

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.
- **Módulo SDCard** para armazenamento de whitelist, base VIN e logs.
- **Bluetooth** para interação com aplicativo móvel (App Inventor).
- **Wi-Fi** para sincronização com servidor remoto.

Dois modos operacionais serão implementados:

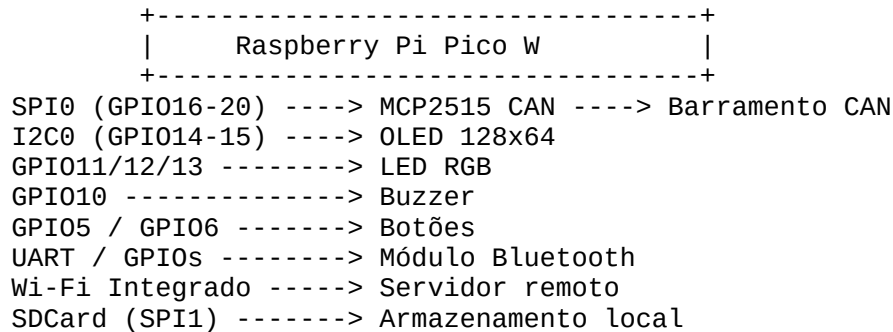
1. **Modo 1 – Verificação Direta:** consulta módulos no CAN e compara Identificações com whitelist.
2. **Modo 2 – Verificação por VIN:** recebe VIN, obtém lista esperada de módulos e verifica se IDs obtidos correspondem.

2.2 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)	
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
Bluetooth	UART / GPIOs a definir	UART/BLE

Componente / Função	Pinos GPIO Pico W	Interface
Wi-Fi	Interno no Pico W	802.11b/g/n

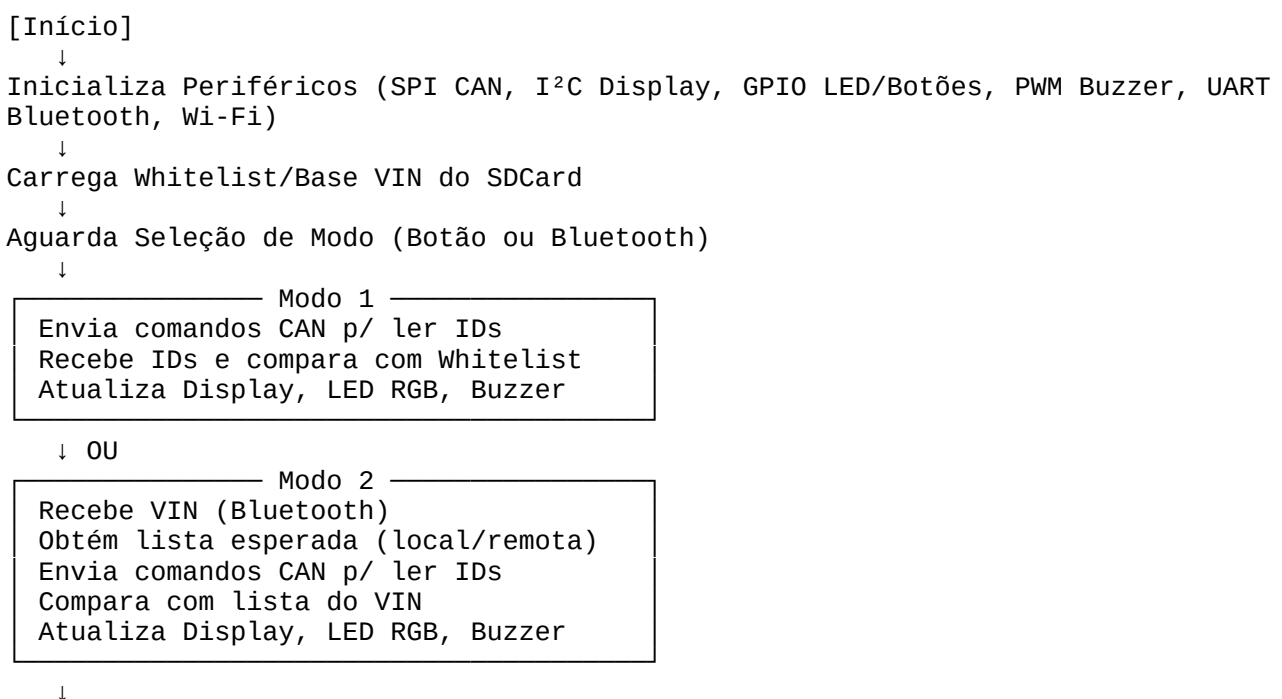
2.3 Diagrama de Hardware



2.4 Blocos Funcionais

1. **Aquisição e Envio CAN** – Recebe e envia quadros CAN.
 2. **Processamento e Validação** – Modo 1 (Whitelist) e Modo 2 (VIN).
 3. **Interface do Usuário** – OLED, botões, LED RGB, buzzer.
 4. **Comunicação Externa** – Bluetooth (App Inventor), Wi-Fi (servidor).
 5. **Armazenamento Local** – Whitelist, base VIN e logs no SDCard.
-

2.5 Fluxograma de Software



Envia resultados via Bluetooth



Se Wi-Fi ativo, sincroniza dados com servidor



[Fim / Novo ciclo]