



SISTEMA ELECTROMIOGRÁFICO (EMG) DE TELECONTROL PARA PACIENTES CON MOVILIDAD REDUCIDA

Pedro Velasco del Val

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Sistemas Encastados

Jordi Bécares Ferrés
Pere Tuset Peiró

11/06/2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Sistema electromiográfico (EMG) de telecontrol para pacientes con movilidad reducida</i>
Nombre del autor:	<i>Pedro Velasco del Val</i>
Nombre del consultor/a:	<i>Jordi Bécares Ferrés</i>
Nombre del PRA:	<i>Pere Tuset Peiró</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Área del Trabajo Final:	Sistemas Encastados
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>EMG, Telecontrol, IoT</i>

Resumen del Trabajo (máximo 250 palabras):

Para las personas con movilidad reducida, y más aún con inmovilización completa, es muy difícil hacer tareas sencillas, como mover un vehículo o interactuar con un dispositivo remoto, pero normalmente son capaces de mover algunos músculos, lo que se puede aprovechar con un sistema mioeléctrico.

Un sistema mioeléctrico o electromiográfico (EMG) es aquel capaz de detectar, mediante el uso de electrodos, los impulsos eléctricos que generan los músculos al contraerse y relajarse, para después tratar esos impulsos y realizar acciones en base a las señales obtenidas.

En este proyecto se plantea una solución de telecontrol en base a electrodos, un sistema de detección EMG y un módulo MSP432P401R para generar diferentes mensajes que servirán para interactuar con un dispositivo, por ejemplo, un vehículo, usando los movimientos de los músculos faciales.

Estos mensajes se enviarán a un servidor web al que se puede acceder desde un cliente web, creando un sistema empotrado y conectado.

El resultado es un sistema empotrado pequeño y portable, capaz de detectar 6 movimientos faciales y enviarlos a un servidor al que se puede acceder desde cualquier dispositivo para visualizarlos a través de un navegador web o para suscribirse mediante el protocolo MQTT y utilizar dichos movimientos para cualquier aplicación.

Abstract (in English, 250 words or less):

For people with reduced mobility, and even more with complete immobilization, it is very difficult to do simple tasks, such as moving a vehicle or interacting with remote devices, but they are usually able to move some muscles, which can be exploited with a myoelectric system.

A myoelectric or electromyographic system (EMG) is one capable of detecting, by means of electrodes, the electrical impulses generated by the muscles when contracting or relaxing, and treating those impulses to perform actions based on the obtained signals.

In this project, a telecontrol solution based on electrodes, an EMG detection system and an MSP432P401R module is proposed to generate different messages that will be used to interact with a device, for example a vehicle, using the movements of the facial muscles.

These messages will be sent to a web server that can be accessed from a web client, creating an embedded and connected system.

The result is a small and portable embedded system, capable of detecting 6 facial movements and sending them to a server that can be accessed from any device to view them through a web browser or to subscribe using the MQTT protocol and use these movements for any application.

Índice

1.	Introducción	1
1.1.	Contexto y justificación del trabajo	1
1.2.	Descripción del trabajo	1
1.3.	Objetivos del TFG.....	2
1.1.1.	Principales.....	2
1.1.2.	Secundarios	2
1.4.	Enfoque y método seguido.....	2
1.5.	Planificación del trabajo.....	3
1.5.1.	Lista de tareas.....	3
1.5.2.	Diagrama de Gantt.....	4
1.5.3.	Comparativa de planificación inicial-final.....	5
1.6.	Recursos empleados.....	6
1.7.	Productos obtenidos.....	6
1.8.	Descripción de los capítulos de la memoria.....	6
2.	Antecedentes.....	8
2.1.	Estado del arte.....	8
2.1.1.	Señales electromiográficas	8
2.1.2.	Sistemas embebidos	9
2.1.3.	Tecnologías de comunicación	10
2.1.4.	Protocolos IoT	11
2.2.	Estudio de mercado.....	12
3.	Descripción funcional.....	13
3.1.	Sistema electromiográfico (EMG) de telecontrol para pacientes con movilidad reducida	13
3.1.1.	Requerimientos	13
3.1.2.	Diagrama de bloques del sistema	14
3.1.3.	Diagrama de bloques de la aplicación.....	14
3.1.4.	Sistema de comunicación	15
3.2.	Aplicación web.....	18
3.2.1.	Diagrama de bloques	18
3.3.	Dispositivo Hardware.....	18
3.3.1.	Diseño Hardware.....	21
4.	Descripción detallada.....	23
4.1.	Diseño hardware.....	23
4.1.1.	Diseño general	23
4.1.2.	MSP432	24
4.1.3.	Módulo CC3100	24
4.1.4.	Circuito sensor EMG	25
4.1.5.	Tratamiento de la señal por hardware.....	26
4.1.5.1.	Rechazo de alta frecuencia.....	26
4.1.5.2.	Amplificador de instrumentación	27
4.1.5.3.	Filtro paso alto.....	28
4.1.5.4.	Amplificador de tensión.....	29
4.1.6.	Alimentación del circuito.....	31
4.2.	Diseño software	31
4.2.1.	Diseño software del dispositivo	31
4.2.2.	Filtrado por software de la señal	32
4.2.1.	Cálculo de umbrales.....	34

4.2.2.	Algoritmo de detección de movimientos.....	34
4.2.3.	Comunicaciones.....	35
4.2.4.	Diseño de la aplicación web.....	37
5.	Viabilidad técnica.....	41
6.	Valoración económica.....	44
6.1.	Prototipo.....	44
6.2.	Industrialización.....	45
7.	Conclusiones.....	46
7.1.	Lecciones aprendidas.....	46
7.2.	Autoevaluación.....	46
7.2.1.	Tiempo dedicación a cada PEC.....	47
7.2.2.	Logro de objetivos.....	47
7.2.3.	Seguimiento de la planificación.....	48
7.3.	Líneas de trabajo futuro.....	48
8.	Glosario.....	50
9.	Bibliografía.....	51

Lista de figuras

Figura 1 - Diagrama de Gantt.....	4
Figura 2 - Días festivos.....	5
Figura 3 - Diagrama de Gantt final.....	5
Figura 4 - Ejemplo de uso 1.....	8
Figura 5 - Ejemplo de uso 2.....	9
Figura 6 - Ejemplo de uso 3.....	9
Figura 7 - Ejemplo de uso 4.....	9
Figura 8 - Diagrama de bloques del sistema.....	14
Figura 9 - Diagrama de bloques de la aplicación.....	14
Figura 10 - Esquema de red.....	16
Figura 11 - Diagrama de bloques de la comunicación.....	17
Figura 12 - Esquema software de la aplicación web.....	18
Figura 13 - Módulo WiFi CC3100.....	19
Figura 14 - Módulo WiFi ESP-01.....	19
Figura 15 - Pololu Muscle Sensor.....	20
Figura 16 - Myoware Muscle Sensor.....	20
Figura 17 - Grove EMG Detector.....	21
Figura 18 - Mikroe EMG click.....	21
Figura 19 - Esquema eléctrico del Hardware.....	22
Figura 20 - Esquema eléctrico del sensor Grove EMG Detector.....	22
Figura 21 - Montaje real del hardware.....	23
Figura 22 - MSP432P401R junto al módulo CC3100.....	24
Figura 23 - Posición final de los electrodos.....	26
Figura 24 - Etapa de rechazo de alta frecuencia.....	26
Figura 25 - Etapa de amplificador instrumental.....	28
Figura 26 - Etapa de filtro paso alto.....	29
Figura 27 - Etapa de amplificación de tensión.....	30
Figura 28 - Caso de uso del software del dispositivo.....	32
Figura 29 - Filtrado de señal.....	33

Figura 30 - Movimiento filtrado	34
Figura 31 - Esquema de comunicación Publish/Subscribe.	35
Figura 32 - Estructura de los mensajes MQTT	36
Figura 33 - Caso de uso de la aplicación web	37
Figura 34 - Flujo de Movimientos en NodeRed	38
Figura 35 - Flujo de Estado en NodeRed	39
Figura 36 - Interfaz gráfica	40
Figura 37 - Gráfica de movimiento del bíceps	41
Figura 38 - Gráfica de movimiento de la mejilla inicial	42
Figura 39 - Gráfica de movimiento de mejilla mejorado	42

Lista de tablas

Tabla 1 - Lista de tareas.....	3
Tabla 2 - Tipos de mensaje MQTT	36
Tabla 3 - Valoración de riesgos y contramedidas.....	43
Tabla 4 - Presupuesto del prototipo	44
Tabla 5 - Presupuesto de industrialización para 1000 unidades	45
Tabla 6 - Tiempo de dedicación	47

1. Introducción

1.1. Contexto y justificación del trabajo

La tecnología debe utilizarse siempre que sea posible para ayudar a las personas con más dificultades, consiguiendo con ello un mundo más accesible y justo para todos.

Para las personas con movilidad reducida, y más aún con inmovilización completa, es muy difícil hacer tareas sencillas, como mover un vehículo o controlar un dispositivo que requiera del uso de sus extremidades para ejecutar las distintas acciones.

En la actualidad, hay multitud de sistemas que permiten interactuar con dispositivos a través de botones, palancas, etc., pero para accionarlos es necesario que el usuario pueda mover al menos una extremidad.

Esto no es posible para las personas tetrapléjicas, ya que no pueden mover ninguna extremidad, pero normalmente son capaces de mover algunos músculos, como por ejemplo los músculos faciales, lo que se puede aprovechar con un sistema mioeléctrico para detectar dicho movimiento y utilizarlo para enviar órdenes a un dispositivo.

1.2. Descripción del trabajo

El proyecto consiste en un sistema electromiográfico (EMG) que detecta las señales eléctricas que generan los músculos al moverse y en base a ellas se realizan acciones mediante un microcontrolador.

Una persona tetrapléjica no puede mover más que los músculos de la cara, lo que dificulta enormemente su interacción con cualquier dispositivo. Por ello el sistema detectaría esos movimientos y si considera que se trata del movimiento adecuado, generará una orden en función del tipo de movimiento detectado.

El objetivo de este proyecto es detectar al menos 5 movimientos faciales diferentes, que serían los mínimos necesarios para poder controlar un sistema motriz, y enviar las ordenes correspondientes a un servidor remoto, donde se podrán visualizar en tiempo real a través de un cliente web.

Una posible utilidad sería usar los movimientos faciales detectados para mover un vehículo a distancia. También podría usarse para controlar un brazo robótico en una fábrica, aunque la implementación de dichos dispositivos queda fuera del alcance de este proyecto, que se limitará a la detección, clasificación, envío y visualización de los diferentes movimientos detectados.

Para hacerlo posible se utilizarán diferentes tecnologías para la captura de señales mediante sensores EMG, su tratamiento y clasificación en diferentes movimientos utilizando un módulo MSP432P401R, y envío inalámbrico a través de un módulo CC3100 de mensajes bajo el protocolo MQTT a un servidor remoto implementado con Node-Red.

1.3. Objetivos del TFG.

El objetivo global de este trabajo es crear un sistema capaz de detectar el movimiento muscular, generar órdenes de movimiento en base a esas detecciones y enviar dichas órdenes a un servidor donde se podrán visualizar mediante un cliente web.

Para lograr el objetivo global, se establecen los siguientes objetivos principales y secundarios.

1.1.1. Principales

- Detección de movimientos faciales vía EMG
- Caracterización y diferenciación de hasta 5 movimientos. (Adelante, atrás, izquierda, derecha y parar)
- Comunicación inalámbrica vía wifi
- Servidor de control y recepción de datos
- Sistema de envíos de alarma en caso de accidente
- Interfaz gráfica basada en cliente web
- Bajo consumo

1.1.2. Secundarios

- Acceso desde teléfono móvil
- Alimentación con baterías
- Pequeño y portable
- Sistema motor remotamente controlado

1.4. Enfoque y método seguido.

Como se verá en capítulos posteriores, no existe actualmente ningún equipo comercial que realice las funciones de este sistema, por lo que el enfoque escogido ha sido el de crear un producto nuevo en vez de mejorar uno existente.

La estrategia seguida ha sido la siguiente:

- Análisis de necesidades del sistema
- Elección y adquisición de componentes principales
- Comprobación de la viabilidad de los componentes elegidos
- Diseño del prototipo
- Adquisición del resto de materiales
- Montaje del hardware del prototipo
- Programación del software

- Pruebas
- Documentación

1.5. Planificación del trabajo.

1.5.1. Lista de tareas

Fase	Nº	Tarea	Descripción	Horas	Total
Fase 1	1	Montaje del hardware	Realizar el montaje del hardware del proyecto	5	80
	2	Programación del software	Programar las partes necesarias para esta fase	40	
		Configuración y lectura de ADCs	Para los 2 sensores		
		Detección de movimiento básica	Detección de al menos 2 movimientos		
	3	Configuración de electrodos	Buscar la zona más adecuada para colocar los electrodos	5	
	4	Pruebas en local	Realizar pruebas a través del puerto serie y exportación del workspace	20	
	5	Documentación	Realizar el informe de la PEC2	10	
Fase 2	6	Programación del software	Programar las partes necesarias para esta fase	40	80
		Filtrado de señal	Filtrar la señal para mayor calidad		
		Detección de movimiento avanzada	Detectar los 5 movimientos necesarios		
		Clasificación de movimientos	Clasificar los distintos movimientos		
	7	Pruebas en local	Realizar pruebas a través del puerto serie y exportación del workspace	20	
	8	Añadir WiFi	Integrar el módulo CC3100	10	
	9	Documentar	Realizar el informe de la PEC3	10	
Fase 3	10	Programación del software	Programar las partes necesarias para esta fase	40	85
		Configuración de mensajería WiFi	Conexión a red y paso de mensajes		
		Configuración del servidor web	Implementar el sistema para notificaciones del usuario		
	11	Pruebas en remoto	Probar el funcionamiento del sistema completo en remoto y exportación del workspace.	20	
	Opc	Alimentación con baterías	Sustituir la alimentación a través de USB por baterías que duren al menos 1 día	5	
	Opc	Acceso desde móvil	Adaptar y probar acceso desde teléfono móvil.	5	
	Opc	Sistema motor	Sistema motor remotamente controlado	5	
	12	Documentar	Realizar el informe del código final	10	

Tabla 1 - Lista de tareas

1.5.2. Diagrama de Gantt

En la Figura 1 se muestran todas las tareas a realizar, teniendo en cuenta 25 horas semanales de dedicación (5 horas de lunes a viernes) y fines de semana libres.

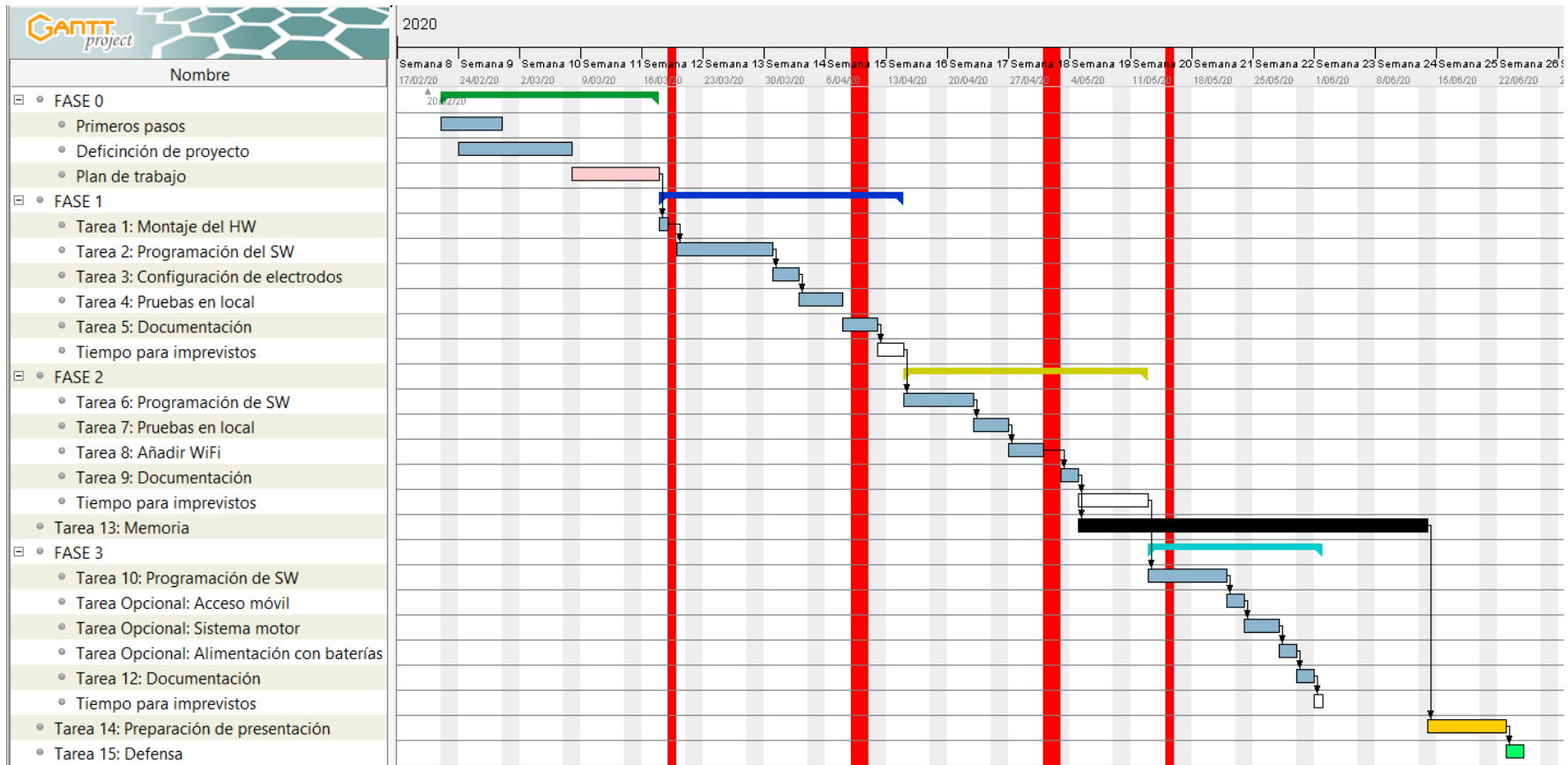


Figura 1 - Diagrama de Gantt

Para la elaboración del diagrama se han tenido en cuenta también los festivos indicados en la Figura 2:

Fechas	Resumen	Tipo	Color
9/04/20	Jueves Santo	Festivo	■
10/04/20	Viernes Santo	Festivo	■
19/03/20	Día del Padre	Festivo	■
1/05/20	Día del Trabajador	Festivo	■
15/05/20	San Isidro	Festivo	■
2/05/20	Día C.Madrid	Festivo	■

Figura 2 - Días festivos

1.5.3. Comparativa de planificación inicial-final

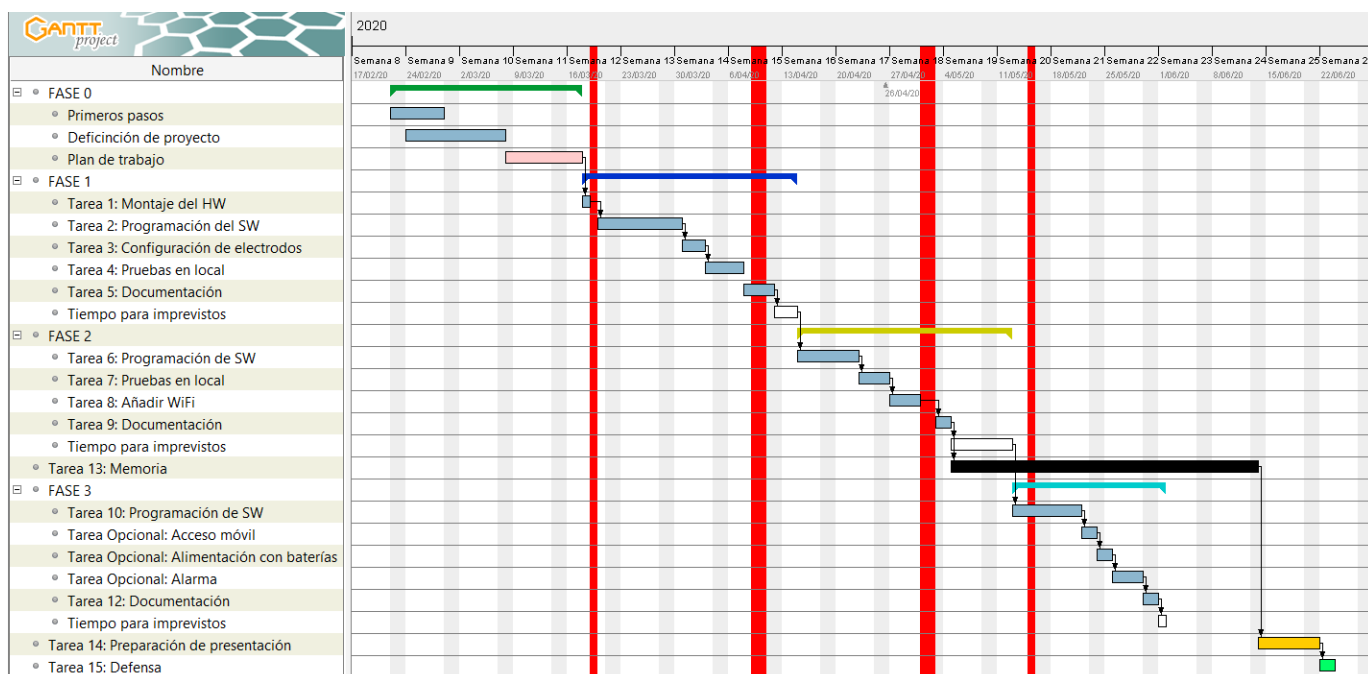


Figura 3 - Diagrama de Gantt final

Como se puede observar en la Figura 3, la planificación final varía ligeramente frente a la inicial, ya que por falta de tiempo no se ha implementado la tarea opcional del sistema motor.

No ha sido posible construir el sistema motor ya que no se han recibido las piezas a tiempo, en parte debido a que, al tratarse de una tarea opcional, no se pidieron las piezas hasta tener el resto de las tareas casi realizadas, y no se tuvo en cuenta que debido a la situación actual por el COVID-19 tardan más de lo normal en llegar.

Como contrapartida, en el tiempo que se iba a dedicar se ha implementado otra tarea opcional, un sexto tipo de movimiento para indicar una alarma.

1.6. Recursos empleados

Los materiales que se han utilizado son:

- Placa desarrollo MSP432P401R de Texas instrument
- Modulo CC3100 BoosterPack de Texas instrument
- Sensor Grove EMG Detector de SeedStudio (x2)
- PC Servidor
- Cables
- Porta-baterías
- Pilas AA de 1.2V/1.5V

Los IDEs que se utilizan para programar son:

- Energía (Lenguaje Arduino/C)
- Node-Red y Node-Red Dashboard (Programación visual)

1.7. Productos obtenidos

El producto final consiste en un sistema remoto de detección de 5 movimientos faciales y un sexto movimiento extra para enviar una alarma, y que puede ser utilizado para múltiples propósitos gracias a su interfaz web, así como a la documentación de todo el sistema mediante esta memoria.

1.8. Descripción de los capítulos de la memoria.

Los diferentes capítulos en los que se estructura esta memoria son:

Capítulo 1 – Introducción: En este capítulo se describe la motivación del proyecto, sus objetivos, el enfoque empleado y la planificación a seguir para su desarrollo.

Capítulo 2 – Antecedentes: En este capítulo se describe el estado del arte y se realiza un estudio del mercado actual sobre las tecnologías empleadas y su presencia en el mercado, analizando si existen sistemas similares en el mismo.

Capítulo 3 – Descripción funcional: En este capítulo se trata el diseño del sistema. Se dará una idea general del proyecto, proporcionando una visión global del sistema y su funcionamiento, que se ampliará en el siguiente capítulo.

Capítulo 4 – Descripción detallada: En este capítulo se explican técnicamente los diferentes apartados del capítulo anterior de forma detallada, incluyendo las características técnicas del hardware y el software.

Capítulo 5 – Viabilidad técnica: En este capítulo se realiza un análisis de la viabilidad del proyecto, indicando si la realización de este es posible con las tecnologías escogidas o si se prevé algún tipo de problema.

Capítulo 6 – Valoración económica: En este apartado se incluye un presupuesto tanto del desarrollo del prototipo como de su industrialización.

Capítulo 7 – Conclusiones: Valoración del trabajo realizado, lecciones aprendidas y posibles mejoras detectadas en el transcurso del proyecto.

Capítulo 8 – Glosario: Definición de términos y acrónimos más relevantes utilizados en la realización de esta Memoria.

Capítulo 9 – Bibliografía: Lista de referencias bibliográficas utilizadas para la realización del proyecto.

2. Antecedentes

2.1. Estado del arte

Los sistemas de telecontrol han evolucionado desde el mando a distancia, que se popularizó con las televisiones, hasta el control por voz de los asistentes personales que cada día se van incorporando en los hogares.

Para una persona tetrapléjica, estos sistemas son inoperables puesto que no pueden mover dedos, brazos, piernas ni incluso hablar en muchas ocasiones, por lo que se requieren sistemas diferentes.

2.1.1. Señales electromiográficas

Las señales electromiográficas (EMG) son señales eléctricas producidas por un músculo durante el proceso de contracción y relajación. [1]

La electromiografía, dedicada al estudio de dichas señales, se comenzó a usar a principios del siglo XX, aunque las primeras referencias a estas señales se encuentran en el de trabajo de Francesco Redi en 1666, quien descubrió que la raya eléctrica poseía un músculo que generaba electricidad. [2]

El término de electromiografía fue introducido por el doctor y fotógrafo francés E. J. Marey, quien en 1890 realizó el primer registro de actividad eléctrica muscular. Alrededor del año 1912, el investigador H.Piper realizó en Alemania las primeras mediciones de señales EMG con un galvanómetro y en 1924, los investigadores Gasser y Erlanger realizaron estudios similares con un osciloscopio. [3]

Poco a poco, muchos otros investigadores se unieron a ellos en el estudio de las señales generadas por los músculos. Con la aparición de los ordenadores, el análisis de la señal EMG mejoró mucho, creándose múltiples modelos de análisis de dicha señal que ayudaron a ampliar su conocimiento y su uso.

Actualmente, las aplicaciones de las señales EMG son muy variadas, por ejemplo:

- Para la fabricación de prótesis mioeléctricas de extremidades (sobre todo de manos), permitiendo la activación de estas con pequeños movimientos musculares en vez de grandes movimientos corporales, como las prótesis tradicionales. Podemos ver un ejemplo en la Figura 4 - Ejemplo de uso 1. [4]



Figura 4 - Ejemplo de uso 1

- Para el entrenamiento deportivo, detectando la activación de los músculos implicados en el ejercicio que se está realizando y analizando si dicha activación se realiza en el orden y con la intensidad adecuada al movimiento deseado. Podemos ver un ejemplo en la Figura 5. [5]



Figura 5 - Ejemplo de uso 2

- Para el análisis de fatiga muscular, ayudando en el diagnóstico de enfermedades como el síndrome de fatiga crónica. Podemos ver un ejemplo en la Figura 6. [6]



Figura 6 - Ejemplo de uso 3

- Para la rehabilitación muscular, ayudando a planificar los programas de entrenamiento muscular para una lesión y/o patología específica durante la rehabilitación. Podemos ver un ejemplo en la Figura 7. [7]



Figura 7 - Ejemplo de uso 4

2.1.2. **Sistemas embebidos**

Un sistema embebido (o empotrado) es aquel diseñado y optimizado para una (o varias) tarea concreta.

“Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas” [8]

Los sistemas embebidos normalmente integran una serie de componentes en un único dispositivo (microprocesador, memoria, software, conectores de entrada/salida, etc) y a su vez permiten su interconexión con otros dispositivos complementarios a través de distintos protocolos. Todo esto, además, en tiempo real.

El primer sistema embebido se considera que fue usado a principios de los años 60 por la NASA en el “Apollo Guidance Computer” del programa Apolo [9]. Este sistema estaba programado en lenguaje ensamblador y usaba palabras de 16 bits, pese a que en aquella época muchas computadoras científicas ya usaban palabras de 24 bits para tener mayor exactitud en los cálculos, pero para mantener un compromiso entre exactitud y simplicidad del circuito optaron por los 16 bits. Su propósito era controlar la orientación y navegación del módulo de mando y el módulo lunar. [10]

Tras este, el siguiente fue usado para el guiado del misil balístico intercontinental “LGM-30 Minuteman”, que entró en servicio en 1962. El sistema embebido tenía como propósito en este caso el guiado inercial del misil, siendo además un sistema reprogramable, algo que el ejército llevaba años intentando lograr sin éxito. [11]

Posteriormente surgieron muchos otros sistemas que se empezaron a producir en masa. Uno de los primeros fue el TMS 1000, de Texas Instruments, desarrollado en 1971 y puesto a la venta en 1974, que integraba un microprocesador de 4 bits, memoria ROM y RAM. Poco después, en 1975, Microchip Technology lanzó al mercado sus microcontroladores PIC, que gracias a su bajo coste y alta disponibilidad se hicieron muy populares y se programaban tanto en ensamblador como en C.

Más delante, en 2005, aparece el primer modelo de Arduino, un sistema embebido muy popular en la actualidad gracias a su versatilidad, bajo precio y sobre todo a su facilidad de programación mediante el lenguaje Processing, similar a C++ y adaptado a sus microcontroladores ARM Cortex M3 [12].

En 2012, después de una serie de diseños de prueba la Raspberry Pi Foundation lanzó la Raspberry Pi 1, con potencia suficiente para ejecutar un sistema Linux en su interior, muy utilizadas también actualmente gracias a su buena relación potencia-coste [13].

Entre todas las opciones disponibles actualmente, también destaca la familia de microcontroladores de bajo consumo y altas prestaciones MSP430, desarrolladas alrededor de microcontroladores ARM Cortex M4. Esta familia, además de ser más potente que Arduino con un consumo energético más bajo que el de Raspberry Pi, se puede programar tanto en lenguaje Processing como a bajo nivel (en lenguaje C) si la aplicación lo requiere, lo que la hace muy versátil a la vez que potente.

2.1.3. Tecnologías de comunicación

Todo sistema embebido está diseñado para interactuar ya sea con humanos o con otros dispositivos, y esa comunicación entre dispositivos se puede hacer a través de una red, lo que da lugar al mundo del IoT, el internet de las cosas, donde cualquier objeto, desde un frigorífico a una prenda de ropa, puede conectarse a internet. [14]

IoT está compuesto por multitud de tecnologías que permiten la interacción de dispositivos, entre las que destacan:

- **RFID:** Tecnología utilizada para transmitir la identidad de un objeto utilizando ondas de radio. Es de muy bajo alcance y baja tasa de transferencia, pero con un coste y consumo muy bajos.
- **Bluetooth:** Tecnología inalámbrica muy extendida, con una tasa de transferencia baja y con alcance limitado, pero muy bajo consumo de energía. Ambos dispositivos deben disponer de esta tecnología para poder comunicarse.
- **WiFi:** Tecnología inalámbrica madura, muy extendida, con una tasa de transferencia alta, aunque con alcance limitado y alto consumo de energía.
- **3G/4G/5G:** Tecnología de la red de telefonía móvil, con gran alcance y bajo consumo, pero limitada por la necesidad de que sean las operadoras de telefonía las que provean de cobertura en caso de no haberla.
- **Sigfox:** Tecnología de red similar a la anterior, bastante extendida, con gran alcance, que atraviesa fácilmente objetos sólidos y de muy bajo consumo de energía. Se denomina también "Low Power Wide Area Network" (LPWAN). Como la anterior, requiere de un operador. [15]
- **LoRa:** Es una red LPWAN similar a la anterior, pero que permite comunicaciones bidireccionales y no requiere de la intervención de un operador. Sin embargo, los módulos tienen mayor coste por tratarse de un sistema propietario. [16]

2.1.4. Protocolos IoT

Una vez vistas las tecnologías para el transporte de la señal, a continuación, se repasan los protocolos de IoT más destacados. Estos son: [17]

- **AMQP** (Advanced Message Queuing Protocol): protocolo de metodología Publish/Subscribe (publicación/subscripción). Utiliza transmisión binaria y se usa especialmente para aplicaciones corporativas, de alto rendimiento o críticas.
- **WAMP** (Web Application Messaging Protocol): protocolo abierto de tipo Publish/Subscribe que se ejecuta sobre WebSockets.
- **CoAP** (Constrained Application Protocol): protocolo para aplicaciones de baja capacidad.
- **sTOMP** (Streaming Text Oriented Messaging Protocol): protocolo simple que busca la máxima interoperabilidad. Se utiliza en HTTP y está basado en cadenas de texto.
- **XMPP** (eXtensible Messaging and Presence Protocol): protocolo abierto empleado en para apps de mensajería instantánea y basado en XML
- **WMQ** (WebSphere Message Queue): protocolo desarrollado por IBM orientado a mensajes. Estos mensajes pueden ser una colección de binarios o caracteres ASCII.

- **MQTT:** Como los anteriores, este protocolo también es del tipo M2M, con metodología Publish/Subscribe (publicación/subscripción), abierto y muy utilizado actualmente para sistemas IoT. Fue desarrollado por IBM pero liberado después.

2.2. Estudio de mercado

Comercialmente hay multitud de sistemas que hacen uso de señales EMG como se detalla en el apartado anterior, pero pocos son similares al desarrollado.

El más parecido que se ha encontrado realizando el análisis para este TFG es el sistema "NanoEMG" [18], de ENGTtotal. Se trata de una banda elástica que se coloca en la frente y permite al paciente encender o apagar la luz, avisar a la enfermera o cambiar de canal en la televisión.

Sin embargo, no dispone de conexión a internet, por lo que no se trata de un dispositivo IoT y su funcionamiento queda limitado a unos pocos metros de distancia. Además, se trata de un sistema cerrado, del que no se disponen de planos, esquemas o código.

Otro producto similar, aunque con una aplicación diferente, es el sistema sensor "Ultium-EMG", de Biomech Solutions [19]. Se trata de un moderno sistema de captura de señales EMG que permite visualizar un gran número de ellas en tiempo real, pero no utiliza dichas señales en una aplicación concreta, sino que se limita a mostrarlas.

Con el tratamiento que se realiza en el sistema propuesto en este TFG no se visualizan las señales, sino que se utilizan estas para detectar diferentes movimientos y enviarlos a un sistema remoto que podrá hacer uso de esas detecciones sin necesidad de analizar las señales.

Esto puede usarse para controlar un vehículo o un brazo robótico, generar diferentes señales de alerta por parte de un paciente hospitalizado, escribir usando los movimientos para desplazarse por un teclado virtual o controlar una casa domótica, entre otros muchos usos.

Todo ello desde cualquier parte del mundo gracias a tratarse de un dispositivo IoT y siendo además un sistema abierto, fácilmente replicable y a disposición del público a través de software libre.

3. Descripción funcional

El proyecto consiste en un sistema electromiográfico (EMG) que detecta las señales eléctricas que generan los músculos al moverse y en base a ellas se realizan acciones mediante un microcontrolador.

Para implementarlo se utilizarán diferentes tecnologías para la captura de señales mediante sensores EMG, su tratamiento y clasificación en diferentes movimientos utilizando un módulo MSP432P401R y envío inalámbrico a través de un módulo CC3100 de mensajes bajo el protocolo MQTT a un servidor remoto implementado con Node-Red.

3.1. Sistema electromiográfico (EMG) de telecontrol para pacientes con movilidad reducida

El funcionamiento del sistema será el siguiente:

Los sensores, conectados mediante electrodos superficiales a los músculos del usuario, generan una señal entre 0V y 3.3V, estando dicha señal centrada en 1.5V. La tensión de alimentación puede ser de 3.3V o de 5V, ambas compatibles con la plataforma de desarrollo MSP432.

Esta señal se mide a través de 2 canales del ADC de la plataforma de desarrollo. Una vez procesada y filtrada la señal, se analiza para detectar el tipo de movimiento realizado y se generan las diferentes ordenes de movimiento.

Dichas órdenes se envían mediante el protocolo MQTT y a través de WiFi, utilizando el módulo CC3100 de Texas Instruments, a un bróker (servidor intermedio) que hace de intermediario entre el módulo de control y el interfaz gráfico, donde se muestran las ordenes detectadas.

3.1.1. Requerimientos

Los requerimientos son:

Principales

- Detección de movimientos faciales vía EMG
- Caracterización y diferenciación de hasta 5 movimientos. (Adelante, atrás, izquierda, derecha y alarma)
- Comunicación inalámbrica vía wifi
- Servidor de control y recepción de datos
- Sistema de envíos de alarma en caso de accidente
- Interfaz gráfica basada en cliente web
- Bajo consumo

Extraordinarios

- Acceso desde teléfono móvil
- Pequeño y portable

- Sistema motor remotamente controlado

3.1.2. Diagrama de bloques del sistema

En la Figura 8 se muestra el diagrama de bloques del sistema completo:

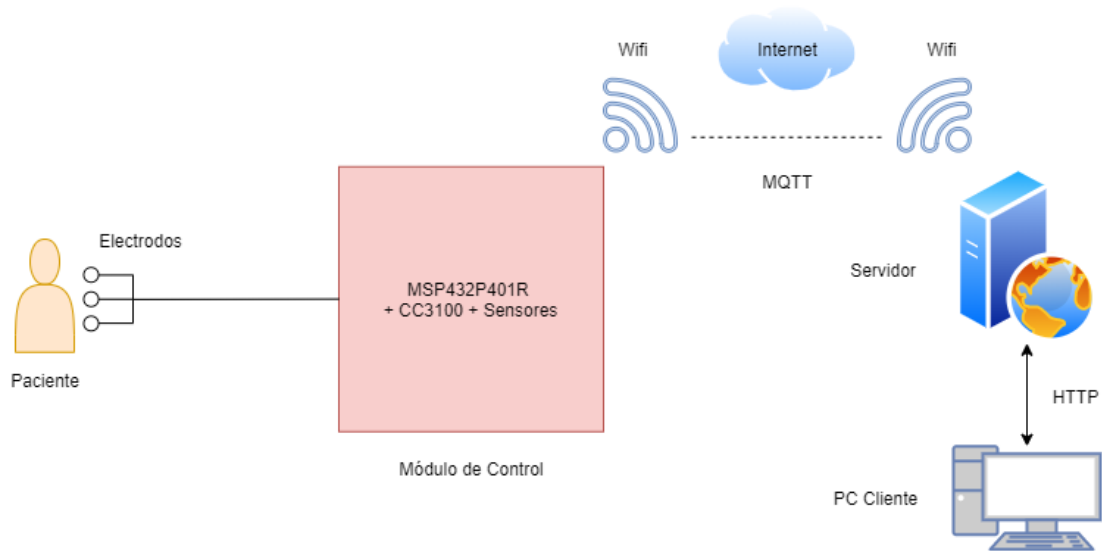


Figura 8 - Diagrama de bloques del sistema

3.1.3. Diagrama de bloques de la aplicación

En la Figura 9, se muestra el diagrama de bloques de la aplicación software.

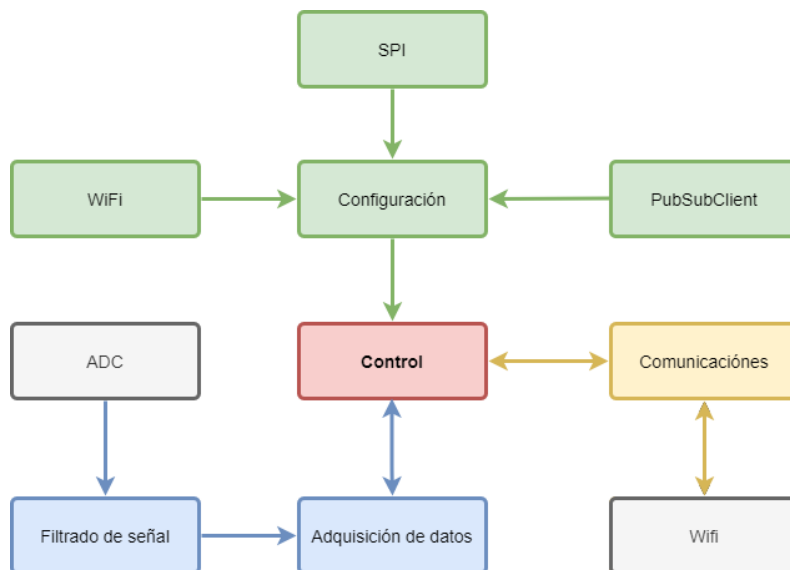


Figura 9 - Diagrama de bloques de la aplicación

Leyenda:

Verde – Módulos de configuración

Rojo – Módulo de control

Azul – Módulos de proceso de señal

Naranja – Módulo de comunicaciones

Gris – Módulos de conexión con el exterior

Descripción de cada bloque:

Librerías: Se necesitarán las librerías SPI y WiFi para el control del módulo CC3100, así como PubSubClient para la comunicación por MQTT.

Configuración: Importación de librerías y configuración de todas las constantes y variables iniciales del sistema.

ADC: Lectura de los ADCs

Filtrado de señal: Aplicación de filtro/s digital/es para eliminar ruido de la señal leída por los ADCs (los sensores ya tienen un filtro paso bajo y otro paso alto por hardware que no será necesario duplicar)

Adquisición: En base a la señal filtrada, detectar distintos tipos de movimiento (activación del lado izquierdo, del derecho, de ambos a la vez ...) y generación de órdenes (Izquierda, Derecha, Delante, Atrás, Parar)

Comunicaciones: Construye los mensajes MQTT a enviar al bróker en base a los movimientos detectados por el módulo de adquisición que le son enviados desde el módulo de control

Wifi: Envío de los mensajes al bróker MQTT.

3.1.4. Sistema de comunicación

La comunicación se realiza a través de la red de modo inalámbrico mediante un módulo CC3100, que proporciona conectividad WiFi al MSP432.

En la Figura 10 se muestra un esquema del sistema de comunicación por red:

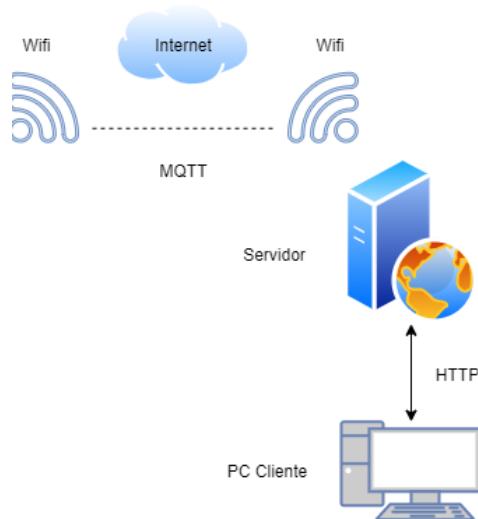


Figura 10 - Esquema de red

A través de dicho módulo se envían mensajes bajo el protocolo MQTT, muy usado en proyectos de IoT.

También se ha valorado el uso de la interfaz REST sobre HTTP, también muy usado actualmente.

REST utiliza los métodos de HTTP (GET, PUT, POST, DELETE) para enviar o recibir datos, y su ventaja sobre MQTT es su capacidad para enviar o recibir grandes cantidades de información de manera eficiente, pero su consumo energético y de recursos es mucho mayor ya que requiere de una conexión y desconexión con cada comunicación, lo que lo hace poco eficiente para transmitir datos pequeños. [20]

MQTT, sin embargo, utiliza el mecanismo “Publish/Subscribe” (publicación/suscripción) lo que le proporciona asincronía, una alta escalabilidad y una necesidad menor de recursos, por lo que su consumo energético es también mucho menor, algo que lo hace ideal para dispositivos IoT que requieren estar siempre conectados con el menor consumo posible y que envían datos pequeños.

Es por esto por lo que se ha elegido el protocolo MQTT.

Al tratarse de un protocolo de publicación/suscripción, los mensajes deben ser enviados a un “bróker” (intermediario) remoto. Este bróker se puede implementar mediante una gran cantidad de programas de código abierto, como, por ejemplo: [21]

- **Mosquitto.** Es posiblemente el broker MQTT más conocido. Se trata de un bróker multiplataforma muy liviano, que requiere pocos recursos y muy adecuado para servidores de baja potencia. Está desarrollado por la fundación Eclipse y distribuido bajo licencia EPL/EDL.
- **Mosca.** Es un broker MQTT para Node.js, desarrollado en Javascript por Matteo Collina. Puede ser empleado como aplicación independiente o embebido en cualquier proyecto de Node.js

- **Aedes.** Es también un broker MQTT Open Source para Node.js diseñado para sustituir a Mosca.
- **HBMQTT.** Este broker está escrito en Python y funciona sobre asincio.
- **EMQTT.** Este broker está desarrollado en Erlang/OTP, y fue diseñado para aplicaciones con grandes exigencias en escalabilidad.
- **RabbitMQ.** Es un popular broker de mensajería AMQP, que también permite emplear el protocolo MQTT a través de un Adaptador.
- **HiveMQ CE.** La versión “Community Edition” del popular HiveMQ es un broker basado en Java.
- **ActiveMQ.** Es un broker de mensajería JMS (Java Message Script) desarrollado por Apache, que también admite protocolo MQTT.
- **Moquette.** Un broker MQTT escrito en Java desarrollado por Eclipse, que destaca por su ligereza.
- **MQTTnet.** Se trata de un broker MQTT para .NET.

Se ha elegido Mosquitto por ser el más conocido y tener un gran soporte de la comunidad.

El bróker se instala en un servidor virtual dentro de la plataforma VirtualBox, con sistema operativo Linux Ubuntu.

A su vez, en el servidor se utiliza Node-Red para recopilar dichos mensajes MQTT y mostrarlos mediante una página web implementada con Node-Red Dashboard, que es accesible desde cualquier cliente web mediante el protocolo HTTP. En la Figura 11 se muestra el diagrama de bloques de dicha comunicación:

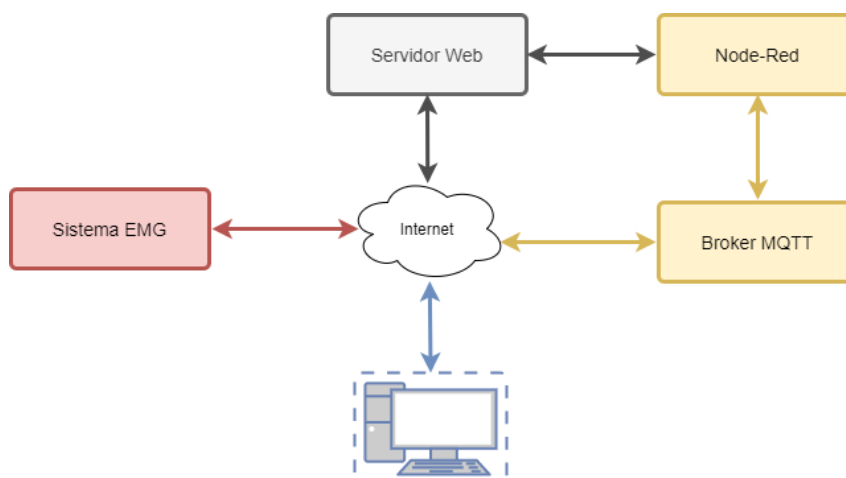


Figura 11 - Diagrama de bloques de la comunicación

3.2. Aplicación web

3.2.1. Diagrama de bloques

La aplicación web se basa en 2 flujos de trabajo que recogen los datos, los identifican y tratan y los muestran en un interfaz gráfico, tal como se ve en la Figura 12:

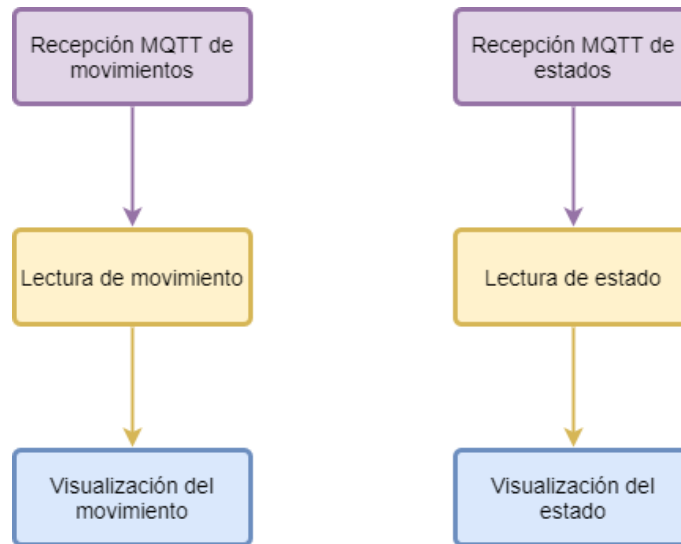


Figura 12 - Esquema software de la aplicación web

El primer flujo, el de la izquierda, es el que recibe los mensajes MQTT del topic con el que se envían los movimientos. La entrada de datos se realiza mediante suscripción por parte de la aplicación web al topic MQTT indicado

El segundo flujo, el de la derecha, es el que recibe los mensajes MQTT del topic con el que se envían la información de estado. Como en el caso anterior, la entrada de datos se realiza mediante suscripción por parte de la aplicación web al topic indicado

3.3. Dispositivo Hardware

Como se mencionó en el apartado 2.1.2, actualmente existe una gran cantidad de microcontroladores que podrían cumplir el cometido de este proyecto.

El bajo consumo y las altas prestaciones de la familia MSP430 de Texas Instrument y sus microcontroladores ARM Cortex M4, junto a la posibilidad de poder programarla tanto en lenguaje Processing como a bajo nivel si la aplicación lo requiere, convierten a esta familia en un gran candidato.

Además, los ADC pueden trabajar con 12 e incluso con 14 bits, mientras que en Arduino trabajan con 10 bits y en Raspberry Pi ni siquiera se disponen de entradas analógicas.

Es por esto por lo que se ha elegido el MSP432P401R.

Para la elección del módulo de conexión que proporciona conectividad WiFi al MSP432 se han valorado 2 opciones: el módulo CC3100.

- **Módulo CC3100:** Este módulo de Texas instrument incluye radio 802.11 b/g/n y un potente motor de cifrado para conexiones de Internet rápidas y seguras de 256 bits. Como se aprecia en la Figura 13 es bastante grande:

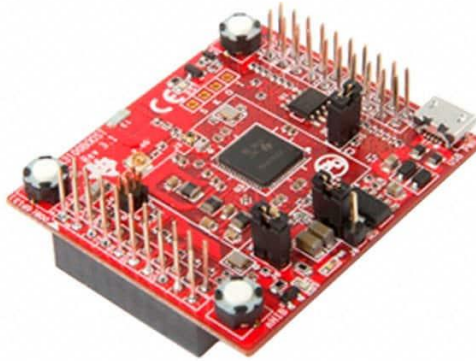


Figura 13 - Módulo WiFi CC3100

- **Módulo ESP-01:** Se trata de un módulo sencillo y muy barato. Puede trabajar en modo o como punto de acceso y se alimenta a 3.3V. Como se puede ver en la Figura 14, es mucho más pequeño que el CC3100.

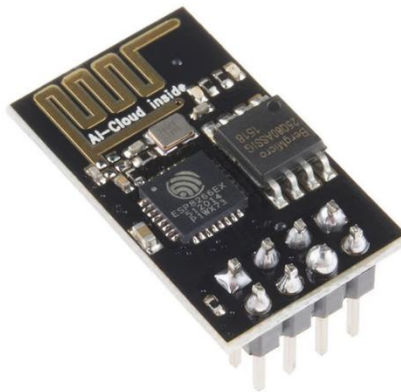


Figura 14 - Módulo WiFi ESP-01

Se ha elegido el CC3100 para el prototipo por formar parte de la misma plataforma y ser totalmente compatible con el MSP432. El ESP-01 se reserva para la industrialización por su reducido coste y tamaño.

Para la elección de los sensores EMG se han tenido en cuenta varias opciones:

- **Pololu Muscle Sensor:** Como se puede ver en la Figura 15, se trata de un sensor compacto, pequeño y ligero, que necesita una alimentación simétrica para el amplificador, lo que implicaría complicar la alimentación del dispositivo. [22]



Figura 15 - Pololu Muscle Sensor

- **Myoware Muscle Sensor:** Se trata de un sensor algo más grande, y tal como se aprecia en la Figura 16, 2 de los electrodos se encuentran integrados en la propia PCB, que se coloca directamente sobre el músculo a medir. Esto es una ventaja para músculos grandes, pero no es cómodo para músculos pequeños como son los faciales.

La alimentación se realiza con una fuente simple de 3.1V a 5.9V, lo que es más conveniente para el uso con microcontroladores (habitualmente alimentados entre 3.3V y 5V). Su coste es mayor al de otros modelos. [23]



Figura 16 - Myoware Muscle Sensor

- **Groove EMG Detector:** Como se aprecia en la Figura 17, es un sensor pequeño, ligero y que puede ser alimentado con una fuente simple de 3.3V a 5V [24]

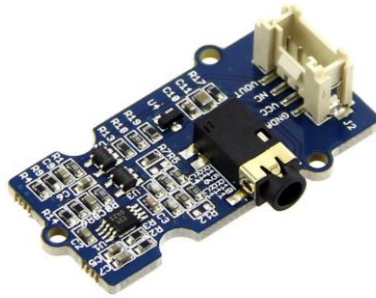


Figura 17 - Grove EMG Detector

- **Mikroe EMG click:** Similar al Grove EMG Sensor que incorpora un potenciómetro para regular la ganancia de amplificación, con alimentación simple, pero sólo a 5V, y un tamaño muy superior, como se aprecia en la Figura 18. El coste es también considerablemente mayor [25]



Figura 18 - Mikroe EMG click

Por su tamaño y precio contenidos, y la posibilidad de alimentarlo con la misma fuente que el MSP432 se ha elegido el Grove EMG Sensor.

3.3.1. Diseño Hardware

Para la realización del diseño del hardware se ha utilizado el programa EasyEDA.

Se trata de un software para diseño de esquemáticos y PCBs gratuito, sencillo y que no requiere instalación, ya que está basado en la nube, aunque dispone de un cliente que se puede instalar en el PC para mayor comodidad.

Pese a ser gratuito es bastante completo. Dispone de multitud de librerías de componentes con los que crear el circuito, y en caso de no disponer del componente en concreto dispone de un editor para crear componentes personalizados.

También permite la importación de esquemáticos creados con las más conocidas herramientas, como Altium Designer, Eagle o KiCAD. Posee también una herramienta de simulación del circuito que permite la visualización tanto de señales digitales como analógicas.

El esquema eléctrico del sistema diseñado es el mostrado en la Figura 19:

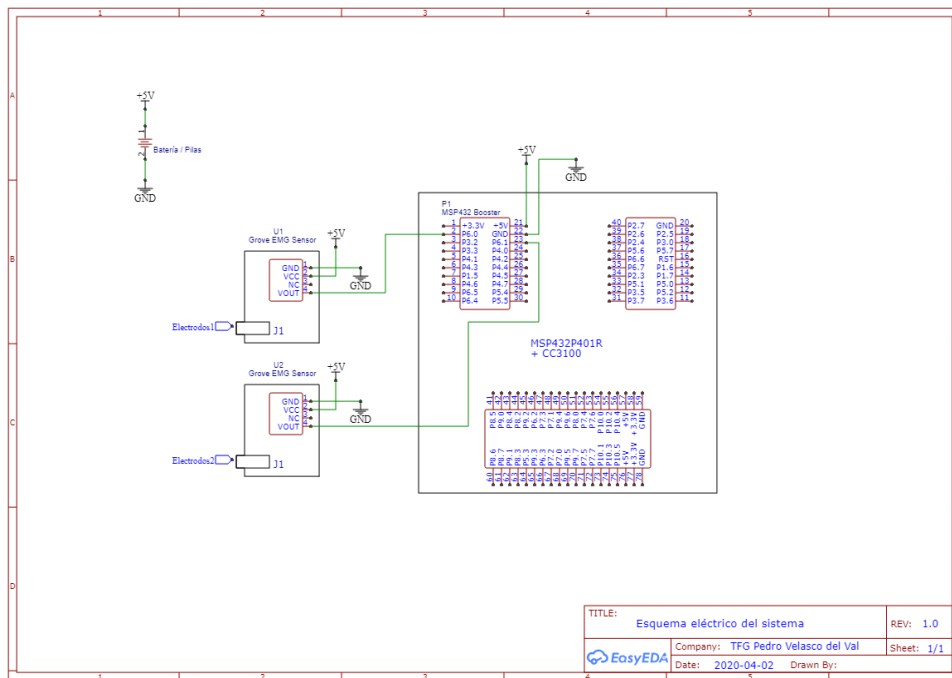


Figura 19 - Esquema eléctrico del Hardware

Por su parte, el esquema eléctrico de los sensores lo proporciona el fabricante en su página web [24] y que se analizará en apartados posteriores, es el mostrado en la figura 20:

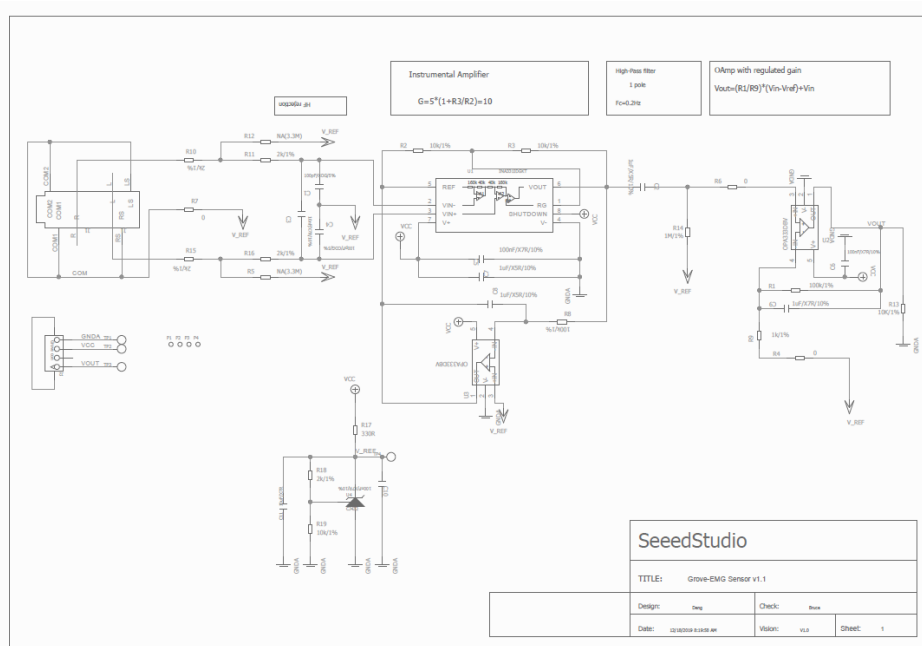


Figura 20 - Esquema eléctrico del sensor Grove EMG Detector

4. Descripción detallada

Como se adelantaba en el apartado 3, los sensores se conectan mediante electrodos a los músculos del usuario y generan una señal entre 0V y 3.3V centrada en 1.5V y filtrada por medio del hardware de los sensores.

Esta señal se mide a través de 2 canales del ADC de la plataforma de desarrollo MSP432. Una vez procesada y filtrada también por software, la señal resultante se analiza para detectar el tipo de movimiento realizado y se generan las diferentes ordenes de movimiento.

Dichas órdenes se envían mediante el protocolo MQTT y a través de WiFi, utilizando el módulo CC3100 de Texas Instruments, al bróker Mosquitto que hace de intermediario entre el módulo de control y el interfaz gráfico de Node-Red, donde se muestran las ordenes detectadas.

4.1. Diseño hardware

4.1.1. Diseño general

El diseño general, tal como vimos en el esquema del capítulo anterior, es el siguiente:

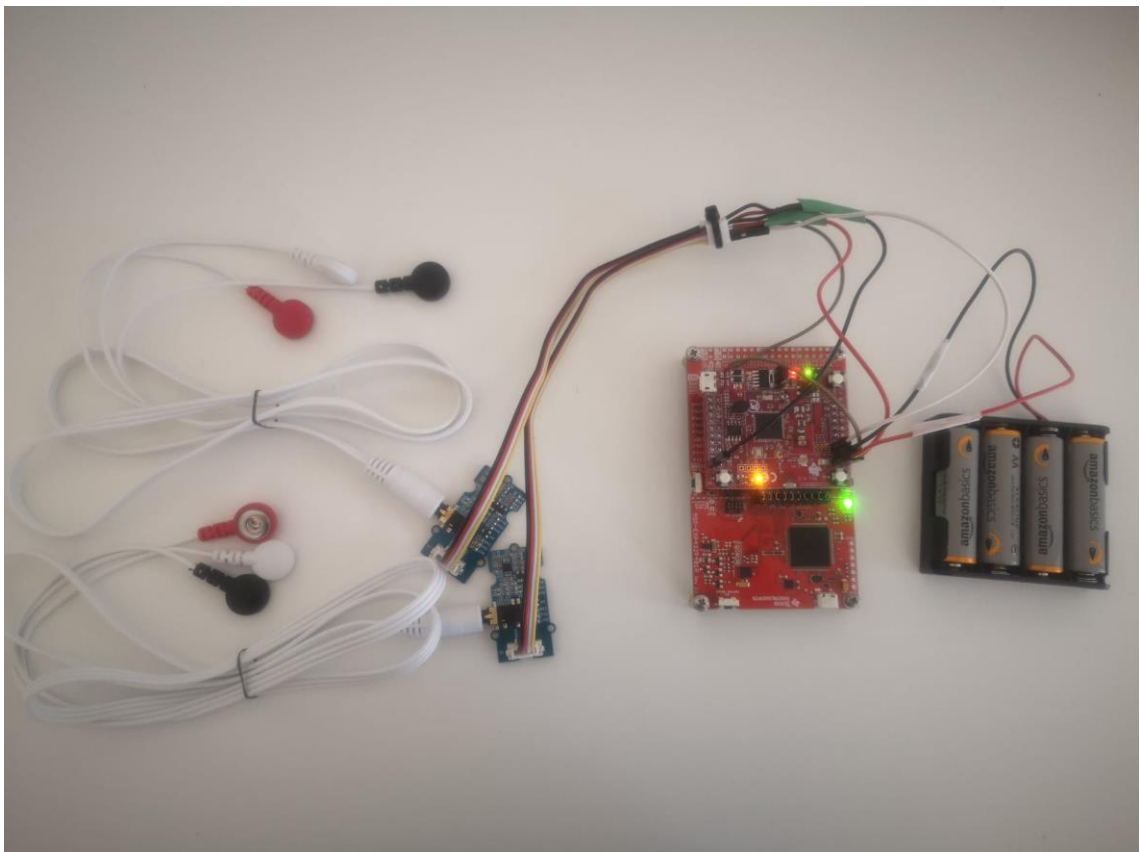


Figura 21 - Montaje real del hardware

A continuación, se detallan todos los componentes que se pueden ver en la Figura 21.

4.1.2. MSP432

A principios del año 2015, Texas Instruments lanzó al mercado el LaunchPad MSP432P401R.

Se trata de un microcontrolador de ultra-bajo consumo, con tecnología ARM Cortex M4 de 32bits y una velocidad de hasta 48Mhz, 256KB de memoria Flash, 64KB de memoria RAM, 32KB de memoria ROM, unidad de coma flotante y un ADC con 24 canales de 14bits de resolución y con una velocidad de muestreo de hasta 1Msps, con un consumo en de 95uA/Mhz en el modo activo y que llega hasta 850nA en el modo RTC.

Posee además 4 conexiones I2C, 4 SPI, 4 UART que le permiten interactuar con multitud de dispositivos.

También posee un conector “BosterPack” de 40 pines, que le permite agregar módulos como el CC3100 de forma muy rápida y sencilla, colocando uno sobre el otro.

4.1.3. Módulo CC3100

En el año 2013 Texas Instrument lanzó al mercado el módulo CC3100, el primero de su clase con el certificado “Wi-Fi Certified”, que es totalmente compatible con el MSP432 y se comunica con este a través del bus SPI.

Este módulo puede funcionar en los modos estación, punto de acceso y Wi-Fi Direct, aunque en este proyecto sólo su usará en modo estación, e incluye seguridad WPA2. Se puede alimentar de 2.1V a 3.6V, con una potencia de transmisión de hasta 18.0 dBm y una sensibilidad de recepción de hasta -95.7dBm. Todo ello, con una velocidad máxima de hasta 16 Mbps.

En la Figura 22 se puede apreciar cómo se integran ambos elementos:

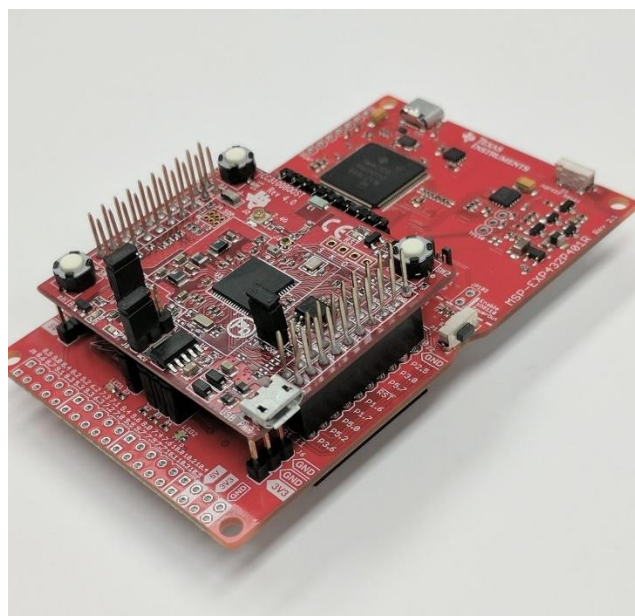


Figura 22 - MSP432P401R junto al módulo CC3100

4.1.4. Circuito sensor EMG

El Grove EMG Detector es un sensor pequeño, ligero que conecta el cuerpo humano con un sistema electrónico.

La tensión de alimentación del sensor puede ser de 3.3V o de 5V, y en este proyecto se conectarán al pin de 3.3V del MSP432.

Cada sensor (izquierdo y derecho) utiliza 3 electrodos para 1 canal de adquisición. El sensor, gracias a 3 electrodos (negro, rojo y blanco), recolecta las señales musculares, las procesa mediante serie de etapas de tratamiento, generando una señal única entre 0V y 3.3V, centrada en 1.5V (para poder detectar tanto la contracción como la extensión), que puede ser leída posteriormente por un ADC.

Como se puede ver en el esquema general de la Figura 20, el sensor izquierdo se conecta al pin 2 del MSP432, que lleva al canal A15 del ADC, y el sensor derecho se conecta al pin 23 del MSP432, que lleva al canal A14 del ADC.

La calidad de la señal que se obtiene de los electrodos es bastante baja, ya que tiene poca amplitud y es muy ruidosa, por lo que requiere una adecuación antes de poder ser tratada. La correcta colocación de los electrodos es muy importante, ya que de ella depende en gran parte la calidad de la señal recibida inicialmente.

Ni en la web del fabricante ni en las hojas de características del sensor se indica cómo se debe realizar la colocación de los electrodos, por lo que se ha recurrido a un polímetro y el análisis del esquemático del sensor para comprobar que la correspondencia entre electrodos y señales es la siguiente:

- Electrodo negro: Corresponde a la línea de entrada COM. Es el electrodo común que actúa de neutro.
- Electrodo rojo: Corresponde a la línea de entrada Vin+. Es el electrodo positivo.
- Electrodo blanco: Corresponde a la línea de entrada Vin-. Es el electrodo negativo.

Por lo tanto, para la correcta detección del movimiento, el electrodo que actúa como neutro (el negro) debe posicionarse en alguna zona que no se mueva. En el caso de un tetrapléjico puede ser el cuello, si no lo puede mover, si el usuario puede mover el cuello otra buena ubicación es el nacimiento del pelo en el centro de la frente.

Los otros 2 electrodos (rojo y blanco) funcionan como entrada diferencial y deben colocarse en el músculo que se desea digitalizar. Como lo que se desea detectar son pestaños de los ojos, el electrodo rojo se colocará encima de la ceja y el blanco sobre la mejilla.

Inicialmente, como se puede comprobar en el estudio de viabilidad, se colocaron los electrodos en las mejillas, pero durante las pruebas del sistema completo se detectó que la señal generada por los músculos de la frente tiene una amplitud mayor que en la mejilla, por lo que se cambiaron los electrodos de posición, utilizando finalmente la indicada en la Figura 23:

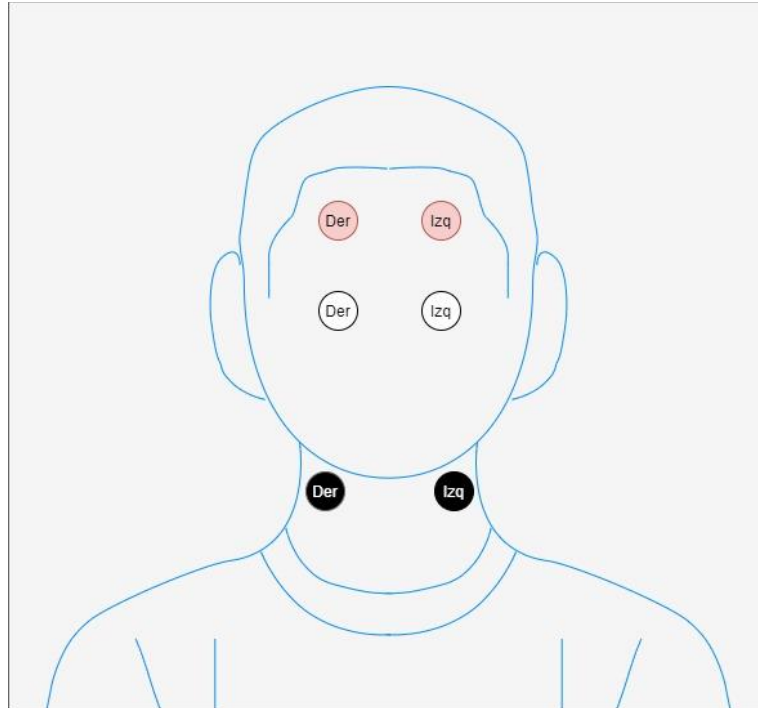


Figura 23 - Posición final de los electrodos

4.1.5. Tratamiento de la señal por hardware

A continuación, se detallan las diferentes etapas de tratamiento de la señal que se realizan en los sensores EMG.

4.1.5.1. Rechazo de alta frecuencia

La primera etapa, es la de adaptación de señal y rechazo de señales de alta frecuencia, o “HF rejection”, tal como se ve en la Figura 24:

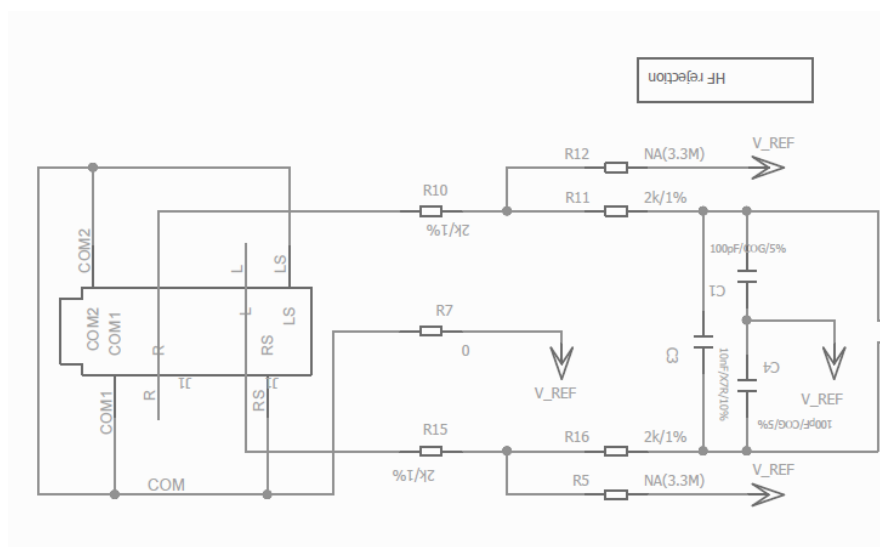


Figura 24 - Etapa de rechazo de alta frecuencia

El objetivo de esta etapa es, por un lado, utilizar el canal común (electrodo negro) como señal de referencia y tener a la salida 2 señales diferenciales referenciadas a dicho canal.

Este paso es muy importante ya que las señales que generan los músculos son muy pequeñas, del orden de 0-10 milivoltios, lo que hace que la relación señal /ruido a la entrada sea muy baja.

Como el cuerpo genera muchas otras señales que pueden introducir ruidos indeseados, esta relación es aún más baja. Con la utilización del electrodo de referencia se consigue reducir en gran medida el ruido introducido por el propio cuerpo y maximizar la relación señal ruido a la entrada.

Por otro lado, esta etapa elimina la alta frecuencia por medio del filtrado con los condensadores C1, C3 y C4.

4.1.5.2. Amplificador de instrumentación

La segunda etapa consiste en un amplificador de instrumentación, tal como se muestra en la Figura 25.

Un amplificador de instrumentación es un dispositivo creado a partir de amplificadores operacionales. Está diseñado para tener una alta impedancia de entrada y un alto rechazo al modo común (CMRR) es una medida del rechazo que ofrece la configuración a la entrada de voltaje común.

El amplificador diferencial realiza la resta de sus dos entradas, suprimiendo cualquier voltaje común a ambas entradas, y luego se multiplica el resultado por un factor de amplificación (ganancia).

Su utilización es común en aparatos que trabajan con señales muy débiles, tales como equipos médicos (por ejemplo, el electrocardiógrafo), para minimizar el error de medida. [26]

La ganancia del amplificador viene dada, como se ve en la Figura 26, por la siguiente fórmula:

$$G = 5 * \left(1 + \frac{R3}{R2}\right) = 5 * \left(1 + \frac{10 * 10^3}{10 * 10^3}\right) = 10$$

Esta etapa elimina la señal común a ambas entradas, las resta y las multiplica por 10, por lo que aún se tiene una señal del orden de milivoltios y sólo se han eliminado el ruido común y el de alta frecuencia. Por tanto, se debe seguir tratando la señal para hacerla viable.

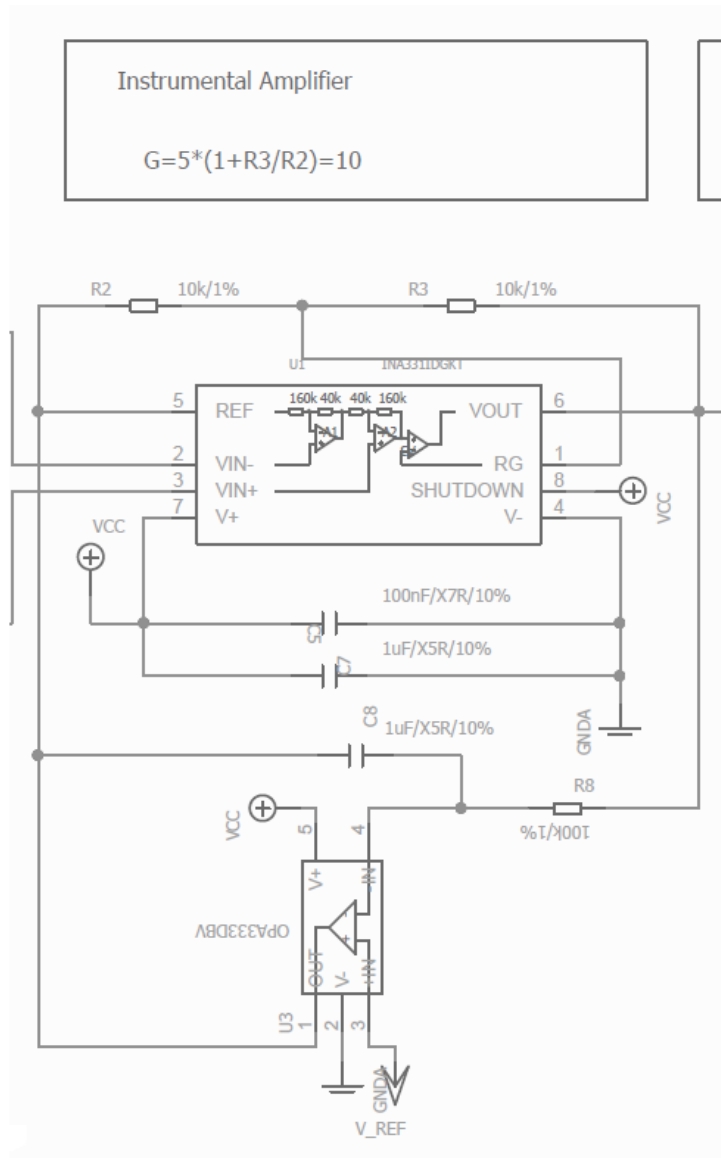


Figura 25 - Etapa de amplificador instrumental

4.1.5.3. Filtro paso alto

En la Figura 26 se muestra la tercera etapa, que consiste en un filtro paso alto.

Un filtro es un circuito electrónico, con una entrada y una salida, y que a su salida atenúa unas frecuencias de la señal de entrada y deja pasar (o incluso amplifica, en el caso de los filtros activos) otras.

El filtro paso alto es un tipo de filtro cuya respuesta en frecuencia atenúa las componentes de baja frecuencia, pero deja pasar las de alta frecuencia. Así, se eliminan ruidos de corriente continua.

Concretamente se trata de un filtro de primer orden (o de un polo), ya que sólo se compone de un condensador en serie con la señal y una resistencia en paralelo.

A la frecuencia a la que ganancia de salida se reduce en 3 decibelios (dB) se le llama frecuencia de corte del filtro, y en este caso se calcula con la fórmula:

$$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi * 1 * 10^6 * 1 * 10^{-6}} = \frac{1}{2\pi} = 0,16 \text{ Hz}$$

Por lo tanto, esta etapa elimina las señales de 0 a 0,16 Hz y deja pasar aquellas cuya frecuencia sea superior a 0,16 Hz, aunque en el esquemático el fabricante lo ha redondeado a 0,2 Hz, como se puede apreciar más adelante en la Figura 27.

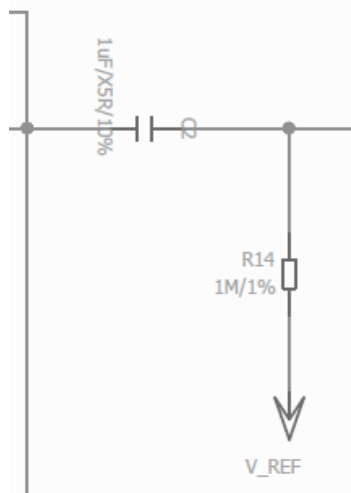
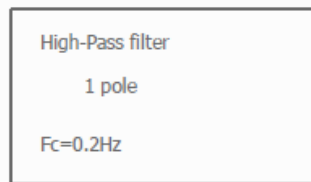


Figura 26 - Etapa de filtro paso alto

4.1.5.4. Amplificador de tensión

La cuarta y última etapa como se aprecia en la Figura 27 consiste en un amplificador de tensión implementado, al igual que el amplificador de instrumentación de la segunda etapa, por medio de un amplificador operacional.

Como se vio en la segunda etapa, el amplificador instrumental no amplificaba la señal lo suficiente, llevándola sólo al orden de decenas de milivoltios. En esta etapa la ganancia del amplificador viene dada, como se ve en la Figura 28, por la siguiente fórmula:

$$G = \frac{R1}{R9} = \frac{100 * 10^3}{1 * 10^3} = 100$$

Por lo tanto, en esta etapa se amplifica la señal 100 veces, por lo que pasamos del orden de decenas de milivoltios al de voltios, lo que ya genera una señal medible con un ADC.

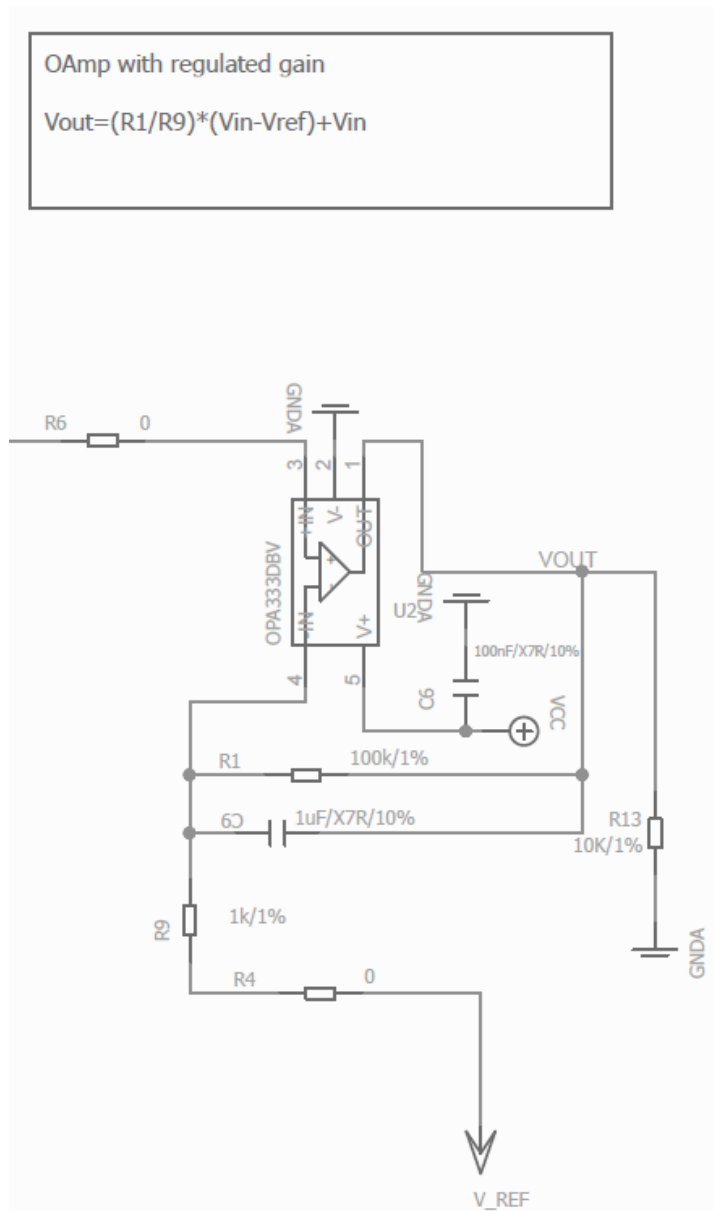


Figura 27 - Etapa de amplificación de tensión

Se podría haber implementado también, tal como se hace en equipos de electrocardiografía, un filtro notch (o de banda eliminada) para eliminar el ruido generado por la línea eléctrica (que funciona a 50 Hz en España), ya que es una fuente de ruido muy presente en cualquier ubicación, pero debido a que la señal EMG tiene una componente muy grande alrededor de los 50Hz el uso de estos filtros no está indicado para esta aplicación.

4.1.6. Alimentación del circuito

El circuito se alimenta con una batería implementada con 4 pilas en serie de tipo AA, recargables de 1,2V (4,8 voltios en total) o alcalinas de 1,5V (6 voltios en total) conectada al pin de 5V del LaunchPad MSP432P401R.

Éste convierte los 4,8/6V en 3,3V, que son los que alimentan a todos los circuitos que componen el sistema, el MSP432, los sensores y el módulo WiFi CC3100.

La tarea por la que se alimenta el sistema mediante baterías, si bien se ha realizado, no cumple el objetivo de durar 1 día completo, ya que el consumo es mayor de lo esperado (0,35A aproximadamente) por lo que la duración no será de 1 día, sino de unas 8 horas con pilas alcalinas de 2700mAh.

Tras medir consumos con el polímetro, se encuentra que el CC3100 consume 0,10A y los sensores 0,02A. El consumo se debe mayormente a la placa de desarrollo (0,24A), muy elevado respecto al consumo teórico de 80uA/Mhz (algo menos de 4mA para 48Mhz), lo que indica un defecto de fabricación de la PCB del LaunchPad.

4.2. Diseño software

4.2.1. Diseño software del dispositivo

Como se vio en la Figura 9, el software del dispositivo se compone de los siguientes módulos:

Configuración: Importación de librerías y configuración de todas las constantes y variables iniciales del sistema.

Filtrado de señal: Aplicación de filtro/s digital/es para eliminar ruido de la señal leída por los ADCs

Adquisición de datos: En base a la señal filtrada, detecta distintos tipos de movimiento (activación del lado izquierdo, del derecho, de ambos a la vez ...) y generación de órdenes (Izquierda, Derecha, Adelante, Atrás, Parar y Alarma). Engloba las funciones de adquisición, cálculo de umbrales y detección de movimientos.

Comunicaciones: Construye y envía a través de WiFi los mensajes MQTT a enviar al bróker en base a los movimientos detectados por el módulo de adquisición que le son enviados desde el módulo de control

Estos módulos se han implementado mediante el IDE Energía, ya que con él se puede programar el MSP432 con el lenguaje Processing de Arduino.



Figura 28 - Caso de uso del software del dispositivo

Como se puede apreciar en el caso de uso de la Figura 28, el software del dispositivo se encuentra siempre disponible tras el cálculo de umbrales realizando la detección de movimientos, una vez recibe un movimiento lo envía a través de un mensaje MQTT.

4.2.2. Filtrado por software de la señal

Además del filtrado que realiza el hardware de los sensores, se ha implementado un filtro de media móvil exponencial (EMA) digital para reducir aún más el ruido de alta frecuencia detectado por los sensores.

Se trata de un filtro paso bajo de primer orden que atenúa el ruido de alta frecuencia, fácilmente implementable en un microcontrolador aplicando la siguiente fórmula de cálculo:

$$A_n = \alpha * M + (1 - \alpha) * A_{n-1}$$

Donde:

- A_n es la muestra actual filtrada.
- α es el factor de suavizado (entre 0 y 1).
- M es la muestra actual en crudo.
- A_{n-1} es la muestra anterior filtrada.

El factor de suavizado (α) tiene un gran impacto sobre el comportamiento del filtro, variando enormemente la magnitud del filtrado y la velocidad de este ante los cambios. Cuanto mayor sea dicho valor menor es el efecto de filtrado, pero este es más rápido ante los cambios, y viceversa.

Tras las pruebas realizadas, se decide utilizar un factor de suavizado de 0.2 para tener una buena relación de compromiso entre la velocidad de reacción y el filtrado de la señal. Tras implementar el filtro en el MSP432, en la Figura 29 se puede comprobar cómo se reduce notablemente el ruido base de la señal:

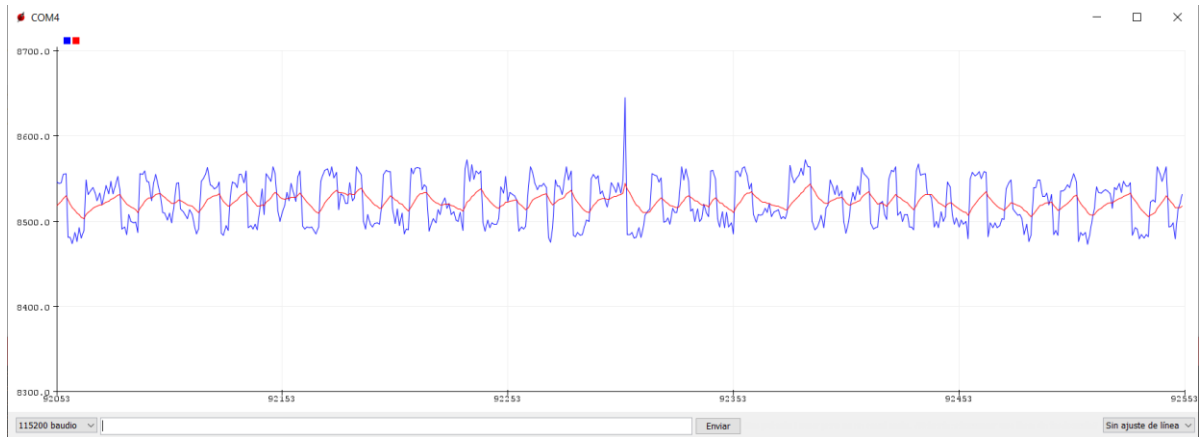


Figura 29 - Filtrado de señal

Leyenda:

Azul - Señal sin filtrar

Rojo - Señal filtrada

En la Figura 30 se muestran tres movimientos completos, marcados con un círculo verde.

En ella se puede ver cómo esta reducción de ruido se aprecia también cuando se realizan movimientos, con una línea mucho más suave y nítida, pero en la que la forma de onda del movimiento se mantiene estable.

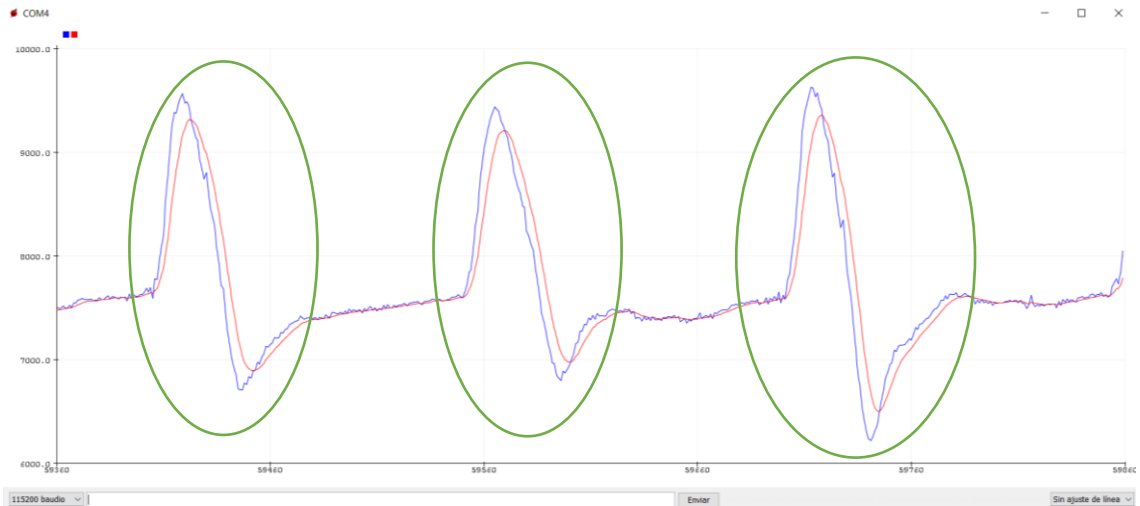


Figura 30 - Movimiento filtrado

4.2.1. Cálculo de umbrales

Para minimizar los efectos del ruido ambiental, al inicio y cada vez que se pulse el botón PUSH2 del launchpad, se realiza un cálculo dinámico de los umbrales de detección.

Esta tarea lee los ADCs de los sensores durante un tiempo mientras el usuario está en reposo y almacena el valor máximo y el valor mínimo leídos, aplica un margen de seguridad a esos valores y los almacena en una variable para su posterior uso en el algoritmo de detección.

Si se detecta un movimiento mientras se está realizando el cálculo, este se repite hasta que no se detecte movimiento.

Cuando finaliza el cálculo se enciende el led verde del launchpad para indicar que el cálculo ha sido correcto, y también se envía un mensaje MQTT para indicar lo mismo al servidor.

4.2.2. Algoritmo de detección de movimientos

El algoritmo de detección de movimientos toma muestras de los ADC continuamente.

Esas muestras se comparan con los umbrales de detección calculados previamente para decidir si se ha producido algún movimiento.

Si se detecta movimiento en alguno de los dos lados se siguen leyendo los ADC, manteniéndose en un bucle hasta detectar un movimiento del otro lado, con lo que se detectaría un movimiento de ambos lados a la vez, o hasta que el nivel de señal de ese lado baje del umbral mínimo, con lo que se tendrá un movimiento simple si no se produce otro igual en menos de un tiempo estipulado en el archivo de configuración, o un movimiento doble en caso de que sí se produjera un nuevo movimiento de ese lado en menos tiempo del configurado.

Con este algoritmo se consiguen detectar 6 movimientos diferentes:

1. Movimiento simple del lado izquierdo (Izquierda)
2. Movimiento doble del lado izquierdo (Adelante)
3. Movimiento simple del lado derecho (Derecha)
4. Movimiento doble del lado derecho (Atrás)
5. Movimiento de ambos lados a la vez (Parar)
6. Movimiento doble de ambos lados a la vez (Alarma)

4.2.3. Comunicaciones

Como se adelantó en el capítulo 3, el protocolo de comunicaciones elegido es MQTT, que utiliza el mecanismo "Publish/Subscribe".

Este mecanismo requiere el uso de un intermediario que recolecta y envía los mensajes entre los clientes, pero los clientes no se comunican directamente entre ellos. Este protocolo consiste en que los clientes suscriptores se suscriben a uno o varios topics (temas) que les interesan enviando un mensaje de suscripción al bróker.

Los clientes publicadores, envían mensajes con uno o varios temas al bróker, para que aquellos suscriptores que estén interesados y se hayan suscrito los reciban, como se puede ver en la Figura 31. [17]

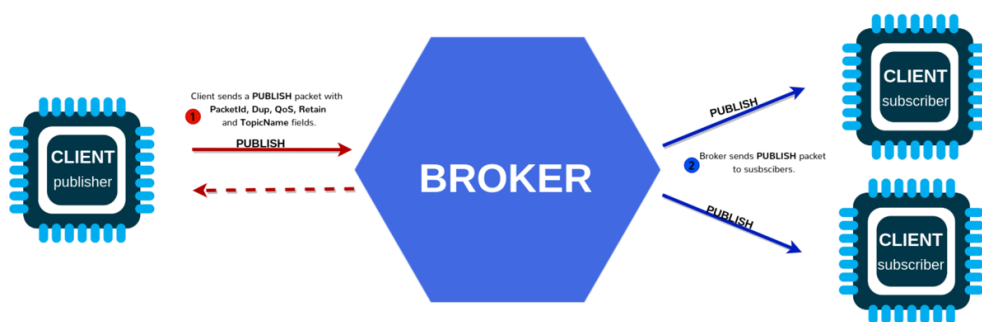


Figura 31 - Esquema de comunicación Publish/Subscribe.

Para enviar dichos mensajes al bróker debe usarse la dirección IP del servidor y enviarlos a través del puerto 1883.

Una de las razones por las que el protocolo MQTT es tan ágil y ligero es tipo y composición de sus mensajes.

En la Figura 32 se puede apreciar la sencillez de dicha composición. [21]

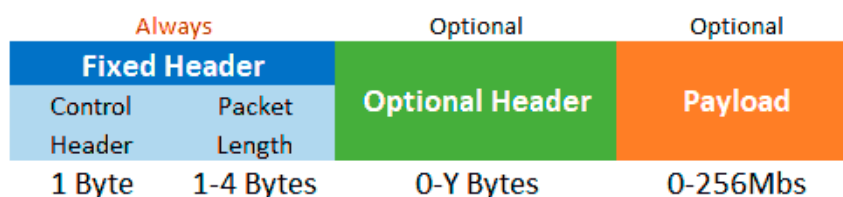


Figura 32 - Estructura de los mensajes MQTT

Cada mensaje se divide en 3 partes:

- **Cabecera fija.** Esta parte es obligatoria y se compone de una cabecera de control que ocupa 1 byte con el que se identifica el código de control que indica el tipo de mensaje, y de la longitud del mensaje, que ocupa de 1 a 4 bytes.
- **Cabecera variable.** Es una parte opcional que contiene información adicional necesaria sólo en algunos mensajes.
- **Contenido (Payload).** Es también opcional y contiene el mensaje propiamente dicho. Puede ocupar entre 0 y 256 Mb.

En la Tabla 2 se muestran los diferentes tipos de mensaje y su respectivo código de control. Marcados en verde se encuentran los mensajes que se utilizan en este proyecto:

Message	Code
CONNECT	0x10
CONNACK	0x20
PUBLISH	0x30
PUBACK	0x40
PUBREC	0x50
PUBREL	0x60
PUBCOMP	0x70
SUBSCRIBE	0x80
SUBACK	0x90
UNSUBSCRIBE	0xA0
UNSUBACK	0xB0
PINGREQ	0xC=
PINGRESP	0xD0
DISCONNECT	0xE0

Tabla 2 - Tipos de mensaje MQTT

Al iniciar la conexión de un cliente (ya sea publicador o suscriptor), se envía un mensaje "CONNECT" al bróker, que al recibirlo responde con un mensaje "CONNACK" si acepta la conexión.

Los clientes suscriptores envían a continuación mensajes “SUBSCRIBE” al bróker en cuyo contenido incluyen el o los temas a los que desea suscribirse. El bróker responde con un mensaje “SUBACK” para confirmar cada suscripción.

Los clientes suscriptores envían mensajes “PUBLISH” en cuyo contenido incluyen el tema del mensaje y el mensaje en sí mismo. El bróker lee el mensaje y tras comprobar qué clientes se habían suscrito al tema de este, se los reenvía.

En este proyecto se usan dos temas:

- **Salida:** Para enviar los mensajes de movimiento y alarma
- **Estado:** Para enviar los mensajes de estado

4.2.4. Diseño de la aplicación web.

La aplicación web es accesible desde la dirección IP de la máquina servidor a través del puerto 1880. El interfaz gráfico es accesible desde la misma dirección y puerto, pero seguido de la carpeta “/ui/”.

Por ejemplo, si se accede directamente desde el servidor, la dirección será <http://localhost:1880> para la aplicación y <http://localhost:1880/ui/> para el interfaz gráfico.

Como se puede apreciar en el caso de uso de la Figura 33, la aplicación web se encuentra siempre disponible a la espera de recibir un mensaje MQTT, una vez lo recibe visualiza el movimiento o estado recibido y si se trata de un mensaje de alarma, además emite un audio.

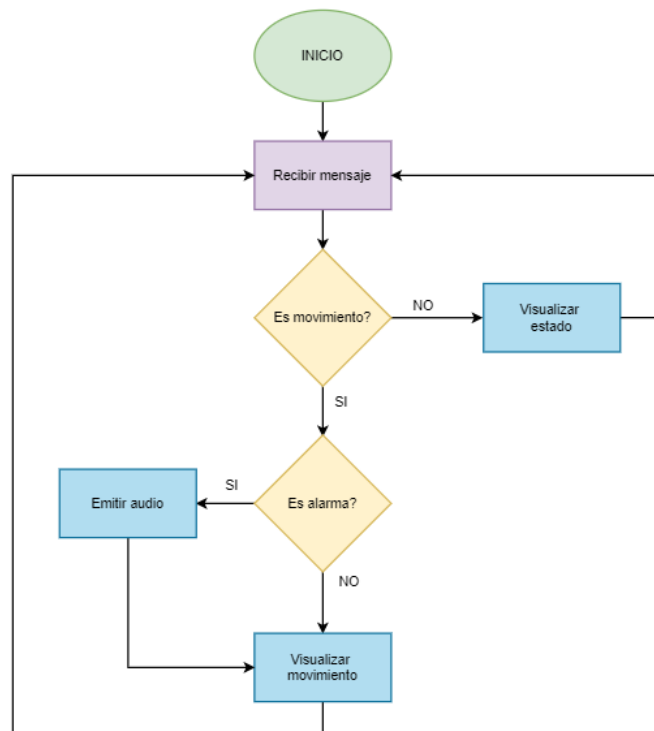


Figura 33 - Caso de uso de la aplicación web

Como se ha mencionado en apartados anteriores, para programar la aplicación web se ha utilizado Node-Red.

Se trata de una herramienta de desarrollo para programación visual, muy utilizada para la interconexión de dispositivos IoT, que se basa en la creación de flujos mediante la interconexión de piezas de código, denominadas nodos.

Estos nodos pueden ser de múltiples tipos diferentes y se pueden clasificar en 3 categorías básicas:

- Nodos de entrada: sólo admiten datos de entrada, que serán enviados a un destino final, como una base de datos, un cuadro de texto, un panel de control, etc.
- Nodos de salida: envían los datos no a otro nodo, sino a algún otro tipo de salida, como por ejemplo un mensaje MQTT, un WebSocket, a través de la interfaz gráfica, etc
- Nodos de entrada y salida: reciben datos de la entrada desde otros nodos, los tratan de alguna manera y luego los envían a través de una o varias salidas. En esta categoría pueden entrar un interruptor, un conversor de unidades de unidades para los datos de entrada, etc.

Para este proyecto se han creado dos flujos, uno para cada tema que publica el MSP432 y a los que se suscribe cada flujo.

El primer flujo es el de movimientos y alarma, denominado simplemente movimientos en la figura 33:

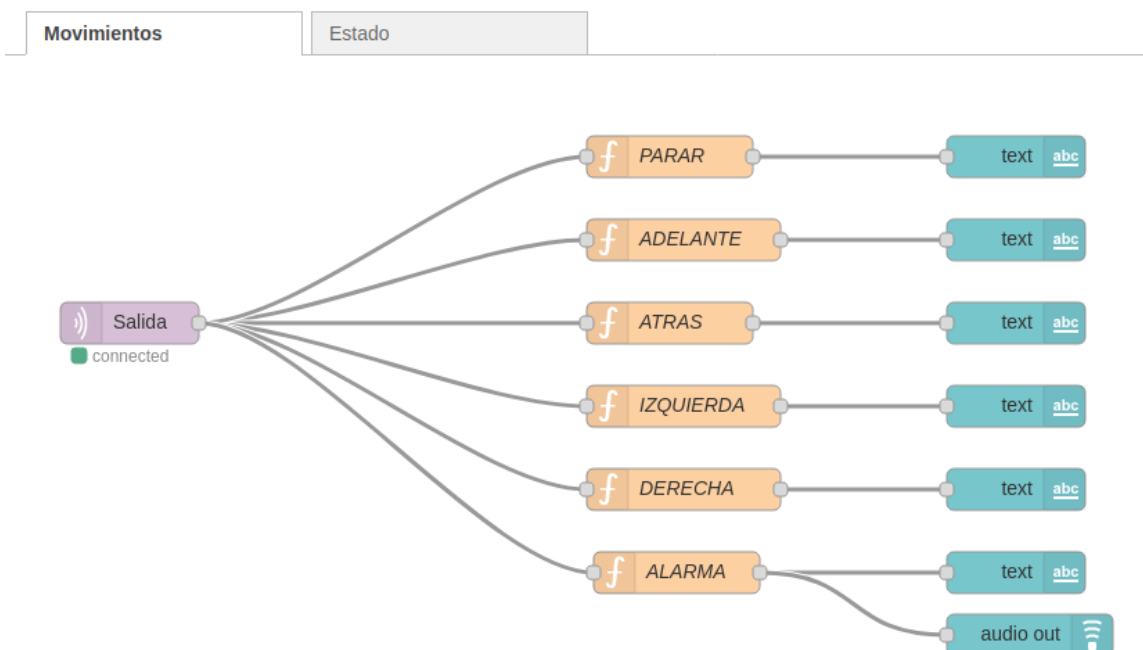


Figura 34 - Flujo de Movimientos en NodeRed

Como nodo de entrada tiene un nodo de recepción MQTT, que permite la suscripción al tema “Salida”.

Esta entrada pasa a 6 diferentes nodos de entrada y salida, del tipo función, donde se ha implementado una función que comprueba si el mensaje de entrada es el indicado. Si lo es, lo pasa a la salida y si no lo es, deja la salida en blanco.

Los nodos de salida son todos de visualización de texto a través de la interfaz gráfica, salvo uno extra en la función de alarma que genera una salida de audio TTS (Text To Speech).

Este audio será audible en cualquier navegador web compatible con la tecnología TTS, Mozilla Firefox, por ejemplo.

El segundo flujo es el de estado, como se ve en la Figura 34:

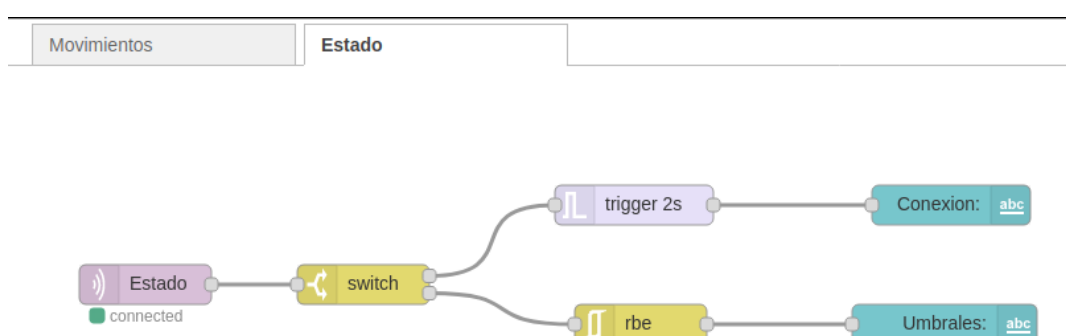


Figura 35 - Flujo de Estado en NodeRed

Como nodo de entrada tiene también un nodo de recepción MQTT, que permite la suscripción al tema “Estado”.

A continuación, se encuentra un nodo de entrada y salida de tipo “switch”, que dependiendo del mensaje recibido lo envía por un camino diferente.

Por el camino superior se envía el mensaje periódico de comprobación de la conexión Wi-Fi, pasa por un nodo de entrada y salida de tipo “trigger”, que si es un mensaje de “Conexión OK” lo mantiene a su salida durante el tiempo indicado, pero lo cambia por un mensaje “No conectado” si no se vuelve a recibir otro mensaje del mismo tipo antes de que acabe ese tiempo (ya que sin conexión no se puede enviar un mensaje de “No conectado”).

Por el camino inferior se envían los mensajes relativos al cálculo de umbrales.

Estos pasan por un nodo de tipo “rbe”, que mantiene el valor recibido hasta que se reciba uno diferente, momento en el que modificará el mensaje y mantendrá el nuevo mensaje recibido hasta que vuelva a cambiar.

Los nodos de salida, como en el caso anterior, son ambos de visualización de texto a través de la interfaz gráfica.

Dicha interfaz gráfica se configura desde la propia interfaz de Node-Red tras instalar la aplicación Node-Red Dashboard.

Una vez configurada para que se adapte a la pantalla de un teléfono móvil, el aspecto de la interfaz gráfica es el de la Figura 35:

Dirección	Dirección	Dirección	Dirección	Dirección	Dirección
IZQUIERDA	DERECHA	ADELANTE	ATRAS	PARAR	ALARMA
Conexión: CONECTADO	Conexión: CONECTADO	Conexión: CONECTADO	Conexión: CONECTADO	Conexión: CONECTADO	Conexión: CONECTADO
Umbral: UMBRALES OK	Umbral: UMBRALES OK	Umbral: UMBRALES OK	Umbral: UMBRALES OK	Umbral: UMBRALES OK	Umbral: UMBRALES OK

Figura 36 - Interfaz gráfica

5. Viabilidad técnica

Como se ha visto en capítulos anteriores, la solución propuesta se basa en la utilización de electrodos, un dispositivo de detección EMG y la plataforma de desarrollo MSP432 de Texas Instruments con un módulo WiFi y un servidor con Node-Red para la interfaz gráfica.

El Hardware es de fácil adquisición y la plataforma MSP432 tiene potencia más que suficiente para poder realizar todas las tareas necesarias para el proyecto.

Las primeras pruebas con los sensores muestran que es viable el uso de las señales EMG para el propósito del proyecto.

En la primera prueba, que podemos ver en la Figura 36, se ha usado un músculo de gran tamaño, el bíceps, y la señal detectada es muy clara.

En la imagen se pueden ver marcados en rojo 2 movimientos completos del bíceps (contracción y extensión rápida):



Figura 37 - Gráfica de movimiento del bíceps

La siguiente prueba se realizó en los músculos de la mejilla.

Como se puede observar en la Figura 37, la amplitud de la señal es bastante inferior, pero también se puede detectar un patrón de cada movimiento:



Figura 38 - Gráfica de movimiento de la mejilla inicial

Tras algunos cambios, como ampliar la resolución del ADC de 10 a 14 bits y cambiar la posición de los electrodos, como vemos en la Figura 38 se consigue una señal en la mejilla aún más clara que es posible analizar e interpretar:

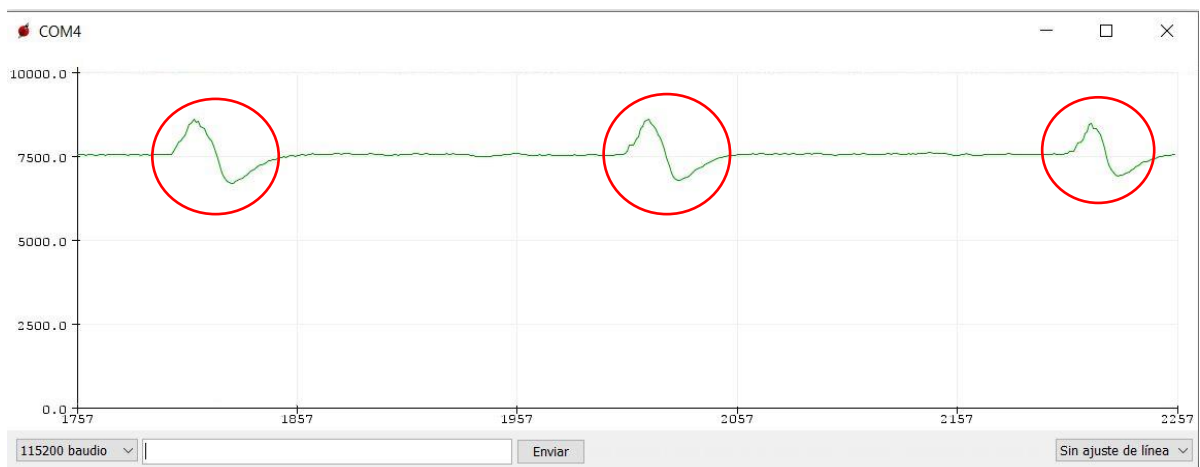


Figura 39 - Gráfica de movimiento de mejilla mejorado

Para la comunicación con el servidor, se pueden utilizar los protocolos REST o MQTT, por ejemplo, ambos muy extendidos (algo menos el segundo por ser más nuevo).

Debido a su mejor eficiencia se ha elegido MQTT, ya que además se está convirtiendo en un estándar para IoT.

Para el estudio de viabilidad es necesario valorar los posibles riesgos que pueden influir en el desarrollo del proyecto, dando una puntuación a cada uno de ellos en base a la probabilidad de que ocurra y el grado de impacto que tendría sobre el proyecto.

Finalmente, se añade la Tabla 3 indicando las contramedidas necesarias para contrarrestar o minimizar dicho riesgo.

Id	Descripción	Prob.	Impacto	Total	Acciones	Conclusión
1	Pandemia sanitaria	5	7	35	Seguir instrucciones de las autoridades. Adelantar trabajo.	Evitar el contagio
2	Fallo disco duro	2	9	18	Realizar copia de seguridad regularmente	Sin repercusión realizando la copia
3	Viaje de trabajo	5	3	15	Recuperar tiempo los fines de semana y avanzar durante los viajes	Planificar horas de imprevistos
4	Retraso con algoritmo de detección	3	5	15	Avanzar con la fase siguiente mientras se encuentran soluciones	Paralelizar tareas
5	Avería del prototipo	1	9	9	Fabricar 2 prototipos	Inviabile económicamente
6	Falta de consumibles	3	3	9	Comprar electrodos de sobra	Realizado pedido

Tabla 3 - Valoración de riesgos y contramedidas

Leyenda:

Rojo - Riesgo alto

Naranja - Riesgo moderado

Verde - Riesgo bajo

Por todo ello, se concluye que el proyecto es viable.

6. Valoración económica

6.1. Prototipo

En la Tabla 4 se muestran los costes del prototipo tanto de materiales como de mano de obra, estos son:

Material	Coste unitario	Unidades	Coste total
Placa desarrollo MSP432P401R	29.69€	1	29.69€
Modulo CC3100 BoosterPack	36.37€	1	36.37€
Sensor Grove EMG Detector	34.01€	2	68.02€
PC Servidor	600€	1	600€
Cables	5€	1 juego	5€
Porta-baterías 4xAA	4.50€	1	4.50€
Pilas AA	1.25€	4	5€
Total material			748,58€
Mano de obra			
Mano de obra desarrollo	30€	285	8550€
TOTAL			9298,58€
Total por unidad			9298,58€

Tabla 4 - Presupuesto del prototipo

Los IDEs que se han utilizado para programar son:

- Energía (Lenguaje Arduino/C)
- Node-Red (Programación visual)
- Node-Red Dashboard (Programación visual de interfaz gráfico)

Como se puede observar, los costes para la unidad de prototipo son muy elevados. Esto se debe a la utilización de una plataforma de desarrollo y al hecho de que, al ser una única unidad, los costes de compra del material y otros costes no recurrentes, como la mano de obra de desarrollo, son muy elevados.

A continuación, se realiza una estimación de industrialización de este prototipo para reducir los costes lo máximo posible de cara a una fabricación en masa.

6.2. Industrialización

Para la comercialización, se deben cumplir las directivas aplicables y en base a ellas conseguir la certificación del producto.

Para la reducción de costes y fabricación en masa se diseñará y fabricará una PCB personalizada para el módulo de control y los sensores, además de sustituir el módulo WiFi por uno mucho más económico, el ESP-01.

En la Tabla 5 se muestran los costes de materiales, mano de obra y certificaciones necesarias para la industrialización de 1000 unidades, estos son:

Material	Coste unitario	Unidades	Coste total
Modulo Control personalizado	7,5€	1000	7500€
PCB	1,5€		
Componentes	4€		
Montaje	2€		
Modulo Wifi ESP-01	1,5€	1000	1500€
Sensor EMG Personalizado	5,5€	1000	5500€
PCB	1€		
Componentes	3€		
Montaje	1,5€		
PC Servidor	600€	1	600€
Cables	1€	1000	1000€
Porta-baterías 4xAA	1€	1000	1000€
Pilas AA	0.4€	4000	1600€
Total material			18700€
Mano de obra			
Mano de obra desarrollo	30€	285	8550€
Mano de obra rediseño y adaptación	30€	40	1200€
Certificaciones			
Certificación IEC 60601-2-40 para dispositivos EMG [27]	10000€	1	10000€
Marcado CE dispositivo <75 VDC [28]	1000€	1	1000€
TOTAL			39450€
Total por unidad			39,45€

Tabla 5 - Presupuesto de industrialización para 1000 unidades

7. Conclusiones

7.1. Lecciones aprendidas

Este proyecto ha permitido poner en práctica muchos de los conocimientos adquiridos durante este grado, desde electrónica para la conexión de los diferentes componentes electrónicos a sistemas distribuidos para la comunicación entre varios dispositivos a través de un servidor.

También se han puesto en práctica conocimientos de programación para desarrollar el software, de señales y sistemas en la parte de adquisición de las señales, o de gestión de proyectos, entre otros muchos.

Por otro lado, se han obtenido conocimientos nuevos en tratamiento digital de señales, protocolos de comunicación, programación visual, o diseño de interfaces que han requerido de bastante tiempo adicional, pero que han sido útiles para la realización de un sistema usable.

Se ha aprendido también que pese a hacer una buena planificación inicial, a lo largo del proyecto surgen circunstancias que pueden alterarla y que requieren de adaptación, como la imposibilidad de completar el sistema motor por falta de tiempo y la compra tardía del material sin tener en cuenta la situación actual.

Finalmente, se ha aprendido que la labor de documentación requiere mucho más tiempo del que puede parecer inicialmente y la importancia que dicha documentación tiene.

7.2. Autoevaluación

Como se verá en el siguiente apartado, este proyecto ha sido un éxito ya que se han conseguido todos los objetivos principales y salvo uno, también todos los objetivos secundarios, consiguiendo un sistema funcional y aplicable también a muchos otros casos de uso con muy pocas modificaciones.

Esto ha sido posible gracias a la gran cantidad de horas invertidas en el proyecto y al seguimiento continuo de la planificación durante todo el proceso, que, aunque como se verá en el apartado 7.3.2 no se ha cumplido al cien por cien, el seguimiento ha sido muy cercano y las modificaciones han sido razonadas y sustituidas por características adicionales en caso de no cumplirse alguna.

La simultaneidad de este proyecto con otras asignaturas además de con un trabajo a jornada completa ha hecho más difícil llevarlo a cabo, y hubiera sido preferible tener la posibilidad de dedicarle aún más tiempo para poder haber finalizado todas las tareas opcionales, e incluso haber incluido alguna mejora más. Aun así, el resultado ha sido satisfactorio y ha merecido la pena el esfuerzo realizado.

7.2.1. Tiempo dedicación a cada PEC

En la Tabla 6 se muestra el tiempo dedicado a cada etapa del proyecto:

	PEC0	Propuesta	PEC1	PEC2	PEC3	Entrega Final	Memoria	Presentación	Total
Planificado	25	30	35	80	80	85	30	10	375
Real	25	30	35	75	90	80	40	10	385

Tabla 6 - Tiempo de dedicación

7.2.2. Logro de objetivos

Para llegar a una conclusión se analizará cada uno de los objetivos del proyecto.

Objetivos principales

- Detección de movimientos faciales vía EMG: Objetivo conseguido. La correcta colocación de los electrodos y los filtros implementados han permitido una muy buena calidad de señal para la detección de los movimientos faciales a través de señales electromiográficas.
- Caracterización y diferenciación de hasta 5 movimientos. (Adelante, atrás, izquierda, derecha y parar): Objetivo conseguido. El algoritmo de detección implementado es capaz de detectar no sólo los 5 movimientos que se planificaron inicialmente, sino un sexto movimiento extra.
- Comunicación inalámbrica vía wifi: Objetivo conseguido. La integración del módulo CC3100 ha servido para conseguir una comunicación inalámbrica por WiFi rápida y sencilla.
- Servidor de control y recepción de datos: Objetivo conseguido.
- Sistema de envíos de alarma en caso de accidente: Objetivo conseguido. Este objetivo no se encontraba en la planificación inicial, pero aun así se añadió como primario por considerarlo de gran importancia.
- Interfaz gráfica basada en cliente web: Objetivo conseguido. Se ha diseñado una interfaz gráfica sencilla y clara mediante Node-Red Dashboard.
- Bajo consumo: Objetivo conseguido. Tras medir consumos con el polímetro, se encuentra que el CC3100 consume 0,10A y los sensores 0,02A. El consumo se debe mayormente a la placa de desarrollo (0,24A), muy elevado respecto al consumo teórico de 80uA/Mhz (algo menos de 4mA para 48Mhz), lo que indica un problema en la PCB. Aun así, se trata de un consumo bastante reducido.

Objetivos secundarios

- Acceso desde teléfono móvil: Objetivo conseguido. El servidor web es accesible desde cualquier navegador, también desde el teléfono móvil, y tras algunos ajustes en la interfaz la visualización es buena.
- Alimentación con baterías: Objetivo conseguido. La tarea por la que se alimenta el sistema mediante baterías, si bien se ha realizado, no cumple el objetivo de durar 1 día completo, ya que el consumo es mayor de lo esperado (0,35A aproximadamente) por lo que la duración no será de 1 días, sino de unas 8 horas con pilas alcalinas de 2700mAh.
- Pequeño y portable: Objetivo conseguido. Pese a que se puede reducir el tamaño con el rediseño que se plantea en el apartado 6.2, el sistema es pequeño y gracias a la alimentación con baterías y a la comunicación inalámbrica por Wi-Fi es muy portable.
- Sistema motor remotamente controlado: Objetivo no conseguido. No ha sido posible construir el sistema motor ya que no se han recibido las piezas a tiempo, en parte debido a que, al tratarse de una tarea opcional, no se pidieron las piezas hasta tener el resto de las tareas casi realizadas, y no se tuvo en cuenta que debido a la situación actual derivada de la pandemia por COVID-19 tardan más de lo normal en llegar. Se debería haber realizado la compra con más antelación.

7.2.3. Seguimiento de la planificación

La planificación inicial fue algo conservadora y aun así en algunos momentos se ha necesitado más tiempo del calculado inicialmente.

Es el caso de la tarea de detección de los 5 movimientos necesarios, ya que al intentar ampliar el algoritmo inicial de detección de 2 movimientos se constató que no era viable usar la misma lógica para detectar más, por lo que se tuvo que cambiar el algoritmo por completo.

La tarea de escribir esta memoria también ha supuesto más tiempo del planificado inicialmente, pero afortunadamente al seguir la planificación y empezar a escribirla en paralelo a las últimas tareas, ha sido posible terminarla en la fecha planificada.

Dejando estos dos detalles a un lado, la planificación se ha llevado al día durante todo el proyecto y ha sufrido sólo algunos cambios menores, por lo que el tiempo invertido en realizarla ha sido muy útil.

7.3. Líneas de trabajo futuro

Una vez finalizado el proyecto, se proponen líneas de trabajo futuro para la mejora del sistema desarrollado.

La primera línea de trabajo es añadir las tareas opcionales no finalizadas para tener un sistema más completo, esto es:

- Construir el sistema motor.
- Reducir el consumo o ampliar las baterías para conseguir un día completo de autonomía.

Como mejora de lo ya implementado, en el algoritmo de detección los tiempos de espera son algo elevados. Esto se debe a la necesidad de esperar un tiempo mínimo para poder detectar los movimientos dobles. En el futuro se podrían buscar soluciones para mejorar esos tiempos, lo que haría que el sistema tenga un tiempo de respuesta más bajo, aunque ya es bastante rápido para las aplicaciones objetivo.

Otra línea de trabajo podría ser utilizar otro tipo de sensores para la detección de movimiento, como la visión artificial, en vez de usar la electromiografía para detectar los movimientos faciales y evitar así la conexión física de los electrodos.

8. Glosario

ADC – Dispositivo que permite convertir una señal analógica en una señal digital.

Broker – Servidor central del mecanismo publish/subscribe encargado de transmitir los mensajes entre publicadores y suscriptores.

CC3100 – Placa de desarrollo para el MSP432 que se conecta a este a través del bus SPI y le proporciona un interfaz WiFi.

Dashboard – Módulo para crear interfaces gráficas en Node-Red

EMA – “Exponential Moving Average”. Filtro de media móvil exponencial usado en el procesamiento de señales digitales.

EMG – Señal electromiográfica. Es una señal eléctrica producida por un músculo durante el proceso de contracción y relajación.

HTTP – “HyperText Transfer Protocol”. Protocolo de comunicaciones mediante el que se comunican los servidores y clientes web.

IoT – “Internet of Things”. Hacen referencia al internet de las cosas, un concepto que define a la capacidad de elementos de todo tipo de conectarse a internet y comunicarse.

M2M – “Machine to Machine”. Tipo de protocolo de comunicación que consiste en el intercambio de información entre 2 máquinas.

MSP432 – Placa de Desarrollo para plataformas embebidas de Texas Instruments.

Mosquitto – Broker gratuito para el protocolo MQTT.

MQTT – “Message Queue Telemetry Transport”. Protocolo de Comunicaciones para el internet de las cosas.

Node-Red – Herramienta de programación basada en flujo visual para conectar dispositivos que forman parte del IoT.

PCB – “Printed Circuit Board”. Tarjeta de circuito impreso.

Publish/Subscribe – Mecanismo de publicación/subscription, utilizado por MQTT.

SPI – “Serial Peripheral Interface”. Protocolo de comunicaciones para la transferencia de datos entre dispositivos electrónicos.

Topic – Tema que publica o al que se suscribe un nodo de un bróker MQTT.

VDC – Tensión en corriente continua.

WiFi – Tecnología para la conexión inalámbrica de dispositivos electrónicos

9. Bibliografía

[1] Roberto Merletti. "Electromyography: Physiology, Engineering, and Non-Invasive Applications". IEEE Press Series on Biomedical Engineering. Wiley-IEEE Press. 2004

[2] Web. Ecured, enciclopedia cubana. Electromiografía
<https://www.ecured.cu/Electromiograf%C3%ADa#Historia>
Fecha de visita: Mayo 2020

[3] John V Basmajian; Carlo J De Luca. "Muscles alive: their functions revealed by electromyography" : 5th ed. Williams & Wilkins. Baltimore. 1985

[4] Web. Wikifab. Prothèse de main commandée par des capteurs musculaires.
https://wikifab.org/wiki/Proth%C3%A8se_de_main_command%C3%A9e_par_des_capteurs_musculaires
Fecha de visita: Mayo 2020

[5] Web. Bret Contreras. What's All The Fuss About EMG?
<https://bretcontreras.com/whats-fuss-emg/>
Fecha de visita: Mayo 2020

[6] Web. Vernier video: EMG and Muscle Fatigue.
<https://www.vernier.com/video/emg-and-muscle-fatigue/>
Fecha de visita: Mayo 2020

[7] Web. ANT-Neuro. EEG and EMG rehabilitation.
<https://www.ant-neuro.com/show-case/eeg-and-emg-rehabilitation>
Fecha de visita: Mayo 2020

[8] Web. Wikipedia. Sistema embebido.
https://es.wikipedia.org/wiki/Sistema_embebido
Fecha de visita: Mayo 2020

[9] Web. Wikipedia. Historia del hardware de computadora.
[https://es.wikipedia.org/wiki/Historia_del_hardware_de_computadora_\(1960-presente\)](https://es.wikipedia.org/wiki/Historia_del_hardware_de_computadora_(1960-presente))
Fecha de visita: Mayo 2020

[10] Web. Wikipedia. Apollo guidance computer
https://es.wikipedia.org/wiki/Apollo_Guidance_Computer
Fecha de visita: Mayo 2020

[11] Web. Wikipedia. LGM-30 minuteman
https://es.wikipedia.org/wiki/LGM-30_Minuteman
Fecha de visita: Mayo 2020

[12] Web. Wikipedia. Arduino.
<https://es.wikipedia.org/wiki/Arduino>
Fecha de visita: Mayo 2020

- [13] Web. Wikipedia. Raspberry pi.
https://es.wikipedia.org/wiki/Raspberry_Pi
Fecha de visita: Mayo 2020.
- [14] Web. Wikipedia. IoT internet of things.
<https://www2.deloitte.com/es/es/pages/technology/articles/loT-internet-of-things.html>
Fecha de visita: Junio 2020
- [15] Web. Wikipedia. Sigfox.
<https://es.wikipedia.org/wiki/Sigfox>
Fecha de visita: Junio 2020
- [16] Web. Wikipedia. LoRaWAN.
<https://es.wikipedia.org/wiki/LoRaWAN>
Fecha de visita: Junio 2020
- [17] Web. HW Libre. MQTT.
<https://www.hwlibre.com/mqtt/>
- [18] Web. ENGTtotal. Nano EMG.
<http://www.engtotal.com/nanoemg.html>
Fecha de visita: Junio 2020
- [19] Web. Biomech Solutions. Ultium EMG.
<https://www.biomech-solutions.com/sistemas-emg-noraxon.html>
Fecha de visita: Junio 2020
- [20] Web. Wikipedia. Transferencia de estado representacional.
https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional
Fecha de visita: Mayo 2020
- [21] Web. Luisllamas. Principales bróker MQTT open source para proyectos iot.
<https://www.luisllamas.es/principales-broker-mqtt-open-source-para-proyectos-iot/>
Fecha de visita: Mayo 2020
- [22] Web. TME
<https://www.tme.eu/Document/8732078ad54d82f6b49c5284341026b5/POLOLU-2726.pdf>
Fecha de visita: Mayo 2020
- [23] Web. Sparkfun
<https://cdn.sparkfun.com/assets/a/3/a/f/a/AT-04-001.pdf>
Fecha de visita: Mayo 2020
- [24] Web. Seedstudio
https://wiki.seeedstudio.com/Grove-EMG_Detector/
Fecha de visita: Mayo 2020
- [25] Web. Mikroe
<https://www.mikroe.com/emg-click>
Fecha de visita: Mayo 2020

[26] Web. Wikipedia

https://es.wikipedia.org/wiki/Amplificador_de_instrumentaci%C3%B3n

Fecha de visita: Mayo 2020

[27] Web. QWE. IEC 60601

https://es.qwe.wiki/wiki/IEC_60601

Fecha de visita: Junio 2020

[28] Web. Europa.eu. Mercado CE

https://europa.eu/youreurope/business/product-requirements/labels-markings/ce-marking/index_es.htm

Fecha de visita: Mayo 2020