

## Documentação do Hardware: Dispositivo IoT Rastreador (Derivador)

### 1. Visão Geral e Contexto do Projeto

Este documento detalha a arquitetura, montagem e funcionamento do dispositivo de hardware (IoT) utilizado no projeto **floatData**. Este componente, referido como "derivador", é a ponta física do sistema, responsável por coletar dados de geolocalização em tempo real e transmiti-los para a plataforma de software floatData para processamento e visualização.

O objetivo deste hardware é ser um rastreador autônomo e eficiente, servindo como a principal fonte de dados para o aplicativo de monitoramento.

### 2. Arquitetura do Hardware

#### 2.1. Componentes Principais

1. **Microcontrolador:** ESP32
2. **Sensor de Localização:** Módulo GPS NEO-6M
3. **Interface de Status:** Display LCD I2C 16x2
4. **Módulo de Conectividade:** Wi-Fi (integrado ao ESP32)

#### 2.2. Justificativa dos Componentes

- **ESP32:** Escolhido por seu excelente custo-benefício, baixo consumo de energia em modo *deep sleep*, processamento dual-core e, principalmente, por sua **conectividade Wi-Fi nativa**. Isso elimina a necessidade de módulos de comunicação externos, simplificando o design e reduzindo custos.
- **Módulo GPS NEO-6M:** É um sensor GPS confiável, de fácil integração via comunicação serial e amplamente utilizado no mercado, com vasta documentação e suporte da comunidade.
- **Display LCD I2C:** Atua como uma interface de diagnóstico e status. Sua função **não é para o usuário final**, mas sim para o desenvolvedor durante a montagem e testes, permitindo verificar rapidamente o status da conexão Wi-Fi, a aquisição de sinal GPS e o envio de dados, sem a necessidade de um monitor serial conectado.
- **Conectividade Wi-Fi:** O módulo Wi-Fi integrado ao ESP32 é o componente chave para a comunicação com a internet, permitindo que o dispositivo envie os dados coletados para a API do backend do floatData.

### 3. Montagem e Conexões

A montagem foi projetada para ser simples e robusta, conectando os periféricos ao ESP32.

Componente	Pino do Componente	Pino do ESP32	Descrição da Conexão
LCD I2C 16x2	VCC	VIN (5V)	Alimentação do display
	GND	GND	Terra
	SDA	GPIO21	Pino de dados para comunicação I2C
	SCL	GPIO22	Pino de clock para comunicação I2C
Módulo GPS	VCC	3V3	Alimentação do módulo GPS
	GND	GND	Terra
	TX	GPIO16 (RX2)	Transmissão do GPS para a Serial 2 do ESP32
	RX	GPIO17 (TX2)	Recepção na Serial 2 (uso opcional)

### 4. Firmware

O firmware embarcado no ESP32 é o cérebro do dispositivo. O arquivo sketch-abp.ino correspondente contém toda a lógica de programação.

#### 4.1. Lógica de Operação

O fluxo de operação do dispositivo segue os seguintes passos:

- Inicialização:** O ESP32 inicializa seus componentes, incluindo a comunicação serial com o GPS e a interface I2C com o LCD.
- Conexão Wi-Fi:** O dispositivo tenta se conectar à rede Wi-Fi pré-configurada. O status ("Conectando...", "Conectado!") é exibido no LCD.
- Aquisição de Sinal GPS:** Em paralelo, o dispositivo aguarda o módulo GPS obter um "fix", ou seja, uma trava de sinal com satélites suficientes para determinar a localização. O LCD exibe "Aguardando GPS...".
- Coleta e Formatação:** Uma vez que o sinal GPS é obtido, o firmware coleta os dados de latitude e longitude.

5. **Transmissão de Dados:** Os dados coletados são formatados (geralmente em um payload JSON) e enviados via uma requisição **HTTP POST** para um endpoint específico da API do floatData.
6. **Feedback e Loop:** O LCD exibe as coordenadas enviadas como feedback visual. Após a transmissão, o dispositivo aguarda um intervalo de tempo definido (delay) antes de repetir o ciclo, para otimizar o consumo de bateria e o volume de dados.

#### 4.2. Bibliotecas Essenciais

- **TinyGPS++:** Biblioteca crucial para decodificar as sentenças NMEA enviadas pelo módulo GPS e extrair de forma simples os dados de latitude, longitude, velocidade e altitude.
- **LiquidCrystal\_I2C:** Facilita o controle do display LCD utilizando apenas dois pinos (SDA e SCL), simplificando as conexões.
- **WiFi.h / HTTPClient.h:** Bibliotecas nativas do ESP32 para gerenciar a conexão Wi-Fi e realizar as requisições HTTP para a API.

#### 5. Configuração do Ambiente de Desenvolvimento

Para compilar e enviar o firmware para o dispositivo, o ambiente de desenvolvimento precisa ser configurado na IDE Arduino.

##### 1. Instalação da Placa ESP32:

- Em Arquivo > Preferências, adicione a URL [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) nas "URLs Adicionais de Gerenciadores de Placas".
- Em Ferramentas > Placa > Gerenciador de Placas, pesquise por ESP32 e instale o pacote da Espressif Systems.

##### 2. Seleção da Placa e Porta:

- Em Ferramentas > Placa, escolha o modelo correto do seu ESP32.
- Em Ferramentas > Porta, selecione a porta COM à qual o dispositivo está conectado.

#### 6. Testes e Validação

O sucesso da validação do hardware não se resume a exibir coordenadas no LCD. O teste principal consiste em:

1. Ligar o dispositivo.
2. Observar o fluxo de status no LCD (Conexão Wi-Fi, Aquisição de GPS).

3. Após a indicação de envio de dados, **verificar na plataforma floatData (aplicativo web/mobile) se um novo ponto de localização foi registrado** no mapa para o dispositivo em questão.

## 7. Guia de Solução de Problemas

- **LCD não exibe informações:** Verifique o endereço I2C do display (os mais comuns são 0x27 e 0x3F). As conexões de SDA e SCL também devem ser conferidas.
- **"Aguardando GPS..." persiste:** O módulo GPS precisa de uma visão clara do céu para obter o sinal. Teste em um ambiente externo ou próximo a uma janela. A primeira aquisição pode levar alguns minutos.
- **Falha na conexão Wi-Fi:** Verifique se as credenciais (SSID e senha) no código estão corretas.
- **Dados não chegam na plataforma:** Confirme se o endereço do endpoint da API no firmware está correto e se o servidor do floatData está online. Verifique também o Monitor Serial na IDE Arduino para mensagens de erro na requisição HTTP.