

Atividade Prática

Comunicação Serial Assíncrona *Bit-Bang*

1 Itens abordados

Para completar esta atividade prática, você precisa conhecer os seguintes tópicos:

- Periféricos:
 - *Ports* de I/O.
 - Interrupções externas.
 - Temporizadores em modo de operação CTC.
- Dispositivos:
 - *Display* de Cristal Líquido (LCD) Alfanumérico.
 - Teclado matricial.

2 Critérios Avaliativos

Esta atividade deve ser postada em um arquivo compactado **em formato ZIP** no SIGAA impreterivelmente até às ____:____:____ do dia ____/____/____. Recomenda-se a postagem da atividade o quanto antes, de modo a evitar problemas associados à indisponibilidade de acesso à Internet.

O arquivo zip deve conter:

- a) O arquivo *workspace* e o diretório *.vscode* do Visual Studio Code.
- b) Todos os códigos-fonte (.c, .cpp, .h, .hpp, .s, .inc, .asm) necessários para a compilação com sucesso do código.
- c) O arquivo de simulação do Proteus, compatível com a versão 8.6 SP2 build 23525.

A nota do atividade prática é baseada no funcionamento e na qualidade da solução apresentada. São critérios indispensáveis:

- Funcionamento da solução;
 - Cumprimento dos critérios estabelecidos.
- Organização do código;
 - Documentação (comentários necessários no código-fonte);
 - Endentação coerente;

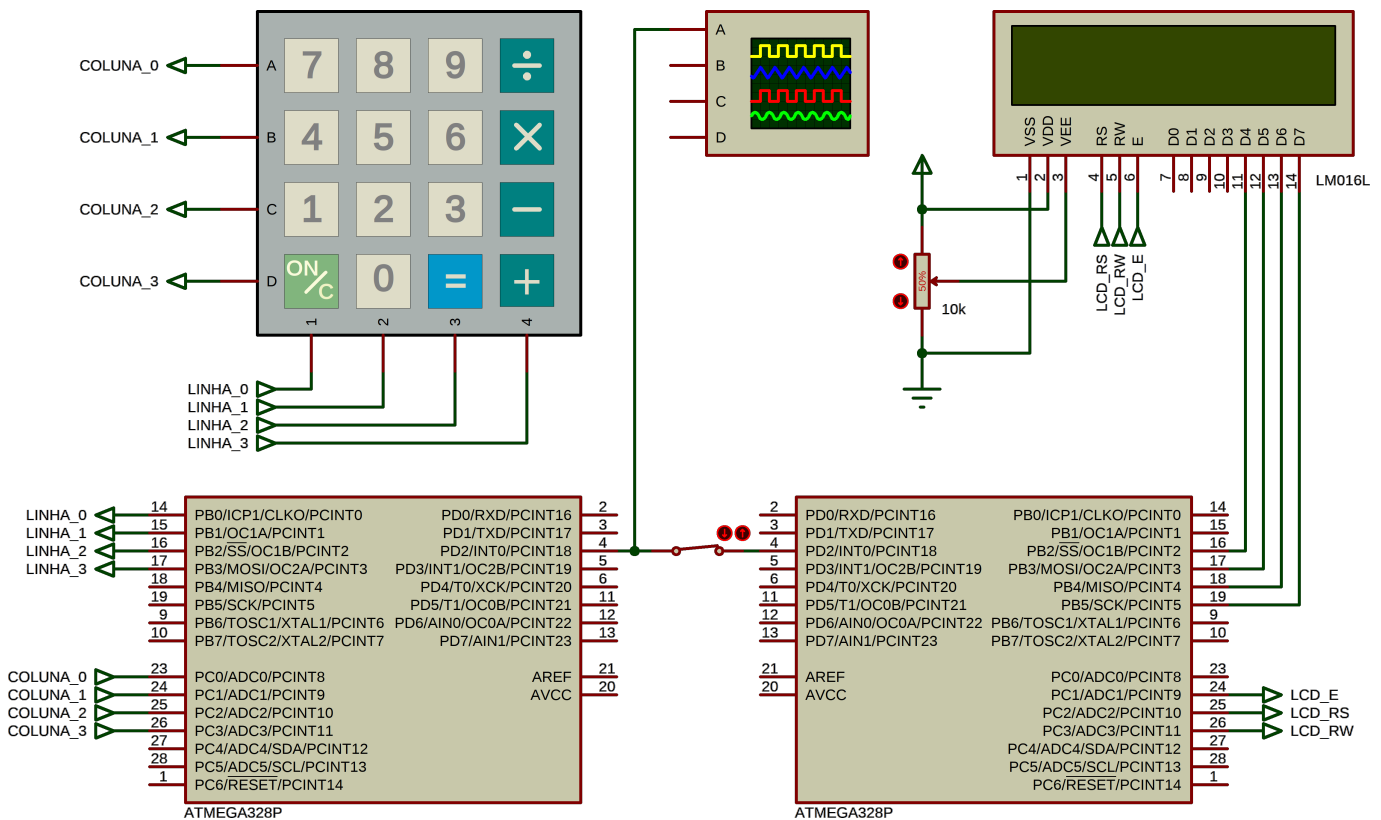
- Formatação adequada.
- Otimização;
 - Uso apropriado dos periféricos;
 - Uso racional de memórias RAM, EEPROM e Flash.

3 Objetivo

Desenvolver um sistema de comunicação serial assíncrona *bit-bang* entre dois microcontroladores.

4 Hardware

A seguir, é apresentado o diagrama esquemático do arquivo de simulação e a lista de componentes e suas configurações:



- Microcontrolador;
 - Device: ATMEGA328P;
 - Library: AVR2;
 - Part Reference: *oculto*;
 - Part Value: *oculto*;
 - Encapsulamento: SPDIL28;
 - Fuse CLKDIV8: (1) Unprogrammed;
 - Fuse CLKSEL: (1111) External crystal 8.0–MHz;
 - Clock Frequency: 16 MHz;

- LCD;
 - Device: LM016L;
 - Library: DISPLAY;
 - Part Reference: *oculto*;
 - Part Value: *oculto*;
 - Encapsulamento: CONN-DIL14;
 - Clock frequency: 250kHz;
 - Row 1: 80-8F;
 - Row 2: C0-CF;
- Potenciômetro;
 - Device: POT-HG;
 - Library: ACTIVE;
 - Part Reference: *oculto*;
 - Resistance: 10k;
 - Law type: LINEAR;
 - Minimum Resistance: 0.1;
 - Switching time: 1m;
- Teclado matricial;
 - Device: KEYPAD-SMALLCALC
 - Library: ACTIVE;
 - Part Reference: *em branco*;
 - Part Value: *em branco*;
- Chave;
 - Device: SWITCH;
 - Library: ACTIVE;
 - Part Reference: *em branco*;
 - OFF Resistance: 100M;
 - On Resistance: 0.01;
 - Switching Time: 1m;

5 Requisitos Obrigatórios

- 1) O código deve ser programado para o microcontrolador ATmega328P.
- 2) O código deve ser programado em C/C++.
- 3) Um dos microcontroladores será o **transmissor** e o outro será o **receptor**.
- 4) O **transmissor** está conectado ao teclado matricial.
- 5) O **receptor** está conectado ao LCD.
- 6) A transmissão deve ser assíncrona, ou seja, nenhuma linha de *clock* ou sincronismo deve ser utilizada.
- 7) Deve haver apenas uma linha (um canal) conectando **transmissor** e **receptor**.
- 8) Quando o canal não estiver sendo utilizado (ocioso), ele deve estar em nível lógico alto.
- 9) Os códigos e projetos de ambos os microcontroladores devem ser entregues e a simulação deve conter ambos os microcontroladores.
- 10) A comunicação deve ser realizada em modo *bit-bang*, ou seja, não é permitido o uso do periférico UART.
- 11) O pacote a ser transmitido é formado por 10 bits: um bit inicializador, oito bits de dados e um bit finalizador.
- 12) Cada bit do pacote possui duração de 104µs e deve ser gerado através de um temporizador.
- 13) **Transmissor:**
 - (a) O microcontrolador do **transmissor** deve fazer a leitura contínua do teclado matricial.
 - (b) Quando uma tecla for pressionada, seu valor (8 bits) deve ser enviado para o **receptor** dentro de um pacote.
 - (c) A transmissão inicia com o o transmissor enviando o bit inicializador do pacote, forçando o canal para

0 e configurando um temporizador para disparo em 104μs, para sinalizar o início de uma transmissão (**START BIT**).

- (d) Após o estouro do temporizador, o transmissor deve iniciar o envio dos dados, colocando cada um dos bits do valor no canal, iniciando pelo bit menos significativo (LSB), e passando ao bit seguinte a cada 10μs (**DATA BITS**).
- (e) Após o envio de todos os 8 bits do valor, o canal deve ser forçado para 1, e permanecer neste nível por pelo menos 104μs para marcar o final da transmissão (**STOP BIT**).
- (f) Ao final deste processo, o temporizador deve ser desativado.

14) **Receptor:**

- (a) O microcontrolador do **receptor** deve receber o pacote do **transmissor** e processá-lo.
- (b) Se for recebida a tecla **ON/C**, o *display* deve ser limpo.
- (c) Se for recebida qualquer outra tecla, o símbolo correspondente (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, -, *, +, =) deve ser escrito no *display*.
- (d) Se o canal estiver ocioso, o receptor deve monitorar a interrupção externa em busca do sinal de início de transmissão.
- (e) Quando for detectado o início da transmissão (borda de descida no canal), o receptor deve configurar um temporizador para disparo em 52μs para receber o bit de início da transmissão (**START BIT**).
- (f) Após o fim dos 52μs, ou seja, aproximadamente no meio do recebimento do **START BIT**, o temporizador deve ser reconfigurado para disparo em 104μs para recepção dos bits seguintes.
- (g) O receptor deve adquirir cada um dos oito **DATA BITS**, lendo o nível do canal a cada estouro de 104μs, até que todos os 8 bits tenham sido recebidos.
- (h) O receptor deve, então, fazer uma nova leitura do valor do canal no próximo disparo em 104μs para receber o sinal de término da transmissão (**STOP BIT**).
- (i) Ao final deste processo, o receptor deve voltar a monitorar a interrupção externa em busca do próximo sinal de início de transmissão.