README – Reading and tuning the data

Team Cambot (Brandon Bravo, Miguel Jiminez Gomez, Shifan Liu, Hayden Webb)

Reading Data from Arduino

Steps to Upload Code and Read Data from Arduino

To upload the provided code to the Arduino UNO and set up the Serial Monitor to read data from the Arduino and tune the data, follow these steps:

1. Select the Correct Board and Port
2. Upload the Code to the Arduino
3. Open the Serial Monitor
4. Set the Serial Monitor Baud Rate (115200)
5. Monitor the Data

Once the Serial Monitor is open and the correct baud rate is set, you should start seeing data printed on the screen. This will include the RPM, torque, and current readings from the sensors connected to the Arduino, based on the mode you select (RPM or Torque/Current).

```
RPM, Torque, and Current Measurement
Waiting for signal...
Waiting for signal...
Waiting for signal...
```

You can input commands (e.g., **r** for RPM, **t** for Torque/Current) directly into the Serial Monitor's input box at the top and press **Enter** to switch between modes and see corresponding data.

If the input is **r (**RPM):

```
Mode: RPM
Waiting for signal...
Waiting for signal...
Waiting for signal...
```

```
Waiting for signal...
RPM: 216.31
RPM: 254.53
RPM: 207.59
```

If the input is **t** (Torque/Current):

```
Mode: Torque & Current
Torque: 0.06 Ncm
Current: -0.106 A
Torque: 0.08 Ncm
Current: -0.053 A
```

Instructions to Adjust Calibration Values for Load Cell and ACS712

To ensure accurate readings from your load cell (HX711) and current sensor (ACS712), it's essential to calibrate and adjust the MULTIPLIER value based on real reference measurements. Follow these steps to perform this adjustment:

Steps for adjusting the Load Cell Calibration (MULTIPLIER)

1. Prepare a Known accurate motor
2. Upload the Code
3. Open the Serial Monitor
4. Fix the known motor and the shaft on the Load Cell
5. Start the motor and observe the output

Copy the Data from the Serial Monitor over a period of time while the system is running. Alternatively, users can use a serial data logger (like the Arduino Serial Plotter or external logging software) to log this data to a file.

Depending on your system, users may expect a linear relationship between output torque and actual torque from the supplier. Users can fit a linear regression line to the data points to compare the actual behavior with the expected linear model.

Example data from MN5006 Antigravity Type 4-6S UAV Motor KV450:

Propeller	Throttle	Voltage (V)	Current (A)	Power (W)	RPM	Torque (N*m)	Thrust (g)	Efficiency (g/W)
T-MOTOR P17*5.8" CF	40%	23.70	1.63	39	2481	0.10	501	13.02
	45%	23.67	2.21	52	2790	0.13	640	12.24
	50%	23.64	2.88	68	3103	0.15	784	11.5
	55%	23.61	3.71	88	3410	0.19	942	10.75
	60%	23.57	4.66	110	3689	0.22	1119	10.18
	65%	23.54	5.68	134	3947	0.25	1289	9.64
	70%	23.50	6.78	159	4203	0.29	1454	9.13
	75%	23.46	8.00	188	4442	0.32	1631	8.69
	80%	23.42	9.28	217	4664	0.35	1798	8.28
	90%	23.32	12.31	287	5105	0.43	2186	7.62
	100%	23.22	15.41	358	5491	0.50	2538	7.09

With the same current implemented on the motor, check the multiplier between the output torque from the dynamometer and the torque from a reliable source.

```
// Tuning multiplier for the loadcell
#define MULTIPLIER 2280.f
```