# CubeSat

---

# Design Document

Jacob Liberman - jliberman2016@my.fit.edu

**Faculty Advisor:**

Dr. Marius Silaghi   msilaghi@fit.edu

**Client:**

Kennedy Space Center

# Introduction

## Overview

This project is a joint collaboration between the Kennedy Space Center (KSC) and Florida Tech to develop a platform for astrobotany experiments at a smaller scale through use of a CubeSat.

## Purpose

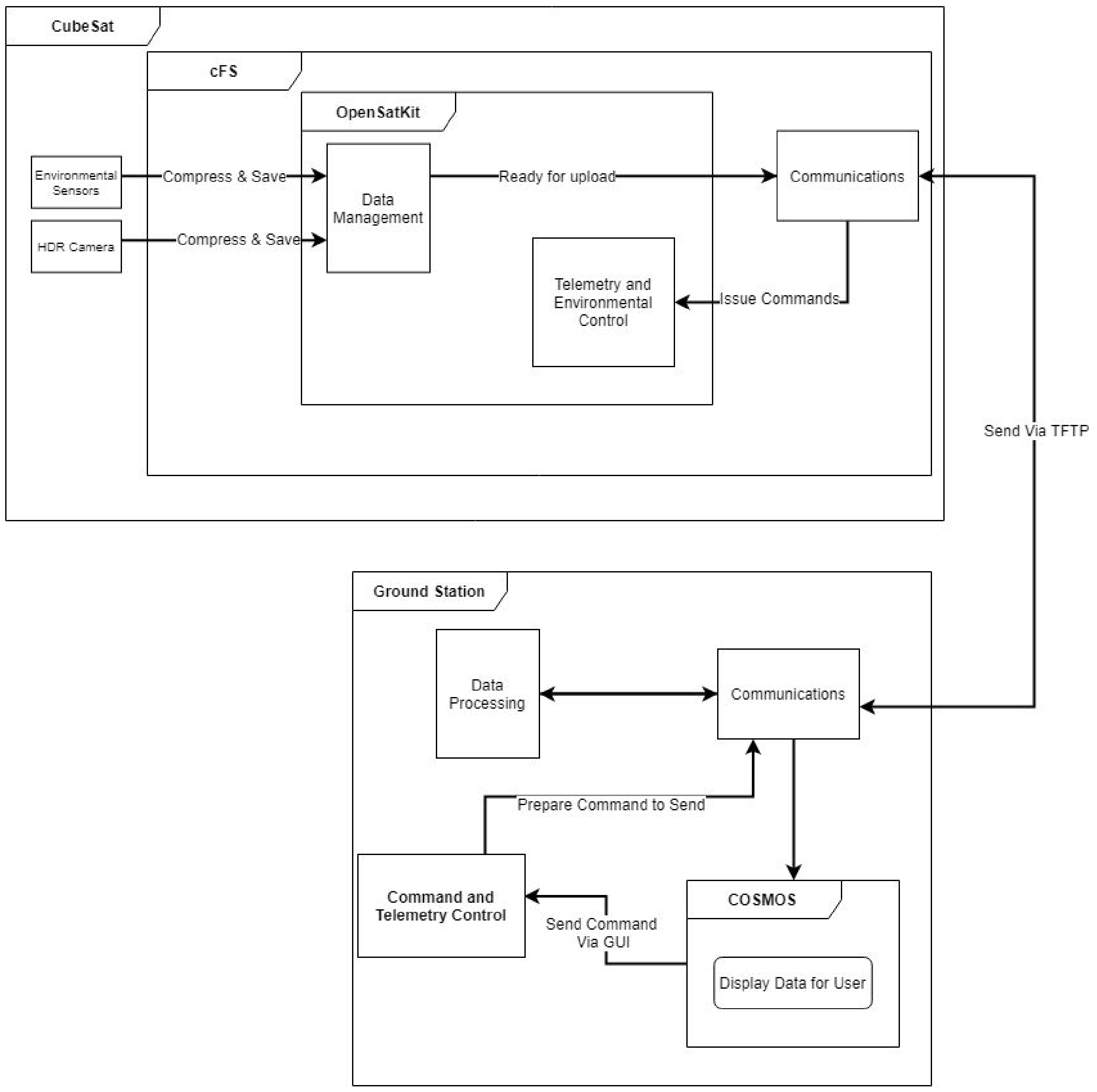
This document describes the implementation of both the ground system and CubeSat systems using a UML diagram, and pseudo code.

## Scope

As the entire project from design to launch will take approximately 3 years to complete, the scope of this year is to create a functioning ground station to receive data from our prospective CubeSat as well as other satellites already in Orbit.

## System Design

### UML Diagram

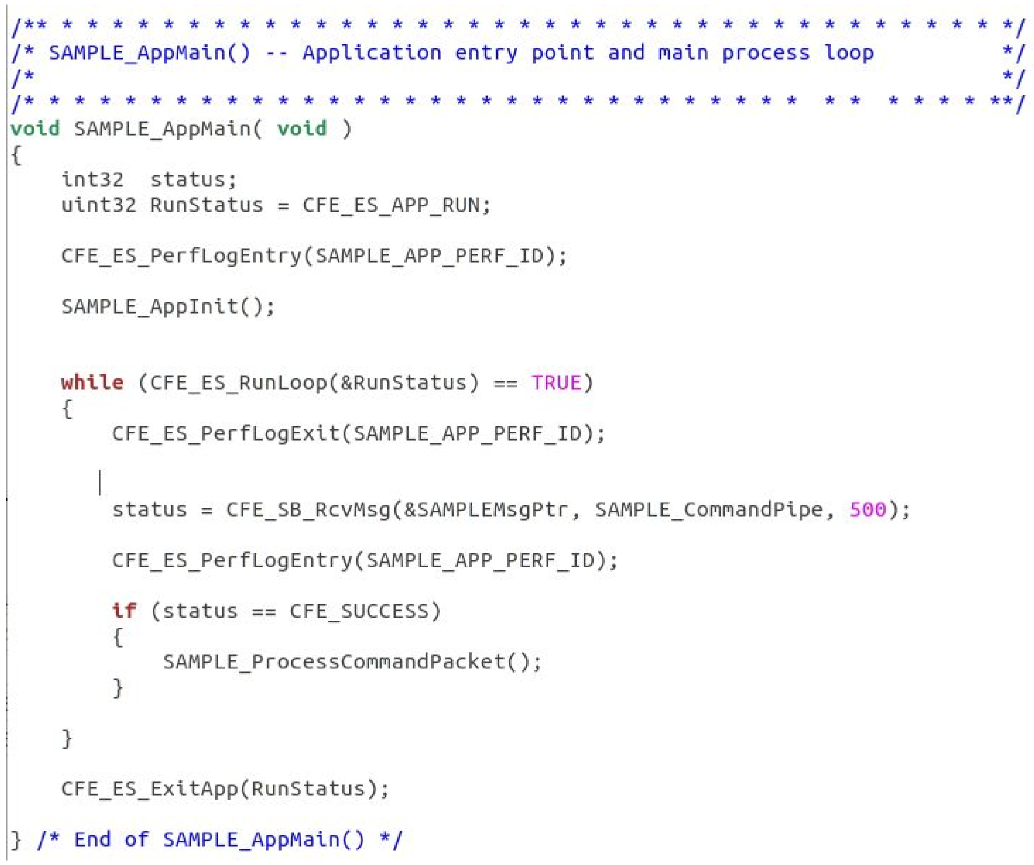


## Design Overview

The design of the project is split into two main systems, the CubeSat and the ground station. The CubeSat will be running NASA’s Core Flight System (cFS) with an OpenSatKit application that allows for greater customization of features and implementation of the design. The application will take in data from the environmental sensors and HDR camera and save only at intervals to conserve space, but it will continuously monitor them to check for sudden abnormalities in the system. This data is then prepared and sent by the cFS via the Trivial File Transfer Protocol (TFTP), which is designated by Space Communications Protocols, to the ground station. It is then processed by the ground station and the relevant information is displayed to the user through COSMOS, which is a user interface for the control of satellites and their embedded systems. Any commands then sent through COSMOS are prepared by the ground station and sent through TFTP.

# Application Design

## Example App Pseudocode



```
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* SAMPLE_AppMain() -- Application entry point and main process loop          */
/*                                                                            */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  * *  * * * * **/
void SAMPLE_AppMain( void )
{
    int32  status;
    uint32 RunStatus = CFE_ES_APP_RUN;

    CFE_ES_PerfLogEntry(SAMPLE_APP_PERF_ID);

    SAMPLE_AppInit();


    while (CFE_ES_RunLoop(&RunStatus) == TRUE)
    {
        CFE_ES_PerfLogExit(SAMPLE_APP_PERF_ID);


        status = CFE_SB_RcvMsg(&SAMPLEMsgPtr, SAMPLE_CommandPipe, 500);

        CFE_ES_PerfLogEntry(SAMPLE_APP_PERF_ID);

        if (status == CFE_SUCCESS)
        {
            SAMPLE_ProcessCommandPacket();
        }

    }

    CFE_ES_ExitApp(RunStatus);

} /* End of SAMPLE_AppMain() */
```

This is a sample of some of the processes needed to successfully run an application on the cFS to utilize some of the built in functionality of the system such as error checking, and easier data storage.