Sistema de reproducción de archivos mp3

César Ávila Sánchez

Escuela de Ingeniería Electrónica

TEC

Cartago, Costa Rica

@hotmail.com

Fabricio Quirós Corella

Escuela de Ingeniería Electrónica

TEC

Cartago, Costa Rica

fabisqc0207@gmail.com

***Resumen***—**El presente documento pretende abarcar todo el proceso de desarrollo ejecutado y los resultados obtenidos por el grupo de trabajo, con el objeto de implementar un sistema capaz de reproducir archivos de audio, con extensión .mp3, tanto de manera local en un ambiente servidor, como en uno remoto o cliente.**

Palabras Claves—BeagleBoard-xM, GStreamer, Pipeline, Qt Creator.

# INTRODUCCIÓN

Como el título de este texto bien lo menciona, asimismo se aclara en el inciso de Resumen respectivo, el objetivo de todo este proyecto básicamente es crear una aplicación que permita la reproducción de audio del tipo mp3, lo cual solamente es posible con un programa que sea capaz de manipular todo los recursos multimedia que incluya una computadora o algún sistema embebido en específico, como lo es, dentro de este contexto, la BeagleBoard-xM.

Es por esta razón que, para efectos de este proyecto, se recurre a una herramienta de software en particular, un framework multimedia, el cual es de carácter libre, que soporta una gama importante de sistemas operativos, procesadores y compiladores, denominado GStreamer, cuya principal funcionalidad consiste en la creación de aplicaciones audiovisuales de todo tipo, tales como aquellas destinadas a la visualización de un video, inclusive reproducción remota a la codificación de cierto tipo de archivos de audio, hasta las que permite efectuar la mezcla de audio y video, por mencionar algunos ejemplos.

Entre las principales características que posee el GStreamer, caben destacar las siguientes: presenta una biblioteca de núcleo compacta que no sobrepasa los 500KB, con aproximadamente 65 000 líneas de código, así como un trasiego de datos extremadamente ligero, lo que implica un alto rendimiento y una acumulación de retardos reducida. De la misma manera, facilita una interfaz de programación de aplicaciones (en inglés, API: *Application Programming Interface*) bastante simple y estable, cuyo acceso es posible realizarlo desde una cantidad considerable de lenguajes, con el objeto de permitir el desarrollo de aplicaciones y complementos (*plugins*),

Otra particularidad importante de la herramienta en cuestión consiste en el hecho que maneja una arquitectura inteligente de complementos, los cuales son cargados dinámicamente y proveen recursos para trabajar con una enorme cantidad de elementos multimedia, además de ello, es posible aumentar las capacidades y la funcionalidad del GStreamer, gracias a la inclusión de nuevos complementos, donde estos están organizados en distintas categorías, tal y como se muestra a continuación.

Primero, se tiene el *gst-plugins-base*, que constituye el conjunto de complementos básico, totalmente soportado por el GStreamer. En segundo lugar, se encuentra el *gst-plugins-good* que incluye todos aquellos complementos bien soportados que emplean licencias libres. Luego, está el *gst-plugins-ugly* que considera todos aquellos plugins que posiblemente presenten inconvenientes en su libre distribución. Y por último, se tiene la categoría de *gst-plugins-bad*, la cual contiene aquel conglomerado de complementos que aún no ha superado rigurosas especificaciones de calidad, impuestas por los desarrolladores de GStreamer.

Como una de las características fundamentales que posee este framework, y que es de suma utilidad a la hora de implementar alguna tarea multimedia en particular, corresponde a la capacidad de la utilización deuna ruta total conformada por una serie de elementos vinculados, los cuales son complementos del GStreamer, donde la denominada ruta constituye el *pipeline* de la función, donde existen infinidad de posibilidades de vinculaciones entre plugins; es decir, hay innumerables pipelines a los que se pueden recurrir y crear.

# DESCRIPCIÓN DEL DISEÑO

**Posibles ideas de esta sección: Incluir descripción y cometarios detallados sobre lo que debe hacer el sistema, lo cual es posible ejemplificarlo mediante un diagrama de bloques. Asimismo, se puede hablar sobre las herramientas que se utilizaron tanto de software como de hardware, también de ideas o planteamientos que fueron necesarios incluir para que se obtuvieran los resultados esperados, esto relativo a las herramientas. Esta sección no debería ser muy extensa y no confundir con la siguiente. De esto se podría encargar Fabricio.**

# 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

**Posibles ideas: Esta sección es más que clara, así que no hay mucho que agregar al respecto, solamente decir que deben ser concisas y relacionadas con las observaciones realizadas en los resultados y concisas con los objetivos de este proyecto, los cuales, a mi criterio, fueron trabajar con el GStreamer, la Beagle, el Qt y el GStreamer, conocer todas las posibilidades de la Beagle y los ambientes de trabajo. Identificar todas implicaciones y particularidades del GStreamer y el Qt, y de aquí lo que sea. De esto se podría encargar Fabricio**

# Referencias

1. Cypress Semiconductor Corporation, “PSoC 5: CY8C55 Family Datasheet”, pp. 3–42, Febrero 2012.
2. Cypress Semiconductor Corporation, “PSoC Creator 2.1”, <http://www.cypress.com/?id=2494>, recuperado el 24 de noviembre del 2012.
3. Cypress Semiconductor Corporation, “PSoC Designer”, <http://www.cypress.com/?id=2522>, recuperado el 24 de noviembre del 2012.
4. Cypress Semiconductor Corporation, “PSoC Programmer 3.16”, <http://www.cypress.com/?id=2522>, recuperado el 24 de noviembre del 2012.
5. R. Jiménez, Universidad de Huelva, “Diseño de Sistemas Empotrados: Introducción” <http://www.uhu.es/raul.jimenez/EMPOTRADO/introduccion.pdf>, recuperdado el 20 de noviembre del 2012.