F1/10 Vehicle Documentation

This document details the process of setting one (or more) of the F1/10 autonomous vehicle as well as summarizes the components involved. By the end, any user, even without prior experience should have a mid to high level understanding of the platform and how to use them.

# System Overview

A car with a blue and black object on top

Description automatically generated

Figure : F1/10 Platform

The F1/10 platform is a scaled modular Ackermann vehicular robot with the sensors and actuators needed for autonomous vehicle research and experimentation in areas of perception, localization, and controls developed by the Autonomous Systems Group within RASLAB at the FAMU-FSU College of Engineering.

A diagram of a car

Description automatically generated

Figure : System Architecture

# Hardware Components (v1)

|  |  |
| --- | --- |
| A close-up of a camera  Description automatically generated  Figure : Intel Realsense D435i (Stereo Camera) | * 1x RGB Visual Camera @ 30FPS * 2x Infrared Cameras @ 30FPS * 1x IMU @ 200Hz * Depth Image from Stereo (infrared cameras) @ 30FPS * Pointcloud from depth with optional color up to 4m @ 30FPS |
| A small blue and black device  Description automatically generated  Figure : YDLIDAR X4 (LaserScanner) | * 2D Pointcloud up to 12m @ 8Hz |
| A black electronic device with a fan  Description automatically generated  Figure 5: Nvidia Jetson Orin Nano (Embedded Computer) | * CPU: 6-core Arm® Cortex®-A78AE * GPU: NVIDIA Ampere architecture with 1024 CUDA cores and 32 tensor cores * RAM: 8GB 128-bit LPDDR5 * Storage: 1TB NVMe SSD * Power consumption: 15W |
| A white and black video game controller  Description automatically generated  Figure : Playstation 5 DualSense Controller | * Bluetooth controller |

|  |  |
| --- | --- |
| A red electronic device with black wires  Description automatically generated  Figure : VESC 6 Mk VI (Actuator) | * Brushless motor ESC * Voltage: 11.1v (3S LiPo battery) * Current: 80A continuous (120 A max) * 1x IMU * Actuator feedback |
| A close-up of a usb flash drive  Description automatically generated  Figure : USB Isolator | * Creates a separate, isolated ground * Breaks the ground loop caused by connecting two batteries with different grounds (LiPo and Li-ion) when connecting the VESC to the PC |
| A black rectangular object with a black background  Description automatically generated  Figure : NP-F750 Battery (Jetson + Sensors) | * Battery Type: Li-ion * Voltage: 7.2v (min), 7.7v (nominal), 8.4 (max) * Capacity: 5600mAh * Current (charge): 1A |
| A black rectangular object with a hole in the center  Description automatically generated  Figure : NP-F Adapter | * Voltage: 7.4 v or 12.0v * Current (discharge): 2A |
| A black battery charger with green text  Description automatically generated  Figure : NP-F Battery Charger | * Input Voltage: 5v * Output Voltage: 8.4v * Current (charge): 2A (1A per battery) |
| A battery with a wire attached to it  Description automatically generated  Figure : Venom LiPo Battery (VESC) | * Battery Type: LiPo * Voltage (3S LiPo): 9.2v (min), 11.1v (nominal), 12.6v * Current (discharge): 175A continuous (350A max) * Current (charge): 5A * Capacity: 5000mAh |
| A blue electronic device with buttons and labels  Description automatically generated  Figure : Traxxas EZ Peak Plus Dual Charger | * Voltage (output): 12v (for 3S LiPo) |
| A close-up of a car  Description automatically generated  Figure 14: Traxxas 4 Tec 2.0 VXL Chassis | * Max speed: 70 MPH * Wheelbase: 0.256m * Wheel radius: 0.033m * Overall Drive Ratio: 6.87 |
| A side view of a car  Description automatically generated  Figure : 2019 Cadillac CTS-V | * Plastic body shell |
| A metal frame with a black background  Description automatically generated  Figure : Rollcage | * Made from Nylon-12 |

# Software Requirements:

* Operating system: Ubuntu 22.04 (Codename: Jammy Jellyfish)
* Python version: 3.10
* CUDA version: 12.2
* CuDNN version: 8.9.4.25
* TensorRT version: 8.6.2.3
* ROS Version: 2
* ROS Distro: Humble
* Packages:
  + C++ Drivers:

|  |  |  |
| --- | --- | --- |
| **Package** | **Version/Branch** | **Link to Repository** |
| Librealsense SDK | v2.55.1 | [GitHub](https://github.com/IntelRealSense/librealsense) |
| YDLIDAR SDK | Master | [GitHub](https://github.com/YDLIDAR/LIDAR_SDK) |
| VESC Driver | 1952e79 | [GitHub](https://github.com/privvyledge/f1tenth_system) |

* + Python packages:

|  |  |  |
| --- | --- | --- |
| **Package** | **Version/Branch** | **Purpose** |
| Numpy | 1.21.4 | Array manipulation |
| Acados | 0.1 | Real-time Optimization |
| Casadi | 3.6.5 | Real-time Optimization |
| Do-MPC | 4.6.5 | Optimization Prototyping |
| OpenCV | 4.5.0 | Image Processing |
| Open3D | 0.18.0 | Pointcloud and RGB-D processing and perception |
| Torch | 2.3.0 | CPU/GPU Tensor manipulation |
| Ultralytics | 8.3.39 | Image-based object detection |

* + ROS Packages

|  |  |
| --- | --- |
| **Package** | **Purpose** |
| Realsense ROS Wrapper | Publish realsense data as ROS messages. |
| YDLIDAR ROS Driver | Publish LaserScans as ROS messages |
| Image transport | Image compression/decompression |
| PointCloud Transport | PointCloud compression/decompression |
| Rviz2 | Visualization |
| Image pipeline | Image processing |
| IMU filters | IMU orientation estimation |
| Robot localization | Extended Kalman Filter based sensor fusion |
| F1tenth\_launch | Launches all nodes with defaults for autonomous driving. |

# Hardware Connection Setup (Sensor Assembly)

|  |  |
| --- | --- |
| Step 1: Get the base plate. | A grey rectangular object with a white background  Description automatically generated |
| Step 2: Mount the computer on the baseplate via the four screws on the outer edge with the DC jack and USB facing the rear of the car. | **A computer chip with a white and black rectangular object  Description automatically generated with medium confidence** |
| Step 3: Mount the VESC upside down under the baseplate. | **A computer chip with different colors  Description automatically generated with medium confidence** |
| Step 4: Attach the rollcage to the baseplate. | **A computer generated machine with a green and red base  Description automatically generated with medium confidence** |
| Step 5: Mount the LIDAR to the top-center of the rollcage. | **A computer chip with a blue cap  Description automatically generated** |
| Step 6: Mount the camera to the top-front of the rollcage. | **A computer chip with a blue cap  Description automatically generated with medium confidence** |
| Step 7: Screw in the NP-F battery adapter to the inclined rear part of the rollcage. Then slide in the battery to the adapter. | **A close-up of a machine  Description automatically generated** |

The resulting sensor assembly should be like the photos below:

|  |  |
| --- | --- |
| A close-up of a machine  Description automatically generated  Figure : VESC Assembly | A close-up of a device  Description automatically generated  Figure : Sensor and PC Assembly |

# Hardware Connection Setup (Body Assembly)

|  |  |
| --- | --- |
| Step 1: Attach the base plate to the chassis on the standoffs. | A close-up of a machine  Description automatically generated |
| Step 2: Attach the shell to the chassis | A close up of a machine  Description automatically generated |

# Wiring (Power) Connection Sequence

|  |  |
| --- | --- |
| Step 1: Remove the shell from the chassis. | A blue round object with wires  Description automatically generated |
| Step 2: Connect the powerbank and its dc barrel connector to the jetson | **A close up of a machine  Description automatically generated** |
| Step 3: Place the battery on the chassis | **A battery in a car  Description automatically generated with medium confidence** |
| Step 4: Connect the VESC to the battery using the XT-90 connectors | **A close up of a device  Description automatically generated** |
| Step 5 (optional): Attach the shell to the chassis |  |

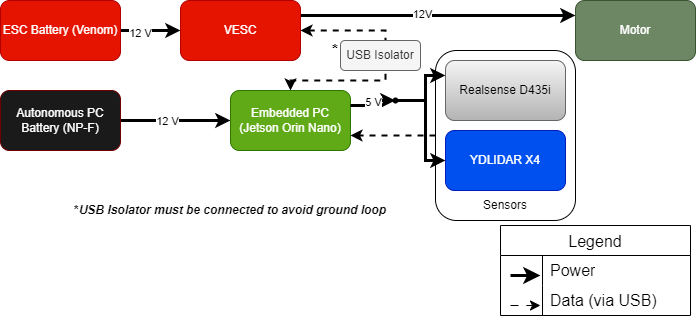
****

Figure : Wiring

# Network Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Computer Name** | **Network Hostname** | **IP Address** | **Network Interface** | **RustDesk ID(Password)** |
| *Remote PC*: Linux Workstation | <remote PC hostname> | 192.168.2.xxx | WiFi: <WiFi name> | <rustdesk ID> : <rustdesk password> |
| *F1/10***:** car1 | <car1 hostname> | 192.168.2.xxx | WiFi: <WiFi name> | <rustdesk ID> : <rustdesk password> |

## Network Connection Steps:

1. (optional: for visualization purposes or multi-machine ROS2 communication) Remote PC:
   1. Connect the Remote PC to the WiFi and enter the password (UVS\_wifi)
   2. Install RustDesk
   3. Setup a permanent password in the RustDesk settings
   4. Open RustDesk
   5. Enter the RustDesk credentials for the F1/10 in the table above
2. On any F1/10
   1. Connect the F1/10 to the WiFi and enter the password.
   2. Install RustDesk
   3. Setup a permanent password in the RustDesk settings

# DualSense\* Controller Setup

|  |  |
| --- | --- |
| Controller front view. Clockwise from the top left A to N.  Figure 20: DualSense Controller (Front)  *Source :* [*https://controller.dl.playstation.net/controller/lang/en/2100002.html*](https://controller.dl.playstation.net/controller/lang/en/2100002.html) | 1. Directional Buttons 2. Create/Share button 3. Light bar 4. Touch pad/button 5. Player Indicator 6. Options button 7. Action buttons 8. Right stick/R3 9. Speaker 10. PS Button 11. Headset Jack 12. Microphone 13. Mute button 14. Left stick/L3 |
| Controller top view. From the left A to E.  Figure : Dualsense Controller (Top)  *Source:* [*https://controller.dl.playstation.net/controller/lang/en/2100002.html*](https://controller.dl.playstation.net/controller/lang/en/2100002.html) | 1. R1 button 2. R2 button 3. USB port 4. L1 Button 5. L2 Button |

\* *Other controllers tested with the exact configuration and performance: DualShock 4, Logitech F710*

## Pairing Instructions:

1. Press and hold the PS button (J) and the Create/Share (B) button at the same time until the Light Bar (C) blinks continuously.
2. Connect to the Jetson via RustDesk as detailed above
3. Open Bluetooth settings and connect to the DualSense Wireless Controller
4. The light bar (C) and Player Indicator (E) should remain solid when paired.

**A white video game controller with blue arrows

Description automatically generated**

Figure : Control Scheme

# Software Setup

1. Install ROS 2 Humble: Follow the official installation instructions [found here](https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html).
2. Configure the environment:

|  |  |
| --- | --- |
| * Automatically load the path to ROS2 packages when a new terminal is opened. | * echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc |
| * Set CYCLONE\_DDS as the default RMW implementation | 1. sudo apt update && sudo apt install ros-${ROS\_DISTRO}-rmw-cyclonedds-cpp 2. echo "export RMW\_IMPLEMENTATION=rmw\_fastrtps\_cpp" >> ~/.bashrc |
| * Create the ros2\_ws | * mkdir -p ~/ros2\_ws/src && colcon build –symlink-install |

1. Clone the necessary packages (listed in Software Requirements section above) to the ros2\_ws and recompile.

## Launch Teleoperation Nodes (For a detailed and comprehensive explanation of every launch argument, check the comments in the launch file [**here**](https://github.com/privvyledge/f1tenth_launch/blob/humble-dev/launch/teleop.launch.py).)

1. Launch the teleop nodes with the following commands: ros2 launch f1tenth\_launch teleop.launch.py launch\_vehicle:=True launch\_sensors:=False launch\_localization:=False vesc\_poll\_rate:=50.0.
2. Nodes launched:

|  |  |
| --- | --- |
| * VESC ROS Driver | * Steering * Motor * IMU |
| * Joystick Driver (DualSense) | * Publishes joystick inputs as ROS messages |
| * Joy to Ackermann Node | * Converts joystick messages to car speed and steering commands. |
| * Ackermann to VESC node | * Converts longitudinal speed and steering commands to motor RPM and steering PWM commands. |
| * VESC to Ackermann node | * Publishes kinematic wheel odometry node |

1. Parameters:

|  |  |
| --- | --- |
| * launch\_vehicle | * Whether to turn on the VESC. Useful for testing only sensors or replaying recorded data. |
| * vesc\_poll\_rate | * Rate at which to communicate with the VESC to send commands and receive feedback. Higher frequencies lead to more sensor noise. |
| * launch\_sensors | * Whether to turn on all sensors. |
| * launch\_localization | * Whether to launch local and global localization nodes. |

## Driving instructions:

1. Hold down the L1 button to arm the motors (Note: the car will not move if L1 is released)
2. Move the left analog stick up and down to drive/reverse. (Note: move slowly otherwise the car will accelerate rapidly).
3. Move the right analog stick left and right to steer the car

NOTES: The provided scripts launch visualization and sensor compression/decompression nodes as well as enabling sensors.

# Battery Management

#### LiPo Batteries (e.g Venom 3S LiPo)

1. Keep the batteries at storage voltage of 11.3v when not in use.
2. Charge the battery using the Traxxas EZ Peak Plus Dual Charger in advance mode. See official instructions [here](https://traxxas.com/sites/default/files/HKC18030-R05_2972X-TraxxasID-Dual%20Charger-INST-EN.pdf) (page 3).
3. The VESC prevents the battery from being drained below safe voltages as well as configurable max charge/discharge current limits.

#### DualSense Battery:

1. The device includes an in-built Battery Management System which includes under-voltage and over-voltage protection.
2. The device automatically shuts off when the battery is low.
3. Recharge via USB.

#### NP-F Battery:

1. The NP-F battery adapter prevents the battery from being drained below safe voltages.
2. Charge via the NP-F chargers shown above.

# Operating Procedures

## Pre-operation checklist

1. Inspect the batteries
2. Charge the batteries (ESC, PC and controller) to full voltage.
3. Ensure all cables are connected and there are no shorts.

## Startup sequence:

### Power-up sequence

1. Connect the DC jack to the Jetson to turn it on.
2. Connect the sensors to the jetson.
3. Connect the LiPo battery to the VESC.
4. Connect the micro-USB from the VESC to the USB isolator
5. Connect the USB isolator to the Jetson
6. Connect the sensors to the Jetson via USB

### Connection establishment

1. The jetsons power status LED should be solid green.
2. The VESCs status LED should be solid red.
3. The USB isolators LED should be solid red.

### Normal Operation Procedures:

* Actuators (steering and throttle) should respond when armed.
* The LIDAR should start spinning (if sensors are activated).

### Shutdown sequence

1. Press CTRL-C on the terminal used to launch the nodes to cancel all running nodes (this could take about 4 seconds to safely shutdown the actuators and sensors).
2. Shut down the PC: sudo shutdown now.
3. Disconnect all batteries, especially the VESC.

### Emergency procedures

1. Elevate the car off the ground.
2. Disconnect the VESC from the battery.
3. Unplug the DC barrel jack from the Jetson.

# Troubleshooting Guide

**Network Connectivity Issues**

|  |  |
| --- | --- |
| **Issue** | **Solution** |
| * ROS topics not visible across machines | * Make sure both machines are connected to the same network (not necessary for RustDesk) * Ping the IP address of machine B from machine A, i.e “ping 192.168.2.194” from Remote PC * Ensure the terminal environment variable $ROS\_LOCALHOST\_ONLY=0 * Ensure the terminal environment variable $ROS\_DOMAIN\_ID is either unset on both machines or equal |
| * ROS topic rates too low on remote PC | * For high bandwidth messages like PointClouds or images, use compressed messages as detailed below. * Reduce the number of devices connected to the WiFi router * Make sure the devices are within the range of the WiFi router |
| * RustDesk (or SSH) disconnects | * Occurs when CPU utilization is at 100% which usually occurs if Rviz is running on the Jetson while running teleoperation with sensing and other nodes. Run Rviz on the remote PC instead. |

**Controller Connectivity Issues**

|  |  |  |
| --- | --- | --- |
| **Issue** | **Explanation** | **Solution** |
| * Controller keeps disconnecting (or does not connect) | * Due to the cheap Network Interface Card included in the Jetson, WiFi and Bluetooth connections can be unstable. | * For now, the only solution is to retry until successful. * (Future) replace with better NIC or adapters. |

**Power Issues**

|  |  |  |
| --- | --- | --- |
| **Issue** | **Explanation** | **Solution** |
| * LiPo Battery connected to VESC drained below safe voltage. | * Due to hardware/software bugs in the VESC 6 VI, the auto-disconnect function does not work and the VESC keeps draining the battery (although at low rates). | * Disconnect the battery from the VESC after each use. * Use a Power Distribution Unit. * Upgrade to VESC 6 HP. |
| * Jetson node rates lower than expected. | * Due to the limited power output of the NP-F battery and adapter which is incapable of supplying more than half the power required by the Jetson, the Jetsons performance is throttled. | * (Future) replace the NP-F components with a Power Distribution Unit capable of supplying the required power. |
| * RealSense or LIDAR fails to connect. | * Due to the limited output of the Jetsons single USB controller, starting up all sensors at once leads to current limits which temporarily disables some of the USB outputs. | * Software delay is used to startup the sensors sequentially. * Add a powered USB hub. |