



**INSTITUTO TECNOLÓGICO DE MEXICO**  
**INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ**  
DEPARTAMENTO DE ELECTRICA Y ELECTRONICA  
INGENIERIA ELECTRONICA

**Reporte de residencia**

“Diseño y construcción de Satélites CANSAT y propuesta de guía de misión espacial 2019-2020”.

Presenta:

Pedro Antonio Liy Gonzálaez

Asesor interno :

Dr. Hector Ricardo Hernandez de Leon

Asesor Externo

Dr. Alejandro Farah Simon

Tuxtla Gutiérrez, Chiapas



## AGRADECIMIENTOS

A mi padre, hermana, abuela materna y paterna por el apoyo brindado durante toda la carrera y en los momentos más difíciles.

A mi hermano el Dr. Gustavo Ramiro Aranda Lij por enseñarme e inculcarme el camino a la investigación y quantum

A mi asesor interno Dr. Héctor Hernández de León que gracias con su apoyo se pudo llevar a cabo esta residencia sin contar las enseñanzas dadas y abrirme paso a la investigación y al igual que a Bruno Hernández que gracias a él tenemos impresa la estructura del CanSat.

Al asesor externo Dr. Alejandro Simón Farah por la oportunidad brindada para realizar un verano científico, una residencia profesional y una estancia en el Programa espacial Universitario en la cual fue muy importante para mí y para el mi crecimiento académico.

A mis mejores amigos Carlos Manuel Rubio Becerra y Pablo Iván Sánchez de León ya que sin ellos y sin su apoyo este proyecto no hubiera pasado de la primera etapa, las desveladas, la programación, el diseño y pruebas del eyector del paracaídas, aprendí mucho de ustedes.

A cada uno de los integrantes de casa beta y anexos Joe, Ovalle, Roger, Farfan, Fredy, Edison, Alexito, El profe iván, Franco, Iber, Pecha, Diego, Penagos, Felipè , Nata y sin olvidar a alguien que nos cuida desde el universo Lester.

Y por último al equipo del Programa Espacial Universitario por el apoyo de los sensores, un lugar donde trabajar dentro de C.U y al Dr. José Francisco Valdés Galicia Coordinador del Programa Espacial Universitario por el apoyo brindado y a todo el equipo del PEU.

## RESUMEN

El objetivo de este proyecto es la construcción de un picosatélite CanSat que servirá para competición y con los conocimientos adquiridos durante la construcción del CanSat y el concurso nacional de satélites enlatados de la UNAM se elaboró la estructura para el concurso del siguiente año.

La construcción del CanSat va desde las simulaciones hechas a cálculos en donde se calculó la velocidad de caída, la fuerza G al igual que también se realizaron cálculos para la distribución de los sensores desde pesándolos y midiendo el tamaño al igual que se le saco su área transversal y se le diseñó unos “barrotes rompe aire” para mantener un descenso vertical.

El diseño de la estructura del CanSat fue hecho y simulado en Solidworks para posteriormente ser impresa.

Se calculó la medida del paracaídas en la cual se tomó como referencia que a 200 metros este se activara, se tomaron los factores de tres temperaturas que podría haber en el día de lanzamiento y se observó que no cambian mucho solo milímetros así que decidimos tener una medida media de las tres temperaturas para poder hacer el paracaídas.

La programación de los sensores se basó de C++ y C para poder hacerlo en Python ya que las librerías existentes nos daban mucho error así que se optó hacer propias librerías y filtros para que nos de datos reales.

Una vez obtenido esto se hizo su diseño esquemático para poder a partir de eso hacer el diseño PCB para ya montar el CanSat.

Los parámetros a medir serán la velocidad angular, el ángulo de caída, presión atmosférica y temperatura.

## ABSTRACT

The objective of this project is the construction of a CanSat satellite that will serve for competition and with the knowledge acquired during the construction of the CanSat and the national canned satellite competition of the UNAM, the structure for the next year's contest was elaborated.

The construction of the CanSat goes from the simulations made to calculations where the fall velocity was calculated, the G force as well as calculations were made for the distribution of the sensors from weighing them and measuring the size just as their area was removed transverse and "air-breaking bars" were designed to maintain a vertical descent.

The design of the structure of the CanSat was made and simulated in Solidworks to later be printed.

The measurement of the parachute was calculated in which it was taken as a reference that at 200 meters east it was activated, the factors of three temperatures that could be present on the launch day were taken and it was observed that they do not change much only millimeters so we decided to have an average measure of the three temperatures to be able to make the parachute.

The programming of the sensors was based on C++ and C to be able to do it in python since the existing libraries gave us a lot of error if we opted to make our own libraries and filters to give us real data.

Once this was obtained, its schematic design was made to be able to do the PCB design and assemble the CanSat.

The parameters to be measured will be the angular velocity, the angle of fall, atmospheric pressure and temperature.

# INDICE GENERAL

<b>AGRADECIMIENTOS</b>	<b>3</b>
<b>RESUMEN</b>	<b>4</b>
<b>ABSTRACT</b>	<b>5</b>
<b>INDICE GENERAL</b>	<b>6</b>
<b>INDICE DE TABLAS Y FIGURAS</b>	<b>8</b>
<b>Capítulo 1 El satélite enlatado.</b>	<b>9</b>
<b>1.1 Introducción al satélite enlatado.</b>	<b>9</b>
1.1.1 Clasificación de los satélites artificiales.	9
<b>1.2 ¿Qué es un CanSat?</b>	<b>10</b>
<b>1.3 Descripción física de un CanSat.</b>	<b>11</b>
<b>1.4 Objetivos</b>	<b>11</b>
1.4.1 Objetivos generales	11
1.4.2 Objetivos específicos	11
<b>1.5 Hipótesis de trabajo</b>	<b>12</b>
<b>1.6 Justificación</b>	¡Error! Marcador no definido.
<b>1.7 Metodología a usar</b>	<b>12</b>
<b>1.8 Antecedentes</b>	<b>12</b>
<b>1.9 Estado del arte</b>	<b>14</b>
<b>Capítulo 2 Fundamento teórico</b>	<b>15</b>
<b>2.1 Introducción al fundamento teórico</b>	<b>15</b>
<b>2.2 Velocidad en caída libre</b>	<b>15</b>
<b>2.3 Fuerza descendente de la gravedad (<math>F_g</math>) y la fuerza resistiva del arrastre (<math>CD</math>)</b>	<b>17</b>
2.3.1 Fuerza Descendente de la gravedad ( $F_g$ )	17
2.3.2 Coeficiente de arrastre ( $C_d$ )	17
<b>2.4 Área Transversal</b>	<b>18</b>
<b>2.5 Densidad del aire</b>	<b>18</b>
<b>2.6 Coeficiente de rozamiento</b>	<b>19</b>
<b>Capítulo 3 Cálculos y diseño</b>	<b>19</b>
<b>3.1 Calculando su velocidad de caída (Simulación en caída libre)</b>	<b>19</b>
<b>3.2 Cálculos para el paracaídas</b>	<b>22</b>
<b>3.3 Diseño CAD del contenedor.</b>	<b>25</b>
<b>Capítulo 4 Sensores y computadora de vuelo.</b>	<b>28</b>

<b>4.1 Sensores</b>	<b>28</b>
4.1.1 BMP280	29
4.1.2 MAG3110	29
4.1.3 MMA 8451	30
4.1.4 GY-80	30
<b>4.2 GPS</b>	<b>31</b>
4.2.1 Ultimate GPS Adafruit	31
4.3 Reloj RTC	32
4.3.1 Tiny RTC	32
<b>4.4 Telemetría</b>	<b>33</b>
4.4.1 XBee pro S3B pro	33
<b>4.5 Computadora de vuelo</b>	<b>33</b>
4.5.1 Raspberry pi zero W	34
4.5.2 Arduino Nano	34
<b>4.6 Reguladores de voltaje</b>	<b>35</b>
<b><i>Capítulo 5 Construcción y programación del CanSat</i></b>	<b>35</b>
<b>5.1 Programación en los sensores seleccionados</b>	<b>35</b>
<b>5.2 Impresión del diseño hecho en CAD</b>	<b>40</b>
<b>5.3 Diseño del PCB y diagrama esquemático</b>	<b>41</b>
<b>5.4 Construcción del paracaídas</b>	<b>45</b>
<b>5.5 Acoplamiento final del CanSat</b>	<b>46</b>
<b><i>Capítulo 6 Pruebas y día del lanzamiento</i></b>	<b>49</b>
<b>6.1 Pruebas a distancia de telemetría</b>	<b>49</b>
<b>6.2 Día del lanzamiento</b>	<b>50</b>
<b><i>Capítulo 7 Resultados</i></b>	<b>51</b>
<b>7.1 Resultados en pruebas</b>	<b>52</b>
<b>7.2 Resultados del día de lanzamiento</b>	<b>52</b>
<b><i>Capítulo 8 Conclusiones</i></b>	<b>55</b>
<b>8.1 Conclusiones del diseño del CanSat</b>	<b>55</b>
<b>8.2 Conclusiones generales.</b>	<b>55</b>

## INDICE DE TABLAS Y FIGURAS

<i>Figura 2.1 Grafica de metros sobre tiempo.</i>	16
<i>Figura 2.2 Grafico entre velocidad entre tiempo</i>	16
<i>Figura 3.1 Grafica de aceleracion en funcion del tiempo.</i>	21
<i>Figura 3.2 Grafica de velocidad con respecto al tiempo.</i>	22
<i>Figura 3.3 Tapa superior rosca externa.</i>	26
<i>Figura 3.4 Contenedora de electronica rosca interna</i>	26
<i>Figura 3.5 Contenedora de tripulante rosca externa</i>	27
<i>Figura 3.6 Simulacion del ensambre</i>	28
<i>Figura 4.1 BMP280</i>	29
<i>Figura 4.2 MAG3110</i>	30
<i>Figura 4.3 MMA8485</i>	30
<i>Figura 4.4 GY-80</i>	31
<i>Figura 4.5 GPS Ultimate por adafruit</i>	32
<i>Figura 4.6 Reloj RTC</i>	33
<i>Figura 4.7 Xbee pro S3B</i>	33
<i>Figura 4.8 Raspberry pi zero W</i>	34
<i>Figura 4.9 Arduino nano</i>	34
<i>Figura 4.10 Ams117 modulo</i>	35
<i>Figura 5.1 Visita previa del impreso en 3D</i>	40
<i>Figura 5.2 Sensores y modulos</i>	41
<i>Figura 5.3 Etapa de potencia</i>	42
<i>Figura 5.4 Computadoras de vuelo.</i>	42
<i>Figura 5.5 Etapa de telemetria.</i>	43
<i>Figura 5.6 Diseño de PCB</i>	44
<i>Figura 5.7 Diagrama general del Esquematico</i>	44
<i>Figura 5.8 Hilo de nylon      Figura 5.9 Tela de paracaidas</i>	45
<i>Figura 5.10 Acoplamiento PCB a rosca interna</i>	46
<i>Figura 5.11 PCB subsistemas</i>	47
<i>Figura 5.12 PCB Arduino</i>	47
<i>Figura 5.13 Acoplamiento ya soldado</i>	48
<i>Figura 5.14 Oasis y el huevo</i>	48
<i>Figura 5.15 Vista posterior de acoplamiento.</i>	48
<i>Figura 5.16 Cansat listo para la mision.</i>	49
<i>Figura 7.1 CanSat listos para la mision espacial.</i>	50
<i>Figura 7.2 CanSat despues del impacto de 400m</i>	51
<i>Figura 7.3 Telemetria enviada por Xbee</i>	52
<i>Figura 7.4 Direcciones I2C de los sensores</i>	53
<i>Figura 7.5 Recopilacion de datos sensores</i>	53
<i>Figura 7.6 Grafico de temperatura      Figura 41 Grafico de la aceleracion</i>	54
<i>Figura 7.7 Grafico de los ejes      Figura 43 Grafico de la precion</i>	54
<i>Figura 7.8 Datos guardados en excel</i>	55
<i>Tabla 1.1 Clasificacion de los satelites.</i>	10
<i>Tabla 3.1 Tabla de velocidad y tiempo en caida</i>	21



## Capítulo 1 El satélite enlatado.

Un CanSat es la simulación de un satélite real ensamblado dentro de un área de una lata de refresco, el reto de este es acoplar cada sub nivel del mismo tales como son la etapa de potencia, sensores, computadora de vuelo y la telemetría, para poder simular el satélite enlatado este se tiene que subir a una cierta altitud para luego ser liberado en el cual el satélite tendrá que medir la telemetría acorde a la misión espacial desde el inicio de la misión hasta el aterrizaje.

El principal objetivo es transmitir el concepto básico diseño y construcción de un satélite.

### 1.1 Introducción al satélite enlatado.

A lo largo de la historia los seres humanos hemos buscado las diferentes formas de comunicación ya sean desde señales de humo, señas, el habla y todo esto fue evolucionando hasta buscar un nuevo método de comunicaciones. Hace 60 años se lanzó el primer satélite artificial llamado Spuntnik el cual fue lanzado y puesto en órbita el cual era una esfera de aluminio que pesaba 80 Kg, logró transmitir datos de temperatura y presión por medio de telemetría en las frecuencias de los 20Mhz a 40Mhz, aunque su tiempo de operación fue muy corto (3 meses) abrió una brecha importante.

El mercado de las telecomunicaciones es el principal para este pero hay muchos que han sido construidos con fines científicos de investigación como son :

- Estudios meteorológicos
- Exploración y observación de la tierra
- Estudios de astronomía

e igual tenemos los que son para fines militares y defensa.

La inversión económica para crear un satélite puede ser muy alta desde el diseño hasta la elaboración del mismo y se requiere de tecnología avanzada y un equipo certificado y muchos países en el cual México no se queda atrás tenemos satélites en órbita y ahora gracias a los avances que tenemos los componentes eléctricos y electrónicos se hacen cada vez más pequeños gracias a estos podemos llevar a cabo el diseño y construcción de un picosatélite.

Estos pequeños satélites juegan un papel muy importante no solo en lo académico, sino que también en proyectos de investigación en lo que los instrumentos compactos pueden ir en la carga útil del satélite, pero sobre todo el tiempo y costo para reducirlos es muy pequeño a comparación de los de tamaño estándar.

#### 1.1.1 Clasificación de los satélites artificiales.

Los satélites pueden ser clasificados por el costo, peso, función, tipo de orbita, pero la clasificación por su masa puede estar relacionada con los costos de lanzamientos.

A continuación, dejamos con una tabla de clasificación de los mismos.

<b>Tipo</b>	<b>Masa (Kg)</b>
<b>Grandes satélites</b>	mayor que 1000 Kg
<b>Medianos Satélites</b>	500 a 1000 Kg
<b>Minisatélites</b>	100 a 500 Kg
<b>Microsatélites</b>	10 a 100 Kg
<b>Nanosatélites</b>	1 a 10 Kg
<b>Picosatélites</b>	0.1 a 1 kg
<b>Femto Satélites</b>	menor que 0.1 Kg

*Tabla 1.1 Clasificación de los satélites.*

## 1.2 ¿Qué es un CanSat?

El concepto CanSat fue propuesto en 1999 por el profesor Robert Twiggs, del Laboratorio de Desarrollo Espacial de Stanford y su principal objetivo es transmitir los conceptos básicos para el diseño y construcción de satélites.

Un CanSat simula una plataforma espacial que en este caso debido a su masa cabe dentro del rango de los picosatélites

Los CanSat no son puestos en órbita, pero son lanzados a diferentes alturas desde 200m hasta 1Km de altura siempre cuidando de no interferir en el tráfico aéreo, pueden ser lanzados desde cohetes propulsados por agua o combustión, globos meteorológicos, aviones a escala o drones.

Para considerar una misión el CanSat debe de ser completamente autónomo al momento de su caída libre la cual deben de desplegar un paracaídas o un sistema de aspas para poder contrar un descenso, deben de transmitir telemetría hacia una estación terrena y una misión puede ser desde enviar y transmitir datos hasta algo más complejo como controlar el giro durante la caída del mismo como poner un ejemplo.

Puede ser considerado un CanSat como introducción a la tecnología espacial debido a su bajo costo y tiempo de creación del mismo.

Gracias a estos simuladores en muchos países se ha difundido y los cuales las universidades de estos países lo ven como un complemento para la formación de los alumnos y al igual que muchas instituciones alrededor del mundo cuentan con competiciones a nivel nacional e internacional porque su diseño, construcción y pruebas se siguen los conceptos operacionales de una misión espacial y estos dispositivos pueden ser utilizados para experimentos científicos de mayor envergadura.

### 1.3 Descripción física de un CanSat.

El diseño y configuración de un CanSat puede ser muy diferentes ya que dependen de la misión a realizar.

Como muestra podemos ver el clásico CanSat que es el que está implementado en una lata de refresco y los componentes de su telemetría pueden ser desde un sensor de temperatura y humedad hasta un IMU donde podremos calcular su velocidad de caída e inclinación entre demás cosas, si se combinan todos estos ya es una misión más compleja, como se muestra en la figura() un modelo de un CanSat en una lata de refresco O si se toma como referencia el CanSat hecho por la Agencia Espacial Mexicana (AEM) durante su curso de capacitación a profesores y el cual está conformado de la siguiente manera.

El subsistema de computadora de vuelo está compuesto por un Arduino nano, el subsistema de potencia tiene como fuente una batería de litio, el subsistema de comunicación es con una XBee pro S2B y la cual el receptor puede ser el mismo o una antena conectada a una computadora portátil y por último el subsistema de sensores el cual contiene un GPS, un sensor de temperatura y humedad DHT11, y un IMU GY-80 que cuenta con acelerómetro, giroscopio, barómetro y compás digital.

Cabe mencionar que cada subsistema tiene su propia placa y son conectadas entre sí para así formar una pequeña torre que se convertiría en nuestra carga útil.

### 1.4 Objetivos

#### 1.4.1 Objetivos generales

Construcción y mejora de satélites CanSat y propuesta de guía de misión 2019-2020.

#### 1.4.2 Objetivos específicos

- Construcción de un picosatélite CanSat de competición
- Participación en el concurso nacional de CanSat 2018-2019

- Construcción de un sistema de liberación de paracaídas
- Diseño en software CAD de la estructura del picosatélite dado los datos obtenidos desarrollar una convocatoria para el concurso nacional del siguiente año
- Desarrollar un esquema de competencia para que el instituto tecnológico de Tuxtla Gutiérrez tenga una competencia CanSat.

## 1.5 Hipótesis de trabajo

Se cuidarán los datos arrojados por los sensores que sean reales y no con ruido para una aplicación con la liberación del paracaídas y su control del autogiro.

## 1.6 Justificación

Gracias a los datos obtenidos y a la experiencia del concurso nacional podremos hacer mejoras para el satélite lo cual sería muy bueno para empezar hacer tecnología espacial en nuestra institución y poder participar en concursos internacionales y nacionales a un nivel alto.

## 1.7 Metodología a usar

Para este caso vamos a utilizar el método inductivo porque nos involucramos desde el diseño y construcción del CanSat y en la cual también tomaremos parte del método científico porque necesitamos resolver el problema del autogiro y esta nos ayudará a la misma desde nuestra dificultad que serían el sistema de paracaídas, la calibración de los sensores y el autogiro y en las cuales vamos a estudiar cada una de las acciones para buscar su solución acorde a nuestra disposición y llegar a una hipótesis verdadera para poder medir también sus consecuencias para poder verificarlas y darle una solución.

## 1.8 Antecedentes

El primer satélite mexicano en optar la tierra fue el Morelos 1 el cual es un satélite de telecomunicaciones construido y puesto en órbita por la SCT, el cual su lanzamiento fue el 17 de junio de 1985 desde cabo cañaveral y con la ayuda del transbordador espacial Discovery bajo el nombre de la misión espacial STS-51-G el cual contenía dos satélites de comunicación los dos siendo los primeros para los países de México y Arabia Saudita.

En 1957 para ser más exactos el 4 de octubre de 1957 fue lanzado el primer satélite artificial este se convirtió en uno de los múltiples satélites lanzados por la Unión soviética el programa se llamó igual que el satélite Spuntnik cabe mencionar que el primer fracaso de estos llegó hasta el Spuntnik.

La universidad politécnica del estado de california (Cal Pot) y la Universidad de Stanford desarrollaron en conjunto las especificaciones del CubeSat con el objetivo de realizar proyectos de ciencia espacial, el cual consta en un estándar de diseño de Nanosatélites cuya estructura es en cubos de 20 cm de arista y una masa inferior a 1.33Kg, su diseño más simple consta de 1U “una unidad” y los cuales son escalables en incremento de medida por ejemplo 2U (20x10x10) los cuales apuntan a experimentos más complejos.

Como lo pone el manual NPR 7120-5 de la NASA el ciclo de vida de la misión tratara de las fases que componen la misión de principio a fin la cual está de la siguiente forma.

- Estudio conceptual, producción de ideas y alternativas para llevar a cabo la misión
- El desarrollo de conceptos y tecnología donde se determina la factibilidad y planeación para establecer una línea base
- Culminación del diseño preliminar y tecnología donde detallaremos la misión y nos saldrán las necesidades para la realización de la misión
- Diseño final y fabricación de los subsistemas de operación incluyendo hardware y software.
- Ensamble del sistema, pruebas e integración, lanzamiento de pruebas e integración de subsistemas.
- Operación y mantenimiento de la misión identificando necesidades durante la operación del sistema
- Cierre y análisis de los datos obtenidos con sus respectivas conclusiones

La secretaria de comunicaciones y transportes (SCT) con la ayuda del gobierno mexicano adquirieron tres nuevos satélites con fines de seguridad los cuales son conocidos como Mex-Sat los cuales están para operar en las bandas Ly Ku mientras que el Mex-Sat opera en las bandas C y Ku, los centros de control se encuentran localizados en Hermosillo sonora y en Iztapalapa con el nombre de Centros de Control Y Monitoreo Satelital Mexicano inaugurandose el 29 de noviembre de 2012.

- Mex-Sat 1 (centenario) el cual obtuvo su nombre como parte de los festejos del centenario de la Revolución Mexicana, este satélite tiene una vida útil de 15 años y el cual el día del lanzamiento 16 mayo del 2015 tras pasar los primeros 8 minutos un fallo en la tercera etapa del cohete portador Protón-M provocó el reingreso a la tierra a una altitud de 170 kilómetros en la cual al momento de reingresar a la atmósfera cayo desintegrándose cuyos restos cayendo en Siberia.
- Mex-Sat 2 (Morelos III) continuación de los dos satélites Morelos puestos en órbita, este satélite secundario para el servicio móvil de la red MEXSAT fue lanzado el 2 de octubre de 2015 en un cohete Atlas V 421 AV-059 echo por United Launch Alliance cuyo lanzamiento fue un éxito y ocupa la posición 113.1W en la órbita geoestacionaria.
- Mex-Sat 3(Mexsat Bicentenario) al igual que el Mex-Sat 1 nombrando así como parte del bicentenario del grito de la independencia y es el primero de la red MEXSAT en ponerse en órbita el cual su fecha de lanzamiento es el 19 de diciembre del 2012 y el cual tiene la tarea de controlar de los otros satélites de la red ya que es un satélite para el servicio fijo de la plataforma STAR-” y fue fabricado por la compañía Orbital

Sciences Corporation, se encuentra en la órbita geoestacionaria en la longitud de 114.9 W.

## 1.9 Estado del arte

El Instituto Nacional de Astrofísica Óptica y Electrónica desarrollo un reporte donde describen un satélite tipo CanSat de diseño ruso la cual se ha integrado y hecho pruebas, en él describen que es del tamaño de la lata de refresco y es de bajo costo, en las cuales en cada uno de sus apéndices nos muestran las etapas de cada una de los sub sistemas al igual que se deja unos ejemplos de la programación de la misma, cabe mencionar que se usaron arreglos de resistencias y capacitores para su tarjeta de radio frecuencia. (Torres, 2015)

Diseño y construcción de un Microsatélites(CanSat) en la cual el equipo llamado La Vagoneta Espacial nos describen la construcción y programación de un CanSat para la competencia europea en esta ocasión nos muestran que utilizaron una Raspberry pi Zero W como computadora de vuelo y nos muestran cómo hacer la vinculación con google earth. (Espacial, 2017)

La Agencia Espacial Mexicana (AEM) crea un laboratorio de diseño, construcción y pruebas de satélites pequeños en las cuales nos enseñan en un curso presencial las etapas de diseño, modelado y construcción del mismo para al final hacer unas pruebas, nos proporciona un pequeño simulador en Arduino pero solo es funcional si se sigue el orden de los pines dados en un documento tipo instructivo para hacer funcionar el simulador, su construcción está diseñada para ocupar todos los espacios de la lata ya que no cuenta con alguna misión extra. ((AEM), 2017)

Análisis dinámico estructural de satélite educativo CanSat hecho por la Universidad Autónoma de Baja California con colaboración con el Instituto Politécnico Nacional nos muestra el análisis dinámico del comportamiento que experimenta la estructura de un picosatélite CanSat en el cual el modelo de la estructura se analiza por medio de la teoría del elemento finito para el estudio de la estructura se utilizó una plataforma dedicada la cual ayudó a predecir la respuesta de la estructura durante el lanzamiento del mismo, nos cuenta que los análisis se realizaron con fuerzas de aceleración de 30g en dirección vertical. (Roa, 2017)

Ángel Colin nos describe en un artículo la construcción de un picosatélite CanSat en el cual nos muestra la descripción física y la arquitectura de su CanSat del cual se basa del programa de capacitación y entrenamiento de la Agencia Espacial Mexicana AEM. (Colin, 2016)

## Capítulo 2 Fundamento teórico

En este capítulo daremos un pequeño resumen sobre los conceptos de física que utilizamos para poder hacer la simulación del CanSat y de ahí parte para los cálculos del paracaídas, con los datos obtenidos también podemos calcular el control para el autogiro.

### 2.1 Introducción al fundamento teórico

Los conceptos de física que aplicaremos para poder simular nuestro CanSat para poder obtener datos como son los siguientes.

- Velocidad máxima de caída.
- su rozamiento del aire.
- el área transversal con las diferentes excepciones de caída.

ya que cada uno de estos conceptos nos dará los datos para la creación del paracaídas y poder predecir el autogiro mientras que este está en caída libre.

### 2.2 Velocidad en caída libre

Se le llama caída libre al movimiento que se debe únicamente a la influencia de la gravedad, todos los cuerpos con este tipo de movimiento tienen una aceleración dirigida hacia abajo, en la tierra este valor es de  $9.8\text{m/s}^2$  es decir los cuerpos dejados en caída libre aumentan su velocidad hacia abajo en  $9.8\text{m/s}^2$  cada segundo.

La gráfica para la caída libre entre posición y tiempo tiene una apariencia de una curva, la pendiente cada vez más negativa nos indica que la velocidad del cuerpo es cada vez más negativa y significa que el movimiento se va haciendo más rápido a medida que transcurre el tiempo.

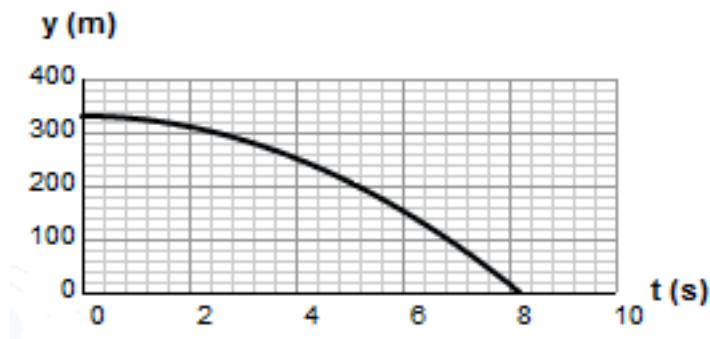


Figura 2.1. Grafica de metros sobre tiempo.

y en una gráfica de velocidad en tiempo de caída libre su forma recta nos indica que la aceleración es constante, es decir que la variación de la velocidad en intervalos regulares de tiempo constante.

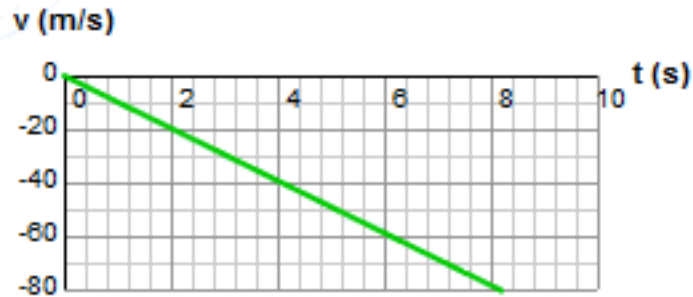


Figura 2.2 Grafico entre velocidad entre tiempo

La pendiente negativa nos indica que la aceleración es negativa y podemos ver que en intervalos de un segundo es siempre la misma (-9.8m/s).

Recordemos las ecuaciones generales del movimiento

$$e = V_0 * t + \frac{1}{2} * a * t^2$$

EC 1.1

$$V_f = a * t$$

EC 2

Por otro lado, en una caída libre la posición que ocupa un cuerpo en un instante es precisamente su altura H en ese momento.

como la aceleración la podemos llamar g que es lo que experimenta en caída libre y la ecuación nos podría quedar de la siguiente manera.



## 2.3 Fuerza descendente de la gravedad (Fg) y la fuerza resistiva del arrastre (CD)

### 2.3.1 Fuerza Descendente de la gravedad (Fg)

Todos los cuerpos son atraídos hacia el centro de la tierra con una aceleración de  $9.8\text{m/s}^2$  despreciando la resistencia del aire y pensando en coordenadas en los tres ejes X, Y y Z el movimiento será en el eje Y donde comprendiendo las expresiones de velocidad y aceleración antes vistas tenemos una serie de ecuaciones para la caída libre

$$Y = Y_0 + V_0 t - \frac{1}{2} g t^2 \quad (\text{Para encontrar la posición})$$

EC 3

$$\Delta y = V_0 t - \frac{1}{2} g t^2 \quad (\text{Para encontrar el desplazamiento})$$

EC 4

$$\Delta y = \frac{V_0 t - V_y}{2 \cdot t} \quad (\text{Para encontrar la rapidez en Y})$$

EC 5

$$V_t = V_0 - g t \quad (\text{Para encontrar en cualquier instante})$$

EC 6

$$V_y^2 = V_0^2 - 2g \Delta y \quad (\text{Para encontrar la rapidez en } y^2)$$

EC 7

Podemos observar que utilizamos las mismas expresiones que utilizamos en velocidad y en aceleración la diferencia es que el movimiento se hace en el eje Y no en el X donde él  $-1/2$  es porque el movimiento es hacia abajo.

### 2.3.2 Coeficiente de arrastre (Cd)

El coeficiente de arrastre es una función del número de Reynolds, Re. Este número es importante para definir el comportamiento de un fluido.

El número Re se define para un objeto esférico de diámetro D

$$Re = \frac{\rho F Dv}{n} \quad EC 8$$

N es la velocidad del fluido

Existen muchas fórmulas que describen con mayor o menor aproximación a las medidas del coeficiente de arrastre (CD) para un objeto en forma de esfera con un amplio número de Reynolds lo cual nos quedará de la siguiente forma.

$$Cd = \frac{\frac{24}{Re} + 2.6\left(\frac{Re}{263000}\right)^{-7.94}}{1 + \left(\frac{Re}{263000}\right)^{-8.00}} + \frac{Re^{0.80}}{461000} \quad EC 9$$

Con esta forma describiremos el coeficiente de arrastre cambia apreciablemente con el número Re.

## 2.4 Área Transversal

Una Sección o área transversal es un corte de dos dimensiones y dependiendo de la orientación del corte o del plano varias secciones transversales pueden resultar en un círculo o rectángulo.

Una vez sabiendo eso el área transversal sería de un círculo el cual sustituyendo la fórmula

$$A = \pi * r^2 \quad EC 10$$

como el radio es la mitad del diámetro lo sustituimos por.

$$A = \pi * \left(\frac{D}{2}\right)^2 \quad EC 11$$

con eso obtenemos su área transversal de nuestro CanSat.

## 2.5 Densidad del aire

La densidad del aire es la cantidad de masa por unidad de volumen, para esto no es necesario hacer un cálculo ya que existen tablas dependiendo de la altitud a nivel de mar y la temperatura del medio ambiente podemos obtener ese dato.

## 2.6 Coeficiente de rozamiento

Coeficiente que expresa la oposición al movimiento que ofrecen las superficies de dos cuerpos en contacto.

El coeficiente de rozamiento es también llamado coeficiente de fricción y se representa con la letra griega  $\mu$  (mu) y a escala microscópica casi todas las superficies aún las pulidas son extremadamente rugosas.

El valor depende además de muchos factores como la temperatura, el acabado del material, velocidad relativa.

Conociendo el valor del coeficiente de rozamiento aplicable, la fuerza de rozamiento máxima que se puede ejercer una superficie sobre otra se expresa como el producto del coeficiente de rozamiento por la fuerza normal a ambas superficies dado por esta fórmula.

$$Fr = \mu Fn \quad EC 12$$

## Capítulo 3 Cálculos y diseño

En este capítulo se mostrarán los cálculos hechos a través de la integración del CanSat siguiendo los lineamientos del concurso nacional de CanSat que igualmente en la dicha competencia nos piden algunos datos de simulación.

### 3.1 Calculando su velocidad de caída (Simulación en caída libre)

Para resolver el tiempo de caída del satélite enlatado se procedió a realizar ecuaciones de equilibrio considerando una caída libre, debido a la altura de 500m se realizó el modelado por el medio de ecuaciones diferenciales tomando en cuenta que la fuerza de rozamiento del aire no lineal y está dada por la siguiente ecuación.

$$Fd = \frac{1}{2} CdApV^2 \quad EC 13$$

Sabemos que la densidad del fluido es una función que depende de la temperatura y la presión y para este caso se tomarán densidades distintas para realizar un muestreo con la finalidad de observar las diferencias en los resultados números.

El área transversal fue considerada como la superficie circular del modelo CanSat y de la muestra ecuación de equilibrio obtenemos la ecuación diferencial.

$$ma(t) = mg - \frac{1}{2} CdApv(t)^2$$

*EC 14*

$$\frac{d}{dt} v(t) = \frac{1}{2} m CdApv(t)^2$$

*EC 15*

Y resolviendo por el método de variables separables para después integrar se obtiene las dos ecuaciones principales.

$$v(t) = \left( \sqrt{\frac{2gm}{Cdpa}} \right) \tanh \left( t \left( \sqrt{\frac{gCdpa}{2m}} \right) \right)$$

*EC 16*

$$t(y) = \text{Cosh}^{-1} \frac{\left( e \frac{Cdpa}{2m} \right) y}{\sqrt{\frac{gCdpa}{2m}}}$$

*EC 17*

Ya obtenida nuestras ecuaciones procedemos a graficar en función del tiempo la distancia para nuestros tres casos de densidad del aire, así mismo graficamos la velocidad que dependiendo del tiempo aprovechando que de esa ecuación podemos obtener la energía cinética de nuestro satélite solo recordamos la expresión dada por la siguiente fórmula.

$$Ec = \frac{1}{2} m v^2$$

*EC 18*

Sustituyendo la expresión de velocidad tenemos.

$$Ec = \frac{1}{2} \left( \left( \sqrt{\frac{2gm}{Cdpa}} \right) \tanh \left( t \left( \sqrt{\frac{gCdpa}{2m}} \right) \right) \right)^2$$

*EC 19*

Una vez que tenemos nuestras ecuaciones procedemos a realizar los tres cálculos para las siguientes temperaturas.

- 20°

- 25°
- 30°

todo esto con la ayuda de Matlab se pudo graficar y obtener la aceleración máxima donde consideramos.

- $Cd = 0.8$
- $m = 0.5 \text{ Kg}$
- $h = 400 \text{ m}$

Temperatura	Velocidad	Tiempo en caída
20°	1.240 $\text{kg/m}^2$	11.4912 s
25°	1.1840 $\text{kg/m}^2$	11.3338 s
30°	1.1650 $\text{kg/m}^2$	11.3790 s

Tabla 3.1 Tabla de velocidad y tiempo en caída

Una vez obtenido estos datos ya podemos calcular las medidas de nuestro paracaídas.

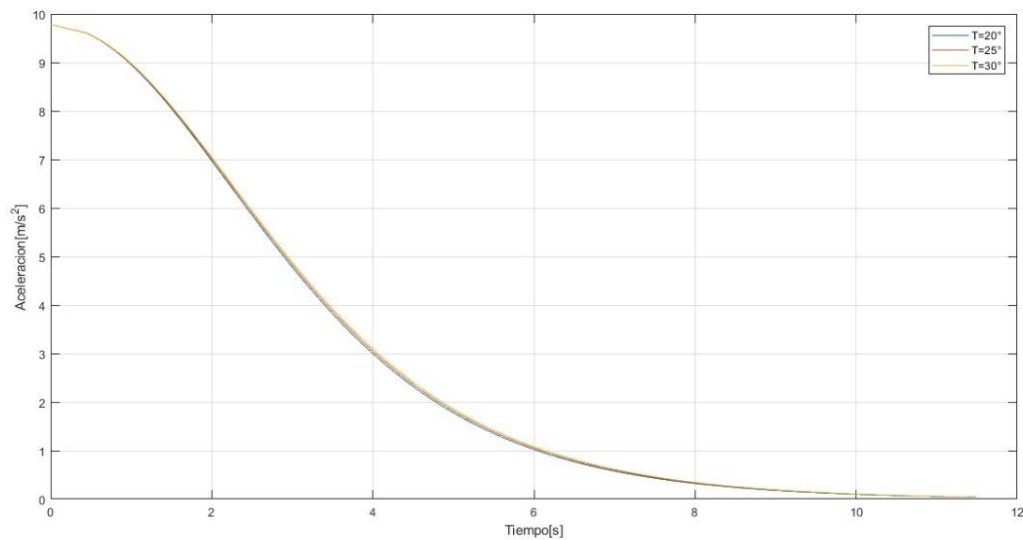


Figura 3.1 Grafica de aceleración en función del tiempo.

Como se puede observar en la figura 3 se obtuvo la gráfica con respecto al tiempo y en la cual se graficaron tres veces correspondientes para las diferentes condiciones del clima para el día de lanzamiento.

Para la figura 4 se graficó su velocidad con respecto al tiempo y de igual manera se graficó en tres partes para las diferentes condiciones climáticas.

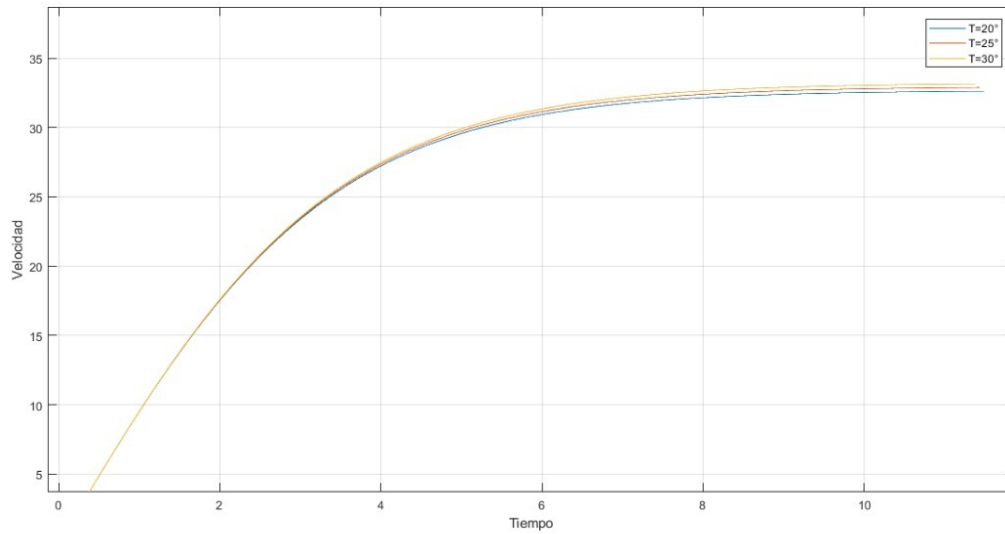


Figura 3.2 Grafica de velocidad con respecto al tiempo.

### 3.2 Cálculos para el paracaídas

para el cálculo del paracaídas se tomó en cuenta tres diferentes tipos de temperaturas que se pueden dar en el día del lanzamiento para esos son 20°, 25° y 30° dado los datos ya obtenidos con los cálculos anteriores podemos determinar las dimensiones de nuestro paracaídas.

Comenzamos calculando el área transversal que está dada de la fórmula.

$$A = \frac{\pi}{4}(d)^2$$

*EC 20*

Donde d es igual al diámetro del CanSat.

$$A = \frac{\pi}{4}(11)^2$$

$$A = \frac{121}{4}\pi$$

$$A = 95.033117777 \text{ cm}^2$$

Una vez obtenida estos datos ahora toca el paso de calcular la densidad del aire para vamos a utilizar la siguiente ecuación.

$$D = \frac{1}{2} \rho C_d A V^2$$

EC 21

Donde.

D = Densidad del aire

$\rho$  = Densidad del fluido

$C_d$  = coeficiente de rozamiento y se utilizará la constante de 0.8

A = Área transversal

$V^2$  = Velocidad al cuadrado

Ya realizando los cálculos para cada temperatura nos dan.

- $20^\circ = 1.204 \text{ kg/m}^3$
- $25^\circ = 1.1840 \text{ kg/m}^3$
- $30^\circ = 1.1650 \text{ kg/m}^3$

Y la velocidad de caída es diferente para cada tipo de temperatura al igual que la densidad del fluido.

Para una velocidad a.

- $20^\circ = 32.6192 \text{ m/s}$
- $25^\circ = 32.8874 \text{ m/s}$
- $30^\circ = 33.1476 \text{ m/s}$

Dado estos datos ya podemos utilizar la siguiente ecuación para poder encontrar la densidad del aire con la siguiente expresión.

$$D = \frac{1}{2} \rho C_d A v^2$$

EC 22

Para cada uno de los tres casos con la temperatura diferentes, primero vamos por el cálculo para la temperatura a  $20^\circ$

$$D = \frac{1}{2} (1.204 \text{ kg/m}^3)(0.8)(95.0331777 \text{ cm}^2)(32.6192 \text{ m/s})^2$$

$$D = 4.869470201 \text{ kg/m}^3 \text{ (para } 20^\circ\text{)}$$

Ahora para  $25^\circ$

$$D = \frac{1}{2} (1.1840 \text{ kg/m}^3)(0.8)(95.0331777 \text{ cm}^2)(33.1476 \text{ m/s})^2$$

$$D = 4.867949077 \text{ kg/m}^3 \text{ (para } 25^\circ\text{)}$$

Ahora para  $30^\circ$

$$D = \frac{1}{2} (1.1650 \text{ kg/m}^3)(0.8)(95.0331777 \text{ cm}^2)(33.1476 \text{ m/s})^2$$

$$D = 4.865924289 \text{ kg/m}^3 \text{ (para } 30^\circ\text{)}$$

Una vez obtenido la densidad del aire nos disponemos a obtener el área del paracaídas de los tres tipos diferentes de temperatura.

Se usará la siguiente expresión para poder obtener el área del paracaídas.

$$A = 2 * g * m / p * Cd * v^2$$

*EC 23*

Y recordando que el área del círculo está dada por.

$$A = \pi * r^2$$

*EC 24*

Queda nuestra ecuación dada por el área del círculo de la siguiente manera.

$$A = \sqrt{\frac{2 * g * m}{\pi * p * Cd * V^2}}$$

*EC 25*

De igual manera se van hacer los cálculos para los tres eventos posibles 20°, 25° y 30° donde  
g = gravedad  
m = masa

Comenzamos los cálculos para 20°

$$A = \sqrt{\frac{2 * (9.79 \text{ m/s}^2) * (500 \text{ gms})}{\pi * (1.204 \text{ kg/m}^3) * (0.8) * (32.6192 \text{ m/s})^2}}$$

$$r = 5.780418627 \text{ cm}$$

$$\emptyset = 11.56083723 \text{ cm (para } 20^\circ\text{)}$$

Ahora para 25°



$$A = \sqrt{\frac{2 * (9.79 \text{ m/s}^2) * (500 \text{ gms})}{\pi * (1.1849 \text{ kg/m}^3) * (0.8) * (32.8874 \text{ m/s})^2}}$$

$$r = 5.718498914 \text{ cm}$$

$$\emptyset = 11.43699783 \text{ cm (para } 25^\circ)$$

Ahora para 30°

$$A = \sqrt{\frac{2 * (9.79 \text{ m/s}^2) * (500 \text{ gms})}{\pi * (1.1650 \text{ kg/m}^3) * (0.8) * (33.1476 \text{ m/s})^2}}$$

$$r = 5.782701674 \text{ cm}$$

$$\emptyset = 11.56540335 \text{ cm (para } 30^\circ)$$

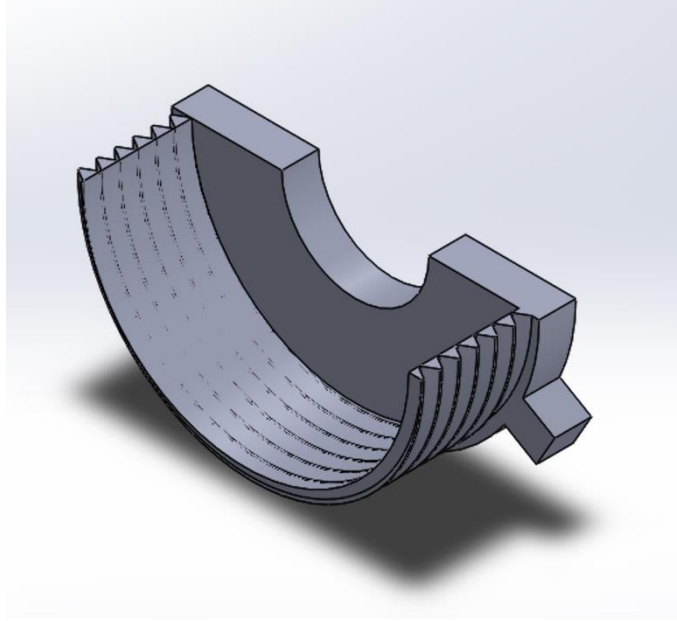
Cabe mencionar que los cálculos se hicieron para que a los 200 metros desplegará el paracaídas.

### 3.3 Diseño CAD del contenedor.

El diseño de la envolvente de nuestro CanSat se va hacer desde el software de Solidworks el cual nos tenemos que acatar a las reglas del concurso nacional las cuales son 22 cm máximos de altura y 10 cm de diámetro como máximo y la cual si contiene unas “patas” para el descenso las cuales tendrán 1 cm de espesor.

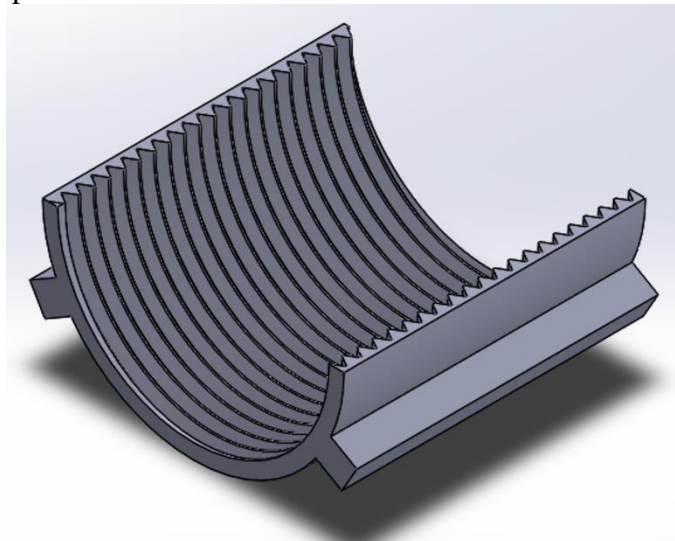
El diseño del cual se hizo en tres partes la cual cuenta con la tapa posterior, la parte contenedora de la electrónica que será una rosca interna y luego la contenedora del tripulante que después se llenará del material llamado “oasis” que sirve para decorar arreglos florales

La tapa superior se diseñó de tal forma que tenga una rosca externa con la cual se pueda acoplar con la parte contenedora de la electrónica cuenta con un diámetro de 10 cm y la parte de la rosca externa es de 3 cm como se muestra en la figura 3.1.



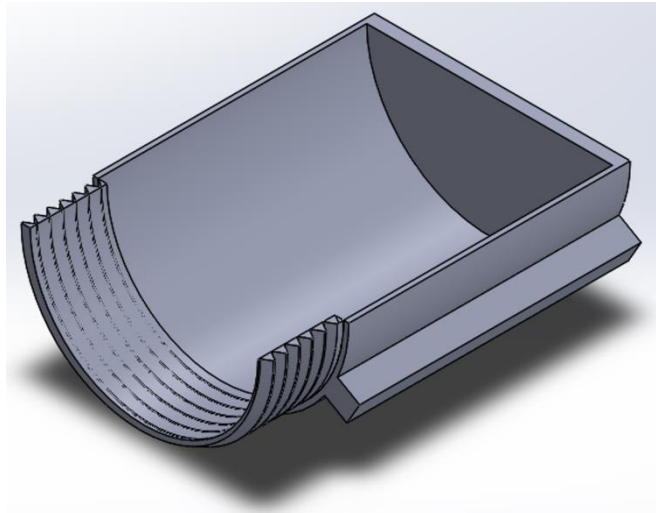
*Figura 3.3 Tapa superior rosca externa.*

La contenedora de la electrónica es un cilindro de 20 de diámetro a la cual se le hizo una perforación de 9.50 cm de diámetro para poder hacer una rosca interna que recorre toda la altura del contenedor como se muestra en la figura 6 figura, se optó por este diseño porque con la rosca interna del contenedor para ayudarnos a colocar el PCB en forma de ir enroscado y sin necesidad de hacer una estructura aparte.



*Figura 3.4 Contenedora de electrónica rosca interna*

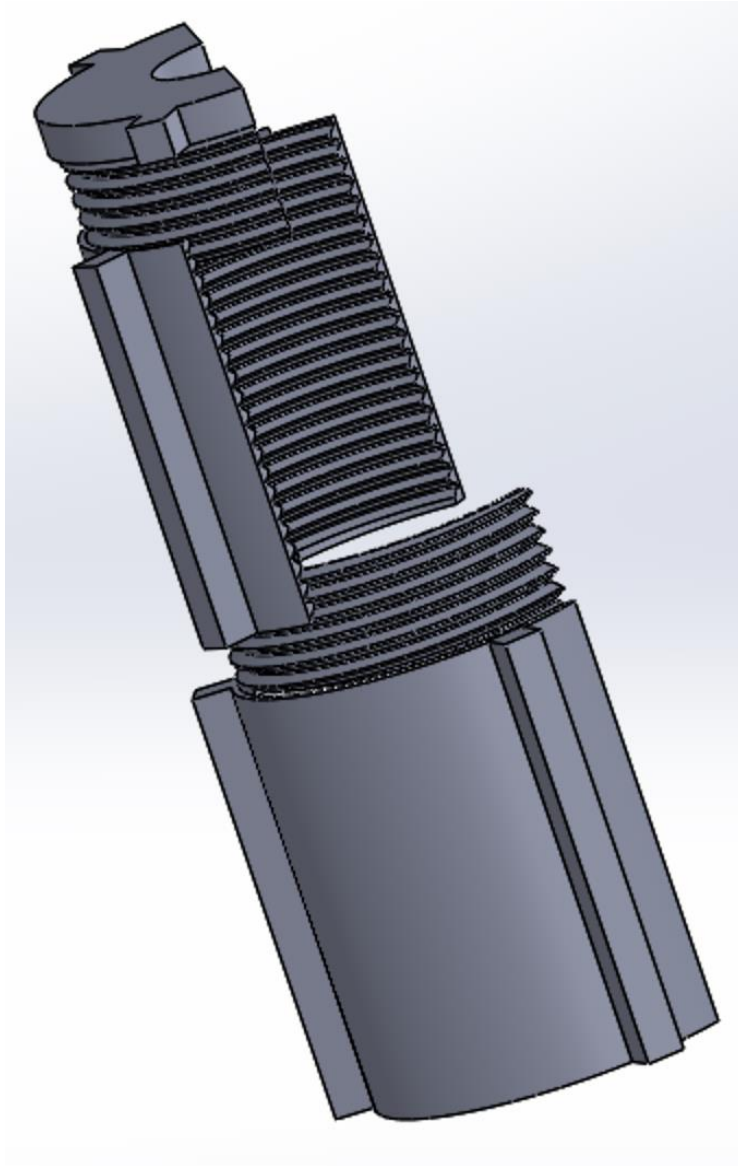
El contenedor del tripulante el cual será un huevo en este caso ya que nos acoplamos para esa misión, como todos los CanSat se tienen que acoplar para las misiones esta deberá de llevar un huevo y en la cual se tendrá que cubrir de un material que al momento de recibir un impacto este se pone blando y el que encontramos y propusimos fue uno llamado “oasis” que este se utiliza para colocar las flores y en arreglos de las mismas, el contenedor tiene una rosca externa al igual que la tapa para poderse acoplar a la parte contenedora de la electrónica y se muestra a continuación.



*Figura 3.5 Contenedora de tripulante rosca externa*

Una vez ya ensamblado cada una de las piezas ve de esta manera en la imagen se muestra con un corte transversal.

El cual será impreso en material PLA en una impresora 3D.



*Figura 3.6 Simulación del ensamble*

## Capítulo 4 Sensores y computadora de vuelo.

En este capítulo se describe cada uno de los sensores utilizados y la computadora de vuelo que nos ayudará con el procesamiento de los datos.

### 4.1 Sensores

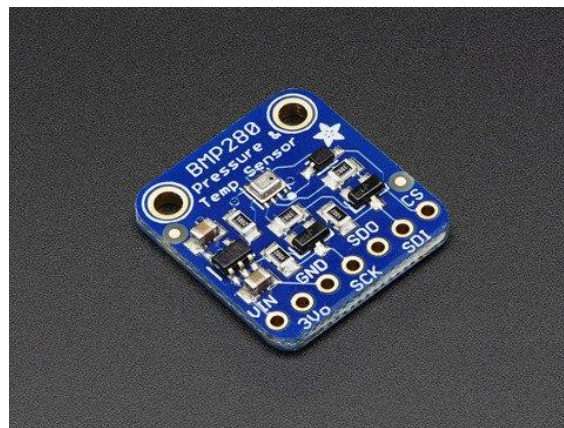
El sensor a utilizar en nuestro CanSat es acorde a la misión espacial CanSat de la UNAM en la

cual nos pide medir telemetría básica al igual que la aceleración y orientación del CanSat Durante todo su trayecto.

#### 4.1.1 BMP280

El BMP280 es un sensor ambiental con temperatura, presión barométrica la cual su comunicación puede ser por I2C o por SPI, el cual es creado por la empresa Bosch y cuenta con ruido de baja altitud de 0.25m y un tiempo de conversión rápida la cual se nos hace un sensor de calidad y cómo puede trabajar en 3.3v y 5v se nos hace ideal.

Cuenta con unas dimensiones de 19.2mm x 17.9mm x 2.9mm y un peso de 1.3g, los cuales son ideales ya que es casi como del tamaño de un centavo de dólar y el peso es muy pequeño.



*Figura 4.1 BMP280*

#### 4.1.2 MAG3110

El MAG3110 es un magnetómetro digital de 3 ejes pequeño y de baja potencia y el cual con un acelerómetro puede ser usado para tener información sobre brújula digital e independiente de su orientación, cuenta con una salida I2C estándar, puede funcionar con 3.3v y al igual que el BMP280 es del tamaño igual que un centavo de dólar.



Figura 4.2 MAG3110

#### 4.1.3 MMA 8451

El MMA 8485 es acelerómetro de 3 ejes de alta precisión, puede detectar movimiento, inclinación y orientación básica, además que este acelerómetro se comunica por protocolo I2C y la cual es perfecta ya que podemos unirlo con los demás sensores para crear nuestro propio IMU, este sensor puede ser usado en conjunto con el magnetómetro MAG3110.

Trabaja a un voltaje de 3.3v que nos cae de maravilla al igual que su peso es de 1.3g y unas dimensiones de 21mm x 18mm x 2mm la cual como los demás sensores ya mencionados son muy pequeños y de un peso similar.

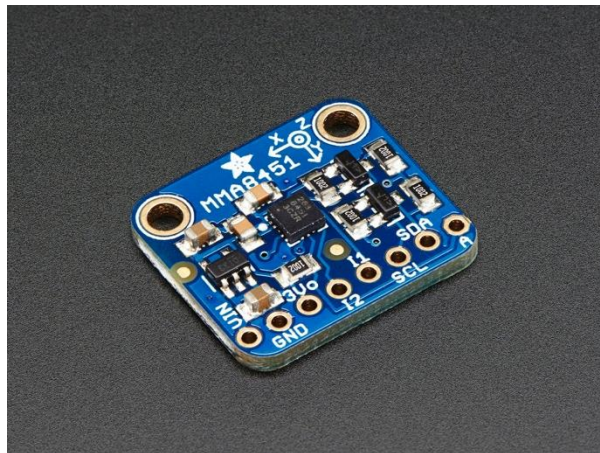


Figura 4.3 MMA8485

#### 4.1.4 GY-80

El GY-80 es un sensor donde incorpora acelerómetro, giroscopio, acelerómetro, brújula digital y de temperatura más que nada ya es un IMU fabricado en masa y el cual tiene un problema que arroja muchos datos con ruido y el cual utilizaremos este sensor es para sacar por separado cada

uno de sus sensores que lo incorporan para poder tener una buena medición y hacerle sus filtros requeridos, este sensor trabaja con 3.3v y 5v lo cual nos viene de maravilla y trabaja con direcciones I2C.



Figura 4.4 GY-80

## 4.2 GPS

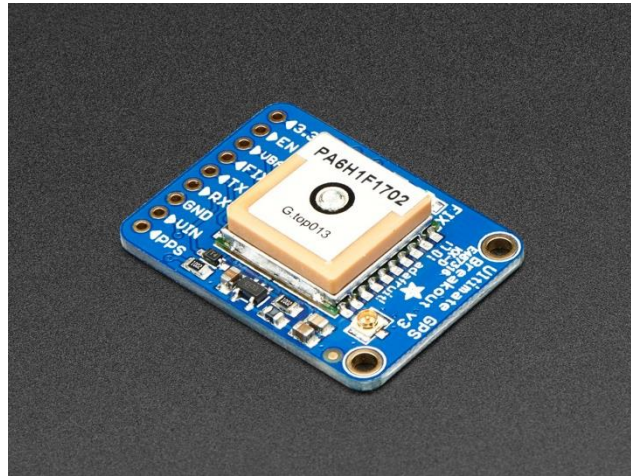
Se optó por utilizar un GPS para saber en dónde se encuentra nuestro CanSat después del lanzamiento la cual se utilizó un módulo GPS y con la ayuda del acelerómetro y el magnetómetro podemos hacer un trazado de punto a punto para saber cuánto se desplazó el CanSat desde su origen(lanzamiento) hasta el momento del aterrizaje y graficarlo con la ayuda de google maps.

### 4.2.1 Últimate GPS Adafruit

El Últimate GPS de la empresa Adafruit es un módulo súper pequeño de GPS que cuenta con las siguientes características.

- Trabaja a una frecuencia de 10Hz
- Alimentación por 5V.
- Batería compatible con RTC

El chip basado para el GPS es el MTK3339 es un módulo de alta calidad que puede rastrear hasta 22 satélites en 60 canales además de una antena incorporada, puede hacer hasta 10 actualizaciones por segundo de la ubicación, dado todo lo anterior utilizamos este GPS para nuestro set point que será el punto de referencia al GPS que se hizo con el magnetómetro y acelerómetro así con esto podremos estar monitoreando el CanSat



*Figura 4.5 GPS Ultimate por Adafruit*

### 4.3 Reloj RTC

Debido a la programación hecha en Python y que nuestra computadora de vuelo principal es una Raspberry pi y al momento de tratar de tomar un muestreo del tiempo desde que se ejecutó la orden hasta que la termine se saturaba el procesador ya que tomaba fechas de años atrás a la fecha así que se decidió optar por un reloj RTC.

#### 4.3.1 Tiny RTC

El Tiny RTC es un módulo DS1307 que consta de un reloj en tiempo real con la ayuda de una batería, el cual este módulo también cuenta con una memoria EEPROM y un socket para poder utilizar un sensor de temperatura que venga en encapsulado tipo transistor, el módulo trabaja con protocolo I2C y con una pequeña memoria RAM no volátil que nos ayudará a almacenar algunos datos durante el proceso, cuenta con una alimentación a 5v la cual se nos hace ideal para poder.



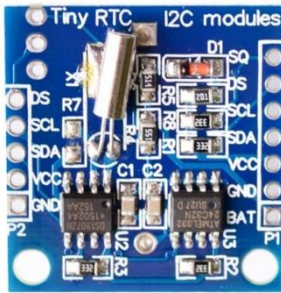


Figura 4.6 Reloj RTC

## 4.4 Telemetría

Para la telemetría seguimos los requisitos para el concurso nacional de la UNAM que nos pedían usar los módulos XBee S3B pro para poder mandar la información entre la estación terrena y el CanSat.

### 4.4.1 XBee pro S3B pro

El XBee pro S3B es un módulo de alta potencia que usado por antenas dipolo de 2.14dB puede alcanzar a enviar datos hasta 14.4km de distancia trabaja a una frecuencia de 928 MHz y una velocidad de transmisión de datos 200Kbps trabaja voltaje de operación de 3.3 que es ideal para el CanSat.



Figura 4.7 XBee pro S3B

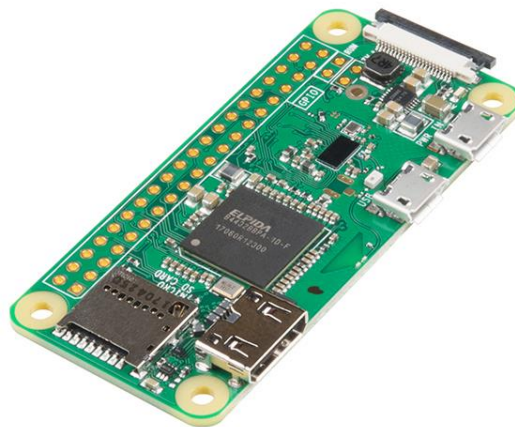
## 4.5 Computadora de vuelo

La computadora de vuelo que utilizaremos es una Raspberry pi Zero W ya que por su tamaño y procesador nos vendrá de maravilla para la construcción del CanSat y además utilizaremos un

Arduino Nano para la ayuda de nuestro setpoint para el GPS externo al CanSat al igual que este accionará un servomotor que acciona el paracaídas.

#### 4.5.1 Raspberry pi Zero W

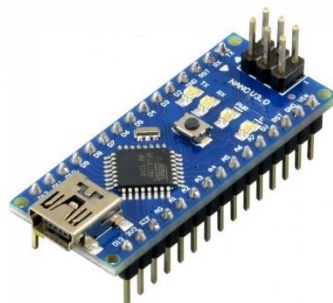
La Raspberry pi Zero W es la versión más pequeña de la pi 2+ la que cuenta con 40 pines GPIO programables al igual que esta versión ya implementa wifi y bluetooth para poderse conectar eso nos ayuda ya que no cuenta con una entrada LAN y todas nuestras actualizaciones las podemos hacer vía wifi otra ventaja es que cuenta con una SD extraíble la cual puede ser conectada en cualquier otra Raspberry para poder hacer configuraciones, cuenta con un procesador a un núcleo que tiene una velocidad de 1Ghz de velocidad y 512 MB de memoria RAM la cual nos hace tener una velocidad de lectura de los sensores muy rápida.



*Figura 4.8 Raspberry pi Zero W*

#### 4.5.2 Arduino Nano

El Arduino Nano es un microcontrolador pequeño y fácil de usar basado del AtMega 328P funciona igual que un Arduino DUE, pero en un paquete diferente y funciona con un cable de miniUSB y cuenta con 32Kb de memoria flash y una velocidad de reloj de 16 MHz 8 pines analógicos y 22 digitales 6 de los cuales son PWM, la cual nos la hace ideal para controlar un pequeño servomotor de 1 kg de torque y el GPS Ultimate de Adafruit.



*Figura 4.9 Arduino nano*

## 4.6 Reguladores de voltaje

Utilizaremos 2 reguladores de voltaje ya que la pila que alimenta todo el CanSat es de 9v y para la cual necesitamos dos reguladores de voltaje para poder hacer funcionar nuestros sensores que son de 3.3v y de 5v para ello había dos formas de hacerlo que es con divisores de voltaje o con módulos y fue por la cual nos inclinamos, los módulos son los AMS 117 para 3.3v y para 5v.

### 4.6.1 AMS 117 3.3v y 5v

El AMS 117 en cualquiera de sus 2 versiones ya sea 3.3v o 5v que es capaz de trabajar en la entrada del módulo desde 4v hasta 12v los cuales nos ayudaran a darle energía a nuestros sensores y computadora de vuelo.

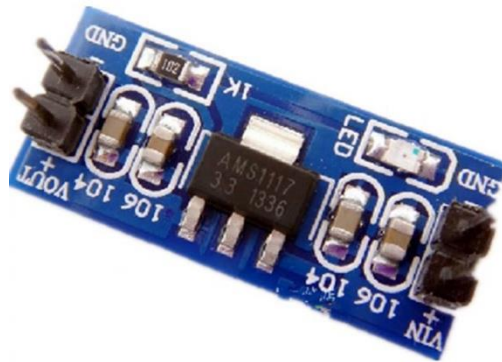


Figura 1 Ams117 modulo

## Capítulo 5 Construcción y programación del CanSat

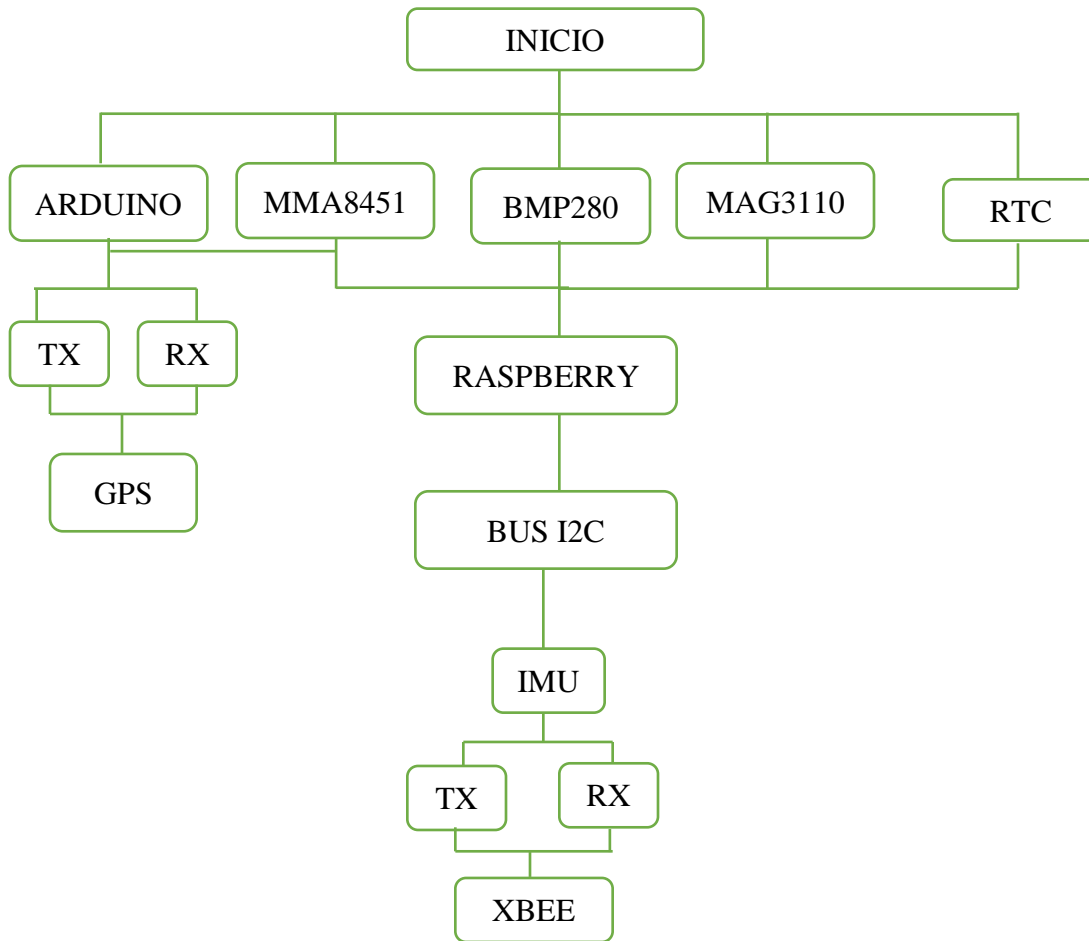
En el siguiente capítulo se describe el lenguaje de programación a utilizar al igual que la programación de los sensores y como se hizo un propio IMU con las calibraciones que se implementaron, al igual que también la impresión en 3D y el diseño, construcción del paracaídas de la misma manera el PCB y el diagrama esquemático.

### 5.1 Programación en los sensores seleccionados

En la programación de los sensores nos dimos a la tarea de programarlos desde el lenguaje Python para la cual es el lenguaje con el que comúnmente se programa la Raspberry pi la cual empezamos por el sensor BMP280 en el cual encontramos que su librería estaba bien calibrada a así que no tuvimos muchos problemas con ese sensor pero para los demás nos dimos cuenta que los valores tienen una variación muy grande y no quedaba que entrar a ver qué sensores componían el GY-80 y el MMA8485 se tuvieron que destripar y con la ayuda de su datasheet se pudo filtrar todos esos datos.

A Continuación, se dejan una pequeña parte de la programación hecha en Python no se decidió poner completa la programación por motivos de espacio.

Desde el siguiente link podrá ingresar al GitHub del equipo `Quantum Control donde se tiene las programaciones <https://github.com/Quantum-Control/Canasat-Pakal>.



```

import
smbus

    from time import sleep
    import math

    # select the correct i2c bus for this revision of Raspberry Pi
    revision = ([l[12:-1] for l in
open('/proc/cpuinfo','r').readlines() if
l[:8]=="Revision"]+['0000'])[0]
    bus = smbus.SMBus(1 if int(revision, 16) >= 4 else 0)

    # ADXL345 constants
    EARTH_GRAVITY_MS2 = 9.80665
    SCALE_MULTIPLIER = 0.004

    DATA_FORMAT = 0x31
    BW_RATE = 0x2C
    POWER_CTL = 0x2D

    BW_RATE_1600HZ = 0x0F
    BW_RATE_800HZ = 0x0E
    BW_RATE_400HZ = 0x0D
    BW_RATE_200HZ = 0x0C
    BW_RATE_100HZ = 0x0B
    BW_RATE_50HZ = 0x0A
    BW_RATE_25HZ = 0x09

    RANGE_2G = 0x00
    RANGE_4G = 0x01
    RANGE_8G = 0x02
    RANGE_16G = 0x03

from adxl345 import ADXL345
from L3G200D import L3G4200D
import smbus
import time

```

```

import math
from datetime import datetime
class IMU():
    def milis(self):
        date=datetime.now().time()
        seconds=(date.minute*60)+date.second
        miliss=seconds*1000
        return miliss

    def __init__(self):

        self.Acc=ADXL345()
        self.Gyro=L3G4200D()
        # self.Gyro._data_config_()

        self.Acc.Calibrate(1500)
        self.Gyro.Calibrate(1500)

        self.GyrosTheta=[0,0]
        self.GyrosThetaprev=[0,0]
        self.AccTheta=[0,0]

        self.Theta=[0,0]
        self.Thetaprev=[0,0]
        self.velocity=[0,0]
        self.t=datetime.now().time()
        #mliseconds to poweronrasp to actual
        self.timeprev=self.milis()
        self.dt=0

        self.isCalibrate=False

    def Calculate_Angle(self):
        self.Acc.Read_INTval()
        ax,ay,az=self.Acc.Get_IT()
        #ejes g\nuevs debido aa posicion en la pcb
        azz=ax
        ayy=ay
        axx=(-1*az)

        self.AccTheta[0]=(math.atan(axx/math.sqrt(pow(ayy,2)+pow(azz,2))))*(180/3.1416)
        self.AccTheta[1]=(math.atan(ayy/math.sqrt(pow(axx,2)+pow(azz,2))))*(180/3.1416)

        self.dt=self.milis()-self.timeprev
        self.timeprev=self.milis()

        self.Gyro.read_value()
        wx,wy,wz=self.Gyro.Get_value()

        wxx=wz*(-1)
        wyy=wy
        wzz=wx

        self.GyrosTheta[0]=(wxx*self.dt)/1000.0+self.GyrosThetaprev[0]
        self.GyrosTheta[1]=(wyy*self.dt)/1000.0+self.GyrosThetaprev[1]

        self.GyrosThetaprev[0]=self.GyrosTheta[0]
        self.GyrosThetaprev[1]=self.GyrosTheta[1]

        self.Theta[0]=0.94*(self.Thetaprev[0]+(wxx*self.dt)/1000.0)+0.06*(self.AccTheta[0])
        self.Theta[1]=0.94*(self.Thetaprev[1]+(wyy*self.dt)/1000.0)+0.06*(self.AccTheta[1])

```

```
        self.Thetaprev[0]=self.Theta[0]
        self.Thetaprev[1]=self.Theta[1]

    def Get_Angle(self):
        return self.Theta[0],self.Theta[1]

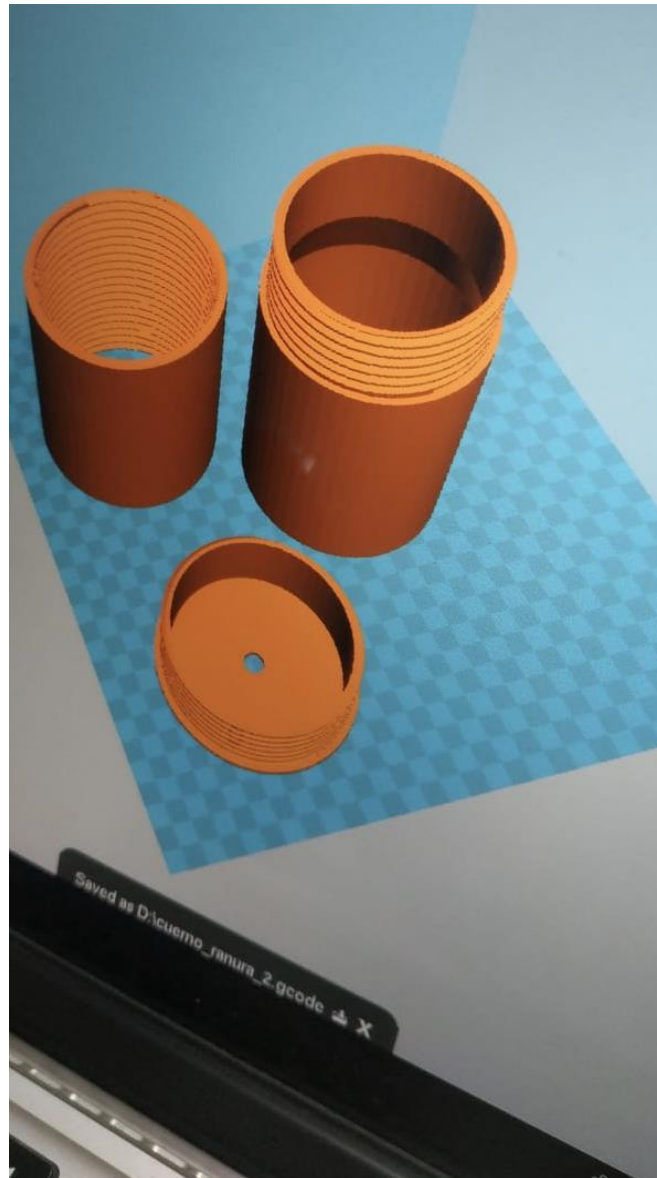
if __name__ == "__main__":
    # if run directly we'll just create an instance of the class and output
    # the curren
    #@ imu=IMU()

    # while 1:
    ####
        #print ("  xz = %.3fG" %XZ)
        #print ("  y = %.3fG" %YZ)
```

## 5.2 Impresión del diseño hecho en CAD

La impresión de diseño con el material PLA fue llevado a cabo en Tuxtla con la ayuda de Bruno Hernández el cual colaboró para el equipo y uno de las partes en donde colaboró fue en la impresión del envoltente.

Como se puede ver en la imagen siguiente (Figura 5.1).



*Figura 5.1 Visita previa del impreso en 3D*



### 5.3 Diseño del PCB y diagrama esquemático

El diseño de nuestro PCB va de mano con nuestro diagrama esquemático, para ello se dividió en cuatro partes como son la etapa de potencia, computadoras de vuelo, etapa de telemetría y sensores.

Se realizó una pequeña simulación en el software Proteus solo a la etapa de potencia para corroborar que este nos diera el voltaje requerido, después se optó por pasar el diagrama a una plataforma online donde se podrían hacer arreglos al diseño si se tenía una equivocación solo con abrir el navegador de cualquier tipo ya está ahí nuestro diseño a continuación se deja con la etapa de potencia.

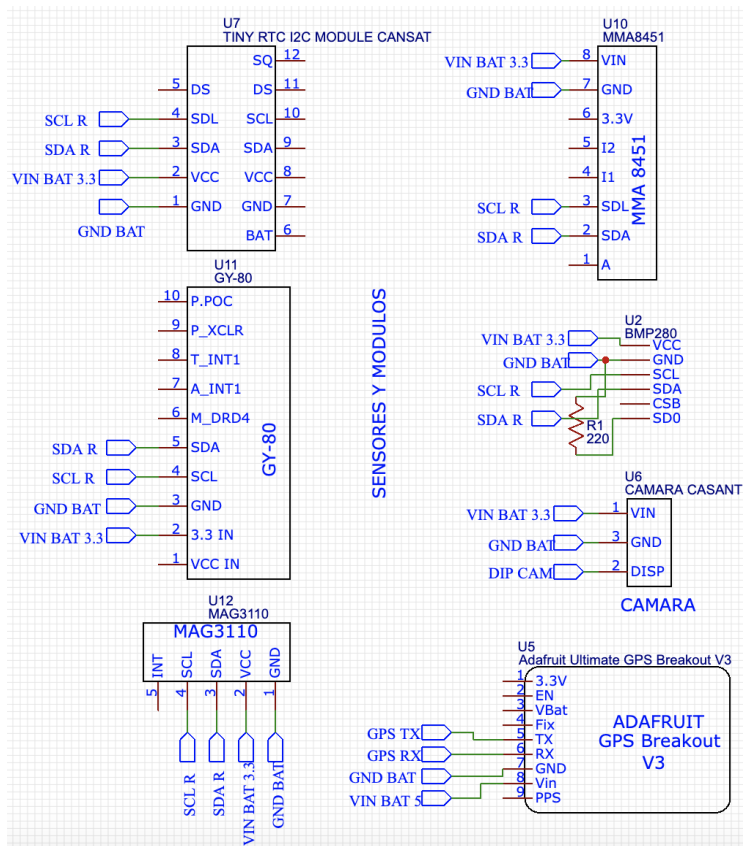


Figura 5.2 Sensores y modulos

Como se muestra en la imagen 5.2 se utilizaron puertos de lugar de bus comunes para así ahorrar espacio y tener mejor ordenado el diagrama esquemático.

Comenzamos que todo va partir de una batería de 9V en la cual está conectada a un dip switch de dos entradas y salidas para que una active nuestro regulador de 3.3v y el otro active el de 5v, cabe mencionar que la simulación hecha en proteus fue para ver si los módulos AMS117 con el arreglo de resistencias y capacitores que trae si nos daban el voltaje correcto.

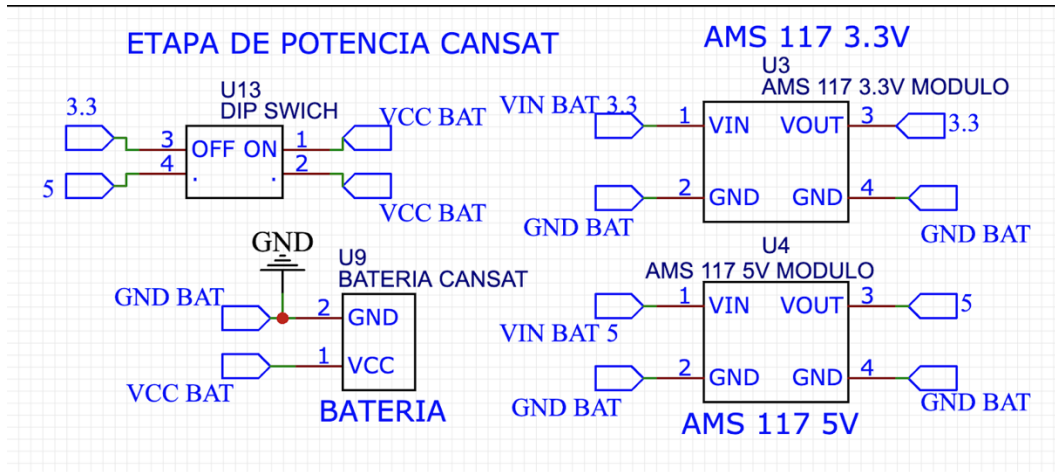


Figura 5.3 Etapa de potencia

Ahora pasamos a la etapa de nuestros sensores en las cuales por motivos del concurso nacional no se pudo implementar la cámara ya que el CanSat solo debería de llevar solo un orificio donde será liberado del dron y pues necesitábamos otro para la cámara y otro punto a rescatar es que todo se fue a los pines de I2C a si nos vamos ahorrar espacio para la creación de las pistas.

Todos los módulos pueden ser alimentados desde 3.3v así que solo reservamos el módulo de 5v para las computadoras de vuelo.

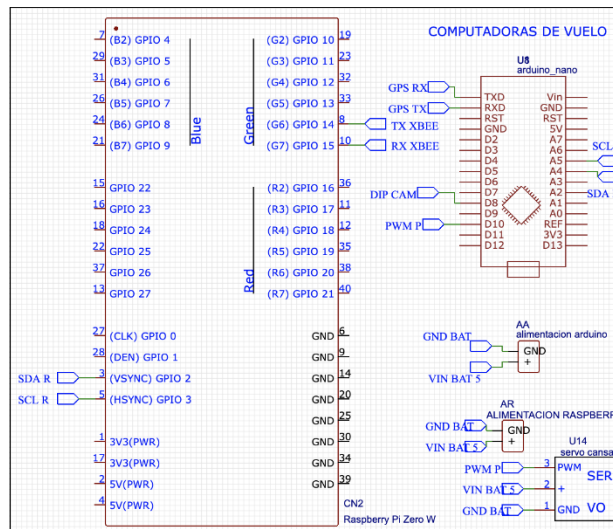


Figura 5.4 Computadoras de vuelo.

Se decidió utilizar un Arduino nano como soporte a la Raspberry para poder accionar el paracaídas y mandar los datos del GPS para compararlos con el magnetómetro y tener la distancia punto a punto.

El servomotor va a recibir la instrucción de cuando esté a la altura deseada este se accionará dejamos una muestra en video (<https://youtu.be/o5ndriUR9yo>) a si se muestra una pequeña simulación de cómo debe de funcionar durante la caída libre.

Como se puede observar en la figura xx solo las alimentaciones de 5v son para las computadoras de vuelo y el servo motor.

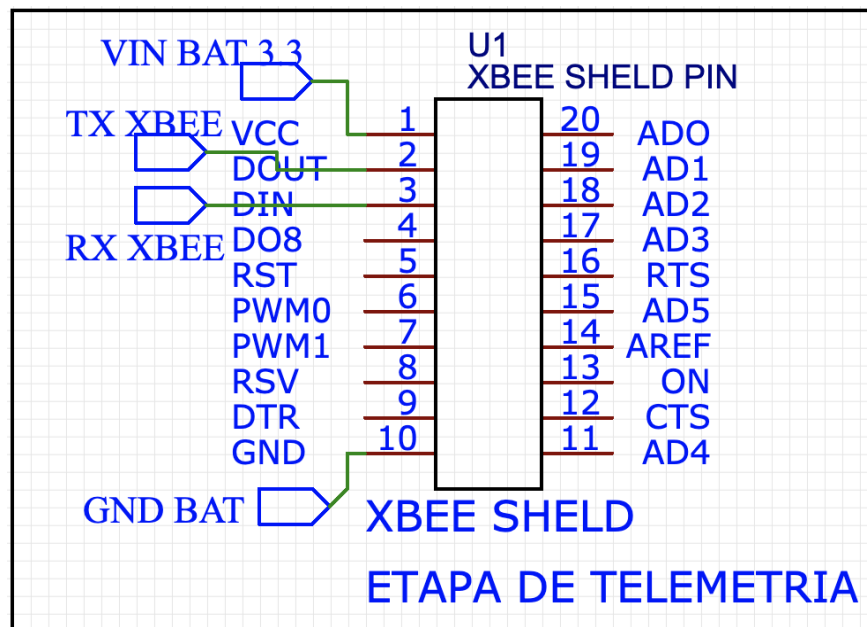


Figura 5.5 Etapa de telemetría.

Para la etapa de telemetría se usó la XBee pro S3B la cual la lleva a bordo una antena de 980 MHz la cual nos va a mandar la telemetría de todos los sensores a la computadora el cual tiene como receptor otro XBee pro S3B.

Para el diseño del PCB se optó hacerlo de doble capa y la cual tiene un radio de 9mm lo que lo hace ideal para poder introducir a la parte de la electrónica del CanSat.

Como se muestra en la figura 5.6 el circuito se logró incorporar en una sola pcb de doble cara, cabe mencionar que el Programa Espacial Universitario proporcionó los componentes más difíciles de conseguir y caros incluyendo las XBee.

El diagrama esquemático completo queda de la siguiente manera de igual manera se encuentran en esta página de internet (<https://easyeda.com/editor#id=|2343104236b94780afc030c405566a50>) en este link encontrarán el diseño del

PCB(<https://easyeda.com/editor#id=|d73dbad02f8d407fb7de0df722137d4e>).

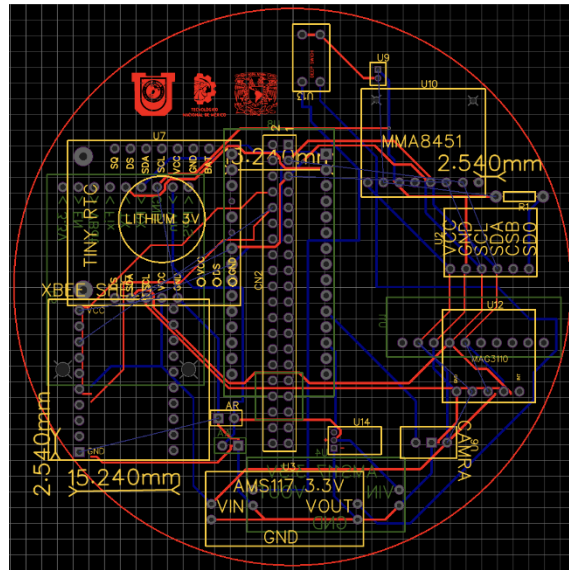


Figura 5.6 Diseño de PCB

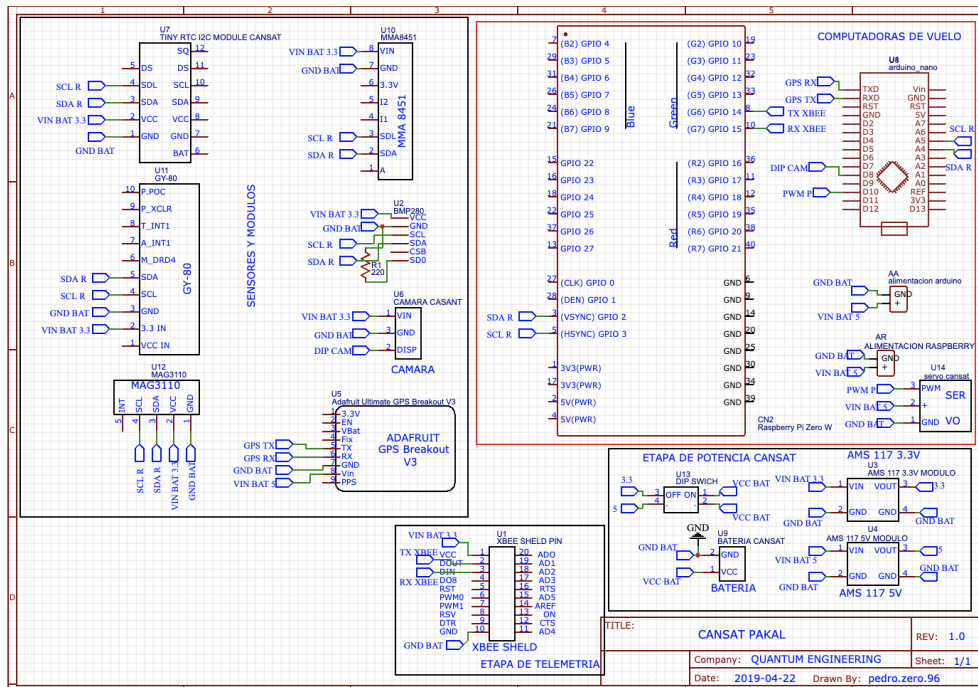


Figura 5.7 Diagrama general del Esquemático

#### 5.4 Construcción del paracaídas

Para la construcción de nuestro paracaídas dado los datos ya obtenidos con los cálculos nos dispusimos buscar una tela que fuera de estilo o tratamiento de nylon para que fuera un poco elástica y que el viento no lo traspasara, se consiguió una tela con esas características y se cortó al diámetro de 12 cm ya que nos dimos cuenta como en los cálculos hechos para

las diferentes temperaturas, una vez cortada la tela se pusieron 6 puntos de tensión donde con la ayuda de hilo de nylon o “omega” como se muestra en la figura 3.8 y se costuro en la tela.

El proceso fue realmente sencillo el cual nomas se cortó un trozo de hilo con el radio del paracaídas para luego en la punta ir marcando la circunferencia, una vez marcada con un plumón la circunferencia del paracaídas se corta para posteriormente hacer una seña cada 60° para poder costura los hilos del paracaídas.



*Figura 5.8 Hilo de nylon*



*Figura 5.9 Tela de paracaídas*

El proceso fue realmente sencillo el cual nomas se cortó un trozo de hilo con el radio del paracaídas para luego en la punta ir marcando la circunferencia, una vez marcada con un plumón la circunferencia del paracaídas se corta para posteriormente hacer una seña cada 60° para poder costura los hilos del paracaídas.

### 5.5 Acoplamiento final del CanSat

Una vez ya teniendo la impresión y las placas ya soldadas con sus respectivos componentes nos disponemos a ensamblar cada uno de los componentes que conformarán el CanSat primero nos disponemos a soldar los componentes a la PCB.

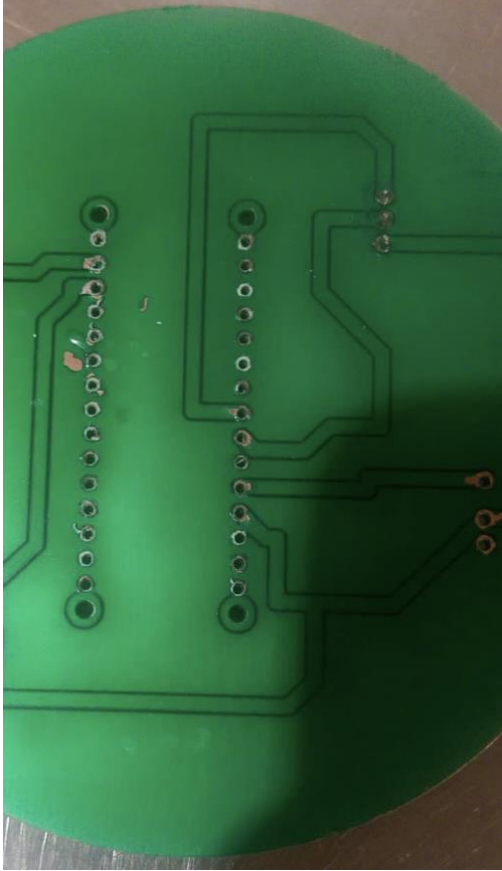
Antes de soldar nuestros componentes cortamos la PCB a la medida y checamos que en efecto entrara y con la ayuda de la rosca interna del CanSat la PCB se mantuviera fija en esa posición.



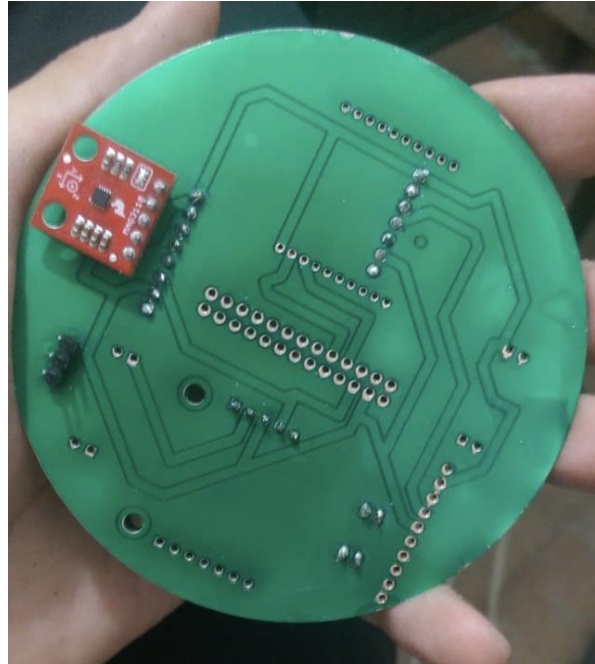
*Figura 5.10 Acoplamiento PCB a rosca interna*

una vez que nos percatamos que queda perfecto nos dispusimos aplicar mascara anisodonte color verde para tener un acabado más profesional y nos ayudará a que nuestra soldadura sea perfecta de igual manera.

Una vez obtenida la PCB con su máscara anisodonte como se muestra en la figura 5.12 nos dispusimos a soldar cada uno de los componentes, primero se buscó la manera de cuales hay que ir soldando primero ya que la placa no cuenta con through holes y se tiene que cuidar al momento de soldar.



*Figura 5.11 PCB Arduino*



*Figura 5.12 PCB subsistemas*

Una vez ya soldados los componentes queda de la siguiente manera mostrado en la figura 5.13, ya que tenemos soldados los componentes y nos percatamos que encaja perfecto en la parte contenedora de la electrónica del CanSat nos disponemos a acoplar el material tipo esponja llamado “oasis” el cual en este caso nos ayuda a mantener a salvo al tripulante que es un huevo de gallina.

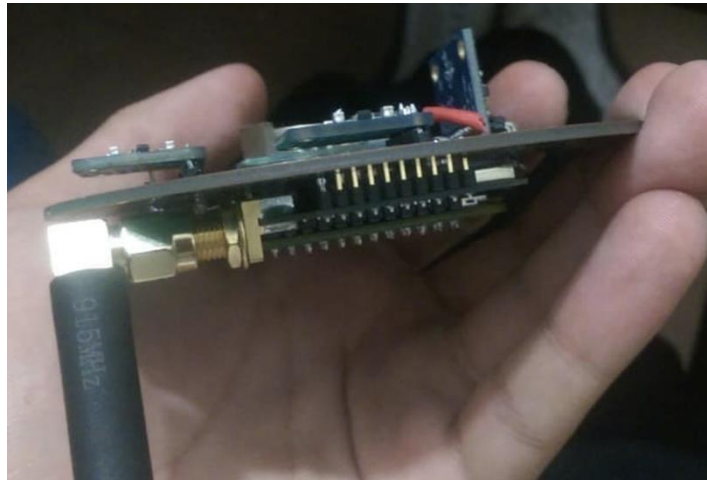
Como se muestra en la figura 5.14 el material llamado oasis se le hizo un hueco del tamaño del huevo para que al momento del impacto este actúe como su amortiguación gracias a que este material cuando se deja caer sufre una deformación que se convierte en un material muy blando que creemos que nos vendrá de maravilla al CanSat.



*Figura 5.13 Acoplamiento ya soldado*



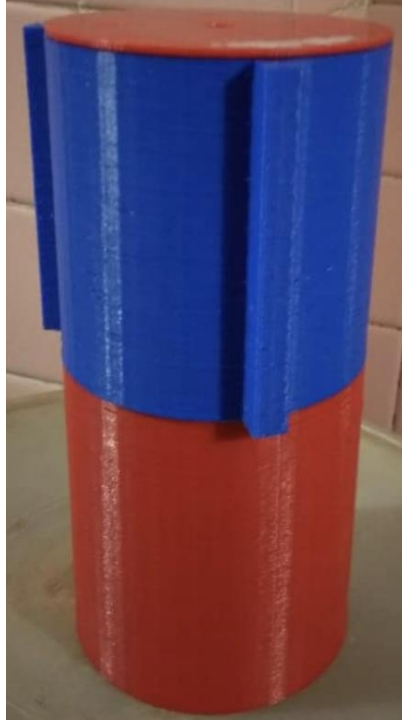
*Figura 5.14 Oasis y el huevo*



*Figura 5.15 Vista posterior de acoplamiento.*

Una vez ensamblado todo el satélite queda de la siguiente manera la cual cumple con los requisitos de la misión del concurso nacional.





*Figura 5.16 CanSat listo para la misión.*

Como se muestran en la figura 5.16 el CanSat está listo para el lanzamiento en el cual se encuentra integrado cada uno de los subsistemas del CanSat.

## Capítulo 6 Pruebas y día del lanzamiento

En este capítulo se describen la realización de una prueba que se llevó a cabo días antes del lanzamiento en este se muestra la distancia de transmisión del XBee S3B, también describimos el día del lanzamiento del CanSat el cual fue en el estadio Tapatio de C.U en el cual el CanSat se dejó caer de una altura de 400m.

### 6.1 Pruebas a distancia de telemetría

La prueba se realizó el día 3 de mayo en el cual aun con los componentes en protoboard se hizo una prueba de telemetría la cual teníamos que transmitir a más de 200m lo cual como no contábamos con un dron para poder subir los componentes a esa altura.

Gracias al google maps se puede medir cómo sería una distancia de 200m y la cual nos mostró que era la distancia de dos cuerdas para esta prueba se dejó corriendo el main para después con la computadora ir caminando y checar la potencia de nuestras antenas, en la cual nos dimos cuenta que los XBee pro S3B son muy potentes porque gracias con las antenas de 914 MHz nos transmite los datos a la perfección a esa distancia pero al momento de caminar más estos se empezaban a retrasar es decir los paquetes de datos recibidos empezaron ser más lentos y eso se supone que ya

estaba dando lo máximo las antenas y con unas más grandes podíamos recibir los datos sin problemas a los 400 m.

## 6.2 Día del lanzamiento

El día 25 de mayo se citaron a los equipos que llegaron a la 6ta etapa del concurso nacional a las 9 am en el estadio “El Tapatío” Méndez en el cual al llegar se nos entregó un formulario el cual se rellenó y teníamos 1h para poner a punto nuestro satélite.

Después de tener todo listo pasamos a que los jueces aprobaran nuestro diseño el cual cumplió con las especificaciones establecidas por el concurso como son.

- Altura no mayor a 22 cm
- Diámetro no mayor a 11 cm
- Agujero en la parte superior no mayor a 0.8 cm

El único requerimiento que no se cumplió y que nos pasamos fue la masa del satélite por 5g por arriba de lo establecido ya que peso 505g.

Como se muestra en la figura 6.1. El CanSat debido a que se pasó por encima de 5g (gramos) fue aceptado para lanzamiento, en la cual de finalizar la inspección por parte de los jueces se dieron 30 min para terminar de acoplar el satélite en el cual no fue necesario ya que estábamos listos para el lanzamiento.



Figura 2 CanSat listos para la misión espacial.

Tras pasar los primeros 9 y cómo observar que solo un CanSat libero paracaídas y salvo a su tripulante empezamos a ver cuáles son nuestras ventajas y desventajas ante esos equipos, llegó el

momento de lanzar nuestro CanSat así que se colocó en el Dron para que éste lo elevará a 400 metros de altura.



*Figura 6.2 CanSat después del impacto de 400m*

Durante el transcurso de la caída libre gracias a que le colocamos unos soportes a cada  $120^\circ$  lo cual los primeros 150 de caída libre se controló el autogiro, pero al momento que desplegó el paracaídas fue un éxito ya que el mecanismo propuesto funciono a la perfección y se pudo liberar, pero los hilos se soltaron a unos metros más así que el CanSat sufrió una caída de 400 m donde el paracaídas funciono, pero se soltó por que sus hilos no aguantaron.

Comparando nuestro diseño con los demás equipos nuestro diseño quedo intacto solo recibió unos cuantos rasguños, pero fue sorprendente que soportó una caída de 400m.

## Capítulo 7 Resultados

En este capítulo se redactaron todos los resultados obtenidos durante la prueba días antes del lanzamiento y de igual manera se adjuntan los resultados del día de lanzamiento donde mostraremos las gráficas y el estado del CanSat después del impacto.

## 7.1 Resultados en pruebas

Los resultados en pruebas que obtuvimos fueron que las antenas que solo nos transmiten a 200 metros efectivos porque después de esos la transmisión de datos se empieza a tener errores y es más lenta.

obtuvimos como iba a hacer la trama de cómo el dato se envía de XBee a XBee.

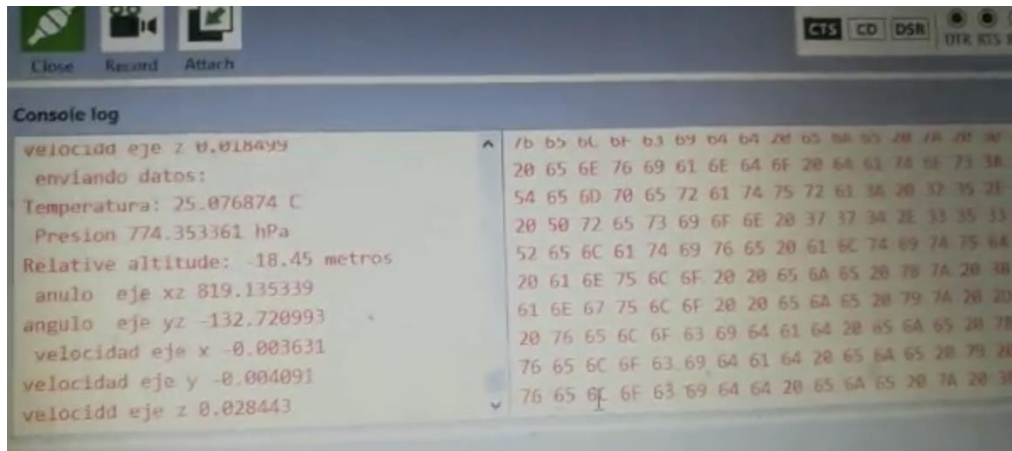


Figura 3.1 Telemetría enviada por XBee

Una vez obtenidos estos datos pues ya sabemos cómo poder graficarlos y que lo guarden en el archivo de Excel.

obtuvimos pocos resultados a comparación al día del lanzamiento.

## 7.2 Resultados del día de lanzamiento

El día del lanzamiento nos arroja muchos resultados los cuales van desde nuevas ideas hasta confirmaciones de predicciones hechas para hacer un lanzamiento perfecto.

Gracias a la Raspberry pi Zero w el procesamiento de la información de los sensores fue muy buena y rápida otra cosa a rescatar es que quedo muy precisa con valores reales como se muestra en la figura 7.2 podemos ver la asignación de los puertos I2C

```

pi@raspberrypi:~$ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50: 50  --  --  53  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  68 69  --  --  --  --  --  --  --
    --  --  --  76 77
pi@raspberrypi:~$

```

Figura 7.2 Direcciones I2C de los sensores

Una vez checando que todo esté en orden en nuestras direcciones de nuestros sensores y reloj empezamos con la recopilación de datos mientras el dron eleva el CanSat como se muestra en la siguiente figura 7.3.

```

('Eje x ACAC', 2.8313001530559236, 'Eje y:', 87.25296999999995)
('Eje x GYR', -3.6334900000000002, 'Eje y:', -87.16622340650419)
('Eje x ACAC', 2.752135936545692, 'Eje y:', 86.99361999999995)
('Eje x GYR', -3.58960000000000017, 'Eje y:', -85.98502473277519)
('Eje x ACAC', 3.4881976751751065, 'Eje y:', 85.43751999999995)
('Eje x GYR', -3.33382000000000016, 'Eje y:', -85.88913729650127)
('Eje x ACAC', 0.8451984608175822, 'Eje y:', 82.53867999999994)
('Eje x GYR', -2.96275000000000015, 'Eje y:', -85.68618857371452)
('Eje x ACAC', 1.4217040928500582, 'Eje y:', 76.37013999999994)
('Eje x GYR', -3.6889300000000002, 'Eje y:', -87.98898990026294)

```

Figura 7.3 Recopilación de datos sensores

Todos los datos se fueron guardando en un archivo de Excel en los cuales fueron 1161 datos durante todo el tiempo de la misión en las cuales se midieron estos parámetros.

- Temperatura
- Aceleración
- Los grados de inclinación sobre los tres ejes (x, y, z)
- Presión

a continuación, se dejan con los gráficos obtenidos de los datos recopilados.

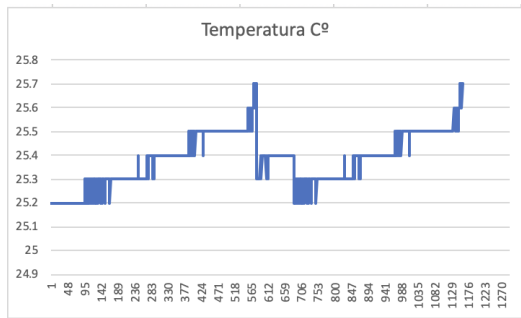


Figura 7.4 Gráfico de temperatura

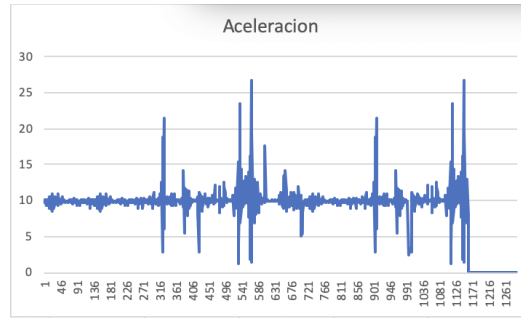


Figura 7.5 Gráfico de la aceleración

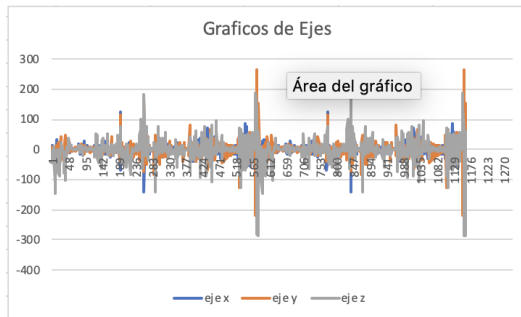


Figura 7.6 Gráfico de los ejes

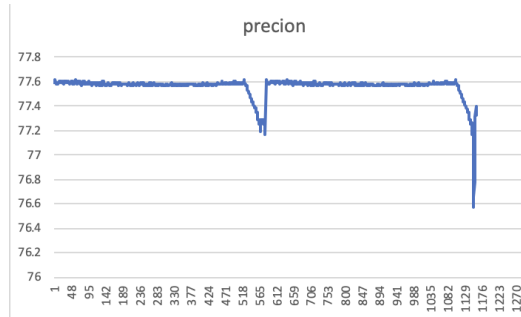


Figura 7.7 Gráfico de la presión

en la cual en la figura 7.4 nos muestra la temperatura la que vemos que se mantuvo entre los 25.2° y los 25.7° y en los grados de los ejes por la orientación del acelerómetro y magnetómetro el eje “x” casi se mantuvo a “0” grados mientras en los demás ejes vemos una variación causada por el autogiro provocado por el desprendimiento de los hilos del paracaídas, la presión atmosférica se mantuvo entre los 77.6 y 76.6 milímetros dados por el barómetro y en la aceleración vimos que se mantuvo entre 10 m/s y 15 m/s ya que vimos que tiene unos picos donde nos midió 25 m/s eso no quiere decir que tenemos que poner mejores filtros para obtener la aceleración bien al igual que después esta se da a 0 ya que fue en el momento donde impactó con la superficie.

Y así como se muestra en la figura 7.8 los datos graficados desde el archivo Excel creado después de cerrar comunicaciones y apagar el CanSat.

Temperatura Cº	Aceleracion	eje x	eje y	eje z	precion
25.2	10.04	-2.13	-6.5	-43.42	77.59
25.2	9.79	-2.31	-9.69	-32.04	77.59
25.2	9.85	14.5	6.66	-6.29	77.59
25.2	9.86	-12	-5.34	7.68	77.59
25.2	9.32	-6.34	-19.09	-18.25	77.61
25.2	9.34	-12.55	-22.5	9.37	77.58
25.2	10.21	-24.88	-20.18	-35.42	77.58
25.2	10.13	-17.44	-5.03	-109.23	77.58
25.2	10.06	-59.93	-5.74	-146.05	77.59
25.2	9.24	-58.92	2.2	-122.72	77.58
25.2	9.8	-32.45	7.76	-74.73	77.58
25.2	9.44	26.01	-20.45	-24.95	77.59
25.2	10	-32.85	12.15	-0.37	77.58
25.2	8.98	-30.26	1.79	-48.76	77.6
25.2	10.49	32.12	-3.89	-11.35	77.59
25.2	9.65	-15.37	-6.25	-2.3	77.59
25.2	9.33	10.53	-0.14	-5.45	77.6
25.2	10.56	-5.99	0.55	-5.7	77.6

Figura 7.8 Datos guardados en Excel

## Capítulo 8 Conclusiones

### 8.1 Conclusiones del diseño del CanSat

En conclusión con el diseño planteado fue muy bueno ya que los jueces al momento de calificar el satélite después de la caída libre de 400m quedó intacto y el cual sorprendió a los jueces, las etapas de potencia, telemetría, computadoras de vuelo y sensores se mantuvieron intactos excepto por la Raspberry pi que en el pin donde se alimenta se dañó pero nada que se pueda reparar, nos queda una gran experiencia para poder aplicar para una siguiente competición como es la colocación en mejor manera de los componentes electrónicos, tratar de jugar más por medio del control a nuestros sensores para que estos estén trabajando con datos reales y no sean datos arrojados en bruto, gracias a las aspas cada 120° puestas se pudo cortar el autogiro los primeros 200 m, debemos de buscar un nuevo sujetador de los hilos del paracaídas para poder asegurar el descenso controlado.

En fin, fue una experiencia que todo esto se llevó a cabo en seis meses para poder hacerlo se tocaron temas de física y aplicamos los conocimientos de electrónica se pudo llevar a cabo.

### 8.2 Conclusiones generales.

En conclusión a partir de todo lo aprendido en cada etapa del concurso nacional igual manera durante la estancia en el Programa Espacial Universitario ayudando con una base de datos para los proyectos espaciales que se encontraban actualmente en proceso, una de las complicaciones que se obtuvieron fueron que se tardó demasiado tiempo en hacer las placas PCB ya sea que se quemaban las placas o pasaba alguna cosa externa, se aprendió nuevos conceptos de física y con los conocimientos adquiridos durante el proyecto integrador nos ayudó para organizar el proyecto, los conocimientos adquiridos durante el día del lanzamiento fueron buenos ya que la idea es llevar un pequeño concurso a la escuela para poder sacar los próximos representantes para el siguiente concurso y en el cual ya se tienen ideas de cómo diseñarlo además de contar con las carreras afines para armar un buen satélite se planea implementar eso.

En el concurso en el cual se participó y se inscribieron 97 equipos cuales 21 llegaron al lanzamiento y conforme al método de calificar de los jueces aún estamos a la espera de los resultados finales esperando que nos situemos en un buen lugar para motivarnos a ir el siguiente año ahora con más experiencia.

Espero esta residencia habrá una brecha para poder hacer un poco de investigación espacial en la escuela y poder compartir a manera nacional con otros estados ya que conlleva un ritmo de trabajo algo alto y mucho trabajo en equipo.