ROS 2 EMBEDDED WORKING GROUP

Pet Your Own Bot!

Learn to build your pet robots with micro-ROS

KAIA.AI

ROS

NEXT WEEK

Embedded Working Group: 27th of February at 17:00 CET

Thank you!

A thank-you to the micro-ROS team

- including Pablo Garrido

A thank-you to the many ROS2 developers!

Micro-ROS Robot

100% ROS2, 100% open source

- Resembles Turtlebot3 Burger
  - LiDAR, 2-wheel differential drive
  - ROS2 mapping, navigation
- Simple, affordable
  - Runs on Arduino ESP32 (not Raspberry Pi)
  - 3D printed
- More information:
  - makerspet.com (diy hardware) + kaia.ai (software)
  - remake.ai (business)

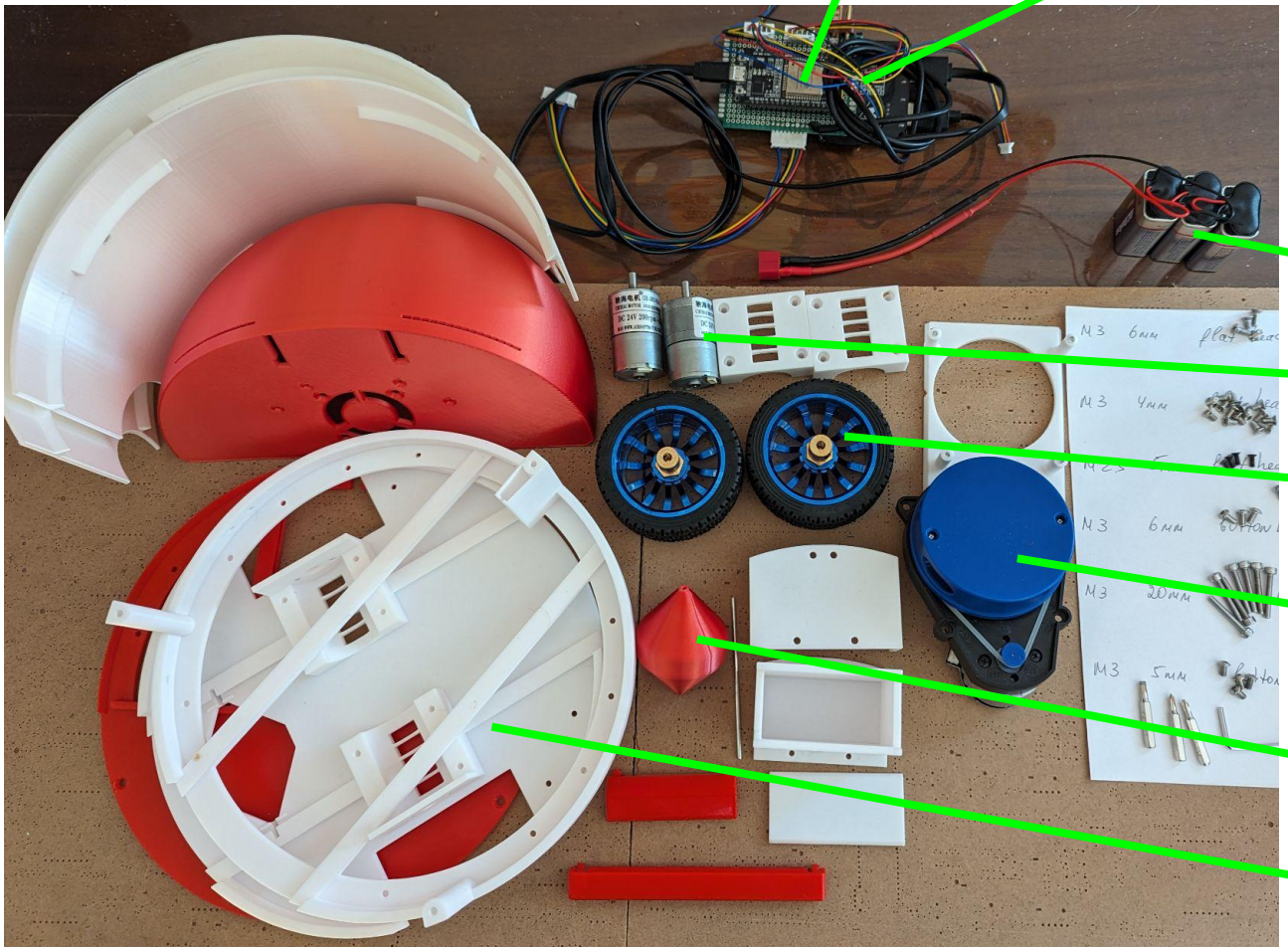
200mm model

300mm model

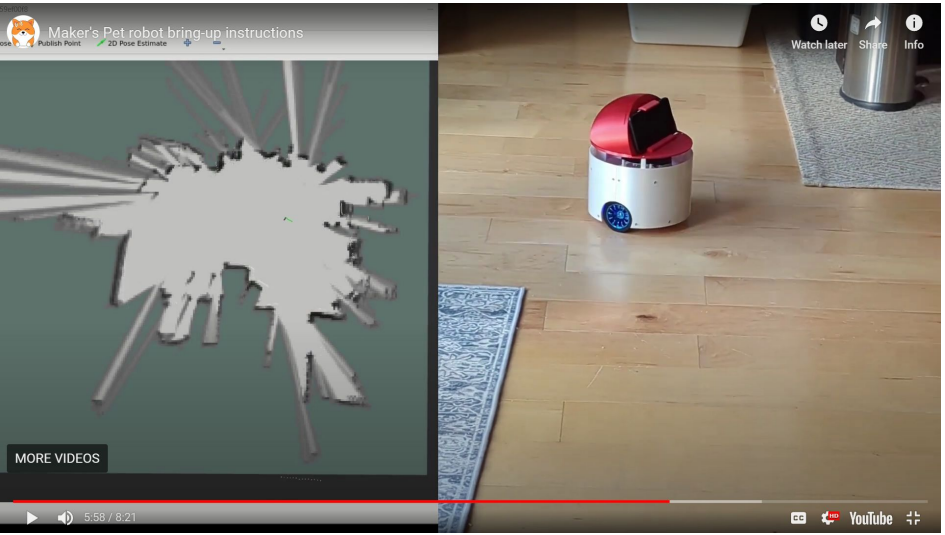
makerspet.com

kaia.ai

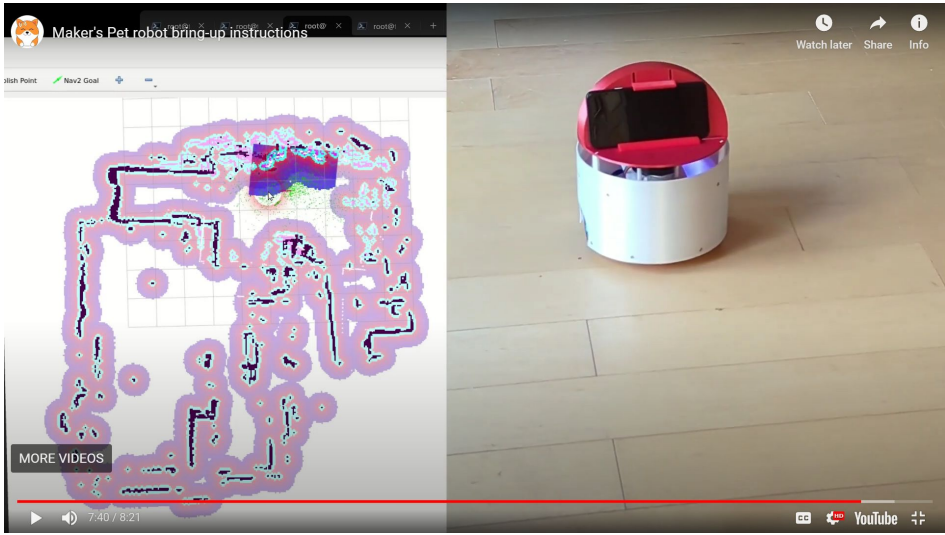
3D-printed DIY Kit



ROS2 Mapping, Navigation



Mapping (Nav2, Cartographer)



Navigation/SLAM (Nav2)

Bring-up instructions video
https://youtu.be/L_XbkA4pwRc

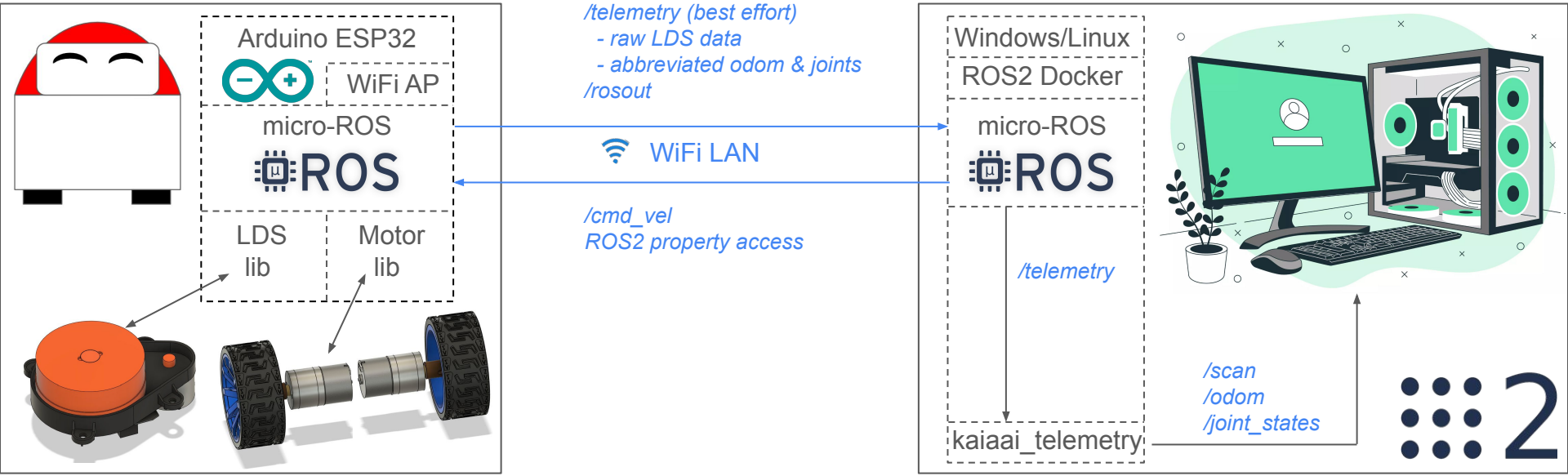
Open Source Links

3D printed body	https://github.com/makerspet/3d_models
Firmware	https://github.com/kaiaai/firmware
PC Docker image	docker pull kaiaai/kaiaai-ros-dev:humble (or :iron)

Instructions

Build, bring-up videos	https://bit.ly/3UTeLbc
Command reference	https://github.com/kaiaai/kaiaai

System Architecture as of 2/2024



Credits: image by storyset on Freepik

Hardware Overview

ESP32 micro-controller module ($3..$5)

- 2.4GHz WiFi, Bluetooth/LE
- 2 cores 240MHz XTensa 32-bit
- Single-precision float multiplier, no divider
- 520KB SRAM; TSMC 40nm

2x 24V BLDC motors w/encoder

- PWM speed, CW/CCW; PID control

24V Power Connector

- Alkaline; 6S1P LiPo optional

LiDAR/LDS connects to MCU serial

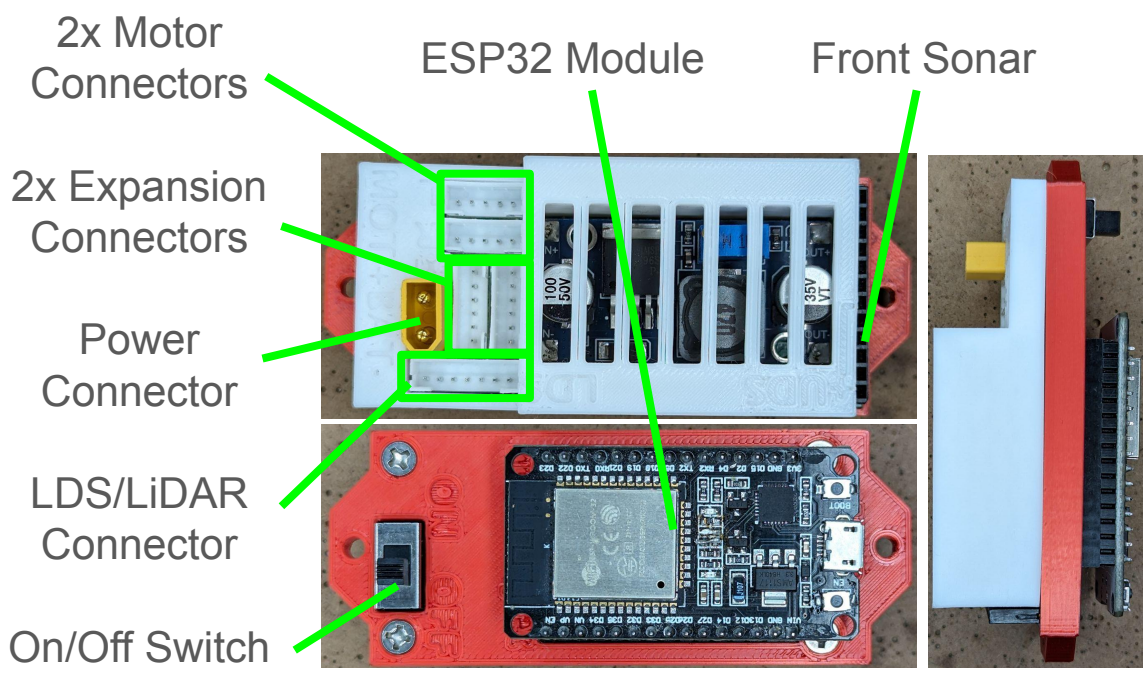
- There is no Raspberry Pi

2x Expansion Slots

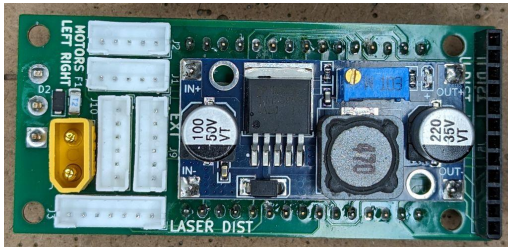
- 1x for the robot's Head
- 1x for the robot's Body

Front Sonar (optional)

- Detect obstacles "invisible" to LDS



ESP32 Breakout Board - Top



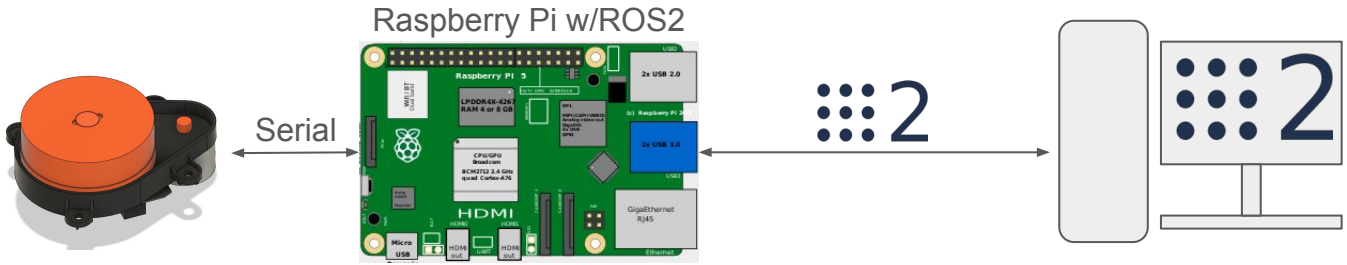
ESP32 Breakout Board - Top

LiDAR/LDS Use Case

Problem: Low-cost LDS OEM SDK/ROS2 nodes require PC (or Raspberry Pi) serial 😕

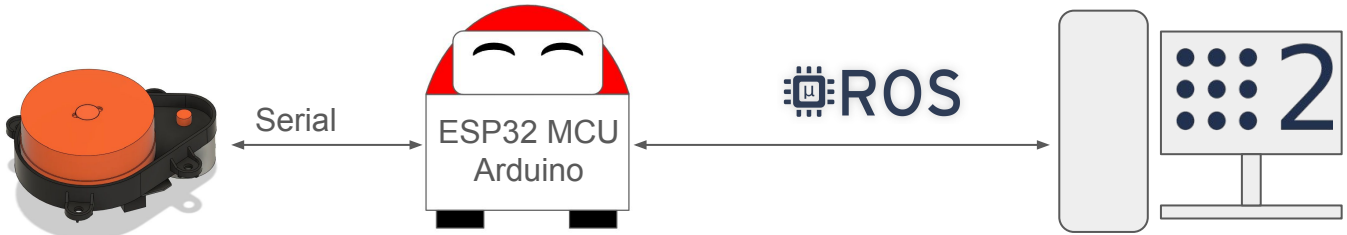
- A low-end LDS generates approx. ~6..16KB/sec for low-cost LDS
- Amount of data per sec ≅ (LDS sampling freq) * (3 bytes) * Overhead

Standard
(e.g. Turtlebot3)



vs.

Desired



Attribution https://commons.wikimedia.org/wiki/File:RaspberryPi_5B.svg

LiDAR/LDS Solution Considerations

Some low-end LDS

- are optimized for vacuum cleaner use
- lack motor control
- have TX only (no RX)
- require expensive calculations (tan, sin)
- lack SDK

* Lack built-in motor control

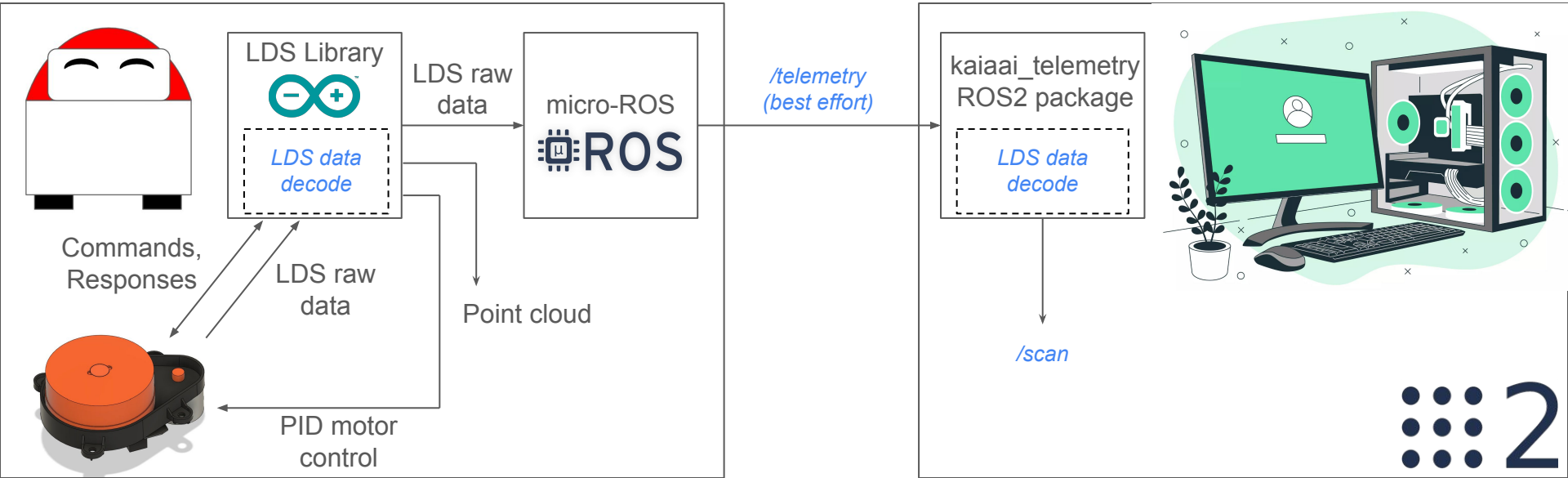
LiDAR/LDS Solution Considerations - contd.

- Serial “extender” solutions available
  - No “Serial over micro-ROS”
- Two types of LiDAR/LDS communications via same serial port
  - Configuration commands and responses - two-way over a reliable transport; lag-insensitive
  - Data - one-way, lag-sensitive stream; lossy transport OK
- Adding another transport in addition to micro-ROS not desirable
  - Websockets, socket.io, MQQT, vanilla UDP, etc.
- Virtual “loss-on/off” bi-directional serial port over micro-ROS? 🤔
  - Complicated
  - Non-lossy when SDK probes/configures/starts LDS; lossy when data streams one way
  - Allows reusing OEM ROS2 SDK 😊
  - Motor control should still run on Arduino (PID lag-sensitive loop)
- Are we are “stressing out” micro-ROS?
  - Is micro-ROS not meant to stream “larger” data like LDS?

LiDAR/LDS Current Solution

Send LDS data *raw, one-way, best effort* over *micro-ROS* WiFi UDP

- I.e. “one-way lossy serial”, encapsulated in micro-ROS messages
- LDS control (commands, motor control PID) runs on MCU only
- Not ideal, requires “porting” SDK 😕



kaiaai_telemetry Package

Receives

- raw LDS data
- abbreviated odom
- abbreviated joint_pos/vel

Calculates & Converts

- raw LDS to /scan
- abbreviated odom to /odom
- abbreviated joint_pos/vel to /joint_states

https://github.com/kaiaai/kaiaai_msgs/
msg/KaiaaiTelemetry.msg

```
builtin_interfaces/Time stamp
uint32 seq
float32 odom_pos_x
float32 odom_pos_y
float32 odom_pos_yaw
float32 odom_vel_x
float32 odom_vel_yaw
float32[] joint_pos
float32[] joint_vel
uint8[] lds
```

KaiaaiTelemetry message

Arduino ESP32 Firmware

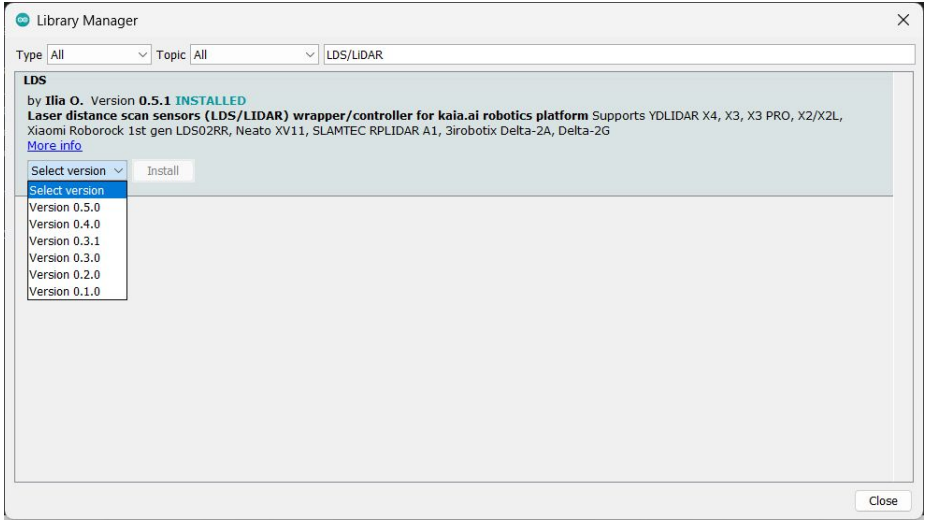
https://github.com/kaiaai/firmware

Required libraries

- micro-ROS
- LDS library (kaia.ai)
- Motor control library (kaia.ai)
  - PWM, PID control
  - Encoder tracking
- Web server for configuration, WiFi access

Miscellaneous Notes

- micro-ROS message size affect transport speed, reliability
  - Currently ~412 bytes (data fields total)
- Firmware parses /cmd_vel
  - calculates speed ramp



LDS Library (kaia.ai)

- Parse packets
  - Read scan frequency
  - Detect scan complete
  - Make raw packets available (to micro-ROS)
  - Signal (to micro-ROS) end-of-rotation, e.g. to flush LDS data to reduce lag
  - (Optional) Compute point cloud, e.g. for obstacle avoidance
- PID control of LDS motor

micro-ROS for Arduino - Customized

https://github.com/kaiaai/micro_ros_arduino_kaiaai/

A micro-ROS/micro_ros_arduino fork

- Apache 2.0 license (same as original repo)
- Published, maintained as Arduino library

Configuration & tweaks

- Added kaiaai_messages to build
- git-clone, build works on Windows
- +5 max services for param server
  - Robot node has ROS parameters
- Took WiFi.begin() out of set_microros_wifi_transports()
- Miscellaneous tweaks

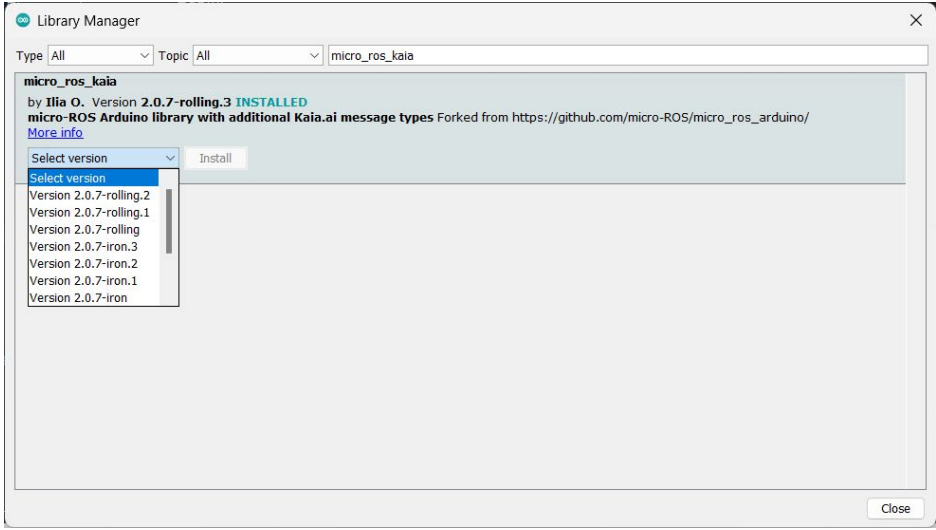


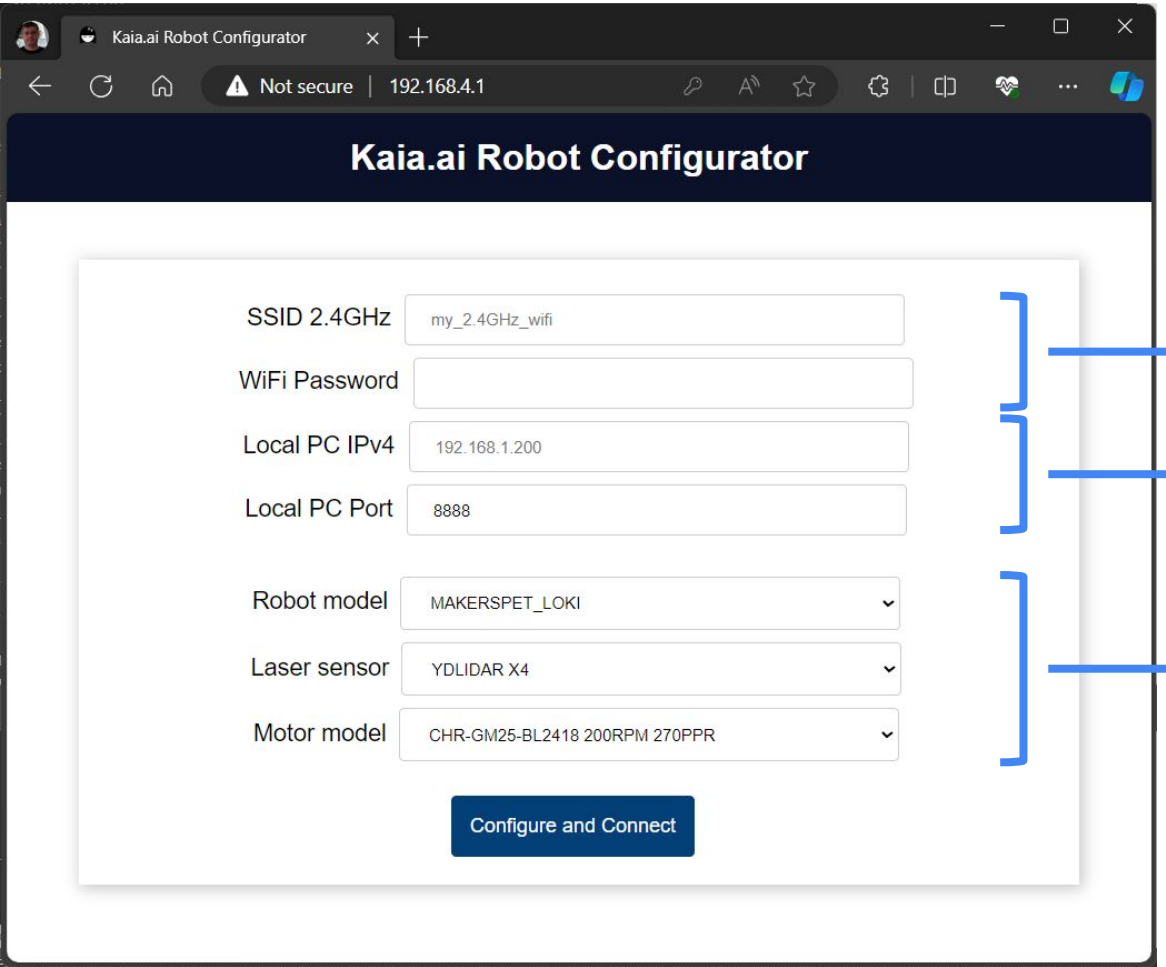
Fig. Arduino Library Manager window showing micro_ros_kaia library

Robot Configuration as of 2/2024

When robot is *unable to connect to WiFi*, robot enters the *configuration mode*.

I.e. robot sets up:

1. A temporary Access Point
2. HTTP server at 192.168.4.1

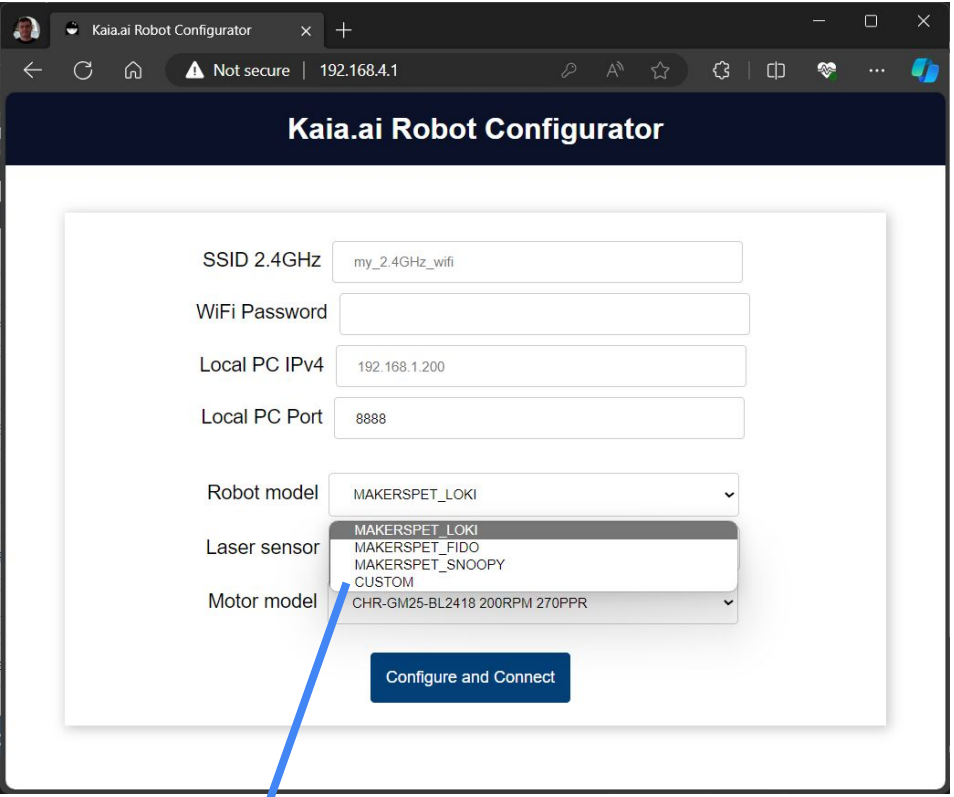


WiFi LAN access

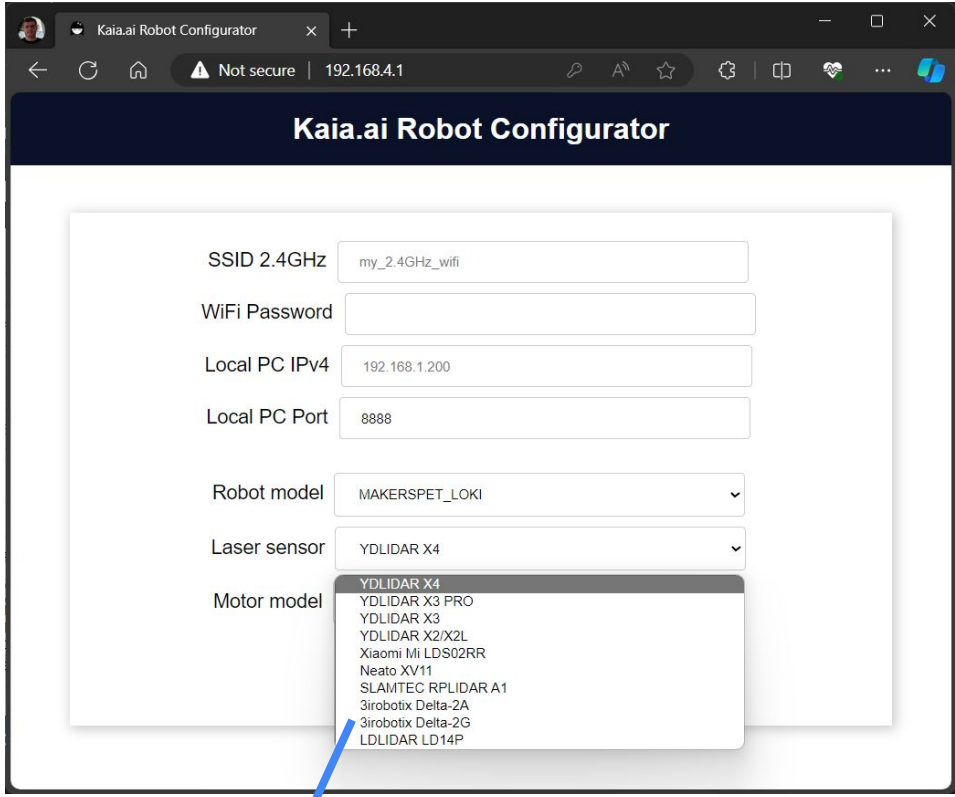
micro-ROS agent

Robot hardware options

Robot Configuration - cont.

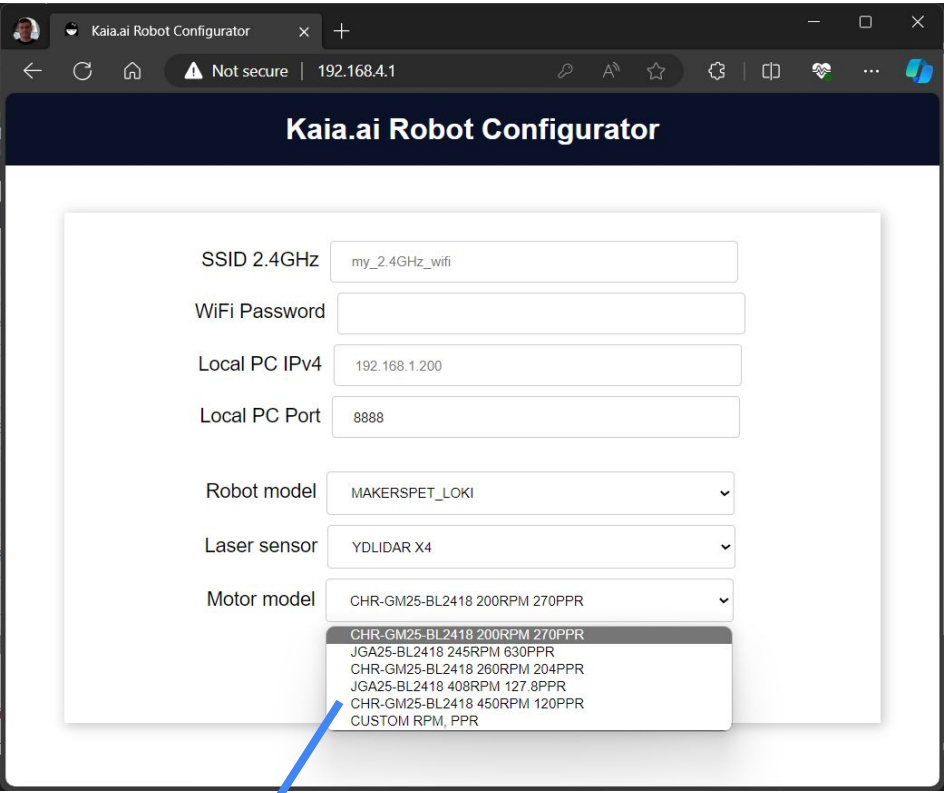


Robot base size options

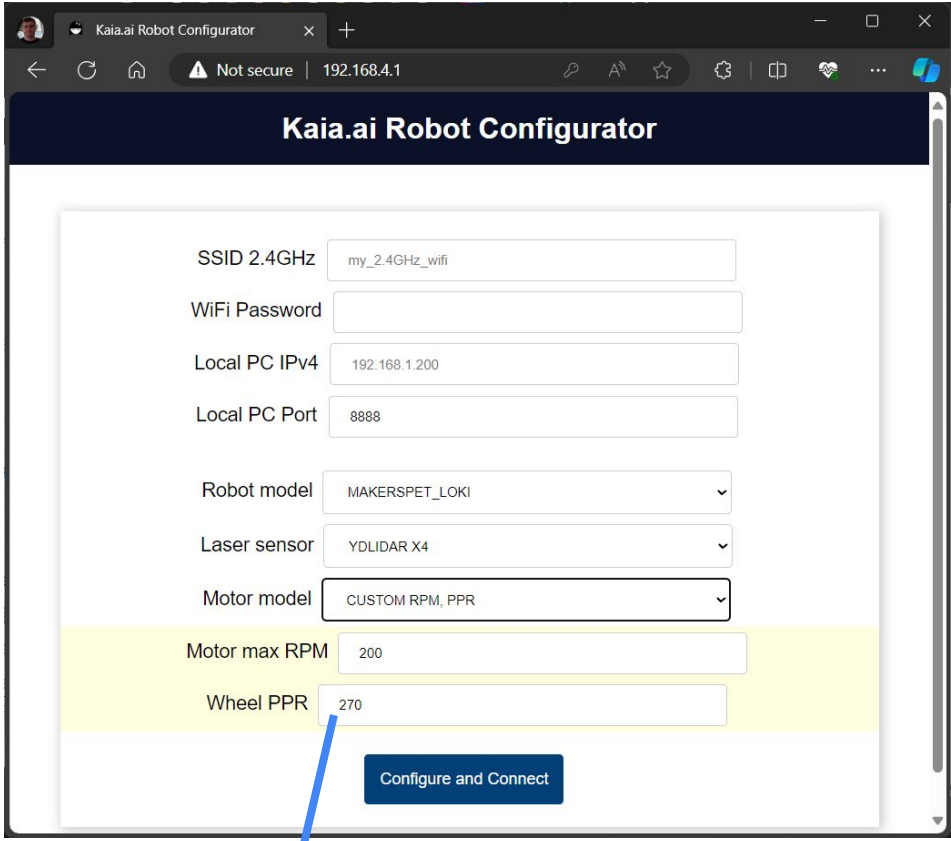


Robot LiDAR/LDS model options

Robot Configuration - cont.

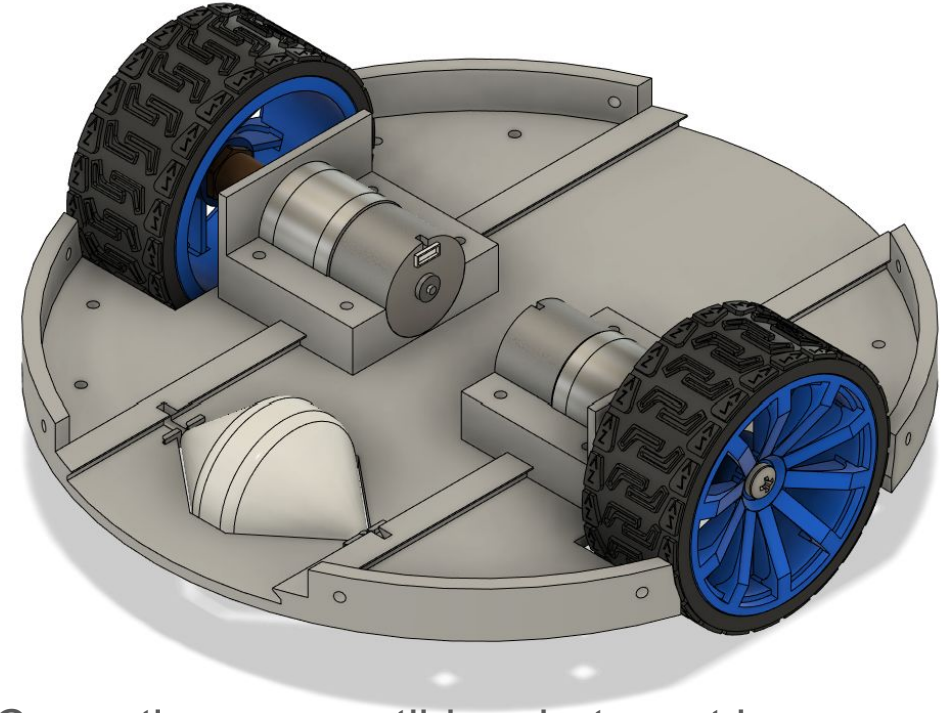


Robot motor model options



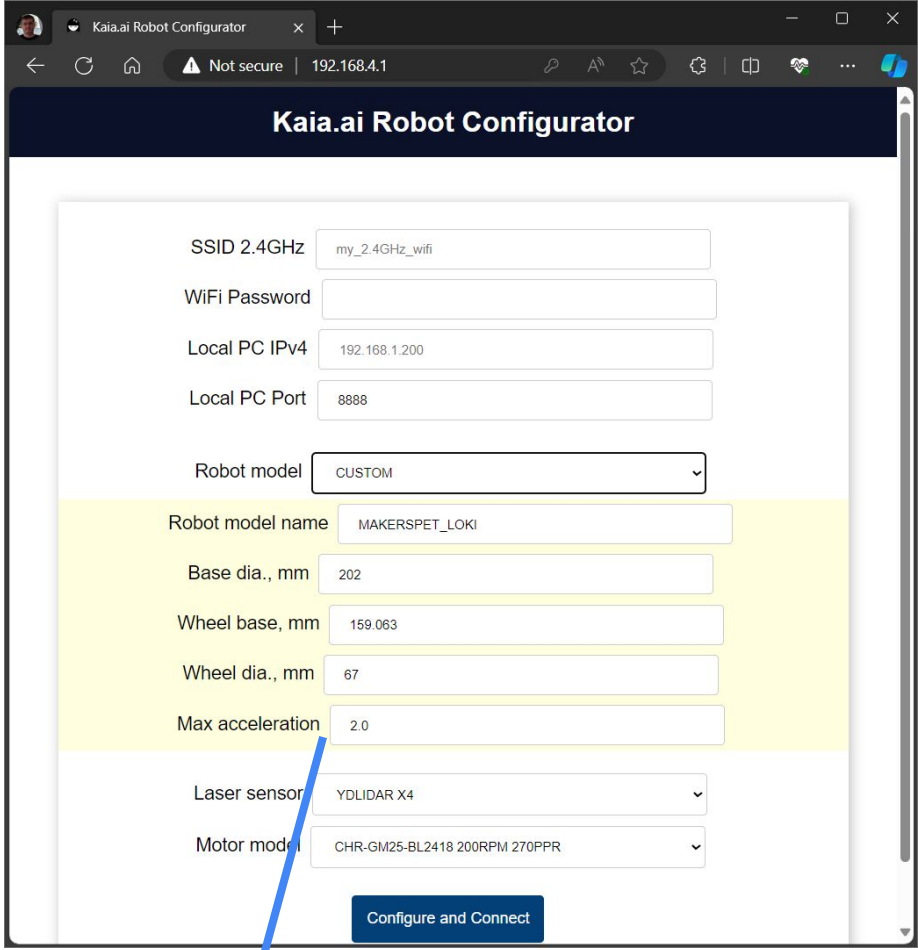
Custom motor parameters

Robot Configuration - cont.



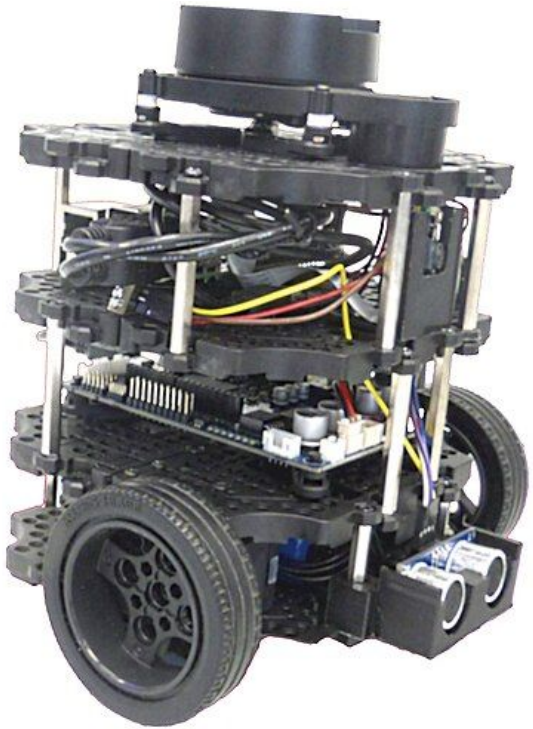
Currently, a compatible robot must have

- a round base
- two-wheel differential drive
- LDS/LiDAR center above base centerpoint



Custom base parameters

Robot Economics and micro-ROS



Turtlebot3 Burger		Maker's Pet
Raspberry Pi 4	->	ESP32
OpenCR 1.0 board	->	Breakout board
Dynamixel XL430-W250 smart actuators	->	~7x cheaper BLDC motors
2x 65mm wheels	->	No change
LiPo battery	->	Alkaline battery ESP32 is low-power
Battery charger	->	None
LDS HLS-LFCD2/3	->	<$20 LDS
Color camera	->	No cost
Robotis US $659.30		Total <$100

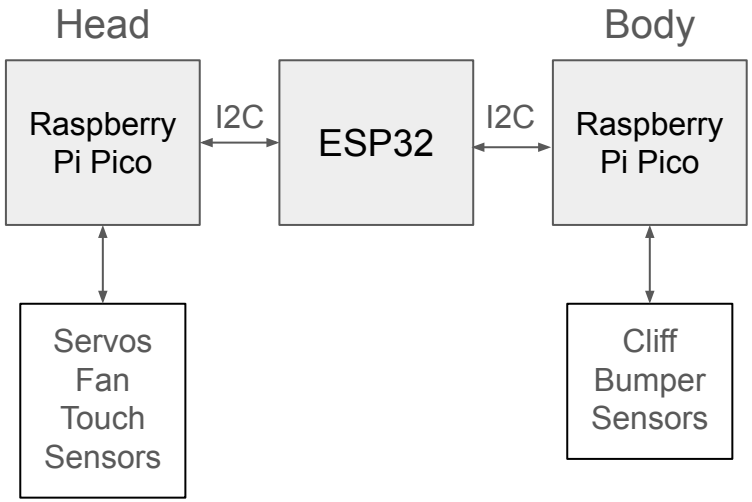
- micro-ROS slashes parts cost for certain robot types
- Proliferation of robotic vacuum cleaners helps slash LDS costs
- Replacing Raspberry Pi with MCU simplifies system (no RPi setup, SSH/VNC, display, etc.)

Image credits https://commons.wikimedia.org/wiki/File:TurtleBot3_Burger.jpg

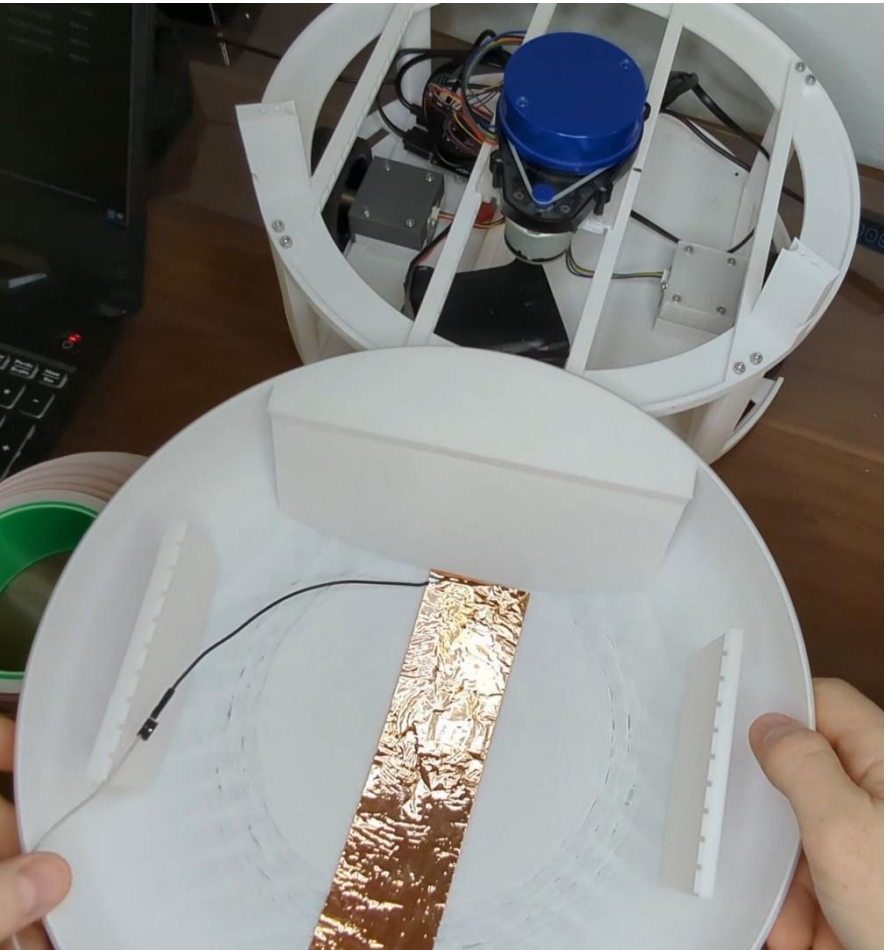
Hardware Extensions

2x ESP32 I2C extensions slots

- Head
- Body

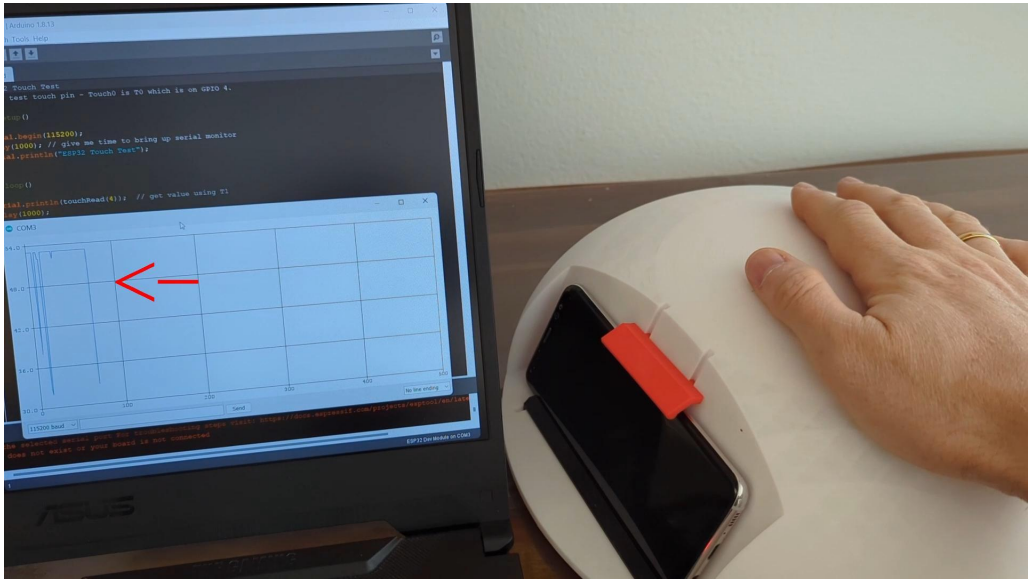


Adding Sense of Touch



- Place sticky copper tape from the inside
- Connect to MCU GPIO
- Place your hand, tape capacitance increases
- MCU ADC measures capacitance (charge/discharge time)

https://github.com/kaiaai/pico_touch_pio
YouTube https://youtu.be/oTpgZuBU10Q



Smartphone/Tablet

Bring-your-own (old) smartphone (or 8” tablet) $0

- Connects to ROS2 PC
  - display
  - color camera feed
  - mic, speakers audio
  - IMU

Display interface runs in a browser

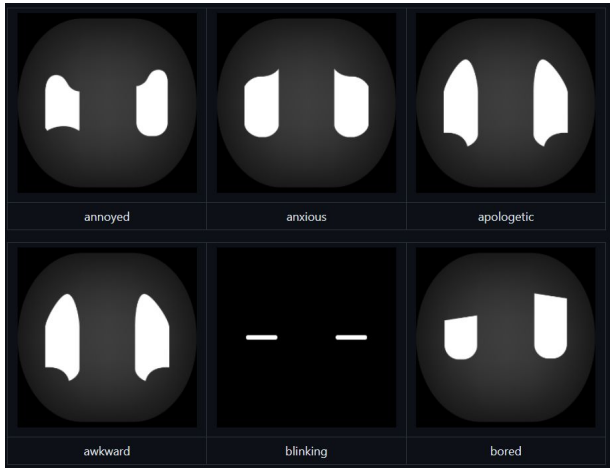
- Websockets transport using RobotWebTools rosbridge_suite

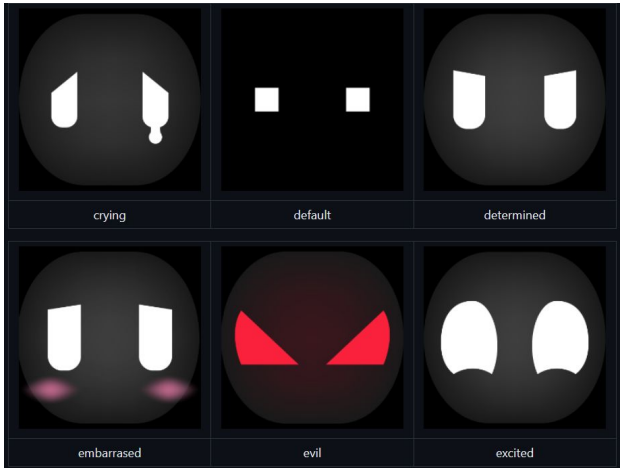
TBD moving ROS2 from PC to smartphone in the long-term

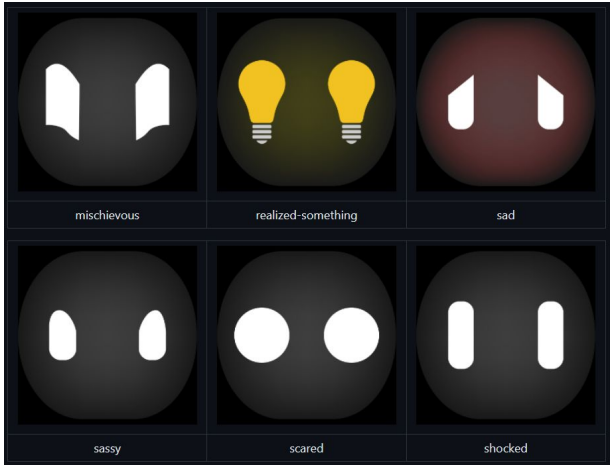
- We’ve built ROS2 on Android smartphone using Termux (no root)

Facial Expressions

https://github.com/kaiaai/kaia-face.js







Development Environment

Docker

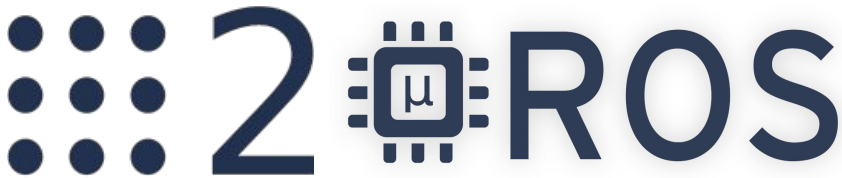
- ROS2 desktop (RViz, Gazebo, etc.)
- micro-ROS
- Windows (Linux OK)

https://hub.docker.com/r/kaiaai/kaiaai-ros-dev

- docker pull kaiaai/kaiaai-ros-dev:humble
- docker pull kaiaai/kaiaai-ros-dev:iron

Dockerfile https://github.com/kaiaai/docker

- FROM osrf/ros:$distro_tag-desktop
- Installs micro-ROS base, agent
- Kaia.ai-specific packages



PC setup requires

- Docker
- XLaunch (for Windows)
- Arduino IDE

Kaia.ai Robot Software Platform

A software platform for multi-skilled home robots.

- Goal: wide consumer acceptance of multi-skilled home robots

Teaser video https://youtu.be/5xbXqkkQknk

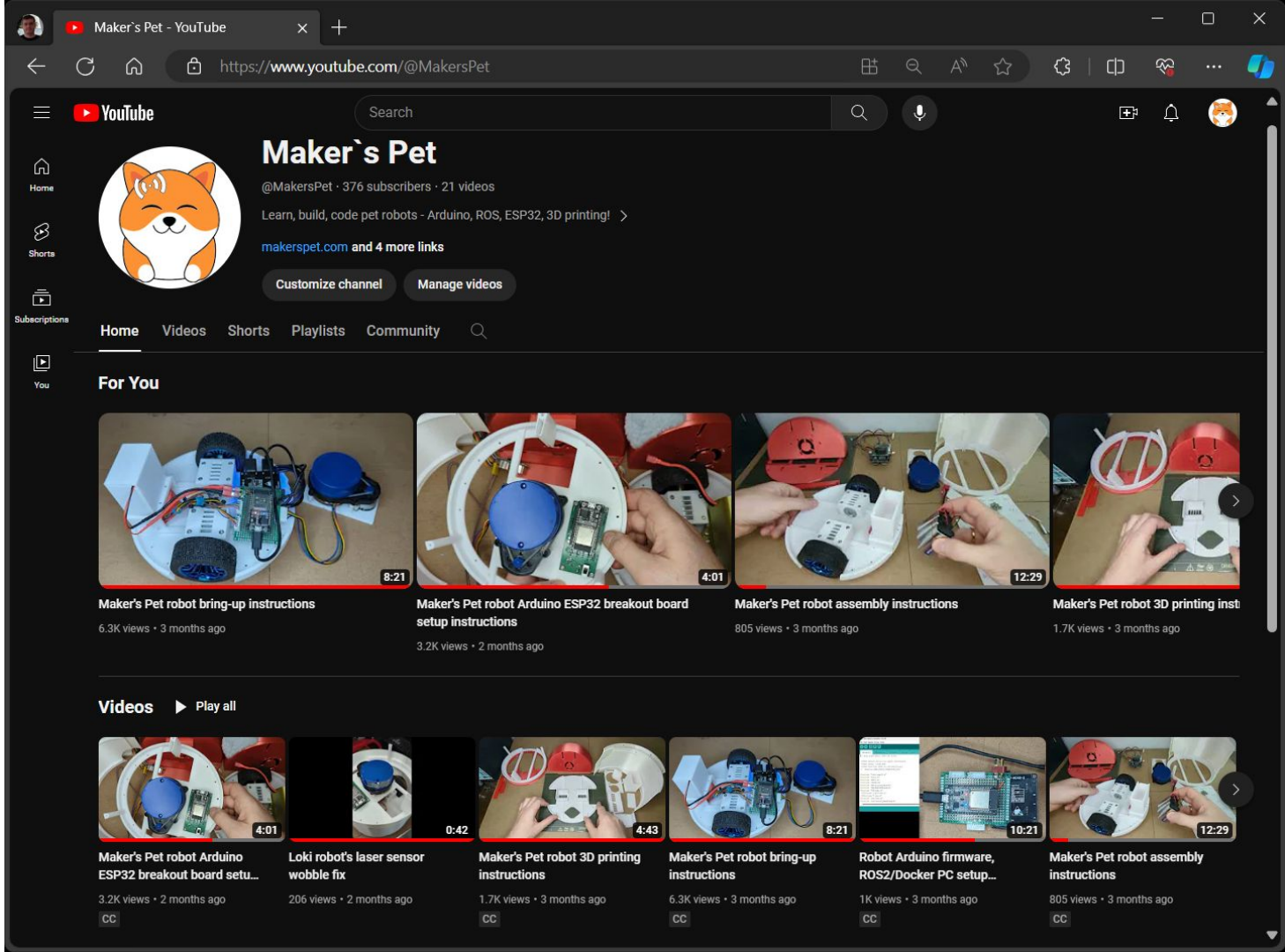
- On top of ROS, ML/AI
- First pre-release around end of 2024
- In stealth mode until then 🕛

YouTube

@makerspet DIY hardware

@kaiaai software

@remakeai startup



Look us up

GitHub

- https://github.com/makerspet
- https://github.com/kaiaai

Web/Blog

- https://makerspet.com DIY hardware
- https://kaia.ai Software
- https://remake.ai Startup

Facebook

- Group https://www.facebook.com/groups/243730868651472/
- Makerspet https://www.facebook.com/profile.php?id=61550661555594
- Kaia.ai https://www.facebook.com/kaiaai/

Launch Mailing List

- http://eepurl.com/gBV7tb

Twitter

- https://twitter.com/makerspet/
- https://twitter.com/RemakeAi
- https://www.instagram.com/remake_ai

Instagram

- https://www.instagram.com/makerspet_com/
- https://www.instagram.com/kaia__ai

Reddit

- https://www.reddit.com/r/3dPrintedHomeRobots/