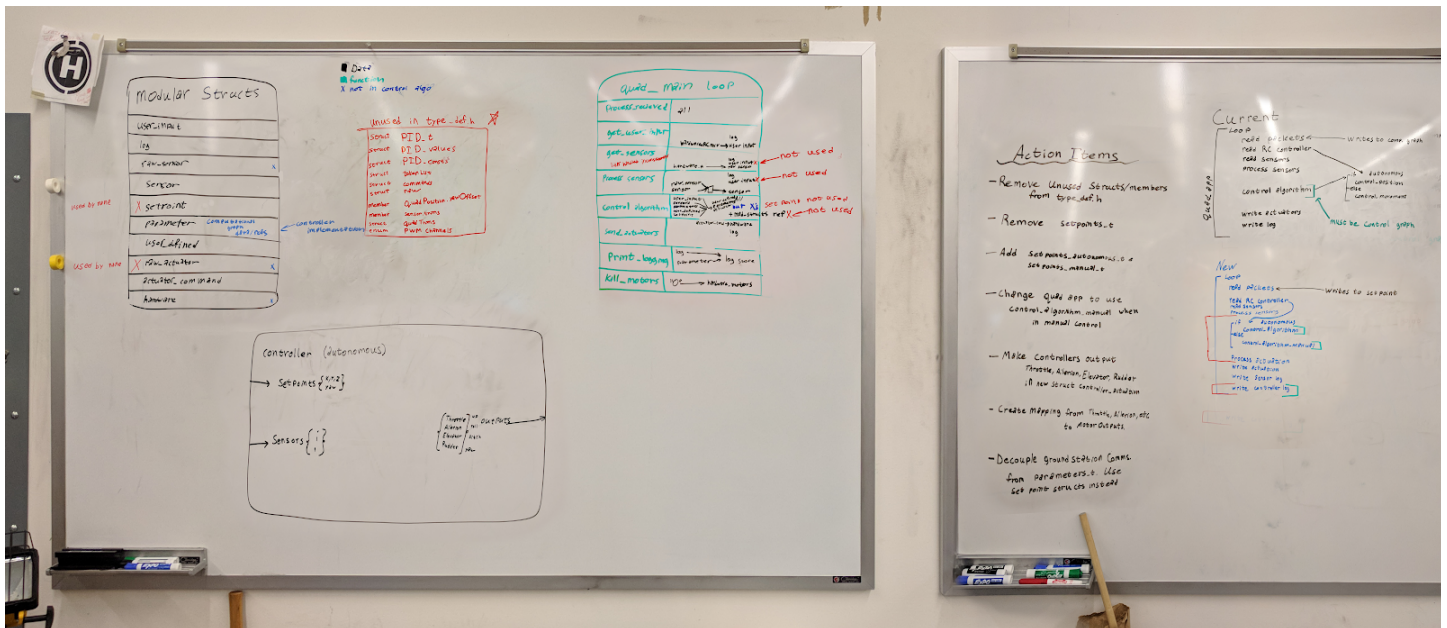# MicroCart (Microprocessor Controlled Aerial Robotics Team)
sdmay19-20

February 4 — February 10
Client/Advisor: Dr. Phillip Jones

Team Members
James Talbert — *Hardware*
Sarah Koch — *Controls*
Anthony Bertucci — *Ground Station*

Summary of Progress this Report

- **James**
  - Worked with Sarah to diagram the existing code structure
    - Identified several unused/leftover structures
    - Identified structural issues to be dealt with before we can hand the system off to users
    - Designed solutions to the handoff issues.
      - Reduce the amount of code the user must maintain
      - Reduce the responsibility of the control algorithm
- **Sarah**
  - Worked with James to diagram the existing code structure.
  - Determined that the control algorithm changes based on if the quad is operating in autonomous or manual mode

- For now, focus is being placed on developing the control algorithm assuming that it will be used for autonomous navigation
      - ○ Worked on possible structural changes to the control algorithm code that would separate it from other functions, like data logging, and make it simple to swap out controllers
      - ○ Met with Dr. Jones to to resolve the identified issues
- **Tony**
    - ○ Met to understand internal quadcopter software structure.
    - ○ Attempted to implement code for grabbing data from backend, but was somewhat hampered from testing (by issues explained in Pending Issues section)

Pending Issues

- **James**
    - ○ The planned software changes affect several subsystems, whose behavior should remain mostly unchanged, but need to be decoupled from each other. More planning is necessary before several of the changes can be made.
- **Sarah**
    - ○ Same issues as mentioned by James
    - ○ If desired, it may be necessary to find a way to implement the control algorithm so that it can function in both manual and autonomous modes
- **Tony**
    - ○ The groundstating development computer is current non-functional due to failed updates.
    - ○ ETG is resolving the issue, but groundstation development is on hold while it is fixed.
    - ○ This issue has prevented any testing of software requiring a full build of the system (i.e. anything that is not a simplistic GUI change).

Plans for Upcoming Reporting Period
- **James** (which of these occur depends on meetings early next week)
    - ○ Merge branches together in git to reduce the number of branches or at least make them more consistent.
    - ○ Continue design of software relating to user handoff for controls research users.
    - ○ Begin implementation of changes relating to user handoff for controls research users.
    - ○ Begin creating a user-handoff package that includes the software interfaces that research users will use to implement the control algorithms.
- **Sarah**
    - ○ Begin creating a user-handoff package that includes mathematical models of the quad and documentation for using the models to generate controllers.
- **Tony**
    - ○ Merge branches together in git to reduce the number of branches or at least make them more consistent. Specifically, this involves the merging of Tina's quad_app changes for real time and my current branch

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
James Talbert	- Code diagramming - Idea Generation - PCB final assembly and re-testing	10	116
Sarah Koch	- Code diagramming - Defining structure of control algorithm code so that it is swappable - Separating control algorithm code from other quad functions	8	96
Anthony Bertucci	- Discussion of reorganization of main quad software control loop. - Worked on implementing real-time data logging tool in backend (ran into issues explained above in Pending Issues section)	6	88

Gitlab Activity Summary

Action: commented on, Sun Feb 10 2019
Author: James Talbert
Title: Refactor Control Algorithm Interface/Research User Handoff, Type: Note
Comment: It appears that the documentation in the the existing `commands.c` file is incorrect. The groundstation uses a different set of callbacks than the quad. Personally, I think this makes more sense as the ground station callbacks will have to interface to a different environment.

Action: commented on, Sun Feb 10 2019
Author: James Talbert
Title: Refactor Control Algorithm Interface/Research User Handoff, Type: Note
Comment: I propose that we have the user provide 1 file for basic operation (no specialized commands) named `controller.c`. We will provide a static `controller.h` file:

```
/**  controller.h  **/

// All controller data goes here
typedef struct controller controller_t;

// Initialize the controller (run once on startup)
// controller should be assumed uninitialized when called
void control_init(controller_t* controller);

// Control algorithm run, equivalent to the controller block in the Simulink model
void control_algorithm(controller_t* controller, setpoint_t* setpoints, sensor_t*
sensors, actuator_t* output);
```

Where the `controller_t` item is an opaque pointer to the MicroCART/Quad systems. The user may choose to leave the definition out from this header. `control_init` and `control_algorithm` must be defined in `controller.c`

If the user requires specialized commands, then we will need:
 - A command definition file (`controller_commands.h`)
 - Command callbacks for the groundstation
 - Command callbacks for hardware platform

At the moment, I am unsure of how the groundstation callbacks will work, they will presumably need to export some sort of user interface to the groundstation GUI.

Action: opened, Sun Feb 10 2019

Author: James Talbert

Title: Refactor Control Algorithm Interface/Research User Handoff, Type: Issue

----

Action: commented on, Sun Feb 10 2019

Author: James Talbert

Title: Transition current hardware platform to Vivado, Type: Note

Comment: The autonomous flight eventually worked. I am going to wait and merge this as part of the PCB transition as there are several hardware changes involved with that. This should be mergable again in a couple weeks or less.

Action: commented on, Sun Feb 10 2019

Author: James Talbert

Title: Clean up Sensor Data structs, Type: Note

Comment: Additionally, there are additional unused structs in type_def.h

----

Action: deleted, Sun Feb 10 2019

Author: James Talbert

ref: debug_levels

Action: deleted, Sun Feb 10 2019

Author: James Talbert

ref: PCB

Action: deleted, Sun Feb 10 2019

Author: James Talbert

ref: write_pwm_generate_hw_tests

Action: deleted, Sun Feb 10 2019

Author: James Talbert

ref: write_pwm_record_hw_tests

Action: deleted, Sun Feb 10 2019

Author: James Talbert

ref: write_i2c_hw_tests

Action: deleted, Sun Feb 10 2019
Author: James Talbert
ref: write_pwm_hw_tests