



## CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

# Firmware de dispositivo de adquisición de señales neurofisiológicas

**Autor:**

**Ing. Leandro Ezequiel Arrieta**

Director:

Dr. Ing. Diego Coulombie (UNLaM)

Jurados:

Mg. Ing. Eduardo Filomena (UNER)

Mg. Ing. Mara Fusco (UTN-FRH)

Mg. Ing. Pablo Slavkin (FIUBA)

*Este trabajo fue realizado en la ciudad de Lomas de Zamora,  
entre junio de 2021 y agosto de 2022.*



## *Resumen*

La presente memoria describe el desarrollo del software de un dispositivo de adquisición de señales neurofisiológicas cuyo hardware fue desarrollado por el Grupo de Tecnología en Neuroseñales de la Universidad Nacional de La Matanza (GTN-UNLaM). Dentro de las características del dispositivo, se destaca la posibilidad de adquirir señales de forma simultánea mediante 8 canales y la comunicación inalámbrica Bluetooth embebida en el propio dispositivo.

Para cumplimentar los objetivos se realizó un firmware embebido en C utilizando un sistema operativo en tiempo real y máquinas de estado finitas.

Fueron fundamentales los conocimientos adquiridos en las materias programación de microcontroladores, testing de software en sistemas embebidos, sistemas operativos de tiempo real I y II y desarrollo de aplicaciones sobre sistemas operativos de propósito general cuyos contenidos se aplicaron en el software implementado y en la herramienta de prueba también desarrollada.



## *Agradecimientos*

A completar más adelante



# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Dispositivo de adquisición de señales neurofisiológicas . . . . .	1
1.2. Estado del arte . . . . .	2
1.3. Marco normativo . . . . .	2
1.4. Motivación . . . . .	3
1.5. Objetivos y alcance . . . . .	3
1.5.1. Objetivos del trabajo . . . . .	3
1.5.2. Alcances del trabajo . . . . .	3
<b>2. Introducción específica</b>	<b>5</b>
2.1. Hardware del DASN . . . . .	5
2.1.1. Circuito de entrada . . . . .	6
2.1.2. Microcontrolador y comunicación inalámbrica . . . . .	7
2.1.3. Conceptos básicos del stack bluetooth 5.0 . . . . .	7
Capa GAP . . . . .	7
Capa GATT . . . . .	8
2.1.4. Requerimientos . . . . .	8
2.1.5. Herramientas de desarrollo . . . . .	10
<b>3. Diseño e implementación</b>	<b>11</b>
3.1. Arquitectura del sistema . . . . .	11
3.2. Módulo de control . . . . .	11
3.3. Módulo de adquisición . . . . .	11
3.3.1. Máquina de estados . . . . .	11
3.3.2. Protocolo de comunicación . . . . .	11
3.3.3. Configuración ADS1299 . . . . .	11
3.4. Módulo BLE . . . . .	11
3.4.1. Servicios bluetooth configurados . . . . .	11
3.5. Módulo Interfaz de usuario . . . . .	11
<b>4. Ensayos y resultados</b>	<b>13</b>
4.1. Banco de pruebas . . . . .	13
4.1.1. Software utilizado . . . . .	13
4.1.2. Hardware utilizado . . . . .	13
4.2. Test Unitarios . . . . .	13
4.3. Ensayos sobre la comunicación inalámbrica . . . . .	13
4.3.1. Verificación de servicios BLE . . . . .	13
4.3.2. Verificación de alcance . . . . .	13
4.4. Medicion de impedancia . . . . .	13
4.5. Adquisición de señal . . . . .	13

<b>5. Conclusiones</b>	<b>15</b>
5.1. Trabajo realizado . . . . .	15
5.2. Trabajo futuro . . . . .	15



# Índice de figuras

1.1. Diagrama en bloques del sistema. . . . .	1
2.1. Diagrama en bloques del hardware. . . . .	5
2.2. Diagrama en bloques del ADS1299. . . . .	6
2.3. Kit de desarrollo launchpad CC2640R2. . . . .	10



# Índice de tablas



*Dedicado a mis hijos, Guillermina y Agustin*



# Capítulo 1

## Introducción general

En este apartado se introducen los conceptos básicos sobre el dispositivo de adquisición de señales neurofisiológicas y sobre el proyecto de investigación del Grupo de Tecnología en Neuroseñales de la Universidad Nacional de La Matanza (GTN UNLaM).

### 1.1. Dispositivo de adquisición de señales neurofisiológicas

El dispositivo de adquisición de señales neurofisiológicas (DASN), es un módulo de hardware y software embebido destinado a ser parte de un sistema electro-médico. El desarrollo está enmarcado en una serie de proyectos de investigación del Grupo de Tecnología en Neuroseñales de la Universidad Nacional de La Matanza (GTN UNLaM) cuya finalidad es generar una plataforma de uso común para fabricantes locales, para investigación en las universidades y para el eventual desarrollo de nuevos productos y empresas de tecnología médica. El objetivo del presente trabajo final de carrera fue diseñar, implementar, ensayar y documentar la primera versión funcional de dicho software embebido de acuerdo a los lineamientos impartidos en las distintas materias de la Especialización en Sistemas Embebidos de la Universidad de Buenos Aires.

En la figura 1.1 se puede ver como ejemplo un diagrama en bloques de un sistema electromédico que incorpora al módulo DASN.

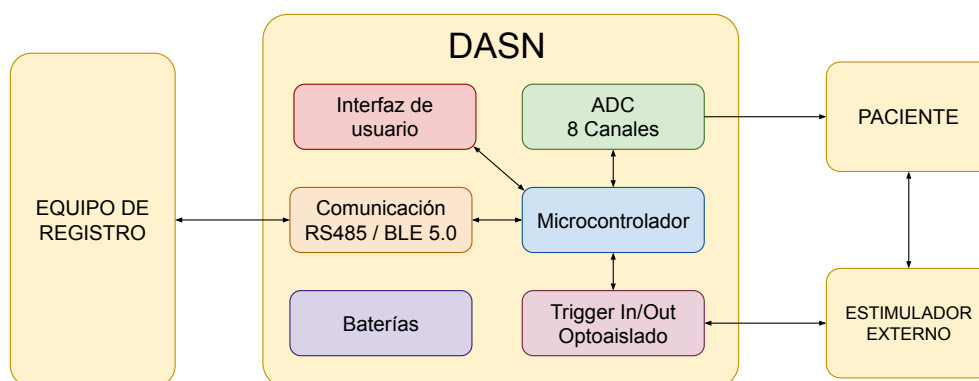


FIGURA 1.1. Diagrama en bloques del sistema.

Forman parte también del sistema un estimulador que interactúa con el paciente y un equipo de registro donde se guardan en formato digital los biopotenciales del paciente. Tanto el estimulador como el equipo de registro se comunican con el módulo DASN cuya función es adquirir señales eléctricas provenientes del sistema nervioso del paciente.

El DASN tiene 8 canales de entrada, comunicación inalámbrica a través de una comunicación bluetooth, y cableada a través de una comunicación RS485. Las señales adquiridas se pueden sincronizar con el estimulador externo, pudiendo funcionar como master o slave. Tiene la posibilidad de alimentarse por baterías y posee una interfaz de usuario para dar indicaciones de encendido, apagado y estado de funcionamiento.

## 1.2. Estado del arte

En particular para el subsector de la Neurofisiología, no existen placas comerciales que cumplan con las condiciones aplicables a un equipamiento médico de registro de señales neurofisiológicas. Si existen algunos ejemplos de placas de uso recreacional, didáctico o comercial, con un principio de funcionamiento similar pero destinado a interfaces cerebro-máquina (BCI). Ninguna de estas placas está destinada a la aplicación directa a la tecnología médica, ni posee certificaciones de calidad, ni garantiza características de funcionamiento esencial ni de seguridad básica.

## 1.3. Marco normativo

Las regulaciones de los dispositivos médicos tienen como objetivo proteger la salud de los pacientes, usuarios y terceros. Intentan asegurar la seguridad y eficacia de los productos disponibles en el mercado. Para cumplir con dichas regulaciones se debe actuar bajo las normas que son aceptadas por los organismos de control de los estados. Normas sobre sistemas de gestión de la calidad, sobre procesos y sobre productos alcanzan al diseño y la producción de equipamiento médico, incluido el software que lo conforma. La norma IEC 62304 [1] define las condiciones que debe cumplir el ciclo de vida del software; es decir, se trata de una norma de proceso. La IEC 60601-1 [2] (cuyo punto 14 se encarga del software del producto electro-médico o PEMS) define las características que debe cumplir el software para que sea seguro y eficaz. La IEC 62304 no impone o prohíbe una metodología de desarrollo de software en particular, pero sí indica cuales son las características que debe tener el proceso de ciclo de vida elegido. Debe tratarse de un proceso controlado, que tenga en cuenta la verificación y validación del software durante el proceso de desarrollo, que provea evidencia documentada para demostrar el cumplimiento del proceso, que responda a una planificación, que tome en cuenta el análisis de los riesgos relacionados con el producto, entre otras consideraciones más. La norma no entra en detalle de cómo esas actividades deben ser ejecutadas, dejando al fabricante la libertad de crear sus propias prácticas para que sean coherentes con los principios regulatorios.



## 1.4. Motivación

La Neurofisiología es una rama de las neurociencias, que se encarga del estudio funcional de la actividad bioeléctrica del sistema nervioso central, periférico y autonómico, mediante la utilización de equipos y técnicas de análisis avanzado, como la Electroencefalografía (EEG), la Electromiografía (EMG), los Potenciales Evocados (PE), la Polisomnografía (PSG) y otras nuevas técnicas como el neuromonitoreo (NM) o la medición de profundidad de anestesia (MPA). La plataforma de adquisición de señales neurofisiológicas está pensada para cubrir las necesidades de fabricantes de equipos del subsector de las neurociencias, para la investigación en las universidades y para el eventual desarrollo de nuevos productos y empresas de tecnología médica. En este marco nació la propuesta para dar vida al primer prototipo mediante la programación de su sistema embebido, con la motivación de aportar a la soberanía tecnológica y a la mejora del sistema de salud.

## 1.5. Objetivos y alcance

### 1.5.1. Objetivos del trabajo

Los objetivos del trabajo fueron: desarrollar el software del DASN de acuerdo a lo aprendido en las distintas materias de la carrera de especialización, lograr seguir un proceso de desarrollo controlado cumpliendo las fases de planificación, revisión y validación, y tener una primera versión funcional del DASN.

### 1.5.2. Alcances del trabajo

El presente proyecto incluye:

- Diseño del firmware embebido del DASN.
- Diseño del protocolo de comunicación entre DASN y equipo de registro.
- Software de prueba para simular un equipo de registro y probar el DASN.

El presente proyecto no incluye:

- Diseño de hardware.
- Filtrado digital de las señales adquiridas.
- Validación de la norma ISO 62304.



## Capítulo 2

# Introducción específica

En este capítulo se desarrollan las partes constituyentes del dispositivo de adquisición de señales neurofisiológicas, se introduce a su hardware y al stack bluetooth. Adicionalmente se muestran las herramientas de desarrollo utilizadas en el proyecto.

### 2.1. Hardware del DASN

El DASN está integrado en una sola placa y es un dispositivo portátil alimentado por baterías. Su arquitectura está basada en dos circuitos integrados principales, el microcontrolador CC2640R2 y el analog front end ADS1299. En la figura 2.1 podemos ver un diagrama en bloques del hardware.

DIAGRAMA EN BLOQUES DEL HARDWARE

FIGURA 2.1. Diagrama en bloques del hardware.

### 2.1.1. Circuito de entrada

El circuito de entrada del DASN está basado en un circuito integrado ADS1299 [Ref], el cual es un *Analog front end* (AFE). Los AFE son dispositivos que incorporan la parte analógica para el acondicionamiento de la señal y el conversor analógico digital para poder digitalizar la señal y leerla desde un microcontrolador. En particular el ADS1299 es un dispositivo de 8 canales, 24 bits por canal, conversión analógica digital zigma delta simultánea, con amplificadores de ganancia programable y un circuito multiplexor que permite conectar las entradas de múltiples formas que lo hacen ideal para el registro de biopotenciales y en particular de señales de EEG. En la figura 2.2 podemos ver un diagrama en bloques del ADS1299

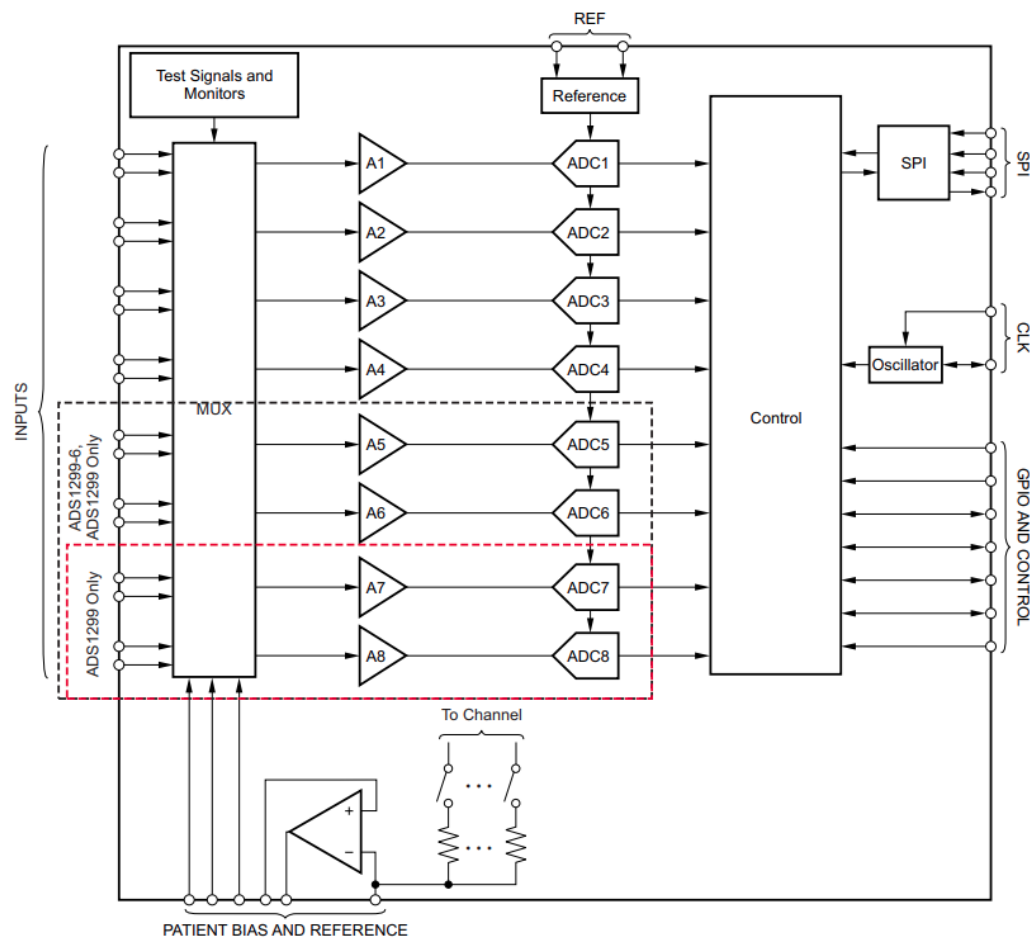


FIGURA 2.2. Diagrama en bloques del ADS1299.

El ADS1299 posee un puerto de comunicación SPI. Adicionalmente se dispone de señales de control para poder configurar y controlar el dispositivo. El microcontrolador del DASN está conectado a todas estas interfaces de control del ADS1299.

El DASN posee siete canales idénticos, los cuales solo poseen un filtro de radiofrecuencia entre el paciente y el AFE. El octavo canal, es un canal de potenciales evocados, los potenciales evocados son señales con amplitudes menores a 1 uVpp, lo que hace imposible medirlos con el ADS1299 ya que el ruido reflejado a la entrada del dispositivo es de 1 uVpp. Para solucionar este inconveniente el DASN incorpora un circuito amplificador de alto rechazo de modo común, el cual acondiciona la señal de potenciales evocados a niveles que puedan ser adquiridos con el ADS1299.

### 2.1.2. Microcontrolador y comunicación inalámbrica

El DASN posee un microcontrolador CC2640R2, el cual tiene una arquitectura Arm® Cortex®-M3. Este microcontrolador posee integrado un *transceiver* de 2.4 GHz compatible con la tecnología bluetooth de baja energía 5.1 y versiones anteriores del stack bluetooth [REF]. El hecho de tener este tipo de microcontrolador con el *transceiver* integrado permite configurar el stack bluetooth por completo, con lo que se logran mejores prestaciones en consumo y ancho de banda al poder elegir el protocolo de comunicación y modo de trabajo. La placa del DASN posee una antena tipo F [REF] integrada en el circuito impreso y también ofrece la posibilidad de conectar una antena externa.

### 2.1.3. Conceptos básicos del stack bluetooth 5.0

Se darán algunos conceptos básicos sobre el stack bluetooth ya que ayudarán a entender las configuraciones realizadas que se verán en el siguiente capítulo. Usaré los nombres en inglés ya que muchos términos no tienen traducción y la mayoría de la bibliografía así los referencia. El bluetooth de baja energía, también conocido por sus siglas en inglés como BLE, se divide en distintas capas. La aplicación de usuario entra en contacto con las dos capas superiores del stack bluetooth, la capa GAP y la capa GATT. A través de estas capas es que el software del DASN configura y maneja la comunicación BLE, por esta razón aborde con un poco más de profundidad estos temas.

#### Capa GAP

La capa GAP del stack bluetooth es responsable de las conexiones. De acuerdo a la definición de esta capa un dispositivo bluetooth se puede encontrar en uno de los siguientes estados:

- *Standby*: el dispositivo se encuentra en el estado inactivo inicial al reiniciarse.
- *Advertising*: El dispositivo se anuncia con datos específicos que le permiten a cualquier dispositivo saber que es un dispositivo conectable (este anuncio contiene la dirección del dispositivo y puede contener algunos datos adicionales, como el nombre del dispositivo).
- *Scanning*: En este estado se escuchan los mensajes enviados por algún *advertiser*.
- *Initiating*: En este estado se inicia la conexión, pero el *advertiser* es quien dependiendo los datos enviados por el *initiator* acepta o no la conexión.

- **Connected:** Cuando se establece una conexión ambos dispositivos pasan a estar en este estado.

A su vez, la capa GAP define cuatro roles para los dispositivos [ref]:

- **Broadcaster:** el dispositivo es un anunciante que no se puede conectar.
- **Observer:** el dispositivo busca anuncios pero no puede iniciar conexiones.
- **Peripheral:** el dispositivo es un anunciante que se puede conectar y funciona como peripheral en una conexión de una sola capa de enlace.
- **Central:** el dispositivo busca anuncios e inicia conexiones y funciona como central en una o varias conexiones.

### Capa GATT

La capa GATT es usada por la aplicación para el envío y recepción de los datos. Los datos son pasados y almacenados desde la aplicación de usuario al *stack* bluetooth en forma de características. La capa GATT define los siguientes roles para los dispositivos que se encuentran conectados:

- **GATT server:** Es el dispositivo que tiene la característica a ser leída o escrita por un *GATT client*.
- **GATT client:** Es el dispositivo que lee o escribe en la característica de un *GATT server*.

Los roles de la capa GATT (*client* o *server*) son independientes de los roles de la capa GAP (*peripheral* o *central*). Un *peripheral* puede ser tanto un *GATT client* como un *GATT server*, y lo mismo ocurre con un *central*.

Es importante destacar que el protocolo de comunicación BLE funciona por conexiones, cuando se tiene un nuevo dato no se transmite sino que se escribe en la característica para que el otro dispositivo que está conectado pueda ir a leerla.

#### 2.1.4. Requerimientos

##### 1. Requerimientos de interfaces externas

- a) El software deberá comunicarse con el equipo de registro utilizando una interfaz BLE 5.0.
- b) El software deberá comunicarse con el equipo de registro utilizando una interfaz RS485.
- c) El software deberá indicar mediante el led #1 que está transmitiendo las señales adquiridas de forma inalámbrica o cableada.
- d) El software deberá indicar mediante el led #2 si está encendido o apagado. También deberá indicar con el mismo led #2 si entra en el modo pairing BLE.
- e) El software deberá manejar una salida para sincronizar la adquisición con un estimulador externo (dispositivo externo como slave). La salida deberá poder configurarse entre normal bajo y normal alto. El pulso

deberá poder configurarse entre 5 anchos de pulso diferentes (0,1 ms; 0,5 ms; 1 ms; 5 ms; 10 ms).

- f) El software deberá manejar una entrada para sincronizar la adquisición con un estimulador externo (dispositivo externo como master). La detección deberá ser por flanco y se deberá poder configurar si el flanco es de subida o de bajada.
- g) El software deberá manejar el pulsador que servirá para encender el dispositivo y para realizar el pairing BLE.
- h) El software deberá generar la señal de impedancia para los canales de potenciales evocados.
- i) El software deberá medir el estado de las baterías con el ADC interno del MCU.

## 2. Requerimientos funcionales

- a) El software deberá configurar la frecuencia de muestreo de la señal adquirida entre 7 diferentes valores ( el ADC de "EEG-FrontEnd\_v1.x.0.SchDoc" ofrece 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz, 8000 Hz y 16000 Hz).
- b) El software deberá poder adquirir de 1 a 8 canales simultáneos.
- c) Mediante comandos recibidos por BLE 5.0 o RS485 el software deberá poder iniciar y parar la adquisición.
- d) El software deberá configurar la ganancia de amplificación de cada canal entre 7 diferentes valores ( el ADC de "EEG-FrontEnd\_v1.x.0.SchDoc" ofrece 1, 2, 4, 6, 8, 12 y 24).
- e) El software deberá configurar el ADC para usar una tipología de entrada diferencial o referencial.
- f) El software deberá medir la impedancia de los electrodos con una señal de medición de impedancia cuya frecuencia deberá ser de 7,8 Hz o 31,2 Hz.
- g) El software deberá seleccionar a qué electrodos le inyecta la señal de medición de impedancia.
- h) El software deberá seleccionar para cada electrodo si usa la señal de impedancia generada por el ADC o la generada por el MCU.
- i) El software deberá enviar la siguiente información de autodiagnóstico: temperatura del ADC, valor de las tensiones del ADC y frecuencia del clock del ADC.
- j) El software deberá prender y apagar el equipo con una pulsación corta del botón, pulsación menor a 1 segundo.
- k) El software deberá entrar en el modo de apareo de la comunicación BLE con una pulsación larga del botón, pulsación mayor a 4 segundos.
- l) Luego de 1 minuto de inactividad, el software deberá entrar a un modo de bajo consumo de energía. Para esto deberá apagar el ADC, el transceiver RS485 y la alimentación de todos los periféricos externos al MCU.

- m) El software deberá manejar una comunicación RS485 hasta 3 MBd.
- n) El software deberá poder recibir comandos mientras está transmitiendo las señales adquiridas.

### 3. Requerimientos de documentación

- a) La documentación debe cumplir los requisitos de la norma ISO 62304.
- b) Toda la documentación del proyecto se almacenará bajo un sistema de control de versiones GIT.
- c) Toda la documentación del código se realizará utilizando la herramienta Doxygen.
- d) Se deberá realizar un informe de avance del proyecto en el séptimo mes de trabajo
- e) Se deberá realizar la memoria técnica del trabajo final con la plantilla elaborada por la cátedra de gestión de proyecto.

#### 2.1.5. Herramientas de desarrollo

Para el desarrollo del firmware se utilizó el entorno de desarrollo integrado (IDE por sus siglas en inglés de Integrated Development Environment) Code Composer Studio (CCS) version: 10.4.0.00006 y el kit de desarrollo de software (SDK por sus siglas en inglés de Software Development Kit) SimpleLink CC2640R2 SDK: 5.10.00.02 ambos de Texas Instrument.

Como plataforma de hardware se dispuso de 2 placas de desarrollo LAUNCHXL-CC2640R2 [REF]. En la figura 2.3 podemos ver una de estas placas. Estas placas fueron de mucha utilidad para comenzar a familiarizarse con el dispositivo y las herramientas.

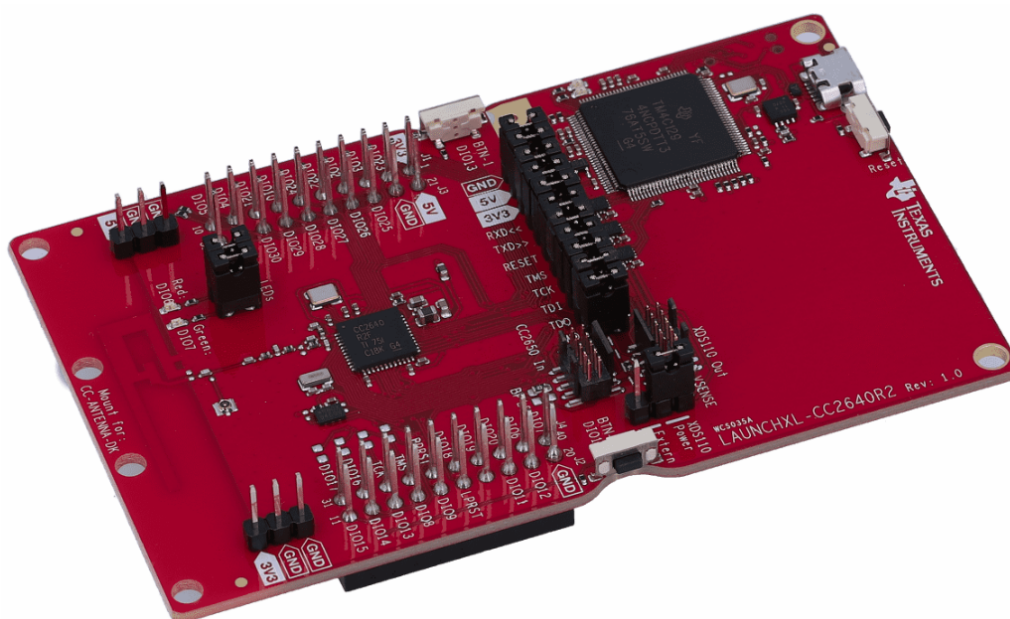


FIGURA 2.3. Kit de desarrollo launchpad CC2640R2.



## Capítulo 3

# Diseño e implementación

En este capítulo se describe la arquitectura de la aplicación, se muestra cómo interaccionan los distintos procesos entre sí y se explican las estrategias utilizadas para cumplir los requerimientos.

### 3.1. Arquitectura del sistema

### 3.2. Módulo de control

### 3.3. Módulo de adquisición

#### 3.3.1. Máquina de estados

#### 3.3.2. Protocolo de comunicación

#### 3.3.3. Configuración ADS1299

### 3.4. Módulo BLE

#### 3.4.1. Servicios bluetooth configurados

### 3.5. Módulo Interfaz de usuario



## Capítulo 4

# Ensayos y resultados

En este capítulo se describen los ensayos realizados y los resultados obtenidos. Adicionalmente se muestra el banco de prueba y herramientas utilizadas.

### 4.1. Banco de pruebas

#### 4.1.1. Software utilizado

#### 4.1.2. Hardware utilizado

### 4.2. Test Unitarios

### 4.3. Ensayos sobre la comunicación inalámbrica

#### 4.3.1. Verificación de servicios BLE

#### 4.3.2. Verificación de alcance

### 4.4. Medicion de impedancia

### 4.5. Adquisición de señal



## **Capítulo 5**

# **Conclusiones**

**5.1. Trabajo realizado**

**5.2. Trabajo futuro**