**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - CAMPUS APUCARANA**

GABRIEL FINGER CONTE

GIAN LUCAS DOS REIS

RAFAEL ZAUPA WATANABE

PROJETO FINAL DE SISTEMAS DIGITAIS (SICO5A)

APUCARANA - PARANÁ

2022

1. **RESUMO**

O presente trabalho trata-se de um projeto desenvolvido na disciplina de Sistemas Digitais, com o intuito de explorar e aplicar os conceitos de processamento digital de sinais. Focou-se na conversão de um sinal analógico captado por um sensor de voz para um sinal digital, e na reprodução do mesmo após ser reconstruído. A inspiração desse projeto foram os gravadores de voz antigos, implementando três funções básicas: gravação de áudio, reprodução de áudio e uma opção para apagar o áudio armazenado.

1. **INTRODUÇÃO**

O presente trabalho é sobre a conversão Analógica para Digital e posteriormente Digital para Analógica.

Seu objetivo é basicamente utilizar os conhecimentos adquiridos nas aulas de Sistemas Digitais a respeito do conteúdo de Sinais Digitais e desenvolver algum projeto “aprovado pelo professor” a respeito do conteúdo.

Este trabalho está separado em Objetivos, Materiais e Esquemas, Ideia e Lógica, Descrição do Algoritmo, Design do circuito, Resultados obtidos e Conclusão.

A metodologia utilizada foi a pesquisa bibliográfica e artigos em sites como StackOverflow e a documentação do Arduino.

1. **OBJETIVOS**

* Utilizar o micro controlador ESP 32 para montar um dispositivo capaz de gravar, armazenar e reproduzir um sinal de áudio.
* Aplicar os conceitos aprendidos na disciplina sobre os conversores ADC e DAC e sobre o processamento digital de sinais.

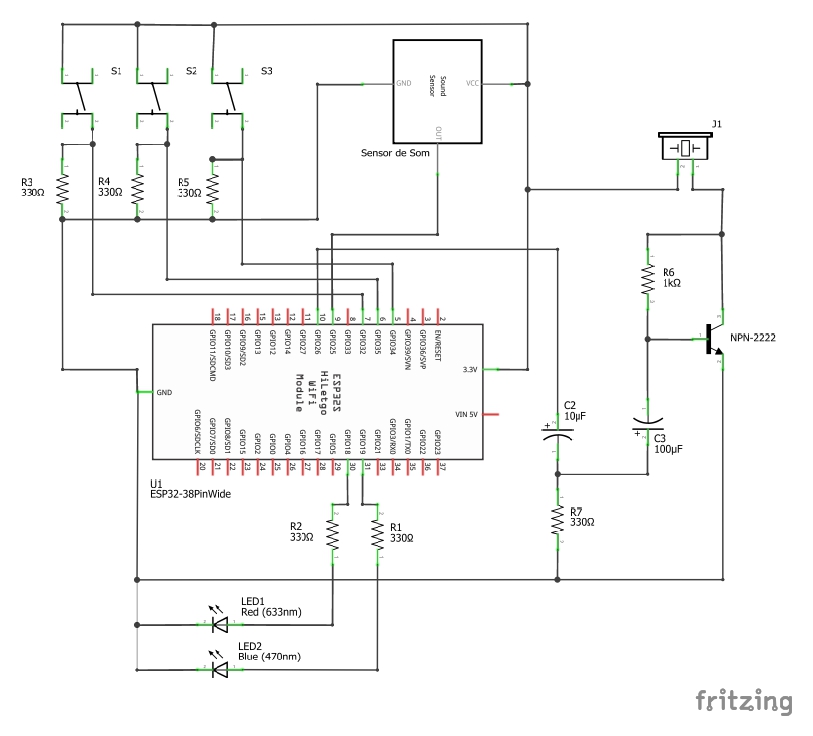
1. **MATERIAIS E ESQUEMAS**

Os materiais utilizados na construção do circuito projetado foram:

* 06 Resistores de 330Ω;
* 01 Resistor de 1 KΩ;
* 03 Push-Buttons;
* 01 LED Vermelho;
* 01 LED Azul;
* 01 Microcontrolador ESP 32;
* 01 Sensor de Som com Comparador LM393;
* 01 Auto falante de 8 Ω e 1.0 W;
* 01 Cabo Micro USB;
* 01 Capacitor eletrolítico de 100 µF;
* 01 Capacitor eletrolítico de 10 µF;
* 01 Transistor NPN-2222;
* Jumpers Macho-Macho e Macho-Fêmea.

Após o desenvolvimento e refinamento do projeto, o seguinte esquemático, desenvolvido no software Fritzing, ilustra as ligações entre os componentes utilizados na versão final do projeto:

Figura 01 - Esquemático das ligações do circuito desenvolvido



Fonte: autoria própria

1. **IDEIA E LÓGICA**

A inspiração para a montagem e definição do comportamento do projeto desenvolvido, veio a partir da associação do tema proposto pelo professor, sendo esse para desenvolver um conversor AD/DA para um alto falante. Tomou-se referência nos gravadores de áudio antigo para desenvolver a ideia, como pode ser visto na figura abaixo:

Figura 02- Gravador de voz antigo

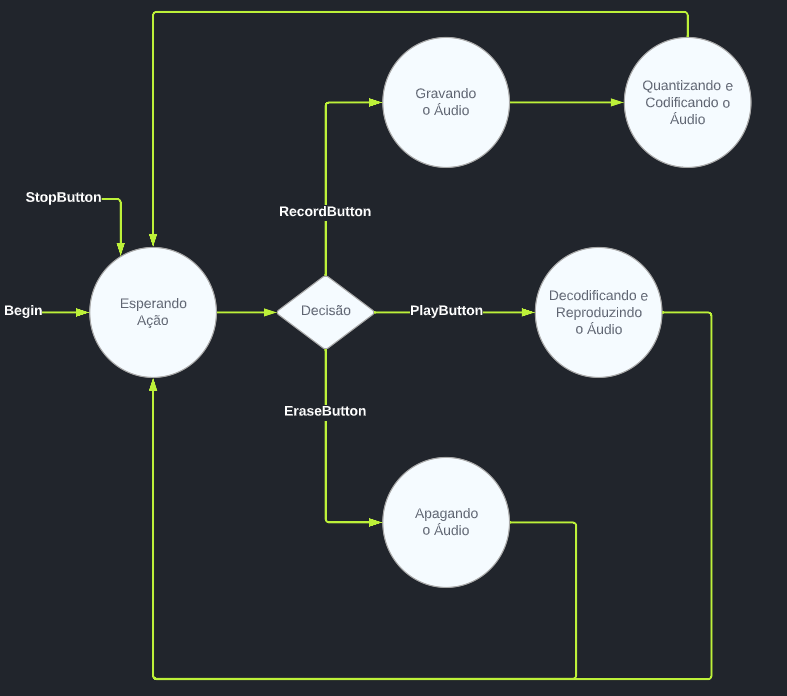


Fonte: https://www.esijmjg.com.br/2019/01/mini-micro-gravador-cassete-k7-sony-m.html

A partir disso, pensou-se na configuração do conversor como uma máquina de estados, onde fica-se esperando alguma entrada do usuário para informar para qual dos três estados se deseja ir:

* Gravação: onde será gravado um áudio de 5 segundos;
* Limpeza: onde será apagado qualquer dado que esteja na memória;
* Reprodução: onde será reproduzido o áudio gravado.

Figura 03- Gravador de voz antigo



Fonte: autoria própria

A partir da máquina de estados pensada, como pode ser observada na figura acima, desenvolveu-se o algoritmo de modo a refletir tal comportamento.

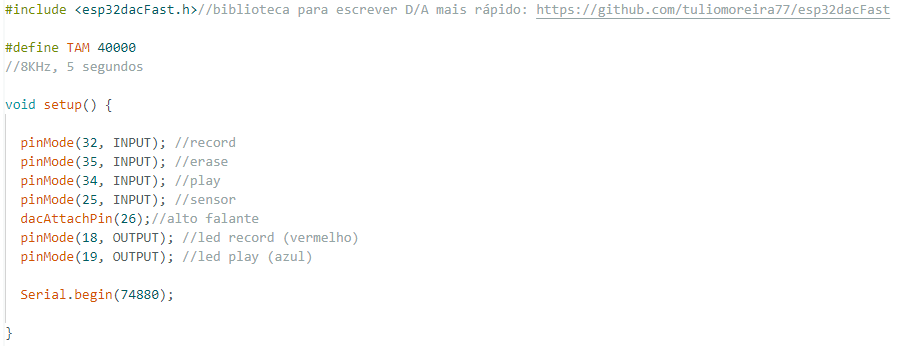
1. **DESCRIÇÃO DO ALGORITMO**

O algoritmo construído na ferramenta de desenvolvimento Arduino IDE tem como objetivo, em suma, atribuir a um vetor de tamanho pré determinado pela frequência desejada e quantidade de tempo da amostra os valores lidos pelo sensor analógico (nesse caso, o microfone). Nesse vetor de dados é possível, então, trabalhar com as informações coletadas e realizar a conversão A/D e posteriormente D/A.

Explicações mais minuciosas sobre a utilidade lógica de cada parte do código estão presentes em formato de comentário no próprio código, e esta seção do relatório serve para explicar os objetivos de cada trecho de maneira mais sucinta, mas permitindo total entendimento.

Como de costume no software utilizado, a primeira parte dedica-se a declarar os pinos de entrada e saída em referência à numeração escolhida no ESP32, logo depois de ser definido o tamanho do vetor que guardará as informações (multiplicando-se 8000Hz por 5s, obtém-se 40000 espaços de amostra necessários) e da importação da biblioteca utilizada para que o processo de leitura analógica ocorra de maneira mais rápida.

Figura 04 - Inputs e outputs no código



Fonte: autoria própria

O vetor é posteriormente criado, com o tamanho anteriormente inferido, bem como uma variável que permite a escolha dos estados atuais; dependendo do botão pressionado, o valor dessa variável mudará, possibilitando que o circuito realize as diferentes funções de gravar, apagar e tocar.

Figura 05 - Variáveis utilizadas

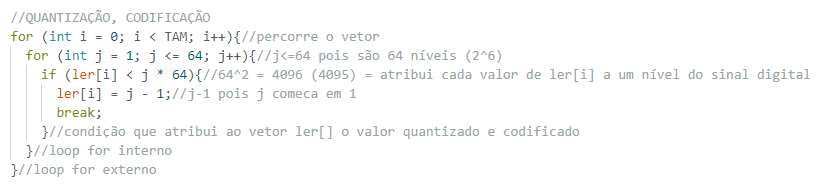


Fonte: autoria própria

Após condições que atribuem o valor desejado à variável auxiliar, esta é utilizada em um switch case onde a opção 1 é gravar, a opção 2 é apagar e a opção 3 é tocar. A luz vermelha acende durante a gravação, e a azul acende enquanto o áudio é tocado.

O caso “gravar” é onde ocorre a atribuição das informações recebidas analogicamente pelo microfone a posições do vetor criado, com um delay que é de natureza obrigatória. Também é quando ocorre a quantização e quantificação, já que as informações do áudio são armazenadas digitalmente; dessa maneira, é onde pode-se dizer que ocorre, de fato, a conversão analógica-digital. Foram utilizados 64 níveis de quantização, já que foram utilizados 6 bits de informação, e 26 = 64. Dentro do laço de repetição que percorre e divide o vetor, há a codificação para os intervalos desejados, isto é, cada valor analógico é atribuído a um dos níveis de quantização para formar o sinal digital.

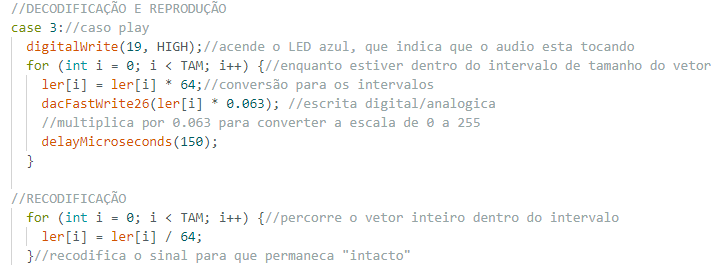
Figura 06 - Quantização e codificação em “gravar”



Fonte: autoria própria

No caso “apagar”, há somente o percorrimento do vetor, sendo zerada cada posição de maneira a apagar o conteúdo. Já no caso “tocar”, é necessário fazer o processo inverso da conversão realizada; pode-se dizer que é onde ocorre a conversão digital-analógica. O sinal, agora digital, tem cada um dos pontos (posições do vetor) multiplicado por 64 e subsequentemente convertido em valores que fazem jus à proporção da escala de 0 a 255 (nesse caso, sendo então multiplicados por 0.063). A última etapa deste caso é a decodificação do sinal, onde este é reconvertido, dividindo cada posição por 64 de maneira a manter os valores dentro do vetor intactos.

Figura 07 - Decodificação, reprodução e recodificação em “tocar”



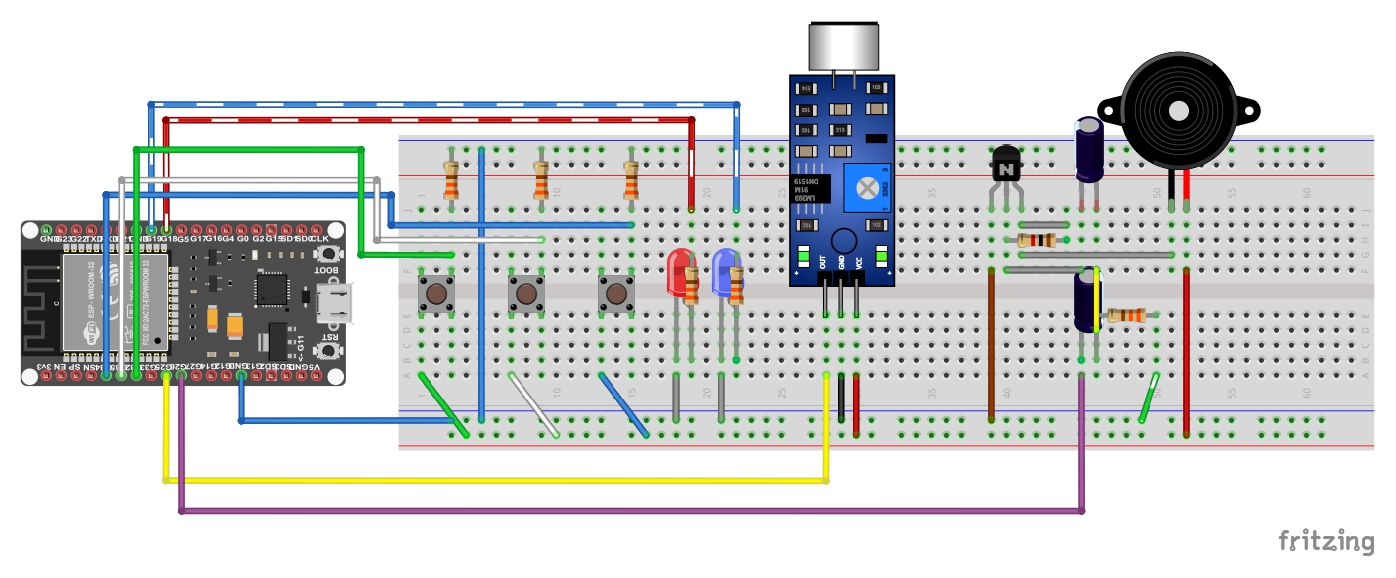
Fonte: autoria própria

É aqui que entra a função importada dacFastWrite26, disponibilizada por tuliomoreira77 (2021) em seu GitHub, a qual demanda a conversão para a proporção de 0 a 255; essa função opera de maneira a dividir os 3.3V que o ESP32 utiliza em 256 intervalos, ajustando as tensões de saída do PWM para os valores de intervalo desejados.

1. **DESIGN DO CIRCUITO**

Conforme está descrito no esquemático da Figura XX, montou-se o circuito obtendo algo semelhante a seguinte simulação construída no software Fritzing:

Figura 08- Simulação do circuito desenvolvido



Fonte: autoria própria

Para tanto, montou-se os botões utilizados para controlar a gravação, outro para apagar o áudio salvo e outro para tocar o som gravado. Cada um dos botões conta com um resistor de pull-down de 330Ω para garantir que quando não ativados permaneçam com tensão nula, ou seja, para evitar alta-impedância.

Os LEDs contam com um resistor de 330Ω para limitar a tensão vinda do microcontrolador para evitar danificar tais componentes.

A saída do som armazenado na ESP 32 é passada por um filtro de reconstrução passa-baixas, a fim de suavizar a saída do sinal digital, filtrando os componentes de alta frequência.

A saída do filtro de reconstrução é passada para um amplificador de áudio de Classe-A montado segundo apresentado por Mais Projetos Bruno xavier (2020), a fim de amplificar o sinal de áudio filtrado para que possa a saída do alto falante possa ser melhor analisado.

Por fim, para montar o seguinte circuito utilizou-se a seguinte configuração de pinagem na ESP 32:

Tabela 01- Mapeamento dos pinos para cada componente

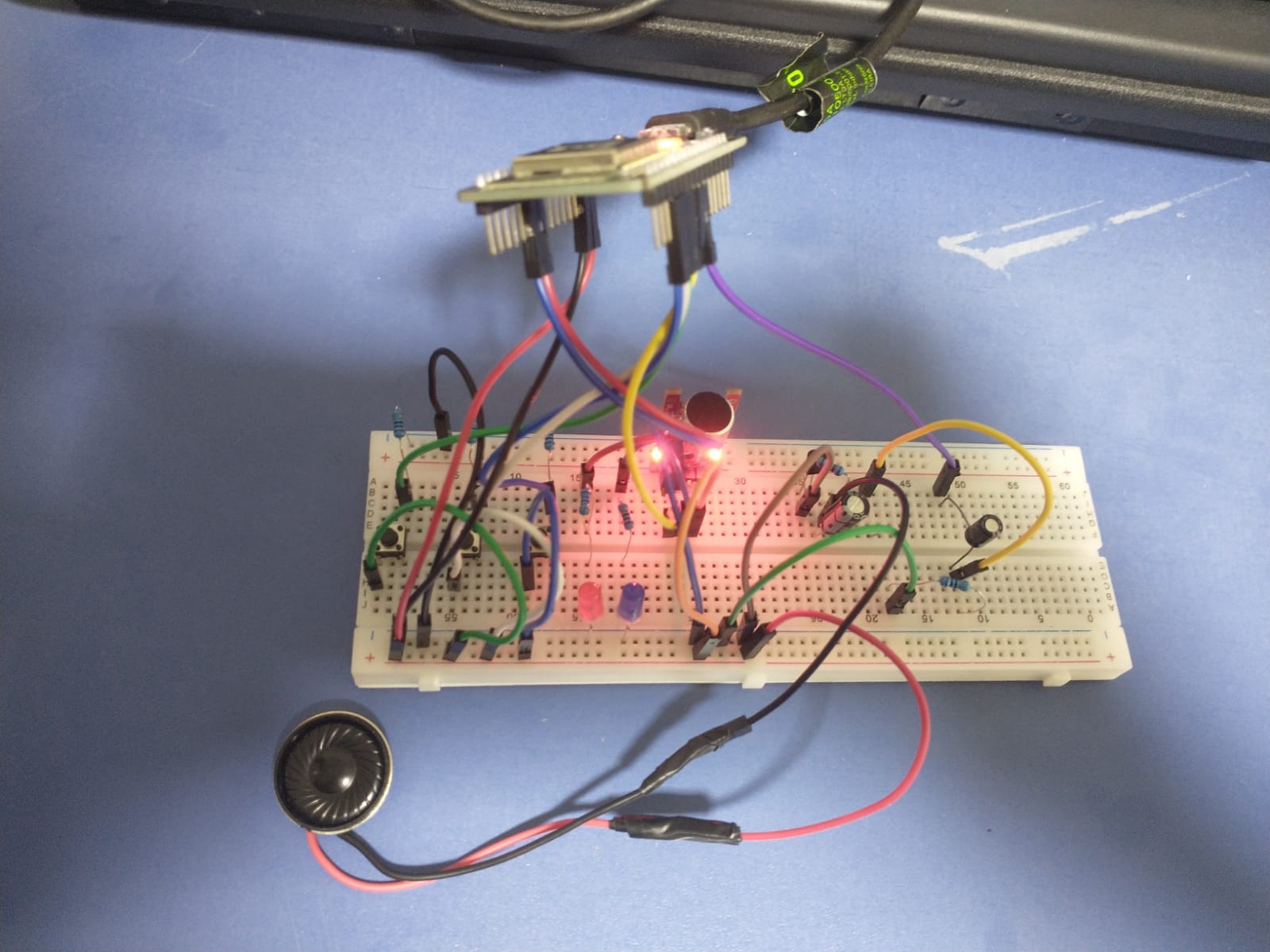
| Elemento | Tipo | Pino |
| --- | --- | --- |
| Record Button | INPUT | 32 |
| Erase Button | INPUT | 35 |
| Play Button | INPUT | 34 |
| Sensor Ruído | INPUT | 25 |
| Alto Falante | OUTPUT | 26 |
| Led Vermelho(Record) | OUTPUT | 18 |
| Led Azul(Play) | OUTPUT | 19 |

Fonte: autoria própria

1. **RESULTADOS OBTIDOS**

Após o desenvolvimento da lógica do algoritmo, seguindo o esquema de conexões da Figura X, montou-se o circuito em uma placa de prototipagem, obtendo o seguinte resultado.

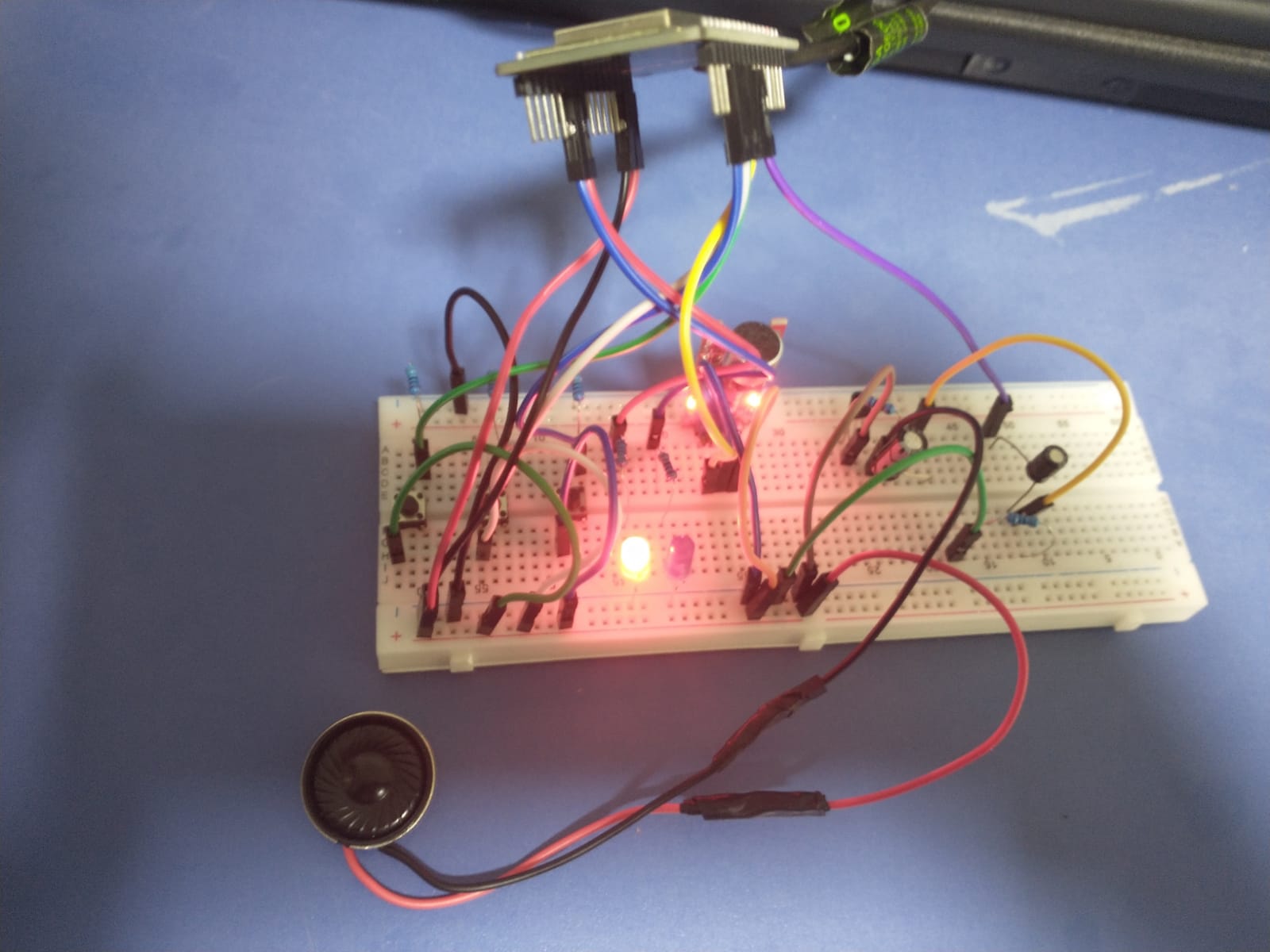
Figura 09 - Circuito implementado



Fonte: autoria própria

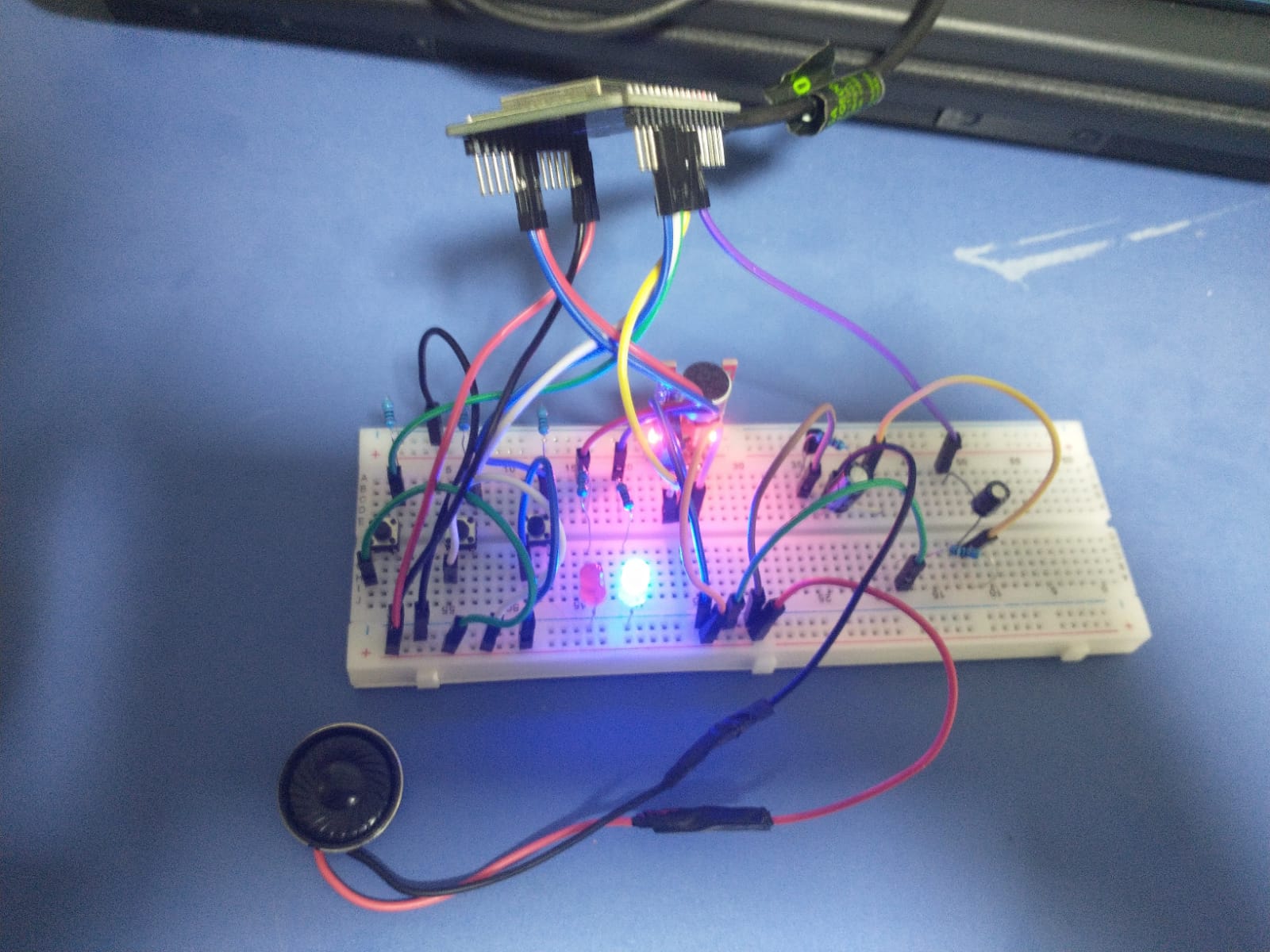
Aplicando o código desenvolvido com base na lógica apresentada anteriormente, verificou-se o correto funcionamento do dispositivo, conforme pode ser observado nas figuras abaixo, onde os LEDS que indicam o estado do dispositivo acenderam corretamente:

Figura XX - Circuito implementado durante o processo de gravação do áudio



Fonte: autoria própria

Figura XX - Circuito implementado durante o processo de reprodução do áudio



Fonte: autoria própria

Audivelmente, constatou-se o correto funcionamento do dispositivo e programa, sem levar em consideração a qualidade do som emitido. Isso pois, devido às limitações tanto do microfone e microcontrolador como a ideia proposta para o projeto era de implementar uma conversão AD/DA considerando menos de 8 bits para representar o sinal digital.

1. **CONCLUSÃO**

Após a pesquisa e realização do presente projeto conclui-se que o projeto foi desenvolvido de modo a cumprir os objetivos propostos, sendo possível vivenciar os conceitos de processamento digital de sinais. Além do dispositivo e lógica desenvolvido terem funcionado conforme o esperado, executando corretamente a gravação e reprodução de um áudio de 5 segundos, e apagando o mesmo da memória quando solicitado.

1. **REFERÊNCIAS BIBLIOGRÁFICAS**

Mais Projetos Bruno xavier. Como montar amplificador caseiro com qualquer transistor NPN. YouTube, 31 de jan. de 2020. Disponível em: <https://www.youtube.com/watch?v=P6AfX4ONVkg&t=196s>. Acesso em: 12 de dez. de 2022.

tuliomoreira77. esp32dacFast. GitHub, 16 de ago. de 2021. Disponível em: <https://github.com/tuliomoreira77/esp32dacFast>. Acesso em: 13 de dez. de 2022.