ROS in Motion: Autonomous Robots and UAV Systems

Prof. Venki Muthukumar

Course Description

Design, implementation and programming of autonomous mobile robots (UAVs* and Rovers), kinematics and dynamics of robots, basic control theory, sensors and actuators for robots, autopilots and autonomous control, and robot application development.

COREQUISITES AND PREREQUISITES

- C++ and Python
- Intermediate understanding of microcontrollers

How is this course different?

- How do Middle School/High School students learn robotics?
  - First Robotics
- How ECE students learn robotics?
  - RPI 4 – Python - https://bityl.co/KlCW
  - ESP32 – Arduino/Blocky - https://bityl.co/KlCN
  - RPI Pico – MicroPython - https://bityl.co/KlCD
- How current CS students or online robot hobbyist learn robotics?
  - **Construct** is an e-learning platform for **ROS** and Robotics
    - https://www.theconstructsim.com/
- How we learn robotics?
  - A complete SW (ROS) + HW interface approach

Textbooks & Classroom Info

- Textbooks:
  - ROS
    - A Systematic Approach to Learning Robot Programming with ROS, Wyatt Newman
  - Robotics
    - Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems, Gregor Klancar , Andrej Zdesar , Saso Blazic.
    - Introduction to Autonomous Robots, Nikolaus Correll, ISBN-13: 978-0692700877 - Freely available online
    - Probabilistic Robotics, Dieter Fox, Sebastian Thrun, and Wolfram Burgard
- Class Schedule: Weekends/TBD
- Location: Virtual - Zoom

Course Topics

- **THEORY**
- Mathematical Fundamental
- Overview of Mobile Robots,
- Kinetics and Dynamics
- Controls
- Sensors and Odometry
- Localization
- SLAM
- Navigation

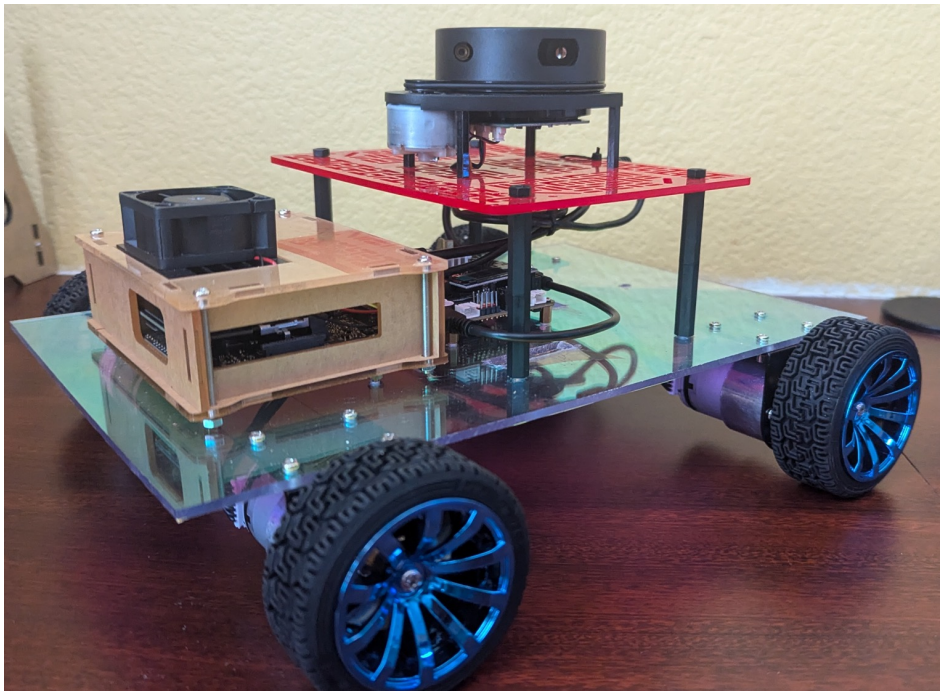
- **HANDS-ON**
- Robotics Operating Systems (ROS).
  - SW Simulation of Robots using python/C++.
- Python Simulations
  - PythonRobotics
    - https://atsushisakai.github.io/PythonRobotics/
  - Modeling of MR, …
- DD Mobile Robot Programming *
- STM32IDE - HW Programming of Robot using Arduino Programming *

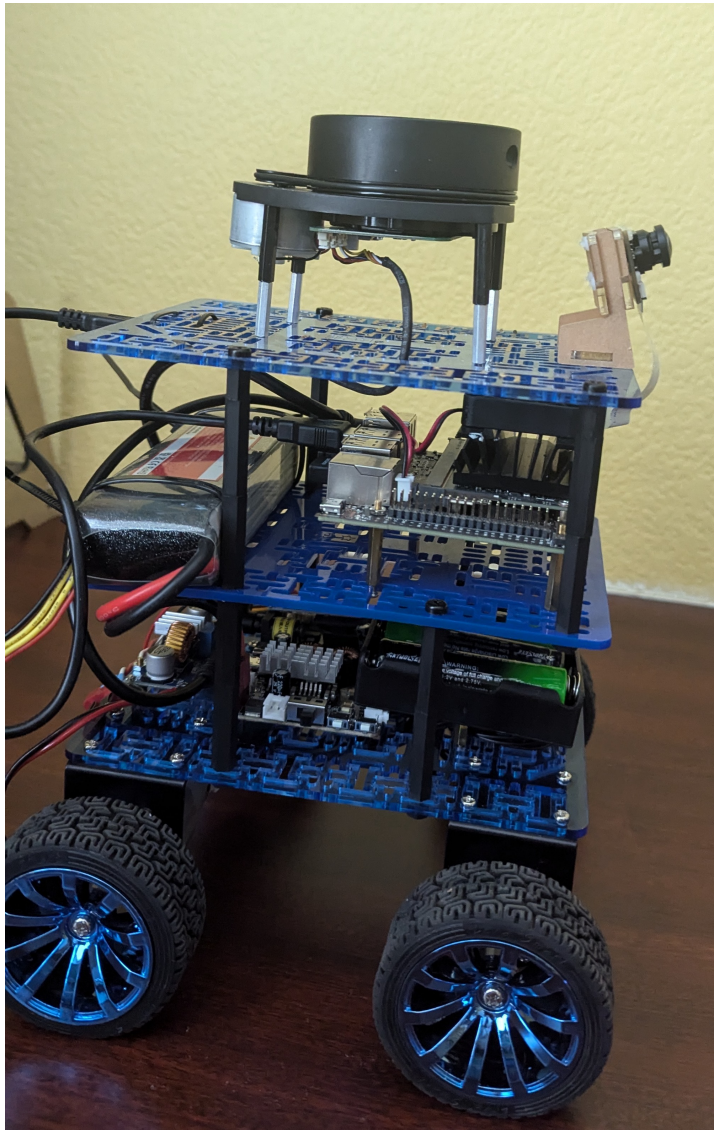
* - Demo only

ROS IN MOTION

Our Robot (Not really)

Our Robots (actual)

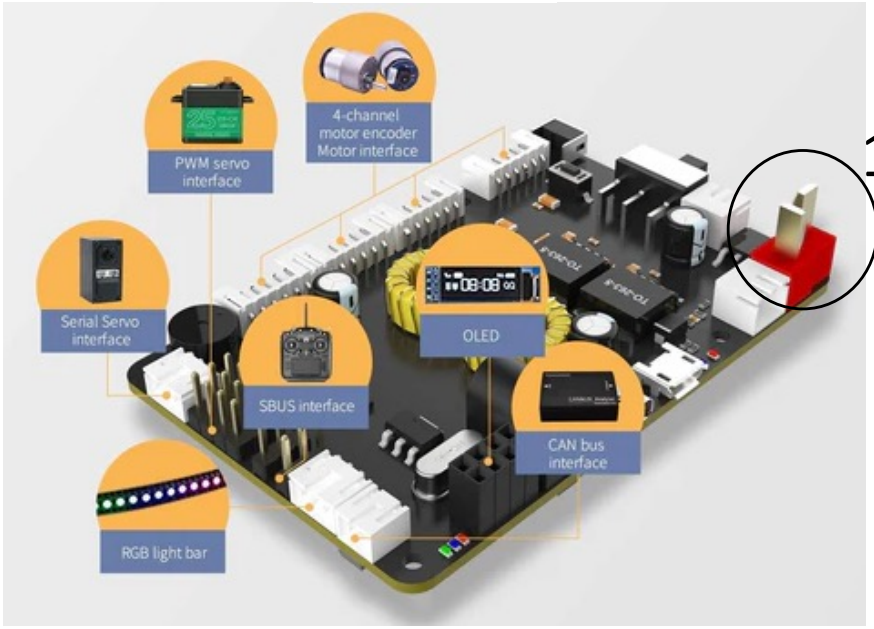




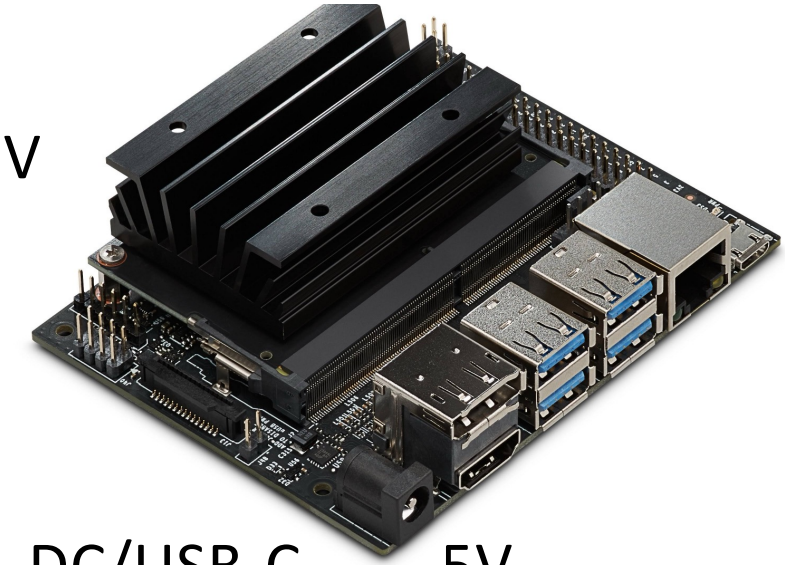
Mobile Wheel Robot Components

12V  X 4

T-plug

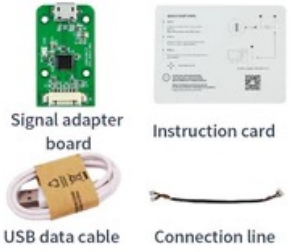


12V



DC/USB-C 5V

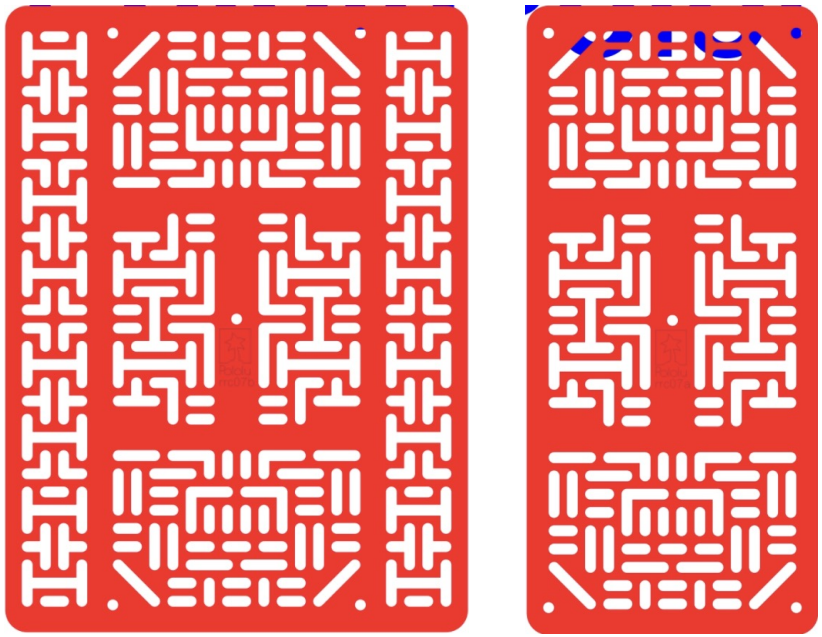
RPLIDAR A1



ROS IN MOTION

Chassis

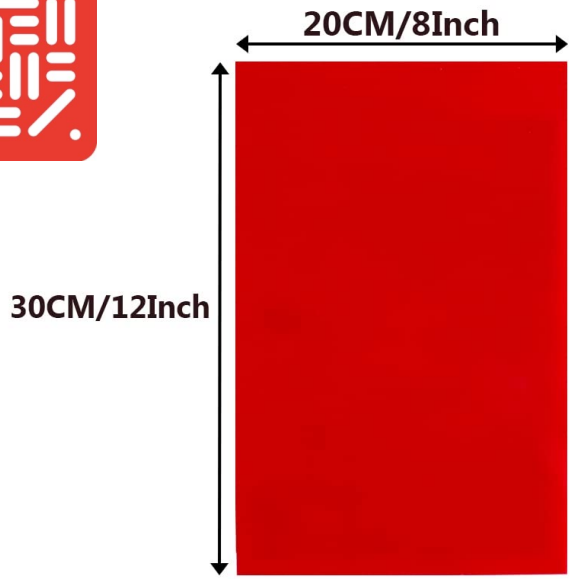
X 1



X 4

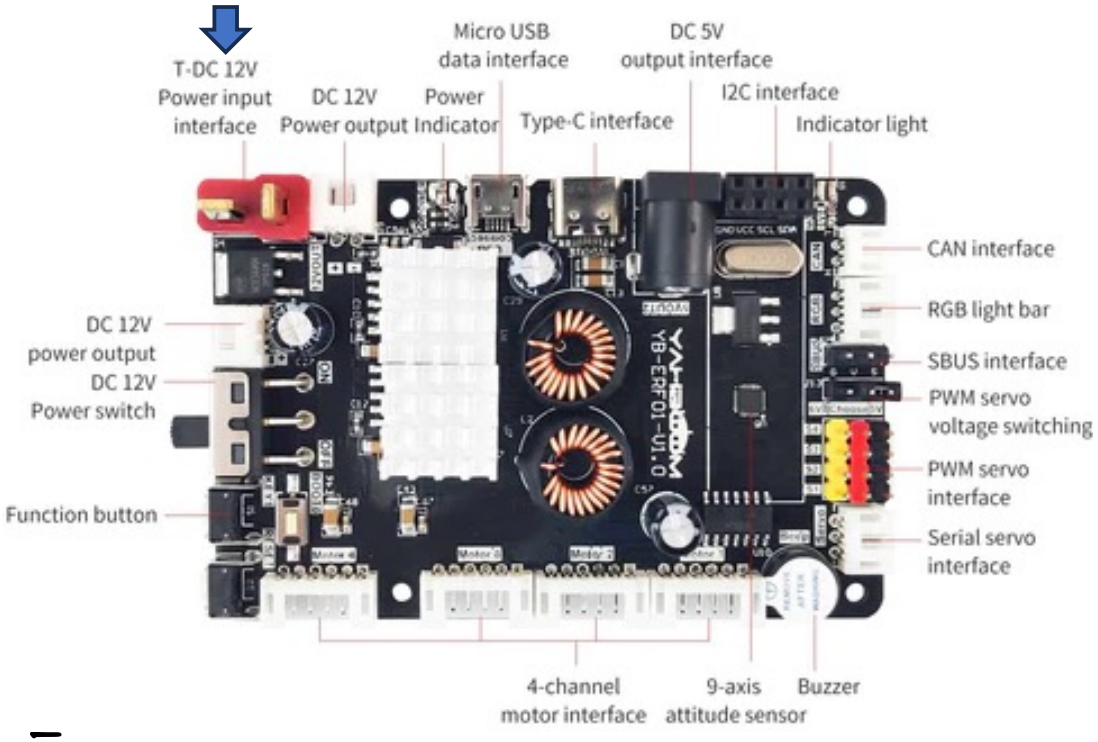
r geared motor + motor bracket
+ coupling + rubber tire

12V motor



X 1

ROS Controller



Basic parameters	
On-board MCU model	STM32F103RCT6
Communication baud rate	115200bps
Data interface	micro USB interface
USB to serial port chip	CH340 chip
Data output method	micro USB data interface
Command control	Support serial commands: whistle, light, drive motor, etc.
IMU chip	MPU9250 9-axis attitude sensor
Motor drive model	AM2857 driver chip*4
Motor with encode	Support 4-channel 12V encoder motor
SBUS model aircraft remote control	Support
CAN communication	Support
Other peripheral interfaces	Support PWM servo, serial servo, RGB light bar, OLED display, buzzer
Button	RESET key, KEY1 key, BOOT0 key
Firmware update method	microUSB data interface + MCUISP tool to update firmware
Operating voltage	T-type DC12V input
Stand-by current	About 50mA
Voltage output	DC 12V output interface*2, DC5V output interface*1,
	Type-C 5V output interface*1
Protection circuit	Servo over-current protection, anti-reverse connection protection
Operating temperature	-40°C~+85°C
Product size	85mm*56mm
Product weight	About 46g

Our OLD Mobile Wheeled Robot – F21

- The physical mobile robot will be built and tested by each students
- Robot Chassis: Romi Chassis Kit
- Motor Controller: Romi 32U4 Control Board
- Encoders: Romi Encoder Pair Kit, 12 CPR, 3.5-18V
- Motor Drivers: Builtin with Romi 32U4 Control Board
- IMU: Builtin with Romi 32U4 Control Board + LSM303D or MinIMU-9
- High-level Processing Engine/Controller: Rpi 3/4
- LIDAR Scanner: RPlidar A1 Scanners, YDLIDAR
- RGB Camera: Rpi Camera

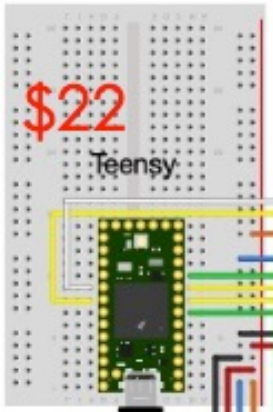
Our OLD Mobile Robot – F22

DWM1001-DEV

LIDAR

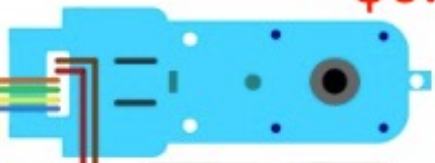
$22

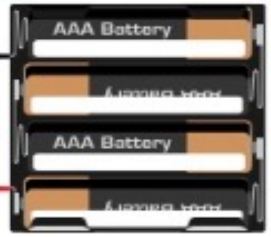
Teensy



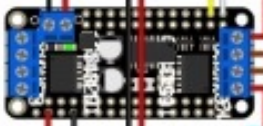
- G (Blue): hall power negative
- H1 (Green): hall H1 output signal, square wave
- H2 (Yellow): hall H2 output signal, square wave
- V (Orange): hall power positive
- M+ (Red): motor positive pole
- M- (Brown): motor negative pole

$6.50

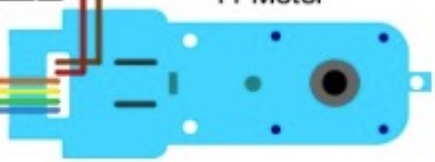




TT-Motor



Motor Driver

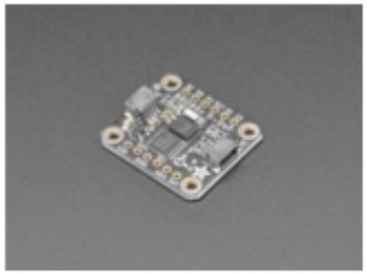


$21.50

$59

NVIDIA Jetson Nano 4G

$20



BNO055

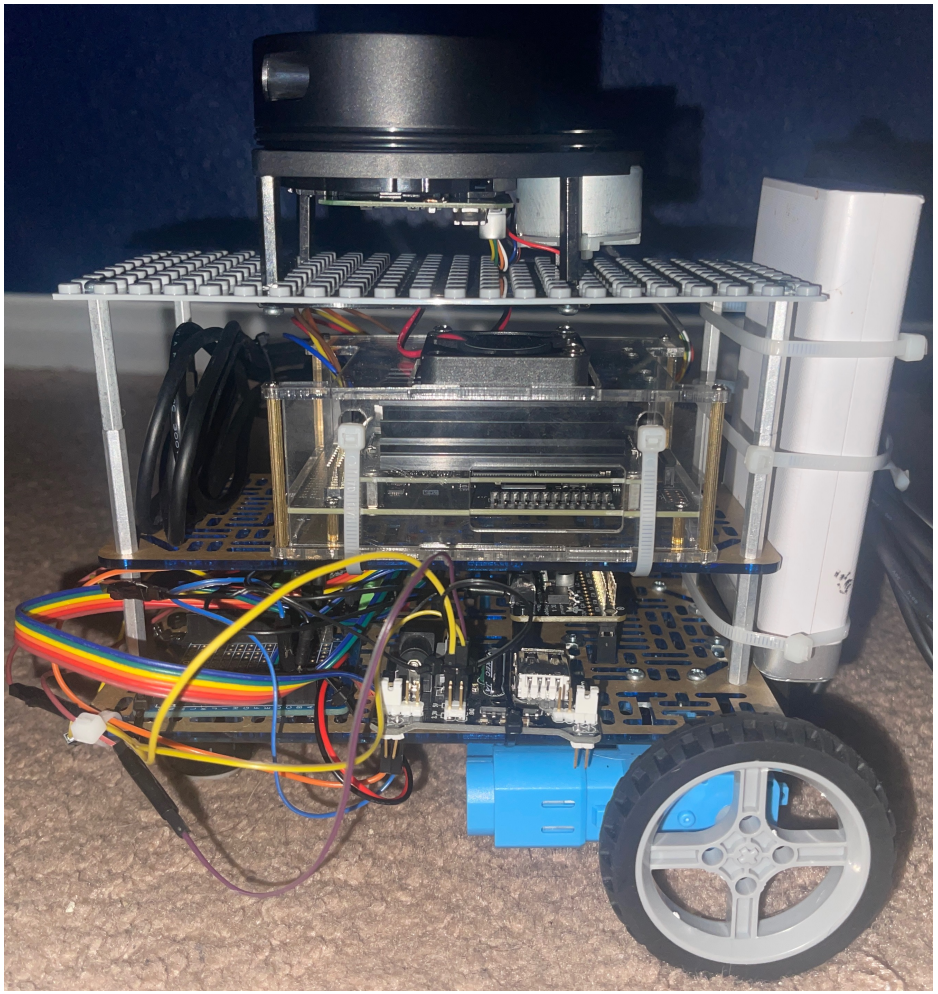
Battery Pack

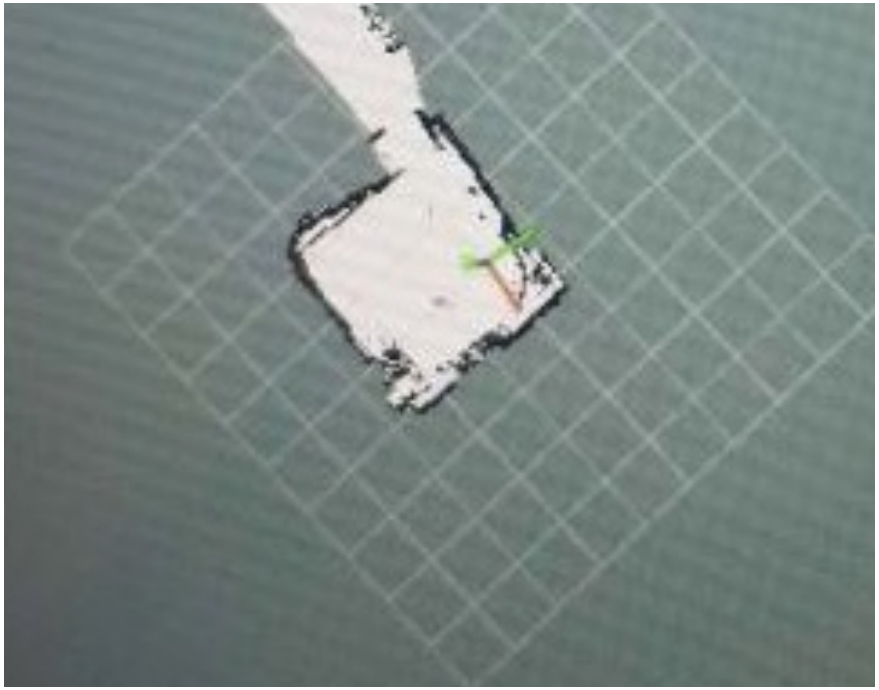
$29

Jetson Camera

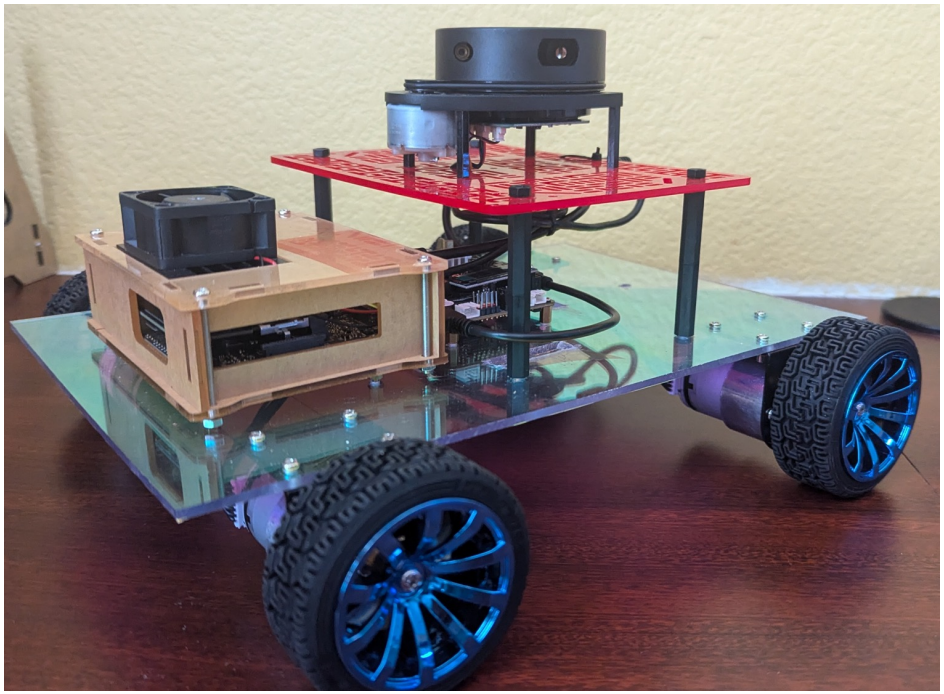
ROS IN MOTION

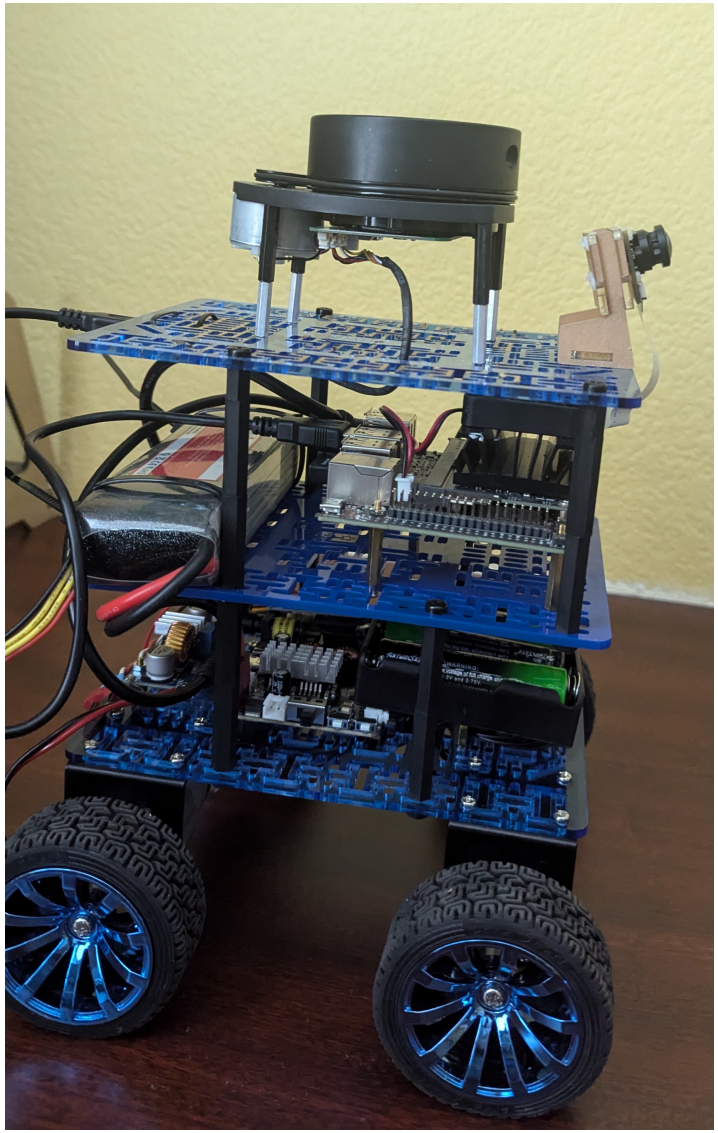
Our OLD Mobile Wheeled Robot – F22





Our Robots – F24



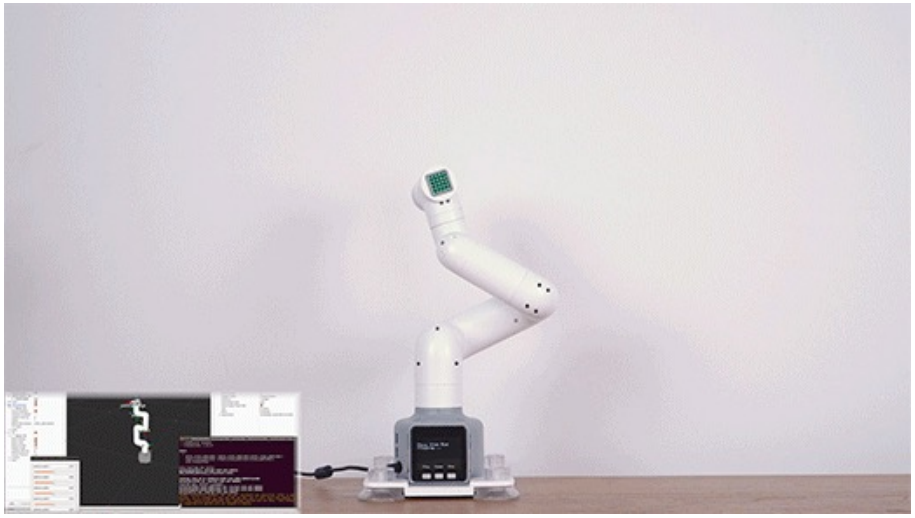


Our Research Robot #3

Rover Robotics (ROS/2 supported)

Quanergy M8 LIDAR _+
CORTEX PC

Elephant Robotics ARM (ROS supported)



The main goal of this course …

- Understand the mathematical fundamentals of motion of mobile robots (Differential Drive Kinematics) and UAVs
- Understand the problem and solutions for localization
- Understand mapping and path planning
- Construct a mobile robot with the following **components**:
  - Mobile robot base with a robotic controller (STM32 based)
  - Interface to encoders and IMU sensors
  - Interface LIDAR
  - High-level Processing Engine/Controller (Jetson Nano (JN) 2GB/4G Developer Kit)
  - USB JN Camera for image processing
- Understand build, configuration, and test flying of a drone/UAV.

Contd …

- **Software** framework used:
  - Mobile robot base running on C++ based software (STM32CubeIDE/Keil uVision – ARM MDK)
  - Mobile processing engine running Ubuntu + C++/Python – ROS (master)
  - Host system running Ubuntu and ROS (slave)

- **Outcome:** Your mobile robot should perform the following functions (simulation only*)
  - Should be controlled by the host system using teleop commands, and PS4 controller.
  - Should be able to detect or avoid obstacles using image processing.
  - Construct a map using LIDAR + Depthsense Camera (?)
  - Navigate autonomously from point A to B.

Hands-on Learning Modes

- We follow three hands-on learning modes in this course
  - Understand mobile robot motion using simple python simulations – **Python** (minimal)
  - Understand and Simulate robot motion, sensors, modeling, localization and mapping using ROS, Gazebo and RVIZ environments – **ROS Simulation**
  - Understand physical robot using C++ (Controller) and Python/C++ - ROS environment to perform autonomous task – **ROS Hardware Interface**

Why and How to Learn ROS?

- Robot Operating System is robotics middleware. Allows us to perform modeling, simulation (SITL and HIL Simulations)
- Understanding ROS includes:
  - ROS Basics and Foundation: ROS topics, Services, Messages, Nodes, ...
  - Motion in ROS
  - Modeling Robots
  - Modeling Environments
  - Generating Sensors
  - Localization,
  - Navigation and
  - SLAM
- ROS Tutorials: ros.org
- Reference books:
  - A Systematic Approach to Learning Robot Programming with ROS, Wyatt Newman
- **Should I learn ROS2?**

Software + Hardware

- **Computer Usage**
  - C programming, Python, Embedded Linux/Linux
- **Simulators/Software:**
  - Python
  - Firmware (STM32 uC)
  - ROS and Gazebo
    - Python/C++ Programming
- **Hardware:**
  - STM32 ROS Controller
- **Host Development System**
  - Ubuntu 20.04 (Noetic) running on parallel/Dual Boot.

Contact

- Instructor
  - Venki Muthukumar
- Instruction:
  - Website: Github
  - Lectures: Virtual - Zoom
  - Labs: Videos/In-person
- Class Times:
    - In-person
- Contact
  - Email: vm at unlv.nevada.edu
  - Chat/Discussion: Slack (TBA)

Grading Scheme

Grading (subject to change):

- Reading, Attendance (20%)
- Lab Work and Reports (50%)
  - Hands-on Lab work includes simulation & hardware
- Final Project (30%)
  - Extension of Lab work
  - Assemble + Testing + Application development (Robot)