Milestone III Progress Evaluation

Project Title: Autonomous Multi-Cycle Farming in Space

From CSE [coordinator and project sponsor: Philip Chan, pkc@cs.fit.edu]:

Christopher Millsap <cmillsap2013@my.fit.edu>
Giampiero Corsbie <gcorsbie2018@my.fit.edu>

From BIO [coordinator: Drew Palmer, apalmer@fit.edu]:

David Masaitis <dmasaitis2016@my.fit.edu>

From MEE [coordinator: Elisabeth Kames <ekames2011@my.fit.edu>]:

Dominic Allard, dallard2016@my.fit.edu
Courtney Cline, ccline2018@my.fit.edu
Joseph Luya, jluya2016@my.fit.edu
Bryce Johnson, bjohnson2016@my.fit.edu,
Philip Bernhard, pbernhard2015@my.fit.edu,
Timothy Frazier, tfrazier2016@my.fit.edu,
Joshua Calhoun, calhounj2008@my.fit.edu,
Kali Jenson- AEE student, kjenson2015@my.fit.edu,
Chris Mateo- ECE student, cmateo2016@my.fit.edu

Clients:

Some potential clients: NASA, SpaceX, Blue Origin, Virgin Galactic.
Current client: Dr. Philip Chan Associate Professor, Computer Engineering and Sciences at Florida Institute of Technology, coordinator and project sponsor.

Progress Matrix

Task Matrix for Milestone 3

Task	Giampiero	Christopher	To Do
Decide which set of boards will be used to control the system other than the microcontroller	Responsible - 50%	Responsible - 50%	Determine with MEE
Implement Python libraries for serial communication	Responsible - 50%	Responsible - 50%	Actual libraries to be used and not just libraries for demos
Finalize development toolchain	Responsible - 50%	Responsible - 50%	
Start a preliminary product breakdown structure with MEE	Responsible - 50%	Responsible - 50%	Derived from BOM
At least one sensor & actuator functioning according to some schedule ○ These will run based on simulated data along with simulated cycles	Responsible - 50%	Responsible - 50%	Water pump and real world timed tests (2-3 weeks +)

Task	Giampiero	Christopher	To Do
Full simulated farm with well-defined input parameters	Responsible - 50%	Responsible - 50%	Procedures need to be fixed and not use units in config file
Configuration file for all phases in system to run in simulation	Responsible - 50%	Responsible - 50%	

Testing Document inheriting from test plan from milestone 1:	Responsible - 50%	Responsible - 50%	**Testing:** **Static** - Done by reading code through formal reviews **Functional** - Verifies that each function of the software application operates in conformance with the requirement specification **Structural -** Whitebox testing, testing trigger conditions, etc **Performance -** Responsiveness and stability under a particular workload **Testing scripts -** Automated testing

	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

Discussion

Task 1 - Decide which set of computers/microcontrollers will be used to control the system other than the microcontroller:

We will need other smaller boards/hats to control peripherals and actuators on the system; this is most likely probable with the creation or integration of a robotic seeding mechanism and integrating canopy lighting. Currently the MEE team is looking to use a custom made mechanical apparatus with four actuators for moving in 3D space and seeding, for this we would need hats particular to that subsystem. For lighting we will need to monitor power consumption but not provide it through the board along with switching the states/frequency. We found several boards that meet the requirements for the hardware chosen by MEE. Currently there are several SEMA stepper motors that need drivers, a linear actuator needing a control board, and a fan driven by the same controller as the SEMA motors. The found items are listed in the BOM file.

Task 2 - Implement Python libraries for serial communication

Controlling and synching subsystems will require more python libraries on the microcontroller. One of the system requirements states that ethernet will be the locker to system interface so we already know that this architecture may need serial libraries. In many cases, serial is cheaper to implement than parallel. Many ICs have serial interfaces, as opposed to parallel ones, so that they have fewer pins and are therefore less expensive. The lux sensor, ULN2003 5V Stepper Motors, and ULN2003 Driver Boards used in the demo have libraries specific to their use and have been implemented as examples.

Task 3 - Create development toolchain

Good DevOps practices dictate that we have our tools used for the final product listed and linked. There is no one tool that will or has provided everything we need to develop the AMCFD. For this project we will have the support and library defined along with linking together programs used in dev by specific stages. During this milestone we will define the toolchain building process that best match the AMCFD. We have considered porting the design to another language that would use a different toolset, as of now we have a defined toolchain for what was used.

Task 4 - Start a preliminary product breakdown structure with MEE

A product breakdown structure is an effective tool that details the physical components of a particular product, or system, under consideration. The formal PBS comes in the form of a hierarchy the begins with the final product at the top of the hierarchy followed by the sub-categorized elements of the product. The product breakdown structure is similar to the work breakdown structure. Like the work breakdown structure, a product breakdown structure serves to reduce a complex project into manageable components. As a result, teams can obtain a clear understanding of a product, its components, and what is required to provide those components. This will make the project much more portable when the system is taken by a larger company. We have a preliminary PBS through our BOM file which is broken down by subsystem and assembly.

Task 5 - At least one sensor & actuator functioning according to some schedule

The full system demo shows an entire four phase cycle being run three times. This represents about 9 months of operation. Over the break we will be conducting tests on the system demo hardware to evaluate how it responds running in real-life-time; multiple week analogs. We will be adding a water pump and setting the config file to maintain wasabi plants in a 3D printed tray.

Issues:
One minor issue we encountered was with the 5v pin on the BBB. We could not understand why the stepper motors were not getting power until we read the BBB manual in depth and found that the Sys_5v pin was the only way to supply 5v when powered via USB.

Task 6 - Full simulated farm with well-defined input parameters

We have well defined & expected real-life settings to run a simulation of our system through all phases including demonstrating error handling. The simulated farm has scaled values to demonstrate in class and we will run the simulation on the board with real expected values for plant growth and monitoring. This could take ten or more days to evaluate. See task 5 for more details.

Task 7 - Configuration file for all phases in system to run in simulation

The configuration file can be set for all phases in the requirements for the AMCFD.

Task 8 - Testing Document inheriting from test plan from milestone 1

The testing plan from milestone 1 has been expounded upon with the following:

- **Static** - Done by reading code through formal reviews
- **Functional** - Verifies that each function of the software application operates in conformance with the requirement specification
- **Structural -** Whitebox testing, testing trigger conditions, etc
- **Performance -** Responsiveness and stability under a particular workload
- **Testing scripts -** Automated testing with real-life scenarios

We will also use SonarQube to review, with static analysis, our code to detect bugs, code smells and security vulnerabilities. A sample excerpt has been provided on the slides shown during milestone 3 presentation.

Team Discussion

Notes are given based on interactions between CSE. They do not necessarily reflect the team member's entire contribution.

- Christopher Millsap <cmillsap2013@my.fit.edu>:
    - Designed the simulation farm and integrated it into the system for testing and presentation
    - Added fixes for error outputs of config file to be more useful
    - Integrated sensors into classes
    - Added full phase multi-cycle to config file
    - Integrated actuators into system
    - Built system hardware demo
    - Added seeding trays to system

    - Class dependency evaluation
    - Added fix to allow the config file to call procedure, i.e. “seeding”
    - Testing in SonarQube and added to test document outline for testing procedures to be implemented
- Giampiero Corsbie <gcorsbie2018@my.fit.edu>:
    - Provided python expertise
    - Sensor read delay mitigation
    - Internal command enumerations
    - Multiphase of system
    - Fix to allow transitions to be inserted into schedule
    - Config value validation
    - Class dependency evaluation
    - Branched actuator class into different base classes for different behavior

Plans for Next Milestone

Task	Giampiero	Christopher
System Interrupts and Error Handling	50%	50%
Allow for Manual Control	50%	50%
System Debug Suite	50%	50%
Config Section for System Settings	50%	50%

Task 1 - System Interrupts and Error Handling

The system will be able to catch and handle errors or interrupts and then, based on the case, may resolve the error, resume execution, terminate execution, or poll for user input. This feature will provide the foundation for the next task.

Task 2 - Manual Control

This feature will allow users to interact with the system. The ability to pause, halt, resume, or reset the system will be included.

Task 3 - System Debug Suite

This feature will be invaluable to both software and mechanical team members for testing and dynamic analysis. It will allow the user to step through system instructions and observe the current state for the actuators, sensors, scheduler and driver. It will also provide details such as the current phase, time elapsed. Additionally, the ability to modify the config file during execution may also be implemented.

Task 4 - Config Section for System Settings

A separate section of the configuration file will be added to specify system settings that remain constant between phases. Some settings that will be included are tray types for each slot, hardware dimensions, number of cycles to execute, and default actuator/sensor parameters.

Feedback:

1. Astronaut GUI
    a. Change config file
    b. Manual control
    c. Select config files
    d. System state
    e. System errors
2. Procedures
    a. For non-trivial tasks like seeding with multiple actuators

Date(s) of meeting(s) with Client during the current milestone 2:

Every two weeks starting 9/13/19

Project Logs			
Action	Date	Time spent in hours	Comments
Met with Dr Chan about config errors, resource sharing, timing of reading sensors accumulating, allow config file to call procedures, seeding for trays	11/8/19	1	Full phase multi-cycle of config
Met with Dr Chan about Milestone 3	11/22/19	1	Tray types as parameters, mode without unit params,

			diagram of virtual sensors and real devices, display of config in demo, small time delays
Milestone 3 review/evaluation	11/25/19	1	

"See Faculty Sponsor Feedback below"

Faculty Sponsor feedback on each task for the current Milestone:

Task 1 - Decide which set of computers/microcontrollers will be used to control the system other than the microcontroller:

Task 2 - Implement Python libraries for serial communication

Task 3 - Create development toolchain (maybe do this next semester when arch is decided?):

Task 4 - Start a preliminary product breakdown structure with MEE

Task 5 - At least one sensor & actuator functioning according to some schedule

--

Reduce threads where possible and minimize interconnection (sizes of relationships) if issues found.
Add sensor to into classes
Add actuator into classes
Diagram or representation of virtual devices and real hardware on system demo
Display config in demo instead of just logger output
Consider small time delays over time and how they accumulate
Sensor value variation to demo transition in config file
Study issues:

- Deadlock
    - Cyclic dependency locking steps
- Race
- Starvation

And the common solutions.

Full phase multi-cycle of config in Sim

--

Task 6 - Full simulated farm with well-defined input parameters

--
Mode in parameters without units
Transition without monitoring
--

Task 7 - Configuration file for all phases in system to run in simulation

--
Config file errors made more useful to user.
Add seeding trays
Trays as parameters
Allow config to call procedures
--

Task 8 - Testing Document inheriting from test plan from milestone 1

Faculty Sponsor Signature: ________________________________ Date: ________

Faculty Sponsor Evaluation:

1. Faculty Sponsor: detach and return this page to Dr. Chan (HC 322)
2. Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Christopher Millsap	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Giampiero Corsbie	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Sponsor Signature: ________________________________ Date: __________