

SISTEMA ÓPTICO PARA LA MEDIDA DE POTENCIALES ELÉCTRICOS EN CÉLULAS CARDIACAS

Ana María Jiménez Pastor

Tutor: Dr. Antonio Mocholí Salcedo

Cotutor: 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, 30 de junio de 2016

Resumen

Las arritmias cardíacas presentan una elevada tasa de mortalidad y son la principal causa de muerte en el mundo. Se caracterizan por una alteración del ritmo cardíaco, afectando considerablemente a la calidad de vida de las personas afectadas por este tipo de enfermedades.

En los últimos años los avances de la tecnología optoelectrónica han permitido el desarrollo de técnicas de registro óptico de la actividad eléctrica cardíaca. Estas técnicas nos permiten obtener un mayor conocimiento de las causas, motivos y perpetuación de las arritmias cardíacas.

Actualmente se están desarrollando técnicas de registro óptico empleando fibra óptica que mejoran los mecanismos tradicionales de cartografía óptica. Por esta razón, se ha diseñado e implementado un sistema de registro óptico basado en fibra óptica.

Para realizar un sistema fácil de emplear por el usuario se ha desarrollado un interfaz gráfico en GUIDE de Matlab que nos permite controlar los distintos componentes que forman nuestro sistema, así como realizar un registro de distintos estudios almacenándolos en un archivo Excel. Para el control del sistema ha sido necesario el empleo de un microcontrolador.

Resum

Les arítmies cardíagues presenten una elevada taxa de mortalitat i són la principal causa de mort en el món. Es caracteritzen per una alteració del ritme cardíac, afectant considerablement la qualitat de vida de les persones afectades per aquest tipus de malalties.

En els últims anys els avanços de la tecnologia optoelectrònica han permès el desenvolupament de tècniques de registre òptic de l'activitat elèctrica cardíaca. Estes tècniques ens permeten obtindre un major coneixement de les causes, motius i perpetuació de les arítmies cardíagues.

Actualment s'estan desenvolupant tècniques de registre òptic emprant fibra òptica que milloren els mecanismes tradicionals de cartografia òptica. Per aquesta raó, s'ha dissenyat i implementat un sistema de registre òptic basat en fibra òptica.

Per a realitzar un sistema fàcil d'emprar per l'usuari s'ha desenvolupat una interfície gràfica en GUIDE de Matlab que ens permet controlar els diferents components que formen el nostre sistema, com ara realitzar un registre de diferents estudis emmagatzemant-los en un arxiu Excel. Per al control del sistema ha sigut necessari la utilització d'un microcontrolador.

Abstract

Cardiac arrhythmias have a high mortality rate and are the leading cause of death worldwide. They are characterized by an abnormal heart rhythm, significantly affecting the quality of life of people affected by these diseases.

In the last years technological advances in optoelectronics have allowed the development of optical mapping techniques. These techniques allow us to obtain a better knowledge of the causes, reasons and perpetuation of cardiac arrhythmias.

Nowadays new optical mapping techniques using optical fiber are being developed, these improve traditional techniques used in this field. For this reason, an optical mapping system based on optical fiber has been designed and implemented.

To make this system user-friendly a graphical interface in Matlab GUIDE has been developed. This interface allow the user to control all the components that form our system and it also allow the user to register the different studies done, storing them on an Excel file. To control the whole system a microcontroller has been used.

Índice

Capítulo 1.	Introducción	6
Capítulo 2.	Objetivos	7
2.1	Motivación	7
Capítulo 3.	Metodología	8
3.1	Consideraciones técnicas.....	8
3.1.1	Potencial de acción cardiaco	8
3.1.2	Cartografía óptica.....	10
3.2	Gestión del proyecto.....	13
3.3	Distribución de tareas.....	13
3.4	Diagrama temporal.....	13
Capítulo 4.	Desarrollo y resultados.....	14
4.1	Búsqueda de componentes	14
4.1.1	Fibra óptica.....	14
4.1.2	Fotoemisor. Diodo LED.....	15
4.1.3	Fotodetector.....	16
4.1.4	Filtros	17
4.2	Hardware	17
4.2.1	Emisor óptico	18
4.2.2	Receptor óptico	19
4.2.3	Estructura de unión fibra óptica - filtro	22
4.2.4	Microcontrolador.....	24
4.2.5	Comunicación inalámbrica.....	26
4.2.6	Caja para instrumental de laboratorio	28
4.3	Software	29
4.3.1	Firmware de Arduino	29
4.3.2	Interfaz gráfica de usuario.....	30
4.4	Montaje circuito sobre placa perforada.....	37
4.5	Resultados	37
4.5.1	Nivel de tensión a la salida en función del voltaje en el LED de iluminación	37
4.5.2	Señal a la salida en función de la señal generada en el diodo LED.	39
4.5.3	Función de transferencia	40
Capítulo 5.	Pliego de condiciones.....	41
5.1	Condiciones de nuestro sistema	41
5.2	Presupuesto	41

5.2.1	Humano	41
5.2.2	Material	42
5.2.3	Total	43
Capítulo 6.	Conclusiones y propuesta de trabajo futuro	44
6.1	Propuesta de trabajo futuro	44
Capítulo 7.	Bibliografía.....	45
Anexo I.	Importación librería temporizador.....	47
Anexo II.	Programación XBee mediante X-CTU	48
Anexo III.	Firmware de Arduino.....	51
Anexo IV.	GUIDE de Matlab	53
IV.1	Pantalla registro de datos	53
IV.2	Pantalla representar medida almacenada	59

Índice Figuras

Figura 1. Potencial de acción cardiaco.....	9
Figura 2. Potencial de acción en diferentes zonas del corazón y su correlación con el electrogradiograma de superficie.	9
Figura 3. Sistema de cartografía óptica.....	10
Figura 4. Bandas de excitación (azul) y de emisión (rojo) del di-4-ANEPPS	12
Figura 5. Bandas de excitación (azul) y de emisión (rojo) del Rhod-2.....	12
Figura 6. Fibra óptica empleada en nuestra aplicación.	14
Figura 7. Atenuación fibra óptica en función de la longitud de onda	15
Figura 8. Estructura interna del diodo LED empleado en nuestra aplicación	15
Figura 9. Diodo LED empleado en nuestra aplicación	16
Figura 10. Fotodetector empleado en nuestra aplicación	16
Figura 11. Filtro empleado para eliminar la señal luminosa procedente del led (Izquierda) y su correspondiente función de transferencia (Derecha).....	17
Figura 12. Pieza en la que introduciremos el filtro para que este no se dañe.....	17
Figura 13. Diagrama general del sistema	18
Figura 14. Esquemático circuito electrónico emisor óptico	19
Figura 15. Emisores ópticos en funcionamiento.	19
Figura 16. Esquemático circuito electrónico receptor óptico	20
Figura 17. Fuente de alimentación simétrica empleada	20
Figura 18. Amplificador operacional TL071 y sus pines.....	21
Figura 19. Comparación del ruido en un amplificador de transimpedancia para distintos valores de R_f (resistencia de realimentación) en función del tipo de amplificador operacional empleado	21
Figura 20. Impresora 3D Makerbot Replicator 2X, empleada para imprimir nuestros diseños 3D	22
Figura 21. Vistas del diseño 3D de la cámara realizada para acoplar el filtro a la fibra óptica ..	23
Figura 22. Diseño 3D del tubo por el que se introduce la fibra óptica hacia el filtro.....	23
Figura 23. Vistas del diseño 3D de la tapa para la cámara.....	23
Figura 24. Cámara completa impresa cerrada	24
Figura 25. Cámara completa impresa abierta (vistas interior y exterior)	24
Figura 26. Arduino Due	25
Figura 27. Instalar tarjeta Arduino Due en el IDE	26
Figura 28. Comprobación correcta instalación tarjeta Arduino Due.....	26
Figura 29 . Módulo XBee.....	27
Figura 30. Esquema conexión XBee al arduino y al PC.	28
Figura 31. Caja diseñada para instrumental de laboratorio	28
Figura 32. Sistema en caja para instrumental de laboratorio impresa.....	29

Figura 33. Menú principal interfaz de usuario	30
Figura 34. Pantalla registro de datos	31
Figura 35. Pantalla adquisición de datos. Bloque conexión.....	31
Figura 36. Pantalla adquisición de datos. Bloque registro de datos	32
Figura 37. Pantalla adquisición de datos. Bloque control LEDs.....	32
Figura 38. Pantalla adquisición de datos. Bloque almacenar medida	33
Figura 39. Representar medida almacenada. Filtrado únicamente por fecha.....	33
Figura 40. Representar medida almacenada. Filtrado por fecha y tipo de medida.....	34
Figura 41. Representar medida almacenada. Selección medida en Excel.....	35
Figura 42. Bloque procesado de la señal medida almacenada	35
Figura 43. Registro con muchas muestras. Mala visualización.....	36
Figura 44. Registro con muchas muestras. Rango visible menor.....	36
Figura 45. Parte delantera (Izquierda) y trasera (Derecha) del circuito soldado sobre una placa perforada	37
Figura 46. Tensión a la salida del sistema (fotodetector + amplificador + filtro) para distintos valores de tensión de entrada al led verde.....	38
Figura 47. Tensión a la salida del sistema (fotodetector + amplificador + filtro) para distintos valores de tensión de entrada al led rojo	38
Figura 48. Respuesta Fotodetector (amarillo) ante una señal generada en el diodo (azul).	39
Figura 49. Respuesta obtenida en el fotodetector registrándola con el interfaz de Matlab	39
Figura 50. Función de transferencia	40
Figura 51. Verificación de librería añadida.....	47
Figura 52. Conexión con el XBee realizada correctamente	48
Figura 53. XBee esclavo	48
Figura 54. Parámetros XBee coordinador	49
Figura 55. Conexión entre los XBee realizada correctamente	50

Índice de Tablas

Tabla 1. Diagrama temporal.....	13
Tabla 2. Características de la fibra óptica empleada en nuestra aplicación	14
Tabla 3. Características diodos LED necesarios para registrar el potencial de acción (rojo) y el nivel de calcio intracelular (verde).....	16
Tabla 4. Características fotodetector empleado	16
Tabla 5. Características Arduino Due	25
Tabla 6. Características Xbee Serie 2.....	27
Tabla 7. Presupuesto horas humanas.....	41

Capítulo 1. Introducción

La Fibrilación Auricular (FA) es la arritmia cardíaca más frecuente en humanos y se asocia a una elevada tasa de morbilidad y mortalidad, aumentando el riesgo de muerte, insuficiencia cardíaca congestiva y fenómenos embólicos, incluido el accidente cerebrovascular. Tanto su incidencia como su prevalencia son altas. En España, la prevalencia de la FA en la población general mayor de 40 años es elevada, del 4,4%. La prevalencia es similar en varones y mujeres y se incrementa escalonadamente a partir de los 60 años. Se estima en más de 1 millón de pacientes con FA en la población española, de los que más de 90.000 están aún sin diagnosticar [1]. A pesar de ser la arritmia más común, sus mecanismos de origen y persistencia aún no son del todo comprendidos, por lo que los tratamientos actuales tienen una baja tasa de éxito. Por ello, son necesarios mayores esfuerzos para mejorar la prevención y el tratamiento de pacientes con FA [2].

La FA está caracterizada por una propagación aparentemente caótica e irregular de la señal eléctrica a través de la aurícula, haciendo ineficaz la contracción del músculo auricular e interrumpiendo el flujo de sangre de la aurícula al ventrículo. Aunque no es una arritmia mortal, implica un serio deterioro en la calidad de vida del paciente y aumenta de manera considerable la posibilidad de sufrir un accidente tromboembólico [3]. Algunos autores sostienen que la hipertensión juega un papel importante en el origen de los procesos fisiológicos. El análisis de las alteraciones electrofisiológicas producidas por aumentos de la presión dentro de la aurícula es de gran importancia para la completa comprensión de los mecanismos de mantenimiento de la arritmia.

En los últimos años los avances de la tecnología optoelectrónica han permitido el desarrollo de técnicas de registro óptico de la actividad eléctrica cardíaca. Estas técnicas se han convertido en una importante herramienta para la investigación experimental de enfermedades cardiovasculares, permitiendo la obtención de registros de la actividad eléctrica del corazón mucho más cercanos a la realidad [4]. El estudio de los cambios electrofisiológicos producidos por la hipertensión mediante el registro óptico de la actividad cardíaca permitirá el desarrollo de mejores terapias para el tratamiento de la FA.

Se emplean mecanismos de registro óptico que, con la ayuda de unos tintes potenciométricos, permiten generar una intensidad lumínica en relación directa al potencial al que se encuentran las células cardíacas excitadas con una luz de cierta longitud de onda. Esta iluminación se puede registrar empleado detectores para, posteriormente, analizar la señal obtenida.

Debido a los latidos del corazón, se produce un movimiento que varía la cantidad de luz recibida en la zona de estudio, por lo que se introduce ruido en la señal registrada por el equipo óptico. Por esta razón, tradicionalmente se ha necesitado emplear unos fármacos (desacopladores electromecánicos) para detener, durante el registro óptico, el movimiento del corazón. Sin embargo, estos fármacos pueden introducir alteraciones en el comportamiento electrofisiológico bajo estudio.

Por esta razón, en la actualidad, se están analizando nuevos métodos de registro óptico empleando fibra óptica de modo que no sea necesario desacoplar la actividad mecánica del corazón y poder, igualmente, realizar un correcto registro del comportamiento electrofisiológico del corazón.

En el presente trabajo se ha diseñado, desarrollado y validado un sistema de registro óptico empleando fibra óptica que permite el registro de variables fisiológicas como son, el potencial de acción de las células cardíacas o el nivel de calcio intracelular. Variables que nos permiten detectar anomalías que se produzcan en el corazón.

Capítulo 2. Objetivos

El objetivo principal del presente trabajo es el diseño, construcción y validación de un sistema de registro óptico capaz de registrar la actividad cardiaca empleando fibra óptica en corazón aislado porcino.

Para ello, se han marcado unos objetivos generales que permitirán llevar a cabo este proyecto:

- Analizar el estado del arte de sistemas de registro óptico.
- Diseñar un sistema de registro óptico empleando fibra óptica.
- Diseño de los circuitos de acondicionamiento electrónicos.
- Programación de un software de control de todo el sistema para PC.
- Montaje del sistema en una placa perforada.
- Comprobación del correcto funcionamiento del sistema.

2.1 Motivación

La principal motivación para el presente trabajo es la de proveer un equipo de registro óptico para la experimentación con corazón aislado porcino empleando fibra óptica. La inexistencia de equipos que se adapten a las necesidades del proyecto ha sido la causa principal de la realización del presente trabajo. Además tiene un coste más reducido implementar un sistema propio y proporciona una mayor versatilidad al usuario.

La principal razón por la que se ha decidido implementar este sistema es la necesidad de emplear desacopladores electromecánicos (fármacos que inhiben el movimiento del corazón) en sistemas tradicionales de registro óptico que no emplean fibra óptica, lo cual introduce pequeñas alteraciones en la señal registrada. Mediante el empleo de fibra óptica se pretende evitar el uso de estos fármacos y obtener, de este modo, una señal pura. Esto se debe a que, al estar la fibra en contacto con el corazón, se mueve la fibra sincronamente con el corazón al latir.

Capítulo 3. Metodología

3.1 Consideraciones técnicas

3.1.1 Potencial de acción cardiaco

El corazón es un órgano muscular localizado en el centro de la cavidad torácica. Su principal función es bombear sangre a través de todo el cuerpo gracias a sus contracciones rítmicas, permitiendo de esta forma un flujo continuo de sangre [5]. Las contracciones del corazón ocurren gracias a la activación eléctrica de las células miocárdicas. Estas activaciones eléctricas se llaman potenciales de acción (PA), un PA es un cambio reversible en el potencial de membrana producido por la activación secuencial de diversas corrientes iónicas generadas por la difusión de iones a través de la membrana a favor de su gradiente electroquímico. Así, durante la despolarización, el interior celular pasa de estar cargado negativamente a estarlo positivamente (alcanzando +20 ó +30 mV) para luego repolarizarse hasta recuperar de nuevo los -85mV. [6]

El término clave en este ámbito es el potencial de membrana, que es la diferencia de potencial electroquímico a ambos lados de una membrana que separa dos soluciones de diferente concentración de iones.

En la Figura 1 podemos observar un esquema del potencial de acción cardiaco y sus distintas fases. [6] [7] [8]

- Durante la fase 0 se produce una rápida despolarización de las células debida a la apertura de los canales de Na^+ , lo cual produce una rápida entrada de los iones de Na^+ hacia el interior celular, esto modifica el potencial de membrana, alejándose del potencial de equilibrio. Estos canales permanecen abiertos durante 1-2 ms y después pasan a estado inactivo (se cierran).

En la repolarización cardiaca distinguimos tres fases:

- Durante la fase 1 se produce una rápida repolarización debida a una corriente de K^+ hacia el exterior de la célula. No todas las células presentan esta corriente de K^+ , sólo aquellos tejidos en los que está presente (i.e. His-Purkinje y epicardio ventricular) generar un PA con una fase 1 muy marcada.
- Durante la fase 2 tenemos dos corrientes de entrada, una de Na^+ y otra de Ca^{2+} , esta última dispara la contracción de la célula.
- Durante las fase 3 la repolarización se acelera por la inactivación de las corrientes entrantes de Na^+ y Ca^{2+} y por la salida de K^+ hacia el exterior celular, alcanzando de nuevo el potencial de reposo.
- Una vez que la célula se encuentra repolarizada, el potencial permanece estable hasta que la célula se despolariza de nuevo. A esta fase entre dos potenciales de acción se le denomina fase 4 (periodo refractario). Durante esta fase se produce la diástole y se restituyen las concentraciones iónicas a ambos lados de la membrana.

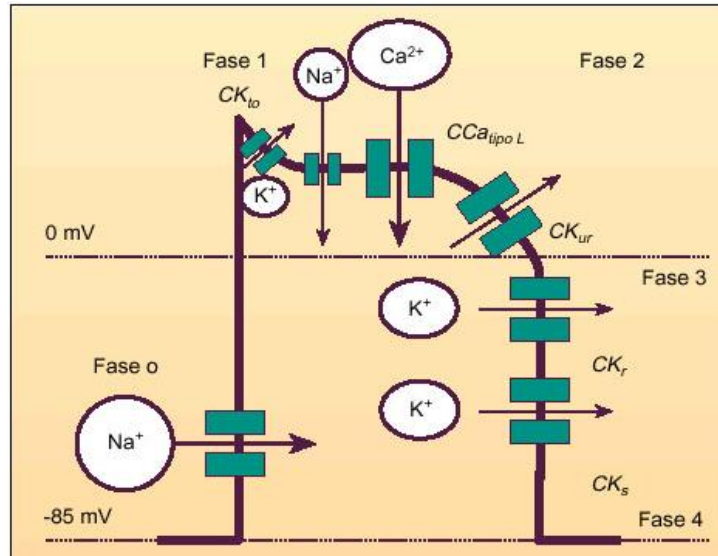


Figura 1. Potencial de acción cardíaco.

Fuente: J.A. Lozano. Arritmias cardíacas y su tratamiento

Como se puede apreciar, el ión Ca^{2+} juega un papel importante en la regulación de la excitación y contracción de las células miocárdicas. En el momento en que comienza el potencial de acción, los canales de calcio se abren permitiendo la entrada de estos iones hacia el interior celular hasta que se produce la contracción. Una vez producida la contracción, el calcio es extraído de la célula gracias a las bombas de calcio. Estas bombas hacen retornar la célula a su estado de reposo y la dejan lista para un nuevo potencial de acción.

Este proceso debe permanecer inalterado durante todos los potenciales de acción, cualquier anomalía en el manejo de Ca^{2+} puede causar anomalías cardíacas.

Los potenciales de acción cardíacos presentan importantes diferencias morfológicas dependiendo del tejido cardíaco analizado (aurícula o ventrículo) o incluso dentro de un mismo tejido (endocardio o epicardio, músculo ventricular o células de Purkinje), que son consecuencia de cambios en la expresión de los canales iónicos implicados en su génesis.

En la Figura 2 podemos observar cómo puede variar el PA en función de la zona de análisis.

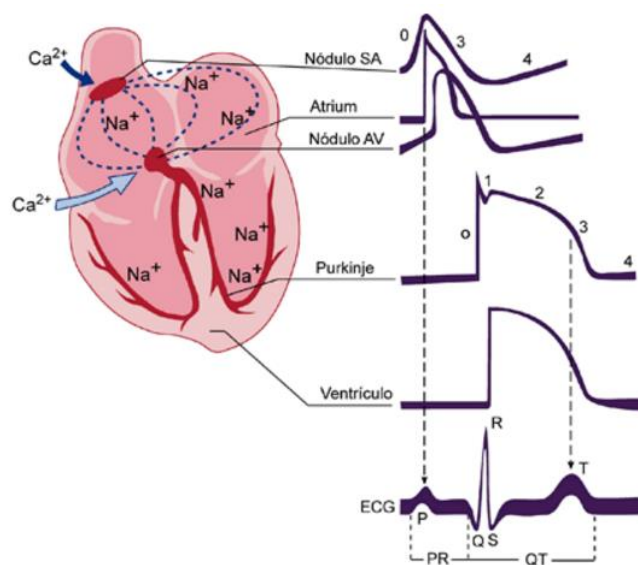


Figura 2. Potencial de acción en diferentes zonas del corazón y su correlación con el electrocardiograma de superficie.

Fuente: Proyecto ITACA. Canalopatías. Conceptos. Potencial de Acción Cardíaco.

Las fases del potencial de acción se corresponden con las del electrocardiograma (ECG), esto queda reflejado también en la Figura 2. La onda P refleja la despolarización auricular (fase 0), el complejo QRS la despolarización ventricular, el intervalo PR refleja la velocidad de conducción a través del nódulo AV, el complejo QRS la velocidad de conducción intraventricular y el intervalo QT la duración del potencial de acción ventricular. La elevación del segmento ST refleja el gradiente transmural del voltaje durante la fase de meseta del potencial de acción. [6]

3.1.2 Cartografía óptica

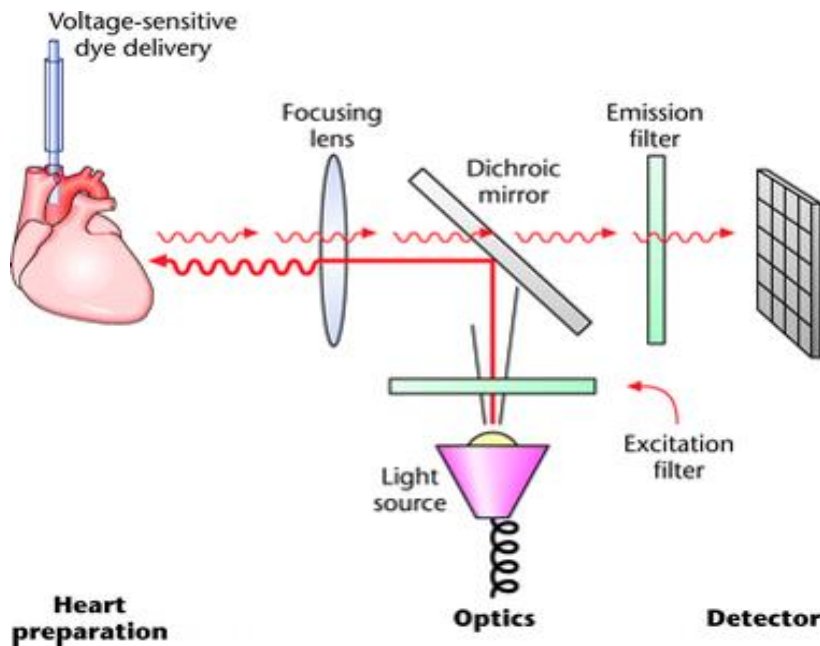


Figura 3. Sistema de cartografía óptica

Formado por tres partes principales: 1) el corazón aislado con el tinte potenciométrico; 2) un sistema óptico; y 3) un detector. Se emplea un filtro de excitación que deje pasar las longitudes de onda apropiadas desde la fuente de luz hacia el espejo dichroico, que refleja estas longitudes de onda hacia el corazón. Como respuesta a la luz de excitación, las moléculas sensibles al voltaje del tinte emiten luz en proporción al potencial de acción de la célula a la que están unidas. Esta luz emitida, de longitud de onda mayor, atraviesa el dichroico y un nuevo filtro hasta llegar al detector. Fuente: *Indian Pacing Electrophysiol. J.* 2003;3(4):187-196

La cartografía óptica es una técnica utilizada en investigación cardiaca para entender la actividad eléctrica del corazón, la homeostasis del calcio y su metabolismo. Estos son parámetros clave para poder entender el origen y mantenimiento de diferentes arritmias [9]

El estudio de las propiedades electrofisiológicas tanto en corazones sanos como enfermos aportan importantes explicaciones sobre como las arritmias cardiacas pueden ocurrir. [10]

Tradicionalmente, los electrodos de superficie se han utilizado, y se siguen utilizando, para medir potenciales cardiacos eléctricos extracelulares. Sin embargo, este tipo de mapeo tiene muy poca resolución espacial, por ello se emplean además pruebas de fluorescencia que ofrecen una mayor resolución espacial y temporal. [11]

Este tipo de técnicas se consiguen empleando tintes potenciométricos, sensibles al voltaje o tintes sensibles a la concentración de iones. Los tintes potenciométricos fueron introducidas por primera vez por Salzberg et al [12] en 1973 para analizar los cambios producidos durante la propagación de los impulsos a lo largo del axón neuronal. Desde entonces, se han empleado

tintes fluorescentes para estudiar una gran cantidad de órganos y tejidos eléctricamente excitables.

Los tintes potenciométricos, sensibles al voltaje, poseen la particularidad de emitir una cierta cantidad de fotones en función del potencial al cual está sometida la membrana celular a la que el tinte está adherido en el momento que recibe la radiación luminosa. Esto significa que al aplicar tinte sobre la superficie de un corazón aislado, dicho tejido emite una intensidad lumínica en relación directa al potencial de acción al cual se encuentran las células. [13]. Además, empleando tintes sensibles a la concentración de calcio (Ca^{2+}) también se puede estudiar el comportamiento del calcio intracelular.

En tejidos cardiacos, el registro de la luz emitida por el tinte en diversas zonas simultáneas se ha realizado empleando arrays de fotodiodos, fotomultiplicadores, sistemas detectores de láser, cámaras CCD y sus derivados más recientes. [14] [15]

Desde principios de los 90, la cartografía óptica empleando cámaras de alta velocidad (CCD) se ha convertido en una herramienta muy eficaz en el análisis de la propagación de señales cardiacas en arritmias auriculares y ventriculares en corazones de animales aislados. [16]

3.1.2.1 Fotodetectores

Los experimentos de cartografía óptica requieren una fuente de luz para iluminar la superficie cardiaca y un receptor óptico que permita registrar los cambios de luz que representan los cambios fisiológicos en el potencial transmembrana o en las concentraciones iónicas intracelulares.

Para poder apreciar estos cambios se requieren fotodetectores de alta velocidad y bajo ruido. Los arrays de fotodiodos (PDAs) y cámaras CCD son muy eficientes debido a su gran resolución espacial y temporal. [11]

Uno de los inconvenientes más importantes que presenta la técnica de mapeo óptico son los artefactos que se introducen en la señal, asociados al movimiento mecánico que presenta el corazón. Debido a los latidos del corazón se produce un movimiento que varía la cantidad de luz recibida, por lo que se introduce ruido en la señal registrada. Para evitar que ocurra dicho ruido se emplean fármacos (como el BDM o la blebbistatina) que desacoplan la actividad mecánica del corazón. De esta manera se inhibe el movimiento del corazón pero no la actividad eléctrica. Sin embargo, aunque estos fármacos permitan reducir el ruido, pueden introducir alteraciones en el comportamiento electrofisiológico.

Según Liu et al. [17] éstos fármacos pueden alterar la duración del potencial de acción y el periodo refractario, provocando la disminución de dichas duraciones, lo cual puede derivar en que estos desacoples electromecánicos presenten efectos pro o antiarrítmicos.

Debido a este hecho actualmente se están desarrollando nuevas técnicas de registro óptico, como puede ser el empleo de fibra óptica. Mediante esta técnica se puede medir la fluorescencia de un punto o zona del corazón sin necesidad de desacoplar la actividad cardiaca.

3.1.2.2 Iluminación

Emplear una iluminación apropiada es vital para realizar un correcto registro de la actividad cardiaca. Una luz estable durante el proceso de registro es esencial para obtener una relación señal / ruido elevada.

Tradicionalmente se han empleado métodos de iluminación como las lámparas de xenón, mercurio o las alógenas [18]. Sin embargo, tecnologías más modernas como los diodos LED ofrecen una alternativa más estable y fácil de emplear. Esta tecnología es, además, barata, eficiente, portátil y flexible. [19]

Actualmente se emplean LEDs de alta potencia que ofrecen mayor potencia y estabilidad para experimentos que requieren mayor cantidad de luz como puede ser el estudio del corazón completo.

3.1.2.3 Tintes

Los tintes potenciométricos se clasifican en tintes de respuesta rápida o de respuesta lenta en función de sus tiempos de respuesta y sus sensibilidades [20].

Los tintes de respuesta rápida detectan cambios del orden de microsegundos [21]. Las más comunes son los tintes ANEP: di-4-ANEPPS y di-8-ANEPPS, ambos presentan un cambio uniforme del 10% en la fluorescencia por cada cambio de 100mV en el potencial de membrana. Ambos responden a incrementos del potencial de membrana con un decremento de luz emitida. En la Figura 4 podemos observar las bandas de excitación (azul) y de emisión (rojo) del di-4-ANEPPS.

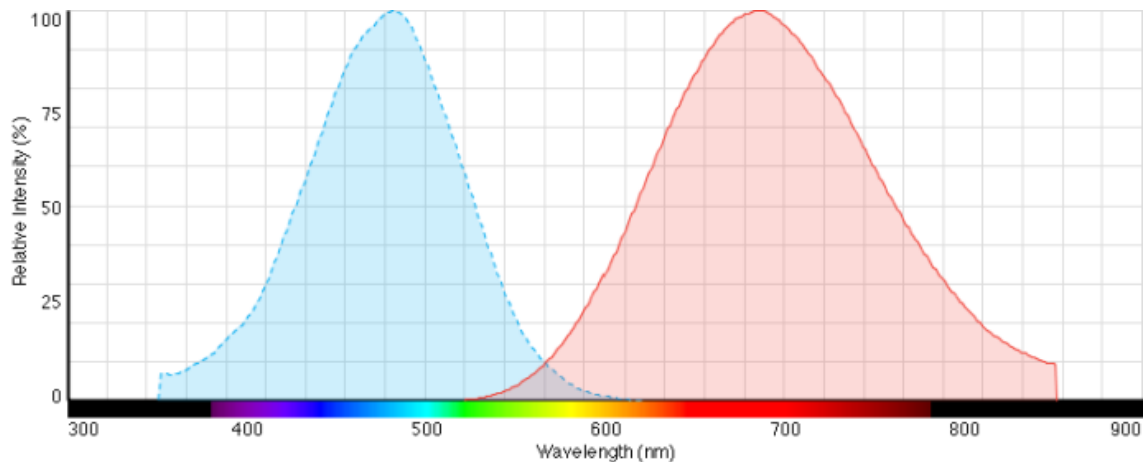


Figura 4. Bandas de excitación (azul) y de emisión (rojo) del di-4-ANEPPS

Como hemos comentado anteriormente, conocer el nivel de calcio intracelular es también esencial para poder estudiar el correcto comportamiento del corazón. Por ello existen también tintes sensibles a este nivel de calcio. Se emplean tintes como por ejemplo el Fluo-4, Fluo-3 o Rhod-2 [22]. En la Figura 5 podemos observar las bandas de excitación (azul) y de emisión (rojo) del Rhod-2.

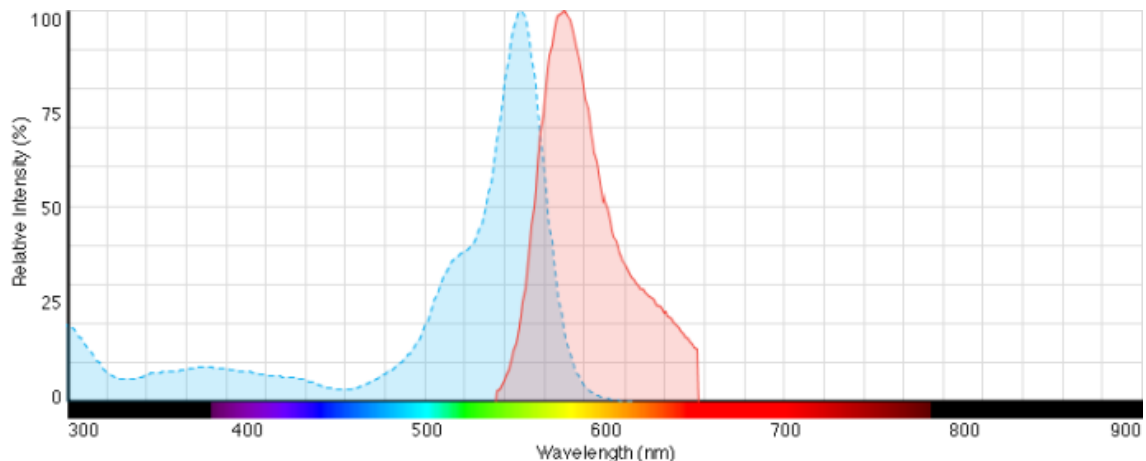


Figura 5. Bandas de excitación (azul) y de emisión (rojo) del Rhod-2.

3.1.2.4 Filtros

En la recepción, antes de que le llegue la señal al fotodetector, será necesario emplear un filtro, de manera que únicamente deje pasar el conjunto de longitudes de onda que emite el tinte, filtrando, así, la señal que emite el diodo LED para que ésta no sature el fotodetector.

3.2 Gestión del proyecto

El presente trabajo ha sido llevado a cabo en el instituto de investigación de ingeniería biomédica ITACA. Este grupo cuenta con una larga experiencia en el estudio de arritmias cardíacas, especialmente la FA. Además, cuenta con el apoyo de otros grupos como la unidad de investigación del Hospital Gregorio Marañón en Madrid, con el que se ha colaborado estrechamente para la elaboración de este trabajo.

Además se han realizado una serie de reuniones periódicas con el grupo para revisar el avance de los proyectos.

El instituto ha proporcionado, además, todo el material necesario para construir el sistema así como para realizar las pruebas necesarias.

3.3 Distribución de tareas

1. Analizar el estado del arte de la cartografía óptica para conocer su funcionamiento y sus posibles mejoras.
2. Diseñar un sistema de registro óptico empleando fibra óptica.
3. Diseño de un sistema de acondicionamiento electrónico para el fotodetector.
4. Gestión de la compra de los distintos componentes necesarios para implementar el sistema diseñado.
5. Programación del firmware del microcontrolador.
6. Programación del software de control para PC.
7. Comunicación inalámbrica.
8. Montaje de una placa de prototipo para comprobar el correcto funcionamiento del emisor y receptor óptico.
9. Test conjunto software-hardware.
10. Montaje sistema sobre placa perforada.
11. Comprobación correcto funcionamiento de la placa perforada.

3.4 Diagrama temporal

	SEP	OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										

Tabla 1. Diagrama temporal

Capítulo 4. Desarrollo y resultados

4.1 Búsqueda de componentes

Lo primero que se hizo para llevar a cabo el proyecto fue realizar una búsqueda exhaustiva de los componentes que se iban a emplear para poder llevar a cabo nuestros propósitos.

4.1.1 Fibra óptica

Se decidió emplear dos fibras ópticas simples para implementar nuestra aplicación, una encargada de llevar la iluminación del diodo LED hacia el corazón y otra encargada de llevar la iluminación generada en el corazón hacia el fotodetector. Dado que se quiere que la fibra esté en contacto con el corazón para analizar este correctamente, era necesario que no hubiera ningún tipo de conector en este terminal.

Tras realizar un estudio de las fibras que hay en el mercado se decidió emplear las fibras ópticas que ofrece el fabricante *Industrial Fiber Optics, Inc.*, (Figura 6), ya que además ofrece un abanico de componentes que se adaptan perfectamente a estas, como veremos a continuación.

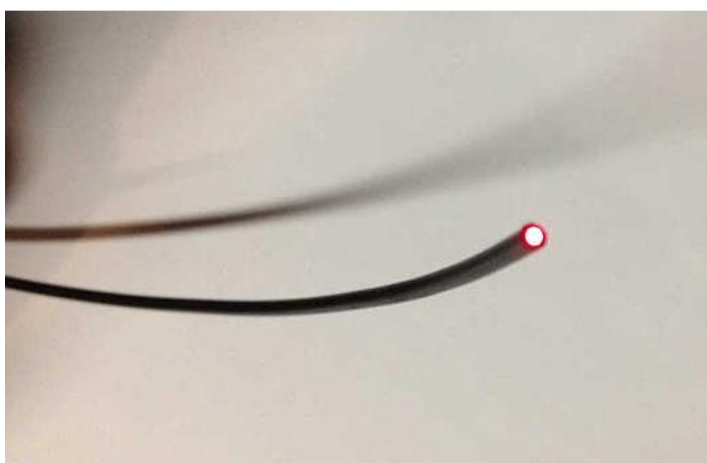


Figura 6. Fibra óptica empleada en nuestra aplicación.

En la Tabla 2 podemos observar las características que ofrece este modelo.

FIBRA ÓPTICA FB140-1-ND [24]	
Tipo	Multimodo
Diámetro del núcleo	980 μ m
Diámetro del núcleo + cobertura	1000 μ m
Apertura Numérica	0,5
Atenuación	0,2 dB/m
Coefficiente de refracción del núcleo	1,492

Tabla 2. Características de la fibra óptica empleada en nuestra aplicación

Esta fibra además posee la característica de que el núcleo no es de cristal sino de plástico, por lo tanto, no son necesarias cortadoras especiales, si necesitamos cortarla, con una cuchilla, realizando un corte de 90°, será suficiente.

En la Figura 7 podemos observar la gráfica correspondiente a la atenuación que ofrece la fibra óptica empleada en función de la longitud de onda de la señal. Para nuestra aplicación emplearemos señales que se encuentran entre los 400 y los 750 nm (espectro visible), por lo tanto podemos observar que para este rango tenemos pérdidas de entre 200 y 2 dB/km (menores cuando mayor sea la longitud de onda). Sin embargo, nosotros vamos a emplear fibras de únicamente 1m de longitud, por lo tanto, las pérdidas se encontrarán únicamente entre 0,2 y 0,002 dB.

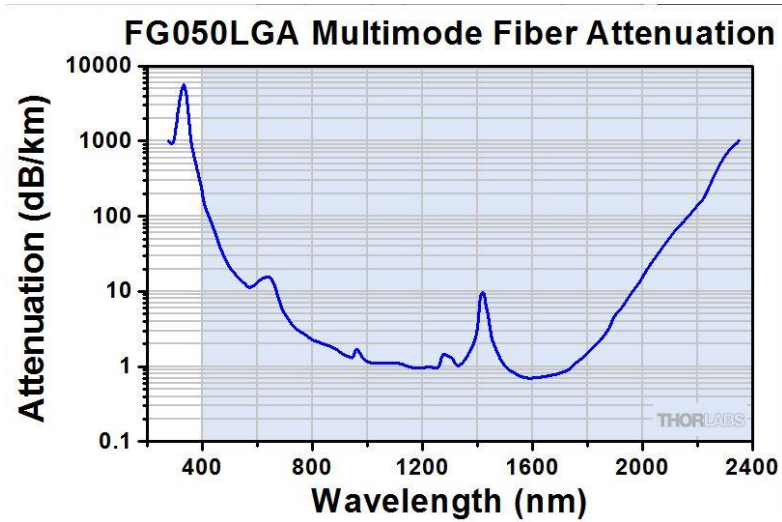


Figura 7. Atenuación fibra óptica en función de la longitud de onda

4.1.2 Fotoemisor. Diodo LED

Como se ha comentado anteriormente, el fabricante *Industrial Fiber Optics, Inc* ofrece una serie de componentes que se adaptan perfectamente a la fibra óptica elegida.

El diodo LED se encuentra encapsulado en una estructura que incluye una lente que dirige los rayos de luz hacia la salida, la cual tiene un conector que permite encajar la fibra óptica perfectamente para que esta no se mueva y haya las mínimas pérdidas posibles. En la Figura 8 podemos observar la estructura interna del LED y en la Figura 9 podemos observar su apariencia física.

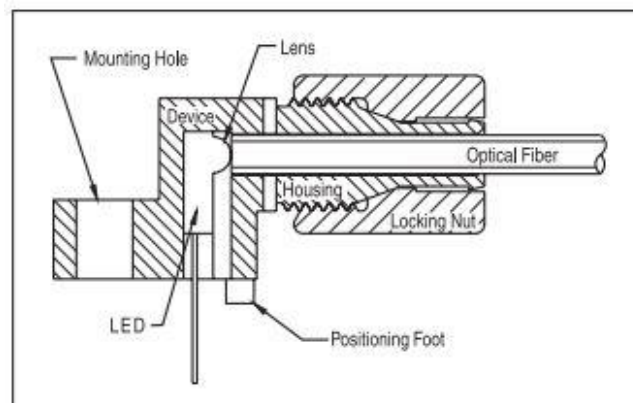


Figura 8. Estructura interna del diodo LED empleado en nuestra aplicación



Figura 9. Diodo LED empleado en nuestra aplicación

Para adaptarse a las condiciones de iluminación de los tintes empleados se requiere un led rojo de 640 nm de longitud de onda, para medir el potencial de acción, y un led verde de 540 nm de longitud de onda, para medir el nivel de calcio intracelular.

En la Tabla 3 se pueden observar sus características.

	VERDE (IF-E93)[25]	ROJO (IF-E96E)[26]
Longitud de onda máxima	522 nm	645 nm
Ancho de banda	40 nm	20 nm

Tabla 3. Características diodos LED necesarios para registrar el potencial de acción (rojo) y el nivel de calcio intracelular (verde)

Como se puede comprobar en la Tabla 3, los diodos empleados se adaptan correctamente a las longitudes de onda que necesitamos para poder excitar los tintes.

4.1.3 Fotodetector.

Como fotodetector se ha elegido un fotodiodo de *Industrial Fiber Optics, Inc* (Figura 10) que, al igual que el diodo LED, viene encapsulado en una estructura (Figura 8) que permite la adaptación perfecta de la fibra al fotodetector evitando, de este modo, que haya pérdidas de luz en el acople. Además, como ocurre en el LED, incluye una lente que dirige los rayos de luz entrante hacia el fotodetector.



Figura 10. Fotodetector empleado en nuestra aplicación

En la Tabla 4 se pueden observar las características principales del fotodetector.

FOTODETECTOR (IF-D91)[27]	
Ancho de banda	450-1050 nm
Responsividad	1,2 $\mu\text{A}/\mu\text{W}$

Tabla 4. Características fotodetector empleado

4.1.4 Filtros

Como hemos comentados anteriormente, antes de que le llegue la señal al fotodetector será necesario realizar un filtrado para eliminar la señal lumínica que proviene del diodo LED y quedarnos, de este modo, únicamente con la señal obtenida del corazón mediante la estimulación de los tintes. Realizaremos, por tanto, un filtrado paso banda para quedarnos con las frecuencias de la banda de trabajo del tinte que estemos empleando. En la Figura 11 podemos observar el filtro que vamos a emplear, de la casa *Thorlabs*, así como la función de transferencia del mismo.

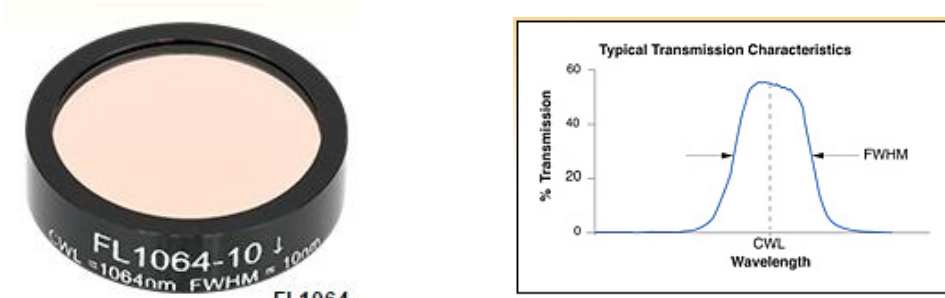


Figura 11. Filtro empleado para eliminar la señal luminosa procedente del led (Izquierda) y su correspondiente función de transferencia (Derecha)

Este filtro irá montado en una pieza (Figura 12) (portafiltros) para evitar que se mueva y se dañe. Esta pieza también está fabricada por *Thorlabs*.



Figura 12. Pieza en la que introduciremos el filtro para que este no se dañe

4.2 Hardware

En la Figura 13 se puede observar el funcionamiento general del sistema diseñado. Se programó un GUIDE de Matlab que permite controlar con el PC tanto el encendido, apagado y nivel de intensidad de los LED como el registro de datos.

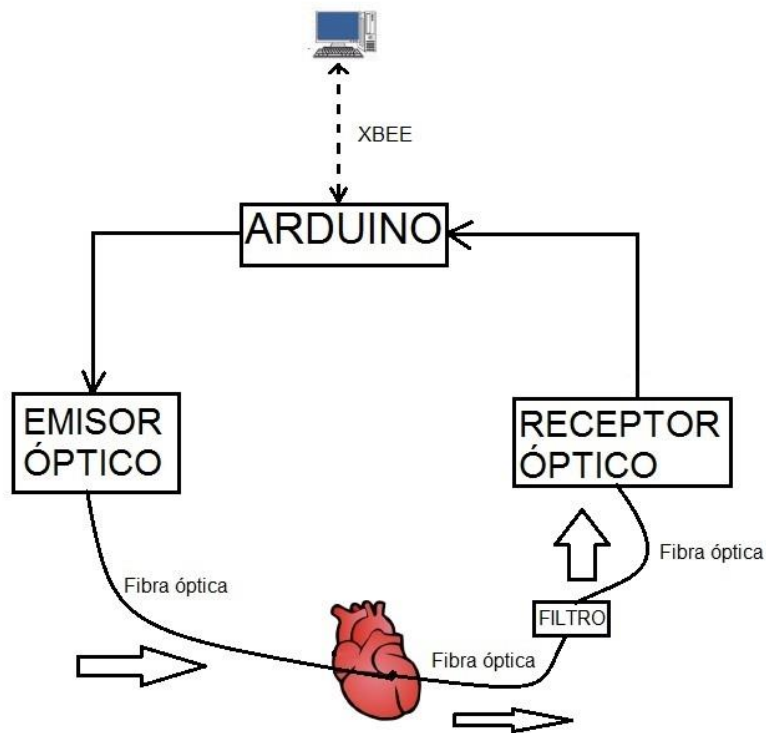


Figura 13. Diagrama general del sistema

Como podemos ver en la Figura 13 el sistema se puede dividir en dos grandes bloques: el emisor y el receptor ópticos. El emisor es el encargado de generar la iluminación, es decir, consta del led necesario para estimular el tinte adherido al corazón. Para transmitir la luz del diodo al corazón emplearemos una fibra óptica. El receptor comienza en una fibra óptica que se dirige hacia el punto que iluminamos para capturar la señal generada por el tinte. Esta señal se filtrará para eliminar la señal que pueda proceder directamente del led de iluminación y se dirigirá al fotodetector. La señal que llega al fotodetector, la acondicionaremos y la registraremos en el ordenador a través de un microcontrolador, Arduino Due en nuestro caso. La comunicación con el ordenador se realizará inalámbricamente empleando unos XBee.

4.2.1 Emisor óptico

El emisor óptico (Figuras 14-15) está formado por los diodos LED, el verde para registrar el nivel de calcio intracelular y el rojo para registrar el potencial de acción.

Mediante el GUIDE de Matlab podremos controlar tanto el encendido y apagado de los LEDs como la intensidad del mismo.

Para poder controlar la intensidad del LED emplearemos los DACs de los que dispone el Arduino empleado. Al LED le añadiremos una resistencia en serie para limitar la corriente que circula a través y evitar, de este modo, su destrucción.

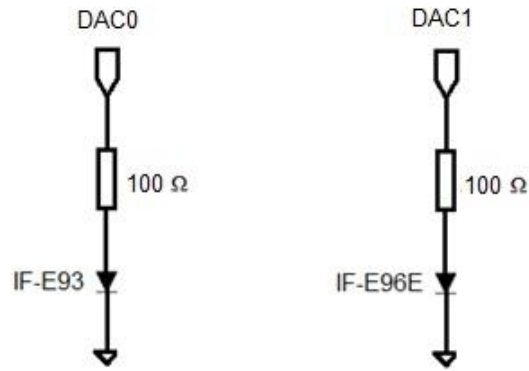


Figura 14. Esquemático circuito electrónico emisor óptico

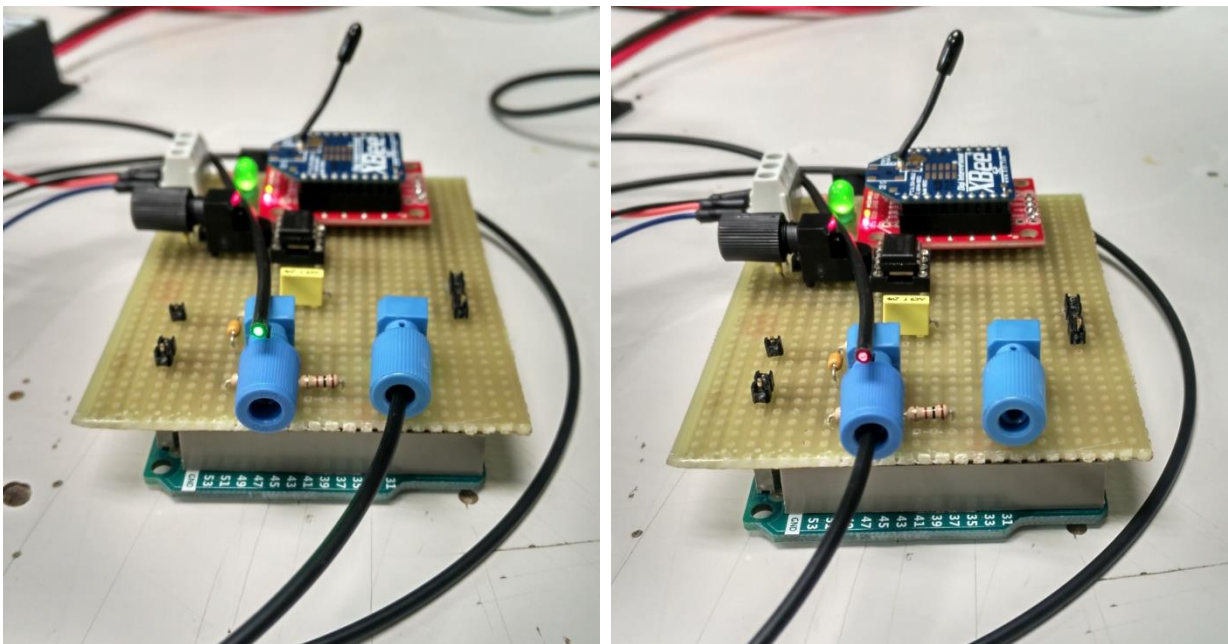


Figura 15. Emisores ópticos en funcionamiento.

4.2.2 Receptor óptico

El circuito del receptor óptico (Figura 16) está formado por el fotodetector y el circuito de acondicionamiento necesario para transformar la corriente proporcionada por éste en tensión que, posteriormente registraremos a través de Arduino y observaremos en el ordenador empleando el GUIDE de Matlab diseñado. Además añadiremos un filtro paso bajo de 1kHz a la salida, encargado de eliminar el ruido de alta frecuencia.

Como acondicionamiento se ha empleado un amplificador de transimpedancia que consta únicamente de un amplificador operacional y una impedancia en la rama de realimentación que proporciona la ganancia necesaria.

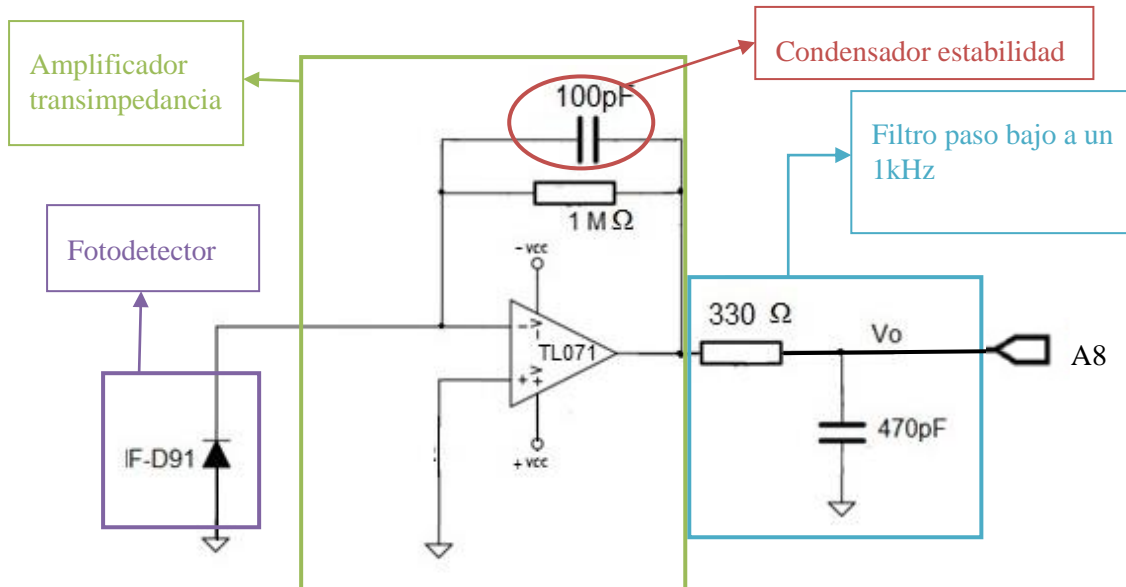


Figura 16. Esquemático circuito electrónico receptor óptico

Para alimentar el amplificador operacional se ha empleado una fuente de alimentación simétrica externa de $\pm 12V$ (Figura 17).



Figura 17. Fuente de alimentación simétrica empleada

En la rama de realimentación se ha añadido un condensador en paralelo con la resistencia para mejorar la estabilidad del circuito, ya que al ver la señal que proporcionaba se podía observar que era inestable. Este condensador de estabilidad también limita el ancho de banda del circuito, por lo tanto, hay que poner una capacidad que establezca el circuito pero que no disminuya el ancho de banda demasiado.

El ancho de banda del circuito se puede calcular según la Ec.1

$$B = \frac{1}{2\pi R_f C_f} \quad (\text{Ec. 1})$$

Siendo R_f y C_f los valores de resistencia y capacidad de realimentación respectivamente. Por lo tanto, para nuestro circuito tenemos un valor de ancho de banda de:

$$B = \frac{1}{2\pi * 1 \cdot 10^6 * 100 \cdot 10^{-12}} = 1591,55 \text{ Hz}$$

4.2.2.1 Amplificador operacional. TL071

En el acondicionamiento del fotodetector se ha empleado el amplificador operacional TL071 (Figura 18).

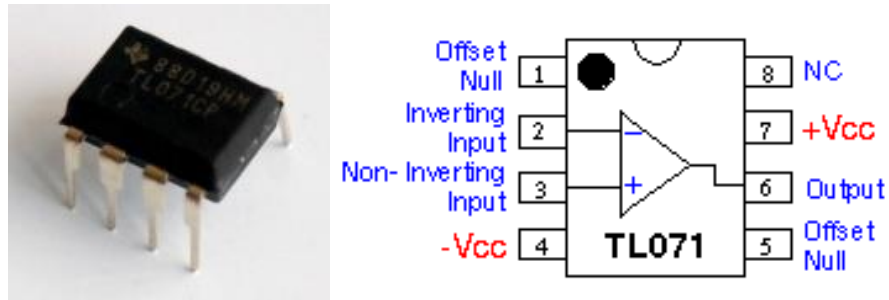


Figura 18. Amplificador operacional TL071 y sus pines

En la Figura 19 podemos observar una comparación de la contribución del ruido que generan distintos tipos de amplificadores operacionales en un amplificador de transimpedancia, en función de la resistencia de realimentación R_f . (para un valor fijo de capacidad de estabilidad). [28]

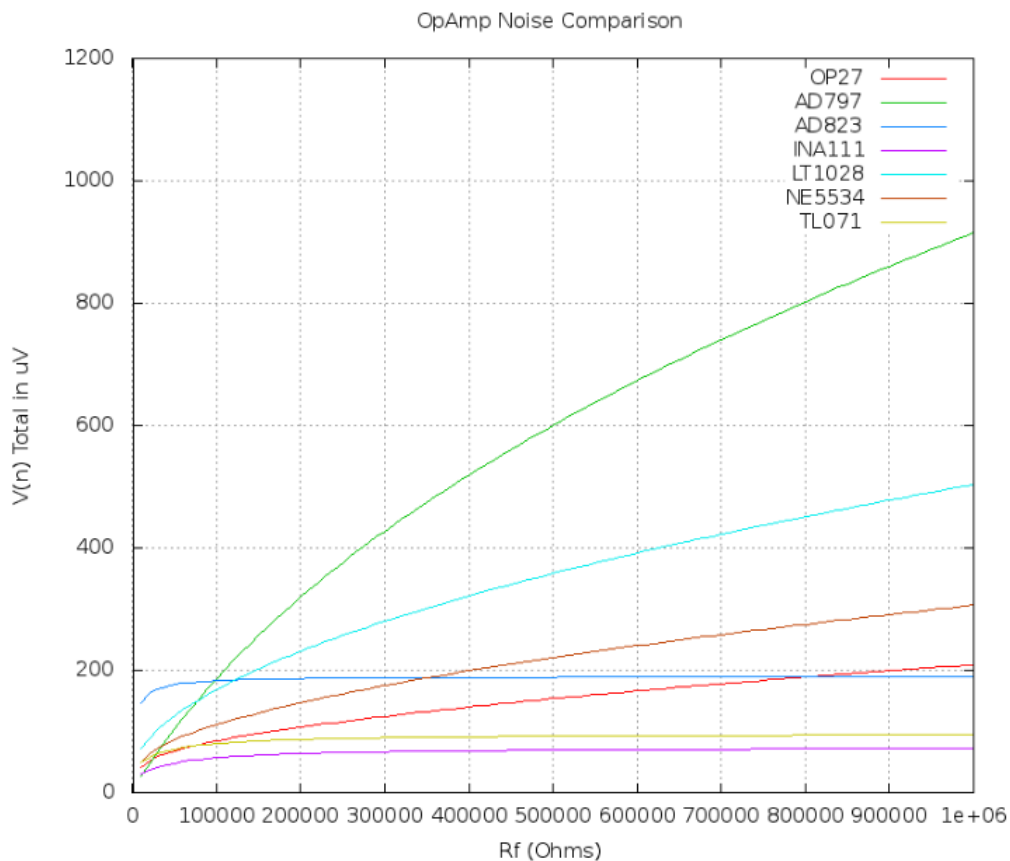


Figura 19. Comparación del ruido en un amplificador de transimpedancia para distintos valores de R_f (resistencia de realimentación) en función del tipo de amplificador operacional empleado

Se puede observar que el TL071 tiene una baja contribución de ruido, por lo tanto, debido a esto y al bajo precio que tiene se ha considerado una buena elección para nuestro sistema.

4.2.3 Estructura de unión fibra óptica - filtro

Para poder adaptar correctamente el filtro a la fibra óptica se realizó un diseño 3D de una cámara en la que introduciremos el filtro y que tendrá unos canales de entrada y salida por la que se introducirán las fibras ópticas.

Esta pieza se ha diseñado empleando el modelador paramétrico *FreeCAD* y ha sido impresa empleando la impresora 3D *Makerbot Replicator 2X*.

FreeCAD es un programa libre que cualquier usuario puede descargar en su ordenador sin ningún tipo de licencia, disponible para Windows, Mac y Linux.

La impresora *Makerbot Replicator 2X* (Figura 20) es un modelo clásico de Makerbot Industries. Es una impresora particularmente estable especializada en la impresión de ABS que integra un doble cabezal. Su plataforma de aluminio está diseñada para mejorar la adherencia del material. Su sistema de inserción de plástico ha evolucionado de manera para facilitar la inserción y el cambio de plástico y evitar los atascos. Además, una puerta y una cubierta de metacrilato permiten estabilizar la temperatura para evitar los problemas de adherencia de la impresión.

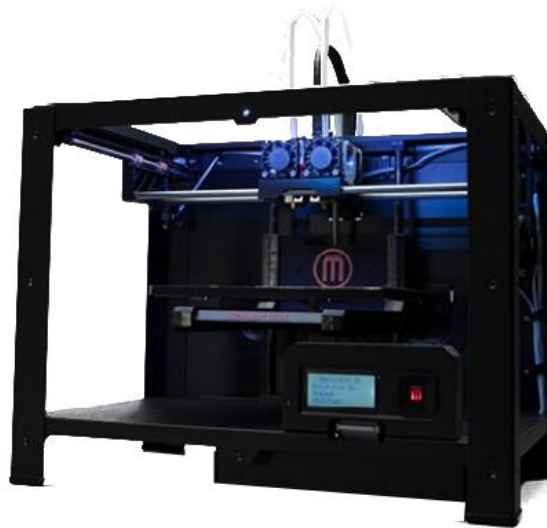


Figura 20. Impresora 3D Makerbot Replicator 2X, empleada para imprimir nuestros diseños 3D

Lo primero que se hizo fue realizar los diseños 3D de la cámara. Se realizaron tres piezas distintas: la cámara en la que va introducido el filtro (Figura 21), un tubo por el que introducir la fibra óptica que, posteriormente, se pegará a la cámara (Figura 22) y la tapa de la propia cámara (Figura 23).

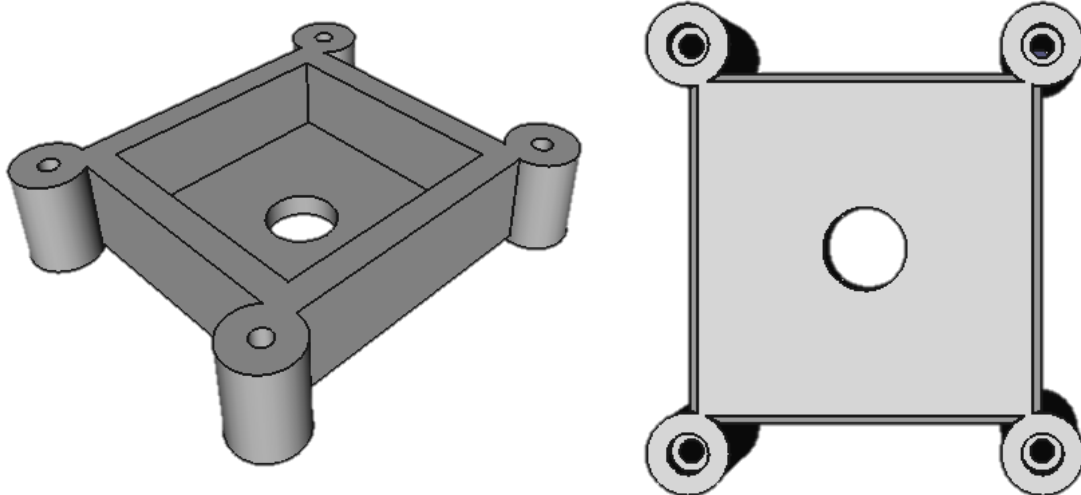


Figura 21. Vistas del diseño 3D de la cámara realizada para acoplar el filtro a la fibra óptica

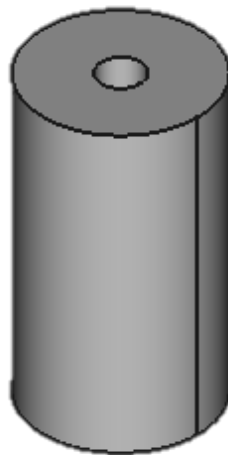


Figura 22. Diseño 3D del tubo por el que se introduce la fibra óptica hacia el filtro

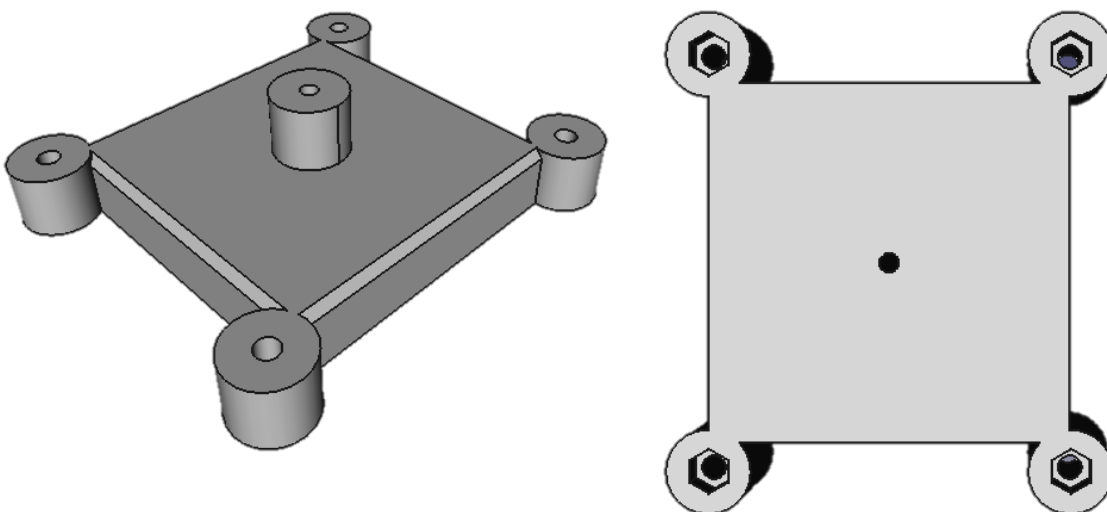


Figura 23. Vistas del diseño 3D de la tapa para la cámara

A continuación, se imprimieron estas piezas en la impresora 3D mencionada, obteniendo las piezas correspondientes (Figuras 24-25).

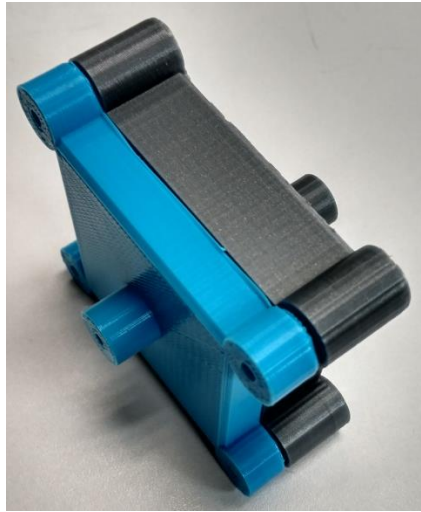


Figura 24. Cámara completa impresa cerrada

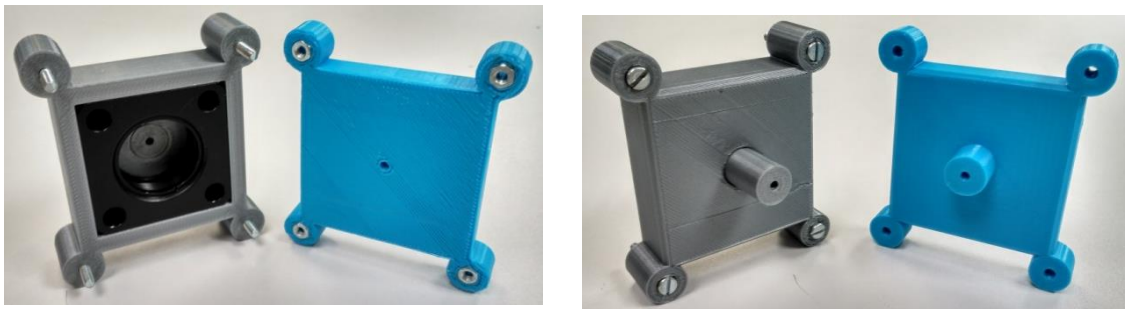


Figura 25. Cámara completa impresa abierta (vistas interior y exterior)

4.2.4 Microcontrolador

Para realizar la adquisición de datos, controlar el encendido y apagado de la iluminación, así como el ajuste de su intensidad y poder establecer una comunicación inalámbrica, se necesita emplear un microcontrolador.

Por lo tanto, de entre todas las posibilidades que ofrece el mercado, hemos decidido emplear *Arduino Due* (Figura 26). Se ha decidido emplear este microcontrolador por las siguientes razones:

- Posee un puerto UART de comunicaciones serie para realizar la comunicación inalámbrica con el PC a través de un XBee.
- Dispone de entrada analógica para registrar la tensión obtenida.
- Dispone de dos DACs para controlar el nivel de intensidad de los LEDs.
- Dispone de la capacidad de procesamiento suficiente para la tarea.

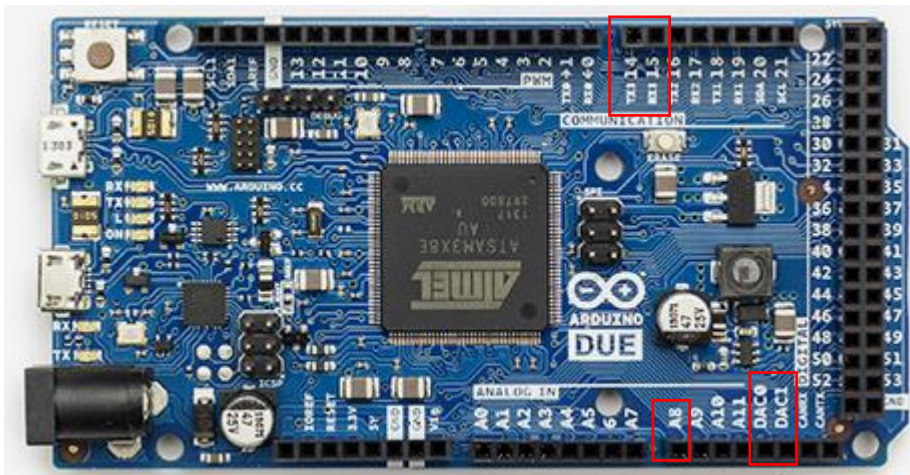


Figura 26. Arduino Due

La principal razón por la que se ha empleado este Arduino es porque incorpora dos DACs, los cuales nos permiten regular el nivel de intensidad de los LEDs sin introducir ruido de alta frecuencia al circuito como ocurre si se emplea la salida digital PWM de la que disponen otros modelos de Arduino.

Arduino Due [29] es un microcontrolador basado en la arquitectura *Atmel SAM3X86E ARM Cortex-M3 CPU*, es la primera placa de Arduino basado en un núcleo ARM de 32-bit. En la Tabla 5 podemos observar sus principales características.

Microcontrolador	AT91SAM3X86E
Voltaje de operación	3.3V
Voltaje de entrada recomendado	7 - 12 V
Voltaje límite de entrada	6-16V
Pines digitales E/S	54 (12 de ellos disponen de salida PWM)
Pines de entrada analógica	12
Pines de salida analógica	2 (DAC)
Corriente de salida total DC en todas las líneas de E/S	130mA
Consumo por pin 3.3V	800mA
Consumo por pin 5V	800mA
Memoria flash	512kB
SRAM	96kB
Velocidad de reloj	84 MHz

Tabla 5. Características Arduino Due

Para poder programar el microcontrolador Arduino Due hay que realizar una serie de pasos:

1. Instalar la última versión del IDE de Arduino (versión 1.6.9).
2. Seguir la ruta: Herramientas - Placa:... - Gestor de tarjetas... y se abrirá la ventana de la Figura 27.

3. Seleccionamos nuestra placa: Arduino SAM Boards (32-bits ARM Cortex-M3) y le damos a Instalar.
4. Tras un tiempo habrá finalizado la instalación.

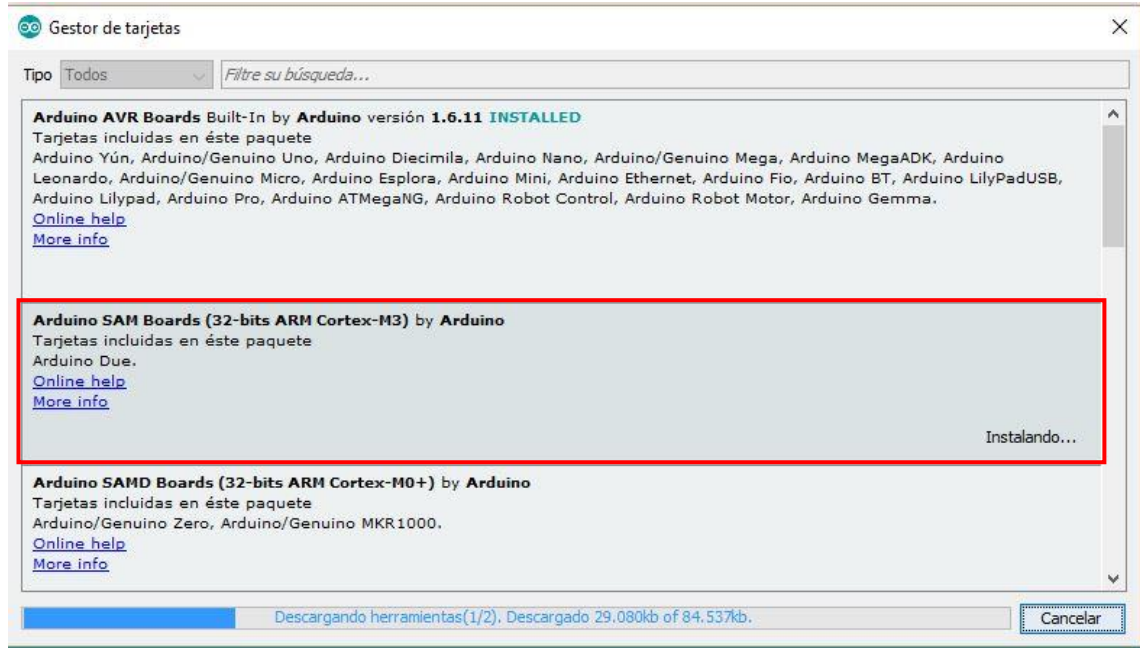


Figura 27. Instalar tarjeta Arduino Due en el IDE

5. Para comprobar que se ha instalado correctamente volvemos al apartado de placas y vemos que se encuentra nuestra tarjeta (Figura 28).

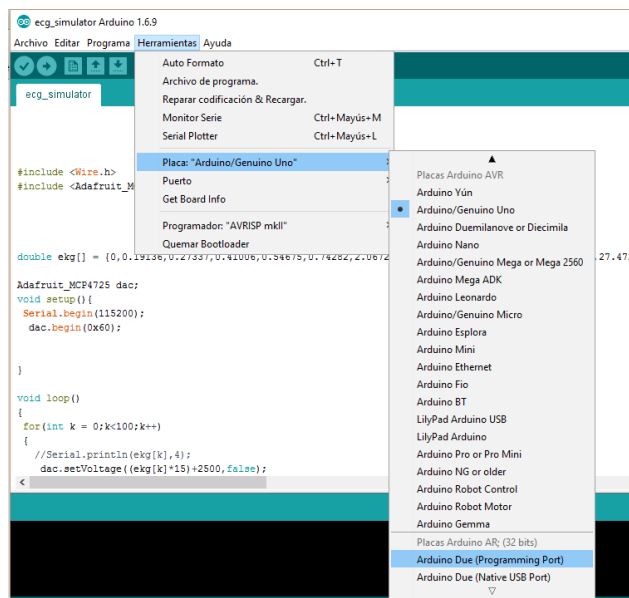


Figura 28. Comprobación correcta instalación tarjeta Arduino Due

4.2.5 Comunicación inalámbrica

Para la comunicación entre nuestro sistema y el ordenador se ha empleado una comunicación inalámbrica basada en el protocolo Zigbee, el cual se emplea fundamentalmente en aplicaciones industriales, médicas y científicas. Sus características principales son su sencillez de uso, su

bajo costo y su bajo consumo eléctrico, en comparación con otros protocolos como, por ejemplo, Bluetooth. Sin embargo, su velocidad es mucho menor que la de otros sistemas de comunicaciones pero para las necesidades de este proyecto son suficientes.

El módulo que emplearemos es el módulo XBee Serie 2 (Figura 29) [30], basado en el protocolo ZigBee. Estos módulos permiten crear redes de malla complejas basadas en el firmware XBee ZB ZigBee. Permiten una comunicación muy fiable y simple entre microcontroladores, ordenadores, sistemas..., cualquier dispositivo con un puerto serie. Soporta redes punto a punto y multipunto. En la Tabla 6 podemos observar sus características principales.

3.3V a 40mV
Velocidad máxima: 200 kbps
6 pines de entrada ADC de 10bit
8 pines entrada/salida digitales
Comandos AT o API

Tabla 6. Características Xbee Serie 2

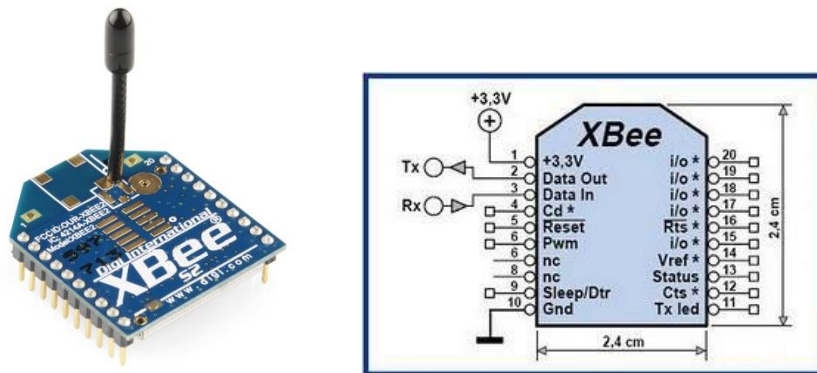


Figura 29 . Módulo XBee

La red estará formada por dos módulos XBee: un maestro y un esclavo.

El XBee maestro se conecta al PC a través de un cable USB creando un puerto serie virtual entre el ordenador y el XBee. Por otra parte, el XBee esclavo va conectado a nuestro sistema utilizando una comunicación tipo UART con el microcontrolador (Arduino).

Estas comunicaciones se controlarán a través de GUIDE de Matlab permitiendo establecer tanto la conexión como la desconexión entre ambos dispositivos.

Cada módulo XBee tendrá que ser programado para poder realizar su papel dentro de la red de comunicación. Cada módulo llevará una programación diferente que podrá realizarse de dos formas distintas. O bien empleando comandos AT, o bien utilizando el programa X-CTU que permite programar cada módulo XBee. En este trabajo se ha decidido emplear X-CTU por su sencillez.

Para más información acerca de cómo configurar el X-CTU véase Anexo II.

El XBee conectado a nuestro sistema (esclavo) irá sobre un adaptador "Xbee Explorer Regulated" de modo que únicamente serán necesarios conectar 4 pines (GND, 5V, RX y TX). El XBee conectado al ordenador (maestro) irá sobre un adaptador "Xbee Explorer USB" de modo que únicamente será necesario conectarlo al ordenador mediante un cable USB. En la Figura 30 podemos observar un esquemático del conexionado.

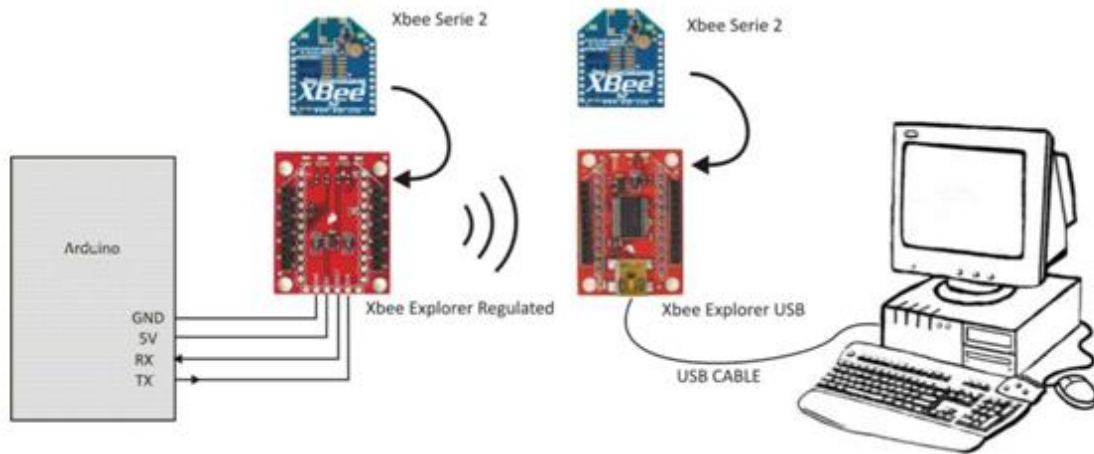


Figura 30. Esquema conexión Xbee al arduino y al PC.

Para la conexión entre el Xbee y el Arduino se emplearán los puertos digitales 14 y 15 del microcontrolador (puerto serie 3), siendo el puerto 14 el de transmisión (TX3) y el puerto 15 el de recepción (RX3).

Para comunicarse con el microcontrolador el módulo Xbee utiliza un puerto serie UART a través de conexión por cable el cual está compuesto por una línea de transmisión (pin Tx) y otra de recepción (pin Rx).

4.2.6 Caja para instrumental de laboratorio

Por último se diseñó una caja, nuevamente empleando *FreeCAD* y la impresora *3D Makerbot Replicator 2X*, en la que introducir todo nuestro sistema, es decir, el Arduino junto con la placa perforada y la fuente de alimentación, para transportar el sistema más fácilmente y para evitar que éste se dañe si se moja. Al cable de la red eléctrica además se le añadió un fusible para intentar evitar dañar el sistema.

En la Figura 31 podemos observar el diseño 3D realizado, tanto de la caja como de la tapa de la misma. Se puede observar que a la tapa se le han realizado una serie de ranuras a modo de ventilación y a la caja unos agujeros en los que irán las tres fibras ópticas, que se conectarán a los LEDs y al fotodetector, y los cables de alimentación de red. En la Figura 32 podemos observar el resultado obtenido.

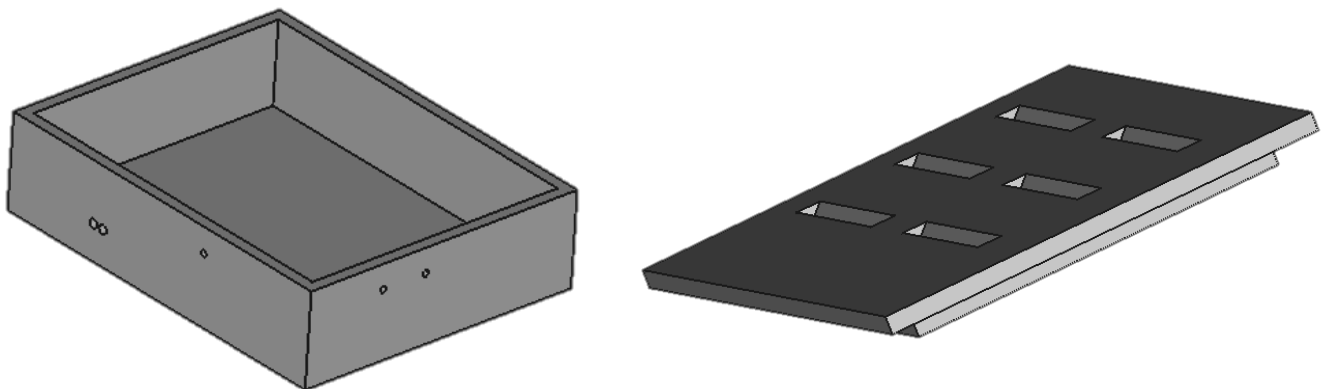


Figura 31. Caja diseñada para instrumental de laboratorio

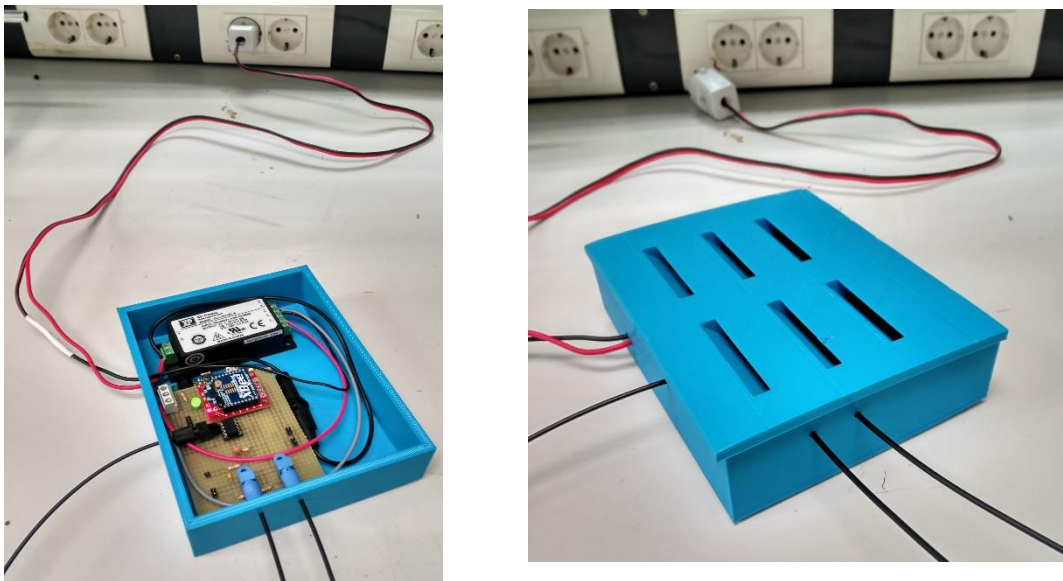


Figura 32. Sistema en caja para instrumental de laboratorio impresa

4.3 Software

4.3.1 Firmware de Arduino

Se ha desarrollado un firmware de Arduino cuya principal misión es responder a las instrucciones que le llegan, a través de la comunicación inalámbrica, del PC.

El firmware está formado por una serie de funciones:

- **setup()**: En esta función establecemos los pines digitales de los LEDs como pines de salida y los inicializamos; inicializamos la comunicación serie con el XBee (Serial3), estableciendo la velocidad de la misma; inicializamos la interrupción Timer4 empleando la función **registro()** y establecemos la frecuencia a la que queremos que se produzca la interrupción (frecuencia de muestreo). Emplearemos una frecuencia de muestreo de 500Hz. Se ha establecido una velocidad de comunicación de 115200 baudios ya que necesitamos mandar datos a una velocidad elevada. Para poder emplear la interrupción será necesario importar la librería "DueTimer".
- **registro()**: En esta función indicamos lo que queremos hacer cada vez que se genere la interrupción. En nuestro caso vamos a leer la salida analógica 8, es decir, la que se encuentra conectada a la salida de nuestro sistema. Además, enviaremos estos datos registrados por el puerto serie hacia el XBee para que sean registrados en el ordenador.
- **loop()**: Mediante esta función estaremos leyendo continuamente lo que recibimos a través del puerto serie proveniente del ordenador. En función del carácter que le llegue realizará una de las siguientes acciones:
 - 'c' : Mandará una 'd' a través del XBee indicando que la conexión se ha establecido correctamente.
 - '1' : Comienza el registro de datos. Inicializamos el temporizador.
 - '2' : Encendemos el LED verde a su máxima intensidad, es decir, le introducimos al DAC un valor de 4095.
 - '3' : Encendemos el LED rojo a su máxima intensidad, es decir, le introducimos al DAC un valor de 4095.
 - '4' : Subimos la intensidad del LED encendido en ese momento. En el caso de que ya no se pueda subir más la intensidad (el DAC ha llegado al valor máximo de 4095), estableceremos el valor fijo de intensidad en el DAC de 4095.

- '5' : Bajamos la intensidad del LED encendido en ese momento. En el caso de que ya no se pueda bajar más la intensidad (el DAC ha llegado al valor mínimo de 0), estableceremos el valor fijo de intensidad en el DAC de 0.
- '6' : Apagamos el LED verde.
- '7' : Apagamos el LED rojo.
- '8' : Detenemos la adquisición de datos finalizando el temporizador.

Para más información acerca del código programado en el Arduino Due véase Anexo III.

Como se ha comentado emplearemos la librería *DueTimer*, la cual no se encuentra incluida por defecto en el IDE de Arduino y será necesario añadirla.

Para más información acerca de cómo importar la librería véase Anexo I.

4.3.2 Interfaz gráfica de usuario

Como se ha comentado anteriormente, para poder controlar el sistema se ha programado un GUIDE de Matlab (Figura 33).

Este programa consta de dos pantallas: una de adquisición de datos y otra de representación de medidas almacenadas.

Para más información acerca del código programado véase el Anexo IV.1 para ver el código de la pantalla de registro de datos y el Anexo IV.2 para ver el código de la pantalla que permite representar una medida almacenada.



Figura 33. Menú principal interfaz de usuario

4.3.2.1 Adquisición de datos

La pantalla de adquisición de datos (Figura 34) nos permite controlar todas las variables que se ven involucradas cuando realizamos un registro.



Como se puede observar esta pantalla está formada por una serie de bloques:

1. **Conexión** (Figura 35): Nos permite establecer la conexión inalámbrica entre el ordenador y nuestro sistema a través de los módulos XBee. Para ello deberemos indicar el puerto en el que el XBee conectado a nuestro ordenador (coordinador) se encuentra. En esta sección además deberemos indicar el número de muestras que queremos registrar. Cuando la conexión se haya producido aparecerá en el cuadro de texto la palabra "Conectado".



Figura 35. Pantalla adquisición de datos. Bloque conexión.

2. **Registro de datos** (Figura 36): Está formado por un botón que, al ser pulsado, inicializa el registro de datos. Por lo tanto, se manda el carácter correspondiente para que el

Figura 34. Pantalla registro de datos

Arduino responda inicializando el temporizador. Una vez se haya registrado el número de muestras establecido por el usuario se envía al Arduino la instrucción correspondiente para que finalice el temporizador y no siga adquiriendo datos, además, una vez finalizado el registro, aparece en la gráfica la señal obtenida.



Figura 36. Pantalla adquisición de datos. Bloque registro de datos

- Control LEDs** (Figura 37): En esta sección tenemos dos botones que controlan el encendido y apagado de los LEDs. Cuando se pulse el botón verde, se mandará la instrucción correspondiente para que el Arduino encienda el LED verde, el cual se empleará para realizar un registro de calcio intracelular. Cuando desactivemos este botón, el LED verde se apagará. Cuando pulsemos el botón rojo ocurrirá lo mismo pero, en este caso, se encenderá el LED rojo encargado de estimular el tinte potenciométrico que permite registrar el potencial de acción. Igualmente, cuando lo desactivemos este LED se apagará. Además tenemos dos botones más: "+" y "-" encargados de mandar las instrucciones correspondientes para que el Arduino suba y baje respectivamente la intensidad del LED que se encuentre encendido en ese momento. En todo momento tenemos un indicador que nos indica que LED se encuentra encendido o si ambos están apagados.

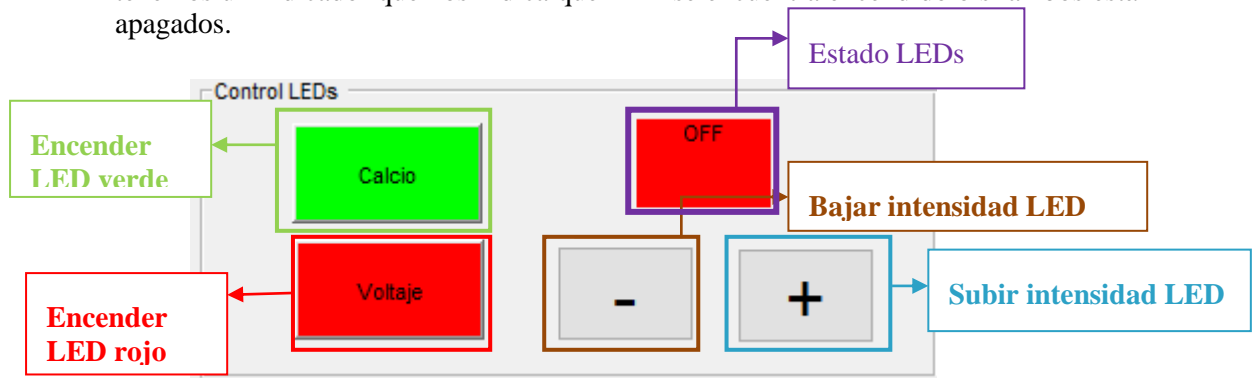
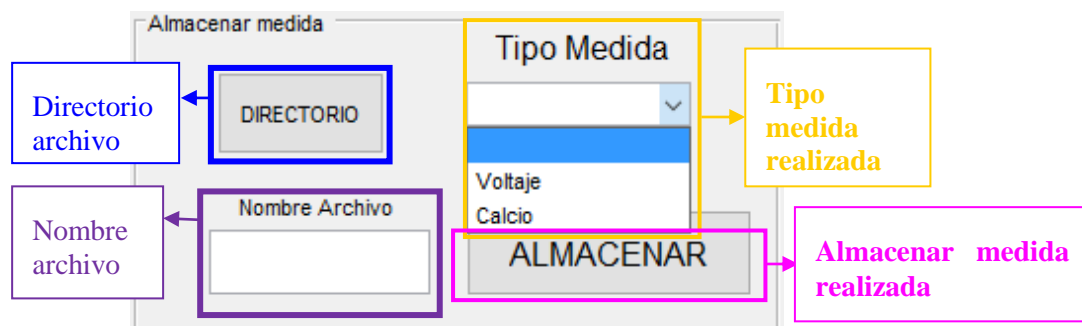


Figura 37. Pantalla adquisición de datos. Bloque control LEDs.

- Almacenar medida** (Figura 38): Esta sección nos permite guardar una medida que hayamos realizado. Para ello primero deberemos seleccionar el directorio del archivo en el que queremos guardar los datos pulsando "DIRECTORIO" e introducir el nombre del mismo en la casilla correspondiente. A continuación deberemos seleccionar el tipo de medida realizada en el menú desplegable y pulsar el botón "ALMACENAR". Esta medida se guardará en un archivo Excel en que tenemos una base de datos con todas las medidas que se han ido guardando a lo largo del tiempo. Estas se encuentran clasificadas por fecha, hora y tipo de medida del experimento realizado.



5. **Des:** **Figura 38. Pantalla adquisición de datos. Bloque almacenar medida** to empleado además de mandar al Arduino la instrucción necesaria para apagar los LEDs si resulta que alguno se encuentra encendido.
6. **Menú:** volvemos al menú principal.

4.3.2.2 Representación de medidas

La pantalla de representación de medidas almacenadas nos permite representar una adquisición de datos realizada previamente y almacenada en la base de datos de Excel. Para poder representar la medida que queremos tenemos varias opciones:

- 1.
2. **Indicar la fecha** (Figura 39): si únicamente queremos filtrar la medida que queremos representar en función de su fecha de realización, introduciremos el día, mes y año y en tipo de medida pondremos "Todos". En el "listbox" de la derecha aparecerán todas las medidas guardadas para la fecha introducida separadas en dos bloques: VOLTAJE y CALCIO. Además también aparece la hora a la que estas medidas fueron realizadas para poder identificarlas.

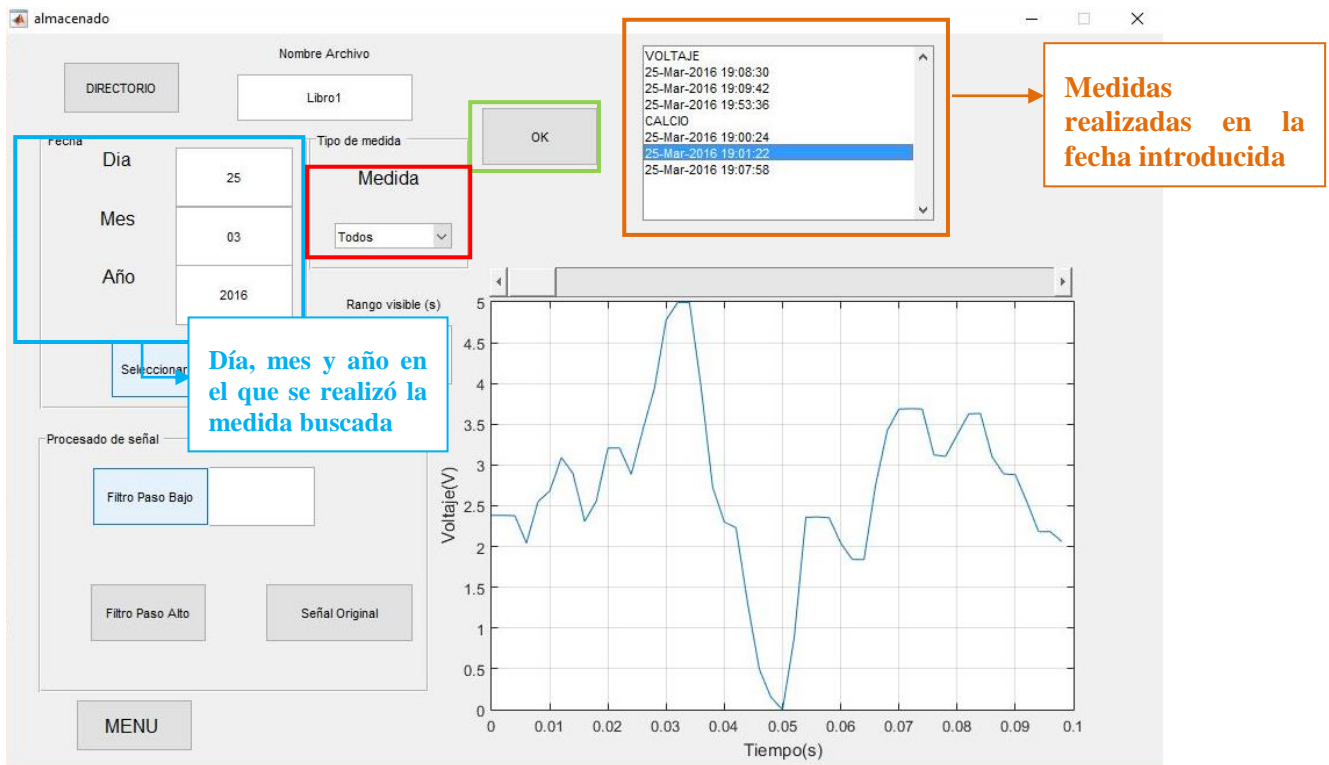


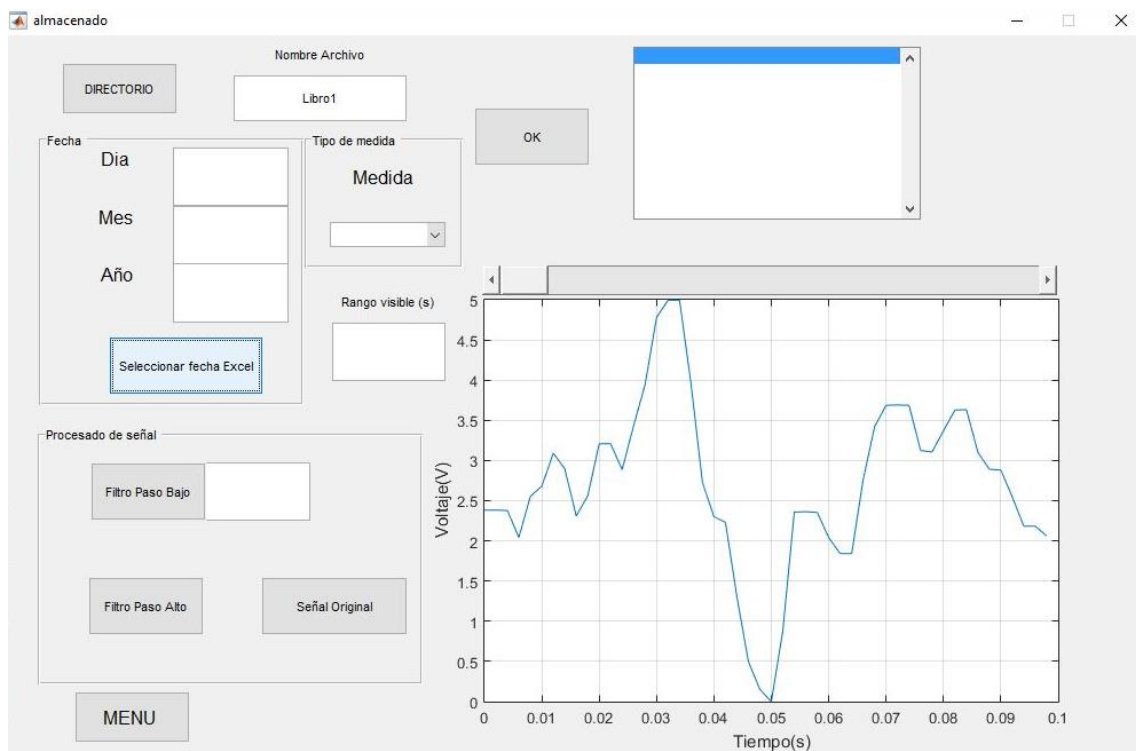
Figura 39. Representar medida almacenada. Filtrado únicamente por fecha

3. **Indicar la fecha y tipo de medida** (Figura 40): aparte de indicar el día, mes y año también podemos indicar el tipo de medida realizada (Voltaje para medida de potencial de acción o Calcio para medida de calcio intracelular). En el "listbox" de la derecha aparecerán las medidas que cumplan estas especificaciones con la hora a la que fueron realizadas.



Figura 40. Representar medida almacenada. Filtrado por fecha y tipo de medida.

3. **Seleccionar fecha en el Excel** (Figura 41): Podemos seleccionar directamente la fecha en la base de datos pulsando "Seleccionar fecha", al pulsar se abrirá el archivo Excel y una ventana emergente. En el Excel seleccionaremos la fila de la medida que queremos representar y, una vez seleccionada, pulsaremos "Ok" en la ventana emergente.



	A	B	C	D	E	F	G	H	I
1	FECHA	MEDIDA	VOLTAJE						
2	15-mar-2016 17:18:46	voltaje	1	2	3	4	5		
3	16-mar-2016 17:18:46	calcio	2	5	8	6	0		
4	17-mar-2016 17:18:46	voltaje	7	8	9	0	5		
5	18-mar-2016 17:18:46	voltaje	7	8	5	3	6		
6	19-mar-2016 17:18:46	calcio	4	6	8	3	8		
7	15/03/2016 21:19	calcio	5	8	2	6	8		
8	20/03/2016 11:42	calcio	2	4	5	6	7		
9	24/03/2016 20:06	calcio	0,94726563	0,94726563	0,94726563	1,63574219	1,72851563	1,11328125	0,94238281
10	24/03/2016 20:22	calcio	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906
11	24/03/2016 20:23	calcio	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906	1,50878906
12	25/03/2016 19:00	calcio	2,14355469	2,14355469	2,14355469	2,14355469	2,14355469	1,42089844	0,87890625
13	25/03/2016 19:01	calcio	2,3828125	2,3828125	2,37792969	2,04101563	2,5371094	2,68066406	3,09082031
14	25/03/2016 19:07	calcio	1,81640625	1,81640625	2,25585938	2,5390625	2,52441406	1,78710938	1,45019531
15	25/03/2016 19:08	voltaje	1,81640625	1,81640625	2,25585938	2,5390625	2,52441406	1,78710938	1,45019531
16	25/03/2016 19:09	voltaje	1,84082031	1,8359375	1,60644531	1,3671875	2,56835938	2,40234375	1,953125
17	25/03/2016 19:53	voltaje	3,28125	3,28125	3,28125	3,28125	3,28125	3,27636719	3,28125
18	12/05/2016 18:37	voltaje	2,47558594	2,50976563	2,50488281	2,49511719	2,50488281	1,91894531	4,44335938
19	13/05/2016 19:52	calcio	2,49511719	2,49511719	2,49023438	2,49511719	2,49511719	2,49023438	2,49511719
20	13/05/2016 19:54	voltaje	2,49023438	2,49511719	2,49511719	2,49511719	2,49023438	2,49511719	2,49023438
21	16/05/2016 15:21	voltaje	2,68554688	2,68554688	2,68554688	2,68554688	2,68554688	2,68554688	2,68554688
22									
23									

Data Selection Dia...

Select data region in Excel worksheet.
Click OK to continue in MATLAB

OK

Figura 41. Representar medida almacenada. Selección medida en Excel

Para todas ellas previamente habrá que seleccionar el directorio en el que se encuentra el Excel con los datos que queremos representar. Para ello habrá que pulsar "DIRECTORIO" y seleccionar la carpeta deseada, posteriormente se introducirá el nombre del archivo Excel en la casilla "Nombre Archivo".

Además se ha añadido una sección de tratamiento de señal (Figura 42) por si fuera necesario. Se ha añadido un filtro paso alto, encargado de eliminar la línea basal, eliminando de este modo el ruido de continua que podemos tener; y un filtro paso bajo Butterworth cuya frecuencia de corte podemos introducir, encargado de eliminar el ruido de alta frecuencia que pueda distorsionar la señal registrada.

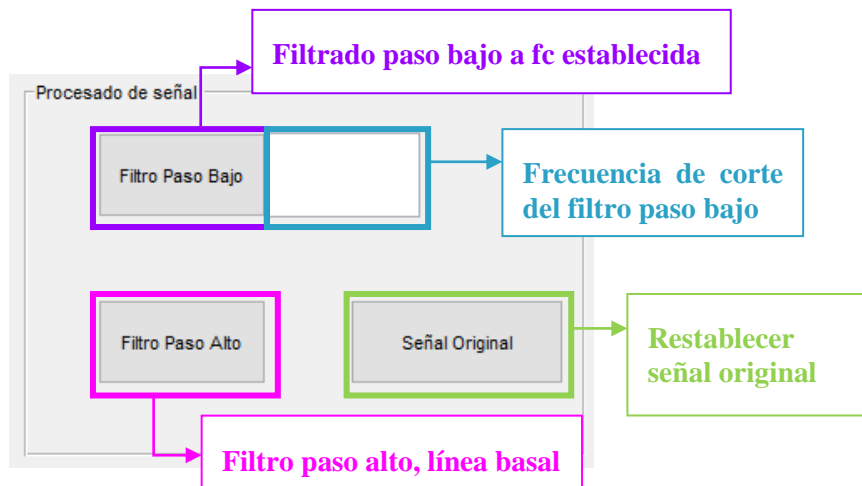


Figura 42. Bloque procesado de la señal medida almacenada

Dado un registro puede ocurrir que únicamente quiera verse un tramo de este ya sea porque el registro tiene un elevado número de muestras y no se vea bien o porque deseemos analizar cierto periodo de tiempo. Para estos casos se ha añadido la casilla "Rango visible (s)" en el cual el usuario puede introducir el margen de tiempo que desee analizar (en segundos). Posteriormente, empleando el "slider" se podrá navegar por la señal en el tiempo.

En Figura 43 podemos observar un registro con un elevado número de muestras. En la Figura 44 podemos observar el resultado que obtenemos si introducimos un rango visible de 1s.

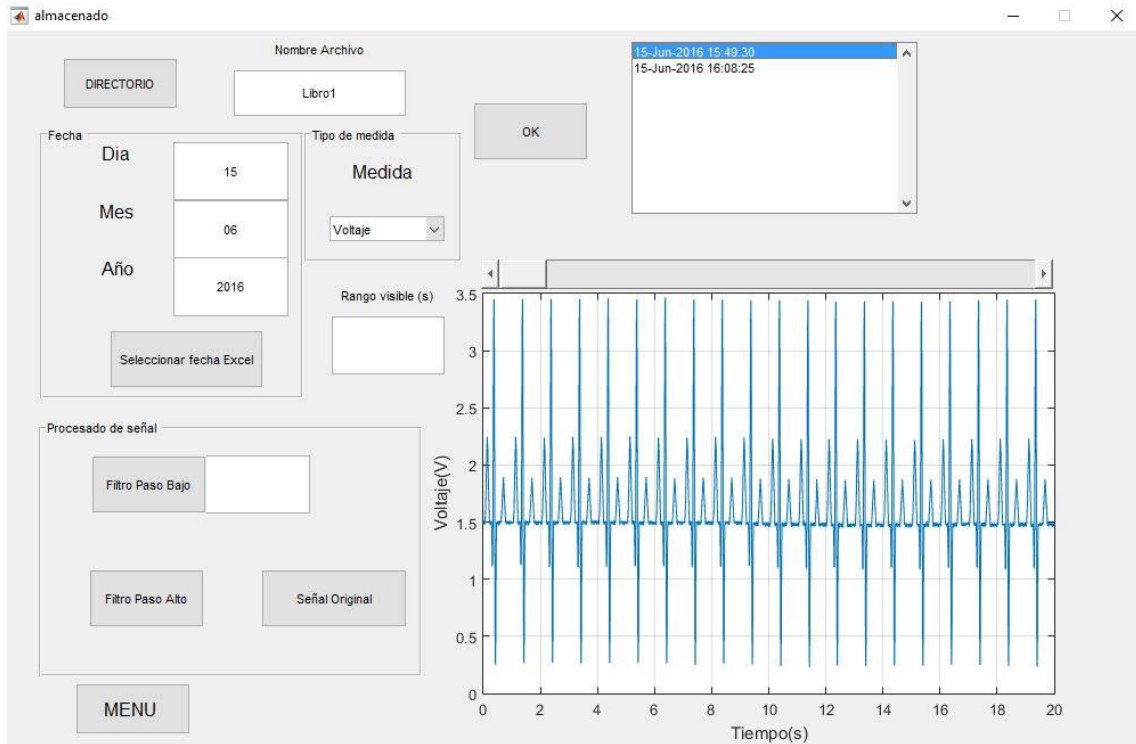


Figura 43. Registro con muchas muestras. Mala visualización



Figura 44. Registro con muchas muestras. Rango visible menor

4.4 Montaje circuito sobre placa perforada

Para tener un sistema más compacto y limpio y evitar, de este modo, la desconexión de algún cable se decidió implementar el sistema en una placa de pruebas perforada.

Esta placa se ha diseñado de forma que se pueda anclar sobre el Arduino y, así, poder transportar el sistema de una forma más compacta.

En la placa se ha incluido el circuito emisor, el receptor y el módulo XBee esclavo.

En la Figura 45 podemos observar la parte delantera y trasera de la placa diseñada respectivamente.

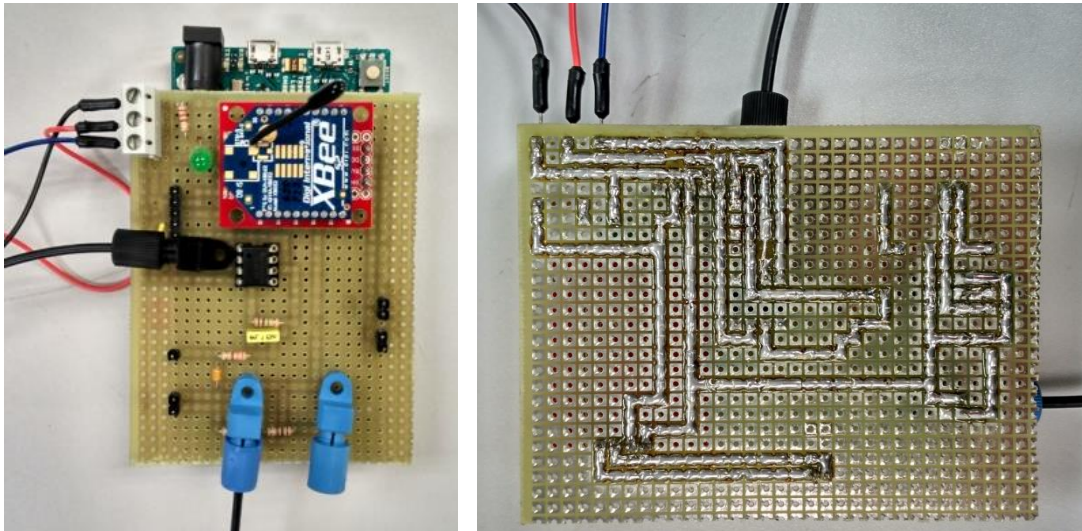


Figura 45. Parte delantera (Izquierda) y trasera (Derecha) del circuito soldado sobre una placa perforada

4.5 Resultados

En este apartado vamos a analizar los distintos resultados obtenidos a la hora de caracterizar nuestro sistema. En primer lugar vamos a analizar el nivel de tensión obtenido a la salida de nuestro sistema en función del voltaje aplicado a la entrada del LED de iluminación, tanto del LED verde como del LED rojo.

A continuación aplicaremos una señal variable en el tiempo a la entrada del LED, esta misma señal la registraremos en el fotodetector llevando una fibra óptica desde el LED en el que aplicamos la señal hasta el fotodetector, podremos observar que el receptor responde adecuadamente ante una señal variable en el tiempo. Además comprobaremos que nuestro sistema de adquisición de datos a través de Arduino y del GUIDE de Matlab funciona correctamente registrando esta señal a través del PC.

Por último analizaremos la función de transferencia de nuestro sistema aplicando señales senoidales a distintas frecuencias a la entrada del diodo LED y viendo la señal obtenida en el fotodetector ante estas señales.

4.5.1 Nivel de tensión a la salida en función del voltaje en el LED de iluminación

En primer lugar vamos a caracterizar el sistema. Para ello vamos a ver la respuesta que obtenemos a la salida del amplificador (incluyendo el filtro) ante diferentes valores de tensión de entrada al diodo led (tensión aplicada a la entrada de la resistencia de protección del diodo).

En la Figura 46 podemos observar la respuesta obtenida iluminando el diodo verde y en la Figura 47 podemos observar la obtenida iluminando el diodo rojo.

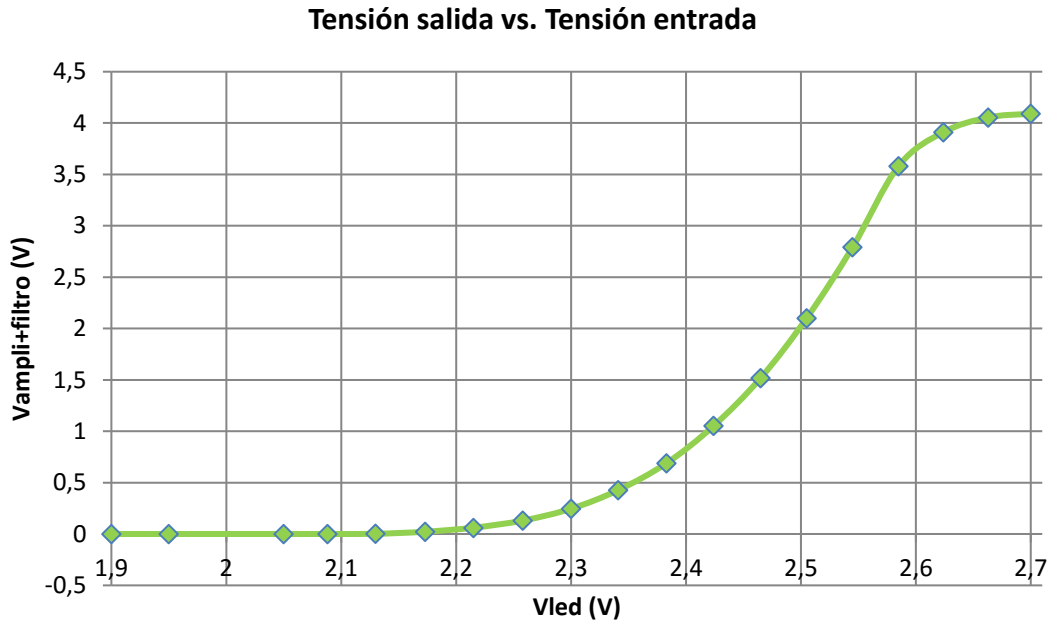


Figura 46. Tensión a la salida del sistema (fotodetector + amplificador + filtro) para distintos valores de tensión de entrada al led verde

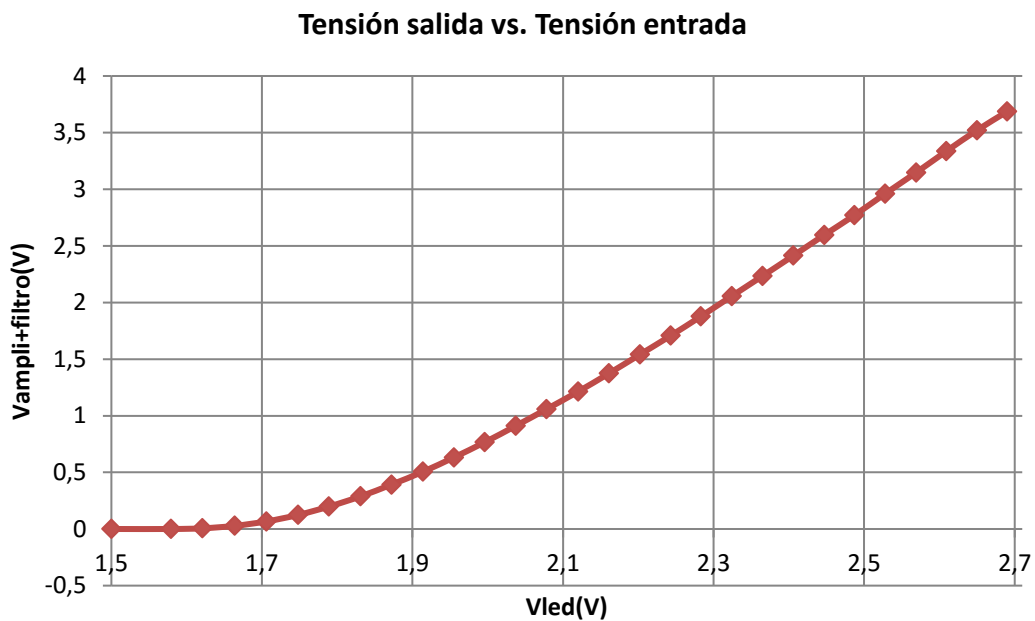


Figura 47. Tensión a la salida del sistema (fotodetector + amplificador + filtro) para distintos valores de tensión de entrada al led rojo

En el caso del diodo verde no obtenemos señal para tensiones inferiores a 2V, mientras que en el caso del diodo rojo no obtenemos señal por debajo de 1,65 V. Por lo tanto, deberemos iluminar los LEDs con tensiones superiores a estos valores para que estos luzcan.

4.5.2 Señal a la salida en función de la señal generada en el diodo LED.

A continuación, para comprobar la correcta respuesta de nuestro sistema vamos a generar una señal analógica con el Arduino Due que le introduciremos a uno de los diodos LED a través de la salida del DAC del Arduino. Vamos a generar una señal que simule la de un ECG. En la Figura 48 podemos observar la señal generada (azul) y la obtenida en el fotodetector (amarilla), en este ejemplo en particular se empleó el diodo rojo para iluminar el fotodetector. Además de obtener la misma respuesta morfológicamente, podemos observar que esta se encuentra amplificada.

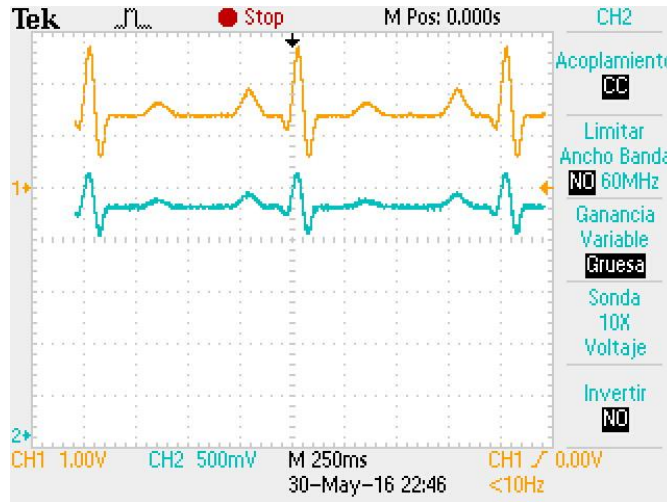


Figura 48. Respuesta Fotodetector (amarillo) ante una señal generada en el diodo (azul).

Para comprobar que nuestro sistema de adquisición en Matlab (GUIDE) funciona correctamente hemos realizado esta misma adquisición registrando los valores en el interfaz (Figura 49) en lugar de en el osciloscopio.

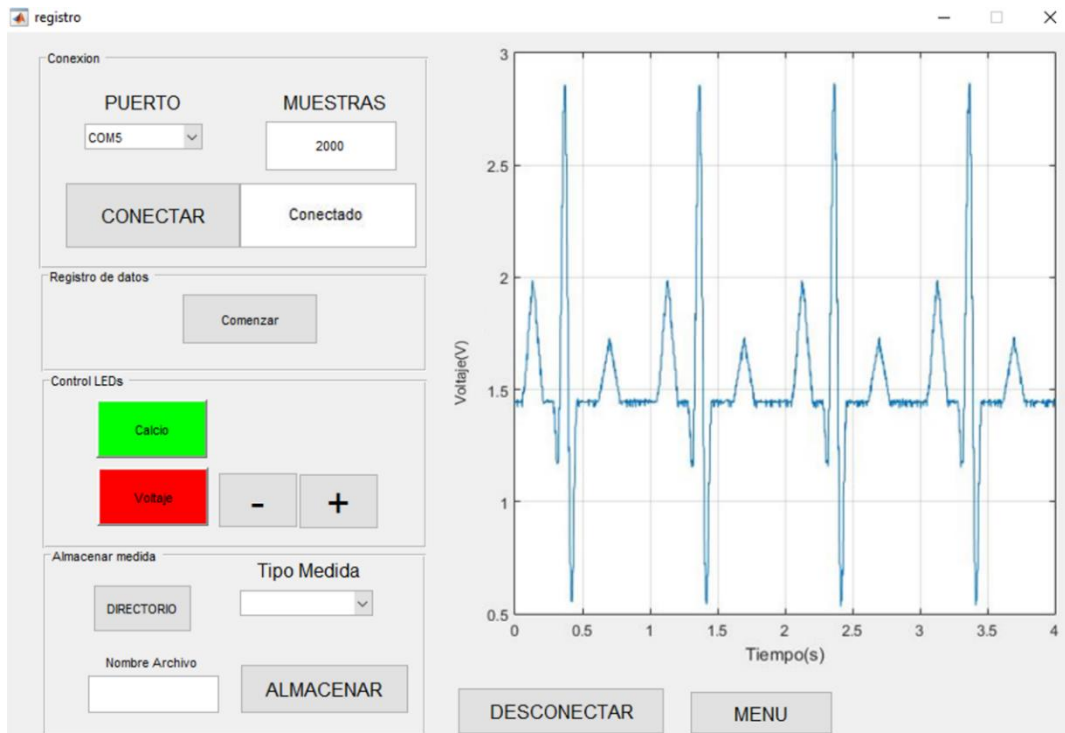


Figura 49. Respuesta obtenida en el fotodetector registrándola con el interfaz de Matlab

4.5.3 Función de transferencia

Para calcular la función de transferencia de nuestro sistema hemos excitado el diodo LED con señales senoidales, empleando el generador de funciones, de distintas frecuencias.

Hemos generado una señal de 220mVpp con un offset DC de 1V. Hemos ido variando la frecuencia desde los 100 Hz hasta los 5kHz obteniendo la función de transferencia de la Figura 50.

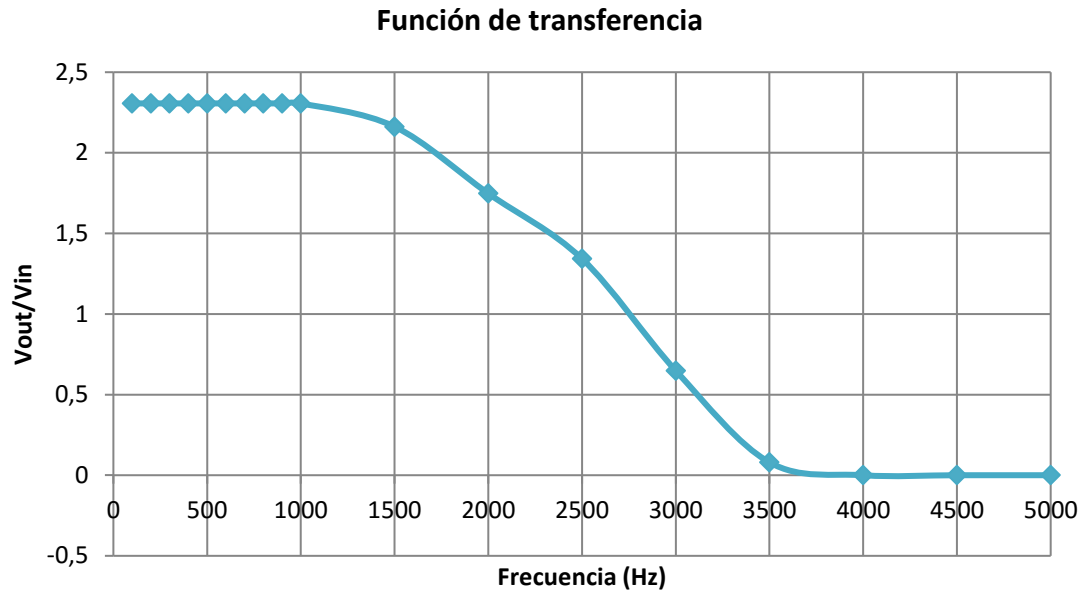


Figura 50. Función de transferencia

Se puede comprobar que la señal empieza a recortar a partir de 1kHz. Esto era de esperar ya que tenemos un filtro paso bajo a esta frecuencia a la salida de nuestro sistema. La caída obtenida no es muy abrupta, esto es debido a que el filtro empleado es de primer orden.

Capítulo 5. Pliego de condiciones

5.1 Condiciones de nuestro sistema

- Rango dinámico de digitalización entre 0 y 5 V, ampliables a $\pm 12V$ mediante la incorporación de un adaptador Hardware.
- Tensión de funcionamiento mínima de 2V para el LED verde y de 1,65V para el LED rojo. Por debajo de estos valores no lucen.
- Registro de una única señal a escoger entre potencial de acción o nivel de calcio intracelular.
- Frecuencia de muestreo máxima de 500Hz mediante comunicación inalámbrica (XBee), ampliables realizando la comunicación con cable.

5.2 Presupuesto

5.2.1 Humano

HORAS DEDICADAS	PRECIO/HORA
350	23,7€
TOTAL	8.295 €

Tabla 7. Presupuesto horas humanas

Se ha empleado un precio aproximado al que se paga en la actualidad.

5.2.2 Material

CANTIDAD	REFERENCIA	NOMBRE	€/UNIDAD	TOTAL	PROVEEDOR
1	IF-D91	Fotodetector	3,68€	3,68€	Digikey
1	IF-E93	LED verde	9,68€	9,68€	Digikey
1	IF-E96E	LED rojo	4,44€	4,44	Digikey
3	FB140-1-ND	Fibra óptica	1,60€	4,80€	Digikey
1	TL071	Amplificador operacional	0,56€	0,56€	Digikey
1	470pF	Condensador	0,32€	0,32€	Digikey
1	100nF	Condensador	0,32€	0,32€	Digikey
1	1MΩ	Resistencia	0,09€	0,09€	Digikey
1	330Ω	Resistencia	0,09€	0,09€	Digikey
2	100Ω	Resistencia	0,09€	0,18€	Digikey
1	1470-1213-ND	Fuente alimentación simétrica	62,7€	62,7€	Digikey
1		Bornes de 3 vías	0,03€	0,03€	Electrónica Gimeno
1	602-1098-ND	XBee Serie2	15,02€	15,02€	Digikey
1	1568-1075-ND	XBee Explorer Regulated	8,79€	8,79€	Digikey
1	32400-ND	Xbee USB Adapter	22,09€	22,08€	Digikey
1		Cable mini USB	3€	3€	Electrónica Gimeno
1		Arduino Due	47,19€	47,19€	Electrónica Gimeno
1		Placa perforada	8,77€	8,77€	Electrónica Gimeno
1		Cable red eléctrica	5€	5€	
100g		Plástico impresora 3D	1,99€	1,99€	
TOTAL				198,73€	

Tabla 8. Presupuesto material

5.2.3 Total

DESCRIPCIÓN	COSTE TOTAL
Humano	8.295 €
Material	198,73 €
TOTAL PROYECTO	8.493,73€

Tabla 9. Presupuesto total del proyecto

Capítulo 6. Conclusiones y propuesta de trabajo futuro

En el presente trabajo se ha diseñado, construido y verificado un sistema de registro óptico empleando fibra óptica para la medida tanto del potencial de acción como del nivel de calcio intracelular en cierto punto del corazón que se desee analizar. Este sistema permite mejorar el estudio de arritmias cardiacas y ofrece ciertas ventajas frente a otros sistemas que realizan la misma función, como puede ser el hecho de no tener que emplear un desacoplador electromecánico al realizar la medida, lo cual introduce artefactos en la señal registrada, ya que la fibra podrá moverse sincrónicamente con los latidos del corazón.

Se barajaron distintas posibilidades a la hora de elegir la fibra óptica más adecuada para nuestro sistema. En primer lugar se analizaron las fibras ópticas bifurcadas, este tipo de fibras presentan dos ramas a un lado, y una rama común al otro lado. De las dos ramas que se encuentran en el mismo lado de la fibra, una de ellas se emplea para excitar y la otra para registrar. El nodo común es lo que iría apuntando a lo que queremos analizar, en nuestra caso, iría tocando el corazón. Esta opción fue descartada debido a su elevado precio. Por ello se decidió emplear dos fibras ópticas independientes, una para excitar y la otra para registrar. Además era necesario que estas fibras no tuvieran ningún tipo de conector especialmente en el nodo que va pegado al corazón.

A continuación se desarrolló un interfaz de usuario para que el registro sea lo más intuitivo posible, así como la posibilidad de realizar una base de datos en Excel con todos los registros que se deseen. Existe además otro interfaz que nos permite visualizar cierto registro archivado en la base de datos únicamente introduciendo la fecha en la que el estudio fue realizado.

También permite que la comunicación entre nuestro sistema y el ordenador sea completamente inalámbrica al emplear un módulo XBee, reduciendo de manera considerable el cableado en el laboratorio.

Finalmente, al ser un sistema construido a partir sus componentes electrónicos resulta más económico que emplear un sistema de registro óptico completo que se ofrezca en el mercado.

6.1 Propuesta de trabajo futuro

En el futuro a este proyecto se le pueden realizar una serie de mejoras como pueden ser:

- Implementar el sistema empleando fibra óptica bifurcada de manera que únicamente sea necesario emplear un cable de fibra en lugar de dos. Debido a que estas están especialmente diseñadas con el objetivo de realizar tanto la excitación como el registro, serán más eficientes ya que se reducen las pérdidas de señal que se producen en el caso de registrar con otra fibra distinta a la de excitación.
- Permitir registrar dos variables fisiológicas simultáneamente. Para esto habrá que introducir un sistema de conmutación de los LEDs que nos permita registrar tanto el potencial de acción como el nivel de calcio intracelular (empleando los tintes necesarios) al mismo tiempo y que, posteriormente, podremos separar al realizar un pequeño procesado a la señal obtenida.
- En el receptor óptico emplear múltiples fibras ópticas en lugar de una sola y usar como receptor una cámara de alta velocidad CCD en lugar del fotodetector. Con esto obtendríamos una mayor resolución espacial.

Capítulo 7. Bibliografía

- [1] Gómez-Doblas J.J., Muñoz J., Alonso Martín J.J., Rodríguez-Roca G., Lobos J.M., Awamieh P., Permanyer-Miralda G., Chorro F.J., Anguita M., Roig E.: *Prevalencia de fibrilación auricular en España. Resultados del estudio OFRECE*. Rev Esp Cardiol. 67(2014):259-69
- [2] Go A.S., Hylek E.M., Philips K.A., Chang Y., Henault L.E., Selby J.V., Singer D.E.: *Singer, prevalence of Diagnoses atrial fibrillation in Adults*. Circulation. 102 (2000) 11-13.
- [3] Hart R.G., Coull B.M., Hart D.: *Early recurrent embolism associated with nonvalvular atrial fibrillation: a retrospective study*. Stroke. 14(1983) 688-93.
- [4] Okuyama Y., Miyauchi Y., Park A.M., Hamabe A., Zhou S., Hayashi H., Miyauchi M., Omichi C., Pak H.N., Brodsky L.A., Mandel W.J., Fishbein M.C., Karagueuzian H.S., Chen P.S.: *High resolution mapping of the pulmonary vein and the vein of Marshall during induce atrial fibrillation and atrial tachycardia in a canine model of pacing induced congestive heart failure*. J Am Coll Cardiol. 42(2003) 348-60.
- [5] Malmivuo J., Plonsey R.: *Bioelectromagnetism. The heart*. Oxford University Press, New York. (1995) 119-132.
- [6] Investigación Traslacional de Arritmias Cardiacas Secundarias a Canalopatías (Proyecto ITACA). *Canalopatías. Conceptos. Potencial de Acción Cardíaco*. <http://www.itaca.edu.es/potencial-accion-cardiaco.htm>. [online]
- [7] Wikipedia. *Potencial de acción cardíaco*. https://es.wikipedia.org/wiki/Potencial_de_acci%C3%B3n_card%C3%ADaco [online]
- [8] Morita H., Wu J., Zipes D.P.: *The QT syndromes: long and short*. Lancet. 30 (2008) 372(9640).
- [9] Chorro Gascó. F. J., Such-Belenguer L., López-Merino V.: *Modelos animales de enfermedad cardiovascular*. Revista Española De Cardiología. 62 (2009) 69
- [10] Jalfie J., Berefeld O., Mansour M.: *Mother rotors and fibrillatory conduction: A mechanism of atrial fibrillation*. Cardiovasc Res. 2002; 54: 204-216
- [11] Todd J. Herron, Peter Lee, José Jalife. *Optical Imaging of Voltage and Calcium in Cardiac Cells & Tissues*. Circulation Research. 2012; 110:609-623
- [12] Salzberg BM, Davila HV, Cohen LB. *Optical recording of impulses in individual neurones of an invertebrate central nervous system*. Nature. 1973; 246:508-509
- [13] Filgueiras-Rama D., Martins R.P., Ennis S.R., Mironov S., Jiang J., Yamazaki M., Kalifa J., Jalife J., Berenfeld O.: *High-Resolution Endocardial and Epicardial Optical Mapping in a Sheep Model of Stretch-Induced Atrial Fibrillation*. J Vis Exp.53 (2011) 53
- [14] Efimov IR, Fedorov VV, Joung B, Lin SF. *Mapping cardiac pacemaker circuits: methodological puzzles of the sinoatrial node optical mapping*. Circ Res. 2010; 106: 255-271
- [15] Entcheva E, Bien H. *Macroscopic optical mapping of excitation in cardiac cell networks with ultra-high spatiotemporal resolution*. Progr Biophys Mol Biol. 2006;92:232-257
- [16] Berenfeld O., Mandapati R., Dixit S., Skanes A.C., Chen J., Monsour M., Jalife J. *Spatially distributed dominant excitation frequencies reveal hidden organization in atrial fibrillation in the Langendorff-perfused sheep heart*. J Cardiovascular Electrophysiology 2000;11:869-879
- [17] Liu Y., Cabo C., Salomonsz R., Delmar M., Davidenko J., Jalife J.: *Effects of diacetyl monoxime on the electrical properties of sheep and guinea pig ventricular muscle*. Cardiovascular Research. 27 (1993) 1991-1997

- [18] Entcheva E., Kostov Y., Tchernev E., Tung L.: *Fluorescence imaging of electrical activity in cardiac cells using an all-solid-state system*. IEEE Trans Biomed Eng. 2014;51:333-341
- [19] Entcheva E., Bien H.: *Macroscopic optical mapping of excitation in cardiac cell networks with ultra-high spatiotemporal resolution*. Progr Biophys Mol Biol. 2006;92:232-257
- [20] Cohen L.B.: *Optical measurement of membrane potential*. Rev Physiol Biomech Pharmacol. 1978;83:35-88
- [21] Lev-Ram V., Grinvald A.: *Ca²⁺- and K⁺- dependent communication between central nervous system myelinated axons and oligodendrocytes revealed by voltage-sensitive dyes*. Proc Nat Acad Sci. 1986;83:6651-6655
- [22] Tsien R.Y.: *Simultaneous optical imaging of membrane potential and intracellular calcium*. J Electrocardiol. 2005;38:107-112
- [23] Rishi Arora, MD, Mithilesh K. Das, MD, Douglas P. Zipes, MD, Jianyi Wu, MD.: *Optical Mapping of Cardiac Arrhythmias*. Indian Pacing Electrophysiol. J. 2003;3(4):187-196
- [24] Industrial Fiberoptics: *High - Performance Plastic Optical Fiber*. Specification Sheet: <http://i-fiberoptics.com/pdf/sh4001.pdf> [online]
- [25] Industrial Fiberoptics: *Plastic Fiber Optic Green LED: IF-E93K*. Specification Sheet: <http://i-fiberoptics.com/pdf/ife93.pdf> [online]
- [26] Industrial Fiberoptics: *Plastic Fiber Optic Red LED: IF-E96E*. Specification Sheet: <http://i-fiberoptics.com/pdf/if-e96edatasheet.pdf> [online]
- [27] Industrial Fiberoptics: *Plastic Fiber Optic Photodiode: IF-D91*. Specification Sheet: <http://i-fiberoptics.com/pdf/ifd91.pdf> [online]
- [28] Mike Smith: *A High Bandwidth, Low Noise Optical VLF Link*. <http://www.vlf.it/smith1/opticalink.html> [online]
- [29] Arduino. *Arduino Due*. <https://www.arduino.cc/en/Main/ArduinoBoardDue> [online]
- [30] Xbee. *Xbee Serie 2*. <http://xbee.cl/xbee-2mw-wire-antenna/> [online]

Anexo I. Importación librería temporizador

Para el registro de datos emplearemos la librería *DueTimer* para poder ajustar una frecuencia de muestreo. DueTimer nos permite establecer la frecuencia a la que queremos realizar determinada acción, en nuestro caso, determinaremos la frecuencia con la que queremos leer la salida del detector óptico y con la que le enviaremos al ordenador estos datos registrados.

Para importar esta librería tenemos que seguir los siguientes pasos:

- 1) Accedemos a la página web: <https://github.com/ivanseidel/DueTimer>
- 2) En "README.md" dentro de "Installation" pinchamos sobre "Download" en el punto 1.
- 3) Nos descargamos el archivo comprimido de la última versión (v1.4.7) en formato .zip o .tar.gz dependiendo del sistema operativo en el que estemos trabajando.
- 4) Descomprimos el archivo descargado y le cambiamos el nombre a la carpeta a "DueTimer" (eliminamos la versión).
- 5) Vamos a la carpeta general de Arduino (en el disco duro C: dentro de Archivos de programa si nos encontramos en Windows).
- 6) En la carpeta "libraries" pegamos la carpeta descomprimida y modificado del paso 4).
- 7) Reiniciamos el IDE de Arduino.
- 8) Si el proceso se ha realizado correctamente la librería debería aparecer en la lista.

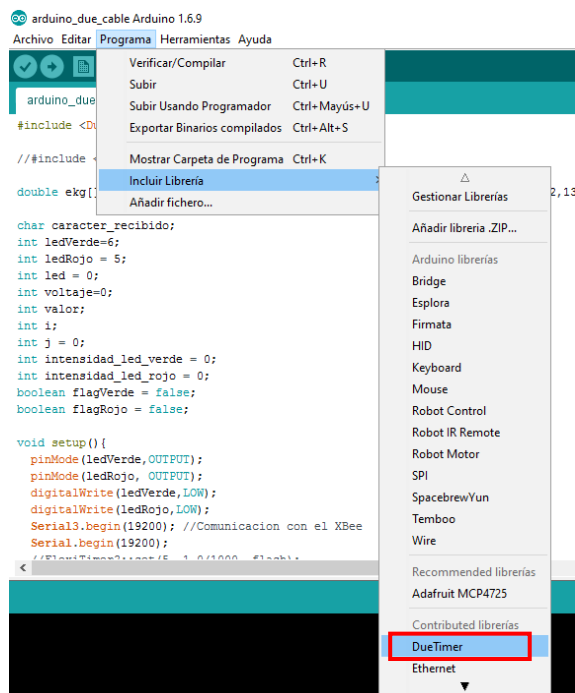


Figura 51. Verificación de librería añadida

Anexo II. Programación XBee mediante X-CTU

Como se ha comentado vamos a emplear dos módulos XBee, un coordinador y un esclavo. Para programarlos vamos a emplear el programa X-CTU.

Para programarlos tenemos que seguir los siguientes pasos:

1. Ponemos en el *XBee USB Adapter* el módulo que queremos configurar primero. Empezaremos por el coordinador.
2. Seleccionamos el puerto USB correspondiente al adaptador USB serie y pulsamos Test/Query. Si éste es detectado aparecerá un mensaje indicando que su estado es correcto así como su número de serie, tipo de modem y versión del firmware. En caso de que no sea detectado aparecerá un mensaje indicándolo.



Figura 52. Conexión con el XBee realizada correctamente

3. Abrimos la pestaña *Modem Configuration* y pulsamos *Read* para leer los comandos AT que tenga ya establecidos. Configuraremos los siguientes parámetros:
 - **PAN ID:** número de identificación de la red. Debe de ser el mismo para todos los XBee que formen la red. En nuestro caso elegimos la 3001.
 - **Destination Address High (SH):** parte alta del número de serie del Xbee con el que se quiere comunicar (esclavo).
 - **Destination Address Low (SL):** parte baja del número de serie del Xbee con el que se quiere comunicar (esclavo).

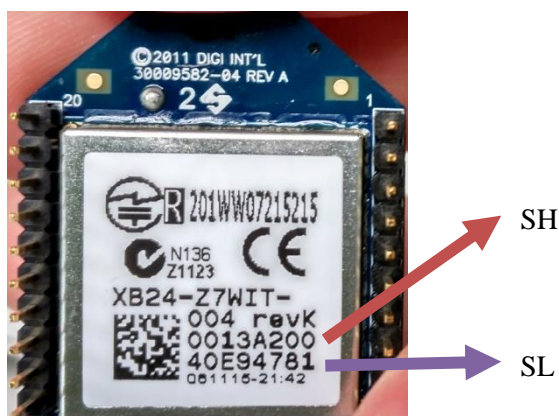


Figura 53. XBee esclavo

- **BaudRate:** velocidad de la comunicación, 115200 baudios en nuestro caso
4. En la opción *Funcion Set*, seleccionamos ZIGBEE COORDINATOR AT

5. Seleccionamos *Write* y esperamos a que termine la configuración.
6. Si todo se ha realizado correctamente no debe aparecer ningún mensaje de error y debe quedar de la siguiente forma:

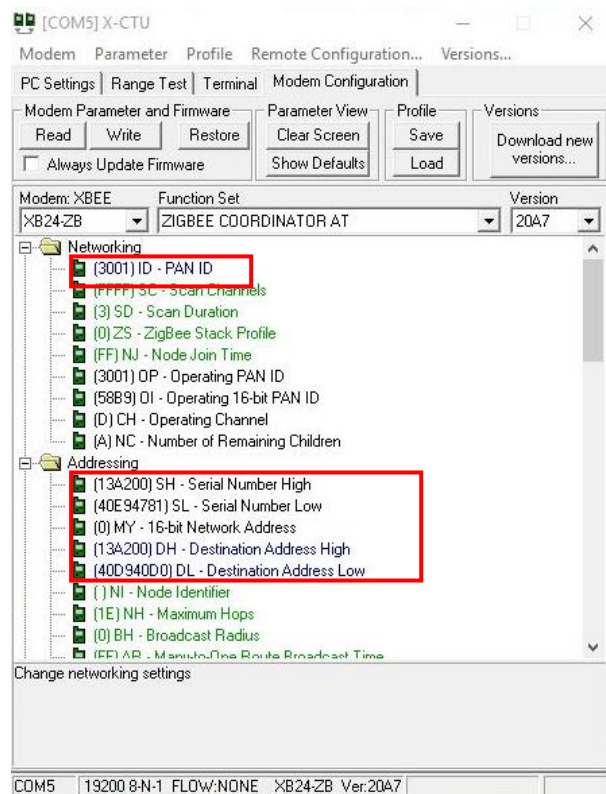


Figura 54. Parámetros XBee coordinador

7. Cerramos el X-CTU y ponemos en el *XBee USB Adapter* el otro XBee (esclavo).
8. Realizamos los mismos pasos descritos anteriormente con los parámetros correspondientes: PAN ID: 3001; DH: 13A200; DL: 40E94781; BD: 115200; Function Set: ZIGBEE ROUTER AT.
Podemos observar que el DH y DL de un XBee corresponden con el SH y SL respectivamente del otro XBee.
9. Una vez configurados podemos comprobar que funcionan, para ello necesitaremos dos adaptadores XBee-USB donde conectaremos cada uno de los módulos. Abrimos dos programas de X-CTU y en cada uno de ellos abrimos la pestaña *Terminal*. Escribimos algo en cada uno de los terminales y vemos que esto mismo aparece en el terminal del otro XBee (Figura 53).

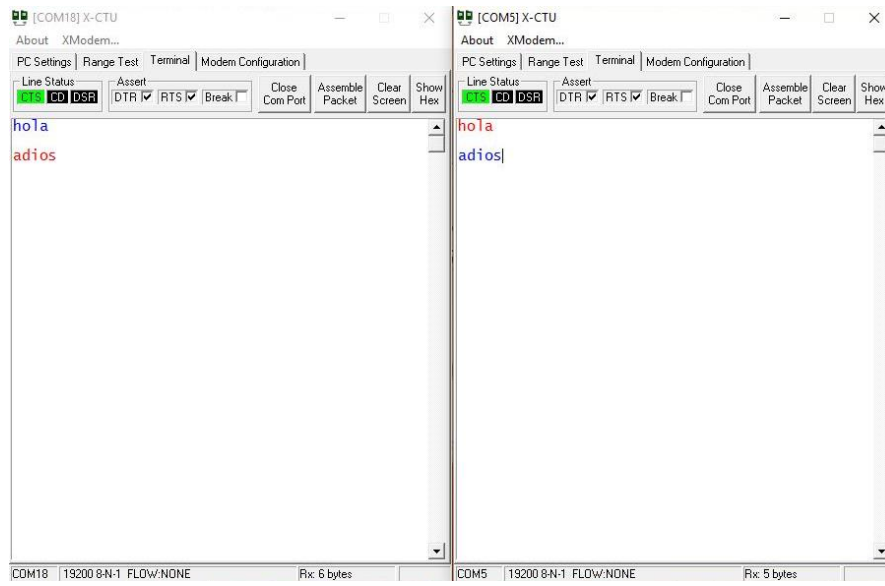


Figura 55. Conexión entre los XBee realizada correctamente

Anexo III. Firmware de Arduino

```
#include <DueTimer.h>

// VALORES ECG
//double ekg[] = {0,0.19136,0.27337,0.41006,0.54675,0.74282,2.0672,4.8761,

char caracter_recibido;
int voltaje=0;
int valor;
int i;
int j = 0;
int intensidad_led_verde = 0;
int intensidad_led_rojo = 0;
boolean flagVerde = false;
boolean flagRojo = false;

void setup(){
  Serial3.begin(115200); //Comunicacion con el XBee
  Serial.begin(9600);
  Timer4.attachInterrupt(registro);
  Timer4.setFrequency(500); //frecuencia de muestreo
}

void registro() {
  voltaje=analogRead(8);
  Serial3.println(voltaje);
}

void loop(){

// _____
// CÓDIGO QUE NOS PERMITE GENERAR LA SEÑAL DE ECG CON LAS QUE SE HAN HECHO LAS PRUEBAS
//   for(int k = 0;k<100;k++)
//     {
//       analogWrite(DAC1, ((ekg[k])*0.5)+190);
//       delay(10);
//     }
// _____

if(Serial3.available(>0){
  caracter_recibido = Serial3.read(); //LEEMOS CARACTER RECIBIDO
  switch(caracter_recibido){
    case 'c': //CONEXION
      //delay(1000);
      Serial3.println('d');
      Serial.println('d');
      break;
    case '1': //REGISTRO DE DATOS
      Timer4.start();
      break;
    case '2': //INSTRUCCION ENCENDER LED VERDE
      intensidad_led_verde = 4095;
      flagVerde = true;
      analogWrite(DAC0, intensidad_led_verde);
      break;
  }
}
```

```

case '3': //INSTRUCCION ENCENDER LED ROJO
    intensidad_led_rojo = 4095;
    flagRojo = true;
    analogWrite(DAC1, intensidad_led_rojo);
    break;

case '4': //INSTRUCCION SUBIR INTENSIDAD
    if(flagVerde==true){
        intensidad_led_verde = intensidad_led_verde + 5;
        if(intensidad_led_verde >= 4095){
            //POR MUCHO QUE LLEGUE LA INSTRUCCION DE SUBIR INTENSIDAD ESTA NO PUEDE SER MAYOR DE 4095
            intensidad_led_verde = 4095;
        }
        analogWrite(DAC0, intensidad_led_verde);
    }
    else{
        if(flagRojo==true){
            intensidad_led_rojo = intensidad_led_rojo + 5;
            if(intensidad_led_rojo >= 4095){
                //POR MUCHO QUE LLEGUE LA INSTRUCCION DE SUBIR INTENSIDAD ESTA NO PUEDE SER MAYOR DE 4095
                intensidad_led_rojo = 4095;
            }
            analogWrite(DAC1, intensidad_led_rojo);
        }
    }
    break;

case '5': //INSTRUCCION BAJAR INTENSIDAD
    if(flagVerde==true){
        intensidad_led_verde = intensidad_led_verde - 5;
        Serial3.println(intensidad_led_verde);
        if(intensidad_led_verde <= 0){
            //POR MUCHO QUE LLEGUE LA INSTRUCCION DE BAJAR INTENSIDAD ESTA NO PUEDE SER MENOR DE 0
            intensidad_led_verde = 0;
        }
        analogWrite(DAC0, intensidad_led_verde);
    }
    else{
        if(flagRojo==true){
            intensidad_led_rojo = intensidad_led_rojo - 5;
            Serial3.println(intensidad_led_rojo);
            if(intensidad_led_rojo <= 0){
                //POR MUCHO QUE LLEGUE LA INSTRUCCION DE BAJAR INTENSIDAD ESTA NO PUEDE SER MENOR DE 0
                intensidad_led_rojo = 0;
            }
            analogWrite(DAC1, intensidad_led_rojo);
        }
    }
    break;

case '6': //INSTRUCCION APAGAR LED VERDE
    intensidad_led_verde = 0;
    flagVerde = false;
    analogWrite(DAC0, intensidad_led_verde);
    break;
case '7': //INSTRUCCION APAGAR LED ROJO
    intensidad_led_rojo = 0;
    flagRojo = false;
    analogWrite(DAC1, intensidad_led_rojo);
    break;
case '8': //DETENER ENVIO DATOS
    Timer4.stop();
    break;
}
}
}

```

Anexo IV. GUIDE de Matlab

IV.1 Pantalla registro de datos

```
function varargout = registro(varargin)
% PANTALLA QUE NOS PERMITE REGISTRAR UN NUEVO ESTUDIO. EN ESTE DEBEMOS
% INDICAR EL PUERTO DEL XBEE COORDINADOR, INDICAR EL NUMERO DE
MUESTRAS QUE
% QUEREMOS REGISTRAR, INICIAR LA COMUNICACION CON EL XBEE ROUTER,
CONTROLAR
% EL ENCENDIDO Y APAGADO DE LOS LEDS ASÍ COMO SU INTENSIDAD, INICIAR
EL
% REGISTRO DE DATOS. UNA VEZ FINALIZADO EL REGISTRO DE DATOS APARECERÁ
LA
% SEÑAL REGISTRADA EN LA GRÁFICA, SI QUEREMOS ALMACENARLA DEBEREMOS
INDICAR
% EL TIPO DE MEDIDA REALIZADA (VOLTAJE O CALCIO) Y LA ALMACENAREMOS.
POR
% ULTIMO PERMITE FINALIZAR LA CONEXION Y VOLVER A MENU PRINCIPAL

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @registro_OpeningFcn, ...
                  'gui_OutputFcn',  @registro_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function registro_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = registro_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function desconexion_Callback(hObject, eventdata, handles)
% BOTON QUE NOS PERMITE CERRAR EL PUERTO DEL MODULO XBEE, REALIZANDO
% LA DESCONEXION
global com;
global comunicacion;
global connect;
global s;
caracter_off_potencial = 55; %CARACTER ASCII 7
```

```

caracter_off_calcio = 54; %CARACTER ASCII 6
set(handles.conectado, 'string', 'Desconectado');
set(handles.text_on_off, 'BackgroundColor', 'r');
set(handles.text_on_off, 'String', 'OFF' );
fwrite(s, caracter_off_potencial);
fwrite(s, caracter_off_calcio);
fclose(instrfind({'Port'}, {com}));
delete(instrfind({'Port'}, {com}))
comunicacion=0;
conect=0;

function menu_Callback(hObject, eventdata, handles)
% VUELTA A LA PANTALLA PRINCIPAL
close
clear all
menu_principal

function seleccion_medida_Callback(hObject, eventdata, handles)
% MENU DESPLEABLE QUE NOS PERMITE INDICAR EL TIPO DE MEDIDA QUE HEMOS
% REALIZADO:
% VOLTAJE: MEDIDA DEL POTENCIAL DE ACCION
% CALCIO: MEDIDA DEL CALCIO INTRACELULAR
medidas = cellstr(get(hObject, 'String'));
seleccion = medidas{get(hObject, 'Value')};
handles.seleccion_medida=seleccion;
guidata(hObject, handles);

function seleccion_medida_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function almacenar_Callback(hObject, eventdata, handles)
% BOTON QUE AL SER PULSADO ALMACENA LA MEDIDA REALIZADA EN UN ARCHIVO
EXCEL
global voltaje;
medida = handles.seleccion_medida;
medida=lower(medida);
% COMPROBAMOS QUE TODOS LOS CARACTERES DE "SELECCION_MEDIDA" SEAN
% LETRAS, HAY VECES QUE SE METEN ESPACION, EN ESTOS CASOS BUSCAMOS LAS
% POSICIONES DE ESTOS Y LOS ELIMINAMOS
comprobacion=isletter(medida);
ind=[];
for c=1:length(comprobacion)
    if comprobacion(c)==0
        ind = [ind c];
    end
end
for h=1:length(ind)
    medida(h)=[];
end
m = msgbox('Guardando datos, ¡Espere!');
dir_actual = pwd;
directorio_archivo = handles.directorio;
archivo = handles.nombre_archivo;
cd(directorio_archivo)
archivos = dir;
for arch = 1:length(archivos)

```



```

    if strcmp(archivos(arch).name, [archivo '.xlsx'])
        [num,txt,row]=xlsread([archivo '.xlsx']);
        fechas=txt(:,1);
        total_fechas = length(fechas);
        celda_libre= total_fechas + 1;
        break;
    else
        celda_libre = 1;
    end
end
fecha = datetime('now');
fechaStr = datestr(fecha);
xlswrite([archivo '.xlsx'],{fechaStr},'Hojal',
sprintf('A%d',celda_libre));
xlswrite([archivo '.xlsx'],{medida},'Hojal',
sprintf('B%d',celda_libre));
xlswrite([archivo '.xlsx'],voltaje,'Hojal',
sprintf('C%d',celda_libre));
%Mensaje al usuario, indaca la medicion finalizada
close(m);
msgbox('Datos guardados');
cd(dir_actual)

function button_calcio_Callback(hObject, eventdata, handles)
%ENCIENCE EL LED VERDE PARA REALIZAR UNA ADQUISICION DE DATOS DE
%CALCIO INTRACELULAR
global s
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    set(handles.text_on_off, 'BackgroundColor', 'g');
    set(handles.text_on_off,'String', 'ON CALCIO' );
    caracter_calcio = 50; %CARACTER ASCII 2
    fwrite(s, caracter_calcio);
else
    set(handles.text_on_off, 'BackgroundColor', 'r');
    set(handles.text_on_off,'String', 'OFF' );
    caracter_off_calcio = 54; %CARCTER ASCII 6
    fwrite(s, caracter_off_calcio);
end

function button_voltaje_Callback(hObject, eventdata, handles)
%ENCIENDE EL LED ROJO QUE PERMITE REALIZAR UNA ADQUSICION DE POTENCIAL
%DE ACCION
global s
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    set(handles.text_on_off, 'BackgroundColor', 'g');
    set(handles.text_on_off,'String', 'ON VOLTAJE' );
    caracter_potencial = 51; %CARACTER ASCII 3
    fwrite(s, caracter_potencial);

else
    set(handles.text_on_off, 'BackgroundColor', 'r');
    set(handles.text_on_off,'String', 'OFF' );
    caracter_off_potencial = 55; %CARACTER ASCII 7
    fwrite(s, caracter_off_potencial);
end

function menos_intens_Callback(hObject, eventdata, handles)
% BOTON QUE NOS PERMITE BAJAR LA INTENSIDAD DEL LED ENCENDIDO EN ESE
% MOMENTO

```

```

global s
global flagMenos
caracter_menos = 53; %CARACTER ASCII 5
fwrite(s,caracter_menos);
flagMenos = 1;

function mas_intens_Callback(hObject, eventdata, handles)
% BOTON QUE NOS PERMITE SUBIR LA INTENSIDAD DE LED ACTIVADO EN ESE
%MOMENTO
global s
global flagMas
flagMas = 1;
caracter_mas = 52; %CARACTER ASCII 4
fwrite(s,caracter_mas);

function registrar_Callback(hObject, eventdata, handles)
%BOTON QUE NOS PERMITE INICIAR EL REGISTRO DE DATOS
global voltaje
global s
global flagMas
global flagMenos
nMuestras=handles.muestras;
nMuestras = str2double(nMuestras);
voltaje=zeros(1,nMuestras);
valorADC=0;
i=1;
flagMas = 0;
flagMenos = 0;
caracter_enviado=49;
fwrite(s, caracter_enviado);
while i<=nMuestras
    valorADC=fscanf(s, '%i')
    [fil,~]=size(valorADC);
    if fil>1
        valorADC(2:end)=[];
    end
    if numel(valorADC)>0
        voltaje(i) = valorADC*5/1024;
    else
        voltaje(i) = voltaje(i-1);
    end
    i=i+1;
end
caracter_detener = 56; %CARACTER ASCII 8
fwrite(s,caracter_detener); %COMANDO PARA QUE PARE LA LECTURA DE DATOS
EN EL ARDUINO
nMuestras = length(voltaje);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltaje);
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on

function conectar_Callback(hObject, eventdata, handles)
% BOTON QUE ABRE EL PUERTO DEL MODULO XBEE ESTABLECIENDO LA CONEXION
global com;
global conect;
global comunicacion;
global s;
delete(instrfind({'Port'},{com}))

```

```

s = serial(com,'BaudRate',115200);
fopen(s);
conect=1;
comunicacion = 0;
if conect == 1
    caracter_enviado = 99;
    fwrite(s, caracter_enviado);
    recibido = fscanf(s, '%c')
    if strcmpi(recibido(1,1), 'd')
        set(handles.conectado, 'string', 'Conectado');
        comunicacion = 1;
    end
end

function conectado_Callback(hObject, eventdata, handles)
%NOS INDICA CUANDO SE ESTABLECE LA CONEXION APARECIENDO LA PALABRA
%CONECTADO EN LA CASILLA DADA

function conectado_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function puerto_Callback(hObject, eventdata, handles)
% MENU DESPLEGABLE QUE NOS PERMITE ELEGIR EL PUERTO DEL MODULO XBEE
% CONECTADO EL ORDENADOR MEDIANTE UN USB
global com
port = cellstr(get(hObject, 'String'));
com = port{get(hObject, 'Value')};
handles.puerto = com;
guidata(hObject,handles);

function puerto_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function muestras_Callback(hObject, eventdata, handles)
%INTRODUCIMOS EL NUMERO DE MUESTRAS QUE QUEREMOS REGISTRAR
nMuestras = get(hObject, 'String');
handles.muestras = nMuestras;
guidata(hObject,handles);

function muestras_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function directorio_Callback(hObject, eventdata, handles)
%DIRECTORIO DEL ARCHIVO EXCEL EN EL QUE QUEREMOS ALMACENAR LA MEDIDA
%REALIZADA
directorio = uigetdir;
handles.directorio = directorio;
guidata(hObject,handles);

function nombre_archivo_Callback(hObject, eventdata, handles)
%NOMBRE DE ARCHIVO EXCEL EN EL QUE QUEREMOS GUARDAR EL REGISTRO
%REALIZADO. SI NO EXISTE UN ARCHIVO EXCEL CON ESTE NOMBRE SE CREARÁ
archivo=get(hObject,'String');
handles.nombre_archivo=archivo;
guidata(hObject,handles);

function nombre_archivo_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

IV.2 Pantalla representar medida almacenada

```
function varargout = almacenado(varargin)
% INTERFAZ QUE PERMITE REPRESENTAR REGISTROS DE DATOS REALIZADOS
% ANTERIORMENTE Y QUE SE ENCUENTRAN ALMACENADOS EN UNA ARCHIVO EXCEL A
% MODO DE BASE DE DATOS

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @almacenado_OpeningFcn, ...
                  'gui_OutputFcn',  @almacenado_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function almacenado_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = almacenado_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function dia_Callback(hObject, eventdata, handles)
% ENTRADA DE TEXTO DONDE INTRODUCIMOS EL DIA DE LA MEDIDA QUE QUEREMOS
% REPRESENTAR
diaStr=get(hObject,'String');
diaNum=str2double(diaStr);
handles.dia=diaNum;
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function dia_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function mes_Callback(hObject, eventdata, handles)
% ENTRADA DE TEXTO DONDE INDICAMOS EL MES DE LA MEDIDA QUE QUEREMOS
% REPRESENTAR
mesStr=get(hObject,'String');
```

```

mesNum=str2double(mesStr);
handles.mes=mesNum;
guidata(hObject,handles);

function mes_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function anyo_Callback(hObject, eventdata, handles)
% ENTRADA DE TEXTO DONDE INDICAMOS EL AÑO DE LA MEDIDA QUE QUEREMOS
% REPRESENTAR
anyoStr=get(hObject,'String');
anyoNum=str2double(anyoStr);
handles.anyo=anyoNum;
guidata(hObject,handles);

function anyo_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function listbox1_Callback(hObject, eventdata, handles)
%LISTBOX DONDE APARECEN LAS MEDIDAS QUE RESPONDEN A LOS FILTROS DE
FECHA Y
%TIPO DE MEDIDA INTRODUCIDOS
global voltaje
global voltaje_original
global cambiadas
lista=get(handles.listbox1,'String');
set(handles.listbox1,'String',lista);
fechas_lista = cellstr(get(hObject, 'String'));
seleccionado=fechas_lista{get(hObject, 'Value')};
if strcmp(seleccionado,'VOLTAJE') || strcmp(seleccionado,'CALCIO')
    helpdlg('Seleccione una fecha')

else
    fecha_seleccionada=datetime(seleccionado);
    num_seleccionada=datetime(fecha_seleccionada);
    [num,txt,row]=xlsread('Librol.xlsx');
    fechas=txt(:,1);
    fechas(1)=[];
    num_fechas=[];

    for i=1:length(fechas)
        for z = 1:length(cambiadas)
            if i==cambiadas(z)
                dia = day(fechas(i));
                mes = month(fechas(i));
                mes2 = mes; mes = dia; dia = mes2;
                anyo = year(fechas(i));
            end
        end
    end
end

```

```

        hora = hour(fechas(i));
        minuto = minute(fechas(i));
        segundo = second(fechas(i));
        date = datetime(anyo,mes,dia,hora,minuto,segundo);
    else
        date=datetime(fechas(i));
    end

end

num_fechas=[num_fechas datenum(date)];
end
for j=1:length(num_fechas)
    if num_fechas(j)==num_seleccionada
        indice_selccionada=j;
    end
end

voltaje=num(indice_selccionada,:);
voltaje_original = voltaje;
nMuestras = length(voltaje);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltaje);
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on
end

function listBox1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in boton_ok.
function boton_ok_Callback(hObject, eventdata, handles)
% AL PULSAR EL BOTON SE BUSCA EN EL EXCEL LA FECHA Y TIPO DE MEDIDA
% REALIZADO
global cambiadas
dia=handles.dia;
mes=handles.mes;
anyo=handles.anyo;
t_buscado=0;
medida=handles.selec_medida;
medida=lower(medida);
% COMPROBAMOS QUE TODOS LOS CARACTERES DE "MEDIDA" SEAN LETRAS, HAY
% VECES QUE SE METEN ESPACION, EN ESTOS CASOS BUSCAMOS LAS POSICIONES
% DE ESTOS Y LOS ELIMINAMOS
comprobacion=isletter(medida);
ind=[];
for c=1:length(comprobacion)
    if comprobacion(c)==0
        ind = [ind c];
    end
end

medida(ind)=[];

```

```

if(anyo~=0 && mes~=0 && dia~=0)
    t_buscado=datetime(anyo,mes,dia);
end

t_buscadoNum=datenum(t_buscado);
dia_buscado=day(t_buscadoNum);
mes_buscado=month(t_buscadoNum);
anyo_buscado=year(t_buscadoNum);
dir_actual = pwd;
directorio_archivo = handles.directorio;
archivo = handles.nombre_archivo;
cd(directorio_archivo)
[num,txt,row]=xlsread([archivo '.xlsx']);
cd(dir_actual)
fechas=txt(:,1);
fechas(1)=[];
medidas=txt(:,2);
medidas(1)=[];
fecha_num=[];
dias=[];
meses=[];
anyos=[];
cambiadas = [];
for j=1:length(fechas)
    fecha_num(j)=datenum(datetime(fechas(j)));
    dias(j)=day(fecha_num(j));
    meses(j)=month(fecha_num(j));
    fecha_buena = cell2mat(fechas(j));
    if dias(j) ~= str2double(fecha_buena(1:2))
        cambiadas = [cambiadas j];
        meses2 = meses(j);
        meses(j) = dias(j);
        dias(j) = meses2;
    end
    anyos(j)=year(fecha_num(j));
    horas(j)=hour(fecha_num(j));
    minutos(j)=minute(fecha_num(j));
    segundos(j)=second(fecha_num(j));

end

fechas_posibles1=[];
fechas_posibles2=[];
fechas_posibles3=[];
fechas_posibles4=[];
fechas_posibles5=[];
fechas_posibles=[];

if strcmp(medida,'todos')
    for f=1:length(fechas)
        if (dia_buscado == dias(f)) && (mes_buscado==meses(f)) &&
        (anyo_buscado==anyos(f))
            fechas_posibles1=[fechas_posibles1 ;
datestr(datetime(anyos(f),meses(f),dias(f),horas(f),minutos(f),segundo
s(f))))];
            end
            if strcmp(cell2mat(medidas(f)),'voltaje')
                fechas_posibles2=[fechas_posibles2 ;
datestr(datetime(anyos(f),meses(f),dias(f),horas(f),minutos(f),segundo
s(f))))];
            end
            if strcmp(cell2mat(medidas(f)),'calcio')

```



```

        fechas_posibles3=[fechas_posibles3 ;
datestr(datetime(anyos(f),meses(f),dias(f),horas(f),minutos(f),segundo
s(f)))]];
    end
end
    for j=1:size(fechas_posibles1)
        for k=1:size(fechas_posibles2)
            if strcmp(fechas_posibles1(j,:),fechas_posibles2(k,:))
                fechas_posibles4=[fechas_posibles4;
fechas_posibles1(j,:)];
            end
        end
    end
    for l=1:size(fechas_posibles1)
        for m=1:size(fechas_posibles3)
            if strcmp(fechas_posibles1(l,:),fechas_posibles3(m,:))
                fechas_posibles5=[fechas_posibles5;
fechas_posibles1(l,:)];
            end
        end
    end
    if isempty(fechas_posibles1)
        fechas_posibles4=fechas_posibles2;
        fechas_posibles5=fechas_posibles3;
    end
    if isempty(fechas_posibles2)
        fechas_posibles4=fechas_posibles1;
    end
    if isempty(fechas_posibles3)
        fechas_posibles5=fechas_posibles1;
    end

fechas_posibles={'VOLTAJE';fechas_posibles4;'CALCIO';fechas_posibles5}
;
else
    for i=1:length(fechas)
        if (dia_buscado==dias(i)) && (mes_buscado==meses(i)) &&
(anyo_buscado==anyos(i))
            fechas_posibles1=[fechas_posibles1 ;
datestr(datetime(anyos(i),meses(i),dias(i),horas(i),minutos(i),segundo
s(i)))]];
        end
        if strcmp(medida,cell2mat(medidas(i)))
            fechas_posibles2=[fechas_posibles2 ;
datestr(datetime(anyos(i),meses(i),dias(i),horas(i),minutos(i),segundo
s(i)))]];
        end
    end

    for n=1:size(fechas_posibles1)
        for p=1:size(fechas_posibles2)
            if strcmp(fechas_posibles1(n,:),fechas_posibles2(p,:))
                fechas_posibles=[fechas_posibles;
fechas_posibles1(n,:)];
            end
        end
    end
    if isempty(fechas_posibles1)
        fechas_posibles=fechas_posibles2;
    end
    if isempty(fechas_posibles2)

```

```

        fechas_posibles=fechas_posibles1;
    end
end

set(handles.listbox1,'String',fechas_posibles);

% --- Executes on button press in excel.
function excel_Callback(hObject, eventdata, handles)
% AL PULSAR EL BOTON ABRE EL ARCHIVO EXCEL PARA QUE PODAMOS ELEGIR LA
% MEDIDA QUE QUEREMOS REPRESENTAR
global voltaje
dir_actual = pwd;
directorio_archivo = handles.directorio;
archivo = handles.nombre_archivo;
cd(directorio_archivo)
voltaje=xlsread([archivo '.xlsx'],-1);
cd(dir_actual)
nMuestras = length(voltaje);
fs = 500;
t = (0:nMuestras-1)/fs;
plot(t,voltaje);
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on

function edit_fpbajo_Callback(hObject, eventdata, handles)
% CASILLA EN LA QUE INTRODUCIMOS LA FRECUENCIA A LA QUE QUEREMOS
% ESTABLECER EL FILTRO PASO BAJO
frecStr=get(hObject,'String');
frec=str2double(frecStr);
handles.edit_fpbajo=frec;
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_fpbajo_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in fpbajo.
function fpbajo_Callback(hObject, eventdata, handles)
% REALIZA UN FILTRADO PASO BAJO CON FRECUENCIA DE CORTE LA INDICADA EN
% LA CASILLA ANTERIOR
global voltaje
frec = handles.edit_fpbajo;
fs = 500;
N = 10;
[b2,a2] = butter(N,frec/(fs/2),'low');
v_filtrado = filtfilt(b2,a2,voltaje);
nMuestras = length(v_filtrado);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,v_filtrado)
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on
voltaje = v_filtrado;

```

```

% --- Executes on button press in fpalto.
function fpalto_Callback(hObject, eventdata, handles)
% ELEMINA EL VALOR MEDIO REALIZANDO UN FILTRADO PASO BAJO PARA
% ELIMINAR EL RUIDO DE CONTINUA
fs = 500;
global voltaje
voltaje = voltaje - mean(voltaje);
N = round(fs./40);
voltajed = decimate(voltaje,N);
fsd = fs/N;
fpa = 0.6;
[b1,a1] = butter(10,fpa/(fsd/2));
baseline = filtfilt(b1,a1,voltajed);
baseline = interp(baseline,N);
nP = length(baseline);
nS = length(voltaje);
nF = min(nS,nP);
voltajef = voltaje(1:nF)-baseline(1:nF);
nMuestras = length(voltajef);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltajef)
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on
voltaje = voltajef;

% --- Executes on button press in pushbutton_menu.
function pushbutton_menu_Callback(hObject, eventdata, handles)
% VUELTA AL MENU PRINCIPAL
close
menu_principal

% --- Executes on button press in pushbutton_original.
function pushbutton_original_Callback(hObject, eventdata, handles)
% VOLVER A LA SEÑAL ORIGINAL DESHACIENDO LOS FILTRADOS REALIZADOS
global voltaje_original
global voltaje
voltaje=voltaje_original;
nMuestras = length(voltaje);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltaje)
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on

% --- Executes on selection change in selec_medida.
function selec_medida_Callback(hObject, eventdata, handles)
% MENU DESPLEGABLE QUE NOS PERMITE SELECCIONAR EL TIPO DE MEDIDA QUE
% QUEREMOS BUSCAR
medidas = cellstr(get(hObject,'String'));
seleccion = medidas{get(hObject,'Value')};
handles.selec_medida=seleccion;
guidata(hObject,handles);

```

```

% --- Executes during object creation, after setting all properties.
function selec_medida_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
%SLIDER QUE NOS PERMITE NAVEGAR POR LA SEÑAL UNA VEZ REPRESENTAMOS UN
% RANGO DADO
global voltaje
global limite_inferior
global limite_superior
global valor
nMuestras = length(voltaje);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltaje)
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on
valor_nuevo = get(hObject, 'Value')
voltaje2 = voltaje;
voltaje2(isnan(voltaje2(:)))=[];
if valor_nuevo>valor
    limite_superior = limite_superior+0.5;
    limite_inferior = limite_inferior + 0.5;
    valor = valor+1;

elseif valor_nuevo<valor
    limite_superior = limite_superior-0.5;
    limite_inferior = limite_inferior - 0.5;
    valor = valor-1;

end

axis([limite_inferior limite_superior min(voltaje)-0.3
max(voltaje)+0.3]);

function slider1_CreateFcn(hObject, eventdata, handles)

if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function margen_vision_Callback(hObject, eventdata, handles)
% MARGEN EN SEGUNDOS QUE QUEREMOS VER DE LA IMAGEN
global voltaje
global limite_inferior
global limite_superior
global valor
margenStr=get(hObject,'String');
margenNum=str2double(margenStr);

```

```

handles.margen_vision=margenNum;
valor = 0;
limite_superior = margenNum;
limite_inferior = 0;
nMuestras = length(voltaje);
fs = 500;
t = [0:nMuestras-1]/fs;
plot(t,voltaje)
ylabel('Voltaje(V)');
xlabel('Tiempo(s)');
grid on
axis([limite_inferior limite_superior min(voltaje)-0.3
max(voltaje)+0.3]);
voltaje2 = voltaje;
voltaje2(isnan(voltaje2(:)))=[];
set(handles.slider1,'Min',0);
mmm = length(voltaje2)/fs-margenNum;
set(handles.slider1,'Max',round(mmm)/0.5);
set(handles.slider1,'Value',0);
set(handles.slider1,'SliderStep',[1/(mmm/0.5) , 10/(mmm/0.5) ]);
guidata(hObject,handles);

function margen_vision_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function directorio_Callback(hObject, eventdata, handles)
%DIRECTORIO EN EL QUE SE ENCUENTRA EL ARCHIVO EXCEL CON LA MEDIDA QUE
%QUEREMOS REPRESENTAR ALMACENADA
directorio = uigetdir;
handles.directorio = directorio;
guidata(hObject,handles);

function nombre_archivo_Callback(hObject, eventdata, handles)
%NOMBRE DEL ARCHIVO EXCEL EN EL QUE SE ENCUENTRO LA MEDIDA QUE
%QUEREMOS REPRESENTAR
archivo=get(hObject,'String');
handles.nombre_archivo=archivo;
guidata(hObject,handles);

function nombre_archivo_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```