



Monitor EMG con conexión inalámbrica

Memoria del Trabajo Final de Carrera
de Ingeniería Electrónica
realizado por
Daniel Abad Navarro
y dirigido por
Francisco Carlos Bogónez Franco
Bellaterra, 15 de Septiembre de 2008



Universitat
Autònoma
de Barcelona



Escola Tècnica Superior d'Enginyeria

El Profesor Francisco Carlos Bogónez Franco, profesor del Departamento de Microelectrónica y Sistemas Electrónicos de la Universidad Autònoma de Barcelona

CERTIFICA

que la memoria ***Monitor EMG con conexión inalámbrica*** que presenta Daniel Abad Navarro como trabajo final de carrera de Ingeniería Electrónica, se ha realizado bajo su dirección.

Bellaterra (Cerdanyola del Vallès), 15 de Septiembre de 2008.

Francisco Carlos Bógonez Franco

Daniel Abad Navarro

Firma:

Bellaterra, 15 de Septiembre de 2008

ÍNDICE

1.	La electromiografía (EMG)	4
	1.1 Introducción.	5
	1.2 Historia de la electromiografía.	5
	1.3 Alternativas de detección.	6
	1.4 Generación de la señal EMG.	7
	1.5 Aplicaciones.	8
	1.5.1 Neurología.	8
	1.5.2 Ergonomía.	9
	1.5.3 Fisiología deportiva.	9
	1.5.4 Rehabilitación médica.	9
	1.5.5 Otros campos de Aplicación.	9
	1.6 La fatiga muscular.	10
2.	Diseño del circuito impreso.	12
	2.1 Introducción.	13
	2.2 Especificaciones del sistema.	13
	2.3 Diagrama de bloques.	14
	2.4 Diseño de la etapa de entrada.	15
	2.5 Diseño de la etapa digital.	20
	2.6 Etapa de comunicación.	21
	2.7 Determinación de la etapa de alimentación.	23
	2.8 Layout del circuito impreso.	25
3.	Circuito impreso	26
	3.1 Introducción.	27
	3.2 Problemas encontrados y soluciones.	27
	3.3 El circuito impreso.	28
	3.4 Diagrama de Bode del canal de entrada.	30
4.	Software del PIC y del PC.	31
	4.1 Introducción.	32
	4.2 Diagrama de flujo del programa del PIC.	32
	4.3 Aspectos técnicos de la programación del PIC.	33
	4.4 Diagrama de flujo del programa Labview.	34
	4.5 Consideraciones para la programación del PC.	36
	4.6 Interfaz gráfica de usuario.	37
5.	Resultados	38
6.	Conclusiones	41
7.	Perspectivas de futuro	43
	Bibliografía	45
	Anexo A – Trazado de pistas del circuito impreso (Layout)	46
	Anexo B – Programa del PIC	49
	Anexo C – Programa del PC	56
	Anexo D – Presupuesto	67

Capítulo 1:

La Electromiografía (EMG)

1.1 Introducción

El cuerpo humano genera diferentes tipos de señales biológicas, que han supuesto un gran interés para muchos investigadores, debido a su amplio campo de aplicación. La decodificación e interpretación de dichas señales, son objeto de trabajo de muchos fisiólogos e ingenieros. Las señales fisiológicas más empleadas son el electrocardiograma (ECG), el encefalograma (EEG), o el electromiograma (EMG), sobre el cual está basado este trabajo.

A lo largo de este primer capítulo, repasaremos brevemente su historia, como se generan dichas señales, describiremos los métodos de detección, enumeraremos las aplicaciones más importantes y finalmente explicaremos el análisis frecuencial para la detección de la fatiga muscular

1.2 Historia de la electromiografía.

El estudio en el campo de la electromiografía se remonta al año 1912, donde el investigador H.Piper [1] realizó en Alemania las primeras mediciones con un galvanómetro. Doce años después, los investigadores Gasser y Erlanger [1] realizan estudios similares con un osciloscopio. En 1928, Proebster [1] observó las señales generadas por músculos enervados, dando así inicio al campo de la electromiografía clínica. A partir de este momento, son muchos los investigadores que aportan importantes contribuciones a este campo. Kugelberg, Petersen, Buchthal, Guld, Gydkov, Kosarov, Pinelli, Rosenfalk, Willison y Stalberg, [1] son los introductores del análisis cuantitativo del potencial de acción de unidad motora (MUAP). En 1964, Willison [1] introduce el análisis de la amplitud de señal EMG. En 1966, se fundó la ISEK (International Society of Electromyography and Kinesiology) que con su actividad a través de conferencias y congresos supuso un gran impulso para el progreso del campo de la electromiografía. La aparición de potentes ordenadores a finales de los 70 y principios de los 80, permite una rápida descomposición de la señal EMG en sus constituyentes MUAP's. Además, se desarrollan modelos y aproximaciones de simulación en el estudio del EMG. Los pioneros en el campo del modelado fueron Dimitrova y Lindstrom. [1] Entre los años 70 y 90 hubo multitud de publicaciones sobre la EMG contradictorias entre sí, creando así confusión entre los investigadores. Debido a esto, se funda en 1996, propuesto por la Unión Europea (UE), el SENIAM (Surface EMG for Noninvasive Assesment of Muscles) que pretende el consenso y la colaboración entre los diferentes laboratorios Europeos activos en este campo. A partir de aquí se llevan a cabo otros proyectos de la (UE), basados en EMG, como el PROCID (Prevention of Neuromuscular Disorders in the Use of Computer Input Devices) llevado a cabo entre el 1998 y el 2001, o el NEW (Neuromuscular Assessment of the Eldery Worker) entre el 2001 y el 2004, donde se emplearon técnicas de electromiografía no invasiva, para el estudio de problemas neuromusculares relacionados con la edad y el trabajo. Todos estos proyectos, han aportado importantes conocimientos sobre la electromiografía superficial. La aplicación de este tipo de técnica tiene muchos campos de aplicación, entre ellos el estudio de la fatiga muscular.

1.3 Alternativas de detección

Existen dos métodos para detectar las señales EMG, la invasiva y la no invasiva. El método invasivo emplea electrodos intramusculares mientras que el método no invasivo utiliza electrodos superficiales. En la figura 1.3 observamos diferentes tipos de electrodos. En la figura 1.4 observamos el tamaño típico de una unidad motora (MU) a escala con la imagen anterior.

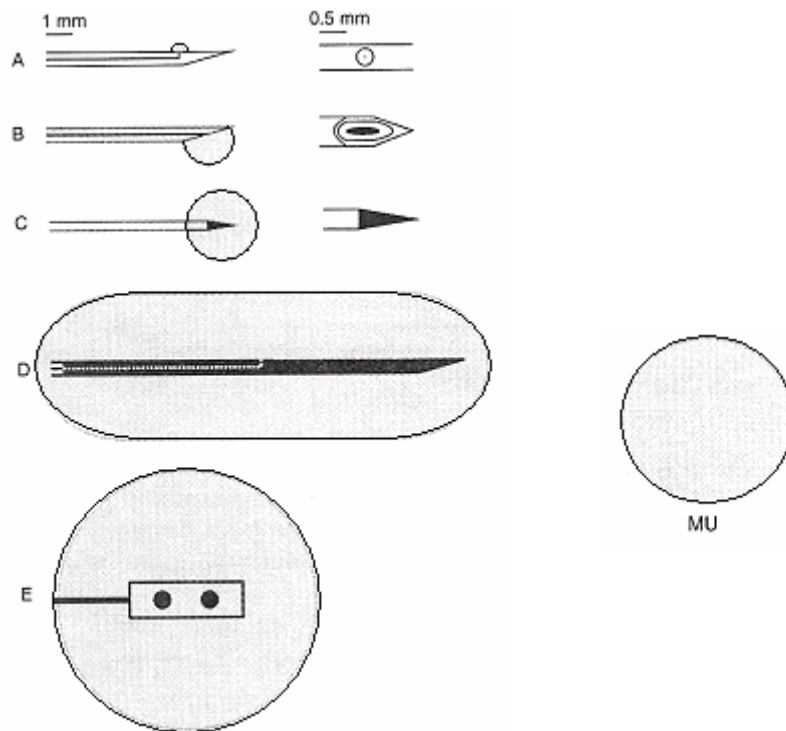


Figura 1.3 Diferentes tipos de electrodos, **Figura 1.4** Representación del tamaño típico de una MU respecto a los electrodos de la figura 1.3 [1]

Del A al C, en la figura 1.3, vemos electrodos de aguja, que por su tamaño suelen ser usados para descomponer señales EMG en MUAPs y detectar así trastornos miógenos o neurógenos. En el caso D, vemos un electrodo macro que se suele utilizar para captar la actividad de todas las fibras de una unidad motora. Esto permite diagnosticar y monitorizar el tamaño de las unidades motoras en los procesos que cursan con contracción y relajación. Finalmente observamos el caso E, donde vemos un electrodo de superficial, que debido a su tamaño se suele usar para observar la señal EMG de manera global, estudiando así aspectos como la fatiga muscular, o el tipo de movimiento, entre otros.

Por otro lado, cabe tener en cuenta que estos electrodos deben estar situados a una cierta distancia respecto a la fuente que genera la señal EMG. Los tejidos que separan dichas fuentes de los

electrodos se conocen con el nombre de conductores de volumen, y sus propiedades determinan las características de las señales detectadas. Uno de los efectos importantes, es el que estos tejidos actúan como un filtro pasa bajas en el dominio frecuencial (distribución de potencia), aunque no en el temporal. Visto esto, es lógico pensar que dicho efecto será mucho más notable en los electrodos superficiales que en los intramusculares, debido a que en estos últimos se reduce la distancia respecto a la fibra muscular. Este es un factor importante a tener en cuenta a la hora de decidirse por un tipo de electrodo u otro.

1.4 Generación de la señal EMG

Las señales EMG (también conocidas como MUAPs) son señales eléctricas producidas por un músculo durante el proceso de contracción y relajación. Las MUAPs son la suma de la actividad eléctrica de todas las fibras musculares que forman parte de una unidad motora (MU). La MU se compone básicamente de una neurona motora (motoneurona alfa) del asta anterior de la médula, su axón y fibras musculares, cuyo número dependerá del tipo de músculo. La neurona motora es la encargada de emitir el impulso que provocará la contracción del conjunto de fibras, y de esta manera, provocar el movimiento del músculo. La figura 1.1 muestra el esquema de una unidad Motora.

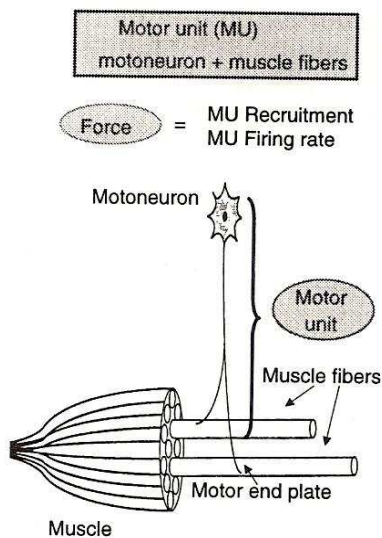


Figura 1.1 Representación esquemática de la Unidad Motora (UM) y sus componentes. [1]

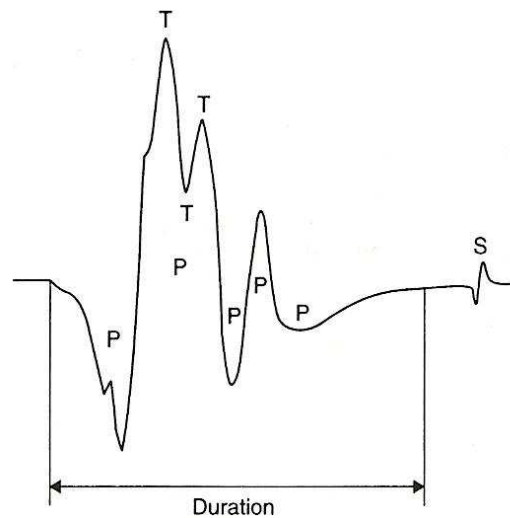


Figura 1.2 Esquema típico de un MUAP, donde se observa el pico principal y también: *P*. Cambio de fase. *T*. Giro de la señal. *S*. Componente espúreo generado por una fibra muscular enervada por un axón lento. [1]

Las principales características de estudio de los MUAPs (representado en la figura 1.2) son: la amplitud del pico principal, los cambios de fase de la señal, la duración y la estabilidad de la misma. Generalmente, la amplitud de la tensión que se puede medir está comprendida entre los centenares de μV hasta unos pocos mV , y el espectro frecuencial entre los 20 y los 500 Hz, concentrando la mayor parte de potencia entre los 50 y los 200 Hz. La duración se define como el tiempo desde la deflexión inicial al retorno a la línea de base, y suele tener unos valores medios de entre 5 y 15 ms. Finalmente, la estabilidad de un MUAP nos da una idea de la constancia de la señal en su morfología en sucesivas excitaciones.

Existen diversos factores que pueden provocar notables alteraciones de dichas características, los tres principales son: el tipo de acción que ejerce el músculo, la energía y oxígeno disponible en el metabolismo, y finalmente la fatiga muscular. Éste último aspecto es el objeto de estudio de este trabajo, que veremos con más detalle en el último apartado de este capítulo.

1.5 Aplicaciones

Como ya hemos comentado anteriormente, son muchas las líneas de investigación que se están llevando a cabo relacionadas con el campo de la electromiografía. Algunas de las aplicaciones más extendidas son las siguientes:

1.5.1 Neurología:

La neurología es la especialidad médica dedicada al diagnóstico y tratamiento de las enfermedades del sistema nervioso. El empleo de las diferentes técnicas que se realizan en el laboratorio de EMG permiten:

- 1) Distinguir entre lesiones del Sistema nervioso central (SNC) y del sistema nervioso periférico (SNP).
- 2) En patología neuromuscular, localizar y cuantificar diferentes tipos de lesiones con gran exactitud y precisión. Específicamente:
 - a) Lesiones de la neurona motora del asta anterior o del tronco (*neuronopatías motoras*) y de las neuronas del ganglio raquídeo posterior (*neuronopatías sensitivas*).
 - b) Lesiones de las raíces motoras o sensitivas (*radiculopatías*), de los plexos (*plexopatías*) y de los troncos nerviosos (*lesiones tronculares*).
 - c) Alteraciones de la *transmisión neuromuscular* y, dentro de ellas, distinción entre trastornos presinápticos y postsinápticos.
 - d) Trastornos primarios del músculo esquelético (*miopatías*)
- 3) Finalmente, una serie de técnicas, varias de ellas incorporadas al software de los electromiógrafos actuales, exploran la función del *sistema nervioso autónomo* (SNA) y sus trastornos.

1.5.2 Ergonomía.

La Ergonomía es un campo multidisciplinar que estudia las características, necesidades, capacidades y habilidades de los seres humanos, analizando aquellos aspectos que afectan al entorno artificial construido por el hombre. El objetivo de esta disciplina es el de adaptar los productos, las tareas, las herramientas, los espacios y el entorno en general a la capacidad y necesidades de las personas.

Dentro de este campo, encontramos la disciplina conocida con el nombre de ergonomía física, la cual trata temas como posturas en el trabajo, manejo manual de materiales, movimientos repetidos, lesiones músculo-tendinosas de origen laboral, diseño de puestos de trabajo, seguridad y salud ocupacional. Muchos de estos aspectos se estudian llevando a cabo técnicas de electromiografía.

1.5.3 Fisiología deportiva.

La fisiología deportiva es la disciplina que estudia cómo el ejercicio altera la estructura y funcionalidad del cuerpo humano. Dentro de este campo, las principales áreas de investigación basadas en el análisis del EMG son las siguientes:

- a) Descripción de la función normal del músculo durante diferentes movimientos y posturas.
- b) Investigación de la actividad muscular en deportes complejos.
- c) Estudios de coordinación y sincronización.
- d) Estudios para estimar la fuerza muscular y la fatiga.

1.5.4 Rehabilitación médica.

En el campo de la medicina de rehabilitación, las técnicas de análisis del EMG se utilizan en el estudio de diversas patologías, como las relacionadas con la zona pélvica de la mujer que ocasiona problemas como la disfunción sexual o la incontinencia. [1] La lumbalgia es otro de los desórdenes estudiado a veces mediante técnicas de análisis del EMG. También podemos encontrar otros campos de aplicación más inusuales como el estudio de cambios musculares debidos a altitudes elevadas o a la microgravedad inducida. [1]

1.5.5 Otros campos de aplicación.

Actualmente, se está trabajando en nuevas aplicaciones enfocadas a facilitar el día a día de personas con algún tipo de discapacidad o parálisis que dificulte sus actividades y su interacción con el mundo que le rodea. Por ejemplo, se están creando interfaces de comunicación entre el usuario y la máquina alternativos a los ya existentes, como son el teclado y el ratón. Un interfaz de este tipo permitiría a cualquier usuario (especialmente a aquellos que padezcan cualquier discapacidad) controlar sistemas informáticos y/o electrónicos mediante la contracción de determinados músculos.

Una de las aplicaciones que consiste en usar las señales EMG para que personas con un alto grado

de discapacidad física puedan desplazarse en silla de ruedas controlados por ellos mismos. [5] Para ello es necesario que el paciente disponga al menos de un músculo sano, que se utilizará para controlar el movimiento de la silla.

Otra posible aplicación es la de crear sistemas software que reproduzcan palabras de un menú en la pantalla de un ordenador conectado a un sistema de adquisición de señales EMG. Para ello se asocian determinadas señales de este tipo a los movimientos de un puntero en la pantalla del ordenador, mediante el que se escoge la palabra adecuada para ser reproducida. [5]

1.6 La fatiga muscular

La fatiga muscular es definida como la incapacidad del músculo para mantener la fuerza o potencia esperada. Las causas que la provocan se deben a diferentes alteraciones específicas dentro del propio músculo, entre las cuales cabe destacar el agotamiento de substratos energéticos (glucógeno muscular). Por otro lado, la fatiga también puede ser consecuencia de alteraciones a nivel de sistema nervioso central, en lo que se ha denominado como “fatiga central”. Esta hipótesis sugiere que un aumento de los niveles cerebrales de serotonina podría alterar las funciones del sistema nervioso central durante el ejercicio prolongado, disminuyendo así el rendimiento físico.

Los efectos de la fatiga muscular en las características de los MUAPs observados mediante técnicas de medición superficial, son los siguientes:

- Decremento de la densidad de potencia de señal.
- Desplazamiento del espectro de la señal hacia frecuencias inferiores a las normales.
- Aumento de la amplitud de la tensión eficaz (valor cuadrático medio de la tensión) de la señal EMG.

En estos efectos, que podemos observar en el ejemplo de la figura 1.5, nos basaremos a lo largo del trabajo, a la hora de realizar la estimación de la fatiga muscular.

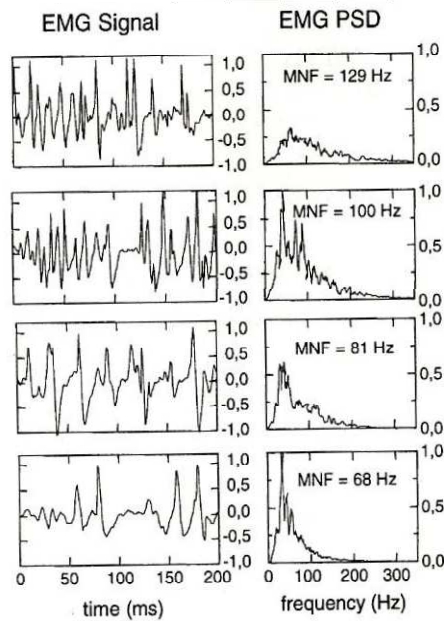


Figura 1.5 Medidas de señal EMG obtenidas a partir de una contracción isométrica del músculo tibial anterior, sostenida durante 100 s. Las medidas fueron obtenidas desde el inicio del ejercicio con intervalos de 30 s. [1]

En la figura anterior, se observa claramente como en el dominio temporal, aumenta la amplitud de señal debido al aumento en el reclutamiento de Mus, hasta que a partir de los 60 s se nota una reducción de la misma. En el dominio frecuencial, a medida que transcurre el tiempo, el contenido espectral de la señal se va desplazando hacia frecuencias inferiores, y además la densidad de potencia de la señal aumenta concentrándose cada vez en un ancho de banda más reducido.

Capítulo 2:

Diseño del circuito impreso

2.1 Introducción

A lo largo de este capítulo vamos a ver todos los aspectos relacionados con el diseño de la parte hardware del sistema para medición de fatiga muscular. En los dos siguientes subcapítulos veremos las especificaciones que el sistema deberá satisfacer y su diagrama de flujo, para acabar abordando en los subcapítulos siguientes el diseño de cada una de las partes que componen la PCB.

2.2 Especificaciones del sistema

Se pretende desarrollar un sistema portátil de bajo consumo que sea capaz de obtener 4 señales EMG de un paciente, acondicionarlas, muestrearlas y enviarlas por medio inalámbrico o a través de una conexión serie hacia un PC, para allí procesarlas mediante un programa diseñado específicamente para este proyecto y obtener así un estudio sobre la fatiga muscular del paciente.

En relación con lo visto en el capítulo anterior, y las funciones que debe implementar nuestro sistema, podemos definir lo siguiente:

- El sistema deberá ser capaz de amplificar señales de amplitud comprendida entre 100 μ V y 10 mV. Para ello utilizaremos un amplificador de instrumentación que proporcione un alto CMRR y una elevada ganancia.
- El sistema deberá filtrar las señales de entrada, que sobrepasen los 500 Hz y sean inferiores a 10 Hz. [1] Para ello se utilizará dos filtros Butterworth de orden 2 en cascada con estructura Sallen-Key, uno de ellos en configuración pasa bajas y el otro en pasa altas. De este modo conseguiremos el filtro pasa-banda que deseamos con la respuesta máximamente plana en la banda de paso.
- La frecuencia de muestreo de datos viene definida por 2 condiciones:
 - Para cumplir con el teorema de Nyquist, debemos muestrear la señal a una frecuencia como mínimo 2 veces superior al ancho de banda de la señal, por lo que al tener un ancho de banda de aproximadamente 500 Hz, la frecuencia de muestreo deberá ser como mínimo de 1 kHz.
 - De manera estándar en los aparatos de EMG, se suelen realizar barridos de 1024 puntos cada 100 ms, lo que implica una resolución temporal de 97,6 μ s/punto, por lo tanto, una frecuencia de muestreo de 10,46 Hz.

Para este proyecto nos centraremos en la primera de las condiciones, ya que dependiendo de la memoria de datos disponible en el microcontrolador que se elija podemos reducir el número de muestras o

aumentar el tiempo de barrido con el fin de reducir la frecuencia de muestreo.

- El sistema deberá tener una parte analógica con la cual se realizará el acondicionado de las señales que se obtendrán del paciente, y una parte digital con la que se digitalizarán dichas señales y se enviarán hacia el PC. Ambas partes deberán ubicarse en la PCB de modo que se interfieran mutuamente lo menos posible.
- Al tratar con señales de baja amplitud, entre 100 μV y 10 mV, en las que sus variaciones serán aún menores, deberemos garantizar una resolución de digitalización del sistema del orden de 1 μV para poder detectar dichas variaciones.
- Se deberá implementar una conexión RS-232 aislada mediante la cual podamos comunicarnos con el PC en caso de fallo de la conexión inalámbrica, sin que exista peligro para el paciente de inyectar una corriente eléctrica demasiado elevada a través del PC.

2.3 Diagrama de Bloques

Una vez tenemos claras las especificaciones que debe cumplir nuestro sistema, pasaremos a abordar su diseño. El sistema implementará cuatro canales de entrada, cada uno con dos electrodos superficiales con los que obtendremos las señales del paciente que se detectan y amplifican con un amplificador de instrumentación diferencial que aportará una elevada ganancia. Posteriormente pasarán por una etapa de filtrado pasa bajas con frecuencia de corte de 500 Hz y finalmente por otra etapa de filtrado pasa altas con frecuencia de corte de 10 Hz.

Una vez acondicionada la señal, se digitalizará con un convertidor analógico digital (ADC) de 16 bits. Los convertidores analógicos digitales de los cuatro canales se comunican con el microcontrolador a través del bus SPI y éste se encargará de la comunicación con el PC a través de un módulo ZigBee o en su defecto, la conexión RS-232 aislada. En la figura 2.1 podemos ver el diagrama de bloques del sistema.

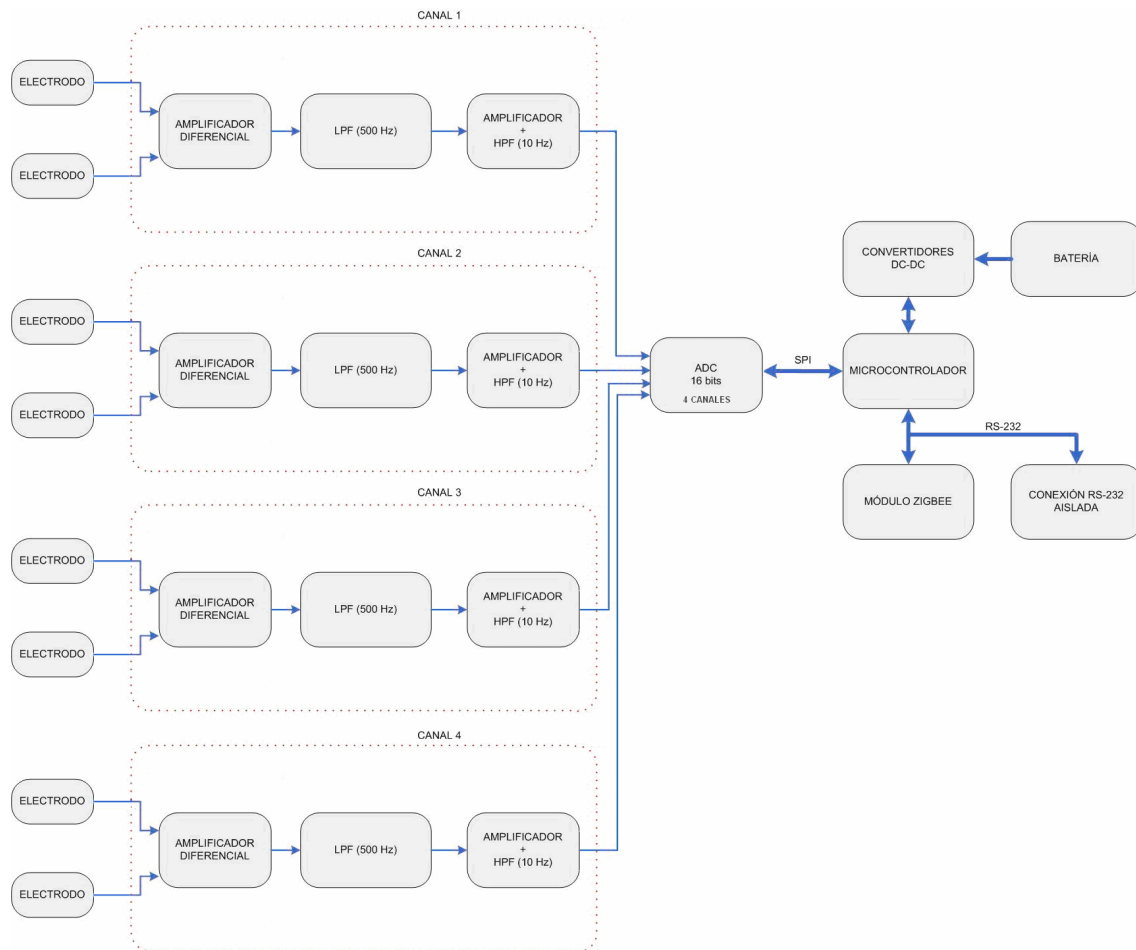


Figura 2.1 Diagrama de bloques del sistema.

2.4 Diseño de la etapa de entrada

Buffer de entrada: Para proporcionar una elevada impedancia de entrada y que la impedancia de los electrodos no nos afecte colocaremos un buffer formado por el amplificador operacional AD8627 en configuración de seguidor de tensión. Este amplificador operacional tiene una corriente de polarización máxima de 1 pA por lo que se polarizará con una resistencia de 10 MΩ colocada entre la entrada positiva y tierra.

Para evitar que se inyecte corriente continua al paciente se coloca un condensador en serie que además debe permitir el paso de la señal de 10 Hz. Para ello se ha elegido una C de 47 μF. Por cuestiones de seguridad se debe impedir que en caso de rotura del amplificador operacional la entrada quede unida a una de las tensiones de alimentación y el transitorio de corriente supere los 500 μA de pico. Para evitar esta situación se coloca una resistencia en serie con el condensador.

$$Resistencia = \frac{V_{cc}}{I} = \frac{3.3V}{500\mu A} = 6,6K\Omega$$

Por mayor seguridad, escogeremos una Resistencia = 10 KΩ que nos asegurará una corriente de entrada inferior.

Amplificador de instrumentación: El amplificador de instrumentación utilizado es el INA118U que tiene un bajo consumo y un ancho de banda variable con la ganancia. Tal y como indica la hoja de características del componente [2] el valor de la resistencia de ganancia (R_g) viene determinada por la ecuación 2.1:

$$G = 1 + \frac{50 \text{ k}\Omega}{R_g} \quad \text{eq 2.1}$$

Para obtener el valor máximo de resistencia a utilizar, hemos generado el esquemático del circuito amplificador mediante el programa ORCAD Capture y realizado un barrido en DC, obteniendo los resultados de la figura 2.2:

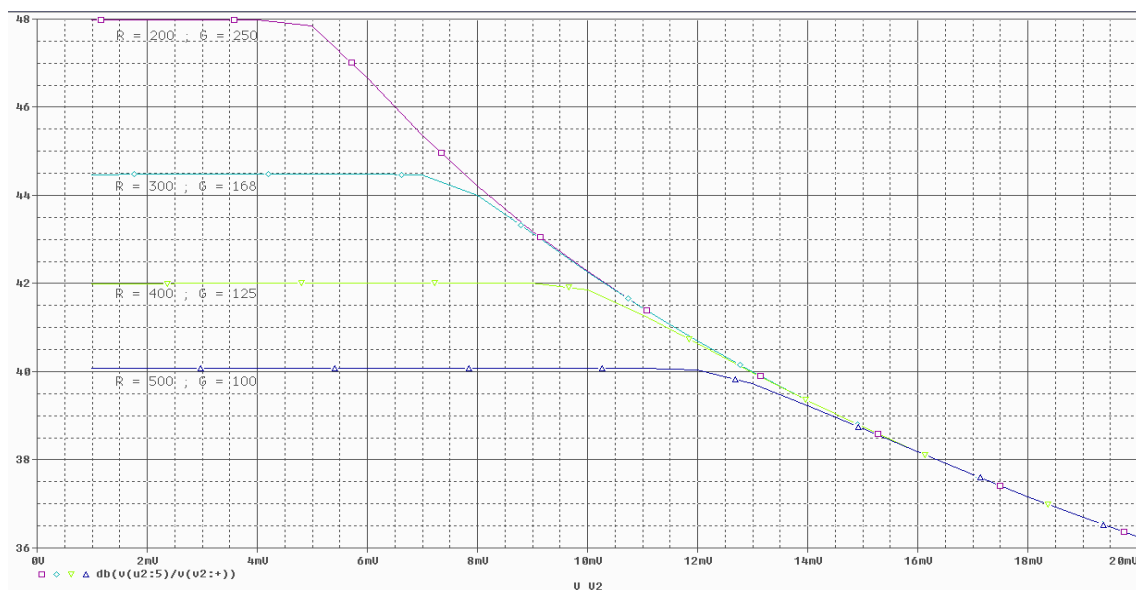


Figura 2.2 Ganancia Vs Tensión de entrada del INA118U

En la gráfica observamos como para ganancias inferiores a 42 dB ($R_g > 400\Omega$) se cumple la condición de ganancia plana hasta el límite de los 10 mV, de modo que hemos optado por una resistencia de 390 Ω . En este caso la ganancia obtenida según la ecuación eq 2.1 es: $G = 130$. De la hoja de características del componente [2] vemos como para $G = 130$ el ancho de banda es de aproximadamente 20 kHz.

Filtro pasa banda: La etapa de filtrado estará formada por 2 filtros Butterworth, uno pasa altas y el otro pasa bajas, que en cascada formarán un pasa banda con frecuencias de corte de 10 Hz y 500 Hz. Se ha elegido el filtro de tipo Butterworth ya que queremos una respuesta máximamente plana y la fase no es un aspecto relevante.

La topología elegida es la de Sallen-Key para ambos filtros, cuyo esquema general es el de la figura 2.3.

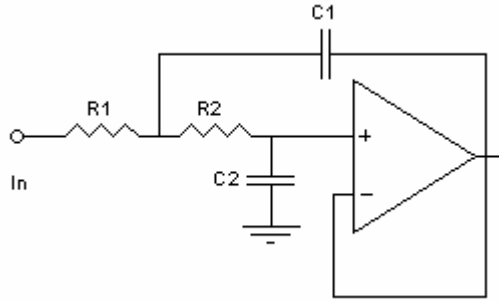


Figura 2.3 Filtro Butterworth con estructura Sallen-Key en configuración pasa bajas con ganancia 1.

Los parámetros del filtro están determinados por las ecuaciones 2.2 y 2.3:

$$F_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad \text{eq 2.2}$$

$$Q = \frac{\sqrt{R_1 R_2 C_1 C_2}}{C_2(R_1 + R_2)} \quad \text{eq 2.3}$$

Estableciendo $R_1 = R_2 = R$ y $C_1 = C_2 = C$ obtenemos:

$$Q = \frac{RC}{2RC} = \frac{1}{2} \quad \text{eq 2.4}$$

$$F_c = \frac{1}{2\pi RC} \quad \text{eq 2.5}$$

Estableciendo un valor de $R = 10 \text{ k}\Omega$ se obtiene:

$$C = \frac{1}{2\pi \times R \times F_c} = \frac{1}{2\pi \times 10000 \times 500} = 32 \text{ nF} \quad \text{eq 2.6}$$

Para el filtro pasa altas hemos utilizado la misma estructura que para el pasa bajas, intercambiando la posición de los condensadores y las resistencias. Debido a que en la etapa de amplificación podemos obtener una ganancia de 42 dB ($G=130$) como máximo, colocaremos las resistencias R_3 y R_4 para establecer la ganancia del filtro y obtener así como mínimo una ganancia total de 54 dB ($G = 500$). En la figura 2.4 vemos el esquema del filtro.

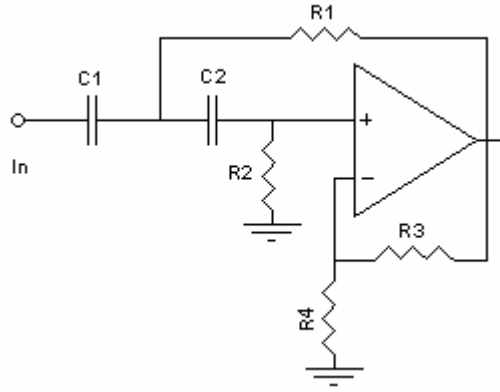


Figura 2.4 Filtro Butterworth con estructura tipo Sallen-Key en configuración pasa altas con ganancia.

Los parámetros del filtro para esta estructura las proporcionan las ecuaciones 2.7, 2.8 y 2.9:

$$F_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad \text{eq 2.7}$$

$$Q = \frac{R_2 C_1 C_2}{(C_1 + C_2)\sqrt{R_1 R_2 C_1 C_2}} \quad \text{eq 2.8}$$

$$G = \frac{V_0}{V_i} = 1 + \frac{R_3}{R_4} \quad \text{eq 2.9}$$

Estableciendo $R_1 = R_2 = R$ y $C_1 = C_2 = C$ obtenemos:

$$Q = \frac{RC^2}{2C \times RC} = \frac{1}{2} \quad \text{eq 2.10}$$

$$F_c = \frac{1}{2\pi RC} \quad \text{eq 2.11}$$

Seleccionando un valor de $R = 22 \text{ k}\Omega$ obtenemos que:

$$C = \frac{1}{2\pi \times R \times F_c} = \frac{1}{2\pi \times 22000 \times 20} = 362 \text{ nF} \quad \text{eq 2.12}$$

Teniendo en cuenta que en la etapa de amplificación conseguimos una $G = 130$, en esta etapa deberemos obtener una $G = 3,8$ como mínimo para obtener la $G = 500$ que se desea, por lo que de la ecuación 2.9 obtenemos la relación:

$$R_3 = R_4 \times (G - 1) = 2,8R_4$$

De modo que R3 debe ser 2,8 veces el valor de R4. Si establecemos $R_4 = 1 \text{ k}\Omega$ obtenemos que $R_3 = 2,8 \text{ k}\Omega$.

Con el fin de establecer con mayor precisión los valores calculados de resistencias y condensadores se ha generado el esquemático de la etapa de entrada completa. Al realizar diferentes barridos en frecuencia de la señal de entrada y comparándola con la señal de salida se ha comprobado como los valores de los componentes pasivos que mejor cumplen con las especificaciones son:

Amplificador de instrumentación: $R = 390 \Omega$.

Filtro pasa bajas: $R = 10 \text{ k}\Omega$ y $C = 22 \text{ nF}$.

Filtro pasa altas: $R = 22 \text{ k}\Omega$, $C = 1 \mu\text{F}$, $R_3 = 3,3 \text{ k}\Omega$ y $R_4 = 1 \text{ k}\Omega$.

Con estos valores obtenemos el diagrama de bode la figura 2.5:

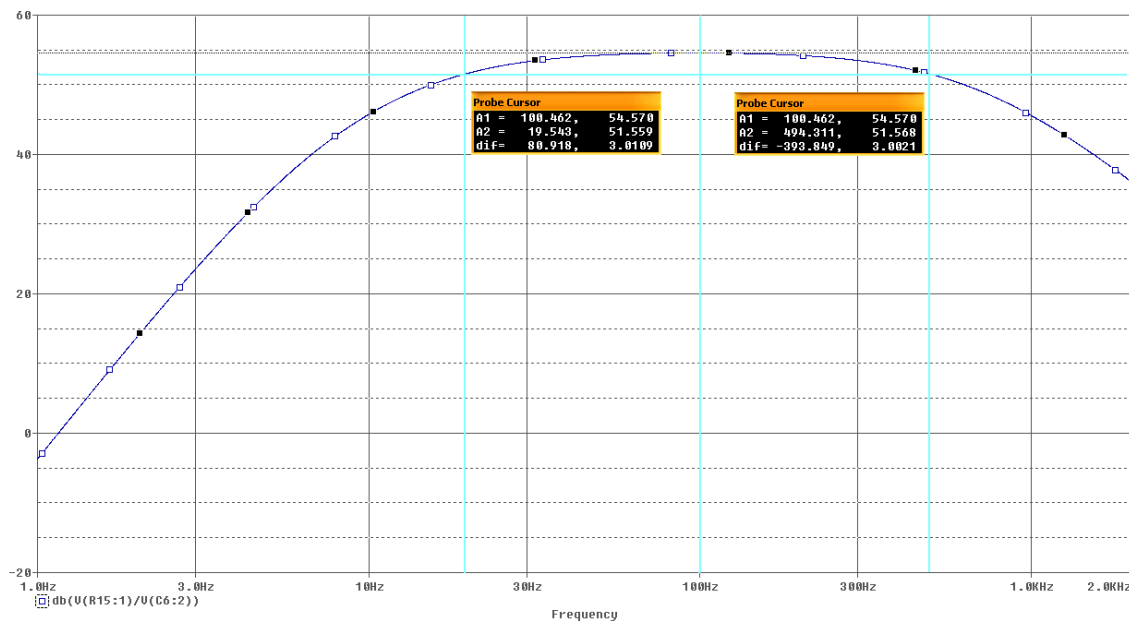


Figura 2.5 Diagrama de Bode del canal de entrada

En el diagrama vemos como en la zona del espectro que va desde los 50 Hz hasta los 250 Hz, que es donde se concentra la mayor parte de la densidad espectral de una señal EMG, la respuesta del canal de entrada es plana y proporciona una ganancia de 55 dB. Ésta se obtiene a partir de la ganancia del amplificador de instrumentación ($G = 130$) y del filtro pasa altas ($G = 4,3$) que en cascada nos proporciona una $G = 559$ (equivalente a 55 dB). Las frecuencias de corte (-3 dB) se encuentran en los 19 Hz y los 494 Hz, de modo que el diseño cumple con las especificaciones preestablecidas.

2.5 Diseño de la etapa digital

La etapa digital estará formada por el convertidor analógico digital y un microcontrolador. En el caso del conversor hemos elegido el ADS8343 de Texas Instruments que dispone de cuatro canales de entrada con una resolución de 16 bits y alimentado a 3,3 voltios. La tensión de referencia para el convertidor es de 1,65 voltios según se indica en la hoja de características del componente. [3] La referencia de tensión se consigue con el componente ISL60002DIH312Z (ver figura 2.9).

El siguiente paso es el de la elección del microcontrolador y para ello debemos decidir en primer lugar el fabricante que se desea utilizar. Para este proyecto nos hemos centrado en el fabricante Microchip.

Para elegir el modelo de PIC de entre la gran variedad existente, se han tenido en cuenta los siguientes factores:

- El PIC debía tener como mínimo SPI y EUSART como interfaces de comunicación para poder llevar a cabo ésta tanto con el convertidor AD como con el PC.
- Debía tener como mínimo 23 pins de entrada/salida (sin tener en cuenta los de alimentación, circuito de reset y circuito oscilador).
- Debía tener como mínimo 2048 Bytes de memoria RAM para poder almacenar los cuatro canales teniendo en cuenta barridos de 256 puntos de 16 bits por canal.
- Debíamos elegir un PIC con un tipo de encapsulado que fuera compatible con zócalo para poder así extraer el PIC de la placa cuantas veces necesitemos para llevar a cabo su programación.
- Para finalizar, debíamos elegir un PIC que tuviera suficiente memoria de programa.

Finalmente, el microcontrolador elegido ha sido el PIC18F2685 con encapsulado tipo SOIC de 28 pines.

El diseño del circuito del microcontrolador se puede ver en la figura 2.6, donde además del circuito de reset y el oscilador, se ha añadido un diodo led que ayuda a la depuración del programa.

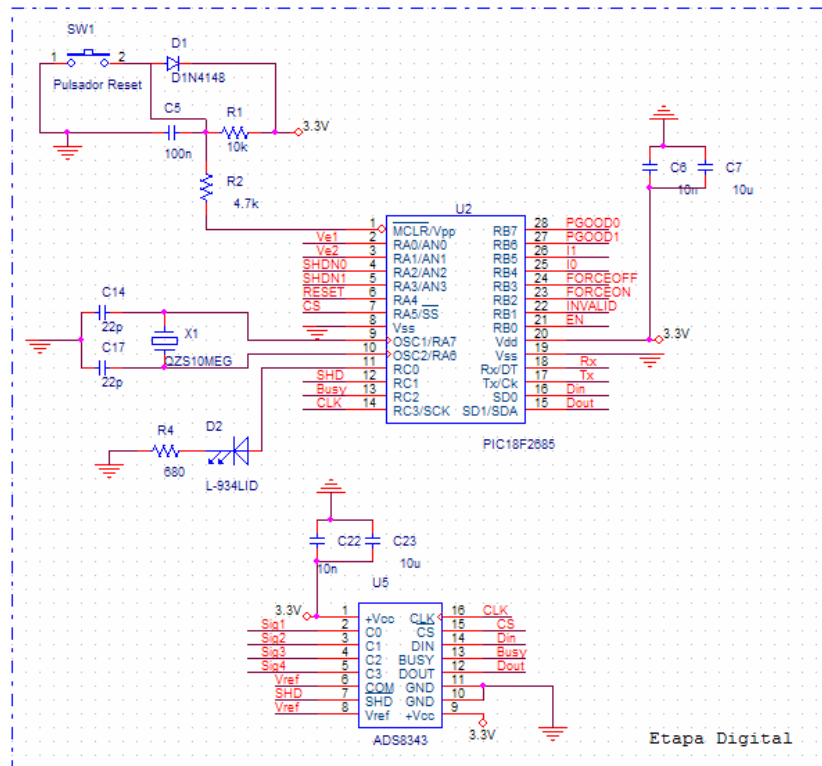


Figura 2.6 Etapa digital formada por el AD de 16 bits y el PIC de 8 bits.

2.6 Etapa de comunicación

Una de las ideas principales de este proyecto es la de realizar la comunicación entre la PCB y el PC en modo inalámbrico a través de un módulo ZigBee. Aún y así, se ha pensado en desarrollar juntamente con la primera, una segunda opción más económica que nos pueda servir también en caso de no obtener resultados satisfactorios con el módulo Zigbee, de modo que se ha optado por una conexión aislada por puerto serie.

En primer lugar, el módulo Zigbee elegido ha sido el TG-ETRX2-PM-001-104 de Telegesis debido a su reducido tamaño y consumo, fácil utilización a través de comandos AT y tensión de alimentación de 3,3 voltios. En la figura 2.7 podemos ver la representación esquemática de conexión del módulo ZigBee, el cual estará conectado al PIC a través de las señales Rx y Tx para enviar y recibir datos y la señal RESET para resetear el módulo.

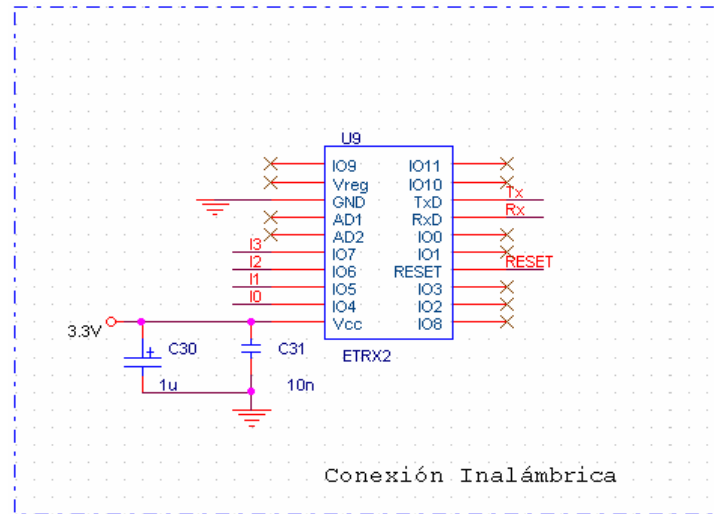


Figura 2.7 Esquema de conexión del módulo ZigBee

Por otra parte tenemos la conexión RS-232 aislada que está formada por 3 componentes principales (ver figura 2.8 para más detalle):

- El ADUM2402 que proporciona el aislamiento galvánico entre las señales del PIC y del PC.
- El SN65C3221PW que realiza la conversión de los niveles TTL de las señales del PIC a los niveles necesarios para la comunicación serie.
- La conexión de la placa con el PC se realiza mediante una conexión de modem nulo (null-modem) es decir, sin utilizar las señales de control RTS, CTS, DSR y DTR.

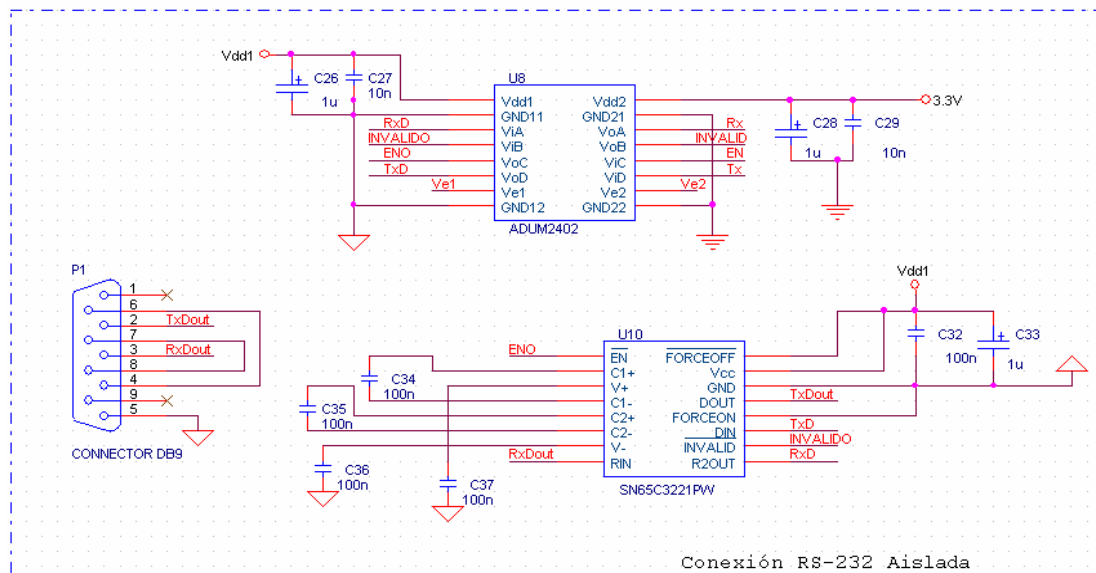


Figura 2.8 Esquemático de la conexión RS-232 aislada.

2.7 Determinación de la etapa de alimentación

Como hemos visto en el capítulo 2.1, el circuito debe tener una conexión RS-232 aislada que nos permita enviar los datos al PC de manera segura para el paciente. Para ello, necesitaremos dos alimentaciones completamente independientes, la primera para alimentar tanto a la parte analógica de entrada como la parte digital, y la segunda que alimente el aislador ADUM2402ARWZ y el SN65C3221PW (convertidor a niveles RS-232) que junto con el conector DB9 forman la conexión aislada.

Para determinar la capacidad de la batería tenemos que determinar el consumo de cada una de las partes, tanto del circuito aislado como de la parte no aislada. Además, añadiremos un factor de un 20% sobre la corriente calculada.

En la tabla 2.1 podemos ver el estudio de consumos:

Componente	Cantidad	+3,3 V aislada	-3,3 V aislada	3,3 V no aislada
INA118U Instrumentation amplifier	4	1,540 (0,385 x 4)	1,540 (0,385 x 4)	-
OPA2244UA Dual operational amplifier	8	0,560 (0,070 x 4)	0,560 (0,070 x 4)	-
ADS8343EBG4 - 16 bits, ADC	1	2,000	-	-
AD8627ARZ Operational amplifier	8	7,200 (0,900 x 4)	7,200 (0,900 x 4)	-
TC7660SEOA - VOLT CONVERTER, 7660, SOIC-8	1	0,080	-	-
ISL60002DIH312Z - REF 1.25V LOW POWER FGA, SOT-23-3	1	0,0009	-	-
MCP1252-33X50I/MS Charge pump regulator, 3.3V, 120mA	2	0,240 (120 x 2)	-	-
Modulo ZigBee ETRX2	1	41,500	-	-
PIC18F2685-I/SP - 8 bits microcontroller	1	39,000	-	-
Led	1	3,000	-	-
ADUM2402ARWZ Digital isolator, channels	4	11,200 mA (2,8 x 4)	-	5,600 (1,4 x 4)
SN65C3221PW Driver/receiver RS-232 SMD	1	-	-	1,000

+3,3 V aislada	-3,3 V aislada	3,3 V no aislada
106,32 mA	9,3 mA	6,6 mA

Corriente total: 122,221 mA	Corriente total + 20%: 146,665 mA
------------------------------------	--

Tabla 2.1 Estudio de consumos del circuito.

Si establecemos que la batería debe ser capaz de alimentar el circuito durante una jornada de trabajo de 8 horas obtenemos:

$$\text{Capacidad} = 146,665 \text{ mA} * 8 \text{ horas} = 1173 \text{ mAh}$$

De modo que la alimentación del circuito aislado se realizará mediante una batería de iones de litio con una tensión nominal de 3,7 Voltios y una capacidad de 1080 mAh.

La alimentación de 3,3 Voltios se obtendrá con el regulador MCP1252-33x50 y a partir de ésta y con el inversor TC7660SEOA se obtendrá la alimentación de -3,3 Voltios.

La figura 2.9 muestra el esquema de la parte de alimentación del sistema.

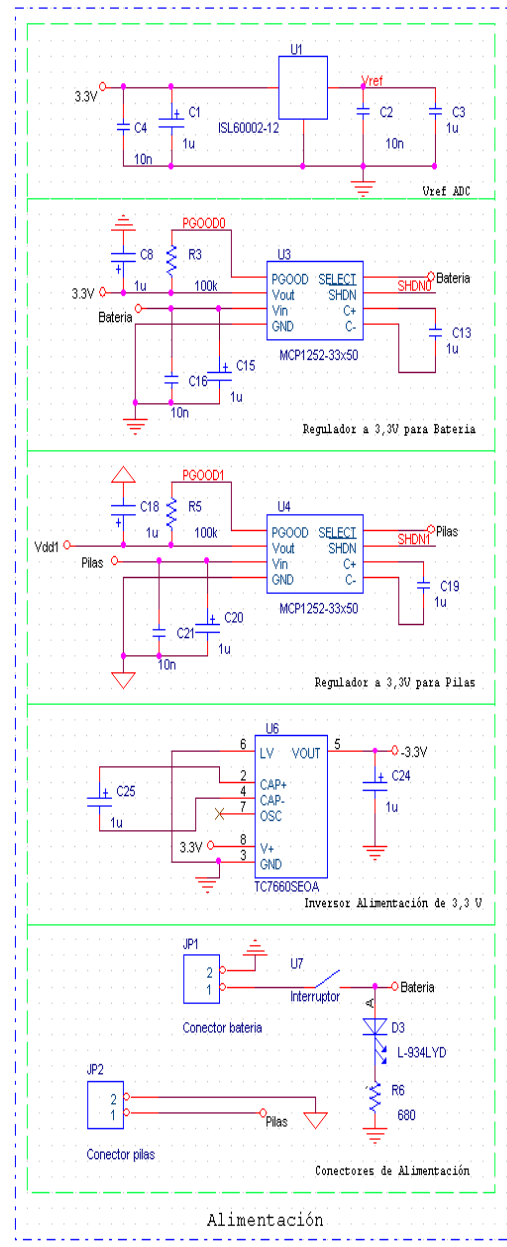


Figura 2.9 Esquema de la etapa de alimentación del sistema.

2.8 Layout del circuito impreso

Llegados este punto en el que ya hemos realizado el diseño completo de la parte hardware y sus esquemáticos en ORCAD el siguiente paso es generar el Netlist y posteriormente realizar el layout del PCB (ver Anexo A). Cuatro han sido los requisitos que hemos impuesto para el diseño del circuito impreso:

- Obtener un circuito impreso del menor tamaño posible. Para ello se ha realizado a doble cara. En general el patrón utilizado ha sido el de colocar los componentes activos en una de las caras y la segunda reservarla para componentes pasivos, aunque hay excepciones y se pueden encontrar de los 2 tipos en ambas caras.
- Separar claramente las 3 partes del diseño (Etapa de entrada analógica, etapa digital y etapa de comunicación) en la PCB.
- Crear 2 planos de masa separados y unidos en un único punto para la parte digital y la parte analógica, para evitar así la circulación de corrientes de alta frecuencia por la parte analógica generadas por la parte digital, es decir, evitar el acoplamiento de la parte analógica.
- Las capacidades de desacoplo de alimentación (10 nF y 1 μ F) se deben colocar lo más cerca posible al pin de alimentación de cada uno de los integrados.

Capítulo 3:

Circuito impreso

3.1 Introducción

En este capítulo vamos a exponer los diferentes problemas que nos hemos encontrado a la hora de realizar el montaje del circuito impreso, así como las soluciones por las que se ha optado. En el último apartado del capítulo veremos como se ha llevado a cabo el montaje del circuito impreso y finalmente mostraremos imágenes de la placa donde se identificarán cada una de las partes del circuito.

3.2 Problemas encontrados y soluciones

Debido a que el presupuesto del proyecto se excedía en cuanto a lo que el departamento de electrónica tenía destinado para cada alumno, se ha llevado a cabo un recorte en cuanto a los materiales del proyecto. Los recortes en cuestión son los siguientes:

- Se pasará de tener 4 canales de entrada a tener únicamente 2, por lo que prescindimos de:
 - 4 amplificadores operacionales ADS8627ARZ.
 - 2 amplificadores de instrumentación INA118U.
 - 4 amplificadores operacionales duales OPA2244UA
 - 52 condensadores.
 - 22 resistencias.
- Se utilizará únicamente la conexión RS-232 aislada como medio de comunicación con el PC, con lo que no dispondremos del módulo ZigBee ETXR2 y los 2 condensadores del pin de alimentación.

Estos recortes han supuesto un ahorro de 99,5 €, lo que supone prácticamente una reducción del 37% respecto al presupuesto inicial, ver presupuesto detallado en el Anexo D.

Debido al retardo en la aceptación del presupuesto reducido por parte de la ETSE, hemos tenido problemas con la fecha de entrega de algunos materiales por parte de Farnell, concretamente la batería de Litio y el convertidor analógico digital ADS8343EBG4. En el caso de la batería se ha optado por sustituirla por una pila de 9 Voltios. En el caso del convertidor se ha optado por sustituirlo por el modelo de 8 canales de entrada del mismo fabricante, ADS8345, [7] debido a que disponíamos de una muestra y su utilización no afectaba demasiado al diseño. Cabe destacar que todos estos inconvenientes han surgido con el circuito impreso ya fabricado, lo que provoca que finalmente la placa del circuito impreso sea más grande de lo que habríamos necesitado debido a que queda mucho espacio vacío. Además hemos tenido que realizar una pequeña placa anexa para el convertidor analógico digital de 8 canales, ya que éste tiene un encapsulado de 20 pines, mientras que el de 4 canales tenía tan solo 16.

Para la alimentación hemos tenido que añadir un nuevo regulador de 5 voltios para evitar disipar demasiada potencia en los reguladores montados.

Otro de los problemas que hemos tenido es que a la hora de probar el circuito impreso alimentamos directamente con 9 voltios el aislador ADUM2402ARWZ. Este hecho provocó la ruptura del componente. Debido a la falta de tiempo para conseguir un componente de repuesto hemos tenido que unir la parte aislada con la no aislada.

Para finalizar, nos encontramos con un problema relacionado con la etapa de comunicación. El convertor de niveles TTL a niveles RS-232, el SN65C3221PW, no funcionaba correctamente, por lo que se optó por utilizar un MAX232 para solucionar el problema.

3.3 El circuito impreso

El circuito impreso, cuyas imágenes de ambas caras podemos ver en la figura 3.1 y 3.2, está dividido en tres partes bien diferenciadas. De mitad de la placa hacia abajo podemos ver los cuatro canales de entrada de los cuales, como ya hemos comentado, tan solo se han utilizado dos. En la mitad superior de la placa podemos ver en la parte derecha la conexión RS-232 aislada y conectado al conector DB9 el módulo con el MAX232 con el que hemos solucionado el problema del convertor SN65C3221PW. En la parte izquierda tenemos el PIC18F2685 con el circuito de reset, el circuito del led y el oscilador. Debajo del microcontrolador podemos ver la parte de alimentación que proporciona +3,3 y -3,3 Voltios.

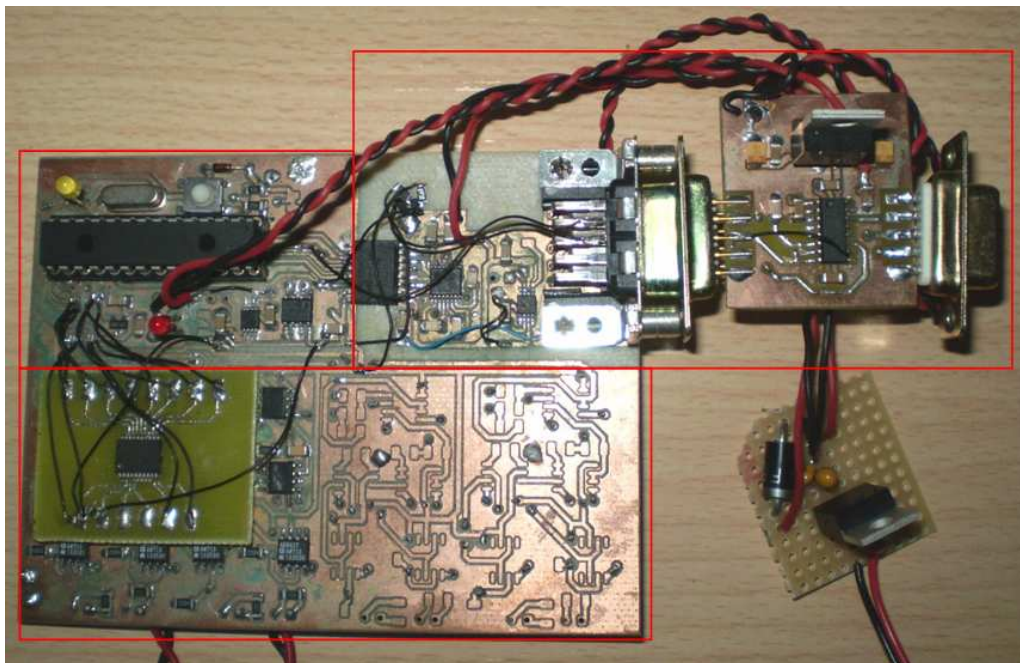


Figura 3.1 Cara superior del circuito impreso

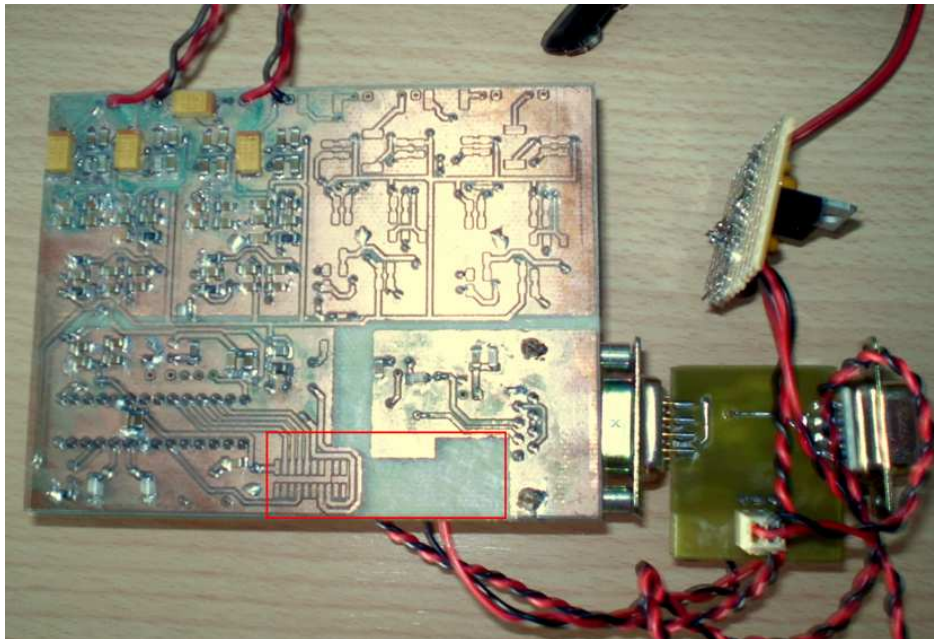


Figura 3.2 Cara inferior del circuito impreso

Podemos ver como la cara inferior se ha reservado principalmente para el trazado de pistas y la colocación de componentes pasivos, además de la zona reservada para el módulo ZigBee en la parte inferior derecha (enmarcado en rojo), que finalmente no ha sido utilizada.

En la figura 3.3 vemos una ampliación de los canales de entrada uno y dos, que en la figura 3.1 aparecen tapados por la placa anexa que contiene el convertidor analógico digital de 8 canales.

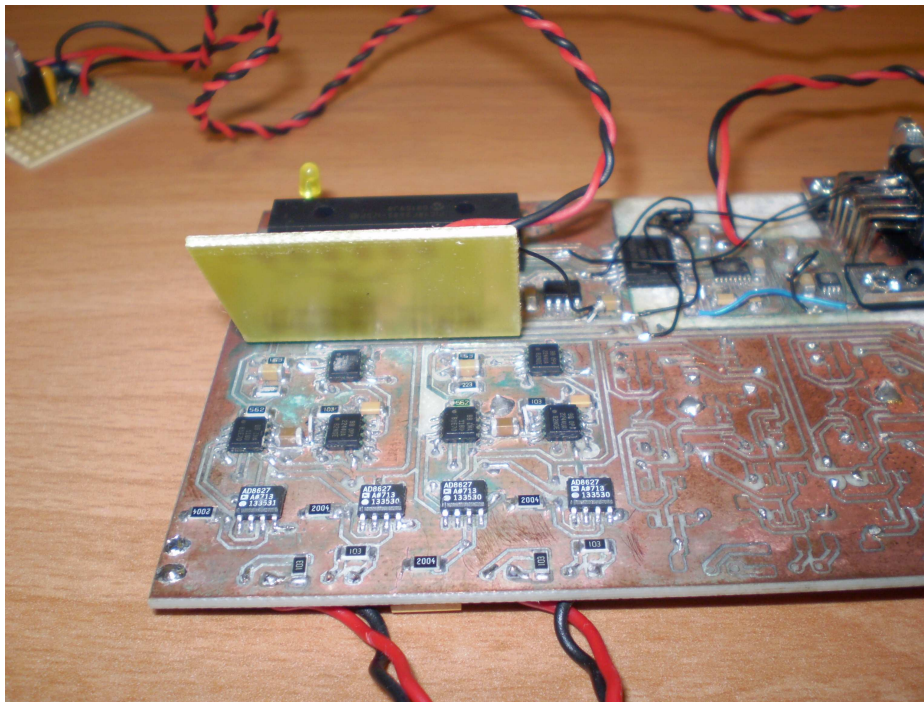


Figura 3.3 Canales de entrada

3.4 Diagrama de Bode del canal de entrada

Tras el montaje de la placa se ha realizado el diagrama de Bode de los dos canales de entrada utilizados, con el fin de comprobar su correcto funcionamiento. Para ello hemos utilizado un analizador de espectro realizando un barrido en frecuencia desde 1 Hz hasta 10 kHz. Los diagramas de Bode se han realizado dibujando la relación entre la salida y la entrada ($20 \cdot \log(V_o/V_i)$) y corresponden a la figura 3.4.

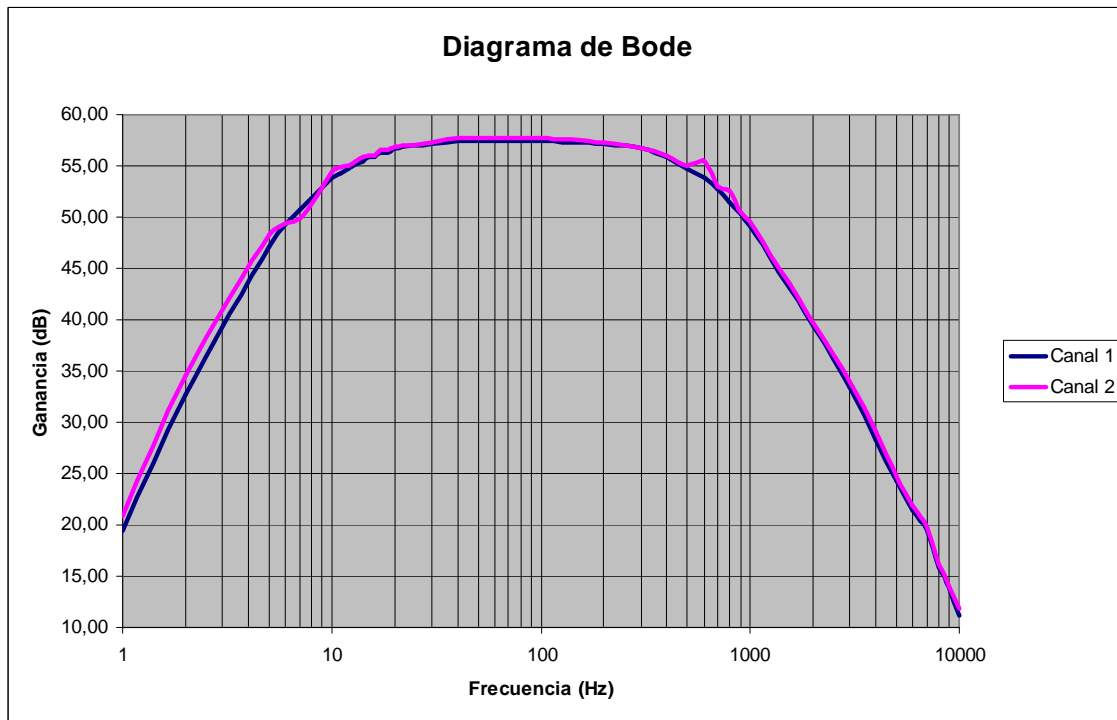


Figura 3.4 Diagrama de Bode del canal 1 y canal 2.

Podemos observar como el diagrama de Bode de ambos canales es prácticamente el mismo y además hemos cumplido con las especificaciones ya que tenemos una ganancia de 58 dB en la banda de paso y unas frecuencias de corte de 12 Hz y 500 Hz.

Capítulo 4:

Software del PIC y del PC

4.1 Introducción

En este capítulo vamos a comentar tanto el programa del microcontrolador, implementado en C, como el programa del PC, Implementado en Labview, de National Instruments, así como sus diagrama de flujo, para comprender el funcionamiento de cada una de las partes y la interacción que se produce entre ellas en el momento de la ejecución de los programas. También repasaremos los aspectos técnicos más relevantes tanto para la programación del PIC como la del PC. El listado de código se puede observar en el anexo B y en el anexo C.

4.2 Diagrama de flujo del programa del PIC

Tras la inicialización del microcontrolador, el programa deberá esperar una trama de inicio por parte del PC que le indique que debe comenzar a obtener los datos del paciente. Para ello se deberá habilitar la recepción de datos a través del módulo EUSART.

El programa deberá tomar muestras de 16 bits para ambos canales en intervalos de 1 ms. Para ello deberemos configurar el módulo MSSP (Master Synchronous Serial Port) en modo SPI de manera que nos permita comunicarnos con el convertidor analógico digital y además configurar el 'Timer 2' (TMR2) para que genere interrupciones en intervalos de 1 ms.

Una vez se han obtenido 512 muestras (1.024 bytes) para cada uno de los canales, el programa deberá transmitir los datos, habiendo configurado antes la transmisión de datos por el módulo EUSART (Enhanced Universal Synchronous Receiver Transmitter) y deshabilitado las interrupciones del TMR2. Para ello deberá esperar una trama mediante la cual el PC le indicará que debe iniciar la transmisión. Una vez recibida, el programa del PIC enviará 2 tramas al PC para indicar qué canal va a transmitir y procederá a la transmisión de las 512 muestras. Al finalizar la transmisión deberá esperar a recibir la confirmación por parte del PC de que ha recibido correctamente las 512 muestras del canal correspondiente. La trama recibida será un byte de un valor concreto. En caso de que el byte recibido no sea el esperado por el programa se procederá a realizar la misma operación 9 veces más. Si ninguna de las veces coincide con el que espera, entendemos que el PC no ha recibido los datos correctamente, por lo que procedemos a realizar de nuevo el envío de las 512 muestras. En caso de que el byte recibido sea el esperado por el programa se procederá al envío del siguiente canal. Una vez el programa ha enviado los datos de los dos canales pasará a capturar de nuevo datos del paciente, entrando así en un bucle infinito.

La figura 4.1 muestra el diagrama de flujo del programa que cumple con todas éstas especificaciones.

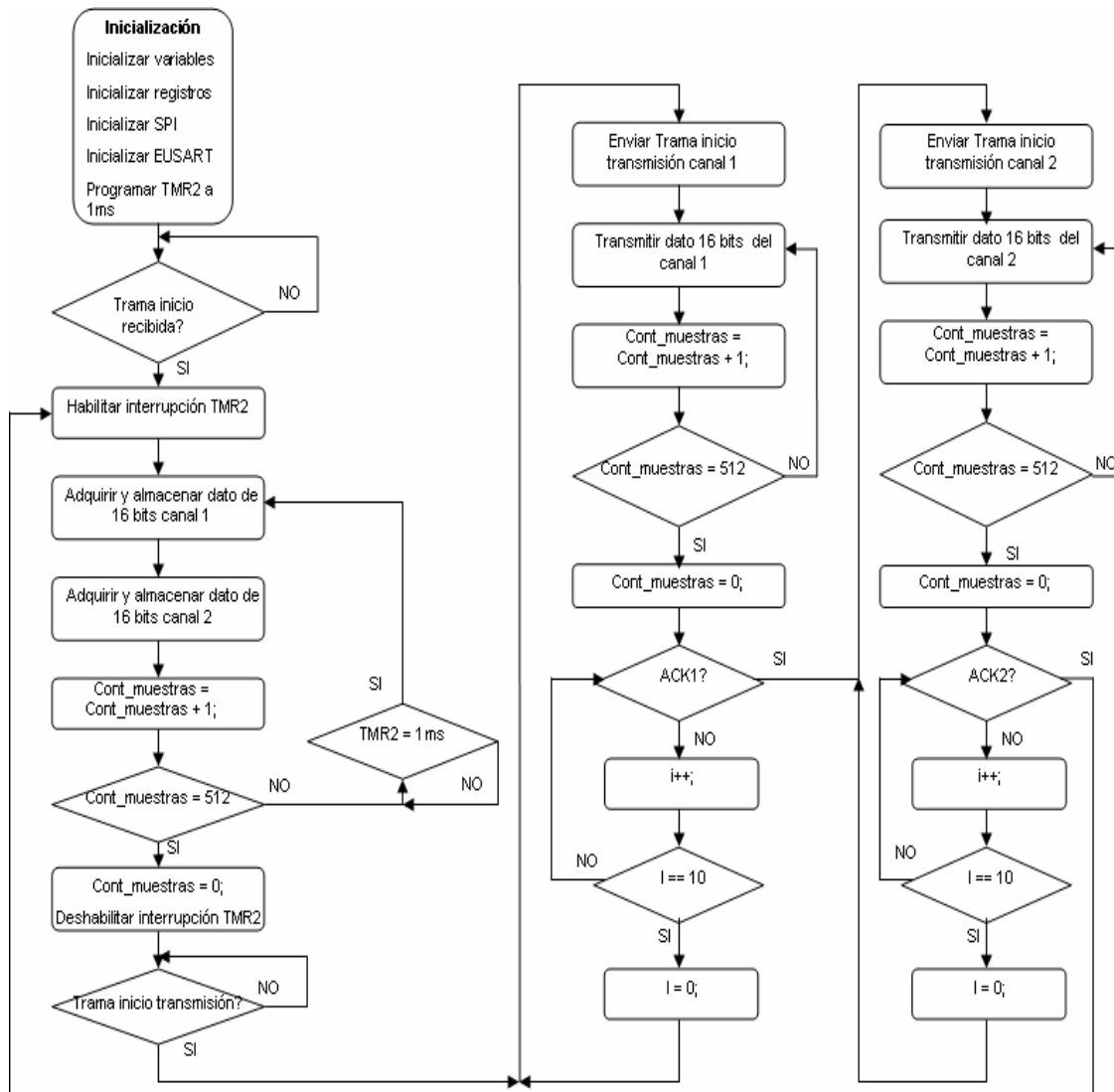


Figura 4.1 Diagrama de flujo del programa del PIC.

4.3 Aspectos técnicos de la programación del PIC

MSSP (modo SPI): Según la hoja de características del convertidor analógico digital [3] la frecuencia de trabajo máxima del dispositivo es de 2,4 MHz. La señal de reloj generada por el PIC no deberá exceder nunca de ese valor. Para ello debemos configurar el módulo MSSP para que genere una señal de reloj de $F_{osc} / 16$, donde F_{osc} es la frecuencia generada por el oscilador, de 10 MHz, por lo que el convertidor trabajará a 625 kHz.

TMR2 (Timer 2): El timer 2 actúa como un contador de 8 bits que genera una interrupción cuando el valor del contador coincide con el del registro PR2. Para establecer el valor de PR2 debemos tener en cuenta lo siguiente:

El PIC trabaja a una frecuencia 2,5 MHz, por lo que el tiempo de ciclo es $T_{ciclo} = 4 / F_{osc}$, es decir $T_{ciclo} = 0,4 \mu s$. Si utilizamos un 'prescaler' de 16, conseguimos que $T_{ciclo} = 6,4 \mu s$. Al establecer el registro PR2 con el valor decimal 157 conseguimos que la interrupción del timer se genere cada $6,4 \mu s * 157 = 1 ms$.

EUSART: Se ha configurado en modo asíncrono con transmisión de datos de 8 bits, sin bit de paridad y con las interrupciones tanto de transmisión como de recepción deshabilitadas. Se ha configurado la velocidad de transmisión a 19.200 bps. Para ello hemos habilitado el modo de alta velocidad y hemos establecido el valor del registro SPBRGH:SPBRG a partir de la ecuación 4.1, obtenida de la hoja de características del PIC. [4]

$$Valor = \frac{F_{osc}}{16 \times Baudrate} - 1 \quad \text{eq 4.1}$$

De modo que el valor del registro es 31.

Recepción / Transmisión de datos de 16 bits: El PIC18F2685 es un microcontrolador de 8 bits, por lo que, tanto la recepción de datos del ADC como la transmisión de datos por el puerto serie se realizará de 8 bits en 8 bits. Debido a que la memoria RAM del PIC está dividida en 16 bancos de datos de 256 bytes cada uno, ha sido necesario modificar el archivo 'linker' consiguiendo así dos bancos de datos de 1.024 bytes donde almacenar las 512 muestras de 16 bits de ambos canales.

Tramas de señalización: Tanto para indicarle al programa que debemos iniciar la adquisición de datos como para indicarle que inicie la transmisión, le enviaremos el valor hexadecimal 55. Para indicarle al PC qué canal estamos transmitiendo utilizaremos un 77 hexadecimal para el canal 1 y un 78 hexadecimal para el canal 2. A su vez el PC nos devolverá un 50 hexadecimal para indicar que ha recibido correctamente el canal 1 y un 61 hexadecimal para indicar que ha recibido correctamente el canal 2.

4.4 Diagrama de flujo del programa Labview

El programa deberá permitir al usuario trabajar en dos modos diferentes. Un primer modo de lectura, donde el usuario podrá visualizar resultados obtenidos en anteriores pruebas, y un segundo modo donde el usuario realiza la adquisición de datos. Centrándonos en el segundo de los modos, tras la inicialización de diferentes variables, el programa deberá configurar una conexión a través del puerto serie. Una vez configurada se enviará la trama de inicio que le indique al microcontrolador que debe iniciar la adquisición de datos. Seguidamente enviamos la trama que le indica al microcontrolador que estamos preparados para recibir datos y pasamos a la espera de recibir la trama que nos indique qué canal nos va a enviar el microcontrolador más las 512 muestras de 16 bits (1.024 bytes). Tras comprobar a qué canal pertenecen los datos recibidos el programa deberá tratar la matriz de datos de modo que obtengamos una matriz de tensiones y podamos así visualizarlo en una gráfica. Al mismo tiempo se deberá realizar la FFT de la señal y visualizarla en otra gráfica. Para finalizar, el programa enviará una trama que le indique al microcontrolador que los datos se han recibido correctamente. También deberá permitir la opción de guardar los datos en un archivo para su posterior visualización.

La figura 4.2 muestra el diagrama de bloques del programa del PC realizado con Labview.

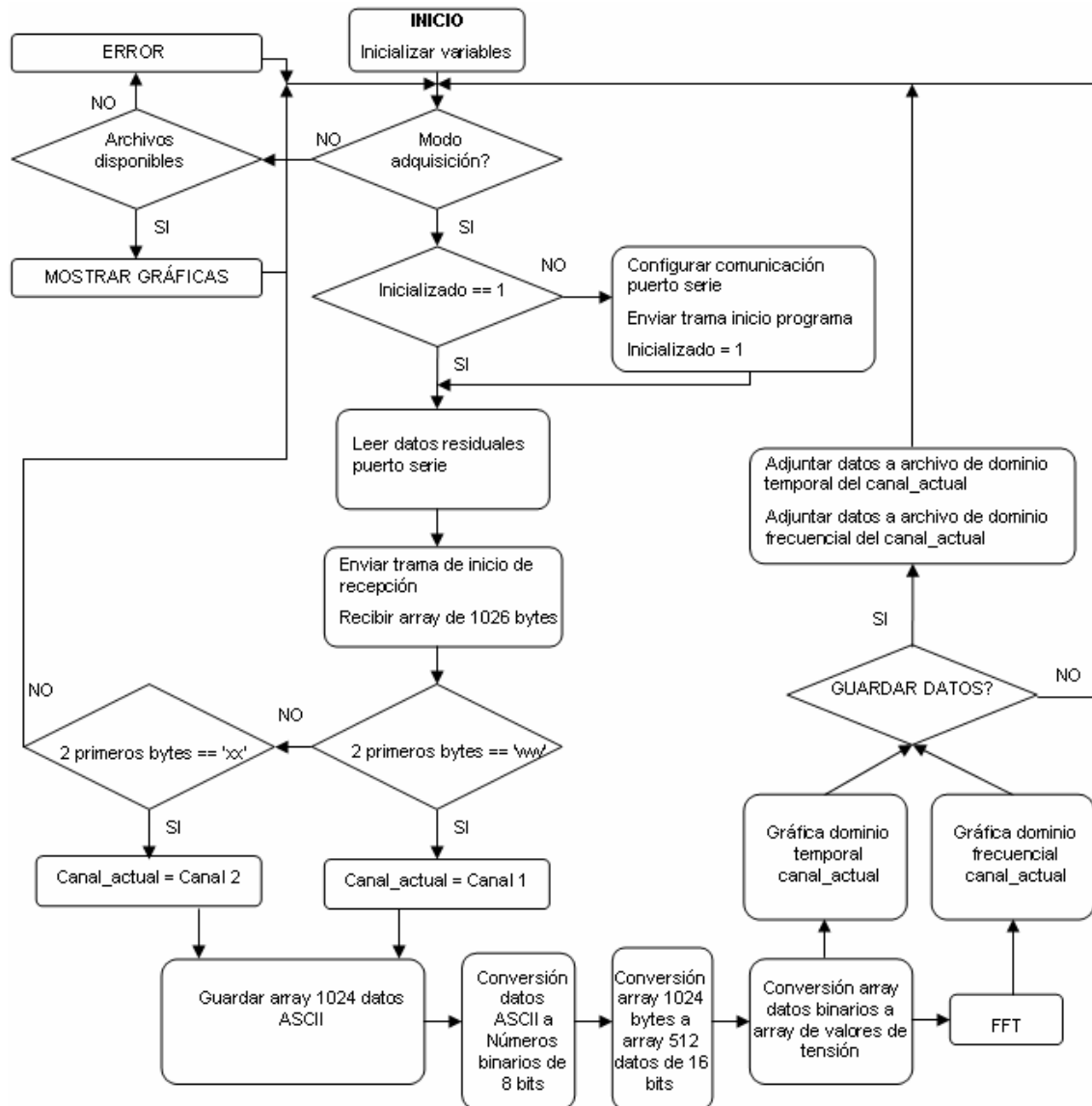


Figura 4.2 Diagrama de flujo del programa del PC.

4.5 Consideraciones para la programación del PC

Para configurar la conexión del puerto serie se ha realizado lo siguiente:

- Se ha configurado una velocidad de transmisión/recepción 19.200 bps.
- Hemos deshabilitado el final de carácter de trama.
- No utilizamos control de flujo.

Otro de los aspectos a tener en cuenta es el hecho de que los valores que queremos transmitir por el puerto serie se deberán escribir como una cadena de caracteres ASCII. La tabla 4.1 muestra la correspondencia entre las tramas utilizadas y el código ASCII correspondiente:

Valor de trama hexadecimal	Valor ASCII
55	U
77	w
78	x
50	P
61	a

Tabla 4.1 Tabla de correspondencias entre valores hexadecimales y valores ASCII

En cuanto a la conversión de datos de 16 bits a valores de tensión, primero convertimos la matriz de datos binarios a matriz de datos decimales. Posteriormente obtenemos la matriz de tensiones aplicando la ecuación 4.2:

$$V_{tensión} = \frac{Dato_Decimal}{65536} \times 3,3 V - 1,65 V \quad \text{eq 4.2}$$

Los valores recibidos se encuentran en el rango de 0 a 3,3 voltios, debido a que tienen un offset de 1,65 voltios. Al realizar la conversión de dato decimal a valor de tensión, le restamos el offset de manera que obtenemos valores comprendidos en el intervalo que va de -1,65 voltios a +1,65 voltios.

4.6 Interfaz gráfica de usuario

La figura 4.3 muestra la interfaz gráfica del programa del PC. En ella hay marcadas diferentes zonas de interés.

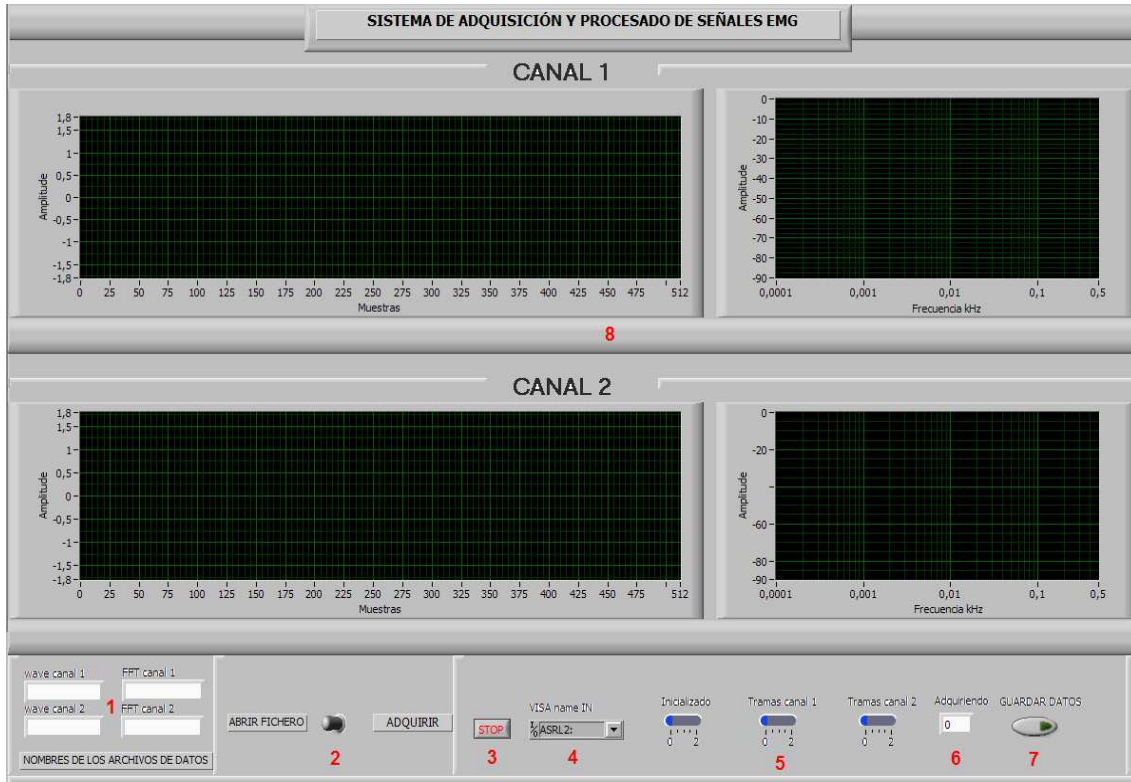


Figura 4.3 Interfaz gráfica de usuario del programa del PC.

En rojo vemos numeradas las diferentes zonas:

- 1: En esta zona vemos cuatro *Textbox* donde indicamos el nombre de los cuatro archivos que guardaremos, dos para guardar la señal temporal y dos para guardar las señales en dominio frecuencial de ambos canales.
- 2: Interruptor mediante el cual le indicamos al programa si deseamos ejecutar en modo lectura o en modo adquisición.
- 3: Botón de parada del programa.
- 4: Botón para indicar el puerto donde se conectará el circuito impreso.
- 5: Conjunto de indicadores que muestran si se reciben las tramas que indican el canal que está transmitiendo el microcontrolador.
- 6: Cuadro que nos indica el canal que se está leyendo.
- 7: Botón que guarda los datos adquiridos mientras permanece activo.
- 8: Zona de visualización de las señales temporales y frecuenciales de ambos canales.

Capítulo 5:

Resultados

5. Resultados

Se ha llevado a cabo una prueba utilizando tres electrodos superficiales colocados en el antebrazo como muestra la figura 5.1.

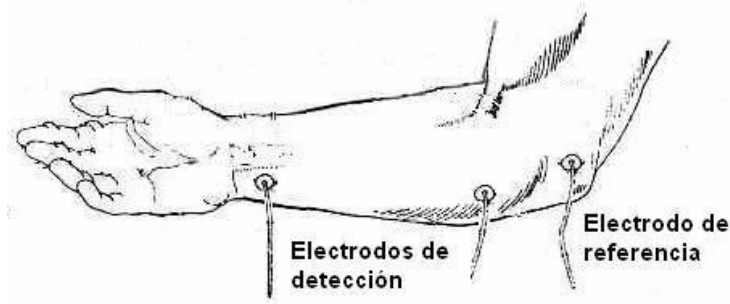


Figura 5.1 Colocación idónea de los electrodos superficiales para la detección de fatiga muscular. [6]

Durante la prueba, que ha durado varios minutos, el paciente ha realizado contracciones y relajaciones sucesivas del antebrazo. Al inicio de la prueba, las imágenes obtenidas por el programa Labview son las de la figura 5.2.

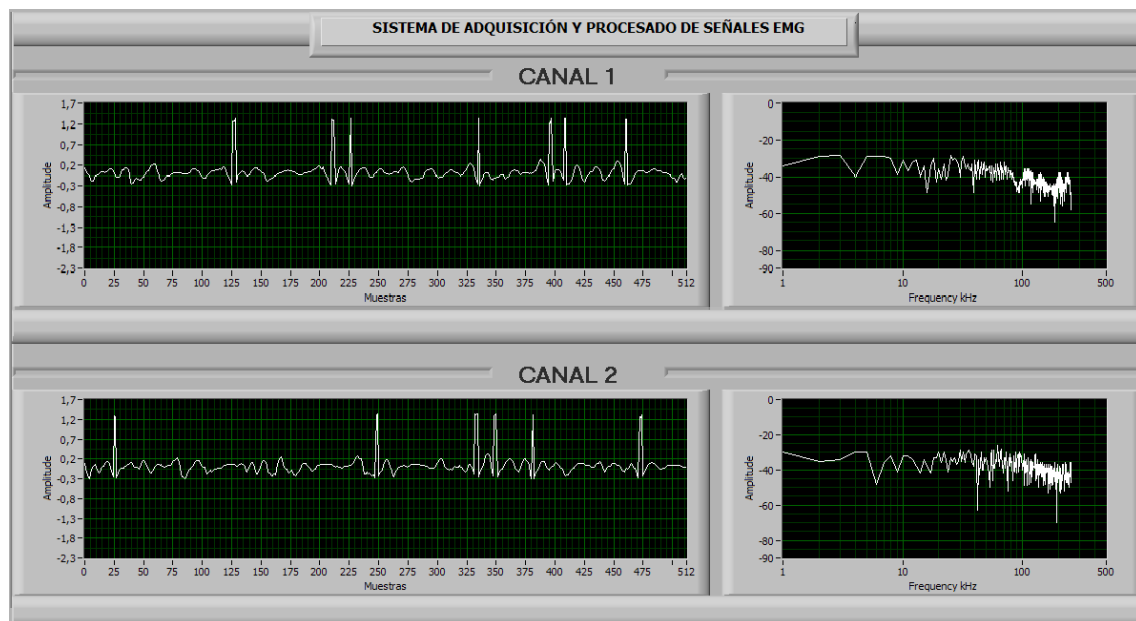


Figura 5.2 Resultados obtenidos al inicio de la prueba.

Se observa de manera clara como la mayor parte de la densidad espectral de la señal se concentra en las frecuencias que van de los 70 Hz a los 260 Hz. También vemos como el nivel de potencia se encuentra a niveles próximos a -40 dB.

En la figura 5.3 podemos obtener los resultados obtenidos al final de la prueba.

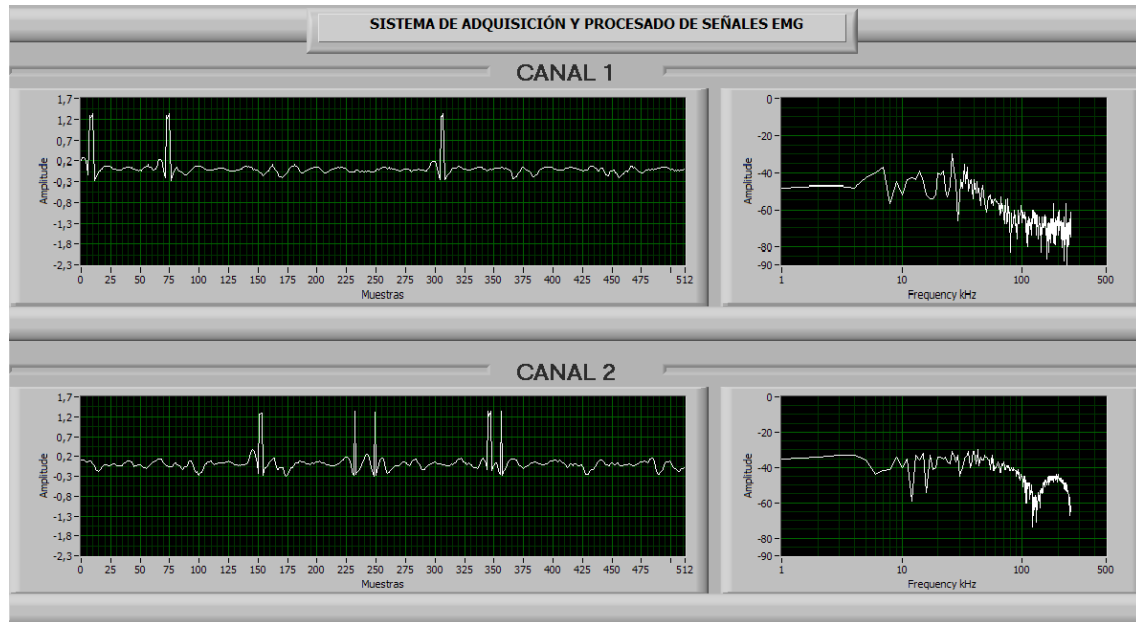


Figura 5.3 Resultados obtenidos al final de la prueba.

En este caso vemos que para las frecuencias más altas, entre 70 Hz y 260 Hz, el nivel de potencia ha descendido notablemente, sobretodo en el canal 1, cuya señal está aproximadamente 15 dB por debajo respecto al inicio de la prueba. También se observa como ha disminuido la densidad espectral en este rango, mientras que ha aumentado para las frecuencias inferiores a 70 Hz, por lo que consideramos que hemos detectado la fatiga del antebrazo tras pocos minutos de ejercicio.

Capítulo 6:

Conclusiones

6. Conclusiones

En primer lugar se ha realizado una pequeña introducción a la historia de la electromiografía y sus fundamentos. También hemos visto sus posibles aplicaciones, haciendo hincapié, sobretodo en el campo de la detección de fatiga muscular.

En segundo lugar hemos definido el sistema a implementar, empezando por el diagrama de bloques y diseñando posteriormente cada uno de ellos que son:

- La etapa de entrada, formada por un amplificador de instrumentación y un filtro pasa-banda con frecuencias de corte de 19 Hz y 494 Hz.
- La parte digital, formada por un convertidor analógico digital de 16 bits y un microcontrolador PIC18F2685.
- La etapa de comunicación formada por un módulo ZigBee y una conexión RS-232 aislada.
- La parte de alimentación que proporciona +3,3 voltios y -3,3 voltios.

Para ello se han realizado los esquemas eléctricos mediante el programa Orcad Capture. En el caso de la etapa de entrada se han realizado simulaciones que han permitido ajustar mejor los valores de los componentes calculados en la etapa de diseño. Posteriormente se ha realizado el trazado de pistas del circuito impreso con el programa Ultiboard.

Tras el proceso de diseño se ha llevado a cabo la fabricación del circuito impreso. Se han soldado todos sus componentes y posteriormente se ha realizado el diagrama de bode para los dos canales de entrada utilizados. Se ha comprobado como los resultados obtenidos corresponden con lo esperado según las simulaciones realizadas en la etapa de diseño.

Tras la fabricación del circuito se ha realizado el programa del PIC, en lenguaje C, que se comunica con el conversor de 16 bits para tomar 512 muestras para cada uno de los dos canales de entrada, transmitiéndolos posteriormente a un PC por el puerto serie. Estos datos son tratados por un programa que se ha implementado en Labview, que muestra el dominio temporal y frecuencial de las señales.

Para finalizar se ha llevado a cabo una prueba con tres electrodos superficiales colocados en el antebrazo de un paciente en la que hemos podido observar el efecto de la fatiga muscular.

Capítulo 7:

Perspectivas de futuro

7. Perspectivas de futuro

Tras la finalización de este proyecto son varias las líneas de desarrollo que han quedado abiertas. En primer lugar queda pendiente la utilización de un medio de comunicación inalámbrico entre el circuito impreso y el PC. Como se ha visto en el capítulo 3, el circuito impreso tiene una zona reservada para la inserción de un módulo ZigBee ETXR2, por lo que no resulta complicado llevar a cabo esta mejora. Por otro lado, existen dos opciones en cuanto a la etapa de entrada. Se pueden utilizar los 4 canales de entrada de modo que se observen más señales EMG y se obtenga una mejor detección de la fatiga del músculo, o bien, se puede desarrollar un nuevo circuito impreso con tan solo dos canales de entrada cuyo tamaño sea menor al fabricado para este proyecto. Además se le puede introducir un filtro notch de 50 Hz para eliminar la señal de línea por hardware. Otra de las tareas pendientes es la de sustituir el componente SN65C3221PW por el MAX232, de modo que se reduciría el tamaño del circuito impreso al poder prescindir de la placa anexa que se ha tenido que colocar junto con el conector DB9. Para finalizar, también se puede sustituir el convertidor analógico digital de 8 entradas por el de 4, o uno de dos entradas si se limita a dos el número de canales de entrada, ya que las restantes están inutilizadas.

Todas las mejoras citadas son de fácil ejecución, ya que no requieren grandes cambios ni en el circuito impreso ni en el propio sistema. Existe la posibilidad de llevar a cabo una mejora que puede resultar más complicada pero a la vez muy interesante. La mejora consiste en hacer el sistema completamente independiente, es decir, que el procesado de los datos no se lleve a cabo en un PC, sino en el propio circuito impreso a partir de un microcontrolador más potente que el utilizado, como podría ser un DSP. Para ello sería necesario profundizar sobre el tema de la alimentación, ya que el DSP ejecuta operaciones mucho más complejas que el PIC18F2685 y por lo tanto tiene un consumo mayor, y éste es un aspecto crítico para los sistemas embebidos.

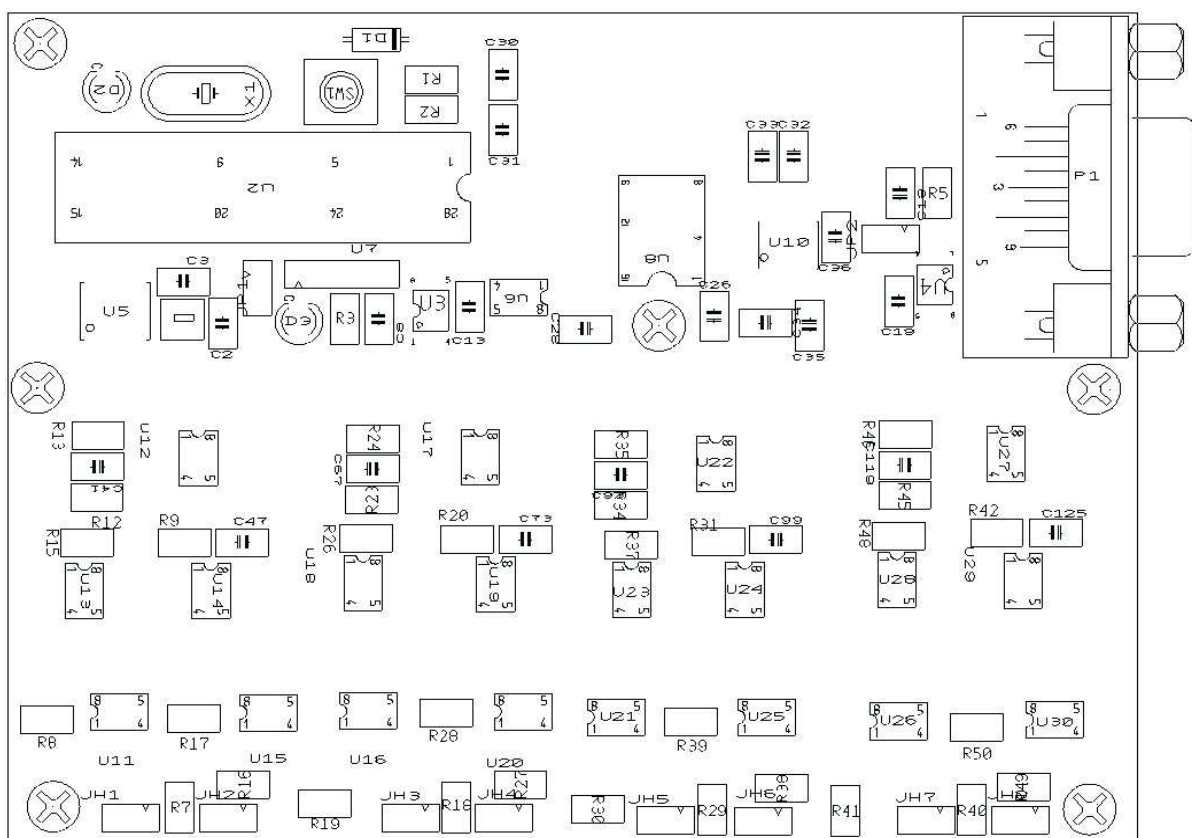
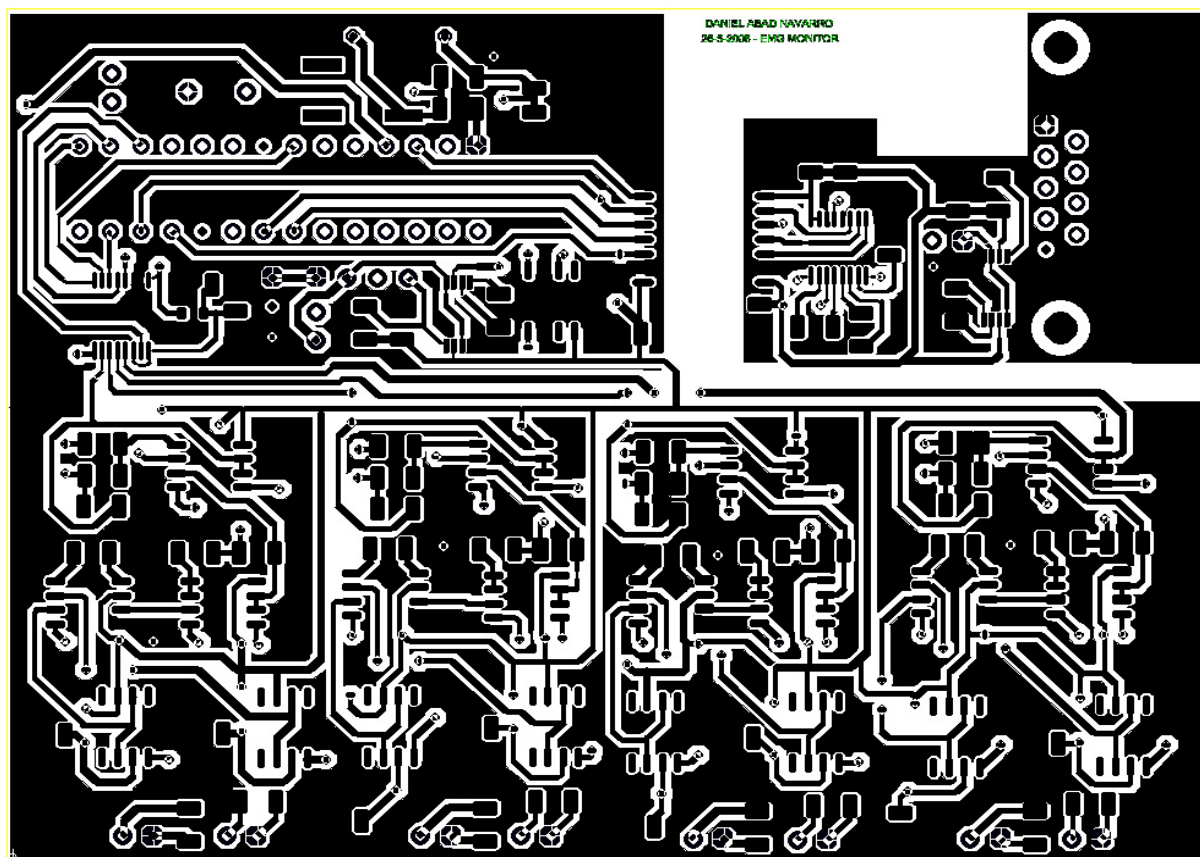
Bibliografía

- [1] Roberto Merletti. **Electromyography: Physiology, Engineering, and Non-Invasive Applications**. IEEE Press Series on Biomedical Engineering. Wiley-IEEE Press. ISBN: 0471675806
- [2] **Precision, Low Power Instrumentation Amplifier INA118U**,
(<http://focus.ti.com/docs/prod/folders/print/ina118.html>)
- [3] **ADS8343: 16-Bit, 4-Channel Serial Output Sampling Analog-to-Digital Converter**
(Rev. C) (<http://focus.ti.com/docs/prod/folders/print/ads8343.html>)
- [4] **PIC18F2682/2685/4682/4685 Data Sheet**,
(<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en026328>)
- [5] Rafael Ruiz Rubio. **Aplicaciones de las señales electromiográficas**,
(<http://www.encuentros.uma.es/encuentros53/aplicaciones.html>)
- [6] Edgar Muñoz Burbano, Oscar H. Paruma y Juan Fernando Florez, **APLICACIONES DE LAS SEÑALES MIOELECTRICAS PARA EL CONTROL DE INTERFACES HOMBRE-MAQUINA**. (http://ai.unicauca.edu.co/publicaciones/ASB_CIHM.pdf)
- [7] **ADS8345: 16-Bit, 8-Channel Serial Output Sampling Analog-to-Digital Converter**
(Rev. C) (<http://focus.ti.com/docs/prod/folders/print/ads8345.html>)

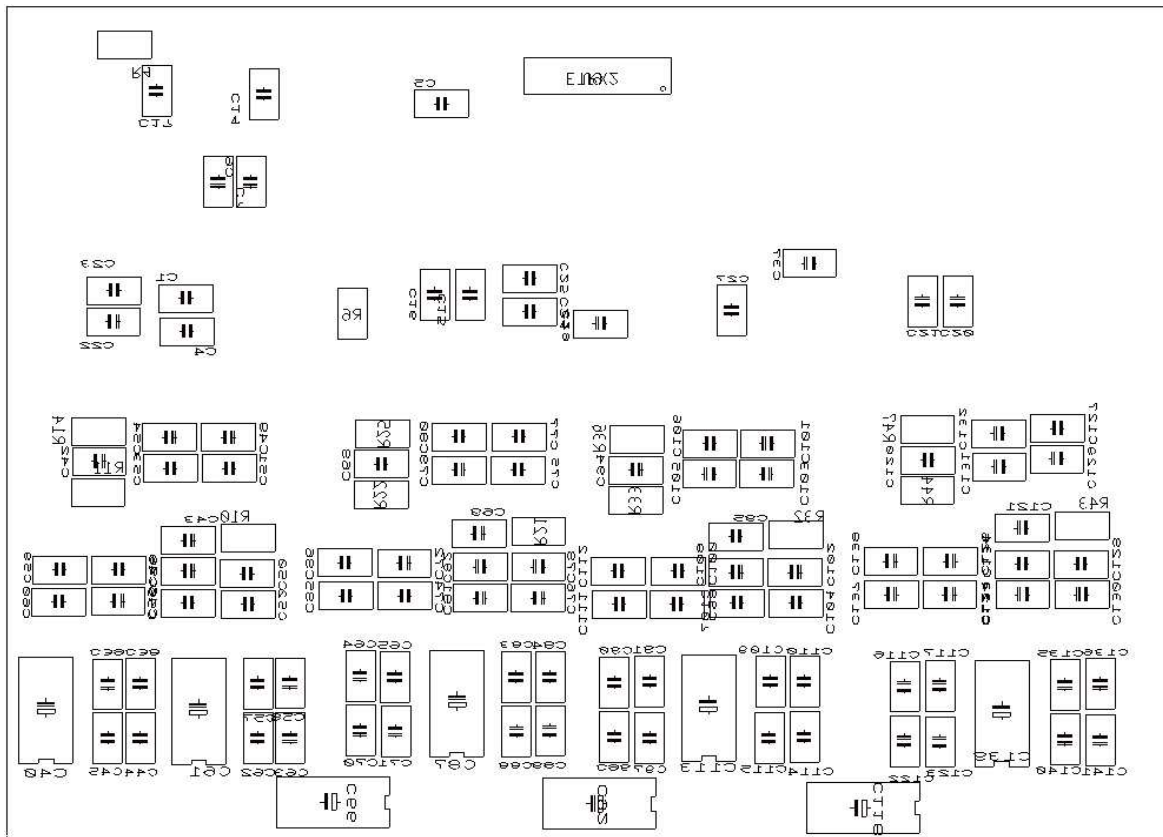
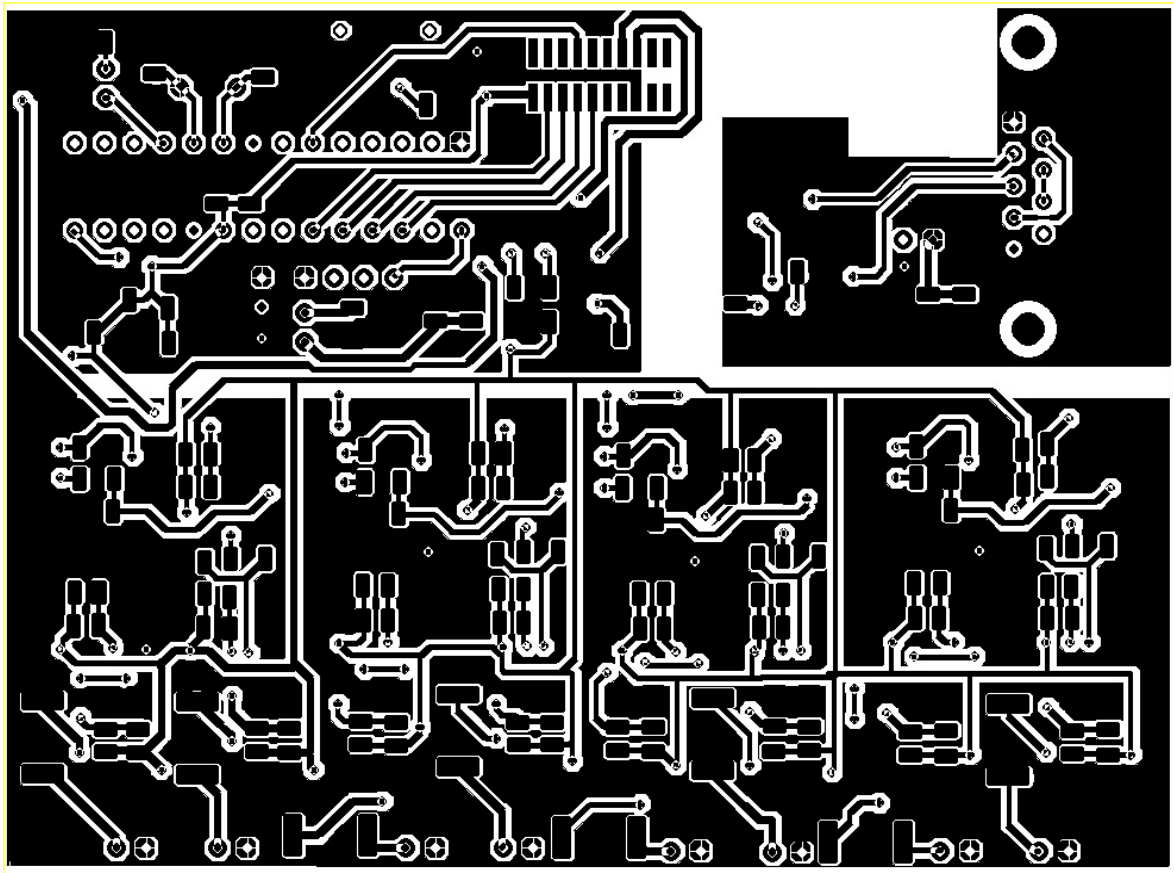
Anexo A:

Trazado de pistas del circuito impreso **(Layout)**

CARA SUPERIOR



CARA INFERIOR



Anexo B:

Programa del PIC

```

//=====
//    Código que realiza la adquisición de datos de 16 bits a través del MSSP
//    y los envía a través de la EUSART con el fin de procesarlos en un PC.
//
//=====

//-----

#include <p18f2685.h>
#include <spi.h>
#include <usart.h>
#include <delays.h>

//-----

#define A2 0x40;
#define A1 0x20;
#define A0 0x10;
#define Internal_Clock 0x03;
#define Single_Ended 0x04;
#define Start_Bit 0x80;
#define Muestras 0x200;

//declaro la sección donde guardaré las 256 muestras del canal 1

#pragma udata datos1
int canal1[0x200];
#pragma udata

//declaro la sección donde guardaré las 256 muestras del canal 2

#pragma udata datos2
int canal2[0x200];
#pragma udata

//declaración de funciones

void main (void);
void Timer2Interrupt (void);
int Inicializar (void);
int LeeCanal (char codebase);
void Transmite (unsigned int valor);
void Espera_Trama (unsigned int valor);
int Espera_Trama2 (unsigned int valor);
char ReadUART (void);

union
{
    struct
    {
        unsigned Timeout:1;           //flag para indicar interrupción del TMR2
        unsigned state:1;             //Flag para indicar el estado del programa
                                        //Adquisición o Transmisión
        unsigned channel:1;           //Flag para indicar qué canal leer.
        unsigned None:5;
    } Bit;
    unsigned char Byte;
} Flags;

//-----
// Main routine

```

```

void
main ()
{
    char code = 0, codebase = 0, valor1 = 0;
    unsigned int index = 0, valor = 0;

    codebase = Inicializar();           //Pasamos a la inicialización.
    Espera_Trama (0x55);                //Espero la recepción de un dato.

while (1)                               //Bucle infinito
{
    if (Flags.Bit.state == 0)           //State == 0 => Adquisición de datos
    {

        TMR2 = 0;                       //Reseteo timer2
        PIR1bits.TMR2IF = 0;            //Reseteo flag interrupción TMR2
        PIE1bits.TMR2IE = 1;            //Habilito interrupción TMR2
        do
        {
            if (Flags.Bit.Timeout == 1) //Si ha ocurrido interrupción leo
                                        //canales
            {
                Flags.Bit.channel = 0; //En primer lugar leer canal 0
                Flags.Bit.Timeout = 0; //Reseteo la variable que indica
                                        //interrupción TMR2
                PORTCbits.RC0 = 1;      //Enciendo led.
                //Lectura de Canal 1
                canal1[index] = LeeCanal (codebase);
                //Lectura de Canal 2
                canal2[index] = LeeCanal (codebase);
                index++;                //Incrementamos apuntador
            }

            }while (index < 0x200);
            //Mientras apuntador < 512 muestras se ejecuta el bucle.
            Flags.Bit.state = 1;
            //Cambio a modo transmisión al obtener 512 muestras
            index = 0;
            //Reseteo la variable que apunta al N° de muestra.

        }
    else
    {

        if (Flags.Bit.state == 1)        //State == 1 => Transmisión de datos
        {
            PIE1bits.TMR2IE = 0;         //Deshabilito la interrupción del TMR2
            Espera_Trama (0x55);         //Espero la recepción de un dato.
            PORTCbits.RC0 = 1;           //Enciendo led.
            do
            {
                while(BusyUSART());      //Espero mientras USART ocupada.
                //Envío primera trama para avisar al PC que empiezo a
                //enviar las muestras del canal1.
                WriteUSART (0x77);
                while(BusyUSART());      //Espero mientras USART ocupada.
                //Envío segunda trama para avisar al PC que empiezo
                //a enviar las muestras del canal1.
                WriteUSART (0x77);
            }
        }
    }
}

```

```

while (index < 0x200)          //Bucle que se repite 512 veces.
{
    //En la variable valor guardo el dato que quiero enviar.
    valor = canal1[index];
    Transmite (valor);          //Paso a transmitir el dato.
    index++;                     //Incremento el apuntador.
}

valor1 = Espera_Trama2(0x50);

}while (valor1 != 0x50);
//Si no detectamos ACK del PC volvemos a enviar muestras.
//Reseteo la variable que me apunta el dato a enviar.
index = 0;
Espera_Trama (0x55);
PORTCbits.RC0 = 1;             //Enciendo led.
do
{
    while(BusyUSART());         //Espero mientras USART ocupada.
    WriteUSART (0x78);
    //Envío primera trama para avisar al PC que empiezo a
    //enviar las muestras del canal2.
    while(BusyUSART());         //Espero mientras USART ocupada.
    WriteUSART (0x78);
    //Envío segunda trama para avisar al PC que empiezo
    //a enviar las muestras del canal2.
    while (index < 0x200)       //Bucle que se repite 512 veces.
    {
        valor = canal2[index];
        //En la variable valor guardo el dato que quiero
        //enviar.
        Transmite (valor);      //Paso a transmitir el dato.
        index++;                //Incremento el apuntador.
    }
    valor1 = Espera_Trama2(0x61);
}while (valor1 != 0x61);
//Si no detectamos ACK del PC volvemos a enviar muestras.
index = 0;
//Reseteo la variable que me apunta el dato a enviar.
Flags.Bit.state = 0;
//Cambio a modo adquisición al enviar 512 muestras
}

}

}

/*-----
-
Función de inicialización.
-----
*/

int Inicializar ()
{
    char codebase = Start_Bit;
    Flags.Byte = 0;
    TRISA = 0x1F;                //Declaro entradas y salidas puerto A.

```

```

TRISB = 0xC2;           //Declaro entradas y salidas puerto B.
TRISC = 0x84;           //Declaro entradas y salidas puerto C.

INTCON = 0x00;          //deshabilito interrupciones globales
ADCON1 = 0x0F;
RCONbits.IPEN = 1;      //habilito niveles de prioridad
IPR1 = 0x02;            //prioridad alta para timer2.
T2CON = 0x06;           //Activamos timer2 -
                        //prescaler 1:16 & postscaler 1:1
PR2 = 0x9D;             //Inicializamos PR2.Tciclo = 4/Fosc = 0,4 µs
                        //157 ciclos = 1 ms.
PIE1bits.TMR2IE = 0;    //deshabilita la interrupción del timer2
PIE1bits.SSPIE = 1;     //Habilita la interrupción del SPI
INTCONbits.GIEH = 1;    //Habilita interrupciones globales
PORTAbits.RA5 = 1;      // Ponemos CS a 1
PORTBbits.RB0 = 0;      // Activamos EN
OpenSPI(SPI_FOSC_16, MODE_00, SMPMID); //Habilito SPI
Flags.Bit.Timeout = 1;
OpenUSART( USART_TX_INT_OFF           //Inicializo EUSART
& USART_RX_INT_OFF
& USART_ASYNC_MODE
& USART_EIGHT_BIT
& USART_CONT_RX
& USART_BRGH_HIGH, 31);
codebase = codebase + Single_Ended;    //Inicializo la variable
                                        //codebase que utilizaremos para
codebase = codebase + Internal_Clock;  //enviar la trama que indica el
                                        //canal de lectura del ADC
PORTCbits.RC0 = 0;                    //Apago led.
return codebase;
}

/*-----
-
Función que espera la recepción de un dato por la EUSART.
-----*/

void Espera_Trama (unsigned int valor)
{
    unsigned int valor1 = 0;
    do
    {
        PORTCbits.RC0 = 1;           //Enciendo led.
        while (!DataRdyUSART());     //Espero a recibir dato por EUSART
        valor1 = ReadUSART();         //Guardo el dato recibido
        Delay10KTCYx(10);             //Introduzco un retardo

    }
    while (valor1 != valor);          //Si la trama recibida es 55h salgo.
    PORTCbits.RC0 = 0;               //Apago led.
    Delay10KTCYx(10);               //Introduzco un retardo
}

/*-----
-
Función que mira si los datos recibidos por la EUSART son los esperados.
En caso negativo se repite la operación 10 veces. Si tras las 10 veces el
valor no es correcto sale del bucle.
-----*/
*/

```

```

int Espera_Trama2 (unsigned int valor)
{
    unsigned int valor1 = 0, i = 0;

loop:
    while (!DataRdyUSART());           //Espero a que el dato esté disponible
    valor1 = ReadUSART();               //Lo guardo.
    if (valor1 != valor)               //Si el valor es diferente al que
    {                                  //esperamos entro
        if (i < 10)                   //Si no hemos leído 10 datos entro
        {
            i++;                     //Incremento índice
            goto loop;               //Repetimos bucle.
        }
    }
    return valor1;                    //Devuelvo el valor final.
}

/*-----
-
    Función que realiza la transmisión de datos de 8 en 8 bits.
-----*/

void Transmite (unsigned int valor)
{
    char msb = 0, lsb = 0;

    lsb = valor;                      //En lsb guardo la parte baja de la
                                     //variable valor.
    msb = valor>>8;                  //En msb guardo la parte alta de la
                                     //variable valor.
    while(BusyUSART());               //Espera a que USART esté disponible.
    WriteUSART (lsb);                 //pasamos a transmitir el lsb.
    while(BusyUSART());               //Espera a que USART esté disponible.
    WriteUSART (msb);                 //Transmito el msb.
}

/*-----
-
    Función para leer el canal que se especifica mediante el flag channel.
-----*/

int LeeCanal (char codebase)
{
    char code;
    unsigned char msb = 0, lsb = 0;
    unsigned int valor = 0;

    if (Flags.Bit.channel == 0)
    {
        code = codebase;              //La variable code indica al ADC que
                                     //queremos leer el canal 1.
        Flags.Bit.channel = 1;        //Indicamos que el próximo canal a
                                     //leer será el 2.
    }
    else

```

```

{
    code = codebase + A2;           //La variable code indica al ADC que
                                    //queremos leer el canal 2.

}
PORTAbits.RA5 = 0;                 // Ponemos CS a 0
WriteSPI(code);                    //Escribo primera trama para iniciar
                                    //conversión.
WriteSPI(0x00);                    //Realizo una escritura para recibir
                                    //los datos
while (!DataRdySPI());             //Espero a que el dato esté en SSPBUF
msb = SSPBUF;                      //Leo el SSPBUF guardándolo en msb
WriteSPI(0x00);                    //Realizo una escritura para recibir
                                    //los datos
while (!DataRdySPI());             //Espero a que el dato esté en SSPBUF
lsb = SSPBUF;                      //Leo el SSPBUF guardándolo en msb
PORTAbits.RA5 = 1;                 // Ponemos CS a 1
valor = msb;                       //guardo msb en la variable valor de
                                    //16 bits
valor = valor<<8;                  //desplazo la variable valor 8 bits a
                                    //la izquierda
valor = valor + lsb;               //a valor le sumo lsb
return valor;                      //devuelvo la variable valor que
                                    //contiene los 16 bits.
}

//-----
// Vector de interrupción de prioridad alta.
//-----

#pragma code InterruptVectorHigh = 0x08
void InterruptVectorHigh (void)
{
    _asm
    goto Timer2Interrupt           //Salta a la Rutina de interrupción del TMR2
    _endasm

}

//-----
// Rutina de interrupción del TMR2
//-----

#pragma code
#pragma interrupt Timer2Interrupt

void
Timer2Interrupt ()
{
    if (PIR1bits.TMR2IF)           //Verifico que ha ocurrido la interrupción
    {
        PIR1bits.TMR2IF = 0;      //Limpio el flag de interrupción
        Flags.Bit.Timeout = 1;    //Indicamos la interrupción.
    }
}

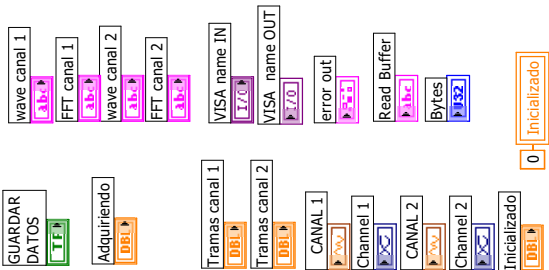
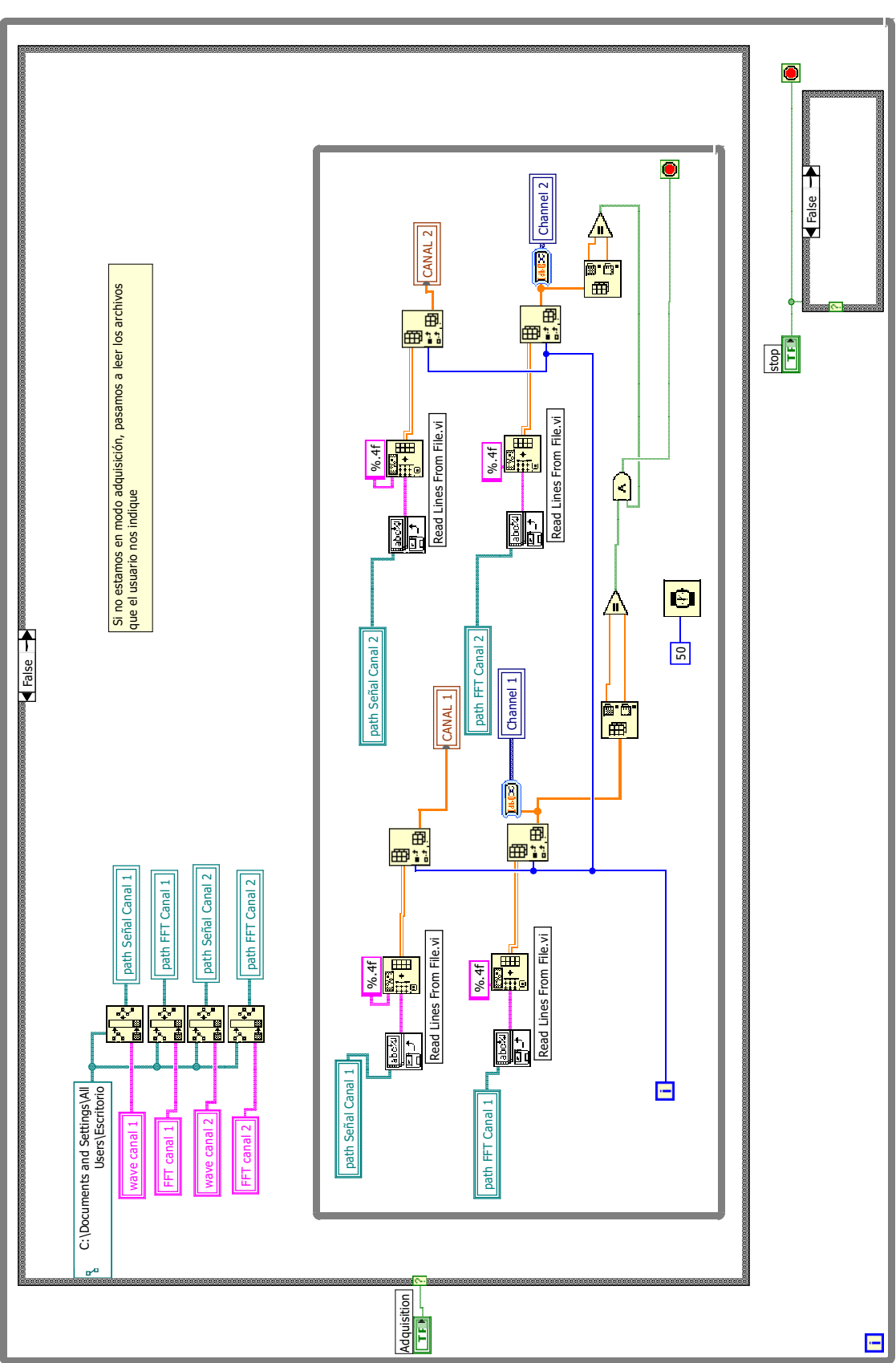
//-----
// Fin del programa.
//-----

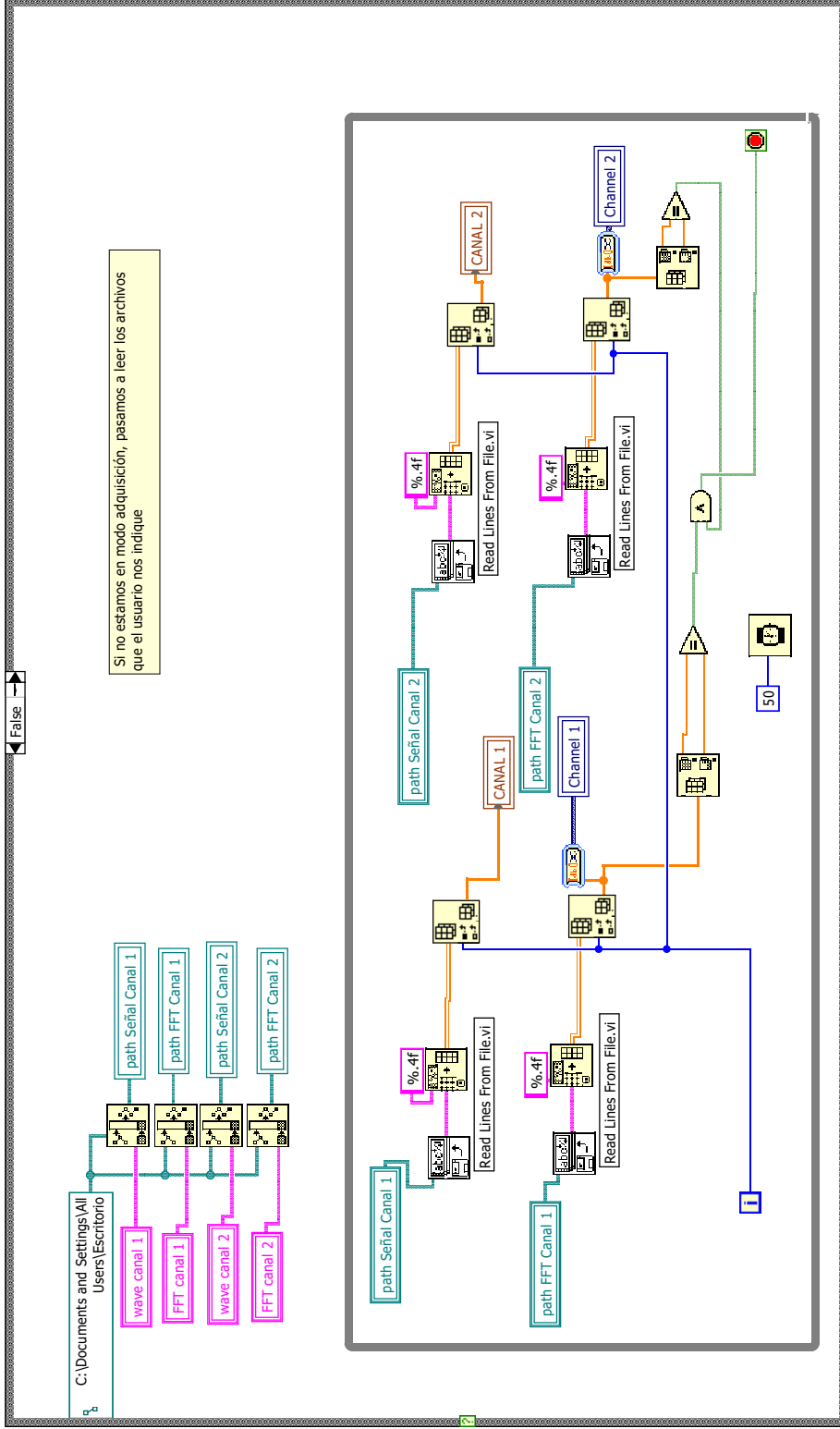
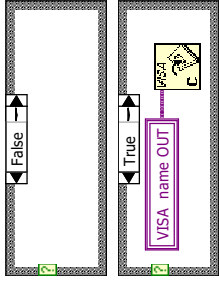
```

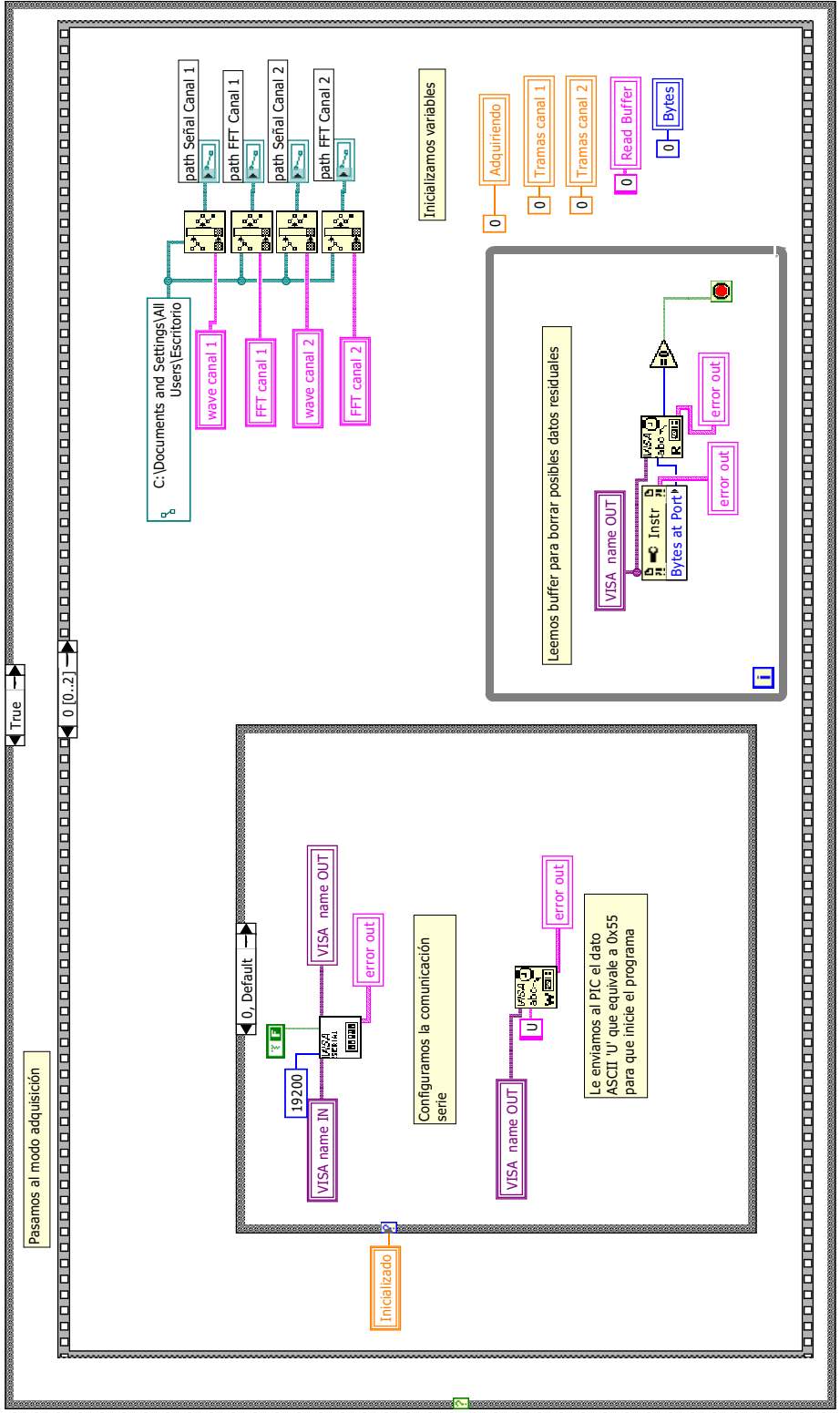
Anexo C:

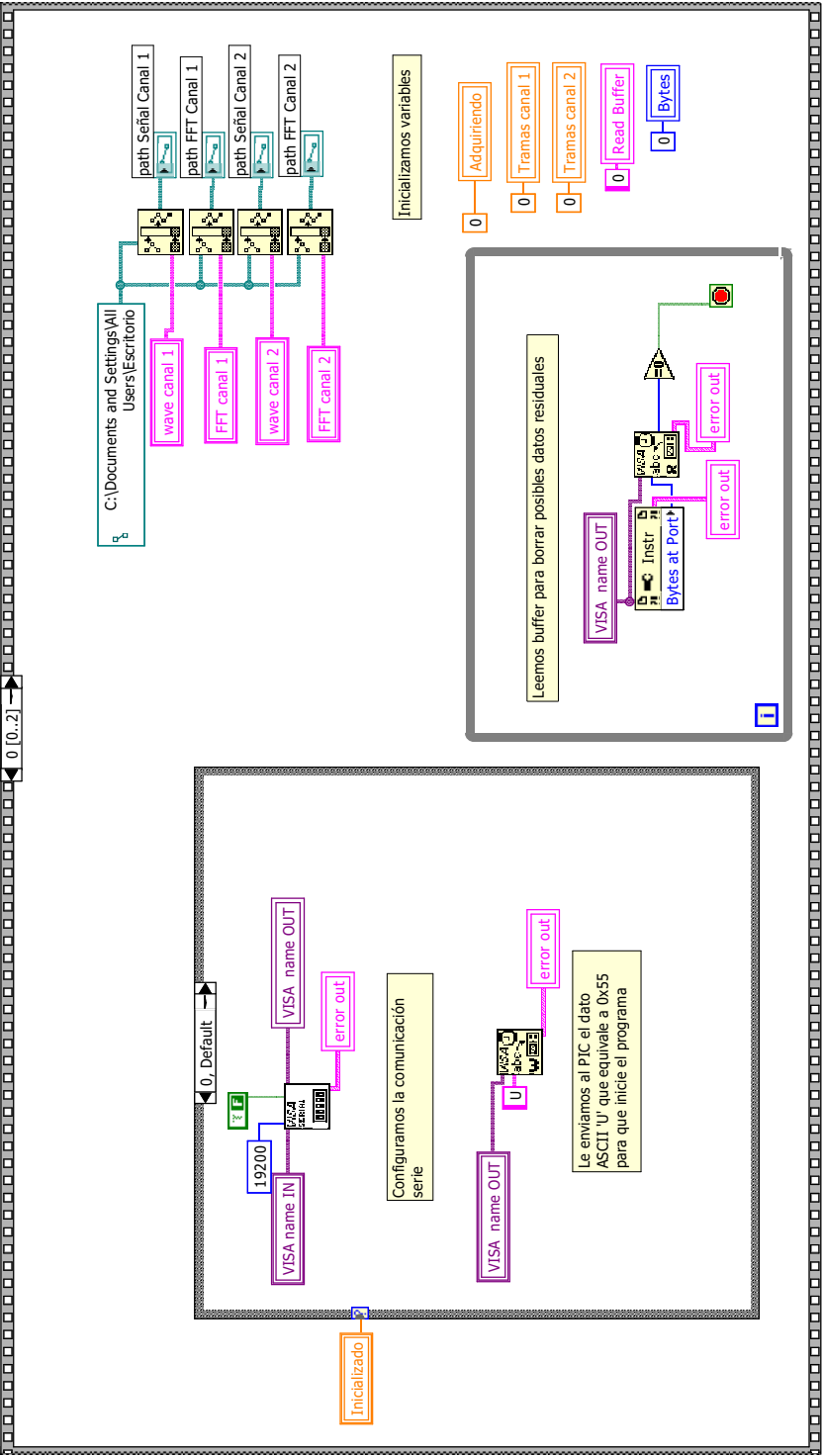
Programa del PC

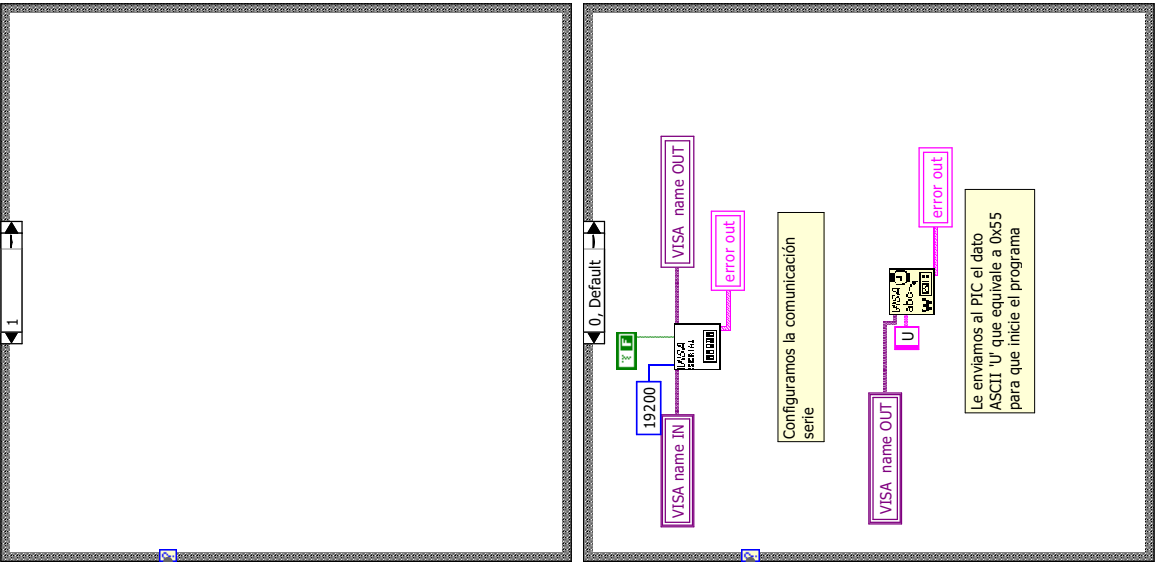
Block Diagram

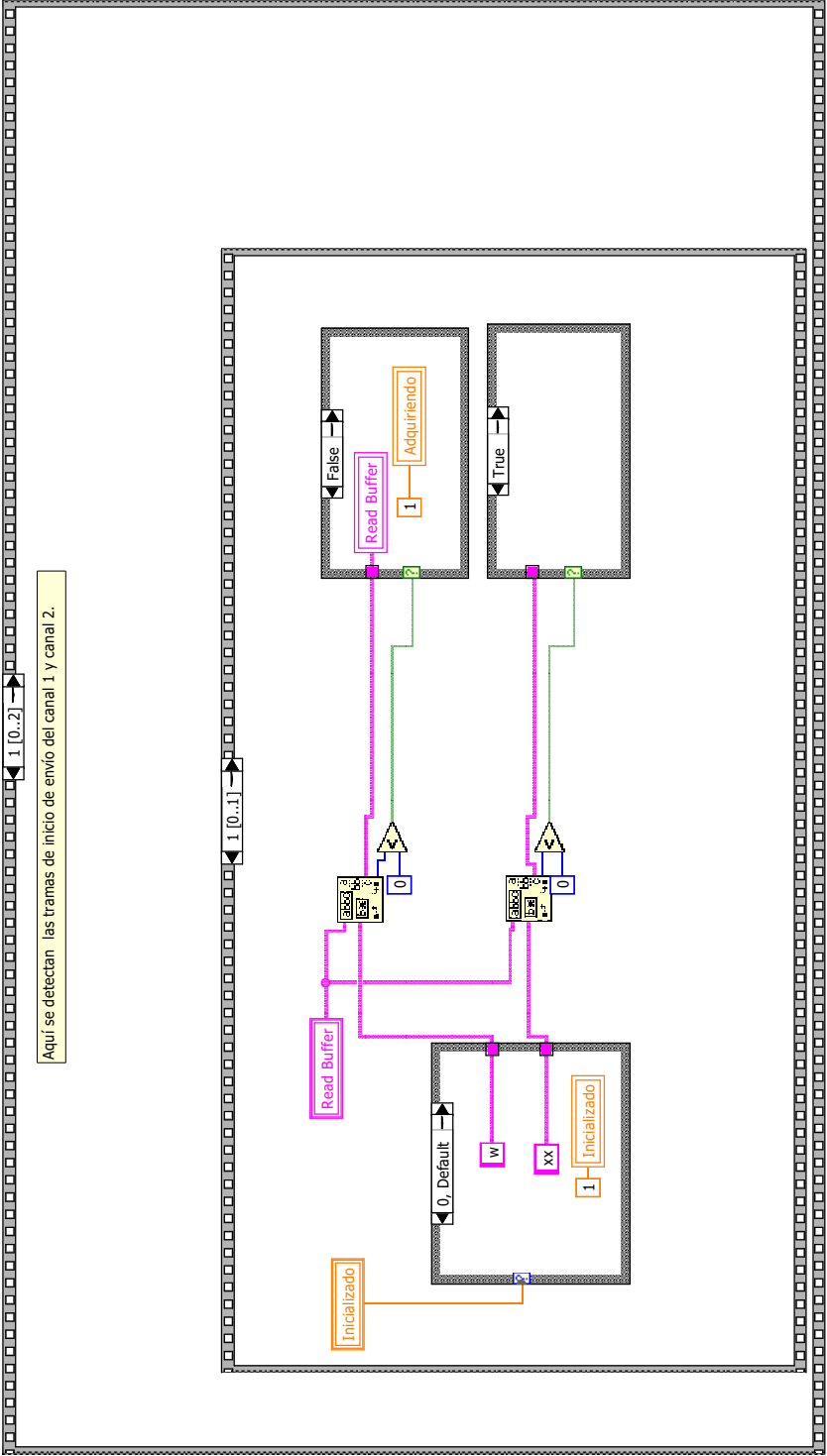


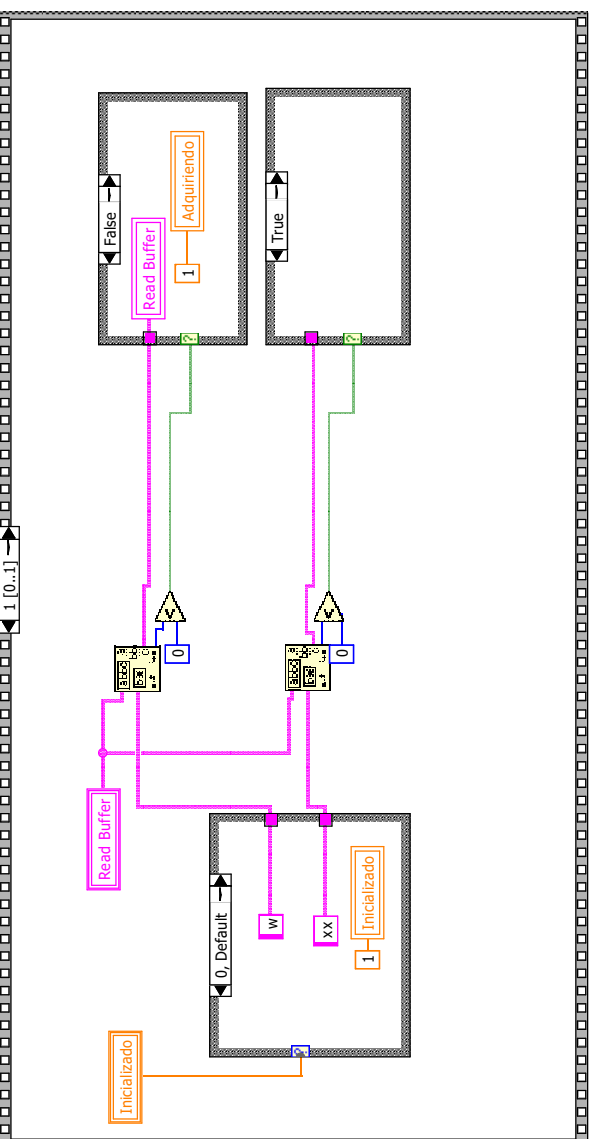


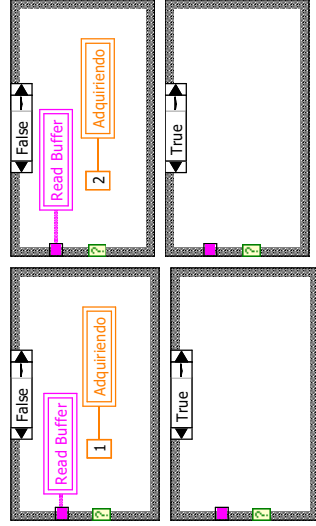


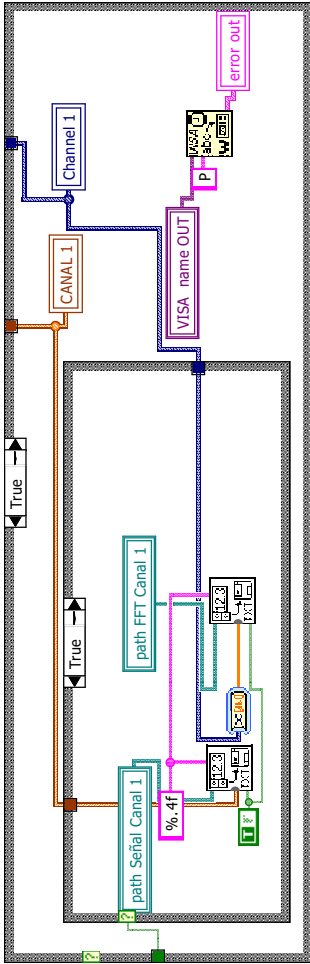
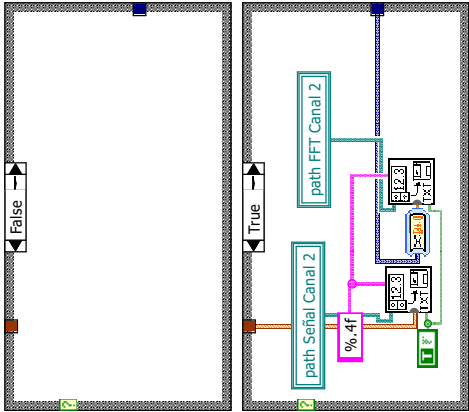
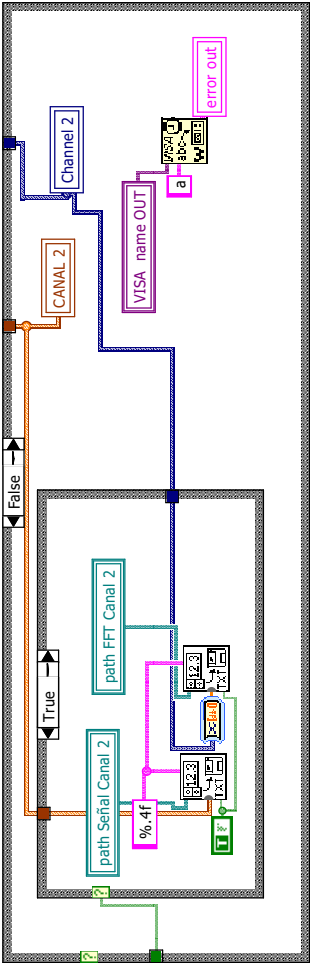


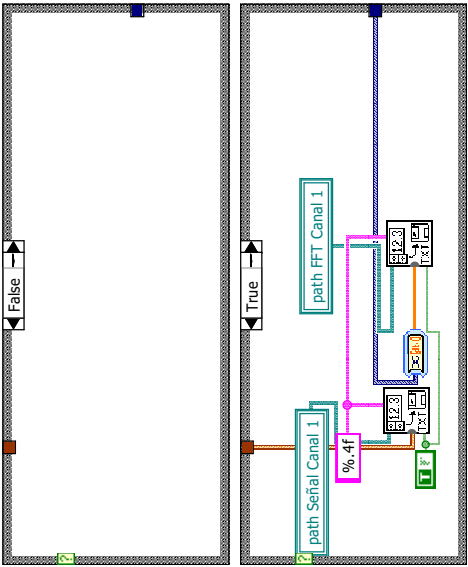






[illegible]





Anexo D:

Presupuesto

PRESUPUESTO INICIAL

CONCEPTO	CANTIDAD	CANTIDAD MÍNIMA	PRECIO	TOTAL
CONDENSADORES				
22 pF - 5% - 50 V - NPO - 1206	2	10	0,081	0,81
10 nF - 10% - 50 V - X7R - 1206	51	10	0,081	4,86
22 nF - 10% - 50 V - X7R - 1206	8	10	0,153	1,53
100 nF - 10% - 25 V - X7R - 1206	6	10	0,084	0,84
1 uF - 10% - 16 V - X7R - 1206	66	1	0,360	23,76
47 uF - 10% - 16 V - Tantalum Electrolytic - D	8	1	1,650	13,20
RESISTENCIAS				
390 - 1% - 200V - 0.125 W - 1206	4	50	0,018	0,90
680 - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
1 k - 1% - 200V - 0.125 W - 1206	4	50	0,036	1,80
2,2 k - 1% - 200V - 0.125 W - 1206	8	50	0,036	1,80
3,3 k - 1% - 200V - 0.125 W - 1206	4	50	0,036	1,80
4,7 k - 1% - 200V - 0.125 W - 1206	1	50	0,036	1,80
10 k - 1% - 200V - 0.125 W - 1206	17	50	0,036	1,80
100 k - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
2 M - 1% - 200V - 0.25 W - 1206	8	50	0,033	1,65
COMPONENTES ACTIVOS				
INA118U - Instrumentation amplifier	4	1	8,45	33,80
OPA2244UA - Dual operational amplifier	8	1	2,2	17,60
ADS8343EBG4 - 16 bits, ADC	1	1	16,00	16,00
AD8627ARZ - Operational amplifier	8	1	3,50	28,00
TC7660SEOA - VOLT CONVERTER, 7660, SOIC-8	1	1	0,68	0,68
SN65C3221PW - Driver/receiver RS-232 SMD	1	1	3,20	3,20
ISL60002DIH312Z - REF 1.25V LOW POWER FGA, SOT-23-3	1	1	2,18	2,18
ADUM2402ARWZ - Digital isolator, 4 channels	2	1	6,25	12,50
MCP1252-33X50I/MS - Charge pump regulator, 3,3 Vout, 120 mA	2	1	1,52	3,04
Modulo ZigBee ETRX2	1	1	42,64	42,64
PIC18F2685-I/SP - 8 bits microcontroller	1	1	10,50	10,50
1N4148 - High speed Diode, D035, 1V, 10mA	1	1	0,02	0,02
CONECTORES				
Conector 2 vías, 1,5 mm, 1 A, batería	10	5	0,085	0,85
Conector header 2 vías, 90DEG, 3 A	10	5	0,34	3,40
Crimp terminal socket	12	10	0,091	1,82
Conector ZigBee a PCB	1	5	1,54	7,70
VARIOS				
Bateria Li-Ion, 3,7V-1080mAh	1	1	13,70	13,70
Cristal 10 MHz, 30 ppm, HC49/4H	1	1	1,22	1,22

2227MC-28-03-05-F1 - Zócalo PIC - pitch 2.54mm, lenght 35.56mm	1	5	0,73	3,65
SOCKET - SMD DIL-28 para PIC	1	1	2,90	2,90
Rocker switch, DP 150 - ON-OFF	1	1	1,19	1,19
CRIMP TERMINAL, FEMALE, 0.5-1.5MM	2	10	0,12	1,19
Led amarillo, 3 mm/T1	1	5	0,15	0,75
Led rojo, 3 mm/T1	1	5	0,15	0,77
Pulsador reset, SPNO, SMT	1	1	0,56	0,56

TOTAL MATERIAL : 270,0 €

MANO DE OBRA	240 Horas		12 €/hora	2.880,0 €
---------------------	-----------	--	-----------	-----------

TOTAL : 3.150,0 €

PRESUPUESTO TRAS RECORTE DE MATERIAL

CONCEPTO	CANTIDAD	CANTIDAD MÍNIMA	PRECIO	TOTAL
CONDENSADORES				
22 pF - 5% - 50 V - NPO - 1206	2	10	0,081	0,81
10 nF - 10% - 50 V - X7R - 1206	31	10	0,081	3,24
22 nF - 10% - 50 V - X7R - 1206	4	10	0,153	1,53
100 nF - 10% - 25 V - X7R - 1206	6	10	0,084	0,84
1 uF - 10% - 16 V - X7R - 1206	42	1	0,360	15,12
47 uF - 10% - 16 V - Tantalum Electrolytic - D	4	1	1,650	6,6
RESISTENCIAS				
390 - 1% - 200V - 0.125 W - 1206	2	50	0,018	0,90
680 - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
1 k - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
2,2 k - 1% - 200V - 0.125 W - 1206	4	50	0,036	1,80
3,3 k - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
4,7 k - 1% - 200V - 0.125 W - 1206	1	50	0,036	1,80
10 k - 1% - 200V - 0.125 W - 1206	9	50	0,036	1,80
100 k - 1% - 200V - 0.125 W - 1206	2	50	0,036	1,80
2 M - 1% - 200V - 0.25 W - 1206	4	50	0,033	1,65
COMPONENTES ACTIVOS				
INA118U - Instrumentation amplifier	2	1	8,45	16,9
OPA2244UA - Dual operational amplifier	4	1	2,2	8,8
ADS8343EBG4 - 16 bits, ADC	1	1	16,00	16,00
AD8627ARZ - Operational amplifier	4	1	3,50	14
TC7660SEOA - VOLT CONVERTER, 7660, SOIC-8	1	1	0,68	0,68

SN65C3221PW - Driver/receiver RS-232 SMD	1	1	3,20	3,20
ISL60002DIH312Z - REF 1.25V LOW POWER FGA, SOT-23-3	1	1	2,18	2,18
ADUM2402ARWZ - Digital isolator, 4 channels	2	1	6,25	12,50
MCP1252-33X50I/MS - Charge pump regulator, 3,3 Vout, 120 mA	2	1	1,52	3,04
Modulo ZigBee ETRX2	0	1	42,64	0
PIC18F2685-I/SP - 8 bits microcontroller	1	1	10,50	10,50
1N4148 - High speed Diode, D035, 1V, 10mA	1	1	0,02	0,02
CONECTORES				
Conector 2 vías, 1,5 mm, 1 A, batería	10	5	0,085	0,85
Conector header 2 vías, 90DEG, 3 A	10	5	0,34	3,40
Crimp terminal socket	12	10	0,091	1,82
Conector ZigBee a PCB	1	5	1,54	7,70
VARIOS				
Bateria Li-Ion, 3,7V-1080mAh	1	1	13,70	13,70
Cristal 10 MHz, 30 ppm, HC49/4H	1	1	1,22	1,22
2227MC-28-03-05-F1 - Zócalo PIC - pitch 2.54mm, lenght 35.56mm	1	5	0,73	3,65
SOCKET - SMD DIL-28 para PIC	1	1	2,90	2,90
Rocker switch, DP 150 - ON-OFF	1	1	1,19	1,19
CRIMP TERMINAL, FEMALE, 0.5-1.5MM	2	10	0,12	1,19
Led amarillo, 3 mm/T1	1	5	0,15	0,75
Led rojo, 3 mm/T1	1	5	0,15	0,77
Pulsador reset, SPNO, SMT	1	1	0,56	0,56

TOTAL MATERIAL : 170,81 €

MANO DE OBRA	240 Horas		12 €/hora	2.880,0 €
---------------------	-----------	--	-----------	-----------

TOTAL : 3.050,81 €

REDUCCIÓN COSTE MATERIAL: 270 € - 170,81 € = **99,2 €**

Castellano:

El sistema desarrollado en este trabajo permite la detección de fatiga del músculo de un paciente. Las señales electromiográficas generadas por el músculo son detectadas mediante electrodos superficiales, que sirven de entrada al circuito hacia los diferentes canales de entrada. Tras una primera etapa de amplificación, las señales son filtradas por un filtro pasabanda. Las señales acondicionadas se digitalizan mediante un conversor analógico digital que está conectado a un microcontrolador. Éste establece la comunicación con el PC enviándole los datos a través del puerto serie. Finalmente, mediante un programa implementado en Labview se procesan los datos y se determina la existencia o no de fatiga muscular.

Català:

El sistema desenvolupat en aquest treball permet la detecció de fatiga del musle d'un pacient. Els senyals electromiogràfics generats pel múscul son detectats mitjançant elèctrodes superficials, que serveixen d'entrada al circuit cap als diferents canals d'entrada. Després d'una primera etapa d'amplificació, els senyals son filtrats per un filtre passabanda. Els senyals condicionats es digitalitzen mitjançant un convertidor analògic/digital que està connectat a un microcontrolador. Aquest estableix la comunicació amb el PC enviant-li les dades pel port sèrie. Finalment, mitjançant un programa implementat en Labview, es processen les dades i es determina l'existència o no de fatiga muscular.

English:

The system developed in this work allows the detection of fatigue in the muscle of a patient. The electromyographic signals generated by the muscle are detected by means of superficial electrodes that they serve as entrance to the circuit towards the different input channels. After one first stage of amplification, the signals are filtered by a pass-band filter. The electromyographic signals are digitized using an analog to digital converter that is connected to a microcontroller. This one communicates with the PC sending the data through the serial port. Finally, by means of a program implemented in Labview the data are processed and the existence or not of muscular fatigue is determined.