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

Project Plan v1.0
Created: 8/28/2019
Revised: 9/06/19

Client Meeting Log

Date	Reason
August 27, 2019	Initial Meeting
Starting Sep 16th, 2019	Meeting every two weeks

Project Plan Version Control:

Revised Version	Date Revised	Collaborators	Version	Reason
N/A	9/06/2019	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 Definitions:

Word/Phrase	Re-worded/*Notes*:	Definition
Autonomous Multi-cycle Farming Device	The AMCFD	The subject of this project and the product being created
Control(s)	*Not re-worded & definition may not always apply to software*	The process by which software dictates the behaviour and movement of certain devices or software triggers.

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

Software Abstraction	*Not re-worded*	A high-level description of implementation instead of giving detailed code. The software abstraction column is given to show where computer science engages with project features
Embedded, microcontroller, single board computer	*May be used interchangeably & not re-worded*	The computer chosen to be the brains of the AMCFD. It will manage the various components of the AMCFD

Motivation and Goals

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.

The computer science goals on this project will be to provide the requirements and software needed to automate various parts of the AMCFD. These software needs can be categorized into sensor reading, actuator controls, timers, scheduling, interfaces, operating system components, and various resource management i.e. mitigating power drawn and pushed/pulled data.

To understand the goals of this project we need to define the needs listed above.

- **Sensor reading:** Data from sensors that will be attached to the AMCFD to be used by the the AMCFD to make decisions at each phase of the cycle.
- **Actuator controls:** The software mechanism to move and command various motors "actuators" to complete certain tasks.

- **Timers:** Timing is extremely important in farming and especially important in space farming; all the sensors and actuators must read and perform at the right time in order for the automation process to cycle properly.
- **Scheduling:** Highly productive people need schedules to initiate and complete their tasks, it is the same with a computer. In order to make sure everything cycles correctly and completely the AMCFD must follow a schedule which will be determined by software. Scheduling will also be a component of the operating system on the chosen microcontroller.
- **Interfaces:** The AMCFD will be fixed in an experiment locker on the ISS that will require ISS and AMCFD programs to talk to one another.
- **Operating system components:** Timers and scheduling will also take part in the operating system of the chosen embedded board.
- **Resource Management:** Resources in space are limited and extremely valuable. The AMCFD will be fed power from within the experiment locker along with push and pull requests[2] from the interfaces that need to be metered and controlled.

Approach

To ensure the wellbeing of the astronauts the system will use single board computers and/or microcontrollers. Sensors will be powered on at appropriate intervals to conserve power. Once the sensors detect the farm is ready to move on to the next stage, various physical systems will be engaged to transition. These stages are described below and each action of the AMCFD has a corresponding software abstraction.

Seed:

Action	Software Abstraction
Planting seeds	Scheduling and timing actuators to plant the seeds while reading in data from the AMCFD's internal environment
Detecting germination	Reading in sensor data to detect that the plants germinate properly

[2] Push and pull requests are basic elements of two-way communication between machines. If machine A has data to send it will push it to machine B and if machine B needs data it can pull it from machine A.

Grow:

Action	Software Abstraction
Providing a nourishing environment by sensing environmental factors	Using sensors to detect environmental compositions like H_2O and O_2 and to trigger appropriate system responses via controls using reactive autonomy
Tracking plant growth phases over time	Using sensors to detect changes in plant growth over time and cataloging those changes in a database to make any adjustments to the actuators if needed using proactive autonomy
Detecting diseased or under-grown plants	Using artificial intelligence[3] to monitor for potential diseased plants based on visual data

Harvest:

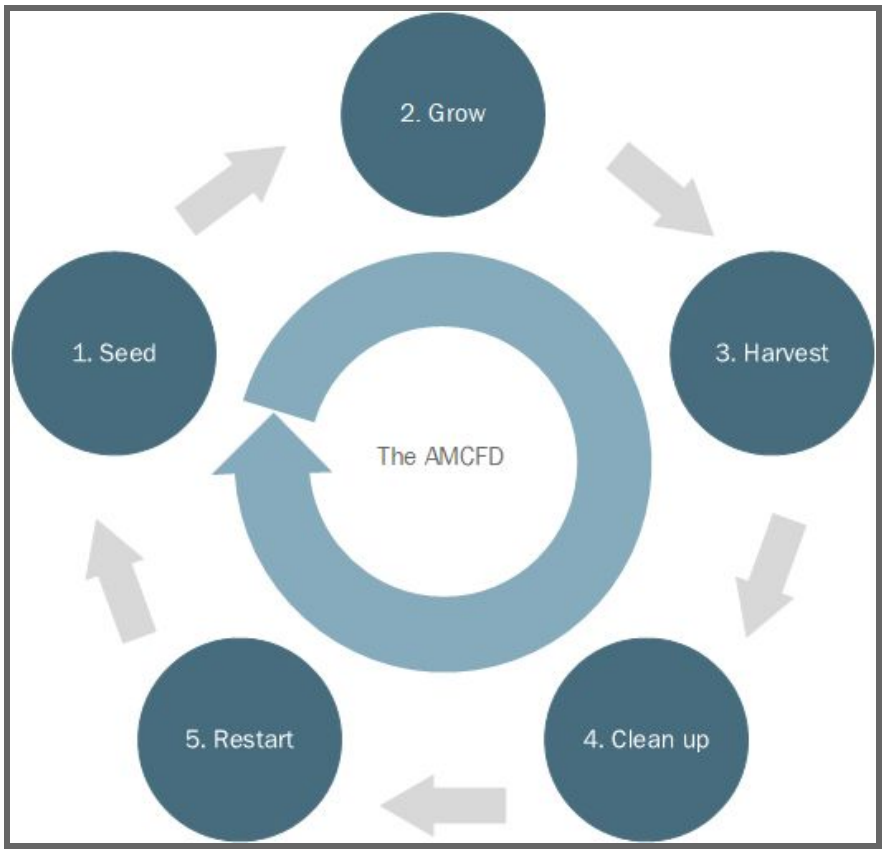
Action	Software Abstraction
Remove the grown plants for consumption	Control actuators to remove the mature plants
Divide harvest for consumption and fertilization	Control actuators to divide the harvest according to a specified ratio

Clean up:

Action	Software Abstraction
Remove dirt and other particulates	Control actuators to move unused particulates to a safe disposable location for recycling
Prepare for the next cycle	Reorder all actuators and queues for the next cycle

Restart ↻

[3] Using AI to analyze visual data is an enormous project on its own and it may not be completed with the first POC.



Novel Features

1. Low-power autonomous multiple-cycle growth and harvesting of food for astronauts driven by software.
2. Utilizing the latest in sensor, actuator, and software technology to protect and optimize throughput of said food.
3. Detecting diseased or under-grown plants using vision-based data.

Technical Challenges

- Programming microcontrollers, possibly in conjunction with and/or parallel to other single board computers.
- Scheduling of sensors and mechanical components to reduce energy consumption and maximise throughput.
- Communicating between sensors and mechanical components.
- Processing and abstracting sensor data.
- Creating a custom embedded linux OS with yocto[4] for power efficiency.
- New algorithms no one on the team has used before.
- Artificial Intelligence.

First Milestone

- Decide on the development OS, Linux/Windows
    - OS may depend on the hardware chosen
- Install and configure Visual Studio
    - This is based on the IDE commonly used with embedded boards
- Compare and select collaboration tools
- Provide demos for selected tools
- Create requirements document with client
    - Collaborate on Cameo Systems Modeler to map requirements derivation, functional requirements, non-functional requirements, & stakeholder requirements
- Create design documents
- Create test plan and matrices
- Define change management process using agile
- Work Breakdown Structure
    - Elements for the WBS are derived from the Requirements Derivation
- Have interdisciplinary meeting with all stakeholders on project.

[4] The Yocto Project is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

Task Matrix for Milestone 1

Task	Giampiero	Christopher
Decide on development OS	Informed[5]	Responsible[6]
Install and configure VS	Responsible	Responsible
Compare and select collaboration tools	Responsible	Informed
Provide demos of tools	Responsible - 50%	Responsible - 50%
Create requirements document	Responsible - 50%	Responsible - 50%
Create design document	Responsible - 50%	Responsible - 50%
Create test plan	Responsible - 50%	Responsible - 50%
Define change management process	Responsible - 25%	Responsible - 75%
WBS	Responsible - 50%	Responsible - 50%

Second Milestone

- Create state diagram and design for the system
- Implement inheritable high-level interfaces for sensor communication
- Create simulated farm for testing
- Create baselining requirements and process

Third Milestone

- Decide which set of computers/microcontrollers will be used to control the system
- Implement C/C++ libraries for serial communication
- Plan the design for a custom OS (Implement next semester when hardware is decided?)
- Create development toolchain (maybe do this next semester when arch is decided?)

[5] Project member is informed but not yet proficient in said task
[6] Project member is proficient and responsible for said task

- Start a preliminary product breakdown structure

Signature of CSE Students:

Signature: _____ Date: _____

Signature: _____ Date: _____

Approval from Faculty Sponsor:

"I have discussed with the team and approved this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____