Software Requirements Specification for Autonomous Multi-cycle Farming in Space

Team Members:

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

Requirements Document v1.0
Created: 9/18/2019
Revised: 9/24/19

Requirements Document Version Control:

Document ID: RD-000001				
Revised Version	**Date**	**Collaborators**	**Version**	**Reason**
N/A	9/18/2019	All	v1.0	Initial creation

[1]

Document Reference:

Log of all document references where version number is needed. To maintain version control.

Document	Date Refenced	Version Referenced
Project Plan v1.0	9/23/19	v1.0

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

Table of Contents

Introduction

Project Definitions:

Word/Phrase	**Re-worded/*Notes*:**	**Definition**
Autonomous Multi-cycle Farming Device	The AMCFD	The subject of this project and the product being created
Decision maker	Driver	The main head process in the system running on the microcontroller
User(s)	*All users are included in the set of stakeholders but not all stakeholders are users. All users are actors.*	The astronaut(s) in direct contact with the AMCFD
Reaction thresholds		The reading from a sensor set against a range of values that determine the usefulness of the reading
On-mission		Collection of activities carried out on the spacecraft by users
Ground/Ground control		The collection of actors that carry out on-mission activities at a remote location
Fully autonomous		The characteristic of the AMCFD where the device runs continuously until resources are depleted or automation is halted
On-device varied manual interaction		The AMCFD will have controls and options available

		to the on-mission users where phases of the system can be completed/interrupted manually by hand or via actuator controls
Ground varied manual interaction	*Controlling and monitoring the AMCFD does not make ground included in the set of users. The boundary for being defined as a user is proximity, where physical contact can be achieved*	The AMCFD will have controls and options available to the on-mission stakeholders off-site where phases of the system can be completed/interrupted manually by ground control.
API varied manual interaction		The AMCFD will have controls and options available to the on-mission users where phases of the system can be completed/interrupted manually by software hosted on another machine within a specified proximity
SS		Software subsystem. The entire software package encompassing all software in the AMCFD
Phases		A set of processes defined by seeding, growing, harvesting, and cleaning operations done in a cyclic sequence within the AMCFD.
PAS		Proactive Autonomous System: A subsystem in the autonomous system that is responsible for repeatable tasks in the phases

RAS		Reactive Autonomous System: A subsystem in the autonomous system that reacts to exceptions and changes in the environment in the AMCFD
Worker		Any sensor, actuator, function, and/or user that receives tasks from the Driver. Since users can take part in the automation process they can be considered workers. For instance, in the harvesting phase the Driver will notify the astronaut (if the device is set in on-device varied manual interaction or API varied manual interaction mode) that crop is ready for harvest. The user will then harvest the crop and set the device to transition to the next phase
Job Queue(s)		An ordered volatile list of tasks that the AMCFD must complete based on current system state

Purpose

Of SRS Document

The purpose of the System Requirement Specification (SRS) document is to describe the main functionalities to be implemented by the NASA KSC Autonomous Farming Device. The functionalities will each have related design constraints and performance requirements. The intended audience for this document is Dr. Philip Chan, the client and sponsor of this project, and project managers at NASA. The goals of this document can be defined as aiding in facilitating reviews, describing the scope of the work, providing a reference, providing a framework for testing, including features, and providing a platform for ongoing refinement.

Of Project

In preparing long lasting missions in space, one of the most important factors for the well being of an astronaut is ensuring they are healthy and well fed. Given the cost of launching materials and the busy schedules of crew members, the optimal way of producing food to meet nutritional needs is through an automated farming system.

This system must complete multiple autonomous cycles from seed to harvest, then cleanup. To ensure astronauts' longevity and health, it must be highly reliable and use few resources. Given this will be initially designed to be aboard the International Space Station, it must fit within one of the standard experimental lockers aboard the ISS; the ultimate goal is to provide a renewable food source for astronauts.[2]

Scope

The software of the AMCFD is a set of logic that dictates the behaviour of various subsystems. This set of logic will operate the AMCFD in full autonomy unless acted upon by other actors within the operational domain. The entire software package for this project is called SS and the subsystem declaration is in reference to the package's place in the entirety of the AMCFD. We consider the Driver to be the highest point in the software hierarchy and this is where all decisions will be made or delegated. The Driver takes input from the Task Manager and sensors, delegates or controls other subsystems based on the input, and creates job queues that it passes to the Task Manager. The Task Manager will operate under the Driver by accepting job queues, ordering or reordering the jobs with offsets, then passing these jobs back to the Driver to delegate tasks to the appropriate workers which enact each defined phase.

References

IEEE Recommended Practice for Software Requirements Specifications. Reference can be found here:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574&tag=1

[2] Referenced Project Plan v1.0

Overall Description

Product Perspective

The software needs can be categorized into sensor reading, actuator controls, timers, scheduling, interfaces, operating system components, lighting, application interface and various resource management metrics and operations. The AMCFD will have four distinct methods of control where these methods can be named as follows; fully autonomous, on-device varied manual interaction, ground varied manual interaction, and portable API varied manual interaction. The four methods of control are subsystems of the SS which is, by definition, a subsystem of the AMCFD. The SS will provide command, interpret and control capabilities for the systems of the AMCFD. The automation system can be defined in terms of PAS and RAS.

Design constraints

- One single microcontroller for system and subsystem operations

User Characteristics

Astronauts will be instructed on system use from ground control. Per the conversation with Senior Project Manager at NASA astronauts are informed on many things but experts in none of it. We must keep UX/UI in mind for all user interfaces.

Constraints, Assumptions and Dependencies

These are assumptions from requirements derived by MEE.

- System shall operate in a NASA express rack
- System shall be open
- Ease of disconnect:
    - A. Electrical connectors which are mated/demated during nominal operations shall require no more than two turns to disconnect. B. Electrical connectors which are mated/demated during ORU replacement operations only, shall require no more than six turns to disconnect.
        - For software this means we must anticipate disconnects on certain hardware

- LIGHTING DESIGN pg 3-118:
  - C. Light sources shall be dimmable.
  - D. Lighting in gloveboxes, excluding spot illumination, shall not exceed a brightness ratio of 3:1.
- Video: The standard video interface to the ISPRs within the U.S. Lab and Columbus is comprised of fiber optic lines using an EIA-RS-170A optical pulse frequency modulated video signal. Fibers are used to support video to and from the ISPR payload. A synchronization and control signal is also provided to the ISPR in accordance with EIA-RS-170A. The video distribution to the JEM is via twisted shielded wire pairs. A video card can be used inside the payload racks to convert the optical video/sync signals to electrical baseband NTSC EIA-RS-170A video/sync. The video signals from the ISPRs are sent to switches that allow distribution to onboard monitors, one of:
  - http://www.spaceref.com/iss/ops/ISS.User.Guide.R2.pdf
- Locker interface will be via ethernet

Functional Requirements

This section describes the functionality of the system service and these requirements depend on the software expected uses. Pheripherials may be listed as starting or end points for the requirements. As is the agile process these requirements may change as use cases evolve to meet the rigorous standards provided by NASA.

FRs:

1. Device Interface:
   - 1.1. User Interface:
     - 1.1.1. User on device(input/output):
       - 1.1.1.1. Shall allow the user to control specified components of the system on the device via buttons
       - 1.1.1.2. Shall allow user to view the status of system
     - 1.1.2. User API (input/output):
       - 1.1.2.1. Shall allow the user to control specified components of the system via software application
       - 1.1.2.2. Allow user to view reports on system
     - 1.1.3. Configuration file for crop type (input):

1.1.3.1. Configuration file shall have defined syntax like a scripting or markup language to organize editable variables and properties of the system

1.1.3.2. Phase 0 - Preseeding environment/Dormant/Off state:

1.1.3.2.1. Shall allow user to set time for start of cycle

1.1.3.2.2. Shall allow user to specify crop type for predefined environmental variables and system phase states

1.1.3.2.3. System shall transition to the next phase at the start of cycle:

Stop: Device/system started

1.1.3.3. Phase 1 - Seeding:

Start: Device has started.

1.1.3.3.1. Shall allow user to set nutrient mixing ratio

1.1.3.3.2. Shall allow obtain, transport, and plant actuator mechanics to be changed based on planting density, crop type, or seed size

1.1.3.3.3. Shall allow user to set sensor thresholds

1.1.3.3.4. Shall allow user to set observation reports metrics

1.1.3.3.5. System shall transition to next phase at germination:

Stop: Germination detected

1.1.3.4. Phase 2 - Grow:

Start: Germination detected

1.1.3.4.1. Shall allow user to set observation/reaction thresholds for various sensors and actuators

1.1.3.4.2. Shall allow user to set crop harvest time

1.1.3.4.3. System shall transition to the next phase when crop is ready for harvest:

Stop: Crop is grown to height or set time has passed

1.1.3.5. Phase 3 - Harvest:

Tolerance variables are the range of variation permitted in maintaining a specified action in each subsystem of the phases

Start: Crop is ready for harvest

1.1.3.5.1. Shall allow user to modify cutting tolerance variables

1.1.3.5.2. Shall allow user to set gather tolerance variables

1.1.3.5.3. Shall allow user to set separate tolerance variables
1.1.3.5.4. Shall allow user to set store tolerance variables
1.1.3.5.5. System shall transition to the next phase when crop is stored:
Stop: Crop has been harvested

1.1.3.6. Phase 4 - Clean
Start: Crop has been harvested
1.1.3.6.1. Shall allow user to set particular environmental variables that match a preseeding environment to crop type
1.1.3.6.2. Shall transition to the 0th phase when device state matches the set environmental variables
1.1.3.6.3. Shall allow user to set device to dormant/off state when n cycles have completed
1.1.3.6.4. System shall transition to next phase when:
Stop 1: All applicable variables match environment needed to start phase 1, cycle
Stop 2: n cycles completed machine enters dormant/off state

1.2. The system shall alert the user when any input is not considered valid

2. Decision Maker (Driver):
2.1. Shall take input from configuration file for system operations
2.2. Shall be the main head process running responsible for creating jobs based off of system input
2.3. Shall run all phases in each cycle based on configuration file input

3. Interfaces:
3.1. Sensors:
3.1.1. Shall read from multiple sensors
3.1.2. Shall be enabled and disabled over a period of time
3.1.3. Shall be read by decision maker
3.1.4. Shall be ignored by decision maker
3.1.5. Shall have reaction thresholds set by decision maker and/or user
3.2. Actuators:
3.2.1. Shall run multiple actuators
3.2.2. Shall be able to turn off by decision maker

3.2.3. Shall be able to turn on by decision maker
3.2.4. Shall be able to vary rotational power by decision maker
3.3. Locker:
3.3.1. Shall provide data for system interfaced
3.3.2. Shall accept data from system interfaced
3.3.3. Shall be interoperable with major locker systems
4. Scheduling:
(Task Manager)
4.1. Shall provide information regarding stage and state of system to decision maker and/or user
4.2. Shall provide API for schedule configuration for user
4.3. Shall signal transition based on sensor data and state
4.4. Shall initialize job queue based on sensor data and state
4.5. Shall configure job queue based on sensor data and state
5. Database
5.1. Shall keep low profile data from operations for validity checks and data integrity

Nonfunctional Requirements

These NFRs explain the system properties and constraints. Meeting NASA requirements may cause changes in the NFRs.

Metrics/MOE:

Since NFRs explain system properties and constraints we must define these constraints along with their MOE (margin of error). MOE TBD.

NFRs:

1. Power Consumption:
1.1. System shall operate under 700W of draw power
1.2. System operate in low power mode(s) where applicable
TBD by BIO assuming low power during grow phase with minimal actuator and sensor checks.
2. Size:
2.1. System shall be the size required to fit in specified locker

3. Flexible inheritance (interoperability):
    3.1. System shall be able to exchange data with various major locker systems
4. Data Integrity:
5. System shall maintain information on system
    5.1. System shall keep reliable information on system
    5.2. System shall keep complete information on system
    5.3. System shall keep attributable information on system
    5.4. System shall keep contemporaneous information on system
6. Reliability:
    6.1. System shall be reliable having an uptime TBD by each phase

Validation Checks

Requirement reviews, prototyping, test case generation

Validity Check

To ensure operations of each cycle the interoperability and user input functions must be checked.

- Numerical values must have type and casting checks when being passed from one subsystem to another and/or when the number is passed to an interface for use outside the immediate system.
- Regular expressions will be employed to validate that characters and strings do not violate any design constraints.

Consistency Check

Primary and secondary data must remain consistent and will be checked based on comparison methods to be created later in the SDLC when more data is available.
The goal for consistency check will be to ensure that the current system is not behaving outside an accepted standard deviation based on previous or expected cycle patterns.

Verifiability/Testability Check

Implement tests to verify code is correct using automated testing tools like sonar qube.

1. Detecting the presence of a bug through regression and unit tests on software

2. Tracking down the root cause of the bug
3. Fixing the bug

Traceability

All requirements are derived from stakeholder needs and can be traced to business use cases which are derived from the initial market need shown in the three powerpoint slides given to the entire team by the professors and NASA senior project manager.

Work Breakdown Structure

In home directory or listed on website.