

**DESARROLLO DE UN ESTIMULADOR ELECTROMECAÁNICO
PROGRAMABLE PARA EL ENTRENAMIENTO
ELECTROMECAÁNICO DE CULTIVOS MONOCAPAS DE
CÉLULAS MIOCÁRDICAS**

Óscar Alberto Pla Terrada

Tutora: Dra. María de la Salud Guillem Sánchez

Cotutor: Dr. Andreu Martínez Climent

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 24 de julio de 2016

Resumen

Diversos estudios médicos han concluido que es posible regenerar tejido cardíaco a partir de células madre. A pesar de ello, la tasa de supervivencia de las células madre implantadas en corazones resulta inferior al 10%. Un precondicionamiento electromecánico de las células a implantar será efectivo a la hora de aumentar el efecto paracrino, puesto que las células estarían ya adaptadas a un entorno con estiramiento cíclico y, adicionalmente, el entrenamiento promoverá la diferenciación hacia tejido cardíaco adulto. Por ello, en este proyecto se ha diseñado e implementado un estimulador electromecánico programable para entrenar cultivos de células miocárdicas.

Se explicará cómo se ha planteado la solución al proyecto, empezando por el estimulador mecánico, continuando con el estimulador eléctrico y finalizando con la interfaz de usuario. Se detallará la coordinación entre los dos estímulos y las características y limitaciones de ambos. La interfaz de usuario basada en una pantalla táctil, se ha implementado y programado pensando en la sencillez de uso para el futuro usuario en un laboratorio.

Por último, pero no menos importante, se ha realizado un presupuesto de nuestro proyecto y se expondrán las conclusiones y propuestas de mejora.

Resum

Diversos estudis mèdics han conclòs que es possible regenerar teixit cardíac a partir de cèl·lules mare. A pesar d'això, la tassa de supervivència de les cèl·lules mare implantades en cors resulta inferior al 10%. Un precondicionament electromecànic de les cèl·lules mare a implantar serà efectiu a l'hora de augmentar l'efecte paracrino, ja que les cèl·lules estarien ja adaptades a un entorn amb estirament cíclic i, adicionalment, l'entrenament promourà la diferenciació cap a teixit cardíac adult. Per això, en aquest projecte s'ha dissenyat i implementat un estimulador electromecànic programable per a entrenar cultius de cèl·lules miocardiàques .

S'explicarà com s'ha plantejat la solució al projecte, començant per l'estimulador mecànic, continuant amb l'estimulador elèctric i finalitzant amb la interfície d'usuari. Es detallarà la coordinació entre els dos estímuls i les característiques i limitacions de ambdós. La interfície d'usuari basada en una pantalla tàctil, s'ha implementat i programat pensant en la senzillesa d'us per al futur usuari en un laboratori.

Per últim, però no menys important, s'ha realitzat un pressupost del nostre projecte i s'exposaran les conclusions i propostes de millora.

Abstract

Several medical studies have concluded that it is possible to regenerate cardiac tissue from stem cells. However, the survival rate of the stem cells implanted in hearts is less than 10%. A previous electromechanic conditioning of the stem cells to implant, can be effective to improve the paracrine effect, as the stem cells would be already adapted to an environment with cyclic elongation, and also, the training will promote the differentiation to an adult cardiac tissue. Therefore, in this project it has been designed and implemented an electromechanical programmable stimulator to train myocardial cell cultures.

It will be explained the solution proposed to the project, starting with the mechanical stimulator, followed by the electrical stimulator and ending with the user interface. The coordination between both stimuli and their characteristics and limitations will be detailed. The user interface based on a touch screen has been implemented and programmed thinking on simplicity for the future user in the laboratory.

Last but not least, it has been realized a budget of our project and the conclusions and the improvement proposals will be exposed.

Índice

Capítulo 1.	Introducción.....	5
Capítulo 2.	Antecedentes	6
2.1	Antecedentes clínicos	6
2.1.1	Tejido cardíaco artificial.....	6
2.1.2	Corazón Bioartificial	7
2.2	Antecedentes técnicos	8
2.2.1	Estimuladores Mecánicos	8
2.2.2	Estimuladores Eléctricos	9
2.3	Antecedentes del proyecto	11
Capítulo 3.	Objetivos	13
Capítulo 4.	Metodología de trabajo	14
4.1	Gestión del proyecto.....	14
4.2	Distribución en tareas	14
4.3	Diagrama temporal	15
Capítulo 5.	Desarrollo del trabajo	16
5.1	Estimulador mecánico	16
5.1.1	Diagrama de bloques	16
5.1.2	Diseño mecánico	17
5.1.3	Control electrónico	21
5.2	Estimulador eléctrico.....	24
5.2.1	Diagrama de bloques	24
5.2.2	Esquema eléctrico.....	25
5.2.3	Control electrónico	26
5.3	Interfaz de usuario	28
5.3.1	Hardware	28
5.3.2	Descripción de la interfaz	29
Capítulo 6.	Resultados	33
6.1	Estiramiento del pocillo.....	33
6.2	Estimulación electromecánica coordinada.....	34
Capítulo 7.	Pliego de condiciones	36
7.1	Características del equipo	36
7.2	Materiales y presupuesto	36

Capítulo 8.	Conclusiones y propuesta de trabajo futuro	38
Capítulo 9.	Bibliografía.....	39
Capítulo 10.	Anexos.....	42
10.1	Anexo I: Código del firmware del Arduino	42
10.1.1	Estimulador Mecánico	42
10.1.2	Interfaz de usuario a través de la Pantalla táctil	44
10.2	Anexo II: Descripción de componentes del sistema y funcionamiento	52
10.2.1	Estimulador Mecánico	52
10.2.2	Estimulador Eléctrico	55

Índice de figuras

Figura 1. Diferentes abordajes realizados en el campo de la ingeniería tisular cardíaca.....	6
Figura 2. Esquema gráfico de los diferentes pasos de la descelularización y recelularización de un corazón.....	7
Figura 3. Deformación uniaxial aplicada a un constructo de tejido bioartificial.....	8
Figura 4. Esquemático del proceso de compresión en los tejidos mediante el SteagePresser.	9
Figura 5. Presupuesto del sistema FX-5000.	9
Figura 6. Dibujo con ejemplo de uso del estimulador.	10
Figura 7. Morfología de una señal de estimulación estándar.	10
Figura 8. MyoPacer estimulador eléctrico celular.	11
Figura 9. Diagrama de los diferentes módulos del sistema de perfusión de aire.....	11
Figura 10. Imagen de la placa de estimulación.....	12
Figura 11. Diagrama general del estimulador electromecánico.	16
Figura 12. Soporte de los pocillos.	17
Figura 13. Pieza fija del estimulador mecánico.	18
Figura 14. Pieza móvil del estimulador mecánico.	18
Figura 15. Estimulador mecánico vista lateral.....	19
Figura 16. Estimulador mecánico vista de planta.	20
Figura 17. Conector DB9 para el conexionado del estimulador mecánico.	20
Figura 18. Tapa para resguardar las células.....	21
Figura 19. Arduino UNO.	22
Figura 20. Drive motor EasyDriver.....	22
Figura 21. Motor paso a paso.	23
Figura 22. Sensor Hall A3144.....	23
Figura 23. Diagrama de bloques del estimulador eléctrico.	24
Figura 24. Panel frontal del sistema.	25
Figura 25. Esquema de la generación de la señal bifásica de la estimulación.....	25
Figura 26. Fotografía PCB utilizada en estimulador.....	26
Figura 27. Arduino MEGA 2560.....	26
Figura 28. Diagrama de funcionamiento de la comunicación inalámbrica del sistema.....	27
Figura 29. Fotografía durante el montaje del estimulador electromecánico de 4 canales.	28
Figura 30. Señal de estimulación de 32 V con un periodo de 1000ms y un ancho de 200ms.	28
Figura 31. Vistas de la pantalla Arduino TFT Touch Shield V2.0.	29
Figura 32. Caja soporte para la pantalla táctil.....	29
Figura 33. Pantalla de inicio.	30
Figura 34. Pantalla de elección de canal de estimulación y detención de cualquier canal.	30
Figura 35. Pantalla de elección del parámetro que queremos modificar.	30

Figura 36. Pantalla de asignación del valor al parámetro deseado.....	31
Figura 37. Pantalla para controlar el valor progresivamente y acceso al menú de estimulación mecánica.	31
Figura 38. Pantalla de control del estimulador mecánico.	31
Figura 39. Sistema completo de estimulación electromecánica.....	32
Figura 40. (A) Pocillo estirado 5% (B) Pocillo estirado 10%	33
Figura 41. Número de pasos del motor en función del porcentaje de estiramiento del pocillo ...	34
Figura 42. Pulsos de estimulación eléctrica a diferentes tensiones y periodos.....	35
Figura 43. Gráfico de la forma de onda de la estimulación eléctrica y mecánica. T1: Tensión, T2: Ancho de pulso, T3: Periodo, T4: Porcentaje estiramiento y contracción, T5: Tiempo de espera con el pocillo estirado, T6: Tiempo de reposo entre estímulos, T7: Retardo entre la estimulación eléctrica y mecánica a causa de la electrónica.	35
Figura 44 – Esquema de conexiones del estimulador mecánico.	52
Figura 45 - Placa del Arduino Uno.....	53
Figura 46. Pinout del Arduino Uno.	54
Figura 47. Especificaciones del motor paso a paso.	55
Figura 48. Esquema de conexiones del estimulador eléctrico.	56

Capítulo 1. Introducción

La ingeniería tisular es una nueva tecnología que está desarrollándose con rapidez y cuyo objetivo es producir sustitutos viables que restauren, mantengan o mejoren las funciones del tejido y órganos humanos. A diferencia de las terapias estándar, los productos de la ingeniería tisular utilizan células vivas que se integran en el paciente [1].

Una célula madre es una célula capaz de replicarse y dar lugar a diversos tipos de células. Algunas de las células "hijas" se especializan mediante diferenciación, y otras mantendrán su capacidad replicativa y serán por tanto nuevas células madre. La existencia de este tipo de células permite mantener la capacidad regenerativa de los tejidos [2].

En los últimos años se han desarrollado numerosas estrategias y ensayos clínicos de terapia celular que han perseguido regenerar el tejido dañado mediante la administración de células con potencial cardiorregenerativo. En concreto, se han implantado células madre adultas procedentes de la médula ósea y tejido adiposo entre otros [3-6]. El principal efecto cardiorregenerativo de la inyección de células madre adultas es fundamentalmente paracrino, es decir, debido a la liberación de citoquinas y otros factores que potencian la vascularización del tejido [7-10]. Los resultados han demostrado que la implantación de células madre adultas es una técnica segura y mejora la función cardíaca, sin embargo, los índices de retención celular y supervivencia han sido inferiores al 10% [11].

Un precondicionamiento electromecánico de las células a implantar será efectivo a la hora de aumentar el efecto paracrino, puesto que las células estarían ya adaptadas a un entorno con estiramiento cíclico y, adicionalmente, el entrenamiento promoverá la diferenciación hacia tejido cardíaco adulto. Diversos autores han estudiado el efecto beneficioso de un entrenamiento eléctrico o mecánico en cultivos de células procedentes de corazones neonatos durante su maduración. Eschenhagen y Zimmerman demostraron la eficiencia de un entrenamiento mecánico durante el cultivo de células cardíacas neonatales [12,13]. Bajo estas condiciones, las células formaron músculo cardíaco altamente diferenciado que exhibía propiedades contráctiles y electrofisiológicas más similares al miocardio adulto que aquellos tejidos que no habían sido entrenados.

Estos estudios han puesto de relieve la importancia del entrenamiento en el proceso de maduración de tejido miocárdico. Es por ello que la principal finalidad del presente proyecto es la adaptación y aplicación de la tecnología necesaria para la construcción, entrenamiento y caracterización de parches de tejido bioartificial.

Capítulo 2. Antecedentes

2.1 Antecedentes clínicos

La insuficiencia cardíaca es la etapa final de muchas enfermedades cardiovasculares como el infarto agudo de miocardio, y sigue siendo uno de los retos más atractivos para la medicina regenerativa actual debido a su alta incidencia y prevalencia [14,15]. Los pacientes con disfunción cardíaca progresiva muestran un alto riesgo de muerte súbita. A pesar de los grandes avances de las últimas décadas, únicamente el trasplante cardíaco consigue restablecer completamente la función cardíaca aunque su uso está limitado por el escaso número de donantes y no está exento de complicaciones. El infarto agudo de miocardio se produce cuando se interrumpe el aporte sanguíneo del corazón provocando una situación irreversible de isquemia miocárdica, pérdida de células musculares cardíacas y formación de una cicatriz no contráctil [16]. Por ello, es necesario desarrollar estrategias terapéuticas capaces de proporcionar las bases adecuadas para asegurar una rápida reconstrucción del tejido afectado y renovar eficazmente su capacidad contráctil.

2.1.1 Tejido cardíaco artificial

El microambiente en que residen las células regenerativas es determinante para el mantenimiento de sus propiedades básicas y su función. La señalización y respuestas de estas células dependen, en gran medida, de sus interacciones con los componentes de la matriz extracelular en las que residen [17]. Se han desarrollado matrices muy semejantes en cuanto a propiedad fisicoquímicas a la matriz extracelular fisiológica a partir de materiales naturales (Figura 1), como el alginato o mezclas de colágeno, entre otros. Su principal ventaja es que se trata de materiales altamente maleables que permiten variar su forma y su tamaño según las necesidades del individuo receptor.

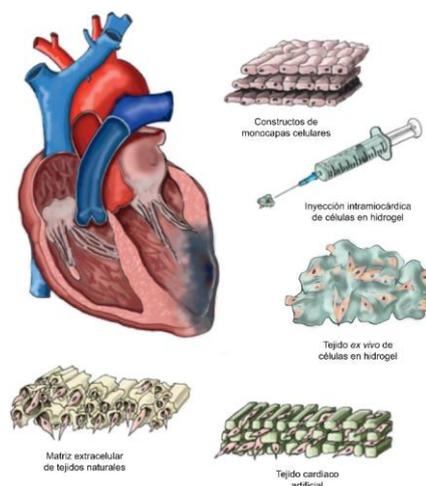


Figura 1. Diferentes abordajes realizados en el campo de la ingeniería tisular cardíaca [18].

En condiciones fisiológicas, el estiramiento mecánico de los cardiomiocitos está inducido por las señales eléctricas cardíacas y el acoplamiento entre el pulso eléctrico y las contracciones celulares, y es crucial para el desarrollo del miocardio [19]. Así, es de vital importancia que las técnicas de ingeniería tisular cardíaca garanticen un acoplamiento celular electromecánico y una función contráctil adecuadas dentro de los constructos generados. Otro abordaje se basa en la inducción de contracciones sincronizadas en la matriz celular mediante estimulación eléctrica [20,21]. Como resultado, los pulsos eléctricos dan lugar a una notable organización ultraestructural y acoplamiento celular de los cardiomiocitos residentes en la matriz.

Sin embargo, los tejidos cardíacos bioartificiales generados hasta la fecha utilizando cardiomiocitos neonatales, células madre embrionarias o células pluripotenciales inducidas no han logrado todavía alcanzar las propiedades mecánicas y eléctricas similares a las del tejido adulto y, por este motivo, todavía no están aconsejadas para su aplicación a corazones adultos, debido a que introducirían heterogeneidades eléctricas que previsiblemente provocarán la aparición de arritmias [22-24].

2.1.2 Corazón Bioartificial

Los estudios de decelularización con matriz extracelular constituyen la prueba de concepto para la obtención de corazones decelularizados. Hasta la fecha, los procesos de decelularización por inmersión directa han sido suficientes para generar matrices de soporte a partir de distintos tejidos cardiovasculares, incluyendo la pared vascular, el pericardio y las válvulas vasculares. En cambio, para poder decelularizar un corazón entero, se ha demostrado que la perfusión coronaria con detergentes es el método más eficaz. Se ha conseguido decelularizar corazones cadavéricos, obteniendo una compleja matriz extracelular cardíaca con el árbol vascular preservado, válvulas competentes y la geometría intacta de las aurículas y los ventrículos [25,26]. Posteriormente, recelularizando estos constructos con células cardíacas neonatales mediante perfusión coronario en un biorreactor simulando la fisiología cardíaca y favoreciendo la maduración del órgano (Figura 2).

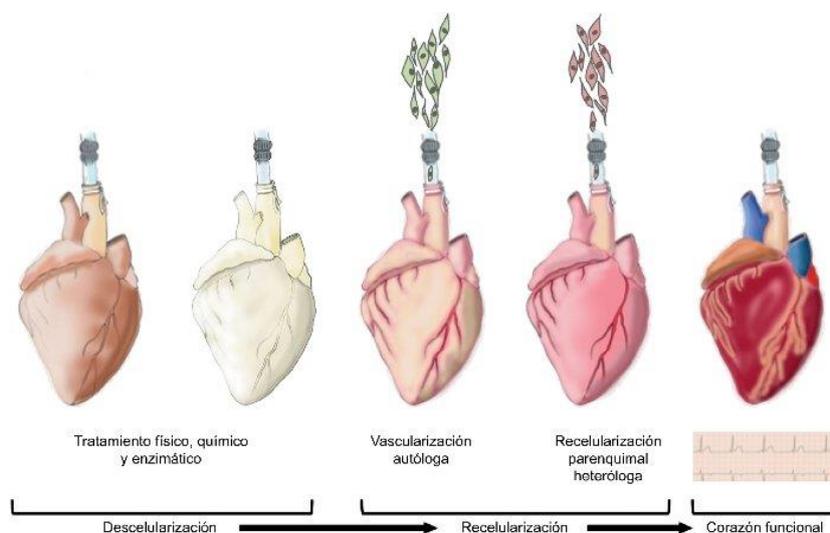


Figura 2. Esquema gráfico de los diferentes pasos de la decelularización y recelularización de un corazón [18].

Estudios más recientes han reportado la decelularización de corazones porcinos como modelo escalable al humano [27]. Un aspecto importante que hay que considerar en la aplicación de órganos enteros o matrices bioartificiales, es la necesidad de obtener una correcta vascularización sin la cual la viabilidad del constructo pueda verse comprometida. Con la decelularización de corazones enteros se ha conseguido mimetizar el tejido miocárdico

vascularizado y repoblarlo con células a través de estructuras vasculares con cierto grado de preservación [25]. A pesar de su éxito, esta estrategia puede verse limitada respecto a las tecnologías disponibles para la expansión de células a gran escala, particularmente de cardiomiocitos, necesaria para la repoblación de todo el órgano. Todavía queda camino por recorrer hasta poder disponer de un corazón bioartificial para su uso en humanos, sin embargo la tecnología avanza a una velocidad vertiginosa. El objetivo es que dentro de pocos años un paciente pueda tener su corazón completamente repoblado con células nuevas y funcionales, a partir de sus propias células o con otras células, implantado en él y funcionando.

2.2 Antecedentes técnicos

2.2.1 Estimuladores Mecánicos

Para la formación de tejido *in vitro* que sea estructuralmente y funcionalmente viable se requiere una serie de condiciones básicas, tales como: células, matriz, medios y factores de crecimiento y estimulación mecánica. Esta última condición es fundamental para crear un tejido estructuralmente robusto. Estos sistemas son capaces de aplicar una tensión o compresión de una duración cíclica estática o variable que permiten cultivar células *in vitro*.

En el mercado podemos encontrar estos sistemas pero hay poca variedad, ya que pocas empresas se dedican a la fabricación de estos equipos por ser muy especializados y exclusivos. Por ello, se ha realizado una búsqueda exhaustiva de las diferentes opciones que podemos encontrar en el mercado.

Flexcell es una empresa especializada en equipamiento para investigadores científicos en mecanobiología e ingeniería tisular. Podemos encontrar sistemas patentados y computerizados como puede ser el FX-5000 Tension System. Este sistema utiliza la presión de vacío regulado para deformar placas de cultivo de fondo-flexibles donde se puede aplicar hasta un 20% de deformación uniaxial (Figura 3).

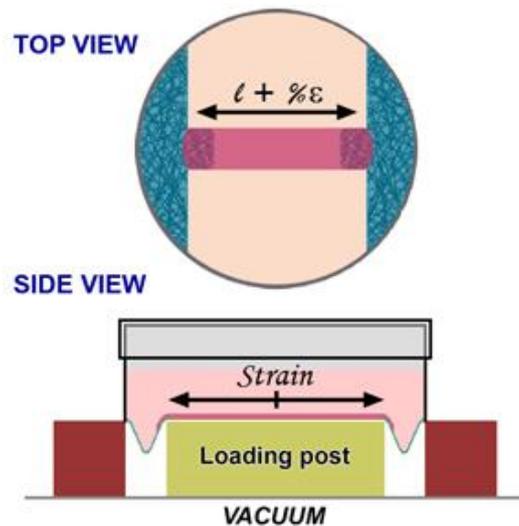


Figura 3. Deformación uniaxial aplicada a un constructo de tejido bioartificial [28].

También podemos encontrar sistemas de compresión como es el StagePresser (Figura 4). Este sistema utiliza un pistón adherido a una membrana de goma para aplicar una determinada fuerza controlada a una muestra de cultivo.

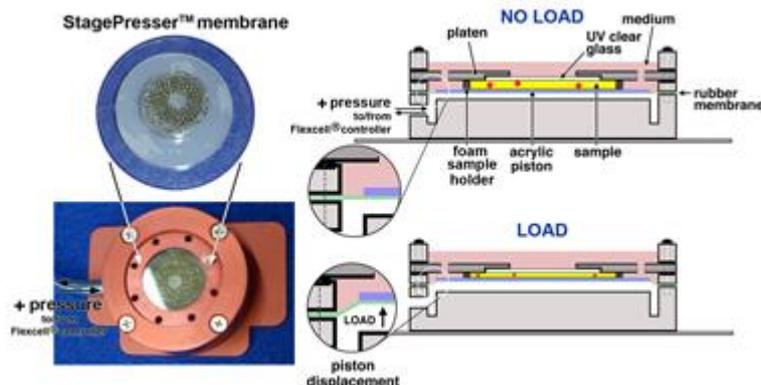


Figura 4. Esquemático del proceso de compresión en los tejidos mediante el StagePresser [28].

Como hemos observado podemos encontrar algunos de estos equipos en el mercado pero tienen un gran inconveniente, al ser productos tan especializados los precios son muy elevados. A continuación en la Figura 5 se mostrará un presupuesto de estos equipos solicitado en 2012.

Pos.	Qty.	Cat.-no.	Description	Price/Euro/ pc./cs.
01	1 pc.	FX-5000TT	Tissue Train Culture System 3-D cell culture in a gel matrix with or without cyclic uniaxial tension. Incl. desktop computer specifically designed to work with the Flexcell system with monitor, mouse and keyboard, installed FlexSoft FX5000 software, FX5000 Tension FlexLink controller, FX5000 TissueTrain system accessory kit (TissueTrain baseplate & gasket, TissueTrain arctangle loading posts (set of 4), TissueTrain Trough Loading posts (set of 4), drying filter, water trap, grease, associated peripherals, Flex in, Flex out, vacuum source tubing, 4 sample linear TissueTrain plates, miscellaneous parts).	38.683,00
02	1 pc.	11255	Trivac D8B Vacuum Pump (required for TT-System!) incl. AR4-8 Exhaust Filter	3.820,00
03	1 cs. (case/10)	TT-4001U/10	Tissue Train Culture Plates, untreated to work with the FX-5000TT to create linear, tethered 3-D cell-seeded gel constructs and to provide uniaxial cyclic strain to cells in a gel matrix.	327,00
04			Freight, installation and instruction	1.800,00

Figura 5. Presupuesto del sistema FX-5000.

2.2.2 Estimuladores Eléctricos

Los estimuladores eléctricos se encuadran entre los sistemas de instrumentación médica utilizados con fines terapéuticos, prótesis e investigación. Existen una gran variedad de tipos para distintas aplicaciones (cardíacos, musculares y nerviosos entre otros).

En nuestro caso utilizaremos el estimulador eléctrico para provocar un impulso eléctrico en la matriz celular. Los pulsos eléctricos dan lugar a una notable organización ultraestructural y acoplamiento celular de los cardiomiocitos residentes en la matriz.

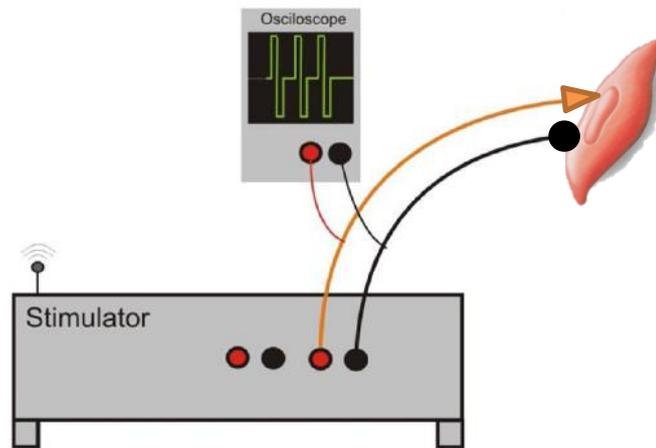


Figura 6. Dibujo con ejemplo de uso del estimulador.

La morfología de señal más utilizada para estimulación cardíaca es la mostrada en la Figura 7. Se trata de una señal bifásica con un ancho, periodo y amplitud configurable por parte del estimulador.

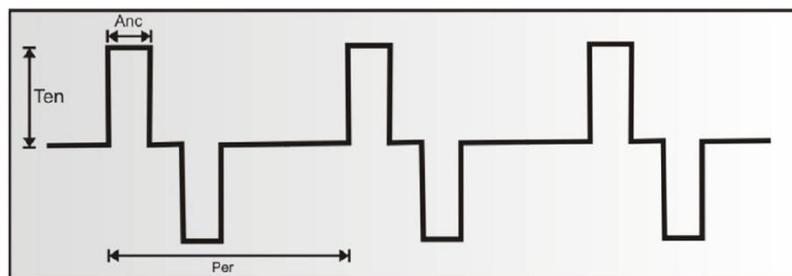


Figura 7. Morfología de una señal de estimulación estándar.

La tensión normalmente está comprendida entre 4V y 40V. Este valor se ajusta dependiendo de las características del experimento.

En este caso también hemos encontrado sistemas disponibles en el mercado. La empresa Ionoptix fabrica estimuladores celulares diseñados especialmente para la comunidad de investigación cardíaca como es el caso de MyoPacer (Figura 8). Este estimulador permite pre-programar cinco frecuencias para permitir cambios de frecuencias fáciles y rápidos. Tiene la capacidad de emitir estímulos bipolares y la tensión (± 40 V), la frecuencia (0.010 a 99 Hz), y la duración (0.4 ms a 90 ms) se pueden ajustar fácilmente desde el panel frontal.



Figura 8. MyoPacer estimulador eléctrico celular [29].

2.3 Antecedentes del proyecto

Este trabajo se enmarca dentro de una línea de investigación sobre ingeniería tisular cardíaca ligada a una colaboración entre la Universidad Politécnica de Valencia y el Hospital Gregorio Marañón, donde se desarrollan y realizan diversos estudios mediante experimentación con cultivos monocapas de células miocárdicas.

Al principio de este proyecto existían diversos prototipos de los sistemas ya presentados que se habían realizado en el mismo instituto de investigación. Respecto a la estimulación mecánica se había creado un prototipo que consistía en un sistema de perfusión de aire (Figura 9). En unos pocillos de goma unidos al dispositivo se colocaban las células. Una vez colocados los pocillos, un globo colocado debajo de la membrana delgada de silicona se inflaba usando una fuerte presión con lo que provocaba un estiramiento en la capa de cultivo.

Este sistema constaba de cuatro componentes: plataforma de cultivo celular, una bomba peristáltica, una electroválvula y el sistema de control.

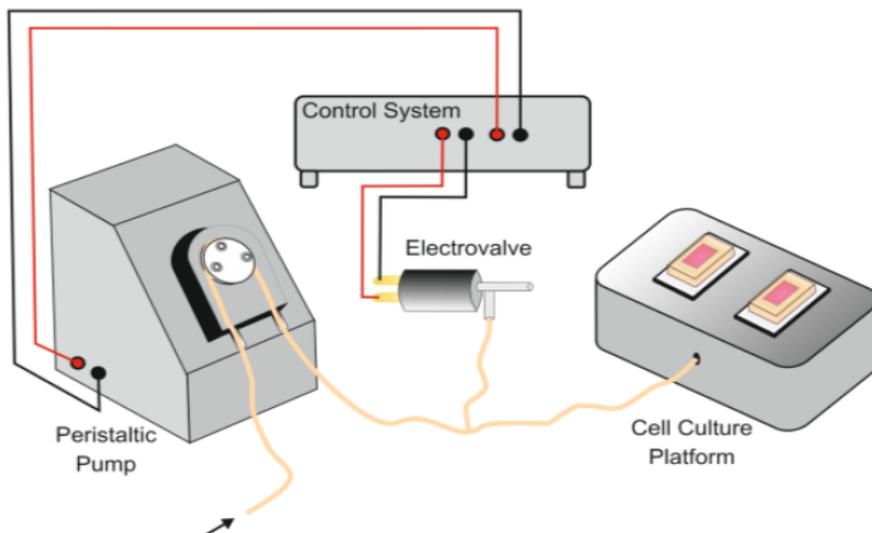


Figura 9. Diagrama de los diferentes módulos del sistema de perfusión de aire [30].

Como se puede observar, este equipo crea un estímulo de abajo a arriba, es decir, transversal. Después de realizar diversos experimentos con este equipo se concluyó que la velocidad de conducción de los canales iónicos todavía era muy inferior a la observada in vivo.

Por ello, se ha desarrollado en este trabajo otro prototipo que en este caso crea un estímulo longitudinal como otra posibilidad de estudio respecto a la estimulación mecánica de células miocárdicas.

Por el otro lado, respecto a la estimulación eléctrica, se disponía de una placa de estimulación de dos canales y comunicación a través de cable USB. Este estimulador permitía la aplicación de estímulos eléctricos de manera independiente a través de dos canales, así como la aplicación de estímulos programados temporalmente para poder realizar cambios de frecuencias a lo largo de la estimulación. Era capaz de aplicar estímulos a una tensión de 40V y un máximo de corriente de 1A por canal (Figura 10).

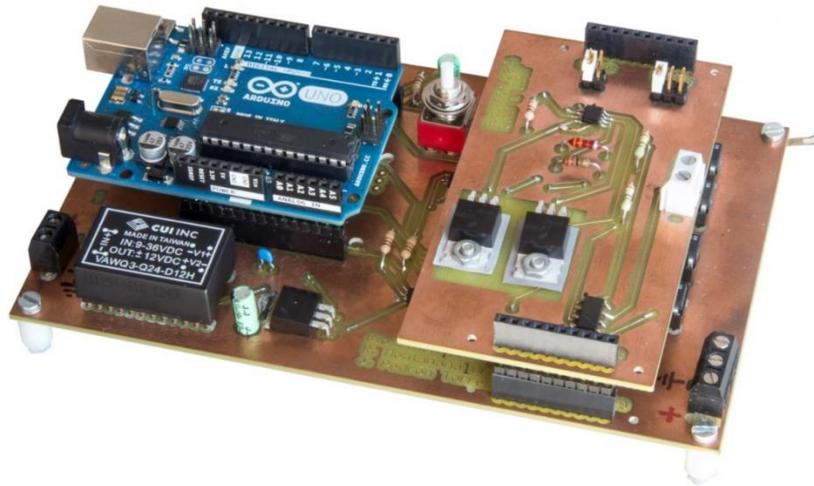


Figura 10. Imagen de la placa de estimulación.

No obstante, este prototipo aún no estaba preparado para ser utilizado en un entorno de laboratorio ni disponía de comunicación inalámbrica. Por ello, se decidió mejorar este prototipo tanto para alargar su vida útil como para aumentar sus funciones y capacidades antes de ser integradas en el conjunto del sistema.

A diferencia de los productos comerciales, estos diseños tienen diversas ventajas. Por un lado, el código del firmware puede ser actualizado cuando deseemos para modificar su funcionalidad. Además, pueden ser replicados y ampliados fácilmente para adaptarlos a las nuevas necesidades que requieran los experimentos. Por último, son sistemas muy versátiles a precios muy económicos.

Capítulo 3. Objetivos

El objetivo del proyecto es el desarrollo e implementación de un estimulador electromecánico programable para el entrenamiento de células en cultivos de monocapas sobre bases flexibles. Para ello se tienen que cumplir los siguientes objetivos:

- Diseño del estimulador mecánico, elaboración de un presupuesto y compra de material.
- Programar el firmware del Arduino para permitir la generación de un estímulo mecánico programable.
- Coordinar el estímulo eléctrico y mecánico bajo diferentes protocolos de entrenamiento.
- Incorporar una interfaz de usuario mediante una pantalla gráfica táctil.
- Diseñar y montar las piezas de nuestro estimulador mecánico que serán impresas en una impresora 3D.
- Testear el funcionamiento completo del estimulador bajo las condiciones necesarias para su prueba en un laboratorio de electrofisiología.

Capítulo 4. Metodología de trabajo

4.1 Gestión del proyecto

Este proyecto se realizará en el Instituto de investigación de ingeniería biomédica ITACA de la Universidad Politécnica de Valencia y en colaboración con el Hospital General Universitario Gregorio Marañón, bajo la supervisión de la Dra. María Guillem Sánchez, el Dr. Andreu Martínez Climent, y D. Ramón Albert Martínez.

Nuestro equipo será enviado al Hospital General Universitario Gregorio Marañón con el fin de ser testado, utilizando el equipo para llevarse a cabo diversos experimentos con el fin de determinar su validez.

Durante la realización del proyecto, se han efectuado reuniones periódicas conjuntas con todo el equipo de investigación para analizar los resultados obtenidos hasta la fecha y para comprobar que estos resultados se ciñen a las especificaciones impuestas.

El grupo COR del Instituto ITACA colaborará en este proyecto facilitando los materiales y herramientas necesarias para su realización.

4.2 Distribución en tareas

1. Estudio de diseños de estimuladores eléctricos y mecánicos ya desarrollados en el grupo de investigación, así como de una búsqueda bibliográfica para comprobar el estado del arte.
2. Búsqueda y compra de un módulo de programación por el usuario mediante pantalla táctil.
3. Programación del firmware del microcontrolador de la pantalla táctil.
4. Desarrollo y compra de material del estimulador mecánico.
5. Programación del firmware del microcontrolador del estimulador mecánico y ajustar el firmware del estimulador eléctrico.
6. Montaje de los componentes del estimulador electromecánico en una caja para su traslado al laboratorio donde realizarán las pruebas y colocación de conectores entre los dos estimuladores para facilitar su conexión.
7. Montaje de la pantalla táctil al módulo del estimulador.

4.3 Diagrama temporal

	JUL	AGO	SEP	OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
1												
2												
3												
4												
5												
6												
7												

Capítulo 5. Desarrollo del trabajo

5.1 Estimulador mecánico

5.1.1 Diagrama de bloques

En esta sección mostraremos los diferentes elementos que conforman nuestro estimulador mecánico y la elección de sus componentes. A continuación podemos observar el diagrama de bloques que presenta nuestro diseño para entender mejor su funcionamiento.

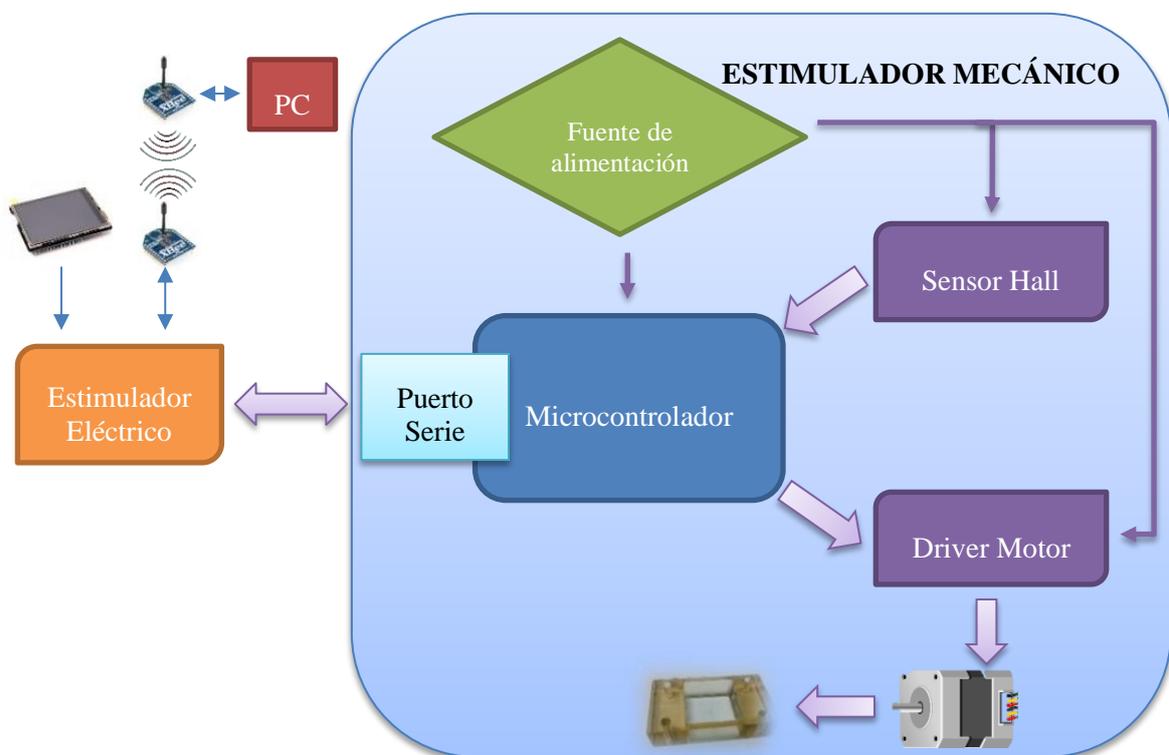


Figura 11. Diagrama general del estimulador electromecánico.

En la Figura 11 podemos observar el funcionamiento general de los módulos que requerirá nuestro sistema y la conexión entre ellos. Hay que destacar que nuestro sistema dispondrá de

tres Arduinos, para la estimulación eléctrica, para la estimulación mecánica y para el control de la pantalla táctil.

La pantalla táctil se ha programado con el fin de poder cambiar los parámetros de manera rápida e intuitiva. La pantalla está conectada a un Arduino UNO el cual está programado para enviar las tramas correspondientes que recibirá el microcontrolador del estimulador eléctrico. Estas tramas enviadas por puerto serie, activaran o desactivaran tanto la estimulación mecánica como la eléctrica pudiendo cambiar los parámetros de nuestro estímulo; tensión, ancho de pulso y periodo del estímulo. Además nos permite a su vez seleccionar cuál de los canales queremos configurar.

Por otra parte, el microcontrolador del estimulador eléctrico está constantemente atento a la recepción de tramas de otro puerto serie por si queremos configurar los parámetros de forma inalámbrica mediante XBee. Esto además nos permitirá aplicar protocolos de estimulación ya programados.

El estimulador mecánico utilizará un sensor hall que hará la función de final de carrera. Así pues, podremos volver a colocar los pocillos de goma en su posición de reposo.

Cabe destacar que los dos estimuladores actuarán de manera coordinada mediante una interrupción en el estimulador mecánico. Esta interrupción se activará a través de una señal cuadrada sincronizada con la señal de estimulación eléctrica que se generará en el estimulador eléctrico.

Por último, para satisfacer las necesidades de alimentación, se ha instalado una fuente de 12 V y 1,3 A que proporcionarán energía tanto al driver del motor como a la electrónica de control del módulo de estimulación mecánico y eléctrico.

5.1.2 *Diseño mecánico*

Para la realización del diseño de la parte mecánica de nuestro sistema se ha utilizado el programa FreeCad. Se trata de una aplicación libre de diseño asistido por ordenador en tres dimensiones para el diseño de elementos mecánicos. Esta aplicación nos ha permitido, diseñar las piezas de nuestro estimulador mecánico.

La parte mecánica del sistema consta de dos piezas impresas, una de ellas fija y la otra móvil. La pieza fija hará la función de base y soportará uno de los lados del pocillo de goma. Estos pocillos tienen dos agujeros en cada lado para fijarlos a la superficie que nosotros deseemos. En nuestro caso, se dibujaron dos agujeros con la forma de una tuerca para así ensamblarlas y poder atornillar los pocillos a la pieza (Figura 12). Además, esta pieza hará de soporte para el motor paso a paso que será atornillado a ella (Figura 13).

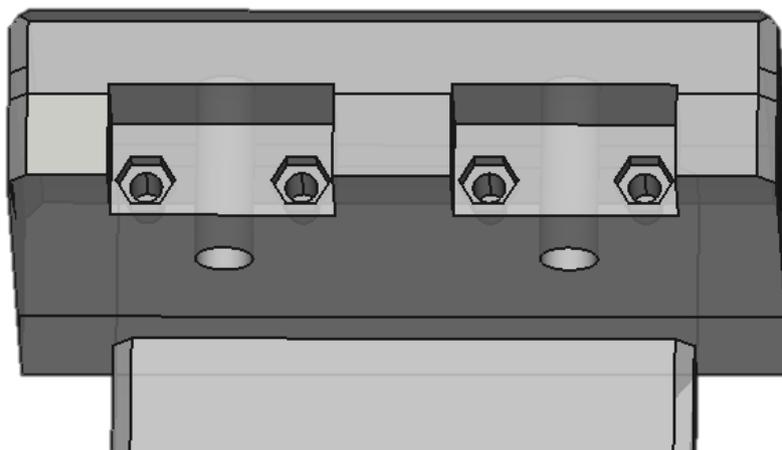


Figura 12. Soporte de los pocillos.

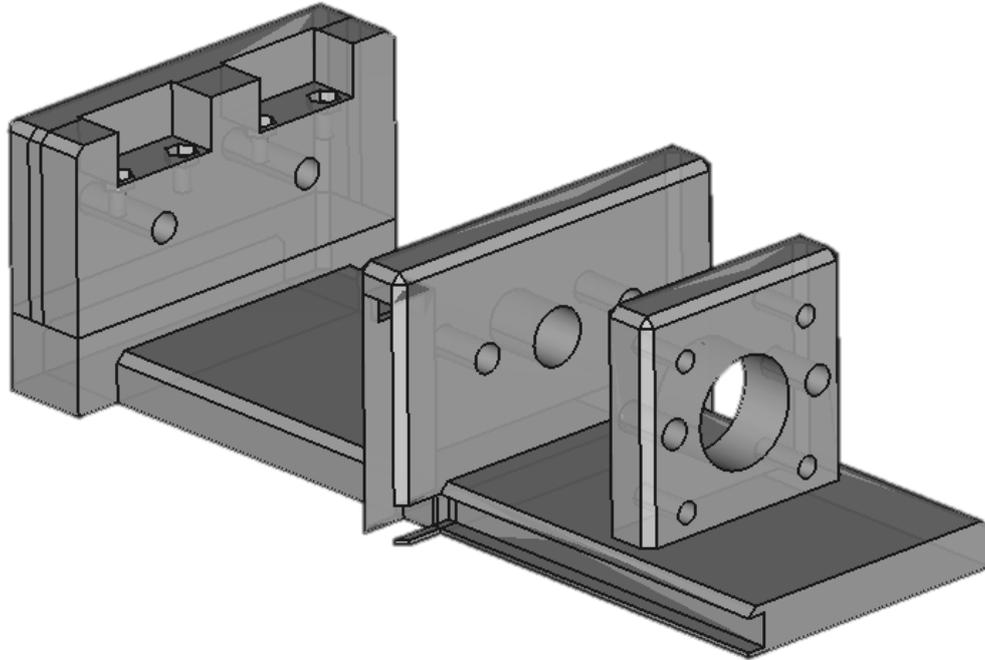


Figura 13. Pieza fija del estimulador mecánico.

Como podemos observar en la Figura 13 se ha dejado una ranura en la parte inferior de la pieza fija para colocar un sensor hall que hará la función de final de carrera y para poder colocar los cables ordenadamente.

Por otro lado, la pieza móvil (Figura 14) soportará el otro extremo del pocillo de goma. Esta pieza se moverá hacia delante y hacia atrás provocando un estímulo mecánico el cual será producido mediante una varilla roscada enganchada al eje del motor paso a paso. En el centro de la pieza, encontraremos una tuerca de diámetro 8 mm para producir el movimiento y a los laterales dos rodamientos con sus respectivas varillas que harán de sujeción y deslizamiento de la misma. Por ello, se han realizado los agujeros correspondientes para el encaje de estas piezas. Además, albergará un imán enfrentado al sensor hall para su activación/desactivación.

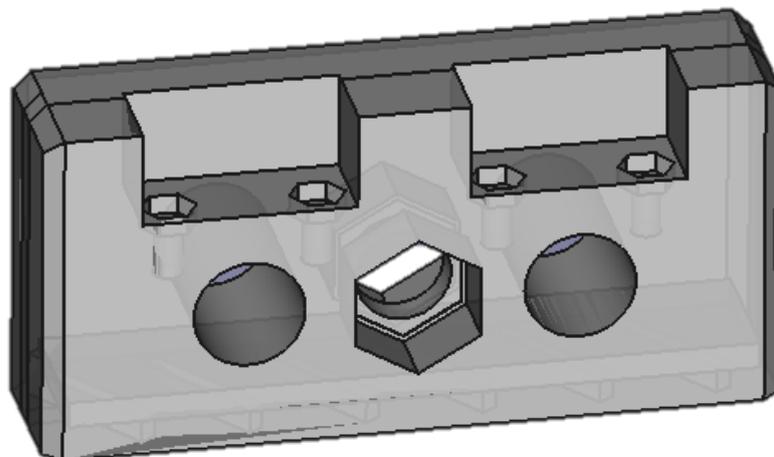


Figura 14. Pieza móvil del estimulador mecánico.

A continuación, se detallarán los componentes utilizados para el montaje final de nuestro sistema. Los componentes que requeríamos eran los siguientes:

- Una impresora 3D para la fabricación de las piezas del estimulador.
- Dos varillas de 10 cm cada una para la sujeción de la pieza móvil.
- Dos rodamientos para el deslizamiento de la pieza móvil.
- Una tuerca y varilla roscada de 8 mm para provocar el movimiento de estimulación.
- Ocho tuercas y ocho tornillos de 2.5 mm para la colocación de los pocillos de goma.
- Un motor paso a paso.
- Driver del motor paso a paso.
- Un microcontrolador programable.
- Sensor Hall para final de carrera y un imán que activará o desactivará nuestro sensor.
- Fuente de alimentación 12 V para alimentar el driver del motor y el microcontrolador.

Una vez tuvimos nuestros diseños acabados se imprimieron con una impresora 3D, en nuestro caso hemos utilizado la impresora *Replicator 2* de la casa *MakerBot* [31].

Así pues, finalmente se realizó el montaje y encaje de todas las piezas (Figuras 15 y 16).

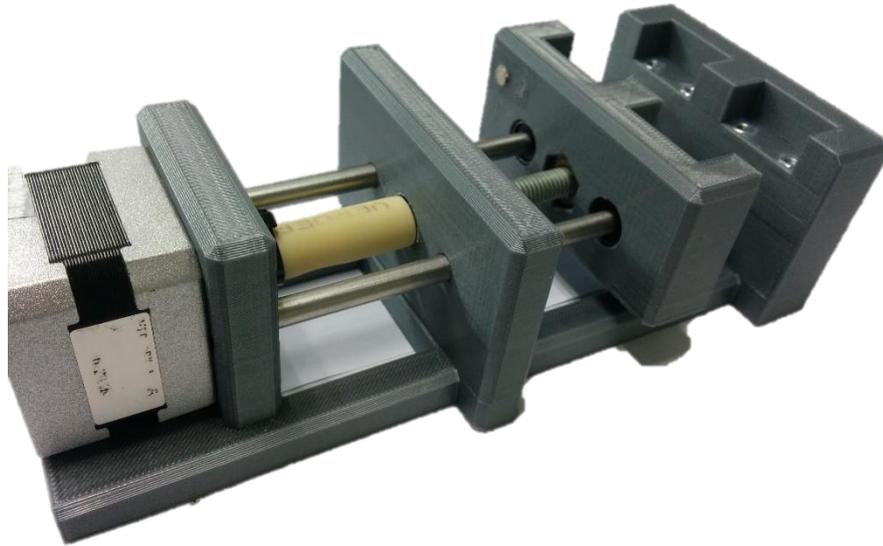


Figura 15. Estimulador mecánico vista lateral.



Figura 16. Estimulador mecánico vista de planta.

Para hacer más fácil la conexión para el usuario, se ha instalado un conector DB9 (Figura 17) soldando los cables a los pines que necesitábamos, en nuestro caso cuatro para el motor y tres para el sensor hall.



Figura 17. Conector DB9 para el conexionado del estimulador mecánico.

Por último, se ha impreso en 3D una tapa que resguarda las células de cualquier partícula no deseada (Figura 18), ya que el sistema estará en una campana de flujo laminar donde se emplea un ventilador para proporcionar aire limpio a la zona de trabajo.



Figura 18. Tapa para resguardar las células.

5.1.3 Control electrónico

En esta sección mostraremos los componentes electrónicos necesarios para la creación de un estímulo mecánico. Se analizarán cada uno de ellos para entender con más detalle su funcionamiento y su elección.

5.1.3.1 Microcontrolador

En nuestro diseño necesitamos un microcontrolador para registrar y generar una serie de señales programables.

Dicho microcontrolador deberá cumplir las siguientes especificaciones:

- Deberá poder conectarse vía USB para poder reconfigurar el microcontrolador de manera sencilla.
- Necesitaremos controlar las señales (dir) para controlar la dirección y (step) para elegir los pasos del motor que serán dirigidas al driver del motor.
- Deberá controlar la velocidad del motor.
- Nuestro sistema dispondrá de un sensor hall que hará la función de final de carrera. Esta señal será registrada por nuestro microcontrolador.
- Puerto UART de comunicaciones serial para comunicarse con el estimulador eléctrico y recibir tramas que contengan acciones de nuestro estimulador.

Buscando entre las diferentes opciones que nos ofrecía el mercado, se optó por elegir como microcontrolador un Arduino Uno (Figura 19), el cual describiremos en el anexo II y comprobaremos que cumple nuestras especificaciones.

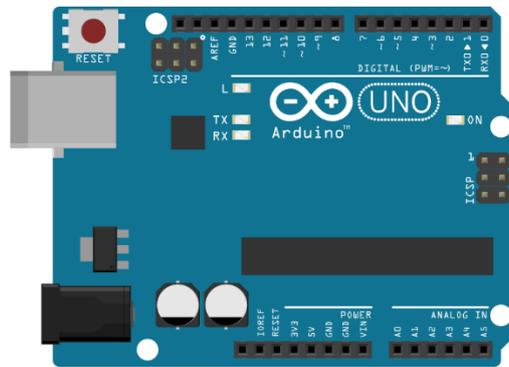


Figura 19. Arduino UNO [32].

5.1.3.2 Driver motor

Con el fin de controlar el motor paso a paso ha sido necesario la incorporación de un driver motor. Como solución, se decidió utilizar el EasyDriver de la casa SparkFun (Figura 20).

El EasyDriver es un controlador de motor paso a paso simple de utilizar. Requiere un suministro de 7V a 30V y hace funcionar el motor a través de dos señales esenciales, (step) para controlar los pasos del motor y (dir) para controlar la dirección del motor. Por otro lado, a través de un potenciómetro de corriente podremos ajustar la máxima corriente que le suministraremos al motor. Además, los pines de microstep (MS1 y MS2) permiten ajustar la resolución dependiendo de nuestras necesidades.



Figura 20. Drive motor EasyDriver [33].

En nuestro caso el driver irá alimentado con una fuente de 12 V y 1.3 A ya que nuestro motor requiere esta tensión. Esta fuente también nos permitirá alimentar el Arduino de nuestro sistema.

5.1.3.3 Motor paso a paso

Nuestro sistema requerirá un motor paso a paso que será el encargado de hacer girar la varilla roscada unida a su eje y provocar el estímulo mecánico. Se ha utilizado el motor SM-42BYG011-25 (Figura 21) de la casa SparkFun Electronics. Se ha elegido este motor por ser un motor bipolar con cuatro cables de salida y un tamaño de marco NEMA 17. El driver y el motor se eligieron de forma conjunta ya que se obtuvieron del mismo fabricante.



Figura 21. Motor paso a paso [34].

Este motor requiere una tensión de 12 V que será suministrada por la misma fuente comentada anteriormente. Podremos observar las características del motor en el anexo II.

5.1.3.4 Sensor Hall

Para controlar que la pieza móvil está situada en buena posición para colocar los pocillos de goma, se ha utilizado un sensor de hall que hará la función de final de carrera. Este sensor permite conocer el punto de referencia y a partir de él, el Arduino sitúa el carro en el lugar adecuado. El sensor se activará/desactivará mediante un imán enfrentado a él y esta información será registrada a través de un pin digital de nuestro Arduino. En nuestro caso hemos elegido el sensor hall A3144 (Figura 22).

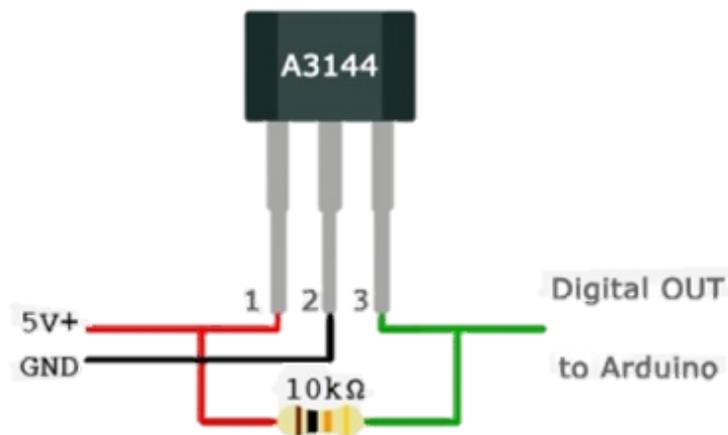


Figura 22. Sensor Hall A3144 [35].

Se ha elegido este sensor ya que tiene un tamaño reducido adecuado para nuestro sistema, no tiene el desgaste mecánico que tendría un interruptor mecánico y evita que se puede activar accidentalmente.

5.1.3.5 Código del firmware del Arduino

A continuación describiremos el código del firmware que controla el comportamiento del Arduino de nuestro estimulador mecánico.

Lo primero que ejecuta el Arduino es la función “setup()”, que únicamente se ejecuta en el arranque y realiza la inicialización de las variables y la configuración de los periféricos. En nuestro caso realiza las siguientes tareas:

- Inicialización de la UART a 9600 baudios.
- Configuración de los pines step, dir y sensorHall como entrada o salida.

- Inicialización de un nuevo puerto serie virtual (Software Serial) para dejar libre la UART que nos permitirá actualizar el programa cuando deseemos.

A continuación se realizaron las funciones startt() y reset(). Estas funciones permiten el manejo del motor en distintas situaciones.

La función startt() calcula la distancia en cm según el porcentaje de estiramiento que le indiquemos, y a partir de esta distancia se calculan los pasos del motor necesarios. Estos pasos serán introducidos en la función step() que será la encargada de realizar el movimiento del motor.

Por otro lado, la función reset(), leerá el valor digital del sensor Hall lo cual nos permitirá retornar la pieza móvil a su lugar de origen para colocar de nuevo los pocillos de goma.

Por último, se ejecutará permanentemente la función loop(). Esta función escuchara el puerto serie virtual declarado anteriormente. Dependiendo de la trama que reciba ejecutará una función u otra. Concretamente podrá recibir las siguientes tramas:

- RESET: Ejecutará la función reset() comentada anteriormente.
- START: Activará la función start(), mediante una interrupción con la instrucción attachInterrupt().
- STOP: Desactivará la interrupción mediante la instrucción detachInterrupt().

El código del firmware del Arduino se puede encontrar en el Anexo I.

5.2 Estimulador eléctrico

5.2.1 Diagrama de bloques

Uno de los pasos fundamentales para inducir un grado similar de diferenciación de los cardiomiocitos es la introducción de pulsos eléctricos. En la Figura 23 podemos observar los diferentes componentes del estimulador eléctrico y como están conectados entre ellos. Se procedió a implementar un estimulador de cuatro canales que irá dirigido por un PC y comunicado mediante un XBee de forma inalámbrica o a través de la pantalla táctil incorporada al sistema.

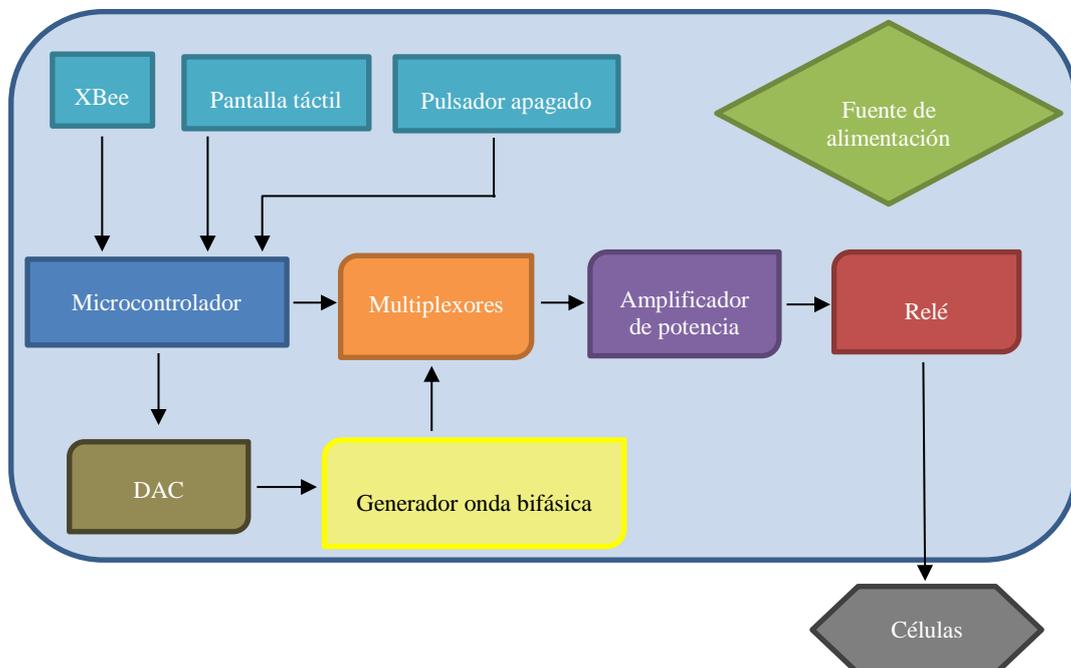


Figura 23. Diagrama de bloques del estimulador eléctrico.

La interfaz de usuario mediante pantalla táctil y la comunicación inalámbrica nos permitirán cambiar todos los parámetros de nuestro estímulo, es decir; ancho, frecuencia y tensión de los pulsos bifásicos. Además, el pulsador de apagado implementado en la caja ofrece la posibilidad de detener o iniciar la estimulación en casos de emergencia o por sí deseamos un control manual del inicio de la estimulación. También cuenta con un indicador LED por cada canal situado en el panel frontal del aparato, que indicará a qué instante se realiza el estímulo y qué canal tenemos activado (Figura 24).

Este equipo posee una salida de pulsos de estimulación con una tensión máxima de 40V y una corriente máxima de funcionamiento de 1A, más que suficiente para nuestras necesidades.

Todos los componentes del sistema han sido montados dentro de una caja de plástico ABS especialmente diseñada para instrumentación de laboratorio. El estimulador se alimenta a través de la red de 230V, posee un fusible de protección y un interruptor para encender/apagar todo el sistema.



Figura 24. Panel frontal del sistema.

5.2.2 Esquema eléctrico

El sistema dispondrá de los componentes mostrados anteriormente, donde cada uno de ellos realizará su función para obtener la señal que deseamos. En la Figura 25 podemos observar los resultados de las distintas etapas del sistema.

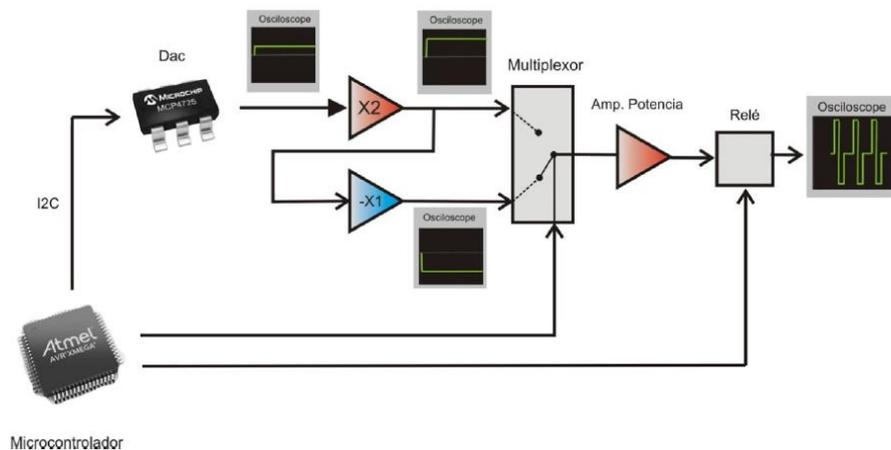


Figura 25. Esquema de la generación de la señal bifásica de la estimulación.

El usuario enviará las órdenes que serán recibidas y ejecutadas por el microcontrolador. Una vez el microcontrolador interpreta las tramas, se establece la tensión de la señal de referencia gracias a los DACs e inmediatamente el microcontrolador envía las señales de conmutación al multiplexor para generar la forma de onda de la estimulación con la morfología deseada por el usuario.

Para permitir la manipulación de la amplitud de la señal de estimulación, se requiere un sistema capaz de convertir una instrucción digital generada por el microcontrolador en una señal analógica. Esta tensión variable de 0 a 5 V generada por los DACs, servirá de referencia para la generación de la señal de estimulación. A continuación, la tensión será amplificada en la siguiente etapa para generar la señal bipolar. Al ser la señal bipolar, el dispositivo deberá generar dos señales diferentes, una señal amplificada directamente proporcional y otra inversamente proporcional. Para ello, se requiere de una etapa de amplificación con un amplificador no inversor de ganancia 2 y otro inversor de ganancia -1 en cascada, implementados con dos amplificadores operacionales, logrando así una señal positiva y negativa entre -10 y +10 V.

Una vez generada la forma de onda deseada, ésta es amplificada tanto en tensión como en corriente para alcanzar los valores requeridos para la experimentación.

Por último, para que los electrodos y todo el sistema estén aislados del tejido a estimular, se ha implementado un relé entre la salida de la etapa de amplificación de potencia y los conectores de los electrodos, lo que permitió aislar eléctricamente los electrodos del sistema cuando no se está estimulando.

Para el montaje del estimulador se utilizaron cuatro PCBs, una por cada canal, ya diseñadas y fabricadas en el laboratorio (Figura 26). Estas PCBs albergan todos los componentes descritos anteriormente que fueron soldados en ella. Por último, se testearon con un osciloscopio comprobando cada una de las etapas. En el anexo II podemos encontrar el esquema de conexiones de las PCBs.



Figura 26. Fotografía PCB utilizada en estimulador.

5.2.3 Control electrónico

Para lograr la generación de los pulsos bifásicos en el estimulador, se ha utilizado un Arduino MEGA 2560 (Figura 27) debido a que posee el número necesario de E/S digitales que requiere este sistema.

La generación de la señal de estimulación comienza en el momento en el que es recibida la orden por parte del usuario. Estas órdenes son recibidas a través de la pantalla táctil o mediante el canal inalámbrico habilitado por el XBee que incluyen las tramas de la frecuencia, ancho de pulso y tensión de estimulación de cada canal.

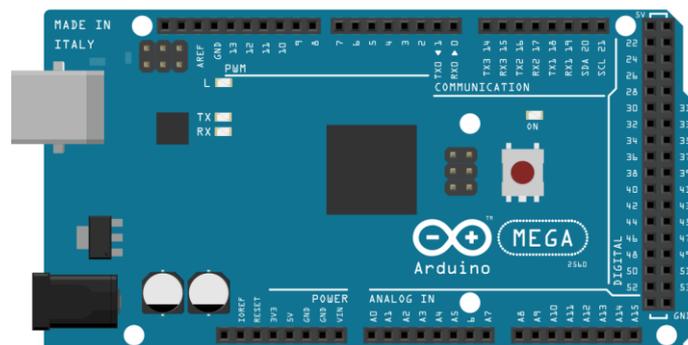


Figura 27. Arduino MEGA 2560 [36].

El código del firmware de este Arduino, ya programado anteriormente en el laboratorio, se ha tenido que modificar para coordinar la estimulación eléctrica con la mecánica y para adaptar la pantalla táctil.

5.2.3.1 Comunicación inalámbrica

La comunicación inalámbrica se ha implementado en base al protocolo *Zigbee*. Este protocolo es normalmente utilizado en necesidades industriales, científicas y médicas. Sus características principales son su sencillez de uso, bajo coste y bajo consumo eléctrico. Por el contrario, su velocidad es mucho menor que otros sistemas de comunicación pero para las necesidades de este trabajo son más que suficientes.

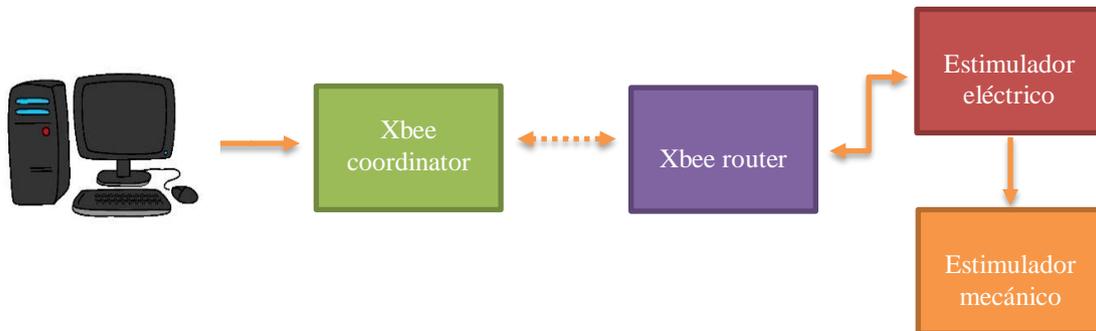


Figura 28. Diagrama de funcionamiento de la comunicación inalámbrica del sistema.

Un módulo XBee es un dispositivo electrónico de comunicación inalámbrica basado en protocolo Zigbee. El PC se conecta con el XBee maestro a través de un cable USB para crear un puerto serie virtual entre el ordenador y el XBee, mientras que el XBee utilizado por el estimulador utilizará una conexión tipo UART con su respectivo microcontrolador. Esto permitirá al usuario establecer una comunicación directa entre el ordenador y el dispositivo, librando el laboratorio de cableado.

Cada módulo XBee deberá ser programado para realizar su función dentro de la red de comunicación. Para ello, cada XBee poseerá una configuración diferente, maestro y esclavo. Esta programación de los módulos puede ser realizada de dos modos, a través de la consola del puerto serie y utilizando comandos AT, o bien utilizando un programa llamado X-CTU que permite programar cada XBee de manera más intuitiva. En el presente trabajo se ha utilizado la segunda opción por su mayor sencillez.

Todos estos componentes se instalaron junto con las fuentes de alimentación en una caja de plástico ABS diseñada para instrumentación de laboratorio (Figura 29).

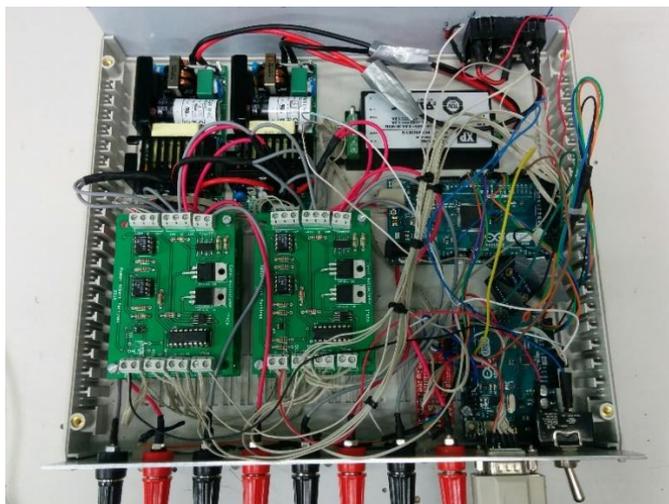


Figura 29. Fotografía durante el montaje del estimulador electromecánico de 4 canales.

Una vez montado el conjunto, se testearon durante varias horas las salidas de cada canal y el funcionamiento de la estimulación mecánica coordinada, donde fue observado el correcto funcionamiento del sistema (Figura 30).

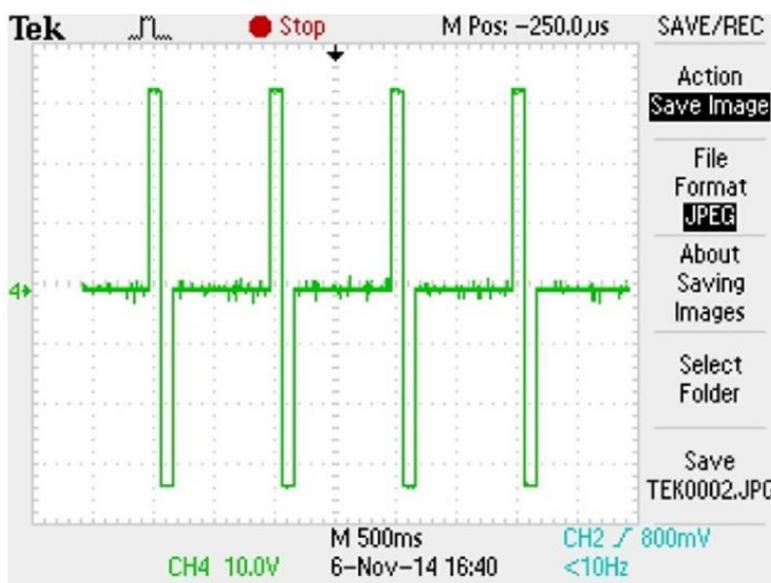


Figura 30. Señal de estimulación de 32 V con un periodo de 1000ms y un ancho de 200ms.

5.3 Interfaz de usuario

5.3.1 Hardware

Para poder configurar los parámetros de forma fácil e intuitiva se decidió incorporar al sistema una interfaz de usuario mediante pantalla táctil. Después de una búsqueda exhaustiva de las opciones que había en el mercado elegimos la pantalla Arduino TFT Touch Shield V2.0 (Figura 31). Esta pantalla puede ser utilizada con los modelos Arduino UNO y MEGA. En nuestro caso, utilizamos un Arduino UNO ya que cumplía con nuestras necesidades.



Figura 31. Vistas de la pantalla Arduino TFT Touch Shield V2.0 [37].

Esta pantalla irá pinchada al Arduino UNO, en el cual configuraremos cada uno de los pines para el buen funcionamiento de la pantalla. Además, incorporamos las librerías necesarias que descargamos de la web del fabricante. El código del firmware del Arduino se puede encontrar en el Anexo I.

Por último, hemos diseñado una caja impresa en 3D para incorporar la pantalla al sistema (Figura 32). Como se puede observar se ha dejado el espacio necesario para poder conectar el cable USB del puerto serie y así poder recargar el programa de manera sencilla.

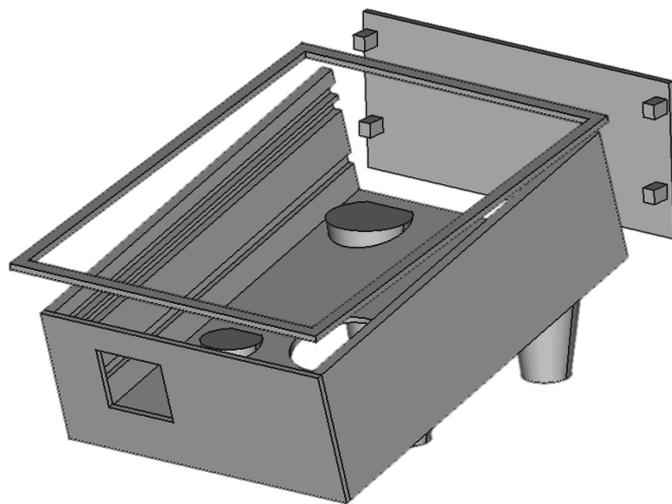


Figura 32. Caja soporte para la pantalla táctil.

5.3.2 Descripción de la interfaz

La interfaz de usuario se ha programado de manera que se puedan cambiar los parámetros del sistema de forma fácil e intuitiva. Para ello se ha desarrollado una serie de menús que irán apareciendo de forma secuencial y que nos permitirá elegir el canal que deseamos estimular, los parámetros que deseamos modificar y el valor que le queremos asignar a estos parámetros.

En primer lugar, se diseñó una pantalla de bienvenida al estimulador electromecánico donde presionando el botón START cambiaremos de menú para empezar nuestra estimulación (Figura 33).

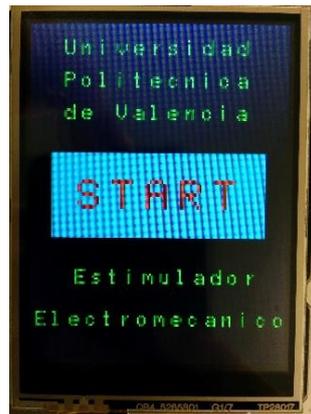


Figura 33. Pantalla de inicio.

En segundo lugar, se programó el firmware para controlar una pantalla con cinco botones, de los cuales cuatro son para elegir que canal deseamos estimular, y uno es el botón de STOP para detener la estimulación eléctrica (Figura 34). Este último botón asigna un 0 a todos los parámetros (tensión, periodo y ancho de pulso) y así, se detiene la estimulación de cada canal independientemente.



Figura 34. Pantalla de elección de canal de estimulación y detención de cualquier canal.

A continuación, se diseñó un menú para elegir el parámetro al cual deseamos asignar un valor (Figura 35). Los parámetros de la estimulación eléctrica que se pueden controlar son tensión, periodo y ancho de pulso.

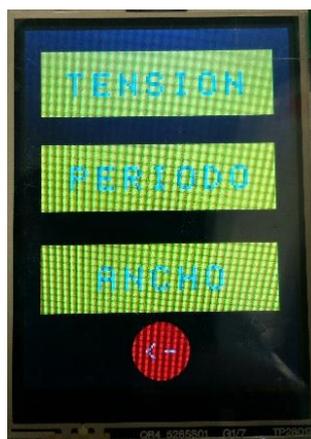


Figura 35. Pantalla de elección del parámetro que queremos modificar.

Una vez hemos elegido un parámetro, nos redirige a un teclado numérico por pantalla donde podremos asignar el valor que deseemos (Figura 36). Asignado dicho valor, vuelve a la misma pantalla hasta que se han asignado los tres parámetros.



Figura 36. Pantalla de asignación del valor al parámetro deseado.

Asignados los tres parámetros, salta al siguiente menú que nos permitirá modificar estos valores de forma progresiva con botones +/- (Figura 37). Los valores de tensión variarán de 1 V en 1 V, el ancho de pulso variará de 1 ms en 1 ms y el periodo variará en intervalos de 50 ms o 500 ms si presionamos débilmente o fuertemente, respectivamente, ya que la pantalla es sensible a la presión aplicada. Además, podremos acceder al menú del estimulador mecánico.



Figura 37. Pantalla para controlar el valor progresivamente y acceso al menú de estimulación mecánica.

Por último, podremos acceder al menú de estimulación mecánica donde podremos iniciar, parar o resetear la pieza móvil, es decir, el carro de desplazamiento (Figura 38).



Figura 38. Pantalla de control del estimulador mecánico.

Como se puede apreciar en la Figura 39, todo el sistema se ha montado de forma conjunta para su uso en un laboratorio de una forma fácil e intuitiva para el usuario final.

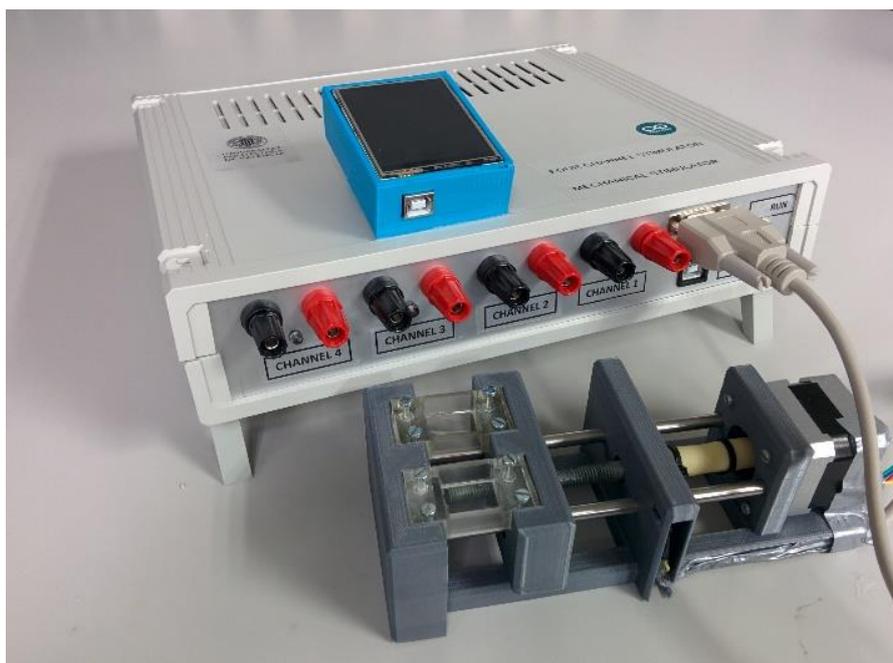


Figura 39. Sistema completo de estimulación electromecánica

Capítulo 6. Resultados

En esta sección se estudiarán y analizarán distintas pruebas que se han realizado con nuestro estimulador para comprobar que cumple con las condiciones de diseño estipuladas.

6.1 Estiramiento del pocillo

Durante el proceso de prueba del sistema hemos podido comprobar que el motor paso a paso soporta la tensión que ejerce el pocillo de goma en sentido contrario. Se han probado diferentes porcentajes de estiramiento comunes para el cultivo del tejido siendo estos porcentajes entre 5% y 10% (Figura 40).

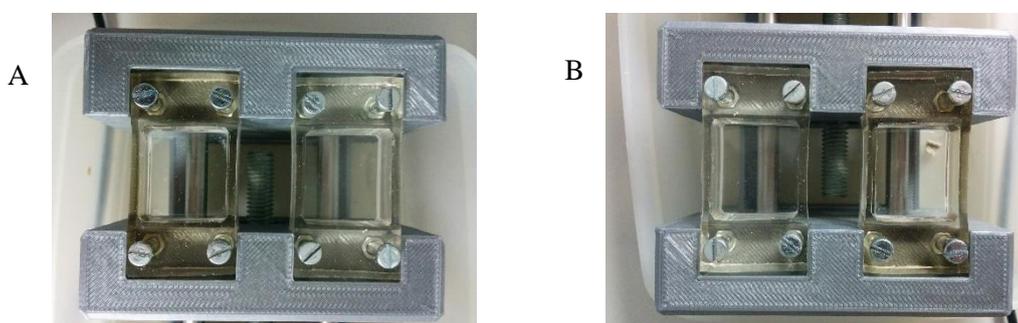


Figura 40. (A) Pocillo estirado 5% (B) Pocillo estirado 10%

Por otro lado, se ha comprobado el número de pasos que da el motor para los diferentes porcentajes de estiramiento. Como era de esperar, obtenemos una gráfica lineal ya que el porcentaje de estiramiento es directamente proporcional al número de pasos del motor, tal y como se muestra en la Figura 41.

La distancia que precisábamos mover la pieza móvil se ha calculado a partir del porcentaje de estiramiento que deseábamos y el número de pasos del motor se ha calculado a partir de esta distancia, sabiendo que cada vez que el motor da una vuelta la pieza se mueve 1,25 mm. Al estar la tuerca fija, el carro se desplaza longitudinalmente por la varilla provocando el estímulo mecánico.

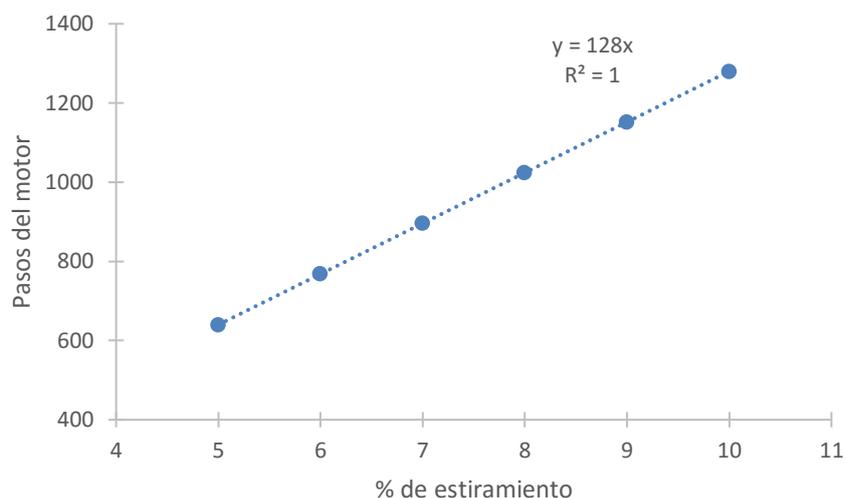


Figura 41. Número de pasos del motor en función del porcentaje de estiramiento del pocillo

Por último, se ha realizado un estudio de los límites que tiene nuestro motor a la hora de realizar un estímulo. Asignando un determinado porcentaje de estiramiento y una determinada frecuencia se ha comprobado si el motor tiene la suficiente capacidad para que no se pierda el sincronismo con la estimulación eléctrica, es decir, que la pieza móvil llegue al final de su recorrido antes de empezar la siguiente estimulación.

La frecuencia cardíaca en un ser humano adulto suele estar entre 60-100 pulsaciones por minuto. Esto equivale a una frecuencia entre 1-1,66 Hz.

En nuestro estudio se han considerado cuatro periodos diferentes en un rango desde los 700ms hasta los 1000ms. Se ha podido comprobar que hasta un 7% de estiramiento el motor cumple todos los periodos propuestos. A partir de 8% el motor ya no tiene suficiente velocidad para coordinar los dos estímulos. A continuación se detallarán en la Tabla 1 los límites de nuestro motor.

	5%	6%	7%	8%	9%	10%
700ms						
800ms						
900ms						
1000ms						

Tabla 1. Tabla gráfica de los límites de funcionamiento del motor según las condiciones de porcentaje de estiramiento y periodo. En verde, las condiciones en las que el motor funciona favorablemente y en rojo, las que funciona desfavorablemente.

6.2 Estimulación electromecánica coordinada

En primer lugar, se ha comprobado que el estimulador eléctrico funcionase correctamente. Se han realizado pruebas con cada uno de los canales cambiando los tres parámetros del estímulo. Además, se han repetido dichas pruebas con varios canales activados comprobando que son canales independientes y que se pueden emplear simultáneamente. Estos canales se pueden emplear para otra aplicaciones, en nuestro caso, para la regeneración de tejido cardíaco utilizaremos solamente un canal.

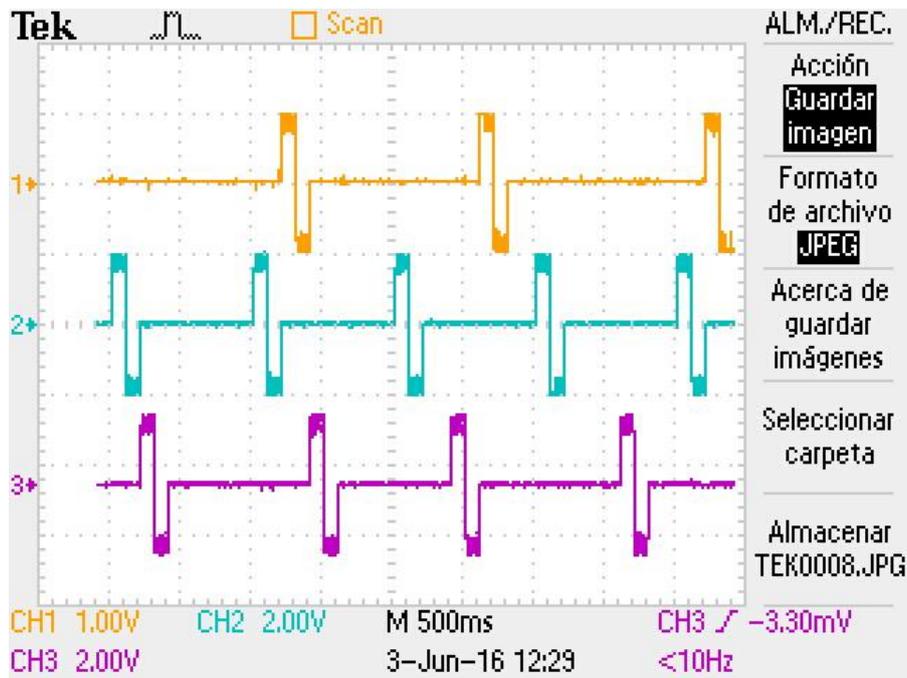


Figura 42. Pulsos de estimulación eléctrica a diferentes tensiones y periodos.

Uno de los objetivos de nuestro proyecto era coordinar la estimulación eléctrica con la mecánica para simular de forma equivalente a la de un corazón. Para llevar a cabo esta tarea se ha modificado el código del estimulador eléctrico y se ha utilizado un pin de cada Arduino para enviar una señal de interrupción que se activará y desactivará al mismo instante que los Relés de la estimulación eléctrica.

En nuestro caso los pines del Arduino UNO que admiten interrupción son el 2 y 3, y la instrucción que habilita/deshabilita esta interrupción es `attachInterrupt()/detachInterrupt()`. A continuación, en la Figura 43 se mostrará un gráfico con las formas de onda de ambas estimulaciones y su sincronismo.

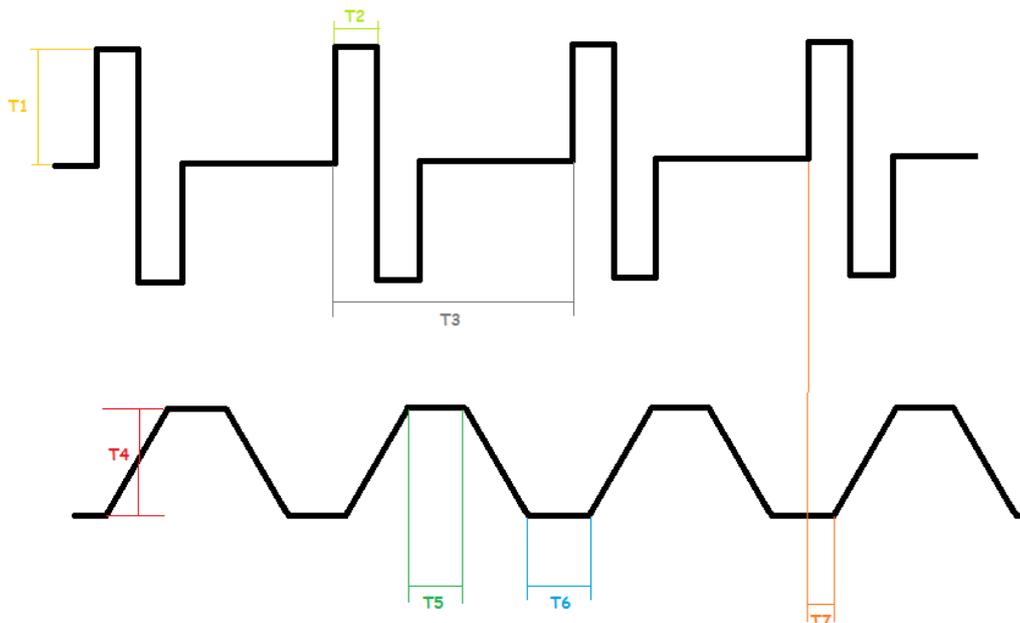


Figura 43. Gráfico de la forma de onda de la estimulación eléctrica y mecánica. T1: Tensión, T2: Ancho de pulso, T3: Periodo, T4: Porcentaje estiramiento y contracción, T5: Tiempo de espera con el pocillo estirado, T6: Tiempo de reposo entre estímulos, T7: Retardo entre la estimulación eléctrica y mecánica a causa de la electrónica.

Capítulo 7. Pliego de condiciones

7.1 Características del equipo

- Voltaje de entrada: 230 V.
- Voltaje de salida de la estimulación eléctrica: -40 V a 40 V.
- Corriente de salida máxima: 1 A.
- Rango de frecuencia: Para el 5 % de estiramiento 1,42 Hz y para el 10 % de estiramiento 1,11 Hz.
- Rango de estiramiento 0-10 % de estiramiento.
- Capacidad para dos pocillos de goma. Dimensiones: 42 cm largo x 23,5 cm ancho x 10 cm alto. Membrana donde residen las células: 20 cm largo x 20 cm ancho.
- Comunicación a través de una interfaz de usuario mediante pantalla táctil o mediante una red ZigBee.

7.2 Materiales y presupuesto

<i>Cantidad</i>	<i>Referencia</i>	<i>Nombre</i>	<i>Precio Unitario</i>	<i>Total</i>	<i>Proveedor</i>
1	1050-1018-ND	Arduino Mega 2560	47,48 €	47,48 €	Digikey
2	1050-1041-ND	Arduino UNO	20,31 €	40,62 €	Digikey
2	102-3355-ND	AC/DC CONVERTER 48V 70W AC/DC CONVERTER +/-12V	25,22 €	50,44 €	Digikey
1	1470-1201-ND	30W	58,31 €	58,31 €	Digikey
4	296-22997-5-ND	OPA 454 Amplificador operacional	6,25 €	25,00 €	Digikey
4	ADG408BNZ-ND	IC MULTIPLEXER 8x1 16DIP BOX ABS GRAY 10.19"L X 11.45"W	5,35 €	21,40 €	Digikey
1	902-1084-ND	11.45"W	45,70 €	45,70 €	Digikey
1	909GPE-ND	Conector DB9 grey plastic	0,64 €	0,64 €	Digikey
1	ND MCP4725A0T-E/CHCT-	DAC 12 bits	0,85 €	0,85 €	Digikey
1	ND MCP4725A1T-E/CHCT-	DAC 12 bits	0,85 €	0,85 €	Digikey
1	ND MCP4725A2T-E/CHCT-	DAC 12 bits	0,85 €	0,85 €	Digikey
1	ND MCP4725A3T-E/CHCT-	DAC 12 bits	0,85 €	0,85 €	Digikey
4		PCB Estimulador	7 €	28,00 €	
1	1568-1105-ND	Stepper Motor With Cable	13,61 €	13,61 €	Digikey
1	1568-1108-ND	EasyDriver Stepper Motor Drive	13,61 €	13,61 €	Digikey
1	602-1098-ND	Module Zigbee W/Wired Ant	16,03 €	16,03 €	Digikey
1	32400-ND	Xbee USB adapter board	26,56 €	26,56 €	Digikey
1	1568-1075-ND	Board Xbee explorer regulated	9,38 €	9,38 €	Digikey
1	ARD-0058	Arduino TFT Touch Shield V2.0	43,00 €	43,00 €	Bricogeek

1	A3144	Sensor HALL	1,29 €	1,29 €	Digikey
2		287 Rodamientos Lineales	1,21 €	2,42 €	Thingibox
1		609 Varilla roscada 1m	1,24 €	1,24 €	Thingibox
1		664 Barra lisa calibrada 1m	5 €	5,28 €	Thingibox
			TOTAL	453,41 €	

Tabla 2. Presupuesto del proyecto.

A este presupuesto debemos añadirle la dedicación en horas para realizar el proyecto. Según el centro de apoyo a la innovación, la investigación y la transferencia de tecnología se establece que un ayudante de universidad tiene un coste de 23,70 €/hora.

Ayudante de universidad	300 horas	7110 €
-------------------------	-----------	--------

El precio final de este proyecto ascendería a 7563,41 €.

Capítulo 8. Conclusiones y propuesta de trabajo futuro

En el presente trabajo se ha logrado el diseño e implementación de un estimulador electromecánico coordinado para el desarrollo de tejido cardíaco artificial. Se diseñó e implementó un sistema capaz de provocar un estímulo mecánico mediante estiramiento longitudinal y coordinarlo con el pulso bifásico eléctrico. Además, se finalizaron e introdujeron mejoras en los proyectos ya iniciados en el laboratorio que fueron incorporados en un único sistema con el fin de ser controlados bajo un único entorno y comunicados entre sí a través de los puertos serie de los microcontroladores. Para poder llevar a cabo el control del sistema de forma intuitiva, se programó una interfaz de usuario controlado mediante una pantalla táctil, aunque también se puede controlar el sistema mediante una red inalámbrica Zigbee. Una de las ventajas de este sistema es que es completamente programable y, por tanto, podemos ajustar todos los parámetros de la estimulación, cosa que no podríamos hacer en otros sistemas comerciales. Finalmente este equipo fue instalado y validado en el laboratorio de electrofisiología del Hospital Gregorio Marañón.

Se han pensado diversas mejoras y actualizaciones que podrían implantarse en una versión futura de nuestro estimulador:

- La realización de un estimulador más grande para abarcar un mínimo de cuatro pocillos de goma para poder realizar más pruebas simultáneamente.
- Monitorizar en todo momento la posición de la pieza móvil del estimulador mecánico para poder moverla electrónicamente a la posición que deseemos desde el interfaz de usuario.
- Programar protocolos de estimulación para poder cambiar los parámetros del estímulo durante el tiempo del experimento.

Capítulo 9. Bibliografía

- [1] Dr. G. F. Falke, Dr. A. Atala, "Reconstrucción de tejidos y órganos utilizando ingeniería tisular" *Arch..argent..pediatr.*, vol. 98, no. 2, pp. 103-115, 2000.
- [2] C. T. Gomillion, K. J. L. Burg "Stem cells and adipose tissue engineering" *Biomaterials*, vol. 27, pp. 6052-6063, 2006.
- [3] A. Bayes-Genis et al. "Human progenitor cells derived from cardiac adipose tissue ameliorate myocardial infarction in rodents" *J Mol. Cell Cardiol.*, vol. 49, pp. 771 -780, 2010.
- [4] J. H. Houtgraaf et al. "First experience in humans using adipose tissue -derived regenerative cells in the treatment of patients with ST-segment elevation myocardial infarction" *J. Am. Coll. Cardiol.*, vol. 59, no. 5, pp. 539-540, 2012.
- [5] K. Le Blanc et al. "HLA expression and immunologic properties of differentiated and undifferentiated mesenchymal stem cells" *Exp. Hematol.*, vol. 31, no. 16, pp. 890 -896, 2003.
- [6] G. Vunjak-Novakovic et al. "Challenges in cardiac tissue engineering" *Tissue Eng. Part B Rev.*, vol. 16, no.2, pp. 169-187, 2010.
- [7] M. Perez-Illarbe et al. "Characterization of the paracrine effects of human skeletal myoblasts transplanted in infarcted myocardium" *Eur. J. Heart Fail.*, vol. 10, no. 11, pp. 1065-1072, 2008.
- [8] B. Pelacho et al. "Multipotent adult progenitor cell transplantation increases vascularity and improves left ventricular function after myocardial infarction" *J. Tissue Eng. Regen. Med.*, vol. 1, no. 1, pp. 51-59, 2007.
- [9] X. L. Aranguren et al. "Multipotent adult progenitor cells sustain function of ischemic limbs in mice" *J. Clin. Invest.*, vol. 118, no. 2, pp. 505-514, 2008.
- [10] R. Uemura et al. "Bone marrow stem cells prevent left ventricular remodeling of ischemic heart through paracrine signalling" *Circ. Res.*, vol. 98, no. 11, pp. 1414-1421, 2006.
- [11] D. Hou et al. "Radiolabeled cell distribution after intramyocardial, intracoronary, and interstitial retrograde coronary venous delivery: implications for current clinical trials" *Circulation*, vol. 112, no. 9, pp. I150-I156, 2005.
- [12] T. Eschenhagen et al. "Three-dimensional reconstitution of embryonic cardiomyocytes in a collagen matrix: a new heart muscle model system" *FASEB J.*, vol. 11, no. 8, pp. 683-694, 1997.
- [13] W. H. Zimmermann et al. "Tissue engineering of a differentiated cardiac muscle construct" *Circ. Res.*, vol. 90, no. 2, pp. 223-230, 2002.
- [14] P. A. McKee, W. P. Castelli, W. B. McNamara "The natural history of congestive heart failure" *N. Engl. J. Med.*, vol. 285, no. 26, pp. 1441-1446, 1971.

- [15] M. R. Cowie et al. "Incidence and aetiology of heart failure: a population-based study" *Eur. Heart J.*, vol. 20, pp. 421-428, 1999.
- [16] G. Olivetti et al. "Cellular basis of chronic ventricular remodelling after myocardial infarction in rats" *Circ. Res.*, vol. 68, no. 3, pp. 856-869, 1991.
- [17] S. Pernagallo, J. J. Diaz-Mochon, M. Bradley "A cooperative polymer-DNA microarray approach to biomaterial investigation" *Lab. Chip.*, vol. 9, no. 3, pp. 397-403, 2009.
- [18] C. Gálvez-Montón et al. "Ingeniería tisular cardiaca y corazón bioartificial" *Rev. Esp. Cardiol.*, vol. 66, no. 5, pp. 391-399, 2013.
- [19] G. Vunjak-Novakovic et al. "Challenges in cardiac tissue engineering" *Tissue Eng. Part B Rev.*, vol. 16, no. 2, pp. 169-187, 2010.
- [20] M. Radisc et al. "Functional assembly of engineered myocardium by electrical stimulation of cardiac myocytes cultured on scaffolds" *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, no. 52, pp. 18129-18134, 2004.
- [21] M. Radisc et al. "Cardiac tissue engineering using perfusion bioreactor systems" *Nat. Protoc.*, vol. 3, no. 4, pp. 719-738, 2008.
- [22] M. Jain et al. "Cell therapy attenuates deleterious ventricular remodeling and improves cardiac performance after myocardial infarction" *Circulation*, vol. 103, no. 14, pp. 1920-1927, 2001.
- [23] T. Siminiak et al. "Autologous skeletal myoblast transplantation for the treatment of, postinfarction myocardial injury: phase I clinical study with 12 month of follow-up" *Am. Heart J.*, vol. 148, no. 3, pp. 531-537, 2004.
- [24] P. Menasché et al. "The Myoblast Autologous Grafting in Ischemic Cardiomyopathy (MAGIC) trial: first randomized placebo-controlled study of myoblast transplantation" *Circulation*, vol. 117, no. 9, pp. 1189-1200, 2008.
- [25] H. C. Ott et al. "Perfusion-decellularized matrix: using nature's platform to engineer a bioartificial heart" *Nature medicine*, vol. 14, no. 2, pp. 213-221, 2008.
- [26] P. L. Sánchez et al. "Acellular human heart matrix: A critical step toward whole heart grafts" *Biomaterials*, vol. 61, pp. 279-289, 2015.
- [27] J. M. Wainwright et al. "Preparation of cardiac extracellular matrix from an intact porcine heart" *Tissue Eng. Part C Methods*, vol. 16, no. 3, pp. 525-532, 2010.
- [28] Flexcell International Corporation "Flexcell FX-5000 Tension System" <http://www.flexcellint.com/FX5000T.htm> [online].
- [29] IonOptix "Myopacer Cell Stimulator" <http://www.ionoptix.com/product/myopacer-cell-stimulator/> [online].
- [30] R. Izquierdo-Albero et al. "Effect of *in vitro* mechanical stimulation on cardiac gene expression" *Libro de actas del XXXIII Congreso Anual de la Sociedad Española de Ingeniería Biomédica, Madrid, 4, 5 y 6 de Noviembre de 2015*, <http://seib.org.es/publicaciones-cientificas-caseib/> [online].
- [31] MakerBot Industries "Replicator 2X" <http://store.makerbot.com/replicator2x> [online].
- [32] Arduino "Arduino Board Uno" <http://arduino.cc/en/Main/ArduinoBoardUno> [online].
- [33] SparkFun Electronics "EasyDriver-Stepper Motor Driver" <https://www.sparkfun.com/products/12779> [online].
- [34] SparkFun Electronics "Stepper Motor with cable" <https://www.sparkfun.com/products/9238> [online].

- [35] Allegro Microsystems “A3144 Sensor Hall”
http://www.alldatasheet.com/view.jsp?Searchword=A3144&gclid=Cj0KEQjws_m6BRCv37WbtNmJs-IBEiQAWKKt0CApO-0q73wxyx_nWwVhJoAF73SQ2QnOIIA1OHPE1IEaAiWk8P8HAQ [online].
- [36] Arduino “Arduino Board Mega 2560”
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560> [online].
- [37] SeedStudio “2.8” TFT Touch Shield V2.0”
https://www.seeedstudio.com/item_detail.html?p_id=1286 [online].

Capítulo 10. Anexos

10.1 Anexo I: Código del firmware del Arduino

10.1.1 Estimulador Mecánico

```
Estimulador_Mecanico
#include <SoftwareSerial.h>

//#define ACTIVACION_ESTIMECANICO 8 //Pin entrada ESTIMULADOR ELECTRICO
#define sensorHallPin 5
//variables pines y banderas
int contador_pasos=18000;
int posicion_inicial=18000;
int dirPin = 10;
int stepperPin = 11;
boolean valueSensorHall=LOW;
unsigned long time=millis();
boolean valueActivacion=false;
volatile byte valueReset= LOW;
boolean stringComplete = false;
String comando= "";

//VARIABLES PARAMETRIZABLES
double porcentaje_estiramiento=6; //de estiramiento
int veloc_estirar=160; //velocidad estirar pocillo + mas lento tope 160
int veloc_encoger=160; //velocidad encoger pocillo - mas rapido tope 160
int tiempo_pocillo_estirado=0; //pausa entre estiramiento y contracción. Habilitar la linea de código si se desea
int pasos_motor=0; //no tocar
double distancia_cm=0; //no tocar

SoftwareSerial mySerial(8,9); //RX, TX

void setup() {
  pinMode(dirPin, OUTPUT);
  pinMode(stepperPin, OUTPUT);
  pinMode(sensorHallPin, INPUT);
  //attachInterrupt(1, startt, RISING);
  Serial.begin(9600);
  while(!Serial){
    ;
  }
  mySerial.begin(9600);
}

void step(boolean dir,double steps){
  digitalWrite(dirPin,dir);
  delay(50);
}
```

```

for(double i=0;i<steps;i++){
  digitalWrite (stepperPin, HIGH);
  if(dir==LOW){ delayMicroseconds (veloc_encoger);}else{delayMicroseconds (veloc_estirar);}
  digitalWrite (stepperPin, LOW);
  if(dir==LOW){ delayMicroseconds (veloc_encoger);}else{delayMicroseconds (veloc_estirar);}
}
}
void reset(){

  valueSensorHall=digitalRead(sensorHallPin);
  // Serial.println(valueSensorHall);
  while(valueSensorHall==HIGH){
    valueSensorHall=digitalRead(sensorHallPin);
    digitalWrite (dirPin,HIGH);
    digitalWrite (stepperPin, HIGH);
    delayMicroseconds (veloc_encoger);
    digitalWrite (stepperPin, LOW);
    delayMicroseconds (veloc_encoger);
  }
  step(LOW,18000);//18300
}
void startt(){
  distancia_cm=(20*porcentaje_estiramiento)/100;
  pasos_motor=(800*distancia_cm)/1.25;
  step(true,pasos_motor); //800 pasos son 1,25 mm
  step(false,pasos_motor);
}
void loop(){

  valueSensorHall=digitalRead(sensorHallPin);
  //Serial.println(sensorHallPin);
  while(mySerial.available()) {
    char inChar = (char)mySerial.read(); //leo un caracter
    if (inChar == '\n') { //si es el caracter de final marco que he terminado de leer
      stringComplete = true;
    }
    else {comando += inChar;}

  }

  if(stringComplete==true){
    if (comando.substring(0,5)=="RESET"){
      reset();comando="";
      stringComplete = false;
    }
    if (comando.substring(0,5)=="START"){
      attachInterrupt(1,startt,RISING);comando="";
      stringComplete = false;
      Serial.println("STARTEADO");
    }
    if (comando.substring(0,4)=="STOP"){
      detachInterrupt(digitalPinToInterrupt(3));comando="";
      stringComplete = false;
      Serial.println("STOPEADO");
    }
  }

  if (stringComplete==true){comando="";stringComplete = false;}
}
}
}

```

10.1.2 Interfaz de usuario a través de la Pantalla táctil

```
Pantalla_arduino_TFG touch
#include <TFTv2.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <Adafruit_TFTLCD.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <SPI.h>
#include <stdint.h>
#include <SeeedTouchScreen.h>
/*#define YP A2 // must be an analog pin, use "An" notation!
#define XM A1 // must be an analog pin, use "An" notation!
#define YM 14 // can be a digital pin, this is A0
#define XP 17 // can be a digital pin, this is A3*/
#define LCD_CS A3 // Chip Select - Pin analogico 3
#define LCD_CD A2 // Command/Data - Pin Analogico 2
#define LCD_WR A1 // LCD Write - Pin Analogico 1
#define LCD_RD A0 // LCD Read - Pin Analogico 0
#define LCD_RESET A4 // LCD Reset - Pin Analogico 4
Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
#define TS_MINX 116*2
#define TS_MAXX 890*2 //320*240
#define TS_MINY 83*2
#define TS_MAXY 913*2
#define TS_MINZ 100
#define TS_MAXZ 1000
#define BOXSIZEH 66
#define BOXSIZEV 55
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define PURPLE 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF
#define ORANGE 0xFBEO
#define LGRAY 0xBDF7
#define DGRAY 0x7BEF
#define BROWN 0x79E0
int toque_x;
int toque_y;
int toque_z;
int menu=0;
int i;
int espacio=0;
int variable=0;
char stringOne[9],stringTen[9],stringPer[9],stringAnc[9];
char stringTenProg[9],stringPerProg[9],stringAncProg[9];
char comando[20];
char comandoTen[20],comandoPer[20],comandoAnc[20];
String str(comando);
String stringtotal,stringfinalTen,stringfinalPer,stringfinalAnc;
double n,valorTen,valorPer,valorAnc;
int x = 320;
int y = 0;
int sensorPin = A5; // select the input pin for the potentiometer // select the pin for the LED
int sensorValue = 0;
int y2 = 0;
unsigned long acumulador=0;

void setup() {
  tft.begin(0x9341);
  Serial.begin(9600);
  TFT_BL_ON; // turn on the background light
  Tft.TFTinit(); // init TFT library
  //Tft.setDisplayDirect(RIGHT2LEFT);
}

void loop() {
  leer_toque();
  switch (menu) {

  case 0: //Pantalla Principal
    delay(100);
    Tft.fillScreen();
    Tft.fillRect(20,120,200,80,CYAN);
    Tft.drawString("START",40,145,4.5,RED);
    Tft.drawString("Universidad",30,15,2,GREEN);
    Tft.drawString("Politecnica",30,45,2,GREEN);
    Tft.drawString("de Valencia",30,75,2,GREEN);
    Tft.drawString("Estimulador",40,230,2.55,GREEN);
```

```

Tft.drawString("Electromecanico", 3, 270, 2.5, GREEN);

do{
  delay(100);
  leer_toque();
  delay(100);
}while(!((toque_x>20) && (toque_x<220) && (toque_y>120) && (toque_y<200)));
menu=1;
Tft.fillScreen();
break;
case 1:
  delay(100);
  Tft.fillScreen();
  Tft.drawString("CANAL", 45, 15, 4, YELLOW);
  Tft.fillRect(30, 70, 80, 80, GREEN);
  Tft.fillRect(130, 70, 80, 80, GREEN);
  Tft.fillRect(30, 230, 80, 80, GREEN);
  Tft.fillRect(130, 230, 80, 80, GREEN);
  Tft.fillRect(30, 160, 180, 60, RED);
  Tft.drawString("1", 55, 90, 5, WHITE);
  Tft.drawString("2", 155, 90, 5, WHITE);
  Tft.drawString("3", 55, 250, 5, WHITE);
  Tft.drawString("4", 155, 250, 5, WHITE);
  Tft.drawString("STOP", 75, 180, 3, WHITE);

for(i=0;i<19;i++){
  comando[i]='\0';
}
valorTen=0;
valorPer=0;
valorAnc=0;
for(i=0;i<9;i++){
  stringfinalTen[i]='\0';
}
for(i=0;i<9;i++){
  stringfinalPer[i]='\0';
}
for(i=0;i<9;i++){
  stringfinalAnc[i]='\0';
}
do{
  delay(100);
  leer_toque();
  delay(100);
  if((toque_x>30) && (toque_x<110) && (toque_y>70) && (toque_y<150)){
    comando[0]='E';
    comando[1]='S';
    comando[2]='T';
    comando[3]='1';
    menu=2;
  }
  if((toque_x>130) && (toque_x<210) && (toque_y>70) && (toque_y<150)){
    comando[0]='E';
    comando[1]='S';
    comando[2]='T';
    comando[3]='2';
    menu=2;
  }
  if((toque_x>30) && (toque_x<110) && (toque_y>230) && (toque_y<310)){
    comando[0]='E';
    comando[1]='S';
    comando[2]='T';
    comando[3]='3';
    menu=2;
  }
  if((toque_x>130) && (toque_x<210) && (toque_y>230) && (toque_y<310)){
    comando[0]='E';
    comando[1]='S';
    comando[2]='T';
    comando[3]='4';
    menu=2;
  }
}
if((toque_x>30) && (toque_x<210) && (toque_y>160) && (toque_y<220)){
  do{
    delay(100);
    leer_toque();
    delay(100);
  }
}

```

```

    if((toque_x>30) && (toque_x<110) && (toque_y>70) && (toque_y<150)) {
        Tft.fillRect(30,70,80,80,ORANGE);
        Tft.drawString("1",55,90,5,WHITE);
        delay(1000);
        Tft.fillRect(30,70,80,80,GREEN);
        Serial.println("EST1TEN0");
        Serial.println("EST1PER0");
        Serial.println("EST1ANCO");
    }
}
if((toque_x>130) && (toque_x<210) && (toque_y>70) && (toque_y<150)) {
    Tft.fillRect(130,70,80,80,ORANGE);
    Tft.drawString("2",155,90,5,WHITE);
    delay(1000);
    Tft.fillRect(130,70,80,80,GREEN);
    Serial.println("EST2TEN0");
    Serial.println("EST2PER0");
    Serial.println("EST2ANCO");
}
if((toque_x>30) && (toque_x<110) && (toque_y>230) && (toque_y<310)) {
    Tft.fillRect(30,230,80,80,ORANGE);
    Tft.drawString("3",55,250,5,WHITE);
    delay(1000);
    Tft.fillRect(30,230,80,80,GREEN);
    Serial.println("EST3TEN0");
    Serial.println("EST3PER0");
    Serial.println("EST3ANCO");
}
}
if((toque_x>130) && (toque_x<210) && (toque_y>230) && (toque_y<310)) {
    Tft.fillRect(130,230,80,80,ORANGE);
    Tft.drawString("4",155,250,5,WHITE);
    delay(1000);
    Tft.fillRect(130,230,80,80,GREEN);
    Serial.println("EST4TEN0");
    Serial.println("EST4PER0");
    Serial.println("EST4ANCO");
}
}

}while(!((toque_x>30) && (toque_x<110) && (toque_y>70) && (toque_y<150) || (toque_x>130) && (toque_x<210) && (toque_y>70)
&& (toque_y<150) || (toque_x>30) && (toque_x<110) && (toque_y>230) && (toque_y<310) || (toque_x>130) && (toque_x<210) &&
(toque_y>230) && (toque_y<310)));

}
}while(!((toque_x>30) && (toque_x<110) && (toque_y>70) && (toque_y<150) || (toque_x>130) && (toque_x<210) &&
(toque_y>70) && (toque_y<150) || (toque_x>30) && (toque_x<110) && (toque_y>230) && (toque_y<310) || (toque_x>130) &&
(toque_x<210) && (toque_y>230) && (toque_y<310)));

Tft.fillScreen();
break;

case 2: //Pantalla 1
delay(100);
Tft.fillScreen();
Tft.fillRect(15,20,210,60,YELLOW);
Tft.fillRect(15,106,210,60,YELLOW);
Tft.fillRect(15,196,210,60,YELLOW);
Tft.fillCircle(120,290,25,RED);
Tft.drawString("TENSION",35,40,3.5,CYAN);
Tft.drawString("PERIODO",38,126,3.5,CYAN);
Tft.drawString("ANCHO",62,216,3.5,CYAN);
Tft.drawString("<-",105,283,2.5,WHITE);

for(i=4;i<19;i++){
comando[i]='\0';
}
for(i=0;i<9;i++){
stringOne[i]='\0';
}
do{
    delay(100);
    leer_toque();
    delay(100);

    if((toque_x>15) && (toque_x<225) && (toque_y>20) && (toque_y<80)) {

        comando[4]='T';
        comando[5]='E';
        comando[6]='N';
        menu=3;
        i=0;
    }
}

```

```

if ((toque_x>15) && (toque_x<225) && (toque_y>106) && (toque_y<166)) {
    comando[4]='P';
    comando[5]='E';
    comando[6]='R';
    menu=3;
    i=0;
}
if ((toque_x>15) && (toque_x<225) && (toque_y>196) && (toque_y<256)) {
    comando[4]='A';
    comando[5]='N';
    comando[6]='C';
    menu=3;
    i=0;
}
if ((toque_x>90) && (toque_x<180) && (toque_y>270) && (toque_y<340)) {
    menu=1;
}
}while (!( (toque_x>15) && (toque_x<225) && (toque_y>20) && (toque_y<80) || (toque_x>15) && (toque_x<225) && (toque_y>106) &&
(toque_y<166) || (toque_x>15) && (toque_x<225) && (toque_y>196) && (toque_y<256) || (toque_x>90) && (toque_x<180) && (toque_y>270) &&
(toque_y<340)));

Tft.fillScreen();
break;
case 3:

Tft.fillScreen();
Tft.fillRect(11, 70, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(11, 132.5, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(11, 195, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(11, 257.5, BOXSIZEH, BOXSIZEV, RED);
Tft.fillRect(87, 70, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(87,132.5, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(87, 195, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(87, 257.5, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(163, 70, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(163,132.5, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(163, 195, BOXSIZEH, BOXSIZEV, ORANGE);
Tft.fillRect(163, 257.5, BOXSIZEH, BOXSIZEV, GREEN);
Tft.drawString("1",32,85,4.5,WHITE);
Tft.drawString("2",108,85,4.5,WHITE);
Tft.drawString("3",184,85,4.5,WHITE);
Tft.drawString("4",32,147.5,4.5,WHITE);
Tft.drawString("5",108,147.5,4.5,WHITE);
Tft.drawString("6",184,147.5,4.5,WHITE);
Tft.drawString("7",32,210,4.5,WHITE);
Tft.drawString("8",108,210,4.5,WHITE);
Tft.drawString("9",184,210,4.5,WHITE);
Tft.drawString("0",108,272.5,4.5,WHITE);
Tft.drawString("DEL",9,272.5,3,WHITE);
Tft.drawString("OK",175,272.5,3,WHITE);

do{

    delay(100);
    leer_toque();
    delay(100);

if ((toque_x>11) && (toque_x<77) && (toque_y>70) && (toque_y<125)) {
    stringOne[i]='1';
    Tft.drawChar('1',espacio,15,4,GREEN);
    i++;
}
if ((toque_x>87) && (toque_x<153) && (toque_y>70) && (toque_y<125)) {
    stringOne[i]='2';
    Tft.drawChar('2',espacio,15,4,GREEN);
    i++;
}
if ((toque_x>163) && (toque_x<229) && (toque_y>70) && (toque_y<125)) {
    stringOne[i]='3';
    Tft.drawChar('3',espacio,15,4,GREEN);
    i++;
}
if ((toque_x>11) && (toque_x<77) && (toque_y>132.5) && (toque_y<187.5)) {
    stringOne[i]='4';
    Tft.drawChar('4',espacio,15,4,GREEN);
    i++;
}
if ((toque_x>87) && (toque_x<153) && (toque_y>132.5) && (toque_y<187.5)) {
    stringOne[i]='5';
    Tft.drawChar('5',espacio,15,4,GREEN);
    i++;
}
}
}

```

```

if((toque_x>163) && (toque_x<229) && (toque_y>132.5) && (toque_y<187.5)) {
    stringOne[i]='6';
    Tft.drawChar('6',espacio,15,4, GREEN);
    i++;
}
if((toque_x>11) && (toque_x<77) && (toque_y>195) && (toque_y<250)) {
    stringOne[i]='7';
    Tft.drawChar('7',espacio,15,4, GREEN);
    i++;
}
if((toque_x>87) && (toque_x<153) && (toque_y>195) && (toque_y<250)) {
    stringOne[i]='8';
    Tft.drawChar('8',espacio,15,4, GREEN);
    i++;
}
if((toque_x>163) && (toque_x<229) && (toque_y>195) && (toque_y<250)) {
    stringOne[i]='9';
    Tft.drawChar('9',espacio,15,4, GREEN);
    i++;
}
}
if((toque_x>87) && (toque_x<153) && (toque_y>257.5) && (toque_y<312.5)) {
    stringOne[i]='0';
    Tft.drawChar('0',espacio,15,4, GREEN);
    i++;
}
}
if((toque_x>11) && (toque_x<77) && (toque_y>257.5) && (toque_y<312.5)) {
    Tft.fillRectangle(1,1,235, 67, BLACK);
    for(i=0; i<9; i++) {
        stringOne[i]='\0';
    }
    i=0;
}
espacio=25*i;
}while(!((toque_z>100) && (toque_x>163) && (toque_x<229) && (toque_y>257.5) && (toque_y<312.5)));
n=atoi(stringOne);
// Serial.print(n);
espacio=0;
i=0;

if(comando[4]=='T' && comando[5]=='E' && comando[6]=='N') {
    valorTen=n;
    itoa(valorTen, stringTen, 10);
    stringtotal= strcat(comando, stringTen);
}

if(comando[4]=='P' && comando[5]=='E' && comando[6]=='R') {
    //n=n/10;
    valorPer=n;
    itoa(n, stringPer, 10);
    stringtotal= strcat(comando, stringPer);
}

if(comando[4]=='A' && comando[5]=='N' && comando[6]=='C') {
    valorAnc=n;
    itoa(n, stringAnc, 10);
    stringtotal= strcat(comando, stringAnc);
}
// stringtotal= strcat(comando, stringTen);

Serial.println(stringtotal);

if(valorTen==0 || valorPer==0 || valorAnc==0) {
    Tft.fillScreen();
    menu=2;
} else {
    Tft.fillScreen();
    menu=4;
}

break;

case 4: //pantalla resultados mas/menos progresivo
delay(100);
    Tft.fillScreen();
    Tft.fillCircle(197, 67, 15, YELLOW); //positivas
    Tft.fillCircle(197, 147, 15, YELLOW);
    Tft.fillCircle(197, 227, 15, YELLOW);
    Tft.fillCircle(37, 67, 15, YELLOW); //negativas
    Tft.fillCircle(37, 147, 15, YELLOW);
    Tft.fillCircle(37, 227, 15, YELLOW);
    Tft.drawString("TENSION", 65, 20, 2.5, CYAN);
    Tft.drawString("PERIODO", 65, 100, 2.5, CYAN);
    Tft.drawString("RANCHO", 80, 180, 2.5, CYAN);
    Tft.drawString("+", 190, 60, 2.5, BLACK);

```

```

Tft.drawString("+",190,140,2.5,BLACK);
Tft.drawString("+",190,220,2.5,BLACK);
Tft.drawString("-",30,60,2.5,BLACK);
Tft.drawString("-",30,140,2.5,BLACK);
Tft.drawString("-",30,220,2.5,BLACK);
Tft.fillCircle(30,290,25,RED);
Tft.drawString("<-",15,283,2.5,WHITE);
Tft.fillCircle(210,290,25,GREEN );
Tft.drawString("SEND",195,287,1,WHITE);
  Tft.drawString(stringTen, 90, 60, 2.7, GREEN);
Tft.drawString(stringPer, 90, 140, 2.7, GREEN);
Tft.drawString(stringAnc, 90, 220, 2.7, GREEN);
Tft.fillRectangle(75, 270, 95, 50, ORANGE);
Tft.drawString("Estimulador",80,280,1,WHITE);
  Tft.drawString("Mecanico",91,300,1,WHITE);

do{

  delay(100);
  leer_toque();
  delay(100);

for(i=4;i<19;i++){
  comando[i]='\0';
}
for(i=0;i<9;i++){
  stringOne[i]='\0';
}

if((toque_x>182)&&(toque_x<212)&&(toque_y>52)&&(toque_y<82)){ //positivo TEN

  valorTen++;
  itoa(valorTen,stringTen,10);
  Tft.fillRectangle(60,40,110, 50,BLACK);
  Tft.drawString(stringTen, 90, 60, 2.7, GREEN);
  comando[4]='T';
  comando[5]='E';
  comando[6]='N';
  stringfinalTen= strcat(comando,stringTen);
}
if((toque_x>182)&&(toque_x<212)&&(toque_y>132)&&(toque_y<162)){ // positivo PER
  if(toque_z>150) { //currentMillis < intervalo pulsacion periodo
    valorPer=valorPer+5;
  }
  if(toque_z<150) {
    valorPer=valorPer+50;
  }
  itoa(valorPer,stringPer,10);
  Tft.fillRectangle(60,120,110, 50,BLACK);
  Tft.drawString(stringPer, 90, 140, 2.7, GREEN);
  comando[4]='P';
  comando[5]='E';
  comando[6]='R';
  stringfinalPer= strcat(comando,stringPer);
}
if((toque_x>182)&&(toque_x<212)&&(toque_y>212)&&(toque_y<242)){ // positivo ANC
  valorAnc++;
  itoa(valorAnc,stringAnc,10);
  Tft.fillRectangle(60,200,110, 50,BLACK);
  Tft.drawString(stringAnc, 90, 220, 2.7, GREEN);
  comando[4]='A';
  comando[5]='N';
  comando[6]='C';
  stringfinalAnc= strcat(comando,stringAnc);
}
if((toque_x>22)&&(toque_x<52)&&(toque_y>52)&&(toque_y<82)){ // negativo TEN
  if(valorTen>0){
    valorTen--;}else{valorTen=0;}
  itoa(valorTen,stringTen,10);
  Tft.fillRectangle(60,40,110, 50,BLACK);
  Tft.drawString(stringTen, 90, 60, 2.7, GREEN);
  comando[4]='T';
  comando[5]='E';
  comando[6]='N';
  stringfinalTen= strcat(comando,stringTen);
}
if((toque_x>22)&&(toque_x<52)&&(toque_y>132)&&(toque_y<162)){ // negativo PER
  if(valorPer>0){
    valorPer--;
    if(toque_z>150) { //currentMillis < intervalo pulsacion periodo
      valorPer=valorPer-5;
    }
  }
  if(toque_z<150) {

```


10.1.2.1 Función touch (leer_toque())

```
void leer_toque(){
  TouchScreen ts = TouchScreen(XP, YP, XM, YM);
  Point p = ts.getPoint();

  if (p.z > __PRESURE) {
    //Serial.print("Raw X = "); Serial.print(p.x);
    //Serial.print("\tRaw Y = "); Serial.print(p.y);
    //Serial.print("\tPressure = "); Serial.println(p.z);
  }

  // if (p.z > TS_MINZ && p.z < TS_MAXZ) {
  p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
  p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);

  toque_x=p.x;
  toque_y=p.y;
  toque_z=p.z;
  //}

  // we have some minimum pressure we consider 'valid'
  // pressure of 0 means no pressing!
  if (p.z > __PRESURE) {
    //Serial.print("X = "); Serial.print(p.x);
    //Serial.print("\tY = "); Serial.println(p.y);
    // Serial.print("\tPressure = "); Serial.println(p.z);
  }

}
```

10.2 Anexo II: Descripción de componentes del sistema y funcionamiento

10.2.1 Estimulador Mecánico

En la Figura 44 podemos observar el conexionado de los componentes electrónicos del estimulador mecánico.

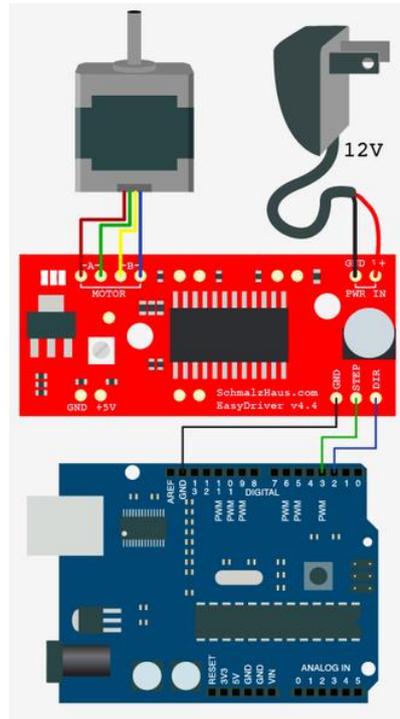


Figura 44 – Esquema de conexiones del estimulador mecánico.

10.2.1.1 Microcontrolador

Arduino es una familia de placas de desarrollo de microcontroladores basadas en microcontroladores de la arquitectura ATMELE AVR y en microcontroladores de arquitectura ARM.

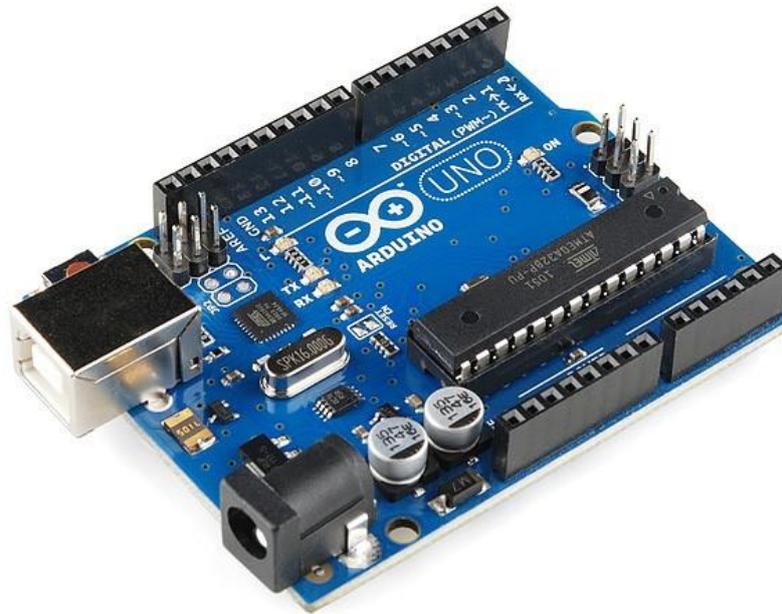


Figura 45 - Placa del Arduino Uno.

El Arduino Uno es una placa de esta familia basada en el microcontrolador ATmega328. Sus características principales son las que figuran en la Tabla 3.

Microcontrolador	ATmega328
Voltaje de operación	5V
Voltaje de entrada recomendado	7-12V
Pines digitales E/S	14 (6 disponen de salida PWM)
Pines de entrada analógica	6
Consumo por pin E/S	40mA
Consumo del pin 3.3V	50mA
Memoria flash	32kB (0.5kB empleados por el bootloader)
SRAM	2kB
EEPROM	1kB
Frecuencia de reloj	16MHz

Tabla 3. Resumen de características del Arduino Uno.

El Arduino puede ser alimentado a través de su puerto USB o de una fuente de alimentación externa. La alimentación externa se puede conectar a través del jack del que dispone la placa o a través de los pines GND y VIN. Ya que deseamos que en nuestro sistema se pueda desconectar el cable USB y que siga funcionando, dicha alimentación queda descartada. Nuestro Arduino irá montado sobre una caja ABS que dispondrá de alimentación, por lo que elegimos alimentarlo a través de los pines GND y VIN.

La placa del Arduino dispone de un regulador lineal de 5V que tomará la tensión de entrada de entre 7-12V y generará una tensión de 5V que sirve para alimentar el microcontrolador.

Adicionalmente dispone de un pin en la placa llamado 5V conectado a la salida de dicho regulador, por lo que nos sirve como pin de alimentación para el resto de dispositivos externos al Arduino que tengamos alimentados a 5V.

El Arduino Uno dispone de un puerto USB y de un Atmega16U2 programado como convertidor USB a puerto serie, por lo que nos permite comunicar el ordenador con el Arduino estableciendo una conexión con el puerto serie emulado.

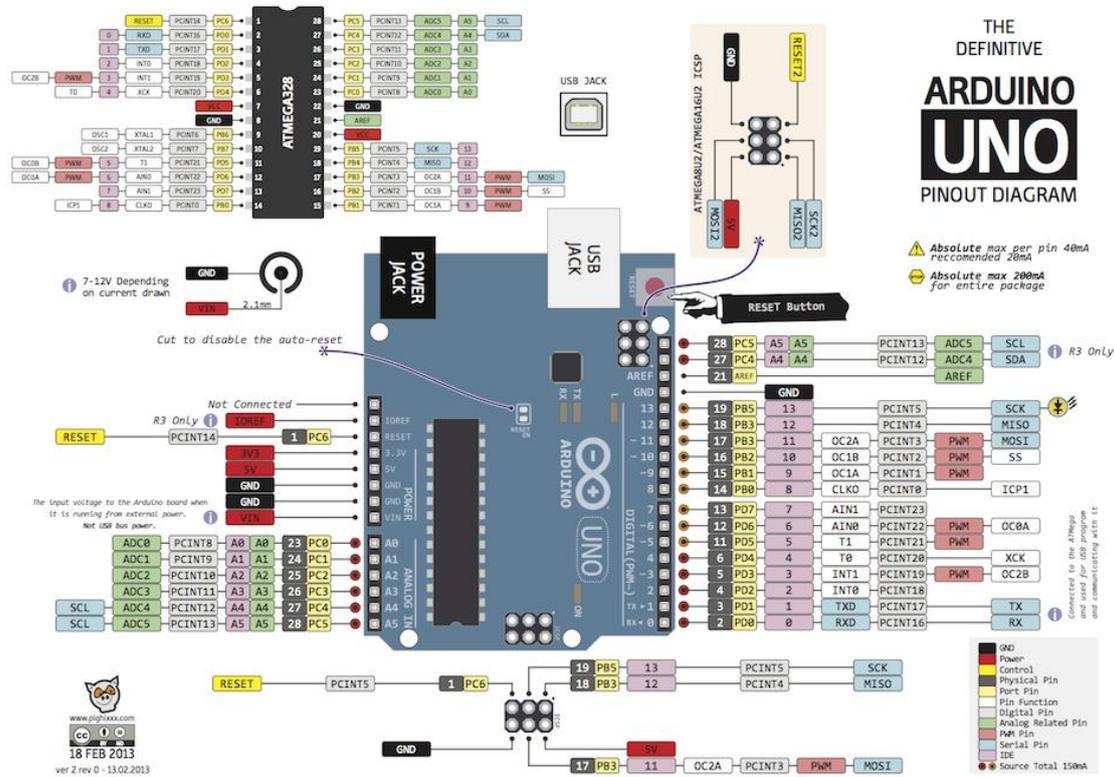


Figura 46. Pinout del Arduino Uno.

En la Figura 46 podemos ver el pinout del Arduino y como van conectados a los distintos pines del ATMEGA328.

10.2.1.2 Motor paso a paso

En la Figura 47 podemos observar el dimensionado y las características de nuestro motor paso a paso.

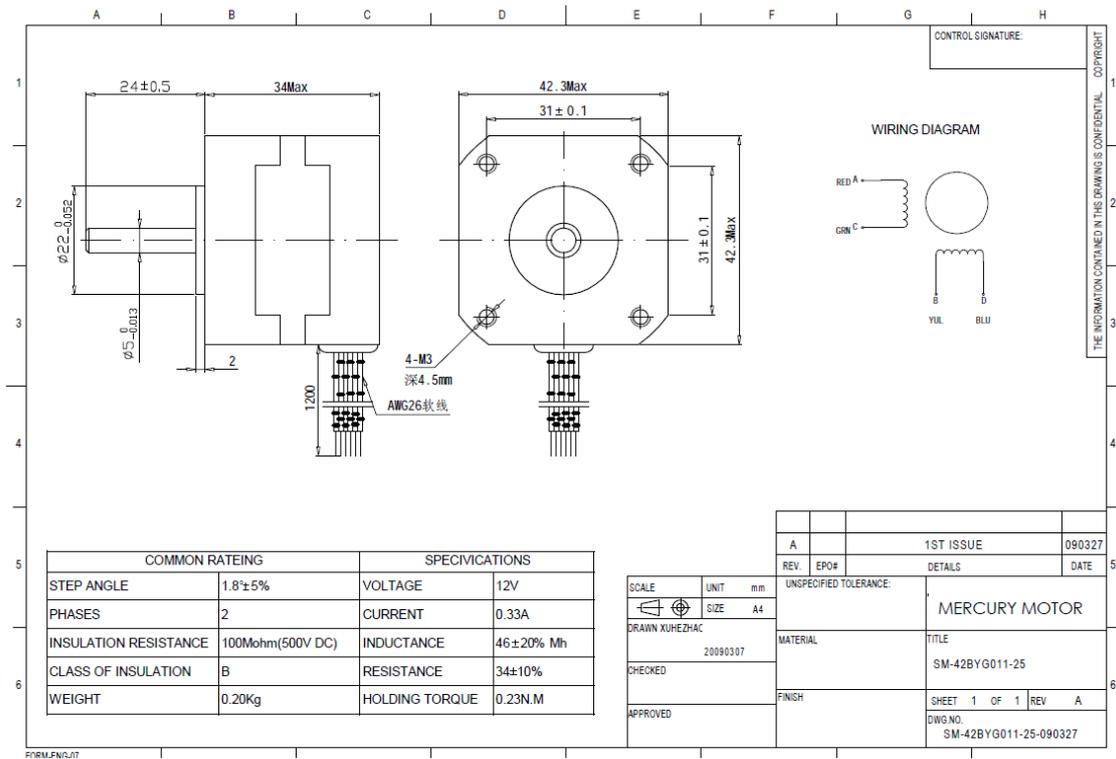


Figura 47. Especificaciones del motor paso a paso.

10.2.2 Estimulador Eléctrico

Como podemos observar en la figura el estimulador eléctrico se compone de cuatro PCBs idénticas, una por cada canal de estimulación. Cada una tendrá incorporado un DAC con diferente código de identificación que permitirá tener cuatro canales de estimulación independientes.

Una vez las placas se encuentren conectadas recibirán órdenes del Microcontrolador a través de 3 comunicaciones diferentes cada una, una comunicación del DAC conectado en bus a todas las placas, las señales de multiplexado y la señal de relé. La conexión SDA y SCL debe ir conectado a los puertos SDA y SCL del Microcontrolador.

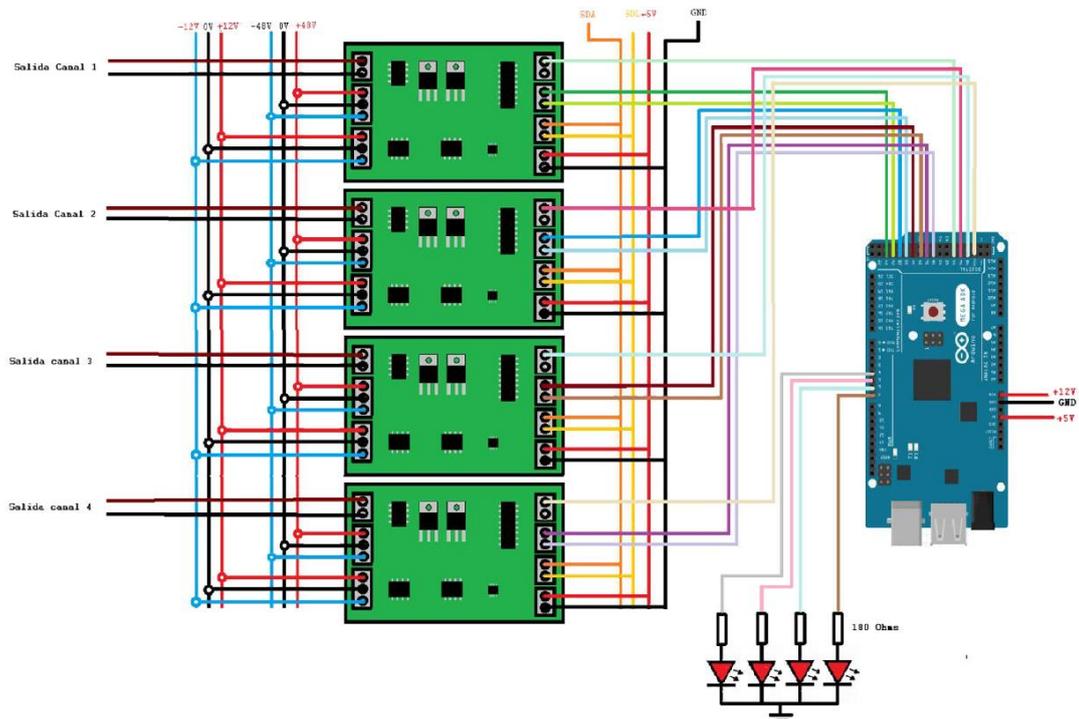


Figura 48. Esquema de conexiones del estimulador eléctrico.

Los diodos LED se instalarán en el frontal de la caja y llevan cada uno una resistencia de 180 Ohms para controlar la corriente. Cada uno de los LEDs va conectado a una salida digital del Microcontrolador.

Para alimentar el sistema se han utilizado tres fuentes de alimentación, dos fuentes de 48 V conectadas en serie formando una fuente simétrica de ± 48 V para la amplificación de la señal de estimulación, y otro fuente de ± 12 V para el funcionamiento del Microcontrolador y de los amplificadores operacionales.