



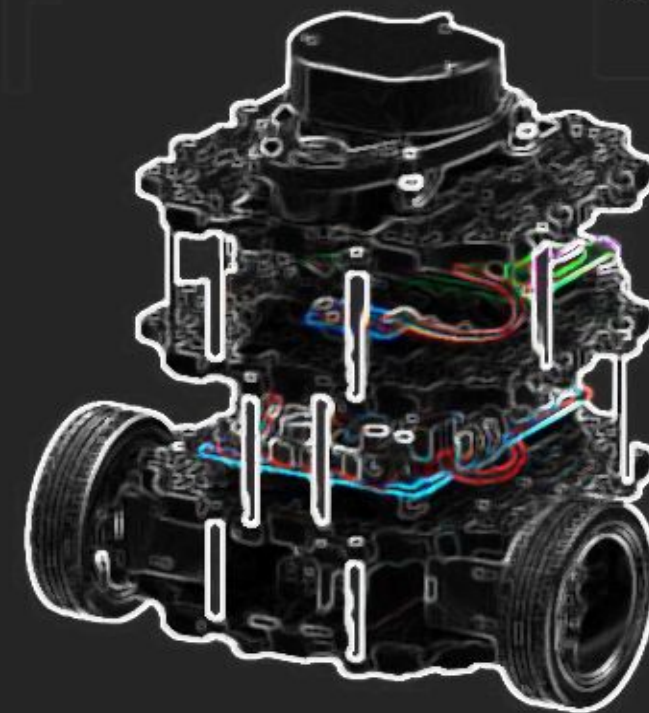
Robô mapeador de baixo custo

DAVI R. G. VALLE
GUSTAVO F. LEWIN
JOSÉ MÁRIO N.

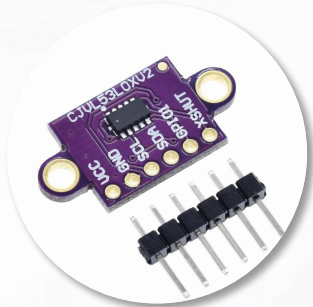
INTRODUÇÃO

- O TurtleBot Burger mostrado na figura é um robô de *hardware* e *software* abertos, servindo de plataforma de entrada para a robótica móvel. No entanto, alguns componentes do *TurtleBot* são de difícil acesso devido ao custo, especificamente os motores e o Lidar.
- Portanto, nosso desafio foi o de desenvolver um robô inspirado no TurtleBot, que consiga mapear, navegar e realizar tarefas básicas, porém com um custo reduzido.

TurtleBot3
Burger

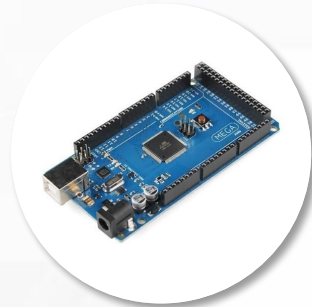


MATERIAIS



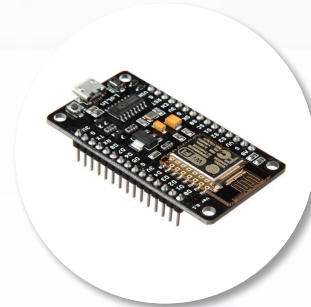
SENSOR VL53L0X

Sensor que é capaz de medir distância por meio de um laser infravermelho



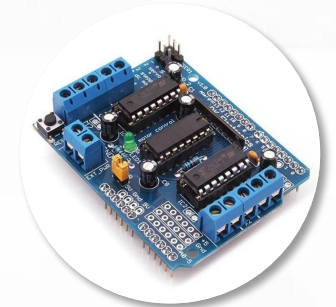
ARDUINO MEGA

Microcontrolador responsável pelo código fonte do robô



MÓDULO ESP

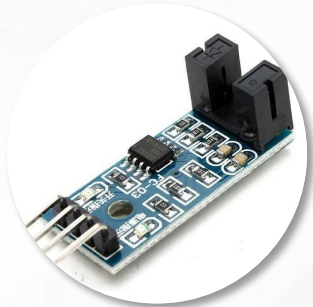
Responsável por conectar nosso microcontrolador ao Wi-Fi



SHIELD L293D

Reúne alta tensão, alta corrente e controle, nos permite alimentar motores, servo e o sistema.

MATERIAIS



PAR DE MÓDULOS ENCODER ÓPTICO

Usado para identificar a mudança de posição das rodas



MOTORES E RODAS

Kit motores DC 3-6V e rodas de borracha



POWER BANK 10000 mAh

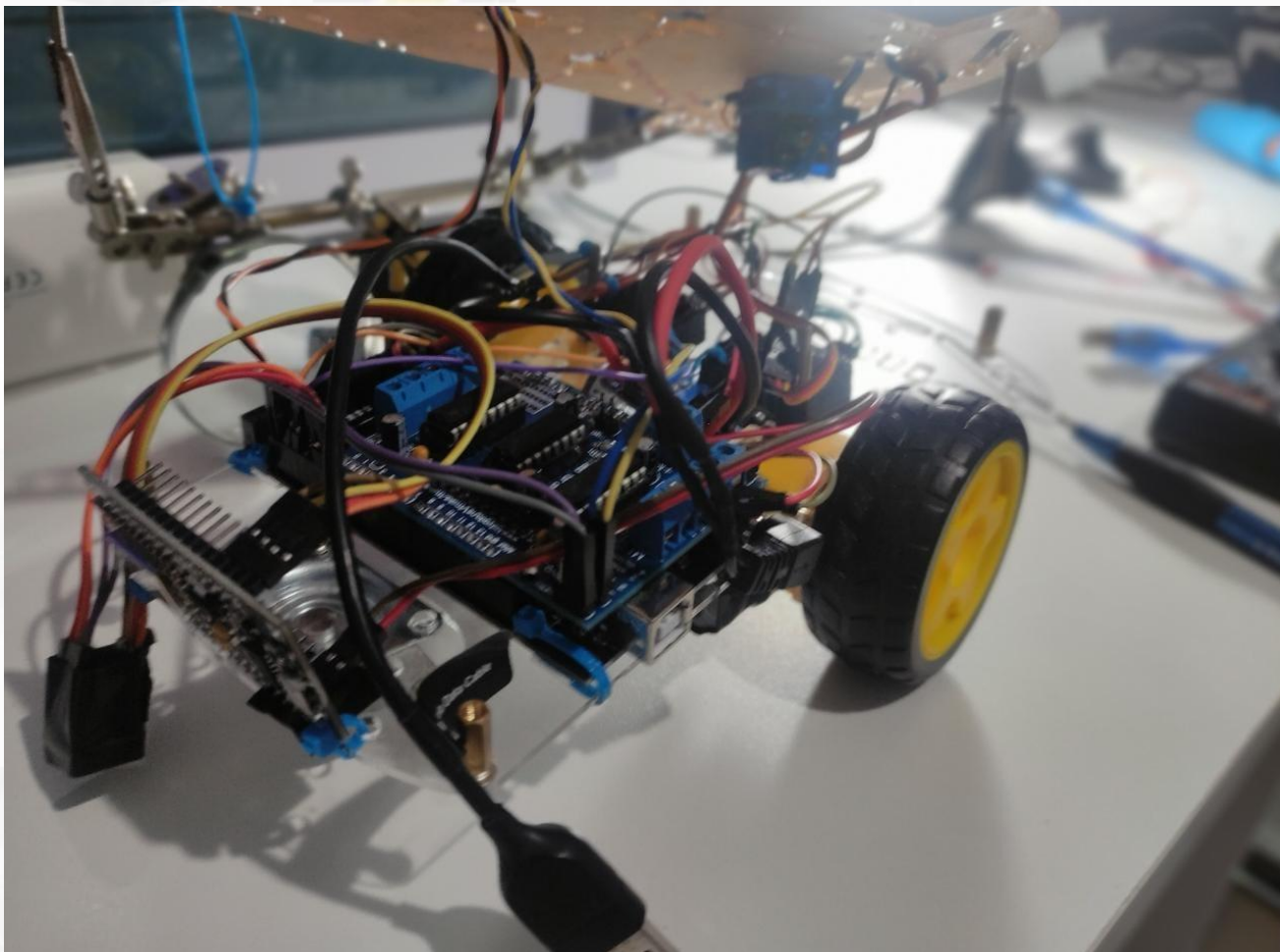
Energia geral do carrinho



PAR DE CHASSIS DE ACRILICO 2WD

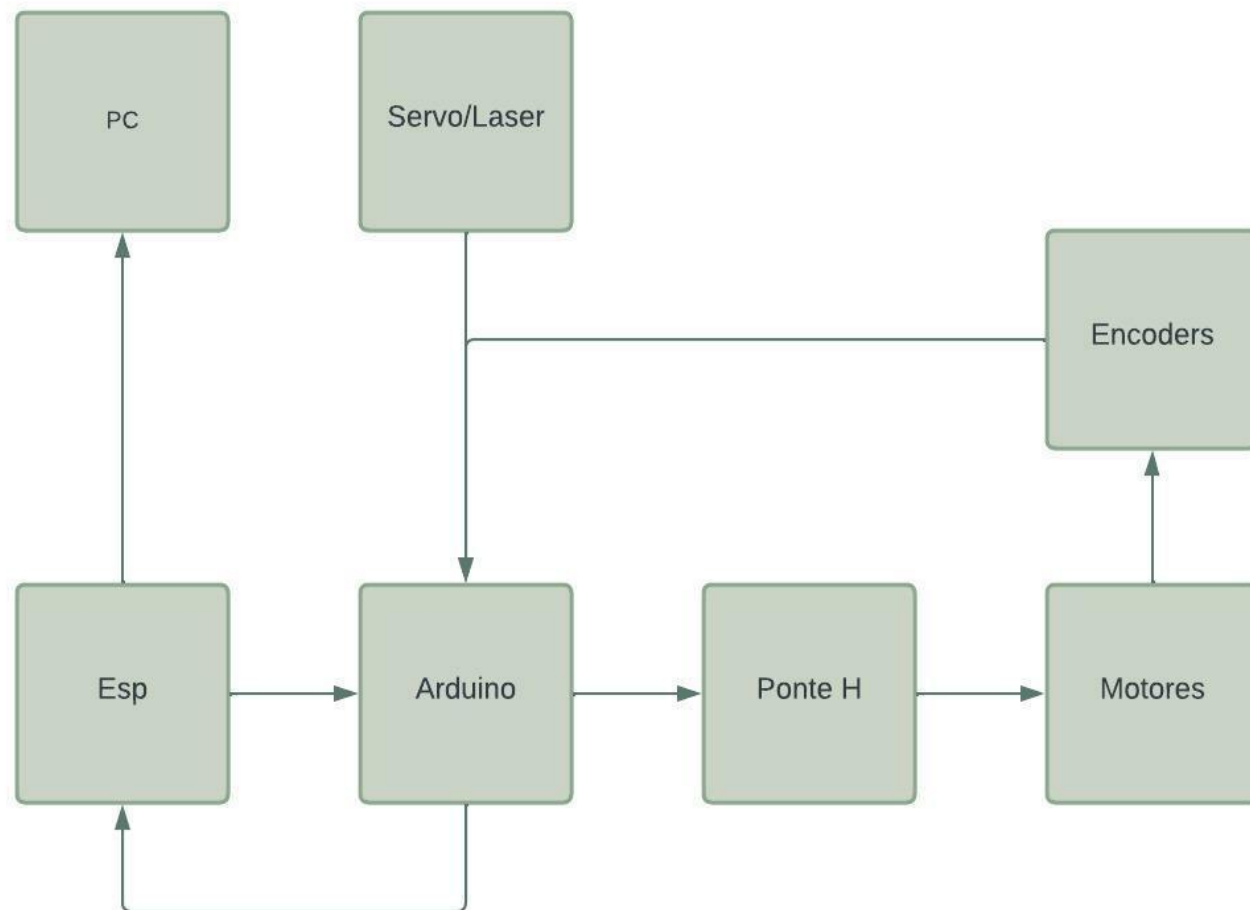
Kit chassi de acrílico. Base do robô

LÓGICA DE FUNCIONAMENTO



- Usamos o KIT 2WD (composto pelo chassi, motores e rodas) como estrutura básica do robô.
- Com os *encoders*, é possível medir a velocidade das rodas, e assim estimar a odometria do robô (i.e. a posição dele em relação às coordenadas iniciais).
- Durante todo o processo, o sensor *laser* capta novos pontos e o servo motor faz a varredura para o preenchimento do mapa.
- Para controle dos motores foi usada uma ponte H L293D. Também é por ela que passa a alimentação de todo o sistema.

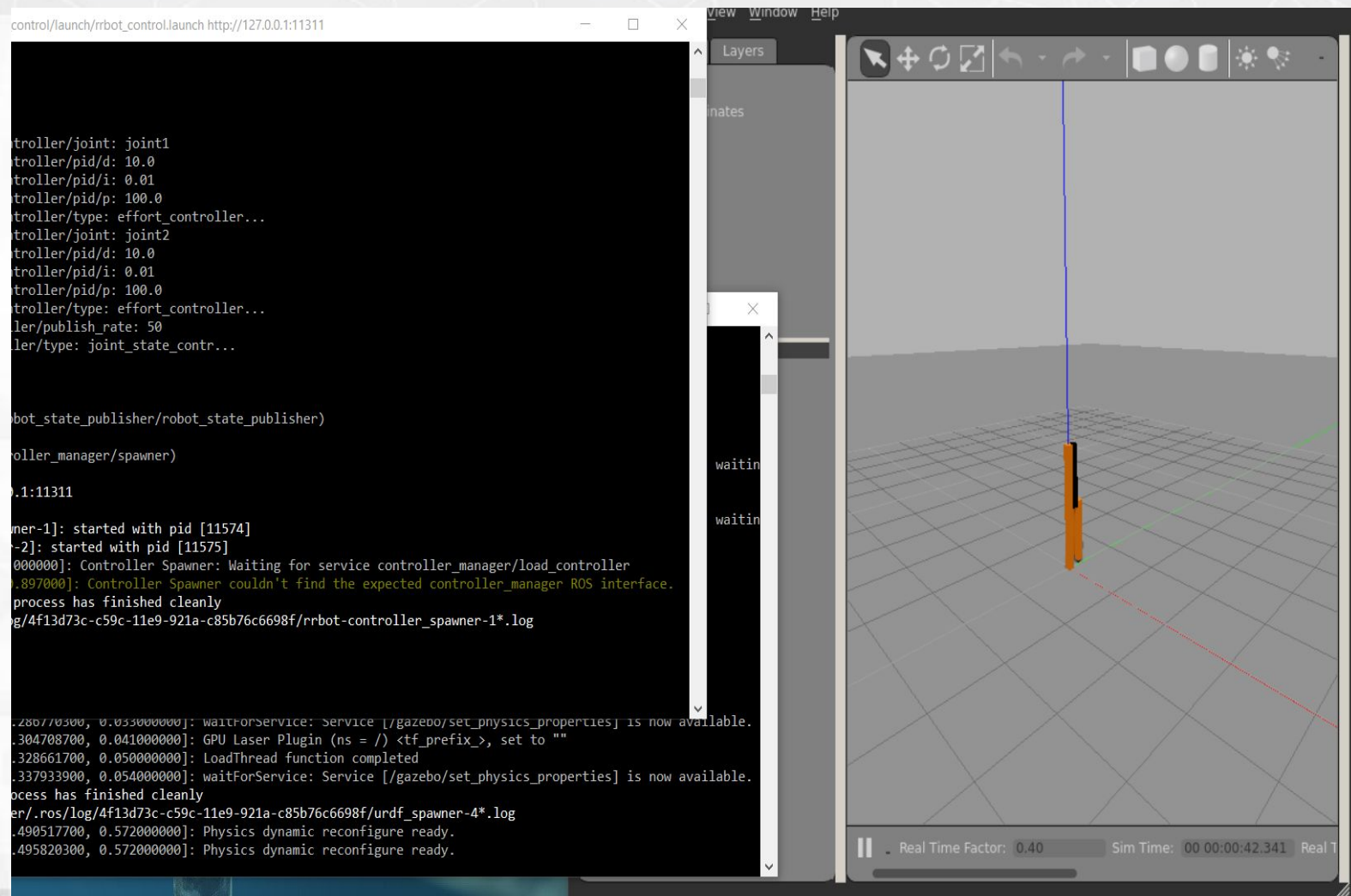
LÓGICA DE FUNCIONAMENTO



- O Arduino Mega controla todo o sistema e recebe as informações do *laser* e dos *encoders*
- ESP intermedia a comunicação do PC com o MEGA por WiFi.
- Já no PC, utilizamos o *Robot Operating System* (ROS) para controlar o robô e visualizar o mapa.

SOFTWARE

- *Robot Operating System (ROS)* é uma coleção de *frameworks* de *software* para desenvolvimento de robôs.
- ROS fornece serviços padrões de sistema operacional, como abstração de *hardware*, controle de dispositivos de baixo nível, a implementação de funcionalidades comumente usadas, passagens de mensagens entre processos e gerenciamento de pacotes



DEMONSTRAÇÃO

CONTROLE PROPORCIONAL

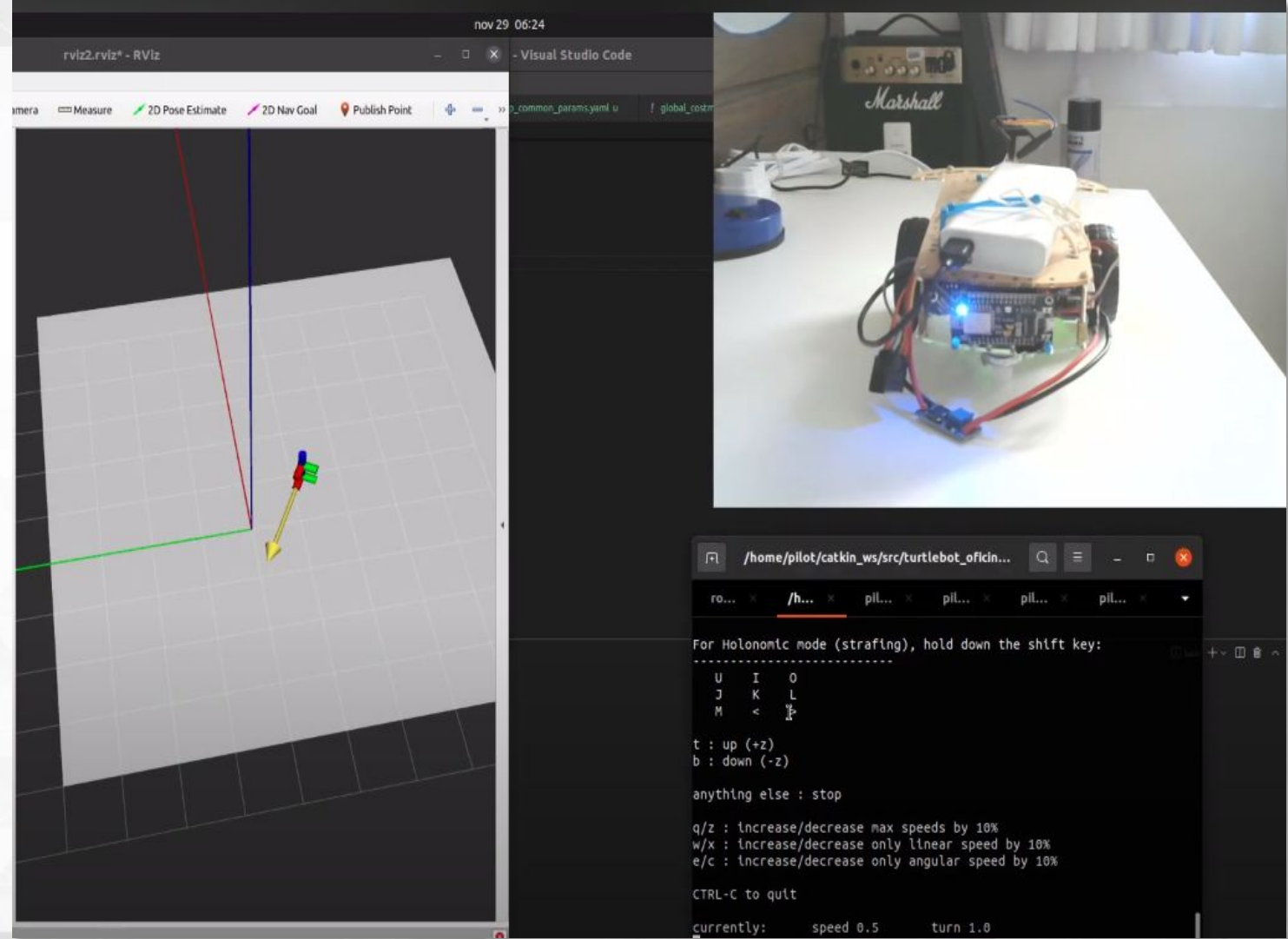
- Como prova de conceito para controle proporcional aplicado às rodas, tentamos primeiramente estabilizar o motor em 60RPM.
- Foi observado que o motor realmente apresentava um tique (indicação sonora) por segundo, o que também é mostrado na figura.



DEMONSTRAÇÃO

ODOMETRIA

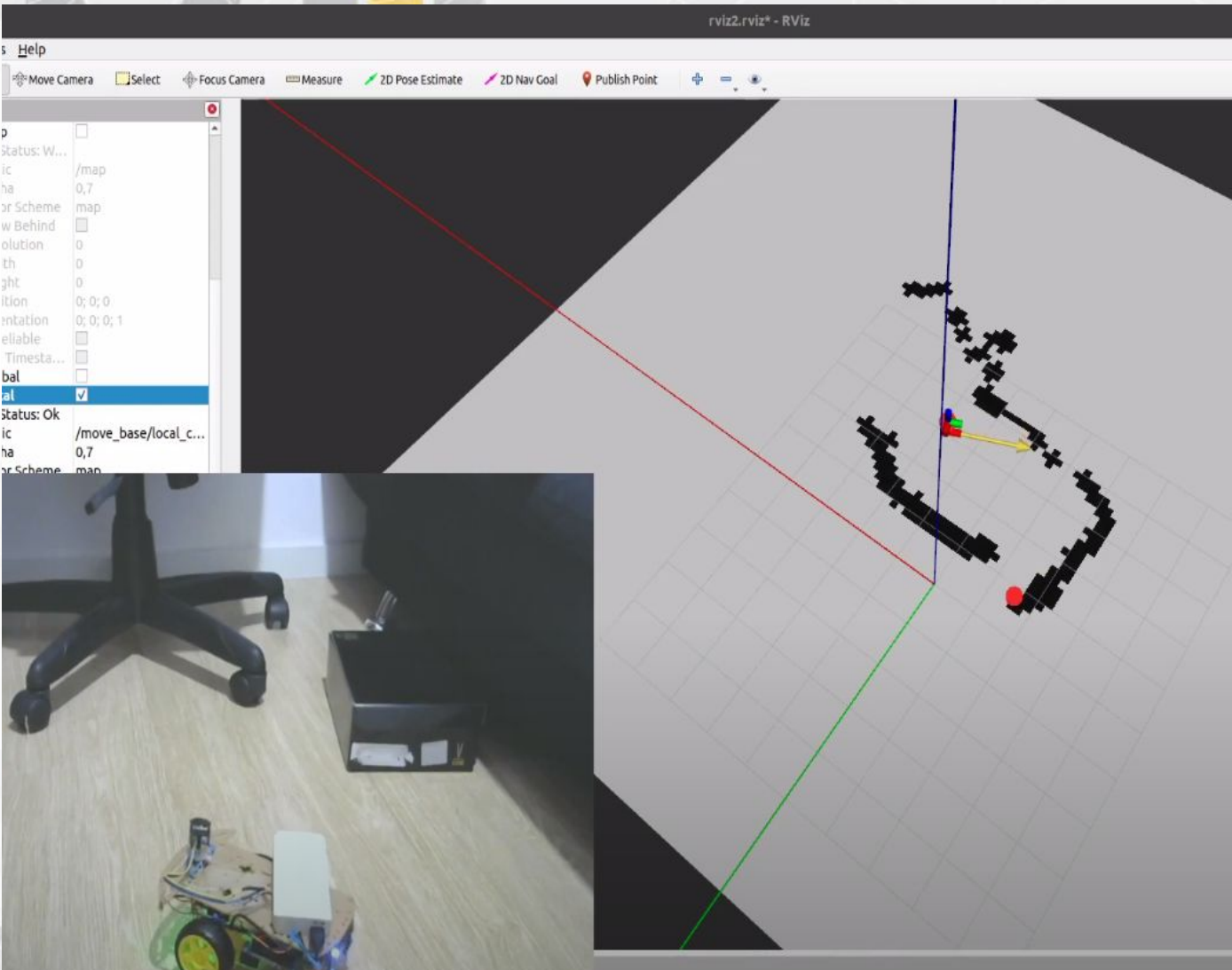
- Através dos *encoders* fixados perto do eixo dos motores conseguimos estimar a odométrica, e assim estimar a posição e ângulo do robô em relação ao eixo principal (Rviz).
- Assim visualizar a posição do robô no mapa em tempo real e com uma boa precisão, além de saber para onde ele esta “apontando”.



DEMONSTRAÇÃO

MAPEAMENTO

- O sensor laser VL53L0X a cada 0,5 segundos nos dá a distância do carrinho até o primeiro obstáculo encontrado. Isso acoplado em um servo motor que está girando 180° constantemente, nos permitiu gerar uma varredura de pontos.
- Assim o robô tem uma visão ampla para qualquer direção que ele vire
- Conseguimos então gerar um mapa com os pontos coletados, e visualizar de forma remota o que o robô enxerga. Completando o objetivo do projeto.



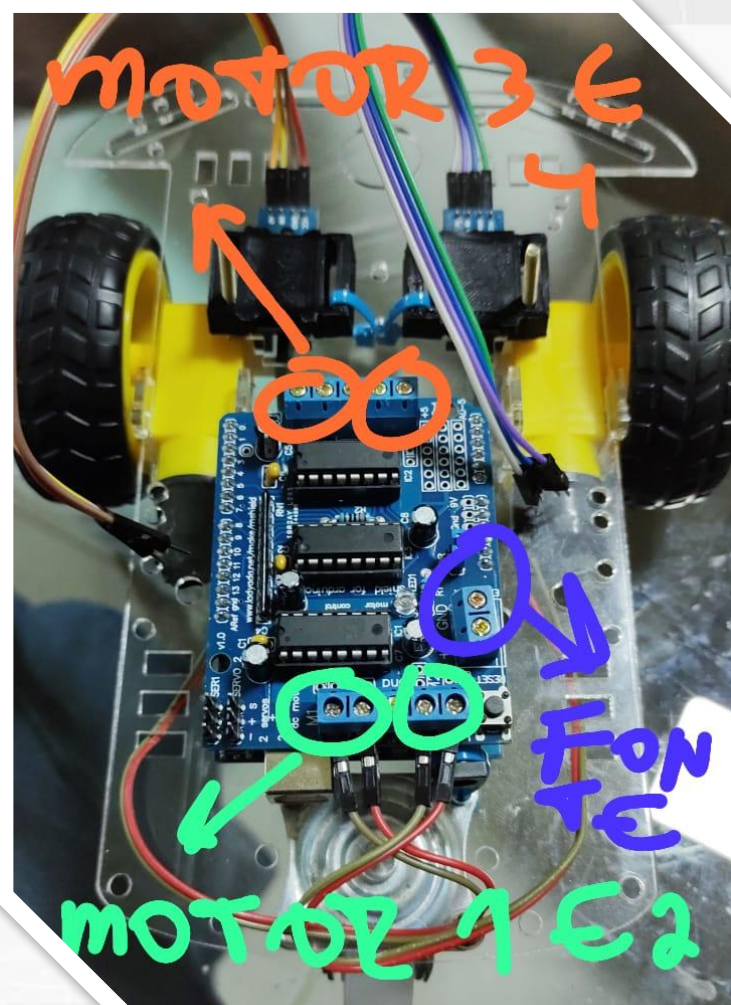
VÍDEO



DIFICULDADES DO PROJETO

MOTORES DE PLÁSTICO: Como estamos trabalhando com um projeto *Low Cost* utilizamos um dos motores mais baratos do mercado, que por sua vez são frágeis, e como sua caixa de redução também é de plástico isso o deixa com uma baixa precisão.

SOLDA DO SENSOR LASER: Por se tratar de uma parte móvel a solda e cabos de rede não foram uma boa opção, uma vez que, com o movimento incessante do servo a solda e os cabos se rompiam com frequência.



RODAS DE PLÁSTICO: As rodas de plástico eventualmente não atiram com o solo e giram em falso, prejudicando a odometria.

ALIMENTAÇÃO DO SISTEMA: Temos um Arduino, uma ponte H, dois motores DC, um servo motor, dois encoders e um ESP sendo alimentado somente por Power Bank, foi gasto bastante tempo até conseguirmos balancear tensão e corrente.

https://github.com/gustavoflw/turtlebot_oficina



OBRIGADO