Nome do Aluno: Elmer Carvalho de Oliveira Filho

Polo: Feira de Santana Data: 29/07/2025

# Registrador de Movimento com Raspberry Pi Pico W

## **Objetivo Geral**

O projeto tem como objetivo desenvolver um sistema embarcado baseado no Raspberry Pi Pico W que funcione como um registrador de dados de movimento. O sistema captura dados de aceleração e giroscópio do sensor MPU6050, armazena essas informações em um arquivo CSV no cartão SD e fornece feedback em tempo real ao usuário por meio dos recursos da plataforma BitDogLab, como display OLED, LED RGB, buzzer e botões.

### Descrição Funcional

O sistema opera em diferentes estados, gerenciados por uma máquina de estados finita, que incluem inicialização, espera, captura de dados, acesso ao cartão SD e erro. A seguir, detalha-se o funcionamento:

- Captura de Dados: O sensor MPU6050 é lido a uma taxa de 10 Hz, coletando aceleração (eixos X, Y, Z, em g) e giroscópio (eixos X, Y, Z, em °/s). Os dados são salvos em um arquivo CSV com cabeçalho numero amostra, accel x, accel y, accel z, giro x, giro y, giro z.
- **Armazenamento**: O cartão SD é montado/desmontado com segurança, e cada sessão de captura gera um novo arquivo CSV com nome único (e.g., imu\_data\_001.csv).
- Interação com o Usuário:
  - Display OLED SSD1306: Exibe o estado do sistema (e.g., "Gravando...", "SD Não Detectado"), número de amostras coletadas e feedback de ações.
  - **LED RGB**: Indica estados com cores: amarelo (inicializando), verde (pronto), vermelho (capturando), azul piscando (acessando SD) e roxo piscando (erro).
  - **Buzzer**: Emite beeps curtos (e.g., 1000 Hz por 100 ms para iniciar captura, dois beeps para parar).
  - Botões: Botão 1 inicia/para a captura, Botão 2 monta/desmonta o SD, e Botão
    3 reinicia no modo BOOTSEL. Interrupções com debounce garantem precisão.
- **Análise Externa**: Um script Python lê o CSV e gera gráficos de aceleração e giroscópio, plotando os dados em função do número da amostra.

Os modos de operação são controlados por eventos assíncronos de botões, com delays para evitar múltiplos acionamentos. A lógica de captura é síncrona com o período de amostragem, garantindo consistência nos dados.

#### **Detalhes Técnicos**

### Periféricos da Plataforma BitDogLab/RP2040

- MPU6050: Conectado via I2C (pinos 0 e 1), fornece dados de aceleração e giroscópio com alta precisão.
- Cartão SD: Interface SPI gerenciada pela biblioteca FatFs, com montagem/desmontagem segura.
- **Display OLED**: I2C (pinos 14 e 15), exibe informações em tempo real.
- LED RGB: Pinos GPIO (11, 12, 13) controlam cores para feedback visual.
- **Buzzer**: Pino PWM (21) para alertas sonoros.
- **Botões**: Pinos 5, 6 e 22 com interrupções e pull-up interno.

### Código

O código principal em C é estruturado com funções modulares para configuração de hardware, gerenciamento de estados, captura de dados e interação com o usuário. Destacam-se:

- Interrupções de Botões: O handler button\_irq\_handler registra eventos com debounce de 200 ms, processados assincronamente por process button events.
- **Máquina de Estados**: Gerencia transições entre estados (STATE\_INITIALIZING, STATE WAITING, etc.) com atualizações visuais e sonoras.
- Captura de Dados: process\_imu\_sample lê o MPU6050 a cada 100 ms e salva no CSV via save sample to csv.
- **Script Python**: Utiliza pandas para leitura do CSV e matplotlib/seaborn para gerar gráficos de aceleração e giroscópio, com validação de colunas e tratamento de erros.

#### Links

- Repositório contendo código e descrição do projeto: https://github.com/Elmer-Carvalho/motion datalogger
- Vídeo Demonstrativo: <a href="https://drive.google.com/file/d/10IpvFW3un6nrV7ygJM97Kd\_TXPYo6JXX/view?us">https://drive.google.com/file/d/10IpvFW3un6nrV7ygJM97Kd\_TXPYo6JXX/view?us</a> <a href="p=sharing">p=sharing</a>