Getting started - oDrive V3.5

Wednesday, 31. July 2019 09:05

- Connect the three phases of one or both motors
- connect the brake resistor to AUX + and AUX - (resistor has no polarity)
- connect the ABI encoder (5 pins: A, B, I (or X, Z), 5V and GND)
- connect the oDrive to the computer via USB
- check that the switch on SW1 is on RUN, not on DFU (= device firmware update)
- connect the oDrive DC+ and DC- to a direct current power supply (pay attention to polarity); minimum voltage for all board versions is 12 VDC
- download and install odrivetool following these steps: https://docs.odriverobotics.com/#downloading-and-installing-tools (offline version Getting Started _ ODrive.pdf)
- in your bash/shell, move to the directory that has odrivetool, typically in odrive/tools/, and run the tool by typing ./odrivetool; for Windows, see description on oDrive homepage(Link)
- turn on power supply with at least half of the oDrive's maximum voltage (there are 24V oDrives and 48V oDrives, higher voltages ~~may be available online soon~~ are available in the oDrive shop), start with low maximal current, e. g. 2A
- check if the PWR LED on your oDrive is on
- odrivetool should recognize the oDrive after some seconds
  but if you want a specific oDrive to be connected, type $ ./odrivetool --serial-number <your serial number here>
- set up for the motor (in this example, connected to M0, which odrivetool knows as axis0)
    - odrivetool should automatically recognize the odrive and put out a 'connected' notice
    - to inspect all available parameters and variables, type odrv0 (or odrv1, depending on how many oDrives are connected)
    - you can go into deeper variable levels pertaining to python objects by typing for example odrv0.config or odrv0.axis0
    - first, make sure the motor will not get overloaded, so set limits (the values given here are for a T-Motor MN4006-23, 0.3A idle current, 16 A max current)
        - ○ odrv0.axis0.motor.config.current_lim = 2
        - ○ odrv0.axis0.controller.config.vel_limit = 2000 (velocity is in encoder counts / sec)
        - ○ odrv0.axis0.motor.config.calibration_current = 1
        - ○ odrv0.config.brake_resistance = 2.0
        - ○ odrv0.axis0.motor.config.pole_pairs = 12

  - odrv0.axis0.motor.config.pole_pairs = 12
  - odrv0.axis0.motor.config.motor_type = 0 (for high current motors; for gimbal motors set motor_type = 1)
  - odrv0.axis0.encoder.config.cpr = 8192 (CPR = 4xPPR, PPR being the pulses per rotation for one channel. You have to multiply by 4, since oDrive is counting both rising and falling edges for both A and B channel)
  - odrv0.save_configuration() (keep this command in mind since for many errors, this command followed by odrv0.reboot() helps)
  - odrv0.reboot()
- now do the calibration, save and reboot again:
  - odrv0.axis0.requested_state = AXIS_STATE_FULL_CALIBRATION_SEQUENCE
- for now, control should work; type
  - odrv0.axis0.requested_state = AXIS_STATE_CLOSED_LOOP_CONTROL ← this automatically sets position control
  - if calibration fails in the first place and the coils are not powered,
    set odrv0.axis0.controller.config.vel_limit_tolerance = 0.0
    - **please not that this setting makes the controller IGNORE any velocity overshoot**!
- tune the controller via its parameters, which you can all check by typing
  - odrv0.axis0.controller.config
- you can change control mode in odrvX.axisX.controller.config.control_mode (for control modes see below)
- if you use the same motor each time, you can avoid the calibration required at each startup by typing odrv0.axis0.motor