



Carrinho Pet com Monitoramento de Vídeo

**Caio Vitor Carneiro de Oliveira
Guilherme Achilles de Oliveira e Aguiar
Humberto Alves Mesquita**

1. Descrição do Problema

Muitos tutores de animais de estimação desejam interagir e monitorar seus pets mesmo quando não estão fisicamente presentes. A falta de mobilidade e a visão limitada de câmeras estáticas podem impedir uma interação mais dinâmica e a exploração do ambiente do pet. Além disso, a ansiedade de separação em animais pode ser mitigada com a presença "virtual" do tutor.

O problema a ser resolvido é a necessidade de um dispositivo móvel e interativo que permita aos tutores monitorar seus animais de estimação remotamente, oferecendo uma visão em tempo real do ambiente do pet e a capacidade de movimentar-se pelo espaço. Isso proporcionará uma experiência mais imersiva e interativa para o tutor e o animal.

2. Requisitos Técnicos

2.1. Requisitos Funcionais

- **RF1:** O carrinho deve ser capaz de se mover para frente, para trás, virar à esquerda e à direita através de comandos remotos.
- **RF2:** O sistema deve transmitir um feed de vídeo ao vivo da câmera (ESP32-CAM) para um dispositivo de controle (smartphone/computador).
- **RF3:** A interface de controle deve ser acessível via navegador web ou aplicativo móvel.
- **RF4:** O sistema deve exibir o status da bateria do carrinho na interface de controle.

2.2. Requisitos Não Funcionais

- **RNF1 - Desempenho:** A latência do vídeo deve ser minimizada para permitir controle em tempo real (idealmente abaixo de 200ms).
- **RNF2 - Usabilidade:** A interface de controle deve ser intuitiva e fácil de usar, com botões claros para movimento e visualização do vídeo.
- **RNF3 - Confiabilidade:** O fluxo de vídeo e os comandos de controle devem ser estáveis e robustos, minimizando desconexões.
- **RNF4 - Custo:** O custo total dos materiais para uma unidade deve ser otimizado para ser acessível.
- **RNF5 - Autonomia:** A bateria deve ter autonomia mínima de 2 horas de uso contínuo.

- **RNF6 - Segurança:** A transmissão de vídeo deve ser protegida para evitar acessos não autorizados.

3. Lista de Materiais

A seguir, a lista inicial de materiais necessários para a prototipagem do sistema:

- **Plataforma de Desenvolvimento:**
 - 1x Raspberry Pi pico w - BitDogLab
 - 1x Módulo ESP32-CAM (com módulo de câmera OV2640 ou similar)
- **Atuadores:**
 - 4x Motores DC 5V com caixa de redução (para as rodas)
 - 2x Driver de Motor (ex: L298N)
- **Estrutura:**
 - 1x Chassi de carrinho robótico (plataforma com rodas)
 - Rodas e pneus
- **Sensores:**
 - Câmera (integrada na ESP32-CAM)
 - 1x Módulo Sensor de Nível de Bateria
 - 1x MPU6050
- **Alimentação:**
 - 1x Bateria Recarregável (ex: Li-ion 18650 ou LiPo, com capacidade adequada - 11v)
 - 1x Módulo Carregador de Bateria (ex: TP4056 para Li-ion)
 - 1x Regulador de tensão de 11.1 para 5V
 - 1x Chave Liga/Desliga
- **Componentes Eletrônicos Auxiliares:**
 - Cabos jumper (macho-macho, macho-fêmea, fêmea-fêmea)
 - Protoboard (para prototipagem inicial)
 - Resistores, capacitores (se necessário)
- **Software/Serviços:**
 - Ambiente de Desenvolvimento Integrado (IDE) (ex: Arduino IDE ou VS Code com PlatformIO e SDK da RaspBerry pico W)
 - Bibliotecas ESP32-CAM (para câmera e Wi-Fi)
 - Bibliotecas para controle de motor e sensores
 - Servidor Web embarcado na ESP32 (para interface streaming)
 - Protocolo de Streaming de Vídeo (ex: MJPEG)
 - AWS IoT para controle do carrinho via internet

4. Arquitetura do Sistema

O sistema será composto por três camadas principais:

4.1. Camada de Hardware (Dispositivo Embarcado - Carrinho Pet)

Esta camada consiste no carrinho físico, na ESP32-CAM e nos demais componentes eletrônicos.

- **BITDOGLAGB:** Responsável por:
 - Movimentação do carrinho via AWS IoT.
 - Gerenciamento da conexão com o AWS IoT.
- **ESP32-CAM:** Responsável por:
 - Captura e processamento do vídeo da câmera.
 - Gerenciamento da conexão Wi-Fi.
- **Câmera (OV2640):** Captura o feed de vídeo do ambiente.
- **Driver de Motor e Motores DC:** Recebem os comandos da BitDogLab para controlar o movimento do carrinho (frente, trás, virar).
- **Bateria e Módulo de Carregamento:** Fornecem energia para todo o sistema.
- **Sensores (Opcionais):**
 - **Sensor de Nível de Bateria:** Monitora a tensão da bateria para exibir seu status.

4.2. Camada de Comunicação

Esta camada gerencia o fluxo de dados entre o carrinho e o dispositivo do usuário.

- **Conexão Wi-Fi:** A Bitdoglab vai conectar-se via WIFI, conectando-se ao broker AWS IoT, responsável por enviar os comandos do carrinho para a placa.
- **Conexão Wi-Fi - ESP32 CAM:** Ela atuará como um Access Point (AP) ou se conectará a uma rede Wi-Fi existente, permitindo que o dispositivo de controle do usuário se conecte a ela.
- **Protocolo de Streaming de Vídeo:** MJPEG (Motion JPEG) será utilizado para transmitir o vídeo em tempo real para o navegador web do usuário.
- **Protocolo de Controle:** Comandos de movimento serão enviados via requisições HTTP (GET/POST) para a AWS IoT, sendo redirecionada para o carrinho via MQTT.

4.3. Camada de Interface de Controle (Dispositivo do Usuário)

Esta camada é a interface com a qual o tutor irá interagir para controlar o carrinho e visualizar o vídeo.

- **Navegador Web/Aplicativo Móvel:** Uma página web simples (HTML, CSS, JavaScript) será servida pela ESP32-CAM e a BitDogLab. Esta página conterá:
 - Um elemento que exibirá o stream MJPEG da câmera.
 - Botões ou um joystick virtual para enviar comandos de movimento (frente,

- trás, esquerda, direita).
- Exibição do status da bateria.

5. Metodologia de Desenvolvimento

O projeto seguirá uma abordagem iterativa e incremental, dividida em fases de desenvolvimento:

5.1. Fase de Hardware e Montagem do Carrinho

- **Montagem do Chassi:** Montar a estrutura do carrinho, rodas e motores.
- **Conexão dos Motores:** Conectar os motores ao driver de motor e o driver à ESP32-CAM.
- **Integração dos microcontroladores:** Fixar a ESP32-CAM e a BitDogLab no carrinho, garantindo uma boa visão da câmera.
- **Sistema de Alimentação:** Conectar a bateria e o módulo de carregamento.

5.2. Fase de Conectividade e Servidor Web Básico

- **Configuração do Ambiente de Desenvolvimento:** Instalar a IDE e as bibliotecas da BitDogLab e ESP32-CAM.
- **Configuração de Rede Wi-Fi:** Programar a ESP32-CAM para operar como um Access Point (AP) ou se conectar a uma rede existente.
- **Servidor Web Básico:** Implementar um servidor web simples na ESP32-CAM para servir uma página HTML estática.

5.3. Fase de Controle de Movimento

- **Controle de Motores:** Desenvolver o código para controlar os motores DC (velocidade e direção) usando o driver de motor.
- **Interface de Controle HTML/JS:** Criar botões na página web para enviar comandos de movimento para a ESP32-CAM.
- **Lógica de Movimento na BitDogLab:** Implementar a lógica na para interpretar os comandos HTTP no AWS IoT e enviar via MQTT para o microcontrolador e acionar os motores.

5.4. Fase de Streaming de Vídeo

- **Inicialização da Câmera:** Configurar a ESP32-CAM para inicializar e capturar imagens da câmera.
- **Implementação do Stream MJPEG:** Desenvolver o endpoint no servidor web da ESP32-CAM para transmitir o fluxo de vídeo MJPEG.
- **Exibição do Vídeo na Interface:** Configurar o elemento na página web para exibir o stream MJPEG.

5.5. Fase de Sensores e Recursos Adicionais (Opcional)

- **Monitoramento de Bateria:** Implementar a leitura do sensor de nível de bateria e exibir o status na interface.

6. Plano de Testes

Serão realizados testes em cada fase do desenvolvimento para garantir a funcionalidade e robustez do sistema.

6.1. Testes Unitários

- **Teste de Controle de Motores:** Verificar se os motores respondem corretamente aos comandos de direção e velocidade.
- **Teste de Câmera:** Confirmar que a câmera está capturando imagens e que o stream básico funciona.
- **Teste de Conectividade Wi-Fi:** Assegurar que a ESP32-CAM e a BitDogLab se conecta à rede Wi-Fi ou atua como AP de forma estável.
- **Teste de Sensores (Opcionais):** Verificar a precisão das leituras do sensor de bateria.

6.2. Testes de Integração

- **Teste de Fluxo de Vídeo e Controle:** Verificar se o vídeo é transmitido em tempo real enquanto os comandos de movimento são recebidos e executados.
- **Teste de Interface de Controle:** Validar a usabilidade e responsividade dos botões/joystick na página web.

6.3. Testes de Sistema

- **Teste de Operação Contínua:** Monitorar o carrinho por um período estendido (ex: 1-2 horas) para identificar falhas de estabilidade, superaquecimento ou vazamentos de memória.
- **Teste de Alcance da Rede:** Avaliar a distância máxima de operação do carrinho em diferentes ambientes.
- **Teste de Autonomia da Bateria:** Medir a duração real da bateria sob uso contínuo.
- **Teste de Qualidade de Vídeo:** Avaliar a fluidez e resolução do vídeo em diferentes condições de iluminação e movimento.

7. Movimentação do carrinho

O projeto do robô utilizará a **Raspberry Pi Pico W** da BitdogLab como microcontrolador principal. A comunicação será realizada via Wi-Fi, conectando-se ao broker **AWS IoT Core** através do protocolo **MQTT**. Os comandos de movimentação, gerados em uma interface web, serão publicados em tópicos MQTT e recebidos pelo microcontrolador



Para garantir uma movimentação precisa e estável, será implementado um **controle PID** para o deslocamento linear, mantendo o robô em uma trajetória reta e com velocidade constante. O feedback para este controle será fornecido pelos **encoders acoplados aos motores**. O controle de rotação também utilizará um **algoritmo PID**, sendo que a medição do ângulo de giro será realizada por um **giroscópio**, assegurando que as curvas sejam executadas com precisão.

8. Próximos Passos

Após a conclusão da Etapa 1 (Definição de Requisitos e Lista de Materiais), os próximos passos incluem:

- **Aquisição de Materiais:** Comprar ou providenciar todos os componentes listados.
- **Configuração do Ambiente de Desenvolvimento:** Instalar a IDE (Arduino IDE ou VS Code com PlatformIO) e as bibliotecas necessárias para ESP32-CAM, motores e sensores.
- **Início da Fase de Hardware e Montagem do Carrinho:** Começar a montagem física do carrinho e a interconexão dos componentes eletrônicos.