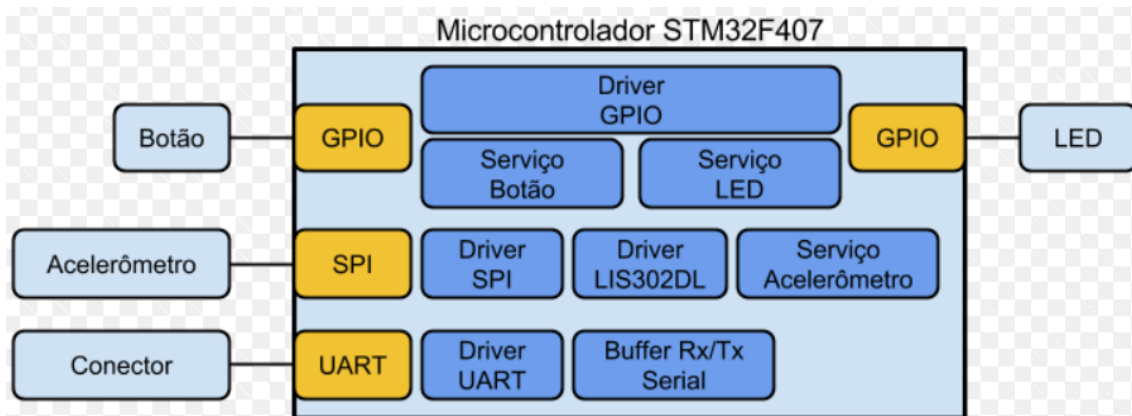




1.1: Diagrama

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





1.2: UART

- Liste e descreva os sinais utilizados na comunicação uart

De uma maneira geral, uma implementação de UART pode ser dividida em quatro blocos, a saber: transmissão, recepção, geração de *baud-rate* e lógica de interface (figura 1.3).

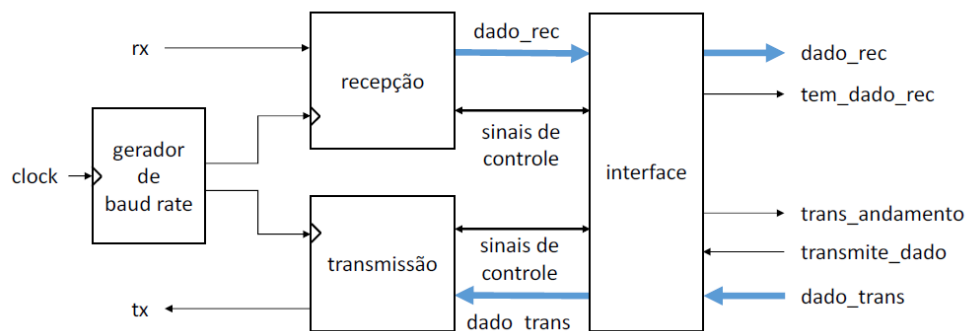


Figura 1.3 – Diagrama de blocos de uma implementação de UART.

Os blocos de **transmissão** e **recepção** são responsáveis pela troca de dados entre os equipamentos, com a comunicação de dados seriais. O bloco **gerador de baud rate** é responsável pela geração de um sinal de controle de temporização para os blocos de transmissão e recepção. Por exemplo, para o projeto do circuito de transmissão serial assíncrona, gera um sinal de 1200 Hz para uma transmissão configurada para 1200 *bauds*. O bloco de **interface** faz a comunicação com outro sistema digital através dos sinais de dados paralelos *dado_trans* e *dado_rec*. Os sinais de controle da interface são:

- *tem_dado_rec*: indica a presença de um dado recebido pelo circuito que deve ser registrado;
- *trans_andamento*: indica que uma transmissão está em andamento; assim, uma nova transmissão não pode ser realizada até que a atual termine;
- *transmite_dado*: submete uma nova transmissão para o UART.



1.3: RS-232

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

Sinais de Intercomunicação

Neste item são descritas as funções dos principais sinais de intercomunicação padronizados pela norma EIA-RS-232C.

- FRAME GROUND: corresponde ao fio terra dos equipamentos. É ligado à carcaça metálica do equipamento e nas partes condutoras do mesmo, expostas ao operador, de modo a evitar diferenças de potencial entre o equipamento e o operador;
- SIGNAL GROUND: estabelece a referência elétrica (terra) para todos os sinais de intercomunicação, exceto para o FRAME GROUND;
- TRANSMITTED DATA: corresponde ao dado serializado, gerado pelo terminal de dados;
- RECEIVED DATA: corresponde ao dado serializado recebido do canal de comunicação de dados;
- REQUEST TO SEND: gerado pelo terminal de dados, informa aos conversores de sinais a ele conectados, que o terminal deseja transmitir dados.
- CLEAR TO SEND: gerado nos circuitos dos conversores de sinais, é utilizado para indicar que os circuitos estão prontos para enviar dados.

Todos estes sinais e os possíveis opcionais devem obedecer às características elétricas relacionadas no item 1.2.

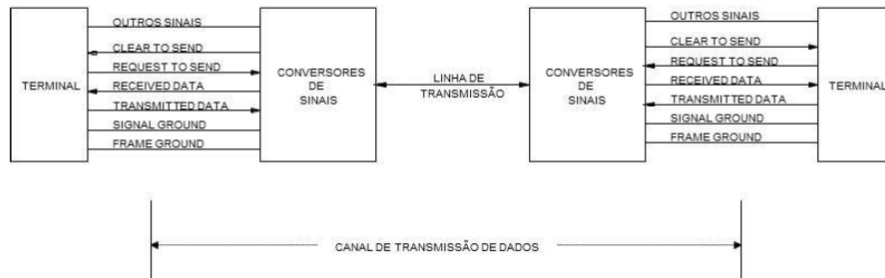


Figura 1.1 – Interligação entre dois Terminais de Dados.



1.4: Driver RS-232

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

Conversão de Níveis de Tensão

Como os padrões de nível de tensão para circuitos digitais (TTL ou CMOS) e para o RS-232C são diferentes, é necessário o uso de circuitos especializados para conversão de níveis de tensão. Por exemplo, o bit 1, que em um circuito digital tem um nível de tensão típico da ordem de +5V, deve ser convertido para um sinal MARK que tem tipicamente um nível de tensão de -12V. Da mesma forma, o bit 0 (tensão de 0V) deve ser convertido para o sinal SPACE (tensão +12V).

Vários componentes estão disponíveis no mercado para realizar a conversão de níveis de tensão. Por exemplo, temos o par 1488/1489 e o MAX232.

O componente 1488 é responsável pela conversão de níveis de tensão TTL para RS-232 e o 1489, de RS-232 para TTL. A figura 1.7 apresenta as pinagens destes componentes.

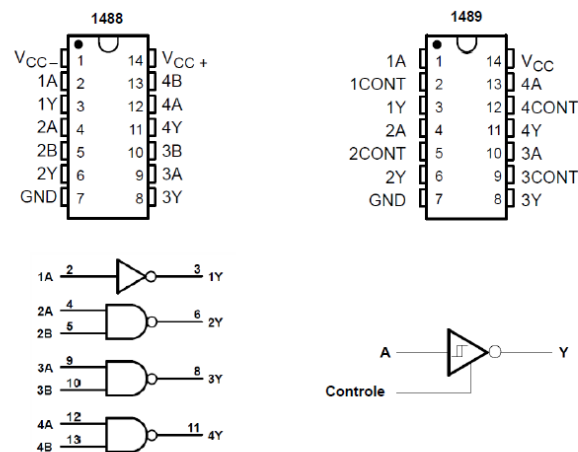


Figura 1.7 – Pinagens e esquemas lógicos dos conversores de tensão 1488 e 1489.

Convém observar que o conversor 1488 tem como pinos de alimentação: V_{CC}+, V_{CC}- e GND (tipicamente, +12V, -12V e 0V, respectivamente). Já o conversor 1489 tem os pinos comuns de alimentação: V_{CC} (tipicamente +5V) e GND (0V).

O circuito integrado MAX232 contém circuitos de conversão de tensão para ambos os sentidos (TTL para RS232C e RS232C para TTL) em um único encapsulamento. A figura 1.8 mostra a pinagem deste CI.

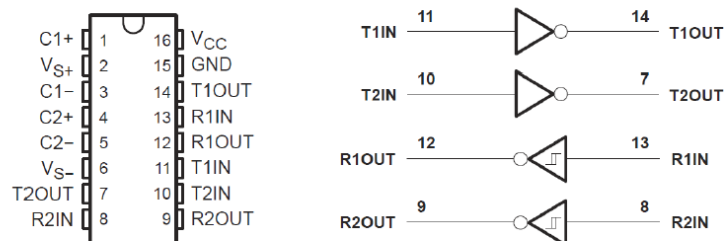


Figura 1.8 – Pinagem e esquema lógico do MAX232.



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).

Ocorre aumento do tamanho do pacote, pois utiliza-se mais bits.

2.2: Acknowledgement

O que é o acknowledgement em uma transmissão ?

mensagens ACK (=Acknowledgement), confirmando a recepção de um segmento.

3.1: Periférico

Sobre o UART :

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

2 (dois)

UART0

UART1

3.2: Periférico Pinos

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

Table 35-1. UART Pin Description

Pin Name	Description	Type
URXD	UART Receive Data	Input
UTXD	UART Transmit Data	Output



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.

35.5.2 Receiver

35.5.2.1 Receiver Reset, Enable and Disable

After device reset, the UART receiver is disabled and must be enabled before being used. The receiver can be enabled by writing the Control Register (UART_CR) with the bit RXEN at 1. At this command, the receiver starts looking for a start bit.

The programmer can disable the receiver by writing UART_CR with the bit RXDIS at 1. If the receiver is waiting for a start bit, it is immediately stopped. However, if the receiver has already detected a start bit and is receiving the data, it waits for the stop bit before actually stopping its operation.

The receiver can be put in reset state by writing UART_CR with the bit RSTRX at 1. In this case, the receiver immediately stops its current operations and is disabled, whatever its current state. If RSTRX is applied when data is being processed, this data is lost.

35.5.2.2 Start Detection and Data Sampling

The UART only supports asynchronous operations, and this affects only its receiver. The UART receiver detects the start of a received character by sampling the URXD signal until it detects a valid start bit. A low level (space) on URXD is interpreted as a valid start bit if it is detected for more than seven cycles of the sampling clock, which is 16 times the baud rate. Hence, a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

When a valid start bit has been detected, the receiver samples the URXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1-bit period) so the bit sampling point is eight cycles (0.5-bit period) after the start of the bit. The first sampling point is therefore 24 cycles (1.5-bit periods) after detecting the falling edge of the start bit.

Each subsequent bit is sampled 16 cycles (1-bit period) after the previous one.

Figure 35-3. Start Bit Detection

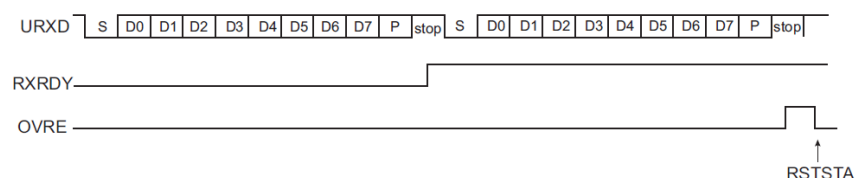
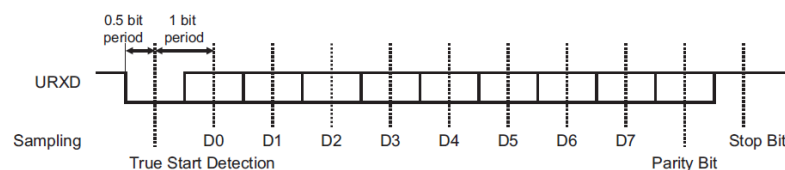


Figure 35-4. Character Reception

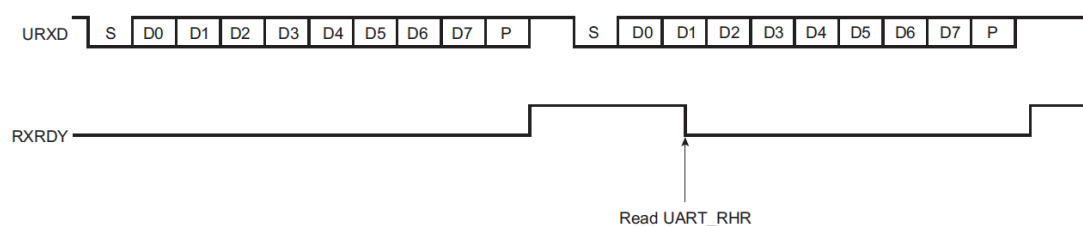
Example: 8-bit, parity enabled 1 stop



35.5.2.3 Receiver Ready

When a complete character is received, it is transferred to the Receive Holding Register (UART_RHR) and the RXRDY status bit in the Status Register (UART_SR) is set. The bit RXRDY is automatically cleared when UART_RHR is read.

Figure 35-5. Receiver Ready

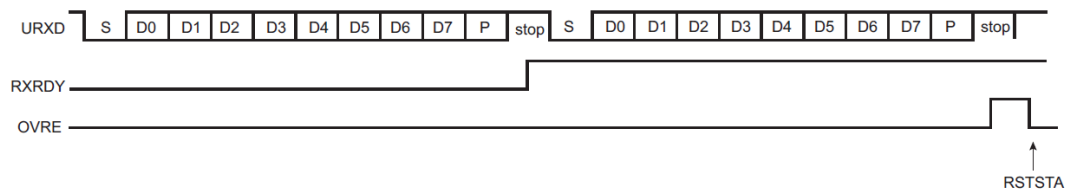




35.5.2.4 Receiver Overrun

The OVRE status bit in UART_SR is set if UART_RHR has not been read by the software (or the PDC) since the last transfer, the RXRDY bit is still set and a new character is received. OVRE is cleared when the software writes a 1 to the bit RSTSTA (Reset Status) in UART_CR.

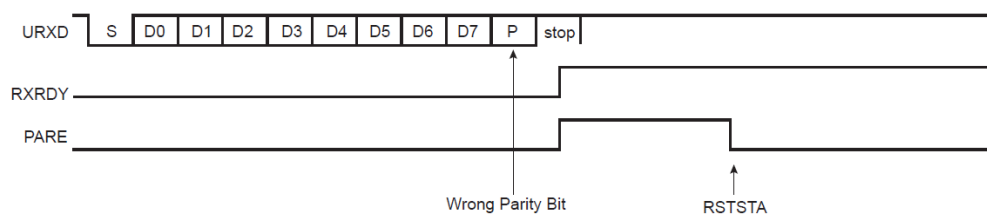
Figure 35-6. Receiver Overrun



35.5.2.5 Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in the Mode Register (UART_MR). It then compares the result with the received parity bit. If different, the parity error bit PARE in UART_SR is set at the same time RXRDY is set. The parity bit is cleared when UART_CR is written with the bit RSTSTA (Reset Status) at 1. If a new character is received before the reset status command is written, the PARE bit remains at 1.

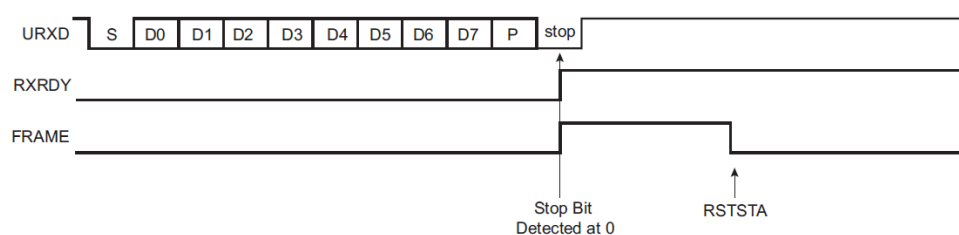
Figure 35-7. Parity Error



35.5.2.6 Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, the FRAME (Framing Error) bit in UART_SR is set at the same time the RXRDY bit is set. The FRAME bit remains high until the Control Register (UART_CR) is written with the bit RSTSTA at 1.

Figure 35-8. Receiver Framing Error





3.4: Periférico Tx diagrama

Explique o diagrama da transmissão de dados a seguir :

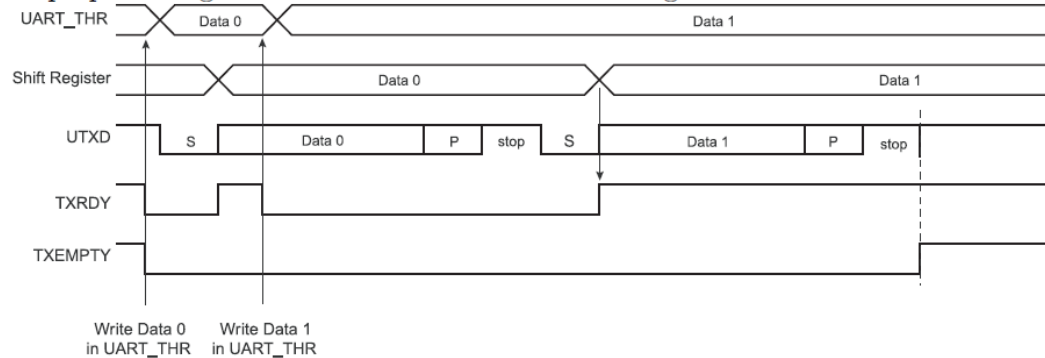


Diagrama da transmissão de dados - Datasheet: SAM4S

35.5.3.3 Transmitter Control

When the transmitter is enabled, the bit TXRDY (Transmitter Ready) is set in UART_SR. The transmission starts when the programmer writes in the UART_THR, and after the written character is transferred from UART_THR to the internal shift register. The TXRDY bit remains high until a second character is written in UART_THR. As soon as the first character is completed, the last character written in UART_THR is transferred into the internal shift register and TXRDY rises again, showing that the holding register is empty.

When both the internal shift register and UART_THR are empty, i.e., all the characters written in UART_THR have been processed, the TXEMPTY bit rises after the last stop bit has been completed.