Aula 19 – Comunicação Serial

**1.1)** No Brasil, a maioria dos carros são feitos a partir do protocolo CAN, mas outros protocolos mais complexos são utilizados, como: FlexRay, bluetooth, MOST entre outros.

**1.2)** [ISA](https://pt.wikipedia.org/wiki/Industry_Standard_Architecture) - *Industry Standard Architecture*)

[ATA](https://pt.wikipedia.org/wiki/ATA) - *Advanced Technology Attachment*

[SCSI](https://pt.wikipedia.org/wiki/SCSI) - *Small Computer System Interface*

[PCI](https://pt.wikipedia.org/wiki/Peripheral_Component_Interconnect) *- Peripheral Component Interconnect*

**1.3)** 24 vias, sendo 16 para endereçamento e 8 para os dado.

Precisamos de 24 vias(bits para permitir a seleção do endereço (32kB = 32768B = 0b1000000000000000)

**1.4)** Quando uma sequência serializada de bytes é convertida em um objeto, chamamos de desserialização, ou seja, passa de serial para paralelo, representando a mesma informação

**1.5)** A penalidade é por conta que é enviada uma palavra por vez, sendo ela com 8 bits.

**1.6)** USB - *Universal Serial Bus*

RS-232 - Padrão de protocolo para [troca serial](https://pt.wikipedia.org/wiki/Comunica%C3%A7%C3%A3o_serial) de dados binários, entre DCE (*Data Communication Equipment*) e DTE (*Data Terminal Equipment*).

RS-485 - Padrão define as características eléctricas dos condutores e receptores para utilização em [comunicação serial](https://en.wikipedia.org/wiki/Serial_communication).

Ethernet - Arquitetura de interconexão para redes locais baseada em envio de pacotes.

**1.7)** Little Edian = UBS, Ethernet, RS-232

Big-Endian: é a forma de organizar dados pela ordem. Os primeiros dígitos a serem representados são os de maior peso.

**1.8)**

**a)** UART – Universal Asynchronous Receiver/Transmitter – Assíncrono

**b)** USART – Universal Synchronous/Asynchronous Receiver/Transmitter - Síncrono

**c)** USB - Assíncrono

**d)** PCIe - Síncrono

**e)** SPI – Síncrono

**f)** I2C - Síncrono

**2.1)** O RS-232 possui várias aplicações como em conexão de mouse, impressora, modem, até em instrumentação industrial.

**2.2)** Bit-Rate : Mede o número de bits de dados transmitidos em um segundo num canal de comunicação, por exemplo: 2400 bps (bits por segundo) é a mesma coisa que 2400 “zeros” ou “uns” que podem ser transmitidos por segundo.

Baud-Rate: Mede o número de vezes que o sinal num canal de comunicação varia o estado. Utilizando o mesmo exemplo (2400 bps), isso significa que o canal pode mudar de estado 2400 vezes por segundo

A principal diferença entre eles é que a mudança de estado pode transmitir um bit dependendo da técnica de modulação usada.

**2.3)** Par = 0; Ímpar = 1

**2.4)** 1 11111100 0 0

   StartBits

   Dados(0xFC)

   Paridade (Par)

   StopBits

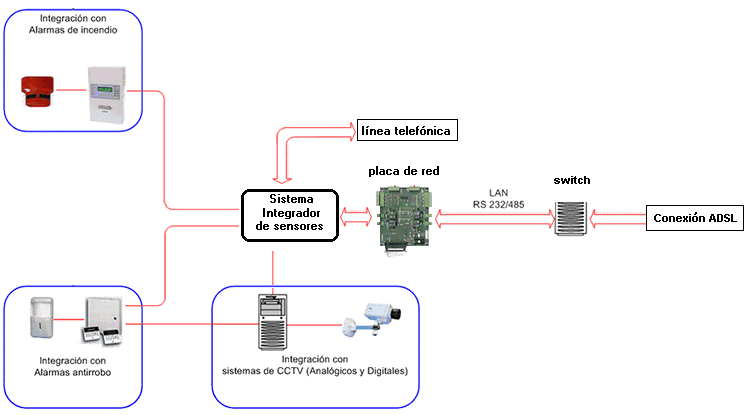
**3.1)** O micro possui 2 periféricos: RS-232 e RS-485.

**3.2)** O protocolo RS232 começa com um start bit em nível lógico baixo, transmitindo dados, logo após é enviado bit a bit para o MSB, por fim o stop bit é enviado em nível alto

Já o RS-485 tem como enfoque a comunicação em rede, com um par de fios é feita a comunicação com vários equipamentos utilizando o mesmo barramento, a linha de dados é balanceada, permitindo comunicação a distância.

Aula 20 – UART

**1.1)**



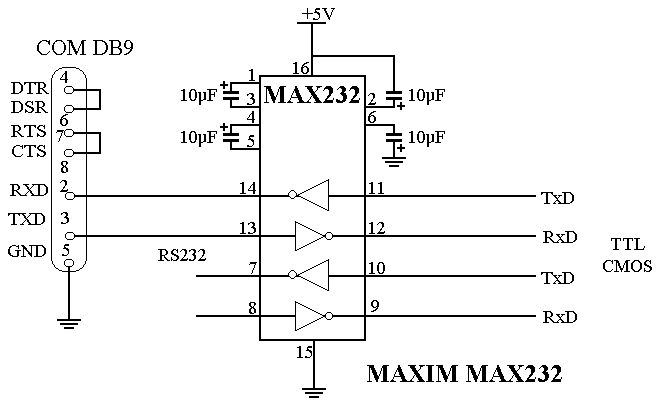
**1.2)** São eles:

* URXD0: Receptor da UART0
* UTXD0: Transmissão da UART0
* URXD1: Receptor da UART1
* UTXD1: Transmissor da UART1

**1.3)** São eles

* CTS: Envie os dados (modem)
* DCD: Portadora detectada
* DSR: Conjunto de dados pronto
* DTR: Terminal de dados pronto
* FG: (Frame Ground)
* RD ou RX: Recepção de dados
* RTS: Pronto para enviar(computador)
* RI: Indicador de telefone tocando
* SG ou GND: Terra
* TD ou TX: Transmissão de dados

**1.4)** Pode ser utilizado o CI MAX232.



**2.1)** Se todos os dados fossem transmitidos como ASCII o micro não suportaria, pois seu valor máximo é de 255.

**2.2)** É um número que o receptor utiliza para confirmar a sequência enviada pelo emissor.

**3.1)** O micro possui 2 periféricos: RS-232 e RS-485. São os ids UART0 id 8 e UART1 id9

**3.2)** RS-232 (MAX232):

* Pino 11: Transmissor 1 entrada
* Pino 12: Receptor 1 saída
* Pino 10: Transmissor 2 entrada
* Pino 9: Receptor 2 saída
* Pino 14: Transmissor 1 saída
* Pino 13: Receptor 1 entrada
* Pino 7: Transmissor 2 saída
* Pino 8: Receptor 2 entrada

**3.3) \* possíveis erros na recepção de dados e como eles são gerenciados.**

**3.4)** O transmissor controla o pino UTXD através do baud rate (número de vezes que o canal muda de estado), inserindo os bits de acordo com as configurações feitas no periférico, esse dado é escrito no UART\_THR e copiado para o buffer interno, onde será usado na transmissão. Um novo bit só pode ser escrito, quando o TXRDY indicar com um nível lógico alto.