



**INSTITUTO
FEDERAL**

Paraíba

Campus
Campina Grande

Instituto Federal de Educação Ciências e Tecnologia da Paraíba

Campus Campina Grande

Bacharelado em Engenharia de Computação

RELATÓRIO DO PROJETO

GRUPO 5 - SISTEMA DE AUTOMAÇÃO HIDROPONIA 2

Sistemas Embarcados

Professor: Alexandre Sales Vasconcelos

EQUIPE: KILVIA CARVALHO
MICAEL MARQUES
PAMELLA LIMA

Campina Grande,
2022

Resumo

Este projeto tem como objetivo apresentar um sistema de automação de irrigação de baixo custo baseado no princípio de Internet das Coisas (IoT) para monitoramento de temperatura da água e volume do reservatório de água, além da implementação de um controle sem fio para a bomba d'água. O hardware do sistema é composto por um microcontrolador ESP32 que controla os sensores HX711 para monitoramento do volume do reservatório e o DS18B20 para monitoramento da temperatura da água. O controle de estados e o gerenciamento destes dispositivos ocorre por meio de uma implementação web para ligar e desligar a bomba, monitoramento do volume do reservatório e monitoramento da temperatura da água, além de enviar mensagens de status para o Telegram.

Palavras-chaves: ESP32; IoT; HX711; DS18B20; Temperatura; Volume.

Abstract

This project aims to present a low-cost irrigation automation system based on the Internet of Things (IoT) principle for monitoring water temperature and water reservoir volume, in addition to implementing a wireless control for the water pump. Water. The system hardware consists of an ESP32 microcontroller that controls the HX711 sensors for monitoring the reservoir volume and the DS18B20 for monitoring the water temperature. The status control and management of these devices takes place through a web implementation to turn the pump on and off, monitoring the volume of the reservoir and monitoring the water temperature, in addition to sending status messages to Telegram.

Keywords: ESP32; IoT; HX711; DS18B20; Temperature; Volume.

SUMÁRIO

Resumo	2
Abstract	2
1. INTRODUÇÃO	4
2. OBJETIVO	4
3. MATERIAIS E MÉTODOS	4
3.1 LISTA DE MATERIAIS	4
3.1.1 ESP32	4
Figura 1: ESP32	5
3.1.2 HX711	5
Figura 2: HX711	5
3.1.3 Célula de Carga 50KG	5
Figura 3: Célula de Carga	6
3.1.4 Sensor de Temperatura DS18B20	6
Figura 4: Sensor de Temperatura DS18B20	6
3.2 LISTA DE BIBLIOTECAS UTILIZADAS	7
3.3 ATIVIDADES EXECUTADAS	7
Figura 5: Componentes Utilizados	7
Figura 6: Testes dos Componentes	8
3.4 ORGANIZAÇÃO DO SISTEMA	8
3.4.1 HARDWARE	8
Figura 7: Diagrama de blocos do Hardware	8
Figura 8: Diagrama do Protótipo	9
Figura 9: Esquema Elétrico	9
3.4.2 FIRMWARE	10
Figura 10: Fluxograma do Firmware	10
3.4.3 O APLICATIVO	11
Figura 11: Página inicial do sistema WEB	11
4. RESULTADOS	12
Figura 12: Protótipo Final	12
Figura 13: Página WEB com o resultado do volume	13
Figura 14: Página WEB com o resultado da temperatura	13
Figura 15: Página WEB com o resultado do acionamento do led	14
Figura 16: Led aceso após seu acionamento (Figura 15)	14
Figura 17: Página WEB para o envio do status ao Telegram	15
Figura 18: Staus no Telegram	15
5. CONCLUSÃO	17
6. REFERÊNCIAS	18

1. INTRODUÇÃO

Com a ascensão de novas tecnologias computacionais e de automação, unidas a consolidação de paradigmas como IoT e WSN, é cada vez mais comum que tais mecanismos sejam combinados para realizar o monitoramento de ambientes e o controle de objetos de forma remota. Dentro deste escopo, vem tornando-se frequente o uso de dispositivos microcontroladores ligados a sensores de temperatura e volume com a finalidade exclusiva de monitorar ambientes onde a temperatura é um parâmetro importante e exerce influência direta na segurança no modo de operação de equipamentos, seja em ambientes industriais ou agrícolas, proporcionando assim uma diretiva efetiva e econômica. Tal ação é igualmente vista em ambientes de centros de processamento de dados no intuito de prover seu gerenciamento térmico, garantindo a segurança destes locais e o total funcionamento das máquinas presentes na área.

Dessa forma, o presente projeto consiste na criação de um dispositivo capaz de coletar dados referentes a temperatura da água e o volume do reservatório, como também o controle da bomba de água de forma remota, a fim de se poder realizar análises no sistema instalado.

2. OBJETIVO

O objetivo do projeto é coletar os dados dos sensores de temperatura da água e volume do reservatório, onde serão monitorados através de uma implementação WEB com controle da bomba e envio de status por Telegram, com capacidade de exibir dados em tempo real enquanto online, realizar a leitura, o monitoramento e o controle de dados que estão sendo recebidos.

3. MATERIAIS E MÉTODOS

Nesta seção são descritas as atividades executadas no desenvolvimento de cada etapa do projeto, os principais componentes do sistema e como estes encontram-se organizados para comunicarem-se entre si, a partir de diagramas de bloco e fluxogramas para uma melhor compreensão.

3.1 LISTA DE MATERIAIS

3.1.1 ESP32

O módulo ESP32 é um módulo de alta performance para aplicações envolvendo WiFi com pilha TCP completa e capacidade microcontroladora produzida pela empresa chinesa Espressif System, contando com um baixíssimo consumo de energia. É uma evolução do já conhecido ESP8266, com maior poder de processamento e bluetooth BLE 4.2 embutido. É combinado com um conector micro-USB com interface USB-Serial para

acoplamento a computadores, um regulador de tensão para 3.3 V e um conversor analógico-digital para compor a placa NodeMCU.

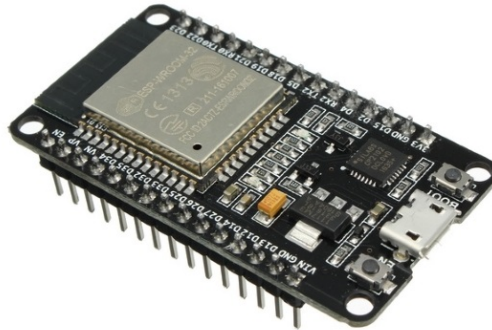


Figura 1: ESP32

3.1.2 HX711

O HX711 é um módulo amplificador operacional de 24 bits que será utilizado para converter o sinal analógico em digital, além de amplificar a saída para que seja reconhecida pelo Arduino. Ele foi criado com a intenção de atuar com sensores de peso ligados em ponte, sendo que o multiplexador interno seleciona a entrada diferencial (A+ e A-) ou (B+ e B-) para o amplificador de ganho.

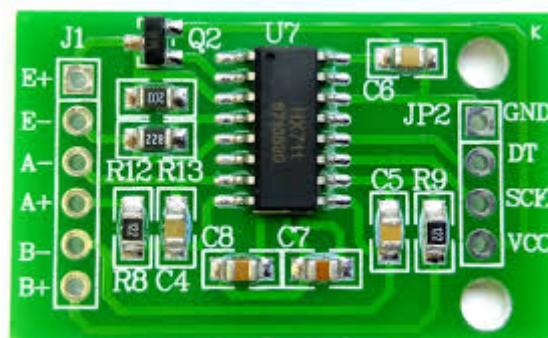


Figura 2: HX711

3.1.3 Célula de Carga de 50KG

A Célula de Carga, ou Sensor de Peso é um componente que trabalha em conjunto com o Sensor HX711 para realizar medições, no nosso caso, de até 50Kg por célula, podendo trabalhar com diversas células ao mesmo tempo para fazer medições de 100 Kg, 150 Kg, 200 Kg e assim por diante.

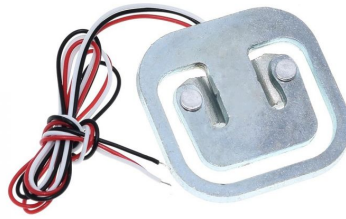


Figura 3: Célula de Carga

3.1.4 Sensor de Temperatura DS18B20

O Módulo DS18B20 Sensor de Temperatura Digital Dallas é um componente eletrônico digital desenvolvido para ser aplicado nos mais diversos ambientes, medindo temperaturas entre -55°C a $+125^{\circ}\text{C}$. Como diferencial possibilita fazer leituras com resolução entre 9 a 12-bit, a qual é configurável, além de possuir uma interface de comunicação simples por meio de um fio único, função conhecida como 1-Wire, que permite ligar vários sensores de temperatura em uma única saída digital.

No nosso caso, como precisamos do sensor para medir a temperatura da água, utilizamos uma versão desse sensor à prova de água. Esse sensor é indicado para medir aplicações onde é necessário medir a temperatura a uma distância maior do microcontrolador ou em ambientes úmidos.



Figura 4: Sensor de Temperatura DS18B20

3.2 LISTA DE BIBLIOTECAS UTILIZADAS

HX711.h
DS18B20.h

Do HX711.h utilizamos as funções *hx711_wait()*, que inicializa o sensor, *hx711_read_average()*, que seta o retorno do volume que será retornado ao usuário.

Do DS18B20.h utilizamos as funções *ds18b20_getTempF()* e *ds18b20_getTempC()*, que, respectivamente, retornam o valor da temperatura em graus fahrenheit e em graus celsius.

3.3 ATIVIDADES EXECUTADAS

Para esse prototipo, utilizamos os componentes HX711 e o DS18B20, que foram apresentados nas seções anteriores.

O componente HX711 utilizado funciona como um amplificador da célula de carga e é necessário utilizá-lo para aumentar a amplitude do sinal de entrada da célula de carga em relação à saída. Por isso, esse componente servirá como um amplificador de sinal.

Já o DS18B20 é um sensor de temperatura digital inteligente que no nosso projeto é utilizado para medir a temperatura da água do reservatório. Utilizamos um sensor à prova d'água porque o sensor ficará sempre acoplado ao reservatório.

Na estrutura do projeto foi criada uma pasta para cada componente contendo seu .h (arquivo que contém a exposição das funções codificadas no .c) e o arquivo .c.

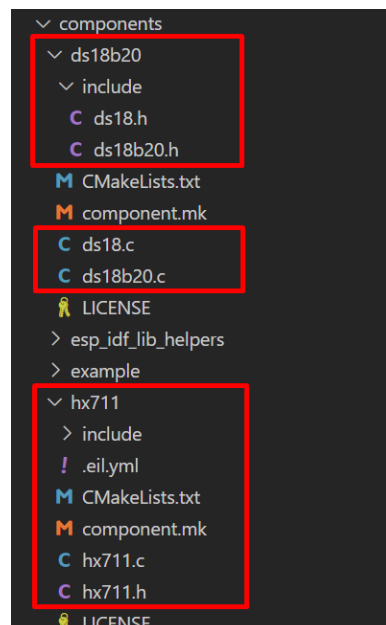


Figura 5: Componentes Utilizados

Na classe *main.c*, foram importadas as bibliotecas, criadas todas as variáveis, constantes e setadas as pinagens necessárias para o monitoramento dos componentes.

Inicialmente testamos os componentes de forma isolada de acordo com a montagem da figura 12.

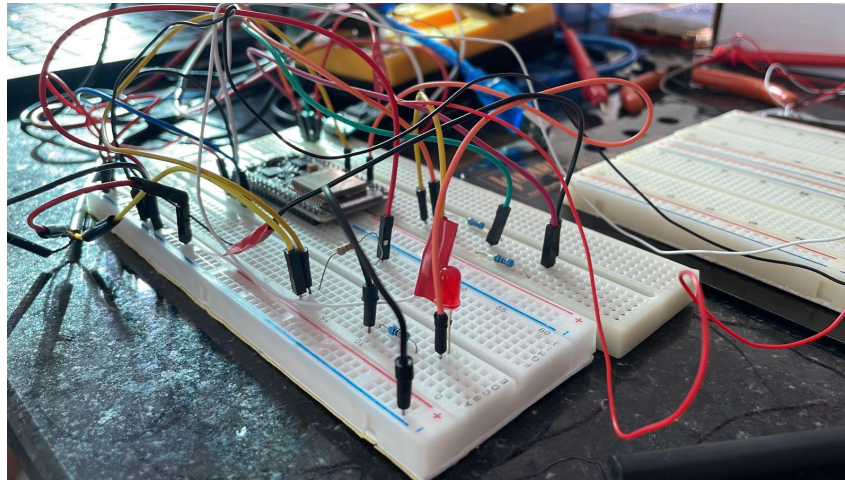


Figura 6: Testes dos Componentes

Para monitoramento da temperatura, volume e estado do acionamento da bomba, foi criado um bot no Telegram, usando a api fornecida pelos mesmos.

3.4 ORGANIZAÇÃO DO SISTEMA

3.4.1 HARDWARE

No ambiente de desenvolvimento do hardware, o microcontrolador ESP32 é alimentado por uma bateria de 3.3V. Nos pinos D2 e D4 é conectado o sensor de peso HX711 e no pino D26 é conectado o sensor de temperatura DS18B20, onde esses sensores ficam mandando dados para os ESP de suas medições. Como podemos observar o diagrama de blocos da Figura 5

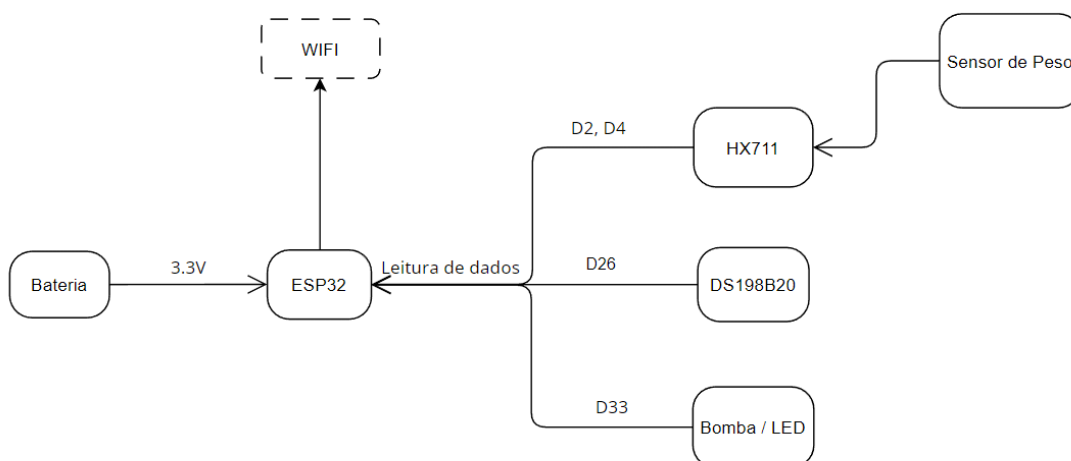


Figura 7: Diagrama de blocos do Hardware

A figura abaixo apresenta o protótipo real.

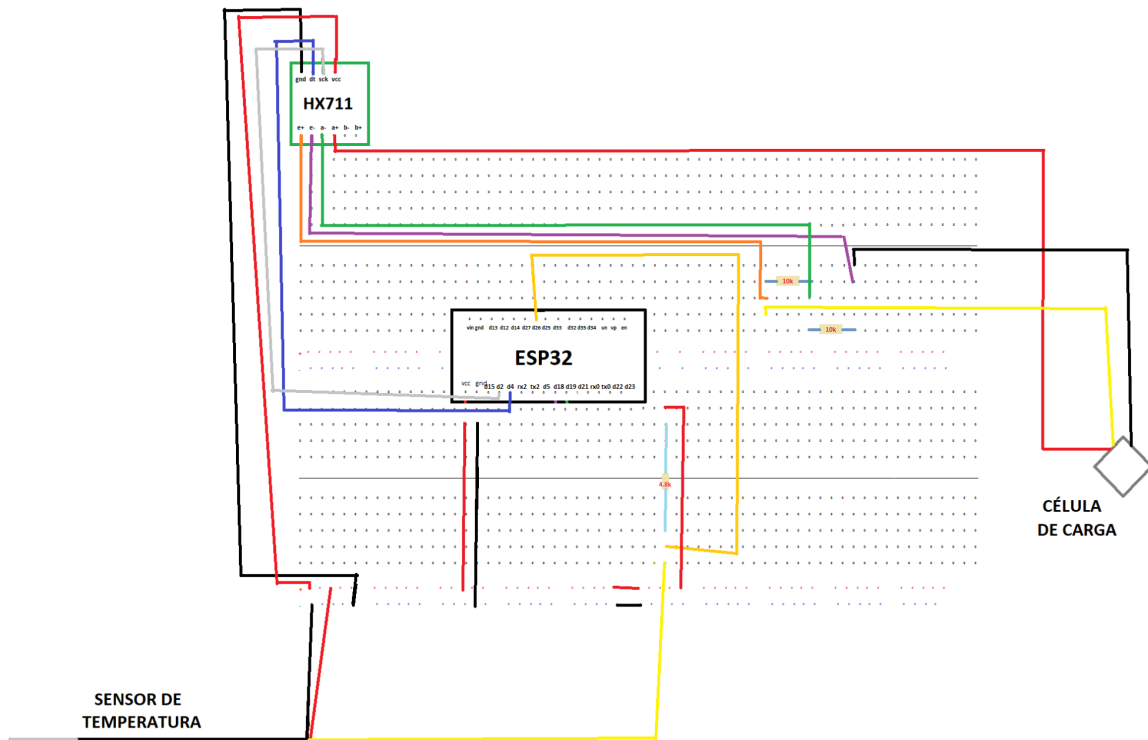


Figura 8: Diagrama do Protótipo

Por fim, apresentamos o esquema elétrico (Figura 9). Foi feita utilizando a ferramenta FUSION 360 para sua criação.

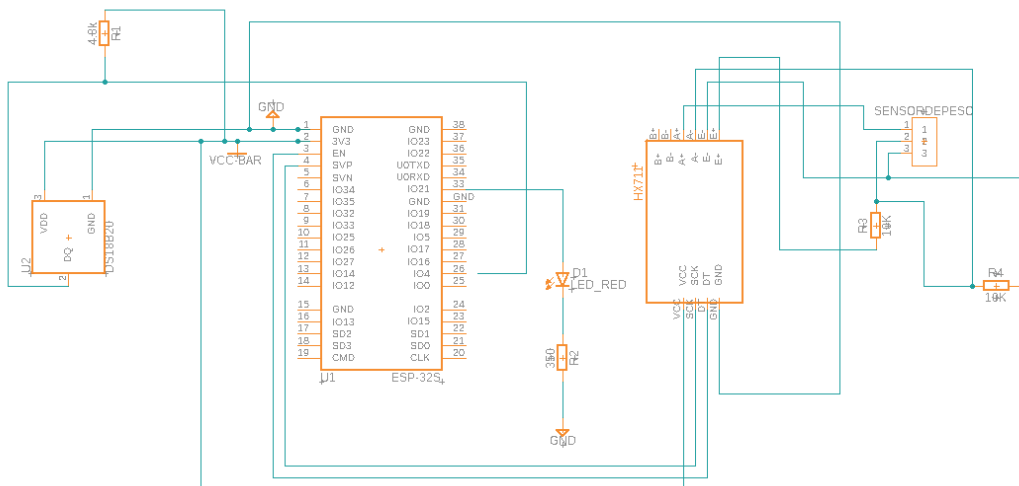


Figura 9: Esquema Elétrico

3.4.2 FIRMWARE

O Firmware é organizado basicamente da seguinte forma:

- A pasta **lib** contém os códigos de nossa autoria que são responsáveis pela leitura dos sensores.
- A pasta **main** contém o código principal do projeto e o que importa para os outros códigos serem utilizados.

Para entender melhor, temos o fluxograma do firmware (Figura 7) onde inicialmente o ESP é conectado com a rede WIFI. Em seguida é instanciado o código criado dentro da pasta lib, que é responsável pelas leituras dos sensores de temperatura e peso. No *app_main* é realizada a inicialização do WIFI. E no *while(1)* é realizado as leituras dos sensores de temperatura e peso. Após guardar em variáveis essas medidas, o código poderá enviar para a página WEB o status do sistema, as variáveis de temperatura e peso, e o estado da bomba, se ligado ou desligado.

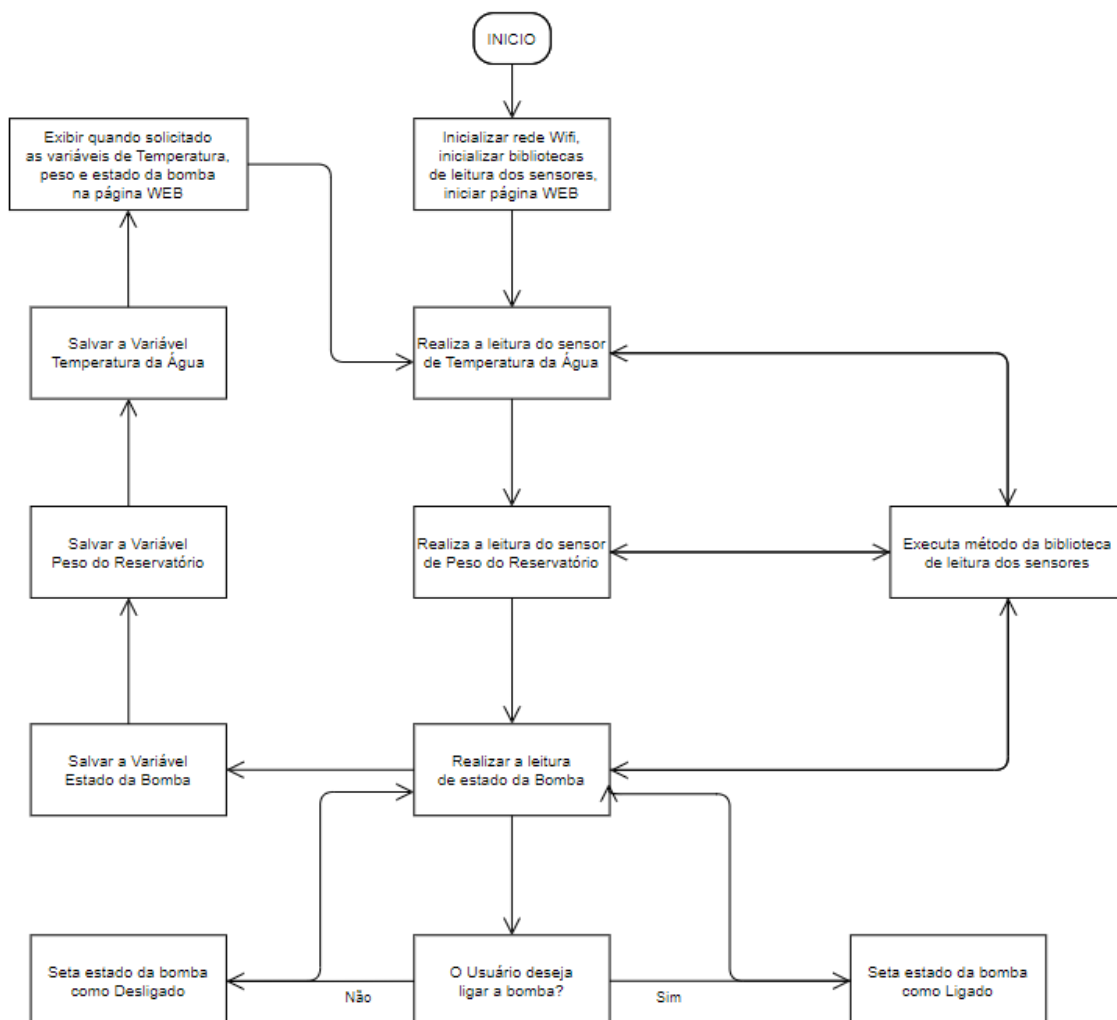


Figura 10: Fluxograma do Firmware

3.4.3 O APLICATIVO

Como aplicativo para controle, ele foi feito utilizando HTML em junção com a linguagem C, e temos a imagem abaixo, onde ele tem acesso e mostra o volume do reservatório, a temperatura da água, o controle para ligar e desligar a bomba, e um botão para enviar um status do projeto para o Telegram, enviando todos os dados acima, e se a bomba está ou não ligada. A figura abaixo exemplifica isso.

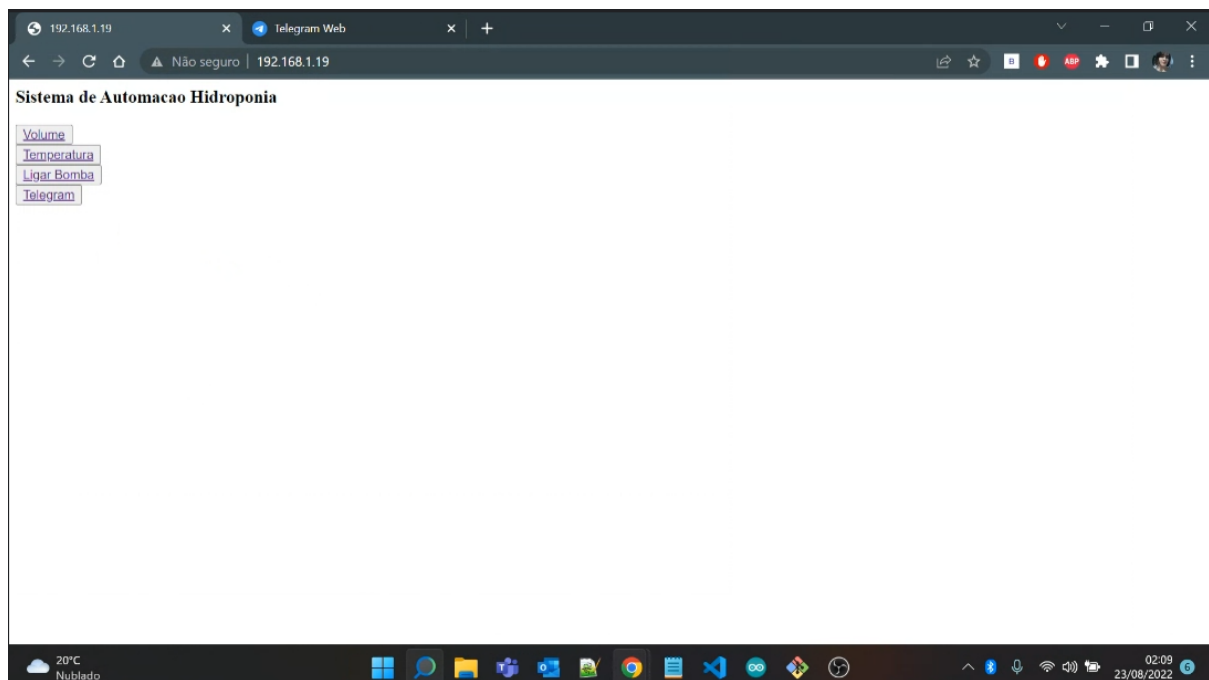


Figura 11: Página inicial do sistema WEB

4. RESULTADOS

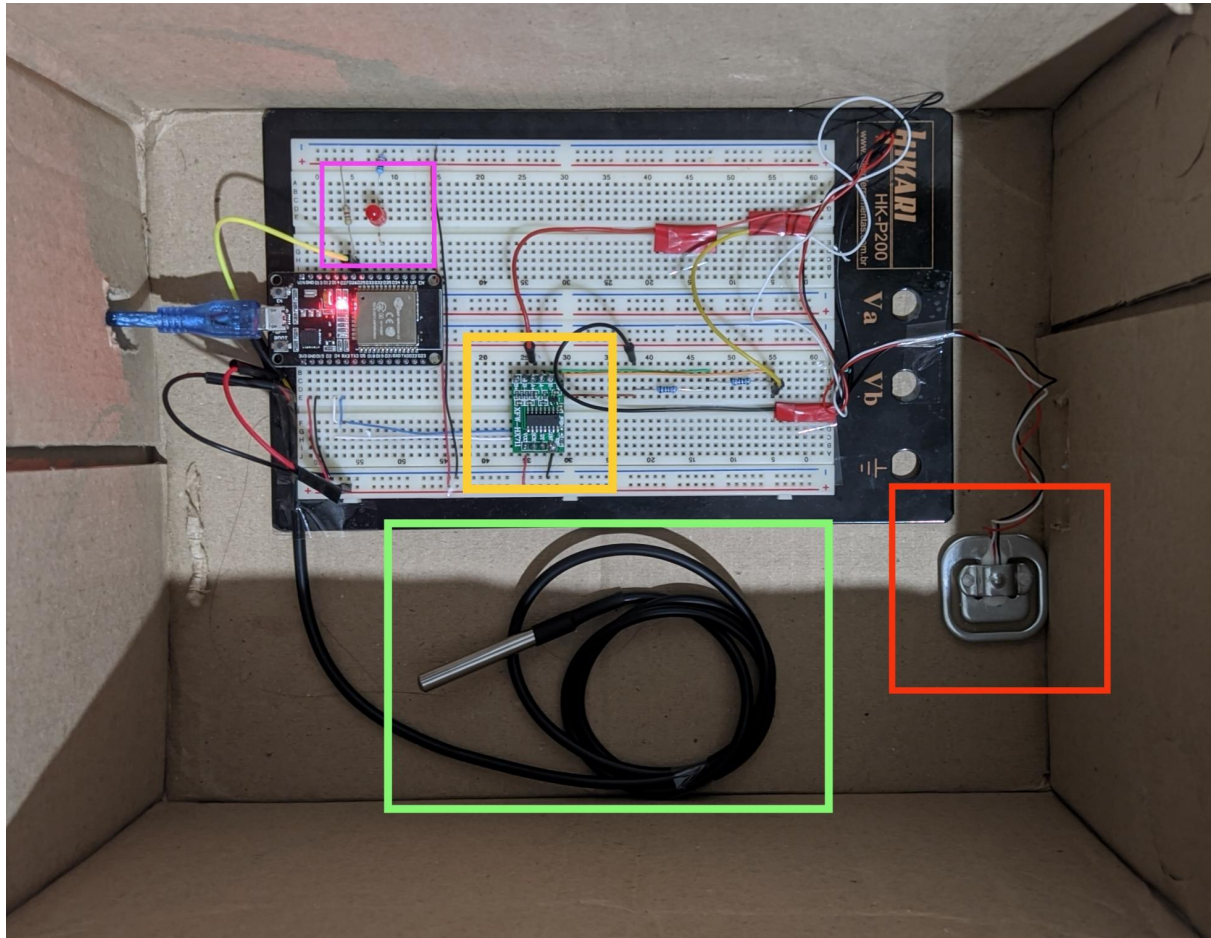


Figura 12: Protótipo Final

A imagem acima apresenta o protótipo final do projeto, onde é possível ver em verde o sensor de temperatura DS18B20, em amarelo o sensor de peso HX711, em vermelho a célula de carga e em roxo o led, que estão posicionados dentro de uma caixa para maior segurança e melhor portabilidade. Também é possível identificar o ESP32, que controla todo o protótipo, recebendo a alimentação suportável de 3.3V que vem da porta USB (5V).

Veremos agora algumas imagens de como o sistema funciona e foi implementado:



Figura 13: Página WEB com o resultado do volume



Figura 14: Página WEB com o resultado da temperatura



Figura 15: Página WEB com o resultado do acionamento do led

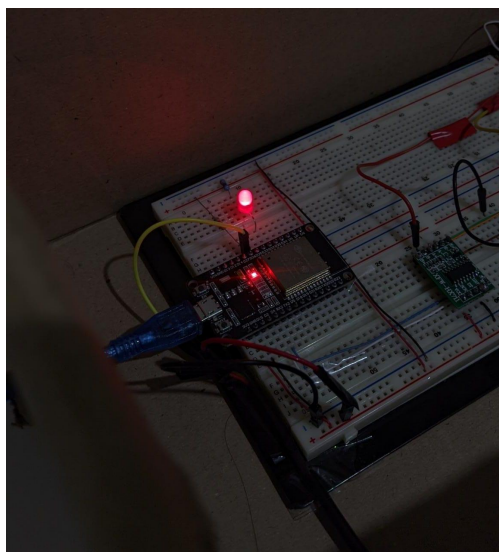


Figura 16: Led aceso após seu acionamento (Figura 15)



Figura 17: Página WEB para o envio do status ao Telegram

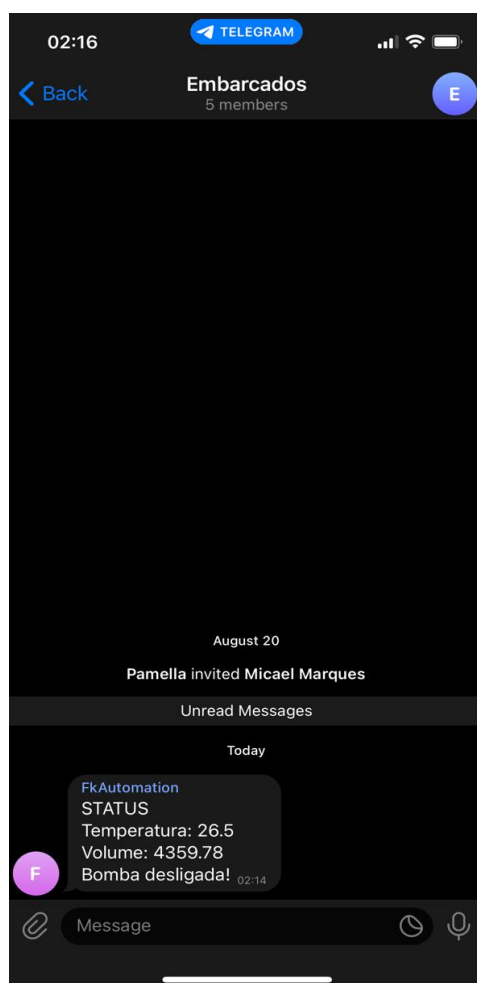


Figura 18: Status no Telegram

Também foi produzido dois vídeos apresentando todo o funcionamento do projeto:

- <https://youtu.be/aw8LmGfO70s>
- <https://youtu.be/Qtd4zbR8mlc>

Tivemos problemas em relação a calibração da célula de carga, pois após testes utilizando o multímetro, concluímos que o cristal da nossa célula de carga quebrou, e não tivemos tempo hábil para realizar sua calibração, uma vez que foi necessário encomendar de João Pessoa uma nova célula. Além disso, utilizamos o microcontrolador ESP32 WROOM que só recebe uma tensão de 3.3V, não tolerando 5V, com corrente de consumo entre 80mA (típica) e 500mA (máxima), por isso, não conseguimos achar um relé que nosso circuito suportasse. Para conseguir contornar esse problema e simular o acionamento da bomba, foi utilizado um led vermelho, quando ele está ligado simula o acionamento da bomba pelo relé, e quando está desligado, simula que a bomba está desligada.

5. CONCLUSÃO

Nesse projeto conseguimos obter mais conhecimento em relação ao uso da ferramenta VSCode, que nos possibilita um controle maior do código e versionamento, onde podemos instalar diversos plugins, como o ESP-IDF Explorer, que são os plugins utilizados no desenvolvimento desse projeto. Além disso, foi possível entender melhor como funciona o sensor de temperatura e a célula de carga com o amplificador HX711. Com relação aos problemas ocorridos, na véspera da entrega, a célula de carga que estávamos utilizando estava com problema e não retornam nenhum valor, por isso, foi necessário adquirir uma nova célula de carga e rever e testar todos as pinagens e resistência do circuito, e por isso não conseguimos calibrá-la da forma correta. Além disso, utilizamos o microcontrolador ESP32 WROOM que só recebe uma tensão de 3.3V, por isso, não conseguimos achar um relé que nosso circuito suportasse. Para conseguir contornar esse problema e simular o acionamento da bomba, foi utilizado um led vermelho, que simula o acionamento da bomba pelo relé. Outro problema enfrentado foi a falta de informação a respeito do framework ESP-IDF na internet, que nos auxiliasse no desenvolvimento do projeto.

Pensando em trabalhos futuros, temos em vista melhorar a calibração da célula de carga, bem como a troca do microcontrolador por um que tenha uma tensão de 5V para colocarmos um relé no circuito.

6. REFERÊNCIAS

- <https://github.com/danielTobon43/hx711-esp-idf-library>
- <https://www.espressif.com/en/products/modules/esp32>
- <https://github.com/nettigo/DS18B20>
- <https://alfacomp.net/2019/04/02/ds18b20-conheca-o-sensor-de-temperatura-digital-inteligente/>
- <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>
- <https://docs.platformio.org/en/latest/platforms/espressif32.html>