
Test Document for Autonomous Multi-cycle Farming in Space



Team Members:

Name	Contact
Client & Sponsor: Philip Chan PhD	pkc@cs.fit.edu
Christopher Milsap	cmillsap2013@my.fit.edu
Giampiero Corsbie	gcorsbie2018@my.fit.edu

Test Document Version Control:

Document ID: TD-000001				
Revised Version	Date	Collaborators	Version	Reason
N/A	9/20/2019	All	v1.0	Initial creation

¹

¹ When updating this document's version update the version found on the title page, name, referenced documents, baseline and footer ↘

Table of Contents:

Test Document Version Control:	2
Table of Contents:	3
Introduction	3
Approach	3
System Operation	3
Configuration File	4
Item Pass/Fail Criteria	4
Pass	4
Fail	5
Test Deliverables	5
Test Environment	5
Simulated Farm	5

Introduction

Throughout all phases of development for the AMCFD, testing will be required to ensure quality. Testing will ensure not only that the device is operating as intended, but also that there are no bugs in the software. This test plan will present the details of how functionality will be tested.

Approach

System Operation

To ensure proper testing, each functionality will be tested in multiple, increasingly complex steps. A simulated farm will be running, providing raw data to the sensor classes. The software will respond with commands to the actuator classes, which will operate on the simulated farm and change its state. The first run will be a single stage, going through the respective queue of actions and jobs. The second run will operate through one cycle, consisting of each stage and its

transitions. The third run will have the simulated farm and software go through multiple cycles. With each testing session following the format of the matrix below:

	Simulated Farm	Sensor	Actuator
Single Stage	State cases	Expected data behavior for features	Expected actions and reactions for features
One Cycle	State and transitions cases	Expected data behavior for features	Expected actions and reactions for features
Multi Cycle	Multiple sets of state and transition cases	Expected data behavior for features	Expected actions and reactions for features

As the software subsystem operates it will log all information about control flow and data transfer between it and the simulated farm. The simulated farm will be configured to present edge cases to the software, then the operations of the software will be logged and compared to expected behavior.

Configuration File

Additionally, the methodology of processing user configuration files must be tested. The software will be provided with valid configuration files and log its operations for later comparison against expected behavior. These will be tested in the same multi-step format as is described above.

To determine whether or not the configuration file is being parsed correctly, an invalid configuration file will be provided to the software. Any exceptions thrown will be compared to the expected list of exceptions for that file.

Item Pass/Fail Criteria

Pass

A test item shall be deemed to pass if and only if it has successfully achieved behavior for each input and if the code corresponding to that item has been reviewed through static and dynamic analysis.

Fail

A test item shall be deemed to fail if it has not achieved the intended behavior for its respective input, or if the code corresponding to that item has been rejected by the review procedures.

Test Deliverables

- Test Plan (this document itself)
- Test Cases and Matrices
- Test Scripts
- Defect/Enhancement Logs
- Test Reports

Test Environment

Simulated Farm

There shall be a separate software subsystem to simulate the behavior and timescales of a farm. This will be used to provide input to the primary software subsystem for testing purposes. All features of its behavior will be configurable to account for any test item.