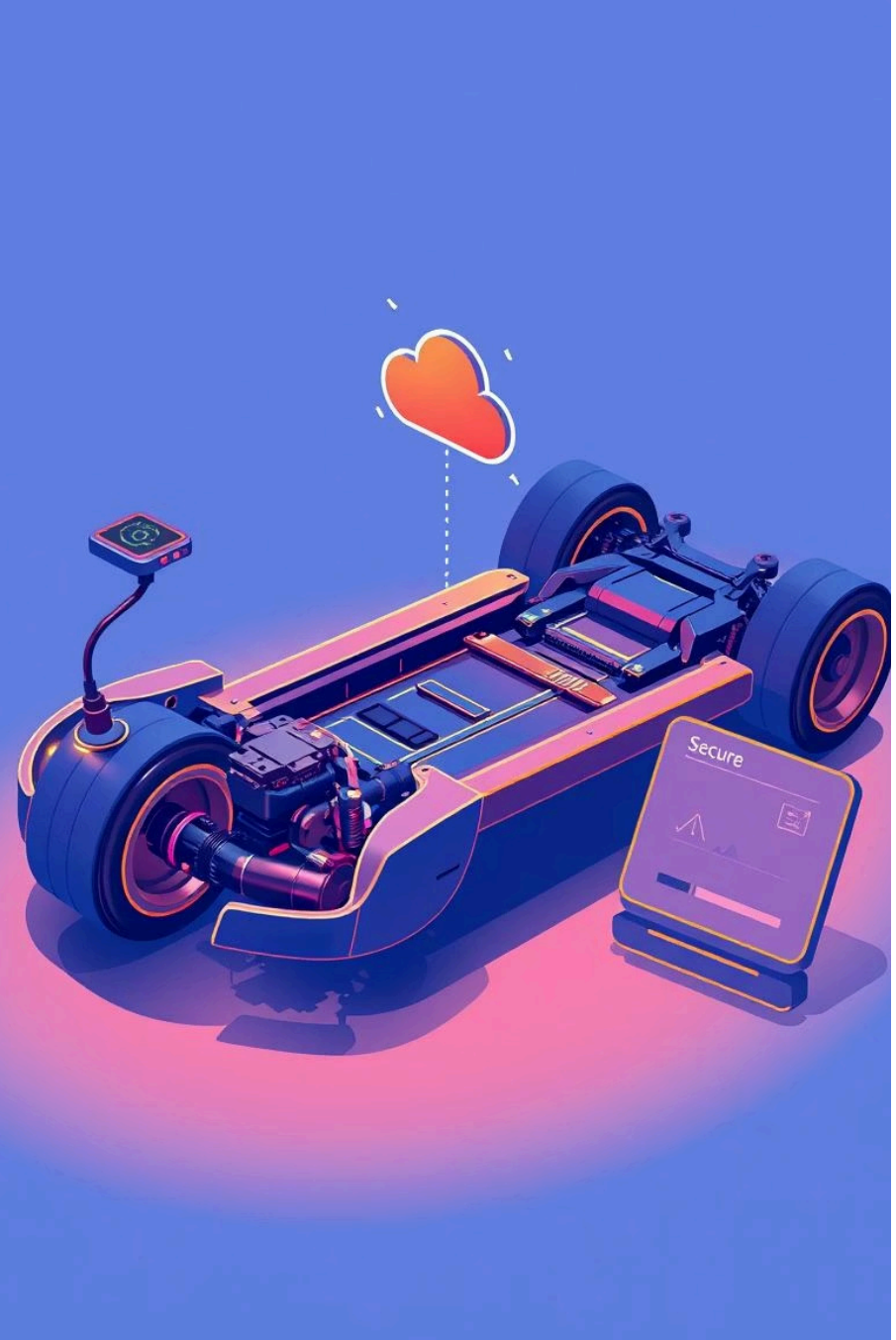


Projeto de Identificação de Módulos automotivos.

Comunicação CAN e MQTT para identificação e validação de módulos automotivos.

O Desafio

Nosso objetivo é desenvolver uma solução completa para a leitura e validação de números de identificação veicular (VIN) e outras informações de módulos automotivos diretamente do barramento CAN. Utilizando tecnologias modernas e eficientes. Buscamos criar uma ferramenta de fiscalização que auxilie no combate à circulação de veículos e módulos roubados.

1

Leitura de VIN

Extrair o VIN de múltiplos módulos (ECU, Painel, ABS, BCM, Airbag) via rede CAN.

2

Comunicação MQTT

Enviar os dados do VIN de forma segura e eficiente para um servidor remoto.

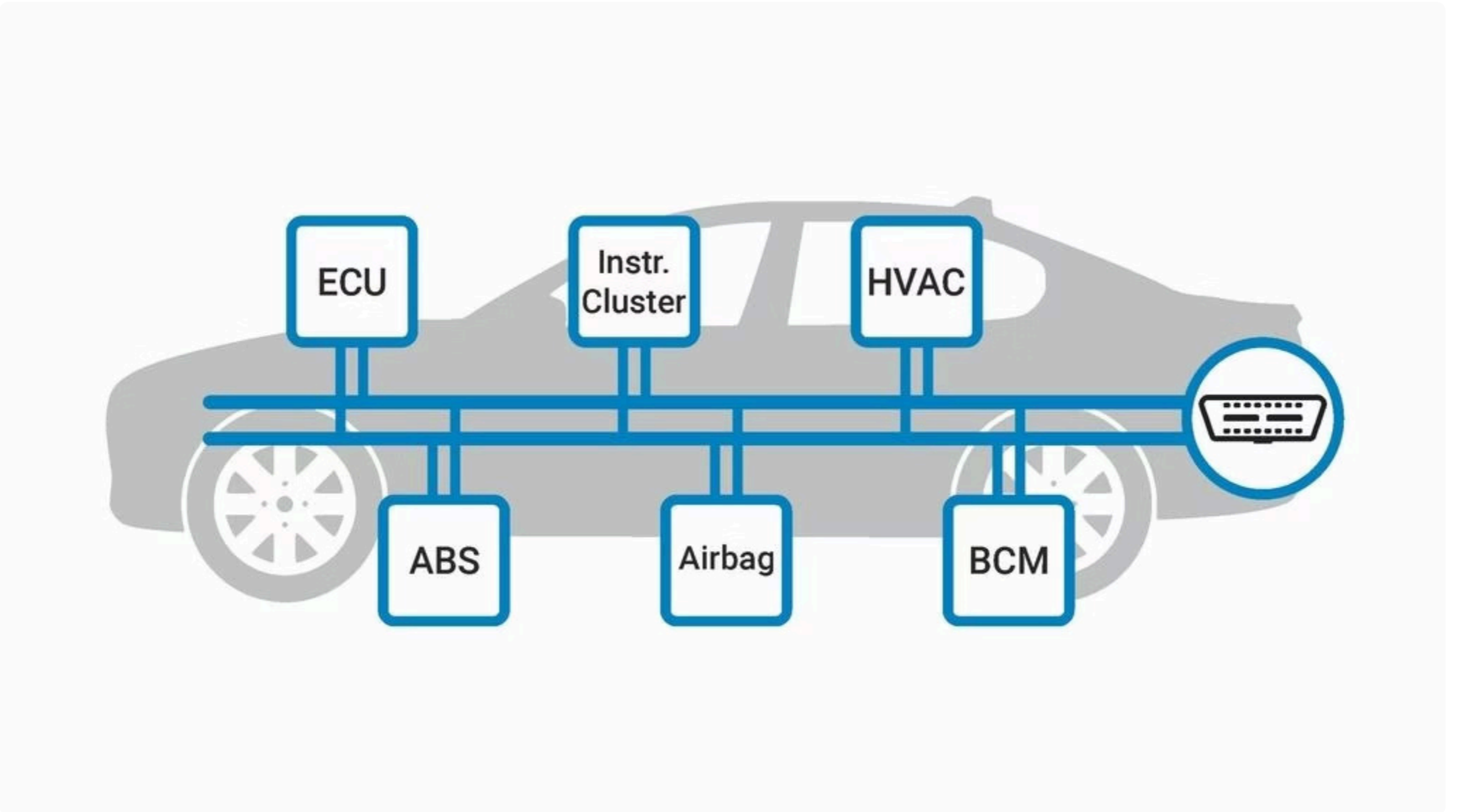
3

Validação e Alerta

Verificar o status do chassi (roubado/OK) e alertar visual e audivelmente.

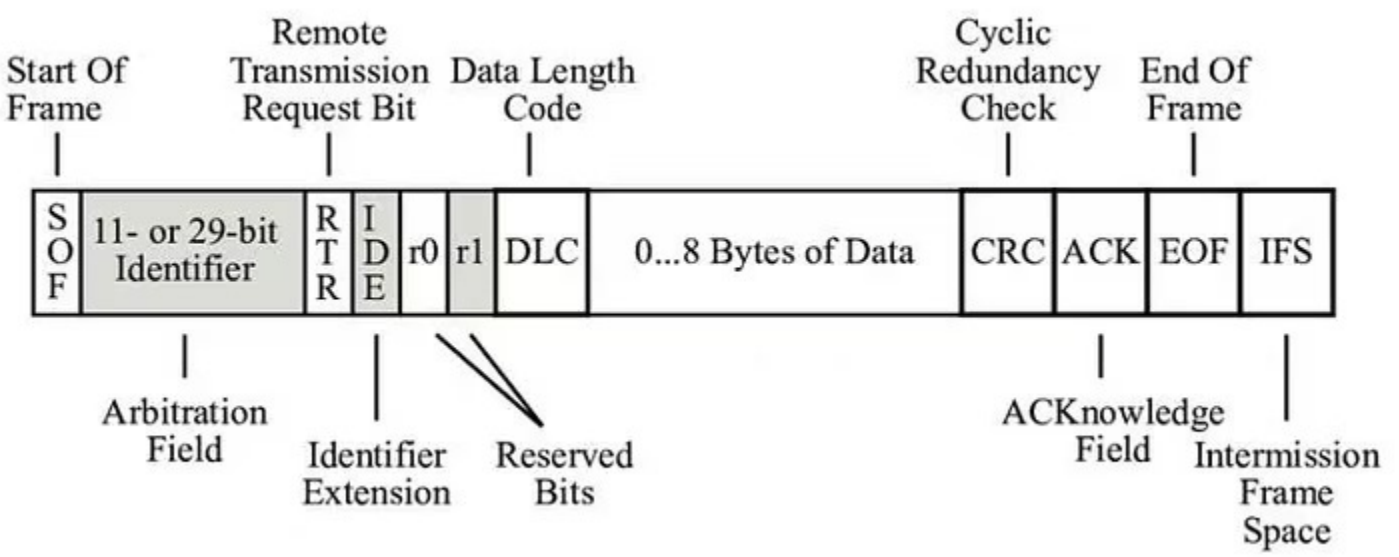
Arquitetura da Solução Embarcada: Raspberry Pi Pico W e Conexão OBD-II

A solução embarcada interage diretamente com o sistema CAN do veículo através da porta OBD-II, permitindo a comunicação com diversos módulos automotivos para extração dos dados através da tomada de diagnostico.



Estrutura de IDs CAN e Comandos de Identificação (ISO 14230 e 14229)

A comunicação no barramento CAN é baseada em identificadores (IDs) que definem a prioridade da mensagem e, em contextos de diagnóstico, servem para endereçar módulos específicos e comandar ações. A norma ISO 14230, também conhecida como KWP2000, é fundamental para padronizar como os módulos eletrônicos de veículos (ECUs) são identificados e como se comunicam para fins de diagnóstico e validação de informações.



Identificadores CAN para Módulos

No CAN, os IDs não apenas determinam a prioridade de uma mensagem, mas também podem ser usados para indicar a origem ou o destino de dados. Em diagnósticos, IDs específicos são reservados para requisições do scanner e respostas dos módulos, permitindo que a ferramenta de diagnóstico "converse" com cada ECU individualmente (ex: ECU do motor, ABS, Airbag).

Norma ISO 14230 (KWP2000)

A ISO 14230 (Keyword Protocol 2000) define um conjunto de serviços de diagnóstico que permitem à ferramenta de diagnóstico solicitar informações de identificação, ler códigos de falha, acionar atuadores, etc. É uma camada de aplicação que opera sobre o barramento CAN, utilizando os IDs CAN para estabelecer a comunicação.

Protocolo UDS (ISO 14229)

O Unified Diagnostic Services (UDS) é o sucessor do KWP2000 e representa um protocolo de comunicação de diagnóstico padronizado globalmente (ISO 14229). Ele oferece um conjunto unificado de serviços de diagnóstico que operam sobre diversas camadas de transporte, incluindo CAN (via ISO 15765-2, conhecido como DoCAN). O UDS permite funcionalidades mais avançadas como leitura de memória, flash reprogramming e controle de rotinas, sendo essencial para diagnósticos modernos e engenharia automotiva.

Arquitetura da Solução Embarcada

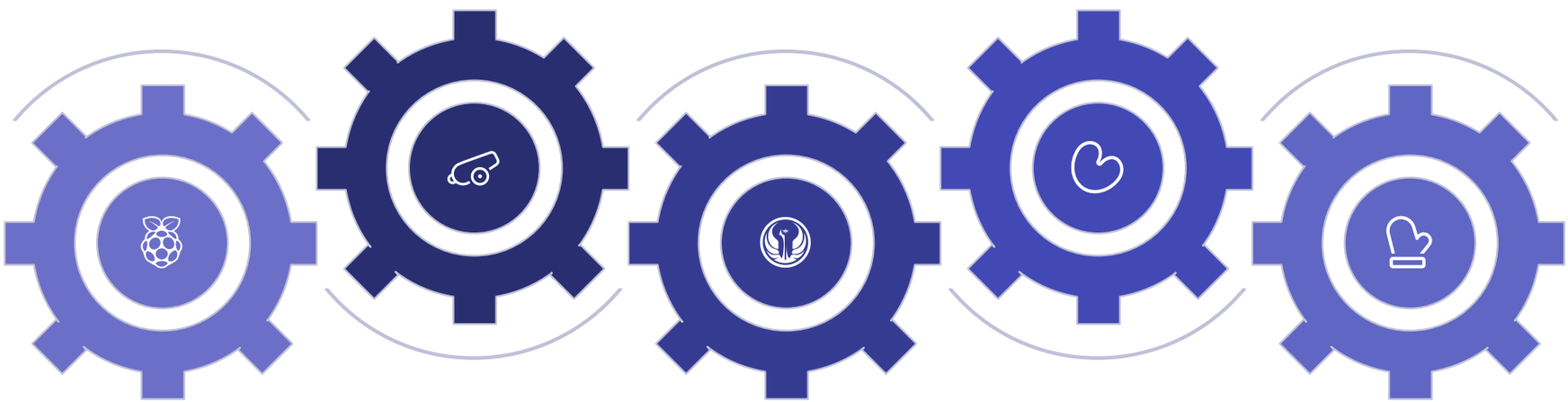
A solução utiliza um Raspberry Pi Pico W como cérebro, integrando capacidades de comunicação CAN e Wi-Fi para uma leitura robusta e envio de dados.

MCP2515 CAN

Handles CAN bus communication

Alert Outputs

LEDs and buzzer for alerts



Pico W Core

Central controller with Wi-Fi and GPIO

OLED Display

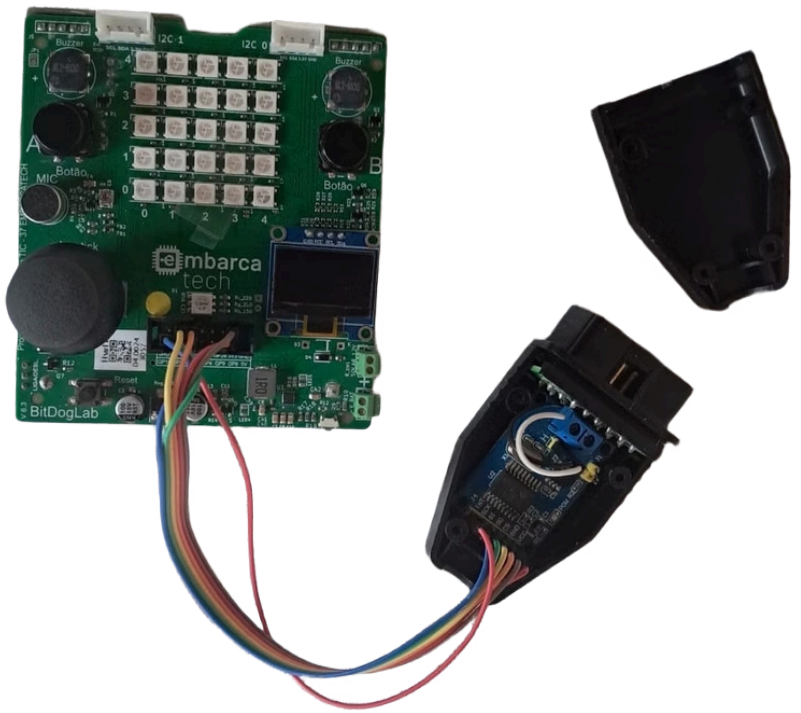
Shows status and telemetry

MQTT Server

Cloud broker via Wi-Fi

O hardware é compacto e de baixo custo, ideal para aplicações embarcadas e de campo.

Prototipagem



A fase de prototipagem envolveu a montagem dos componentes para garantir a comunicação eficiente entre o Raspberry Pi Pico W e o barramento CAN do veículo.

Montagem do Shield CAN

A integração do chip controlador CAN MCP2515 com o transceptor TJA1050 é crucial. Este "shield" é conectado ao Raspberry Pi Pico W através da interface SPI (Serial Peripheral Interface). Os pinos MOSI, MISO, SCK e CS do Pico W são mapeados para os pinos correspondentes do MCP2515, permitindo a transmissão e recepção de dados CAN.

Montagem do Cabo OBD com CAN HI e CAN LOW

Para interagir com o veículo, foi preparado um cabo OBD-II. Os pinos mais importantes para a comunicação CAN são:

- **Pino 6 (CAN-H):** Responsável pela linha de dados de alta tensão do barramento CAN.
- **Pino 14 (CAN-L):** Responsável pela linha de dados de baixa tensão do barramento CAN.

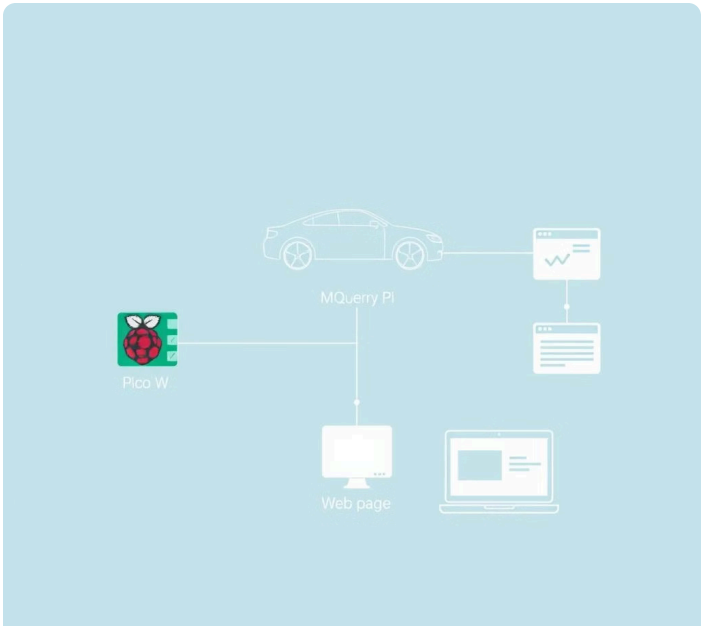
Esses pinos foram conectados diretamente ao transceptor TJA1050 no shield CAN, garantindo a interface física com a rede veicular.

Integração de Dados e Interface Web

Os dados coletados pelo Pico W são processados e validados por uma interface web intuitiva, desenvolvida com Flask e TinyDB.

Fluxo de Comunicação

- Pico W lê VIN de módulos automotivos.
- Pico W envia VIN via MQTT para o servidor Flask.
- Servidor Flask valida VIN contra um banco de dados (TinyDB).
- Servidor Flask envia resposta de status de volta para o Pico W via MQTT.
- Pico W indica o status (ROUBADO/OK) localmente.
- Usuários autorizados tem a permissão de adicionar e remover veículos com pendências.



Funcionalidades-Chave e Casos de Uso

Esta solução oferece funcionalidades essenciais para agilizar a validação de módulos, com foco na segurança e eficiência operacional.

Leitura Multi-Módulo

Capacidade de ler VIN de diversas ECUs (ABS, Airbag, BCM), aumentando a confiabilidade.

Atualização em Tempo Real

Integração MQTT permite atualização instantânea de status na interface web e no dispositivo.

Gestão Web Simples

Adicione, remova e consulte chassis manualmente via Flask, com feedback imediato.