



# UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO  
INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

## Control de drones mediante bioseñales y FPGA Icezum Alhambra

---

### Autor

José Luis Arco López

### Directores

Encarnación Castillo Morales  
JoseMaria Cañas Plaza



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

—  
Granada, noviembre de 2019







# **Control de drones mediante bioseñales y FPGA Icezum Alhambra**

---

## **Autor**

José Luis Arco López

## **Directores**

Encarnación Castillo Morales  
Jose María Cañas Plaza



# **Control de drones mediante bioseñales y FPGA Icezum Alhambra**

José Luis Arco López

**Palabras clave:** Bioseñal, EMG, FPGA, dron, control

## **Resumen**

El propósito de este proyecto es la investigación, estudio y finalmente la realización de un sistema de adquisición de señales EMG, con el objetivo de poder usar dichas señales para controlar un dron radiocontrol. Se presenta como una oportunidad de usar las nuevas tecnologías de las FPGAS con un supuesto bastante necesario a nuestros días: el procesamiento de bioseñales para hacer la vida más fácil. Se comienza estudiando las señales electromiográficas y sus formas de obtención. Para ello se realiza un estudio de estas bioseñales, desde cómo obtenerlas hasta la instrumentación adecuada para tal fin. Tras realizar un banco de datos se desarrolla un algoritmo en Matlab para la extracción de la información de dichas señales para controlar luego el robot. A continuación se implementará este algoritmo en la FPGA de software libre Icezum Alhambra. Por último se realizarán las pruebas necesarias para depurar el sistema con simuladores para luego su puesta en marcha con robots reales.



# Drone control using biosignals and FPGA Icezum Alhambra

José Luis Arco López

**Keywords:** biosignal, EMG, FPGA, drone, control

## Abstract

The purpose of this project is the research, study and finally the realization of an EMG signal acquisition system, with the aim of being able to use these signals to control a radio-control drone. It is presented as an opportunity to use the new technologies of the FPGAS with a fairly necessary assumption to this day: the processing of biosignals to make life easier. It begins by studying the electromyographic signals and their ways of obtaining them. For this, a study of these biosignals is carried out, from how to obtain them to the appropriate instrumentation for this purpose. After performing a data bank, an algorithm is developed in Matlab for the extraction of information from these signals to then control the robot. This algorithm will then be implemented in the free software FPGA Icezum Alhambra. Finally, the necessary tests will be carried out to debug the system with simulators and then start up with real robots.



---

Yo, **José Luis Arco López**, alumno de la titulación Ingeniería de Tecnologías de Telecomunicación de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77142961H, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: José Luis Arco López

Granada a 11 de mes de 2019.



---

D.<sup>a</sup> **María Encarnación del Castillo Morales**, Profesora del Área de Electrónica del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

D. **Jose María Cañas Plaza**, Profesor del Área de Telemática del Departamento de Teoría de la Señal, Comunicaciones, Sistemas Telemáticos y Computación de la Universidad Rey Juan Carlos de Madrid.

**Informan:**

Que el presente trabajo, titulado ***Control de drones mediante bioseñales y FPGA Icezum Alhambra***, ha sido realizado bajo su supervisión por **José Luis Arco López**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes 11 de 2019 .

**Los directores:**

**María Encarnación del Castillo Morales      Jose María Cañas Plaza**



# Agradecimientos



# Índice general

<b>1. Introducción</b>	<b>19</b>
1.1. Motivación del proyecto y objetivos . . . . .	19
1.2. Planificación . . . . .	20
1.3. Estructura de la memoria . . . . .	20
<b>2. Estado del arte</b>	<b>23</b>
2.1. Robótica, drones . . . . .	23
2.1.1. Evolución histórica . . . . .	23
2.1.2. Tipos . . . . .	24
2.1.3. Protocolos de comunicaciones de un dron . . . . .	25
2.2. Bioseñales y el EMG . . . . .	27
2.2.1. Tipo de señal y ruido . . . . .	29
<b>3. Herramientas software y hardware</b>	<b>31</b>
3.1. Biosignalplus . . . . .	31
3.2. Matlab . . . . .	32
3.3. FPGAS . . . . .	32
3.3.1. FPGAs vs MCU . . . . .	33
3.3.2. FPGAS libres . . . . .	35
<b>4. Diseño del sistema</b>	<b>39</b>
4.1. Adquisición . . . . .	39
4.1.1. Preparación de la piel . . . . .	40
4.1.2. Los electrodos . . . . .	40
4.1.3. Obtención de señales . . . . .	40
4.1.4. Procesamiento en Matlab . . . . .	42
4.1.5. Comunicación serie en Matlab . . . . .	48
4.2. Diseño en icestudio . . . . .	48
4.2.1. Receptor serie . . . . .	48
4.2.2. Modulador de PWM . . . . .	50



# Índice de figuras

2.1. Patente del Kettering Bug . . . . .	23
2.2. Ala fija . . . . .	25
2.3. Ala rotatoria . . . . .	25
2.4. Modulación PWM . . . . .	26
2.5. Modulación PPM . . . . .	27
2.6. ECG de un corazón en ritmo sinusal normal . . . . .	28
2.7. EEG y colocación de los electrodos . . . . .	28
2.8. EMG y colocación de los electrodos . . . . .	29
3.1. Hub de Biosignalplus . . . . .	31
3.2. Ventana de OpenSignals con una señal EMG . . . . .	32
3.3. Estructura interna de una FPGA . . . . .	33
3.4. FPGA EP4CE6E22C8N . . . . .	33
3.5. MCU PIC16F887 . . . . .	33
3.6. La ley de Moore . . . . .	35
3.7. Icezum Alhambra II . . . . .	36
3.8. Icezum Alhambra II . . . . .	37
3.9. Interfaz de Icestudio . . . . .	38
4.1. Proceso para obtener y procesar bioseñales . . . . .	39
4.2. Posición de los electrodos . . . . .	40
4.3. Músculo flexor del carpo en el antebrazo . . . . .	41
4.4. Señal captada por OpenSignals . . . . .	41
4.5. Señal captada representada en Matlab . . . . .	43
4.6. Señal con el filtro paso banda . . . . .	44
4.7. Señal con el filtro paso banda y el rechazo banda . . . . .	45
4.8. Comparativa antes y después del los filtros . . . . .	45
4.9. Representación de la IAV . . . . .	46
4.10. Receptor serie en icestudio . . . . .	48
4.11. Primera parte del receptor serie . . . . .	49
4.12. Segunda parte del receptor serie . . . . .	49
4.13. Bloque PWM . . . . .	50
4.14. Estructura del Bloque PWM . . . . .	50



# Índice de Tablas

4.1. Señales de prueba y resultados . . . . .	47
4.2. Frecuencias y Reloj del sistema . . . . .	51



# Capítulo 1

## Introducción

### 1.1. Motivación del proyecto y objetivos

Este trabajo pretende poner en práctica todos los conocimientos adquiridos durante la carrera en base a un supuesto cada vez con más importancia en nuestros días: el control referido a señales biomédicas. Tras realizar mis estudios me di cuenta de que sabía un poco de muchas cosas, pero si quería tener un conocimiento específico en este tema tenía que trabajar e investigar por mi cuenta en mi propio proyecto.

Nuestros cuerpos están constantemente comunicando información sobre nuestra salud. Las señales biomédicas son la representación de las actividades del cuerpo, que pueden ser capturadas a través de instrumentos que miden la frecuencia cardíaca, la presión arterial, los niveles de saturación de oxígeno, la glucosa en sangre, la conducción nerviosa, la actividad cerebral, etc.

Tradicionalmente el procesamiento de señales lo que hacía era extraer información significativa de estas señales con el objetivo de comprender mejor el cuerpo humano. Este procesamiento es lo que permite a los médicos monitorear distintas enfermedades y tener toda la información posible para sus diagnósticos.

En nuestros días los ingenieros están haciendo un arduo trabajo descubriendo nuevas formas de procesar estas señales, ya no sólo para conseguir determinar el estado de salud de un paciente a través de medidas menos invasivas; si no también con el objeto de usar estas señales junto a las nuevas tecnologías para hacer la vida más fácil a las personas. Una nueva apuesta de trabajo surge en base al control de robots usando señales biomédicas que permitan hacer tareas por nosotros que no podríamos hacer de otra manera.

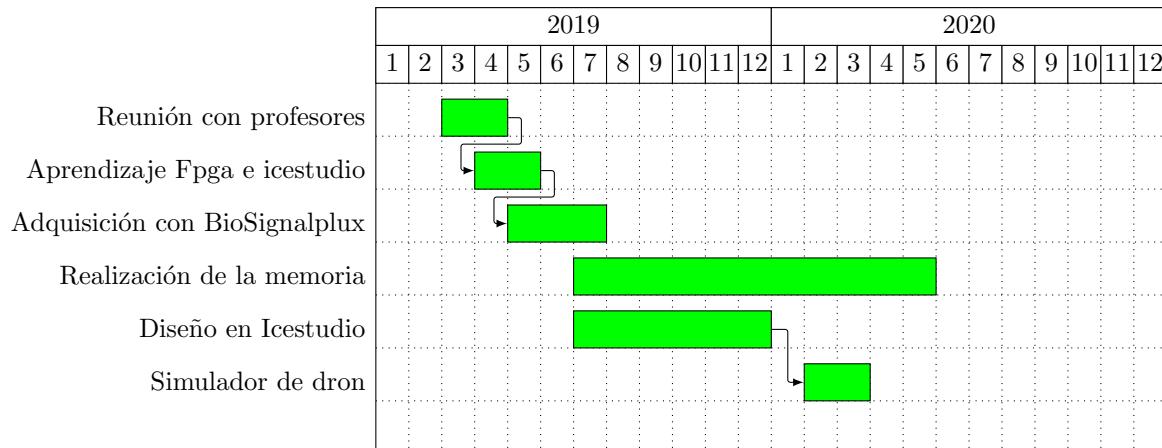
Se me presenta como ingeniero por lo tanto, la oportunidad de ofrecer un granito de arena a esta nueva apuesta investigando y usando una de las tecnologías más innovadoras hasta el momento, las FPGAS libres. Esta tecnología digital se encuentra cada vez más en auge debido a que es muy barata y permite unos procesamientos muy rápidos. Así usando estas FPGAS podríamos procesar señales adquiridas de nuestro cuerpo y usarlas para control de un sistema robótico.

Como sistema robótico que controlar se propone usar un dron telerigido ya que cada vez tienen más funcionalidades y su uso está muy extendido; por lo que tenemos bastante información sobre ellos.

El objetivo principal y más importante del proyecto sería por lo tanto conseguir el diseño de un sistema de control para robots a través de bioseñales. Para conseguir el objetivo principal, hay que conseguir otros objetivos más pequeños:

- **Estudio de las diferentes bioseñales disponibles.** Existen distintos tipos de bioseñales que pueden ser usadas para el método de control, entre las que vamos a destacar el *ECG*, *EEG* y *EMG*. Se determinará cuál es la más óptima para el desarrollo del proyecto.
  - **Adquisición de conocimientos básicos sobre distintos lenguajes y herramientas en los que incluimos:**
    - Lenguajes de descripción hardware, entre los que destacará *Verilog*.
    - Matlab.
    - Software de diseño digital para la FPGA, como *icestudio*.
    - Hardware de obtención de señales, como *Biosignalplus*.
    - Software de visualización de señales.
  - **Estudio del funcionamiento de las FPGAS existentes y diseño con ellas el sistema.**
  - **Uso estas herramientas y las señales captadas para desarrollar un modo de control.**

## 1.2. Planificación



### 1.3. Estructura de la memoria

Esta memoria se plantea a lo largo de 5 capítulos y un sexto capítulo de conclusión, en los que se exponen los primeros conceptos e ideas del proyecto hasta la realización del mismo; así como su final puesta en marcha.

En este mismo capítulo inicial se define la estructura de la memoria, se trata la planificación que se ha seguido así como la motivación a la hora de comenzar el proyecto.

En el capítulo 2 definimos el estado del arte en el cual se estudia la evolución y los tipos de dron que existen en la actualidad, además de los tipos de señales biomédicas existentes, entre ellas el EMG o señal electromiográfica que será con la cuál vamos a trabajar.

Posteriormente en el capítulo 3 exponemos las herramientas que se van a utilizar en toda la idea para su desarrollo, y a continuación en el capítulo 4 se realiza el diseño del sistema y su implementación.

Finalmente en el capítulo 5 se hacen las pruebas correspondientes y se exponen los resultados obtenidos.



# Capítulo 2

## Estado del arte

## 2.1. Robótica, drones

Un dron se puede definir como una *aeronave* pilotada de forma remota. El uso de los drones se ha ido instaurando poco a poco en la sociedad; actualmente con sus nuevas mejoras y reducido coste, estos han pasado de una situación en la que era más “amateur” a una función más profesional. En los últimos años han estado cada vez más presentes ya que ofrecen una gran herramienta de trabajo para diferentes sectores tanto comerciales como de seguridad.

### 2.1.1. Evolución histórica

La palabra inglesa *drone* (dron en español) tiene varios significados, aunque en su origen *drone* se refiere a los zánganos de las colmenas y al zumbido que emiten. El ruido que se escucha en tierra cuando vuelan a cierta altura estos vehículos recordaba a estos zánganos y de ahí que se quedaran con este nombre. Su evolución proviene sobretodo en el ámbito militar donde tenían funciones de reconocimiento y permitían tener ventaja en la batalla.

El inicio de los primeros vehículos no tripulados se remonta a la Primera Guerra Mundial. Estos primeros modelos fueron lanzados por catapulta o volados usando radio control. En enero de 1918, el ejército de los Estados Unidos comenzó la producción de torpedos aéreos telerididos de forma remota; desarrollando lo que podemos considerar el primer dron de la historia, el Kettering Bug (figura 2.1). El Kettering Bug fue volado con éxito en algunas pruebas; sin embargo aunque la guerra terminó antes de que pudiera desarrollarse más, marcó el inicio de la revolución de los drones .

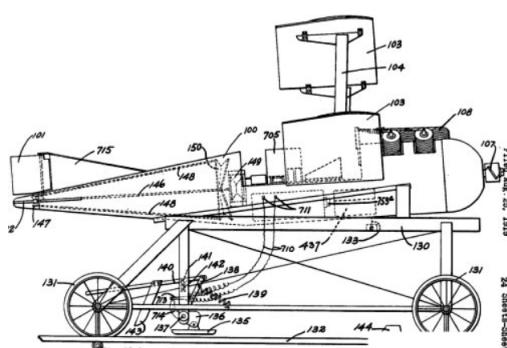


Figura 2.1: Patente del Kettering Bug

Los drones que conocemos hoy en día eran llamados *UAV* (Vehículos aéreos no tripulados). Los UAV con fines de reconocimiento se desplegaron por primera vez a gran escala en la Guerra de Vietnam. En esta época también comenzaron a usarse en una variedad de nuevos roles, como actuar de señuelos en combate, lanzar misiles contra objetivos fijos y lanzar folletos para operaciones psicológicas.

Después de la Guerra de Vietnam, otros países fuera de Gran Bretaña y Estados Unidos comenzaron a explorar la tecnología de los *UAV*. Los nuevos modelos se volvieron más sofisticados, con una resistencia mejorada y la capacidad de mantener una mayor altura. En los últimos años, se han desarrollado modelos que utilizan tecnología como la energía solar para abordar el problema de alimentar vuelos más largos.

Los drones ahora tienen muchas funciones, que van desde monitorear el cambio climático hasta llevar a cabo operaciones de búsqueda después de desastres naturales, fotografía, filmación y entrega de bienes. Pero su uso más conocido y controvertido es el uso militar para reconocimiento, vigilancia y ataques dirigidos. Desde los atentados terroristas del 11 de septiembre, Estados Unidos aumentó significativamente el uso de drones, aplicándose posteriormente a un nivel global para la seguridad. Se utilizan principalmente para la vigilancia en áreas y terrenos donde las tropas no pueden ir con seguridad; aunque, tristemente también tienen un gran potencial ofensivo.

[1] [2]

### 2.1.2. Tipos

Los drones están en su punto más alto y cada vez son más las empresas que definen sus productos como *drones*, por lo que existe un gran abanico de posibilidades en cuanto a su clasificación. [3] Sin embargo la más extendida son **según su uso** y **según el tipo de ala**.

Según su uso:

- **Uso Militar.** Se usan con fines de reconocimiento, como ofensiva e incluso reabastecimiento de provisiones. En este campo se pueden incluir los usados por la policía.
- **Uso Comercial.** Se encuentran desde los usados como diversión hasta los usados para publicidad,etc.

Según el tipo de ala:

- **Drones de ala fija 2.2.** Estos drones proporcionan una gran autonomía durante el vuelo ya que son muy aerodinámicos; sin embargo necesitan mucho espacio para aterrizar y un sistema de lanzamiento para despegar, lo que los hace muy engorrosos. Se utilizan sobre todo para hacer mapas y labores de riego



Figura 2.2: Ala fija

- **Drones de ala rotatoria****2.3.** Son los más extendidos en el ámbito comercial. A diferencia con los de ala fija estos pueden permanecer suspendidos en el aire gracias al sistema de hélices que llevan; no obstante tienen mucha menos autonomía que los de ala fija haciéndolos útiles solo para vuelos cortos. Cómo se pueden mantener estables gracias a sus giroscopios y estabilizadores son ideales para sacar fotos y hacer videos.



Figura 2.3: Ala rotatoria

Dentro de cada tipo, cada dron cuenta con un receptor radio determinado. A la hora de elegir el dron para nuestro proyecto será importante mirar las especificaciones del receptor radio que posee, ya que las comunicaciones entre el dron y la emisora (En este caso la FPGA) puede funcionar con distintos protocolos de comunicaciones.

### 2.1.3. Protocolos de comunicaciones de un dron

En el ámbito de los drones, las más importantes son PWM, PPM y PCM.

#### Modulación por ancho de pulsos (PWM)

La modulación por ancho de pulsos o PWM es un método que permite generar una señal analógica desde un sistema digital. Una señal de PWM es periódica y tiene dos elementos principales: un ciclo de trabajo y una frecuencia.

El ciclo de trabajo describe la cantidad de tiempo que la señal está en estado alto, como

un porcentaje con respecto a la cantidad de tiempo que está en estado bajo. La frecuencia determina cómo de rápido la señal completa un ciclo, y por lo tanto que tan rápido cambia entre los estados alto y bajo. Este ciclo de trabajo será modificado para transmitir una información o controlar la energía de la señal enviada, mientras que la frecuencia debe ser siempre la misma para poder medir el tiempo que la señal está en alto.

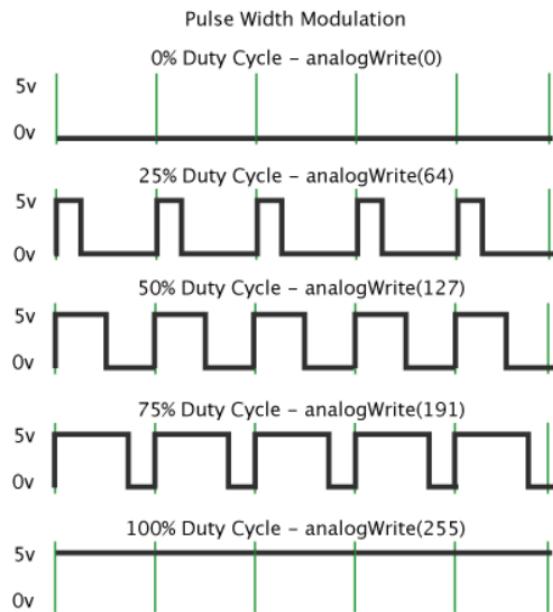


Figura 2.4: Modulación PWM

Esta modulación se utiliza mucho en la industria(ya que es la más común y generalmente la más barata.), y es con la que se controlan la mayoría de los servos en la robótica. Cada canal tiene su propio cable único, por lo que si tiene un receptor de 8 canales necesitará conectar 8 cables para leer las entradas en su controlador de vuelo.

### Modulación de posicion de puntos (PPM)

En la Modulación por Posición de Pulso o PPM la amplitud y el ancho son fijos, pero la posición del pulso es variable. Es un tipo de modulación en la cual una palabra de M bits se codifica para la transmisión de un pulso que se encuentra en las N posiciones posibles, donde N corresponde al tipo de modulación PPM (N-PPM).

$$N = 2^M \quad (2.1)$$

La ventaja de PPM es que solo se necesita un cable de señal para varios canales (típicamente 8 canales como máximo), en lugar de varios cables individuales. Por lo tanto, solo debe conectar el cable de tierra, alimentación y señal.

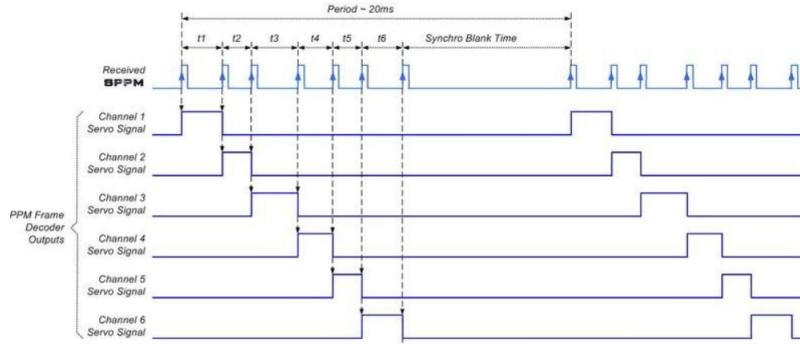


Figura 2.5: Modulación PPM

### Modulación de código de pulso (PCM)

La modulación de código de pulso o PCM, es un tipo de modulación similar a PPM; sin embargo, la señal modulada en PCM es una señal digital (usando unos y ceros) mientras que la señal PPM es analógica. La modulación PCM tiene un gran potencial de detección de errores e incluso de corrección, es más confiable y menos susceptible a las interferencias; no obstante, requiere varias conversiones adicionales para su funcionamiento lo que se traduce en equipos muy costosos.

## 2.2. Bioseñales y el EMG

Entendemos por bioseñal [4] cualquier señal de los seres vivos que se puede medir así como monitorear. Normalmente el término bioseñal se usa para refirse a señales bioeléctricas variantes en el tiempo; pero puede usarse también para señales no eléctricas, como por ejemplo la termografía o la respiración.

Estas bioseñales eléctricas generalmente se caracterizan por un cambio de corriente eléctrica; producido por la diferencia de potencial eléctrico a través de un tejido, órgano o sistema. Entre las más conocidas tenemos:

### Electrocardiograma

Un *electrocardiograma* [5] (ECG) es el registro de la actividad eléctrica del corazón durante un período de tiempo, utilizado por los médicos para predecir y tratar diversas enfermedades cardiovasculares. Se capta a través de unos electrodos en la piel que detectan la despolarización y repolarización del corazón durante cada latido cardíaco, lo que proporciona una visión de la actividad muscular del corazón.

Un trazado de ECG típico es un ciclo de tres entidades:

- **La onda P**, que representa la despolarización de las aurículas.
- **El complejo QRS**, que representa la despolarización de los ventrículos .
- **La onda T**, que representa la repolarización de los ventrículos.

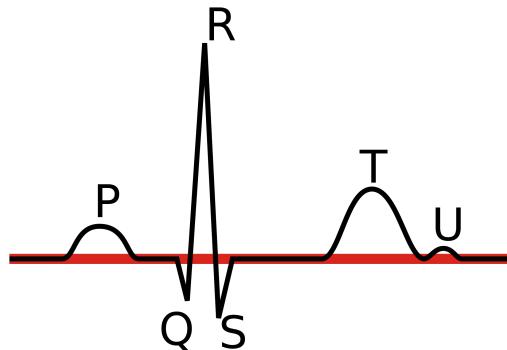


Figura 2.6: ECG de un corazón en ritmo sinusal normal

### Electroencefalografía

La *electroencefalografía* (EEG) mide la actividad eléctrica del cerebro, registrada a partir de electrodos colocados en el cuero cabelludo. Cuando se analizan estas señales, se utilizan como una herramienta de diagnóstico para detectar patologías asociadas con un comportamiento eléctrico extraño. Se usa con mayor frecuencia para diagnosticar la epilepsia , lo que causa anormalidades ev en las lecturas del EEG. También se usa para diagnosticar trastornos del sueño , profundidad de la anestesia , coma , encefalopatías y muerte cerebral.

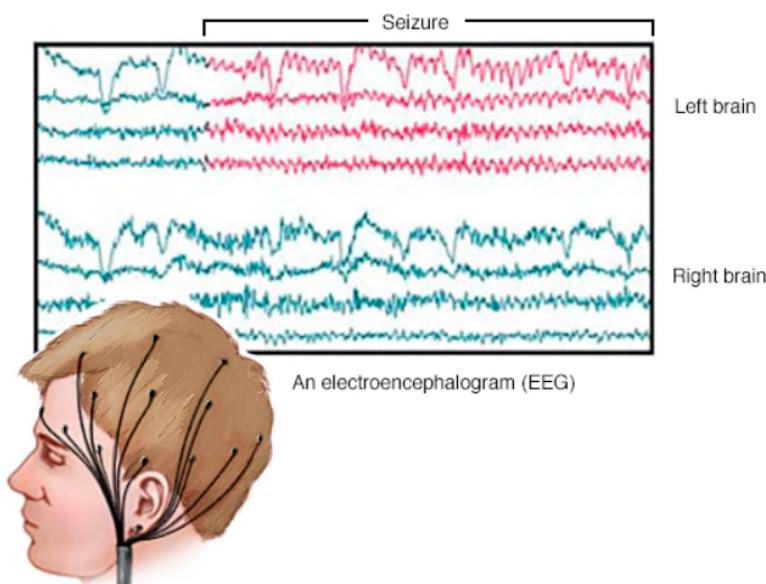


Figura 2.7: EEG y colocación de los electrodos

### Electromiograma

La *electromiografía* ( EMG ) es una técnica de medicina electrodiagnóstica para evaluar y registrar la actividad eléctrica producida por los músculos esqueléticos . Cada vez que se ejecutan los músculos del cuerpo para una determinada actividad, el cerebro envía señales de excitación a través del *sistema nervioso central*. Los músculos están inervados en grupos llamados *unidades motoras*, que son el punto de unión donde se encuentran la neurona motora y las fibras musculares. Cuando la *unidad motora* se activa, produce un potencial de acción y el sistema nervioso central se activa continuamente durante el tiempo que se requiera que

el músculo genere fuerza. Este *potencial de acción de la unidad motora* es la señal EMG que vemos resultante.

Para obtener el EMG se utiliza un electromiógrafo, el cuál detecta el potencial eléctrico generado por las células musculares cuando estas células se activan eléctrica o neurológicamente. En el ámbito médico por lo tanto el EMG se utiliza como herramienta de diagnóstico para enfermedades neuromusculares, pero sus usos se han ampliado actualmente a la rehabilitación de pacientes en forma de prótesis robótica.

Y es que en nuestros días un mecanismo robótico con varios grados de libertad puede imitar perfectamente el movimiento de una extremidad humana.

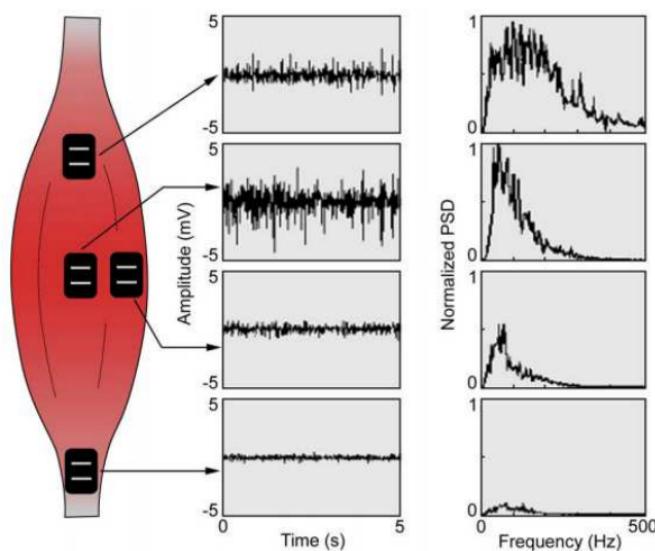


Figura 2.8: EMG y colocación de los electrodos

Se propone usar la bioseñal EMG en este proyecto debido a la facilidad para discriminar movimientos, que será lo que permita controlar el dron y definir su comportamiento. Sin embargo, antes de poder usar la señal EMG hay que tener varios factores en cuenta, que va desde su captación hasta las posibles interferencias que pueden acusar a este tipo de señal.

### 2.2.1

#### 2.2.1. Tipo de señal y ruido

Antes de pasar a la fase de adquisición de la señal, es importante familiarizarse con los distintos factores que afectan a las propiedades cualitativas de la señal.

Una señal de EMG tiene las siguientes características:

- La amplitud de la señal EMG se encuentra entre 1-10 mV, lo que la convierte en una señal considerablemente débil.
- La señal se encuentra en el rango de frecuencia de 0-500 Hz y la más dominante es entre 50-150 Hz.
- La señal EMG está muy influenciada por el *ruido*:

- El ruido ambiental puede ser causado por fuentes de radiación electromagnética, por ejemplo, dispositivos de transmisión de radio, luces fluorescentes y la interferencia de la línea de alimentación de los cables eléctricos. Estas interferencias son casi imposibles de evitar por medios externos. Este ruido particular existe en el rango de frecuencia de 50-60 Hz.
- El ruido también se puede generar a partir de artefactos de movimiento. Las dos fuentes principales de este ruido son la inestabilidad de la interfaz de la capa del electrodo y el movimiento del cable del electrodo y se encuentra principalmente en el rango de 0-20 Hz. Se puede eliminar mediante un conjunto adecuado de equipos y circuitos de EMG. La máxima fidelidad de la señal está determinada por la relación señal / ruido EMG adquirida.

El kit biosignalplus nos va a permitir adquirir las señales EMG y poder ver la interacción eléctrica en tiempo real. Este kit será el que utilicemos para obtener las señales EMG y procesarlas posteriormente.

## Capítulo 3

# Herramientas software y hardware

### 3.1. Biosignalplus

Para la adquisición usaremos el kit de investigación de Biosignalplus de 8 canales. Utiliza el software OpenSignals para adquirir y visualizar simultáneamente hasta 8 sensores de EMG.

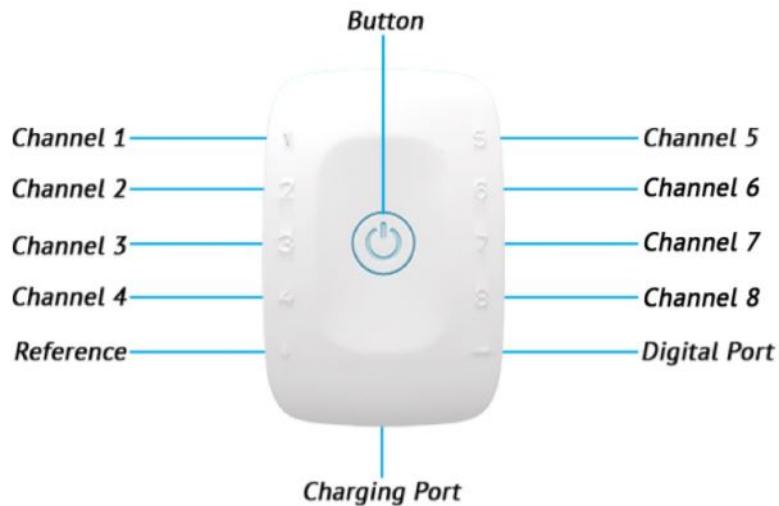


Figura 3.1: Hub de Biosignalplus

Con respecto a su utilización simplemente hay que conectar los electrodos para la captación de señales EMG que vienen con el kit Biosignalplus, y conectarlos en cualquiera de los 8 canales de los que disponemos. Cabe destacar la importancia del puerto de referencia, que colocándolo en una superficie neutra de nuestro cuerpo nos permitirá obtener señales EMG mucho más precisas.

El hub de Biosignalplus cuenta con conexión Bluetooth, por lo que para captar los datos que nos interesan basta con conectarlo mediante Bluetooth a un ordenador con el software de OpenSignals. Tras la selección del canal ya empezaremos a ver las señales en tiempo real automáticamente, las cuales podrán ser guardadas para su posterior estudio en este mismo programa.

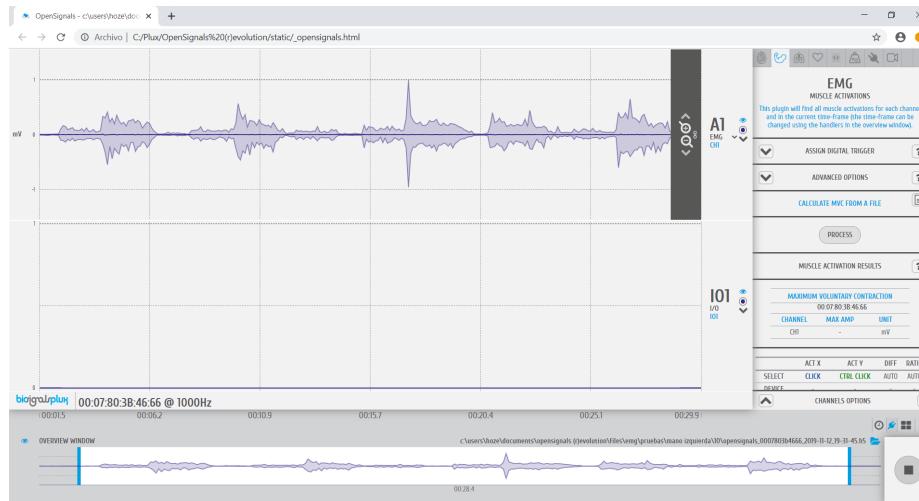


Figura 3.2: Ventana de OpenSignals con una señal EMG

## 3.2. Matlab

Es un programa de cálculo numérico el cual cuenta con su propio lenguaje de programación. Permite la comunicación con dispositivos hardware así como la visualización y programación de manera fácil e intuitiva dónde las soluciones se expresan como funciones matemáticas.

Como ventaja permite resolver problemas computacionales en un tiempo menor de lo que tardaría un lenguaje escalar no interactivo como C.

## 3.3. FPGAs

Una *FPGA* (Field Programmable Gate Array) es un dispositivo programable con bloques lógicos cuya interconexión puede ser configurada mediante un lenguaje de descripción para crear circuitos digitales.

Están compuestos internamente por puertas lógicas, cables y biestables; todo ello sin conectar, como una plantilla en blanco. Para su programación requiere la carga de un archivo con la descripción del circuito - o bitstream- que define las uniones entre los biestables y puertas lógicas para crear el circuito que deseemos.

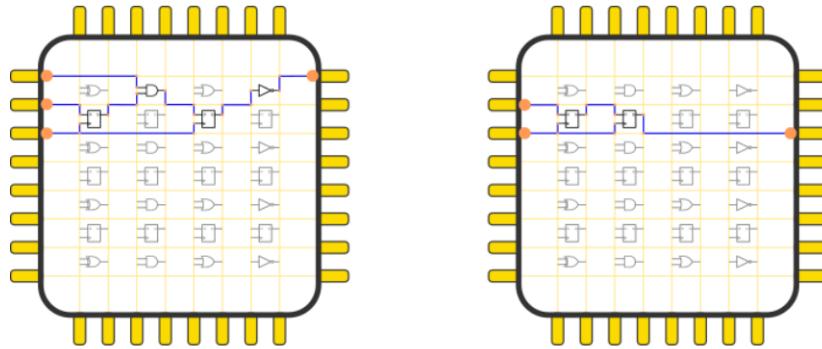


Figura 3.3: Estructura interna de una FPGA

### 3.3.1. FPGAs vs MCU

Las *FPGA* y los *MCU* (microcontroller unit - microcontrolador) son dos de las herramientas más poderosas disponibles para un ingeniero electrónico en la actualidad. Es esencial para cualquier persona que trabaje en electrónica digital, como para un profesional en cualquier industria, o incluso un aficionado, obtener una comprensión completa de las herramientas disponibles para ayudar en su trabajo. Esto es especialmente cierto en una industria acelerada, donde las herramientas y tecnologías están sujetas a cambios repentinos y frecuentes.



Figura 3.4: FPGA EP4CE6E22C8N

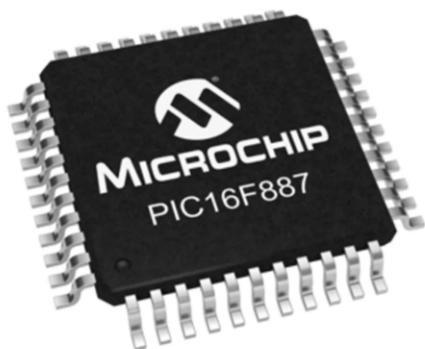


Figura 3.5: MCU PIC16F887

Estos dispositivos se usan generalmente en escenarios muy similares; para decirlo de manera muy amplia: los dos permiten introducir unos valores de entrada y obtener los cambios que deseemos en las salidas. De hecho, casi cualquier aplicación que use un microcontrolador podría emplearse en una FPGA para el mismo efecto, y viceversa. Esto no quiere decir que las dos tecnologías sean intercambiables; a veces lo que es muy difícil lograr usar en un dispositivo, podría ser una cuestión simple para el otro.

La diferencia más importante entre los microcontroladores y las FPGAs es la forma en que procesan las instrucciones. Los primeros procesan comandos secuencialmente, lo que significa que un micro lee cada línea del programa de una en una, en secuencia. Por el contrario, los FPGA procesan comandos simultáneamente, por lo que las líneas de código se ejecutan a la vez; traduciéndose en un incremento de la velocidad. Además en un microprocesador cada instrucción está asociada a un hardware de manera fija, y el programador no puede usar más instrucciones de las que le ofrece el fabricante. Las FPGAs no tienen nada conectado de manera fija, el usuario es el que determina mediante conexiones el comportamiento del circuito.

Debido a esto las FPGAs generalmente se usan en escenarios que requieren un procesamiento de datos en paralelo de alta velocidad, o un alto grado de personalización, pero pueden resultar engorrosas por su relativa complejidad de configuración. Mientras tanto, los MCU son mucho más fáciles de configurar y usar desde el principio, manejan de manera aceptable los datos en serie de alta velocidad y su producción es más barata.

Sin embargo, el panorama actual de los MCUs está a punto de cambiar debido al estancamiento de la ley de Moore. 3.3.1

### El estancamiento de la ley de Moore

La Ley de Moore es la observación empírica realizada el 19 de abril de 1965 por Gordon Moore, cofundador de Intel , en la que postulaba que el número de transistores por unidad de superficie de un microprocesador iba a duplicarse cada 2 años- y por lo tanto la velocidad del procesador o la potencia de procesamiento general-, al menos durante las siguientes dos décadas.

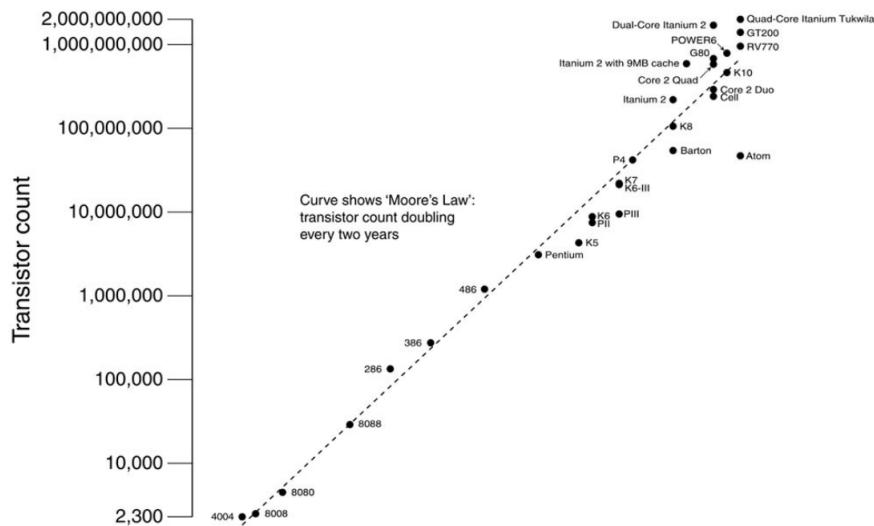


Figura 3.6: La ley de Moore

La mayoría de los expertos, incluido el propio Moore esperan que esta apreciación se mantenga vigente hasta 2020-2025 donde dejará de cumplirse definitivamente . El estancamiento de la ley de Moore es la consecuencia de los límites físicos de la tecnología actual, en la que un número mayor de transistores para un mismo volumen se traduce en un mayor calor generado que no se va a poder disipar sin dañar el microcontrolador (Enfriar los transistores requiere más energía de la que pasa por los transistores).

Este estancamiento de la tecnología de los microprocesadores, hará que no se pueda equiparar la potencia a la continua mejora de la velocidad de sus algoritmos; por lo que será necesario la búsqueda de nuevas tecnologías para que las empresas puedan seguir desarrollando sus productos.

Aquí es donde entran en juego las FPGAs y el hecho de la importancia que están adquiriendo para las empresas su uso; ya que les permite multiplicar por un factor considerable la velocidad en sus circuitos. Las FPGAS se conocen desde los años 80, pero era una tecnología cerrada que muy pocos podían usar; además, las herramientas para manejarlas eran muy complejas. Todo cambió con la llegada de las FPGAS "libres", que ha propiciado que muchos aficionados creen herramientas para poder manejarlas de una manera más intuitiva y sencilla.

#### 3.3.2. FPGAS libres

El término *libre* se refiere a los productos que se han diseñado específicamente para que sean totalmente accesibles para todos los usuarios, disponiendo de total libertad para utili-

zar, modificar y trabajar con estos productos. Las ventajas que permiten son enormes para el patrimonio tecnológico. Normalmente son mucho más fiables ya que hay un gran número de usuarios que trabajan para hacer que llegue a más gente.

A las FPGAs les llegó su momento en mayo del 2015, en el panorama cambió con el Proyecto IceStorm creado por el ingeniero austriaco Clifford Wolf.

Wolf hizo ingeniería inversa de una de la FPGA ICE40 de Lattice para obtener toda la información de funcionamiento y poder crear herramientas libres que no dependieran de ningún fabricante. Esto permitió un gran desarrollo del sector ya que rápidamente se creó una gran comunidad. La placa que usaremos para el proyecto será la Icezum Alhambra, pero existe una amplia variedad de placas con FPGAs libres.

### **Icezum Alhambra II**

Tras un pequeño estudio del mercado de las FPGAs libres, la placa Icezum Alhambra II será la que utilicemos para este proyecto, ya que esta enfocada a la docencia. Tanto la primera versión como la revisión fueron diseñadas en Pinos del Valle(Granada) por Eladio Delgado. Sus dimensiones y apariencia es muy similar al de la placa Arduino UNO: el objetivo era que fuera compatible con la mayoría de Shields de Arduino, que son placas de circuitos modulares que le dan funcionalidades extras. Además, como hemos comentado anteriormente es una placa que está pensada para el aprendizaje, por lo que cuenta con bastantes añadidos como LEDs y pulsadores para que sea más fácil y cómodo su uso para pequeños proyectos.



Figura 3.7: Icezum Alhambra II

Las características principales de la *Icezum Alhambra II* son:

- Placa de desarrollo con FPGA iCE40HX4K (Lattice) .
- Hardware abierto.
- Compatible con Icestudio.
- Se pueden reutilizar la mayoría de los shield disponibles en Arduino.
- Placa recomendada para proyectos robóticos.
- El dispositivo USB FTDI 2232H permite la programación de la FPGA y la interfaz UART la conexión a un PC.
- Interruptor electrónico de encendido/apagado.

- 8 LED de uso general (LED de usuario).
- 2 botones de propósito general.
- Memoria flash de 32 Mb.
- 20 pines de entrada / salida de 3.3V (tolera 5V).
- Todos los pines de E/S incluyen resistencias en serie de 200 ohmios para la activación directa de LED.
- Convertidor A/D de 4 canales y 12 bits.
- Pines de arranque en frío accesibles desde GPIO.
- Reguladores de commutación para procesamiento de alta frecuencia a baja potencia de entrada.
- Oscilador MEMS de 12 MHZ.
- Botón de reinicio.
- Fuente de alimentación USB, dos conectores (hasta 4.8A).

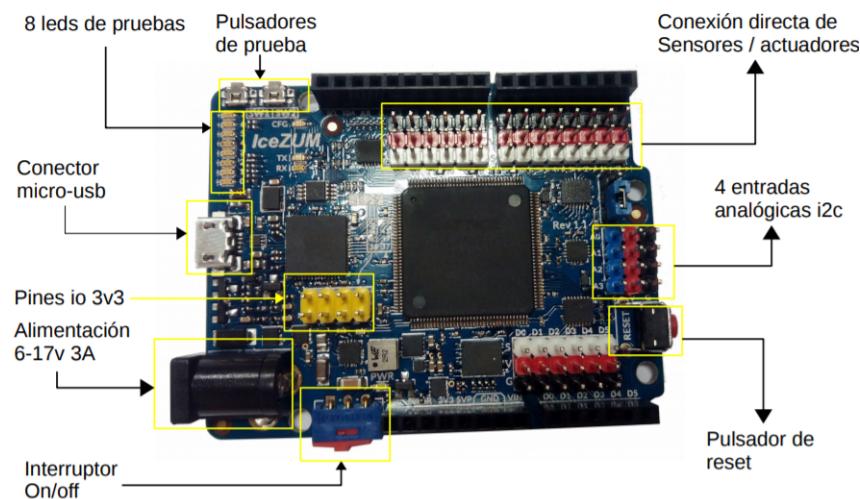


Figura 3.8: Icezum Alhambra II

## Icestudio

Icestudio se trata de un editor visual para FPGAs libres, el cual está construido sobre el proyecto IceStorm usando Apio. Apio es una serie de herramientas multiplataforma para FPGAS libres, con paquetes preconstruidos estáticos, herramientas de configuración de proyectos y una interfaz de comando fácil para verificar, sintetizar, simular y cargar diseños en Verilog.

Icestudio surge de la necesidad de simplificar la ardua tarea que implica el aprendizaje de lenguajes HDL, haciéndolos más accesibles para todos. Éste permite el diseño de forma bastante sencilla para FPGAS mediante la creación de bloques con funcionalidades definidas. Se utiliza el lenguaje de descripción hardware Verilog para la creación de estos bloques, que se indica al usuario como puertas lógicas y biestables.

En la figura 3.9 se muestra la interfaz del programa. En la parte central es donde se colocan los bloques creados previamente, los cuales se relacionan para crear funcionalidades. También podemos seleccionar qué puertos de estos bloques son entradas y cuáles salidas; así como introducir bits como lo deseemos. En la parte superior se encuentran todas las herramientas disponibles para trabajar con Icestudio, para compilar el código una vez creado y cargarlo en la FPGA.

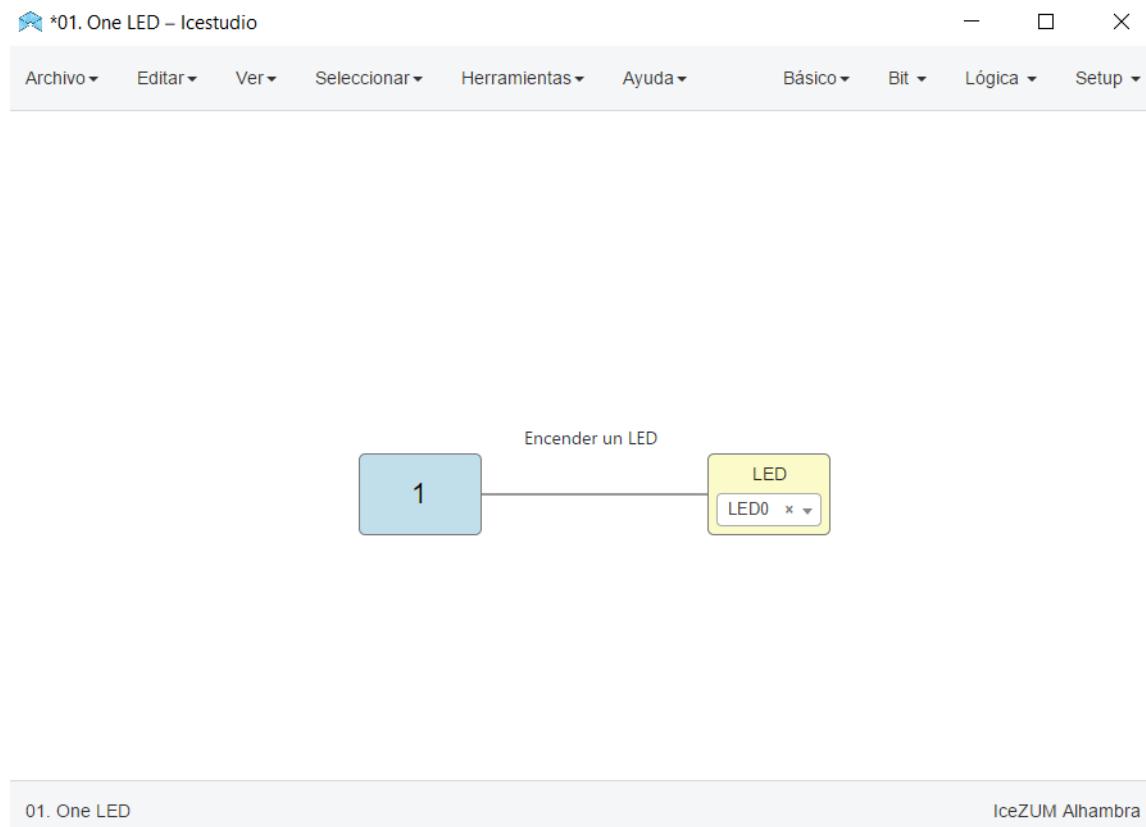


Figura 3.9: Interfaz de Icestudio

## Capítulo 4

# Diseño del sistema

Para realizar el procesamiento más adecuado al sistema, en primer lugar se tendrá que entrenar con una serie de señales obtenidas de prueba con tal fin. Una vez captadas y ajustado el algoritmo, probaremos con una batería de señales en cada brazo que este funcione como lo esperado y cumple con su objetivo. Con estos datos ya no se modificará el algoritmo de procesamiento.

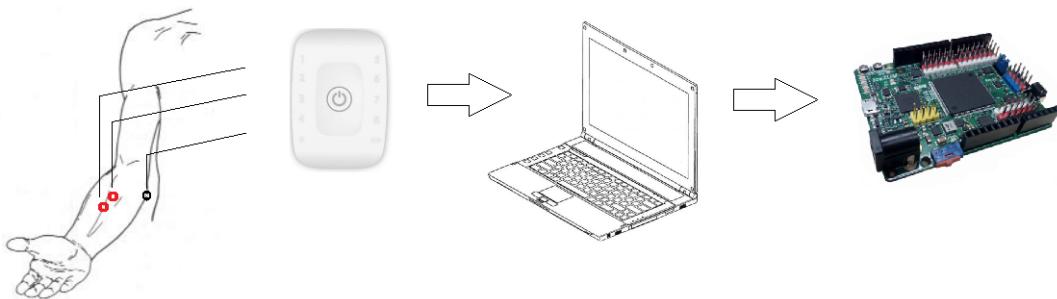


Figura 4.1: Proceso para obtener y procesar bioseñales

En primer lugar se captan las señales de prueba o entrenamiento, que mediante Bluethooth son enviadas al ordenador en tiempo real y recogidas en el software OpenSignals. Estas señales que hemos guardado son posteriormente procesadas en Matlab para su filtrado y captación de parámetros que nos permitan distinguir los distintos estados en los que se encuentra el músculo. El filtrado debe de ser capaz de eliminar el ruido y de acondicionar la señal para que sea más fácil distinguir sus parámetros de bondad.

Una vez las señales se procesan son enviadas mediante comunicación serie a la FPGA, en la que se va a definir el modo de control. Para ello se utilizará la modulación PWM, que como hemos visto en 2.1.3 es la tecnología más barata y utilizada.

En los siguientes capítulos vamos a ver más detenidamente todo el proceso de obtención así como la implementación del proyecto.

### 4.1. Adquisición

En cuanto a la adquisición de señales EMG hay que tener en cuenta un correcto acondicionamiento de la piel y una colocación adecuada de los electrodos de medida.

### 4.1.1. Preparación de la piel

Es necesario a la hora de obtener señales EMG una correcta preparación de la piel dónde vamos a colocar los electrodos; de forma que la calidad de la señal obtenida sea la mejor posible. Para este propósito es recomendable usar un gel abrasivo o alcohol tanto para eliminar las células muertas de la piel como para reducir la sequedad de la misma. Tras la correcta limpieza con un paño suave se asegura que la zona quede totalmente limpia y seca.

### 4.1.2. Los electrodos

Para ello se usan 3 electrodos (2 y uno de referencia) que se colocan directamente sobre la piel y son capaces de captar la actividad bioeléctrica. La ubicación de los electrodos va a influir directamente sobre la calidad de las señales obtenidas; una correcta colocación sería en paralelo a las fibras musculares, en la zona central del músculo del que queramos obtener la actividad eléctrica. Estos deberán estar separados de 1 a 2 cm y el de referencia deberá estar colocado en un sitio dónde sepamos que la actividad será mínima; en los tendones y el borde del músculo las fibras musculares se vuelven más delgadas y pequeñas por lo que son el sitio ideal para colocar el electrodo de referencia.

Antes de colocar los electrodos se puede usar una pasta conductiva que mejora considerablemente la captación de estos de las señales.

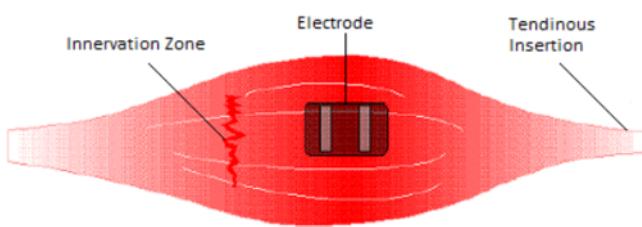


Figura 4.2: Posición de los electrodos

### 4.1.3. Obtención de señales

Con lo visto en 4.1.1 y 4.1.2 tendríamos que definir el músculo que va a hacer de instrumento de control para el sistema. Se plantea la idea de usar un músculo del antebrazo por la facilidad para la colocación de los electrodos, así como a la hora de discriminar los distintos movimientos que nos van a permitir el control del sistema. Un correcto sitio sería con los 2 electrodos en la parte central y el de referencia en el codo que es un punto con poca actividad eléctrica.

El *músculo flexor del carpo* (figura 4.3) será el que eligiremos debido a su posición central, ya que permite contraerlo y relajarlo con facilidad.

Por lo tanto este será el músculo sobre el cual vamos a colocar los electrodos para realizar la toma de señales y empezar a discriminar los distintos movimientos. Las pruebas deberán

hacerse para los dos brazos, ya que nos permitirá más grados de libertad a la hora de elegir el método de control del sistema.

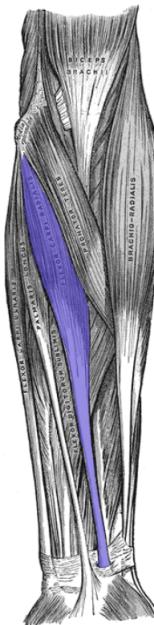


Figura 4.3: Músculo flexor del carpo en el antebrazo

Se tomarán unas señales EMG de entrenamiento, que mediante Bluethooth son enviadas al ordenador en tiempo real y vistas en OpenSignals. En el mismo OpenSignals podemos seleccionar el intervalo de tiempo de la captación que queremos guardar, y nos ofrece distintos formatos para hacerlo: .edf .h5 y .txt. Entre ellos guardaremos las señales como un archivo de texto (.txt), que tendrán el número de muestra y su valor correspondiente.

Estos archivos .txt serán lo que lea Matlab para realizar el procesamiento.

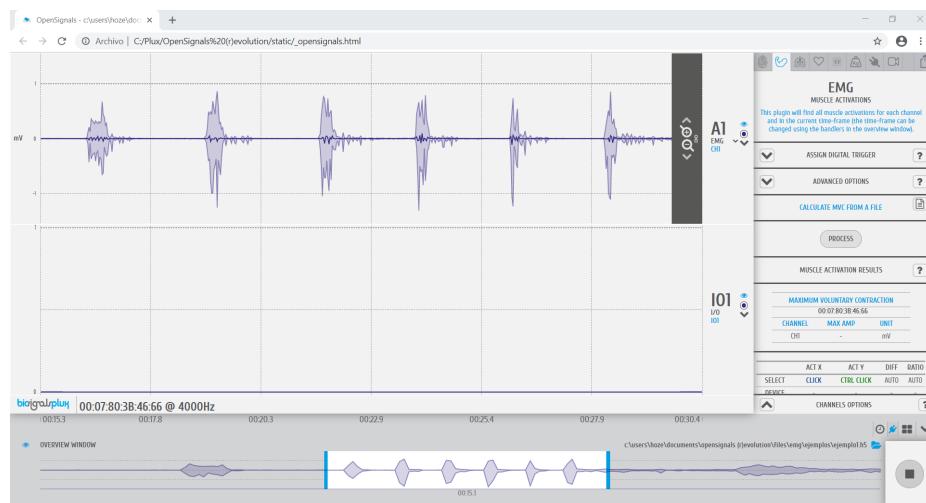


Figura 4.4: Señal captada por OpenSignals

En la figura 4.4 se puede distinguir bastante bien las diferentes contracciones que ha

sufrido el músculo en el proceso de obtención. Las zonas con una menor amplitud se tratan de relajaciones o momentos en los que el flexor se encuentra relajado y con poca actividad eléctrica. En el momento en el que contraemos voluntariamente el músculo se alcanzan picos de gran amplitud; que dependiendo de la fuerza de la contracción serán de mayor o menor intensidad. El algoritmo que creemos tendrá que ser carpaz por lo tanto de discriminar si el músculo se encuentra relajado o contraído. Tras la obtención, un procesamiento con Matlab será necesario para acondicionarlas y preparar las señales para su uso.

#### 4.1.4. Procesamiento en Matlab

A la hora de trabajar con bioseñales para el control de mecanismos robóticos, se necesitan señales sin interferencias. Las interferencias más comunes, como hemos visto en 2.2.1 es el ruido de 50Hz de la red eléctrica; aunque también existen de otro tipo como las que ocurren en la medición, como el movimiento de los cables o incluso los problemas del mismo equipo que hace las medidas.

Para realizar el procesamiento digital de la señal EMG además será necesario tener en cuenta que la mayor parte de su información se encuentra en el rango de frecuencias de 5Hz a 500Hz. Debido a su comportamiento no determinista, debe procesarse con técnicas de caracterización que permitan conocer características definidas de la señal, a través de las cuales aplicar los determinados métodos de control.

Las señales en formato .txt son cargadas en Matlab mediante la función *dlmread*, que lee los datos del fichero y los guarda en una matriz, donde el valor de la frecuencia de muestreo de la señal se almacena en una variable. Hay que tener en cuenta que los valores del fichero no son los valores reales de la señal, son valores muestreados que devuelve el programa; por lo que deben ser convertidos a voltaje usando una expresión matemática que nos facilita el manual de BioSignalplus.

La expresión matemática se define en 4.1:

$$EMG(V) = \frac{((ADC/2^n) - 1/2) * V_{cc}}{G} \quad (4.1)$$

Donde:

- $V_{cc}$ = 3V (Voltaje de operación)
- $G= 1000$  (Ganancia del sensor)
- $EMG(V)$  - Valor de la señal EMG en voltios
- ADC - Valor sampleado del canal que devuelve el programa
- $n$  -Número de bits del canal

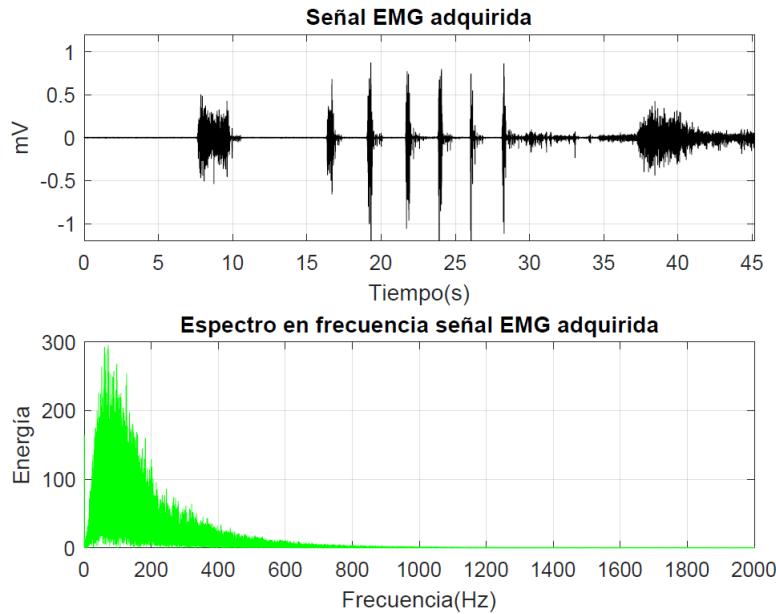


Figura 4.5: Señal captada representada en Matlab

En la figura 4.5 se muestra la señal guardada una vez representada en Matlab en el dominio del tiempo y en el dominio de la frecuencia. Una vez tenemos la señal, si le hacemos la transformada rápida de fourier con el comando *fft* podemos obtener su espectro correspondiente. Para que la señal sea óptima en las siguientes etapas será necesario filtrarla adecuadamente.

#### Filtro de paso de banda EMG: 4Hz a 500Hz

Como se ha comentado las señales EMG tiene su información más relevante en la banda de frecuencia entre 4Hz y 500Hz. Por esa razón, esto se toma como criterio para el diseño de un filtro de paso banda, que puede ser entendido como un filtro pasa alto con frecuencia de corte de 4Hz en serie con un filtro paso bajo con frecuencias de corte de 500 Hz.

Se elige que este filtro sea de Butterworth, ya que permite la respuesta más plana posible. Su función de transferencia es la siguiente:

$$|H(w)|^2 = \frac{1}{1 + (w/wc)^{(2N)}} \quad (4.2)$$

Donde:

- N es el orden del filtro
- wc es la frecuencia de corte
- w la frecuencia compleja

El filtro de manera matemática puede ser complejo de calcular, pero Matlab cuenta con la función *butter* que lo hace automáticamente; especificando las frecuencias de corte y el orden del filtro.

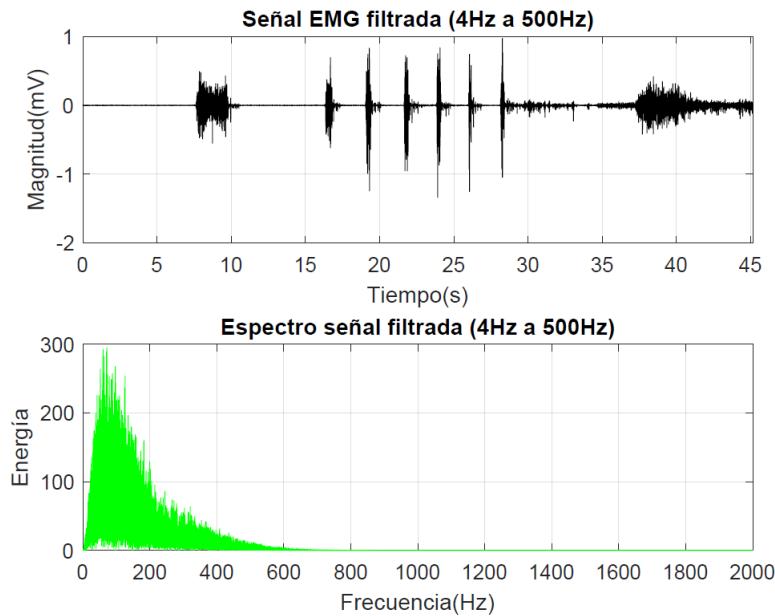


Figura 4.6: Señal con el filtro paso banda

En la figura 4.6 la señal ya quedaría con la información más relevante en la banda 4Hz-500Hz, desechando las altas frecuencias que interfieran en la señal.

#### Filtro de rechazo banda EMG: 50Hz

En España, la frecuencia de la línea eléctrica es de 50Hz, por lo que se requiere el diseño de un filtro rechazo banda que elimine este ruido. Se lleva a cabo el mismo procedimiento que en 4.1.4 para el diseño de un filtro de rechazo banda de 50Hz. Para ello se usa de nuevo la función *butter* pero esta vez con el parámetro *stop* para que sea de rechazo. Esta nueva filtración se añade a la anterior, dejando las señales libres de ruido y acondicionadas para sacar sus parámetros de bondad.

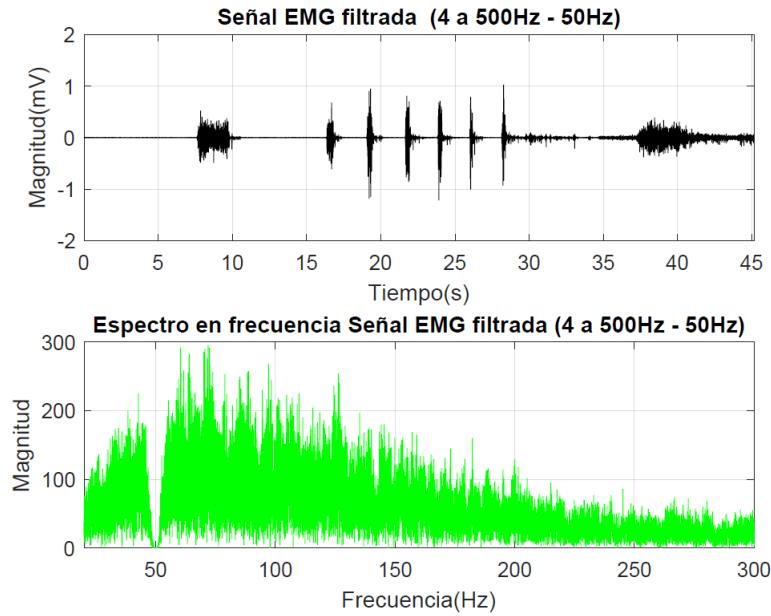


Figura 4.7: Señal con el filtro paso banda y el rechazo banda

Haciendo la comparación de la señal antes y después del filtrado, se puede ver el resultado de los filtros. En la figura 4.8 se representa en rojo la señal original y en negro la señal una vez acondicionada.

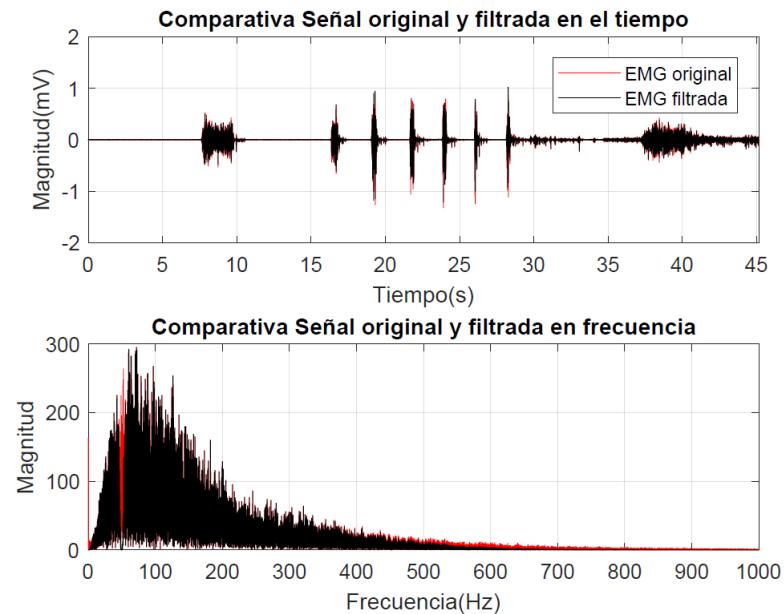


Figura 4.8: Comparativa antes y después del los filtros

### Caracterización de la señal

Una vez la señal está acondicionada adecuadamente, se implementan técnicas de caracterización para sacar parámetros que nos permitan reconocer los distintos movimientos. Se necesitan herramientas para obtener la energía promedio de la señal que nos permitan carac-

terizar los distintos cambios que sufre esta dependiendo de la contracción. La señal EMG es muy oscilante (puede cambiar rápidamente en un milisegundo) por lo que resultaría imposible utilizarla directamente para controlar un sistema robótico. Se necesita un método que permita obtener características de la señal y poder definir patrones o comportamientos de ésta.

Una manera de caracterizar una señal es usando el *método por ventanas* que calcula la envolvente de la señal EMG. Para ello dividimos la señal en segmentos o ventanas de longitud L de 250ms con un solapamiento entre ventanas de 125ms, y caracterizamos usando la *Suma de valores absolutos* o *IAV*. Cuanto menor sea L y más alto el solapamiento, mayor número de ventanas y mejor caracterización; pero el procesamiento será más lento.

La IAV *suma de valores absolutos* realiza la suma de los valores absolutos contenidos en cada una de las secciones o ventanas en las que hemos dividido la señal, para dar una idea aproximada de la intensidad de esa ventana.

La expresión de la IAV se define en 4.3:

$$EMG_{IAV} = \sum_{k=1}^N |EMG_k|, i = 0, \dots, L - 1. \quad (4.3)$$

Esto nos permite distinguir mediante umbrales los grados de intensidad de la señal obtenida, cuanto más tensión se aplique más energía tendrá la señal EMG, lo que se traducirá en un pico más grande de la IAV. Con esto ya podemos decir que si una señal pasa determinado umbral será una contracción fuerte, contracción leve o relajación.

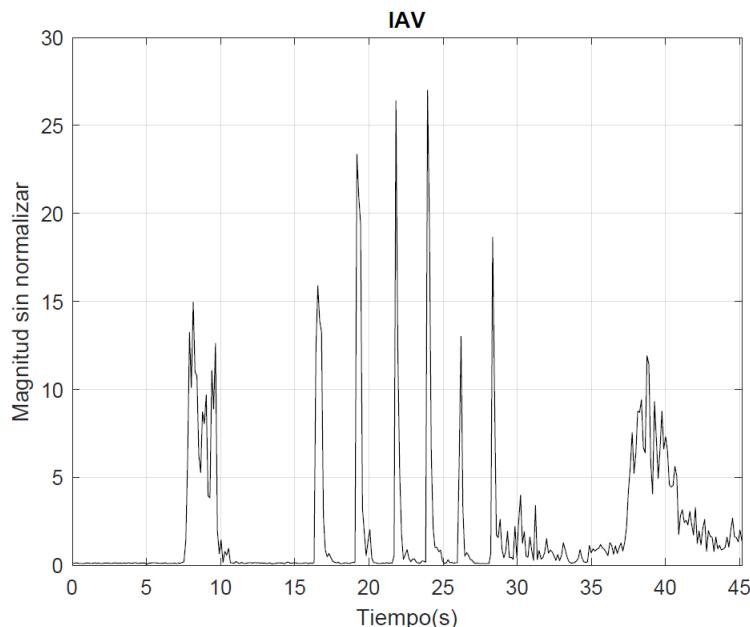


Figura 4.9: Representación de la IAV

### Testeo del algoritmo

Para el testeo del algoritmo se propone usar varias señales tomadas posteriormente, sin modificar el algoritmo de procesamiento. Para ello recogemos una serie de señales de pruebas, en concreto 10 con el brazo derecho y 10 con el brazo izquierdo. Se ha tratado de realizar en todos las señales captadas los mismos movimientos a diferentes velocidades y orden, para ver si el algoritmo es capaz de distinguirlos correctamente.

En principio nos interesa poder identificar en las señales tres movimientos con cada brazo, que tendrán distintos niveles de tensión para el músculo.

- **El músculo relajado.** En el que la contracción, y por lo tanto la IAV será mínima.
- **El músculo contraído levemente.** La IAV presentará mayor amplitud.
- **El músculo contraído y con el brazo girado.** La contracción será máxima y mucho mayor la IAV que en los otros dos movimientos.

En la tabla 4.1 recogemos las pruebas hechas con el algoritmo para las diferentes señales captadas con cada brazo. El algoritmo es capaz de procesar y discriminar cada señal correctamente.

		Relajación	Contracción	Giro
Brazo derecho	Señal 1	OK	OK	OK
	Señal 2	OK	OK	OK
	Señal 3	OK	OK	OK
	Señal 4	OK	OK	OK
	Señal 5	OK	OK	OK
	Señal 6	OK	OK	OK
	Señal 7	OK	OK	OK
	Señal 8	OK	OK	OK
	Señal 9	OK	OK	OK
	Señal 10	OK	OK	OK
Brazo izquierdo	Señal 1	OK	OK	OK
	Señal 2	OK	OK	OK
	Señal 3	OK	OK	OK
	Señal 4	OK	OK	OK
	Señal 5	OK	OK	OK
	Señal 6	OK	OK	OK
	Señal 7	OK	OK	OK
	Señal 8	OK	OK	OK
	Señal 9	OK	OK	OK
	Señal 10	OK	OK	OK

Tabla 4.1: Señales de prueba y resultados

#### 4.1.5. Comunicación serie en Matlab

Una vez procesadas las señales tenemos que pasarlas a la FPGA para el diseño del sistema de control. El *puerto serie* nos abre la posibilidad de comunicar el ordenador con nuestros circuitos en la FPGA, y por lo tanto poder enviar datos de x bits. Se trata de comunicaciones serie asíncronas y se utiliza un único cable de datos para recepción y transmisión. Tras la clasificación de las señales con la IAV ajustamos el rango de datos obtenido de 0 a 255 (8bits) en Matlab y estos se envían mediante el puerto serie (COM7) a la FPGA.

Para ello será necesario abrir el puerto serie del ordenador con el comando *fopen*, enviar los datos con *fwrite* y definir la velocidad de transmisión, que será para toda la transmisión y recepción de 115200 baudios.

#### 4.2. Diseño en icestudio

#### 4.2.1. Receptor serie

La FPGA icezum Alhambra que vamos a utilizar ofrece a través del USB una interfaz de comunicaciones serie síncronas (por la que se realiza la carga del programa creado en Icestudio a la placa) y una interfaz de puerto serie. Por lo tanto para realizar la conexión entre PC-FPGA bastaría con conectarlos mediante un USB para tener listo el hardware de envío de datos. No solo es necesario abrir el puerto serie por Matlab, también la FPGA necesita para recibirlos un receptor serie, que habrá que crearlo en Icestudio.

El esquema del receptor de bits mediante el puerto serie planteado se muestra en la figura 4.10:

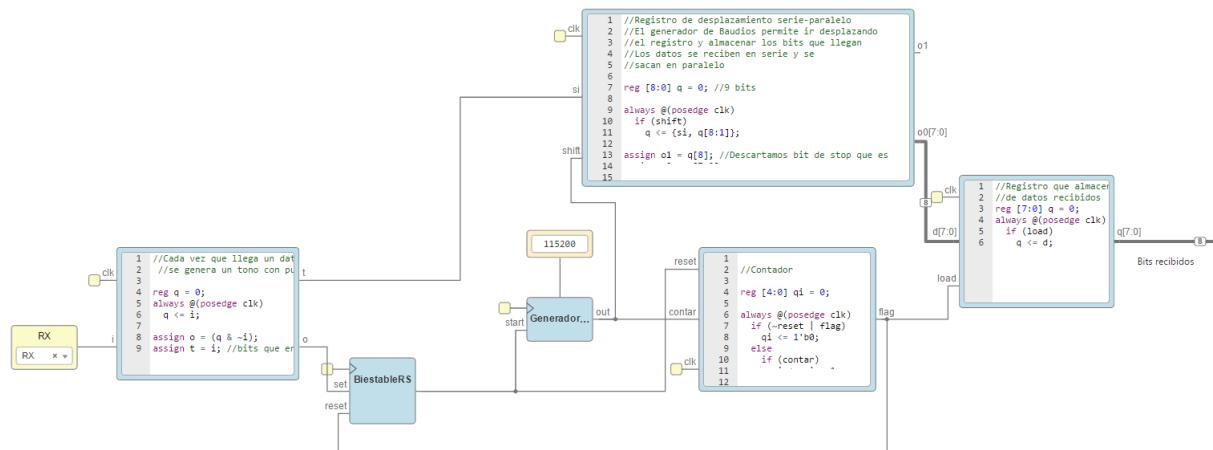


Figura 4.10: Receptor serie en icestudio

Los bits deben de llegar por el puerto serie y salir por un bus de 8 bits en paralelo para poder utilizar los datos que llegan.

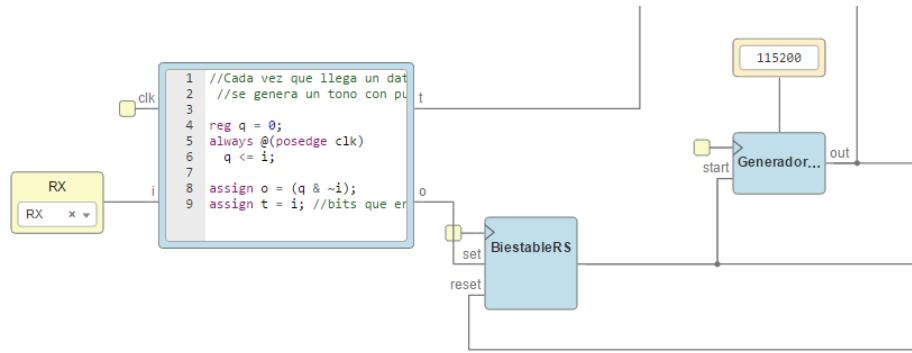


Figura 4.11: Primera parte del receptor serie

Por RX llegan los bits en tramas en serie, primero el bit de start, después los bits empezando por el menos significativo y luego el bit de stop. Cada vez que llega un bit de start (que una trama es enviada) se genera un tono puro que activa un biestable RS; el cuál se reseteará para recibir un nuevo dato. Éste a su vez activa un generador de baudios a la velocidad de recepción serie de 115200 baudios.

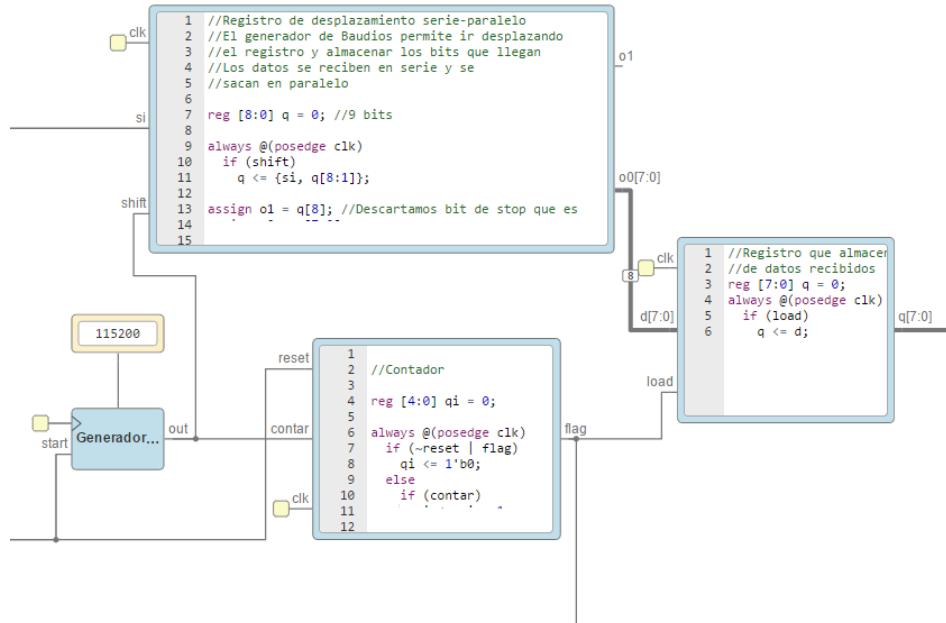


Figura 4.12: Segunda parte del receptor serie

Con el generador de baudios se generan los tics para leer los bits serie que van llegando a una velocidad determinada. La salida del generador va a un registro de desplazamiento

serie-paralelo y a un contador. El generador de Baudios permite ir desplazando el registro y almacenar los bits que llegan. Los datos se reciben en serie y se sacan en paralelo. A su vez que el generador de baudios genera un tic se incrementa el contador. Al llegar a 9 bits (8 datos y 1 stop- el de start solo es para arrancar) significa que ya ha llegado el dato completo y se activa el flag que reinicia el biestable RS y se queda listo para recibir un nuevo dato.

#### 4.2.2. Modulador de PWM

Como hemos visto en 2.1.3 el dron necesita recibir las señales mediante un protocolo de comunicaciones. Por lo tanto habrá que adaptar las señales recibidas en la FPGA a una modulación PWM.

La Modulación por anchura de pulsos nos va a permitir generar señales analógicas a partir de sistemas digitales. Esto se puede usar por ejemplo para mover un motor a diferentes velocidades, o cambiar la intensidad con la que brilla un led. Para nuestra aplicación nos va a permitir enviarle las órdenes al dron, y definir un sistema de control en base a las señales que hemos obtenido con el receptor serie. 4.2.1

Para ello se ha creado el bloque PWM de la figura 4.13



Figura 4.13: Bloque PWM

Para cada entrada de datos de 8 bits se genera una señal de PWM en la salida. La estructura del bloque PWM es la siguiente:

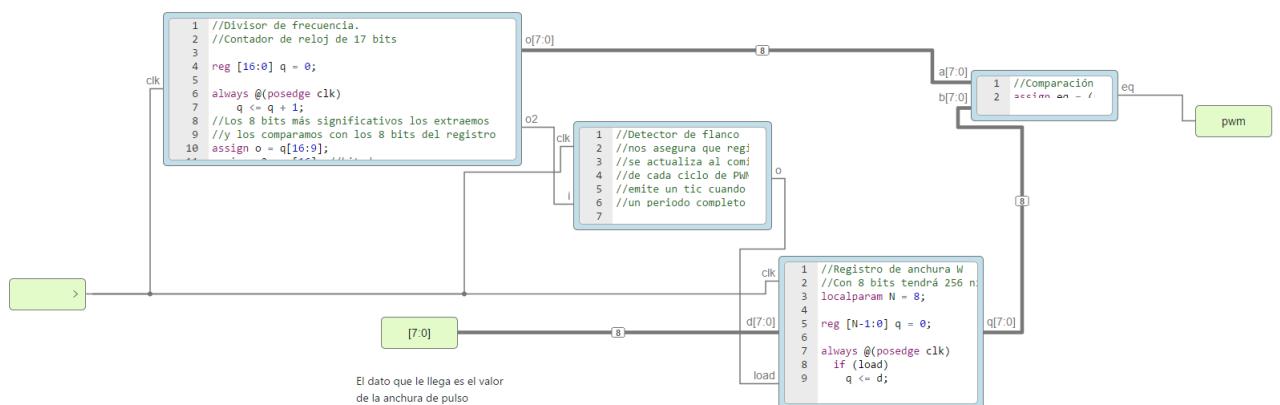


Figura 4.14: Estructura del Bloque PWM

Ésta cuenta con 2 elementos principales:

- **Un contador de reloj.** El contador tiene N bits. N va a determinar la frecuencia y tendremos que elegirla por lo tanto antes de el diseño del contador
- **Un registro de anchura.** El valor de anchura de pulso será el dato que llega por la entrada; es la información que se recibe del sistema de adquisición.

La frecuencia que tenga la señal PWM por lo tanto, podremos obtenerla a partir de la señal de reloj del sistema de 12 MHz con un divisor de frecuencia. Para ello tenemos que tener en cuenta la frecuencia que deseemos, que se presentan en la tabla 4.2.

Frecuencia	Reloj(S)
12 MHZ	S
6 MHZ	S/2
3 MHZ	S/4
1.5 MHZ	S/8
750 KHZ	S/16
375 KHZ	S/32
187.5 KHZ	S/64
93.8 KHZ	S/128
46.9 KHZ	S/256
23.4 KHZ	S/512
11.7 KHZ	S/1024
5.9 KHZ	S/2048
2.9 KHZ	S/4096
1.5 KHZ	S/8192
732 Hz	S/16384
366 Hz	S/32768
183 Hz	S/65536
92 Hz	S/131072

Tabla 4.2: Frecuencias y Reloj del sistema

Si queremos por ejemplo una señal PWM de 92 Hz, tenemos que usar un contador de 17 bits 4.4. Esta corresponde a un periodo de 11ms aproximadamente, que será el que usemos para nuestra aplicación.

$$Bits(N) = \frac{Reloj\,del\,sistema(12\,Mhz)}{131072} = 17\,bits. \quad (4.4)$$

En el divisor de frecuencias los 8 bits más significativos se extraen y estos se comparan con los 8 bits que llegan del registro de anchura W. Esta comparación es lo que permite generar los ciclos de PWM.

- Si el contador es **menor** que el registro de anchura. La salida será 1.
- Si contador es **mayor** que el registro de anchura . La salida será 0.



# Bibliografía

- [1] Victor Delgado. Historia de los drones. <http://eldrone.es/historia-de-los-drones/>, 2016. Accessed 19-06-2019.
- [2] IWM Staff. A brief history of drones. <https://www.iwm.org.uk/history/a-brief-history-of-drones/>, 2018. Accessed 25-06-2019.
- [3] Droneymas. Tipos de drone en el mercado. <https://www.droneymas.com/tipos-de-drone/>, 2019. Accessed 25-08-2019.
- [4] Amine Naït-Ali. *Advanced biosignal processing*. Springer Science & Business Media, 2009.
- [5] Balambigai Subramanian. Ecg signal classification and parameter estimation using multiwavelet transform. 2017.



