

# EA871 – Laboratório de Programação Básica de Sistemas Digitais

Atividade 05 – 2º semestre de 2023

## 1. Objetivos

- Aprender a configurar os registradores da USART a fim de atingir um modo de operação desejado.
- Empregar a USART e a estratégia de varredura para transmitir uma sequência de caracteres para o computador.

## 2. Atividade de aula

Determine as configurações dos registradores da USART a fim de que ela opere segundo as seguintes especificações:

Especificações do protocolo:

- Baud rate igual a 115.200 bps.
- Número de bits de dados por frame igual a 8;
- Uso de um bit de paridade (par);
- Uso de dois bits de parada;

Especificações adicionais (particulares para a USART do ATmega328P):

- Velocidade de transmissão normal (i.e., modo double-speed desativado);
- Modo de transmissão multi-processor desabilitado;
- Todas as interrupções relacionadas à USART desabilitadas;
- Somente o transmissor ativado;
- Modo assíncrono de funcionamento da USART;

Se desejar, utilize as tabelas abaixo para preencher as configurações dos registradores.

Bit	7	6	5	4	3	2	1	0
UBRR0H								

Bit	7	6	5	4	3	2	1	0
UBRR0L								

Bit	7	6	5	4	3	2	1	0
UCSR0A								

Bit	7	6	5	4	3	2	1	0
UCSR0B								

Bit	7	6	5	4	3	2	1	0
UCSR0C								

## 3. Resumo da atividade para entrega

O desafio proposto nesta atividade é a construção de um programa que utiliza a USART para transmitir uma mensagem ao computador de acordo com o estado de um botão conectado ao terminal PB1 do microcontrolador. **O monitoramento do botão e a verificação do término da transmissão de cada caractere devem ser implementados utilizando a estratégia de varredura.**

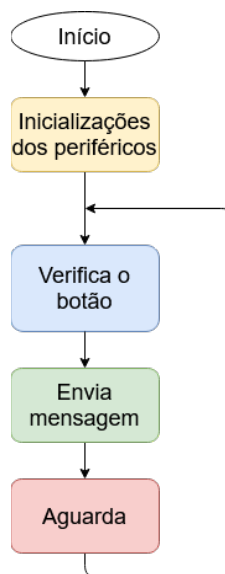
O botão deve ser monitorado todo o tempo e, se ele não está pressionado, deve ser enviada a seguinte mensagem para o computador:

```
char msg_np[] = "Transmissao serial utilizando a USART: verificacao do termino da transmissao por varredura. O botao nao esta pressionado.\n\n";
```

Cada vez que a sequência inteira de caracteres da mensagem for transmitida, o programa deve aguardar **500ms** (use a função `_delay_ms()` para produzir a espera) antes de enviar a próxima mensagem. Se o botão estiver pressionado, a seguinte mensagem deve ser enviada:

```
char msg_p[] = "Transmissao serial utilizando a USART: verificacao do termino da transmissao por varredura. O botao esta pressionado.\n\n";
```

Utilize o resistor de *pullup* interno da porta no acionamento do botão, como feito na Atividade 3. Não é necessário filtrar o transitório do botão. As configurações da USART são as mesmas discutidas na atividade discutida em aula. O fluxograma a seguir sumariza o funcionamento do sistema.



#### Importante:

- 1) Antes de enviar o próximo caractere da mensagem, é necessário verificar se a transmissão do caractere atual já terminou, o que deve ser feito através da estratégia de varredura (*polling*). Consulte o manual para entender como funciona o mecanismo de sinalização do evento de término de transmissão. Há mais de uma opção, escolha a que julgar conveniente. **Não é permitido o uso de atrasos para aguardar o término da transmissão de cada caractere individualmente.**
- 2) As strings das mensagens (`msg_np` e `msg_p`) são terminadas com o caractere especial `'\0'`. Esse caractere pode ser usado para identificar o final da mensagem.

#### Materiais de consulta

Para esta atividade, é recomendado consultar o material sobre interface serial assíncrona e os vídeos de apoio fornecidos na página do curso.

#### Instruções para a submissão do trabalho

- 1) Nos comentários do código-fonte, explique o funcionamento geral do programa e justifique as operações e os valores carregados em todos os registradores. É fundamental explicar o mecanismo de varredura adotado para verificar o término da transmissão dos pacotes. Lembre-se que seus comentários fazem parte do relatório. Programas sem comentários terão nota máxima 4.
- 2) Ao final da atividade, salve o código-fonte com nome “seu\_ra.txt” (Exemplo: 025304.txt).
- 3) Faça o upload da sua solução da atividade 5 no Google Classroom.