2 Project Plan

2.1 Project Management/Tracking Procedures

Our team plans to use a waterfall+agile management style. The nature of our project is such that we can immediately begin adding features to the CrazyFlie, allowing an agile approach to work well. We plan to track our progress in the project through a GitLab kanban board. Tasks/issues will be created on the board as we begin development, which will then be assigned to different members of the team. Additionally, milestones with due dates will be created to make sure we are keeping a good pace as the project progresses.

2.2 Task Decomposition

Tasks to complete:

1. Investigate CrazyFlie firmware architecture
    a. Learn how to modify and flash new firmware to the CrazyFlie
    b. What is the current architecture structure?
    c. Can the control code be easily modified?
    d. How easy is the control code to understand for new users?
2. Modify CrazyFlie firmware to be as modular as possible
    a. Abstract the control code to a standardized interface to allow other control algorithms to be easily implemented through an adapter architecture
    b. Rewrite the existing control code to utilize the new interface
3. Add wifi capabilities to the CrazyFlie
    a. Research current communication protocols in the CrazyFlie firmware
    b. Electrical connection between wifi chip and CrazyFlie
    c. Test communication over wifi
    d. Control drone over wifi with less than 20 ms of latency
4. Test different control algorithms on the CrazyFlie
    a. Write a basic PID control loop to maintain a stable hover, using the new interface
5. Develop ground station software to communicate with and control CrazyFlie
    a. Start with command line interface on linux
    b. Build a GUI/frontend once the backend is mostly working
6. Develop test stand hardware
    a. Determine what electronics will be used to record & communicate data
    b. Integrate chosen electronics into test stand for data collection
    c. Design and print test stand model to mount CrazyFlie
7. Develop test stand software to measure and log rotation of the CrazyFlie while held in test stand
    a. Should collect and record all desired data from the CrazyFlie in real time
    b. Should communicate with the ground station to allow for easy saving of log data

8. Write lab instructions and documentation for interfacing and using the modified CrazyFlie
    a. Basic quick start guide
    b. Detailed proposed lab activities
9. Stretch goal: Convert control algorithm from MatLab code to C code that works with our interface

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Project Milestones:

1. Working test stand prototype
    a. 3D printed and assembled
    b. Electronics selected and assembled
    c. Firmware written and reporting back
2. Custom firmware running on CrazyFlie
    a. First modified firmware running on CrazyFlie
    b. Control software abstracted with adapter interface
    c. Existing control code running through adapter
3. Custom ground station CLI
    a. Control commands can be sent to CrazyFlie and a acknowledgement is sent back
4. Ground station GUI based on CLI
    a. Basic GUI that sends pre configured commands through the CLI
    b. More advanced GUI that displays flight data and allows for gamepad controls

2.4 Project Timeline/Schedule

Gantt Chart attached as .xlsx file on canvas

2.5 Risks And Risk Management/Mitigation

Risks:

1. Firmware is much harder to adapt to an adapter architecture, takes longer than expected, 60%
    a. Mitigation plan: Begin researching and modifying the CrazyFlie firmware as soon as possible. May need to push back tasks that are dependent on this, the different control algorithms and lab instructions.
2. CLI cannot communicate with CrazyFlie, 30%
3. Wifi chip is not compatible, 50%
    a. Mitigation plan: Look for other wifi alternative chips that are proven compatible with the CrazyFlie. Else fall back to current radio communication protocols.
4. Wifi communication latency is greater than 30 ms, 70%
    a. Mitigation plan: review previous years code, as they have worked with wifi in the past and gotten the latency very low. speak with jones about this.

5. GUI takes too long to create, 40%

2.6 Personnel Effort Requirements

Task	Estimate (hours)	Explanation
Investigate CrazyFlie firmware architecture	21 ± 5	Depends on firmware complexity
Learn how to modify and flash new firmware to the CrazyFlie	4	- Read official documentation - Build stock version of firmware and upload to drone
What is the current architecture structure?	5	- Read official documentation - Read existing code - Map out function calls for normal operations
Can the control code be easily modified?	10	- Map out normal operation loops - Modify the control logic to be more general
How easy is the control code to understand for new users?	2	- Evaluate code readability to novice programers - Frequency and quality of comments?
Modify CrazyFlie firmware to be as modular as possible	35 ± 8	Depends on firmware complexity
Abstract the control code to a standardized interface to allow other control algorithms to be easily implemented through an adapter architecture	20	- Will likely require extensive modifications to firmware to allow for a modular, adapter architecture
Rewrite the existing control code to utilize the new interface	15	- Depending on the complexity of the existing control code, it could take a long time to work around checks put in place.
Add wifi capabilities to the CrazyFlie	36 ± 5	

Research current communication protocols in the CrazyFlie firmware	5	- Research the documentation and potential changes need to be made
Electrical connection between wifi chip and CrazyFlie	3	- Basic hardware connection between the wifi card and the CrazyFlie
Test communication over wifi	8	- requires minor firmware modifications
Control drone over wifi with less than 20 ms of latency	20	- The main development work for this task, requires major firmware modifications
Test different control algorithms on the CrazyFlie	15	
Write a basic PID control loop to maintain a stable hover, using the new interface	15	- Create and refine a control algorithm to control the CrazyFlie using the new interface
Develop ground station software to communicate with and control CrazyFlie	30	
Start with command line interface on linux	10	- Some basic commands, basic C program
Build a GUI/frontend once the backend is mostly working	20	- Dependent on CLI - Has to interface with controller
Develop test stand hardware	16	
Determine what electronics will be used to record & communicate data	3	- Evaluate desired data to be collected - Select sensors to be used to collect selected data types

Integrate chosen electronics into test stand for data collection	6	- Design a solution to integrate the sensors into the test stand design
Design and print test stand model to mount CrazyFlie	7	- Complete a design for the test stand - Work with the ETG to 3D print it
Develop test stand software to measure and log rotation of the CrazyFlie while held in test stand	20	
Should collect and record all desired data from the CrazyFlie in real time	8	- Create the firmware to manage sensors and record desired data from the drone
Should communicate with the ground station to allow for easy saving of log data	12	- Create the software to communicate the recorded data to the ground station
Write lab instructions and documentation for interfacing and using the modified CrazyFlie	12	
Basic quick start guide	3	- Create a basic start manual to allow users to operate the CrazyFlies
Detailed proposed lab activities	9	- Create mock lab activity documentation to make use of the CrazyFlies in an embedded systems lab
Stretch goal: Convert control algorithm from MatLab code to C code that works with our interface	TBD	

2.7 Other Resource Requirements.

- Access to several CrazyFlie drones to test and develop on
- Development computers running linux to build and test the system on
- ETG access for 3D printing components of the test stand