Document Description: ADIRU Test Plan	Revision: 2
ADIRU Test Plan	Authors: Chris Andrews, Kayla Seliner, Trang Nguyen, Mark Craig

Revision	Date	Note	Initial
0	2012-2-03	Format incorporated from WAAM (2011 PSU practicum).	CA
1	2012-2-13	Power, Bus, and sensor tests added	CA
2	2012-2-19	Formatting improved	CA

Table of Contents

1. Purpose

This document explains the component, system and integration tests to verify functionality of the the Air Data Inertial Reference Unit implemented by the 2012 PSU ADIRU Capstone team.

2. Scope

Complete information necessary to implement the ADIRU test plan is contained or referenced in this document. This includes testing methods, test instructions and forms, and criteria for evaluating tests. Test equipment is also listed.

3. References

All code necessary for software testing that is referenced in this document is contained in the project wiki ADIRUproject.

4. Definitions

ADIRU: Air Data Inertial Reference Unit
GPS: Global Positioning System
IMU: Inertial Measurement Unit: incorporates gyroscopes for angular displacement and accelerometers for detecting linear displacement.

5. Test Overview

The test strategy follows a bottom up approach:

- Individual components are verifed
- Systems are verified
- System level software is verified

The inputs and outputs of each module of the system are verified before testing progresses. In this way faults can be identified and troubleshot at the earliest opportunity.

5.1 Tools Techniques and Methodologies

- Functional Testing. Tests designed to verify that all the functional requirements have been satisfied. This test category is termed success oriented testing because the tests are expected to produce successful results.
- Robustness Testing. Tests designed to evaluate performance given unexpected inputs. This test category is termed failure oriented because the test inputs are designed to cause the product to fail given foreseeable and reasonably unforeseeable misuse of the product.
- Stress Testing. Tests designed to evaluate software in a stress condition in which the amount or rate of data exceeds the amount expected. There are also hardware stress tests that evaluate hardware performance in response to bad data or bus function.

- Regression Testing. Performed whenever a change is made in software to evaluate that the system has maintained functionality.

6. Integration Test Strategy

Integration testing will focus on ensuring that device sub-systems work as specified within the system, focusing on startup activities, shutdown, and data flow across internal and external interfaces.

7. System Test Strategy

Confirm that requirements as listed in the requirements document are met by passing certain tests.

7.1 Anomaly Reporting

Any failures or exceptions during tested will be noted.

8. Test Protocols

8.1 Power

8.11 Power Supply

Requirements: 5.12.1

Component(s) under test: Each board with a power supply will have a variatint of this test.

Purpose: Verify stability of power supply

Setup: Attach jumpers to Vsupply and Gnd between board and power supply.

Type of Test: functional test

Acceptance Criteria:

1. This test passes if all individual tests are (yes)

Test Instrumentation:

- Power Supply with adjustable voltage output
- Multimeter

Step	Description	Expected Result	Yes/No
1.	Attach jumpers to Vsupply and GND from power supply.		
2.	Set power supply to battery voltage high range (4.2 V if LiPo battery) and measure Vcc on the board.	This is specific to the board. Maybe 1.8V, 3.3V, 3.5V 5.0V? The output voltage should be within some small millivolt range of target. Current should be within the target range of the the board.	
3.	Set power supply to low end of battery range (2.5 V if LiPo battery)	Vcc on the board should not change more than a few millivolts. Current should be appropriate level	
4.	Toggle the power switch on the board (if any)	Power should go on and off appropriately.	

8.12 Battery Test

Requirements: 5.12.1

Component(s) under test: Battery

Purpose: Verify battery charge capacity

Setup: Attach multimeter1 in series with the battery terminals and a 1kOhm resistor(or other resistor with resistance comparable to system) (to measure curent). Attach multimeter2 in parrallel (to measure voltage)

Type of Test:

1. Acceptance Criteria:This test passes if all individual tests are (yes)

Test Instrumentation:

- two multimeters

Step	Description	Expected Result	Yes/No
1.	Connect the jumpers to appropriate power and supply terminals and mark when the voltage on the battery drops below some threshold that marks safe operation of the battery (like 3.0V)	Battery should maintain voltage above the floor theshold (3.0V) for the expected operational time of the battery (10min, or 30 min).	

8.2 Module Interfacing

8.21 I2C Communication

Requirements: 5.2.1

Component(s) under test: I2C Bus

Purpose: Verify communication between different modules.

Setup: All components under test must be powered and hooked up to the appropriate I2C bus.

Type of Test: Functional

Test Instrumentation:

- Appropriate cable connections between component under test and PC with serial terminal window.
- SW code to send/receive data

Step	Description	Expected Result	Yes/No
1.	Power up boards under test. Connect appropriate cable to appropriate terminals.		
2.	Load SW onto appropriate components		
3.	Iniate SW communication program	Should see appropriate data going between modules	
4.	Delete test code from device		
5.	Power Down		
6.	Add additional microcontroller on I2C bus with appropriate device address	Bus should function and be expandable.	

8.3 Mechanical Integrity

The weight of the flyable system will be evaluated against the payload ability of the aircraft

The ADIRU must be properly insulated from heat/rain and excessive mechanical shock.

8.5 Radio

Requirements: TBD

Component(s) under test: Radio that controls RC aircraft

Purpose: Verfy basic functionality of radio

Setup: With aircraft on ground and engine off, manipulate the rudder, elevator, and aileron controls.

Type of Test: Functional

Test Instrumentation:

Step	Description	Expected Result	Yes/No
1.	Manipulate rudder, elevator, aileron controls	Rudder, elevator and aileron should all move in appropriate directions	

8.6 Sensors

8.61 Altimeter

Requirements: 5.3.4

Component(s) under test: Altimeter

Purpose: Verify precision and accuracy of altimeter

Setup: Power up the altimeter board. Attach multimeter in series with output terminals.

Type of Test: Functional

Test Instrumentation:

- Multimeter
- Computer with Google Earth Software to find exact altitude on map
- Aircraft
- Datalogger

Note, test steps 1 – 4 should be done independently for each altimeter.

Step	Description	Expected Result	Meas.
1	Static test:Measure and record voltage on the ground with multimeter.	According to the formula in the datasheet for the altimeter, the altitude is a function of the differential voltage of the outputs.	
2.	Monitor the voltage for 5 minutes with multimeter to determine drift (if any)	Voltage should remain stable within x millivolts.	
3.	Compare calculated altitude with actual altitude (from google earth)	Expect calculated altitude to be within x meters of real altitude.	
4.	Dynamic Test: Measure and record differential voltage while climbing/descending in airplane.	Altitude should increase/decrease	
4.	Compare sensor readings between all four altimeters for precision during static and dynamic tests.	They should all correlate within x millivolt spread	

8.62 GPS

Requirements: 5.3.3, 5.3.5, 5.3.6

Component(s) under test: GPS

Purpose: Determine accuracy, time to first fix, precision.

Setup: Go to an unobstructed outside location. Power up the GPS board and log the data to memory or a terminal window.

Type of Test: Functional

Test Instrumentation:

- GPS unit with antennae
- appropriate cable to computer or memory card.
- PC with google earth loaded.

Step	Description	Expected Result	Meas.
1.	Record time. Power up the GPS and record time when position is fixed.	This is the time to first fix	
2.	Verify position accuracy with respect to google earth		
3.	Drift measurement of precision. (static test) Log the data over five minutes and note the size and shape of the distribution of locations.		
4.	Plot some predetermined points on a map. Drive through these locations noting the time of arrival at each location. Include some 90 and 180 changes in direction.See how the GPS tracks these changes.		

8.63 Airspeed

Requirements: 5.3.1

Component(s) under test: Airspeed Indicator

Purpose: Verify functionality of airspeed indicator

Setup: Pitot tube must be attached to airplane and logging data to the SD or computer OR perhaps we can run this test in a wind tunnel if we can locate a piece of this equipment.

Type of Test: Functional

Test Instrumentation: Aircraft or wind tunnel. A means to log or see the data.

Step	Description	Expected Result	Yes/No
1.	Record airspeed = 0 readings	0	
2.	Record the airspeed readings as the airspeed increases well above the cruising speed of the aircraft	The airspeed measurements should increase and decrease to track the airspeed of the aircraft	

8.64 Inertial Measurement Unit

Requirements: 5.3.2, 5.3.8, 5.3.10

Component(s) under test: IMU board

Purpose: Determine that IMU board has functionality of gyroscopes, accelerometers and magnetometer.

Setup: Power up IMU board and connect with serial port to computer. Load test program onto IMU board.

Type of Test: Functional

Test Instrumentation: Computer, and FTDI cable.

Step	Description	Expected Result	Yes/No
1.	Verify data generation	Accelerometer, gyroscope, magnetometer data should be displayed over serial port.	
2.	Place IMU motionless on table	Data should not show any significant variation or drift over time.	
3.	Leave IMU motionless and flat on table.	Z axis accelerometer should show 9.8 m/s^2 acceleration. X and Y axis should be negligible.	
4.	Shake the IMU	Shaking the IMU more vigorously should produce larger accelerometer values	
5.	Rotate and spin IMU	Magnetometer and gyroscope data should move.	

Datalogging (basic functionality)

Requirements: 5.1.21

Component(s) under test: EEPROM card on appropriate board

Purpose: Verify operation of the memory system

Setup: SD board connected to computer with appropriate cable.

Type of Test: functional

Test Instrumentation: Computer and appropriate wires or cables to connect to EEPROM board.

Step	Description	Expected Result	Yes/No
1.	Power up EEPROM board and connect to computer via Serial cable. Load test software.		
2.	Write data to the EEPROM board. Read the memory location to determine that data was correctly written to EEPROM	Data should write to EEPROM. Data should be read from EEPROM	
3.	Check data logging frequency	Should match requirement	

Datalogging (advanced functionality)

Requirements: 5.1.8

Component(s) under test: System

Purpose: Verify that the EEPROM can log the data from each uP on the i2c bus

Setup: Setup ADIRU benchtop system

Type of Test: Functional

Test Instrumentation: Computer, appropriate cable.

Step	Description	Expected Result	Yes/No
1.	Power up four node ADIRU benchtop system. Load test software.		
2.	Run the system. After test, read EEPROM and see that there is data for each node plus the voted outcome at every sample time in the test interval.	There should be data for each node plus the voted outcome at every sample time in the test interval.	

Fault Injection

Requirements: 5.1.4

Component(s) under test: Fault Injection micro controller

Purpose: Verify that faults can be injected to system

Setup: ADIRU benchtop system plus fault injection micro controller

Type of Test: Functional

Test Instrumentation: Computer with appropriate serial cable

Step	Description	Expected Result	Yes/No
1.	Power up ADIRU benchtop system with fault injection microcontroller and load test software.		
2.	Run the system for a time interval with the FI microcontroller introducing certain faults to the data on the bus.		
3.	Read the data from the EEPROM	Verify that the faults injected caused the correct changes in the voting algorithm	

Software

Requirements: 5.5.1, 5.1.14, 5.1.15, 5.1.16

Component(s) under test: Software

Purpose: Verify that software is correct

Setup: Load software into arduino IDE

Type of Test: Functional

Test Instrumentation: Computer

Step	Description	Expected Result	Yes/No
1.	Compile source code	Code should compile	
2.	Should use good programming techniques to separate program into functions separate header files according to functionality		

Voting Outcomes

Requirements: 5.1.11

Component(s) under test: System

Purpose: Verify function of voting algorithm running on the hardware system

Setup: Load software to ADIRU benchtop system.

Type of Test: Functional

Test Instrumentation: Computer and appropriate cables

Step	Description	Expected Result	Yes/No
1.	Examine voting data on EEPROM after test run	Correct results should be displayed. Single faults should be masked. Single byzantine faults should be masked	

Clock Synchronization

Requirements: TBD

Component(s) under test: Voting microcontrollers

Purpose: Verify clock synchronization between microcontrollers

Setup: Microcontrollers under test should all be connected on I2C and powered up

Type of Test: Functional

Test Instrumentation: TBD

Step	Description	Expected Result	Yes/No
1.	The synchronization code is still under development and the method of verification is unclear at this time.		

Bus

Requirements:

Component(s) under test: I2C bus

Purpose: Verify functionality of I2C bus

Setup: Connect oscilloscope test lead 1 to the SCL line and connect test lead 2 to the SDA line. Connect ground on the test board to ground on the oscilloscope. Power up the test board. Load test program on boards and run.

Type of Test: Functional

Test Instrumentation: Oscilloscope (Logic Analyzer would also work)

Step	Description	Expected Result	Yes/No
1.	Verify clock speed of the bus	Should be 100k bits/second	
2.	Verify wave form of the bus SDA and SCL lines.	The rise and fall time should be within the limits of the I2C specification. The large the RC value of the bus (bus capacitance x Rp pullup resistor), the larger rise and fall time.	
3.	Verify that all of the different microcontrollers on the I2C that can vote have master and slave capability	As the voting algorithm cycles, each microcontroller should in turn use the bus as a master and as a slave.	