CARMA 1tenth Hardware and Software Setup

Main hardware setup instructions: https://racecarj.com/pages/build-instructions

1. Chassis prep: https://www.jetsonhacks.com/2017/11/30/racecar-j-chassis-preparation/
2. Platform prep: https://www.jetsonhacks.com/2018/01/24/3578/
    a. Ignore the instructions for TX2 and FocBox mounting
3. Mount ESC: https://www.jetsonhacks.com/2019/08/16/install-vesc-6-plus-on-racecar-j/
4. Setting up the Xavier
    a. Note: You will need to insert a micro-sd card
    b. Install OS: https://www.stereolabs.com/blog/getting-started-with-jetson-xavier-nx/
    c. Set it to boot from the ssd: https://www.jetsonhacks.com/2020/05/29/jetson-xavier-nx-run-from-ssd/
    d. Install ROS and ZED SDK: https://www.stereolabs.com/blog/ros-and-nvidia-jetson-xavier-nx/
    e. Mounting: Remove the plastic base and detach the wifi antenna from the bottom. Then, mount it to the 3 available mounting holes (4th is covered by the ESC) with the usb ports on the same side as the usb hub.
5. Assemble the car: https://www.jetsonhacks.com/2018/01/28/racecar-j-initial-assembly/
6. Install the Lidar: https://www.jetsonhacks.com/2018/02/20/racecar-j-hokuyo-ust-10lx-lidar/
    a. For power wiring, the Lidar should be soldered to the 3.5mm jack. Then there should be a single cable, comprised of 4 wires soldered together: power cables that connect to the Krisdonia power bank output, USB hub, Xavier, and the Lidar 3.5mm jack.
7. Install the ZED camera
    a. Using one of the ¼-20 screws, bolt it to the hole in the upper platform, and get it as straight as possible.
8. Set up the IMU
    a. Note - you will need to install the Arduino IDE and flash the IMU on a machine that isn't arm (PC, but not the xavier nx)
    b. The following instructions are a supplement to this guide - http://wiki.ros.org/razor_imu_9dof
    c. You do not need the visual python dependency on the pc you are using to flash the IMU.
    d. Follow the steps in the 'Install Arduino Firmware' section
    e. Once that is complete, connect the IMU to the car and install the visual-python library on the Xavier.
    f. If you want a visualization, Install visual-python library - Instructions partially from https://askubuntu.com/questions/1072817/installing-visual-python-in-ubuntu
        i. sudo apt install python3-pip
        ii. apt install libffi-dev
        iii. pip3 install Cython
        iv. pip3 install vpython
    g. Now you can follow the rest of the steps to run and view the output from the IMU.
9. Test the sensors

a. Setup: Turn on the battery pack, switch the output to 12v. Plug in the power cable, so that the Xavier, USB hub, and Lidar are powered on. Connect the Xavier to a monitor and keyboard, and log in.

b. Testing the Lidar: Ensure that you have followed the steps in the jetsonhacks install tutorial, and that you can ping the lidar at 192.168.0.10. Now connect the Xavier to the internet by swapping the Lidar Ethernet for a real Ethernet, and install the ROS uirg_node – sudo apt install ros-melodic-urg-node. Next, follow the instructions starting at step 3 from: https://www.programmersought.com/article/6077522524/.

c. Testing the camera: Ensure that you have followed the ZED ROS tutorial and installed the ZED SDK. To view the camera feed, run - roslaunch zed_wrapper zed2.launch, and then open rviz with the command – rviz, and then open up one of the camera topics; ex: zed2/zed_node/left/image_rect_color. You can also launch a trajectory estimation program with - roslaunch zed_display_rviz display_zed2.launch.

d. Testing the IMU: Once you have the IMU flashed using the above steps, you should be able to test it with the steps described in - http://wiki.ros.org/razor_imu_9dof part 6 – Testing the AHRS.

10. Set up the VESC driver and test vehicle motion.

a. For this we will use libraries from the MIT 1tenth racecar and racecarj.

b. Install the udev rules and drivers/nodes

i. clone https://github.com/RacecarJ/installRACECARJ

ii. run ./scripts/installRACECARUdev.sh

1. Make sure to either reboot or reconnect each device so that the udev rules are applied
2. Note – IMU goes to /dev/imu, and the VESC goes to /dev/vesc
3. Might want to add udev rule for joystick in the future

iii. mkdir ~/racecar-ws

iv. run ./scripts/installMITRACECARVESC6.sh

1. This script installs the MIT racecar vesc driver, control nodes, imu driver, and teleop nodes.

c. This script may fail, complaining about an opencv error on the zed-ros-wrapper library. If so, do the following:

i. rm -rf ~/racecar-ws/src/racecar/zed-ros-wrapper

ii. cd ~/racecar-ws

iii. catkin_make // This should succeed

iv. cd src

v. git clone https://github.com/stereolabs/zed-ros-wrapper.git

vi. cd ..

vii. catkin_make

d. Running the car – Based on this tutorial: https://www.jetsonhacks.com/2018/08/19/racecar-j-ros-teleoperation/

i. Plug in the battery to the VESC, and make sure you see lights on the VESC

ii. Plug in the Logitech joystick, and make sure the controller type is set to 'D' (slider on the front)

iii. Go to https://gamepad-tester.com/ and make sure that the joystick is responding. It also should be first in the list. If it is not, reboot the Xavier and try again. (This ensures that it is registered as /dev/input/js0)

iv. source ~/racecar-ws/devel/setup.bash

v. roslaunch racecar teleop.launch

   1. Note: The only error you should see is: [ERROR] [1597353502.587704947]: Sent 12 'r' commands to the IMU but did not receive an expected response matching format 'IMU rate set to (\d+) Hz' with value '50'.

vi. To control the car, hold LB (Dead-man switch). Then, the left trigger controls the wheel turn angle and the left stick controls the wheel speed (though it is backwards).

   1. These settings can be changed, using instructions in the teleoperation tutorial.

11. Pictures of final build: