D.E.A.R Drone

Group 16:

Dan Biller, CPE

Ellie Lane, CPE

Austin Perkins, EE

Raymond Chenoweth, EE

Project Description

- D.E.A.R Drone is an autonomous delivery drone that will deliver a package of ½ pounds.
- The D.E.A.R drone will use GPS, ultrasonic sensors, and a cameras to ensure the package makes it to the destination safely.
- Our mission is to help streamline supply chains to get people their products faster

Project Pros/Cons

Pros

- Fast Delivery
- More Efficient than Truck Delivery
- No human delivery person
- Fully electric
- Cheaper
- Can deliver to remote or inaccessible areas
- Multi-rotors are more maneuverable than fixed-wing aircraft

Cons

- Short range
- Single delivery per trip
- Low payload capacity
- Multi-rotors are less efficient than fixed wing aircraft
- Multi-rotors are slower than fixed wing aircraft

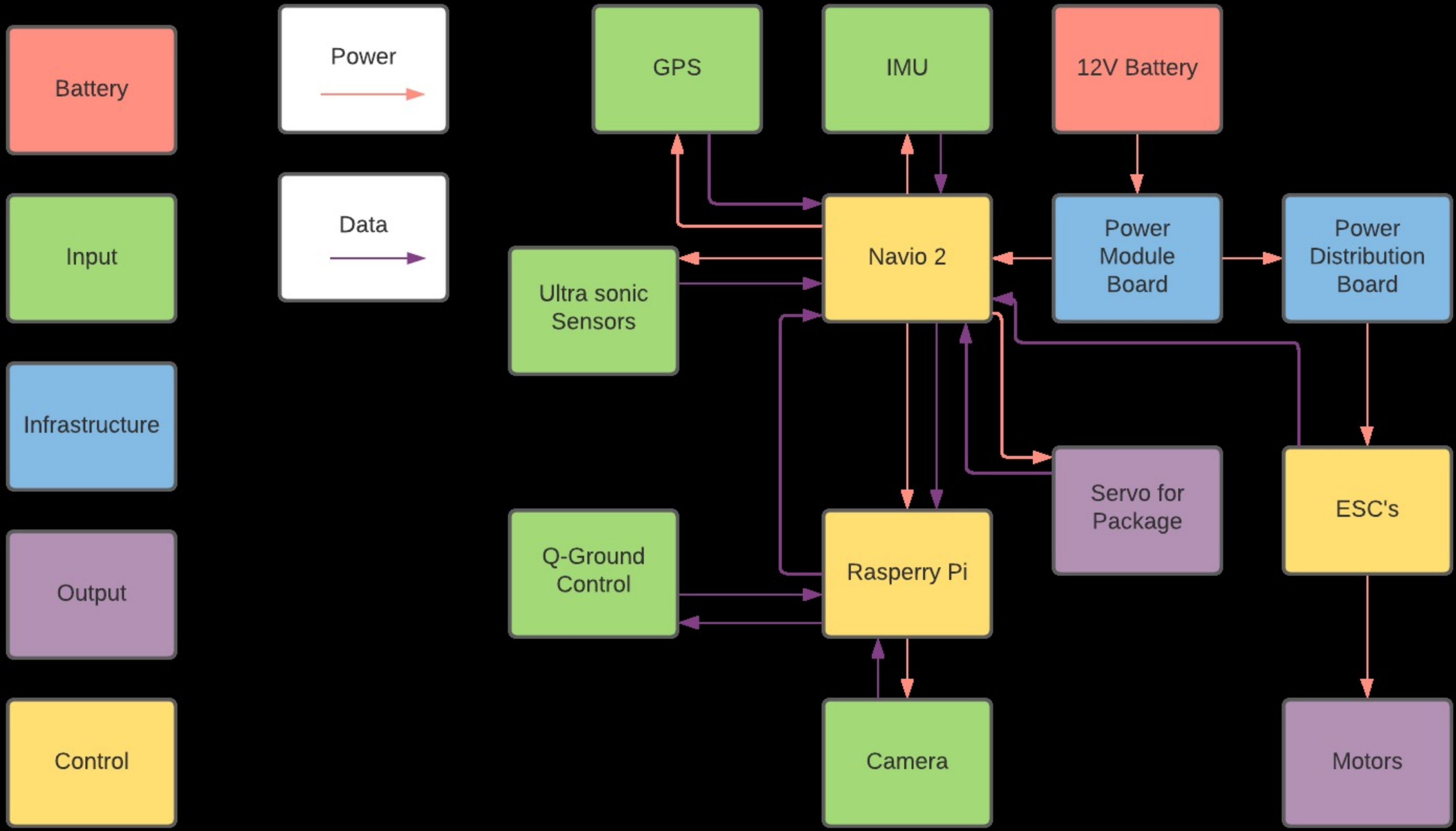
Specs/Requirements

Specifications	
Payload Weight	.5 Pounds
Battery Capacity	6000mAH
Flight time	20min(+-10 vertical)
Field of view	60 degree horizonal
Range of view	.5-10feet
Measuring Frequency	10Hz
Autopilot	GPS

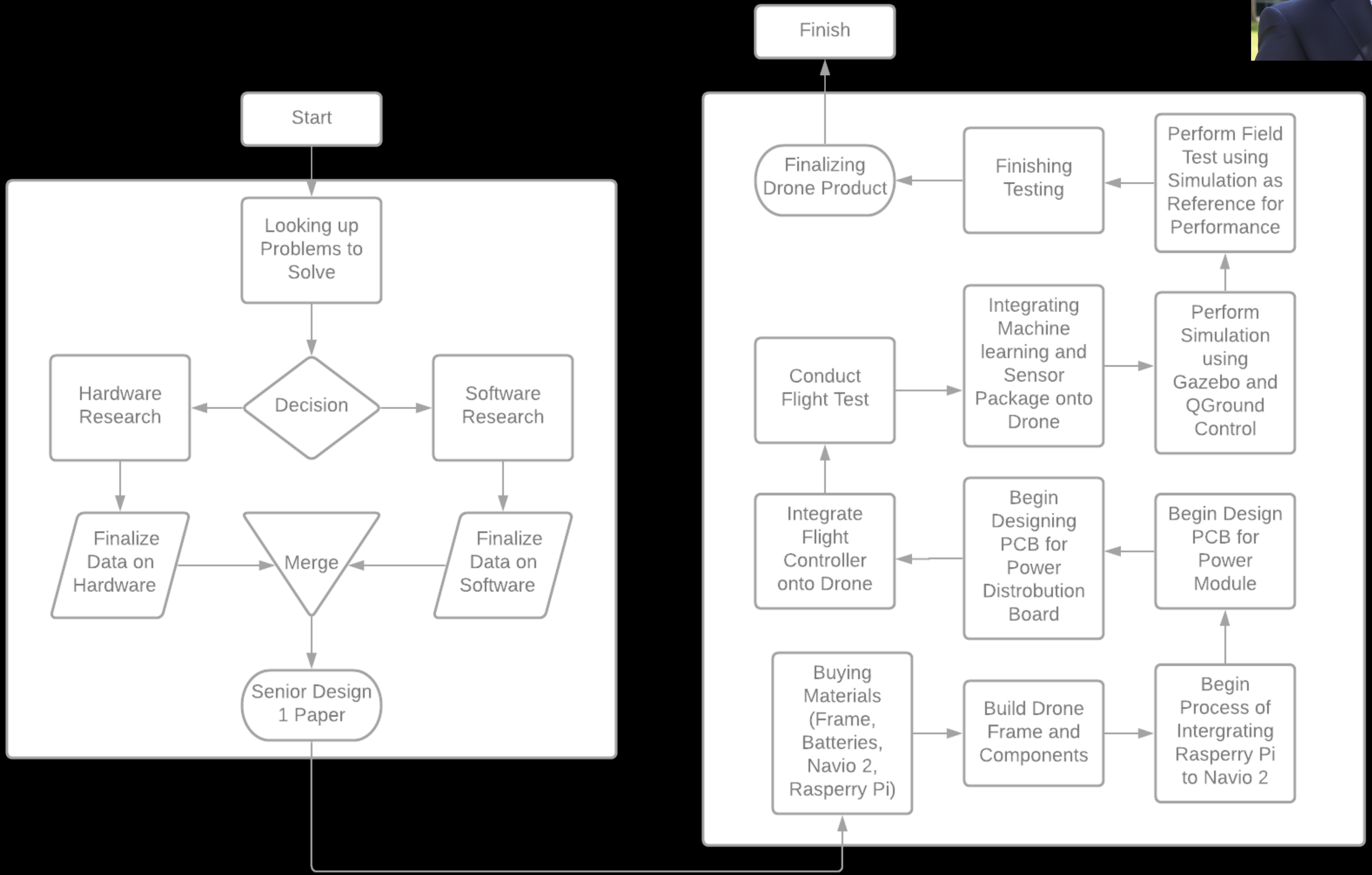
Standards

Standard	Component
IPC-2221 PCB Standard	PCB design and manufacturing
NFPA Electrical Safety Standards	Safety standards for electrical components
PEP8	Python style guide
IEEE standard 802.ac/n	Wi-Fi
Batteries IEEE 485-2010	Standard for Lead Acid and LiPo Batteries
FAA part 107 UAS	Drone Flight

Project & System Diagram



Block Diagram – Overall System or Project



Drone Design and Frame

- Our frame was chosen for cost as well as functionality
- Plastic when it breaks snaps apart, while carbon fiber splitters
- Carbon fiber also interferes with RF signal
- The cost was kept down using a bundle deal that included the ESCs, rotors, motors, and PDB

Battery

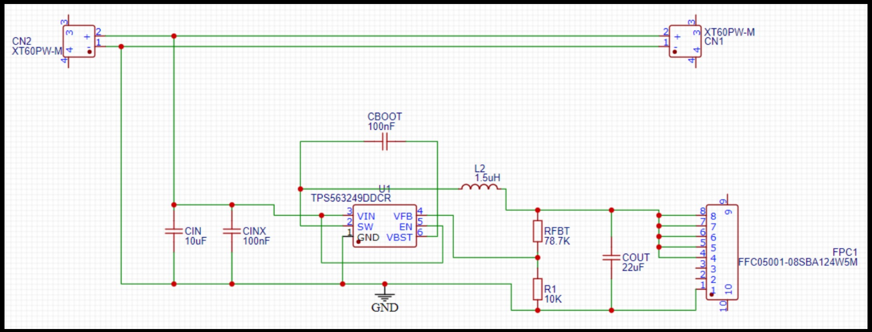
- We decided on the Turnigy nanotech 6000-mAh 3s LiPo
- This battery provided us, after utilizing the rate of consumption formula, that we could achieve a flight time of 20 minutes

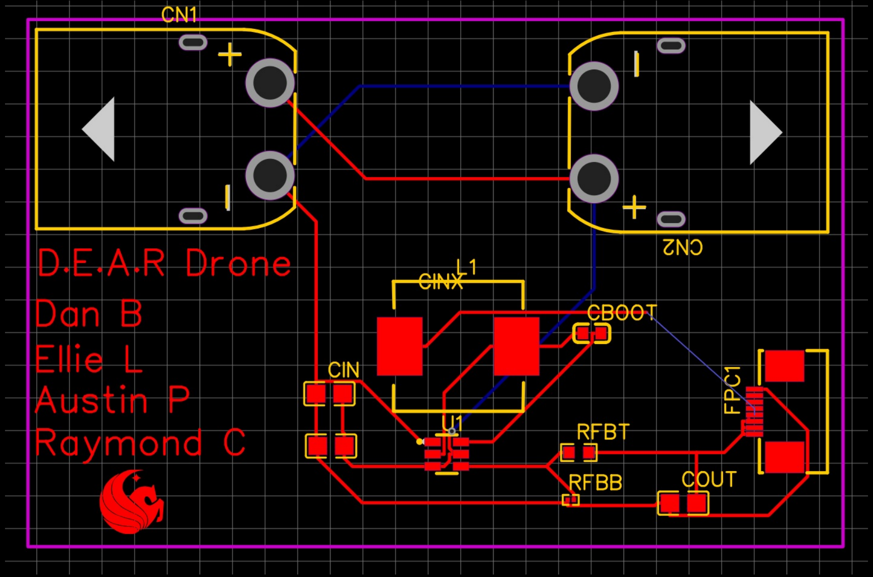
Consumption Formula:
(Milliamp hours/ avg amp draw) * 60 = amount of flight time in minutes

PCB Designs

PCB Design

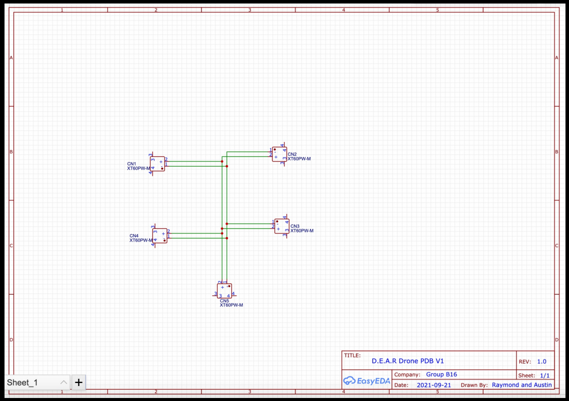
- Power Module
  - Sends 11V to our power distribution board
  - Steps down 11V to 5V and 2.5mA for the flight controller
  - XT60 to XT60 connection

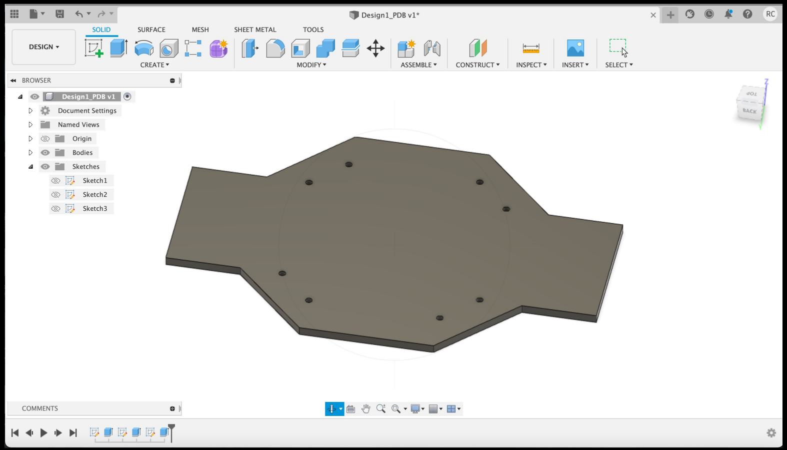


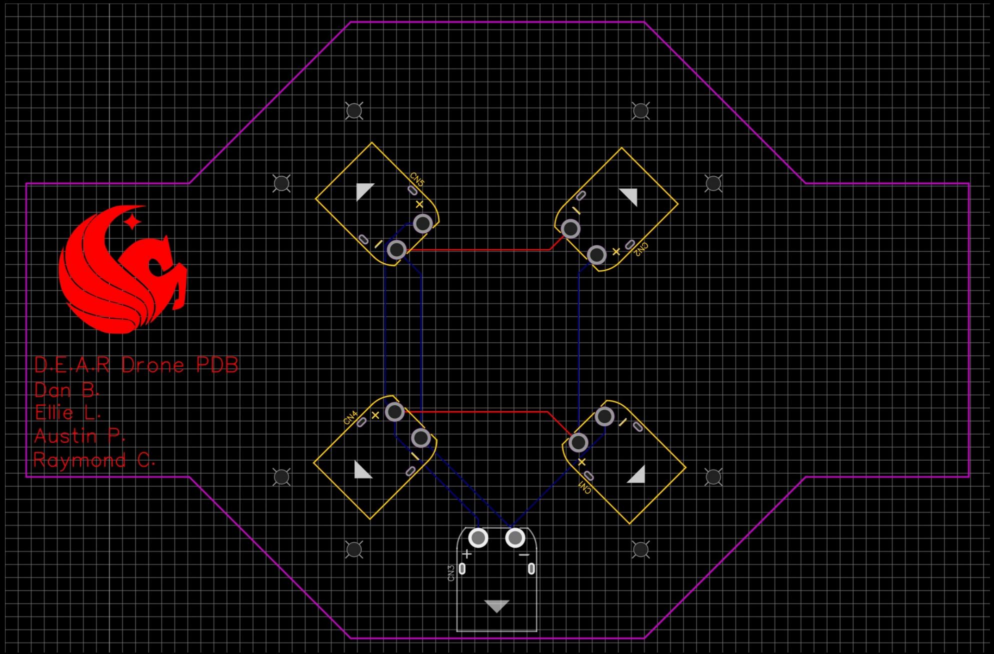


PCB Design

- Power Distribution Board
  - Send 11V evenly to all our ECS's

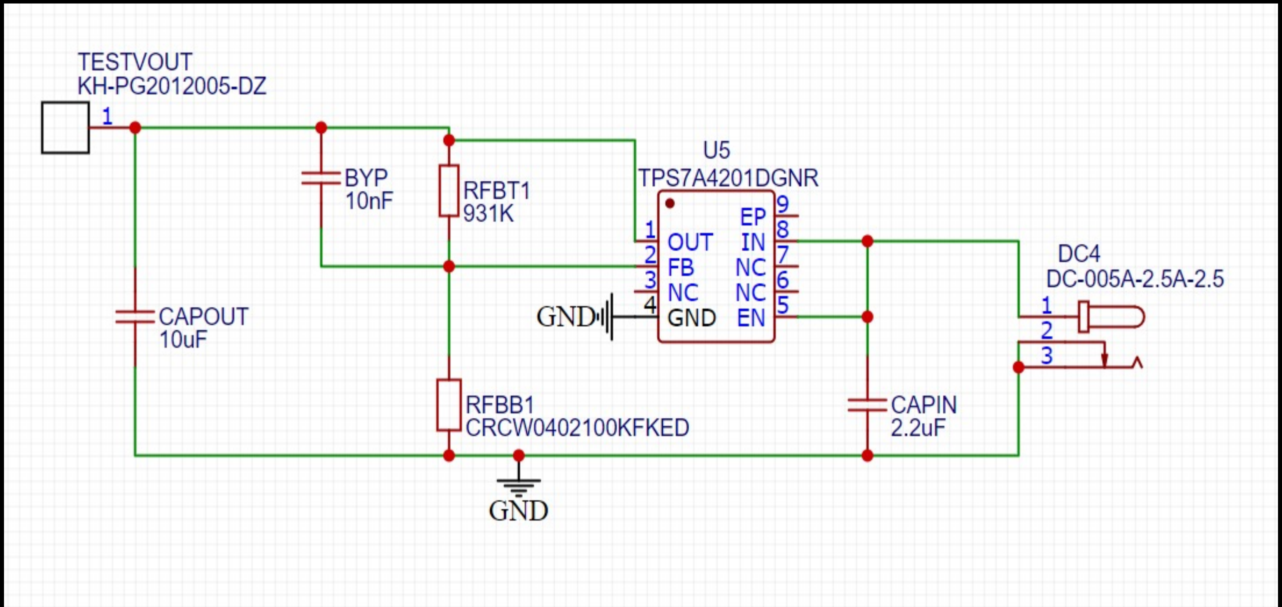


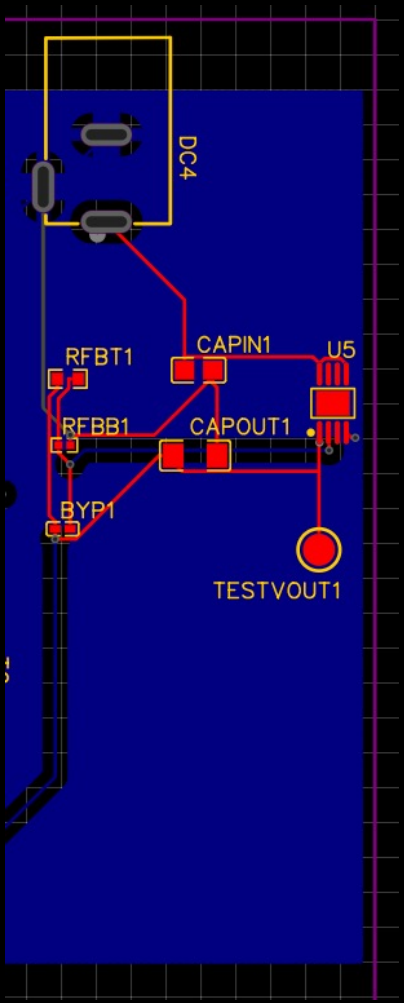




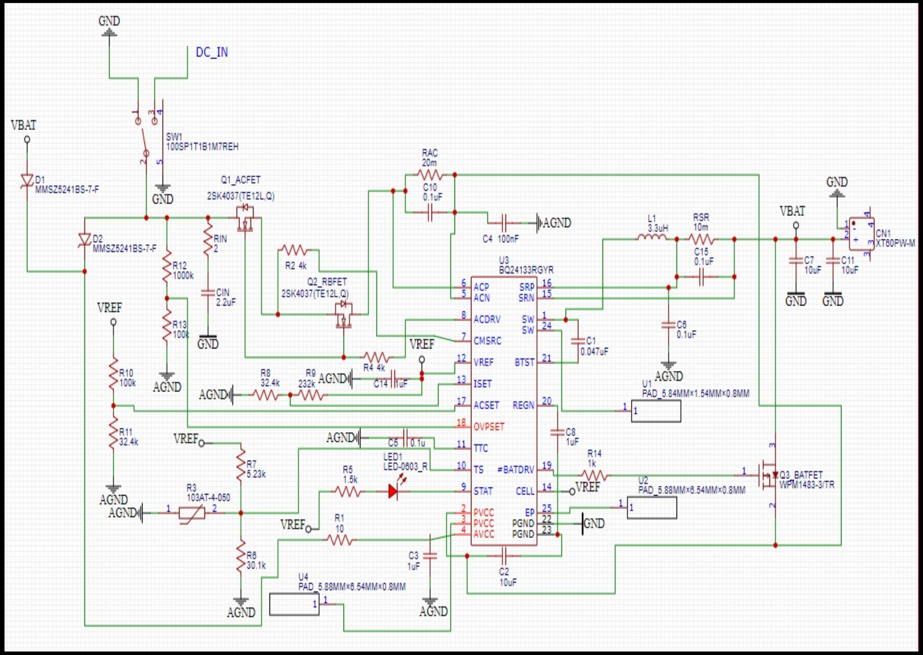
PCB Schematic [DC-DC Converter]

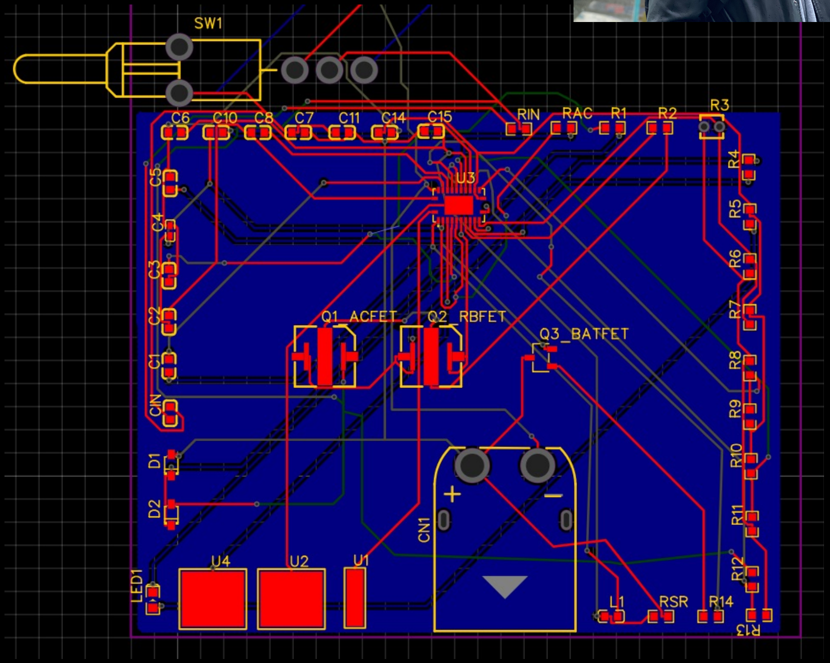
- DC-DC Converter
- Barrel Jack/ Laptop Charger
- Input DC Power 19V
- Output to Battery IC 12V at 1.5mA





PCB Schematic [Battery Management IC]





- The design of the Battery Management system was done using the "Typical Application" provided by the Datasheet

- Main Chip component: BQ24133RGYR under the Family component: BQ24133

- Using this aspect of our chip, we can charge 3 Cells at one time, protect them, and manage them, it was an ideal choice.

TYPICAL APPLICATIONS

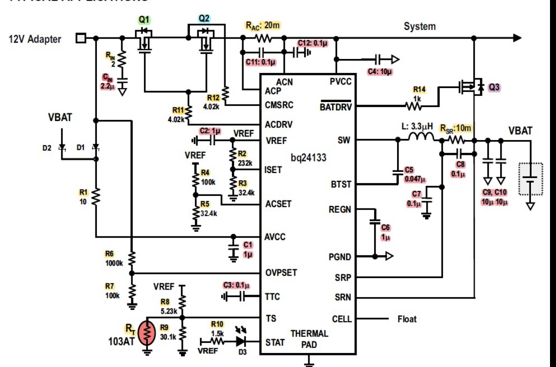
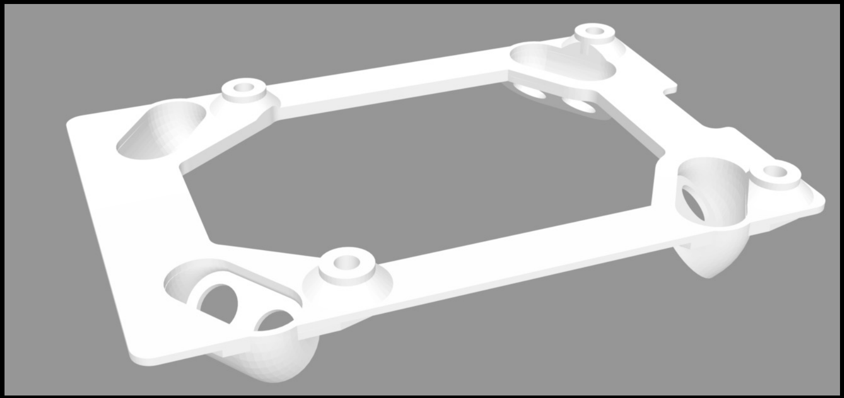
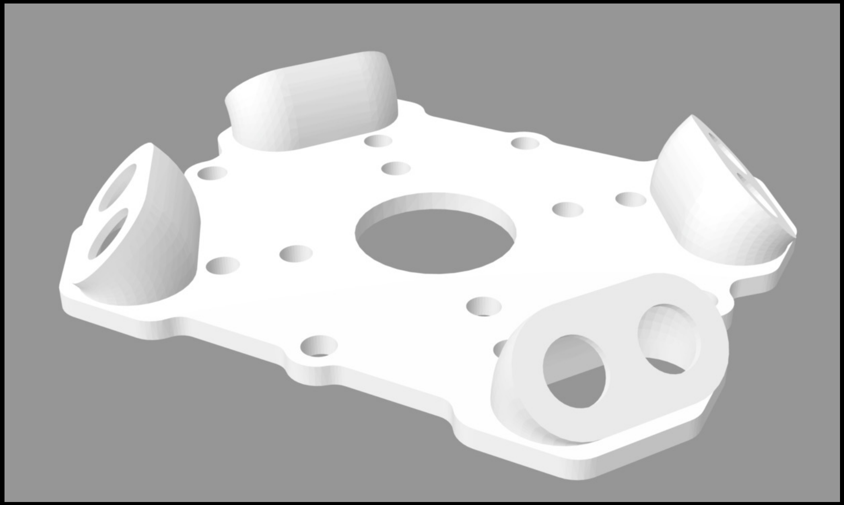
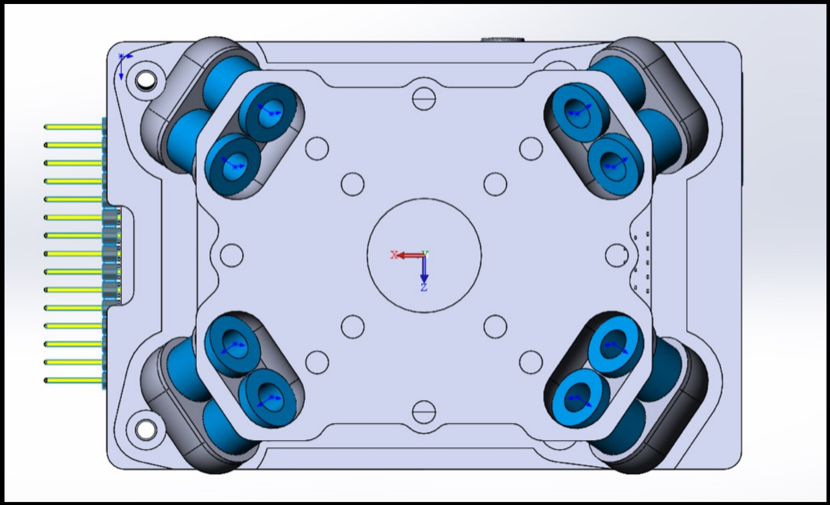


Figure 1. Typical Application Schematic (12V input, 2 cell battery 8.4V, 2A charge current, 0.2A pre-charge/termination current, 2A DPM current, 18V input OVP, 0 – 45°C TS)

Vibration Dampening Mount







Flight Control Board

Board	Price ($)	Weight (g)	Firmware	OS	Popularity	Bus Width
Pixhawk	80	60	ArduPilot, PX4	No	High	32 Bit
Pixhawk Cube	250	150	ArduPilot, PX4	No	High	32 Bit
Navio2 + Raspberry Pi 4	168 + 35/55/75	23	ArduPilot, PX4	Linux	Medium	64 Bit
BeagleBone Blue	80	52	ArduPilot, PX4	Linux	Low	32 Bit

Flight Control Board

Board	Processor	RAM	Ports	Sensors	Networking
Pixhawk 1	180 MHz ARM® Cortex® M4 with single-precision FPU	256 KB	I2C, CAN, ADC, UART, CONSOLE, PWM, PWM/GPIO/PWM input, S.BUS/PPM/SPEKTRUM, S.BUS out, microUSB	gyroscope, accelerometer/ magnetometer, accelerometer/gyroscope, barometer	wifi, DSM-X
Pixhawk Cube	180 MHz ARM® Cortex® M4 with single-precision FPU	256 KB	Same as Pixhawk 1 but more UART ports + UART HW flow control	gyroscope, accelerometer/ magnetometer, accelerometer/gyroscope, barometer	wifi, DSM-X
Navio2 + Raspberry Pi 4	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	2GB, 4GB or 8GB	UART, I2C, ADC, 12 PWM servo outputs, PPM/S.Bus input, HDMI, MIPI display port, MIPI camera port, microSD, USB3.0, USB 2.0	Accelerometers, gyroscopes, and magnetometers, Barometer	2.4GHz WiFi, Bluetooth, BLE
BeagleBone Blue	Octavo Systems OSD3358 1GHz ARM Cortex-A8	512 MB	4x UART, 2-cell LiPo, 2x SPI, I2C, 4x A/D converter, CAN bus (w/ PHY), 8x 6V servo motor, 4x DC motor, 4x quadrature encoder, USB 2.0 480Mbps Host/Client Port, USB 2.0 Host Port	9 axis IMU (access, gyros, magnetometer), barometer, thermometer	2.4GHz WiFi, Bluetooth, BLE

Data Processing Computer

	Raspberry Pi 4	NVIDIA Jetson Nano
CPU	Quad-core ARM Cortex-A72 64-bit @ 1.5 Ghz	Quad-Core ARM Cortex-A57 64-bit @ 1.42 Ghz
GPU	Broadcom VideoCore VI (32-bit) (Integrated)	NVIDIA Maxwell w/ 128 CUDA cores @ 921 Mhz
Memory	4 GB LPDDR4**	4 GB LPDDR4
Networking	Gigabit Ethernet / Wifi 802.11ac	Gigabit Ethernet / M.2 Key E (for Wifi support)
Display	2x micro-HDMI (up to 4Kp60)	HDMI 2.0 and eDP 1.4
USB	2x USB 3.0, 2x USB 2.0	4x USB 3.0, USB 2.0 Micro-B
Other	40-pin GPIO	40-pin GPIO
Video Encode	H264(1080p30)	H.264/H.265 (4Kp30)
Video Decode	H.265(4Kp60), H.264(1080p60)	H.264/H.265 (4Kp60, 2x 4Kp30)
Camera	MIPI CSI port	MIPI CSI port
Storage	Micro-SD	Micro-SD
Price	$55 USD	$99 USD

Data Processing Computer

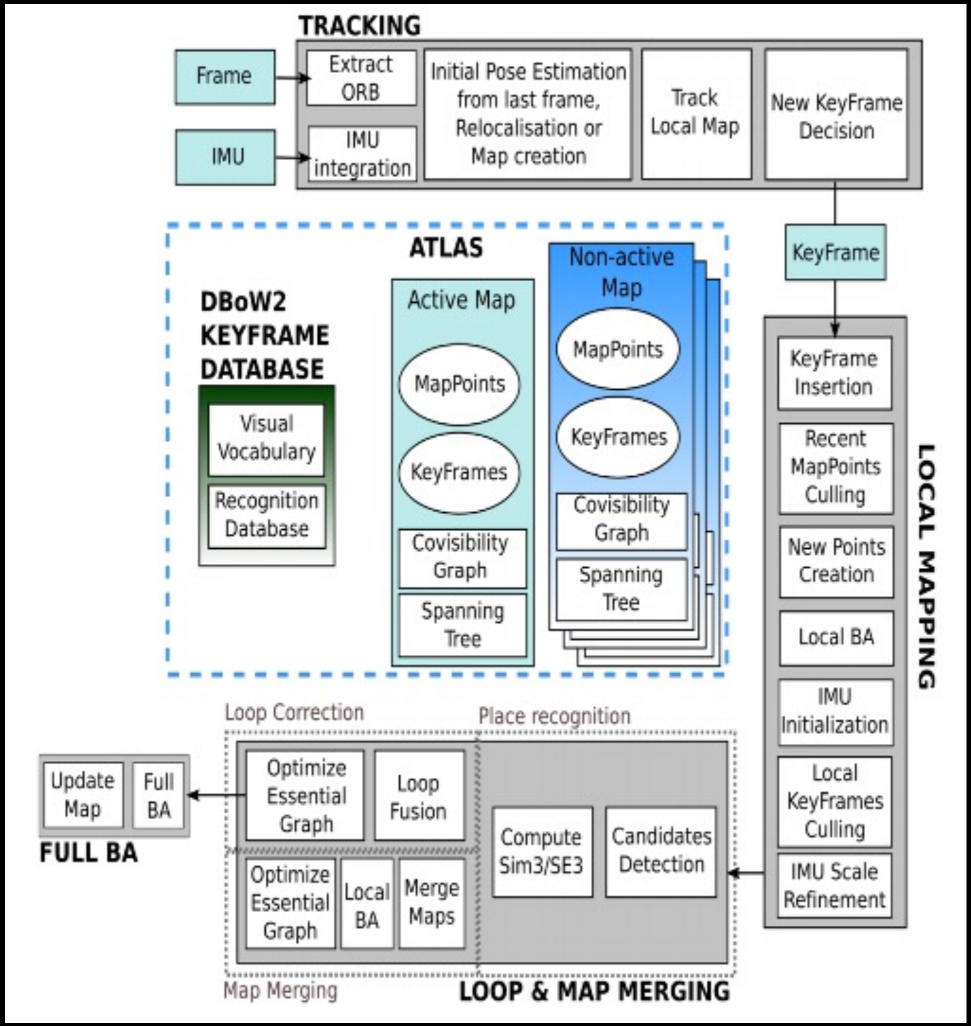
Test	Raspberry Pi 4 B 8GB	Jetson Nano 4GB
Tinymembenc Standard Memcpy	2526.8 MB/s	3501 MB/s
TTSIOD 3D Renderer Phong Rendering With Soft-Shadow Mapping	32.3075 FPS	41.00 FPS
7-Zip Compression Compress Speed Test	3701 MIPS	3501 MIPS
C-Ray Total Time – 4K, 16 Rays Per Pixel	609.431 Seconds	932 Seconds
Primesieve 1e12 Prime Number Generation	519.238 Seconds	468 Seconds
AOBench Size: 2048 x 2048	125.643 Seconds	187 Seconds
FLAC Audio Encoding WAV To FLAC	85.805 Seconds	104.01 Seconds
LAME MP3 Encoding WAV To MP3	124.952 Seconds	144.21 Seconds
Perl Pod2html	0.58082059 Seconds	0.7114 Seconds
Redis GET	491168.39 Requests/Second	568431 Requests/Second
PyBench Total For Average Test Times	5673 Milliseconds	7080 Milliseconds

Mapping Software

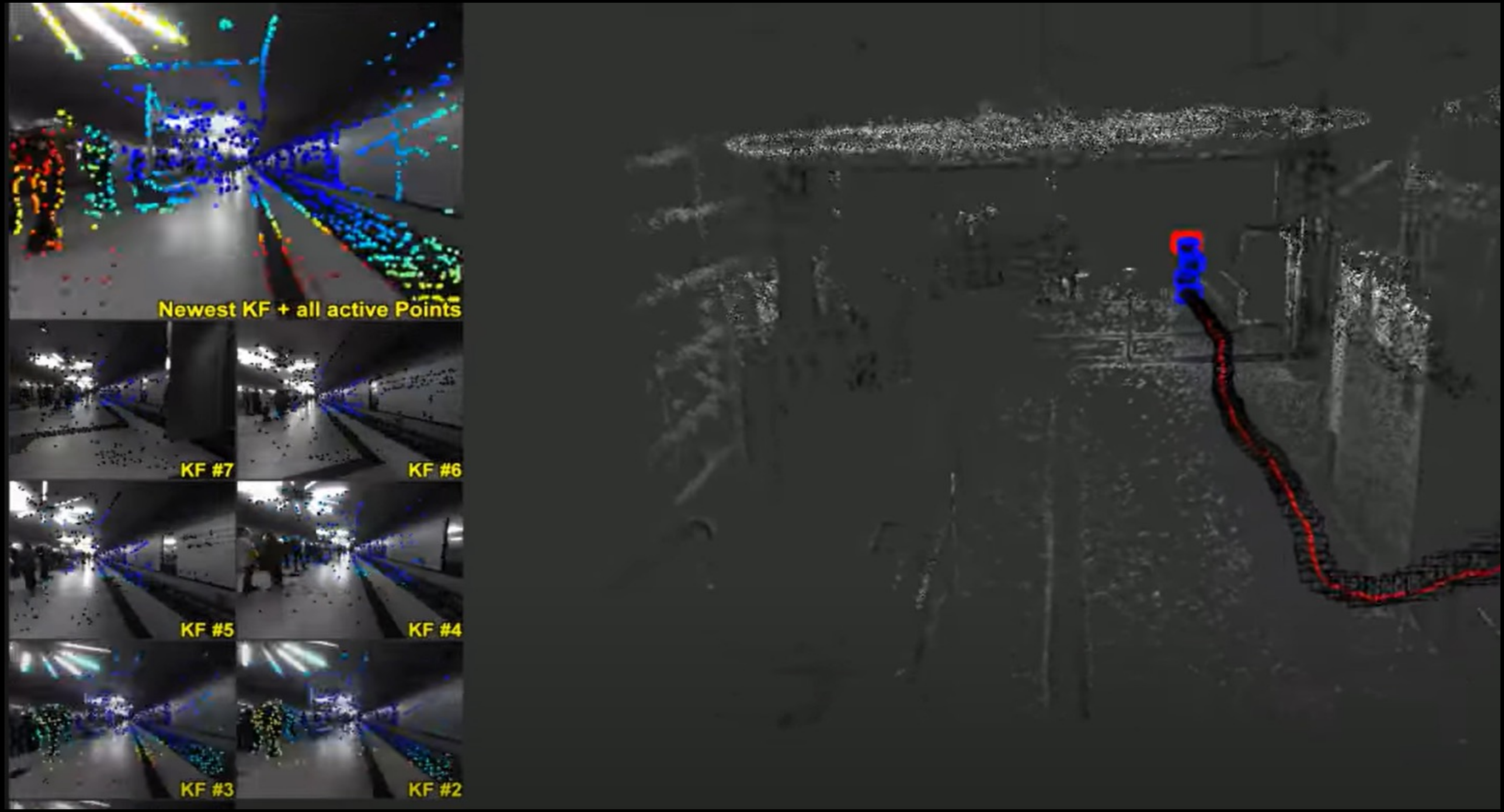
	Frame Times (ms)		
Mapping Algorithm	**Mean**	**St.D.**	**CPU @20fps**
ORB Mono SLAM	29.81	5.67	187
LSD Mono SLAM	23.23	5.87	236
DSO Mono	9	-	-
SVO Mono	2.53	0.42	55

- We decided to use ORB-SLAM, and keep the others as backups in case ORB-SLAM does not run fast enough.

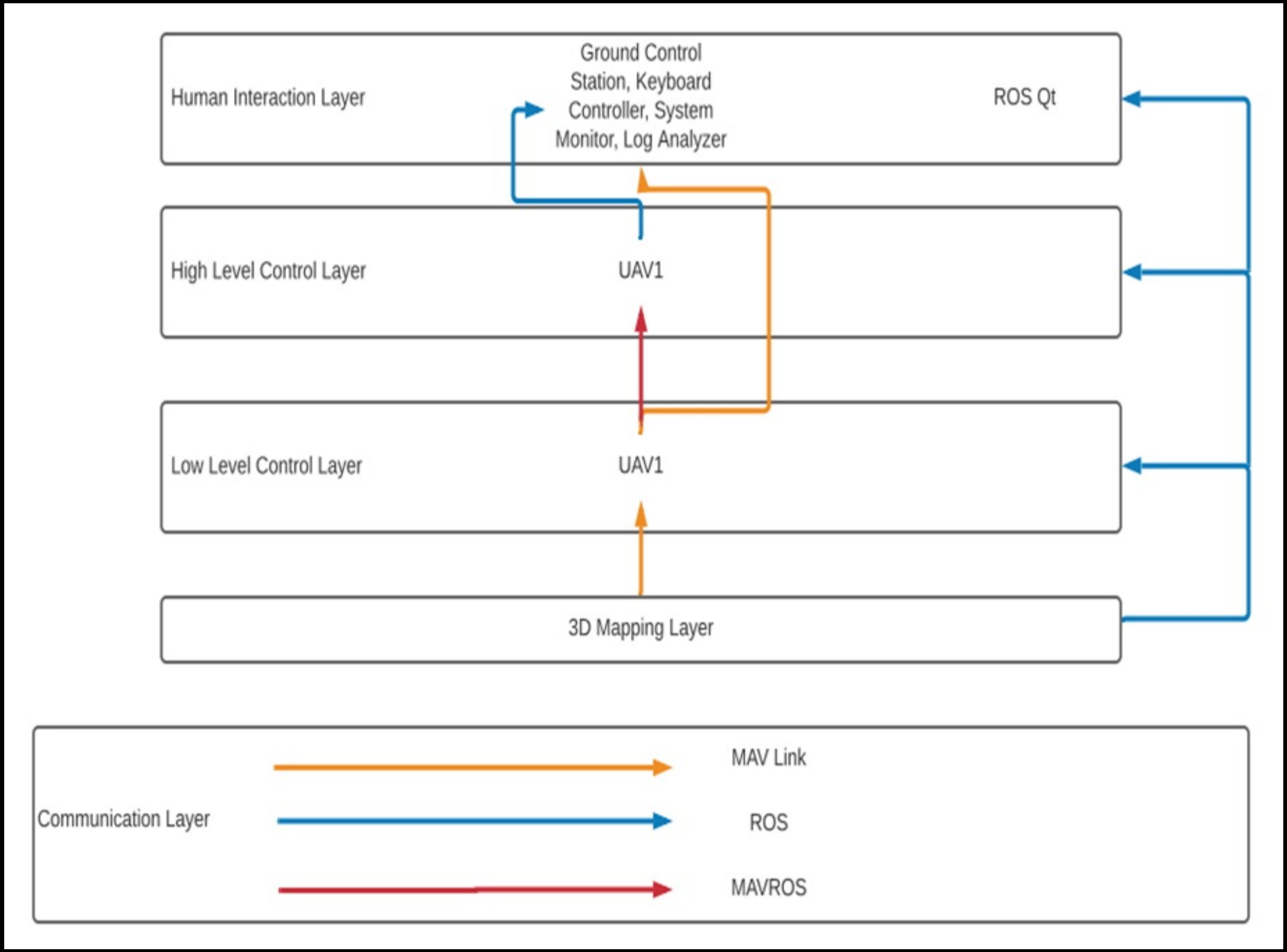
ORB-SLAM 3.0


TRACKING
Frame
IMU
Extract ORB
IMU integration
Initial Pose Estimation from last frame, Relocalisation or Map creation
Track Local Map
New KeyFrame Decision
KeyFrame
ATLAS
DBoW2 KEYFRAME DATABASE
Visual Vocabulary
Recognition Database
Active Map
MapPoints
KeyFrames
Covisibility Graph
Spanning Tree
Non-active Map
MapPoints
KeyFrames
Covisibility Graph
Spanning Tree
LOCAL MAPPING
KeyFrame Insertion
Recent MapPoints Culling
New Points Creation
Local BA
IMU Initialization
Local KeyFrames Culling
IMU Scale Refinement
Loop Correction
Place recognition
Optimize Essential Graph
Loop Fusion
Compute Sim3/SE3
Candidates Detection
Optimize Essential Graph
Local BA
Merge Maps
Map Merging
LOOP & MAP MERGING
Update Map
Full BA
FULL BA

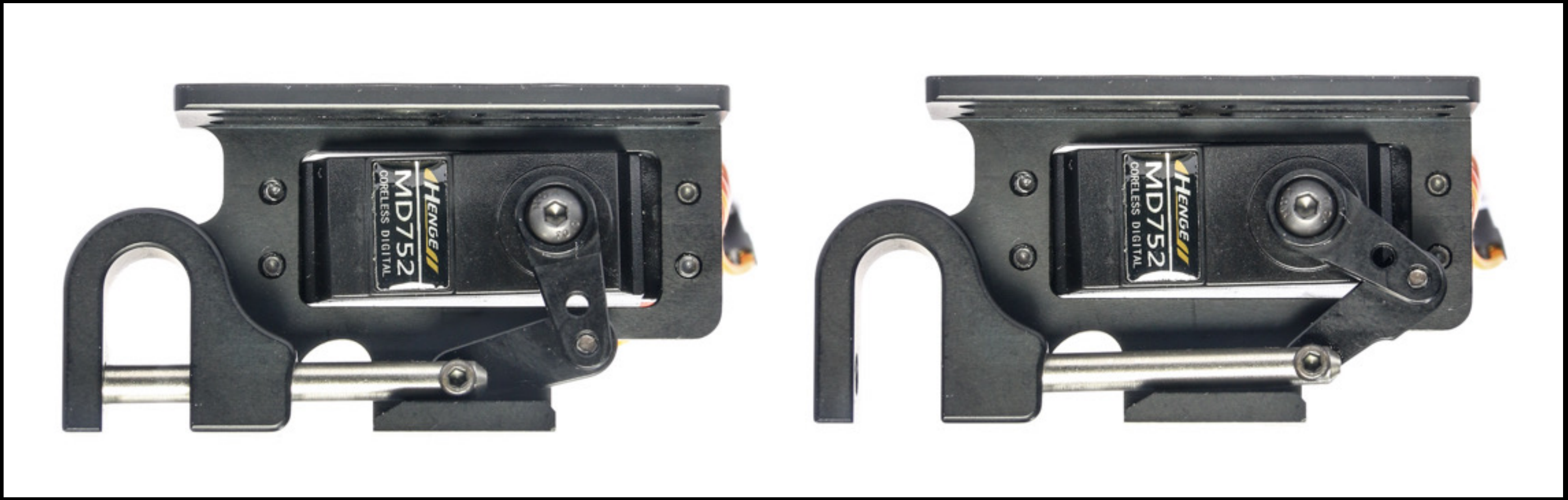
ORB-SLAM 3.0



Software Flowchart



Package Release Mechanism



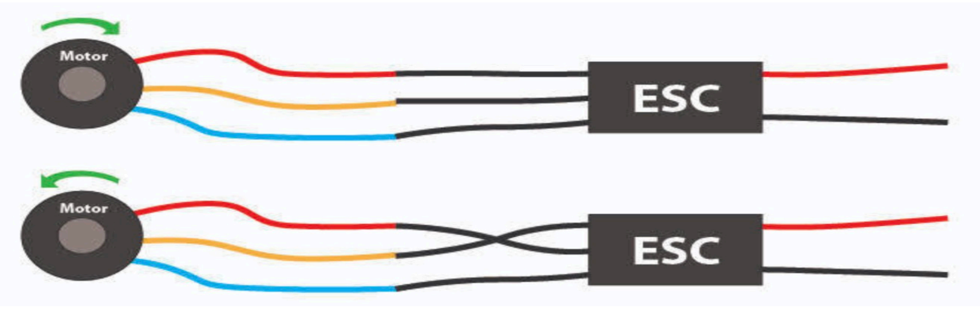
Sensors

	Long Range	Cost	Can handle External Conditions	3D Imaging
Ultrasonic	No	Low	No	No
IR	No	Low	Yes	No
LiDar	Yes	High	Yes	Yes
Time of Flight	Yes	Moderate	Yes	Yes
Camera	Yes	Low	Yes	Yes

- With having a camera we did not have a need for 3D imaging
- Most cost-effective option for object detection

Electric Speed Controller

- Electric Speed Controller
  - Translator from the 'pilot' to the motor by giving specific voltage/current based on input from 'user'
  - Options we had to choose from
    - 4 in 1 packages or individual
      - Individual is most cost effective because if one breaks only one must be replaced
    - BEC vs no BEC
      - Battery Elimination Circuit: provides control of power to other components
      - Not necessary for our drone project, but was included in the kit purchased as a cost-effective decision

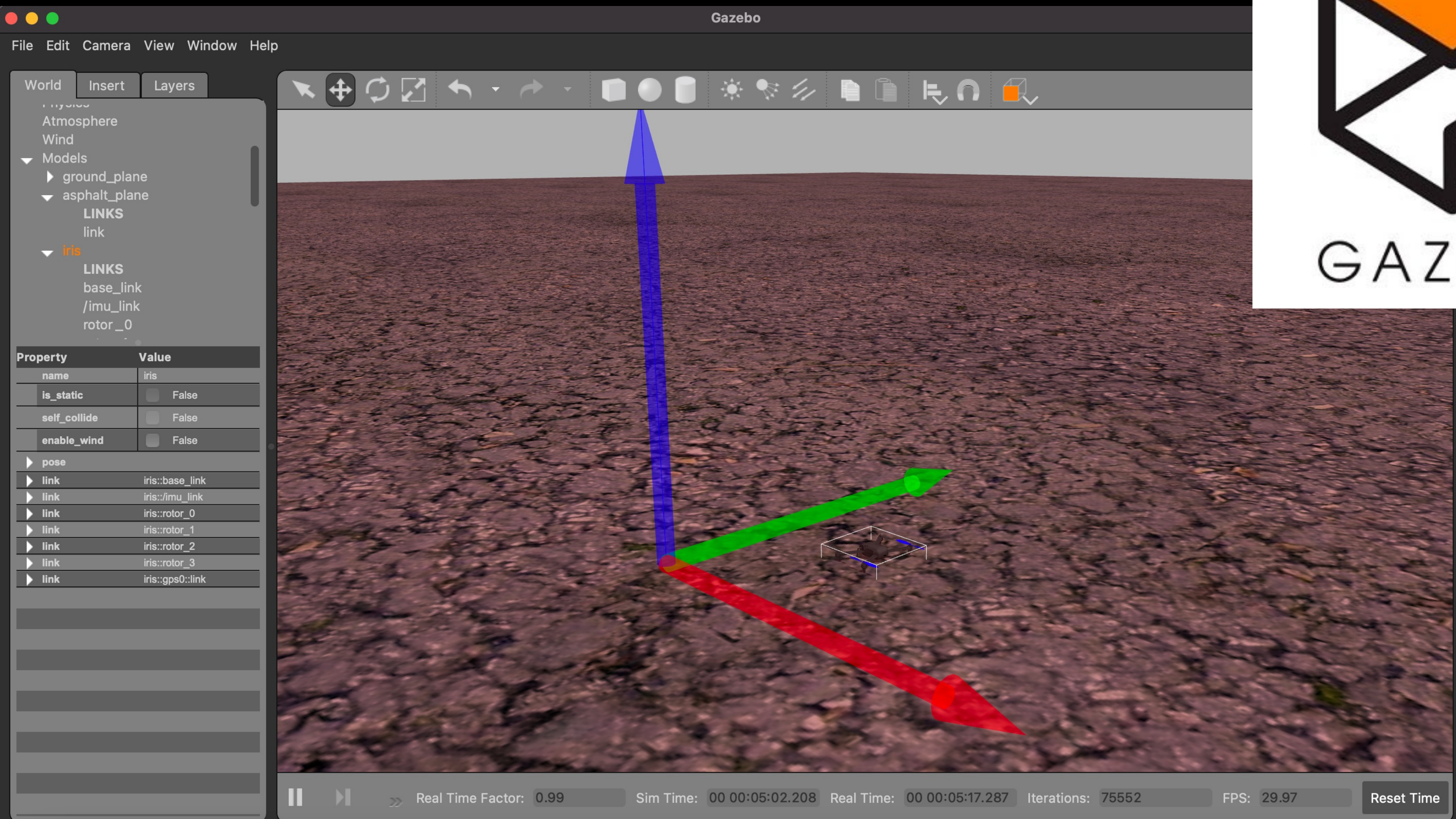


PX4 vs Ardupilot

	PX4	Ardupilot
Coding Language	C++	C++
Majority Hardware Capabilities	Yes	Yes
Majority Software Capabilities	Yes	Yes
Release Date	2009	2012

Simulation

GAZEBO



Administration Content

Budget

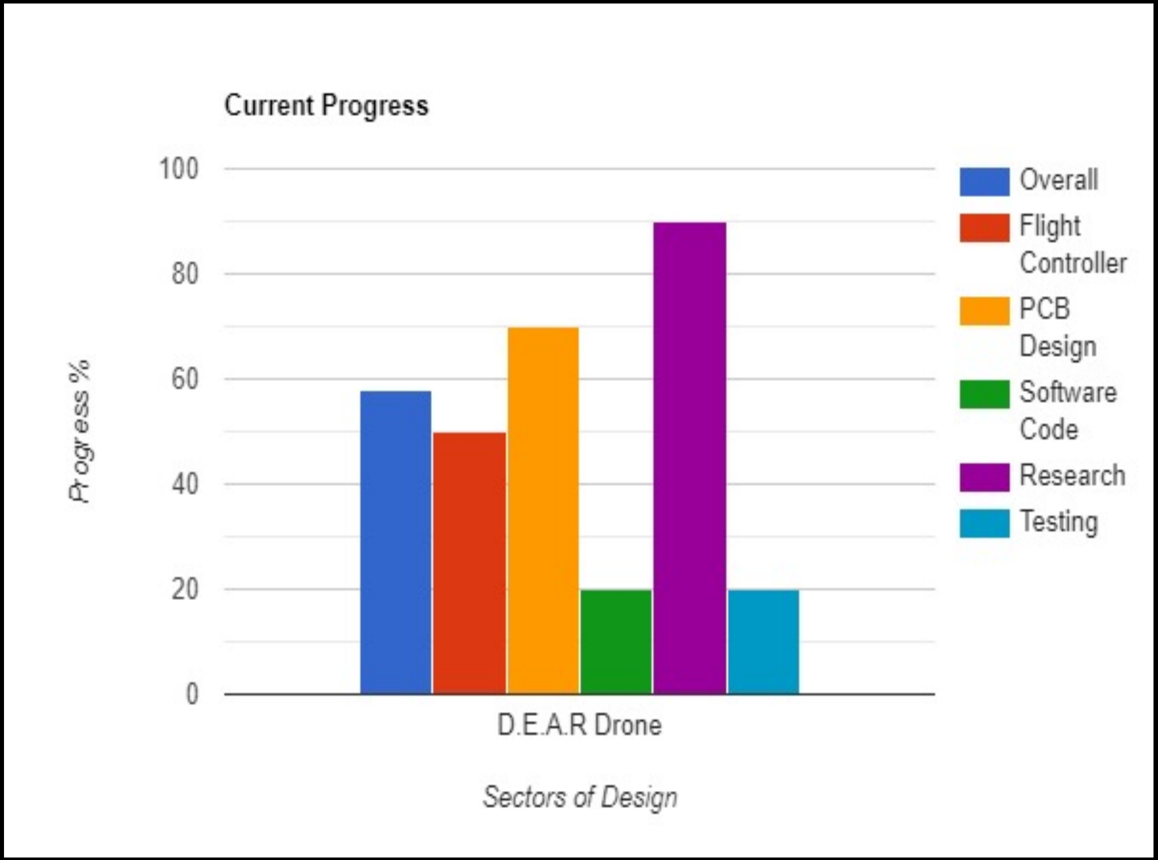
Frame	1	
Brushless Motors	4	
ESC	4	$56.72
Navio 2 - Flight Controller	1	$246
Raspberry Pi 4b	1	$75
Battery	2	$92.84
PCB	3	$133
Electronic Components	6	$6.99
Ultrasonic sensors	6	$10.59
Camera	1	$13.10
Total		$634.24

Challenges

- Hardware
  - Components Available
  - Manufactures
  - Shipping Time
- Software
  - Setting up development environment
  - Communication between development computer and flight controller
  - Flashing firmware onto flight controller
- Overall Challenge
  - Self-Funded

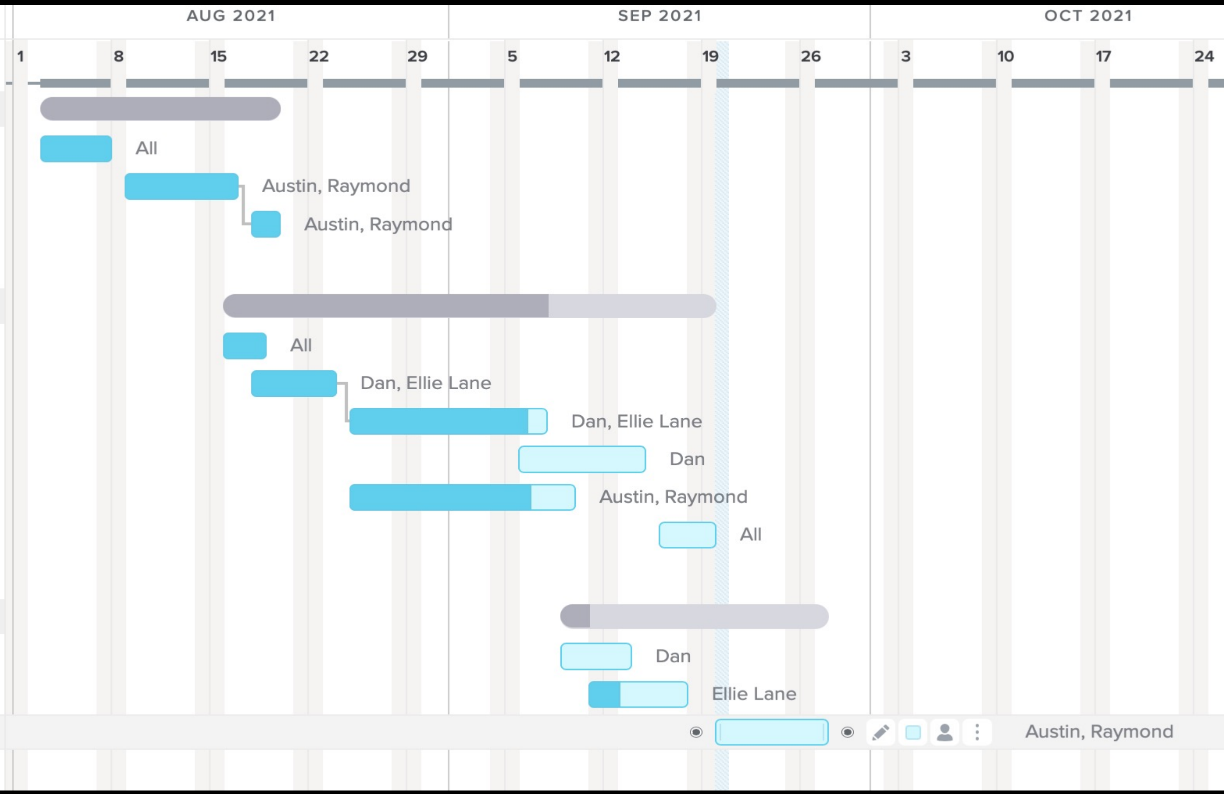
Current Progress

- Overall build: 58%
- Flight controller: 50%
- PCB designed: 70%
- Code: 20%
- Research: 90%
- Testing: 20%



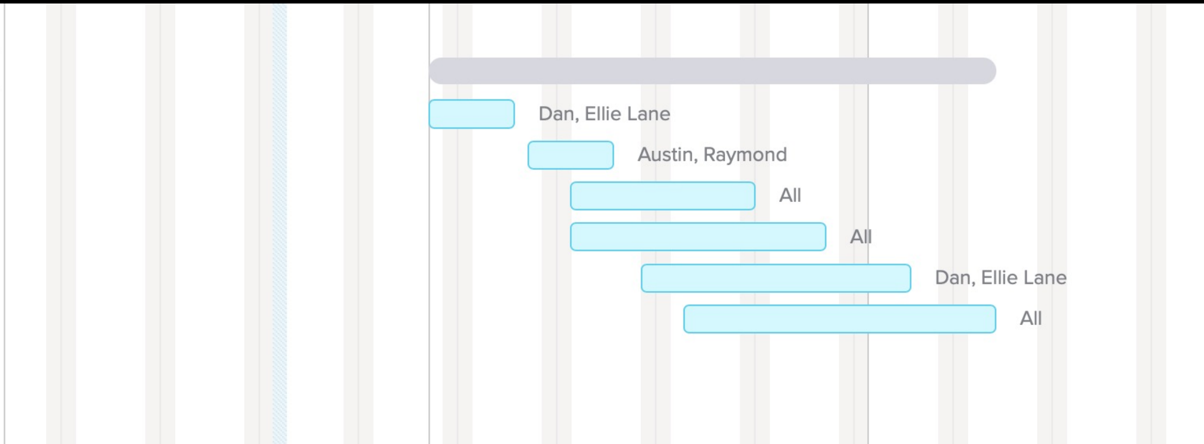
Gaant Chart

	Assigned	Progress
D.E.A.R.		30%
▼ Ordered Parts		100%
Purchased first round	All	100%
Design PCB	Austin, Raymond	100%
Purchased PCB	Austin, Raymond	100%
▼ Build		66%
Assemble Frame, motors and p...	All	100%
Configure Raspberry Pi	Dan, Ellie Lane	100%
Configure Navio onto Rasp.	Dan, Ellie Lane	90%
Connect to frame	Dan	0%
Test electrical components	Austin, Raymond	80%
Hover Test	All	0%
▼ Develop		11%
Create mapping alg. files	Dan	0%
Create computer vision files	Ellie Lane	30%
Adjust components for balance	Austin, Raymond	0%



Gaant Chart Cont.

▼ Test		0%
Test using simulation	Dan, Ellie Lane	0%
Test electrical components	Austin, Raymond	0%
Take off test	All	0%
Path test	All	0%
Object detection Test	Dan, Ellie Lane	0%
Full path test	All	0%



Future Progress

- Hardware Progress
  - PCB
  - PDB
- Software Progress
  - Flight path
  - Machine Learning

Thank you!

Q&A