Project 4: Motor Control

Project 4: Motor Driver Control

Four ducted fans for our hovercrafts:

- One lift fan: unidirectional control
- Three lateral fans: bidirectional control

Component 1: Circuit

- H-bridge to power
- H-bridge to fans

- H-bridge to Teensy
  - For each fan: PWM magnitude and 2 direction control signals
    - Lift fan: hard-wire direction to push air into the lower chamber
  - Teensy power and ground

Be careful with direct battery power!

Component 2: Supporting Types/Implementation

Top of program:

```
// Promise that we will implement this function later void fsm_step();
// Create a task that will be executed once per 50 ms
PeriodicAction fsm_task(50, fsm_step);

// Gains to be used for reverse thrust
const float FAN_GAIN[] = {1.0, 1.0, 1.0};
```

Loop:

```
void loop()
{
       // Check to see if it is time to execute the fsm_task
       fsm_task.step();
}
```

Component 3: Interface Functions

```
int16_t clip(float value, float min_value,
             float max_value)

void set_lift_fan_magnitude(float magnitude)

void set_lateral_fan_magnitudes(float magnitude[3])
```

Component 4: Finite State Machine

fsm_step() will implement the following behavior:

Depending on switch state:

- Central fan: Ramp up, then down
- Lateral fans:
  - Ramp left up, then down,
  - Ramp right up, then down
  - Ramp back up, then down

Coding

- fsm_step():
  - Called once every 50ms
  - Does not include for, while or sleep. Instead, rely on the fact that the function will be called regularly
- Make sure that each function that you implement does exactly what the specification says & no more
- Stick to the documentation specification

New Hardware for Today

- Dual H-Bridge modules

Be careful with the battery power! (go slow)