



Carrinho Pet com Monitoramento de Vídeo Etapa 2

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

Documentação Técnica: Arquitetura do Sistema

1. Arquitetura do Sistema

A arquitetura abaixo ilustra os componentes físicos e suas interconexões:

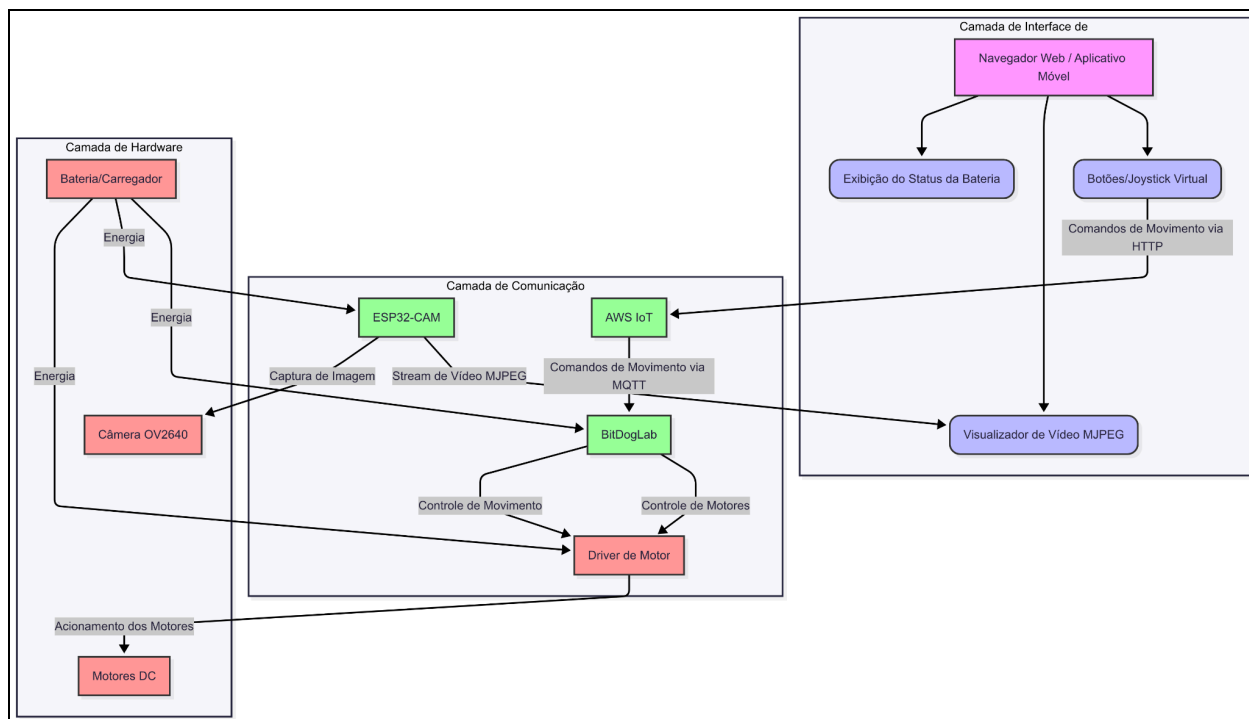


Figura 1 - Arquitetura do Sistema

- **ESP32-CAM:** Responsável pela captura e processamento do vídeo da câmera.
- **BitDogLab (Raspberry Pi Pico W):** Atua como microcontrolador principal, gerenciando a conexão com o AWS IoT e controlando a movimentação do carrinho.
- **Atuadores (Driver e Motores):** Recebem os comandos da BitDogLab para controlar a direção e velocidade do carrinho.
- **Sensores (Câmera):** A câmera (OV2640) captura o vídeo.
- **Alimentação:** Uma bateria recarregável fornece energia para o sistema, e um regulador de tensão garante que os componentes recebam a tensão correta.

1.2. Fluxograma do Software

O software do sistema opera de maneira assíncrona, com o controle de movimento e

o streaming de vídeo funcionando em paralelo. O fluxograma abaixo detalha a lógica de operação dos microcontroladores.

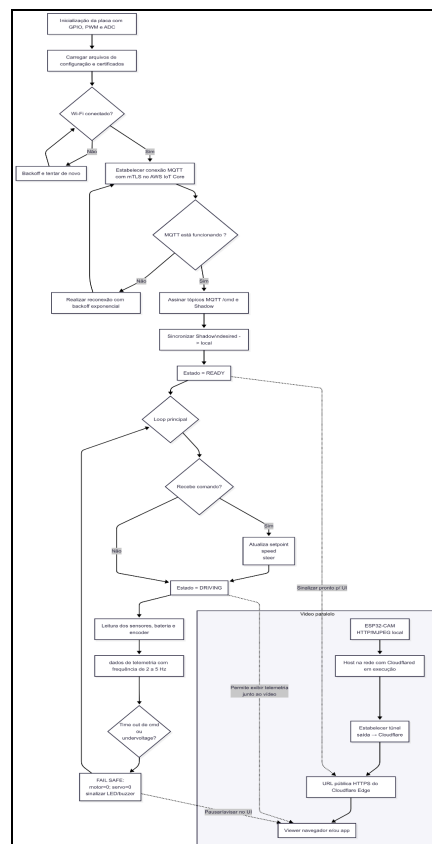


Figura 2 - Fluxograma do Software

- **Lógica da BitDogLab:** Após a inicialização, o microcontrolador entra em um loop principal, esperando comandos MQTT do AWS IoT. Ao receber um comando, ele o interpreta e ajusta os motores ou lê o sensor de bateria e publica o status de volta para a AWS IoT.
- **Lógica da ESP32-CAM:** Opera em um loop separado, capturando frames da câmera. Cada frame é codificado no formato MJPEG e transmitido através do servidor web embarcado para a interface do usuário.
- **Lógica da Interface Web:** A página web do usuário carrega o visualizador de vídeo MJPEG. O script da página detecta a entrada do usuário (cliques nos botões) e envia comandos via HTTP para a AWS IoT, que por sua vez os encaminha para a BitDogLab via MQTT

1.3 Diagrama de Hardware

O esquema a seguir apresenta a alimentação, processamento, captura de imagem, sensoriamento e controle de motores. Entre os módulos principais, destaca-se a câmera integrada para captura e transmissão de vídeo, conectada ao microcontrolador, que atua como unidade central de processamento e comunicação com os demais dispositivos.

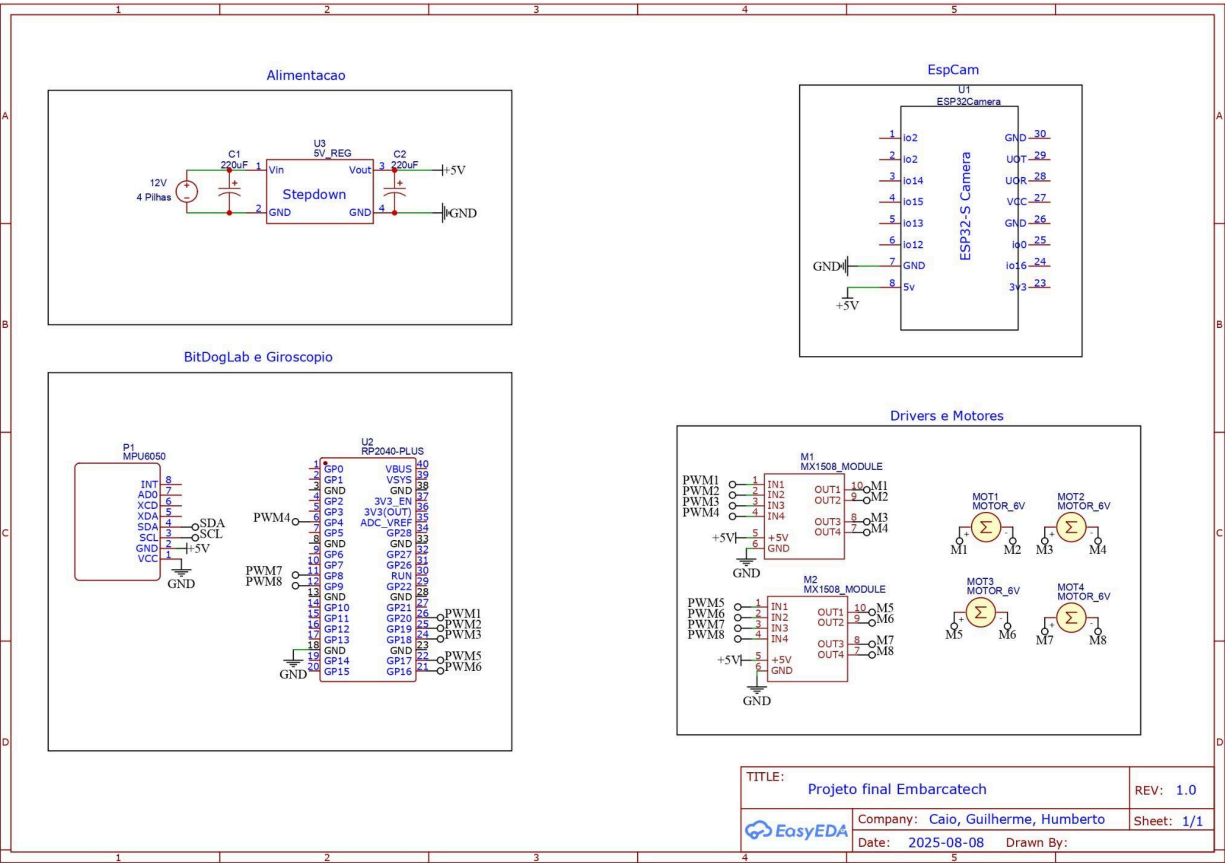
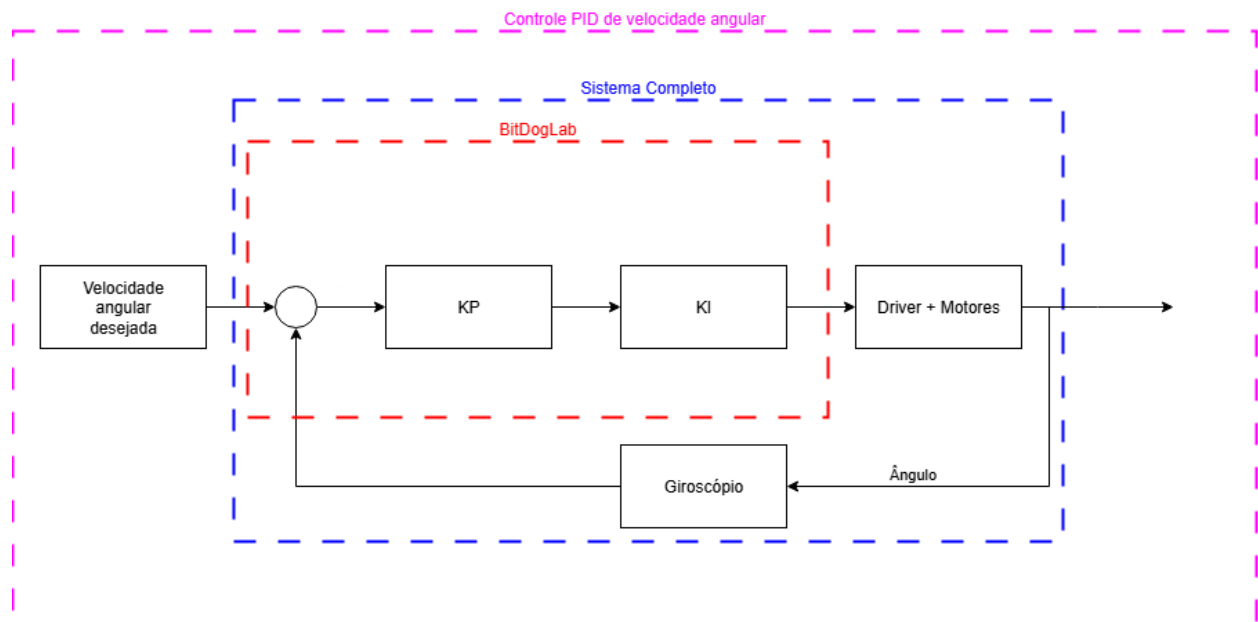


Figura 3 - Esquemático do circuito do projeto final

1.4 Bloco funcional do controle PID para a velocidade angular

Este controle vai ser usado para garantir que o robô ande reto e faça as curvas da melhor forma possível.



Referências:

Github - Disponível em:

https://github.com/EmbarcaTech-2025/projeto-final-caio_guilherme_humberto_projeto_final

Figura 1 - Disponível em:

https://github.com/EmbarcaTech-2025/projeto-final-caio_guilherme_humberto_projeto_final/blob/main/Etapa%202/arquitetura_de_software.png

Figura 2 - Disponível em:

https://github.com/EmbarcaTech-2025/projeto-final-caio_guilherme_humberto_projeto_final/blob/main/Etapa%202/fluxograma_de_software.png

Figura 3 - Disponível em:

https://github.com/EmbarcaTech-2025/projeto-final-caio_guilherme_humberto_projeto_final/blob/main/Etapa%202/Diagrama_de_Hardware.jpg

Figura 4 - Disponível em:

https://github.com/EmbarcaTech-2025/projeto-final-caio_guilherme_humberto_projeto_final/blob/main/Etapa%202/controlador_pid_posicao.drawio_1.webp