais e 1.500 fios.
- ▶ Tornando complexa a manufatura, instalação e manutenção. Por exemplo, apenas a porta poderia ter mais que 50 fios.



Proposta multiplexada (por exemplo o CAN)

- ▶ Redução massiva nos custos de cabeamento (por exemplo, a porta ficou apenas com 4 fios)
- ▶ Custo adicional mínimo com a inclusão de controladores CAN, transceivers, etc.
- ▶ Atualmente os dispositivos CAN são chips.



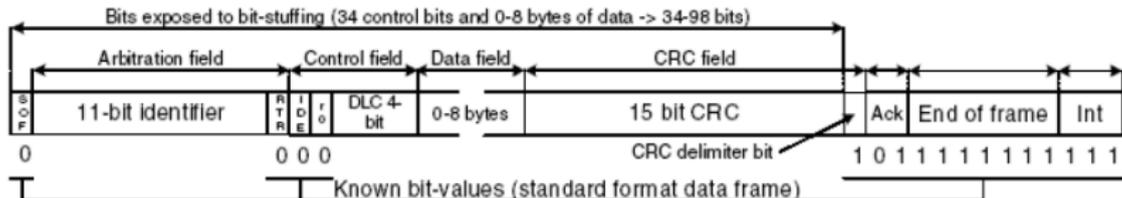
O CAN é um barramento serial multi-master.

- ▶ Colisão é resolvida com base em prioridade
- ▶ A camada física suporta dois estados: “0” dominante e “1” recessivo.

A transmissão da mensagem:

- ▶ Os nós CAN esperam que o barramento fique em “idle” antes de iniciar uma transmissão
- ▶ O sincronismo ocorre com o bit SOF (“0”)
- ▶ Cada nó inicia a transmissão do identificador
- ▶ Se o nó transmitir “1” e observar um “0” no barramento, ele para a transmissão
- ▶ Os nós que completarem a transmissão de seu identificador continuam até o fim da mensagem
- ▶ Deve haver um identificador único para assegurar que todos os nós não gerem colisão.

## CAN - Control Area Network - Formato do Pacote



- ▶ Start of Frame (sincronismo)
- ▶ Identificador que determina a prioridade de acesso para o barramento (bits 11 a 29)
- ▶ Campo de controle (código de comprimento do dado)
- ▶ Bytes 0-8 de dados
- ▶ CRC com 15 bits
- ▶ Campos de Acknowledgement
- ▶ End of frame marker
- ▶ Inter-frame space (3 bits)

## Identifiers

11001000110

CAN controller 1

11011000111

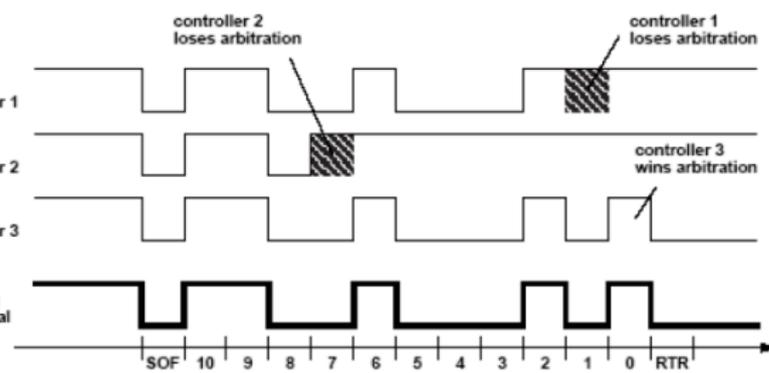
CAN controller 2

11001000101

CAN controller 3

11001000101

resulting bus signal



**The End!**