# Implementación de una Plataforma Ubicua de Monitorización de Pacientes basada en el Estándar ISO/IEEE11073

J. Escayola[1], M. Martínez-Espronceda[2], I. Martínez[1], L. Serrano[2], J.D. Trigo[1], S. Led[2], J. García[1]

[1] Instituto de Investigación en Ingeniería de Aragón (Univ. Zaragoza) - c/ María de Luna, 3. 50018 - Zaragoza, Spain

[2] Dep. Ingeniería Eléctrica y Electrónica (Univ. Pública Navarra) - Campus de Arrosadía s/n. 31006 - Pamplona, Spain

{jescayola, imr, jtrigo, jogarmo}@unizar.es, {miguel.martinezdeespronceda, lserrano, sled}@unavarra.es

## Resumen

*Este artículo aborda la implementación de una plataforma ubicua de monitorización de pacientes basada en el estándar ISO/IEEE11073 (X73) para interoperabilidad de dispositivos médicos. Para ello, se analiza la evolución tecnológica de X73, orientada a entornos ubicuos y dispositivos llevables (Personal Health Devices, X73-PHD), y abierta a nuevas funcionalidades. Tras presentar la arquitectura básica, se detalla la implementación de la plataforma que posibilita el desarrollo de sistemas end-to-end basados en estándares. Se analiza el diseño e implementación del modelo agente-manager, particularizado en X73-PHD al protocolo de comunicación entre un dispositivo médico (Medical Device, MD) y un gateway concentrador (Compute Engine, CE). Por último, se valoran los resultados obtenidos orientados a los nuevos casos de uso para entornos ubicuos y a la implantación sobre dispositivos wireless, objetivo clave dentro del Comité Europeo de Normalización.*

## 1. Introducción

En los últimos años las implementaciones de servicios de e-Salud se han enfocado al control centralizado de los pacientes en los hospitales o a su posterior seguimiento domiciliario. Sin embargo, cada vez son más los casos de uso que demandan servicios ubicuos, independientes de la localización, en los que las nuevas tecnologías se adapten al escenario específico creado alrededor del paciente que se coloca en el centro del sistema sanitario. Así surge el concepto de redes de área personal y corporal (*Personal/Body Area Network*, PAN/BAN) y el uso de nuevos dispositivos médicos personales (*Personal Health Devices*, PHD) o llevables (*wearables*) marcados por el gran auge de las nuevas tecnologías *wireless*.

Uno de los grandes retos en este contexto tecnológico es lograr interoperabilidad entre estos dispositivos médicos. Tradicionalmente el papel del ingeniero biomédico se basaba en una labor técnica que, a menudo, quedaba reducida al mantenimiento de los equipos ante eventuales fallos o su renovación y actualización. Cada fabricante realizaba un equipo propio y el hospital que utilizaba dicho dispositivo quedaba sujeto a sus condiciones, lo que implicaba grandes barreras y limitaciones para avanzar en soluciones de e-Salud homogéneas y globales. Ante esta situación, la interoperabilidad entre equipos de distintos fabricantes a través de la estandarización de protocolos, se convierte en un requisito básico para avanzar hacia la telemedicina ubicua: u-Salud [1].

Este es un largo proceso impulsado por el Comité Europeo de Estandarización (CEN/TC251) [2] y por la Asociación Española de Normalización (AENOR/CTN139) [3] que colaboran en el desarrollo del estándar ISO/IEEE11073 (X73): en su primera versión para interoperabilidad de dispositivos médicos en el punto de cuidado (*Point-Of-Care*, X73-PoC) [4], y recientemente reorientado para dispositivos médicos llevables (*wearables*) en entornos personales (*Personal Health Devices*, X73-PHD) [5].

Existen contribuciones previas de implementación [6]-[7], desarrolladas en EE.UU. por el grupo de investigación del Dr. Warren, que estudian la viabilidad de implantar estándares en entornos sanitarios e implementan plataformas similares de monitorización de pacientes en el PoC. Sin embargo, hasta el trabajo del PHD Working Group (WG) [8] no encontramos antecedentes europeos en este campo, ni tampoco propuestas de soluciones globales extremo a extremo que introduzcan nuevos casos de uso de monitorización de pacientes en entornos ubicuos y que estén orientadas a ser compatibles con la nueva versión de la norma X73-PHD, como se plantea en este artículo.

Así, y a partir de desarrollos anteriormente publicados [9]-[10], en los que se presentaba una propuesta de diseño basada en estándares extremo a extremo, en este artículo se avanza en la implementación de dicha plataforma. Su diseño y la arquitectura implementada ha de permitir la integración tanto de X73-PoC como de X73-PHD. Además, se debe adaptar a las futuras actualizaciones de la norma, por lo que el nuevo diseño propuesto ha de contemplar las futuras necesidades y permitir a la par la investigación, la experimentación de los más recientes estándares de tecnología médica, y su demostración en un entorno de pruebas integrado.

En la Sección II se presenta la arquitectura básica de la plataforma diseñada. En la Sección III se describe la implementación de la solución propuesta, detallando sus características técnicas y los aspectos claves que se han desarrollado para cumplir el estándar X73. La Sección IV valora y discute los resultados de esta implementación orientada a X73-PHD y analiza la incorporación de las nuevas funcionalidades en lo que constituirá la versión final de este trabajo de cara a convertirse en una solución transferible al sistema de salud. Las conclusiones globales del trabajo se discuten en la Sección V.

## 2. Arquitectura de la plataforma

La arquitectura básica de la plataforma ubicua de monitorización (ver Fig. 1) se basa en un elemento concentrador (*Compute Engine*, CE) que recopila toda la información adquirida por los diferentes dispositivos médicos (*Medical Devices*, MDs) de seguimiento del paciente que conforman las redes BAN/PAN. Este CE se comunica, a través de las redes de comunicaciones, con el servidor remoto del hospital que gestiona los diferentes CEs y centraliza la información proveniente de cada escenario de monitorización de paciente actualizando la Historia Clínica Electrónica (HCE).

Esta arquitectura cubre diversos requisitos fundamentales que pueden resumirse en las siguientes reglas de diseño:

- Primero, buscar un diseño funcional y conforme a X73. Para ello, se ha de eliminar la dependencia con la tecnología de transporte buscando una solución genérica y configurable (denominada gestor de capa de transporte o *handler*). Así, los datos adquiridos desde los MDs, primero se transforman a X73 (actualizándose cada vez que hay una nueva medida) para, después, iniciar el envío de datos (ya conforme a X73) del adaptador X73 al CE a petición del usuario. Por tanto, se deja al desarrollador la inclusión de los archivos que den soporte a la tecnología de transporte que estime oportuno para cada MD.
- Segundo, el diseño debe seguir el modelo de comunicaciones basado en la pila de protocolos genérica X73. Para este diseño se ha empleado un conjunto de archivos, agrupados en librerías, escritos en lenguajes C/C++ y Java, o generados a través de compiladores ASN1.C que recogen las reglas de codificación básicas y de paquetes (*Basic/Packet Encoding Rules*, BER/PER). Además, en el diseño se han de incluir las reglas específicas que contempla X73 para los dispositivos médicos (*Medical Devices* ER, MDER). Para ello se ha empleado un código optimizado de ASN1.C, denominado ASNX en X73, que enlace correctamente el vínculo desde el entorno de desarrollo para todos los recursos y con las clases abstractas que implementan la máquina de estados finita (*Finite State Machine*, FSM) definida por CEN para X73-PHD.
- Por último, al no incorporar tecnología de transporte concreta, los datos que se encapsulan a través de las distintas capas llegan a una disposición en estructuras de *buffers*, que recogen el conjunto de *Protocol Data Units* (PDUs) de las distintas capas de la pila. Estos *buffers* han de ser correctamente gestionados para que responda al protocolo de comunicaciones que marca la norma.

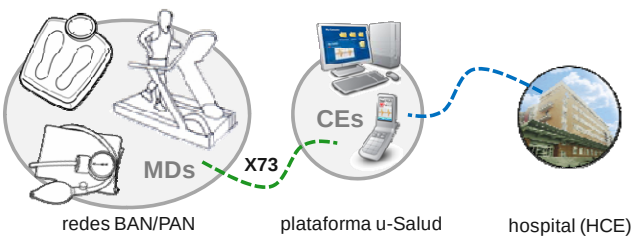


Fig. 1. Arquitectura básica de la plataforma de u-Salud

## 3. Implementación de la solución

A partir de la arquitectura de la plataforma propuesta y las reglas de diseño establecidas, se desglosa a continuación su implementación técnica y funcional.

### 3.1. Gestor de capa de transporte (*handler*)

Ante la necesidad de X73-PHD de una capa genérica de transporte, la comunicación entre ambos extremos a dicho nivel es responsabilidad del desarrollador. Para ello, se incluye una capa genérica TRANSPORTS (ver Fig. 2). La comunicación con la capa de sesión se realiza a través del interfaz genérico de pila (*stack*) para cada extremo (MD y CE), y con la capa física a través de un sistema de *buffers*, como se detalla adelante. Propuestas previas establecían un enlace en tiempo de ejecución (externo al programa principal *main*, tanto de MD como de CE) para llamar al archivo de transporte sobre IrDA o TCP/IP. Al eliminar la dependencia, *main* queda esperando peticiones de conexión, independientemente del protocolo de transporte asociado. Esta función se implementa mediante un interfaz genérico gestor de capa de transporte (*handler*), transparente al protocolo implementado.

A nivel de implementación, este *handler* es autosuficiente puesto que solo hace de enlace entre la capa de sesión y la capa física a través del sistema de *buffers*. En caso de querer introducir un protocolo de transporte determinado, se debería recompilar el *software*, y enlazarlo con la aplicación a través de funciones específicas (*transport_[nameProt]_l.cc* y *.h*). Además, habría que enlazarlo con el interfaz de pila, a través de un archivo tipo *transport_if.h*. Este *handler* es un objeto C++ que implementa los métodos más importantes de una interfaz de transporte y, además se le han incorporado otros adicionales que dan solución a ciertos requerimientos:

- Averiguar el perfil de comunicación que permite X73: episódico (*polling*) o periódico (*baseline*).
- Determinar si el protocolo de transporte que se implemente está soportado o no por el *manager* de la comunicación (CE).
- Manejar las estructuras recibidas por parte del interfaz del sistema de *buffers* y posterior envío al *handler*.

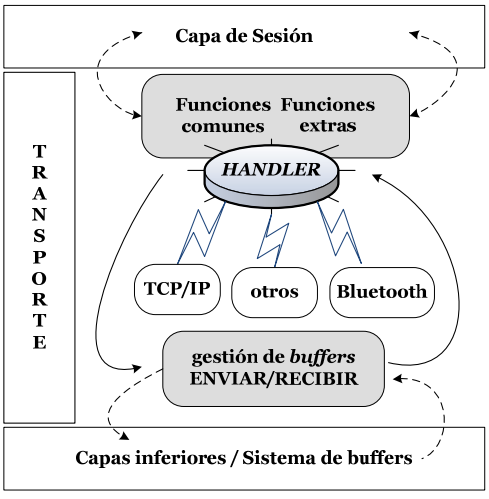


Fig. 2. Detalle del gestor de capa de transporte en la pila X73-PHD

### 3.2. Maquina de estados finita (FSM)

La plataforma se caracteriza por ser conforme a X73, siguiendo con meridiano rigor el diseño de las nuevas máquinas de estados (FSM) de comunicación para los extremos del sistema: *agente* (MD) y *manager* (CE). La filosofía de diseño de esas FSM (ver Fig. 3) ha sido clave en el diseño junto con la implementación de los estados por los que atraviesa la comunicación MD/CE. Para diseñar estas máquinas FSM se ha creado en el programa principal un bucle continuo que implementa los estados indicados en X73: DISCONNECTED, CONNECTED, UNASSOCIATED, ASSOCIATED, CONFIGURING y OPERATING. El proceso de funcionamiento sería como sigue:

- Los extremos están inicialmente apagados, por lo tanto antes de establecer una conexión, deben encenderse.
- A partir de esta inicialización y según cada FSM, se intentará establecer una conexión a través de la capa de transporte; si tiene éxito, hará que ambos entren al estado CONNECTED, pero estén UNASSOCIATED. Para asociarse, MD pide una petición de asociación a CE.
- Si CE sabe la configuración de MD, directamente quedan ASSOCIATED y podrán operar. Si no, CE acepta asociarse con MD pero necesitará configurarse (CONFIGURING) y guardar los parámetros específicos en una base de datos para evitar hacerlo más veces (esto facilita las funcionalidades Plug-and-Play).
- En el estado OPERATING, se realiza el proceso normal de toma de datos: codificación a X73, y envío de MD a CE.
- En cualquier momento del estado ASSOCIATED se puede querer desasociar, voluntariamente o no cualquier par (MD y CE). Voluntariamente pasan a UNASSOCIATED e involuntariamente se envía un mensaje de abortar.
- Así mismo durante el período de tiempo que dura la conexión entre ambos a nivel de transporte, pueden desconectarse, también voluntariamente por el final de la comunicación o involuntariamente por algún fallo.

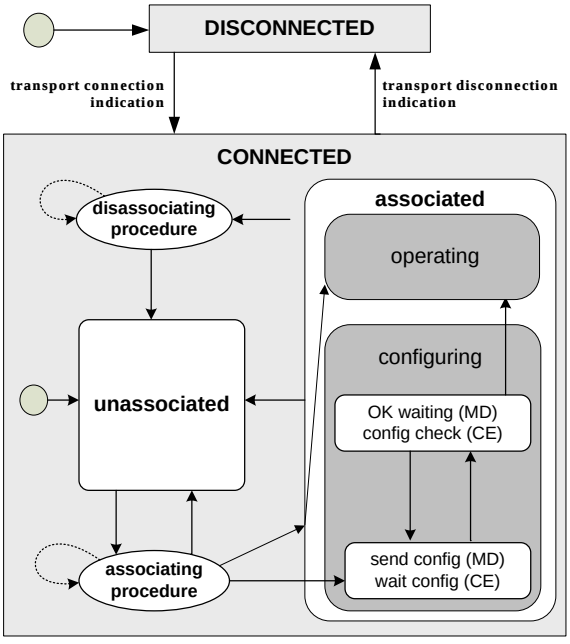


Fig. 3. Máquina de estados FSM genérica de MD y CE

### 3.3. Modelo de comunicación MD-CE (*buffers*)

De la misma manera que en la capa de transporte, se ha dejado la libertad de implementar cualquier capa física. Esta capa física se encarga de recibir y enviar las estructuras *st_buffer* que se envían MD y CE a través del *handler*. La estructura *st_buffer* es un contenedor de datos en memoria. Los bits que se codifican en cada PDU de cada capa siguen esta estructura de *st_buffer*, así son más fáciles de manejar. A su vez, un conjunto de *st_buffer* se puede agrupar en *st_packet*. Como se muestra en Fig.4, el modelo de comunicación MD-CE agente-manager conforme a X73 y a través del sistema de *buffers* sigue el siguiente proceso:

- Desde *main* se invoca un método que genere un evento en la capa de aplicación de MD (1). Este evento consiste en un mensaje para CE que se va encapsulando capa tras capa hasta llegar al nivel de sesión donde forma un *st_packet*. Este *st_packet* se encapsula en otra PDU que la capa del *handler* utiliza en forma de *st_buffer* (2). Este *st_buffer* contiene el mensaje que se ha ido encapsulando y que MD envía a CE.
- Se encapsula *st_buffer* de MD en un *buffer* que es transmitido a través de la conexión X73 hasta CE (3). Ese *buffer*, una vez recibido por CE, atraviesa las capas de la arquitectura del CE *down-up*, invocando la capa *handler* de CE, y se desencapsula hasta llegar a la capa de aplicación que lo lee y genera una respuesta (4).
- Esa respuesta de CE a MD recorre la pila desde la capa de aplicación a *handler* y genera otro *st_buffer* de respuesta que será enviado hacia MD (5).
- El mensaje de respuesta llega a MD (6), pasa por *handler*, sube hacia capa de aplicación y genera otro mensaje (evento o respuesta), permitiendo comenzar el proceso de comunicación de manera reiterativa (7).

Una clave de esta implementación es que se pueden controlar los bits *on the wire* (es decir, los bits que forman el *st_buffer* y que se pasan ambas pilas a través de su interfaz). De esta manera se permite realizar control de flujo y control de errores desde el programa principal, objetivo perseguido desde el CEN y que posibilita la gestión centralizada de datos y alarmas.

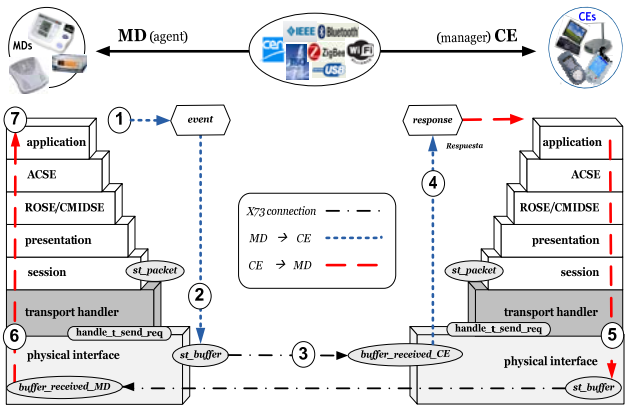


Fig. 4. Modelo de comunicación entre MD y CE conforme a X73

## 4. Resultados. Demostrador X73-PHD

La plataforma implementada es conforme a X73 y constituye un útil y eficaz demostrador X73-PHD, como muestra Fig. 5. Este demostrador comienza preguntando al usuario qué dispositivo MD desea usar de una lista disponible. Tras la elección del MD (actualmente está disponible el tensiómetro y en breve se implementará pulsioxímetro, báscula, y adquisidor de ECG), se muestra su información específica, y se informa de las mediciones disponibles (en este caso, presión arterial y pulso).

A continuación se muestra el menú de control de FSM basado en X73-PHD que atraviesa los extremos de comunicación (MD-CE). El demostrador permite desplazarse a cualquiera de los estados según las relaciones establecidas en la FSM (ver Fig. 4).

A partir de aquí, MD se inicializa, se crean las capas de las pilas y los interfaces de funcionamiento (*stacks*), y se genera toda la estructura de dispositivo estandarizada por X73. También se pregunta por el sistema de transporte que soporta la comunicación, preparando el *handler* para soportar los correspondientes protocolos. Además, en pantalla se muestran líneas de ejecución del programa que ayudan a conocer los métodos de cada capa atravesada. También se enseña cómo los *buffers* mandan los datos X73-PHD y parámetros de configuración de los eventos y respuestas intercambiados entre MD y CE.

Tras asociarse, en CE se crea un contexto de recepción de datos. Así, MD queda listo para la toma de medidas (siempre a petición del usuario) entrando en el estado OPERATING de la FSM. Al usuario se le pregunta si desea tomarse alguna medida. Si responde que sí, se informa de los valores que deberían resultar de la medición. Así el sistema está listo para adquirir. En este caso, pulsando el correspondiente botón en el tensiómetro, el dispositivo tomará automáticamente la tensión, sonando al acabar un pitido informativo de que hay datos disponibles porque las medidas han sido leídas. El proceso finaliza tras asegurarse de que los datos leídos son correctos y el envío a CE se constata, permitiendo el inicio del proceso.

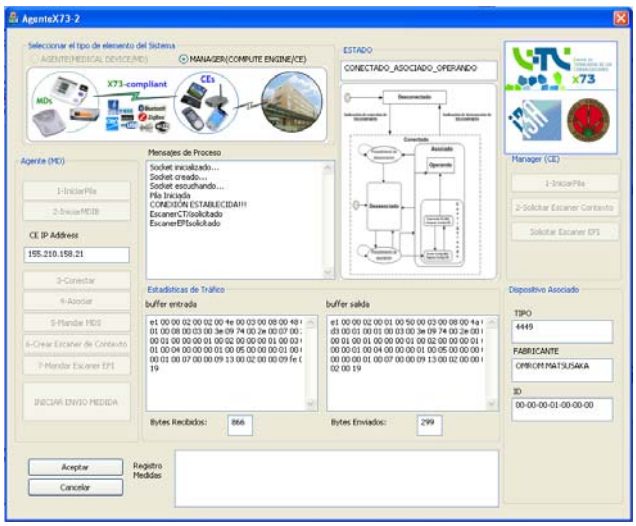


Fig. 5. Interfaz gráfico implementado para el demostrador X73-PHD

## 5. Conclusiones

En este trabajo se ha presentado la implementación de una plataforma de monitorización conforme al estándar X73 que es robusta, flexible, y cuyo diseño permitirá, con la inmediata incorporación de nuevas funcionalidades, completar una solución *plug-and-play* basada en estándares, y transferible al sistema de salud para la monitorización personal de pacientes en entornos ubicuos.

Además, su diseñado modular (que permite la incorporación de cualquier tecnología a nivel de transporte y físico) y completamente conforme al estándar la convierten en un útil y eficaz demostrador X73-PHD como entorno de pruebas para los retos que se debaten dentro del CEN: control de flujo y errores, gestión de errores y alarmas, conexión de múltiples MDs a uno o varios CEs, o implantación en micro-controladores, dispositivos *wearables* o equipos *wireless*.

## Agradecimientos

Este trabajo ha sido parcialmente subvencionado por los proyectos TSI2007-65219-C02-01 y TSI2005-07068-C02-01 de la Comisión Interministerial de Ciencia y Tecnología (CICYT) y Fondos Europeos para el Desarrollo Regional (FEDER), PET2006-0579 del Programa de Estímulo de Transferencia de Resultados de Investigación (PETRI), una beca FPI a M. Martínez-Espronceda (res. 1342/2006 de la Universidad Pública de Navarra), y una beca a Jesús Trigo (ref. IT7/08) de Diputación General de Aragón (DGA), Consejo Asesor de Investigación y Desarrollo (CONAID) y Caja de Ahorros de la Inmaculada (CAI).

## Referencias

[1] J. L. Monteagudo, O. Moreno, “e-Health for patient empowerment in Europe,” *Instituto de Salud Carlos III, Ministerio de Sanidad y Consumo*, Spain, 2007.

[2] Comiteé European de Normalisation (CEN). http://www.cen.eu. Technical Committee 251 – Health informatics (CEN/TC251). http://www.centc251.org.

[3] Asociación Española de Normalización (AENOR/CTN139) http://www.aenor.es/desarrollo/inicio/home/home.asp

[4] ISO/IEEE11073 Point-of-Care Medical Device Communication standard (X73-PoC). Health informatics. [Part 1. Medical Device Data Language (MDDL)] [Part 2. Medical Device Application Profiles (MDAP)] [Part 3. Transport and Physical Layers]. http://www.ieee1073.org. See also the previous standards: IEEE13734-VITAL and ENV13735-INTERMED of CEN/TC251, http://www.medicaltech.orgh.

[5] ISO/IEEE11073 - Personal Health Devices standard (X73-PHD). Health informatics. [P11073-00103. Technical report - Overview] [P11073-104xx. Device specializations] [P11073-20601. Application profile - Optimized exchange protocol]. IEEE Standards Association webpage: http://standards.ieee.org/.

[6] J. Yao and S. Warren, "Applying ISO/IEEE 11073 standards to wearable home health monitoring systems," *Journal of Clinical Monitoring and Computing*, vol.19, pp.427-36, 2005.

[7] S. Warren, J. Lebak and J. Yao, "Lessons learned from applying interoperability and information exchange standards to a wearable point-of-care system," *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, 2006, pp. 101-104.

[8] M. Clarke, D. Bogia, K. Hassing, L. Steubesand, T. Chan and D. Ayyagari, "Developing a standard for personal health devices based on 11073," *Int Conf IEEE Eng in Medicine and Biology Society (EMBS)*, 2007, pp. 6174-6176.

[9] M. Galarraga *et al.*. "Standards for Medical Device Communication: X73 PoC-MDC", *Medical & Care Compunetics 3. IOS Press – Studies in Health Technology and Informatics (ISSN: 978-1-58603-630-0)*, vol. 121, pp. 242-256, 2006.

[10] I. Martinez, *et al.*, "Implementation of an end-to-end standard-based patient monitoring solution," *IET Commun.*, vol. 2, issue 2, pp. 181-191, 2008.