



BEATRICE ARAÚJO DA SILVA
MICAHEL SHUMAKER QUEIROZ LOULA DE CARVALHO

Programação de Microcontroladores: Espelho Inteligente

Brasil
Salvador, 2025

BEATRICE ARAÚJO DA SILVA
MICAHEL SHUMAKER QUEIROZ LOULA DE CARVALHO

Programação de Microcontroladores: Espelho Inteligente

Trabalho apresentado a matéria de Programação de Microcontroladores do Curso de Bacharelado em Ciências da Computação, da Universidade Uniruy Wyden, como requisitos avaliativos finais à obtenção da conclusão da matéria.

Centro Universitario Uniruy Wyden
Bacharelado em Ciências da Computação

Orientador: Heleno Cardoso da Silva Filho

Brasil
Salvador, 2025

Resumo

Mudanças climáticas são mais perceptíveis hoje em dia do que a duas décadas atrás. A disseminação de informações climáticas de forma acessível e mais visível pode ajudar na prevenção de perdas e ajudar em tomadas de decisões em caso de situações climáticas desfavoráveis. Esse projeto consiste no desenvolvimento experimental de um espelho inteligente que monitora a situação climática e ilumina o ambiente de forma econômica para informar e alertar de forma mais presente sobre o clima no período de uma semana.

Palavras-chaves: Espelho inteligente, espelho, microcontroladores, monitoramento climático, clima, tecnologia climática, ESP32.

Resumo

El cambio climático es más notorio hoy que hace das décadas. Difundir información climática de forma accesible y más visible puede ayudar a prevenir pérdidas y facilitar decisiones en caso de condiciones climáticas adversas. Este proyecto consiste en desarrollo experimental un espejo inteligente que monitoriza la situación climática y ilumine el entorno de forma económica para informa y alertar de forma que sea mas presente sobre el clima en uno período de uma semana.

Palabras clave: Espejo inteligente, espejo, microcontroladores, monitoreo del clima, clima, tecnología climática, ESP32.

Lista de ilustrações

Figura 1 – Circuito completo	10
Figura 2 – Definições de pinos do Display LCD	11
Figura 3 – Circuito Photo Photoresistor (LDR) e EPS32	12
Figura 4 – Circuito Neopixel e ESP32	13
Figura 5 – Circuito RTC DS1307 e EPS32	14
Figura 6 – Circuito Display LCD e EPS32	15
Figura 7 – Circuito LED e EPS32	16

Lista de símbolos

Ω	Ohmos
----------	-------

Sumário

	Introdução	7
I	PROJETO X MICROCONTROLADORES	8
1	ESPELHO INTELIGENTE, MICROCONTROLADORES E SEU CONCEITO	9
1.1	Componentes	9
1.2	Circuitos	10
1.3	Simulação	16
2	OBJETIVO	18
3	JUSTIFICATIVA	19
4	METODOLOGIA	20
	Conclusão	21
	REFERÊNCIAS	22

Introdução

Nos tempos atuais as mudanças climáticas estão ficando mais perceptíveis a cada dia, de 2020 para cá, diversas regiões no Brasil e no mundo sofreram com alagamentos e mudanças climáticas. Em setembro de 2024 Portugal passou por um dos maiores incêndio já registrado, os pesquisadores informaram um aumento nas ondas de calor da região, resultado dessas mudanças, assim como em maio de 2024, o Rio Grande do Sul, no Brasil, passou por uma das maiores enchentes já registradas causados pelas fortes chuvas.

Em notícias podemos acompanhar as variações climáticas que podem ocorrer durante o dia ou a semana, todavia, por conta da correria do dia a dia, são informações que podem passar despercebidas. A apresentação de forma mais direta e acessíveis dessas informações, podem ajudar em tomadas de decisões em situações não favoráveis climaticamente.

Parte I

Projeto x Microcontroladores

1 Espelho inteligente, microncontroladores e seu conceito

Microcontrolador consiste em um dispositivo que possui memória e poder de processamento para gerenciar pequenos componentes ligados a ele. Responsável por executar instruções programadas para realizar tarefas de forma simples com baixo custo, solucionados problemas do dia a dia, seja de segurança ou tarefas rotineiras de monitoramento para obtenção de dados variados.

A ideia de se criar um espelho inteligente integrado a um microcontrolado, foi pensando com base na facilidade de informar diretamente sobre situações climática de acordo com a localização. Mesmo com o avanço da tecnologia, smartphones, tablets, notebooks e computadores, além da Internet das Coisas (Iot), que liga uma vasta gama de dados armazenados na nuvem, conectados a sistemas e plataformas, ainda assim, informações como essas acaba passando despercebidas e as vezes não perceptíveis durante o dia a dia.

A união dessas informações a um objeto físico presente no dia a dia da população, com a capacidade de processamento e monitoramento climático, passa a ser relevante quando associamos a um objeto sempre visível, alertando e prevenindo contra possíveis situações climáticas durante a semana, principalmente ao nível de chuva.

1.1 Componentes

Para criação do projeto foram incluídos os seguintes componentes:

- Módulo principal, responsável por processar e se conectar com cada componente: Placa ESP32;

- Sensor Photoresistor;
- Módulo RTC DS1307;
- Led simples;
- Resistor de $1k\Omega$ (*quilohms*);
- NeoPixel;
- Display ILI9341 Touch;

A placa ESP32 foi selecionada como microcontrolador devido a versatilidade, capacidade de processamento e conectividade. Ela realiza o processamento necessário e

possui facilidade em se comunicar.

O photoresistor foi implementado para medir a intensidade da luz no ambiente, assim alterando a luminosidade dos neoPixels.

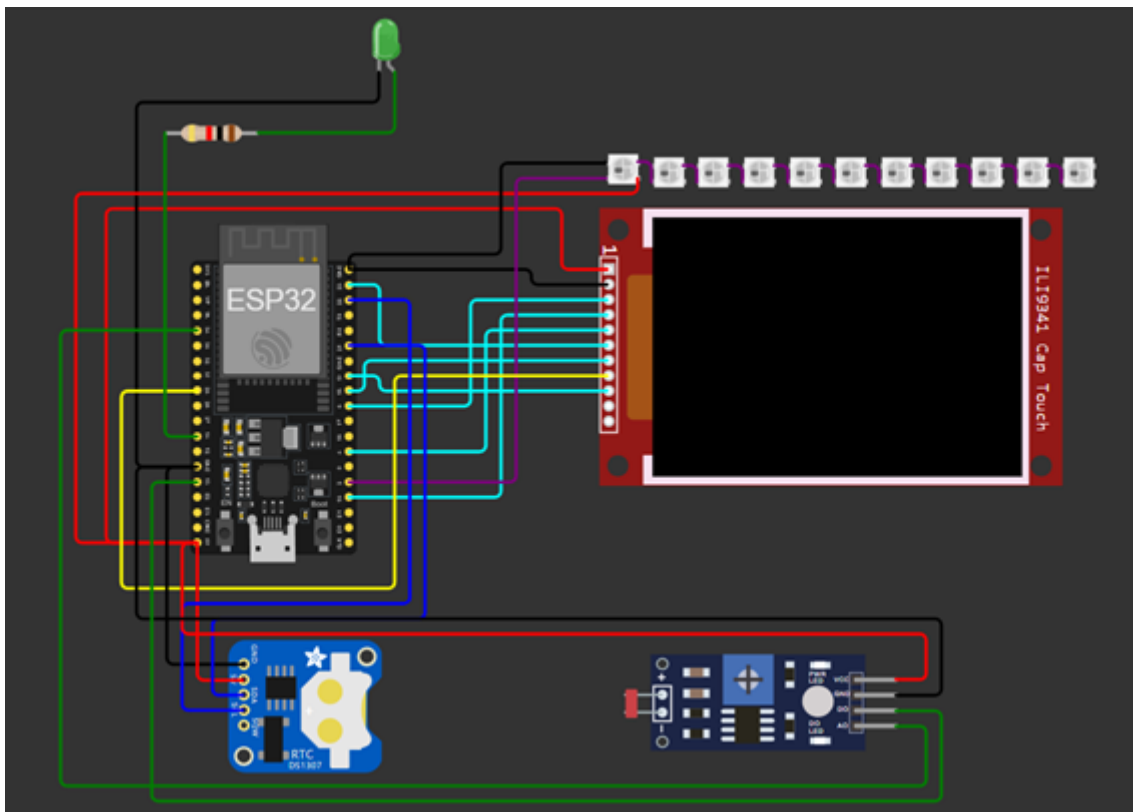
O display exibi as informações climáticas adquiridas através das APIs e dos dados processados pelo módulo RTC DS1307, no qual proporciona a sincronização de data e hora de forma mais exata.

O Led foi utilizado para informar se o sistema está operante ou não, e o resistor para controlar a carga direcionada ao led.

1.2 Circuitos

O circuito possui um conjunto de funcionamento dividido em três partes ligadas ao modulo principal ESP32, que recebe os componentes de entrada, processa e envia sinais aos componentes de saída.

Figura 1 – Circuito completo



Fonte: Autoria própria

Todas as inclusões de bibliotecas externas e internas são definidas no início do programa, assim como os pinos que estão conectados a cada componentes através dos jumpers, cabos que ligam os pinos dos componentes aos pinos do ESP32.

Figura 2 – Definições de pinos do Display LCD

```
/* ***** Definições de Pinos ***** */  
// Definição dos pinos do display  
#define TFT_CS    5  
#define TFT_DC    4  
#define TFT_RST   15  
#define TFT_LED   25 //D8 controla a luz de fundo da tela  
#define LED_ONOFF 14  
#define RTC_SDA   21  
#define RTC_SCL   22
```

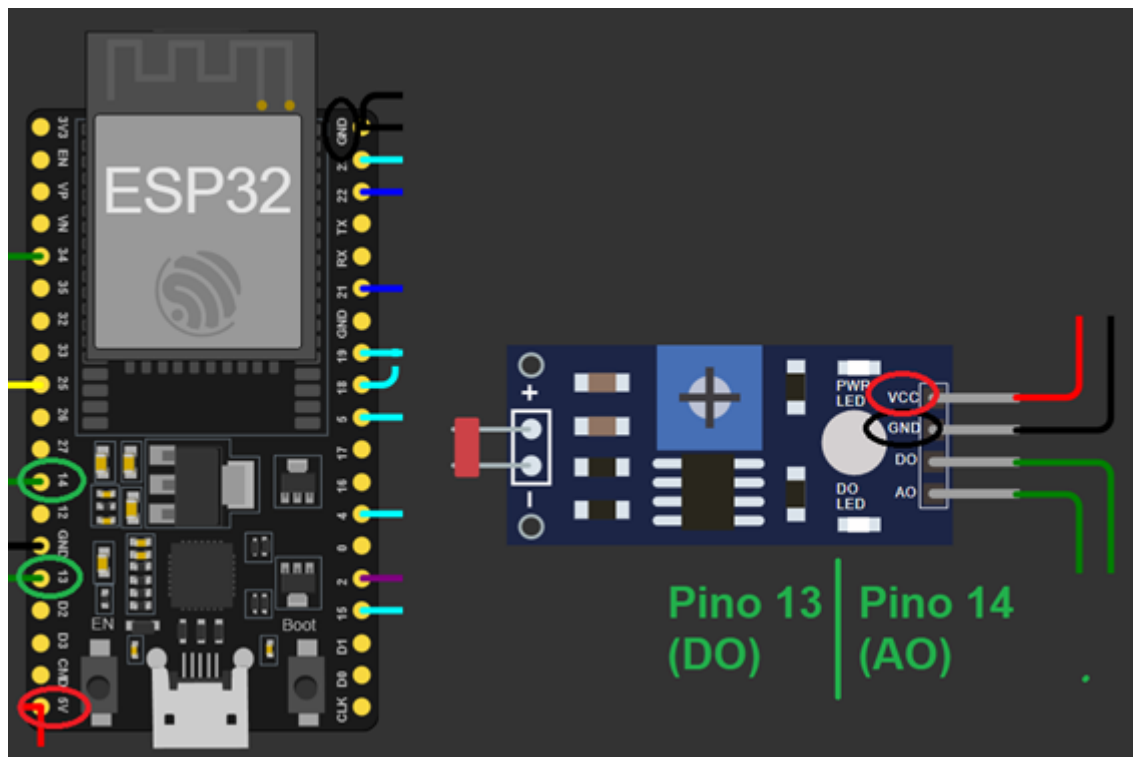
Fonte: Autoria própria

Os componentes definidos como entradas são aqueles que recebem informações e passam ao módulo principal para processá-las e enviar sinais, de acordo com os dados recebidos, aos componentes de saída.

O sistema possui dois componentes de entrada, o primeiro é o Photoresistor (LDR), sensor que monitora a luminosidade ambiente, ele possui quatro pinos, o VCC, fio vermelho, GND, fio preto, DO e AO, ambos fios verdes. Seus pinos DO e AO, representam respectivamente a saída digital e saída analógica, conectados por um jumper verde aos pinos 13 e 14 do ESP32. Sua funcionalidade permite que adquira os dados necessários para serem passados ao ESP32 e de acordo com esses dados, diminua ou aumente a intensidade dos neoPixel.

GND, fio terra, é a potência elétrica, ponto da tensão no circuito responsável pelo retorno para fechamento do circuito, sempre estará conectado a outro pino GND, normalmente ligado por um cabo jumper preto, representado pelo símbolo de negativo nas protoboards, já o VCC é responsável pela alimentação de energia em voltagem, tensão positiva, ligada sempre a um pino com nomenclatura de 5V, normalmente por um jumper de cor vermelha, sendo representado pelo símbolo positivo nas protoboards.

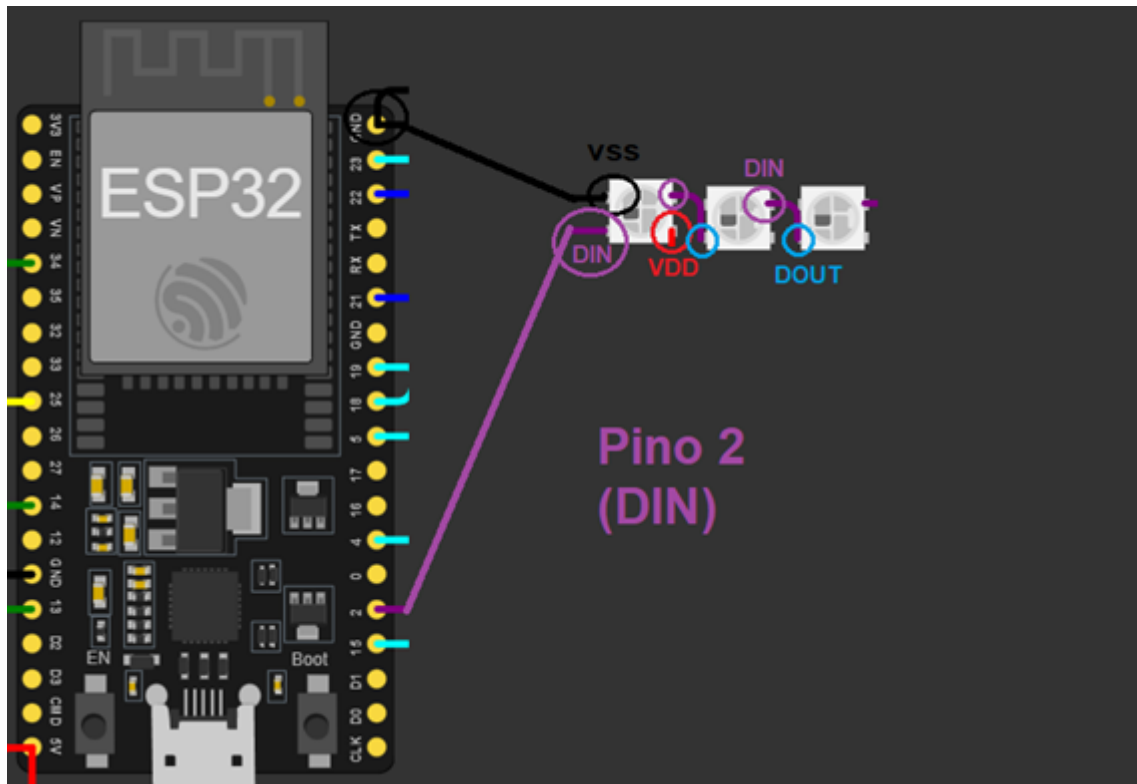
Figura 3 – Circuito Photo Photoresistor (LDR) e EPS32



Fonte: Autoria própria

Os neoPixels possuem quatro pinos, o VSS, fio terra conectado ao GND do ESP32, o VDD, fonte de alimentação ligado ao 5V do ESP32, o DIN, no qual recebe os dados que o ESP32 processa, conectado por um jumper roxo ao pino 2 do EPS32, e o DOUT, apenas utilizado quando se conectado outros neoPixels a ele, pois o dado passado para o primeiro neoPixel através do pino DIN passa para o próximo através do pino DOUT de um conectado ao pino DIN do outro.

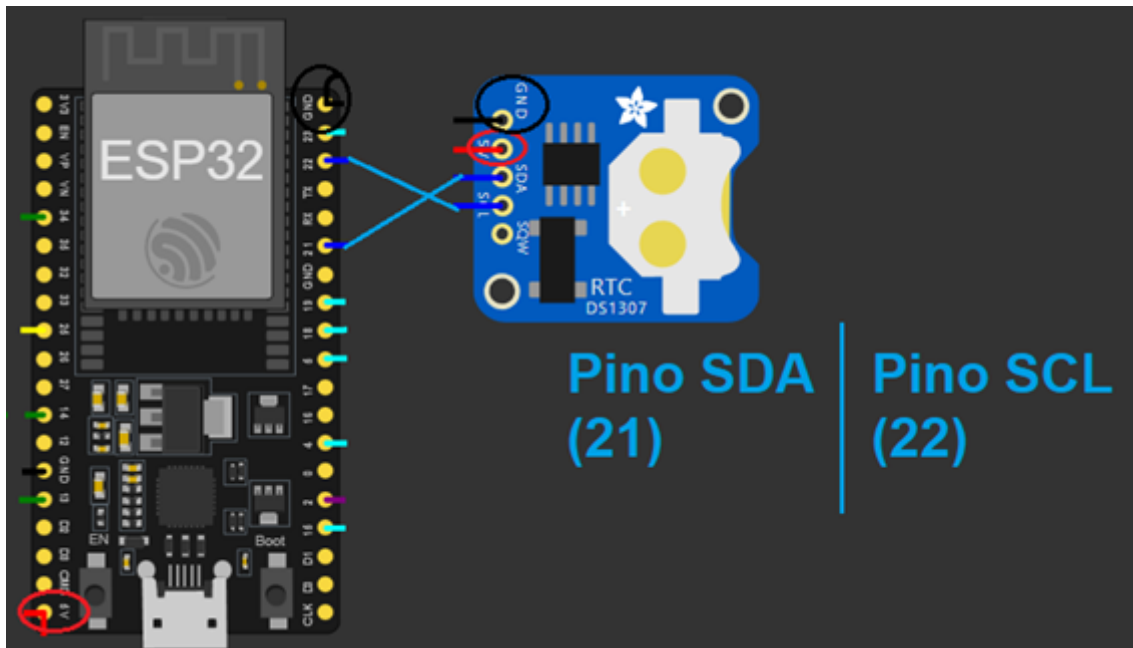
Figura 4 – Circuito Neopixel e ESP32



Fonte: Autoria própria

A segunda parte e componente de entrada é o módulo RTC DS1307, conhecido como Real Time Clock, possui cinco pinos, os padrões GND e 5V, além dos pinos SDA, linha de dados I2C, SCL, linha de clock I2C e SQW, saída de onda quadrada. Nesse circuito foram utilizados apenas os pinos SDA, que responsável por transmitir os dados e receber ao conectar com o protocolo I2C, responsável por pegar horário e data precisa no sistema, e o SCL, que sincroniza os sinais que transferem os dados entre o módulo EPS32 e o RTC DS1307. Os pinos DAS e SCL foram ligados aos pinos 21 e 22 respectivamente do EPS32.

Figura 5 – Circuito RTC DS1307 e EPS32



Fonte: Autoria própria

A parte principal do projeto e última, consiste em um dispositivo de saída, responsável por renderizar todos os dados processados pelo EPS32 graficamente. O display possui onze pinos disponíveis para conexão, nesse projeto foram utilizados apenas nove dos onze, não sendo necessário a utilização do pino SCL, mesmo pino do módulo RTC DS1307, responsável pela comunicação com o protocolo I2C, e SDA, que transmite os dados de toque da tela ao EPS32.

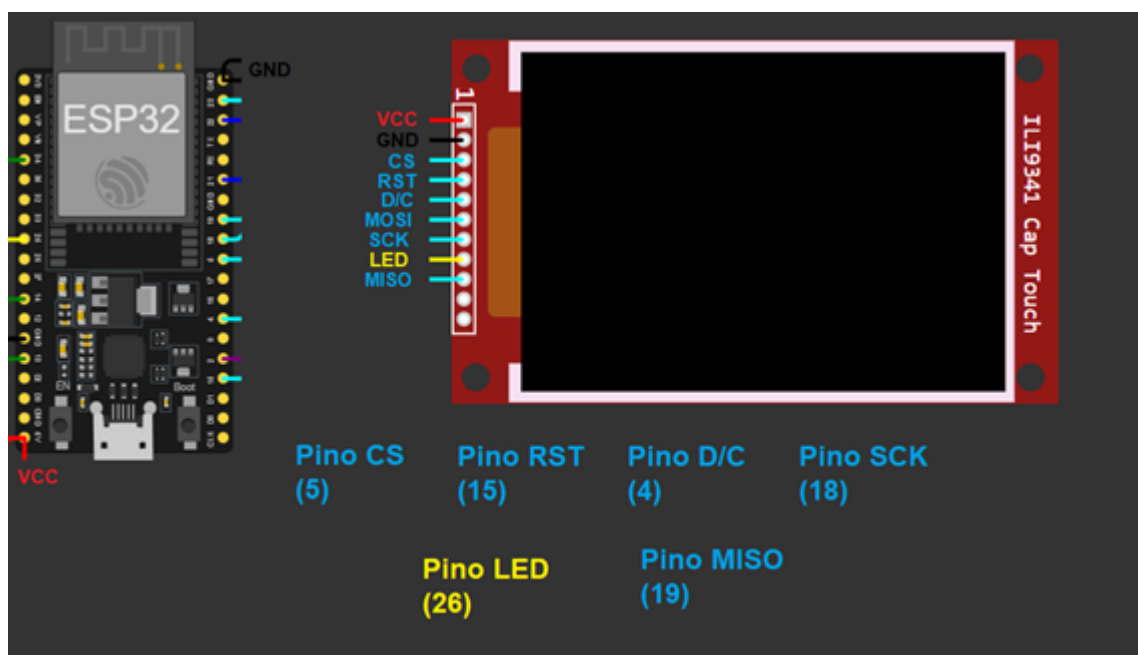
Os pinos utilizados foram os de energia e terra, CVV e GND, além dos pinos:

- CS: Cria a comunicação entre o display e o módulo EPS32, conectado ao pino 5 do ESP32;
- RST: Responsável por reiniciar quando ocorre algum problema de comunicação, ele reseta o display, recarregando os dados novamente e renderiza sua interface, conectado ao pino 15 do ESP32;
- D/C: Controla os tipos de dados passados ao display, se é gráfico ou instruções, conectado ao pino 4 do ESP32;
- MOSI: Porta de comunicação que recebe os dados do módulo EPS32 para o display, conectado ao pino 19 do ESP32;
- SCK: Determina quando os dados devem ser enviados e recebidos pelo display através do protocolo SPI, sincronizando a frequência em que recebe e envia dados, conectado ao pino 18 do ESP32;

- LED: Pino controla a iluminação da tela, sua intensidade, além de controlar quando a tela do display deve ficar ligada ou não, como por exemplo modo descanso de tela, conectado ao pino 26 do ESP32;
- MISO: Responsável por renderizar funcionalidades que captar dados do touch do display, processar esses dados e passar para o ESP32, conectado ao pino 19 do ESP32.

Os pinos 5, 15, 19 do EPS32, são considerados como pino de strapping, que renderiza e realiza todo o processo na memória flash, controlando o modo de inicialização, são conhecidos também como GPIO5, GPIO15 e GPIO0.

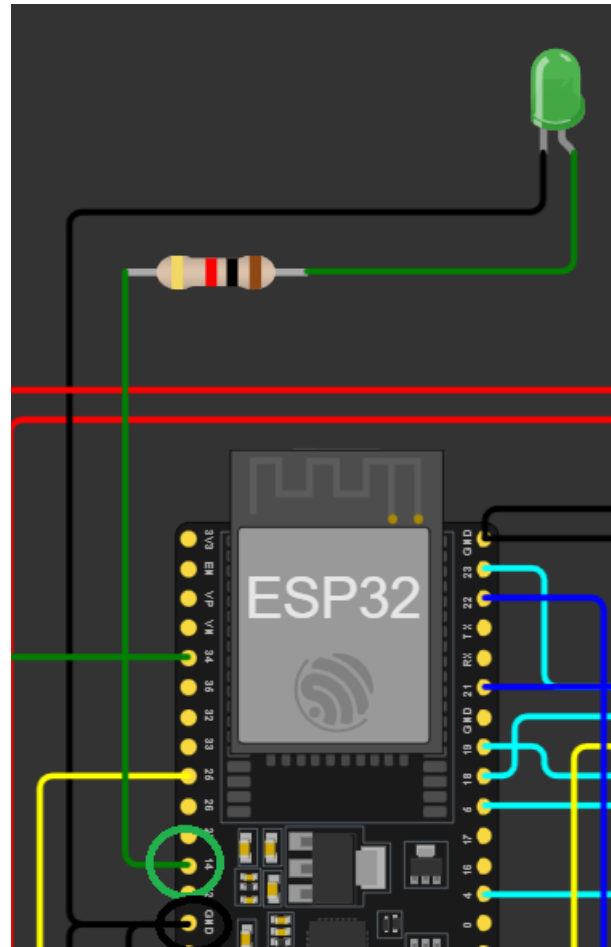
Figura 6 – Circuito Display LCD e EPS32



Fonte: Autoria própria

Uma led foi incluída no projeto para identificar se o EPS32 foi ligado, apenas como uma sinal, indicando que o modulo principal esta operante. Suas conexões consistem na porta GND do EPS32, e ao resistor no caminho da conexão ao pino 14 do modulo principal.

Figura 7 – Circuito LED e EPS32



Fonte: Autoria própria

1.3 Simulação

A simulação consiste na integração dos componentes, que é realizada por meio da linguagem de programação C++. O ambiente de simulação foi montado com as seguintes ferramentas:

- Visual Studio Code como IDE principal;
- Wokwi, operando dentro do Visual Studio Code, mostrando a simulação em tempo real, sendo executado junto com o Platform.io;
- EEZ Studio, para montar a parte de UI gráfica a ser renderizada pelo display;
- Platform.io, extensão do Visual Studio Code responsável por baixar e integrar as bibliotecas necessárias a serem utilizadas no projeto.

O processamento das informações é feito em tempo real, dividido em três processos, todas as partes são conectadas e inicializadas através do microcontrolador ESP32, ele é o

responsável por:

- Processar as bibliotecas e inicializar os componentes conectados a ele;
- Recebe os dados coletados do sensor, processa-os e envia as diretrizes de acordo com os dados coletados as neoPixel;
- Inicializar o modulo RTC DS1307 e processar, além de passar os dados disponibilizados por ele ao display;
- Requisitar os dados das APIs para processar e passar para o display;
- Inicializar funcionalidades gráficas e o display;
- Informar se o sistema está funcionando ou não através da Led.

O primeiro processo consiste na identificação da luz ambiente pelo sensor. Cada vez que há alteração na luz ambiente, o sensor photoresistor identifica, lendo a variação de luminosidade. Essa variação é processada pelo modulo principal, ESP32, que informa aos neoPixel qual o ajustando de luminosidade deve ser feito.

O segundo processo é através do módulo RTC DS1307, ter a horário e da data atual de forma precisa, sempre atualizando o display com essas informações. Ao ser inicializado pelo EPS32, o módulo obtém do sistema a hora e data atual, retorna essa informação que é formatada pelo EPS32 e renderizada no display LCD. Devido a ser uma informação com constantes alterações, sua verificação é feita em loop, ou seja, por milissegundos.

O terceiro processo, consiste no objetivo principal do projeto, disponibilizar informações temporais através do display LCD. Quando o módulo EPS32 inicia, ele inicializa as bibliotecas gráficas para ser renderizado a interface no display. A principal biblioteca a ser renderizada é a tft e lvgl, essas bibliotecas permitem uma gama de configurações na utilização de imagens, criação de formas e utilização de cores em bits de acordo com o tamanho do display. Isso permite que ao ser renderizado não danifique os componentes e sobrecarregue a memoria RAM e Flash do ESP32.

Após todas as inicializações, o foco é obter os dados climáticos necessários, para serem renderizados dinamicamente a cada determinado tempo. Esses dados são obtidos através das APIs Wheater e Open Meteor, requisitadas via HTTP pela conexão com o wifi, funcionalidade presente no EPS32.

A interface foi projetada para o display LCD de dimensão 320x240, a fim de apresentar as informações de forma clara e de fácil entendimento, oferecendo uma experiência interativa e prática. Após a montagem, testes foram realizados para validar as funcionalidades, incluindo a calibração do photoresistor e a verificação do processamento e exibição dos dados no display.

2 Objetivo

Esse projeto tem como objetivo demonstrar que nas atividades diárias mais simples a tecnologia pode-se fazer presente, trazendo a percepção humano informações que podem ser vistas em jornais, seja televisão ou impresso, ao acessar os dispositivos moveis ou ao escutar radio, seja no carro, celular, entre outros lugares. Além de trazer consigo pequenas funcionalidades que podem transmitir conforto.

3 Justificativa

A utilização de microcontroladores para criação de pequenos sistemas tecnológicos acaba sendo viável devido aos seus componentes serem de baixo custo e não necessitar de grande poder de processamento, sendo assim, situações que requer monitoramento de segurança e que necessitam de facilidade em processos e/ou situações diárias, acabam sendo bons focos para implementação desses sistemas.

O projeto do espelho inteligente permite o monitoramento de forma prática durante a semana referente ao tempo, clima, temperatura, umidade e velocidade do vento, presente em um espelho, facilitando a visualização das informações de forma direta, em determinados momentos do dia, além de conter luminosidade que varia de intensidade de acordo com a luz ambiente, reduzindo o consumo e trazendo usabilidade ao espelho de forma mais tecnológica e informativa.

4 Metodologia

Esse projeto segue a metodologia de desenvolvimento experimental, cujo o foco está na criação de um sistema para exibição da situação climática em um espelho. A necessidade de um sistema integrado e eficiente se fez como base do projeto utilizando os demais componentes como parte do processo de simulação.

- A placa ESP32 que foi selecionada por ser versátil e também a sua capacidade de processamento.
- O Photoresistor foi utilizado para medir a incidência da luz no ambiente, o display LCD foi necessário para exibir as informações/atualizações do clima.
- O módulo RTC DS1307 é o responsável por fornecer tanto a data quanto a hora atualizados.
- Utilizando para a integração a linguagem C++ e as ferramentas Visual Studio Code, Platform.io e Wokwi.

Conclusão

Com o aumento de dados, rede e a aceleração da tecnologia, incorpora microcontroladores no dia a dia em objetos físico para facilitar tarefas e processos diários fica cada vez mais fácil. A utilização de componentes como sensores e atuadores conectados à rede de dados Iot (Internet das Coisas), permitem que esses sistemas possam ser acessados em qualquer lugar de forma física ou virtual, conectando sistemas que utilizam sensores físicos, permitindo o controle dos mesmos.

Referências

ELETROGATE, B. *Introdução aos Microcontroladores PIC – Parte 1*. [S.l.]. Disponível em: <<https://blog.eletrogate.com/introducao-aos-microcontroladores-pic-parte-1/>>. Acesso em: 19 de abril de 2025. Nenhuma citação no texto.

ENGINNERS, L. M. *Getting Date Time From NTP Server With ESP32*. [S.l.]. Disponível em: <<https://lastminuteengineers.com/esp32-ntp-server-date-time-tutorial/>>. Acesso em: 24 de maio de 2025. Nenhuma citação no texto.

EXAME. *Incêndios florestais em Portugal causaram sete mortes e destruíram 10 mil hectares*. [S.l.]. Disponível em: <<https://exame.com/mundo/incendios-florestais-em-portugal-causaram-sete-mortes-e-destruiram-10-mil-hectares/>>. Acesso em: 30 de maio de 2025. Nenhuma citação no texto.

GERAQUE, E. *Estudo mapeia as causas e as circunstâncias que ocasionaram em 2024 a maior tragédia ambiental do RS*. [S.l.]. Disponível em: <<https://exame.com/mundo/incendios-florestais-em-portugal-causaram-sete-mortes-e-destruiram-10-mil-hectares/>>. Acesso em: 30 de maio de 2025. Nenhuma citação no texto.

MENDES, C. *Pesquisadores apontam janeiro de 2025 como o mais quente da história*. [S.l.]. Disponível em: <<https://www12.senado.leg.br/radio/1/noticia/2025/02/11/pesquisadores-apontam-janeiro-de-2025-como-o-mais-quente-da-historia>>. Acesso em: 30 de maio de 2025. Nenhuma citação no texto.

METEO, O. *Weather Forecast API*. [S.l.]. Disponível em: <<https://open-meteo.com/>>. Acesso em: 14 de abril de 2025. Nenhuma citação no texto.

ORACLE. *O que é IoT?* [S.l.]. Disponível em: <<https://www.oracle.com/br/internet-of-things/>>. Acesso em: 24 de maio de 2025. Nenhuma citação no texto.

SUHANKO, D. *Display ILI9341 com touch e SD - configurar o touch*. [S.l.]. Disponível em: <<https://www.manualdomaker.com/article/display-ili-9341-com-touch-e-sd-configurar-o-touch/>>. Acesso em: 19 de maio de 2025. Nenhuma citação no texto.

WEATHER. *Weather API*. [S.l.]. Disponível em: <<https://www.weatherapi.com/weather/>>. Acesso em: 06 de maio de 2025. Nenhuma citação no texto.

WEATHER, O. *Open Weather API*. [S.l.]. Disponível em: <<https://openweathermap.org/city/2643743>>. Acesso em: 06 de maio de 2025. Nenhuma citação no texto.

WOKWI. *Referência do wokwi-ds1307*. [S.l.]. Disponível em: <<https://docs.wokwi.com/pt-BR/parts/wokwi-ds1307>>. Acesso em: 15 de maio de 2025. Nenhuma citação no texto.

WOKWI. *Referência do wokwi-photoresistor-sensor*. [S.l.]. Disponível em: <<https://docs.wokwi.com/pt-BR/parts/wokwi-photoresistor-sensor>>. Acesso em: 15 de maio de 2025. Nenhuma citação no texto.

WOKWI. *Simulador ESP32*. [S.l.]. Disponível em: <<https://docs.wokwi.com/pt-BR/guides/esp32>>. Acesso em: 15 de maio de 2025. Nenhuma citação no texto.