

# 20 - UART

Rafael Corsi Ferrão  
corsiferrao@gmail.com

24 de agosto de 2016



*Entregar na próxima aula em formato PDF via github.*

## 1 Visão geral

A utilização da comunicação serial (UART) é presente em grande parte dos dispositivos eletrônicos embarcados e possui grande utilidade na depuração de aplicações. Em muitos casos,, são utilizados microcontroladores ou ferramentas que não possuem suportam debug (JTAG,ICE), como por exemplo Arduino UNO, a alternativa nesses casos se dá pelo envio via a porta UART de informações relevantes sobre a execução do código.

### 1.1: Diagrama

Desenhe um diagrama de um sistema embarcado utilizando a comunicação UART com o computador para depuração do sistema.

Uma grande vantagem da utilização dessa técnica é a de que em algumas ocasiões certos tipos de "bugs" não são detectados quando o dispositivo está sendo debugado utilizando-se "break-points". Outra questão é que em modo debug o compilador assume que certas otimizações não devem ser feitas, e o código embarcado sem debug (release) pode diferenciar daquele embarcado em modo de debug.

Atmel Studio : no software quando em modo de debug, a otimização selecionada é **-O1** já em modo de release, a otimização selecionada é : **-OS** (*for size*).

Outra diferença é que em modo debug, a flag do -g3 é selecionada :



option	description
-g0	no debug information
-g1	minimal debug information
-g	default debug information
-g3	maximal debug information

A comunicação UART pode ser encapsulada por diferentes padrões, tais como : RS-232, RS-485, USB-FTDI, .... Esses protocolos são um intermediário entre os dois nós (microcontrolador e dispositivo). A figura a seguir ilustra a comunicação entre dois dispositivos (A e B) via a comunicação serial RS-232.

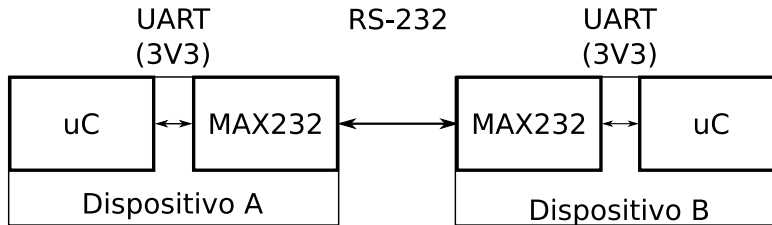


Figura 1: UART-RS232-UART

## 1.2: UART

- Liste e descreva os sinais utilizados na comunicação uart

## 1.3: RS-232

- Liste e descreva os sinais utilizados na comunicação RS-232

## 1.4: Driver RS-232

- Pesquise por um componente (CI) que realize a ponte entre UART e RS-232

# 2 Camadas

A comunicação UART (RS-232,...) define somente a camada física (camada 1 do modelo OSI), com isso, o desenvolvedor fica responsável por definir como os dados devem ser transmitidos e interpretados.

Por exemplo, um dispositivo transmite dados de temperatura (T) e umidade (U) para um computador, ambos os dados são de 8 bits de largura. Esses dados podem ser transmitidos de diversas formas, como por exemplo, puramente hexadecimal, ou codificados em ASCII-II. Porém, alguns caracteres devem ser adicionados para a identificação dos dados (qual valor corresponde a qual medida).

Se os dados a serem transmitidos fossem maiores que 8 bits, deveríamos especificar uma maneira na qual eles seriam fragmentados, isso também fica a cargo do desenvolvedor.

Nesse exemplo os bytes 'T', 'U' e '\n' devem ser interpretados como caracteres da tabela ASCII, já os dados de temperatura e umidade como valores brutos.

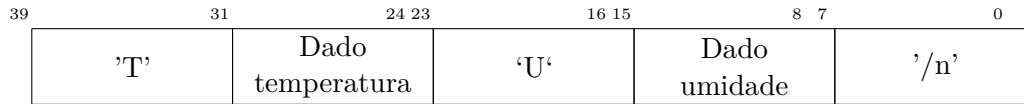


Figura 2: Definição de pacote para envio de dado de Temperatura e Umidade

## 2.1: ASCII

Qual seria o impacto no tamanho do pacote, se todos os dados fossem transmitidos como ASCII, lembrando que o valor máximo a ser transmitido de temperatura e umidade para esse caso é de 255 (decimal).

A seguir uma lista de opções que podem ser realizadas quando na criação de um protocolo de comunicação entre dois dispositivos:

- Codificação;
- empacotamento;
- caracteres de controle;
- acknowledgement

## 2.2: Acknowledgement

O que é o acknowledgement em uma transmissão ?

## 3 UART - SAM4S-EK2

O periférico UART presente no uC SAM4S possui a arquitetura ilustrada na figura a seguir :

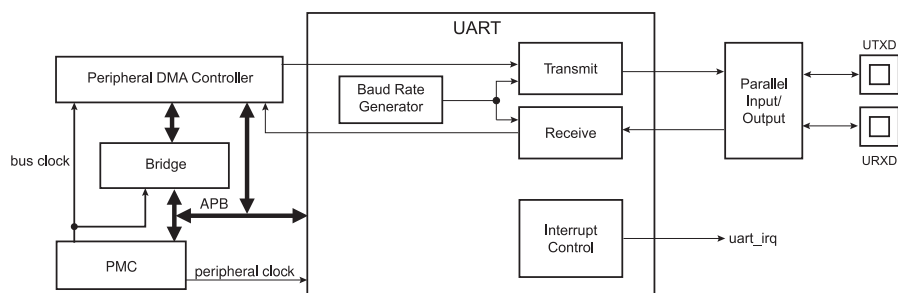


Figura 3: Diagrama periférico UART - Datasheet: SAM4S

### 3.1: Periférico

Sobre o UART :

- Quantos periféricos o uC possui ?
- Qual os seus IDs ?

O periférico é responsável por serializar os dados a serem transmitidos (Tx) e desserializar os dados recebidos (Rx). Possui as seguintes características configuráveis :

- BaudRate
- Paridade (par ou ímpar)
- Stop bit
- Local loopback
- Gerador de interrupção
- ....

### 3.2: Periférico Pinos

Liste os pinos e os respectivos sinais utilizados pelos periféricos UART.

### 3.1 Receptor

Quando um caractere completo é interpretado (8 bits + controles), o dado é transferido para o registrador UART\_RHR e o bit de status RXRDY é setado. Esse bit se mantém em 1 até a o recebimento do próximo dado.

O diagrama a seguir ilustra o recebimento de um dado :

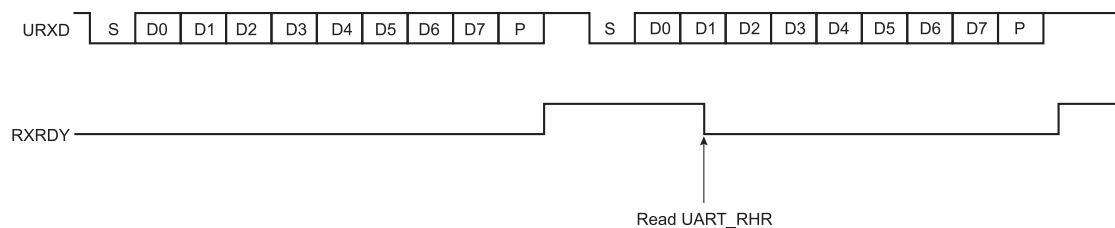


Figura 4: Recebimento de pacote - Datasheet: SAM4S

### 3.3: Periférico Rx errors

Liste e explique os possíveis erros que podem haver na recepção de um dado e como eles são gerenciados pelo periférico.

## 3.2 Transmissor

O transmissor controla o pino UTXD na taxa (baudrate) configurada, inserindo os bits de controle conforme as configurações realizadas no periférico (stop bit, paridade).

O dado a ser transmitido deve ser escrito no registrador UART\_THR, esse dado é então copiado para um buffer interno (de apenas um byte) que será utilizado na transmissão. O bit TXRDY indica quando um novo dado pode ser escrito no periférico.

### 3.4: Periférico Tx diagrama

Explique o diagrama da transmissão de dados a seguir :

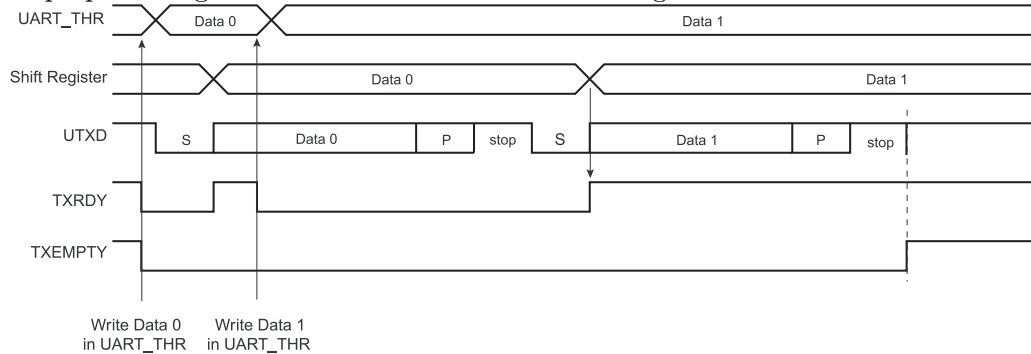


Diagrama da transmissão de dados - Datasheet: SAM4S

## 4 Interrupção

O periférico suporta diversos modos de interrupção, sendo eles :

- RXRDY: Disable RXRDY Interrupt
- TXRDY: Disable TXRDY Interrupt
- ENDRX: Disable End of Receive Transfer Interrupt
- ENDTX: Disable End of Transmit Interrupt
- OVRE: Disable Overrun Error Interrupt
- FRAME: Disable Framing Error Interrupt
- PARE: Disable Parity Error Interrupt
- TXEMPTY: Disable TXEMPTY Interrupt
- TXBUFE: Disable Buffer Empty Interrupt
- RXBUFF: Disable Buffer Full Interrupt

Para utilizar essas interrupções, deve-se primeiramente configurar o NVIC para aceitar as interrupções vindas desse periférico e então configurar o registrador UART\_IDR para gerar a interrupção desejada.

## 5 Projeto

Com base no código exemplo 20-UART, desenvolva um projeto que permita ao usuário configurar via comunicação RS-232 as seguintes opções :

- Qual LED será ativado (dentre os 3)
- A frequência de oscilação desses leds (individualmente)

O projeto deve utilizar interrupção para recepção dos dados da serial.

**Deve ser disponibilizado junto com o código um manual de usuário descrevendo como o protocolo funciona e quais devem ser os comandos para configurar a placa.**