**DOCUMENTACIÓN TÉCNICA**

**TALLER DE PROYECTOS I**

**SISTEMAS ELECTRÓNICOS**

Logotipo, Icono, nombre de la empresa

Descripción generada automáticamente

**Datalogger for IoT**

*Autores*: *Tutor*:

Andrés Martín Jesús Hernández Mangas

Pablo Villacorta

Rubén Serrano

Óscar Martín

XX de xxxxx de 2023

**Índice**

**Índice de figuras**

# Capítulo 1

# 1. Resumen Cierre de Proyecto

## 1.1 Datos básicos del proyecto

**Nombre del proyecto**: Datalogger for IoT

**Tipo de proyecto**: Diseño electrónico

**Sector**: Telecomunicaciones

**Subsector**: Electrónica

**Ubicación**: Escuela Técnica Superior de Ingenieros de Telecomunicación

**Localidad**: Valladolid

**Dimensiones del proyecto**: Cuatrimestral

**Nombre de los responsables del proyecto**: Andrés Martín

**Nombre de los responsables del proyecto**: Pablo Villacorta

**Nombre de los responsables del proyecto**: Rubén Serrano

**Nombre de los responsables del proyecto**: Óscar Martín

**Email de los responsables del proyecto**: andres.martin@estudiantes.uva.es

**Email de los responsables del proyecto**: @estudiantes.uva.es

**Email de los responsables del proyecto**: @estudiantes.uva.es

**Email de los responsables del proyecto**: oscar.martin.casares@estudiantes.uva.es

## 1.2 Resultado económico

## 1.3 Informe de situación final

# Capítulo 2

# Informe Técnico Detallado

## 2.1 Objetivos del proyecto

Antes de comenzar con el diseño de un proyecto, es conveniente definir cuál es el objetivo que se persigue. En primera instancia, se busca desarrollar un sistema de comunicación inalámbrica que permita capturar diferentes magnitudes físicas del medio ambiente a través de diferentes sensores. Asimismo, se quiere que los datos capturados mediante los sensores sean transmitidos a una infraestructura de ordenadores (*cloud*) para poder procesarlos posteriormente.

La propuesta inicial para obtener el objetivo mencionado fue el desarrollo de una satélite de comunicaciones o CubeSat, como los utilizados por agencias espaciales NASA o ESA. Dichos satélites poseen unas dimensiones de alrededor de 10x10x10 cm tal y como podemos observar en la figura XXXX, se estima que desde que se lanzó el primero en el año 2003 hasta día de hoy, se encuentran en el espacio más de 3.200 CubeSats. (Referencia: Wikipedia en inglés: <https://en.wikipedia.org/wiki/CubeSat>).

[[[SE PODRÍA PONER MUCHA MÁS INFORMACIÓN SI SE QUISIERA]]]



Figura 1. CubeSat desarrollado en Noruega.

Sin embargo, debido a su alta complejidad y coste, es completamente inviable para el taller de proyectos.

En segunda instancia, surge otra idea de diseño consistente en el desarrollo de una radio sonda capaz de obtener diferentes magnitudes. Éstas son comúnmente utilizadas en meteorología, se usan en globos meteorológicos que usan gases de elevación con el objetivo de medir parámetros atmosféricos y transmitirlos a un receptor fijo. Entre los parámetros más importantes nos podemos encontrar la presión, la altitud, la temperatura o la humedad relativa. (Referencia: Wikipedia en español <https://es.wikipedia.org/wiki/Radiosonda>)

[[[SE PODRÍA PONER MUCHA MÁS INFORMACIÓN SI SE QUISIERA]]]

Al contrario del satélite miniatura, la opción de la radiosonda es más plausible, por ello buscaremos aproximarnos a esta idea. Sin embargo, dado que una radiosonda se puede poner en marcha sin necesidad de lanzarla a las alturas, nosotros simplemente la desplazaremos con la ayuda de un vehículo y obtendremos datos de temperatura, humedad, presión, monóxido de carbono, partículas que estén geolocalizadas y transmitiremos dicha información mediante radio.

## 2.2 Especificaciones del proyecto

En conocimiento del sistema que se quiere diseñar, es conveniente presentar las ideas y componentes generales necesarios para la herramienta, podemos observar estos en la figura 2. Entre los componentes principales nos encontramos con una antena, una CPU, una FPGA o matriz de puertas lógicas programable en campo, una memoria *flash*, los sensores y su correspondiente alimentación.

Antena 
Inalámbricas 
Depuración 
Programación 
USB 
Señal Sensor 1 
Acondicionamiento 
Señal Sensor 2 
Acondicionamiento 
Señal Sensor 3 
Acondicionamiento 
Actuachr 1 
Memoria 
Acondicionamiento 
Actuaú»r 2 
Acondicionamiento 
Actuador 3 
Sensor 1 
Sensor 2 
Sensor 3 
Actuador 1 
Actuador 2 
Actuador 3 

Figura 2. Esquema general del Datalogger.

Dicho sistema se dispondrá en diferentes placas de circuito impreso o PCBs, montadas unas sobre otras de la manera mostrada en la figura 3. Con este montaje conseguiremos que se puedan reusar o sustituir partes no funcionales de manera muy sencilla, además de poder así rediseñar nuevas partes en el futuro.

En la primera placa nos encontraremos con la alimentación y la programación incluida en la unidad central de procesamiento o CPU. Contendrá los conectores de la batería y USB, el microntrolador o un puerto USB – serie. En segundo lugar, tendremos otra placa que incluirá la FPGA, conectores de acceso a parte del patillaje de la FPGA o la memoria *flash.* La tercera placa estará destinada a la incorporación de los diferentes de sensores, así como sus circuitos de acondicionamiento necesarios. Por último, la PCB restante contendrá las comunicaciones, es decir, el módulo de comunicaciones inalámbricas WiFi y el transceptor LoRa.

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Figura 3. Disposición de las placas de circuito impreso.

## 2.3 Planificación

Para poder gestionar el proyecto, existen diferentes herramientas de planificación y control. Una de dichas herramientas es el Diagrama de Gantt, la cual hemos utilizado para planificar nuestro trabajo y que nos proporciona una vista general de las tareas programadas además de cuáles han de completarse y en qué fecha.

En primer lugar, se debe definir la estructura de descomposición del trabajo en la cual deben listarse las tareas y proporcionarlas una estimación del tiempo necesario para completar cada una de ellas, uno de los cometidos más complicados. En los anexos Ay B [[[FALTA INCLUIR ANEXOS (GANTT)]]] se pude observar que la planificación inicial dista de la final. Al comienzo de un proyecto es muy complicado anticipar los problemas que van a surgir y más cuando se carece de experiencia en ello.

[[[EXPLICAR LAS DIFERENCIAS QUE HABRÁ ENTRE AMBOS DIAGRAMAS]]]

## 2.4 Herramienta de control de versiones

En el mundo profesional cada vez es más común el trabajo colaborativo y por ende las herramientas destinadas a ello. Es importante hoy en día estar familiarizado con sistemas de control versiones, la cuales permiten almacenar todas las versiones que ha habido durante la realización del proyecto, permitiéndonos además acceder a versiones antiguas en cualquier momento, actuando así a modo de ‘copias de seguridad’.

Para la realización de este proyecto se ha utilizado la mundialmente herramienta conocida como GitHub, disponible nuestro desarrollo en <https://github.com/pvillacorta/TP1SE>. Dicho repositorio ha sido organizado en distintas subcarpetas creadas sobre la marcha según las necesidades, con el objetivo de poder encontrar todo de manera ordenada e intuitiva, tal y como podemos observar en la figura 4. [[[ACTUALIZAR FIGURA]]]

En éste además hemos ido añadiendo en el fichero README.md todos los comentarios o instrucciones de utilidad necesarias durante todo el proyecto, para poder así recordar en cualquier momento cómo hicimos determinada acción o cosas de suma importancia.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 4. Esquema del repositorio de GitHub

Para manejarnos con la herramienta se han hecho uso de comandos básicos, entre los cuales nos podemos encontrar con:

* git clone ’url’: con este comando se ha podido clonar el repositorio en los respectivos ordenadores para poder trabajar de forma más cómoda
* git add \*: comando utilizado para añadir cambios de archivos en el directorio clonado.
* git commit -m ’nombre del commit’: Este comando toma todos los cambios de archivos descritos por el ’git add’, crea un nuevo objeto de confirmación y se establece una rama para que apunte a esta nueva confirmación.
* git push: comando que envía todos los objetos modificados localmente el respositorio remoto. Con el uso de este comando se ha podido ir subiendo las nuevas actualizaciones.
* git pull: comando utilizado para actualizar el repositorio local.
* git status: este comando muestra el estado de los archivos que hay en el repositorio remoto en comparación con los del directorio de trabajo. Esto ha servido para verificar que en todo momento el repositorio local se encontraba en buen estado sin errores.

[[[SE PODRÍAN INCORPORAR ESTADÍSTICAS DE GITHUB SI QUISIERAMOS]]]

## 2.5 Diseño electrónico y esquemático

Para llevar a cabo el desarrollo del *Datalogger* se ha recurrido a una herramienta software de automatización de diseño electrónico, en la actualidad nos encontramos con numerosas opciones punteras como pudieran ser *Eagle, Altium, KiCad* u *Orcad*. Sin embargo, en nuestro caso hemos utilizado *Proteus Professional* debido a lo familiarizados que nos encontrábamos con la herramienta. Ésta ha sido desarrollada por *Labcenter Electronics Ltd.*

En primer lugar, conociendo cuáles son las especificaciones del proyecto estamos en disposición de buscar qué componentes vamos a utilizar para elaborar el dispositivo.

### 2.5.1 Búsqueda de componentes

En el proyecto se ha trabajo principalmente con uno de los principales distribuidores de productos para el diseño de sistemas electrónicos, la empresa *Premier Farnell Ltd.* En ésta se han obtenido la mayoría de los componentes como pueden ser el microcontrolador ARM, diodos, leds, sensores, conectores, batería o transceptor. De cada uno de los componentes se han anotado sus números de referencia y su precio, para así poder elaborar un presupuesto al proyecto completo, el cual desglosaremos más adelante.

Además, dado que en este momento nos encontramos en una situación complicada a nivel mundial en cuanto al abastecimiento de este tipo de sistemas electrónicos, hemos adquirido un transceptor LoRa en la conocida empresa china *Aliexpress*.

Dado que no se quieren cometer fallos de diseño, una parte importante del proyecto consiste en analizar cuál será el consumo de cada elemento, por ello hemos obtenido la hoja de características de cada uno de ellos. Esta parte la podremos conocer en profundidad en el apartado de consumo eléctrico y térmico.

### 2.5.2 Creación de componentes

Una parte de los componentes los podemos encontrar en diferentes librerías, aunque haya que verificar con la hoja de especificaciones que dicho diseño sea correcto. Sin embargo, no todos la poseen, por lo que se requiere realizar su diseñoi para poder generar así el esquemático. Para crear la huella de un componente se han seguido los siguientes pasos en la herramienta de dibujo 2D:

1. Dibujar el contorno del componente.
2. Crear los pines de cada componente.
3. Dar nombre e indicar el tipo de pin.
4. Seleccionar todo el componente.
5. Seleccionar a la opción *make device* con el botón derecho.
6. Asignar las características del componente:
   1. Nombre completo.
   2. Tipo de componente.
   3. Asignarle una huella

Un ejemplo de componente creado [[PONER UN EJEMPLO, NO RECUERDO BIEN AHORA CUÁL ES NUESTRO PROPIO]]

Importante tener cuidado en la numeración de cada uno de los pines.

### 2.5.3. Posicionado y rutado de componentes

El diseño del *Datalogger* se ha llevado a cabo en diferentes hojas *Proteus*, donde cada una de ellas tiene un nombre representativo en función de los componentes que contiene. Nos encontramos con la siguiente estructura de hojas:

* PCB0 Power-USB.
* PCB1 FPGA.
* PCB2 Sensors.
* PCB3 Wireless.

En la hoja **PCB0 Power-USB** (figura 5) [[[ACTUALIZAR FIGURA]]] se han posicionado la batería, el microcontrolador ARM, el interruptor de alimentación, un cristal de cuarzo, un puente USB-serie, leds y los diferentes conectores, así como el USB mini B.

En la segunda hoja **PCB1 FPGA** (figura 6) [[[ACTUALIZAR FIGURA]]] se han incluido la FPGA, un regulador lineal ajustable, una memoria flash, un oscilador integrado y los diferentes conectores necesarios.

En la segunda hoja **PCB2 Sensors** (figura 7) [[[ACTUALIZAR FIGURA]]] se han incorporado los diferentes sensores incorporados, así como sus circuitos de acondicionamiento, el GPS, un convertidor ADC con interfaz SPI, un regulador conmutado síncrono *step-up* 5V, condensadores electrolíticos, un inductor y los diferentes conectores necesarios.

En la última hoja **PCB3 Wireless** (figura 8) [[[ACTUALIZAR FIGURA]]] se encuentran los diferentes transceptores, es decir el transceptor LoRa y su alternativa, el transceptor WiFi y los diferentes conectores necesarios.

En todas las hojas mencionadas se han ido posicionando diferentes pines de test, los cuales nos permitirán verificar el correcto funcionamiento del proyecto cuando las placas de circuito impreso hayan sido fabricadas y montadas.

Finalmente, se realiza el conexionado de todos los componentes, prestando mucha atención para no confundir cables y poner nombres equivocados. Destacar que sobre la marcha, dado que es un proceso largo, se han ido detectando fallos y se han corregido antes de mandar a fabricar.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 5. Esquemático de la placa PCB0 - Parte 1.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 5. Esquemático de la placa PCB0 - Parte 2.

Tabla, Excel

Descripción generada automáticamente

Figura 5. Esquemático de la placa PCB0 - Parte 3.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 6. Esquemático de la placa PCB1 - Parte 1.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 6. Esquemático de la placa PCB1 - Parte 2.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 6. Esquemático de la placa PCB1 - Parte 3.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 7. Esquemático de la placa PCB2 - Parte 1.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 7. Esquemático de la placa PCB2 - Parte 2.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Figura 7. Esquemático de la placa PCB2 - Parte 3.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 8. Esquemático de la placa PCB3 - Parte 1.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Figura 8. Esquemático de la placa PCB3 - Parte 2.

### 2.5.4. Generación de listado de materiales

Una vez seleccionados todos los componentes e insertados debidamente en el proyecto, de manera automática *Proteus* nos generara un listado con todos ellos, o también llamado *Bill of materials (BOM)*. Asimismo, para tener estos anotados correctamente como antes mencionamos, registramos su código de referencia asociado en la distribuidora *Farnell* y su precio, para así poder tener el presupuesto total.

Dicho listado de materiales requiere de una configuración previa de sus ajustes para poder ver el precio en unidades de euros y separar las décimas de céntimo mediante comas. Cada una de las placas diseñadas poseerá su propio listado, por ellos nos encontramos con cuatro de ellos, los podemos observar a continuación. Nos encontramos con que la placa donde se encuentra el microcontrolador asciende a 57,36 € (figura 9), la placa que contiene la FPGA asciende a 32,48 € (figura 10), la placa correspondiente a los sensores posee el precio más elevado, debido al su elevado precio, con 83,45 € (figura 11) y la última placa destinada a las comunicaciones posee un precio de 46,13€ (figura 12). [[[ACTUALIZAR FIGURAS]]]

Tabla

Descripción generada automáticamente

Figura 9. Listado de materiales de la placa PCB0 - Parte 1.

Imagen que contiene Tabla

Descripción generada automáticamente

Figura 9. Listado de materiales de la placa PCB0 - Parte 2.

Tabla

Descripción generada automáticamente

Figura 10. Listado de materiales de la placa PCB1 - Parte 1.

Escala de tiempo

Descripción generada automáticamente

Figura 10. Listado de materiales de la placa PCB1 - Parte 2.

Imagen que contiene Tabla

Descripción generada automáticamente

Figura 11. Listado de materiales de la placa PCB2 - Parte 1.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Figura 11. Listado de materiales de la placa PCB2 - Parte 2.

*Imagen que contiene Escala de tiempo

Descripción generada automáticamente*

Figura 12. Listado de materiales de la placa PCB3 - Parte 1.

*Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media*

Figura 12. Listado de materiales de la placa PCB3 - Parte 2.

### 2.5.5. Análisis de consumo eléctrico

El objetivo es realizar una lista con todos los consumos eléctricos de los componentes clasificándolos mediante tensiones, es decir, contabilizar cuál es la corriente máxima consumida para cada uno de ellos en el peor caso.

Este proceso se ha llevado a cabo a través de una serie de tablas, donde para cada tensión de alimentación hemos evaluado los distintos componentes que tiene conectados. Para cada elemento se ha evaluado su máximo consumo, condiciones en las que se dan dichos consumos, así como la ubicación de donde se encuentra dicha información en su hoja de especificaciones. Asimismo, se ha anotado cuál sería el consumo total para cada una de las cuatro placas, esta información la podemos observar en las figuras 13 a 16. [[[ACTUALIZAR FIGURAS]]]

Además, para alguno de los componentes se ha anotado la corriente quiescente o *quiescent current*, es decir, la cantidad de corriente utilizada por un circuito integrado cuando está en un estado quiescente. Este estado se da en cualquier periodo de tiempo en el que el circuito integrado está habilitado, pero sin actividad.

Nos encontramos entonces en disposición de calcular cuál sería la duración de la batería recargable de polímero de litio de 1000 mAh. Se ha obtenido que el consumo aproximado será de 1433,2 mA, por lo que utilizando la siguiente fórmula:

Tenemos que la duración aproximada de la batería será de 0,7 horas, o lo que es lo mismo, una duración de 42 minutos en el peor de los casos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente **Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

Figura 13. Análisis Eléctrico PCB0. Figura 14. Análisis Eléctrico PCB1.

### Una captura de pantalla de una red social Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación Descripción generada automáticamente

Figura 15. Análisis Eléctrico PCB2. Figura 16. Análisis Eléctrico PCB3.

### 2.5.6. Análisis de consumo térmico

Una vez realizado el análisis eléctrico de las cuatro placas, nos interesa conocer cuál será el calentamiento aproximado de los componentes electrónicos, es decir, poder controlar así que los semiconductores no varíen sus características técnicas y que posea una mayor longevidad de vida.

Para proceder con este análisis se ha calculado cuál es la potencia consumida por cada componente, las condiciones de temperatura en las que sucede, así como la ubicación en la hoja de características de dicha información. Asimismo, se ha calculado la resistencia térmica para los componentes que la precisan. Podemos encontrar dichos cálculos en formato de tabla de la figura 17 a la 20. [[[ACTUALIZAR FIGURAS]]]

Para llevarlo a cabo el cálculo de la potencia se ha utilizado la siguiente fórmula, en la que tendremos en cuenta cuál es la corriente que consume el elemento y la alimentación que éste posee.

Calculada dicha potencia, ha servido para calculas las distintas resistencias térmicas mediante la siguiente fórmula, en la que se ha tenido en cuenta la temperatura ambiente.

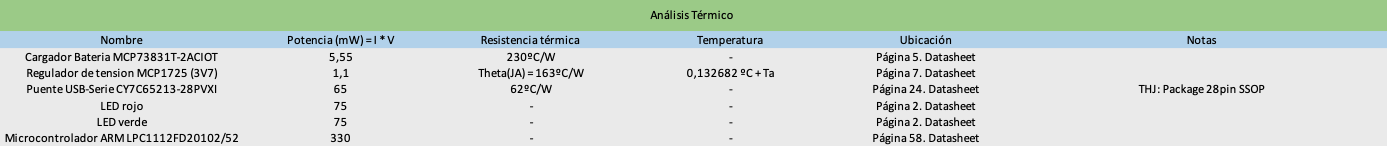
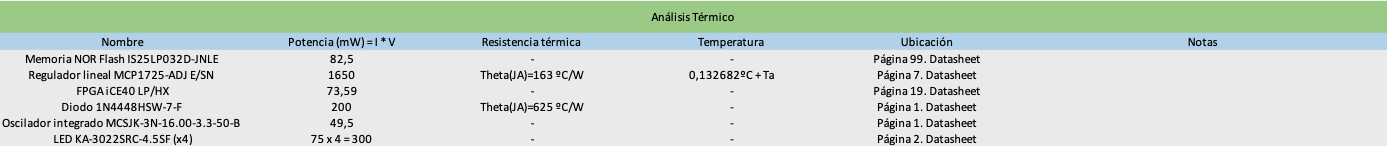
 

Figura 17. Análisis Térmico PCB0. Figura 18. Análisis Térmico PCB1.

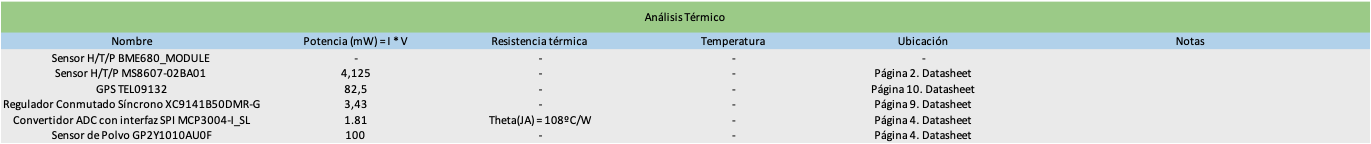
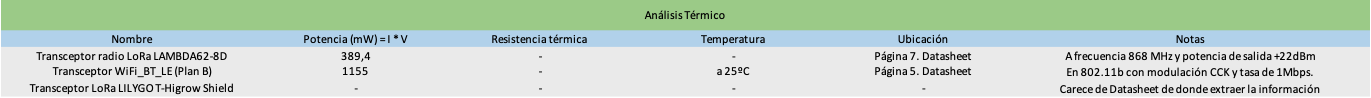
 

Figura 19. Análisis Térmico PCB2. Figura 20. Análisis Térmico PCB3.

### Siguiente apartado