* ~~Controller Data Sent~~
  + ~~Establish Protocol~~
  + ~~Raw Sensor (Found from Experiment)~~
    - Which Sensor?
    - ~~Which Lighthouse?~~
    - ~~X or Y Axis~~
    - ~~Validity (If sweep is > 8333 us, not valid)~~
    - ~~Use Lighthouse Redox + Equations given!!~~
      * ~~Length \* 48 = ticks~~
      * ~~(Ticks - 2501)/500 -> Truncate -> Data Byte for identifying lighthouse~~
* ~~Send data to unity, and use there.~~
  + ~~Write data to unity~~
  + ~~Send data from unity~~
  + Convert binary into ASCII characters so each message is a 32 bit number
    - [char 1] [char 2] [char 3] [char 4];

Fun fact: the light covered by the swinging axis travels at a rate of 1 degree per 46.3 microseconds. Or (1/21600) seconds.

Also, the wait time absolutely cannot exceed 8333 microseconds, since reaches 180 degrees

* ~~Decode Dtime (Computer)~~
  + ~~Raw Sensor Data (USE THE CODE FROM DEBUG ARDUINO)’~~
  + ~~Angle From Lighthouse (From delta time, 2π rads (360°) every (1/60)s)~~

X = Microseconds

* + - ~~A(x)= 360x \* (60/10^6)~~

OR

* + - ~~A(x)= 2πx \* (60/10^6)~~

***2/2/2023***

* + ~~ON UNITY Create a 2d representation of a singular sensor from a singular lighthouse (raw data)~~
    - ~~Create sensor object~~
    - ~~Get x and y coordinates from lighthouse angles~~
    - ~~Plot out sensor~~
    - ~~Try with multiple sensors~~
    - ~~Try with multiple lighthouses~~

***2/16/2023***

* + ~~Previous but now with filter data~~
    - ~~Apply~~ [~~Kalman Filter~~](https://www.youtube.com/watch?v=OiUS2926nQM)~~/Some custom way to filter data~~
      * ~~Change in Angle - Weighted Average~~
        + ~~Quadratic Eq.?~~

~~-(x-180)(x+180)~~

~~Or use whatever max angle~~

* + - * + ~~Bell Curve Eq. (Normal Dist)?~~

~~Find Standard Deviation~~

* + - ~~try with multiple sensors~~
    - ~~Try with multiple lighthouses~~
  + Test out limit switches with Arduino and Unity functionality
    - Send alongside sensor data in 32 bit number

***3/3/2023***

**The frame of the gun should be done now, and should be constructed and ready to go**

* Translate sensors into real world space relative to sensors
  + Get lighthouse frame dimensions
    - Identify where lighthouses go
  + Sensor Positions Relative to Object
    - Get Model From Chris
  + Log Distance Between Each Sensor

***3/30/2023***

* Find Position/Rotation of Object based on [SolvePnP](https://docs.opencv.org/3.4/d5/d1f/calib3d_solvePnP.html) or Trammel Hudson’s method
  + Compare with IMU, average data out through another filter.