UNIVERSITY of WASHINGTON

ARES Controls / Electronics Documentation

Ethan Armstrong (CE Lead)
Jimmy Fowler (RE)
Oliver Buehrer

Supplemental Links

ARES Requirements and Deliverables

https://docs.google.com/spreadsheets/d/1TXDLYTX0K8XlAvZRNpm9V4OVancLeH42qMZSyQYIy2c/edit?gid=0#gid=0

ARES Hardware / Firmware / Software Github

https://github.com/explosion33/ARES

Radio Board Github

https://github.com/stars/explosion33/lists/cc1200-radio-project

SARP HomePage

https://sarpuw.com/

Ethan Armstrong Personal

https://ethan.armstronglabs.net/

https://www.linkedin.com/in/warmst/

Jimmy Fowler Personal

https://www.linkedin.com/in/fowler-james/

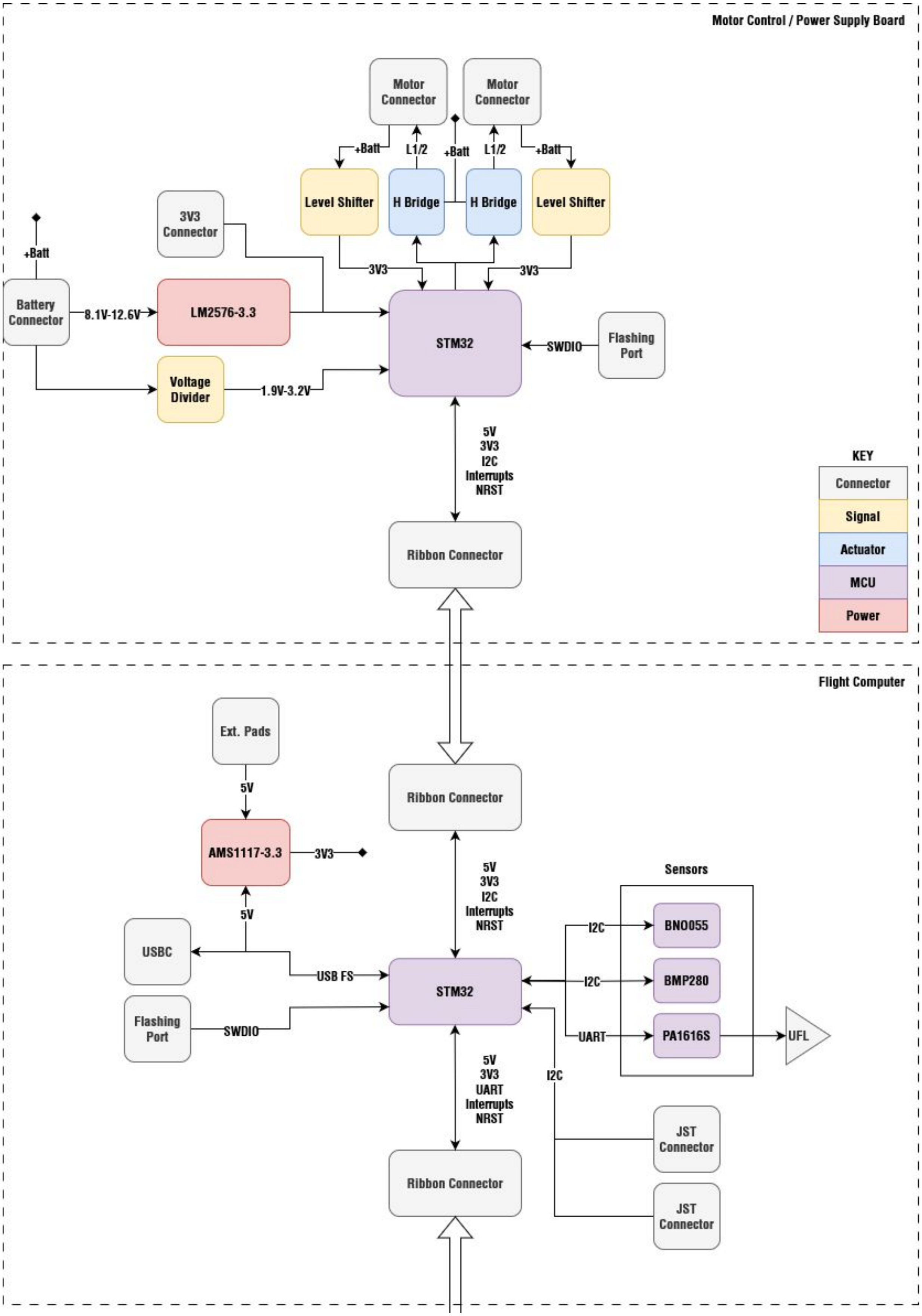
Oliver Buehrer Personal

https://www.linkedin.com/in/oliver-buehrer/

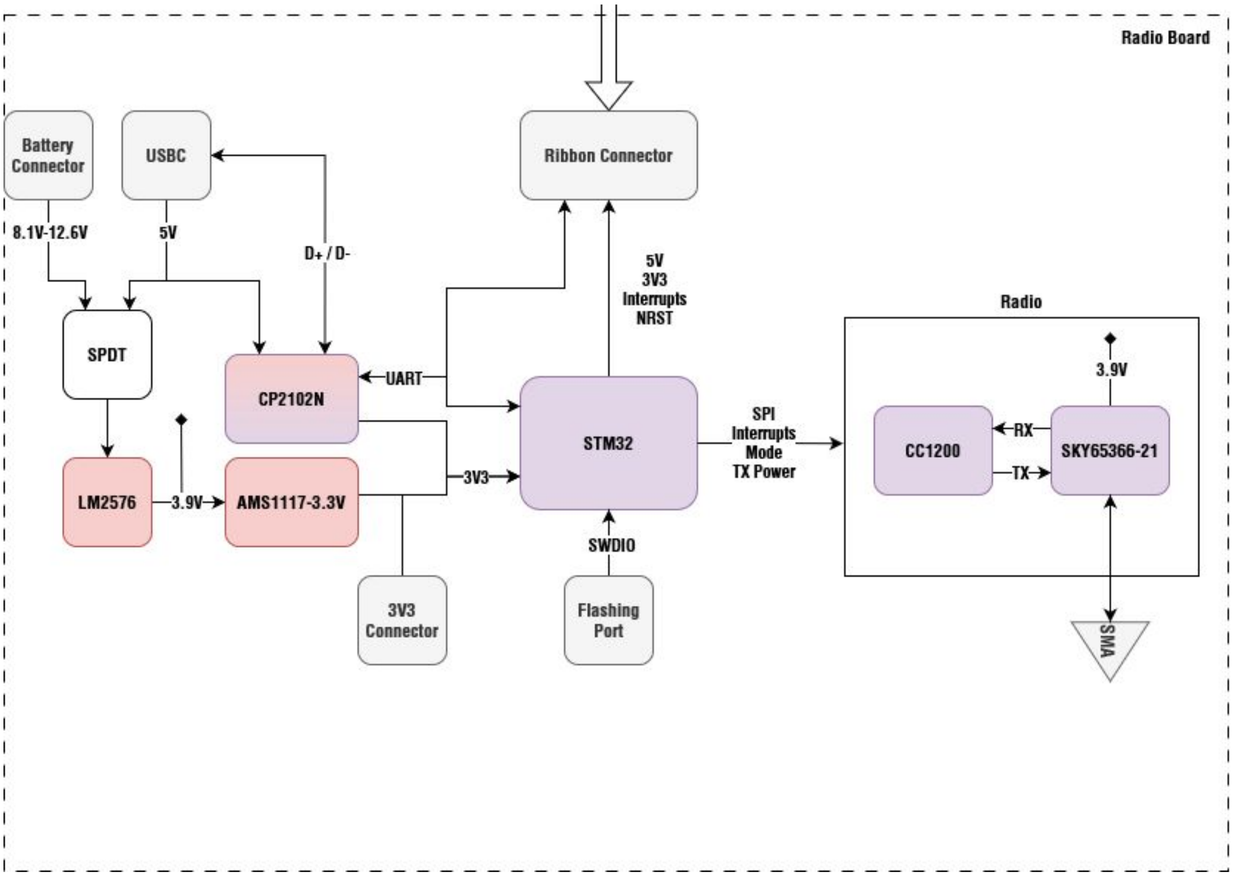
Table of Contents

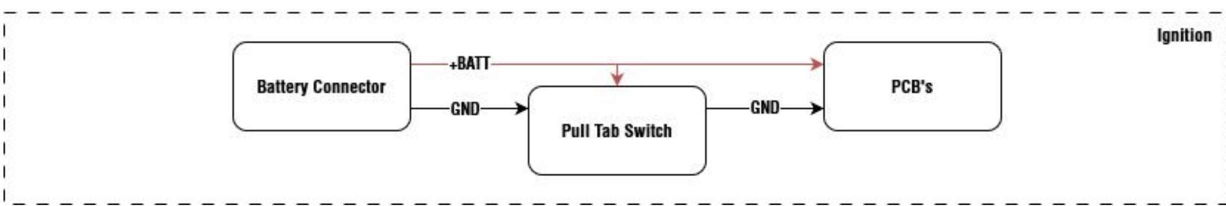
Hardware

Hardware Block Diagram



Hardware Block Diagram





Current Draw Estimations

Max (Over)Estimated Active Current Draw

Motors	500	mA x2	
Radio	500	mA	
PCB	150	mA x3	* STM32, Sensors, Peripherals, LDOs, etc.
Total	~2000	mA	

Max (Over)Estimated Idle Current Draw

Radio	500	mA	* Still 500mA TX current, but only on ACK's. Idle is expected to be >1h and <100 commands so generally low power
PCB	150	mA x3	
Total	950	mA	

Max (Over)Estimated Pad Current Draw

Radio	50	mA
PCB	100	mA x3
Total	350	mA

Time Per State

Active	0.5	h	1000	mAh
Idle	0.02	h	20	mAh
Pad	3	h	950	mAh

Total Expected milliAmp hours

1970mAh

Battery Selection

Power Requirements

13V max (due to motors)

6V min (due to buck converters)

2A max active current draw

12V nominal voltage

18650s (due to competition regulations)

2.5V - 4.2V

3.7V nominal

2200mAh - 3500mAh (depending on manufacturer)

3S1P

7.5V - 12.6V

11.1V nominal

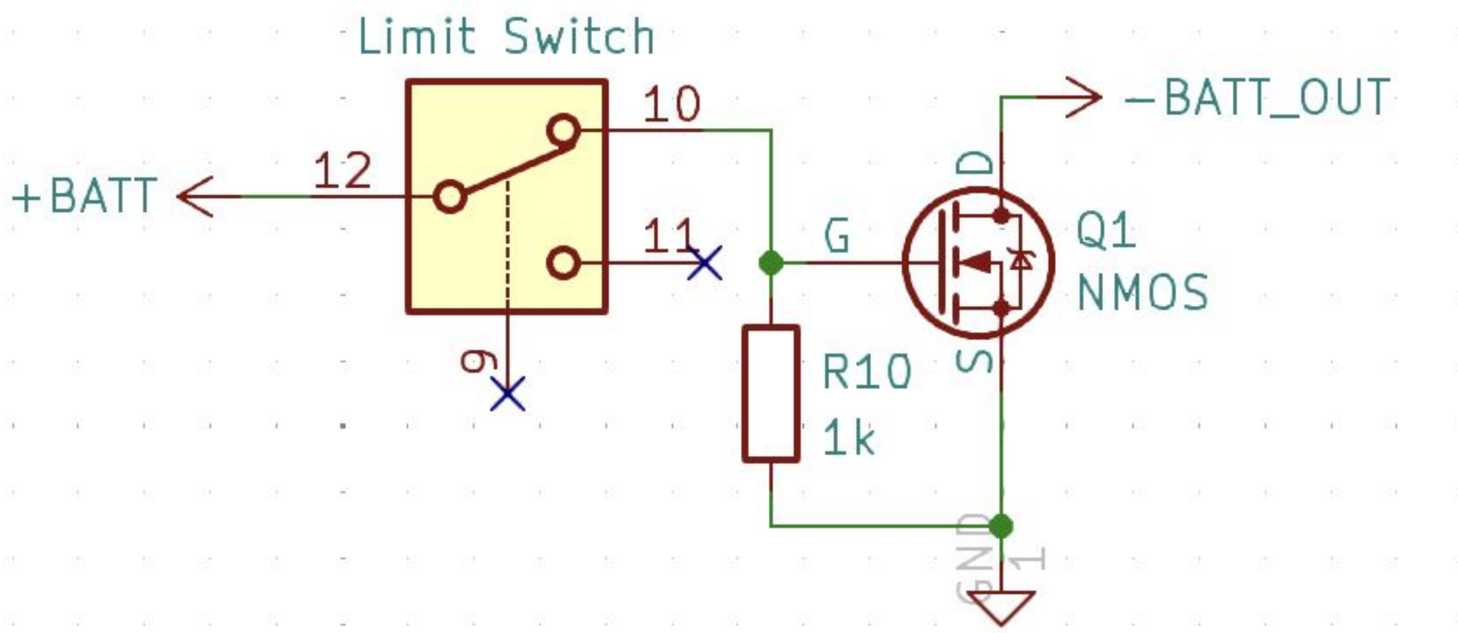
2200mAh

3S2P / 3S3P

Depending on space / weight requirements and final current draw measurements

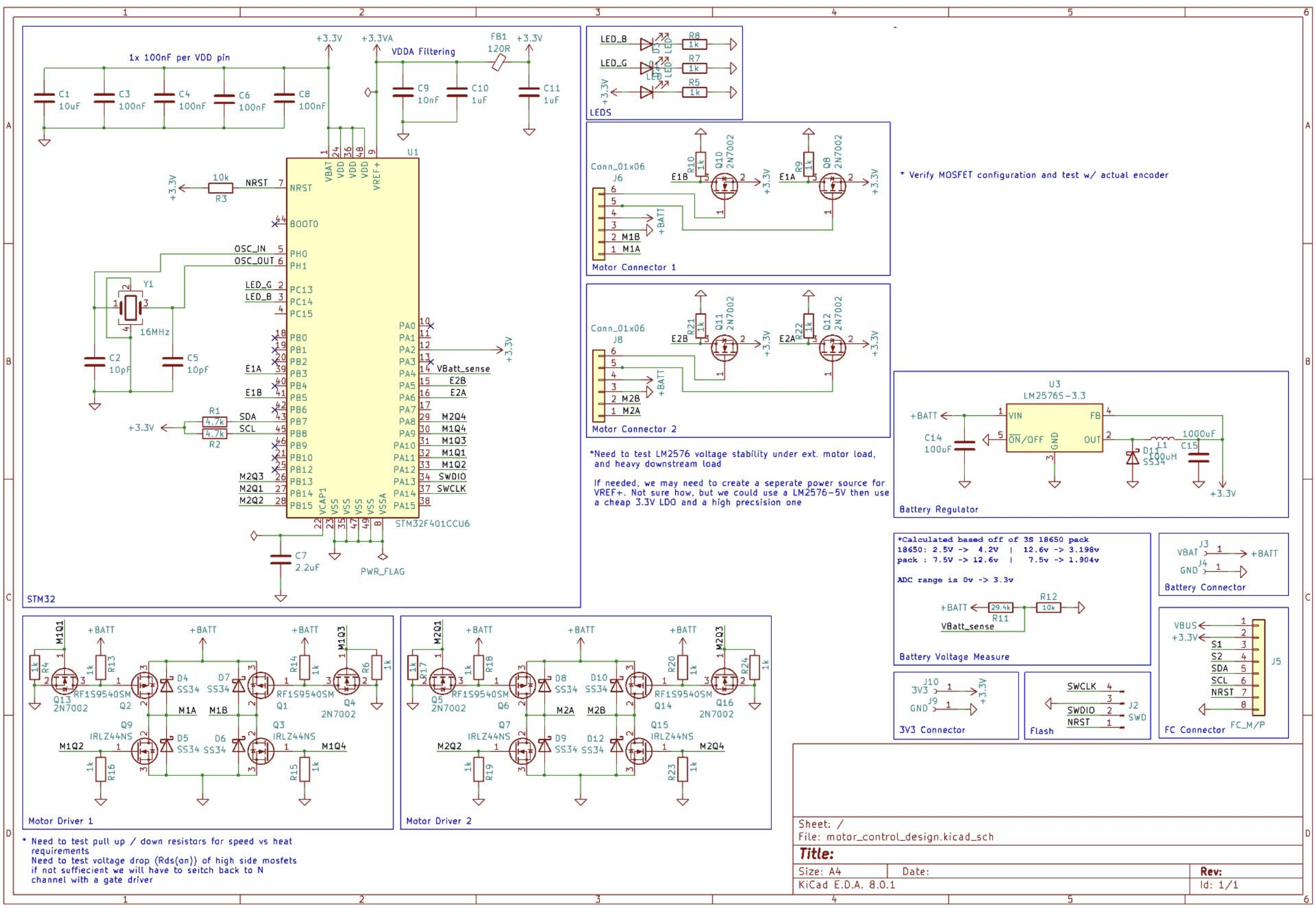
More batteries in parallel is ideal as our nominal voltage stays near 12V for longer giving greater control authority

Pull Tab Schematic



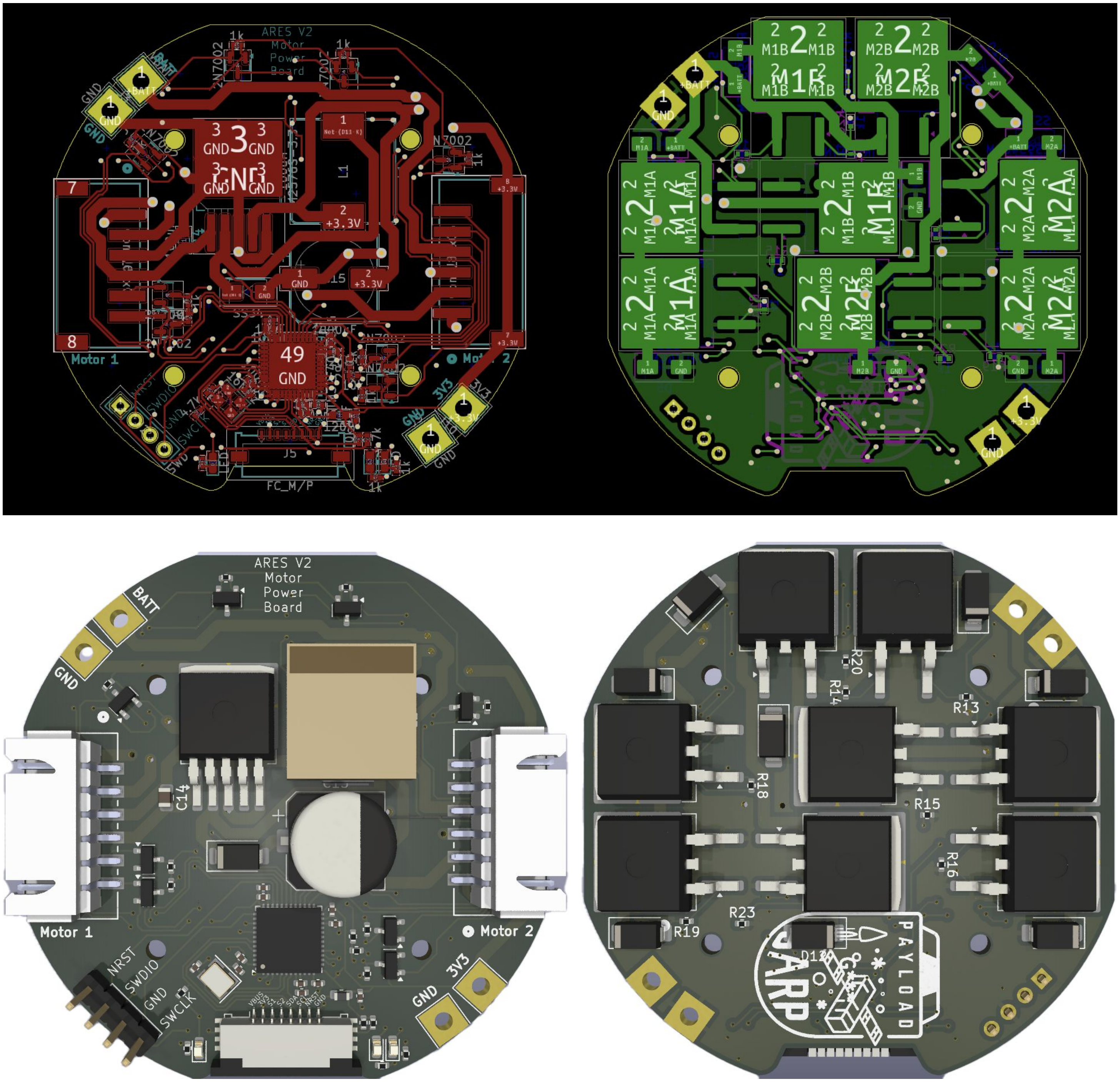
- Limit Switch is held in the closed position (which is disconnected) by a metal rod acting as a low side switch.
- N-Mosfet is used to handle high current requirements
- When rod is removed the NC pin is closed again connecting ground to the rest of the circuit
- Limit switch will be mounted so tab is depressed in the X/Y planes
    - Allow tab to be pulled out from side of rocket
    - Reduce chance of shock / vibrations triggering switch

MCPS* Schematic



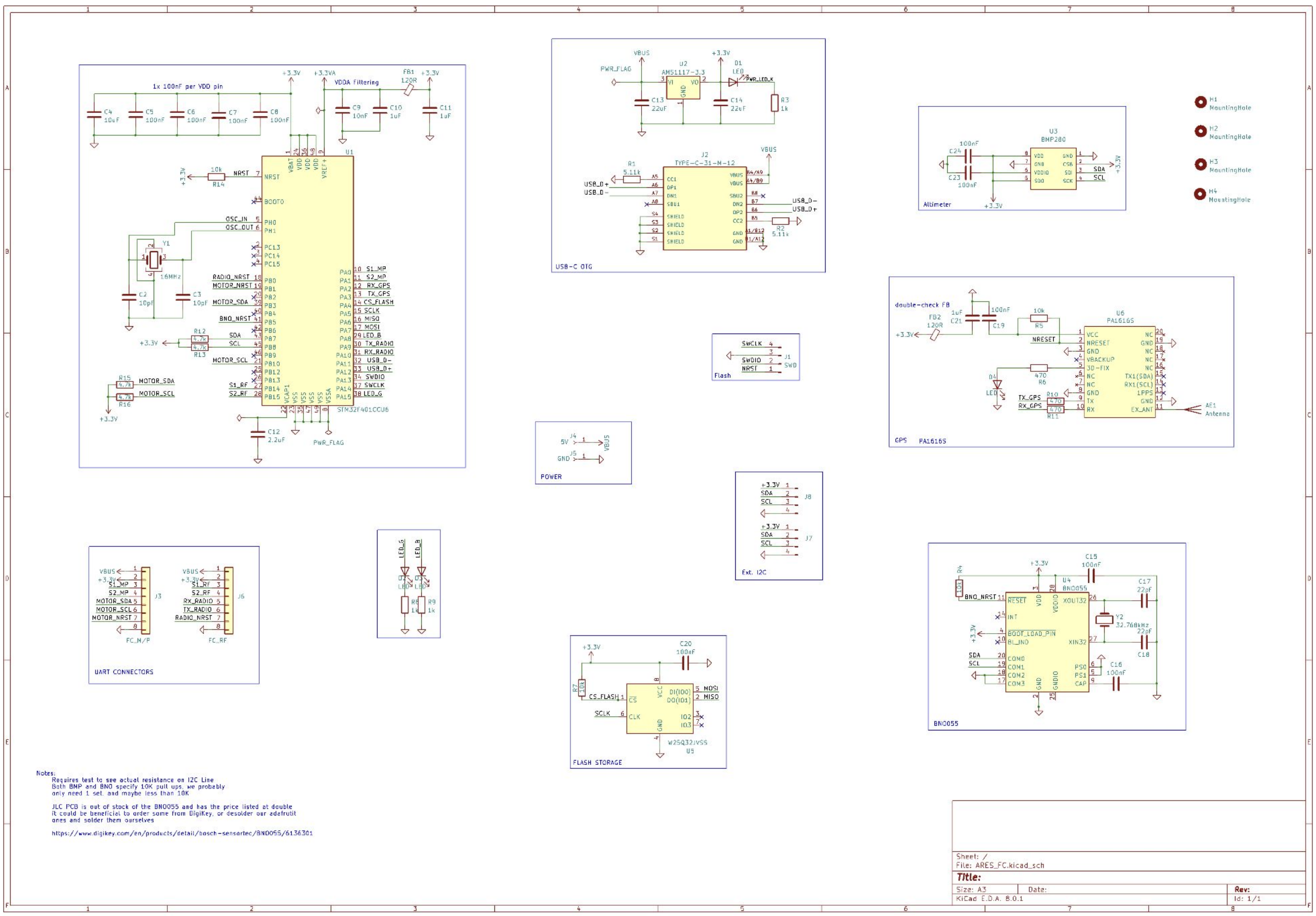
* Motor Control / Power Supply

MCPS* Routing + Render

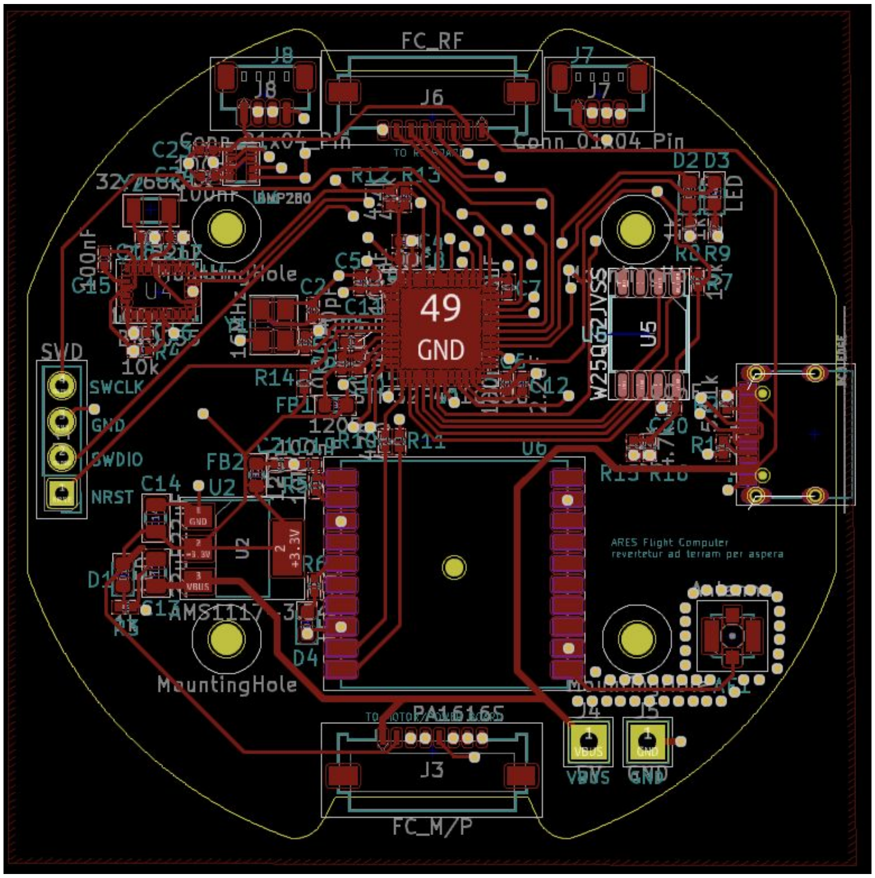


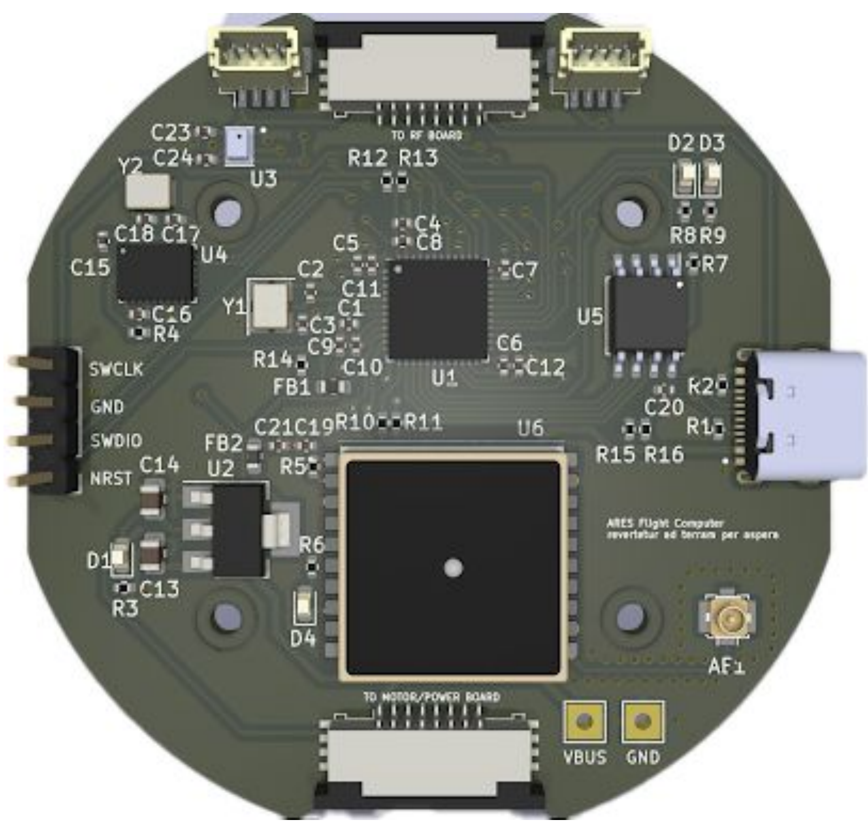
* Motor Control / Power Supply

Flight Computer Schematic

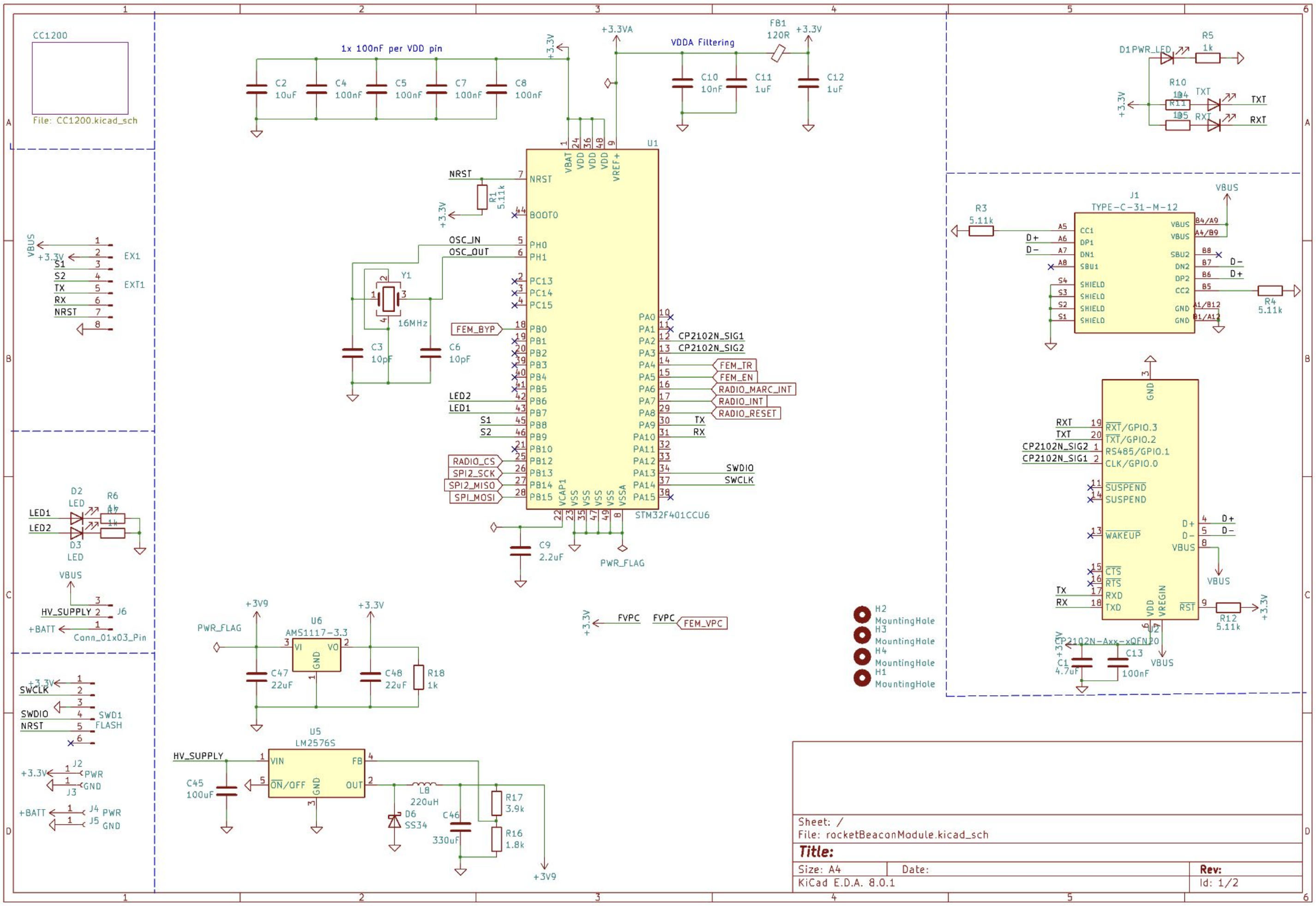


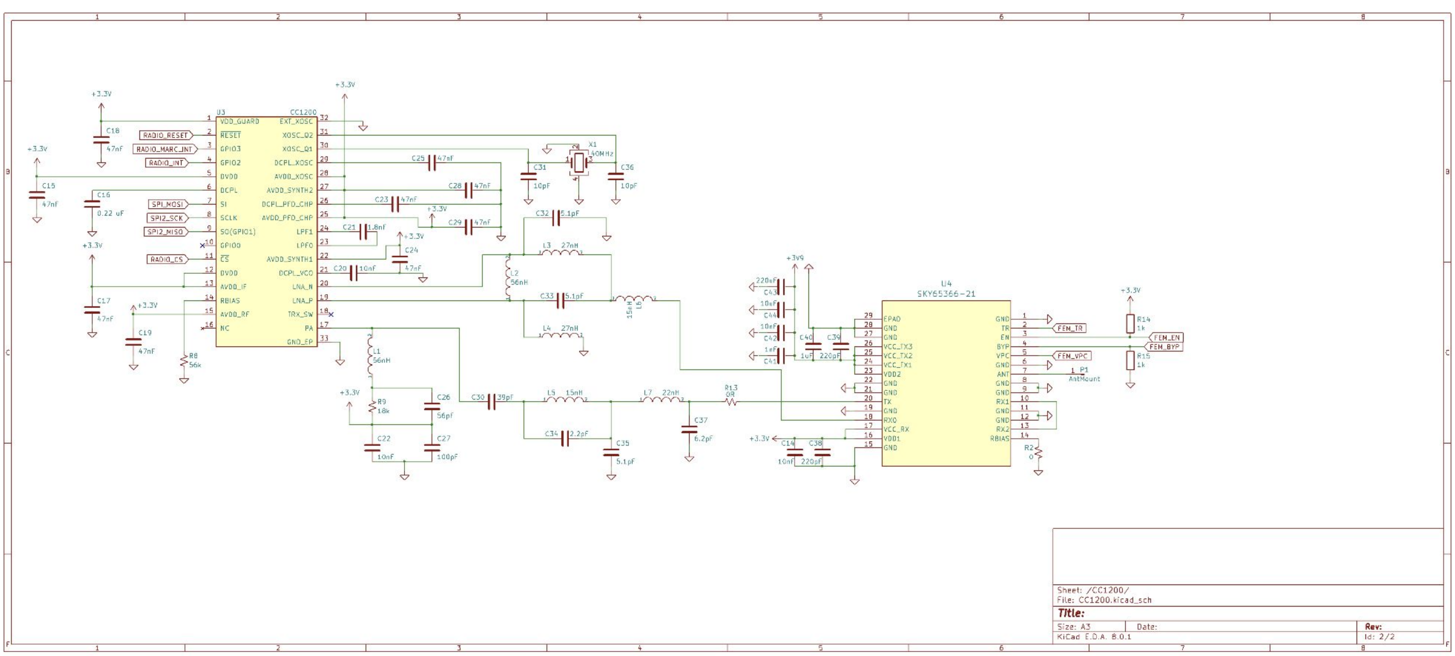
Flight Computer Routing + Render



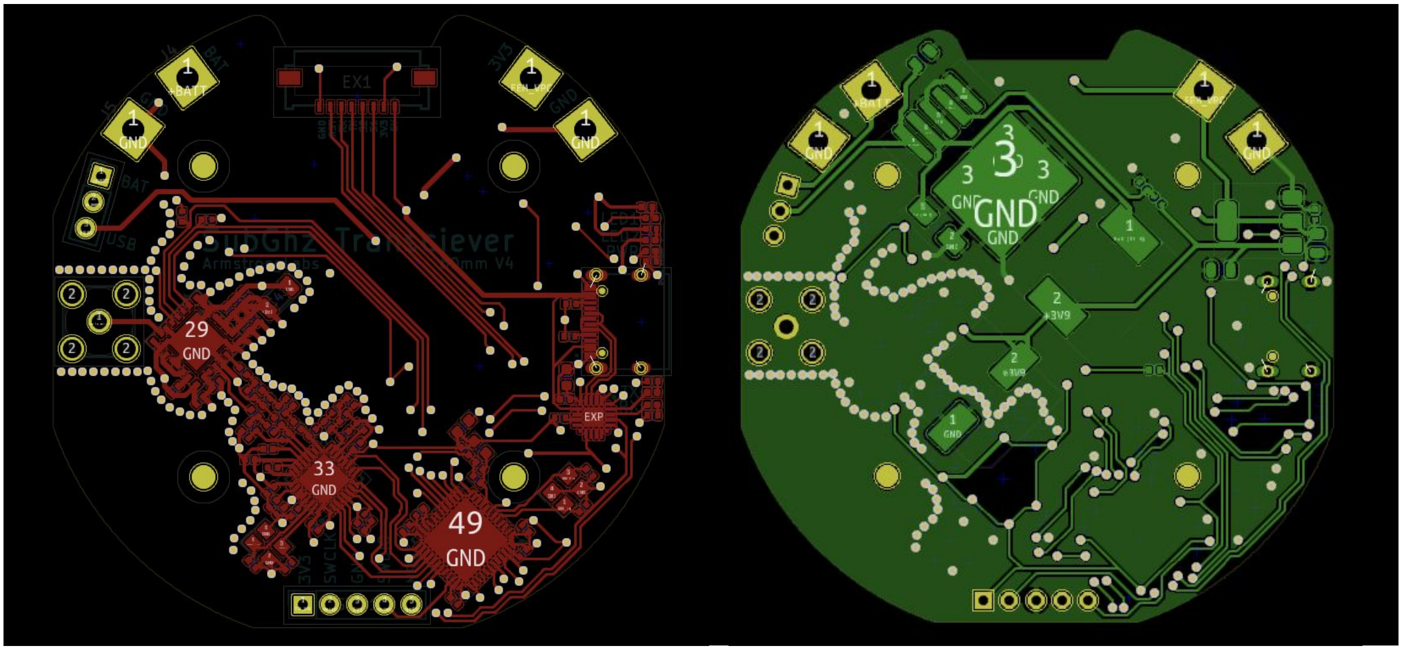


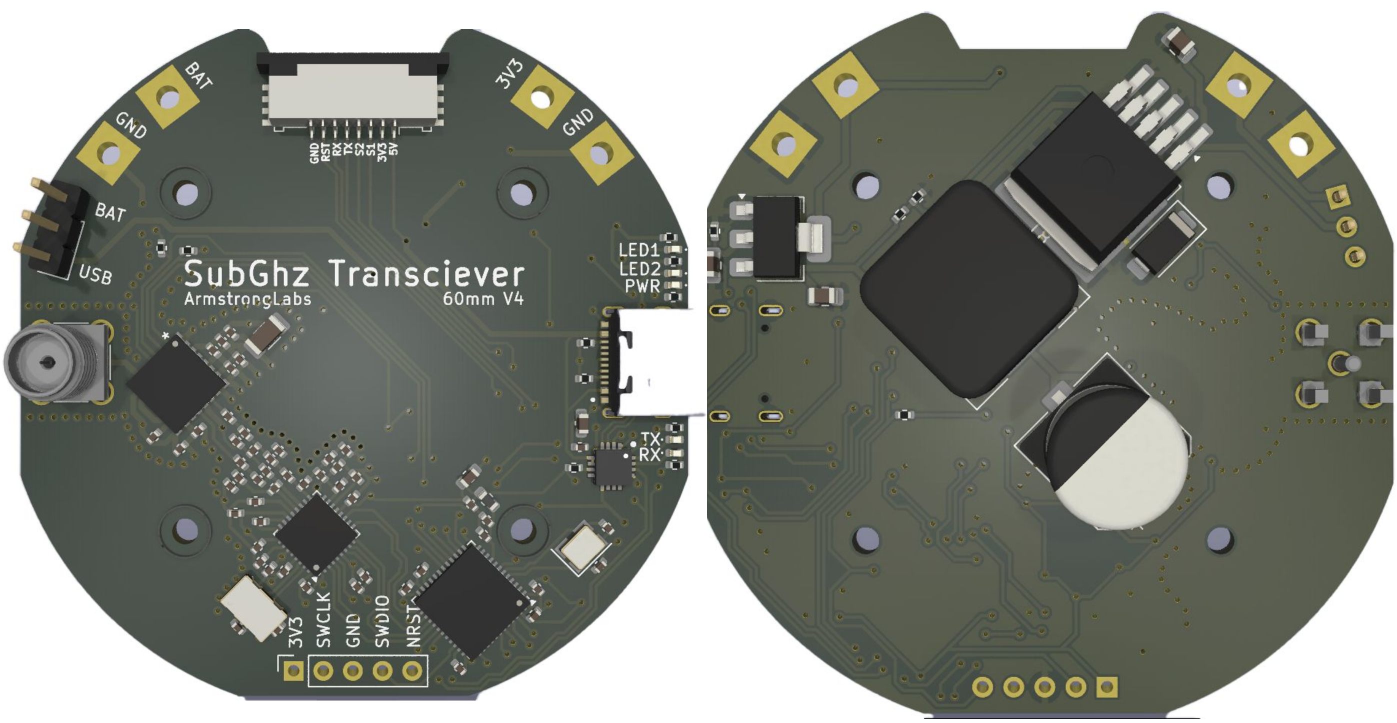
RF Schematic



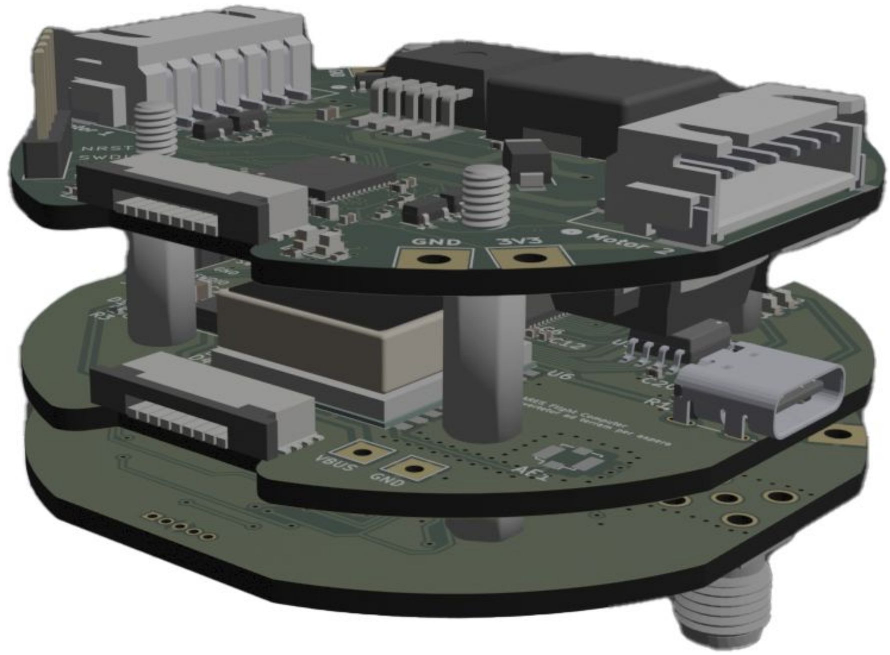


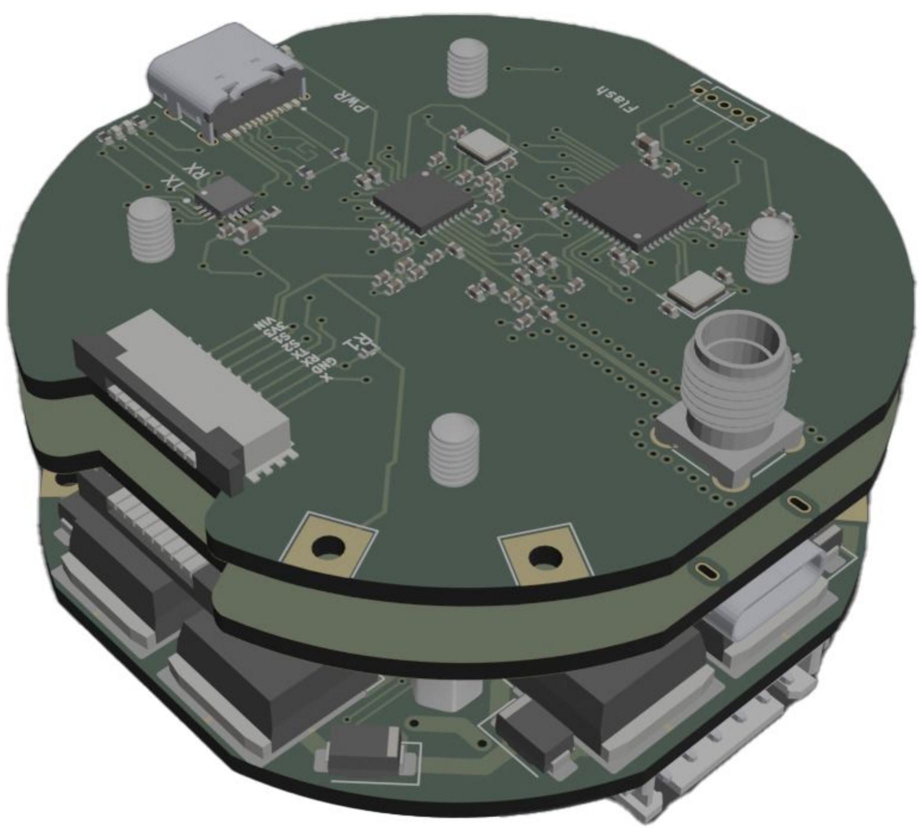
Radio Routing + Render





PCB Stack Render





PCB Descriptions

Flight Computer

- Reads sensors
- Stores data
- Interfaces with both other boards
- Kalman Filters
- Main PID Controller stack

Motor Control / Power Supply

- Regulates battery voltage down to 3.3V @3A
- Monitors Battery Voltage
- Sets each motors exact needed position
    - PID Control
    - Input from Flight Computer

Radio

- Regulates Battery Voltage / USB Supply to 3.9V @1A
- Handles Radio Link
    - Encoding / Decoding (434MHz GFSK @100kbps)
    - Checksums
    - Keep Alive
    - COBS
- Interfaces with Flight Computer to exchange packets
- Interfaces with laptop for ground station downlink

UNIVERSITY of WASHINGTON

Data Link and Ground Systems

Modified OSI

7	APPLICATION	Packet Type and Subtype “C”: Command, “T”: Telemetry. 0-255 subtype byte, X bytes datastream
4	TRANSPORT	Ack + Retries
2	DATA LINK	Start Byte (7 bits) (1 bit if ACK needed), COBS, Checksum
1	PHYSICAL	434Mhz (G)FSK

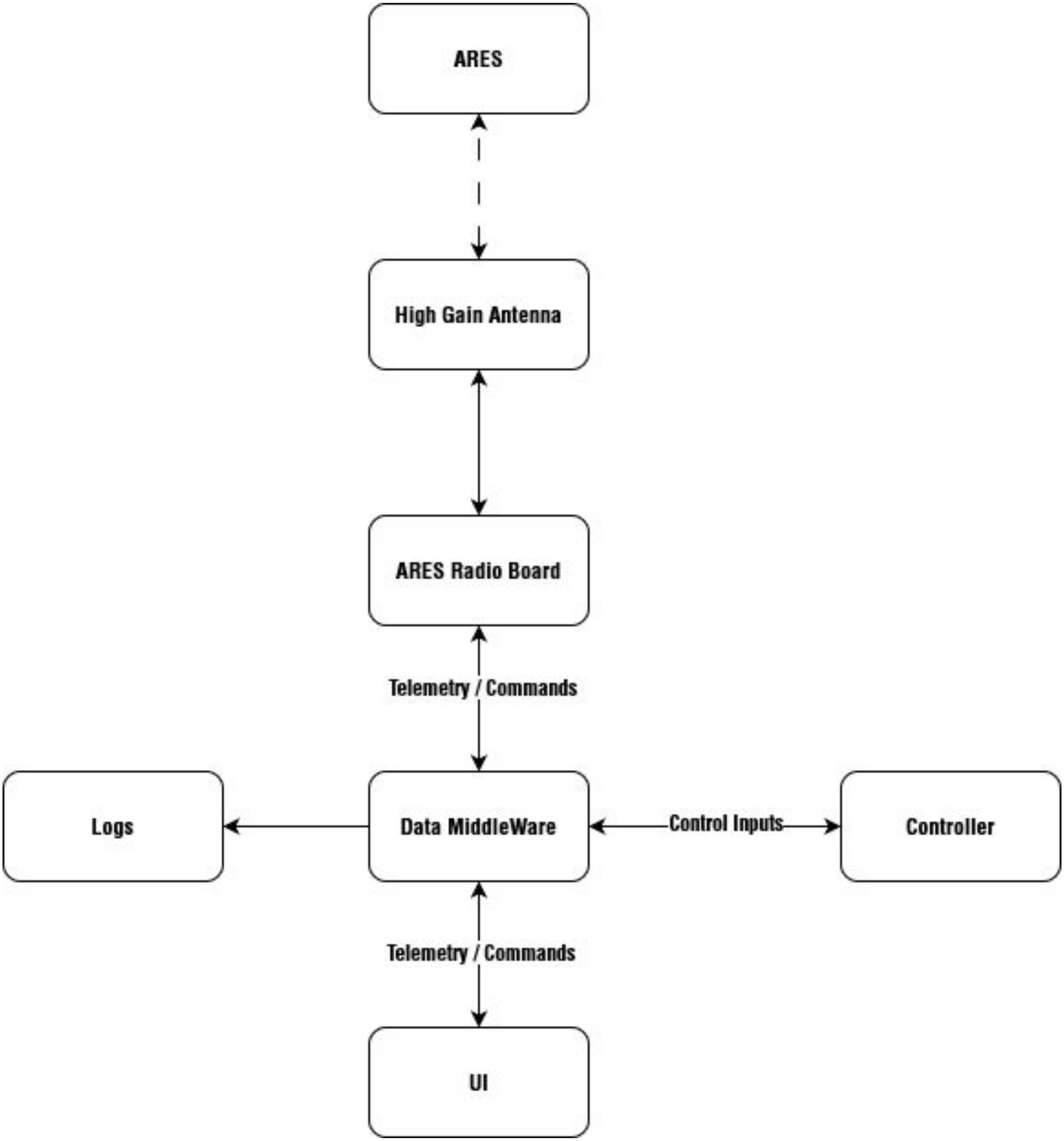
Modified OSI

1. **Physical** (Handled by CC1200 chip internally)
    - 434MHz (G)FSK @100kbps
    - Preamble + Sync Word
    - Length Byte
    - CRC 16-bit
2. **Data Link** (Handled by Radio STM32)
    - Start Byte 1101011 + (0|1) to indicate if an ACK is expected
    - Consistent Overhead byte stuffing to remove start byte
    - Packet length
    - Additional CRC checksum

4. **Transport** (Handled by Radio STM32)
    - If start byte indicates an ACK, wait for / send ACK. On failure add to a retry queue up to 3 times
    - ACK packet: “C0”

7. **Application** (Handled by Flight Computer / Ground Station)
    - Packets start with two bytes indicating packet (sub)type
        - “C” for command “T” Telemetry
        - Followed by “0-255” to indicate a preset packet type referenced from lookup table
    - Packets are then made up of X bytes of data following format found in lookup table

Ground System



Ground System

Data Middleware

- Rust program that abstracts radio communication into a REST API
- Telemetry logged to a file and made available as a time series
- Commands sent via POST request and packaged
- Radio interface over USB Serial
- Controller interface over USB Serial

UI

- React based front end
- Pulls data from backend for display in real-time charts
- Buttons for common commands
- Form / alternative input for custom commands

Controller

- Standard Xbox controller wired via USB
- Middleware packages joystick / button info into Command packets
- Used for manual flight

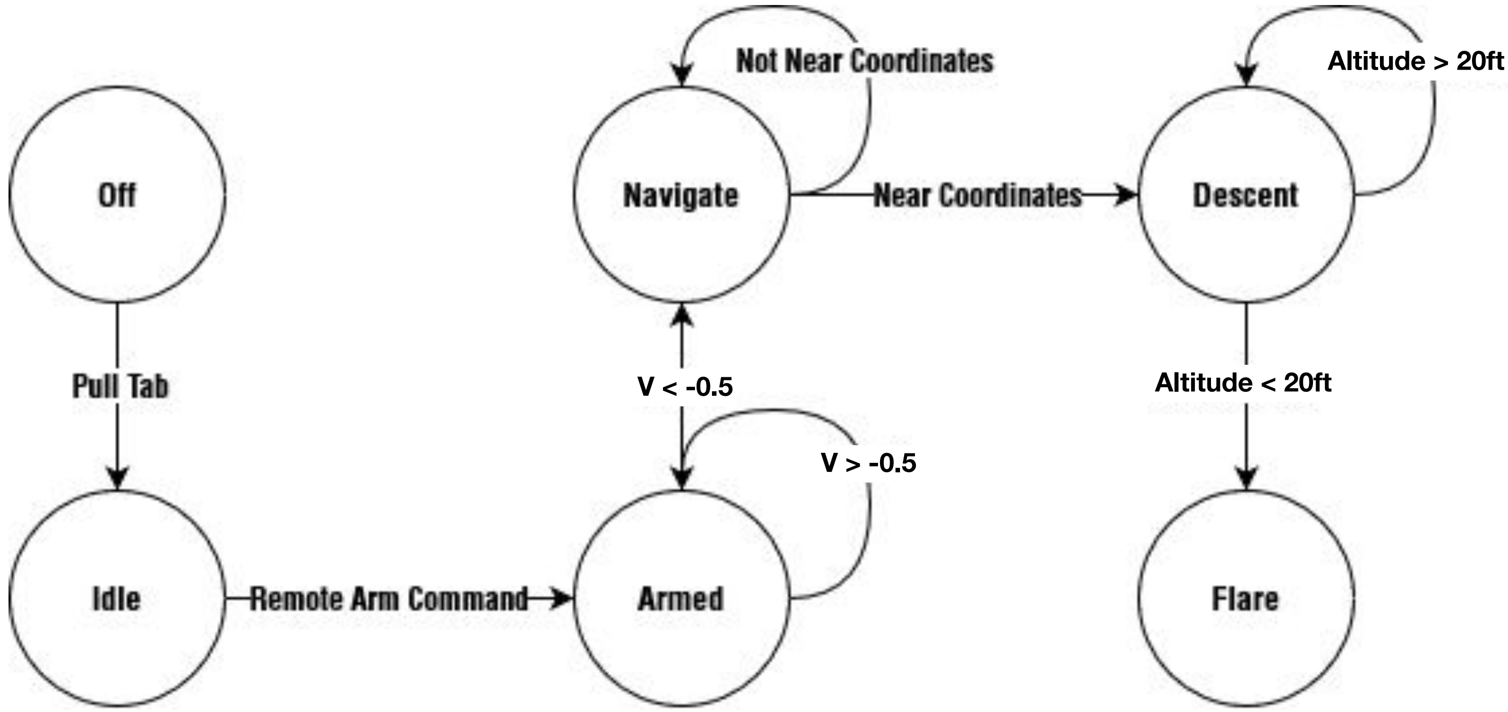
High Gain Antenna

- 70cm handheld YAGI

UNIVERSITY of WASHINGTON

Control Systems

Finite State Machine



Off -> Idle

Physical pull tab to connect battery to rest of system

Idle -> Armed

Remote command from Radio board "C1" + ACK

Armed -> Navigate

Average downward velocity is greater than 0.5 m/s (from Vertical Kalman Filter)

Ensures package is ejected and glider is deployed before actuation occurs

Navigate -> Descent

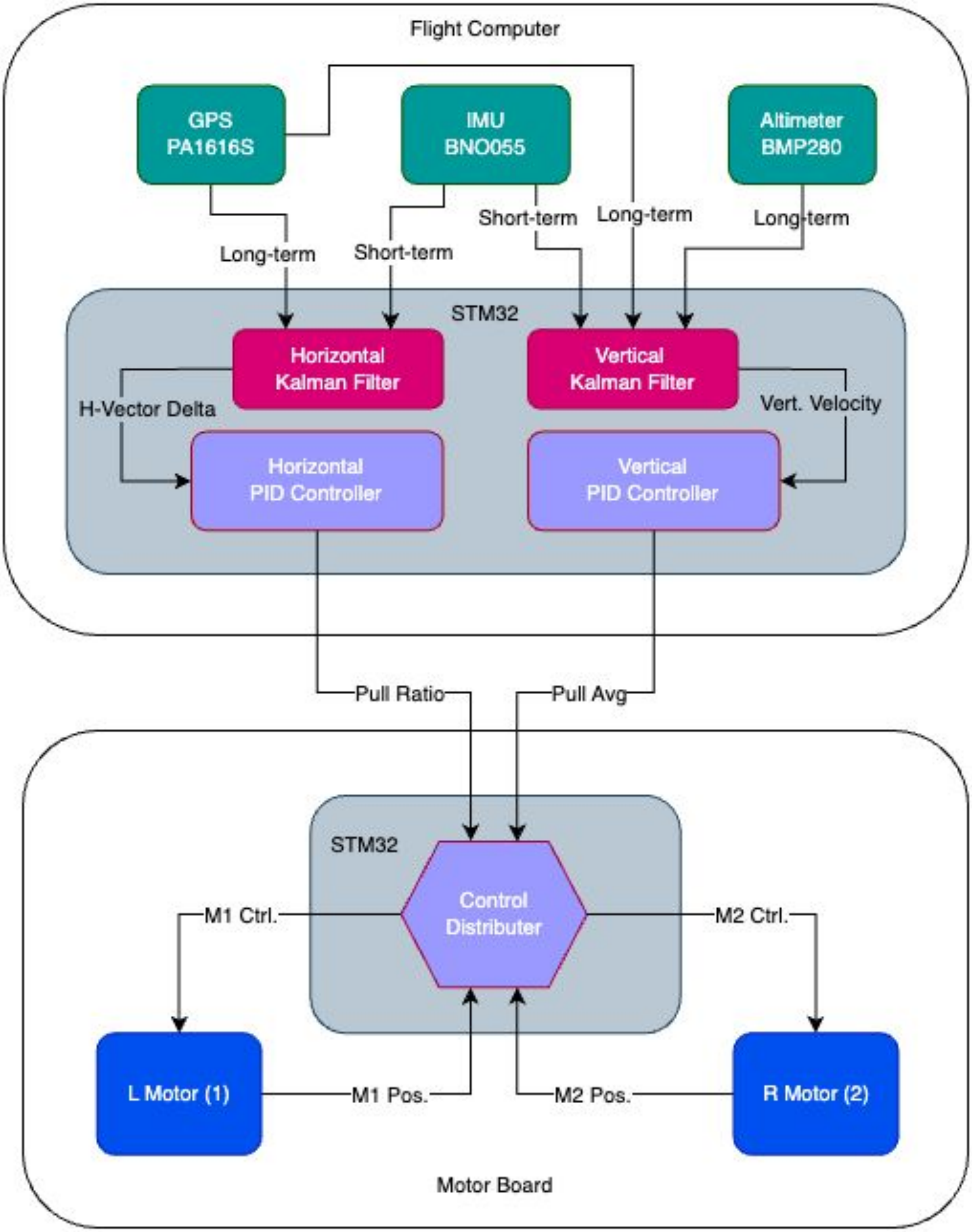
Once ARES has reached target destination begin Vertical Descent / Spiral Downward

Descent -> Flare

Once ARES is close to ground flare parachute for soft landing

End active control

Controller Stack



Controller Stack

Horizontal Kalman Filter

- Input:
  - GPS (measurement)
  - IMU (predict)
- Output:
  - Heading Vector (direction and speed)
  - GPS coordinates

Horizontal PID Controller

- Compares heading vector to Heading needed for GPS setpoint
- Outputs pull ratio between control lines
  - Determines turn left / turn right / go straight

Vertical Kalman Filter

- Input:
  - GPS (measurement)
  - Altimeter (measurement-ish)
  - IMU (predict)
- Output:
  - Vertical Speed
  - Altitude

Vertical PID Controller

- Compares Velocity to wanted velocity based on remain altitude
- Outputs Average pull on the control lines

Controller Stack

Control Distributor

- Input:
  - Line Ratio and Average from Flight Computer
- Calculates absolute motor positions
- Internal PID Controller:
  - Setpoint:
    - Motor Position (encoders) vs desired Position
  - Output:
    - Motor speed as a percent

Tuning Plan

- Flight Computer
  - Simulation
    - Matlab / Simulink ← In Progress
    - Get rough PID values for all controllers
  - Hardware in the Loop
    - Use previous flight data and generated data
    - Observe actuator response
    - Refine PID gains
  - Rocket Testbed launch
    - Validate control systems
    - Define control responses
    - Gather flight data
    - Adjust gains
  - Hardware in the loop
    - Use testbed data
  - Simulation
    - Re-verify with simulation
- Motor Controller (Control Distributor PID)
  - Manual tuning on ground to reach setpoint