



**INSTITUTO TECNOLÓGICO SUPERIOR
DE ATLIXCO**

Organismo Público Descentralizado del Gobierno del Estado de Puebla

**DESARROLLO DE UN SISTEMA DE COMUNICACIÓN
INALÁMBRICA PARA CANSAT Y UNA ESTACIÓN
TERRESTRE.**

OPCIÓN I.

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN MECATRONICA

PRESENTA:

OSCAR GALLEGOS LOPEZ

ASESOR: DR. FERNADO SANCHEZ TESIS

ATLIXCO, PUE. JUNIO DE 2019

Introducción

La agencia espacial europea define a los satélites CanSat como un conjunto de sistemas electrónicos que simulan las funciones de un satélite de tamaño real, integrado dentro de un volumen y forma de una lata de refresco. Entre las funciones que pueden desempeñar estos aparatos (por ejemplo, realizar mediciones meteorológicas a grandes alturas)

Los CanSats ofrecen una oportunidad única para estudiantes de tener una primera experiencia muy práctica de un proyecto espacial real; ya que ellos son responsables de todos los aspectos de diseño: diseñar el Cansat, seleccionar su objetivo o misión, integración de cada componente, pruebas en tierra, preparaciones de lanzamiento y análisis de los datos recibidos.

Estos acercamientos a la tecnología espacial entre los jóvenes son muy importantes ya que no solo aprenden acerca de cuestiones técnicas de un satélite sino también crea un interés sobre la tecnología aeroespacial, electrónica y en las ciencias como en física, matemáticas, etc., que pueden llevar a la generación de nuevo conocimiento científico e innovaciones tecnológicas.

Por lo tanto, la motivación para realizar trabajos de diseño y desarrollo de Cansat como herramienta didáctica toma una mayor relevancia puesto que es para inspirar a grupos de jóvenes al estudio de las ciencias aeroespaciales y fomentar el crecimiento tecnológico en nuestro país. En este documento en particular se muestra el desarrollo del sistema de comunicación del Cansat hacia la estación terrestre.

Objetivos

Desarrollar un sistema de comunicación inalámbrica entre el pico-satélite y la estación terrestre.

Los objetivos específicos son:

1. Elegir el tipo de comunicación.
2. Desarrollar el protocolo e interfaz de transmisión de datos.
- 3.-Implementar un algoritmo que aproxime los datos obtenidos de los sensores del Cansat.
- 4.-Probar el sistema de comunicación.
- 5.-Validacion de datos de Cansat.

Justificación

Los sistemas de comunicación de los Cansat la mayoría de los diseños ocupan como sistema de comunicación el Xbee.

También existen los llamados módulos Xbee PRO de la Serie 1 que se diferencian en la capacidad de alcance, permitiendo en algunos casos doblar la distancia de transmisión, ya que poseen una mayor potencia en la señal. Con los módulos Xbee PRO de la Serie 2, es posible crear redes más complejas, como las llamadas MESH. Estas permiten acceder a un punto remoto, utilizando módulos intermedios para llegar como routers. Además, los módulos automáticamente generaran la red entre ellos, sin intervención humana alguna, permitiendo la reparación de la red en caso de que algún nodo falle. Al mismo tiempo la red por sí sola resuelve la mejor ruta para un determinado paquete.

Para mejorar el sistema de comunicación en particular en este trabajo utilizaremos otro dispositivo que es LoRa Ra-02 el cual es una mejor alternativa para poder lograr una mejor comunicación en el Cansat ya que los dispositivos como la Xbee no logran tener una mejor eficiencia en:

El módulo Ra-02 LoRa está fabricado por la empresa Ai-Thinker, el fabricante del módulo ESP32S, el módulo utiliza SX1278 IC, funciona a una frecuencia de 433MHz. Es un módulo hecho de alta calidad. La dimensión es muy pequeña, sólo 17 x 16 mm, con todos los broches SX1278 ruptura

Los transceptores SX1287 cuentan con el módem de largo alcance LoRaTM que proporciona una comunicación de espectro extendido de gama ultra larga y una inmunidad de alta interferencia mientras minimiza el consumo de corriente. La alta sensibilidad combinada con el amplificador de potencia integrado de +20 dBm genera un presupuesto de enlace líder en la industria lo que lo hace óptimo para cualquier aplicación que requiera rango o robustez. LoRa también proporciona ventajas significativas tanto en el bloqueo como en la selectividad respecto a las técnicas de modulación convencionales, resolviendo el compromiso de diseño tradicional entre el alcance, la inmunidad a las interferencias y el consumo de energía. Estos dispositivos también admiten modos de FSK de alto rendimiento (G) para sistemas que incluyen WMBus, IEEE802.15.4g. El SX1287 ofrece excepcional ruido de fase, selectividad, linealidad del receptor y IIP3 para un consumo de corriente significativamente menor que los dispositivos de la competencia.

Marco Teórico

Modelos de satélites CanSat

Los CanSat son prototipos de modelos de satélites que se utilizan principalmente como introducción a la tecnología espacial; tienen un volumen de 350 ml (Véase Fig. 1), este volumen coincide con el de una lata de refresco, de ahí que su nombre se derive de la combinación del inglés Can Satélite. Éstos son lanzados a determinada altura (desde cientos de metros, hasta 4 km) mediante un cohete, un globo sonda o un helicóptero a control remoto; y transmiten información de telemetría mientras descienden



Figura 1.- Prototipo de Cansat UNAM

Los CanSats no orbitan la Tierra como un satélite real, pero simulan funciones y sistemas similares a estos. Cuentan con un subsistema de alimentación eléctrica; sistema de sensores; unidad de procesamiento con software de vuelo; y un sistema de comunicación inalámbrica. En la Fig. 2 se muestra un diagrama de bloques de un CanSat que transmite a una estación terrena los datos que lee de los sensores y del sistema de posicionamiento global (GPS). Los sensores que se utilizan más frecuentemente, sin estar limitados a ellos, son: presión, altitud, temperatura, acelerómetros, giroscopios y humedad relativa.

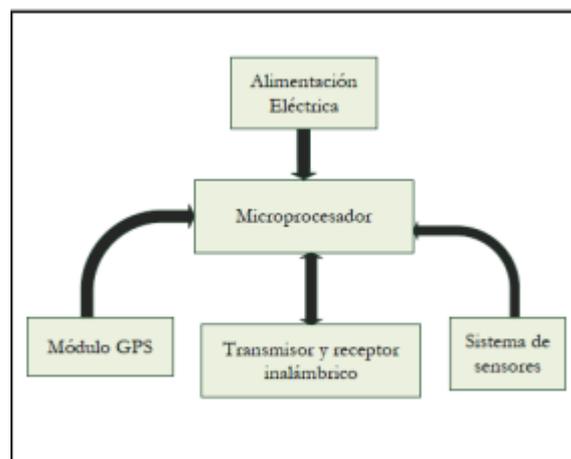


Figura 2.- Diagrama de composición de un Cansat

Es común que contengan un GPS; memoria para almacenar datos relevantes; y un paracaídas para su recuperación, lo que los hace sistemas reutilizables. Además de los sistemas mencionados, se necesita de una estación terrena, ésta recibe los datos enviados desde el CanSat, los grafica en tiempo real y almacena para su análisis posterior. La estación terrena consiste de una Interfaz Gráfica de Usuario (GUI) que permita al usuario visualizar los datos leídos por el CanSat; se puede realizar en los programas: LabVIEW, Matlab, o Visual Studio, por mencionar algunos ejemplos. En la Fig. 3 se muestra un ejemplo de GUI que grafica la altitud (en metros sobre el nivel del mar) del CanSat en tiempo real, además de mostrar otros datos como: número de paquetes enviados, tiempo de la misión en segundos y la presión barométrica en Pascales.

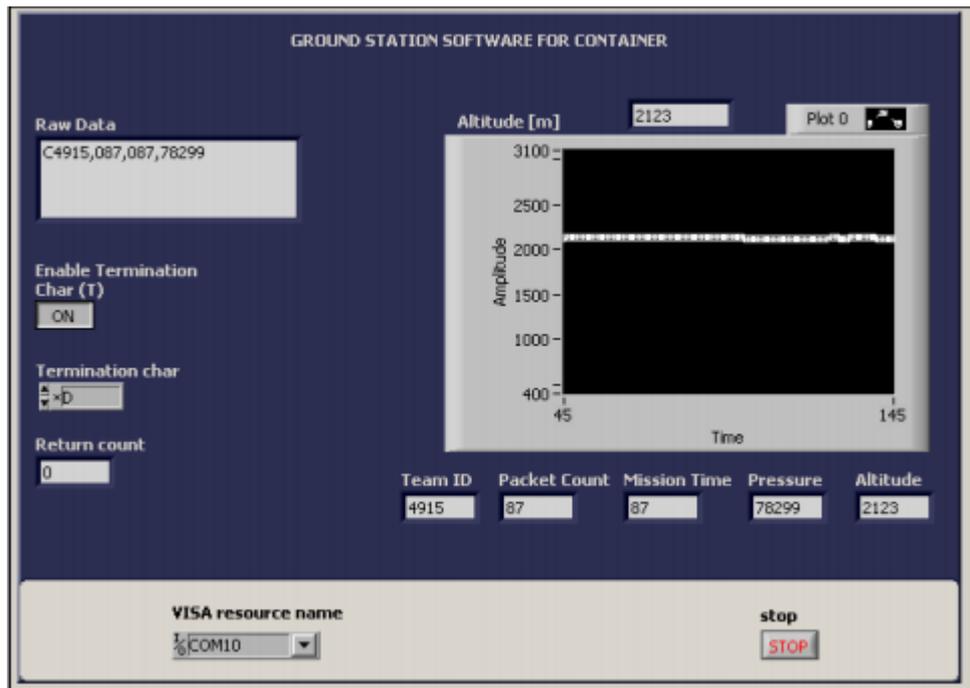


Figura 3.- Interfaz de estación Terrena

Elementos de un CanSat

Todos los cansats son un poco diferentes ya que se diseñan de acuerdo a la misión que tengan que completar, pero todos se componen básicamente de los siguientes elementos:

Sensores: Un sensor un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Existe una gran variedad de sensores pero en el cansat se utilizan principalmente aquellos que le permitan obtener información de los parámetros de su entorno como la temperatura, presión atmosférica, humedad, ubicación, etc.

Sistema de transmisión de datos: Este sistema sirve para que el cansat pueda comunicarse con una estación terrena, de tal forma que pueda enviar a esta la información recolectada en tiempo real. La opción más popular para realizar

esta tarea es utilizar módulos de radiofrecuencia. La radiocomunicación se hace por medio de ondas radioeléctricas. La radiofrecuencia es el conjunto de frecuencias situadas entre 3Hz y 300GHz.

Sistema de control: Es el cerebro del cansat, es responsable de recibir las señales de los sensores y les manda instrucciones para que actúen de acuerdo a la programación hecha. La mayoría de los microcontroladores tienen capacidad de almacenar cierto número de datos, que es útil para almacenar datos de los sensores durante el vuelo.

Fuente de alimentación: Esta es la fuente de alimentación de todos los sistemas dentro del CANSAT, y es indispensable para que los componentes funcionen correctamente. Las que se utilizan más comúnmente son las de Polímero de Litio, es importante que no sean muy pesadas y que sean lo más eficiente posibles.

Estructura: Podemos decir que la estructura del Cansat se divide en dos partes:

- **Interna:** En ella se sujetan los componentes electrónicos del cansat. Podemos encontrar modelos muy variados que van desde placas de aluminio donde se atornillan las tarjetas hasta postes de metal que sirven para espaciar las tarjetas cuando estas se van a colocar una sobre otra.
- **Externa:** Es la encargada de proteger los componentes del Cansat durante las pruebas, principalmente al impactar con el suelo. Se fabrican de metal pero también las hay de cartón o incluso plástico.

Estación terrena: Es la que recibe los datos recolectados por el Cansat y muestra al usuario final la evolución de ellos en el tiempo. Se compone principalmente por una computadora y un módulo de radiofrecuencia (mediante el cual se recibe la información enviada por el CANSAT)

Que es la comunicación

Hasta el momento se ha explicado qué es y cómo está construido un satélite, sin embargo, para que este pueda lograr el fin por el que fue construido, es necesario establecer un enlace con una estación en tierra conocida como estación terrena, y tiene el objetivo de permitir una comunicación directa con el pico-satélite para mandar o recibir información.

Estas estaciones consisten de una serie de equipos interconectados entre sí con una antena o un conjunto de antenas, que puede tener un extremo de entrada y salida de señales de comunicación en banda base o en frecuencia intermedia y otro de transmisión y recepción de radiación hacia y desde uno o más satélites. Por ellos realiza funciones como la modulación, demodulación, conversión de frecuencias, codificación, multiplicación, conversión analógico-digital, etc. Además, algunos tipos de estaciones terrenas solo tienen la capacidad de transmitir o recibir y pueden ser el punto de origen o final de señales o estar enlazadas también por medio de redes terrenales con sitios distantes de origen y destino. Por ello la designación de estación terrena es utilizada indistintamente para indicar todo equipo terminal que se comunica

Sensores

Los sensores son dispositivos que transforman magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas. (Referencia)
Las variables de instrumentación dependen del tipo de sensor y pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc. Una magnitud eléctrica obtenida puede ser una resistencia eléctrica, una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como un fototransistor), etc.

Características

Entre las características principales técnicas de un sensor Se pueden clasificar en dos tipos.

- Estáticas
- Dinámicas

Estáticas

Las características de los instrumentos cuando estos están midiendo cantidades estables, o sea, mientras no presentan variaciones bruscas en su magnitud

Rango

El conjunto de valores que puede tomar la señal de entrada comprendidos entre el máximo y el mínimo detectados por el sensor con una tolerancia de error aceptable.

Resolución

Menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida

Sensibilidad

Variación de la salida producida por una variación de entrada. Pendiente de la curva de calibración.

Linealidad

Expresa lo constante que resulta la sensibilidad del sensor.

Precisión

Cualidad por la que tiende a dar lecturas muy próximas unas a otras.

Exactitud

Diferencia entre la salida real y el valor teórico de dicha salida (valor verdadero).

Dinámicas

Puede ocurrir que la cantidad bajo medición sufra una variación en un momento determinado y por lo tanto es necesario que conozcamos el comportamiento dinámico del instrumento cuando sucedan estas variaciones

Velocidad de respuesta

Capacidad para que la señal de salida siga sin retraso las variaciones de la señal de entrada.

Respuesta frecuencial

Relación entre la sensibilidad y la frecuencia cuando la entrada es una excitación senoidal. Representación mediante un gráfico de Bode.

Estabilidad

Desviación de salida del sensor al variar ciertos parámetros exteriores distintos del que se pretende medir.

Sensor MPU6050

El MPU6050 es un sensor y acelerómetro con 6 grados de libertad, contiene un acelerómetro y giroscopio en un solo empaque. Cuenta con una resolución de 16-bits, lo cual significa que divide el rango dinámico en 65536 fracciones, estos aplican para cada eje X, Y y Z al igual que en la velocidad angular. El sensor es ideal para diseñar control de robótica, medición de vibración, sistemas de medición inercial (IMU), detector de caídas, sensor de distancia y velocidad, y muchas cosas más. El MPU-6050 contiene un giroscópico, un acelerómetro, además de un sensor de temperatura, mediante I2C regresa unos valores conocidos como raw o "crudos" según el registro seleccionado.

En la parte del giroscopio el dispositivo funciona midiendo velocidades angulares basándose en el mantenimiento del impulso de rotación. Si intentamos hacer girar un objeto que está girando sobre un eje que no es el eje sobre el que está rotando, el objeto ejercerá un momento contrario al movimiento con el fin de preservar el impulso de rotación total. El giroscopio muestra el cambio de rango en rotación en sus ejes X, Y y Z

Mide la aceleración, inclinación o vibración y transforma la magnitud física de aceleración en otra magnitud eléctrica que será la que emplearemos en los equipos de adquisición estándar. Los rangos de medida van desde las décimas de g, hasta los miles de g.

Especificaciones

- Salida digital de 6 ejes.
- Giroscopio con sensibilidad de ± 250 , ± 500 , ± 1000 , y ± 2000 dps
- Acelerómetro con sensibilidad de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$
- Algoritmos embebidos para calibración
- Sensor de temperatura digital

- Entrada digital de video FSYNC
- Interrupciones programables
- Voltaje de alimentación: 2.37 a 3.46V
- Voltaje lógico: 1.8V±5% o VDD
- 10000g tolerancia de aceleración máxima

Sensor BME280

El sensor BME280 integra en un solo dispositivo sensores de presión atmosférica, con gran precisión, bajo consumo energético y un formato ultra compacto. Basado en tecnología BOSCH piezo-resistiva con gran robustez EMC, alta precisión y linealidad, así como con estabilidad a largo plazo. Se conecta directamente a un microcontrolador a través de I2C o SPI.

Este tipo de sensores pueden ser utilizados para calcular la altitud con gran precisión (barómetro), por lo que es un sensor muy utilizado en sistemas de Autopiloto para Drones (UAVs) entregando medidas de altitud con una precisión de hasta 1m. Otras aplicaciones son: Monitoreo de clima, Internet de las Cosas, Monitor de salud/fitness, Automatización del hogar o Domótica y Aire acondicionado.

Sensor TMP102

El TMP102 se comunica con el protocolo I2C. El sensor tiene una precisión de 0.5°C, una resolución de 12 bits, y es compatible con otros microcontroladores. El rango recomendado de alimentación para el TMP102 es de 1.4v a 3.6v. Además, posee un rango de medición de -25°C a 85°C. Puede ser aplicado en la industria por su tamaño y fácil manejo, así como en la vida cotidiana para computadoras portátiles e inclusive en la instrumentación médica.

I2C es un protocolo de comunicación en el que hay solo dos vías, SDA y SCL. Serial Data (SDA) es por la que se manda y se recibe información. Serial Clock (SCL) es por donde se envía la señal de reloj. Este protocolo ayuda a disminuir el cableado entre los componentes. Al igual que a facilitar la comunicación de un dispositivo maestro con diversos esclavos

Características:

- 12-bit, 0.0625 ° C de resolución
- Precisión: 0,5 ° C (-25 ° C a +85 ° C)
- Baja corriente de reposo
- 10µA Activa (max)
- 1uA apagado (max)
- 1.4V a 3.6VDC rango de suministro
- Interfaz serie de dos hilos

Sistema de control

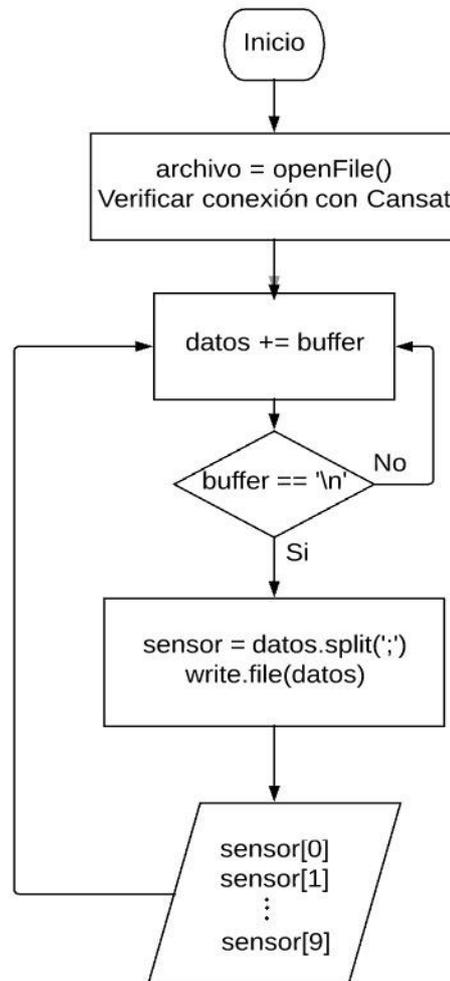
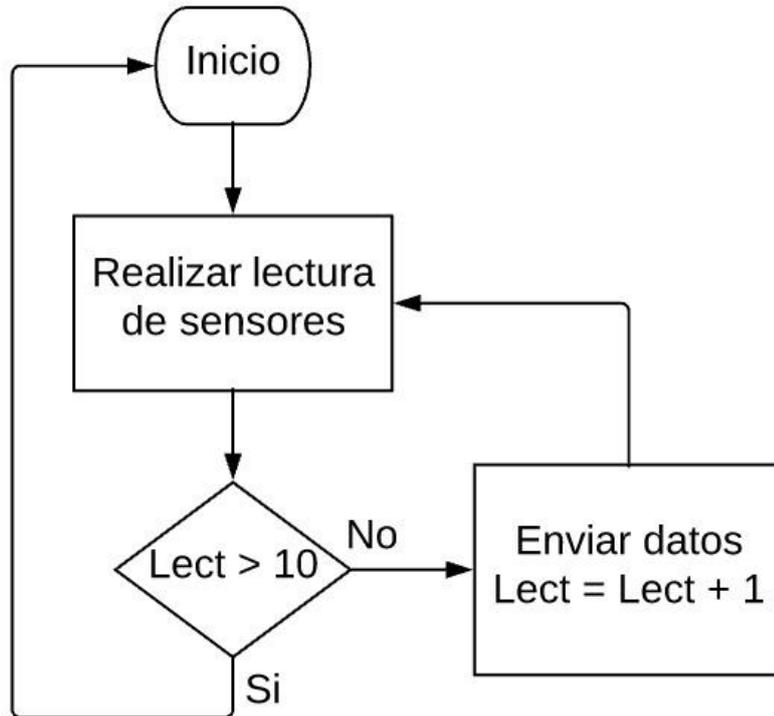
El control es la parte central del CanSat, se encarga de procesar las señales recibidas de los sensores, y de mandar en el formato correcto la información necesaria al sistema de comunicación inalámbrica, por eso el microcontrolador STM32F407VGT6 debe tener los puertos y periféricos necesarios para realizar estas tareas. Existe gran variedad de microcontroladores, los hay de Intel, Microchip, Motorola, National Semiconductor y Texas Instruments, por mencionar algunos. Para el sistema aquí diseñado se eligió un microcontrolador STM32F407VGT6 basado en ARM® Cortex™. Se optó por este circuito debido a que se cuenta con experiencia previa en el manejo de este tipo de procesadores.

Fuente de alimentación

El sistema diseñado requiere dos voltajes, uno de 5 V y otro de 3.3 V, esto es así porque son los voltajes de alimentación más utilizados para componentes electrónicos comerciales. El consumo de corriente se estima en unos 400 mA: los sensores generalmente consumen muy poca corriente, incluso del orden de los micro amperes, los microprocesadores consumen pocos mili amperes, pero el sistema de comunicación inalámbrica se estima que consuma algunos cientos de mili amperes debido a que tiene que transmitir una señal a varios metros de distancia.

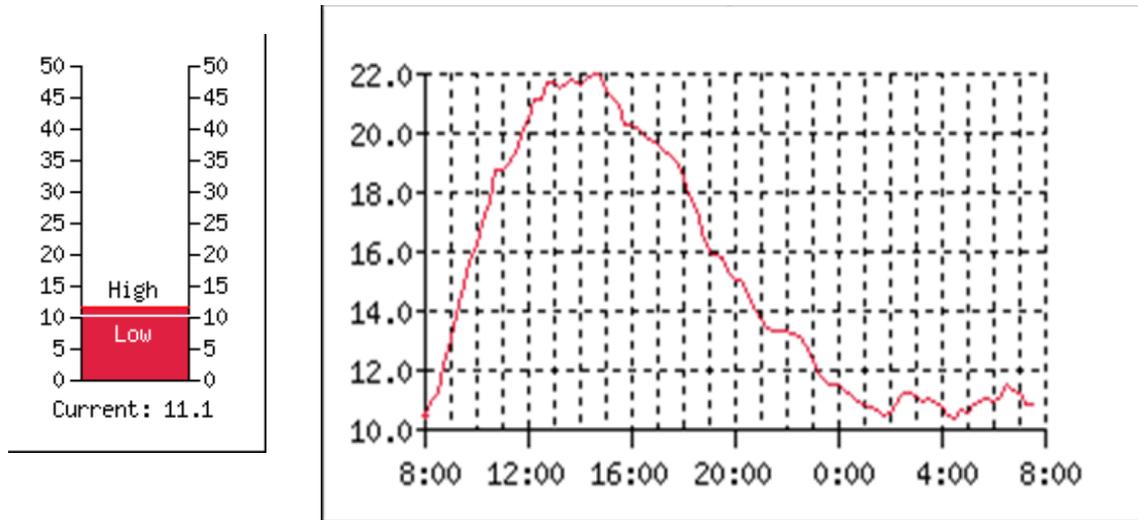
Existen diferentes tipos de baterías, diferentes formas, tamaños y materiales, su principio de funcionamiento se basa en una reacción química. Para este diseño se eligió una batería alcalina de 9 V, principalmente por su precio y alta disponibilidad. El valor de voltaje nominal facilita el uso de los reguladores mencionados arriba. Además, puede proporcionar una corriente de 500 mA durante aproximadamente media hora continua, suficiente para lo que dura la operación del CanSat, si es que éste demandara esta cantidad de corriente. Además de estos componentes, se necesitan otros elementos como capacitores para el correcto funcionamiento de los reguladores.

Desarrollo



Resultados

Estas son las gráficas de datos recolectados por el sensor de temperatura del CanSat en un rango de 12 hrs de 8 am a 8 pm



Conclusiones

Durante el proceso de prueba de los dispositivos de comunicación de la estación terrena y el Cansat se determinó que la antena Lora Ra-02 es la mejor opción para poder lograr la comunicación correcta y estable.

La miniaturización de dispositivos espaciales es una realidad que permite acceder a tecnología espacial.

Estos pico-satélites no son exclusivos para los estudiantes del área aeroespacial, sino que promueven las actividades en equipos disciplinarios y multidisciplinarios.

Además, son útiles en aplicaciones para resolver problemas específicos de electrónica y comunicaciones, inclusive en física, por ejemplo, en el cálculo de trayectorias, mecánica orbital, etcétera o en temas de aeronáutica como el diseño y cálculo de paracaídas, diseño y construcción de cohetes, aeromodelos y drones.

Que es Python (mandarlo a un anexo)

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

El creador del lenguaje es un europeo llamado Guido Van Rossum. Hace ya más de una década que diseñó Python, ayudado y motivado por su experiencia en la creación de otro lenguaje llamado ABC. El objetivo de Guido era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se hacía en Unix usando C.

El desarrollo de Python duró varios años, durante los que trabajó en diversas compañías de Estados Unidos. En el 2000 ya disponía de un producto bastante completo y un equipo de desarrollo con el que se había asociado incluso en proyectos empresariales. Actualmente trabaja en Zope, una plataforma de gestión de contenidos y servidor de aplicaciones para el web, por supuesto, programada por completo en Python.

Características del lenguaje

Propósito general

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

Multiplataforma

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

Interpretado

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos byte codes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

Orientado a Objetos

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

Funciones y librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

Librería Matplotlib

Los seres humanos son criaturas muy visuales: entendemos mejor las cosas cuando vemos las cosas visualizadas. Sin embargo, el paso para presentar análisis, resultados o perspectivas puede ser un cuello de botella: es posible que ni siquiera sepa dónde comenzar o que ya tenga un formato correcto en mente, pero luego preguntas como "¿Es esta la forma correcta de visualizar las ideas que Quiero traer a mi audiencia?"

Cuando estás trabajando con la biblioteca de ploteo de Python, Matplotlib, el primer paso para responder las preguntas anteriores es acumular conocimientos sobre temas como:

La anatomía de un gráfico de Matplotlib: ¿qué es un argumento secundario? ¿Qué son los ejes? ¿Qué es exactamente una figura?

Creación de gráficos, lo que podría generar preguntas sobre qué módulo necesita exactamente importar (¿pylab o pyplot?), Cómo debería iniciar

exactamente la figura y los ejes de su gráfico, cómo usar matplotlib en los cuadernos de Jupyter.

Trazar rutinas , desde formas sencillas de trazar sus datos hasta formas más avanzadas de visualización de sus datos.

Personalizaciones básicas de la trama , con un enfoque en las leyendas y el texto de la trama, títulos, etiquetas de ejes y diseño de la trama.

A primera vista, parecerá que hay algunos componentes a tener en cuenta al comenzar a trazar con esta biblioteca de visualización de datos de Python.

Afortunadamente, esta biblioteca es muy flexible y tiene muchos valores predeterminados prácticos que lo ayudarán enormemente. Como tal, no necesita mucho para comenzar: necesita realizar las importaciones necesarias, preparar algunos datos y puede comenzar a trazar con la ayuda de la `plot()` función. Cuando esté listo, no olvide mostrar su trama utilizando la `show()` función.

Matplotlib, pyplot y pylab: ¿cómo se relacionan?

Cuando se habla de "Matplotlib", se habla de todo el paquete de visualización de datos de Python. En segundo lugar, `pyplot` hay un módulo en el `matplotlib` paquete. Es por eso que a menudo se ve `matplotlib.pyplot` en el código. El módulo proporciona una interfaz que le permite crear de forma implícita y automática figuras y ejes para lograr la trama deseada.

Esto es especialmente útil cuando desea trazar algo rápidamente sin crear una instancia de Figuras o Ejes, como se vio en el ejemplo en la primera sección de este tutorial. Usted ve, no ha especificado explícitamente estos componentes, ¡sin embargo, logra generar un gráfico que incluso ha personalizado! Los valores predeterminados se inicializan y cualquier personalización que realice se realizará teniendo en cuenta la figura y los ejes actuales.

Por último, `pylab` es otro módulo, pero se instala junto al `matplotlib` paquete. Importa grandes cantidades `pyplot` y la `numpy` biblioteca, y generalmente se recomendó cuando trabajaba con matrices, hacía matemáticas de forma interactiva y quería acceder a las funciones de trazado. Como solución, puede usar `%matplotlib` magia en combinación con el backend correcto, como `inline`, `qt` etc. La mayoría de las veces, querrá usarla `inline`, ya que esto asegurará que las parcelas estén incrustadas dentro del cuaderno.

Librería Tkinter

Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python, con estos queremos decir que Tk se encuentra disponible para varios lenguajes de programación entre los cuales se encuentra Python con el nombre de Tkinter. Este no es más que una capa de esta librería para el lenguaje Python con lo cual usar Tk en otro lenguaje no nos supondrá un inconveniente.

Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python y es el que viene por defecto con la instalación para Microsoft Windows y

preinstalado en la mayoría de las distribuciones de GNU/Linux. Con Tkinter podremos conseguir resultados casi tan buenos como con otras librerías gráficas siempre teniendo en cuenta que quizás con otras herramientas podamos realizar trabajos más complejos donde necesitemos una plataforma más robusta, pero como herramienta didáctica e interfaces sencillas nos sobra, dándonos una perspectiva de lo que se trata el desarrollo de una parte muy importante de una aplicación si deseamos distribuirla. Gracias a Tkinter veremos cómo interactuar con el usuario pidiéndole el ingreso de datos, capturando la pulsación de teclas, movimientos del mouse, entre algunas de las cosas que podremos lograr.

Librería Panda

Pandas es un paquete de Python que proporciona estructuras de datos similares a los dataframes de R. Pandas depende de Numpy, la librería que añade un potente tipo matricial a Python. Los principales tipos de datos que pueden representarse con pandas son:

- Datos tabulares con columnas de tipo heterogéneo con etiquetas en columnas y filas.
- Series temporales.

Pandas proporciona herramientas que permiten:

- Leer y escribir datos en diferentes formatos: CSV, Microsoft Excel, bases SQL y formato HDF5
- Seleccionar y filtrar de manera sencilla tablas de datos en función de posición, valor o etiquetas
- Fusionar y unir datos
- Transformar datos aplicando funciones tanto en global como por ventanas
- Manipulación de series temporales
- Hacer gráficas

En pandas existen tres tipos básicos de objetos todos ellos basados a su vez en Numpy:

- Series (listas, 1D),
- DataFrame (tablas, 2D) y
- Panels (tablas 3D).

Series

Las Series se pueden crear tanto a partir de listas como de diccionarios. De manera opcional podemos especificar una lista con las etiquetas de las filas. Primero necesitamos cargar la librería correspondiente:

```
>>> from pandas import Series
>>> serie = Series(['a', 'b', 'c'])
>>> serie
0    a
```

```

1    b
2    c
>>> serie = Series(['a', 'b', 'c'], index=['pregunta1', 'pregunta2', 'pregunta3'])
>>> serie
pregunta1    a
pregunta2    b
pregunta3    c

```

En el caso de que usemos un diccionario, el nombre de las filas toma el valor de las keys del diccionario (index).

```

>>> dict = {'pregunta1': 'a', 'pregunta2': 'b', 'pregunta3': 'c'}
>>> serie = Series(dict)
>>> serie
pregunta1    a
pregunta2    b
pregunta3    c

```

Podemos controlar tanto los datos que queremos incluir como su orden especificando el index:

```

>>> serie = Series(dict, index=['pregunta3', 'pregunta1', 'pregunta4'])
>>> serie
pregunta3    c
pregunta1    a
pregunta4    NaN

```

DataFrame

Los DataFrame se pueden crear de diferentes maneras, una forma común de crearlos es partir de listas o diccionarios de listas, de diccionarios o de Series. En los DataFrame tenemos la opción de especificar tanto el **index** (el nombre de las filas) como **columns** (el nombre de las columnas).

```

>>> from pandas import DataFrame
>>> list_listas = [[1, 2, 3, 4],[4, 3, 2, 1]]
>>> dataframe = DataFrame(list_listas, columns=['uno', 'dos'])
>>> dataframe
   uno dos
0    1  4
1    2  3
2    3  2
3    4  1
>>> dict_listas = {'uno' : [1, 2, 3, 4], 'dos' : [4, 3, 2, 1]}
>>> dataframe = DataFrame(dict_listas)
>>> dataframe
   uno dos
0    1  4
1    2  3
2    3  2
3    4  1

```

```

>>> dataframe = DataFrame(dict_listas, index=['a', 'b', 'c', 'd'])
>>> dataframe
  uno dos
a  1  4
b  2  3
c  3  2
d  4  1
>>> dict_series = {'uno' : Series([1, 2, 3], index=['a', 'b', 'c']),
                  'dos' : Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
>>> dataframe = DataFrame(dict_series)
>>> dataframe
  uno dos
a  1  1
b  2  2
c  3  3
d NaN  4

```

Al igual que con las Series, podemos controlar tanto los datos que queremos incluir como su orden especificando el index y/o columnas que nos interesen:

```

>>> dataframe = DataFrame(dict_series, index=['d', 'b', 'a'])
>>> dataframe
  uno dos
d NaN  4
b  2  2
a  1  1
>>> dataframe = DataFrame(d, index=['d', 'b', 'a'], columns=['dos', 'tres'])
>>> dataframe
  dos tres
d  4 NaN
b  2 NaN
a  1 NaN

```