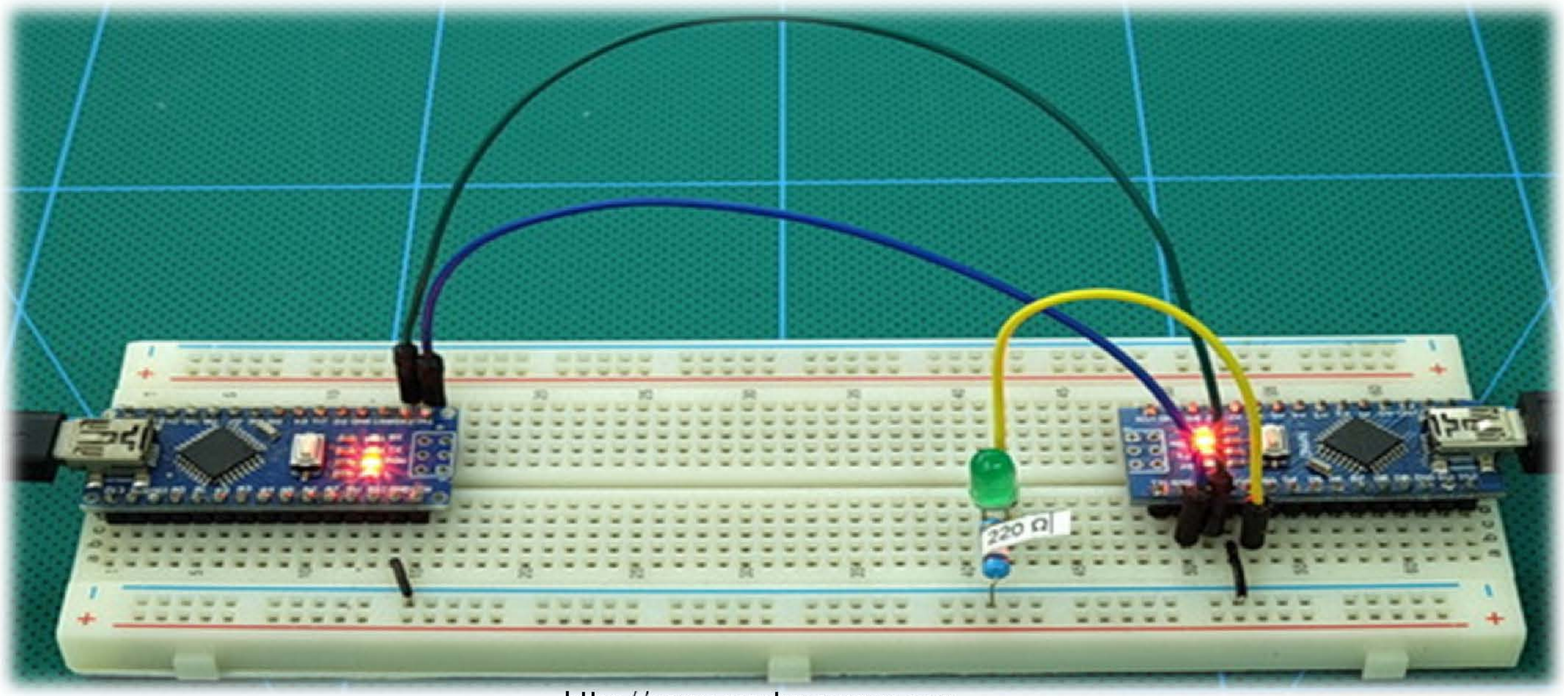
COMUNICACAO SERIAL



http://www.martyncurrey.com

© Prof. Engº esp Luiz Antonio Vargas Pinto

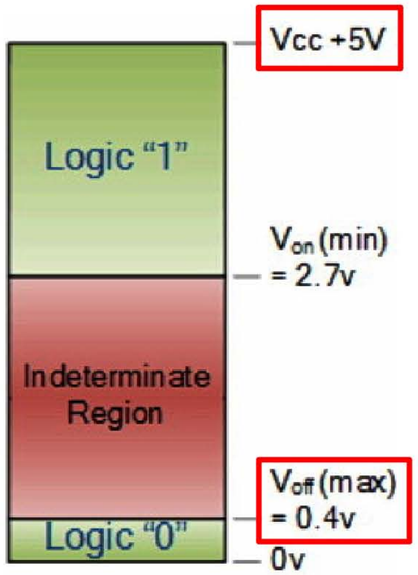
www.vargasp.com

Rev. 2022

© Prof. Vargas

Características

1. Alcance limitado pela distância



2. Sujeito a distorções por ruídos ambientais
3. A Transmissão é diretamente proporcional a potência do sinal e inversamente proporcional ao ruído. Razão para uso de protocolo.

© Prof. Vargas

Canais de comunicação

- Caminhos para o trânsito da informação:
  - Fio elétrico de cobre
  - Ondas de rádio (Wireless)
  - LASER (Light Amplification by Stimulated Emission of Radiation)
  - Futuramente – comunicação quântica

© Prof. Vargas

Comunicação Serial

- Paralela x Serial
  - Taxa de transferência paralela é mais rápida
  - Muitas linhas $\Rightarrow$ **R$** $\uparrow$
- Velocidade de envio (Baud Rate)
  - Taxa em Kbps, Mbps (Bit Per Second)
  - Exemplo:

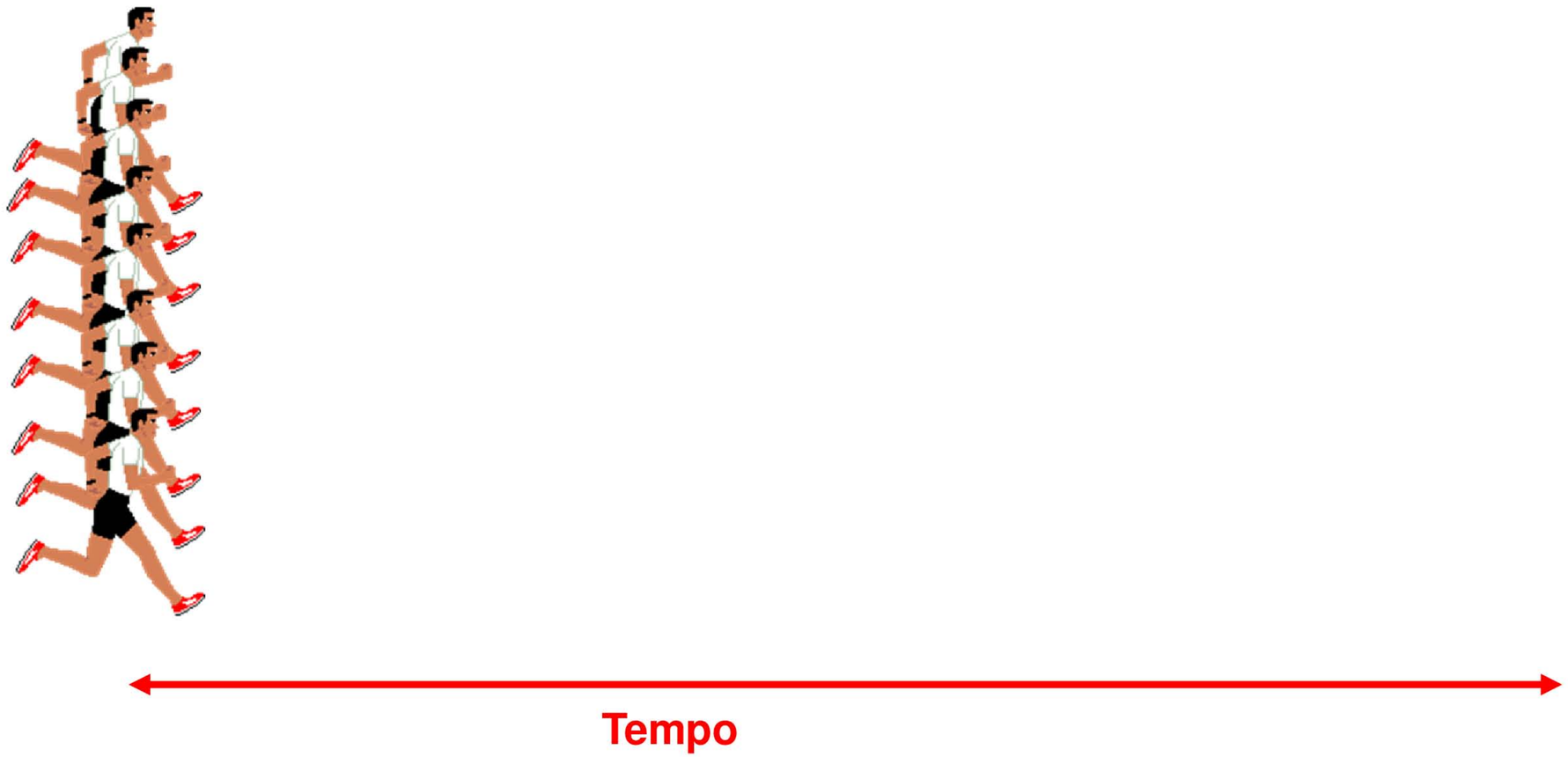
Taxa de 9600 bauds ⇨ 9600 bits por segundo – Crédito a Èmile Baudot inventor do telégrafo

© Prof. Vargas

8 bits simultâneos (paralelo)

© Prof. Vargas

Um bit de cada vez (Série)



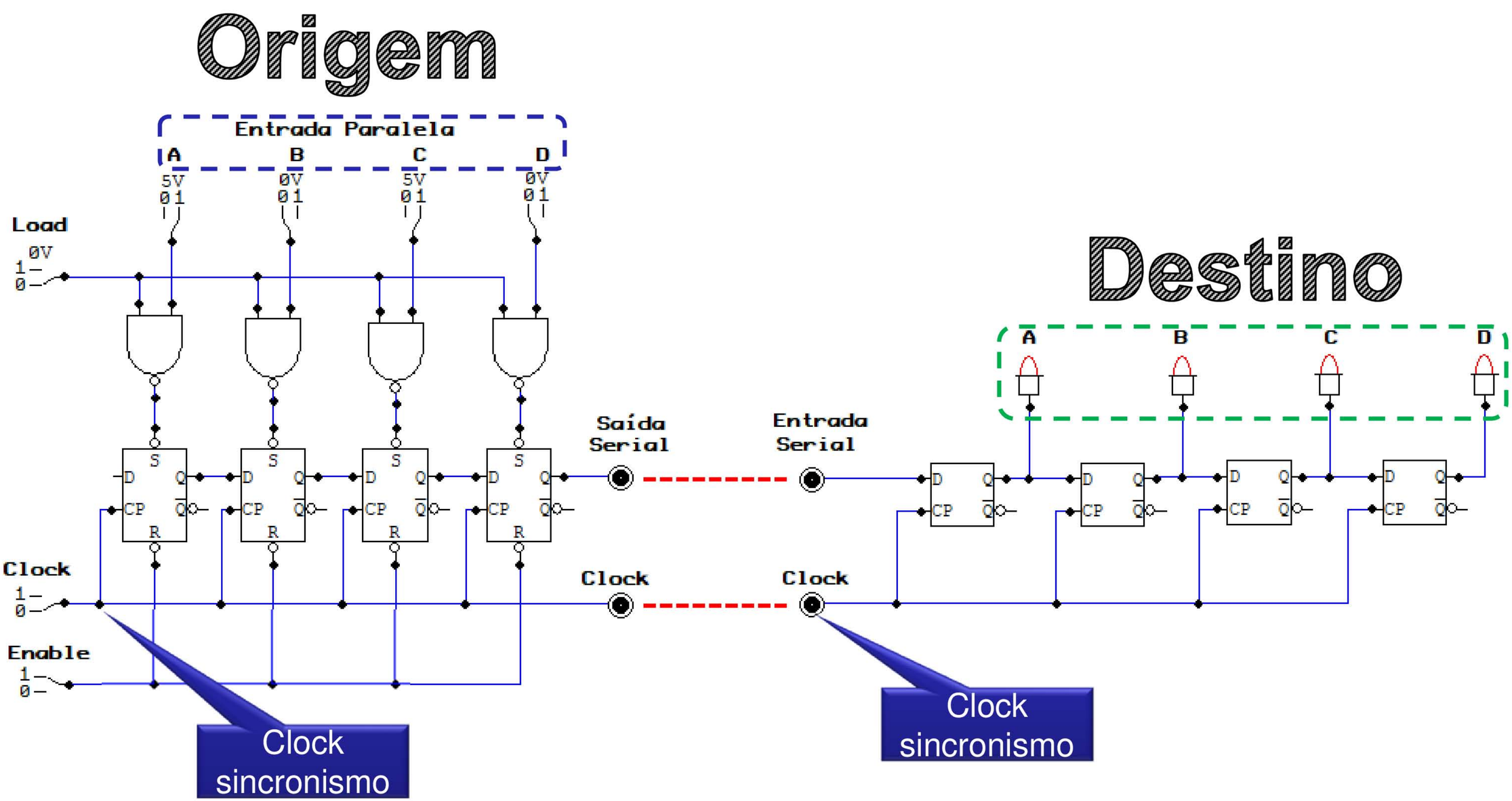
© Prof. Vargas

Comunicação Síncrona

- Dados e Clock em canais separados
- Sincronizado pelo clock
  - receptor armazena o valor do bit coletado naquele momento.
- Aguarda o próximo clock
  - Opcionalmente clock e dados podem usar o mesmo canal – caso dos MODEM

© Prof. Vargas

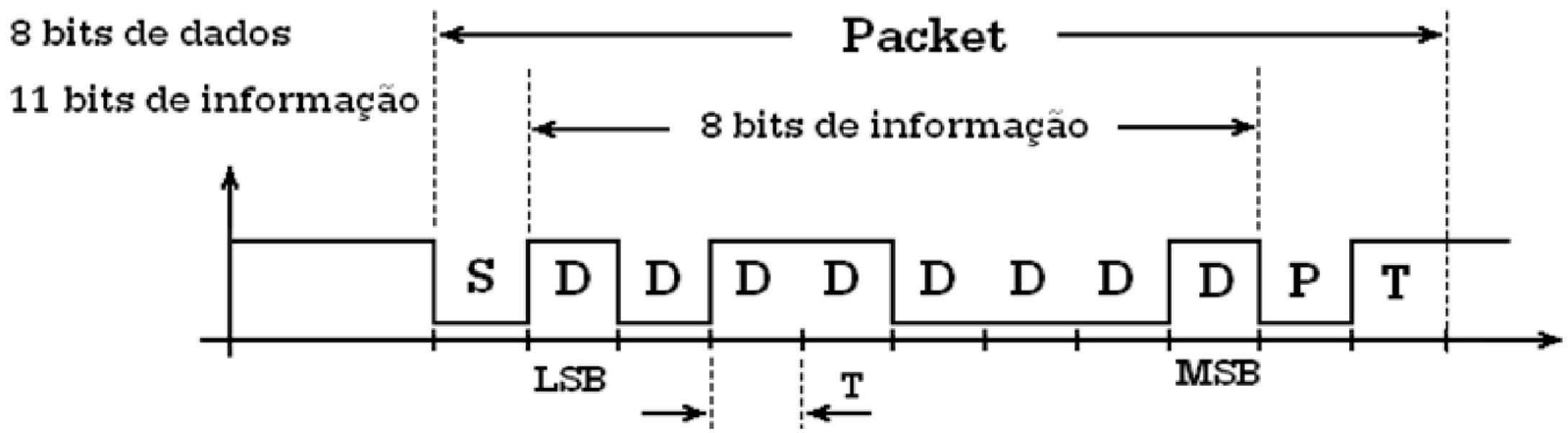
Síncrono



© Prof. Vargas

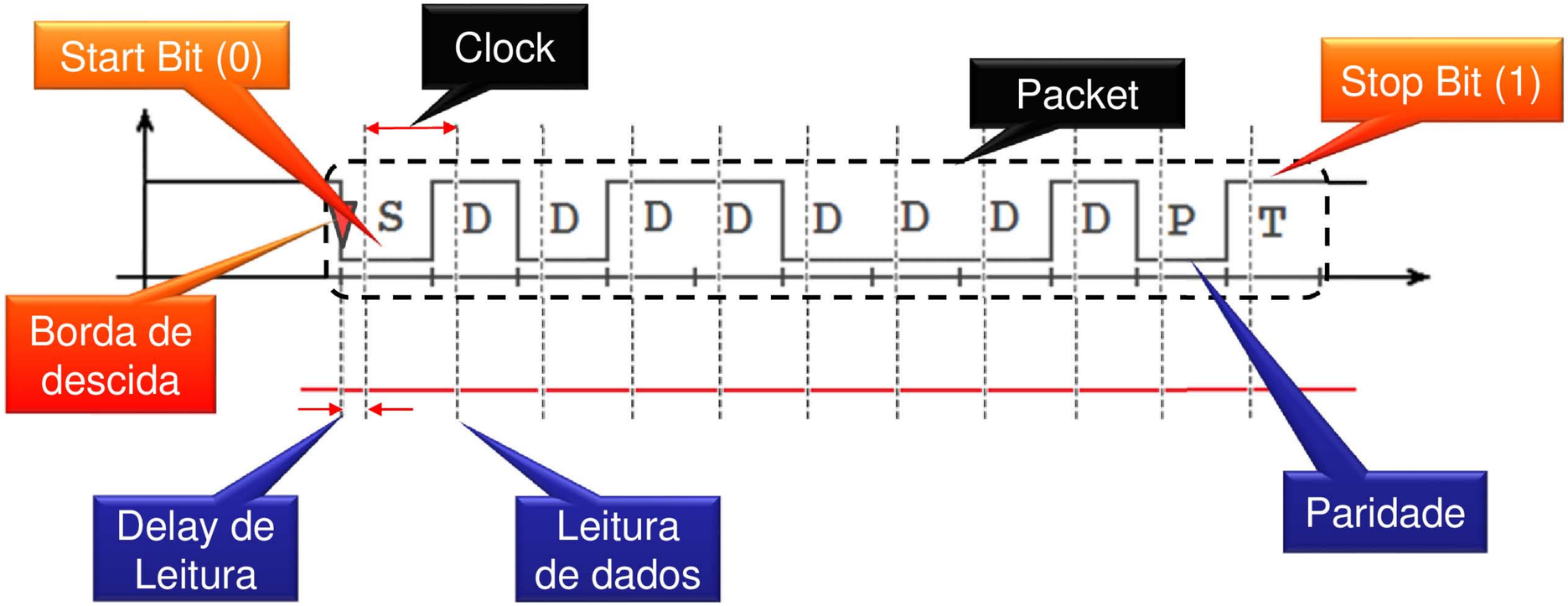
Comunicação Assíncrona

- Dado e clock trafegam em um único canal
- Clock de precisão em ambos Tx e Rx
- Packet mais comum: de 10 ou 11 bits com 8 de mensagem.
  - O comprimento do packet deve ser pequeno para minimizar variações dos osciladores Tx e Rx



© Prof. Vargas

Comunicação Assíncrona



- O packet é concluído com bits paridade e stop bit
- 1 ou 2 bits de erros podem ser corrigidos pelo receptor

© Prof. Vargas

Paridade

- O bit paridade é incluído para detecção de erro
- Em Rx a paridade é recalculada

Dado	Bit de Paridade
10110010	0
10001010	1

© Prof. Vargas

Paridade

- Mas se um número par de bits for trocado, a paridade é mantida e o erro será validado
- "Análises estatísticas de erros de comunicação de dados tem mostrado que um erro com bit simples é muito mais provável que erros em múltiplos bits na presença de ruído randômico. Portanto, a paridade é um método confiável para a detecção de erro."

(CANZIAN, E. **Comunicação Serial RS232**: conceitos básicos sobre comunicação serial. http://www.cnz.com.br. CNZ Engenharia e Informática Ltda)

© Prof. Vargas

Checksum

- **CHECKSUM** é uma técnica que insere na comunicação um byte de verificação
  - Σ (bytes) da mensagem completa em complemento de 2.
- Se a Σ (bytes) + checksum $\neq$ **0** $\Rightarrow$ Erro
  - Na ocorrência de erro é improvável (mas não impossível) que corrupção de dados resultem Σ (bytes) + checksum = **0**
- Um número + (seu negativo) = 0
  - Complemento de um e/ou dois

© Prof. Vargas

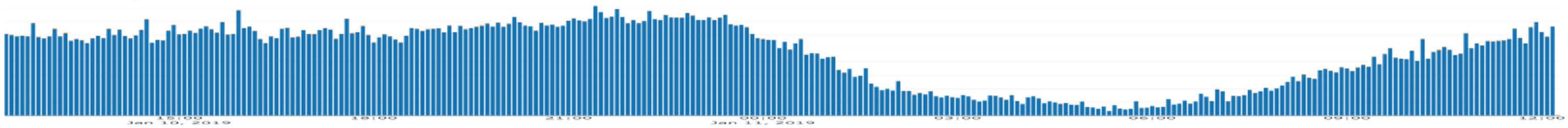
Checksum

```
      1011.0001 ........... 0xB1
      1000.0110 ........... 0x87
      0100.1100 ........... 0x4C
      1111.1111 ........... 0xFF
      1010.0000 ........... 0xA0
  ~~0011~~.0010.0010   Soma Aritmética
      0010.0010   Soma Truncada de 8 bits
  +   1101.1110   Checksum (complemento de 2)

      0000.0000   Soma + Checksum = 0
```

Bytes da mensagem

A correção de erros em uma transmissão, contudo, abaixa a eficiência do canal, e o resultado é uma queda na transmissão.



© Prof. Vargas

A Interface Serial RS-232C

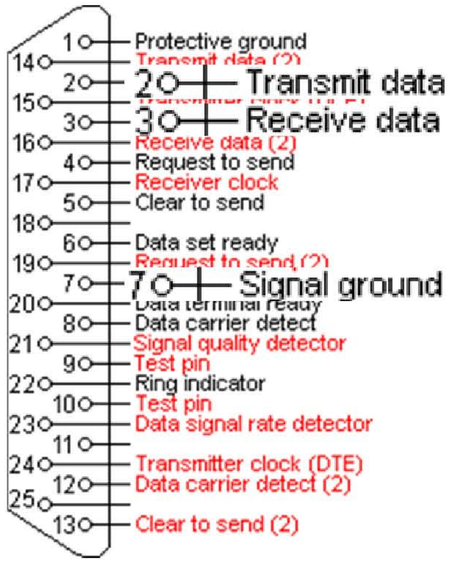
© Prof. Vargas

Interface Serial RS-232C

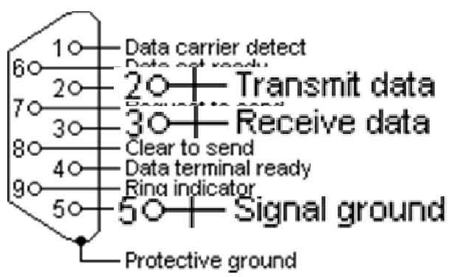
- O termo RS232 (Recommended Standard 232) será utilizado para fazer referência à interface de comunicação.
- O termo EIA232 será utilizado para fazer referência à norma estabelecida pela EIA (Electronics Industries Association).

© Prof. Vargas

Conectores



DB25

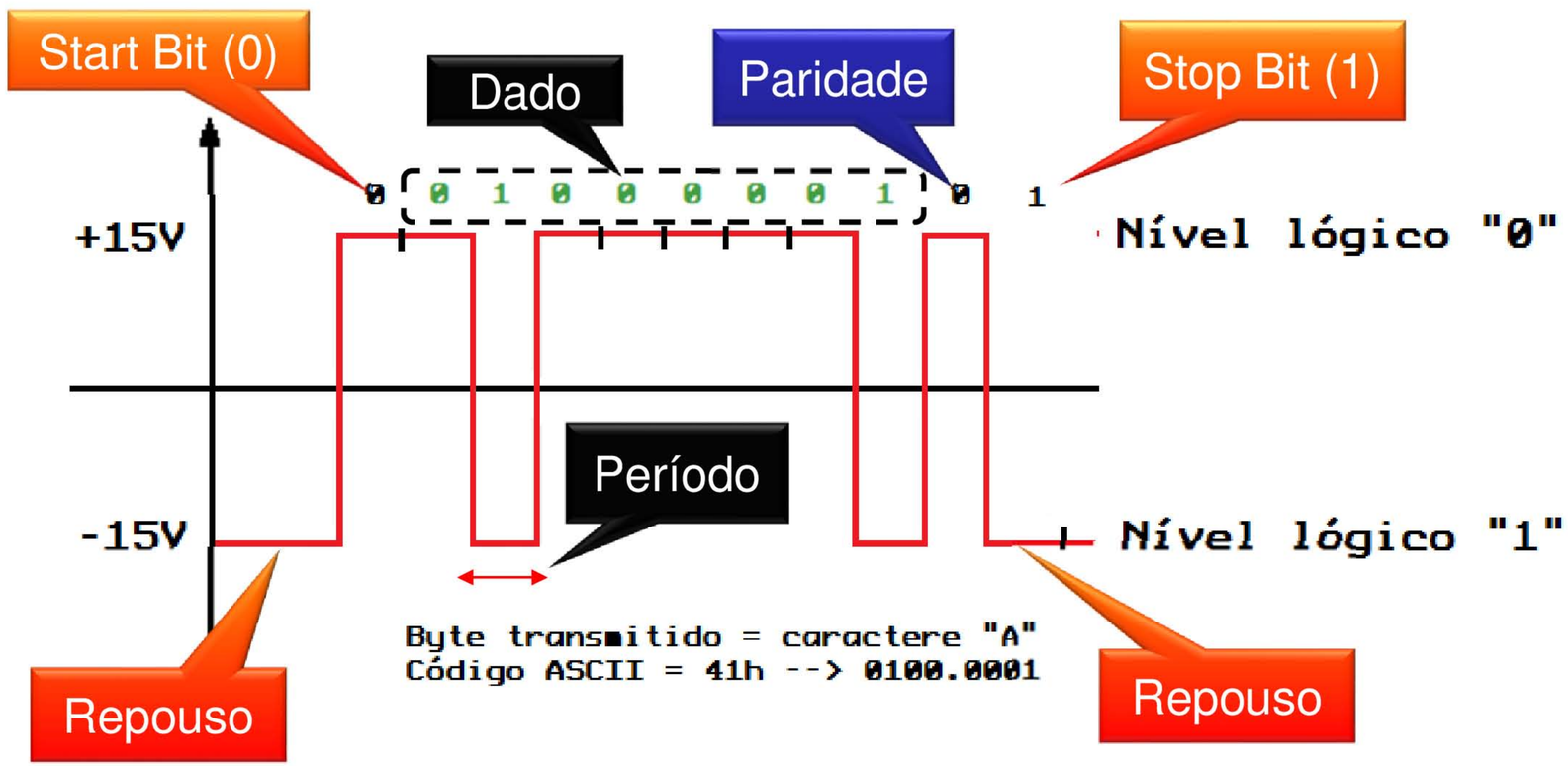


DB9

© Prof. Vargas

Níveis de tensão

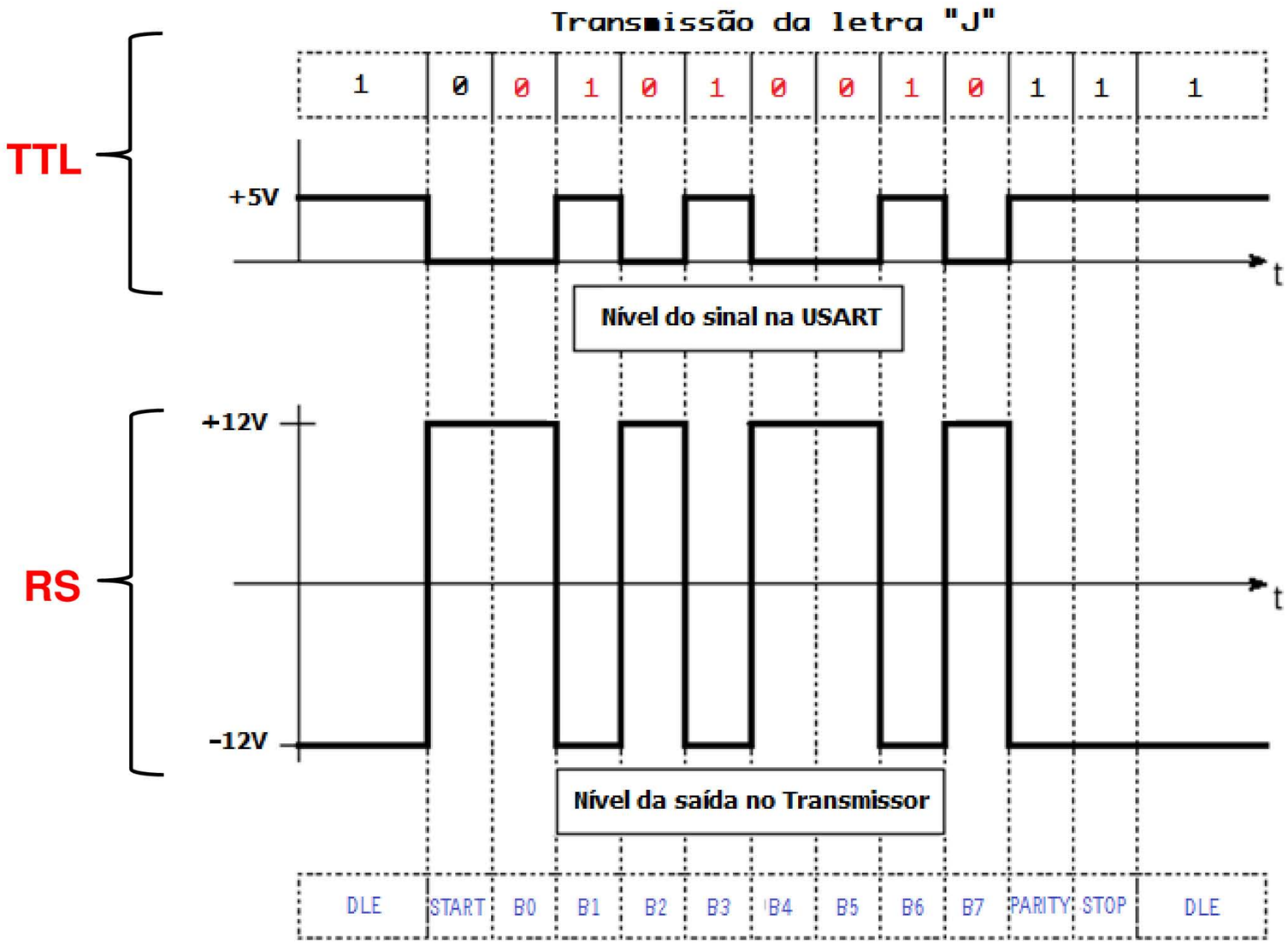
- Converte os níveis de tensão da USART (0 - 5 V) TTL para os níveis utilizados na RS232 (-15 a +15 V)



© Prof. Vargas

Níveis de tensão

- Atualmente o padrão aceita de -12V a +12V

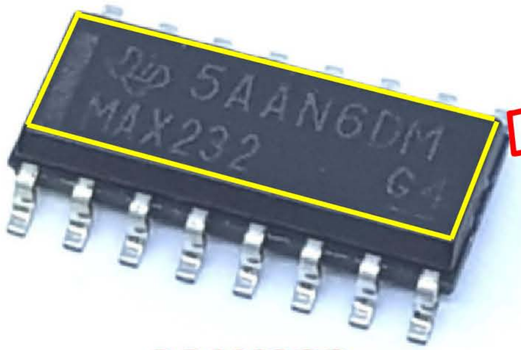


© Prof. Vargas

MAX-232

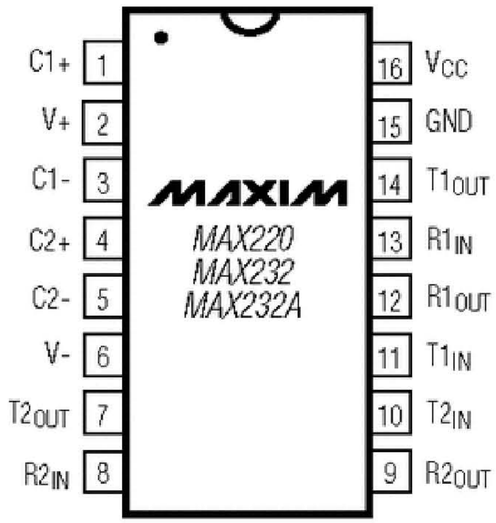
- A adequação de níveis de tensão podem ser obtidos com o uso do CI MAX232

DIP



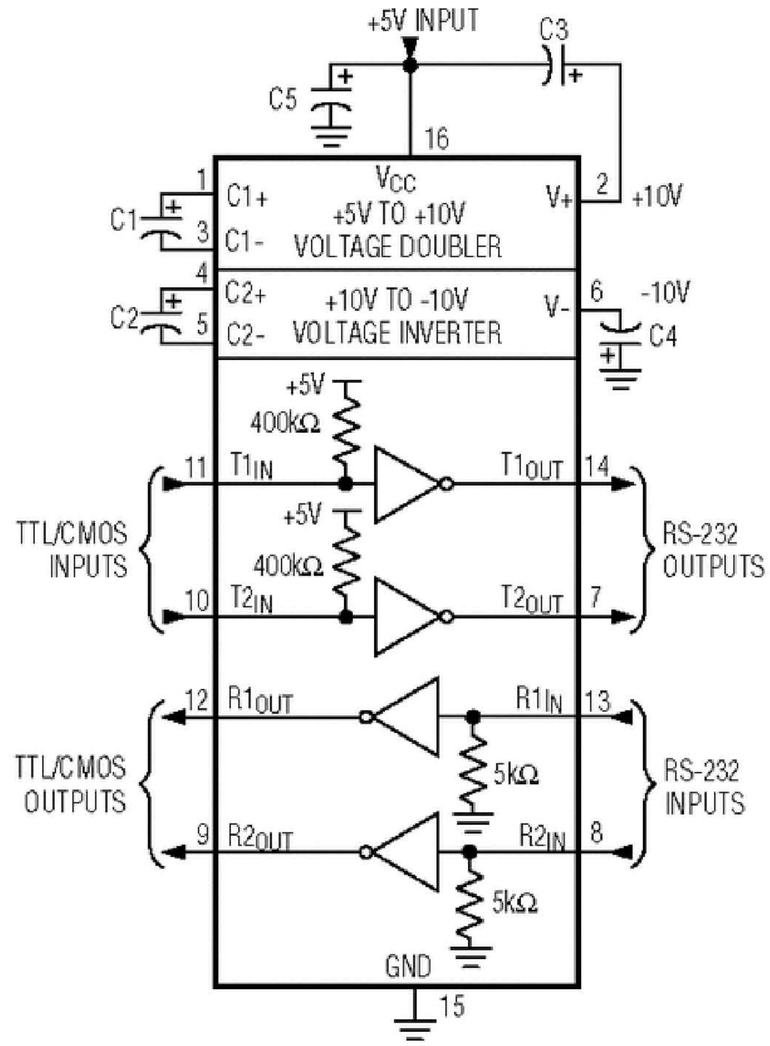
MAX232

SMD



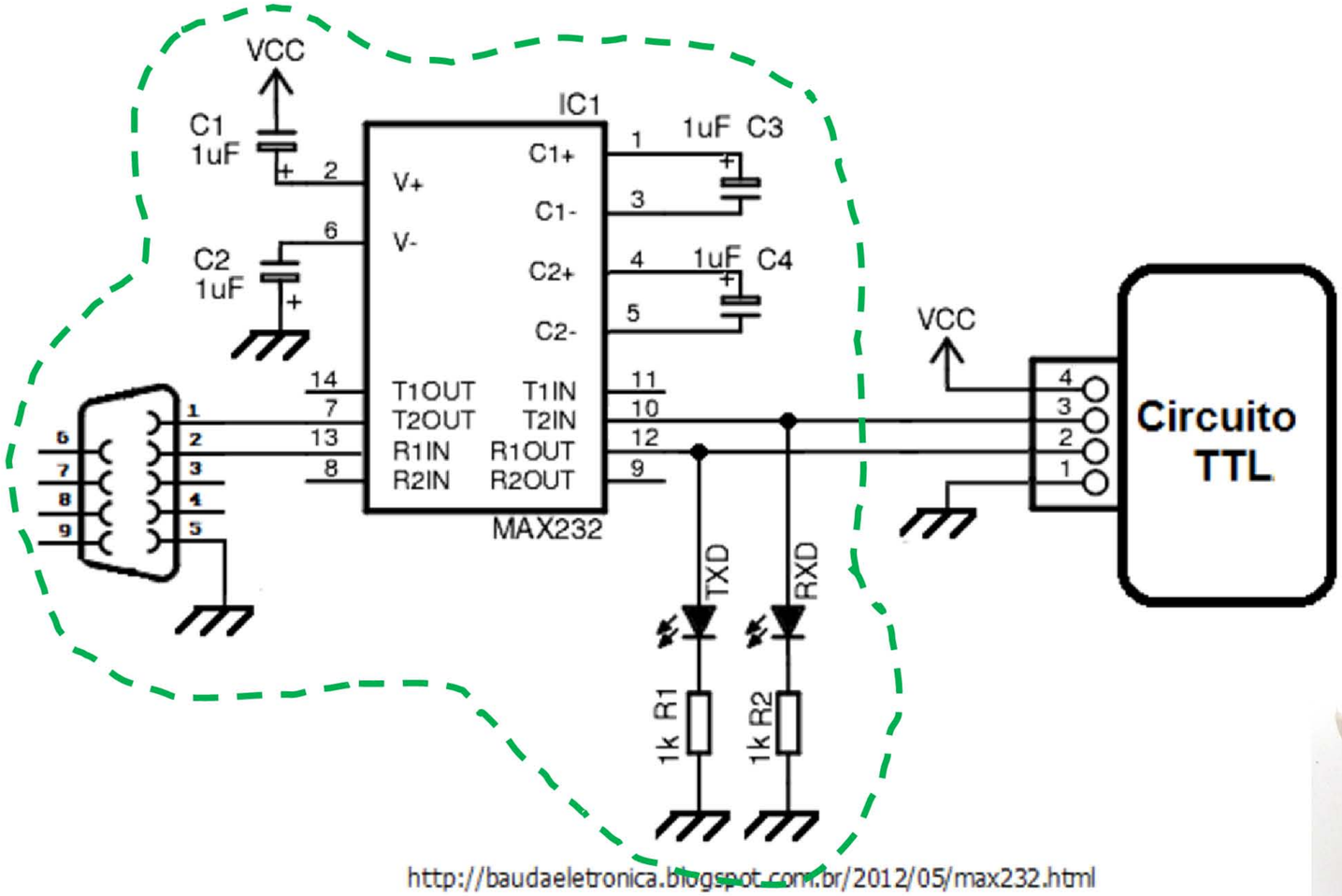
DIP/SO

CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

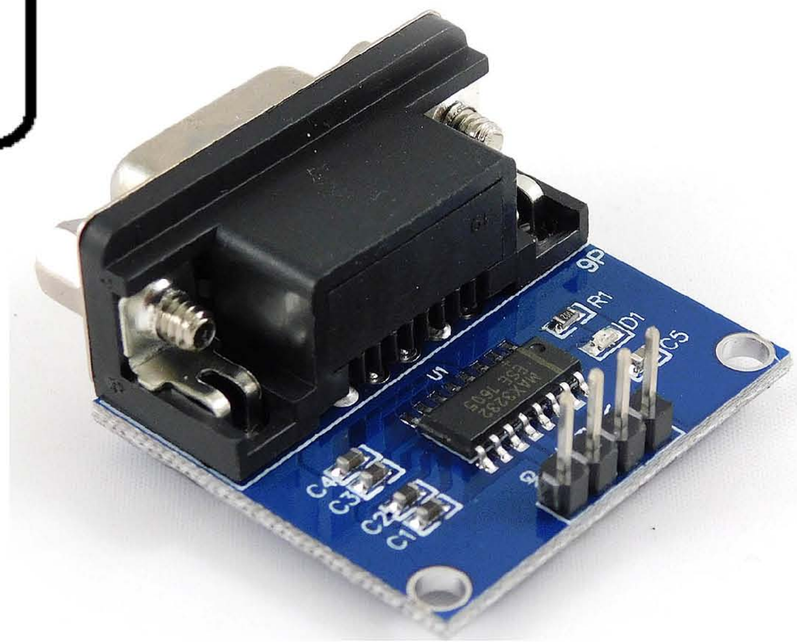


© Prof. Vargas

Aplicação



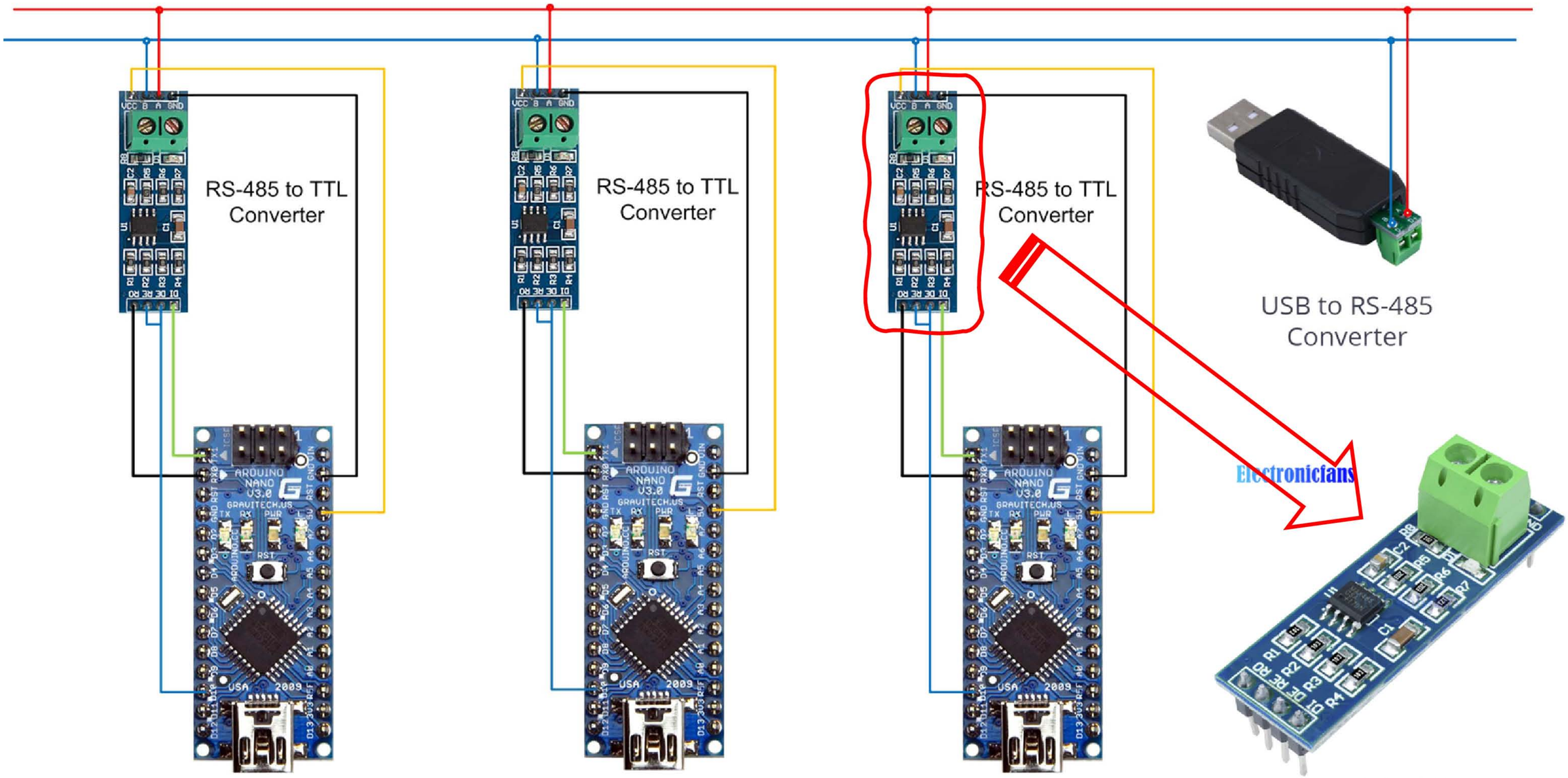
http://baudaeletronica.blogspot.com.br/2012/05/max232.html



© Prof. Vargas

RS-485

- Na indústria é mais comum o uso de RS-485



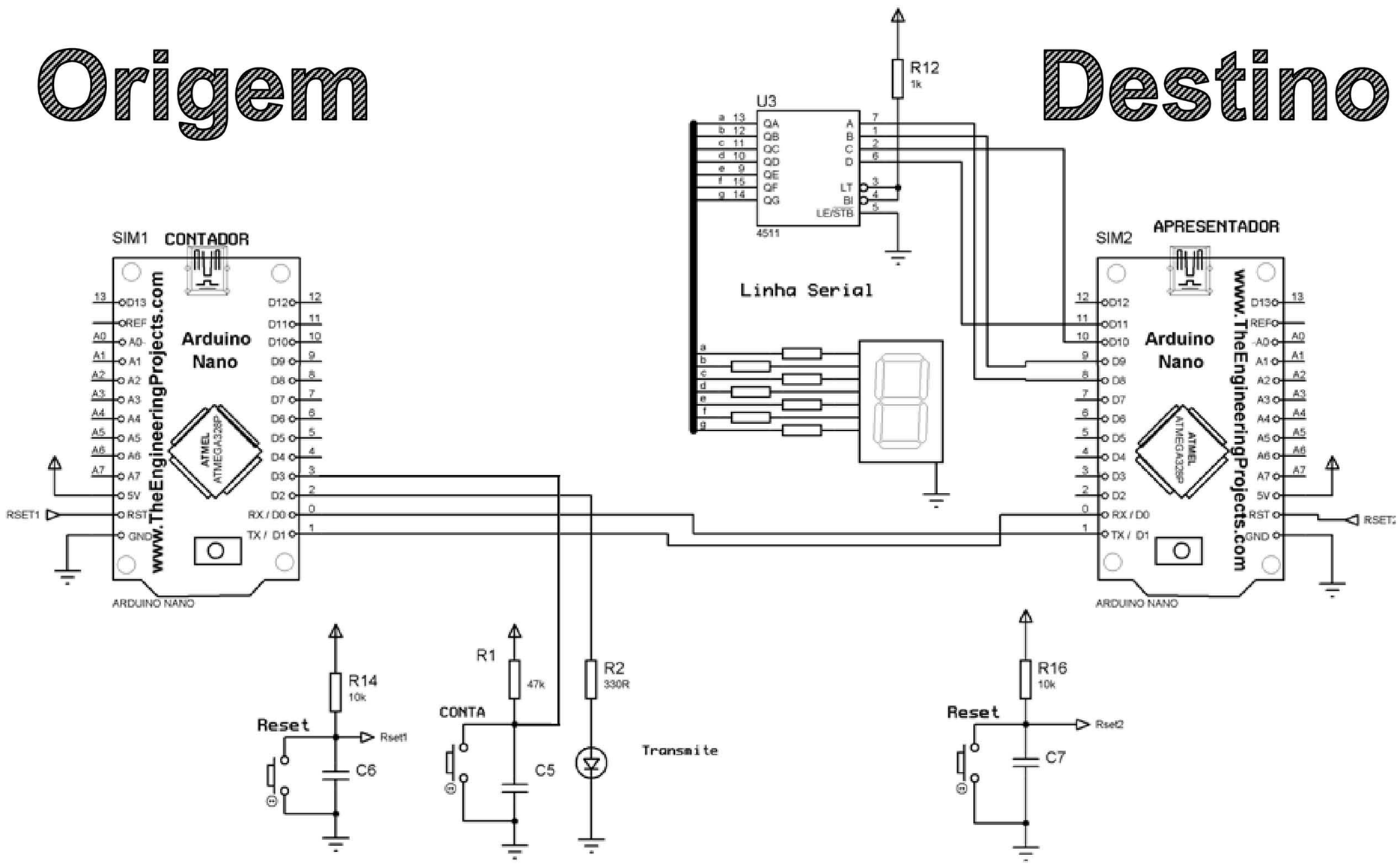
https://hubot.cl/product/modulo-transceptor-rs485-arduino-raspberry/

© Prof. Vargas

Serial com o Arduino nano

© Prof. Vargas

Circuito de teste



© Prof. Vargas

O Transmissor TX

```
1  /*-----------------------------------------------------------------
2    Programa : TEX.ino
3    Data     : 30/06/2020
4    Autor    : Prof. Vargas
5    Descrição:
6   -----------------------------------------------------------------*/
7  const int Led_transmite = 2;   //Batizamos o port de I/O digital 2 de Led
8  const int Sw_conta = 3;        //Batizamos o port de I/O digital 3 de Switch
9
10 int Botao = 0;    //Variável para receber o estado do botão
11 int contador = 1;
12
13 void setup()
14 {
15     Serial.begin(9600);
16     pinMode(Led_transmite,OUTPUT);   //Definimos Led como Output.
17     pinMode(Sw_conta,INPUT); //Definimos Switch como Input.
18     digitalWrite(Led_transmite,LOW);
19 }
20 void loop()
21 {
22     Botao = digitalRead(Sw_conta);  //Recolhe o estado do botão
23     if (Botao == LOW)  //Botão foi apertado
24     {
25         do //Aguarda o usuário tirar o dedo do botão
26         {
27             Botao = digitalRead(Sw_conta);
28         }while (Botao==LOW);
29         digitalWrite(Led_transmite,HIGH);
30         delay(120);
31         digitalWrite(Led_transmite,LOW);
32         Serial.write(contador);      // envia o valor via serial em modo binário
33         contador++;
34         if (contador==10) contador=0;
35     }
36 }
```

Baud rate

Transmite

© Prof. Vargas

O Receptor RX

```
1  /*-----------------------------------------------------------------
2    Programa : REX.ino
3    Data     : 30/06/2020
4    Autor    : Prof. Vargas
5    Descrição:
6   -----------------------------------------------------------------*/
7  byte chegando;
8
9  void setup()
10 {
11     Serial.begin(9600);    // Habilita comunicação serial
12     pinMode (8, OUTPUT);
13     pinMode (9, OUTPUT);
14     pinMode (10, OUTPUT);
15     pinMode (11, OUTPUT);
16 }
17
18 void loop()
19 {
20     if (Serial.available()>0)     // Se há bytes livres vindo da porta serial
21     {
22         chegando = Serial.read();  // resgata o valor (byte)
23         PORTB = chegando;
24     }
25 }
```

Baud rate

Verifica

Recebe

© Prof. Vargas

Até a próxima !

© Prof. Vargas