

Processo de Desenvolvimento e Construção de um *Theremin* Digital

Lucas F. P. Costa¹, Rafael H. Albarello¹

¹Engenharia de Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
Toledo – PR – Brasil

{lucascosta74,albarellorafael123}@gmail.com

1. Introdução

Instrumentos musicais eletrônicos ou que fazem uso de eletricidade tem sido desenvolvidos há séculos, podendo ter início reconhecido com o lançamento do *clavecin électrique* (cravo elétrico) inventado por Jean-Baptiste Delaborde em 1759 [8]. Por sua vez, o Theremin foi descoberto em 1920, patenteado em 1928 e teve nome de acordo com seu inventor, Lev Theremin [5].

O funcionamento do Theremin é peculiar se comparado a outros eletrônicos, uma vez que não há contato físico direto para ser manuseado. Sua construção é primordialmente analógica, e seu controle é feito através de duas antenas de metal posicionadas nas suas extremidades, servindo como sensores de proximidade para as mãos do músico, que operam osciladores de frequência e amplitude (intensidade, volume), cujos sinais são por fim amplificados por um alto-falante [6].



Figura 1. Theremin analógico comum moderno.

Mais recentemente, o uso de computadores pessoais e técnicas digitais tem possibilitado o surgimento de instrumentos de música diferenciados com um leque de soluções extremamente amplo, onde as informações podem ser processadas de infinitas formas e reproduzidas posteriormente. Ao invés da manipulação de circuitos elétricos, pode-se lidar com dados em um grau mais elevado de detalhe, programando comportamentos diferenciados de acordo com o desejo do desenvolvedor [7].

Neste projeto, foi proposto o desenvolvimento e construção de um Theremin utilizando técnicas de processamento digital. Os dispositivos necessários foram escolhidos

e um circuito funcional elaborado para a confecção de uma Placa de Circuito Impresso (PCI) onde os componentes puderam ser conectados. Um *software* foi embarcado em um microcontrolador, com o suficiente para operação do sistema. Testes individuais foram realizados para cada componente, e finalmente para o sistema como um todo. O resultado foi o completo funcionamento do dispositivo, atingindo o objetivo proposto.

2. Materiais e Métodos

Para a elaboração deste projeto, foi preciso elencar quais seriam os dispositivos e sensores necessários para solução da proposta. Tendo isto, foi montado o esquemático do circuito e seu equivalente para confecção da PCI. Além disso, também foram desenvolvidos os códigos para serem embarcados e possibilitarem o correto processamento de informações obtidas pela leitura dos sensores. Com a conclusão destes passos, tem-se o sistema completo.

2.1. Escolha de Dispositivos Para o Sistema Proposto

São três componentes principais para que seja possível construir o Theremin digital, descritos a seguir:

- **Sensores:** para capturar os movimentos e distâncias das mãos do músico, foram usados os sensores de distância ultrassônicos HC-SR04, em substituição à função das antenas no Theremin analógico original. Com este sensor, é possível medir distâncias de 2cm a 400cm com precisão aproximada de 3mm [4];
- **Microcontrolador:** a leitura da resposta dos sensores, processamento e manipulação das informações para conversão em dados de áudio e por fim, a reprodução sonora podem ser feitos com uso de um microcontrolador. Neste projeto, foi utilizado o ATmega328P, comumente empregado em placas como Arduino Uno R3 de forma destacável[3]. Seu conversor A/D tem resolução de 10 bits com taxa de amostragem aproximada de 9615Hz, e usando suas saídas PWM é possível gerar sinais que podem ser amplificados por um alto-falante em frequências e intensidades desejadas[1]. Para sua operação, é preciso somente de alimentação de $\pm 5V$ e a utilização de um cristal oscilador de 16MHz para geração de *clock*;
- **Alto-falante:** para cumprir com o objetivo final de geração sonora, um amplificador genérico deve ser usado como saída. Este, adequadamente conectado ao microcontrolador, será responsável por entregar o resultado do processamento feito a partir da leitura obtida dos sensores de distância.

2.2. Confecção da PCI

Um dos objetivos deste projeto foi a montagem de uma PCI própria como solução para o problema proposto. Dentre as diversas formas para montagem, decidiu-se pelo método artesanal fotográfico [10].

Partindo de uma placa de fenolite e cobre pintada com tinta fotossensível, uma máscara com o circuito modelado para PCI é colocada sobre deixando visíveis apenas as partes onde se quer corroer o cobre, sendo exposta à luz ultravioleta, o que faz com que a tinta que estiver exposta adira à placa. Posteriormente, a placa deve ser colocada em uma solução de percloroeto, realizando a corrosão do cobre. O resultado é a PCI, devendo-se ainda perfurar nos locais indicados [10].

2.3. Lógica de Implementação

A implementação do código para ser embarcado no microcontrolador escolhido foi feito aproveitando-se da interface de desenvolvimento própria do Arduino, permitindo utilizar linguagem C, bem como todas as bibliotecas tanto nativas quanto customizadas por usuários.

O código precisa solucionar o seguinte problema: obter valores da leitura dos sensores, processá-los e reproduzir nas saídas de PWM. Para trabalhar com os sensores, a biblioteca *Ultrasonic* [9] foi usada, tendo sido desenvolvida para trabalhar especificamente com o sensor HC-SR04 servindo como um facilitador com chamadas prontas, sem a necessidade de maiores implementações.

Por padrão na biblioteca do Arduino, encontra-se o utilitário `tone()` que permite a reprodução sonora de um tom em uma frequência especificada pelo usuário usando PWM e um alto-falante comum. Entretanto, não é possível determinar a intensidade do som. Em solução a este problema, a biblioteca *toneAC* permite definir tanto frequência como intensidade sonora, além de melhorar a durabilidade do alto-falante e a qualidade, utilizando uma estratégia de corrente alternada entre dois pinos de saída PWM [2].

3. Resultados

A aplicação da metodologia proposta retornou resultados, após uma sequência de testes até chegar ao resultado final, que foi o completo funcionamento do sistema. A seguir, estes são descritos e discutidos.

3.1. Utilização da Placa Arduino Uno

Objetivando testar cada elemento do circuito para determinar sua correta funcionalidade, bem como a implementação de código, testes rápidos foram efetuados fazendo uso da placa Arduino Uno.

O primeiro teste foi para verificar a saída obtida da leitura do sensor HC-SR04 usando a biblioteca *Ultrasonic*. Um rápido código fora implementado e observada a saída para quando o sensor era apontado para paredes e teto do espaço local e também para a distância da mão de um voluntário que se movimentava, averiguando o funcionamento esperado.

Depois, a funcionalidade `toneAC()` foi posta à prova para diferentes valores de frequência e intensidade escolhidos manualmente. Tendo verificado seu correto comportamento, a união de ambos os testes resultou no código final para ser embarcado (detalhado no Apêndice A), como pode ser visto *online* através deste vídeo: <https://youtu.be/5GCRpEJFvAo>.

3.2. Circuito e Elaboração para PCI

Tendo o entendimento de como deve operar o sistema e como seus componentes devem estar conectados, foi elaborado o circuito para o sistema sem a placa Arduino, ilustrado na Figura 2. A alimentação se dá pela utilização de uma bateria 9V, garantindo uma entrada CC estável. Atendendo ao padrão do microcontrolador, a entrada passa pelo regulador de tensão LM7805 para definir a tensão de entrada em 5V, necessário também para operação dos sensores. A configuração do oscilador é de acordo com o indicado na referência do

ATmega328P [3]. Por fim, a conexão dos sensores e do alto-falante é similar a quando se usou o Arduino.

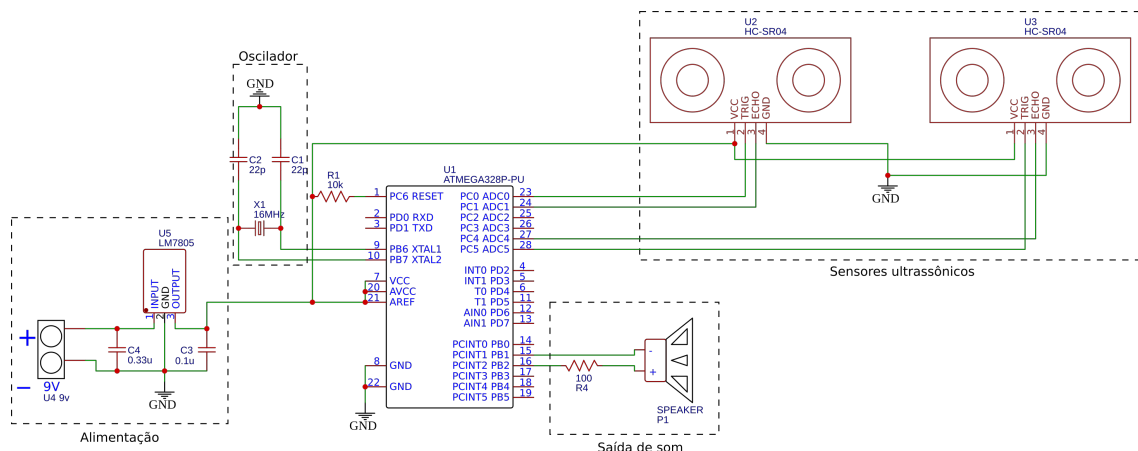


Figura 2. Esquemático do circuito para o Theremin digital.

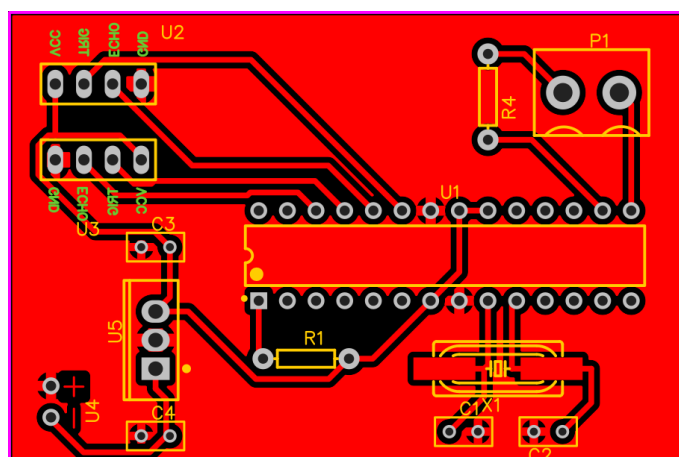


Figura 3. PCI resultante a partir do circuito ilustrado na Figura 2.

A partir do esquemático definido anteriormente, foi possível elaborar a máscara para confecção da PCI, como pode-se ver na Figura 3. A disposição dos elementos na placa permitiu organizar o sistema de forma que cada seção ficasse em uma de suas extremidades. No canto superior esquerdo, foram posicionados os sensores de forma que um fique oposto ao outro, o que causou um problema quando definido o plano terra, deixando uma parte isolada. Este problema foi resolvido conectando-se manualmente um cabo entre ambas as referências dos sensores (representadas por GND). No canto superior esquerdo, aparece a configuração para o alto-falante. Abaixo, à esquerda, os pontos para alimentação, e à direita, o cristal oscilador.

3.3. Sistema Embarcado Funcional

Utilizando o método fotográfico, foi possível elaborar a PCI usando a máscara anteriormente citada (Figura 3) com qualidade razoável, tendo por resultante o mostrado na Figura 4. A colocação manual dos componentes foi feita com soldador e estanho comuns, também com resultado razoável ilustrado na Figura 5.

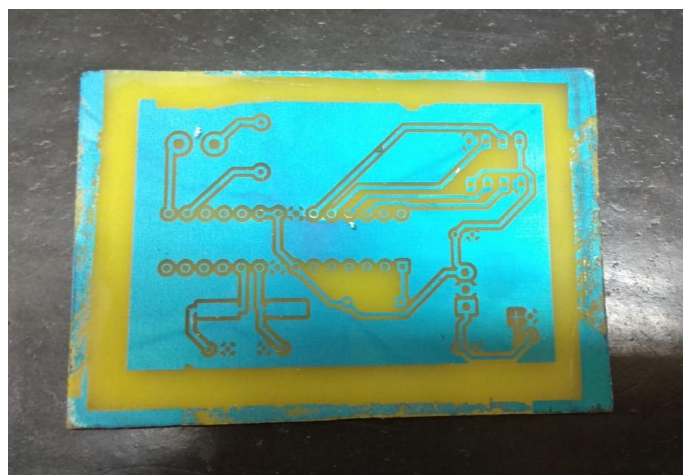


Figura 4. Resultado da aplicação do método fotográfico para confecção da PCI.

A gravação de código no microcontrolador funcionou corretamente, e a alimentação também. No entanto não foi possível reproduzir som a partir do alto-falante provavelmente devido a baixa corrente provida pela bateria (aproximadamente 15mA).

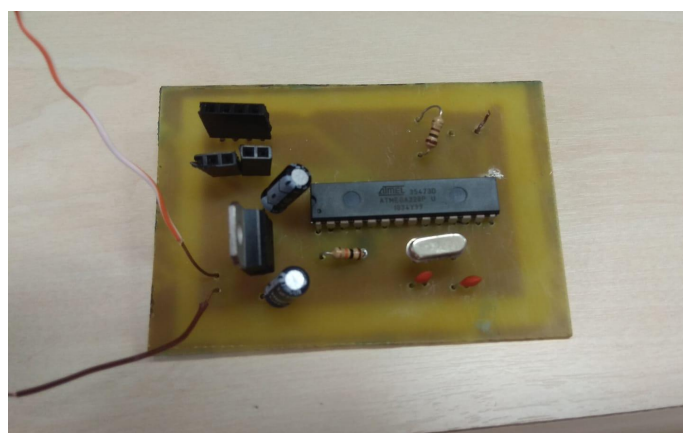


Figura 5. Sistema embarcado final com componentes soldados.

4. Conclusão

A proposta de desenvolvimento de um Theremin digital foi estabelecida, bem como a metodologia para tal, que foi aplicada resultando em um sistema embarcado. Uma PCI foi confeccionada artesanalmente com base em um circuito projetado, e o código para funcionamento do sistema foi desenvolvido. Embora o sistema não tenha funcionado de fato devido a pormenores, testes com a placa Arduino Uno confirmaram o correto comportamento conforme esperado. Como proposta de alteração e melhoria, poderia ter-se usado como alimentação uma fonte com maior capacidade de corrente, uma vez que a tensão era naturalmente regulada graças a implementação do circuito. Além disso, outros métodos podem ser aplicados para a confecção da PCI, desde projeto de máscara até métodos artesanais ou industriais.

Referências

- [1] ATMEL. **8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash**, fev. 2009.
- [2] T. Eckel. **toneAC Arduino Library for Arduino**. Disponível em <https://bitbucket.org/teckel12/arduino-toneac/wiki/Home>. Acesso em 20 out. 2019.
- [3] FEC. **Arduino Uno R3**. Disponível em https://www.fecegypt.com/uploads/dataSheet/1522237550_arduino%20uno%20r3.pdf. Acesso em 30 nov. 2019.
- [4] E. Freaks. **Ultrasonic Ranging Module HC-SR04**. Disponível em <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. Acesso em 30 nov. 2019.
- [5] A. Glinsky, B. Moog, and R. Moog. *Theremin: Ether Music and Espionage*. Music in American life. University of Illinois Press, 2000.
- [6] T. Holmes and S. Martin. **The 'Theremin' or 'Theremin-vox'**, out. 2015. Disponível em <http://120years.net/the-thereminleon-termensoviet-union1922/>. Acesso em 30 nov. 2019.
- [7] ISI-TICs. **Instrumentos Musicais Digitais**, jan. 2019. Disponível em <https://isitics.com/2019/01/15/instrumentos-musicais-digitais/>. Acesso em 30 nov. 2019.
- [8] I. Kipnis. *The Harpsichord and Clavichord: An Encyclopedia*. Encyclopedia of Keyboard Instruments. Taylor & Francis, 2013.
- [9] C. Nobile. **Ultrasonic.h - Library for HC-SR04 Ultrasonic Sensing Module**. Disponível em <https://github.com/filipeflop/Ultrasonic/blob/master/Ultrasonic.h>. Acesso em 20 out. 2019.
- [10] T. P. Vendruscolo. **Projeto e fabricação de placas de circuitos impressos**. Universidade Tecnológica Federal do Paraná, out. 2019.

A. Código Embarcado no Microcontrolador

O código implementado para ser embarcado no microcontrolador ATmega328 pode ser conferido na íntegra a seguir, implementado usando linguagem C.

```
1 //Carrega as bibliotecas de sensor ultrassonico e toneAC
2 #include "Ultrasonic.h"
3 #include "toneAC.h"
4
5 //Define os pinos para trigger e echo
6 int pino_trigger1 = A0;
7 int pino_echo1 = A1;
8 int pino_trigger2 = A5;
9 int pino_echo2 = A4;
10
11 //Inicializa os sensores nos pinos definidos acima
12 Ultrasonic ultrasonic1(pino_trigger1, pino_echo1);
13 Ultrasonic ultrasonic2(pino_trigger2, pino_echo2);
14
15 void setup() {}
16
17 void loop() {
18     //Le as informacoes dos sensores em cm
19     float cm1, cm2, nota, intensidade;
20     long microsec1 = ultrasonic1.timing();
21     long microsec2 = ultrasonic2.timing();
22     cm1 = ultrasonic1.convert(microsec1, Ultrasonic::CM);
23     cm2 = ultrasonic2.convert(microsec2, Ultrasonic::CM);
24
25     // Mapeia a leitura dos sensores entre 5cm e 30cm
26     // Notas vao de La2 ate La4 (2 oitavas)
27     nota = map(cm1, 5, 30, 220, 880);
28     // Intensidade varia entre minino e maximo possivel
29     intensidade = map(cm2, 5, 30, 1, 10);
30
31     toneAC(nota, intensidade);
32 }
```