**Destilador Inteligente**

Sérgio Manuel Figueira do Carmo

**TRABALHO DE FINAL DE CURSO**

Curso: **Engenharia Informática**

Orientador: Sérgio Correia

Ano Letivo: 2023 | 2024

Setembro | 2024

**Agradecimentos**

.

**Resumo**

Este projeto, tem como finalidade modernizar um destilador. Este destilador, era completamente analógico, foi realizado uma modernização na máquina removendo-se todos os componentes antigos e substituídos por componentes digitais, tal como LEDs (*light emitting diode*), relé e um microprocessador chamado de ESP32.

Devido ao ESP32 ser um microprocessador potente, é assim possível ter o *firmware* e um servidor WEB em execução num único microprocessador.

O *firmware* controla todo o funcionamento da máquina, tal como a administração do tanque de água. O *firmware* é programado em linguagem de programação C

A página WEB, é uma interface intuitiva para controle e monitorização da máquina.

A página WEB é programada em linguagem de programação HTML, JavaScript e AJAX.

**Palavras-Chave**: Destilador, HTML, C, Arduino, HTML, JS, AJAX, ESP32.

**Abstract**

g.

**Keywords**: Lorem, Ipsum, Lorem, Ipsum, Lorem, Ipsum.

Lista de Abreviaturas, Siglas e Símbolos

AJAX - Asynchronous JavaScript and XML.

XML - Extensible Markup Language.

HTML - Hypertext Markup Language.

JS – JavaScript

CSS - Cascading Style Sheets

IDE - Integrated Development Environment

JSON - JavaScript Object Notation

SoC - System-on-Chip

SRAM - Static Random Acess Memory

LED – Light-emitting diode

AC - Alternating current

DC - Direct current

SPI - Serial Peripheral Interface

SRAM - *Static Random Acess Memory*

NC - N*ormally Closed*

NO - N*ormally Open*

*GPIO - General-Purpose Input Output*

ÍNDICE GERAL

[1. Lista de Abreviaturas, Siglas e Símbolos iv](#_Toc174995214)

[2. ÍNDICE GERAL vi](#_Toc174995215)

[3. ÍNDICE DE ANEXOS viii](#_Toc174995216)

[4. ÍNDICE DE FIGURAS ix](#_Toc174995217)

[5. ÍNDICE DE TABELAS x](#_Toc174995218)

[INTRODUÇÃO 2](#_Toc174995219)

[1. Enquadramento e Justificação do Tema 2](#_Toc174995220)

[2. Requisitos Gerais e Específicos 2](#_Toc174995221)

[2.1. Requisitos Gerais 2](#_Toc174995222)

[3.2. Requisitos específicos 2](#_Toc174995223)

[4. Metodologia e Meios Utilizados 4](#_Toc174995224)

[6. Limitações da Pesquisa/Trabalho 4](#_Toc174995225)

[7. Estrutura Geral do Trabalho 4](#_Toc174995226)

[8. Hardware 6](#_Toc174995227)

[8.1. Arquitetura Geral 6](#_Toc174995228)

[8.2. Destiladora 6](#_Toc174995229)

[8.3. ESP32 7](#_Toc174995230)

[8.4. CPU e Memória 8](#_Toc174995231)

[8.5. Inputs/Outputs 8](#_Toc174995232)

[8.5.1. Inputs 10](#_Toc174995233)

[8.5.2. Outputs 10](#_Toc174995234)

[8.6. Prototipagem em Placa de Ensaio 11](#_Toc174995235)

[8.7. Componente Eletromecânica 12](#_Toc174995236)

[8.7.1. Relés 12](#_Toc174995237)

[8.7.2. Filtro para os inputs 14](#_Toc174995238)

[8.7.3. Depósito e sensores de nível 15](#_Toc174995239)

[8.7.4. Bomba 16](#_Toc174995240)

[8.7.1. Válvulas Solenoide 16](#_Toc174995241)

[8.7.2. Fonte de Alimentação de 5VDC e Proteção 17](#_Toc174995242)

[8.8. Painel Frontal 18](#_Toc174995243)

[8.8.1. Indicadores LED 18](#_Toc174995244)

[8.8.2. Botões Frontais 18](#_Toc174995245)

[9. Software 20](#_Toc174995246)

[9.1. Página Web 20](#_Toc174995247)

[10. CONCLUSÃO 21](#_Toc174995248)

[11. REFERÊNCIAS BIBLIOGRÁFICAS 23](#_Toc174995249)

[12. ANEXOS 24](#_Toc174995250)

[12.1. Anexo 1 – “Esquema Elétrico” 24](#_Toc174995251)

[13.1. Anexo 2 – “Instruções de Uso” 25](#_Toc174995252)

ÍNDICE DE ANEXOS

[Anexo 1 Circuito Elétrico 24](#_Toc174995253)

[Anexo 2 Instruções de Uso 25](#_Toc174995254)

ÍNDICE DE FIGURAS

[Figura 1 – Arquitetura Geral da Destiladora 6](#_Toc174995255)

[Figura 2 - Destiladora 7](#_Toc174995256)

[Figura - 3 ESP32­WROVER­E 7](#_Toc174995257)

[Figura 4 – Layout dos pins do ESP32 10](#_Toc174995258)

[Figura 5 - ESP32 na Placa de Ensaio 11](#_Toc174995259)

[Figura 6 - Exemplos de Montagem de Circuito Elétrico 12](#_Toc174995260)

[Figura 7 - Relês e ESP32 13](#_Toc174995261)

[Figura 8 - Esquema Elétrico Relê Modo Manual 13](#_Toc174995262)

[Figura 9 - Esquema Elétrico do Filtro Passa-Baixo Passivo 14](#_Toc174995263)

[Figura 10 Equação Filtro Passa-Baixo 14](#_Toc174995264)

[Figura 11 - Esquema Elétrico do ESP32 com o Filtro Passa-Baixo Incorporado 15](#_Toc174995265)

[Figura 12 - Reservatório de Água 15](#_Toc174995266)

[Figura 13 - Digital Inputs 16](#_Toc174995267)

[Figura 14 - Bomba de Água 16](#_Toc174995268)

[Figura 15 - Válvula 17](#_Toc174995269)

[Figura 16 - Fonte 5VDC e proteção DC 17](#_Toc174995270)

[Figura 17 - Indicadores LED 18](#_Toc174995271)

[Figura 18 - Botões Frontais 19](#_Toc174995272)

ÍNDICE DE TABELAS

[Tabela 1 - ESP32 GPIO's 9](#_Toc174995273)

INTRODUÇÃO

Lorem, felis ut adipiscing.

1. Enquadramento e Justificação do Tema
2. Requisitos Gerais e Específicos
   1. Requisitos Gerais

1. Modo Automático

1.1. Liga/desliga a máquina com o temporizador

2. Modo Manual

2.1. Máquina sempre ligada em modo manual

3. Gestão de nível de água

3.1. Nível máximo desliga a bomba

3.2. Nível mínimo liga a bomba de água, liga resistência de calor e abre as válvulas

3.3. Nível de alarme desliga a resistência de calor, liga o led luminoso de alarme, fecha todas as válvulas

4. Controlo da destiladora via página WEB

* 1. Requisitos específicos

1. Uso do *ESP32 WROVER-E DevKitC V4 board*

1.1. Vantagens:

1.1.1. 26 I/O pins usáveis

1.1.2. WIFI integrado - 802.11n (2.4 GHz), up to 150 Mbps

1.1.3. CPU com 2 cores (core 0 corre *firmware* da máquina e core 1 o servidor WEB)

1.1.4. CPU trabalha a 40-MHz

1.1.5. *Firmware* desenvolvida em C

1.1.6. Arquitetura do CPU é *open* *source*, é possível programar ao nível do CPU se necessário, ao contrário se fosse usado, por exemplo, um raspberry pi

1.2. Desvantagens:

1.2.1. Memória para *firmware* + página WEB disponível muito limitada (8MB)

1.2.2. Por limitação de memória disponível no ESP32, velocidade de processamento e I/O disponíveis, apenas é possível fazer a página WEB em HTML usando CSS, e JavaScript, não havendo mais pins I/O disponíveis para ligar um módulo para cartão SD.

1.2.3. Limitação de CPU (40MHz)

1.3. Alternativas ao *ESP32 WROVER-E DevKitC V4*:

1.3.1. Raspberry pi

1.3.1.1. Uma opção boa seria usar um RP, teria muito mais memória e poder de processamento, mas os I/O não seriam suficientes (14 pins usáveis) para o projeto

1.3.2. Arduino Mega

1.3.2.1. O Arduino Mega tem muitos I/O (54 I/O), mas não tem WIFI e tem um CPU muito inferior (16MHz e 1 só core)

2. Página WEB

2.1. Página WEB vai ser desenvolvida em HTML, CSS, JavaScrip e AJAX

2.1.2. É usada a biblioteca “ESPAsyncWebServer.h” para criação e gestão de um servidor assíncrono WEB.

3. WIFI

3.1. É usada a biblioteca “WiFiManager.h” para gestão do WIFI

3.1.1. Com esta biblioteca, se não é conhecida a rede WIFI, ela cria um AP (*access point*) para o utilizador poder escolher, ligar e guardar as credenciais da nova rede WIFI na memória EEPROM e vai ler as credenciais quanda a máquina é ligada para se ligar à nova rede.

1. Metodologia e Meios Utilizados

O *firmware* da destiladora é programado em linguagem C, inclui os ficheiros de inclusão (ficheiros *header* com a extensão .h). A página *WEB* é programada em HTML com os respetivos ficheiros de CSS para adicionar estilos e os ficheiros JS, para executar JavaScript.

Todo o projeto foi elaborado no IDE *Visual Studio Code* com as extensões *PlatformIO IDE, GitHub Copilot* e *Doxygen.*

Todo o projeto foi também elaborado no sistema operativo Fedora 40 (Linux).

Limitações da Pesquisa/Trabalho

Não encontrei muitas dificuldades/limitações na pesquisa acerca do trabalho, a comunidade Arduino é muito grande e por consequência, há muita documentação disponível, talvez por haver tanta documentação, tenha havido uma certa dificuldade em filtrar o que realmente se adequasse ás minhas necessidades, tal como o método de comunicação (AJAX) entre o *firmware* e a página WEB.

Estrutura Geral do Trabalho

Na pasta raiz, estão as pastas “data”, “Documentation”, “include”, “lib” e “src”.

* data - Tem tudo o que é relacionado com a página de internet, tal como os ficheiros HTML, CSS e JS.
* Documentation – Contém tudo o que está relacionado com a documentação, tal como o relatório.
* Include – Contém todas os ficheiros *headers* necessários para o funcionamento do *software*.
* Lib – Esta pasta, contém as bibliotecas necessárias para o *software*.
* Src – Contém os ficheiros ‘.cpp’
* Na pasta raiz, estão ainda 2 ficheiros necessários para o funcionamento:
  + “partitions.csv”, que contém a configuração das partições da memória do ESP32.
  + “platformio.ini”, contém a configuração da extensão PlatformIO.

Hardware

## Arquitetura Geral

É representado o funcionamento da destiladora pelo esquema de processos representados pela Figura 1, esta é constituído por um depósito com água destilada em que é alimentado através de uma bomba de água (BMB) e controlado com três sensores de nível de água (SMIN, SMAX e ALARM), dentro do depósito há uma resistência de calor (RAQ) que vai evaporar a água destilada. O vapor é controlado por uma válvula elétrica (V2) que direciona o vapor para destilar a amostra. Uma segunda válvula elétrica (V1) que abre para entrar água fria para o condensador, esta vai arrefecer o vapor vindo da amostra e condensar. Há também uma terceira válvula (V3) que é manual, esta serve para despejar a água destilada existente no depósito.

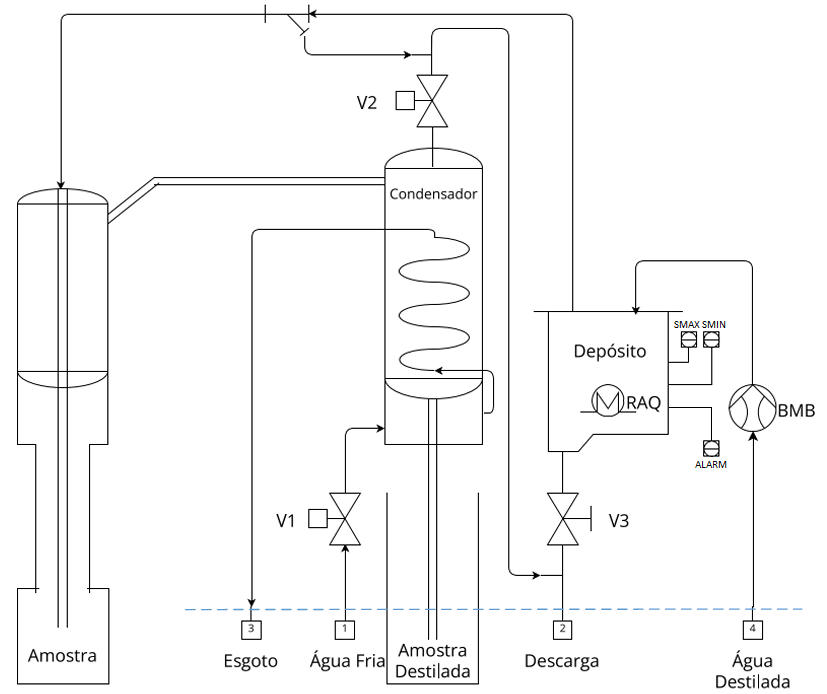


Figura 1 – Arquitetura Geral da Destiladora

## Destiladora

Foi usada a destiladora mostrada na

Figura 2, sendo esta uma destiladora totalmente analógica, constituída por um temporizador, indicadores, botões, operado através da rede elétrica 230V/50Hz, foram removidos alguns componentes como o temporizador analógico e cabos, estes foram trocados por componentes digitais, tal como relés, indicadores LED, um módulo para processamento ESP32 WROVER-E, fonte de alimentação de 230VAC/50Hz para 5V DC.

Uma imagem com máquina, texto, Eletrodoméstico, interior

Descrição gerada automaticamente

Figura 2 - Destiladora

## ESP32

O módulo de processamento usado no projeto é um *ESP32 WROVER-E* [1], representado na

Figura - 3, o qual é um SoC (*System on a Chip*) desenvolvido pela empresa *Espressif Systems* [1](uma empresa Chinesa com sede em Xangai e é fabricado pela TSMC), é um microcontrolador de baixo custo e baixo consumo de energia. Tem integrado, Wi-Fi, Bluetooth e tem antena RF (Radio Frequência) integrada*.*

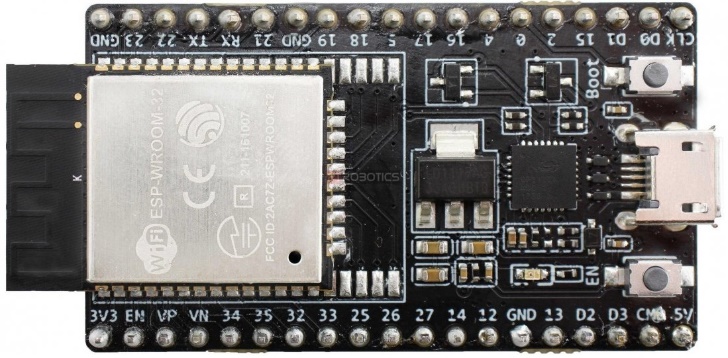


Figura - 3 ESP32­WROVER­E

O microcontrolador ESP32 é compatível com a linguagem de programação Arduino (linguagem de programação C/C++), por isso, pode ser programado pelos mesmos IDE’s, no caso deste mesmo projeto, foi programado na plataforma *Visual Studio Code* [2] com o plug-in *PlatformIO* [3]em linguagem de programação C.

## CPU e Memória

O ESP32 tem um processador com 2 núcleos de 40nm que podem ser controlados individualmente e a frequência pode ser ajustada entre 80MHz e 240MHz [4, p. 7], 520KB de SRAM (*Static Random Acess Memory*) e 4MB de memória FLASH. O módulo tem também incorporado um módulo de Wi-fi e Bluetooth com antena incorporada, sendo o Wi-Fi capaz de suportar os protocolos 802.11 b/g/n (802.11n até 150Mbps).

A memória SRAM [5] é um tipo de memória volátil usada em sistema computacionais, sendo que esta memória usa circuitos *flip-flop* que assegura que os dados fiquem presentes enquanto seja alimentada com energia elétrica. Este tipo de memória tem a vantagem de oferecer alta velocidade de acesso e ser energeticamente muito eficiente.

No caso da memória FLASH, é uma memória não volátil que pode ser apagada e reprogramada, que serve para armazenar o *firmware* desenvolvido, esta memória, pode ser dividida em partições e as partições serem redimensionadas às necessidades do programador, no caso deste projeto, foi particionado da seguinte forma:

Partição para APP – 3145728 bytes,

Partição para SPIFFS – 917504 bytes

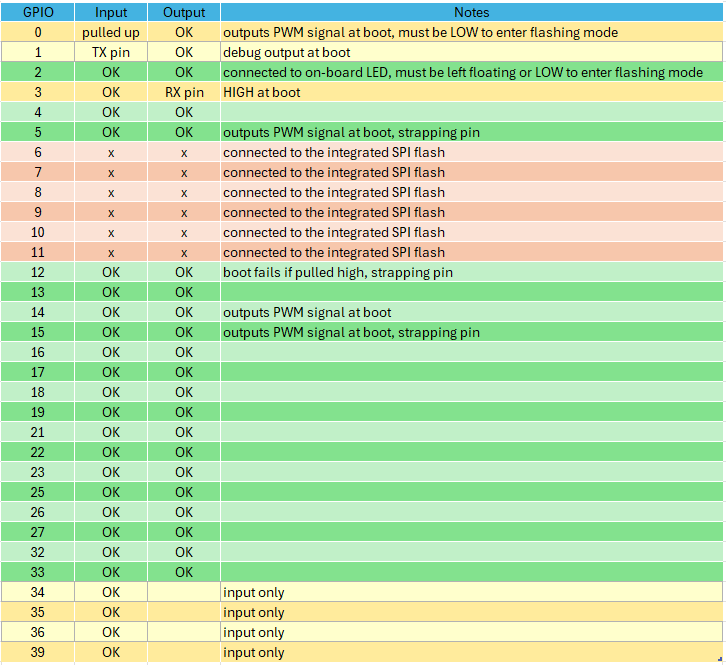
A partição APP é onde o *firmware* vai ser armazenado e a partição SPIFFS [6] é onde podem ser armazenados ficheiros diversos, no caso deste projeto, é armazenado todos os ficheiros relacionados com a página WEB (\*.html, \*.css, \*.js, \*.png)

## Inputs/Outputs

Os GPIO’s são um conjunto de pinos digitais controlados pelo microprocessador, estes pinos, podem ser definidos como *inputs* para ler dados de por exemplo sensores, botões, etc, para serem processados e também podem ser definidos como *output* para depois de processados, definir o estado de um componente, como por exemplo, um motor, válvula, etc. (ver Figura 4)

O módulo tem 32 GPIO’s dos quais 19 são completamente usáveis como é constatado pela Tabela 1 pelos tons de cor verde, estes GPIO’s podem ser usados tanto para *inputs* como para *outputs*. Os GPIO’s 0 e 1 não podem ser usados como *inputs* e os GPIO’s 3, 34, 34-36 e 39 não podem ser usados como *outputs* (tons de cor amarelo na Tabela 1). Os GPIO’s de 6 até 11 são reservados para interação com a memória FLASH, não podendo ser usados como I/O (tons de cor vermelho na Tabela 1).

Tabela 1 - ESP32 GPIO's



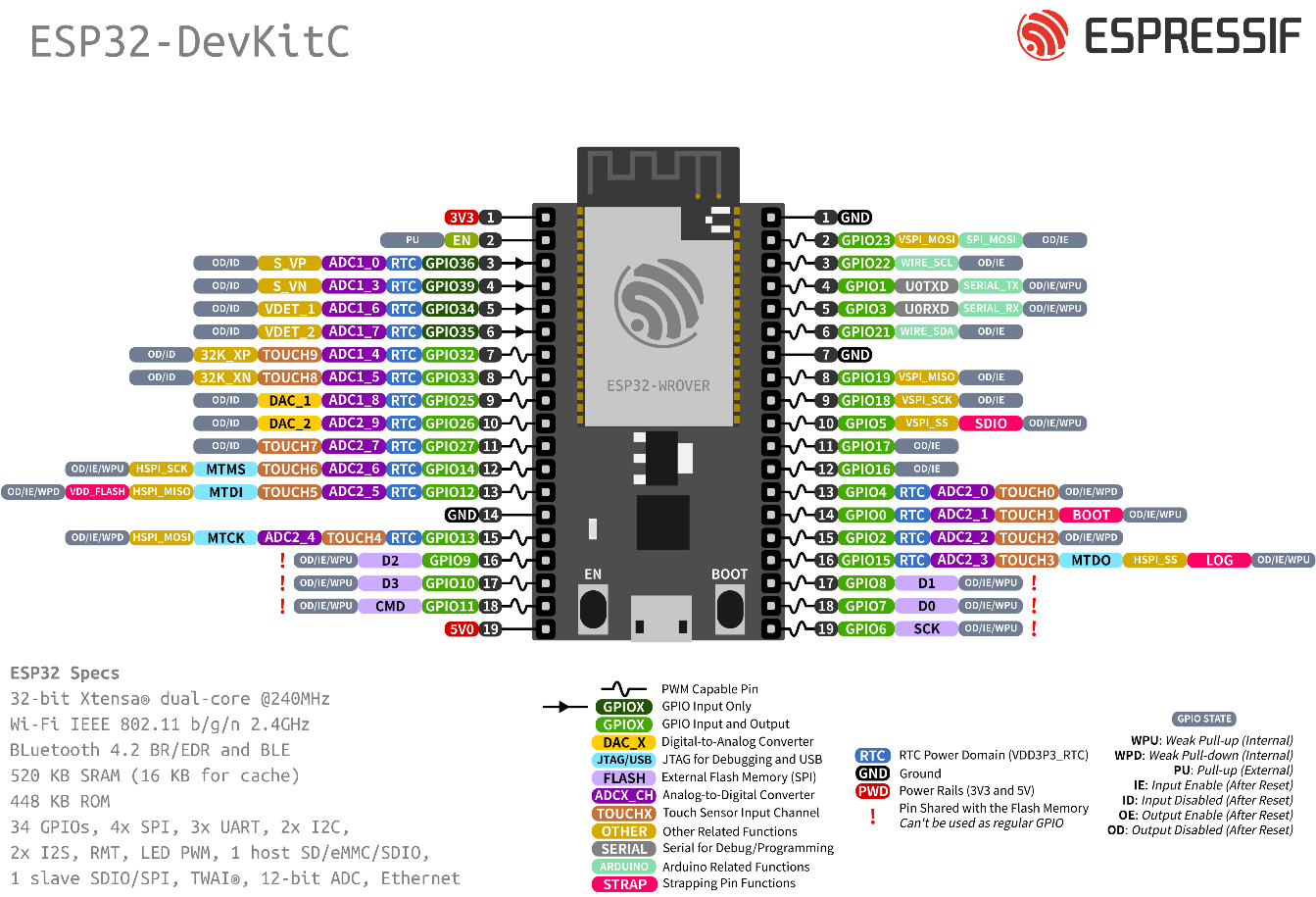


Figura 4 – Layout dos pins do ESP32

Foram usados cinco *inputs* (ver Tabela 2) e foram usados nove *outputs* (ver Tabela 3) de um total de dezanove GPIO’s, os I/O’s são declarados no *header* “OS.h”.

Tabela 2 - GPIO INPUTS



Tabela 3 - GPIO OUTPUTS



## Prototipagem em Placa de Ensaio

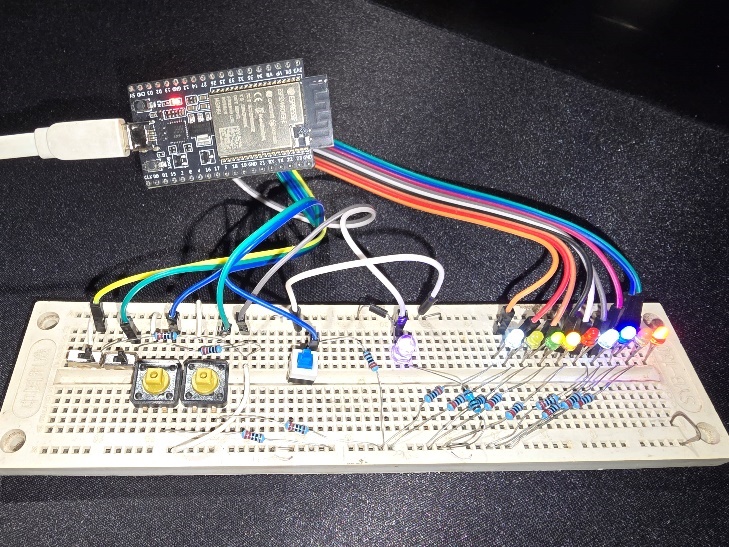


Figura 5 - ESP32 na Placa de Ensaio

Para desenvolver o *Software* em casa, foi simulado toda a destiladora em uma placa de ensaio com botões, interruptores e indicadores LED para simular os I/Os da máquina real, em que os botões e interruptores, simulam os *inputs* e os indicadores LED os *outputs*.

## Componente Eletromecânica

Uma imagem com máquina, interior, eletrónica, Engenharia eletrónica

Descrição gerada automaticamente

Figura 6 - Exemplos de Montagem de Circuito Elétrico

O que saiu e entrou

O circuito elétrico antigo e os seu completamente analógico foi removido e substituído por um novo e digital, todos os cabos foram substituídos, como mostrado pela Figura 6, mas foram aproveitados alguns componentes, tal como os botões, válvulas, bomba de água e resistência de calor.

### Relés

Para controlo destes componentes, foram adicionados 2 módulos com 4 relês cada (ver

Figura 7), o módulo da esquerda, está dedicado a controlar os componentes de 250V como a bomba de água, as 2 válvulas e a resistência de calor e o módulo da direita, controla os indicadores LED frontais (nível de água máximo, mínimo, alarme e modo automático), as bobines recebem 5V do microcontrolador para ativar o respetivo *output*. Cada relê é capaz de operar nos seus contactos 250VAC a 10A

O relê do interruptor de modo manual, a bobine é acionada com 230VAC e os dois contactos, 250VAC a 8A cada.

Uma imagem com Fios elétricos, Engenharia eletrónica, eletrónica, cabo

Descrição gerada automaticamente

Figura 7 - Relês e ESP32

Para aproveitar o botão de manual, que tem um indicador embutido de néon de 250V, foi adicionado um relê dedicado somente ao botão de modo manual, em que ao ser ligado, aciona o relê para o microprocessador poder receber essa informação (ver Figura 8).

Uma imagem com diagrama, texto, file, Paralelo

Descrição gerada automaticamente

Figura 8 - Esquema Elétrico Relê Modo Manual

### Filtro para os inputs

Como se está a trabalhar com alta tensão de onda alternada (~240V AC) juto com tensão baixa continua (5V DC) e o microprocessador é muito sensível a ruido nos sinais de *input*, foi detetado nos testes em que ao ligar/desligar o botão de modo manual, os *outputs* aleatoriamente trocavam de estado sem que fosse propositado, depois de muitos testes, foi detetado com recurso ao osciloscópio, que ao ser acionado botão de modo manual, causava ruido de alta frequência nos I/Os, já que os fios agem como antenas aos campos eletromagnéticos gerados pela alta tensão nos fios e pelas bobines dos relês [7].

Depois de pesquisado, foi encontrado um vídeo no Youtube [8] que representa o mesmo problema que estava presente no circuito da destiladora, foi a partir desse mesmo vídeo, que foi estudado o problema e resolvido com a implementação de um filtro passa-baixo passivo (ver Figura 9), que visa eliminar as altas frequências geradas pelos componentes e fios que operam em alta tensão

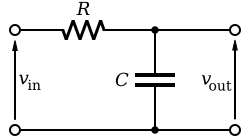


Figura 9 - Esquema Elétrico do Filtro Passa-Baixo Passivo

Foi então aplicada a equação representada na Figura 10 e como foi usado (ver Figura 11) um condensador de tântalo de 220nF e uma resistência de 47 OHMs, o resultado da equação é aproximadamente ~15392Hz, esta é a frequência máxima que o filtro permite.

Foi adicionado um filtro para cada *input* e também adicionado também em paralelo um condensador eletrolítico de 16V e 1000uF e um condensador de tântalo de 220nF para ajudar na filtragem ligados diretamente na saída da fonte de alimentação.



Figura 10 Equação Filtro Passa-Baixo

Uma imagem com texto, diagrama, Esquema, file

Descrição gerada automaticamente

Figura 11 - Esquema Elétrico do ESP32 com o Filtro Passa-Baixo Incorporado

### Depósito e sensores de nível

O vapor de água destilada é gerado dentro de um depósito (ver Figura 12) em inox. Este depósito incorpora uma resistência de calor e três sensores de nível de água (ver Figura 13).

Os sensores de nível, estão localizados na lateral do depósito através de furações feitos na parede do depósito e estão seguros com porcas M14 impressas no laboratório FabLab com uma impressora 3D de resina.

Uma imagem com máquina, interior, lavatório, ferramenta

Descrição gerada automaticamente

Figura 12 - Reservatório de Água

Uma imagem com texto, diagrama, file, Esquema

Descrição gerada automaticamente

Figura 13 - Digital Inputs

### Bomba

A água é bombeada por uma bomba do tipo oscilante que opera a 230VAC/50Hz e consome 0.17A representada na Figura 14 e é controlada por um relê (ver 8.7.1 Relés).

Ver “Anexo 1 Circuito Elétrico” para mais detalhes sobre marca e modelo.

Uma imagem com texto, pessoa

Descrição gerada automaticamente

Figura 14 - Bomba de Água

### Válvulas Solenoide

Existem 2 válvulas (ver Figura 15) solenoide com tamanho 1/8” acionadas a 240VAC que são NC (*normally closed*).

Uma controla a entrada de água para arrefecer a serpentina do condensador de vapor e a outra, controla a a saída de vapor.

Ver “Anexo 1 Circuito Elétrico” para mais detalhes sobre marca e modelo.

Uma imagem com Fios elétricos, cabo, máquina, Engenharia eletrónica

Descrição gerada automaticamente

Figura 15 - Válvula

### Fonte de Alimentação de 5VDC e Proteção

É usada uma fonte de alimentação de 5VDC tensão e 2A de corrente elétrica para fornecer energia ao microprocessador e componentes de baixa voltagem.

O circuito de 5VDC está também protegido por meio de um fusível de 1A ligado em série com o circuito de baixa tensão.

Foi adicionado um disjuntor bipolar de 16Amperes para proteção do circuito elétrico de 230VAC.



Figura 16 - Fonte 5VDC e proteção DC

## Painel Frontal

### Indicadores LED

Existem quatro indicadores LED (ver Figura 17) no painel frontal da máquina, são todos acionados com 5VDC e são todos operados pelo 2º módulo de relês (ver 8.7.1 - Relés). São usados dois indicadores de cor laranja, um de cor verde e 1 de cor vermelho.

O modo automático, usa um indicador de cor laranja, o indicador verde, está responsável pelo nível máximo de água no reservatório, o nível mínimo, tem cor laranja e o nível de alarme, tem cor vermelho e acende intermitente para destacar a prioridade de atenção.

Uma imagem com texto, captura de ecrã, círculo

Descrição gerada automaticamente

Figura 17 - Indicadores LED

### Botões Frontais

O interruptor de “*POWER*” é o que liga/desliga toda a máquina, é um interruptor bi-polar, isto é, interrompe a passagem de corrente elétrica na fase e neutro, tem um indicador incorporado de néon de 250VAC e quando pressionado, fica na posição que foi pressionado.

O botão “Auto.” é um botão “*push-button*”, só passa energia quando pressionado, depois de ser pressionado, volta ao estado desligado em que não passa energia. Ao ser pressionado, ativa/desativa o modo automático

O interruptor “Man.” é idêntico ao interruptor de “*POWER*”, tem também um indicador néon incorporado e vai ligar um relê (ver 8.7.1 - Relés) para ativar/desativar o modo manual.

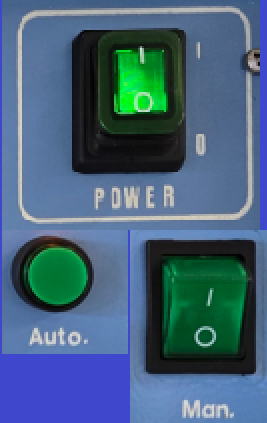


Figura 18 - Botões Frontais

Software

## Página Web

CONCLUSÃO

Lorem.

REFERÊNCIAS BIBLIOGRÁFICAS

|  |  |
| --- | --- |
| [1] | Espressif, “Espressif-About,” [Online]. Available: https://www.espressif.com/en/company/about-espressif. [Acedido em 08 2024]. |
| [2] | Microsoft, “Visual Studo Code,” [Online]. Available: https://code.visualstudio.com/. [Acedido em 08 2024]. |
| [3] | P. Labs, “PlatformIO,” [Online]. Available: https://platformio.org/. [Acedido em 08 2024]. |
| [4] | espressif, “espressif ESP32,” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e\_esp32-wrover-ie\_datasheet\_en.pdf. [Acedido em 08 2024]. |
| [5] | L. Harvie, “Embedded Systems Memory Types: Flash vs SRAM vs EEPROM,” [Online]. Available: https://medium.com/@lanceharvieruntime/embedded-systems-memory-types-flash-vs-sram-vs-eeprom-93d0eed09086. [Acedido em 08 2024]. |
| [6] | espressif, “SPIFFS Filesystem,” [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/spiffs.html. [Acedido em 08 2024]. |
| [7] | V. Muthukrishnan, “electrical4u,” 27 05 2024. [Online]. Available: https://www.electrical4u.com/electromagnetic-interference/. [Acedido em 08 2024]. |
| [8] | M. Nardi, “Elimine INTERFERÊNCIAS dos seu projetos feitos com ARDUINO!,” YouTube, 2022. [Online]. Available: https://www.youtube.com/watch?v=uyItIeIbdCg. [Acedido em 08 2024]. |
| [9] | A. Ahmad, “Calculating RC Low-Pass Filter Cut-Off Frequency and Transfer Function,” 17 07 2023. [Online]. Available: https://eepower.com/technical-articles/calculating-rc-low-pass-filter-cut-off-frequency-and-transfer-function/. [Acedido em 08 2024]. |
| [10] | E. Wings, “Digital GPIO of Arduino,” [Online]. Available: https://www.electronicwings.com/arduino/digital-gpio-of-arduino. [Acedido em 08 2024]. |

ANEXOS

## Anexo 1 – “Esquema Elétrico”



Anexo 1 Circuito Elétrico

## Anexo 2 – “Instruções de Uso”



Anexo 2 Instruções de Uso