

# Implementación de una plataforma embebida para la interoperabilidad de dispositivos médicos

CARBALLO ARELY<sup>1</sup> AND ACEVES PÉREZ RITA MARINA<sup>1</sup>

<sup>1</sup>Laboratorio de Investigación y Posgrado, Universidad Politécnica de Puebla, 3er Carril del Ejido Serrano S/N, San Mateo Cuanalá, Juan C. Bonilla, Puebla, CP 72640.

Compiled 25 de septiembre de 2017

*Este artículo presenta los resultados de la implementación y pruebas de una plataforma de código abierto que permite la interoperabilidad y monitoreo de varios dispositivos médicos (OpenICE). En la actualidad, en los hospitales y en particular en las salas de terapia intensiva se presenta el problema de que para monitorear a un paciente se tienen muchos dispositivos médicos, cada uno con su interfaz y diferentes filosofías de operación, lo que dificulta la detección temprana de alarmas y que puede representar severos daños al paciente. En este trabajo, se realizaron pruebas para comprobar la funcionalidad de OpenICE. Con la simulación se lograron resultados alentadores, comprobar la facilidad para la implementación, así como constatar que cumple con los requisitos de interoperabilidad médica.*

**Palabras clave:** OpenICE, BeagleBone Black, dispositivos médicos © 2017 Universidad Politécnica de Puebla

**OCIS codes:** (140.3490) Lasers, distributed feedback; (060.2420) Fibers, polarization-maintaining; (060.3735) Fiber Bragg gratings.

<http://dx.doi.org/10.1364/ao.XX.XXXXXX>

## 1. INTRODUCCIÓN

Actualmente, una unidad de terapia intensiva neonatal es un ambiente altamente complejo donde se mezclan diversas tecnologías médicas tales como monitores de paciente, bombas de infusión, bombas peristálticas, ventiladores pulmonares, sistemas de succión, cuna térmica, etc. La mayoría de estas áreas contienen tecnología de diferentes fabricantes por lo que es un verdadero problema reunir información de todos los dispositivos en un solo formato o verificar cual dispositivo es el que emite la alarma, por lo que se sigue empleando el método tradicional de recolección de datos. Debido a la complejidad de una unidad de cuidados intensivos surgió la necesidad de hacer los dispositivos interoperables que se quiere decir con eso, que los dispositivos puedan ser interconectados entre sí y que puedan compartir información bajo un mismo estándar. Observando esta particularidad surgen algunas propuestas [1], [2], [3] y [4]

de las cuales destaca una llamada OpenICE, la cual atiende este concepto de interoperabilidad.

OpenICE [4] es una plataforma de investigación clínica abierta para sistemas distribuidos que permite la colaboración de personas interesadas en contribuir a la aplicación actual. Esta plataforma fue propuesta por el CIMIT a partir del programa de interoperabilidad MD PnP establecido en el 2004 [5]. Esta aplicación es implementada con base en el estándar ASTM F2761-09 (ICE- Ambiente clínico integrado) [6] el cual muestra la arquitectura clave para una implementación ICE, ver Figura 1. Además de este estándar OpenICE se basa en una pila de protocolos tales como la familia IEEE-11073 [7] de la cual se deriva la IEEE-11073-10101 donde obtienen toda la terminología utilizada para las variables sensadas por los dispositivos médicos, la IEEE-11073-10201 en la cual se establece el modelo del dominio de información y la IEEE-11073-20601 que define la comunicación del dispositivo de salud personal, por último para la distribución de datos utiliza el estándar DDS [8] el cual le proporciona un paradigma de comunicación publicador/suscriptor.

OpenICE integra los dispositivos médicos y las aplicaciones clínicas de un ecosistema de salud existente [9]; esto significa que es el punto de entrada y salida de los datos de los nodos dentro y fuera de una red, por nodos se entiende:

- Los Dispositivos médicos que se encuentran en un quirófano o la UCI.
- Las Aplicaciones de grabación de la caja negra (es decir, de registro de datos)
- Las Solicitudes de apoyo a la decisión clínica (CDS)
- Y las interfaces externas a otro sistema de tecnologías de la información de la salud como diagramas de flujo de EMR.

OpenICE tiene dos modos de funcionamiento [10]: El modo supervisor, obtiene los datos y muestra los gráficos a partir de los dispositivos conectados por medio del adaptador, esta puede ser ejecutada en una PC y mediante unos campos se puede suscribir a todos los dispositivos médicos en el mismo dominio y contexto del paciente; el segundo modo de funcionamiento es el adapter-device (Adaptador del dispositivo) el cual actúa como un puente entre los dispositivos médicos y OpenICE, traduciendo los propios protocolos de comunicación del dispositivo en las estructuras de datos estándar de OpenICE y el protocolo de comunicación. Este puede ser instalado en sistemas embebidos, laptops, Raspberry Pis, Intel Galileo y Beagleboards.

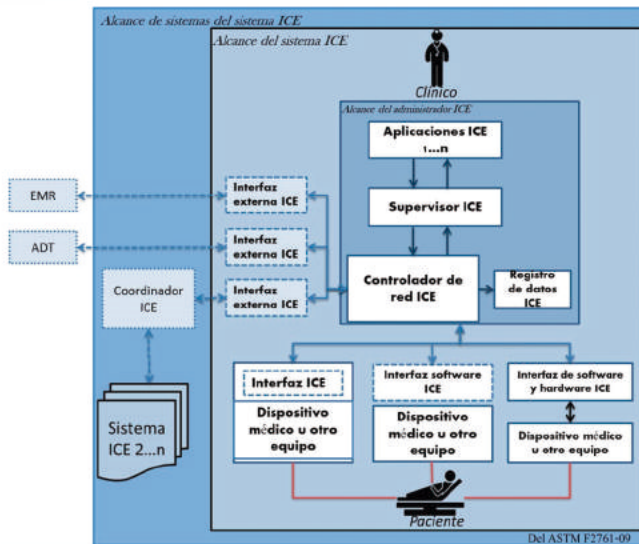


Fig. 1. Arquitectura del estándar ASTM F2761-09

## 2. METODOLOGÍA PARA LA IMPLEMENTACIÓN

Para poder realizar la implementación de OpenICE se necesita primeramente una tarjeta embebida, en este caso se utilizó una BeagleBone Black y descargar la aplicación de OpenICE, también son necesarias algunas configuraciones previas que se mencionan a continuación.

### A. Configuraciones en BeagleBone Black

Para poder realizar la instalación en la tarjeta embebida fue necesario:

- El sistema operativo DEBIAN,
- El software de Oracle SE JDK,
- El ejecutable de OpenICE y
- NTP Daemon que es el software del protocolo de tiempo que es requerido para sincronizar todos los nodos del sistema.

En la página de OpenICE [10] se puede descargar la imagen de todos los elementos anteriores. Después de descargar la imagen de disco se copió a una tarjeta MicroSD, esta tarjeta se debe configurar para que cuando se inserte en la BeagleBone sea reconocida automáticamente. Para hacer esta configuración se debe tener en una PC cualquier versión del sistema operativo Linux, se introduce la MicroSD y a continuación en la ventana de comandos se digita el siguiente comando: `xz -dc bone-debian-7.8-lxde-4gb-armhf-2015-03-01-4gb-openice-0.6.2.img.xz | dd of=/dev/sdX bs=1024K`

Enseguida se procedió a "flashear" la memoria de la BeagleBone, esto significa que obtendrá los archivos de la memoria MicroSD y los instalará en la memoria interna que contiene la BeagleBone, para esto se realizaron los siguientes pasos:

- Insertar la MicroSD en la ranura de la BeagleBone.
- Conectar la BeagleBone a un suministro de voltaje a no más de 5v o bien se conecta por medio del cable USB a una computadora para que suministre energía eléctrica.

- Al conectar la BeagleBone, automáticamente copiará la imagen de la MicroSD y los cuatro leds se iluminarán de extremo a extremo.
- Cuando el flasheo de la memoria haya terminado los leds quedarán iluminados y después de un minuto se apagarán al igual que la BeagleBone, seguido de que suceda esto se deberá retirar la MicroSD.
- Ahora la BeagleBone tiene instalados los cuatro software que se mencionaron anteriormente.

Para poder acceder a la tarjeta BeagleBone desde una computadora se necesitó de un emulador de puerto serial se eligió Putty por su fácil manejo y la característica de que su descarga es gratuita. Al conectar la BeagleBone a la computadora se observó que no era detectada por lo que se debieron instalar en la tarjeta los drivers correspondientes al sistema operativo que se está utilizando y que pueden ser descargados de la página oficial de la BeagleBone [11]. Después de copiar los drivers a la tarjeta, se pudo acceder a ella por medio de Putty. Al abrir la terminal Putty se ingresó la dirección IP de la BeagleBone Black la cual es 192.168.13.2 para acceder por SSH y al entrar solicita un usuario y contraseña por lo que para la imagen que se le instaló corresponden a "debian" para ambos casos. Al acceder a la tarjeta se tuvo que actualizar los repositorios y paquetes existentes por lo que se le conectó un cable de red a la tarjeta para que pudiera acceder a internet y en la terminal se digitó el siguiente comando: `Sudo apt-get update` Y después de varios minutos se actualizaron los paquetes, posteriormente se verificó la versión de OpenICE para ver que se hubiera actualizado pero se encontró que no, por lo tanto se tuvieron que hacer los pasos siguientes:

- Acceder a la BeagleBone por escritorio remoto con la tecla Windows + r y escribir mstsc para entrar al escritorio remoto.
- Se ingresa la dirección IP de la tarjeta, el usuario y la contraseña.
- Al estar en el escritorio de la tarjeta se accede a un buscador que tenga instalado y se descarga la versión más reciente de OpenICE como aparece en un archivo comprimido se debe acceder a la raíz para obtener otros permisos requeridos como lo es instalar otro programa, esto es necesario para poder descomprimir el archivo descargado.
- Salir del escritorio remoto y acceder nuevamente por Putty pero ahora se inició sesión desde la raíz entrando con nombre de usuario "root", para esto no pide contraseña y posteriormente se instala winzip con el siguiente comando: `Apt-get install zip`

Y para descomprimir el archivo primero se debe dirigir al escritorio que es donde se desea descomprimir el archivo y se utilizan los siguientes comandos:

```
cd Desktop
unzip OpenICE-0.6.3
```

Posteriormente se verificó que la versión actual estuviera en el escritorio de la BeagleBone digitando ls como se observa en la Figura 2.

Para conocer los parámetros que maneja la aplicación se accede con la siguiente línea:

```
OpenICE-0.6.3/bin/OpenICE -help
```

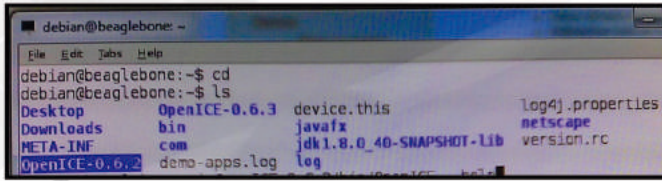


Fig. 2. Archivos ubicados en el escritorio de la BeagleBone Black.

Y muestra dos opciones de funcionamiento una para configurar dispositivos médicos y otra para simular dispositivos, así como también muestra los nombres de los dispositivos que pueden ser simulados y los reales que son compatibles con la aplicación, como se observa en la Figura 3.

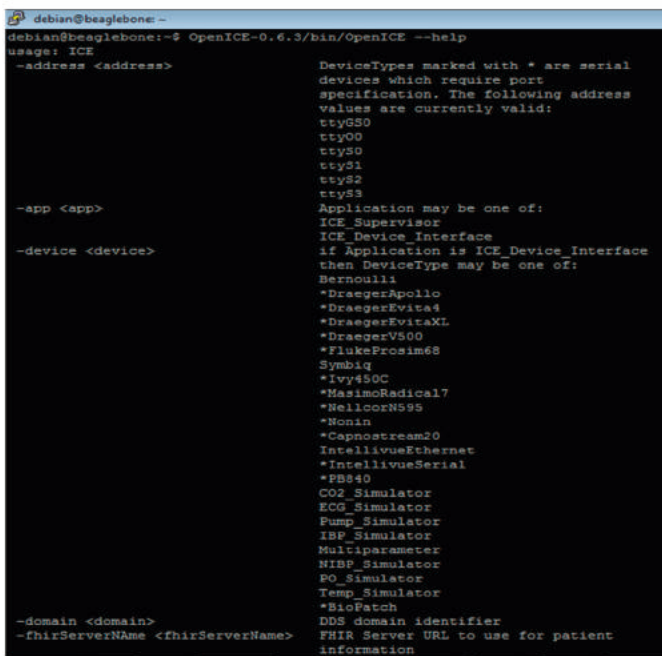


Fig. 3. Funciones del adaptador de OpenICE

### B. Accediendo a OpenICE en la computadora

Para obtener la aplicación se descargó la versión correspondiente a Windows de [10]. Seguido se instaló la aplicación haciendo doble clic sobre el ejecutable y se siguieron los pasos, hasta finalizar la instalación. Al abrir la aplicación apareció una pantalla como la que se muestra en la Figura 4 donde se tienen tres campos:

Application; en este apartado se pueden seleccionar el modo de software que se quiere ejecutar hay dos opciones: el ICE Supervisor que corresponde a todas las aplicaciones clínicas y el ICE Device Interface que corresponde a la aplicación de adaptador de dispositivo y que permite conectar un dispositivo médico a la red de OpenICE.

Domain ID; Se asigna un valor para el dominio que requiere la red, es necesario para identificar en cual red se quiere conectar el dispositivo o donde se requiere extraer los datos, esto es por si existen múltiples sistemas ICE con muchos dominios.

FHIR; este apartado es opcional, aquí se asigna el URL de un HL7 FHIR (recursos para interoperabilidad del cuidado de salud rápida) es un servidor donde la información del paciente

debiera ser recibida. Este campo puede dejarse en blanco para usar una base de datos local del paciente.

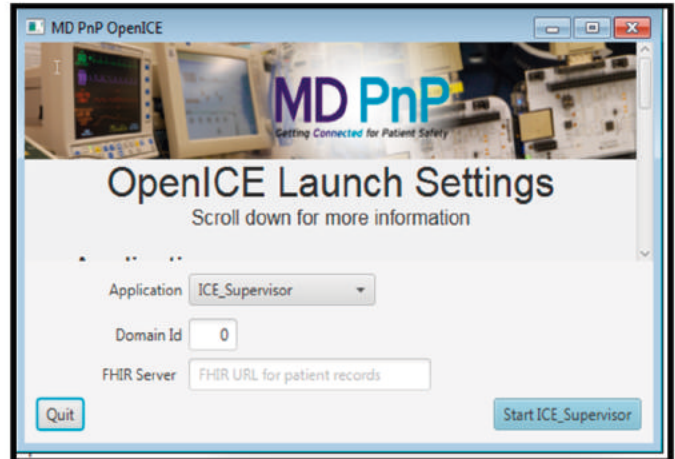


Fig. 4. Pantalla de inicio de OpenICE

La implementación realizada es mostrada por el diagrama mostrado en la Figura 5, donde se tienen los dispositivos simulados a partir de la tarjeta BeagleBone Black y que a su vez esta manda los datos a la aplicación supervisor donde se verán los graficos correspondientes al dispositivo que se esté simulando.

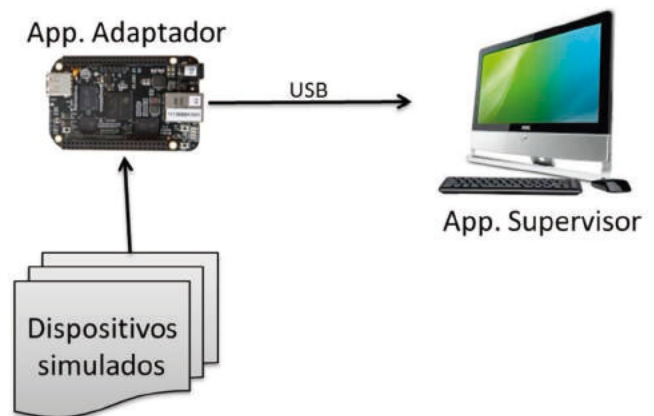


Figura 5. Diagrama de la implementación

Fig. 5. Diagrama de la implementación

### 3. RESULTADOS Y DISCUSIÓN

En la Figura 6 se muestra la prueba realizada para comunicar los dispositivos simulados en una BeagleBone Black con la aplicación supervisor de ICE.

Para simular un dispositivo en la BeagleBone se llevan a cabo los siguientes pasos

- Se ingresa el siguiente comando en la terminal Putty para simular un electrocardiograma en la tarjeta BeagleBone Black. `echo domain 5 device ECG_simulator >device.this`
- Al dar un enter el dispositivo comenzará a simular, pero no se logra ver nada debido a la versión de java instalada ya que para versiones más recientes la BeagleBone no

puede mostrar gráficos. Para lograr verlos se necesita de la aplicación Supervisor.

Posteriormente se abre el supervisor ICE y se ingresa el mismo dominio que fue utilizado para simular el dispositivo, en este caso el dominio fue 5. En seguida se iniciará la aplicación, apareciendo el icono del dispositivo que se está simulando y observando que la dirección IP que aparece corresponde a la tarjeta BeagleBone (Figura 6); al abrir ese icono veremos en la pantalla los gráficos correspondientes a la simulación del electrocardiograma (Figura 7).

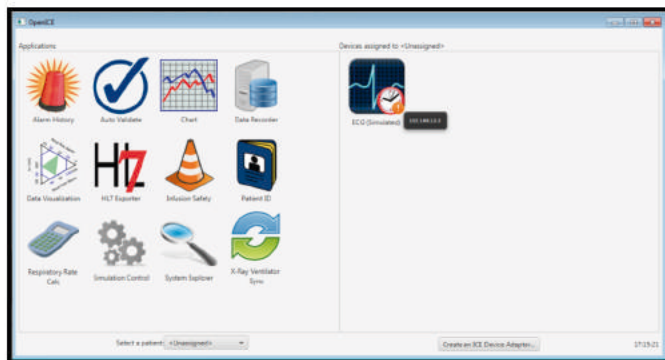


Fig. 6. Supervisando dispositivo simulado mediante la BeagleBone Black

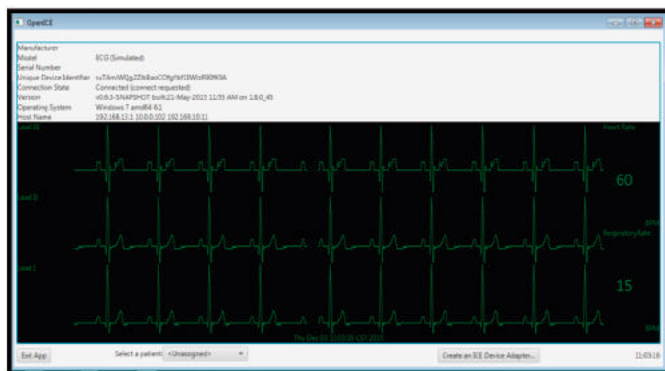


Fig. 7. Señal recibida mediante el supervisor

La siguiente prueba que se realizó con OpenICE fue simular dispositivos en una computadora por medio de la aplicación supervisor pero en la categoría *adaptador* y detectarlos en otra computadora corriendo la misma aplicación pero en función de supervisor.

Se conectó por medio de un router de DHCP en una red LAN y se obtuvo la comunicación obteniendo las señales correspondientes a cada simulación que se realizó. En la Figura 8, se pueden observar las computadoras, una está simulando un dispositivo y en la otra se está recibiendo información del mismo.

Dentro de lo que se pudo observar al estudiar esta plataforma es que no cuenta con una base de información que pueda ayudar a futuros investigadores o interesados en el tema a poder agregar un dispositivo más a la aplicación. No obstante esta plataforma continúa en proceso de desarrollo para lograr ser la base de una implementación completa de un ICE. Algunos de los imprevistos que se tuvo fue que si el adaptador y el supervisor



Fig. 8. Simulando y recibiendo señales de un dispositivo mediante diferentes PC's

no están sincronizados nunca se podrán observar los gráficos del dispositivo, sin embargo si se observan los datos estáticos como números. Al menos que se sincronice la hora en ambas partes antes de empezar el envío de datos se logrará ver estos gráficos.

#### 4. CONCLUSIÓN

En este artículo se presentó la implementación de OpenICE en una tarjeta de desarrollo y se observó el funcionamiento de esta plataforma, obteniendo que la implementación cumple con los principales requerimientos que presenta el protocolo ICE y verificando la conectividad plug and play que se requiere en las unidades médicas. Se pudo realizar la simulación de los dispositivos, los trabajos futuros apuntan a la verificación del funcionamiento con dispositivos conectados y también aumentar el número de dispositivos.

Dentro de lo que se pudo observar al estudiar esta plataforma es que no cuenta con una base de información que pueda ayudar a futuros investigadores o interesados en el tema a poder agregar un dispositivo más a la aplicación. No obstante esta plataforma continúa en proceso de desarrollo para lograr ser la base de una implementación completa de un ICE. Algunos de los imprevistos que se tuvo fue que si el adaptador y el supervisor no están sincronizados nunca se podrán observar los gráficos del dispositivo, sin embargo si se observan los datos estáticos como números. Al menos que se sincronice la hora en ambas partes antes de empezar el envío de datos se logrará ver estos gráficos.

#### 5. AGRADECIMIENTOS

Al CONACYT, Consejo Nacional de Ciencia y Tecnología por la beca de Maestría en Ingeniería de la Ing. Blanca Arely Carballo Almendra.

#### REFERENCIAS

1. "Ihe international, integrating the healthcare enterprise," Disponible en: <http://www.ihe.net/> (2015).
2. "HL7 international, hl7(health level seven international) standards,," Disponible en: <http://www.hl7.org/> (1987).
3. S. Schlichting and S. Pöhlsen, HL7, 2014. 11073/HL7 Standards Week (2014).
4. M. R. T. N. . C. T, "State of the art technology- protocols, networks and standards for point-of-care device communication," Disponible en: [http://www.iso.org/iso/wsc-medtech\\_23\\_Melvin\\_Reynolds.pdf](http://www.iso.org/iso/wsc-medtech_23_Melvin_Reynolds.pdf) (2004). [En línea]. Consultada en junio, 2015.

5. J. M. Goldman, "Medical device plug-and-play (md pnp) interoperability standardization program development," Tech. rep., MASSACHUSETTS GENERAL HOSPITAL BOSTON (2009).
6. J. M. Goldman, ASTM International (2008).
7. "Iso/ieee, iso/ieee11073-x, medical health device communication standards family," (2004).
8. OMG, "Extensible and dynamic topic types for dds," (2014).
9. J. Plourde, D. Arney, and J. M. Goldman, "Openice: An open, interoperable platform for medical cyber-physical systems," in "Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on," (IEEE, 2014), pp. 221–221.
10. CIMIT, "Openice," Disponible en: <https://www.openice.info/> (2014). [En línea]. Consultada en Agosto, 2015.
11. B. Foundation, "Beaglebone black," Disponible en: <http://beagleboard.org/BLACK> (2014). [En línea]. Consultado en Noviembre, 2015.