

Instituto Federal do Rio Grande do Sul – Campus Canoas

Relatório De Sistemas Microcontrolados I  
(Atividade Prática 3)

Autoria: João Pedro Tassoni

Turma: Eletrônica 3

### Descrição do trabalho:

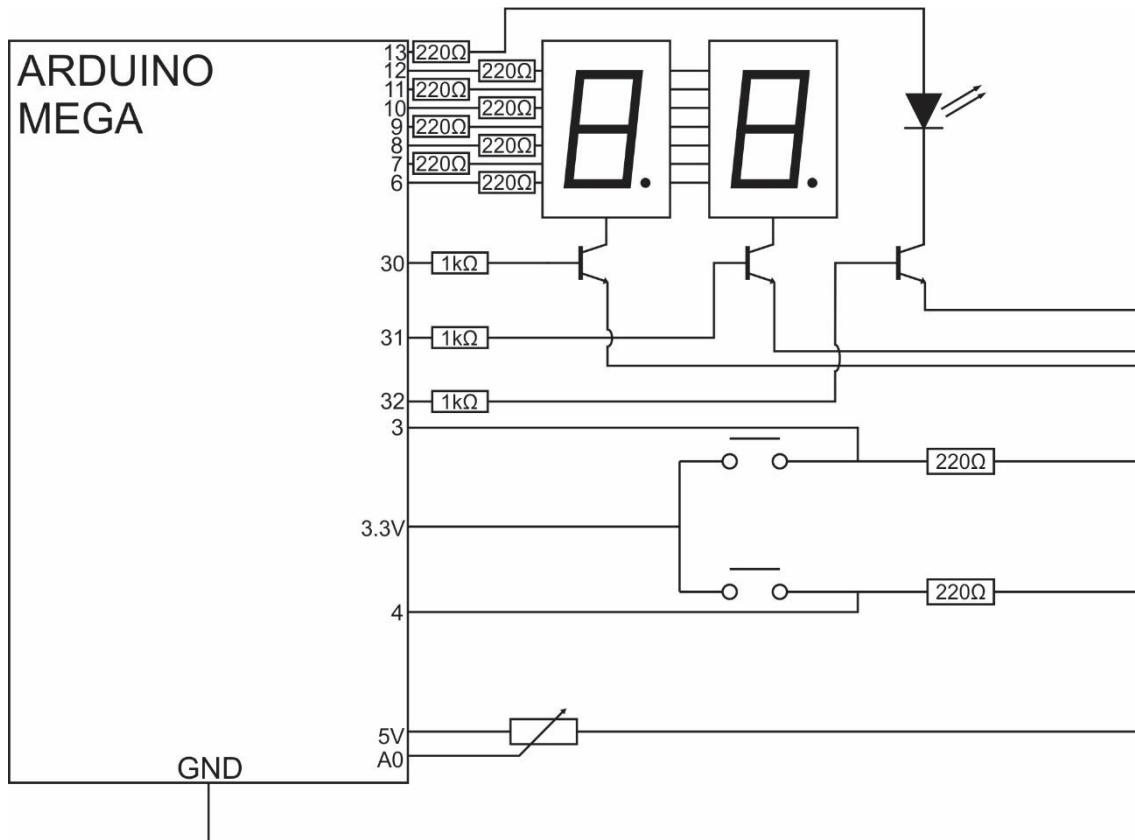
O aluno foi solicitado a criar um código que operasse em duas etapas: a determinação do tempo, que poderia ser feita por meio do comando serial PARAR e pela leitura de um potenciômetro, e a segunda parte seria uma contagem de 00 até o número determinado, contagem essa que começaria após o pressionamento de um botão, e recomençaria sempre que o mesmo fosse pressionado novamente.

O código deveria operar de forma que quando um segundo botão fosse pressionado. O valor do tempo determinado para parar a contagem pudesse ser escolhido novamente, e a contagem fosse exibida em dois displays que compartilhariam os pinos da placa Arduino, e para funcionar, tivessem seu aterramento chaveado com transistores.

A última exigência é simplesmente que ao final da contagem, um LED (também chaveado com transistor) fosse aceso, para sinalizar o fim.

Esquemático:

Essa é uma reprodução da montagem utilizada em aula, representada por um esquema.



Os resistores utilizados são comerciais, no valor mais próximo ao necessário para exercer as funções. O potenciômetro não precisa ter um valor específico. Os transistores são do modelo BC338 (NPN). O LED é um comum, muito utilizado em diversas atividades de aula.

Os pinos da placa foram definidos de forma a organizar melhor a montagem na protoboard.

### Lógica:

O programa ficou relativamente grande, comparado a outras atividades realizadas anteriormente em aula.

Para facilitar a realização da tarefa e a possibilidade de expansão futura do código, a parte relacionada a comando via serial utilizou um sketch já pronto, disponibilizado pelo professor.

Para começar, é feita a declaração de todas as variáveis globais, de forma organizada e comentando ao lado as suas funções.

Logo após, todas as variáveis e funções relacionadas a leitura da serial se encontram.

### **Função parar**

O parâmetro que acompanha o comando é passado e convertido para uma string Arduino, e logo após, convertido para um inteiro usando uma função de manipulação de strings (to.Int). Seguidamente, uma comparação é feita para saber se esse valor está entre 00 e 99, e se o botão de começo não foi pressionado ainda (ou seja, a leitura do valor de parada ainda pode ser feita), ele cria uma string com uma mensagem para exibir na interface serial e a variável numeroAtual recebe o valor de pin (parâmetro convertido para inteiro). Se as condições citadas acima não forem verdadeiras, uma mensagem avisando que o tempo excedeu o limite aparece na interface.

### **Função cmdProcess**

Basicamente falando (não entrarei em detalhes pois as funções relacionadas a serial já foram explicadas em aula), essa função separa o comando (parar) do parâmetro (número que será usado para parar o contador) achando o separador (=) e guardando a sua posição. Logo depois disso o programa pega os números entre a posição do separador e o fim da string e guarda em uma nova string que é passada para a função parar (explicada acima).

### **Função serialEvent**

Também falando brevemente pelo mesmo motivo citado na função anterior, essa função vai testar para saber se tem bits disponíveis no buffer da serial, e se tiverem, lê todos até o “terminador” (!) e armazena numa string passada para a função cmdProcess.

## **Função setup**

Aqui são feitas as declarações de modo de todos os pinos utilizados no programa, como os dos botões por exemplo. Além disso, é feita a primeira leitura dos botões, do potenciômetro e dos contadores de tempo, e o pino ligado ao LED é definido como uma saída constante de 5V (pois o mesmo está com uma chave transistorizada).

## **Função loop**

No começo, é feita uma leitura do potenciômetro, que só vai ser mandada para o display se variar em pelo menos 1 ponto. A leitura dos botões é feita em seguida. Os dois estão ligados em pull-down. O primeiro, quando pressionado, manda o valor que está sendo exibido no display (e foi obtido por serial ou potenciômetro) para uma variável que guarda o valor de parada do contador, além de zerar a variável numeroAtual, pois a mesma vai ser utilizada para contar de 0 até numeroFinal, e a variável com (usada para controlar grande parte do programa) recebe o valor 1. Quando o segundo botão é pressionado, basicamente o valor recebido pela serial é zerado junto com a variável com, e o chaveamento do LED é desligado.

A seguir aparece o primeiro switch, que conta com 5 casos. O primeiro serve para ligar o display correspondente a dezena, ativando o transistor 1, e passando o valor da dezena da variável numeroAtual para o número a ser exibido no display. O segundo mantém esse mesmo transistor ativado e desliga todos os pinos do display. O terceiro repete o processo do primeiro, mas faz isso no display da unidade, e o número a ser exibido é o correspondente a unidade de numeroAtual. O quarto desliga o display das unidades inteiro. O quinto somente reseta a máquina de estados. Para controlar a passagem de tempo dessa máquina, um if está mais para o final do programa, incrementando a variável controle a cada 5 milissegundos.

A próxima etapa é um próximo switch, controlado pela variável numeroExibido, e ela basicamente exhibe o número que existe dentro dessa variável (e é passado a mesma pela máquina de estados explicada anteriormente). Um caso 10 existe somente para o display ser totalmente limpo antes da exibição de informação no próximo.

As últimas partes são os controles de tempo. Um if conta 5 milissegundos para incrementar a primeira máquina de estados, e outro if conta um segundo para incrementar a variável numeroAtual enquanto o cronômetro não chegar no valor de parada. Além desses if, existe um para controlar o chaveamento do LED de parada, que tem como condições o número de parada ser igual ao número atual do contador, e o botão 1 ter sido pressionado.

## Considerações Finais:

Esse trabalho não parecia ser tão difícil ao início, mas o seu maior desafio foi fazer tudo funcionar ao mesmo tempo. Os botões podem ser lidos sempre desde que a placa seja ligada, o valor do potenciômetro ou o da serial devem ser selecionados com base no último que foi escolhido, etc. Muito bom para prática e desenvolvimento de habilidades relacionadas a programas mais complexos.