University of Arkansas – CSCE Department

Capstone I – Final Proposal – Fall 2020

Quadcopter Drone Controller

Team Members: Zachary Heil, Lily Phu, Stephanie Phillips, Spencer Ward, Andy McCoy, Joel Parker, Christ Somophounout, Dishoungh White II

Abstract

Drones are fun to build and control. The CC3D drone controller board is a cheap microcontroller board. However, it currently does not support the more modern features such as a barometer, GPS functionality, and Wi-Fi support. The project contains multiple components that will need to be implemented: the PCB (Printed Circuit Board), a firmware program that will assist the drone's flight capabilities with onboard sensors and other control systems logic, and a planned application that will allow for the drone to be controlled with a mobile device. This project will be a collaboration with the Electrical Engineering (ELEG) and Computer Science / Computer Engineering (CSCE) departments. The Electrical Engineering team will design the hardware components and layout for the board to implement these features. The CSCE team will be primarily responsible for designing the firmware and possibly a mobile application that will control the quadcopter via Wi-Fi. The application is also planned to display key information to the pilot. These features will be highlighted further in the Design portion of the report.

1.0 Problem

One of the issues that the Electrical Engineering team faces is deciding on the microprocessor to use for the board's design. The CC3D board currently supports flight stabilization, satellite receiver support, and S-Bus support with the outdated STM32F1 microcontroller. One of the major problems for the CSCE team is to adapting the existing open-source firmware for drone controllers into the CC3D board. Without the firmware adaptation, the drone would not be able to perform basic flight control functions such as hovering and power delivery management.

2.0 Objective

The objective of this project is to create a quadcopter flight controller based upon an existing obsolete flight controller model, the “CC3D” with similar or better functionality to commercially available flight controllers. The scope of this objective is focused around the flight controller such as features, drone behavior, and method of flight control. This flight controller must work with the DYS XCITE 320 quadcopter but should be compatible with almost any quadcopter design like most off-the-shelf flight controllers. And because of this, the quadcopter should remain largely unchanged as the modifications should be focused on the power delivery circuitry in the DYS 320 quadcopter as well as the flight controller PCB. The controller will accept PWM RC control signals and output servo control signals to control the quadcopter. The system will be stabilized to maintain a hover as well as supporting LED indicators, sound indicators, and a battery protection circuit. While this is the basis of the controller, L3Harris wants to add, if time permits, stretch goals such as the ability to accept PPM and S-Bus control signals. Further goals also include attaching a GPS module and extended functionality to be controlled by an app on a device. However, these goals may not be in the final design of the project due to time constraints.

3.0 Background

3.1 Key Concepts

CC3D Flight Control Board [1]: The CC3D Flight Control Board is the hardware that will run the firmware that will be chosen for the project. Originally run with OpenPilot, the flight controller has become obsolete, but it provides a good foundation for creating the quadcopter flight controller for the project. The “small volume, [tidy circuitry] and affordable price” made it a popular board and an ideal candidate for the flight controller.

Firmware: Firmware is a type of computer software that gives information and instructions for communicating between the device and hardware. The software is stored in the hardware and is interacted with by the device by sending signals most commonly by certain frequencies. Firmware is especially common for everyday hardware such as TV remotes, routers, refrigerators, and many other daily devices and appliances.

Betaflight [4]: Betaflight is the firmware that will most likely be stored into the controller of the quadcopter. Advertised as the best FPV firmware, the focus on flight performance as opposed to other firmware forks makes it a good pick for the firmware for the project’s flight controller.

System: The system refers to the circuitry that makes up the quadcopter controller. In the design, the system features will refer to the features of the final design overall.

PWM Signals [6, 7]: Pulse Width Modulation signals are signals that are transmitted with the average length of information between the transmitter and receiver. The width of the pulses vary which is why it is pulse width modulation. The amplitude is constant while the positions are changed based on the signal. This is the signal that the final design of the controller will use if stretch goals of PPM signals are not met.

PPM Signals [6, 8]: Pulse Position Modulation signals are signals composed of pulses of a fixed length in a series to transmit information between the transmitter and receiver. It is similar to Pulse Width Modulation signals however PPM changes the position of the impulse without variation to the amplitude. The final design may or may not have PPM signals integrated in since it is a stretch goal

3.2 Related Work

The quadcopter is obviously similar to the original CC3D board which ran from OpenPilot firmware. This project will use BetaFlight which is a fork of other projects and it improves on the flight performance compared to the old OpenPilot. The commercial flight controllers that are sold have their software closed source. The project aims to make it open source which is a large improvement allowing a foundation to be built for future teams that need a quadcopter for a project such as extension of certain functionalities.

Furthermore, the CC3D mentioned in [1] has one feature that the project improves on. The GPS module will be integrated into the project if time permits which will be an improvement on the current CC3D flight controller that only supports GPS and data transmission. The GPS module will allow tracking of the controller location at all times which can help with understanding the altitude and elevation of the quadcopter. The project also will have a gyroscope as well as barometer. These improvements will make it better than the commercial flight controllers that lack these capabilities.

4.0 Design

4.1 Requirements and Use Cases and Design Goals

4.1.1 Requirements

For this project, there are a few important functional requirements for the quadcopter flight controller. The controller must be a fully functional controller based on the CC3D flight controller model. The updated CC3D controller must decode remote control Pulse-Width Modulation signals to control the drone during flight. The drone must also output a minimum of 4 motor control signals that are, each, sent to the respective motor's Electronic Speed Controller (ESC) unit. The updated CC3D controller must have hardware support for communication protocols such as Wi-Fi and GPS. When the drone is hovering (maintaining a near-constant altitude), no other control signals such as throttle should be sent from a radio controller source using a barometric pressure sensor. The drone should also have a mechanism for battery protection to prevent permanent damage to the battery due to weather or any external phenomena.

Alongside the functional requirements, there are some non-functional requirements as well. The drone should include LEDs and audio output devices to aid the user while operating the drone by indicating location, if crashed, fallen, lost, low battery, and other things. Features implemented past this point would exceed the foundational scope of this project. Another feature that could be implemented is an external application to control the drone with a smartphone device instead of an RC controller. This application would allow the pilot to configure GPIO inputs and view information such as battery life, RSSI, and last known location. At this stage, final implementation may vary.

4.1.2 Use Cases

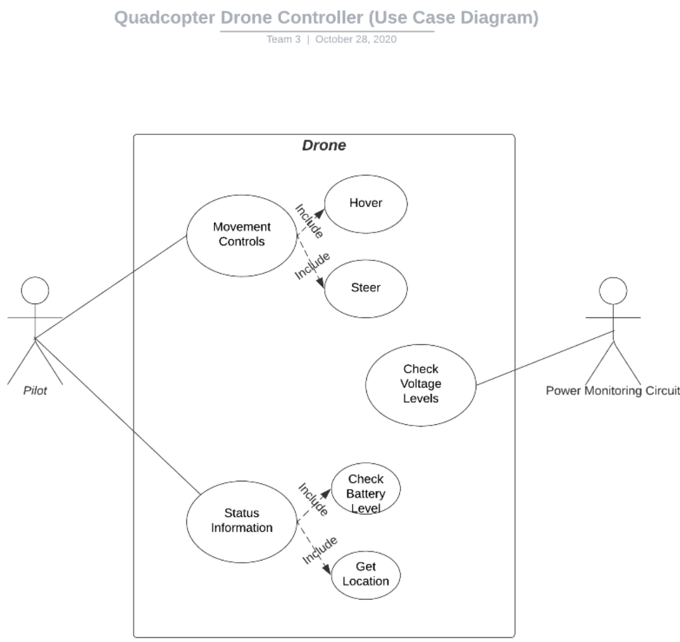

Figure 1: UML Use Case Diagram

In the UML Use Case Diagram, the diagram shows two "actors" (the pilot and the Power Monitoring Circuit) interacting with the "system" or the drone in this case. The pilot, either by means of RC controller or through Wi-Fi mobile app, controls the drone's movements such as hovering and steering. The pilot also checks status information about the drone such as the drone's battery levels, the location, and RSSI. Alongside the pilot as the main user, there will be a power distribution/monitoring circuit built into the controller to detect any under-voltages and check for any damaged parts in the drone's battery unit.

4.1.3 Design Goals

The "CC3D" flight controller model is currently outdated and lacks an open-source model. The current controller has basic flight stabilization, satellite receiver support, and S-Bus support. However, it lacks the more modern features that many controllers have such as simultaneous S-Bus and PPM support for more user-modifications, GPS support, Wi-Fi support, status alerts, and improved flight stabilization. Those are some of the features that are intended to be implemented in this improved version of the CC3D model.

4.2 High Level Architecture

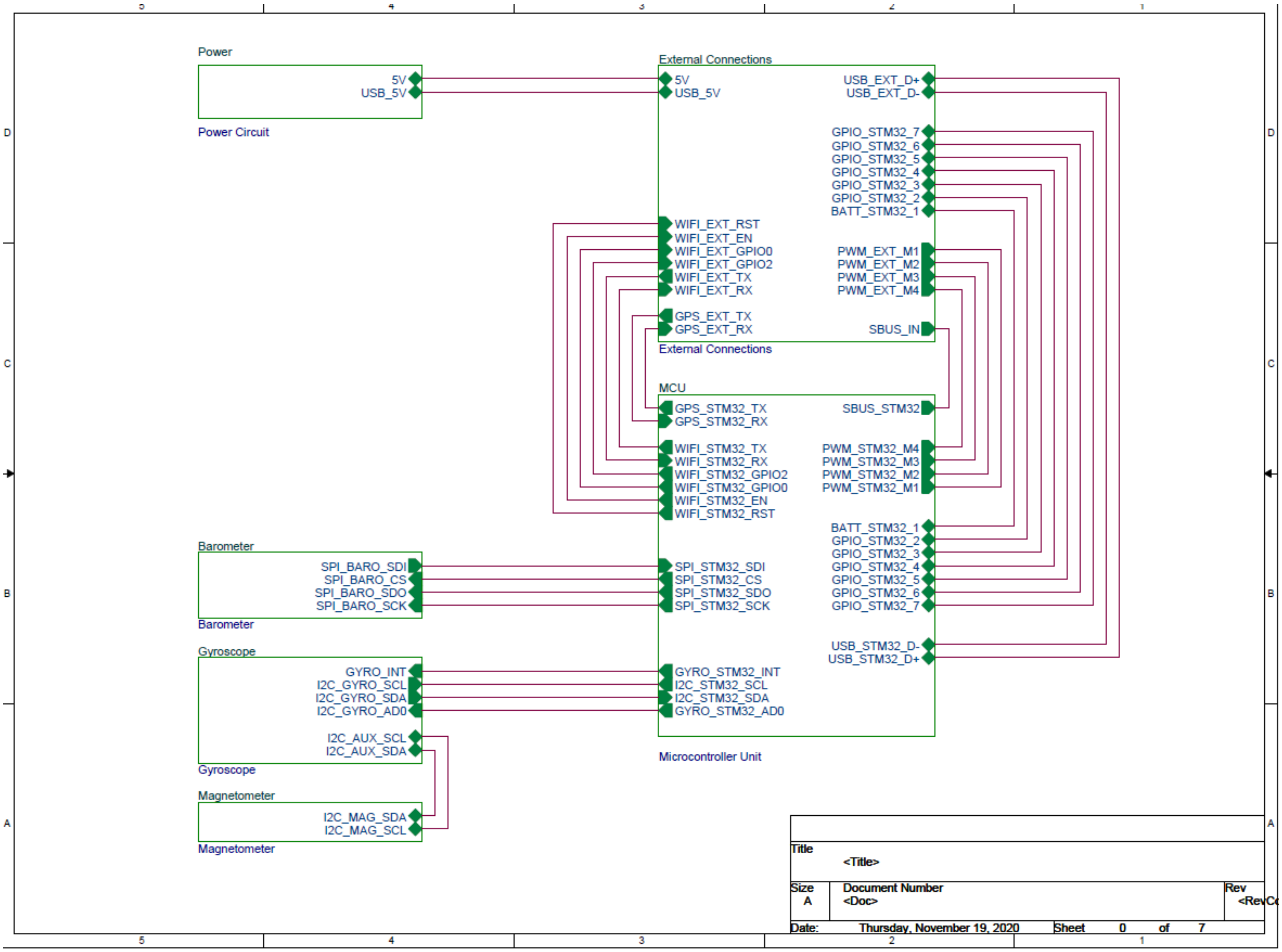

Figure 2: Component Connections Schematic

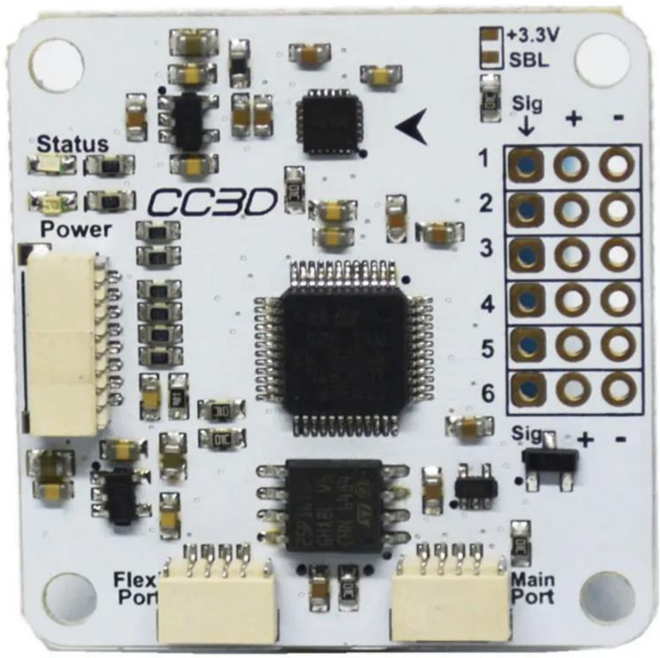

Figure 3: CC3D Controller Board

Above are the general schematic of the flight controller circuit [Figure 2] and a picture of the CC3D [Figure 3]. While the circuit design in Figure 2 has been improved, it has yet to be fully implemented and tested in the prototyping process. This design will likely change over time throughout the testing stage specified in the schedule. The design of the quadcopter itself should remain mostly unchanged as the only modifications that are of concern are the power delivery circuitry to the DYS 320 and the flight controller PCB features. These features will be divided into two portions: Phase 1 and Phase 2.

Phase 1 will be the implementation of the base features of the controller such as drone maneuvers, configuring the controller's GPIO for LEDs and sound systems on status checks (low battery, crash, lost, etc.). Implementation of Phase 2 features will largely depend on the time constraints from Phase 1 and their complexity. These features are Wi-Fi, the gyroscope, and other components in the general schematic. Due to possible time constraints and the complexity of the features, the drone controller will not support First-Person View video signal nor will the controller be supporting firmware/PID configurations for drones other than a quadcopter since the design will be based off the quadcopter prompt.

4.3 Risks

Risk	**Risk Reduction**
Board Failure	This design is focused on the two boards: First the flight controller board, then the power monitoring board. In the case of a complete failure of the flight controller board, the motors will lose power and the drone will fall out of the air. If the board were to have a partial failure of the flight controller board, systems of the drone could be shut down reducing the chance of injury to by standers. In the case power monitoring board failure, there is the chance that the lithium ion (or lithium polymer) battery could be ruined. In order to prevent this from happening, we will be incorporating a sensor to monitor the level of the battery to avoid under-volting. Using the battery monitor and a speaker, the user will be notified that the battery is running low, giving them a chance to land the drone before it loses power.
Misuse	If not implemented properly, the design of the controller can be misused. The controller can be attached to almost any quadcopter with any intentions. The end user must have a warning about safety and misuse of the device, as well as the laws surrounding it.
Safety Issues	Without regulation, the quadcopter can fly up to 400 ft into the air, where it has the possibility to interfere with low flying air traffic. This must be monitored to fit within drone flying regulations set by the government. Also, the end user must be warned about the risks of flying near power/utility lines.
Environmental Issues	Since the finished quadcopter will use a lithium polymer battery, there is a concern about proper disposal of such battery. Proper disposal must be advised on final product.

4.4 Tasks

Understanding:

- Base flight controller model
- Mobile app (if time allows)

Design:

- Create schematic
- Final PCB layout

Implementation:

- Print new PCB
- Program board
  - PWM RC control signals
  - Servo control signals
  - Hover when no other control signals are given
- Hardware
  - Solder flight controller
  - Battery protection circuit
  - Support LED and sound indicators

Testing:

- Functionality of controller
- Basic firmware
- Finished firmware
- App (if time allows)

Documentation:

- Pin layout
- Instructions for final product
- All required FAA warnings

4.5 Schedule

Tasks	Dates
1. Project Planning	9/15/20 – 12/1/20
2. Design: Schematics (Provided by Electrical Engineers), Layouts	10/1/20 - 12/1/20
3. Implementation: PCB layout, Print board, Program board, control signals, Hover mechanic, Hardware, Soldering, Battery protection circuit, LED & sound.	10/15/20 – 2/20/21
4. Testing: Functionality, Firmware, App	2/15/21 – 4/1/21
5. Documentation: Pin layout, Instructions, FAA warnings	10/12/20 – 3/30/21
6. Programming System: Controller Functionality, Firmware, Testing, Coding, Application (potentially)	1/30/21 – 4/1/21
7. Final Flight Controller Presentation	4/15/21

4.6 Deliverables

The deliverables for this project are as follows:

- A System Block Diagram & Specifications. This will contain both the hardware and software components and the PCB layout.
- Schematics of the flight controller. This will contain the architecture of the flight controller.
- The website for the project. This template is provided.
- Final Documentation and PCB.
- The finished flight controller and accompanying power board.
- The code used for firmware/software/potential application developed. These languages are to be determined. This will be supplied in a zip file.
- Github repositories. The link and all associated files will be provided.
- Trello task list. This will be provided if we choose to use Trello to keep up with tasks.
- Research paper. This will be the final report for the project and include results and a bug report.

5.0 Key Personnel

Zachary Heil - Heil is a senior Computer Engineering and Electrical Engineering double major. He will be responsible for being the team leader of both the ELEG and the CSCE team for the project.

Lily Phu – Phu is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed software engineering and is experienced in many programming languages. She will be responsible for the meeting notes and the software design of the project.

Stephanie Phillips - Phillips is a senior Computer Engineering major at the University of Arkansas. She has completed software engineering, digital design, computer organization, embedded systems and is experienced with C, Verilog, VDHL, and Python. She will be working on the software and hardware design of the project.

Spencer Ward – Ward is an undergraduate senior computer engineer at the University of Arkansas. He has experience with VHDL and Verilog, as well as C and C++. Currently in training for Dr. Di, he is researching asynchronous design technologies and will be flexible working on software and/or hardware.

Dishoungh White II - White is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Digital Design, Software Engineering, Microprocessor Systems Design, and Embedded Systems. He will be responsible for scheduling meetings for the CSCE team and assisting with the software and hardware design of the project.

Andy McCoy – McCoy is a senior Electrical Engineering major in the Engineering Department at the University of Arkansas. He is responsible for researching information about the project.

Joel Parker - Parker is a senior Electrical Engineering major in the Engineering Department at the University of Arkansas. He is responsible for communicating between the team and the L3 Representative.

Christ Somophounout – Somophounout is a senior Electrical Engineering major in the Engineering Department at the University of Arkansas. He is responsible for documenting information on the project.

Alex Cutsinger (Champion): Cutsinger is a Software and Electrical Engineer for L3 Technologies, who graduated from the University of Arkansas with a Bachelor’s Degree in Electrical Engineering. Cutsinger’s interest are robotics and mathematics.

6.0 Facilities and Equipment

- Quadcopter
- Development Board

7.0 References

[1] CC3D Flight Control Board (Users Manual), https://www.geeetech.com/Documents/CC3D%20flight%20control%20board.pdf

[2] Complete List of Flight Controller Firmware Projects, https://blog.dronetrest.com/flight-controller-firmware/

[3] CopterControl / CC3D / Atom Hardware Setup, https://opwiki.readthedocs.io/en/latest/user_manual/cc3d/cc3d.html

[4] KiCad STM32 Hardware Design and JLCPCB Assembly, https://www.youtube.com/watch?v=t5phi3nT8OU&feature=emb_title

[5] PAM vs PWM vs PPM, https://circuitglobe.com/difference-between-pam-pwm-and-ppm.html

[6] PPM Definition, https://sourceforge.net/p/arduinorclib/wiki/PPM%20Signal/

[7] PWM Definition, https://www.analogictips.com/pulse-width-modulation-pwm/

[8] XCITE 320 Quadcopter, https://hobbyking.com/en_us/dys-320-glass-fiber-folding-quadcopter-with-storage-case-pnf.html