ROS2 VICON RECEIVER FULL GUIDES/RESOURCES

Table of Contents:

Guide 1: Safety Guidelines

Hardware/Physical Safety Notes

When dealing with the drone when armed the following four rules must be followed:

- The RC Controller must be on hand, ready to kill the drone if necessary.
- The user must wear kevlar gloves and safety goggles when dealing with the drone when the power is on.
- The safety net must be drawn up and closed.
- Before arming the drone, make sure all on-board parts are away from the propellers and are firmly attached along with the propellers to the drone

Software Safety Notes:

- Do not change between flight modes when flying, unless an emergency calls for it
    - User should then switch to either stabilised or landing flight mode
- Do not kill any of the nodes that were started until the drone has been disarmed and the kill-switch is on.
- Do not fly in stabilised flight mode unless the user is an experienced drone flyer.
- Make sure that the drone is receiving the position lock correctly - reference the Quick-Start Guide for how to do so.
- Always run offboard missions in Gazebo Simulation Software before running it on the real drone

Guide 2: Further Installation/Calibration Resources

- For basic information on how the PX4 firmware and drone controller works, see the Installation/Calibration Guide PDF.
- For more information on offboard control and the usage of ROS2 with PX4 firmware, the following links are for the user below:
    - https://docs.px4.io/main/en/getting_started/
        - Getting started on understanding PX4 Drone Configuration
    - https://docs.px4.io/main/en/frames_multicopter/holybro_x500v2_pixhawk6c.html
        - Hardware Method of Building Holybro_x500v2
    - https://docs.px4.io/main/en/assembly/quick_start_pixhawk6c.html
        - Understanding Pixhawk 6C Controller Wiring
    - https://docs.px4.io/main/en/config/
        - Basic Drone Configuration Guides
    - https://docs.px4.io/main/en/log/flight_log_analysis.html
    - https://logs.px4.io/
        - Flight Log Guide and Flight Log Analysis (Upload Flight Logs to second link)
    - https://docs.px4.io/main/en/advanced_config/parameter_reference.html
        - Full Parameter List in Drone Controller
    - https://docs.px4.io/main/en/advanced_config/tuning_the_ecl_ekf.html
        - EKF Tuning for Position Lock
    - https://docs.px4.io/main/en/dev_setup/building_px4.html
        - Manually building PX4 firmware/PX4 Gazebo Simulation
    - https://docs.px4.io/v1.13/en/middleware/micrortps.html
        - Installing PX4 MicroRTPS bridge onto Onboard
    - https://docs.px4.io/v1.13/en/computer_vision/visual_inertial_odometry.html
        - Integrating visual odometry with vicon guidelines
    - https://docs.px4.io/v1.13/en/ros/ros2.html
        - ROS2 User Guide and Offboard Control Application + Running Gazebo Simulations with MicroRTPS bridge’

Guide 3: Customization/Pipeline Guide

vicon_position_bridge

This package largely contains ROS2 executables regarding tracking and communicating the drone's position in the vicon system to the drone controller. If a user would like to configure how to record the vicon position data, use this package.

- see graphing.launch.py file to configure which of the position graphing nodes and data savers would the user like to run
- in each of the pose_grapher_… nodes, there is one subscriber to vehicle_odometry for simulation use, and one subscriber to vicon for real drone use. Please comment out one of the two subscribers for whichever situation the user would require.
- List of nodes/launch files (see each code description for more details):
    - node: fake_pose_pub
    - node: pose_data_save
    - node: pose_grapher_xy
    - node: pose_grapher_xyz
    - node: pose_grapher_yz
    - node: pose_grapher_xz
    - node: pose_pub
    - node: pose_sub
    - launch: graphing_launch

drone_offboard_missions

This package largely contains ROS2 executables regarding singular offboard missions flying in set trajectories. If the user would like to see examples and create their own mission from source, use this package.

- List of nodes/launch files (see each code description for more details):
    - node: offboard_control
    - node: offboard_control_L
    - node: offboard_control_circle
    - node: offboard_control_circle_tilted
    - node: offboard_control_helix
    - node: offboard_control_square
    - node: offboard_hover
    - node: offboard_ex

drone_keyboard_controller

This package largely contains ROS2 executables, config files, and launch files surrounding a keyboard controller to manually control the drone and have it easily fly in set trajectories. If the user would like easy use of the drone for data collection, demonstration, or for other reasons, use this package.

- see config0.yaml file to configure the parameters regarding the drone's flight behaviour while using keyboard controller

- see param_trajectory.py file (not under config folder, under drone_keyboard_controller) to configure the trajectory in continuous trajectory autonomous mission
- List of nodes/launch files (see each code description for more details):
    - config: config0
    - config: param_trajectory
    - launch: command_control
    - node: set_path_hover
    - node: set_path_circle
    - node: set_path_helix
    - node: set_path_square
    - node: set_path_return
    - node: set_path_linear_setpoint
    - node: set_path_continuous_setpoint
    - node: offboard_control
    - node: keyboard_controller
    - node: command_control

Guide 4: Troubleshooting Guide

Refer to the Guide for quick fixes to certain issues when troubleshooting

Had to follow v.1.13 documentation and use micro RTPS instead of micro XCRE dds client, which is only supported by unstable v1.14

- https://docs.px4.io/v1.13/en/ros/ros2_comm.html

Making RTPS client/ disabling microdds client

- Changed px4 firmware build file and added line to configure client
- https://github.com/PX4/PX4-Autopilot/issues/19572
- Added line to disable microdds client
- https://github.com/PX4/PX4-Autopilot/issues/19482

Fixing USB Port Unresponsive

- Changed Baud-Rate from default value of 460800 to 57600 to prevent unstable USB connection between Onboard and Controller
- https://discuss.px4.io/t/px4-ros-com-applications-wont-receive-any-data-via-micrortps/25293

Radio Telemetry

- Do not try using Radio Telemetry to directly communicate ROS2 messages from GCS to the drone controller. Instead, connect to the onboard computer via wifi and then establish the microRTPS bridge over USB cable to the controller.

Onboard Computer

- Onboard Computer Sign-in:
- Username: rob498
- Password: 99bobotw
- Both the onboard computer and the GCS should be connected to TP-Link_rob498 wifi

Reference Frame Issue:

- If the controller does not accept the local position of the drone, or the yaw rate is incorrect, there is likely a reference frame issue in the position message sent to the controller.
- The Drone Reference Frame is in NED to NED, whereas vicon is from FLU to FLU
- Rotation/Reference Changes Links below:
- https://stackoverflow.com/questions/49790453/enu-ned-frame-conversion-using-quaternions

If there is a lag/drift in position lock:

- Try to tune EKF Parameters in QGC
    - Not always true, but the following steps may help (see PX4 guidelines for more details):
        - setting all noise parameters to max except for visual odometry noise parameters
        - setting visual odometry gate parameter to minimal value
        - decreasing priority of in-built sensors
        - Checking for at least 30 hz of frequency and less than 0.01 seconds of latency in ROS2 Vicon messages being received by controller

Simulation Drone not flying:

- If the simulation drone is not flying/arming when testing offboard nodes, run the following commands in the pxh terminal where the simulation was launched.

    - param set COM_RCL_EXCEPT 4
    - param set NAV_DLL_ACT 0
    - param set NAV_RCL_ACT 0

If a failed drone keeps spawning instead of a default drone in Gazebo Simulation

- run the following command and remake the drone
    - killall gzserver

Known Issues to work on:

- Configurable keybindings on the keyboard controller via config file
- Easily switch pose graphers between simulation and real drone tracking
- Switching between Offboard Mode and Position Mode causes drone to crash in Position Mode from flying rapidly upwards
- Smooth out and perfect continuous customizable setpoint trajectory mission in keyboard controller