

Flight Control System
PSAS Capstone: Spring-Summer 2016
Work Breakdown Structure

The Work Breakdown Structure for the PSAS Capstone Project lists deliverables based on project requirements for both the Capstone class as well as the Flight Control System for the Capstone project. The primary deliverables are the documentation, the code compilation for inflight navigation, the code compilation for simulating a flight environment to test the navigation software, and the presentation or demonstration of the software on the specified hardware. Additional “stretch components” comprising code that adds capability to the base software package if time and resources allow are also listed as are additional options for project demonstration. A preliminary work percentage is assigned to all specifications through level 3. Note that these will likely need to be updated once the more detailed Level 4 Specification is finalized.

Document deliverables are partitioned by intended objective i.e. to meet Capstone requirements, to allow for software maintenance and extension, to allow for software use, and to address internal needs. These are respectively labeled Capstone, Technical, End-User, and Non-Deliverable. Each of these categories is broken down further in the Level 4 Specifications which list the actual documents comprising the particular category.

Both the Flight and the Test Software deliverables are partitioned by functionality. For the Flight Software, this functionality is comprised of reading incoming data from a sensor using the I2C protocol, processing this data to determine a response, sending the response out to an actuator using the GPIO interface, and sending out telemetry data to the telemetry viewer using the PSAS Packet Definition.

For the Testing Software, this functionality is comprised of providing a simulated environment in which the Flight Software can run while capturing its output. Creating the simulated environment requires connecting to JSBSim, a popular software library, to retrieve flight simulation data and then using that data to simulate a sensor which then connects to the Flight Software. To capture the output of the Flight Software, the Test Software additionally simulates an actuator. The Test Software then feeds the data from the simulated actuator back into JSBSim in order to facilitate the next batch of simulated data. Level 4 Specifications detailing specific functionality and work tasks, while tentative at this stage, should further break down all deliverables into constituent code files or blocks.

The primary deliverable for Project Testing and Demonstrations is the final Capstone Presentation. This demonstrates the Flight Software on the RCS Prototype and the Testing Software on a generic Linux system. Secondary deliverables are internal in nature and at a macro level will likely include demonstrations of the Test Software as well as the Flight Software, the last on both the test hardware and the RCS Prototype. Level 4 Specifications here are also tentative at this point and should be developed further.

Additional “stretch goals” are listed in the Upgrades section. These include code that allows the Flight Software to run on the CubeSat Reaction Wheel Prototype; that allows the Test Software to run on a Windows 10 Platform; that extends Flight Software capabilities by allowing for more sensors, collecting more data, and, interfacing with non-GPIO actuators; and that extends the Testing Software by adding a Parser or Code Generator to provide additional testing options. This section will need to be further developed as the time necessary to create these components becomes available.

Work Breakdown Structure:

1.	Flight Control System		100%
1.1	Documentation		20%
1.1.1	Capstone	___%	
1.1.2	Technical	___%	
1.1.3	End-User	___%	
1.1.4	(<i>internal</i>) Non-Deliverable	___%	
1.2	Flight Compilation		25%
1.2.1	Core	10%	
1.2.2	Sensor Interface	5%	
1.2.3	Actuator Interface	5%	
1.2.4	Telemetry Interface	5%	
1.3	Test Compilation		30%
1.3.1	Test Controller	10%	
1.3.2	JSBSim Binding	10%	
1.3.3	Simulated Sensor	5%	
1.3.4	Simulated Actuator	5%	
1.4	Capstone Project Demo		10%
1.4.1	Test Compilation Demo on Generic Linux System	___%	
1.4.2	(<i>internal</i>) Flight Compilation Demo on Test System	___%	
1.4.3	(<i>internal</i>) Flight Compilation Demo on RCS Prototype	___%	
1.4.4	Flight Compilation Demo on RCS Prototype	___%	
1.5	Upgrades		15%
1.5.1	Flight Compilation Demo on CubeSat Reaction Wheel Prototype	___%	
1.5.2	Functional Windows 10 Compilation	___%	
1.5.3	Extended Sensor Functions	___%	
1.5.4	Arbitrary Data Bus	___%	
1.5.5	Non-GPIO Controllers	___%	
1.5.6	Parser / Code Generator to Test New Constants	___%	

1.1 Documentation: Level 4 Specification

- | | | |
|-------|---|--|
| 1.1.1 | Capstone: | As required by Professor Massey. |
| | 1.1.1.1 Requirements and Specifications | |
| | 1.1.1.2 Risk Management | |
| | 1.1.1.3 Work Breakdown Structure | |
| | 1.1.1.4 Architecture | |
| | 1.1.1.5 Project Schedule | |
| | 1.1.1.6 | |
| 1.1.2 | Technical: | Required to compile & maintain software |
| | 1.1.1.1 Compilation Notes | |
| | 1.1.1.2 | |
| | 1.1.1.3 | |
| 1.1.3 | End-User: | Necessary to use software. |
| | 1.1.1.1 FAQ | |
| | 1.1.1.2 | |
| | 1.1.1.3 | |
| 1.1.4 | Non-Deliverable: | Internally required but not deliverable. |
| | 1.1.1.1 Team Contact Data | |
| | 1.1.1.2 | |
| | 1.1.1.3 | |

1.2 Flight Compilation: Level 4 Specification

- | | | |
|---------|-----------------------------|---|
| 1.2.1 | Core: | Process Data and Determine Response. |
| 1.2.1.1 | Process Data | |
| 1.2.1.2 | Determine Response | |
| 1.2.1.3 | Package Telemetry | |
| 1.2.2 | Sensor Interface: | Provide interface to Sensor Hardware. |
| 1.2.2.1 | Sensor Connect | |
| 1.2.2.2 | Sensor Get Data Loop | |
| 1.2.2.3 | Sensor Disconnect | |
| 1.2.3 | Actuator Interface: | Provide interface to Actuator Hardware. |
| 1.2.3.1 | Actuator Connect | |
| 1.2.3.2 | Actuator Send Response Loop | |
| 1.2.3.3 | Actuator Disconnect | |
| 1.2.4 | Telemetry Interface: | Provide interface for Telemetry Output. |
| 1.2.4.1 | Telemetry Connect | |
| 1.2.4.2 | Telemetry Send Data Loop | |
| 1.2.4.3 | Telemetry Disconnect | |

1.3 Test Module: Level 4 Specification:

- | | | |
|---------|---------------------------------|--------------------------------------|
| 1.3.1 | Test Controller | Integrate Test Compilation Functions |
| 1.3.1.1 | Process JSBSim Data | |
| 1.3.1.2 | Process Simulated Actuator Data | |
| 1.3.2 | JSBSim Binding | Provide Interface to JSBSim |
| 1.3.2.1 | JSBSim Connect | |
| 1.3.2.2 | JSBSim Send / Receive Loop | |
| 1.3.2.3 | JSBSim Disconnect | |
| 1.3.3 | Simulated Sensor | Simulate Sensor Functionality |
| 1.3.3.1 | Simulate Sensor Connection | |
| 1.3.3.2 | Get Data from Test Controller | |
| 1.3.3.3 | Send Data to Core | |
| 1.3.3.4 | Simulate Sensor Disconnection | |
| 1.3.4 | Simulated Activator | Simulate Actuator Functionality |
| 1.3.4.1 | Simulate Actuator Connection | |
| 1.3.4.2 | Get Data from Core | |
| 1.3.4.3 | Send Data to Test Controller | |
| 1.3.4.4 | Simulate Actuator Disconnection | |

1.4 Capstone Project Demo: Level 4 Specification

1.4.1 Test Compilation Demo on Generic Linux System

1.4.1.1 Set Up

1.4.1.2 Demo

1.4.1.3 Tear Down

1.4.2 *(internal)* Flight Compilation Demo on Test System

1.4.2.1 Set Up

1.4.2.2 Demo

1.4.2.3 Tear Down

1.4.3 *(internal)* Flight Compilation Demo on RCS Prototype

1.4.3.1 Set Up

1.4.3.2 Demo

1.4.3.3 Tear Down

1.4.4 Flight Compilation Demo on RCS Prototype

1.4.4.1 Set Up

1.4.4.2 Demo

1.4.4.3 Tear Down

- 1.5 Upgrades: Level 4 Specification
 - 1.5.1 Flight Compilation Demo on CubeSat Reaction Wheel Prototype
 - 1.5.1.1 Set Up
 - 1.5.1.2 Demo
 - 1.5.1.3 Tear Down
 - 1.5.2 Test Compilation Demo on Windows System
 - 1.5.2.1 Set Up
 - 1.5.2.2 Demo
 - 1.5.2.3 Tear Down
 - 1.5.3 Extended Sensor Functions
 - 1.1.3.1 Multiple Sensors
 - 1.1.3.2 Enhanced Data Types and/or Volume
 - 1.5.4 Arbitrary Data Bus
 - 1.5.4.1
 - 1.5.5 Non-GPIO Controllers
 - 1.5.5.1
 - 1.5.6 Parser / Code Generator to Test New Constants
 - 1.5.6.1