

# EA871 – Laboratório de Programação Básica de Sistemas Digitais

## Atividade 07

Profs. Levy Boccato, Rafael Ferrari e Tiago Tavares – 2º semestre de 2019

### 1. Objetivos

- Trabalhar simultaneamente com o receptor e o transmissor de uma UART, juntamente com a ideia de interrupção, para receber comandos de um computador e trocar mensagens.

### 2. Resumo da Atividade

O desafio proposto nesta atividade é desenvolver um programa que utilize o receptor da UART para criar um sistema de acionamento de um LED RGB a partir de comandos fornecidos pelo teclado. Além disso, também vamos utilizar o transmissor da UART para encaminhar mensagens de confirmação e *status*.

#### Recepção

A interrupção do tipo “recepção completa” será explorada nesta atividade para que o microcontrolador ATmega328P receba comandos de controle através do teclado do computador. Uma vez que o microcontrolador pode estar realizando operações que consomem tempo enquanto novos comandos são enviados pelo usuário, vamos utilizar a estrutura de um *buffer* circular para armazenar a sequência de comandos recebidos. O processamento dos comandos, então, se dará a partir do primeiro elemento da fila (ou seja, do comando mais antigo armazenado no *buffer*). Por simplicidade, vamos trabalhar com um *buffer* circular relativamente pequeno, de tamanho igual a 10 (dez). A tabela abaixo mostra os possíveis comandos e os respectivos efeitos sobre o LED RGB.

Operação	Comando
Acender – Vermelho	r
Acender – Verde	g
Acender – Azul	b
Acender – Amarelo	y
Acender – Ciano	c
Acender – Magenta	m
Acender – Branco	w

A rotina de serviço associada à recepção deve, portanto, inserir cada novo comando na próxima posição livre do *buffer* circular, caso haja.

Com isso, sempre que houver comandos disponíveis no *buffer* circular, o programa deve processar o comando mais antigo, levando o LED a permanecer aceso na cor especificada durante 200 ms, além de enviar uma mensagem de confirmação pela porta serial. Caso seja detectado um comando inválido, o programa deve encaminhar uma mensagem de erro. A tabela abaixo apresenta as mensagens referentes a cada comando recebido.

Comando	Mensagem
R	"Comando: Acender LED - cor vermelha\n"
G	"Comando: Acender LED - cor verde\n"
B	"Comando: Acender LED - cor azul\n"
Y	"Comando: Acender LED - cor amarela\n"
C	"Comando: Acender LED - cor ciano\n"
M	"Comando: Acender LED - cor magenta\n"
w	"Comando: Acender LED - cor branca\n"
Qualquer outro caractere	"Comando incorreto\n"

## Transmissão

Conforme já descrito, cada vez que um comando for processado, uma mensagem de texto deve ser retornada pela porta serial, segundo a tabela já apresentada. Para enviar cada mensagem, *i.e.*, cada sequência de caracteres, vamos novamente utilizar o mecanismo de interrupção. No caso da transmissão, há duas opções de eventos que disparam interrupções: (1) “transmissão completa” ou (2) “*buffer* de transmissão vazio”.

Além disso, sempre que não houver comandos a processar no *buffer* circular (ou seja, o *buffer* está vazio), a seguinte mensagem deve ser enviada: **"Vazio!\n"**.

### Cuidado:

- A interrupção associada ao evento “*buffer* de transmissão vazio” é disparada continuamente caso o *buffer* esteja limpo. Sendo assim, é preciso desligar esta interrupção ao terminar de transmitir uma sequência completa de caracteres; caso contrário, uma nova interrupção será gerada mesmo se não queremos transmitir outra mensagem.

**Lembrete:** caso as rotinas de serviço de interrupção tenham de acessar ou modificar conteúdos de variáveis do programa, é importante declará-las como variáveis globais com o qualificador **volatile** para assegurar o correto funcionamento.

### Especificações da USART:

- Velocidade de transmissão normal (*i.e.*, modo *double-speed* desativado);
- Modo de transmissão multi-processador desabilitado;
- Número de bits de dados por *frame* igual a 8;
- Modo assíncrono de funcionamento da USART;
- Sem bits de paridade;
- Uso de dois bits de parada;
- *Baud rate* igual a 9.600 bps.

## Convenção de pinos

Por convenção, vamos empregar os pinos 0 a 2 da porta B (ou seja, PB0 a PB2, indicados no diagrama de pinos do Arduino UNO) para acionar os terminais correspondentes às cores vermelha (R), verde (G) e azul (B) do LED RGB, respectivamente. Note que é a mesma montagem explorada na atividade 05.

## Instruções para a submissão do trabalho

- 1) Nos comentários do código-fonte (*main.c*), justifique as operações e os valores carregados em todos os registradores.
- 2) Crie um projeto chamado ‘atividade7’ (letras minúsculas, sem espaço) no Atmel Studio e, ao final da atividade, salve o diretório completo em um arquivo no formato .zip (Aviso: não use .tar.gz nem .rar), com nome ‘seu\_ra.zip’ (Exemplo: 025304.zip).
- 3) Faça o **upload** da sua solução da atividade 7 no Google Classroom.