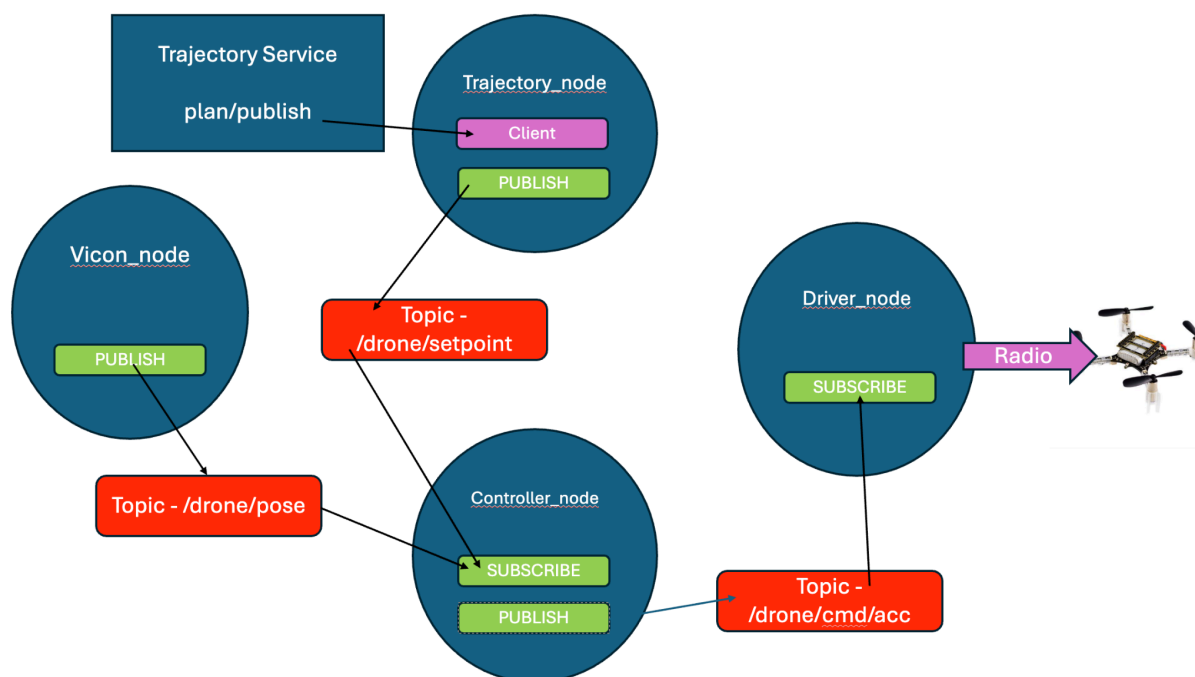
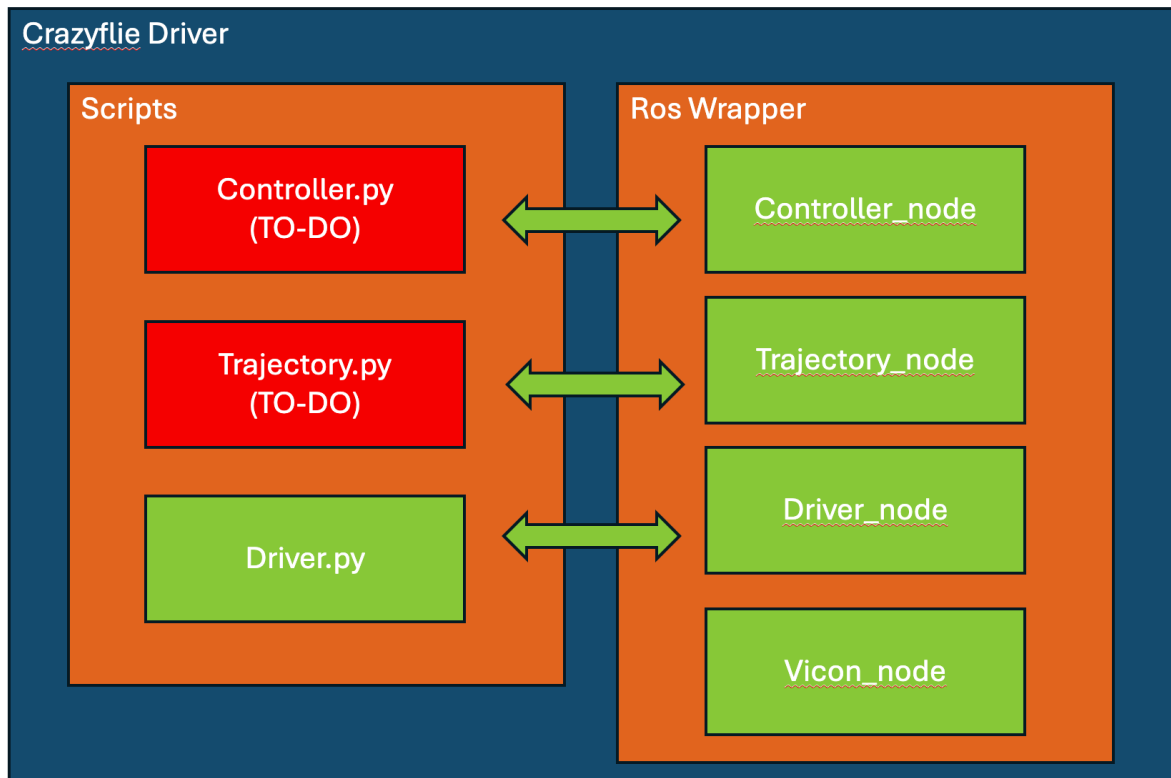


# INSTRUCTIONS

REPO - <https://github.com/IllinoisReliableAutonomyGroup/CrazyflieDrivers>

Crazyflie Code Structure -



Exporting code from FalconGym

TO-DO:

Complete [controller.py](#) in crazyflie\_ros/src/crazyflie\_ros/[controller.py](#)

- Copy scripts/ece484\_state\_controller.py state\_controller() function to crazyflie\_ros/src/crazyflie\_ros/[controller.py](#) Dronecontroller.solve()
- Note Dronecontroller.solve() function takes current state and reference/setpoint state.
- crazyflie\_ros/src/crazyflie\_ros/[controller.py](#) is a class based controller if you implement mpc/lqr or pid use Dronecontroller.iinit() to initialize your controllers parameters accordingly

Complete [Trajectory.py](#) in crazyflie\_ros/src/crazyflie\_ros/[trajectory.py](#)

- If you haven't implemented a trajectory function before then use this script to generate one. Complete Planner.plan() function. Implement B-spline/cubic spline or other interpolating polynomial splines.
- If you have already used a trajectory planner then simply paste the planning algorithm in [Planner.py](#). Refer to each function head for more details.
- NOTE: planner is expected to output full reference state [poses, velocities, accelerations, yaws, timestamps]. If you plan on not using velocity and acceleration reference waypoints simply append them with zero.