

Instituto Federal do Rio Grande do Sul – Campus Canoas

Relatório de Sistemas Microcontrolados
(Atividade Prática 2)

Aluno: João Pedro Tassoni

Turma: 3º Ano de Eletrônica

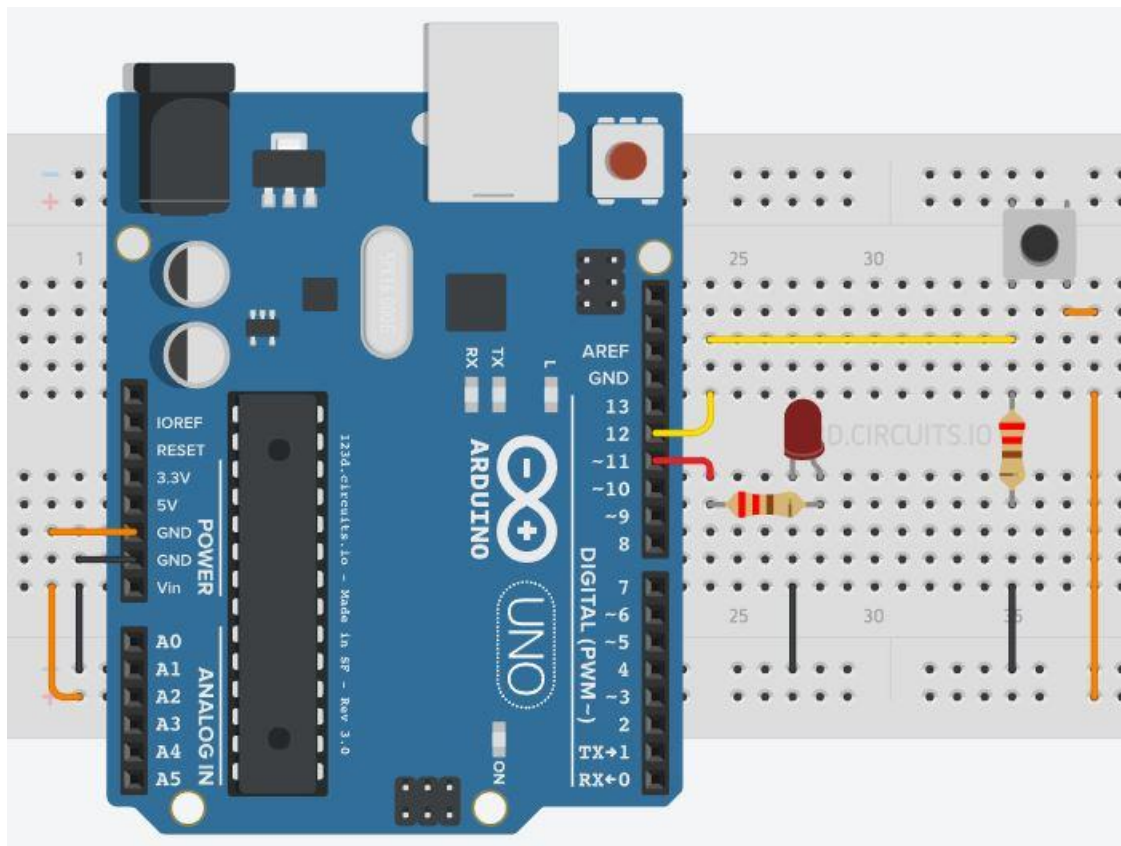
Descrição do Trabalho:

O aluno deveria montar um circuito capaz de converter uma string em um padrão de acionamento seguindo o código Morse, com um LED vermelho e um push-button, agindo da seguinte forma:

- A função deve ser capaz de traduzir uma string qualquer na sequência de acionamento para código Morse, seguindo o protótipo abaixo.
- void toMorse(char msg[], int msgsize).
- Ao pressionar o botão, o código na função loop() deve iniciar a conversão da mensagem SOS (usando a função do primeiro passo).
- A duração do PONTO deve ser de 0,5 segundo e a do TRAÇO deve ser de 1,5 segundo. O tempo entre PONTO e TRAÇO deve ser de meio segundo. O tempo entre letras deve ser de 0,5 segundo. O tempo entre palavras deve ser de 2 segundos.

Não há nenhuma restrição ao aluno além destas condições. É permitido o uso de loops FOR e delay.

Montagem:



Essa é uma reprodução da montagem que foi utilizada em aula para a execução do programa, com resistores de 220 ohm no LED e no botão, o LED vermelho, o push-button ligado em pull-down e a placa Arduino UNO.

A saída do LED (porta 11) passa por um resistor e vai até o LED, que está aterrado.

A saída de 5V da placa passa por uma trilha na protoboard e vai até o botão (pelo fio laranja), que na outra perna tem conectado a porta de leitura (porta 12) e um aterramento.

OBS: os fios pretos são de aterramento (GND), o laranja de energia da placa, o amarelo de leitura do botão e o vermelho é a ligação com o LED.

Código e Lógica:

O código foi escrito com base em alguns exemplos mostrados em aula e pesquisa feita em fóruns na internet, e será brevemente explicado nesse tópico pois o arquivo .ino está comentado para facilitar o entendimento do programa.

Para começar, todas as variáveis foram declaradas e uma struct foi usada para converter as letras do alfabeto, números e caracteres de maior importância (como o espaço e pontuações por exemplo) em código Morse.

Dentro da função setup, foram declarados o modo de operação de todos os pinos utilizados (pinMode) e ocorre a primeira leitura do botão, atribuída a variável oldstt (status antigo).

No começo da função loop, ocorre a leitura do novo valor de status do botão (atribuído à stt), e a sua comparação com o antigo para dizer se o botão foi pressionado. Se isso acontecer, a função toMorse é executada.

Dentro da função toMorse, a primeira coisa que acontece é a declaração de duas variáveis que vão servir de contador e de uma string que vai abrigar toda a frase convertida em código Morse. Depois disso, dois loops FOR são colocados em sequência com o âmbito de comparar cada letra da string “frase” com as letras da struct (função toupper converte letras em minúsculo para maiúsculo para o programa sempre funcionar). Após essa comparação, a letra convertida em Morse é adicionada à variável morsePhrase, com um espaço entre as letras e quatro entre as palavras (para fechar os 2 segundos de delay quando houver um espaço entre palavras).

Depois que os loops FOR acabam e toda a string foi convertida em código Morse, um novo loop FOR acontece reaproveitando a variável “cont1”, e serve para fazer o LED piscar de acordo com os pontos e traços (como está descrito nos critérios para a realização da atividade). Nessa parte foi feito o uso de delays, pois o código com uma tentativa de implementação da função millis() passou por muitos problemas de funcionamento. Nesse ponto a função toMorse acaba e o programa retorna para a função void loop(), que iguala a variável oldstt com stt, (para perceber quando o botão for pressionado novamente), e o programa acaba.