

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RODRIGO JOHANN

**J-WAH: Projeto e Implementação de um
Auto-Wah em Plataforma Teensy 3.6**

Porto Alegre

2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RODRIGO JOHANN

J-WAH: Projeto e Implementação de um Auto-Wah em Plataforma Teensy 3.6

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Adalberto Schuck Jr.

Porto Alegre

2018

Resumo

O projeto tem como objetivo implementar uma versão digital do pedal de efeito conhecido como Mu-tron III. É realizado inicialmente um estudo do circuito do Mu-tron III, sendo este dividido em três blocos, chamados de blocos de Pré-processamento, Detector de Envoltória e Filtro de Variáveis de Estado. A placa de desenvolvimento Teensy 3.6 é utilizada para reproduzir de forma digital as etapas de processamento do bloco Detector de Envoltória e do bloco do Filtro de Variáveis de Estado. É elaborado um circuito que possibilite a leitura e escrita do sinal de áudio pela placa, onde é reproduzido o bloco de Pré-processamento do circuito original.

Os algoritmos de processamento são elaborados inicialmente no software MATLAB® R2012b, e após são implementados no Teensy 3.6. É desenvolvido um software para a escrita dos algoritmos de processamento na placa de desenvolvimento, onde são determinadas as configurações de taxa de amostragem e resolução de leitura do sinal.

São realizadas medições nas etapas de hardware e software do projeto, que são comparadas com medições realizadas em simulações do circuito original. São medidas as saídas dos blocos de Pré-processamento do circuito original e do circuito implementado para uma entrada senoidal, para os valores mínimo e máximo de ganho. O bloco Detector de Envoltória do circuito original e do projeto implementado têm sua resposta medida para uma entrada senoidal para seus dois modos de operação. O bloco do Filtro de Variáveis de Estado, tanto do circuito original quanto do projeto implementado, tem sua resposta em frequência medida nas suas três saídas para valores mínimo e máximo do fator Q .

Ao realizar a leitura das medições realizadas, concluiu-se que o método proposto possibilitou uma reprodução satisfatória do efeito para valores baixos de ganho na etapa de Pré-processamento. A etapa de Pré-processamento proposta no projeto não apresentou êxito pleno na reprodução da etapa do circuito original devido à sua faixa limitada de valores de ganho de operação, constituindo-se na principal limitação do projeto implementado.

Palavras-chave: processamento digital de sinais, efeitos, áudio, auto-wah

Abstract

The project has as its main objective to build a digital version of the effect pedal known as the Mu-tron III. Initially a study of the Mu-tron III is realized, with it being divided in three blocks, known as the Preprocessing block, the Envelope Detector block and the State Variable Filter block. The Teensy 3.6 developing board is used to reproduce digitally the processing steps of the Envelope Detector and the State Variable Filter blocks. A circuit that makes the reading and writing of the audio signal from the board is built, and there the Preprocessing block of the original circuit is reproduced.

The processing algorithms are made initially in the MATLAB® R2012b software, and later are implemented in the Teensy 3.6. A software for the reading/-writing process in the developing board is made, where configurations like sampling rate and reading resolution are determined.

Measurements are made in the project's hardware and software steps, and these measurements are compared with measurements made in simulations of the original circuit. The outputs of the Preprocessing block of the original circuit and of the implemented circuit are measured for a sinusoidal input, for minimum and maximum gain values. The Envelope Detector block from the original circuit and from the implemented circuit have their responses measured for a sinusoidal input for both of their operating modes. The State Variable Filter block, from the original circuit and from the implemented circuit, have its frequency responses measured in the three filter modes and for minimum and maximum Q factor values.

Analysing the measurements made, it's concluded that the proposed method made possible to reproduce successfully the original pedal for low Preprocessing gain values. The Preprocessing step that is proposed in the project have not been fully reproduced the original block due to it's limitations in the operational gain, being the main limitation in the implemented project.

Keywords: digital signal processing, effects, audio, auto-wah

Lista de ilustrações

Figura 1 – Esquemático do Mu-tron III	14
Figura 2 – Diagrama de Blocos do Mu-tron III	15
Figura 3 – Bloco de entrada do Mu-tron III	17
Figura 4 – Detector de Envoltória do Mu-tron III	18
Figura 5 – Detector de Envoltória do Mu-tron III	19
Figura 6 – Sinais de entrada e saída do detector de envelope	20
Figura 7 – Filtro de Variáveis de Estado	21
Figura 8 – Filtro de Variáveis de Estado do Mu-tron III	22
Figura 9 – Filtro de Variáveis de Estado Digital	23
Figura 10 – Diagrama de Blocos do Auto-Wah Digital	25
Figura 11 – Teensy 3.6	26
Figura 12 – Estrutura de um compilador cruzado	27
Figura 13 – Bloco de Pré-Processamento do Projeto	29
Figura 14 – Bloco de Entrada do Projeto	29
Figura 15 – Filtros Anti-Aliasing de Entrada	30
Figura 16 – Parâmetros de Regulagem do Sistema	31
Figura 17 – Diagrama de Blocos do Detector de Envelope	32
Figura 18 – Resposta a) passa-alta, b) passa-banda e c) passa-baixa do filtro para diferentes valores de frequência de corte (parâmetro <i>Range</i> na posição <i>Low</i>)	33
Figura 19 – Resposta a) passa-alta, b) passa-banda e c) passa-baixa do filtro para diferentes valores de frequência de corte (parâmetro <i>Range</i> na posição <i>High</i>)	34
Figura 20 – Diagrama de blocos do Filtro de Variáveis de Estado	34
Figura 21 – Circuito de saída da placa	36
Figura 22 – Bloco de Entrada do Mu-tron III em simulação no Multisim	37
Figura 23 – Bloco de Entrada do circuito implementado em simulação no Multisim	37
Figura 24 – Detector de envoltória do Mu-tron III em simulação no Multisim	38
Figura 25 – Filtro de Variáveis de Estado do Mu-tron III em simulação no Multisim	38
Figura 26 – Sinal de entrada (em magenta) e sinal de saída (em turquesa)	40
Figura 27 – Sinal de entrada (em magenta) e sinal de saída (em amarelo)	41
Figura 28 – Resposta em frequência do bloco de entrada do <i>Mu-tron III</i>	42
Figura 29 – Sinal de entrada (em vermelho) e sinal de saída (em azul)	42
Figura 30 – Sinal de entrada (em vermelho) e sinal de saída (em azul)	43
Figura 31 – Resposta em frequência do bloco de entrada do circuito implementado	43

Figura 32	– Sinal de entrada do circuito implementado	44
Figura 33	– Valor de saída do bloco de entrada com alto ganho	45
Figura 34	– Valor de saída do bloco de entrada ganho mínimo	45
Figura 35	– Saída do detector de envoltória para <i>Drive</i> na posição <i>Up</i>	46
Figura 36	– Saída do detector de envoltória para <i>Drive</i> na posição <i>Down</i>	46
Figura 37	– Saída do detector de envoltória para <i>Drive</i> na posição <i>Up</i>	47
Figura 38	– Saída do detector de envoltória para <i>Drive</i> na posição <i>Down</i>	48
Figura 39	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Mínimo	48
Figura 40	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Máximo	49
Figura 41	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Mínimo	49
Figura 42	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Máximo	50
Figura 43	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Mínimo	50
Figura 44	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Máximo	51
Figura 45	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Mínimo	51
Figura 46	– Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e <i>Peak</i> Máximo	52
Figura 47	– Resposta em frequência para os filtros passa-alta (em laranja), passa-banda (em vermelho) e passa-baixa (em azul) e <i>Peak</i> Mínimo	52
Figura 48	– Resposta em frequência para os filtros passa-alta (em laranja), passa-banda (em vermelho) e passa-baixa (em azul) e <i>Peak</i> Máximo	53
Figura 49	– Ruído branco utilizado nas medições	54
Figura 50	– Saída passa-alta do filtro para <i>Peak</i> mínimo	54
Figura 51	– Saída passa-alta do filtro para <i>Peak</i> máximo	55
Figura 52	– Saída passa-banda do filtro para <i>Peak</i> mínimo	55
Figura 53	– Saída passa-banda do filtro para <i>Peak</i> máximo	56
Figura 54	– Saída passa-baixa do filtro para <i>Peak</i> mínimo	56
Figura 55	– Saída passa-baixa do filtro para <i>Peak</i> máximo	57
Figura 56	– Saída passa-alta do filtro para <i>Peak</i> mínimo	58
Figura 57	– Saída passa-alta do filtro para <i>Peak</i> máximo	58
Figura 58	– Saída passa-banda do filtro para <i>Peak</i> mínimo	59
Figura 59	– Saída passa-banda do filtro para <i>Peak</i> máximo	59
Figura 60	– Saída passa-baixa do filtro para <i>Peak</i> mínimo	60

Figura 61	–Saída passa-baixa do filtro para <i>Peak</i> máximo	60
Figura 62	– <i>Layout</i> da placa para o circuito desenvolvido	74
Figura 63	–Esquemático do circuito desenvolvido	75
Figura 64	–Esquemático do Boss Auto Wah	79
Figura 65	–Esquemático do EHX QTron+	80
Figura 66	–Esquemático do DOD Envelope Filter 440	80
Figura 67	–Filtro de Variáveis de Estado	81
Figura 68	–Filtro de Variáveis de Estado	81
Figura 69	–Pinagem do <i>Teensy 3.6 (1)</i>	82
Figura 70	–Pinagem do <i>Teensy 3.6 (2)</i>	83
Figura 71	–Fonte simétrica para alimentação do circuito	84

Lista de tabelas

Tabela 1 – Especificações do projeto	76
Tabela 2 – Orçamento do projeto	77

Lista de abreviaturas e siglas

AC	Alternating Current, do inglês, Corrente Alternada
ADC	Analog-to-digital Converter, do inglês, Conversor Analógico-digital
DAC	Digital-to-analog Converter, do inglês, Conversor Digital-analógico
DC	Direct Current, do inglês, Corrente Contínua
FFT	Fast Fourier Transform, do inglês, Transformada Rápida de Fourier
LDR	Light Dependent Resistor, do inglês, Resistor Dependente de Luz
LED	Light Emitting Diode, do inglês, Diodo Emissor de Luz

Lista de símbolos

Ω	<i>Ohm</i> , unidade de resistência elétrica
ζ	<i>Dumping Factor</i> ou <i>Fator de Crista</i> de um filtro

Sumário

1	Introdução	12
2	Fundamentação Teórica	14
2.1	O Pedal Mu-tron III	14
2.1.1	Pré-processamento	17
2.1.2	Detector de Envoltória	17
2.1.2.1	Ciclo Positivo	18
2.1.2.2	Ciclo Negativo	19
2.1.2.3	Saída	20
2.1.3	Filtro de Variáveis de Estado	20
2.2	Discretização do Circuito	23
3	Metodologia	25
3.1	Materiais Utilizados	25
3.1.1	Hardware	26
3.1.2	Software	26
3.1.3	Projeto da Placa e Simulações	26
3.2	Projeto do Software	26
3.2.1	Ferramentas de Compilação	27
3.2.2	Organização e Fluxo de Desenvolvimento	27
3.2.3	Implementação	28
3.3	Pré-processamento	28
3.4	Parâmetros de Regulagem	30
3.5	Leitura do Sinal	31
3.6	Detector de Envelope	32
3.7	Filtro de Variáveis de Estado	32
3.7.1	Cálculo da Frequência de Corte	32
3.7.2	Implementação do Filtro	34
3.8	Pós-Processamento	35
3.9	Testes e Validação do Método	36
3.9.1	Bloco de Entrada	36
3.9.2	Detector de Envoltória	37
3.9.3	Filtro de Variáveis de Estado	38
4	Resultados e Discussões	40
4.1	Bloco de Entrada	40

4.1.1	Mu-tron III - Simulação	40
4.1.2	Circuito Implementado - Simulação	41
4.1.3	Circuito Implementado	44
4.2	Detector de Envoltória	46
4.2.1	Mu-tron III - Simulação	46
4.2.2	Software implementado - Simulação no MATLAB®	47
4.3	Filtro de Variáveis de Estado	47
4.3.1	Mu-tron III - Simulação	47
4.3.2	Software Implementado - Simulação no MATLAB®	50
4.3.3	Software Implementado - Implementação Física	52
	Conclusão	61
	Referências Bibliográficas	63
	Apêndices	65
	APÊNDICE A Rotinas para testes em MATLAB®	66
	APÊNDICE B Código completo em MATLAB®	68
	APÊNDICE C Código em C	70
	APÊNDICE D Layout e Esquemático do Circuito	74
	APÊNDICE E Especificações do Projeto	76
	APÊNDICE F Orçamento do Projeto	77
	Anexos	78
	ANEXO A Esquemáticos de Pedais de Efeito Auto-Wah Comerciais	79
	ANEXO B Esquemáticos de Filtros de Variáveis de Estado	81
	ANEXO C Pinagem do <i>Teensy 3.6</i>	82
	ANEXO D Fonte de Alimentação	84

1 Introdução

Pedais de efeito são frequentemente utilizados por músicos para modificar a sonoridade de seus instrumentos, e com isso adicionar novos timbres e texturas em suas performances. O sinal de áudio gerado pelos instrumentos pode ser modificado através de circuitos analógicos ou processado de forma digital através de microprocessadores (ZÖLZER, 2002). Com o avanço tecnológico de dispositivos microprocessadores e novas técnicas de processamento digital de sinais estes pedais tem se mostrado capazes de emular pedais de efeito analógicos com grande fidelidade e baixo custo (CHAMBERLIN, 1987).

O pedal de efeito Mu-tron III, escolhido como o objeto de estudo do projeto, foi criado por Mike Beigel e patenteado em 1973 (BEIGEL, 1973). Foi o primeiro pedal que implementa o efeito conhecido como Auto-Wah ou *Envelope Filter* (KEEN, 1999). Pode-se citar outros pedais do efeito Auto-Wah posteriormente desenvolvidos, como o Auto-Wah, da Boss®), o Q-Tron Plus, da EHX®) e o DOD Envelope Filter 440, da DOD Electronics®).

R. G. Keen (1999, p. 2) descreveu a construção de seu projeto chamado Neutron Filter como um pedal "elaborado com a mesma tecnologia do Mu-tron III para adquirir o mesmo som". Neste documento é mostrada a análise de cada etapa do seu circuito. O bloco principal do circuito é apresentado como um filtro de variáveis de estado, cuja saída pode ser selecionada entre passa-altas, passa-banda ou passa-baixa.

A revista Elektor Magazine (1986) descreveu uma configuração de um filtro de variáveis de estado para aplicação em um equalizador paramétrico. Chamberlin (1987, p. 214) também apresentou uma configuração de um filtro de variáveis de estado e sua conversão para um filtro digital. Zölzer (2002, p. 56) descreveu um filtro Auto-Wah como um "filtro cuja frequência de corte é controlada pelo sinal de entrada".

Dentro deste contexto, o projeto tem como objetivo reproduzir o efeito Auto-Wah de forma digital, utilizando como base o pedal de efeito Mu-tron III. É realizado inicialmente um estudo do circuito do pedal original, com seu processamento sendo dividido em três blocos, a saber: Pré-processamento, Detector de Envoltória e Filtro de Variáveis de Estado. Cada bloco tem seu comportamento analisado e sua resposta em frequência é determinada. Os resultados da análise são utilizados na implementação prática do projeto.

O bloco de Pré-processamento é implementado no projeto em um circuito analógico, que também é responsável por modificar o sinal de entrada de forma que a sua leitura pelo microprocessador possa ser realizada.

Para a implementação prática do efeito foi escolhida a plataforma de desenvolvi-

mento Teensy 3.6, onde os blocos Detector de Envoltória e Filtro de Variáveis de Estado são implementados. A implementação de efeitos de áudio em plataforma digital embarcada é explorada em trabalhos recentes como o desenvolvimento de um *mixer* de áudio em plataforma Teensy/Arduino (POPP et al, 2015) e a construção de um pedal de distorção em plataforma Arduino (MURTHY et al, 2014). No endereço eletrônico da Sparkfun (BYRON, 2018) é apresentado um tutorial para montagem de um pedal de efeito utilizando a plataforma Teensy. Com base nos trabalhos encontrados e na bibliografia utilizada é desenvolvido um software para a escrita e implementação de algoritmos de processamento na placa Teensy 3.6.

O funcionamento do projeto é testado realizando medições no circuito analógico, onde são medidos valores máximo e mínimo de ganho do bloco de Pré-processamento, bem como a resposta em frequência do circuito para diferentes valores de amplitude de entrada, verificando o funcionamento do bloco do Filtro de Variáveis de Estado. Os resultados encontrados são comparados com os dados de simulação do circuito original do Mu-tron III.

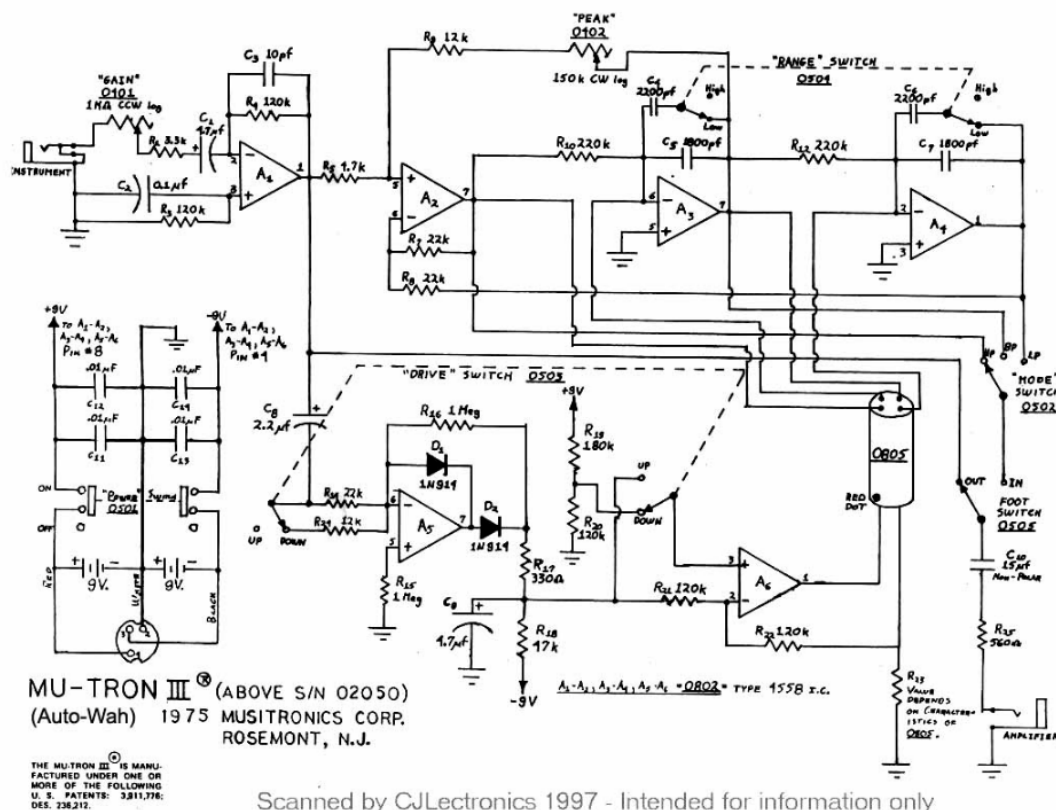
Espera-se que o projeto implementado reproduza com fidelidade o efeito produzido pelo pedal Mu-tron III, tanto nos testes realizados na placa quanto em testes práticos utilizando instrumentos musicais na entrada e ouvindo o sinal de saída reproduzido em um amplificador de áudio. Com a reprodução do efeito validada, o projeto final poderá ser utilizado em aplicações musicais e disponibilizado como produto comercial.

2 Fundamentação Teórica

2.1 O Pedal Mu-tron III

O pedal Mu-tron III, cujo esquemático é mostrado na figura 1, foi escolhido como base para o projeto por se tratar da primeira versão do efeito dentre os pedais disponíveis comercialmente, e também a versão que possui documentação mais detalhada (KEEN, 1999).

Figura 1 – Esquemático do Mu-tron III

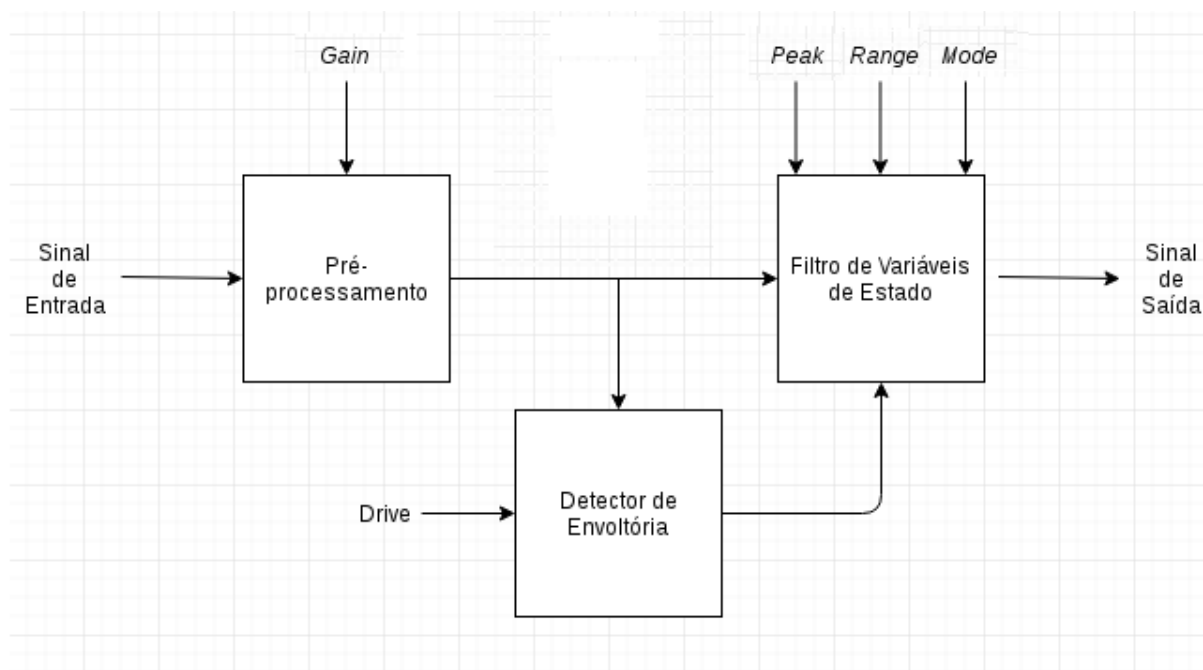


Fonte: copiado de <<https://www.talkbass.com/threads/cloning-pedals-from-circuit-schematics.1260298/>>

O funcionamento do circuito do Mu-tron III mostrado na figura 1 é explicado pelo diagrama de blocos da figura 2. O sinal de áudio de entrada, vindo dos captadores do instrumento musical, é manipulado por um Filtro de Variáveis de Estado cuja frequência de corte é uma variável do próprio sinal de entrada (BEIGEL, 1973). A amplitude do sinal terá seu valor máximo no momento em que as cordas do instrumento são postas a vibrar, com o sinal sendo atenuado até se extinguir. O valor máximo, ou valor de pico, do sinal irá depender da intensidade física com a qual as cordas são postas a vibrar e do

captador utilizado. Como regra geral, a amplitude máxima do sinal dos captadores está no intervalo entre 200 mV e 500 mV para captadores simples, e entre 400 mV e 1000 mV para captadores do tipo *Humbucker* (DAILEY, 2013).

Figura 2 – Diagrama de Blocos do Mu-tron III



Fonte: Autoria Própria, 2018

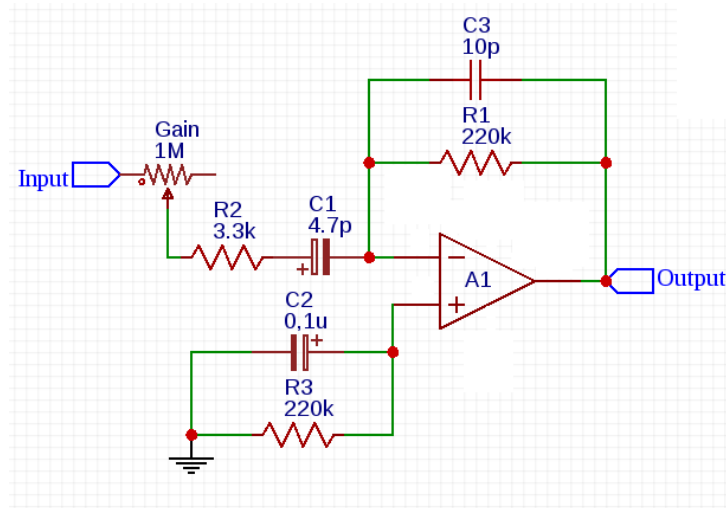
A figura 2 mostra também os parâmetros externos de regulação do Mu-tron III.

- *Gain*: controla o ganho de pré-processamento do sistema.
- *Drive*: controla a direção da envoltória do sinal. Irá determinar se a frequência de corte do filtro aumentará ou diminuirá com a amplitude do sinal de entrada.
- *Peak*: controla a ressonância na frequência de corte, ou fator Q do Filtro de Variáveis de Estado. Um aumento do fator Q representa um ganho maior do sinal na frequência de corte do filtro (CHAMBERLIN, 1987).
- *Range*: controla a faixa de valores de frequência de corte na qual o filtro atua.
- *Mode*: seleciona o sinal de saída do Filtro de Variáveis de Estado entre passa-alta, passa-banda e passa-baixa.

2.1.1 Pré-processamento

A etapa de pré-processamento do circuito analógico recebe o sinal de áudio vindo dos captadores do instrumento utilizado (BEIGEL, 1973). O sinal é amplificado e filtrado pelo circuito mostrado na figura 3.

Figura 3 – Bloco de entrada do Mu-tron III



Fonte: BEIGEL, 1973

O bloco de entrada é um amplificador de configuração inversora, com capacitores colocados para corte de frequências. O ganho deste bloco é regulável pelo potenciômetro *Gain* da figura 3. A função de transferência desta configuração é uma função das resistências R_1 , R_2 , do potenciômetro *Gain* e dos capacitores C_1 e C_3 (KEEN, 1999). O capacitor C_2 e o resistor R_3 são colocados na entrada positiva do amplificador e conectados ao terra para compensação de correntes de polarização do amplificador (KLIMACH, 1992). O circuito resultante é um filtro passa-banda cuja função de transferência é definida pela equação 1.

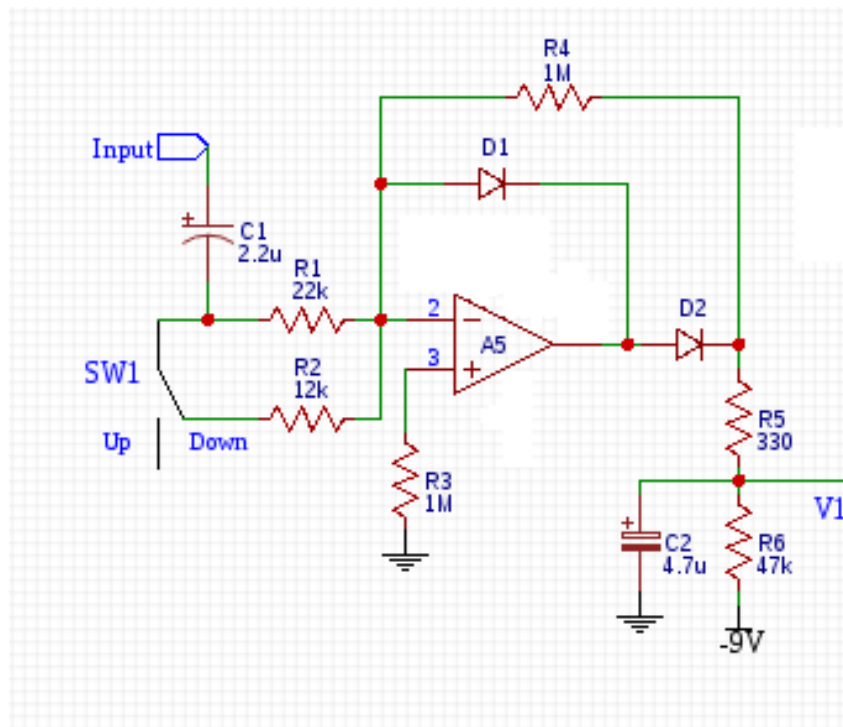
$$H(s) = -\left(\frac{R_1}{Gain + R_2}\right)\left(\frac{\frac{1}{R_1 C_3}}{s + \frac{1}{R_1 C_3}}\right)\left(\frac{s}{s + \frac{1}{(Gain + R_2) C_1}}\right) \quad (1)$$

2.1.2 Detector de Envoltória

Um detector de envoltória, ou detector de envelope, é um sistema cujo sinal de saída é igual ao contorno dos extremos do sinal de entrada. Este contorno é chamado de envelope ou envoltória do sinal (LATHI, 1998). No circuito do Mu-tron III, o detector de envoltória é o bloco responsável pelo controle do sistema.

O sinal passa inicialmente pelo amplificador A_5 mostrado na figura 4. Os diodos D_1 e D_2 mostrados no circuito são responsáveis por impedir a passagem de valores negativos de tensão, retificando o sinal.

Figura 4 – Detector de Envoltória do Mu-tron III



Fonte: BEIGEL, 1973

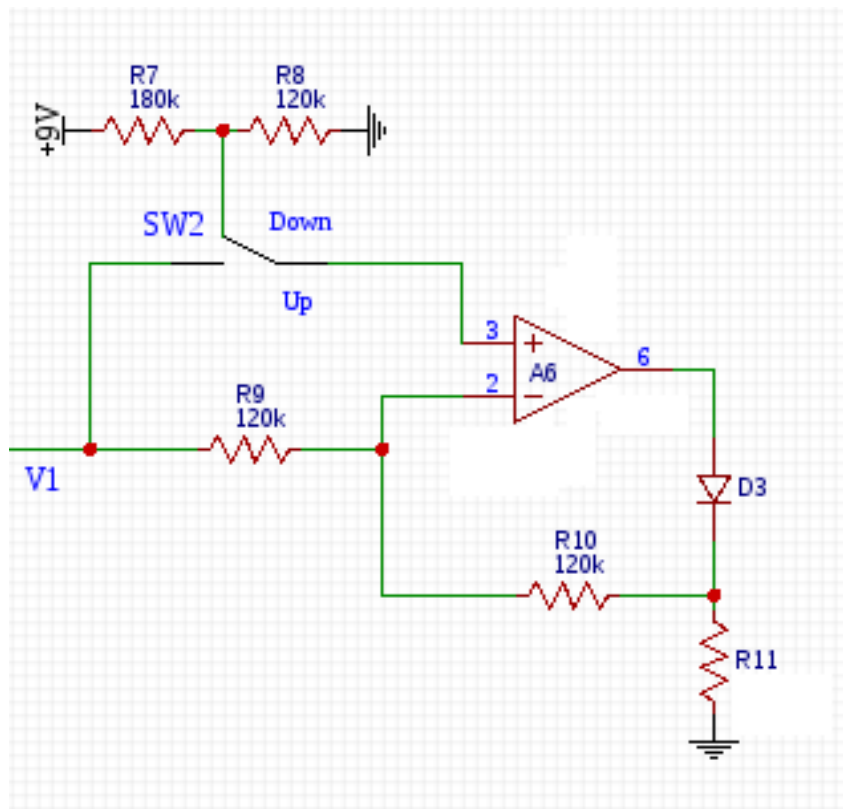
Após, o sinal passa pelo circuito mostrado na figura 5, que pode ser um circuito de configuração retificadora ou inversora, dependendo da posição da chave $SW2$. Dadas as características do circuito, há funções de transferência distintas para os ciclos positivo e negativo do sinal de entrada.

2.1.2.1 Ciclo Positivo

Durante o ciclo positivo do sinal de entrada, os diodos D_1 e D_2 do amplificador A_5 são diretamente polarizados, com o sinal passando pelo resistor R_5 e carregando o capacitor C_2 . O sinal passará então para o amplificador A_6 mostrado na figura 5.

As chaves $SW1$ e $SW2$ são acionadas simultaneamente e controlam o parâmetro de regulação *Range* do Mu-tron III. Se as chaves estiverem na posição *Down*, o amplificador A_6 da figura 5 estará em uma configuração inversora, com o valor da referência de tensão dado pelos resistores R_7 e R_8 . A tensão na saída do amplificador será o inverso da tensão em sua entrada e os resistores R_9 e R_{10} possuem valores iguais, tornando o ganho unitário. Se as chaves estiverem na posição *Up*, a referência de tensão será a própria tensão de entrada, multiplicada por um ganho em função das resistências R_5 e R_6 , e o amplificador A_6 estará em uma configuração seguidora de tensão, ou *buffer*. Nessa configuração a tensão de saída é igual à tensão de entrada. A função de transferência do bloco para as chaves na posição *Up* é dada pela equação 2.

Figura 5 – Detector de Envoltória do Mu-tron III



Fonte: BEIGEL, 1973

$$\frac{V_o}{V_i} = \frac{R_6}{R_1 + R_5 + R_6} \quad (2)$$

Quando as chaves estiverem an posição *Down* a função de transferência do bloco é dada pela equação 3

$$\frac{V_o}{V_i} = -\frac{R_6}{(R_1//R_2) + R_5 + R_6} + \frac{9R_8}{R_7 + R_8} \quad (3)$$

2.1.2.2 Ciclo Negativo

Quando o sinal começar a decair, o capacitor C_2 da figura 4 começará a descarregar através do resistor R_6 , ligado em -9 V. Durante o ciclo negativo do sinal de entrada, os diodos D_1 e D_2 da figura 4 estão em polarização reversa, cortando o sinal de entrada. A tensão de descarga do capacitor então será a tensão de entrada do amplificador A_6 , que será em função da tensão de pico do sinal de entrada V_{max} . A função de transferência do bloco com as chaves na posição *Up* é dada pela equação 4.

$$\frac{V_o}{V_i} = V_{max}e^{-t/R_6C_2} \quad (4)$$

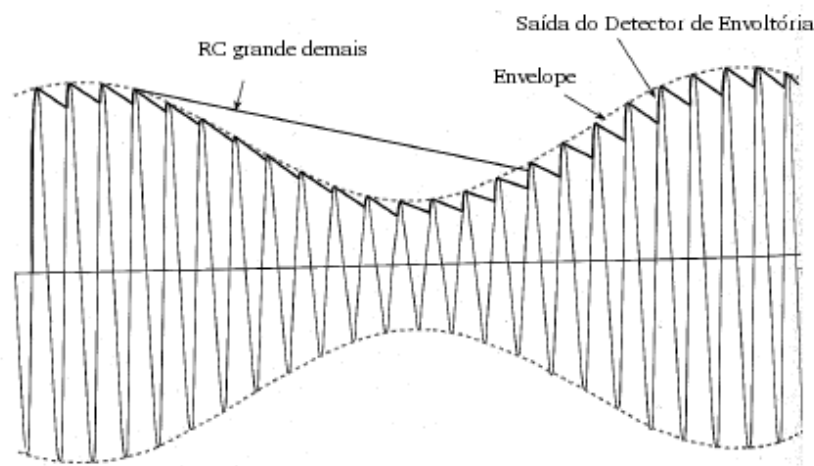
Com as chaves na posição *Down* a função de transferência é dada pela equação 5.

$$\frac{V_o}{V_i} = -V_{max}e^{-t/R_6C_2} + \frac{9R_8}{R_7 + R_8} \quad (5)$$

2.1.2.3 Saída

O sinal amplificado passa pelo Diodo representado por D_3 na figura 5. Este diodo é um LED que está envolvido no controle da frequência de corte do bloco do Filtro de Variáveis de Estado. Combinando as respostas para os ciclos positivo e negativo do sinal de entrada, o sinal de saída que passa pelo LED é o contorno entre os picos de tensão do sinal de entrada. Os sinais de entrada e saída para diferentes valores de R_6 e C_2 são mostrados na figura 6.

Figura 6 – Sinais de entrada e saída do detector de envelope



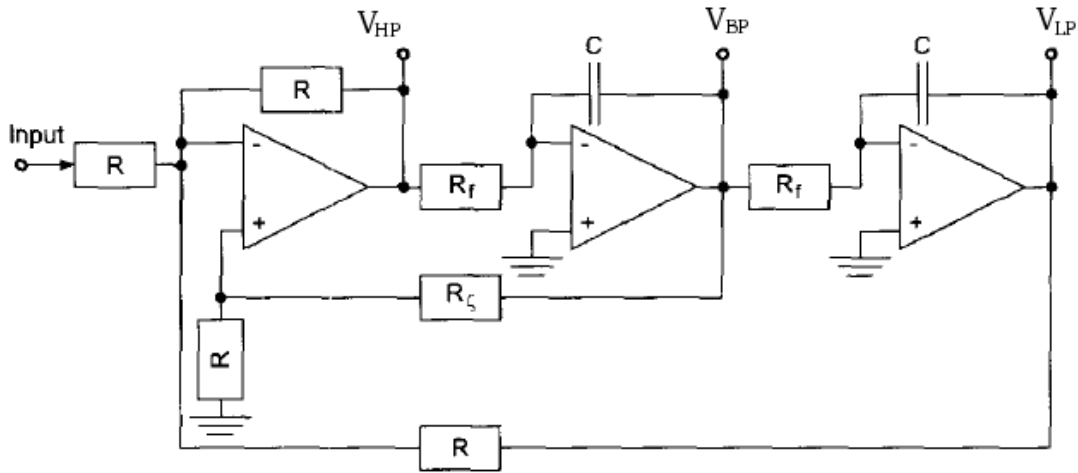
Fonte: LATHI, 1998

2.1.3 Filtro de Variáveis de Estado

O bloco principal do circuito é composto por um Filtro de Variáveis de Estado (BEIGEL, 1973). Este filtro é composto por três amplificadores operacionais ligados em cascata, sendo o amplificador da entrada em configuração subtratora e os dois amplificadores seguintes em configuração integradora. Cada saída dos amplificadores constitui uma saída do filtro, sendo uma passa-alta, uma passa-banda e uma passa-baixa, todas com a mesma frequência de corte (KLIMACH, 1992). A topologia descrita por Zölzer (2002), utilizada no projeto, é mostrada na figura 7.

A frequência de corte f_c e o fator de ressonância Q são controlados pelas resistências R_f , R e R_c , mostradas na figura 7. Com os valores de resistências e capacitâncias de

Figura 7 – Filtro de Variáveis de Estado



Fonte: ZÖLZER, 2002

mesmo nome sendo iguais, a frequência de corte é determinada de acordo com a equação 6 (ZÖLZER, 2002).

$$f_c = \frac{1}{2\pi R_f C} \quad (6)$$

Da mesma forma, o fator de ressonância Q é determinado pela equação 7.

$$Q = \frac{1}{\zeta} = \frac{2(R + R_\zeta)}{R} \quad (7)$$

A função de transferência, utilizando como exemplo o filtro da figura 7, pode ser calculada a partir das saídas passa-alta, passa-banda e passa-baixa do circuito (ZÖLZER, 2002).

$$V_{LP} = -\frac{1}{R_f C s} V_{BP} \quad (8)$$

$$V_{BP} = -\frac{1}{R_f C s} V_{HP} \quad (9)$$

$$V_{HP} = -\frac{R_f}{R} V_{LP} + \left(1 + \frac{R_f}{R}\right) \left(\frac{R_\zeta}{R + R_\zeta} V_i + \frac{R}{R + R_\zeta} V_{BP}\right) \quad (10)$$

Ao isolar as respectivas equações para deixá-las em função apenas da tensão de entrada V_i e uma das tensões de saída, encontra-se as funções de transferência do Filtro de Variáveis de Estado (ZÖLZER, 2002).

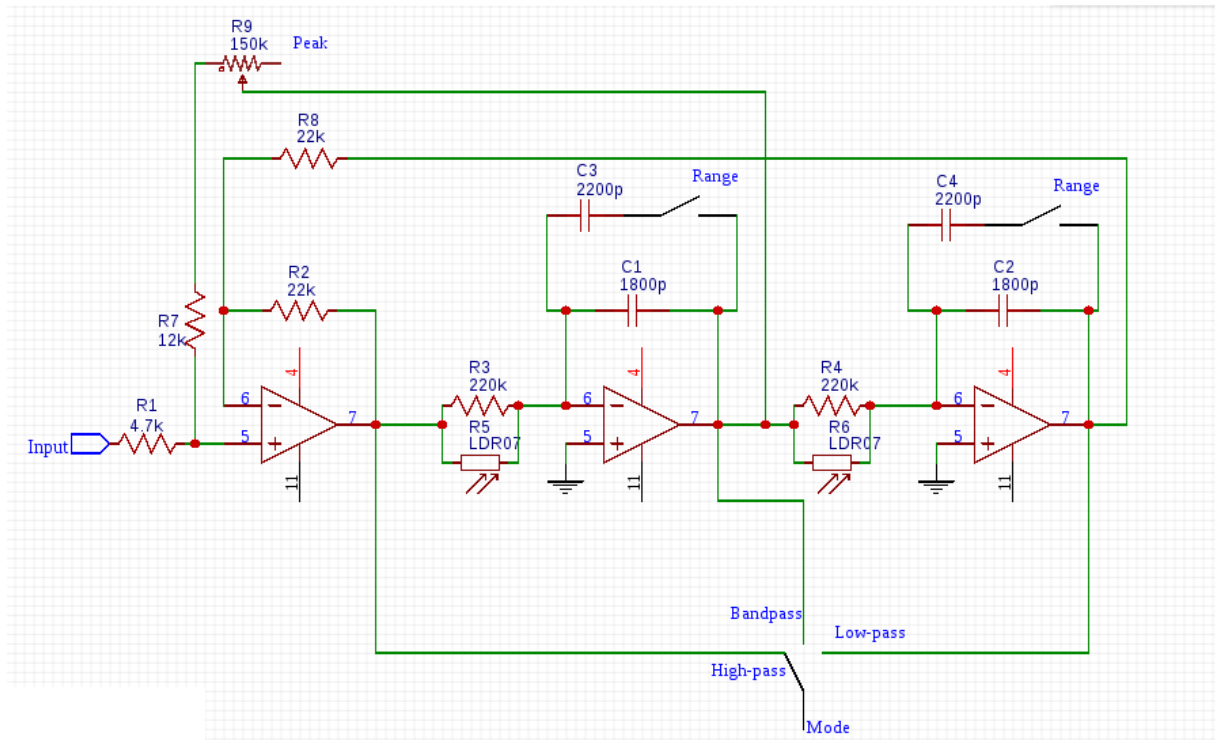
$$T_{HP}(s) = \frac{\frac{2}{1+\frac{R}{R_c}} s^2}{s^2 + \frac{2}{\frac{R}{R_c}} \frac{1}{R_f C} s + \frac{1}{R_f^2 C^2}} \quad (11)$$

$$T_{BP}(s) = \frac{-\frac{2}{1+\frac{R}{R_c}} \frac{1}{R_f C} s}{s^2 + \frac{2}{\frac{R}{R_c}} \frac{1}{R_f C} s + \frac{1}{R_f^2 C^2}} \quad (12)$$

$$T_{LP}(s) = \frac{-\frac{2}{1+\frac{R}{R_c}} \frac{1}{R_f^2 C^2}}{s^2 + \frac{2}{\frac{R}{R_c}} \frac{1}{R_f C} s + \frac{1}{R_f^2 C^2}} \quad (13)$$

No Mu-tron III, mostrado na figura 8, o controle do filtro é realizado pela saída do bloco detector de envoltória. O circuito original do Mu-tron III possui um LED na saída do circuito detector de envoltória, como mostrado na figura 5. Este LED emite luz cuja intensidade depende da amplitude da tensão do sinal de áudio de entrada e que incide sobre os LDRs do Filtro de Variáveis de Estado mostrado na figura 8.

Figura 8 – Filtro de Variáveis de Estado do Mu-tron III



Fonte: Adaptado de BEIGEL, 1973

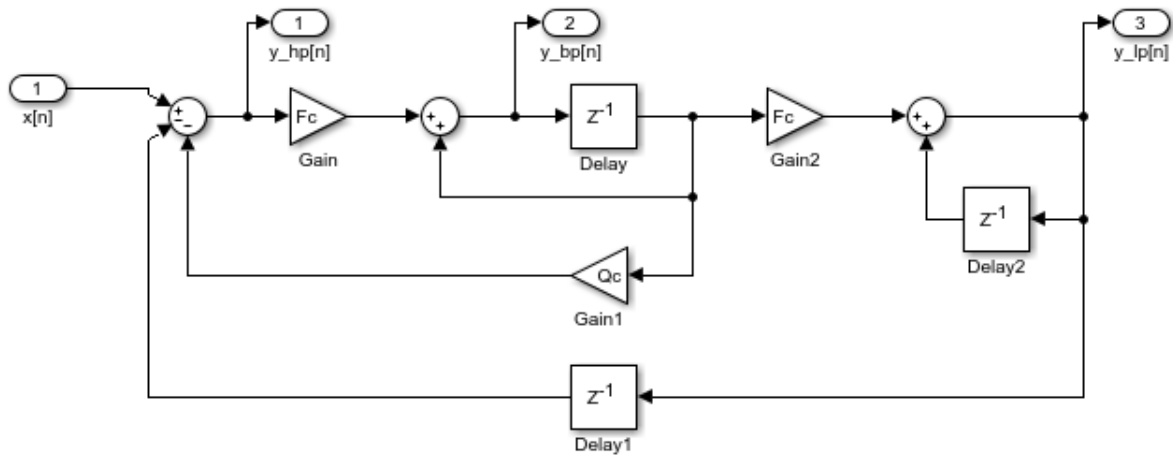
A incidência da luz nos LDRs causa uma variação nos valores de resistência responsáveis pelo valor da frequência de corte do filtro. Como a luz é incidida pelo LED conectado na saída do Detector de Envoltória do circuito, o valor da frequência de corte do filtro irá variar com a amplitude do sinal de entrada do sistema.

O potenciômetro R_9 da figura 8 é responsável pelo fator de ressonância, ou Q , do filtro e pode ser regulado pelo usuário dependendo da aplicação desejada. A chave *Range* adiciona capacitâncias em paralelo nos integradores, modificando a faixa de valores que a frequência de corte do filtro pode assumir.

2.2 Discretização do Circuito

Zölzer (2002) propõe uma adaptação digital do Filtro de Variáveis de Estado, como mostrado na figura 9. Os amplificadores integradores são representados por blocos de atraso, o amplificador subtrator por um bloco somador e as resistências por blocos de ganho.

Figura 9 – Filtro de Variáveis de Estado Digital



Fonte: Adaptado de ZÖLZER, 2002

O ganho F_C do sistema é proporcional à frequência de corte f_c do filtro de acordo com a equação 14 (ZÖLZER, 2002).

$$F_C = 2 * \sin\left(\frac{\pi * f_c}{F_s}\right) \quad (14)$$

Da mesma forma, o ganho Q_C é proporcional ao fator de ressonância Q do filtro de acordo com a equação 15 (ZÖLZER, 2002).

$$Q_C = \frac{2}{Q} \quad (15)$$

As saídas y_{HP} , y_{BP} , e y_{LP} do diagrama da figura 9, são calculadas através das equações 16, 17 e 18 (ZÖLZER, 2002).

$$y_{HP}[n] = x[n] - y_{LP}[n-1] - Q_C y_{BP}[n-1] \quad (16)$$

$$y_{BP}[n] = F_C y_{HP}[n] + y_{BP}[n-1] \quad (17)$$

$$y_{LP}[n] = F_C y_{BP}[n] + y_{LP}[n-1] \quad (18)$$

Para encontrar a função de transferência do filtro, as expressões de saída são passadas para o domínio da transformada z , equivalente da transformada de Laplace para sinais discretos (OPPENHEIM, 1998). Pela propriedade do deslocamento no tempo, um atraso no tempo resulta em uma multiplicação por z^{-1} no domínio da transformada z (OPPENHEIM, 1998). Com isso, as expressões tornam-se:

$$Y_{HP}(z) = X(z) - Y_{LP}(z)z^{-1} - Q_C Y_{BP}(z)z^{-1} \quad (19)$$

$$Y_{BP}(z) = F_C Y_{HP}(z) + Y_{BP}(z)z^{-1} \quad (20)$$

$$Y_{LP}(z) = Y_{LP}(z)z^{-1} + F_C Y_{BP}(z)z^{-1} \quad (21)$$

Resolvendo o sistema de três equações são encontradas as funções de transferência do filtro:

$$H_{HP}(z) = \frac{Y_{HP}(z)}{X(z)} = \frac{(1 - z^{-1})^2}{1 + (F_C^2 + F_C Q_C - 2)z^{-1} + (1 - F_C Q_C)z^{-2}} \quad (22)$$

$$H_{BP}(z) = \frac{Y_{BP}(z)}{X(z)} = \frac{F_C(1 - z^{-1})}{1 + (F_C^2 + F_C Q_C - 2)z^{-1} + (1 - F_C Q_C)z^{-2}} \quad (23)$$

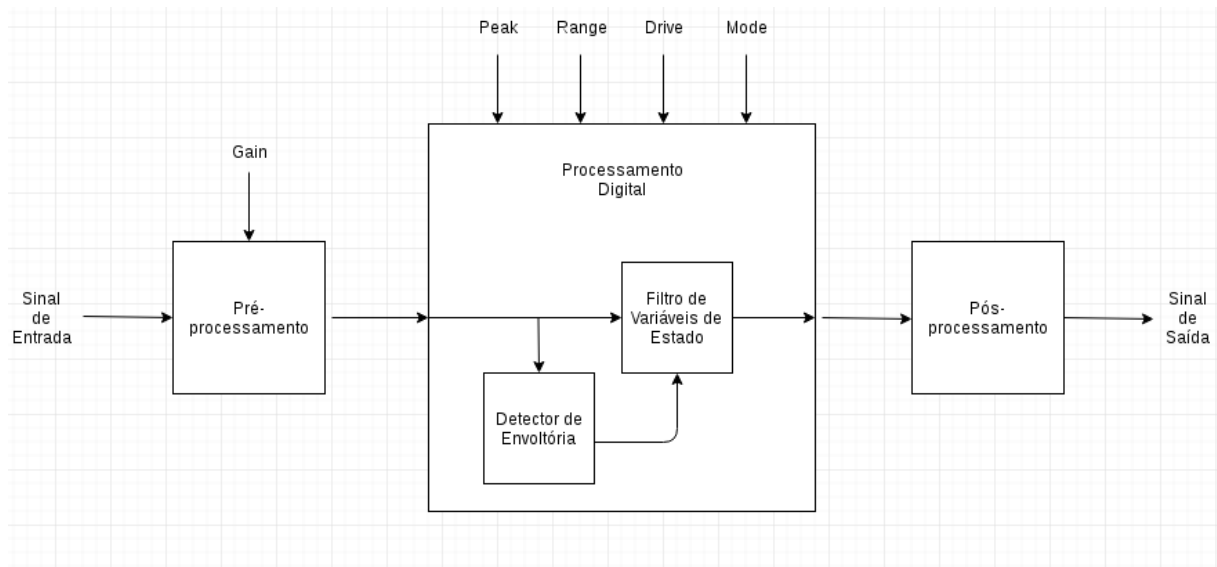
$$H_{LP}(z) = \frac{Y_{LP}(z)}{X(z)} = \frac{F_C^2 z^{-1}}{1 + (F_C^2 + F_C Q_C - 2)z^{-1} + (1 - F_C Q_C)z^{-2}} \quad (24)$$

A resposta do filtro é selecionável entre as saídas passa-alta, passa-banda ou passa-baixa. Os valores de saída dependerão das constantes F_C e Q_C , sendo F_C um valor dependente da frequência de corte do filtro que será calculada no bloco Detector de Envoltória do sistema. O valor de Q_C é dependente do parâmetro regulável Q , com um maior valor de Q representando um maior ganho do sinal na frequência de corte do filtro.

3 Metodologia

O projeto tem como objetivo realizar o processamento do sinal de áudio de acordo com o diagrama de blocos da figura 10, que é uma adaptação do diagrama de blocos do Mu-tron III mostrado na figura 2.

Figura 10 – Diagrama de Blocos do Auto-Wah Digital



Fonte: Autoria Própria, 2018

O diagrama da figura 10 reproduz as etapas de processamento do diagrama da figura 2, com a mudança no método de processamento do meio analógico para o meio digital nas etapas do Detector de Envoltória e do Filtro de Variáveis de Estado. Para a realização dessas etapas, é necessário adicionar etapas de pré e pós processamento para que o sinal seja lido e posteriormente escrito pelo microprocessador.

As variáveis de regulagem são definidas como parâmetros de entrada no microprocessador, com exceção do parâmetro *Gain*, que faz parte do bloco analógico de Pré-processamento.

3.1 Materiais Utilizados

A escolha dos materiais foi realizada levando em conta a sua disponibilidade, seu custo-benefício e sua capacidade de realizar os procedimentos requeridos. Os materiais escolhidos mostraram-se adequados para a aplicação realizada.

3.1.1 Hardware

Para a etapa de processamento digital foi escolhida a placa de desenvolvimento Teensy 3.6, que possui um processador ARM Cortex M4 de 180 MHz com Unidade de Ponto Flutuante, 1 MB de Memória Flash, 256 kB de Memória RAM e 4 kB de Memória EEPROM. O Teensy 3.6 possui 2 Conversores Analógico-Digitais (ADC) com resolução de 13 bits, podendo ser acessados através de 25 pinos analógicos, 58 pinos digitais e entrada Micro USB, utilizada para conexão com o computador.

Figura 11 – Teensy 3.6



Fonte: STOFFREGEN, 2015

3.1.2 Software

O algoritmo de processamento foi desenvolvido no software MATLAB® R2012b, que possui ferramentas de trabalho e bibliotecas que facilitam a análise do código e a detecção de erros. Através deste software é possível utilizar ferramentas gráficas para analisar o sinal, detectar erros de sintaxe rapidamente e adquirir um sinal de áudio e realizar seu processamento para ouvi-lo posteriormente. Após, o algoritmo é convertido para a linguagem C e por meio da biblioteca GNU Compiler Collection (GCC) é implementado um software para implementação do código de processamento no Teensy 3.6.

3.1.3 Projeto da Placa e Simulações

O circuito com as etapas de pré e pós processamento foi elaborado e simulado nos programas Multisim 14, da National Instruments® e Microcap 10, que possuem ferramentas de simulação adequadas para verificar o seu funcionamento. Após o seu desenvolvimento, o circuito é projetado em uma placa de circuito impresso utilizando-se o software EasyEDA.

3.2 Projeto do Software

Para a realização do processo de leitura, processamento e posterior escrita do sinal no Teensy 3.6 é implementado um software que permite o controle do microprocessador a

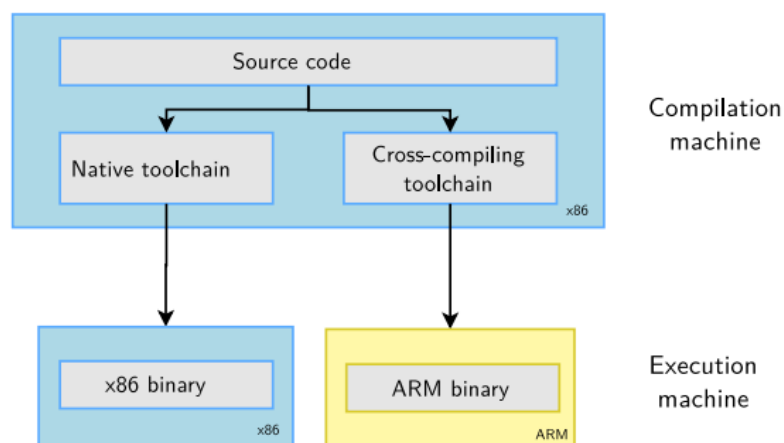
partir do Desktop. Através da porta Micro USB são inseridas instruções para o controle do microprocessador e de seus periféricos.

3.2.1 Ferramentas de Compilação

A escrita do software no microprocessador é realizada através de um compilador cruzado. Utilizando um computador com sistema operacional Linux o compilador cruzado é implementado através da biblioteca de compiladores GNU Compiler Collection (GCC) com a biblioteca de ferramentas GNU Binary Utilities (Binutils). A construção do compilador cruzado é realizada através da *toolchain* disponibilizada por Johann (2018).

Como mostrado na figura 12, o compilador cruzado é responsável por interpretar o código escrito em linguagem C e gerar uma aplicação que é escrita na memória flash do microprocessador, onde a aplicação é executada e traduz os comandos do código para o microprocessador.

Figura 12 – Estrutura de um compilador cruzado



Fonte: BOOTLIN, 2018

3.2.2 Organização e Fluxo de Desenvolvimento

O software é organizado em seções responsáveis por realizar sequencialmente cada etapa de execução. É realizado inicialmente o *bring-up* da placa, onde é verificado o funcionamento dos componentes do microprocessador para a execução do programa. Após, são compilados os componentes de software básico, sendo estes o C Runtime, que é o sistema responsável pela execução do código em linguagem C durante seu funcionamento e as bibliotecas *kinetis.h* e *core_pins.h*, disponibilizadas por Stoffregen (2018), que são responsáveis pelo controle dos periféricos da placa, entre eles o ADC, DAC e pinos digitais. A aplicação do programa então é compilada para a implementação do algoritmo de processamento. Então é construída a imagem binária que é carregada na memória *flash*

da placa, onde é incluído um *linker script* utilizando o *linker arm-elf-ld* para a alocação das seções do código e dos dados para os endereços de memória do processador e a sua configuração. A organização do software é automatizada em um *makefile* que executa as seções na ordem estabelecida.

3.2.3 Implementação

O software é configurado para realizar a leitura do sinal com uma resolução de 12 bits e taxa de amostragem de 48 kHz. Os valores adquiridos são registrados em um *buffer* sobre o qual são realizadas as operações de processamento. O *buffer* é implementado em uma aplicação FIFO (*first-in-first-out*), onde cada novo valor adquirido pelo ADC é registrado no início do *buffer*, ao mesmo tempo em que o último valor do *buffer* é registrado na saída. Cada valor do *buffer* é então utilizado na rotina de processamento do sinal, onde é calculado o valor de saída. Ao fim da etapa de processamento, o valor de saída é então escrito na DAC do microprocessador.

Para garantir o funcionamento das operações dependentes de *timing*, o processo de leitura e escrita é implementado em uma interrupção que é realizada a cada intervalo de tempo pré-determinado. Essas operações são realizadas de forma assíncrona em relação às operações de processamento para evitar que elas sejam afetadas pelo tempo do processamento.

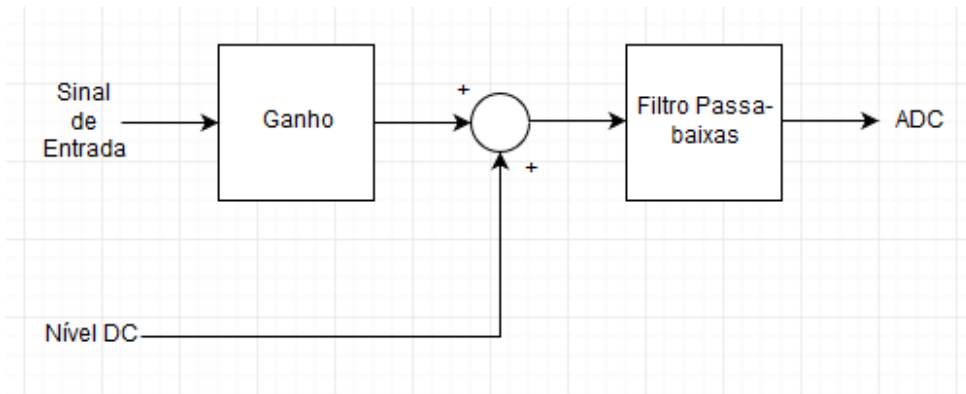
3.3 Pré-processamento

A etapa de entrada do sistema, responsável por tornar o sinal adequado para ser adquirido pelo microprocessador, é mostrada na figura 13. O ADC do Teensy 3.6 aceita valores de tensão entre 0 V e 3.3 V (YIU, 2007), e com isso o nível médio do sinal é deslocado de 0 V para 1.65 V para que o sinal de áudio possa ser lido pelo ADC. Sua amplitude é também amplificada para preencher um maior número de valores do ADC e melhorar a resolução da leitura.

A etapa de entrada do projeto é mostrada na figura 14. O sinal passa inicialmente pelo capacitor de desacoplamento C_3 , com os resistores R_2 e R_3 sendo utilizados para casamento de impedância com os captadores do instrumento e como impedância de entrada, respectivamente.

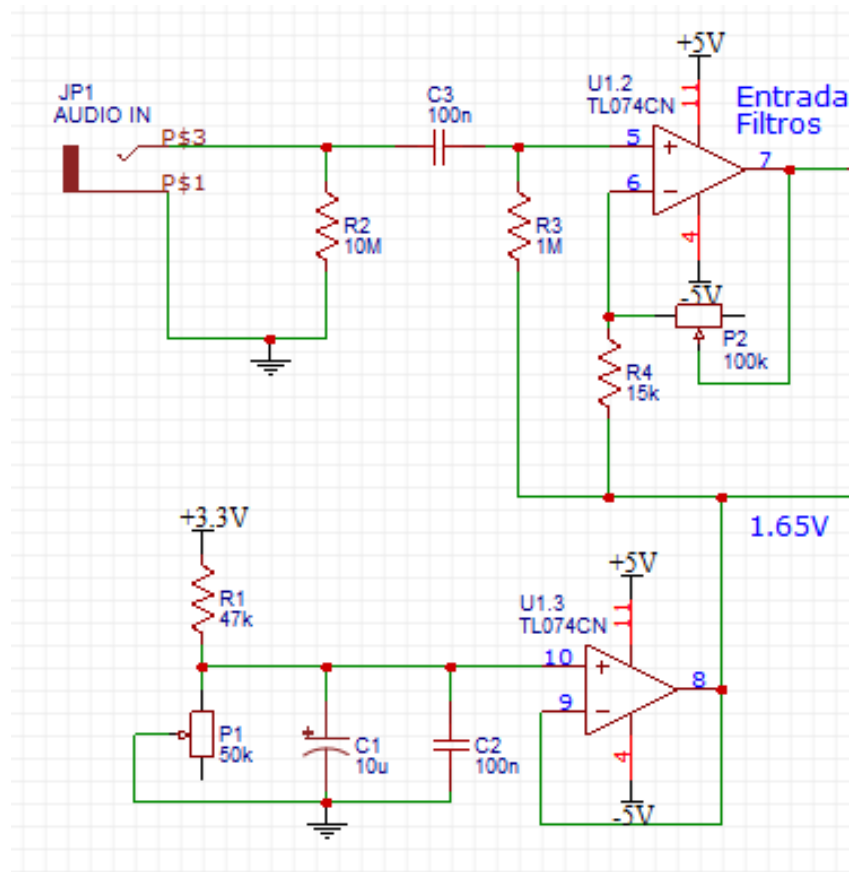
O nível DC de 1.65 V é somado ao sinal de entrada através da configuração amplificadora formada pelo amplificador $U_{1,2}$, que faz parte de uma configuração não-inversora. A entrada negativa do amplificador é ligada na referência de tensão gerada pelo divisor de tensão formado pelo resistor R_1 e pelo potenciômetro P_1 , gerando um terra artificial para o sinal em 1.65 V ao invés dos 0 V do circuito. Os capacitores C_1 e C_2 são colocados como filtros para oscilações e o amplificador $U_{1,3}$ como *buffer*.

Figura 13 – Bloco de Pré-Processamento do Projeto



Fonte: Autoria Própria, 2018

Figura 14 – Bloco de Entrada do Projeto



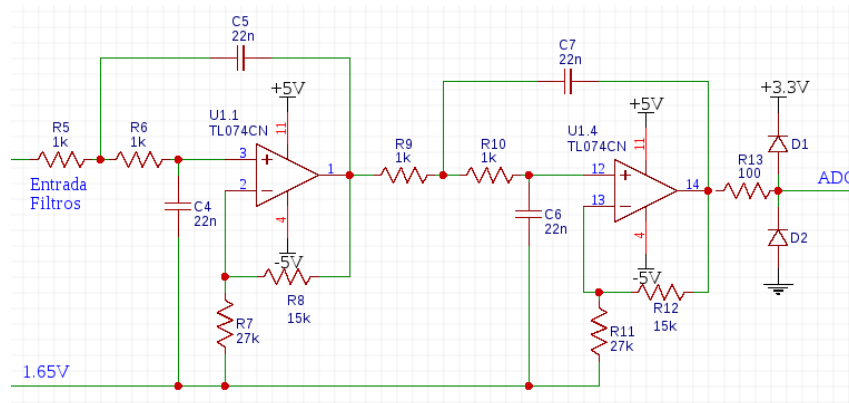
Fonte: Autoria Própria, 2018

O ganho desta configuração será dado pelo potenciômetro P2 e pelo resistor R4 através da equação 25.

$$G = \frac{P2}{R4} \tag{25}$$

Antes de ser lido pelo ADC, o sinal passa por um bloco de filtro passa-baixas para limitar a banda de frequência do sinal e evitar que a aquisição pelo ADC do microprocessador sofra com fenômenos de *aliasing*.

Figura 15 – Filtros Anti-Aliasing de Entrada



Fonte: Aatoria Própria, 2018

São utilizados no circuito dois filtros passa-baixa *Butterworth* de segunda ordem, que atenuam o sinal em 80 dB por década acima da frequência de corte. Optou-se por utilizar um valor de corte de 8 kHz já que, de acordo com Dailey (2007), a maior parte dos valores de frequência no som de guitarras e baixos está abaixo deste valor. Com isso, os valores comerciais de resistores e capacitores utilizados foram escolhidos de forma que a frequência de corte dos filtros seja de aproximadamente 8 kHz. Os resistores de 15 k Ω e 27 k Ω inserem um ganho de aproximadamente 1.55 em cada um dos filtros.

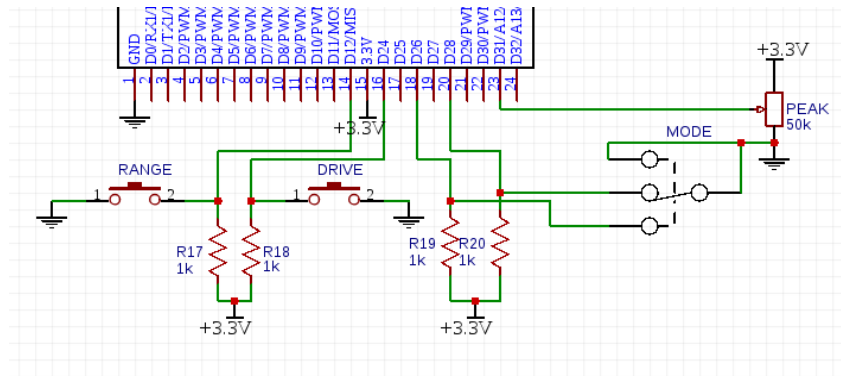
Para compensar erros causados pelo *offset* dos amplificadores operacionais e garantir um nível médio de 1.65 V no sinal lido pelo ADC, o divisor de tensão da figura 14 é feito com o potenciômetro P_1 no lugar de um dos resistores, que permite ajustar o valor médio do sinal em 1.65 V.

Os diodos D_1 e D_2 e o resistor R_{13} na saída do bloco da figura 15 são utilizados para proteção do ADC. Os diodos limitam a tensão de saída entre 0 V e 3.3 V, valores máximos de leitura do ADC, cortando o sinal caso ele ultrapasse estes valores. Já o resistor R_{13} é utilizado para limitar a corrente de entrada do ADC.

3.4 Parâmetros de Regulagem

Com exceção do parâmetro *Gain*, implementado no bloco de pré-processamento do sistema, os parâmetros de regulagem do sistema são valores de tensão aplicados em pinos do microprocessador, sendo lidos e utilizados no algoritmo de processamento. Sua disposição é mostrada na figura 16.

Figura 16 – Parâmetros de Regulagem do Sistema



Fonte: Autoria Própria, 2018

Os parâmetros *Range* e *Drive*, são implementados através do mesmo método. Uma chave seleciona o valor de tensão lida em um dos pinos digitais do Teensy 3.6 entre nível alto ou baixo. O valor de tensão lido no pino é posteriormente utilizado no algoritmo.

O parâmetro *Mode* possui 3 (três) valores distintos no circuito analógico. Para reproduzi-lo no projeto são utilizados dois pinos digitais do Teensy 3.6 e uma chave que possui 3 posições. A posição da chave irá deixar um dos pinos em tensão de nível lógico alto, ou os dois pinos em nível lógico baixo. O valor lido pelos pinos é utilizado no algoritmo.

O parâmetro *Peak* é lido por meio de um potenciômetro que seleciona a tensão lida por um dos ADCs do Teensy 3.6. A tensão na entrada irá variar entre 0 V e 3.3 V, e seu valor é lido e utilizado no algoritmo.

3.5 Leitura do Sinal

O sinal lido pelo ADC é representado por valores que variam entre 0 e 4095, que representam as tensões entre 0 V e 3.3 V lidas com resolução de 12 bits. Para o uso do sinal no código, os valores de entrada são modificados, com o nível DC adicionado para o sinal ser adquirido pelo ADC sendo removido e o sinal sendo normalizado, para que seus valores variem entre -1 e 1. Esta operação é realizada através da equação 26.

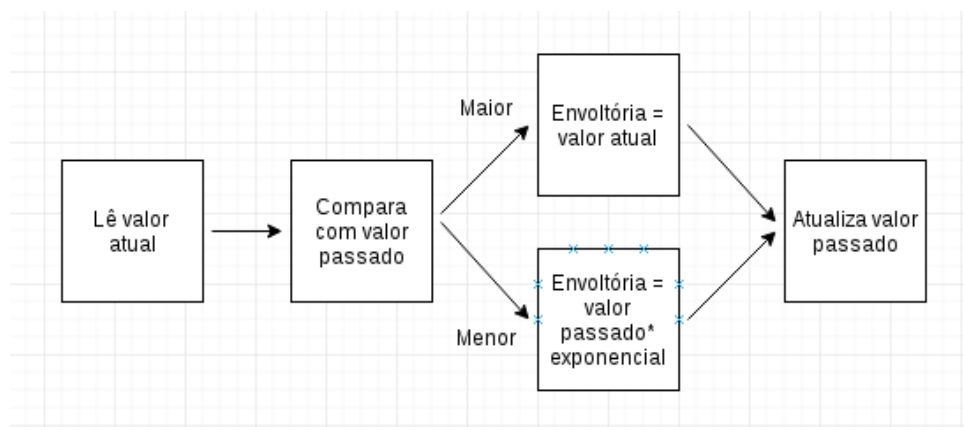
$$signal = \frac{(signal - 2048.0)}{2048.0} \quad (26)$$

Nota-se que essa operação é aproximada, tendo em vista que um dos valores é ocupado pelo valor zero. Com isso, o número de valores positivos não será igual ao número de valores negativos, variando entre -2048 e 2047 ou entre -2047 e 2048.

3.6 Detector de Envelope

O bloco detector de envelope é implementado de forma digital no microprocessador. Para reproduzir o processamento realizado pelo circuito do Mu-tron III nessa etapa foi elaborado um algoritmo que segue o diagrama de blocos da figura 17.

Figura 17 – Diagrama de Blocos do Detector de Envelope



Fonte: Autoria Própria, 2018

Baseando-se nas funções de transferência do circuito analógico mostradas na fundamentação teórica, no ciclo positivo o valor de saída é igual ao valor de entrada multiplicado por um ganho, e no ciclo negativo o valor de saída é igual ao valor de saída anterior multiplicado por uma exponencial. O algoritmo utiliza de um condicional para determinar se o sinal está em seu ciclo positivo ou negativo e após calcula o valor de saída. Após o cálculo, o valor de saída anterior é atualizado com os valor atual para realizar o cálculo nos próximos valores que o ADC adquirir. Como adaptação ao circuito analógico, a direção da envoltória controlada pela chave *Drive* é determinada no bloco do Filtro de Variáveis de Estado, quando a frequência de corte do filtro é calculada. Como o sinal de entrada foi normalizado após sua leitura, valor da tensão de entrada está dentro do intervalo de valores entre 0 e 1. Para a utilização no algoritmo do cálculo da frequência de corte, o ganho do bloco do Mu-tron III não é considerado, com o ganho na implementação sendo considerado unitário.

3.7 Filtro de Variáveis de Estado

3.7.1 Cálculo da Frequência de Corte

Após o cálculo da envoltória do sinal, o seu valor é utilizado no código para calcular a frequência de corte do filtro. A variável de entrada *Drive* tem valor lógico 0 ou 1 e controla a direção da envoltória. Se o seu valor for 0 a frequência de corte irá aumentar com o aumento do valor da envoltória, e se o seu valor for 1 o valor da frequência de corte

será inversamente proporcional ao valor da envoltória. A frequência de corte é calculada para valor 0 da variável *Drive* através da equação 27.

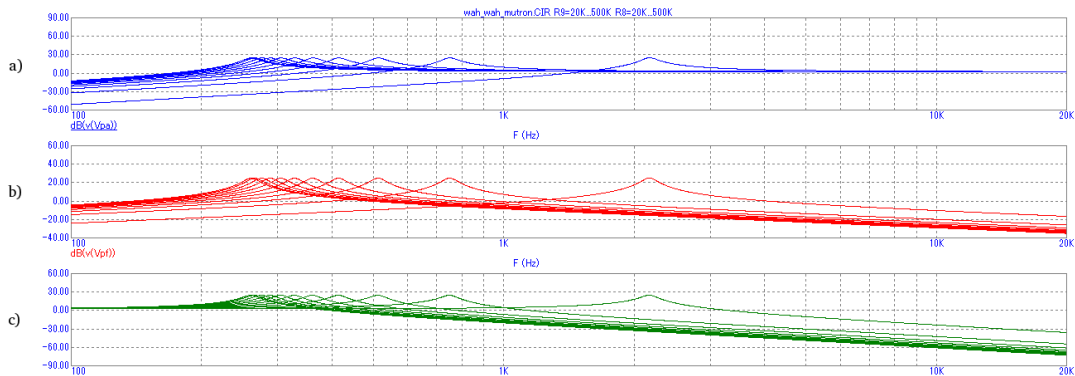
$$f_c = f_{min} + (f_{max} - f_{min}) * env \quad (27)$$

Para a variável *Drive* com valor 1, a frequência de corte é calculada através da equação 28.

$$f_c = f_{min} + (f_{max} - f_{min}) * (1 - env) \quad (28)$$

A variável f_c é a frequência de corte do filtro e env é o valor da envoltória. f_{max} e f_{min} são constantes e representam a frequência máxima e mínima de corte do filtro, respectivamente. Seus valores são obtidos através de testes em simulações do circuito analógico do Mu-tron III realizadas no software Microcap 10. Nos testes, mostrados nas figuras 18 e 19, o filtro de variáveis de estado do Mu-tron III tem sua resposta em frequência analisada para sinais de diferentes amplitudes.

Figura 18 – Resposta a) passa-alta, b) passa-banda e c) passa-baixa do filtro para diferentes valores de frequência de corte (parâmetro *Range* na posição *Low*)



Fonte: Autoria Própria, 2018

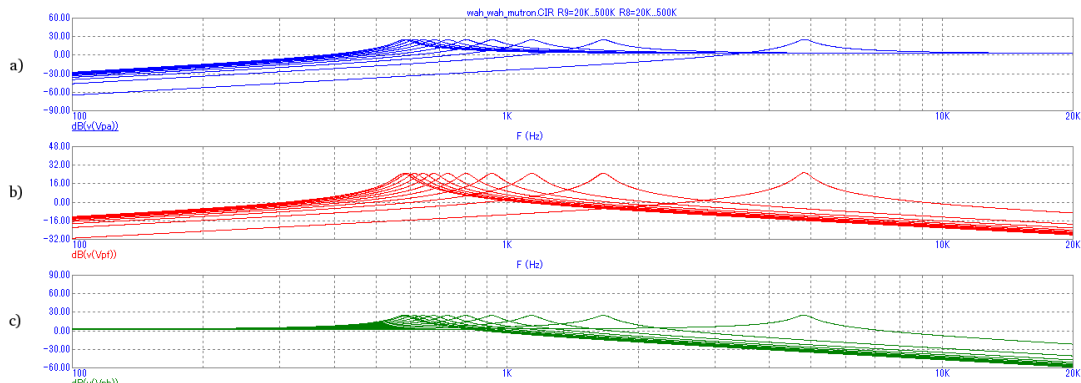
Os resultados dos testes mostraram valores de frequência de corte máximo e mínimo de 2200 Hz e 260 Hz quando a chave *Range* estiver na posição *Low* e 4900 Hz e 480 Hz quando a chave *Range* estiver na posição *High*.

Quando a envoltória tiver seu valor máximo (valor 1) a frequência de corte consequentemente atingirá seu maior valor, como mostrado na equação 29.

$$f_c = f_{min} + (f_{max} - f_{min}) * 1 = f_{min} + f_{max} - f_{min} = f_{max} \quad (29)$$

Da mesma forma, quando a envoltória tiver seu valor mínimo (valor 0) a frequência de corte terá seu menor valor, como mostrado na equação 30.

Figura 19 – Resposta a) passa-alta, b) passa-banda e c) passa-baixa do filtro para diferentes valores de frequência de corte (parâmetro *Range* na posição *High*)



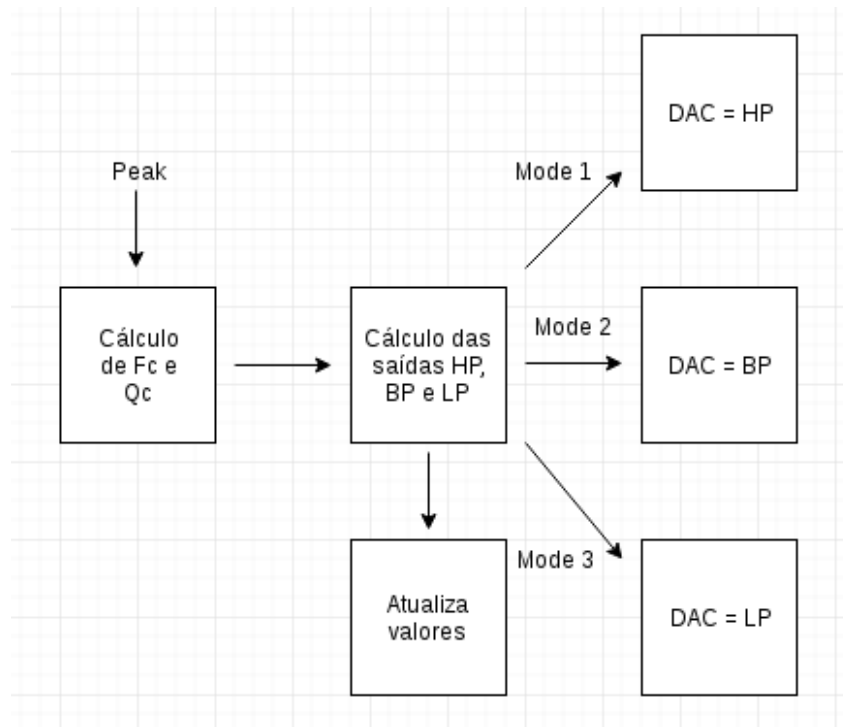
Fonte: Autoria Própria, 2018

$$f_c = f_{min} + (f_{max} - f_{min}) * 0 = f_{min} \tag{30}$$

3.7.2 Implementação do Filtro

O Filtro de Variáveis de Estado é implementado no microprocessador através de um algoritmo que implementa o diagrama de blocos da figura 20.

Figura 20 – Diagrama de blocos do Filtro de Variáveis de Estado



Fonte: Autoria Própria, 2018

Os parâmetros F_C e Q_C do filtro digital são calculadas através das equações 14 e 15. O valor de f_c é calculado anteriormente no algoritmo através das equações 27 e 28. O valor de Q_C é determinado pela variável de regulagem *Peak*. Utilizando a equação 7 são calculados os valores máximo e mínimo do fator Q do Mu-tron III variando-se o valor do resistor R9 da figura 8 entre 0 e 150k Ω , como mostrado nas equações 31 e 32.

$$Q_{min} = \frac{2(22k + (12k + 0))}{22k} = 3.0909 \quad (31)$$

$$Q_{max} = \frac{2(22k + (12k + 150k))}{22k} = 16.7272 \quad (32)$$

As respostas passa-alta, passa-banda e passa-baixa do Filtro de Variáveis de Estado são calculadas através das equações de diferença do filtro mostradas nas equações 16, 17 e 18, com $x[n]$ sendo o sinal de entrada, $y_{HP}[n - 1]$, $y_{BP}[n - 1]$, e $y_{LP}[n - 1]$ sendo as respostas das leituras anteriores e F_C e Q_C os parâmetros calculados anteriormente no algoritmo. As variáveis contendo os valores de resposta das leituras anteriores são então atualizadas para o cálculo do próximo ciclo.

Por fim, para cada um dos valores do parâmetro de regulagem *Mode*, a variável de saída recebe o valor de uma das respostas do filtro, e seu valor é modificado para variar no intervalo de valores entre 0.0 e 4095.0. Após, o valor de saída é registrado na DAC do microprocessador.

3.8 Pós-Processamento

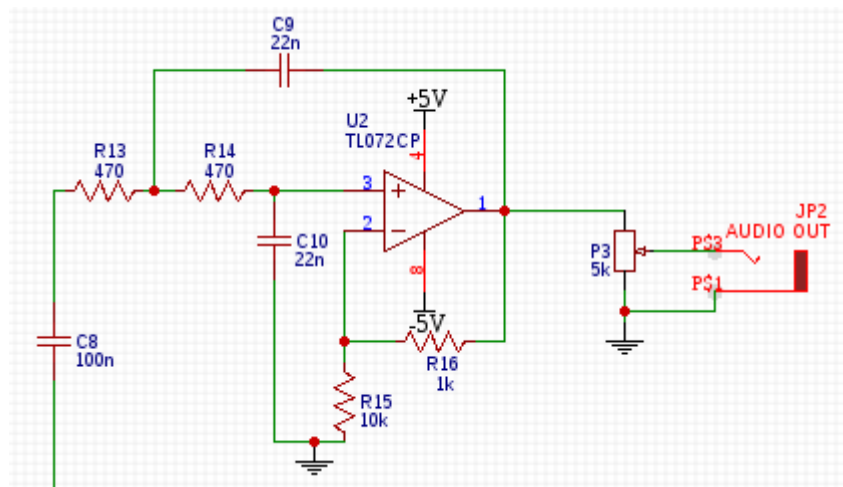
O sinal na saída do microprocessador passa por uma etapa de pós-processamento antes de ser lido na saída da placa. O circuito de pós-processamento é mostrado na figura 21.

O capacitor C_8 é utilizado para remover o nível DC do sinal de saída da DAC e deixar o sinal com tensão média de 0 V. Após, é implementado um filtro passa-baixas de topologia *Butterworth* de segunda ordem para reconstrução do sinal e remoção de ruídos gerados pela DAC. O ganho do filtro, determinado pelos resistores R_{15} e R_{16} é de aproximadamente 1.1 com os valores de resistores utilizados no projeto. A frequência de corte do filtro com os valores utilizados é calculada na equação 33.

$$f_c = \frac{1}{2\pi\sqrt{R_{13}R_{14}C_9C_{10}}} = \frac{1}{2\pi * 22n * 470} = 15392.1608Hz \quad (33)$$

Por fim, o ganho de saída do bloco de pós-processamento é controlado pelo potenciômetro P_3 , que implementa um divisor de tensão que determina o valor de tensão no *jack* de saída que é posteriormente lido pelo dispositivo de áudio.

Figura 21 – Circuito de saída da placa



Fonte: Autoria Própria, 2018

3.9 Testes e Validação do Método

A reprodução do efeito de áudio é avaliada realizando medições, através de simulações, no circuito original e no modelo implementado. As medições são realizadas para cada um dos blocos nos quais os sistemas foram divididos, em diferentes configurações dos parâmetros de regulação tanto na simulação do circuito original quanto na do circuito implementado. Por fim, são realizadas medições no circuito físico desenvolvido para validar o modelo desenvolvido.

3.9.1 Bloco de Entrada

O circuito do pedal Mu-tron III é implementado no software Multisim 14 de acordo com a figura 22.

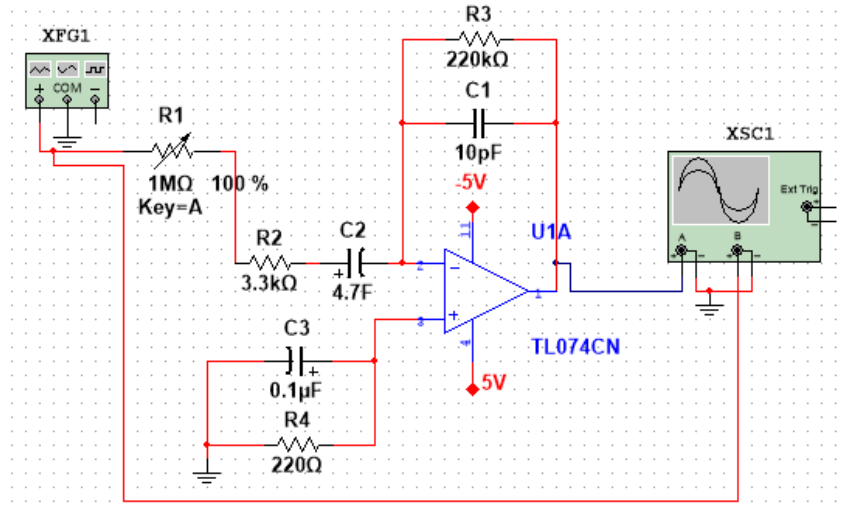
Na entrada do circuito é colocado um gerador de funções configurado para fornecer um sinal senoidal de amplitude de 10 mV e frequência de 1 kHz. Então são medidos os valores de entrada e saída do circuito utilizando o osciloscópio do Multisim.

O circuito de entrada utilizado no projeto é implementado da mesma forma no software Multisim, como mostrado na figura 23. Um gerador de funções configurado com os mesmos parâmetros utilizados no teste do circuito original. São medidos então os valores de entrada e saída utilizando a ferramenta de osciloscópio.

É realizada a medição da resposta em frequência dos dois circuitos através da simulação *AC Sweep* do Multisim. É medida a resposta para os valores de frequência entre 1 Hz e 30 kHz.

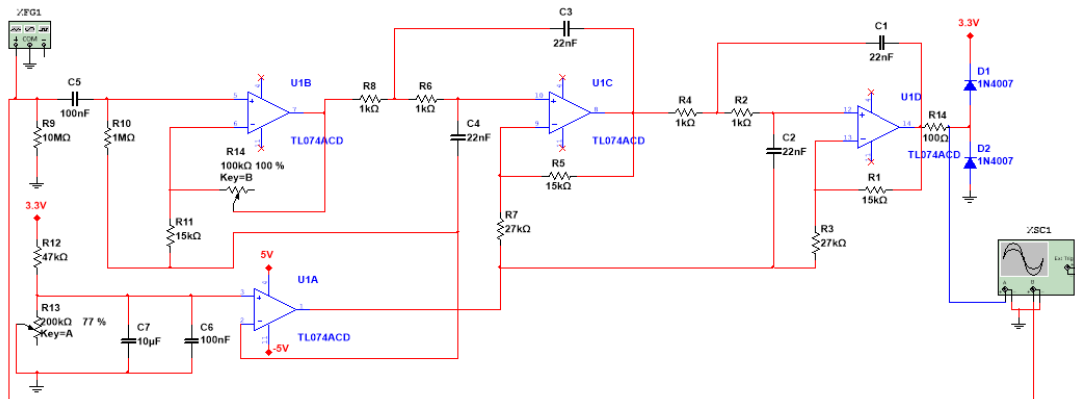
Por fim, a implementação é validada realizando medidas no circuito físico. Um gerador de funções configurado para fornecer um sinal senoidal de amplitude de 160 mV

Figura 22 – Bloco de Entrada do Mu-tron III em simulação no Multisim



Fonte: Autoria Própria, 2018

Figura 23 – Bloco de Entrada do circuito implementado em simulação no Multisim



Fonte: Autoria Própria, 2018

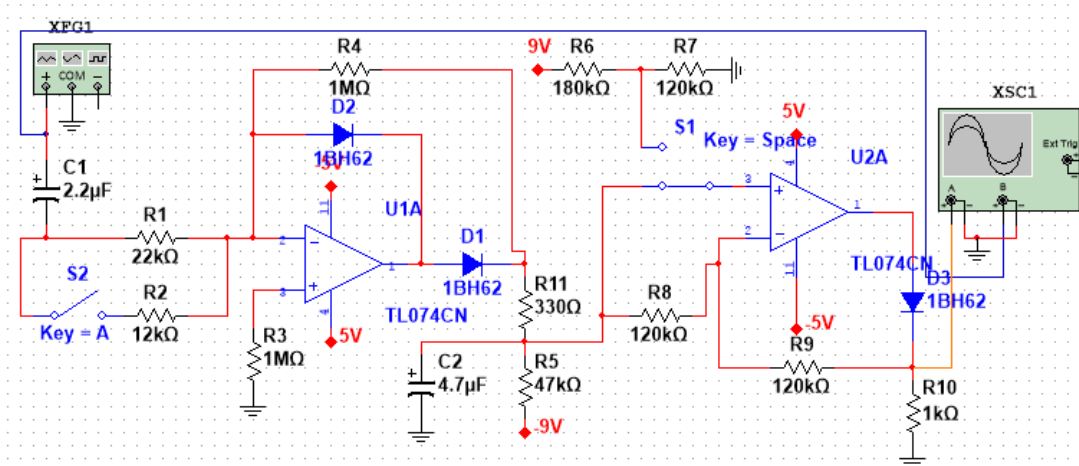
e frequência de 1 kHz é ligado na entrada do circuito. A tensão na entrada do ADC é então medida para diferentes valores de ganho, e os resultados são comparados com os resultados das simulações.

3.9.2 Detector de Envoltória

O bloco detector de envoltória do Mu-tron III é implementado no Multisim de acordo com a figura 24.

Um gerador de funções é conectado à entrada do circuito, utilizando como parâmetros amplitude de 350 mV e frequência de 5 Hz. Os valores de entrada e saída do circuito são então medidos utilizando o osciloscópio. São realizadas medições utilizando as configurações *Up* e *Down* do parâmetro de regulação *Drive*.

Figura 24 – Detector de envoltória do Mu-tron III em simulação no Multisim



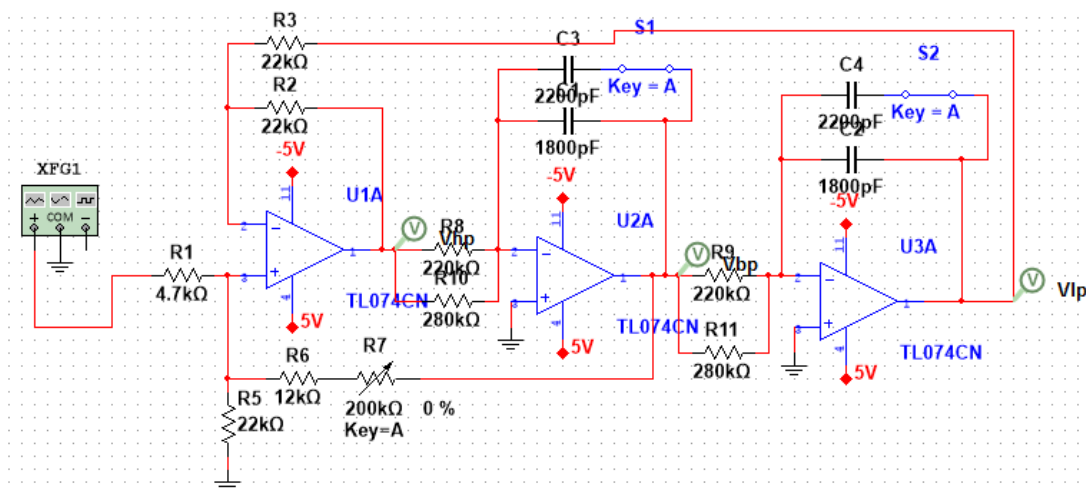
Fonte: Autoria Própria, 2018

Como o detector de envoltória foi implementado de forma digital no projeto, seu teste é realizado utilizando o software MATLAB® R2012b, onde é gerado um sinal equivalente ao sinal de entrada do gerador de funções do simulador, e seus valores de saída para ambas as configurações são registrados.

3.9.3 Filtro de Variáveis de Estado

O filtro de variáveis de estado do Mu-tron III é implementado no software Multisim como mostrado na figura 25.

Figura 25 – Filtro de Variáveis de Estado do Mu-tron III em simulação no Multisim



Fonte: Autoria Própria, 2018

São realizadas medidas das respostas em frequência para as saídas passa-alta, passa-banda e passa-baixa através da simulação *AC Sweep* do Multisim. São realizadas

medidas nas três saídas do circuito para os valores máximo e mínimo de frequência de corte, nas duas configurações do parâmetro de regulação *Range* e para os valores máximo e mínimo do parâmetro de regulação *Peak*. É medida a resposta para os valores de frequência entre 1 Hz e 30 kHz.

O filtro de variáveis de estado implementado no projeto é testado inicialmente através do software MATLAB®, onde as funções de transferência vistas na teoria são calculadas com os parâmetros do filtro original. São então determinadas as respostas em frequência para as três saídas do filtro, para os valores máximo e mínimo de frequência de corte, nas duas configurações do parâmetro de regulação *Range* e para os valores máximo e mínimo do parâmetro de regulação *Peak*.

Por fim, são realizadas medidas no circuito físico implementado. É injetado ruído branco na entrada do circuito, e são medidas as respostas em frequência para as três saídas do filtro através da função FFT do osciloscópio. São realizadas medidas para valores 0 e 1 do parâmetro *Range*, e para valores máximo e mínimo do parâmetro *Peak*.

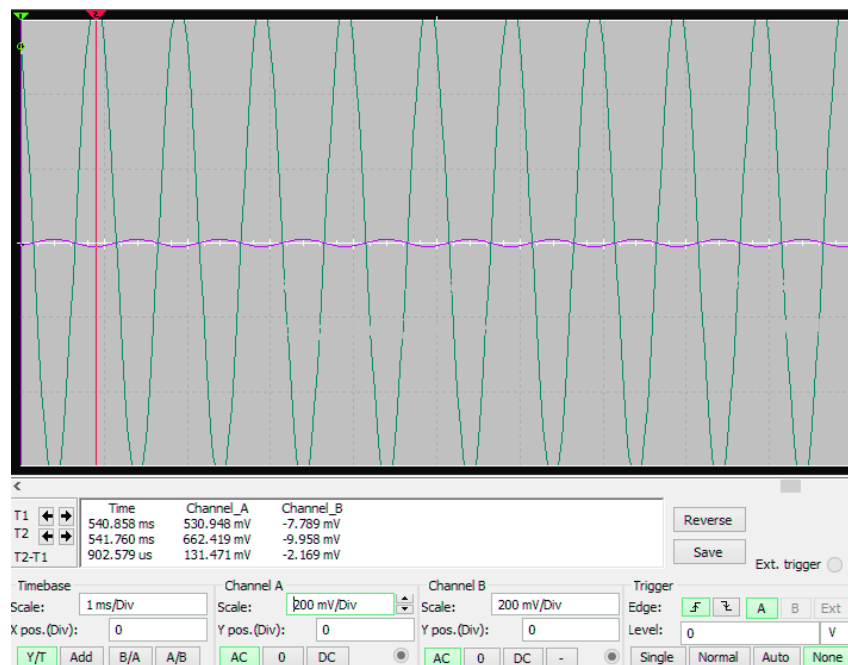
4 Resultados e Discussões

4.1 Bloco de Entrada

4.1.1 Mu-tron III - Simulação

Os valores de entrada e saída na simulação do circuito do Mu-tron III para o valor máximo de ganho são mostrados na figura 26.

Figura 26 – Sinal de entrada (em magenta) e sinal de saída (em turquesa)



Fonte: Autoria Própria, 2018

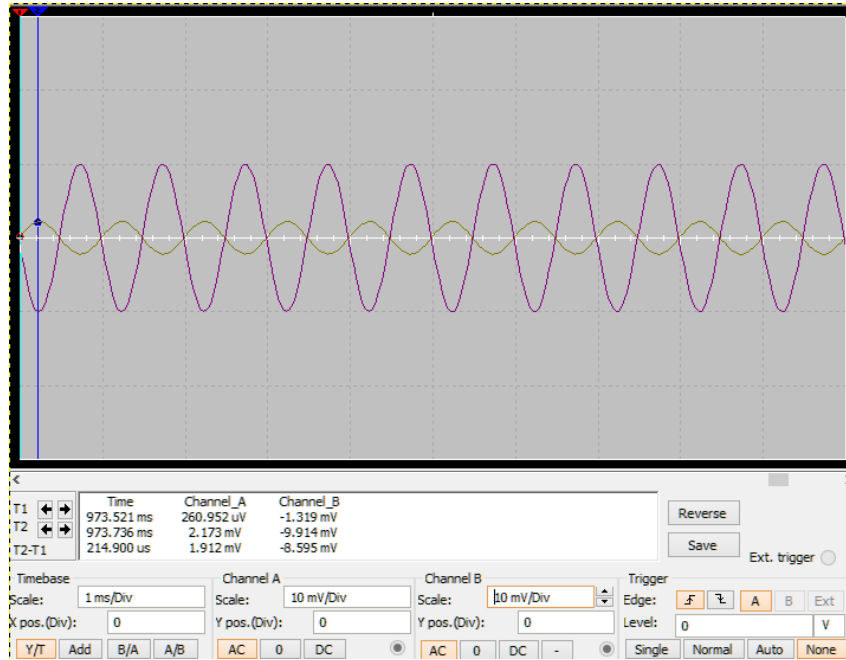
Como mostrado na figura 26 a tensão de saída é de 662.419 mV para um valor de entrada de -9.958 mV. Através desses valores, o ganho máximo é calculado através da equação 34.

$$G_{max} = \frac{662.419}{-9.958} = -66.52 \quad (34)$$

Os valores de entrada e saída para o valor mínimo de ganho são mostrados na figura 27.

Para um valor de entrada de -9.914 mV, é medido um valor de saída de 2.173 mV. O ganho mínimo do bloco de entrada é então calculado através da equação 35.

Figura 27 – Sinal de entrada (em magenta) e sinal de saída (em amarelo)



Fonte: Autoria Própria, 2018

$$G_{min} = \frac{2.173}{-9.914} = -0.219 \quad (35)$$

Pode-se atestar que o ganho do sinal é negativo devido à configuração inversora do circuito de entrada do Mu-tron III.

A resposta em frequência do circuito de entrada medida através da simulação *AC Sweep* é mostrada na figura 28. Pode-se notar a atenuação nos valores de frequência acima de 10 kHz.

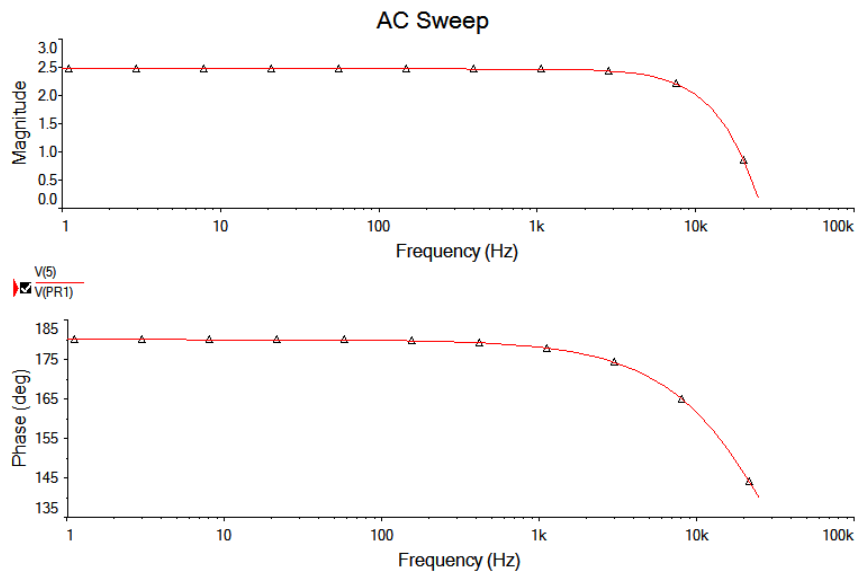
4.1.2 Circuito Implementado - Simulação

Os valores de entrada e saída para o valor máximo de ganho são mostrados na figura 29.

Nota-se um nível DC de 1.623 V no sinal de saída, com seu nível AC variando com uma amplitude de 184 mV. O sinal de entrada varia com uma amplitude de 9.968 mV. Com isso, o ganho de entrada é determinado pela equação 36.

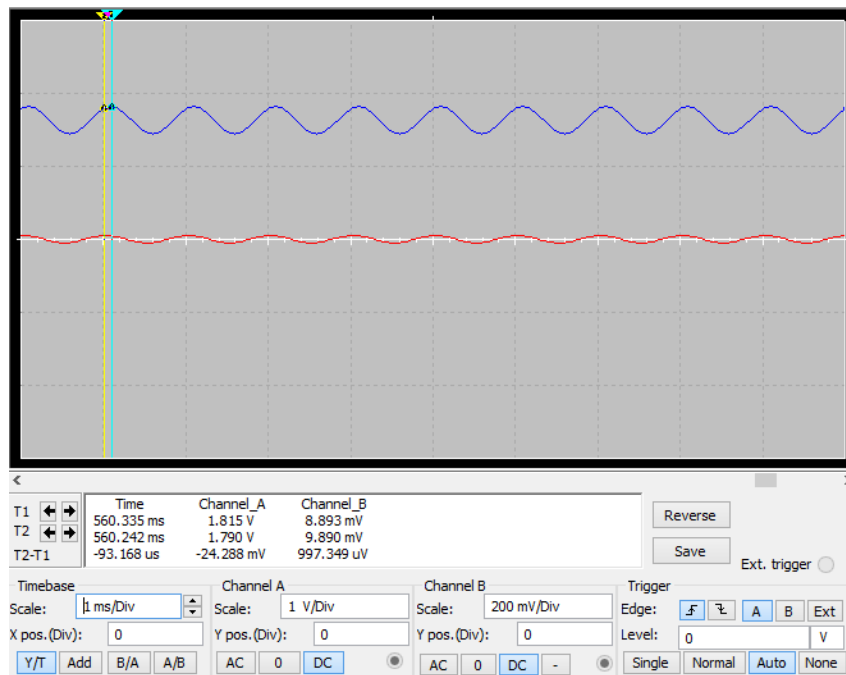
$$G_{max} = \frac{184}{9.968} = 18.46 \quad (36)$$

Para um ganho mínimo, os valores de entrada e saída são mostrados na figura 27.

Figura 28 – Resposta em frequência do bloco de entrada do *Mu-tron III*

Fonte: Autoria Própria, 2018

Figura 29 – Sinal de entrada (em vermelho) e sinal de saída (em azul)

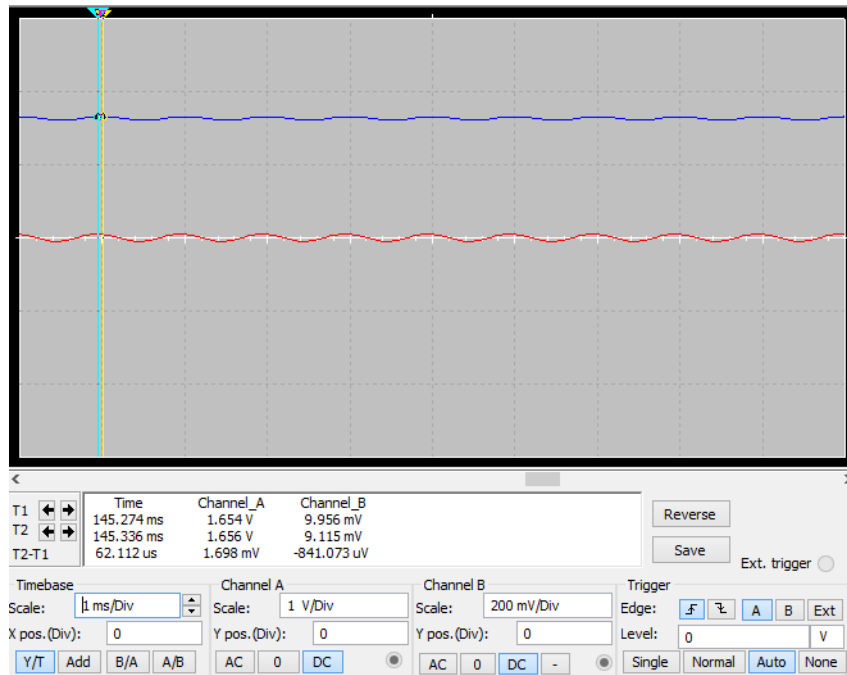


Fonte: Autoria Própria, 2018

O nível DC apresentado no sinal de saída é de 1.637 V, com seu nível AC variando com uma amplitude de 24 mV. O sinal de entrada varia com uma amplitude de 9.968 mV. O ganho mínimo de entrada é determinado pela equação 37.

$$G_{min} = \frac{24}{9.968} = 2.407 \quad (37)$$

Figura 30 – Sinal de entrada (em vermelho) e sinal de saída (em azul)

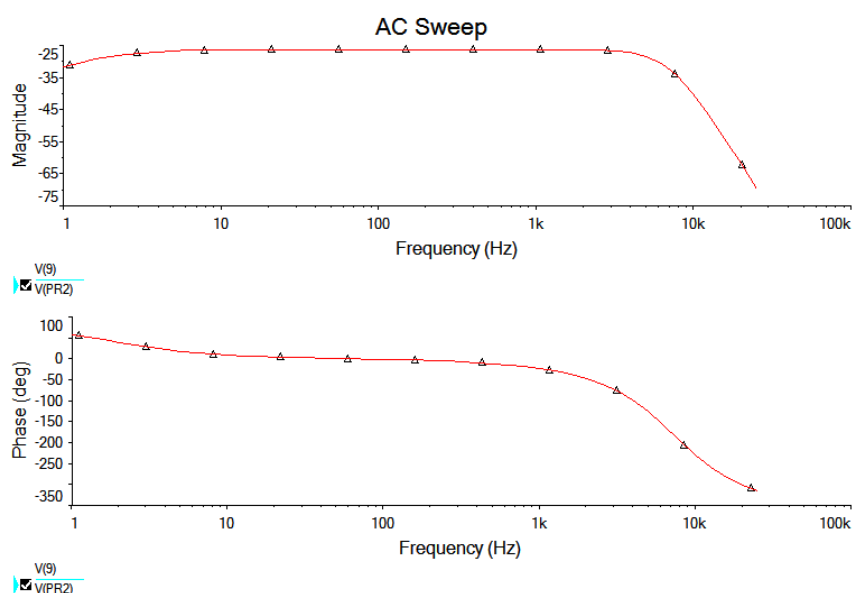


Fonte: Aatoria Própria, 2018

Ao contrário do circuito original, o circuito implementado possui ganho positivo e sempre maior que 1, tendo em vista que foi utilizada uma configuração não-inversora para adição de ganho DC necessário para o sinal ser adquirido pelo ADC do microprocessador.

A resposta em frequência do circuito, medida através da simulação *AC Sweep* do software Multisim é mostrada na figura 31.

Figura 31 – Resposta em frequência do bloco de entrada do circuito implementado



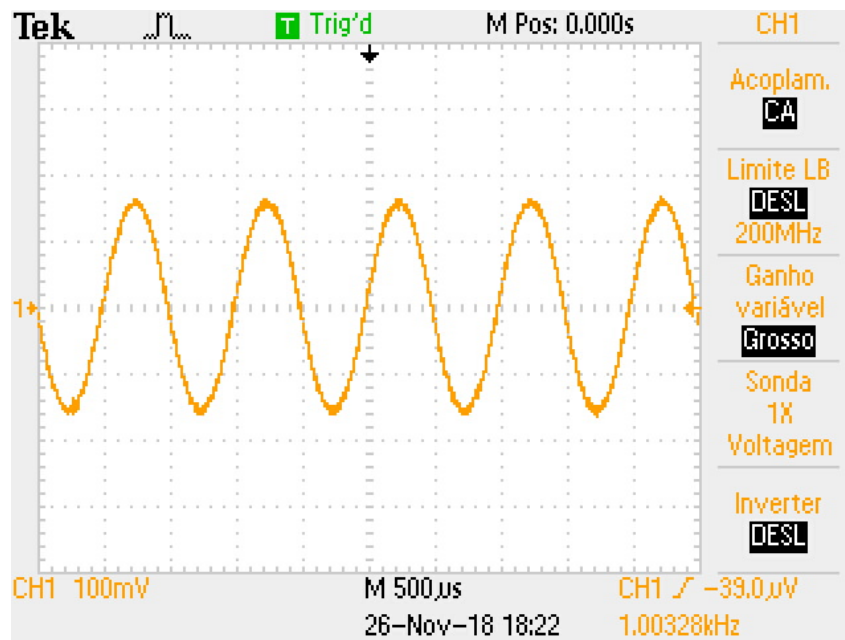
Fonte: Aatoria Própria, 2018

Assim como no circuito original, também é vista uma atenuação nos valores de frequência acima de 10 kHz.

4.1.3 Circuito Implementado

O sinal conectado à entrada do circuito implementado é mostrado na figura 32. Sua amplitude é de 170 mV e sua frequência é de 1 kHz.

Figura 32 – Sinal de entrada do circuito implementado



Fonte: Autoria Própria, 2018

O valor de saída do bloco com alto ganho é mostrado na figura 33. Devido ao sinal na entrada do ADC estar limitado à faixa de valores entre 0 V e 3.3 V o ganho de entrada não pode ser medido em seu valor máximo.

Como pode ser visto na figura 33, o nível DC do sinal é de 1.71 V, e seu nível AC possui amplitude de 2.52 V pico a pico, ou 1.26 V de pico. O ganho é determinado através da equação 38.

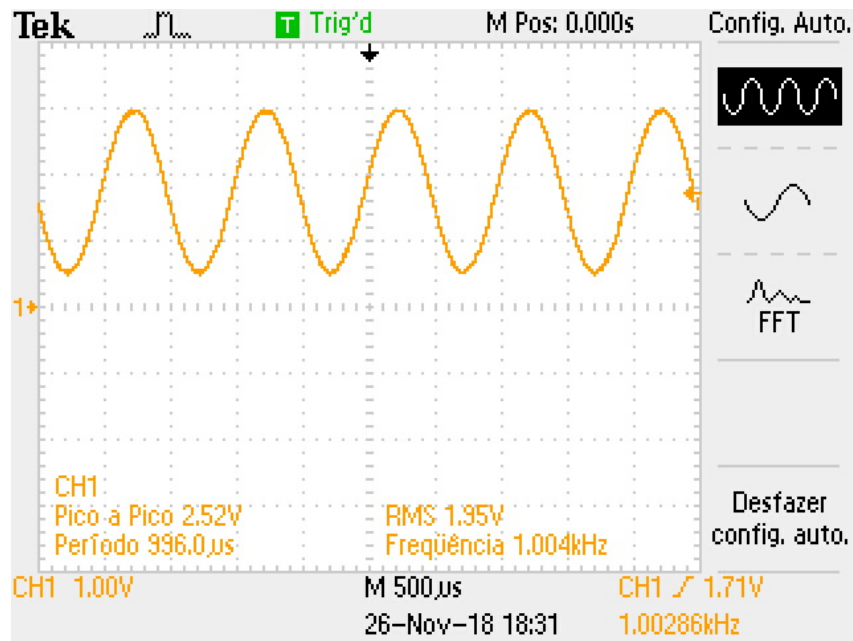
$$G_{max} = \frac{1.26}{0.170} = 7.41 \quad (38)$$

O sinal de saída para o valor mínimo de ganho é mostrado na figura 34.

O nível DC do sinal é de 1.71 V, e seu nível AC é de 840 mV pico a pico, ou 420 mV de pico. O ganho mínimo é calculado na equação 39

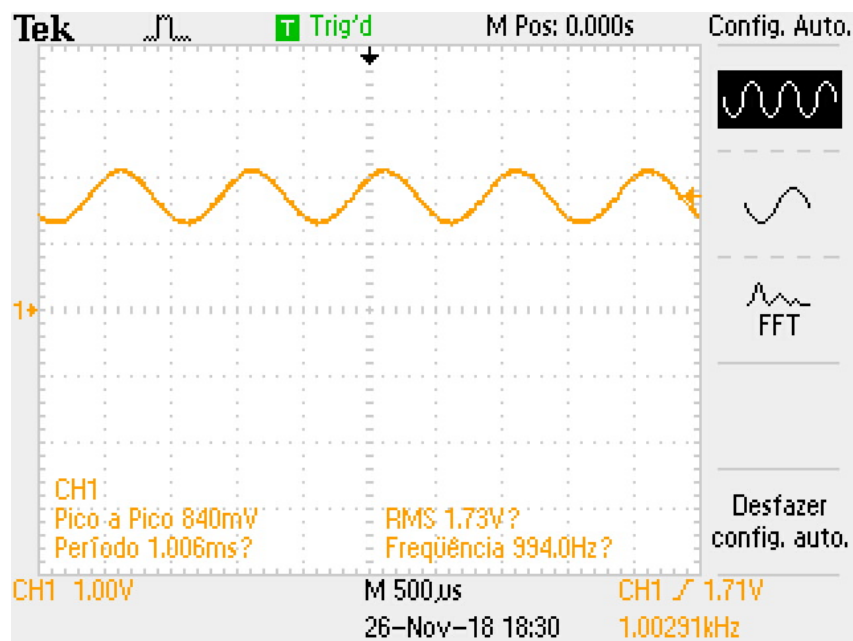
$$G_{min} = \frac{420}{170} = 2.47 \quad (39)$$

Figura 33 – Valor de saída do bloco de entrada com alto ganho



Fonte: Autoria Própria, 2018

Figura 34 – Valor de saída do bloco de entrada ganho mínimo



Fonte: Autoria Própria, 2018

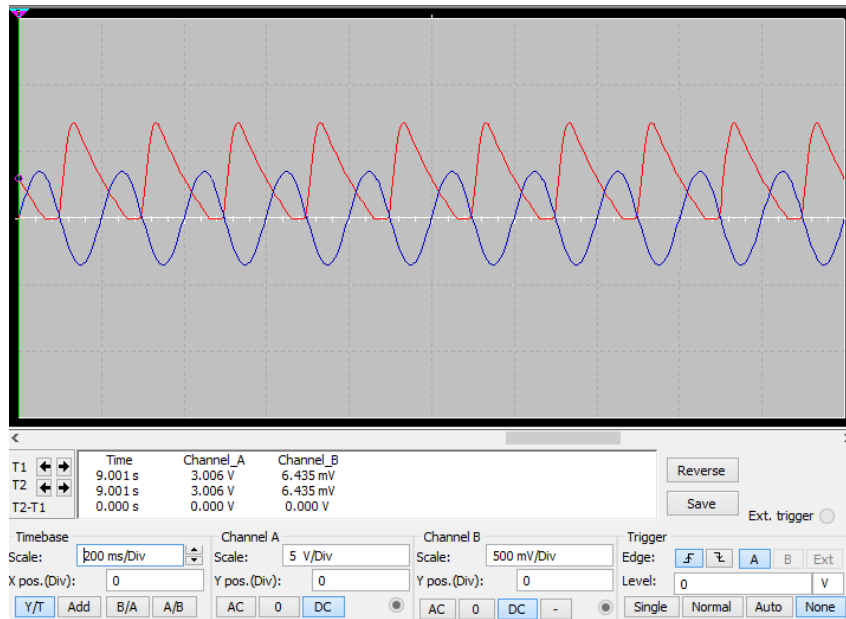
Comparando com os valores encontrados em simulação a implementação física apresentou valores semelhantes do ganho DC e do ganho mínimo AC.

4.2 Detector de Envoltória

4.2.1 Mu-tron III - Simulação

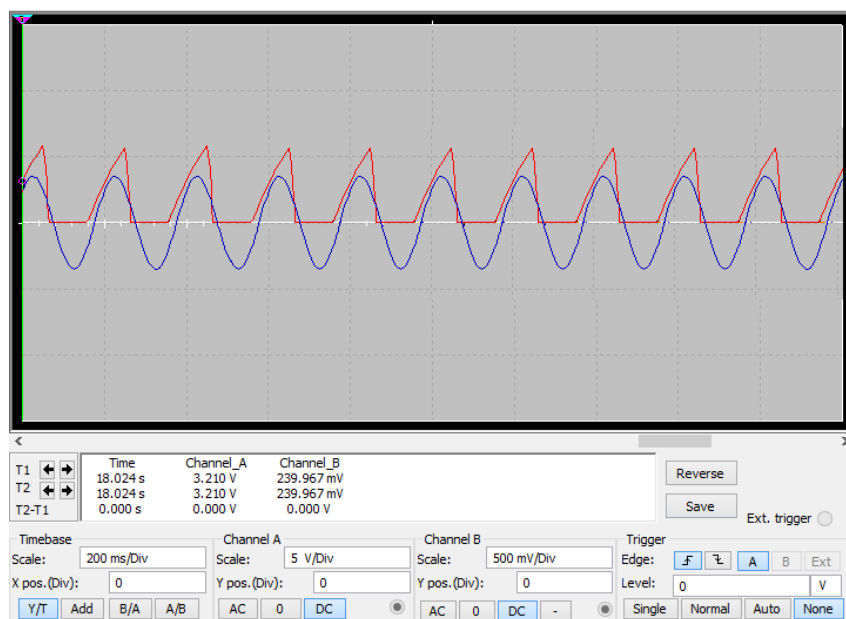
Os valores de tensão de saída do detector de envoltória do Mu-tron III para as duas configurações do parâmetro *Drive* são mostrados nas figuras 35 e 36.

Figura 35 – Saída do detector de envoltória para *Drive* na posição *Up*



Fonte: Autoria Própria, 2018

Figura 36 – Saída do detector de envoltória para *Drive* na posição *Down*



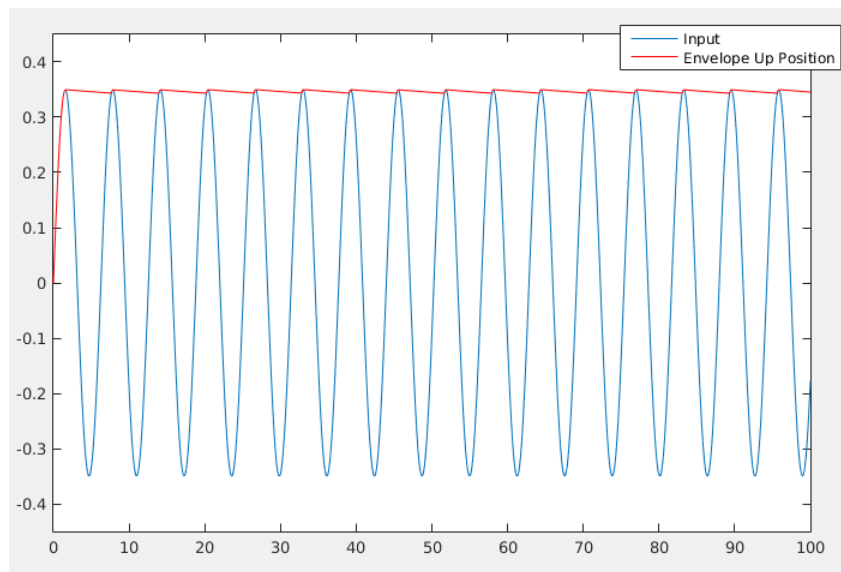
Fonte: Autoria Própria, 2018

Pode-se atestar a inversão de fase da tensão de saída em relação à tensão de entrada devido à configuração inversora do circuito detector de envoltória.

4.2.2 Software implementado - Simulação no MATLAB®

A rotina para detecção de envoltória utilizada em software é implementada no MATLAB® e seus resultados para ambas as configurações do parâmetro *Drive* são mostrados nas figuras 37 e 38.

Figura 37 – Saída do detector de envoltória para *Drive* na posição *Up*



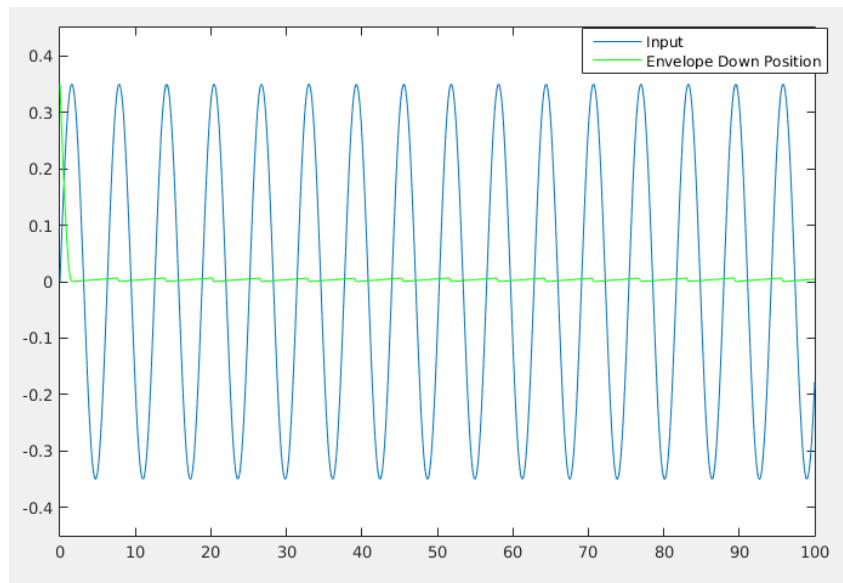
Fonte: Autoria Própria, 2018

Pode-se atestar que, apesar da semelhança no formato dos sinais de saída da simulação do circuito e em MATLAB®, há uma significativa diferença no ganho de saída entre os testes realizados. Contudo, essa diferença não é prejudicial à implementação do projeto, tendo em vista que o projeto implementado não utiliza de valores absolutos da envoltória do sinal, e sim do seu valor normalizado. Com isso, o fator de maior relevância é a semelhança entre as formas de onda.

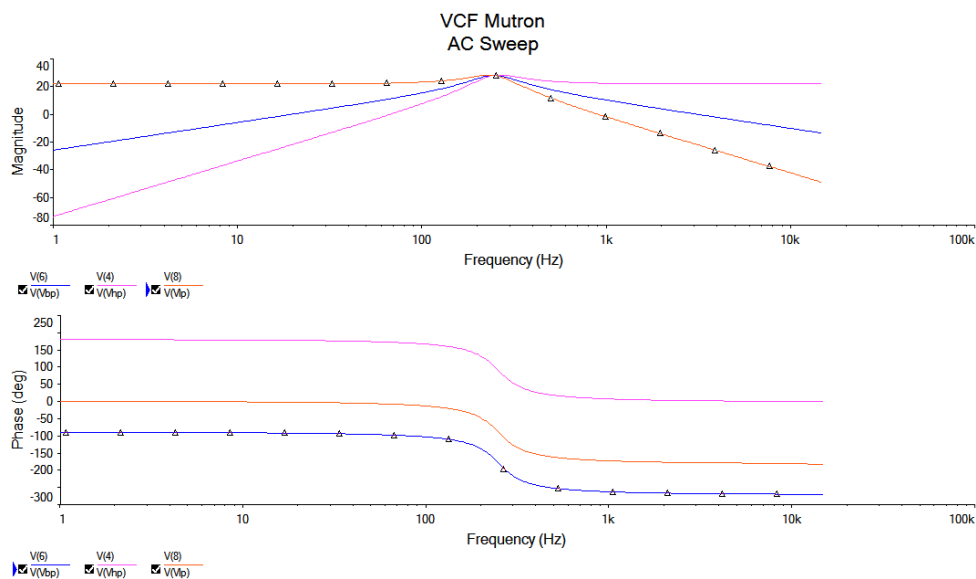
4.3 Filtro de Variáveis de Estado

4.3.1 Mu-tron III - Simulação

As respostas em frequência para as três saídas do filtro de variáveis de estado são determinadas através da simulação *AC Sweep* do Multisim. Inicialmente a simulação é realizada com um valor de 280 k Ω nos LDRs. Os valores medidos são mostrados nas figuras 39 e 40 para o parâmetro *Range* em nível alto.

Figura 38 – Saída do detector de envoltória para *Drive* na posição *Down*

Fonte: Autoria Própria, 2018

Figura 39 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak Mínimo*

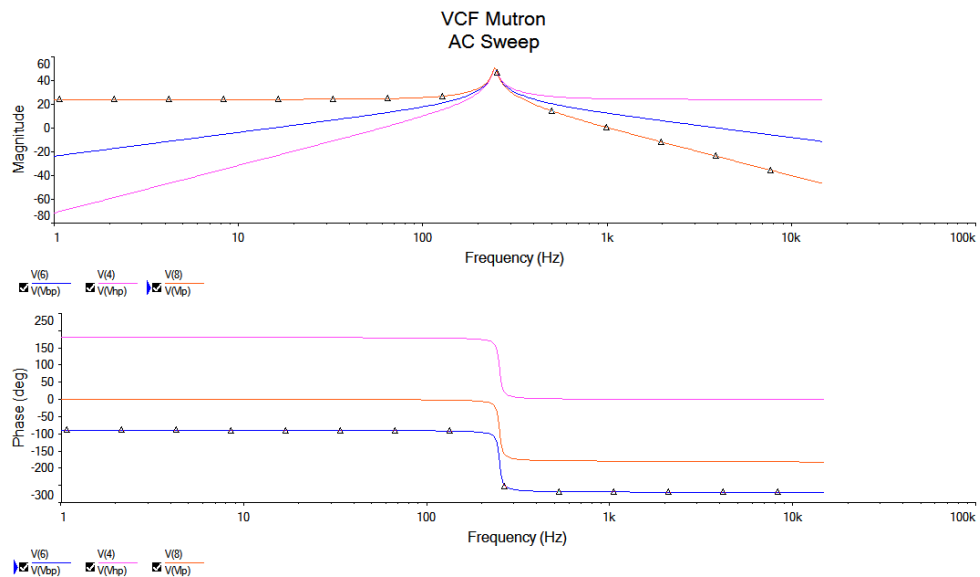
Fonte: Autoria Própria, 2018

Alterando o parâmetro *Range* para nível baixo, as respostas medidas são mostradas nas figuras 41 e 42.

O valor dos LDRs é então modificado para 65 k Ω . A resposta em frequência medida é mostrada nas figuras 43 e 44 para o parâmetro *Range* em nível baixo e nas figuras 45 e 46 para o parâmetro *Range* em nível alto.

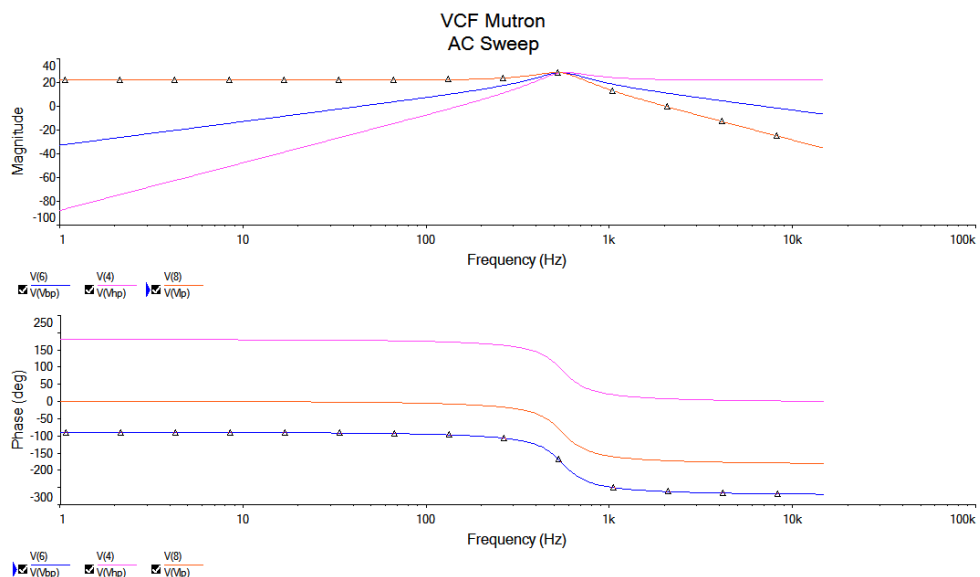
Pode-se atestar que a frequência de corte medida para o valor dos LDRs de 280

Figura 40 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak* Máximo



Fonte: Autoria Própria, 2018

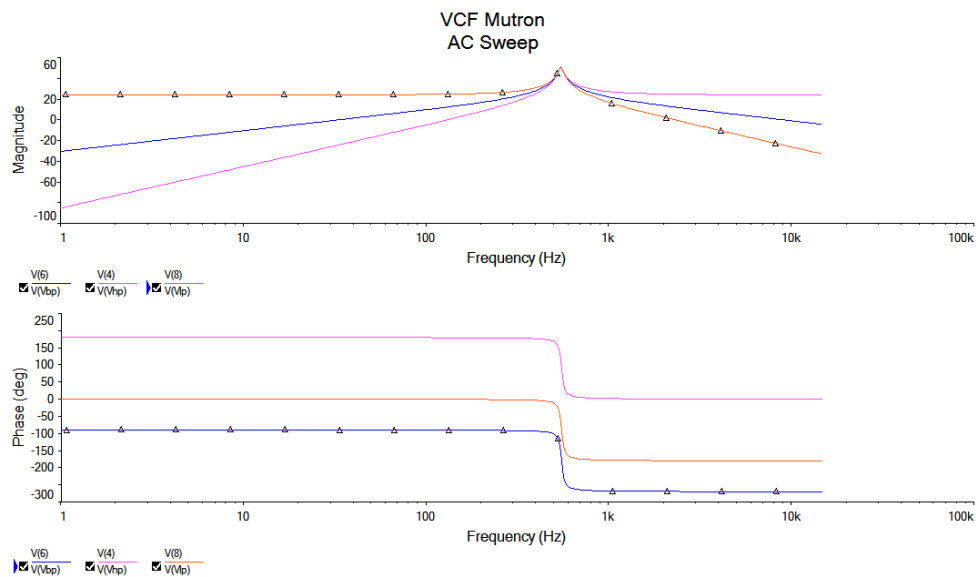
Figura 41 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak* Mínimo



Fonte: Autoria Própria, 2018

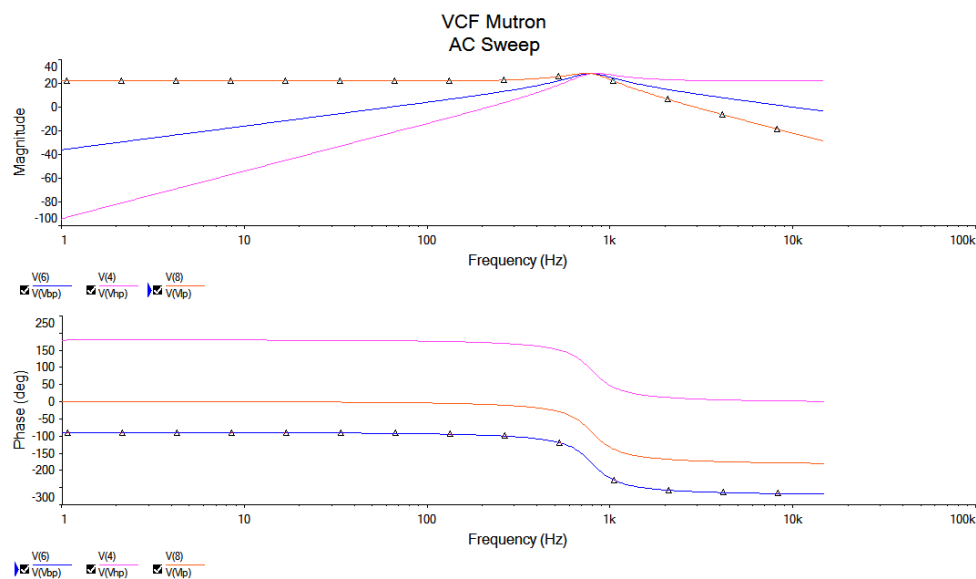
$k\Omega$ variou entre aproximadamente 380 Hz e 800 Hz, para cada valor do parâmetro *Range*. Já para o valor dos LDRs de 65 k Ω a frequência de corte medida foi de aproximadamente 800 Hz e 1700 Hz, para cada valor do parâmetro *Range*.

Figura 42 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak* Máximo



Fonte: Autoria Própria, 2018

Figura 43 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak* Mínimo

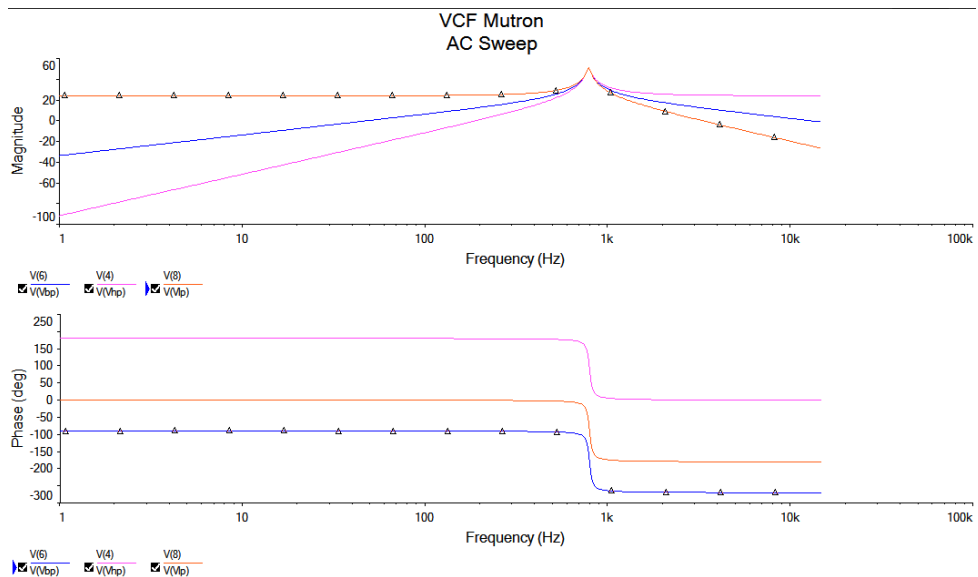


Fonte: Autoria Própria, 2018

4.3.2 Software Implementado - Simulação no MATLAB®

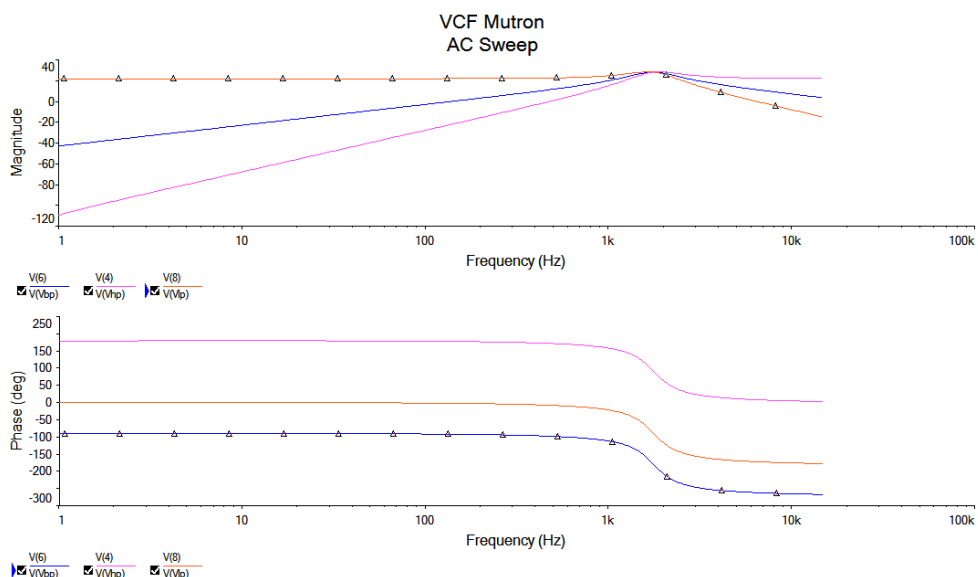
O software implementado realiza o cálculo do filtro de variáveis de estado para os valores de frequência de corte calculados através do bloco detector de envoltória. É elaborada uma rotina no MATLAB® que implementa as funções de transferência utilizadas para uma frequência de corte de 380 Hz. São utilizados valores de Q de 3 e 16, aproximadamente os valores reais para o parâmetro *Peak* do Mu-tron III. A resposta em

Figura 44 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak Máximo*



Fonte: Autoria Própria, 2018

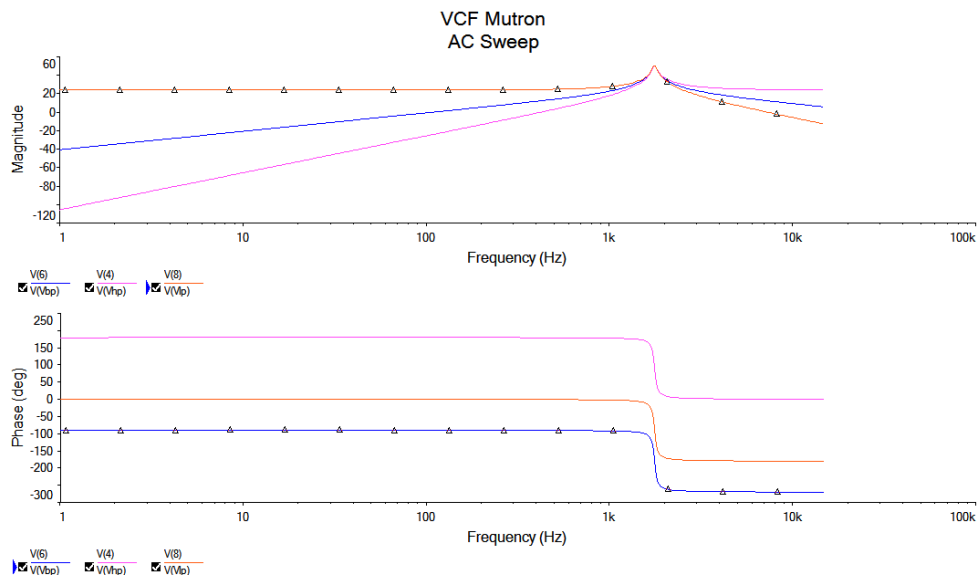
Figura 45 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak Mínimo*



Fonte: Autoria Própria, 2018

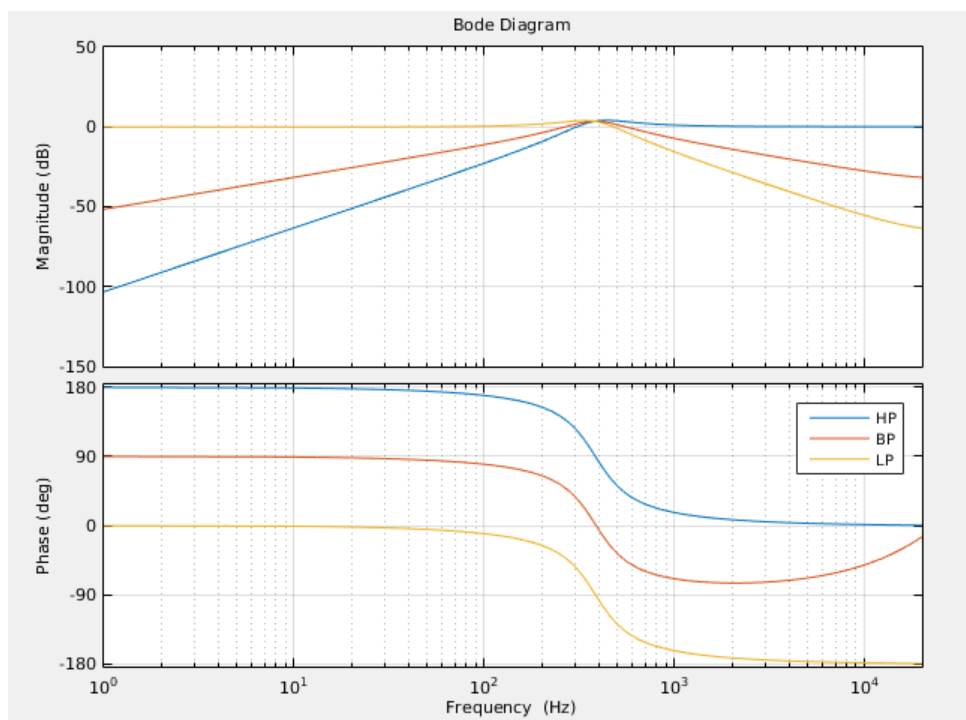
frequência dos filtros é mostrada nas figuras 47 e 48.

Figura 46 – Resposta em frequência para os filtros passa-alta (em magenta), passa-banda (em azul) e passa-baixa (em laranja) e *Peak* Máximo



Fonte: Aatoria Própria, 2018

Figura 47 – Resposta em frequência para os filtros passa-alta (em laranja), passa-banda (em vermelho) e passa-baixa (em azul) e *Peak* Mínimo

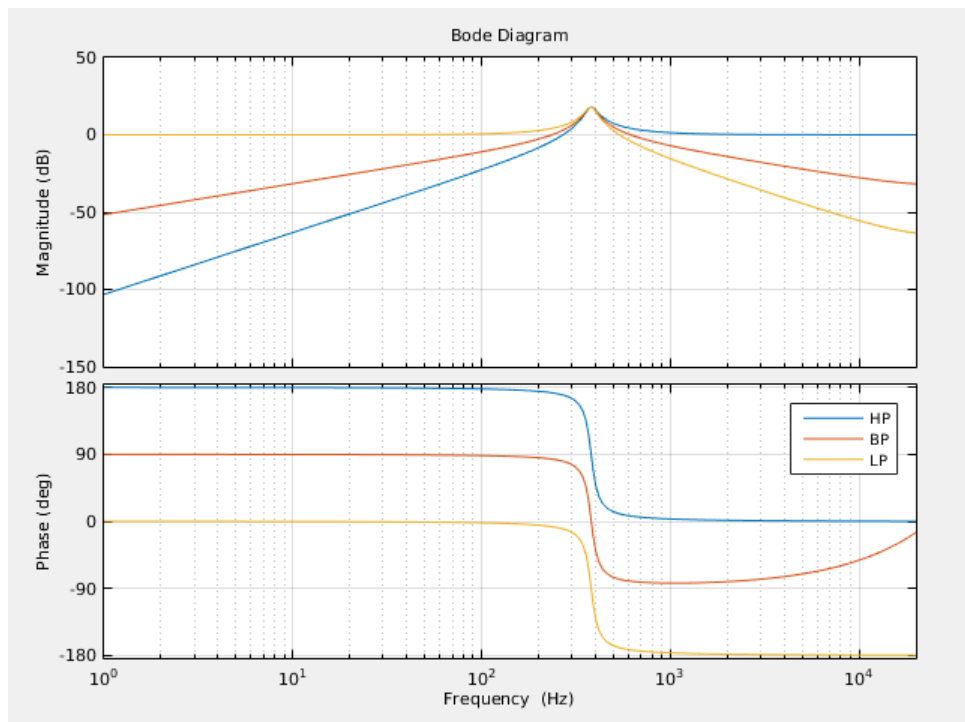


Fonte: Aatoria Própria, 2018

4.3.3 Software Implementado - Implementação Física

A validação do projeto é realizada através das medições no circuito físico implementado. Na entrada do circuito é injetado um sinal de ruído branco, que possui distri-

Figura 48 – Resposta em frequência para os filtros passa-alta (em laranja), passa-banda (em vermelho) e passa-baixa (em azul) e *Peak* Máximo



Fonte: Aatoria Própria, 2018

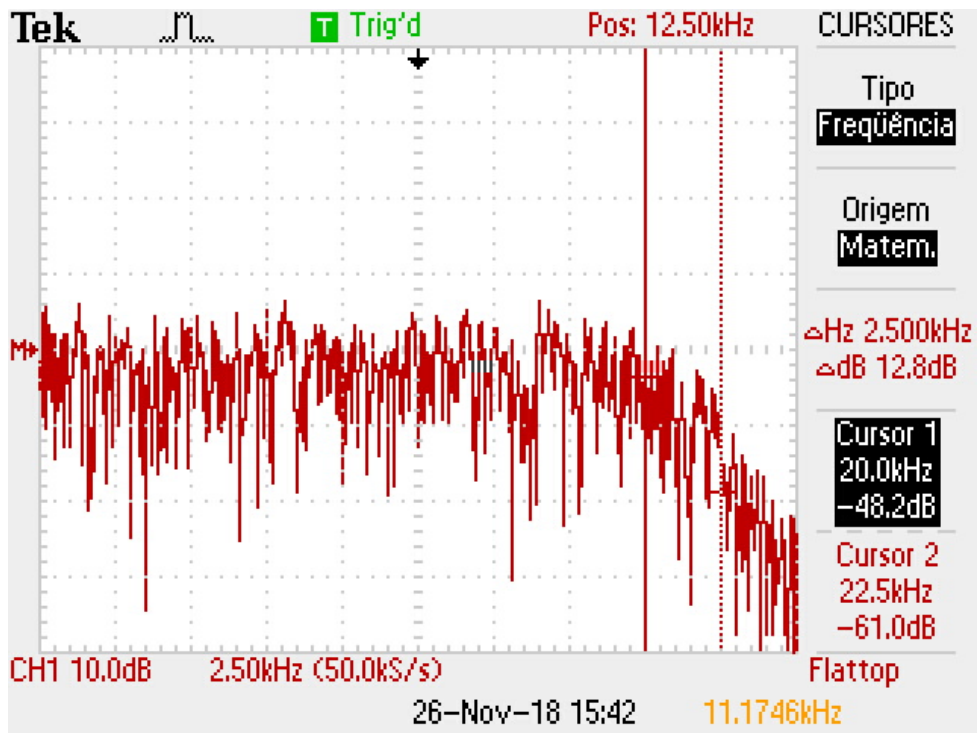
buição uniforme de valores em frequência. O sinal de entrada é medido no osciloscópio através da função FFT e é mostrado na figura 49.

Pode-se atestar que o sinal possui distribuição uniforme de frequências na faixa abaixo de 20 kHz, que é a faixa de interesse no projeto.

Utilizando inicialmente um valor baixo de ganho de entrada e com o parâmetro *Range* em nível baixo, são medidas as respostas em frequência para as saídas passa-alta, passa-banda e passa-baixa do filtro para valores alto e baixo do parâmetro *Peak*, como mostrado nas figuras 50, 51, 52, 53, 54 e 55.

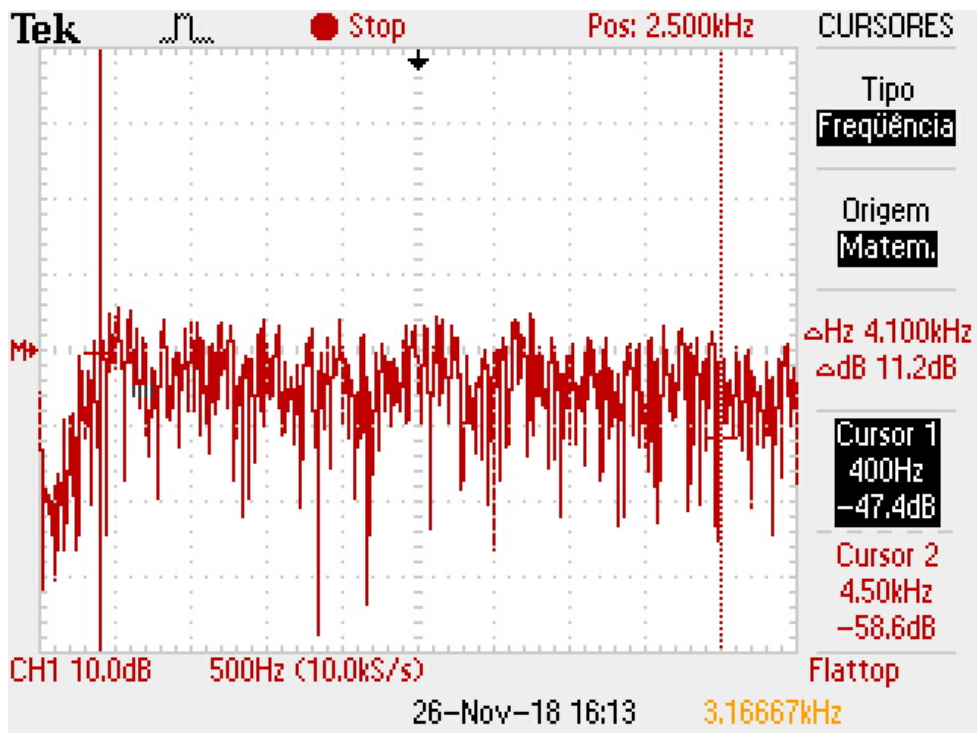
Nota-se que o valor da frequência de corte encontrado é de aproximadamente 400 Hz para as três saídas do filtro.

Figura 49 – Ruído branco utilizado nas medições



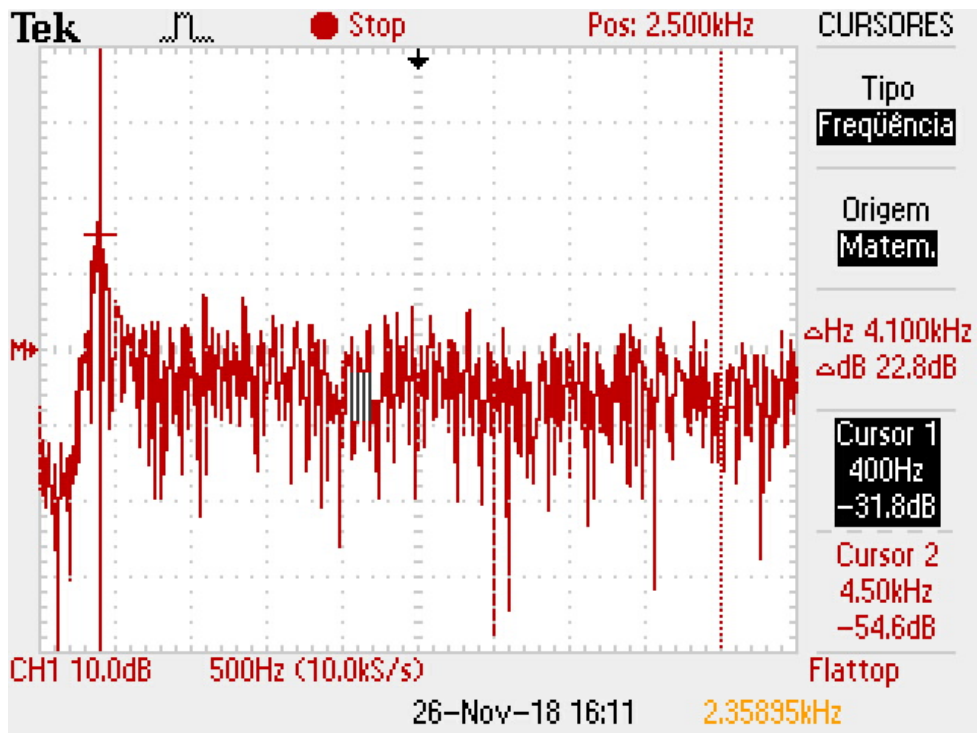
Fonte: Autoria Própria, 2018

Figura 50 – Saída passa-alta do filtro para *Peak* mínimo



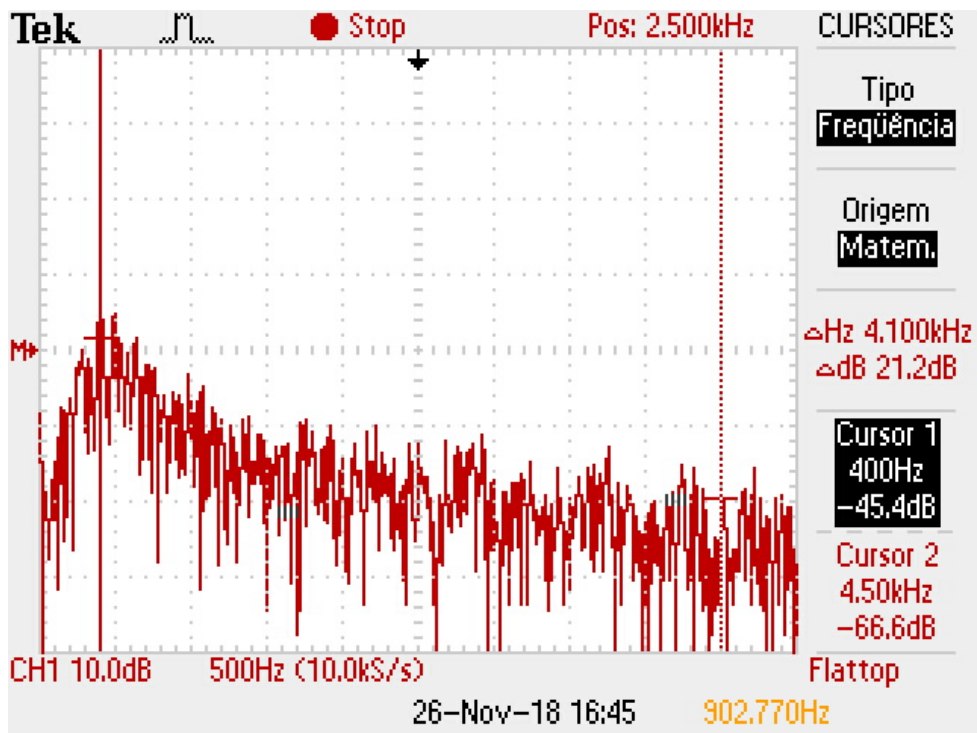
Fonte: Autoria Própria, 2018

Figura 51 – Saída passa-alta do filtro para *Peak* máximo



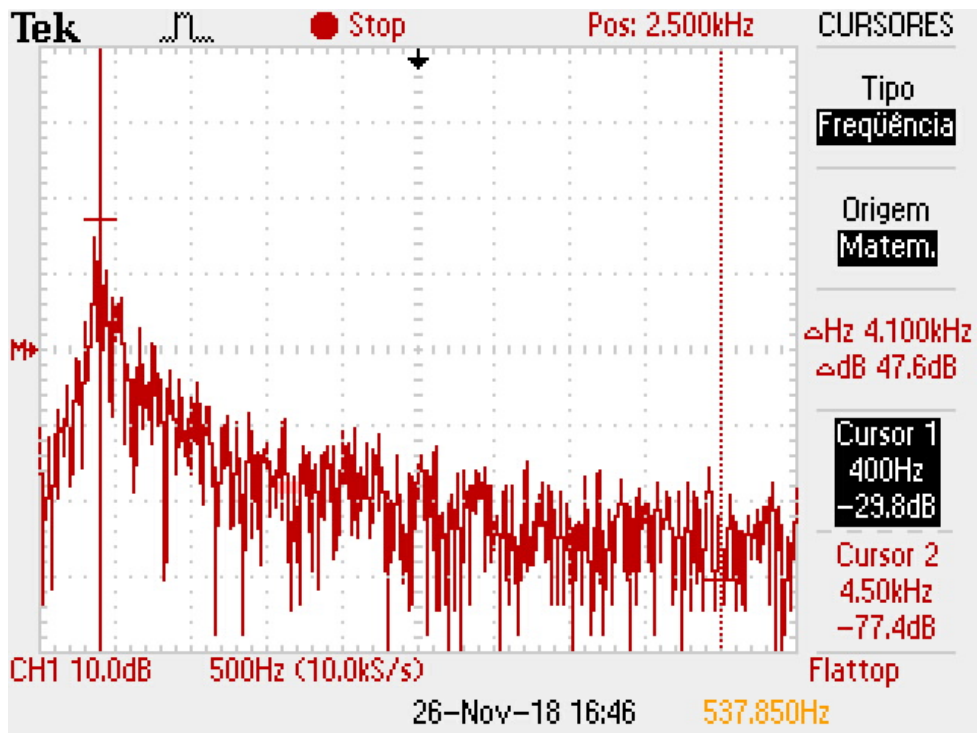
Fonte: Aatoria Própria, 2018

Figura 52 – Saída passa-banda do filtro para *Peak* mínimo



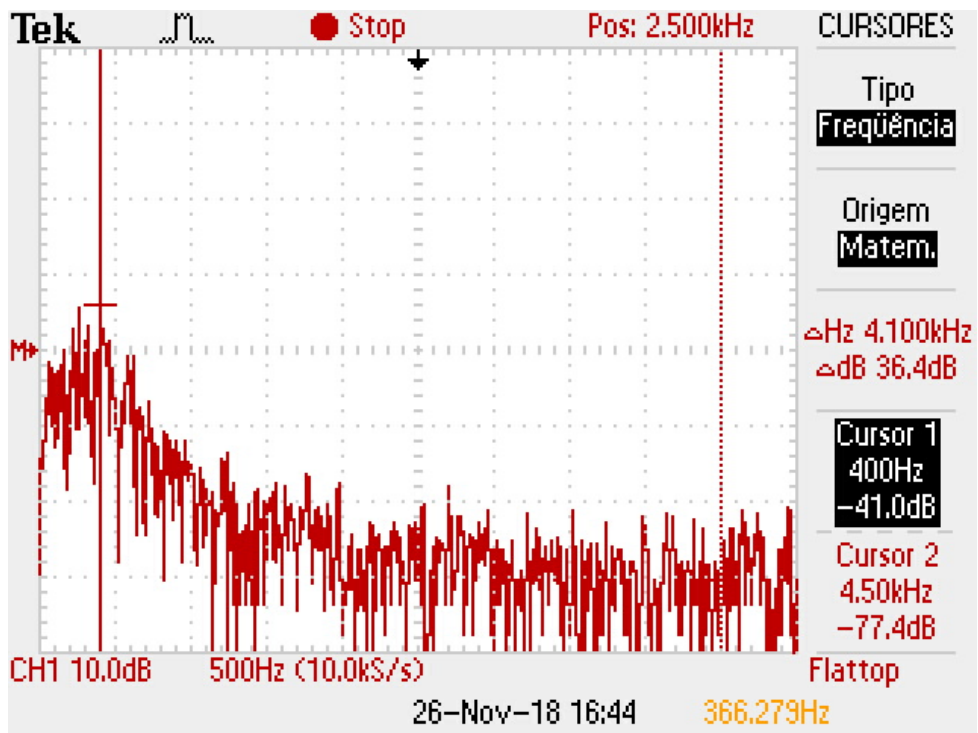
Fonte: Aatoria Própria, 2018

Figura 53 – Saída passa-banda do filtro para *Peak* máximo

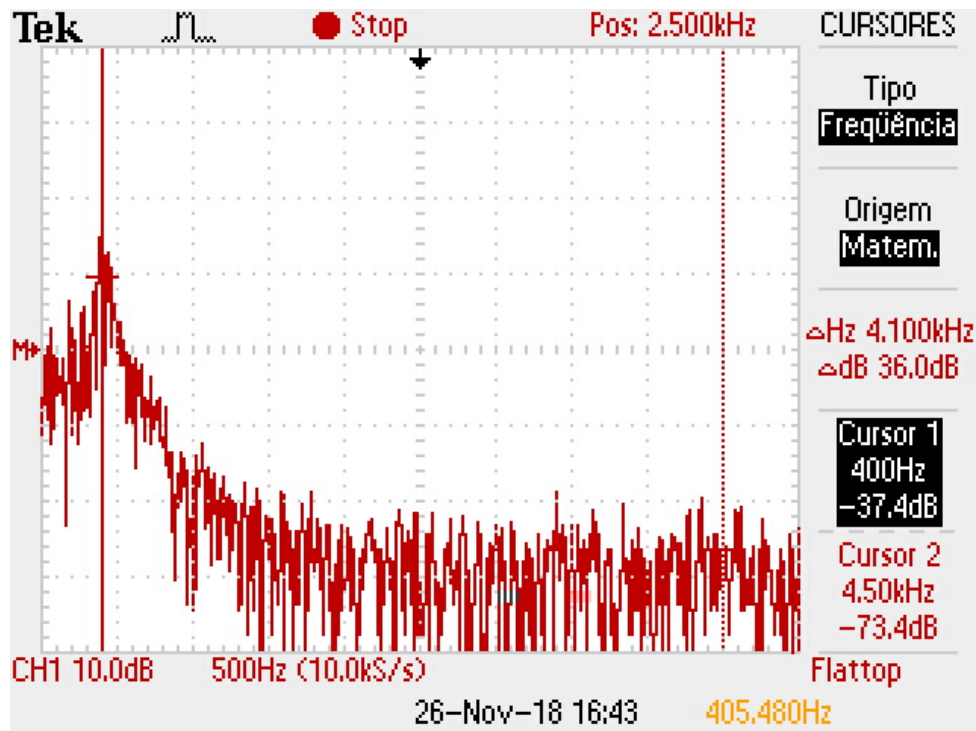


Fonte: Aatoria Própria, 2018

Figura 54 – Saída passa-baixa do filtro para *Peak* mínimo



Fonte: Aatoria Própria, 2018

Figura 55 – Saída passa-baixa do filtro para *Peak* máximo

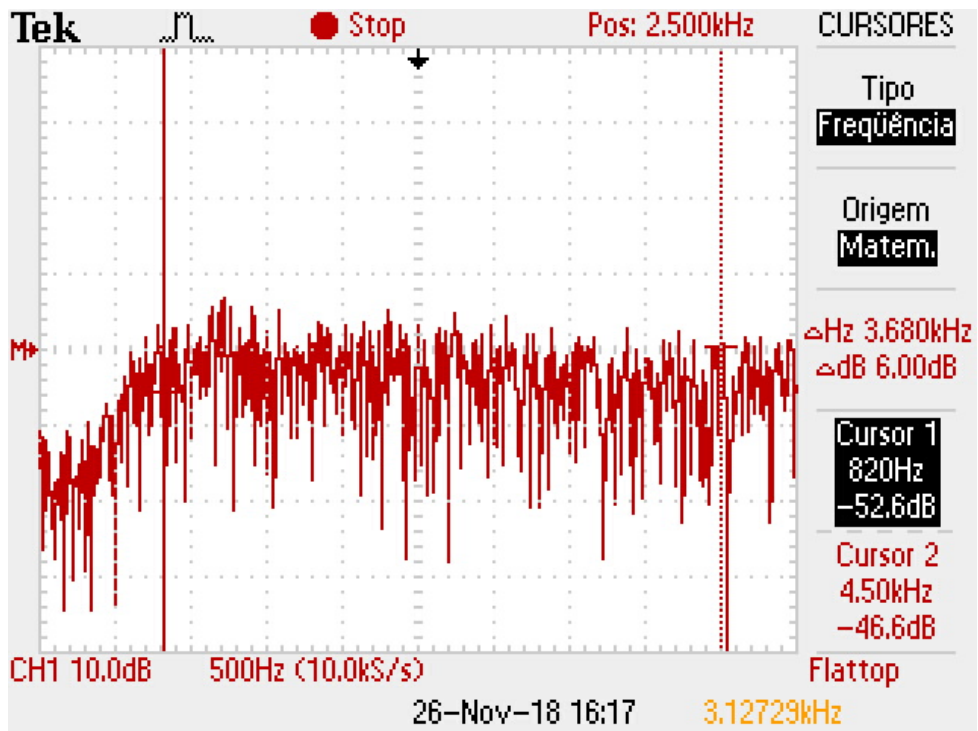
Fonte: Autoria Própria, 2018

Com o parâmetro *Range* posto em nível alto são medidas as respostas em frequência para as três saídas do filtro para valores máximo e mínimo do parâmetro *Peak*, como mostrado nas figuras 56, 57, 58, 59, 60 e 61.

É possível atestar o funcionamento dos filtros como o esperado. Os valores das respostas em frequência encontradas nas medições do circuito físico possuem significativa semelhança com as respostas encontradas nas simulações.

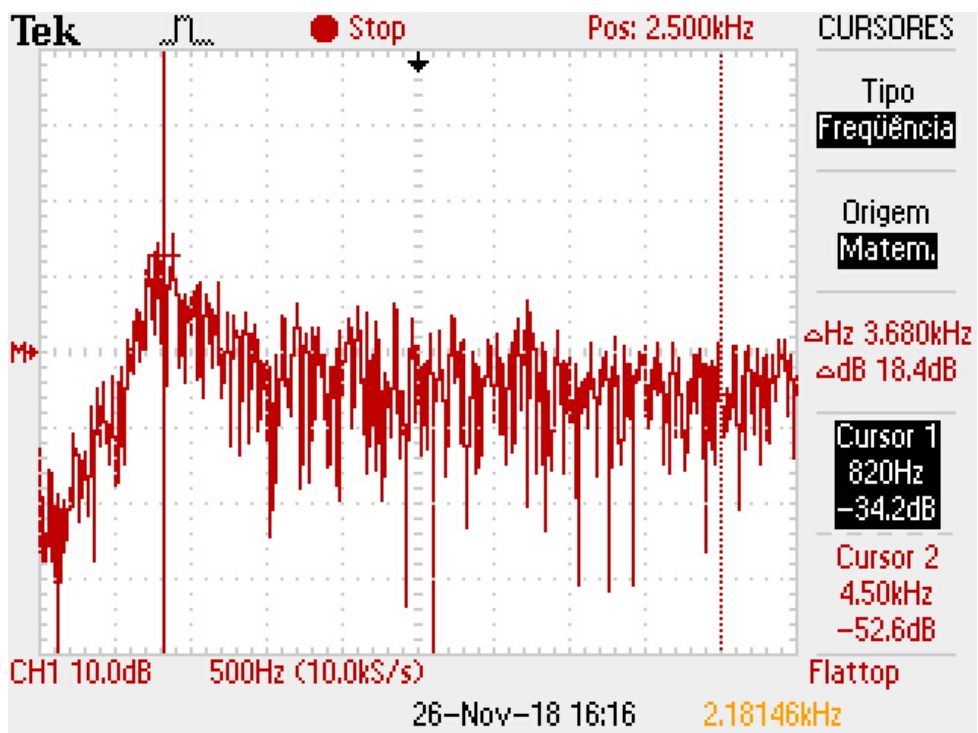
As frequências de corte encontradas foram as mesmas para as três saídas do filtro. Ao variar o parâmetro *Peak* foi atestada modificação na magnitude do sinal nos valores próximos à frequência de corte como o esperado. Variando o parâmetro *Range*, a mudança no valor de corte foi aproximadamente a mesma que o encontrado na simulação do circuito do Mu-tron III.

Figura 56 – Saída passa-alta do filtro para *Peak* mínimo



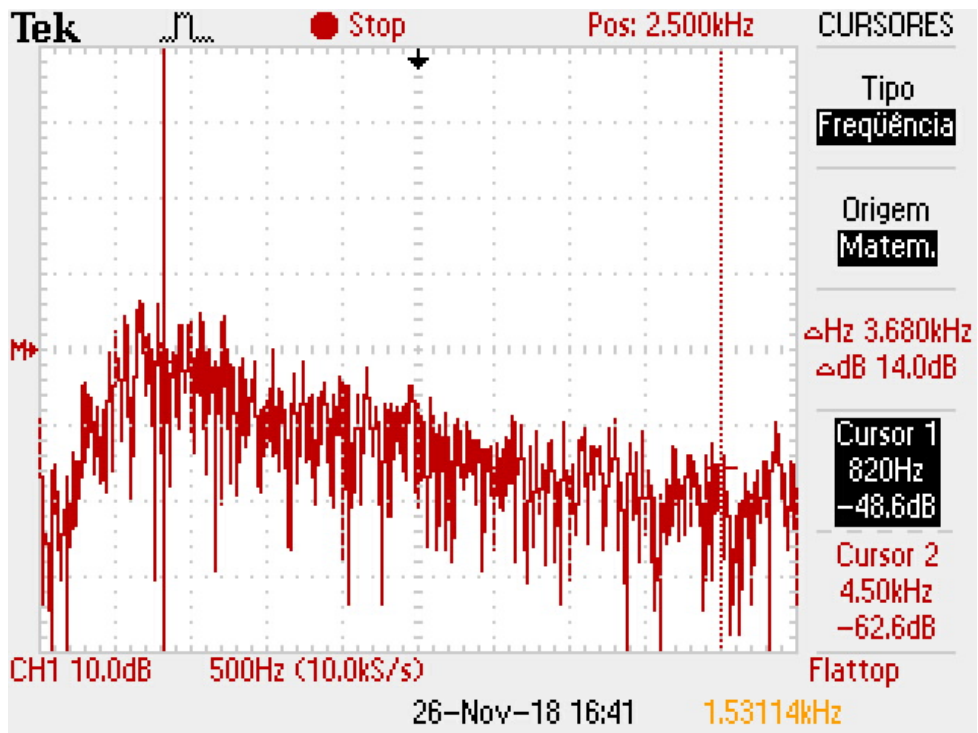
Fonte: Aatoria Própria, 2018

Figura 57 – Saída passa-alta do filtro para *Peak* máximo



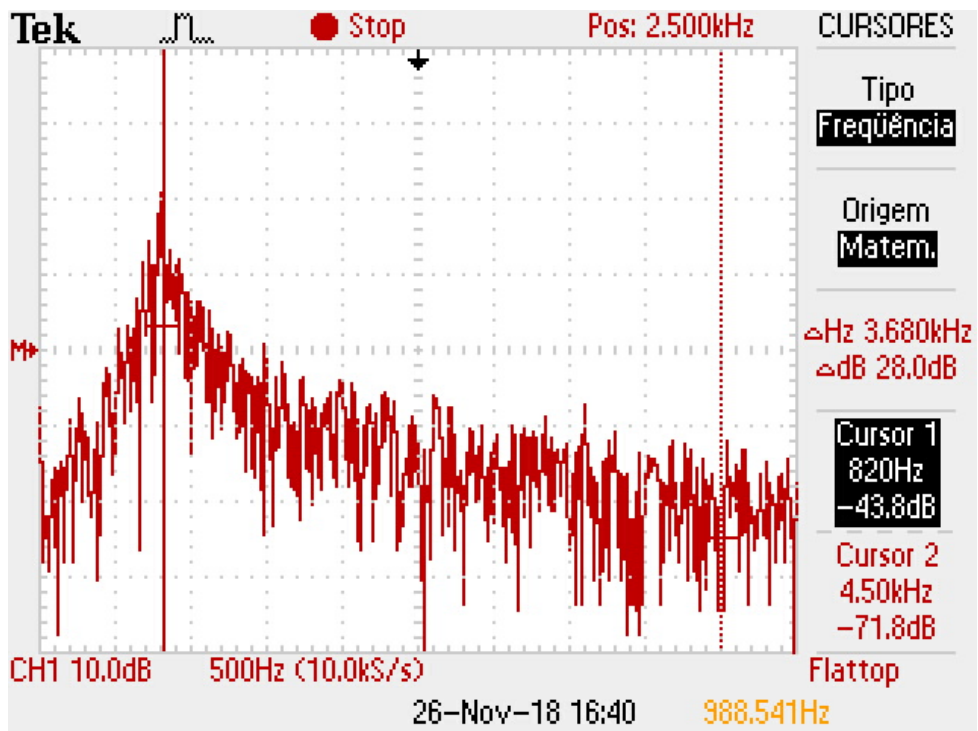
Fonte: Aatoria Própria, 2018

Figura 58 – Saída passa-banda do filtro para *Peak* mínimo



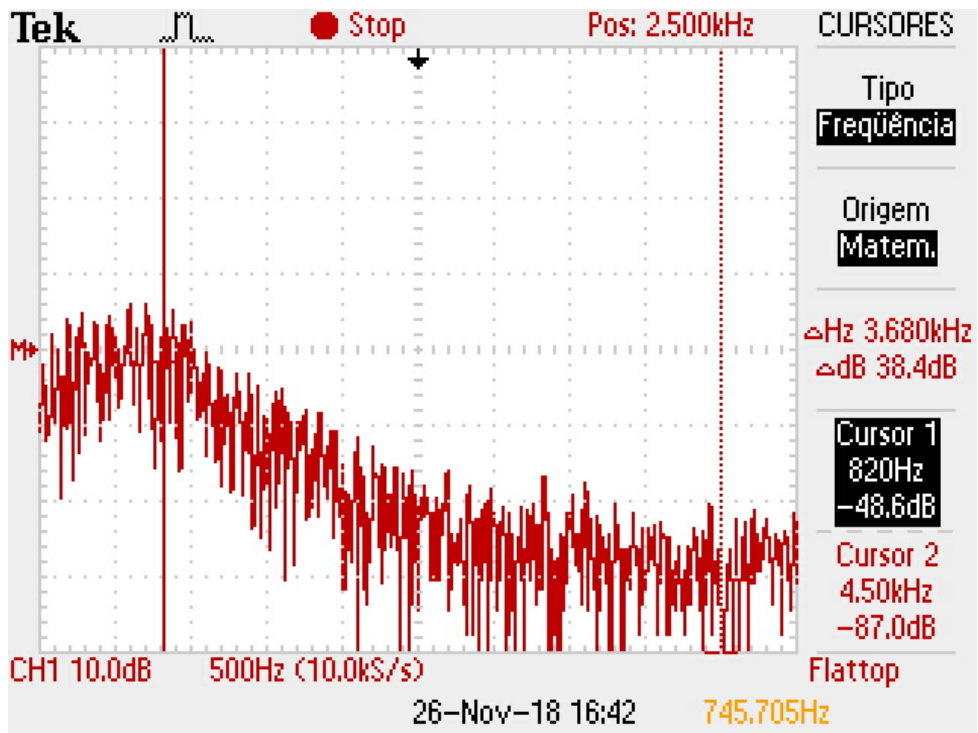
Fonte: Aatoria Própria, 2018

Figura 59 – Saída passa-banda do filtro para *Peak* máximo



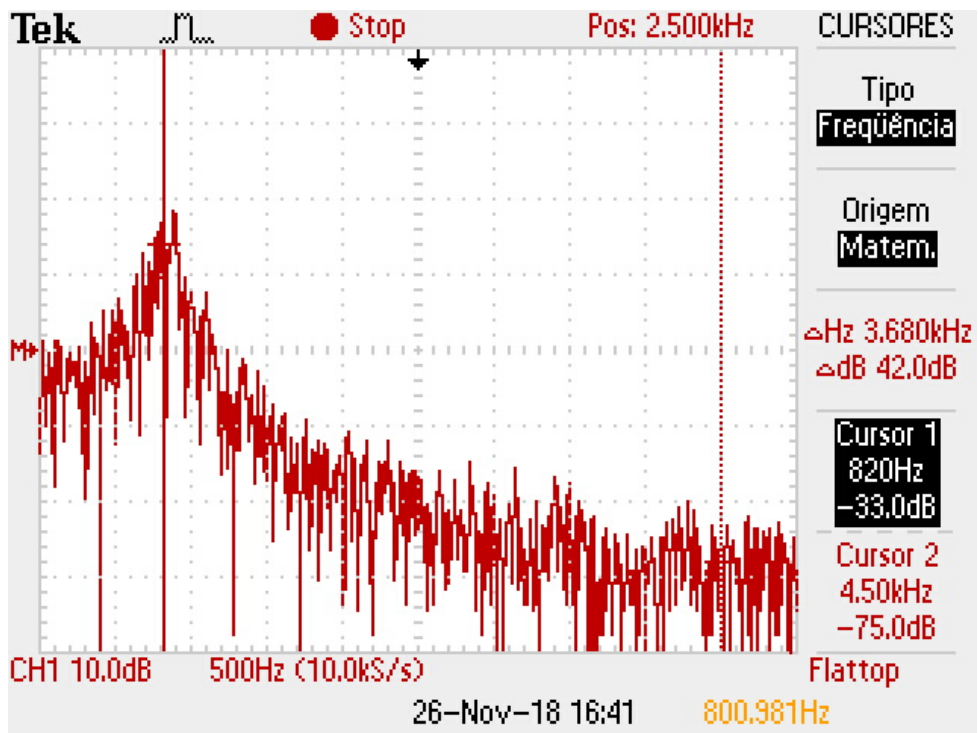
Fonte: Aatoria Própria, 2018

Figura 60 – Saída passa-baixa do filtro para *Peak* mínimo



Fonte: Autoria Própria, 2018

Figura 61 – Saída passa-baixa do filtro para *Peak* máximo



Fonte: Autoria Própria, 2018

Conclusão

Realizando os procedimentos descritos na metodologia e analisando os resultados obtidos dos testes no circuito em comparação com os resultados das simulações no circuito original, foi possível atestar a reprodução fidedigna do projeto implementado. Apesar da impossibilidade de realizar testes físicos em um circuito do pedal de efeito original Mu-tron III, e a conseqüente impossibilidade de verificar uma correlação numérica entre os valores do circuito original com o circuito implementado, uma comparação qualitativa das formas de onda encontradas nos resultados das medições mostrou uma significativa semelhança para ambos os circuitos.

O bloco de entrada do circuito implementado apresentou as diferenças mais significativas com relação ao pedal original, tendo em vista a necessidade de adequar o sinal de entrada para a leitura no ADC do microprocessador. A limitação nos valores de tensão de entrada do ADC entre 0 V e 3.3 V acaba por limitar os valores de ganho na entrada do circuito, tornando-a reduzida em relação ao circuito original. Apesar da limitação apresentada, as medições do circuito físico em relação à simulação do circuito implementado mostraram coerência e comprovaram a implementação física correta do projeto. Uma possível solução a ser explorada para a limitação no ganho de entrada encontrada seria implementar o ganho do bloco de pré-processamento na etapa digital do projeto, limitando o bloco de pré-processamento à adição do ganho DC ao sinal de entrada.

A implementação da rotina para cálculo da frequência de corte e implementação do filtro de variáveis de estado mostrou-se bem sucedida. As medições realizadas no osciloscópio mostraram uma equivalência nas respostas em frequência do projeto em relação às respostas encontradas na simulação do circuito original. Testes realizados com instrumentos musicais mostram semelhança notável do som reproduzido pelo circuito implementado com o som mostrado em demonstrações encontradas do pedal original. Todavia, as reproduções do efeito para valores maiores de ganho de entrada foram impossibilitadas pela limitação no ganho de entrada do projeto.

A plataforma de desenvolvimento Teensy 3.6 possibilitou a implementação de todas as operações pretendidas. Foi possível realizar a leitura do sinal de áudio a uma frequência de amostragem de 48 kHz, que apresentou suficiente qualidade na reprodução do áudio de entrada. Deixa-se em aberto a possibilidade de realizar testes em valores maiores de taxa de amostragem em projetos futuros. A quantidade de pinos de entrada da plataforma permitiu a implementação de todos os parâmetros de regulação do projeto.

Para fins de implementações futuras, pode-se listar como sugestões, além da transferência da etapa de ganho de pré-processamento anteriormente descrita, melhorias no

circuito analógico como a redução nas dimensões da placa de circuito impresso de forma que o circuito projetado possa ser incluso em uma caixa física que permite maior proteção aos componentes e diminuição de interferência eletromagnética. O circuito de alimentação também pode ser adaptado de forma que permita alimentar o circuito com tensão de 9 V, ao invés dos -5 V e +5 V utilizados no projeto.

Especificações mais detalhadas do circuito implementado estão inclusas no relatório (APÊNDICE E), bem como uma estimativa de orçamento para a construção do projeto (APÊNDICE F). Espera-se que o projeto realizado possa ser utilizado como base para projetos futuros na área de processamento digital de sinais com ênfase em processamento de áudio.

Referências Bibliográficas

- BEIGEL, Michael L. Sound Effects Generator: Mu-tron III. 3911776. 11 de nov. de 1973. Warwick, N.Y., n. 411764, p. 4 Disponível em: <<http://www.mu-tron.org/patent1.html>>. Acesso em: 11 set. 2017.
- ZÖLZER, Udo. *DAFX - Digital Audio Effects*. 2. ed. [S.l.]: John Wiley & Sons, Ltd, 2002. 553 p.
- CHAMBERLIN, Hal. *Musical Applications of Microprocessors*. 2. ed. Indianapolis, Indiana: Hayden Books, 1987. 835 p.
- BORGES, Antônio N. *Introdução à Física Acústica* 1. ed.
- TUPAC *Cloning pedals from circuit schematics?* Disponível em: <<https://www.talkbass.com/threads/cloning-pedals-from-circuit-schematics.1260298/>>. Acesso em: 20 jan. 2018
- DAILEY, Denton J. *Electronics for Guitarists* 2. ed. Butler, USA. Springer, 2013. 415 p.
- SEDRA, S; SMITH, K. *Microeletrônica* 4. ed. São Paulo, Brasil. Pearson, 2005. 696 p.
- ELECTRONICS TUTORIALS *Second Order Filters* Disponível em: <<https://www.electronics-tutorials.ws/filter/second-order-filters.html>>. Acesso em: 10 de outubro de 2018.
- BYRON, J. *Proto Pedal Example: Programmable Digital Pedal* Disponível em: <<https://learn.sparkfun.com/tutorials/proto-pedal-example-programmable-digital-pedal>>. Acesso em: 5 de agosto de 2018.
- YIU, Joseph *The Definite Guide to the ARM® Cortex-M3* 2. ed. Burlington, USA. Stellaris, 2007. 531 p.
- KLEIN, Barry. *Electronic Music Circuits*. 1. ed. Indianapolis, Indiana: Howard W. Sams & Co., Inc., 1982. 302 p.
- KLIMACH, Hamilton D. *ENG04-033 - Circuitos Eletrônicos II: Polígrafo da Disciplina e Listas de Exercícios*. 1. ed. [S.l.: s.n.], 1992. 79 p.
- LATHI, B. P. *Modern Digital And Analog Communication Systems*. 3. ed. New York, Oxford: OXFORD UNIVERSITY PRESS, 1998. 781 p.
- OPPENHEIM, Alan V.; SCHAFER, Ronald W.; BUCK, John R. *Discrete-Time Signal Processing*. 2. ed. Upper Saddle River, New Jersey: Prentice Hall, 1998. 896 p.
- KEEN, R.G. *Neutron Filter: A Mutron III Workalike*. 1. ed. [S.l.: s.n.], [1999-2001]. 8 p. Disponível em: <http://www.generalguitargadgets.com/wp-content/uploads/neutron_filter.pdf>

JOHANN, S.F. *ARM Toolchain* Disponível em: <https://github.com/sjohann81/hellfireos/blob/master/usr/tools/arm-toolchain/gcc-7.1.0/build_arm_toolchain>. Acesso em: 13 de setembro de 2018

BOOTLIN *Embedded Linux System Development* Disponível em: <<https://bootlin.com/doc/training/embedded-linux/embedded-linux-slides.pdf>>. Acesso em: 13 de setembro de 2018

STOFFREGEN, P. *kinetis.h* Disponível em: <<https://github.com/PaulStoffregen/cores/blob/master/teensy3/kinetis.h>>. Acesso em: 13 de setembro de 2018

STOFFREGEN, P. *core_pins.h* Disponível em: <https://github.com/PaulStoffregen/cores/blob/master/teensy/core_pins.h>. Acesso em: 13 de setembro de 2018

POPP, C.; SORIA-LUZ, R. *Developing mixer-style controllers based on Arduino/Teensy microcontrollers* 5 p. Artigo Científico Disponível em: <http://www.academia.edu/15763916/DEVELOPING_MIXER-STYLE_CONTROLLERS_BASED_ON_ARDUINO_TEENSY_MICROCONTROLLERS>. Acesso em: 5 de julho de 2018

MURTHY, A. *Design and Construction of Arduino-Hacked Variable Gating Distortion Pedal* 9 p. Artigo Científico Disponível em: <<https://ieeexplore.ieee.org/document/6965623/>>. Acesso em: 3 de julho de 2018

Apêndices

APÊNDICE A – Rotinas para testes em MATLAB®

Listing A.1 – envdetector.m

```

1
2 t=[0:0.1:100];
3 buffer=0.35*sin(t);
4 past_env=0;
5 EXPVAL=0.9997;
6 env=zeros(size(buffer));
7
8 for i=1:length(buffer)
9
10     buff=buffer(i);
11
12     if buff >= past_env
13
14         env(i)=buff;
15     else
16         env(i)=past_env*EXPVAL;
17     end
18     past_env=env(i);
19 end
20
21 inenv=0.35-env/2;
22
23 plot(t,buffer)
24 % hold on
25 % plot(t,env,'r')
26 hold on
27 plot(t,inenv,'g');
28 hold off
29 ylim([-0.45 0.45])
30 %legend({'Input','Envelope Up Position'})
31 legend({'Input','Envelope Down Position'})

```

Listing A.2 – svfilter.m

```

1
2 fc=380;
3 Q=16;
4 Fs=48000;
5
6
7 q1=2/Q;
8 %f1=2.0*sin(pi*fcshigh(5)/Fs);
9 f1=2.0*sin(pi*fc/Fs);
10
11 D11=f1^2+f1*q1-2; %elemento do denominador que multiplica z^-1
12 D22=1-f1*q1;
13
14 %funcao de transferencia passa altas

```

```
15 nh=[1 -2 1];
16 dh=[1 D11 D22];
17
18 %funcao de transferencia passa banda
19 nb=[f1 -f1 0];
20 db=[1 D11 D22];
21
22 %funcao de transferencia passa baixas
23 nl=[0 f1^2 0];
24 dl=[1 D11 D22];
25
26 hp=filt(nh,dh,1/Fs);
27 bp=filt(nb,db,1/Fs);
28 lp=filt(nl,dl,1/Fs);
29
30 options = bodeoptions;
31 options.FreqUnits = 'Hz';
32 options.Xlim = [1,20000];
33 bode(hp,bp,lp,options);
34 legend('HP','BP','LP');
35 grid;
```

APÊNDICE B – Código completo em MATLAB®

Listing B.1 – autowah.m

```

1  %parametros de regulagem
2  gain = 6;          %ganho de entrada
3  peak = 8;         %fator Q do filtro
4  drive = 0;        %direcao envoltoria 0==up, 1==down
5  mode = 3;         %selecao do filtro 1==HP, 2==BP, 3==LP
6  range = 0;        %frequencia central 0==low, 1==high
7  EXPVAL=0.9997;
8  PI=3.14159265358979323846;
9
10 %%
11 Gain=0.2*gain;
12 Peak=1.5*peak;
13
14 %%
15 %aquisicao do sinal de entrada
16 [signalstereo ,Fs]= wavread('BassC3.wav');
17 Ts=1/Fs;
18 Gsig = sum(signalstereo , 2) / size(signalstereo , 2);
19 signal = (Gsig/max(Gsig)*2048)+2048;      %ganho de entrada
20
21 %parametros de regulagem
22 gain = 6;          %ganho de entrada
23 peak = 8;         %fator Q do filtro
24 drive = 0;        %direcao envoltoria 0==up, 1==down
25 mode = 3;         %selecao do filtro 1==HP, 2==BP, 3==LP
26 range = 0;        %frequencia central 0==low, 1==high
27
28 %%
29 Gain=0.2*gain;
30 Peak=1.5*peak;
31
32 %%
33 %aquisicao do sinal de entrada
34 [signalstereo ,Fs]= wavread('BassC3.wav');
35 Ts=1/Fs;
36 Gsig = sum(signalstereo , 2) / size(signalstereo , 2);
37 signal = (Gsig/max(Gsig)*2048)+2048;      %ganho de entrada
38
39 yh=zeros(size(Gsig));
40 yb=zeros(size(Gsig));
41 yl=zeros(size(Gsig));
42 past_env=0.0;
43 past_yb=0.0;
44 past_yl=0.0;
45
46 %%
47 for i=1:length(signal)
48

```

```
49     buff=(signal(i)-2048)/2048;
50
51     f0=1350;
52     df=1250;
53     f_max=f0+df;
54     f_min=f0-df;
55
56     if buff>=past_env
57
58         env=buff;
59     else
60         env=past_env*EXPVAL;
61     end
62
63     nenv=env;
64     fc=f_min+(f_max-f_min)*nenv;
65
66     f1=2.0*sin(PI*fc/Fs);
67     q1=2.0/Peak;
68
69     yh(i)=buff - past_y1 - q1*past_yb;
70     yb(i)=f1*yh(i) + past_yb;
71     yl(i)=f1*yb(i) + past_y1;
72
73     past_env=env;
74     past_yb=yb(i);
75     past_y1=yl(i);
76
77 end
```

APÊNDICE C – Código em C

Listing C.1 – fifo.h

```

1
2 #define BUFFER_SIZE      512
3 #define BUFFER_MASK     (BUFFER_SIZE - 1)
4
5 struct data_s {
6     int16_t buffer[BUFFER_SIZE];
7     uint16_t head, tail, size;
8 };
9
10 void flushfifo(struct data_s *buf);
11 int16_t putfifo(struct data_s *buf, int16_t data);
12 int16_t getfifo(struct data_s *buf);
13 uint16_t getfifosize(struct data_s *buf);

```

Listing C.2 – main.c

```

1 #include "kinetis.h"
2 #include "core_pins.h"
3 #include "fifo.h"
4 #define BUF_SIZE 50
5 // #define DF ((float)1250.0)
6 #define EXPVAL ((float)0.9997)
7 #define PI ((float)3.14159265358979323846)
8 #define TWOPI ((float)6.28318530717958647692)
9 #define FS ((float)48000.0)
10
11 static struct data_s adc_data;
12 static struct data_s dac_data;
13
14 /* 60 MHz / 1 / 10 / FS Hz */
15 #define PDB_PERIOD (F_BUS / 10 / FS)
16 #define PDB_CONFIG (PDB_SC_TRGSEL(15) | PDB_SC_PDBEN | PDB_SC_PDBIE | PDB_SC_CONT |
17     PDB_SC_PRESCALER(0) | PDB_SC_MULT(1))
18 float sin(float rad)
19 {
20     float sine;
21
22     if (rad < -PI)
23         rad += TWOPI;
24     else if (rad > PI)
25         rad -= TWOPI;
26
27     if (rad < 0)
28         sine = 1.273239544735162686 * rad + 0.405284734569351086 * rad * rad;
29     else
30         sine = 1.273239544735162686 * rad - 0.405284734569351086 * rad * rad;
31
32     if (sine < 0)

```

```

33     sine = 0.225 * (sine * -sine - sine) + sine;
34     else
35         sine = 0.225 * (sine * sine - sine) + sine;
36
37     return sine;
38 }
39
40 void setup_pdb(void)
41 {
42     __disable_irq()
43     /* Enable PDB clock */
44     SIM_SCGC6 |= SIM_SCGC6_PDB;
45     /* Timer period */
46     PDB0_MOD = PDB_PERIOD;
47     /* Interrupt delay */
48     PDB0_IDLY = 0;
49     /* PDB status and control */
50     PDB0_SC = PDB_CONFIG;
51     /* Software trigger (reset and restart counter) */
52     PDB0_SC |= PDB_SC_SWTRIG;
53     /* load OK */
54     PDB0_SC |= PDB_SC_LDOK;
55     /* Enable interrupt request */
56     NVIC_ENABLE_IRQ(IRQ_PDB);
57     __enable_irq();
58 }
59
60 void pdb_isr() {
61     /* get data from ADC and put it in the ADC FIFO */
62     putfifo(&adc_data, analogRead(A0));
63
64     /* if there is data on the DAC FIFO, write it out */
65     if (getfifosize(&dac_data))
66         analogWrite(A21, getfifo(&dac_data));
67
68     /* clear interrupt flag */
69     PDB0_SC &= ~PDB_SC_PDBIF;
70 }
71
72 int main(){
73     float buff;
74     float env;
75     float past_env;
76     float f0, f_min, f_max, df;
77     float f1, q1, fc;
78     float peak;
79     float yh, yb, yl;
80     float past_yb, past_yl;
81     float output;
82
83     init();
84     pinMode(A0, INPUT);      //signal input
85     pinMode(12, INPUT);     //range control
86     pinMode(24, INPUT);     //drive control
87     pinMode(26, INPUT);     //mode control
88     pinMode(28, INPUT);     //mode control
89     pinMode(31, INPUT);     //peak control
90     pinMode(A21, OUTPUT);   //signal output
91
92     analogWriteResolution(12);

```

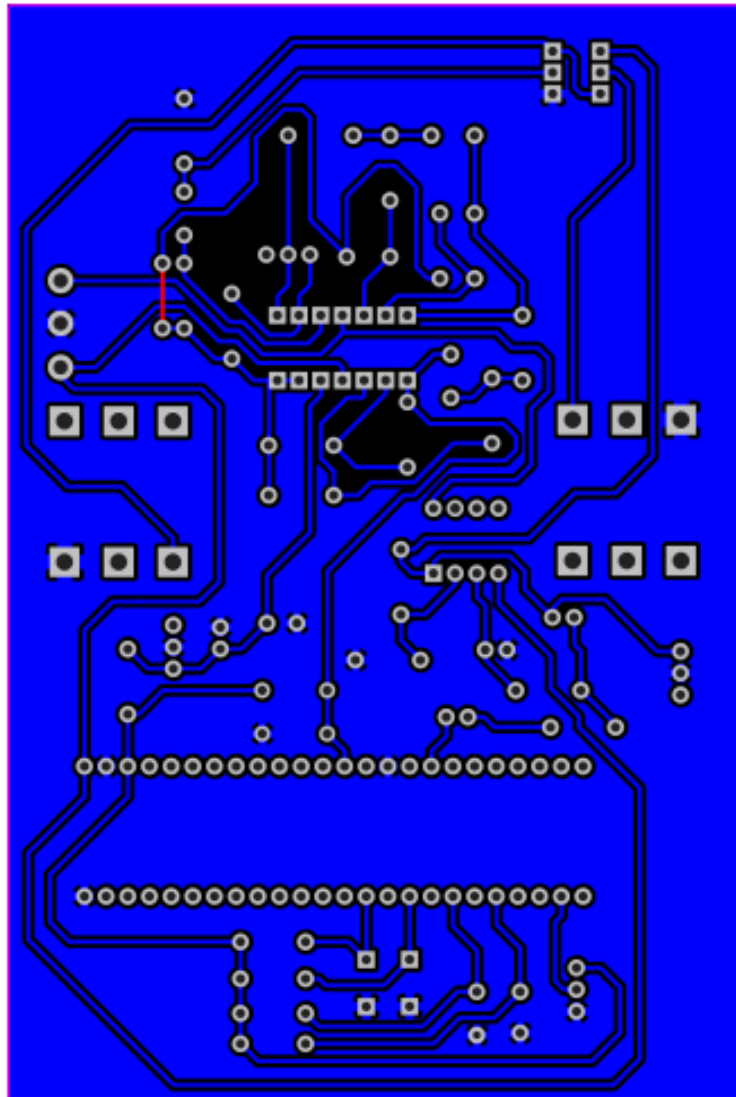


```
93     analogReadResolution(12);
94
95     setup_pdb();
96
97     flushfifo(&adc_data);
98     flushfifo(&dac_data);
99
100
101     past_env=0.0;
102     past_yb=0.0;
103     past_yl=0.0;
104
105     while (1){
106         if (getfifo(&adc_data) > 20) {
107
108             buff = (getfifo(&adc_data) - 2048.0) / 2048.0;
109
110             if (digitalRead(12) == HIGH) {
111                 f0 = 1230.0;
112                 df = 970.0;
113             }
114             else {
115                 f0 = 2690.0;
116                 df = 2210.0;
117             }
118             f_max = f0 + df;
119             f_min = f0 - df;
120
121
122             if (buff >= past_env) {
123
124                 env = buff;
125             }
126
127             else {
128
129                 env = past_env * EXPVAL;
130             }
131
132             if (digitalRead(24) == HIGH) {
133                 fc = f_min + (f_max - f_min) * env;
134             }
135             else {
136                 fc = f_min + (f_max - f_min) * (1 - env);
137             }
138
139             f1 = 2.0 * sin(PI * fc / FS);
140             peak = 3.09 + analogRead(31) / 300.05; //1 + analogRead(31) / 220;
141             q1 = 2.0 / peak;
142
143             yh = buff - past_yl - q1 * past_yb;
144             yb = f1 * yh + past_yb;
145             yl = f1 * yb + past_yl;
146
147             past_env = env;
148             past_yb = yb;
149             past_yl = yl;
150
151             if (digitalRead(26) == HIGH && digitalRead(28) == HIGH) {
152                 output = (yh * 2048.0) + 2048.0;
```

```
153     }
154     else if (digitalRead(26)==HIGH && digitalRead(28)==LOW){
155         output=(yb*2048.0)+2048.0;
156     }
157     else {
158         output=(yl*2048.0)+2048.0;
159     }
160     putfifo(&dac_data, output);
161 }
162 }
163 }
```

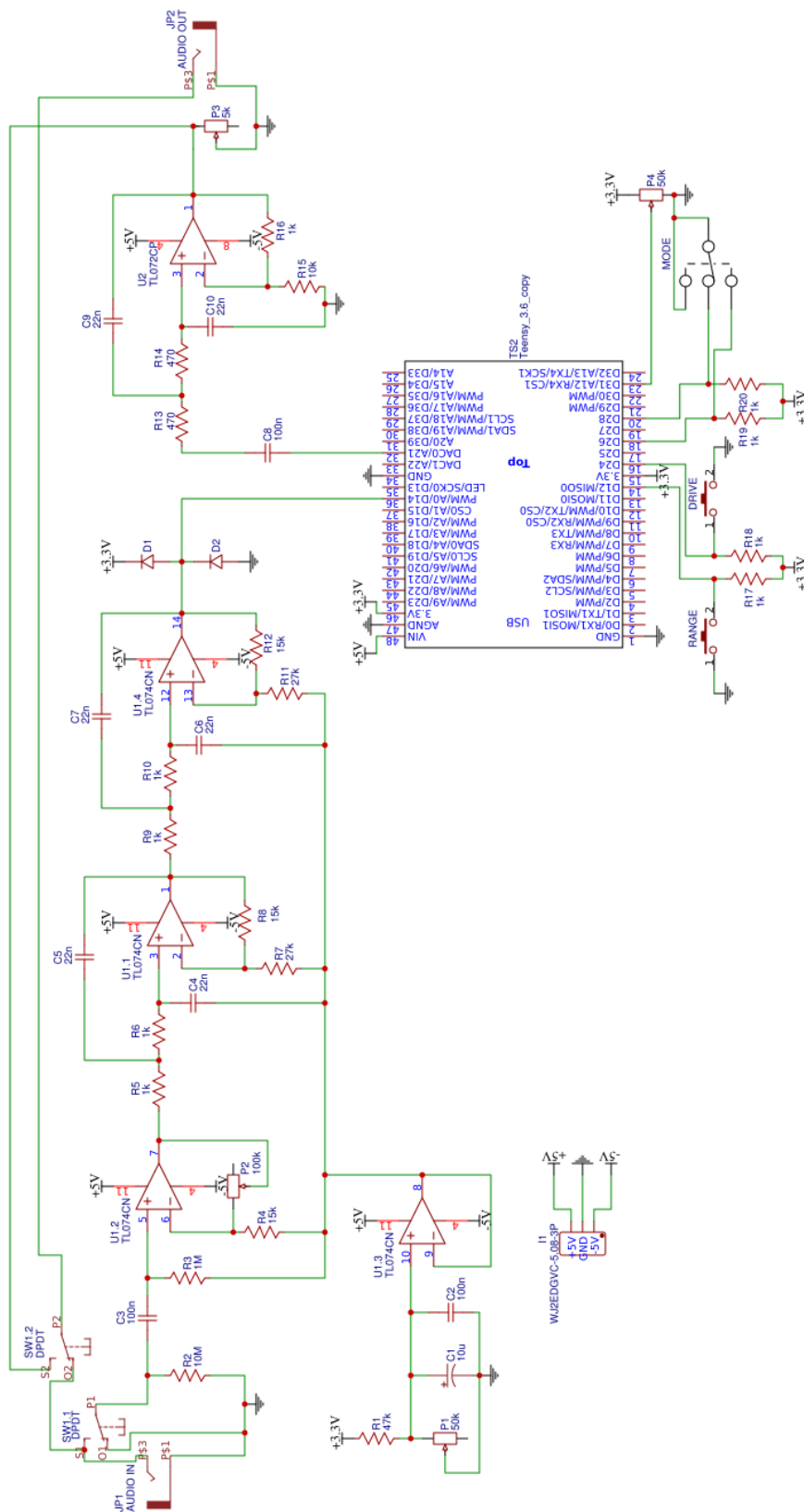
APÊNDICE D – Layout e Esquemático do Circuito

Figura 62 – *Layout* da placa para o circuito desenvolvido



Fonte: Autoria Própria, 2018

Figura 63 – Esquemático do circuito desenvolvido



Fonte: Autoria Própria, 2018

APÊNDICE E – Especificações do Projeto

Tabela 1 – Especificações do projeto

Pedal	
Impedância de entrada	10 M Ω
Impedância de saída	5 k Ω max
Nível do sinal de entrada	-17.78 dBu
Nível do sinal de saída	3.56 dBu max
Frequência de amostragem	48 kHz
Resolução de leitura	12 bits

Fonte	
Tensão de entrada	127/220 V
Tensão de saída	-5/0/5 V
Consumo de corrente	200 mA max

APÊNDICE F – Orçamento do Projeto

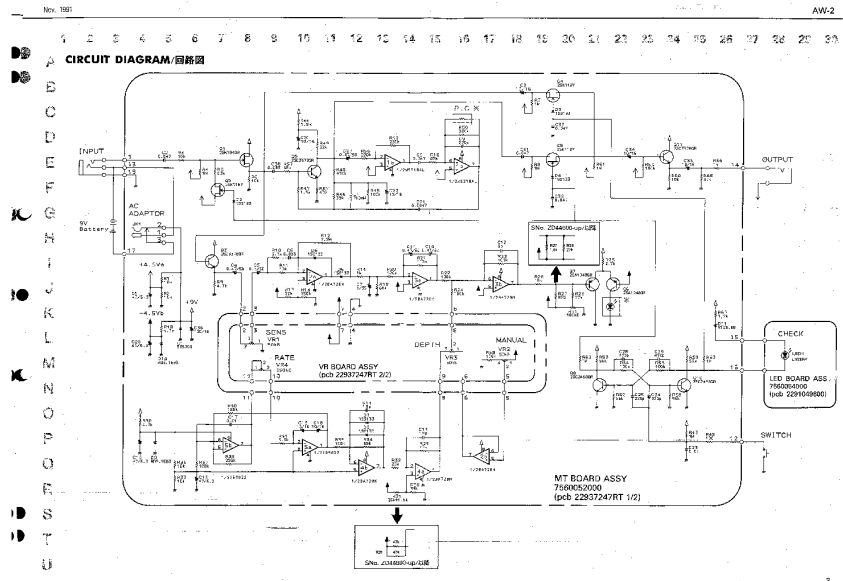
Tabela 2 – Orçamento do projeto

Pedal	
Componente	Preço (R\$)
Placa Fenolite 10x15cm	R\$ 12,00
Componentes Eletrônicos	R\$ 60,00
Teensy 3.6	R\$ 260,00
Fonte	
Transformador 110/220 V -9+9 V	R\$ 20,00
Componentes Eletrônicos	R\$ 15,00
Total	R\$ 367,00

Anexos

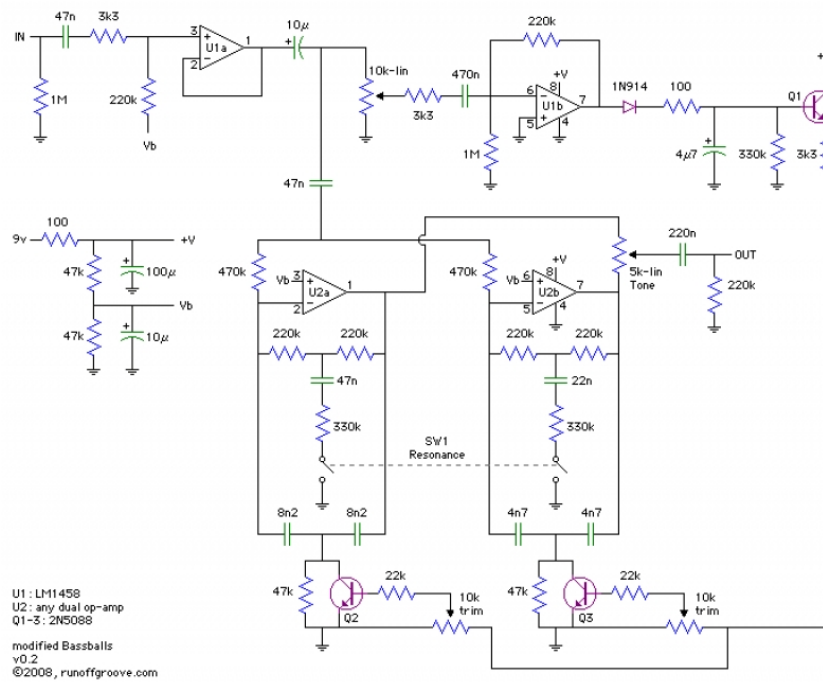
ANEXO A – Esquemáticos de Pedais de Efeito Auto-Wah Comerciais

Figura 64 – Esquemático do Boss Auto Wah



Fonte: BOSS CORPORATION, 2000

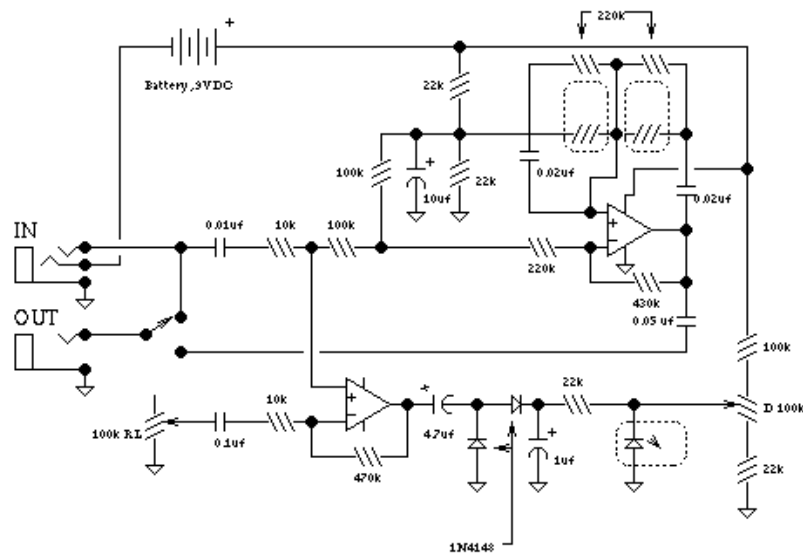
Figura 65 – Esquemático do EHX QTron+



Fonte: Modified Bassballs, 2008

Figura 66 – Esquemático do DOD Envelope Filter 440

DOD Envelope Filter 440

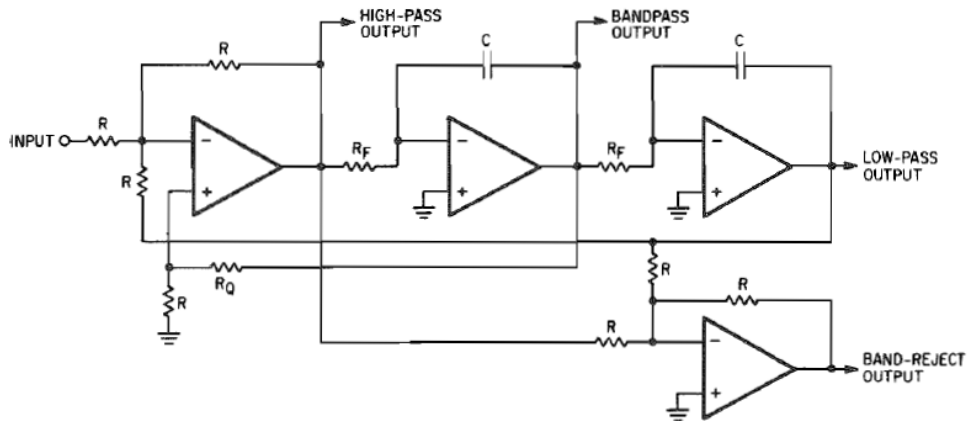


Opamps are each 1/2 of TL022 dual low power op amp. LED/LDR module is unknown, but is probably a Vactec VTL module with LED to center-tapped LDR.

Fonte: DOD Electronics, 2014

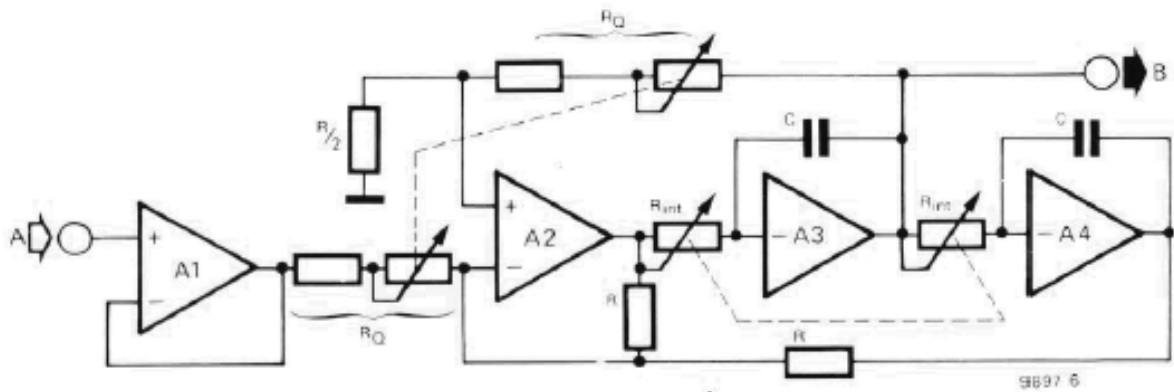
ANEXO B – Esquemáticos de Filtros de Variáveis de Estado

Figura 67 – Filtro de Variáveis de Estado



Fonte: CHAMBERLIN, H., 1987

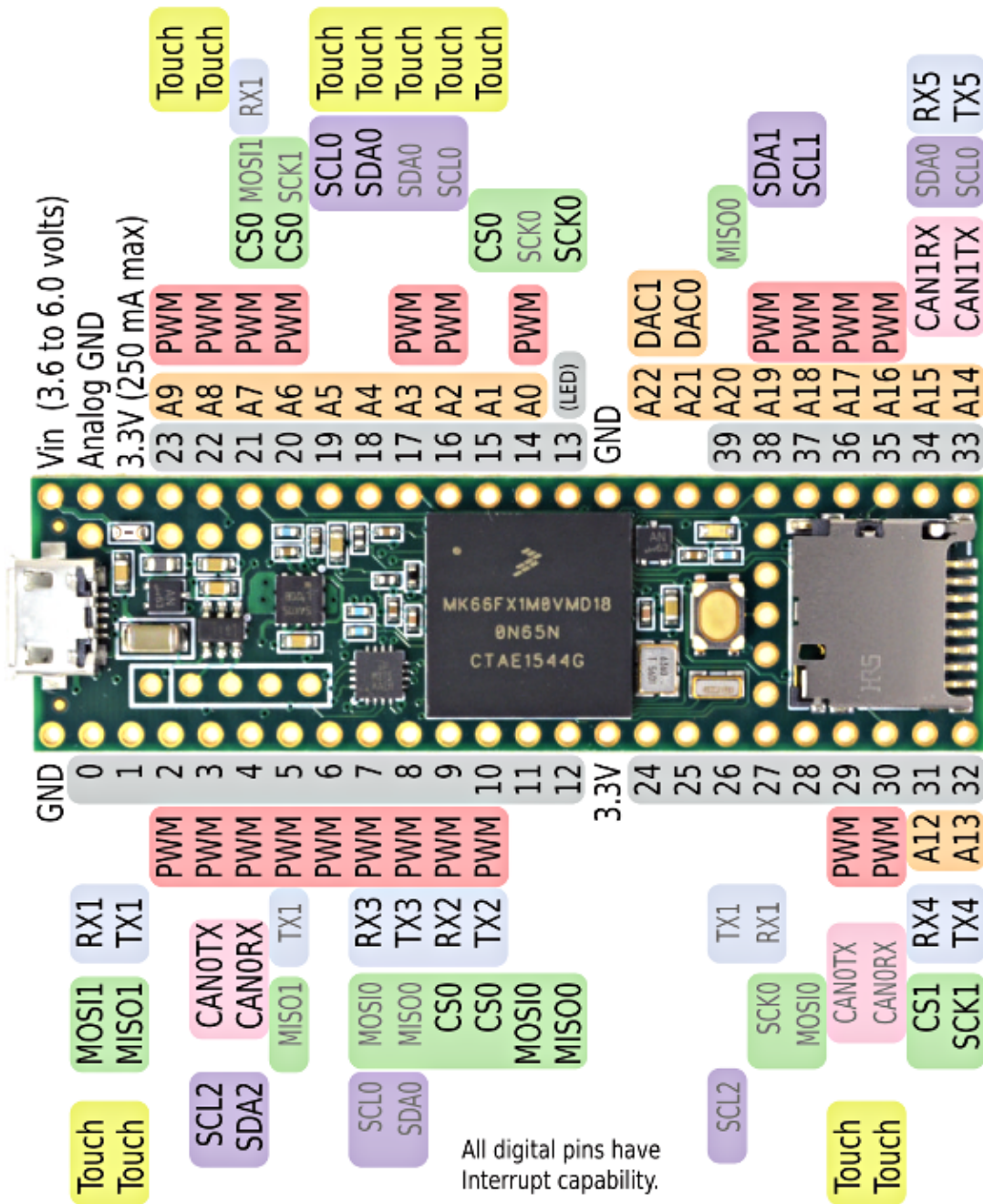
Figura 68 – Filtro de Variáveis de Estado



Fonte: Elektor Magazine, 1986

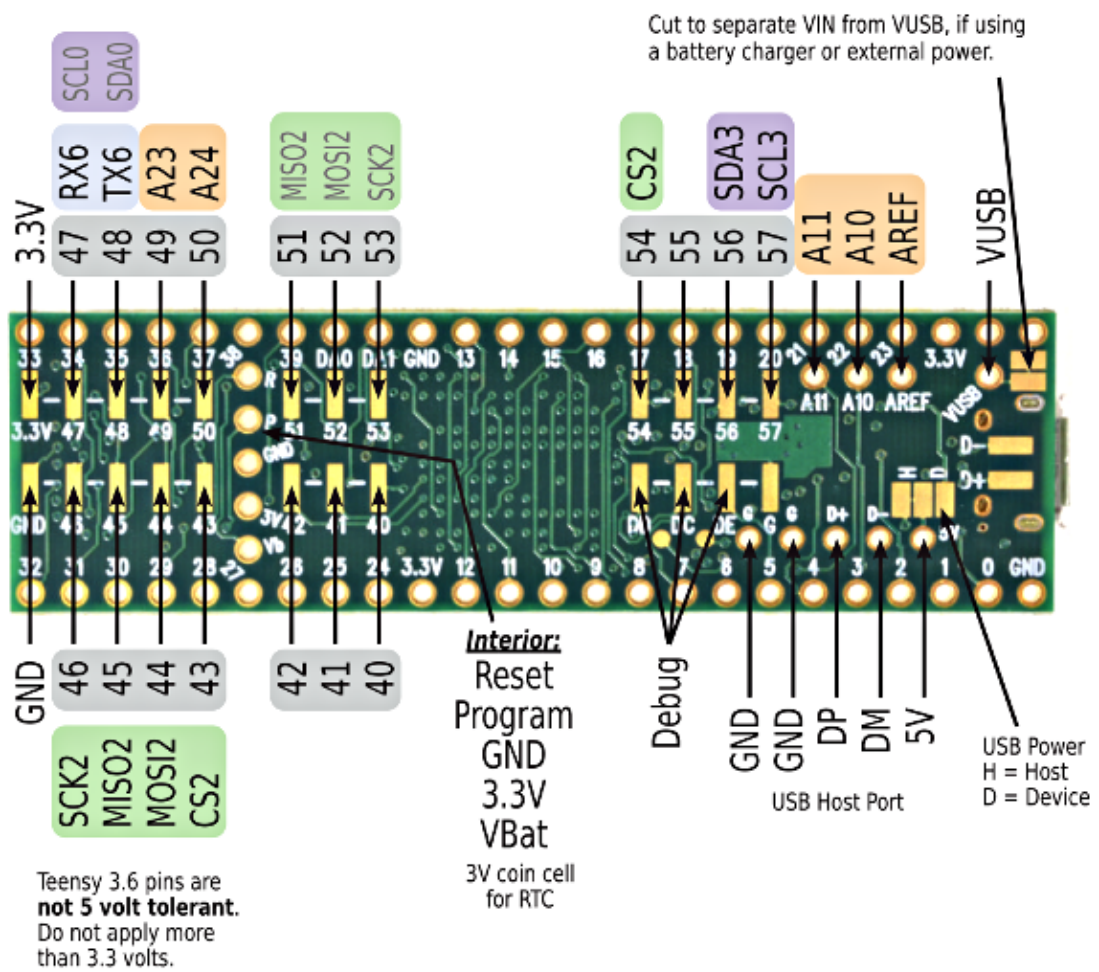
ANEXO C – Pinagem do *Teensy 3.6*

Figura 69 – Pinagem do *Teensy 3.6* (1)



Fonte: PJRC, 2018

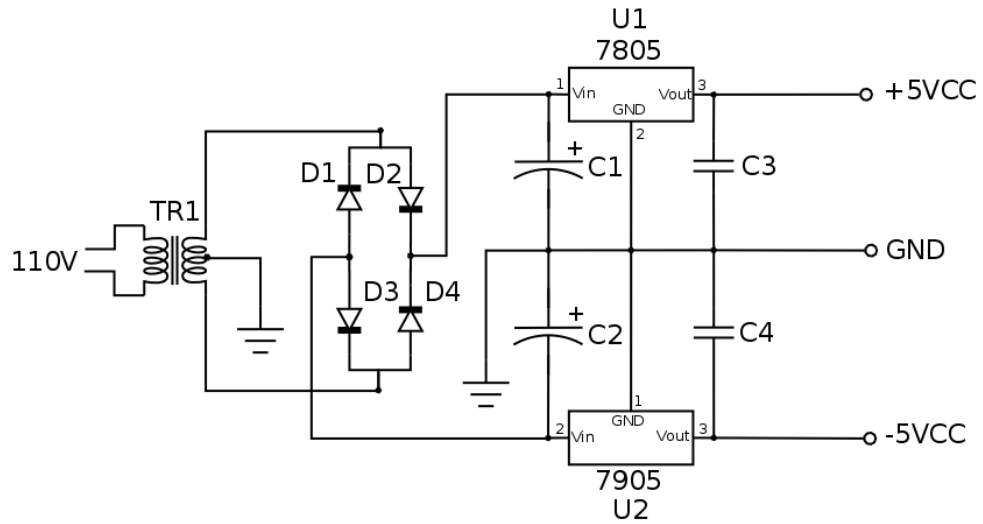
Figura 70 – Pinagem do *Teensy 3.6* (2)



Fonte: PJRC, 2018

ANEXO D – Fonte de Alimentação

Figura 71 – Fonte simétrica para alimentação do circuito



Fonte: <<http://www.c2o.pro.br/hackaguas/apb.html>>