

Aplicación Móvil que Implementa un Catálogo de Circuitos Integrados Reduciendo Requerimientos de Memoria usando SVG

Basurto, Juan C.

Grupo TAWS

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5, vía Perimetral

Facultad de Ingeniería Eléctrica y Computación

jbasurto@espol.edu.ec

1. Introducción

La demanda actual de teléfonos móviles a nivel mundial ha traído consigo no solamente la necesidad de diversificar los servicios, sino que también de crear soluciones a los problemas cotidianos. Con 2.9 mil millones de móviles alrededor del mundo, el desarrollo de aplicaciones móviles para resolver algunos de ellos resulta una alternativa viable para entregar información que el usuario requiere tener a la mano con frecuencia.

Este requerimiento de disponibilidad de información se observa cuando se trabaja en el diseño de Circuitos Electrónicos, pues es necesario consultar constantemente la asignación de pines de integrados Digitales y Analógicos. Un integrado es la reunión de Puertas Lógicas y otros integrados, tiene como objetivo ahorrar espacio, tiempo y dinero en el diseño de un Circuito. Los pines son los medios por los cuales el integrado se conecta con el exterior y cada pin tiene una función diferente. Por el gran número de integrados existentes, memorizar la asignación individual para cada uno es imposible. Una solución preliminar podría ser la consulta en el Web o en catálogos impresos.

Este es un problema que experimentan estudiantes de las materias de los laboratorios de Sistemas Digitales, Electrónica A y Electrónica B, dada la necesidad de tener a mano los medios

de información mencionados, en función del desarrollo de los proyectos que se envían en dichas materias.

Sin embargo, para los estudiantes no es común tener acceso a Internet o a los catálogos impresos fuera del laboratorio, impidiendo así que se pueda trabajar eficientemente en los circuitos que diseñan cuando trabajan en casa. Por lo tanto, el problema consiste en la dificultad de acceder a los medios de información.

Si el problema principal es la accesibilidad, entonces la solución debería estar relacionada a la movilidad. Una solución óptima es el desarrollo de una Aplicación Móvil que dado el número de un integrado recupere su información de un catálogo almacenado en el dispositivo. Es importante que dicho catálogo permita almacenar datos de varios cientos de integrados.

El objetivo principal de este trabajo es presentar una Aplicación Móvil que resuelva este problema, considerando las ventajas de un teléfono celular: la movilidad y el acceso y explorando alternativas para superar una limitación de estos dispositivos: capacidad limitada de memoria.

2. Materiales y Métodos

En el desarrollo de Aplicaciones Móviles, es necesario tomar en cuenta ciertos puntos muy importantes:

- Herramientas de Desarrollo

- Usabilidad
- Espacio que ocupa en Memoria

El IDE escogido para desarrollar la aplicación es Netbeans 5.5.1, pues no solamente dispone del Mobility Pack 5.5+ para trabajar con dispositivos móviles, sino que también ofrece un entorno de programación amigable al disponer de una interfaz gráfica que simplifica el desarrollo de la aplicación. En términos de usabilidad, el proyecto debe cumplir con lo siguiente:

- Debe proveer una interfaz sencilla pero llamativa
- Debe contener la mayor cantidad de integrados posible, tomando en cuenta que la cantidad de memoria que ocupa debe ser mínima

Al implementar la aplicación es necesario en primer lugar determinar la presentación gráfica que se usará.

Para la presentación de cada integrado, es necesario visualizar la asignación de pines, por lo que debe existir una imagen por cada integrado.

Sin embargo, al tener un gran número de elementos es necesario tomar en consideración el tamaño que tiene la imagen en diferentes formatos.

La Figura 1 muestra el integrado 7400 que tiene 14 pines y cuatro puertas NAND.

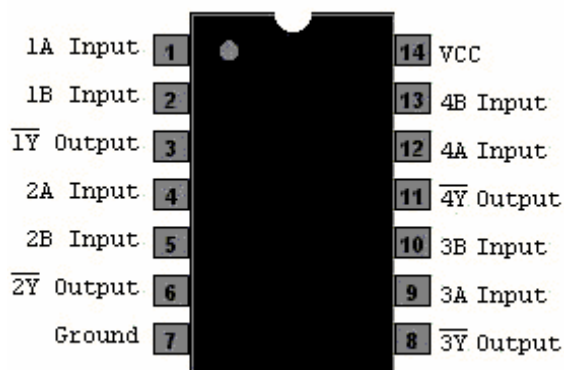


Figura 1. Integrado 7400

La Tabla 1 muestra el tamaño en KB de la misma imagen en varios formatos posibles.

Formato	Tamaño en KB
BMP (256 colores)	59
JPEG	41
GIF	6
PNG	4

Tabla 1. Formato versus Tamaño en KB para la Figura 1

Como se puede observar, el formato que requiere menos espacio es PNG (Portable Network Graphics).

Con 4 KB por imagen, para 200 integrados se requieren 800 KB.

Esto es un problema si en un futuro se necesita ampliar la cantidad de integrados. Y si no es posible aumentar esta cantidad, entonces habrán menos integrados disponibles y el Usuario tiene menos probabilidades de encontrar el que busca.

Además, las imágenes resultan ser muy básicas. Si se aumenta la cantidad de colores que tiene la Figura 1, su tamaño en KB aumentará y por ende el conjunto. Es necesario contar con imágenes que tengan una mejor presentación, pero habría que sacrificar espacio en memoria. Existe un formato denominado SVG cuyas ventajas se analizan a continuación

2.1 Introducción de SVG

SVG es un estándar de la W3C para representación de imágenes vectoriales. El uso de SVG en dispositivos móviles es posible gracias a que se ha definido un subconjunto de elementos de este estándar denominado SVG Tiny exclusivo para uso en celulares. El Wireless Toolkit 2.5.1 incluido en el Mobility Pack 5.5.1 agrega la funcionalidad necesaria para manipular imágenes SVG en aplicaciones móviles desarrolladas usando Netbeans.

La figura 2 muestra tres partes de la Aplicación: Presentación (Figura 2a),

Menú (Figura 2b) y el presentación del integrado (Figura 2c) que en comparación a la Figura 1 resulta ser superior. Sin embargo, la imagen SVG asociada a esta Figura tiene un tamaño de 5 KB, 1 KB más que en la Figura 1, y por ende para 200 imágenes se tendría un total de 1000 KB.

El problema del incremento de memoria requerida se soluciona tomando en cuenta que una imagen SVG guarda la información en texto.

Se puede tener cuatro plantillas de imágenes de integrados de 4, 8, 14 y 16 pines de 5 KB cada una y en un archivo de texto guardar 200 líneas, donde cada línea representa un integrado. El archivo de texto resultante tiene un tamaño de aproximadamente 11 KB.

Cuando se solicita la información de un integrado, se consulta el archivo de texto y se modifica la imagen SVG usando código escrito en Java.

Las imágenes adicionales para presentación, menú y error (si no se encontrara el integrado). Representan en conjunto solamente 22.7 KB.

Debido a que no es posible modificar el contenido dentro un archivo “.jar” y volverlo a compilar (como modificar una imagen SVG dentro del archivo “.jar”), se utilizaron métodos para trabajar con archivos que se encuentren fuera del entorno de la aplicación, de modo que las imágenes SVG se generan en una ubicación determinada de la memoria del dispositivo móvil y se modifican conforme se generen las solicitudes.

3. Resultados

Se obtiene una Aplicación Móvil que resuelve el problema de no disponer de un medio de consulta inmediato para la creación de proyectos electrónicos y que además proporciona una interfaz amigable.

Adicionalmente, se obtuvo un ahorro notable de la memoria requerida para la

aplicación. El paquete generado tiene un tamaño de 31 KB para 200 Integrados. Considerando únicamente el catálogo de los Integrados usando imágenes PNG, el espacio requerido para 200 integrados es 800 KB, mientras que con SVG se requieren solamente 25.4 KB, un ahorro total de 774.6 KB que representa un ahorro porcentual de 3049%. Estas mediciones de espacio no consideran tamaño del código escrito utilizando J2ME que en ambos casos representa 17 KB.

Esta comparación se puede hacer desde otro punto de vista: si se tiene disponible 800 KB de espacio para toda la aplicación, utilizando imágenes PNG se logran almacenar 200 Integrados a razón de 4 KB por imagen. Por otro lado, utilizando SVG se requieren 20 KB para 4 plantillas y un documento de texto de 780 KB con la distribución de los pines, esto en conjunto representa 800 KB en memoria, ya que la información se guarda en el archivo de texto. En estos 780 KB de texto alcanzan 70.9 archivos de texto de 11 KB cada uno. Considerando que cada archivo de texto representa la información de 200 integrados, utilizando el número de archivos mencionado, el catálogo puede contener un total de 14181 integrados aproximadamente, 13981 más de lo que se obtendría usando PNG.

Por otra parte, con respecto al archivo “.jar” ejecutable del programa, para el caso de PNG, el archivo tiene un tamaño de 612 KB, mientras que en SVG son 37 KB. Es decir hay un Ahorro Real de memoria de 575 KB, es decir un 1654 por ciento.

4. Discusión y Conclusiones

El desarrollo de aplicaciones móviles debe enfocarse a la solución de problemas cotidianos.

Gracias a la implementación de imágenes SVG Tiny en aplicaciones para celulares, es posible no solamente mejorar la presentación, sino que también es posible ahorrar un gran espacio en memoria del celular.

El uso de Netbeans como IDE facilita el trabajo con aplicaciones móviles gracias a su ambiente de trabajo amigable.

El ahorro en memoria es importante por cuanto permite el crecimiento del

número de elementos en un proyecto. Esto a su vez permite aumentar las probabilidades para el Usuario de encontrar lo que desea.

5. Agradecimientos

Este trabajo se pudo llevar a cabo gracias a la estandarización W3C de imágenes SVG Tiny investigado por TAWS y a la Ing. Carmen Vaca por su ayuda en la redacción de este trabajo.



Figura 2. Aplicación Móvil.