A interface RS-232C

REVERSE ENGINEERING

deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

José Luís Azevedo, Bernardo Cunha

RS-232C

- Standard para comunicação série assíncrona entre um Equipamento Terminal de Dados (DTE, e.g. computador) e um Equipamento de Comunicação de Dados (DCE, e.g. Modem) - 1969
- Apenas suporta ligações ponto-a-ponto (implementações multi-ponto não standard)
- Permite comunicação bidirecional, *full-duplex*
- Conheceu uma grande utilização, que se estendeu muito para além do seu objetivo inicial (ligar DTEs a modems)
- Com o aparecimento do USB os computadores deixaram de disponibilizar comunicação RS-232C

RS-232C

- Por ser um modo de comunicação série muito fácil de implementar e de programar continua a ser muito usado em microcontroladores

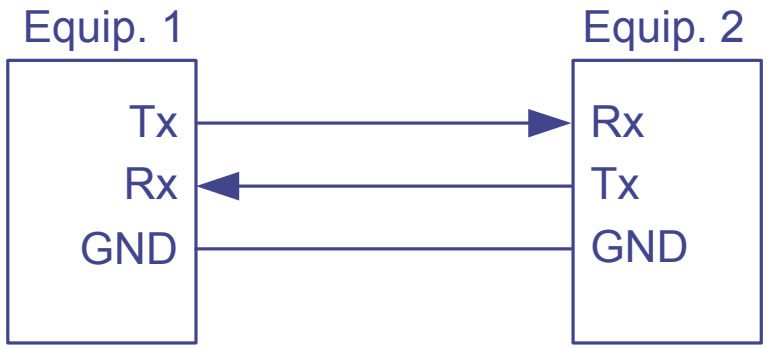
- Dispositivo que implementa a comunicação RS232C: **UART** (Universal Asynchronous Receiver-Transmitter)

- Existem no mercado conversores USB/RS-232C que permitem a ligação a PCs de equipamentos que implementam RS-232C (por exemplo o FT232R - FTDI)

RS-232C

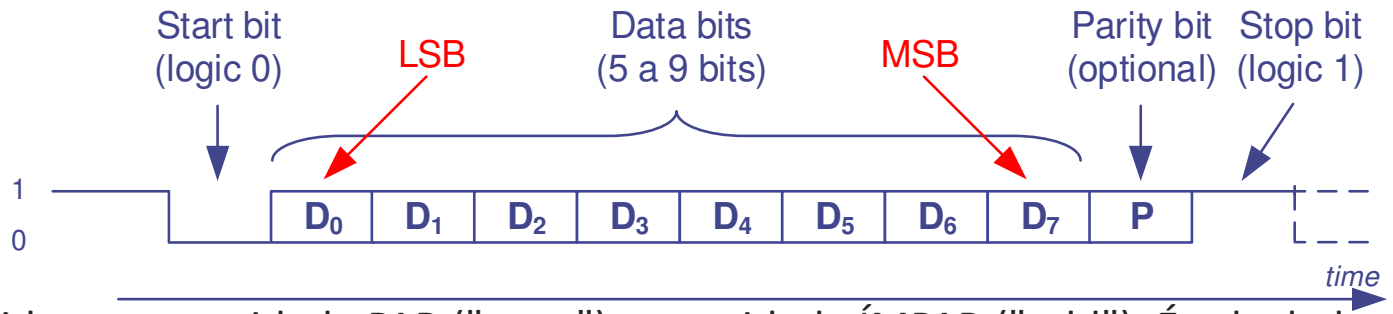
Sinalização

- Na sua forma mais simples, a implementação da norma RS-232C requer apenas a utilização de 2 linhas de sinalização e uma linha de massa



- Podem ser usadas linhas adicionais para protocolar a troca de informação entre os dois equipamentos (*handshake*)
  - RTS (Request to send), CTS (Clear to send), DTR (Data terminal ready), DSR (Data set ready)

Estrutura da trama

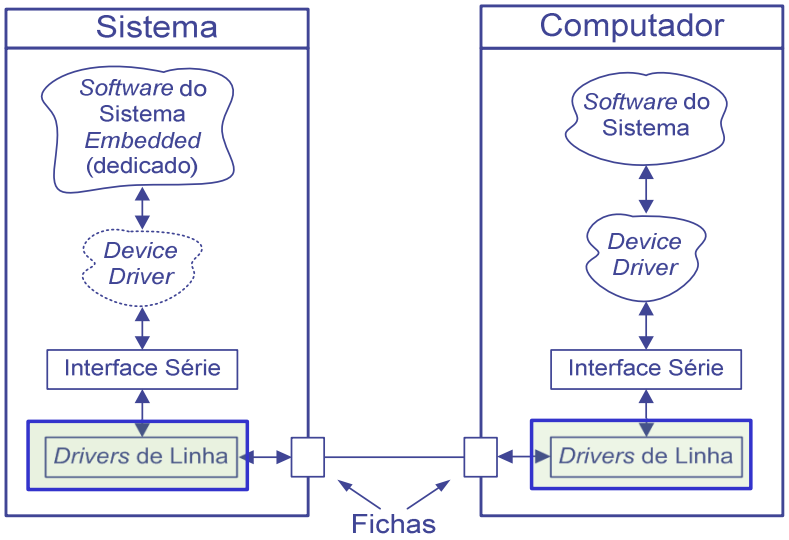


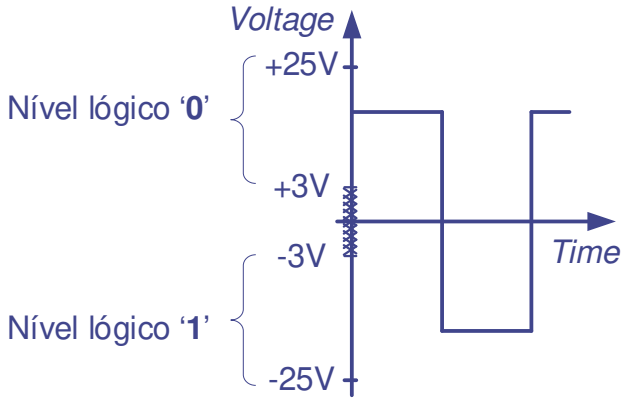
- Start bit
- 7 ou 8 bits de dados
- Parity bit (opcional)
  - Quando usado pode ser definido como paridade PAR ("even") ou paridade ÍMPAR ("odd"). É calculado pelo controlador de comunicação em cada trama
  - Paridade PAR: o bit é obtido através do XOR de todos os bits do campo de dados.
    Exemplo: data bits – 10011000 -> parity bit: 1
  - Paridade ÍMPAR: o bit é obtido através do XNOR de todos os bits do campo de dados.
    Exemplo: data bits – 11001001 -> parity bit: 1
  - Permite detetar erros de comunicação sempre que houver um número ímpar de bits errados
- Stop Bit(s): Podem ser usados 1 ou 2 bits
  - Coincidem com o estado de linha inativa (idle)
  - Proporcionam um intervalo de tempo de guarda mínimo entre o envio consecutivo de dois valores

Camada física

Codificação dos níveis lógicos

- Ao nível físico RS-232C os bits da trama são codificados em NRZ-L (*Non Return to Zero - Level*)
  - Nível lógico 1: codificado com uma tensão negativa (na gama -3V a -25V)
  - Nível lógico 0: codificado com uma tensão positiva (na gama +3V a +25V)
- A codificação e descodificação da trama com estes níveis de tensão é assegurada por circuitos eletrónicos designados por **drivers de linha**

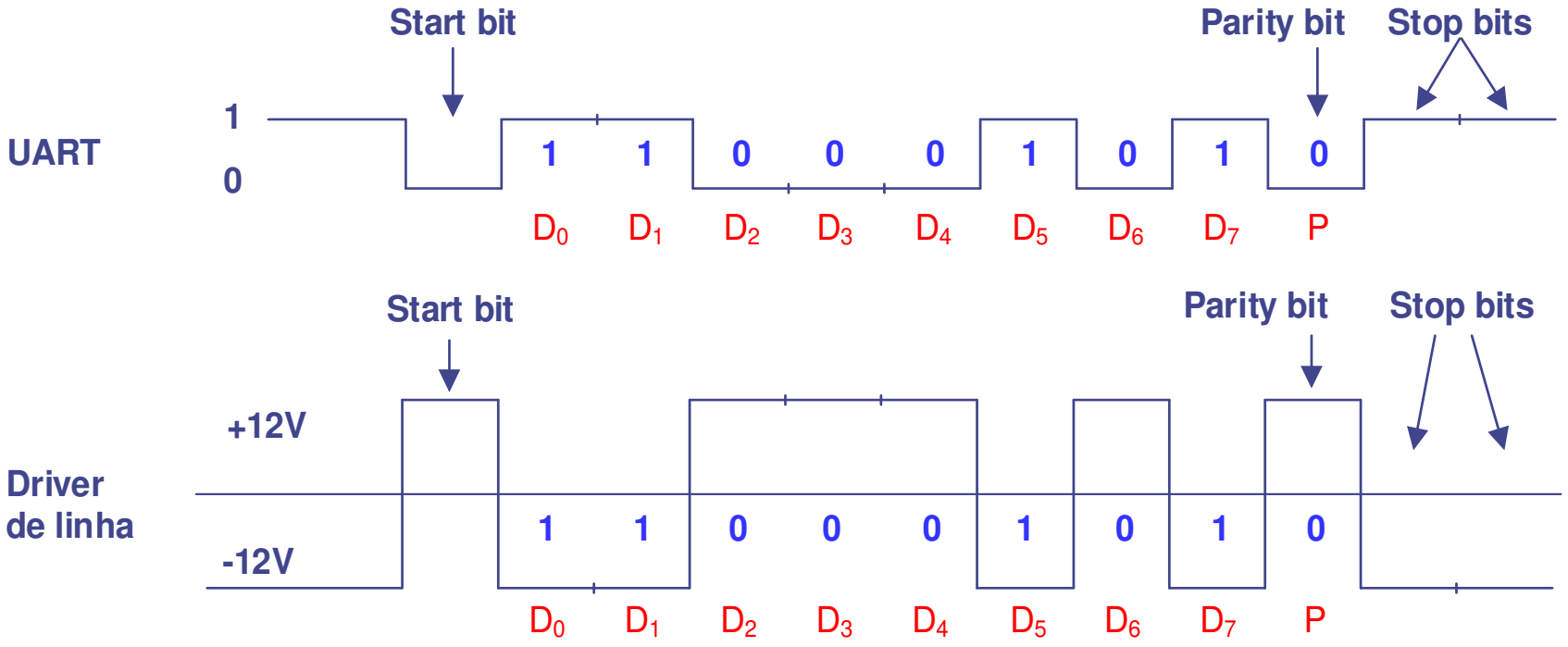




Exemplo

8 bits de dados, 2 stop bits, paridade par

- Valor a transmitir: 0xA3 (10100011)



Taxa de transmissão (*baudrate*)

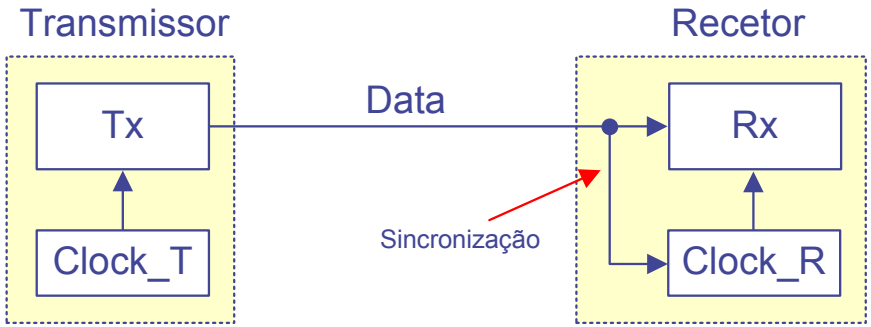
- Baudrate = frequência do relógio do transmissor
  - Ex: baudrate=115200 bps => freq_tx = 115200 Hz => t_bit = 1 / 115200 ~ 8,7 µs
- Exemplos comuns de *baudrates* em RS-232C [bps]: 600, 1200, 4800, 9600, 19200, 38400, 57600, 115200, 230400

- No exemplo anterior o número total de bits a serem transmitidos é 12 (1 start bit, 8 bits de dados, 1 bit de paridade, 2 stop bits)
- Considerando um *baudrate* de 57600 bps a transmissão completa de uma trama demora ~208 µs (12 / 57600)
  - Taxa de transmissão líquida: (8 * 57600) / 12, i.e., 38400 bps

Receção de dados

Sincronização de relógio (relógio implícito)

- Comunicação assíncrona
  (i.e. não há transmissão do relógio)
- O transmissor e o recetor têm
  relógios locais (independentes)



- Os instantes de amostragem no recetor são sincronizados no início da receção de cada nova trama, sinalizado pelo *start* bit:
  - transição de "1" para "0" na linha após um período de inatividade, por exemplo depois da receção completa de uma trama
- Este método é robusto (dentro de certos limites) a diferenças de frequência entre os relógios do transmissor e do recetor

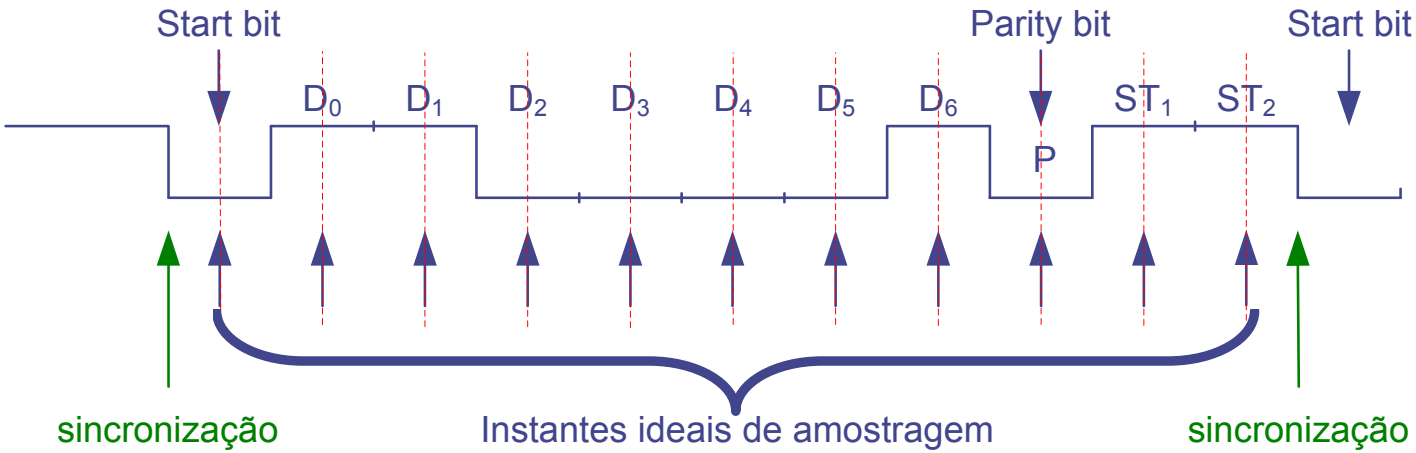
Receção de dados

- Para que a comunicação se processe corretamente, o transmissor e o recetor têm que estar configurados com os mesmos parâmetros:
  - Baudrate (relógios com a mesma frequência)
  - Estrutura da trama: nº de bits de dados, tipo de paridade, número de stop bits

Receção de dados

Sincronização da amostragem no recetor

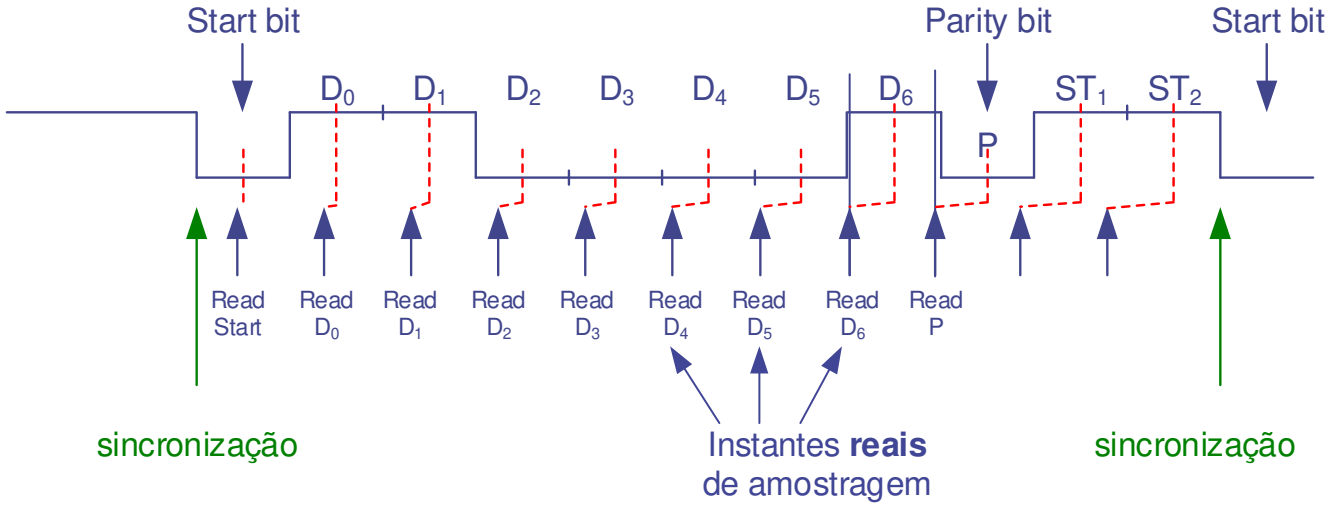
- O recetor deve sincronizar-se pelo flanco negativo (transição do nível lógico "1" para o nível lógico "0") da linha (Start bit) e, idealmente, fazer as leituras a meio do intervalo reservado a cada bit
- Exemplo da receção do valor 0x43: estrutura da trama 7, O, 2 (7 data bits, odd parity, 2 stop bits); 0x43: 1000011_2



Receção de dados

Sincronização da amostragem no recetor

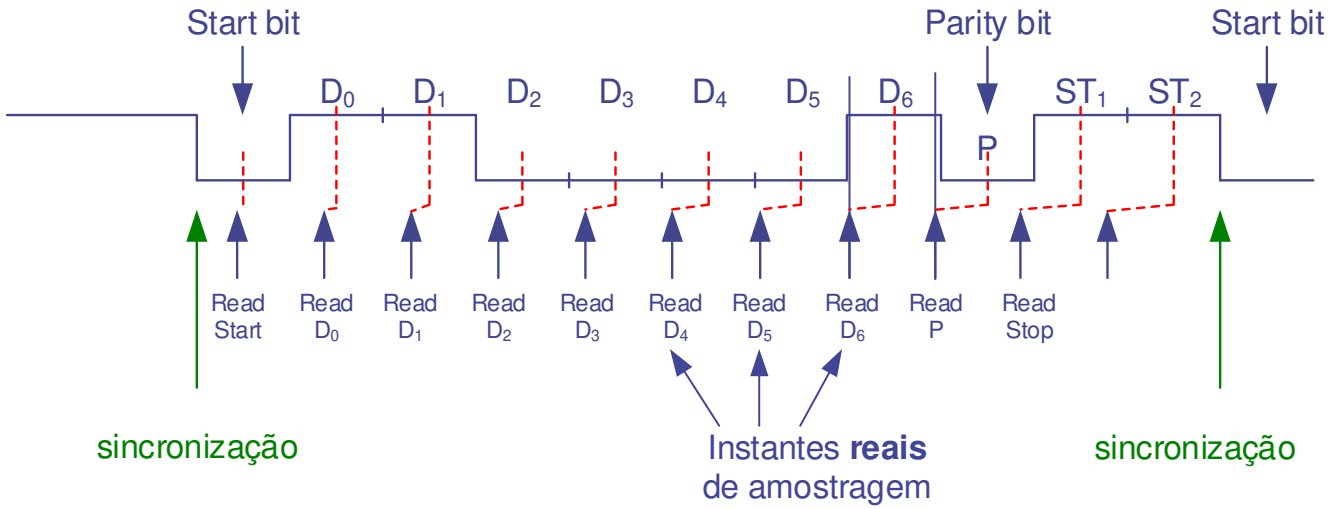
- Entre instantes de sincronização o desvio de frequência dos relógios depende da estabilidade/precisão dos relógios de transmissor e do recetor



- Neste exemplo a receção não é corretamente realizada devido a um desvio da frequência dos relógios do transmissor e do recetor (frequência do recetor > frequência do transmissor)

Receção de dados

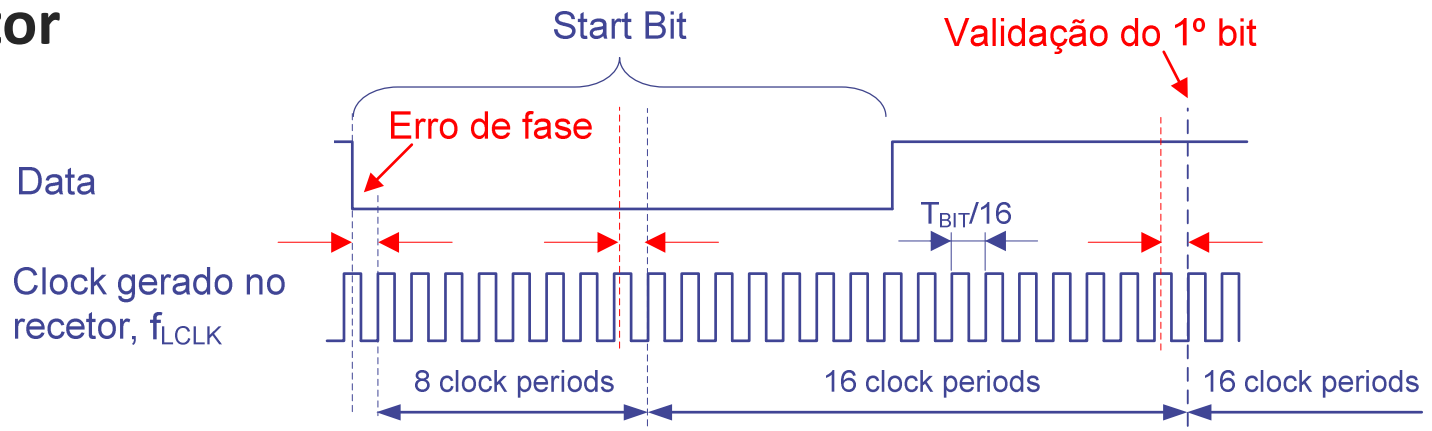
Erros



- Neste exemplo o recetor detetaria dois erros:
  - ***Parity Error*** (se o bit D6 for detetado como 1): o bit de paridade devia ser 0 e é lido como 1
  - ***Framing Error***: é detetado o nível lógico 0 no instante em que era esperado um stop bit (nível lógico 1)

Receção de dados

Relógio do recetor



- No recetor é gerado um relógio com uma frequência N vezes superior ao relógio do transmissor (sincroniza a receção a partir desse relógio)
  - N típicos: 4, 8, 16, 64
- Se N = 16
  - "Start bit", validado ao fim de 8 ciclos de relógio
  - Restantes bits validados a cada 16 ciclos de relógio

Receção de dados

Máximo desvio de frequência entre emissor e recetor

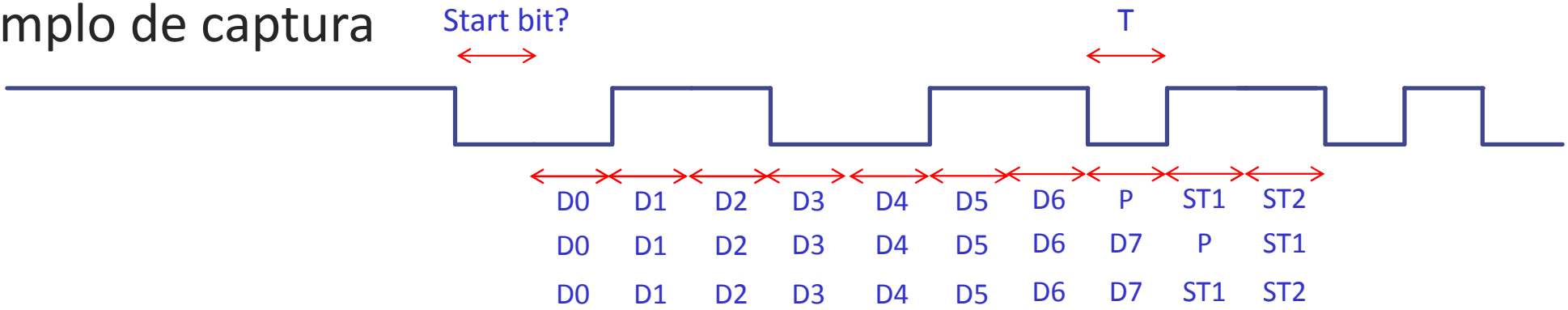
- Como referência, a máxima discrepância que poderá ser tolerada entre os relógios do transmissor e do recetor é $\Delta f \approx \pm 3.0\%$

- Exemplo: taxa de transmissão de 115200 bps. Para que a comunicação se processe sem erros:
  - Frequência ideal no recetor:
    - $freq_{RX} = 115200\ Hz$
  - Desvio aceitável ($\Delta f \approx \pm 3.0\%$):
    - $freq_{RX} \in [111700, 118600]\ Hz$

Identificar parâmetros de comunicação

Baudrate, #data bits, parity, stop bits

- Exemplo de captura



  - Encontrar o menor tempo a 1 ou a 0 e medir: baudrate = 1 / T
  - Identificar o start bit; candidato: transição de 1 para 0 após período de inatividade
  - #data bits (7 ou 8) + paridade + stop bits (1 ou 2); candidatos:
    - 7 data bits, paridade par, 2 stop bits
    - 8 data bits, parity ímpar, 1 stop bit
    - 8 data bits, sem paridade, 2 stop bits

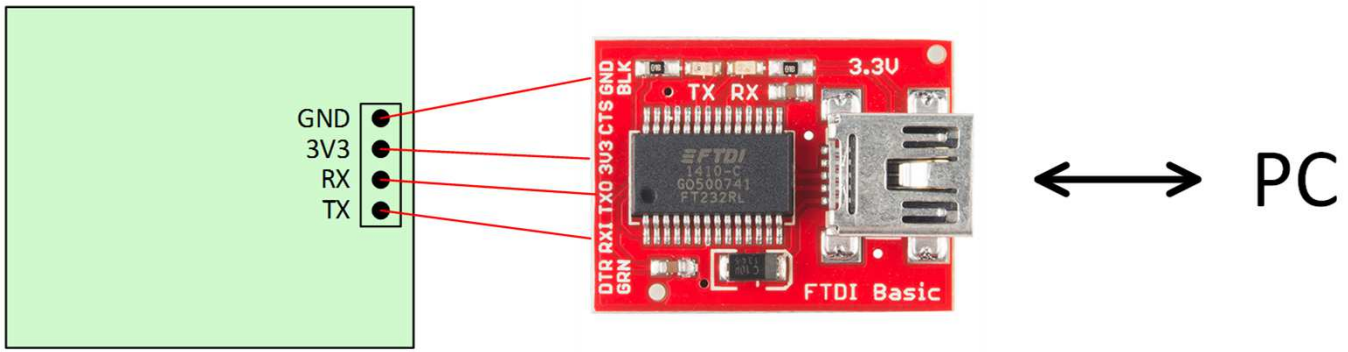
Encontrar a UART…

- Verificar se existe na placa referência explícita à UART (com ou sem ficha soldada)
  - Pinos: TX, RX, GND / T, R, G, …
- Fichas não identificadas com 3 ou 4 pinos, são candidatas
  - 4 pinos: TX, RX, GND, 3V3
- Procurar documentação da placa, online

Identificação dos pinos da UART

- Com a ajuda de um multímetro
  - Identificar o pino de "ground"
  - Identificar o pino de alimentação
    - Tensão fixa de, por exemplo, 3.3V
  - Identificar o pino TX
    - Tensão positiva aproximadamente igual à tensão de alimentação (tip. +3.3V) (pode apresentar flutuações se o dispositivo estiver a transmitir)
    - Se estiverem a ser usados drivers de linha a tensão será tendencialmente negativa (-5V a -12V)
  - Identificar o pino RX
    - Com a alimentação do circuito desligada, medir a resistência entre o ponto em análise e o ground
    - Resistência elevada: possivelmente é o RX

Ligação a um PC

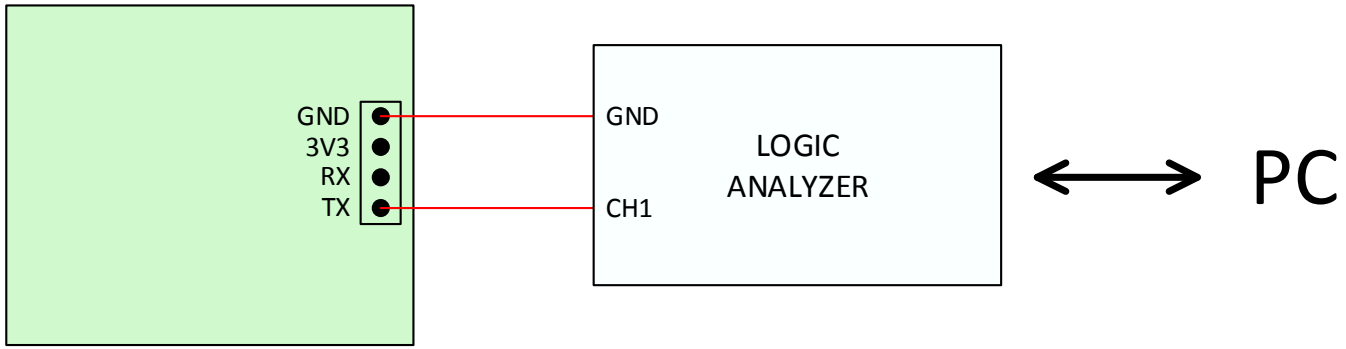


- No PC, programa terminal
  - putty, pterm (made in DETI), minicom, …

- Cuidado: esta ligação só pode ser feita se a placa não tiver drivers de linha
  - Com drivers de linha: é necessário um conversor entre a placa e o módulo RS232/USB

Captura de sinais

Ligação a um analisador lógico



- No PC, logic analyzer sw
  - PulseView (sigrok), Logic 2 (saleae)