



## Design and Development of a 1U CubeSat

Aanchal Rakesh Haswani  
Mr. Sunny Kabrawala

Date of Submission: 20-10-2023

Date of Acceptance: 04-11-2023

The advent of CubeSats has ushered in a new era of space exploration, offering unprecedented opportunities for researchers, students, and organisations to engage in innovative, cost-effective space missions. CubeSats, often referred to as the "Lego bricks" of space, are miniature satellites that adhere to specific size, weight, and shape parameters. The 1U CubeSat, measuring a mere 10cm x 10cm and weighing around 1.33 kilograms, exemplifies this trend. These small yet highly capable spacecraft have opened doors to a wide array of applications, from scientific experiments to technology demonstrations and beyond.

In this research paper, we delve into the intricate world of CubeSat design and development, focusing specifically on the 1U CubeSat form factor.

We explore the fundamental components and considerations that underpin the creation of these tiny satellites. The 1U CubeSat serves as a microcosm of the broader CubeSat ecosystem, exemplifying the principles of miniaturisation and modularity.

The CubeSat standard was created by California Polytechnic State University, San Luis Obispo, and Stanford University's Space Systems Development Lab in 1999 to provide access to space for university students. The standard and dimension facilitate not only academic projects, making space more accessible, but also aid in scientific and commercial space missions.

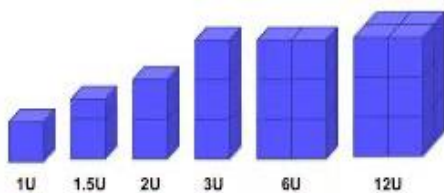


FIGURE: CubeSats of various dimensions

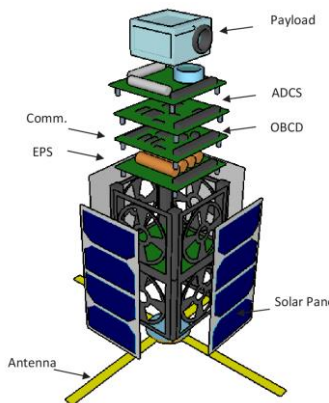


FIGURE: CubeSat: structure



FIGURE: an assembled 1U CubeSat

### Basic Working Principles of CubeSat

- **Power Supply:** The electrical power system (EPS) consists of solar panels and batteries. Solar panels hold solar cells that convert the solar light from the sun into electricity. Having deployable panels adds solar cell area but also extra mechanical complexity. For the panels to deploy, they need a burn wire release mechanism.

- **Communication System:** CubeSats use radio-communication systems. The satellite uses an antenna, usually deployed once in orbit, to help with communication. Antennas range from commercial measuring tapes to more complicated inflatable dish antennas. A burn wire release mechanism can be employed to deploy antennas too.



● **On-Board Computers OBCs:** This is the brain of the satellite, responsible for processing mission commands and coordinating all other subsystems, including the payload.

● **Attitude Determination and Control System:** The attitude determination and control system (ADCS) controls the orientation of the CubeSat concerning an inertial frame of reference and includes reaction wheels, magnetometers, thrusters, star trackers, sun and earth sensors, angular rate sensors, and GPS receivers and antennas.

● **Payload:** Depending on the kind of mission, the CubeSat's payload or purpose can vary. The examples range from testing a flight control system to collecting data for some sort of biological experiment.

● **Thermal Control:** To have the temperature of the CubeSat maintained to a set parameter and to detect and avoid temperature fluctuations, a thermal control system is required.

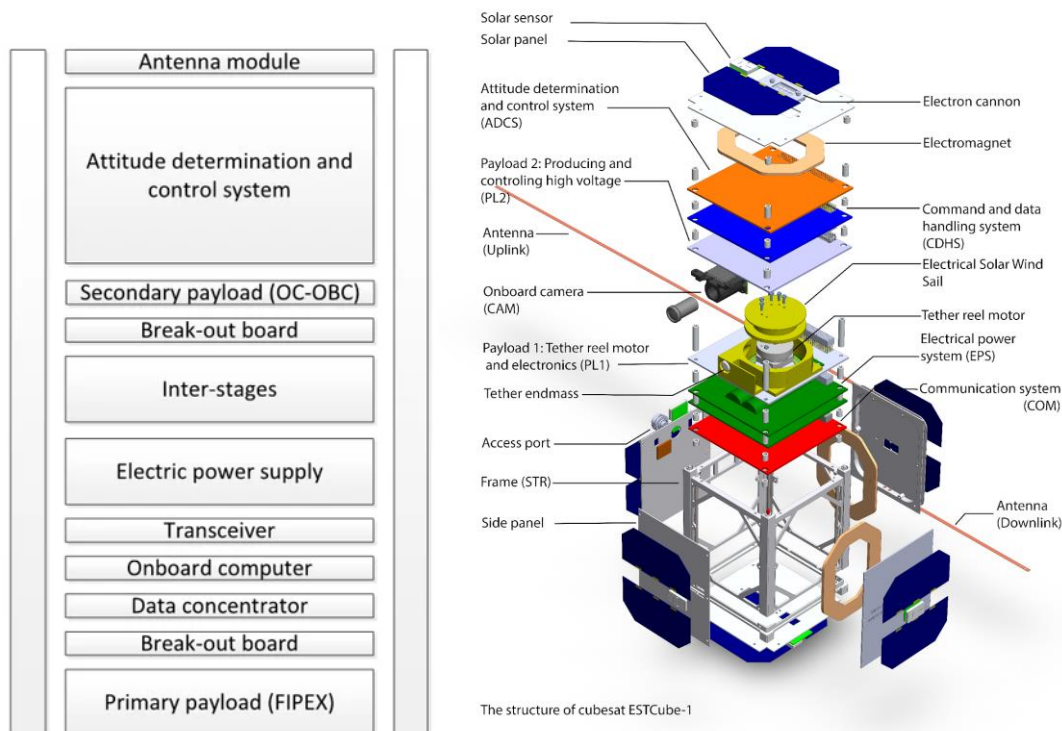


FIGURE: Structure of CubeSat ESTCube-1

### NIRVANA and Nebula Nest:

Given the growth and rapid advancement in technology, the possibility of humans living in space is only increasing. Though a controversy, space commercialization and space colonies have become a subject of interest to wealthy businessmen who are willing to readily invest in the area. Before a time like this comes, we need to test and confirm the effects of space, including microgravity and cosmic radiation, on the human body and know the parameters essential for survival in the area.

Therefore, we have come up with certain experiments to be carried out in space without having to send human beings for a prolonged period. Here, the concept of microfluidics comes into play. Lab-on-a-chip technology has been in practice for the last few decades and was originally used as a method of drug testing instead of its small size, which aided in obtaining better yields. With time, this technology evolved to be able to perform experiments on cells of various organs in the human body, giving birth to organs on a chip. Organ on a Chip is one of the many LoC and PoC applications.



Our organ on a chip is designed to control cell microenvironments and maintain tissue-specific functions. The concept can be further explained by the given depictions:

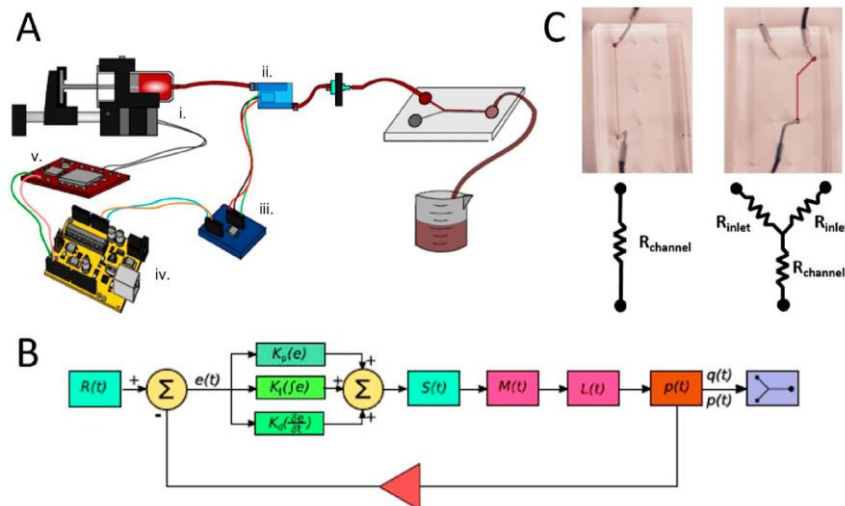
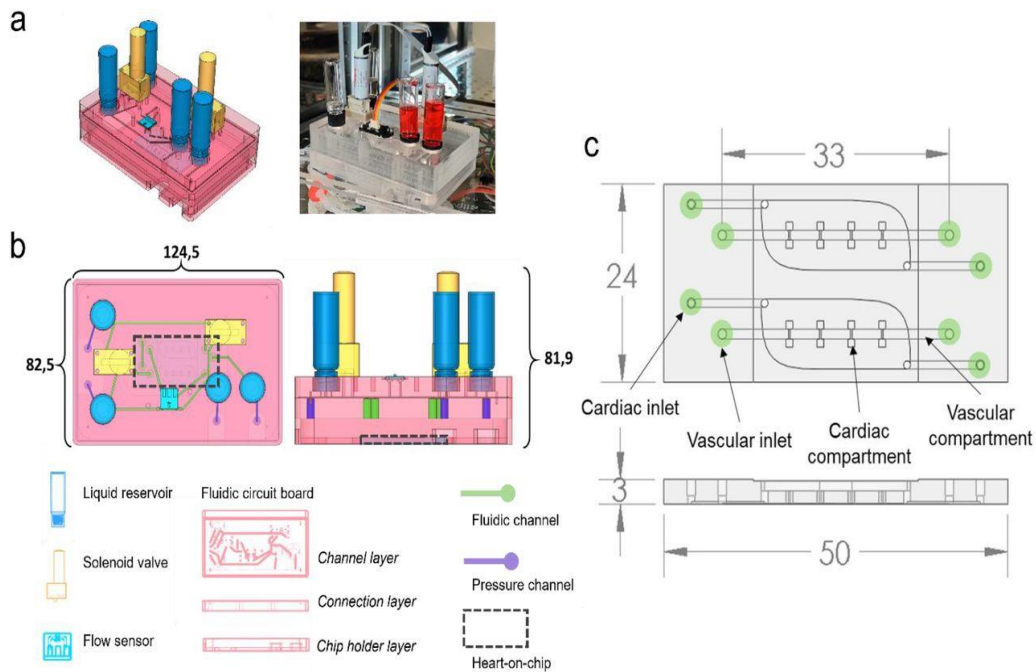


FIGURE: Arduino controls pressure in a microfluidic system to control the desired flow pattern.

### What Makes NIRVANA Different from Other CubeSats

The assembly mechanism, the antenna range, and the payload make Nirvana different from all other CubeSats. The assembly mechanism

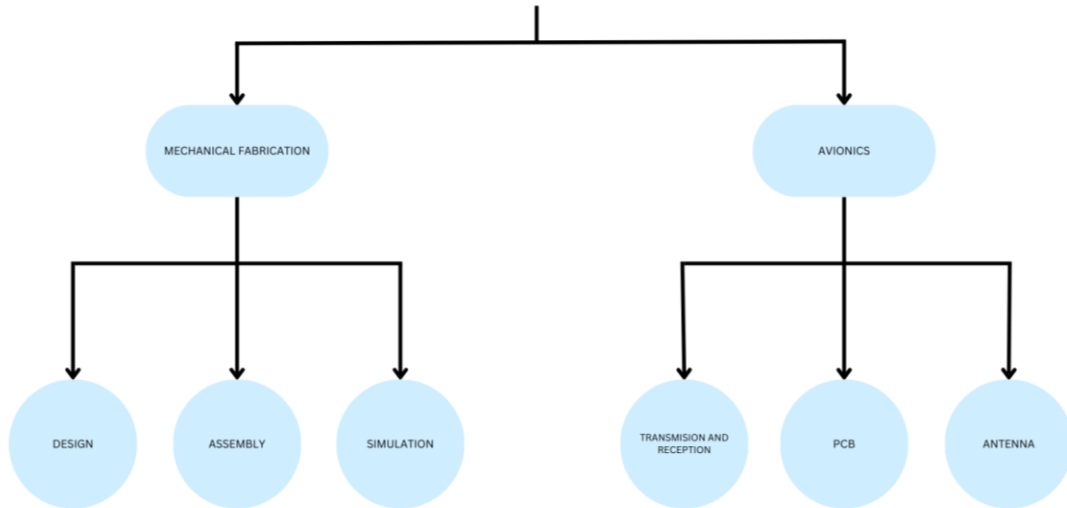
involves sliding walls and a lid. To get ready access to the PCB, only 4 screws from the top need to be unfastened; the walls can be slid up too. Alongside, the entire CubeSat contains only 8 screws in total; unscrewing those will dismantle our entire CubeSat.



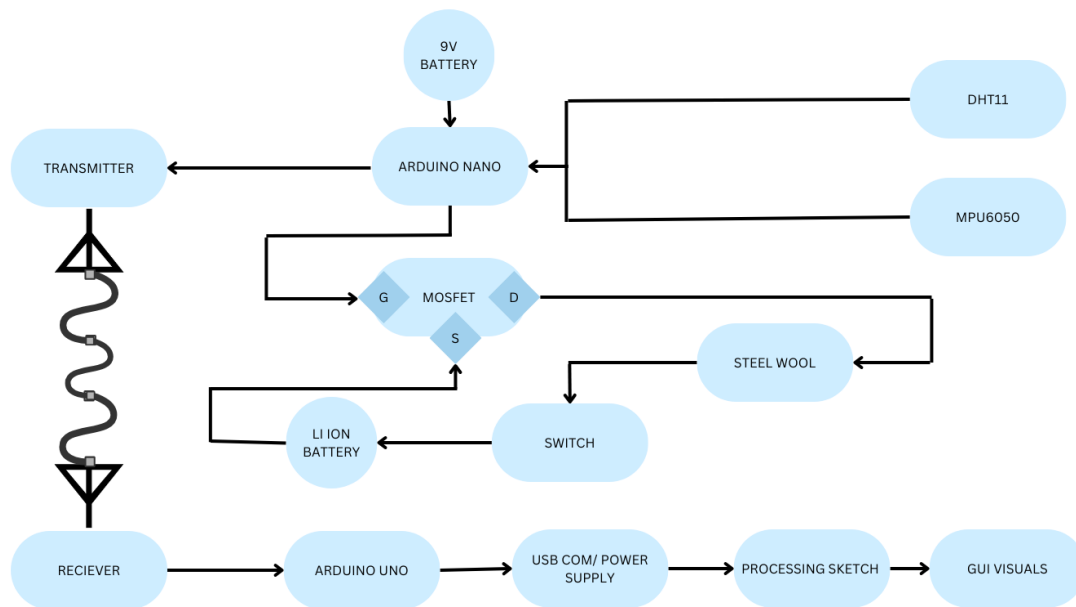
The CubeSat has a minimum range of 640 metres without using any amplifier, which is a milestone.

**Architecture:**

**ARCHITECTURE**






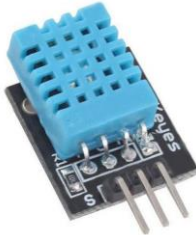

**Avionic Circuit Connections:**



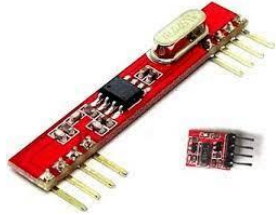
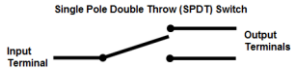





**Basic Components and Software Required:**

**List of Components:**

Component Name	Component Image	Functioning
Arduino Uno		Microcontroller connected to receiver on the ground station
Arduino Nano		The microcontroller connected to the transmitter in the satellite
MPU6050		3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor; I2C communication; 8-bit and 16-bit register access modes
DHT11		Temperature and humidity sensor, Operating Voltage: 3.5V to 5.5V, Operating current: 0.3mA, can measure from 0°C to 50°C and humidity from 20% to 90%
Mosfet		A three-terminal device having source(S), gate (G), and drain (D); controls the voltage and current flow between the source and drain terminals



434MHz Transmitter and Receiver		To transmit data from the CubeSat to the ground station
SPDT Switch		Single Pole Double Throw Switch; used to control two dissimilar circuits through a similar single input
9V Battery		
Li-ion Battery		
Screw Terminal		

#### List of Software and their Purpose

1. **GMAT**: Designing the orbit of the CubeSat
2. **EAGLE**: Designing the PCB of the CubeSat
3. **Arduino IDE**: for writing code, it connects to the Arduino hardware to upload programmes and communicate with them.
4. **Fusion 360**: Designing the Mechanical Structure of the CubeSat
5. **CST Studio Suite**: Estimating the Dimension of the Antenna for the CubeSat
6. **Ultimaker Cura**: For Slicing the Mechanical Design to Put into the 3D Printer

**Altium Designer was used as intermediate software to shift the designed PCB from EASY to EAGLE.**

#### Procedure:

##### UNDERSTANDING:

1. Introduction to satellites and their types, classification based on their size following the payload, and additional necessary devices.
2. A thorough understanding of orbital mechanics, understanding, and selection of orbits where the satellite is to be put in Understanding various terms like reference line, ascending node,

RAAN (angle between reference line and ascending node), AOP (angle between ascending node and perigee), etc., and experimenting with various orbitals using the GMAT software.

3. Researching the purpose of the CubeSat, and developing a further understanding of the payload, The payload can be divided into two types: mandatory and mission-specific payloads.

3.1. Mandatory payload: Payloads such as a spectrometer, CDH, communication system (transmitter and receiver), electrical power system, and ADCS are necessary for every CubeSat to be launched into space.

3.2. Mission Specific Payload: Payload such as earth observation payload, scientific instrument, technology demonstrator, educational payload, etc.

##### DESIGNING:

1. ELECTRONICS AND ELECTRICAL SYSTEM:

1.1. The power system for the CubeSat here is a 9-volt battery along with a lithium-ion battery connected to the MOSFET, which is further used to deploy the burn wire release mechanism for the unfolding of antennas.



**1.2.** Programme all sensors and components by coding and testing them by placing them on a temporary breadboard with a microcontroller, checking for different values.

**1.3.** Communication Systems: Transmission and Reception: The transmission from the antenna will take place through the onboard computers, a transmitter connected to a microcontroller (in this case, an Arduino Nano). The microcontroller extracts data from sensors and transmits it through an antenna. The ground system will use a receiver connected to an Arduino Uno microcontroller to receive the information transmitted. The components have been programmed as such; libraries like Adafruit, Virtual Wire and Tockn have been integrated to provide ease of access.

#### Transmitter Code:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <MPU6050_tockn.h>
#include <Wire.h>
#include <VirtualWire.h>
#define DHTPIN A2
#define DHTTYPE DHT11
const int led_pin = 13;
const int transmit_pin = 12;
int mosfet = 5;
DHT_Unified dht(DHTPIN, DHTTYPE);
MPU6050 mpu6050(Wire);
void setup() {
  Serial.begin(9600);
  Wire.begin();
  // Initialize device.
  dht.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets();
  vw_set_tx_pin(transmit_pin);
  vw_setup(6000); // Bits per sec
  pinMode(mosfet, OUTPUT);
  pinMode(led_pin, OUTPUT);
  delay(10000);
  digitalWrite(mosfet, HIGH);
  delay(1000);
  digitalWrite(mosfet, LOW);
}
void loop() {
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  mpu6050.update();
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));
```

```
} else {
  String datVal = String(int(event.temperature)) +
  "/" + String(int(mpu6050.getAngleX())) + "/" +
  String(int(mpu6050.getAngleY())) + "/" +
  String(int(mpu6050.getAngleZ())) + "/"; //+ "/1111";
  //datVal.trim();
  char msg[VW_MAX_PAYLOAD];
  datVal.toCharArray(msg,
  VW_MAX_PAYLOAD);
  // digitalWrite(led_pin, HIGH); // Flash a light to
  show transmitting
  vw_send((uint8_t *)msg,
  VW_MAX_PAYLOAD);
  vw_wait_tx(); // Wait until the whole message is
  gone
  // digitalWrite(led_pin, LOW);
  Serial.println("Data sent: " + datVal);
}
}
```

#### Receiver Code:

```
// receiver.pde
//
// Simple example of how to use VirtualWire to
// receive messages
// Implements a simplex (one-way) receiver with an
// Rx-B1 module
//
// See VirtualWire.h for detailed API docs
// Author: Mike McCauley (mikem@airspayce.com)
// Copyright (C) 2008 Mike McCauley
// $Id: receiver.pde,v 1.3 2009/03/30 00:07:24
mikem Exp $
#include <VirtualWire.h>
const int led_pin = 13;
const int receive_pin = 11;
void setup()
{
  delay(1000);
  Serial.begin(250000); // Debugging only
  //Serial.println("setup");
  // Initialise the IO and ISR
  vw_set_rx_pin(receive_pin);
  vw_setup(6000); // Bits per sec
  vw_rx_start(); // Start the receiver PLL running
  pinMode(led_pin, OUTPUT);
}
void loop()
{
  char buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
  if (vw_get_message(buf, &buflen) // Non-
  blocking
  {
```



```
String datVal = String(buf);  
Serial.println(datVal);  
}  
}
```

**1.4. On-board Board Computers:** The sensors, connected to a microcontroller, used for mission Nirvana comprise the MPU6050, which is a gyroscope and measures the orientation of the CubeSat, and the DHT11, which is a temperature and humidity sensor functioning to maintain the temperature of the satellite. A mosfet is used to deploy the antenna.

**1.5. GUI (Graphical User Interface):** A graphical interface has been integrated to obtain the orientation (roll, pitch, and yaw) of the CubeSat in space.

**GUI CODE:**

```
import processing.serial.*;  
import java.awt.event.KeyEvent;  
import java.io.IOException;  
Serial myPort;  
String data="";  
int roll, pitch, yaw, temp;  
String startbit, endbit;  
void setup() {  
  size(1924, 1055, P3D);  
  myPort = new Serial(this, "COM3", 250000); //  
  starts the serial communicationz  
  myPort.bufferUntil('\n');  
}  
void draw() {  
  translate(200, 100, 0);  
  background(255, 255, 255);  
  textSize(30);  
  fill(250, 0, 0);  
  text("Pitch: " + pitch + " Roll: " + roll + " Yaw:  
" + yaw + " Temp: "+temp+" °C", 120, 100, 265);  
  translate(750, 400, 0);  
  // Rotate the object  
  rotateX(radians(-pitch));  
  rotateZ(radians(roll));  
  rotateY(radians(yaw));  
  // 3D Object
```

```
  textSize(30);  
  fill(0, 0, 255);  
  box(400, 400, 400); // Draw box  
  textSize(50);  
  fill(255, 165, 0);  
  text("CubeSat", -90, 25, 210);  
  
  fill(0, 0, 0);  
  translate(-310, -190, 0);  
  box(340, 5, 45);  
  translate(610, 0, 0);  
  box(340, 5, 45);  
  //delay(10);  
  //println("ypr:\t" + angleX + "\t" + angleY); // Print  
  the values to check whether we are getting proper  
  values  
}  
// Read data from the Serial Port  
void serialEvent (Serial myPort) {  
  try {  
    // reads the data from the Serial Port up to the  
    character '.' and puts it into the String variable "data".  
    data = myPort.readStringUntil('\n');  
    // if you got any bytes other than the linefeed:  
    if (data != null) {  
      data = trim(data);  
      // split the string at "/"  
      String items[] = split(data, '/');  
      if (items.length > 1) {  
        //--- Roll,Pitch in degrees  
        // startbit = items[0];  
        temp = int(items[0]);  
        pitch = int(items[1]);  
        roll= int(items[2]);  
        yaw = int(items[3]);  
        // endbit = items[4];  
      }  
    }  
  }  
  catch(RuntimeException e) {  
    e.printStackTrace();  
  }  
}
```

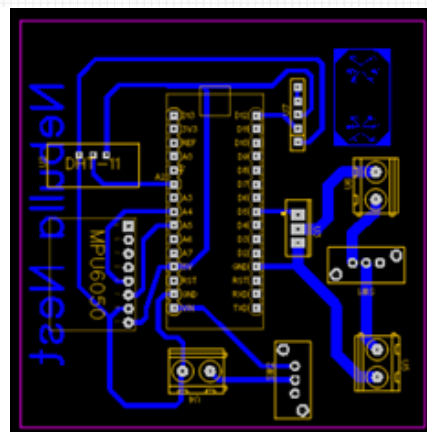
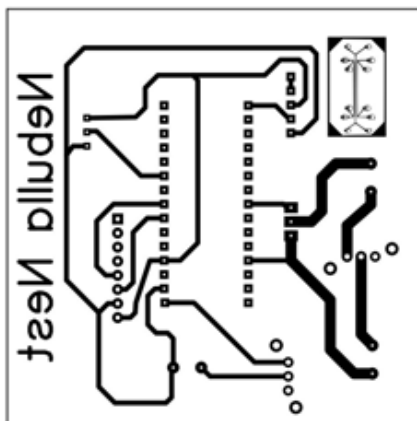
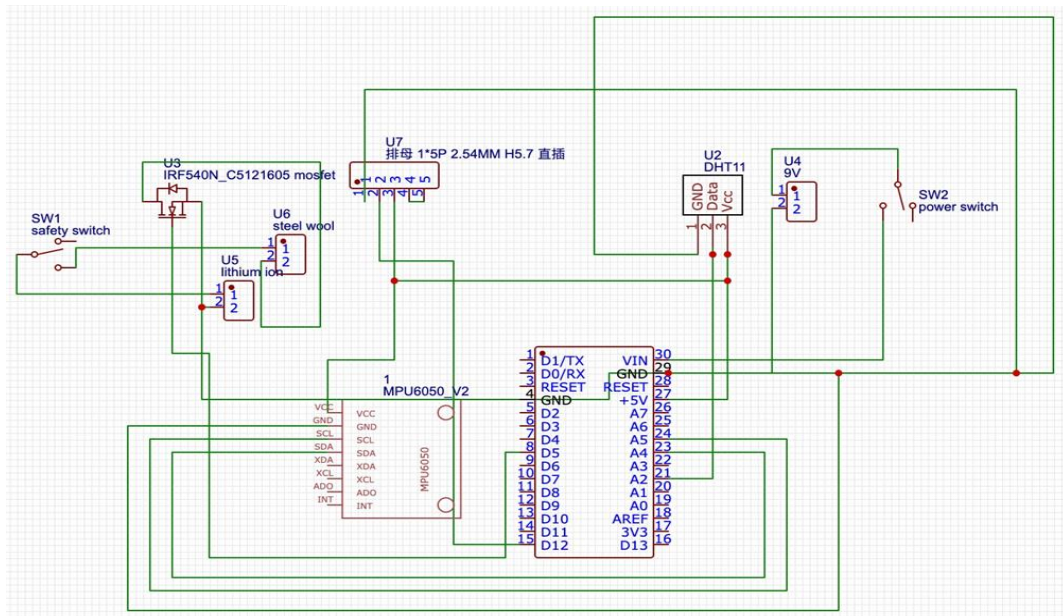
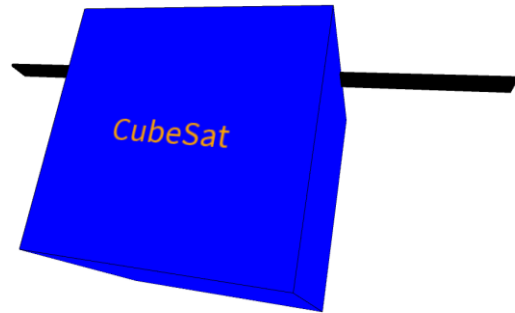


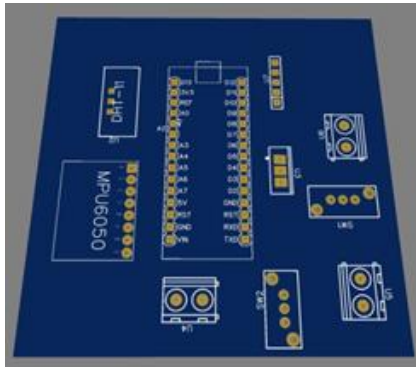


}

Pitch: -14 Roll: 10 Yaw: -16 Temp: 29 °C

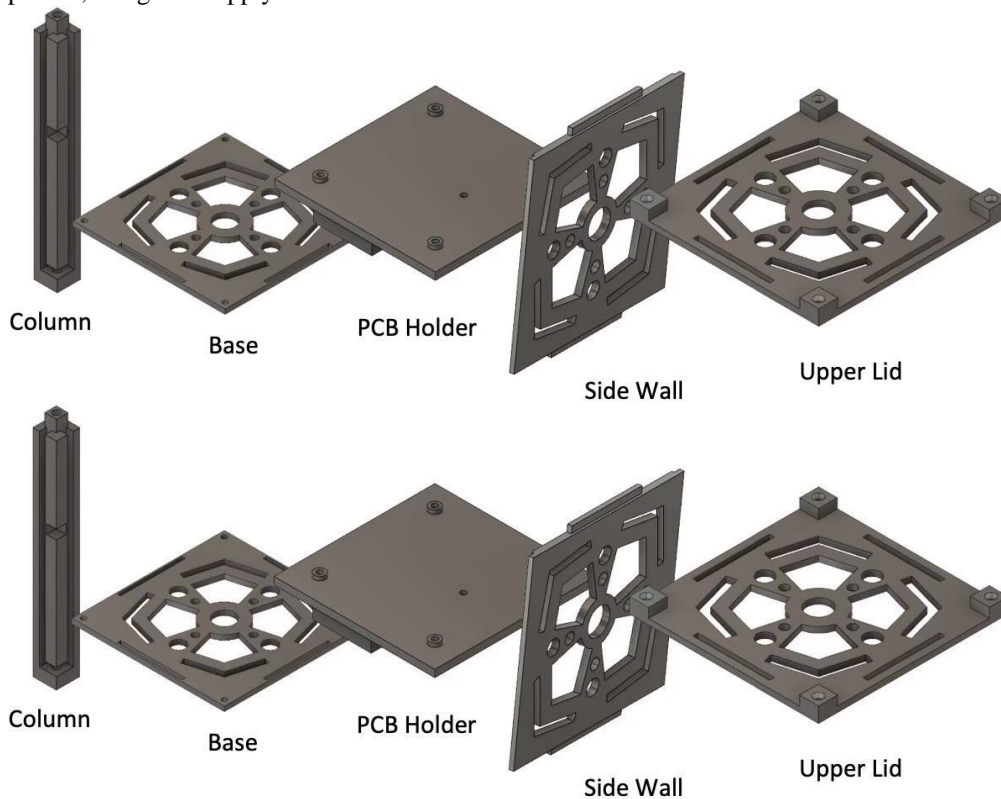
**1.6. PCB (Printed Circuit Board):** The circuit has been made compact to occupy all components while covering as little area as possible; it comprises all the electrical components mentioned above. The schematics and CAD models of the printed circuit board are-



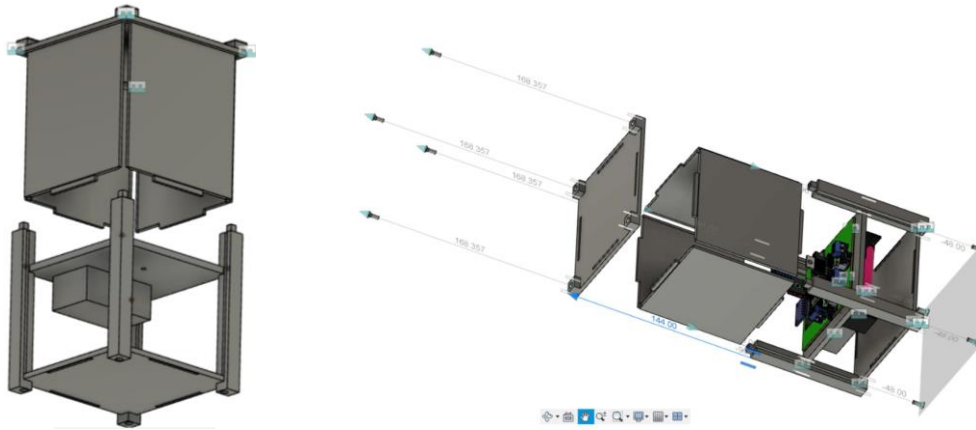


### 1. MECHANICAL FABRICATION:

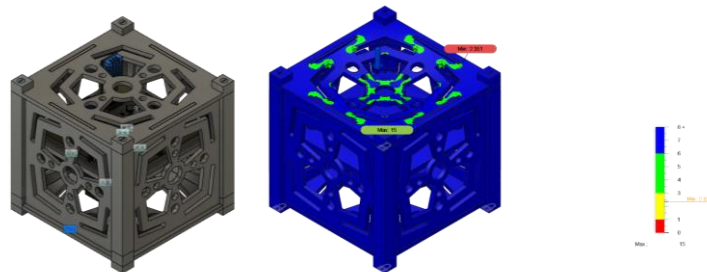
Designing CAD models is the first step in the process of mechanical fabrication. Formulate the dimensions of each component, design and apply constraints and force to check the maximum the satellite can withstand.



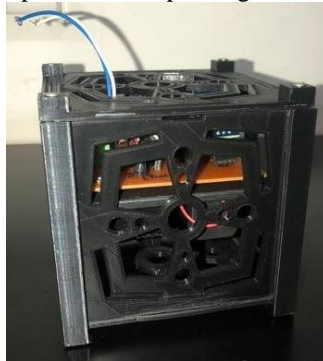
Simulate the designed components and pay particular attention to areas possessing holes and edges. Assemble the components after the result of above mentioned process is perfect and precise. Create a motion study to get a brief idea about the motion; check for any interference.



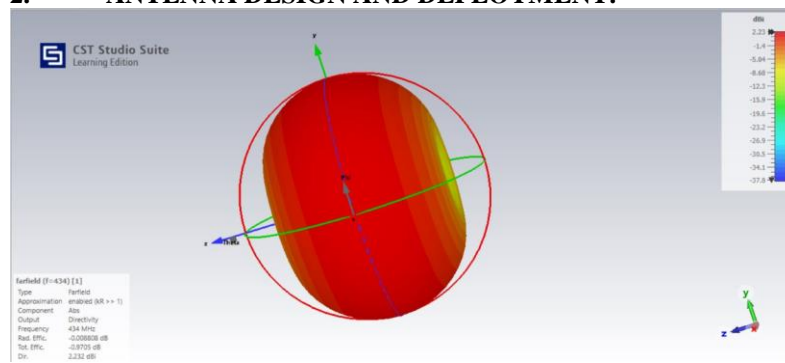
Next, in accordance with the PCB components, design the walls, the base and the lid. The following is the design and simulation for our mission;



Through the software Ultimaker Cura, print the components using a 3D printer; leave additional space for marginal error in printing. Sand the components after printing and fix them in place with the help of screws.



## 2. ANTENNA DESIGN AND DEPLOYMENT:



Compute the length of the antenna with the given values and using the software CST Studio Suite simulate the signal waves.



## CALCULATION AND DATA

<b>Frequency</b>	433/434 MHz
<b>C (speed of light)</b>	299792458 m/s
<b>Wavelength</b> =speed of light/frequency	690.7660323 mm
<b>Half Dipole</b> =speed of light/frequency	345.3830161 mm
<b>Length of Wire</b> =Wavelength/4	172.6915081 mm
<b>Radius of Wire</b> =Wavelength/1000	0.6907660323 mm
<b>Gap Length</b> =Wavelength/400	1.726915081 mm
<b>Range</b>	640 m
<b>Directivity</b>	1.67

### Observations:

1. The CubeSat development involves both theoretical and technical know-how.
2. The simulated data is almost accurate.
3. The printed components are made of PLA (polylactic acid), giving the satellite the ability to stay in orbit for about 8–10 minutes.

### Conclusion:

The successful testing of the CubeSat prototype denotes a significant development milestone and verifies that it is prepared for launch. The extensive testing and assessment procedures demonstrate that the CubeSat is fully prepared to function as intended once it is launched into orbit. Extensive performance standards have been tested, ensuring that the CubeSat is not only functional but also reliable in the harsh conditions of space.

The engineering and design of the CubeSat are remarkable since they made it possible for it to spend eight minutes in orbit. It will carry out important tasks and collect information throughout this period, signifying an important phase in its purpose. The stable orbit of the CubeSat guarantees that it can achieve its objectives for the duration of its mission.

The CubeSat may now confidently start its mission because of the testing's success and its demonstrated capacity to stay in orbit for eight minutes. The process of launching and deploying it

could produce significant data and insights that advance space exploration and scientific research.

### Works Cited:

- [1]. Best University in India | BIHER (to-Be-Deemed University),  
<https://www.bharathuniv.ac.in/downloads/aero/DESIGN%20AND%20FABRICATION%20OF%20IU%20CUBESAT.pdf>. Accessed 28 Oct. 2023.
- [2]. Chris, et al. "Arduino and MPU6050 Accelerometer and Gyroscope Tutorial." *How To Mechatronics*, 18 Feb. 2022, <https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>.
- [3]. Coutinho, Diego Anestor, et al. "Design, Prototyping and Stratospheric Launch of CubeSats for University Competition." *Journal of Aerospace Technology and Management, Departamento de Ciência e Tecnologia Aeroespacial*, 17 Apr. 2023, [www.scielo.br/j/jatm/a/sVfwMVDqphW8j5kqdVJWGzF/](http://www.scielo.br/j/jatm/a/sVfwMVDqphW8j5kqdVJWGzF/).
- [4]. NASA, [www.nasa.gov/wp-content/uploads/2017/03/nasa\\_csl\\_i\\_cubesat\\_101\\_508.pdf](http://www.nasa.gov/wp-content/uploads/2017/03/nasa_csl_i_cubesat_101_508.pdf). Accessed 28 Oct. 2023.