

# Plataforma de desarrollo digital basado en Game Boy Advance y la arquitectura ARM7

*Digital development platform based on Game Boy Advance and the ARM7 architecture*

Jhon A. Esquivel

Makser S.A.S.

jaesquivel@correo.udistrital.edu.co

Norbey Marín Moreno

Kasai Ltda

nmarinm@correo.udistrital.edu.co

Fernando Martínez Santa

Universidad Distrital Francisco José de Caldas

fmartinezs@udistrital.edu.co

En este artículo se presenta el desarrollo de una plataforma base para aplicaciones digitales basadas en la arquitectura ARM7 del GBA (Game Boy Advance). Este desarrollo pretende incursionar en el manejo de sistemas embebidos utilizando la arquitectura ARM7 y el hardware disponible en la consola de video juegos GBA. La plataforma está compuesta de una interfaz de comunicación entre la GBA y una tarjeta de interfaz de señales externas desarrollada con un microcontrolador, la cual se ha llamado PICGBA\_01. Esta plataforma permite utilizar los periféricos incluidos en la consola GBA, y adicional a ellos, los módulos disponibles en la tarjeta PICGBA\_01. La plataforma fué evaluada exitosamente en laboratorio mediante un conjunto de aplicaciones básicas de seguimiento y control.

*Palabras clave:* ARM7, GBA (Game Boy Advance), plataforma digital

This paper presents the development of a platform for digital applications based on the ARM7 architecture GBA (Game Boy Advance). This development aims to break into the management of embedded systems using the ARM7 architecture and the hardware available on the GBA console video games. The platform consists of a communication interface between the GBA and a data acquisition board developed with a PIC16F877A microcontroller from Microchip which we have called PICGBA\_01. This platform allows the use of peripherals included on the GBA console and additional modules available to them on the card PICGBA\_01. The platform was successfully tested in laboratory by means a set of basic applications for monitoring and control.

*Keywords:* ARM7, digital platform, GBA (Game Boy Advance)

## Introducción

Los sistemas embebidos ocupan un lugar importante en la electrónica de consumo, y se han convertido en herramientas fundamentales para la solución de necesidades de bajo costo y alto desempeño (Jensen, Lee, y Seshia, 2011; Yang, Chang, Wu, Huang, y Luo, 2011). Estos sistemas tienen la característica de ser flexibles o adaptables, lo cual garantiza el poder aprovechar sus cualidades tecnológicas con el fin de hacer herramientas multifuncionales y modificables para mejorar el rendimiento de un producto electrónico en particular.

La consola de video juegos GBA es un sistema embebido que cuenta con un procesador ARM7, y a este se integra un

grupo de periféricos: LCD, botones, sonido, cartucho y sistema de comunicación. Este conjunto conforma un sistema con una potencia suficiente para permitir el desarrollo de juegos.

El objetivo que se busca cumplir con el desarrollo de este proyecto es la construcción del hardware y software necesario para el intercambio de información entre la GBA y una tarjeta que permita el manejo de señales externas, con el fin de integrarlas a los recursos disponibles en la consola. Esto permite aprovechar su procesador de arquitectura ARM7 sin modificar sus características originales (Chun y Zhen, 2011; de Lima Lages y da Silva-Filho, 2011; Pohronska y Krajcovic, 2009). De esta forma se obtiene una plataforma que facilita el diseño de aplicaciones y permite generar un nuevo uso, encaminando la consola de video

juegos a un entorno académico. Con este nuevo uso se busca situar las bases de conocimiento para la interacción con la arquitectura ARM y una futura adaptación en diferentes aplicaciones electrónicas. Es así como en el presente artículo se muestra el cumplimiento de los objetivos propuestos y su descripción en particular.

### Metodología

Para el desarrollo del proyecto se hizo necesario en primera instancia el estudio del funcionamiento y las características técnicas de la GBA, así como la elección de un compilador que permitiera programar la consola en un lenguaje de alto nivel, en este caso C. Seguidamente se buscó una interfaz que facilitara el acceso a la memoria de arranque de la GBA con el fin de cargar los programas realizados. Ya teniendo la posibilidad de ejecutar programas dentro de la consola se procedió a generar una interfaz de comunicación bidireccional entre la tarjeta PICGBA\_01 y la GBA, generando así la posibilidad de interactuar con señales externas aplicándolas a los procesos dentro de la consola. Finalmente, a través de pruebas prácticas se optimizó la plataforma completa, teniendo como resultado el dominio de las herramientas de programación, un hardware apropiado para la interacción con señales externas al GBA y un protocolo de comunicación entre estos.

### Hardware del sistema

**Microprocesador ARM7.** Se denomina ARM (*Advanced RISC Machines*) a una familia de microprocesadores RISC (*Reduced instruction set computing*) diseñados por la empresa Acorn Computers y desarrollados por Advanced RISC Machines Ltda, una empresa derivada de la anterior. La utilización de la tecnología ARM se generalizó con el procesador ARM7 TDMI (*Thumb instruction sets*), con millones de unidades en teléfonos móviles y sistemas de videojuegos portátiles entre estos la GBA (Chien y Tai, 2006; Li, Liu, Ye, Zhang, y Li, 2011).

El microprocesador ARM7 posee un bus de datos de 32 bits, cuenta con un total de 37 registros, 31 de pro-

pósito general y 6 registros de estado. El ARM7 emplea una arquitectura conocida como THUMB lo que lo convierte en ideal en aplicaciones de gran volumen donde la densidad de código es un gran problema. La idea clave de THUMB es un set de instrucciones reducido a 16 Bits con el mismo efecto de las instrucciones de 32 Bits (ARM, 1995; Barahan, Custodio, Madamba, y Roque, 2011; Yujun Bao y Jiang, 2009) (Fig. 1).

**Game Boy Advance (GBA).** La Game Boy Advance (GBA) es una popular consola de videojuegos de la compañía Nintendo, fabricada desde marzo de 2001 hasta 2008.

### Características:

1. Procesador ARM/TDMI de 32 Bits a 16.78 MHz que permite programación por medio de lenguaje C.
2. Memoria:
  - System ROM: 16 Kbytes
  - Working RAM: 32 Kbytes
  - VRAM: 96 Kbytes
  - OAM: 64 bits x 128
  - Palette RAM: 16 bits x 512 (256 colores para OBJ, de 256 colores para BG)
  - Game Pack: Hasta 32 MB
3. Display:
  - 240 x 160 puntos RGB
  - 32.768 colores simultáneamente
  - Efectos especiales: la rotación y escala, una mezcla, *fade-in/fade-out*, y mosaico
  - Cuatro modos de imagen de sistema
4. Operación:
  - Teclas de operación (A, B, L, R, START, SELECT, y cuadro de control)
5. Sonido:
  - Cuatro sonidos (que corresponden a sonidos CGB) + 2 CPU directo sonidos (formato PCM)
6. Comunicación:
  - Comunicación serie (8 bit/32) (Chien y Tai, 2006) (Fig. 2)

### Herramientas de programación

**Compilador: DevkitPro.** Para la programación de aplicaciones para la GBA se utiliza el compilador *DevkitPro*. Este es basado en el GCC (Colección de Compiladores GNU), el cual por ser software libre y de código abierto permite ser descargado de forma gratuita (Prieto y Jacinto, 2011).

Este entorno de programación se encarga de traducir las instrucciones de los programas a instrucciones máquina, que son ejecutadas por el procesador de la GBA. La instalación de este entorno permite tener acceso a las herramientas:

- DevkitARM: Sirve para programar el ARM7 de la GBA (incluye librerías libgba).
- ProgrammersNotepad: Es un Editor de Texto con resaltado de sintaxis, el cual permite invocar el compilador en len-

---

Fecha recepción del manuscrito: Octubre 17, 2012

Fecha aceptación del manuscrito: Diciembre 10, 2012

Jhon A. Esquivel, Makser S.A.S.; Norbey Marín Moreno, Kasai Ltda; Fernando Martínez Santa, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas.

Esta investigación fue financiada por: Universidad Distrital Francisco José de Caldas.

Correspondencia en relación con el artículo debe ser enviada a: Jhon A. Esquivel. Email: jaesquivel@correo.udistrital.edu.co

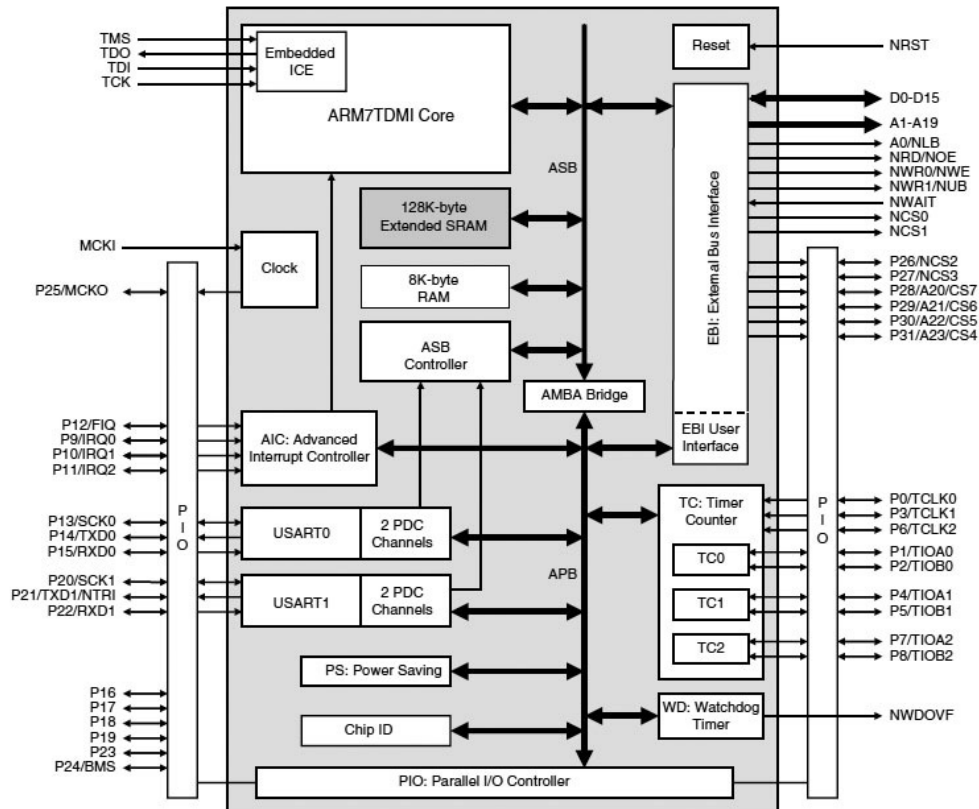


Figura 1. Arquitectura ARM7 TDMI (ARM, 1995).

guaje C. Este es muy útil ya que facilita la edición del código y evita la compilación de este desde la línea de comandos.

- **Insight:** Son un conjunto de pequeñas aplicaciones utilizadas para depurar el código.

#### Herramienta para la creación de gráficos USENTI.

USENTI es una herramienta que permite crear gráficos para la GBA. Este permite crear y editar imágenes y exportarlas a un formato que el GBA pueda leer. Con esta herramienta se trabajan imágenes en formato Bitmap de 256 o 16 colores exportándolas en forma de arreglos en lenguaje C con extensiones .h y .c, que posteriormente permiten ser compiladas con el DevkitPro (Fig. 3).

**Emulador: Visual BoyAdvance.** El Visual BoyAdvance es un emulador de la consola portátil desarrollada por Nintendo. Este emulador hace parte de los programas de dominio público que permite visualizar los programas realizados en DevkitPro antes de ser cargados en la consola GBA (Fig. 4).

#### Programación del Game Boy Advance

Con el objetivo de tener un punto de referencia para la creación de aplicaciones, se debe conocer y estandarizar: los tipos de datos, los nombres de registros, las definiciones y las librerías. Esto se hace por la forma en la que se almacenan los datos en la memoria, ya que se emplean registros con direcciones hexadecimales.



Figura 3. Captura: ventana principal del USENTI.

El lenguaje C da la posibilidad de crear macros o cabeceras en archivos con una extensión .h, que son utilizados para facilitar el proceso de programación. El paquete básico que permite programar la GBA en el DevkitPro se llama DevkitARM, en el cual se incluyen librerías que contienen las definiciones de los registros y los tipos de datos que se pueden manipular en la creación de aplicaciones, estos archivos se incluyen en la carpeta libgba del compilador.

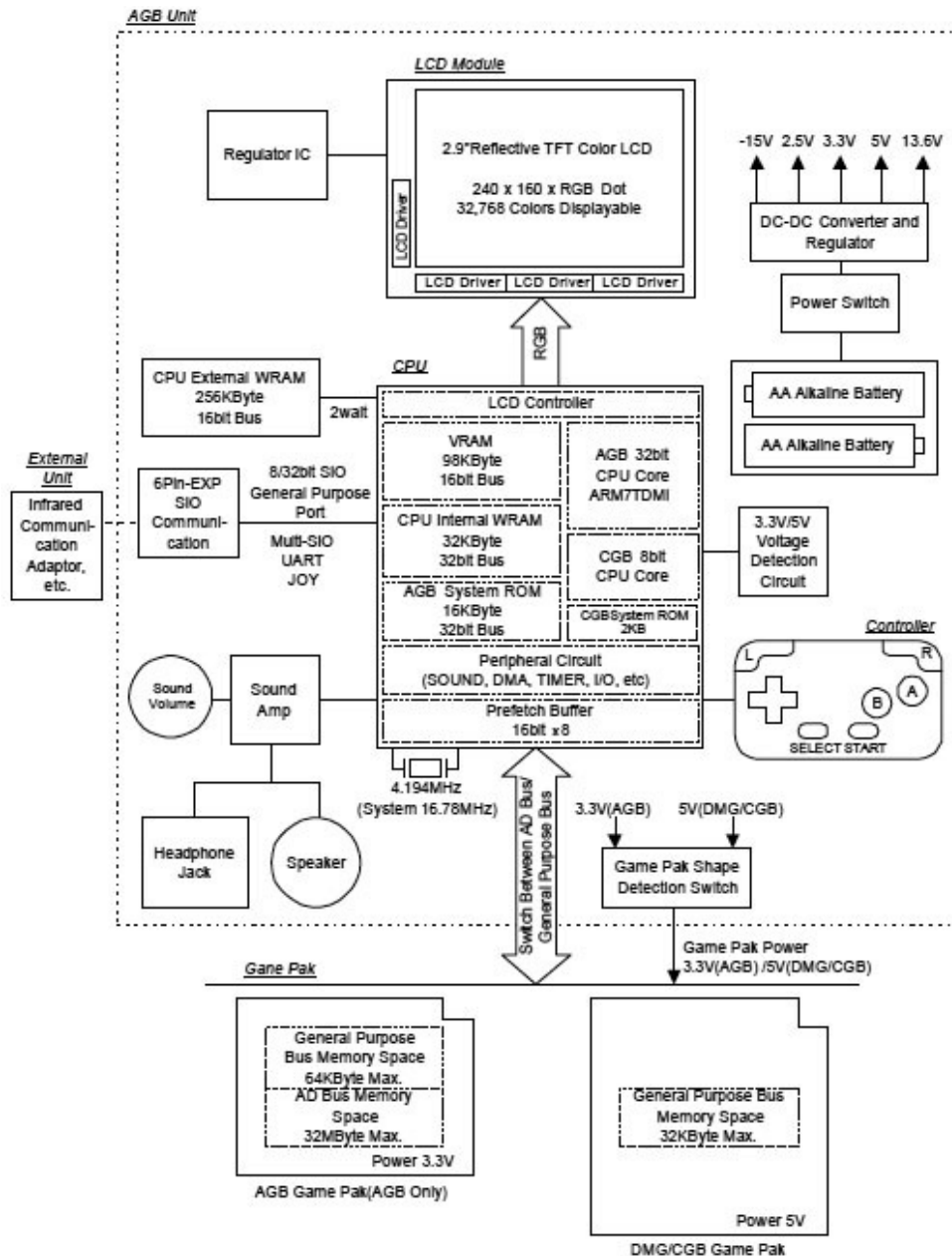


Figura 2. Diagrama de bloques de la GBA (Chien y Tai, 2006).

**Cabeceras principales.** En la carpeta libgba se encuentra la cabecera gba\_base.h que define las direcciones de memoria, las cuales dan acceso a diferentes efectos en la GBA, dependiendo del uso que se les quiera dar. La configuración de memoria de la GBA en términos generales es la siguiente: el área 0h - 07FFFFFFh se asigna como la memoria interna del GBA y el área comprendida entre 08000000h-0EFFFFFFh se asigna como memoria de juego.

Las definiciones utilizadas en la cabecera gba\_base.h para hacer referencia a cada una de las direcciones de memoria son:

- VRAM: Cuenta con 96Kbytes a partir de la dirección 06000000h y es usada para BG y Objetos:

```
#define VRAM 0x06000000
```

- IWRAM: La Memoria Interna de Trabajo cuenta con 32Kbytes a partir de la dirección 03000000h:

```
#define IWRAM 0x03000000
```

- EWRAM: La Memoria de Trabajo Externo cuenta con 256Kbytes a partir de la dirección 02000000h:





Figura 4. Emulando en Visual BoyAdvance.

- ```
#define EWRAM 0x02000000
#define EWRAM_END 0x02040000
```
- I/O y registros: Área usada para varios registros:

```
#define REG_BASE 0x04000000
```

Otra de las cabeceras principales que tiene el compilador es la `gba_types.h` en la cual se define el tipo y tamaño de los datos que se van a usar en la programación.

**Sintetización de la programación.** Al interactuar con el entorno de desarrollo original del DevkitPro se observa que utiliza estructuras de configuración bastante elaboradas pero de difícil comprensión y utilización. Es por esta razón que se modificaron algunas de las cabeceras más utilizadas dentro de las aplicaciones generadas en este proyecto con el fin de hacer más comprensible y reducido el proceso de programación. Estas cabeceras son:

- `gba_keys.h`: Cabecera utilizada para interactuar con los 10 botones disponibles en el hardware de la GBA.
- `gba_video.h`: Cabecera en la cual se configuran los seis modos de video permitidos en la GBA, los cuales se manipulan con el registro `REG_DISPCNT`.
- `gba_sprites.h`: Un sprite es todo gráfico que tenga movimiento en la pantalla (BG) sin afectar a ésta. En esta cabecera se definen los atributos para generar movimientos de estos gráficos.
- `gba_bg.h`: El *background* (BG) es un gráfico que no tiene movimiento en la pantalla. Los registros para manipular los BG se encuentran definidos entre las direcciones `0400008h` hasta `0400001Eh`.

### Comunicación entre el computador y la GBA

La GBA tiene una característica llamada *multiboot*, la cual permite cargar un programa en la RAM interna, o también llamada memoria de arranque, en la cual se puede ejecutar una aplicación en el GBA sin necesidad de una memoria externa. Para hacer uso del modo *multiboot* es necesario el cable Xboo y el software Xcomm.

Esta herramienta fue creada por Martin Korth (ARM, 1995). Usa un protocolo de comunicación serial síncrono

donde hay una línea de datos para transmitir, una línea de datos para recibir y una línea de reloj generada por el maestro. En este modo es posible transmitir datos de 8 y 32 bits (Fig. 5).

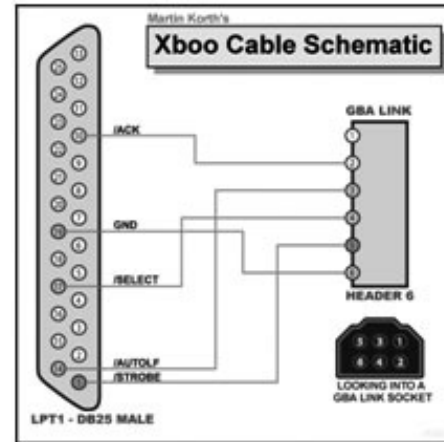


Figura 5. Cable Xboo (ARM, 1995).

El cable *multiboot* es conectado al puerto paralelo del computador y al puerto serie de la GBA. Para enviar las aplicaciones al GBA, se utiliza el software Xcomm, que toma las aplicaciones en formato `*.gba` o `*.elf` y las almacena en la memoria de arranque de la GBA, la cual tiene una capacidad máxima de 256 Kbytes (Fig. 6).

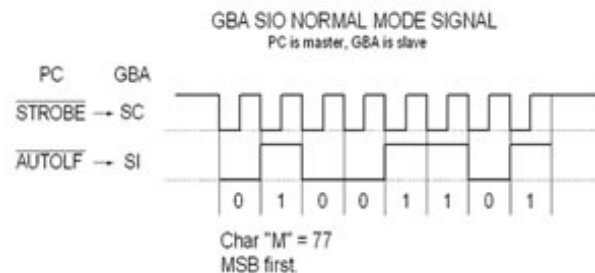


Figura 6. Protocolo de comunicación Multiboot.

### Tarjeta de interfaz de señales externas PICGBA\_01

La tarjeta `PICGBA_01` fue diseñada para ser controlada por los procesos que se ejecutan dentro de la GBA, permitiendo a la consola interactuar de una forma bidireccional con el exterior. El núcleo de la tarjeta `PICGBA_01` es un microcontrolador `PIC18F452`, en el cual se han dispuesto ocho entradas digitales, 4 entradas análogas, ocho salidas digitales y 2 salidas PWM. La `PICGBA_01` se comunica con la GBA vía RS232 a velocidad de 115.200 Baudios, siendo esta la máxima velocidad disponible entre los dos dispositivos (Fig. 7).

**UART en la GBA.** Para construir las funciones que permiten configurar y utilizar el modo UART en la GBA se hace




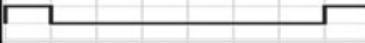
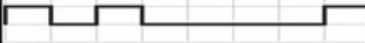


| Modulo              | Sintaxis | Codificación del Byte |             |                                                                                    |
|---------------------|----------|-----------------------|-------------|------------------------------------------------------------------------------------|
|                     |          | Binario               | Hexadecimal |                                                                                    |
| Salidas Digitales   | OUT_D    | 0b01100001            | 0x61        |  |
| Entradas Digitales  | IN_D     | 0b10000001            | 0x81        |  |
| Entradas Analógicas | IN_A     | 0b10100001            | 0xA1        |  |
| Salida PWM 1        | PWM1     | 0b11000001            | 0xC1        |  |
| Salida PWM*         | PWM2     | 0b11100001            | 0xE1        |  |

Figura 8. Protocolo de comunicaciones: Byte comando.

3. Aplicación que permite controlar y monitorear la velocidad de un motor DC, utilizando las salidas PWM y las entradas análogas de la tarjeta PICGBA\_01. Dicha aplicación tiene visualización en la pantalla de la GBA.

- Documentación referente al proyecto, con la finalidad de que pueda ser utilizada en futuros proyectos dentro del grupo de investigación DIGITI que utilicen procesadores con arquitectura ARM. Esta documentación incluye:

1. Arquitectura del microprocesador ARM7 TDMI.
2. Arquitectura de la Game Boy Advance.
3. Instalación y uso del compilador.
4. Definiciones, cabeceras y funciones.
5. Comunicación PC-GBA.
6. Manual de la tarjeta PICGBA\_01.

### Conclusiones

La arquitectura ARM es una opción muy versátil para el desarrollo de aplicaciones digitales robustas. Esta ofrece la posibilidad de usar un set de instrucciones de 32 bits (ARM) y otro de 16 bits (THUMB) con resultados similares.

El hardware disponible en la consola GBA facilita la implementación de aplicaciones gráficas, haciendo más fácil plasmar diseños gráficamente complejos, los cuales tendrían un mayor grado de dificultad en su implementación si no se contara con dicho hardware.

Al usar un compilador basado en GCC para programar los microprocesadores ARM es posible extender el trabajo a otras consolas o dispositivos que utilicen la arquitectura ARM, teniendo en cuenta que las librerías contenidas en el DevkitARM son un estándar para este tipo de procesadores. GCC es un compilador estándar que da la posibilidad de implementar cualquier algoritmo escrito en C, en este caso, con la única limitación de espacio de memoria.

Al implementar un protocolo de comunicaciones usando RS232 se logra tener una comunicación satisfactoria a una velocidad de 115200 Baudios, lo cual presenta una limitación en cuanto a la velocidad de procesamiento de las señales externas. Esto último puede ser mejorado utilizando el

puerto Game Pack de la GBA, con el cual es posible una comunicación directa con el procesador ARM.

La programación en lenguaje C presenta la ventaja de desarrollar aplicaciones en un menor tiempo comparado con la programación en lenguaje ensamblador. Además, como el ARM es un procesador de 32 bits, aprovecha de mejor forma las posibilidades y ventajas del lenguaje C.

Con la aplicación implementada para la programación de entradas y salidas, se demuestra que no es necesario el uso de un PC para realizar programas de interacción entre el GBA y el mundo externo.

### Referencias

- ARM. (1995). ARM 7TDMI data sheet (E ed.) [Manual de software informático].
- Barahan, Custodio, J. J. M., Madamba, J. A. R., y Roque, C. R. K. (2011). SCarm : A memory simulator with a compiler-assembler for the 32 bit arm7 microprocessor. En *Tencon 2011 - 2011 ieee region 10 conference* (p. 1409-1413). doi: 10.1109/TENCON.2011.6129041
- Chien, J.-R. C., y Tai, C.-C. (2006). The design of a portable ECG measurement instrument based on a gba embedded system. En *Ieee international conference on industrial technology icit 2006* (p. 1782-1787). doi: 10.1109/ICIT.2006.372490
- Chun, W., y Zhen, L. (2011). Research on arm7-based USB nc system. En *2011 international conference on system science, engineering design and manufacturing informatization (icsem)* (Vol. 2, p. 328-330). doi: 10.1109/ICSSEM.2011.6081311
- de Lima Lages, D., y da Silva-Filho, A. G. (2011). An analysis energy consumption tool for arm7 processor. En *2011 brazilian symposium on computing system engineering (sbesc)* (p. 108-110). doi: 10.1109/SBESC.2011.40
- Jensen, J. C., Lee, E. A., y Seshia, S. A. (2011). An introductory capstone design course on embedded systems. En *2011 ieee international symposium on cir-*

- uits and systems (iscas)* (p. 1199-1202). doi: 10.1109/ISCAS.2011.5937784
- Li, L., Liu, T., Ye, Y., Zhang, Y., y Li, J. (2011). Embedded arm-based automatic gate bias control system for ldmos RF power amplifiers. En *2011 international conference on electronics, communications and control (icecc)* (p. 2636-2639). doi: 10.1109/ICECC.2011.6066342
- Pohronska, y Krajcovic, T. (2009). Arm7 based embedded system for education. En *Applied electronics ae 2009* (p. 207-210).
- Prieto, J. M., y Jacinto, E. (2011). Sistema de control de acceso biométrico sin contacto con tarjeta inteligente. *Revista Tekhnê*, 8(1), 13-20.
- Yang, K.-C., Chang, Y.-T., Wu, C.-M., Huang, C.-M., y Luo, H.-H. (2011). Application-oriented teaching of embedded systems. En *2011 ieee international conference on microelectronic systems education (mse)* (p. 118-121). doi: 10.1109/MSE.2011.5937109
- Yujun Bao, C., y Jiang, X. (2009). Construction of embedded system platform which based on  $\mu$  c/os-ii and arm7 Kernel microprocessor. En *International symposium on computer network and multimedia technology cnmt 2009* (p. 1-4). doi: 10.1109/CNMT.2009.5374648