

13 - (UART) Olá Você !

Rafael Corsi - rafael.corsi@insper.edu.br

Abril - 2017

Introdução



Figura 1:

- Quais são os principais protocolos de comunicação entre dispositivos de um PC?

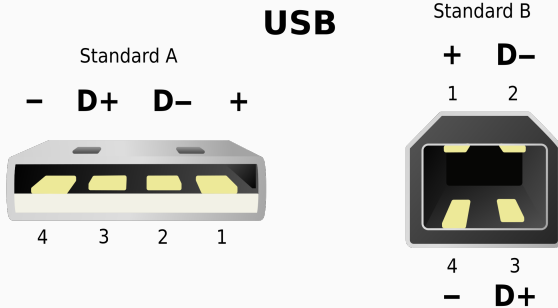
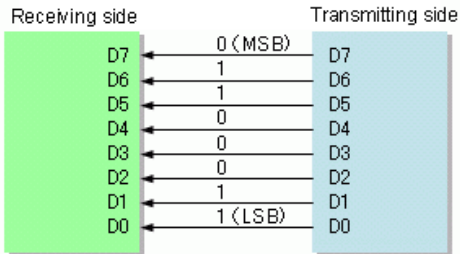


Figura 2: usb

Parallel interface example



Serial interface example (MSB first)

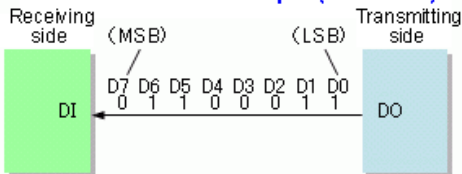


Figura 3: usb

- Comunicação serial é amplamente utilizada para interface entre dois dispositivos, normalmente referenciado como :
 - Machine Two Machine (m2m)
- Interface entre Sensores, atuadores e outras partes digitais do sistema.

UART (universal asynchronous receiver/transmitter)

UART (universal asynchronous receiver/transmitter)



UART (universal asynchronous receiver/transmitter)



- Protocolo utilizado amplamente em eletrônica embarcada
- Assíncrono
- Comunicação com dispositivos

UART



Figura 4: Protocolo UART

UART parâmetros

- Bits pro segundo (baudrate), típico : 1200, 2400, 4800, 9800, 19200, 38400, 57600, and 115200.
- Data : 8 ou 9 bits
- Paridade : 0 ou 1 bit
- Stop : 1 ou 2 bits

Classificação de comunicação Serial

Podemos classificar uma comunicação serial nas seguintes categorias :

- Assíncrona/ Síncrona
- Half Duplex/ Full Duplex
- Com/Sem controle de fluxo
- Multi mestre/ Multi escravo

XPLD

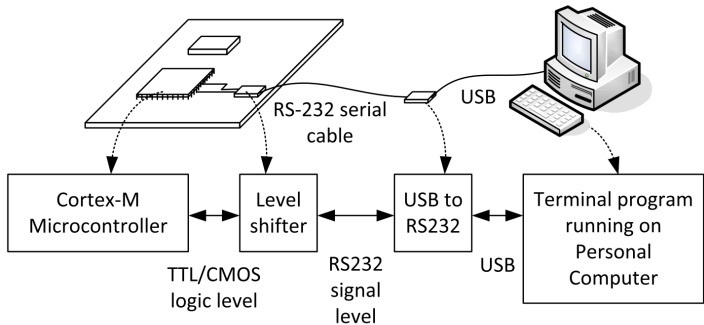


FIGURE 2.12

Using a UART to communicate with a PC via USB

Figura 5: Yiu, Joseph. pg. 46

Debug computação embarcada

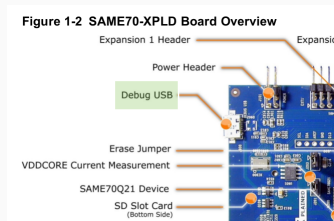


Figura 6: SAME70-XPLD

(SAME70-XPLD User Guide, sec. 3.1)

- The SAME70-XPLD contains the Atmel Embedded Debugger (EDBG) for on-board debugging. The EDBG is a composite USB device of three interfaces: a debugger, **Virtual COM Port**, and a Data Gateway Interface (DGI).

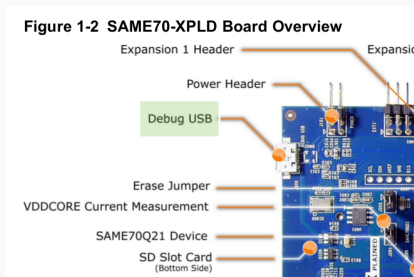


Figura 7: SAME70-XPLD EDBG

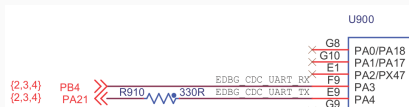


Figura 8: SAME70-XPLD EDBG

Tá mais qual ?

- Pinos
 - PB4
 - PA21

105	A12	C13	VDDIO	GPIO_MLB	PB4	I/O	TDI ⁽⁹⁾	I	TWD1	I/O	PWMC0_PWMH2	O	–	–	TXD1	I/O	PIO, I, PD, ST
-----	-----	-----	-------	----------	-----	-----	--------------------	---	------	-----	-------------	---	---	---	------	-----	----------------

Figura 9: PB4 mux

32	K2	M1	VDDIO	GPIO_AD	PA21	I/O	AFED_AD1/ PIOCEN2 ⁽¹⁰⁾	I	RXD1	I	PCK1	O	PWMC1_PWMF0	I	–	–	PIO, I, PU, ST
----	----	----	-------	---------	------	-----	--------------------------------------	---	------	---	------	---	-------------	---	---	---	----------------

Figura 10: PA21 mux

Periféricos

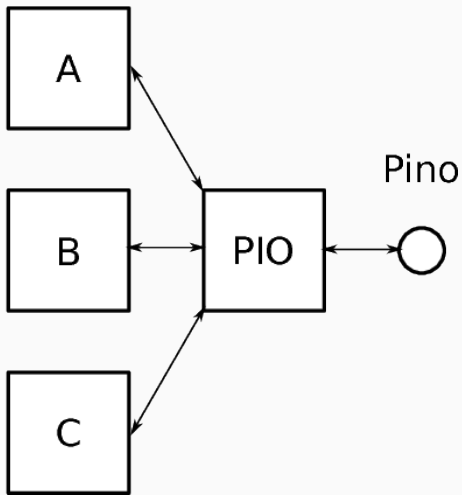


Figura 11: PA21 mux

PIO ?

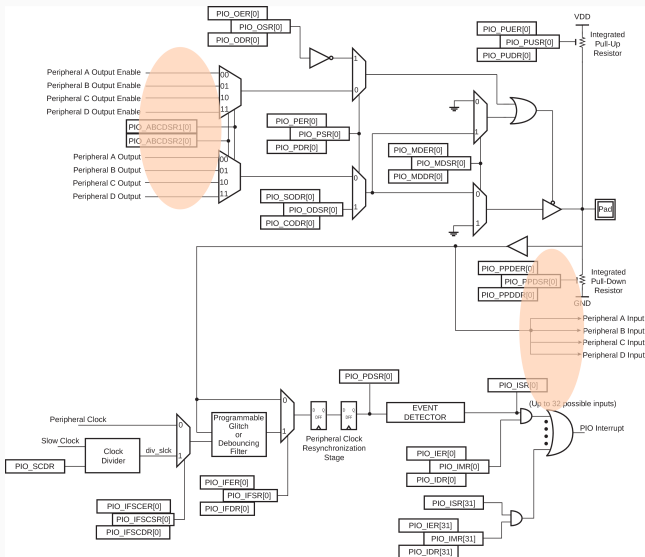


Figura 12: PA21 mux

Periféricos

Periféricos UART

Existem dois periféricos no SAME70 que implementam a comunicação UART, são eles :

- Universal Synchronous Asynchronous Receiver Transceiver (USART)
- Universal Asynchronous Receiver Transmitter (UART)

Table 45-3. Peripheral IDs

Instance	ID
USART0	13
USART1	14
USART2	15

Table 46-3. Peripheral IDs

Instance	ID
UART0	7
UART1	8
UART2	44
UART3	45
UART4	46

Tá mais qual ?

- Pinos
 - PB4
 - PA21

Estão conectados ao USART1, entrada do mux D e A

Table 45-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
USART0	CTS0	PB2	C
USART0	DCD0	PD0	D
USART0	DSR0	PD2	D
USART0	DTR0	PD1	D
USART0	RI0	PD3	D
USART0	RTS0	PB3	C
USART0	RXD0	PB0	C
USART0	SCK0	PB13	C
USART0	TXD0	PB1	C
USART1	CTS1	PA25	A
USART1	DCD1	PA26	A
USART1	DSR1	PA28	A
USART1	DTR1	PA27	A
USART1	LONCOL1	PA3	B
USART1	RI1	PA29	A
USART1	RTS1	PA24	A
USART1	RXD1	PA21	A
USART1	SCK1	PA23	A
USART1	TXD1	PB4	D
USART2	CTS2	PD19	B

- Devemos configurar o periférico USART primeiramente para operar em modo serial Asynchronous
- Devemos configurar os parâmetros (baudrate/ paridade/ data/ stopbit)
- Devemos inicializar a transmissão e a recepção

Os dados recebidos são salvo no registrador :

- USART Receive Holding Register (US_RHR)
 - Pode ser acessado pela função :

Os dados a serem transmitidos devem ser salvo no registrador :

- USART Transmit Holding Register (US_THR)
 - pode ser acessado pela função :

Interrupção ?

- Podemos configurar algumas interrupções no USART, são algumas delas elas :
 - RXRDY: RXRDY Interrupt Enable
 - TXRDY: TXRDY Interrupt Enable
 - OVRE: Overrun Error Interrupt Enable
 - FRAME: Framing Error Interrupt Enable
 - PARE: Parity Error Interrupt Enable
 - TIMEOUT: Time-out Interrupt Enable
 - TXEMPTY: TXEMPTY Interrupt Enable
- Podendo ser configurada pela função :
`usart_enable_interrupt(USART_SERIAL, US_IER_RXRDY);`