UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERIA

DIPLOMADO EN LINUX EMBEBIDO

Proyecto: Medición de pulso cardiaco

Autores:
Victor Hugo Garcia Ortega
Ricardo Balderas Paredes

Planteamiento del problema

Los signos vitales son parámetros clínicos que reflejan el estado fisiológico del organismo humano, y esencialmente proporcionan los datos (cifras) que darán las pautas para evaluar el estado homeostático del paciente, indicando su estado de salud presente, así como los cambios o su evolución, ya sea positiva o negativamente. Los signos vitales incluyen: Temperatura, frecuencia respiratoria, frecuencia cardíaca y presión arterial, entre otros.

El monitoreo de signos vitales y biomédicos cubre un amplio espectro de aplicaciones en diferentes contextos hoy en día, por lo que ha sido un tema concurrente de investigación en la última década. Actualmente, existen sistemas diseñados para monitorear signos vitales y son utilizados principalmente en clínicas y hospitales.

Actualmente, es común que el monitoreo de signos vitales sea realizado con un método en el que es necesario que el personal médico acuda físicamente a verificar el estado de salud de cada uno de los pacientes, haciendo imposible un monitoreo continuo de los signos vitales de todos ellos.

Los signos vitales deben ser verificados cada uno de diferente manera, por lo que el tiempo necesario para la revisión de cada paciente puede llegar a ser prolongado con el método actual; por lo tanto, no es posible garantizar que los signos vitales de los demás pacientes se mantengan en el rango de estabilidad, a menos que se tenga personal suficiente para esto.

La Organización Mundial de la Salud (OMS)[1] estima que menos de 23 trabajadores de la salud por cada 10 mil habitantes son insuficientes para alcanzar la cobertura de las necesidades de atención primaria de salud.

El Instituto Nacional de Estadística y Geografía (INEGI)[2] indicó que existen sólo 3 doctores por cada 2 mil habitantes en nuestro país y una enfermera (o) por médico.

Al no cumplir con la recomendación de la OMS respecto al personal médico para cubrir las necesidades de atención en el sector salud, se propone este sistema para incrementar la eficacia del personal médico en nuestro país y con ello la calidad de vida de los mexicanos en cuanto a salud se refiere.

Introducción

En este proyecto se describe la medición del signo vital para determinar el pulso cardiaco de un paciente de forma remota. El proyecto se desarrolló en el sistema embebido Raspberry Pi 3. La arquitectura del sistema se muestra en la ilustración 1.

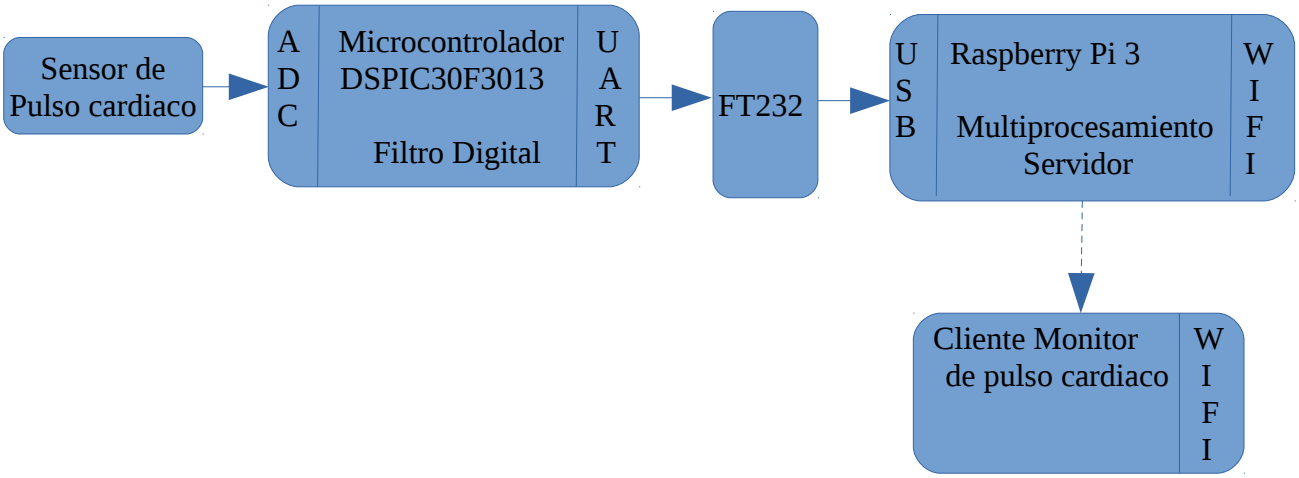

Ilustración 1: Arquitectura del Sistema Propuesto

La explicación de cada bloque se presenta a continuación:

1. El sensor de pulso cardiaco es un sensor de la compañía “pulse sensor” [], el cual trabaja con 3.3v y 5v, entrega la señal de pulso cardiaco con un rango dinámico igual al voltaje de alimentación.

2. La señal de pulso cardiaco tiene un ancho de banda de 120Hz, por lo que de acuerdo al teorema de muestreo debe ser digitalizada como mínimo a 240Hz. La norma médica indica que debe usarse una frecuencia de muestreo de 512Hz con una resolución mínima de 12 bits en el convertidor analógico digital (ADC – Analog to Digital Converter). Por esta razón se utiliza un controlador digital de señales (DSC) de 16 bits de la compañía Microchip, el DSPIC30F3013, el cual tiene un ADC de 12 bits que convierte la entrada usando la técnica de aproximaciones sucesivas y trabaja hasta a 200ksps (miles de muestras por segundo). La señal digitalizada de pulso cardiaco es filtrada digitalmente para eliminación de ruido usando la unidad DSP includa en el DSC. Posteriormente las muestras digitalizadas son enviadas usando la interfaz UART hacia la tarjeta Raspberry Pi 3. El UART se configura a una velcidad de 9600 baudios. Con esta velocidad se transmite 960 bytes por segundo, la cual es una velocidad adecuada puesto que nosotros estaremos enviando 512 bytes por segundo.

3. Como se conecta por USB a la tarjeta Raspberry Pi 3 se utiliza un FT232 de la compañía FTDI, el cual es convertidor UART-USB.

4. La tarjeta Raspberry Pi 3 recibe los datos digitales de pulso cardiaco y realiza el procesamiento de la señal. Este procesamiento consiste en dos algoritmos:

a) Ventaneo de la señal. Este algoritmo utiliza una ventana hamming para eliminar el efecto Gibbs que se produce al digitalizar una señal analógica.

Matemáticamente: $v[n] = a0 - a1 \cos(2*PI*n/(N-1))$
Donde: a0 = 0.53836, a1 = 0.46164

b) Calculo de la frecuencia de la señal. Este algoritmo consiste en el cálculo de la función de autocorrelación, la cual permite determinar la frecuencia fundamental en señales cuasi periódicas. La señal de pulso cardiaco es una señal de este tipo.

Matemáticamente: $rxx(l) = \sum_{n=l}^{N-1} x(n)x(n-l)$

Un punto interesante acerca de esta ecuación es que, computacionalmente hablando, la misma es completamente paralelizable, debido a que los resultados de la ecuación en cierto punto son independientes de los resultados en otros puntos; por lo que en un procesador multinúcleo se puede calcular de forma paralela y obtener el resultado de una forma más rápida. Con llamadas al sistema de linux, fork, se realiza la evaluación de forma paralela usando 4 procesos, puesto que Raspberry Pi 3 tiene 4 núcleos. La obtención de resultados de cada proceso se realiza mediante una tuberia, pipe, en el proceso padre. Para mandar el resultado final se plantea usar un servidor con sockets para mandar de forma remota el resultado al personal médico correspondiente.

5. Finalmente se usó un cliente TCP que permite leer el resultado de pulso cardiaco y mostrar el resultado.

Diagrama de procesos.

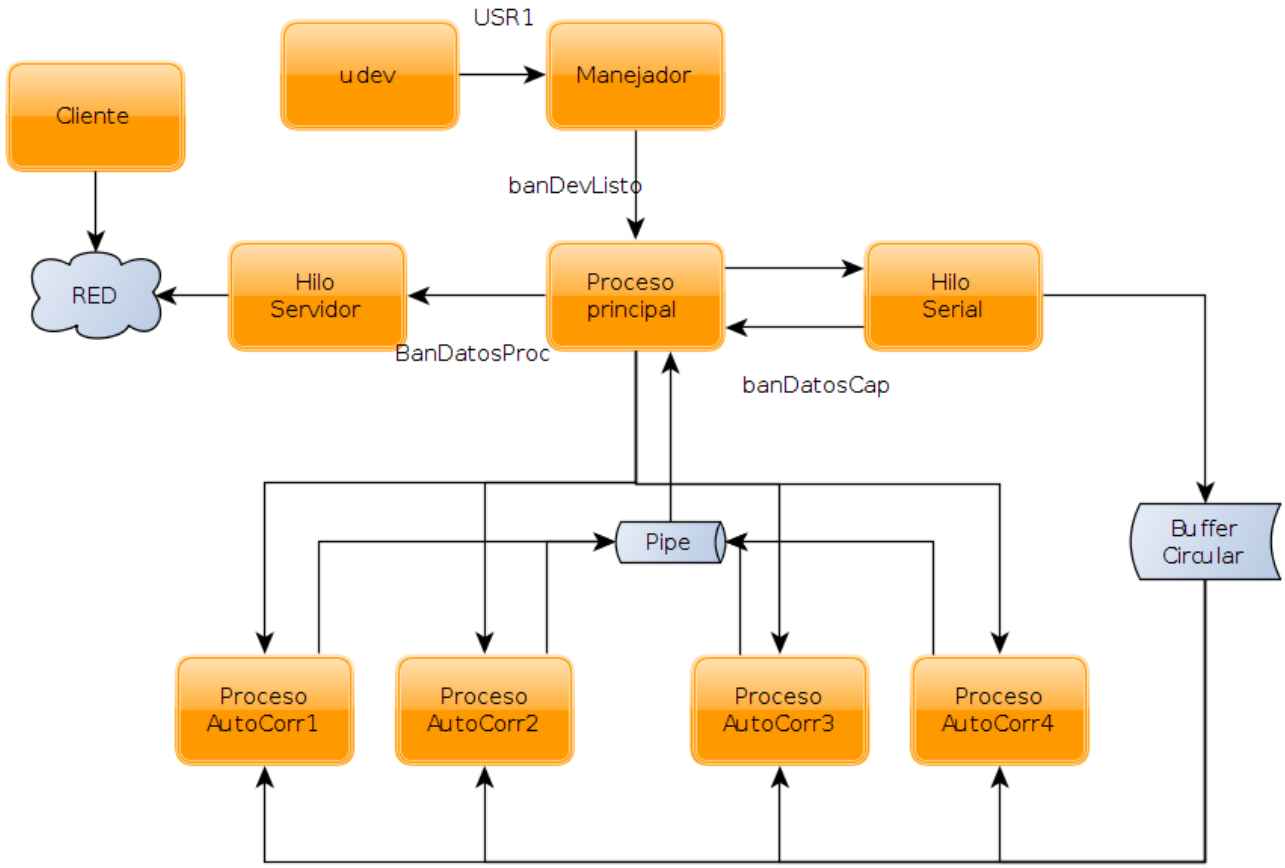

Ilustración 2: Diagrama de procesos

El sistema comienza esperando la señal USR1, la cual es enviada cuando se conecta el módulo FT232 al conector USB de RaspBerry, esta detección la realiza el gestor de dispositivos UDEV. Esto se realiza mediante una regla UDEV que ejecuta un script el cual manda la señal USR1 al demonio de la aplicación. Al recibir esta señal se ejecutan dos hilos: el hilo serial y el hilo servidor. El hilo serial inicializa el UART y comienza la adquisición de los datos en un buffer circular de 4kbytes. En este buffer se realiza un traslape cada 256 bytes, lo cual quiere decir que se toman 256 muestras de la señal de pulso cardiaco y entonces se activa la bandera banDatosCapturados, haciendo que el proceso principal ejecute 4 procesos para calcular la función de autocorrelación en bloques de 4kbytes.

Una vez que son ejecutados los procesos, cada uno calcula un elemento de la función de autocorrelación de forma alternada, es decir:

El proceso 1 calcula los elementos 0, 4, 8, 12,… de la función.
El proceso 2 calcula los elementos 1, 5, 9, 13,… de la función.
El proceso 3 calcula los elementos 2, 6, 10, 14,… de la función.
El proceso 4 calcula los elementos 3, 7, 11, 15,… de la función.

Al terminar la ejecución de cada proceso, estos mandan su resultado parcial mediante una tuberia, pipe, al proceso padre, quien determina el resultado final. Cuando esto ocurre, el proceso principal activa una bandera, banDatosProcesados, la cual permite que el hilo servidor mande el resultado final al cliente que se conectó al servidor.

Generación de la imagen mímina con el proyecto Yocto.

El proyecto Yocto es un proyecto de colaboración de código abierto que proporciona plantillas, herramientas y métodos para ayudar a crear sistemas personalizados basados en Linux para productos embebidos independientemente de la arquitectura de hardware.

Este proyecto esta compuesto por: Poky = BitBake + Metadata

Poky, es el sistema de construcción usado por Yocto.
BitBake, es el ejecutor y planificador de tareas.
Metadata, contiene la definición de las tareas, formado por:

- Archivos de configuración (. conf). Contiene la definición global de variables.
- Archivos de clases. (.bbclass). Contiene el encapsulamiento y herencia de la lógica de construcción, empaquetamiento, etc.
- Archivos de recipientes (.bb). Unidades lógicas de software.

La imagen generada con el proyecto yocto se denomina: core-image-base. Es una imagen de solo consola que tiene soporte completo para el hardware del dispositivo seleccionado.

Creación de los scripts de arranque

La imagen de Linux generada por el proyecto Yocto utiliza el sistema de arranque System V, por lo que fue necesario crear un script compatible con dicho sistema.

Dicho script contiene manejadores para lanzar y detener el demonio de la aplicación que calcula la frecuencia de pulso cardiaco. Además tiene un manejador de estado del demonio para poder consultar dicha información de una forma sencilla.

Reglas de manejo de dispositivo

Con el objetivo de que el servidor no esté trabajando cuando no hay un paciente tomando su pulso, es necesario incluir una forma de detectar que el FT232 que se comunica con el microcontrolador ha sido conectado o desconectado. Para poder hacer eso, es necesario registrar una regla de UDEV que de alguna forma le indique al demonio de la aplicación que se ha conectado el paciente. Ello lo hace lanzando uno de dos scripts dependiendo de la acción que se realizó (conectar o desconectar). Las reglas usadas son las siguientes.

ACTION=="add", SUBSYSTEM=="tty", ATTRS{idProduct}=="6001",
ATTRS{idVendor}=="0403", ATTRS{serial}=="A402JTR3", SYMLINK+="FTDI_ADQ",
RUN+="/usr/local/bin/proyStart.sh"

#ACTION=="remove", SUBSYSTEM=="tty", ATTRS{idProduct}=="6001", ATTRS{idVendor}=="0403", #ATTRS{serial}=="A402JTR3", RUN+="/usr/local/bin/proyStop.sh"

La primera regla se ejecuta al conectar el FTDI, y en la misma se le asigna la liga simbólica /dev/FTDI_ADQ al dispositivo que sea detectado con los atributos mencionados en la regla; y posteriormente manda a ejecutar el script proyStart.sh que le envía una señal USR1 al servidor, para que este ultimo comience a leer de la Interfaz Serie.

La segunda regla se ejecuta al desconectar el FTDI y ejecuta el script proyStop.sh, el cual envía una señal USR2 al servidor para que deje de leer de la Interfaz Serie.

Referencias

[1] «Organización Mundial de la Salud,» [En línea]. Available: http://www.who.int/es/. [Último acceso: 11 diciembre 2015].

[2] «Instituto Nacional de Estadística y Geografía,» [En línea]. Available: http://www.inegi.org.mx/. [Último acceso: 11 diciembre 2015].