

UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA  
CARRERA DE ESPECIALIZACIÓN EN SISTEMAS  
EMBEBIDOS



MEMORIA DEL TRABAJO FINAL

**Sistema de adquisición portátil de  
parámetros biomédicos para animales  
grandes**

**Autor:**  
**Ing. Federico Roux**

Director:  
Dr. Ing. Damian Craiem (Universidad Favaloro)

Jurados:  
Dr. Ing. Mariano Casciaro (CONICET - Fundación Favaloro)  
Esp. Ing. Jerónimo La Bruna (FIUBA)  
Ing. Manuel Alfonso (UTN.BA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre mayo  
de 2018 y diciembre de 2018.*



## *Resumen*

En la presente memoria se describe el desarrollo de un sistema de adquisición portátil para medición de parámetros biomédicos en animales grandes desarrollado para la Universidad Favaloro. En particular, se tiene interés en la medición de la velocidad de onda de pulso (VOP), cuyo método más aceptado de cálculo consiste en el registro simultáneo de señales de presión intraarterial en dos puntos del árbol arterial. Conociendo la distancia y el desfase temporal entre estas mediciones de presión, se puede estimar la VOP como su cociente.

Para este proyecto se utilizaron en forma intensiva técnicas de gestión de proyectos, documentación manual y automática del trabajo y sistema de versionado de software y hardware. Se emplearon conocimientos específicos sobre arquitectura del microcontrolador, modelos de programación, sistema operativo de tiempo real freeRTOS, protocolos de comunicación (BLE, SPI, USB, y de alto nivel), testing unitarios, etc.



## *Agradecimientos*

Se agradece especialmente al director de este proyecto, Dr. Ing. Damián Craiem y a su equipo de trabajo, por el esfuerzo y dedicación diarias en la investigación, los cuales generan nuevos campos de aplicación de la ingeniería electrónica. Y en particular por la continuidad de este proyecto a través del tiempo y la oportunidad de poder participar en el mismo.



# Índice general

<b>Resumen</b>	<b>III</b>
<b>1. Introducción General</b>	<b>1</b>
1.1. Descripción General del Trabajo . . . . .	1
1.1.1. Bases físicas de la medición . . . . .	2
1.1.2. Estado del arte . . . . .	2
1.1.3. Medicion ambulatoria de presión arterial . . . . .	4
1.2. Motivación y Aplicaciones del equipo . . . . .	4
1.3. Objetivos y alcance . . . . .	5
<b>2. Introducción Específica</b>	<b>7</b>
2.1. Detalle de las necesidades . . . . .	7
2.2. Requerimientos . . . . .	9
2.3. Planificación . . . . .	10
<b>3. Diseño e Implementación</b>	<b>13</b>
3.1. Implementación del firmware . . . . .	13
3.1.1. Resumen del sistema . . . . .	13
3.1.2. Modelo de desarrollo del firmware . . . . .	13
3.1.3. Analisis de los requerimientos . . . . .	14
3.1.4. Diseño de alto nivel . . . . .	20
3.1.5. Diseño Detallado . . . . .	23
3.1.6. Implementación del código . . . . .	32
3.1.7. Pruebas Unitarias . . . . .	34
3.1.8. Pruebas De Integración . . . . .	36
3.1.9. Pruebas de Sistema . . . . .	36
3.2. Implementación del Hardware . . . . .	37
3.2.1. Prototipo para adquisición del ADS1292 . . . . .	37
3.2.2. Prototipo para carga de batería y USB . . . . .	39
3.2.3. Diseño final del circuito impreso . . . . .	40
<b>4. Ensayos y Resultados</b>	<b>45</b>
4.1. Pruebas funcionales del hardware . . . . .	45
<b>5. Conclusiones</b>	<b>47</b>
5.1. Conclusiones generales . . . . .	47
5.2. Próximos pasos . . . . .	47
<b>Bibliografía</b>	<b>49</b>





# Índice de figuras

1.1. Esquema de medición de VOP . . . . .	1
1.2. Forma de onda característica de la presión humana . . . . .	2
1.3. Curva de inflado de la almohadilla de método oscilométrico vs. presión diferencial y ruidos detectados. . . . .	3
1.4. Oveja instrumentada con un equipamiento antiguo con interfaz cableada. . . . .	5
2.1. Conector, cable y sensores Königsberg de presión . . . . .	8
2.2. Señal típica de presión arterial de un mamífero. . . . .	9
2.3. Diagrama de gantt - Primer parte . . . . .	11
3.1. Diagrama en bloques del equipo. . . . .	14
3.2. Modelo de desarrollo y validación en V . . . . .	14
3.3. Diagrama de relaciones entre subsistemas. Patrón productor/consumidor . . . . .	22
3.4. Capas de software de la librería FatFs . . . . .	23
3.5. Diagrama de tareas completo . . . . .	24
3.6. Diagrama de tareas en modo inactivo . . . . .	25
3.7. Diagrama de tareas en modo adquirir y enviar . . . . .	26
3.8. Diagrama de tareas en modo adquirir, enviar y almacenar . . . . .	27
3.9. Diagrama de tareas en modo adquirir y almacenar . . . . .	28
3.10. Diagrama de tareas en modo conexión USB . . . . .	29
3.11. Diagrama de tareas en modo configuración . . . . .	30
3.12. Máquina de estados de tarea de interfaz de usuario . . . . .	31
3.13. Máquina de estados de configuración de usuario . . . . .	31
3.14. Esquemático de conexiones del conversor analógico digital de alta resolución ADS1292 . . . . .	37
3.15. Router CNC empleado para los dos primeros prototipos. . . . .	38
3.16. Vista superior del primer prototipo realizado. . . . .	38
3.17. Vista inferior del primer prototipo realizado. . . . .	38
3.18. Esquemático de conexión de la entrada USB . . . . .	39
3.19. Esquemático de conexión de la entrada USB . . . . .	39
3.20. Esquemático de conexión del LTC4411 que actúa como un diodo ideal. . . . .	40
3.21. Esquemático de conexión los reguladores. . . . .	40
3.22. Circuito impreso del prototipo para carga de batería y USB . . . . .	41
3.23. Vista superior del prototipo para carga de batería y USB . . . . .	41
3.24. Conexiones del microcontrolador LPC1769 . . . . .	42
3.25. Vista frontal y trasera del diseño CAD del impreso final . . . . .	43
3.26. Vista frontal y trasera del Circuito impreso Fabricado sin Componentes . . . . .	43



# Índice de Tablas

3.1. Matriz de Trazabilidad . . . . .	20
3.2. Subsistemas del equipo . . . . .	21



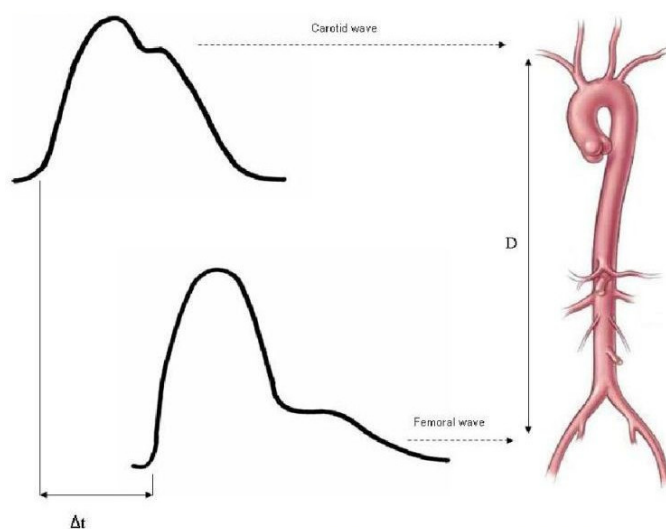
# Capítulo 1

## Introducción General

En el presente capítulo se describen los lineamientos generales del trabajo, sus motivaciones en el campo de la Ingeniería Biomédica, el estado del arte actual y los objetivos específicos alcanzados.

### 1.1. Descripción General del Trabajo

El trabajo aquí presentado consiste en un sistema de adquisición portátil para medición de parámetros biomédicos en animales grandes. Se tiene particular interés en la medición del parámetro biológico denominado «velocidad de onda de pulso (VOP)», cuyo método de cálculo más aceptado consiste en el registro simultáneo de señales de presión intraarterial en dos puntos del árbol arterial. Conociendo la distancia y el desfase temporal entre estas mediciones de presión, se puede estimar la VOP como su cociente. Esto puede verse graficado en la figura 1.1.



*La medición de VOP se realiza a partir de dos mediciones de la curva de presión, con un desfase temporal conocido.*

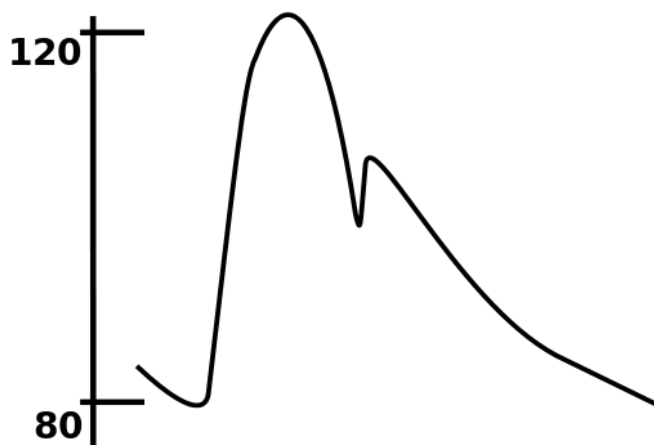
FIGURA 1.1: Esquema de medición de VOP

El dispositivo desarrollado en este trabajo está orientado a ser utilizado por investigadores médicos, ingenieros biomédicos, biólogos y veterinarios en grupos de trabajo en general aplicado a cardiología. En particular este trabajo se realiza para la Universidad Favaloro, y es importante porque permitirá avanzar a futuro

en el conocimiento de formas de medición de presión ambulatoria indirecta, más precisas y menos invasivas.

### 1.1.1. Bases físicas de la medición

De acuerdo a la Organización Mundial de la Salud, la presión sanguínea humana es medida de la presión que ejerce la sangre circulante sobre las paredes de los vasos sanguíneos [1]. Esta onda de presión es originada por el latido del corazón y transmitida hacia todo el árbol arterial. La composición entre la onda incidente y la reflejada dan origen a la onda estacionaria de presión, cuyo máximo y mínimo dan origen a los valores característicos conocidos como **presión sistólica** y **presión diastólica**. Estos valores tienen una gran importancia en la clínica. En un adulto sano, estos valores son aproximadamente 120mmHg para la presión sistólica y 80mmHg para la presión diastólica. En la figura 1.2 puede observarse una forma de onda característica.



*La forma de onda característica de la presión humana se puede caracterizar como una onda estacionaria conformada por una onda pulsátil incidente y una onda reflejada.*

FIGURA 1.2: Forma de onda característica de la presión humana

### 1.1.2. Estado del arte

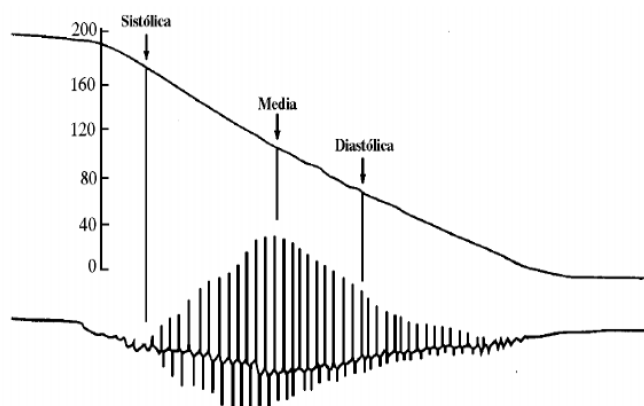
La medición ambulatoria de presión arterial (MAPA) braquial durante 24 horas es un método ampliamente aceptado para predecir riesgo cardiovascular y mortalidad[2] [3] [4]. Se sabe que la variabilidad de la presión arterial y su pulsatilidad son el resultado de una compleja interacción entre el corazón y la red vascular. En particular, y para la evaluación de las características biomecánicas de la red arterial, se utiliza la velocidad de onda de pulso (VOP) como indicador indirecto de rigidez arterial [5].

Actualmente, existen diferentes técnicas para la medición de presión [6]. Pueden agruparse entre los métodos que brindan mediciones puntuales (presión sistólica, diastólica y media) y los que brindan mediciones continuas. Entre los que brindan mediciones puntuales están el método auscultatorio, el método oscilométrico, método palpatorio, técnicas de ultrasonido, etc. Por otro lado, los que

brindan mediciones continuas suelen ser métodos invasivos como el catéter o no invasivos como la medición indirecta a partir de la medición de VOP [7].

La técnica más utilizada y difundida en la medicina clínica es el método oscilométrico, que basa su funcionamiento en monitorear las variaciones de la señal de presión en una banda inflable que se aplica alrededor del brazo izquierdo, logrando determinar a través del análisis de esta señal los valores de presión sistólica, diastólica y media de los pacientes. Mientras la banda se desinfla desde un nivel por encima de la presión sistólica, las paredes de la arteria comienzan a oscilar a medida que la sangre fluye a través del vaso parcialmente ocluido, y estas vibraciones son captadas en el transductor que monitorea la presión en la banda. Cuando la presión en la banda sigue disminuyendo, las oscilaciones aumentan hasta una amplitud máxima y luego disminuyen hasta que la banda se desinfla completamente y el flujo de sangre regresa a la normalidad.

La presión en la banda en el punto de máxima oscilación normalmente se corresponde con la presión arterial media 1.3. El punto por encima de la presión media, en el cual las oscilaciones comienzan rápidamente a aumentar en amplitud se corresponde con la presión sistólica. El punto en que esta variación de las oscilaciones disminuye de forma más abrupta, se corresponde con la presión diastólica. Este método es de gran utilidad para la medicina clínica hace muchos años, sin embargo, tiene la contrapartida de que se pierde el detalle de la forma de onda de la curva de presión, lo cual es solo posible apreciar utilizando un sistema de adquisición continuo con una tasa de muestreo adecuada.



*La curva descendente corresponde a la presión de inflado de la almohadilla, mientras que la curva pulsátil inferior corresponde a la diferencia entre la presión arterial y la ejercida por la almohadilla. Esta última es la medición que se realiza con el manómetro manual. Con ayuda de un estetoscopio, se pueden detectar los ruidos correspondientes a la presión sistólica y diastólica.*

FIGURA 1.3: Curva de inflado de la almohadilla de método oscilométrico vs. presión diferencial y ruidos detectados.

Es claro que la medición de la onda continua aporta muchos más datos que la medición de puntos característicos de la curva. A partir de la onda completa, se pueden calcular diferentes parámetros que permiten modelizar el árbol arterial y predecir enfermedades como hipertensión, arteriosclerosis, rigidez arterial, etc [8] [9].

### 1.1.3. Medicion ambulatoria de presión arterial

Una de las técnicas para medición continua de presión es la medición indirecta a partir del VOP. El método más aceptado para medir VOP en pacientes consiste en el registro simultáneo de señales de presión en carótida y femoral. Conociendo la distancia y el desfase temporal entre estas mediciones de presión, se puede estimar la VOP como su cociente. La medición de VOP ha ingresado en las últimas guías de recomendación europeas como indicadora (Clase IIa; Nivel de evidencia B) para la evaluación subclínica de daño en órganos en pacientes hipertensos. El interés clínico en esta medición ha llevado al desarrollo de numerosos dispositivos para medir VOP y cuyas características y limitaciones están siendo analizadas actualmente para hallar un estándar internacional [10] Así como la MAPA braquial ha impulsado el desarrollo de dispositivos de registro y análisis cada vez más sofisticados, la posibilidad de realizar un registro ambulatorio de VOP genera nuevos campos de investigación, así como la necesidad de que existan nuevos dispositivos [11].

Existen otros metodos de estimacion de la MAPA, pero se generan numerosas controversias debido a la complejidad matemática de las estimaciones y al uso de modelos matemáticos arteriales unificados que suponen ser válidos para todos los pacientes. La presión arterial humana responde a una curva que tiene un pico y un valle, y una forma de onda característica. De esta curva pueden calcularse toda una serie de parámetros que modelizan el arbol arterial. Sin embargo, el método más utilizado en la medicina clínica para estimar la presión arterial, el método oscilométrico, solo toma dos valores característicos de esta curva, el máximo y el mínimo, denominados presion sistólica y presión diastólica.

## 1.2. Motivación y Aplicaciones del equipo

En este contexto, el estudio ambulatorio de la VOP en animales grandes podría brindar nuevas oportunidades para validar diferentes algoritmos de cálculo. La adquisición invasiva de señales de presión en dos sitios alejados del sistema arterial y a una distancia conocida, permite mejorar el conocimiento actual sobre la VOP en distintas condiciones del animal.

Las experiencias realizadas con este equipo permitirán profundizar la investigación sobre la medición indirecta de presión ambulatoria a partir de la medición de VOP y las técnicas para su cálculo. La aplicación de esta técnica para la MAPA permitirá desarrollar a futuro equipos portátiles que puedan medir de manera continua la presión de un paciente, con más información, mejor resolución y más comodidad.

Las experiencias de medición de VOP se realizan sobre animales grandes conscientes, como ovejas o cerdos. Estos animales se encuentran previamente instrumentados con sensores de presión intraarteriales en forma crónica y tienen un prolongado período de adaptación a la vida en un corral de un laboratorio de investigación y al trato con los veterinarios. En la figura 1.4 puede verse una fotografía de una oveja durante una experiencia real con un equipamiento antiguo. La correcta adaptación del animal a la vida en el corral del laboratorio es muy importante porque la presión arterial se ve severamente afectada por la comodidad



y bienestar del animal. Por ejemplo, algunas de las líneas de investigación estudian justamente la diferencia de los valores medios de presión entre el período de vigilia y de sueño. Este tipo de experiencias sobre un animal en estado de alerta se hace totalmente imposible.



FIGURA 1.4: Oveja instrumentada con un equipamiento antiguo con interfaz cableada.

El equipamiento de medición debe ir sobre el animal para evitar que existan cables que limiten el movimiento y dificulten la vida diaria del mismo. Además, luego de un período de adaptación, el equipo debe ser imperceptible para el animal. Esto se logra con un tamaño reducido, un sistema de carga cómoda, como una pequeña mochila, bajo peso, ausencia de ruido, etc. Durante toda la experiencia de medición, el equipo debe estar sobre el animal, y, en lo posible, solo debe acercarse personal de veterinaria para tareas de higiene y alimentación. El operador del equipo debería solamente acercarse al animal para la instalación del equipo, y para retirarlo al finalizar la experiencia. El resto del tiempo, el operador debe tener una interacción mínima con el animal.

### 1.3. Objetivos y alcance

El proyecto incluyó el desarrollo completo de un dispositivo de medición y adquisición de dos sensores de presión intraarteriales para ser utilizado en animales grandes, junto con su documentación técnica y manual de usuario.

Las señales además se pueden visualizar en tiempo real para realizar algún eventual ajuste sobre el animal instrumentado antes de comenzar la experiencia. Este software de visualización no se incluye entre los alcances del proyecto, por lo que se utiliza solamente una interfaz en desarrollo a modo de prueba.

El equipo se diseñó para ser portátil, alimentado por batería, con una autonomía aproximada de 24 horas. A la vez, se hizo énfasis en lograr un bajo peso que no moleste al animal durante la experiencia. La interfaz de carga es a través de la misma conexión USB.

El desarrollo del proyecto no incluye la fabricación del gabinete final ni del soporte para ser llevado por el animal. Tampoco se incluye el software de la terminal de configuración, solamente una versión preliminar para pruebas.

## Capítulo 2

# Introducción Específica

En este capítulo se hará énfasis en las características de la experiencia en la que se utilizará el equipo y las condiciones en que surgen cada uno de los requerimientos de diseño.

### 2.1. Detalle de las necesidades

Este proyecto se destaca principalmente por ser un sistema de adquisición diseñado especialmente para realizar experiencias en animales grandes sin alterar su comportamiento durante lapsos prolongados, y de esta manera no incidir sobre los valores de tensión arterial medidos. Para ello, el desarrollo se enfocó en lograr un equipo de pequeño tamaño y bajo peso que cuente con una interfaz de usuario inalámbrica que permita configurar el equipo y comprobar esporádicamente su funcionamiento evitando que el operador se acerque al animal.

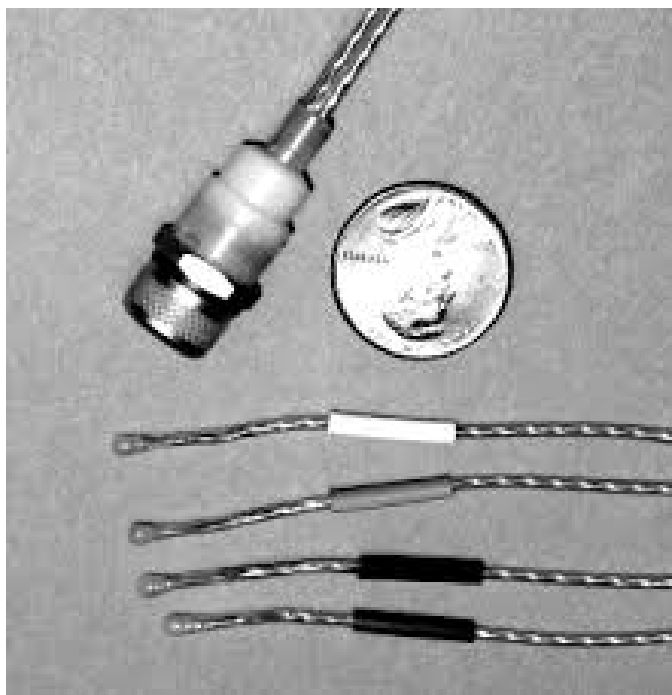
El equipo diseñado digitaliza las señales analógicas y las guarda en una memoria flash de almacenamiento masivo (SD) durante períodos de tiempo prolongados, de aproximadamente 24hs. Puede configurarse y controlarse desde una terminal remota a través de una interfaz comunicada por BLE.

Las experiencias de medición consisten básicamente en colocar el equipo en una especie de pequeña mochila que se ajusta sobre el animal instrumentado con sensores de presión intra-arteriales. Una vez instalado, el equipo digitaliza y adquiere las señales provenientes de los dos sensores durante un período de 24hs. Esto da un margen de tiempo para que el animal se tranquilice luego de la instalación del dispositivo, normalice su comportamiento, y pase por diferentes fases a lo largo del día: vigilia, sueño, alimentación, interacción con veterinarios, etc.

La presión arterial es un parámetro que es fuertemente dependiente del bienestar del animal, por lo que es importante que la batería tenga autonomía suficiente para toda la experiencia y que no sea necesario un recambio o recarga. Es necesario que sea portátil para que el animal no esté conectado a ningún equipo a través de cables. Como se trata de una experiencia larga, también es importante que el equipo no sea pesado, voluminoso, ruidoso, ni se caliente.

Los sensores utilizados son del tipo strain-gauge. Pueden visualizarse en la imagen 2.1. Se trata de una resistencia que varía su valor de acuerdo a la presión aplicada. El sensor además cuenta con propiedades aptas para ser implantado en un animal. Este sensor tiene un conector propio de la marca Königsberg al que se le conecta un compensador por temperatura calibrado específicamente para cada

sensor, y de allí se utiliza un conector AMP - CPC Series I roscado de 4 terminales dorados para ingresar al equipo [12]. Esta última conexión es muy importante para evitar que se agregue ruido a la señal medida.



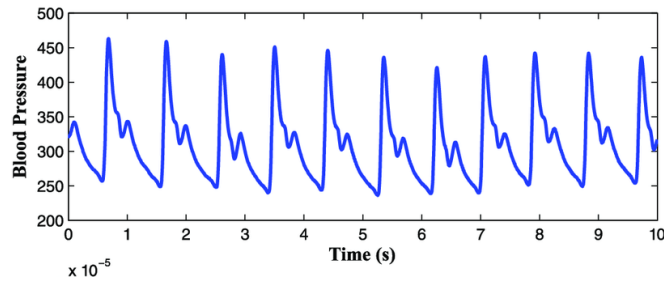
*Sensores empleados para la medición intra arterial. El conector roscado de la figura se conecta a un módulo de compensación por temperatura para luego ingresar al equipo.*

FIGURA 2.1: Conector, cable y sensores Königsberg de presión

La señal de presión arterial tiene un patrón normal con diferentes puntos característicos. Puede observarse en la figura 2.2 un gráfico típico de una señal de presión arterial de un mamífero. La forma de la curva corresponde a una onda estacionaria conformada por una onda incidente pulsátil y su correspondiente onda reflejada. Puede analizarse como un sistema de parámetros distribuidos.

Desde el punto de vista del análisis frecuencial, la señal de presión tiene una frecuencia fundamental apenas superior a 1 Hz, sin embargo, contiene componentes de hasta los 100 Hz, con los que es necesario contar para obtener una buena resolución en el momento de realizar el análisis de la forma de onda. Una frecuencia de muestreo típica utilizada para adquirir este tipo de señales es **500 Hz** o **1 kHz**. Una excursión normal de una señal de presión de un ser humano puede tener un máximo de 140mmHg, por lo que es útil contar con un rango dinámico mayor a este, por ejemplo 200 mmHg. Una adquisición con una resolución del 0.5 %, es decir, de 1 mmHg sería adecuada.

Es muy común que antes de comenzar la experiencia, o durante la experiencia, el animal se mueva o intente tocar los cables que van desde los sensores hasta la mochila donde se encuentra alojado el equipo. Estos movimientos pueden ocasionar cambios en el estado de las conexiones de los sensores, o eventualmente, en el estado del equipo. Es útil, desde el punto de vista del operador del equipo, poder tener algún tipo de información sobre la señal que se está midiendo, de manera de saber si es necesario realizar algún ajuste en las conexiones. Sin embargo, esta observación se debe poder realizar sin molestar al animal, es decir



*La onda de presión arterial de un mamífero corresponde a una señal mayormente periódica con un valor medio variable que puede corresponder a los ciclos diarios y actividad física.*

FIGURA 2.2: Señal típica de presión arterial de un mamífero.

en forma inalámbrica. También es importante compensar una eventual pérdida de ganancia analógica por movimiento de los sensores para no ver disminuída la resolución de la adquisición.

Finalmente, una vez comenzada la medición, al tratarse de una experiencia costosa en cuanto a recursos y tiempo, es importante poder controlar esporádicamente que la adquisición se esté realizando de forma correcta, sin afectar el comportamiento del animal. Por lo tanto es muy útil desde el punto de vista operativo de recursos poder visualizar la señal que se está adquiriendo en forma remota sin influir en el animal, es decir, de forma remota.

## 2.2. Requerimientos

En base a todo lo comentado en la sección anterior, se desprenden los siguientes requerimientos asociados de acuerdo a criterios de energía, tamaño y peso, conectividad y adquisición, con los que fue desarrollado el equipo:

### 1. Necesidades asociadas a la alimentación:

- 1.1. El equipo debe tener una autonomía mayor a 24 horas en el modo de adquisición y almacenamiento, a una dada frecuencia de muestreo.
- 1.2. El equipo debe ser portátil e inalámbrico, por lo tanto, alimentado a batería

### 2. Requerimientos asociados al tamaño físico del equipo:

- 2.1. El peso del equipo debe tener un peso aproximado de 500 gr
- 2.2. El tamaño debe ser aproximadamente de 10cm x 10cm
- 2.3. El equipo no debe calentarse demasiado

### 3. Requerimientos asociados a la conectividad e interfaz de usuario:

- 3.1. Se deben poder visualizar las señales medidas en tiempo real previo a iniciar la experiencia.
- 3.2. Se debe realizar la configuración de la adquisición (ganancia, canales habilitados, frecuencia de muestreo, etc.) desde una terminal Bluetooth.

3.3. Se debe poder acceder a la memoria SD a través de la conexión USB.

4. **Requerimientos asociados a la adquisición y almacenamiento:**

4.1. El equipo debe tener una resolución de 1 mmHg en alguna de las escalas de ganancia.

4.2. Debe poder manejar almacenamiento suficiente para la máxima resolución elegida y frecuencia de muestreo.

4.3. Se debe poder configurar el tamaño de muestra.

## 2.3. Planificación

En la siguiente figura 2.3 puede observarse la planificación original del trabajo en un diagrama de Gantt.

En este diagrama puede verse la organización del trabajo separado en los siguientes módulos:

1. **General:** Documentación inicial del proyecto, selección de rtos y microcontrolador, documentación de memoria técnica, manuales de usuario y técnico, presentaciones, compras de componentes, pedido de servicios, etc.
2. **Drivers y módulos:** driver del ADC, tests de adquisición, driver de almacenamiento, manejo del DMA, interfaz de prueba para configuración, programación de drivers de periféricos, etc.
3. **Batería, carga y autonomía:** medición de consumo, elección de batería, circuito de carga USB, circuito de regulación.
4. **Módulos analógicos:** acondicionamiento analógico de la señal, interconexión con módulo digital.
5. **Diseño de PCB:** integración de módulos, esquemáticos, diseño de PCB, fabricación, soldadura, etc.
6. **Interfaz de Usuario:** Rutinas de menú, protocolo de comunicación, configuración, fuentes, imágenes, testing unitarios, testing de sistema, etc.

En los capítulos 3 y 4 se presenta una descripción de las tareas realizadas y los resultados obtenidos.

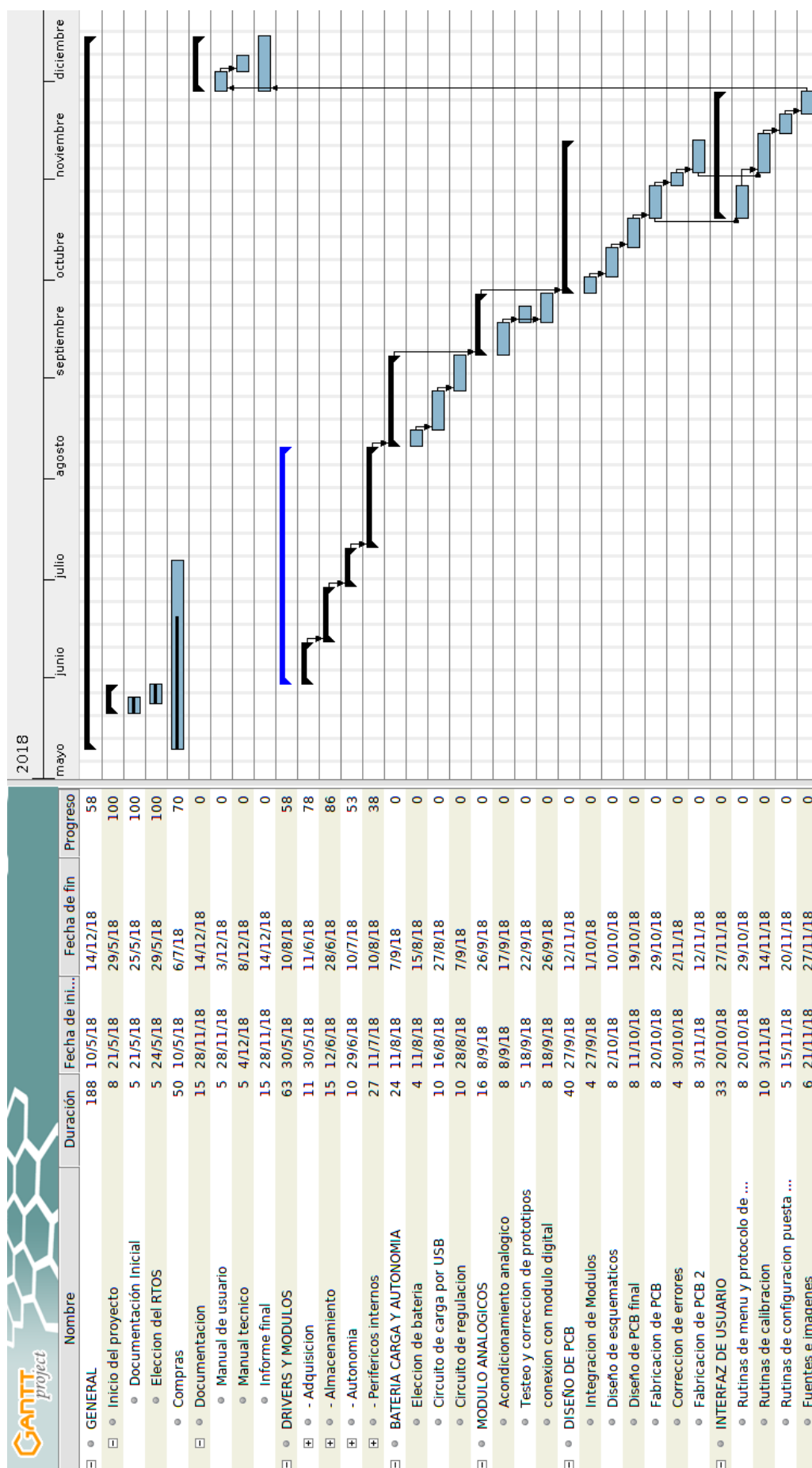


FIGURA 2.3: Diagrama de gantt - Primer parte





## Capítulo 3

# Diseño e Implementación

Este capítulo se centra en el proceso de desarrollo del dispositivo, tanto el firmware como hardware. Se exponen todas las fases de avance a partir del modelo en V, que es el modelo de construcción elegido.

### 3.1. Implementación del firmware

#### 3.1.1. Resumen del sistema

El equipo digitaliza señales analógicas provenientes de dos sensores de presión intrarterial del tipo strain-gauge y almacena las señales por períodos de alrededor de 24 hs. Estas señales adquiridas se guardan en una memoria de tipo flash (memoria micro SD), y pueden descargarse a una PC a través de una interfaz USB. Desde la PC se accede a los archivos guardados como un medio de almacenamiento masivo, y se pueden visualizar en cualquier software que permita procesar un archivo de tipo ".csv".

Previo a cada experiencia, el operador configura el dispositivo desde una terminal Bluetooth (una tablet o una PC, por ejemplo) a través de un protocolo de comunicación que se desarrolló para este dispositivo. A través de esta interfaz se configuran parámetros como la frecuencia de muestreo, la cantidad de canales a usar, ganancia del amplificador programable, seteo de hora y fecha, y finalmente se da inicio a la medición.

El equipo consiste en un sistema embebido portátil basado en un microcontrolador ARM de 32 bits, Cortex M3: LPC1769, un conversor analógico digital de alta resolución (ADS1292), módulos de comunicación (serie, BLE, USB), almacenamiento masivo (microSD) y módulo de regulación de energía y carga de batería. Puede verse un diagrama en bloques en la Figura 3.1.

#### 3.1.2. Modelo de desarrollo del firmware

El proceso elegido para el desarrollo del firmware es el modelo en V. Se trata de un modelo descripto en términos de un proceso de construcción descendente ("Software Development Life Cycle", SDLC) y un proceso de verificación y validación ascendente ("Software Test Life Cycle", STLC) [13]. El punto que une estos dos procesos es la implementación del código. Durante el proceso descendente se descomponen y clarifican las necesidades del cliente, dando lugar a los requisitos

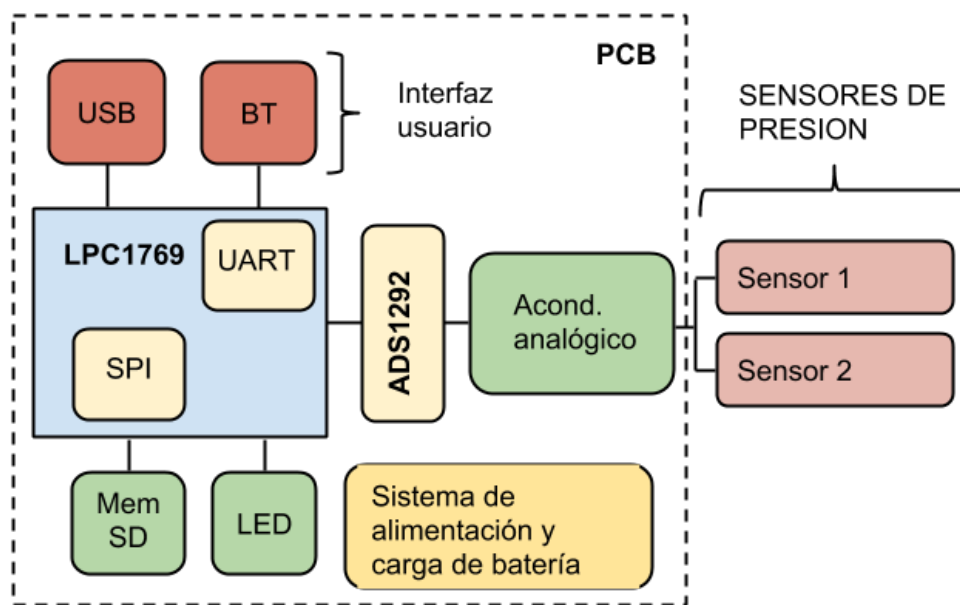


FIGURA 3.1: Diagrama en bloques del equipo.

específicos, diseño de la arquitectura en términos generales y diseño detallado de cada uno de los bloques de código. Luego, cada uno de estos niveles de la fase de construcción se asocia con un nivel de la fase de pruebas, donde se realizan las pruebas unitarias, pruebas de integración y pruebas de sistema. Gráficamente puede visualizarse como una V, de acuerdo a la Figura 3.2.

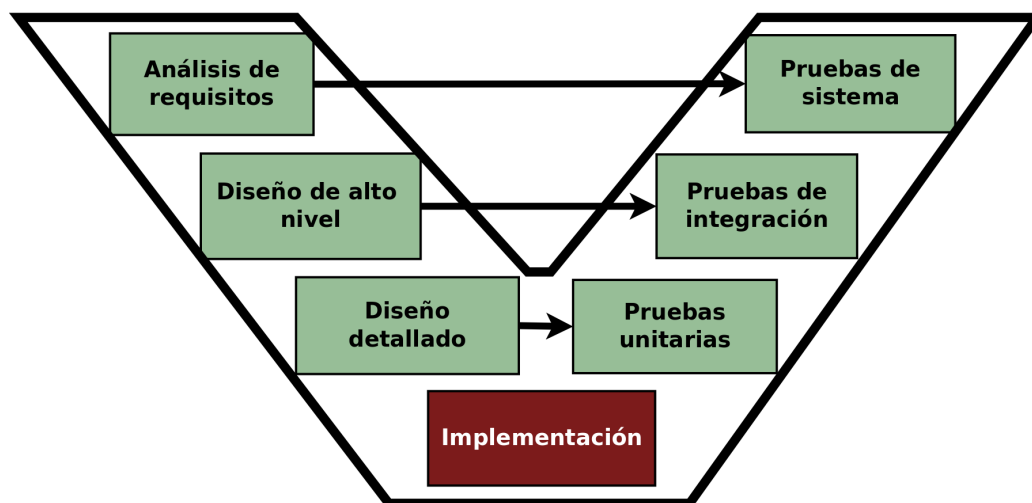


FIGURA 3.2: Modelo de desarrollo y validación en V

En las subsecciones 3.1.3, 3.1.4 y 3.1.5 se pone en relieve cada una de estas instancias del proceso de desarrollo.

### 3.1.3. Análisis de los requerimientos

A partir de un análisis de los requerimientos explicitados en la sección 2.2 se puede determinar que los Requerimientos 1 y 2 están estrictamente asociados al

hardware. Alguno de estos requerimientos se ligan al firmware, como por ejemplo el R1.1, ya que el firmware es el responsable de la entrada a modos de bajo consumo o la administración de energía dentro del microcontrolador. Sin embargo, la mayor carga de responsabilidad de cumplimiento de estos requerimientos están volcados sobre el hardware. Los requerimientos 3 y 4 son muy cercanos al desarrollo del firmware del equipo. Solamente el requerimiento 4.2 está relacionado al tamaño de la memoria utilizada, que es puramente una cuestión de hardware.

Una vez iniciado el proceso de desarrollo se notó que la lista de requerimientos del capítulo 2 no es exhaustiva y da por supuesto las funcionalidades básicas del equipo. Se procedió entonces a elaborar una serie de requerimientos más específicos sobre el funcionamiento detallado de la operación del usuario y los parámetros que es posible configurar, tal que permita luego asociar requisitos puntuales a los casos de uso del equipo.

A continuación se plantean requerimientos adicionales a los presentados en la sección 2.2:

#### **Requerimientos adicionales :**

##### **5. Requerimientos estructurales:**

- 5.1. El equipo debe poder adquirir señales analógicas de hasta dos canales de entrada.
- 5.2. El desfase entre las señales adquiridas debe ser menor a diez veces la frecuencia de muestreo.

##### **6. Requerimientos operativos:**

- 6.1. Al estar inactivo, el equipo debe entrar en modo configuración para poder ponerse en hora y mantenerla mientras tenga batería.
- 6.2. Al estar en configuración, se debe poder chequear el correcto funcionamiento del hardware (SD, conversor AD, etc).
- 6.3. Si el equipo esta inactivo, al enviar un comando desde la terminal, se comienza a adquirir y enviar vía Bluetooth (esta señal puede estar decimada y con menor resolución) con la configuración previamente seleccionada.
- 6.4. Si el equipo está adquiriendo y enviando vía Bluetooth, se puede enviar una orden para que el equipo finalice y vuelva al modo inactivo.
- 6.5. Si el equipo está adquiriendo y enviando vía Bluetooth, se puede enviar una orden para que el equipo comience a almacenar en la memoria SD.
- 6.6. Si se esta adquiriendo, enviando señal vía Bluetooth y almacenando en la SD, se debe poder enviar una orden para que el equipo deje de hacerlo y vuelva al modo inactivo.
- 6.7. Si se esta adquiriendo, almacenando y enviando por Bluetooth, se debe poder enviar una orden para que el equipo deje de enviar la señal por Bluetooth y continúe la experiencia en forma silenciosa.

- 6.8. Si el equipo está adquiriendo y almacenando en forma silenciosa, se debe poder enviar un comando para que vuelva a enviar la señal por Bluetooth sin interrumpir ni alterar la adquisición y almacenamiento.
- 6.9. Si el equipo está adquiriendo, enviando señal via Bluetooth y almacenando, se puede enviar un comando para que el equipo deje almacenar.
- 6.10. Si el equipo esta inactivo, se puede requerir el envío de una señal patrón para chequeo del canal de comunicación.
- 6.11. Si el equipo esta enviando la señal patrón para chequeo de canal de comunicación, se puede enviar un comando para que el equipo vuelva a modo inactivo.
- 6.12. Se debe almacenar por cada registro la hora, los canales activados y el nivel de batería.
- 6.13. Cada vez que se comienza un nuevo almacenamiento, se genera un archivo nuevo con nombre auto numerado.
- 6.14. Cada vez que comienza o finaliza un almacenamiento, se registra en un archivo que debe contener el nombre del archivo y la fecha y hora de inicio y fin de todas las experiencias.
- 6.15. Cada uno de estos modos debe estar señalizado por uno o mas leds para comprobar visualmente el funcionamiento.

## 7. Requerimientos de configuración

- 7.1. El equipo debe tener una configuración por default y la posibilidad de modificarla mientras no se esta adquiriendo.
- 7.2. El equipo debe poder volver a la configuración de fábrica.
- 7.3. El equipo debe guardar la última configuración y utilizarla.
- 7.4. Se debe poder elegir la frecuencia de muestreo dentro de algunas opciones.
- 7.5. Se debe poder elegir si activar uno o los dos canales.
- 7.6. Se debe poder elegir la ganancia del amplificador.
- 7.7. El equipo se debe poder calibrar con dos puntos de presión conocidos.

## Casos de Uso:

De toda la lista completa de requerimientos surgen los casos de uso con operaciones básicas del usuario. A continuación se muestran los diferentes casos de uso que se desprenden de los requerimientos:

### Caso de uso CU0001:

- 1. Nombre: CU0001
  - 1.1. Descripción: Autonomía
  - 1.2. Actor Principal: Usuario
  - 1.3. Disparador: Encendido del equipo

2. Flujo de eventos
  - 2.1. Flujo básico: el usuario enciende el equipo, realiza la configuración y comienza la experiencia. Luego de 24 hs finaliza la experiencia desde una terminal.
  - 2.2. Flujo alternativo: Si la batería no tiene carga total, es responsabilidad del usuario. Según la configuración se puede calcular autonomía aproximada.
3. Requerimientos especiales: conexión Bluetooth
4. Pre-condiciones: cargar la batería al 100 % antes de comenzar a usar el equipo.
5. Post-Condiciones: Finalizar correctamente la experiencia.

Caso de uso CU0002:

1. Nombre: CU0002
  - 1.1. Descripción: Configuración
  - 1.2. Actor Principal: Usuario
  - 1.3. Disparador: Ingreso a configuración desde la terminal
2. Flujo de eventos
  - 2.1. Flujo básico: el usuario desde la terminal ingresa al modo configuración, modifica los parámetros que necesita o calibra un canal, y vuelve a salir del modo configuración.
  - 2.2. Flujo alternativo:
    - Si el equipo no está sincronizado, indica error y da la oportunidad de reiniciar.
    - Si no se puede abrir la última configuración, debe abrir la configuración de fábrica y mostrar error.
    - La nueva configuración elegida se guarda y se comprueba al comenzar la experiencia. Hasta ese momento no se comprueba si es correcta.
    - Si se comprueba algún error de hardware, se debe enviar un mensaje de reparación.
3. Requerimientos especiales: conexión Bluetooth
4. Pre-condiciones: equipo en modo inactivo
5. Post-Condiciones: salir correctamente del modo configuración.

Caso de uso CU0003:

1. Nombre: CU0003
  - 1.1. Descripción: Experiencia
  - 1.2. Actor Principal: Usuario
  - 1.3. Disparador: Ingreso a experiencia desde la terminal

## 2. Flujo de eventos

2.1. Flujo básico: el usuario desde la terminal ingresa al modo de adquisición y visualización, comprueba el correcto funcionamiento de los sensores habilitados observando la señal adquirida. Luego ingresa al modo almacenamiento y comprueba que no haya error en el almacenamiento de datos. Luego selecciona dejar de enviar para que la experiencia continúe en modo silencioso. Al finalizar la experiencia, el usuario envía la orden para terminar la adquisición. El usuario luego comprueba que la adquisición fue completa.

### 2.2. Flujo alternativo:

- Si los sensores no muestran una señal adecuada, se debe salir de la adquisición y chequear conectores.
- Si los sensores no muestran ninguna señal, se debe comprobar el canal.
- Si hay error al comenzar a almacenar, se detiene la adquisición para comprobar el funcionamiento de la tarjeta SD.
- Si se necesita ver las señales por una eventual desconexión, se puede conectar mediante Bluetooth para visualizar las señales y luego salir.
- Si no se puede volver a conectar con la terminal, se debe acceder físicamente a conectar el equipo a una PC.
- Si la batería está en un nivel crítico, el equipo debe guardar el archivo y finalizar la experiencia. El usuario se notifica de esto al intentar conectarse.

## 3. Requerimientos especiales: conexión Bluetooth

## 4. Pre-condiciones: equipo en modo inactivo

## 5. Post-Condiciones: salir correctamente de la experiencia.

### Caso de uso CU0004:

#### 1. Nombre: CU0004

##### 1.1. Descripción: Experiencia

##### 1.2. Actor Principal: Usuario

##### 1.3. Disparador: Ingreso a experiencia desde la terminal

## 2. Flujo de eventos

2.1. Flujo básico: el usuario desde la terminal ingresa al modo de adquisición y visualización, comprueba el correcto funcionamiento de los sensores habilitados observando la señal adquirida. Luego ingresa al modo almacenamiento y comprueba que no haya error en el almacenamiento de datos. Luego selecciona dejar de enviar para que la experiencia continúe en modo silencioso. Al finalizar la experiencia, el usuario envía la orden para terminar la adquisición. El usuario luego comprueba que la adquisición fue completa.

## 2.2. Flujo alternativo:

- Si los sensores no muestran una señal adecuada, se debe salir de la adquisición y chequear conectores.
- Si los sensores no muestran ninguna señal, se debe comprobar el canal.
- Si hay error al comenzar a almacenar, se detiene la adquisición para comprobar el funcionamiento de la tarjeta SD.
- Si se necesita ver las señales por una eventual desconexión, se puede conectar mediante Bluetooth para visualizar las señales y luego salir.
- Si no se puede volver a conectar con la terminal, se debe acceder físicamente a conectar el equipo a una PC.
- Si la batería está en un nivel crítico, el equipo debe guardar el archivo y finalizar la experiencia. El usuario se notifica de esto al intentar conectarse.

## 3. Requerimientos especiales: conexión Bluetooth

## 4. Pre-condiciones: equipo en modo inactivo

## 5. Post-Condiciones: salir correctamente de la experiencia.

Caso de uso CU0005:

## 1. Nombre: CU0005

## 1.1. Descripción: Descarga de datos y carga de batería.

## 1.2. Actor Principal: Usuario

## 1.3. Disparador: Conectar el equipo por USB a PC

## 2. Flujo de eventos

## 2.1. Flujo básico: el usuario conecta el equipo a la PC y comprueba que se reconozca como un medio de almacenamiento masivo. Luego descarga los archivos que sean necesarios y deja el equipo conectado mientras necesite cargar la batería. Finalmente, luego de finalizadas todas las transferencias de datos, desconecta el equipo del USB.

## 2.2. Flujo alternativo:

- Si la PC no reconoce al equipo como medio de almacenamiento masivo, se debe comprobar la tarjeta SD.
- Si el equipo no carga la batería, se debe hacer una revisión técnica del equipo y de las baterías.

## 3. Requerimientos especiales: conexión USB

## 4. Pre-condiciones: equipo en modo inactivo

## 5. Post-Condiciones: desconectar el USB cuando no haya ninguna transferencia activa y cuando haya terminado de cargar la batería. El nivel de batería se puede comprobar desde la terminal entrando al modo configuración.

A partir de los requerimientos y casos de uso se arma la matriz de trazabilidad, que consiste en una tabla de doble entrada que asocia cada requerimiento con uno o más casos de uso. La matriz de trazabilidad se presenta en la Tabla 3.1

TABLA 3.1: Matriz de Trazabilidad

REQ/CU	CU0001	CU0002	CU0003	CU0004	CU0005	Ensayo
R1.1	X					Comprobación post-experiencia
R1.2						Medición de corriente consumida
R2.1						Conexión desde terminal
R2.2						Pesaje del equipo
R2.3						Medición del equipo
R3.1			X			Medición temperatura <40°
R3.2			X			Test MDE
R3.3		X				Test MDE
R3.4					X	Comprobación post-experiencia
R4.1			X			Test conexión PC
R4.2			X		X	Test Calibración
R4.3					X	Test conexión PC
R5.1			X			Comprobación post-experiencia
R5.2			X			Comprobación post-experiencia
R5.3						Comprobación post-experiencia
R6.1			X			Medición de corriente consumida
R6.2			X			Test MDE
R6.3			X			Test MDE
R6.4			X			Test MDE
R6.5			X			Test MDE
R6.6			X			Test MDE
R6.7			X			Test MDE
R6.8			X			Test MDE
R6.9			X			Test MDE
R6.10				X		Test MDE
R6.11				X		Test MDE
R6.12			X		X	Comprobación post-experiencia
R6.13			X		X	Comprobación post-experiencia
R6.14					X	Comprobación post-experiencia
R6.15			X			Comprobación post-experiencia
R7.1		X				Comprobación visual
R7.2		X				Comprobación post-experiencia
R7.3		X				Comprobación post-experiencia
R7.4		X				Comprobación post-experiencia
R7.5		X				Comprobación post-experiencia
R7.6		X				Comprobación post-experiencia

### 3.1.4. Diseño de alto nivel

Una vez especificados y analizados todos los requerimientos, se da a lugar al diseño del sistema en alto nivel, es decir, analizar los subsistemas que lo conforman y su relación entre sí. Los subsistemas que entran en juego se presentan en la Tabla 3.2, son los bloques básicos desde de los cuales se conforma el dispositivo. Es necesario recurrir a patrones de diseño [14] para articular entre sí los diferentes módulos.



TABLA 3.2: Subsistemas del equipo

Subsistema	Responsabilidad	Entrada	Salida
Sensores	Sensar señal de presión	Señal física de presión	Señal analógica
Acondicionamiento analógico	Filtrar y ajustar ganancia de señal analógica	Señal analógica de presión	Señal analógica de presión acondicionada
ADS1292	Adquirir señal analógica	Señal analógica de presión acondicionada	Señal digital via SPI-Pin Data Ready
SD	Almacenar señal adquirida	Paquete de muestras de señales adquiridas	Archivos csv con muestras y archivo con registro de actividad
Led	Indicadores de estado	Niveles lógicos	Encendido o apagado como indicador
Sistema de alimentación y carga de batería	- Cargar la batería al conectar por USB- Alimentar el sistema al no tener conectado el USB a partir de la batería	Tensión USB o de batería	-Corriente de alimentación de USB - Tensión regulada
USB	- proveer tensión para carga de batería- conectividad a PC via medio de almacenamiento masivo	Conector USB a PC	Visualización de sistemas de archivos interna
Bluetooth	Conectividad con terminal	Comandos vía BT	Comandos y datos vía BT

### Patrones de Diseño:

En primer lugar, la cadena de adquisición, envío de datos y almacenamiento se asocia a un patrón productor/consumidor, donde el conversor analógico digital ADS1292 y ADC interno son los productores de datos, y las salidas por Bluetooth y almacenamiento en la SD son los consumidores. La información producida se guarda en un ping-pong buffer [15] del cual leen las demás tareas. Esto puede visualizarse en la figura 3.3.

En este patrón también participa el esquema de *"proxy pattern"*, que se refiere a un esquema de funcionamiento en el que un elemento de software es el responsable del acceso a un recurso de hardware, y todas las tareas que quieran acceder al recurso lo hacen a través de este elemento. Se utiliza una estructura o clase que encapsule todos los accesos al hardware, sin importar su interfaz física. Este proxy publica los servicios que permiten leer y escribir en el dispositivo, así como inicializar, configurar y apagar el dispositivo. En el diagrama de la figura 3.3 pueden verse en verde todos los elementos de hardware, y con flechas la interfaz física través del cual se comunica.

Los módulos de hardware que se muestran en la figura 3.3 son los siguientes:

- ADS1292: se trata de un conversor analógico digital de alta resolución (24 bits), con dos canales de entrada diferencial simultáneos, de tecnología Sigma-Delta. Incluye internamente un amplificador de ganancia programable (PGA). Puede trabajar a una frecuencia de muestreo máxima de 8kSPS. La interfaz

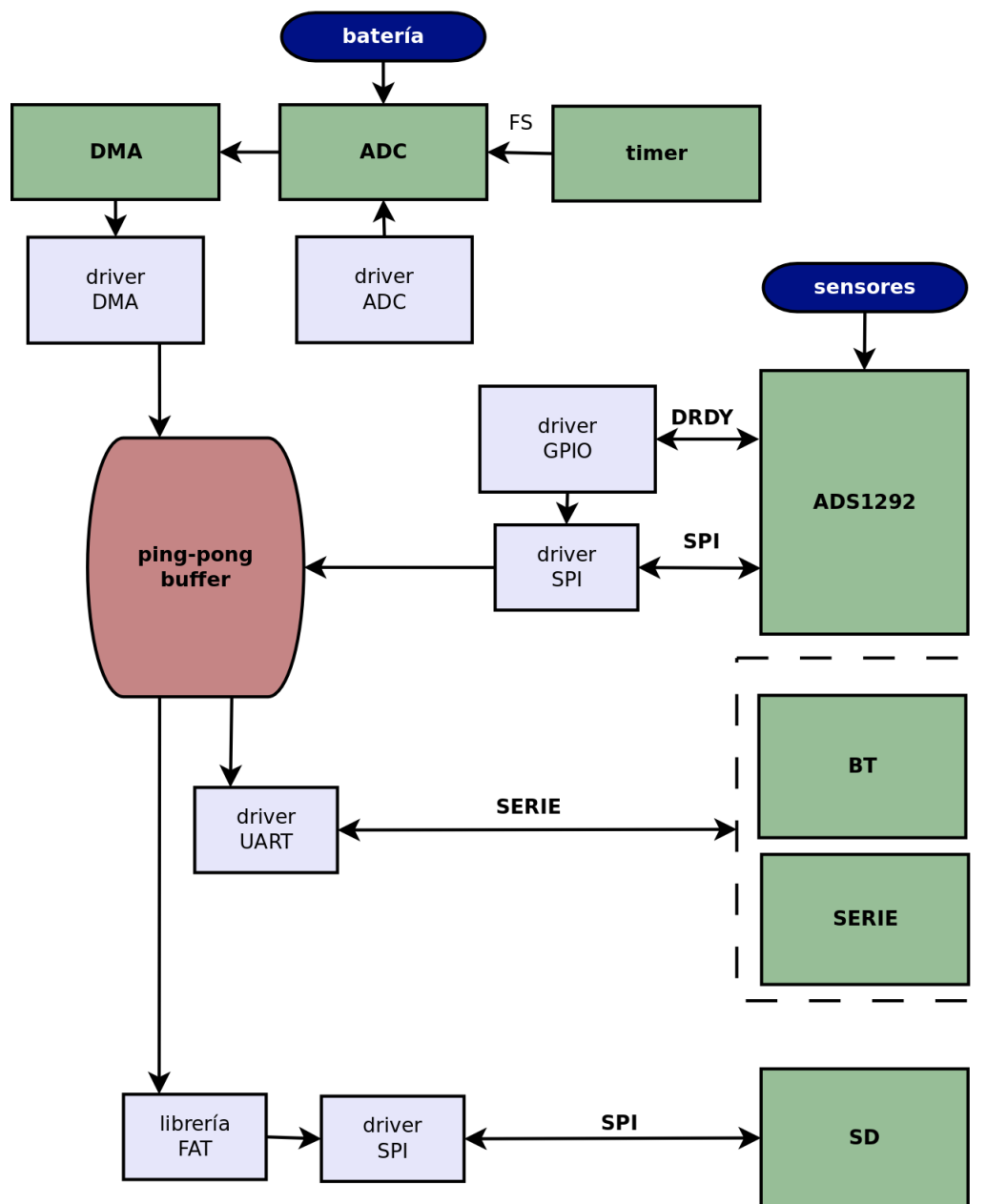


FIGURA 3.3: Diagrama de relaciones entre subsistemas. Patrón productor/consumidor

de comunicación es a través de un bus SPI [16]. El conversor tiene dos modos de operación: puede indicar con el pin DRDY que cuenta con datos a ser leídos y esperar el comando RDATA, o puede estar en el modo “Read Data Continuous” (RDATA\_C) y enviar comandos constantemente. En el diagrama anterior se muestra la conexión a través del pin DRDY para mostrar el caso general y ser compatible con cualquiera de los dos modos de envío de datos.

- Bluetooth: la conexión Bluetooth se realiza a través de un módulo HC-06. Se conecta al microcontrolador a través de un puerto serie y permite la comunicación entre el software de la terminal y la aplicación guardada en el firmware. Cuenta con un pin de habilitación y una salida de STATE para indicar cuando hay vigente una conexión Bluetooth.
- Serie: solamente se utiliza para desarrollo y pruebas de comunicación. Físicamente se trata de un conversor USB-TTL y se maneja a través de un driver de la UART.
- SD: el medio de almacenamiento masivo del dispositivo es una tarjeta microSD. La placa cuenta con un slot de microSD que permite la conexión entre la tarjeta y el microcontrolador a través de un bus SPI, además de un pin que permite detectar cuando hay una tarjeta conectada. La interfaz entre la aplicación y el hardware se hace en dos capas: la aplicación se comunica con el nivel de manejo de archivos (librería FatFs), y esta librería se comunica con las funciones del driver SPI [17], como se visualiza en la figura 3.4.

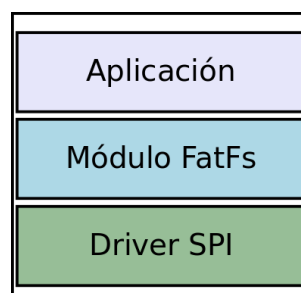


FIGURA 3.4: Capas de software de la librería FatFs

- ADC: se trata de un conversor analógico digital interno al microcontrolador, de 12 bits de resolución, 8 canales multiplexados, de tecnología SAR (aproximaciones sucesivas). Se habilita la conversión a través de la habilitación de un timer que temporiza la frecuencia de muestreo. Los resultados de la conversión del ADC se transfieren a memoria a través del DMA.
- DMA: uno de los canales está configurado en modo “PERIPHERAL TO MEMORY”, realizando las transferencias entre el periférico ADC y un puntero que apunta a un buffer. En cada transferencia el puntero se incrementa hasta llegar al final y generar una interrupción.

### 3.1.5. Diseño Detallado

En base al diseño de alto nivel realizado, se da a lugar al diseño detallado de los elementos del programa como tareas y máquinas de estado.

### Diseño de tareas

El firmware está basado en el sistema operativo de tiempo real freeRTOS [18]. En el diagrama de la figura 3.5 se puede observar un esquema detallado de las tareas involucradas y las relaciones entre ellas.

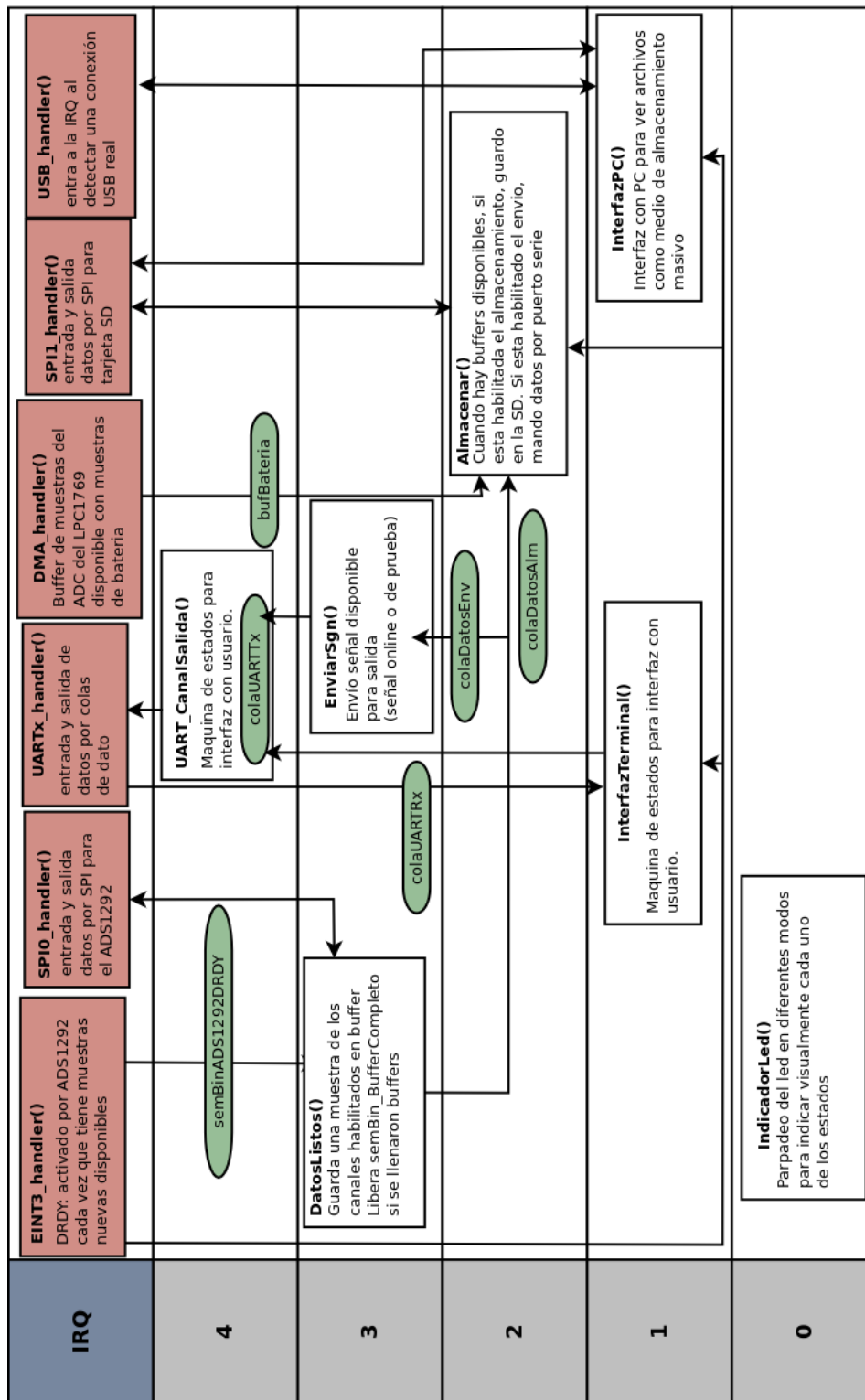


FIGURA 3.5: Diagrama de tareas completo

Cada una de las filas del gráfico 3.1.5 corresponde a un nivel de prioridad, siendo “cero” el nivel de menor prioridad y “cuatro” el de máxima prioridad utilizado. El nivel “IRQ” corresponde a las rutinas de atención de interrupción, que en el sistema freeRTOS tienen una prioridad siempre mayor a cualquiera de las tareas. En verde se pueden ver los mecanismos de comunicación entre tareas, como semáforos o colas de datos. Las flechas que no tienen ninguna indicación corresponden a flags, estados o asignación directa de variables de menor complejidad.

El diagrama de la figura 3.5 ilustra en detalle el funcionamiento del firmware. En el nivel de prioridad 1 se encuentran las tareas de interfaz con el usuario, que son de menor prioridad porque no tienen una exigencia crítica de tiempo. Se explica a continuación el funcionamiento en diferentes casos:

- Al iniciar el equipo, solo está activa la tarea “InterfazTerminal()”, que es la encargada de comunicarse con el usuario remoto respetando el diagrama de estados de la figura 3.6. En este caso solamente intervienen esta tarea, la tarea UART\_CanalSalida(), que es una tarea cercana al hardware que hace la interfaz entre la cola de datos de salida y el registro de la UART, y finalmente la propia rutina de interrupción de la UART que escribe en la cola de datos de entrada cada vez que recibe un mensaje, y envía los datos de salida permitiendo escribirse en bloques hasta del tamaño de la FIFO interna del periférico (16 bytes). En nivel de prioridad cero siempre se ejecuta una tarea que señala con parpadeo de leds según el modo de funcionamiento activo. En este caso, es un parpadeo cada 500ms.

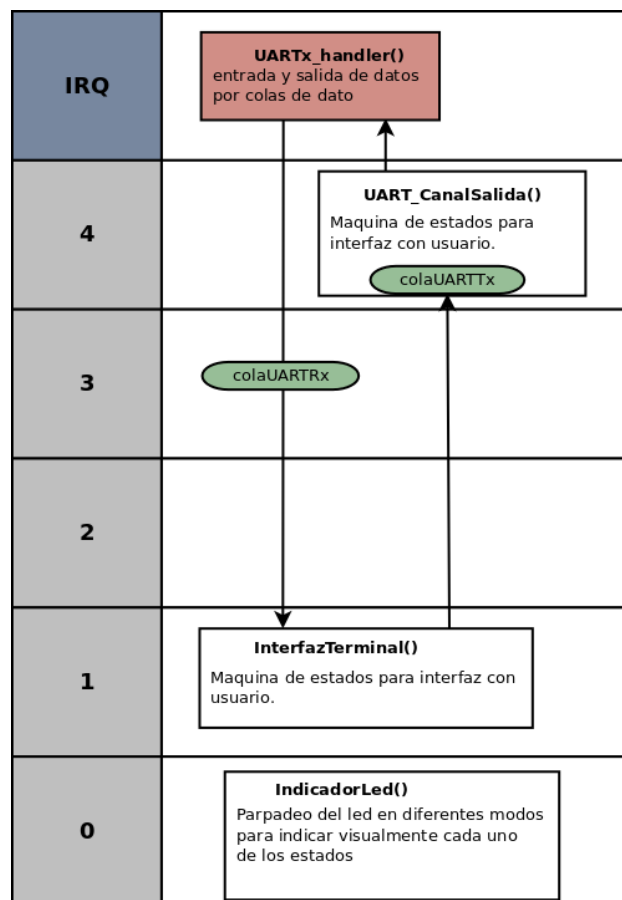


FIGURA 3.6: Diagrama de tareas en modo inactivo

- Cuando el equipo pasa al modo “adquirir+enviar”, se inicializa el hardware de adquisición (ADS1292), que se comunica a través del bus SPI0 y de una entrada de propósito general (GPIO) conectada al pin DRDY. Como se explicó anteriormente, uno de los modos de funcionamiento del ADS1292 es señalar a través del pin DRDY la disponibilidad de datos nuevos para ser leídos. La rutina de interrupción de GPIO “EINT3\_handler()” se activa en cada nivel bajo del pin DRDY, y libera el semáforo “semBinADS1292DRDY”. Por otro lado, esto libera la tarea “DatosListos()” que se encontraba en estado bloqueado en espera de este semáforo, para luego enviar el comando RDATA a través del bus SPI, y recibir finalmente el dato leído para almacenarlo en un buffer.

Al ingresar al modo “adquirir+enviar” se instala una tarea llamada “EnviarSgn()”, que se encuentra bloqueada a través de la cola de datos “colaDatosEnv”. La tarea “DatosListos()” envía un puntero a buffer a través de esta cola de datos cada vez que se completa un buffer de adquisición, de manera que la tarea “EnviarSgn()” también escriba en la cola de datos de salida de la UART, “colaUARTx”.

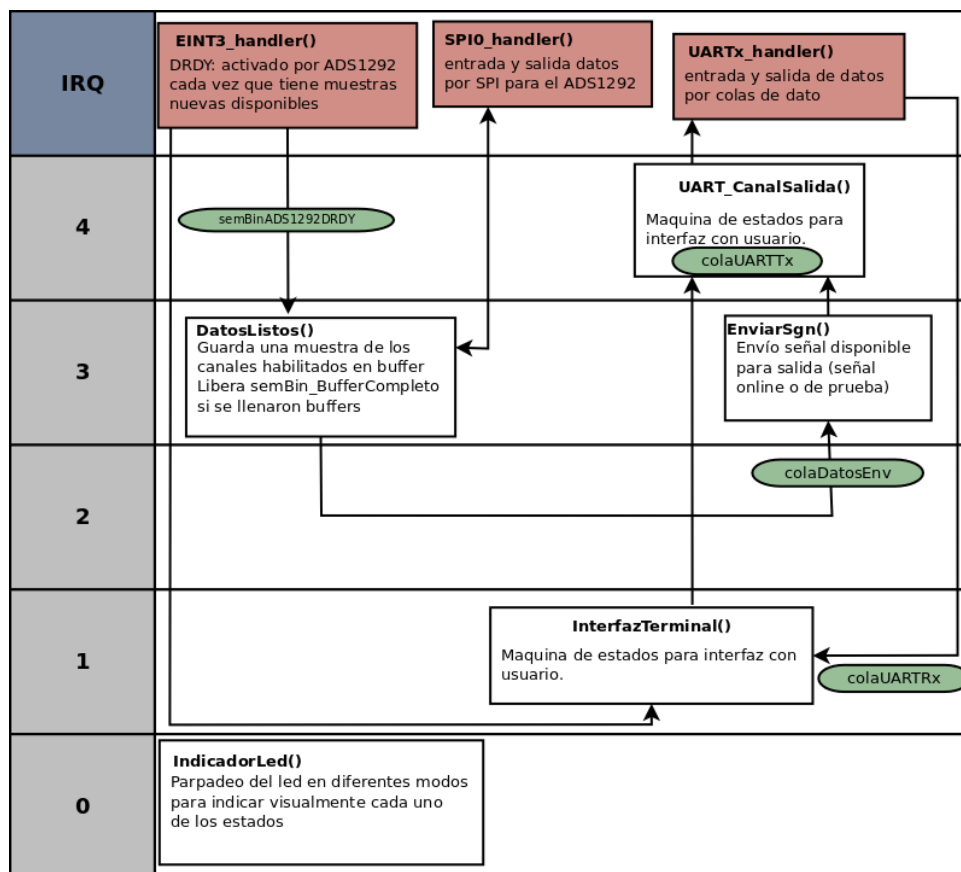


FIGURA 3.7: Diagrama de tareas en modo adquirir y enviar

- Al ingresar al modo “adquirir+enviar+almacenar”, se agrega la tarea “Almacenar()”, que también se encuentra bloqueada a través de una cola de datos llamada “colaDatosAlm”, a través de la cual se envía un puntero a buffer a ser guardado en la memoria SD.

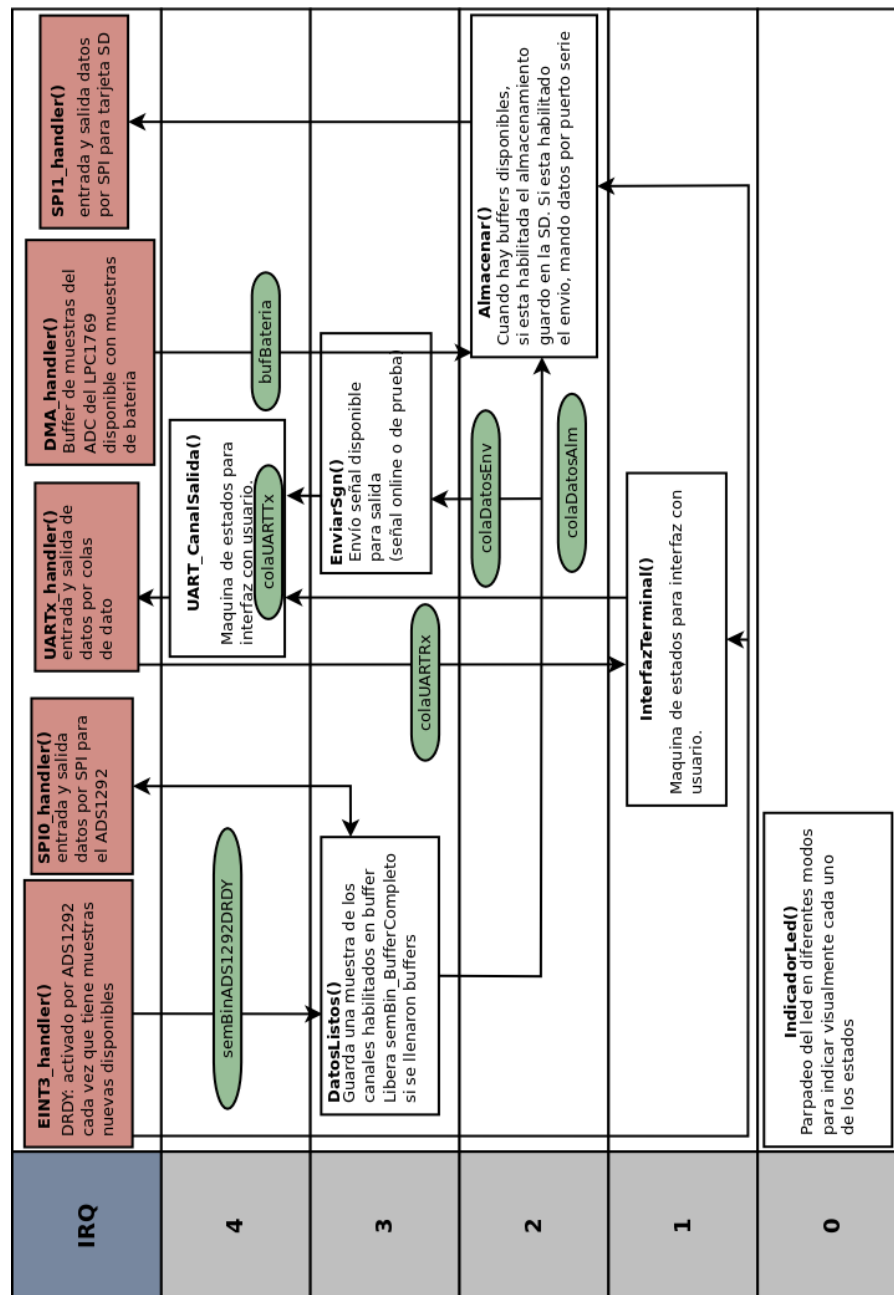


FIGURA 3.8: Diagrama de tareas en modo adquirir, enviar y almacenar

La tarea Almacenar() se comunica a través de un bus SPI directamente con la tarjeta SD. El tamaño del buffer de almacenamiento se definió haciendo pruebas entre el tiempo de llenado del buffer (cantidad de muestras multiplicado por período de muestreo) versus el tiempo de almacenamiento que demora escribir el buffer como archivo en la memoria SD. Este último tiempo no es lineal a la cantidad de muestras, por lo que se hicieron varias pruebas hasta determinar un tamaño óptimo para el peor caso que es la máxima frecuencia de muestreo posible. El tamaño determinado es de 1024 muestras y se comprobó a 500 muestras por segundo con dos canales habilitados. Al almacenar en la SD también se guardan las muestras de medición

de batería. Para ello se hace un promedio de una ráfaga de muestras que se guardan en un buffer “bufBateria” a través del DMA.

- Durante el modo “adquirir+almacenar” se suspende la tarea “EnviarSgn()” y el resto de las tareas continúa igual.

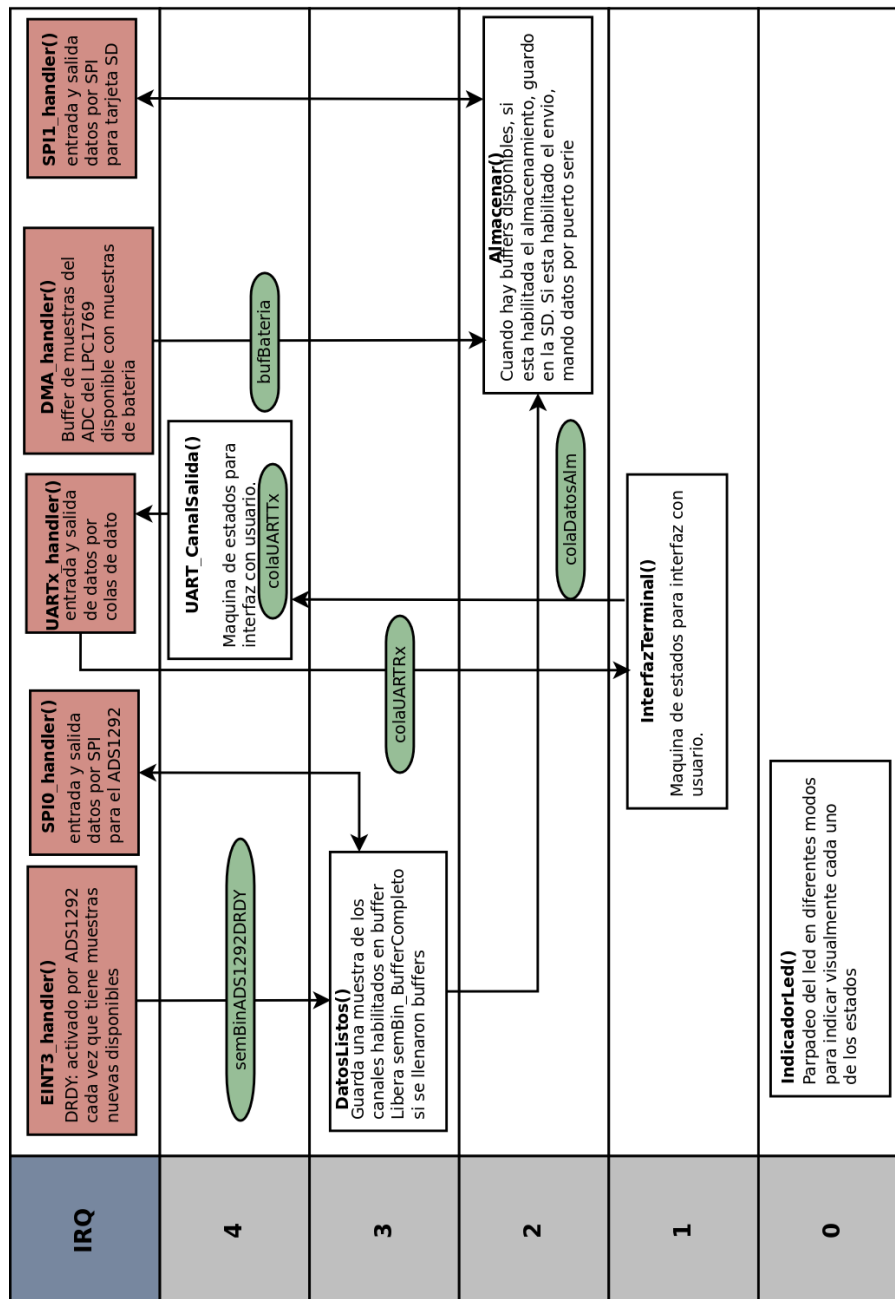


FIGURA 3.9: Diagrama de tareas en modo adquirir y almacenar

El led pasa a realizar dos parpadeos en un segundo y luego permanece apagado el siguiente segundo. Esta es la indicación visual de que el equipo se encuentra correctamente funcionando mientras el usuario no tenga acceso mediante Bluetooth.



- Cuando se conecta el USB a una PC, se activa el handler de la interrupción USB\_Handler y se instala una tarea “InterfazPC()” que espera a que el equipo esté en modo inactivo para luego permitir ver el contenido de la tarjeta de memoria desde la PC como un medio de almacenamiento extraíble. Todas las demás tareas se bloquean mientras el equipo se comunice con la PC: no se puede conectar mediante Bluetooth ni usar ninguna de las funcionalidades anteriores hasta que se desconecte el USB.

Mientras se mantiene el USB conectado a una PC, el led permanece oscilando para indicar carga y se habilita la carga de la batería. Una vez finalizada la carga, el led queda prendido permanentemente hasta desconectar el USB.

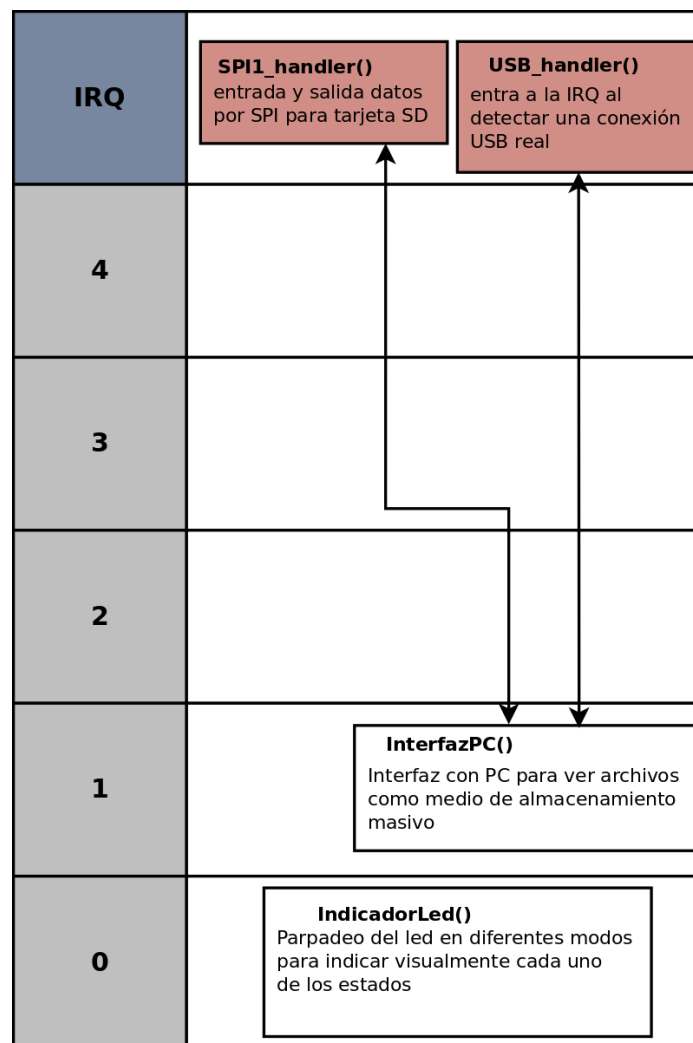


FIGURA 3.10: Diagrama de tareas en modo conexión USB

- Durante el modo “configuración” interviene solamente la comunicación a través de la UART para el caso de modificar hora y fecha. Al momento de calibrar, es necesario activar la medición con el ADS1292, pero en este caso, el buffer adquirido se promedia para obtener solo una muestra como salida. La comunicación con la tarea “DatosListos()” que toma las muestras en el buffer se hace a través de la cola de datos “colaDatosEnv”.

La medición de batería solamente se calibra al momento de grabar el código en la flash, no se ofrece calibración por menú.

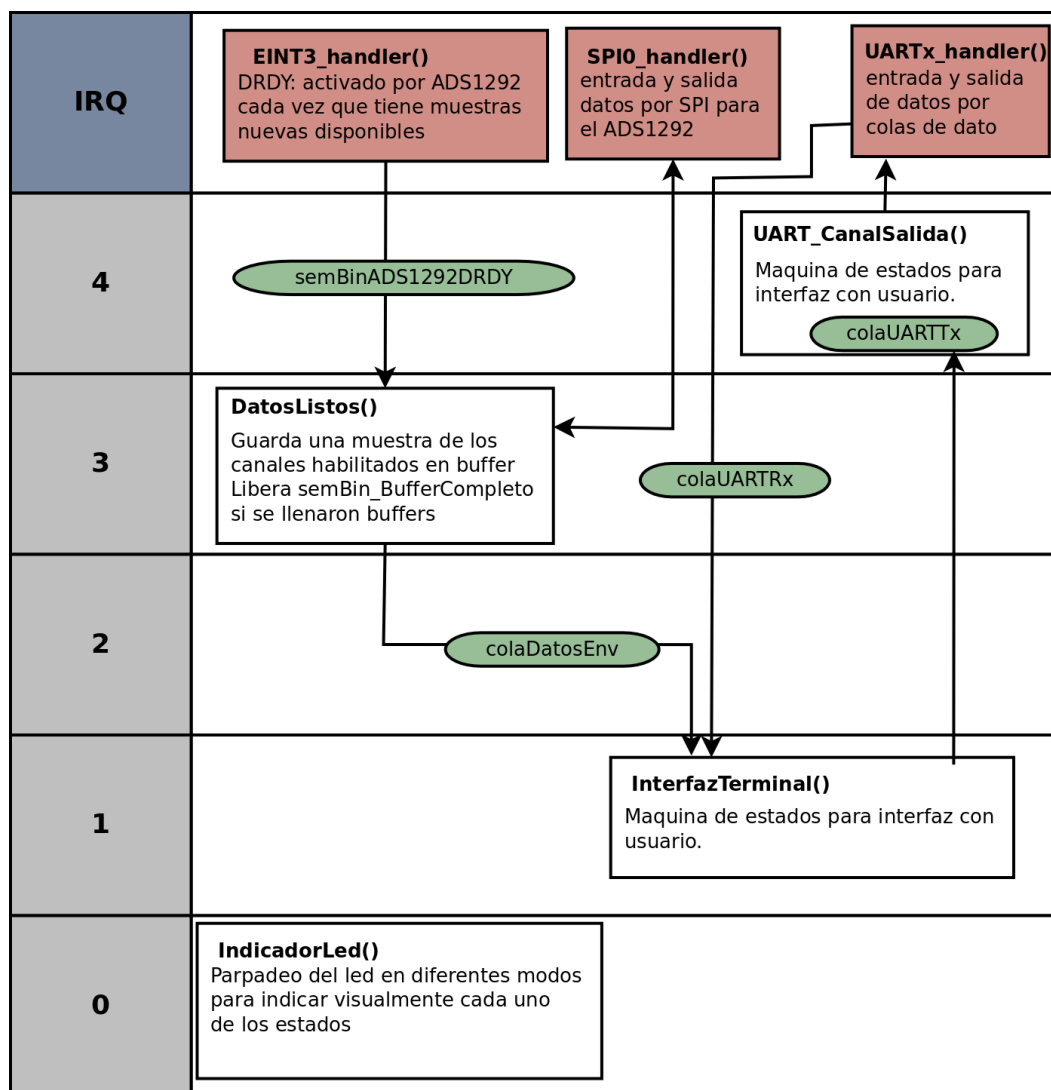


FIGURA 3.11: Diagrama de tareas en modo configuración

### Diseño de máquinas de estado

La tarea InterfazTerminal es la encargada de establecer comunicación con el usuario. Se modela como máquina de estados. En la figura 3.12 solamente el flujo posible del programa, no se muestran en el gráfico los comandos específicos que se envían.

El estado de configuración se trata de un estado compuesto en el cual se puede modificar todos los parámetros de adquisición como frecuencia de muestreo, canales habilitados y ganancia PGA. También se puede cambiar la configuración de fecha y hora y realizar la calibración de un canal. Se modela igualmente como máquina de estados, de acuerdo al diagrama de la figura 3.13.

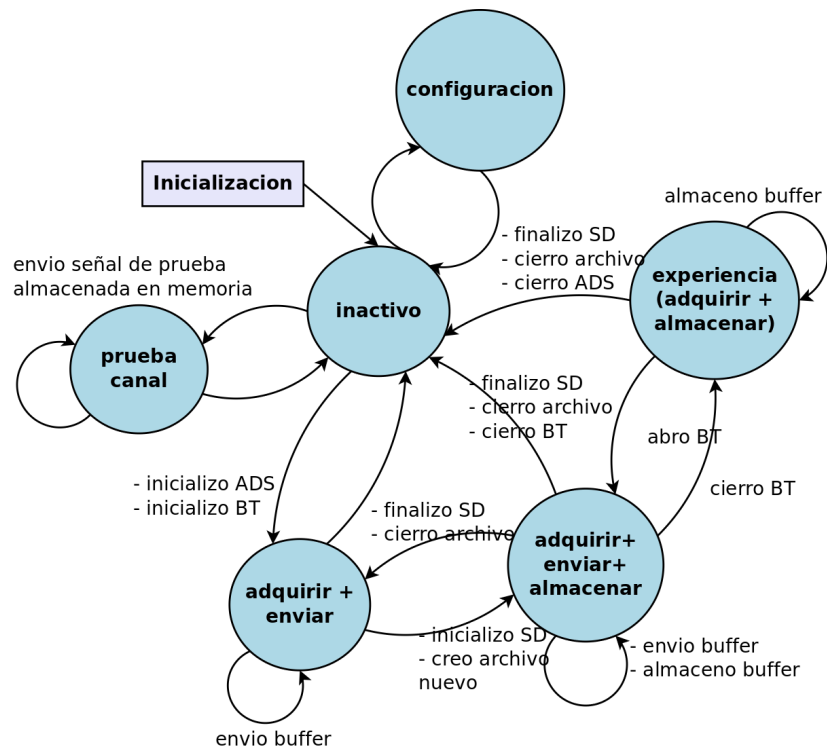


FIGURA 3.12: Máquina de estados de tarea de interfaz de usuario

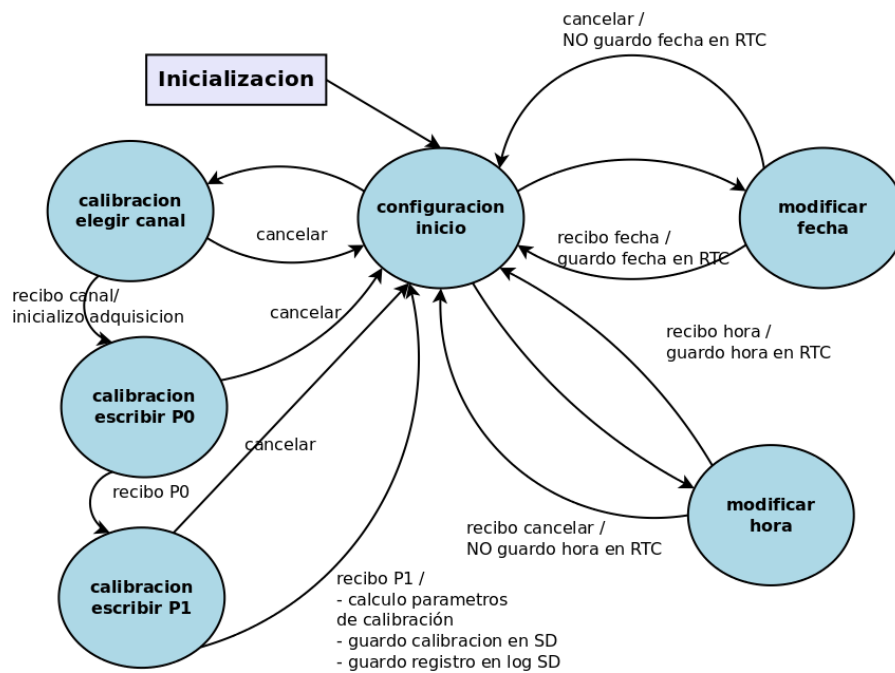


FIGURA 3.13: Máquina de estados de configuración de usuario

### 3.1.6. Implementación del código

La implementación de código se realizó en etapas, comenzando por los drivers y capas de abstracción de hardware. La librería utilizada para la configuración de los periféricos es la de LPCOpen, que es libre y cuenta con drivers, middleware y gran cantidad de ejemplos [19].

#### Código de bajo nivel: periféricos

En primer lugar, de acuerdo al diagrama de gantt de la figura 2.3, se trabajó sobre cada uno de los drivers de bajo nivel para lograr la funcionalidad individual de los periféricos e integrados asociados. A continuación se mencionan cada una de las particularidades de los diferentes módulos de hardware configurados:

- **Puerto Serie:** Se utilizaron las funciones de LPCOpen Chip\_UART para generar funciones una capa por encima de ellas. Se utilizó un esquema de productor-consumidor como se puede visualizar en la figura 3.3. El acceso a la UART se simplificó en dos colas de datos, tal cual puede verse en la figura 3.6, además de la participación de la tarea UART\_CanalSalida() que es la encargada de dar curso a los bytes escritos en la cola de salida. Se realizaron funciones de inicialización de las estructuras lógicas y de instalación de la tarea involucrada. La estructura de datos utilizada es la siguiente:

```

1
2 /** @brief tipo de dato para el manejo general de uart. */
3
4 typedef struct {
5     uint32_t baudrate;           /* velocidad del puerto */
6     LPC_USART_T *pUART;         /* puntero a la estructura de LPCOpen */
7     QueueHandle_t colaUARTRx;   /* cola de datos para recepcion */
8     QueueHandle_t colaUARTTx;   /* cola de datos para transmision */
9     SemaphoreHandle_t semThre;  /* mutex para controlar la salida de
                                datos */
10 } uart_t;

```

ALGORITMO 3.1: Estructura de datos asociada al manejo de la UART

Las colas de datos manejan punteros a char, que apuntan a buffers de bytes del largo del frame que utiliza el protocolo, en este caso, 4 bytes. Se agrega además un manejo simple de las FIFO de transmisión y recepción de 16 bytes con las que cuenta el microcontrolador. Por un lado, al escribir sobre el registro de salida, se tiene en cuenta que se pueden escribir hasta 16 bytes y luego hay que esperar a que se libere la FIFO de transmisión. Para ello se creó un semáforo binario llamado semThre que se suelta cada vez que el handler de interrupción detecta una entrada por vaciamiento del registro de transmisión (THRE: transmit holding register empty).

- **Puerto SPI:** El manejo del bus SPI dedicado a la comunicación con el ADS1292 se realizó utilizando directamente las funciones de bajo nivel de LPCOpen Chip\_SPI\_, ya que la comunicación con el mismo no es muy compleja ni demanda una atención más sofisticada. Una vez inicializado el puerto SPI, se intercambian una serie de comandos con el ADS1292 y no es necesario comunicarse nuevamente con él hasta comenzar la adquisición. Durante la misma, se genera una interrupción a través del pin DRDY y en ese momento se realiza una breve comunicación con el ADS1292 para recibir la muestra adquirida.

Por otro lado, el puerto SPI dedicado a la comunicación con la memoria SD también se comunica a través de las funciones básicas de la librería LPCOpen, pero se agrega una rutina de atención de interrupción para manejar comandos que requieren envío y respuesta inmediata.

- **Timer:** Se utiliza un timer que realiza una cuenta repetitiva para generar la frecuencia de muestreo. No se utilizó el handler de interrupción, simplemente se inicializa y se activa o desactiva cuando es necesario muestrear con el ADC interno.
- **ADC:** El ADC está configurado para ser disparado por el Timer 0 al setear los bits correspondientes del registro de control. Esta conexión entre el ADC y el Timer es interna por hardware, al configurarlo de esta manera, con cada interrupción del Timer se genera un disparo del ADC. Se muestrea un solo canal para la adquisición de la batería.
- **DMA:** El DMA está configurado para generar una transferencia desde un lugar fijo de memoria de periféricos (el registro AD0CR) a un puntero que se incrementa con cada copia. Al finalizar el conteo del buffer se genera una interrupción del DMA y se pasa a calcular un promedio de los valores, a la vez que se intercambia el puntero a un segundo buffer.
- **GPIO:** Las entradas utilizadas como propósito general son el pin DRDY para señalar que el conversor ADS1292 tiene datos listos, un pin de estado del conversor Serie-BT, un pin de detección de tarjeta conectada del slot microSD y la salida de detección de fuente de alimentación. En la rutina de interrupción simplemente se liberan semáforos o se escriben flags. También se utilizan salidas digitales manejadas directamente con las funciones de LPCOpen para la activación del integrado de carga de batería. Este último está normalmente apagado porque aumenta el nivel de ruido de todo el circuito.
- **USB:** Se trabajó a partir del ejemplo de LPCOpen `usbd_lib_hid_generic`, que muestra como utilizar la librería USB para enumerar al dispositivo como Device. También se recurrió a los recursos de software de ejemplo de la placa Embedded Artist[20], que contiene una implementación de la clase Mass Storage (MSC), con todas las funciones de acceso a memoria y los wrappers para los comandos SCSI que utiliza la MSC [21] [22] [23]. La aplicación configura el endpoint 1 para transferencias BULK y permite que el USB tome el control del acceso a la memoria SD, hasta que la PC finalice la comunicación o se detecte mediante el pin USB\_CONNECT que se perdió el vínculo.

### Configuración de drivers

- **ADS1292:** El ADS1292 requiere una serie de comandos para su inicialización[16] con delays intermedios. Se envía el comando RESET, luego se espera 1 segundo para después enviar el comando SDATAC. Si el envío es correcto, el ADS1292 responde con una cadena que contiene un ID asociado al modelo del chip (los demás chips de la familia responden ID diferente). Luego se pasa a escribir los registros de configuración CONFIG1 y CONFIG2, donde se especifican las referencias de tensión, la frecuencia de muestreo, los canales habilitados y la ganancia del PGA de cada canal. Al finalizar, se baja el pin de Chip Select del bus SPI. Luego de inicializarlo, el manejo

del ADS1292 se realiza en la tarea DatosListos(), que está bloqueada por el semáforo del pin DRDY. Cada vez que este semáforo desbloquea la tarea, activo el canal SPI con el pin Chip Select y leo los bytes recibidos. Luego empaqueto los 3 bytes en una variable de 32 bits utilizando una union y los almaceno en un buffer. Cuando los buffers de los canales habilitados están completos, se intercambian los punteros de almacenamiento y adquisición, y finalmente se escribe en la cola de datos colaDatosAlm o colaDatosEnv según el modo actual. Se utiliza una estructura de datos para cada uno de los canales con los siguientes campos:

```

1
2 typedef enum {INACTIVO, PENDIENTE, PROCESANDO, FINALIZADO}
   T_Estado_Buffer;
3
4 typedef struct
5 {
6     /* Defino un doble buffer para adquisicion. */
7     int32_t buf1 [CANAL_BUFFER_L];
8     int32_t buf2 [CANAL_BUFFER_L];
9
10    /* Indice de adquisicion */
11    int32_t ind;
12    /* Indice de envio */
13    int32_t ind_envio;
14
15    /* Defino puntero a buffer de adquisicionn y a buffer de
       almacenamiento: */
16    int32_t* pAct;
17    int32_t* pAlm;
18
19
20    /* flag para saber si hay un buffer lleno esperando a almacenarse
       */
21    T_Estado_Buffer estado_buffer;
22    bool buffer_lleno_pendiente;
23 } t_canal;

```

ALGORITMO 3.2: Estructura de datos asociada al canal del ADS1292

- **Bluetooth:** La comunicación por Bluetooth es transparente y se utiliza directamente como un puerto serie. La configuración del módulo HC-05 se realiza mediante una PC y el módulo se emplea ya inicializado.
- **Memoria SD:** El manejo de la memoria SD se realiza a través de las librerías FatFs. Se cuenta con funciones de inicialización de la capa más baja de hardware(disk\_initialize) y de inicialización del filesystem (f\_mount). Una vez que la memoria ya se encuentra montada, se pueden utilizar funciones de acceso a archivos como f\_open(), f\_write(), f\_sync(), f\_close(). Todas estas funciones recurren al driver SPI.

### 3.1.7. Pruebas Unitarias

Para cada uno de los periféricos y drivers mencionados anteriormente se realizaron scripts individuales para comprobar el funcionamiento de cada uno de los periféricos en forma aislada.

#### Pruebas sobre periféricos:

- **Puerto Serie:** se hizo la prueba de comunicación utilizando inicialmente la placa LPCXpresso1769 y un conversor USB-TTL conectado directamente a los pines del microcontrolador. Se utilizó el programa microcom [24] para realizar las pruebas de comunicación serie entre la PC y el microcontrolador.
- **Puerto SPI:** Las pruebas de comunicación SPI se realizaron en primer lugar con un osciloscopio digital Tektronix TDS2012C para verificar la frecuencia de la señal SCK. Luego se realizaron las pruebas directamente sobre los dispositivos a los que se conectan, como la memoria SD y el ADS1292, y ello permitió corregir errores de configuración del puerto.
- **Timer:** Se comprobó el funcionamiento del timer instalando un handler de interrupción básico que haga cambiar el estado de un pin. Luego se modificó la frecuencia de la interrupción y se comprobó para cada caso la frecuencia de parpadeo del pin en un osciloscopio digital Tektronix TDS2012C.
- **ADC:** Se realizaron pruebas de adquisición utilizando un generador de señales UTG4082A con señales senoidales para el análisis de frecuencia y señales tipo rampa para la comprobación del rango dinámico. Se utilizaron diferentes frecuencias de muestreo y para cada caso se realizó un descargo de memoria RAM del microcontrolador a un archivo, para su posterior visualización y análisis utilizando un script de python.
- **DMA:** Se comprobó la configuración del DMA utilizando el ADC y realizando adquisiciones, y se comprobó el correcto funcionamiento descargando los buffers de memoria transferidos por este periférico. La descarga de memoria se realizó igual que en el caso de la comprobación de funcionamiento del ADC, utilizando la herramienta de "Memory Dump" de la IDE LPCXpresso, que permite exportar un sector de la memoria del microcontrolador a un archivo en la PC, para luego analizarlo en un script de python.
- **GPIO:** Se realizaron pruebas de la configuración de los pines de salida utilizando la placa LPCXpresso1769 y leds conectados a través de un protoboard. Se realizaron pruebas para los pines de entrada utilizando la placa LPCXpresso1769 y un botón conectado a través de un protoboard, y comprobando el acceso a la rutina de interrupción.
- **USB:** Las pruebas de USB se realizaron en conjunto con las pruebas de hardware ya que es necesario para la comunicación con la PC. Para los test iniciales se utilizó un programa que registra al microcontrolador como un puerto serie a través del USB, y con ello se realizaron algunas correcciones de hardware. Luego se realizaron las pruebas individuales de la memoria SD y se comprobó su funcionamiento utilizando un lector de memorias USB. Finalmente se integró todo en un mismo programa para realizar la prueba en conjunto del USB y la memoria SD.

#### Pruebas sobre drivers:

- **ADS1292:** Se realizaron pruebas de adquisición armando un programa básico que solamente cuente con una rutina de interrupción para el pin de aviso de datos disponibles (DRDY) y una tarea que permita realizar la copia de las muestras de los canales habilitados hacia un buffer. Se colocó un breakpoint que detenga el programa al finalizar la escritura del buffer y se realizó una descarga de la memoria a un archivo para verificar la adquisición, al igual que con la prueba del periférico ADC interno. Se conectó



la entrada analógica del ADC a un generador de señales arbitrario UNI-T UTG4082A y se realizaron adquisiciones sobre señales senoidales y del tipo rampa, para verificar la característica de respuesta en frecuencia y de rango dinámico.

- **Bluetooth:** Se utilizó un módulo HC-05 conectado al puerto serie, y se emparejó luego a una tablet Android, con la aplicación Blueterm[25] instalada. Se realizaron pruebas de envío de comandos individuales, y luego de señales guardadas en la memoria del microcontrolador en bloques de 128 bytes, decimando las muestras a 100Hz y bajando la resolución de 24 a 16 bits. En la aplicación se realizó un descargo de memoria para comprobar que los bloques de memoria hayan sido transferidos correctamente.
- **Memoria SD:** Las pruebas de este driver se realizaron en conjunto con las pruebas de hardware. En primer lugar se realizó la prueba de conexión con la memoria SD, escritura de un archivo y luego apertura de un archivo. Finalmente se realizaron pruebas más exigentes en tiempo haciendo escrituras de buffers completos y midiendo el tiempo de escritura en la SD con un pin digital de salida: antes de ingresar a la función de escritura de la SD se estableció el estado del pin en 3.3V, y al salir de la función de escritura del bloque se volvió a establecer el estado del pin en 0V. Esto es necesario ya que se debe demorar menos tiempo en escribir el bloque adquirido en la SD que en adquirir dicho bloque. Luego de estas pruebas se configuró el tamaño del buffer de muestras en 1024 muestras.

### 3.1.8. Pruebas De Integración

Las pruebas de integración se realizaron a partir de la aplicación principal esquematizada en la figura 3.12. Esta aplicación abarca todas las funcionalidades del equipo excepto el acceso por USB. Al ser la máquina de estados totalmente conocida, y poder acceder mediante el debugger a la posición real del firmware del controlador (estados observables), se puede realizar una prueba exhaustiva de la función de interfaz de usuario. Utilizando la aplicación Blueterm en una tablet Android se realizó el recorrido por todos los estados posibles de la máquina (All-Path Coverage [26]), así como también se enviaron comandos erróneos o fuera de contexto. Estas pruebas permitieron corregir errores en las prioridades y tamaño de stack asignado a cada tarea. Luego se realizaron las mismas pruebas, pero en lugar de verificar el funcionamiento mediante el debugger, se realizó mediante un puerto serie auxiliar, y agregando mensajes de testing en el firmware, que luego se removieron en la versión final. Esto permitió independizar la prueba de las demoras del debugger.

### 3.1.9. Pruebas de Sistema

Las pruebas de sistema se discutirán en el capítulo 4 como "Ensayos y Resultados".



## 3.2. Implementación del Hardware

El desarrollo del hardware se realizó en tres etapas: dos etapas como prototipo experimental y una tercera con todas las correcciones como modelo final. Todos los modelos fueron diseñados íntegramente en Kicad.

### 3.2.1. Prototipo para adquisición del ADS1292

En primer lugar se realizó una placa como prototipo para prueba del ADS1292 y su conexión con el microcontrolador. En este caso se colocaron dos tiras de pines para conectar la placa LPCXpresso1769, y el ADS1292 a través del puerto SPI0, así como también un filtro analógico de entrada, basado en un circuito recomendado por el fabricante de los sensores. Se utilizó un regulador de tensión LDO 7333 para independizar la alimentación utilizando baterías o un adaptador de 4V a 9V, y de esta manera reducir el ruido que introduce la alimentación por USB de la placa LPCXpresso1769. Se realizó el ruteo independiente de la alimentación digital (DVCC-DGND) y analógica (AVCC-AGND). En la figura 3.14 puede verse un esquemático con las conexiones de entrada del ADS1292.

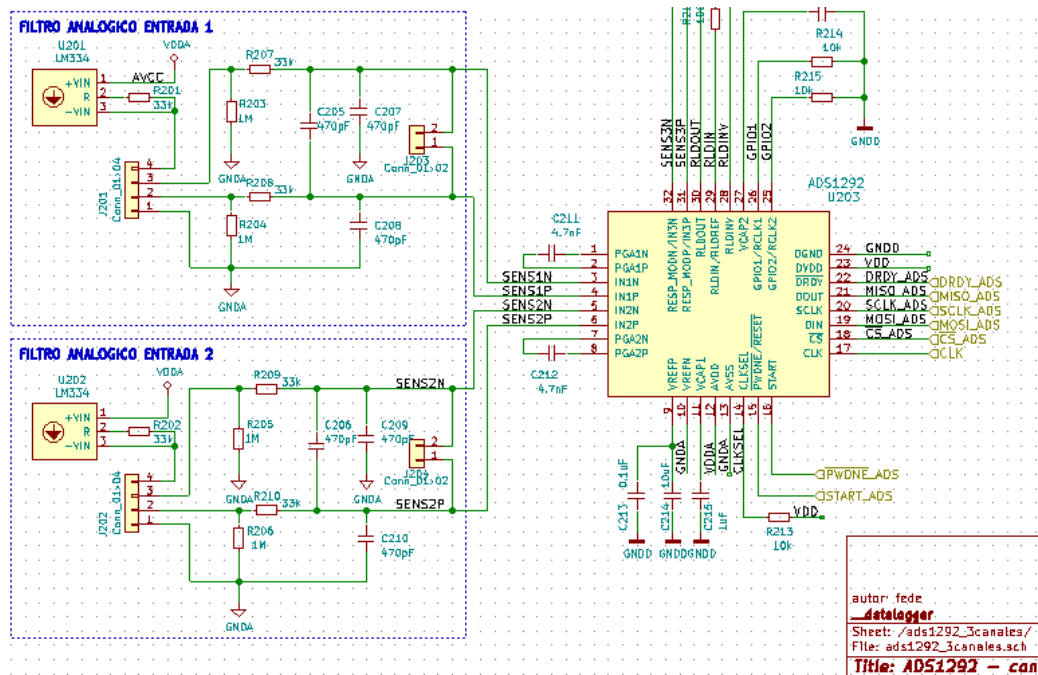


FIGURA 3.14: Esquemático de conexiones del convertidor analógico digital de alta resolución ADS1292

Este PCB se realizó en un router CNC redt MAGIC300 (Figura 3.15) sobre una placa de cobre virgen doble faz de 10x5 cm. Se realizaron varias correcciones sobre la misma. Se agregó también un slot de memoria micro SD conectado al puerto SPI0 para realizar las primeras pruebas de adquisición. Pueden verse fotografías de este prototipo luego de haber realizado varias correcciones en las Figuras 3.16 y 3.17



FIGURA 3.15: Router CNC empleado para los dos primeros prototipos.

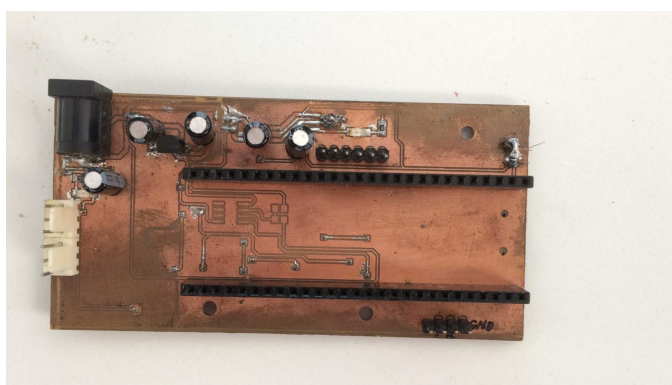


FIGURA 3.16: Vista superior del primer prototipo realizado.

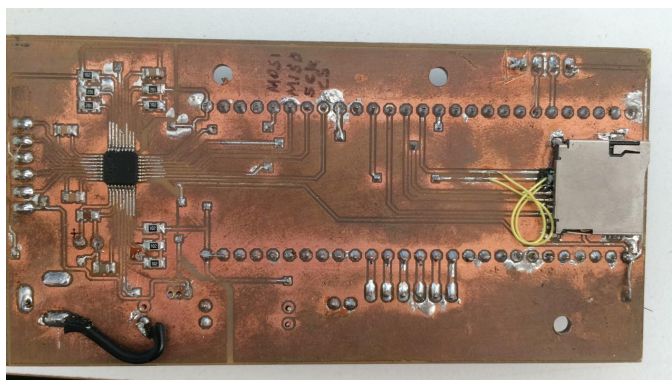


FIGURA 3.17: Vista inferior del primer prototipo realizado.





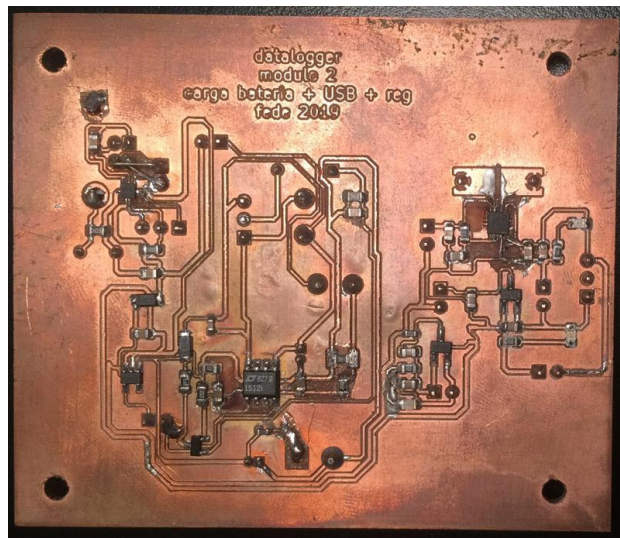


FIGURA 3.22: Circuito impreso del prototipo para carga de batería y USB

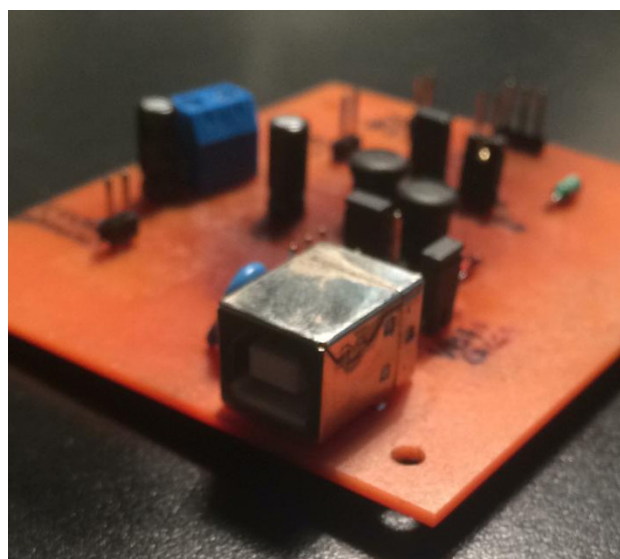


FIGURA 3.23: Vista superior del prototipo para carga de batería y USB



En este diseño también se agregaron los siguientes módulos:

- Amplificador operacional en modo seguidor conectado a una entrada del ADC para medir el nivel de batería y una entrada analógica adicional de baja resolución para una futura expansión. El resto de los esquemáticos son los mismos que los utilizados para los prototipos anteriores con leves correcciones.
- Puerto serie UART0 conectado a una tira de pines con el orden del módulo Bluetooth HC-05
- Puerto serie UART3 conectado a una tira de pines para conectar un módulo USB-TTL para log y debug.
- Se optó por el regulador de tensión TPS63001 ya que la diferencia del nivel de ruido con los reguladores LDO es mínima.
- El módulo del microcontrolador LPC1769, que es relativamente sencillo: contiene los cristales de alta y baja frecuencia y una serie de capacitores de filtro de tensión. A diferencia del esquemático de la placa LPCXpresso1769, se separaron las conexiones de tensión digital y analógica, de acuerdo a lo indicado en el manual del microcontrolador. Puede verse parte de su esquemático en la figura 3.24. También se incluyó una tira de pines al puerto SWD para programar el microcontrolador utilizando la placa LPCLink de NXP [28].

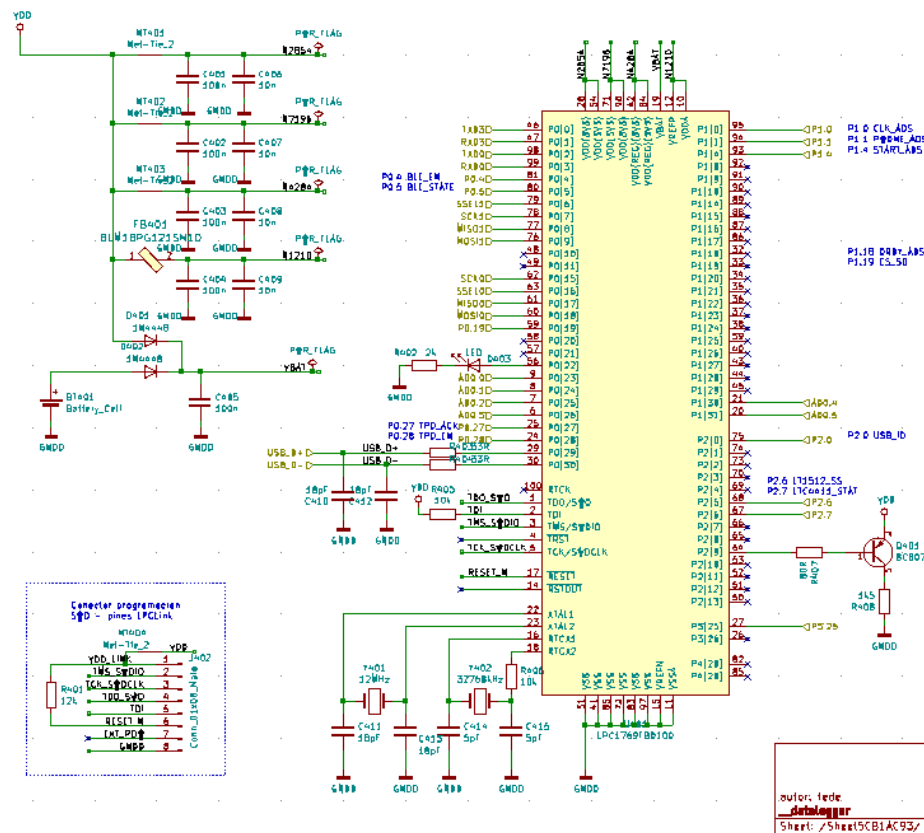


FIGURA 3.24: Conexiones del microcontrolador LPC1769

De acuerdo a los Requerimientos 2.1 y 2.2 se procedió al diseño de un circuito impreso lo más pequeño posible. Se establecieron las medidas de 4x4cm, pero

resultó ser insuficiente y se optó finalmente por un tamaño de 6x6 cm. Se realizó una placa doble faz y se terciarizó su fabricación en la empresa Seeed Studio (<https://www.seeedstudio.io/>). En las figuras

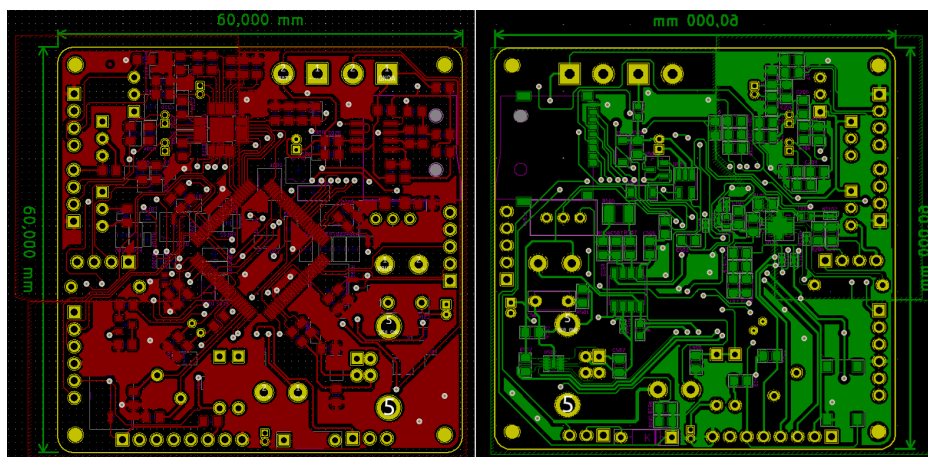


FIGURA 3.25: Vista frontal y trasera del diseño CAD del impreso final

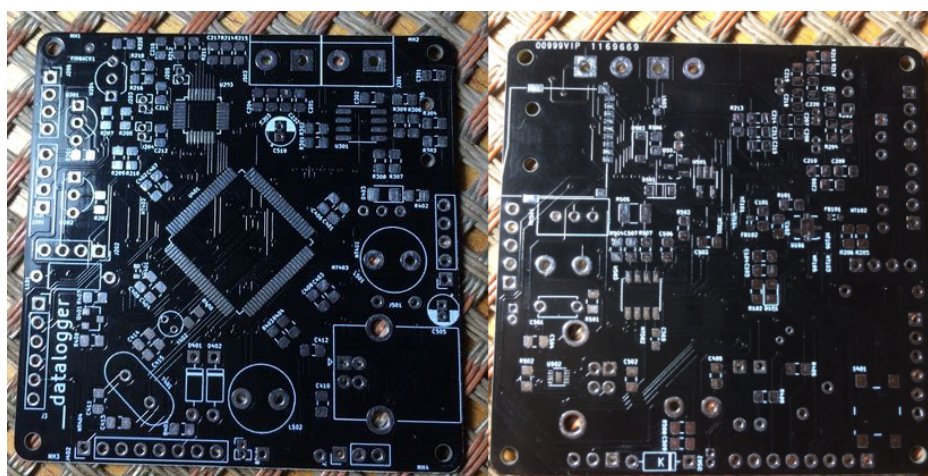


FIGURA 3.26: Vista frontal y trasera del Circuito impreso Fabricado sin Componentes





## Capítulo 4

# Ensayos y Resultados

### 4.1. Pruebas funcionales del hardware

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.



## Capítulo 5

# Conclusiones

### 5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

Para este proyecto se utilizaron en forma intensiva la mayoría de los contenidos y herramientas vistas durante el Carrera de Especialización en Sistemas Embebidos (CESE). Se pusieron en práctica técnicas de Gestión de Proyectos, documentación manual y automática del trabajo, sistema de versionado de software y hardware. En cuanto a lo técnico se emplearon conocimientos específicos sobre arquitectura del microcontrolador, modelos de programación, sistema operativo de tiempo real freeRTOS, protocolos de comunicación (BLE, SPI, USB, y de alto nivel), testing unitarios, etc.

### 5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.



# Bibliografía

- [1] World Health Organization. *QAs on hypertension*. 2015. URL: <https://www.who.int/features/qa/82/en/>.
- [2] T. Hansen y col. «Ambulatory blood pressure monitoring and risk of cardiovascular disease: a population based study.» En: *Amer. J. Of Hypertension, Ltd.* (2006). URL: <https://doi.org/10.1016/j.amjhyper.2005.09.018>.
- [3] J. Staensen y col. «Predicting Cardiovascular Risk Using Conventional vs Ambulatory Blood Pressure in Older Patients With Systolic Hypertension». En: *JAMA* (1999). URL: <https://doi.org/10.1001/jama.282.6.539>.
- [4] P. Verdecchia y col. «Ambulatory blood pressure. An independent predictor of prognosis in essential hypertension». En: *Hypertension* (1993). URL: <https://doi.org/10.1161/01.HYP.24.6.793>.
- [5] W.W. Nichols y col. «Effects of arterial stiffness, pulse wave velocity, and wave reflections on the central aortic pressure waveform.» En: *J of Clinical Hypertension*. (2008). URL: <https://doi.org/10.1111/j.1751-7176.2008.04746.x>.
- [6] G. Ogedegbe y T. Pickering. «Principles and techniques of blood pressure measurement». En: *National Institutes of Health* (2010). URL: <https://doi.org/10.1016/j.ccl.2010.07.006>.
- [7] R.Ruso y col. «Una implementación efectiva del método oscilométrico para la medición de la presión arterial.» En: *Memorias II Congreso Latinoamericano de Ingeniería Biomédica* (2001).
- [8] M.Saito y col. «One-Dimensional Model for Propagation of a Pressure Wave in a Model of the Human Arterial Network: Comparison of Theoretical and Experimental Results». En: *Journal of Biomechanical Engineering* (2011).
- [9] C.A.Figueroa y J.D. Humphrey. «Pressure wave propagation in full-body arterial models: A gateway to exploring aging and hypertension». En: *23rd International Congress of Theoretical and Applied Mechanics* (2014).
- [10] S. Laurent y col. «Expert consensus document on arterial stiffness: methodological issues and clinical applications.» En: *European Heart J.* (2006). URL: <https://doi.org/10.1093/eurheartj/ehl254>.
- [11] S. Omboni y col. «Twenty-four-hour ambulatory pulse wave analysis in hypertension management: current evidence and perspectives.» En: *Current Hypertension Rep.* (2016). URL: <https://doi.org/10.1007/s11906-016-0681-2>.
- [12] Tyco Electronics. *AMP Circular Connectors for Commercial Signal & Power Applications*. Inf. téc. Tyco Electronics, 2007.
- [13] Fowler K. «V-Model». En: *Developing and Managing Embedded System and Products* (2015).
- [14] Douglass B. *Design Patterns for Embedded Systems in C*. Newnes, 2011.

- [15] Zhou J. «DMA Ping-Pong Application». En: (2017). URL: <https://community.nxp.com/docs/DOC-334851>.
- [16] Texas Instruments. *ADS1292: Low-Power 2-Channel 24-Bit Analog Front-End for Biopotential Measurements*. Inf. téc. Texas Instruments, 2012.
- [17] ELM-Chan. «FatFs – Generic FAT Filesystem Module». En: (2014).
- [18] Amazon Web Services. *The FreeRTOS Reference Manual*. AWS, 2017.
- [19] NXP. *LPCOpen Libraries and Examples*. 2014. URL: <https://www.nxp.com/support/developer-resources/microcontrollers-developer-resources/lpcopen-libraries-and-examples:LPC-OPEN-LIBRARIES>.
- [20] Embedded Artists. *LPCXpresso Base Board Rev B User's Guide*. Inf. téc. Embedded Artits, 2013.
- [21] ARM. *ARM KEIL USB Component 6.13.6 MDK Middleware for USB Device and Host Communication*. 2019. URL: <https://www.keil.com/pack/doc/mw/USB/html/index.html>.
- [22] Craig Peacock. *USB in a Nutshell*. Inf. téc. Beyond Logic, 2002.
- [23] Jan Axelson. *USB Mass Storage. Designing and programming devices and embedded hosts*. Lakeview Research, 2006.
- [24] ubuntu. *Microcom - ubuntu manuals*. 2016. URL: <http://manpages.ubuntu.com/manpages/bionic/man1/microcom.1.html>.
- [25] pymasde.es. *Blueterm - Google Play*. 2013. URL: [https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=es\\_AR](https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=es_AR).
- [26] Nai-Wen Lin. *Finite State Machine Testing*. 2013. URL: <https://www.cs.ccu.edu.tw/~naiwei/cs5812/st7.pdf>.
- [27] Embedded Artists. *LPCXpresso LPC1769 rev B*. Inf. téc. Embedded Artits, 2011.
- [28] NXP. *LPCXpresso Getting Started with NXP LPCXpresso*. Inf. téc. NXP, 2013.