



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



**APLICACIÓN DE NUEVAS TECNOLOGÍAS PARA EL  
SEGUIMIENTO DE LA TORTUGA BOBA.  
IMPLEMENTACIÓN DE UN PROTOTIPO PARA SU  
VALIDACIÓN.**

**Autor: Francisco José Sabater Estellés**

**Tutor: José Pelegrí Sebastía**

**Cotutor: Tomás Carlos Sogorb Devesa**

Trabajo Fin de Máster presentado en el  
Departamento de Ingeniería Electrónica de la  
Universitat Politècnica de València para la  
obtención del Título de Máster Universitario en  
Ingeniería de Sistemas Electrónicos

Curso 2021-22

Valencia, marzo de 2022

## Resumen

Este proyecto muestra la importancia de hacer un seguimiento de algunas especies por el hecho de que se encuentran amenazadas y sin la ayuda de la tecnología de seguimiento podrían estar expuestas a la extinción. Es por esto, que se plantea una solución a esta problemática, haciendo una investigación de compañías de comunicación satelital para elegir un *hardware* con el que trabajar. Se incorporan al módulo satelital otros sensores con los que poder obtener distintas medidas, así como la posición global mediante coordenadas, estudiando la problemática que existe y anticipándose en el desarrollo de la solución a los obstáculos que se puedan encontrar en un futuro. Se implementa un código de funcionamiento para el dispositivo, con lo que queda diseñada una primera versión de la solución final, la cual cuenta con comunicación satelital que utilizará para el envío y recepción de los datos obtenidos de los sensores adjuntos en el dispositivo de seguimiento. Se estudian resultados y se hacen medidas de consumo del conjunto para asegurar un buen funcionamiento y para implementar mejoras que puedan alargar la vida útil del dispositivo que en un futuro llevará adherido un animal.

## Resum

Aquest projecte mostra la importància de fer un seguiment d'algunes espècies pel fet que es troben amenaçades i sense l'ajuda de la tecnologia de seguiment podrien estar exposades a l'extinció. És per això, que es planteja una solució a aquesta problemàtica, fent una investigació de companyies de comunicació satel·litària per a triar un *hardware* amb el que treballar. S'incorporen al mòdul satel·litari altres sensors amb els quals poder obtindre diferents mesures, així com la posició global mitjançant coordenades, estudiant la problemàtica que existeix i anticipant-se en el desenvolupament de la solució als obstacles que es pugen trobar en un futur. S'implementa un codi de funcionament per al dispositiu, amb el que queda dissenyada una primera versió de la solució final, la qual compta amb comunicació satel·litària que utilitzarà per a l'enviament i recepció de les dades obtingudes dels sensors adjunts en el dispositiu de seguiment. S'estudien resultats i es fan mesures de consum del conjunt per a assegurar un bon funcionament i per a implementar millores que pugen allargar la vida útil del dispositiu que en un futur portarà adherit un animal.

## Abstract

This project shows the importance of tracking some species because they are threatened and without the help of tracking technology could be exposed to extinction. For this reason, a solution to this problem is proposed, making an investigation of satellite communication companies to choose a hardware to work with. Other sensors are incorporated to the satellite module to obtain different measurements, as well as the global position through coordinates, studying the existing problems and anticipating in the development of the solution to the obstacles that could be found in the future. An operating code is implemented for the device, and a first version of the final solution is designed, which has satellite communication that will be used to send and receive the data obtained from the sensors attached to the tracking device. Results are studied and consumption measurements of the hardware are made to ensure proper operation and to implement improvements that can extend the life of the device that in the future will be attached to an animal.

## Índice

Capítulo 1.	Introducción .....	4
Capítulo 2.	Estado del arte .....	5
2.1	Contexto del proyecto .....	5
2.2	Compañías de telemetría .....	7
2.2.1	Argos .....	7
2.2.2	Globalstar .....	11
2.2.3	Iridium.....	12
2.3	Hardware aplicable a la propuesta.....	12
2.3.1	Argos .....	12
2.3.2	Globalstar .....	14
2.3.3	Iridium.....	17
2.4	Empresas del sector.....	20
2.4.1	Wildlife Computers .....	20
2.4.2	Sirtrack/Lotek.....	21
Capítulo 3.	Diseño e implementación del proyecto .....	22
3.1	Necesidades del proyecto .....	22
3.2	Propuesta.....	25
3.3	Sensores del integrado.....	26
3.3.1	KIM1 .....	26
3.3.2	Módulos GPS GY-GPS6MV2 y L86-M33 .....	27
3.3.3	RTC RV-3028-C7 .....	29
3.3.4	Sensor de temperatura MCP9808.....	30
3.3.5	Higrómetro HD-38 .....	30
3.4	Montaje del módulo de seguimiento completo .....	31
3.5	Implementación del código .....	33
3.6	Diseño de módulo de seguimiento .....	39
3.6.1	Esquemático .....	39
3.6.2	Diseño PCB.....	44
Capítulo 4.	Resultados .....	49
4.1	Medidas de consumo.....	49
4.2	Pruebas de funcionamiento .....	58
4.3	Pruebas de campo.....	60
Capítulo 5.	Trabajos futuros.....	63
Capítulo 6.	Conclusión.....	65
Capítulo 7.	Referencias .....	66

## Tabla de figuras

Fig. 2.1 (a) Zonas frecuentadas por las tortugas (b) Comparación de las zonas frecuentadas por las tortugas con la Red Natura 2000.....	6
Fig. 2.2 Etapas del procesamiento de datos.....	8
Fig. 2.3 Procesado de distintos tipos de mensajes.....	9
Fig. 2.4 Tratamiento y distribución de los datos .....	11
Fig. 2.5 (a) KIM1 (CLS Argos) (b) Diagrama de bloques del KIM1 .....	13
Fig. 2.6 Diagrama de bloques KIM1 .....	13
Fig. 2.7 Mapa de cobertura de Globalstar .....	14
Fig. 2.8 STX3 (Globalstar).....	15
Fig. 2.9 STINGR (Globalstar).....	17
Fig. 2.10 Iridium 9602 (Iridium).....	18
Fig. 2.11 Iridium 9603 (Iridium).....	19
Fig. 2.12 16130 Adaptador Mini-PCIE (Iridium).....	20
Fig. 3.1 KIM1 Shield .....	26
Fig. 3.2 GPS GY-GPS6MV2 .....	27
Fig. 3.3 Conexión GY-GPS6MV2 .....	28
Fig. 3.4 GPS L86-M33.....	28
Fig. 3.5 RTC RV-3028-C7.....	29
Fig. 3.6 Sensor de temperatura MCP9808 .....	30
Fig. 3.7 Higrómetro HD-38.....	31
Fig. 3.8 Dispositivo de seguimiento completo .....	32
Fig. 3.9 Codificación de datos.....	34
Fig. 3.10 Inicialización del RTC .....	35
Fig. 3.11 Función bucle.....	35
Fig. 3.12 Interrupción por temporizador .....	36
Fig. 3.13 Máquina de estados.....	37
Fig. 3.14 Función KIM1.....	38
Fig. 3.15 Franja horaria de paso de satélites .....	38
Fig. 3.16 Esquemático KIM1 .....	39
Fig. 3.17 Esquemático LDO.....	40
Fig. 3.18 Esquemático sensor de temperatura y RTC .....	41
Fig. 3.19 Esquemático GPS L86 M-33 .....	41
Fig. 3.20 Esquemático higrómetro .....	42
Fig. 3.21 Esquemático microcontrolador ATMEGA328P .....	43
Fig. 3.22 Vista frontal del PCB .....	45

Fig. 3.23 Vista trasera del PCB .....	45
Fig. 3.24 Cálculo de la impedancia de pista del KIM1 .....	46
Fig. 3.25 PCB fabricada .....	47
Fig. 3.26 PCB montado .....	48
Fig. 4.1 Medida de consumo con multímetro digital .....	49
Fig. 4.2 Medida de consumo del módulo GPS (50.000 muestras) .....	50
Fig. 4.3 Medida de consumo del módulo GPS GY-GPS6MV2 (1.000 muestras) .....	51
Fig. 4.4 Medida de consumo del RTC RV-3028-C7 (50.000 muestras) .....	52
Fig. 4.5 Medida de consumo del sensor de temperatura MCP9808 (45.000 muestras) .....	53
Fig. 4.6 Medida de consumo del sensor de temperatura MCP9808 (5.000 muestras) .....	53
Fig. 4.7 Medida de consumo del sensor de humedad HD-38 (45.000 muestras) .....	54
Fig. 4.8 Medida de consumo del dispositivo (50.000 muestras) .....	55
Fig. 4.9 Medida de consumo del dispositivo (Un ciclo) .....	56
Fig. 4.10 Medida de consumo del dispositivo (Un envío) .....	56
Fig. 4.11 Medida de consumo del dispositivo con cambio en la potencia de envío (50.000 muestras) .....	57
Fig. 4.12 Pruebas de envío de datos .....	58
Fig. 4.13 Aplicación Argos Web .....	60
Fig. 4.14 Pruebas de campo del dispositivo (1) .....	61
Fig. 4.15 Pruebas de campo del dispositivo (2) .....	61

## Capítulo 1. Introducción

En estos últimos años el cambio climático está siendo un problema notable en el planeta Tierra, y como consecuencia muchas especies de seres vivos se están viendo afectados por cambios en sus hábitats que muchas veces conllevan la extinción de dichas especies.

Una de las especies que se está viendo afectada gravemente por esta variación global del clima de la Tierra es la tortuga boba (*Caretta caretta*), una especie de la familia de las tortugas marinas que ahora mismo está en peligro de extinción debido sobre todo a la acción humana y al cambio climático. Son muchas las amenazas que sufre esta especie, entre ellas las redes de pesca abandonadas denominadas redes fantasma, el vertido descontrolado de plástico en océanos y mares que confunden con comida, la explotación de las playas donde anidan, la destrucción de sus hábitats y las variaciones anuales de las temperaturas climáticas que les hacen cambiar de zona, entre otras cosas. Recientemente e inusualmente, algunas tortugas han empezado a poner sus huevos en las costas del Mediterráneo Occidental por lo que, para protegerlas es necesario incorporar en su caparazón un dispositivo de seguimiento con el que poder conocer sus hábitos y saber cuándo desovan.

Con todo esto, la idea del proyecto es diseñar un dispositivo de seguimiento de tortugas mediante un sistema embebido que disponga de un microprocesador central, sensores periféricos y un *GPS*.

En un primer lugar, se hará una investigación de compañías que ofrezcan servicios de geolocalización, para de esta manera poder usar estos servicios en el módulo que se diseñe. A continuación, se propondrá un microcontrolador, así como los distintos sensores que recogerán los datos necesarios para hacer el seguimiento de la tortuga y se prepararán para su funcionamiento mediante el uso del microcontrolador. Con esto, se implementará un código que se adapte y tenga en cuenta las condiciones en las que va a trabajar el dispositivo al ser adherido a la tortuga. Además, se realizará el diseño del dispositivo de seguimiento una vez se hayan elegido los distintos integrados que se van a utilizar y se fabricará una primera versión de este. Por último, se harán medidas de consumo del módulo intentando minimizarlo para de esta manera alargar la vida útil de la batería lo máximo posible, y algunas pruebas tanto en el laboratorio como pruebas de campo para poder asegurar el correcto funcionamiento del módulo de seguimiento.

## Capítulo 2. Estado del arte

En este apartado, y antes de empezar a diseñar una solución para el proyecto que nos ocupa, que es el seguimiento de tortugas, se va a realizar un trabajo de investigación con el objetivo de conocer la actualidad sobre esta materia y estar de esta manera bien informados sobre las opciones disponibles actualmente para obtener una solución lo más óptima y actualizada posible.

### 2.1 Contexto del proyecto

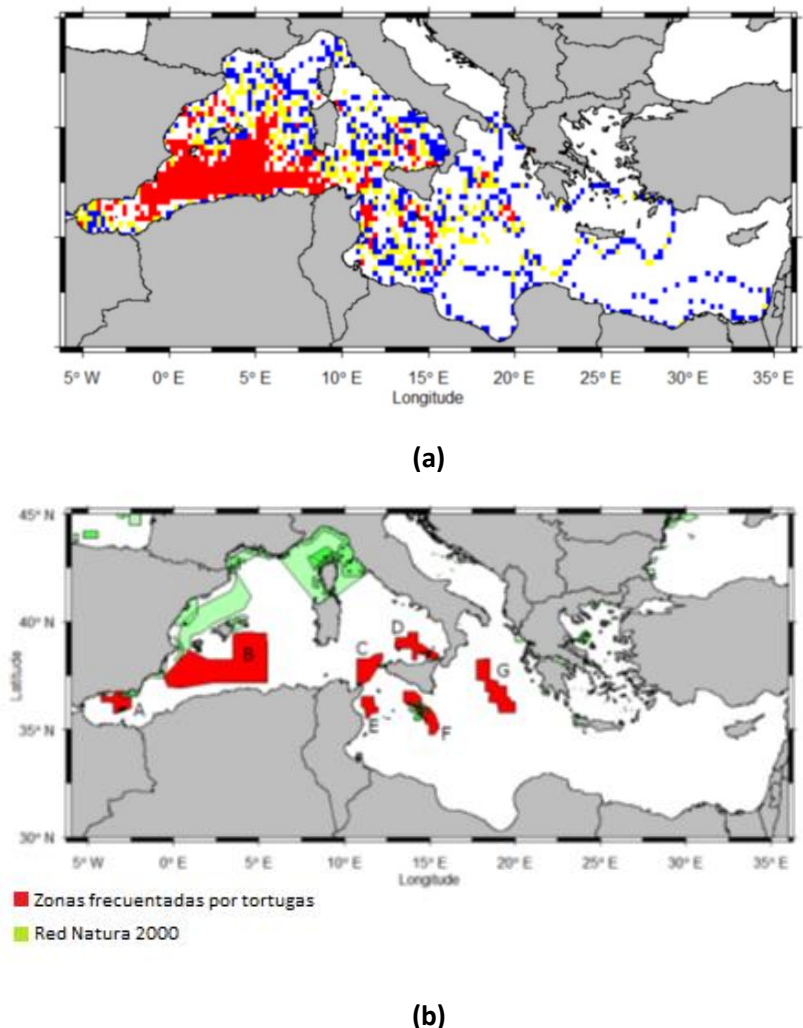
En la actualidad, la tortuga boba (*Caretta caretta*), una especie de tortuga marina que habita en el océano Atlántico, Pacífico e Índico y en el mar Mediterráneo, está considerada una especie en peligro de extinción. Entre los factores que amenazan a esta especie, tenemos la acción humana, que perjudica la reproducción de estos animales por el hecho de que se han invadido y destruido muchas de las zonas donde las tortugas anidan. Las tortugas ponen sus huevos en playas, y estos animales necesitan grandes extensiones de arena donde desovar, y debido a las construcciones y la sedimentación que estas producen, cada vez el hábitat natural de puesta de huevos de estos animales es más escaso. También, a causa del aumento del calentamiento global y de las variaciones de temperatura que este produce, el género de los neonatos se ve afectado. El género de las crías de tortuga cambia según la temperatura, con una temperatura elevada la probabilidad de que las crías sean hembra es mucho mayor, y al contrario si la temperatura es baja, entonces hay más probabilidad de que las crías sean macho.

Por esto y por otras razones, desde hace unas décadas las tortugas bobas han empezado a poner sus huevos en el mediterráneo occidental, lugar en el que antes no desovaban estos animales. Desde 2011, todos los años se han encontrado nidos, y al contrastar datos anteriores hay pocos registros de anidación en esta zona en los últimos siglos. Es por esto, que se ha llegado a la conclusión de que esto es una consecuencia del cambio climático, y esta especie de tortugas ha encontrado un nuevo hábitat de anidación, por lo que conservarlo y protegerlo es de vital importancia para la conservación de la especie *Caretta caretta*.

Mediante el seguimiento vía satélite de las tortugas que aniden en las costas del occidente mediterráneo, es posible registrar componentes demográficos importantes en la vida y rutas de estos animales a tiempo real, como el área en la que pasan sus vidas, movimientos a zonas donde se alimentan, periodos de inactividad o la época de anidación, todos ellos afectados por el cambio climático y su consiguiente elevación de la temperatura del mar. Con estos datos, se podría tener un mejor control de los ejemplares de esta especie, sabiendo en todo momento donde se encuentran y el momento crítico del desove. Además, son datos muy relevantes para poder hacer un estudio del efecto que tiene el cambio climático en estos animales cuando se llegue a reunir un gran volumen de datos de seguimiento, puesto que existen numerosos estudios de la fase de apareamiento terrestre de estos animales, pero pocos de cómo les ha afectado en el mar u océano, y esto es debido al desconocimiento de la actividad de las tortugas en este medio. Con los datos que se reúnan se puede hacer una comparación del presente y años anteriores para de este modo poder pronosticar y disminuir los efectos adversos que puedan sufrir.

Por todo esto, mediante este trabajo la Universidad Politécnica de Valencia pretende ayudar a poner una solución dentro del proyecto europeo *LIFE INTEMARES* [1] de conservación del medio marino, una iniciativa que integra diversos fondos y políticas con el objetivo de conseguir la gestión de toda una red de espacios protegidos de la *Red Natura 2000* [2] dirigido por el Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente de una manera eficaz. Durante el año 2021, LIFE INTEMARES ha sido participe del marcaje de 10 crías de tortuga boba en la costa mediterránea, de manera que con el seguimiento de estos especímenes se consiga asegurar la conservación de la especie, protegiendo sus nidos y estudiando los movimientos de la especie para conocer mejor a estos animales.

Además, con los datos de geolocalización de las tortugas que se marquen con el dispositivo que se diseñe en este proyecto, será posible hacer una reestructuración o adición de los espacios protegidos de la *Red Natura 2000*, por el hecho de que como hemos comentado anteriormente, las tortugas están cambiando sus hábitats en estos tiempos, y es de vital importancia para este tipo de especies poseer datos actualizados en todo momento de sus rutas y movimientos. En la siguiente figura se muestran las zonas en las que pasan la mayor parte de sus vidas las tortugas bobas en comparación con la *Red Natura 2000*:



**Fig. 2.1 (a) Zonas frecuentadas por las tortugas (b) Comparación de las zonas frecuentadas por las tortugas con la Red Natura 2000**



En la Fig. 2.1 (a) Zonas frecuentadas por las tortugas (b) Comparación de las zonas frecuentadas por las tortugas con la Red Natura 2000, se puede apreciar como las zonas actuales de la Red Natura 2000 no coinciden con las zonas en las que esta especie es más frecuente que habite, y con una monitorización de más ejemplares de esta especie se podrá incluso obtener un mapa más detallado de las zonas en las que estos animales se mueven, además de la posibilidad de hacer una reforma de la actual área protegida para una conservación más eficaz de las tortugas bobas del Mediterráneo.

LIFE INTEMARES tiene como fecha límite el 2024, fecha hasta la que llevará a cabo un amplio programa de acciones que tendrán que ver con la investigación, el seguimiento y vigilancia, la conservación, la gobernanza y la participación, así como la comunicación, sensibilización y educación ambiental. La parte del proyecto que se trata en este trabajo, va a ser el seguimiento y vigilancia de la especie, y va a ser posible gracias a la financiación de LIFE INTEMARES por el hecho de que ha puesto a disposición de la universidad un presupuesto para poder llevar a cabo las labores de investigación y de diseño. Para ello, se va a hacer un estudio de compañías de telemetría con las que será posible conseguir la localización de los ejemplares de tortuga boba que se marquen con el dispositivo de seguimiento mediante comunicación satelital.

## 2.2 Compañías de telemetría

En este apartado se investigarán las diferentes empresas del sector, así como los servicios de tratamiento y visualización de datos de seguimiento de animales. Mencionar, que algunas de las empresas que se mostrarán a continuación se dedican a tareas de seguimiento, pero no necesariamente de animales. La idea es dar a conocer ampliamente el sector del seguimiento mediante localización GPS, así como profundizar en alguno de los *hardware* que tienen disponibles las compañías de manera que mediante comparación podamos llegar a entender sus características y cuál es el más adecuado para llegar a la solución que se adapte mejor al proyecto que se está realizando.

### 2.2.1 Argos

Argos [3] es un sistema creado en 1978 por la Agencia Espacial Francesa (CNES), la Administración Nacional de Aeronáutica y del Espacio (NASA) y la Administración Nacional Oceánica y Atmosférica (NOAA) con la finalidad de recolectar datos y ubicación con el objetivo de estudiar y proteger el medio ambiente.

En 1986, se crea CLS (*Collecte Localization Satellite*) [4] siendo en la actualidad los proveedores exclusivos de servicios Argos para aplicaciones científicas y medioambientales. Además, recientemente otras agencias espaciales como la Organización Europea de Explotación de Satélites Meteorológicos (EUMETSAT), la Organización de Investigación Espacial de la India (ISRO) y otras han empezado a participar activamente en el sistema Argos.

La constelación de satélites con la que cuenta Argos consta de siete satélites en una órbita baja a 850 km de la superficie terrestre, y estos completan una vuelta alrededor del planeta Tierra aproximadamente cada 100 minutos, cubriendo una superficie de 5000  $km^2$  cada uno de ellos. Las aplicaciones del sistema *Argos* son varias, entre ellas cabe destacar la monitorización ambiental, que incluye tanto el seguimiento de animales como la oceanografía, para el estudio de las aguas y los fondos marinos desde el punto de vista físico, químico y biológico, y también por otro lado la climatología.

Por otro lado, tenemos que además se usa para la gestión de los recursos marinos mediante la gestión y seguimiento de las áreas marinas protegidas, la monitorización de la pesca para hacer frente a la pesca ilegal y apoyar el desarrollo de la pesca artesanal.

También se utiliza para temas relacionados con la seguridad marítima, haciendo un seguimiento de los transportes marítimos y evitando de este modo la piratería, así como también tiene aplicaciones para la seguridad en eventos deportivos como las regatas.

Por último, tiene aplicación en la monitorización de la contaminación marítima y de equipo que a su vez se encarga de monitorizar en el mar.

En cuanto al tratamiento y la visualización de datos en *Argos*, los mensajes que se van transmitiendo entre *tags*, satélites y estaciones terrestres son secuencias de dígitos binarios que contienen los datos. Posteriormente se procesan los mensajes para ser distribuidos por el sistema *Argos*. En la siguiente figura tenemos un esquema de las diferentes etapas del procesamiento de los datos en el sistema de *Argos*:

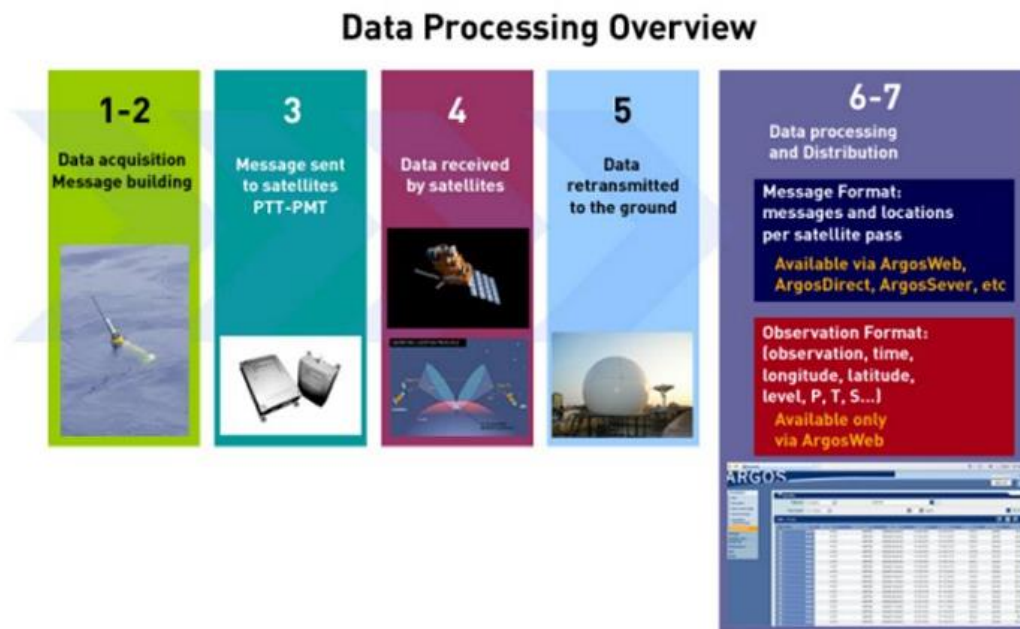


Fig. 2.2 Etapas del procesamiento de datos

Como podemos ver, las primeras etapas son la de adquisición de los datos (etapa 1), en la que el *tag* recoge los datos de rastreo, y la de creación de los mensajes (etapa 2) donde los datos recogidos son fechados, codificados y escritos en mensajes.

En la siguiente etapa, los mensajes son enviados al satélite (etapa 3), y este envío se puede realizar mediante un transmisor *Argos (PTT)* o mediante un módem *Argos (PMT)*. A continuación, los mensajes llegan al satélite (etapa 4), donde son fechados y se mide la frecuencia de la señal para poder diferenciarlos unos de otros. Los satélites retransmiten los mensajes a estaciones terrestres (etapa 5), cada uno diferenciado por la fecha y la frecuencia de su señal. Una vez llegan los mensajes, se calcula la posición del *tag* procesando las frecuencias que se han recibido mediante el efecto *Doppler* o extrayendo las posiciones *GPS* (etapa 6) que se encuentran contenidas en los datos del mensaje, y se decodifican. Por último, los datos que ha obtenido el *tag* se extraen de los mensajes, son fechados, geolocalizados y validados (etapa 7).

Los resultados para el usuario pueden tener dos formatos según se requiera. El primer formato es el de mensaje, en el que los resultados son agrupados por el satélite y distribuidos al usuario juntos con la localización que ha obtenido el satélite. Y el otro formato es el de observación, en el que los datos físicos obtenidos por el *tag* son fechados y se les asigna una localización según el momento en el que fueron obtenidos y con la mejor aproximación de localización posible.

El sistema de *Argos* es capaz de manejar varios formatos de mensajes, incluso para un mismo transmisor, y esto se debe a que cada formato tiene una estructura específica. A continuación, en la siguiente figura podemos ver cómo funciona el sistema de procesamiento de mensajes para un único *tag* que envía distintos tipos de mensajes al sistema de *Argos*:

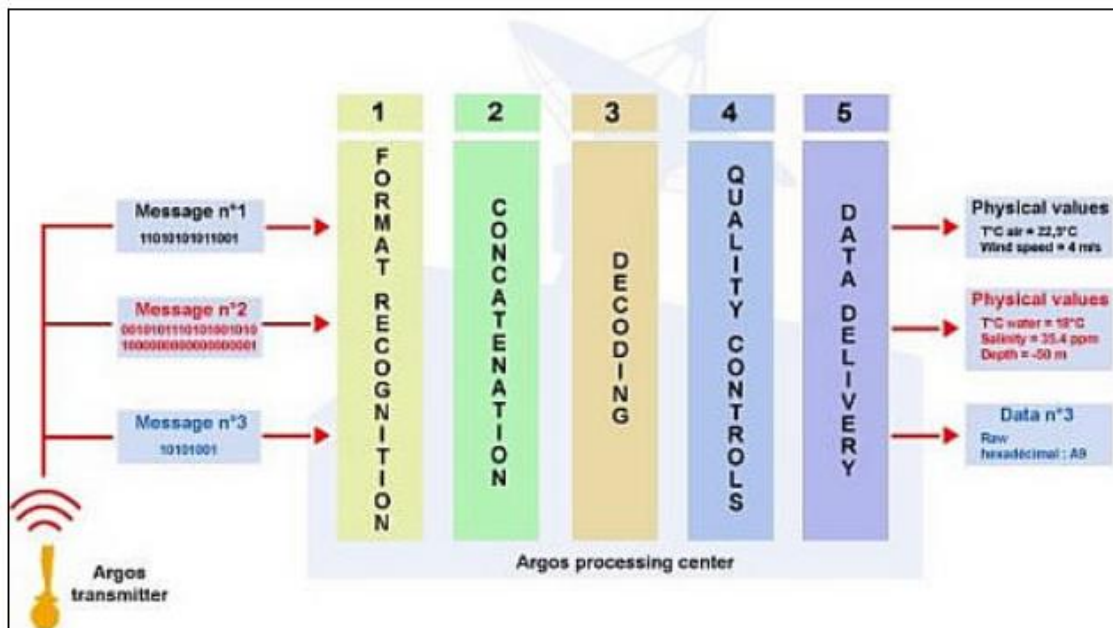


Fig. 2.3 Procesado de distintos tipos de mensajes

Cuando el sistema de *Argos* reconoce un mensaje binario que viene con una identificación para diferenciarlo, dicho sistema procede a reconocer el formato del mensaje para poder operar con este. En este paso se pueden aplicar distintos procedimientos.

Por una parte, *Argos* puede distinguir los mensajes de un *tag* específico y entonces aplicar sistemáticamente el mismo procedimiento a todos estos mensajes, de este modo es posible procesar mensajes de manera idéntica, aunque estos tengan diferente longitud.

Por otra parte, existe la posibilidad de reconocer el formato de mensaje mediante su longitud y *Argos* procesa todos los mensajes con esta longitud del mismo modo. Existe un formato predeterminado de manera que, si la longitud del mensaje es diferente a la declarada, estos datos se distribuyen en palabras hexadecimales de 8 bits.

Por último, tenemos el método de reconocimiento del formato del mensaje mediante una ecuación del tipo  $f(Ax + By + C)$ , donde  $x$  e  $y$  son valores codificados en una parte del mensaje. En el caso de que no se reconozca una ecuación en el mensaje, se aplica el formato predeterminado y los datos se distribuyen en palabras hexadecimales de 8 bits.

Una vez se ha aplicado el formato del mensaje, la siguiente operación que se realiza es la de segmentar el mensaje según el formato. Los segmentos vienen definidos por su posición en el mensaje, la longitud en bits y el tipo de decodificación binario.

Cada segmento es denominado elemento binario y existen tres tipos. En primer lugar, tenemos el elemento estándar, que se basa en una serie de bits que al ser decodificados proporciona un valor o una cadena de caracteres. Por otro lado, tenemos el elemento *checksum*, que es usado para validar la integridad de un mensaje por el hecho de que pueden aparecer errores de bits al transmitir los mensajes a través del espacio cuando estos son transmitidos del *tag* al satélite y de este a las estaciones de procesamiento terrestres. Este elemento funciona mediante un algoritmo de codificación en el que el *tag* realiza una suma y esta es codificada en el mensaje. En el centro de procesamiento de *Argos* y con el mismo algoritmo se decodifica esta suma, y en el caso de que las dos sumas son idénticas se considera que el mensaje es válido, además es posible codificar varias de estas sumas para la comprobación de los mensajes. Y como última opción de segmentación de los mensajes, tenemos el elemento bloque, que básicamente es un conjunto de elementos estándar repetidos  $n$  veces. Los elementos bloque son de longitud fija y como se ha mencionado repetidos  $n$  veces, conteniendo uno o más elementos estándar. Se suele usar para declarar conjuntos de medidas repetidos varias veces en el mensaje.

Una vez obtenidos los datos, lo siguiente es la decodificación de estos. En este proceso se convierte el valor binario del mensaje sin procesar en un valor físico y cuantificable. En primer lugar, se decodifican los elementos binarios. En segundo lugar, mediante una función de transferencia se restauran los datos que se calculan a partir de dos elementos binarios y se definen con un formato de salida que puede ser decimal, hexadecimal o científico.

También hay disponible otro método de procesamiento de los datos a parte del procesamiento de los mensajes recibidos, que es el procesamiento de observación, y que es independiente, es

decir, no interfiere en el procesamiento del mensaje por lo que el formato de mensaje seguiría estando disponible para el usuario en los servicios de distribución de *Argos* (*ArgosDirect*, *ArgosServer* o *ArgosWeb* entre otros). En la siguiente figura podemos ver el proceso y como están separados un procesamiento de otro:

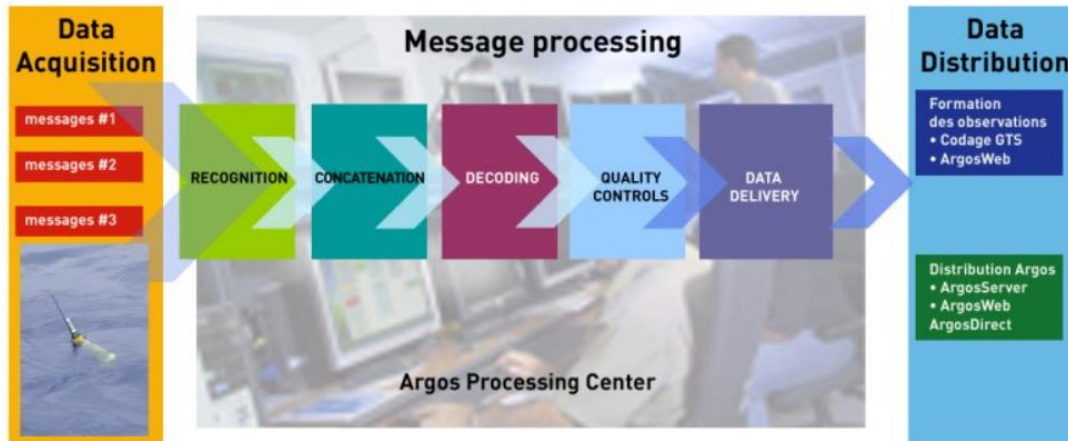


Fig. 2.4 Tratamiento y distribución de los datos

Con el procesamiento de observación es posible obtener los datos físicos geolocalizados y fechados que ha realizado un *tag* en un formato simplificado y listo para interpretar. Estos datos se ordenan cronológicamente y luego por nivel de la medida física que obtengamos. Además, con ese método se reduce considerablemente la cantidad de datos distribuidos por el hecho de que se realizan controles automáticos y se ignoran los mensajes redundantes.

### 2.2.2 *Globalstar*

*Globalstar* [5] es una compañía que suministra tecnología vía satélite de manera personalizada. Ofrece soluciones como conectividad y rastreo para dispositivos móviles de campo y control de equipos, a través de la Red Satelital *Globalstar* de la que disponen, que es capaz de captar señales de más del 80% de la superficie terrestre. Los satélites de esta red transmiten las señales mediante tecnología *CDMA* (*Code Division Multiple Acces*), con la que, por un único canal de transmisión, y utilizando tecnología de espectro expandido y una distintiva forma de codificación, es posible diferenciar los mensajes que se envían, aunque se envíen todos por el mismo canal de comunicación, sin que aparezcan interferencias entre los mensajes. Esto se consigue asignando un código único a cada transmisor de manera que cuando el receptor recibe la señal sea capaz de distinguir la que le interesa, aunque las señales transmitidas compartan la misma frecuencia. *Globalstar* ofrece comunicación ininterrumpida gracias a su tecnología que ofrece diversidad de rutas y conexiones de satélite, de manera que es posible conectarse automáticamente de un satélite a otro.

La infraestructura de la compañía está formada por 24 estaciones terrestres y por una red de satélites situados en la órbita terrestre baja, más conocida como *LEO*, y a través de la cual ofrecen comunicación a más de 120 países del mundo. Además, poseen una infraestructura terrestre de segunda generación, centrada en la configuración del subsistema *Internet Media Services* (*IMS*) [6], lo que proporciona una mejor comunicación con los usuarios por el hecho de

que es líder en comunicaciones digitales y en marketing, representando a plataformas como *LinkedIn*, *Twitter*, *Twitch*, *Warner Music*, etc. Al asociarse con *IMS*, las empresas experimentan un rápido desarrollo y una expansión de sus mercados.

La compañía se centra en proporcionar servicios de calidad a bajos costes, además de dar viabilidad a aplicaciones en zonas remotas sin que haya problemas de cobertura intermitente. Su uso, mayoritariamente se centra en los sectores de energía, agricultura, silvicultura, gobierno, transporte, construcción y marítimo, pero no deja de ser una opción muy viable gracias a que los productos que ofrece se pueden adaptar a las necesidades del usuario.

### 2.2.3 *Iridium*

La compañía *Iridium* [7] ofrece conectividad con cobertura en todo el globo terrestre mediante la constelación de 66 satélites en la órbita *LEO*, de manera que tanto personas como dispositivos se mantienen comunicados en tierra, aire y mar. Su red de satélites conecta cada satélite con hasta otros cuatro que permite un fácil enrutamiento del tráfico de datos y garantiza una cobertura global incluso en entornos remotos. En 2019 se llevó a cabo una mejora de la constelación de satélites y de la infraestructura de las bases terrestres, lo que supone una considerable mejora para las comunicaciones en entornos críticos y/u hostiles en cuanto a conectividad.

A parte de otros sectores en los que se especializa la compañía, *Iridium* ofrece cada vez más soluciones innovadoras centradas en sensores remotos, monitores, seguimiento, computación en la nube y control de sistemas autónomos. De esta manera la empresa se adapta a la nueva era *IoT* (*Internet of Things*).

## 2.3 Hardware aplicable a la propuesta

### 2.3.1 *Argos*

Entre los distintos productos *hardware* que ofrece la compañía, el más eficiente y asequible en la actualidad es el que *KIM1*, pero cabe decir que próximamente lanzarán al mercado un nuevo módulo con una mejora considerable en la tecnología satelital.

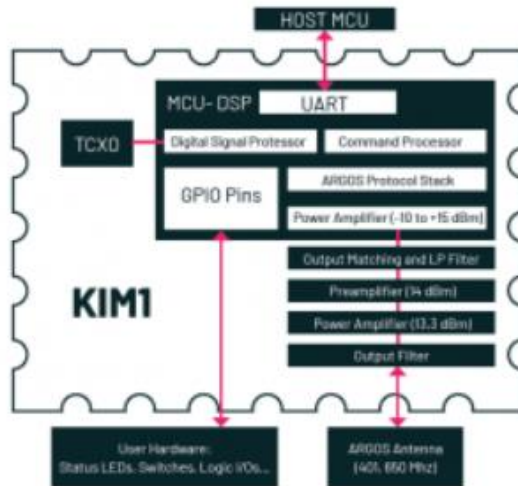
#### 2.3.1.1 *KIM1*

El hardware *KIM1* [8] se trata de un módulo transmisor basado en formas de onda *ARGOS-2* con el que se puede disponer de comunicación global gracias a su conexión con la constelación de satélites *ARGOS/Kinéis*. Este dispositivo proporciona datos y la localización mediante el efecto *Doppler*, por lo que no es necesaria la tecnología *GPS*. La transmisión de este módulo es unidireccional, transmite información a los satélites y estos la procesarán.

En la siguiente ilustración tenemos tanto la placa *KIM1*, así como un diagrama de bloques del interior de esta:



(a)



(b)

Fig. 2.5 (a) KIM1 (CLS Argos) (b) Diagrama de bloques del KIM1

Las dimensiones del *hardware* son de 31,2mm x 21mm x 3,5mm. Del diagrama de bloques cabe destacar que el KIM1 tiene integrado un *UART (Universal Asynchronous Receiver-Transmitter) AT* para el control de dispositivos y puertos que se conecten. Además, también tiene una serie de pines *GPIO (General Purpose Input/Output)* que pueden ser utilizados tanto como entrada o salida para diferentes propósitos. La antena incorporada en el KIM1 es una antena *Argos* que trabaja a una frecuencia entre 399,910 MHz y 401,680 MHz. La tarjeta KIM1 facilita la incorporación de un módem satelital a sus proyectos gracias a su forma *mini PCIe* y su interfaz *USB 2.0* que podemos ver desglosada en la siguiente figura:

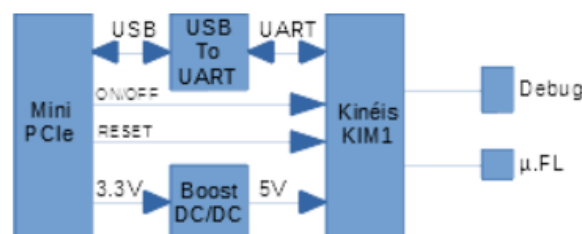


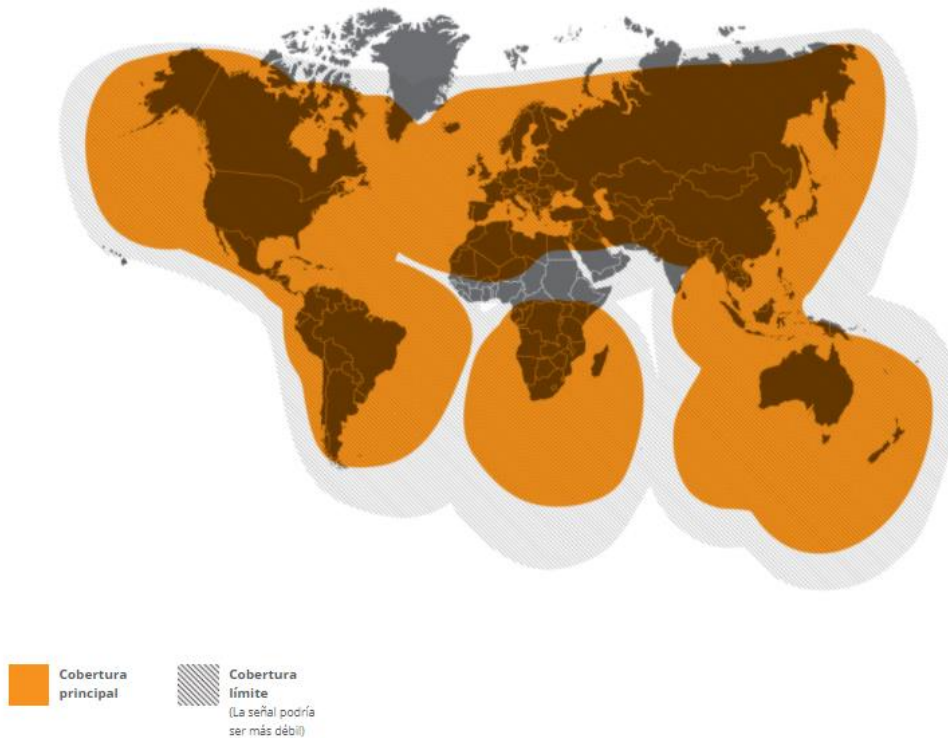
Fig. 2.6 Diagrama de bloques KIM1



Dispone de certificación por *Kinéis* y la Agencia Espacial Francesa (*CNES*), y está diseñado con una interfaz para su fácil integración en dispositivos finales. La misma compañía de telemetría ofrece una placa que es específica para la integración del *KIM1*, denominada *KIM1 Shield* [9], que a su vez se puede incorporar a una plataforma de desarrollo que puede ser *Arduino*, *STM32 Nucleo* o que también se puede conectar a un *PC* para poder hacerla funcionar como una placa de interfaz. Las dimensiones de tarjeta de interfaz del *KIM1* son de 8,5 cm x 5,56 cm x 1,8 cm.

### 2.3.2 *Globalstar*

Entre los productos que ofrece la compañía *Globalstar* para las soluciones que implican tecnología vía satélite, podemos destacar algunos modelos interesantes para el proyecto, teniendo la posibilidad de integrarlos al *hardware* del que disponemos. Estos ofrecen una cobertura en prácticamente todo el globo terrestre como podemos comprobar en el siguiente mapa de cobertura:



**Fig. 2.7** Mapa de cobertura de *Globalstar*

#### 2.3.2.1 *STX3*

El *STX3* [10] es un transmisor por satélite compacto y asequible, con un consumo de batería reducido lo que lo hace en gran medida eficiente para su integración. Este componente envía mensajes de datos unidireccionales cuando funciona aplicado a un dispositivo de rastreo con la capacidad de trabajar remotamente.



Este dispositivo *hardware* dispone de las siguientes características que lo hacen una buena opción para su integración en un equipo de seguimiento:

- Realiza múltiples transmisiones, aumentando así su fiabilidad.
- Cobertura global.
- Bajo consumo de energía.
- Diseño para un mejor montaje en una superficie.
- Uso versátil en diversos tipos de aplicaciones.

Al diseñar utilizando el *STX3*, se deben tomar las medidas necesarias para realizar pruebas reglamentarias, incluidas pruebas de *RF* continuas para mediciones espectrales y de potencia. Todos los productos que utilizan el módulo *STX3 Simplex* requieren la certificación de la red *Globalstar*. Por lo que es altamente recomendable que durante el desarrollo de una solución utilizando este hardware se certifique con el equipo de *Globalstar*.

El componente tiene un tamaño de 28,7 mm x 20,57 mm, y cuenta con un grosor de 4,13 mm incluyendo los componentes. En la siguiente imagen tenemos una vista general del componente:



**Fig. 2.8 STX3 (Globalstar)**

Además, cuenta con la certificación modular *FCC (Federal Communications Commission) CFR (Code of Federal Regulations) parte 25* [11], que hace referencia a las comunicaciones vía satélite. Así como el certificado por parte de la Industria de Canadá, y la certificación *CE*.

La tecnología mediante la que funciona el *STX3* es a través de la red de datos *Simplex* de *Globalstar*, por lo que sabemos que la transmisión va a ser unidireccional, es decir, única y de una sola dirección. El *tag* transmitirá información a los satélites y estos la procesarán. La solución *OEM (Original Equipment Manufacturer)* que ofrece *Globalstar* incluye:

- Una unidad de transmisión *Simplex (STX-3)*.
- Sensores específicos para la transmisión de información a los satélites.
- Un panel de control de interfaz.
- Fuente de alimentación con *CA*.
- Reloj en tiempo real.
- Antena de conexión.

En cuanto al funcionamiento, el *hardware STX3* dispone de cuatro modos de trabajo programables dependiendo del uso que se le quiera dar al *tag* en el que esté implantado. A continuación, se detallan estos modos de operación del *STX3*:

#### **MODO SLEEP (SUSPENSIÓN)**

Se aplica tensión a la unidad, pero no hay transmisiones pendientes ni actividad en serie.

#### **MODO ACTIVO**

El *STX3* está activo y responde al puerto serie, pero no transmite.

#### **MODO STANDBY (ESPERA)**

El *STX3* está inactivo entre ráfagas de transmisión, pero no está transmitiendo.

#### **MODO DE TRANSMISIÓN**

La unidad está transmitiendo un paquete de *RF*.

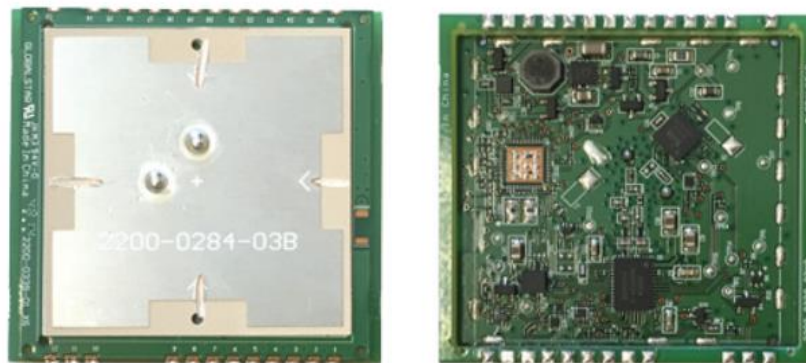
### **2.3.2.2 STINGR**

El *STINGR* [12] es la solución que presenta *Globalstar* para el rastreo, integrando en un único módulo un transmisor por satélite con receptor *GPS*. Este dispositivo *OEM* envía información unidireccionalmente mediante la red de datos *Simplex* de *Globalstar* y su receptor *GPS* de alto rendimiento le permite la transmisión de datos incluso en áreas donde no se dispone de cobertura móvil en todo el planeta. Además, el módulo incluye una antena de doble banda, lo que simplifica la integración de las comunicaciones por satélite para los dispositivos con fines de rastreo o monitorización.

Las características que hacen de este módulo una opción a considerar para su integración en un *hardware* de seguimiento y monitorización son las siguientes:

- Utiliza la red de datos *Simplex*.
- Antena *GPS* integrada.
- Comandos mejorados respecto al *STX3*.
- Modos de rastreo autónomos.
- Múltiples transmisiones que mejoran su fiabilidad.
- Cobertura global.
- Consumo de potencia reducido.
- Diseñado para una fácil integración en un módulo.
- Versátil para una gran variedad de aplicaciones.

El *STINGR* cuenta con un tamaño de 45,05 mm x 47,37 mm, y cuenta con un grosor de 6,91 mm incluyendo los componentes y la antena. En la siguiente imagen tenemos una vista del módulo:



**Fig. 2.9 STINGR (Globalstar)**

El módulo cuenta con la certificación modular *FCC (Federal Communications Commission) CFR (Code of Federal Regulations) parte 25* [11], referente a las comunicaciones vía satélite. Además, también posee la certificación *ISED (Innovation, Science and Economic Development)* por parte de la Industria de Canadá, la *ANATEL (Agencia Nacional de Telecomunicaciones)* y cuenta con el certificado *CE*.

Como ya se ha especificado, el *STINGR* utiliza la tecnología de la red de datos *Simplex* de *Globalstar*, con transmisión de datos en una única dirección.

En cuanto al funcionamiento, el *STINGR* tiene los mismos modos que el *hardware STX3*. Se tienen disponibles cuatro modos programables atendiendo al uso que se le quiera dar. A continuación, se detallan estos modos de operación del *STINGR*:

#### **MODO SLEEP (SUSPENSIÓN)**

Se aplica tensión a la unidad, pero no hay transmisiones pendientes ni actividad en serie.

#### **MODO ACTIVO**

El *STINGR* está activo y responde al puerto serie, pero no transmite.

#### **MODO STANDBY (ESPERA)**

El *STINGR* está inactivo entre ráfagas de transmisión, pero no está transmitiendo.

#### **MODO DE TRANSMISIÓN**

La unidad está transmitiendo un paquete de *RF*.

### **2.3.3 Iridium**

La compañía ofrece facilidades para el acceso a datos globales mediante la red *IoT* global que permite el acceso a datos en todas partes del mundo. *Iridium* ofrece la solución de *CloudConnect*, basada en una nube con cobertura global para las aplicaciones de *IoT* y que trabaja con *Amazon Web Services (AWS)*. Esta herramienta se ha diseñado para administrar dispositivos que trabajan como plataformas de comunicaciones, desde cualquier parte del mundo.

De entre los productos que ofrece *Iridium* para soluciones que implican tecnología vía satélite, se destacan a continuación algunos modelos, teniendo en cuenta la posibilidad de integrarlos al hardware del que disponemos.

### 2.3.3.1 *Iridium 9602*

El *Iridium 9602* [13] es una de las soluciones que ofrece la compañía para su aplicación en un *hardware* de manera que proporcione una solución para la aplicación de datos satelitales, haciéndolo una opción a valorar para el seguimiento y monitoreo de animales. Este dispositivo transmite datos a través del servicio *Iridium Short Burst*, que es una red eficaz que transmite mensajes de datos entre equipos.

El dispositivo cuenta con las siguientes características a tener en cuenta, y con las que podemos verificar que es una buena opción para aplicarlo al *hardware* que se desea diseñar:

- Transceptor de placa única, es decir, cuenta con transmisor y receptor en la misma placa.
- No necesita tarjeta *SIM* externa.
- Formato pequeño, para incorporarse en una solución *OEM*.
- *RoHS (Restriction of Hazardous Substances)*, directiva de la *UE* establecida para reducir el uso de sustancias peligrosas.

El *Iridium 9602* cuenta con un tamaño de 41 mm x 45 mm, tiene un grosor de 13 mm y su peso es de 30g. En la siguiente figura podemos ver el componente:



Fig. 2.10 *Iridium 9602* (*Iridium*)

### 2.3.3.2 *Iridium 9603*

El *Iridium 9603* [14] es la evolución del anterior *Iridium 9602*. Esta nueva versión cuenta con una cuarta parte del volumen y ocupa la mitad del espacio. Utiliza también la transmisión de datos mediante el servicio *Iridium Short Burst* como su antecesor.

El *hardware* cuenta con las mismas características que el anterior *Iridium 9602*, por lo que también se puede utilizar como una buena opción para aplicarlo al *hardware* de seguimiento. Las características son las siguientes:

- Transceptor de placa única, con transmisor y receptor en la misma placa.
- No necesita tarjeta *SIM* externa.
- Formato más pequeño, para incorporarse en una solución *OEM*.
- *RoHS (Restriction of Hazardous Substances)*, directiva de la *UE* establecida para reducir el uso de sustancias peligrosas.

El *Iridium 9603* cuenta con un tamaño de 31,5 mm x 29,6 mm, tiene un grosor de 8,1 mm y su peso es de 11,4g. En la siguiente figura podemos ver el componente:



**Fig. 2.11 Iridium 9603 (Iridium)**

### **2.3.3.3 Mini-PCle Adapter**

El *hardware 16130 Adaptador Mini-PCle* [15] es un módem que trabaja vía satélite proporcionando conectividad mediante el servicio *Iridium Short Burst*. Este dispositivo tiene integrado un transceptor *Iridium 9603N*, de manera que tiene comunicaciones bidireccionales en todo el planeta gracias a la constelación de satélites de la red *Iridium*, que está formada por un total de 66 satélites trabajando las 24 horas del día. Este módem requiere de un plan de datos mensual, que se puede establecer con proveedores externos.

El dispositivo cuenta con una serie de características que lo hacen idóneo para su uso tanto en seguimientos activos, telemetría, datos oceanográficos, monitoreo y aplicaciones de rastreo en general. Las características son las siguientes:

- Transceptor de satélite *Iridium 9603N SBD*, comentado en la sección anterior.
- Módulo pequeño y ligero para aplicaciones integradas.
- Ofrece comunicaciones por satélite de bajo costo y bajo ancho de banda.
- Carga útil máxima del mensaje: carga de 340 bytes, descarga de 270 bytes.
- Notificación automática de mensajes en mensajes entrantes.
- Opera globalmente, incluyendo los polos norte y sur.
- Admite mensajes de correo electrónico o *HTTP POST*.
- Interfaz de comando simple *UART AT*.
- Frecuencia de funcionamiento: 1616 MHz a 1626,5 MHz.
- Potencia de transmisión media de 1,6 W.

- -117dBm Sensibilidad del receptor
- Operación de temperatura industrial completa.
- No se requiere tarjeta *SIM*.
- Puerto de antena *u.FI* integrado para conexión de antena.
- Factor de forma mecánico *Mini-PCle* de tamaño completo.
- Utiliza el canal *Mini-PCle USB 2.0* para las comunicaciones.
- 1 año de garantía

El *16130 Adaptador Mini-PCle* cuenta con un tamaño de 30 mm x 50,95 mm, tiene un grosor de 11,6 mm y su peso es de 19g. En la siguiente figura podemos ver el componente:



**Fig. 2.12 16130 Adaptador Mini-PCle (Iridium)**

## 2.4 Empresas del sector

### 2.4.1 *Wildlife Computers*

Fundada en 1984, *Wildlife Computers* [16] es un proveedor de soluciones avanzadas de telemetría para la vida salvaje, con el objetivo de monitorizar en tiempo real a los animales portadores. Además, también ponen a la venta sistemas de geolocalización aparte de asesorar a los clientes para que estos elijan la opción que más les convenga.

La compañía utiliza distintas tecnologías en los *tags* que ofrece, dependiendo de la especie de animal a la que se quiera hacer un seguimiento, y esto se debe a que según el entorno en el que se encuentre el animal, se va a necesitar un tipo de *tag* u otro.

La empresa utiliza la telemetría por satélite de *Argos* para transmitir datos y rastrear movimientos de animales marinos que salen regularmente a la superficie del agua, por el hecho de que el *tag* debe de estar fuera del agua para poder transmitir datos con éxito a los satélites *Argos* que se encuentran en una órbita baja. El cálculo de la localización del animal se realiza mediante el efecto *Doppler* y el satélite retransmite el mensaje con los datos a estaciones receptoras terrestres.

Otras tecnologías que utiliza son la geolocalización por la luz que recibe el *tag*, también conocida como *GPE3*, que es útil para animales que pasan la mayor parte del tiempo en las profundidades del agua y con la que se pueden obtener observaciones del crepúsculo, temperatura y

profundidad. Por otra parte, también tenemos el *GPS Fastloc*, una tecnología que opera en entornos donde la señal del satélite solo está disponible durante cortos periodos de tiempo ya que destaca por ser capaz de trabajar muy rápidamente. Y, por último, también utilizan la tecnología *GPS*, que proporciona información sobre la localización y el momento en el que se ha localizado al animal a un receptor *GPS* continuamente.

La empresa también utiliza además algunas herramientas para la transmisión y la lectura de datos de los *tags* que llevan los animales. Mediante *Argos Pass Analysis*, se puede saber en qué momento exacto un satélite pasará por encima de un *tag*, una herramienta útil si lo lleva un animal con poco rango de movimiento por el hecho que permite el ahorro de batería del dispositivo al conectarse con el satélite tan solo cuando este está al alcance. Para el análisis y la representación gráfica de los datos se utiliza *IGOR Pro*, un software con el que se puede manipular, procesar y trazar los datos.

#### 2.4.2 *Sirtrack/Lotek*

*Sirtrack* es una compañía proveedora de productos de telemetría para la monitorización de aves, especies marinas y animales terrestres con sede en Nueva Zelanda. Esta fue adquirida por *Lotek* en 2011 [17], y en 2019 *Sirtrack* oficialmente pasó a llamarse *Lotek NZ Ltd*. Esta compañía, fundada en 1984, en 2007 adquiere también la empresa *Biotrack Ltd.*, otro proveedor de telemetría para la monitorización de animales con sede en el Reino Unido y que en 2019 pasó a llamarse *Lotek UK Ltd*.

Las tecnologías que ofrece se adaptan a las necesidades puesto que permiten rastrear animales de casi cualquier tamaño y entorno. Las opciones que ofrecen son la monitorización vía satélite, radio, acústica y de archivos.

Con el rastreo mediante satélite, se equipa a los animales con *tags* que hacen uso de tecnología *GPS*, *Iridium*, *Globalstar* o *Argos*. Estos *tags* determinan la ubicación precisa de un animal por medio de satélites *GPS* que transmiten los datos de posición de donde se encuentran geográficamente. Estos datos junto con los datos de otros sensores se envían a los satélites de comunicación que a su vez transmiten la información a estaciones terrestres.

Con la tecnología de radio, acústica y de archivos que utiliza la compañía, transmisor y receptor deben estar próximos para que las señales se puedan recibir. Mediante la tecnología radio, se monitorizan animales del medio terrestre y de agua dulce utilizando antenas para la comunicación con señales de radio entre transmisor y receptor. Por otra parte, tenemos la tecnología acústica, que se usa para el rastreo de especies que viven en aguas profundas y saladas por el hecho de que el sonido viaja cuatro veces más rápido en el agua que en el aire, por lo que es método muy útil en este entorno. Para este método se utilizan transmisores y receptores que trabajan con un intermediario, un hidrófono que recibe las señales acústicas del emisor y las almacena o las convierte en señales de radio para su transmisión. Y por último tenemos la tecnología de archivo, que trabaja almacenando la información y se usa generalmente en especies marinas. Esta última tecnología tiene el inconveniente de que los *tags* necesitan ser recuperados para poder obtener la información que almacenan.

## Capítulo 3. Diseño e implementación del proyecto

En este apartado se va a desarrollar el montaje y programación en conjunto de los distintos componentes *hardware* que se han elegido para el proyecto. En un primer lugar, creando un entorno para comprobar el funcionamiento de todos los dispositivos en consonancia, para a continuación aplicar las normas o restricciones que se requieran para el objetivo final del proyecto, que es el seguimiento de tortugas.

Para ello, en primer lugar, se expondrá la propuesta que se ha planteado, explicando los distintos puntos a tener en cuenta para el diseño del proyecto final y una solución que cumpla con el objetivo final de este. A continuación, se definirán los sensores que van a ser necesarios y que se han elegido para la realización de la idea, examinando sus características principales y su conexión con la placa de microcontrolador *Arduino UNO*. Para finalizar, se explicarán las herramientas de programación que se han utilizado para la implementación del código que permite que el módulo funcione siguiendo la propuesta explicada y el montaje de todos los integrados utilizados en un solo *PCB* el cuál se va a diseñar para este proyecto en concreto.

### 3.1 Necesidades del proyecto

Para poder definir una propuesta que ofrezca una solución al proyecto, en un primer lugar será necesario definir las necesidades de este, para de esta manera tener en cuenta cuáles serán los sensores que necesitaremos y tener una idea de los métodos de programación que se deberán utilizar para cumplir con las necesidades exigidas. Además, también hay que tener en cuenta los distintos factores que puedan llegar a suponer un problema en un futuro, cuando queramos poner a prueba la solución final, y de esta manera, anticiparnos y proponer una solución que se adapte a dichos problemas.

Empezando por las necesidades del proyecto, y teniendo en cuenta que la finalidad de este es el seguimiento de tortugas, podemos asegurar que va a ser necesario un dispositivo de seguimiento que nos permita conocer la localización más exacta posible del quelonio, y para ello se ha decidido integrar un dispositivo *GPS* que nos proporcionará las coordenadas de dónde se encuentre el animal. Por otro lado, necesitaremos un dispositivo con comunicación satelital con el que enviar los datos que recojan los distintos sensores, este será de extrema importancia por el hecho de que va a ser el encargado de que se pueda saber dónde se encuentra la tortuga y los datos de todos los sensores que integremos en la solución final, dado que una vez que se le aplique el dispositivo al animal ya no podremos volver a recuperarlo hasta que se dé la oportunidad en un tiempo indefinido, o incluso no tengamos la oportunidad de recuperarlo. Además, las tortugas al ser especies marinas hay que tener en cuenta que van a sumergirse debajo del agua durante algún tiempo, y es por esto que hay que utilizar algún dispositivo con el que sea posible conocer en todo momento si el animal se encuentra en la superficie del agua o por el contrario se encuentra sumergido, dado que dependiendo de esta situación se tendrá comunicación con los satélites o no, ya sea para que el *GPS* reciba las coordenadas o para poder enviar los datos que se recojan de los sensores al satélite mediante el módulo transmisor elegido.



Otro sensor que también será necesario será un dispositivo capaz de mantener informado al módulo de la hora y fecha en todo momento, y para ello se utiliza un *RTC* que controlará todas las operaciones que requieran precisión temporal. Por último, y para tener información adicional, se añade al dispositivo de seguimiento un sensor de temperatura con el que contaremos con información de la temperatura del agua en la que se mueven las tortugas marcadas. Un factor muy importante de los sensores que se decidan utilizar en el proyecto va a ser su consumo, debido a que hay que tener en cuenta que el dispositivo de seguimiento final va a ir acoplado a una tortuga con una batería recargable con paneles solares para que cada vez que el animal suba a la superficie se recargue y no deje de funcionar el dispositivo. Al no saber exactamente cuánto tiempo va a pasar la tortuga en la superficie y cuánto va a pasar debajo del agua, lo más sensato es conseguir que el consumo de los componentes sea el mínimo para que la vida de la batería que se incorpore sea lo más larga posible. Cabe decir, que se va a tener en cuenta en todo momento la posibilidad de añadir más sensores al módulo, con la finalidad de aportar información adicional que sea útil para el seguimiento de las tortugas y/o para supervisar el correcto funcionamiento del dispositivo de seguimiento.

Una vez se han aclarado y definido las necesidades del proyecto, el siguiente paso es adquirir y unir los distintos sensores para que funcionen conjuntamente orquestados por la placa de microcontrolador *Arduino UNO*. La universidad tenía a disposición una placa *Arduino UNO*, un sensor de localización *GPS GY-GPS6MV2* y un higrómetro *HD-38*, por lo que para el proyecto se utilizan estos dispositivos en un primer momento a la espera de que funcionen correctamente y cumplan con la función deseada. Para los sensores restantes, se realizará una búsqueda y una posterior comparación entre distintos modelos y marcas para obtener el que más se ajuste tanto en funcionamiento como en presupuesto.

Para el dispositivo encargado de la comunicación satelital, y una vez decidido que la tecnología de telemetría que vamos a utilizar es la de *Argos*, respaldada por *CLS*, hay dos opciones en el mercado. La primera es el *KIM1*, con tecnología *Argos-2*, una placa diseñada expresamente para su rápida integración en aplicaciones de seguimiento de animales con *Arduino UNO* y con un coste significativamente asequible. Y por otro lado tenemos el chipset *ARTIC-R2*, con tecnología *Argos-2/3/4*, pero sin ninguna placa sobre la que trabajar, por lo que habría que invertir tiempo y dinero para su diseño por lo que deja de ser asequible para el presupuesto dado, además de que las tecnologías que usa no estarán disponibles hasta 2 años en adelante. Por lo que la elección definitiva ha sido el módulo *KIM1*.

En cuanto al *RTC*, comparando distintos modelos de diferentes fabricantes según su consumo, precisión y otras características como podemos ver en la tabla que se muestra a continuación, y teniendo también en cuenta que el precio se ajusta al presupuesto, se ha elegido el *RTC RV-3028-C7* por su bajo consumo, su precisión y por ofrecer unas mejores prestaciones que los otros sensores comparados:

	Consumo	Rango de Voltaje	Precisión	Bus	Dimensiones
RV-3028-C7	45nA typ. 60nA max.	1.2 ~ 5.5V	±1ppm	I <sup>2</sup> C	3,2 x 1,5 x 0,8 mm
RV-3032-C7	160nA typ. 210nA max.	1.2 ~ 5.5V	±1.5ppm	I <sup>2</sup> C	3,2 x 1,5 x 0,8 mm
MAX31341B	180nA typ. 330nA max.	1.6 ~ 3.6V	-	I <sup>2</sup> C	2 x 1,5 x 0,5 mm
MAX31342	150nA typ. 330nA max.	1.6 ~ 3.6V	-	I <sup>2</sup> C	1 x 2 x 0,5 mm
DS1347	350nA typ. 700nA max.	2 ~ 5.5V	-	SPI	3.1 x 3.1 x 0.8 mm
DS1343	250nA typ. 500nA max.	1.7 ~ 5.5V	-	SPI/3-wire	3.1 x 3.1 x 0.8 mm

**Tabla 3.1 Comparación RTCs**

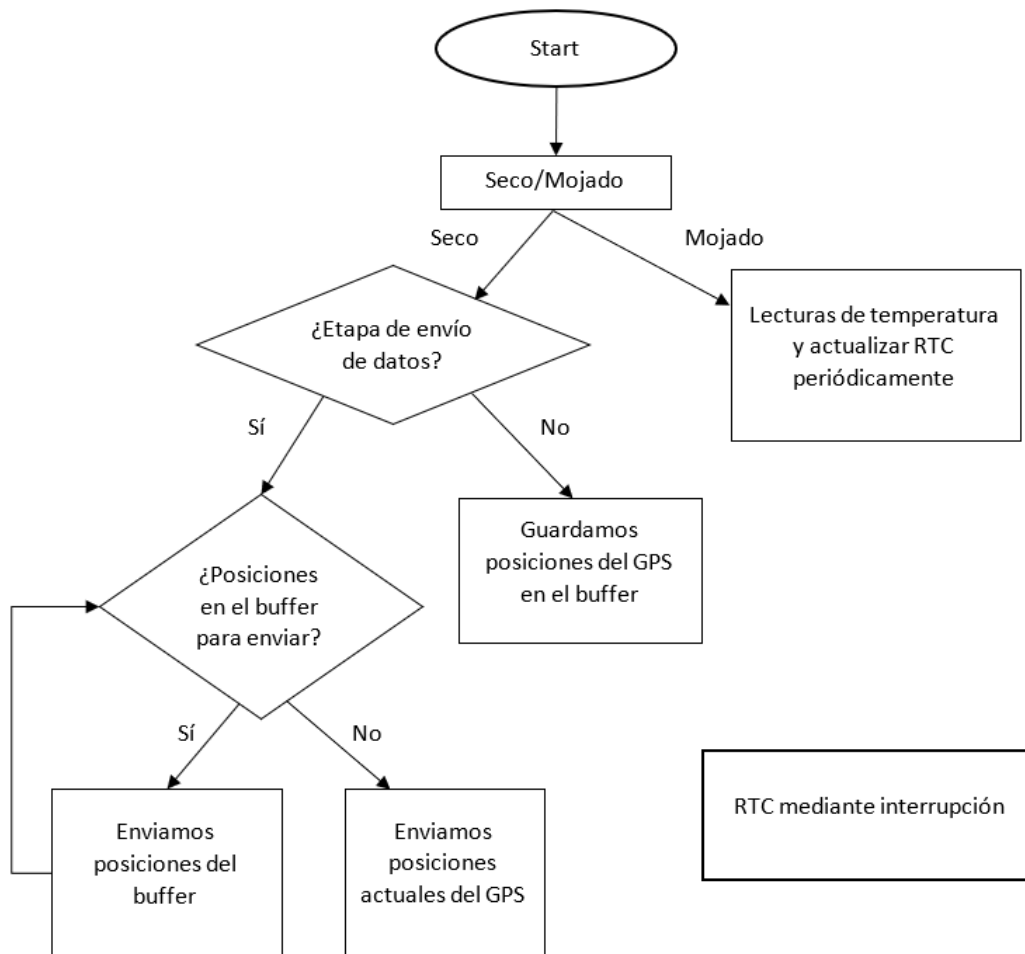
Lo mismo para el sensor de temperatura, haciendo una comparación de distintos modelos según el consumo del dispositivo, la precisión, el rango de medida y otras características que podemos observar en la tabla que se presenta más abajo, y contando con que el precio sea asequible y se ajuste al presupuesto, se ha elegido el sensor de temperatura *MCP9808*, que ofrece una buena precisión, un consumo muy bueno en comparación con los otros dispositivos que se muestran, además de un rango de medida que cubre ampliamente su propósito:

	Consumo	Rango de Voltaje	Precisión	Bus	Rango de medida	Dimensiones
STS30-DIS	45µA typ. 600µA max.	2.15 ~ 5.5V	±0.2 °C	I <sup>2</sup> C	0 °C ~ 65 °C	2.5 x 2.5 x 0.9 mm
STS31A-DIS	45µA typ. 600µA max.	2.4 ~ 5.5V	±0.3 °C	I <sup>2</sup> C	-40 °C ~ 90 °C	2.5 x 2.5 x 0.9 mm
MAX31889	0.55µA typ. 68µA max.	1.7 ~ 3.6V	±0.25 °C	I <sup>2</sup> C	-20 °C ~ 105 °C	2 x 2 x 0.8 mm
MAX30208	0.5µA typ. 67µA max.	1.7 ~ 3.6V	±0.15 °C	I <sup>2</sup> C	0 °C ~ 70 °C	2 x 2 x 0.75 mm
	120µA max.					
MCP9808	0.1µA typ. 200µA max.	2.7 ~ 5.5V	±0.25 °C	I <sup>2</sup> C	-40 °C ~ 125 °C	-

**Tabla 3.2 Comparación sensores temperatura**

### 3.2 Propuesta

Con los sensores que vamos a utilizar decididos, y teniendo en cuenta las dificultades que se nos presentan al definir la solución al proyecto, pasamos a desarrollar el planteamiento que hay que llevar a cabo para el correcto funcionamiento del dispositivo de seguimiento plenamente funcional. Para ello, vamos a guiarnos del siguiente diagrama de flujo en el que se resumen las tareas que se realizarán:



Como podemos comprobar en el diagrama de flujo, en un primer momento el dispositivo de seguimiento comprobará mediante el higrómetro si la tortuga se encuentra en la superficie del agua o por el contrario está sumergida. En el caso de que se encuentre en la superficie, será cuando el *tag* aprovechará que puede comunicarse con los satélites por el hecho de que las señales electromagnéticas usadas por el *GPS* y el dispositivo de comunicación satelital se debilitan notablemente debajo del agua, no llegando a penetrar a profundidades superiores a 50 cm. Es por esto que es cuando se realizará la comunicación *GPS* con el objetivo de obtener las coordenadas donde está posicionado el animal, y a continuación, se ejecutará la lectura de los sensores restantes y por último se enviará la información al satélite mediante el *KIM1*, información que el satélite transmitirá y se podrá visualizar en la web de *Argos* con todos los

datos de las lecturas de los sensores. En el caso contrario, al estar el dispositivo por debajo de la superficie nos interesará ahorrar batería ya que los paneles solares que darán energía a la batería estarán inutilizados al no llegarles energía solar, por lo que si no mantenemos un bajo consumo del dispositivo y la tortuga tarda un largo periodo de tiempo en volver a subir, o se mantiene durante poco tiempo en la superficie podría significar que el módulo se quede sin batería y deje de funcionar. Además, cabe señalar como se ha comentado anteriormente que el módulo de seguimiento no tendrá señal de los satélites y no se podrá comunicar con estos, por lo que no tiene sentido intentar una comunicación que se sabe que no va a ser funcional y que va a ocasionar un alto consumo de energía para el *tag*. También se va a limitar el envío de mensajes a los satélites dependiendo si la tortuga se encuentra o no en la franja horaria idónea para el envío de datos, y de esto hablaremos más adelante en el apartado 3.5 Implementación del código del porqué.

### 3.3 Sensores del integrado

#### 3.3.1 *KIMI*

El módulo transmisor *KIMI* de *Argos* viene integrado en una placa diseñada específicamente para facilitar su integración en la plataforma de desarrollo de código abierto *Arduino*, más específicamente a la placa *Arduino UNO* de esta compañía. A esta placa se le denomina *KIMI Shield*. Este dispositivo cuenta con la tecnología *Argos-2*, que tiene comunicación satelital unidireccional.

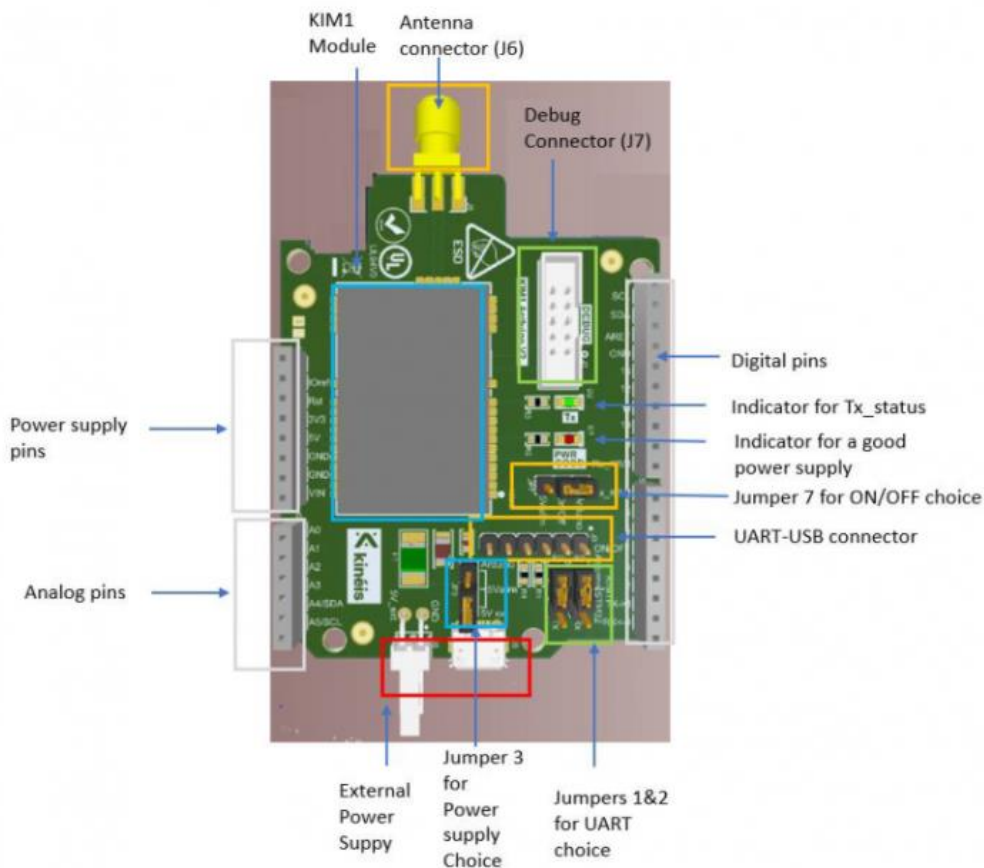


Fig. 3.1 KIMI Shield

El rango de voltaje de alimentación del *KIM1 Shield* es de entre 4.7V a 5.3V, y se configurará de tal manera que la placa *Arduino UNO* lo alimente con los 5V que esta puede proporcionar. La potencia de transmisión del *KIM1* es configurable, siendo las potencias recomendadas 250mW, 500mW y 1000mW, y según la potencia elegida para la transmisión tendremos un consumo de corriente u otro cuando se realiza la transmisión, siendo de 480mA si la potencia es de 250mW, 650mA para 500mW y de 965mA para los 1000mW. Lo mismo ocurre con la corriente consumida cuando el *KIM1* está en modo *sleep*, depende de la potencia de transmisión a la que esté configurado el *KIM1*, variando en unos pocos miliamperios, entre 3mA y 5mA.

Al usar *Arduino* para el desarrollo de las funciones que deberá realizar el módulo *KIM1*, en la placa de este deberemos colocar *jumpers* para configurarlo de tal manera que responda a los comandos que el *Arduino* le trasmita. Tenemos cuatro distintos *jumpers* que habrá que colocar en la placa, en primer lugar, tenemos los *jumpers* de transmisión y recepción y que están definidos en la *Fig. 3.1 KIM1 Shield* como *jumpers 1* y *2*. Seguidamente, el *jumper* de alimentación también habrá que colocarlo para que sea la placa *Arduino* la que alimente el módulo con los 5V que necesita para su funcionamiento. Y, por último, el *jumper* de encendido/apagado, funciones que serán controlados por el *Arduino* también según se le ordene desde el código que se implemente.

### 3.3.2 Módulos GPS GY-GPS6MV2 y L86-M33

Para conocer la ubicación exacta del dispositivo de seguimiento utilizaremos la tecnología de localización *GPS*, y se ha elegido el módulo *GY-GPS6MV2*. Este encapsula un transceptor *GPS NEO-6* de la compañía *u-blox* e incluye una memoria *EPROM* que se utiliza para guardar la configuración una vez el sensor se apaga, una batería de *back-up* y una antena cerámica. Se conecta en serie con el *Arduino* y precisa de una alimentación de 3 a 5 V con una velocidad de comunicación de 9600 bps.

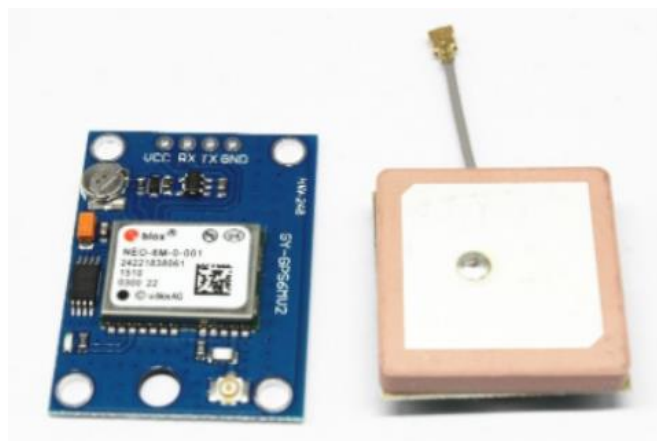
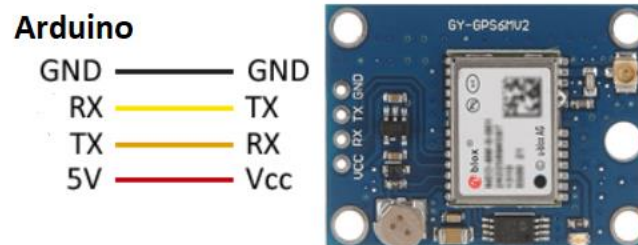


Fig. 3.2 GPS GY-GPS6MV2

La conexión del módulo *GPS* y el *Arduino* se realiza mediante comunicación *UART* disponible en ambos dispositivos, conectando los pines de transmisión y recepción del *GY-GPS6MV2* a los pines digitales que configuremos en el código para la comunicación serie. Y la alimentación del

GPS se lleva a cabo conectando  $V_{cc}$  y GND a la alimentación y masa del *Arduino* tal y como se puede observar en la siguiente figura:



**Fig. 3.3 Conexión GY-GPS6MV2**

Esta familia de receptores *GPS* están especialmente diseñados para tener un consumo y un tamaño reducidos, además de que su coste es accesible para todo el mundo, tanto individuales como empresas. Hablando de consumo, el módulo que se está presentando requiere de una intensidad de corriente de unos 37mA en cuando está en el modo de medición continuo. En cuanto a la precisión del módulo *GPS*, para la posición cuenta con una precisión de 2.5 m, para la velocidad de 0.1 m/s y para la orientación de 0.5°, por lo que se puede considerar un sensor de posicionamiento bastante fiable y con el que podremos obtener los resultados que deseamos para el proyecto.

Por otro lado, para el diseño final de *tag*, se ha elegido el *GPS L86-M33* de la compañía *Quectel*. Esto es porque este *GPS* ofrece unas características y utilidades mejores a las del *GPS GY-GPS6MV2*, además de que lleva la antena directamente integrada encima del componente, evitando de esta manera que en el *tag* tengamos que hacer una pista específicamente para la antena reduciendo así el espacio que necesitamos para este integrado, y evitando también tener que adquirir una antena por separado del componente.



**Fig. 3.4 GPS L86-M33**

Este componente pide una alimentación den entre 3V y 4,3V siendo 3,3V el voltaje típico con una velocidad de comunicación de entre 4800 bps y 11520 bps. El consumo máximo del *L86-M33* es de 30mA en adquisición, mientras que en el modo de *Standby* tiene un consumo de 1mA y en modo *Backup* el consumo es de 7 $\mu$ A.

Por otro lado, el *L86-M33* cuenta con algunas aplicaciones que pueden ser muy útiles a la hora de conseguir que el *GPS* sea lo más eficiente posible en el *tag*. Debido a que las tortugas son animales que pasan la mayor parte de su tiempo debajo del agua y se va a disponer de escasos momentos donde se pueda realizar una comunicación con los satélites hay que aprovechar esos momentos muy bien. Este *GPS* permite ser programado para que, realizando un cálculo de predicción de las órbitas de los satélites, se consiga reducir de manera considerable los tiempos de adquisición. El *L86-M33*, también permite guardar posiciones en su *flash* interna con un periodo de 15 segundos, con una capacidad de hasta 16 horas de datos los cuales se pueden recuperar posteriormente para su uso. Además, posee una aplicación que lo protege de otras señales de radiofrecuencia que puedan interferir cancelándolas.

### 3.3.3 *RTC RV-3028-C7*

Para llevar un control del tiempo preciso en todo momento se ha elegido el módulo *RV-3028-C7*, de la empresa *Micro Crystal*, un *RTC* que lleva incorporado un oscilador de cristal para poder crear señales eléctricas con frecuencias muy precisas. Además, de entre algunas de las opciones en su librería de *Arduino* está la de programar alarmas y timers con interrupciones.



Fig. 3.5 *RTC RV-3028-C7*

Este módulo está optimizado de manera que su consumo de energía es notablemente bajo, de unos 45nA al estar alimentado a 3V y el oscilador de cristal cuenta con una precisión de  $\pm 1$  ppm. Tiene una memoria interna *EEPROM* en la que guarda los ajustes de configuración, con espacio adicional para uso del usuario y ofrece una amplia variedad de funciones con las que trabajar como contadores programables, alarmas, funciones de interrupción y además también un contador de tiempo *UNIX* de 32 bits.

La comunicación de este módulo *RTC* es *I2C* a una velocidad de 400 kHz, y la conexión de los pines *SCL* (*System Clock*) y *SDA* (*System Data*) del *RV-3028-C7* se realiza a los pines analógicos *A5* (*SCL*) y *A4* (*SDA*) de la placa *Arduino*.

### 3.3.4 *Sensor de temperatura MCP9808*

Para medir la temperatura del entorno en el que el animal se mueve, se ha elegido el sensor de temperatura digital *MCP9808* de la empresa de *hardware Microchip*, y llevado a una *PCB* por la empresa *Adafruit*.



**Fig. 3.6** Sensor de temperatura *MCP9808*

Este sensor convierte temperaturas dentro de un rango de entre  $-40^{\circ}\text{C}$  y  $+125^{\circ}\text{C}$  a un dato digital con el que podemos trabajar, con una precisión de  $\pm 0.5^{\circ}\text{C}$  máximo. También dispone de un amplio rango de voltaje, entre 2.7 V y 5.5 V, además de que cuenta con un consumo de corriente muy bajo para su funcionamiento de  $200\mu\text{A}$ , y tiene opción de poder programarlo para que el sensor esté en modo *sleep* mientras no se esté midiendo la temperatura con este, con un consumo de  $0.1\mu\text{A}$ . Por otro lado, el sensor nos permite medir la temperatura con diferentes precisiones decimales, según nuestras necesidades, con la posibilidad de calcular hasta una precisión de diezmilésimas.

Al igual que el anterior sensor, el *RTC RV-3028-C7*, el *MCP9808* posee un interfaz *I2C* que transmite a una velocidad de 400kHz. Por lo tanto, para utilizar este bus de comunicaciones conectaremos los pines *SCL* (*System Clock*) y *SDA* (*System Data*) del sensor de temperatura *MCP9808* a los pines analógicos *A5* (*SCL*) y *A4* (*SDA*) de la placa *Arduino*.

### 3.3.5 *Higrómetro HD-38*

El sensor utilizado para saber si el animal se encuentra dentro o fuera del agua mediante dos electrodos resistivos es el higrómetro *HD-38*. Midiendo la resistencia entre los electrodos que dependerá de la humedad del ambiente, sabremos que nos encontramos en contacto con el agua si la resistencia es muy baja, o por lo contrario en la superficie si esta resistencia es muy alta.





**Fig. 3.7 Higrómetro HD-38**

En la *Fig. 3.7 Higrómetro HD-38* vemos que el electrodo resistivo está conectado a un *PCB* en la que se encuentra el sensor *HD-38* capaz de entregar tanto una salida digital (pin *D0*) como analógica (pin *A0*), y dependiendo de la salida que utilicemos la conectaremos a un pin digital o analógico del *Arduino*. La salida digital actúa como comparador, activándose cuando el ambiente llega a un nivel de humedad menor al que nosotros le marquemos, y podemos controlar este nivel con el potenciómetro que tiene acoplado el *PCB*. Por otro lado, la salida analógica funciona como la salida de un divisor de tensión entre una resistencia fija y la resistencia situada entre los electrodos, dando como resultado un voltaje entre 0V y 5V para un suelo húmedo y seco consecutivamente.

El sensor trabaja con un rango de voltaje de alimentación de entre 3.3V a 12V, y su consumo de corriente es de unos 30mA aproximadamente.

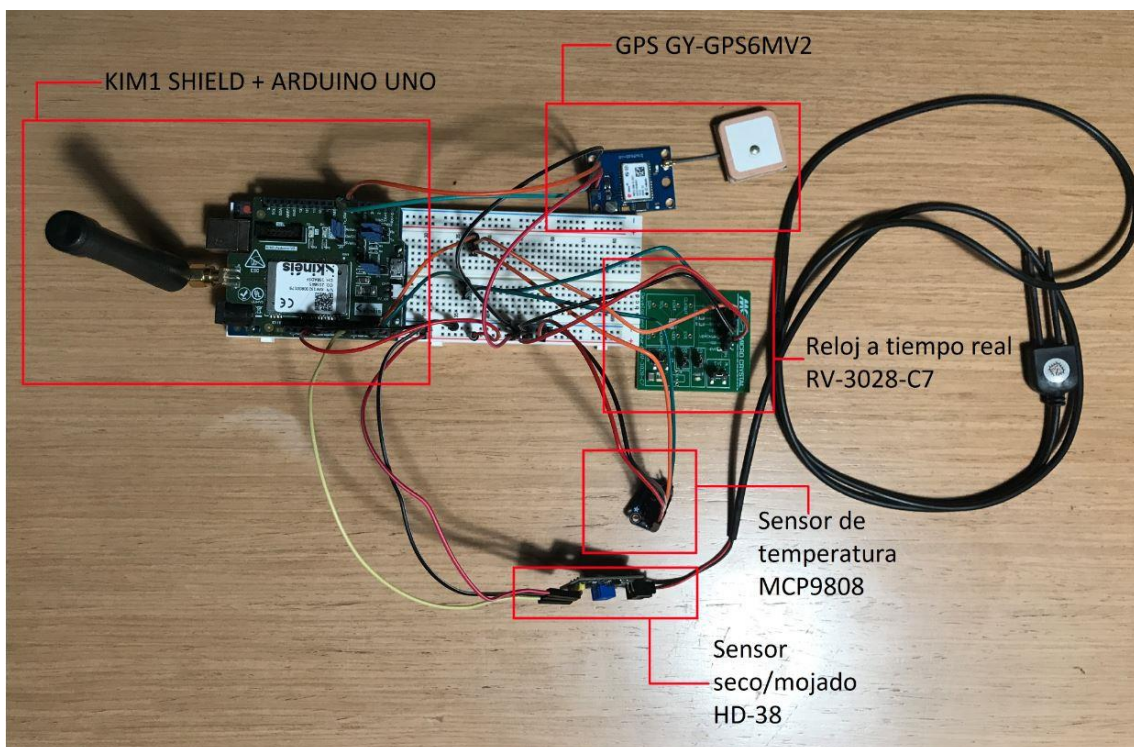
### **3.4 Montaje del módulo de seguimiento completo**

Una vez explicada la conexión de cada sensor con la placa de microcontrolador *Arduino UNO* por separado, para el montaje del módulo completo necesitaremos una *protoboard* que facilite la integración de los componentes conjuntamente y tener presente los puertos de la placa *Arduino UNO* que van a ser utilizados por los sensores.

Teniendo en cuenta esto, los puertos que vamos a utilizar en la conexión de los sensores utilizados en el proyecto son, en primer lugar, los pines de alimentación de 5V, 3V3 y *GND*, que alimentarán los sensores anexos al *Arduino UNO*, por lo que conectaremos un cable a cada uno de estos pines y los llevaremos a la *protoboard* para poder alimentar cada uno de los sensores con la tensión que requieran. Por otro lado, de los pines analógicos *A4* y *A5* del *Arduino* obtendremos los buses de comunicaciones *I2C SDA (System Data)* y *SCL (System Clock)* respectivamente, que van a ser utilizados por el sensor de temperatura *MCP9808* y el *RTC RV-3028-C7*. Además, tenemos el protocolo de comunicación de dispositivos en serie *UART*

(*Universal Asynchronous Receiver-Transmitter*), que ocupará los pines digitales del *Arduino* que programemos al implementar el código, y que para el caso de este proyecto son el 5 y el 6 para el sensor *GPS GY-GPS6MV2* y los pines 7 y 8 para el *KIM1 Shield*. Otro pin que hay que tener en cuenta es el que se va a utilizar para la salida analógica que se precisa para el cálculo de la humedad del higrómetro *HD-38*, el pin analógico *A0*, que se conecta directamente a la salida analógica del sensor de humedad. Por último, para la alimentación del *Arduino UNO* se utiliza el conector *USB* que lo alimenta con 5V, y se puede alimentar con un ordenador, con una batería portátil o con pilas recargables.

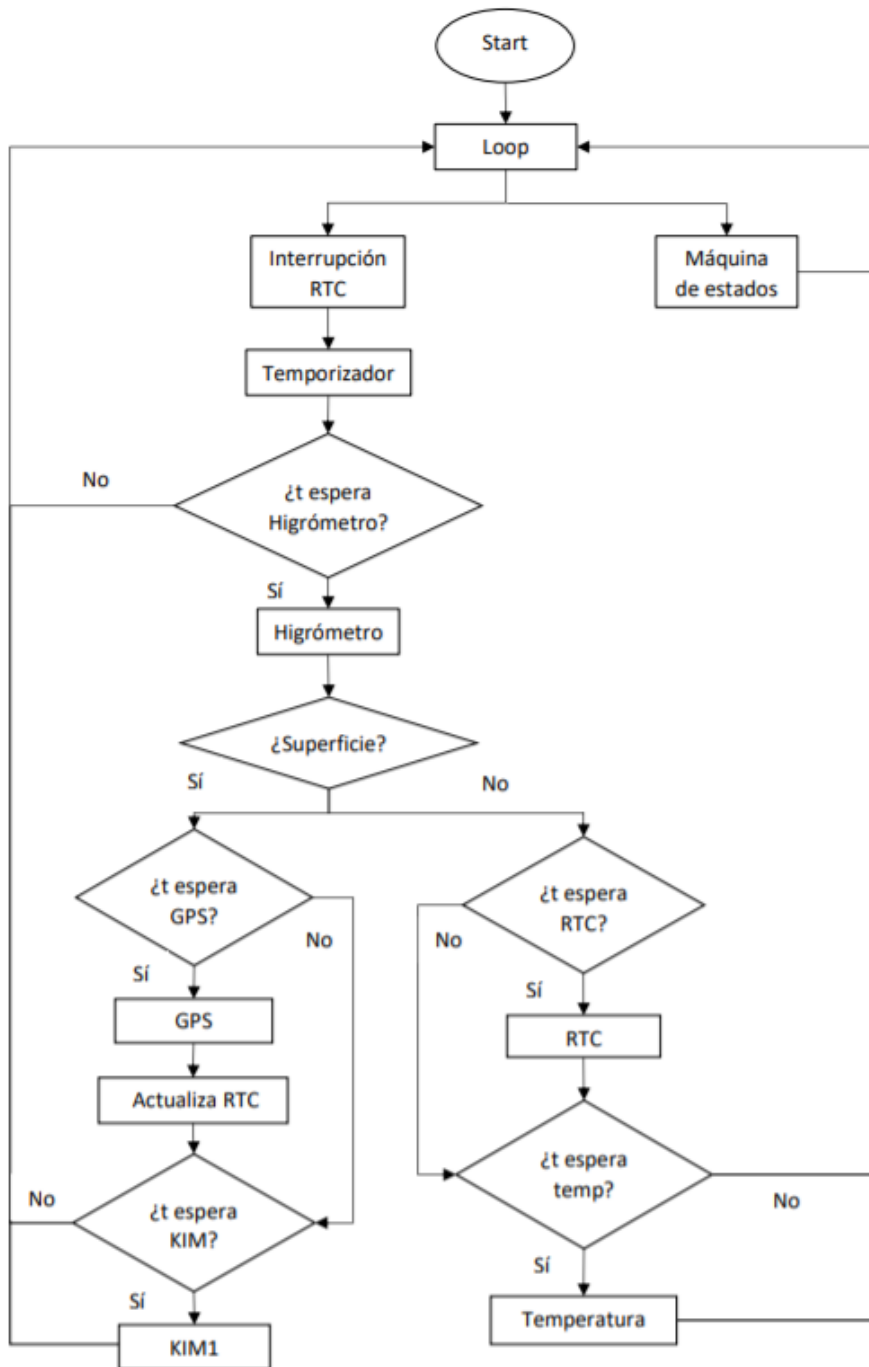
Con todo esto comentado anteriormente, en la siguiente figura podemos ver el montaje de todos los sensores utilizados conectados al *Arduino UNO*, con su respectivo cableado y conectados mediante la *protoboard*:



**Fig. 3.8** Dispositivo de seguimiento completo

### 3.5 Implementación del código

En este apartado se va a comentar los diferentes métodos utilizados en el código que se ejecuta para que el sensor de seguimiento funcione según la propuesta que se ha planteado. Para la implementación del código, se va a seguir el diagrama de bloques que se muestra a continuación en el que podemos ver un esquema detallado del programa que va a hacer que el dispositivo de seguimiento funcione correctamente:



Una vez visto el esquema que se va a seguir para el funcionamiento del código, se va a explicar más detenidamente los diferentes métodos que se utilizan para el funcionamiento del módulo de seguimiento. En primer lugar, después de haber incluido todas las librerías de las que vamos a hacer uso para los sensores y para la comunicación entre dispositivos, y a continuación de definir las variables de los datos que se manejarán en el código, se va a hacer uso de los métodos de programación *union* y *struct* para la codificación de los datos, con los que se concatenarán varias variables de diferentes tipos y tamaños en una única matriz para poder enviarla usando el *KIM1*, que manda los datos al satélite mediante una matriz. En las líneas de código de la siguiente figura podemos ver la implementación de las estructuras:

```

/*****
*****Data Encoding*****
*****/

union
{
    uint8_t data_sensors[sizeofdata];

    struct
    {
        byte month;
        byte day;
        int year;
        byte hour;
        byte minute;
        byte second;
        float latitude;
        float longitude;
        int humidity;
        int temp;
        char sindatos[sizeofdata-sizeofsensors];
    }sensor;

}data;

/*****/

```

**Fig. 3.9 Codificación de datos**

En la *Fig. 3.9 Codificación de datos* vemos que se hace uso de un *union* que incluye dentro un método *struct*. La función del *struct* es la de definir un conjunto de variables, que en el caso del código del proyecto que nos ocupa son los distintos datos que obtenemos de cada sensor y que tienen distintos tipos y tamaños, en una única estructura. El *struct* del código es denominado como *sensor*, y contendrá el conjunto de datos de la misma manera que cualquier otro tipo de variable, pero de manera compuesta. Por otra parte, la estructura *union* nombrada como *data*, contiene el *struct* que se ha definido anteriormente y una matriz a parte definida como *data\_sensors*, con lo que tendremos una única región de memoria en la que se almacenarán todos los datos cuando se lean de los sensores tanto en el *struct sensor* como en la matriz *data\_sensors*, permitiéndonos hacer uso de las dos opciones y podremos utilizar la matriz para enviar los datos de los sensores al satélite mediante el *KIM1*.

A estar trabajando con *Arduino* y con su entorno de programación, la primera función que se ejecutará será la de *setup*, y tan solo se ejecutará una única vez. Será en este método donde vamos a inicializar las comunicaciones seriales con las que trabajamos, que son la del *GPS* y la del módulo satelital *KIM1*. Además, en esta función se hace una llamada a otra función a la que

se le da el nombre de *start* y que va a ser la encargada de inicializar los sensores que precisen de un encendido programado. Cabe destacar, que en la función *start* se va a inicializar un *timer* controlado por el *RTC*, que será el encargado de controlar la frecuencia con la que los sensores harán sus medidas mediante una interrupción cada vez que se acabe el tiempo del *timer* y un contador que se utilizará para llevar la cuenta de los segundos que han pasado, en la siguiente figura tenemos la inicialización del *RTC* y del *timer* dentro de la función *start*:

```

/*****
*****RTC initialization*****
*****/

if (rtc.begin() == false) {
  Serial.println("Something went wrong, check wiring");
  while (1);
}
else{
  Serial.println("RTC ON");
}

/*****TIMER*****/
rtc.setTimeout(true,1,2,true,true,false);

// Timer parameters
// 1   Repeat
// 2   Frequency
// 3   Timer value in seconds
// 4   Interrupt
// 5   Start
// 6   clock output
*****/

```

Fig. 3.10 Inicialización del RTC

Una vez se han puesto en funcionamiento los sensores y el *timer* que vamos a utilizar, el programa entra en la función *loop*, la cual se va a estar repitiendo de forma indefinida hasta que el microcontrolador se apague o se reinicie. En el *loop* del código que se ha implementado, tenemos llamadas a dos funciones, la primera denominada *rtc\_trigger* y la segunda *state*, lo podemos comprobar en la siguiente figura:

```

void loop() {

  /*****
  *****Loop*****
  *****/

  rtc_trigger();
  state();

}

```

Fig. 3.11 Función bucle

La función *rtc\_trigger* es la función encargada de comprobar si se debe hacer la lectura de datos de alguno de los sensores según el intervalo de tiempo que hayamos programado para dicha acción y es en esta función donde dependiendo de si el módulo de seguimiento se encuentra en

la superficie o por debajo del agua, se activarán unos u otros sensores para su posterior lectura. Para el control temporal de la lectura de sensores y como se ha comentado anteriormente, se va a hacer uso de un *timer* del *RTC*, por lo que nada más se entre en la función de control temporal hay una llamada a la función *timerCounter*, y podemos ver su funcionamiento en la siguiente figura:

```

void timerCounter(){

    /*******
    *****Counter for execute*****
    *****sensors*****
    *****/

    if(counter >= 50000){

        counter = 0;
        previous_t_soil = 0;
        previous_t_rtc = 0;
        previous_t_temp = 0;
        previous_t_gps = 0;
        previous_t_RIM = 0;

    }else if (rtc.readTimerInterruptFlag()) {
        rtc.clearTimerInterruptFlag();
        counter++;
    }

}

```

**Fig. 3.12 Interrupción por temporizador**

En un primer lugar, en la función *timerCounter* se hace un control de desbordamiento de la variable *counter*, para que en ningún momento llegue al máximo y deje de hacer su función, dejando entonces de funcionar el código completo por el hecho de que no se harían lecturas de los sensores. A continuación, se hace una lectura de la interrupción del *timer*, y en caso de que el tiempo del temporizador haya pasado, se entrará en esta condición en la que se limpia la interrupción para poder volver a activarla cuando el temporizador vuelva a activarse y a continuación se aumenta el valor de la variable *counter*, con la que vamos a controlar el tiempo que ha pasado para de esta manera poder controlar la lectura de los sensores.

Dentro del bucle del microcontrolador tenemos también la función *state*, la cuál es una máquina de estados con las funciones de lectura de cada sensor. Según el estado, controlado por la variable *currentState*, que irá cambiando su valor según la función *rtc\_trigger* de paso a la lectura de los sensores o al acabar la lectura de algún sensor, entraremos en la función del sensor asociado a ese estado.

```

void state() {

    /*******
    *****State machine*****
    *****/

    switch(currentState)
    {
        case gpsSearch:
            gpsFunction();
            break;
        case gpsEncode:
            gpsEncodeData();
            break;
        case rtcSat:
            rtcSatellite();
            break;
        case temperature:
            temperatureFunction();
            break;
        case hygrometer:
            hygrometerFunction();
            break;
        case kimData:
            kimFunction();
            break;
        case kimSendData:
            kimSend();
            break;
        default:
            break;
    }
}

```

**Fig. 3.13 Máquina de estados**

Para el envío de datos mediante el *KIM1* al satélite, se deben tener los datos almacenados en un array de caracteres hexadecimales, por lo que los datos de los sensores que se han ido almacenando gracias el método *union* en un array del tipo *uint8\_t*, se convertirán mediante el bucle de la siguiente figura en la cadena hexadecimal necesaria para su envío, que va a ser almacenada en una variable denominada *data\_satellite*.



```

void kimFunction() {

    serialgps.flush();
    serialgps.end();
    kserial.listen();

    /*****Data for Satellite*****/
    /*****Binary uint8_t[] to hex-string (char[]) convert function.*****/

    for(uint8_t i=0; i<31; i++){
        sprintf((char *) (data_satellite + i * 2), "%02X", data.data_sensors[i]);
    }
    /*****/

    currentState = kimSendData;
}

```

Fig. 3.14 Función KIM1

Por último, como se ha planteado en el apartado 3.2 en el que se plantea la propuesta de funcionamiento del *tag*, se han limitado las horas a las que el módulo *KIM1* va a enviar mensajes a una franja horaria en la que los satélites con los que se comunica sean más visibles, de esta manera evitamos un consumo excesivo de energía en periodos de tiempo en los que es muy probable que el *KIM1* no se comunique con ningún satélite al hacer el envío de datos ahorrando batería, lo que es un factor muy importante a tener en cuenta para el *tag*. En la siguiente figura podemos ver la franja horaria del paso de satélites de *Argos*:

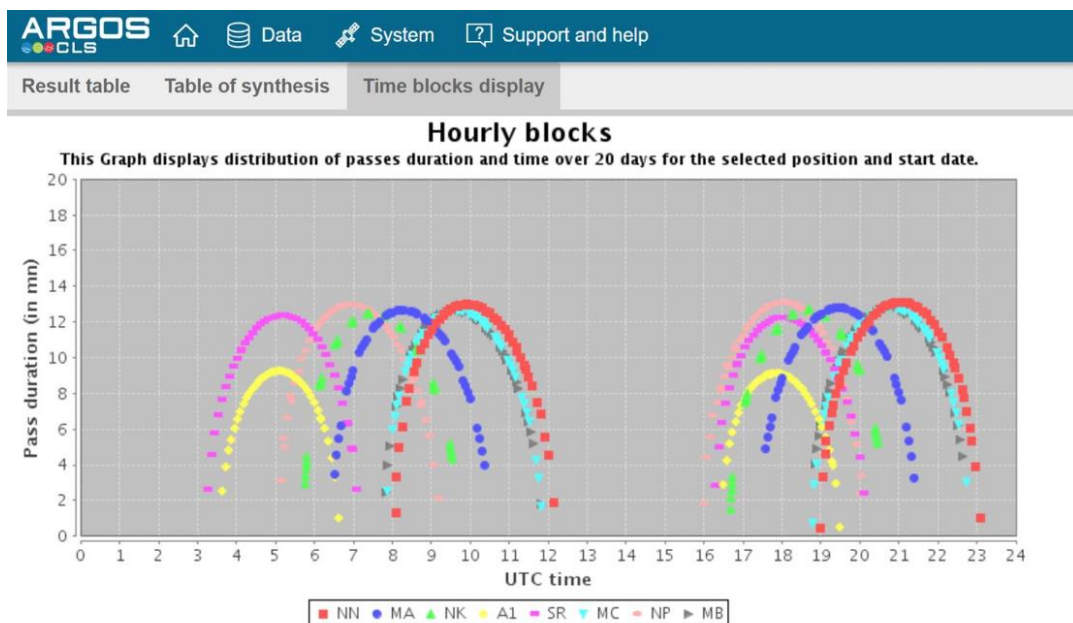


Fig. 3.15 Franja horaria de paso de satélites

Debido a esto, los datos se van a enviar durante un periodo de tiempo limitado al día, y para no perder los datos del resto de día, se ha implementado un *buffer* circular que va a ir guardando



las posiciones que reciba el *GPS* en las franjas horarias en las que el *KIM1* no realiza envíos. Una vez se entre en el periodo de tiempo en el que el *KIM1* envía datos, la primera acción que se realizará será hacer una lectura del *buffer* para comprobar si hay posiciones guardadas que no han sido enviadas, y estas serán las primeras en enviarse a la vez que se irán guardando las posiciones actuales en el *buffer*. Cabe decir que, al ser un *buffer* circular, una vez este se llene con datos, se volverá a la primera posición, y volverá a guardar posiciones eliminando las que se hayan guardado primero.

### 3.6 Diseño de módulo de seguimiento

En este apartado, se va a diseñar el esquemático y el *PCB* del módulo de seguimiento que se utilizará para el seguimiento de las tortugas una vez se adhiera al caparazón de estas. Para ello, al haber probado el correcto funcionamiento de los componentes que se han elegido para el proyecto, se procede a ensamblar estos en una misma placa, con el fin de compactar y de hacer operativo el módulo para el seguimiento de tortugas. Para ello se utiliza el programa de *software* libre para el diseño electrónico *KiCad*, con el que se diseñará en primer lugar un esquemático de módulo, diferenciando cada sensor con los componentes que necesite para su funcionamiento. A continuación, se procederá al diseño del *PCB*, distribuyendo entonces cada componente de la manera más eficaz posible para que la placa no ocupe demasiado y que asegure el correcto funcionamiento de todos los componentes.

#### 3.6.1 Esquemático

En primer lugar, tenemos la representación esquemática del módulo *KIM1* [18]. A este componente se le añaden algunos elementos para su correcto funcionamiento. En la Fig. 3.16 *Esquemático KIM1* tenemos la representación esquemática con las conexiones necesarias para el funcionamiento del componente:

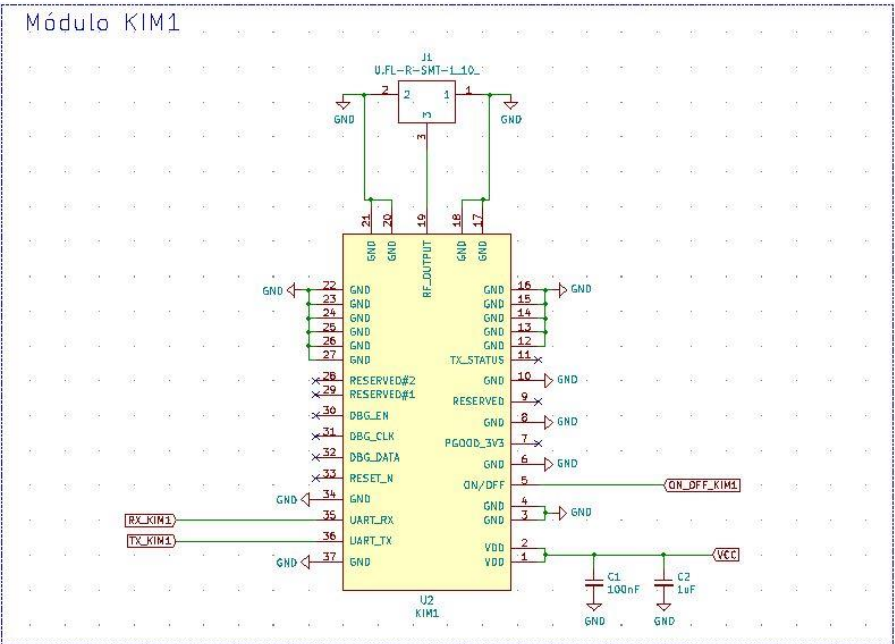
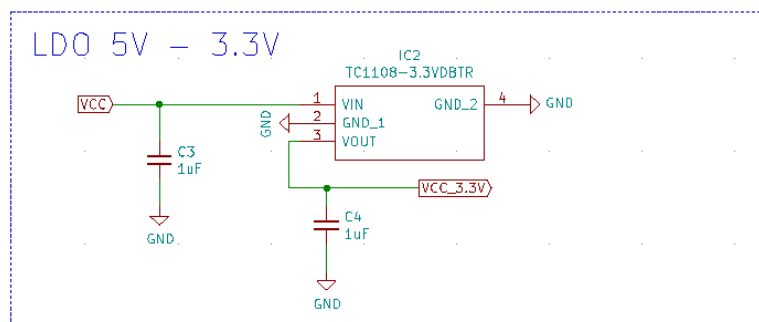


Fig. 3.16 Esquemático KIM1

Por un lado, tenemos los dos condensadores de desacoplo que se van a encargar de eliminar el ruido de la señal de corriente continua que alimenta al *KIM1*, conduciendo la corriente alterna a masa, y haciendo de esta manera que la corriente continua sea lo más limpia posible. Por otro lado, tenemos la antena que se encargará de la comunicación satelital del módulo *KIM1* con los satélites a los que se les enviará los datos recogidos por los distintos sensores. También se han realizado las conexiones necesarias con el microcontrolador para poder hacer uso de las funciones del *KIM1* necesarias para este proyecto, que son el control de encendido y de apagado del módulo, muy importante para poder tener controlado el consumo del *tag* apagando el módulo satelital y tan solo encendiéndolo cuando este vaya a ser utilizado, y la transmisión y recepción de datos satelitales para poder enviar estos datos a los satélites *Argos* cuando esté programado y con toda la información que le haya proporcionado el microcontrolador con los datos de los otros sensores.

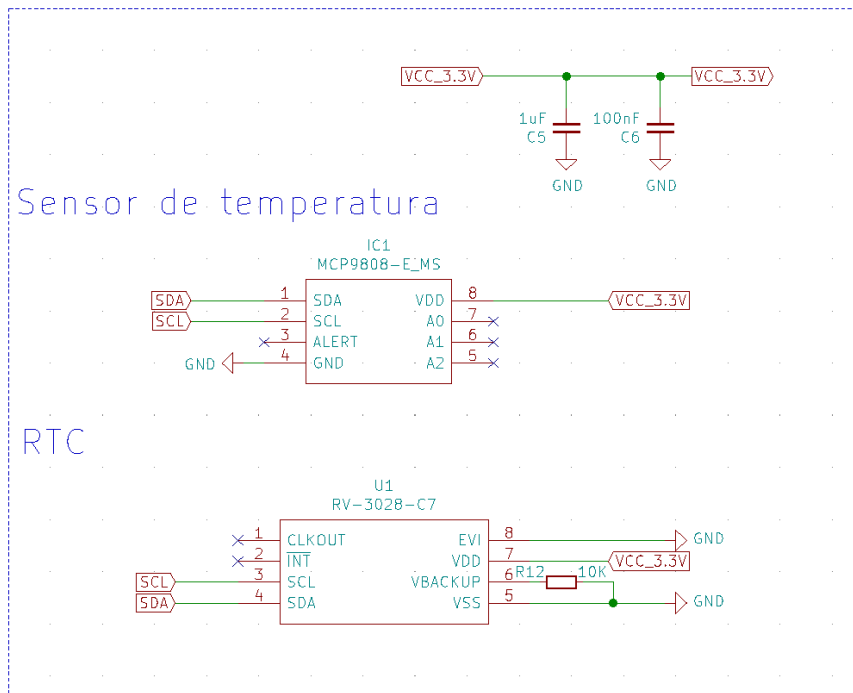
La alimentación de esta primera versión del *tag* será suministrada por una batería portable que entrega un voltaje de 5V, y algunos de los componentes que se van a utilizar tan solo soportan un voltaje de 3.3V, es por esto que se va a hacer uso del regulador de voltaje *TC1108* [19] que se encargará de alimentar los sensores que precisen menos voltaje.



**Fig. 3.17 Esquemático LDO**

En la *Fig. 3.17 Esquemático LDO* tenemos la representación del esquemático del regulador en cuestión, al que se le han añadido tanto en la entrada como en la salida los condensadores de desacoplo que la hoja de datos del componente indica que se deben utilizar para el correcto funcionamiento de este.

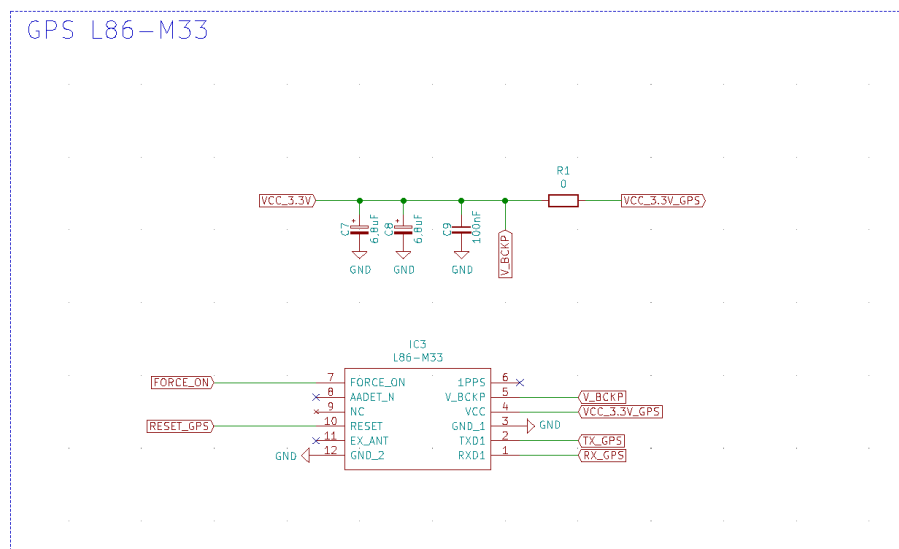
Los siguientes componentes que tenemos en el esquemático son el sensor de temperatura *MCP9808* [20] y el reloj a tiempo real *RV-3028-C7* [21], para el montaje de los cuáles se ha hecho una revisión de las hojas de datos de cada uno de ellos para una correcta aplicación de estos componentes. Al ser sensores de un tamaño muy reducido y ambos tener comunicación *I2C* con el microcontrolador, vamos a tener en cuenta que en el diseño del *PCB* van a ser colocados muy próximo el uno del otro, y es por esto que se van a utilizar los mismos condensadores de desacoplo para ambos sensores como podemos ver en la figura a continuación, donde se representa el esquemático de estos componentes:



**Fig. 3.18** Esquemático sensor de temperatura y RTC

En el esquemático de la figura anterior *Fig. 3.18 Esquemático sensor de temperatura y RTC* se puede ver como se han hecho las conexiones necesarias para la comunicación I2C con el microprocesador, mediante los pines *Serial Clock Line (SCL)* y *Serial Data Line (SDA)* así como la conexión de la tensión de respaldo del RTC a tierra mediante una resistencia de 10kΩ para la protección de este pin, según se menciona en la hoja de datos del componente.

A continuación, el siguiente componente del que vamos a hacer un análisis desde el punto de vista esquemático es el *GPS L86-M33* [22], y que está representado esquemáticamente como podemos observar a continuación, incluyendo las conexiones y componentes externos necesarios para su funcionamiento:



**Fig. 3.19** Esquemático GPS L86 M-33

En la Fig. 3.19 Esquemático GPS L86 M-33 vemos que, para el funcionamiento del GPS, se han añadido los condensadores de desacoplo encargados de eliminar el ruido en la alimentación del componente, además de una resistencia de  $0\Omega$  que hará de puente en la alimentación del dispositivo, y que si se retira en algún momento se tendrá la posibilidad de alimentar el GPS mediante la alimentación de respaldo  $V\_BCKP$ . Por otro lado, tenemos la conexión de transmisión y recepción de datos de las efemérides con el microcontrolador, así como algunas de las funciones que ofrece el componente, y que se ha considerado necesario su uso para el correcto funcionamiento del GPS. Estas funciones son, la función  $FORCE\_ON$ , que se encarga de encender el dispositivo si en algún momento este se usa en el modo *backup*, en el que tendrá un consumo muy bajo, del orden de  $7\mu A$ , debido a que dejará de adquirir efemérides y de rastrear satélites. Y por otro lado la función de  $RESET$ , función importante a tener en cuenta por el hecho de que el componente puede dejar de funcionar correctamente y consecuentemente se dejarán de recibir datos, por lo que hacer uso de esta función para poder hacer un *reset* del dispositivo periódicamente y de esta manera prevenir fallos es importante.

Otro componente del *tag* es el higrómetro, que funcionará mediante el comparador  $LMV331$  [23] y que comparará dos señales de tensión de entrada dando lugar a una señal de salida digital que determinará cuál de las dos señales es mayor. El transistor que hay en el  $LMV331$  se encenderá o apagará dependiendo de cuál de las señales sea la mayor, produciendo de esta manera una señal de salida alta o baja. Si la tensión que llega a la entrada no inversora es mayor el transistor está apagado por lo que la salida es alta, mientras que si la tensión en la entrada inversora es mayor el transistor está encendido y la salida es baja. A continuación, tenemos una figura de la representación esquemática del comparador completo:

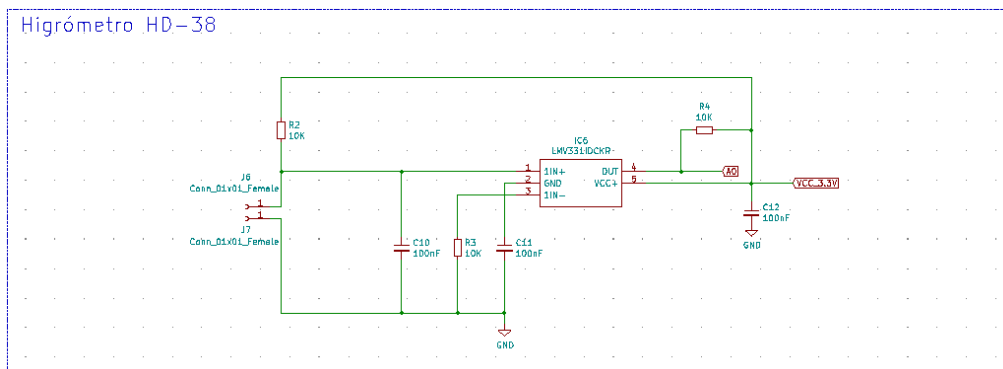


Fig. 3.20 Esquemático higrómetro

El  $LMV331$  tiene una etapa de salida de drenaje abierto,  $A0$  en este caso, por lo que se requiere una resistencia de *pull-up* a una tensión de alimentación positiva para que la salida conmute correctamente.

Por último, tenemos el esquemático del microcontrolador y otros componentes que requiere este para su funcionamiento. Se ha utilizado el  $ATMEGA328P$  [24] para el control del *tag*, y en la siguiente figura se muestra el esquemático con todas las conexiones que tiene con los sensores y componentes del *tag* para que todo funcione conjuntamente:

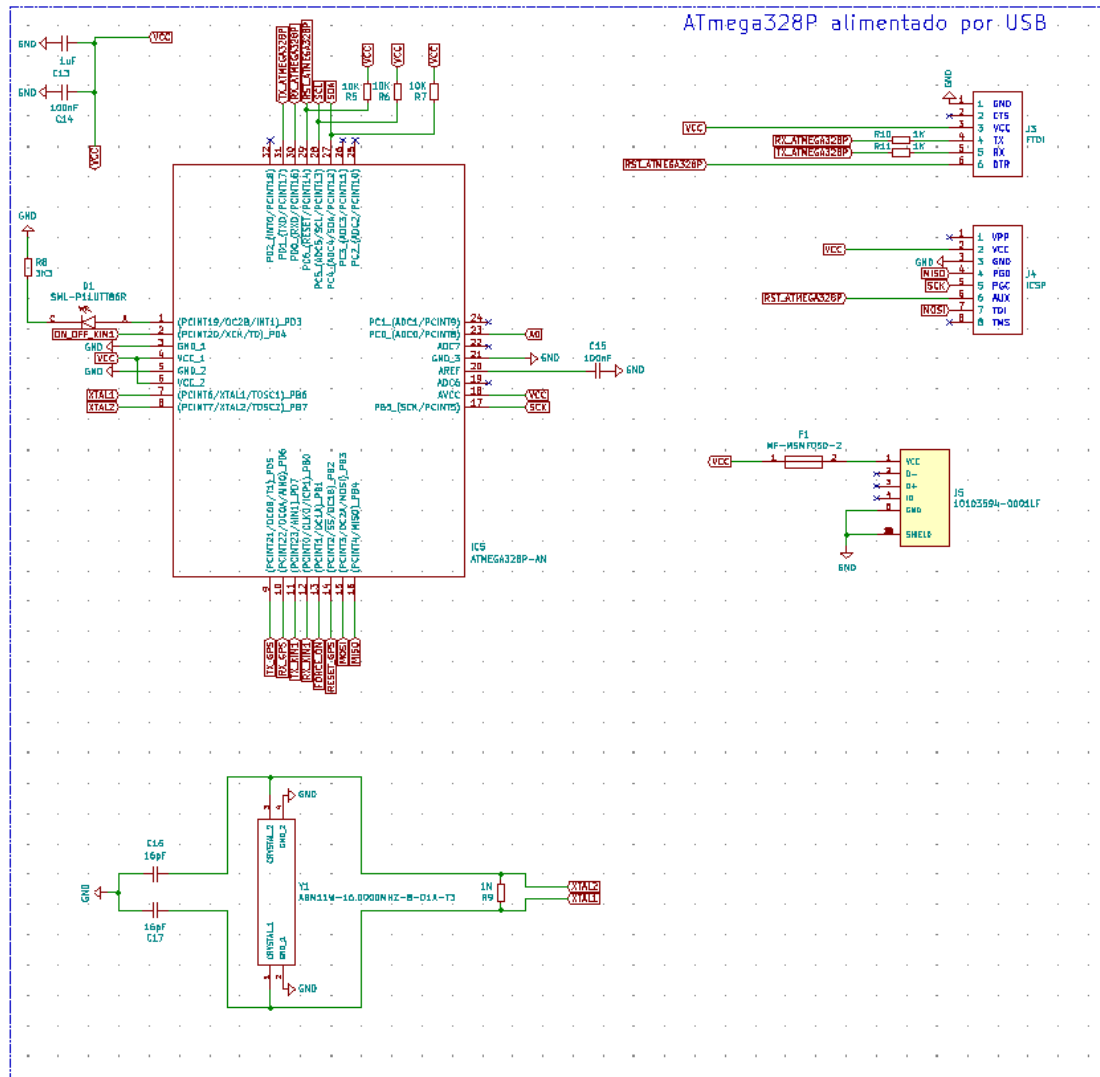


Fig. 3.21 Esquemático microcontrolador ATMEGA328P

En la Fig. 3.21 Esquemático microcontrolador ATMEGA328P se pueden distinguir los condensadores de desacoplo, un LED que se utiliza para indicar distintos estados del tag, así como también todas las conexiones entre los distintos componentes del tag con las entradas y salidas digitales. A parte del microcontrolador ATMEGA328P, hay algunas partes más que tienen su función en el tag en este esquemático, por ejemplo, tenemos el circuito del oscilador que se encarga de indicar a qué velocidad o frecuencia va a trabajar el microcontrolador generando señales cuadradas de una específica frecuencia que serán las que se utilicen para sincronizar todas las operaciones del sistema. Por otra parte, tenemos el que va a ser el puerto de alimentación del tag, donde se conectará la batería portable que suministrará voltaje al dispositivo y que se trata de un puerto USB Micro tipo B. También tenemos otros dos puertos, uno es el puerto ICSP (In Circuit Serial Programming) mediante el que se va a cargar el bootloader en el microprocesador para que en este se pueda ejecutar el software, con el ATMEGA328P ya soldado en el tag sin la necesidad de que se tenga que programar el chip antes de su montaje en la placa y que se cargará con la herramienta de programación y depuración PICKIT 4 [25]. El otro puerto es el FTDI (Future Technology Devices International), con el que se utiliza la placa FTDI modelo FT232RL [26] para convertir señales de USB a serial, cargando de manera muy sencilla de esta manera el sketch de Arduino que se ha implementado en el microprocesador

*ATMEGA32P* instalado en la placa, dándonos además la posibilidad de poder editar y volver a subir este *sketch* tantas veces como sea necesario.

También existe la posibilidad de utilizar la placa *Arduino UNO* tanto para la puesta a punto quemando el *bootloader* como para programar el microcontrolador *ATMEGA-328P*, usando las conexiones *ICSP* y *FTDI* que se han preparado para estos propósitos. Para cargar el *bootloader* en el microcontrolador, será tan sencillo como utilizar el *Arduino* como *ISP (In System Programming)* y mediante las conexiones *ICSP* al *ATMEGA-328P* utilizar la herramienta para quemar *bootloader* que ofrece la *IDE (Integrated Development Environment)* de *Arduino*. Una vez al microcontrolador del *PCB* se le ha cargado el *bootloader*, si retiramos el microcontrolador de la placa *Arduino* y la conectamos mediante las conexiones *FTDI* al *ATMEGA328-P* de la *PCB*, podremos subir programas al nuevo microcontrolador de la misma manera que se suben a la placa *Arduino UNO*, conectando esta placa al puerto *USB* de un ordenador y subiendo el *sketch* desde el *IDE* de *Arduino*.

Con todas estas partes mencionadas anteriormente se crea el esquemático completo del *tag*, y con todas las conexiones y los componentes necesarios, pasamos a diseñar la placa donde se incorporarán y distribuirán de la manera más eficaz posible.

### 3.6.2 *Diseño PCB*

En este apartado, aparte de mostrar el resultado final del *PCB*, se van a comentar los aspectos más importantes que se han tenido en cuenta con respecto al diseño de este como por ejemplo tamaño de las pistas y de las vías, prevención de interferencias electromagnéticas y reglas de diseño de algunos componentes. El *PCB* que se ha diseñado consta de dos capas, y se ha tomado esta decisión en un principio por varios motivos que se van a mencionar a continuación. En primer lugar, un factor por el que se ha decidido que el *PCB* se diseñe de dos capas es por tiempo y coste, por el hecho de que un *PCB* de más capas hubiera sido considerablemente más costoso de diseñar tanto temporal como monetariamente hablando, y el tiempo del que se disponía para realizar una primera versión del *tag* para el proyecto europeo *LIFE INTEMARES* era limitado. Además, al ser una primera versión del *tag*, interesa realizar pruebas de que todo funciona correctamente de manera conjunta con un diseño más sencillo para de esta manera cuando se compruebe la funcionalidad de los componentes, realizar un diseño más complejo con la certeza de que el diseño funcionará y que se han obtenido buenos resultados.

En las siguientes figuras vemos el diseño que se ha realizado con la herramienta de *software* libre *KiCad*, tanto por la capa frontal del *PCB* como por la capa trasera:

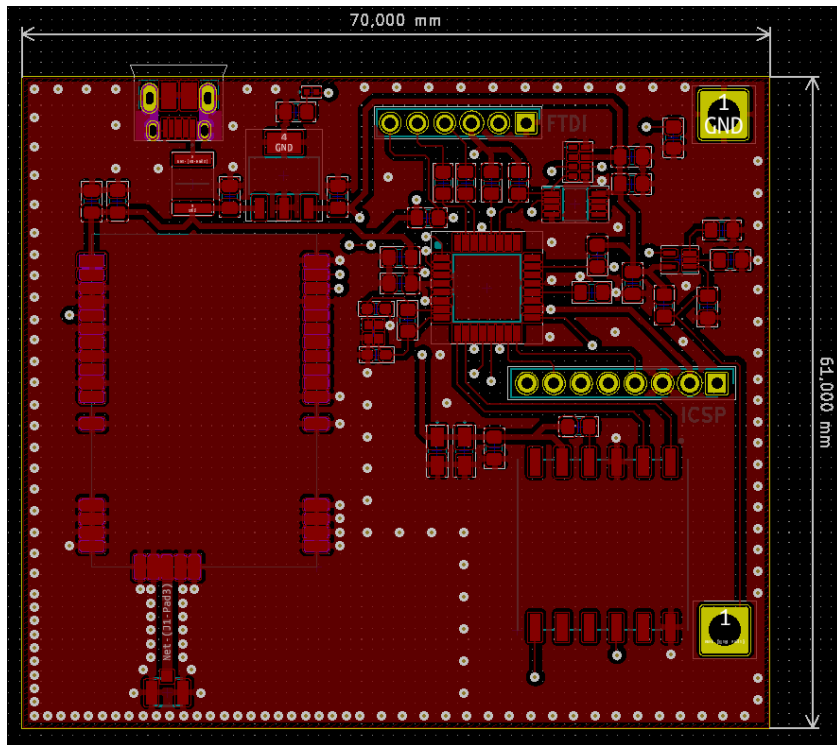


Fig. 3.22 Vista frontal del PCB

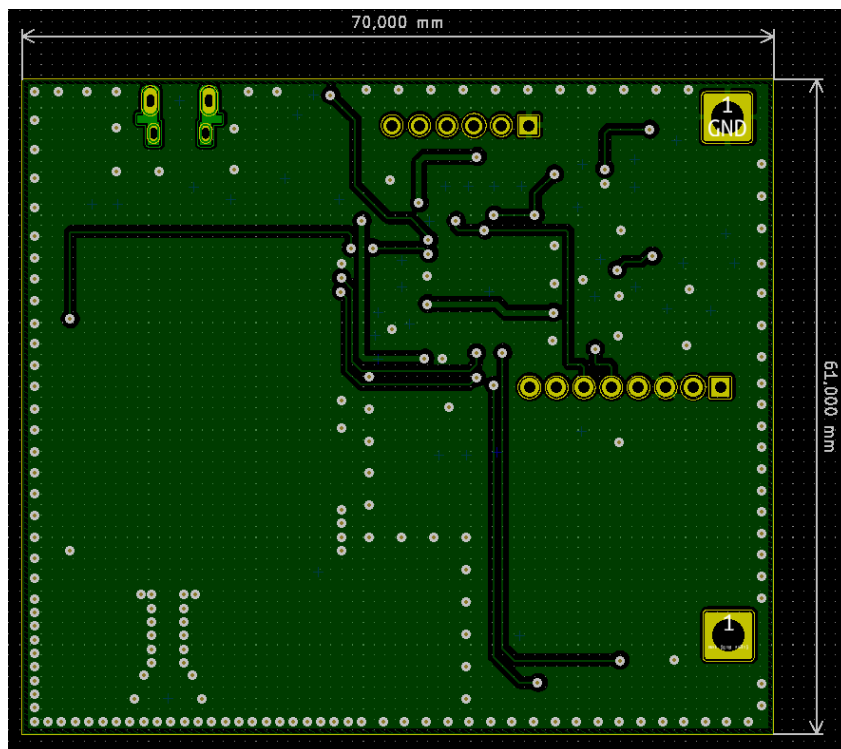


Fig. 3.23 Vista trasera del PCB

Comentando las figuras Fig. 3.22 Vista frontal del PCB y Fig. 3.23 Vista trasera del PCB, se aprecia que el PCB se ha cosido por los laterales y entre algunos componentes creando de esta manera una jaula de Faraday que los protege de campos eléctricos estáticos y de señales no deseadas.

En cuanto a la colocación de los componentes, los dos más críticos del *PCB* son los que envían o reciben señales, y en el caso del *tag* que se ha diseñado son el *KIM1* y el *GPS*, por lo que tienen algunas reglas de diseño que se han seguido a la hora de colocarlos en la placa. Estas reglas están presentes en las hojas de datos de los componentes y se centran, como se ha mencionado, en la colocación de estos en la placa, así como la separación que hay que dejar con otros integrados que puedan crear interferencias de señal. Además, también hay que tener en cuenta las antenas de estos componentes, que para el caso del *GPS* la lleva adherida en la parte superior del integrado por lo que lo único que hay que tener en cuenta es que se respete la separación con otros componentes de radio frecuencia, mientras que el *KIM1* tiene su antena separada por una pista de la cual se ha medido la impedancia como se puede ver en la siguiente figura:

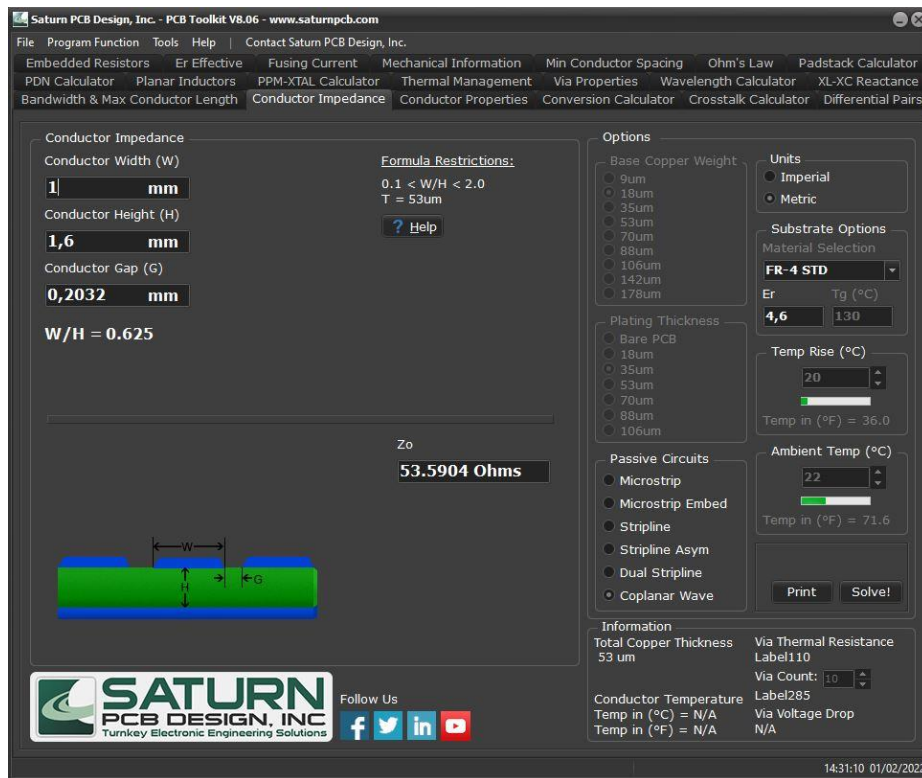


Fig. 3.24 Cálculo de la impedancia de pista del *KIM1*

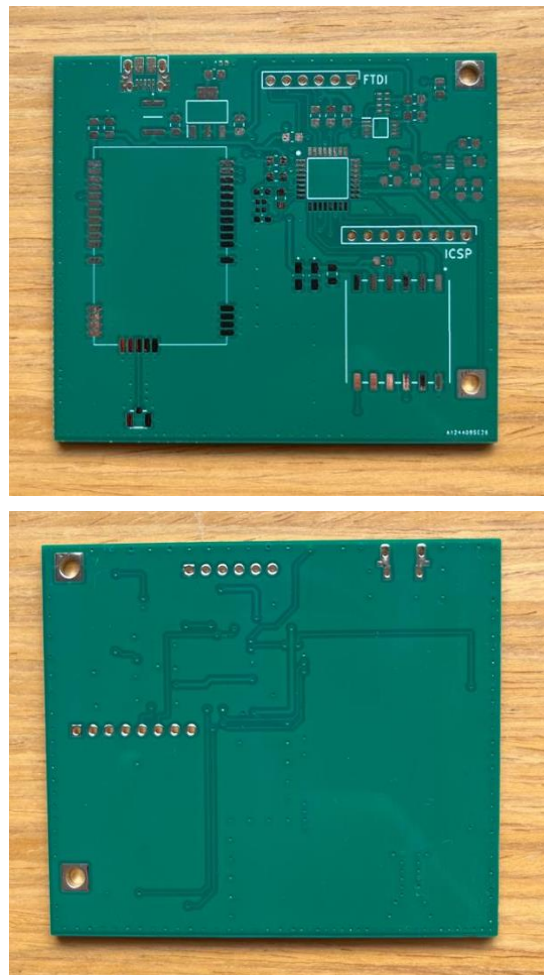
Para este cálculo, se ha utilizado el *software PCB Toolkit* de *Saturn*, y añadiendo los datos de los que disponemos se hace un cálculo de la impedancia de la pista. En este caso, con la frecuencia de trabajo del *KIM1*, que es de 401,650 MHz, y sabiendo que la impedancia ( $Z_0$ ) que tiene que tener la pista es de 50Ω, que la anchura del *PCB* es de 1,6mm y que el hueco del conductor es de 0,2032mm, vamos probando que anchura de pista vamos a necesitar para que la impedancia que resulte sea de 50Ω, que en este caso es de 1mm.

En cuanto al resto de pistas, van a ser de alimentación o de señal. Para las pistas de alimentación vamos a tener un diseño más ancho de 0,5mm, ya que por estas va a pasar más cantidad de corriente, mientras que para las pistas de señal se requiere una menor anchura y lo correcto es que sean pistas muy finas del orden de 0,2mm ya que por estas van a pasar señales lógicas con una corriente muy baja y suelen ir dirigidas a pines de alta impedancia.



También podemos ver como el *PCB* tiene una cantidad considerable vías entre componentes que van al plano de masa, de esta manera se reduce el camino de retorno de las señales que seguirán el camino más corto asegurando de esta manera una buena integridad de señal. El plano de masa cubre toda el área del *PCB* tanto en la capa frontal como en la capa trasera creando de esta manera un potencial de referencia común para todos los componentes, además con el plano cubriendo el *PCB* entero se reduce el ruido eléctrico y las interferencias *EMI* debido a los bucles de tierra que son causados por corrientes no deseadas entre dos puntos debido a una diferencia de potencial.

Con el *PCB* diseñado, el siguiente paso es obtener los archivos *Gerber* que se enviarán para que se fabrique el *PCB*. Estos archivos son necesarios para de esta manera seguir una directiva estándar para la producción de circuitos impresos, creando así una compatibilidad universal entre diseñadores de *PCB* y fabricantes, estableciendo de esta manera la norma para la producción de circuitos impresos. El *PCB* una vez fabricado es el que se puede ver en la siguiente figura:



**Fig. 3.25 PCB fabricada**

Una vez se ha obtenido la placa y se sueldan todos los componentes, el resultado final es el siguiente:



**Fig. 3.26 PCB montado**

Como podemos ver en la *Fig. 3.26 PCB montado*, el PCB que se ha diseñado está finalizado con todos los componentes que se han comentado en los anteriores apartados y listo para ponerse en funcionamiento después de programarlo como es debido.

## Capítulo 4. Resultados

En este capítulo, se va a presentar la interpretación de resultados analíticos, así como de simulación y experimentales, de la solución adoptada para el proyecto y que hemos estado introduciendo en los apartados anteriores.

En un primer lugar, se presentarán las medidas de consumo del dispositivo de seguimiento en funcionamiento, de manera que podamos analizar los resultados. A continuación, mediante pruebas en el laboratorio y de campo, se comentarán los resultados obtenidos y se verá como la solución adoptada funciona y cumple con las funciones que se le han indicado que desempeñe.

### 4.1 Medidas de consumo

Para las medidas de consumo se va a tener en cuenta el consumo de corriente del módulo en funcionamiento, mientras este realiza las funciones que en un futuro realizará pegado al animal. Hay que tener en cuenta que un consumo bajo es muy importante dado que el dispositivo funcionará alimentado por una batería externa que se cargará mediante la energía solar cuando se disponga de esta, por lo que se debe procurar un consumo bajo del dispositivo de seguimiento en todo momento para que con la batería externa y la carga mediante la luz solar sea suficiente para que se mantenga en funcionamiento durante un tiempo que será indefinido.

Se va a utilizar un multímetro digital para realizar las medidas de corriente, para ello se configura el multímetro y se intercala al circuito del dispositivo de seguimiento, y al alimentar el circuito el módulo sigue su funcionamiento habitual mientras que el multímetro mide la corriente que consume en cada operación. En la siguiente figura se observa cómo se han realizado las medidas:

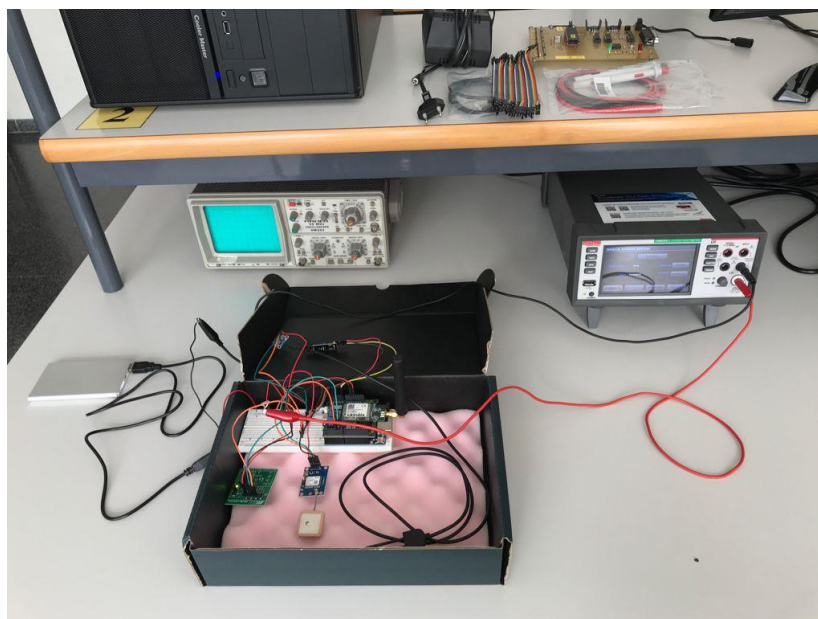
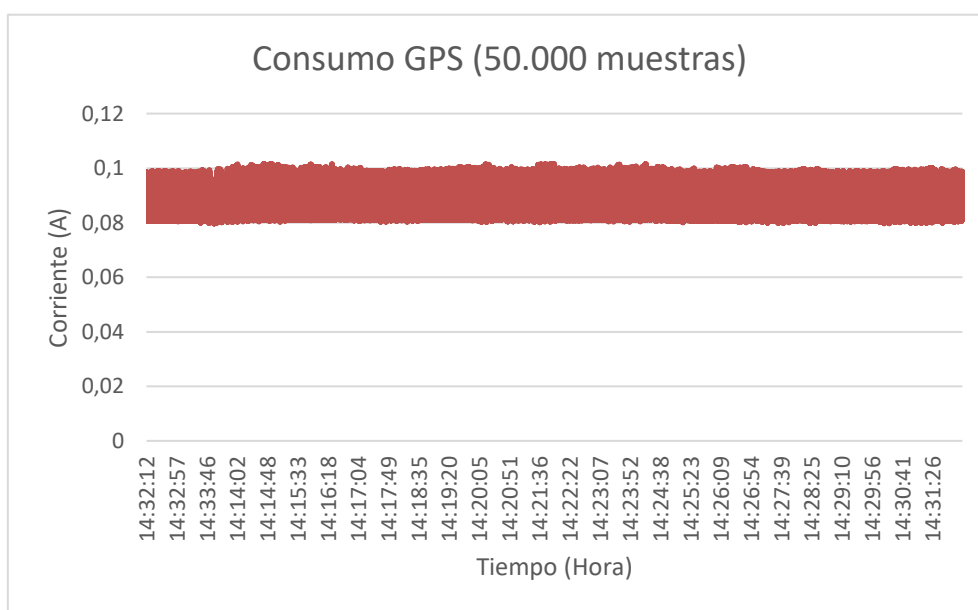


Fig. 4.1 Medida de consumo con multímetro digital

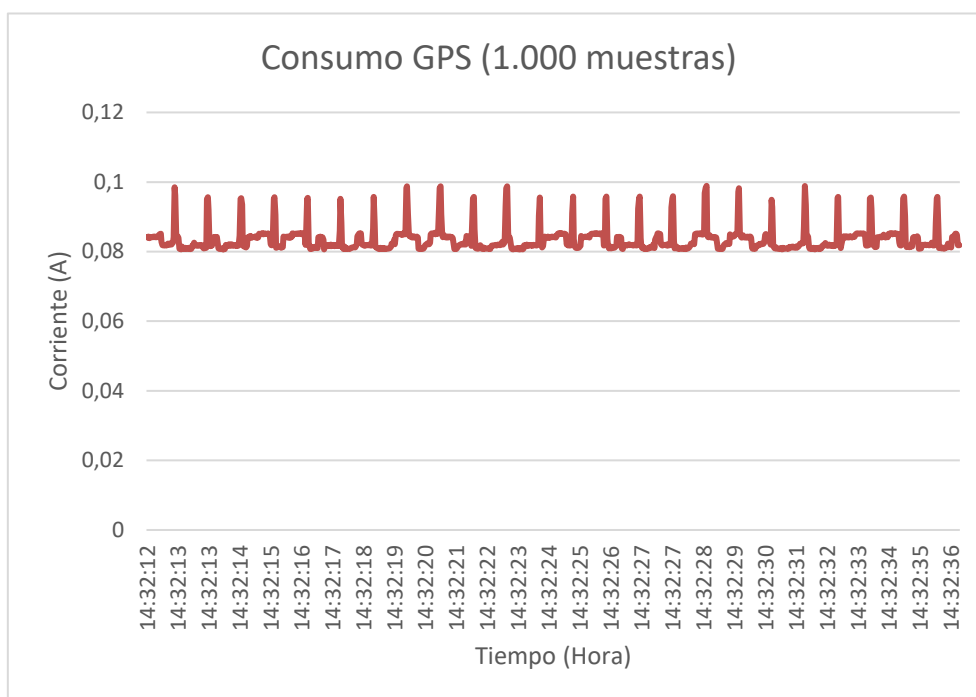
Una vez se ha completado la medida, obtenemos los datos del multímetro mediante un *USB* con los que se puede crear un gráfico donde podemos ver los distintos consumos de corriente del dispositivo de seguimiento durante el tiempo en el que se están realizando las medidas. A continuación, se va a analizar el consumo de cada sensor, así como el del módulo completo con todos los sensores funcionando según la implementación de código que se ha propuesto.

En primer lugar, se realiza la medida del sensor *GPS GY-GPS6MV2*, el que con el modo de ahorro de energía *Power Save Mode (PSM)* reduce el consumo de corriente del sensor, apagando y encendiendo partes del módulo, a aproximadamente 11mA según fuentes que han comprobado el consumo del sensor [27]. Este consumo de corriente lo podemos comprobar en la siguiente gráfica en la que se ha hecho la medida del módulo *GPS* con el multímetro digital:



**Fig. 4.2 Medida de consumo del módulo GPS (50.000 muestras)**

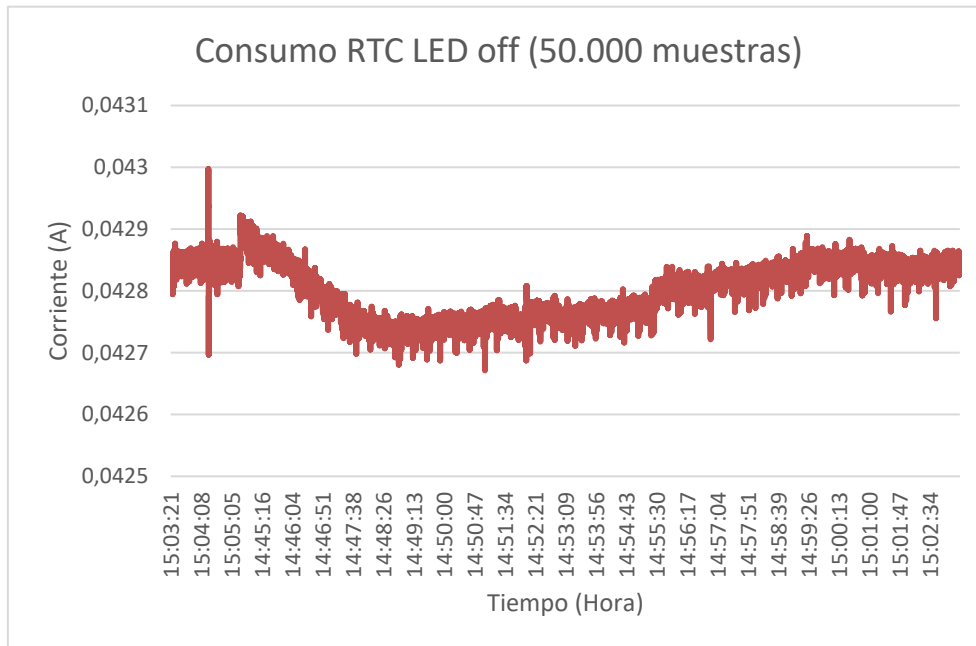
En la gráfica de la figura *Fig. 4.2 Medida de consumo del módulo GPS (50.000 muestras)* podemos comprobar como el consumo oscila entre los 8mA y los 11mA, siendo 8,4mA la media de las 50.000 muestras, confirmando de esta manera el consumo del sensor. Teniendo en cuenta que el *GPS GY-GPS6MV2* tiene un indicador *LED* que avisa de que el módulo ha obtenido efemérides, se hace un análisis más minucioso para que sea posible obtener los resultados más exactos:



**Fig. 4.3 Medida de consumo del módulo GPS GY-GPS6MV2 (1.000 muestras)**

En la figura *Fig. 4.3 Medida de consumo del módulo GPS GY-GPS6MV2 (1.000 muestras)* vemos una gráfica más precisa del consumo del sensor, en la que podemos ver cómo los picos que llevan al sensor a consumir entre 10mA y 11mA son debidos al *LED* indicador de que el módulo ha conseguido obtener las efemérides, por lo que podríamos definir un consumo más real del sensor *GPS* que sería de 8mA aproximadamente.

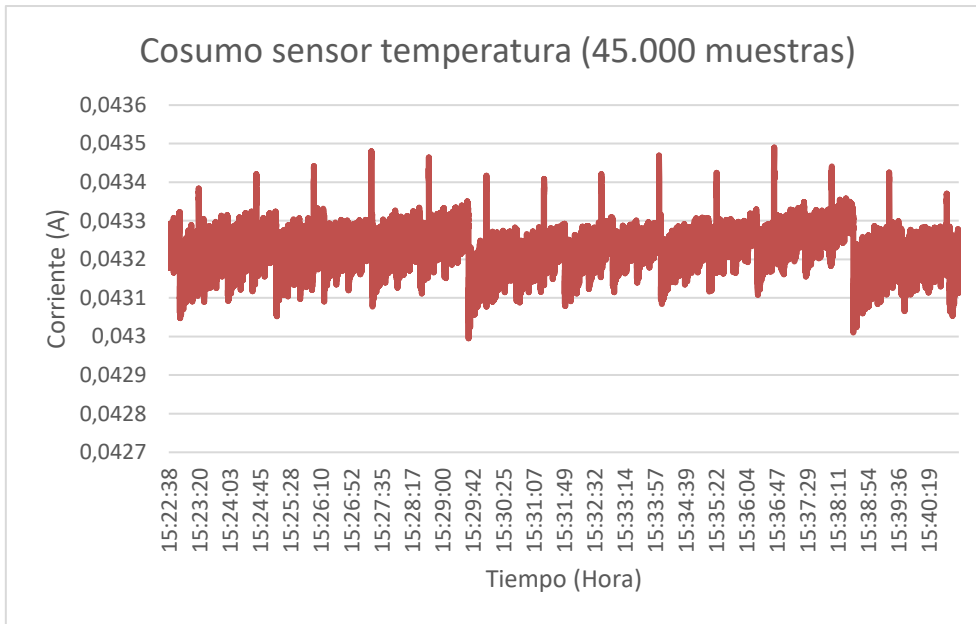
El siguiente sensor del que se ha hecho un análisis de la corriente que consume en funcionamiento es de la placa de evaluación del *RTC RV-3028-C7*. Viendo el *datasheet* del componente el consumo de corriente de este *RTC* es de 60nA, y no se especifica un consumo para la placa de evaluación. Al hacer las medidas con el multímetro digital obtenemos los siguientes resultados:



**Fig. 4.4 Medida de consumo del RTC RV-3028-C7 (50.000 muestras)**

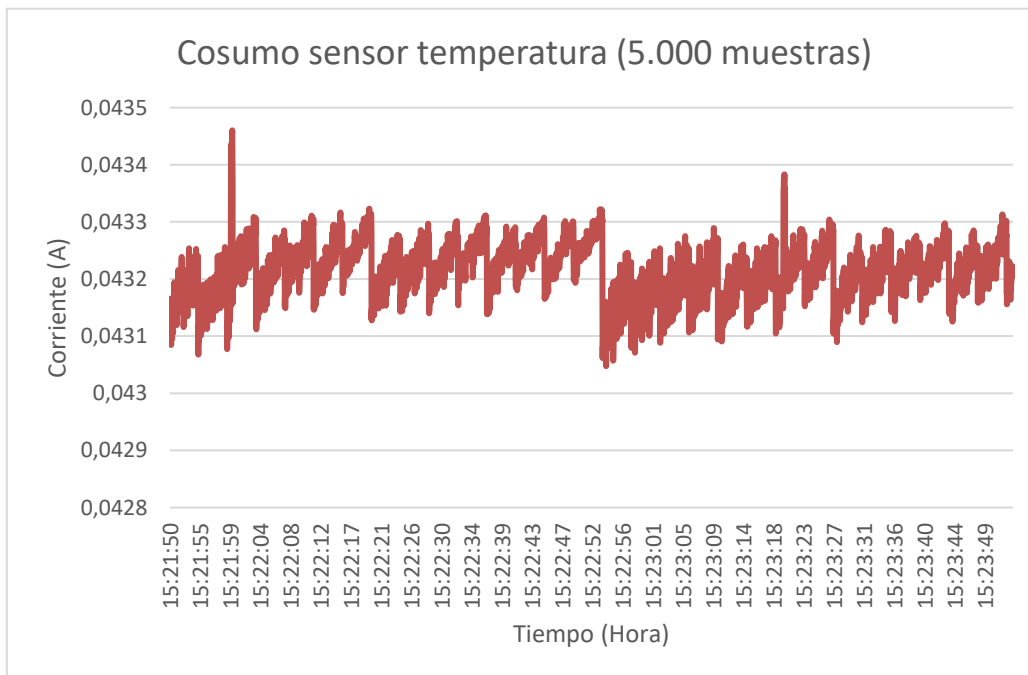
Las medidas de consumo que obtenemos y que podemos ver en la gráfica de la figura *Fig. 4.4 Medida de consumo del RTC RV-3028-C7 (50.000 muestras)* muestran que la corriente de la placa de evaluación del RV-3028-C7 se mantiene entorno a los 42,8mA siendo esta la media de las 50.000 muestras, un consumo bastante superior al que se especifica para el módulo RTC sin su placa de evaluación, pero teniendo en cuenta que en estas medidas también se cuenta el consumo del *Arduino UNO*, es algo que se preveía.

Seguidamente vamos a analizar el consumo del sensor de temperatura *MCP9808*. Según el *datasheet* de este sensor, el consumo de corriente en funcionamiento es de 200µA, y cuando este se apaga mientras no está midiendo temperatura para ahorrar el consumo es de 0,1µA. En la siguiente gráfica tenemos el consumo medido con el multímetro digital:



**Fig. 4.5 Medida de consumo del sensor de temperatura MCP9808 (45.000 muestras)**

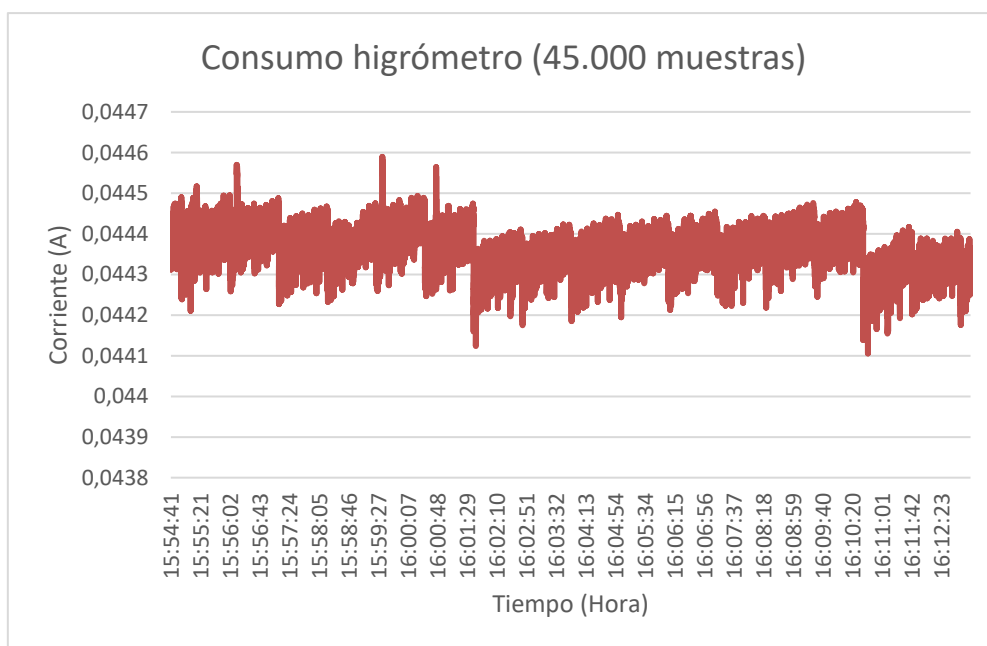
Podemos ver las medidas de consumo del sensor de temperatura en la gráfica de la figura *Fig. 4.5 Medida de consumo del sensor de temperatura MCP9808 (45.000 muestras)* que el consumo del módulo de temperatura es de unos 43mA, siendo ese valor de consumo de corriente el promedio de las 45.000 muestras. En cuanto a los picos que vemos en la figura, coinciden con el momento en el que el sensor está midiendo temperatura, y lo podemos ver más detalladamente en la siguiente figura, donde vemos la representación de 5.000 muestras y en la que se pueden apreciar dos medidas de temperatura:



**Fig. 4.6 Medida de consumo del sensor de temperatura MCP9808 (5.000 muestras)**

En el momento en el que se toman las medidas de corriente el código tiene configurado el sensor para que tome medidas de temperatura cada 80 segundos, y así se ve en la gráfica de la figura *Fig. 4.6 Medida de consumo del sensor de temperatura MCP9808 (5.000 muestras)*, donde los picos de medida se encuentran separados este tiempo.

El último sensor del que se van a tomar medidas de consumo de corriente es del higrómetro *HD-38*, el cuál según las referencias que se pueden obtener es de menos de 20mA en funcionamiento. En la siguiente gráfica podemos comprobar el consumo de este sensor en funcionamiento:

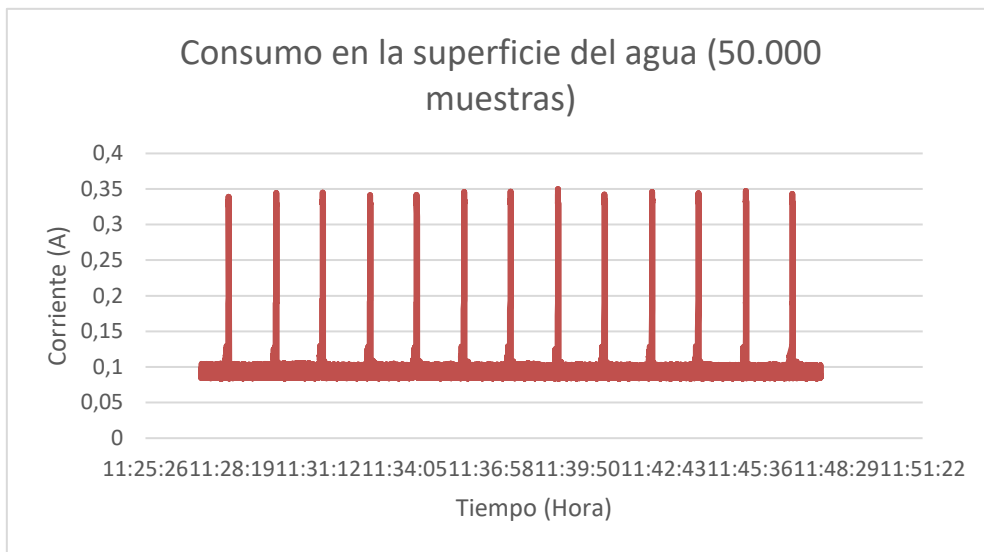


**Fig. 4.7 Medida de consumo del sensor de humedad HD-38 (45.000 muestras)**

Como podemos comprobar en la gráfica de consumo del higrómetro de la figura *Fig. 4.7 Medida de consumo del sensor de humedad HD-38 (45.000 muestras)*, el consumo del sensor de humedad es de unos 44mA y así lo indica el promedio de las 45.000 muestras tomadas en la medida de este módulo.

Una vez se ha comprobado el consumo individual de los sensores que se van a utilizar para el dispositivo de seguimiento propuesto, se hace el montaje del módulo completo y se procede a realizar las medidas de consumo necesarias para el estudio de este:

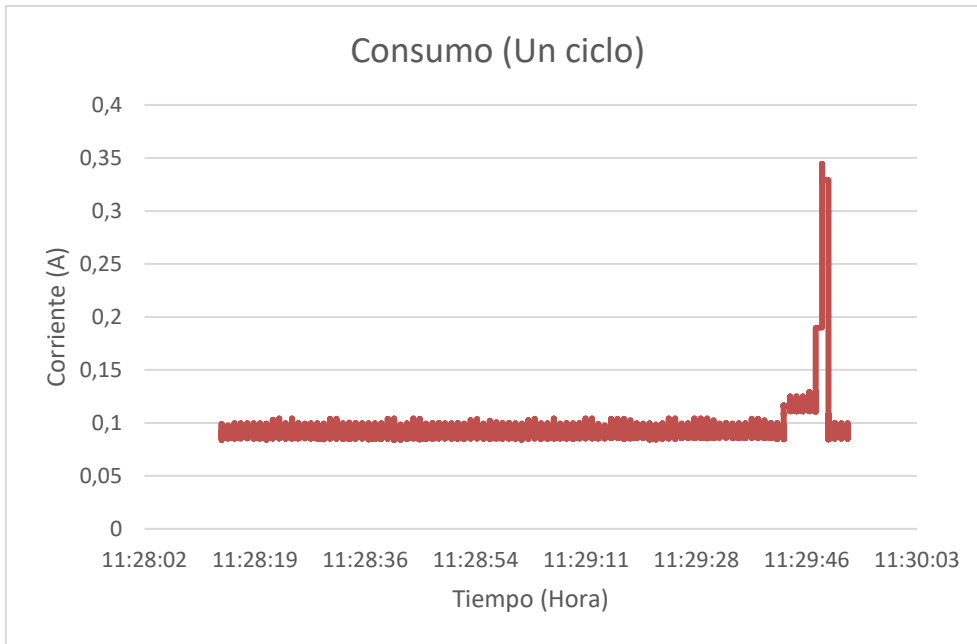




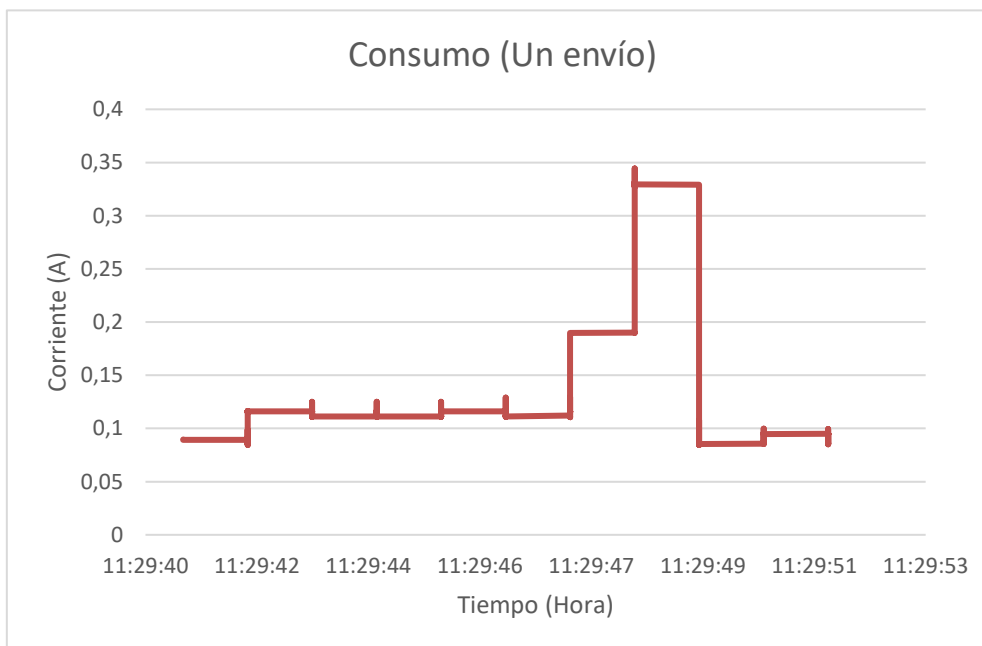
**Fig. 4.8 Medida de consumo del dispositivo (50.000 muestras)**

Cómo podemos ver en el gráfico de la *Fig. 4.8 Medida de consumo del dispositivo (50.000 muestras)*, que representa 50.000 muestras tomadas mientras el módulo de seguimiento estaba en funcionamiento, el consumo del dispositivo cuando se encuentra en la superficie del agua se mantiene entre los 70mA y los 100mA, y este sube a un valor de entre 110mA y 115mA cuando el *KIM1* que está en modo *sleep* se enciende para enviar los datos que se han recogido con los sensores integrados. El envío del *KIM1* se puede apreciar en la gráfica cuando el consumo sube a un valor entre los 345mA y los 350mA, siendo el pico máximo de consumo del dispositivo de seguimiento. Con los datos que se obtienen después de realizar las medidas de corriente del módulo de seguimiento, con 50.000 muestras tomadas durante un intervalo de tiempo de unos 20 minutos en los que el dispositivo sigue su funcionamiento normal, el promedio de consumo en todo este tiempo es de 91.857164mA exactamente.

El tiempo que transcurre desde que el módulo satelital *KIM1* se enciende hasta que envía los datos al satélite es de 7 segundos, y esto se puede comprobar en la siguientes figuras donde tenemos el gráfico de la *Fig. 4.9 Medida de consumo del dispositivo (Un ciclo)* en el que podemos ver más detalladamente el consumo de un ciclo de envío de datos que tiene un periodo de un minuto y medio, y otro gráfico de la *Fig. 4.10 Medida de consumo del dispositivo (Un envío)* que nos muestra en profundidad el consumo del momento del envío, que va desde que el *KIM1* se enciende hasta que este envía los datos para que los reciba el satélite:



**Fig. 4.9 Medida de consumo del dispositivo (Un ciclo)**

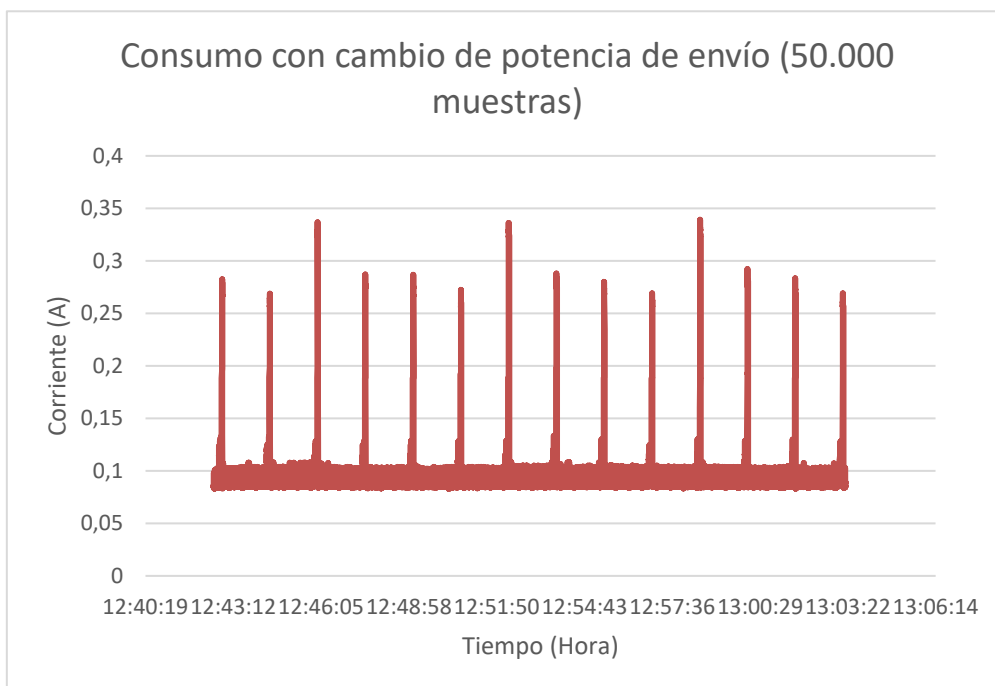


**Fig. 4.10 Medida de consumo del dispositivo (Un envío)**

De esta manera, teniendo en cuenta el consumo en reposo y el consumo al realizar el envío del módulo de seguimiento, se puede hacer un cálculo para saber la batería que se necesitará para mantenerlo activo durante el tiempo que se considere necesario. Para ello, primero se calcula el consumo de amperios-hora (Ah), que cuando el módulo esté en reposo será de 0,1Ah y cuando se realicen envíos cada 90 segundos será de 0,135Ah. Una vez tenemos estos datos, y sabiendo que el módulo va a transmitir 4 horas al día y las otras 20 horas se encontrará en reposo, podemos calcular el consumo diario que es de 2,54 Ah/día, lo que con una tensión de alimentación de 5V son 12,7Wh/día de energía consumida. A partir de estos datos, se puede calcular también el consumo anual del módulo de manera que obtenemos que consume 927Ah anuales, que por otra parte sería la capacidad mínima de la batería para que el módulo

funcionase durante un año, y que transformado a energía son 4635Wh anuales. Con estos cálculos de consumo podemos pasar al siguiente paso que sería realizar un diseño de los paneles solares necesarios para que el *tag* tenga una mayor vida útil haciendo uso de la energía solar. Todos estos datos que se acaban de mencionar son aproximaciones a la realidad y se han calculado teniendo en cuenta que el *tag* transmitirá todo el tiempo en las horas que se le han asignado, pero no se puede calcular exactamente por el hecho de que no se sabe cuánto tiempo la tortuga estará sumergida con el módulo en reposo o cuánto tiempo estará en la superficie transmitiendo.

Estas medidas que acabamos de ver anteriormente con sus correspondientes consumos, son para una potencia de 1000mW que es la potencia que viene de serie configurada en el *KIM1 Shield* y con la que se consigue una mejor transmisión del mensaje con los datos de los sensores a los satélites, pero debido a que se deben examinar todas las posibilidades y, como se ha comentado en la introducción de este apartado, el consumo es un parámetro crítico en este proyecto debido a que no se tendrá la posibilidad de cargar la batería que alimente al dispositivo de manera manual, sino que se cargará mediante la energía de la luz solar, se procede a hacer una prueba con distintas potencias de envío para de esta manera observar cómo cambia el consumo. Después de implementar un código que cambia la configuración del parámetro de potencia de envío del *KIM1* después de cada envío, se puede ver en el gráfico de la *Fig. 4.11* Medida de consumo del dispositivo con cambio en la potencia de envío (50.000 muestras):



**Fig. 4.11** Medida de consumo del dispositivo con cambio en la potencia de envío (50.000 muestras)

En el gráfico se puede comprobar efectivamente como si se disminuye la potencia de envío del *KIM1*, el consumo total del dispositivo también decrece, llegando a bajar casi 0.1A, lo que es una disminución a tener en cuenta para seguir haciendo pruebas y que de esta manera el dispositivo de seguimiento final tenga un consumo idóneo para su finalidad y las necesidades que esta implica.

## 4.2 Pruebas de funcionamiento

Para comprobar la funcionalidad del dispositivo de seguimiento, se han hecho pruebas de su funcionamiento en las que se tomaban medidas de los distintos sensores que hay en el módulo, se han almacenado estas medidas en una trama de datos y se han enviado al satélite, que después ha recibido la estación terrestre publicándolos en la web de *Argos* donde se pueden ver y estudiar. Estas pruebas se han realizado en un entorno adecuado para facilitar al máximo que el módulo funcione adecuadamente, para ello, se ha buscado un lugar al aire libre, con la mínima cantidad de edificios alrededor y donde se pueda instalar una estación de trabajo para ponerlo a funcionar. En la figura *Fig. 4.12 Pruebas de envío de datos* que tenemos a continuación vemos uno de los entornos en los que se han realizado pruebas de funcionamiento del dispositivo:



**Fig. 4.12 Pruebas de envío de datos**

Existe la posibilidad de descargar los datos en diferentes formatos dependiendo del uso que queramos darles, lo único que hay que hacer es introducir los criterios de búsqueda que se prefieran en el apartado *View your data* de la web y al buscar tendremos todos los mensajes que hayan sido recibidos por el satélite con los datos que se han enviado de los sensores. A continuación, se muestra una tabla con los resultados obtenidos descargados directamente de la web de *Argos* y con una breve descripción de cada parámetro:

<b>Platform ID No.</b>	205845	Identificación de la plataforma
<b>Platform</b>	FRAN	Nombre de la plataforma
<b>Prg No.</b>	7213	Identificación del programa
<b>Pass dur. (s)</b>	92	Duración del paso del satélite sobre la plataforma
<b>Msg Date</b>	44409,74822	Fecha y hora en la que se envió el mensaje
<b>Sat.</b>	A1	Identificador del satélite que ha recibido los datos
<b>Alarm</b>	N	Alarma de mensaje: N=None, D=Detection, C=Confirmation
<b>Format name</b>	FORMAT TEMPL. 31X8-A2-HEXA #1	Formato del mensaje
<b>Loc. date</b>	44409,74769	Fecha y hora en la que se localizó el satélite
<b>Longitude</b>	6.33709	Longitud calculada en la que se encuentra la plataforma
<b>Latitude</b>	38.40409	Latitud calculada en la que se encuentra la plataforma
<b>Altitude</b>	0.000	Altitud de la plataforma en km
<b>Heading</b>		Dirección de la plataforma
<b>Speed</b>		Velocidad de la plataforma
<b>Loc. quality</b>	B	Calidad de la localización
<b>Frequency</b>	401679734.82	Frecuencia de transmisión
<b>Comp.</b>	1	Índice de compresión: número de mensajes idénticos recibidos durante el paso del satélite
<b>SENSOR 01</b>	08	Valor del sensor 1 - Mes
<b>SENSOR 02</b>	01	Valor del sensor 2 - Día
<b>SENSOR 03</b>	E5	Valor del sensor 3 - Año
<b>SENSOR 04</b>	07	Valor del sensor 4 - Año
<b>SENSOR 05</b>	11	Valor del sensor 5 - Hora
<b>SENSOR 06</b>	39	Valor del sensor 6 - Minuto
<b>SENSOR 07</b>	13	Valor del sensor 7 - Segundo
<b>SENSOR 08</b>	E4	Valor del sensor 8 - Latitud
<b>SENSOR 09</b>	5C	Valor del sensor 9 - Latitud
<b>SENSOR 10</b>	1E	Valor del sensor 10 - Latitud
<b>SENSOR 11</b>	42	Valor del sensor 11 - Latitud
<b>SENSOR 12</b>	1C	Valor del sensor 12 - Longitud
<b>SENSOR 13</b>	7B	Valor del sensor 13 - Longitud
<b>SENSOR 14</b>	0E	Valor del sensor 14 - Longitud
<b>SENSOR 15</b>	BF	Valor del sensor 15 - Longitud
<b>SENSOR 16</b>	09	Valor del sensor 16 - Humedad
<b>SENSOR 17</b>	00	Valor del sensor 17 - Humedad
<b>SENSOR 18</b>	60	Valor del sensor 18 - Temperatura
<b>SENSOR 19</b>	0B	Valor del sensor 19 - Temperatura

Tabla 4.1 Datos recibidos del satélite en la web de Argos

También existe la opción de visualizar los mensajes que se han enviado a los satélites con los datos de los sensores en la aplicación *Argos Web*, en la que te tienes que registrar con un *ID* y una contraseña, y donde se puede consultar el historial de envío de mensajes de la plataforma satelital que elijas:



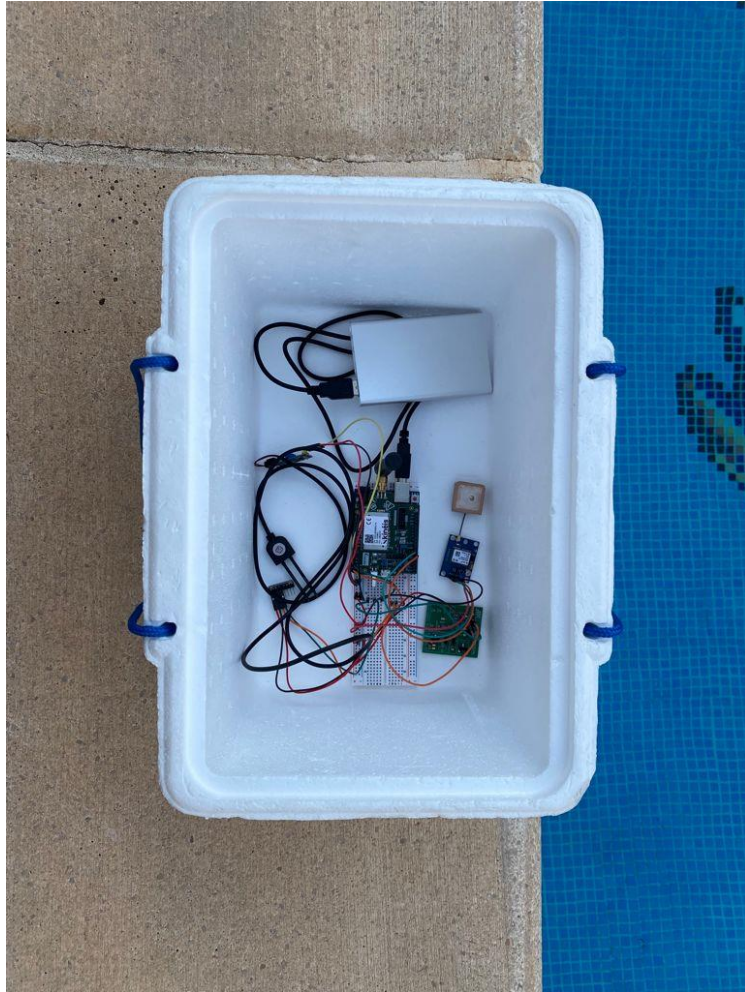
**Fig. 4.13** Aplicación Argos Web

### 4.3 Pruebas de campo

A parte de las pruebas de laboratorio, también se querían realizar pruebas de campo para asegurar que el funcionamiento en el hábitat de las tortugas era el correcto. Para ello, se quería utilizar la embarcación que utilizan los biólogos para hacer el seguimiento de los animales, utilizando una superficie que flotará simulando que la tortuga se encontraba en la superficie y moverse de posición poco a poco de manera que se comprobará que el dispositivo de seguimiento funciona correctamente. Esta prueba no ha sido posible realizarla por el hecho de que la embarcación tuvo una avería en el motor y sigue en proceso de reparación.

Para poder realizar una prueba de campo que se asemejara lo máximo posible a la situación antes descrita, se optó por utilizar como superficie flotante una nevera de poliespán y realizar las pruebas en una piscina, donde se puede simular que la tortuga está en la superficie del agua y que el dispositivo de seguimiento es capaz de enviar mensajes al satélite. En las siguientes figuras vemos la realización de las pruebas tal y como se han descrito:





**Fig. 4.14 Pruebas de campo del dispositivo (1)**



**Fig. 4.15 Pruebas de campo del dispositivo (2)**

Las pruebas resultaron satisfactorias por el hecho de que el satélite recibió los datos de los sensores mediante los mensajes enviados por el *KIM1*, y que a continuación se pudieron ver y descargar en la web de *Argos* tal y como se ha comentado en el apartado anterior.

Además, también se pudo simular que la tortuga se encontraba debajo de la superficie del agua, cogiendo los electrodos resistivos del sensor de humedad *HD-38* y sumergiéndolos en el agua, comprobando de esta manera que el dispositivo de comunicación satelital *KIM1* no envía mensajes y después de estar un tiempo sumergido el sensor y sacándolo a la superficie de nuevo, lo primero que hace el dispositivo de seguimiento es enviar un mensaje al satélite con los datos que tenga en ese momento.



## Capítulo 5. Trabajos futuros

En este capítulo vamos a plantear y justificar las siguientes tareas a realizar para que el proyecto evolucione y finalmente sea una solución totalmente funcional para la propuesta que se plantea, siendo este apoyado por *LIFE IP INTEMARES*, el proyecto europeo más grande de conservación del medio marino. Este proyecto, en el ámbito de seguimiento y vigilancia, tiene el objetivo de mejorar el seguimiento en los espacios marinos, con su consiguiente mejora del estado de conservación de los hábitats y especies que los habitan. Esta solución propuesta y el futuro trabajo que queda por realizar para conseguir que sea del todo funcional va a ser realizado por el personal de investigación de la Universidad Politécnica de Valencia, concretamente por el instituto de investigación para la gestión de integrada de zonas costeras del campus de Gandía.

Centrándonos en los trabajos futuros a realizar para llegar a la mejor solución para la propuesta expuesta, es seguro que se van a tener que añadir componentes y otros sensores al dispositivo de seguimiento como por ejemplo una batería que alimente el módulo en todo momento para que no deje de funcionar, así como también algunas placas solares con las que cargar la batería cuando la energía solar sea la suficiente como para alimentar estas placas. A parte, también hay otros sensores interesantes para el proyecto, como podría ser un acelerómetro con el que poder deducir el comportamiento del animal por el tipo de movimientos que realice, si se está moviendo a más velocidad significará que está desplazándose de zona por alimento o por necesidad, pero en cambio se mueve a una velocidad más lenta puede indicar que está alimentándose o buscando donde hacer su siguiente puesta de huevos. Además de añadir elementos al dispositivo, también puede que sea interesante cambiar algunos de los que se están utilizando actualmente, como puede ser el microcontrolador, que actualmente es el *Arduino UNO*, por otro que ofrezca mejores prestaciones como puede ser el microcontrolador *ESP32*, con más conectividad por el hecho de que tiene más del doble de pines, una mejora considerable de memoria, aunque carece de *EEPROM*, y algunas características adicionales como *Wifi* y *Bluetooth*. Otro elemento del dispositivo de seguimiento que va a tener que ser sustituido por una mejor versión y que se adapte a la solución va a ser la antena que se utiliza para comunicar el módulo de comunicación satelital *KIM1* con el satélite, buscando una opción más pequeña e integrable en el proyecto.

Otro aspecto que se va a tener que tener en cuenta para la solución final y que va a ser de vital importancia para que el proyecto funcione de forma adecuada y dé unos resultados óptimos, es que se reduzca lo máximo posible el consumo del dispositivo de seguimiento, para que de esta manera mediante el añadido de un diseño de *harvesting* con el que el dispositivo sea capaz de recargar la batería mediante el uso de la energía solar y el uso de placas solares, la energía del módulo no se agote en ningún momento y la carga de las placas solares sea suficiente para mantener el módulo funcionando constantemente desde el momento en el que se le incorpora al animal. Para ello, se va a tener que prescindir de todos los elementos que consuman y que no sean estrictamente necesarios, como por ejemplo pueden ser los *LEDs* que se utilizan en muchos sensores para indicar que están encendidos o cualquier funcionamiento en estos.

La aplicación del dispositivo va a ser en el medio marino, donde pasan la mayor parte de sus vidas las tortugas marinas, y es por esto que se va a necesitar un encapsulado con el que se proteja la electrónica del *tag* del agua y la salinidad. Para ello, se hará uso de una resina de epoxi de manera que como resultado se obtenga un encapsulado duro y resistente, aislando totalmente la electrónica del dispositivo del agua salada, además de protegerla de golpes e

impactos que pueda recibir, así como de vibraciones que puedan ocasionar que los componentes sufran de alguna manera. Además, la resina de epoxi tiene unas características muy interesantes que beneficiarán el diseño del *tag*, por el hecho de que tiene una conductividad térmica alta, y disipará el calor que puedan producir los componentes, así como también es un buen aislante eléctrico de manera que aislará el dispositivo del exterior.

Además, será necesario una mejora en el *PCB* en cuanto a optimización del espacio y distribución de los componentes, de manera que se reduzca lo máximo posible el tamaño y el peso del *tag*, sin quitarle funcionalidad. Hay que tener en cuenta en todo momento que este dispositivo va a ir adherido al caparazón de una tortuga que se va a pasar su vida nadando por el mar, y es de vital importancia que el *tag* sea lo menos molesto posible para el animal ya que no puede interferir en absoluto con el día a día de este. Por ejemplo, un módulo demasiado grande podría ocasionar problemas al engancharse en cualquier superficie restándole movimiento a la tortuga.

También será muy importante una depuración y optimización del *software*, de manera que el *tag* funcione muy controlada y precisamente, ya que una vez se lance el dispositivo al mar en el caparazón de una tortuga ya no se tiene la certeza de que este se vaya a recuperar nunca más. El objetivo del dispositivo es que proporcione datos durante el mayor tiempo posible antes de que deje de funcionar, por lo que para que la vida útil del *tag* sea lo más larga posible, aparte de hacer una selección de componentes buenos y eficientes, es importante que el *software* que se utilice también lo sea.

Con esto tendríamos el dispositivo de seguimiento completo y listo para poder adherir al animal con el fin de dar comienzo a las actividades de seguimiento de este. Otra futura tarea que se podría plantear, sería la de desarrollar una aplicación para el almacenamiento de los datos que se reciban desde el satélite y que se descarguen de *Argos*, de manera que se pueda tener una base de datos propia en la que se almacenen todos y cada uno de los mensajes que el dispositivo de seguimiento envíe desde la posición del animal, y de esta manera sea posible hacer un mejor estudio del comportamiento y de las distintas rutas que este pueda tomar.

## Capítulo 6. Conclusión

Los resultados de esta primera aproximación a la solución final de la propuesta, aunque indican que el funcionamiento del dispositivo de seguimiento es el correcto, también muestran que son muchas las necesidades que se deben abarcar en este proyecto para que pueda ser totalmente funcional.

Haciendo un sumario del planteamiento del proyecto y de los objetivos que se han conseguido, con este trabajo se dan a conocer algunas de las compañías de telemetría más importantes en el panorama de comunicaciones satelitales, haciendo una investigación de cada una de ellas y profundizando en los sectores a los que se dirigen, así como también dando a conocer que soluciones *hardware* ofrecen. También se dan a conocer compañías del sector de seguimiento y algunos de los métodos que utilizan para la monitorización de especies según el medio en el que habiten.

Por otro lado, se ha realizado un análisis de diferentes tipos de sensores para su uso en el dispositivo final de seguimiento, comparando entre distintos modelos y fabricantes hasta dar con los que más beneficiaban al proyecto. Para ello, se ha recopilado una gran cantidad de información de las hojas de datos de los componentes, de manera que se ha llegado a conocer la tendencia en este tipo de dispositivos o sensores en la actualidad.

Se ha presentado una primera versión del dispositivo de seguimiento, después de realizar un diseño esquemático de los distintos componentes que forman el *tag*, y su posterior disposición en un *PCB* el cuál se ha fabricado y puesto a prueba para comprobar que tiene un correcto funcionamiento. Además, se ha implementado un código con las distintas operaciones que se han planteado seguir teniendo siempre en cuenta la finalidad del dispositivo que es el seguimiento de tortugas marinas.

Teniendo en cuenta las medidas que se han tomado del módulo del que se ha tratado en este estudio y como se ha indicado en el capítulo de futuros trabajos, los resultados no son completamente eficientes, por lo que hay una tarea de mejora a realizar en este aspecto. Por otro lado, las pruebas realizadas tanto en el laboratorio como en el campo de funcionamiento del dispositivo, han sido satisfactorias y los resultados obtenidos han sido los esperados, pudiéndose comparar con los resultados obtenidos de marcas de seguimiento que son totalmente funcionales y que están siendo utilizadas para el estudio del comportamiento y el seguimiento de animales desde hace tiempo.

Con esta solución que aquí se propone, se pretende crear una base desde la que partir para un proyecto de gran envergadura en el que se ha partido del estudio e investigación de las diferentes compañías de telemetría que dominan el mercado actualmente, estudiando a su vez también los módulos de comunicación satelital que ofrecían, comparándolos y contactando con las compañías para una mejor comprensión de las diferentes características de dichos módulos.

## Capítulo 7. Referencias

- [1] LIFE INTEMARES, «Proyecto Red Natura 2000,» 2017. [En línea]. Available: <http://intemares.es/>.
- [2] Gobierno de España, «Red Natura 2000,» 1992. [En línea]. Available: <https://www.miteco.gob.es/es/biodiversidad/temas/espacios-prottegidos/red-natura-2000/>.
- [3] «Argos System,» 1978. [En línea]. Available: <https://www.argos-system.org/>.
- [4] «Collecte Localisation Satellites,» 1986. [En línea]. Available: <https://www.cls-telemetry.com/>.
- [5] «Globalstar,» 2021. [En línea]. Available: <https://www.globalstar.com/es-es/>.
- [6] «IMS Corporate,» 2021. [En línea]. Available: <https://www.ims corporate.com/>.
- [7] «Iridium Communications Inc.,» 2005. [En línea]. Available: <https://www.iridium.com/>.
- [8] Argos CLS, «KIM1,» [En línea]. Available: <https://www.cls-telemetry.com/argos-solutions/argos-products/modems/kim1-beta/#downloads>.
- [9] Argos CLS, «KIM1 Shield,» [En línea]. Available: <https://www.cls-telemetry.com/argos-solutions/argos-products/modems/kim1-interface-board/>.
- [10] Globalstar, «SRTX3,» [En línea]. Available: <https://www.globalstar.com/es-es/products/commercial-iot/stx3>.
- [11] FCC, «Federal Communications Commission,» [En línea]. Available: <https://www.fcc.gov/wireless/bureau-divisions/technologies-systems-and-innovation-division/rules-regulations-title-47>.
- [12] Globalstar, «STINGR,» [En línea]. Available: <https://www.globalstar.com/es-es/products/commercial-iot/stingr>.
- [13] Iridium, «Iridium 9602,» 2005. [En línea]. Available: <https://www.iridium.com/products/iridium-9602/>.
- [14] Iridium, «Iridium 9603,» 2005. [En línea]. Available: <https://www.iridium.com/products/iridium-9603/>.
- [15] Iridium, «16130 Mini-PCIe Adapter,» 2005. [En línea]. Available: <https://www.iridium.com/products/16130-mini-pcie-adapter/>.
- [16] «Wildlife Computers Inc.,» 1984. [En línea]. Available: <https://wildlifecomputers.com/>.
- [17] «Lotek wireless Inc.,» 2021. [En línea]. Available: <https://www.lotek.com/>.
- [18] Kinéis, «Hoja de datos KIM1,» 2019. [En línea]. Available: <https://www.cls-telemetry.com/wp-content/uploads/2019/09/KIM1-Beta-product-sheet.pdf>.
- [19] Microchip, «Hoja de datos TC1108,» 2012. [En línea]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/21357C.pdf>.
- [20] Microchip, «Hoja de datos MCP9808,» 2016. [En línea]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP9808-0.5C-Maximum-Accuracy-Digital-Temperature-Sensor-Data-Sheet-DS20005095B.pdf>.

- [21] Micro Crystal, «Hoja de datos RV-3028-C7,» 2021. [En línea]. Available: [https://www.microcrystal.com/fileadmin/Media/Products/RTC/App.Manual/RV-3028-C7\\_App-Manual.pdf](https://www.microcrystal.com/fileadmin/Media/Products/RTC/App.Manual/RV-3028-C7_App-Manual.pdf).
- [22] Quectel, «Hoja de datos L86 GNSS,» 2020. [En línea]. Available: [https://www.evelta.com/content/datasheets/Quectel\\_L86\\_Hardware\\_Design\\_V1.3.pdf](https://www.evelta.com/content/datasheets/Quectel_L86_Hardware_Design_V1.3.pdf).
- [23] onsemi, «Hoja de datos LMV331,» 2019. [En línea]. Available: <https://www.onsemi.com/pdf/datasheet/lmv331-d.pdf>.
- [24] Microchip, «Hoja de datos ATMEGA,» 2020. [En línea]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>.
- [25] Microchip, «Guía de usuario PICKit 4,» 2018. [En línea]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/PICKit4%20In-Circuit%20DebuggerUG-DS50002751B.pdf>.
- [26] FTDI Chip, «Hoja de datos FT232R,» 2020. [En línea]. Available: [https://ftdichip.com/wp-content/uploads/2020/08/DS\\_FT232R.pdf](https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf).
- [27] Acoptex, 2018. [En línea]. Available: <https://acoptex.com/project/258/basics-project-053b-neo-6m-gy-gps6mv2-gps-module-at-lex-c/>.
- [28] S. Abdel-Rahman, «Resonant LLC Converter: Operation and Design: Operation and Design 250W 33Vin 400Vout Design Example,» 2012.