Manufacturing Status Review

Lockheed Martin LLAMAS Team

sateLLite ADCS fault MAnagement System

Dalton Anderson
Daniel Greer
Ben Hutchinson
Kent Lee
Andrew Levandoski
Andrew Mezich
Samuel O'Donnell
Zach Reynolds
Kristyn Sample
Corwin Sheahan
Pol Sieira
Zack Toelkes

Project Overview

Mission Objective

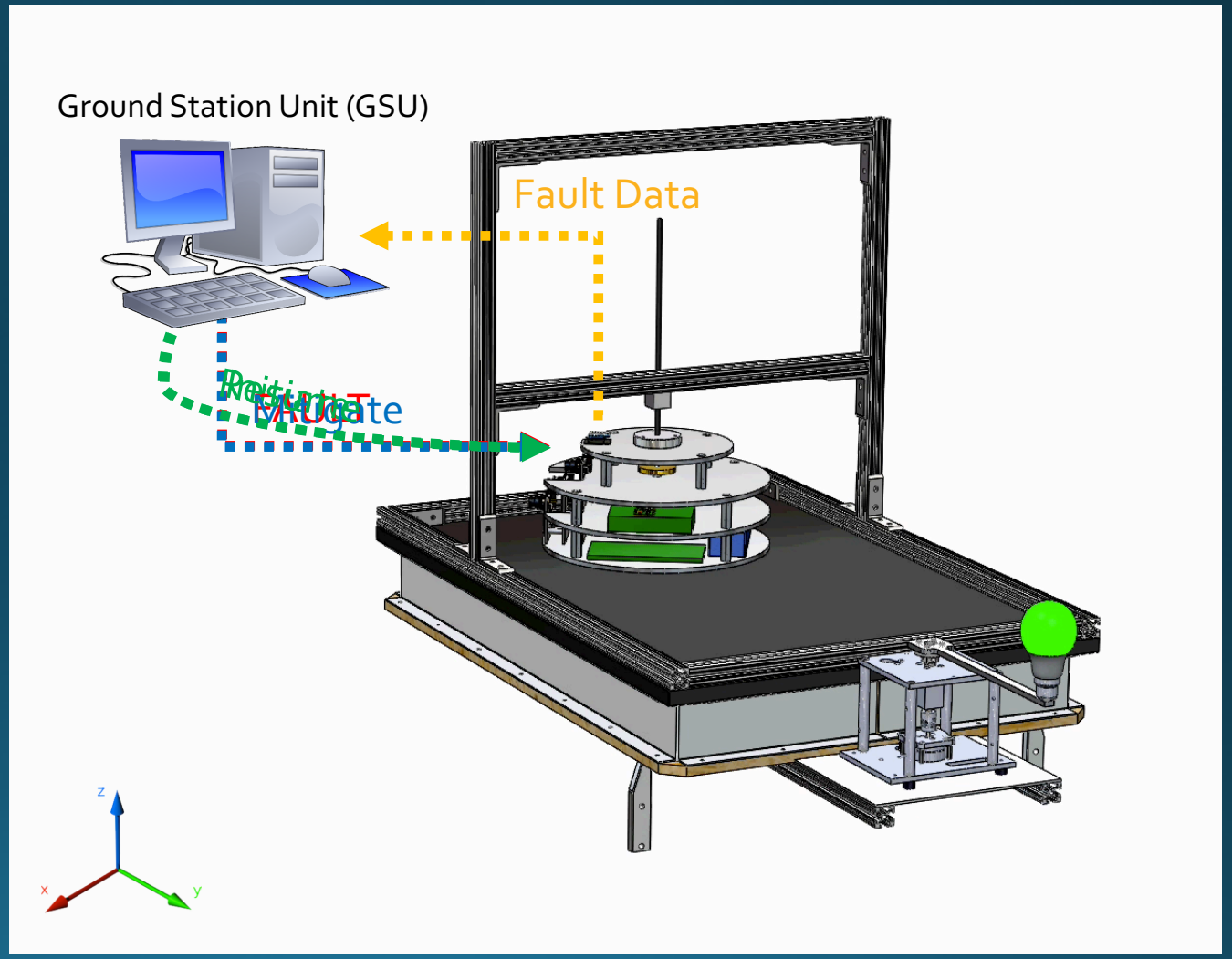
- The team will develop a fault management test bed which allows for testing of fault management software by fault injection into the attitude determination and control system (ADCS) of a mock-satellite (MockSat).
- The MockSat will be representative of the GOES-16 satellite, capable of relaying telemetry and fault data to a ground station unit, allow user selection of faults, and will be tested on a reduced-friction TestTable.

Levels of Success

	Level 1	Level 2	Level 3
TestTable	Construct a TestTable to allow for 2D translation dynamics with passive control, 1D rotation dynamics, support weight of MockSat, stationary attitude reference		Moving attitude reference
MockSat Hardware	Power source, position sensor, coarse orientation sensor, fine orientation sensor, redundant reaction wheels, ADCS/fault injection processor, data storage, 15 minute constant operating time	30 minute constant operating time	60 minute constant operating time
Fault Injection	Inject fatal operating fault into primary reaction wheel after pre-determined time from testing start	Inject fatal operating fault into coarse sensor	
Fault Management	Upon fault injection, the MockSat will recognize the presence of the fault and enter a safe mode		Upon user command, MockSat responds in a way that maintains operational integrity
MockSat Control	Active planar rotational control with passive translational control		
Comm/Data Handling	Flight software and fault uploaded prior to testing, telemetry data stored on-board MockSat, Ground Station data analysis post-test	Wired, real-time telemetry and fault injection	Wireless, real-time telemetry and fault injection

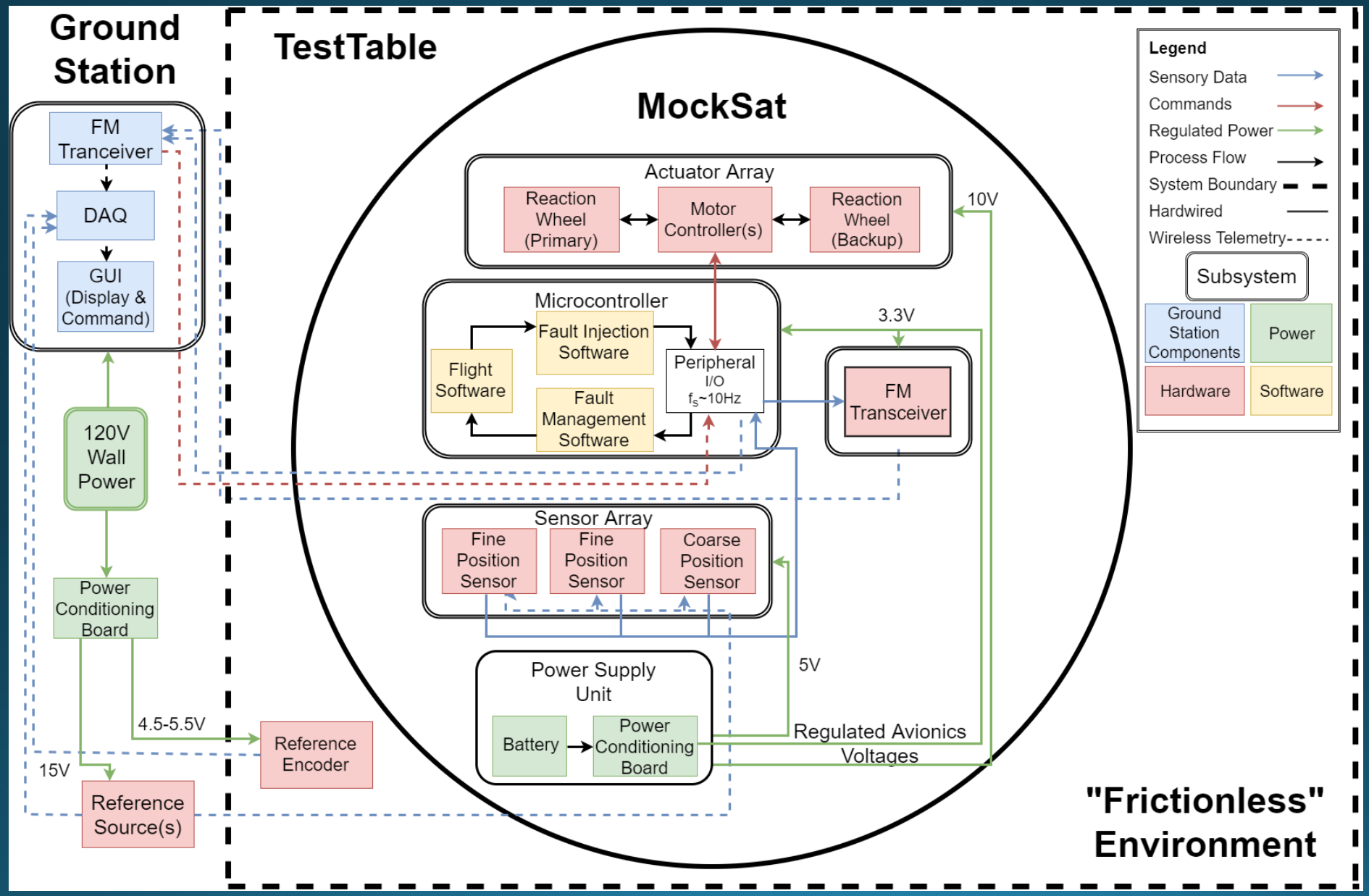
Concept of Operations (CONOPs)

- Test Initiation
  - MockSat initializes and begins searching for target.
- Nominal Operation
  - MockSat has acquired target and tracks motion to within ±2.5°.
- Faulted
  - Fault Injection has introduced a fault that inhibits the MockSat from tracking the target.
- Management of Fault
  - Fault Management has detected and identified the fault and relayed that information to the Ground Station Unit.
  - MockSat is in a faulted state and not maintaining any attitude.
- Initiation of Recovery Sequence
  - MockSat has regained attitude control and is awaiting command to resume searching.
- Recovering
  - MockSat has received command to resume searching for target.
- Return to Nominal Operation
  - Target has re-acquired the target and is tracking to within ±2.5°.



CONOPS (10x speed)

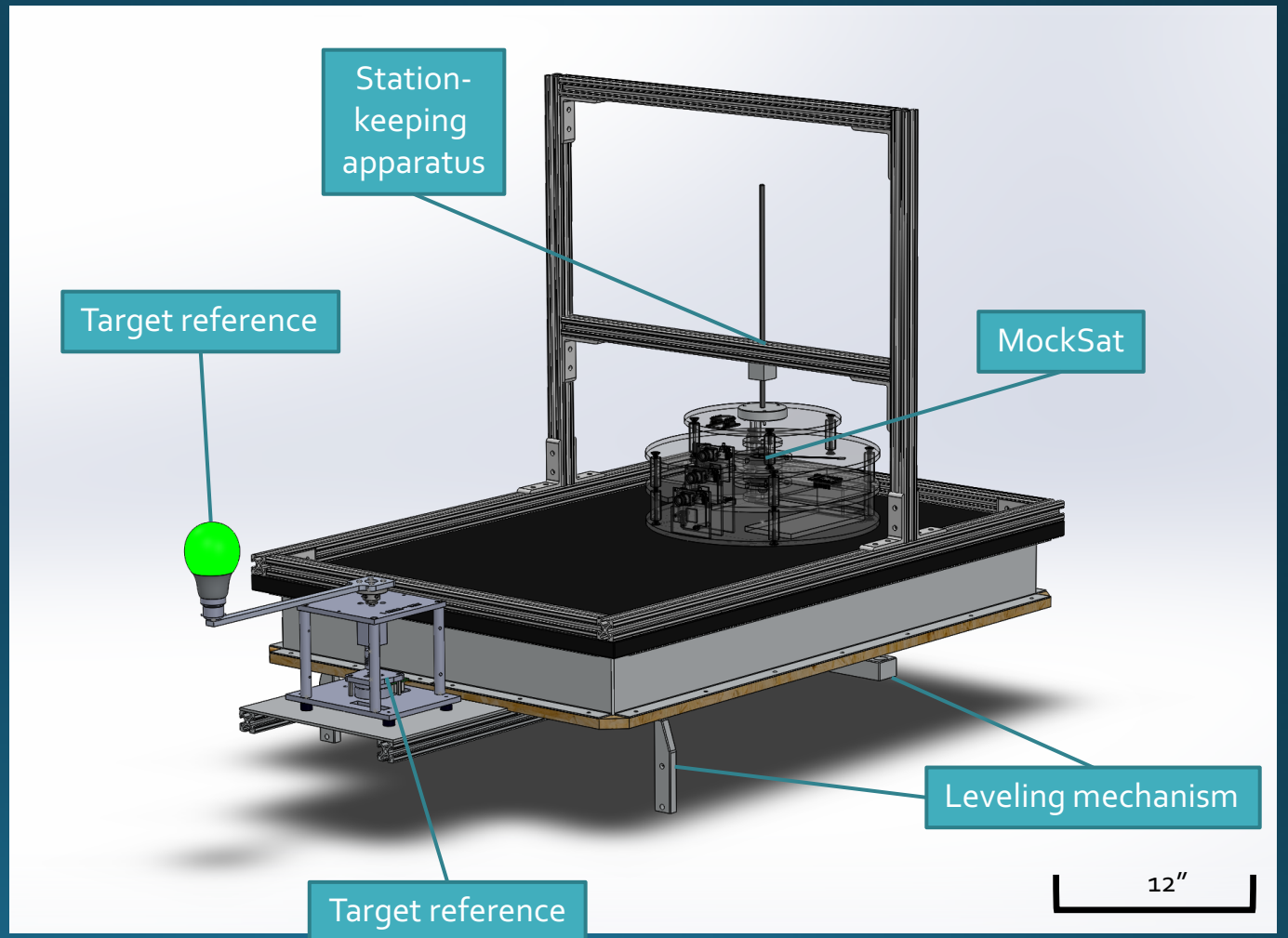
Functional Block Diagram (FBD)



Baseline Design – TestTable

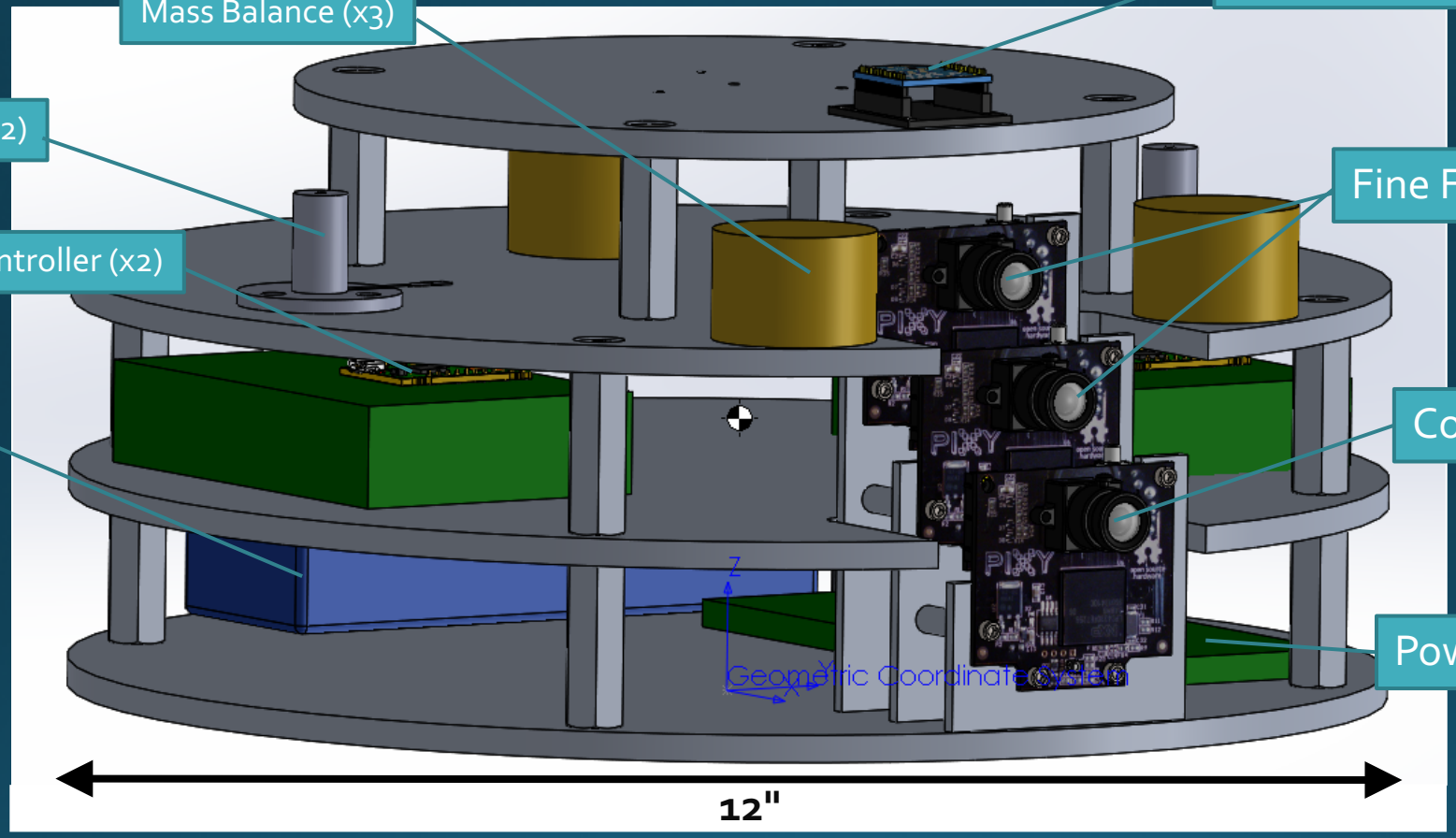
The majority of the TestTable is heritage equipment from the *TracSAT* senior project, with the following modifications:

- Half of the TestTable surface has been taped-over in order to increase the lifting capacity of the TestTable (~9lb ⟶ ~24lb).
- A table leveling mechanism has been added.
- Station-keeping is accomplished via a removable bearing-block and rigid shaft apparatus.
- The target reference provides an object for the MockSat to track optically.



Testing suite

Baseline Design – MockSat



MockSat Concept

Total Weight	12.9 *lbm*
MOI about Axis of Rotation	224.5 $lb*in^2$
Height	5.9 *in*
Width	12 *in*

Critical Project Elements - Motors

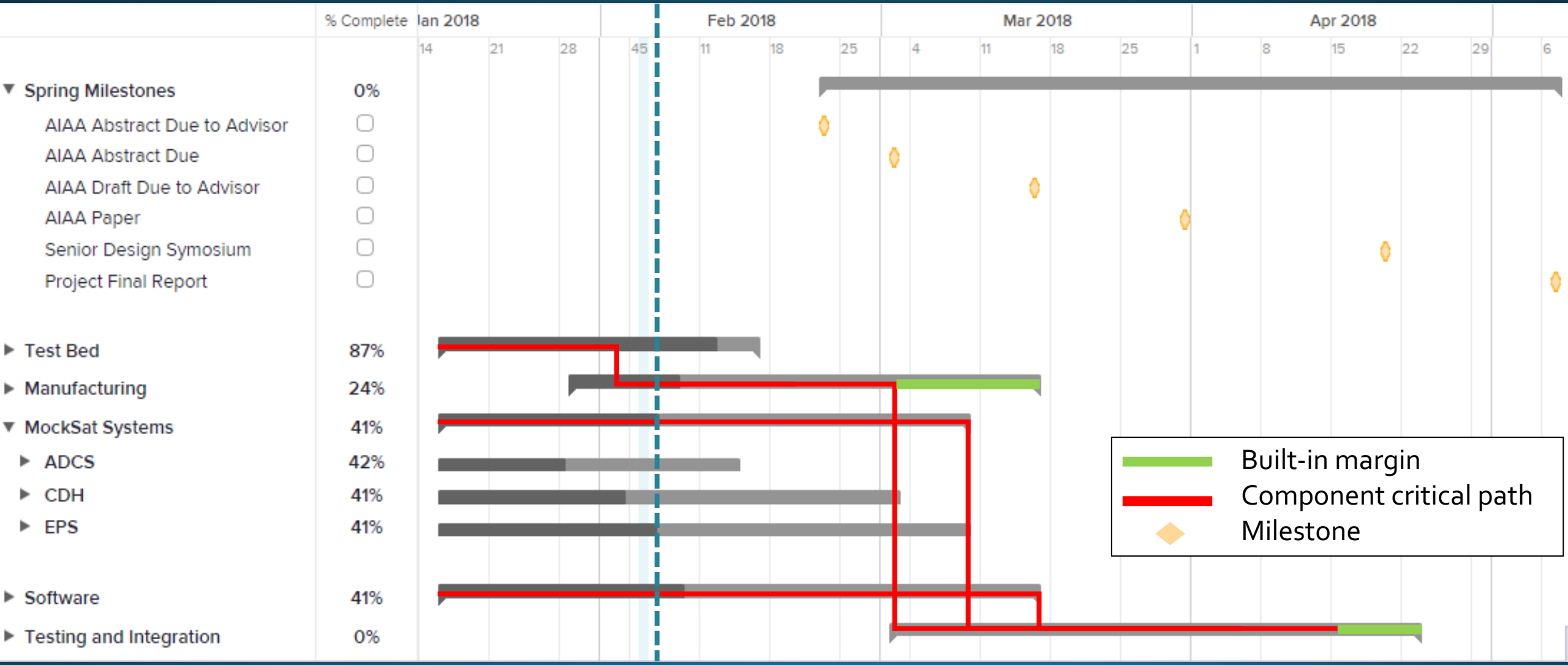
- Proper motor selection will enable team to meet control system bandwidth requirements.
- Careful and accurate characterization of motor performance and internal friction of the motors is absolutely essential.
- Internal friction data is needed for both the control system and the ability to introduce reaction wheel faults.

Critical Project Elements - Fault Injection (FI) and Management

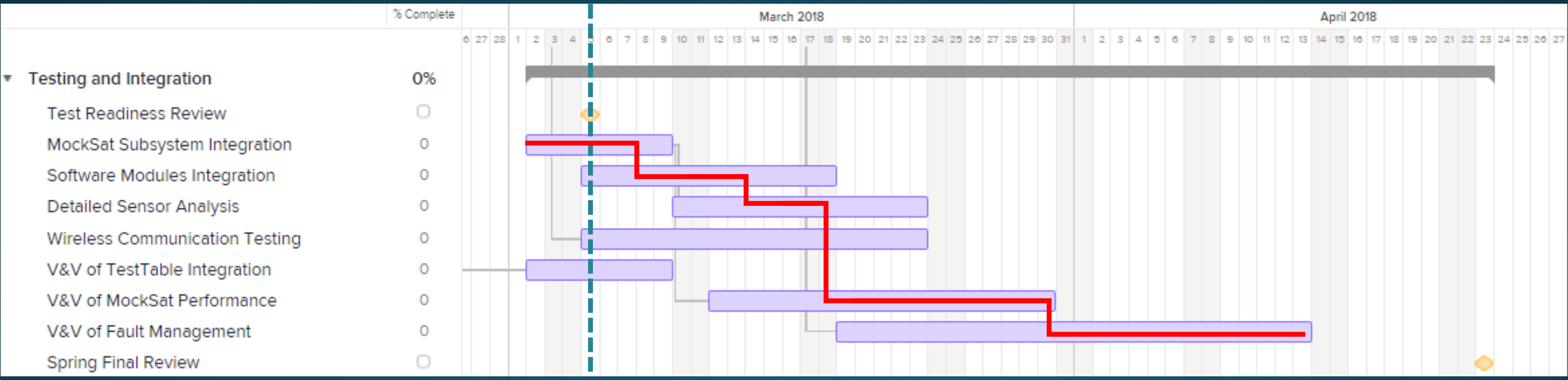
- Ultimate goal of the project revolves around fault management (FM).
- The ability to introduce fault requires extremely accurate characterization of the entire MockSat system.
- Difficult to test and evaluate FI & FM until the various subsystems have been characterized and fully integrated into the MockSat system.

Schedule

Schedule Overview



Schedule – Testing and Integration



Manufacturing

Mechanical

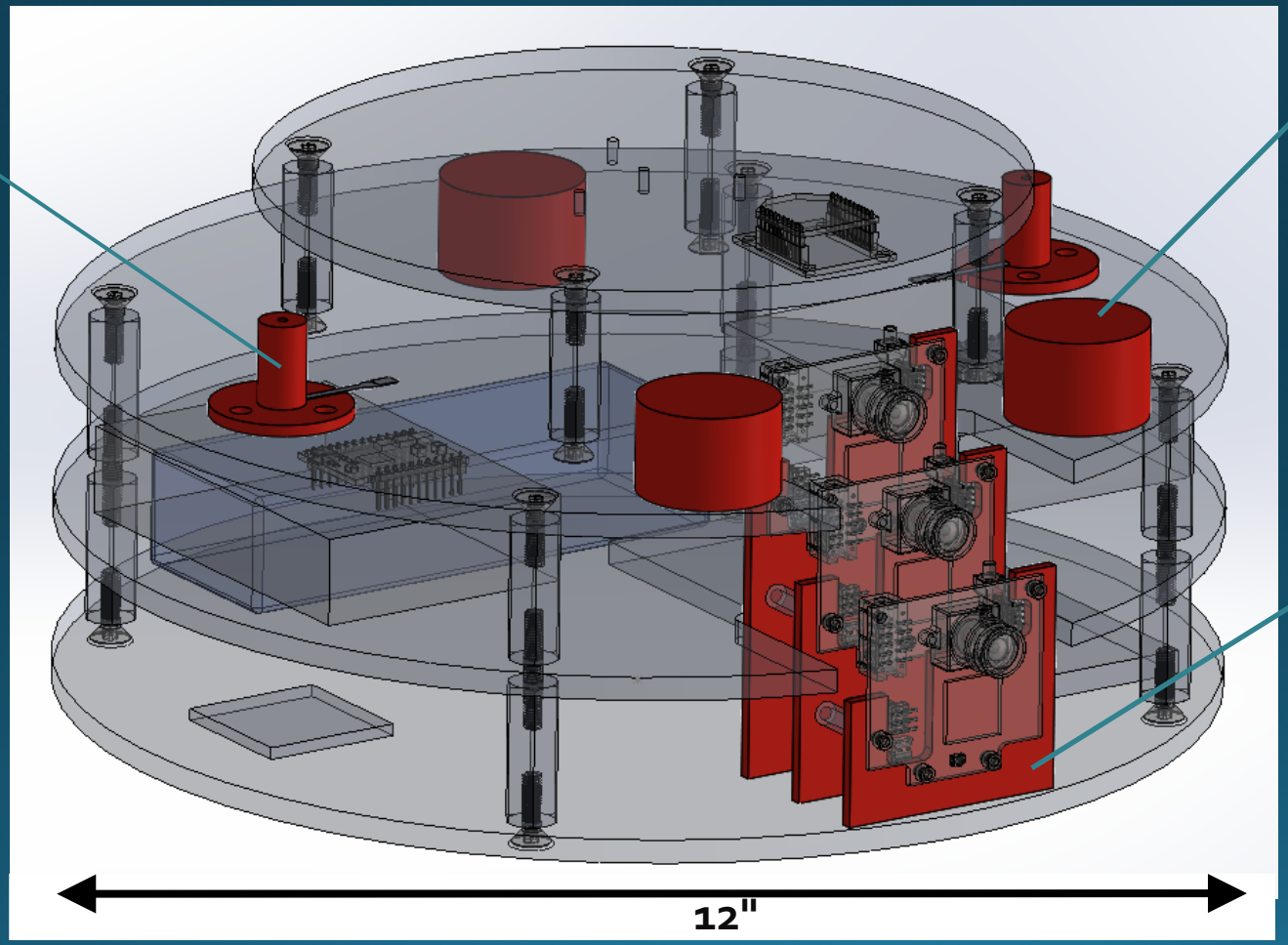
Scope: What is involved?	Specific components	Status
MockSat		
Structure	plates, standoffs, hardware	100%
Internals	motor mounts, PIXY assembly, misc. brackets/tie-downs/etc.	15%
TestTable		
Leveling mechanism	ball joint mounting assembly, jack screw blocks	100%
Station-keeping apparatus	80-20 gantry, bearing block, rod, MockSat interface	100%
Target actuator	spin module housing, stepper motor	33%
Reference target	LED bulb, battery, rigid arm	25%

Mechanical – Remaining items

Motor Mounts (x2)

Misc. mounting items:
- brackets
- PCB stand-offs
- tie-downs
- cable management

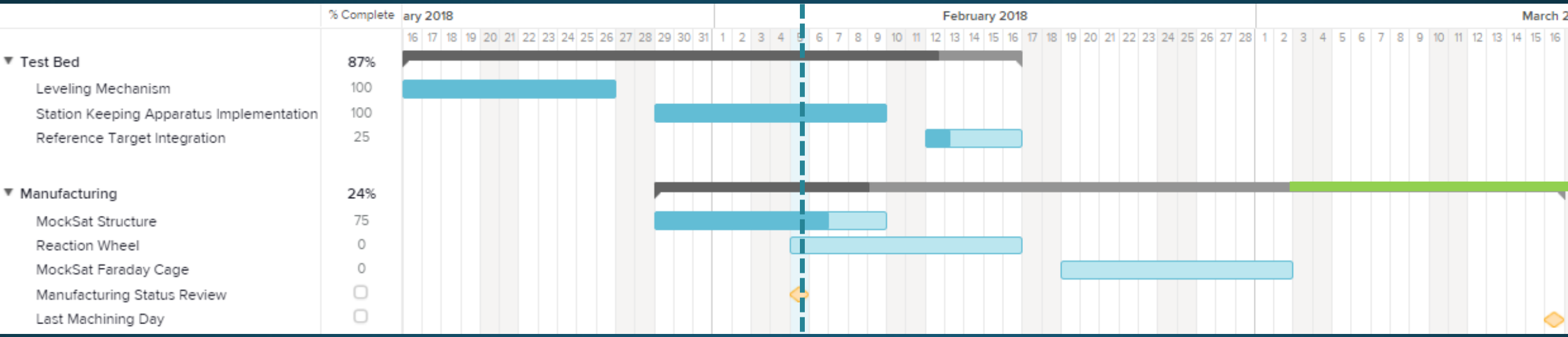
Faraday cage for interference-sensitive components



Mass Balance Weights (x3)

Pixy Mounting Assembly

Schedule – Manufacturing

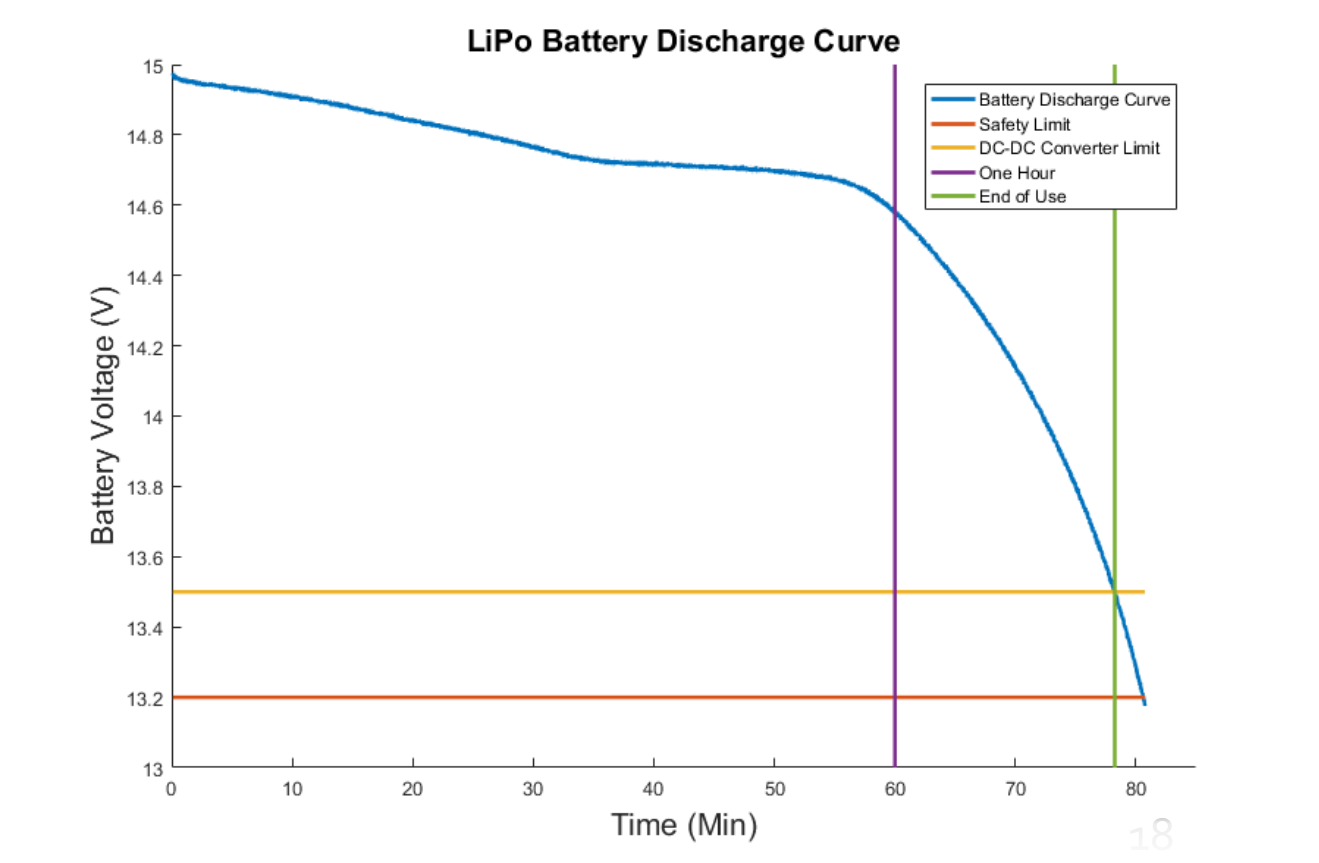


EPS

- What is involved?
  - Battery, power conditioning board, connection board
- What has been completed?
  - Testing battery
  - Initial power conditioning schematics and PCB design

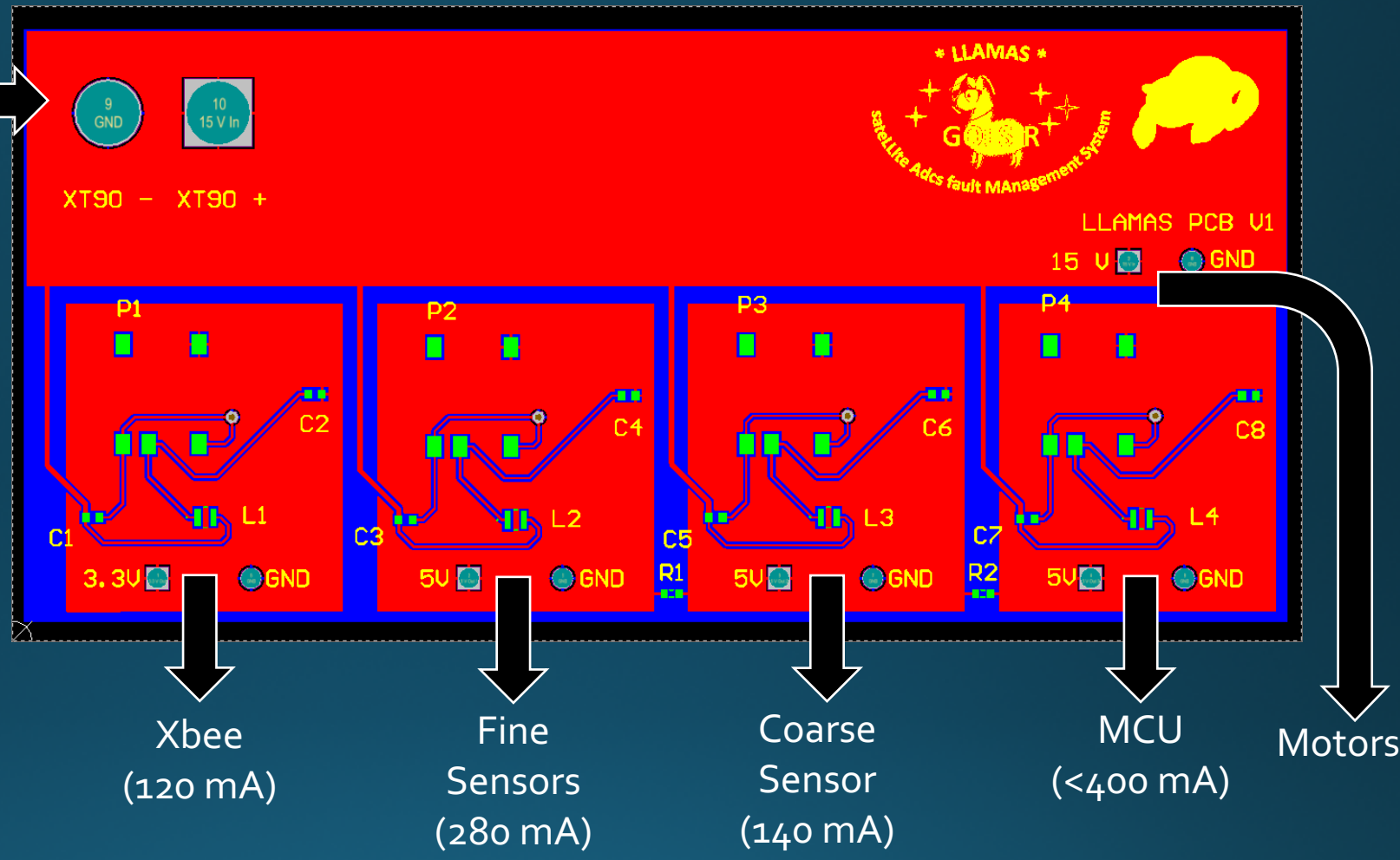
Test:

- LiPo battery: 5000 mAh 25C 4s
- 15 Ohm power resistor
- ~1 A discharge (~0.75 A expected in system)
- Level 3 Success verified: battery can power system for over 1 hour



EPS

- What is involved?
  - Battery, power conditioning board, connection board
- What has been completed?
  - Testing battery
  - Initial power conditioning schematics and PCB design



- PCB connects to XT-90 battery connector
- DC-DC Converters have current output limits of 606 mA for 3.3 V and 400 mA for 5 V
  - 3.3 V plane and 5 V planes have margin below current limit

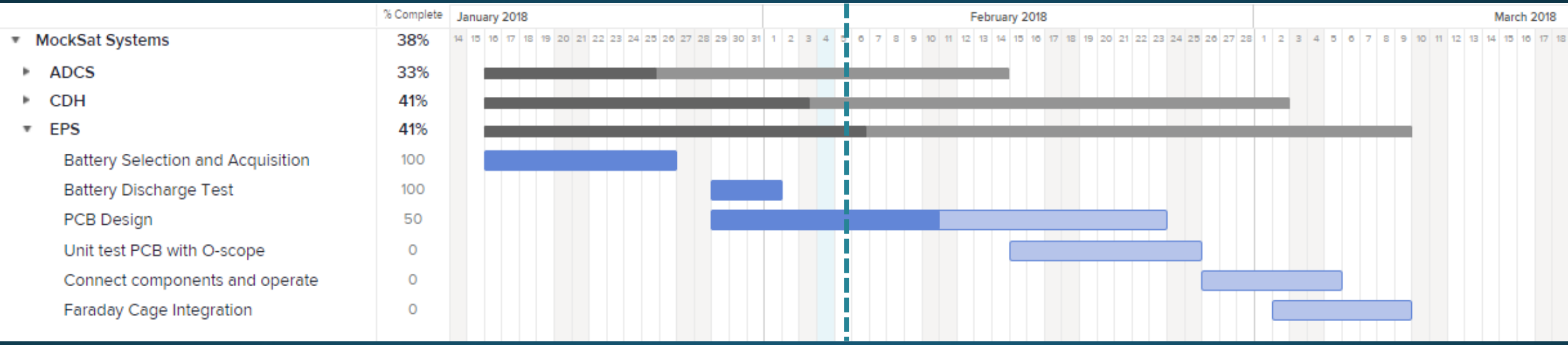
EPS

- What is left to do?
  - Receive PCB and test/characterize it
    - Solder on all components and connect to battery
    - Use multimeter/oscilloscope to verify electrical characteristics: output voltage, noise
- System integration
  - Connect battery to PCB and other MockSat components to verify component functionality
    - Verify that Pixy Cameras, MCU, and Xbee work correctly when powered by battery

EPS

- Important for project success:
  - Testing time of 1 hour. Achievable due to results of battery test
  - Regulated power for every component
- Difficult and time consuming aspects
  - Populating and characterizing PCB
    - Possible redesign of component placement or trace width
  - Full system integration, verifying that each component works

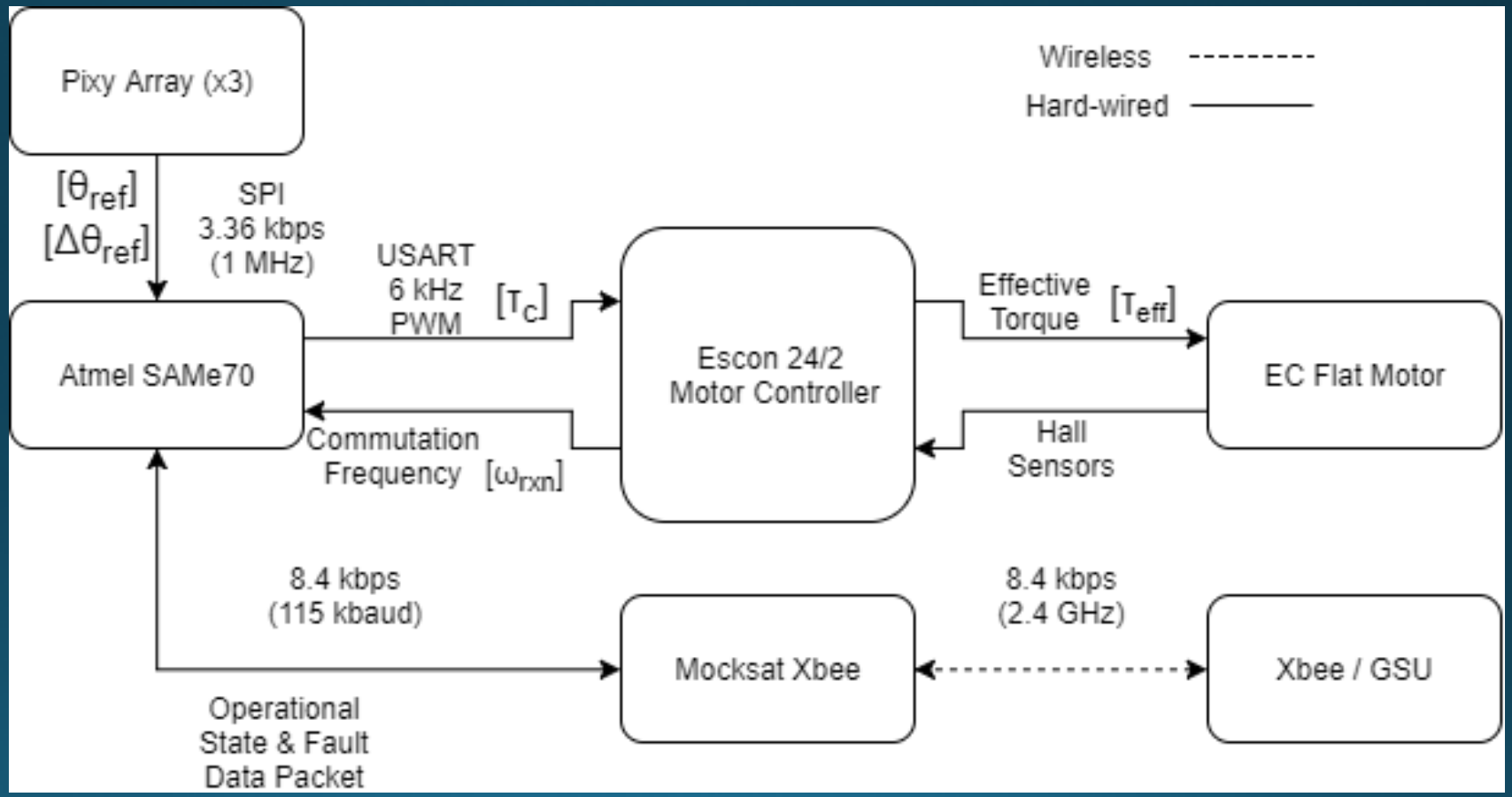
Schedule – EPS



CDH- Overview

- 12 MHz Internal Oscillator
- PLL Multiplier of 16
- 192 MHz System Clock
- Digital I/O
- Single Interrupt Priority
- 100 ms S/W loop limit
- 16-bit PWM
- 3 SPI chip selects

- Two 24 Ch. 12-bit ADC/DAC (backup)



CDH – Visual Sensors

- 3 pixy cameras, 1 object, 14 bytes, 8 bits, 10 fps = 3360 bits/sec
- 1 MHz default programmable peripheral clock (Atmel SAMe70)
- 3360 bits / 1 MHz --> 3.36 msec

- Recycled Code: Atmel SPI & Pixy SPI

- Offramp: Pixy analog output mode 5

CDH- Xbee Wireless Comms

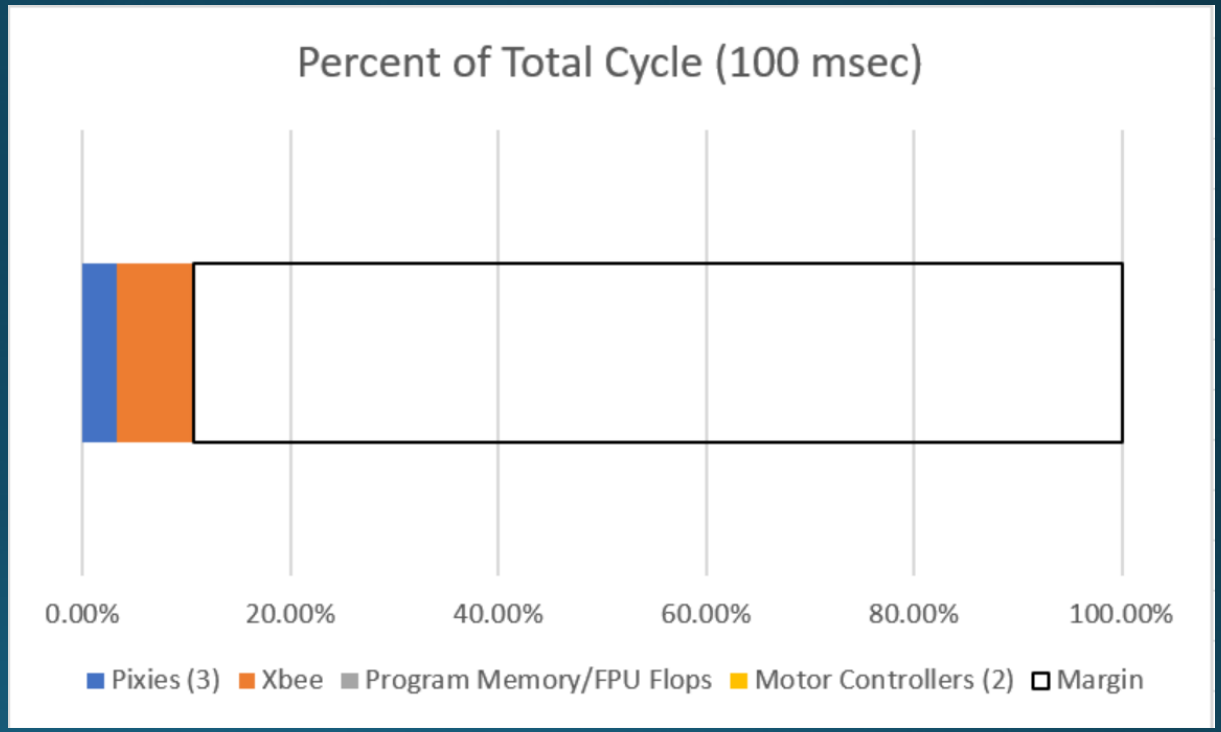
- Data throughput: 8.4 kbps
- Xbee throughput: 250 kbps (maybe)
- Expected baud rate: 115 kbaud
- Programmable Clock 4 of MCU is the baud generating clock.

- 8400/115000 = 7.3 msec

Information	Data Transfer Mechanism	Bandwidth
Satellite Position	Wired Encoder	N/A
Target Position	Wired Encoder	N/A
Begin Test	Wireless from GUI	1 bit (1 time)
Fault Injection Initiation	Wireless from GUI	1 bit (1 time)
Recovery Initiation	Wireless from GUI	1 bit (1 time)
Fault Management Status	Wireless from MCU	2 bits (1 time)
Reaction Wheel Speed	Wireless from MCU	1.6 kbps
Redundant Reaction Wheel Speed	Wireless from MCU	1.6 kbps
Mems Gyro Data	Wireless from MCU	1.6 kbps
Commanded Torque	Wireless from MCU	1.6 kbps
$\Delta\theta$	Wireless from MCU	1.6 kbps
Total		**8.4 kbps**

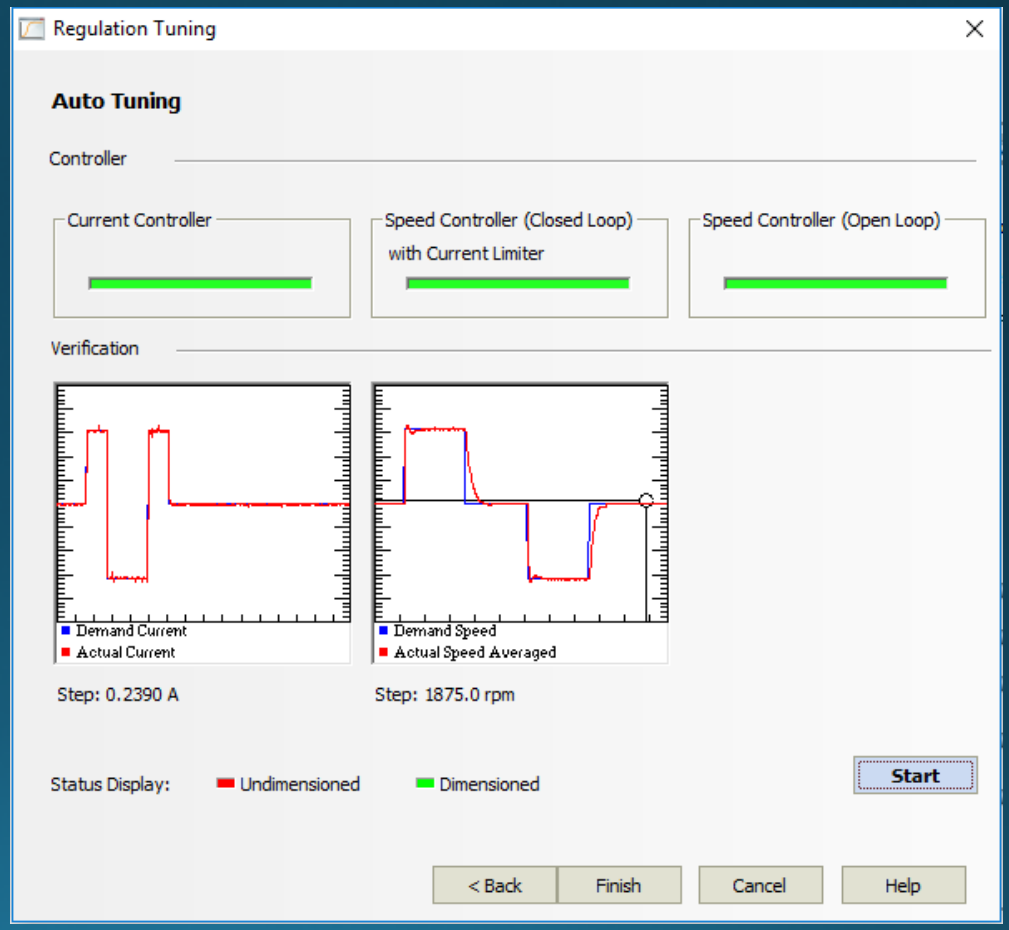
CDH- Timing Diagram

- Xbee TX: 7.3 msec
- Pixy Array: 3.36 msec
- Mainline Code: 6 usec
- Motor Controller: 40 usec

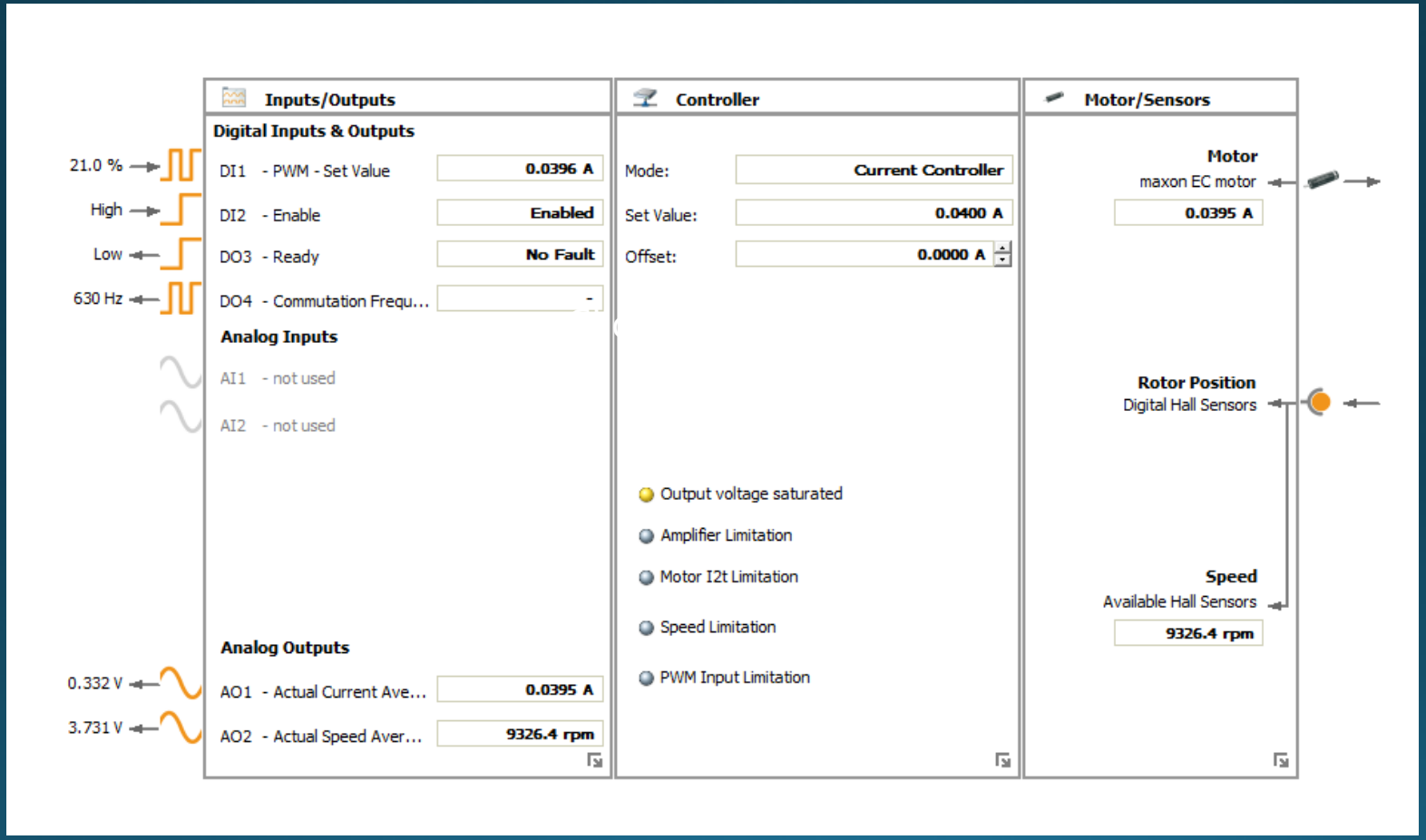


CDH- Actuators

- 6 kHz PWM (10-90% duty cycle)
- Programmable torque and current control via Escon Studio
- Digital I/O:
  - DIO1- PWM
  - DIO2- Enable
  - DO3- Ready
  - DO4- Commutation Frequency
- Analog Out:
  - AO1- Actual Current (Averaged)
  - AO2- Actual Speed (Averaged)



Actuator Testing- Escon Studio



Live data readout with multiple options for data feedback

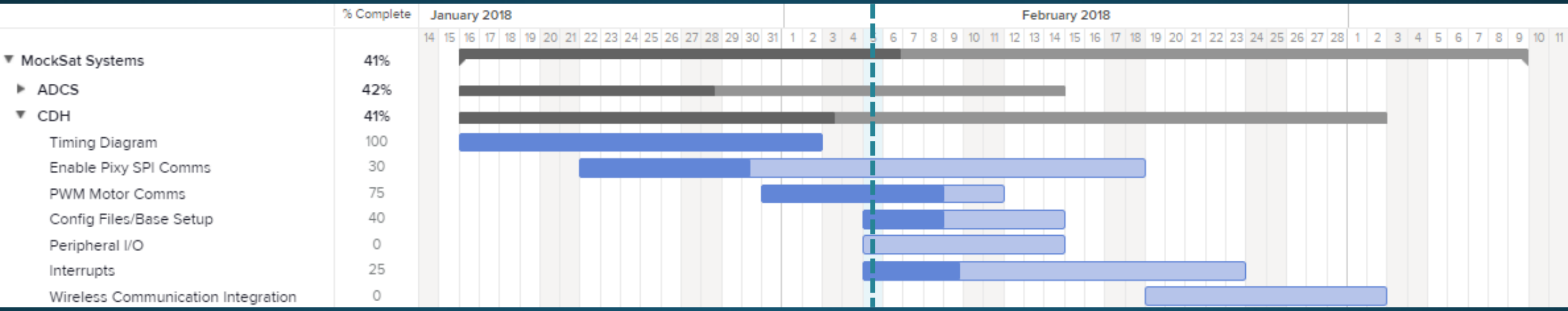
Actuators

- What is involved?
  - Set up motor controller in ESCON environment
  - Test procedure (motor characterization)
  - Tuning control law to desired bandwidth
- What has been completed?
  - Procurement of motors, motor controller and motherboard
  - Motor & controller integrated with ESCON Studio
  - Initial stage of PWM software (Arduino)
- What is left to do?
  - Complete software for MockSat deployment (Atmel SAMe70)
  - Unit tests for bandwidth response and pointing accuracy
  - Integration into MockSat for V & V of ADCS requirements

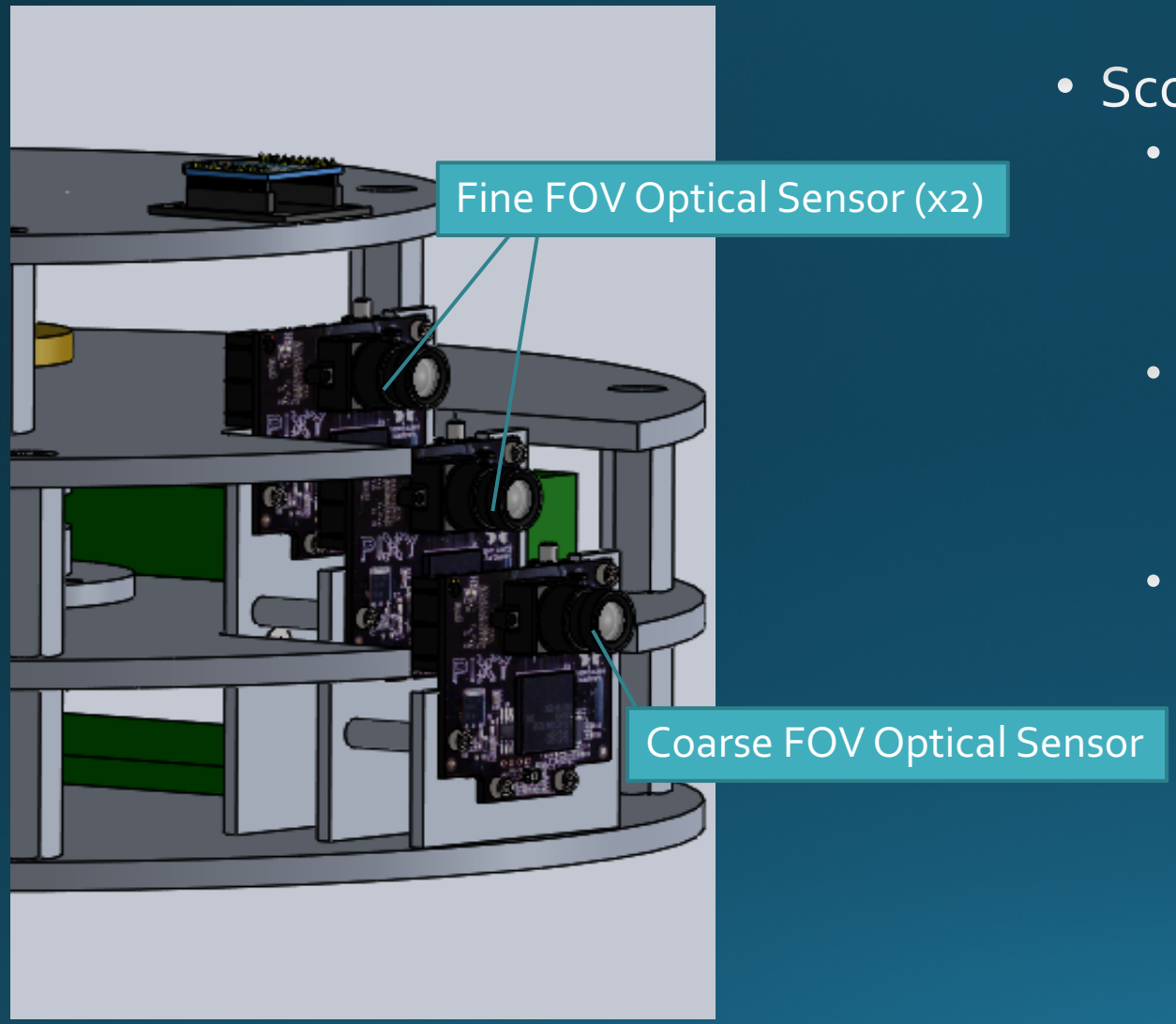
CDH- Software Development

- Atmel provides example code for most of the functionality desired.
  - PWM generation
  - U(S)ART baud generation
  - SPI configuration
  - Interrupts
- Getting these example codes to work together in the same project/solution is an ongoing process.
- Currently using an Arduino for a majority of unit testing that requires deployment onto the Mocksat. (motor characterization)

Schedule – CDH



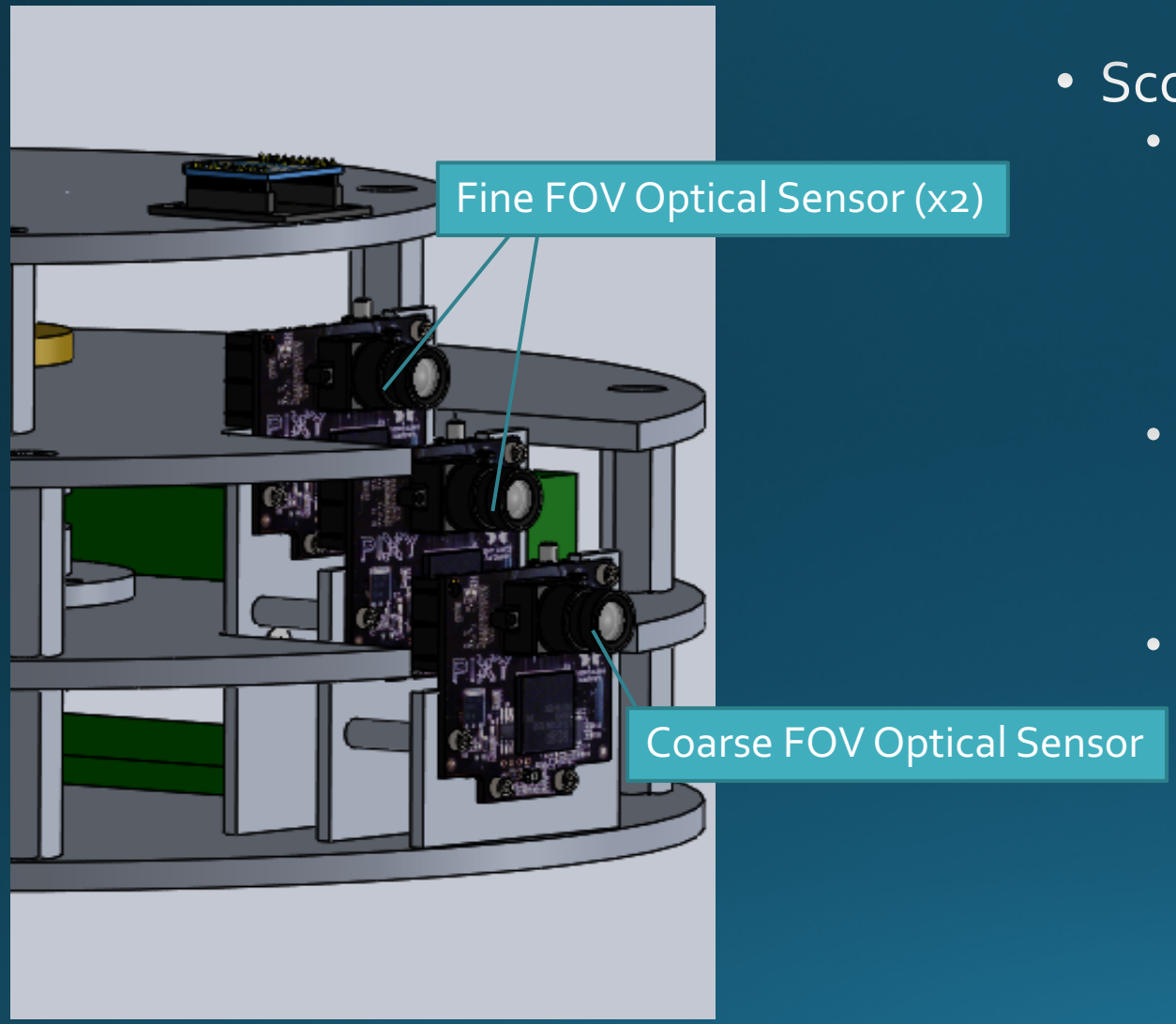
Sensors Overview - Hardware



Sensors Location

- Scope:
  - What is involved?
    - Three Pixy Cameras
    - Two types of lenses
    - Magnetic Hall Effect encoders
  - What has been completed?
    - All hardware has been purchased
    - Testing and Calibration of the Coarse Sensor
  - What is left to do?
    - Select Fine Sensor lens size of two options
    - Identify final target source, which is dependent on Fine Sensor lens
    - Characterize encoder performance

Sensors Overview - Software



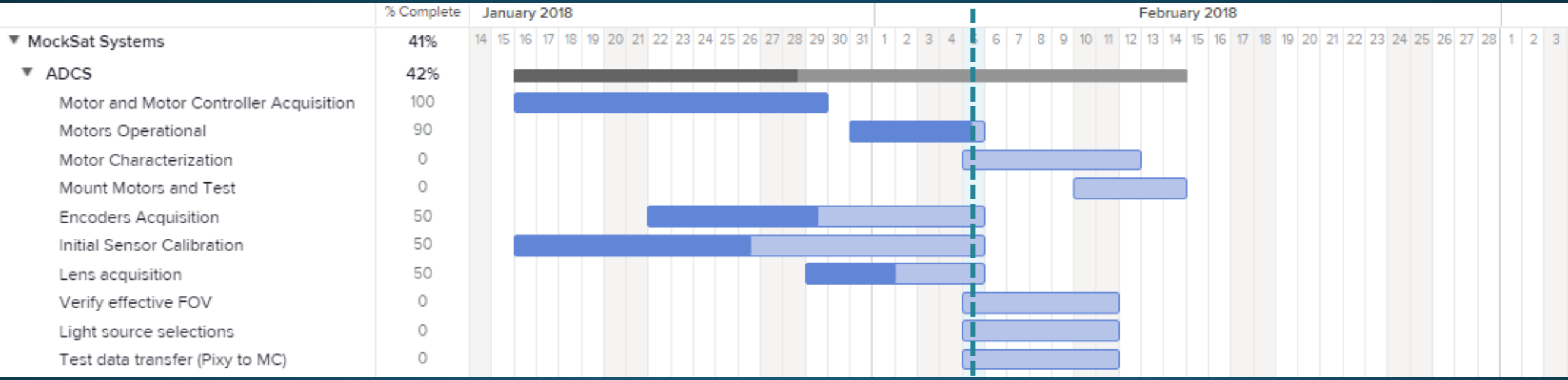
Sensors Location

- Scope:
  - What is involved?
    - Pixy source code and tuning software is provided by manufacture
    - Encoder has a demo-board which will be used as the baseline for the encoder code.
  - What has been completed?
    - A prototype of Pixy outputs controlling a motor has been developed using an Arduino board and a servo motor.
  - What is left to do?
    - Receive encoder to begin development
    - Enable communication between Pixy and SAM E70 Microcontroller

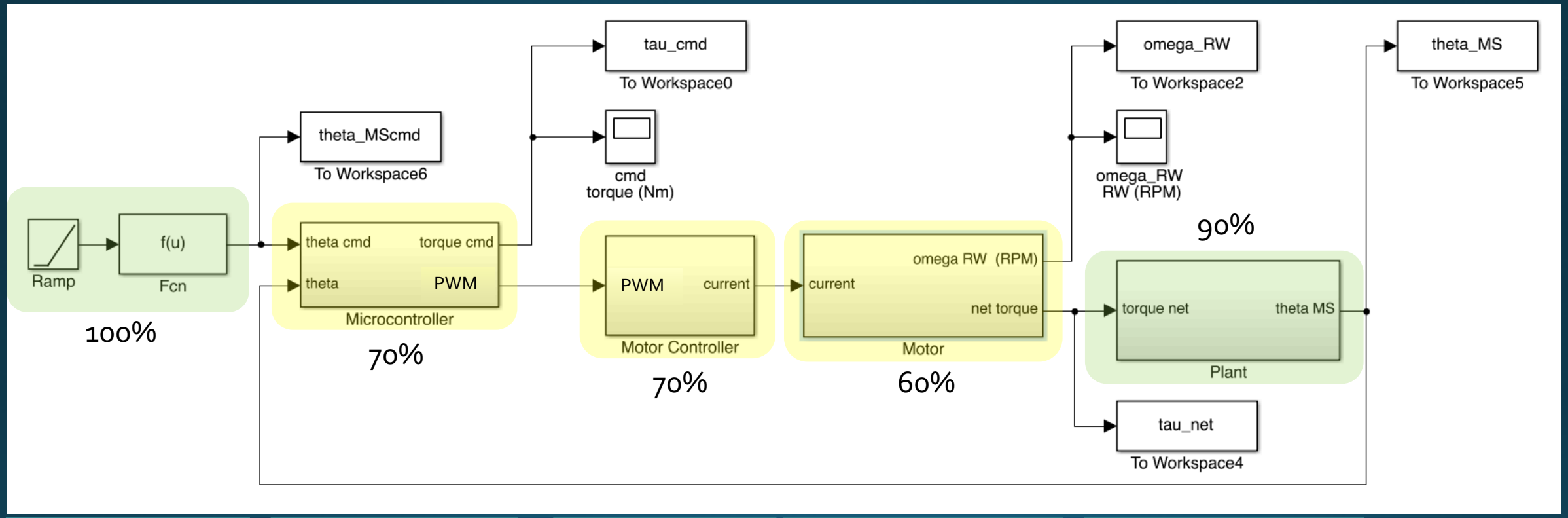
Sensors Status

Scope: What is involved?	Specific components	Status	Level of Confidence/ Main Concern
Hardware			
Pixy Cameras	Camera board, lens, connections	**80%**	**None: Missing percentages are delivery only**
Encoders	Breakout board, magnets, wiring/connections	**20%**	**Magnet Mounting Tolerance**
Software			
Coarse/Fine sensor calibration	Parameter tuning	**45%**	**Final Target Selection and Fine Lens Calibration**
Encoder configuration	Communication, performance confirmation	**15%**	**Converting demo-board info to final configuration**
Prototype replication	Replicate Arduino Prototype on SAM E70	**15%**	**Communication from Pixy to SAM E70**

Schedule – ADCS



Control Overview



Expected input signal derived and implemented	Need torque to voltage conversion (TF: ongoing)	Need PWM to current conversions built into simulation (TF: ongoing)	Need current/PWM to torque for the motor and engine parameters (TF: ongoing)	Need to rerun TestTable friction quantification (TF: 1 week)

Controller Status

Complete

- Simulation Runs
- Simulated Input Accurate
- Output Integrity (correct units)
- PID Tuning Procedure

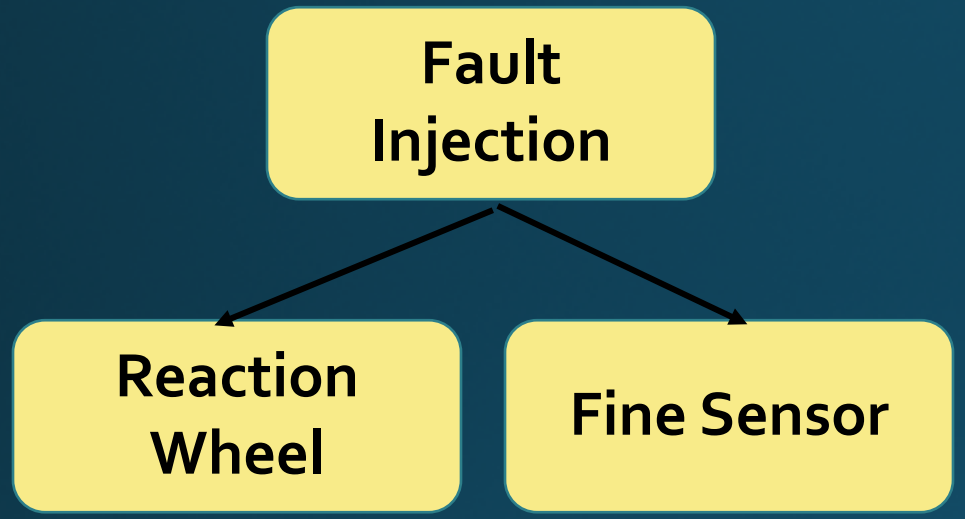
Incomplete

- Export to C++ Code
- Characterize Motor Parameters
- Identify Signal Types and Time Delays
- Mitigate Numerical Instability

Potential Problems

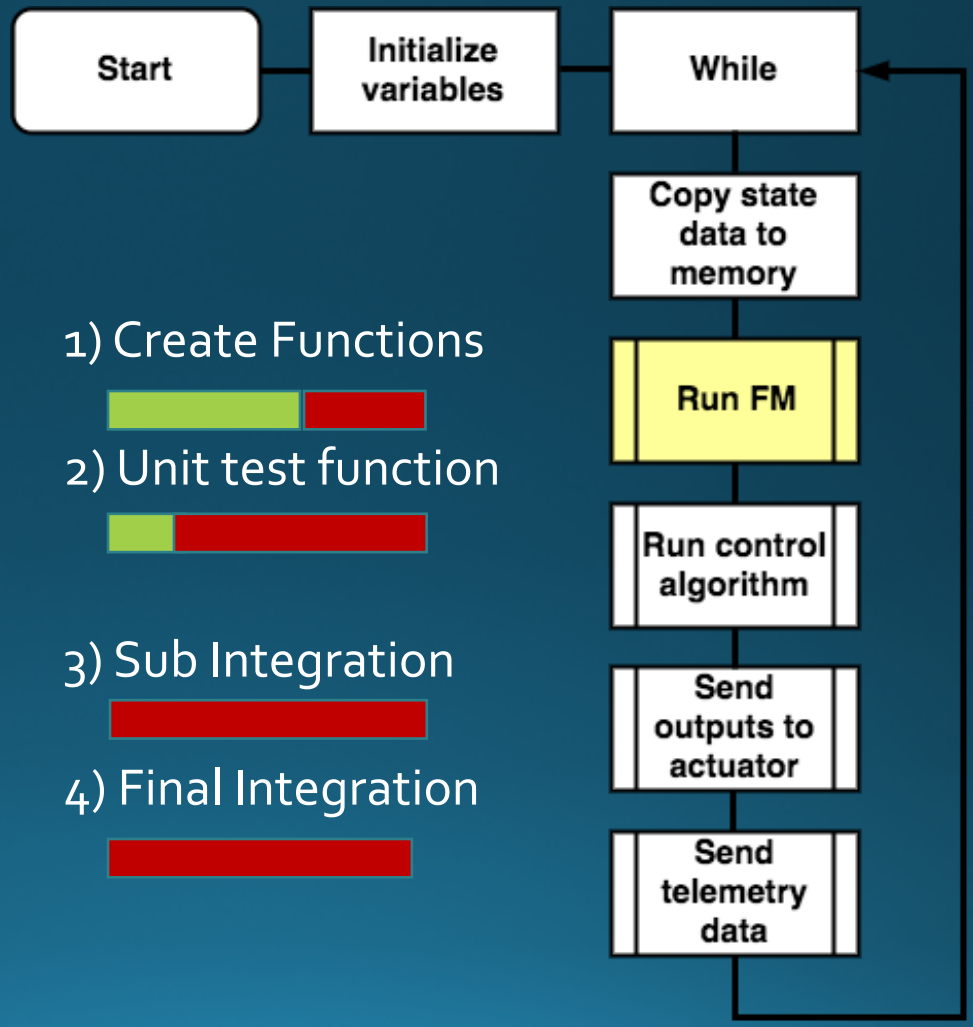
- SIMULINK understanding and compatibility with other software
- Motor testing and potential hardware swap
- Compatible SIMULINK integration
- Dividing by small numbers (shaft inertia) causes numerical blow ups

Fault Injection/Management



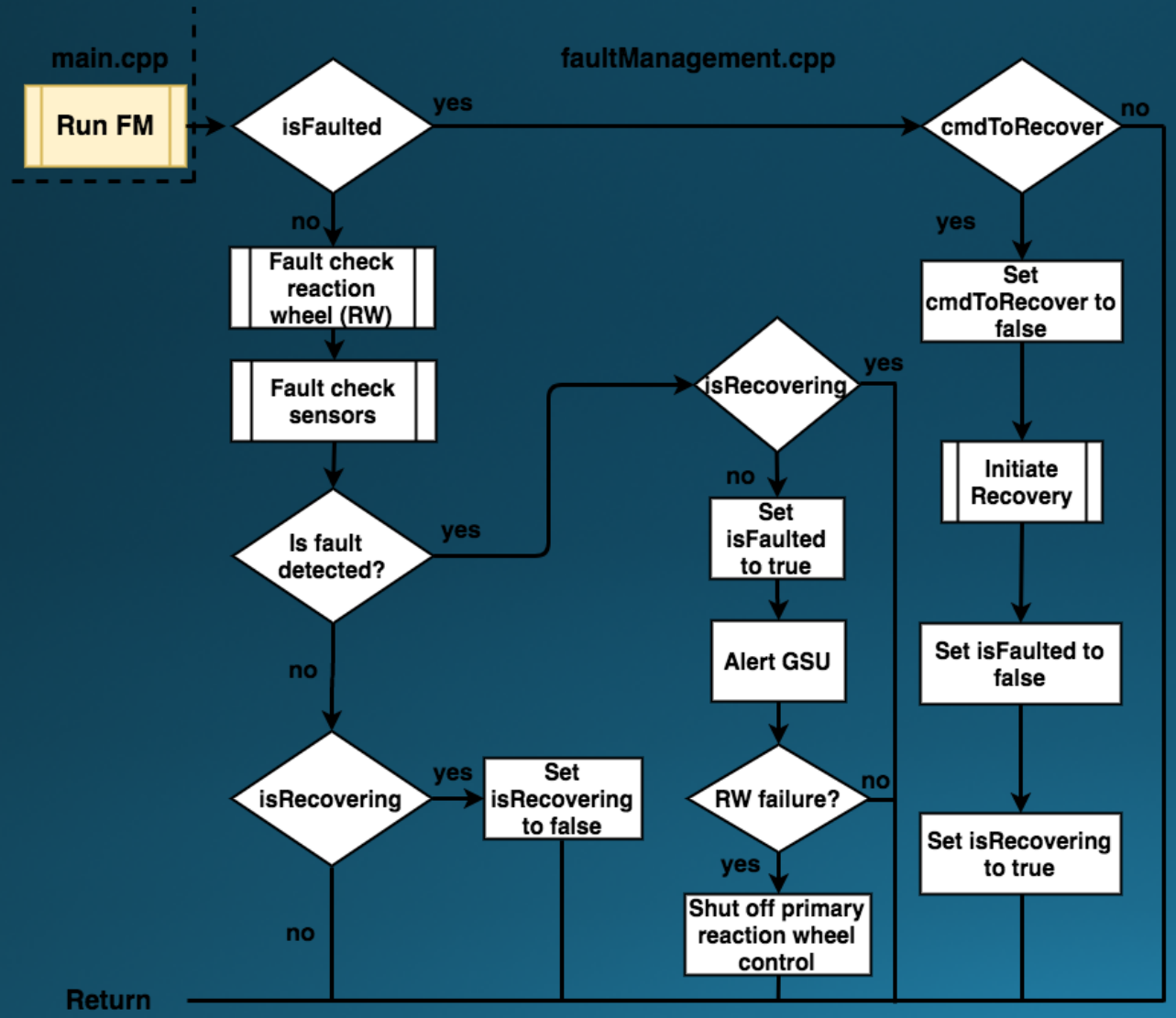
1) Create Function
2) Unit test function
3) Sub Integration
4) Final Integration

1) Create Function
2) Unit test function
3) Sub Integration
4) Final Integration



1) Create Functions
2) Unit test function
3) Sub Integration
4) Final Integration

Fault Injection/Management



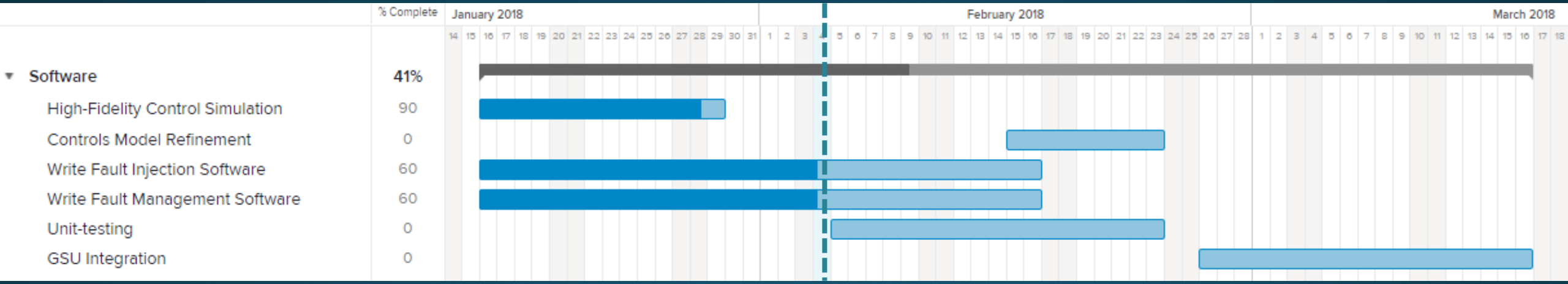
- Difficult and time consuming to unit test
  - Complex logic
  - Nested function calls
- Nested functions to fault check reaction wheel and sensors will be easier to unit test
- Alerting GSU will need assistance from communications
  - GSU has yet to be developed
  - No experience with Xbees
- Shutting off primary reaction control wheel requires help from someone with MCU experience

Fault Injection/Management

Concerns:

- Reaction Wheel Fault Injection: low SNR for nominal reaction wheel friction
  - This affects our ability to inject a reasonable friction to detect
    - Plan B is to increase the induced friction to a value that is easily detectable for the fault management system
- Reaction Wheel Fault Injection: classify the induced friction

Schedule – Software



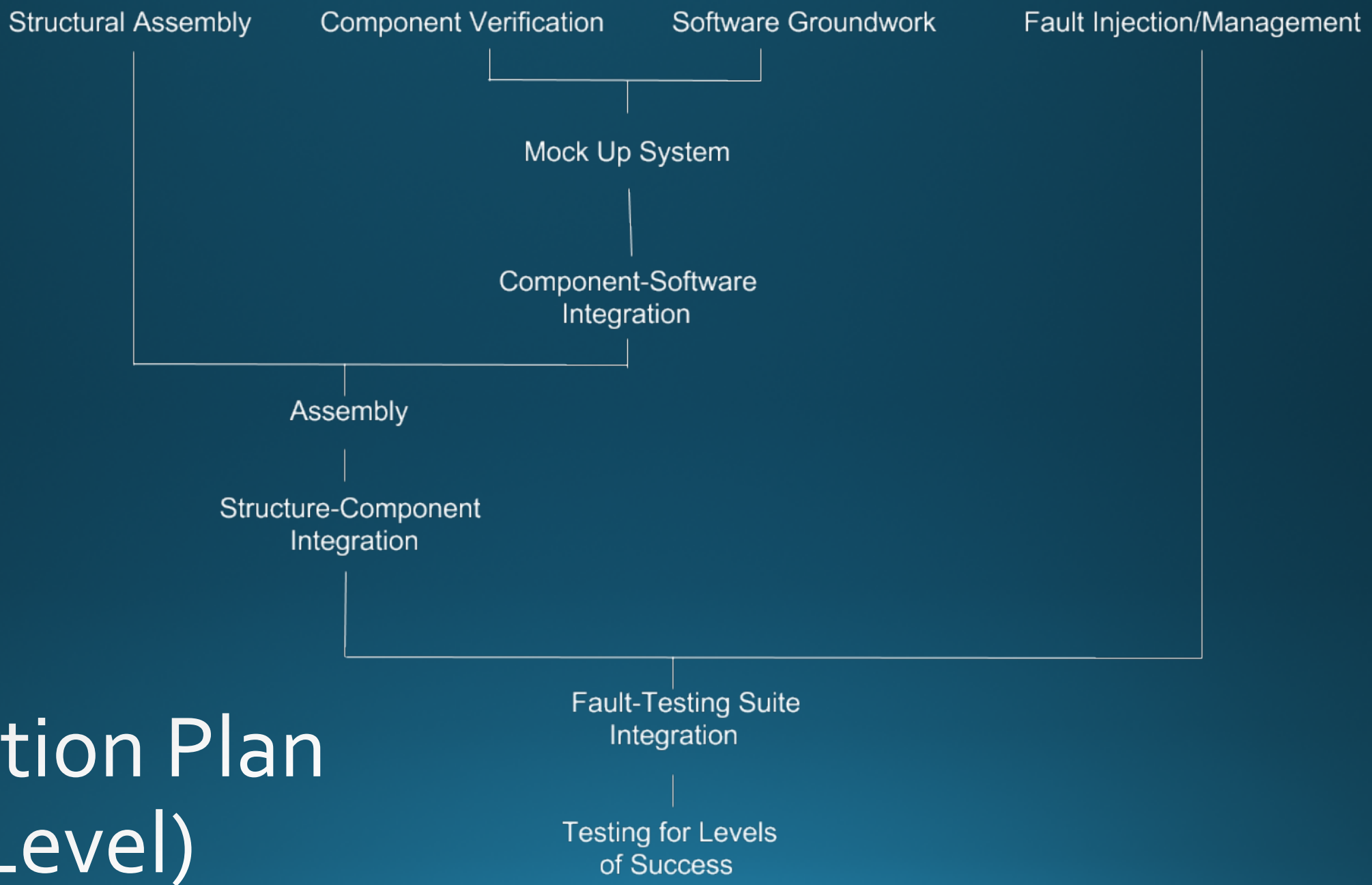
Manufacturing status for testing

TestTable-
- Blower---------- Completed
- Leveling System---------- Completed
- Mobile cart---------- Completed
- Encoder---------- Ordered
- Station Keeping Apparatus
  - Structure---------- Completed
  - Hardware---------- Completed

Reference Target Actuator-
- Housing---------- In-Hand
- Motor---------- In-Hand
- Arduino---------- In-Hand
- Light Source---------- In-Progress
- Arm---------- Determination Required
- Encoder---------- Ordered

MockSat-
- Structure---------- Completed
- Brackets/Mounts------- In-Progress
- Wiring---------- In-Hand
- Hardware
  - PCB---------- Designed and Ordered
  - MCU---------- In-Hand
  - Pixys---------- In-Hand awaiting lenses
  - Battery---------- Characterized and ready
  - Motors---------- In-Progress
  - Motor Controllers--- In-Progress
- Software
  - FI/FM---------- In-Progress
  - Controls---------- In-Progress

GSU-
- GUI ---------- Incomplete
- Software ---------- In-Progress

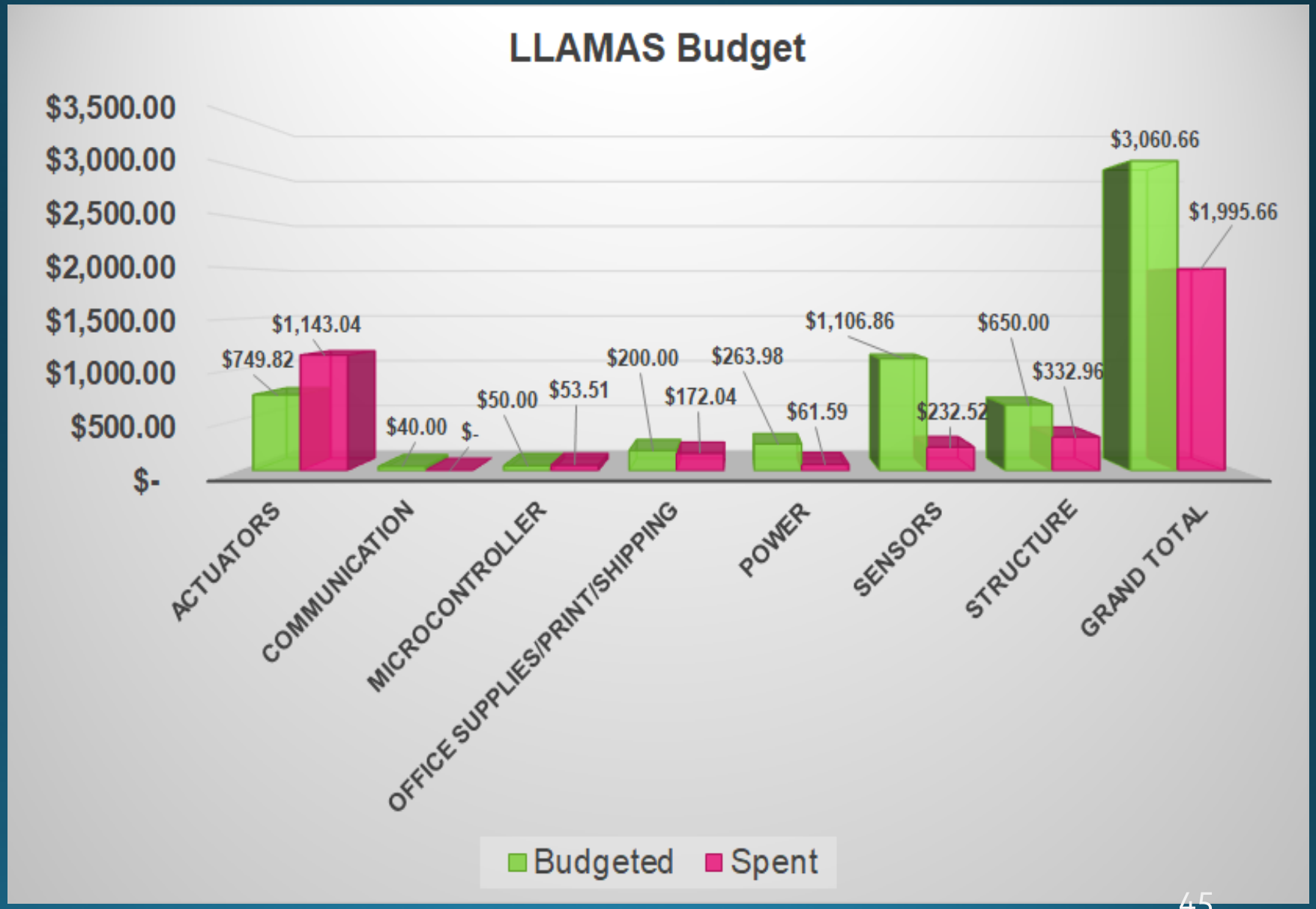


Integration Plan (High Level)

Budget

Budget

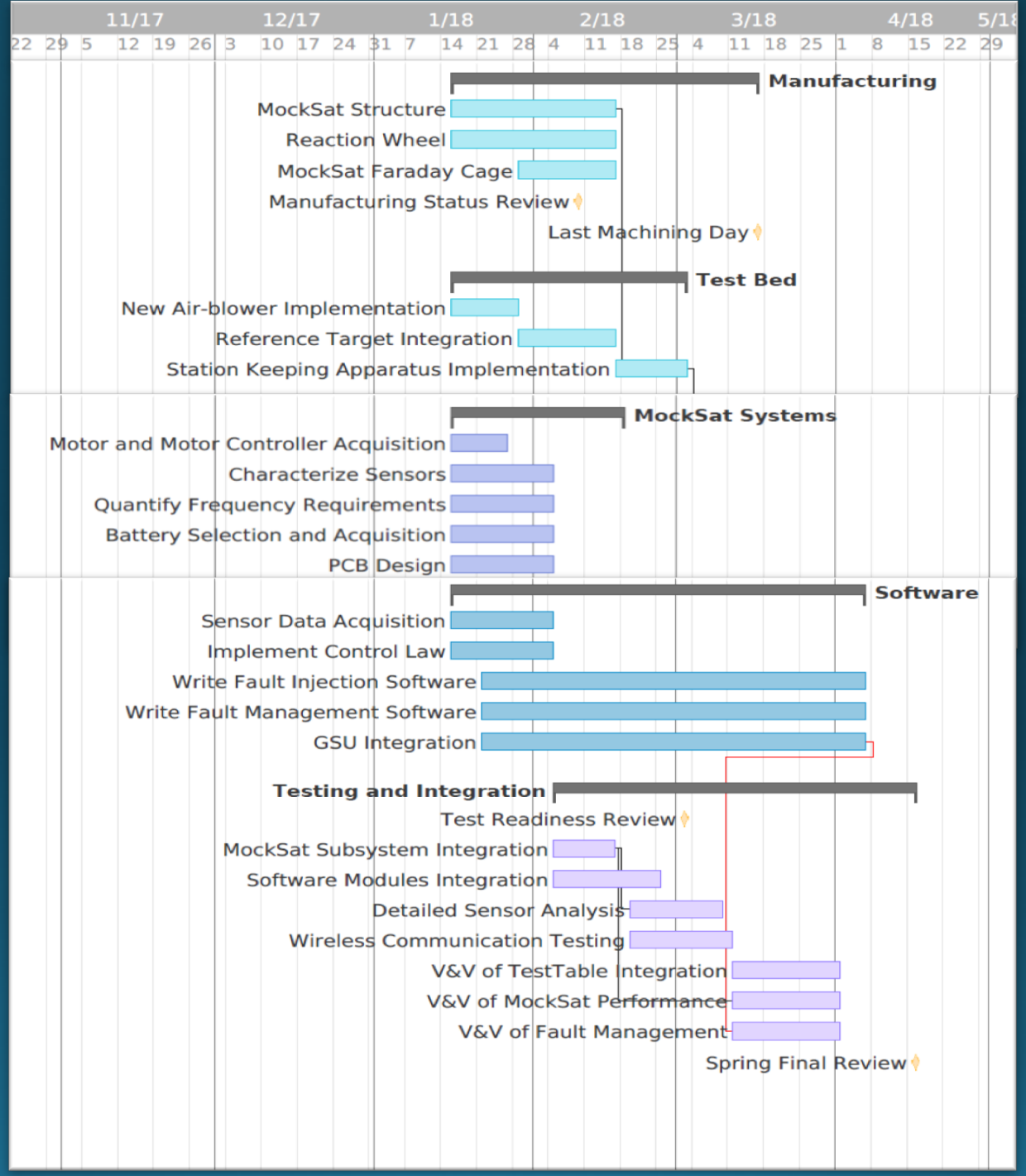
- Discrepancies derive from:
  - Purchase of extra motors to test two different families of motors
  - Lack of communication purchase
  - Overestimation of power
    - Have yet to purchase PCB
  - Pricy IMU deemed unnecessary
  - Overestimation of cost of structural material



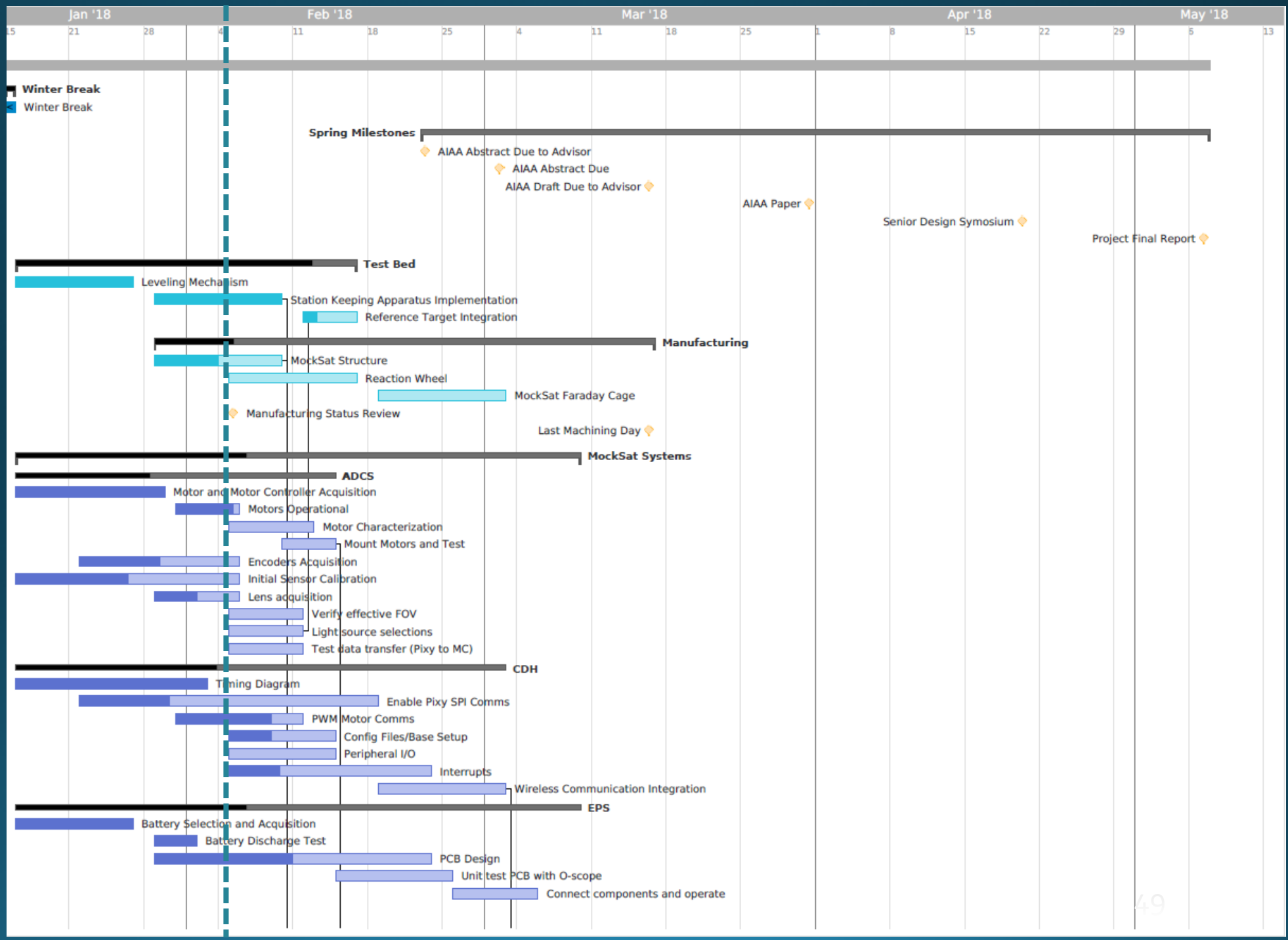
Questions

Backup Slides

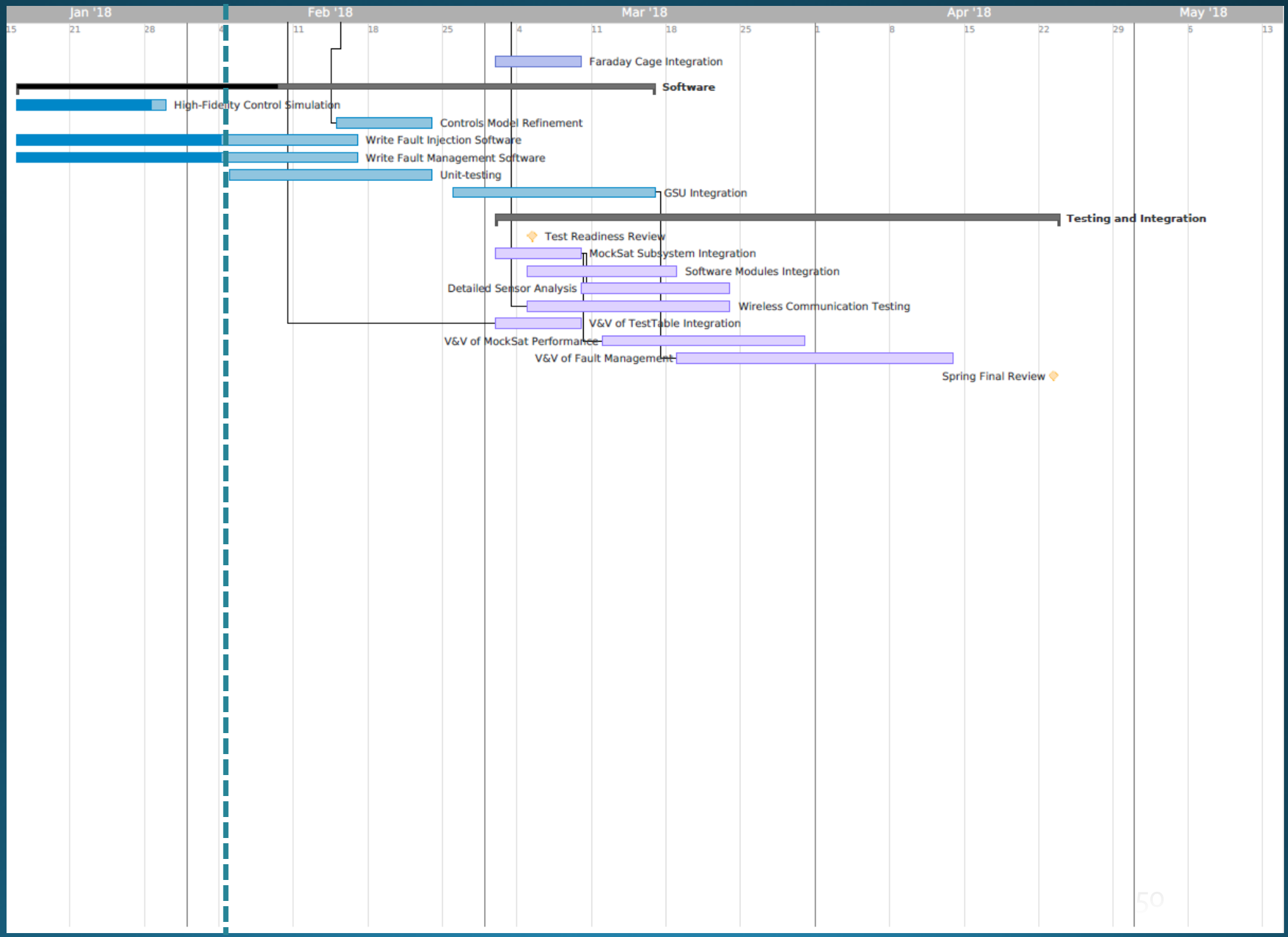
Schedule - CDR



Schedule


Jan '18
Feb '18
Mar '18
Apr '18
May '18
15
21
28
4
11
18
25
4
11
18
25
1
8
15
22
29
6
13
Winter Break
Winter Break
Spring Milestones
AIAA Abstract Due to Advisor
AIAA Abstract Due
AIAA Draft Due to Advisor
AIAA Paper
Senior Design Symosium
Project Final Report
Test Bed
Leveling Mechanism
Station Keeping Apparatus Implementation
Reference Target Integration
Manufacturing
MockSat Structure
Reaction Wheel
MockSat Faraday Cage
Manufacturing Status Review
Last Machining Day
MockSat Systems
ADCS
Motor and Motor Controller Acquisition
Motors Operational
Motor Characterization
Mount Motors and Test
Encoders Acquisition
Initial Sensor Calibration
Lens acquisition
Verify effective FOV
Light source selections
Test data transfer (Pixy to MC)
CDH
Timing Diagram
Enable Pixy SPI Comms
PWM Motor Comms
Config Files/Base Setup
Peripheral I/O
Interrupts
Wireless Communication Integration
EPS
Battery Selection and Acquisition
Battery Discharge Test
PCB Design
Unit test PCB with O-scope
Connect components and operate

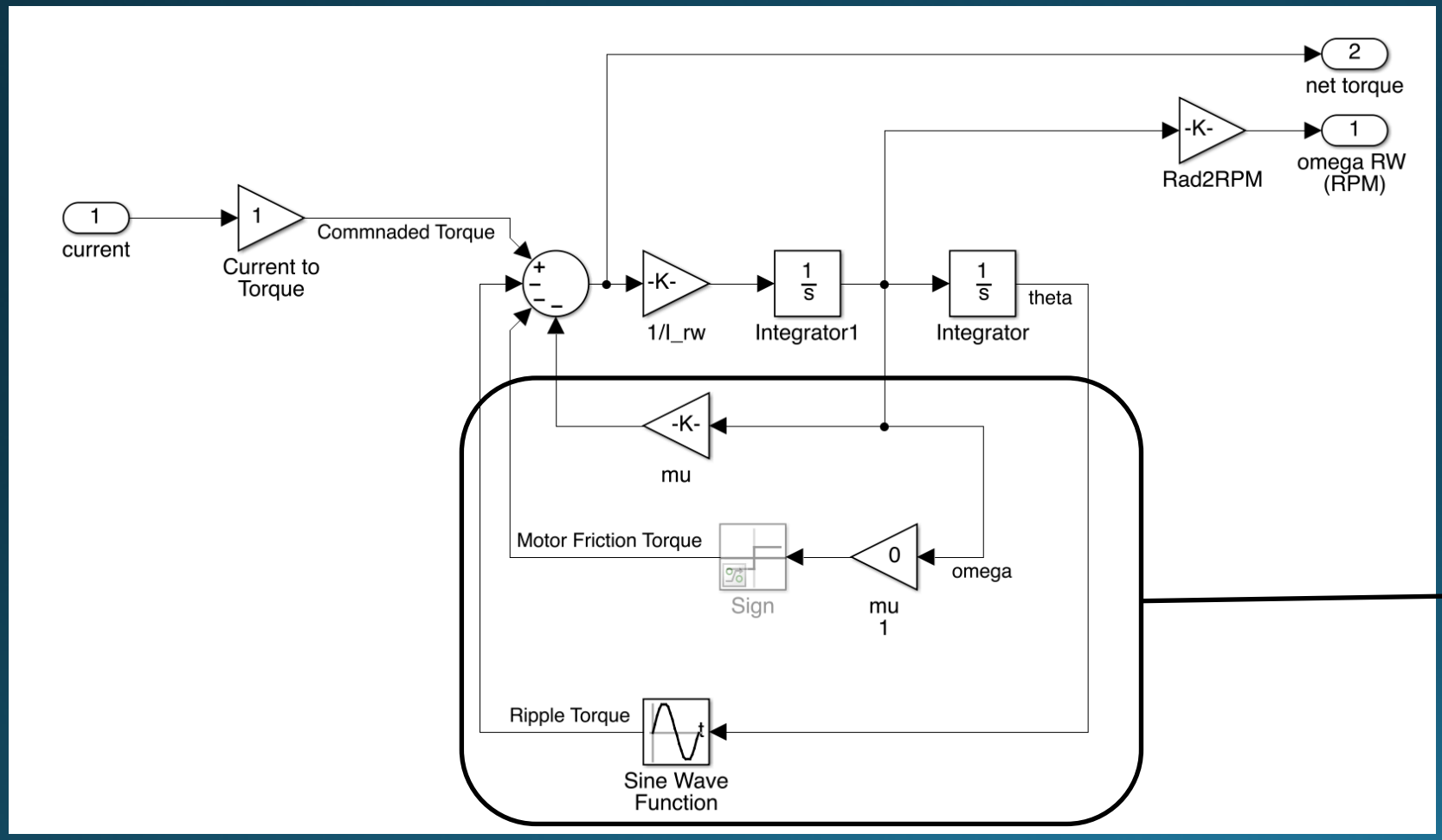
Schedule


Jan '18
Feb '18
Mar '18
Apr '18
May '18
15
21
28
4
11
18
25
4
11
18
25
1
8
15
22
29
5
13
Faraday Cage Integration
Software
High-Fidelity Control Simulation
Controls Model Refinement
Write Fault Injection Software
Write Fault Management Software
Unit-testing
GSU Integration
Testing and Integration
Test Readiness Review
MockSat Subsystem Integration
Software Modules Integration
Detailed Sensor Analysis
Wireless Communication Testing
V&V of TestTable Integration
V&V of MockSat Performance
V&V of Fault Management
Spring Final Review

Reference Target Actuator

- Using stepper motor coupled with Arduino to control speed.
- Structure is housing from spin modules.
- Arm length is to be determined, based off of seamless rotational speed as well as the 1 full rotation every 256 seconds.
- Source as of now is a full sized green 8W LED bulb.

Control Overview: Motor



Key To-Do Item:

- Test and Characterize Motor Parameters

Sensors Status

- Major Accomplishments:
  - Pixy- Motor Mockup gives confidence in programming the sensor outputs to perform to requirements for highest level of success.
- Future Challenges:
  - Encoders have a manufacturing tolerance of .25 mm (.0098 in) in order to minimize error.
  - Encoders have yet to arrive for characterization