



**INSTITUTO  
FEDERAL**

Paraíba

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
CAMPUS CAMPINA GRANDE  
UNIDADE ACADÊMICA DE INFORMÁTICA  
ENGENHARIA DE COMPUTAÇÃO

SISTEMAS EMBARCADOS

SISTEMA DE HIDROPONIA

ALUNO:

HARRISON WENDEL RODRIGUES SANTOS

CAMPINA GRANDE, 25/08/2022

# Objetivos

## Objetivos gerais

- Monitoramento do funcionamento dos sistemas de resfriamento e temperatura da água (enviar estado do driver, temperatura da água);
- Gera alarmes e enviar pelo Telegram quando estiverem fora do range ideal de funcionamento;
- Controle de acionamento remoto do sistema de resfriamento (driver).

## Requisitos:

- Controle e driver para sistemas de resfriamento da água
- Sensor de temperatura água (DS18B20);
- IHM (Display e key).

## Etapas de desenvolvimento

### Leitura de temperatura

A leitura da temperatura ambiente é realizada através de um sensor DHT11, capaz de ler temperaturas de 0 a 50 °C, com resolução de 1°C. Lida a temperatura do ambiente, foi possível identificar se a temperatura está no intervalo ideal (definido manualmente durante o desenvolvimento dessa etapa) ou não. Quando não está, um led verde se mantém aceso, indicando visualmente o bom estado da temperatura. Caso contrário, um led vermelho pisca em uma frequência de 10Hz, indicando visualmente o mau estado da temperatura.

### Impressão da temperatura em display 16x2 e controle do intervalo ideal

Nesta etapa, foi necessário conectar o Display 16x2 ao módulo I2C, e o módulo I2C ao ESP32. A ligação entre o módulo e o ESP32 é intermediada por um conversor de nível lógico bidirecional, uma vez que o ESP32 tem nível lógico alto em 3.3V e o módulo I2C em 5V. Feita essa ligação, foi utilizada uma biblioteca externa que faz a comunicação com o módulo I2C, restando apenas controlar o cursor do display e os caracteres impressos em cada espaço disponível. Assim, foi possível imprimir em tempo real no display 16x2 as temperaturas lidas e o intervalo de operação ideal.

Após isso, foram instalados 4 botões, a fim de realizar o controle do intervalo de operação ideal em tempo de funcionamento. Um par de botões é responsável por aumentar e diminuir o limite inferior do intervalo, e o outro par por aumentar e diminuir o limite superior do intervalo. Quando não pressionado, o pino de cada botão que está ligado a uma porta do ESP se mantém em nível lógico alto, devido a porta do ESP estar com o PULL\_UP ativo. Quando pressionado, o pino do botão é ligado ao GND do sistema, fazendo o sinal ir para o nível lógico baixo. Uma interrupção ligada a porta do ESP identifica a existência de uma borda de descida e aciona a função que fará o incremento ou decremento do limite associado ao botão.

### Controle ativo da temperatura

Nesta etapa, foi implementado um sistema de controle ativo da temperatura. Caso a temperatura se torne maior que o limite superior, um led azul ficará piscando em 10Hz e um relê conectado ao ESP é acionado, permitindo que um sistema de resfriamento externo possa ser ativado para diminuir a temperatura. Caso a temperatura se torne menor que o limite inferior, um led vermelho ficará piscando em 10Hz e um relê conectado ao ESP é

acionado, permitindo que um sistema de aquecimento externo possa ser ativado para aumentar a temperatura. Trinta segundos após o acionamento do sistema externo de aquecimento ou resfriamento, uma nova comparação é feita verificando se a temperatura voltou ao intervalo ideal de funcionamento. Se sim, o sistema que foi ativado é desativado, assim como seu led indicador. Caso contrário, uma nova comparação é feita após novos 30 segundos. Esse ciclo se repete até que a temperatura retorne ao intervalo ideal de funcionamento.

Também foram adicionados dois botões que permitem o acionamento manual dos sistemas externos de aquecimento ou resfriamento. Observe que, mesmo que acionados manualmente, caso a temperatura esteja dentro do intervalo ideal 30 segundos após o acionamento, eles serão desativados da mesma forma que seriam desativados caso tivessem sido ativados automaticamente.

## Comunicação com o Telegram

Nesta etapa, foi desenvolvida a comunicação do ESP32 com a API do Telegram. Inicialmente, foi necessário criar um bot no aplicativo do Telegram. O nome escolhido para o bot foi “Hidroponia”, e o nome de usuário “hidroponia\_hw\_bot”. Criado o bot, o Telegram disponibilizou o TOKEN que deve ser utilizado nos métodos da API para controlar esse bot. Após isso, foi enviado uma mensagem para o bot através de um Telegram pessoal, e assim foi possível obter o ID da conversa entre o bot e esse Telegram pessoal. Obtidas todas essas informações, foi possível montar uma requisição para qualquer método da API do Telegram.

Mas para que qualquer requisição partisse do ESP, foi necessário conectá-lo a uma rede WiFi com acesso à internet, montar a requisição com todos os campos necessários e enviá-la através de um POST HTTP. A partir de então, a cada ativação ou desativação dos sistemas externos de aquecimento ou resfriamento, o código envia uma requisição para a API do Telegram, que por sua vez a recebe e comanda o bot a enviar na conversa com o Telegram pessoal utilizado a ação tomada pelo ESP e o horário, informação a qual é passada a requisição. Os acionamentos manuais também são comunicados.

Adicionalmente, a cada minuto, o ESP envia para o bot a temperatura lida e o intervalo ideal de funcionamento, independentemente se a temperatura está no intervalo ideal de funcionamento ou não.

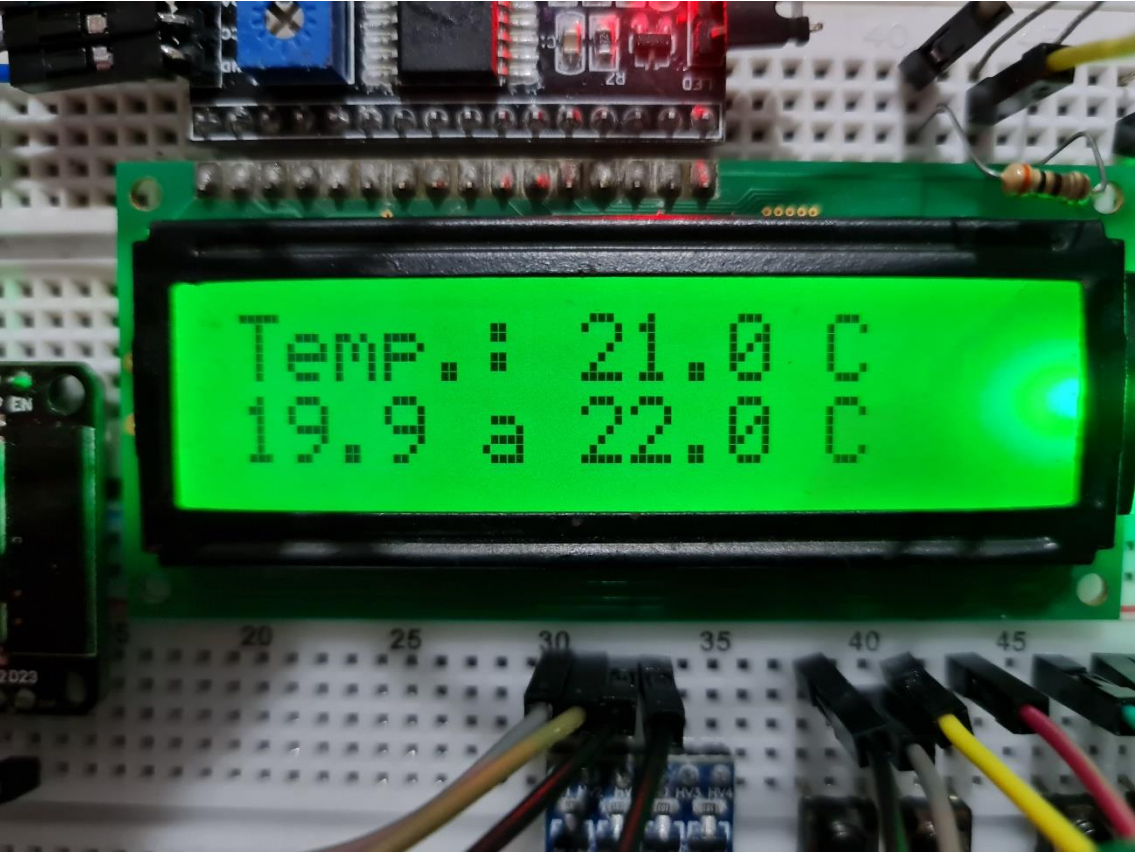
## Material utilizado

- 1 Sensor DTH11
- 4 LEDs 3mm
- 4 Resistores 300 ohms
- 6 Push bottons
- 1 Circuito conversor de nível lógico bidirecional de 4 canais
- 1 ESP32
- 1 Display 16x2
- 1 Módulo I2C para display 16x2
- 1 Módulo com 4 relés
- 2 Protoboards 830 pontos
- Jumpers diversos

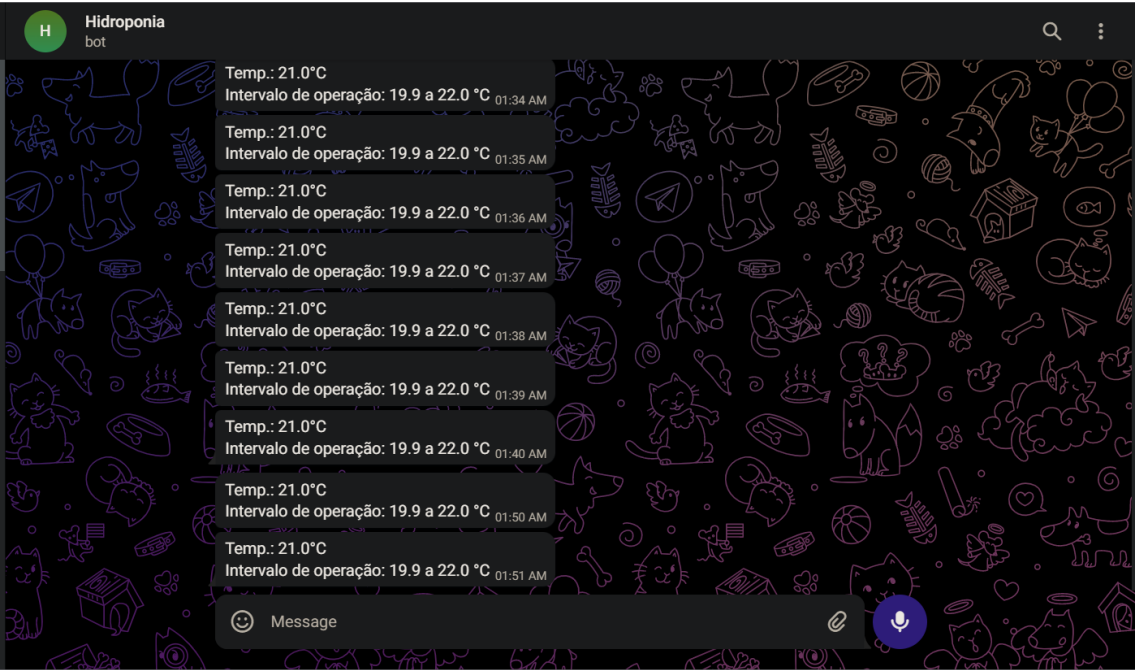
## Bibliotecas externas

- <https://github.com/DavidAntliff/esp32-i2c-lcd1602>
- <https://github.com/DavidAntliff/esp32-smbus>
- <https://github.com/Anacron-mb/esp32-DHT11>

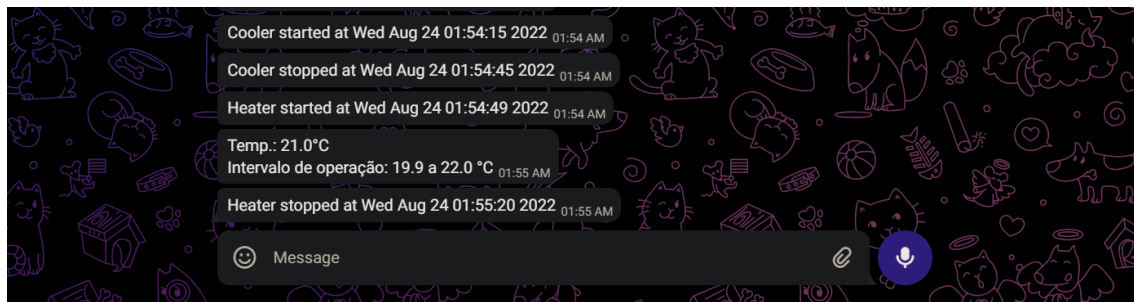
# Resultados



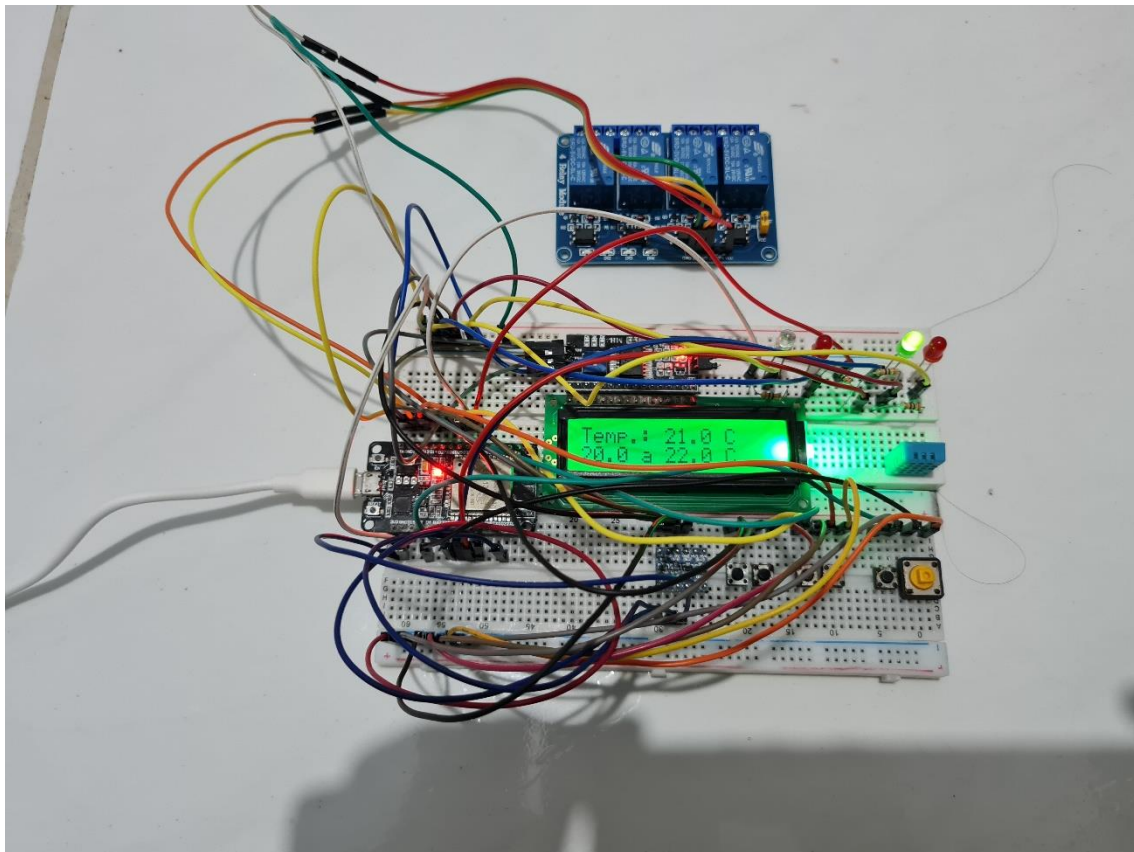
Temperatura lida e intervalo ideal de operação impressos no display 16x2.



Temperatura lida e intervalo ideal de operação a cada minuto pelo Telegram

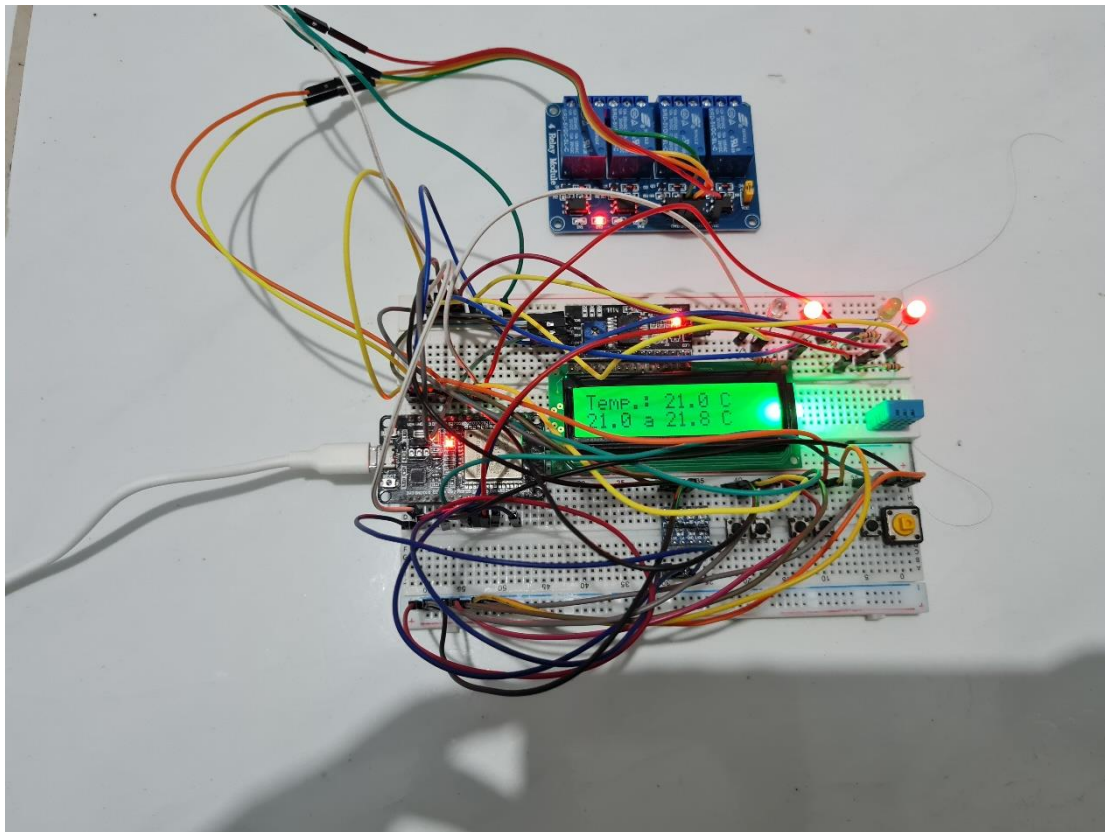


Comunicação de eventos pelo Telegram

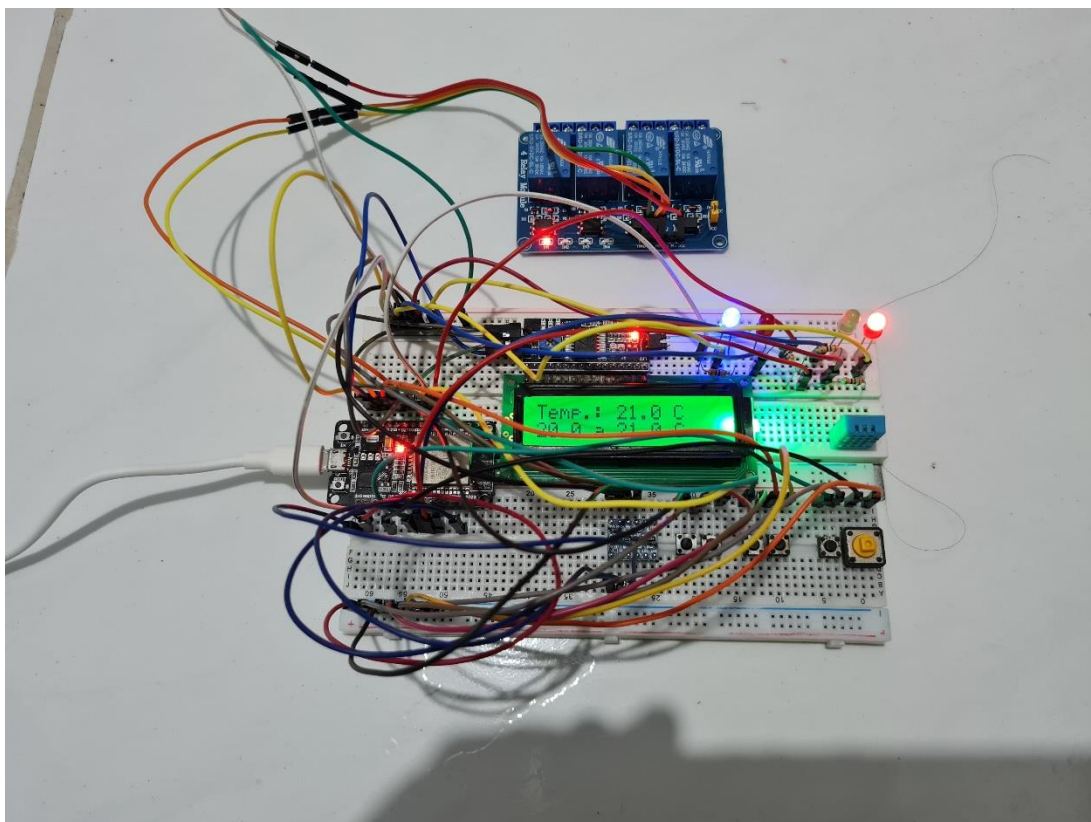


Temperatura dentro do intervalo ideal de funcionamento





Sistema de aquecimento externo ativado



Sistema de resfriamento externo ativado



## Conclusão

Tendo em vista os resultados obtidos, pode-se considerar que o projeto foi um sucesso, uma vez que satisfaz todos os requisitos necessários com robustez e ainda adicionou uma funcionalidade de grande valia, que foi a adição do controle de acionamento de um sistema de aquecimento externo.

Quanto aos conhecimentos adquiridos, pude colocar em prática vários conceitos adquiridos essa disciplina, sendo eles: Tasks, semáforos, debounce, escalonamento e interrupções.

Durante o desenvolvimento desse projeto enfrentei muitas dificuldades, sendo essas as mais relevantes:

- Configuração do ambiente
- Sincronização do relógio do ESP32
- Comunicação do ESP32 com o Telegram como um todo

Para o futuro, deixo como sugestão a implementação das seguintes features:

- Controle do intervalo ideal de funcionamento pelo Telegram
- Controle do tempo mínimo de ativação dos sistemas de aquecimento ou resfriamento
- Botão para encerrar imediatamente a atuação dos sistemas de aquecimento ou resfriamento