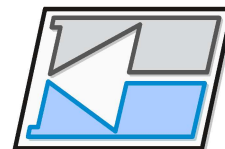




INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

UNIDADE CURRICULAR: INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO
PROFESSOR(A): CARLOS GONTARSKI SPERANZA

ALUNO: LEONARDO HENRIQUE DILL BRUXEL



Medidor VU implementado em Arduino

Projeto Final de LGP

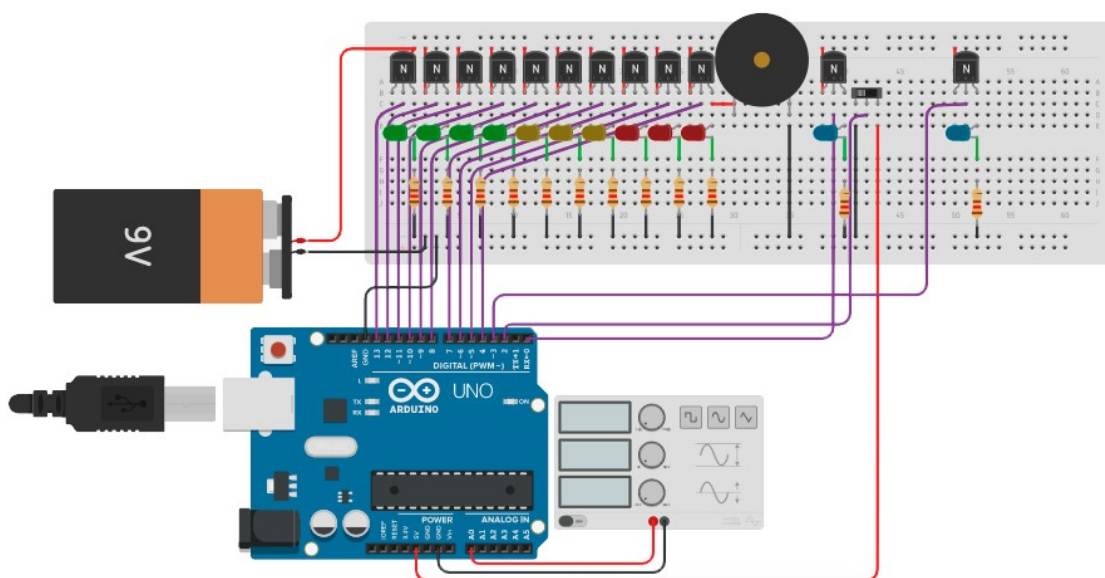


Figura 1: Diagrama do circuito, feito usando o software Autodesk® Tinkercad®.

1. Introdução.

Um VU (Volume Unit) é um instrumento de medição muito utilizado em estúdios musicais para diversas aplicações, normalmente para evitar clipping (“estouro” do áudio) ou mesmo evitar que o volume de um instrumento esteja muito alto na mixagem, etc. A implementação de um VU normalmente é feita usando CIs específicos como o LM3914¹ ou construídos usando componentes analógicos que por sua vez processam o áudio e controlam os LEDs. A intenção desse projeto é de facilitar a construção de um VU, usando um Arduino², componentes acessíveis e fácil implementação, para que um usuário leigo possa construir o seu próprio equipamento sem complicações. Inúmeras limitações são impostas à usuários leigos de sistemas de áudio, pois o funcionamento eletrônico dos mesmos muitas vezes é por demais complexo, e portanto, inacessível.

O Arduino®, por sua vez, é um componente extremamente acessível, pois seu funcionamento pode ser explicado muito facilmente, de modo que leigos possam entender o comportamento de seus equipamentos e até mesmo se inclinarem a realizar novos projetos. A interface open-source traz uma infinidade de aproximações pela base de usuários que constrói coletivamente o conhecimento comunitário. O Arduino® ainda tem um fator amplificador da sua facilidade de uso, a linguagem de programação. As linguagens utilizadas em um Arduino® são C e C++, conhecidas principalmente pela fácil aproximação por iniciantes, por serem linguagens intermediárias entre baixo e alto nível, e por terem comandos extremamente intuitivos.



Figura 2: Arduino® Uno³.

-
1. Circuito Integrado desenhado em 1980 pela empresa estadunidense National Semiconductors. Tem a função de permitir a visualização da magnitude de sinais analógicos.
 2. Arduino é uma plataforma open-source para a realização de protótipos eletrônicos básicos, como o VU deste projeto ou até mesmo voltímetros e conversores ADC e DAC.
 3. O modelo mais básico desenvolvido pela Arduino® Corporation.

2. Desenvolvimento do projeto.

Inúmeros problemas foram encontrados ao desenvolver o módulo VU. Diversas estruturas falharam por conta de erros de desenvolvimento. Para exemplificar, LEDs que não ligavam, não desligavam, não se comportavam como esperado, etc. Para atender as especificações pedidas pelo professor, inúmeras vezes o projeto foi editado, mas não houve controle de versão por conta das limitações do software utilizado¹.

O primeiro obstáculo encontrado foi o desenvolvimento do código, pois apesar da programação de um Arduino® ser semelhante à programação em C e C++, as pequenas diferenças são suficientes para que um usuário acostumado com certas bibliotecas em C cometa pequenos erros. Os principais erros cometidos foram causados majoritariamente por falta de atenção. Inúmeros “crashes” e “bugs” foram detectados e corrigidos, porém ainda podem haver erros básicos e certamente imperceptíveis.

O primeiro problema encontrado foi com a interpretação dos dados recebidos na porta analógica A0 do Arduino®. Por uma má interpretação, usou-se como valor máximo 5, que seria na verdade apenas a tensão máxima, que após a conversão ADC do Arduino® seria interpretada como um valor numérico entre 0 e 1024, devido aos 32 bits do conversor da plataforma. Esse erro ocasionou o não funcionamento do circuito, uma vez que o valor 5, para a plataforma, era menor até mesmo do que a média dos valores recebidos, logo, todos os LEDs indicadores ligavam, inutilizando o circuito. O erro foi corrigido com a reinterpretação dos dados e construção do circuito usando 1024 como valor máximo, como devido. Outro problema encontrado foi que os LEDs não ligavam, pois a potência fornecida pelo Arduino® não é capaz de alimentar os 12 diodos do circuito. A solução foi o uso de uma bateria e transístores com a base conectada às saídas digitais da plataforma. Assim, ao receber a informação, o transístor ligava ou desligava, interrompendo o fluxo de corrente para os LEDs, que consequentemente tinham de ser alimentados por uma fonte externa qualquer. Uma pilha de 9 V foi utilizada pela portabilidade e facilidade de ser encontrada.

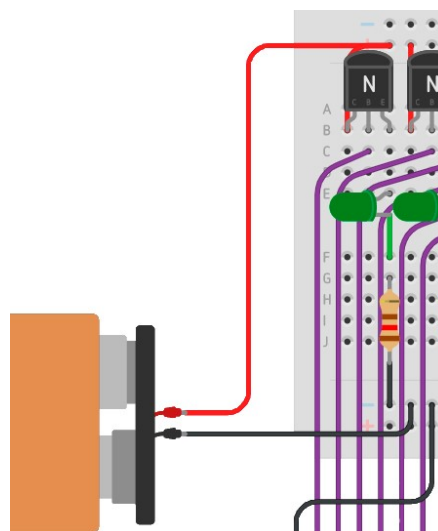


Figura 3: Esquema da alimentação dos LEDs.

Código de cores: Fio vermelho (9V); Fio preto (GND); Fio roxo (Sinal Digital).

1. O software Autodesk® TinkerCad® não possui uma função de salvar versões ou até mesmo salvar diretamente o arquivo no computador do usuário. Inviabilizando assim um controle estável e simples de versões.

3.1. Funcionamento e programação.

O circuito utiliza estruturas de repetição `while` e `for`, em conjunto com o `void loop`, para realizar diversas funções. A função `loop` por exemplo, faz o reconhecimento do modo de funcionamento do circuito por meio de uma estrutura de decisão `if`. Ao mudar a posição da chave, a função é automaticamente alterada. A função `todos` faz com que todos os LEDs ultrapassados pelo inteiro `AnalogTensao`¹ liguem, assim como em um VU comum. Por outro lado, a função `mais alto` liga apenas o LED de maior valor ultrapassado pelo inteiro `AnalogTensao`.

A identificação do modo de operação é feita pelos LEDs indicadores do VU, uma vez que ambos os modos são visivelmente distinguíveis, e por um último LED que opera diferentemente entre os dois modos. No modo de operação `todos` o LED D11 fica ligado, enquanto no modo `mais alto`, D11 entra em um loop de liga-desliga, feito usando um loop `for`. Infelizmente, a plataforma utilizada para a elaboração do circuito não permite a demonstração deste efeito, uma vez que os LEDs não desligam imediatamente na simulação, pois passam por um processo semelhante a uma descarga de capacitor, que dificulta a visão do efeito.

O circuito consta também com um alto-falante Piezo², ligado ao LED de maior valor, o que transforma o Piezo em um indicador sonoro de que haverá clipping ou outros problemas decorrentes de tensões muito altas em circuitos de áudio. Há também um último LED que serve para indicar que o circuito está ligado.

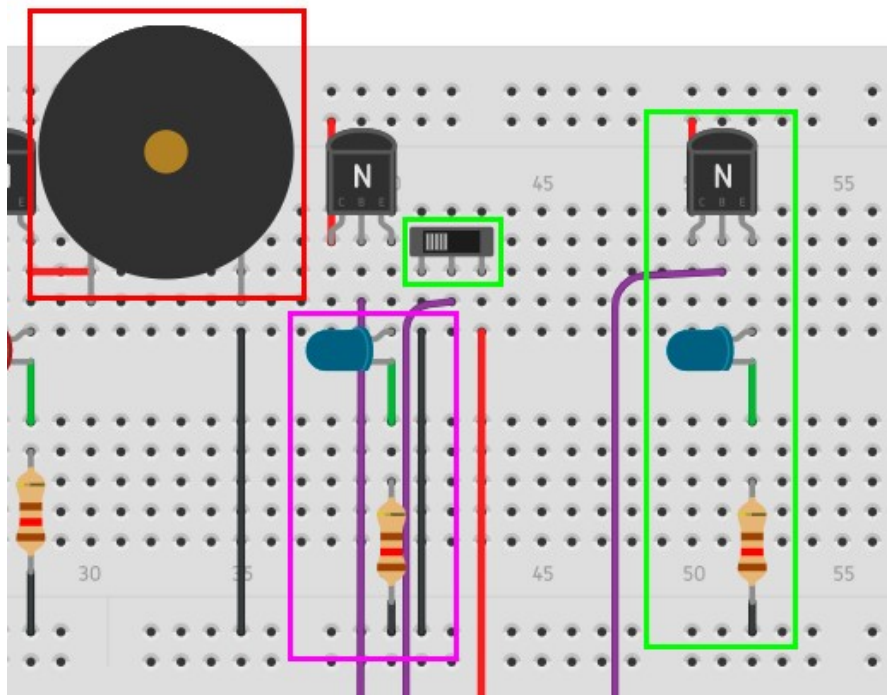


Figura 4: Indicadores de funcionamento.

Legenda: Verde (Modo de Operação); Rosa (Indicador de Funcionamento); Vermelho (Piezo)

-
1. Inteiro com valor digital entre 0 e 1024, que representa o intervalo de tensão entre 0 e 5 V aplicado em A0.
 2. Componente que utiliza o efeito piezoelétrico para produzir vibrações sonoras. Possui um alcance de até 100 kHz e podem ser implementados diretamente em saídas TTL.

3.2. Simulação

A parte da simulação foi a etapa mais complicada do projeto. O software TinkerCad® é muito limitado e muitos dos componentes tem funcionamento diferente da expectativa de um componente real. LEDs, por exemplo, tem um efeito de cooldown para desligar, como se estivesse sendo alimentado por um capacitor. Os transístores e os LEDs disponíveis no software não possuem especificações ou modelo para o cálculo e dimensionamento dos resistores, a placa Arduino® não faz a alimentação correta nos pinos digitais, por conta da potência fornecida, etc...

Uma das possíveis soluções para os problemas é utilizar o software Proteus®, produzido pela Labcenter Electronics®. O software, no entanto, não possui suporte nativo ao componente Arduino®, sendo necessária a instalação de uma biblioteca exterior. Duas bibliotecas compatíveis foram encontradas, a biblioteca Simulino® e ArduinoTEP®. No entanto, problemas na instalação inviabilizaram a simulação do projeto. Esses problemas foram resolvidos posteriormente, e a biblioteca Simulino foi escolhida para a realização do projeto, pela facilidade do uso e tamanho compacto do componente. Novos problemas, no entanto, foram encontrados, pois a implementação do código-fonte na simulação usando o Simulino® é um processo difícil, complexo e sem documentação.

A simulação em SPICE® (o modelo de simulação usado pelo software Proteus®) se mostrou inviável, então a única maneira de simular o circuito voltou a ser o software livre TinkerCad®.

4. Conclusão.

Com este projeto, aprendeu-se na prática a dificuldade para a implementação de um VU e o alto preço e complexidade de equipamentos já existentes. O equipamento, por si, apesar de básico, possui componentes-chave que fazem com que o circuito não seja indicado para estudos amadores ou iniciantes. Um VU implementado em Arduino® possui inúmeros detalhes quase que imperceptíveis que fazem essa implementação imprática, sendo melhor projetado e desenhado usando um CI como o já citado LM3914. No entanto, muito foi aprendido sobre programação e projetos. No geral, foi uma ótima experiência.

Agradecimentos

Esta página, dedico aos colaboradores do projeto. São eles:

Geovane Martins, que me ajudou com as linhas complexas de código e detecção de erros e bugs.

Professor Carlos Speranza, que ministrou com sucesso as aulas da disciplina e instruiu-me com o conhecimento necessário para a realização do projeto.

GitHub®, ferramenta necessária para a busca por equipamentos semelhantes de código aberto, que ajudaram-me a desenvolver o projeto.

Legião Urbana, grupo musical que deu-me o suporte emocional necessário para lidar com a pressão dos projetos finais.

Novos Baianos, pela mesma razão de Legião Urbana

Kanye West, pelo mesmo motivo.

Demônios da Garoa, pelo mesmo motivo.

Laura Hobold, minha namorada, pelo suporte emocional.

Valter Rogério, por ser o Valter.

Cervejas Original e Brahma, que me apoiaram nas horas tristes e vagas enquanto uma nova solução para o projeto não surgia.

Licenças

Este dispositivo, misto de software e hardware, é de livre uso e comercialização, esta última desde que o autor (Leonardo Henrique Dill Bruxel) seja creditado.

A versão final do projeto pode ser encontrada em <https://www.tinkercad.com/things/eqnbQlz07uF-indicador-de-volume>