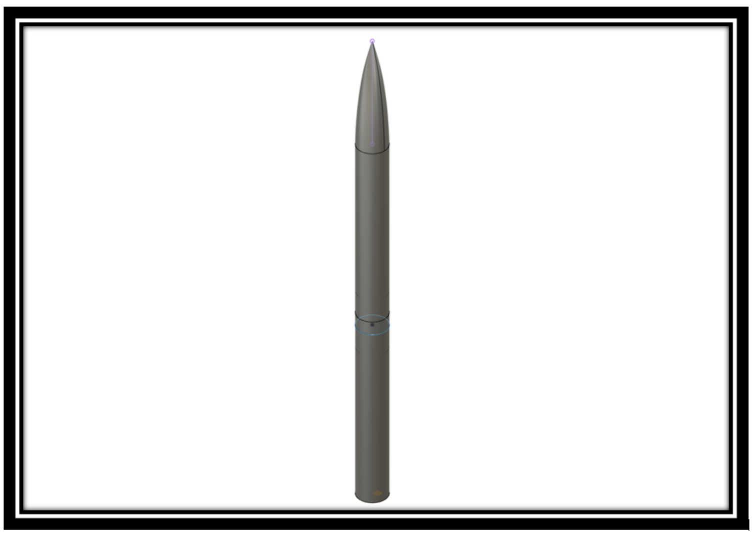
Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
Fall 2022



Team 4
TVC GIMBAL ROCKET

Demoria Sherman
Milad Noori
Murphy Balsomi
Sergio Guerrero

Revision History

Revision	Date	Author(s)	Description
0.1	08.23.2022	MN	Document layout created and shared
0.2	09.02.2022	MN	Diagrams designed and uploaded for each section
0.3	09.03.2022	MN	Complete section 1; Introduction
0.4	09.03.2022	MN	Complete section 7; Communications Layer
0.5	09.04.2022	DS	Complete Section 2; Overview
0.6	09.04.2022	DS	Complete Section 3; Data Flow
0.7	09.04.2022	DS	Complete Section 4; Flight Computer Layer
0.8	09.04.2022	SG	Complete Section 5; Control Systems Layer
0.9	09.04.2022	MB	Complete Section 6; Rocket Recovery Layer
1.0	09.04.2022	MN	Final Review
1.1	12.15.2022	MN	Final Review and Edit

Contents

List of Figures

List of Tables

1 INTRODUCTION

TVC Gimbal Mount Rocket is designed and developed with two main requirements in mind. Be stabilized through Gimbal Mount system and be reusable through rocket recovery system, Parachute ejection system.

The ultimate goal of this model rocket is to return to the ground safely, if done successfully this would indicate a reusable rocket that can be launched and returned numerous times which will be efficient and sustainable for the industry.

The ability of re-usability opens up many doors for this rocket. If engineered properly it can be broken into sub-parts and distributed to younger aspiring engineers to get hands-on experience in constructing and testing their rockets.

Also, it can be used in classrooms across the nation as a demonstration to students of the heights that modern technologies can achieve in todayâs world. Lastly, it can be scaled up along other available technologies to perform a more complex task such as taking people or satellites to low earth orbit.

TVC Gimbal Rocket is developed in four layers of architecture:

Layer 1: Flight Computer

Layer 2: Control System (Gimbal Mount)

Layer 3: Rocket Recovery (Parachute Ejection)

Layer 4: Communications

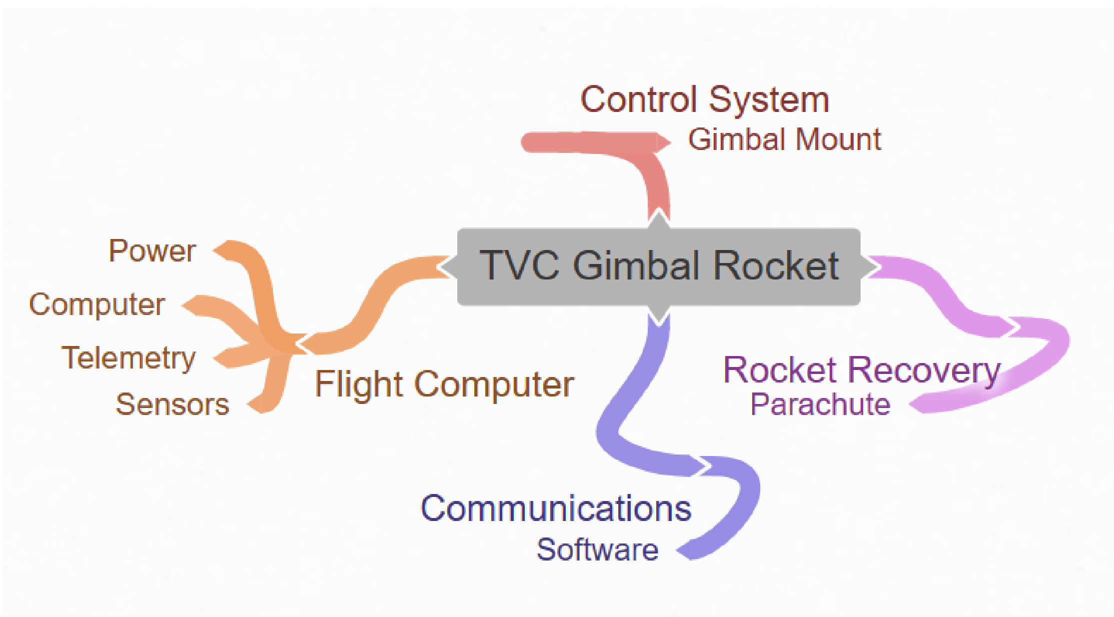


Figure 1: A simple architectural view of the system

2 System Overview

We will start with a Script that culminates the overall data recovery system of the flight computer, then integrate the subsystem in the code for controlling the TVC mount with the IMU data. After sufficient testing we will move on into incorporating the code for the ejection system and other peripherals such as LED's, buzzers, buttons etc. Then after we have culminated everything we verify that it works functionally we can start looking at how to most efficiently structure the code to reiterate for better efficiency.

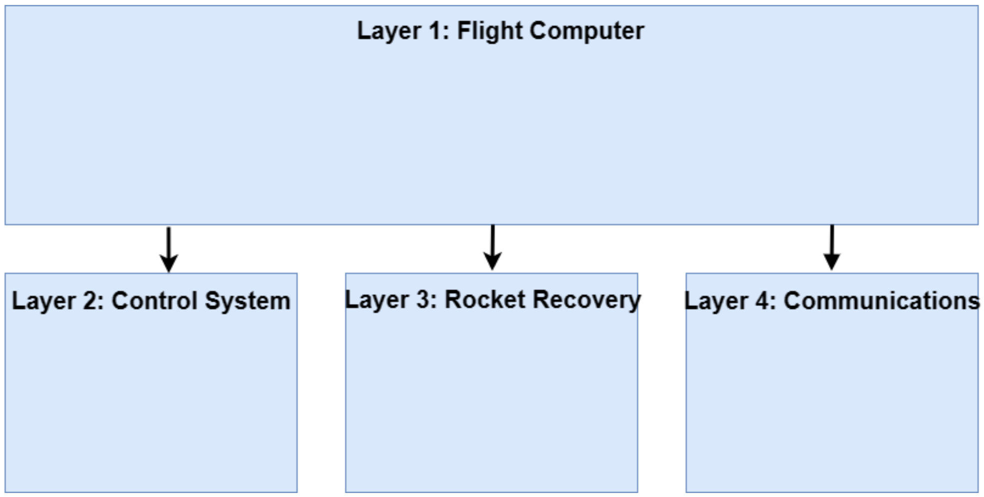


Figure 2: A simple architectural layer diagram

2.1 Layer 1: Flight Computer

The function and purpose of his layer is the avionics of the model rocket.It interacts with the control system via the servo's connected to the controller. The rocket recovery system is also connected to the flight controller via a MOSFET switch. Finally, the communication sub layer is connected via the telemetry modules on board and on the ground. This layer provides switching via MOSFETS, thrust vector control via the on board imu and the servo's attached to the engine of the rocket. It also provides data such as altitude, temperature, pressure etc. via the on board IMU and barometric pressure sensor. This avionics flight computer provides useful real time data and stores it in an SD card on the micro controller, The data can also be seen relatively live from the ground computer. Meanwhile the rocket is using this data to detect it's orientation and altitude for staging.

2.2 Layer 2: Control System

The Control System Layer consists of all subsystems that work to control the rocket on it flight. The subsystems have been combined to on subsystem that better depicts its purpose in this project. There are four main parts to this system that all work together to control the rocket, those parts being the Gimbal Mount, Servos, Engine, and the Battery. Each of these will be linked together in some way either it being physically or through software to assure control on the rocket.

2.3 Layer 3: Rocket Recovery

For the rocket recovery system, the team will be utilizing a N Channel Mosfet Switch which will ignite a ignition charge and release two parachutes at apogee. The Mosfet switch will be controlled by the Teensy 4.1 which will send a high signal to the proper pin which in turn signals the switch to turn on thus igniting the charge. The ignition charge is a homemade charge consisting of a old Christmas light, filled with a precise amount of black powder. When current is sent thought the light, it will turn on and light the powder which is inside releasing a charge sufficient to release the parachutes.

2.4 Layer 4: Communications

For this layer we are currently using Arduino IDE and visualize the data statistics manually. The data to the application will be fetching to this layer from Layer 1, Flight Computer Layer. The focus currently is to focus on achieving product goals. The communication layer can be extended to another project to build an application that would provide better visualization using data from Telemetry. The current version saves data in the memory card that can be used by the user for further analysis. Developing a ground system application can enable user to view the telemetry data while the rocket is in flight. A good example to follow for project extension is Mission Planner desktop application by Ardupilot.

3 Subsystem Definitions & Data Flow

Here you can see a general layout of the data flow of the entire system with subsystems. As you can see in figure 3 data will be communicated back and forth using an I2C connection protocol to the micro controller. telemetry will be connected to the micro controller using the same communication protocol as well as a wireless communication to the ground telemetry module. The data flow coming from the two sensors on board will be used in software to be converted from raw data to data necessary to control the servo's connected to the engine of the rocket. The data from sensors Will also be used to send a signal to the parachute ejection system when at the correct altitude. data and status will be sent wireless via the telemetry modules to a ground computer. through out the flight the rocket will be showing which state it's in via data being sent from the micro controller to the peripheral devices connected such as LED's Buzzers and buttons. this gives a basic general description of the data flows being sent through the computer during flight.

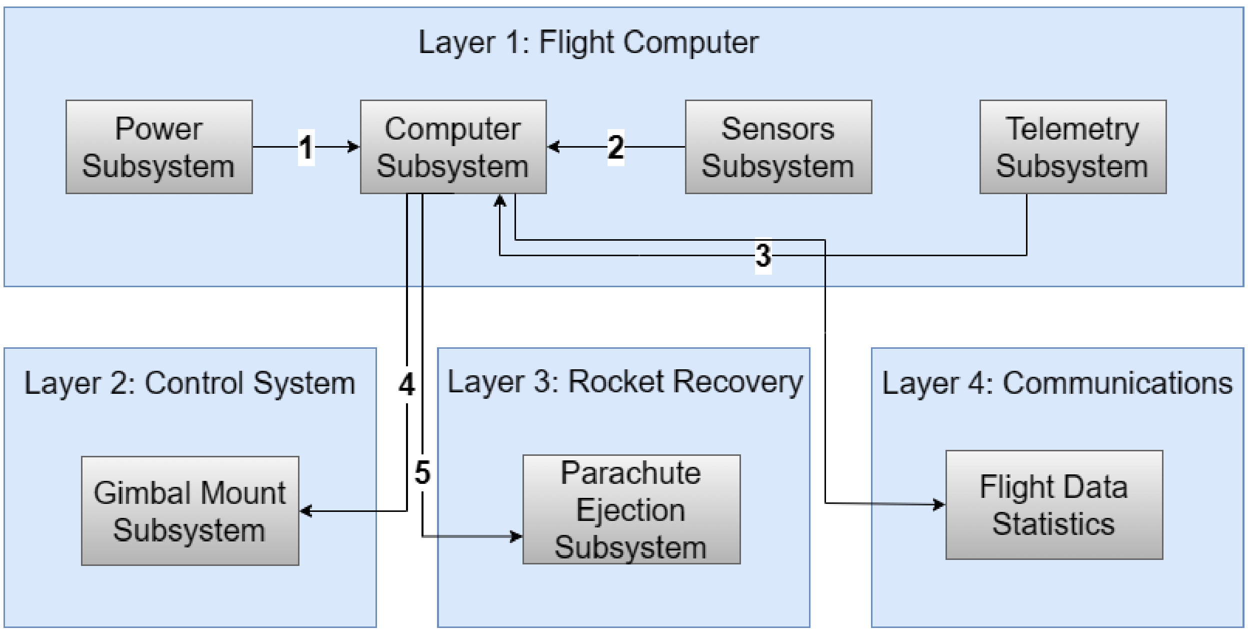


Figure 3: A simple data flow diagram

4 Layer 1 (Flight Computer) Subsystems

4.1 Subsystem 1 - Power

The power system is built to take in voltage in a range of voltages from 5V to 12V giving it a range of lipo batteries to use to power the device overall, as the buck converter steps the voltage down to 5V. This power system is then controlled via a switch on the flight computer by the user. then the 5V voltage is then directed to power the necessary subsystems of the flight computer. It powers all subsystems in the flight computer.

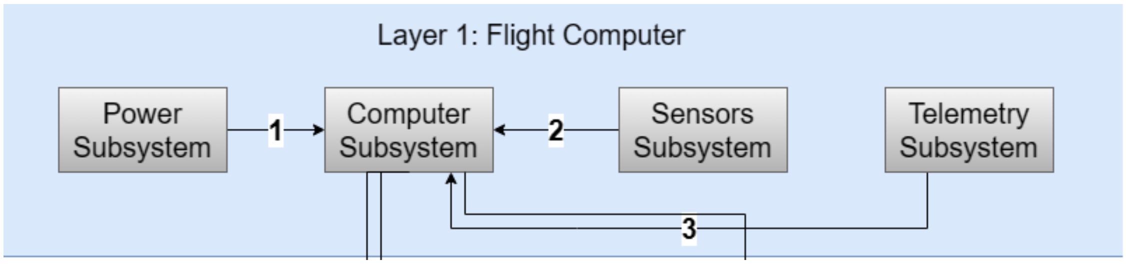


Figure 4: Example subsystem description diagram

4.1.1 Assumptions

We assume this will sufficiently power the system during flight without interference with other devices sensitive to it's effects.

4.1.2 Responsibilities

The responsibility of the subsystem is to power the overall systems on the flight computer efficiently without any trouble.

4.1.3 Subsystem Interfaces

The inputs to the system is simply the positive and negative inputs from the battery to the inputs in the buck converter, before connecting to the buck converter module we included a switch for the user to initiate the sequence.the positive and negative pins out the buck converter then power the 5V layer on the PCB and the ground layer on the PCB as well as traces to power the mosfets.

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Buck Converter	input 1 Positive input 2 Negative	output 1 positive 5V output 2 Negative 5V

4.2 Subsystem 2 - Computer

The Micro Computer Subsystem is the brain of the entire flight computer system and controls and processes the data being sent to it by it peripheral devices connected to it. Powered by the 5V from the buck converter, it connects to the telemetry module via the TX,RX pins on the micro controller which connects to the radio telemetry module that connects to the ground radio telemetry module, this is will allow communication to transfer data wireless. The micro controller also connects to the sensory subsystem via the SDA, SCL pins on the controller, the micro controller will process data from the sensor devices via a I2C communication protocol and send that data to the gumball mount subsystem via the PWM pins on the micro controller.a Buzzer, Led, and mosfet switches will also be controlled by the micro controller by PWM pins. In future design we wish to include extra header pins to be used by other devices and also a button.

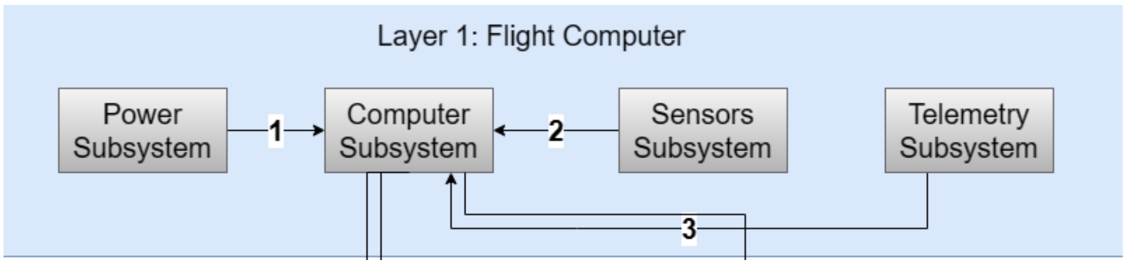


Figure 5: Example subsystem description diagram

4.2.1 Assumptions

We assume this flight computer will be capable to control all the devices connected to it during flight and will be sufficient enough to use for thrust vector controlling and processing data throughout the project.

4.2.2 Responsibilities

This system is responsible for controlling all subsystems of the entire rocket. It's responsible for controlling the servo's of the gumball mount system, it's responsible for gather accurate data from the sensors on board and sending the signals at the correct moment for the parachute ejection system. it's responsible for detecting launch , apogee, flight, landing, and gliding. It's responsible for firing the mofets for any switches or staging needed during flight. This device functions as the main controller of the rocket itself. As well as gathering data during flight.

4.2.3 Subsystem Interfaces

This device has PWM outputs to two servos (x and y), Three LED's (red, blue and green), a buzzer, and 4 mosfet switches(PY1, Py2, Py3, and, PY4). it also has an SDA and SCL pin for I2C communication connected to the two sensors, and a RX, and TX pin connected to the radio telemetry.

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	PWM pins	N/A	output 1 X output 2 Y output 3 Red output 4 blue output 5 green output 6 buzzer output 7 py1 output 8 py2 output 9 py3 output 10 py4
#xx	I2C	N/A	output 1 SDA output 2 SDL
#xx	Radio	N/A	output 1 RX output 2 TX

4.3 Subsystem 3 - Sensors

The Sensory subsystem is the way the overall flight computer connect and knows the world around it vial 2 breakout board sensors the BNO085(Inertial measurement unit) and MBP388(barometric pressure sensor). These two sensors connect to the micro controller via an SDA and SDL pin which uses I2C communication to transfer data in between themselves.

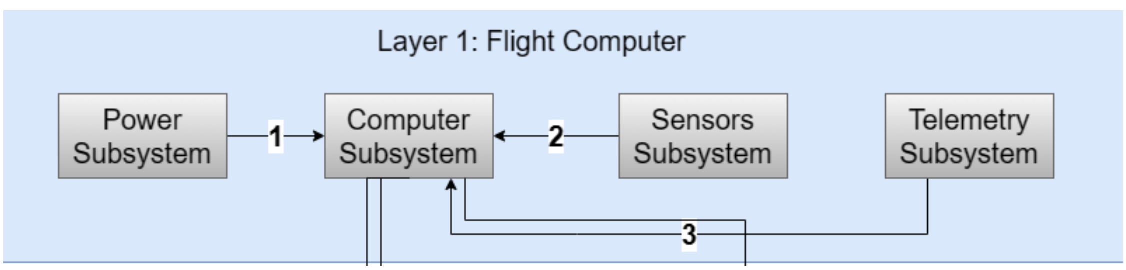


Figure 6: Example subsystem description diagram

4.3.1 Assumptions

We assume These sensors function properly and behave as expected during flight and after the parachute ejection system is fired.

4.3.2 Responsibilities

This system is responsible for sending accurate data back to the micro controller, and performing tasks such as data collection.

4.3.3 Subsystem Interfaces

These two devices are simply connected to the micro controller via two wires (SDA and SCLA). these devices are used by the micro controller as slave deice for an I2C communication protocol. and are powered by the flight computers power system (5V and Ground).

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	BMP388	input 1 5V input 2 GND	output 1 SDA output 2 SDA
#xx	Bno085	input 1 5V input 2 GND	output 1 SDA output 2 SDA

4.4 Subsystem 4 - Telemetry

The Telemetry Subsystem is the way the flight computer communicates the data back to the ground computer during flight. It connect to the micro controller via an RX and TX wire and is powered by the 5V and ground planes from the power system on the flight computer.

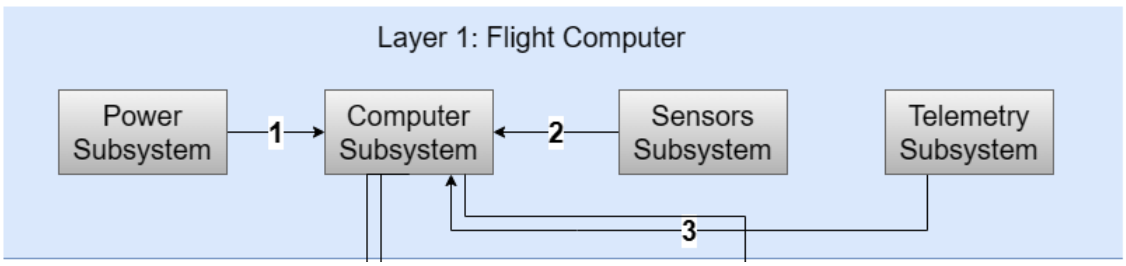


Figure 7: Example subsystem description diagram

4.4.1 Assumptions

We assume the range will be within the the limits of the telemetry modules and there isn't any communication breaks or interference's while in flight.

4.4.2 Responsibilities

This system is responsible for sending accurate data back to the ground computer for live viewing.

4.4.3 Subsystem Interfaces

It connect to the micro controller via an RX and TX wire and is powered by the 5V and ground planes from the power system on the flight computer.

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	radio telemetry module	input 1 5V input 2 GND	output 1 RX output 2 TX

5 Layer 2 (Control System Layer) Subsystems

The Control System Layer consists of all subsystems that work to control the rocket on it flight. The subsystems have been combined to on subsystem that better depicts its purpose in this project. There are four main parts to this system that all work together to control the rocket, those parts being the Gimbal Mount, Servos, Engine, and the Battery. Each of these will be linked together in some way either it being physically or through software to assure control on the rocket.

5.1 Gimbal Mount Subsystem

This subsystem consists of the Gimbal Mount, Servos, Engine, and Battery. The way in which these are all linked are as follows, the Engine of the rocket will be tied to inside the Gimbal Mounts, the engine will be wired to the Teensy 4.1 micro controller to trigger its propulsion. The Servos wired to the Teensy 4.1 that will send it various signals, then it will be mounted to the Gimbal Mounts which will move the mount/engine pro pulse the rocket in a certain direction depending on the signals received by the servos from the micro controller. And the Battery is what will give power to all the subsystems.

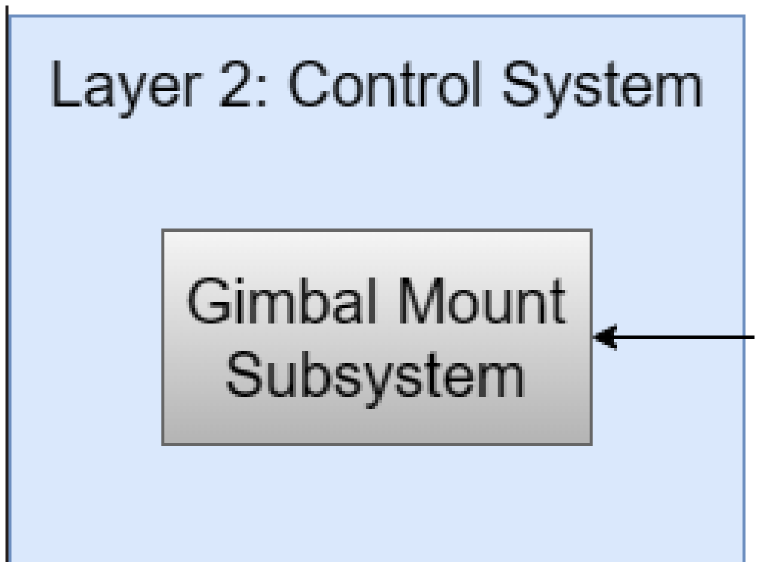


Figure 8: Example subsystem description diagram

5.1.1 Assumptions

Some assumptions that must be made for this layer to work will are as follow. To begin in order for everything to work properly it is assumed that the battery used is one that works and provides a constant 5V to all subsystems in all the layers. We then must assume that the Teensy 4.1 micro controller is sending the correct values and signals to both the engine and the servos. This will ensure that the engine will ignite when needed and the servos will direct the engine in the correct directions to produce a controller rocket. Lastly, we must also assume that the gimbal mount will withhold and stand the propulsion from the engine produced during flight.

5.1.2 Responsibilities

The Gimbal Mounts main responsibility is to be able to free move in both the x and y axis to allow the engine to pro pulse in which ever directions necessary. It should also be able to hold the engine we are placing inside of it as well as be able to withstand the force/stress generated by the engine. The Servos responsibility relies on taking in data from the micro controller and moving the servos, which is connected to the gimbal mount, in which every direction needed. The main purpose will be to stabilize the rocket so the servo should be able to take in data at a high rate and move the servo with enough force and in the right direction to be able to move the gimbal mount and engine and stabilize the rocket. The Engine has one single responsibility, and that is to ignite and provide a force so the rocket will go into flight. The Battery also has one responsibility which is to provide a power source to all subsystems of not only this layer but also to all the other layers. The battery will be tied to a buck converter which will be used to step down the voltage to 5V.

5.1.3 Subsystem Interfaces

This layer will only have 2 inputs and 2 outputs. Both inputs will come from the Teensy 4.1 micro controller, one will go to the servos and the other will go to the the engine. The input going to the servos will have data that will determine how far to move a servo. The output of this interaction would be the physical movement from the servos to the direction desired. The second input will also come from the Teensy 4.1 and will be a data bit to ignite the engine. The output of this interaction would be either an igniting engine or no action at all.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Servos	Teensy 4.1 (Stabilization Values)	Physical Servo Movement
#02	Engine	Teensy 4.1 (Data)	Ignite ON/OFF

6 Layer 3 (Rocket Recovery) Subsystems

A Rocket recovery system is any component built into the model with the intention of bringing it safely down to the ground. All recovery systems function by creating additional lift or drag to reduce the force of gravity. the rocket recovery layer has one major component: Parachute Ejection. The rocket recovery system is connected to the flight computer component, that control all the rocket system.

6.1 Parachute Ejection Subsystem

A parachute is an umbrella-like device that slows a falling body's descent by providing resistance to its motion through the air

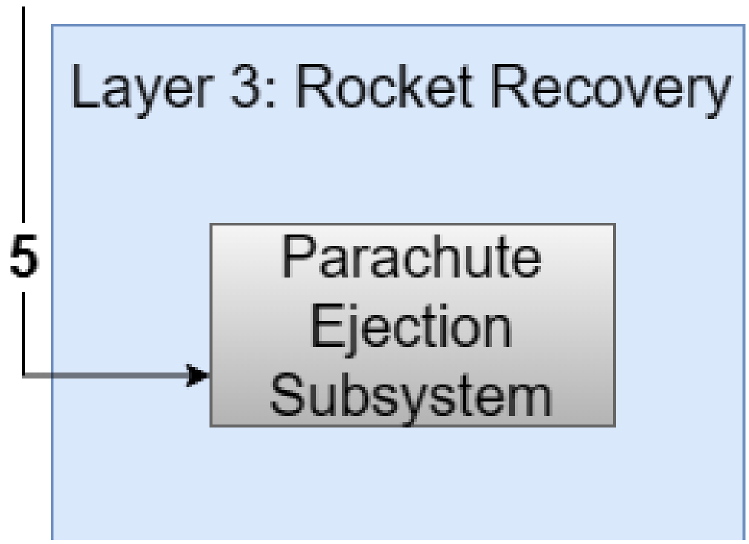


Figure 9: Rocket Recovery Diagram

6.1.1 Assumptions

We are assuming that the signal from layer 1, flight computer subsystem reach on time to the parachute to deploy.Also The parachute ejection system is connect to the power supply from layer 1 that able the ejection to allow communication with it .

6.1.2 Responsibilities

A ballistic parachute is one that drops straight to the ground when deployed. The Ballistic parachutes are the most popular type of recovery technique since they are relatively simple to build and allow for very slow descent speeds this is the reason why we will use it for our project.

6.1.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Parachute Ejection Signal	input 5	Ejection of Parachute

7 Layer 4 (Communications) Subsystems

For this layer we are currently using Arduino IDE and visualize the data statistics manually. The data to the application will be fetching to this layer from Layer 1, Flight Computer Layer. The focus currently is to focus on achieving product goals. The communication layer can be extended to another project to build an application that would provide better visualization using data from Telemetry. The current version saves data in the memory card that can be used by the user for further analysis. Developing a ground system application can enable user to view the telemetry data while the rocket is in flight. A good example to follow for project extension is Mission Planner desktop application by Ardupilot.

7.1 Application

he application is going to be the only subsystem for the layer and is connected to two subsystems of layer 1; Flight Computer and the Telemetry. The visibility aspect of User Interface is key for Data Visualization. Therefore, the application is planned to have a window with speedometer on the left, that will show the speed of the rocket and window on the right, to show the status of the rocket launch stages; launch, apogee height, and recovery (ground) status. For now, we have decided to use launch pad for launch and calibrate manually.

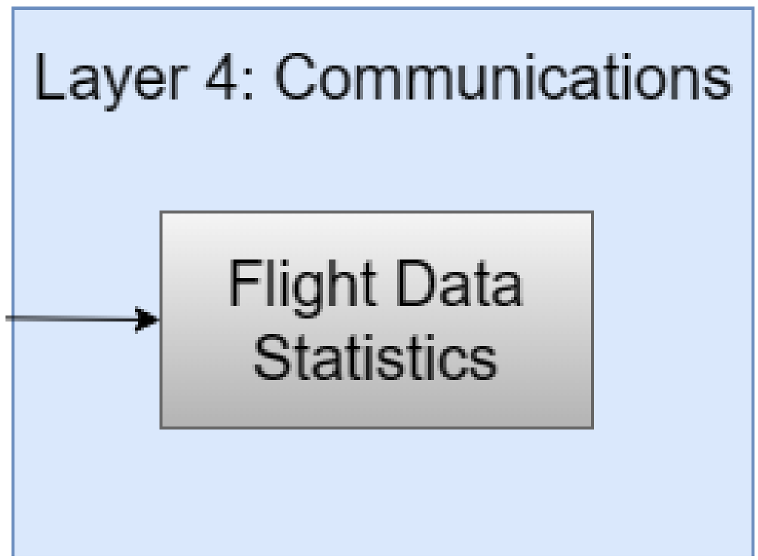


Figure 10: Example subsystem description diagram

7.1.1 Assumptions

We are assuming that we are going to enable the link between the application and flight computer layer through the data link 7 showed in the diagram above. We are still researching best method to enable the data link 7 to complete the layer tasks.

7.1.2 Responsibilities

Connect with the flight computer subsystem of layer 1 and receive the launch data through link 6. Provide the user data to be used for analysis. This could be extended to developing a software to view the flight statistics such as Mission Planner from ArduPilot.

7.1.3 Subsystem Interfaces

Currently we are using Arduino application to get the data and programmed the system to save data in memory card for further analysis. An application similar to Mission planner can be developed to include a window that visualize the flight data. Besides, it can be automated with a rocket launcher to provide better user experience for calibrating the system and launching the rocket.

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
01	Launch Data	input 6	Show Flight stats

REFERENCES