REMOTE AUTONOMOUS MAPPING OF RADIO FREQUENCY OBSTRUCTION DEVICES

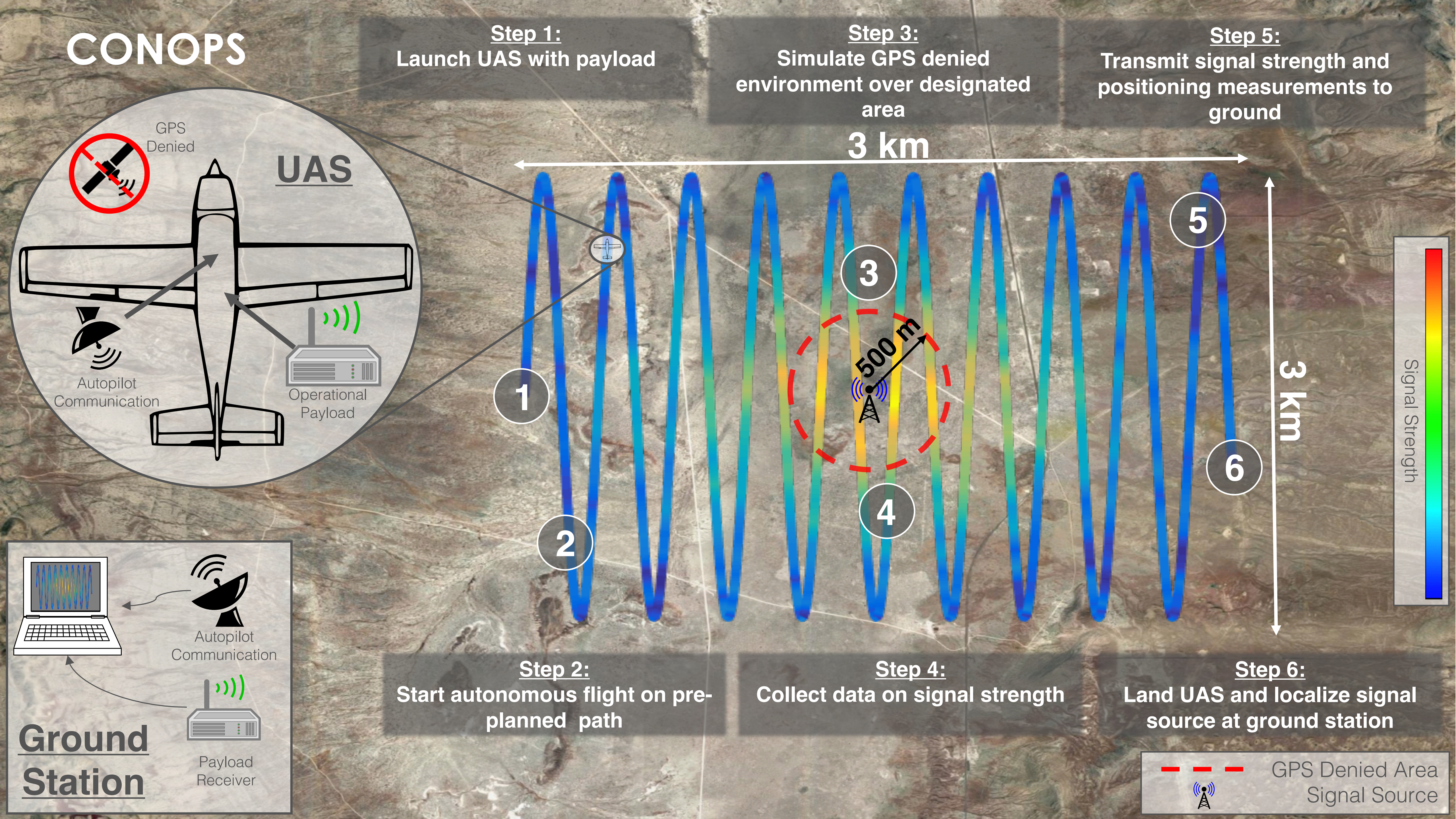
Team: Jorgen Baertsch, Ian Cooke, Kennedy Harrmann, Mary Landis, Sarah Larson, Harrison Mast, Ethan Morgan, Selby Stout, Jake Ursetta, Justin Williams, Samantha Williams

Sponsor: Dennis Akos

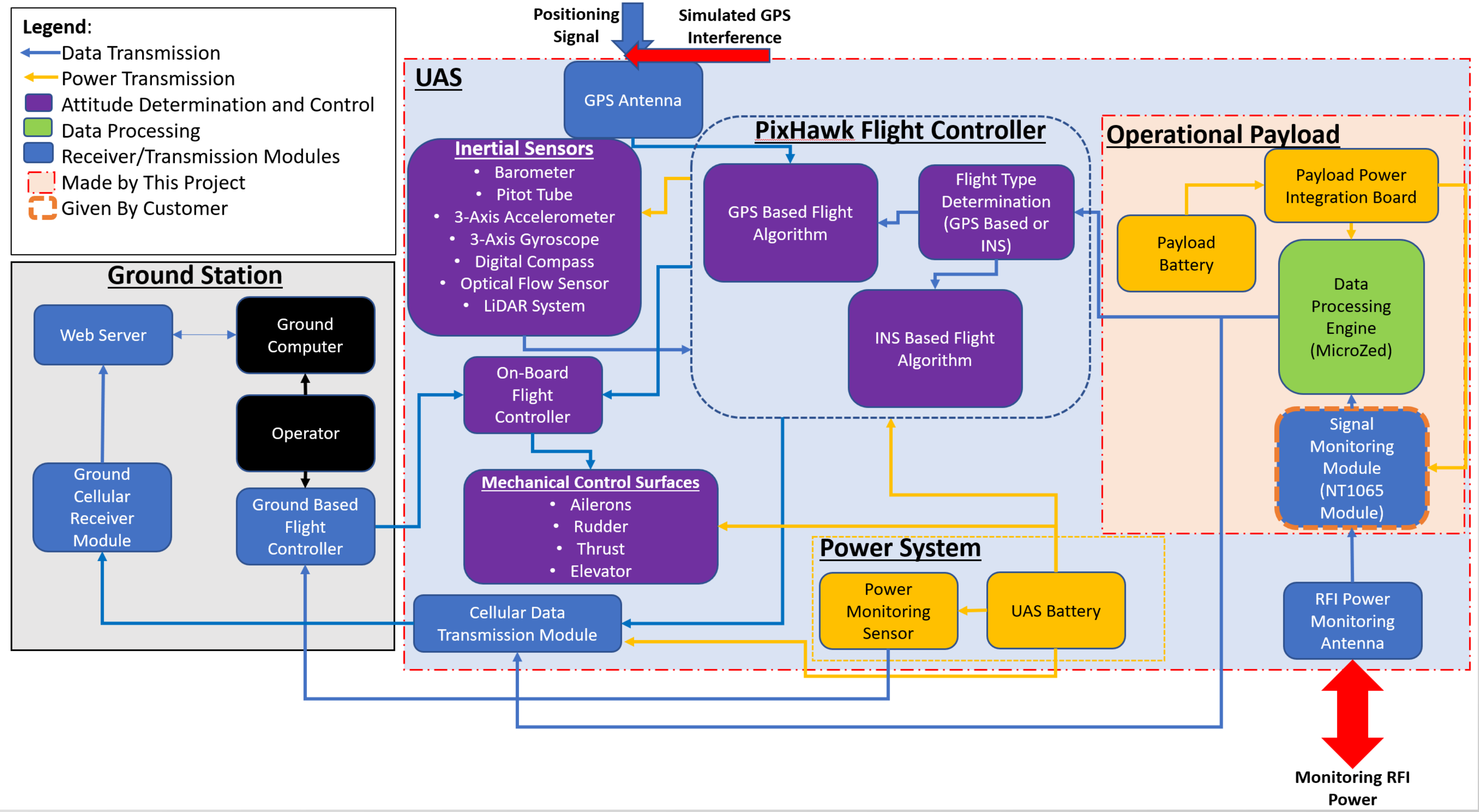
Advisor: Jade Morton

MISSION STATEMENT

<u>RAMROD</u> will utilize an autonomous **UAS** and self-contained **sensor payload** to **localize** Radio Frequency Interference and Emerging Threat sources in a **GPS-denied environment** to allow civilian and military GNSS endeavors to continue without disruption.


CONOPS
Step 1:
Launch UAS with payload
Step 3:
Simulate GPS denied environment over designated area
Step 5:
Transmit signal strength and positioning measurements to ground
3 km
GPS Denied
UAS
Autopilot Communication
Operational Payload
1
2
3
4
5
6
500 m
3 km
Signal Strength
Autopilot Communication
Payload Receiver
Ground Station
Step 2:
Start autonomous flight on pre-planned path
Step 4:
Collect data on signal strength
Step 6:
Land UAS and localize signal source at ground station
GPS Denied Area
Signal Source

FUNCTIONAL BLOCK DIAGRAM



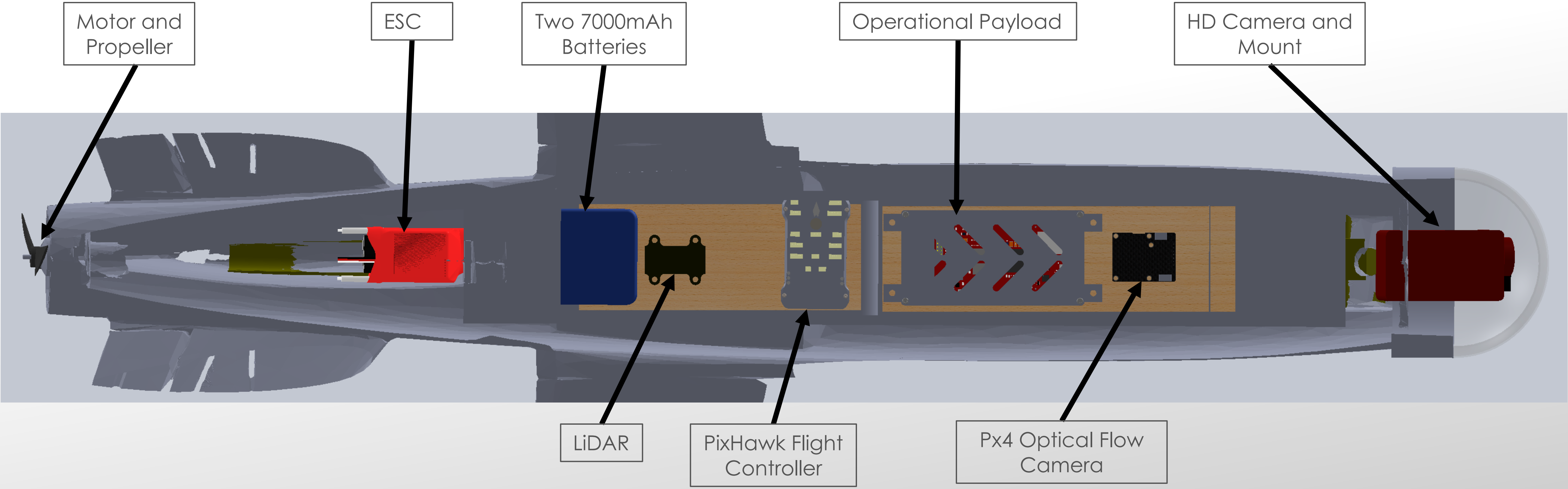
CRITICAL PROJECT ELEMENTS

CPE	Description	Reason
UAS Platform	Develop a UAS platform capable of maintaining flight in a GPS denied environment while supporting all RFI measuring equipment	A UAS capable of supporting the necessary sensors would be the best means of covering the required area.
GPS Denied Flight software	Maintain autonomous flight while in a simulated GPS denied environment for up to 200 seconds at a time	A PPD or ET will cause GPS data to be inaccurate.
Payload	Self-powered sensor payload that can monitor, store and transmit RFI signal data while interfaced with the UAS platform	To measure the RF source all necessary sensors must be integrated together. By customer request the payload must be capable of taking RF measurements without UAS integration

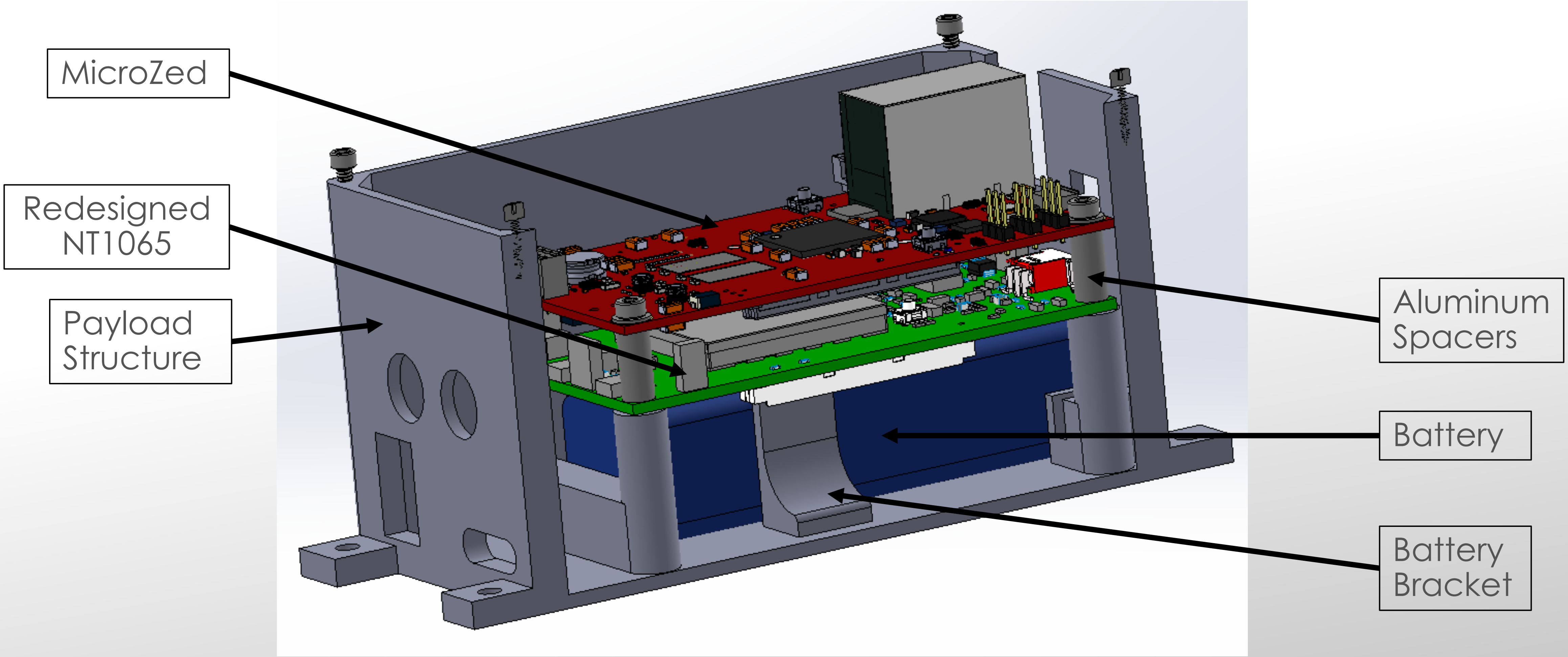
LEVELS OF SUCCESS

	Operational Payload	UAS Platform	GPS Denied Flight Software	RF Localization
Level 1	Store power measurement and location data	Minimum total flight time of 60 minutes. Maintain steady level flight over 1 km without GPS	Shall allow for GPS denied flight for 1 km	Shall be able to establish an RFI power profile without GPS
Level 2	Transmit data up to 4.25 km. Communicate power and location data with PixHawk	Fly in GPS denied area for a total of 10 minutes	Autopilot switches seamlessly to GPS denied flight	Localize RFI source within 40 m
Level 3			Enable flight with dynamic waypoints	

TALON BASELINE DESIGN



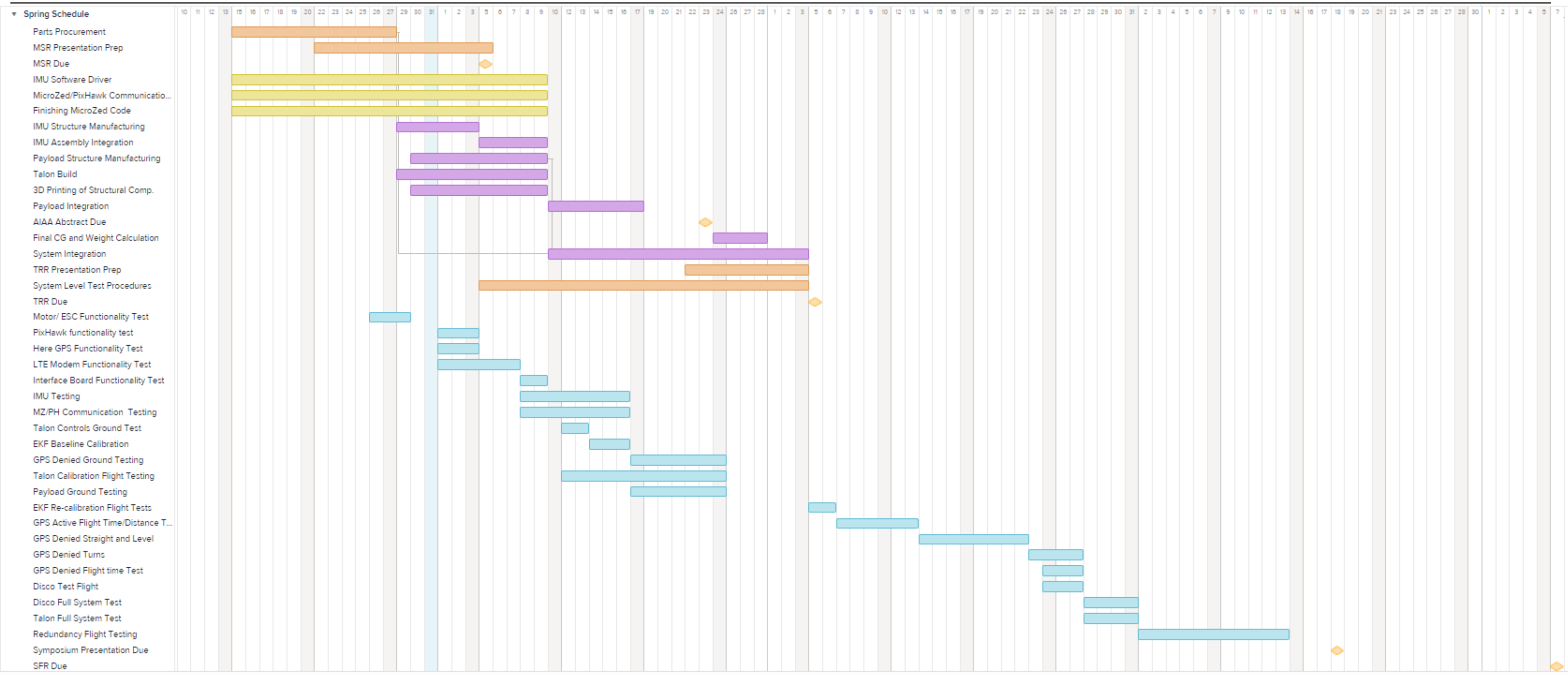
PAYLOAD BASELINE DESIGN



CHALLENGES/CONCERNS IN DESIGN

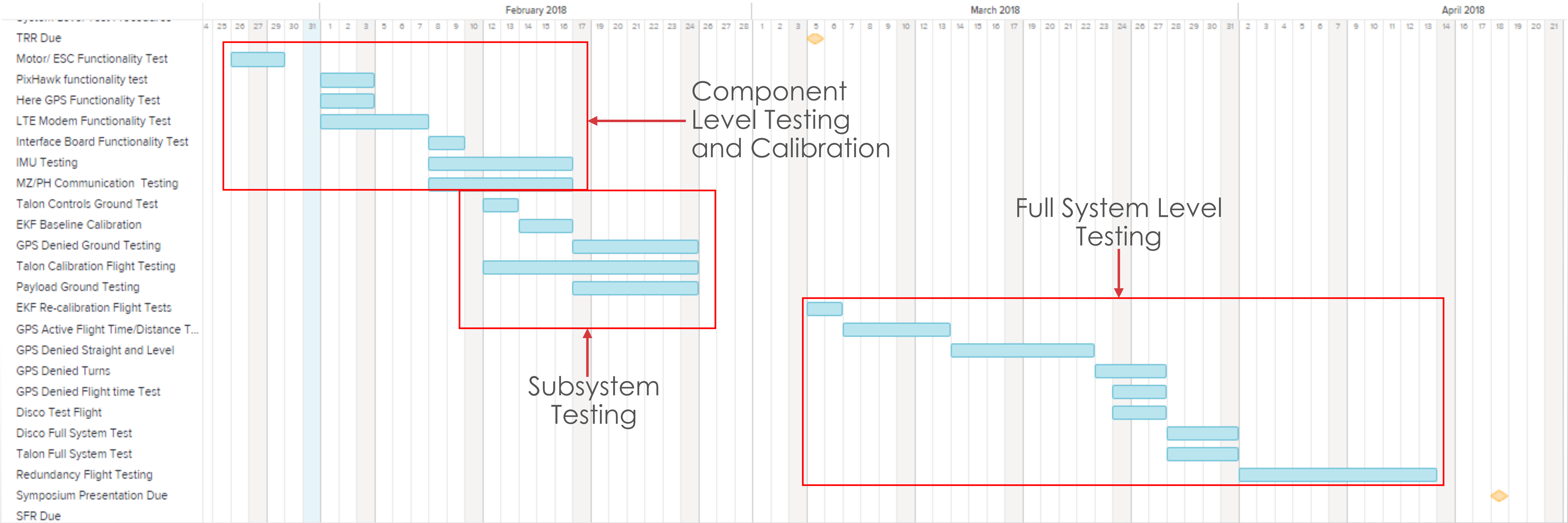
- Tuning the Kalman Filter
- Positioning the hardware so that the CG of the aircraft is in an appropriate spot
- Component redesign for 3D printing compatibility
- Communication between different software packages
  - MicroZed
  - PixHawk

SPRING SCHEDULE



Overview | Schedule | Manufacturing | Budget

TESTING SCHEDULE



MANUFACTURING

Color Code:

Task completed and verified

Task completed but not verified

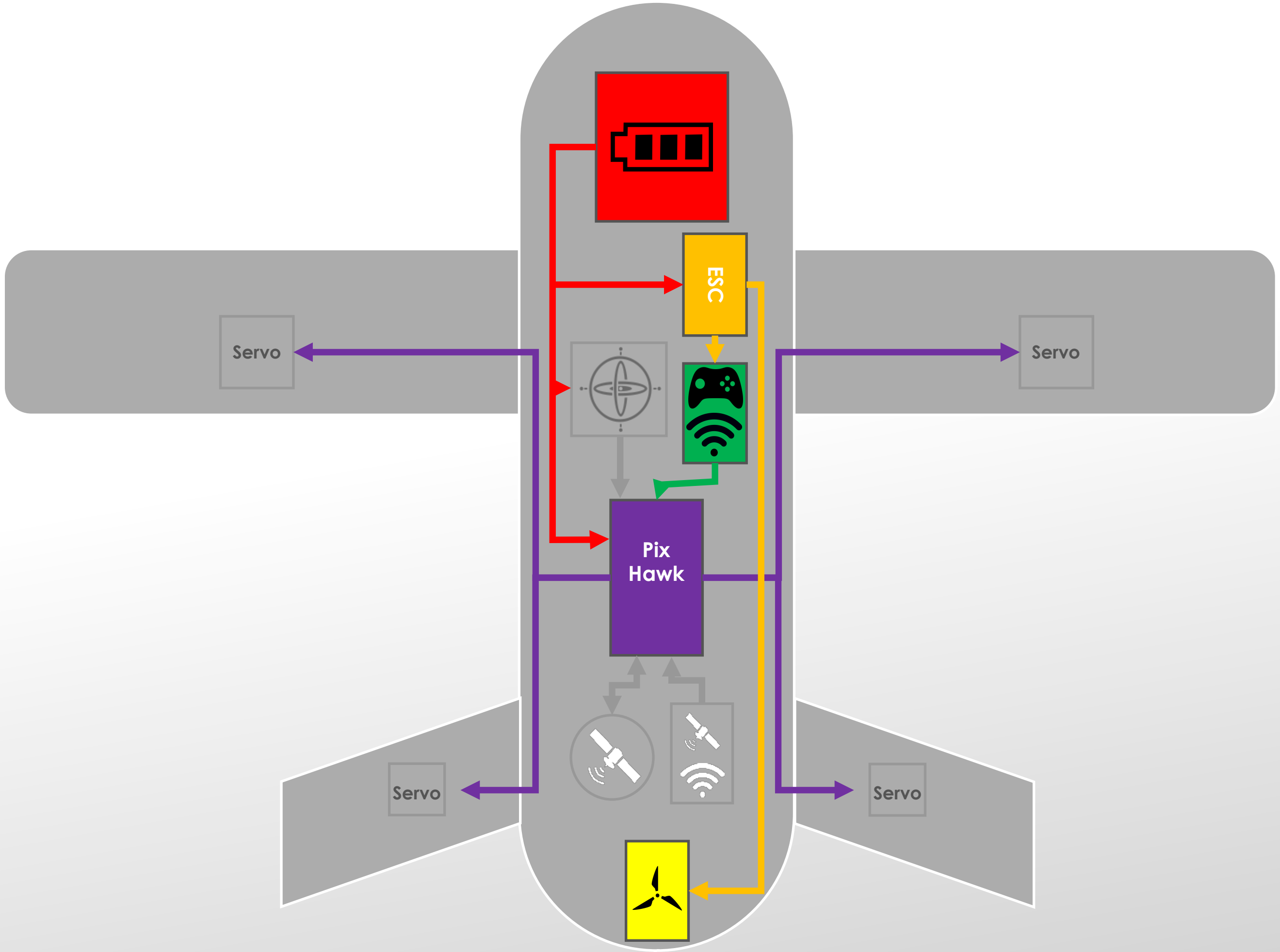
Task in progress

Task not started

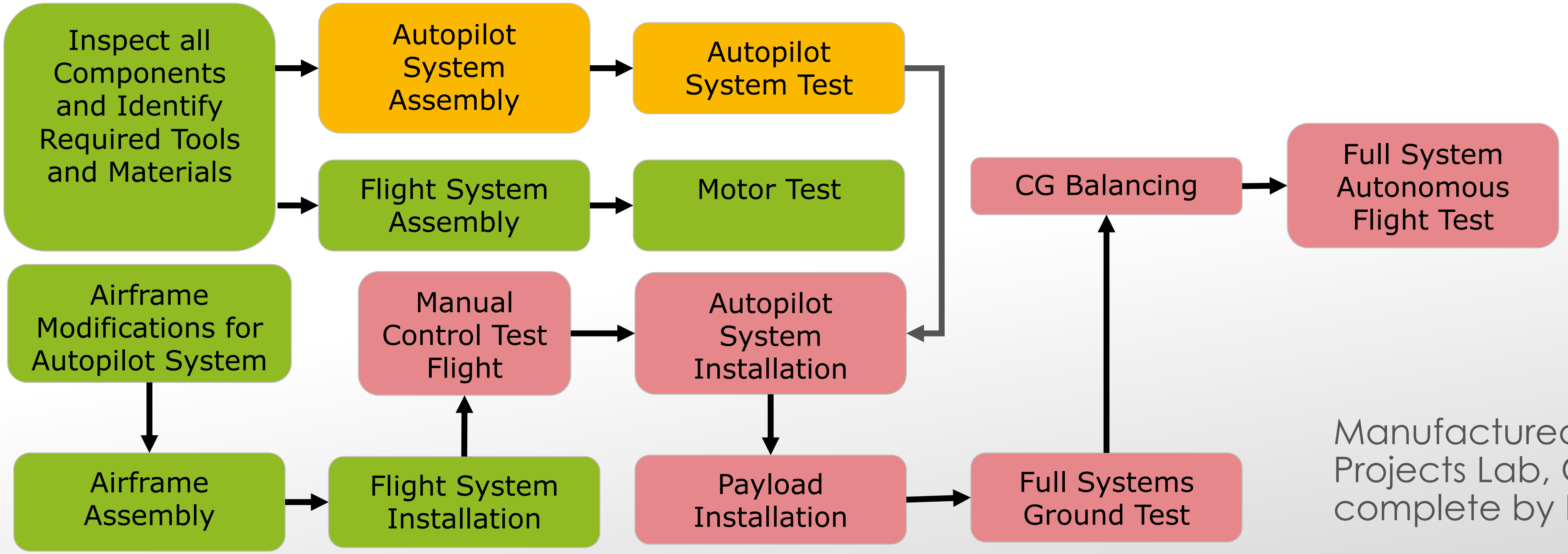
MANUFACTURING SCOPE

What to Manufacture:	Expected Completion Date:	Remaining Man Hours:			
Assembly of Talon	February 9th	10 hours	Main frame assembled	Motor Mounted	Component fitting/mounting
GPS Denied Software	February 9th	10 hours	Code written	Code implemented	Code testing
Bungee Launcher	February 11th	8 hours	System designed	Parts Ordered	Assembly of launcher
Interface between MicroZed and PixHawk	February 9th	12 hours	Code identified	Code implemented	Code testing
IMU Driver	February 9th	15 hours	Code written	Code implemented	Code testing
MicroZed Drivers for PixHawk	February 9th	20 hours	Code written	Code implemented	Code testing
3D printing	February 11th	6 hours	Parts designed	Most parts printed	Print last parts

UAS PLATFORM



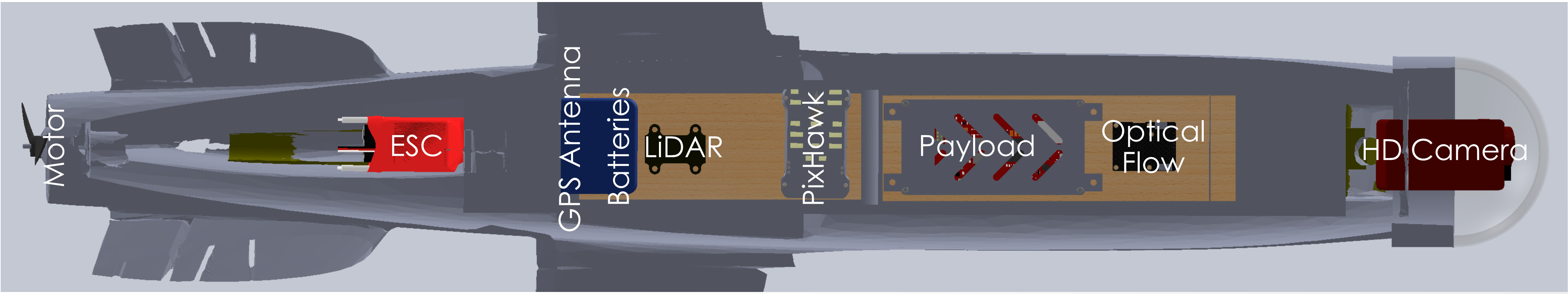
UAS PLATFORM INTEGRATION

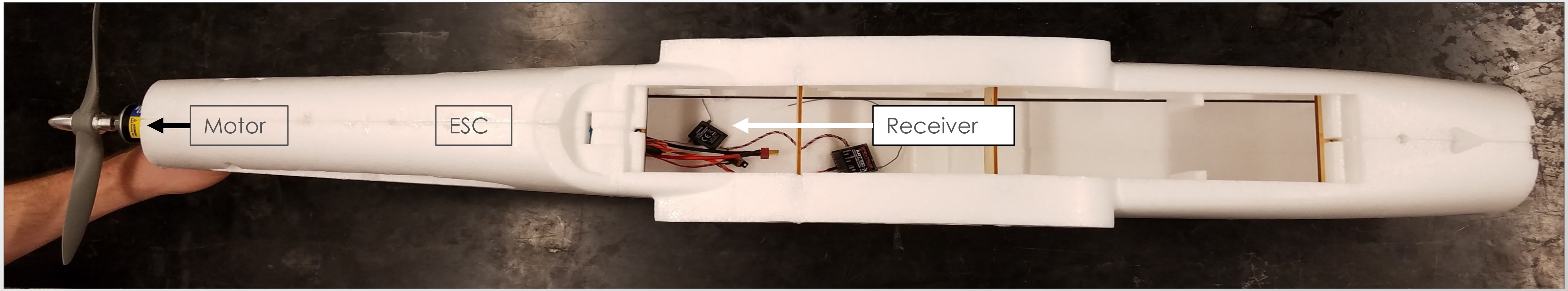


Manufactured in Senior Projects Lab, On track to complete by Feb 9th

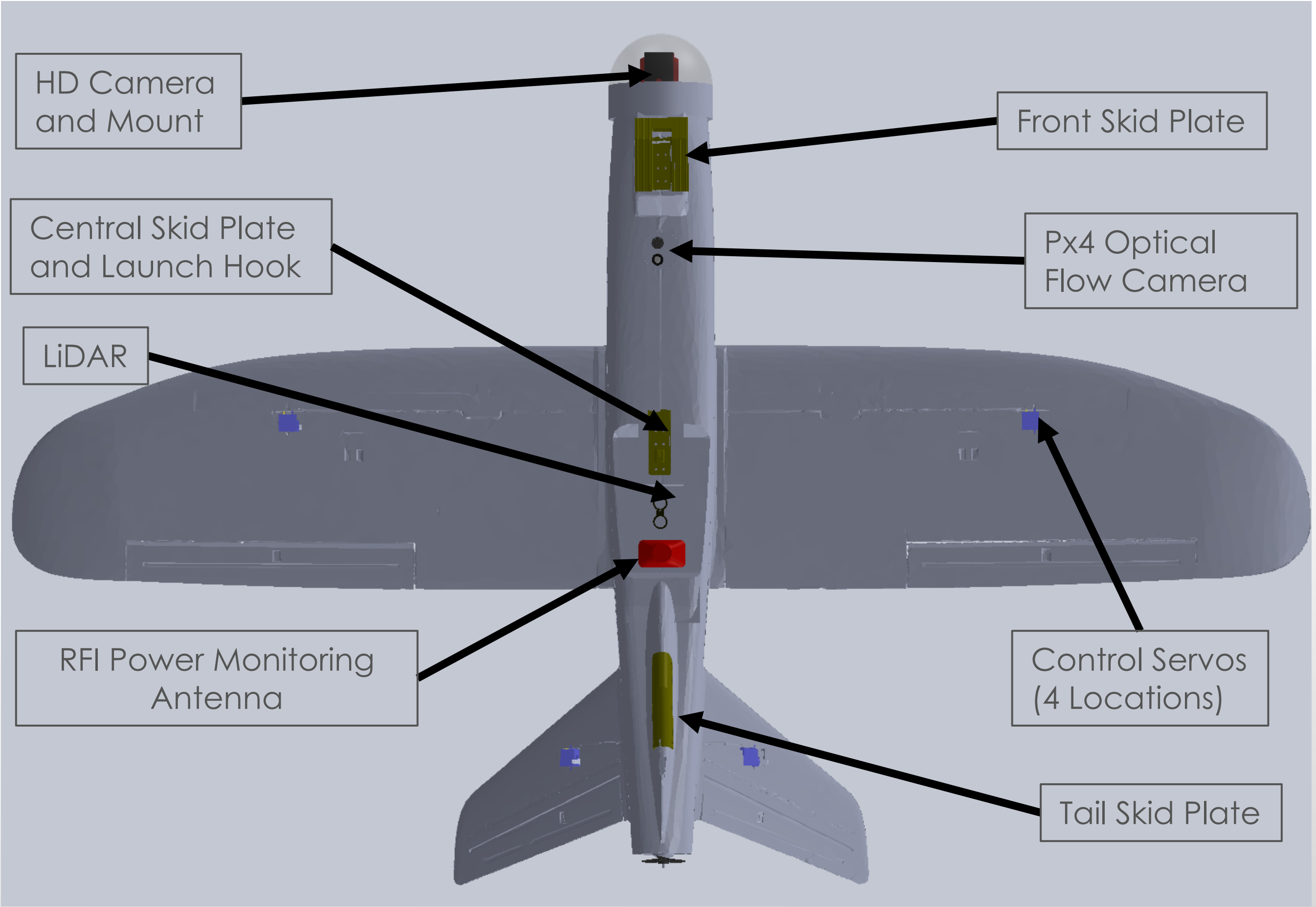
Airframe Assembly → Flight System Installation → Autopilot System Installation → Payload Integration → CG Placement and Complete Assembly → Bungee Launcher Assembly

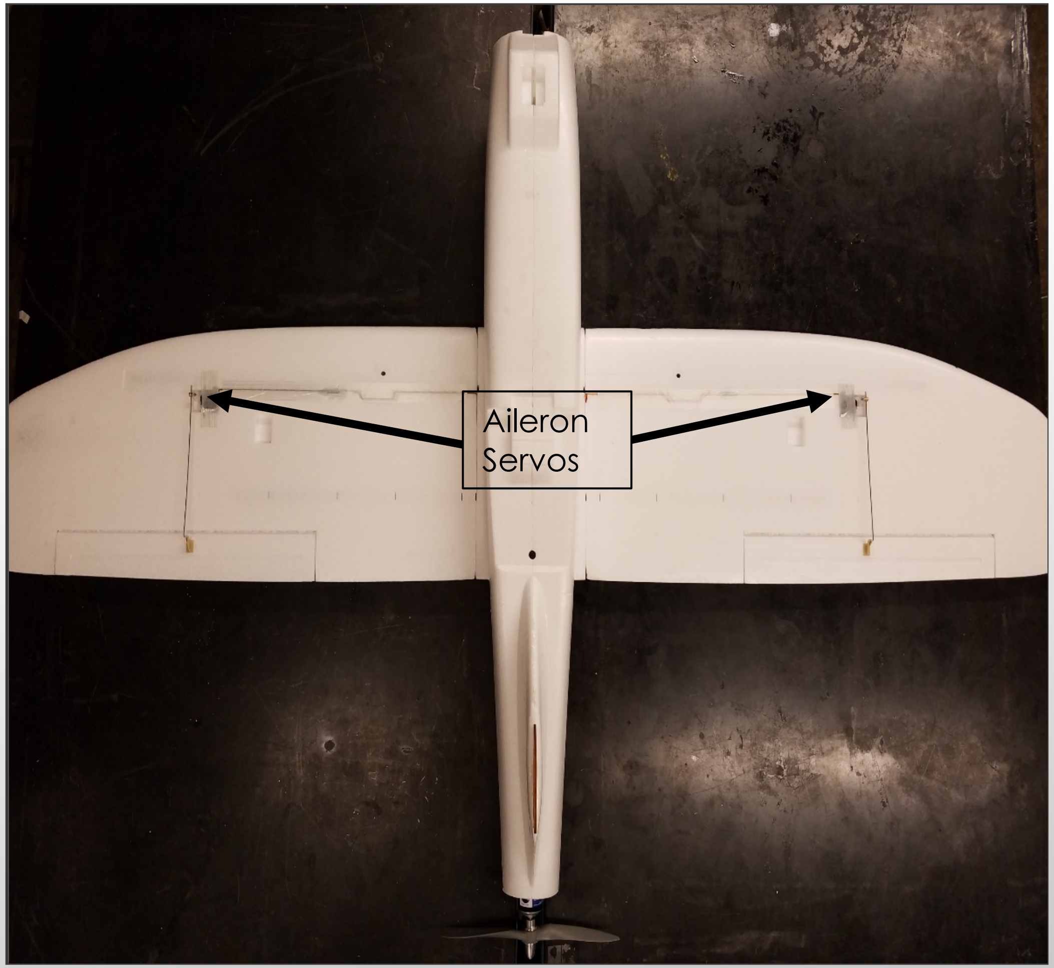
MANUFACTURING STATUS





MANUFACTURING STATUS





TALON MANUFACTURING STATUS

Component	Airframe	Flight System	Autopilot System	Bungee Launcher
Parts	Wings, Tail, Fuselage modifications, skid plates, 3d printed parts	Motor, Battery, ESC, Receiver, Servos	Pixhawk, GPS module, Optical flow, LiDAR, Pitot Tube, IMU, Power Monitoring Antenna	Frame, aircraft harness, bungee cables
Hours to Complete Manufacture and Integration	7/8	3/4	2/10	2/10
Most Challenging Aspect	Modifications for the autopilot system, CG placement	Elevons and ruddervater mixing	Interfacing the Pixhawk with the flight system, power monitoring antenna, and MicroZed	Ensuring smooth takeoff

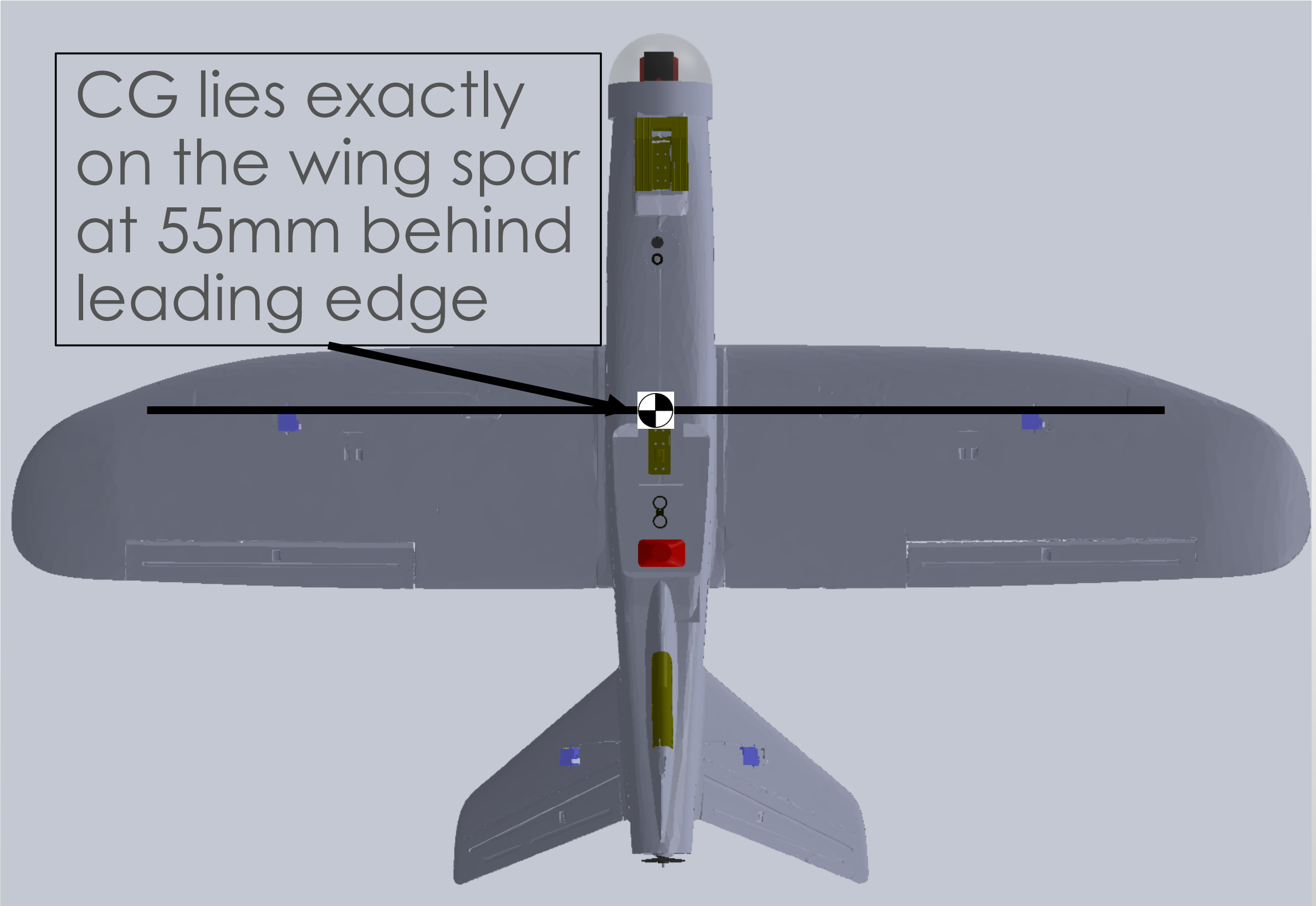
This covers the UAS CPE because completion of the UAS platform will allow the support of the RFI monitoring equipment and support systems that allow for GPS-denied flight

CG PLACEMENT

CG Placement is critical for stable flight

Limited space for internal components and finite wire lengths requires careful placement of components

Will perform MOI calculations to optimize placement of heaviest components near the CG



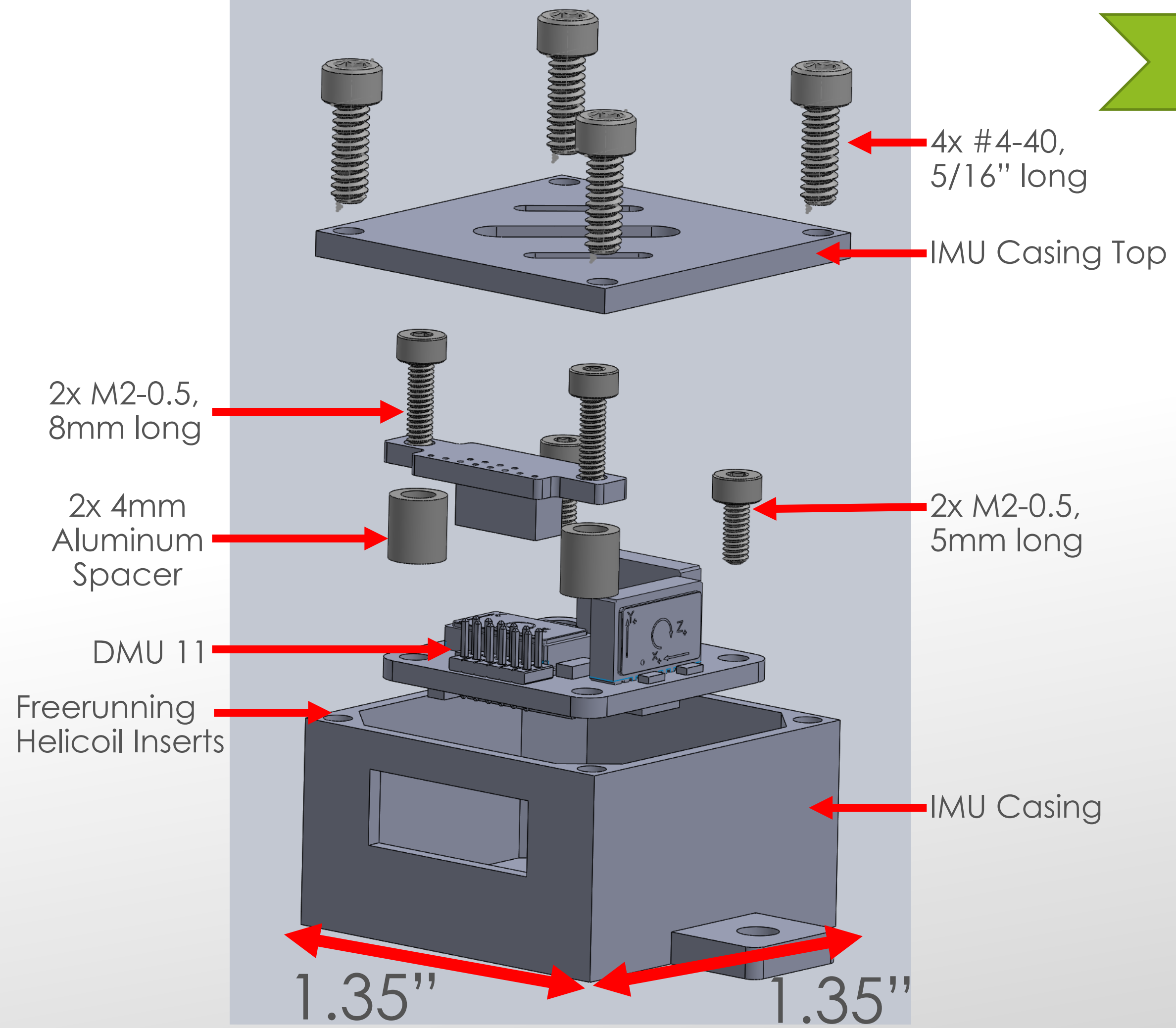
3D PRINTING

- **To save weight, all custom parts are being 3D Printed**
- Components are printed out of **NylonX**- Carbon Fiber Reinforced nylon
- **11** Parts are being 3D printed- Currently **5/11** complete
  - Front and rear skid plates are printed and ready for assembly ✓
  - IMU casing and top are printed and ready for assembly✓
  - Battery bracket is printed ✓
  - Payload structure has been prototyped: 0/6 hours remaining (discussed in later slide)
  - Launch hook required redesign for less overhang material: 0/3 hours remaining
  - Housing for pitot tube: 0/3 hours remaining
- **3 day delay on printing start date** due to issues with 3D printer and material characteristics. **Delay was absorbed in design margin**

IMU CASING ASSEMBLY

Parts Procured → Custom Parts Printed → Subsystem Integrated → Full System Integration

On Track



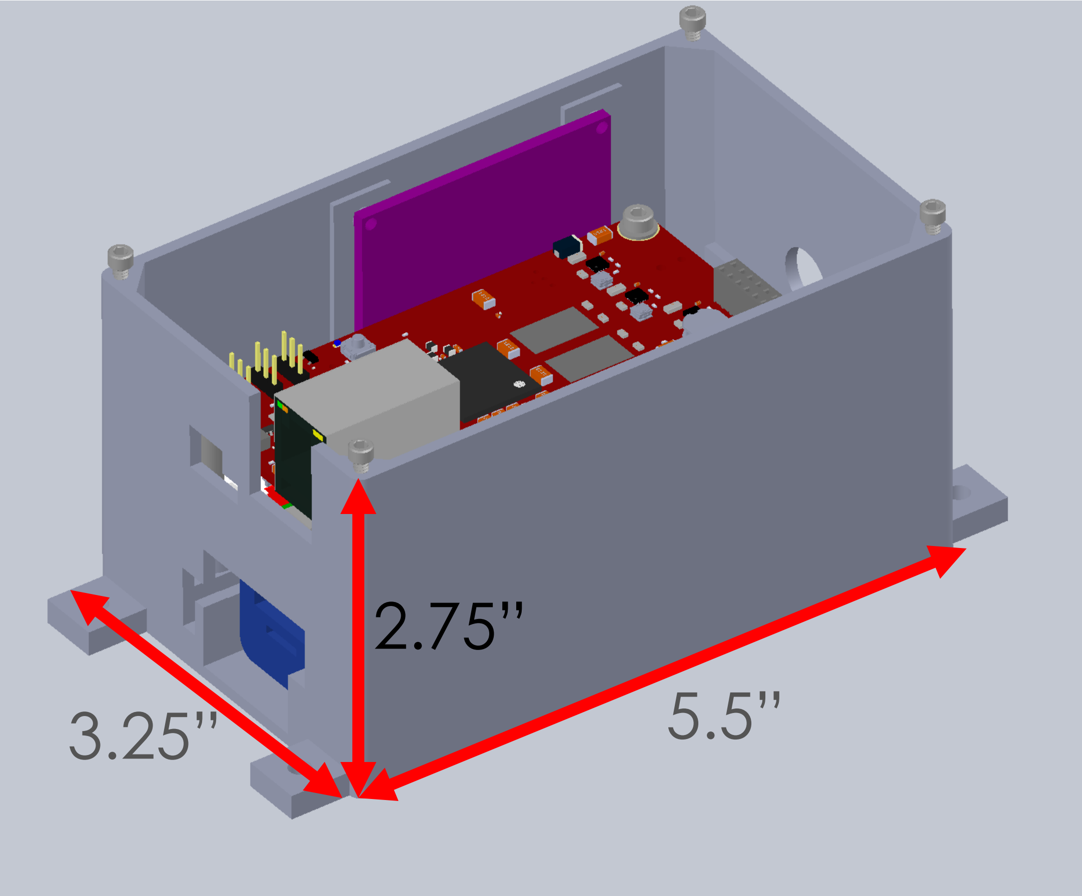
- Unit Testing has already been completed on the DMU 11
- Had to be redesigned since the corners were too small to tap mounting holes

Future Work

- Tap holes on IMU Box
- Subsystem assembly
- Estimated time remaining: 3 hours

PAYLOAD STRUCTURE

Parts Procured → Custom Parts Printed → Subsystem Integrated → Full System Integration



2 Days Behind Schedule
(Accounted for in design margin)

- Main Payload Structure was prototyped with PLA to ensure proper fit of components
- All interior components have been unit tested
- Parts being printed: main structure, battery bracket, casing top

<u>Future Work</u>
- Print main payload structure from NylonX
- Tap holes and attach battery bracket
- Subsystem assembly
- Estimated time remaining: 10 hours

INTERFACE BOARD

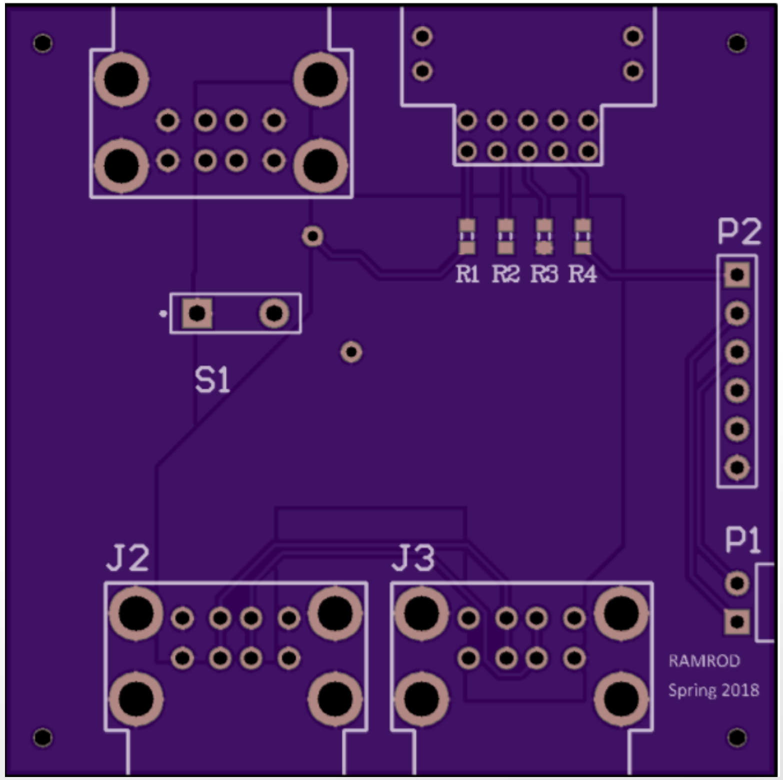
Board Ordered → Unit Testing → Subsystem Integrated → Full System Integration

On Track

Battery USB

LTE Ethernet

Master Switch



UART PixHawk

UART to MicroZed

MicroZed USB

MicroZed USB Programming Interface

- Power splitting
- LED indicators
- Board ordered from Osh Park Circuits in Oregon.
- Main Concerns
  - Arrival Date ~ Feb 9-15
  - LED Connector reliability

Future Work

- Soldering connectors
- Testing for reliability
- Estimated time remaining: 4 hours

PAYLOAD STATUS

Hardware	Status	Work to be done
Cell Modem/Pocket PORT	Data sent over ethernet and WiFi connection	Unit testing Cell plan procurement Send data using LTE
GPS Antenna	Has been connected to NT1065 and receives GPS readings	Full system testing
MicroZed/NT1065	Main function threads written Hardware configured	Connection to PixHawk written and tested • Data in • Data out
Interface Board	Designed and ordered (delivered Feb. 9-15)	Soldering Functionality testing
Battery	Ordered	Power draw and time testing

Time remaining: 12 hrs

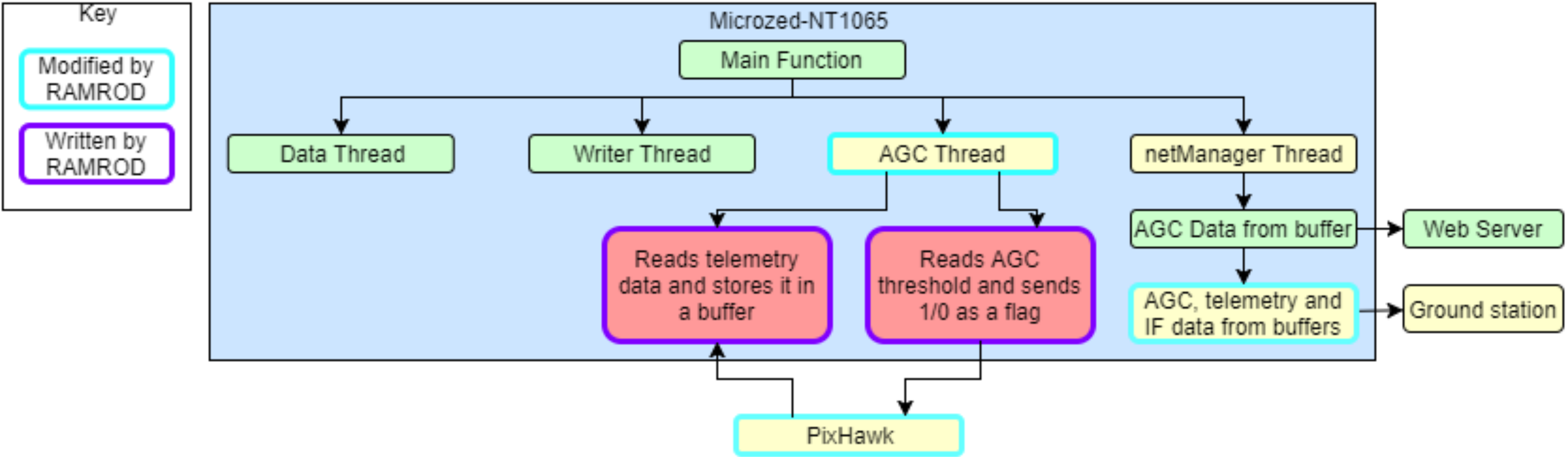
Time remaining: 4 hrs

On Track

Parts Procured → Custom Parts Manufactured → Subsystem Integrated → Full System Integration

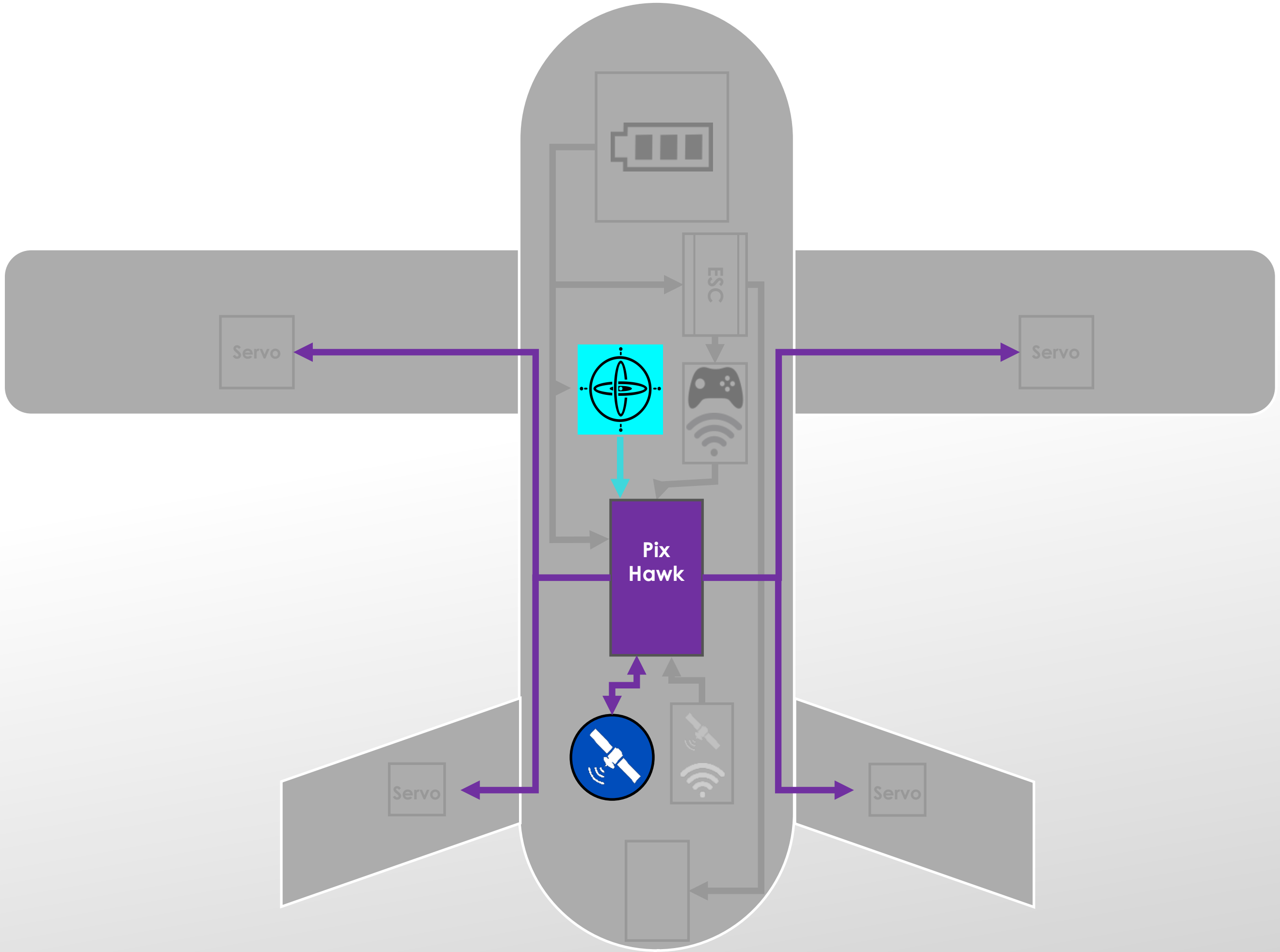
SOFTWARE

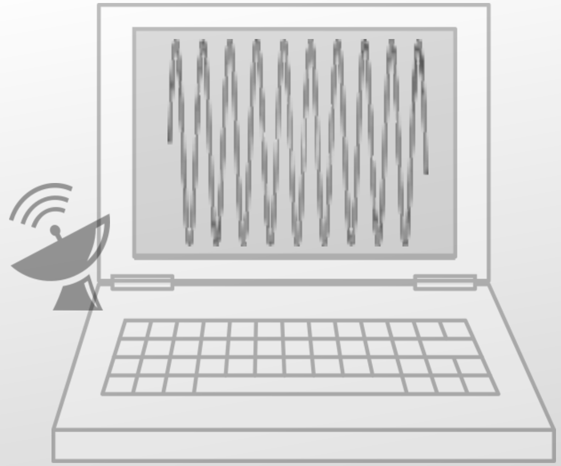
MICROZED CODE



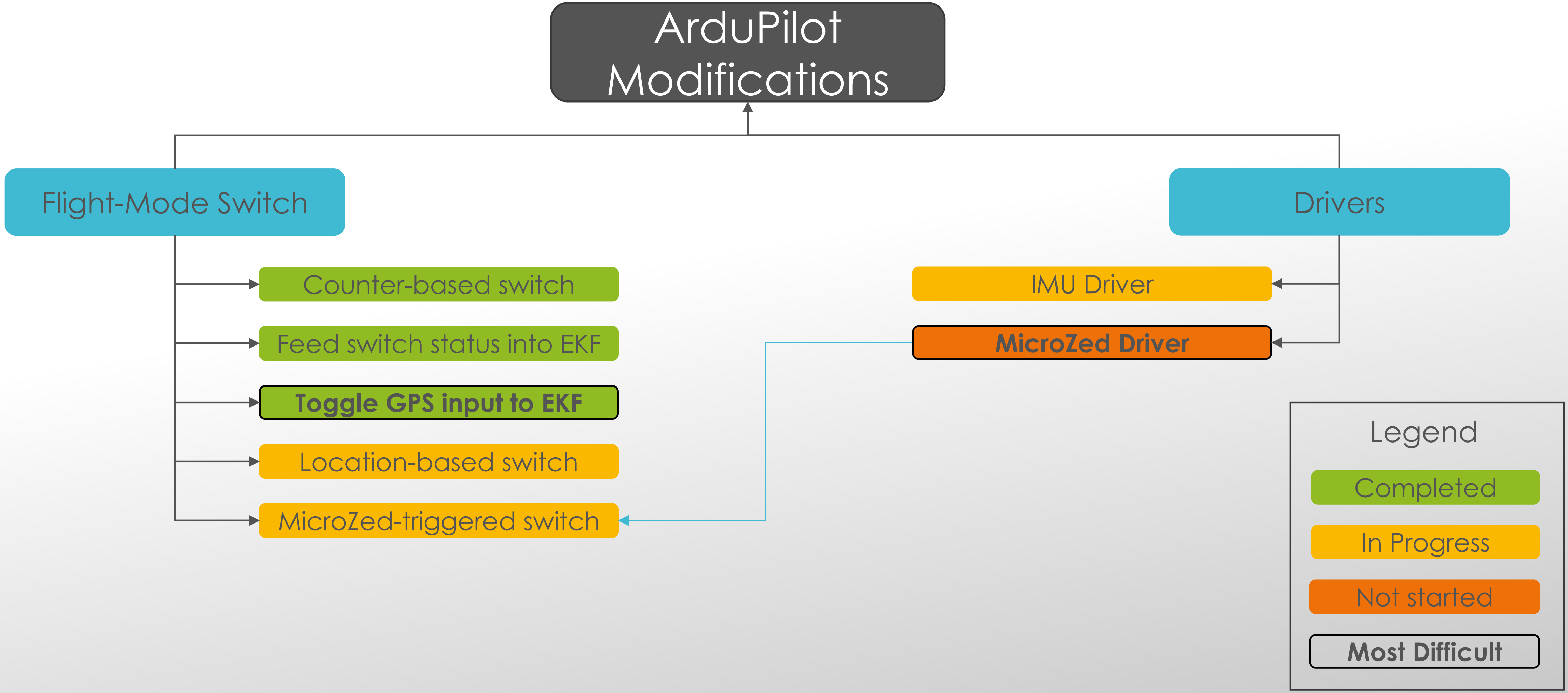
Time remaining:
12 hrs

FLIGHT SYSTEM

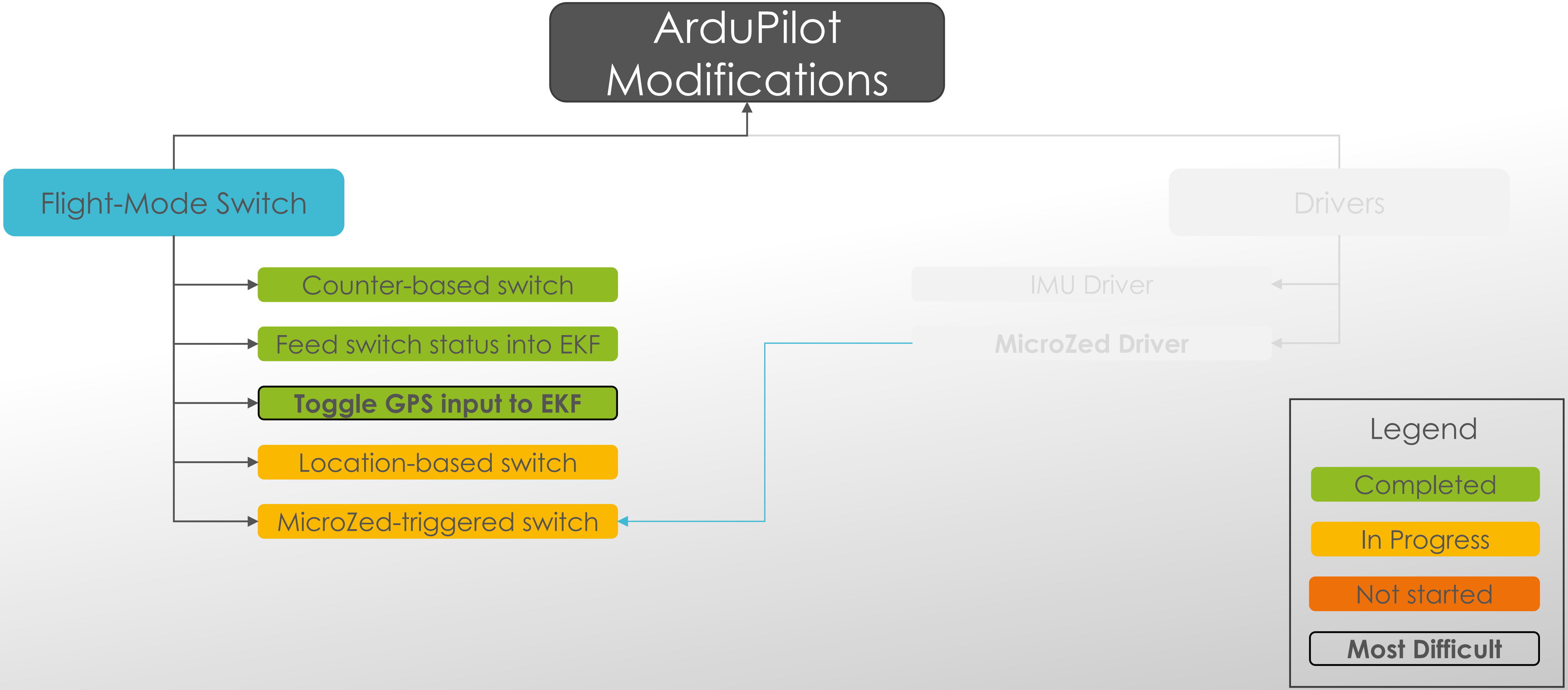




ARDUPILOT CODE MODIFICATIONS



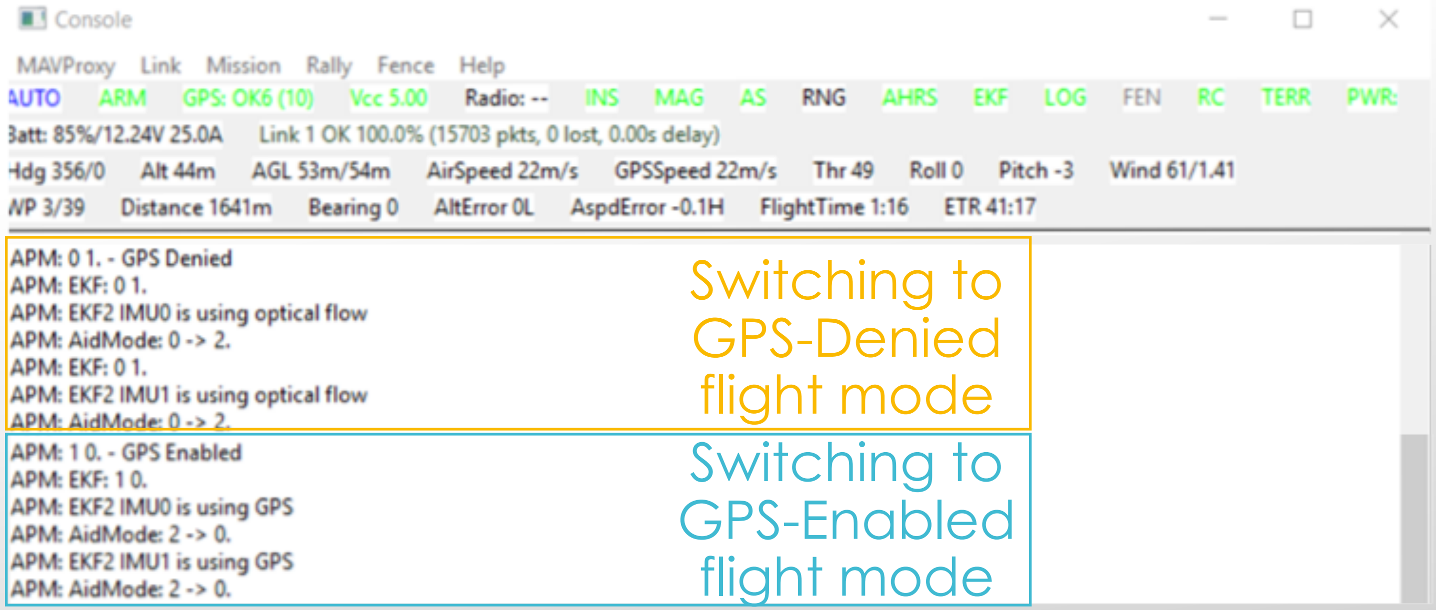
ARDUPILOT CODE MODIFICATIONS



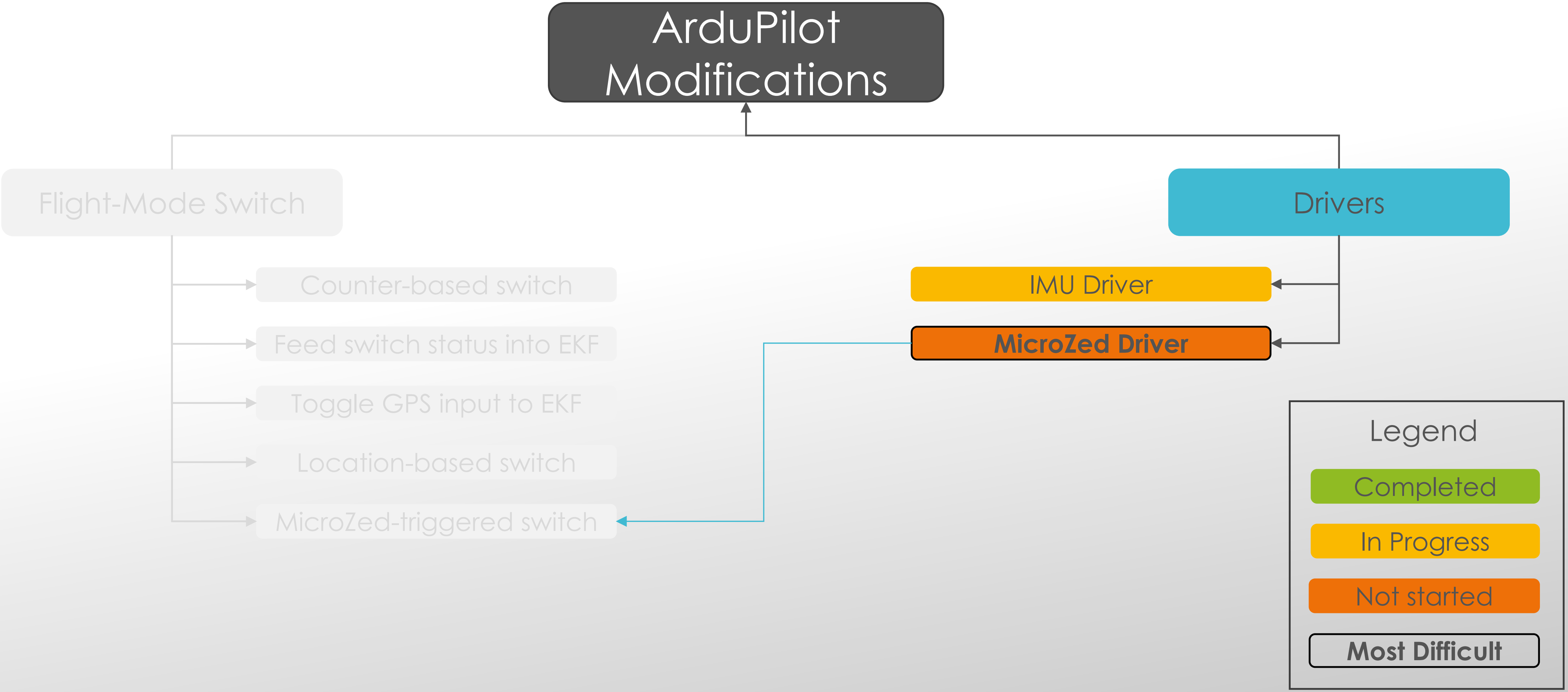
FLIGHT-MODE SWITCH STATUS

Function/Step	Status	Work to be done
Switch autopilot flight modes and update **switch status** based on counter	Completed + Tested using SITL: • Correct outputs to Ground Station console • Timing executed correctly • Switches modes and EKF inputs seamlessly based on switch-status command	HIL verification: • Used to verify the same metrics as SITL, running on PixHawk hardware Requires flight testing to assess actual drift and performance of EKF
Pass **switch status** into EKF background thread		
Toggle GPS and Optical Flow input to EKF navigation based on **switch status**		
Replace counter with **location-based** switch to simulate mission-level testing while adhering to FAA regulations	• Front-end switching code completed • 500-meter 'GPS Denied' radius established based on GPS coordinates	• Need to finish code passing switch status to front-end • Need to test full 1-hour flight-plan using SITL and HIL Time remaining: 5 hrs
Integration of Payload: Switch between **GPS-denied** and **GPS-enabled** flight based on trigger from Payload	Switching code completed	• Requires Payload driver for input • HIL testing with Payload Time remaining: 5 hrs

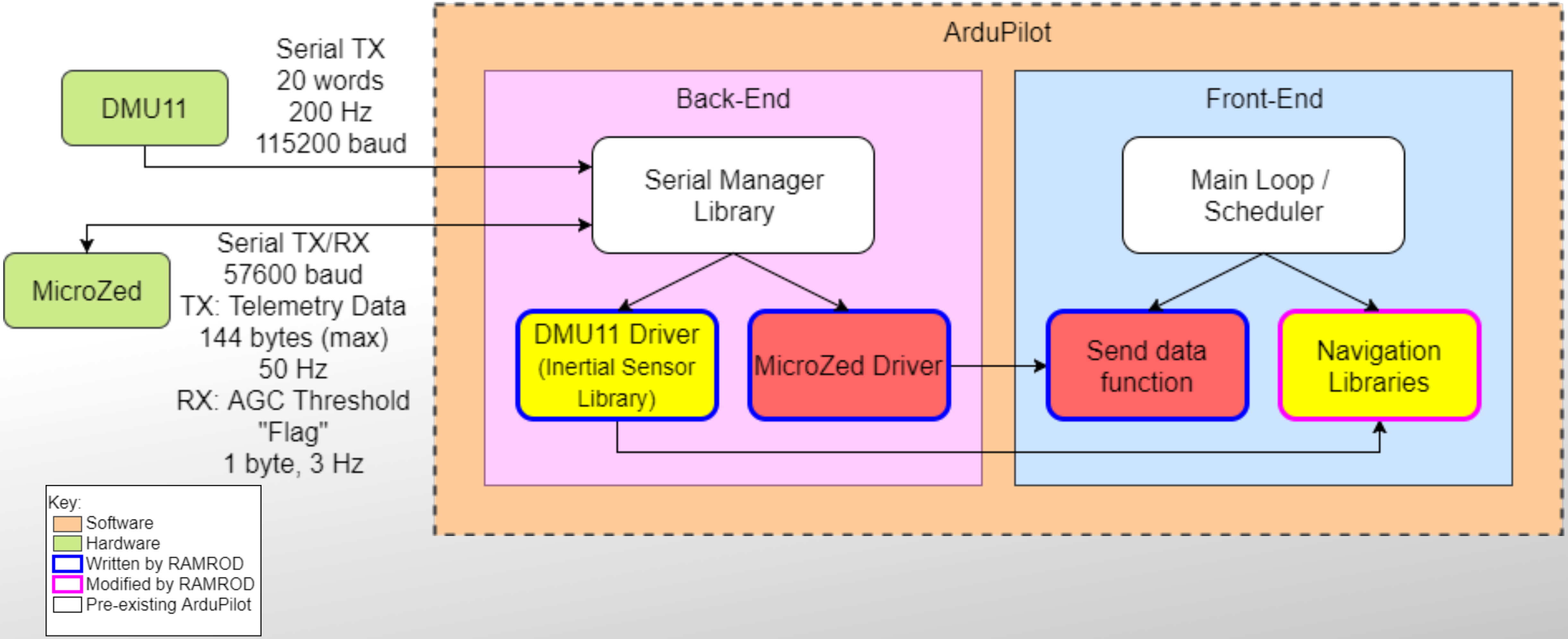
ARDUPILOT MODIFICATIONS



ARDUPILOT CODE MODIFICATIONS



DRIVER ARCHITECTURE



DMU11 DRIVER STATUS

Driver Function	Status	Work to be done
Establish UART connection to DMU11: • 40 bytes at 115200 baud every 200 Hz	Code written to configure, begin, and detect the serial connection	Unit Test: • Read DMU11 data into ArduPilot back-end • Send data to the PixHawk console • Verify data by turning the DMU11 90° and checking acceleration values
Read data from UART buffer	Code written to read available bytes and save inertial data to a struct	
Update DMU11 device status	Code written to update the device health based on time between samples	
Read DMU11 status and data from the driver back-end • 400 Hz	Function written to read IMU data to front-end at 400 Hz	Unit test: • Output data to console from front-end • Verify with 90° turn test again
Forward data to the navigation libraries • 400 Hz	Not started Time remaining: 15hrs	Write code to access the DMU11 data in the navigation libraries. Unit Test: • Run HIL simulation and examine log files to verify use of DMU11 data

ARDUPILOT/MICROZED DRIVER STATUS

Driver Function	Status	Work to be done
Establish UART connection to MicroZed: • 57600 baud • RX one byte at 3 Hz • TX up to 144 bytes at 50 Hz	Not started	Write code to read in and send data Unit test: • On PixHawk: write to console to verify received byte matches byte sent by MicroZed • On MicroZed: write incoming MAVlink message to the command line to verify message contents
Read in "flag" from UART buffer from MicroZed	Not started	
Construct and send MAVlink data packet to MicroZed	Not started	
Poll incoming AGC feedback flag • 400 Hz	Not started	Write code to read flag into RAMROD switch function Unit Test: • Run HIL simulation to verify switch upon reception of flag
Send telemetry data packet to MicroZed • 50 Hz	Not started	Write code to command the back-end to send telemetry data at 50 Hz Unit Test: • Compare data received by MicroZed to PixHawk log files to verify telemetry message contents • On MicroZed: time-stamp incoming data to verify 50 Hz frequency of transmission

Time remaining:
20-25 hrs

PROCUREMENT STATUS AND BUDGET

PROCUREMENT OVERVIEW

Obtained

- XUAV Talon as well all internal components
- Transmitter and Reciever
- NylonX Material
- LTE Modem

In Progress

- Support Hardware for Talon Build
- Indicator PCB
- PocketPort
- Launcher Components

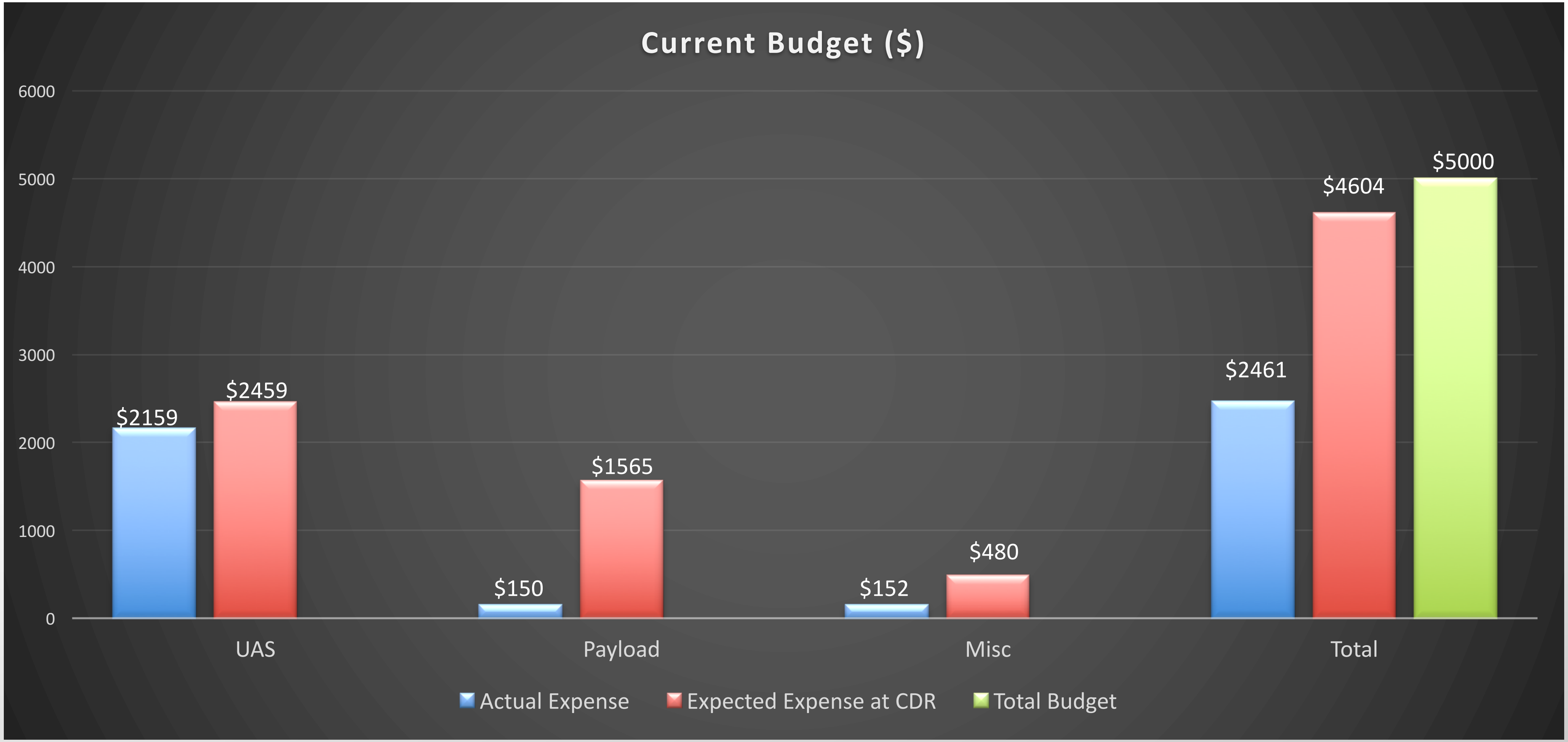
Not Ordered

- 2nd LTE Modem

UAS Procurement

Payload Procurement

CURRENT EXPENSES



COST PLAN

System	Current Expense	Expected Expense	Margin	Total
UAS	$2159	$2259	$250	**$2509**
Payload	$150.20	$284	$1150	**$1434**
Support Hardware	$151.66	$250	$250	**$500**
Overall	$2461	$2793	$1650	**$4443**

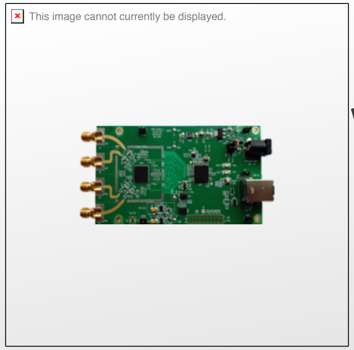
Remainder: $557

QUESTIONS?

Backup Slides

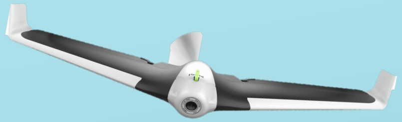
TESTING PLANS

Jamming GPS bands is illegal



Sampling WiFi band would require manufacturing an additional payload

Map WiFi power profile using Parrot Disco

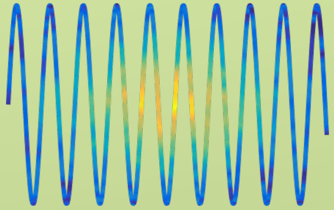


Download data

AGC Data

Position Data

Localize signal source in post-processing



Use AGC Profile to trigger locational flight-mode switch on Talon

POSITION/VELOCITY AIDING MODES

By default, if no GPS/Optical Flow available, ArduPilot attempts to go into **constant position** or **velocity** mode to constrain tilt drift

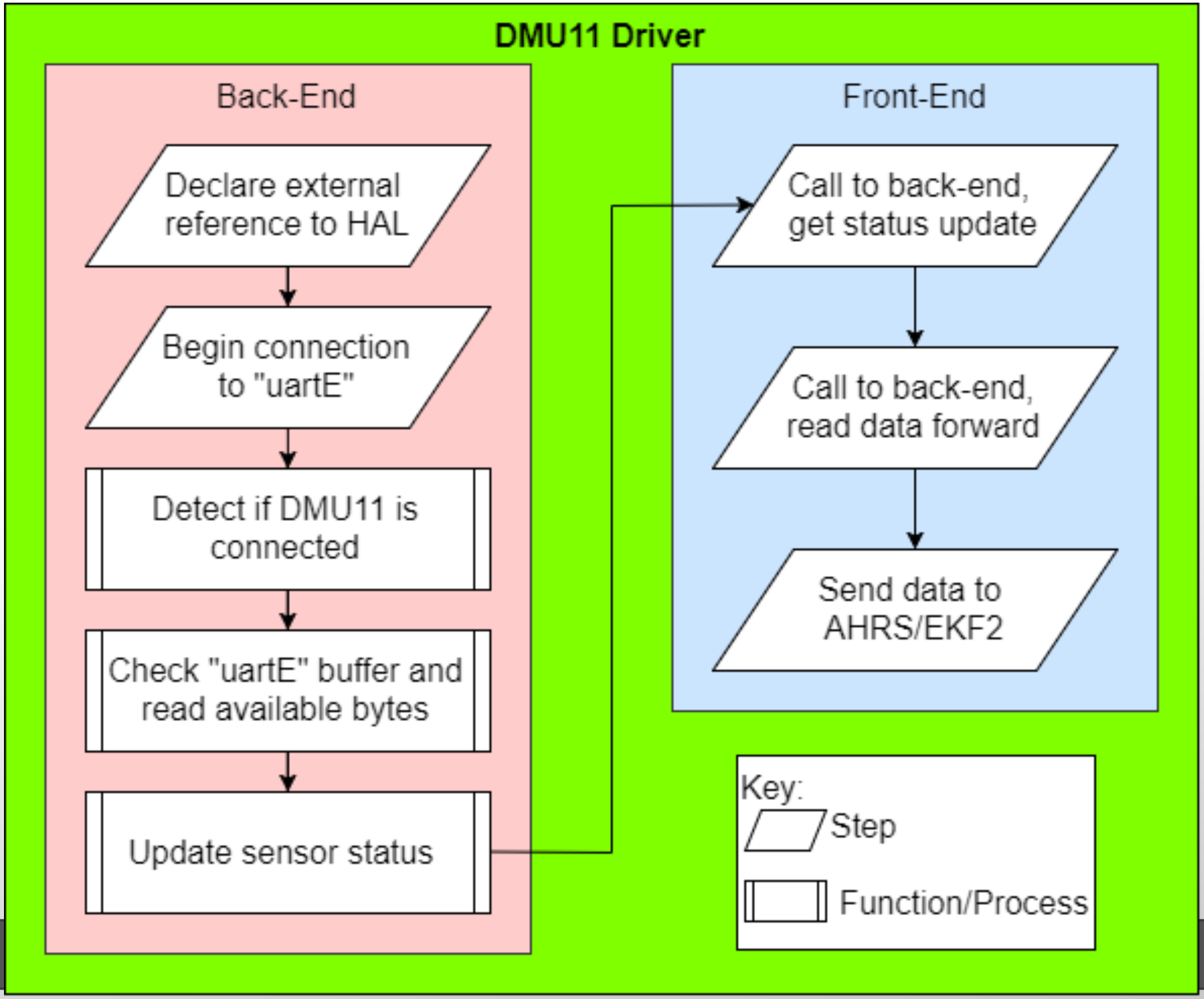
Changes:

AID_ABSOLUTE (Mode 0)
- Allow GPS input to EKF

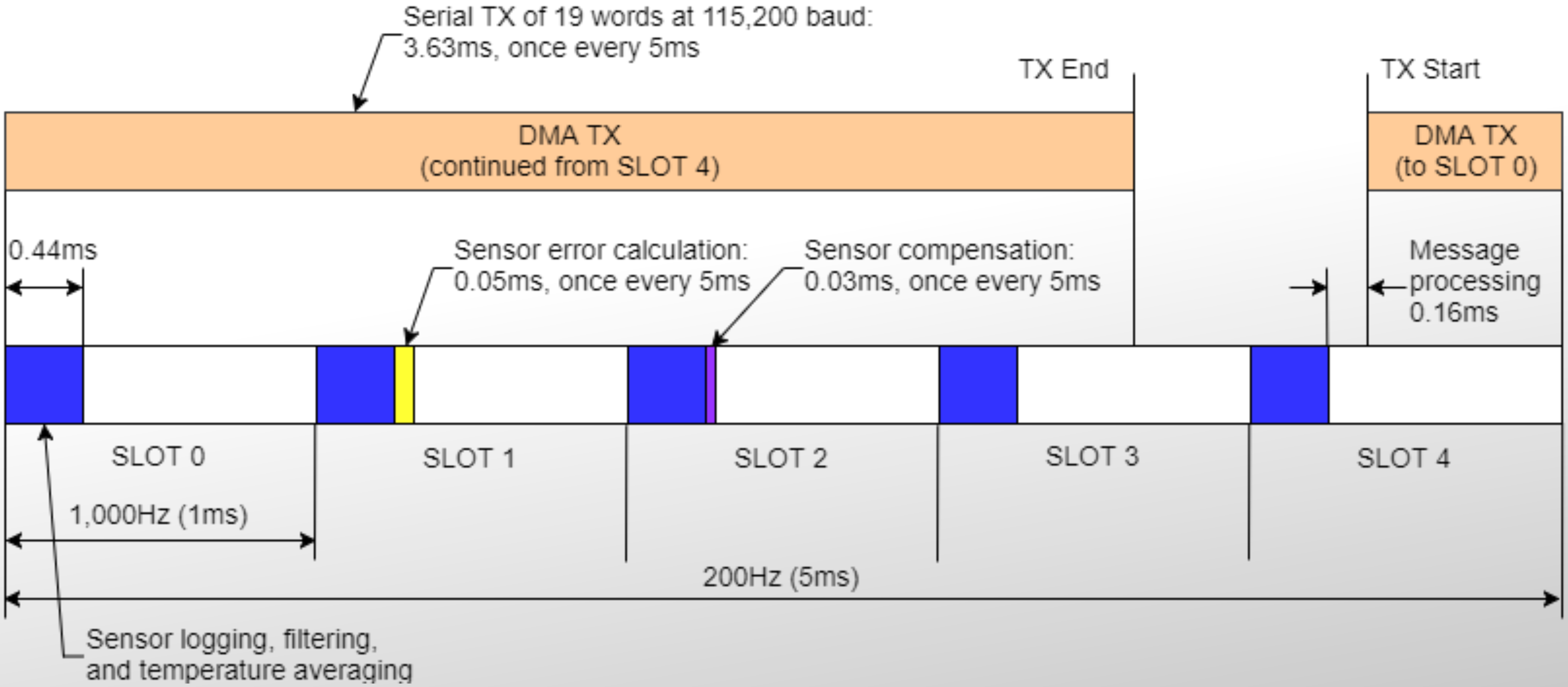
AID_NONE (Mode 1)
- No GPS or Optical Flow inputs to EKF
- Only use IMU, mag, baro, velocity sensor, etc

AID_RELATIVE (Mode 2)
- Allow Optical Flow but not GPS input to EKF

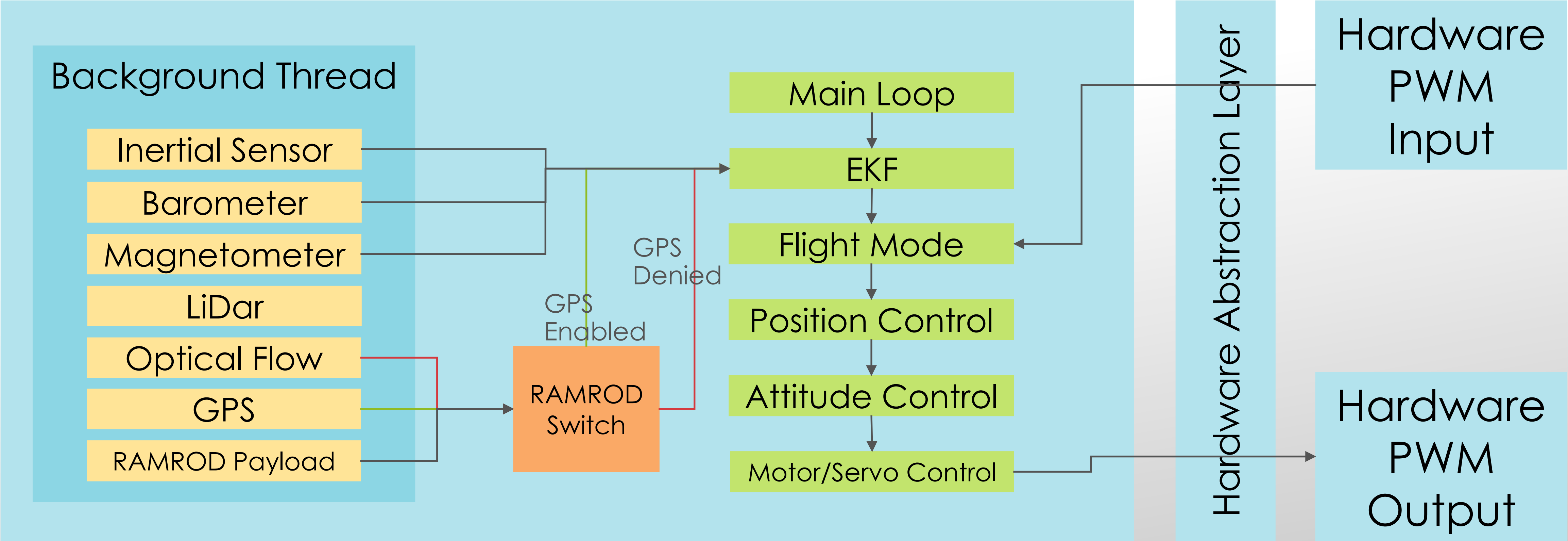
DMU11 DRIVER FLOW-CHART



DMU11 DATA TIMING

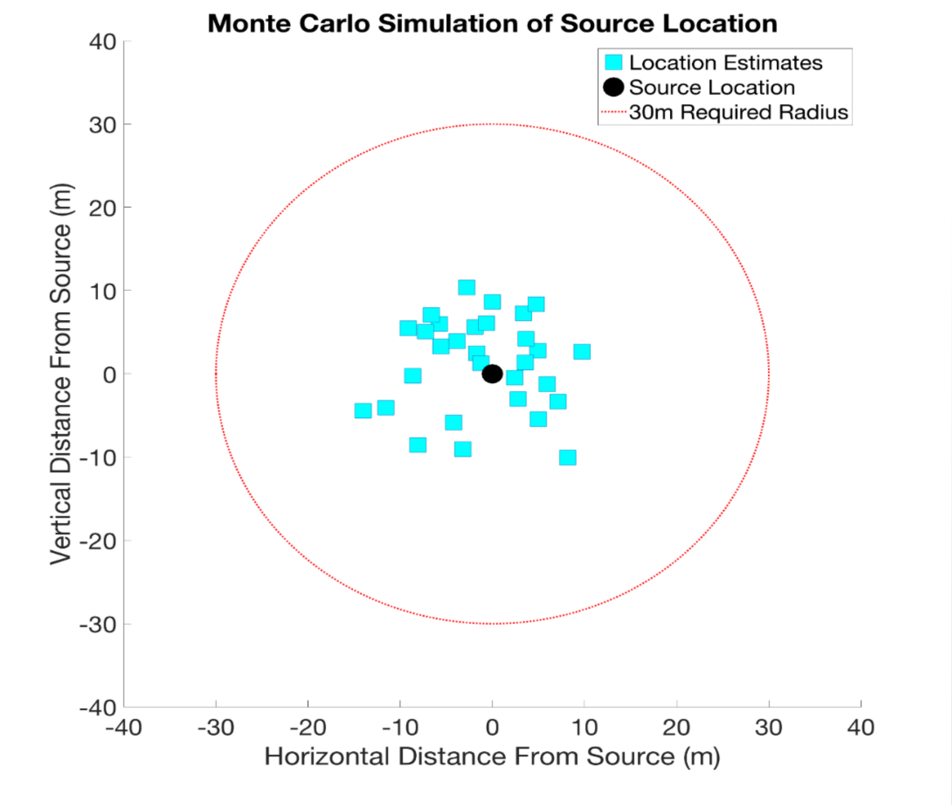


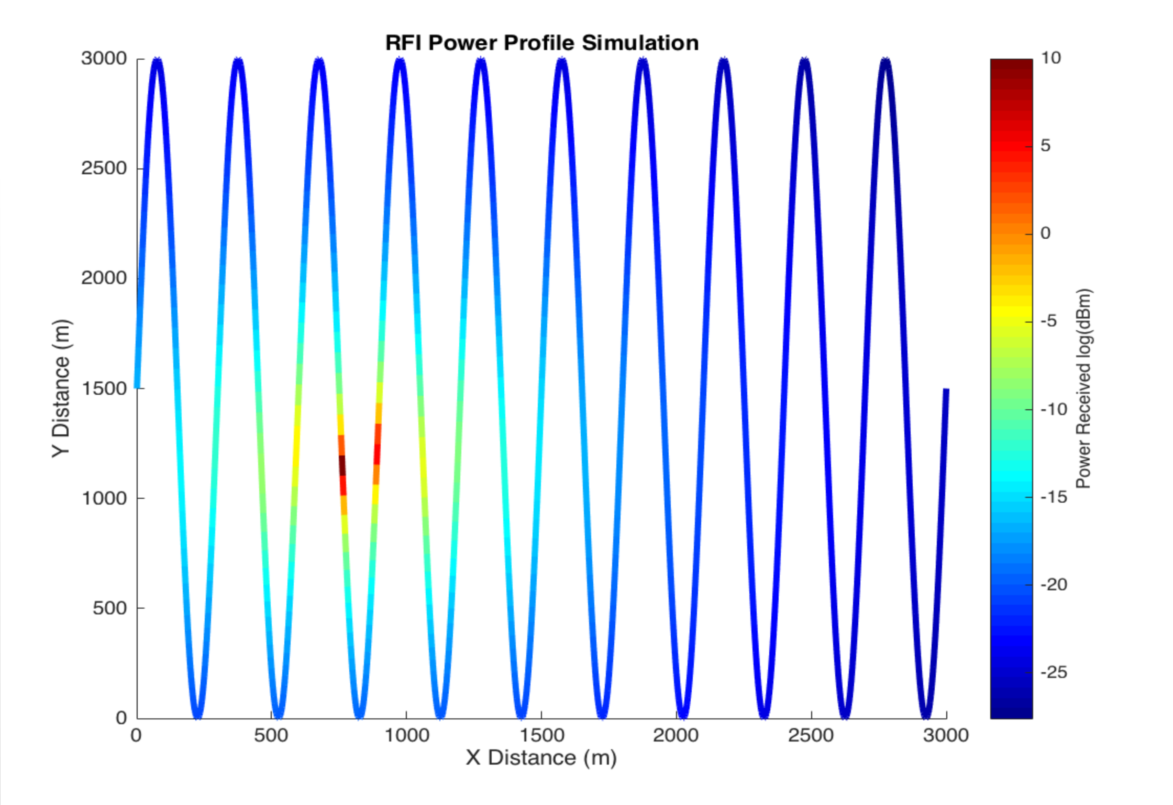
ARDUPILOT MODIFICATIONS



UNIT TEST 1: PDOA

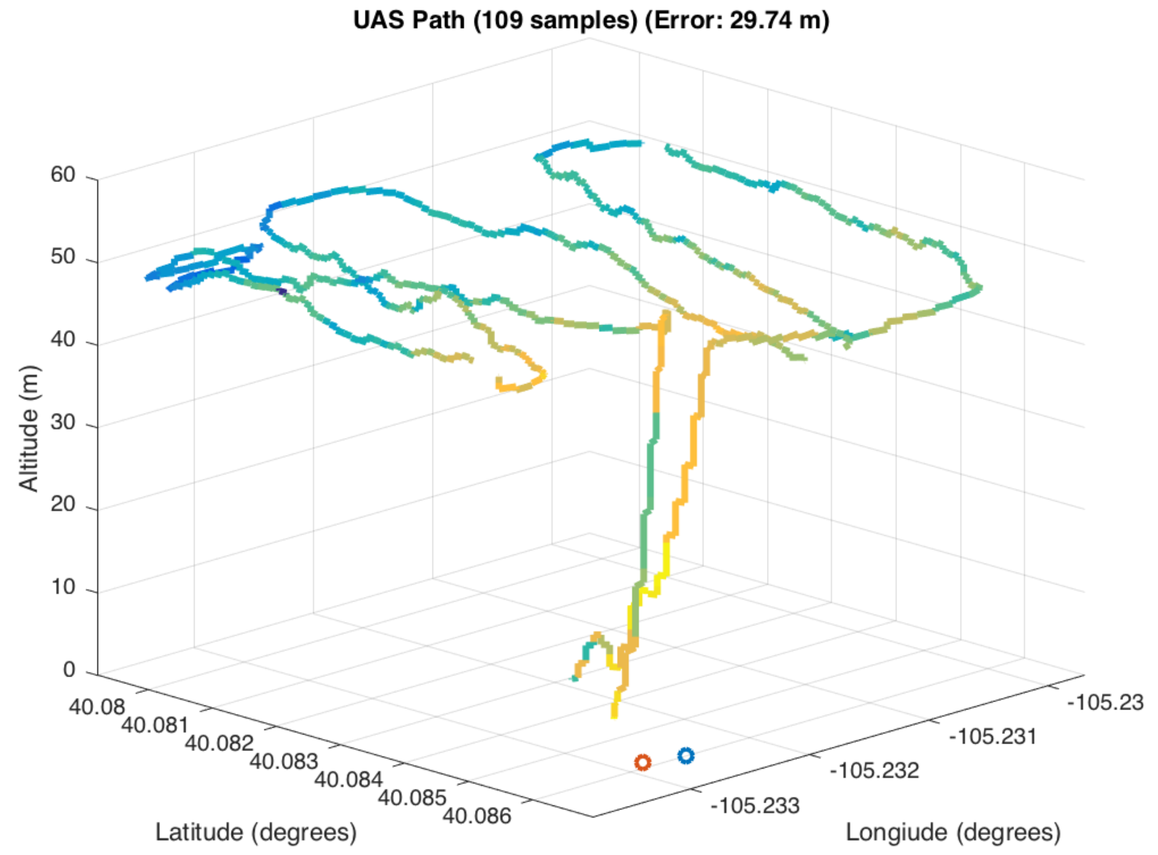
- Simulated data used to prove concept of localization by Power Difference of Arrival (PDOA)



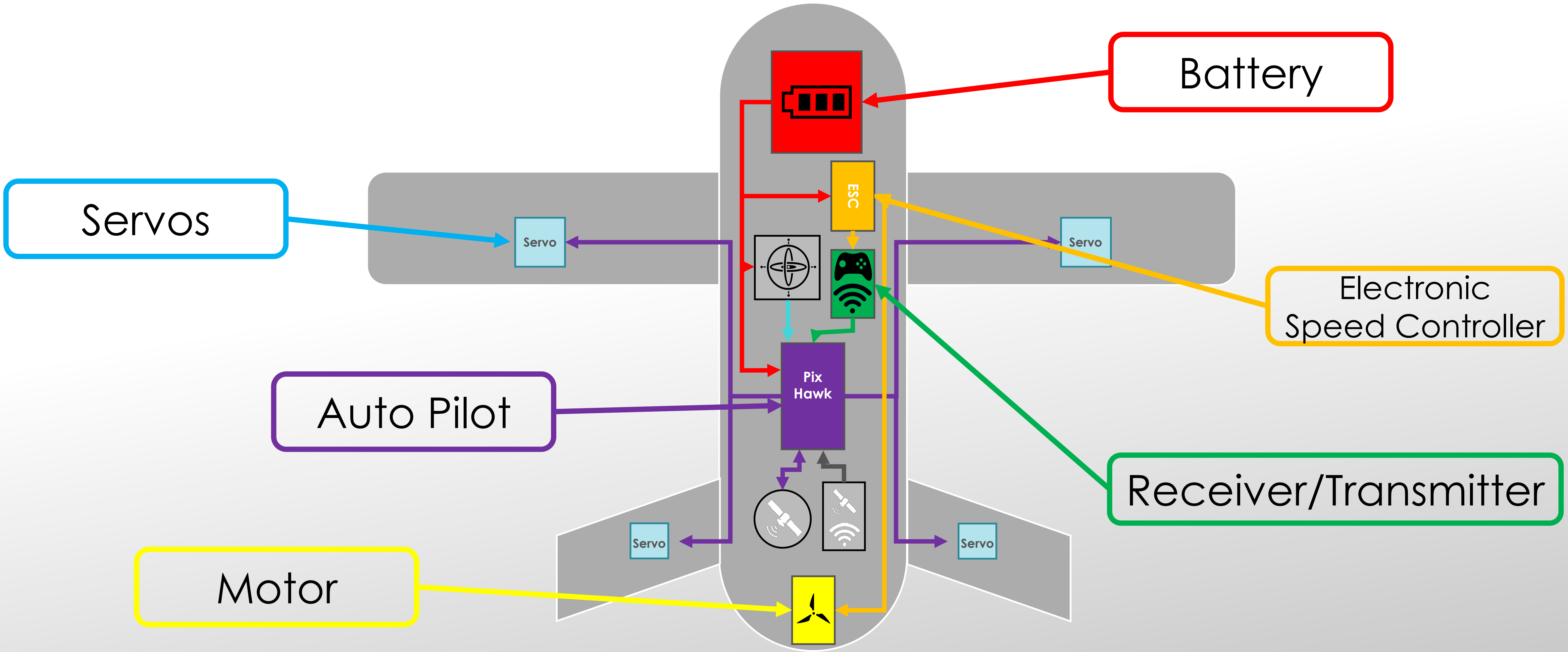


UNIT TEST 2: PDOA

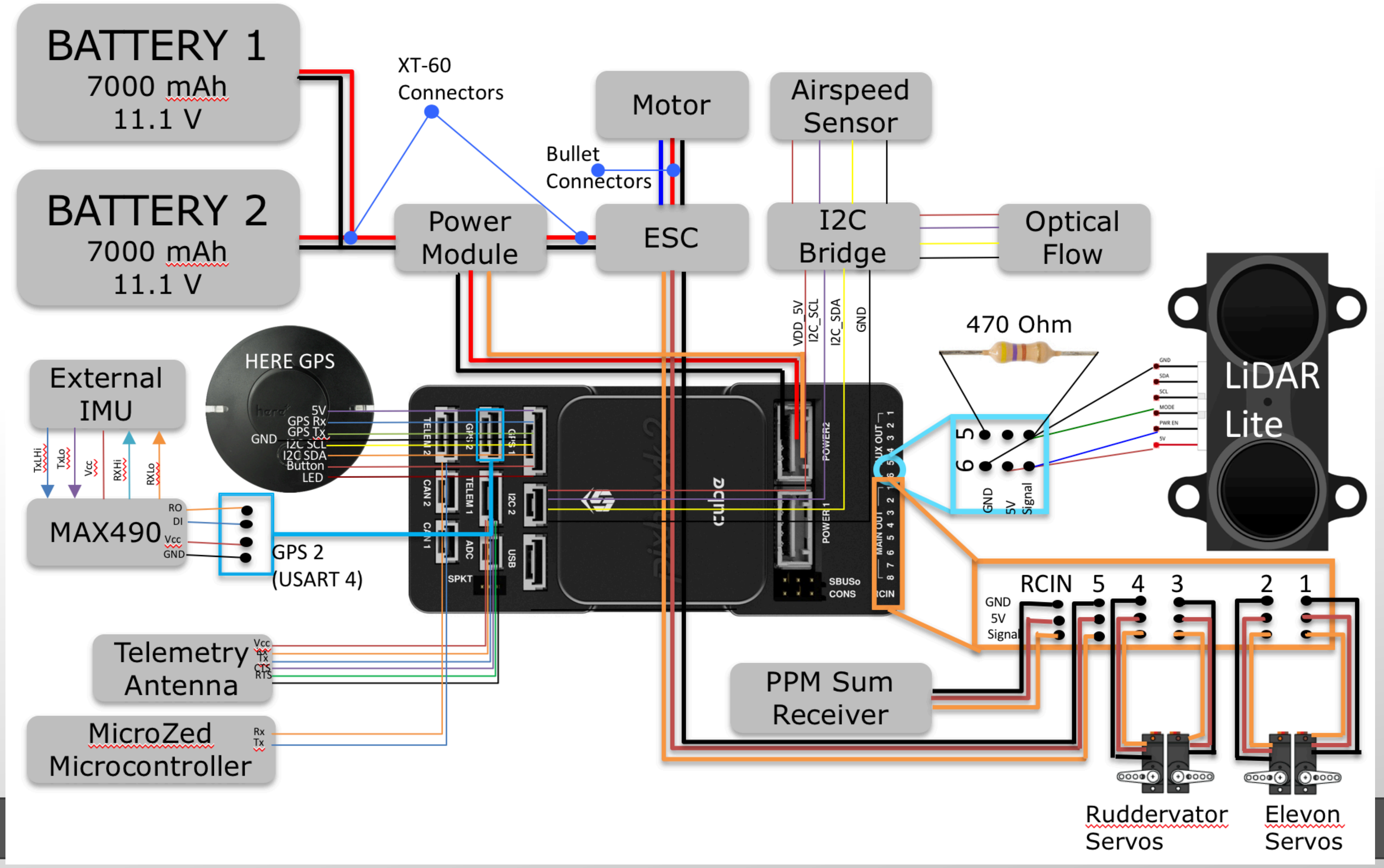
- Unit test used real AGC data + GPS position to test localization functionality and accuracy.
- Results: Accurate localization within 40m with only 109 samples



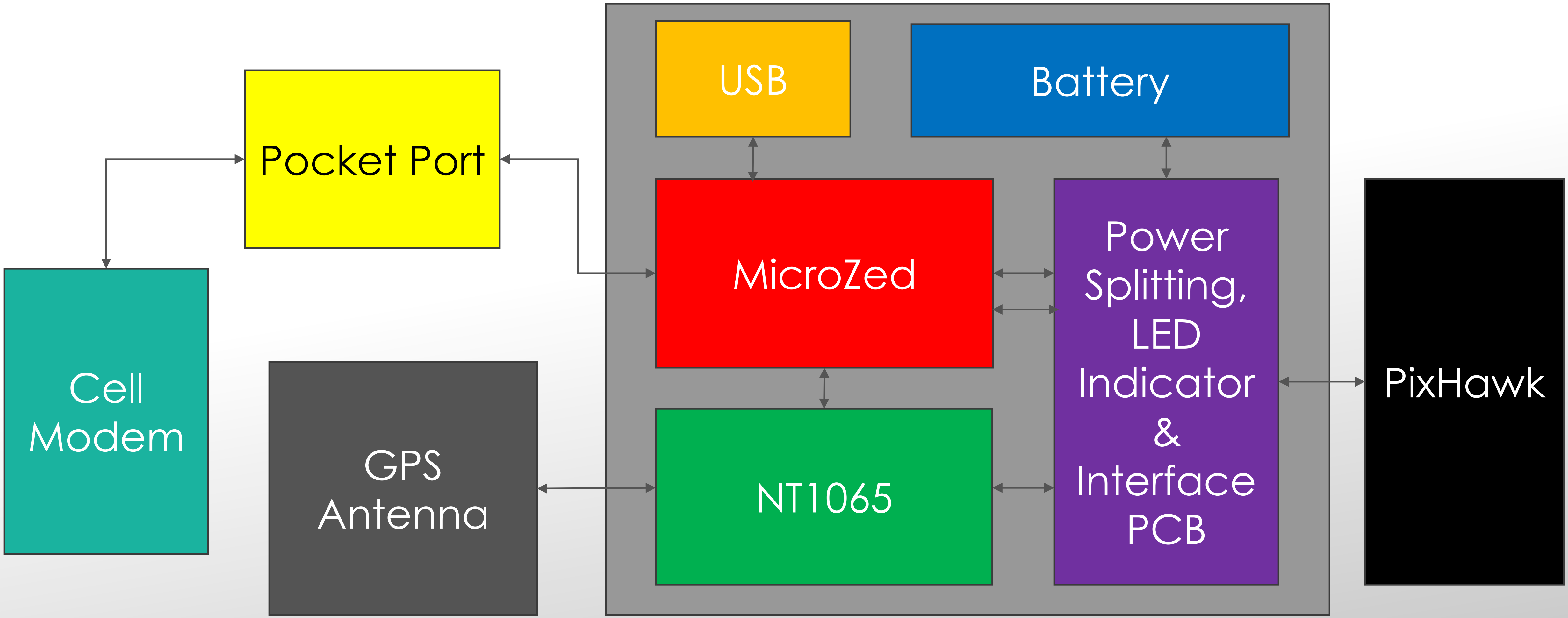
TALON BASELINE DESIGN



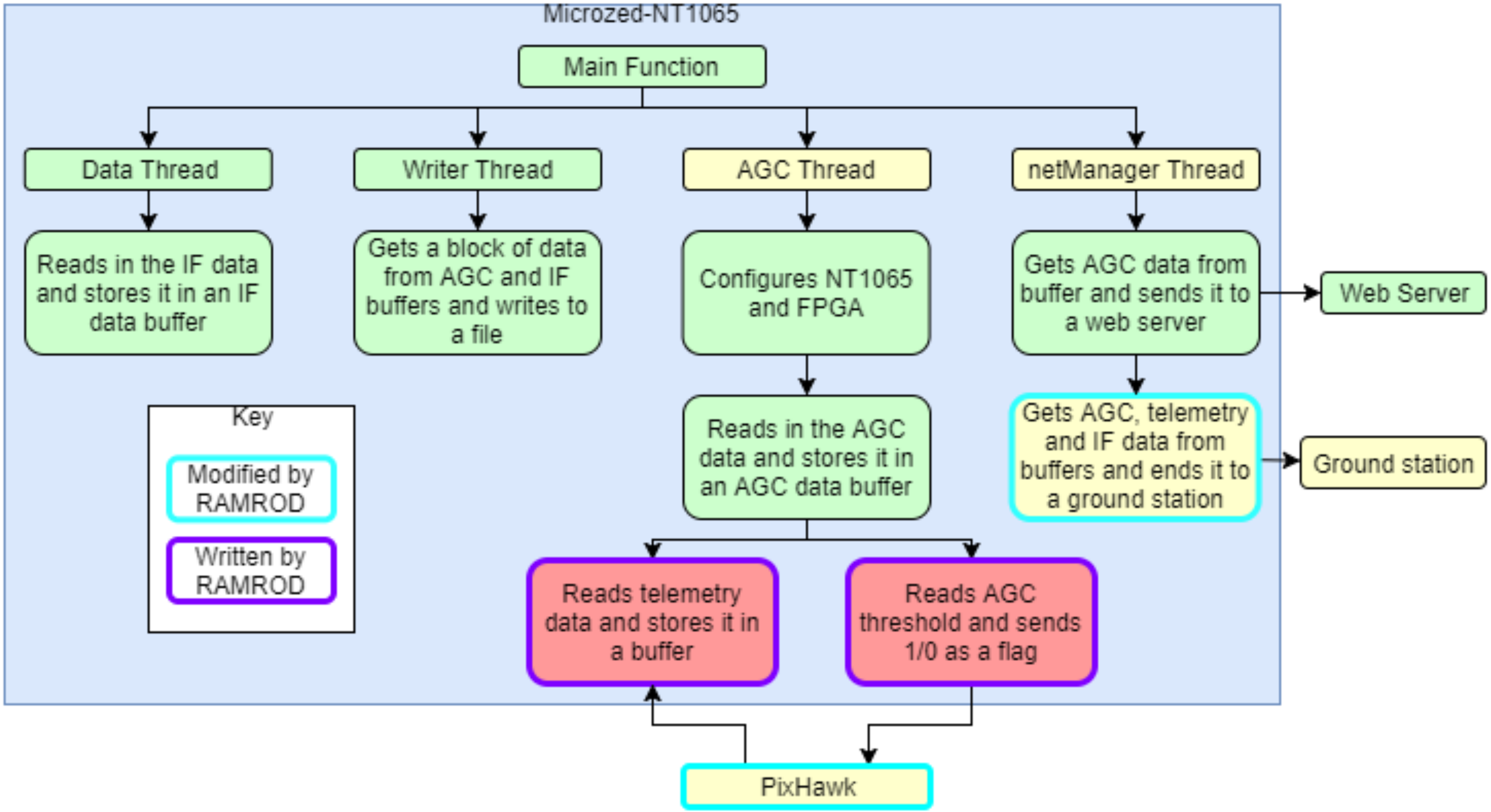
UAS WIRING DIAGRAM



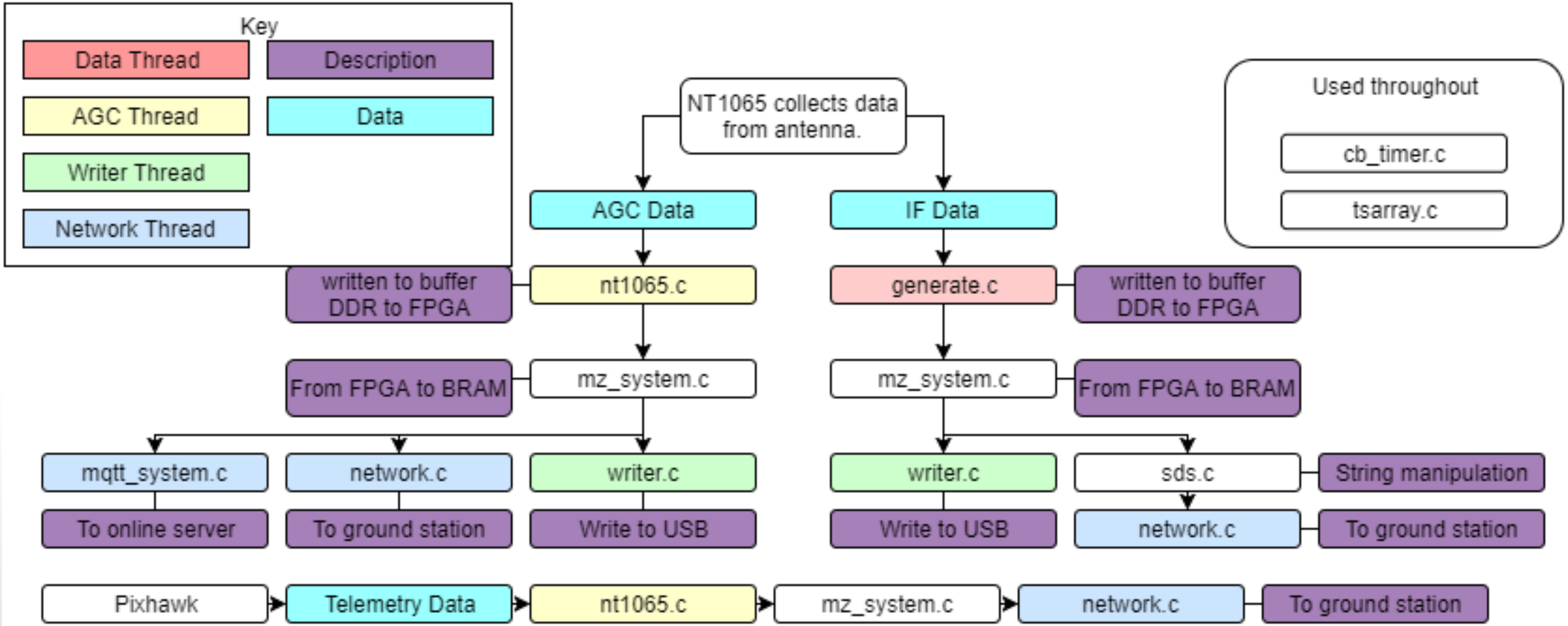
PAYLOAD BASELINE DESIGN



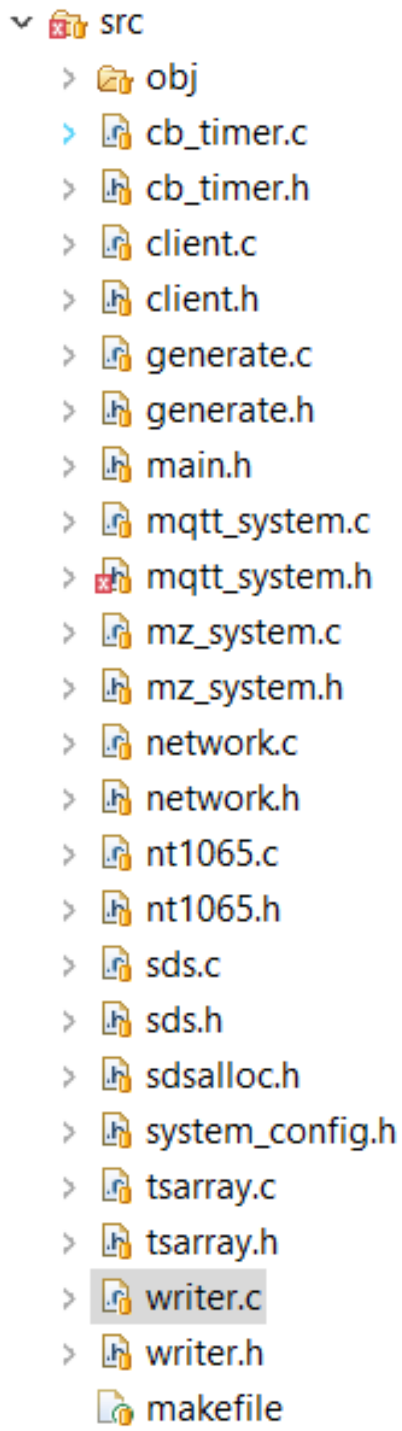
MICROZED CODE



MICROZED CODE



Basic Code Structure



cb_timer.c	Handles timing - everything syncs up.
~~client.c~~	~~Main function - Declares threads~~
generate.c	Generates fake data - No real AGC & IF interference
mqtt_system.c	Sets up messaging and server connection - Real time AGC data
mz_system.c	Maps DDR to BRAM (ie FPGA to BRAM)
network.c	Dynamically sends data to an IP address
nt1065.c	Spi connection between MZ and NT, configures NT1065 and FPGA
sds.c	Makes C strings easier to manipulate
tsarray.c	Sets up buffer and formats how to use them
writer.c	Write from buffer to data file

Main Function

The 4 threads run in parallel on the Microzed Linux computer

Data Thread-IF

- Gets IF data
- Transfer data from BRAM to DMA Buffer
- Transfers from DMA Buffer to IF Buffer
- Closes files.

Writer Thread

- Get block(256kB) of data from the buffer.
- Writes data to a file with a unique name
- Records elapsed time
- Closes current file and repeats the process

AGC Thread

- Opens log file
- Initializes SPI mapping between NT and Microzed
- Reads in data, stores in buffer
- Unmaps locations
- Adds data to buffer
- Close file
- Configure NT1065
- Configure FPGA

NET Manager Thread

- Maps addresses to a web server
- Get data from queue
- Takes in an input file, Inputs filename, opens the file
- Sets output filename, writes to output file
- Closes all files
- Free strings
- Disconnect from broker

BATTERY

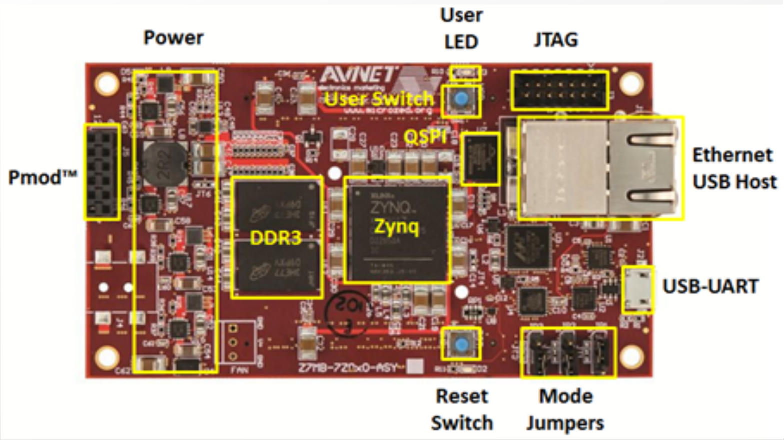
Concerns

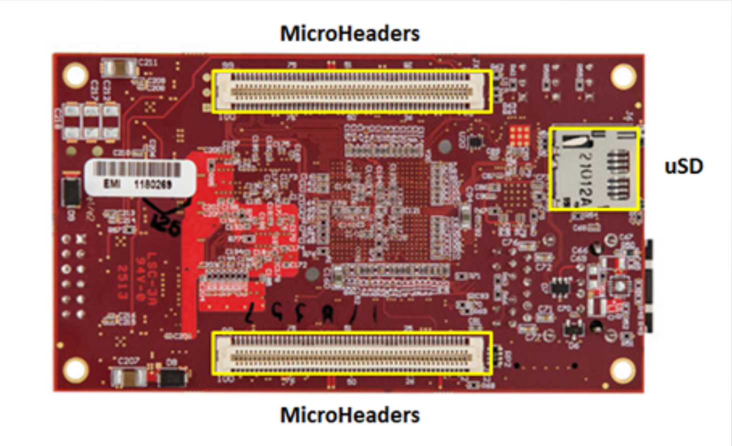
- Microzed power draw with NT Board and cellular modem may be large enough to drain battery in under an hour.
  - RAMROD has chosen the battery so we would have at least a 200% margin of error

MICROZED AND NT BOARD

Potential Concerns

- RF Interference with the NT Board
- Large power draw from NT Board
- Supplying the Microzed enough power to fully operate





INTERFACE BOARD TIMELINE

November 2017:
- Block diagram and connections defined

December 2017:
- Layout schematics and PCB in Altium
- Order connectors from Samtec

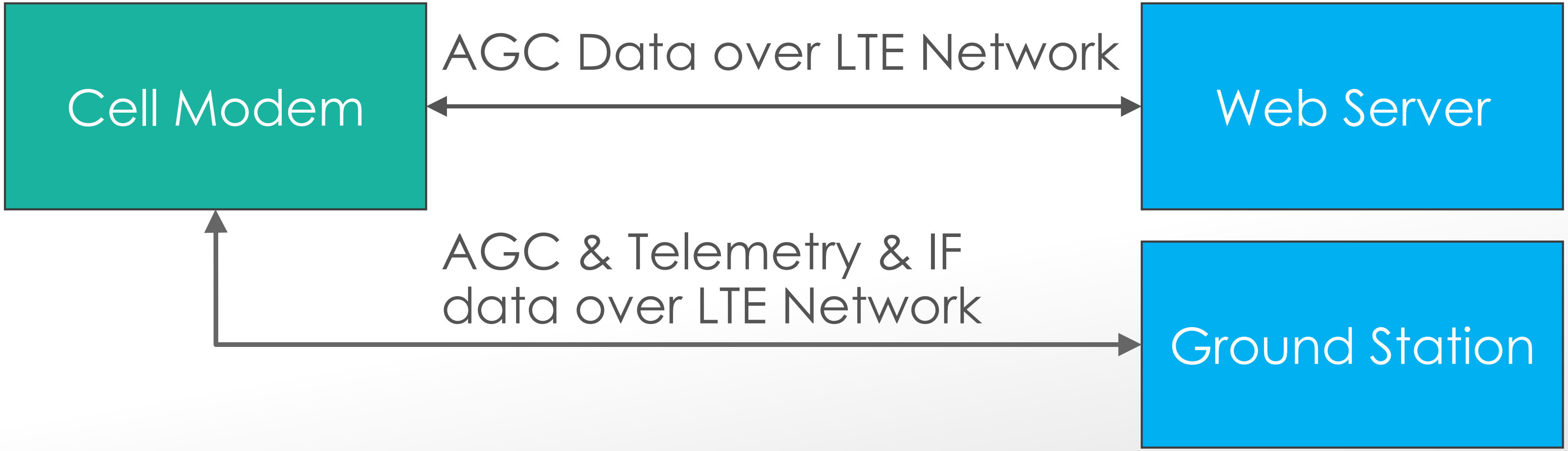
January 2018:
- Test LED Circuit design on breadboard
- Finalize PCB Layout
- Order PCB from Osh Park Circuits

February 2018:
- Test LED circuitry
- Test with Payload system

UART CONNECTION: PIXHAWK AND MICROZED

- Data In
  - Collect a data stream with telemetry data to go to ground station
  - New function pulling data from Pmod through ttyps0 or ttyps1
  - Send through BRAM (not FPGA) to send through network.c to the ground (no need to save to USB)
- Data Out
  - Send an indicator flag to PixHawk when AGC threshold is reached to switch between GPS and GPS denied flight
  - In AGC thread (open file, send data(1,0), close file) (nt1056.c)

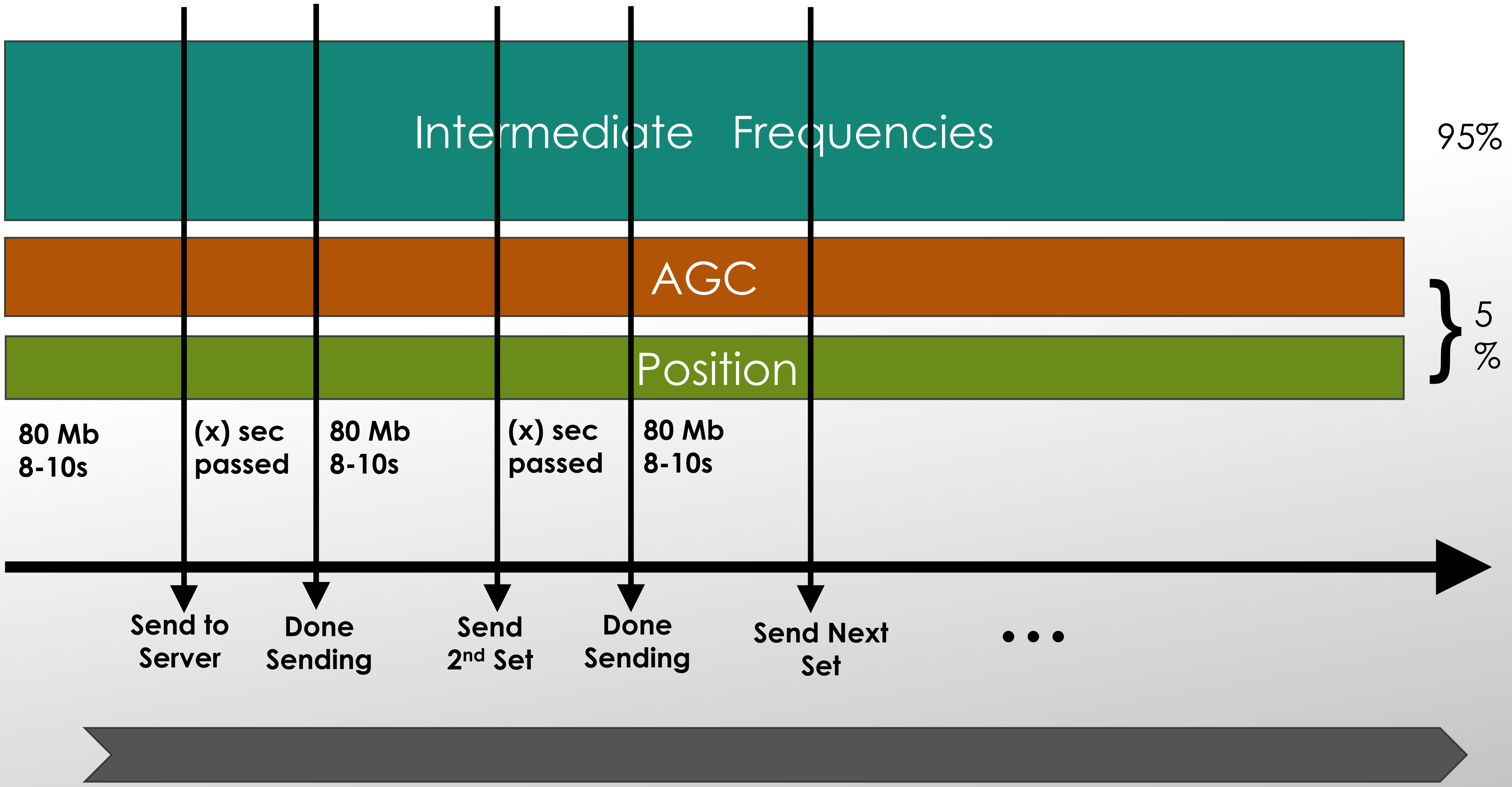
GROUND STATION



Machine Running Linux

- 2 GB RAM and 50 GB Storage
- Capable of running MATLAB
- Connected through a publicly accessible IP address
- Listening on an open port above 1024

DATA TRANSFER



LOCALIZATION ALGORITHM

Purpose: Fulfill requirement for localizing RFI source within 40m

Initial Design → Programming → Unit Test: Simulated Data → Unit Test: Real Data From Customer

Proof of Concept: Completed

Next Steps: → Field Test: RAMROD Data → Requirement Fulfilled

LOCALIZATION: FUTURE WORK

Unit Tests: Completed
❑ **Proof of concept:** PDOA with Least Squares method
❑ **Accuracy:** Localization possible within 40m with 100 samples

Field Testing: TO-DO	Estimated Completion & Level of Difficulty
❑ **Compatibility Test:** Ensure algorithm operates with RAMROD data	❑ **March 17**: low priority
❑ **Accuracy:** Prove that the source can be localized within 40m with RAMROD data	❑ **March 30:** low priority

TEST PLAN

Ground Level

Test	Date (2018)
❑ Talon Hardware Calibration	February 9-12
❑ Payload Functional Ground and Downlink Testing	February 17-24
❑ GPS Denied Algorithm Ground Testing	February 20-24
❑ System Integration	February 9-26

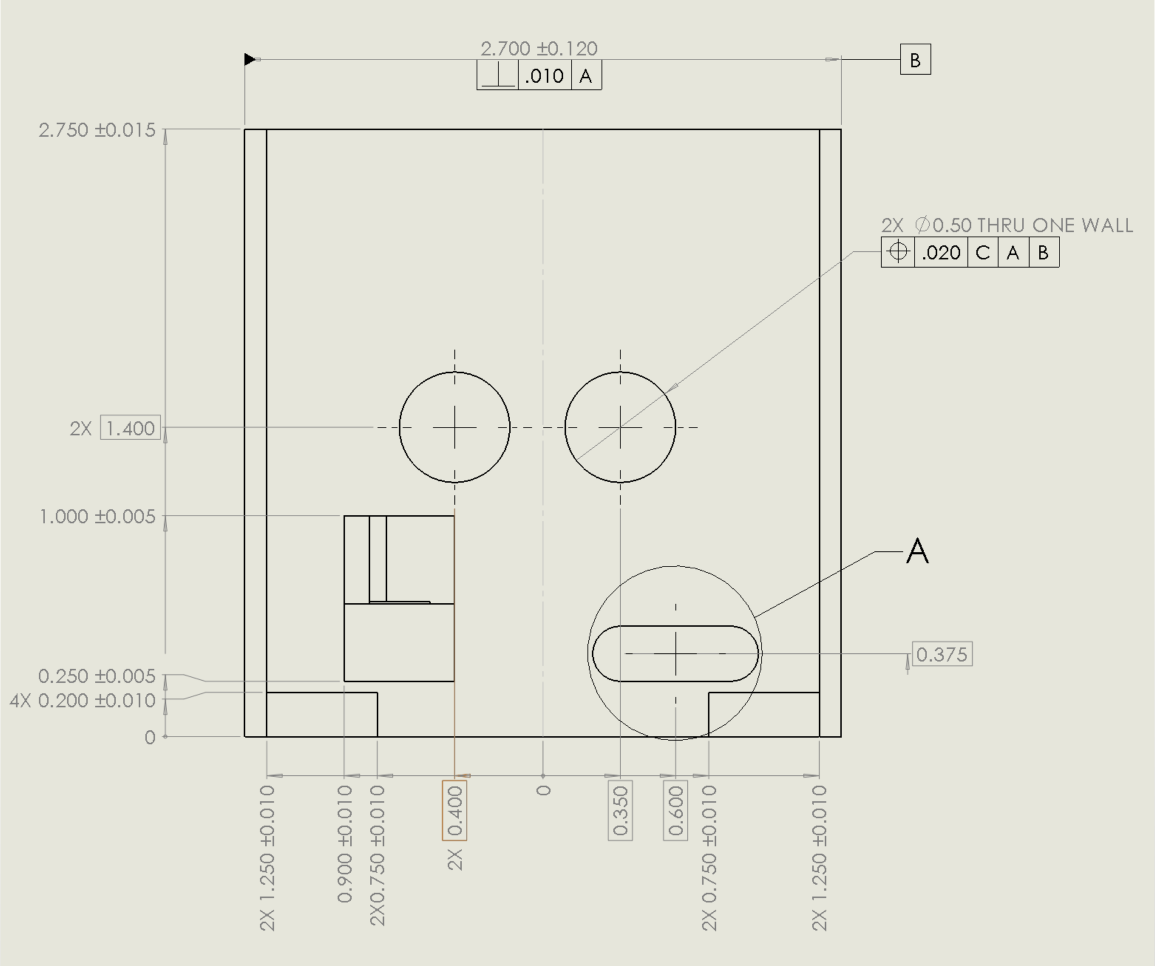
Preliminary Flight Test

Test	Date (2018)
❑ Talon EKF Calibration and Functionality Test	February 12-28
❑ GPS Guided Flight Test, Endurance test	February 24-28
❑ INS Guided Flight (Straight and Level)	March 1-17
❑ Localization & Power Profile	March 1-17

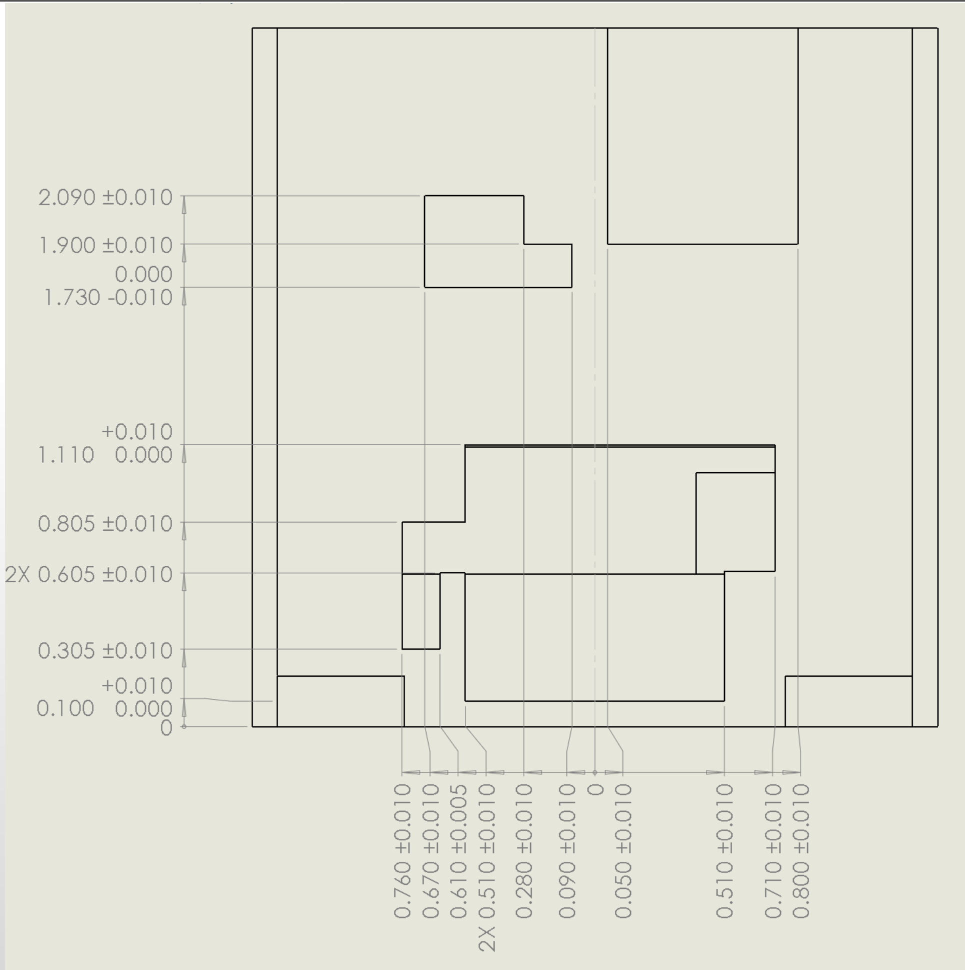
Mission Level Test

Test	Date (2018)
❑ Maneuvers in GPS Denied Environment	March 19-30
❑ Mission Level Flight Test and Localization	March 19-30
❑ Redundant Tests	March 20-30

PAYLOAD STRUCTURE BOX DIMENSIONS



Left View



Right View

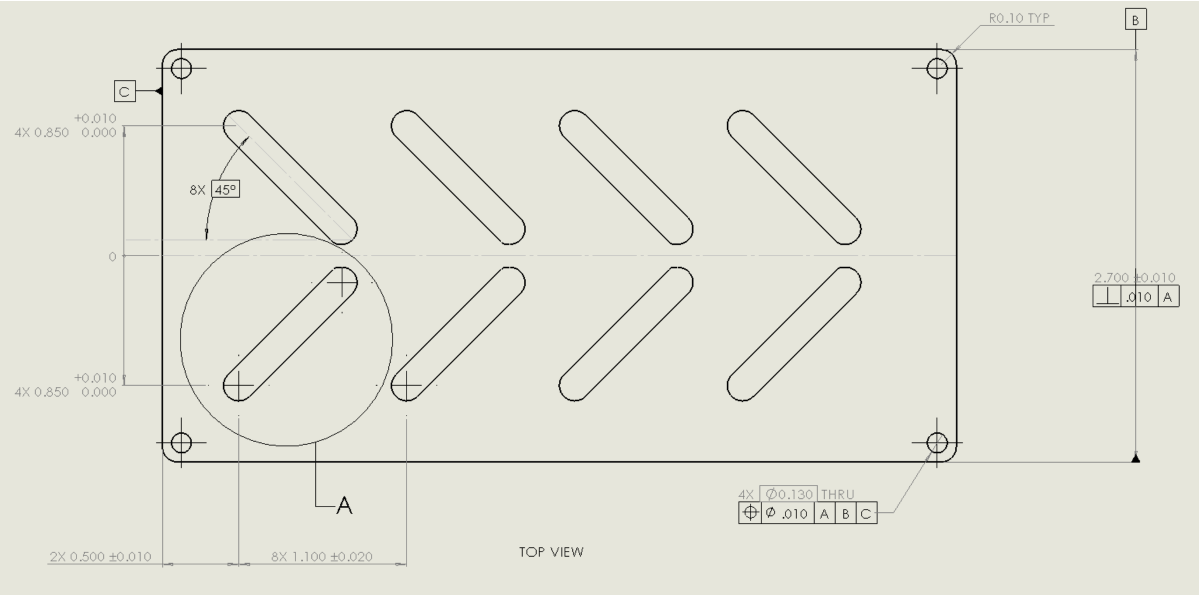
Back

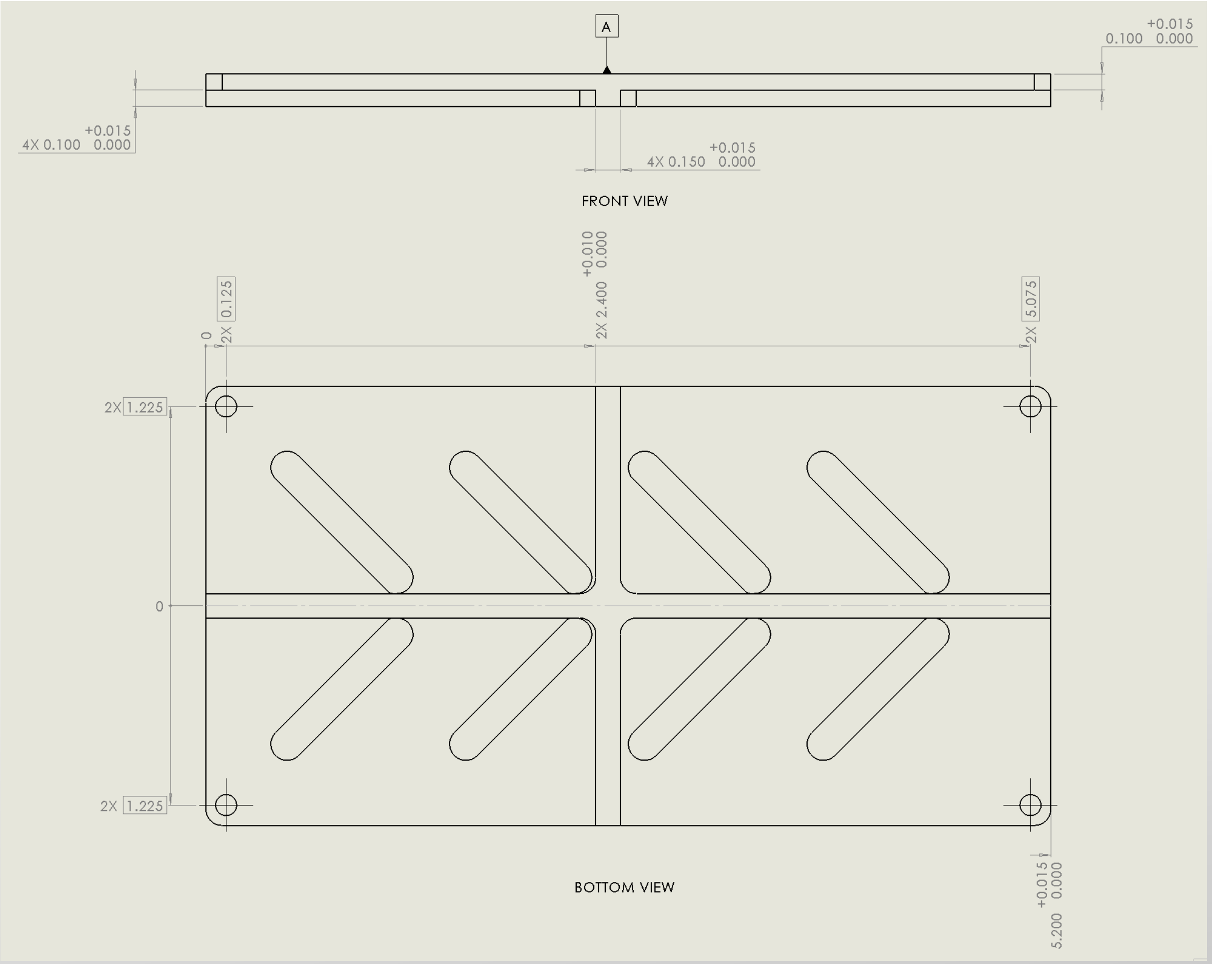
PAYLOAD STRUCTURE BOX DIMENSIONS

Front View

Back

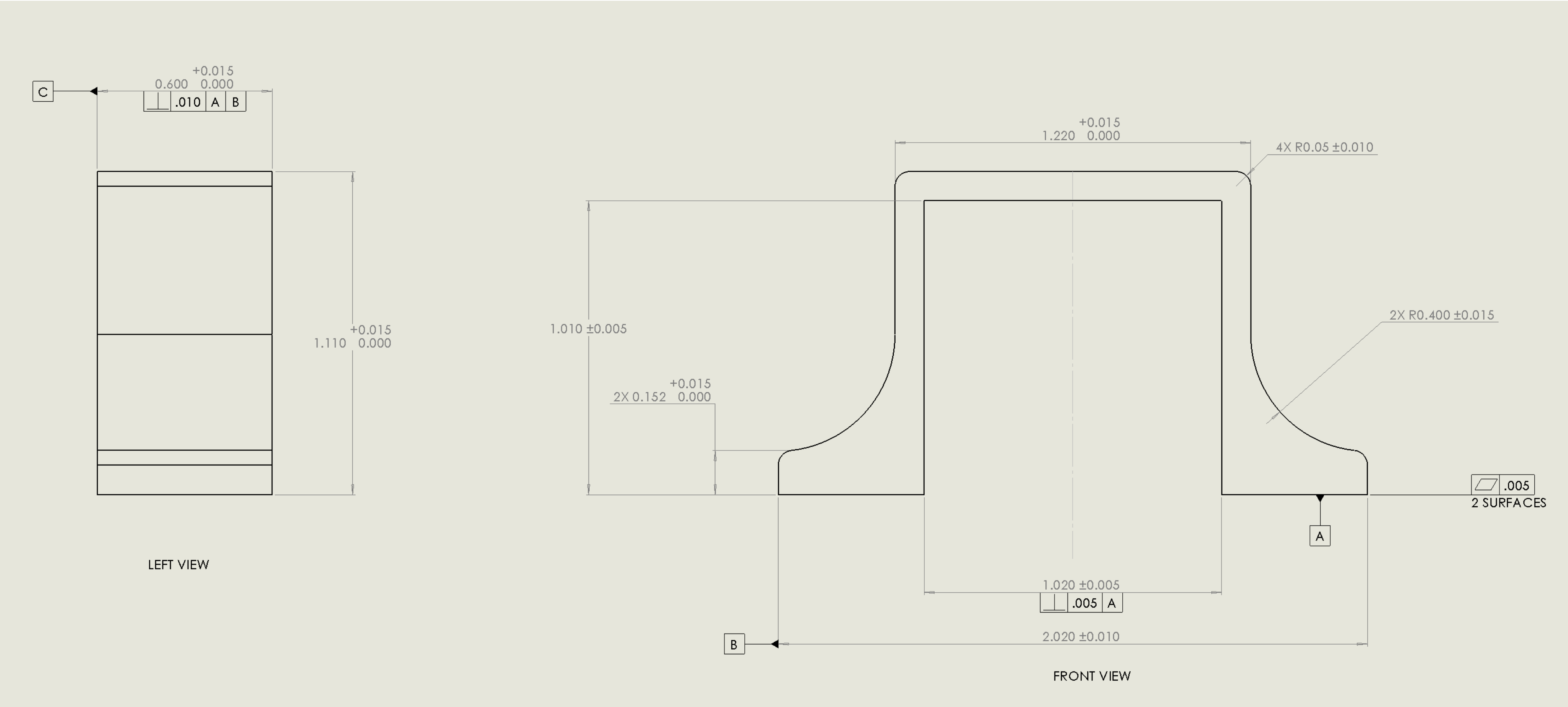
PAYLOAD STRUCTURE TOP DIMENSIONS





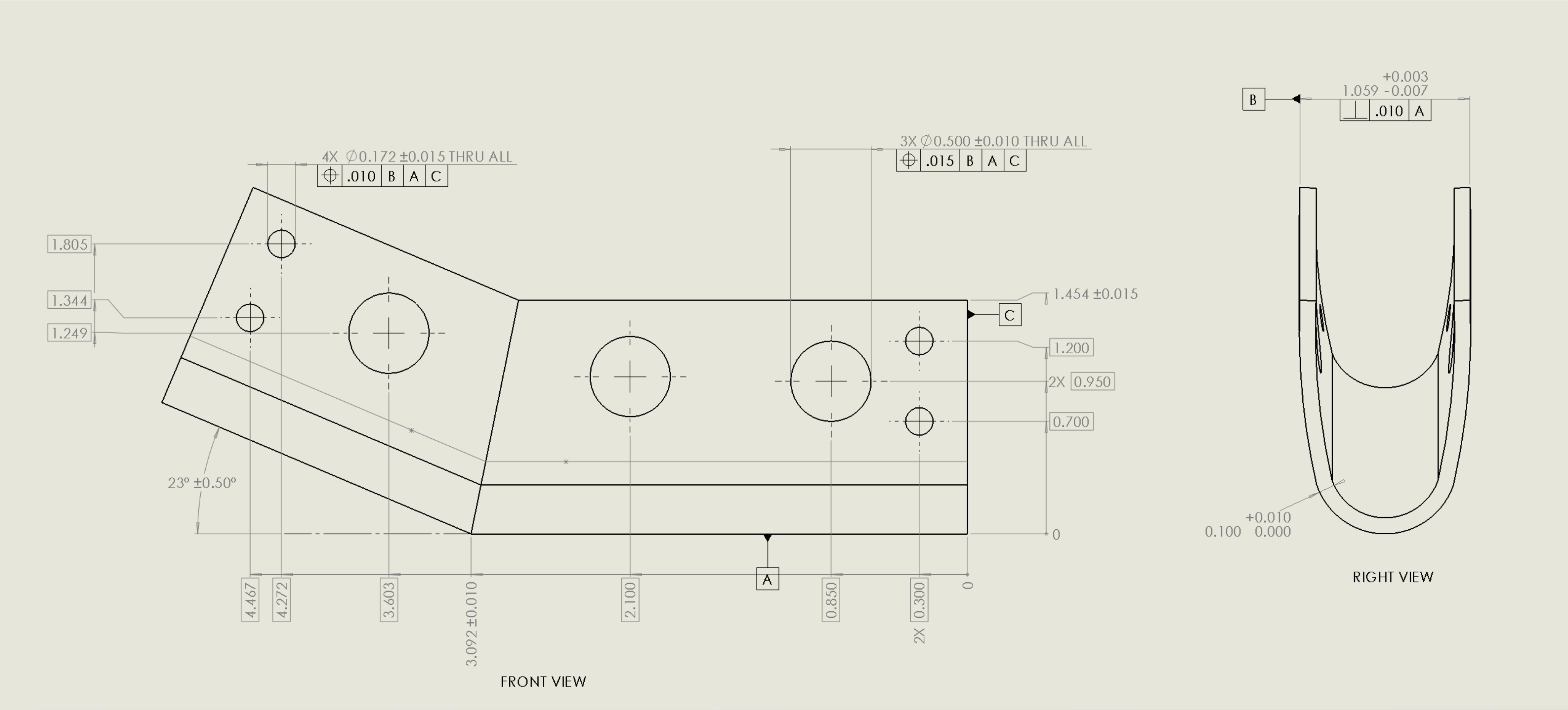
Back

BATTERY BRACKET DIMENSIONS



Back

TAIL SKID DIMENSIONS



Back

UAS LAUNCH METHOD

Bungee Launcher

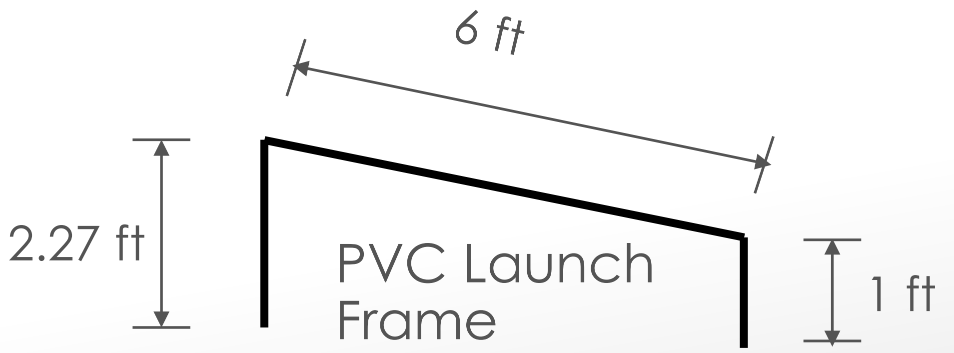
Elastic cord used to launch aircraft

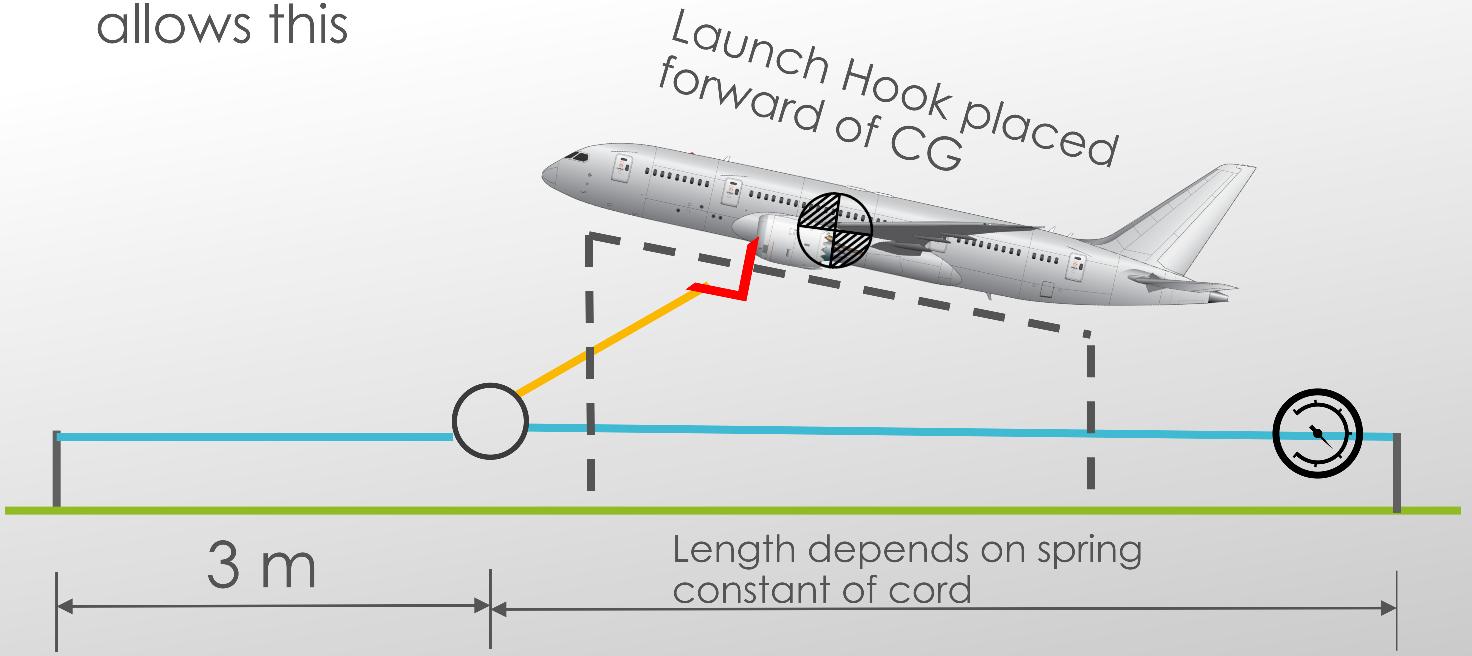
- cord junction
- launch frame
- elastic cord
- inelastic para-cord
- ground
- ground stakes
- force gauge
- launch hook

Aircraft must leave launch frame at V_{stall}

↓

Cord stretched to tension force that allows this





PROCUREMENT BREAKDOWN 1/2

Item	Price Per Unit ($)	Quantity	Total Expense ($)
NylonX	65	2	130
60-Amp ESC	85	1	85
Power 25 Motor	69.99	1	69.99
Pixhawk 2.1	198	1	198
Here GNSS Antenna	48	1	48
Garmin Lidar Lite	119.67	1	119.67
2TP7000 Batteries	105	2	216.37
Wing Servos	33.99	2	61.38
Tail Servos	15	2	35.5
Pitot Tube	13.99	1	13.99
Airspeed Sensor	57.22	1	57.22
IMU Conversion Module	10.5	1	10.5
Indicator PCB	20.2	1	20.20
Spacers and Foam	4.59	1	4.59

PROCUREMENT BREAKDOWN 2/2

Item	Cost per Unit ($)	Quantity	Total Expense ($)
Talon Airframe	114.67	1	114.67
TP820HVC charger	140.38	1	140.38
PPM Encoder	37.86	1	37.86
DMU11	397.43	1	397.43
AR7700 Receiver	59.99	1	59.99
Dx6 Transmitter	199.99	1	199.99
APC 10x7 Prop	3.19	1	3.191
Px4Flow	124.18	1	124.18
Radio Telemetry Kit	36.99	1	36.99
Mobius Hd Action Cam	79.95	1	79.95
Wet Noodle	27.5	2	55
General Hardware			151.66
Total Expense:			**$2461**