

15. USART

Neste capítulo, é apresentada a USART, um periférico com características muito flexíveis e amplamente utilizado na comunicação serial dos microcontroladores com o mundo externo. O seu emprego é exemplificado na comunicação do ATmega com um computador, através de uma porta USB, na comunicação entre módulos básicos de rádio frequência e na comunicação com dispositivos *bluetooth*. No Arduino, a USART é usada principalmente para a gravação do ATmega328 através do computador.

A *Universal Synchronous and Asynchronous serial Receiver and Transmitter* é um módulo de comunicação serial com inúmeras possibilidades de configurações de trabalho, o que lhe permite ser aplicada em uma infinidade de sistemas eletrônicos. Como por exemplo, nas comunicações RS232 e RS485, que apesar de não serem mais utilizadas em computadores pessoais, são, ainda, largamente usadas em sistemas industriais de controle. A grande vantagem da USART é que muitos dispositivos eletrônicos modernos suportam seu protocolo de comunicação.

15.1 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*).

- 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*.
- Filtro para ruído, incluindo bit de início falso e filtro digital passa-baixa.
- Três fontes separadas de interrupção (transmissão completa, recepção completa e esvaziamento do registrador de dados).
- Modo de comunicação assíncrono com velocidade duplicável.
- Pode ser utilizada como interface SPI mestre.

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. Assim, 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 *baud rate* 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. 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 (para valores previamente calculados, ver a tab. C1 do apêndice C).

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	$TAXA = \frac{f_{osc}}{2(UBRR0 + 1)}$	$UBRR0 = \frac{f_{osc}}{2.TAXA} - 1$

Para duplicar a taxa de comunicação no modo assíncrono, basta ativar o bit U2X0 do registrador UCSR0A. Esse bit deve ser colocado em zero na operação síncrona.

Emprega-se um sinal externo de *clock* no pino XCK para os modos síncronos de recepção escravo. A frequência máxima desse sinal está limitada pelo tempo de resposta da CPU, devendo seguir a seguinte condição:

$$f_{XCK} < \frac{f_{osc}}{4} \quad (15.1)$$

A frequência do sistema (f_{osc}) depende da estabilidade da fonte de *clock*. É recomendado usar alguma margem de segurança na eq. 15.1 para evitar uma possível perda de dados.

Quando o modo síncrono é usado (UMSEL0=1), o pino XCK será empregado como *clock* de entrada (escravo) ou saída (mestre). O princípio básico é que o dado de entrada (no pino RXD) é amostrado na borda oposta do XCK quando a borda do dado de saída é alterada (pino TXD). O bit UCPOL0 do registrador UCRS0C seleciona qual borda de XCK é usada para a amostragem do dado e qual é usada para a mudança do dado. Na fig. 15.1, o efeito do bit UCPOL0 é apresentado.

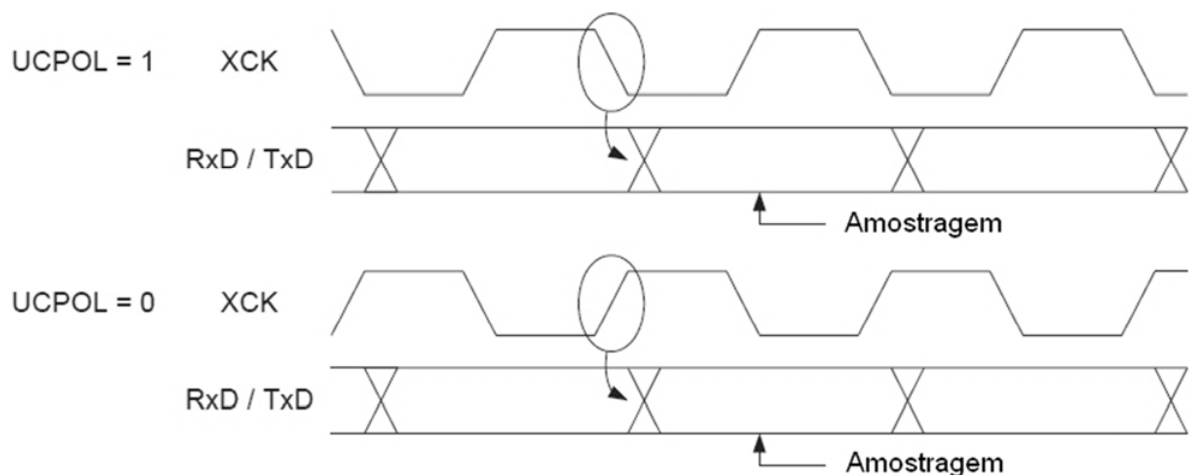


Fig. 15.1 – Efeito do bit UCPOL0 na amostragem dos dados.

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, ímpar ou nenhum.
- Um ou dois bits de parada.

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 número total de bits (*frame*) é ilustrado na fig. 15.2.

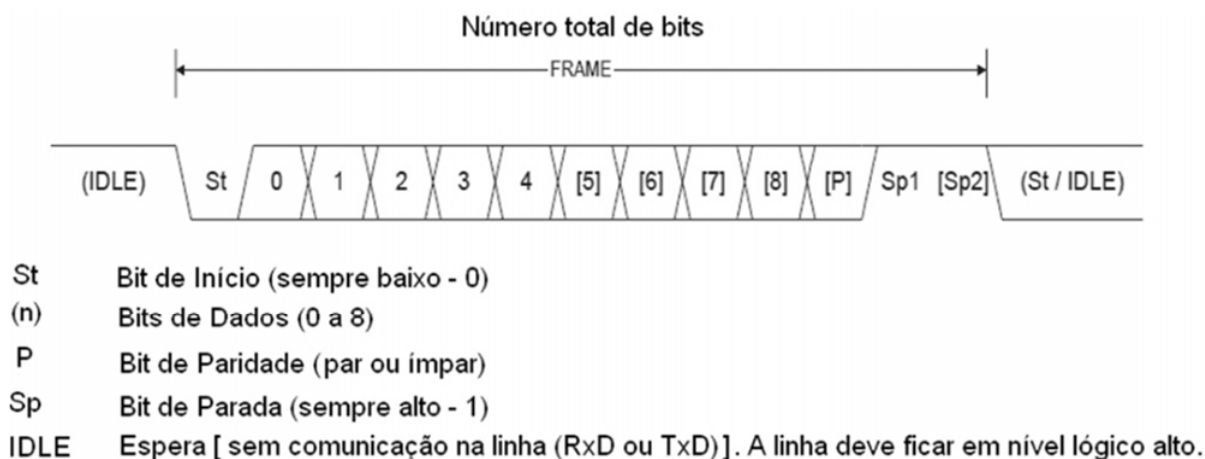


Fig. 15.2 – Formato do *frame* da USART.

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.

Antes de qualquer comunicação, a USART deve ser inicializada. Os detalhes de configuração necessitam ser ajustados nos registradores específicos, detalhados posteriormente. A seguir, são apresentados exemplos para trabalho com a USART a partir de códigos extraídos do manual do fabricante.

```
//===== //
//          INICIALIZANDO A USART          //
//===== //
#define FOSC    1843200          //Frequência de trabalho da CPU
#define BAUD    9600
#define MYUBRR  FOSC/16/BAUD-1
//-----
void main(void)
{
    ...
    USART_Init(MYUBRR);
    ...
}
//-----
void USART_Init(unsigned int ubrr)
{
    UBRR0H = (unsigned char)(ubrr>>8); //Ajusta a taxa de transmissão
    UBRR0L = (unsigned char)ubrr;
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);    //Habilita o transmissor e o receptor
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);    //Ajusta o formato do frame:
                                         //8 bits de dados e 2 de parada
}
//=====

//===== //
//          ENVIANDO FRAMES COM 5 A 8 BITS          //
//===== //
void USART_Transmit(unsigned char data)
{
    while(!(UCSR0A & (1<<UDRE0))); //Espera a limpeza do registr. de transmissão
    UDR0 = data;                    //Coloca o dado no registrador e o envia
}
//=====

//===== //
//          ENVIANDO FRAMES COM 9 BITS          //
//===== //
void USART_Transmit(unsigned int data)
{
    while(!(UCSR0A & (1<<UDRE0))); //Espera a limpeza do registr. de transmissão

    UCSR0B &= ~(1<<TXB80);          //Copia o 9º bit para o TXB8

    if(data & 0x0100)
        UCSR0B |= (1<<TXB80);

    UDR0 = data;                    //Coloca o dado no registrador e o envia
}
//=====
```

```

//===== //
//          RECEBENDO FRAMES COM 5 A 8 BITS          //
//===== //
unsigned char USART_Receive(void)
{
    while(!(UCSR0A & (1<<RXC0))); //Espera o dado ser recebido
    return UDR0;                  //Lê o dado recebido e retorna
}
//=====

//===== //
//          RECEBENDO FRAMES COM 9 BITS          //
//===== //
unsigned int USART_Receive(void)
{
    unsigned char status, resh, resl; //Espera o dado ser recebido
    while(!(UCSR0A & (1<<RXC0))); //Obtêm o status do 9º bit, então, o dado do registr.
    status = UCSR0A;
    resh = UCSR0B;
    resl = UDR0;

    if(status & (1<<FE0)|(1<<DOR0)|(1<<UPE0)) //Se ocorrer um erro retorna -1
        return -1;

    resh = (resh >> 1) & 0x01; //Filtra o 9º bit, então, retorna
    return ((resh << 8) | resl);
}
//=====

//===== //
//          LIMPANDO O REGISTRADOR DE ENTRADA (quando ocorre um erro p. ex.) //
//===== //
void USART_Flush(void)
{
    unsigned char dummy;
    while(UCSR0A & (1<<RXC0)) dummy = UDR0;
}
//=====

```

A faixa de operação do receptor é dependente do descasamento entre a taxa de comunicação do transmissor e a sua taxa de comunicação, gerada internamente (*baud rate*). Se o transmissor estiver enviando bits muito rapidamente, ou muito devagar, ou ainda, se o *clock* gerado internamente não tiver a frequência base similar a do transmissor, o receptor não será capaz de sincronizar os *frames* relativos ao bit de início. Isso gerará um erro na taxa de recepção, que, segundo a Atmel, deve estar entre $\pm 1\%$ e $\pm 3\%$ (ver manual do componente, tabelas detalhadas são apresentadas no apêndice C). O erro é calculado por:

$$Erro[\%] = \frac{Taxa\ de\ Transmissão\ Aproximada}{Taxa\ de\ Transmissão\ Ideal} \cdot 100\% \quad (15.2)$$

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.

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.

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. 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.

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, o qual é forçado a ser um pino de saída independente da configuração do bit DDD1 em DDRD.

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. Neste momento, 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.

Quando o bit RXEN0 no registrador UCSR0A está em 0 lógico, a recepção está desabilitada. Isso 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.

REGISTRADORES DA USART

UDR0 – USART I/O Data Register.

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	0	0	0	0	0	0	0	0
Inicial								

Os registradores de recebimento e envio de dados compartilham o mesmo endereço lógico. Entretanto, uma leitura é feita no UDR0 de leitura e uma escrita, no UDR0 de escrita, o hardware se encarrega da distinção. Para frames com 5, 6 ou 7 bits, os bits não utilizados serão ignorados e na recepção colocados em zero. O UDR0 só deve ser escrito quando o bit UDRE0 do registrador UCSR0A estiver ativo.

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.

Bit 4 – FE0 – Frame Error

Indica se existe um erro no *frame* recebido. Este bit deve sempre ser zerado quando se escreve no registrador UCSR0A.

Bit 3 – DOR0 – Data OverRun

Ocorre quando o registrador de entrada está cheio, não foi lido e um novo bit de início é detectado. Este bit deve sempre ser zerado quando se escreve no registrador UCSR0A.

Bit 2– UPE0 – USART Parity Error

Indica se existe um erro de paridade no dado recebido e fica ativo até UDR0 ser lido. Este bit deve sempre ser zerado quando se escreve no registrador UCSR0A.

Bit 1 – U2X0 – Double the USART transmission speed

Este bit só tem efeito no modo de operação assíncrona.

Bit 0 –MPCM0 – Multi-processor Communication Mode

Habilita a comunicação com vários processadores. Quando ativo, todos os *frames* recebidos serão ignorados se não contiverem uma informação de endereço.

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.

Bit 2 – UCSZ02 – Character Size

Este bit, combinado com os bits UCSZ01:0 do registrador UCSR0C, ajusta o número de bits de dados do *frame*.

Bit 1 – RXB80 – Receive Data Bit 8

É o nono bit de dados recebidos quando o *frame* for de 9 bits. Deve ser lido antes dos outros bits do UDR0.

Bit 0 – TXB80 – Transmit Data Bit 8

É o nono bit de dados a ser transmitido quando o *frame* for de 9 bits. Deve ser escrito antes dos outros bits no UDR0.

UCSR0C - USART Control and Status Register C:

Bit	7	6	5	4	3	2	1	0
UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
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	1	1	0

Bit 7:6 – UMSEL01:0 – USART Mode Select

Estes bits selecionam o modo de operação da USART, conforme tab. 15.2.

Tab. 15.2 – Ajuste dos bits UMSEL01:0 para o modo de operação da USART.

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

Bits 5:4 – UPM01:0 – Parity Mode

Estes bits habilitam e ajustam o gerador de paridade e de conferência. Se habilitado, o transmissor irá gerar e enviar automaticamente o bit de paridade em cada *frame*; o receptor irá gerar o valor de paridade para comparação. Se uma desigualdade for detectada, o bit UPE0 torna-se ativo. Na tab. 15.3, são apresentadas as possíveis configurações para os bits UPM01:0.

Tab. 15.3 – Bits UPM01:0.

UPM01	UPM00	Modo de Paridade
0	0	Desabilitado
0	1	Reservado
1	0	Habilitado, paridade par
1	1	Habilitado, paridade ímpar

Bit 3 – USBS0 – Stop Bit Select

Este bit seleciona o número de bits de parada a serem inseridos pelo transmissor (USBS0=0, 1 bit de parada; USBS0=1, 2 bits de parada).

Bits 2:1 – UCSZ01:0 – Character Size

Estes bits, combinados com o bit UCSZ02 do registrador UCSR0B, ajustam o número de bits de dados no *frame* do transmissor e receptor, conforme tab. 15.4.

Tab. 15.4 – Ajuste dos bits UCSZ01:0.

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

Bit 0 – UCPOL0 – Clock Polarity

Este bit é usado somente no modo síncrono. Deve ser zero quando o modo assíncrono é usado. Ele ajusta o sinal de *clock* (XCK) para amostragem e saída de dados, conforme tab. 15.5.

Tab. 15.5 – Ajustando a polaridade do *clock*.

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

UBRR0L e UBRR0H - USART Baud Rate Register

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	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

Bits 15:12 – Reservado

Devem ser zero quando se escreve em UBRR0H.

Bits 11:0 – UBRR011:0: USART Baud Rate Register

Este é o registrador de 12 bits que contém o valor da taxa de comunicação. Qualquer transmissão em andamento será corrompida se houver mudança desse valor. Qualquer escrita atualiza imediatamente a taxa de comunicação, ver a tab. 15.1.