



Universidade Tecnológica Federal do Paraná – UTFPR
Bacharelado em Ciência da Computação

BCC34C – Sistemas Microcontrolados

Aula Hoje

- Capítulo 15. USART
 - 15.1 USART do ATmega328
 - 15.3 Comunicação entre o Microcontrolador e um Computador

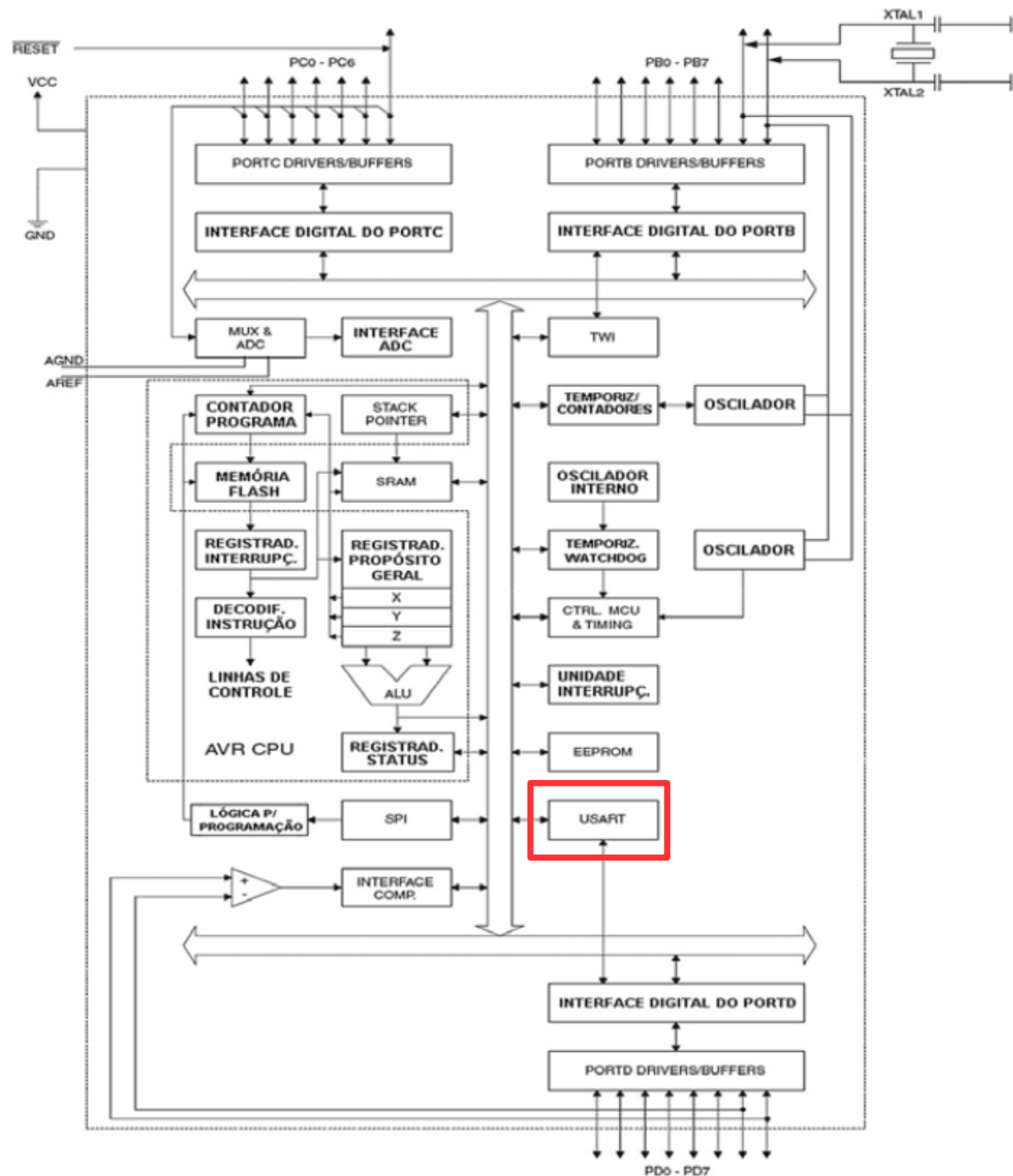
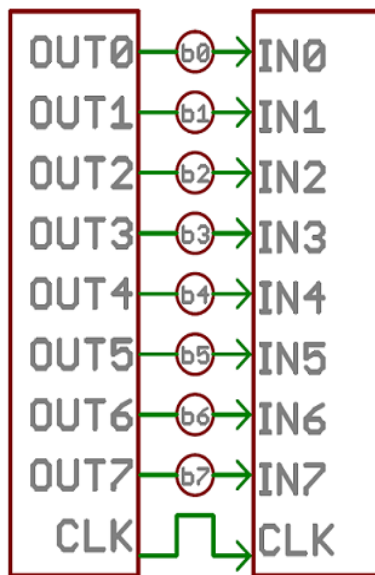


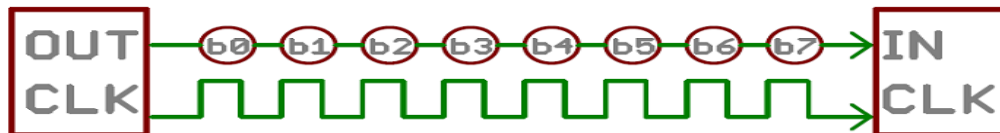
Fig. 2.3 – Diagrama em blocos do ATmega328.

Tipos de Comunicação Serial

- Nem sempre é possível utilizar barramentos paralelos.



Um barramento de dados de 8 bits, controlado por um relógio, a transmissão de um byte a cada pulso de clock. 9 fios são usados.

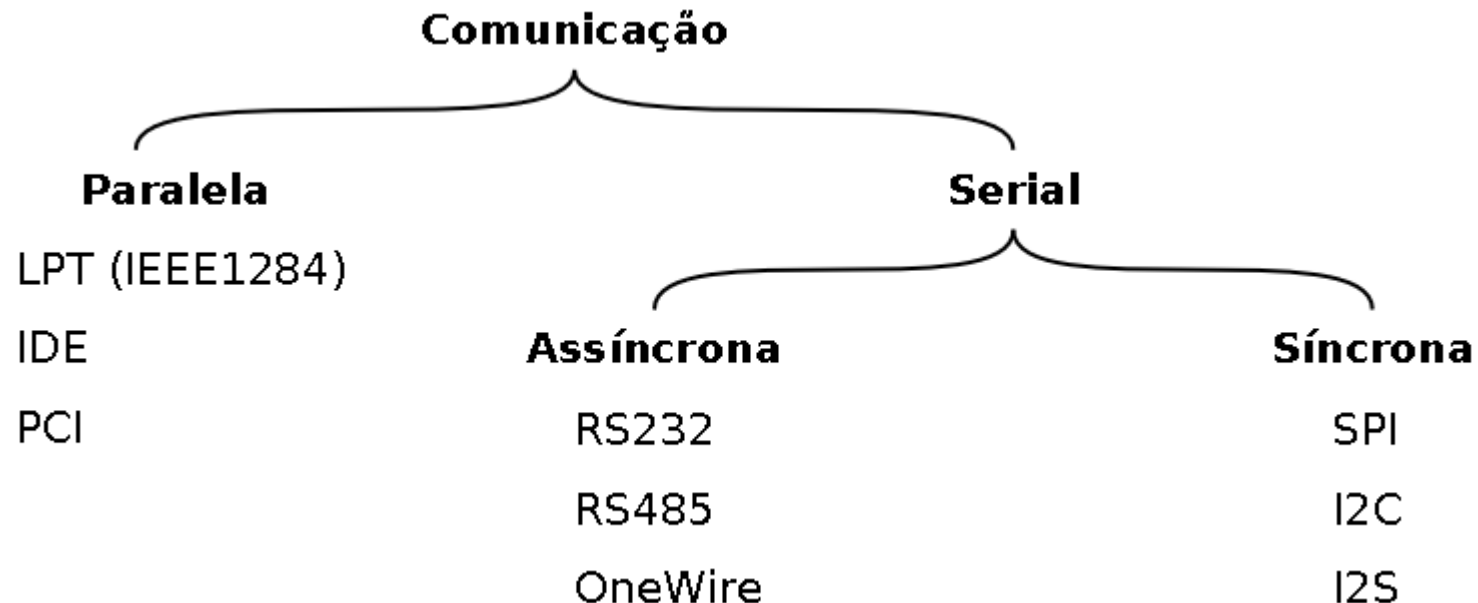


Exemplo de uma interface serial, transmitindo um bit a cada pulso de clock. Apenas 2 fios necessário!

<https://learn.sparkfun.com/tutorials/serial-communication>

Tipos de Comunicação Serial

- As tecnologias de interligação serial pode ser classificadas em: Síncronas e Assíncronas



Tipos de Comunicação Serial

- Dentre os métodos de comunicações mais conhecidos:
 - UART: Universal Asynchronous Receiver Transmitter
 - SPI: Serial Peripheral Interface
 - I2C: Inter Integrated Circuit

Tecnologia	Barramento de comunicação	Taxa máxima	Fluxo de dados
UART (RS232)	2 (sem controle de fluxo)	115.200 bps	Half ou Full Duplex
SPI	3 + nº de Slaves	2 Mbps	Full Duplex
I2C	2 (até 127 dispositivos)	400 Kbps	Half Duplex

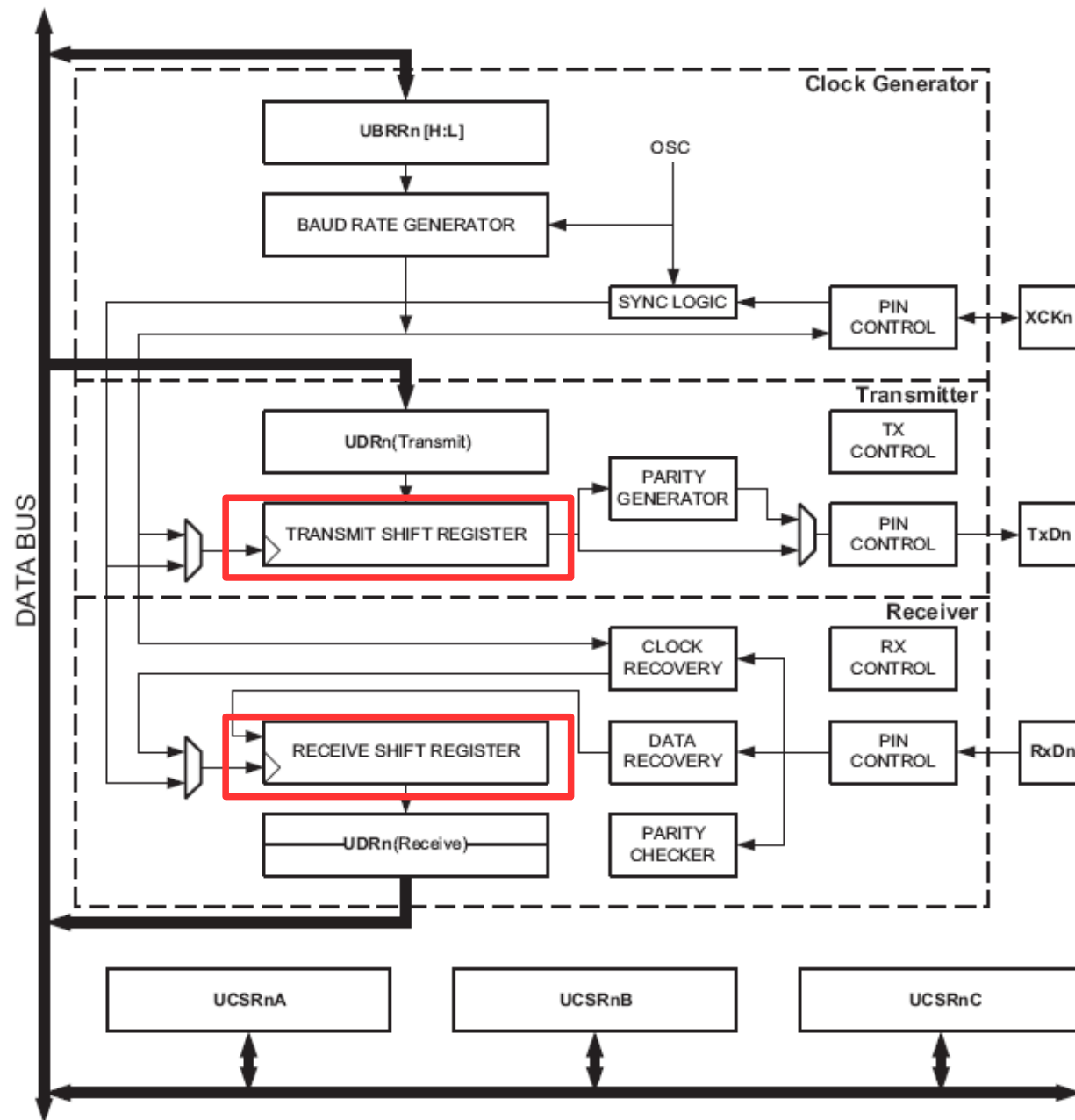
USART do ATmega328

- USART - Universal Synchronous and Asynchronous serial Receiver and Transmitter
 - Um periférico utilizado na comunicação serial dos microcontroladores com o mundo externo:
 - Na comunicação do ATmega com um computador, através de uma porta USB
 - Na comunicação entre módulos básicos de rádio frequência
 - Na comunicação com dispositivos bluetooth
 - No Arduino:
 - A USART é usada principalmente para a gravação do ATmega328 através do computador
 - A USART pode ser utilizada também para receber e enviar dados para o Arduino, através do Monitor Serial, durante a execução de um programa

USART do ATmega328

- As principais características do módulo de comunicação USART do ATmega328 são:
 - Operação Full Duplex
 - Registradores independentes de recepção e transmissão
 - Operação síncrona ou assíncrona
 - Operação síncrona com clock mestre ou escravo
 - Gerador de taxa de comunicação de alta resolução
 - Baud Rate Generator

Figure 20-1. USART Block Diagram⁽¹⁾

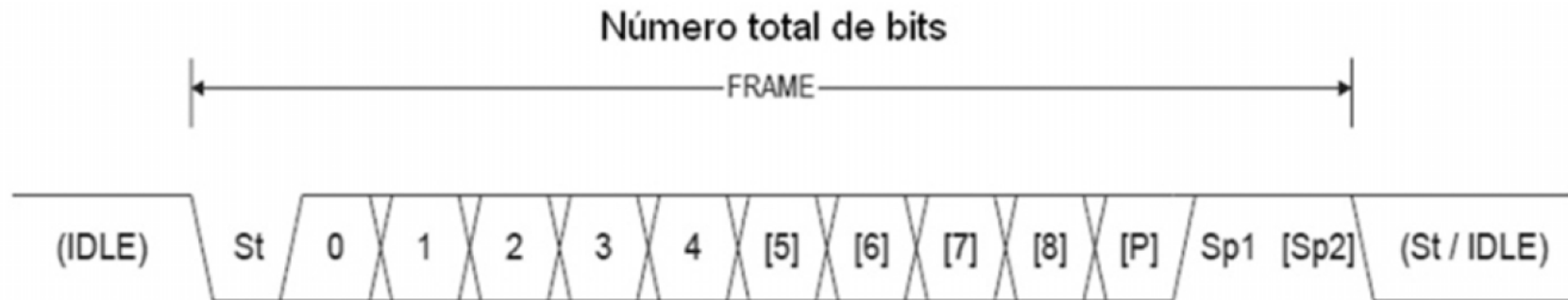


Note: 1. Refer to [Figure 1-1 on page 2](#) and [Table 14-9 on page 89](#) for USART0 pin placement.

USART do ATmega328

- As principais características do módulo de comunicação USART do ATmega328 são:
 - Suporta frames seriais com 5, 6, 7, 8 ou 9 bits de dados e 1 ou 2 bits de parada
 - Gerador de paridade par ou ímpar e conferência de paridade por hardware
 - Detecção de colisão de dados e erros de frames
 - Três fontes separadas de interrupção
 - Transmissão completa
 - Recepção completa
 - Esvaziamento do registrador de dados
 - Modo de comunicação assíncrono com velocidade duplicável

Frame USART



St Bit de Início (sempre baixo - 0)

(n) Bits de Dados (0 a 8)

P Bit de Paridade (par ou ímpar)

Sp Bit de Parada (sempre alto - 1)

IDLE Espera [sem comunicação na linha (RxD ou TxD)]. A linha deve ficar em nível lógico alto.

Fig. 15.2 – Formato do *frame* da USART.

USART do ATmega328

- As principais características do módulo de comunicação USART do ATmega328 são:
 - Para gerar a taxa de comunicação no modo mestre
 - É empregado o registrador UBRR0 (USART Baud Rate Register 0)
 - Um contador decrescente trabalhando na velocidade de clock da CPU é carregado com o valor de UBRR0 cada vez que chega a zero ou quando o UBRR0 é escrito
 - Um pulso de clock é gerado cada vez que esse contador zera, determinando a taxa de comunicação (baud rate)
 - O transmissor dividirá o clock de baudrate por 2, 8 ou 16, de acordo com o modo programado
 - A taxa de transmissão de saída é usada diretamente pela unidade de recepção e recuperação de dados

USART do ATmega328

- As principais características do módulo de comunicação USART do ATmega328 são:
 - Na tab. 15.1, são apresentadas:
 - As equações para o cálculo da taxa de comunicação (bits por segundo - bps)
 - E para o cálculo do valor de UBRR0 para cada modo de operação usando a fonte de clock interna

Tab. 15.1 – Equações para o cálculo do registrador UBRR0 da taxa de transmissão.

Modo de operação	Equação para o cálculo da taxa de transmissão	Equação para o cálculo do valor de UBRR0
Modo Normal Assíncrono (U2X0 = 0)	$TAXA = \frac{f_{osc}}{16(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{16.TAXA} - 1$
Modo de Velocidade Dupla Assíncrono (U2X0 = 1)	$TAXA = \frac{f_{osc}}{8(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{8.TAXA} - 1$
Modo Mestre Síncrono	$TAXA = \frac{f_{osc}}{2(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{2.TAXA} - 1$

UBRR0L e UBRR0H – USART *Baud Rate Registers*

Bit		15	14	13	12	11	10	9	8
	UBRR0H	-	-	-	-	UBRR[11:8]			
	UBRR0L	UBRR[7:0]							
Bit		7	6	5	4	3	2	1	0
Lê/Escreve		L/E	L	L	L	L/E	L/E	L/E	L/E
		L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

Equações para o cálculo do registrador UBRR0 de taxa de transmissão.

Modo de operação	Equação para o cálculo da taxa de transmissão	Equação para o cálculo do valor de UBRR
Modo Normal Assíncrono (U2X0 = 0)	$TAXA = \frac{f_{osc}}{16(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{16.TAXA} - 1$
Modo de Velocidade Dupla Assíncrono (U2X0 = 1)	$TAXA = \frac{f_{osc}}{8(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{8.TAXA} - 1$
Modo Mestre Síncrono	$TAXA = \frac{f_{osc}}{2(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{2.TAXA} - 1$

```
#ifndef _DEF_PRINCIPALIS_H
#define _DEF_PRINCIPALIS_H

#define F_CPU 16000000UL //define a frequencia do microcontrolador - 16MHz
```

```
#ifndef _USART_H
#define _USART_H
```

```
#include "def_principais.h"
```

```
#define BAUD 9600 //taxa de 9600 bps
#define MYUBRR F_CPU/16/BAUD-1
```

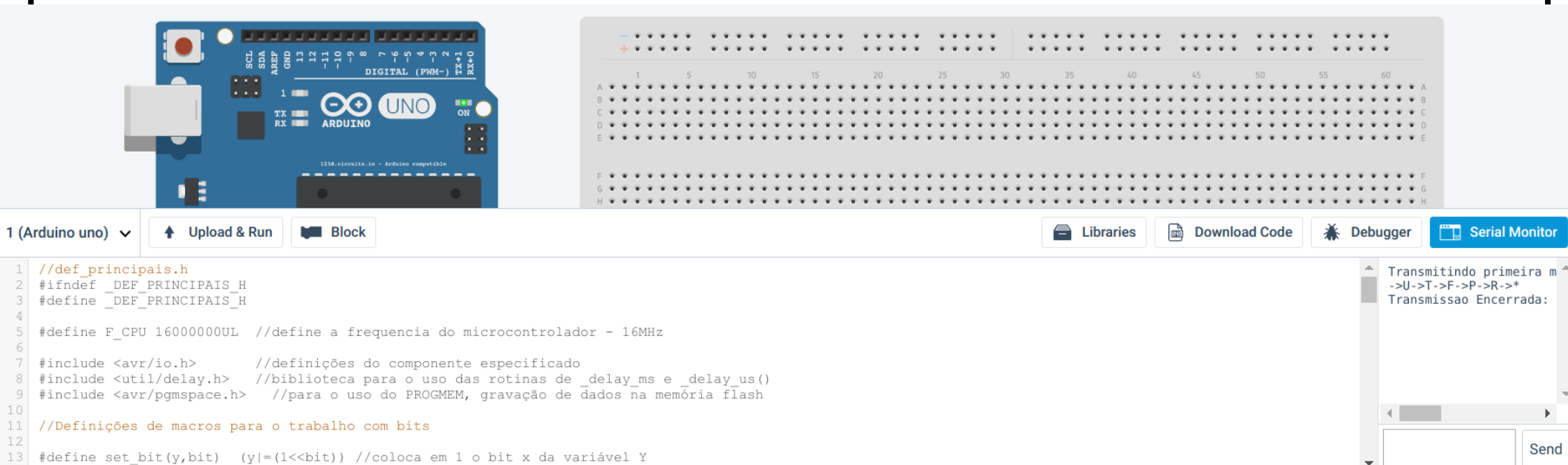
```
void USART_Init(unsigned int ubrr0)
{
    UBRR0H = (unsigned char)(ubrr0>>8); //Ajusta a taxa de transmissão
    UBRR0L = (unsigned char)ubrr0;

    UCSRA = 0; //desabilitar velocidade dupla (no Arduino é habilitado por padrão)
    UCSRB = (1<<RXEN0)|(1<<TXEN0); //Habilita a transmissão e a recepção
    UCSRC = (1<<UCSZ01)|(1<<UCSZ00); /*modo assíncrono, 8 bits de dados, 1 bit de parada, sem paridade*/
}

int main()
{
    unsigned char dado_recebido;
    |
    USART_Init(MYUBRR);

    escreve_USART_Flash(primeira_msg);
}
```


<https://www.tinkercad.com/things/eTfKB1c3F9I-usartpc>



1 (Arduino uno) ▾ Upload & Run Block Libraries Download Code Debugger Serial Monitor

```
1 //def_principais.h
2 #ifndef _DEF_PRINCIPALIS_H
3 #define _DEF_PRINCIPALIS_H
4
5 #define F_CPU 16000000UL //define a frequencia do microcontrolador - 16MHz
6
7 #include <avr/io.h> //definições do componente especificado
8 #include <util/delay.h> //biblioteca para o uso das rotinas de _delay_ms e _delay_us()
9 #include <avr/pgmspace.h> //para o uso do PROGMEM, gravação de dados na memória flash
10
11 //Definições de macros para o trabalho com bits
12
13 #define set_bit(y,bit) (y|=(1<<bit)) //coloca em 1 o bit x da variável Y
```

Transmitindo primeira m
->U->T->F->P->R->*
Transmissao Encerrada:

Send

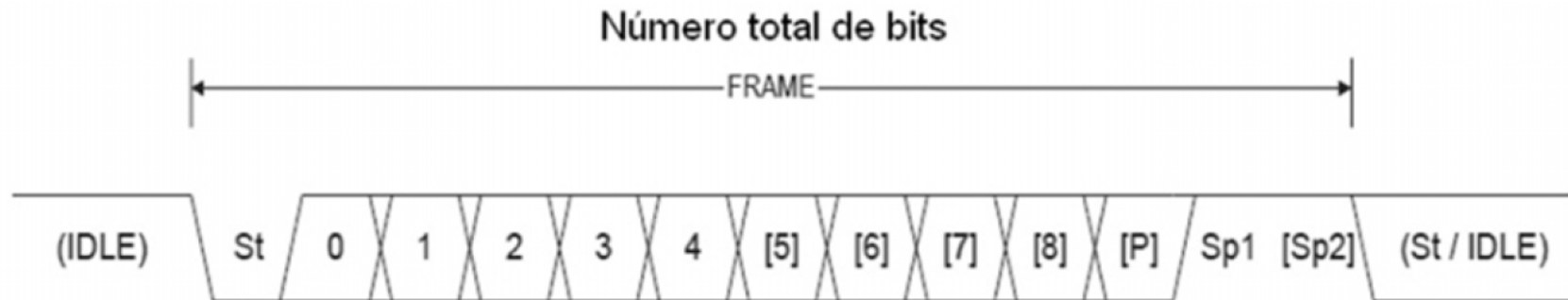
USART do ATmega328

Taxa de comunicação (bps)	$f_{osc} = 14,7456 \text{ MHz}$				$f_{osc} = 16,0000 \text{ MHz}$			
	U2X0=0		U2X0=1		U2X0=0		U2X0=1	
	UBRR0	Erro	UBRR0	Erro	UBRR0	Erro	UBRR0	Erro
2400	383	0,0%	767	0,0%	416	-0,1%	832	0,0%
4800	191	0,0%	383	0,0%	207	0,2%	416	-0,1%
9600	95	0,0%	191	0,0%	103	0,2%	207	0,2%
14,4 k	63	0,0%	127	0,0%	68	0,6%	138	-0,1%
19,2 k	47	0,0%	95	0,0%	51	0,2%	103	0,2%
28,8 k	31	0,0%	63	0,0%	34	-0,8%	68	0,6%
38,4 k	23	0,0%	47	0,0%	25	0,2%	51	0,2%
57,6 k	15	0,0%	31	0,0%	16	2,1%	34	-0,8%
76,8 k	11	0,0%	23	0,0%	12	0,2%	25	0,2%
115,2 k	7	0,0%	15	0,0%	8	-3,5%	16	2,1%
230,4 k	3	0,0%	7	0,0%	3	8,5%	8	-3,5%
250 k	3	-7,8%	6	5,3%	3	0,0%	7	0,0%
0,5 M	1	-7,8%	3	-7,8%	1	0,0%	3	0,0%
1 M	0	-7,8%	1	-7,8%	0	0,0%	1	0,0%
Máximo	921,6 kbps		1,8432 Mbps		1 Mbps		2 Mbps	

USART do ATmega328

- Frame:
 - O grupo de bits é transmitido/recebido em um bloco (frame) composto pelos:
 - bits de dados
 - bits de sincronização (bits de início e parada) e
 - opcionalmente, por um bit de paridade para a conferência de erro
 - A USART aceita várias combinações possíveis de formato de dados:
 - Um bit de início
 - 5, 6, 7, 8 ou 9 bits de dados
 - Bit de paridade par, impar ou nenhum
 - Um ou dois bits de parada

Frame USART



St Bit de Início (sempre baixo - 0)

(n) Bits de Dados (0 a 8)

P Bit de Paridade (par ou ímpar)

Sp Bit de Parada (sempre alto - 1)

IDLE Espera [sem comunicação na linha (RxD ou TxD)]. A linha deve ficar em nível lógico alto.

Fig. 15.2 – Formato do *frame* da USART.

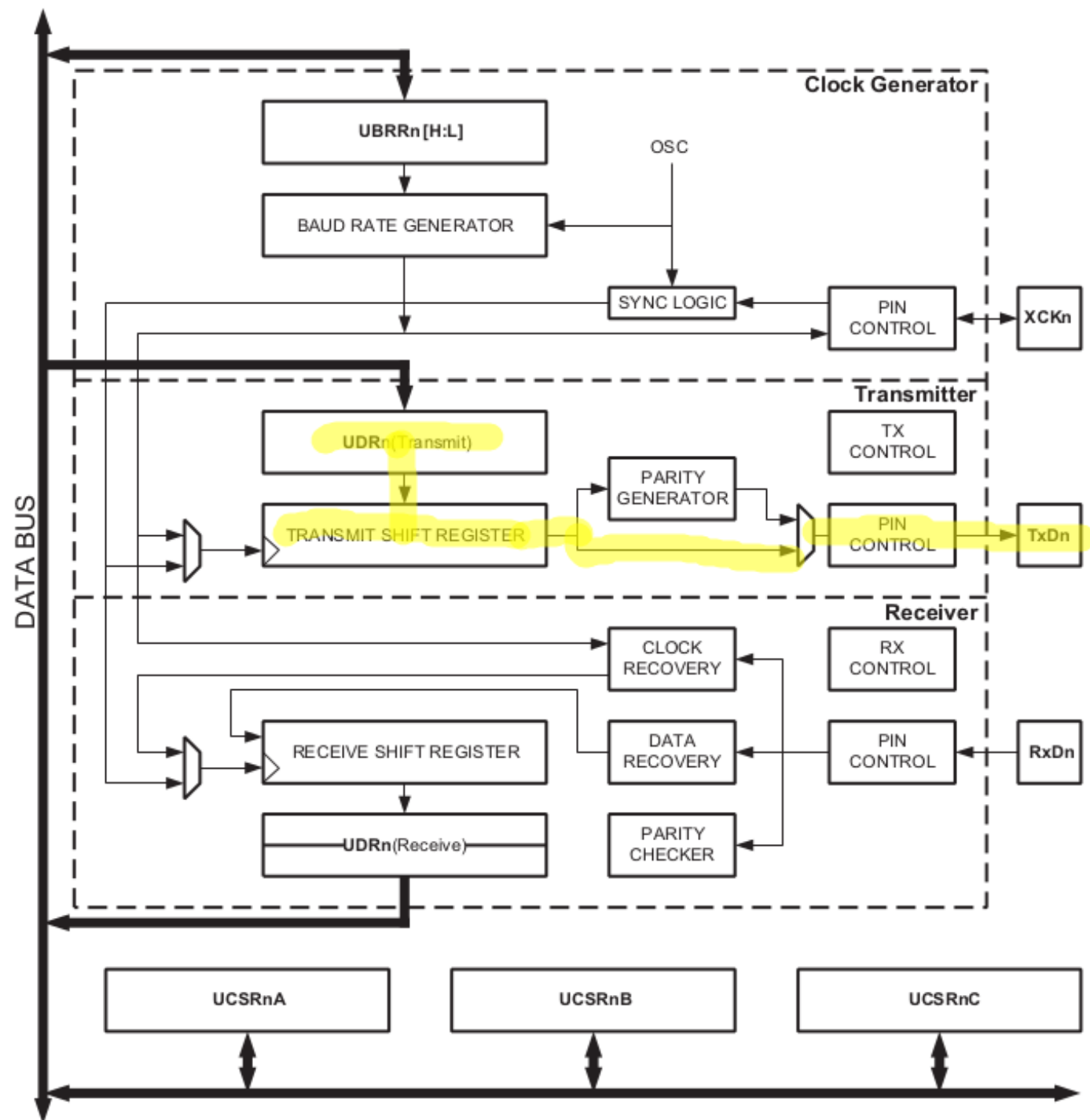
USART do ATmega328

- Um frame inicia com:
 - Um bit de início,
 - Seguido pelo bit menos significativo (LSB)
 - Seguem os outros bits de dados, num total de até 9 bits.
 - O frame termina com o bit mais significativo (MSB)
 - Se habilitado, o bit de paridade é inserido após os bits de dados, antes dos bits de parada
- Após a transmissão completa dos bits, pode-se seguir outra transmissão ou aguardar-se nova transmissão
- O formato do frame é definido:
 - Pelos bits UCSZ02:0, UPM01:0 e USBS0 nos registradores UCSR0B e UCSR0C
 - O transmissor e o receptor usam a mesma configuração
- O formato do número total de bits (frame) é ilustrado na fig. 15.2

USART do ATmega328

- Transmissão de dados:
 - Uma transmissão é iniciada ao se escrever o dado a ser transmitido no registrador de I/O da USART – UDR0
 - O dado é transferido de UDR0 para o registrador de deslocamento de transmissão quando:
 - Ele está no estado idle (ocioso)
 - Ou imediatamente após o último bit de parada do frame anterior ter sido deslocado para a saída

Figure 20-1. USART Block Diagram⁽¹⁾



USART do ATmega328

- Transmissão de dados:
 - Se o registrador de deslocamento do transmissor está vazio
 - O dado é transferido do registrador UDR0 para o registrador de deslocamento
 - Neste momento, o bit UDRE0 (USART Data Register Empty 0) no registrador UCSR0A é posto em 1 lógico
 - Quando esse bit está em 1 lógico, a USART está pronta para receber o próximo caractere

UCSR0A – USART Control and Status Register A:

Bit	7	6	5	4	3	2	1	0
UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
Lê/Escr.	L	L/E	L	L	L	L	L/E	L/E
Valor Inicial	0	0	1	0	0	0	0	0

Bit 7 – RXC0 – USART Receive Complete

Este bit é posto em 1 lógico quando um caractere recebido é transferido do registrador de deslocamento de recepção para o registrador de recepção. O bit RXC0 é posto em 0 lógico na leitura de UDR0.

Bit 6 – TXC0 – USART Transmit Complete

Este bit é posto em 1 lógico quando um caractere completo (incluindo o bit de parada) no registrador de deslocamento de transmissão for transferido e nenhum dado foi escrito em UDR0. Este bit é especialmente útil em interfaces de comunicação *half-duplex*, onde um aplicativo de transmissão deve entrar no modo de recepção e liberar o barramento de comunicação imediatamente após completar a transmissão. O bit TXC0 é posto em 0 lógico pelo hardware ao executar o vetor correspondente de tratamento de interrupção. Alternativamente, o bit TCX0 é limpo pela escrita de 1 lógico.

Bit 5 – UDRE0 – USART Data Register Empty

Este bit é posto em 1 lógico quando um caractere escrito no UDR0 é transferido para o registrador de deslocamento de transmissão. Um lógico neste bit indica que o transmissor está pronto para receber um novo caractere para transmissão. O bit UDRE0 é posto em 0 lógico na leitura de UDR0. Quando uma interrupção acionada pela transmissão de dados é usada, a rotina de interrupção por USART *Data Register Empty* deve escrever em UDR0 a fim de limpar UDRE0, se não o fizer, uma nova interrupção ocorrerá a cada vez que a rotina de interrupção terminar. UDRE0 é posto em 1 lógico durante a inicialização para indicar que o transmissor está pronto.

USART do ATmega328

- Transmissão de Dados:
 - No ciclo de clock da taxa de comunicação seguinte à operação de transferência para o registrador de deslocamento
 - O bit de início é deslocado para o pino TXD, seguindo o dado com o LSB primeiro.
 - Quando o último bit de parada é deslocado para a saída
 - O registrador de deslocamento é carregado se algum novo dado foi escrito no UDR0 durante a transmissão
 - Durante a carga, UDRE0 é posto em 1 lógico

USART do ATmega328

- Transmissão de dados:
 - Se não há nenhum dado novo no registrador UDR0 para ser enviado quando o bit de parada é deslocado
 - O flag UDRE0 permanecerá em 1 lógico até UDR0 ser escrito novamente
 - Se nenhum dado novo foi escrito e o bit de parada estiver presente em TXD pelo tempo de um bit
 - O flag de transmissão completa (TXC0) em UCSR0A é posto em 1

UCSR0A – USART Control and Status Register A:

Bit	7	6	5	4	3	2	1	0
UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
Lê/Escr.	L	L/E	L	L	L	L	L/E	L/E
Valor Inicial	0	0	1	0	0	0	0	0

Bit 7 – RXC0 – USART Receive Complete

Este bit é posto em 1 lógico quando um caractere recebido é transferido do registrador de deslocamento de recepção para o registrador de recepção. O bit RXC0 é posto em 0 lógico na leitura de UDR0.

Bit 6 – TXC0 – USART Transmit Complete

Este bit é posto em 1 lógico quando um caractere completo (incluindo o bit de parada) no registrador de deslocamento de transmissão for transferido e nenhum dado foi escrito em UDR0. Este bit é especialmente útil em interfaces de comunicação *half-duplex*, onde um aplicativo de transmissão deve entrar no modo de recepção e liberar o barramento de comunicação imediatamente após completar a transmissão. O bit TXC0 é posto em 0 lógico pelo hardware ao executar o vetor correspondente de tratamento de interrupção. Alternativamente, o bit TCX0 é limpo pela escrita de 1 lógico.

Bit 5 – UDRE0 – USART Data Register Empty

Este bit é posto em 1 lógico quando um caractere escrito no UDR0 é transferido para o registrador de deslocamento de transmissão. Um lógico neste bit indica que o transmissor está pronto para receber um novo caractere para transmissão. O bit UDRE0 é posto em 0 lógico na leitura de UDR0. Quando uma interrupção acionada pela transmissão de dados é usada, a rotina de interrupção por USART *Data Register Empty* deve escrever em UDR0 a fim de limpar UDRE0, se não o fizer, uma nova interrupção ocorrerá a cada vez que a rotina de interrupção terminar. UDRE0 é posto em 1 lógico durante a inicialização para indicar que o transmissor está pronto.

USART do ATmega328

- Transmissão de dados:
 - O bit TXEN0 no registrador UCSR0B habilita o transmissor da USART quando em 1 lógico
 - Se esse bit é posto em 0 lógico, o pino PD1 pode ser usado para I/O de dados
 - Quando TXEN0 é posto em 1 lógico
 - O transmissor da USART é conectado a PD1
 - É forçado a ser um pino de saída independente da configuração do bit DDD1 em DDRD

UCSR0B – USART Control and Status Register B:

Bit	7	6	5	4	3	2	1	0
UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 7 – RXCIE0 – RX Complete Interrupt Enable

Este bit habilita a interrupção por recepção de dados completa (bit RXC0). Uma interrupção de recepção será gerada somente se o bit RXCIE0, o bit I (SREG) e o bit RXC0 estiverem ativos.

Bit 6 – TXCIE0 – TX Complete Interrupt Enable

Este bit habilita a interrupção por transmissão de dados completa (bit TXC0). Uma interrupção de transmissão será gerada somente se o bit TXCIE0, o bit I (SREG) e o bit TXC0 estiverem ativos.

Bit 5 – UDRIE0 – USART Data Register Empty Interrupt Enable

Este bit habilita a interrupção por registrador de dados vazio (bit UDRE0). Uma interrupção por registrador de dados vazio será gerada somente se o bit UDRIE0, o bit I (SREG) e o bit UDRE0 estiverem ativos.

Bit 4 – RXEN0 – Receiver Enable

Este bit habilita a recepção da USART. O receptor irá alterar a operação normal do pino RXD. Desabilitando o receptor, ocorrerá o esvaziamento do registrador de entrada, invalidando os bits FE0, DOR0 e UPE0.

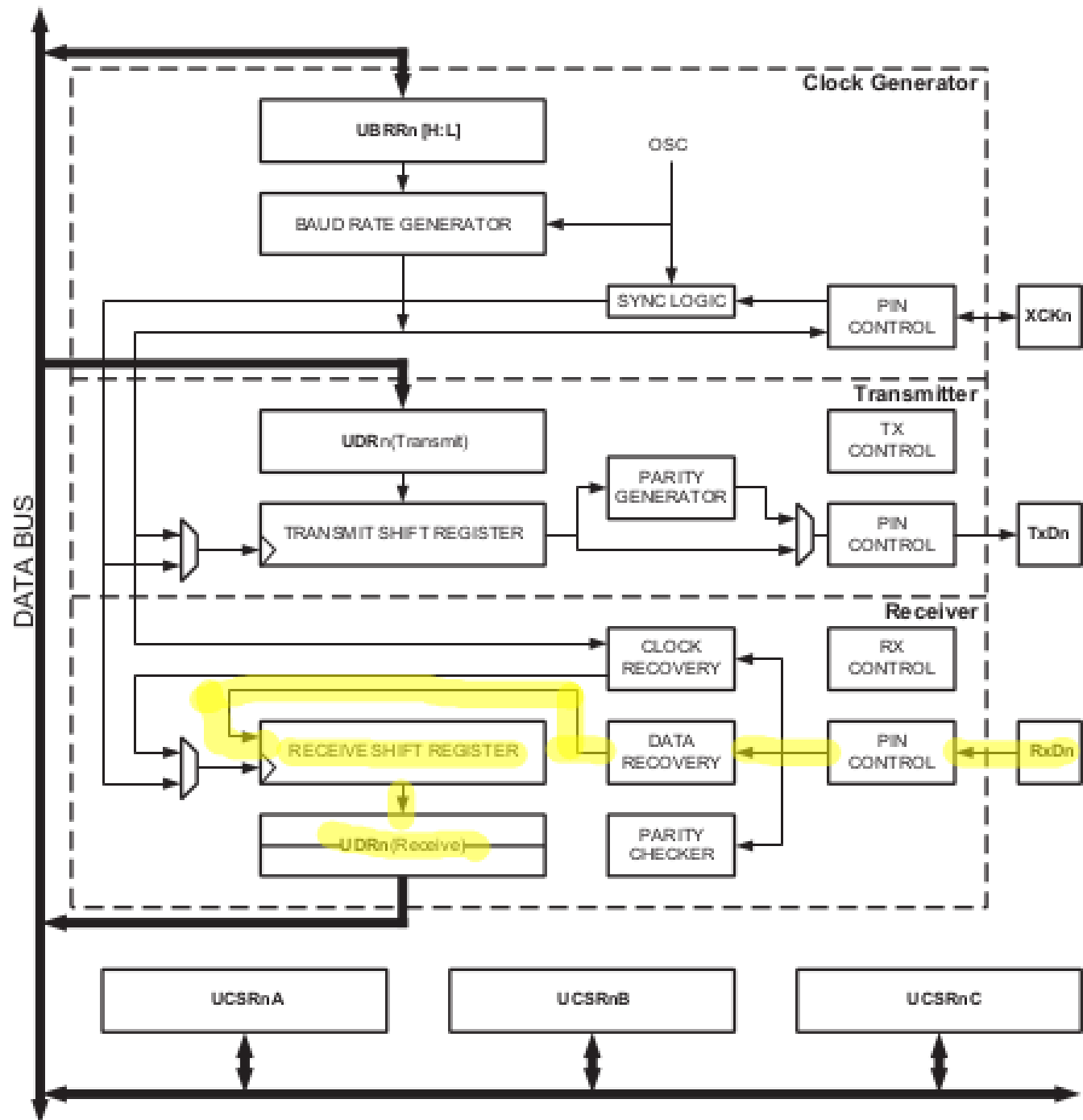
Bit 3 – TXEN0 – Transmitter Enable

Este bit habilita a transmissão da USART. O transmissor irá alterar a operação normal do pino TXD. A desabilitação do transmissor só terá efeito após as transmissões pendentes serem completadas.

USART do ATmega328

- Recepção de dados:
 - O receptor inicia a recepção de dados quando um bit de início válido é detectado
 - Os bits seguintes são amostrados
 - E deslocados para o registrador de deslocamento de recepção até o primeiro bit de parada de um frame ser recebido
 - O conteúdo do registrador de deslocamento é movido para o registrador de recepção e o bit RXC0 em UCSR0A é posto em 1 lógico
 - O dado recebido pode ser lido a partir do registrador UDR0

Figure 20-1. USART Block Diagram⁽¹⁾



UCSR0A – USART Control and Status Register A:

Bit	7	6	5	4	3	2	1	0
UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
Lê/Escr.	L	L/E	L	L	L	L	L/E	L/E
Valor Inicial	0	0	1	0	0	0	0	0

Bit 7 – RXC0 – USART Receive Complete

Este bit é posto em 1 lógico quando um caractere recebido é transferido do registrador de deslocamento de recepção para o registrador de recepção. O bit RXC0 é posto em 0 lógico na leitura de UDR0.

Bit 6 – TXC0 – USART Transmit Complete

Este bit é posto em 1 lógico quando um caractere completo (incluindo o bit de parada) no registrador de deslocamento de transmissão for transferido e nenhum dado foi escrito em UDR0. Este bit é especialmente útil em interfaces de comunicação *half-duplex*, onde um aplicativo de transmissão deve entrar no modo de recepção e liberar o barramento de comunicação imediatamente após completar a transmissão. O bit TXC0 é posto em 0 lógico pelo hardware ao executar o vetor correspondente de tratamento de interrupção. Alternativamente, o bit TCX0 é limpo pela escrita de 1 lógico.

Bit 5 – UDRE0 – USART Data Register Empty

Este bit é posto em 1 lógico quando um caractere escrito no UDR0 é transferido para o registrador de deslocamento de transmissão. Um lógico neste bit indica que o transmissor está pronto para receber um novo caractere para transmissão. O bit UDRE0 é posto em 0 lógico na leitura de UDR0. Quando uma interrupção acionada pela transmissão de dados é usada, a rotina de interrupção por USART *Data Register Empty* deve escrever em UDR0 a fim de limpar UDRE0, se não o fizer, uma nova interrupção ocorrerá a cada vez que a rotina de interrupção terminar. UDRE0 é posto em 1 lógico durante a inicialização para indicar que o transmissor está pronto.

USART do ATmega328

- Recepção de dados:
 - Quando o bit RXEN0 no registrador UCSR0B está em 0 lógico, a recepção está desabilitada
 - Significa que o terminal PD0 pode ser usado como um pino de I/O
 - Quando RXEN0 está em 1 lógico
 - O receptor da USART está conectado a PD0
 - O que o força a ser uma entrada, independente da configuração do bit DDD0 em DDRD
 - Quando PD0 é forçado como entrada pela USART
 - O PORTD pode ainda ser usado para controlar o resistor de pull-up do pino

UCSR0B – USART Control and Status Register B:

Bit	7	6	5	4	3	2	1	0
UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 7 – RXCIE0 – RX Complete Interrupt Enable

Este bit habilita a interrupção por recepção de dados completa (bit RXC0). Uma interrupção de recepção será gerada somente se o bit RXCIE0, o bit I (SREG) e o bit RXC0 estiverem ativos.

Bit 6 – TXCIE0 – TX Complete Interrupt Enable

Este bit habilita a interrupção por transmissão de dados completa (bit TXC0). Uma interrupção de transmissão será gerada somente se o bit TXCIE0, o bit I (SREG) e o bit TXC0 estiverem ativos.

Bit 5 – UDRIE0 – USART Data Register Empty Interrupt Enable

Este bit habilita a interrupção por registrador de dados vazio (bit UDRE0). Uma interrupção por registrador de dados vazio será gerada somente se o bit UDRIE0, o bit I (SREG) e o bit UDRE0 estiverem ativos.

Bit 4 – RXEN0 – Receiver Enable

Este bit habilita a recepção da USART. O receptor irá alterar a operação normal do pino RXD. Desabilitando o receptor, ocorrerá o esvaziamento do registrador de entrada, invalidando os bits FE0, DOR0 e UPE0.

USART

UDR0 – USART I/O Data Register 0

Bit	7	6	5	4	3	2	1	0
UDR0 (leitura)	RXB[7:0]							
UDR0 (escrita)	TXB[7:0]							
Lê/Escreve	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

UCSR0A – USART Control and Status Register 0 A

Bit	7	6	5	4	3	2	1	0
UCSR0A	RXC0	TXC0	UDRE0	FE	DOR0	UPE0	U2X0	MPCM0
Lê/Escr.	L	L/E	L	L	L	L	L/E	L/E
Valor Inicial	0	0	1	0	0	0	0	0

RXC0 = 1, recepção completa, limpo quando UDR é lido.

TXC0 = 1, *frame* enviado, nenhum *frame* novo, limpo também pela escrita de 1.

UDRE0 = 1, UDR vazio, pronto para novo dado.

FE0 = 1, *stop* bit recebido = 0.

DOR0 = 1, UDR não foi lido e novo *start* bit é detectado.

PE0 = 1, erro de paridade no byte recebido.

U2X0 = 1, dobra a taxa no modo assíncrono.

MPCM0 = 1, modo multiprocessador ativo.

UCSR0B – USART *Control and Status Register 0 B*

Bit	7	6	5	4	3	2	1	0
UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
Lê/Escr.	L/E	L/E	L/E	L/E	L/E	L/E	L	L/E
Valor Inicial	0	0	0	0	0	0	0	0

RXCIE0 = 1, habilita a interrupção por recepção completa.

TXCIE0 = 1, habilita a interrupção por transmissão completa.

UDRIE0 = 1, habilita a interrupção por UDR vazio.

RXEN0 = 1, habilita a recepção da USART.

TXEN0 = 1, habilita a transmissão da USART.

RXB80/TXB80 = nono bit recebido/transmitido.

UCSR0C - USART Control and Status Register 0 C

Bit	7	6	5	4	3	2	1	0
UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
Lê/Escr.	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	1	1	0

UMSEL01:0 - selecionam o modo de operação da USART.

UPM01:0 - selecionam o modo de paridade.

USBS0 = 1, dois bits de parada.

UCSZ01:0 - determinam o número de bits de um *frame*.

UCPOL0 – válido para o modo síncrono, ajusta a relação entre a alteração do dado transmitido, a amostragem do dado recebido e o *clock* síncrono.

Ajuste do modo de operação da USART.

UMSEL01	UMSEL00	Modo de operação
0	0	assíncrono
0	1	síncrono
1	0	reservado
1	1	SPI mestre

UCSR0C - USART Control and Status Register 0 C

Bit	7	6	5	4	3	2	1	0
UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
Lê/Escr.	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	1	1	0

UMSEL01:0 - selecionam o modo de operação da USART.

UPM01:0 - selecionam o modo de paridade.

USBS0 = 1, dois bits de parada.

UCSZ01:0 - determinam o número de bits de um *frame*.

UCPOL0 – válido para o modo síncrono, ajusta a relação entre a alteração do dado transmitido, a amostragem do dado recebido e o *clock* síncrono.

Ajuste do modo de paridade.

UPM01	UPM00	Modo de Paridade
0	0	desabilitado
0	1	reservado
1	0	habilitado, paridade par
1	1	habilitado, paridade ímpar

UCSR0C - USART Control and Status Register 0 C

Bit	7	6	5	4	3	2	1	0
UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
Lê/Escr.	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	1	1	0

UMSEL01:0 - selecionam o modo de operação da USART.

UPM01:0 - selecionam o modo de paridade.

USBS0 = 1, dois bits de parada.

UCSZ01:0 - determinam o número de bits de um *frame*.

UCPOL0 – válido para o modo síncrono, ajusta a relação entre a alteração do dado transmitido, a amostragem do dado recebido e o *clock* síncrono.

Ajuste do Frame.

UCSZ02	UCSZ01	UCSZ00	Tamanho do Caractere
0	0	0	5 bits
0	0	1	6 bits
0	1	0	7 bits
0	1	1	8 bits
1	0	0	reservado
1	0	1	reservado
1	1	0	reservado
1	1	1	9 bits

UCSR0C - USART Control and Status Register 0 C

Bit	7	6	5	4	3	2	1	0
UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
Lê/Escr.	L/E	L/E	L/E	L/E	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	1	1	0

UMSEL01:0 - selecionam o modo de operação da USART.

UPM01:0 - selecionam o modo de paridade.

USBS0 = 1, dois bits de parada.

UCSZ01:0 - determinam o número de bits de um *frame*.

UCPOL0 – válido para o modo síncrono, ajusta a relação entre a alteração do dado transmitido, a amostragem do dado recebido e o *clock* síncrono.

Ajuste da polaridade do *clock* síncrono.

UCPOL0	Mudança do Dado Transmitido (saída do pino TxD0)	Amostragem do Dado Recebido (entrada do pino RxD0)
0	Borda de subida de XCK.	Borda de descida de XCK.
1	Borda de descida de XCK.	Borda de subida de XCK.

USART.h (arquivo de cabeçalho do USART.c)

```
#ifndef _USART_H
#define _USART_H

#include "def_principais.h"

#define BAUD 9600 //taxa de 9600 bps
#define MYUBRR F_CPU/16/BAUD-1
#define tam_vetor 5 //número de dígitos individuais para a conversão por ident_num()
#define conv_ascii 48 /*48 se ident_num() deve retornar um número no formato ASCII
(0 para formato normal)*/

void USART_Inic(unsigned int ubbr0);
void USART_Transmite(unsigned char dado);
unsigned char USART_Recebe();
void escreve_USART(char *c);
```

Taxa de comunicação (bps)	f _{osc} = 14,7456 MHz				f _{osc} = 16,0000 MHz			
	U2X0=0		U2X0=1		U2X0=0		U2X0=1	
	UBRR0	Erro	UBRR0	Erro	UBRR0	Erro	UBRR0	Erro
2400	383	0,0%	767	0,0%	416	-0,1%	832	0,0%
4800	191	0,0%	383	0,0%	207	0,2%	416	-0,1%
9600	95	0,0%	191	0,0%	103	0,2%	207	0,2%
14,4 k	63	0,0%	127	0,0%	68	0,6%	138	-0,1%
19,2 k	47	0,0%	95	0,0%	51	0,2%	103	0,2%
28,8 k	31	0,0%	63	0,0%	34	-0,8%	68	0,6%
38,4 k	23	0,0%	47	0,0%	25	0,2%	51	0,2%
57,6 k	15	0,0%	31	0,0%	16	2,1%	34	-0,8%
76,8 k	11	0,0%	23	0,0%	12	0,2%	25	0,2%
115,2 k	7	0,0%	15	0,0%	8	-3,5%	16	2,1%
230,4 k	3	0,0%	7	0,0%	3	8,5%	8	-3,5%
250 k	3	-7,8%	6	5,3%	3	0,0%	7	0,0%
0,5 M	1	-7,8%	3	-7,8%	1	0,0%	3	0,0%
1 M	0	-7,8%	1	-7,8%	0	0,0%	1	0,0%
Máximo	921,6 kbps		1,8432 Mbps		1 Mbps		2 Mbps	

USART do ATmega328

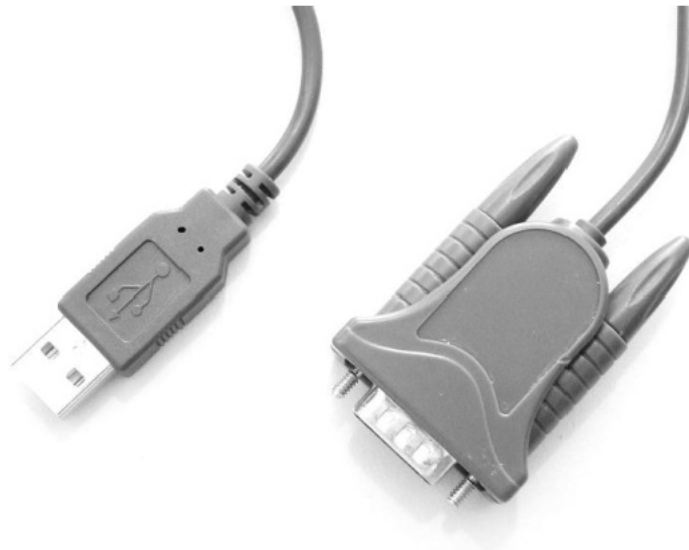


Fig. 15.3 – Terminais de um cabo para ligar um dispositivo com comunicação RS232 a um computador com entrada USB.

USART do ATmega328

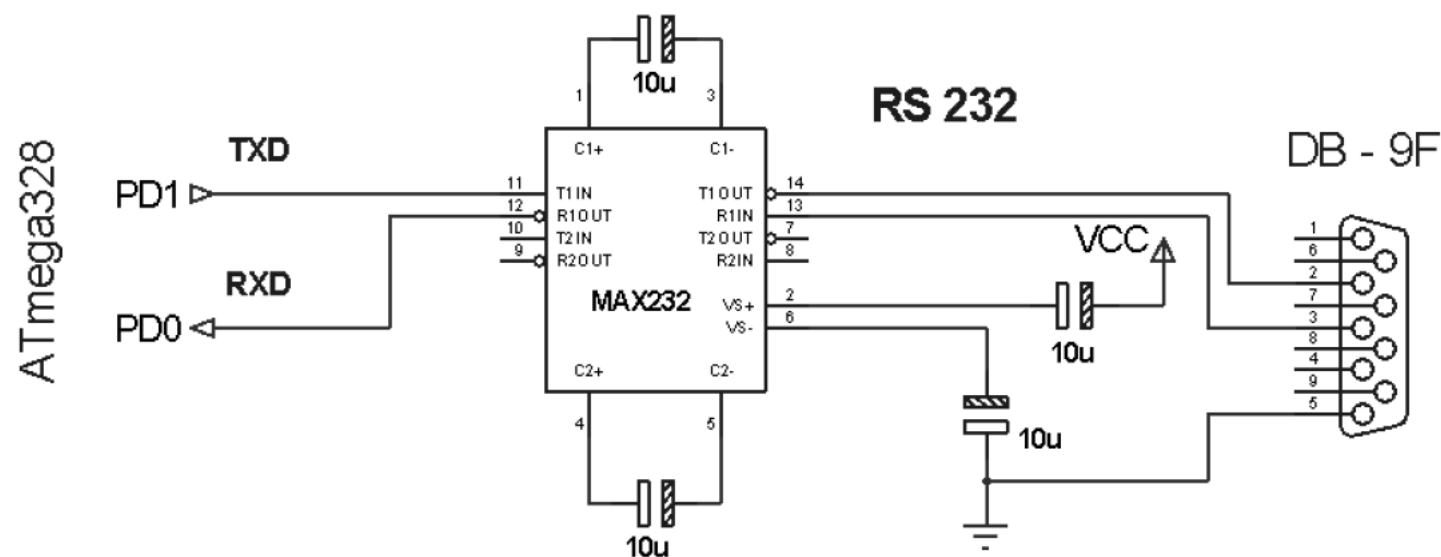


Fig. 15.5 – Circuito para adequação da tensão proveniente de um microcontrolador para a comunicação RS232.

USART do ATmega328



<http://mcukits.com/2009/03/12/assembling-the-freeduino-serial-v20-board-max232/>

Referências

- **AVR e Arduino – Técnicas de Projeto.**

- **-15. USART**

- **Datasheet - Atmel 8-bit Microcontroller with 4/8/16/32KBytes**

- **<http://www.embarcados.com.br/arduino-rs-232-rs-485-hardware/>**

- **<https://learn.sparkfun.com/tutorials/serial-communication>**

-