Helicopter Collective Control for Flight Simulators

8/3/2021

Group 20

Mark Green – Computer Engineering

David Green – Computer Engineering

Sven Hall – Computer Engineering

Joseph Pergola – Electrical Engineering

Table of Contents

1.0 Executive Summary 1

2.0 Project Description 2

2.1 Project Motivation 2

2.2 Technical Background 2

2.3 Goals and Objectives 5

2.4 Project Functionality 6

2.5 Engineering Requirement Specifications 7

2.6 House of Quality 8

2.7 Division of Responsibilities Block Diagram 9

2.8 Estimated Budget and Financing 11

2.9 Project Milestones 12

3.0 Design Constraints and Standards 13

3.1 Design Constraints 13

3.1.1 Economic 14

3.1.2 Environmental 14

3.1.3 Health and Safety 15

3.1.4 Manufacturability 15

3.1.5 Sustainability 16

3.1.6 Time 16

3.1.7 Ethical 17

3.2 Design Standards 17

3.2.1 Human Interface Devices (USB-HID) 17

3.2.2 Bluetooth Core Specification 18

3.2.3 Bluetooth Low Energy 18

3.2.4 Generic Attribute Profile (GATT) 19

3.2.5 HID Over GATT 20

3.2.6 C Style Guide SEL-94-003 20

3.2.7 Globally Harmonized System of Classification and Labelling of Chemicals (GHS) 20

3.2.8 RoHS & WEEE 21

3.2.9 IEEE Standards 22

3.2.10 Electrical Safety Standards 23
4.0 Project Research 23
4.1 Similar Products and Projects 23
4.1.1 Xbox Controller 24
4.1.2 X52 H.O.T.A.S 24
4.1.3 Komodo Simulations Collective 25
4.1.4 Angular Position Project 25
4.2 Rotation and Angle Sensors 26
4.2.1 Rotary Potentiometers 26
4.2.2 Rotary Encoders 26
4.2.2.1 Incremental Rotary Encoders 26
4.2.2.2 Absolute Rotary Encoders 27
4.2.2.3 Rotary Encoder Resolution and Pulses Per Revolution (PPR) 27
4.3 Wireless Communication 28
4.3.1 Wireless Technologies 28
4.3.1.1 Infrared 28
4.3.1.2 Wi-Fi Direct (IEEE 802.11) 29
4.3.1.3 Zigbee 30
4.3.1.4 Bluetooth Low Energy (BLE) 30
4.3.2 Bluetooth Modules 31
4.3.2.1 RN42 31
4.3.2.2 RN4870/71 32
4.3.2.3 BM70/71 33
4.3.2.4 Initial Bluetooth Module Selection 34
4.3.2.5 Final Bluetooth Module Selection 34
4.4 Microcontroller 34
4.4.1 Atmega2560 35
4.4.2 MSP430FR6989 35
4.4.3 ESP32-WROOM-32SE (ESP32-D0WD-V3) 36
4.4.4 Texas Instruments CC2640 36
4.4.5 Microcontroller Comparison 36
4.4.5.1 Clock Rate Comparison 37

4.4.5.2 Flash Memory Comparison.. 37
4.4.5.3 GPIO Comparison.. 38
4.4.5.4 Timer Comparison 38
4.4.5.5 Power Comparison.. 39
4.4.5.6 Cost Comparison.. 39
4.4.5.7 Microcontroller Selection 40
4.5 User Interface 41
4.5.1 Toggle Switches 42
4.5.2 Multi-Axis Input.. 43
4.5.2.1 Hat Switch.. 43
4.5.2.2 Analog Joystick.. 43
4.5.2.3 Multi-Axis Input Selection 43
4.5.3 Information Display.. 44
4.5.3.1 OLED screen.. 44
4.5.3.2 LED Indicators.. 45
4.5.3.3 LCD Panel.. 45
4.5.3.4 Information Display Selection 45
4.6 Fabrication.. 45
4.6.1 Fabrication Methods.. 46
4.6.1.1 Machining 46
4.6.1.2 3D Printing.. 46
4.6.1.3 Fabrication Method Comparisons and Selection 47
4.6.2 3D Printing Materials 47
4.6.2.1 ABS.. 47
4.6.2.2 PLA 48
4.6.2.3 PETG.. 48
4.6.2.4 Material Comparison and Selection.. 49
4.7 Power System.. 49
4.7.1 Power System Options.. 50
4.7.1.1 Arduino Prototyping and Power Considerations 50
4.7.1.2 Power by Wall Outlet.. 51
4.7.1.3 Power by USB.. 51

4.7.1.4 Power by Battery 51
4.7.1.5 Power Source Selection 52
4.7.2 Battery Options 52
4.7.2.1 Nickel Cadmium (Ni-Cd) 52
4.7.2.2 Lithium-Ion 53
4.7.2.3 Lithium-Polymer 53
4.7.2.4 Battery Decision 53
4.7.3 Linear vs Switching Regulators 54
4.7.3.1 LM1117MP 55
4.7.3.2 L7805 55
4.7.3.3 TPS63070 56
4.7.3.4 Initial Regulator Decision 56
4.7.3.5 Final Regulator Decision 57
4.7.4 Battery Safety & Management 57
4.7.4.1 Protective Charging 57
4.7.4.1.1 LM 317 58
4.7.4.1.1 BQ25611D 58
4.7.4.1.2 MCP73831 58
4.7.4.1.3 LTC1730 and TP4056 58
4.7.4.1.4 Initial Charging IC Decision 59
4.7.4.1.5 Final Charging IC Decision 59
4.7.5 Power Consumption 59
5.0 Project Design 61
5.1 Electrical Hardware Design and Schematics 61
5.1.1 Initial Power Design and Prototyping Overview 61
5.1.1.1 Electrical Design Software 61
5.1.1.1.1 EAGLE 61
5.1.1.1.2 Altium Designer 62
5.1.1.1.3 KiCad 62
5.1.1.1.4 Fritzing 62
5.1.1.1.5 NI Multisim 62
5.1.1.2 Electrical Design Software Decision 63

5.1.2 Logic and Input Subsystem ... 63
5.1.2.1 Collective Head Input ... 65
5.1.2.2 Axis Sensors ... 66
5.1.2.3 Physical User Interface ... 66
5.1.2.4 Bluetooth Module ... 66
5.1.2.5 Programming Interfaces ... 67
5.1.2.6 Microcontroller ... 67
5.1.2.7 Logic and Input Subsystem Schematic ... 68
5.1.3 Power Design ... 69
5.1.3.1 Architecture ... 69
5.1.3.2 USB-C Module ... 70
5.1.3.3 Battery Charging Module ... 71
5.1.3.4 Power Regulation Module ... 73
5.1.3.5 Power Passthrough ... 75
5.1.4 PCB ... 76
5.1.4.1 PCB Manufacturers ... 77
5.1.4.1.1 PCBWay ... 77
5.1.4.1.2 JLCPCB ... 77
5.1.4.1.3 OSH PARK ... 77
5.1.4.2 PCB Assembly and Fabrication ... 78
5.1.4.2.1 Bill of Materials ... 81
5.2 Software Design ... 81
5.2.1 Software Tools ... 82
5.2.1.1 USB-IDF HID Descriptor Tool ... 82
5.2.1.2 Microchip IS187x BLEDK3 Bluetooth UI Configuration Tool ... 82
5.2.1.3 Microchip BLEDK3 Manual Test Host Emulation Tool ... 83
5.2.1.4 Arduino IDE (Integrated Development Environment) ... 83
5.2.2 HID Configuration ... 84
5.2.2.1 Report Descriptor ... 84
5.2.3 Bluetooth Low Energy Configuration ... 85
5.2.4 High-Level Software Design ... 86
5.2.5 Encoder Input ... 88

5.2.5.1 Encoder Input Processing 88
5.2.5.2 Encoder Calibration 89
5.2.5.2.1 Calibration-Based Value Modification 90
5.2.5.2.2 Calibrated Encoder Value Lookup Table 93
5.2.6 Collective Head Input 94
5.2.6.1 Toggle Switch Logic 94
5.2.7 Bluetooth Communication 95
5.2.7.1 BM70 Command/Response Format 95
5.2.7.2 Bluetooth Pairing 96
5.2.7.3 Gamepad Input Commands 96
5.2.8 Collective Base User Interface 97
5.2.8.1 User Interface Functionality 97
5.3 Mechanical Design 97
5.3.1 Collective Base and Lever 99
5.3.1.1 Throttle Assembly 99
5.3.1.2 Base Assembly 102
5.3.2 Collective Head 106
5.3.2.1 Collective Head Modeling 106
5.3.2.2 Collective Head Attachment 107
6.0 Prototyping and Testing 108
6.1 Bluetooth Testing 108
6.1.1 BM71-XPRO Development Board 109
6.1.2 Initial Bluetooth Testing 109
6.1.3 Breadboard Bluetooth Testing 111
6.2 Functional Prototyping 111
6.2.1 Functional Prototyping Methodology 111
6.2.2 Encoder Input Prototyping 112
6.2.3 Bluetooth Communication Prototyping 112
6.2.4 Procedural Steps for Electrical Hardware Testing 113
6.2.5 Mechanical Testing 114
6.2.5.1 Mechanical Tolerance 114
6.2.5.2 Mechanical Strength 115

6.2.5.3 Mechanical Assembly 116
6.3 Specification Testing 116
7.0 Administrative Content 117
7.1 Product Setup and Operation 117
7.1.1 Assembly and Physical Installation 118
7.1.2 Initial Setup and Configuration 119
7.1.3 Flight Simulation Software Configuration 119
7.2 Final Budget 120
7.3 Conclusion 122
8.0 Appendices 124
8.1 Copyright Permissions 124
8.2 References 127

List of Figures

Figure 1 - Airplane Flight Control Axes ... 3
Figure 2 - Helicopter Flight Control Axes ... 4
Figure 3 - Effect of Collective Lever Movement on Rotor Blade Pitch Angle ... 4
Figure 4 - House of Quality ... 8
Figure 5 - Division of Responsibilities Block Diagram ... 9
Figure 6 - GATT-Based Profile Hierarchy ... 19
Figure 7 – Overview of Logic and Input Subsystem Components ... 63
Figure 8 – Logic and Input Subsystem Schematic ... 68
Figure 9 – Power System Component Overview ... 70
Figure 10 – Graph of Regulated Charge Current vs Regulated Battery Voltage ... 72
Figure 11 – Battery Charging Module Schematic ... 72
Figure 12 – Voltage Regulator Schematics ... 74
Figure 13 – Power Passthrough Schematic ... 76
Figure 14 - PCB Layout ... 80
Figure 15 - 3D Render of Completed PCB ... 80
Figure 16 – HID Report Descriptor ... 85
Figure 17 – Bluetooth Low Energy GATT Service Table Configuration ... 86
Figure 18 - High-Level Software Block Diagram ... 87
Figure 19 – EMS22A Absolute Encoder Timing Diagram ... 88
Figure 20 – High-Level Calibration Process Diagram ... 90
Figure 21 – Low-Level Calibration Logic Diagram ... 93
Figure 22 – BM70 General Command / Response Format ... 96
Figure 23 – Complete Collective Assembly ... 98
Figure 24 – Complete Collective Assembly, Exploded View ... 99
Figure 25 – Collective Throttle Control ... 100
Figure 26 – Collective Throttle Shaft Assembly ... 100
Figure 27 – Throttle Encoder Housing ... 101
Figure 28 – Collective Lever Collar-Bearing Combination ... 102
Figure 29 – Collective Lever Geartrain ... 103
Figure 30 – Throttle Base Unit Structure ... 104
Figure 31 – Collective Lever Friction Adjustment Mechanism ... 104
Figure 32 – Collective User Interface ... 105
Figure 33 – UH-1 Huey Head and Bell 206 Head ... 106
Figure 34 – Collective Head Quick Detach Mechanism ... 107
Figure 35 – Collective Lever Threaded End Cap ... 108
Figure 36 - Tolerance Testing Component ... 114
Figure 37 - Product Setup Flowchart ... 118
Figure 38 – Complete Helicopter Collective Control Unit ... 123
Figure 39 – Interchangeable UH-1 Huey and Bell 206 Heads ... 123

List of Tables

Table 1 – Engineering Requirement Specifications 7
Table 2 – Block Status 10
Table 3 – Estimated Budget 11
Table 4 – Semester 1 Project Milestones 12
Table 5 – Semester 2 Project Milestones 13
Table 6 – Microcontroller Clock Frequency Comparison 37
Table 7 – Microcontroller Flash Memory Comparison 38
Table 8 – Microcontroller GPIO Pin Comparison 38
Table 9 – Microcontroller Timer Comparison 39
Table 10 – Microcontroller Power Comparison 39
Table 11 – Microcontroller Cost Comparison 40
Table 12 – Microcontroller Rankings 40
Table 13 – Battery Type Comparison 54
Table 14 – Component Worst Case Power Values 59
Table 15 – Component Typical Power Values 60
Table 16 – Comparison of PCB Manufacturers 78
Table 17 – PCB Bill of Materials 81
Table 18 – Final Specification Testing Results 117
Table 19 – Final Budget 120

List of Equations

Equation 1 – Rate of Charge Regulated Current Equation 71
Equation 2 – Voltage Regulator Resistor Equation 74
Equation 3 – Normal Equation for Calculating Size of Encoder Range........................... 91
Equation 4 – Rollover-to-Zero Equation for Calculating Size of Encoder Range 91
Equation 5 – Normal Equation for Shifting Encoder Values 91
Equation 6 – Rollover-to-Zero Equation for Shifting Encoder Values when Raw Encoder Value Exceeds Lower Bound... 91
Equation 7 – Rollover-to-Zero Equation for Shifting Encoder Values when Lower Bound Exceeds Raw Encoder Value 91
Equation 8 – Calculation of Final Calibrated Encoder Value... 92

1.0 Executive Summary

In the current market of consumer flight simulation equipment, there exists a marked lack of options, especially when it comes to low-cost options, for helicopter-specific flight controls. This project aims to address this fact by designing and developing a relatively inexpensive, portable, modular, and accurate helicopter collective control unit designed for use as a controller in flight simulators. This specific control was chosen for this project as the function of the collective in a helicopter is rather unique in that, although it does affect movement in the vertical axis by altering lift, it does so in a manner that is very different than a traditional aircraft such as an airplane. It is this uniqueness and lack of available options that led our team to decide upon the construction of a helicopter collective unit as the focus of this project.

The helicopter collective unit consists of three major physical components: the collective heads, which house various forms of input including buttons and switches; the collective lever, which performs the actual adjustment of the angle of the rotor blades to alter lift; and the collective base to which the collective lever is attached. Multiple collective heads were constructed, each based on different existing helicopter collective designs, that can be swapped out in order to more accurately match the simulation of different types of helicopters. These heads connect to the collective lever, which has two axes of rotation: the rotation of the collective lever itself relative to the collective base and the rotation of a throttle control embedded in the collective lever shaft. The various inputs from the attached collective head as well as the sensor data from both the collective lever and throttle control are then interpreted by a microcontroller in the collective base. The microcontroller then outputs simulated game controls that are sent wirelessly via Bluetooth to an attached computer for use in flight simulation software.

As mentioned previously, the primary aspects of our design that make it unique are its relatively low cost, portability, modularity, and accuracy. Our design addresses each of these goals in different ways. In order to maintain a low cost, our design utilizes 3D printing as the primary method of fabricating the physical components of the collective, as the materials used in 3D printing are inexpensive and allow for rapid prototyping. To address the aspect of portability, our design implements wireless capability via Bluetooth Low Energy as well as a power system based off battery power. In terms of modularity, our design allows for swappable collective heads to better simulate various kinds of helicopters. Finally, in regard to the issue of accuracy, our design utilizes an absolute rotary encoder in order to determine the angle of the collective lever at any given moment, which provides a high level of accuracy compared to other sensor options. Overall, these features collectively form a product that is both a realistic analogue to existing helicopter collectives and a preferable alternative to existing options for simulated helicopter flight controls.

2.0 Project Description

2.1 Project Motivation

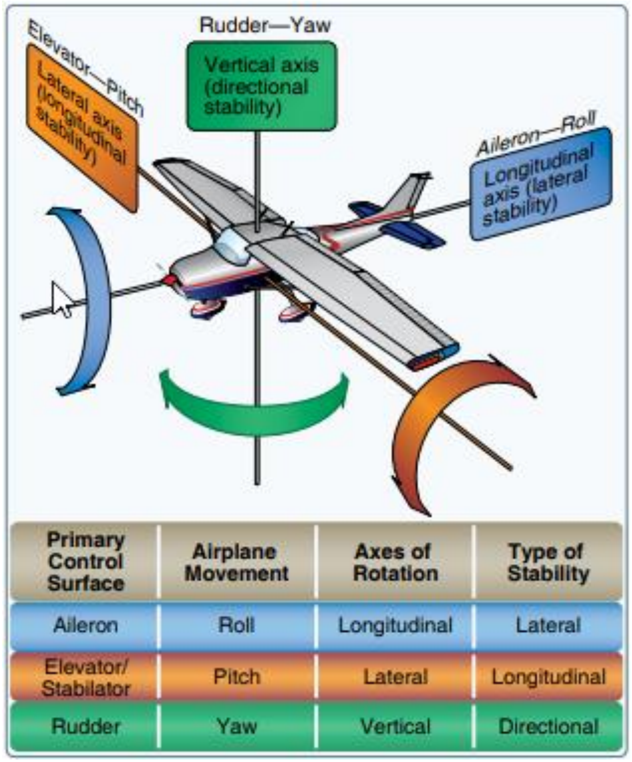
The primary motivation in selecting this project stems from a void currently present in the market of consumer flight simulation equipment. As modern flight simulation software has become increasingly advanced and realistic, so have consumer products that aim to assist in accurately recreating the experience of flying an airplane or helicopter in a virtual environment. These products range from aftermarket or replica plane seats to physical instrumentation to link into a simulator, realistic flight controls, and many others. Among these products, there specifically exists a wide range of options for consumer flight controls designed for simulation of flying an airplane, of which options vary from simple and affordable to expensive and rather costly.

For example, the most common kind of controller available for flight simulators to the average consumer are HOTAS ("hands on throttle-and-stick") style devices that simply implement a throttle lever, flight control stick, and an array of buttons. This kind of controller is rather ubiquitous in the realm of flight simulators due to its generic nature that allows it to serve as an effective and at least somewhat realistic control for a wide variety of airplanes. These types of HOTAS controllers can be acquired at varying price points and capabilities very easily and are extremely prevalent among consumer flight simulation equipment. In contrast, there is a marked lack of options for helicopter-specific flight controls that are both affordable, realistic, and designed for flight simulators. There are likely multiple reasons for this, some of which may be the rather niche experience required to operate a helicopter realistically in a flight sim, the relatively more complex control scheme of a typical helicopter when compared to average plane controls, or the primary focus of modern flight simulators on providing a wide range of airplanes to simulate as opposed to a smaller number of helicopters. Whatever the case may be, the problem remains that it is difficult to acquire affordable, yet somewhat realistic, helicopter flight controls designed for flight simulators, which is the issue that has motivated the creation of this project.

2.2 Technical Background

In order to fully understand the purpose and motivation behind the development of a helicopter flight simulator collective, some technical background knowledge related to how aircraft are controlled in real life is required. The primary flight mechanisms that most people are familiar with are those of traditional airplanes, which consist of the ailerons, elevators, and rudder as shown in Figure 1 below. The ailerons provide rotation around the longitudinal axis which "rolls" the aircraft. The elevators provide rotation around the lateral axis which "pitches" the aircraft up or down. Finally, the rudder provides rotation around the vertical axis which alters the "yaw" of the aircraft. These primary mechanisms and their axes are shown in Figure 1 below. The primary physical controls which the pilot operates to affect these axes of motion are the yoke or joystick to control the ailerons and elevators, and pedals to control the rudder. Also, while not directly correlated to axial movement, a throttle control is present to adjust engine speed and thrust in flight and could be considered a primary control.

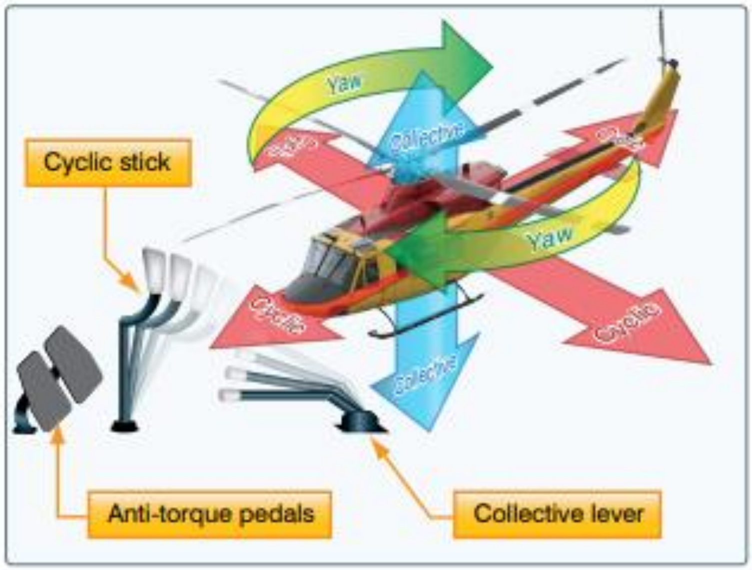
Figure 1 - Airplane Flight Control Axes



Note. This figure shows the three primary airplane controls and the axes of movement which they control. Reprinted from *Pilot's Handbook of Aeronautical Knowledge* (p.6-3), by the Federal Aviation Administration, 2016, United States Department of Transportation, Federal Aviation Administration, Airman Testing Standards Branch.

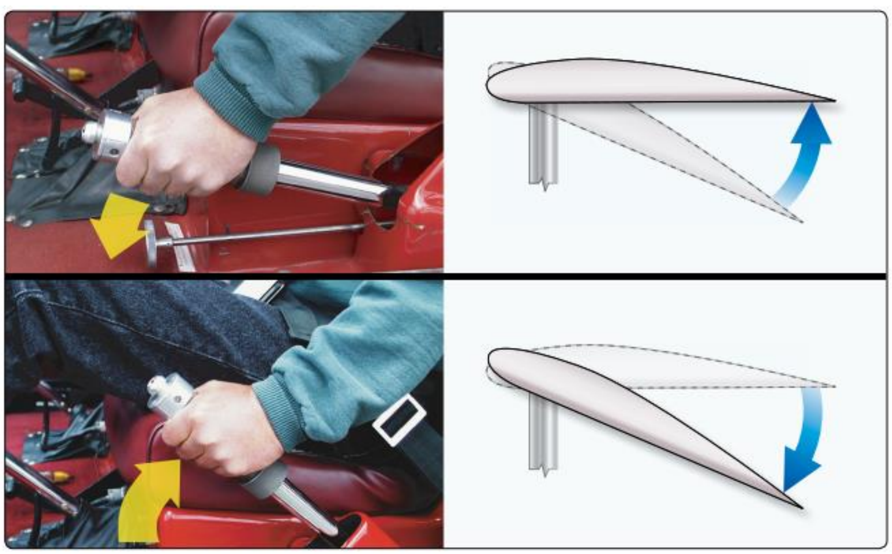
A helicopter functions in a similar manner to a plane in that it has controls which alter the rotation of all three axes, however the mechanisms used to achieve movement are significantly different from that of an airplane. The primary physical controls present in a helicopter are the cyclic (also known as joystick), the collective lever, and the pedals. Also, like an airplane, a throttle control is present to affect engine speed and is usually present as a control on the collective lever. The collective lever changes the pitch of all blades on the rotor simultaneously in order to affect the overall lift generated. This change in lift results in the helicopter moving in the vertical axis. The cyclic control tilts the entire rotor disk assembly which allows the helicopter to move in any desired horizontal direction. Finally, the pedals control the tail rotor blades which affect the yaw of the helicopter. The effects of these controls are shown in Figure 2 and Figure 3 below. While the cyclic is similar to the yoke of an aircraft and the pedals are similar to airplane pedals, the collective lever is a unique control specific to helicopters.

Figure 2 - Helicopter Flight Control Axes



Note. This figure shows the three primary helicopter controls and the axes of movement which they control. Reprinted from *Pilot's Handbook of Aeronautical Knowledge* (p.6-2), by the Federal Aviation Administration, 2016, United States Department of Transportation, Federal Aviation Administration, Airman Testing Standards Branch.

Figure 3 - Effect of Collective Lever Movement on Rotor Blade Pitch Angle



Reprinted from *Helicopter Flying Handbook* (p.3-2), by the Federal Aviation Administration, 2019, United States Department of Transportation, Federal Aviation Administration, Airman Testing Branch.

2.3 Goals and Objectives

In forming this project, there were a few key aspects of design that were decided upon as the primary goals and objectives for the project. These goals stem from the issues presented when discussing the motivations for this project, and they each aim to cover a specific key component in what we consider to be a satisfactory final product. The primary goals for this project are thus that we develop a controller that is relatively low cost, portable, accurate, and modular.

The first of these goals, a relatively low cost, is directly related to the issue discussed previously of a lack of affordable options for this kind of unique helicopter control for use in flight simulators. Even when considering the lower end of similar consumer products, most options range from multiple hundreds to thousands of dollars, with cheaper options lacking some functionality such as no complex collective heads or any sort of portability. As such, a relatively low cost is extremely important when considering the motivation of this project to make a more easily accessible and affordable consumer controller option for helicopter control in flight simulators.

The second key goal for this project, portability, is related to the fact that flight simulator controls are ideally not meant to be permanently affixed or placed. It is the nature of flight simulation that you can experience a realistic flying experience outside of flying an actual aircraft, and as such it is important that we keep in mind the need for a flight simulator controller to be portable to be used on multiple devices or to be moved from its initial location with relative ease.

The third goal for this project, accuracy in relation to simulator input, is absolutely essential to the overall success of this project. Piloting a helicopter in a flight simulator using physical controllers provides a sense of realism that is not possible using normal game controllers. If the accuracy of the physical controller itself, in this case the collective control, is not sufficient, an issue arises in which there is a large disconnect between the physical operation of the controller and the representation of the movement of the controller in a flight simulator. If this were to be the case, the entire benefit of having a dedicated physical control for the collective is negated. As such, it is a priority for this project that any input on the collective is accurately represented in the simulation environment.

The fourth and final goal for this project, modularity, is in some ways an offshoot of the same concerns that make accuracy a key goal in this project. Modern flight simulators are extremely detailed, often implementing a complete view of the interior of aircraft, including helicopters, that show all aspects of the controls of an aircraft, including the collective. The goal of modularity in this project aims to utilize this detail by providing a modular solution to customizing the collective such that it may better represent different models and types of helicopters. This will provide a more realistic and satisfying end result if the physical controller you have can be modified in some way to look or act more like the simulated aircraft.

2.4 Project Functionality

Core Features

- The unit will be able to measure the current angular position of the collective lever.
- The unit will be able to measure the current angular position of the twist throttle.
- The unit will have a head which is present on the top of the collective lever with additional controls.
- The unit will convert collective lever angle and twist throttle angle to gamepad-like joystick axis inputs.
- The unit will be able to connect to a computer and control a flight simulator.

Advanced Features

- The unit will be able to connect wirelessly to a computer rather than requiring a physical connection.
- The unit will have a rechargeable battery to allow for more flexible usability and portability.
- The unit will have a physical display which reports the current values of the collective lever and twist throttle axes.
- The unit will have physical controls for calibrating the minimum and maximum position of the collective lever.
- The unit shall have at least 2 swappable control heads to mimic 2 different helicopters.

Stretch Goals

- The unit will have a battery life indicator to allow the user to check when it must be recharged.
- The unit shall have additional swappable heads to mimic more helicopter control schemes.
- The unit shall provide the option of USB connection as an alternative to wireless connection.
- The unit shall have an adjustable amount of friction/resistance of the collective lever to accommodate user preference.
- The unit shall have a collective brake which will allow the user to lock the position of the collective lever.

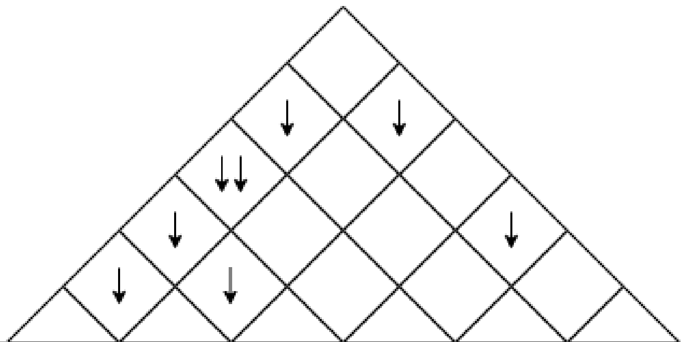
2.5 Engineering Requirement Specifications

Table 1 – Engineering Requirement Specifications

No.	Requirement	Specification
1	**Cost**	< $500
2	**Weight**	< 20 lbs.
3	**Line of Sight Wireless Connection Range**	≥ 15 ft
4	**Idle Runtime (Wireless)**	> 10 hours
5	**Collective Base Size**	< 1 cu. ft
6	**Minimum Measurable Change of Lever Angle**	< 1°
7	**Angular Variance at Idle**	±5% of total angular travel
8	**Number of Unique Collective Heads**	≥ 2 Heads
9	**Angular Poll Rate**	≥ 3 Hz
10	**Collective Lever Angular Travel Range**	≥ 30°
11	**Calibrated Digital Output Range**	Min: 0, Max: 1023
12	**Operating System Compatibility**	Windows 10

2.6 House of Quality

Figure 4 - House of Quality

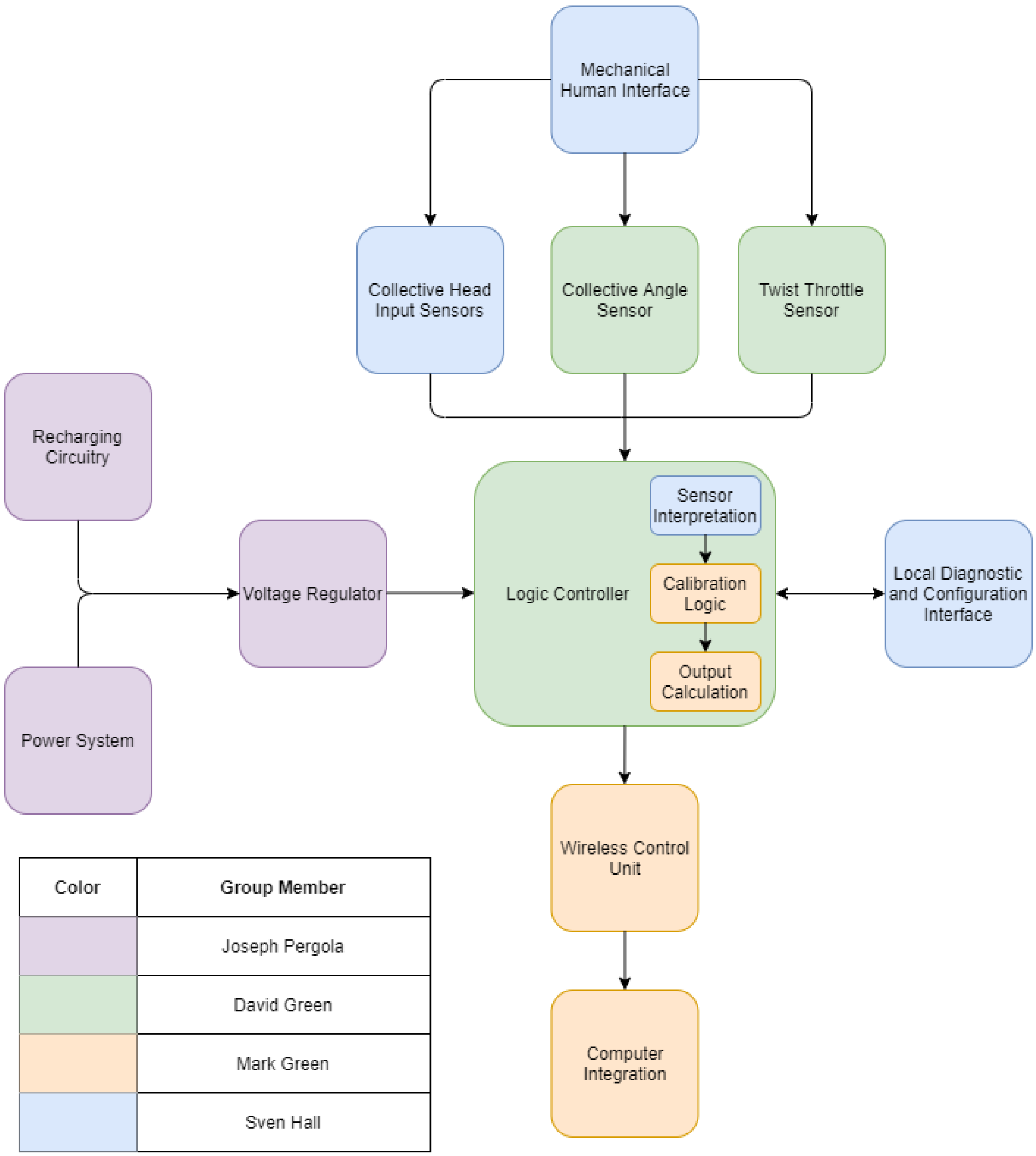


		Cost	Idle Runtime	Collective Base Size	Min. Measureable Change of Lever Angle	Number of Unique Collective Heads	Angular Poll Rate
		-	+	-	-	+	+
Low Cost	+	↑↑	↓	↓	↓↓	↓	
Accurate	+	↓			↑↑		↑↑
Portable	+	↓	↑	↑			
Modular	+	↓				↑↑	
Target Requirements		< $500	> 6 Hrs.	< 1 cu. ft	< 1°	≥ 2 Heads	≥ 3 Hz

Strong Positive	↑↑
Positive	↑
Strong Negative	↓↓
Negative	↓
Postive Polarity	+
Negative Polarity	-

2.7 Division of Responsibilities Block Diagram

Figure 5 - Division of Responsibilities Block Diagram



Color	Group Member
(purple)	Joseph Pergola
(green)	David Green
(orange)	Mark Green
(blue)	Sven Hall

Table 2 – Block Status

Block	Status
Power System	Complete
Voltage Regulator	Complete
Recharging Circuitry	Complete
Mechanical Human Interface	Complete
Collective Head Input Sensors	Complete
Collective Angle Sensor	Complete
Twist Throttle Sensor	Complete
Logic Controller	Complete
Sensor Interpretation	Complete
Calibration Logic	Complete
Output Calculation	Complete
Local Diagnostic and Configuration Interface	Complete
Wireless Control Unit	Complete
Computer Integration	Complete

2.8 Estimated Budget and Financing

This project will be self-financed by the members of our group.

Table 3 – Estimated Budget

Component	Estimated Cost ($)
Input Devices and Sensors	
Collective Shaft Angle Sensor	25-75
Throttle Twist Sensor	5-50
Switches, Hat Switches, Buttons	10-20
Mechanical	
3D Printing Filament	10-40
COTS Hardware	50-75
Electronics	
Batteries	25-50
Circuit Components (Resistors, Capacitors, Inductors, etc.)	10-25
Logic Controllers	5-25
Wireless Communication Module	5-15
PCB Manufacturing	5-30
Total Estimated Cost Range	**150-405**

2.9 Project Milestones

Semester 1

Table 4 – Semester 1 Project Milestones

Milestone	Deadline
Divide and Conquer 1.0	01/29/2021
Divide and Conquer Advisor Review Meeting	02/03/2021
Divide and Conquer 2.0	02/12/2021
Research Input/Sensor Devices	02/15/2021
Research Logic Controller Options	02/19/2021
Research Wireless Communication Standard	02/22/2021
Research Power Design Options	02/26/2021
Initial Circuit Design	03/15/2021
Software Design/Programming	03/26/2021
Initial Mechanical Design	04/09/2021
Initial Prototyping	04/16/2021
PCB Design	04/26/2021

Semester 2

Table 5 – Semester 2 Project Milestones

Milestone	Deadline
Finish Prototyping	May 2021
Completed Mechanical Design	June 2021
PCBs Manufactured	June 2021
Mechanical and Electrical Integration	July 2021
Product Assembly	July 2021
Function Testing	July 2021
Final Product	August 2021

3.0 Design Constraints and Standards

In performing the task of designing this project, it was necessary that the design process included considerations of design limitations and industry requirements in order to ensure the design and creation of a successful end product. These come in the form of both design constraints and design standards, each of which will be explored throughout this section in order to determine what factors affected our overall design throughout the process of designing and creating a satisfactory final product.

3.1 Design Constraints

Design constraints refer to specific limitations imposed upon the project either by external forces or by the design team when considering necessary constraints for a project. These constraints can cover a wide range of potential topics such as economic constraints, time constraints, health constraints, and many others that will be explored throughout this section. Design constraints are not necessarily entirely concrete and may have some abstract elements as long as they are relevant limitations or considerations that must be taken into account in the overall design. The following subsections explore in detail all of the kinds of design constraints taken into account throughout the process of designing this project.

3.1.1 Economic

One of the most apparent and most significant type of constraints that can be identified regarding the design of this project are economic constraints. The two major forms of economic constraints that are relevant to this project are that of funding and overall motivation for the project.

The first of these constraints, funding, refers to acquiring the funds necessary to procure the materials and parts necessary to design, prototype, build, and test the product throughout this project. Our team decided that this project would be entirely self-funded, meaning that rather than having a private business or public entity providing the necessary funds to complete the project, we instead used our own money. This constraint was taken into account at every point of the project, from choosing microcontrollers and sensors to choosing a material for the collective heads, in order to ensure that we were able to complete the project within a reasonable budget for all team members. This also means that there were less funds available to purchase multiple kinds of parts to initially test to help choose which parts to implement in our design, as well as less funds to replace parts in the event that they did not work as expected or are fault or broken. As this was the case, it was extremely important that we performed sufficient research into the most cost-effective options for every part of our project that would still perform as required for the design to function in a satisfactory manner. All team members agreed to equally provide funding for the project as necessary.

The second primary form of economic constraint on this project refers to the overall motivation in choosing this project. It was stated earlier in this document that one of the primary goals for this project was to maintain a low cost in order to differentiate this project as a low-cost alternative to some similar available options in the flight simulation market. This being the case, we as a team did our utmost to uphold the budget set out for this project in order to achieve one of the primary goals identified for this project.

3.1.2 Environmental

The environmental constraints on this project primarily come in the form of constraints related to the effects of this project on the environment as opposed to the effect of the environment on this project. As our project is intended to be used primarily indoors at common room temperatures, the environment as an external force does not meaningfully constrain the design in this project. This project, however, does indirectly have an effect on the environment which creates environmental constraints on the project. Most notably, the materials used in this project may have adverse effects on the environment in either how they are manufactured or eventual disposed of. For example, the material used to 3D print parts, as well as the chemicals contained within the batteries used for the system. Such materials can possibly be dangerous for humans and act as environmental contaminants if not utilized correctly. As such, we constantly took this constraint into consideration throughout this project when selecting parts and materials in order to minimize the possible environmental concerns that may be related to this project.

3.1.3 Health and Safety

The health and safety constraints associated with this project are related to the fact that this project is creating a product that is both intended for indoor use and intended for direct physical interaction by an end user. As a result, there are potential health and safety concerns that had to be taken into account when designing this product. For example, due to the fact that the collective unit is designed to be used indoors as a game controller attached to a computer, the issue of flammability arising from the use of electronics in an enclosed casing must be addressed. We addressed this concern by doing thorough research into and implementing proper power standards and by attempting to use materials that are not extremely flammable. There is also a potential safety issue that arises from the fact that a user is going to be operating the collective lever attached to a physical axis. This safety issue is in the form of an individual being injured in some form due to the mechanical movement of the collective lever by way of pinching or crushing. We kept this constraint in mind as we designed the collective lever mechanics in order to minimize the possibility for injury during regular use of the end product.

3.1.4 Manufacturability

One manufacturability constraint that had to be considered when designing this project was the availability of selected parts. In order to address this constraint, we did research to ensure that any parts or materials used in this project are commonly available and not obsolete or difficult to acquire. Despite this research and consideration, there were some issues of parts availability that had to be addressed throughout this project due to parts shortages. Another manufacturability constraint on this project is simply the feasibility of prototyping and constructing the physical structure of the collective, including the collective heads and enclosure. In order to address this constraint, we utilized 3D printing technology in order to print parts out of plastic. This allowed for more rapid product manufacturing on a small scale and helped to facilitate rapid prototyping. However, this solution would not scale well to a larger scale of manufacturing and was selected for this project due to only making a single product.

The final product would also likely require some amount of assembly to be performed by the consumer if the product were to go to market. For this reason, it is necessary that the design take this into account in order to minimize assembly difficulty. For example, avoiding the necessity of specialized tools or equipment in order to assemble the product, as well as simple, user-serviceable parts and assemblies wherever possible.

3.1.5 Sustainability

There are a few sustainability constraints that had to be taken into account during the design of this project. Firstly, as the collective has some mechanically moving parts, it had to be ensured that these parts were designed in such a way that they were not prone to being obstructed or easily broken. It also had to be taken into consideration that the enclosure in which the collective lever operates is sufficiently strong enough to hold up in terms of structural stability over time.

As the collective unit consists of several mechanical parts coming into contact with each other as well as a large number of plastic parts, it is expected that parts may wear or break over time. To combat this issue, the design takes into account factors such as the ability to disassemble the collective and replace individual parts. By designing the collective unit in this way, the lifespan of the product is vastly increased, as a single part failure or material degradation through repeated use will not render the entire product inoperable.

3.1.6 Time

Along with the economic constraints on this project, the simple constraint of time was one of the most significant constraints on the project as a whole. This stems from the fact that the entirety of this project, from planning and design to building and testing, had to take place within the timespan of Senior Design 1 and Senior Design 2, a roughly eight-month span of time. As such, the scope and scale of this project was carefully and constantly considered throughout the entirety of all phases of the project. For example, it had to be carefully considered what desired features or goals could feasibly be designed and implemented such that they could all be satisfactorily completed by the end of the project. It also means that every part of the project had to be paced appropriately, and thus a schedule by which the timing of various parts of the project was determined, and said schedule had to be upheld to the best of our team's abilities in order to not get behind and prevent us from completing all of the desired functionality and fulfilling all of the goals that we set out to achieve.

One concrete way in which we addressed this constraint is by meeting regularly once a week in order to evaluate our progress and change plans as necessary. By meeting regularly, we were able to more accurately and more efficiently determine issues of pacing or scope that could be addressed on a week-by-week basis in order to promote a smooth progression of the project over the course of the eight months in which it had to be completed. We also addressed this constraint by attempting to separate responsibilities as early as possible in order to ensure parallel progression of the project in a time-efficient manner. By distributing responsibilities amongst team members, we were able to cater the necessary tasks to each team member's skillset and thus result in tasks being completed sooner and progress throughout the entire project occurring much more quickly.

3.1.7 Ethical

A major ethical constraint that had to be kept in mind during the design of the helicopter collective is that while we intend to make it as realistic as possible, it is important that the distinction be made between a simulation collective intended for the consumer market and for the aviation market. It would be unethical to claim that our helicopter collective, which is generally intended to be used by consumers in flight simulation games and for casual flight simulation purposes, is an accurate replica of a true helicopter collective that could be used to train helicopter pilots or used for military purposes. Thus, our helicopter was designed to be as close to real life as possible in appearance and function but is not intended to be used for official aviation purposes.

3.2 Design Standards

In designing any product, it is important to consider relevant standards that may be applicable to the design. A standard is a document defining specific aspects of a product or service such as dimensions, implementation guidelines, safety aspects, or performance requirements. These standards are defined by various organizations or companies in order to standardize the use and creation of products that fall under specific designations. The following section identifies and explores some standards that were taken into consideration during the design of this project.

3.2.1 Human Interface Devices (USB-HID)

USB-HID, also known simply as HID or Human Interface Devices, is a specification which is maintained by the USB-IF (USB Implementers Forum), a non-profit organization which oversees USB and related specifications. HID is a specification which describes how computer peripherals and input/output devices such as keyboards, mice, gamepads, bar code readers, sensors, and a variety of other devices can be recognized and interpreted by the host device. While originally intended to be used with USB specifically, HID itself is generic and may be implemented via different physical connection methods.

HID is generally defined in the Device Class Definition for HID document created by the USB-IF which describes the specifics of how HID works and how to create devices and device drivers which adhere to the HID specification. HID facilitates peripheral to host communication through entities called Reports, which are described by Report Descriptors. Reports contain the actual data being passed between peripheral and host, and Report Descriptors are additional values which describe the format, purpose, and type of data that the peripheral utilizes. Reports are generally categorized as Input Reports (data sent from the peripheral to the host), Output Reports (data sent from the host to the peripheral), and Feature Reports (generally used for configuration).

Usages are defined categories which indicate to applications the purpose and meaning of data that is sent via an HID report. Usages are also used to define and categorize collections of input or output present on the peripheral. By receiving and analyzing the Usages reported by the HID device, the application can interpret the data in a different way, for example, as a button press, an x, y, or z axis, or a variety of other things. These usages are defined in the HID Usage Tables document published by the USB-IF which provides a list of all available Usages available in the HID Specification.

3.2.2 Bluetooth Core Specification

The Bluetooth Core Specification is the primary specification developed and maintained by the Bluetooth SIG (Special Interest Group), a nonprofit organization which oversees the qualification, development, and marketing of Bluetooth as a whole. The Bluetooth Core Specification defines multiple Core Configurations which Bluetooth devices must adhere to: Basic Data Rate, Enhanced Data Rate, High Speed, Low Energy, Basic Rate and Low Energy Combined, and Host Controller Interface. These different configurations indicate what Bluetooth protocols, speeds, and roles must be supported in order to meet qualification as a Bluetooth device.

Bluetooth itself is a primarily short range, wireless communication technology which operates in the 2.4 GHz band. Bluetooth is extremely common and is used in a variety of applications including cell phones, audio equipment, game controllers and consoles, and a variety of others. The two most essential modes of Bluetooth operation are Basic Rate and Low Energy, with Basic Rate also allowing for the optional addition of Enhanced Data Rate for more rigorous bandwidth requirements. Basic Rate Bluetooth has a transmission speed of 721.2 kb/s, with Enhanced Data Rate boosting it to 2.1 Mb/s, and 802.11 AMP boasting 54 Mb/s. Bluetooth Low Energy, on the other hand, is specifically designed for devices which require low-power consumption and are designed to run on battery power.

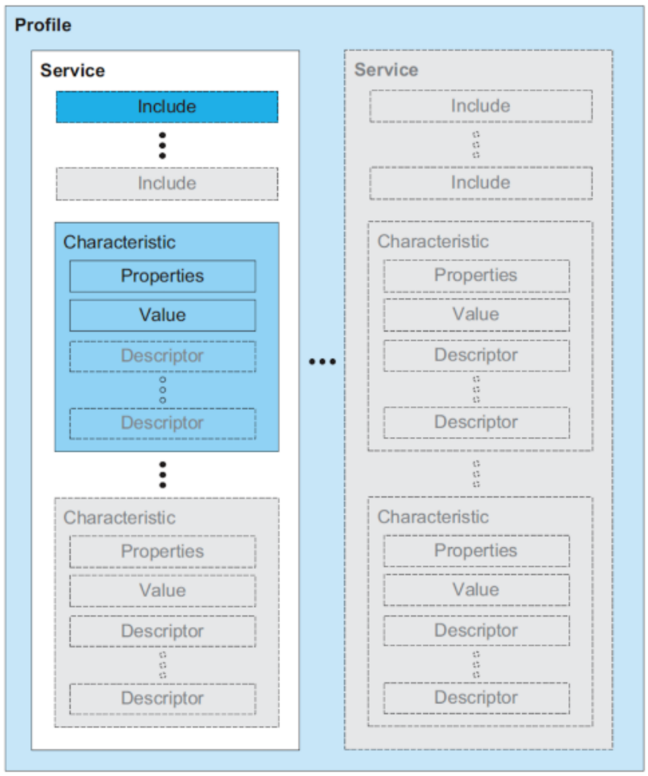
3.2.3 Bluetooth Low Energy

Bluetooth Low Energy operates in the 2.4 GHz ISM band and either utilizes FDMA (Frequency Division Multiple Access) or TDMA (Time Division Multiple Access) as its access scheme. For FDMA, Bluetooth Low Energy implements 40 channels separated by 2 MHz each, with 3 of the channels being advertising channels and the rest being data channels. Bluetooth Low Energy operates using 4 types of events: Advertising, Extended Advertising, Periodic Advertising, and Connection. Devices utilize the first three advertising events to broadcast that they are available for connection, and once a connection has been established data is sent via Connection events. Once a connection has been established, a piconet is formed which includes the master (the device which initiated connection) and the slave (the device which was connected to).

3.2.4 Generic Attribute Profile (GATT)

The Generic Attribute Profile, or GATT, is an essential profile in the Bluetooth specification which establishes how data is presented and transmitted between Bluetooth devices. GATT describes two kinds of roles which a Bluetooth device may satisfy: A Server role and a Client role. The GATT Profile specifies the structure of profiles in general; a profile contains Services, Characteristics, Characteristic Values, and Characteristic Descriptors as shown in Figure 6 below.

Figure 6 - GATT-Based Profile Hierarchy



Note. This figure shows the service hierarchy of the Generic Attribute (GATT) Profile as defined by the Bluetooth Core Specification 5.2 Reprinted with permission from *Bluetooth Core Specification Version 5.2* (p. 285) by the Bluetooth SIG, 2019.

A Service represents data which is required to implement a particular function of a device and contains multiple characteristics within it. A Characteristic is used in a service to contain information relevant to the function implemented by the service. Characteristics have Characteristic Values which are data relevant to the Characteristic and Characteristic Descriptors which describe the value.

3.2.5 HID Over GATT

The HID over GATT Profile Specification is a pre-defined profile published by the Bluetooth SIG which defines how HID functions may be implemented over Bluetooth Low Energy using GATT. The HID over GATT Profile specifies three roles: HID Device, Boot Host, and Report Host. The HID Device role is filled by a peripheral which sends HID Reports to a Boot Host or Report Host which utilizes and interprets such Reports. HID Devices as defined by the HID over GATT specification must implement the HID Service, Battery Service, Device Information Service, and, optionally, the Scan Parameters Service. The HID over GATT Profile specifies that an HID device shall implement an HID Service containing a Report Map Characteristic and a Report Characteristic. The Report Map characteristic contains the HID Report Descriptor of the HID device as defined by the USB-HID Specification. The Report Characteristic is used to transfer HID data between the Report Host and HID Device, and has a Characteristic Descriptor known as the Client Characteristic Configuration Descriptor. It is through this Report Characteristic that HID input data is passed over Bluetooth and interpreted as an HID by the host computer or device.

3.2.6 C Style Guide SEL-94-003

In an effort to increase code readability, maintainability, and to limit integration concerns between team members, the SEL-94-003 C style guide from The Software Engineering Laboratory (SEL) was chosen as a common style guide that all C programming follows for this project. SEL as an organization is sponsored by the National Aeronautics and Space Administration (NASA) and operates to improve software development processes and methodology. This style guide was chosen as our standard for this project as it thoroughly outlines all code organization aspects that would be faced during development. These include small aspects, such as the specific spacing used in the code, to the much larger scale of how multi-file projects should be organized.

Due to the fact that the programming for our product was divided amongst multiple group members, the aspect of the SEL-94-003 style guide that was most beneficial is section seven, the “statements and control flow” section of the document. This section clearly outlines the logical flow that should take place inside statements and functions, while also setting some limits on the complexity of individual functions. This is vital to a team-based development environment as it drastically increases code comprehension and the ability for another developer to debug existing code more rapidly.

3.2.7 Globally Harmonized System of Classification and Labelling of Chemicals (GHS)

The Globally Harmonized System of Classification and Labelling of Chemicals, also known as the GHS, is an international standard developed and managed by the United Nations that deals with the classification and labelling of chemicals in order to better identify and communicate any potential hazards associated with specific substances. The primary goals of the GHS are to better protect the environment and humans from potentially dangerous chemicals by harmonizing communication of hazardous chemicals, providing a base for countries without any current method of classifying chemicals, standardizing testing of chemicals for potential hazards, and help facilitate trade by providing safety information for chemicals on a global scale. The GHS has been adopted by most major countries in the world in some form, including the European Union and the United States. The United States specifically implements the GHS by aligning OSHA (The Occupational Safety and Health Administration) hazard communication standards with the GHS standard.

The GHS includes various parts necessary in the classification and communication of potentially hazardous materials, including distinct hazardous substance classifications, standards regarding how to properly label hazardous substances, standards regarding the creation of Safety Data Sheets containing chemical information about a substance, testing methods for various physical, health, and environmental hazards, as well as guidance on the use of images and graphics to visually convey hazards associated with a substance. These various elements all come together to provide comprehensive requirements for chemical classification that must be followed by members of industry in countries that implement the GHS standard.

The most relevant portion of the GHS standard in terms of this project is the standard for production of Safety Data Sheets for chemicals. Safety Data Sheets provide comprehensive information about a chemical, including identification of potential hazards, chemical composition, physical/chemical properties, disposal considerations, ecological information, and other factors. By viewing the Safety Data Sheet associated with particular substances, it was easier to identify the key aspects of certain materials or substances when considering them as potential options for use in this project.

3.2.8 RoHS & WEEE

RoHS stands for "Restriction of Hazardous Substances" and was established by the European Parliament and Council. The RoHS lays the foundation of standards for which the electronics industry and most electronics products must comply with. Any products that are deemed compliant must not contain more than the permitted amounts of the following substances: Cadmium, Lead, Mercury, Hexavalent Chromium, Polybrominated Biphenyls, Polybrominated Diphenyl Ethers, Bis(2-Ethylhexyl) phthalate, Benzyl butyl phthalate, Dibutyl phthalate, and Diisobutyl phthalate.

Directive 2002/95/EC, as stated in the table above, is the original RoHS, however, additional standards have been published since. Directive 2011/65/EU, also known as RoHS 2, was published in 2011 and added the requirement of CE Markings on certain products, which indicates that the product meets strict safety, health, and environmental requirements. Directive 2015/863, also known as RoHS 3, took effect on July 22, 2019 and added the last four substances mentioned previously to the list of controlled substances.

The RoHS integrated with the Waste Electrical and Electronic Equipment Directive (WEEE) and formed the Directive 2002/96/EC. This directive lays out standards for treating, recovering, and recycling any type of electronic or electrical device. The directives' overall objective is to promote the preservation, protection, and improvement of the environment. Towards that objective, the RoHS sets and adjusts restrictions on the material content and amounts that manufacturers can use in their products.

These directives apply to any manufacturer or distributor of electronics or electrical products, devices, equipment, components, etc. Our project complies with RoHS standards to the best of our ability. We were interested in incorporating environmentally friendly practices in our project, which is why we included some information on the WEEE Directive standards in our document. Finally, efforts were made to ensure all components used in our project are RoHS verified, and that our finalized product complies with RoHS standards, to the best of our ability.

3.2.9 IEEE Standards

The Institute of Electrical and Electronics Engineers Standards Association (IEEE SA), is an internal committee within IEEE that develops and establishes global standards throughout a wide range of industries. It is important that any initiation of projects involving the use of electronics abides by the standards set forth by IEEE, if the intent is to manufacture, commercialize, and distribute the product commercially.

At the commencement of our project, we referenced IEEE 1233 as guidance throughout the formation of our system requirements and specifications. This was in an effort to ensure the successfulness of our final product and that its design and performance are satisfactory to the needs we have envisioned. This guide outlines information on following a critical protocol, which includes: Effectively analyzing the overall theoretical procedure of our project, gathering supportive and relevant background information, developing a collaborative floorplan of the overall project, assessing components and performing useful prototyping relating to the overall design, uncovering and/or defining constraints which may pose limitations and enforce modifications on design, testing for met or unmet criteria, determining product traceability, validating components and systems to ensure they meet specified requirements, verifying overall system functioning and determining if it upholds project requirements that were initially imposed, and more.

The IEEE SA provides several guidelines for their copyright policy, which is outlined and found in Clause 7 of the IEEE SA Standards Board Bylaws. The policy includes information on the major limitations regarding copyrighted work. This information can be used to determine different cases that may be in danger of violating the rights of a copyright owner. The main terms and descriptions are listed below:

The term *contribution* is defined as any material that is presented verbally or in the form of a text, drawing, chart, video, etc. within IEEE standards development activity.

The term *Public Domain* is defined as any material that does not present any claim of rights or ownership or did not meet copyright protection requirements.

The term *Published* is defined as any material that has been distributed or made public and presents rights of ownership by the stated author. This can also be determined if the copyright symbol © is included in the document.

The term *Work Product* is defined as the collective or collaborative work amongst all members involved.

It is important to abide by any copyright policies that may be in effect regarding the nature of our project. In general, it is crucial to verify that no infringement on rights of real or intellectual property, is committed while producing any type of work.

As per project requirements and professional practice, our project complies with the applicable IEEE standards and follows advice from their guidelines to the best of our ability. We understand and will adhere to the limitations imposed by the above-mentioned list. We cautiously and collaboratively verified that we are within the IEEE Copyright Policy standards and obtained any permission that was needed for use and/or reproduction in our project.

3.2.10 Electrical Safety Standards

IPC International, Inc. is a non-profit organization that specializes in helping OEMs, EMS, and PCB manufacturers with improving electronics. IPC is also a leader in providing industry standards, and aiding in bettering related industry practices, teaching, and overall industry execution. Because our project will ultimately result in a fully integrated PCB, we followed the IPC-2220 series of standards and implemented it in our project to the best of our ability.

Furthermore, our project also follows the UL 2054 standard, which lays out the requirements on portable primary (non-rechargeable) and secondary (rechargeable) batteries when utilized as a power source for a product. The requirements give caution considering the chemical profile of the battery and its possible behavior under certain environments, while also suggesting proper use and handling of the battery. Some of the details included are the implementation, transportation, storage, and disposal of the battery, as well as risk of explosion and flammability of the battery during the previously mentioned scenarios.

4.0 Project Research

4.1 Similar Products and Projects

The consideration of existing products and previous projects allows us to properly frame the development of our project in a way that capitalizes on satisfactory aspects previously developed, while improving on aspects that we find less desirable. The selected products and projects to consider before our development began were chosen to collectively encompass all aspects of our design or as a reference for similar technology used in a different context. We believe that our product incorporates and improves upon the relevant aspects of the products and projects described below, while minimizing the pitfalls of the collective set.

4.1.1 Xbox Controller

Out of all similar products that will be discussed, the Xbox controller and all similar gamepad controllers are by far the most common input devices. The reason for this is the proliferation of gaming consoles that require these controllers and their wide range of uses due to their relatively diverse input types, especially when their cost-to-input ratio is considered. Most gamepads, including the Xbox controller, maintain two analog inputs via small joysticks and a “D-pad” type input, any combination of which would allow the user to bind the axes of the collective and throttle to inputs on the controller. The largest benefit of this device type is the extremely high availability and low cost compared to any flight-sim oriented device. However, while maintaining a low cost and high modularity across multiple games and simulation, these controllers offer near-zero physical simulation benefits. The physical inputs of one of these devices has no corresponding real-world equivalents and are generally only used by casual flight-sim users.

4.1.2 X52 H.O.T.A.S

The X52 Hands on Throttle-And-Stick (H.O.T.A.S) by Logitech is a popular single axis throttle and multi-axis flight stick that, like the Xbox controller, will be used as a reference for the larger class of similar products. While being sold as the pair of throttle and stick, for our purposes we will exclusively be considering the throttle portion of the pair due to the fact that it is the most widely used analog to our desired end product.

For the majority of household flight-sim users, the X52 and similar devices act as a relatively affordable and versatile H.O.T.A.S set that can easily be adapted to a wide range of airplane and helicopter simulations. This is achieved by maintaining a generic shape and control scheme, thus sacrificing absolute realism for modularity. Due to the fact that the only requirements for replicating the throttle and collective controls of a helicopter are the ability to provide two separate axes inputs, these devices fill the role but lack the physical realism due to their airplane focused shape, axis direction, and range of movement. Our design addresses all three of these aspects as the physical implementation of our collective lever and replaceable heads are strictly designed around actual aircraft controls. One aspect that we consider to be important that the X52 does well in particular is the variable resistance within its main axis. The variable resistance integration allows the user to set the amount of force required to move the throttle, thus allowing for greater comfort or a more accurate feel when operating specific aircrafts. While our design will likely differ greatly in implementation, the ability to adjust friction on the collective lever to better meet user expectations was identified as a stretch goal for our project and implemented in the final product.

4.1.3 Komodo Simulations Collective

The Komodo simulations collective line of products is the gold standard for the flight-sim community when it comes to the simulation of helicopter collectives. The line includes multiple aircraft types with custom collective heads and levers that accurately depict the real-world control schemes for those aircrafts. The Komodo collectives rely upon a central base unit with variable friction adjustment and a separate simulated lever and head that connects via a common connection to the base. The Komodo base unit is also one of the highest accuracy units available on the market with a 12-bit resolution output for the collective angle lever.

While being very accurate to the actual helicopter that the collective is from, there are a few downsides related to their product. One such feature that is lacking in their design but is implemented in our design as an advanced feature is the ability to connect the base unit wirelessly to the host computer. This ability drastically increases the ease of use and setup when compared to a device that requires a USB cable connection. Another aspect that is more glaring than the lack of wireless connection is the overall cost of the unit. The base unit and collective lever with head are sold separately with the base unit priced at £480 and various lever/head modules priced between £430 and £495. Thus, the total cost to simulate a single helicopter is approximately $1,300 when converted to USD, with an added cost of approximately $640 for each additional helicopter lever/head unit. While the Komodo is highly accurate when compared to the actual helicopter controls, we believe the price point of the device is ultimately too large of a hindrance for the vast majority of flight-sim enthusiasts. It is for this reason that we strived to produce a final product that supplies a similar simulation experience to the end user but outperforms the Komodo collective in final product pricing. We experienced cost savings in many areas of the development process by fabricating many of our own components and limiting our reliance on commercial off-the-shelf (COTS) components. This tradeoff is effectively trading design time for an end product that is able to be produced for a less expensive price.

4.1.4 Angular Position Project

The Angular Position (AP) Project was a senior design project at UCF conducted in 2018. The goal of the project was to measure angular data from an encoder that was attached to a shaft, determine if the angle was within preset bounds, and wirelessly transmit the results of this test to an external programmable logic controller (PLC). While this set of steps was ultimately to monitor and control an industrial piece of equipment, the overall concept of measuring angular change via an encoder and sending the results wirelessly is very similar to the methodology that was implemented in our own project.

In addition to measuring the angle via an encoder and sending wireless data, the AP Project also includes a zeroing feature in which a button can be pushed to set the initial “zero” position of the encoder. This allows the device to measure angular position from this reference angle. While not exactly equivalent, our product has a similar calculation phase prior to any output value being sent to the host computer due to our need for calibration. Unlike a simple single point zero implemented in the AP Project, our design requires two-point calibration where all output values from the collective base unit are scaled to return values 0 to 1023 between the two calibration points. While the software is likely to differ dramatically due to this difference, the AP Project did offer a great data point for some of the physical elements, such as robustness of the shaft to encoder coupling.

4.2 Rotation and Angle Sensors

Two of the main components of a helicopter collective are the collective lever itself and the twist throttle which is attached to the collective lever. Both components involve detecting the angular position of the control around a given axis and interpreting that position as either blade pitch angle (for the collective lever) or engine RPM (revolutions-per-minute, for the twist throttle). In order to measure the angular position of these two components, a variety of sensors are available. For our project, the main concerns when choosing an angle sensor were cost, accuracy, granularity (how small of a difference in angular position can be measured), and calibration.

4.2.1 Rotary Potentiometers

A rotary potentiometer is an extremely simple device consisting of 3 terminals and a rotating shaft. As the shaft rotates, a rotating wiper inside the potentiometer moves across a plane of resistive material which produces a variable resistance. By applying a known supply voltage across the potentiometer, the voltage drop across the potentiometer can be used to extrapolate the position of the rotating shaft and therefore the angular position of the attached component. As potentiometers are essentially resistors, they are extremely simple to use, manufacture, and are widely available at a very low cost. However, potentiometers also have significant disadvantages. Rotary Potentiometers rely on physical contact between the wiper and the resistive surface, which can wear out over time. Also, the resistive element is not necessarily completely uniform in its resistance throughout the entire plane, and therefore the same angular shift of the wiper may result in varying magnitudes of voltage drop. Finally, due to the physical nature of the device and the necessity of converting an analog voltage to a digital output, the resulting angular position is less accurate and can be extremely noisy.

4.2.2 Rotary Encoders

A rotary encoder is a generic term for a device which measures the rotation of a small shaft affixed to an encoder. Conceptually, by measuring the rotation of the shaft, it is possible to determine rotation speed, rotation direction, or angular position depending on the desired application. For our project specifically, our main concern is with angular position as the position is converted to a game input rather than the speed or direction of rotation. The two main types of rotary encoders are incremental rotary encoders and absolute rotary encoders, which will be explored further in the following sections.

4.2.2.1 Incremental Rotary Encoders

Incremental rotary encoders are a specific kind of rotary encoder which produce an output when a change in angle occurs. As the shaft rotates, the incremental rotary encoder will generate output pulses indicating that a certain degree of angle has been rotated. Typically, incremental rotary encoders also allow the direction of rotation to be measured. However, the exact angular position of a rotary encoder is not measurable at any given time. This means that in order to accurately measure the current angular position of the shaft, the incremental encoder must be "calibrated" by beginning to measure the number of pulses generated starting from a fixed, known reference point. This "calibration" is also lost when power is lost to the device, as the reference point measurement will be lost. From a practical perspective, this means that each time the collective operator turns on the device, they must complete a calibration sequence by pushing the collective lever and twist throttle all the way to a zero position prior to flight. The main advantages of incremental rotary encoders are their availability and cost. Incremental rotary encoders are relatively simple to manufacture, widely available, and inexpensive. The main downsides of incremental rotary encoders are as mentioned earlier; the angular position must be maintained in software and must be recalibrated after a power cycle.

4.2.2.2 Absolute Rotary Encoders

Absolute rotary encoders are a specific kind of rotary encoder which output the current angular position of the shaft at any given time. This position is determined via some sort of "code" which is physically encoded onto a disc that rotates along with the shaft. The actual implementation of how this code is read are primarily via magnetism, capacitance, or optical coupling. The main advantage of absolute rotary encoders is that they have "memory"; that is, the current position of the encoder can be determined at any given time and its state is maintained through power loss or program failure as the position is physically determined, not programmatically. This provides a significant advantage for our project, as our main concern is with angular position and the ability for an absolute rotary encoder to maintain its position measurement throughout power cycles reduces the need for the user to constantly recalibrate the device. Also, absolute rotary encoders are available with high resolutions and accuracy which allow for extremely precise movements of the collective lever and twist throttle. The main downside of absolute rotary encoders is in their cost; in order to achieve higher accuracy and measure an absolute position, they are significantly more complex to manufacture and are therefore significantly more expensive than potentiometers or incremental rotary encoders in general.

4.2.2.3 Rotary Encoder Resolution and Pulses Per Revolution (PPR)

The resolution of a rotary encoder refers to the smallest angle or change of angle mechanically measurable by any given rotary encoder. While there are multiple metrics used to describe this resolution, one of the most common and most relevant to our project in particular is PPR, or pulses-per-revolution. For an incremental rotary encoder, PPR refers to the number of high pulses in a single full revolution of the encoder. For an absolute encoder, it refers to the number of measurable bits, and therefore number of measurable angular positions, in a single revolution. By knowing the PPR of any given rotary encoder, the minimum measurable change of angle is able to be easily calculated; for example, if a rotary encoder has 16 pulses-per-revolution, then a change in angle is only measured every

$\frac{360°}{16} = 22.5°$ of shaft rotation. For our project in particular, our goal was to be able to measure, at a minimum, a 1° change of rotation angle. This implies that any rotary encoder would require a PPR of at least 360 in order to meet our project requirements.

4.3 Wireless Communication

One of the major goals of this project was to create a portable helicopter simulation solution which can be used in a variety of situations. In order to achieve this, one of the distinguishing features of our helicopter collective is in its ability to connect wirelessly to a computer rather than be tethered by a physical connection. This flexibility allows the end user to use the collective in a variety of situations and to accommodate changes in simulation equipment location, mounting solution, or connected device. In order to achieve this, a wireless connection between the collective lever itself and the computer running the flight simulator software is necessary and the inputs must be translated from physical switch and button inputs to virtual gamepad inputs the game can recognize and utilize.

4.3.1 Wireless Technologies

In order to implement wireless connectivity, it was first necessary to investigate the variety of wireless technologies commonly in use today in order to determine which one would best suit our needs. Our major concerns when looking for a suitable wireless technology were cost of implementation, ease of implementation, transmission range, data transmission rate, and power consumption. All of these factors are important as it is necessary for the collective lever to constantly transmit inputs to the flight simulator while maintaining a decent runtime on battery power, remaining relatively low cost, and being portable enough to be used in a variety of installations without hassle. Thus, the aforementioned factors were used to decide which wireless technology to utilize in developing the collective lever module.

4.3.1.1 Infrared

Infrared communications are one method of transmitting information wirelessly through the use of Infrared light. Infrared light is a form of electromagnetic radiation which exists within the 0.7 µm to 300µm wavelength range which is similar in wavelength to visible light. Generally, commonly available Infrared links operate in the 780 nm to 950 nm wavelength range. Infrared transmitters and receivers are commonly available and relatively cheap and are therefore an attractive option for basic wireless communications. Infrared is also generally unregulated and is therefore convenient and easy to implement without worrying about violating regulatory standards. However, care must be taken to ensure that the Infrared light is utilized in a manner which is eye-safe, as Infrared radiation that is too intense or too focused has the possibility of causing permanent eye damage. To combat this, most commonly available LED-style Infrared links emit Infrared radiation over a large surface area which minimizes the risk of eye damage.

Infrared light has predictable properties similar to those of visible light; being absorbed by dark surfaces, reflected by light surfaces, and some wavelengths of Infrared can pass through transparent materials such as glass. This means that Infrared communication is intuitive to implement, and Infrared devices are simple to manufacture. However, this is also one of Infrared's greatest downfalls as a wireless communication medium; Infrared light is easily blocked by many surfaces and therefore generally requires line of sight in order to directly communicate. Furthermore, other sources of infrared light such as the sun and household lighting, amongst a variety of others, can act as interference which can disrupt Infrared communications entirely. Generally, Infrared links are either directed or undirected. Directed Infrared links can provide greater range and power at the receiver but must be more precisely oriented and aimed toward the receiver. Non-directed Infrared links are more convenient to implement due to the lack of aiming requirements but have a generally shorter range and less power at the receiver.

Infrared communication is simply a method of transmitting raw data over a wireless link. From the perspective of our project's requirements specifically, this means that an additional unit would be required which connects to a computer, receives Infrared communication and converts it into an understandable format for the computer, and then performs the communication. This would involve additional hardware and software which other wireless technologies may not require.

4.3.1.2 Wi-Fi Direct (IEEE 802.11)

IEEE 802.11 standards describe standards for implementation of WLAN (Wireless Local Area Networks). IEEE 802.11 focuses on the Physical Layer and MAC (Medium Access Control) layer implementation of WLANs and has evolved throughout the years from only about a 2 Mbps data rate at its conception to modern data rates of 540 Mbps and above with newer iterations on the standard. The Wi-Fi Alliance was formed as a nonprofit organization to manage the certification of products which adhere to 802.11 standards and nowadays the terms Wi-Fi and 802.11 have become almost synonymous with each other. Wi-Fi is an extremely common wireless communication medium today, and typically is implemented via use of a wireless access point (AP) which provides a typical communication range of about 30-50 meters.

Wi-Fi Direct is a relatively recent evolution of Wi-Fi technology which allows Wi-Fi enabled devices to connect to each other directly without the need for an intermediate access point. This allows for direct data transfer between devices by creating their own local Wi-Fi networks. Wi-Fi direct functions using both the 2.4 GHz and 5 GHz frequency bands. Wi-Fi direct has the advantage of allowing for extremely high data rates up to 25 Mbps as well as a relatively large range, with some Wi-Fi direct implementations allowing for connections up to 200 meters. This makes Wi-Fi direct an extremely advantageous option for applications requiring medium range and large amounts of data transfer, and is used in devices such as phones, printers, and game consoles.

The main downsides of Wi-Fi and Wi-Fi direct are complexity, cost, and power consumption. Wi-Fi connections typically require more configuration to establish connections and may be more complex to implement than other simpler options. Also, as Wi-Fi can support relatively large ranges and high data rates, it consumes more power than other options which is a downside when attempting to create portable, battery-powered devices.

4.3.1.3 Zigbee

Zigbee is a wireless communications standard designed for establishing PANs (Personal Area Networks) and for creating IoT (Internet of Things) devices. Zigbee is especially designed to be used in a mesh network fashion, with many Zigbee devices communicating with one another. Zigbee operates at 2.4 GHz (global), 915 MHz (Americas), and 868 MHz (Europe) at established data rates of 250 Kbps (2.4 GHz), 40 Kbps (915 MHz), and 20 Kbps (868 MHz). The typical range of Zigbee based communication devices is between 10 and 100 meters, making it optimal for short range communication.

As Zigbee is specifically designed for IoT applications, it is a very low power wireless communication solution. Also, the focus on a mesh-based networking solution allows for extremely convenient addition and removal of Zigbee devices as the mesh-network eliminates the need for direct connection to a central node. However, this comes at the cost of a relatively short transmission range and a low data throughput maxing out at 250 Kbps. While this data rate is more than sufficient for items such as sensors, smart home devices, and simple PAN applications, it restricts Zigbee's effectiveness as a more robust wireless communication solution.

4.3.1.4 Bluetooth Low Energy (BLE)

Bluetooth Low Energy is a wireless communication standard that evolved from Classic Bluetooth to specifically target low power applications. Bluetooth Low Energy functions in the 2.4 GHz frequency band with 40 channels with 2 MHz spacing between each channel. The Bluetooth Low Energy specification is maintained by the Bluetooth Special Interests Group, an organization which publishes Bluetooth specifications and certifies products which implement Bluetooth functionality. Bluetooth Low Energy offers data rates between 125 Kbps and 2 Mb/s, with transmission ranges up to 100 meters, but typically around 10 meters. Bluetooth Low Energy uses only a fraction of the power of Classic Bluetooth while still offering decent data rates.

Bluetooth Low Energy communication functions by sending small bursts of data between a Central (Client) device and a Peripheral (Server) device. To achieve this, a Bluetooth Low Energy device implements a variety of Profiles, which are specifications for how a certain class of device works. Bluetooth Low Energy data is transmitted as chunks of data known as attributes via the GATT (Generic Attribute) Profile. The GATT profile transmits data via Characteristics, Descriptors, and Services. Characteristics contain a single value and themselves contain multiple Descriptors which are attributes that describe the characteristic. A Service is a collection of characteristics which define a specific application of Bluetooth Low Energy and are established by the Bluetooth SIG in the official Bluetooth Low Energy Specification. Profiles implement one or more services in order to fulfill their purpose.

The main advantages of Bluetooth Low Energy are cost, compatibility, and power consumption. Bluetooth Low Energy modules are readily available and cheap compared to other alternatives. Bluetooth Low Energy is extremely common and is readily available in most consumer devices such as phones, computers, gaming devices, and many others. Also, Bluetooth Low Energy is extremely power efficient while still boasting a decent data transmission rate which makes it an optimal choice for general-purpose short range wireless communication via a battery powered device. Finally, the Bluetooth Low Energy Specification inherently supports HID via the HID over GATT profile, which allows Bluetooth Low Energy devices to be recognized as input devices such as mice, keyboards, or gamepads, a fact which is extremely advantageous to our project in particular.

4.3.2 Bluetooth Modules

Determining which Bluetooth module to use is an important decision as it affects circuit design, transmission range, data transmission rate, and overall cost of the collective module. There exists a variety of commercially available Bluetooth modules which each have various advantages and disadvantages. The goal of the following investigation was to select a Bluetooth module which was compatible with HID, easy to implement, low cost, and power efficient. These factors are important in adhering to our project goals and requirements; therefore, they were be used as the major criteria during module selection.

4.3.2.1 RN42

The RN42 is a Class 2 Bluetooth module designed by Roving Networks, a company which has since been absorbed by Microchip. The RN42 is a Bluetooth 2.1 qualified module and supports Bluetooth 2.1 + Enhanced Data Rate, which features an improved data transfer rate compared to standard Bluetooth 2.1. The RN42 is a drop-in module with an integrated PCB antenna which can be integrated easily into a variety of product designs to add Bluetooth functionality. The RN42 is pre-certified to be compliant with FCC regulations and is also compliant with government regulations in Canada, Europe, Australia, New Zealand, Japan, Korea, and Taiwan. The RN42 features a built-in crystal oscillator and voltage regulator which allow for ease of integration into an existing design. The advertised range of the RN42 is up to 10 meters and it contains its own power amplifier. The RN42 also has low power consumption, with 26 µA in sleep mode, 3 mA when connected passively, and 30 mA when actively transmitting, along with configurable low power modes.

The RN42 can communicate over UART and USB, with a rate of 240 Kbps (slave) and 300 Kbps (master) over UART, and 1.5 Mbps sustained over USB in HCI mode. One of the main advantages of the RN42 is that when communicating over UART, an ASCII command interface is pre-programmed onto the module which makes configuring and operating the module simple. This ASCII command interface significantly reduces the programmer load to implement the RN42 as the commands are human-readable and simple.

By far the most advantageous aspect of the RN42 is the inclusion of embedded Bluetooth profiles, specifically the Bluetooth HID (Human Interface Device) profile. As our project must be connected to a computer in order to function, the pre-configured HID profile offered by the RN42 significantly reduces the amount of configuration and programming required in order to properly connect and communicate as an input device to a computer. The RN42 also implements other embedded Bluetooth profiles such as GAP and SPP. Also, as the RN42 is a Bluetooth 2.1 certified device, it can take advantage of SSP (Secure Simple Pairing), a new method of managing Bluetooth pairing and connections which improves security and ease-of-use from Bluetooth 2.0.

The major downsides of the RN42 are related to its age. While the RN42 is extremely popular especially in the hobbyist and prototyping space, it is now considered by Microchip to be “Not recommended for new designs”. This means that firmware updates, documentation updates, and support from Microchip are not as prevalent as with their more modern offerings. Furthermore, the RN42 utilizes Bluetooth 2.1 + EDR, which is now considered a deprecated version of the Bluetooth specification. This means that new product designs utilizing Bluetooth 2.1 + EDR will not be qualified by the Bluetooth SIG and the specification is no longer officially supported. This fact would significantly inhibit the market prospects of any device created using the RN42 and would likely require significant product redesign before going to market.

4.3.2.2 RN4870/71

The RN4870/71 is a BLE (Bluetooth Low Energy) module designed by Microchip that adheres to the Bluetooth 5.0 standard. The RN4870/71 is available as a pre-shielded module with an integrated antenna which allows a product to easily add in Bluetooth Low Energy functionality. The main difference between the RN4870 and the RN4871 is simply that of size and therefore range, with the RN4870 boasting up to a 50-meter range versus the RN4871 only advertising a 10-meter range. The RN4870/71 is certified for use in the United States, Canada, Europe, Japan, Korea, Taiwan, and China, making it an apt module for use in products which are intended to be sold internationally.

The RN4870/71 has a complete on-board Bluetooth Low Energy stack which allows it to be used in a variety of applications. While the RN4870/71 has integrated GAP, GATT, SM, and L2CAP profiles, the RN4870/71 also allows the developer to integrate up to 5 public or 4 private custom Bluetooth Low Energy services. This configuration and operation take place over a UART connection, and the RN4870/71 includes an ASCII command interface over UART which simplifies the development process. The RN4870/71 also includes a pre-defined simple scripting engine which allows the module to be potentially used without a host controller for simple Bluetooth applications. Also, the RN4870/71 includes both a UART Transparent Service which allows for extremely quick and simple direct UART Serial Data passthrough as well as a Beacon Private Service for Bluetooth Low Energy Beacon services.

The main advantages of the RN4870/71 are in its flexibility and ease-of-use. As it is a drop-in module that is pre-qualified by the Bluetooth SIG and pre-certified by the FCC as well as many other regulatory government agencies around the world, it is an optimal candidate to be included in a product design without worrying about standards compliance. The RN4870/71 is also Bluetooth 5.0 certified, which guarantees long product relevance and support as well as reaps the benefits of more recent Bluetooth versions. Also, the ASCII command interface as well as the Transparent UART Service make configuration, operation, and testing of the unit much simpler than some other Bluetooth modules. Finally, the availability of the RN4870/71 on Microchip PICtail development boards for prototyping would allow for easier development and testing.

The main disadvantages of the RN4870/71 are the limits of the ASCII command interface and custom services, as well as the lack of documentation and community support. While the ASCII command interface offers many benefits, it also restricts the capabilities of what the developer can do. For example, the RN4870/71 does not support HID over GATT out of the box, and some of the required configuration to implement HID over GATT as a custom service may be difficult or impossible via the ASCII interface. Also, the RN4870/71 has only minimal community support and a medium amount of documentation from Microchip themselves, which could pose troublesome if issues occur during implementation.

4.3.2.3 BM70/71

The BM70/71 is another Bluetooth Low Energy module designed by Microchip that is qualified for the Bluetooth SIG 5.0 specification. Overall, the BM70/71 is extremely similar in nature to the aforementioned RN4870/71. The BM70/71 is available as a pre-shielded complete Bluetooth module with built-in antenna which allows for drop-in Bluetooth Low Energy functionality to a variety of products and designs. The BM70/71 is certified by the FCC, ISED, MIC, KCC, NCC, and SRRC. The BM70/71 is also RoHS compliant. Similar to the RN4870/71, the main difference between the BM70/71 is in size and connection range, with the BM70 being generally larger and having a 50-meter range and the BM71 being generally smaller and having a 10-meter range.

The BM70/71 supports communication over UART as well as a Transparent UART Data service which allows for simple UART Serial Data passthrough over Bluetooth Low Energy. The BM70/71 also has a Precision Temperature Sensor, an ADC, and has 18 GPIO pins (BM70) or 9 GPIO pins (BM71), as well as an integrated 32 MHz crystal. These peripherals allow the BM70/71 to be used for simple Bluetooth Low Energy Applications without the need of a host or allow some GPIO functions to be offloaded from the host to the BM70/71. The BM70/71 also supports two low power modes and has embedded BLE profiles such as GAP, GATT, SMP, and L2CAP.

The main advantage of the BM70/71 is that instead of an ASCII interface, the BM70/71 supports byte-based communication over UART which allows for much finer configuration and operation of the module. This allows for the creation of custom GATT services, which is especially important as HID over GATT must be implemented in order for the connected computer to recognize the Bluetooth device as a controller. However, this fact is also somewhat a disadvantage, as the ease-of-use of an ASCII interface is lost, and more low-level configuration and operation is required.

4.3.2.4 Initial Bluetooth Module Selection

After considering the aforementioned options, we initially selected to move forward with the BM71 module as our primary Bluetooth connection module. Our main factors used to select a module were HID-compatibility, ease of implementation, and Bluetooth version. The RN42 meets and exceeds the first two factors by far, as it has extensive community support, an easy-to-use ASCII interface, is pre-certified by the FCC, and has built-in HID support over Bluetooth Classic. However, the RN42 utilizes Bluetooth 2.1 + EDR, which is a legacy version of Bluetooth that is no longer supported and new products utilizing the standard will not be qualified by the Bluetooth SIG. For this reason, we were forced to choose another module which is more modern and up to date in its Bluetooth qualification. The RN4870/71 would be the next easiest to implement and has a current version of Bluetooth (BLE based on the Bluetooth 5.0 standard), but due to the limitations of the ASCII interface we found that HID-over-GATT may be difficult or impossible to implement. Therefore, we chose the BM71 as while it is more complex, it allows for complete configuration of the Bluetooth Low Energy stack via a byte-based command set and supports HID-over-GATT, while also being certified for Bluetooth Low Energy under the Bluetooth 5.0 specification.

4.3.2.5 Final Bluetooth Module Selection

While we initially selected the BM71 to act as our primary Bluetooth module, we ended up implementing the BM70 in our final design. This change was made primarily due to part availability, with the BM71 going out of stock while the BM70 was readily available. However, as mentioned earlier, the BM70 and BM71 are essentially the same module except for the BM70 being physically larger, having a larger Bluetooth range, and having extra GPIO ports. Thus, all our design decisions which were based off the BM71 remained unchanged and valid, and the only significant change that was necessary was to swap the physical module in the PCB design.

4.4 Microcontroller

For this project, we determined that a microcontroller would be the most logical choice of logic controller platform to process and transmit the input from the collective unit to the flight simulator. We came to this conclusion due to the fact that our project is not safety-critical and has no critical real time requirements, factors which would possibly require a FPGA or similar device. Also, we determined that due to the relatively simple nature of listening for physical inputs and performing simple communications that a single-board computer would be overkill in terms of processing requirements. Microcontrollers provide adequate processing speeds, are generally low cost, and are small and easily integrable into the space of the collective unit. Our main considerations when researching microcontrollers were processing speed, power consumption, number of GPIO (general purpose input output) pins, and cost.

4.4.1 Atmega2560

The Atmega2560 is a microcontroller developed by Microchip that has been popularized through its use on the Arduino Mega 2560 and similar development boards. The Atmega2560 has a configurable clock speed up to 16 MHz, 256 KB of flash memory, 8 KB of SRAM, 4 KB of EEPROM, 86 GPIO pins, and other common microcontroller peripherals such as a 16-channel 10-bit ADC, six timers, PWM, and four USARTs. The Atmega2560 also claims to have low power consumption needs and the configurable clock speed allows the performance to power consumption to be adjusted.

The main hardware advantages of the Atmega2560 related to our project in particular are the large number of GPIO pins as well as the configurable clock speed and 4 KB of EEPROM. As each collective head will have a variable number of inputs and therefore a variable number of GPIO requirements, the Atmega2560 provides adequate leeway to accommodate a variety of collective heads and overall system implementations. The configurable clock speed would allow us to only run the microcontroller at the minimum required speed in order to reliably process inputs while also maintaining low power consumption if so desired. Finally, the 4KB of EEPROM is desirable as it opens up the possibility of storing calibration data between power cycles so as to provide a more seamless, pick-up-and-play user experience.

Another major advantage of the Atmega2560 is that, as mentioned previously, it is available on the Arduino Mega 2560 development board. This is a significant advantage as it allows for quick and easy prototyping and development using the Arduino development environment and libraries, as well as a plethora of community support and online tutorials due to the popularity of the Arduino platform. Furthermore, our group members are familiar with Arduino from previous hobby projects, making this capability even more desirable.

4.4.2 MSP430FR6989

The MSP430FR6989 is a microcontroller produced by Texas Instruments that we considered for use in this project. The MSP430FR6989 has a configurable clock speed up to 16 MHz, 128 KB of flash memory, ultra-low power consumption, five 16-bit timers, a 12-bit ADC, multiple USCI modules supporting UART, I^2C, and SPI, 83 GPIO pins, and an integrated LCD driver. The MSP430FR6989 is available on the MSP-EXP430FR6989 LaunchPad Development Kit board for prototyping purposes.

The MSP430FR6989 is similar to the Atmega2560 in terms of capabilities; it sports a configurable clock up to 16 MHz, has 83 GPIO pins in its largest package, and has similar peripherals. The main advantages of the MSP430FR6989 in particular are its ultra-low power consumption and its familiarity. The MSP430FR6989 LaunchPad was utilized in our course curriculum, and it can be programmed using Code Composer Studio, which we are familiar with. Also, the MSP430FR6989 has a built-in LCD Driver which could be useful for driving a display on the collective unit itself.

4.4.3 ESP32-WROOM-32SE (ESP32-D0WD-V3)

The ESP32-WROOM-32SE is an all-in-one module solution from Espressif which integrates an ESP32-D0WD-V3 microcontroller unit onto a small PCB with an embedded antenna and is shielded and FCC certified so it can be easily implemented into any design. The ESP32-D0WD-V3 has a dual core, Xtensa 32-bit microprocessor which is configurable up to 240 MHz. The ESP32-D0WD-V3 also has 448 KB of ROM, 520 KB of SRAM, 32 GPIO pins, and supports Wi-Fi 802.11b/g/n and Bluetooth V4.2 BR/EDR and Bluetooth LE (Low-Energy). It also has various peripherals such as an ADC, support for UART, SPI, and I^2C, and a 40 MHz crystal oscillator.

The ESP32 series in general is extremely popular and there are existing libraries to program ESP32 series devices using the Arduino development environment, which would make prototyping more efficient. Some of the main hardware advantages of the ESP32-WROOM-32SE are the integrated Bluetooth capability, PCB antenna, and FCC certification. This would allow the module to drop-in to our design without the need for any additional Bluetooth chip and is a significant advantage. However, the processing capabilities of the ESP32 are overkill for the simple processing that is required for the collective and will therefore consume more power. Finally, the ESP32-WROOM-32SE only has 32 GPIO pins, which would significantly restrict the kinds of collective heads that could be effectively implemented.

4.4.4 Texas Instruments CC2640

The Texas Instruments CC2640 is a Bluetooth Wireless MCU with an ARM Cortex M3 processor that can be configured up to 48 MHz. The CC2649 has 128 KB of flash memory, 20 KB of SRAM, 31 GPIO pins, four timers, a 12-bit ADC, UART, SPI, and I^2C support, low power consumption, and integrated Bluetooth 4.2 Low Energy support. The main advantages of the CC2640 are that it has Bluetooth Low Energy built-in, which would eliminate the need for an additional Bluetooth chip. However, unlike the ESP32-WROOM-32SE module, the CC2640 is a standalone SoC, and therefore would require an external antenna of some kind. Also, the CC2640 only has 31 GPIO pins, which would significantly limit the kinds of collective heads that could be implemented in our project. Similar to the MSP430FR6989, however, the CC2640 is a Texas Instruments product and can therefore be programmed in Code Composer Studio and is a member of TI's SimpleLink product line which provides examples, software tools, and documentation related to developing with the CC2640.

4.4.5 Microcontroller Comparison

When comparing the abilities of the researched microcontrollers, we first established the core attributes that would ultimately affect our previously stated requirement specifications. While many of the microcontrollers researched have numerous differences, we have determined that many of them are minute and would have a negligible effect on our end product. However, in an effort to compare the microcontrollers across standards that are important to us and that all of the microcontrollers possess, the attributes of clock rate, flash memory size, GPIO pin availability, timer integration, power consumption, and cost were all considered. Other subjective aspects for the microcontrollers, such as previous experience and product knowledge, will also be considered when making the final decision.

4.4.5.1 Clock Rate Comparison

The microcontrollers clock rate as a vital component to our end product was determined based on the fact that many aspects of the design require computation, while maintaining a high enough polling rate of user input to ensure smooth operation and feel when operating the collective. With this goal in mind, we are able to consider the researched microcontrollers, compare their clock rates, and determine which of them would be sufficient for our design. As the number of instructions able to be performed on the microcontroller is directly correlated with the actual frequency of the clock, we will be weighing higher clock rates as more positive in our final choice.

Table 6 – Microcontroller Clock Frequency Comparison

Microcontroller	Clock Freq.
Atmega2560	16 MHz
MSP430FR6989	16 MHz
ESP32-WROOM-32SE	240 MHz
CC2640	48 MHz

4.4.5.2 Flash Memory Comparison

Flash Memory is the on-board nonvolatile memory used to store data between power cycles, while maintaining the ability to be rewritten many times over. When considering aspects of our project that include the ability to store user settings, sensor calibration data, and collective head profiles, it is imperative that the chosen microcontroller contain enough flash memory to store necessary information for the operation of the collective. For our purposes, we will be considering flash memory available to the programmer, excluding blocked partitions, as we are ultimately concerned with the ability to store the collective's information. For our application, a larger flash memory size would allow for more configuration information and a greater variety of collective head profiles.

Table 7 – Microcontroller Flash Memory Comparison

Microcontroller	Flash Memory Size
Atmega2560	256 KB
MSP430FR6989	128 KB
ESP32-WROOM-32SE	4 MB
CC2640	128 KB

4.4.5.3 GPIO Comparison

Due to the fact that our product is centered around the recreation and simulation of real-world controls, it is imperative that the chosen microcontroller maintain enough General-Purpose Input/Output (GPIO) pins to accommodate all required inputs. When comparing the GPIO pin availability and requirements, all possible inputs and outputs must be considered, including battery voltage sensing, the information display, axis position sensing, and all collective head inputs. We believe that the total available GPIO pins will likely be the limiting factor of the specific collective heads we are able to replicate, due to the fact that many collective heads in real life maintain a large number of buttons, toggles, and hat switches. It is for this reason that we will be heavily weighting GPIO pin availability in our final assessment.

Table 8 – Microcontroller GPIO Pin Comparison

Microcontroller	GPIO pins
Atmega2560	86
MSP430FR6989	83
ESP32-WROOM-32SE	32
CC2640	31

4.4.5.4 Timer Comparison

As the polling of human input and the sending of the updated values will need to be based on a consistent and measurable rate, the existence of an accurate timer will be a requirement for any microcontroller we are considering. Some user input will likely be based upon a press-and-hold mechanism where the duration of the press is important, which will require our design to accommodate the timing of events. This event timing would benefit from timers that contain larger memory size, allowing for longer or more accurate time measurement.

Table 9 – Microcontroller Timer Comparison

Microcontroller	Timers	Compare Mode
Atmega2560	2 x 8-bit, 4 x 16-bit	Yes
MSP430FR6989	5 x 16-bit	Yes
ESP32-WROOM-32SE	2 x 64-bit	Yes
CC2640	8 x 16-bit, 4 x 32-bit	Yes

4.4.5.5 Power Comparison

Considering that one important aspect of our design is to ensure portability and to maintain a runtime that is sufficiently long, it is very important that our chosen microprocessor is power efficient. Due to the fact that the normal state of our design will be sending information to the computer, it can be assumed that the MCU will need to be active for the majority of the power-on time of the device. This is because despite no change in the human input, the values of the controller will still be expected by the simulator software regularly. It is for this reason that the comparison of the microcontrollers will attempt to exclude low-power mode settings and will only concentrate on the power consumption when the CPU is in active mode. In this case, the lower amperage that the microcontroller requires the better it suits our needs.

Table 10 – Microcontroller Power Comparison

Microcontroller	Amperage
Atmega2560	20mA
MSP430FR6989	16mA
ESP32-WROOM-32SE	68mA
CC2640	4mA

4.4.5.6 Cost Comparison

As a final component to consider in deciding upon a microcontroller, cost ultimately will be used as a metric to determine if the benefits or detriments of a particular microcontroller are worth compromising for. As this design is meant to be a more affordable option for flight simulation, it is important to pick a microcontroller with a cost that would not hinder the scalability of production. The use of a specific device is ultimately a critical factor as they could all be used to satisfy our needs; thus, the cost should influence the final choice to pick a microcontroller that fits our needs while being budget-friendly.

Table 11 – Microcontroller Cost Comparison

Microcontroller	Cost
Atmega2560	~ $6
MSP430FR6989	~ $9.50
ESP32-WROOM-32SE	~ $3
CC2640	~ $7

4.4.5.7 Microcontroller Selection

After identifying and comparing the key features we would require in our final product, it is vital to consider all data points shown above and make an educated selection based on our findings. The most simplistic method of selecting the microcontroller for our project would be to rank each microcontroller in each feature individual comparison, assign a value between one and four for each microcontroller, and finally average the results per microcontroller and select the microcontroller with the smallest average. However, considerations must be made for specific aspects of the microcontrollers that we may find more vital than others and for microcontrollers that fall outside an acceptable boundary in any of the given aspects. In order to select our microcontroller, the naive approach mentioned above will be conducted first, then specific aspects from the top results will be considered.

In the table below, each microcontroller is ranked based on each comparison made previously. The lower the final average across all aspects considered, the more appropriate we consider the microcontroller for our project. In the case of a tie the same value will be placed on both microcontrollers. In the case of timers, which is a more subjective measurement than the other attributes due to the multi-dimensional differences, the values will be selected based on what we believe will be required for our final product.

Table 12 – Microcontroller Rankings

	Atmega2560	MSP430 FR6989	ESP32-WROOM-32SE	CC2640
Clock Rate	3	3	1	2
Flash	2	3	1	3
GPIO	1	2	3	4
Timers	2	3	3	1
Power	3	2	4	1
Cost	2	4	1	3
Final Average	**2.16**	**2.83**	**2.16**	**2.33**

As seen in the chart above, the Atmega2560 and ESP32-WROOM-32SE are tied for being the most likely candidates based on our criteria using a naive method of selection. Now, when considering the more particular properties of the aspects compared, we can do so with a base ranking in mind.

When selecting possible attributes of the microcontroller that may disqualify some microcontrollers all together, the original engineering requirement specifications and hypotheses about future design constraints are considered. In the case of clock rate, power consumption, timer availability, and cost, we believe that all selected microcontrollers will over-satisfy the requirements previously enumerated. However, based upon research related to our implementation of the collective heads and the software responsible for interpreting the specific head attached at a given time, we believe that GPIO pin availability and flash memory size could become limiting factors.

Flash memory is an important aspect as it will dictate the number of helicopter profiles and user settings which we are able to store, thus limiting the overall scalability of the project. For the core and advanced features that we have identified, we believe that the flash memory available on any of the microcontrollers will suffice. However, as we consider the stretch goals, it is possible that the flash memory of the MSP430FR6989 and CC2640 will be inadequate to store large number of collective head profiles.

The availability of GPIO pins is one of the largest considerations due to the demand for a large number of human inputs for some collective heads. With the GPIO pins on the microcontroller being divided between human interfaces, collective angle sensing, battery controller I/O, and the on-board diagnostic display, it is imperative that the selected microcontroller have the available pins to accommodate all peripherals. When researching and hypothesizing the number of GPIO pins needed, we estimated approximately 45 when using the most complex collective heads. It is for this reason, that both the ESP32-WROOM-32SE and CC2640 are both poor candidates for the final selection of our microcontroller.

After making these considerations, we determined that the only two possible microcontrollers that fulfill our core and advanced feature requirements are the MSP430FR6989 and the Atmega2560. When comparing these two alone, it is shown by both our flash memory concerns and the overall naive comparison technique that the MSP430FR6989 is outclassed by the Atmega2560 for our application. Thus, we chose the Atmega2560 for its GPIO availability, flash memory size, and relatively low cost as the main attributes, while still meeting requirements in all other categories.

4.5 User Interface

The user interface components discussed below encompass all of the aspects of our design that are able to be interacted with by the user and will act either as input to our microcontroller for processing or as a means of notification to the user. In the case of the tactile interface elements, these controls will be implemented inside each collective head, each of which will require a unique set of inputs based on their real-world counterparts. The main considerations we will make when selecting and ordering specific parts are their similarity to the real-world part, their quality of construction, and the cost of the component.

4.5.1 Toggle Switches

When considering requirements for toggle switches that we want to implement in our design, it is first necessary to identify the different aspects of toggle switches and which properties best suit our project's requirements.

One aspect that is switch dependent is the number of poles that the switch controls. In most cases, a toggle switch is either single or double pole. The number of poles refer to the number of individual circuits that the switch controls. A double pole switch is typically used in high voltage applications, in which the neutral and live wires are switched together to increase the safety factor. For our application this is not necessary as low voltages and currents will be used as input to the microcontroller through the switches.

The next aspect of the toggle switches that will be considered on a case-by-case basis is the switch function type. The switch function refers to the number of possible positions the switch has, the states of the circuit at each of those positions, and way in which the physical switch moves. The majority of toggle switches, which includes all switches being considered for this project, are either two or three position switches. These switches come in a large variety of possible states, typically denoted by the "on" or "off" condition at each physical location of the switch. For example, a three-position switch will typically be an "on-off-on", "off-on-off", or "on-on-on" switch. Due to the fact that we are able to make the switch selection and that we are writing the software responsible for determining which selection of the switch is made, any of these choices will suit our needs because simple bit-based checks can occur for the "on" or "off" state for each pin the toggle switches are connected to. The final toggle switch property we will focus on is the momentary type of toggle. Momentary toggle switches are typically found in a three-position switch configuration, where the middle position is neutral, the up and down switching mechanism acts to complete or break two distinct circuits, and the switch will automatically return to center upon release. These types of switches are typically used in the trimming of a value as the user can increment a decrement a value by flicking up or down on the switch.

As mentioned previously, the other aspects of the toggle switch we are concerned with are the likeness to the real-world switches and the electrical capabilities of the switch. For our specific application, the electrical properties are only a minor consideration because the vast majority of toggle switches will be able to control the relatively small current and voltage that we will be using as input to the microcontroller. However, the ability to select switches that are accurate facsimiles to the switches we are attempting to simulate is significantly more difficult. In an effort to select switches of similar appearance, the actuator length, shape, material, and panel mounting hardware is taken into account. This desire to properly replicate the switches from the real collective heads is tempered by the price and availability of the switches. In some cases, caps and heads for toggle switches will need to be designed and 3-D printed independently to meet the requirement of creating an accurate appearance.

While the specific shape and appearance of the selected toggle switches may differ greatly depending upon the specific collective heads attempting to be simulated, the "on/off" states of the toggle switch functions will remain consistent in an effort to reduce code complexity. In this case, we will be able to determine if a switch is toggled in a specific direction based on the high or low state of predefined pins. This is an important aspect to the electrical and software design aspect because all collective heads will share the same circuitry input to the main microcontroller.

4.5.2 Multi-Axis Input

For many helicopter collective heads, there exists a multi-axis input that allows the pilot to interface with aspects of the helicopter that require a 360-degree input, such as a two-axis trim or the panning of an exterior search light. In an effort to simulate this class of user inputs, two different solutions were explored.

4.5.2.1 Hat Switch

A hat switch is comprised of a single top plate that is typically affixed to four or eight microswitches in a centrally captive manner to allow the user the ability to press up to two of the switches with a single movement. These microswitches are evenly distributed in a circle around the top plate, such that the number of pressed switches and position of pressed switches can be used to determine an approximate angle at which the user pushed the top plate. The number of microswitches within the hat switch dictate the granularity in which the switch can detect movement. A hat switch with four microswitches can detect a total of eight approximate angles, while a hat switch comprised of eight microswitches can approximate sixteen angles as input from the user. The hat switch is able to approximate these set angles by using a simple technique in determining how many switches were pressed and where the pressed switches are. For example, in a four-switch hat switch, a press of the "top" and "left" switch together would allow the software to interpret an approximate angle of "top-left" or 130 degrees.

4.5.2.2 Analog Joystick

An analog joystick is an input device that produces two analog values, one to output the X-axis of the input and one for the Y-axis. These values can then be converted to digital values through an analog-to-digital converter in order to accurately get the stick position in an XY-plane, typically ranging from values 0 to 1023 for each axis. This high precision allows for an extremely accurate movement and direction information. The analog values outputted from the joystick are produced by two separate potentiometers integrated into a gimble system, in which each potentiometer outputs a single axis from the user input.

4.5.2.3 Multi-Axis Input Selection

As with all user interface devices, the desire to accurately replicate the real-world flight controls is a crucial aspect to selecting possible collective head inputs. In the case of multi-axis inputs, in the vast majority of cases, hat switches are used on the real aircrafts. This is a logical selection for both real aircraft and our simulation due to the requirements that exist for these controls. While the analog joystick does have much higher precision in stick location when compared to a hat switch, it also lacks the fundamental simplicity that can be leveraged by the hat switch approach.

One of these simplicities exists in the fact that the hat switch is inherently composed of multiple digital inputs rather than analog inputs. This allows a hat switch to be connected to the digital GPIO pins of the microcontroller with zero requirement for an analog-to-digital converter needing to pre-process values, unlike an analog joystick. These simple inputs also allow for simpler code due to the fact that the direction pressed on a hat switch is very finite and well defined based on a simple truth table of buttons selected. In the case of the analog joystick, boundaries must be set in the code to determine if the axis values represent a user attempting to give an input in a specific direction. These factors ultimately result in a decrease in efficiency from the microcontroller when using an analog stick as it requires ADC calculations and processor time every time the value is polled.

One advantage, other than the accuracy, that the analog joystick does have over the hat switch is the much larger availability of parts. Due to the proliferation of game consoles and controllers that have analog joysticks, the selection of analog joysticks is significantly greater than the more niche hat switch. However, we believe that it is worth the effort and possibly higher cost to source hat switches for our designs as the other benefits far outweigh this small hurdle. Therefore, we decided to utilize hat switches in our final design due to their similarity to the real-life control and due to their ease of implementation.

4.5.3 Information Display

In order to display diagnostic information, calibration instructions, and battery status to the user, a display will be required on the collective base. While there are many types of display and notification types, we have narrowed down our selection to an OLED screen, an LCD screen, and a set of simple indicator LEDs. Like all other aspects of component selection, factors such as usability, durability, and cost will all be taken into account when considering which of these will best suit our needs.

4.5.3.1 OLED screen

An Organic Light Emitting Diode (OLED) screen is a relatively high-resolution screen that boasts energy efficiency, a very thin profile, and no need for a backlight unlike a Liquid Crystal Display (LCD) panel. OLED displays are able to maintain power efficiency due to their ability to maintain images on the screen that were previously set without the need for constant power being drawn by the screen. The images produced on an OLED screen are also viewable at a larger viewing angle, thus facilitating easier reading of the screen if the collective base were to be mounted below eye level. However, despite these benefits, one of the major downsides of OLED displays are the size to cost ratio, which is heavily driven by the positive factors previously mentioned.

4.5.3.2 LED Indicators

As an inexpensive and simple to implement alternative, LED indicators could be used. In this configuration, specific colors or combinations of LEDs would correspond to predefined status codes that would communicate information to the user. However, this method would drastically limit the quantity and clarity of the information that is able to be provided due to the fact that a very select number of possible statuses could be indicated to a user with LEDs alone. This method also decreased ease of use because a translation via a lookup card would likely be required to encompass all possible status messages.

4.5.3.3 LCD Panel

An LCD panel uses liquid crystals in combination with a backlight to produce the image displayed on the screen. This is achieved via the two polarized glass pieces contained within the LCD. The backlight passes light through the first piece, while electrical signals are used to align the liquid crystals between the glass pieces, thus allowing the same light to pass through to the outer piece. This process is made much easier due to the fact that LCD panels for simple data display are typically implemented in a way that only supports black and white outputs with the display of number, text, and simple symbols.

4.5.3.4 Information Display Selection

While being lower resolution than OLED and higher cost than a naive LED solution, we determined that the LCD panel best fits our application as it acts as a good compromise between the other options. The LCD panel far outclasses the OLED display in cost, especially when considering the quantity of information able to be displayed on the LCD device, as compared to many of the very small OLED screens that are available at a comparable price. We also made the determination that many of the benefits of the OLED screen, when compared to the LCD panel, were benefits that would be hard to capitalize on due to the generally small role the screen plays in relation to our overall design. When considering the comparison between the LCD panel and the basic LED indicators, it can be seen that the ability to display human-readable text rather than having to interpret the state of the LEDs is vastly preferable. While likely being more expensive and larger than basic LEDs, we believe that the benefits seen in the overall ease of use of the system and the advantage of being able to display dynamic values, such as values gathered from the current state of the collective, far outweighs the cost and size difference.

4.6 Fabrication

In order to create the entirety of the physical helicopter collective control, there exists some elements of physical design that must be considered. Of most relevance to this project in particular are the issues of fabrication methods and materials for use in constructing the actual collective unit. Throughout this section, various options for both fabrication methods and materials will be explored in order to determine what approach is most suitable for this project.

4.6.1 Fabrication Methods

There are multiple possible approaches to constructing the physical collective unit. The methods that we have identified for research are machining and 3D printing. These fabrication methods each have their own advantages and disadvantages, including cost, speed of iteration during prototyping, ease of access, structural stability, and others. This section aims to explain the major facets of each of these fabrication methods and use them to choose the method most suitable for this project.

4.6.1.1 Machining

The term machining simply refers to removing material from some material base in order to manufacture a specific shape or part. Machining in general can refer to many specific methods of removing material including laser cutting, plasma cutting, water jet cutting, lathing, milling, CNC machining, and many more. For the purposes of our project, the most relevant form of machining is likely CNC machining. CNC machining is a kind of machining in which a machining tool is guided by a software-generated program in order to machine specific parts. CNC machining as a fabrication method is extremely prevalent for both hobbyist and industrial use, as it can machine extremely precisely and can utilize robust materials such as metal. This is one of the largest benefits of machining as a fabrication method, as making the collective unit out of a material like metal would result in a very strong and durable product. CNC machining can also produce very complex shapes, as it is guided by a program and not by a human operator.

Machining also, however, has some aspects that may be disadvantageous. Firstly, to have our designs machined would require the use of a third-party entity such as a machining company that would actually perform the machining. This would be both costly, as we would have to pay to have our designs machined and would prevent rapid iteration of prototyping as we do not ourselves have access to machining equipment or prior experience in machining.

4.6.1.2 3D Printing

The term 3D printing refers to a process in which three-dimensional objects are made from a reference model by combining a base material into a final shape or part. 3D printing exists in a few forms, but for the purposes of this project, 3D printing refers specifically to the use of a 3D printer to extrude some form of plastic or composite material out of a heated end in order to build up a physical part based on a computer model. 3D printing, while a relatively new technology in large industrial applications, has been very popular for smaller projects and structures.

The primary benefits of 3D printing are in the form of cost, speed of iteration, and ease of access. In terms of cost, 3D printing is rather inexpensive, as it primarily uses plastics as the primary material for structures, which can be obtained very easily and for low cost. It is also relatively cheap to acquire a 3D printer for individual use, however in the case of our team specifically we would not need to purchase a 3D printer for this project as our team already has access to multiple 3D printers. In terms of prototyping, 3D printing has a clear benefit in that printing complex parts can be done very quickly, usually under a day for most components. This would allow our team to iterate on our designs rather quickly in case of mistakes or design changes.

3D printing is also not without any downsides. The main downside of 3D printing over other fabrication methods is that the materials it uses are not as strong or as durable as a material such as metal, hence 3D printing is not very well suited to heavy-duty applications such as being put under heavy stress or being used in harsh environments like areas of high moisture or heat. This is not to say that 3D printing cannot produce satisfactory results, however, as most 3D printing materials under normal conditions of standard room temperatures, moderate moisture, and light load can last for extended periods of time with minimal wear or loss of integrity.

4.6.1.3 Fabrication Method Comparisons and Selection

Both machining and 3D printing have clear advantages and disadvantages. If we were to purely consider the durability and strength of the final product, machining would be the most suitable for our project. However, our project is intended for use in an indoor environment, in which the need for extreme strength and durability is less of a factor. With the exception of these aspects of strength and durability, 3D printing is more advantageous. 3D printing is less expensive than machining purely in terms of materials, is easier to access as we have 3D printers already available to us for use in this project, allows for faster iteration during prototyping and testing, and is more familiar to the members of this project in general. When considering the massive advantages that 3D printing presents over machining parts, our team found that 3D printing was clearly superior in this context and was the most suitable choice for this project.

4.6.2 3D Printing Materials

The method of 3D printing, as mentioned, primarily uses plastics or composites in order to construct parts. However, there exist many different types of these materials, all of which have various advantages and disadvantages. This section aims to determine which material will be most suitable for use in 3D printing to construct the collective unit by exploring the properties of some different common 3D printing materials.

4.6.2.1 ABS

Acrylonitrile butadiene styrene, more commonly known as ABS, is a thermoplastic polymer that is widely used in 3D printing. ABS is formed by polymerizing styrene and acrylonitrile in the presence of polybutadiene. These individual substances are acquired by various chemical processes, some of which involve the use of natural gases to chemically break down a substance into its parts. The primary characteristics of ABS that make it stand out compared to other plastics are its durability and strength. ABS possesses, relative to most polymers used in 3D printing, very high impact resistance. ABS can be used in many applications, including those that may be subject to significant forces such as door liners or machine housings. ABS also has rather high chemical and heat resistance, meaning that it is more durable than some other options and can last a very long time in consumer applications.

Some primary downsides to ABS, however, are related to its ease of use, safety, and environmental impacts. In terms of ease of use, ABS is less easy to use when 3D printing due to a few reasons. Firstly, ABS requires a constant increased temperature while printing in order to prevent warping and cracking, and also prints at higher temperatures in general. ABS is also potentially dangerous, as it has been found that 3D printing ABS results in higher emissions of toxic ultrafine particles during the printing process that can have adverse health effects in humans. Additionally, ABS poses some environmental concerns as ABS is made using natural gas as opposed to renewable resources and also, while ABS can technically be recycled, it is not common. ABS is also prone to degradation when exposed to sunlight.

4.6.2.2 PLA

Polylactic acid, more commonly known as PLA, is a thermoplastic polyester that is formed from lactic acid. PLA is made from sugars, most often from corn as it is the most common source, which is used to ferment the lactic acid to make PLA. Some standout characteristics of PLA are its basis in renewable resources, relatively low cost, and ease of printing. PLA is by far the most common material used in 3D printing as it is widely available and inexpensive. PLA is very easy to print with, as it requires relatively low print temperatures and does not require a constant heated environment. PLA also produces little amounts of ultrafine particles and is nontoxic. PLA can also be recycled effectively and decomposes relatively well when chemically composted.

PLA typically has a few notable downsides as well. For example, PLA compared to some other 3D printing materials has only moderate strength and durability. PLA is also susceptible to extreme heat and moisture, more so than some other polymers used in 3D printing. PLA also tends to crack as opposed to stretch. Some of these downsides, however, can be mitigated through the use of PLA+, a type of PLA that mixes in other compounds in order to increase strength, durability, heat resistance, and moisture resistance.

4.6.2.3 PETG

Polyethylene terephthalate glycol-modified, also known as PETG, is a thermoplastic polymer used in 3D printing. PETG is a version of PET (Polyethylene terephthalate) that has been modified to be more suitable for use in 3D printing. PETG is a less common material for use in 3D printing, however it presents certain clear advantages. PETG possesses very high strength and durability, similar to ABS, as well as good chemical, heat, and UV resistance. PETG is also relatively easy to print with, as it prints at moderate temperatures and does not require a constant heated environment. PETG also produces little amounts of ultrafine particles with a low number of compounds that are potentially harmful to humans. PETG is commonly known to remedy some common issues with ABS while also having most of the advantageous aspects of PLA.

PETG is not perfect, however. PETG’s primary downside is that it comes at a relatively high cost when compared to ABS and PLA. PETG also requires a higher temperature nozzle in order to print. Additionally, PETG is somewhat susceptible to moisture, more so than ABS. Finally, PETG is not produced from renewable resources, as the processes by which the substances required to produce PETG involve the production of carbon monoxide from coal.

4.6.2.4 Material Comparison and Selection

There are many different factors to consider when choosing which material is best for use in this project. In terms of strength and durability, ABS and PETG are the clear leaders, with PETG slightly behind due to it being more susceptible to moisture. PLA, however, is only mildly behind in these aspects, and our product is not intended for use under extreme stress or harsh conditions. These downsides of PLA can also be mitigated by using PLA+. In terms of ease of printing, PLA and PETG are the most optimal, with PLA edging out PETG due to PETG requiring a higher nozzle temperature in order to print. In terms of cost, PLA and ABS are both relatively low cost, with PLA being slightly less expensive due to its wide availability and use. PETG tends to be more costly than either PLA or ABS. In terms of health concerns, both PLA and PETG are better than ABS, as ABS produces a larger amount of dangerous and potentially toxic ultrafine particles. In terms of environmental impact, PLA is a clear winner. PLA is produced from renewable resources, can be easily recycled, and can be chemically decomposed effectively. PETG is slightly worse, as it can easily be recycled, but the process to produce it involves the use of a fossil fuel in the form of coal. ABS is also slightly worse as it is not commonly recycled and involves the use of natural gases in its production.

Overall, after reviewing the various options for 3D printing materials, we determined that PLA+ was the most suitable for this project. While PETG incorporates some of the desirable aspects of ABS and most of the desirable aspects of PLA into one material, its relatively high cost and base in fossil fuels, along with our project’s lack of a specific need for very high strength and durability, make it not the optimal choice. ABS, while extremely durable and strong, is hard to work with, is potentially toxic to humans, and is somewhat more expensive than PLA. PLA+, as opposed to either of these alternatives, possesses the most desirable attributes that we required for this project: low cost, ease of use, nontoxic, low negative environmental impact, and decent strength and durability. It is for these reasons that we have decided in the end to primarily use PLA+ when designing our collective unit.

4.7 Power System

The power system is the heart of operations for any electrically involved system. When designing a power system, the main goal is to design and provide an energy-efficient system defined by its safety, robust performance, reliability, and ability to remain uncompromised throughout steady and transient conditions. To fully understand the extreme conditions that a system will face during various use conditions, detailed analysis of each component of that system is conducted. In the case of an electronic device, it is important to understand each of the individual components’ functionalities and operating conditions and take careful consideration of their electrical parameters when presented with a dynamic environment. Therefore, it is crucial to select the right power system to enable optimal performance overall.

When choosing a power supply, several options are available and amongst the factors previously mentioned, there are objectives specific to our project that must also be accounted for and will influence our decision-making. The overall objective for our project is to create a low-cost, consumer-friendly, quality product, and provide a realistic and enjoyable experience. To accomplish that objective, the following sections will explore fundamental background information that is essential to the decision-making process regarding power options, constraints, safety and working design management, power consumption, and more.

4.7.1 Power System Options

Two power solutions were considered: Powering through a physical connection or using batteries, as well as the possibility of implementing both. Although the collective will essentially remain stationary, one of our goals is to provide a product that is easy to assemble, disassemble, and transport, therefore the option of battery power would be favorable. However, because our device will be used in flight simulation, that would mean wall outlets and/or computer stations will be available and may serve as viable options for supplying power to the unit. Because of the readily available nature of these power sources, the first option for power, a physical power connection was first investigated and will be documented in the following sections. Not only is this way easier to go about, but it will also allow us to analyze and put our project to the test more quickly.

4.7.1.1 Arduino Prototyping and Power Considerations

The Arduino Mega 2560 REV3 is a fully functioning programmable development board with open-source design documentation such as schematics and datasheets. This is why Arduino boards are typically used for fast prototyping of electronic devices for projects, since it helps in understanding the components involved and their functions in the overall design. Due to our selection of the Atmega2560 microcontroller for our project, we decided to use the Arduino Mega 2560 REV3 board for prototyping, since it utilizes the microcontroller that we want and serves as a good representation of what will ideally be required when creating our own PCB design. The Arduino uses a barrel jack, a USB slot, and two powering pins as inputs for supplying power, with each option having its own parameters. As per the data sheet specifications, the barrel jack on the Arduino board requires an input voltage range of 7V to 12V; Any amount in this range, when supplied, will be stepped down to 5V (which is the operating voltage for the Atmega2560) by a voltage regulator that is integrated on the Arduino board. The USB that is included with the Arduino board is of type A/B, which supplies a max current supply of 500mA at 5V. The 3.3V powering pin is rated at a max current draw of 50mA at steady 3.3V input. Finally, the 5V powering pin requires a constant and steady 5V input, with a current draw limited to whatever is being supplied. Understanding these constraints helps us make discoveries about the opportunities we have to optimize our own design. For example, the internal voltage regulator being used is linear and not the most efficient. It tends to draw too much power and creates too much heat when supplied with voltage close to its limit of 12V. This helped us in realizing that a switching mode regulator may be a better option to implement in our own design, as it is more efficient. Also, seeing how crucial it is to provide a steady 5-volts at the 5V pin, the voltage regulator will need to be very accurate and capable of stabilizing any fluctuations of voltage that the device may face.

4.7.1.2 Power by Wall Outlet

Households across the US typically provide 15A at 120V through wall outlet connections, which some home appliances can use without regulation. However, smaller electronic devices do not require this amount of power, and simply cannot handle it. Therefore, if this form of external power supply is going to be used, it would require an AC-to-DC adapter, also known as a wall-wart which is composed of diodes, PCBs, capacitors, and transformers. Some wall-warts will convert the 120V-AC down to 12V-DC and can reduce current all the way down to 100mA. Powering the collective unit this way would be simpler and would allow our power source to be the same between prototyping and implementation. However, while this method may be convenient for prototyping, it has multiple downfalls when compared to USB or battery power. As the unit will likely be near a computer, USB would likely be a more convenient candidate for a physical power connection.

4.7.1.3 Power by USB

There are several types of USB, each with their own standards and specifications. When looking at the different types from the perspective of supplying power, we will focus mainly on the amount of current each one supplies. The USB 1.0 and 2.0 have a downstream current capability of up to 500mA. The 3.0 version increases that current supply to 900mA. USB 3.1 brought along what is known as super speed charging, delivering up to 1,500mA. As can be seen, there is a wide range to pick from, and modern devices have a protective and dedicated charging mechanism that allow for the use of any of the USB types without an issue. However, the Atmega2560 will theoretically draw a maximum of 200 mA and is the major power-consuming component in the design, which means that even when accounting for other components on the board, it is likely that any version of USB will provide sufficient current for the collective unit.

4.7.1.4 Power by Battery

In general, using batteries to power electronic devices allows for more flexibility, but it can also create more room for error. In the previous cases, the power supply is essentially fixed and standard. In the case of batteries, however, the supply is limited, and the types, amounts, and arrangements will have a direct impact on the adequacy and availability of supply. Recall that a linear regulator is stepping down the supply voltage; therefore, if a battery of 7V is chosen, after a short period of time of battery discharge, the difference across the voltage regulator along with losses of energy may not satisfy the 5V operating requirement, resulting in a useless and short-lived device. The idea of then using a 12V battery to resolve this issue is easy coming. However, a higher voltage will just cause the voltage regulator to dissipate more heat, resulting in more inefficiencies across the system. An alternative solution could be to provide more current capacity instead of a higher voltage to increase battery life, but this also puts stress on the voltage regulator, also resulting in higher heat dissipation. In that case, it is possible that the issue is met with the type of voltage regulator being used. A linear regulator produces a lot of heat which, although it is doing its job, tends to waste a lot of energy, a trait that is common in linear regulators. On the other hand, switching regulators have a better reputation in more efficient power conversion. More detail on this will be covered later in the document.

4.7.1.5 Power Source Selection

The goal for the previous set of sections was to investigate some of the constraints that are present when choosing different methods of supplying power. As explained, although some methods may require more work, having more flexibility can be beneficial. In the "Power System Options" section, it was stated that a physical connection will be used for breadboard prototyping because it is a simpler solution, and it will allow for earlier device testing. However, we decided to implement a combination of both USB power and battery power in the final product. This allowed the most flexibility of use by the end-user and the best user experience overall.

4.7.2 Battery Options

Considering our goal of low-cost production, primary batteries would not be ideal as they must be constantly replaced as they are depleted. Secondary batteries would make more sense because of their reusability; however, the capacities of rechargeable batteries are typically lower than their non-rechargeable counterparts. Not only that, each of the types of batteries vary in their own chemical structure and technology, which will ultimately define their characteristics and capabilities, making it even more challenging in choosing which type would best fit our needs.

Due to the nature of application of the collective and a longer run-time in flight simulation being desirable, it was determined that secondary (rechargeable) batteries were the most optimal solution. The following sections will explore various types of secondary battery options available. The battery types researched were selected using the following criteria: Popularity in electronic devices (often a good indicator of reliability), energy-density, capacity, charge cycles (lifetime), and cost.

4.7.2.1 Nickel Cadmium (Ni-Cd)

As the name implies, this battery's makeup involves two chemical compounds, Nickel and Cadmium. The nickel is present in the Nickel Hydroxide ($Ni(OH)_2$) make-up of the cathode electrode. The cadmium is present in the Cadmium Hydroxide ($Cd(OH)_2$) make-up of the anode electrode. These are the two compounds that are being discharged from the positive and negative electrodes, with specific reactions of 0.49V and -0.81V, respectively, and a total charge between both of 1.3V. The 1.3V is the maximum voltage of the Ni-Cd cell, with an average voltage of 1.2V. This battery comes as a secondary type (rechargeable), and is a good option when low-cost, long-life span, and large charge cycles are a requirement.

The disadvantage comes in careful battery charging because the life cycles depend on how deep the discharge is allowed on the battery. A very long lifecycle of around 50,000 discharges can be achieved if the battery is kept at a discharge depth of 10%. Another disadvantage is the toxicity of the battery due to the presence of the cadmium compound, therefore classifying it as being unfriendly to the environment. For these reasons, their use in industry is waning and their successors, Nickel-metal hydride, have taken the lead in overall capabilities.

4.7.2.2 Lithium-Ion

The well-known Lithium-Ion batteries, first introduced by Sony Corporation in 1991, are widely used in many electronics as they provide a high power density, high capacity, and long shelf life. Even more, lithium-ion batteries have great energy-to-weight ratios, low self-discharge, slow loss of charge, and no memory effect. All these qualities are what has made the lithium-ion battery so prominent. Its reputation holds strong as it is so prevalent in the market for important and popular uses such as: portable electronics, electric vehicles, military purposes, and much more. As the name implies, lithium metal is the metal used in the battery, and different versions exist composed of different lithium driven compounds. The materials present inside of these batteries are Lithium Cobalt Oxide, Lithium Manganese Oxide, Lithium Iron Phosphate, Lithium Nickel Manganese Cobalt, and Lithium Cobalt Aluminum Oxide. Each type has its own specification, due to the specific energy of the material or compounds. Several aspects of the lithium-ion battery also involve graphite material and graphene structures such as the 2D honeycomb which allows for the storage of li-ions, and the 3D aerogel which allows for the flow and transportation of the li-ions.

Although lithium-ion batteries are impressive in their capabilities, they require more delicate attention when it comes to their maintenance. Unlike many of the other batteries, it is strongly discouraged to fully charge and discharge lithium-ion batteries; instead, it is optimal to keep them at around 50% between their maximum and minimum “rated” charge.

4.7.2.3 Lithium-Polymer

Lithium-Polymer batteries, a subgroup of the lithium-ion batteries, have a slightly higher energy density to weight ratio, are encapsulated in aluminum-laminated films, and are slimmer and lighter than lithium-ion batteries. Their physical attributes make them a common choice for smartphone and GPS applications. Although lithium-polymer batteries very closely resemble lithium-ion batteries, they are preferred when weight and space are of main concerns. The manufacturing process of shaping and using thin film technologies enables flexible designs to accommodate different shapes and applications.

4.7.2.4 Battery Decision

After careful consideration, we decided that the Lithium-Ion battery will be chosen for this project. The lithium-polymer battery also proved to be a good choice, with Nickel-metal hydride coming in third. However, lithium-ion batteries better satisfied our requirements in that they hold their integrity longer than other batteries before their capabilities start to degrade, they have amongst the highest energy density (which typically implies longer runtime for applications), a low self-discharge rate, and they are friendlier towards the environment. Even more, their prominence in the market for a wide variety of electronic devices ranging from small microchips to some high power-demanding tools, serves as a testimonial to their effectiveness. The table provided below simply summarizes some of the battery types mentioned in this document and their specifications which were taken into account when selecting a battery type.

Table 13 – Battery Type Comparison

Battery Type	NiMH	Lithium-Ion/Lithium-Polymer
Nominal Cell Voltage	1.2V	3.6V – 3.7V
Operating Cell Voltages	1.0V – 1.4V	3.0V – 4.2V
Max charging voltage	1.4V – 1.6V	4.2V – 4.3V
Max charging current	1C (with ΔV/ΔT)	~ 1C
Max discharging current	3C – 15C	1C – 2C
Specific energy	60 – 120 Wh/kg	150 – 250 Wh/kg
Specific power	250 – 1000 W/kg	100 – 400 W/kg
Internal series resistance per cell of (18650)	5 – 50Ω	15 – 100Ω

In the table above, the NiMH battery was included instead of the Ni-Cd because it is its successor. Note, the internal series resistance corresponds to the 18650-battery size, which is the one that will be used for this project.

4.7.3 Linear vs Switching Regulators

In power distribution, different types of power plants provide power throughout large regions to supply industrial and residential consumers. Large transformers convert the massive amount of power from the powering plants to a lower, safer, and standardized level of use. At a much smaller scale, such as in electronic devices, voltage regulators are essentially doing the same thing. In an imperfect world, supply conditions are often not optimal; therefore, there is often a need to step-up, step-down, and maintain a stabilized voltage throughout different parts of an electronic device, so that all components are operating optimally and safely. Thanks to the advancement of technology, we can provide those conditions to our devices, as well as protect them from voltage spikes and fluctuations. The most common types of voltage regulators are linear and switching regulators. The linear regulator is the simplest amongst them in terms of functionality and construction. Within the device, a combination of resistors, diodes, and transistors work together to take reference and input voltage and provide a specific output voltage. However, its simplicity comes at a price. Linear voltage regulators are known to dissipate a lot of heat, which is typically why, when given a specified input voltage range for a device, it is discouraged to operate it towards the higher end of that range, because the step-down ratio will be larger. A heatsink will typically be used to cool off the regulator. Because a large amount of heat being dissipated is common, linear regulators are deemed inefficient, since energy is being constantly being wasted.

The switching regulator often comes in first place as the device to be chosen in electronics. Although more complex in terms of functionality and construction, the pros outweigh the cons. The switching regulator has the capability of stepping-up voltages to levels higher than the input level and of different polarities, something the linear regulator cannot accomplish. It is also known to waste very little energy, thus creating very little heat. The different topologies are known as buck (step-down), boost (step-up), and buck-boost (steps up/down and inverts), each with their own efficient manner of regulation. For example, a buck converter will use passive components to charge and discharge them using a pulse width modulator. The modulator outputs a certain pulse to determine the voltage at the output. The process is very efficient, and the device loses a minimal amount of power after regulation. As mentioned, it does come with its disadvantages, those being that it creates ripple voltages, a lot of noise, and although smaller in size, it is more expensive.

Overall, both types of regulators can be used effectively, if wisely chosen for the right conditions. As for our project, a switching regulator will most likely be chosen, mainly due to its efficiency of conversion and its accuracy of feedback and response, making it more reliable in maintaining stable out conditions overall. Stability will be crucial considering the necessity of maintaining a constant and steady voltage to power the system reliably. The first step in choosing a voltage regulator, is to determine the parameters needed. The following sections contain information based on some of the voltage regulators that were considered.

4.7.3.1 LM1117MP

The LM1117MP is available in a variety of different versions. The version of interest is the adjustable voltage regulation version, where it can set the output from 1.25V to 13.8V. It can also output fixed voltages of 1.8V, 2.5V, 3.3V, and 5V, our desired value being the 5V for the Atmega2560. Its key features include current limiting and thermal protection where a shutdown will occur if limits are surpassed. It has load and line regulation of 0.4% and 0.2% maximum, respectively. The LM1117MP promises voltage output accuracy within ± 1%, using a Zener trimmed bandgap. It also uses a tantalum capacitor at the output for improved stability and transient response.

4.7.3.2 L7805

The L7805 was commonly mentioned on several sites and personal projects, therefore it has earned its place in our list. The L7805 is manufactured by STMicroelectronics and has several versions to it, known as the L78 series. The series is available in different packages that are known to be used in high power applications, such as D²PAK, DPAK, TO-220, and TO-220FP. Its key features include thermal overload protection, short circuit protection, output transition SOA protection (which protects internal components, like MOSFETS), ability to provide output current up to 1.5A and a large range of output voltages of 5V, 6V, 8V, 8.5V, 9V, 12V, 15V, 18V, and 24V. The L7805 is one hefty voltage regulator and sure proves its worth with its usefulness in a wide range of applications.

Although it is packed with so many features, too many of them extend far beyond the needs of our project. Even if it is suitable, choosing a regulator that maintains a narrower focus to the requirements of our project would be optimal. If the case should arise where our selected regulator is not viable for some reason, this may serve as a viable alternative.

4.7.3.3 TPS63070

The TPS63070 is not a voltage regulator but is rather a buck-boost converter. However, it is included in this section because it essentially behaves in the same way. This converter can take a voltage range of 2V to 16V and convert it to an output range of 2.5V to 9V with up to 95% efficiency. Both buck and boost modes can output 2A of current. Some of its key features are fixed and adjustable output voltage, input/output overvoltage protection, overtemperature protection, output discharging option, and automatic transitioning between step-down and step-up modes. Some very promising features that it also includes are automatic power saving mode for improved efficiency during low power demand, and a disconnection from load during shutdown. Some of the typical applications in which the TPS63070 can be found are industrial metering equipment, mobile PC's and mobile internet devices, personal medical products, and dual li-ion applications. The buck-boost converter uses synchronous rectification which is how it is able to achieve such high efficiency. This, along with the capability of both stepping-down and stepping-up voltage, makes it a very desirable converter.

4.7.3.4 Initial Regulator Decision

An important note to make when comparing voltage regulators and converters is their potential deficiencies when they are not used in the expected or suggested environment. A voltage regulator is typically used for lower current outputs whereas a converter is typically used for higher current outputs. If utilized at these high current outputs, they will work efficiently, however, the efficiency can drop significantly at low current loads. Considering these risks, the LM1117MP comes as the best choice. Even though the LM1117MP is a linear regulator, and it was stated before that a switching mechanism would be better, a misuse of their application could be far worse than the losses of energy that occur in linear regulators. However, due to the capabilities of the TPS63070 and its stated 95% efficiency, it will also be tested and compared with the LM1117MP.

4.7.3.5 Final Regulator Decision

While we initially selected the LM1117MP as the voltage regulator of choice for our project, in the end we opted to use the TPS63020 switching regulator due to parts availability. The TPS63020 switching regulator is extremely similar to the TPS63070 regulator discussed previously and allows for an input voltage range of 1.8V to 5.5V and an output voltage range of 1.2V to 5.5V. The TPS63020 can provide up to 4A of current at over 90% efficiency. While we initially selected a linear regulator due to concerns of low efficiency at low current load, the TPS63020 has an integrated power save mode which provides a minimum efficiency of 60% even at extremely low load currents, mitigating the downsides that a switching regulator would typically present.

4.7.4 Battery Safety & Management

Battery management is an important aspect for any electronic device as there are different battery types and chemistries and each will behave accordingly. Considering our decision in using secondary lithium-ion batteries, parameters concerning its use and recharging were carefully noted. Li-Ion batteries can be charged at high initial currents which can reach up to the Ah rating of the battery. This may be beneficial for devices that need a quick recharge for immediate use. However, it is not suggested to use such high rates of charge, as this stresses the battery due to an increase in its internal temperature, which also brings to attention safety concerns like ignition (unlikely, unless battery is overcharged) and ambient temperature. Another notable disadvantage with fast charging is that a higher rate of charge may result in a lower capacity reached, at the point a full rated voltage charge is achieved. Conversely, a slower rate of charge will ensure full capacity by the time the battery achieves its fully charge recommended voltage rating. A good rule-of-thumb to follow is to charge the battery at 50% of the current rating value of the battery. Charging it at a rate even lower than 50% (say 30%) would work in favor of the battery, although charging time will extend. As we plan to implement USB power passthrough while charging the batteries, charging at a lower rate is suitable for our project, since there is no urgency for its use and USB charging is expected to be readily available most of the time.

The second biggest concern when using Lithium-Ion batteries is the voltage parameters. Like the rate of charging, the range of charged voltage is equally, if not more important to consider. The nominal voltage per the average cell for all lithium-ion batteries is 3.7V, with rated charge and discharge limits of 4.2V(max) down to 3.2V(min); At no point should the battery surpass these limits as permanent damage can be caused, reducing its shelf-life and in certain instances, possible ignition.

4.7.4.1 Protective Charging

A wide range of charging ICs for Lithium-Ion batteries are available in the market, each with their own unique features and capabilities for proper charging, discharging, and overall monitoring of the battery. Because the main issues we care about are overcharging and rate of charging, the features we want in the charging IC are constant voltage and current input supply management, and auto cut-off at full charge. Typically, these ICs use diodes to sense for overcharge or high thermal temperature. When a threshold is reached, the diodes will block the path of current. A MOSFET is then used to measure the voltage drop to then activate a new path and redirect the current, or to open the circuit and totally prevent current flow towards a certain section of the circuit. A temperature detection feature would be desirable, but it is not necessary if careful management and design of the circuit is conducted. This brings us to some of the charging IC solutions that were explored and selected for review, which are mentioned below.

4.7.4.1.1 LM 317

Although it is a positive linear (variable) voltage regulator, it is commonly used as a main device in charging circuits. This popular IC provides an adjustable three-terminal voltage regulator with the capability of supplying up-to and above 1.5A, over an output-voltage range of 1.25V to 37V. When assembled with the right components, it is easily made capable of current limiting, internal thermal protection in the case of an overload, and safe operating voltage protection. Along these features, the useful benefit is the adjustability and large range of regulation it is capable of, via the use of two external resistor that can be picked in accordance with the desired output voltage in respect to the load requirements.

The main disadvantage with this IC is that it is known to dissipate a large amount of heat due to the voltage drop, typically 2.5V, caused by the regulator. Therefore, heatsinks are typically used for cooling purposes.

4.7.4.1.1 BQ25611D

The BQ25611D is a 3-A switch-mode battery charging and power management IC that promises up to 92% charge efficiency at a stated 2A by a 5V input and reduces overall charging time as well as prolongs the run time during the discharge phase of the battery. It integrates safety and optimization features including: Battery thermistor monitoring, Input OVP, Power Path, Integrated USB OTG, VINDM (which tries to adapt and regulate the system to any limits that may be present when an adapter is introduced), constant current limiting, and last but not least, 130-ns fast auto cut-off input voltage to protect it from overcharging.

4.7.4.1.2 MCP73831

The MCP73831 is a linear charge management controller that requires a relatively low number of external components to operate. This makes it a suitable device for small-spaces and portable applications. Some promising features included are: Thermal regulation and shutdown, charge termination, input overvoltage protection, reverse discharge protection, and more. A neat feature is the option of tuning the charging current that will be supplied to the battery, which allows flexibility if a change in battery occurs.

4.7.4.1.3 LTC1730 and TP4056

The TP4056 is a popular standalone linear battery charger commonly used for portable devices. It provides thermal regulation to protect the battery from high temperatures during high powering operations, its internal architecture involves a PMOSFET that prevents negative charge (feedback) current to the circuit, automatic recharge, constant-current and constant-voltage, and programmable charge current output (great for managing heat production). However, it is not an officially branded and regulated product, therefore, an alternative was found. The LTC1730 is a credible and branded alternative, manufactured by Linear Technology. It is a single cell pulse charger with overcurrent protection. It has programmable charge termination, low-battery drain, automatic battery refresh, charge qualification and temperature sensing, and similar to the TP4056, it requires no blocking diode due to the internal MOSFET. An interesting fact about this IC is that as the battery is being charged, the internal MOSFET will begin switching from on-and- off states, gradually decreasing its duty cycles, as it gets closer to its programmed voltage limit. This comes as a key feature because we can set the desired battery charge voltage and ensure no overcharge. This feature would become useful if, for example, the device is faced with transient conditions and fluctuations of voltage, which would be detrimental to the battery.

4.7.4.1.4 Initial Charging IC Decision

It would have been a simple application using the LM317 in a battery charging circuit; PNP transistors and diodes would have acted as the sensing components in the circuit that would monitor the battery's condition. However, it ultimately made sense to pick a complete and fully integrated IC over the LM317 which would have required the use of several external components to allow it to function as a sensible and protective battery charging circuit. Utilizing existing IC's that are fully packed and designed to fit our needs requires less design and lowers the risk of improperly charging the battery. We initially settled on utilizing the BQ25611D, which is more than capable for our application.

4.7.4.1.5 Final Charging IC Decision

While we initially selected the BQ25611D as our charging IC, we decided to implement the MCP73831 in our final PCB design primarily due to parts availability and our PCB manufacturer maintaining their own stock of the MCP73831. This meant that we were able to have the charging IC assembled onto our board along with the rest of the parts which saved significantly on cost and time, while requiring little trade off in terms of charging capabilities.

4.7.5 Power Consumption

At the start of the project, it is not intuitive to know the precise amount of power that is consumed by the full system, as the system has not been fully built and the types and amounts of components have not been finalized. However, a good estimate can be made by just looking at the major and most power-hungry components involved. This is the first step in finding an appropriate and sufficient power supply. Below is a table that details all major components involved in our project thus far.

Table 14 – Component Worst Case Power Values

Device/Component	Operating Voltage	Current Consumption	Power Draw
Atmega2560	5V	200 mA	1 W
BM71	3.6 V	13 mA	0.0468 W
EMS22A50	5 V	20 mA	0.1 W

First, it is necessary to mention that worst-case scenarios have been assumed. The 200mA current consumption by the Atmega2560 is achieved by assuming all I/O pins are in use and the possible current draw by each are maxed out. The BM71 is at peak current draw which was observed to be an unusual case compared to the average current consumption of 230uA during advertising events, which are typically the most heavy-loading events to occur. Here the advertising event, although it occurs for fractions of a second, it was assumed to be running continuously as if it was a constantly interrupting event on the clock or I/O pins. According to the above table, the total current consumption will be chosen to be approximately 400mA. A notable mention is that a safe-guard rule of providing a power supply with a capacity of 1.5 times more than the system's total current consumption, will be a minimum threshold value when choosing the power source, to compensate for any irregularities and losses. Considering the total current consumption stated above, it was decided that the Lithium-Ion 18650 rated at 3.7V with a capacity of 2600mAh was sufficient for our project and should provide 6.5 hours of run-time. This does not meet our goal of >10 hours of run-time, but again, this is under extreme and unlikely loading conditions, therefore a good indicator that our goal can be easily met under efficient settings.

In the same manner as for the worst-case scenario, we decided to investigate the typical scenario to see just how much leeway we would have in available power. Using the same previous table and once again, considering major components, we have the following typical values under typical conditions.

Table 15 – Component Typical Power Values

Device/Component	Operating Voltage	Current Consumption	Power Draw
Atmega2560 (16 MHz)	5V	20 mA	0.125 W
BM71	3.6 V	1.3 mA	0.00468 W
EMS22A50	5 V	20 mA	0.1 W

As seen above, a great reduction in current consumption occurred for the Atmega2560. Therefore, the 20mA corresponds to a supply voltage of 5V at 16 MHz. The value of 1.3mA for the BM71 was acquired by taking an average of typical measurements that occur during a connected status. Some of these measurements occurred as small incremental peak values, as short as 18.75ms, 50ms, and 100ms, with current consumptions reaching 2.23mA, 2.13mA, and 2.10mA, respectively. Bigger increments of 500ms and 1000ms occurred with current consumption as low as 83µA and 80 µA. Clearly, most of the time, the BM71 is pulling mostly micro-amps worth of current.

Even more, these values can be achieved based on how the BM71's power modes and current consumptions are configured to operate. The same can be done with the Atmega2560; depending on the configuration, functionalities such as sleep mode, idle mode, power-down mode, power-saving mode, power reduction registers, clocking, and more, it can be optimized to reduce overall power consumption.

5.0 Project Design

5.1 Electrical Hardware Design and Schematics

In order to create the helicopter collective, two major electrical hardware subsystems were realized: a logic and input subsystem as well as a power subsystem. The logic and input subsystem facilitates the receipt of input such as collective head button presses, throttle and collective axis positions, and user interface buttons. The logic and input subsystem is also responsible for processing these inputs by interpreting, formatting, and transmitting input to the connected host device while also displaying information to the user on the base unit LCD. The power system facilitates the creation and distribution of stable power sources for the logic and input subsystem in order for the helicopter collective to operate.

5.1.1 Initial Power Design and Prototyping Overview

At the start of any design phase, testing hardware and software in theoretical and/or imitation environments will prove beneficial towards the pursuit of a successful final result. Performing individual component testing and then fully integrated system testing will allow for easier discovery of any faults in the design. The following sections entails information on the steps that were taken to transition from the research phase to the design phase of our project. It also provides a more in-depth analysis of initial design models regarding the power source, charging module, and power regulation module for our device.

5.1.1.1 Electrical Design Software

After hours of investigating components, power supplies, MCUs, and any other necessary components, it can be a bit overwhelming at first to attempt to design and piece all the parts together to get a working system. Thankfully, several programs tailored for circuit design and simulation are available and make the initial stages of the circuit design easier and safer. It is crucial to practice utilizing these programs as the first step of initial design because it provides a safe method and environment of testing, for both the user and the electrical components involved. It is beneficial in that it provides the user the ability to simulate a theoretical running environment for individual components and for the system as a whole; In the case of a fault, the user has the flexibility of easily modifying and re-arranging the design as needed. Overall, it becomes essential to use these resources for the sake of time, money, and safety, all of which are either constraints or goals one may have in their project. With that being said, we look at some of the top software on the market for circuit design and simulation.

5.1.1.1.1 EAGLE

Eagle is a prestigious electronic design automation (EDA) software that is widely used and supported by universities, engineering industries, electrical component manufactures, and many hobbyists. Eagle presents advanced features such as circuit schematic layout, PCB layout, an extensive component library, simulation capability brought by the built-in SPICE software, and share mode for collaborative work amongst a team. There are also some manufactures that provide an ECAD model for several components that can be downloaded and easily imported for use in Eagle; The ECAD model includes the component's PCB symbol, footprint, and a 3D model. Finally, Eagle is cross-platform compatible and can run on Windows, Mac, and Linux operating systems. Some drawbacks that come with Eagle is that it may not handle more complex heavy-duty PCB designs, therefore other software with more advanced features like 3D model layout viewing may be preferable. Eagle is also a subscription-based software; however, it is not terribly expensive.

5.1.1.1.2 Altium Designer

Altium Designer is another EDA software favorited by the circuit design community and comes close in comparison with Eagle. Altium is known to be more advanced in its capabilities than Eagle and can better handle more complex and multi-layered design and processing. Also, Altium is licensed based software but comes at a heftier price. Overall, Altium holds outstanding capabilities that is ultimately preferred, however, for smaller scaled projects and lower budgets, Eagle would be the better option.

5.1.1.1.3 KiCad

KiCad is a free and open-source modeling software. Its features include schematic drawing, PCB layout, 3D model viewing, and an abundant library of components, packages, and symbols. Simulation is possible with SPICE as an additional download and through file transfer. Although a great software, it may not be the easiest to use. KiCad is quite popular amongst hobbyists but it is not comparable to Eagle or Altium as far as advanced capabilities and applications.

5.1.1.1.4 Fritzing

Fritzing is an open-source CAD software with an extremely intuitive interface, great for amateur hobbyists and circuit designers. Although it will not play much of a role throughout our project, it is a great program worth mentioning due to the realistic models of components and devices, color coding of wires, breadboarding, and more; It makes it possible to closely imitate an actual physical design which provides an ease of transition from software to hardware implementation.

5.1.1.1.5 NI Multisim

NI Multisim serves as a great environment for circuit design, simulation, layout, and testing. NI Multisim is considered to be at industry standards and is one of the few circuit design programs to house-in the original SPICE program for simulation. It integrates a former version, Multisim 14.2, with the Ultiboard software, and includes over 36,000 electrical components, validated by semiconductor manufacturers. The use of NI Multisim is present in UCF and has been used as a main program for circuit design and testing prior to implementation, for several courses. Its user interface is a quick-learn, easy to use, and easy to mimic in real life. Some of the main and most common useful features is the multimeter test, oscilloscope test, and measurement probes. These features make it very easy to get a good idea on how the circuit being built will behave. In fact, as per experience using the software, real physical circuits would be compared to the simulated circuit in NI Multisim for verification and testing for accuracy; Often times, the case would be that both resulted in very similar values with any error deviating just around ± 5%, which was considered acceptable.

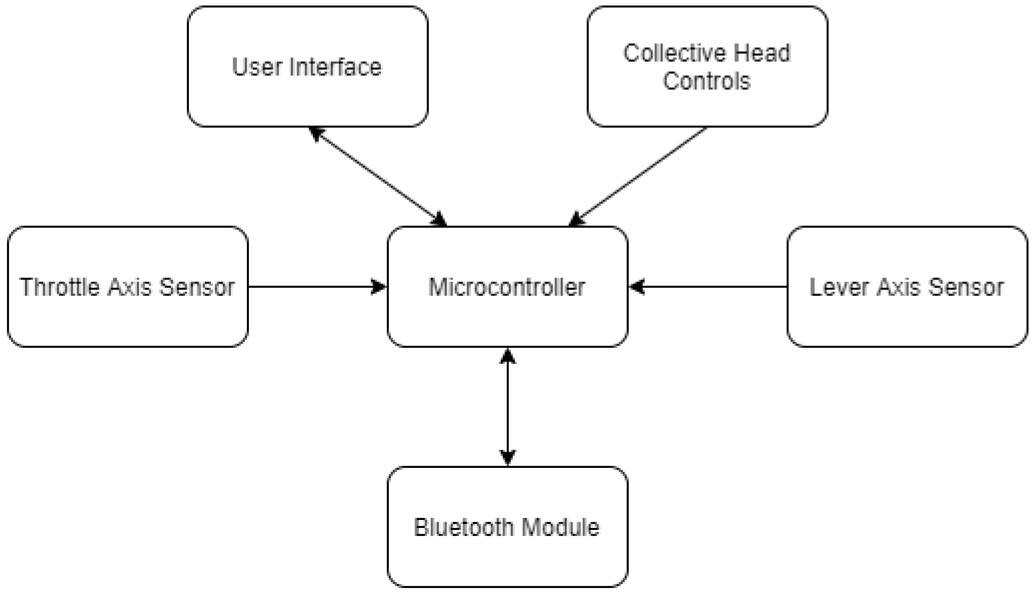
5.1.1.2 Electrical Design Software Decision

For this project, we decided on using Eagle due to the fact that our project specifications fall in-line with the features that Eagle has to offer. For example, the available ECAD models provided by some component manufacturers for import and use in Eagle not only shows its prominence in the market, but it also allows for accurate component selection and ease of design, which was crucial in developing a completed PCB design. Also, Eagle has been implemented into Fusion 360 as Fusion 360 Electronics, which allowed for cloud storage of our PCB design and allowed for multiple team members to collaborate on the same design.

5.1.2 Logic and Input Subsystem

As mentioned previously, the primary purpose of the logic and input subsystem is to process inputs and transmit them to the connected device for use in the flight simulator. A high-level overview is present in the Figure below which highlights the main components of the subsystem. Each of these individual components' purpose and implementation details will then be discussed in the following sections.

Figure 7 – Overview of Logic and Input Subsystem Components


User Interface
Collective Head Controls
Throttle Axis Sensor
Microcontroller
Lever Axis Sensor
Bluetooth Module

5.1.2.1 Collective Head Input

In order to receive input from the collective head, the controls on the head must be connected to input pins on the microcontroller which can detect the state of the buttons, switches, and other head inputs. However, not only is the collective head located at the top of the collective lever whereas the main PCB is located in the base unit, but one of our advanced features is that the collective head must be swappable. This presents a challenge in two areas in particular: the head must be able to electrically connect and disconnect during use and also the microcontroller must be able to detect which head is currently connected and adapt its logic appropriately. This eliminates the possibility of simply running wires all the way up the collective lever shaft and connecting the head inputs directly to the PCB via a long wire. Also, the collective itself will require assembly which would render a permanently connected wire between the PCB and the collective head ineffective.

In order to solve the first problem of the head needing to physically disconnect and reconnect, from an electrical perspective, a small female circular connector was mounted in the collective lever shaft near the top. A matching male circular connector was wired to the inputs inside of the collective head and protrudes out the bottom of the head when disconnected. To swap which head is currently installed, the current head's circular plug simply needs to be unplugged from the collective lever shaft and the new collective head's plug needs to be inserted. This allows quick, easy, and effective swapping of the collective head which is essential to the function of our product.

In order to solve the second problem of the microcontroller being able to adapt to which head is installed, an indicator line was used to determine which head is connected. This was achieved by dedicating a pin of the circular connector to a line whose voltage is read by the microcontroller. The microcontroller can then interpret this voltage level and distinguish which head is connected and execute the proper programming logic associated with such a collective head. For example, with two different collective head options, the indicator line was tied to either GND, 2.5V, and 5V to distinguish between No Head, Bell 206 Head, or Huey Head, respectively. This indicator line along with the plug-and-play implementation described previously both allows for user friendly head-swapping but also allows for the possibility of easy integration of additional collective heads. Should a new style of head be desired for the system, it can utilize the same circular plug connector and simply be assigned a new voltage level for the indicator pin, and after a firmware update the head would function as intended with little to no electrical redesign.

Finally, while the collective head itself is able to be easily plugged and unplugged into the female connector mounted in the top of the collective shaft, it is also necessary for assembly that the PCB is not directly hardwired but rather is able to connect after the collective shaft is installed into the base unit. In order to achieve this, the female circular connector is wired to a removable 18 pin male Molex plug that hangs out of the bottom of the collective lever shaft. Thus, after mechanical assembly, the connector can simply be connected to a corresponding female receptacle on the main PCB to connect the inputs to the microcontroller. This solution allows for user friendly assembly and disassembly, and also allows for flexibility in the wire length and orientation which is important as the collective lever shaft will be rotating along with the wire inside.

5.1.2.2 Axis Sensors

There are two axis sensors present which are essential to product function: the throttle axis sensor and the collective axis sensor. These axis sensors have 6 pins which must connect to the microcontroller in order to read their current angular position. The microcontroller then sends an enable signal as well as a clock signal to probe the axis sensors for their position. The axis sensors report this position via single data line which reports 1 bit per clock cycle. However, these two sensors must be separate from the PCB as they must engage mechanically with the collective lever and throttle, and the throttle axis sensor will constantly be moving as the collective lever is pushed up and down. For these reasons, a similar approach to the collective head inputs was utilized in which a 6-pin connector is connected to each encoder, which is then connected to a wire with a male Molex plug on the other end. These plugs (one for each axis sensor encoder) mate with female Molex receptacles on the PCB in order to connect the microcontroller pins to the encoders. Similar to the collective head inputs, this allows for ease of assembly, disassembly, and allows flexibility in movement for the throttle sensor to move without breaking the system.

5.1.2.3 Physical User Interface

While the helicopter collective will generally be operated without much need for user interface interaction, some input is required in order for the collective to function properly. The main functions of the user interface system are to calibrate the axis sensors and to manage the Bluetooth Low Energy connection, such as pairing and connecting to a remote Host device. For this purpose, a simple user interface consisting of three push buttons, a power switch, a 16x2 LCD, and a display contrast adjustment knob are present on the outside of the base unit which the user can use to manage the collective. These components must connect to the main PCB but cannot be directly mounted to the PCB due to their location on the base unit. Thus, a similar solution to the previous sections is utilized in which a 24-pin male Molex connector is wired to the various user interface components and mates with a 24-pin female Molex connector on the main PCB to connect to the main system.

5.1.2.4 Bluetooth Module

The Bluetooth module is an integral component which communicates with the microcontroller over UART in order to transmit HID Input reports from the collective to the Host device in order to control the flight simulator. The Bluetooth Module we have selected for this purpose is the BM70 by Microchip, which is a fully self-contained Bluetooth Low Energy module including a full BLE stack and an integrated PCB antenna.

The Bluetooth module functions by receiving UART commands from the Atmega2560 microcontroller, which means that there must be a direct 2-pin UART connection between the two modules to facilitate this communication. However, the BM70 also requires an additional connection to indicate whether the module should operate in Configuration

Mode or Application (Standard) Mode. Also, the BM70 runs on 3.3V whereas the Atmega2560 runs on 5V and outputs 5V on its pins. This means that the transmit pin from the Atmega2560 cannot be directly connected to the receive pin of the BM71 or it will be overvolted and fail. Thus, a simple voltage divider is required to convert the 5V signal to a 3.3V signal. A similar voltage division is also required for the P2_0 pin which is used to set Configuration or Application mode. A Reset pin on the BM70 is also connected via a switch to ground in case the Bluetooth module requires a manual reset by the user.

5.1.2.5 Programming Interfaces

During the development and breadboarding process, the Bluetooth module's configuration and the microcontrollers' software was simply able to be uploaded via the use of development boards with integrated computer connections and software. However, once the PCB itself was manufactured, the microcontroller software and the Bluetooth configuration had to be flashed directly onto the Atmega2560 and BM70 on the board. To achieve this, two different sets of programming pins are necessary, one for each module.

For the Atmega2560, the programming interface used is AVR ISP (AVR In-System Programming Interface) which operates using the SPI protocol. This requires a 6-pin header which connects to the MISO, SCK, RST, MOSI, GND, and VCC pins of the Atmega2560. A programmer such as the Atmel AT AVRISP Mk II programmer then connects to a computer as well as the 6-pin header on the board and is used to flash new firmware to the Atmega2560.

For the BM70, the configuration is written over UART, however a few additional pins are required to program the BM70's configuration. Thus, another 6-pin header with a connection for P2_0, UART TX, UART RX, GND, and VCC is required to program the BM70's configuration on board. This header was then connected to a USB-to-UART adapter which connects to a PC running Microchip's Firmware Configuration tools to flash a new configuration onto the BM70.

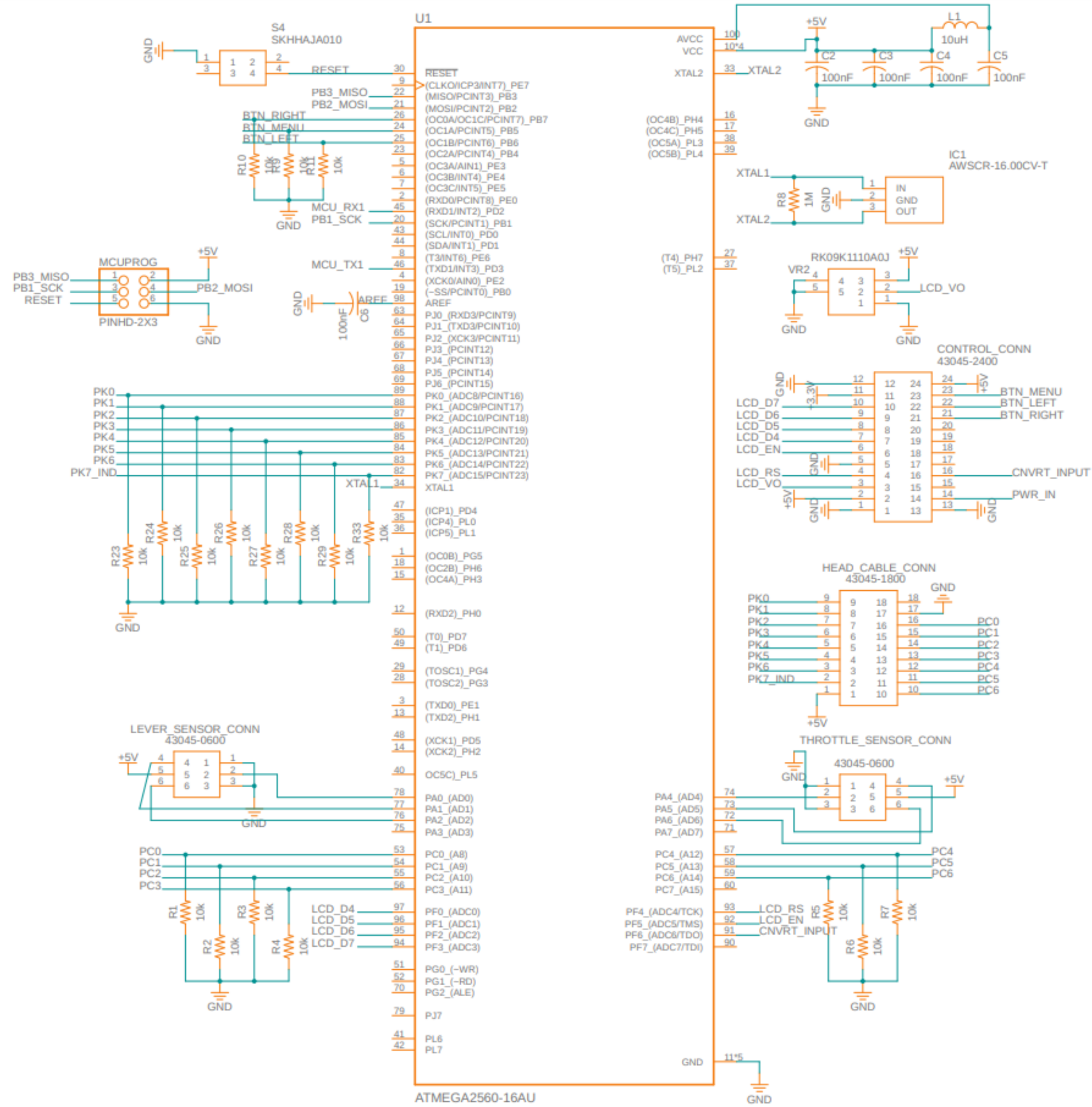
5.1.2.6 Microcontroller

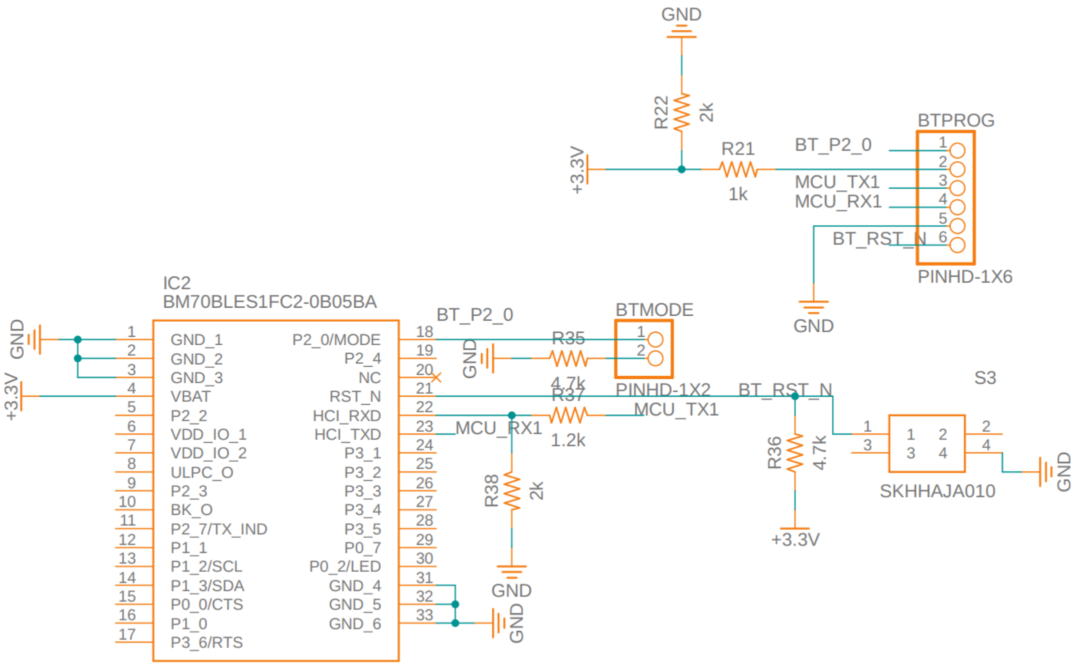
The microcontroller is the central and most essential component of the logic and input system as it manages all other components. The microcontroller which we selected for this purpose is the Atmega2560 which is present on the Arduino Mega 2560 Rev 3 development board. The Atmega2560 offers many benefits, of which a high GPIO count is particularly desirable due to the intense focus on physical input such as buttons, switches, angle sensors, etc. However, in order to function effectively, some additional circuitry is required. By analyzing the Arduino Mega 2560 Rev 3's open-source design documentation as well as the Atmega2560's datasheet, an additional 16 MHz crystal oscillator as well as a low-pass filter were added to the design. The oscillator provides a more precise clock source for timing purposes and the low-pass filter is connected to the supply voltage for the ADC in order to improve the accuracy of voltage conversion. Also, a simple reset button as well as a noise-reducing capacitor on the ADC reference voltage pin improve the usability of the microcontroller overall.

5.1.2.7 Logic and Input Subsystem Schematic

The overall schematic of the Logic and Input Subsystem is shown below.

Figure 8 – Logic and Input Subsystem Schematic





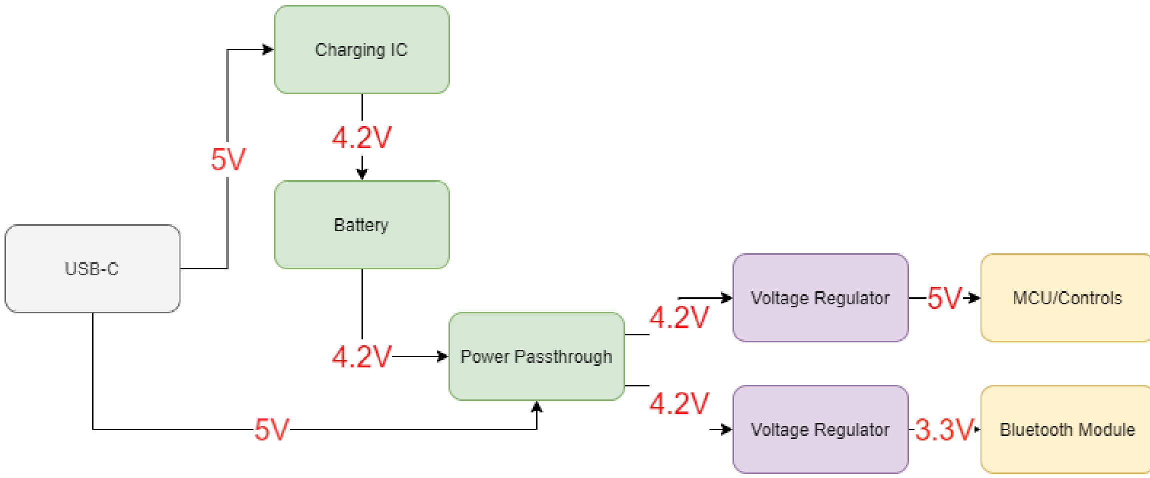
5.1.3 Power Design

The power system uses two forms of supplying power to the rest of our device, those being via USB and a 1-Cell Lithium-Ion battery, as discussed previously. Due to the different subsystems in our device and their individual power requirements, there are two main channels through which power will be delivered, one providing 5V and the other providing 3.3V.

5.1.3.1 Architecture

Before individually dissecting each module in the powering system, it is a good idea to depict the system as a whole and get a clear understanding of orderly functional flow. Below, a flowchart is provided to portray the high-level setup of the power system and a depiction of how the power flows across and into other major subsystems of our device.

Figure 9 – Power System Component Overview



The flow chart above expresses the general structure of our device's power system at full integration. The USB provides power to the charging IC module which will then be regulated and monitored for safe battery charging. When the USB is disconnected, the battery provides power to the two voltage regulators which is then converted and monitored for safe powering of the load system. When the USB is connected, the battery is disconnected from the regulators via the power passthrough circuitry and the USB power source powers the load system. The MCU/Controls and Bluetooth Module represent the load system, which is where much of the reading and writing of data, and overall functionality of the helicopter collective control, takes place. As can be seen, the voltage values are shown because it poses the most crucial input requirement for each device in order to safely and/or optimally operate the system.

Other notable mentions: The voltage values shown represent what the preceding device is outputting and what the device after will and needs to take. The 4.2 voltage value shown is under the assumption that the battery is at its rated max charge. Finally, the color coding is used to simply categorize the system in groups of similar-functionality and/or to represent the major subsystems.

5.1.3.2 USB-C Module

With the increasing popularity and use of USB-C in many types of electronics like storage drives, laptops, and the latest smartphones, USB-C was selected as our choice of USB power. USB-C is a 24-pin reversible connector capable of transmitting both data and power through one cable. Its specifications support high speed data transfer, which is a necessary requirement for todays advanced electronics to work desirably. A version of USB-C known as the USB-C PD can deliver up to 100W of power, which fiercely exceeds its prior 3.1 USB model of only 15W capability. The helicopter collective control will communicate via Bluetooth, and USB data transmission is not needed; it is solely utilized for its power supply capabilities. However, as technology advances and new ideas are born, the 24-pin accessibility and high data-transferring may be of use if changes in our project are to be made in the future. The USB-C will take over and provide power to the battery when it is in need of a recharge and can no longer supply the helicopter collective controller with sufficient power on its own.

5.1.3.3 Battery Charging Module

The battery is intended to be the typical source of power for the helicopter collective control, without the need of having the USB constantly plugged in, therefore its lifetime and reliability are important. A charging IC manages the battery's full functioning profile and ensures that these requirements are satisfied. The charging IC in action is called the MCP73831 and contains 5 pins of functionality. Power is fed through V_{DD} (pin 4) and uniquely distributed inside the IC. Power comes out of V_{BAT} (pin 3) and feeds a 1-Cell battery. To control the rate at which the battery will be charged, the resistor value connected between PROG (Pin 5) and ground can be modified. To monitor the battery's charge level, an LED and resistor can be connected to STAT (Pin 1). Finally, VSS (Pin 2) is connected to ground. The 4.7uF capacitors at both the input and output of the IC are recommended as per typical applications; This helps maintain loop stability, fight off any noise within or around the circuit, and compensate for the inductive nature of the battery.

The Lithium-Ion battery we are using is rated at a nominal 3.7V and a maximum voltage of 4.2V. It was found that charging a lithium-ion battery too fast could result in efficiency losses. To elaborate, when charging a battery at high rates, the voltage of the battery will reach its rated max value, but the max rated capacity value may not have been met, meaning the battery will not last long for the following life cycle. In our case, a 2.0 kΩ resistor was chosen so that around 500mA is fed to our battery, providing a slower rate of charge of roughly around 20% of its total capacity and hopefully minimizing losses. The datasheet for the MCP73831 provides a formula to calculate the value of R_{PROG} that is required to achieve a desired current for the rate of charge. The equation is provided below,

Equation 1 – Rate of Charge Regulated Current Equation

$$I_{REG} = \frac{1000\,V}{R_{PROG}} \qquad (1)$$

Where,

R_{PROG} = Programming resistor (kΩ)

I_{REG} = Regulated current used to determine rate of charge (mA)

As for battery voltage protection, the R_{PROG}, V_{DD}, V_{BAT}, and V_{REG} each have important roles. The typical regulated voltage (V_{REG}) that this IC will try to achieve is 4.2V. The chosen value for R_{PROG}, as mentioned earlier, will provided a regulated current (I_{REG}). There is a correspondence between I_{REG} and V_{REG}, hence, there is a correspondence between R_{PROG} and V_{REG}. The regulated voltage at 4.2V corresponds to an I_{REG} of 10mA. An I_{REG} of 450mA corresponds to a regulated voltage of around 4.183V. Both cases mentioned are observed for an input voltage of 5V. A graph showing this relationship is shown in the figure below.

Figure 10 – Graph of Regulated Charge Current vs Regulated Battery Voltage

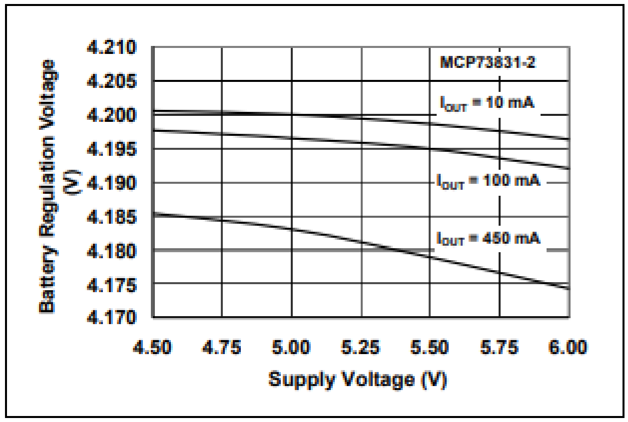
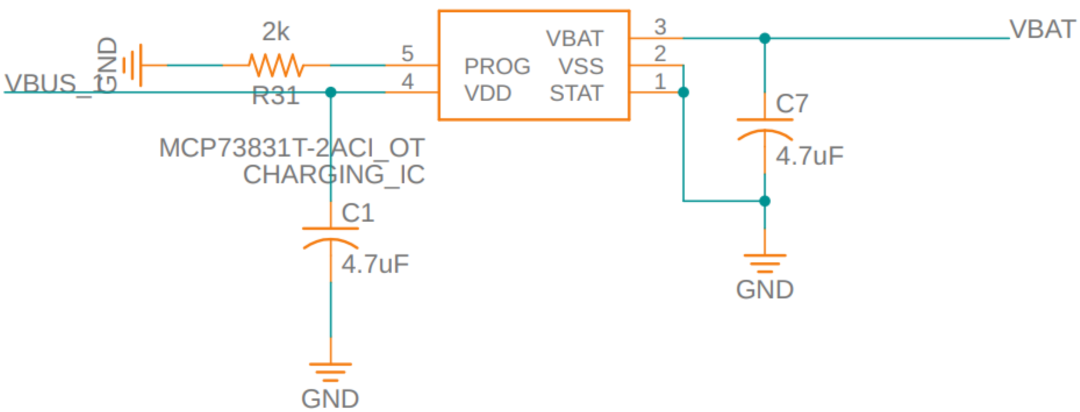


FIGURE 2-1: *Battery Regulation Voltage (V_{BAT}) vs. Supply Voltage (V_{DD}).*

Note. Reprinted with permission from *MCP73831/2 Datasheet*, by Microchip Technology Inc.

Recall that the I_{REG} we are getting for our project is around 500mA for a R_{PROG} value of 2.0 kΩ, which results in a regulated voltage of around 4.183V as shown in the graph. The corresponding supply voltage input is 5V, which is what we are also providing in our project via USB-C. Another safety measure that the MCP73831 provides is by also directly referencing the typical V_{REG} value and V_{BAT} value. If V_{BAT} is greater than (V_{REG} + 100mV), the device will read "No-Battery-Present" and power down. In our case, since the IC will try to regulate the voltage of our battery up to around 4.183V, the IC will shut down as soon as our battery reaches a greater value, which is in range of the max voltage charge that our battery is rated for, therefore, it should not suffer any terrible damage. The battery-charging module schematic is provided below, with component values set to satisfy our project requirements.

Figure 11 – Battery Charging Module Schematic



5.1.3.4 Power Regulation Module

As mentioned previously, there are two channels in which power flows, supplying different sections or subsystems in our device. The two channels are delivering power differently, one is supplying 5V and the other is supplying 3.3V, and this is being achieved with the use of a regulator. To be more precise, the regulator being used is a switch current buck-boost converter called the TPS63020. Voltage is fed through V_{IN} and power is uniquely distributed throughout the various components inside that make-up this IC. Voltage is then supplied out of V_{OUT} to be used for the connected load system. To control and adjust the voltage that is supplied out of the IC at V_{OUT}, the external resistor divider with resistors R1 and R2 can be modified in value. The resistor divider is and must be connected to $V_{OUT,}$ F_B, and Ground. F_B delivers voltage feedback to the IC; this is used as a reference to determine whether proper regulation is occurring. Similarly, PG can also be used as reference to determine if the output power is good, failing, or detect open drain. PS/SYNC can be used to enable or disable power save mode by using logic-high or logic-low. V_{INA} is utilized by connecting a capacitor between it and ground; its functionality serves to ensure that the internal control circuitry is being provided with a stable low-noise voltage supply. It is recommended to connect a 0.1uF ceramic capacitor in close proximity to the V_{INA} pin. If a different value of capacitance is to be chosen, it should not be any more than 0.22uF. L1 and L2 are connection pins for an inductor. The role of this inductor is to minimize conduction losses of the converter. The larger the conductor, the smaller and lower the inductor ripple current and conduction losses, respectively. EN is also used with logic detection to enable or disable the device. If EN is logic-high (1), the device will operate, otherwise, it enters shutdown mode, and all internal circuitry is switched off. GND and PGND are simply connected to ground, which also grounds a selection of internal components (MOSFETS). Finally, as mentioned previously, input and output capacitors are used for maintaining stability and fighting off noise or voltage fluctuations (transient behaviors). It is recommended to use ceramic capacitors as close as possible to the device, on both sides, with a minimum capacitance value of 20uF and a typical value of 66uF at the input and output, respectively.

As stated previously, to achieve a desired voltage at V_{OUT} to use for the load system, the resistor divider simply needs to be adjusted. The datasheet for the TPS63020 provides typical values of voltages and currents across the resistor divider, FB pin, GND pin, and $R_{2,}$ under the assumption that voltage is being regulated properly. With that being the case, the following equation (1) is provided and can be used to determine the value of R_1, when given the V_{OUT} that is desired for a particular load.

Equation 2 – Voltage Regulator Resistor Equation

$$R_1 = R_2 * \left(\frac{V_{OUT}}{V_{FB}} - 1 \right) \qquad (2)$$

Where,

V_{FB} is typically 500 mV

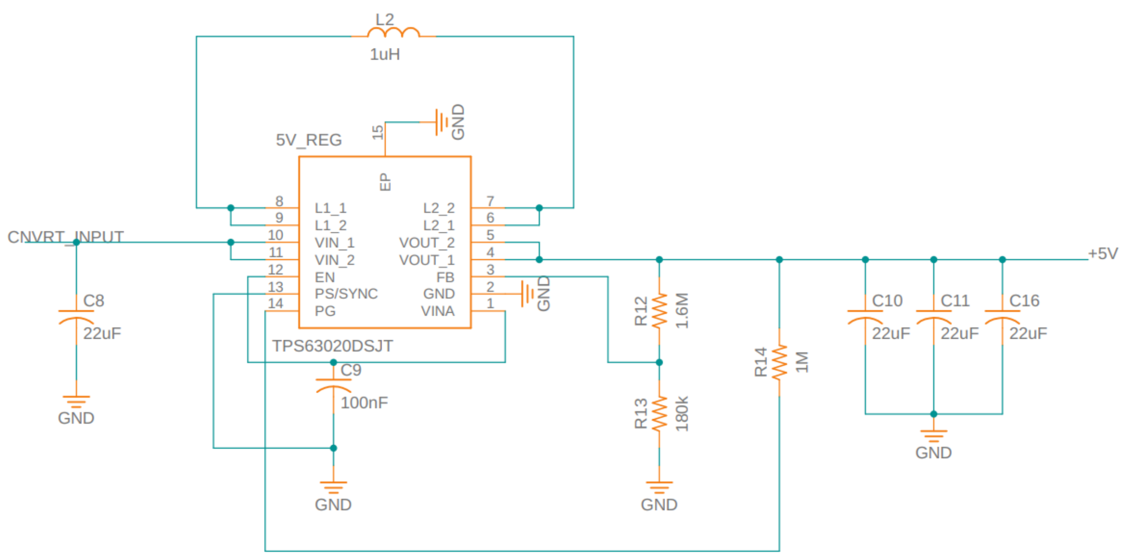
R_2 should be around 200 kΩ

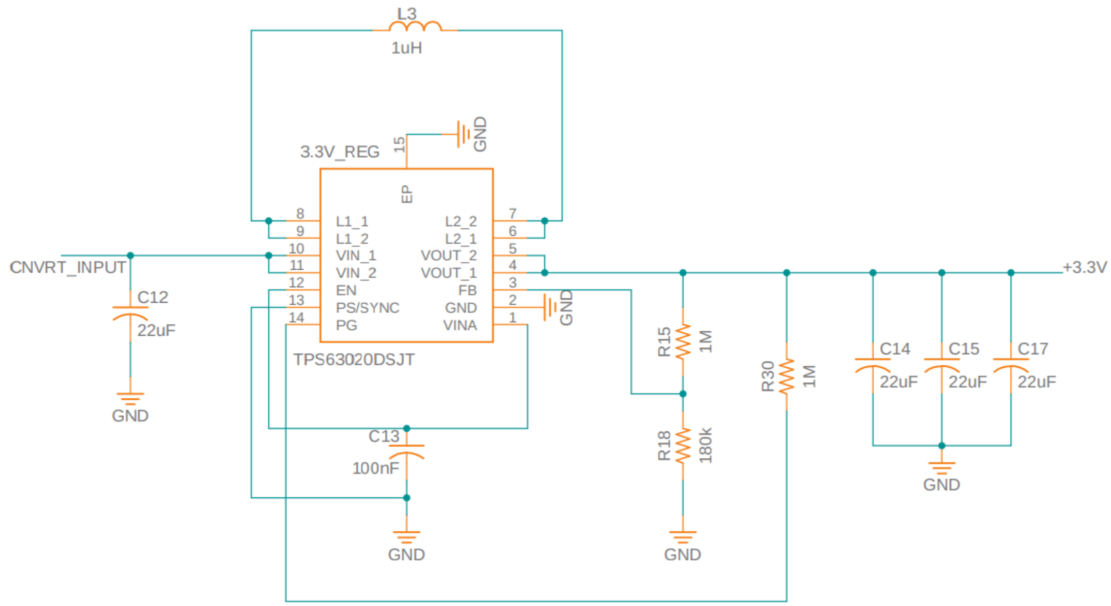
V_{OUT} is the desired output voltage.

The voltage-regulating module schematic is provided in the figure below, with component values set to satisfy our project requirements. Note that the 5-volt outputting regulator which is the one providing power to the MCU and physical controls system. The 5V regulator is primarily increasing the voltage from 4.2V to 5V, hence, the IC is primarily operating in boost-mode.

Recall that in order to control and output a desired voltage value, the values of the resistor divider can be adjusted and used in equation (2). For the second voltage regulator, the value of the R_1 resistor is changed to be able to achieve the 3.3V requirement for the Bluetooth module. In that case, the voltage is being stepped-down from 4.2V to 3.3V, which would indicate that the IC is primarily operating in buck-mode. As a final note, the 4.2V is being provided from the battery, again, under the assumption that it is fully charged. However, this will obviously not be the case all of the time, the battery will discharge as the device is being used but the voltage regulators will maintain their output voltage levels.

Figure 12 – Voltage Regulator Schematics



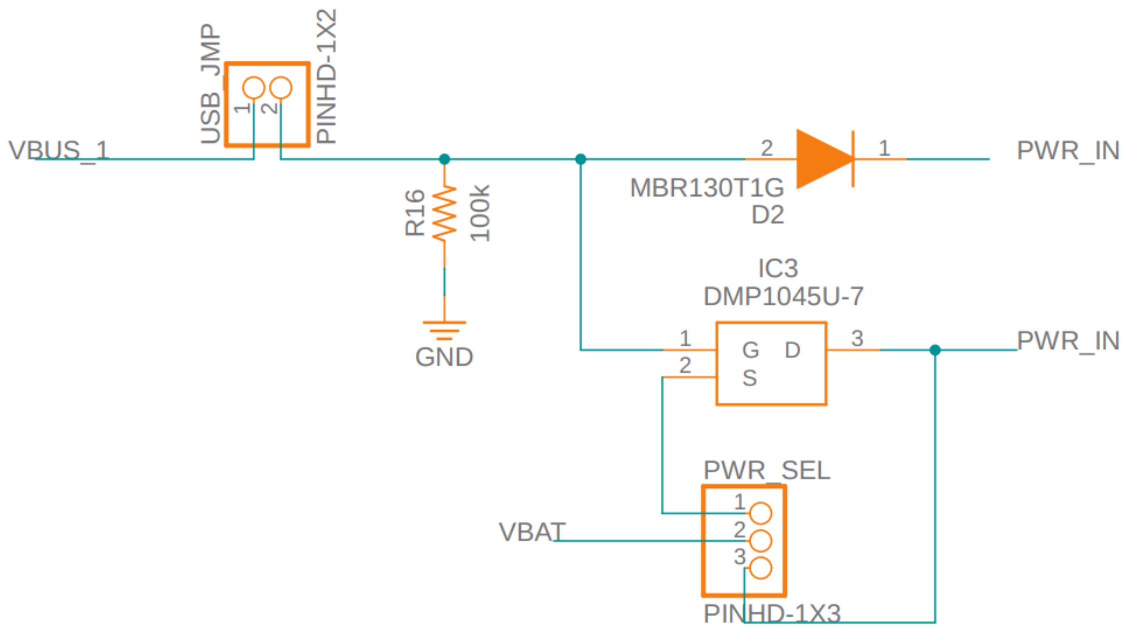


5.1.3.5 Power Passthrough

Due to the need for our system to be used while charging, it was necessary for a power passthrough system to be designed to allow for the system's power source to switch between USB power and battery power. This system is necessary due to how a lithium-ion charging IC works; the charging IC provides a charge current to the lithium-ion battery and simultaneously measures the current drawn by the battery in order to determine its level of charge. If no power passthrough were to be implemented, the charging IC would likely never determine that the battery was charged as the rest of the system's current draw would make it appear as if the battery is pulling a large charge current. Also, without a power passthrough system, the battery would share its charge current with the load system which would likely lead to extremely slow battery charging.

To create this power passthrough system, a simple circuit involving a P-Channel MOSFET, a Schottky Diode, and a resistor was created. When the USB-C power is not plugged in, the P-Channel MOSFET allows current to flow from the battery to the load system. When the USB-C cable is plugged in, the MOSFET acts as a switch and cuts off current flow from the battery to the system. The Schottky diode is present to prevent current from backflowing from the battery into the USB-C cable, and the resistor is present to pull down the gate of the MOSFET when the USB-C cable is removed to ensure that the gate does not "float". The charging IC's power input comes directly from the USB-C cable, which means that as soon as the USB-C cable is plugged in, the battery is disconnected from the load and the charging IC is able to begin charging the battery in isolation from the rest of the system while the USB-C power source takes over the role of primary power source. Shown below is the schematic of this power passthrough system.

Figure 13 – Power Passthrough Schematic



5.1.4 PCB

A printed circuit board (PCB) was created for integration into our fully integrated working electronic device. PCBs have been brought along with the advancement of technology and are extremely common and useful in modern electronics. PCBs relieve the stress of dealing with messy and unreliable wiring and component assembly, as well as sizing a complex circuit system down to a small scale. The composition of PCBs also plays an important role in providing a secure environment in which the electrical assembly can safely function. The composition of PCBs is generally made up of 4 layers of different materials and properties. The top layer is called a silkscreen layer, which is used for outlining component locations, labeling, symbols, and any lettering and numbering for a better reading and understanding of the board. The second layer is called the Soldermask and it represents the typical green color on PCBs. The purpose of the soldermask is to insulate the metal layer underneath of it. The metal traces are covered with the soldermask to prevent accidental and/or unwanted contacts with other objects that may be conductive, but the connection pins and any SMD pads are left exposed for clean assembly and soldering. As mentioned, the third layer is a metal layer, typically copper, and it is laminated onto the board using heat and adhesive. Different number of layers can be added to increase the thickness of metal, which is typically done for higher-power and more advanced circuity. This layer allows all components to communicate with each other. The fourth and final layer is the substrate layer and is known as the base layer. The base layer is the canvas holding all other layers and components together, therefore it is important to choose the right material, as wear and tear quality will vary. The substrate material that is typically used is FR-4, which is a fiberglass that is bonded with epoxy resin, providing thermal protection.

5.1.4.1 PCB Manufacturers

PCBs are available for purchase from a wide variety of manufacturers. The importance of picking a vendor is equal in magnitude to designing our project. It is suggested to investigate each manufacturer and consider their credibility and overall quality as a company. Some good criteria to follow are to look at their experience and how long they have been in the industry, the manufacturing equipment and materials being used, certifications, if they do any quality testing during the manufacturing procedure, quality of customer service and their engagement, cost of fabrication and shipping and any other fees, shipping time, etc. Although the mentioned criteria should be followed fully and truthfully, to ensure quality and reliability to our device, we mainly focused on cost, shipping time, and design flexibility, due to budget and time constraints. Below is a list of the following PCB manufacturers that were considered.

5.1.4.1.1 PCBWay

PCBway is an experienced and popular PCB manufacturer that was founded in 2014 and is located in China. They have a good reputation with on-time delivery, quick turnarounds, quality customer service, and decent values. PCBWay is known to be flexible with component selection, where other manufacturers might lack.

5.1.4.1.2 JLCPCB

JLCPCB is one of the largest PCB manufacturers in China and has been at the spearhead of the PCB industry since its establishment in 2006. JLCPCB is well known to be convenient for orders at small and medium quantities, as well as very cheap pricing and good quality boards. They also have a good production and shipping status management system that is favored by their community.

5.1.4.1.3 OSH PARK

OSH PARK is based in the U.S. and was founded in 2012. They are focused on hobbyist needs for cheap prices, good quality PCBs, fast delivery, and convenient quantities per order. They hold a good reputation for on-time and early arrivals, as well as free shipping, however, other shipping fares are available. Whether an amateur hobbyist is looking to order just a few units to prototype a simple design, or a skilled designer with more complex projects wanting to buy in bulk, OSH PARK provides several service packages that can satisfy either needs. A note on OSH PARK: The cheapest package was chosen for comparison in the table below, which comes at the cost of a longer delivery time, however, orders are typically received earlier than later; According to OSH PARK, 95% of orders are received in 8 days. Also, the package only includes 3 units, instead of the 5-unit amount that would be received by the other manufacturers.

Table 16 – Comparison of PCB Manufacturers

Manufacturer	PCBWay	JLCPCB	OSH PARK
Build Time	1 Day	1 - 2 Days	N/A
Shipping Time	3 - 7 Days	2 - 4 Day	9 – 12 Days
Unit Cost	$45	$9	$30
Shipping Cost	$25	$12.80	(Included)
Total Cost	$70	$25.80	~ $30

From the table above, JLCPCB was selected as the manufacturer for our PCB. Although build time may take longer, its total cost comparison with the other manufacturers is far cheaper, which aligns with our goal of minimizing the overall cost of our project. When retrieving the dollar amounts, we used the example that our PCB would be 150mm x 150mm in size, have 2 layers, with an order quantity of 5 units. The shipping method selected was DHL, which came out to be the second cheapest method and a faster delivery. These settings were used for all manufacturers mentioned above to create the above comparison.

5.1.4.2 PCB Assembly and Fabrication

The design and assembly of the PCB was conducted in Fusion 360, which directly integrates and is compatible with Eagle CAD. All circuit components were combined into one overall schematic. Each subsystem was given a name relating to its function or role of operation, and net classes were defined for overall communication between subsystems and devices. Once the schematic was completed, it was translated from an electronics schematic file type (.fsch) to an Electronics Board file type (.fbrd), which is the PCB layout. In the PCB layout, all components and devices are shown in their PCB footprint form and are arranged and placed on the PCB board. A plan of execution was followed for the PCB design with the intent of better achieving an optimal and clean working design. The steps are listed and summarized below:

- Determine and set the number of layers for the PCB (considering possible effects on manufacturing cost). Define each layer with its corresponding purpose, for example, power and ground layer.
- Learn and practice using useful tools and commands that are provided. This includes routing techniques, selecting, moving, and placing objects, adjusting default values (such as the widths of traces and drill size of vias), and more. This effort was made to efficiently work through the design.
- At the start, consider different arrangements of components to reduce crossover of connections and lower complexity of design. The arrangements do not necessarily need to follow the electrical schematic design; however, it is important to determine which components must stay within range of their corresponding subsystem, such as decoupling capacitors. In our design, most subsystems were arranged as they were in the electrical schematic design.

- Consider important sizing of different traces. It is suggested to use a larger width value for power and ground traces, which may help with efficiency in transporting power across the circuit, reducing resistance, thus, reducing voltage drops, maintaining lower temperatures, and more.
- Consider the length of tracing between components. It is suggested to keep traces short to reduce any losses that may occur. If using vias for under or cross-tracing, it is important to keep the path short. It is important to mention that some devices must be strategically placed further away from certain components and/or devices, due to their nature of operation, therefore, these are cases where long traces cannot be avoided.
- Consider categorized tracing. In our design, the top layer was dedicated for signal and power traces, except for some that needed to run through the bottom layer. The bottom layer was dedicated for ground traces. This helped in developing a more organized design and in providing a common ground.
- Consider thermal relief and techniques to achieve better thermal management. Vias were used for transitioning traces between layers, when needed. However, vias provide thermal relief between one side of the board to the other. Therefore, placing additional vias and minding their hole size could prove to be beneficial. Certain placement of extra copper around the surface of the layer could also provide thermal relief. When adding certain components or adjusting certain settings such as for copper fill, layers, and vias, it is important to look for any thermal relief application that is available and include it. Finally, setting a good perimeter of restriction will keep signals and components at a safe distance, which will provide some ventilation for heat dissipation. Thermal relief is an important aspect to consider because it will inherently affect the efficiency of components and the overall system.

The final PCB design for our project was completed and an image of it is shown below, as well as a 3D render of the completed board:

Figure 14 - PCB Layout

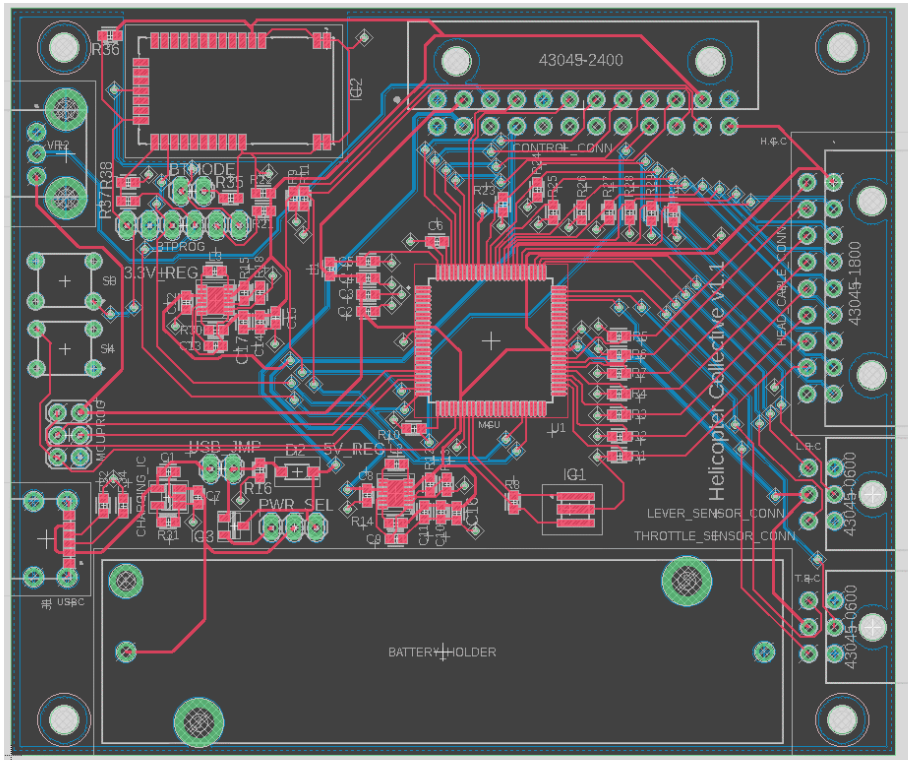
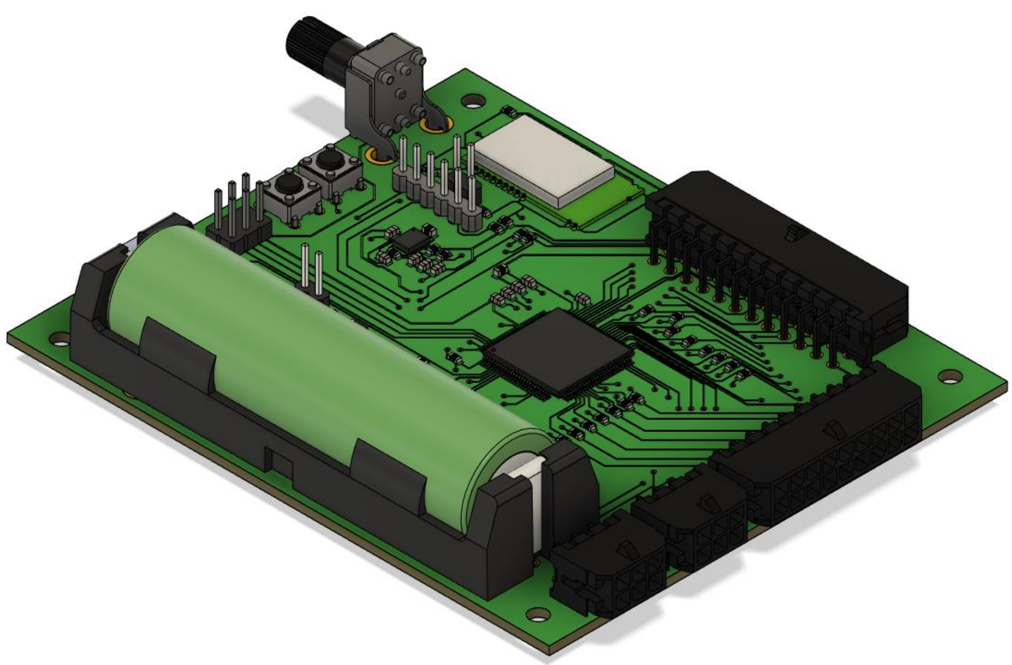


Figure 15 - 3D Render of Completed PCB



Aside from the electrical components, additional considerations were taken into account, such as the location of the PCB on the base of the helicopter collective unit to ensure user-accessibility and device safety. The location of the PCB also took into consideration the wires that will run throughout the collective unit and connect to other components, such as the LCD and switch/button controls. The orientation of the Bluetooth module was set at the very top of the PCB, and interactive components, including the USB-C input, were placed on the left of PCB for ease of access. Finally, mounting holes were included at the corners of the PCB; the holes are 3.2mm in size, which corresponds to the M3 screws that will be used to mount the PCB on the back panel of the helicopter collective base.

5.1.4.2.1 Bill of Materials

Table 17 – PCB Bill of Materials

Item Name	Part Number	Price Per Item	Quantity	Cost
Buck-Boost Converter	TPS63020DSJR	$ 0.77	2	$ 1.54
18650 Battery Holder	1043P	$ 2.50	1	$ 2.50
Battery Charging IC	MCP73831T-2ACI_OT	$ 1.33	1	$ 1.33
24-pin Molex Female	43045-2400	$ 2.67	1	$ 2.67
18-pin Molex Female	43045-1800	$ 0.89	1	$ 0.89
USB-C Port	TYPE-C-31-M-17	$ 0.18	1	$ 0.18
6-pin Molex Female	43045-0600	$ 0.33	2	$ 0.66
6mmx6mmx4.5mm Switch	SKHHAJA010	$ 0.10	2	$ 0.20
Microcontroller Unit	ATMEGA2560-16AU	$ 8.73	1	$ 8.73
Bluetooth Module	BM70BLES1FC2-0B05AA	$ 7.38	1	$ 7.38
10k Potentiometer	RK09K1110A0J	$ 0.62	1	$ 0.62
16MHZ Resonator	AWSCR-16.00CV-T	$ 0.82	1	$ 0.82
Total Price				$ 27.52

5.2 Software Design

In order to implement the functionality of the collective, a significant amount of software design was required. This includes design decisions regarding choice of software tools, configuration of the Bluetooth module to be recognized as a Human Interface Device, software to take in various forms of input, and software to send and receive data between the microcontroller and the Bluetooth module. All of these design decisions are explored in-depth throughout the following sections in order to explain both the design choices themselves and the reasons behind them.

5.2.1 Software Tools

Selecting the appropriate suite of software tools is essential when developing any project as they can significantly affect the efficiency and feasibility of the design. In terms of the helicopter collective, the three main areas which require software tools are the Human Interface Device configuration, Bluetooth Configuration, and Microcontroller software development. Each of these areas require specific software tools which are intended for the task at hand and are essential in the creation of the helicopter collective unit as a whole.

5.2.1.1 USB-IDF HID Descriptor Tool

As mentioned previously, the HID Report is the essential piece of data which must be transmitted between the helicopter collective unit and the connected host device in order to communicate gamepad inputs to the flight simulator. These Reports must adhere to a specific format in order to communicate to the connected device what features the Human Interface Device has and in what data formats and value ranges the data is constrained to. This format is known as the Report Descriptor, and it is necessary in order for the connected device to properly parse and understand the incoming Report data. In order to develop these HID Reports, it is possible to utilize solely the HID Specification and Usages described by the USB-IDF and generate the descriptor by hand, but this method is prone to errors and is quite arduous. To solve this problem, the USB-IDF also publishes a free tool called the HID Descriptor Tool which allows you to create and validate your own custom Report Descriptor through an easy-to-use user interface that is guaranteed to be compliant with the HID specification. This tool was crucial in forming the HID Report Descriptor utilized by the helicopter collective to be recognized as a gamepad input.

5.2.1.2 Microchip IS187x BLEDK3 Bluetooth UI Configuration Tool

The BM70 Bluetooth module which we chose to use as our wireless module implements a complete Bluetooth Low Energy stack and byte-based command set. While it is possible to configure the module entirely through the use of UART-based byte-formatted commands, this method of configuration is extremely tedious, error-prone, and work-intensive. Also, it is difficult to swap-out configurations quickly as each individual option or field must be reconfigured manually. To avoid this task, Microchip provides a software tool called the IS187x BLEDK3 UI tool which allows the developer to much more easily configure the BM70 for Bluetooth Low Energy operation. Not only does this tool let you individually select configuration options for all relevant Bluetooth Low Energy related fields, but it also allows configurations to be saved and loaded at will through the use of simple text documents which greatly improves testing and productivity. Most importantly, the UI tool allows the Bluetooth Low Energy Services to be easily edited, which is normally a complex and difficult task which involves juggling permissions, values, and documentation. In the UI tool, these Services can be drag-and-dropped to be added or removed, and the permissions and values of these Services can be edited with simple checkboxes and textboxes. This is especially important for our purposes as the HID Service requires custom configuration in order to properly function.

5.2.1.3 Microchip BLEDK3 Manual Test Host Emulation Tool

As mentioned earlier, the BM70 Bluetooth module is traditionally interacted with via UART-based byte commands that must be individually crafted, sent, and interpreted through the use of documentation. As such, testing the BM70 without the use of a software tool would involve manually scanning through the byte command set from the User's Guide, calculating your own byte-command, and using the User's Guide to interpret the associated byte response from the BM70 over UART. While this is how the microcontroller will actually communicate with the BM70 Bluetooth Module in the final product, it is unnecessarily frustrating and time-consuming for manual testing. Thus, the BLEDK3 Manual Test Host Emulation Tool allows the developer to utilize a simple, user-interface based tool to send commands to the BM71 and interpret their responses quickly and easily. This is essential for both testing the Bluetooth Low Energy configuration written to the BM70 using the UI Configuration Tool, as well as for sending manual test inputs to ensure that the HID Reports are interpreted properly by the connected device. This tool was also used to help determine what commands, and in what format, must be added to the Microcontroller software in order to interact with the BM70 and function properly.

5.2.1.4 Arduino IDE (Integrated Development Environment)

The Arduino IDE is an open-source development environment for Arduino development boards. The Arduino IDE allows the developer to create programs known as "sketches" which are written in C/C++ and may be uploaded to the microcontroller or development board. The main advantages of the Arduino IDE are the extensive community support, documentation, and library support. As the Arduino IDE is a community-driven, open-source development environment, the existence of things such as user guides, help forums, and openly available resources for learning how to use it are widely available and are extremely helpful. This also means that the documentation relating to all integrated functions, libraries, and features are openly available, constantly maintained, and easy to understand. Most importantly, due to its popularity, the availability of libraries for the Arduino IDE is unparalleled. This significantly reduced development time requirements as rather than recode common functions and features from scratch, well-tested and proven implementations were quickly and easily imported into our project. Finally, and most importantly, as the Arduino Mega 2560 Rev 3 was used as our development board and we are using its on-board Atmega2560 as our microcontroller, we were able to quickly and easily iterate our software design and be sure that the software would translate over directly to our finished design and PCB. This was an invaluable feature, as it allowed for software development and breadboard testing to begin much earlier than otherwise would be possible.

5.2.2 HID Configuration

In order for the helicopter collective to be recognized by a connected device as a gamepad and therefore be capable of controlling a flight simulator, it must be recognized as a Human Interface Device, or HID for short. A Human Interface Device interacts with a connected Host device which receives data in the form of HID Reports. The information regarding creating, populating, and sending these reports are available in the Device Class Definition for HID published by the USB-IDF. For our purposes, the helicopter collective must be able to present itself as a gamepad with multiple buttons and 2 axes (a throttle axis and collective lever axis) and send data regarding these inputs to the Host Device.

5.2.2.1 Report Descriptor

In order to send data between the Human Interface Device and the Host device, a Report Descriptor must be formed which details the various logical and physical features available on the HID. In general, there are two kinds of HID devices: Boot Report devices and standard Report devices. Boot Report devices are generally mice and keyboards and have fixed Report Descriptors assigned by the USB-IDF so that their Reports can be parsed during Boot (such as during computer startup to enter BIOS). However, in our case, the helicopter collective must report as a standard Report device and therefore have a custom Report descriptor which the connected Host then interprets. Creating an HID Report Descriptor involves creating a description of the various Usages available on the HID. Usages are values defined by the USB-IDF which are interpreted by the Host in order to determine the format and meaning of the various data reported by the Human Interface Device. Alongside each Usage is a description of the Usage Range and Logical Range as well as a description of the Report Size and Report Count associated with the Usage. The Usage Range describes the number of items available for that usage; for example, when defining many buttons or many keyboard keys. The Logical Range describes the range of possible logical values a Usage may report; for example, a keyboard key or button may report a 0 (unpressed) or 1 (pressed), whereas more complex Usages such as an axis may report values in a large range, such as from 0 to 1023. The Report Size indicates the number of bits in the HID Report which will be used to report that Usage’s value to the Host. The Report Count indicates how many fields of that size will be reported to the Host; for example, each axis may have a Report Size of 16 bits and there will be one 16-bit field per axis, so a device with 2 axes would have a Report Count of 2.

Shown in the Figure below is the HID Report Descriptor designed for the Helicopter Collective. As can be seen, in order to be recognized as a gamepad, the collective is described as a Generic Desktop peripheral of the Game Pad variety. It then reports that it has Buttons 1 through 16, which can each have a value of 0 (unpressed) or 1 (pressed), as well as 2 axes (X and Y) which each have can have a value between 0 and 1023. It also indicates that all 16 button inputs will be sent in a single 16-bit field with each bit representing 1 button. However, the two axes will be reported as 2 separates 16-bit fields. Although the axes values are only from 0 to 1023 and therefore do not require 16 bits, the HID specification requires that Report data must be sent in multiples of 8 bits, therefore the additional bits in the axis fields are used as padding to satisfy this requirement.

Figure 16 – HID Report Descriptor

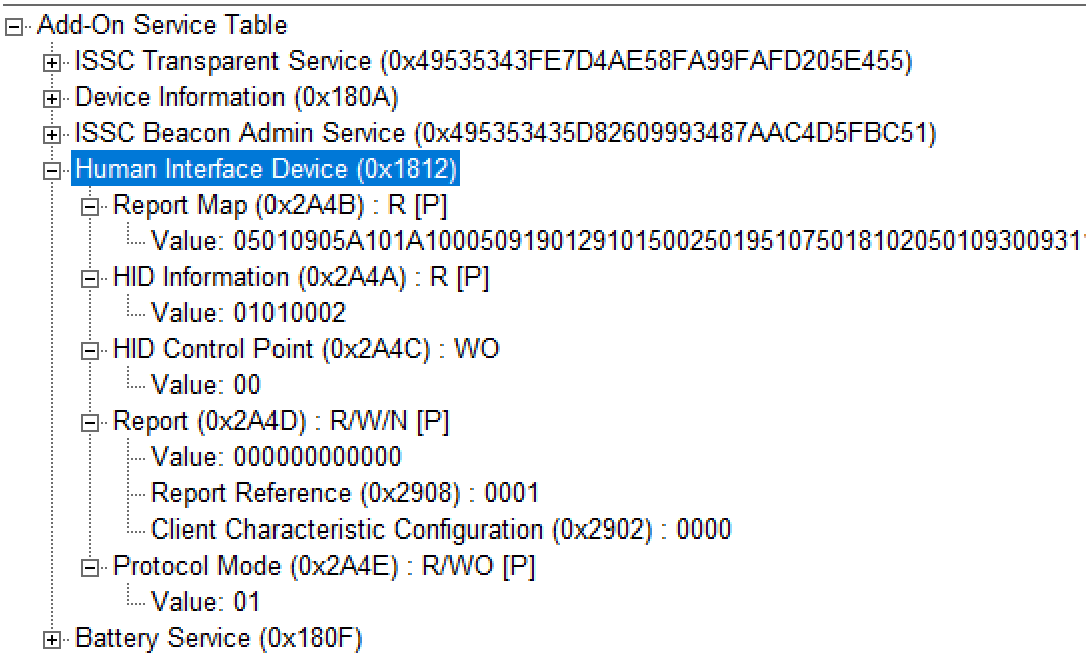
```
USAGE_PAGE (Generic Desktop)                05 01
USAGE (Game Pad)                            09 05
COLLECTION (Application)                    A1 01
  COLLECTION (Physical)                     A1 00
    USAGE_PAGE (Button)                     05 09
    USAGE_MINIMUM (Button 1)                19 01
    USAGE_MAXIMUM (Button 16)               29 10
    LOGICAL_MINIMUM (0)                     15 00
    LOGICAL_MAXIMUM (1)                     25 01
    REPORT_COUNT (16)                       95 10
    REPORT_SIZE (1)                         75 01
    INPUT (Data,Var,Abs)                    81 02
    USAGE_PAGE (Generic Desktop)            05 01
    USAGE (X)                               09 30
    USAGE (Y)                               09 31
    LOGICAL_MINIMUM (0)                     15 00
    LOGICAL_MAXIMUM (1023)                  26 FF 03
    REPORT_SIZE (16)                        75 10
    REPORT_COUNT (2)                        95 02
    INPUT (Data,Var,Abs)                    81 02
  END_COLLECTION                            C0
END_COLLECTION                              C0
```

5.2.3 Bluetooth Low Energy Configuration

As HID is traditionally a USB-related specification, in order to successfully appear as an HID and send data over Bluetooth Low Energy requires separate configuration in order to adhere to the HID over GATT specification published by the Bluetooth SIG. The HID over GATT specification indicates what services and information a Bluetooth Low Energy device must implement in order to appear as a Human Interface Device to the Host device. Thus, from the Hosts point of view, it believes it is effectively connected to a USB device implementing HID.

In order to implement HID over GATT, the Bluetooth Low Energy Device must implement the Device Information Service, Human Interface Device Service, and Battery Service. These services directly map Bluetooth Low Energy characteristics to HID data that is interpretable by the Host device. The Device Information Service describes things such as the Manufacturer of the device as well as device information and the appearance of the device, which affects how the device is handled by the Host. The Battery Level Service simply reports a value between 0 and 100 representing the Battery Percentage of the Bluetooth Low Energy device to the Host. Finally, the Human Interface Device Service actually contains the HID Report Descriptor in the Report Map characteristic as well as the HID Report data in the Report characteristic which is read by the Host to transmit inputs. Shown below is the relevant Human Interface Device service configuration which was developed by following the HID over GATT specification, the Bluetooth Core Specification, and the HID sample provided by Microchip for the BM70 and allows for HID data to be transmitted over Bluetooth Low Energy.

Figure 17 – Bluetooth Low Energy GATT Service Table Configuration

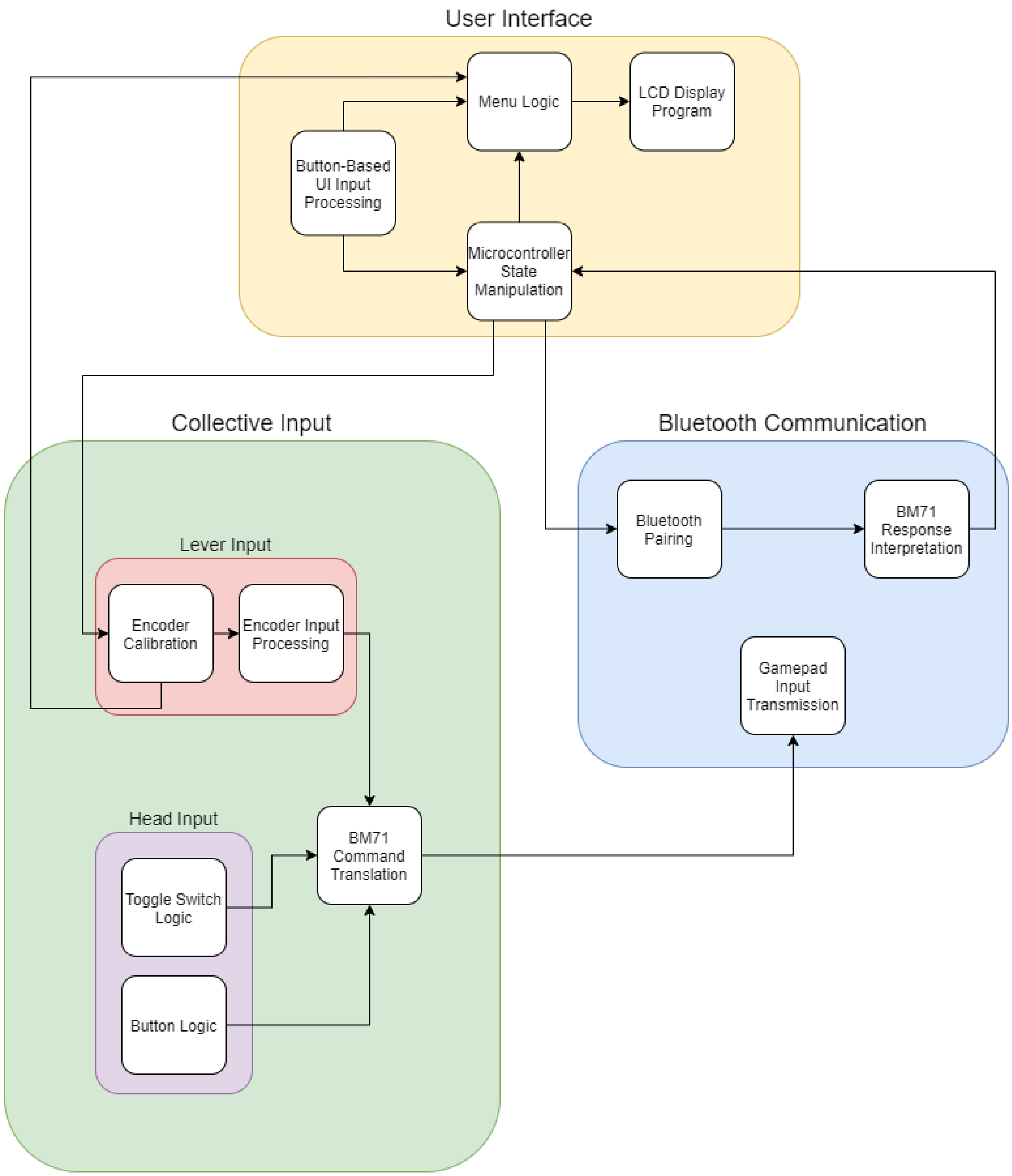


5.2.4 High-Level Software Design

Before delving into the low-level design decisions and reasoning behind implementation choices for the programming and software aspect of this project, it is first necessary to look at software design from a high-level perspective in order to structure the software in a more organized manner. This helps to not only logically separate each portion of the software, but it also serves as a sort of guide for concrete software implementation such that high-level functionality of various parts of the software that will have to be implemented can be considered in context, resulting in a cleaner and more efficient final result.

At the absolute highest level, the overall software design can be broken down into three sections: user interface software, collective input software, and Bluetooth communication software. These are the three largest logical classifications by which the software was designed and implemented. The user interface portion of the software governs functionality of the collective base controls and LCD, including presenting information to the user on the LCD and operating functions of the collective outside of direct gamepad input. The collective input software is responsible for the input, processing, and transmission to the Bluetooth module of any and all forms of input designed to be utilized as gamepad input on the computer. This includes encoder calibration, encoder input and input processing, button/switch input logic, and translating raw input to commands that can be sent to the BM70 for Bluetooth transmission. The Bluetooth communication software is solely responsible for managing the process of Bluetooth communication via the BM70 with a computer. This includes initiating pairing, sending formulated commands to and from the BM70, and interpreting responses from the BM70. These high-level sections of software, their general functionality, and the interactions between them can be seen in the figure below.

Figure 18 - High-Level Software Block Diagram



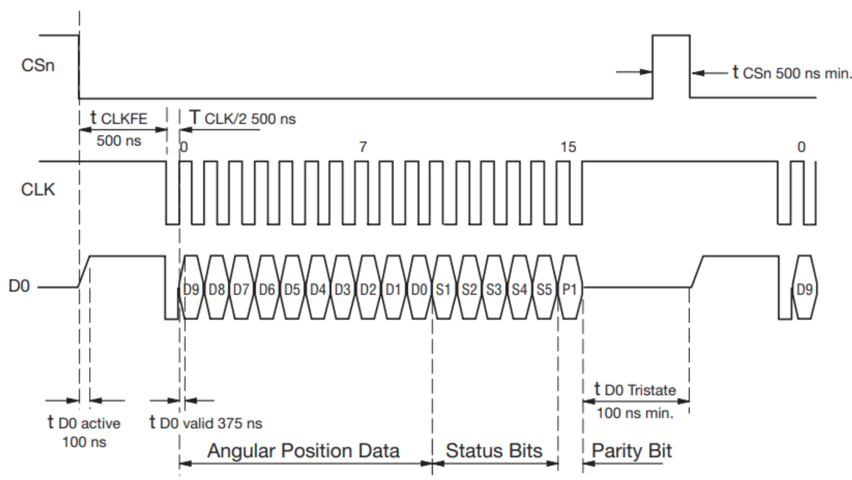
5.2.5 Encoder Input

The collective as a whole has two different encoder inputs: the collective lever angle sensor and the collective lever throttle. As explained previously, both of these inputs are absolute rotary encoders, specifically the EMS22A50-B28-LS6 encoder from Bourns Inc. When programming the logic to detect and process the input from these encoders, there are multiple facets of design that were be considered, each of which will be explored throughout this section.

5.2.5.1 Encoder Input Processing

The absolute rotary encoders are each connected to three pins: a clock select pin, a clock pin, and a data pin. The purpose of the clock select pin is to signal the encoder to prepare to send data, the clock pin manages the rate at which data is sent, and the data pin is the ping along which individual bits are sent as digital input to the governing microcontroller for processing. This entire process must be handled by software in order to facilitate proper sending of data between encoder and microcontroller. The process that is followed when sending data can be seen in the figure below.

Figure 19 – EMS22A Absolute Encoder Timing Diagram



Data Content	Description
D9:D0	Absolute angular position data
S1	End of offset compensation algorithm
S2	Cordic overflow indicating an error in cordic part
S3	Linearity alarm
S4	Increase in magnetic magnitude
S5	Decrease in magnetic magnitude
P1	Even parity for detecting bits 1-15 transmission error

Note. Reprinted with permission from *EMS22A Non-Contacting Absolute Encoder* (p. 2), by Bourns Inc.

Firstly, as can be seen in the figure above, the clock select pin (CSn) falling from high to low serves as a signal for the encoder to begin the process of sending angular position data.

After this has been detected, a single period of the clock (CLK) indicates the beginning of data bits being sent along the data line (D0). This is then followed by a series of 16 data bits, of which the first 10 represent the absolute angular position data as a binary value in the range of 0-1023. These data bits are sent along with each subsequent period of the clock.

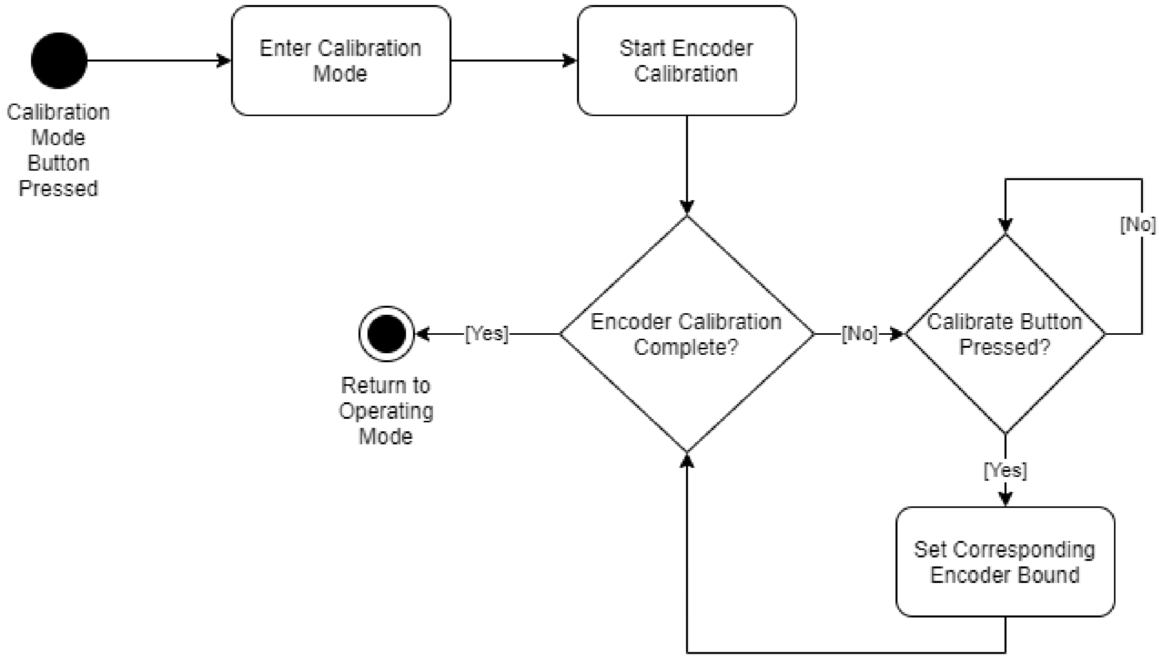
The entire process noted above is handled directly by a software implementation that manually alters the value of the clock select and clock pins in order to govern the sending of data. The clock select is manually toggled between low and high to begin a transmission, and the clock is then toggled between low and high periodically in order to send each bit of data. As each of the first 10 data bits are sent, a process occurs in which a temporary value (starting at 0) is bit shifted to the left at each step and undergoes a bitwise OR operation with the incoming bit of data (0 or 1). This process is repeated at each clock cycle in order to effectively build up one consolidated binary value containing a value in the range of 0-1023 representing the reported value from the encoder. The remaining data bits which do not represent absolute angular position data are then simply skipped over by cycling the clock as they are not necessary for our implementation.

5.2.5.2 Encoder Calibration

When reporting values of axis-based input as a gamepad to a host computer, axis values range from 0-1023. This, in theory, would be very convenient as the values reported from the absolute encoders are in this same range and thus would in a normal case be able to be reported to the computer directly as it is read from the encoder, as the encoder also reports values in a range from 0-1023. However, for our purposes, this is not quite the case. The range of 0-1023 reported from the encoder represents the range of a full rotation of the encoder, in which the entire 360° rotation of the encoder is mapped to values between 0 and 1023. In the specific application of the collective, however, neither the collective lever itself nor the lever throttle control will operate in a range of 360°. This being the case, it became necessary to implement some form of calibration operation for values in the limited range of motion of these controls to be properly reported to the host computer in the desired range of 0-1023.

From a software design perspective, the encoder calibration process consists of three major parts: a function to enter calibration mode, a function to calibrate the encoders, and a function to convert raw input from the encoders to a calibrated value in the range of 0-1023. The function to enter calibration mode is triggered by a button on the collective base and essentially enters a loop that waits for the user to finish calibrating the encoders. The function to calibrate the encoders also operates by pushing a button on the collective base, and functions by waiting for 2 button presses in total: one press for the lower bound of the lever encoder's range (lever all the way forward) and for the lower bound of the throttle encoder's range (throttle all the way off), and another press for the upper bound of the lever encoder's range (lever all the way back) and the upper bound of the throttle encoder's range (throttle all the way on). These values are each recorded in order to serve as lower and upper bounds when extrapolating a final calibrated value from raw input data. The actual function to convert raw input from the encoders to a calibrated value is performed every time the microcontroller reads in input from the encoders. Once a raw value from the encoder has been modified based on the calibration, it can then be sent along to the Bluetooth module in order to serve as gamepad input. The high-level process by which encoder calibration will take place can be seen in the figure below.

Figure 20 – High-Level Calibration Process Diagram



5.2.5.2.1 Calibration-Based Value Modification

As mentioned previously, once the encoders on both the collective lever and the throttle have been calibrated, the raw values sent from these encoders to the microcontroller must undergo modification in order to expand the raw values reported from the limited physical range of motion of each encoder into a proportional value between 0 and 1023. This process occurs in a few necessary steps implemented into a function that converts raw encoder values to values that will be reported to the Bluetooth module.

The first of these steps is to determine if either of the ranges determined when calibrating the encoders rolled over zero. For example, if the lower bound of the lever encoder set when calibrated was at a value of 800 and the upper bound was set at a value of 200, it is clear that at some point between these two values the encoder rolled over 0 in its position. This is determined in software simply by checking if the upper bound of either range is lesser than or equal to the lower bound of the same range, and a corresponding flag will be set for each encoder representing if there was a rollover to zero. This is absolutely necessary information when performing the actual algorithm to convert from a raw encoder value to a final encoder value.

The second step is to determine the size of the range of values that have been calibrated to be output from the encoders. Depending on if there was a rollover to zero event, the equation used to determine this size can be one of two different equations, each shown in the equations below.

Equation 3 – Normal Equation for Calculating Size of Encoder Range

$$Calibrated\ Size\ of\ Encoder\ Range = Upper\ Bound - Lower\ Bound$$

Equation 4 – Rollover-to-Zero Equation for Calculating Size of Encoder Range

$$Calibrated\ Size\ of\ Encoder\ Range = Upper\ Bound + (1024 - Lower\ Bound)$$

As can be seen in Equation 1, this calculation is very simple in the normal case, as the difference between the upper bound and lower bound in the normal case is inherently the size of the encoder's range of values. However, in the rollover-to-zero case, rather than subtracting the lower bound from the upper bound, it is necessary to instead find the difference between 1024 and the upper bound, and then add on the value from the lower bound. This essentially breaks the determination of the size of the range of the encoder's values into two parts, the part below 0 and the part above 0, and combines them into a single value. This value is then used in order to determine what proportion of the full range of encoder values the calibrated range represents.

The third step is then to shift the incoming raw value from the absolute encoder down such that the bottom of our range is effectively zero. The purpose of this is to facilitate direct alteration of values in our range to a value from a range that starts at 0 (the max range of 0-1023) by utilizing a simple proportion. In order to perform this step, there are three different cases: no rollover-to-zero, rollover-to-zero when raw encoder value is at least equal to lower bound, and rollover-to-zero when raw encoder value is less than equal to the lower bound, each of which require a different equation to shift the value. These equations are each shown in the equations below.

Equation 5 – Normal Equation for Shifting Encoder Values

$$Shifted\ Encoder\ Value = Raw\ Encoder\ Value - Calibrated\ Lower\ Bound$$

Equation 6 – Rollover-to-Zero Equation for Shifting Encoder Values when Raw Encoder Value Exceeds Lower Bound

$$Shifted\ Encoder\ Value = Raw\ Encoder\ Value + (1023 - Calibrated\ Lower\ Bound) - 1023$$

Equation 7 – Rollover-to-Zero Equation for Shifting Encoder Values when Lower Bound Exceeds Raw Encoder Value

$$Shifted\ Encoder\ Value = Raw\ Encoder\ Value + (1023 - Calibrated\ Lower\ Bound)$$

Once again, in the normal case, the calculation to shift the values is fairly simple. Taking the difference between the raw encoder value read and the calibrated lower bound will result in the direct offset from the lower bound that the raw encoder value represents, for example a raw value of 600 and a lower bound of 400 would provide an offset of 200 from the lower bound, which is equivalent to a value of 200 on a scale of values that starts at 0.

This calculation becomes more complicated, however, if there is a rollover-to-zero during calibration. In this case, it is possible that the raw encoder value read at any given time can be either lesser or greater than the value of the lower bound (meaning that the raw value is either before or after the rollover-to-zero in the range of values). This being the case, a separate equation must be implemented for each case. In any case, the resulting value from the equation used is a value that represents the absolute offset from the lower bound that the raw encoder value represents, such that it parallels the offset from 0 in a zero-based range of values.

The fourth and final step is the actual modification of the raw value provided by an encoder in order to represent a value from a limited calibrated range as a value in a range of 0-1023. This step is the culmination of the three previous steps and is performed by simply utilizing the size of the calibrated encoder range that was determined in step two and the shifted encoder value determined in step three in order to take a raw encoder value and convert it to its corresponding value using a proportion of the size of the calibrated encoder range to the size of the overall range of the encoder. The equation used to perform this calculation is shown below.

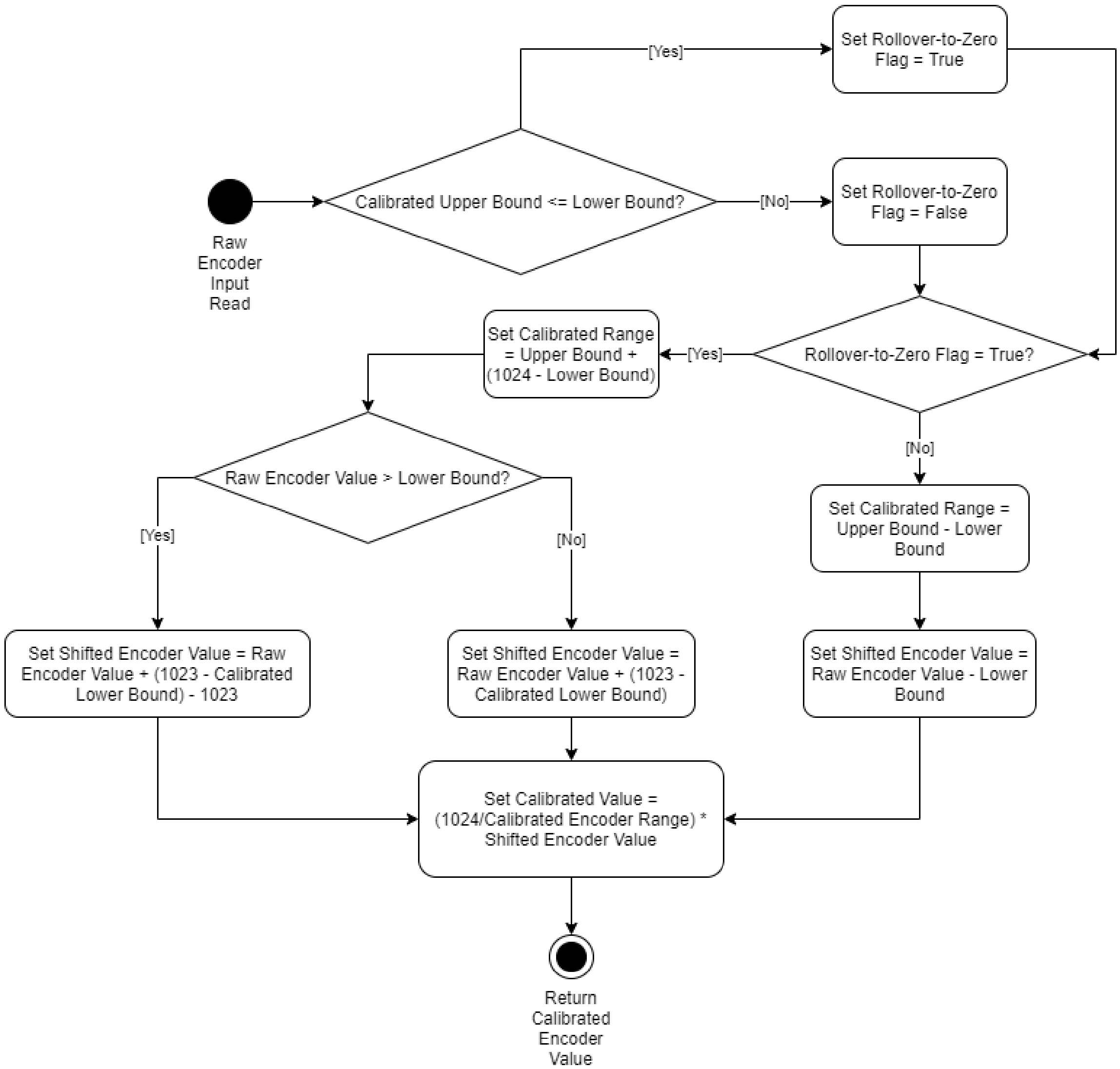
Equation 8 – Calculation of Final Calibrated Encoder Value

$$Calibrated\ Value =$$

$$\left(\frac{1024}{Calibrated\ Size\ of\ Encoder\ Range} * Shifted\ Encoder\ Value\right)$$

As mentioned previously, this value is proportionally equivalent to the raw encoder's value along the calibrated range of the encoder. For example, given a calibrated range between 400 and 800 and a raw encoder value of 600, the final calibrated value would result in a final value of 512, which is exactly halfway along the range of 0-1023. This entire process is what allows the collective to have a limited range of movement but have the microcontroller still report a value to the Bluetooth module in the desired range of 0-1023 in order to simulate an axis on a gamepad to the host computer. The entirety of the previously described process is visualized in the figure below.

Figure 21 – Low-Level Calibration Logic Diagram



5.2.5.2.2 Calibrated Encoder Value Lookup Table

In the initial phases of software design, it was intended that the previously described calibration-based modification of raw encoder values was to occur upon taking in raw values from the encoders and then the modified encoder values would be sent over Bluetooth as gamepad input. However, it was later determined that this approach would be incredibly inefficient from a performance standpoint, as the somewhat complex modification of raw encoder values would occur many times a second, as often as input is sent over Bluetooth. In order to remedy this, the use of a lookup table was decided upon. Rather than modifying raw encoder values as they come in, instead it is possible to frontload the computation by performing the calibration-based value modification for all possible raw values immediately upon calibrating the encoders. This is facilitated through the use of 1024 value arrays in which each index of the arrays represents a raw encoder value and each value in the arrays represents a post-calibration modified encoder value. Initially, the indices of the arrays and the value at each index will correspond directly, representing uncalibrated encoder values. Once the encoders are calibrated, however, the corresponding array for each encoder will be iterated through, modifying each value using the calibration-based value modification logic previously described. After this process is complete, converting raw encoder values to their final modified value is as simple as utilizing the value in the encoder array that corresponds to the index matching the raw encoder value taken in.

5.2.6 Collective Head Input

Although not quite as complex as calibrating and processing angular input from an encoder, the various forms of input on each of the collective heads require some specific software design in order to properly translate from various physical forms of input including buttons, switches, and hats into a generic form of digital output representing a button state of 1 or 0, high or low. This need arises from the fact that gamepad input as it is sent to the computer exists in only two forms: axis input and button input. Axis inputs have a variable range from 0 to 1023 as was explained when discussing encoder input processing, however button inputs are a simple 1 or 0 representing if a button is pressed or not. This simplicity is convenient for sending information from the microcontroller to a host computer, however it means that some physical inputs, namely toggle switches and hats, require special software implementation to map to gamepad inputs that accurately represent the state of the physical switch in the game.

5.2.6.1 Toggle Switch Logic

The process of converting the input from a physical toggle switch into multiple button inputs that can be recognized by a host computer requires a few special considerations as opposed to momentary inputs like buttons. The first issue that must be resolved is the fact that a toggle switch, when switched to a certain position, physically remains in that position and continues to send input to the microcontroller that the switch is in that state. For example, in a two-position toggle switch, the switch will constantly read high on one pin and low on the other. This is a problem because, from the perspective of gamepad input, toggle switches inside of flight simulation software are toggled by pressing a button for the corresponding state of the switch a single time and releasing it in order to move the switch into the correct position inside of the simulation.

In order to solve this issue, the microcontroller’s software stores information regarding the previous state of each switch. By storing the previous state of a switch, it is possible to compare the previous state of the switch to the current state of the switch to determine if, in the time between the previous reading of input and the current reading, the switch has changed position. Using this information, the program determines whether or not to send a new Bluetooth HID report to the host computer containing the state of the switch. Checking the state of the switch in this manner thus results in a single button input being sent to the computer representing any specific position on a switch. Whenever the switch changes position, the new position is sent as a single button input to the computer and the status of the switch is updated accordingly.

The second problem that was considered in the software design of the toggle switches is specific to three-position toggle switches. These switches, although three position, are only connected to the microcontroller via two pins, each of which are pulled down to ground by pull-down resistors. The middle pin on a three-position switch is connected to a constant 5V line that is connected to either side of the switch if the switch is in the up or down position. This particular setup means that a three-position switch actually only outputs two values to the microcontroller despite having three states. This issue is easily overcome through the use of Boolean logic to determine the actual state of the switch at any given time. If either of the pins coming from the switch are high, then the switch is in the opposite position (switching to the up position sets the pin at the bottom of the switch to high). If neither of the pins coming from the switch are high, then it must be the case that the switch is in the middle position. Utilizing this simple logic, the software on the microcontroller takes two inputs from the switch and map them to three button outputs representing the up, middle, or down position which is otherwise sent utilizing the state-maintaining logic described previously.

5.2.7 Bluetooth Communication

One of the largest systems that must be handled via software is the matter of Bluetooth communication between the Atmega2560 microcontroller and the host computer via the BM70 Bluetooth chip. Although it is the BM70 chip itself that will be sending data over Bluetooth to the computer, the BM70 must be controlled via serial commands sent over UART from the microcontroller. This includes both the process of managing the pairing process between the BM70 and the computer as well as the process of sending input data and receiving response data to and from the BM70.

5.2.7.1 BM70 Command/Response Format

Communication between the microcontroller and the Bluetooth module can be either in the form of a command from the microcontroller to the module or a response from the module to the microcontroller. Both commands and responses are formatted in such a way that they consist of fields of a varying number of bytes represented by hexadecimal values. These values always start with a start byte, 0xAA in hexadecimal, followed by a two-byte length field, a one-byte opcode field indicating the command or response type, a variable amount of parameter bytes, and a single checksum byte. This general command/response format can be seen in the figure below.

Figure 22 – BM70 General Command / Response Format

	HEAD		MID	DATA	CRC
	Start	**Length**	**OP Code**	**Parameter**	**Checksum**
Byte No	0	1 - 2	3	4 - xx	Length + 3
Size (Byte)	1	2	1	0	1
Value	0xAA	1	Command/ Event	Command/ Event parameter	Checksum
	SYNC WORD	Checksum to be calculated			
			TARGET LENGTH		

Note. Reprinted with permission from *BM70/71 Bluetooth Low Energy Module User's Guide* (p. 66) by Microchip Technology Inc.

This format is extremely important, as the software on the microcontroller must both send over UART a string of values in the proper format to indicate a command to the BM70 as well as be able to properly parse incoming responses in order to determine the state of the BM70 and make decisions as to how to proceed.

In order to send commands in this format the program stores multiple arrays of bytes that are pre-populated with the correct values to represent certain commands, most notably the command to begin advertising to pair the BM70 to a device and the command to send a GATT characteristic value containing gamepad data, such that they can be altered to contain varying parameters and sent to the BM70.

5.2.7.2 Bluetooth Pairing

In order to send Bluetooth data from the BM70 to a device, in this case a computer, the BM70 must pair to the device. This is done by sending the BM70 a command to enter advertising mode from the microcontroller. From a software perspective, this is handled by the implementation of a pairing mode, effectively a waiting loop, that will be entered based on user input on the base of the collective. Once the microcontroller enters pairing mode it performs a couple of operations. Firstly, the pairing command sends the contents of the pre-defined byte array containing the command for the BM70 to enter advertising mode over UART, which is a completely static command that is always known. Then, the program will receive the response from the BM70 indicating the status of the chip, such as pairing complete or command failure, and exit pairing mode with an indication to the user on the base of the collective if pairing was successful. Once this has been completed, the BM70 is ready to send GATT characteristic values to the computer to be interpreted as gamepad input.

5.2.7.3 Gamepad Input Commands

As opposed to simply sending a command to pair the BM70 to a device, sending commands to the BM70 containing gamepad input data is significantly more complicated. This is due to the fact that only part of the pre-defined command format for the command to send gamepad input (send a GATT characteristic value) is static, as the parameter data and the checksum must both be set in order to properly send a command to the BM70 to transmit gamepad input. Firstly, the input for the head is read in, and the corresponding bits in the 2 byte parameter representing buttons are set and that value is stored. Second, the values of the lever encoder and throttle encoder are read in, and their values are translated into their respective 2 byte parameters and stored. Third, the checksum is calculated by summing the hexadecimal values of every other field excluding the start field and subtracting the value from 0x100. Once all of these steps are completed, the program then modifies the byte array containing the command to send a GATT characteristic value to contain the previously described parameters and checksum. This command is then sent over UART to the BM70 which sends the data to the computer.

5.2.8 Collective Base User Interface

In order for the user to interact with the collective unit outside of manipulating inputs on the collective lever and head to be translated to gamepad input, a simple user interface is present on the collective base. This user interface consists of an LCD and three buttons. The purpose of the LCD is both to display current information about the state of the collective to the user (such as current encoder values) as well as to serve as a sort of menu display through which the user can interface with the collective unit as a whole in order to perform certain operations. These operations all stem from the idea of different "modes" or states that the collective unit can be in at any given time, as has been explained in both the sections regarding Bluetooth pairing and encoder calibration. The purpose of the outward-facing user interface is for the user to be able to switch between these modes as necessary to perform essential functions such as pairing the collective unit via Bluetooth to a computer or performing manual calibration of the encoders.

5.2.8.1 User Interface Functionality

In order to implement multiple different functions using a small number of inputs, a menu system was implemented in which the LCD will present information to the user and then the buttons on the collective base can be used to navigate between different menus or modes. In order for this functionality to be implemented, each button must serve multiple purposes depending on the state of the collective unit. One button is dedicated to the purposes of selecting a mode/initiating a command to the microcontroller, whether this be to start calibration, start pairing, set the bound for the current calibration step, or other commands. This button essentially serves as a sort of a confirm or selection button that initiates functionality. The other two buttons serve as navigation controls between the different menu options, either backward or forward. This simple implementation allows for the use of only three buttons to accomplish a wide range of menu options and functions.

5.3 Mechanical Design

The mechanical design aspect of this project acts to meet our goals of producing an end product that is both functional and as visually accurate to a real-world counterpart as possible. With this goal in mind, all components, including COTS parts, were modeled inside computer-aided design (CAD) software in order to expedite design and ensure correct tolerances on all custom manufactured parts. The final assembly of our design can be seen below in Figure 23, while the exploded view showing the breakdown of all components individually can be seen in Figure 24 below.

Figure 23 – Complete Collective Assembly

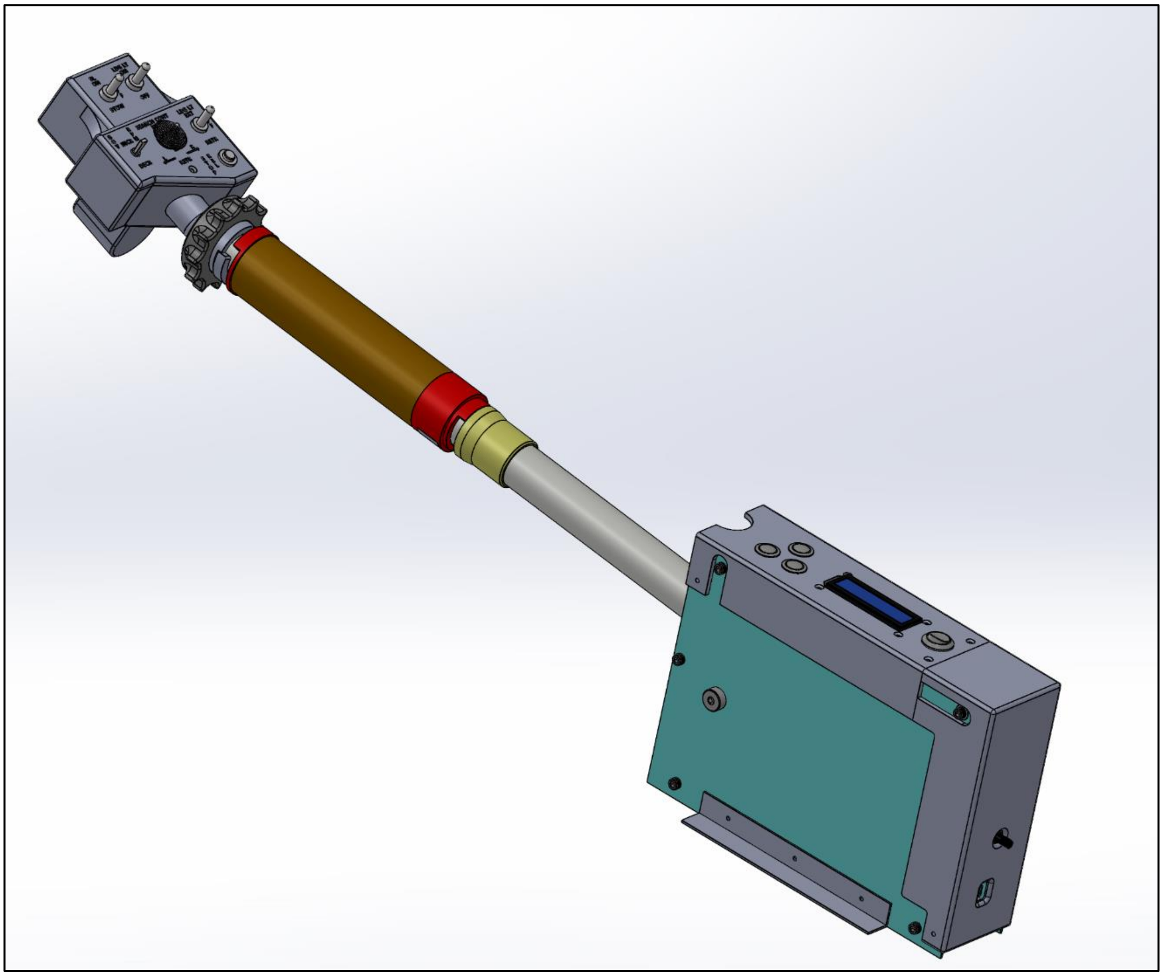
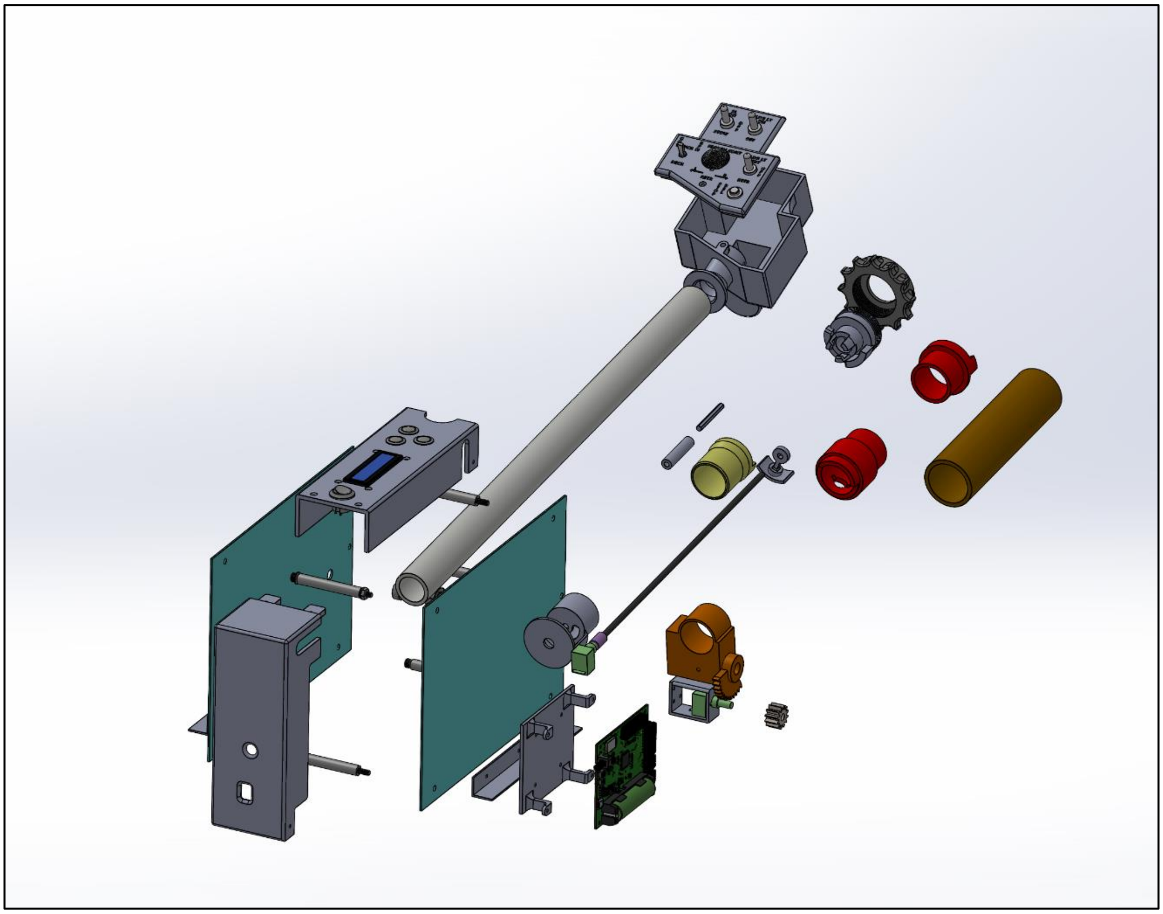


Figure 24 – Complete Collective Assembly, Exploded View



5.3.1 Collective Base and Lever

The elements we considered when designing the physical implementation of the collective base and lever arm were the elements that directly related to meeting or exceeding our stated engineering requirement specifications. Due to the fact that, unlike the collective heads, the base and lever arm are inherently permanently attached after assembly, the design was done in tandem to ensure proper fitment and operation.

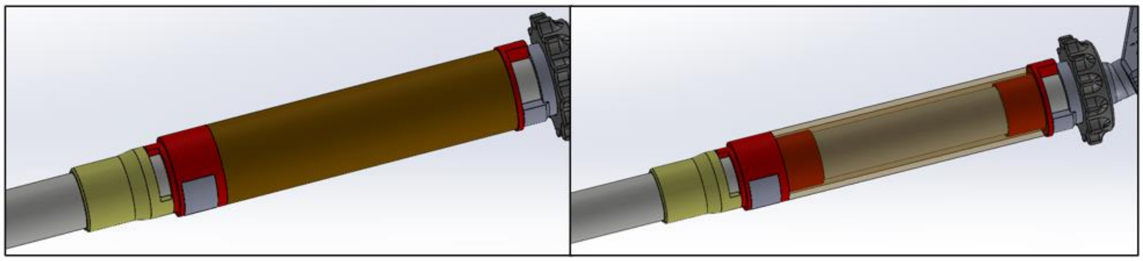
5.3.1.1 Throttle Assembly

The first design element that was decided upon was the main lever arm and throttle control dimensions. This was an important starting location as the throttle control was identified as the driving factor in the lever design due to the fact that it was the human interface portion. This allowed us to immediately have an upper bound reference for the interior lever arm, as well as start the design process for how the throttle mechanism would connect to the throttle encoder. In the testing of different diameters and through the reference material we had for existing real-world collectives, we determined that a nominal outside diameter of approximately two inches would be the best fit for our design. After determining the desired dimension, we researched and determined that the use of 1.5 inch

PVC Schedule 40 pipe, with an outside diameter of 1.9 inches, would act as a close facsimile to real collective throttle controls and be an economic choice as it would require no custom fabrication. Based on this selection, the interior lever arm, used to connect the base unit to the collective heads, was chosen to be a 1-inch PVC schedule 40 pipe for the same reasons as the throttle pipe selection.

After determining the lever and throttle control dimensions, the connection between the pipes and the connection to the throttle encoder was considered.

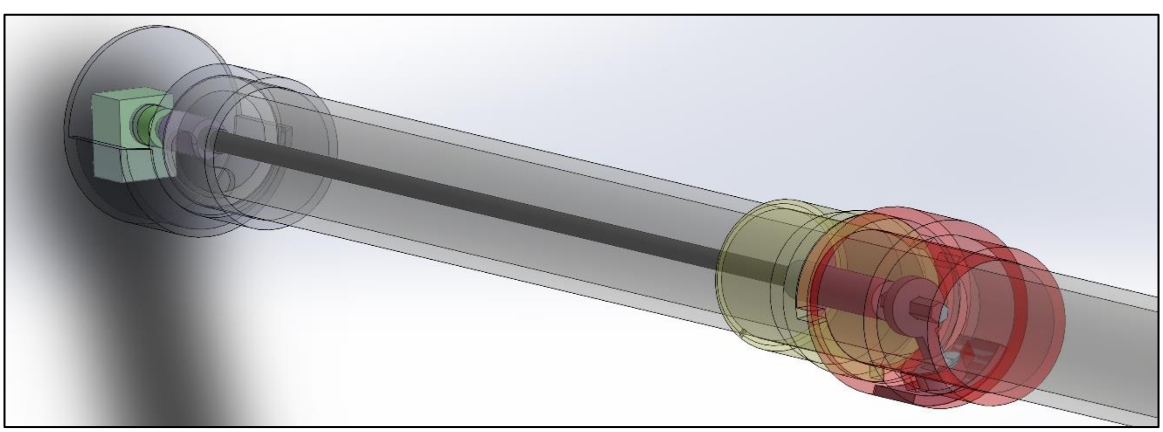
Figure 25 – Collective Throttle Control



As shown in Figure 25 above, the connection of the main lever arm (shown in light-grey) and the throttle control (shown in brown) is facilitated by two bearings (shown in red) that are affixed to the throttle control and freely move around the central axis of the lever arm. Both bearings are designed to include blocking tabs that interface with permanently affixed collars on the main lever shaft (shown in yellow) to limit the throttles total range of motion to 140 degrees. This configuration of the throttle control allows the user to freely rotate the control with the necessary friction to maintain a set position provided by the friction between the bearing surfaces and the interior main lever arm. When assembling, to ensure proper alignment of the bearing surfaces, the tabs of each are simply lined up inside the inner diameter of the throttle control and glued in place.

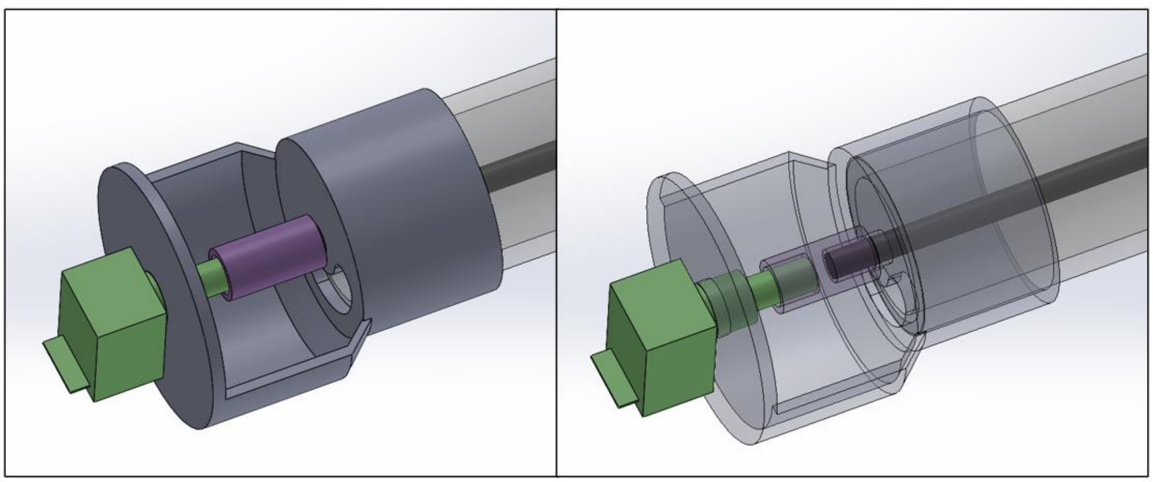
Once the fixture between the throttle control and the main lever shaft was established, the methodology for connecting an exterior rotating shaft to the throttle encoder was contemplated. This design posed a particular challenge as all possible options explored contained some inherent disadvantage related to either physical stability or accuracy.

Figure 26 – Collective Throttle Shaft Assembly



We eventually concluded that a shaft assembly as shown in Figure 26 above, which is constructed of a statically mounted plate (shown in grey) on the throttle control bearing (shown in red) that rotates around the central axis of the lever arm, was the most effective way to translate movement from the throttle control into the lever arm. The decision to mount a separate plate directly into the bearing of the throttle control was made for a few reasons. First, as alluded to previously, the downside of this design is the need to remove a sizable portion of the main lever's pipe in order to allow the arm of the outer-plate to reach the inner-shaft. This problem is mitigated by this design by allowing the slot cutout on the main lever's pipe to be directly centered over the throttle bearing, thus allowing the bearing to act as a coupler between the two intact sections of the pipe. Secondly, this design allows for nearly perfect translation of the throttle to the shaft, thus allowing for a far higher accuracy when compared to designs considered involving the reading of position in a non-mechanical way, such as hall-effect sensors through the pipe. Finally, the decision to manufacture the plate with an attached arm and the bottom bearing separately was made in order to increase the ease of assembly and decrease the amount of removed material from the lever arm that would be required.

Figure 27 – Throttle Encoder Housing



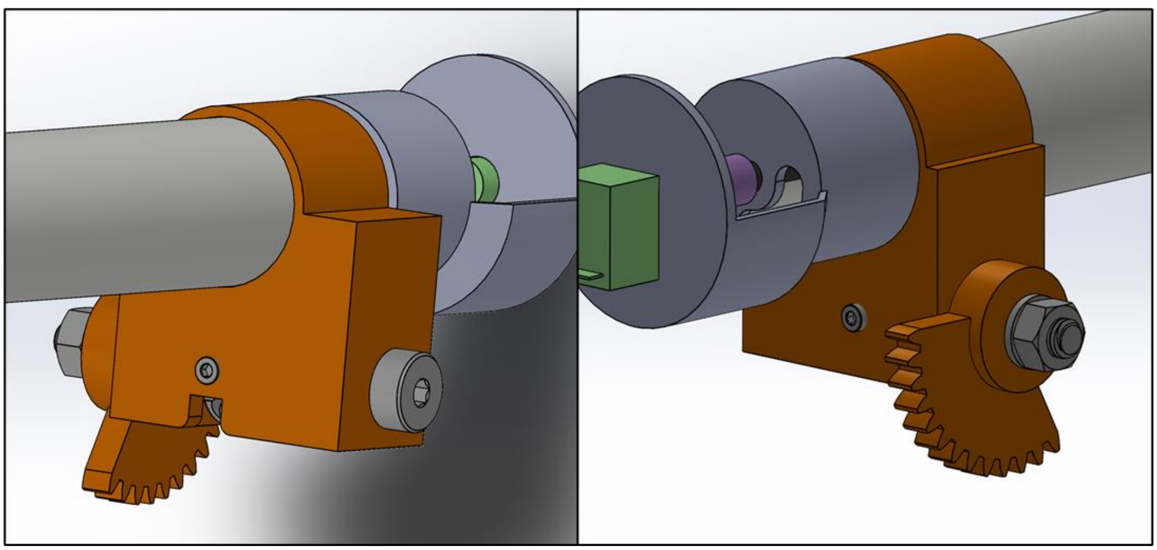
The physical mating between the central shaft and the encoder, shown in Figure 27 above, is achieved by an end cap (shown in grey) and a simple coupler (shown in purple). The end cap acts as both a mounting location for the encoder and as an alignment bearing for the shaft connected to the throttle control. The bottom base plate of the cap has a sufficiently large surface area and wall thickness to allow a panel mounting style attachment of the encoder, which ensures near-perfect central axial alignment. The shaft of the throttle is aligned on this same axis via a bushing built into the endcap and the two are connected with a simple coupler that mates to the end of the shaft and encoder. The end cap is designed to provide maximum rigidity while having a large enough opening in the side wall to properly insert a wrench to facilitate the panel mounting of the encoder. The end cap also maintains a small radial cutout that exposes the inside of the lever arm pipe. This cutout allows for the installation of the wire that will run from the PCB, housed inside the base unit, to the head of the collective. The combination of these elements and the other elements of the complete throttle assembly were directly designed in an effort to meet core features of our project that were previously enumerated in the project functionality section. The creation of the complete throttle assembly acts as large portions of the feature dictating that the current angular position of the throttle can be measured and the feature that dictates the unit will be able to convert the angular value to a joystick axis. The accuracy provided by the shaft-based design ensures the first feature is possible, while the mating between the shaft and the encoder ensures the latter feature is possible.

5.3.1.2 Base Assembly

The base of the collective is designated as the location for storing the bulk of the electrical components, including the PCB, LCD information display, and the buttons used to directly interact with the unit. It also contains physical fixtures that facilitate the movement of the collective lever arm, the measuring of the collective angle, and the variable friction system for the collective.

The first consideration made when designing the base of the unit was the methodology and physical implementation of the collective lever rotating around a set axis with the ability to translate movement to an encoder, while also being confined to a preset total range of motion of 45 degrees.

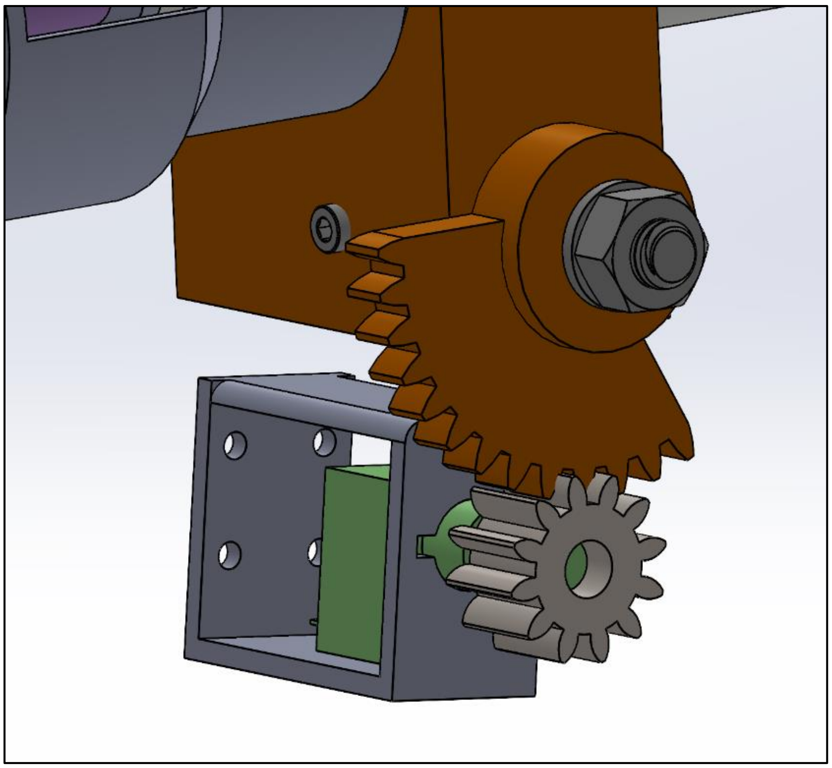
Figure 28 – Collective Lever Collar-Bearing Combination



The final design for allowing the collective to rotate about an axis is shown above in Figure 28. The main component attaching the collective lever arm to the axis is a collar and bearing combination (shown in orange) that is slipped over and fixed in place over the lever arm, while being allowed to freely rotate around a shoulder bolt with a diameter of 10mm. One design attribute that had to be taken into account when creating this part was the offset axis from the center of the lever arm. This would ultimately add a small amount of complexity when considering the mounting of these elements between sidewalls, however this choice was made as a centrally mounted axis would be infeasible due to the central shaft connecting the throttle and the need to run wires from the base of the lever to the head of the collective.

Similar to the throttle assembly discussed previously, the collective lever's angle must also be translated into an encoder in order to accurately determine its position. However, unlike the throttle control that has 140 degrees of rotational movement, the collective lever's full range of motion is only 45 degrees. If a 1:1 ratio was used to measure the angular position of the lever, like the throttle, a total of 128 out of the possible 1024 encoder steps would be used. This would result in the software scaling the final output value by approximately 8:1, thus resulting in a significant loss of granularity and fluidity of movement.

Figure 29 – Collective Lever Geartrain

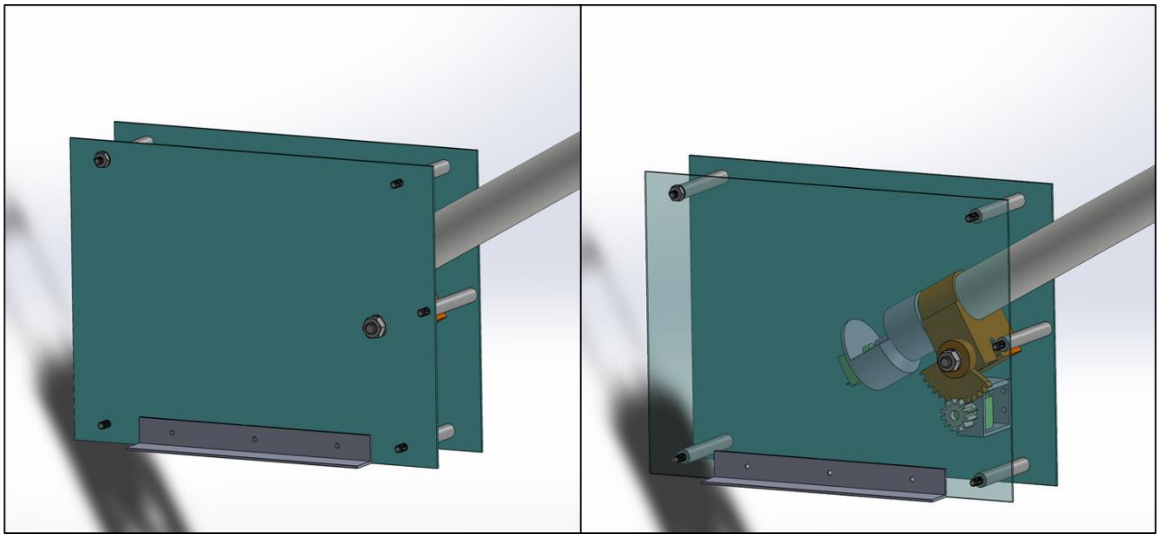


In an effort to improve resolution of the output from the collective lever arm a simple gear train was decided on, as shown in Figure 29 above. A gear ratio of 3:1 was decided upon as we believe that this gear ratio acts as a good balance between gear size and increase in resolution. With this overdrive the total effective usable encoder steps increases to approximately 384. The mounting solution for the encoder itself is of similar design to the throttle encoder mounting assembly in that it allows for the use panel mounting hardware. This ensures that the encoder is nearly perfectly centered when assembling. The collar and bearing combination combined with the gear train facilitates the core feature of the unit being able to measure the current angular position of the collective lever.

After determining the methodology for measuring the required axes, the physical design of maintaining the set position for both the throttle and collective were considered. We determined through preliminary testing that the friction provided by the set of bushings on the throttle control were sufficient for maintaining the set position of the throttle pipe.

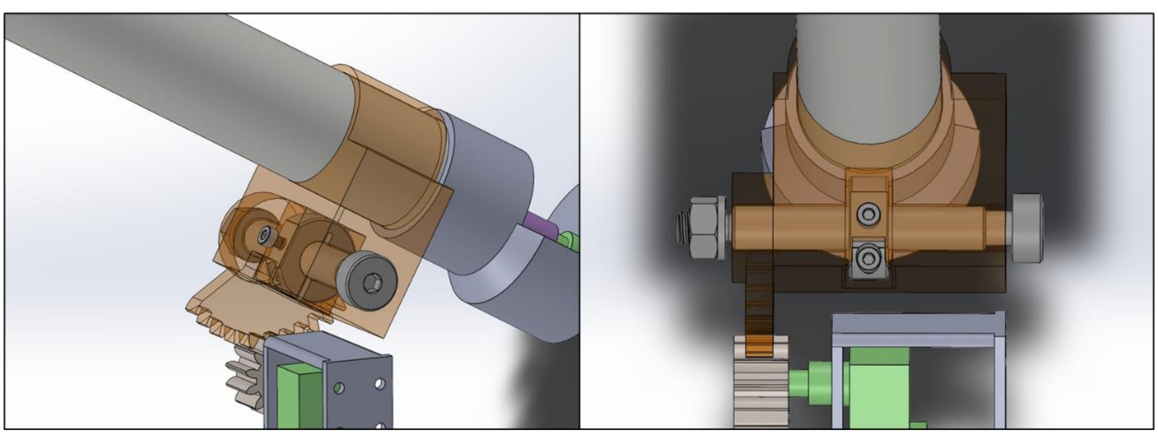
However, due to the fact that the collective lever was actively working against gravity, a friction system at the base of the collective unit was required.

Figure 30 – Throttle Base Unit Structure



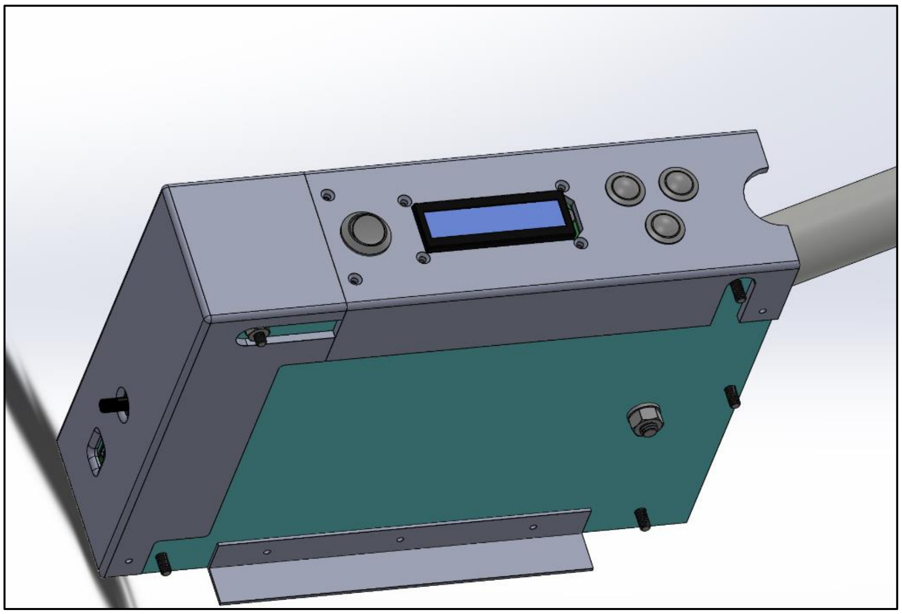
In order to achieve the goal of adding friction to the lever arm assembly, the side walls of the base assembly were added and designed to add pressure to the lever arms axis of rotation, as seen in Figure 30 above. The total allowable strain put on the axis of rotation is controlled by the spacers separating the plates, ensuring that the lever arm is still able to move while still having a small degree of friction adjustment based on the amount of force generated by the shoulder bolt and nut combination that acts as the axis of rotation. The addition of these side walls also acts as a reference point for all other components in the total base unit assembly, such as the collective encoder offset seen in this figure. While this method does support the core feature of being able to measure the current angular position by not allowing it to change when the user is not holding the lever, it does not meet the stretch goal of having an adjustable friction mechanism as the adjustment is too coarse and difficult.

Figure 31 – Collective Lever Friction Adjustment Mechanism



In an effort to increase the granularity of friction and ease of adjustment, it was decided to add a shaft collar with a variable friction screw to the design inside the collar and bearing combination holding the collective lever, as shown in Figure 31 above. The top-and-bottom mount shaft collar is installed with its central axis in line with the axis of rotation allowing the user to adjust the friction by simply tightening or loosening a single socket head screw. With the addition of this shaft collar, the initial friction provided by the plates will be set as the minimum allowable friction, while any additional desired friction coming from the tightening of the collar around the central shaft by the user. This addition to the design fully satisfies our stretch goal of allowing the user to select the amount of resistance felt when operating the collective lever and does so in a way that minimizes the amount of metal-on-plastic contact, thus resulting in a more resilient end product.

Figure 32 – Collective User Interface



The final consideration made for the base unit was the method of mounting the user interface elements to the end product. These elements include an LCD panel to display information to the user and three buttons to accept user input. In an effort to make the interface intuitive and easy to use while seated, it was decided to mount these elements on top of the base unit facing the user of the collective, as seen in Figure 32 above. The holes for panel mounting all user interface elements were designed directly into the part to ensure proper alignment when assembling the final product. Similarly, a rear panel was designed which exposes a USB-C port for charging and a potentiometer knob to adjust the contrast of the LCD display.

5.3.2 Collective Head

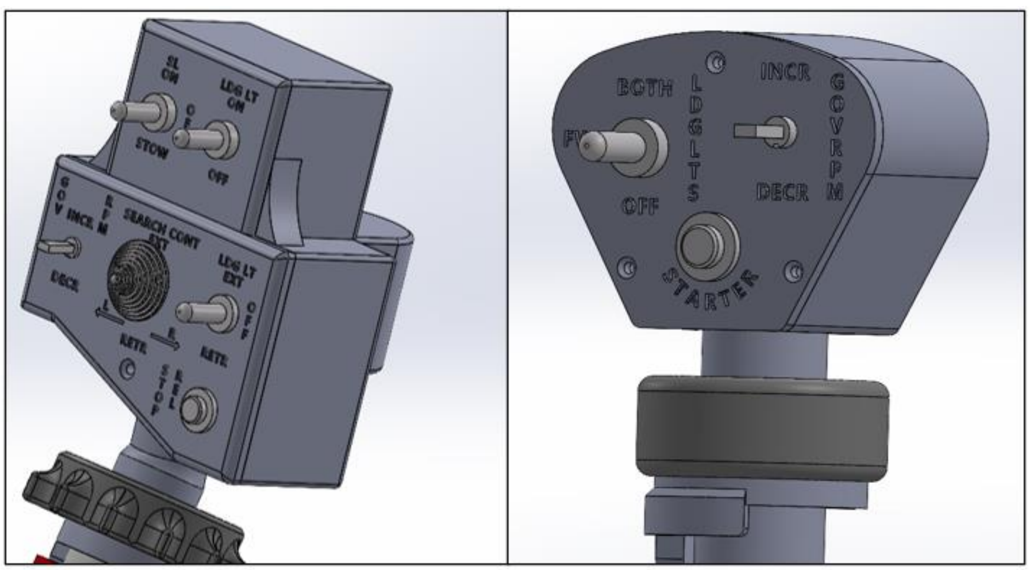
The head of the collective was quickly identified by us as one the major components that would greatly add or subtract from the realism based on the quality of execution and likeness to the real-world counterpart. It is for this reason that we decided to select real-world aircraft to model our collective heads from.

5.3.2.1 Collective Head Modeling

The two helicopters initially selected by us to simulate are the Bell UH-1, commonly know as the "Huey", and the single engine Bell 206 family of aircraft. Both of these real-world collectives are great candidates for our design, as they both maintain a similar collective lever shape and function. This is due to the fact that they are both Bell Helicopter designs and both maintain only a single engine, thus not requiring two separate throttles on the collective lever. Independently, the Huey collective head is a great choice because it is an iconic design that is implemented in many of the largest flight simulation software products. The head also contains a large number of bindable switches, a hat switch, and a button, which allow it to be versatile even across other aircraft. Conversely, the Bell 206 collective head has a much similar design allowing for the design process and initial prototyping to happen much faster and allowing for new users of the end product an option that is not overwhelming to use.

The dimensions for both the Bell 206 and Huey collective heads were taken from publicly available sources or derived from images and other known dimensions on the collective.

Figure 33 – UH-1 Huey Head and Bell 206 Head

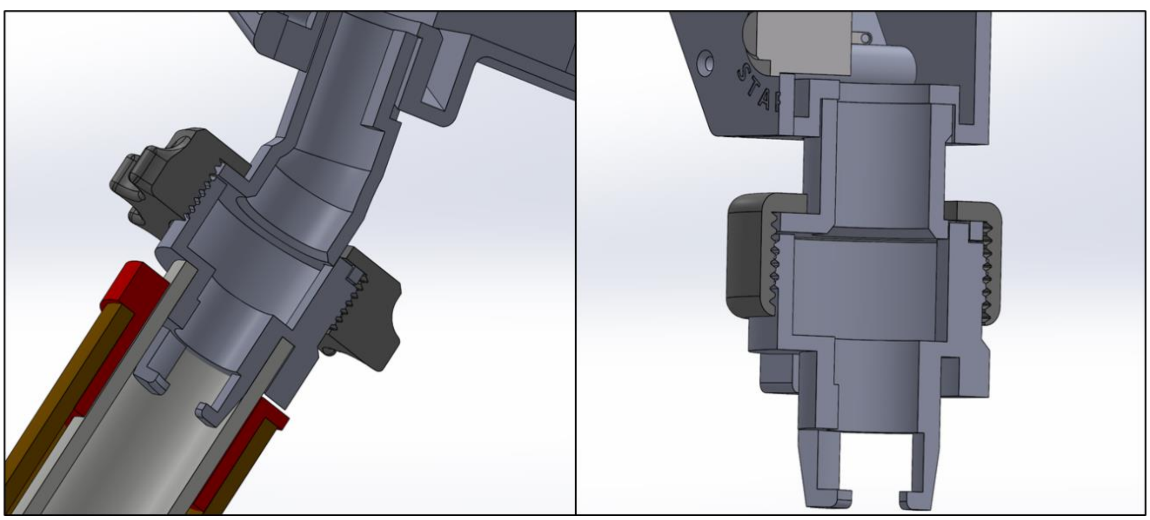


As shown in Figure 33 above, the collective heads for both the Huey and Bell 206 were modeled to match the form of their respective real-world collectives when assembled, however the actual structural assembly is different from their counterparts. While the real Bell 206 collective head assembly relies on a stamped and printed face plate that covers a metal assembly holding the electronics, our design incorporates the text directly embossed inside a 3D printed face plate that has the panel mount holes designed in place. Similarly, the Huey also maintains separate printed face plates to denote button and switch functionality, but it also has a more complex housing due to the fact that the top two switches are in a separate compartment that must be screwed onto the rest of the head. In our design, due to the choice of using 3D printing, we are able to manufacture a single compartment housing that allows us to maximize the use of space inside the head, while also maintaining a nearly identical assembled shape.

5.3.2.2 Collective Head Attachment

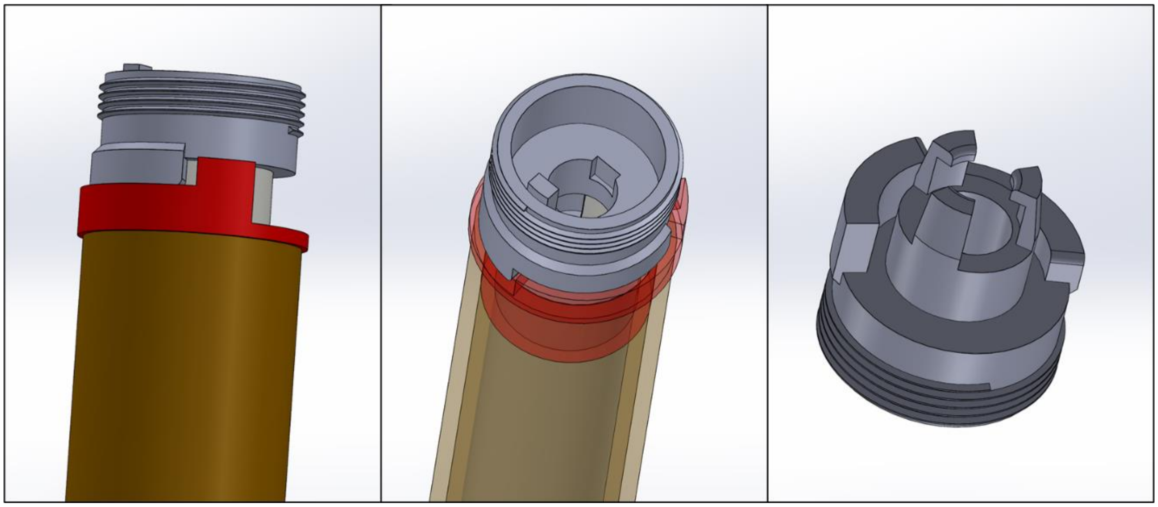
One of the more complex considerations made in our design was the method of attaching and swapping collective heads on a single collective lever, throttle, and base. This is an important design element as it fulfills both the core feature of having a head be attached to the collective and the advanced feature of having two swappable controls heads to mimic two different helicopters. The difficult in the design is the fact that the geometry of both the Huey and Bell 206 heads are significantly different, as can be seen in Figure 33. Due to the fact that many collectives have a functional mechanism in between the head and throttle control, such as the friction adjustment nut on the Huey or the idle stop on the Bell 206, a compromise was made between realism and end product usability by not including these features to allow for hardware that facilitates the swapping of heads.

Figure 34 – Collective Head Quick Detach Mechanism



As seen in Figure 34 the friction adjustment nut and idle stop were both eliminated in order to allow for captive nuts on both collective heads. These captive nuts act to tension the neck of both collective heads to the top of the collective lever arm, however both nuts are modeled after the friction adjustment nut and idle stop to ensure that the end product still maintains the same appearance as the real-world collective. The captive nuts are manufactured separately for both collective heads and assembled around the neck of their respective heads, thus making a single unit that can be attached and detached from the end of the collective lever. In order to achieve cross-compatibility, the captive nuts' inner-diameter are designed with a consistent thread size of 1.75-12, which mates to the lever arm end cap shown below in Figure 35.

Figure 35 – Collective Lever Threaded End Cap



The lever end cap, shown in Figure 35 above, acts, as mentioned previously, to mount the swappable collective heads physically to the collective lever arm. However, it also acts as both a mounting point for the female 17-pin Molex connector and as an additional limiter for the throttle, similar to the lower collar mechanism discussed previously. The ability to have a statically mounted Molex connector inside the end of the collective lever was an important design decision as we believe it is the most simplistic way to swap collective heads as an end user and allows us to store any extra wire from the connection inside the neck of the collective head itself. The small tab shown in Figure 35 above slots into a notch cut into the collective head's neck in order to ensure that the collective head is installed facing the user.

6.0 Prototyping and Testing

Now that the major aspects of electrical, mechanical, and software design have been considered, the next step is to begin prototyping the collective itself. This prototyping step will be largely broken down into two phases: functional prototyping and PCB construction. The following sections will explore the entire process of prototyping for this project throughout both phases.

6.1 Bluetooth Testing

One of the essential functions of the helicopter collective unit is the ability to connect wirelessly via Bluetooth Low Energy to a remote device and act as a gamepad. In order for this to be achieved, the BM70 Bluetooth Module must be configured and tested appropriately to ensure that the collective can effectively connect to a remote device, send data, and be recognized as a Human Interface Device so that it may be utilized as a gamepad. The testing of this functionality will be achieved via a development board offered by Microchip known as the BM71-XPRO board. The main steps of testing in regards to Bluetooth are testing the ability to pair to a remote device, appear as a gamepad, report battery level to the remote device, and appear as a gamepad and send relevant gamepad input data to remote device.

6.1.1 BM71-XPRO Development Board

The BM71-XPRO Development Board is a convenient breakout board which allows the developer to focus exclusively on testing their Bluetooth configuration and functionality without worrying about the electrical requirements of powering or communicating with the BM71. The BM71-XPRO provides a micro-USB port which can be used to both power the BM71 as well as act as appear as a COM port on a connected PC in order to directly communicate with the BM71 from a PC-based serial application. This is an extremely valuable feature as it integrates directly with the Microchip BLEDK3 Manual Test Host Emulation Tool mentioned earlier. This means that the BM71-XPRO can be connected directly to a PC via USB and have new firmware flashed, have a new configuration written, and also have the PC appear to the BM71 in the same manner as a host MCU and send test commands.

The BM71-XPRO is also an invaluable testing tool as it exposes the pins of the BM71 on standard pin headers which means that the BM71 can be hooked up to a breadboard for testing. This allows for nearly complete system testing of the helicopter collective on a breadboard via the Arduino Mega 2560 Rev 3 and the BM71-XPRO boards. This capability is crucial to virtually all aspects of testing, including not only Bluetooth Configuration testing but also electrical and software testing. While the BM70 was utilized in our end design, as mentioned previously, it is nearly identical in function to the BM71 which means that the BM71-XPRO board was able to be used to test function regardless of the its usage of the BM71.

6.1.2 Initial Bluetooth Testing

The most basic of Bluetooth Testing was conducted via the BM71-XPRO board connected directly to a host PC via USB. The BM71-XPRO was first set via the included DIP switches to run the BM71 in Configuration Mode, which allows the host PC to flash new firmware as well as write a new Bluetooth Configuration to the module. After setting the BM71-XPRO to Configuration Mode, the newest firmware version of the BM71 must be written in order for HID over GATT to function properly according to Microchip's BM71 HID Example documentation. After the new firmware has been uploaded, the Bluetooth Configuration mentioned previously involving the configuration of all GATT Services and related Bluetooth Low Energy options was written to the module via the IS187x BLEDK3 UI Configuration Tool. After this configuration had been written to the module, the BM71-XPRO was then switched back to Application (Run) Mode and reset in order to test the effectiveness of the aforementioned Bluetooth Configuration.

Once the BM71-XPRO had its configuration written and had been reverted to Application mode, the Manual Host Emulation Tool was used in order for the PC to simulate a host microcontroller sending commands to the BM71. Once a serial connection had been established with the BM71, a simple command called “Read Local Information” was sent from the Test tool and provided a response indicating information about the BM71 and served as a simple test to ensure that the connection has been established and the BM71 was ready to receive commands. After receiving a valid response from the “Read Local Information” command, the Local GATT Table was then requested from the BM71 which returned a list of the GATT Services and their associated configuration values. These were then verified against the values in the UI Configuration Tool to ensure that the new configuration was written to the module correctly.

After verifying that the BM71 was connected and the new configuration had been written, the BM71 was instructed to begin searching for a device to pair with. This was achieved by sending a “Set Advertising Enable Command” in the Manual Test Tool along with a “Enter Standby Mode” parameter. This instructs the BM71 to exit idle mode and begin advertising that it is pairable to nearby Bluetooth Low Energy capable devices. A computer with Bluetooth Low Energy capability then connected to the BM71 and verified that the BM71 appears as a gamepad and connects properly. After connecting, the Manual Test Host Emulation tool then reported that the status of the BM71 had changed to “BLE Connected”.

Once the BM71 and the host PC were connected, the gamepad functionality of the BM71 as well as the ability to send data in general were tested. First, the remote Windows PC was used to open the built-in Game Controllers configuration application and confirm that a new gamepad device was detected. If a device appears in this menu, it verifies that the BM71 is properly reporting itself as a gamepad and Windows is able to detect that it is a Human Interface Device. Next, by referring to the format of the HID Report as defined by the HID Report Descriptor, which was previously configured, the Manual Test Host Emulation Tool was used to send HID Report Data to the connected PC. This is done by using a “Send Characteristic Value” command in the Manual Test Tool with the characteristic to be updated being the Report characteristic and the value to be sent being a pre-formed test value. After sending this report, Windows 10’s built in Game Controllers application reflects the buttons pressed or axis movements which were sent by the BM71. These button presses and axis movements were reflected properly, which indicated that the BM71 had successfully connected to a remote device as a gamepad and sent gamepad inputs. Similarly, the “Send Characteristic Value” command was used to update the Battery Level characteristic with a value between 0 and 100. This value was reflected in the Windows’ Bluetooth Devices menu as the current battery percentage of the device.

6.1.3 Breadboard Bluetooth Testing

Previously, all testing was done via a USB connection the PC using the Manual Test tool. While this testing is valuable for validating the Bluetooth configuration and HID functionality, it is not representative of the conditions in which the BM70 will actually operate. In the helicopter collective unit itself, the BM70 is connected to the Atmega2560 host microcontroller and must receive programmatically constructed byte commands over UART which instruct the BM71 what to do. In order to test this functionality, the BM71-XPRO board exposes the BM71's relevant pins as standard 2.54mm pin headers which means that the BM71-XPRO can be disconnected from the PC and instead connected directly to the Arduino Mega 2560 Rev 3 development board. This setup most closely simulates the actual operating conditions of the BM70 as it will be implemented in the final design. This allowed for simple testing software to be written and loaded onto the Arduino which sent strings of bytes to the BM70 in order to test the same functionality as before (pairing, sending HID reports, sending battery level, etc.). However, this method also allowed the commands themselves to be tested, as rather than relying on the pre-formed commands and simple GUI of the Manual Test Host Emulation Tool, the commands were developed by interpreting the BM70's documentation and the UART responses from the BM70 to the Arduino were parsed and interpreted as well. This method of testing provided far more valuable information in regards to software implementation and the viability of the Bluetooth configuration in the final implementation.

6.2 Functional Prototyping

The term functional prototyping, in this context, simply refers to a form of preliminary prototyping in which essential functions of the collective unit as a whole are implemented on a small scale prior to PCB manufacturing in order to verify the core aspects of design that are planned for implementation when constructing the collective. These essential functions include reading in and calibrating encoder input, formatting and sending and receiving commands/responses to and from the Atmega2560 to the BM70, pairing the BM70 to a computer over Bluetooth, and correctly sending encoder input over Bluetooth such that it can be interpreted by the computer as an axis-based game input.

6.2.1 Functional Prototyping Methodology

Functional prototyping was conducted via the use of two development boards, the BM71-XPRO and the Arduino Mega 2560 Rev 3, connected to each other via the use of a breadboard. Essential functions that were tested were triggered via individual buttons on the breadboard. A compact software implementation that only implements the most critical aspects of software required to test essential functionality was used in order to simply and effectively prototype essential collective functions. The details of the implementation of each essential function during the functional prototyping phase will thus be described.

6.2.2 Encoder Input Prototyping

The absolute most essential function of the collective as a whole is the ability to take in and read angular input via absolute rotary encoders. Thus, it is only natural that this is the first and most important step of functional prototyping; If this functionality is not possible, the entire viability of the project comes into question. In order to implement this functionality in a very simplistic manner for prototyping, the EMS22A Encoder was directly connected via jumper wires to the Arduino Mega 2560 Rev 3. A function that implements the process of receiving direct input from the encoder, as described previously in this document, was implemented such that it manually requests a single value from the EMS22A. This function was then used in a larger function that takes in the raw input from the encoder and converts it to a calibrated value using previously described calibration logic. The actual process of calibration was performed via a single button on the breadboard. The calibrated encoder output was then sent over serial to a PC connected to the Arduino Mega 2560 Rev 3 and was displayed on a serial monitor. Values on the serial monitor were then analyzed as the encoder was rotated utilizing various values of calibration in order to verify that encoder input was being correctly taken in and modified to a corresponding value in the range of 0-1023.

6.2.3 Bluetooth Communication Prototyping

After input from the encoder has been successfully read in by the microcontroller, the next functionality to be implemented is Bluetooth communication. This was implemented via programming the microcontroller on the Arduino Mega 2560 Rev 3 to directly send commands to the BM71 on the BM71-XPRO development board. Bluetooth communication prototyping occurred in three steps: sending and receiving Bluetooth commands/responses to and from the BM71, pairing the BM71 to a computer, and sending calibrated encoder input as gamepad input.

The first step of Bluetooth communication prototyping, sending and receiving Bluetooth commands/responses to and from the BM71, was achieved via sending hexadecimal values that are predefined in a byte array to the BM71 over a serial connection between the Arduino board and the BM71-XPRO and receiving a serial response from the BM71. For the purposes of this initial prototyping, there is no processing of the response from the BM71, commands were simply sent, and responses received. The contents of both the commands sent and responses received were then displayed on the serial monitor of a connected PC.

The second step of Bluetooth communication prototyping, pairing the BM71 to a computer, is rather straightforward following the first step. Once commands were successfully being sent to the BM71 over a serial connection, a predefined command formed from information found in the BM71 User’s Guide was sent on a button press to indicate to the BM71 to begin advertising over Bluetooth. The success of this operation was then immediately apparent by searching for available Bluetooth devices on a separate computer and attempting to connect to the BM71. Success of this step was judged both by an indication of a successful connection by the connecting PC as well as by manual observation of the response code from the BM71 in order to ensure that the connection was securely established.

The third, and final, step of Bluetooth communication prototyping is to actually send the input from the encoder as a command to the computer to be recognized as a gamepad input. This was achieved by pre-configuring the BM71 via the manual tool described previously in this document and then sending commands to the BM71 in the format described by the configuration set. In order to do this, the implementation of the previous step in functional prototyping was adapted to send a different predefined command format that the computer will recognize as gamepad input. This byte array was then modified to contain the input value from the encoder and was then sent as a command to the BM71. The success of this step of functional prototyping was determined utilizing Window's built-in gamepad testing functionality, which not only shows axis-based input on a visual plane but also reports the value it receives. This value was then matched to the actual value being sent by the microcontroller to the BM71 to ensure that the proper input was being received at the computer itself.

6.2.4 Procedural Steps for Electrical Hardware Testing

After finalizing the design phase, it is important to organize an effective plan of action when transitioning to the implementation phase. This involves proper and efficient testing of individual components and subsystems, and correctly integrating and executing the full system. The following section includes a step-by-step protocol that was followed during implementation.

Protocol:

- Determine and set-up the power supply that corresponds to each main component, at the specified nominal operating input.
- Test the main components or subsystems individually (USB-C, Charging IC, Voltage Converters) and record their outputs.
- Charge and discharge the battery under expected loading conditions and observe its behavior. Verify battery integrity by confirming that it achieves max rated charge.
- Test each main sub-system under known conditions and compare acquired outputs to expected outputs. Test under desired load and determine if valid under expected results
- Combine all subsystems and set up under expected conditions. Measure each critical input and output point throughout the system and compare to recorded results from individual testing. Measure the total output of the system.
- Attempt to simulate different loads and repeat the step above for unusual and extreme conditions.

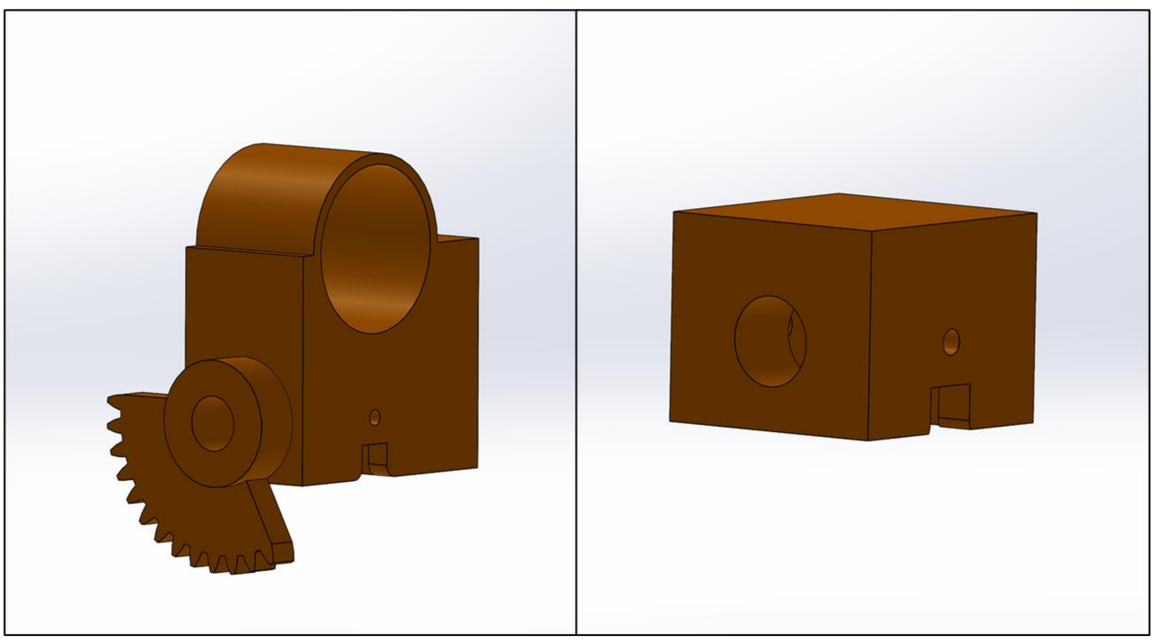
6.2.5 Mechanical Testing

The mechanical design testing differs greatly from the electrical or software testing as the number of factors that determine viability is much larger and more difficult to determine prior to the manufacture of parts. These factors are caused by the fact that the mechanical components will undergo real-world variance, such as manufacturing tolerances, which are difficult to predict in all cases. In contrast, electrical circuits and software are both able to be simulated in almost all cases and are typically able to be tested prior to the manufacture or assembly of costly components. Because of the fact that physical components had to be manufactured prior to mechanical testing, it was important to incorporate as many of the known factors affecting the end-product into the design.

6.2.5.1 Mechanical Tolerance

Due to the fact that the final mechanical assembly is comprised of both COTS components, which are made of different materials and maintaining different tolerances, and custom 3D printed components, the tolerances built-in to the design are difficult to calculate and required post-manufacture fitment. It is for these reasons that a method of printing dimensionally vital sections of complete parts was used to test mechanical components, while using a minimal amount filament in the process.

Figure 36 - Tolerance Testing Component



An example of this process is shown above in Figure 36, in which a section (shown on the right) was extracted from the collective lever collar-bearing combination (shown on the left). The smaller subsection of the complete bearing part contains five of the six dimensionally vital features, which are the two shoulder bolt holes, the two screws holding the friction shaft collar, and the channel that holds the shaft collar itself in the center of the part. This subsection allowed for a fitment test of all related components while only requiring 4.5 hours to print with 15 grams of materials. This is considerably less when compared to the 17.5 hours and 65 grams of material needed for the entire part.

The inherent dimensional changes that occur between the design and the fabricated piece, in the case of 3D printed components, are largely caused by the cooling of the part. During the cooling process some features of a part, such as holes and gaps, are likely to shrink, thus requiring edits to oversize these features in the design. While some of these changes in dimension are heavily influenced by surrounding features, such as the amount of material around a printed hole, in general the added dimensional tolerance can be extrapolated to all features serving the same purpose. This allowed for a singular test fit result to be used across all parts that interact with the common part that is being tested on. In our design, one such common part is the main collective lever arm, which has a total of six printed components that are directly designed around its diameter. The ability to determine a singular additional clearance required to fit this diameter in a printed component is vital as it allowed us to save time and resources when fabricating the end-product.

6.2.5.2 Mechanical Strength

While the total assembly is not expected to undergo large amounts of physical stress, the final assembly must maintain a workable level of strength. Components that are of particular concern when considering their strength are 3D printed components that are of low density, like the collective heads, and small components that simply do not have many layers of filament. In order to test the strength of the printed parts, test pieces were printed and assembled in place when possible. These components were then used in the same manner they would be in the final assembly and concerns, such as high friction points that could cause material degradation, were noted. Any possible concerns found during this testing were considered and design or fabrication changes were incrementally added to compensate for these potential failures.

While occasionally a change to the actual design of the part was required, more often changes to the fabrication procedure were enough to ensure proper function while meeting strength requirements. These changes normally revolve around changes to print density or the orientation in which the part is printed. The orientation of the piece during printing has an effect on the strength as 3D printed pieces are inherently prone to sheer force between the print layers due to the additive nature of 3D printing. However, changing the orientation from the default can often lead to much higher material usage and print times. This is caused by the fact that additional printing of support structures in the print are likely. One example of this is the mounted plate that connects the collective throttle to the encoder via the inner shaft. The piece itself is constructed of a small arm that must fit inside the inner diameter of the collective lever and must be able to translate the rotational force of the throttle to the encoder. Due to the overall size requirements of this piece, the ability to simply increase the total mass of the piece to increase its strength was not an option. Therefore, we tested multiple printing orientations of the piece and measured the rotational force able to be exerted on the arm in the same axis as the piece would experience in the final assembly. This type of experiment allowed us to balance the print time, material usage, and strength of our parts.

6.2.5.3 Mechanical Assembly

In order to ensure that the final assembly of all mechanical components meets the initially defined requirement specifications, testing of the complete mechanical assembly was done. The engineering requirement specifications that are able to be directly tested via the physical assembly alone are the collective base size restriction, the collective lever angular travel range, and the requirement for multiple collective heads. Once fully assembled, the collective base size was quickly and easily verified to match the CAD design via calipers and a measuring tape, which ensured that the specification was met. The total range of motion for the collective lever is limited by the spacers that are between the collective base side plates. These spacers are placed in an orientation that allows for a total of 45 degrees of movement, which was confirmed via reading the raw encoder values reported with the lever touching each spacer. Finally, the fitment component of the specification requiring the ability to attach multiple collective heads was tested. This is simply a test to ensure that the multiple head designs were able to be affixed to the common thread on the end of the collective lever while maintaining acceptable tolerances.

6.3 Specification Testing

After completing the design and construction of the helicopter collective control unit, it was necessary to test it against the specifications set forth at the beginning of this project. These specifications detail the requirements which the final product must adhere to in order to be considered a successful design. Shown below is a table detailing the results of the specification testing which we performed.

Table 18 – Final Specification Testing Results

Description	Specification	Final/Measured
Cost	< $500	$329.62
Weight	< 20 lbs.	4.3 lbs. (max)
Line of Sight Wireless Connection Range	≥ 15 ft	Satisfied (Measured at 17 ft)
Idle Runtime (Wireless)	> 10 hours	Satisfied (Tested up to 11 hours)
Collective Base Size	< 1 cu. ft	2.9"x11.3"x8.25" = 0.1565 cu. ft
Minimum Measurable Change of Lever Angle	< 1°	0.129°
Angular Variance at Idle	±5% of total angular travel	±0.49% of total angular travel
Number of Unique Collective Heads	≥ 2 Heads	2 Heads
Angular Poll Rate	≥ 3 Hz	10.1 Hz
Collective Lever Angular Travel Range	≥ 30°	44.18°
Calibrated Digital Output Range	Min: 0, Max: 1023	Satisfied
Operating System Compatibility	Windows 10	Satisfied

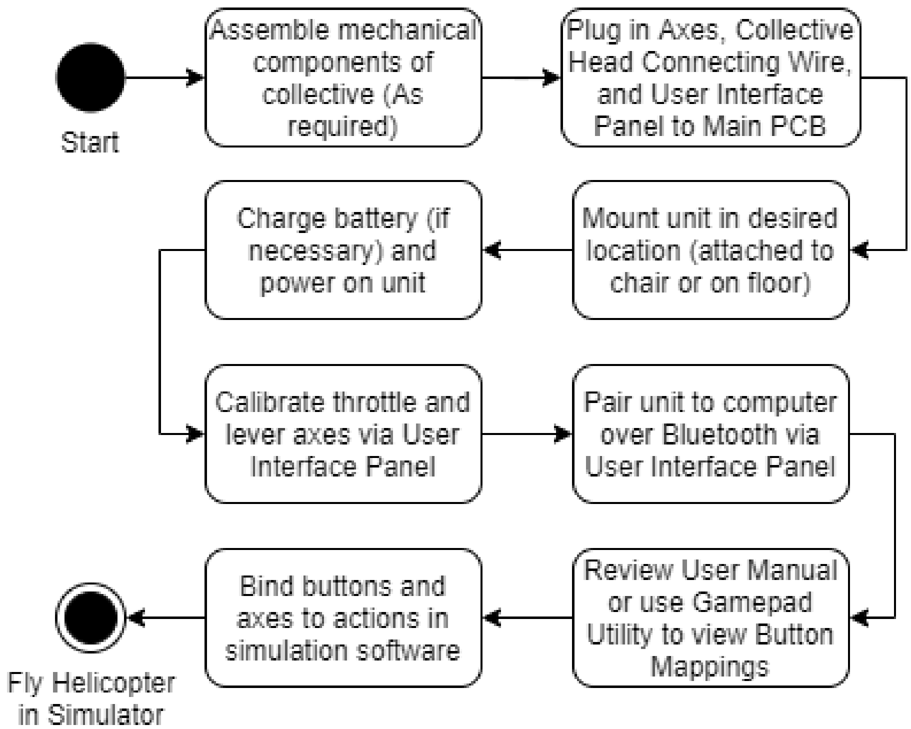
As can be seen above, all of our original requirement specifications were satisfied by our final design. This means that our design was able to effectively solve the problem which we aimed to tackle, namely, creating an affordable, realistic helicopter collective control.

7.0 Administrative Content

7.1 Product Setup and Operation

The overarching goal of this project as a whole was to create an affordable, realistic, fun-to-use helicopter collective for use in flight simulator software. A significant factor in meeting this goal was the ease of setup and use of the final product. This product is primarily targeted towards the flight sim hobbyist and typical consumer, not professional flight simulator users or companies. Thus, the helicopter collective has been designed so as to incur as little setup and configuration by the end user as possible in order to work towards an "out-of-the box" simulation solution. This is crucial as a typical consumer expects products to require little setup and interact with their existing setup as seamlessly as possible. Shown in the Figure below is a broad overview of the setup and usage of the helicopter collective which will be expanded upon in the following sections.

Figure 37 - Product Setup Flowchart



7.1.1 Assembly and Physical Installation

The initial stage of product setup is the essential physical setup of the unit. While we produced a single prototype from scratch and therefore do not necessarily ourselves have to worry about end-user assembly, it is important to consider what physical assembly would be required if this product were to go to market. Towards this end, the collective has been designed so that as many components as possible could be assembled by the manufacturer, such as the entire collective lever assembly, user interface plate, and collective heads. Thus, the only physical installation required by the end user would require assembling the base unit and connecting the electronics to the main PCB. To simplify this process, the base unit has been designed so that it is able to be assembled with a handful of screws and common hand tools. To simplify the connection of the electronics, the collective head wire, lever axis encoder, throttle axis encoder, and user interface panel are pre-wired to plugs which can be easily plugged in to matching ports on the main PCB.

After initial physical assembly, the unit must be installed in the location which it is intended to be used. As we intend for the unit to be portable, the unit is able to be both mounted to a dedicated chair to simulate how a collective is typically present in real helicopters or is able to simply mount to feet which allow the unit to sit on the floor. Due to the usage of Bluetooth and the presence of a rechargeable lithium-ion battery in the collective unit, a permanent installation near the simulation host device or a power source is not required.

7.1.2 Initial Setup and Configuration

Once the unit has been assembled and mounted in its operating location, the user must first undergo a few configuration steps to calibrate and connect the collective to their desired device. First, the helicopter collective must be connected via USB-C to a power source in order to charge the internal battery. The collective unit will be able to be used entirely unplugged for the duration of the battery's charge or may be left plugged in for extended usage. Once charged, the user must then select the collective head which they would like to use. Once selected, the user will simply plug the collective head they are going to use into the exposed plug in the top of the collective lever assembly and then screw on the collective head nut to secure the collective head in place.

After powering the unit and installing the desired head, the unit may be powered on. At this point, if the unit has been previously calibrated and connected to a device, the user may simply begin using the device. However, upon first use the axes must be calibrated and a Bluetooth connection must be established to a remote device. To calibrate the lever and throttle axes, the user will utilize the face buttons and LCD on the user interface plate to navigate to the Calibration menu option. The LCD will provide instructions to move the lever and throttle to their extreme minimum and extreme maximum positions so as to calibrate the unit. This step is crucial to the proper operation of the collective and is required to ensure accurate simulation. After calibrating the axes, the user can then navigate to the Bluetooth Pairing menu option on the LCD and begin searching for a device to pair with. The collective will then appear as a pairable Bluetooth device on the user's host computer and may be connected to. After a connection has successfully been established, the collective unit is operational and ready to be used.

7.1.3 Flight Simulation Software Configuration

While the collective unit itself may be fully assembled and operational, some additional configuration may be required on the user's device which is running the simulation software in order to utilize the helicopter collective properly. Firstly, the user can use Windows 10's built in Gamepad calibration to test that the inputs of the collective are recognized properly by their computer. However, the collective unit simply reports generic Button and Axis inputs to the computer, which the simulation software does not initially associate with an operation in the simulation. Thus, the end user must refer to the product manual to see what buttons are associated with which inputs on the collective head there are using, and which input axis is associated with the lever and throttle axes. After reviewing these mappings, the user must then enter their simulation software of choice and bind these operations to functions in the simulation software. While this takes some effort, this also allows much more freedom from the user's perspective. For example, while the user may only have two collective heads, they can theoretically bind the controls on the head to any control in the simulation software they wish. While it may provide a less-than-realistic experience when using a mismatched collective head between real life and the simulation software, it is still available as an option so that the end user is not limited to solely simulating the few helicopters associated with the physical collective heads available on the unit.

7.2 Final Budget

Table 19 – Final Budget

Item Name	Part Number	Price Per Item	Quantity	Cost
18-8 Socket Screw	92196A283	$1.01	5	$5.05
Steel Hex Nut, Class 8, M8 x 1.25 mm Thread	90592A022	$0.53	1	$0.53
18-8 Stainless Steel Washer	98689A116	$0.07	1	$0.07
Stainless Steel Washer, Number 10 Screw Size	90107A011	$0.04	5	$0.20
Steel Hex Nut, 10-32 Thread Size	90480A195	$0.02	5	$0.10
10mm x 65mm M8x1.25 Shoulder Screw	92981A757	$3.92	1	$3.92
Shaft Collar, 10mm Shaft, 24 mm OD	7165N112	$14.20	1	$14.20
10 ft. Sch. 40 1 in. PVC Pipe	531194	$5.28	1	$5.28
10 ft. Sch. 40 1.5 in PVC Pipe	531111	$7.96	1	$7.96
eSun PLA PRO (PLA+) Filament 1 kg	781520911976	$22.99	1	$22.99
17 Position Male Circular Molex Connector	2021131730	$11.24	2	$22.48

17 Position Female Circular Molex Connector	2021131710	$21.68	1	$21.68
20-24AWG Tin Crimp Connector	430300001	$0.05	54	$2.81
18 Pos Male Molex Receptacle	430251800	$0.99	1	$0.99
24 Pos Male Molex Receptacle	430252400	$1.45	1	$1.45
6 Pos Male Molex Receptacle	430250600	$0.46	2	$0.92
EG2011-ND SPST Push Button	PS1024ABLK	$1.27	1	$1.27
EG2387-ND SPDT ON-OFF-ON Switch	100SP3T8B13M1QEH	$4.04	3	$12.12
EG2372-ND SPDT ON-ON Switch	100SP1T8B13M1QEH	$3.58	1	$3.58
EG2392-ND SPDT OFF-MOM Switch	100SP4T6B11M2QEH	$4.54	2	$9.08
EG1923-ND SPST Push Button	RP3502MABLK	$2.72	1	$2.72
1024 Position Absolute Magnetic Encoder	EMS22A50-B28-LS6	$35.51	2	$71.02
2.00 mm 6 Pos Female Plug Connector	5024390600	$0.18	2	$0.36
22-26AWG Tin Crimp Connector	5024380100	$0.06	12	$0.72
6x6x4.3mm Momentary Switch	Generic	$0.05	4	$0.18
2mmx8.5"x11" Adhesive-Backed Cork Sheets (2)	Generic	$10.77	1	$10.77
LCD1602 16x2 Display	ACM1602K-FL-YBW	$4.76	1	$4.76
Anti-Vandal OFF-MOM Push Button Switch	PV2S240NN	$4.53	3	$13.59
360 Brass Round Bar	BRR18	$5.73	1	$5.73
8"x11"x1/16" 3003-H14 Aluminum Sheet	S3063	$15.26	2	$30.52

3/4"x3/4"1/16" Aluminum Angle	A334116	$4.25	2	$8.50
SPST OFF-ON Rocker Switch	RR3130ABLKBLKES	$2.35	1	$2.35
Additional PCB Components	N/A	$14.92	1	$14.92
4000 mAh 18650 Li-Ion Battery	N/A	$3.80	1	$3.80
PCB Fabrication (Assembly, SMT Components)	N/A	$23.00	1	$23.00
Total Cost	**$329.62**			

7.3 Conclusion

Overall, we were able to produce a helicopter collective control that satisfied all of the requirements and objectives that we set out to achieve at the beginning of this project. As can be seen in the figures below, the final unit is portable, reports accurate inputs to the host PC, resembles a real-life collective, is able to accommodate multiple collective heads, and was extremely budget friendly compared to similar commercially available options, thus satisfying our original goals of portability, modularity, low-cost, and accuracy. Furthermore, in our opinion, the helicopter collective control satisfies its primary purpose: it is easy and fun to use in a flight simulator. We were able to test the unit in two different consumer flight simulator products and it functioned without problem in each, and vastly improved the experience of flying a helicopter by increasing the realism factor in both appearance and operation.

Figure 38 – Complete Helicopter Collective Control Unit

Figure 39 – Interchangeable UH-1 Huey and Bell 206 Heads

8.0 Appendices

8.1 Copyright Permissions

Send an Email

* indicates a required field.

Subject: From www.faa.gov: Email

* Your email address: david.green@knights.ucf.edu

* Your message: I am a Senior in Computer Engineering at the University of Central Florida. As part of our capstone project, I would like to utilize some figures from the Pilot's Handbook of Aeronautical Knowledge, FAA-H-8083-25B. Namely, Figure 6-4 and Figure 6-3. I would also like to utilize Figure 3-1 from the Helicopter Flying Handbook, FAA-H-8083-21B. As these are government works, I presume they are copyright free, however I am unable to validate this. May I have permission to use these Figures for academic purposes?

Thank you,
David

I'm not a robot

reCAPTCHA
Privacy - Terms

Send Email

HB Hyde, Barry (FAA) <Barry.Hyde@faa.gov>
Fri 4/30/2021 9:00 AM
To: David Green
Cc: 9-AMC-AFS630-Comments (FAA) <AFS630Comments@faa.gov>

Happy Friday morning, Mr. Green!

Thank you for your great question in reference to using the FAA's handbooks. It is our understanding that anything placed in the public domain, such as our handbooks, are available without copyright concerns or further authorizations. Hope this response answers your questions!

Thank you for taking the time to contact us and good luck with your new project.

Dr. Barry Hyde
Aviation Safety Analyst
Airman Testing Standards Branch, AFS-630
Barry.Hyde@faa.gov

To: copyright@bluetooth.com

I am a Senior in Computer Engineering currently attending the University of Central Florida. As part of our capstone project, I would like to utilize Figure 2.7, "GATT Profile Hierarchy", from the Bluetooth 5.2 Core Specification. This information will be utilized solely for academic and research purposes and will not be used commercially. May I have permission to utilize this Figure for academic purposes?

Thank you,
David

LP

Lindsay Peattie <lpeattie@bluetooth.com>
Mon 4/26/2021 10:05 AM
To: David Green

Hi David,

Thank you for your mail. Yes – you are welcome to use the figure from the specification – please ensure you attribute it to the Bluetooth SIG accordingly.

Thank you!
Lindsay

To: ClientSuccess@microchip.com

I am a Senior currently studying Computer Engineering at the University of Central Florida. For our capstone project, we are planning to utilize multiple Microchip components. I would like to utilize images from the MCP73831/2 datasheet and the BM70/BM71 User's Guide. These images and information will be used for academic and research purposes. May I utilize the diagrams and figures from these documents?

Thank you,
David

Case History (Reverse Chronological Order):

Comment by : Thasneena Muhammedali C 05/03/2021 06:54:14 GMT

Hi David,

Thank you for contacting the Microchip Technical Support,

If you would like to reproduce, translate, and/or reuse Microchip copyrighted material for personal use or educational (non-profit) purposes, Microchip's written permission is not required. There is no restrictions to use images and figures of datasheet for your academic purpose.

Please refer the following link for more information about the copyright material usage and legal aspects,

https://microchipsupport.force.com/s/article/Copyright-material-usage-and-legal-aspects

All the best!

Thanks & Regards

Thasneena

To: americus@bourns.com

I am a Senior currently studying Computer Engineering at the University of Central Florida. For our capstone project, we plan to utilize the EMS22A product line. I would like to utilize some of the figures in the EMS22A datasheet. These figures will be used for academic and research purposes only. May I have permission to use these images?

Thank you,
David

Frances Robles <Frances.Robles@bourns.com> on behalf of AMERICUS <americus@bourns.com>

Mon 4/26/2021 4:17 PM

To: David Green; AMERICUS <americus@bourns.com>

Good Afternoon David,

Thank-you for taking us into consideration. Bourns gives you permission to use these images.

Thank-you for choosing Bourns and have a great day.

Frances on behalf of Americus

8.2 References

Atmel. (2014). *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V Data Sheet.* Retrieved from Microchip: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

Bluetooth SIG Core Specification Working Group. (2019, December 31). *Core Specification 5.2.* Retrieved from Bluetooth Technology Website: https://www.bluetooth.com/specifications/specs/core-specification/

Bluetooth SIG HID Working Group. (2011, December 27). *Bluetooth Specifications.* Retrieved from Bluetooth Technology Website: https://www.bluetooth.com/specifications/specs/hid-over-gatt-profile-1-0/

Bourns, Inc. (2018, May 17). *EMS22A - Non-Contacting Absolute Encoder*. Retrieved from Bourns: https://www.bourns.com/docs/product-datasheets/EMS22A.pdf

Doland, J., & Valett, J. (1994). *C Style Guide.* Greenbelt, Maryland: Goddard Space Flight Center.

EE Power. (n.d.). *Potentiometer*. Retrieved from EE Power: https://eepower.com/resistor-guide/resistor-types/potentiometer/#

Espressif Systems. (2021, February). *ESP32-WROOM-32SE Datasheet.* Retrieved from Espressif Systems: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32se_datasheet_en.pdf

eSUN Industrial Corporation. (2018, March 1). *PLA PLUS/PLA+ Filament Safety Data Sheet.* Retrieved from eSUN 3D: https://www.esun3d.net/UploadFiles/Download/MSDS_Esun_PLA_PLUS_filament.pdf

Federal Aviation Administration. (2016). *Pilot's Handbook of Aeronautical Knowledge.* Oklahoma City: United States Department of Transportation, Federal Aviation Administration, Airman Testing Standards Branch.

Federal Aviation Administration. (2019). *Helicopter Flying Handbook.* Oklahoma City: United States Department of Transportation, Federal Aviation Administration, Airman Testing Branch.

Komodo Simulatons. (2021). *UH-1 - Komodo Simulations*. Retrieved from Komodo Simulations: https://komodosimulations.co.uk/collections/collective/products/b207

Linear Technology. (n.d.). *LTC1730-4/LTC1730-4.2 Lithium-Ion Battery Pulse Charger with Overcurrent Protection.* Retrieved from Analog Devices: https://www.analog.com/media/en/technical-documentation/data-sheets/1730fs.pdf

Logitech. (2021). *Logitech G X52 Professional Space & Flight Simulator HOTAS Joystick*. Retrieved from Logitech: https://www.logitechg.com/en-us/products/space/x52-pro-space-flight-simulator-controller.945-000022.html

Maker.io Staff. (2021, January 6). *Understanding the Basics of Infrared Communications*. Retrieved from Digi-Key: https://www.digikey.com/en/maker/blogs/2021/understanding-the-basics-of-infrared-communications

Microchip Technology. (2014, September 12). *BM70/71 Bluetooth Low Energy Module User's Guide.* Retrieved from Microchip: http://ww1.microchip.com/downloads/en/DeviceDoc/50002542A.pdf

Microchip Technology. (2017, October 25). *BM70/71 Data Sheet.* Retrieved from Microchip: http://ww1.microchip.com/downloads/en/DeviceDoc/BM70-71-Bluetooth-Low-Energy-BLE-Module-Data-Sheet-DS60001372H.pdf

Microchip Technology. (2020, Feburary 28). *MCP73831/2 Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers Datasheet.* Retrieved from Microchip: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf

Microchip Technology. (2020, February 28). *RN42/RN42N Data Sheet.* Retrieved from Microchip: https://ww1.microchip.com/downloads/en/DeviceDoc/Class-2-Bluetooth-Module-with-EDR-Support-DS50002328B.pdf

Microchip Technology. (2020, February 28). *RN4870/71 Data Sheet.* Retrieved from Microchip: https://ww1.microchip.com/downloads/en/DeviceDoc/RN4870-71-Data-Sheet-DS50002489E.pdf

Pini, A. (2018, December 11). *Rotary Encoders Digitize Mechanical Position*. Retrieved from Digi-Key: https://www.digikey.com/en/articles/how-to-use-rotary-encoders-convert-mechanical-rotation-digital-signals

Smoot, J. (2018, November 20). *Rotary Encoder Options: Absolute or Incremental?* Retrieved from Digi-Key: https://www.digikey.com/en/articles/rotary-encoder-options-absolute-or-incremental

Smoot, R. (2021). *What's an Encoder's PPR, CPR, and LPR?* Retrieved from CUI Devices: https://www.cuidevices.com/blog/what-is-encoder-ppr-cpr-and-lpr

Soto, J., Haley, A., & Williamson, B. (2018). *Angular Position Project*. Retrieved from UCF Department of Electrical and Computer Engineering: https://www.ece.ucf.edu/seniordesign/sp2018su2018/g14/

STMicrocelectronics. (2018, September). *L78 Datasheet.* Retrieved from STMicrocelectronics: https://www.st.com/resource/en/datasheet/l78.pdf

STMicroelectronics. (2019, July). *LM217, LM317 Datasheet.* Retrieved from STMicroelectronics: https://www.st.com/resource/en/datasheet/lm217.pdf

Texas Instrument. (2019, March). *TPS63070 2-V to 16-V Buck-Boost Converter With 3.6-A Switch Current Datasheet.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/tps63070.pdf?ts=1619509194596

Texas Instruments. (2016, July). *CC2640 SimpleLink Bluetooth Wireless MCU Data Sheet.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/swrs176b/swrs176b.pdf?ts=1619439593000

Texas Instruments. (2018, August). *MSP430FR6989.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/msp430fr6989.pdf?ts=1619503896399&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430FR6989

Texas Instruments. (2019, October). *TPS6302x High Efficiency Single Inductor Buck-boost Converter with 4-A Switches Datasheet.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/tps63020.pdf?ts=1627880401069&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS63020

Texas Instruments. (2020, January). *BQ25611D I2C Controlled 1-Cell 3.0-A Buck Battery Charger with USB Detection and 1.2-A Boost Operation Datasheet.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/bq25611d.pdf?ts=1619466685066&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ25611D

Texas Instruments. (2020, June). *LM1117 800-mA, Low-Dropout Linear Regulator Datasheet.* Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/lm1117.pdf?ts=1619433318012&ref_url=https%253A%252F%252Fwww.google.com%252F

United Nations Economic Commission for Europe. (2019). *Globally Harmonized System of Classification and Labelling of Chemicals (GHS).* New York and Geneva: United Nations.

USB Implementer's Forum. (2001, May 27). *Device Class Definition for HID 1.11*. Retrieved from USB-IF: https://www.usb.org/document-library/device-class-definition-hid-111

USB Implementer's Forum. (2021, April 5). *HID Usage Tables 1.22*. Retrieved from USB-IF: https://usb.org/document-library/hid-usage-tables-122

Wi-Fi Alliance. (2021). *Wi-Fi Direct*. Retrieved from Wi-Fi Alliance: https://www.wi-fi.org/discover-wi-fi/wi-fi-direct

Zigbee Alliance. (2020). *What is Zigbee?* Retrieved from Zigbee Alliance: https://zigbeealliance.org/solution/zigbee/