* include Arduino and Freefly libraries
* define necessary freefly vars (max/min command vals, etc)
* define other vars (pins, PID vars, timers, etc)
* maybe build unions/structures for combining serial bytes into x.fval/y.fval vars?
* **setup**
* setup pins (laser, RC Rx)
* start serial ports (PC serial mon (out), Raspi (in), MoVI (out))
* start freefly api
* **loop**
* if not tracking target (manual mode), read PWM data from RC Rx
  + set corresponding vars to pin values
  + if (tagPin == 1) //from RC Rx
    - trackingTarget = TRUE
  + else
    - convert PWM value to panCommandValue
    - convert PWM value to tiltCommandValue
    - convert PWM value to zoomCommandValue
* if tracking target (auto mode), read serial data from Pi (and check disengage PWM pin)
  + if disengage PWM pin == 1
    - disengage
  + else continue
  + set x/y errors and targetVisible vars (use unions x.fval/y.fval?)
  + if (targetVisible == 1) //from Pi
    - trackingTarget = TRUE
    - calculate x PID and set panCommandValue
    - calculate y PID and set tiltCommandValue
    - if(x.fval and y.fval <= targetWindowSize)
      * fire laser
      * else turn off laser
  + else trackingTarget = FALSE
    - * notify user (cam OSD?)
* if (counter % 10 == 0) to slow down output to gimbal
  + send gimbal current pan/tilt command vals to MoVI over serial