REVISIÓN CRÍTICA DE DISEÑO

2021-2022 Proyecto Laikata

Equipo: Laikata Team

Integrantes: Pepe Masanet
Josep Mengual
Daniel Kauss
Jairo Salvà
Yeray Olalde
Unai Nieto
Izan Castellano
Arnau Camprubi

Mentor: Javier Gómez

Colaboradores: Marc Colomer
Ana María Ruiz

Patrocinadores:


MADRE ARNAU


CRISTINA M.

LAIKATA

CanSat 2021-2022

Índice

1. Introducción

Todos nosotros estamos acostumbrados a aprender por proyectos. Lo hacemos en el instituto, y también en la academia de robótica a la que asistimos. Dejando un poco de lado los proyectos personales, terminamos el curso del año pasado haciendo un “Curiosity” a escala, con sus seis ruedas, distintos drivers para controlar servos y motores, tres placas de control para gobernar los distintos módulos, todas ellas conectadas con un mini-PC a través de I2C, al cual nos conectábamos por SSH y canales TCP/UDP para controlarlo. Fue todo un reto ya que era la primera vez que un proyecto requería de tantos sistemas. Aprendimos, y logramos ponerlo en funcionamiento.

Este año, 2021-2022, decidimos que, entre otros proyectos, nos presentaríamos al desafío CanSat. Nos pareció muy interesante el tener que lidiar con restricciones como el espacio, sistemas de alimentación, links de radio con no mucho ancho de banda, etc. Así es que tras hablarlo unos días y dar con una misión secundaría que nos motivase a todos, nos pusimos manos a la obra, le pusimos nombre a nuestro pico-satélite (Laikata) y al equipo (Laikata Team).

1.1. Organización y roles del equipo

Laikata Team está formado por 8 personas, mayoritariamente estudiantes de secundaria, con alrededor de 16 años. Tenemos en común un gran interés por la programación y la robótica, pero no llegamos a ponernos de acuerdo de si es mejor programar en lenguajes de alto nivel o ensamblador.

Para afrontar el proyecto y poder avanzar más rápido, decidimos dividir la empresa en

Figura 1: Organigrama Laikata Team

departamentos. En cada uno de ellos nombramos a una persona responsable, la cual podía solicitar la colaboración de quien quisiese, por lo que todos terminaríamos haciendo de todo, pero con ciertos supervisores, que serían los máximos responsables en sus áreas.

Los departamentos/equipos creados han sido los siguientes:

- **Equipo de diseño estructural:** Tendrá como objetivo el diseño de Laikata para que ésta pueda integrar todos los sistema requeridos para afrontar las dos misiones, así como el sistema de parapente.
- **Equipo de diseño electrónico:** Su misión será el diseño de todo circuito y conexión electrónica que requieran los distintos sistemas.
- **Equipo de desarrollo de software:** Encargado de desarrollar tanto el software como el firmware necesario, seleccionar el IDE a utilizar, y asegurarse de que todos lo sabían manejar.
- **Equipo de radio comunicaciones:** Encargado de todo el sistema de comunicación, tanto entre Laikata y la base, como del rover. También llevará la parte técnica de las comunicaciones internas entre los distintos componentes de los equipos.
- **Equipo de testeo:** Su misión, preparar cuantas pruebas sean necesarias para verificar que el diseño supera las especificaciones requeridas, tanto de software como de hardware. Será el encargado de planificar las posibles salidas para realizar pruebas de vuelo.
- **Equipo de redes sociales, comunicación, marketing, branding y financiación:** Este equipo se encargará de las publicaciones en redes sociales, comunicaciones con los distintos medios, y el posible marketing de la misión para conseguir financiación.

1.2. Objetivo de la misión

El desafío CanSat se compone de dos misiones:

Misión principal: viene dada por la organización, y consiste en que tras la liberación y descenso del CanSat, este deberá enviar por telemetría hasta la estación de tierra, mediciones de temperatura y presión atmosférica al menos una vez por segundo.

Misión secundaria: elegida por nosotros. Consiste en una demostración de tecnología. Vamos a conseguir que Laikata vuelva navegando hasta un punto determinado mediante el control de un parapente. Aprovecharemos también el canal Down-link de Laikata para enviarnos una gran cantidad de

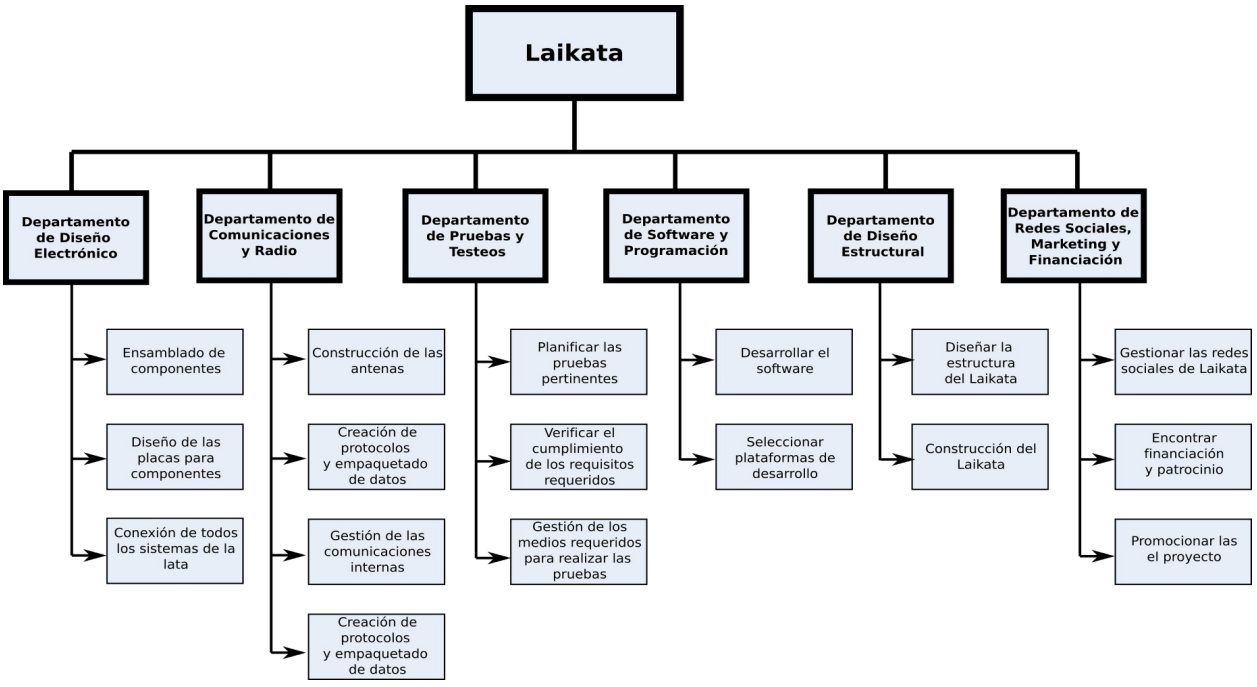


Figura 2: Estructura de descomposición del trabajo

datos, como la posición exacta mediante GPS, el valor de los 9 ejes de un sensor inercial, la tensión de la batería, así como información sobre las decisiones tomadas por el navegador de forma autónoma. Con todo ello, crearemos un sistema de monitorización muy completo de nuestro Laikata en la estación de control de tierra.

2. Descripción del proyecto CanSat

2.1. Esquema de la misión

Nuestra misión consiste en hacer que nuestro CanSat navegue de forma autónoma hasta ciertas coordenadas de destino. A su vez, enviaremos una gran cantidad de datos de telemetría, entre los que se incluirán temperatura y presión, para cumplir con la misión principal, e información sobre sensores inerciales, monitorización de batería, e información sobre decisiones tomadas para la navegación. Todos estos datos se recibirán en la estación de tierra, donde se implementará un software para el registro y visualización de todos ellos. La estación de tierra dispondrá de dos antenas con posicionamiento automático síncrono, una de ellas con polarización horizontal y la otra vertical. Las dos antenas recibirán las señales enviadas por Laikata y enviarán los datos recibidos al software de monitorización, y éste será el encargado de coger únicamente las tramas que hayan llegado de forma íntegra, tras realizar la verificación.

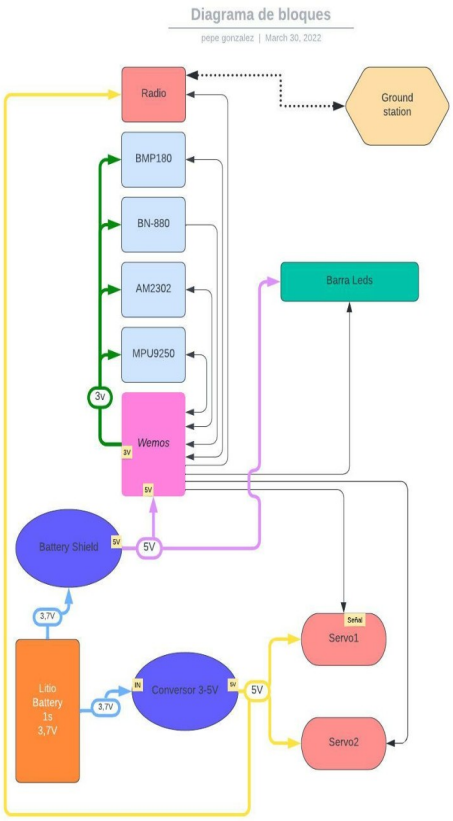


Figura 3: Diagrama de conexiones

El objetivo de la misión de nuestro CanSat es que mediante el gps, el CanSat maniobre con el parapente para dirigirse a una ubicación exacta donde se sitúa nuestro apoyo terrestre. Durante el lanzamiento la radio pasará continuamente tramas con telemetría que nos ayudará a saber en todo momento donde está ubicado y que está pasando.

Daremos por cumplidos los objetivos de nuestra misión secundaria siempre y cuando Laikata logre aterrizar en un radio de 10m de su punto de destino. Si las condiciones meteorológicas impidiese el correcto desempeño de la misión, siempre nos quedará una baliza de radio en la que se enviarán tramas con la localización exacta para su recuperación.

Para la misión secundaria se necesitan de varios componentes como son, los servos para frenar y girar el parapente; un módulo de comunicaciones por radio para comunicarse con la estación base y un GPS para conocer la ubicación del CanSat y que así el mismo pueda dirigirse al punto de caída. Para alimentar el microcontrolador ESP8266 y todos los

sensores usamos el módulo “battery shield” de Wemos. Los servos y el módulo de radio se alimentan con una fuente de alimentación independiente.

Todo el sistema está alimentado con una batería de Litio de una celda con capacidad de 1400mAh. Se ha estimado que durante el descenso Laikata consumirá unos 800mA durante 200s. Una vez en tierra, se desactivarán, y se reducirán las transmisiones a una baliza cada 5 segundos, por lo que su consumo descenderá a unos 100mA. Haciendo una estimación calculamos que la batería nos permitirá unas 9h de autonomía tras el despegue, tiempo más que suficiente para recuperar a Laikata aún y cuando no haya podido cumplir su misión secundaria con éxito.

Si, por lo tanto todo se produce con normalidad el CanSat aterrizará en la zona de aterrizaje sin desviarse. Además usando el GPS incorporado podremos localizar dónde se encuentra en todo momento el cansat y así poder recuperarlo.

2.3. Diseño mecánico y estructural

2.3.1. Diseño de la lata

Para el diseño de nuestra lata hemos utilizado un diseño modular, ya que nos permite de forma fácil trabajar con los distintos módulos, pudiendo modificar

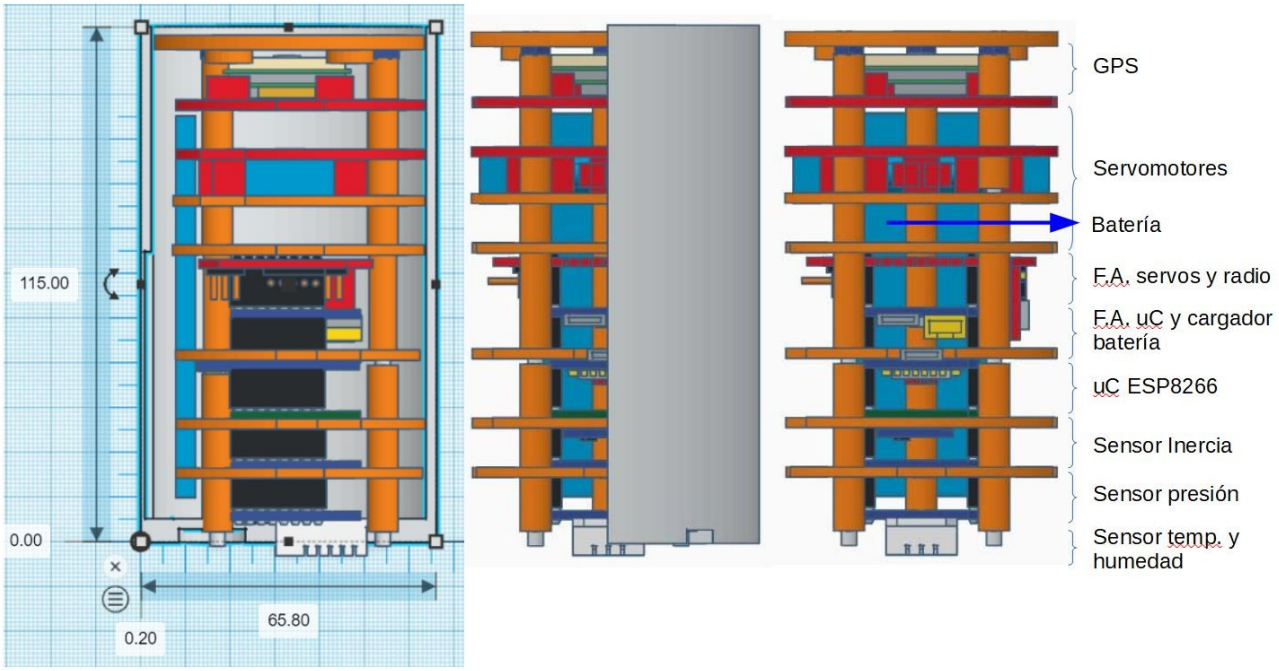


Figura 4: Diseño del cansat

2.2. Demostración de tecnología

Se demostrará con el CanSat que se es capaz de, mediante cálculos hechos en tiempo real, calcular el vector actual del CanSat respecto a la zona indicada de donde debe aterrizar. Y con este vector mover los servos para que respectivamente se mueva el paracaídas y dirigir la nave al lugar de aterrizaje.

alguno de ellos sin afectar al conjunto.. Como herramienta para el diseño se ha utilizado Tinkercad de Autodesk. Y como material se ha optado por la utilización de PLA[1], por ser más sostenible con el medio ambiente que el ABS[2], aunque sea un poco menos robusto. Unas varillas de acero de 3mm serán

[1] PLA (ácido pliláctico) es un termoplástico cuyos materiales de base se obtienen a partir de almidón de maíz o yuca. Es utilizado ampliamente en la impresión 3D.

[2] ABS (acrilonitrilo butadieno estireno) es un termoplástico amorfo muy utilizado en la industria por su alta resistencia al impacto.

las utilizadas para unir los distintos módulos que posteriormente se introducirán dentro de la lata, la cual tiene un espesor de 2.3mm para darle la suficiente solidez..

Cada módulo contiene una parte diferente de la lata, como los sensores, los módulos de radio, o los servos. Estos últimos son los utilizados para regular los frenos del parapente para su guiado.

2.3.2. Diseño del parapente

Como sistema para frenar y al mismo tiempo guiar la caída optamos por utilizar un parapente. "*Un parapente es un aerodeslizador ultraliviano, que consta de un ala de estructura no rígida o blanda llamada VELA o VELAMEN, construida de tela sintética denominada Rip Stop cuya trama tiene hilos de refuerzo que frenan eventuales desgarros* [i] ".

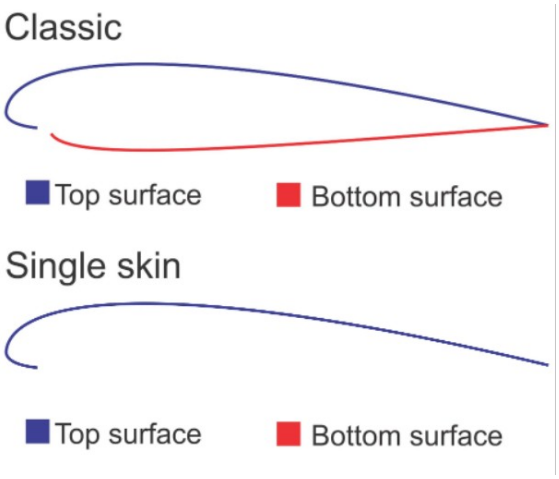


Figura 5: Esquema del ala del parapente

El efecto del aire en movimiento sobre el perfil aerodinámico puede descomponerse en dos fuerzas, una de sustentación, y otra de resistencia al avance. Por ello, se necesita que el parapente esté continuamente avanzando y cayendo para conseguir algo de fuerza de sustento.

Los parapentes clásicos consiguen que la vela mantenga su forma aerodinámica gracias a una cámara de aire que mantienen aportando aire con cierta presión gracias a unas aperturas frontales en la misma. La construcción de este tipo de parapentes es muy laboriosa ya que requiere de muchas celdas debidamente cosidas en su interior para mantener esas cámaras aisladas entre si.

Por otro lado, los parapentes "single skin" se basan en el mismo efecto aerodinámico ya al avanzar el aire se sigue viendo afectado por esa forma. Del mismo modos que los clásicos, éstos disponen de unas costillas que le ayudan a mantener esa forma, aunque debido a que no tiene vela por la parte inferior, se hace mucho más sencilla su construcción. Otra ventaja evidente, es la reducción de peso.

Antes de fabricar el parapente, tuvimos que calcular el área necesaria para cumplir los requisitos de la ESA. Se nos exigió que la velocidad durante el descenso no superara los 12 m/s, pero tampoco fuera menor que 6 m/s.

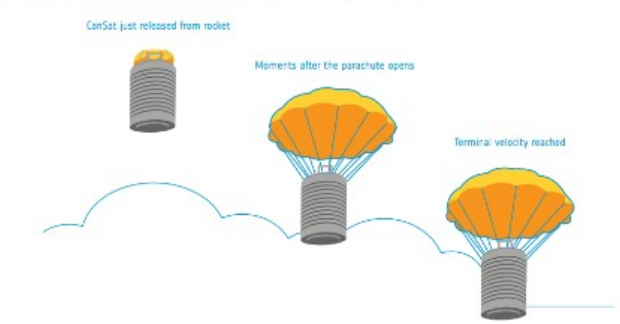


Figura 7: Etapas vuelo cansat

La fuerza debida a la gravedad que experimenta un objeto en caída libre es:

$$F = m \cdot g$$

Y la fuerza que experimenta por la fricción del paracaídas con el aire al caer:

$$F = \frac{-1}{2} CpAv^2$$

Siendo:

C: coeficiente de arrastre. Depende de la forma del paracaídas usado. En nuestro caso, 0,75

p: densidad del aire. Se calcula con 1225 kg/m³

A: área del paracaídas

v: velocidad

Cuando el paracaídas llega a su velocidad terminal, ésta permanece constante, y se cumple:

$$\sum F = 0$$

Por tanto:

$$m \cdot g - \frac{1}{2} CpAv^2 = 0$$

Como buscamos conocer el área necesaria para la velocidad mínima, despejamos el área:

$$\frac{1}{2}CpAv^2 = m \cdot g$$

$$CpAv^2 = 2m \cdot g$$

$$A = \frac{2mg}{Cpv^2}$$

El área requerida para la velocidad máxima (6m/s) sería:

$$A = \frac{2 \cdot 0{,}3 \cdot 9{,}8}{0{,}75 \cdot 1225 \cdot 12^2} = 0{,}04\, m^2$$

Mientras que el área para obtener la mínima velocidad de descenso el área debe de ser de:

$$A = \frac{2 \cdot 0{,}3 \cdot 9{,}8}{0{,}75 \cdot 1{,}225 \cdot 6^2} = 0{,}1777\, m^2$$

Con estos datos procedimos a realizar el diseño definitivo del parapente, utilizando el software "SingleSkin" [ii] , el cual permite diseñar parapentes de piel única, siendo estos más sencillos de implementar que los de cámara.

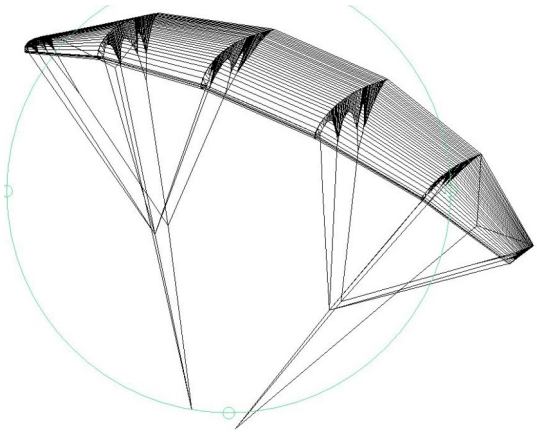


Figura 8: Esquema parapente

Al software te tuvimos que dar las características del parapente que queríamos diseñar. El primer dato que le aportamos fue el del área. El software está preparado para diseñar parapentes y cometas más grandes, por lo que la resolución de los m² del área era de solo 0'1m^2. Teníamos únicamente dos opciones: 0'1m^2 o 0'2m^2.

Otro dato que tuvimos que modificar de los valores por defecto fue el número de celdas. A mayor número de celdas, mayor rigidez estructural del parapente, es decir, más iba a mantener la forma diseñada. Sabíamos que iba a ser laborioso de fabricar, por lo que reducimos el número de celdas a 5. Debido a ello estimamos que sufriría un mínimo plegado que reduciría su área final en un 15%. Teniendo estos dos datos en cuenta, y siguiendo como objetivo obtener la velocidad mínima de descenso dentro de los parámetros válidos (6-12m/s) nos decantamos por construir un parapente de 0'2m^2. Revisamos los cálculos de nuevo para asegurarnos que nos encontrábamos aún por encima de los 6m/s:

$$v = \sqrt{\frac{2 \cdot 0{,}3 \cdot 9{,}8}{0{,}75 \cdot 1{,}225 \cdot 0{,}2 * 0{,}85}} = 6{,}13\, m/s$$

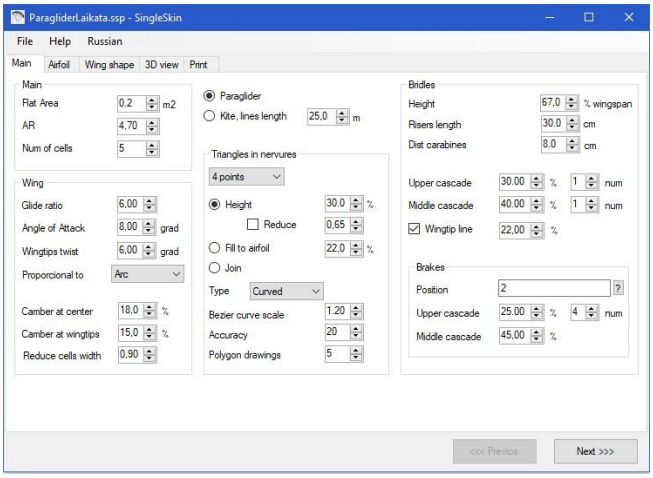


Figura 9: Configuración software "SingleSkin"

Una vez introducidos los parámetros de diseño, imprimimos las plantillas en folios de papel para luego transferirlas a la tela especial anti-desgarros con la que crearemos nuestro parapente. Estas plantillas incluían unos puntos de referencia para realizar las uniones entre las distintas piezas.

Figura 10: Corte de costillas del parapente

Tras esto, empezó el trabajo de unir todas las

piezas para más tarde coser también las líneas del parapente.

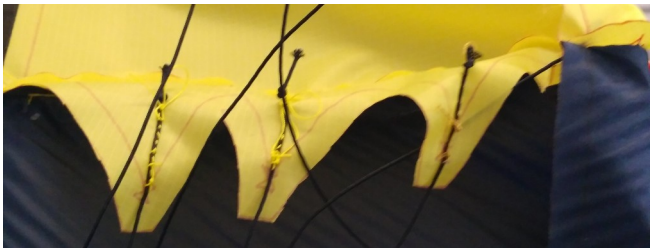


Figura 11: Detalle costuras cuerdas a parapente

Para las líneas se usa una cuerda especial de kevlar[3] de 1mm, la cual soporta según el fabricante una fuerza de 60kg. Uno de los puntos a cumplir para el diseño del paracaidas es que las sean capaces de soportar una fuerza de 500N. Esto sería equivalente a que soportasen 51kg, por lo que estamos dentro del margen.

$$F = mg \rightarrow m = 500/9.8 \rightarrow m = 51\,kg$$

Figura 12: Ajustando la longitud de las cuerdas

Finalmente, se unieron las cuerdas en los puntos especificados por el software, ya que no todas ellas van desde el parapente hasta la lata, sino que se unen a mitad camino. Esto reduce por un lado la fricción avance, y por otro el peso total del parapente.

2.4. Diseño electrónico

Para la electrónica nos hemos basado en la familia Wemos D1 Mini, que contiene un ESP8266. Nos hemos decidido por esta placa ya que con ella se puede implementar fácilmente un sistema modular. Dispone de distintos "hats" como el de baterías, sensores de presión, de temperatura, etc. Su formato es muy reducido en tamaño, lo que nos ayudaría a conectar cuantos módulos necesitásemos en vertical.

Para la misión principal se ha escogido el sensor BPM180 para medidas de presión, con un margen de error de error de 0.02hPa, el AM2302 para realizar medidas de temperatura con una precisión de ±0.5ºC. Como módulo de radio para transmitir la telemetría se ha escogido el módulo HC12.

Para la misión secundaria, hemos añadido el MPU9250 como sensor inercial de 9 ejes, como GPS utilizamos el BN-880, que soporta GPS, GLONASS, Galileo, BeiDou,QZSS y SBAS, y dos microservos.

Para completar nuestro sistema se añadió el "hat" de baterías, que nos permite tanto cargar la batería como proporcionar los 5V que necesita la placa de nuestro microcontrolador para trabajar. Se añadió también una pequeña fuente de alimentación conmutada StepUp para proporcionar 5V a los servos y al módulo de radio.

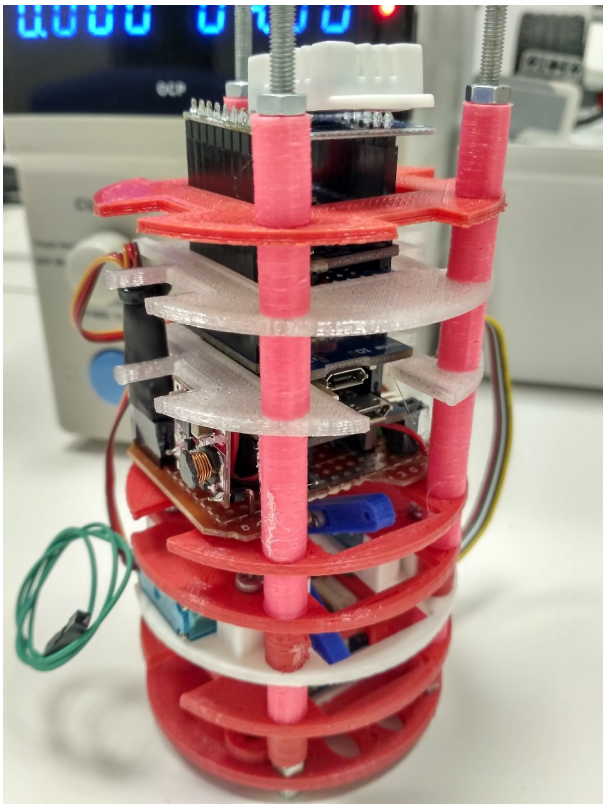


Figura 13: Laikata ensamblado

[3] Kevlar es un material especial (poliparafenileno tereftalamida). Posee una excepcional resistencia a la tracción, de en torno a los 3,5 Gpa, mientras que el acero es de 1,5 Gpa.

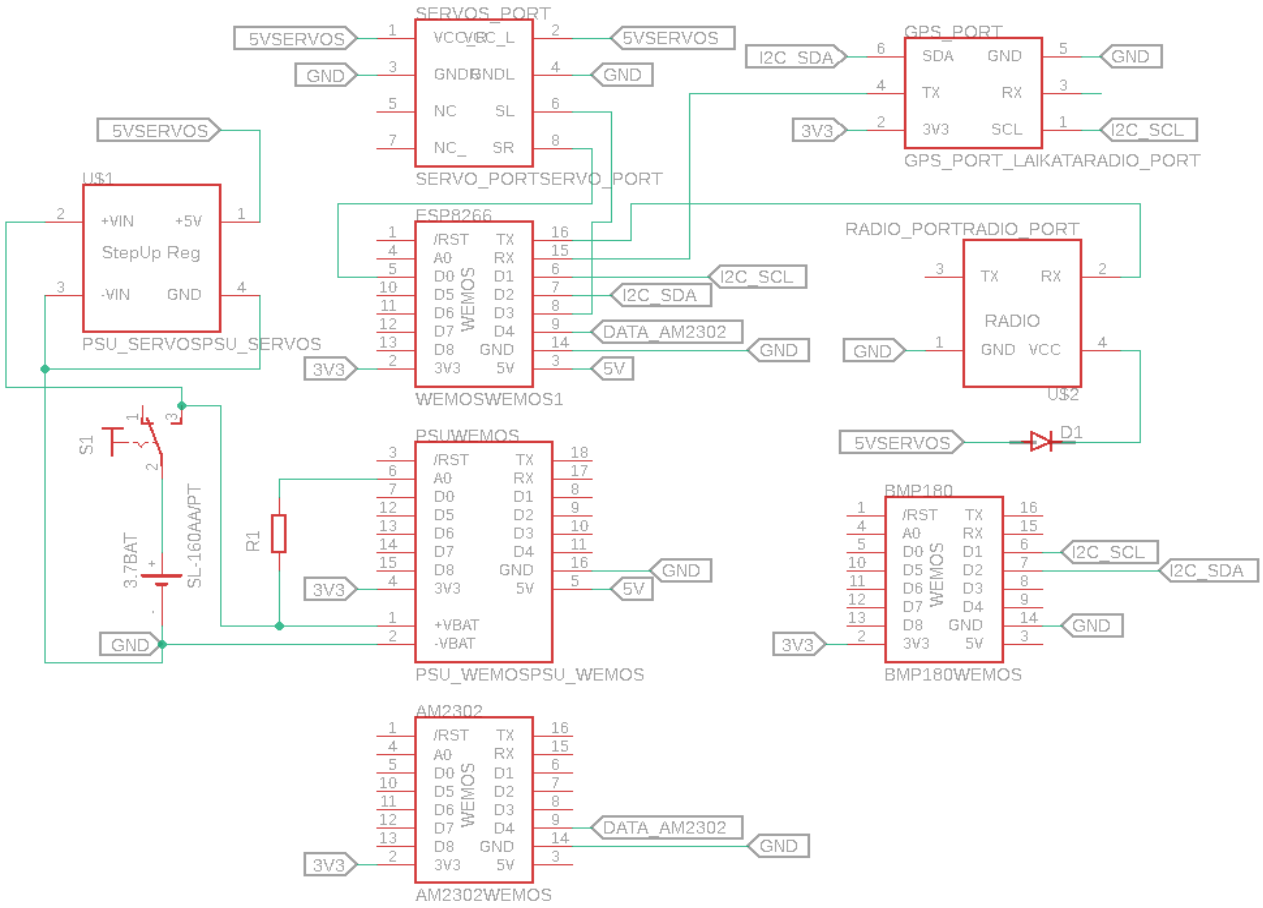


Figura 14: Esquema electrónico Laikata

Como se puede apreciar, alguno de ellos usan el bus I2C, otros la UART, y otros un protocolo propio. El GPS y el módulo de radio comparten UART, ya que la comunicación con ambos es unidireccional. De este modo, nos ha sido suficiente con la única UART por hardware de que dispone el ESP8266. En caso de requerir en un futuro comunicación bidireccional, por ejemplo con el enlace de tierra, se implementaría una UART por software en dos puertos no utilizados.

En lo que respecta a la batería, se ha hecho un cálculo teórico atendiendo a las hojas de especificaciones de los distintos fabricantes. Con estos datos, se ha procedido a la estimación de dos estados de consumo:

- Estado de vuelo, consumo de 253mA
- Estado de recuperación, consumo 114mA

Con estos datos, y teniendo en cuenta de que disponíamos de una batería de 1400mAh, superamos ampliamente las 4h de autonomía requerida para la recuperación del cansat.

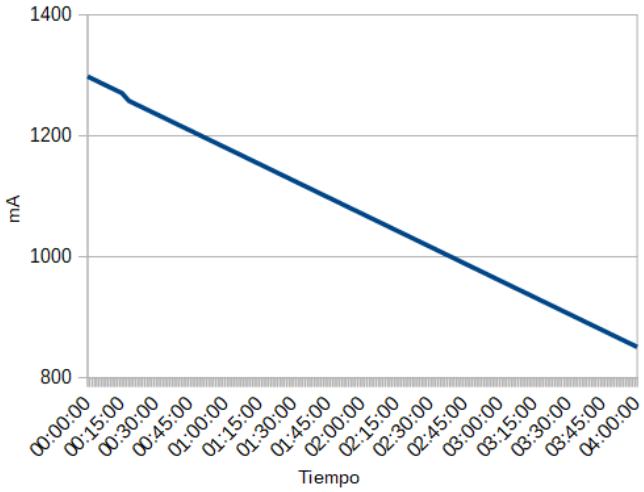


Figura 15: Capacidad de la batería en lanzamiento estimado

2.5. Software

2.5.1. Diagrama de flujo de la programación

Para ilustrar mejor el funcionamiento del código principal del CanSat, hemos representado los procedimientos de este en un diagrama de flujo.

2.5.2. Protocolo de comunicación con la estación de tierra

Hemos creado un protocolo ligero, que empaqueta los datos a enviar y añade un *checksum*[4] en caso de que haya interferencias en el canal de radio.

	SYN	HEADER	COUNT	DATA	CHECKSUM
bytes	1	1	1	1-255	4

Figura 16: Esquema paquetes transmitidos

DATA		
Header	Longitud	Latitud
0x01	1 float (4 bytes)	1 float (4 bytes)

Figura 17: Campo DATA GPS

DATA		
Header	Voltaje	Servo A
0x04	float (4 bytes)	float (4 bytes)

Figura 18: Campo DATA sensores adicionales

DATA		
Header	Magnetómetro	Acelerómetro
0x02	3 floats (4 bytes)	3 floats (4 bytes)

Figura 19: Campo DATA IMU

DATA		
Header	Temperatura	Humedad
0x03	1 float (4 bytes)	1 float (4 bytes)

Figura 20: Campo DATA temp. y humedad

Para enviar datos a través del protocolo se envían paquetes que contienen lo siguiente:

- Un carácter que marca el inicio del paquete (usamos *SYN* o 0x16)
- Un *header* que indica el tamaño de los datos en bytes. El *header* mide un *byte* (rango 0 a 256)
- Un contador que nos ayuda a determinar cuándo perdemos un paquete. Asciende uno por cada paquete enviado.
- Los datos a transmitir (del tamaño especificado en el *header*)
- El *CRC-32* de los datos (cuyo tamaño es de 32 *bits* o 4 *bytes*)

Utilizando este protocolo como base hemos definido un conjunto de subprotocolos para agilizar el desarrollo del software de comunicación entre la estación base y el CanSat. Hay un total de cuatro subprotocolos:

2.5.3. Cantidad de datos recopilados

El CanSat contiene un IMU que recopila la aceleración, la señal magnética y orientación del CanSat en tres ejes cada uno. También recopilamos la temperatura y humedad a través de placas dedicadas. Toda esta información se transmite en tiempo real a través del módulo de radio.

2.5.4. Lenguajes de programación utilizados

Hemos usado el *subset* de *C++* del framework de *Arduino* para la programación del ESP8266. El código se ha compilado y subido a las placas utilizando PlatformIO[iii], un IDE y compilador multiplataforma y multiarquitectura.

Todo el código se encuentra en el GitHub del proyecto[iv]. Hemos usado git para el control de versiones.

2.6. Sistema de recuperación

[4] Checksum es una suma de redundancia. Es una función de redundancia que tiene como objetivo detectar cambios accidentales en la secuencia de datos para proteger su integridad.

Una vez que el CanSat haya aterrizado en el terreno con la ayuda del paracaídas, una matriz LED delataría la posición de la lata. Además, para poder recuperarlo mejor se usaría el recorrido recibido del GPS para saber la zona aproximada de aterrizaje, incluso en el caso en el que no aterrice en la zona designada. También se seleccionó para la tela del paracaídas colores que contrastan tanto con el terreno del suelo, como con el cielo cuando aún está en el aire. Estos colores son azul oscuro y amarillo. Gracias a esto se facilita el poder visualizar dónde se encuentra con la propia vista, así se puede recuperar en caso de que todo lo anterior falle.

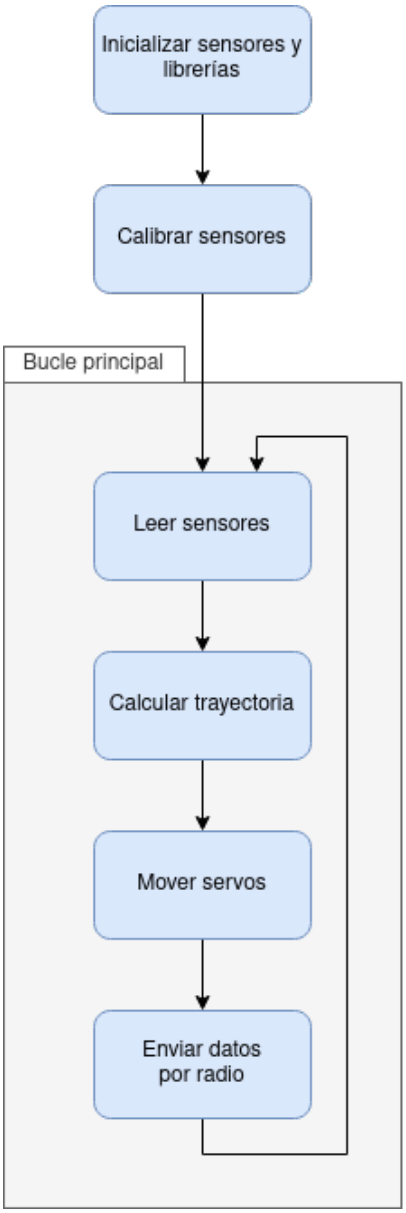


Figura 21: Diagrama de flujo del código principal del CanSat

2.7. Estación de tierra

La estación de tierra consiste en dos receptores de radio (HC12) conectados a dos antenas con polarización cruzada. Éstos estarán conectados a un Arduino, el cual se comunica con un portátil sobre mediante USB.

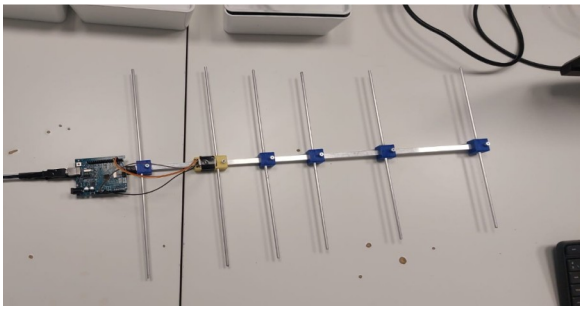


Figura 22: Antena Yagi 6 elementos con receptor incluido

La antena que compone la estación base es una antena Yagi, ya que su direccionalidad y facilidad de construir concordaba con nuestras necesidades. La antena está conformada por 6 elementos y opera en la frecuencia de ~450MHz, hemos elegido esta frecuencia porque se encuentra bastante alto en el rango de canales de nuestro transmisor y así en la competición nuestro canal asignado no influirá mucho en nuestra capacidad de recibir.

El software principal de la estación de tierra está programado en Unity[v] en lenguaje C#. El arduino recibirá los datos sobre radio, y los mandará al portátil, donde se procesarán, visualizarán y finalmente guardarán en un archivo. Usamos Ardity para la comunicación entre Unity y Arduino.

El programa de monitorización consiste en un mapa en el que se ven las últimas posiciones de la lata desde una vista azimutal ortogonal. Para que el mapa tenga una tamaño consistente, cada vez que llega una nueva posición, se guarda el mínimo y el máximo, y luego se utilizan para mapear los valores entre el mínimo y el máximo a los valores mostrados. También se incluye una gráfica que muestra la altura de la lata y, al igual que el mapa, se actualiza cada vez que llega un nuevo valor. Sin embargo, como los valores máximos y mínimos son conocidos, se pueden introducir manualmente. El último elemento son los datos de los sensores de la lata, los cuales se pueden ver en forma de texto.

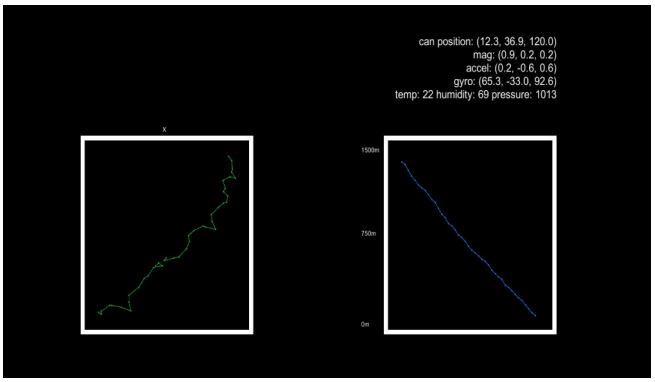


Figura 23: Programa monitorización de telemetría y posicionamiento

3. Planificación

3.1. Planificación del proyecto CanSat

Al ser un proyecto que iba a requerir bastante tiempo, con muchos hitos intermedios a los que debíamos llegar antes de poder afrontar los siguientes desafios, nos creamos unas fechas de referencia para tener una idea de como iba el proyecto.

A continuación se encuentran varios diagramas representado la economización del tiempo disponible.

	Fecha		
Actividad	Inicio	Fin	Duración
Desarrollo del Cansat	16/11/21	09/05/22	174
Decisión de participar	16/11/21	01/12/21	15
Lluvia de ideas	01/12/21	15/12/21	14
Presupuesto y materiales	04/01/22	11/01/22	7
Programación	11/01/22	09/05/22	118
Diseño de la lata	11/01/22	08/03/22	56
Preparación de parapente	09/02/22	09/03/22	28
Diseño del parapente	15/02/22	22/03/22	35
Montado electrónico	20/02/22	26/03/22	34
Prueba de la lata y parapente	22/02/22	08/03/22	14
Practicas de modulo de radio	05/03/22	19/03/22	14
Diseño del CDR	15/03/22	04/05/22	50
Diseño de la antena	29/03/22	12/04/22	14
Prueba de la antena	12/04/22	19/04/22	7

Figura 24: Planificación del proyecto

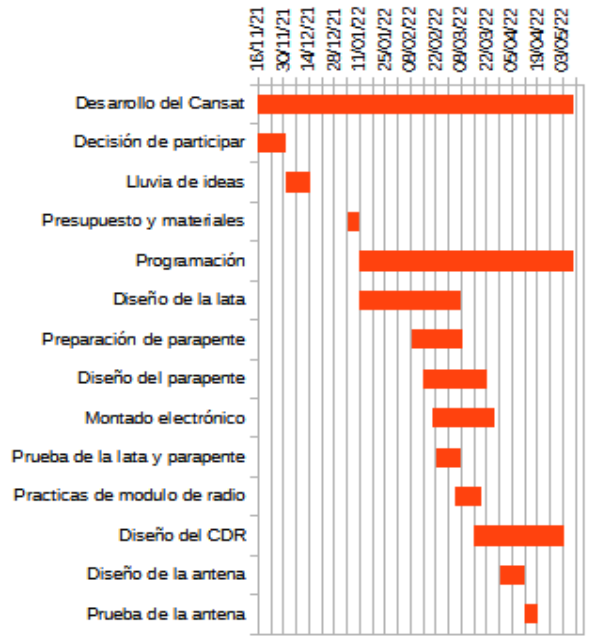
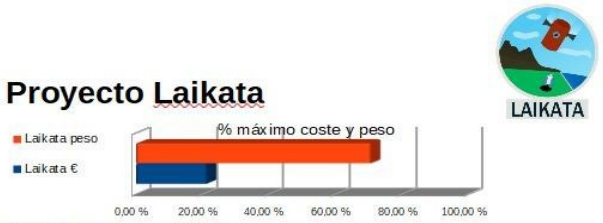


Figura 25: Diagrama de Gantt



Presupuesto

Unidad	Descripción	Coste unitario	Cantidad	Coste total	Peso Unitario	Peso Total
Microcontroladores			1			
WemosD1mini	ESP8266	4,95 €	1	4,95 €	7,00 g	7,00 g
Sensores						
MPU9250	IMU 9 ejes	4,95 €	1	4,95 €	4,00 g	4,00 g
AM2302	Humedad y T	5,95 €	1	5,95 €	8,00 g	8,00 g
BMP180	Presión	2,95 €	1	2,95 €	7,00 g	7,00 g
BN-880	GPS	24,95 €	1	24,95 €	13,00 g	13,00 g
				0,00 €		0,00 g
Actuadores						
FS0403-FB	Servo	6,95 €	2	13,90 €	7,00 g	14,00 g
8x-ws2812	Barra leds	6,10 €	1	6,10 €	5,00 g	5,00 g
Sistemas de alimentación				0,00 €		
Bateria Lipo		6,50 €	1	6,50 €	12,00 g	12,00 g
Battery Shield		4,12 €	1	4,12 €	7,00 g	7,00 g
Conversor 3-5V		3,30 €	1	3,30 €	6,00 g	6,00 g
				0,00 €		0,00 g
Componentes electrónicos discretos						
Placa islas		1,00 €	1	1,00 €	10,00 g	10,00 g
Placa islas Wemos		2,00 €	1	2,00 €	5,00 g	5,00 g
Condensadores		1,00 €	1	3,00 €	3,00 g	3,00 g
Conectores		3,00 €	1	3,00 €	3,00 g	3,00 g
Estructura						
Nylon impern	Parapente	15,00 €	1	15,00 €	15,00 g	15,00 g
Dyneema 1m	Cuerda	0,20 €	15	3,00 €	0,90 g	13,50 g
Carcasa exte	PLA		1	2,00 €	70,00 g	70,00 g
Separadores			4	0,00 €	3,00 g	12,00 g
Varilla 3mm		1,00 €	1	1,00 €	28,00 g	28,00 g
Tuercas varias		1,00 €	1	1,00 €	5,00 g	5,00 g
			Total	106,67 €	Total	247,5
			Laikata €	21,33 %	Laikata peso	70,71 %

Figura 26: Presupuesto y estimación de peso

3.2. Estimación de recursos

3.2.1. Presupuesto

En todo momento se ha intentado que el gasto en materiales fuese el mínimo requerido para hacer el proyecto viable. Esto queda muy bien reflejado en el presupuesto, que siendo de 106'67€ está muy por debajo del máximo permitido.

3.2.2. Apoyo externo

Hemos hablado con los centros educativos de la zona incluyendo colegios e institutos pero estos tienen múltiples dificultades para patrocinar un proyecto de este estilo debido a la escasez de fondos de los que disponen. No hemos recibido por el momento respuestas definitivas, pero la búsqueda de apoyo por parte de instituciones públicas sigue en pie.

Además de lo descrito, se han recaudado 20€ de la madre de uno de los integrantes del equipo que decidió que era un buen momento para patrocinar el proyecto.

La empresa Pineda Electrodomésticos ha considerado que era un proyecto interesante al que patrocinar, así como la academia DiverBOT.

3.3. Pruebas realizadas

Hemos realizado múltiples pruebas con el fin de refinar nuestro proyecto hasta el punto de mayor rendimiento y precisión. Algunas de ellas acabaron en la inutilización parcial de nuestro CanSat, que en esta fase aún no contenía electrónica por lo que no se produjo ninguna pérdida.

3.3.1. Primera prueba

En la primera prueba de nuestro CanSat, elaboramos un módulo de lanzamiento (más detalles a continuación) para poder realizar lanzamientos con un drone.

El propósito de esta prueba era el de comprobar el funcionamiento de nuestro parapente, ya que no había sido probado hasta el momento. Ya que tampoco teníamos el CanSat en un estado funcional, decidimos lanzar una réplica en su lugar. La réplica consistía de la carcasa del CanSat y objetos en su interior que servían de peso (ya que tenía que imitar el peso del Cansat verdadero).

Al intentar volar el drone, tuvimos un problema técnico. Durante el montaje acelerado del drone esa misma tarde, una pieza fue colocada de forma incorrecta. Dicho fallo hizo que una de las hélices se soltara en el momento del despegue, resultando en

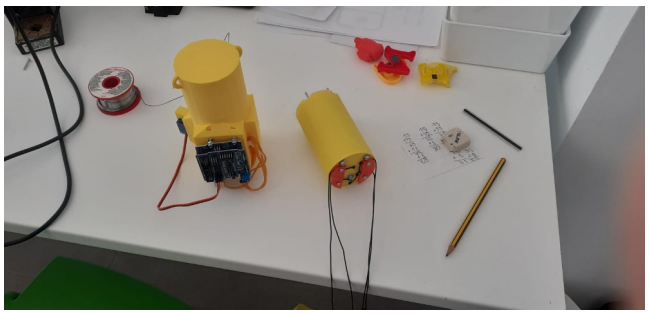


Figura 27: Lanzadera y réplica del cansat

una caída repentina del drone.

Una vez arreglado, despegó sin problemas. Pero otro problema ocurrió. La lanzadera (o módulo de lanzamiento) estaba programada para soltar el CanSat 60 segundos después del lanzamiento, pero, debido a un error en el código de esta no realizó su función. Se bajó el CanSat a tierra firme para poder solucionar el *bug*.

En el tercer intento, por fin se soltó el CanSat. Se desplegó el parapente correctamente y sin enredarse. Pero por desgracia, debido a no estar el parapente bien calibrado, este no voló en línea recta, como era de esperar. Más tarde, ajustando las cuerdas del parapente, se consiguió que el radio de la trayectoria del parapente fuese más grande. Lo dimos por válido, ya que ese pequeño desajuste se podrá compensar con los servos, una vez estos controlen los frenos.

Figura 28: Estado del cansat tras finalizar primeras pruebas

En su último vuelo, el CanSat sufrió una caída sobre una zona de cemento, por lo que acabó destrozado.

Esta prueba fue muy útil ya que nos preparó para las siguientes pruebas al mismo tiempo que nos ayudó a ajustar el parapente para conseguir un vuelo óptimo.

3.3.2. La lanzadera

Para las pruebas desarrollamos el módulo de lanzamiento detallado a continuación.

3.3.2.1. Hardware

El módulo de lanzamiento estaba formado por un microcontrolador *Atmega328*, un servomotor *SG90* y una batería de litio.

Las conexiones electrónicas son bastante sencillas.

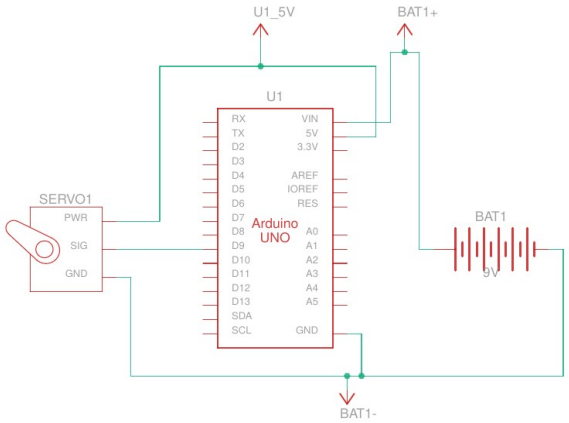


Figura 29: Diagrama lanzadera

Las diferentes partes mecánicas del módulo de lanzamiento fueron impresas con una impresora *3D* y montadas con tornillería, bridas y gomas (que sirven como resortes).

3.3.2.2. Software

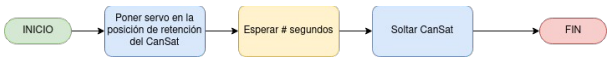


Figura 30: Diagrama de fluco del código de la lanzadera

Para la programación de la lanzadera se utilizó C++ en el framework de Arduino.

El código simplemente se espera 30 segundos (o la cantidad configurada) y acciona el servomotor que libera el CanSat. A continuación se encuentra un diagrama de flujo de dicho código junto con una simplificación de este.

3.3.3. Prueba de distancia de la antena

En la prueba de distancia de la antena se queria averiguar si la antena es capaz de recibir datos a la distancia maxima que llegara el CanSat. En esta prueba se decicio que se intentaria recibir a aproximadamente 1.1 km.

Para realizar esta prueba se colocó una Arduino conectada a una batería y a la misma antena que se encontrara en la lata emitiendo paquetes en el punto A y el resto del equipo nos dirigimos al punto B donde con la antena Yagi conectada a un portatil recibiendo, apuntamos hacia donde se encontraba la otra Arduino para intentar recibir los datos.

Figura 31: Separación TX y RX

En el primer intento no se recibió ningún paquete. Se decidió entonces que dos integrantes se fueran hasta el punto A para ver si el causante era provocado por la Arduino. Se descubrió que el problema si era causado por la Arduino, en especifico que la batería que alimentaba a aquesta se desconectaba automáticamente al pasar cierta cantidad de tiempo.

Tras reactivar esta y volver a intentar la prueba ya se pudo recibir la totalidad de paquetes sin ninguna perdida, también se demostró la gran direccionalidad de la antena Yagi.

4. Programa de difusión y patrocinio

La difusión mediática del proyecto se está realizando a través de la plataforma Instagram y una página web habilitada para el proyecto. A través de Instagram se publicaron imágenes y videos sobre el avance y las pruebas realizadas, mostrando así de forma simple y rápida los objetivos del proyecto al público. Mediante la página web se detalla de mejor forma, gracias a lo que permite poder escribir texto, los avances de la lata en un blog. Además, se explicaba en detalle la misión y quienes somos. Esto permitió ser una entrada, y una forma de enseñar en que consiste Laikata a los patrocinadores.

Para el patrocinio se llevaron a cabo campañas en los colegios, para pedir a estos el patrocinio a cambio de mostrar para el alumnado los objetivos de la misión y cómo llegamos a ello. También se organizó una recolecta de dinero de particulares, donde la gente donaba al proyecto a cambio de participar en un concurso que se realizará una vez acaben los lanzamientos.

Por último se le ofreció a marcas y empresas locales el enseñar su marca en zonas como la camiseta del equipo a cambio de patrocinio.

- La página web: http://laikata.diverbot.es/
- El instagram: https://www.instagram.com/laikata_team/?hl=es
- El GitHub: https://github.com/Laikata

5. Bibliografía, referencias y recursos utilizados

[i] Guillermo Alberto Saez, “Manual del parapentista”, Rev 5.4 . p. 10 [Abstracto]
Disponible en: https://www.facl.es/sites/default/files/documentos/Librodigitaldemanualdeparapente.pdf
[Accedido: 20 de Abril de 2022]

[ii] SingleSking V0.3, actualizado el 7 de Abril de 2013, por Nikolay Rainsiv, Moscow Russia. Programa de código abierto para el diseño de parapentes single sking y cometas.
Dirección: http://www.laboratoridenvol.com/projects/singleskin/sk-nr.html [Accedido: 15 de Abril de 2022]

[iii] PlatformIO, “Professional collaborative platform for embedded development”. [Online]
Dirección: https://platformio.org/ [Accedido: 15 de Abril de 2022]

[iv] Repositorio Laikata, “Repositorio proyecto CanSat 2021-2022 del Laikata Team”, Laikata Team, Denia. Código abierto.
Dirección: https://github.com/Laikata [Accedido: 04 de Mayo de 2022]

[v] Unity, plataforma para creación de videojuegos y animaciones. Programa de pago con versiones gratuitas si se cumplen ciertos requisitos.
Dirección: https://unity.com/ [Accedido: 04 de Mayo de 2022]