Sistema básico de vigilancia sobre la plataforma BeagleBoard-xM

César Ávila Sánchez

Escuela de Ingeniería Electrónica

TEC

Cartago, Costa Rica

cesaravila22@gmail.com

Fabricio Quirós Corella

Escuela de Ingeniería Electrónica

TEC

Cartago, Costa Rica

fabisqc0207@gmail.com

***Abstract***—**El propósito de este documento consiste en describir de manera clara y detallada los resultados obtenidos, así como el proceso de desarrollo de un sistema básico de vigilancia, mediante la utilización de una cámara web y la implementación de un algoritmo de detección de movimiento en una plataforma empotrada.**

Palabras Claves—BeagleBoard-xM, Narcissus, OpenCV, GStreamer.

# INTRODUCCIÓN

Como es bien sabido, los sistemas embebidos, gracias a la enorme cantidad de recursos que disponen, pueden ser empleados para ejecutar una gran variedad de aplicaciones específicas, donde, en este caso, se consideró implementar un sistema básico de vigilancia, tomando en cuenta una BeagleBoard-xM, como unidad central de procesamiento, y elementos externos de interacción con el usuario, como lo son una cámara web, unos parlantes, un teclado y un monitor, posteriormente se detallará la funcionalidad de cada uno de estos componentes.

Es más que evidente que la BeagleBoard-xM se encuentra en la capacidad de efectuar tareas multimedia, tales como reproducción de audio, de video y detección de movimiento, las cuales son funciones fundamentales que es necesario que desempeñe el sistema de vigilancia pensado. No obstante, para cumplir con este objetivo, es necesario recurrir a una librería libre y multiplataforma, desarrollada por Intel, denominada OpenCV, cuyo nombre completo es *Open Source Computer Vision Library*, la cual incluye una gama importante de algoritmos optimizados de visión artificial, destinados a aplicaciones de tiempo real sofisticadas, como el reconocimiento de objetos para el control de procesos, la calibración de cámaras, tratamiento de imágenes, visión robótica y entre muchas otras.

Una de las particularidades de la librería en cuestión es el hecho que proporciona un entorno de desarrollo sencillo de utilizar, así como altamente eficiente, puesto que soporta código en Python, Java, C y C++, además de ello, aprovecha las capacidades que facilitan los procesadores multi-núcleo. OpenCV presenta una estructura modular, esto quiere decir que se compone de sub-librerías compartidas o estáticas, tales como aquellas que se encargan de funciones matemáticas básicas, el módulo de procesamiento de imágenes, otro de estimación de movimientos y localización de objetos. Igualmente, una librería que considera algoritmos de vistas geométricas múltiples, de procesamiento de gráficos, así como de detección de objetos, como rostros, ojos, personas, automóviles, entre otros.

# DESCRIPCIÓN DEL DISEÑO

Primero que todo, el principio de operación de este sistema de vigilancia es muy sencillo: reproducir un sonido, como una sirena, la cual funcione como un tipo de alarma, al momento de detectar algún movimiento en una cámara, con la finalidad de informar y alertar al usuario la ocurrencia de algún fenómeno anómalo donde se disponga dicha aplicación. Para efectuar de manera efectiva la tarea descrita, este sistema considera cinco bloques principales, los cuales fueron mencionados en el inciso anterior: la BeagleBoard-xM, la cámara web, el teclado, altavoces o parlantes y monitor.

La unidad central de procesamiento, en este caso la plataforma BeagleBoard-xM se encarga básicamente de ejecutar en su procesador el programa, cuyo propósito es efectuar la detección de algún movimiento, el cual es visualizado en el monitor con entrada de protocolo de comunicación DVI, también puede ser del tipo Super-Video, gracias a la cámara web de tipo USB, donde si es afirmativo que se produjo algún cambio en la imagen mostrada en pantalla, el algoritmo deberá generar el sonido de una sirena en el puerto de salida de audio que dispone el sistema empotrado en cuestión, donde se adaptan los altavoces, hasta que el usuario, por medio del teclado USB, presionando una tecla en específico, detenga dicha reproducción, y así, la alarma.

Dicho proceso se volverá a ejecutar en el caso que ocurra nuevamente algún movimiento dentro del rango que abarca la cámara web, siendo así un sistema básico de seguridad, el cual perfectamente puede ser empleado en algún recinto donde sea necesario y la seguridad sea un aspecto de suma importancia. La siguiente figura ejemplifica de manera general el diagrama de bloque del sistema diseñado e implementado.

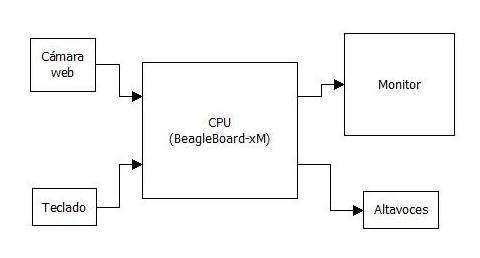


Figura 1. Diagrama de bloques básico del sistema de vigilancia implementado.

Nótese que los elementos que se encuentren a la lado izquierdo del bloque del CPU en la figura anterior corresponden exclusivamente a dispositivos de entrada, y cuya comunicación es USB, como lo son la cámara web y teclado, donde el primero de ellos capta todo lo que se ubique y suceda en dirección a su lente, y el segundo es aquel que permitirá al usuario apagar la alarma cuando esta se active y ya se tenga total conocimiento sobre la ocurrencia de movimiento.

Por su parte, a la derecha del bloque principal, el cual consiste en la BeagleBoard-xM, encargado de realizar todo el procesamiento de los datos de entrada, así como el algoritmo de la detección del movimiento y la reproducción del audio, se encuentra los dispositivos de salida: los parlantes de conector del tipo auxiliar y un monitor DVI o Super-Video, cualquiera de estos dos protocolos los soporta este sistema empotrado. Por supuesto que los altavoces son empleados con el propósito de emitir el sonido proveniente de la BeagleBoard-xM, mientras que el monitor muestra lo que está pasando por la cámara web, con la idea de facilitarle al usuario la vigilancia de la ocurrencia de algún evento capturado por dicha cámara.

# PROCESO DE IMPLEMENTACIÓN

**Posibles ideas de esta sección: Definitivamente, incluir todos los pantallazos que sean necesarios, de la interfaz del cliente, y fotos del servidor trabajando, para corroborar el correcto funcionamiento del sistema de reproducción. También es necesario hablar de todos los pasos que empleamos, procesos de instalación de herramientas, desde correr el ambiente cruzado, pasando por los pasos de la tarea 5, los pipelines en C y, por supuesto, todo lo ejecutado en el Qt, capturas de pantalla de la programación, y etc. De esto se podría encargar César.**

# RESULTADOS OBTENIDOS

**Posibles ideas de esta sección: Explicar básicamente la experiencia o lo que fue que ocurrió cuando el sistema estaba operando, que si se escuchaba bien, que si había algún problema con la lectura de archivos .mp3, posibles errores o pulgas encontrados o que podrían ocurrir eventualmente, por ejemplo, se podría hablar de aquel error que se quedaba ejecutando el pipeline, aún si se cerraba la interfaz de Qt, igualmente se puede mencionar el hecho de que no nos funcionó usando un monitor con entrada HDMI, sino que tuvimos que recurrir a aquel que tuviera un puerto DVI. También cualquier otro inconveniente encontrado que se puede incluir, sería útil y lo que se considere pertinente. De esto se podría encargar César.**

# CONCLUSIONES

* Los sistemas de vigilancia perfectamente pueden ser implementados en plataformas embebidas, sin la necesidad de recurrir a enormes servidores o computadoras de escritorio.
* La librería OpenCV representa un recurso inherente en el desarrollo de aplicaciones destinadas al procesamiento y manipulación de imágenes, de enorme utilidad en la actualidad.
* La herramienta en línea Narcissus permite construir imágenes rootfs prácticamente personalizadas que se adapten a las necesidades de los usuarios y de sus respectivas aplicaciones.

# Referencias (HAY QUE ORDENARLAS)

1. Intel Coporation, “OpenCV Reference Manual”, pp. 1–10, Julio 2001.
2. Open Source Computer Vision Library, “OpenCV 2.4.7.0 documentation”, <http://docs.opencv.org/modules/core/doc/intro.html>, recuperado el 30 de octubre del 2013.
3. Angstrom Distribution, “Narcissus”, <http://narcissus.angstrom-distribution.org/>, recuperado el 23 de octubre del 2013.
4. Cédric Verstraeten, “OpenCV Simple Motion Detection”, <http://www.cedricve.me/2013/02/05/opencv-simple-motion-detection/>, recuperado el 11 de noviembre del 2013.
5. M. López, Linux Hispano, “Instalar OpenCV 2.4.2 en Ubuntu 12.04 LTS” <http://www.linuxhispano.net/2012/11/05/instalar-opencv-2-4-2-ubuntu-12-04/> recuperdado el 30 de octubre del 2013.