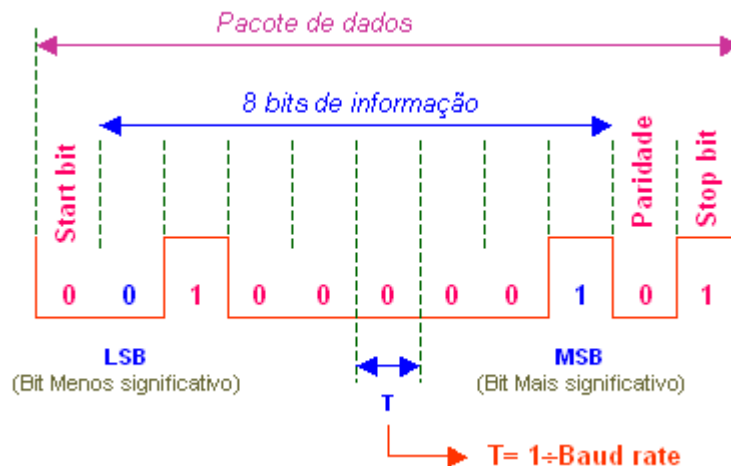


Comunicação de dados

Introdução

Os microcontroladores AVR estão equipados com dois pinos destinados à comunicação serial, UART (**U**niversal **A**synchronous **R**eciever/**T**ransmitter), ou seja, permitem trocar informações entre o PC e o microcontrolador.

A transmissão de dados entre o PC e o microcontrolador é série e assíncrona, série que significa que os bits são enviados um após o outro e assíncrona significa que não há sinal de clock. Para transmitir um byte são necessários 11 bits, sendo **Start bit**, **o pacote de dados**, um **bit de paridade** e **Stop bit**, como você pode ver na figura abaixo.



Taxa de Transferência (Baud Rate)

Baud rate é a velocidade com que os dados transmitidos por um canal de comunicação, ele é medido em transições elétricas ou pulsos por segundo. NA norma EIA232 especifica uma taxa máxima de transferência de dados de 20.000 bits por segundo. Valores mais comuns são:

Bits por segundo (baud rate)	Tempo de transição
110	9,09ms
300	3.33ms
1200	833,33µs
2400	416,67µs
4800	208,33µs
9600	104,17µs
19200	52,0µs

Verificação de erros

Paridade⇒ é um bit extra que é anexado ao pacote de dados à ser transferido de uma localidade à outra, ele tem a função de detectar erros momentâneos provocados por ruídos e distúrbios elétricos que podem acarretar mudanças indesejáveis no código. A paridade pode ser ímpar ou par e seu conceito é bastante simples.

A paridade será par quando o número de bits de valor '1' for par; caso contrário, será ímpar. Se o número de bits '1' for nulo (ou seja, caso se trate do binário '0'), a paridade do mesmo será par.

Paridade par (even-parity):

Start bit	Pacote de dados								Paridade par	Stop bit
0	0	1	0	0	0	0	1	0	1	
	Caractere "B" em ASCII									

Start bit	Pacote de dados								Paridade par	Stop bit
0	0	1	0	0	0	1	0	1	1	1
Caractere "E" em ASCII										

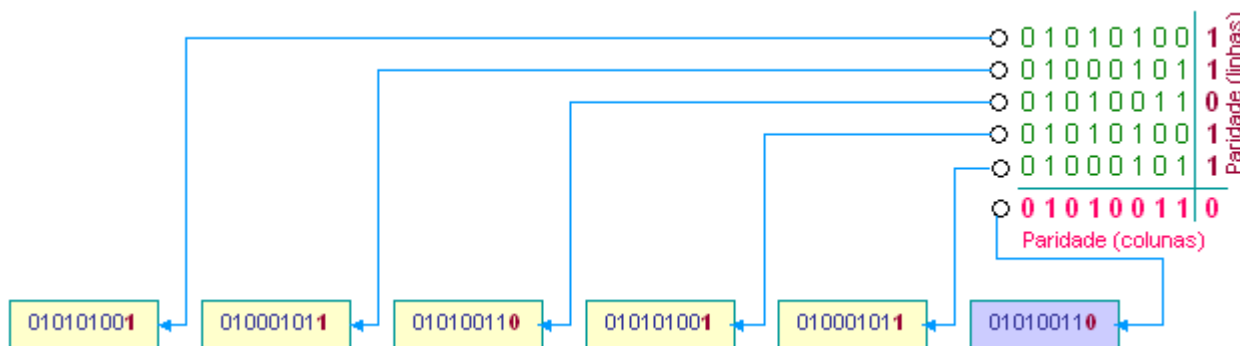
Paridade ímpar (odd parity):

Start bit	Pacote de dados								Paridade ímpar	Stop bit
0	0	1	0	0	0	0	1	0	1	1
Caractere "B" em ASCII										

Start bit	Pacote de dados								Paridade ímpar	Stop bit
0	0	1	0	0	0	1	0	1	0	1
Caractere "E" em ASCII										

A paridade simples, como visto acima, tem limitações quanto à detecção de erros, uma forma de otimizar o método é utilizando a paridade bidimensional. Vejamos:

Exemplo ilustrativo:



A paridade não corrige erros, apenas sinaliza a falha, podendo até solicitar que o pacote seja enviado novamente.

Exemplo com erro

Considere que os pacotes de informação contendo a palavra TESTE sejam enviados, porém, devido a alguma interferência, os pacotes chegaram alterados.

Transmitidos	Pacotes enviados	Paridade Colunas (enviada)
	010101001 010001011 010100110 010101001 010001011	0 1 0 1 0 0 1 1 0



Recebidos	Pacotes recebidos	Nova Paridade Colunas
	011111001 010001011 010100110 010101001 010001011	0 1 1 1 1 0 1 1 0

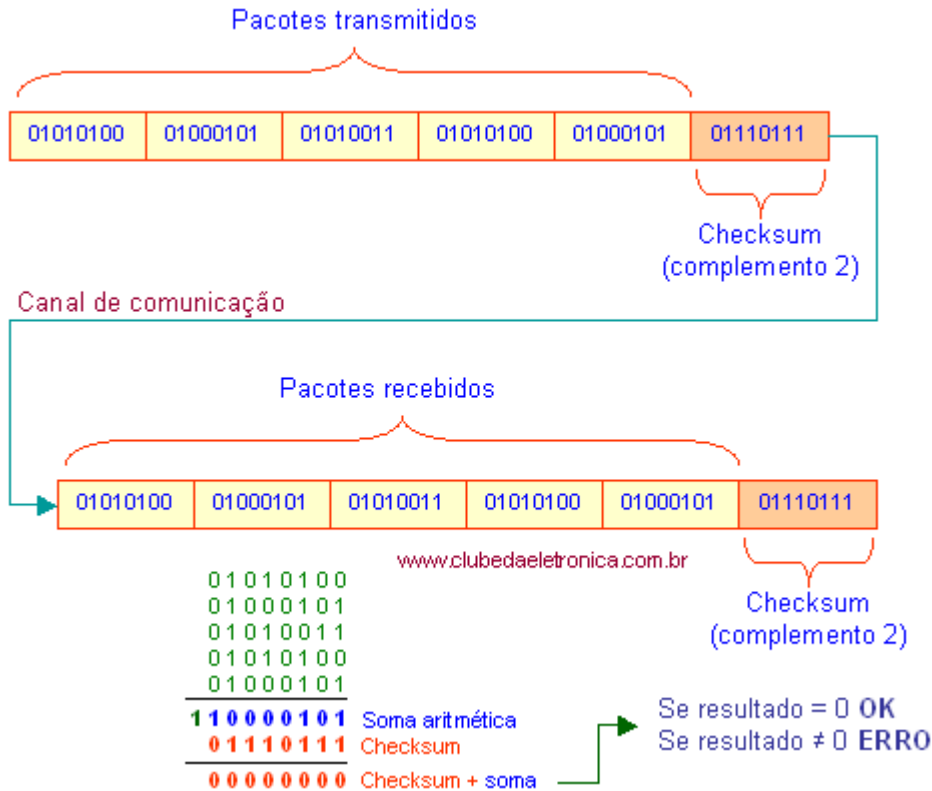
Na paridade das linhas OK (a paridade par foi incapaz de detectar o erro ocorrido, já no primeiro pacote), porém na verificação da paridade (colunas) gerada pelo receptor não condiz com a paridade (coluna) recebida junto com o pacote indicando que um erro ocorreu na transmissão.

Checksum ⇒ é outro método bastante eficiente para verificar a integridade dos dados. Baseia-se na transmissão do pacote juntamente com o resultado da sua soma binária e o código checksum. Quando recebido, o pacote é novamente somado e comparado com o checksum se o resultado for diferente de zero, ocorreu um erro e o pacote deve ser transmitido novamente.

Exemplo:

Caracteres	Código ASCII	Operações executadas
T	0 1 0 1 0 1 0 0	
E	0 1 0 0 0 1 0 1	
S	0 1 0 1 0 0 1 1	
T	0 1 0 1 0 1 0 0	
E	0 1 0 0 0 1 0 1	
	1 1 0 0 0 1 0 1	Soma binária
	0 1 1 1 0 1 1 1	Checksum (complemento 2)
	0 0 0 0 0 0 0 0	Soma +checksum

Exemplo ilustrativo:



Exemplo com erro

Considere que os pacotes de informação contendo a palavra TESTE sejam enviados, porém, devido a alguma interferência, os pacotes chegaram alterados.

Transmitidos	Pacotes enviados	Checksum (enviada)
	01010100 01000101 01010011 01010100 01000101	0 1 1 1 0 1 1 1

↓

Recebidos	Pacotes recebidos	Comparação
	01111100 01000101 01010011 01010100 01000101	(soma) 1 1 0 1 0 1 1 0 1
		(Check) 0 1 1 1 0 1 1 1
	(≠zero) 0 0 1 0 0 1 0 0 1	

Fazendo a soma do pacote recebido com o checksum o resultado foi diferente de zero o que indica erro, assim, o pacote deve ser enviado novamente.

Além da paridade e checksum existe também o **CRC - Cyclic Redundancy Check** e **LRC - Longitudinal Redundancy Check** que não fazem parte deste trabalho, porém, tem o mesmo objetivo, verificar a integridade dos dados.

Vale lembrar que nenhum método é infalível, devemos sim ter um meio eficaz para transmissão de dados, observando que ambientes ruidosos provocam muita interferência.

O padrão RS232

RS-232 é um padrão para troca serial dados entre um DTE (terminal de dados, de *Data Terminal equipment*) e um DCE (comunicador de dados, de *Data Communication equipment*).

Níveis de tensão (padrão RS232)

- ❑ "0" tem um nível de tensão entre -15V e -5V
- ❑ "1" tem um nível entre +5 V e 15 V.

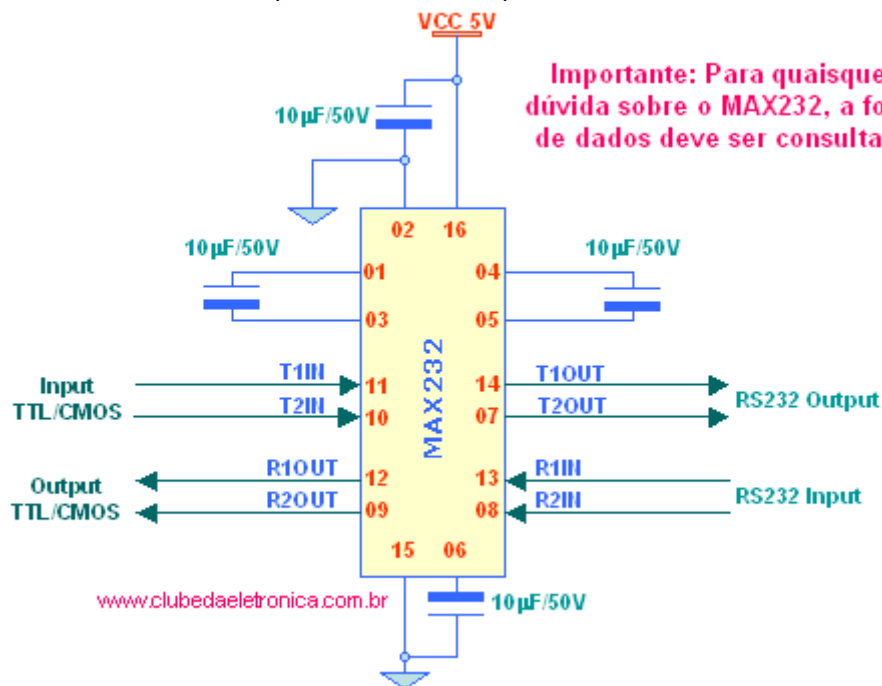
Definitivamente isto não nos ajuda muito porque os microcontroladores AVR utilizam lógica TTL, ou seja:

- ❑ "0" tem um nível de tensão de 0 a 1,8V
- ❑ "1" tem um nível de tensão de 3,5 a 5V

Então, para que possamos traçar informações entre PC e microcontrolador, devemos converter os sinais do padrão RS232 para TTL e vice versa.

Interface para transmissão (MAX232)

É o circuito integrado responsável pela conversão RS232/TTL, ou seja, deve ser utilizado para fazer a conexão entre o microcontrolador e a porta serial do computador.



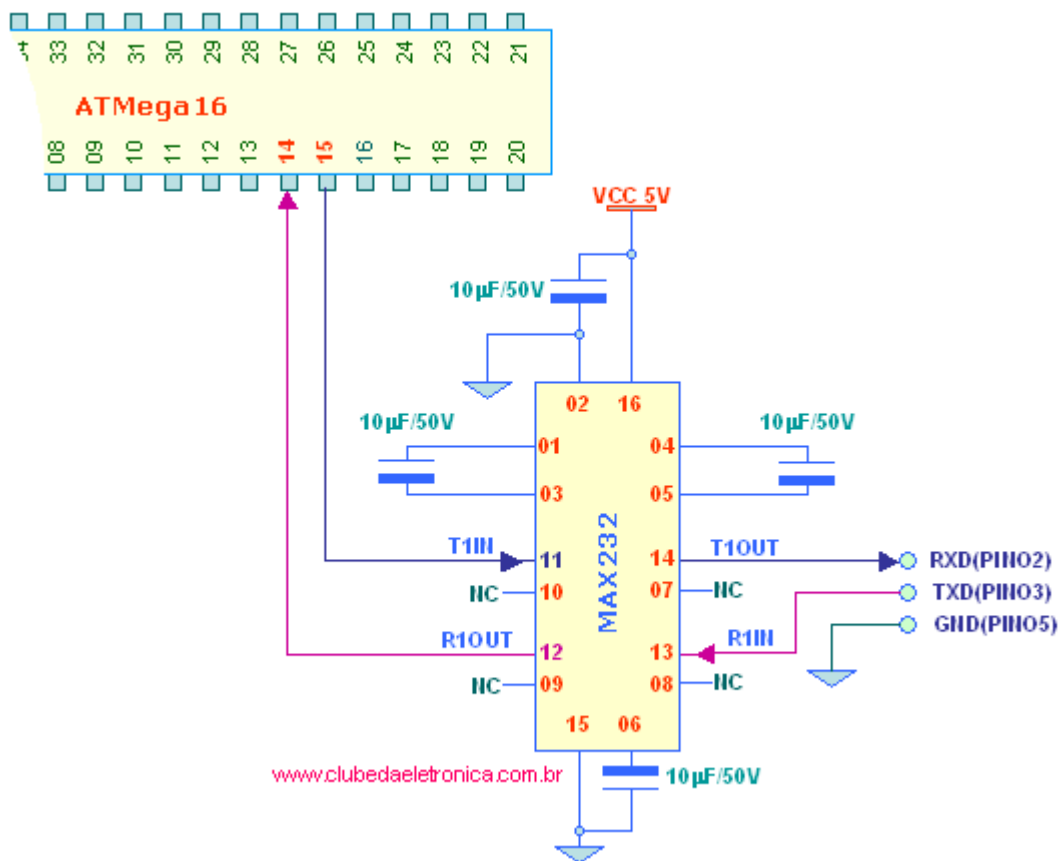
A conexão com o microcontrolador (ATMEGA16)

Os microcontroladores em sua grande maioria disponibilizam terminais para comunicação, com o AVR, não é diferente, em nosso exemplo usaremos o ATMEGA16, porém, o conceito é válido para todos.

Terminais para comunicação do ATMEGA16

- ❑ 14 – RXD (Recebe os dados do computador)
- ❑ 15 – TXD (Envia dados ao computador)





Breve comentário: Os sistemas operacionais ainda suportam as portas seriais, isso porque muitos dispositivos ainda necessitam delas. A porta paralela já praticamente extinta acabou sendo substituída pela USB, meio mais recente e que em breve, muito breve mesmo, substituirá todas as portas do computador. Hoje, já não se compra mais computadores com portas seriais e paralelas, uma pena, pois foi um “hobby” divertido.

Continua ...