PROYECTO FIN DE CARRERA

TÍTULO: Diseño e implementación de un servicio de acceso inalámbrico a dispositivos médicos mediante H2H

AUTOR**:** Tomás Valencia Noheda

TUTOR**:** Álvaro Araujo Pinto

DEPARTAMENTO**:** Ingeniería electrónica

TRIBUNAL:

Presidente**:** Da. Ma Elena Hernando Pérez

Vocal: D. Alvaro Araújo Pinto

Secretario**:** D. Fernando Fernández Martínez

VocalSuplente**:** D. Octavio Nieto-Taladriz García

FECHA DE LECTURA**:** Madrid, de de 2016

CALIFICACION**:**

Resumen

En este proyecto de fin de carrera se ha desarrollado el software para el servicio de autentificación *Heart To Heart*(H2H). El objetivo es conseguir un método de acceso seguro a dispositivos médicos implantables que utilice claves generadas dinámicamente a partir de la señal biométrica del corazón. El sistema estará compuesto por un dispositivo de acceso que será utilizado por el usuario que quiera conseguir acceso al dispositivo medico implantable, y un dispositivo que simulará dicho dispositivo médico. Ambas plataformas hardware han sido diseñadas e implementadas en el proyecto “Diseño e implementación de una plataforma hardware para un sistema de acceso inalámbrico a dispositivos médicos mediante *Heart-To-Heart*”.

Para realizar este proyecto se ha hecho un estudio del protocolo de autentificación H2H, que se caracteriza principalmente por eliminar las claves estáticas y utilizar características físicas dinámicas, en nuestro caso la señal del corazón, para generar claves de un solo uso, de manera que no solo se incremente la seguridad si no que se evite el uso de cualquier autoridad de certificación.

A continuación, se ha realizado un proceso de diseño de los distintos módulos software que realizaran las funcionalidades necesarias para conseguir la implementación del servicio *Heart To Heart* tanto en el lado del dispositivo acceso como en el simulador de dispositivo médico.

Por último, se han realizado una serie de pruebas para comprobar el correcto funcionamiento del sistema y su resistencia a ataques maliciosos, además de su posible utilidad como herramienta de ayuda al diagnóstico. De estas pruebas realizadas se han extraído una serie de conclusiones y se han establecido varias líneas de trabajo futuro para el proyecto.

Palabras clave

Dispositivo medico implantable, protocolo H2H, seguridad biométrica (…)

Agradecimientos

Índice

Índice de figuras

Índice de tablas

Acrónimos

H2H *Heart To Heart*

ECG Electrocardiograma

IMD Implantable Medical Device

# Introducción

## Motivación

El uso de los dispositivos médicos implantables está ampliamente extendido debido a las facilidades que ofrecen a la hora de monitorizar constantemente al paciente y proporcionarle tratamiento que de otra manera seria más complejos o agresivos de aplicar. Los dispositivos médicos activos (marcapasos, bombas de insulina, etc.) normalmente disponen de módulos radio que permiten tanto la extracción de datos del paciente como la reprogramación/reajuste de dichos dispositivos de una manera no invasiva.

En la actualidad, los protocolos de comunicación son poco seguros ya que se prioriza el que el acceso sea rápido y sencillo en casos de riesgo crítico para la salud. Sin embargo, diversos estudios como el realizado en [3] han demostrado que dicho planteamiento es altamente inseguro, ya que con una formación y unos recursos “básicos” se pueden realizar ataques que permitan el acceso a los datos monitorizados e incluso desactivar o alterar el funcionamiento del dispositivo médico implantables (IMD).

Para solucionar esta deficiencia de seguridad, se plantean diversos enfoques entre los que destacan el uso de métodos criptográficos para cifrar las comunicaciones inalábrambricas y métodos para el control de acceso a los dispositivos. Dentro de este segundo grupo, se encuentran las políticas de acceso que emplean alguna señal biométrica para conseguir el acceso al IMD, de manera que el paciente debe dar su consentimiento explícito para que dicha señal se monitorizada.

Una de estas políticas de acceso, y cuya implementación será el objetivo de este proyecto, es la llamada *Heart-To-Heart*(H2H). En esta política de acceso, se emplea un dispositivo de acceso externo que monitoriza la señal del corazón durante unos segundos para extraer la información necesaria que permita la validación de este dispositivo. Una vez cumplido el periodo de monitorización, el dispositivo de acceso envía la medida al IMD para que este la compare con su propia medida y decida si el intento de acceso es legítimo.

Utilizando este método de acceso conseguimos que el paciente sea consciente de los intentos de acceso a su dispositivo y eliminamos inconvenientes de otros métodos de control de acceso como el uso de certificados, que requiere infraestructura adicional para la gestión de dichos certificados.

Por último, para evitar las situaciones de riesgo en las que sea impracticable dicha monitorización, ya sea porque el paciente este sufriendo convulsiones o porque no se pueda perder ese tiempo, el protocolo propone un modo promiscuo de acceso en el que se prima la facilidad de acceso ante la seguridad.

## Objetivos

El objetivo principal de ambos proyectos en conjunto es desarrollar e implementar un sistema electrónico portátil que permita acceder a la configuración y registros de un marcapasos simulado utilizando una política de acceso “*touch-to-access*” a través de una conexión inalámbrica.

El hardware de dicho dispositivo ha sido diseñado en el proyecto “Diseño e implementación de una plataforma hardware para un sistema de acceso inalámbrico a dispositivos médicos mediante *Heart-To-Heart*” realizado por Samuel López Asunción.

Por lo tanto, el objetivo de este proyecto es el de diseñar e implementar un servicio de acceso inalámbrico a dispositivos médicos mediante H2H. El sistema se encarga de proporcionar un método de acceso sencillo y seguro a dispositivos médicos implantables.

Dentro de este objetivo principal, pueden distinguirse los siguientes subobjetivos:

* Desarrollo de un módulo software que se encargará del tratamiento de la señal recogida por los electrodos y la extracción de la información necesaria para generar la clave.
* Desarrollo de un módulo software que controlará el acceso al dispositivo médico, bien validando la contraseña recibida o bien entrando en modo promiscuo en caso de que situación de riesgo para la salud.
* Desarrollo de un módulo software que gestionará la comunicación inalámbrica, tanto el proceso de descubrimiento de dispositivos como el de emparejamiento de los mismos
* Desarrollo de un módulo software de interfaz de usuario, lo cual permitirá al usuario comenzar el proceso de acceso de una manera sencilla e intuitiva, así como la presentación de otros datos de interés.
* Integración y pruebas de cada uno de los módulos en el sistema final.

## Planificación del desarrollo

El desarrollo del proyecto ha seguido las siguientes fases:

Análisis y definición del sistema

En esta fase se ha realizado el estudio de los trabajos en las que se basa este proyecto y la plataforma hardware en la que se va a desarrollar con el objetivo de poder definir los requisitos que tiene que cumplir la solución propuesta.

Diseño de la solución propuesta

Una vez definidos los requisitos que debe cumplir nuestro sistema, se hizo el diseño de cada uno de los módulos principales que integran el proyecto, estableciendo para cada uno de ellos los submódulos de los que estará formado y detallando las decisiones que se han tomado para conseguir que se cumplan los requisitos definidos antes.

Implementación del software del sistema

El siguiente paso en el desarrollo fue la implementación del software diseñado, lo cual implica un análisis de las distintas estrategias posibles para abordar el problema, la decisión de cual seguir y la programación de software necesario para dicha solución.

Integración y pruebas

La última fase fue la integración de cada uno de los módulos individuales y la realización una serie de pruebas para comprobar el correcto funcionamiento de todos del sistema conjunto, tanto del hardware como del software desarrollado, y su conformidad con los requisitos planteados. Por último, se han extraído una serie de conclusiones sobre la solución final implementada y se sugieren varias líneas futuras a partir del trabajo realizado.

## Estructura de la memoria

En primer lugar, se expone en esta memoria un análisis de requisitos en el que se especifican los requisitos del software de la plataforma que se deben cumplir en el sistema final.

A continuación, se describe el proceso de diseño que se ha seguido para definir los distintos módulos software que integran el proyecto y justificando las decisiones que se han tomado. Dicho software está dividido en dos prototipos, ya que durante la implementación del software del primero se detectaron problemas insalvables que obligaron a abandonar dicho prototipo.

En tercer lugar, se detalla el proceso de implementación del software de la plataforma, lo cual incluye los problemas encontrados durante la implementación del primer prototipo y los motivos por los cuales fue descartado el primer prototipo hardware.

Seguidamente se describen las pruebas realizas al sistema para determinar el correcto funcionamiento del mismo, y se añaden las conclusiones que se han extraído de dichas pruebas, así como las posibles líneas de trabajo futuro.

En la última parte de la memoria se puede encontrar, además, un apéndice con una lista de las herramientas empleadas y una lista de referencias utilizadas a lo largo de este proyecto.

# Análisis de requisitos

## Introducción

En este capítulo se analizarán los requisitos que deben cumplir los distintos módulos software. En el desarrollo del servicio de acceso H2H se deben diseñar diversos módulos que gestionen tanto la lógica del algoritmo de validación como la comunicación inalámbrica de ambos dispositivos y, en el caso del dispositivo de acceso, la interfaz de usuario.

Este proyecto está basado en el contenido del trabajo [1], por lo que los requisitos han sido definidos acorde con lo necesario para implementar su propuesta. Los motivos de la elección de este método de control de acceso frente a otros métodos son el hecho de que no se necesite ningún tipo de infraestructuras/servicios adicionales para su funcionamiento y el tipo de señal biométrica que utiliza. El primer motivo es una ventaja inmediata si comparamos este protocolo con el uso de certificados, en los que es necesaria la intervención de una entidad certificadora cada vez que se establece una comunicación. El segundo motivo se debe a la variabilidad que tiene la señal cardiaca que permite generar claves de un solo uso, lo cual supone una ventaja frente a otras características fisiológicas estáticas (huella dactilar).

A pesar de que el sistema cuenta con dos dispositivos distintos (programador y simulador de marcapasos), dado sus similitudes en el hardware se ha decidido hacer un único software y diseñarlo de manera que pueda activar/desactivar las funcionalidades que sean necesarias en cada uno de los dispositivos.

## Requisitos generales

El sistema que vamos a diseñar e implementar basa su funcionamiento en el muestreo periódico de una señal externa, la cual será procesada tanto para su uso en el algoritmo de autentificación como, en el caso del dispositivo de acceso, ser mostrada a través de la interfaz de usuario. Dicha interfaz de usuario además proporcionará una interactividad con el dispositivo, lo cual permitirá desde comenzar el procedimiento de acceso hasta la configuración de ciertos parámetros tanto del propio dispositivo como del simulador de marcapasos.

Estas características hacen que el diseño de este sistema deba hacerse teniendo en cuenta las restricciones de un sistema de tiempo real estricto, ya que un *jitter* en el muestreo de la señal puede provocar errores en la detección del ritmo cardiaco y, por lo tanto, en el algoritmo de autentificación. Por otro lado, la interfaz de usuario está sujeta a las restricciones típicas de responsividad y disponibilidad necesarias para que la experiencia de usuario sea positiva.

Por lo tanto, en el diseño de cada uno de los módulos deberá cumplir con unos requisitos de tiempo de ejecución estrictos que permitan cumplir los plazos de todas las tareas necesarias para su funcionamiento.

Adicionalmente, dado que se trata de un sistema alimentado por batería, se deberá tener en cuenta el consumo desde el punto de vista software de manera que en cada instante solo estén en funcionamiento aquellos módulos, tanto hardware como software, que sean estrictamente necesarios en ese momento.

## Servicio

### Tratamiento de la señal cardiaca

Este módulo deberá acondicionar la señal adquirida con el objetivo de extraer la información de la manera más precisa posible. Dado que el sistema hardware no cuenta con un módulo especifico de tratamiento de señal, el diseño de esta parte deberá tener especial cuidado con el tiempo de ejecución, con lo que habrá que llegar a un compromiso entre la exactitud y el tiempo de ejecución.

Dado que la señal con la que vamos a trabajar presenta unos rasgos distintivos muy marcados, es conveniente hacer una pequeña caracterización de dicha señal para entender los requisitos que vamos a imponer a este módulo.

Dejando de la caracterización fisiología de la señal, la cual esta descrita con más detalle en la memoria del proyecto (conjunto), si realizamos el análisis desde un punto de vista eléctrico, debemos tener en cuenta dos cosas: los tipos de ruido que afectan a la señal y el ancho de banda donde se sitúa la información de interés.

La mayor contribución de ruido en ese tipo de señal proviene del ruido mioelectrico (muscular) y de las imperfecciones en los electrodos, si bien como cualquier sistema electrónico está expuesto a interferencias electromagnéticas, estas a menudo están corregidas en el propio hardware (por ejemplo, el Analog Front End que utilizamos filtra la interferencia de red).

Dicho ruido se sitúa en frecuencias bajas que solapan con la propia señal del corazón, con lo cual es muy difícil de eliminar, pero hay un fenómeno que afecta especialmente y que si se puede eliminar con cierta facilidad. Este fenómeno se denomina *baseline wander*, y consiste en una variación brusca del nivel de continua debido generalmente a malos contactos de los electrodos o a espasmos musculares del paciente. Esta componente de ruido tiene concentrada su energía entre 0 y 1 Hz, con lo que se puede eliminar a través de filtrado.

Respecto al ancho de banda de información, la bibliografía existente al respecto distingue entre adultos e infantes: para el primer caso el ancho de banda está entre 1 y 150 Hz y para el segundo caso, entre 1 y 250 Hz. Esta diferencia es debido a características fisiológicas del sistema cardiaco de los infantes en las que no entraremos en detalles, pero que desde el punto de vista de nuestro análisis presentan un problema a la hora de definir nuestro ancho de banda de trabajo y también el diseño del modo promiscuo mencionado previamente.

Por último, dado que los algoritmos de detección del ritmo cardiaco diseñados para tiempo real basan su funcionamiento en la amplitud de la señal, es conveniente una caracterización de la “forma” de la señal. En la figura X (ya veremos el numero) se puede ver una muestra de esta señal con la que vamos a trabajar:



Como se puede observar, se producen una gran cantidad de eventos durante el proceso de un latido los cuales pueden ser utilizados para la detección del ritmo cardiaco. En nuestro caso, utilizamos la transición abrupta de los complejos QRS (intervalo formador por las ondas Q, R y S) para calcular la distancia temporal R-R.

Vistas las características de la señal del corazón, podemos pasar a definir los requisitos que debe cumplir las etapas de tratamiento de señal.

#### Requisitos de resolución y fase de los filtros

El filtrado digital de este sistema se encargará de eliminar la máxima cantidad de ruido proveniente de la adquisición de la señal cardiaca conservando las componentes que contienen la información fundamental para el sistema y respetando las restricciones de tiempo de ejecución que tenemos.

Los filtros elegidos deben lo más sencillos posible, ya que un mayor orden del filtro implica un mayor número de operaciones y esto impacta negativamente en el tiempo que tarda en realizarse el filtrado. Además, su fase deberá ser lo más lineal posible, ya que un retardo de grupo no constante puede distorsionar el cálculo de la frecuencia cardiaca.

Por último, hay que tener en cuenta la cuantización que van a sufrir los coeficientes, en especial si no disponemos de unidad de coma flotante hardware y tenemos que trabajar en coma fija. Esta cuantización no debe afectar a la respuesta en frecuencia de nuestros filtros (o en caso de hacerlo, debe ser compensada). Dicha respuesta en frecuencia debe ser estable y presentar un rizado mínimo en la banda de interés para que no afecte a la detección del ritmo cardiaco.

#### Algoritmos de detección de ritmo cardiaco

EL algoritmo utilizado deberá ser capaz de extraer la información de la señal cardiaca en tiempo real, ya que en el hardware dispone de una cantidad limitada de memoria y no es viable el almacenamiento de las muestras para un procesado offline.

##### El algoritmo utilizado para este sistema deberá ser capaz de extraer la información pulso a pulso ya que en la generación de la contraseña se toma la distancia entre los máximos de la onda R del ECG. Adicionalmente, con el objetivo de no acaparar los recursos del microprocesador, se evitará el uso de algoritmos que almacenen gran cantidad de muestras.

### Algoritmo de autenticación

El algoritmo de autenticación consiste a grandes rasgos en tres etapas.

En la primera etapa, ambos dispositivos monitorizan la señal del corazón y extraen una clave concatenando los cuatro últimos bits de la distancia temporal R-R, los cuales son independientes e idénticamente distribuidos.

Una vez que tienen dicha clave, calculan el espacio de posibles claves validas, dado que la lectura de la señal del corazón está sujeta a ruido y es casi imposible obtener exactamente la misma clave. Dicho cálculo se hará un umbral fijo calculado previamente *offline*, en nuestro caso con Matlab.

Cuando ya se tiene toda la información, se inicia un proceso en el que el IMD envía primero su clave con el objetivo de que el dispositivo de acceso decida si efectivamente está intentando a un IMD o impostor que intenta conseguir su clave para luego usarla para acceder al IMD. Una vez que el dispositivo de acceso ha hecho la verificación, envía su clave para que el IMD la valide y proporcione o no acceso.

Todos los detalles sobre este algoritmo y las mejoras que seguridad que supone su uso, están detalladas en [1].

Visto el procedimiento de validación, este módulo será deberá ser capaz de hacer todos los cálculos necesarios para llevar a cabo la generación de la clave y la comprobación de la validez de esta, así como el de gestionar todo el proceso de intercambio de información.

### Comunicación inalámbrica

La comunicación inalámbrica es un punto fundamental de nuestro diseño. Dado que el acceso físico a los IMD requiere de intervenciones quirúrgicas muy agresivas para el paciente, las nuevas generaciones están siendo diseñadas con una interfaz inalámbrica como requisito básico para evitar dichas intervenciones.

En nuestro caso, la interfaz no solo debe proporcionar un medio para comunicarnos con el IMD, sino que además tiene también la función de proporcionar algún método para proteger dicha comunicación, ya que, si los datos que se van a intercambiar por esta fuera fácilmente interceptarles, carecería de sentido el implementar cualquier control de acceso.

En cuanto a la inicialización de la comunicación, este módulo debe ser capaz de ser capaz de realizar una serie de funciones para que esta pueda llegar producirse. Estas funciones se pueden clasificar de la siguiente forma:

* Descubrimiento de dispositivos: En el lado del dispositivo de acceso, dado que en principio carecemos de la información concreta de cada IMD, como podría ser una dirección física a la que conectarse, la interfaz deberá ser capaz de detectar los dispositivos cercanos para poder iniciar el vínculo. En el lado del simulador de marcapasos, debe poderse configurar como descubrible.
* Vinculación de dispositivos: Una vez que se dispone de la lista de los dispositivos a los que podemos conectarnos, se debe proporcionar un método en el que el dispositivo de acceso y el dispositivo elegido establezcan un vínculo. Esta vinculación incluye el establecimiento del canal físico y el intercambio de un identificador que permita que la comunicación solo se produzca entre los dos dispositivos vinculados.
* Conexión y transmisión de datos: una vez establecidos todos los parámetros necesarios para la conexión, este módulo debe ser capaz de enviar y recibir datos a través de ese canal físico para poder intercambiar información entre ambos dispositivos.

La tecnología radio utilizada vendrá impuesta por el hardware proporcionado, y la tarea será proveer de una pila de protocolos capaz de gestionarlo. Dicha pila deberá proporcionar además de los servicios básicos para poder establecer la comunicación, algún método de protección de los datos transmitidos en por el canal físico para evitar que estos sean interceptados.

Adicionalmente, se valoran protecciones adicionales ante ataques del tipo *man in the middle,* mediante alguna capa adicional a la pila de protocolos que implemente métodos de transmisión seguros, como puede ser encriptación de los datos.

### Interfaz de usuario

El dispositivo de acceso está pensado para sea usado por un usuario que intenta acceder al IMD con el objetivo de reconfigurar o consultar ciertos datos. Por este motivo, es necesario proveer de una interfaz que le permita manejar todas estas operaciones de una forma sencilla y presentar todos esos datos sin necesidad de ningún método externo.

Por lo tanto, este módulo debe gestionar todo lo relacionado con la interacción usuario-dispositivo y con la presentación de información útil. Dado que nuestra interfaz de usuario viene impuesta por el hardware del que disponemos (LCD + touchpad resistivo), estos requisitos pueden resumirse en lo siguiente:

* Controles básicos del sistema electrónico, que permitirán el inicio de adquisición de datos, apagado por software, etc. Adicionalmente se valora la posibilidad de añadir un input de datos más complejo, como puede ser un teclado en pantalla al estilo de los Smartphone.
* Se ofrecerá la posibilidad de mostrar por pantalla el ECG capturado con el objetivo de que este dispositivo pueda tener futuros usos de ayuda al diagnóstico.
* Para iniciar la comunicación con el simulador de marcapasos, la interfaz debe ofrecer la posibilidad de seleccionar el dispositivo al que nos queremos conectar y mostrar el estado del establecimiento de la comunicación.
* Manipulación de parámetros del simulador de marcapasos, con vistas a un futuro uso de reprogramación de los IMD.

# Diseño del sistema

En esta sección se detalla el diseño del software del sistema. En primer lugar, comenzamos con un análisis de la plataforma hardware proporcionada, poniendo especial atención en el alcance de la capa de abstracción hardware, para poder determinar la flexibilidad de la que disponemos a la hora de modificar algún parámetro de un módulo hardware. Tras esto, daremos una vista general de los bloques que componen el sistema y como se relacionan con los distintos módulos hardware previamente analizados. Por último, detallaremos la funcionalidad individual de cada módulo y las decisiones que se han tomado para el diseño.

Antes de entrar en más detalle, para contextualizar el hecho que se haga referencia a dos prototipos distintos hay que decir que durante la implementación del software del primero de los prototipos se encontraron varios problemas que obligaron a descartar dicho prototipo. Estos problemas fueron causados por el funcionamiento incorrecto del multiplicador hardware y por un incorrecto dimensionamiento de la memoria RAM necesaria para la pila de protocolos (en la sección de implementación se darán más detalles).

## Análisis de la plataforma hardware

El primer paso para el diseño del sistema es el determinar los elementos hardware de los que disponemos y poner especial cuidado en aquellos aspectos que nos impongan algún tipo de restricción, ya sea en funcionalidad, resolución o temporización. Dado que también se ofrece una capa de abstracción hardware, en el análisis obviaremos los detalles de control y configuración que cubre dicha capa de abstracción hardware y al final haremos un breve análisis de dicha capa de abstracción.

### Hardware del primer prototipo

#### Microcontrolador

La plataforma cuenta con un MSP430FR5972 con frecuencia máxima de trabajo de 16MHz y una capacidad de memoria de 64Kb de FRAM para código y 2Kb de RAM, y es utilizado para las funciones de control, calculo y comunicación con otros elementos hardware. Dicho microcontrolador cuenta con multiplicador hardware de 32 bits y carece de unidad de coma flotante, lo cual nos impone el que las operaciones aritméticas se hagan en coma fija.

En cuanto a la comunicación con los demás elementos hardware, cada elemento cuenta con su propio puerto salvo el Analog Front End y el Fuel Gauge que están conectados al mismo puerto (aunque en distintos pines), por lo que habrá que tener cuidado de reconfigurar el *pinout* del puerto adecuadamente cada vez que accedamos a cada uno. Para las operaciones de acceso directo a memoria, tenemos tres canales de *Direct Memory Access*(DMA).

Con respecto a los requisitos de tiempo real, debido a la cantidad de reducida de memoria RAM de la que dispone este microcontrolador, nos obliga a usar una solución basada en ejecutivo cíclico, ya que no se han encontrado sistemas operativos para sistemas empotrados que puedan funcionar con una cantidad tan pequeña de RAM.

Por último, cabe comentar que dispone una unidad de encriptación hardware (AES256) que nos evitará tener que implementar una encriptación por software.

#### Adquisición de datos

Para la digitalización de la señal cardiaca, se utiliza un Analog Front End especializado para aplicaciones de electrocardiograma (ADS1291), con una velocidad máxima de muestreo de 8000 muestras por segundo y 24 bits por muestras, lo cual es más que suficiente para los requisitos de resolución del algoritmo. Este periférico cuenta con una línea de interrupción para indicar cuando los datos están listos (lo cual nos evitar tener que tener que hacer *polling* de la señal de *data ready*) y transmite los datos por SPI.

#### Comunicación inalámbrica

El dispositivo utilizado para la comunicación inalámbrica es el CC2560B, que es un circuito integrado que soporta BT ® clásico, lo cual nos impone el uso de la pila de protocolos para esa tecnología y nos ofrece la posibilidad de establecer comunicaciones autenticadas para la transmisión de los datos necesarios entre el dispositivo de acceso y el simulador de marcapasos. Este módulo se comunica por un puerto UART con control hardware de flujo.

#### Interfaz de usuario

Como elementos de interacción con el usuario, cuenta con tres leds de color rojo, verde y azul, y un *buzzer* para emitir avisos sonoros y visuales básicos, un interruptor para cortar la alimentación. El PWM necesario para el buzzer está generado por uno de los temporizadores del microcontrolador y los leds se controlan con entradas/salidas de propósito general.

También disponemos *display* LCD que ofrece la posibilidad de mostrar todo tipo de información visual, desde mensajes de texto hasta representación de señales digitales. Dicho display además cuenta con una pantalla táctil resistiva, lo cual nos permite implementar un intérprete de comandos para incrementar la interacción que se tiene con el sistema. El envió de comandos al sistema se hará asociando cada zona de la pantalla a cierto comando, de manera que cuando se detecte que dicha zona ha sido pulsada, se enviara el comando asociado al interprete.

El LCD no tiene un puerto serie hardware específico para él, sino que se ha tenido que implementar un puerto paralelo software mediante varios gpios, lo cual limita la tasa de refresco. El touchpad se comunica por SPI y dispone de una línea de interrupción para avisar cuando hay un cambio de presión en la pantalla.

#### Control de consumo

Para este objetivo contamos con dos elementos principalmente: el primero de ellos son los modos de bajo consumo del microcontrolador, y el segundo es la disposición de *LDO’s* independientes para cada módulo, lo cual nos permite apagar de forma independiente cada elemento hardware, pudiendo así reducir el consumo sin necesidad de apagar completamente el sistema.

### Capa de abstracción hardware del primer prototipo

Esta capa de abstracción hardware está basada en los registros definidos en el *header* de nuestro micro (msp430fr5972.h) que proporciona el entorno de desarrollo de Texas Instruments Code Composer Studio.

Esta capa cuenta con funciones que permiten la configuración y el uso de los distintos módulos hardware utilizados en el sistema hardware, lo cual permite construir todo el sistema alrededor de dichas funciones.

#### Funciones de configuración

Estas funciones de configuración nos permitirán configurar los periféricos del sistema con los valores necesarios para su correcto funcionamiento e interacciona con los módulos hardware, y también nos permitirán configurar los módulos hardware con los parámetros que nosotros deseemos. Dichas funciones agrupadas por bloques son las siguientes:

* Relojes del sistema: Se encarga de configurar los registros necesarios para poder trabajar a la frecuencia escogida (8 MHz). Adicionalmente, se encargará de seleccionar las fuentes de reloj para cada una de las salidas de reloj que utiliza el microcontrolador para proporcionar el reloj de los módulos hardware que lo necesitan (AFE y módulo de radiofrecuencia)
* *Analog Front End*: Configura el puerto SPI para que se adecue a las necesidades del front end (velocidad, flanco de lectura, etc.) y también se encarga de enviar la secuencia de comandos necesaria para configurar los registros del Analog Front End(AFE). Esta secuencia de inicialización incluye la selección de velocidad de muestreo, calibración y configuración de modo de envío de datos (envió continuo por interrupciones).
* Interfaz inalámbrica: Realiza la configuración de la UART a través de la cual nos comunicaremos con el modulo radio.
* Buzzer: Esta función configura los registros del timer utilizado para generar el *pwm* y también los del que se utiliza como base de tiempos para la duración del pitido.
* Pantalla táctil: Estas funciones engloban tanto la configuración del SPI para el touchpad como la secuencia de inicialización del display.

#### Funciones de uso de periféricos

Estas funciones encapsulan las instrucciones necesarias para la interacción del microcontrolador con los módulos hardware. Estas instrucciones van desde el control de las señales de “preparado” de algunos periféricos como el SPI, de manera que se bloquea el envío hasta que dichos periféricos estén listos, hasta la gestión de las señales de control, como por ejemplo el *chip select* del AFE. Estas funciones se agrupan de la siguiente forma

* Envío/recepción de datos: Estas funciones gestionan la recepción y transmisión de datos al periférico para el que están implementadas. Todas ellas están diseñadas de manera que bloquean la ejecución hasta que se completa la operación, lo cual debe ser tenido en cuenta a la hora de la temporización. Aunque el uso es el mismo, según el periférico al que se dirijan, deben gestionar además ciertas señales:
  + Analog front end y touchpad: Gestiona la activación del *chip select*
  + LCD: Gestiona la generación y temporización de todas las señales necesarias (referencia al *datasheet*)
* Activación del buzzer: Permiten la emisión de un pitido con el buzzer de una frecuencia y duración determinadas.

### Hardware del segundo prototipo

Como se ha comentado antes, fue necesario diseñar un segundo prototipo hardware que nos permitiera seguir el proyecto. Dicho hardware tiene diversas mejoras, pero para nuestro trabajo, solo nos influye el cambio de microcontrolador y el cambio de display.

Desde el punto de vista de diseño, el cambio de microcontrolador supone reducir las restricciones de memoria, con lo que podemos permitirnos utilizar un sistema operativo de tiempo real para cumplir los requisitos de tiempo real.

Por otro lado, el cambio de LCD no supone demasiado cambio, ya que el controlador de la pantalla tactil es el mismo y la comunicación se hace también por un puerto paralelo de 16 bits de datos.

Dado que el segundo prototipo tiene en común varios elementos, solo detallaremos las diferencias.

#### Microcontrolador

La plataforma cuenta con un STM32L162VD que ofrece mayores prestaciones en diversos aspectos, algunos de los cuales son especialmente interesantes de cara al servicio.

Primero, su frecuencia de trabajo máximo es mayor (32MHz) así como su memoria para código (384 Kb de flash) y RAM (64Kb).

Por otro lado, para el control del LCD disponemos de un puerto paralelo hardware, lo cual nos va a permitir alcanzar velocidades sensiblemente mayores. En cuanto al resto de periféricos, cada uno cuenta con su propio puerto serie y además disponemos de seis canales de DMA, tres más que en el anterior.

(…) cambiar alguna cosa cuando este el micro nuevo

#### Interfaz de usuario

Aunque el esquema es similar (*leds* + *buzzer* + LCD con touchpad resistivo) tenemos ciertas diferencias respecto el otro prototipo.

El trio de led ha sido sustituido por un único led *rgb*, y el interruptor se ha cambiado por un pulsador conectado a una línea del *wakeup* del microcontrolador, lo que nos permitirá implementar un “apagado” por software y alguna otra funcionalidad básica, como puede ser apagar la luz de retroiluminación de la pantalla, de una manera similar al pulsador que tienen los Smartphone.

### Capa de abstracción hardware del segundo prototipo

En este caso, esta capa está basada en el código de inicialización generado por el programa STM32CubeMX, el cual permite configurar todos los periféricos del microcontrolador y proporciona funciones para el uso de los puertos serie (entre otras muchas cosas).

La funcionalidad que ofrece esta nueva capa es idéntica a la del primer prototipo, salvo que en este caso no bloquean la ejecución del programa principal, ya que utilizan las funciones que ofrece el código de inicialización diseñadas para este propósito, bien mediante el uso de interrupciones o acceso directo a memoria.

## Visión general del sistema

En la siguiente figura se puede ver un esquema de la estructura del sistema software en que se detalla el nivel de separación entre la capa de abstracción hardware que proporciona el proyecto conjunto y el software que vamos a diseñar nosotros.

(FIGURA TOPE MOLONA)

El funcionamiento de este servicio será diferente según se ejecute en el dispositivo de acceso o en el IMD, pero dado que ambas plataformas comparten la gran mayoría de la funcionalidad, se ha decidido diseñar un único sistema, el cual después incluirá o no ciertas partes según se trate del dispositivo de acceso o el simulador de marcapasos.

Debido a que este es un sistema de uso puntual, la mayor parte del tiempo se encontrara en estado de suspensión para reducir el consumo, lo cual es especialmente crítico en un hipotético IMD, donde la recarga de la batería es especialmente problemática.

Durante el acceso, el simulador de marcapasos entrará en modo escucha para poder ser detectado por el dispositivo de acceso, el cual establecerá una comunicación autenticada mediante un proceso que se detallará más adelante. Una vez el simulador de marcapasos nos haya concedido el acceso, el dispositivo ofrece una serie de opciones que simulan la manipulación de parámetros del marcapasos. Finalmente, el dispositivo cierra la conexión enviando un comando que “apaga” el modulo radio del simulador.

## Arquitectura Software del primer prototipo

La arquitectura de este sistema está basada en la capa de abstracción hardware que se nos proporciona. Nuestro objetivo será el de ampliar la funcionalidad de dicha capa para que después podamos implementar a nivel de aplicación nuestro protocolo. El software se divide en tres niveles que son los siguientes:

* *Hardware Abstraction Layer* (HAL): Implementada en el proyecto “Diseño e implementación de una plataforma hardware para un sistema de acceso inalámbrico a dispositivos médicos mediante Heart-To-Heart”, consiste en una serie de archivos .c que permiten la configuración y gestión de los periféricos del microcontrolador, así como la configuración inicial del propio microcontrolador.
* Drivers: Son los módulos software expanden la funcionalidad de comunicación con los módulos hardware que se nos proporciona en la HAL. Esta ampliación será más detallada a continuación, pero consiste sobretodo en el tratamiento de los datos recibidos por parte de los periféricos que los que recibidos datos y el encapsulamiento de algunas instrucciones para una tarea concreta, como puede ser el pintar una línea en el LCD
* Aplicación: Esta capa se encarga de controlar las funciones de más alto nivel, como son el algoritmo de detección de ritmo cardiaco, la interfaz de usuario, la inicialización de la pila de protocolos, etc.

## Diseño del software del primer prototipo

### Adquisición y tratamiento de la señal

El diseño de este módulo incluye todo lo referente a la sincronización de la lectura de datos del AFE, la adaptación de dicha señal para se adecue al algoritmo de detección del ritmo cardiaco y el propio algoritmo de detección de ritmo cardiaco.

#### Adquisición

La lectura de los datos a través del puerto SPI se gestionará a través de la interrupción de la línea DRDY de la que dispone el AFE. Dicha línea indica en que instante están disponibles los datos leídos y está conectada a una entrada/salida de propósito general que puede configurarse como línea de interrupción externa.

Dicha interrupción se encargará leer los 24 bits de datos que envía el AFE, formatearlos los tres bytes recibidos en un único entero con signo para que sea manejable por el resto de funciones y los escribirá en un buffer circular para que sean accesibles a los otros consumidores de dichos datos. Estos consumidores serán la etapa de procesado digital y la interfaz de usuario, la cual los utilizara para la muestra del ECG en el display.

#### Procesado digital

El procesado digital tiene como objetivos eliminar la mayor cantidad de ruido que pueda afectar a señal, tanto fisiológico (respiración, ruido electromiografico, etc.) como electromagnético (interferencia de red) y también resaltar la información que facilite la detección del ritmo cardiaco. La estructura de este procesado se ha hecho atendiendo al paradigma que siguen las etapas de procesado de señal que tienen todos los algoritmos de detección de ritmo cardiaco que se han encontrado en la bibliografía.

Dicha estructura se basa en una primera etapa en la que se emplean filtros digitales para la eliminación de ruido. Esta etapa siempre empieza con un filtro paso alto para eliminar el *baseline wander* y le sigue un filtro paso bajo para eliminar el ruido de alta frecuencia que cae fuera del ancho de banda de nuestra señal. Algunos esquemas limitan el ancho de banda en el filtro paso bajo a aproximada 40 Hz, de manera que no es necesario tratamiento adicional para la eliminación de la interferencia de red, pero dado que nosotros disponemos de una etapa de filtrado en el AFE que dispone de dicha interferencia, respetaremos todo el ancho de banda.

Después de este filtrado, tenemos la etapa de resaltado de la onda R. Esta etapa sí que varía mucho entre esquema y esquema, desde alguno en que directamente no lo incluye y basa su detección en la amplitud tal cual, hasta algunos que implementan incluso filtrado no lineal. Para nuestro diseño, vamos a utilizar un esquema intermedio en el que la esta etapa va a consistir en un filtro derivador.

La estructura en la que se organiza las distintas etapas de procesado se muestra en la figura X

(aquí esquema de los filtros)

* Filtro paso alto: Este bloque tiene dos funcionalidades: por un lado, elimina la tensión continua necesaria para la polarización del amplificador en modo común y el fenómeno de *baseline wander* (tensión de baja frecuencia causa por un mal contacto de los electrodos que modula la señal). Dicho filtro filtrará las frecuencias entre 0 y 1 Hz, será de tipo FIR, dado estos filtros son de fase lineal, y su orden estará limitado a 200.
* Filtro paso bajo: Este bloque se encarga de eliminar de filtrar la señal por encima de los 145 Hz, ya que no contiene información útil a nuestro algoritmo y solo añade ruido. Sera de tipo FIR y el orden estará limitado a 200.
* Filtro diferenciador: Este bloque resalta las transiciones bruscas, como la del complejo QRS que es el que queremos detectar.

##### Selección del tipo de filtros

Como se ha comentado en los requisitos, la selección de filtros ha de hacerse con especial cuidado, ya que la fase de estos filtros puede distorsionar la señal y llevar a una incorrecta detección del ritmo cardiaco.

La elección más evidente para evitar esto es el uso de filtros de respuesta la impulso finita(FIR), los cuales presentan no solo la ventaja de ser de fase perfectamente lineal, sino que además son insensibles a la cuantización de los coeficientes, que era otro de los problemas que se nos planteaba. No obstante, el uso de este tipo de filtros es prácticamente inviable cuando el ancho de la transición entre la banda de paso y la eliminada en muy estrecha (como sería el caso del filtro paso alto) ya que el orden del filtro se dispara para conseguir atenuaciones moderadas. En este caso, se plantea el uso de filtro de respuesta al impulso infinita (IIR) al que se aplican técnicas de lineal izado de fase para mitigar el efecto de la distorsión de fase.

Teniendo esto en cuenta, para el filtro paso bajo y diferenciador, los cuales no tienen restricción de banda de transición estrecha, se ha seleccionado el diseño con un filtro tipo FIR con lo que conseguimos los siguientes resultados:



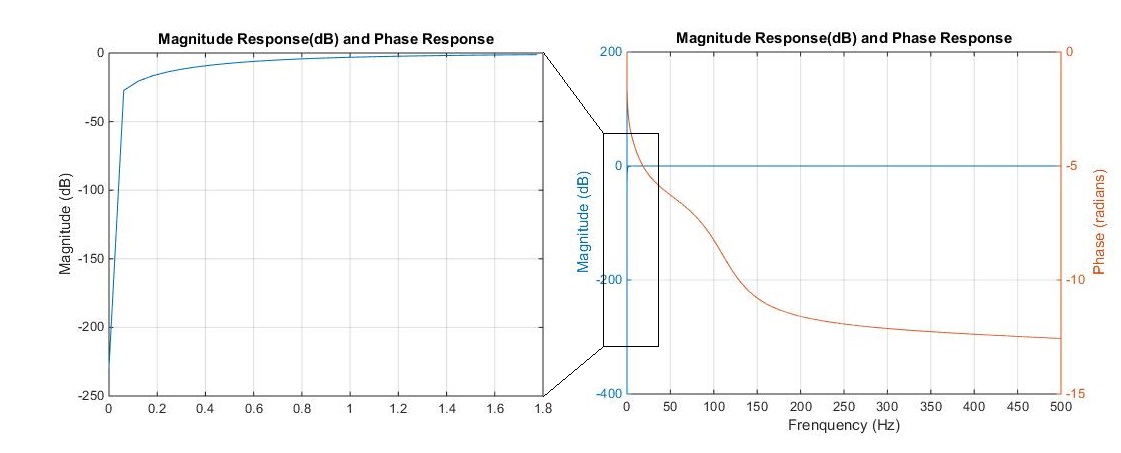
Figura X. Respuesta en amplitud y fase del filtro diferenciador



Figura X. Respuesta en amplitud y fase del filtro paso bajo

Como se puede observar, obtenemos la respuesta en amplitud que buscamos, tanto en banda como en rizado, el cual está limitado a 0.1 dB, y una fase perfectamente lineal. El orden del filtro paso bajo es de 140 y el del diferenciador es de 10, lo cual cumple nuestras restricciones de diseño

Para el filtro paso alto, se ha elegido usar un filtro IIR con un ecualizado de fase basado en un filtro paso todo de orden 4, el cual linealiza la fase en la banda de paso. El motivo por el cual el filtro ecualizador no se ha elegido de mayor orden ha sido porque con la perdida de precisión que supone la cuantificación de los coeficientes provoca que para un orden mayor la respuesta variara tanto que en vez de sintonizar, distorsionaba más a aun la fase. La respuesta conjunta de filtro paso alto y sintonizador es la siguiente:



En este caso, si bien la respuesta en amplitud es la buscada, la fase tiene una ligera distorsión. Esta distorsión podría ser corregida si el microcontrolador que usamos dispusiera de unidad de coma flotante hardware y por lo tanto pudiéramos diseñar un ecualizador de fase de mayor orden. No obstante, las pruebas realizadas más adelante muestran que esta pequeña distorsión de fase es aceptable para nuestra aplicación, y por lo tanto el diseño es válido.

#### Detección de ritmo cardiaco

##### Umbrales estáticos

##### Umbrales dinámicos

### Algoritmo de autenticación

#### Uso del ritmo cardiaco como contraseña dinámica

#### Estimación del umbral de validación

### Comunicación inalámbrica

#### Pila de protocolos

Como ya se ha dicho en el análisis del hardware, la tecnología para la comunicación es bluetooth clásico, en concreto la versión 4.1, para el cual Texas Instruments ofrece una pila de protocolos para bluetooth (Bluetopia).

Aprovechando esto-, nuestro diseño se va a centrar en hacer los cambios necesarios para portar dicha pila a nuestro microcontrolador. Estos cambios están especificados en la documentación que incluida cuando te descargas la pila.

#### Gestión de la red por el master

#### Conexión segura

### Interfaz de usuario

#### Visualización de la señal ECG

#### Inicio de comunicación con un IMD

### Módulo de gestión de consumo

#### Apagado por software

#### Control de consumo de la pantalla

#### Apagado de elementos hardware

#### Gestión de consumo en RF

## Arquite+

## ctura Software del segundo prototipo

El software se divide en una serie de niveles en los que podemos distinguir:

* Hardware Abstraction Layer (HAL): Implementada en el proyecto DER SAMUE, consiste en una serie de librerías que permiten la configuración y gestión de los periféricos del microcontrolador, así como la configuración inicial del propio microcontrolador.
* Drivers: Son los módulos software que permiten configurar y utilizar los distintos módulos hardware, como puede ser el AFE, la pantalla o el medidor de batería.
* Middleware: Este módulo consiste en un sistema operativo de tiempo real(FreeRTOS) que se encarga de sincronizar los distintos módulos software.
* Aplicación:

## Diseño del software del segundo prototipo

### Adquisición y tratamiento de la señal

En diseño de este módulo software

#### Filtrado digital

##### Selección del tipo de filtros

#### Detección de ritmo cardiaco

##### Umbrales estáticos

##### Umbrales dinámicos

### Algoritmo de autenticación

#### Uso del ritmo cardiaco como contraseña dinámica

#### Estimación del umbral de validación

### Comunicación inalámbrica

#### Pila de protocolos

Como ya se ha dicho en el análisis del hardware, la tecnología para la comunicación es bluetooth clásico, en concreto la versión 4.1, para el cual Texas Instruments ofrece una pila de protocolos para bluetooth (Bluetopia).

Aprovechando esto, nuestro diseño se va a centrar en hacer los cambios necesarios para portar dicha pila a nuestro microcontrolador, ya que esta se está diseñada para la serie L4. Estos cambios están especificados en la documentación que incluida cuando te descargas la pila.

#### Gestión de la red por el master

#### Conexión segura

### Interfaz de usuario

#### Visualización de la señal ECG

#### Inicio de comunicación con un IMD

### Módulo de gestión de consumo

#### Apagado por software

#### Control de consumo de la pantalla

#### Apagado de elementos hardware

#### Gestión de consumo en RF

# Implementación de los prototipos software

## Primera versión

### Implementación de las etapas de tratamiento de la señal

### Filtro paso banda FIR

### Diferenciador

### Elevar al cuadrado punto por punto

### Integrador

### Interfaz de usuario

#### Presentación de la señal ECG

#### Problemas encontrados

### Segunda versión (persimmon)

#### Filtrado digital

##### Optimización en los filtros

### Algoritmo de autentificación

#### Extracción de los bits de alta entropía

#### Calculo de distancia hamming y umbral de validación

#### Etapas de emparejamiento master-slave

#### Modo promiscuo

### Interfaz de usuario

#### Pantalla táctil

#### Menú

#### Indicador de batería

### RF

#### Pila de protocolos

#### Gestión de la red por el master

#### Conexión segura

### Gestión de consumo

#### Apagado por software

#### Control de consumo de la pantalla

#### Apagado de elementos hardware

#### Gestión de consumo en RF

# Pruebas y caracterización

## Pruebas con un generador de señal cardiaca

## Pruebas con un individuo real

## Respuesta del modo promiscuo

## Respuesta a un ataque

# Manual de uso

# Conclusiones y líneas futuras

## Seguridad

## Mejora sobre otros sistemas actuales

## Mejoras/Otros posibles usos

### Detección automática de segmentos del ECG

### Cálculo del eje eléctrico

### Ayuda al diagnóstico

### Detección de marcapasos

# Apéndices

# Referencias

[1] “Heart-to-Heart (H2H): Authentication for Implanted Medical Devices”

[2] “Signal Processing Methods for Heart Rate Variability”

[3] Compromising a Medical Mannequin