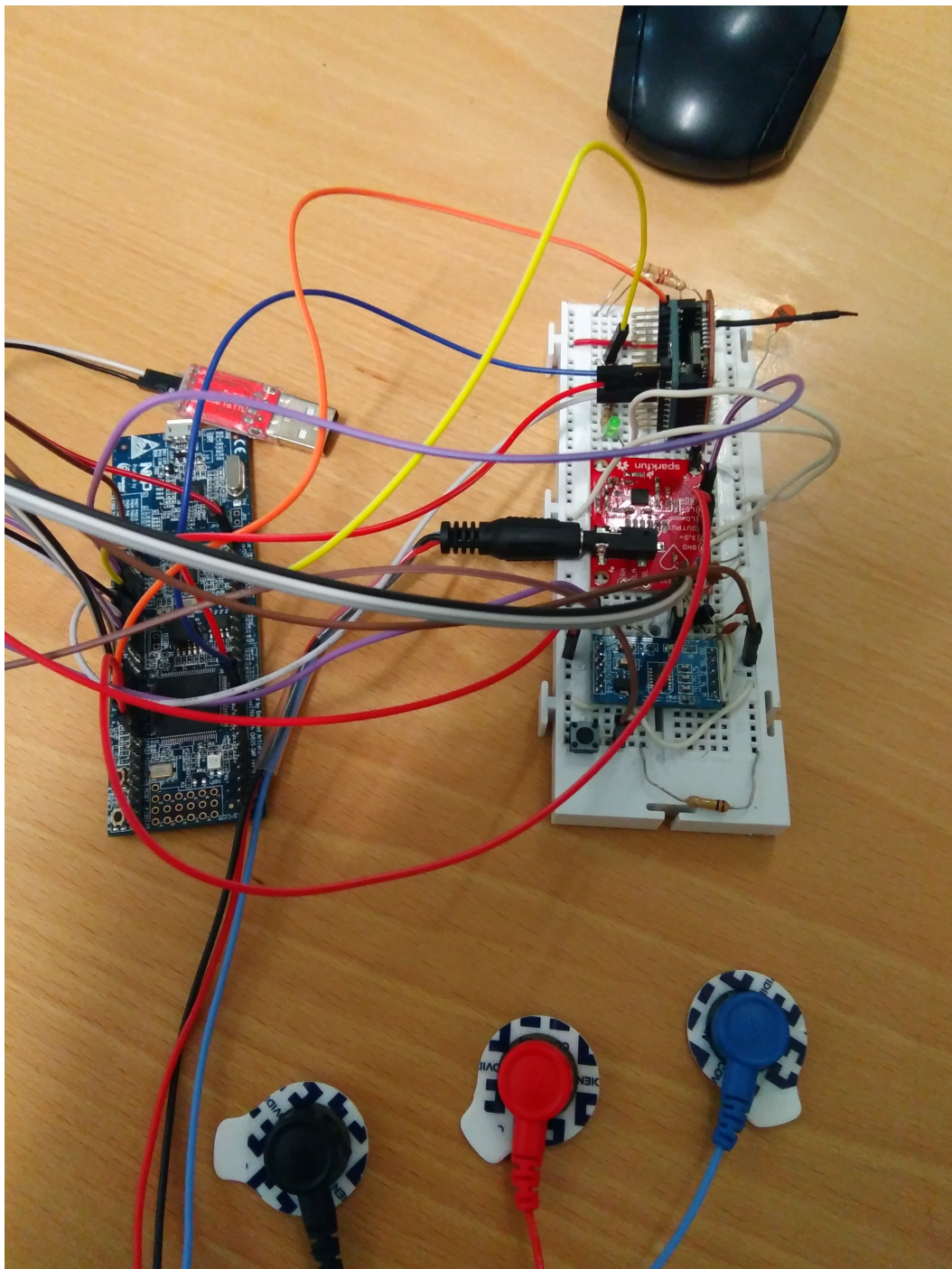


## Entrega final del código

Alumno : César López Bermúdez



## **Introducción**

Se ha concluído finalmente el desarrollo del proyecto como PoC del sistema global propuesto. Adjunto al presente documento se entrega copia del *workspace*. Para la comprobación del correcto funcionamiento del DMP en cuanto a comunicaciones vía *wifi* se refiere, se ha implementado un pequeño servidor muy simple en PHP que escucha el socket y procesa los datos que recibe. Dichos son trasladados a una base de datos implementada en MySQL bajo un servidor local WAMPP. No obstante en la versión que se envía en esta entrega, se han comentado las líneas de código que actúan sobre la base de datos, de modo que el servidor mostrará los datos recibidos por pantalla, debidamente formateados en caso de tratarse de alarmas o señales de información, o de forma masiva en caso de tratarse del envío de un stream procedente del pulsómetro. Se adjunta fichero de extensión *.bat* para lanzar el servidor mediante la consola de Windows.

## **Objetivos**

Tal y como se ha especificado en documentos previos los objetivos del proyecto se jerarquizan en tres niveles, se muestran en color verde los conseguidos:

### **Objetivos básicos**

Desarrollar un sistema embebido que satisfaga la necesidad descrita incluyendo:

- Funcionamiento en modo *indoor*.
- Botón de pánico pulsable por el usuario y generador de alarma activando LED rojo(Led y alarma se desactivarán tras una segunda pulsación del botón).
- Sensorización de movimiento(accelerómetro): que permita detección de desvanecimientos, sujeto inusualmente inmóvil, etc...
- Procesado de datos adquiridos y generación de alertas.
- Integrar conectividad vía Wifi con nodo central.
- Lograr la comunicación efectiva de alertas generadas por el DMP con nodo central.

### **Objetivos secundarios**

- Integrar sensorización de pulso cardíaco y de temperatura.
- Lograr un sistema que facilite posibles ampliaciones futuras en materia de sensorización.
- Realizar un sistema que permita la ampliación (modo outdoor, etc...) e integración en el sistema completo en caso de que éste resulte viable.

### **Objetivos extra**

- Explorar otras opciones en cuanto a sensorización se refiere que permitan dotar al sistema de otras funcionalidades.(presión sanguínea, medición de glucosa en sangre, etc...).

- **Analizar viabilidad técnica de una posible y futura ampliación del DMP que incluya el modo de funcionamiento *outdoor* y lo integre en un sistema completo descrito al inicio de este documento.**

Tal y como se observa se han logrado los objetivos establecidos. Por lo que respecta a los extras mostrados en color naranja, su consecución se reflejará en la memoria de proyecto ya que no tienen aplicación directa en el código y por tanto no afectan a esta entrega.

### **Consideraciones generales**

A lo largo del desarrollo del proyecto se han aplicado diversas medidas de contingencia tal y como se ha descrito en las correspondientes PACs, y que se referenciarán debidamente en la memoria de proyecto. En la tercera fase ha sido preciso activar la medida correctora correspondiente al riesgo #2 debido a no viabilidad del sensor de temperatura elegido siendo sustituido por el sensor TMP36 dentro del plazo de dos días establecido para dicha medida incluido el tiempo de envío del sensor por parte del proveedor(24 horas). Esto ha implicado el uso de un nuevo canal del ADC para el procesamiento de las medidas realizadas. Según se ha establecido al inicio del proyecto, este consta de las siguientes tareas a realizar, mostrándose en verde las finalizadas.

#### **Fase uno.**

- Desarrollo de un módulo que permita monitorizar (Log) el funcionamiento del sistema a través UART (CP2102) y de un terminal(putty o Teraterm).
- Botón del pánico generando la consiguiente alarma.
- Acelerómetro permitiendo detectar caídas, o situaciones de inmovilidad prolongada.
- Desarrollo de la lógica de procesado de datos y generación de acciones en base a éstos(generación de alarmas, envío de datos a través al Log)

**Fase dos. En esta fase se ampliará el sistema incluyendo las siguientes mejoras:**

- Desarrollo del driver del sensor de pulso cardíaco.
- Integración de módulo Wifly para envío de datos al servidor mediante socket.
- Creación de servidor para recepción de alarmas mediante socket.

**Fase tres. En la fase tres se incrementaron funcionalidades incorporando más sensores y habilitando la posibilidad de una futura extensión del proyecto al modo outdoor:**

- Desarrollar driver e integrar sensor de temperatura en el sistema.
- Implementar detección de pérdida de conectividad wifi habilitando conmutación de modo de operación del sistema(de indoor a outdoor).

Tal y como se observa se han completado todas las tareas. No obstante se ha ampliado el alcance de varias de las tareas aumentando así el desarrollo del proyecto. Concretamente se ha ampliado el alcance de las siguientes tareas(ampliación mostrada en color azul):

#### **Fase 1**

- **Desarrollo de un módulo que permita monitorizar (Log) el funcionamiento del sistema a través UART (CP2102) y de un terminal(putty o Teraterm). Además de la función propia del Log para mostrar por terminal información relativa al funcionamiento del**

sistema, se ha añadido funcionalidad a modo de Shell con filtros, de modo que puede decidirse que mensajes se deben mostrar en función de su origen(General, termómetro, ADC, acelerómetro, etc....)

- **Acelerómetro permitiendo detectar caídas, o situaciones de inmovilidad prolongada.**  
Se añada el envío cada 60 segundos aproximadamente de datos de monitorización de actividad y esfuerzo. Concretamente se envía el número de veces que el usuario realiza esfuerzos sobre su centro de gravedad(agacharse, inclinarse en gran medida hacia los lados o atrás,...) y un valor suma de las aceleraciones experimentadas en los ejes Z(sentido de la marcha al caminar) y X(giros sobre sí mismo) excepto en situaciones de caída o inactividad que pueden proporcionar valores sin utilidad y siempre que los valores a enviar no sean nulos.
- **Desarrollo de la lógica de procesamiento de datos y generación de acciones en base a éstos.**  
En consonancia con la ampliación de la anterior tarea, también se ha ampliado la lógica de procesamiento y transmisión de datos, permitiendo enviar además de las alarmas(caída, inactividad...) señales de datos(indices de esfuerzo, variaciones de temperatura ambiente determinadas,...) y stream de datos para transmitir secuencias de pulso cardíaco (ECG) muestreadas para su análisis en tiempo real por personal sanitario de forma remota.

**Fase dos.**

- **Integración de módulo Wifly para envío de datos al servidor mediante socket.**  
Además de las conectividad propuesta inicialmente, se ha añadido una nueva funcionalidad que aprovecha el cliente sNTP del dispositivo Wifly para añadir un TimeStamp a cada alarma o señal enviada.

**Fase tres.**

- **Desarrollar driver e integrar sensor de temperatura en el sistema.**  
En lugar de implementar una única función que posibilite generar alarmas en caso de exposición a temperaturas extremas, se han implementado dos funcionalidades. Por un lado la librería permite definir seis límites que definen áreas de temperatura(en la aplicación se usan 4) de modo que si se efectúa una transición de una zona a otra el sistema genera señal indicando la nueva zona y la temperatura medida. Por otro lado se permite establecer un valor de temperatura en grados °C de modo que se generará señal de datos con TimeStamp y valor de temperatura cada vez que esta presente una variación superior a dicho valor a fin de monitorizar con mayor resolución la exposición de la persona. Se define también el tiempo de exposición mínimo para generar la alarma correspondiente.
- **Implementar detección de pérdida de conectividad wifi habilitando conmutación de modo de operación del sistema(de indoor a outdoor).** Además de la conmutación entre modos *INDOOR* y *OUTDOOR* se ha desarrollado un nuevo modo *HYBRID*, que permitiría postergar el uso de GPS en un hipotético sistema ampliado de modo que si el sistema emplearía GSM para comunicaciones pero Wifi para posicionamiento mientras sea capaz de recibir alguna red wifi vecina al domicilio del usuario.

A continuación se describen brevemente los módulos más relevantes:

### **Descripción del módulo acelerómetro.**

Consiste en una tarea que realiza llamada a una función de procesamiento de datos del acelerómetro leídos por el ADC:

Esta función principal del módulo acelerómetro emplea una estructura de datos enviada(puntero desde main.c) y a partir de dichos datos calcula el valor de las aceleraciones en cada eje que serán

utilizadas para decidir el lanzamiento de la alarma de inactividad y las relaciones angulares que servirán para detectar las posibles caídas del usuario además de datos de monitorización(actividad).

### **Descripción del módulo Termómetro**

En el fichero de cabecera se establecen las zonas térmicas a considerar y el tiempo mínimo de exposición para producir una alarma o señal. Otro parámetro que se puede configurar es la variación mínima de temperatura independientemente de las zonas definidas para que se envíe datos de monitoreo(temperatura medida, TimeStamp, zona).

### **Módulo monitor**

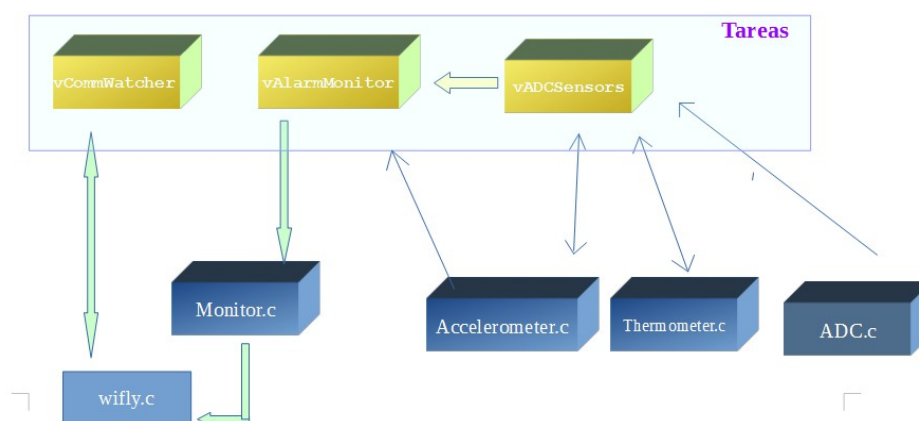
Encapsula el modo de operaciones del sistema(indoor, hybrid, outdoor), encapsula la cola de transmisión de señales y alarmas y se gestiona su retransmisión si es procedente(reingresa en la cola)

### **Descripción del módulo de conectividad.**

El módulo de conectividad se basa en la tarea **vCommWatcher()** que se encarga de revisar el estado de las comunicaciones cada cierto período de tiempo definido en **mMILIS\_COMM**. Esta tarea gestiona la conmutación entre los tres modos posibles de funcionamiento del sistema.

La tarea realiza tres funciones:

- Ordenar la conmutación entre los modos de funcionamiento según las comunicaciones disponibles.
- Crear y actualizar cada mMILIS\_UPDFW milisegundos la tabla de redes Wifi accesibles desde la vivienda del usuario.
- Generar la señal de alarma de sistema de modo que el nodo central(NCG) sepa en tiempo real la situación del usuario.



### **Modificaciones según sesiones Skype**

Tras lo comentado en las tres sesiones de Skype con el tutor de la asignatura se han realizado las siguientes mejoras:

- Mejorar el encapsulado.(uso de static y adición de métodos)
- Ordenar ficheros .h de la librería, descongestionando config.h. Se han añadido comentarios.
- Emplear nomenclatura Módulo\_Método().

### 1. ¿Cual ha sido el punto más difícil de la PAC

Tal y como ocurría en la PAC 2 la parte más dura y complicada es la interacción con el hardware y el desarrollo de los drivers y manipulación del módulo Wifly a través de la UART del LPC1769.

La configuración del cliente NTP ha generado problemas a la hora de configurar la zona horaria. El módulo no acepta el formato propuesto en la documentación y ha sido necesario conectar directamente el módulo Wifly al PC a través del conversor USB-UART para efectuar pruebas y obtener el resultado.

### 2. Errores en el sistema.

El sistema no muestra errores de compilación y funciona correctamente, no obstante no ha sido posible obtener la hora mediante NTP exactamente. Se obtiene el TimeStamp con una hora menos.

### 3. Warnings en compilación o volcado al LPC1769

Surgen 4 warnings en la compilación del proyecto. Estos son debidos a una asignación sin cast a la hora de adquirir el número serie del LPC1769(Utils\_ReadSerialNumber), no obstante el funcionamiento del software es correcto no presentando mal funcionamiento.

### 4. Tests

Para realizar los tests de la parte de drivers se han realizado pruebas por driver de forma independiente unos de otros forzando temperaturas(hielo, calefactores,...) para comprobar el comportamiento correcto del termómetro, realizando inclinaciones manuales del acelerómetro y todo ello modificando los valores umbrales para forzar situaciones límite. Para test del módulo de conectividad y su tarea vigilante, se han empleado dos teléfonos móviles como AP en ausencia de cualquier otra wifi para comprobar la conmutación entre modos y la recepción de datos en el servidor PHP. Para el test de la lógica del sistema se han proporcionado mediante el propio código valores ficticios para comprobar que dicha lógica funcionaba correctamente(ejemplo: introducir valores ficticios en la lógica del termómetro desde el main para comprobar que conmuta entre zonas y genera la alarma)

### 5. Tiempo dedicado.

El tiempo total empleado en el desarrollo de los objetivos descritos en este documento ha sido de 3 o 4 horas en función del día durante 5 días hábiles por semana y 14 semanas resultando un total de 280 horas.

6. **¿Se han cumplido los objetivos?** Se han cumplido los objetivos establecidos y ampliado el alcance de algunas tareas.