

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO – IFSP

BRUNO VINICIUS LIMA DA SILVA  
CÂNDIDA ROSA PARAIZO  
MARCOS JUDENSAIDER KNIJNIK

Sistema de aquisição de sinais utilizando o microcontrolador ESP 32

SÃO PAULO  
2023

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO – IFSP

BRUNO VINICIUS LIMA DA SILVA  
CÂNDIDA ROSA PARAIZO  
MARCOS JUDENSAIDER KNIJNIK

Sistema de aquisição de sinais utilizando o microcontrolador ESP 32

Monografia apresentada ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP, Câmpus São Paulo, como parte dos requisitos para a obtenção da aprovação na disciplina de Projeto Integrado.

Orientadores:

Prof. Gustavo Neves Margarido

Prof. Rodrigo Rech

Prof. Tarcísio Fernandes Leão

SÃO PAULO  
2023

## **DEDICATÓRIA**

Dedicamos este trabalho aos alunos, professores e a todos aqueles a quem esse projeto possa ajudar, seja em aperfeiçoamento de conhecimentos ou auxílio em aplicações de eletrônica e automação em geral.

## **AGRADECIMENTOS**

Aos professores, por todas as orientações nos momentos de dúvidas. Aos colegas de classe, por compartilhar conhecimento e experiências que ajudaram na realização deste trabalho. Aos pais dos alunos envolvidos, por incentivar a resiliência ao longo todas as etapas desta jornada.

## **LISTA DE SÍMBOLOS E SIGLAS**

ESP 32 - Espressif Systems 32 bits

IFSP - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

IDE - Integrated Development Environment

## LISTA DE FIGURAS

Figura 1. Exemplo do layout.....	<b>Erro! Indicador não definido.</b>
Figura 2. Circuito de redução e linearização da tensão lida .....	2
Figura 3. Tratamento de sinais.....	3

## RESUMO

O osciloscópio digital de baixo custo foi constituído para aprimorar ainda mais o conhecimento dos estudantes e profissionais que vão utilizá-lo. Nesse Projeto usamos um microcontrolador chamado ESP 32, esse componente tem diversos recursos onde conseguimos aplicar nossos conhecimentos para o desenvolvimento do projeto. Usamos uma entrada analógica do ESP 32 para coletar o sinal lido e a partir disso geramos um sinal elétrico em formato de ondas, que são elas: senoidal, triangular e quadrada. Para a visualização desses sinais utilizamos um recurso do ESP 32 para ilustrar no navegador os formatos de onda que foram lidos. Com isso podemos acessar uma rede Wireless que é gerada pelo próprio ESP 32, essa rede pode ser acessada por qualquer dispositivo que tenha Android ou qualquer computador com acesso a rede Wireless, ao entrar na rede é necessário abrir um navegador que irá direcionar o usuário direto para o site onde será mostrado os sinais lidos, criamos um site com sua interface de fácil acesso para o usuário que irá visualizar todos os dados necessário que envolvem o sinal lido, alguns exemplos desses dados são: tensão, período e frequência da onda. Para a aquisição desses sinais, foi desenvolvido uma placa com circuitos eletrônicos que adaptam o valor de tensão lido para a tensão que o ESP 32 consegue trabalhar de 0 até 3,3 volts. devido ao seu baixo custo cada aluno ou profissional poderá construir o seu para uso pessoal em pesquisas. Uma das suas características notáveis é o peso que tem aproximadamente 500 g, isso facilitará no seu manuseio evitando qualquer tipo de risco ergonômico para seu portador. O método de pesquisa desse trabalho foi totalmente experimental, onde foi possível obter resultados satisfatórios que tornaram o projeto promissor.

Palavras-Chave: osciloscópio, Projeto, Portátil.

## **ABSTRACT**

The low-cost digital oscilloscope was created to further improve the knowledge of students and professionals who will use it. In this Project we use a microcontroller called ESP 32, this component has several resources where we can apply our knowledge to the development of the project. We use an analog input of the ESP 32 to collect the read signal and from that we generate an electrical signal in the form of waves, which are: sinusoidal, triangular, and square. To visualize these signals, we used an ESP 32 resource to illustrate the waveforms that were read in the navigator. With this we can access a Wireless network that is generated by the ESP 32 itself, this network can be accessed by any device that has Android or any computer with access to the Wireless network, when entering the network it is necessary to open a browser that will direct the user directly for the website where the read signals will be shown, we created a website with its user-friendly interface that will visualize all the necessary data involving the read signal, some examples of these data are: voltage, period and frequency of the wave. For the acquisition of these signals, a board was developed with electronic circuits that adapt the voltage value read to the voltage that the ESP 32 can work from 0 to 3.3 volts. due to its low cost, each student or professional can build their own for personal use in research. One of its notable features is its weight of approximately 500 g, which will facilitate its handling, avoiding any type of ergonomic risk for its user. The research method of this work was completely experimental, where it was possible to obtain satisfactory results that made the project promising.

Keywords: oscilloscope, Project, Portable.



# SUMÁRIO

1.	INTRODUÇÃO .....	1
1.1	Objetivo .....	1
1.2	Justificativa .....	1
2.	REVISÃO DA LITERATURA .....	3
3.	DESENVOLVIMENTO – MATERIAIS E MÉTODOS .....	4
3.1	Descrição .....	6
3.2	Diagrama de Blocos Funcionais .....	6
3.3	Cronograma e responsabilidades .....	6
3.4	Lista de materiais.....	7
3.5	Orçamento .....	7
4	RESULTADOS .....	8
5	DISCUSSÃO .....	16
5.1	Lições Aprendidas.....	17
6	CONCLUSÃO .....	19
	REFERÊNCIAS.....	20
	APÊNDICE A – PROGRAMAS COMPUTACIONAIS .....	29
	APÊNDICE B – MANUAL DO USUÁRIO .....	66

# **1. INTRODUÇÃO**

O osciloscópio digital é uma ferramenta para visualizar um sinal elétrico na forma de um gráfico, ou seja, uma imagem em duas dimensões, na qual o eixo x é o tempo e o eixo y, a amplitude em volts. Observando o formato de onda gerado, analisa-se propriedades como amplitude, frequência, RISE TIME (tempo de alteração do valor do sinal), DUTY CYCLE, intervalo de tempo e distorção. São instrumentos muito utilizados em laboratórios de pesquisas, medicina, engenharia, indústria automotiva, sistemas de telecomunicações e manutenção de equipamentos eletrônicos.

O custo de aquisição de um osciloscópio digital varia muito de quais elementos ele possui, como, quantidade de canais, faixa de frequência ou largura de banda na qual o aparelho funciona, podendo fazer o preço ultrapassar 10 mil reais. Esse é um dos maiores motivos para que estudantes e/ou instituições de ensino desistam de adquiri-lo. Além disso, a portabilidade do mesmo não está entre suas principais vantagens, já que é um aparelho pesado, robusto e espaçoso.

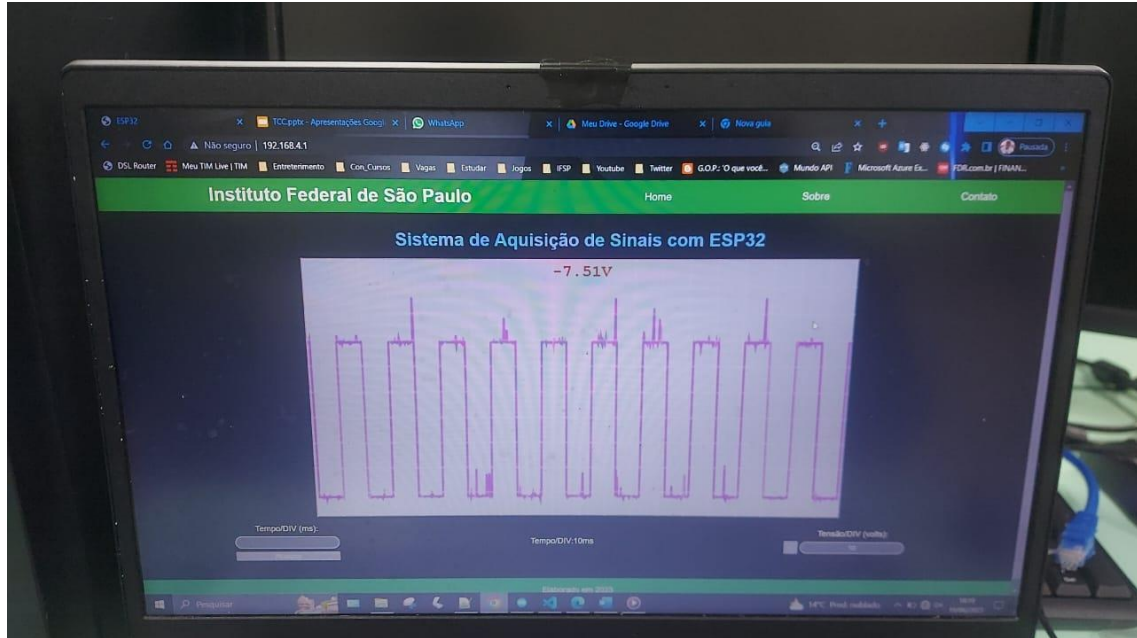
## **1.1 Objetivo**

Desenvolver um osciloscópio digital com base no microcontrolador ESP32, para que o custo e a portabilidade não seja um empecilho para estudantes que desejam aprimorar seus conhecimentos.

## **1.2 Justificativa**

Em muitas escolas de cursos técnicos e superiores, diversos alunos não conseguem estudar fora das salas ou laboratórios de aula, ou aprofundar seus estudos de forma prática, por não termos condições de utilizar os equipamentos necessários. Atacando a essa dor, esse projeto visa ajudar alunos que têm interesse em realizar atividades com osciloscópios ou

gerador de sinais, orientando para uma fácil montagem, com baixo custo e que possa ser aplicada em múltiplas áreas, como por exemplo: eletrônica, elétrica e automação residencial.



**Figura 1 Exemplo de layout**

## **2. REVISÃO DA LITERATURA**

Osciloscópio como instrumento de medida de sinais elétricos e sonoros. Apresenta-se a importância desse equipamento na descrição e análise de sistemas estudados na Eletrônica e na Física, discutindo aspectos históricos, de operacionalidade, e de distinção existente entre os instrumentos analógicos e digitais. Além disso, aponta-se no estudo algumas alternativas, já presentes na literatura, aos osciloscópios convencionais, comumente utilizados em instituições de ensino superior. Por meio desse estudo foi possível observar que, atualmente, existem uma gama de possibilidades de instrumentos capazes de analisar sinais elétricos e sonoros. Entende-se que um conhecimento maior sobre estes aparatos pode ser uma forma de melhorar a acessibilidade quanto aos estudos experimentais de fenômenos elétricos e sonoros.

### **3. DESENVOLVIMENTO – MATERIAIS E MÉTODOS**

Para atingir os objetivos propostos e testar as hipóteses experimentais formuladas, realizou – se um estudo teórico e prático, onde o estudo teórico foi dividido em duas fases e o estudo prático foi realizado em apenas uma etapa.

A primeira fase do estudo teórico constituiu na pesquisa para utilizar métodos de testes, onde seria possível gerar diferentes tipos de sinais na interface de visualização, como por exemplo sinais senoidais, quadrados e triangulares.

A segunda fase do estudo teórico teve como objetivo analisar códigos de interfaces já existentes de projetos com propostas bem parecidas, além de verificar as variáveis que capturam os dados, os manipulam e os apresentam na tela.

Na etapa do estudo prático foi taxado alguns objetivos, como analisar e testar as entradas e saídas analógicas e digitais do ESP32. Incluir também na análise DAC e ADC.

#### **Delineamento da Pesquisa**

Nessa Etapa do artigo, foi utilizado um tipo de pesquisa experimental para efetuar as pesquisas e realizar os testes práticos.

Para o desenvolvimento desse processo foi necessário criar uma organização das informações que foram tratadas, então foi criado um plano de ordem cronológica.

Esse plano consiste em coletar dados referentes as três fases desse tópico, foi feito uma pesquisa sobre como gerar sinais senoidais e outros tipos de ondas eletromagnéticas, houve uma preocupação quanto a essa parte da pesquisa, por que como foi dito anteriormente, foi seguido uma ordem cronológica, se algumas dessas etapas não forem alcançadas, todo o projeto poderia ser comprometido.

Após a pesquisa foi visto que era possível realizar a geração desses sinais, com a utilização do controlador ESP32, ou seja, o controlador que será utilizado no projeto irá atender a necessidade.

Dando seguimento a pesquisa, foi feito uma análise de métodos para gerar uma interface de visualização para o usuário, essa pesquisa indicou diversos tipos de softwares que possibilitaram a criação das interfaces, foi utilizado um critério de custo e praticidade para fazer a escolha do software que seria utilizado.

Na última etapa dessa pesquisa foi realizado o teste experimental utilizando o controlado ESP32.

Nessa etapa do processo analisamos todos os tipos de entradas e saídas que esse controlador contém, após isso iniciamos uma sequência de testes para avaliar seu comportamento quanto ao recebimento de sinais e envio deles, tivemos êxito em todos os testes realizados, o projeto se consolidou promissor após a realização desses métodos

## Procedimento Específicos

Para geração de sinais elétricos no Esp32 será necessário utilizar alguns componentes eletroeletrônicos para atenuar alguns sinais quando necessários, como resistores, capacitores e alguns diodos.

O formato de onda senoidal gerou uma preocupação porque dentre os outros sinais existentes, ele é o mais difícil para ilustrar, será necessário modular um sinal que será recebido numa entrada PWM do controlador.

Ao receber esse sinal no controlador devemos modificar ele, devido a esse empecilho houve a necessidade de usar uma função de DAC, que faz a conversão de um sinal digital para analógico, então entre um intervalo de tempo iremos modular um sinal de entrada com um valor inicial de 0% seguindo até 100%, isso ocorrerá em 60 vezes por segundo.

No desenvolvimento das interfaces gráficas para a visualização dos usuários, foi necessário criar um layout contendo informações de frequência, tensão e resistência. Esses valores serão lidos e alterados pelo usuário podendo modificar escalas para melhorar a visibilidade desses sinais.

A comunicação com a interface e o controlador será feito pelo modo rs232 modbus, esse módulo de rede permite a comunicação serial entre o hardware e software que utilizaremos.

Os testes com o ESP32 foram executados numa bancada, com a utilização de fontes dc e chaves de contato seco, para gerar variáveis nas entradas do controlador, então tivemos uma sequência de testes gerando pulsos em todas as entradas do controlador e cada entrada era processada e gerava um pulso nas saídas, esse foi o método utilizado para testar todas as funções que a placa tem.

### **3.1 Descrição**

Sistema de aquisição de sinais utilizando o microcontrolador ESP 32 para leitura de tensões entre -10V até +10V

### **3.2 Diagrama de Blocos Funcionais**

Descrição do equipamento/projeto em geral, considerando seus principais aspectos no formato de blocos.

### **3.3 Cronograma e responsabilidades**

Bruno:

Desenvolver os descritivos que compõem o projeto e auxiliar na montagem da placa.

Cândida:

Desenvolver a programação do ESP 32 e Layout de visualização dos sinais.

Marcos:

Desenvolver o projeto elétrico da placa, montagem da placa e auxiliar no desenvolvimento da programação.

### **3.4 Lista de materiais**

ESP 32 - 1

Módulo de alteração de tensão - 1

Placa padrão tipo ilha 10x10 – 1 unidade

Estanho - 1 metro

Amp-op LM348 - 3 unidades

A0512SDL – 1 unidade

Resistores - 7

Cabo Rígido – 8 vias

STEP UP MT3608 - 1

### **3.5 Orçamento**

ESP 32 - \$50,60

Módulo de alteração de tensão - \$42,65

Placa padrão tipo ilha 10x10 - \$17,01

Estanho – \$27,22

Amp-op LM348 - \$1,79

A0512SDL - \$31,00

Resistores - \$3,00

Capacitores - \$3,00

Cabo Rígido – \$1,72

STEP UP MT3608 - \$17,01

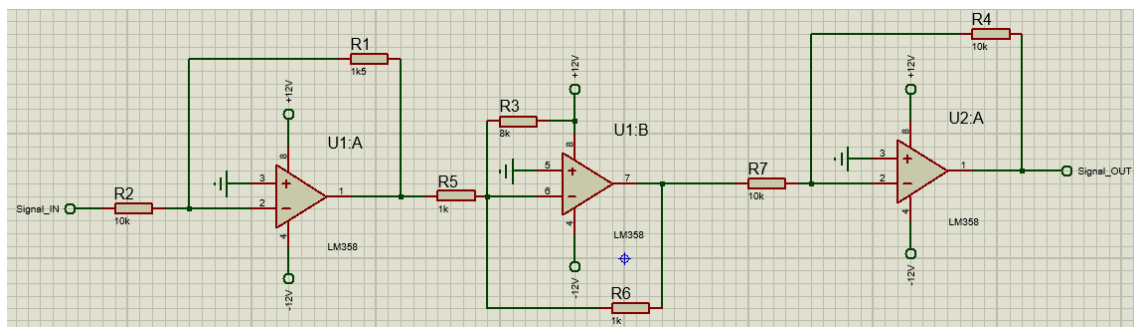
Total: \$ 195,00.



## 4 RESULTADOS

Para permitir a leitura de sinais entre  $\pm 10V$  pelo ESP32, que em seu pino aceita valores de tensão apenas dentro da faixa de 0 e 3,3V, foi desenvolvido um circuito eletroeletrônico para tratamento de sinais. Esse circuito teria como função transformar um sinal de  $\pm 10V$  em um sinal com características idênticas, porém em uma faixa de 0 a 3V (deixando assim 0,3V de faixa de segurança, para diminuir as chances de uma tensão maior a 3,3V que possa danificar o ESP32).

O desenvolvimento do projeto desse circuito foi feito utilizando circuitos com amplificadores operacionais. Para auxiliar no desenvolvimento e teste do projeto, foi utilizado o software Proteus (Proteus 8 Demonstration). O projeto final do circuito ficou conforme figura abaixo:



**Figura 2 Circuito de redução e linearização da tensão lida**

Utilizando a função de simulação do Proteus, foi possível realizar a leitura do sinal tratado e do sinal de entrada em um osciloscópio, o que resultou no exposto na figura a seguir:

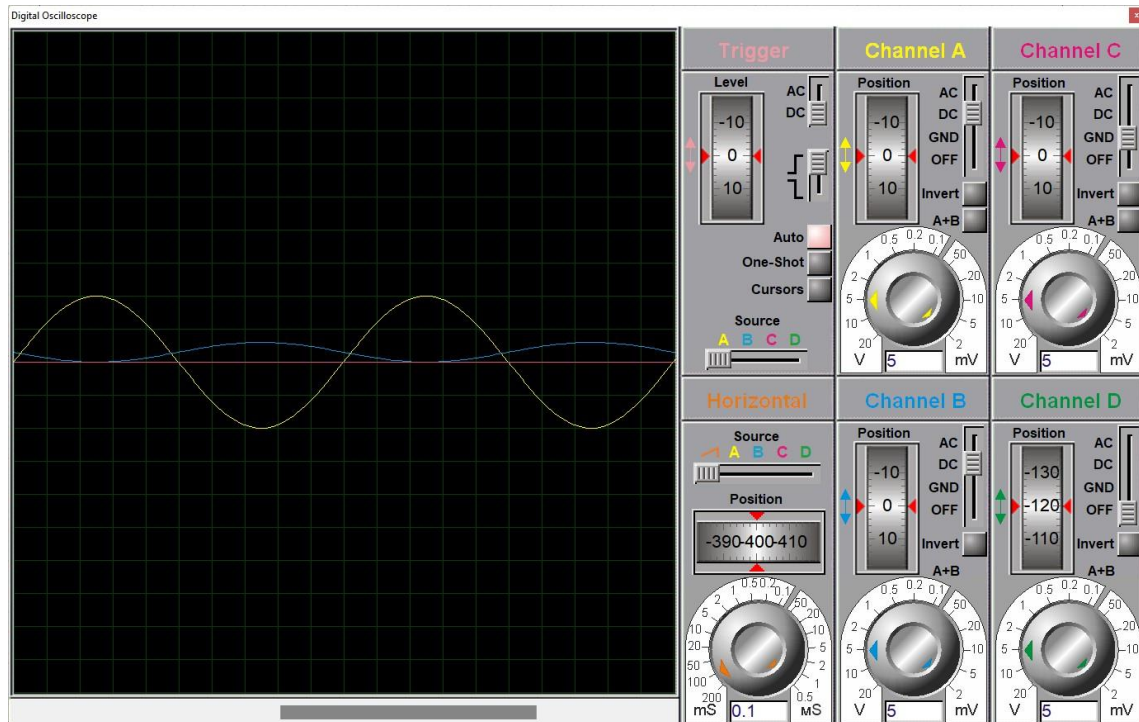


Figura 3 - Print da tela do osciloscópio digital do Proteus

No Canal A, temos o sinal de entrada, que neste caso tratou-se de uma onda senoidal de 10V a -10V. E no canal B temos a onda de saída, tratada pelo circuito, que será então lida pelo ESP32. Como percebe-se, a onda de saída manteve as exatas mesmas características da onda da entrada, porém as tensões foram diminuídas para uma faixa entre 0 e 3V, e a onda também foi invertida.

A onda tratada então será lida pelo ESP32, que além de capturar os valores de tensão, deve convertê-los matematicamente para recuperar a onda que originalmente desejou-se ler. Para tanto, foi elaborado uma equação matemática, exposta a seguir:

$$medicaoReal = -\left(valorMedido * \frac{20}{3} - 10\right)$$

Onde *valorMedido* é o valor de tensão entre 0 e 3V lido pelo pino do ADC do ESP32, e *medicaoReal* representa o valor original da onda que se deseja ler.

A partir dessa equação, foi possível desenvolver uma tabela de referência para comparação entre os valores lidos pelo ESP32 e os valores originalmente inseridos na placa, tabela esta que foi utilizada frequentemente na parte de testes dos equipamentos. A tabela é exibida na figura a seguir:

valorMedido	valorReal
3	-10,0
2,9	-9,3
2,8	-8,7
2,7	-8,0
2,6	-7,3
2,5	-6,7
2,4	-6,0
2,3	-5,3
2,2	-4,7
2,1	-4,0
2	-3,3
1,9	-2,7
1,8	-2,0
1,7	-1,3
1,6	-0,7
1,5	0,0
1,4	0,7
1,3	1,3
1,2	2,0
1,1	2,7
1	3,3
0,9	4,0
0,8	4,7
0,7	5,3
0,6	6,0
0,5	6,7
0,4	7,3
0,3	8,0
0,2	8,7
0,1	9,3
0	10,0

Figura 4 - Tabela de conversão entre valores tratados e valores originais

Para a utilização dos amplificadores operacionais, fazia-se necessário ainda a produção de uma alimentação simétrica para os mesmos.

A primeira solução desenvolvida foi a produção de uma fonte de alimentação DC simétrica de  $\pm 12V$ . Para tanto, foi seguido o projeto de Newton C. Braga (2018), conforme figura a seguir:

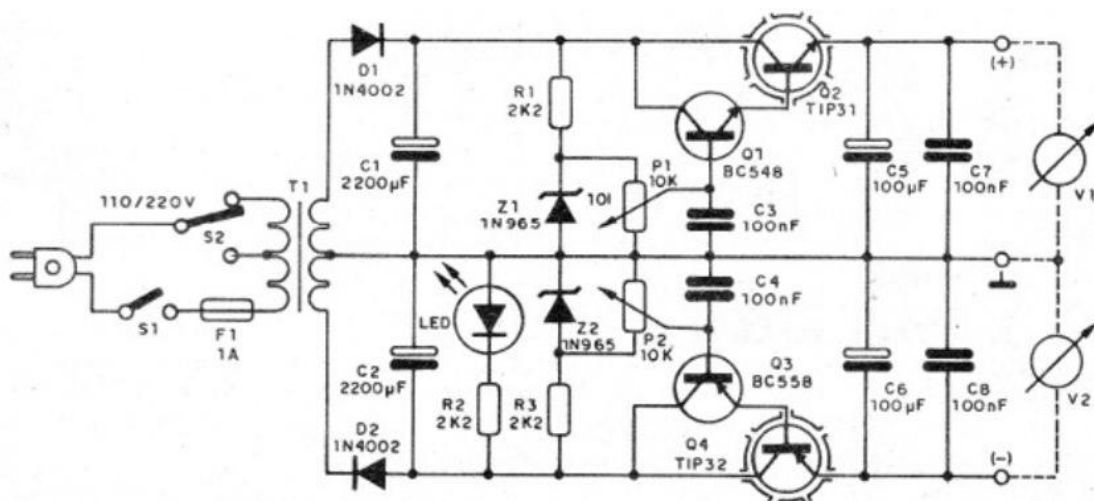


Figura 5 - Esquemático de fonte de alimentação simétrica de  $\pm 12V$

O circuito foi montado em uma placa de circuito impresso padrão tipo ilha, conforme figura a seguir:

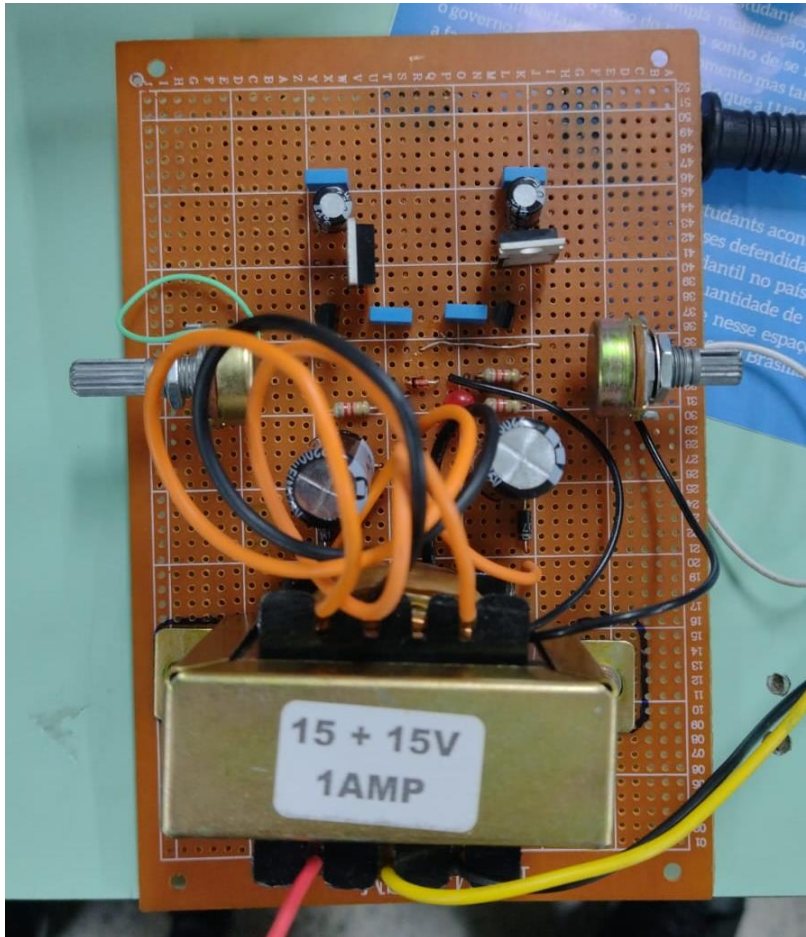


Figura 6 - Foto de montagem de fonte de alimentação simétrica

Porém, após várias testagens, constatamos que o circuito estava com um problema, que fazia com que ao conectar a fonte a uma carga, um dos lados da saída sofresse uma queda de tensão, de 10V para 2V. Além disso, o tamanho e peso da placa acabou fazendo com que a equipe procurasse outras alternativas para realizar a alimentação dos amplificadores operacionais.

Após vários testes e tentativas, optou-se por uma solução de utilizar a alimentação via USB do próprio ESP32 como alimentação primária, passando ela por um step-up que a converteria em uma tensão de 12VDC, e em seguida passar a tensão por um módulo isolador simétrico de conversão de 12VDC em  $\pm 12$ VDC.

O módulo utilizado foi o A1212S. Ele acabou mostrando-se extremamente útil no desenvolvimento do projeto, por seu tamanho diminuído, e a capacidade de produzir uma saída perfeitamente simétrica sem ruídos e sem grandes complicações. As duas primeiras páginas do datasheet do equipamento podem ser observadas nas figuras a seguir:



## DC/DC CONVERTER

## A\*\*\*\*S/D-2W(R2) Series

1600VDC Isolated 2W Double Output SIP/DIP-Package

## Features

- RoHS compliant
- Efficiency up to 83%
- 1.6kVDC Isolation
- Double output
- Short circuit protection( Suffix “R2” only SIP,optional)
- Industry standard required
- Wide temperature performance at full 2 Watt load,-40°C to 85°C



## Model Selection Guide

Order Code	Vin(V)		Output		Max capacitive Load	Efficiency(%) (Typ)
	Nominal	Range	Vo(V)	Io(mA)		
A0505S/D-2W(R2)	5	4.5-5.5	±5	+200	150	82
A0509S/D-2W(R2)			±9	+111	150	84
A0512S/D-2W(R2)			±12	+83	100	85
A0515S/D-2W(R2)			±15	+67	100	82
A0524S/D-2W(R2)			±24	+42	68	84
A1205S/D-2W(R2)	12	10.8-13.2	±5	+200	150	81
A1209S/D-2W(R2)			±9	+111	150	84
A1212S/D-2W(R2)			±12	+83	100	85
A1215S/D-2W(R2)			±15	+67	100	82
A1224S/D-2W(R2)			±24	+42	68	84
A2405S/D-2W(R2)	24	21.6-26.4	±5	+200	150	80
A2409S/D-2W(R2)			±9	+111	150	84
A2412S/D-2W(R2)			±12	+83	100	83
A2415S/D-2W(R2)			±15	+67	100	83
A2424S/D-2W(R2)			±24	+42	68	84

\*All the specifications typical at Ta=+25°C resistive load, nominal input voltage and rated output current unless otherwise noted.

## Input Characteristics

Parameter	Condition	Min	Typ	Max	Units
Input Surge Voltage (1 sec. Max.)	3.3V Input Models	-0.7	--	6	VDC
	5V Input Models	-0.7	--	9	
	12V Input Models	-0.7	--	18	
	24V Input Models	-0.7	--	30	
Input Filter	All Models	Internal Capacitor			

## Output Characteristics

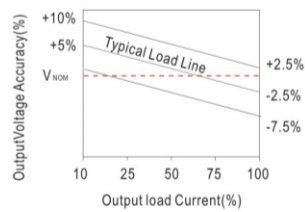
Parameter	Condition	Min	Typ	Max	Units
Line regulation	Vin change 1%	±1.2	--	±1.5	%
Switching frequency	Full load,nominal input	--	100	--	KHz
Load regulation	10%~100% load	6.5	--	15	%
Ripple and noise	BW=DC to 20MHz	--	75	≤100	mVp-p
Short circuit Protection	Suffix “R2”	Continuous, Automatic Recovery			



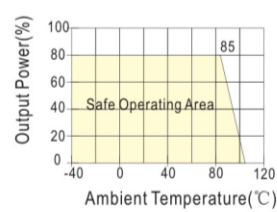
## General Characteristics

Parameter	Condition	Min	Typ	Max	Units
Operating Temperature	All output types	-40	---	+85	°C
Storage		-55	---	+125	°C
Storage humidity		---	---	+95	%
Cooling	Free air convection	---	---	---	
Isolation voltage	1mA ≤ 1minute	---	1600	---	VDC
Isolation resistance	500VDC	1000	---	---	MΩ
Switching Frequency		---	100	110	KHz
MTBF	$3.5 \times 10^6$				K hours

## Tolerance Envelopes Curve

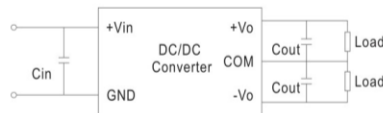


## Temperature Derating Graph Curve



## Input/Output Ripple Reduction

Reduce output ripple, it is recommended to use capacitors at the input/output.



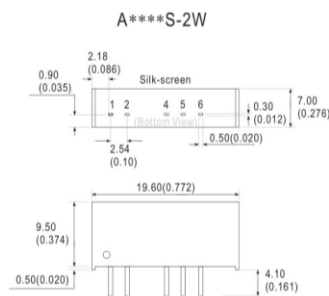
## External Capacitor Table

Vin(VDC)	5	12	15	24	
Cin(uF)	4.7	2.2	2.2	1	
Vout(VDC)	5	9	12	15	24
Cout(uF)	10	4.7	2.2	1	0.47

## Note

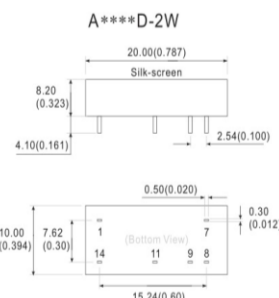
1. To ensure this module can operate efficiently and reliably, During operation, the minimum output load is not less than 10% of the full load.
2. Other input and output voltage may be available, please
3. Specifications subject to change without notice

## Mechanical Dimension &amp; Pin Connections



Pin	1	2	4	5	6
Function	Vin	GND	-Vo	COM	+Vo

Note:  
Unit:mm(inch)



Pin	1	7	8	9	11	14
Function	GND	NC	COM	+Vo	-Vo	Vin

Add

REV:3.1

Page 2 of 2

www.reicu.com

E-mail:service@reicu.com

Figura 8 - Página 2 do datasheet do A1212S

Por fim, foi montado todos os circuitos em uma placa padrão tipo ilha de 10x5cm, já contendo o circuito de tratamento de sinais e as interconexões do

ESP32 com o mesmo e com sua alimentação, além do ground comum. A placa final ficou conforme a figura a seguir:

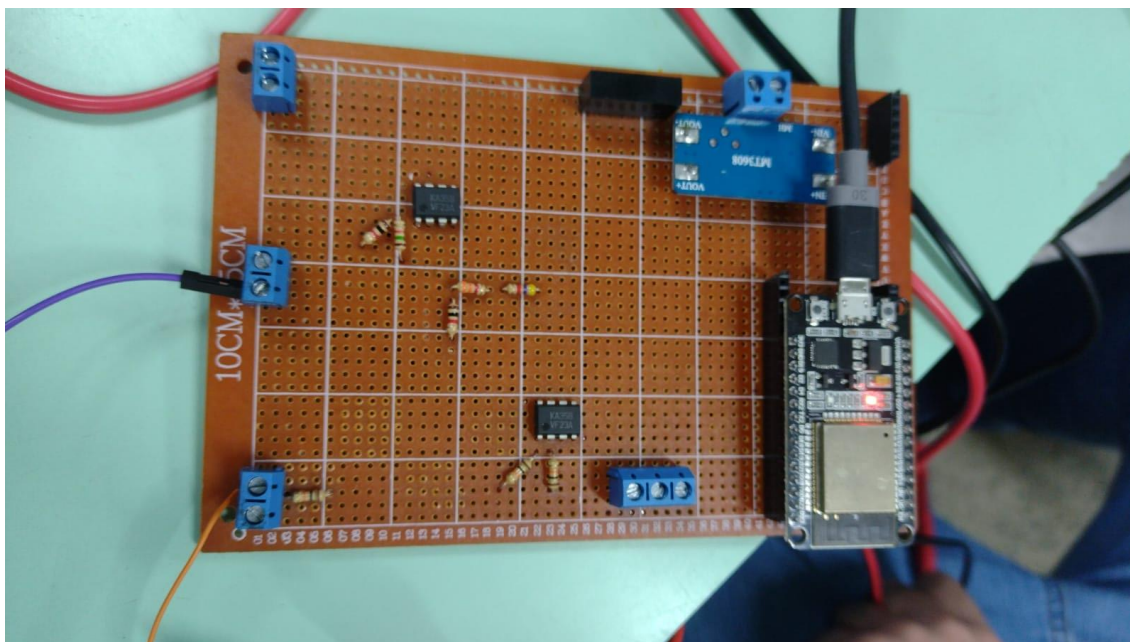


Figura 9 - Montagem final de placa

Por fim, o desenvolvimento do programa do ESP32 produziu um código, disponibilizado nos apêndices deste trabalho. Para a melhor compreensão de seu funcionamento, foi produzido um fluxograma, exibido a seguir, que demonstra as diferentes etapas do funcionamento da programação do ESP32:

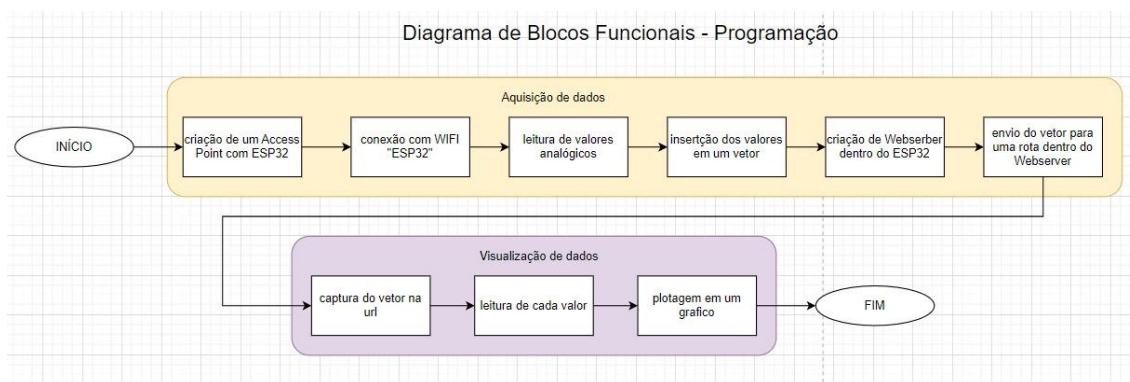


Figura 10 - Funcionamento do programa contido no ESP32



## 5 DISCUSSÃO

As duas partes principais do projeto foram o desenvolvimento da placa eletrônica para tratamento do sinal de tensão, e o desenvolvimento da programação do ESP32.

Em relação ao desenvolvimento da placa eletrônica, o primeiro desafio foi o desenvolvimento do projeto com os amplificadores operacionais para tratar o sinal de entrada.

Foi utilizado os diferentes modos de utilização de amplificadores operacionais, conforme explicado por WENDLING (2010). Foram utilizados três circuitos de amp-ops, conforme lista a seguir:

1. O primeiro foi um circuito amplificador operacional em modo de amplificador inversor. Nele, a onda de entrada é multiplicada por uma constante de  $-R_f/R_i$ , no caso,  $-1K5/10K$ , o que resulta em uma constante de multiplicação de  $-0,15$ . Assim, é possível diminuir a amplitude do sinal de entrada, de  $20V$  ( $\pm 10V$ ) para  $3V$  ( $\pm 1.5V$ ).
2. Em seguida, era necessário realizar um offset DC da onda, dado que o ESP32 não é capaz de realizar a leitura de tensões negativas. Para tanto, foi implementado um segundo circuito amplificador operacional, na função somador inversor. Nele, o sinal de saída corresponde a  $-V_{in} - 1,5$ . Percebe-se, portanto, duas coisas: primeiro, que o offset é realizado “para baixo”; e segundo, que a onda é invertida (multiplicada por  $-1$ ). Assim, temos na saída do segundo amp-op uma onda com a mesma amplitude de  $3V$  que saiu do primeiro amp-op, porém invertida e em uma faixa de  $-3V$  a  $0V$ .
3. Por fim, para passar a onda que saiu do segundo amp-op para a faixa positiva de tensão que pode ser lida pelo ESP32, a onda é passada por um terceiro amp-op, com função amplificadora inversora, onde a onda é multiplicada por uma constante de  $-1$ . Assim, temos na saída do terceiro amp-op uma onda com amplitude de  $3V$ , compreendida em uma faixa de tensão de  $0V$  a  $+3V$ .

### 5.1 Lições Aprendidas

Foram encontrados dois principais obstáculos no desenvolvimento dos projetos do trabalho.

Primeiro, em relação ao circuito eletrônico. O fato de termos sido obrigados a utilizar um step-up, que possuía seu pino “comum” equipotencializado com o ground do sistema, utilizado pelo ESP32 para ler o sinal da onda, fez com que um ruído fosse incorporado ao sinal lido pelo ESP32. Esse ruído pode ser percebido na foto exibida abaixo:

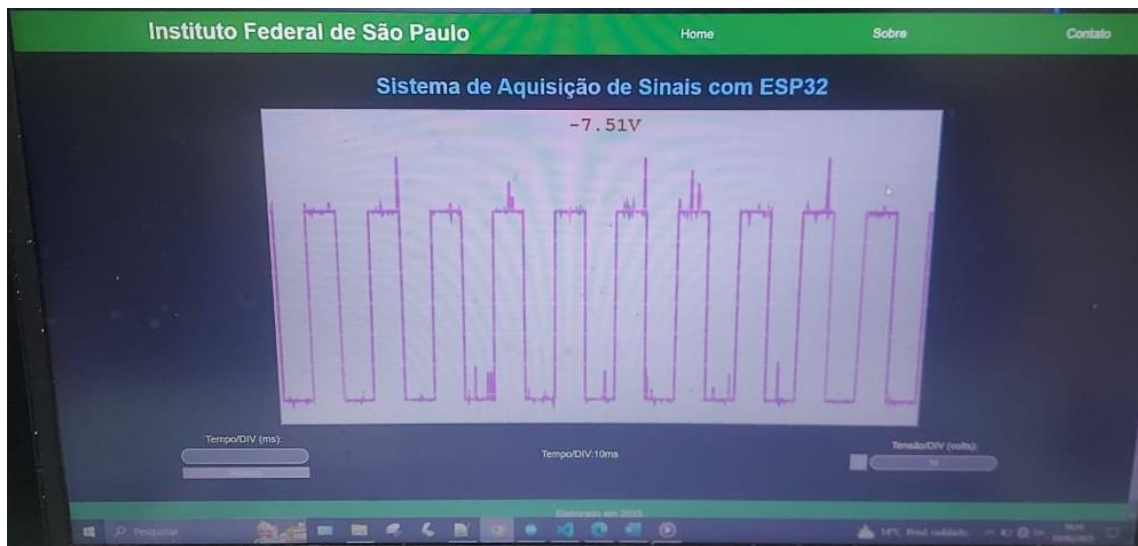


Figura 11 - Foto demonstrando ruído existente no sinal lido pelo ESP32

Esse ruído é resultado da utilização do step-up, que, conforme é sabido, produz esse tipo de ruído ao realizar a conversão das tensões, que acaba sendo lido pelo sistema devido ao fato da equipotencialização dos grounds do sistema.

Há duas formas de eliminar esse problema: utilizando uma alimentação separada para os amp-ops e para o ESP32. Ou então, utilizando um segundo módulo isolador, de forma a isolar eletricamente os dois circuitos, impedindo que o ruído produzido em uma parte da placa se espalhe para o restante dela. Porém, devido à falta de tempo, nenhuma das duas soluções pôde ser implementada neste projeto.

O segundo problema encontrado foi no desenvolvimento da programação do ESP32. A captura dos valores de tensão lidos pelo ADC do ESP32 é realizado em um intervalo de tempo determinado, porém tivemos dificuldade em garantir

que esse intervalo seria sempre o mesmo. Isso fez com que a onda produzida não tivesse sua simetria garantida. Ou seja, como às vezes alguns valores demoravam mais ou menos tempo para ser lidos dos outros, isso fazia com que a onda produzida não seguisse uma mesma escala temporal, o que distorcia totalmente a sua forma.

Para resolver esse problema, foi implementada a biblioteca síncrona do ESP32, e foi implementado a seguinte solução: realizar 1000 capturas de valores do ADC, de forma que seria possível garantir o mesmo tempo de duração da captura para cada uma das 1000 capturas, e após sua finalização enviá-los para o gráfico da WEB para ser plotados. Em seguida, repetir o processo.

Assim, conseguimos garantir a simetria da onda, e uma forma final de onda fiel à onda produzida originalmente.

## **6 CONCLUSÃO**

O Sistema de aquisição de dados tornou-se um projeto promissor após os resultados vistos em testes, ainda que houvesse algumas complicações, foi possível atender ao objetivo proposto. Vale salientar que será necessário utilizar um filtro para reduzir os ruídos de leitura que aparecem durante a visualização dos sinais, dessa maneira o projeto conseguirá atender as necessidades dos estudantes que utilizaram esse dispositivo.

## REFERÊNCIAS

DO MECÂNICO FRANQUEADORA, M. E. O que é um osciloscópio e por que utilizar? Disponível em: <<https://escoladomecanico.com.br/o-que-e-osciloscopio-e-como-usar/>>. Acesso em: 2 nov. 2022.

GUIMARÃES, F. Gerando sinal alternado com Arduino. Mundo Projetado Fábio, , 8 maio 2019. Disponível em: <<https://mundoprojetado.com.br/gerando-sinal-alternado-com-arduino/>>. Acesso em: 2 nov. 2022

VALLEJO, H. D. Osciloscopios. [s.l.] Editorial Quark, 1997.

Lucas F. Cardoso, Verônica M. L. Silva, Francisco C. G S Segundo - Como são mais de 2 autores, podemos fazer a referência da seguinte maneira:

CARDOSO, L. F, et. al. Osciloscópio de baixo custo utilizando a plataforma arduino. In: ECOP/UFERSA 2017, Pau dos Ferros. Anais do Encontro de Computação do Oeste Potiguar, RN: UFERSA, v. 1, p. 183-190 , 2017.

Disponível em: <[https://repositorio.ul.pt/bitstream/10451/1129/10/21575\\_ulsd057887\\_td\\_materiais\\_metodos.pdf](https://repositorio.ul.pt/bitstream/10451/1129/10/21575_ulsd057887_td_materiais_metodos.pdf)>. Acesso em: 2 nov. 2022.

BRAGA, N. C. Fonte simétrica de 0 1 12V com 1A. Instituto NCB, 27 set 2018. Disponível em: <<https://www.newtoncbraga.com.br/index.php/projetos/15449-fonte-simetrica-de-0-a-12-v-com-1-a-art1733.html>>. Acesso em: 22 jun 2023.

WENDLING, Marcelo. Amplificadores operacionais. UNESP: Colégio Técnico Industrial de Guaratinguetá, Guaratinguetá, 2010. Disponível em: <https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/3---amplificadoresoperacionais-v2.0.pdf>. Acesso em: 11 dez. 2021.

## APÊNDICE A – PROGRAMAS COMPUTACIONAIS

### 1 - Programa principal de aquisição dos dados analógicos

```
//-----Bibliotecas utilizadas-----

#include "SPIFFS.h"
#include <WebServer.h>

//-----DAC-----
/* Definição do canal DAC0 (GPIO 25) */
/*
#define CANAL_DAC0 25

//variáveis utilizadas para gerar uma onda triangular
bool turn = 0;
uint16_t sinal = 0;
uint32_t time_old = 0;
*/

//-----WebServer-----

// Credenciais do Access Point
const char *ssid = "ESP32";
const char *password = "12345678";

// Porta do Web Server
WebServer server(80);

//-----Leitura Analógica-----
```

```

// pino do ESP32 correspondente a leitura analógica
const int pinoAnalogico = 34;

// quantidade de valores capturados de uma vez
const int quantValoresCapturados = 1000;
// vetor com valores capturados por vez
int valoresAnalog_capturados[quantValoresCapturados];
// valores serão inseridos dentro de uma string
String valores = "";
String voltagem = "";
float tempoCapturaValores = 10; //delayMicroseconds
float novo_tempoCapturaValores = 0;

//-----

// Função que captura valores e formata como uma string
String juntarValores()
{

    if (novo_tempoCapturaValores==0){
        tempoCapturaValores = 10; //delayMicroseconds
    } else {
        tempoCapturaValores = novo_tempoCapturaValores;
    }
    //Serial.println(tempoCapturaValores);

    valores = "{";

    for (int i = 0; i < quantValoresCapturados; i++)
    {
        valores += " ";
    }
}

```

```

valores += "valor";
valores += i;
valores += "";
valores += ':';
valoresAnalog_capturados[i] = analogRead(pinoAnalogico);

delayMicroseconds(tempoCapturaValores);

valores += valoresAnalog_capturados[i];

if (i < quantValoresCapturados - 1)
{
    valores += ',';
}
}
valores += "}";
return valores;
}

//-----Definição do conteúdo de cada página-----

void pag_index()
{
    // Sub-rotina para caso o servidor fique online
    server.send(200, "text/html",
html_index(juntarValores(),tempoCapturaValores)); // Envia ao servidor, em
formato HTML, o script, com os parâmetros de pressão e temperatura
}

void pag_sobre()
{
    // Sub-rotina para caso o servidor fique online
    server.send(200, "text/html", html_sobre()); // Envia ao servidor, em formato
HTML, o script, com os parâmetros de pressão e temperatura
}

```



```

void pag_contato()
{
    // Sub-rotina para caso o servidor fique online
    server.send(200, "text/html", html_contato()); // Envia ao servidor, em formato
    HTML, o script, com os parâmetros de pressão e temperatura
}

```

```

void valores_Debug()
{
    // Sub-rotina para caso o servidor fique online
    server.send(200, "text/plain", juntarValores().c_str()); // Envia ao servidor, em
    formato HTML, o script, com os parâmetros de pressão e temperatura
}

```

```

void nao_encontrado()
{
    // Sub-rotina para caso seja retornado um erro
    server.send(404, "text/plain", "Não encontrado"); // Retorna a mensagem de
    erro em caso de um retorno 404
}

```

```

//-----

```

```

//Tempo/DIV
void capturaTempo () { //define novo valor de delay na captura de valores
    String tempo = server.arg("tempoNovo"); //captura variavel do html_index
    novo_tempoCapturaValores = tempo.toFloat();
    //Serial.println(tempoNovo);
    server.sendHeader("Location","/"); //redirecina para página default
    server.send(302, "text/plain", "redirecionado");
}

```

```

//-----

```

```

void setup()
{
  Serial.begin(115200);
  analogReadResolution(12); // resolução de 12 bits do ADC

  // -----Iniciando SPIFFS-----
  if (!SPIFFS.begin(true))
  {
    Serial.println("Ocorreu um erro na criação do SPIFFS");
    return;
  }

  // -----Connectando WI-FI-----
  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("Endereço IP - Access Point: ");
  Serial.println(IP);
  Serial.println(WiFi.localIP());

  // -----Rota das páginas-----
  server.on("/", pag_index);
  server.on("/sobre", pag_sobre);
  server.on("/contato", pag_contato);
  server.on("/texto", valores_Debug);

  server.on("/frequencia", capturaTempo);

  server.onNotFound(nao_encontrado);
  server.begin(); // Inicia o servidor
  Serial.println("Servidor Online");
}

```

```

void loop()
{
    //para teste:
    /*
    if (micros() - time_old >= 10)
    {
        // Atualiza a variável para a próxima iteração
        time_old = micros();

        //*****
        // Gera uma onda triangular de 1KHz no pino 25
        if (turn == 0)
        {
            sinal += 5;
            if (sinal >= 255)
            {
                turn = 1;
            }
        }

        else if (turn == 1)
        {
            sinal -= 5;
            if (sinal == 0)
            {
                turn = 0;
            }
        }

        // Escreve o valor no DAC (0 - 255)
        dacWrite(CANAL_DAC0, sinal);
        //*****
    }
    */
}

```

```
server.handleClient();  
}
```

2 - Página principal

```
String html_contato()  
{
```

```
String conteudo;
```

```
conteudo = F(  
  

```

```
"<!DOCTYPE html>\n"  
"<html lang=\"pt-br\">\n"
```

```
"<head>\n"  
  "<meta charset=\"UTF-8\">\n"  
  "<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\">\n"  
  "<meta    name=\"viewport\"    content=\"width=device-width,    initial-  
scale=1.0\">\n"  
  "<title>ESP32</title>\n"  
  "<link rel=\"stylesheet\" href=\"contato.css\">\n"  
"</head>\n"
```

```
"<header>\n"  
  "<div class=\"logof\">\n"  
    "<h1>Instituto Federal de São Paulo</h1>\n"  
  "</div>\n"
```

```
"<nav class=\"menu\">\n"  
  "<div class=\"topnav\">\n"  
    "<a class=\"active\" href=\"http://192.168.4.1/\">Home</a>\n"  
    "<a href=\"http://192.168.4.1/sobre\">Sobre</a>\n"
```

```
"<a href=\"http://192.168.4.1/contato\">Contato</a>\n"
"</div>\n"
"</nav>\n"
"</header>\n"

"<body>\n"
  "<div class=\"titulo\">\n"
    "<h1>Contato</h1>\n"
  "</div>\n"
  "<main class=\"conteudo\">\n"
    "<div>\n"
      "<h2>Alunos</h2>\n"
      "<ul>\n"
        "<a                href=\"https://www.linkedin.com/in/bruno-vinicius-81a3181b1\">\n"
          "<li>Bruno Vinicius Lima da Silva</li>\n"
        "</a>\n"
        "<a href=\"https://www.linkedin.com/in/crparaizo\">\n"
          "<li>Cândida Rosa Paraizo</li>\n"
        "</a>\n"
        "<a                href=\"https://www.linkedin.com/in/marcos-knijnik-70244748\">\n"
          "<li>Marcos Judensnaider Knijnik</li>\n"
        "</a>\n"
      "</ul>\n"
    "</div>\n"
    "<div>\n"
      "<h2>Professores</h2>\n"
      "<ul>\n"
        "<a href=\"#\">\n"
          "<li>Gustavo Neves Margarido</li>\n"
        "</a>\n"
        "<a href=\"#\">\n"
```

```
"<li>Rodrigo Rech</li>\n"
"</a>\n"
"<a href=\"#\">\n"
  "<li>Tarsício Fernandes Leão</li>\n"
"</a>\n"
"</ul>\n"
"</div>\n"
"<div>\n"
  "<h2>Instituição</h2>\n"
  "<a href=\"https://www.ifsp.edu.br/\">\n"
    "<li>Instituto Federal de São Paulo</li>\n"
  "</a>\n"
"</div>\n"
"<div>\n"
  "<h2>Links do projeto</h2>\n"
  "<ul>\n"
    "<a href=\"https://github.com/crparaizo/tcc_ifsp_2023\">\n"
      "<li>Repositório Github</li>\n"
    "</a>\n"
    "<a href=\"https://github.com/vthayashi/labhome\">\n"
      "<li>Projeto base</li>\n"
    "</a>\n"
  "</ul>\n"
"</div>\n"
"</main>\n"
"</body>\n"

"<footer class=\"rodape\">\n"
  "<p>Elaborado em 2023</p>\n"
"</footer>\n"

"</html>\n"
```

```
"<style>\n"
/* customização aplicada a todos os elementos da página */
"* {\n"
  "margin: 0;\n"
  "padding: 0;\n"
  "box-sizing: border-box;\n"
  "font-family: \"open_sansbold\", 'Open Sans', Arial, Helvetica, sans-
serif;\n"
  "}\n"

/* customização do cabeçalho */
"header {\n"
  "display: flex;\n"
  "justify-content: space-between;\n"
  "align-items: center;\n"
  "padding: 10px 0;\n"
  "background-color: #17882c;\n"
  "}\n"

".logoIF {\n"
  "width: 50%;\n"
  "display: flex;\n"
  "align-items: center;\n"
  "justify-content: space-evenly;\n"
  "color: white;\n"
  "}\n"

".menu {\n"
  "width: 50%;\n"
  "}\n"

".topnav {\n"
```

```
"display: flex;\n"
"justify-content: space-around;\n"
"}\n"
```

```
".topnav a {\n"
"color: white;\n"
"font-size: 18px;\n"
"text-decoration: none;\n"
"}\n"
```

```
".topnav a:hover {\n"
"color: yellow;\n"
"text-decoration: underline;\n"
"}\n"
```

/\* customização do corpo da página \*/

```
".titulo {\n"
"display: flex;\n"
"justify-content: center;\n"
"padding-top: 40px;\n"
"color: #2c66ce;\n"
"margin: 1em 0;\n"
"}\n"
```

```
".conteudo {\n"
"display: flex;\n"
"align-items: center;\n"
"justify-content: center;\n"
"flex-direction: column;\n"
"}\n"
```

```
".conteudo div {\n"
```



```
"align-items: center;\n"
"display: flex;\n"
"flex-direction: column;\n"
"margin: 15px;\n"
"}\n"
```

```
".conteudo div h2 {\n"
  "font-size: 17px;\n"
}\n"
```

```
".conteudo div a {\n"
  "text-decoration: none;\n"
  "list-style-type: none;\n"
}\n"
```

```
".conteudo div a li {\n"
  "padding-top: 13px;\n"
  "justify-content: center;\n"
  "display: flex;\n"
}\n"
```

```
".conteudo div a li:hover {\n"
  "text-decoration: underline;\n"
  "color: red;\n"
}\n"
```

```
/* customização do rodapé */
```

```
".rodape {\n"
  "width: 100%;\n"
  "display: flex;\n"
  "justify-content: center;\n"
  "align-items: center;\n"
```

```
"height: 50px;\n"
"background-color: #17882c;\n"
"color: white;\n"
"margin-top: 30px;\n"
"}\n"

/* Aplicando responsividade de acordo com tamanho da tela */

"@media screen and (max-width: 1300px) {\n"

  ".logoF h1 {\n"
    "font-size: 25px;\n"
  "}\n"
}\n"

" @media screen and (max-width: 1100px) {\n"

  ".logoF h1 {\n"
    "font-size: 20px;\n"
  "}\n"
}\n"

"@media screen and (max-width: 800px) {\n"

  ".logoF h1 {\n"
    "font-size: 15px;\n"
  "}\n"

  ".topnav a {\n"
    "font-size: 13px;\n"
  "}\n"
}\n"
```

```
"@media screen and (max-width: 600px) {\n"
```

```
  ".titulo h1 {\n"
```

```
    "font-size: 20px;\n"
```

```
  "}\n"
```

```
  ".logoIF {\n"
```

```
    "width: 150%;\n"
```

```
  "}\n"
```

```
  ".menu {\n"
```

```
    "width: 40%;\n"
```

```
  "}\n"
```

```
  ".topnav {\n"
```

```
    "flex-direction: column;\n"
```

```
    "justify-content: center;\n"
```

```
    "align-items: center;\n"
```

```
  "}\n"
```

```
  ".topnav a {\n"
```

```
    "font-size: 16px;\n"
```

```
    "padding-bottom: 4px;\n"
```

```
  "}\n"
```

```
"}\n"
```

```
"</style>\n"
```

```
);
```

```
return contenido;
```

```
}
```

```
String html_index(String dado,float tempo)
```

```
{
```

```
String conteudo; // variável que armazenará o script HTML
```

```
conteudo = F(
```

```
"<!DOCTYPE html>\n"
```

```
"<html lang=\"pt-br\">\n"
```

```
"<head>\n"
```

```
    "<meta charset=\"UTF-8\">\n"
```

```
    "<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\">\n"
```

```
    "<meta      name=\"viewport\"      content=\"width=device-width,      initial-  
scale=1.0\">\n"
```

```
    "<title>ESP32</title>\n"
```

```
"</head>\n"
```

```
"<header>\n"
```

```
    "<div class=\"logoIF\">\n"
```

```
        "<h1>Instituto Federal de São Paulo</h1>\n"
```

```
    "</div>\n"
```

```
"<nav class=\"menu\">\n"
```

```
    "<div class=\"topnav\">\n"
```

```
        "<a class=\"active\" href=\"http://192.168.4.1/\">Home</a>\n"
```

```
        "<a href=\"http://192.168.4.1/sobre\">Sobre</a>\n"
```

```
        "<a href=\"http://192.168.4.1/contato\">Contato</a>\n"
```

```
    "</div>\n"
```

```
"</nav>\n"
```

```
"</header>\n"
```

```
"<body>\n"
```

```
"<div class=\"titulo\">\n"
```

```
"<h1>Sistema de Aquisição de Sinais com ESP32</h1>\n"
```

```
"</div>\n"
```

```
"<!-- Limitações da tag canvas: alguns navegadores mais antigos"
```

```
"(em particular, versões do Internet Explorer anteriores a 9)"
```

```
"não suportam o elemento -->"
```

```
"<div class=\"main\">\n"
```

```
"<div class=\"grafico\">\n"
```

```
"<canvas id=\"grafico\">\n"
```

```
"<!-- Se seu navegador não suportar HTML5 você verá a mensagem  
abaixo -->"
```

```
"Este navegador não suporta HTML5, não será possível visualizar esse  
gráfico"
```

```
"</canvas>\n"
```

```
"<div id=\"fundo_grafico\"></div>\n"
```

```
"</div>\n"
```

```
"<div id=\"botoes\">\n"
```

```
"<div class=\"frequencia\">\n"
```

```
"<form id=\"formulario\" action=\"frequencia\" method=\"post\">\n"
```

```
"Tempo de coleta(us): <input class=\"tempo_div\" type=\"text\"  
name=\"tempoNovo\">\n"
```

```
"<input type=\"submit\" value=\"Atualizar\">\n"
```

```
"</form>\n"
```

```
"</div>\n"
```

```

"<div class=\"acompanhamento\">\n"
);
conteudo += F("<p>Tempo de coleta:");
conteudo += tempo;
conteudo += F("us</p>\n");

conteudo += F("<p>Tempo/DIV:");
conteudo += tempo*50;
conteudo += F("us</p>\n");

conteudo += F(
    "</div>\n"

"<div class=\"amplitude\">\n"
    "<p id=\"textoAmp\">Tensão/DIV (volts):</p>\n"
    "<button    id=\"botaoAnteriorAmp\"    onclick=\"diminuirAmp()\">-
</button>\n"
    "<input class=\"tensao_div\" id=\"valorAmp\" value=\"10\">\n"
    "<button                                id=\"botaoPosteriorAmp\"
onclick=\"aumentarAmp()\">+</button>\n"
    "</div>\n"

" </div>\n"

"</div>\n"

"</body>\n"

"<footer class=\"rodape\">\n"
    "<p>Elaborado em 2023</p>\n"
"</footer>\n"

"</html>\n"

```

```
"<style>\n"
```

```
/* customização aplicada a todos os elementos da página */
```

```
"@charset \"UTF-8\";\n"
```

```
"* {\n"
```

```
  "margin: 0;\n"
```

```
  "padding: 0;\n"
```

```
  "box-sizing: border-box;\n"
```

```
  "font-family: \"open_sansbold\", 'Open Sans', Arial, Helvetica, sans-serif;\n"
```

```
"}\n"
```

```
/* customização do cabeçalho */
```

```
"header {\n"
```

```
  "display: flex;\n"
```

```
  "justify-content: space-between;\n"
```

```
  "align-items: center;\n"
```

```
  "padding: 10px 0;\n"
```

```
  "background-color: #17882c;\n"
```

```
"}\n"
```

```
".logoIF {\n"
```

```
  "width: 50%;\n"
```

```
  "display: flex;\n"
```

```
  "align-items: center;\n"
```

```
  "justify-content: space-evenly;\n"
```

```
  "color: white;\n"
```

```
"}\n"
```

```
".menu {\n"
```

```
  "width: 50%;\n"
```

```
"}\n"
```

```
".topnav {\n"
  "display: flex;\n"
  "justify-content: space-around;\n"
}\n"
```

```
".topnav a {\n"
  "color: white;\n"
  "font-size: 18px;\n"
  "text-decoration: none;\n"
}\n"
```

```
".topnav a:hover {\n"
  "color: yellow;\n"
  "text-decoration: underline;\n"
}\n"
```

```
/* customização do corpo da página */
```

```
".titulo {\n"
  "display: flex;\n"
  "justify-content: center;\n"
  "padding-top: 30px;\n"
  "color: #2c66ce;\n"
}\n"
```

```
/* gráfico que mostra o sinal gerado */
```

```
".grafico {\n"
  "display: flex;\n"
  "justify-content: center;\n"
  "margin: 15px 0;\n"
}\n"
```



```

"#grafico {\n"
    "width: 1000px;\n"
    "height: 500px;\n"
    "border: 2px solid;\n"
    "border-color: black;\n"
}\n"

/* ----- */

/* gráfico que simula os aspectos da tela do osciloscopio */

"#fundo_grafico {\n"
    "background-color: transparent;\n"
    "z-index: 2;\n"
    "position: absolute;\n"

    "background: rgba(255, 255, 255, 0.5);\n"
    "background-image: linear-gradient(rgba(198, 147, 147, 0.5) .5px,
transparent 5px),\n"
    "linear-gradient(90deg, rgba(198, 147, 147, 0.5) .5px, transparent
5px);\n"
    "background-size: 50px 50px;\n"
/* ----- */
    "width: 1000px;\n"
    "height: 500px;\n"
/* ----- */
}\n"

"#canvasGrafico {\n"
    "z-index: 1;\n"
    "position: absolute;\n"
}\n"

```

```
"main {\n  \"display: flex;\n  \"flex-direction: row;\n}\n"
```

```
"#formulario {\n  \"display: flex;\n  \"flex-direction: column;\n  \"text-align: center;\n}\n"
```

```
"#botoes {\n  \"display: flex;\n  \"flex-direction: row;\n  \"text-align: center;\n  \"justify-content: space-around;\n  \"align-items: center;\n}\n"
```

```
".amplitude button {\n  \"height: 2em;\n  \"width: 2em;\n}\n"
```

```
".tempo_div,\n\".tensao_div {\n  \"border-radius: 50px;\n  \"text-align: center;\n  \"margin: 7px 0;\n  \"width: 200px;\n  \"height: 25px;\n}\n"
```

```
"#botaoPosteriorAmp{\n"
  "visibility: hidden;\n"
}\n"
```

```
/* ----- */
```

```
/* customização do rodapé */
```

```
".rodape {\n"
  "width: 100%;\n"
  "display: flex;\n"
  "justify-content: center;\n"
  "align-items: center;\n"
  "height: 50px;\n"
  "background-color: #17882c;\n"
  "color: white;\n"
  "margin-top: 40px;\n"
}\n"
```

```
/* Aplicando responsividade de acordo com tamanho da tela */
```

```
"@media screen and (max-width: 1300px) {\n"
```

```
  ".logoIF h1 {\n"
    "font-size: 25px;\n"
  }\n"
}\n"
```

```
"@media screen and (max-width: 1100px) {\n"
```

```
  ".logoIF h1 {\n"
    "font-size: 20px;\n"
```

```
"}\n"
```

```
"#grafico {\n"
```

```
  "width: 800px;\n"
```

```
"}\n"
```

```
"}\n"
```

```
"@media screen and (max-width: 900px) {\n"
```

```
"#grafico {\n"
```

```
  "width: 600px;\n"
```

```
"}\n"
```

```
"}\n"
```

```
"@media screen and (max-width: 800px) {\n"
```

```
".logoI F h1 {\n"
```

```
  "font-size: 15px;\n"
```

```
"}\n"
```

```
".topnav a {\n"
```

```
  "font-size: 13px;\n"
```

```
"}\n"
```

```
".titulo h1 {\n"
```

```
  "font-size: 25px;\n"
```

```
"}\n"
```

```
"}\n"
```

```
"@media screen and (max-width: 700px) {\n"
```

```
"#grafico {\n"
```

```
  "width: 500px;\n"
```

```
"}\n"
"}\n"
```

```
"@media screen and (max-width: 600px) {\n"
```

```
" .titulo h1 {\n"
  "font-size: 20px;\n"
"}\n"
```

```
" .logoIF {\n"
  "width: 150%;\n"
"}\n"
```

```
" .menu {\n"
  "width: 40%;\n"
"}\n"
```

```
" .topnav {\n"
  "flex-direction: column;\n"
  "justify-content: center;\n"
  "align-items: center;\n"
"}\n"
```

```
" .topnav a {\n"
  "font-size: 16px;\n"
  "padding-bottom: 4px;\n"
"}\n"
```

```
"#grafico {\n"
  "width: 400px;\n"
"}\n"
```

```
"@media screen and (max-width: 450px) {\n"
```

```
".titulo h1 {\n"
  "font-size: 15px;\n"
}\n"
```

```
"#grafico {\n"
  "width: 300px;\n"
}\n"
}\n"
```

```
"@media screen and (max-width: 415px) {\n"
```

```
".titulo h1 {\n"
  "text-align: center;\n"
}\n"
```

```
"#grafico {\n"
  "width: 250px;\n"
}\n"
}\n"
"</style>\n"
```

```
"<script>\n"
```

```
//definições dos botões de mudar tensão/div
```

```
"function aumentarAmp() {\n"
  "var tensao_volts = document.getElementById("valorAmp").value;\n"
```

```
"if (tensao_volts == 2.5) tensao_volts = 5;\n"
```

```
"else if (tensao_volts == 5) tensao_volts = 10;\n"
```

```
"if (tensao_volts != 2.5) {\n"
```

```

        "document.getElementById(\"botaoAnteriorAmp\").style.visibility
\"visible\";\n"
    "}\n"
    "if (tensao_volts == 10) {\n"
        "document.getElementById(\"botaoPosteriorAmp\").style.visibility
\"hidden\";\n"
    "}\n"
    "document.getElementById(\"valorAmp\").value = tensao_volts;\n"
    "}\n"

"function diminuirAmp() {\n"
    "var tensao_volts = document.getElementById(\"valorAmp\").value;\n"

    "if (tensao_volts == 5) tensao_volts = 2.5;\n"
    "else if (tensao_volts == 10) tensao_volts = 5;\n"

    "if (tensao_volts == 2.5) {\n"
        "document.getElementById(\"botaoAnteriorAmp\").style.visibility
\"hidden\";\n"
    "}\n"
    "if (tensao_volts != 10) {\n"
        "document.getElementById(\"botaoPosteriorAmp\").style.visibility
\"visible\";\n"
    "}\n"
    "document.getElementById(\"valorAmp\").value = tensao_volts;\n"
    "}\n"

//Espera elemento canvas renderizar para depois executar o script
>window.onload = function () {\n"

    //definições iniciais para o gráfico
    "var canvas = document.getElementById(\"grafico\");\n"
    "if (canvas) {\n"

```

```

"var altura = canvas.offsetHeight; //altura da canvas\n"
"var largura = canvas.offsetWidth; //largura da canvas\n"
"var deslocHorizontal = 0; //posição horizontal inicial do gráfico\n"
"var valorEP32; //valor dos pontos do gráfico, que será lido do ESP32\n"
"var zeroGrafico = 20; //determina o 0 do gráfico, para melhorar vizualização
do usuário\n"
"var voltagem;\n"

//formatando a canvas
"canvas.setAttribute(\"width\", largura);\n"
"canvas.setAttribute(\"height\", altura);\n"

//obtendo o contexto 2d
"var ctx = canvas.getContext(\"2d\");\n"
"ctx.lineWidth = 4;\n"

"ctx.fillStyle = \"white\";\n"
"ctx.fillRect(0, 0, largura, altura);\n"
"ctx.font = \"bold 30px Courier\";\n"
"}\n"

//variavel que controla amplitude do grafico
"var fatorDivisao = 1;\n"

// atualizar valores a cada 50ms
"setInterval(function () {\n"
  "pegarTexto();\n"
"}, 1);\n"

//criando sincronismo entre as funções
"var pegouTexto = false;\n"

```



//função feita com AJAX que pegar texto do Placeholder

//Formato é "Plain text"

```
"function pegarTexto() {\n  "var xhttp = new XMLHttpRequest();\n  "xhttp.onreadystatechange = function () {\n    //se chamada ajax foi concluida (4) e status for ok (200)\n    "if (this.readyState == 4 && this.status == 200) {\n      "var vetor = document.getElementById(\"placeholder\")\n      "vetor = this.responseText;\n      "var json = JSON.parse(vetor);\n      "pegouTexto = true;\n      "desenharGrafico(json);\n    }\n  };\n  "xhttp.open(\"GET\", \"texto\", true);\n  "xhttp.send();\n}\n"
```

//função desenharGrafico

```
"function desenharGrafico(json) {\n\n  //condições a depender do tensao/div escolhido:\n  "var valorTensao = document.getElementById(\"valorAmp\").value;\n\n  "if (valorTensao == 2.5) {\n    "fatorDivisao = 4;\n\n  "}" else if (valorTensao == 5) {\n    "fatorDivisao = 2;\n\n  "}" else if (valorTensao == 10) {\n    "fatorDivisao = 1;\n\n  "}" \n}
```

```

    "if ((pegouTexto == true)) {\n"

        //caso deslocamento do eixo horizontal for maior que a largura do
        grafico, eixo volta para posição 0 ~/\n"
        "if (deslocHorizontal >= largura) {\n"

            //onde começa x, onde termina y, tamanho dele (largura e altura):
            "ctx.clearRect(0, 0, 1, altura); //apagar inicio do grafico\n"
            "ctx.fillStyle = \"white\";\n"
            "ctx.fillRect(0, 0, 1, altura); //reconstruir inicio do grafico\n"
            "deslocHorizontal = 0; //deslocamento horizontal vou para inicio do
            grafico\n"

            "ctx.beginPath();\n"
            "}\n"

            "for (let valorJson in json) {\n"

                //exibe valores no Console do navegador - para debug

                "ctx.clearRect(0, 0, deslocHorizontal + 1, altura);\n"
                "ctx.fillStyle = \"white\";\n"
                "ctx.fillRect(0, 0, deslocHorizontal + 1, altura);\n"

                "voltage = - ((json[valorJson] * (3.3 / 4095)) * (20 / 3) - 10) /
                fatorDivisao;\n"
                "parsedValue = parseFloat(voltage);\n"
                "resultado = parsedValue.toFixed(2);\n"
                //console.log(resultado);

                "deslocHorizontal += 1; //deslocamento na horizontal\n"

```

```
        "valorEP32 = (json[valorJson] / 10) / fatorDivisao; //valor lido do  
ESP32\n"
```

```
        "ctx.lineTo(deslocHorizontal, altura - valorEP32 - zeroGrafico);  
//desenha uma linha até a posição gerada\n"
```

```
        "ctx.strokeStyle = \"magenta\";\n"
```

```
        "ctx.stroke();\n"
```

```
        //desenha um retangulo onde está sendo escrito o valor do gráfico
```

```
        "ctx.fillStyle = \"white\";\n"
```

```
        "ctx.fillRect(0, 0, largura, 30);\n"
```

```
        //desenha o texto indicando o valor do gráfico, na posição horizontal
```

```
        atual
```

```
        "ctx.fillStyle = \"red\";\n"
```

```
        "ctx.fillText(resultado + \"V\", 450, 30);\n"
```

```
    "}\n"
```

```
"}\n"
```

```
    "pegouTexto = false;\n"
```

```
"}\n"
```

```
"};\n"
```

```
"</script>\n"
```

```
    );
```

```
    return conteudo; // Retorna o script
```

```
}
```

4 - Página de “contato”

```
String html_sobre()
```

```
{
```

```
    String conteudo;
```

```
    conteudo = F(
```

```
"<!DOCTYPE html>\n"
"<html lang=\"pt-br\">\n"

"<head>\n"
  "<meta charset=\"UTF-8\">\n"
  "<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\">\n"
  "<meta    name=\"viewport\"    content=\"width=device-width,    initial-
scale=1.0\">\n"
  "<title>ESP32</title>\n"
  "<link rel=\"stylesheet\" href=\"sobre.css\">\n"
"</head>\n"

"<header>\n"
  "<div class=\"logof\">\n"
    "<h1>Instituto Federal de São Paulo</h1>\n"
  "</div>\n"

  "<nav class=\"menu\">\n"
    "<div class=\"topnav\">\n"
      "<a class=\"active\" href=\"http://192.168.4.1/\">Home</a>\n"
      "<a href=\"http://192.168.4.1/sobre\">Sobre</a>\n"
      "<a href=\"http://192.168.4.1/contato\">Contato</a>\n"
    "</div>\n"
  "</nav>\n"
"</header>\n"

"<body>\n"
  "<div class=\"titulo\">\n"
    "<h1>Sobre</h1>\n"
  "</div>\n"

  "<main class=\"conteudo\">\n"
```

"<p>O osciloscópio é uma ferramenta de extrema importância para a indústria eletroeletrônica, pois ele permite"

"visualizar um sinal elétrico na forma de gráfico. Em outras palavras, é possível mostrar uma imagem em dois"

"eixos, no qual o eixo x representa o tempo e o eixo y representa os volts em amplitude. </p>\n"

"<p>O objetivo desse trabalho é desenvolver um sistema de aquisição de sinais com algumas funcionalidades"

"similares a de um osciloscópio. Com ele será possível fazer medições em diversos tipos de ambientes (já que"

"sua portabilidade é fácil), sua usabilidade é bem simples (tornando-se intuitiva), e poderá ser utilizado em"

"escolas e universidades. Devido ao seu baixo custo, tanto alunos como professores poderão construir o seu,"

"para uso pessoal, em pesquisas.</p>\n"

"<p>Uma das suas características notáveis é o peso que tem aproximadamente 250g, isso facilitará no seu"

"manuseio evitando qualquer tipo de risco ergonômico para seu portador. O método de pesquisa desse trabalho"

"foi totalmente experimental, onde foi possível obter resultados satisfatórios que tornaram o projeto"

"promissor.</p>\n"

"</main>\n"

"</body>\n"

"<footer class=\"rodape\">\n"

"<p>Elaborado em 2023</p>\n"

"</footer>\n"

"</html>\n"

"<style>\n"

```
/* customização aplicada a todos os elementos da página */
"* {\n"
  "margin: 0;\n"
  "padding: 0;\n"
  "box-sizing: border-box;\n"
  "font-family: \"open_sansbold\", 'Open Sans', Arial, Helvetica, sans-
serif;\n"
  "}\n"
```

```
/* customização do cabeçalho */
"header {\n"
  "display: flex;\n"
  "justify-content: space-between;\n"
  "align-items: center;\n"
  "padding: 10px 0;\n"
  "background-color: #17882c;\n"
  "}\n"
```

```
".logoIF {\n"
  "width: 50%;\n"
  "display: flex;\n"
  "align-items: center;\n"
  "justify-content: space-around;\n"
  "color: white;\n"
  "}\n"
```

```
".menu {\n"
  "width: 50%;\n"
  "}\n"
```

```
".topnav {\n"
  "display: flex;\n"
  "justify-content: space-around;\n"
```

```
"}\n"
```

```
".topnav a {\n"
```

```
"color: white;\n"
```

```
"font-size: 18px;\n"
```

```
"text-decoration: none;\n"
```

```
"}\n"
```

```
".topnav a:hover {\n"
```

```
"color: yellow;\n"
```

```
"text-decoration: underline;\n"
```

```
"}\n"
```

```
/* customização do corpo da página */
```

```
".corpo {\n"
```

```
"padding: 150px 0;\n"
```

```
"}\n"
```

```
".titulo {\n"
```

```
"display: flex;\n"
```

```
"justify-content: center;\n"
```

```
"padding-top: 100px;\n"
```

```
"color: #2c66ce;\n"
```

```
"font-size: 20px;\n"
```

```
"}\n"
```

```
".conteudo {\n"
```

```
"width: 60%;\n"
```

```
"display: flex;\n"
```

```
"flex-direction: column;\n"
```

```
"margin: 70px 20%;\n"
```

```
"}\n"
```

```
".conteudo p {\n"
  "padding: 20px 0;\n"
  "font-size: 17px;\n"
  "text-align: justify;\n"
"}\n"
```

```
/* customização do rodapé */
```

```
".rodape {\n"
  "width: 100%;\n"
  "display: flex;\n"
  "justify-content: center;\n"
  "align-items: center;\n"
  "height: 50px;\n"
  "background-color: #17882c;\n"
  "color: white;\n"
  "margin-top: 45px;\n"
"}\n"
```

```
/* Aplicando responsividade de acordo com tamanho da tela */
```

```
"@media screen and (max-width: 1300px) {\n"
```

```
  ".logoIF h1 {\n"
    "font-size: 25px;\n"
  "}\n"
}\n"
```

```
"@media screen and (max-width: 1100px) {\n"
```

```
  ".logoIF h1 {\n"
    "font-size: 20px;\n"
  "}\n"
}\n"
```



```
"@media screen and (max-width: 800px) {\n"
```

```
  ".logoF h1 {\n"
```

```
    "font-size: 15px;\n"
```

```
  "}\n"
```

```
  ".topnav a {\n"
```

```
    "font-size: 13px;\n"
```

```
  "}\n"
```

```
"}\n"
```

```
"@media screen and (max-width: 600px) {\n"
```

```
  ".titulo h1 {\n"
```

```
    "font-size: 20px;\n"
```

```
  "}\n"
```

```
  ".logoF {\n"
```

```
    "width: 150%;\n"
```

```
  "}\n"
```

```
  ".menu {\n"
```

```
    "width: 40%;\n"
```

```
  "}\n"
```

```
  ".topnav {\n"
```

```
    "flex-direction: column;\n"
```

```
    "justify-content: center;\n"
```

```
    "align-items: center;\n"
```

```
  "}\n"
```

```
  ".topnav a {\n"
```

```
        "font-size: 16px;\n"
        "padding-bottom: 4px;\n"
    "}\n"
    "}\n"
    "</style>\n"

);

return conteudo;

}
```

## APÊNDICE B – MANUAL DO USUÁRIO

### Segurança

**IMPORTANTE:** Este aparelho deve ser mantido longe do alcance de crianças. Deve-se também observar as recomendações da OMS para que o uso venha promover a prevenção.

1. Mantenha o equipamento longe do alcance de crianças para evitar acidentes;
2. Mantenha o equipamento em um local adequado, longe de condições climáticas ruins para que a vida útil possa ser estendida;
3. Antes de manusear a placa, deve-se tocar em algum metal para evitar danificar os componentes eletrônicos por meio de descargas eletrostáticas;
4. Verifique se a tensão da rede elétrica é a mesma que a indicada na fonte de alimentação do equipamento.
5. Com o intuito de evitar acidentes, em hipótese nenhuma mergulhe o plugue de alimentação do equipamento na água ou qualquer outro líquido.

**NOTA:** Utilize apenas fontes de alimentação originais do fabricante/importador ou que possuam as mesmas especificações de tensão e corrente que constam na fonte do fabricante, a não observação da recomendação pode acarretar no funcionamento incorreto do aparelho.

### Utilização

1. Conectar ESP32 na porta USB do computador;
2. Acessar página do github < [https://github.com/crparaizo/tcc\\_ifsp\\_2023](https://github.com/crparaizo/tcc_ifsp_2023)>;
3. Fazer download dos arquivos do repositório no formato ZIP;
4. Extrair arquivos;
5. Instalar IDE do Arduino;
6. Abrir arquivos extraídos na IDE;

7. Gravar código no microcontrolador;
8. Se conectar na rede WiFi "ESP32" com a senha "12345678";
9. Acessar o endereço de ip "192.168.4.1";
10. Visualizar as ondas e interpretar sinais;