Etapa 2 – Arquitetura e Modelagem

Projeto: Sistema Embarcado de Identificação de Módulos Roubados via CAN
Autor: Arthur Franco Neto

2.1 Arquitetura do Sistema

O sistema será desenvolvido na plataforma **BitDogLab (Raspberry Pi Pico W)**, integrando:

- **MCP2515** (com transceiver CAN) para comunicação com o barramento CAN via SPI.
- **Display OLED 128x64** para interface visual.
- **LED RGB** para indicação de status.
- **Botões** para seleção de modos e interação.
- **Buzzer** para alertas sonoros.
- **Bluetooth** para interação com aplicativo móvel (App Inventor).
- **Wi-Fi** para sincronização com servidor remoto.

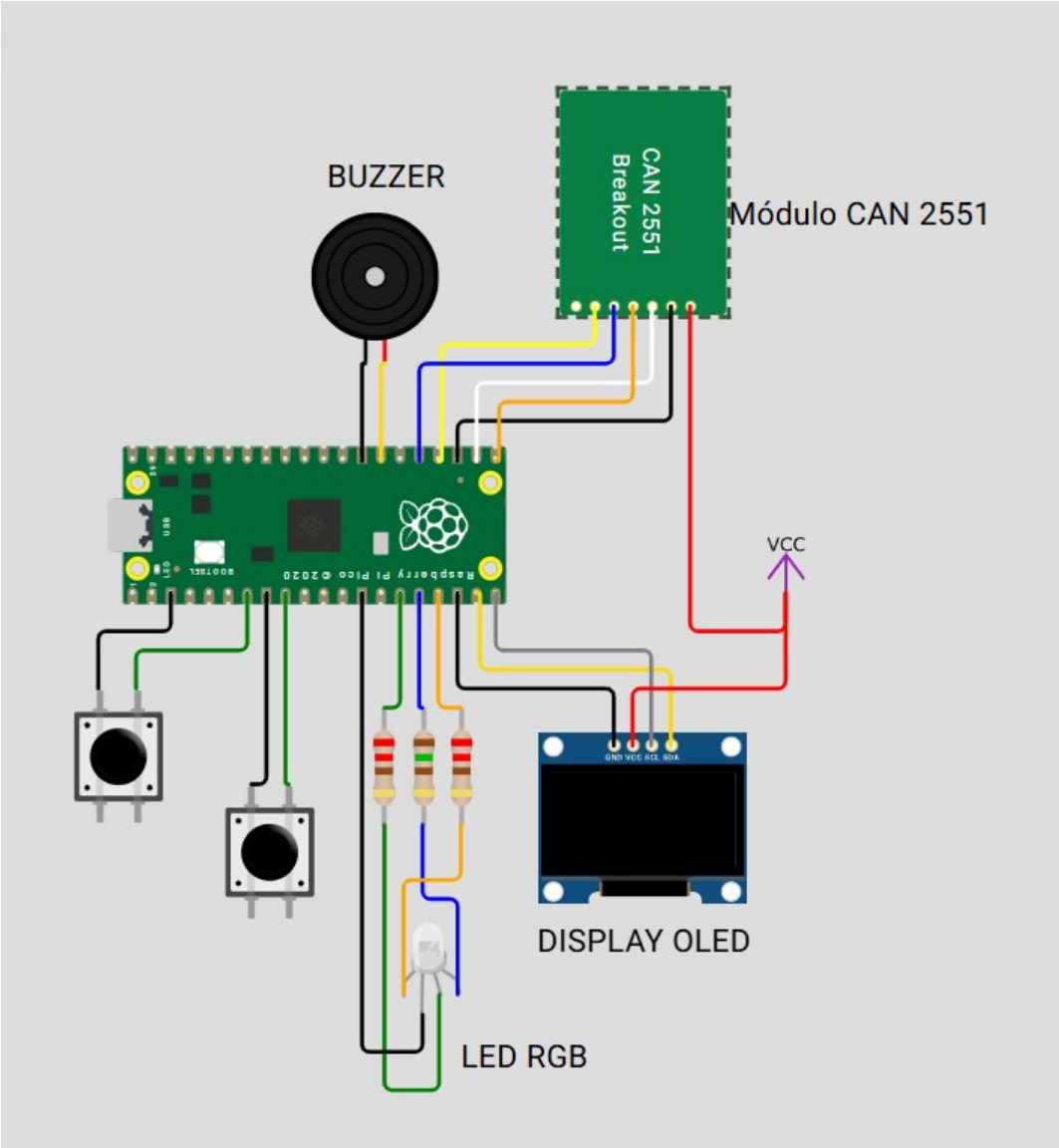
Dois modos operacionais serão implementados:

1. **Modo 1 – Verificação Direta:** consulta módulos no CAN e compara Identificações com whitelist.
2. **Modo 2 – Verificação por VIN:** recebe VIN, obtém lista esperada de módulos e verifica se IDs obtidos correspondem.

Pinagem e Conexões

Componente / Função	Pinos GPIO Pico W	Interface
MCP2515 (CAN)	16 (SPI0_MISO)	SPI0
	17 (SPI0_CS)	
	18 (SPI0_SCK)	
	19 (SPI0_MOSI)	
	20 (RESET/INT)	Controle
Display OLED 128x64	14 (I²C0_SDA)	I²C0
	15 (I²C0_SCL)	
Botão 1	5	Digital IN
Botão 2	6	Digital IN
Buzzer	10	PWM / OUT
LED RGB	11 (R)	PWM / OUT
	12 (G)	PWM / OUT
	13 (B)	PWM / OUT
Wi-Fi	Interno no Pico W	802.11b/g/n

2.3 Diagrama de Hardware



2.4 Blocos Funcionais

Aquisição de Dados CAN

- **Responsabilidade:** Receber informações dos módulos conectados ao barramento CAN via MCP2515.
- **Funções principais:**
    - Envio de comandos de leitura para módulos específicos.
    - Recepção de respostas dos módulos.
    - Detecção de erros de comunicação CAN (timeout, CRC, ausência de resposta).

Processamento e Validação

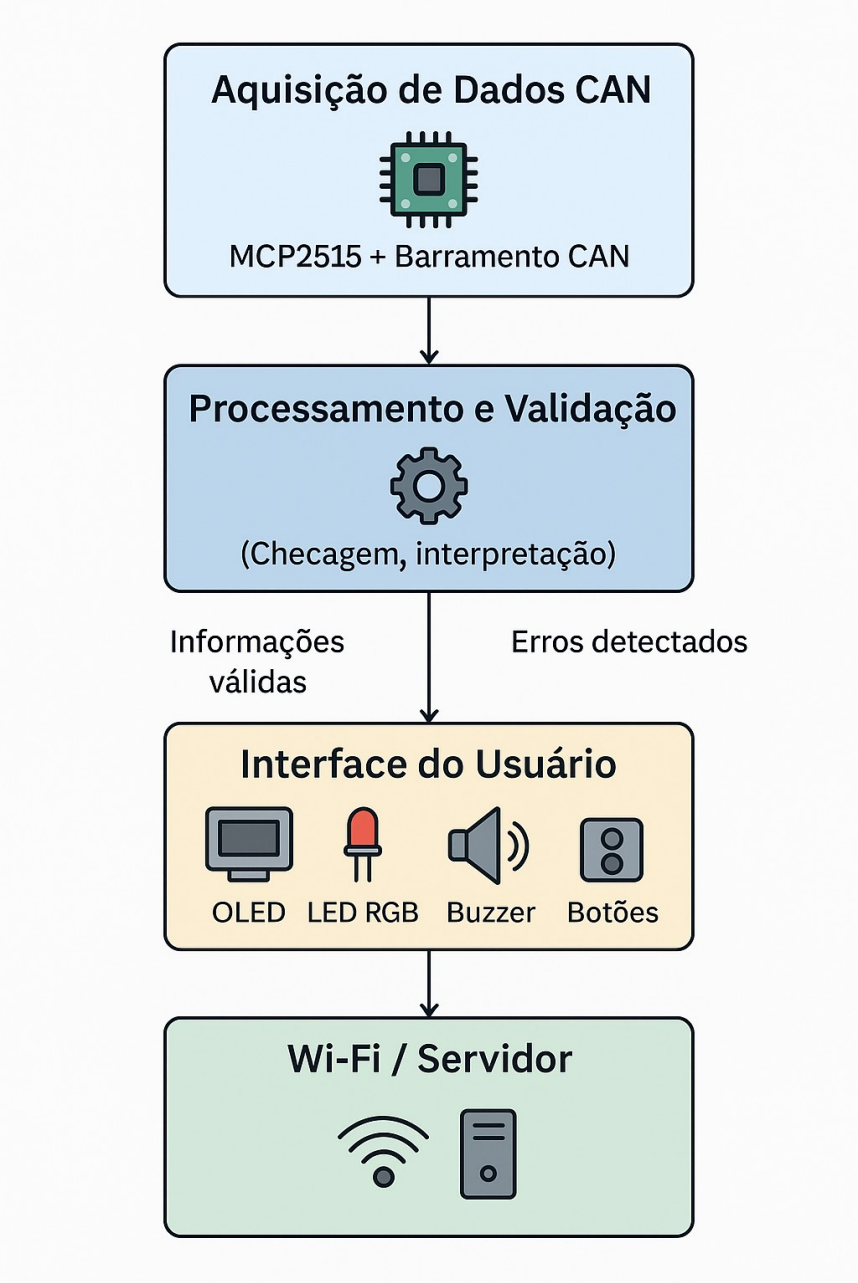
- **Responsabilidade:** Interpretar e validar os dados recebidos dos módulos.
- **Funções principais:**
    - Validação de dados (checksum, consistência).
    - Conversão de informações para formato legível.
    - Identificação de falhas ou módulos não respondendo.

Interface do Usuário

- **Responsabilidade:** Exibir informações e alertas ao usuário.
- **Funções principais:**
    - Display OLED 128x64: status dos módulos e mensagens de erro.
    - LED RGB: indicações visuais (verde: OK, amarelo: aviso, vermelho: falha).
    - Buzzer: alertas sonoros para erros críticos.
    - Botões: navegação entre módulos e atualização manual de dados.

Comunicação Externa via Wi-Fi

- **Responsabilidade:** Sincronizar dados com servidor remoto.
- **Funções principais:**
    - Envio periódico de informações dos módulos.
    - Recebimento de confirmações ou comandos do servidor.
    - Notificação de erros críticos via Wi-Fi.



Fluxograma de Software – Sistema de Monitoramento de Módulos via CAN

Início

- Inicialização do sistema
    - Configuração do MCP2515 (CAN)
    - Inicialização do Wi-Fi
    - Inicialização do Display OLED, LED RGB e Buzzer
    - Configuração dos botões

Loop Principal

1. **Aquisição de Dados CAN**
    - Envia comando de leitura para módulo específico
      ↪ Se envio falhar → registra erro de comunicação
    - Recebe resposta do módulo
      ↪ Se timeout/erro → registra falha no módulo
    - Repete para todos os módulos conectados
2. **Processamento e Validação**
    - Verifica integridade dos dados (checksum/consistência)
      ↪ Se inválido → marca erro
    - Converte dados para formato legível
    - Atualiza status interno do módulo (OK / Aviso / Falha)
3. **Interface do Usuário**
    - Atualiza Display OLED com status dos módulos
    - Atualiza LED RGB:
        - Verde: Todos os módulos OK
        - Amarelo: Aviso em algum módulo
        - Vermelho: Falha crítica
    - Emite Buzzer se houver erro crítico
    - Lê botões:
        - Navegação entre módulos
        - Atualização manual de dados
4. **Comunicação Externa via Wi-Fi**
    - Envia dados coletados para servidor
    - Recebe confirmações ou comandos

- Notifica erros críticos via Wi-Fi