P2 Starter Code Usage - A Brief Summary

Some General Rules

- When working with a robot, always be ready to catch it when it moves
    - This is especially important when moving leg motors
- Never forcibly move body joints unless they are powered off
- Never forcibly move head joints under any circumstances
- Always power down the body joints as the last thing in any of your scripts
- Once your robot power reading is below 10.8V, consider saving your work and charging it
    - Each robot has an 11.1V, 3-cell LiPo battery
    - Each cell is 3.7V ±0.5V
    - So, the total safe range of the battery is 9.6V-12.6V
    - However, many robots start to power down at around 10.0-10.5V
- Always clear your working directory on the robot before putting it back

---

Helper Functions - *helper_utils.py*

This file is a set of helper functions used in all the other files, and there are a few additional unused functions that you may find useful. This file must be in the same directory as any other scripts used on the robot.

Some key functions: [TODO: FINISH THIS SECTION]

`init_robot_data(safe_mode, swap_hands)`

- Initializes all the internal data that should be kept track of
- `safe_mode`: whether to terminate on an invalid function call (default: True)
- `swap_hands`: whether to swap left and right hand IDs (default: False)

get_body_pos(id)

- Returns single or multiple body positions based on ID
- id: None, a single ID, or a list of IDs

Driver Script - *p2_driver.py*

This file currently performs the motion defined in a .csv file as generated by pose_mode.py, though you can change this behavior if needed. It reads the .csv movement file, runs the motion, and simultaneously collects accelerometer and gyroscope data on a separate thread. The sensor data will be output as a .csv file.

Usage:

```
>> python3 p2_driver.py [-h] -i input_file [-o output_file] [-s]
```

Example:

```
>> python3 p2_driver.py -i my_movement
```

- Reads *my_movement.csv* as input
- Runs movement defined in *my_movement.csv* while collecting sensor data
- Writes sensor data to *my_movement_output.csv*

Example:

```
>> python3 p2_driver.py -i my_movement -o my_sensors
```

- Reads *my_movement.csv* as input
- Runs movement defined in *my_movement.csv* while collecting sensor data
- Writes sensor data to *my_sensors.csv*

Example:

```
>> python3 p2_driver.py -h
```

- Displays usage information for *p2_driver.py*

Pose Mode Utility - *pose_mode.py*

This is a utility for generating .csv files that define a robot’s movement. It shuts off all the body servos and then runs an infinite loop so you can essentially “keyframe” the robot’s poses. This loop will run until you press Ctrl+C or type “exit” into the prompt. It will output a .csv file containing timings, body joint positions, and head joint positions that can be used by *p2_driver.py*. The script will ask you to input a timing for each pose, but you can leave this blank and it will use a default timing of 750ms.

Tip: I personally find it easier to stick with the default timings and then edit them later
Tip: You can directly edit .csv files in a text editor or Microsoft Excel

Warning: This script only shuts off the body servos - **never forcibly move the head**

Usage:

```
>> python3 pose_mode.py [-h] -o output_file [-s]
```

Example:

```
>> python3 pose_mode.py -o my_movement
```

- Runs the keyframing loop
- Outputs *my_movement.csv* upon completion

Example:

```
>> python3 pose_mode.py -h
```

- Displays usage information for *pose_mode.py*

Movement Demo - *movement_demo.py*

This is a simple demo that shows off a few different movements. This does not read any movement files and instead relies on setting the motor positions and timings directly. Additionally, it demonstrates printing all the information from the servo and shutting off power to the servos at the end.

Usage:

```
>> python3 movement_demo.py
```

Shutdown Servos - *shutdown_servos.py*

This script can be run to ensure that all servos are shut down. Remember that the last step of any script should be to shutdown power to all servos to prevent overheating or burning out.

Usage:

```
>> python3 shutdown_servos.py
```

Print Demo - *print_test.py*

This script shows off the function that can be used to print out all the position, temperature, and voltage data of the servos. There are two calls to the function that prints all the data out, one with the verbose flag, and one without.

Usage:

```
>> python3 print_test.py
```