Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 7, Issue 1; January-March, 2020 pp. 14-20

http://www.krishisanskriti.org/Publication.html

CUBESAT
A functional cube satellite designed, programmed, and built by high school students for data collection and transmission in low sun synchronous orbit.

Krrish Jain and Shridhar Sinha

Introduction

A CubeSat is a is a type of miniaturized satellite for space research that is made up of multiples of 10 cm × 10 cm × 10 cm cubic units. Due to our interest in space exploration and robotics we had the idea of building our own CubeSat towards the end of 2017. Following a lot of initial designs, mission ideas, research and prototypes we have decided on a 1U CubeSat designed for data collection, transmission and Flight Tracking with ADS-B.

The satellite is planned to be launched as a secondary payload in a sun synchronous orbit around 650 kilometers above the surface of the Earth.

Mission Overview

Automatic Dependent Surveillance – Broadcast or ADS-B is a recent development in technology which allows an aircraft to transmit its instantaneous position through the navigation system, to be received by Air traffic Control. A lot of commercial flights have already made this system mandatory as it is incredibly important for safety and accurate navigation and would help prevent further situations such as the case of Malaysia Airlines 370.

The CubeSat will be equipped to gather transmitted data from ADS-B systems of aircraft and effectively tabulating flight history data in an Excel sheet that will be sent back to us. This comprises the main mission and the bulk of the payload.

The CubeSat also has a secondary mission, to establish communication with earth and transmit Gyroscope, temperature, and magnetic field data along with other mission data concerning power usage, timings and cycles to give a more complete picture of the mission.

The collected data will be used with a plethora of different representations to reconstruct the findings of the mission and to be made available to the public on a project website. We will be using scientific datain comparison with similar datasets to further our own understanding of atmospheric phenomena and especially magnetic fluctuations.

Sub Systems

1. Structure

The structure will meet the specifications mentioned in the cubesat design specification for a 1U cubesat. The structure will be made by an Indian manufacturer using Aluminium 6061 for the main body and anodized rails. The maximum mass of entire satellite will be under 1.33 kg.

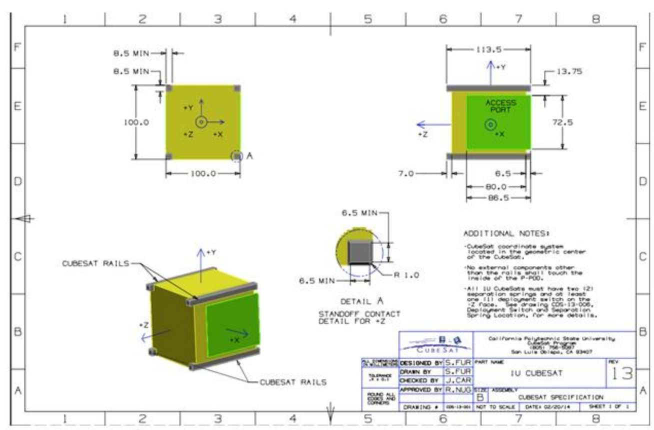

Figure 1. CubeSat structure specifications

2. Communications and Data Handling (CnDH)

The Raspberry pi 3 is the onboard computer of choice. Based on our research it can survive in space with some shielding. It is a powerful compact Linux computer with a 1.25Ghz processer, 1 GB RAM and expandable memory via and SD card slot. It is also compatible with other sensors and can connect via i^2c, serial bus and has GPIOs. It runs on 5V just like the other sensors and due its enormous community, there are a large number of libraries available which will make debugging and fixing software issues easier, making it an ideal choice.

We have selected the MPU6050 as the inertial measurement unit(IMU) because of its great capability and size. It contains a 3-axis Accelerometer and 3-axis Gyroscope which allows us to calculate velocity, acceleration and orientation. It comes space ready and is capable of operating in the temperature range of -40C to +85C. It also consumes low current of <5mA, operates at 3.3V-5V and can communicate via i^2c making it easy to use and efficient.

The DS18B20 temperature sensor in the TO-92 package is used because of its large temperature measuring range of -55C to +125C, it's one-wire protocol and low current draw of 1-1.5mA and 3-5.5V operating voltage. One-wire protocol allows multiple DS18B20s to be used on one-wire as each of them has a unique serial code. We are using two of these, one on the inside and one on the outside of the CubeSat.

The NEO-6M GPS module is used because it is an easy to use plug and play module which communicates via the serial bus or USB. It connects to GPS satellites and obtains latitude, longitude, ground speed, Coordinated universal time(UTC), etc. According to the Cubesat specifications, the satellite must be turned off during launch. The pi does not have an onboard real time clock and so the ability to obtain UTC from this module is crucial.

The raspberry pi was chosen over an Arduino because it allows us to run simultaneous tasks. We cannot change/add/remove Arduino scripts but having access to the command line allows us to do this remotely on a pi. Furthermore, the Arduino does not have the capabilities to implement the SDR.

Figure2.1 Raspberry Pi 3

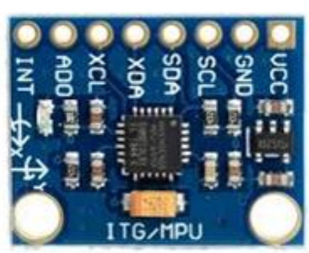

Figure2.2. MPU6050 IMU

Figure2.3 DS18B20 Temperature Sensor

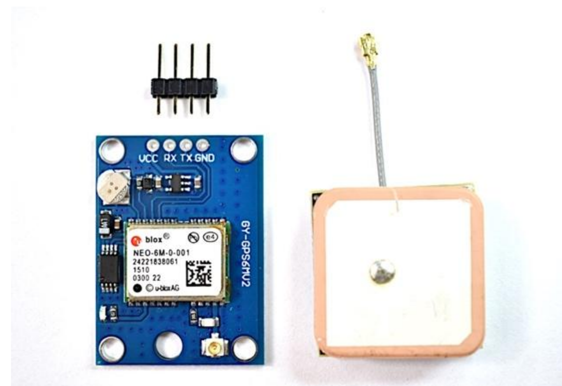

Figure2.4 NEO 6M GPS Module

3. Electrical and Power Systems (EPS)

We have chosen to use Li-ion Batteries because of its robust nature and high energy density. We need only 2 cells in series because of the nominal voltage of 3.7V(max of 4.2V) giving us a combined voltage of 7.4V(8.4V max), therefore there is very little power loss when stepping the voltage down to 5V. It also allows us to keep more cells in parallel to increase battery capacity. We will use solar panels to charge the batteries.

We will use the INA219 power management board to monitor voltage of the batteries, charging current and discharging current. The Raspberry Pi will be connect to MOSFETS which can switch certain sensors on and off to conserve power.

We have made a power budget in order to calculate the capacity of batteries and the solar panels required.

Table 1. Power Budget

	V(V)	I(A)	Peak Power (W)	Standby Power (W)	Duty Cycle	Power Required (W)
GPS	3.3	0.055	0.1815	0.18	1	0.1815
ADS-B	5	0.3	1.5	0	0.1052	0.1579
DS18B20	5	0.0015	0.0075	0.0075	1	0.0075

MPU6050	5	0.002	0.01	0.01	1	0.01
Magnetorquer	5		2	0.6	0.7	1.02
Comms	3.3	1.2	3.96	3.96	0.1052	0.4168
Raspberry Pi	5	0.5	2.5	1	0.1052	1.1578
Total						2.9516

We used the voltage and current requirements for different components to calculate the power they will use at peak and on standby. Since sensors will periodically turn on at some points during the orbit, we calculated the duty cycle and therefore the average power required by sensor per orbit.

We assumed an orbital period of 95 minutes.

Sample calculations are for the Magnetorquers.

$$Duty\ cycle = \frac{time\ used\ in\ orbit}{total\ time\ orbit}$$

$$= \frac{67}{95} = 0.7$$

$$Power\ required = Duty\ cycle \times Peak\ Power + (1 - Duty\ Cycle) \times standby\ power$$

$$= 0.7 \times 2 + 0.3 \times 0.6 = 1.02W$$

4. Attitude Determination and Control Systems (ADCS)

In order to maintain stability we decided to use magnetorquers. Magnetorquers are electromagnetic coils that interact with the Earth’s magnetic field to stabilize and detumble a satellite. They are passive which means they do not have moving parts, have a relatively simple algorithm to make them work and require relatively low energy.

5. Communication

For communications we have used an Xbee S2C Serial radio transceiver. It is only used during prototyping and the parachute test because of its relatively large range of about 1km and easy implementation via the serial protocol.

Figure 3. Xbee S2C radio transceiver

6. Payload

The main component of the mission is to track airplanes using ADS-B, we will be doing this with a software defined radio(SDR). We are using an antenna tuned for 1090MHz radio waves, which is the frequency at which aircraft transmit. We will use open source code from the dump1090 file which reads all the signals and outputs the flight ID, speed, location, and altitude and writes to an excel sheet. We have written python scripts to sort through the data which we will execute once the ground station receives the data.

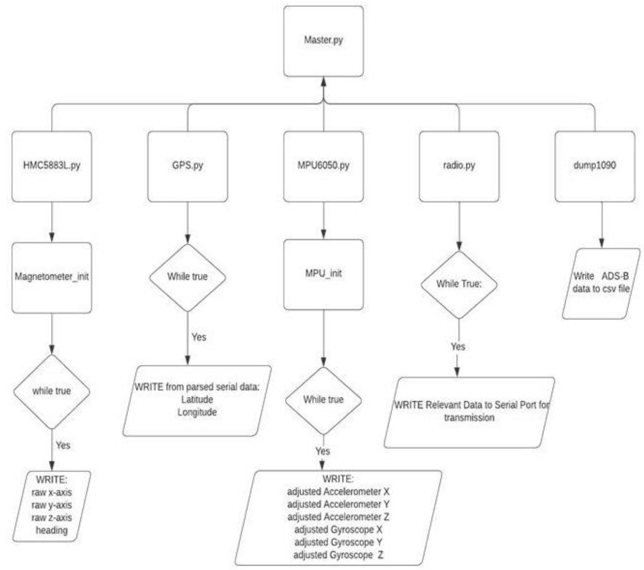

Figure 4. Systems Coding Hierarchy

Code

A master file automatically controls the execution of the equipment as per the current cycle of the satellite. The code for the individual sensors initializes the component and writes the collected data to a CSV file. The ADS-B uses the open source ‘dump1090’ to write ADS-B data to CSV file. These CSV files are temporarily stored and operated on by a few formatting codes for better data representation and then transmitted using the radio.

The satellite will only transmit raw data and the ground station will handle the compiling and formatting the data, while flagging incomplete or unexpected data. This allows us to get maximum information from the data and not stress the on board computer with computational tasks.

Mass Budget

Table 2. Mass Budget

Sub System	**Mass(g)**
Structures	200
CNDH	
Raspberry Pi 3	50
HMC5883L	1
MPU6050	2
NEO6M	17.6
DS18B20	1
EPS	
Power Management Board(including Batteries)	200

Solar Panels	52
ADCS	196
Comms	
Radio	25
Antenna	30
Payload	25
Total	**799.6**

The total mass of around 800g is under the maximum of 1.3kg in the CubeSat specification for a 1U cubesat.

Prototypes

V1

Starting April of 2018, we started testing different systems and sensors for the mission. We also built a few structures including a 3D printed one(Fig 5.4) and one made from aluminium(Fig 5.5). These tests were done with Arduinos for their convenience. These were still early stages and we were trying to figure out how to get communications to work and so we designed our own communication and transmission protocol(Fig 5.1 and 5.6). At that stage we were having a speaker output the data with sound signals and manually decoding them(Fig 5.2). Later we wrote an algorithm to store this data onto an SD card.

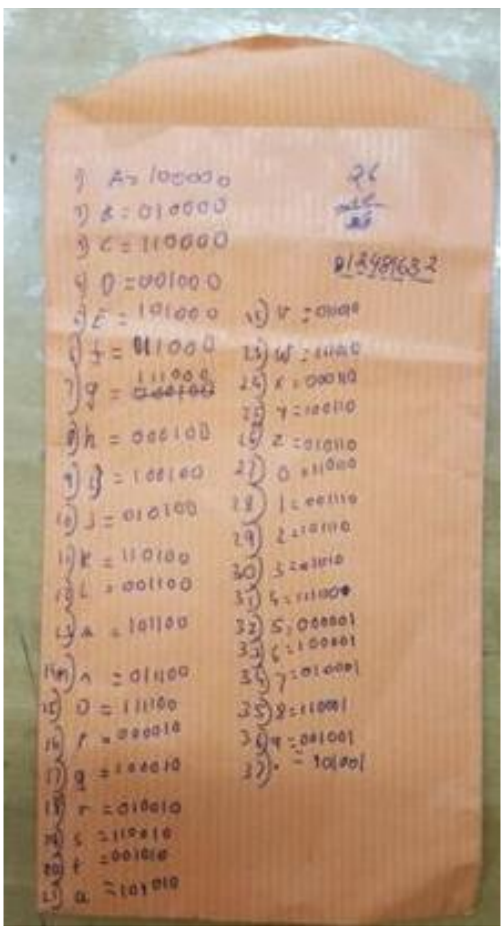

Figure 5.1 Transmission Protocol Key

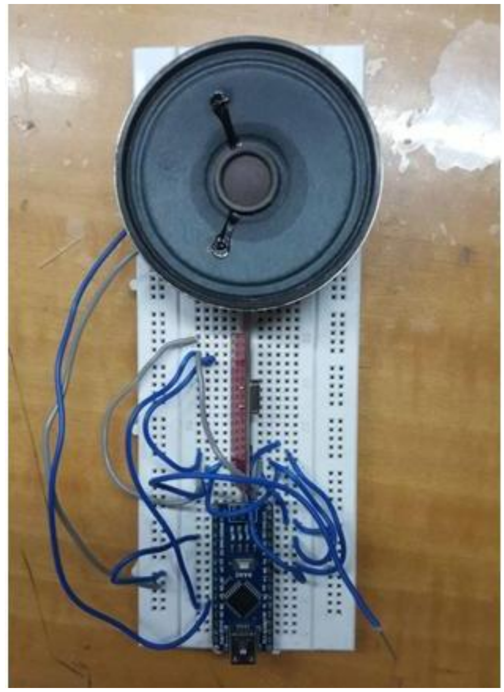

Figure 5.2 Ground Station

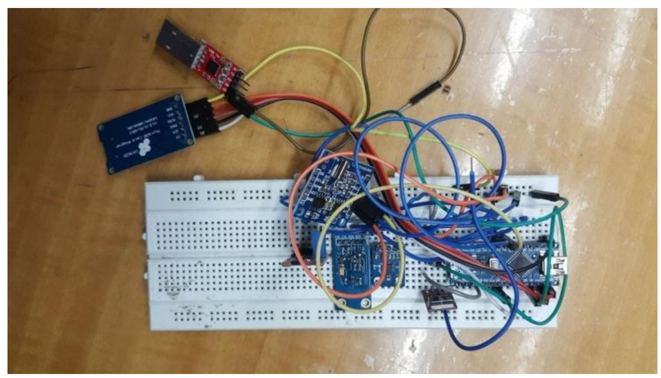

Figure 5.3. Breadboard Satellite

Figure 5.4 3D printed CubeSat structure

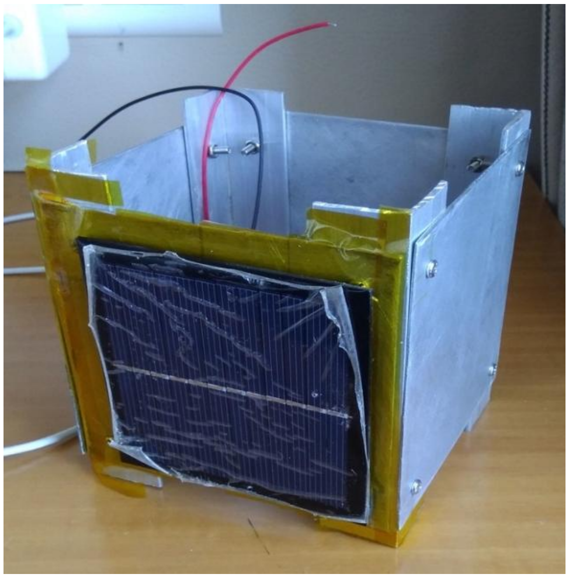

Figure 5.5 Aluminum CubeSat structure

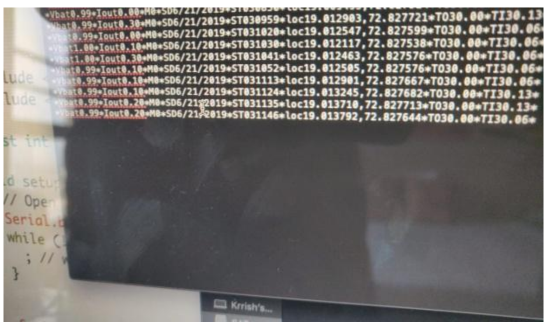

Figure 5.6 Communication Protocol

Figure 5.6 depicts the packets that were transmitted from the satellite to the ground station. Vbat refers to the battery voltage followed by its value, Iout is the current being drawn from the batter, and so on.

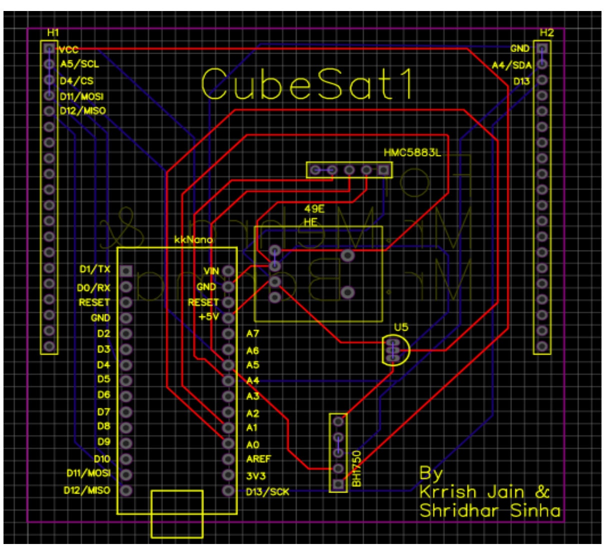

Figure 5.7 PCB. For V1

We designed a PCB for the V1 in such a way to make it stackable. Figure 5.7 is the CNDH board. This board was never printed however, because the platform did not meet our requirements.

V1.1

In order to increase onboard storage and processing power, we shifted from an Arduino nano to an Arduino Mega at the start of 2019. Instead of designing a PCB directly, we decided to solder and make a prototype from perf board. This was also designed to be stackable, with different layers for different sub-systems. The CNDH board is in Figure 6.1 and 6.2. We continued to test different sensors and alternatives. Being a new board, it look time to adjust to the Mega. There were still a lot of problems with software and the large number of variables was bottlenecking the processor. We spent a large portion of the time making the code more efficient.

Figure 6.1 Perf Board prototype

Figure 6.2 Perf Board prototype

Parachute test

In of June 2019, we decided to test our prototype electronics and software in the air. Originally planned to be a glider sat, we were unable to deploy a glider from a high enough height.

So we attached a homemade parachute to a box housing the electronics and set up a ground station to receive the transmitted radio data. The test was a success and the data received was operational and accurate.

We designed it to be a single shield on an Arduino Mega. This platform tested everything except the ADS-B, that is radio transmission, electrical data, GPS, IMU, tempreature and data logging, onto an SD card.

Initally, we designed the PCB in Fig 7.1 but due to size limitations, it was redesigned to Fig 7.2. Fig 7.3. 7.4 and 7.5 depict final printed versions.

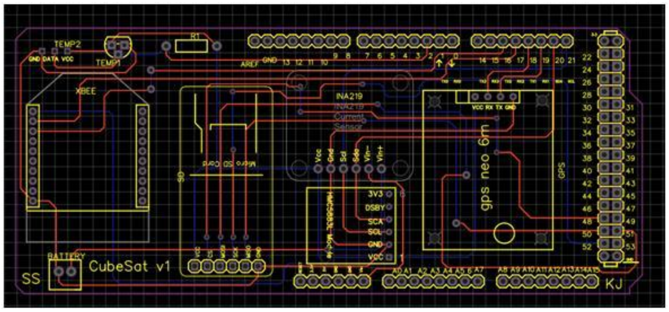

Figure 7.1 Initial PCB design for the glider sat

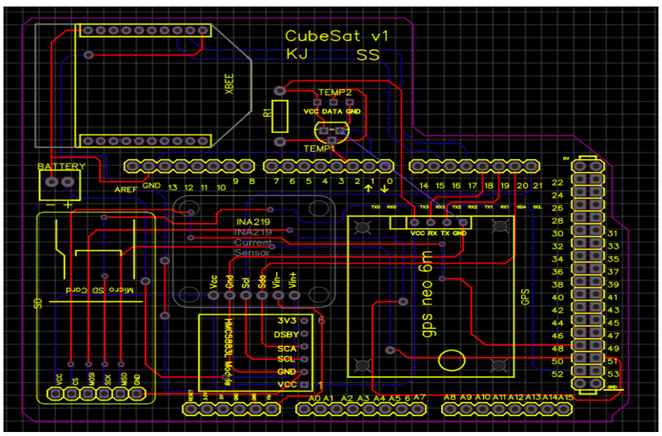

Figure 7.2 Final PCB design for the glider sat

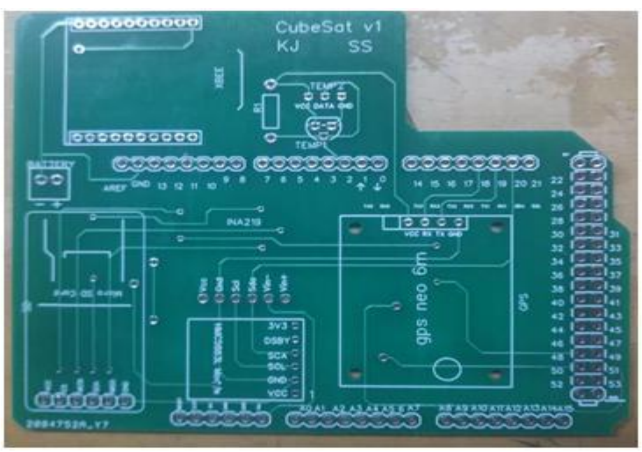

Figure 7.3 Final PCB for the glider sat(Front)

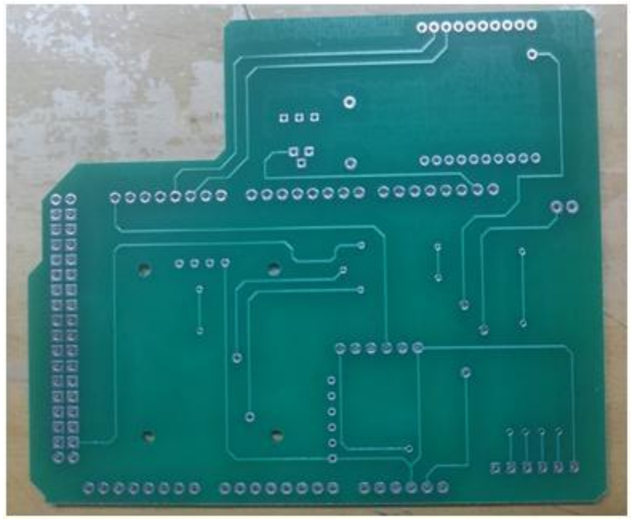

Figure 7.4 Final PCB for the glider sat(Back)

Figure 7.5 Final glider sat

Figure 7.6 Final glider sat in its box with parachute

V2

This is the final prototype. We successfully tested all sensors with the Raspberry Pi and have setup a code flow structure as in Fig 4. This was designed to have a master program running at all times and direct access to the terminal. This would allow us to be as flexible as possible and even change or add code. These are major upgrades from the Arduino. Fig 8.1 depicts a google drive with a backup of all the code.

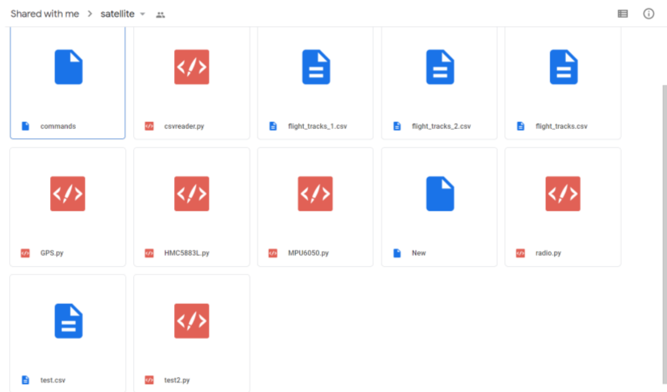

Figure 8.1 Google drive with all the code

Figure 8.2 Results from ADS-B test

Fig 8.2 shoes successful data received from the ADS-B. This was also saved locally and transmitted to a ground station. We were unable to complete this prototype and prepare it. For launch due to the COVID-19 pandemic. However, we are optimistic that we will complete it soon.

References

[1] "1-Unit CubeSat Structure: ISIS - Innovative Solutions in Space." *ISIS*, 25 Sept. 2019, www.isispace.nl/product/1-unit-cubesat-structure/.

[2] "Addicore GY-521 MPU-6050 3-Axis Gyroscope and 3-Axis Accelerometer IMU." *Www.addicore.com*, www.addicore.com/GY-521-MPU6050-p/170.htm.

[3] "Addicore u-Blox NEO-6M GY-GPS6MV2 GPS Module with on Board EEPROM." *Www.addicore.com*, www.addicore.com/NEO-6M-GPS-p/231.htm.

[4] Brown Space Engineering. "EQUiSat Digital Archive." Brown University , 2018.

[5] *Buy a Raspberry Pi 3 Model B – Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[6] "DS18B20." *DS18B20 TO-92 DIGITAL TEMPERATURE SENSOR*, Twins Chip, www.twinschip.com/image/cache/catalog/Products%20Twins%20Chip%20Store%202020/Shield%20Modules/Sensors/DS18B20%20%20TO-92%20Digital%20Temperature%20Sensor%20Twins%20Chip%201-550x550.jpg.

[7] "DS18B20." Maxim Integrated Products, Inc., 2019.

[8] "HMC5883L Triple-Axis Magnetometer Compass Module GY-273." *Www.addicore.com*, www.addicore.com/HMC5883L-Module-p/ad306.htm.

[9] "IMTQ CubeSat Magnetorquer Board - ISIS - Innovative Solutions In Space." *ISIS*, 12 Aug. 2019, www.isispace.nl/product/isis-magnetorquer-board/.

[10] Lee, Simon, et al. "CubeSat Design Specification Rev. 13." The CubeSat Program, Cal Poly SLO, 20 Feb. 2014.

[11] "MPU-6000 and MPU-6050 Product Specification Revision 3.4." InvenSense Inc., 19 Aug. 2013.

[12] "MPU6050." *Elementz Engineering Guild Pvt Ltd.*, www.elementzonline.com/image/cache/catalog/data/products/Sensors%20and%20Modules/Accelerometers/MPU6050/sku_154602_2-500x500.jpg.

[13] "NEO 6M." *NEO-6M GPS Module*, Maker's Portal, images.squarespace-cdn.com/content/v1/59b037304c0dbfb092fbe894/1561135682906-WWEYOIG7JWB7N7W2NKEE/ke17ZwdGBToddI8pDm48kCJYY-h00dNzpJuBIa-665MUqsxRUqqbr1mOJYKfIPR7LoDQ9mXPOjoJoqy81S2I8N_N4V1vUb5AoIIIbLZhVYxCRW4BPu10St3TBAUQYVKcXCfBfm0f5Ee_9WQJsxv3IHUyKDPayImqZ-wIZeZ3phkwVjVAetzDTY9s_x8A-rhc/neo6m_main.JPG?format=2500w.

[14] "Online Diagram Software & Visual Solution." *Lucidchart*, 2020, www.lucidchart.com/pages/.

[15] "Raspberry Pi 3." *Element 14*, https://in.element14.com/productimages/large/en_GB/SC14013-40.jpg.

"Xbee S2C." *XBEE/Zigbee S2C*, Micron, 5.imimg.com/data5/GM/AE/MY-9380557/zigbee-s2c-500x500.jpg.