Setup Instructions

1. Files you need to have in the same folder
   1. GroundSoftware#
      1. (highest number available is the most current version)
   2. UDIP\_Lib#
      1. (highest number available is the most current version)
   3. Data file from flight
   4. Output folder if wanted
      1. Line 27 in GroundSoftware
         1. String name should be folder\_name/ format
         2. If Output is “” it will just output any files created into the folder with all the other files and the output folder would not be needed

Completed

Plan to do

Write a readme file

How to run it

What the file setup should be / what and where the parts should be

What to do to fix some common errors

1. Initialization
   1. Load\_data()

Reads data and sets up UDIP\_Packets, ind\_medium, ind\_burst, ind\_large, ind\_sensor

* 1. UDIP\_read(filename)

Conversion software that thats what the flight software wrote and turns it into an array of sweep and sensor objects

1. First scan: quickly scan the data for points of interest
   1. Plot\_Percentiles(start,stop,type,percentile=95)

Go through all of the sweeps in the time interval finding the chosen percentile for each sweep and plot on a time vs current graph

Preliminary scan to check if our data contains any promising regions

* 1. Plot\_Sensors(start,stop,accel=False,gyro=False,magno=False,temp=False)

Plots the sensor data in the chosen interval where the keyword is true. Would plot each type on its own graph with legends to differentiate (accelerometer x, y, and z on same graph)

Used to see the sensor information of the flight

* + 1. Maybe create it as a wrapper
    2. Maybe Stack plots

1. Packet exploration: check on individual sweeps
   1. Plot\_sweep(time=None,index=None,fit=False,hysteresis=False)

If time and index both have values or are both none the function returns invalid

If time has a value it will plot the sweep closest to that time

If index has a value it will plot the sweep at that index in the packet array

The value of fit will determine if the fitting function is used and the result displayed on the plot.

The value of hysteresis will determine if the hysteresis if removed before the fitting function is called

The plot will have a graph of voltage vs current

* 1. Plot\_Many\_Sweeps()

Brings up a loop in which you can plot many different sweeps one after the other. You control it by following the input instructions with the keyboard. Press q to exit

* 1. Plot\_constant(time=None,index=None)

If time and index both have values or are both none the function returns invalid

If time has a value it will plot the constant closest to that time

If index has a value it will plot the constant at that index in the packet array

The plot will have multiple plots on the same graph of time vs current

1. Data analysis: determining the over all trends in data
   1. Plot\_all(start,stop,type,hysteresis=False,height=False)

Plots the temperature and density of the chosen time interval for the given type on two separate graphs. Hysteresis determines if the hysteresis is accounted for before going to the fitting function. Depending on the value of height the plots are graphed with height as the y axis or time as the x axis.

1. Helper / Miscellaneous functions
   1. find\_index(UPID\_packets)

Returns arrays containing the indexes of all packet types in UDIP\_packets

* 1. find\_closest(target,type)

Returns the index in UDIP\_packets of the nearest in time sweep of a given type

* 1. fit\_sweep(index,type,hysteresis)

Returns fit parameters with regard to the hysteresis

* 1. remove\_hystersis(x\_h,y\_h)

Returns x,y arrays with the hysteresis removed

* 1. model(x,parameters)

Returns array of y values based on the parameters

* 1. get\_fit(x,y)

Returns fit parameters

* 1. Packet\_Test()

Calls Load\_data if UDIP\_Packets is empty