

Adquisición de imágenes a través del sistema embebido *ADSP-BF533* de *Analog Devices*

Acquisition of images using the embedded system *Analog Devices ADSP-BF533*

Carolina Zuluaga Arias, Daniel Zuluaga Arias, Edwin Andrés Quintero Salazar
Programa de Ingeniería Electrónica, Universidad Tecnológica de Pereira, Pereira, Colombia
 carozuluaga@utp.edu.co
 daniel_ing2006@utp.edu.co
 equintero@utp.edu.co

Resumen— En este documento se presentan los resultados obtenidos al desarrollar un sistema embebido de adquisición de imágenes basado en el procesador digital de señales *ADSP-BF533* de la marca *Analog Devices*®. Así mismo, se realiza una descripción general del entorno de desarrollo integrado *VisualDSP++5.0*, incluyendo los pasos necesarios para la inicialización, carga, y puesta en operación del algoritmo que permite almacenar las imágenes en los bancos de memoria del dispositivo; propiciando de esta manera la posterior aplicación de diferentes algoritmos propios del procesamiento digital de imágenes.

Palabras clave— Sistema embebido, procesador digital de señales, tratamiento digital de imágenes, sistema en tiempo real.

Abstract— In this document we present the results obtained after developing a system embedded with the acquisition of images based on the digital signal processor *ADSP-BF533* of the brand *Analog Devices*®. Thus, a general description of the integrated development environment *VisualDSP++5.0* is carried out, including the necessary steps for the initialization, loading and operating of the algorithm that allows to keep images in the memory banks of the device, this way, promoting the subsequent application of different specific algorithms of the digital process of the images.

Key Word — Embedded system, digital signal processor, digital image processing, real-time systems.

I. INTRODUCCIÓN

Los sistemas de procesamiento de información en tiempo real han encontrado gran aplicación en diferentes áreas del conocimiento tales como medicina, seguridad, comunicaciones, control de calidad, física, astronomía, química, etc.; ya que permiten adelantar el tratamiento de una gran cantidad de datos en intervalos de tiempo muy inferiores a los utilizados por los sistemas de cómputo convencionales. Esto es posible gracias a que en este tipo de dispositivos todos los recursos físicos disponibles se ponen al servicio de la aplicación de interés, de tal manera

que no son desperdiciados en tareas que no tienen ninguna relación con el proceso en cuestión [1].

Uno de los sistemas embebidos más utilizados actualmente es el procesador digital de señales (*DSP*). Este dispositivo reúne en una tarjeta electrónica, además del procesador principal, una serie de elementos periféricos que lo convierte en una excelente opción a la hora de desarrollar sistemas de tratamiento en tiempo real. Entre los dispositivos más comunes es posible citar osciladores, puertos *RS232* y *RS485*, unidades *UART*, puertos paralelos, entre otros. En el caso particular de la *DSP* referencia *ADSP-BF533* desarrollada por *Analog Devices*®, además de los elementos anteriormente mencionados, también se encuentran codificadores y decodificadores de audio y video, con sus respectivas entradas y salidas tipo *RCA*, los cuales convierten este dispositivo en ideal para aplicaciones multimedia que demandan tiempos de procesamiento relativamente bajos para paquetes de datos más grandes [2].

Sin embargo, la utilización de este tipo de procesadores digitales de señales implica un amplio dominio del hardware asociado al mismo, al igual que el conocimiento de la herramienta de desarrollo incluida con la tarjeta. Así pues, el presente documento presenta los resultados obtenidos al realizar la adquisición de imágenes mediante la tarjeta *ADSP-BF533* de *Analog Devices*®, con el objetivo de documentar la utilización de este tipo de equipo electrónico. Así mismo, se realiza una descripción de los pasos seguidos en el hardware y en el software con el fin de cargar las imágenes a procesar en la memoria interna de la *DSP*.

II. CONTENIDO

1. DESCRIPCIÓN GENERAL DE LAS UTILIDADES MULTIMEDIA

En la figura 1 es posible observar la estructura interna del procesador *ADSP-BF533*. Como característica especial, vale la pena mencionar que este contiene un codificador de vídeo *ADV7171* y un decodificador de vídeo *ADV7183*, los cuales le permiten convertir vídeo analógico en formato *PAL* o, *NTSC* a señales digitales *CCIR601/CCIR656*, y viceversa [3]. El

codificador y decodificador de procesamiento de vídeo están conectados a la interfaz periférica, conocida como puerto paralelo (*PPI*). De esta forma, esta tarjeta de procesamiento digital de señales facilita enormemente el tratamiento de imágenes, ya que no es necesario implementar algoritmos de codificación y decodificación de los sistemas de vídeo más utilizados

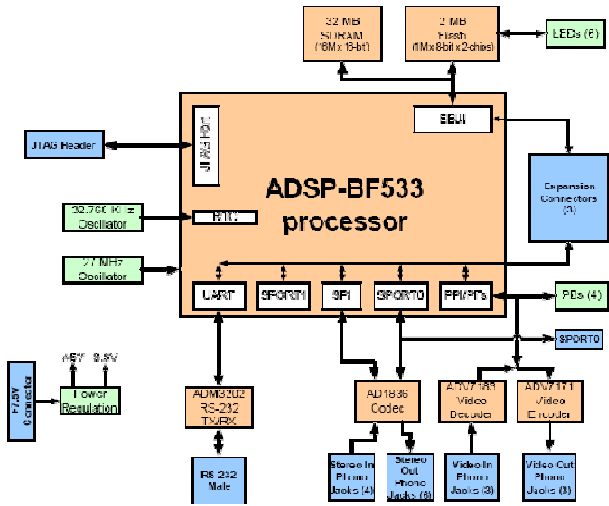


Figura1. Diagrama de bloques de depuración de la ADSP-BF533. [2]

2. DESARROLLO DEL ALGORITMO

La matriz de pixeles que resulta al pasar las señales analógicas a través del decodificador de vídeo *ADV7183* debe ser inicialmente almacenada en la memoria interna del dispositivo, razón por la cual es necesario adelantar la inicialización de la *SDRAM*, y la configuración de la *DMA* y de la *EBIU*, para de esta forma transferir los datos directamente a la memorias, propiciando de esta manera el flujo de la información a través del *Bus* principal sin mayores contratiempos. También es necesario configurar el puerto paralelo *PPI* para establecer los registros en modo de entrada.

En la figura 2 es posible observar el diagrama de flujo que describe el proceso de captura de las imágenes en el procesador digital de señales *ADSP-BF533*. Cada una de las etapas que intervienen en el procedimiento se describen a continuación.

2.1. Configuración de la *SDRAM*.

La inicialización del controlador *SDRAM* (*SDC*) es necesaria para permitirle al procesador la transferencia de datos hacia y desde la *DRAM* de forma sincrónica. El controlador de memoria *SDRAM* puede ser programado para soportar una conexión de cuatro bancos internos dentro de la *SDRAM*.

Los procesadores *Blackfin*® también son apoyados por un espacio de memoria externa asíncrona para *RAM* estática y espacio de memoria síncrona para *RAM* dinámica, como los dispositivos *SDRAM*.

Inicialmente, el algoritmo diseñado ordena leer los píxeles bloque por bloque de la *SDRAM* para el posterior procesamiento de los mismos. Acto seguido, el *PPI* captura el siguiente cuadro de vídeo, mientras que el núcleo de otro búfer se encuentra procesando el bloque de datos anterior. Puesto que, tanto el núcleo principal como la *DMA* están accediendo a la memoria *SDRAM* de forma simultánea, se hace necesario involucrar en el código al *Video Frame* y otros *Buffers*, para de esta manera reducir al mínimo la latencia [3].

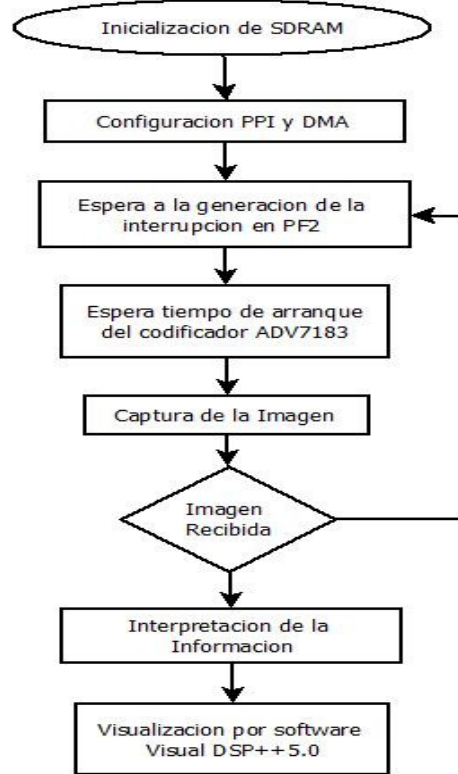


Figura 2. Diagrama de flujo del algoritmo desarrollado para la captura de la imagen.

Al realizar la captura de la imagen, tanto el registro *Code*, como los *Buffers Frame* de vídeo, son asignados a la *SDRAM* interna del *Banco 0*. Este método de asignación causa latencia en la *SDRAM*, porque los ciclos de activación por fila se producen en casi todos los ciclos, debido a la alternancia de los accesos principales (búsqueda de las instrucciones), y al acceso de la *DMA* a las distintas páginas dentro del mismo banco interno de la *SDRAM*. [3]

La latencia puede provocar un error de desbordamiento del *PPI* tipo *FIFO* (en el caso de la captura de imágenes) o inferior (en el caso de la visualización de la imagen). Con el fin de aumentar el rendimiento de los accesos a memoria externa, es necesario asignar *video-buffers* de referencia, de tal manera que sólo

la *DMA* acceda a un banco de la *SDRAM* en cualquier instante de tiempo, tal como se observa en la figura 3.

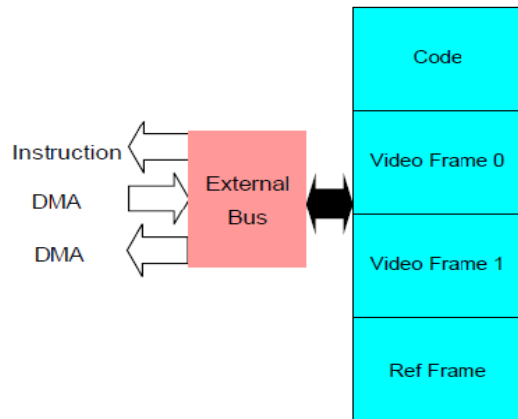


Figura 3. Asignación de memoria *SDRAM* optimizada. [3]

En la figura 3 se muestra la distribución optimizada de la memoria *SDRAM* dentro de los bancos internos. En este ejemplo, la asignación de memoria se da en cualquier instante de tiempo, reduciendo la latencia al mínimo, ya que los ciclos de activación de filas se extienden por cientos de accesos a la *SDRAM*.

2.2. Configuración del Puerto PPI

La configuración como entrada de la interfaz periférica paralela (*PPI*), posibilita la comunicación directa con el decodificador de video *ADV7183*, lo cual permite comenzar con la digitalización de la imagen capturada [4]. En este puerto es necesario configurar lo siguiente:

- El ancho de bus programable (8 a 16 bits en pasos de 1-bit).
- La interfaz paralela bidireccional (*half-duplex*).
- Interfaz síncrona.

2.3. Acceso Directo a Memoria.

El procesador utiliza el acceso directo a memoria (*DMA*) para transferir datos dentro de los espacios de memoria, o entre un espacio de memoria y un periférico [4]. El procesador puede especificar las operaciones de transferencia de datos y volver al proceso normal, mientras que el controlador *DMA* realiza la transferencia de datos independiente de la actividad del procesador [4].

2.4. Configuración del EBIU.

La Unidad de interfaz de bus externa (*EBIU*) proporciona una interfaz de conexión para memorias externas. El procesador soporta sincronismo con la *DRAM* (*SDRAM*) y es compatible con la *PC100* y *PC133 SDRAM* estándar. El *EBIU* atiende el servicio de las peticiones de la memoria externa del núcleo o de un canal *DMA*. La prioridad de las solicitudes está determinada por el bus. La dirección de la

solicitud realizada determina si esta es atendida por el controlador *EBIU SDRAM* o por el controlador de memoria asíncrono *EBIU*.

2.5. Configuración del SIC

El sistema de control de interrupción (*SIC*) se encarga de interrogar a diferentes bits de estado de interrupción para determinar el origen de la misma. Una de las primeras instrucciones ejecutadas en una rutina de interrupción debe ser la *SIC_ISR* para determinar si más de uno de los periféricos de entrada ha afirmado su salida de la interrupción.

Durante la configuración de las interrupciones, el pin 6 del interruptor 3 (*SW3*) debe estar activo. Una vez verificado este requerimiento, con las siguientes líneas de código es posible habilitar el canal de video.

$$*pSIC_IMASK=0x00000100 \rightarrow (1)$$

Esta instrucción crea una máscara de forma tal que solo las interrupciones del chip *ADV7183* (decodificador de video) son percibidas.

El sistema de registro de máscara de interrupción (*SIC_IMASK*), permite enmascarar cualquier fuente de interrupción periférica en el controlador de interrupción del sistema (*SIC*), independientemente de si está habilitada en el propio periférico.

2.6. Arquitectura del Sistema

Para el desarrollo del proyecto se utilizó una cámara réflex marca *Panasonic*® modelo *LUMIX*, una tarjeta *DSP* de alto desempeño *ADSP-BF533*, el software de administración de recursos *VisualDSP++5.0*, y una computadora personal *HP 530*. En la figura 4 es posible observar los elementos que intervienen en el desarrollo de la aplicación.

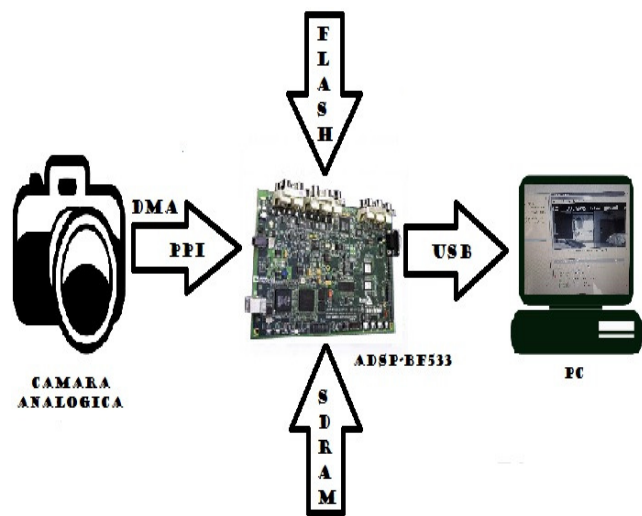


Figura 4. Arquitectura del sistema.

A continuación se describen los pasos que se deben seguir desde el software *VisualDSP++* para adelantar la captura de las imágenes de interés.

Paso 1: Se ejecuta el proyecto o código fuente (en este caso "BF533_EZ_KIT_Video_Receive_C.dpj"), en el entorno de desarrollo integrado *VisualDSP (IDDE)*, el cual será de vital importancia para realizar la adquisición de la imagen (figura 5).

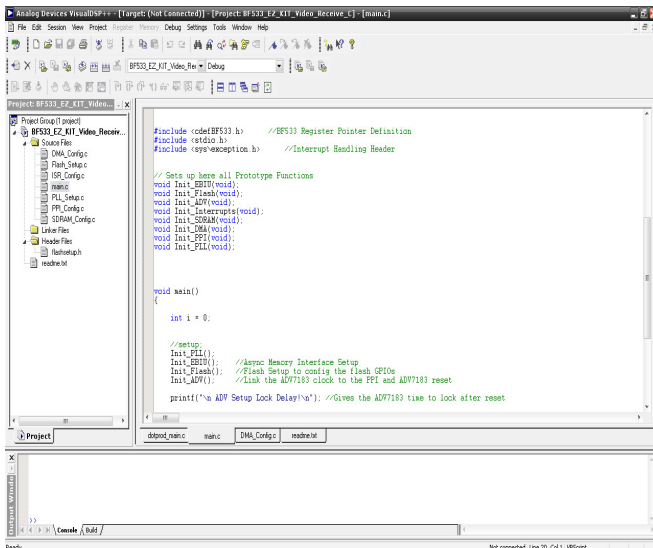


Figura 5. Entorno de desarrollo integrado *VisualDSP++5.0*.

Paso 2: Una vez abierto el proyecto, en el marco "Project", se selecciona la opción "Build Project", con lo cual el programa se carga automáticamente en el *DSP* (figura 6).

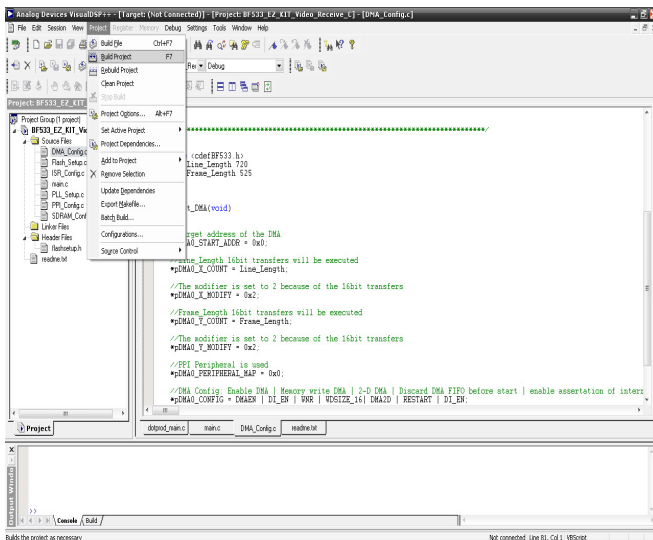


Figura 6. Carga automática del programa en el *DSP*.

Paso 3: En este punto es necesario cambiar la posición del terminal #6 del interruptor 3 (*SW3*), con el objetivo de determinar si la bandera programable (*PF2*) se encuentra conectada a la señal del decodificador *ADV7183*. Cuando el interruptor se encuentra apagado, *PF2* se puede utilizar para otras operaciones, y la salida del decodificador de habilitación se coloca en alto con una resistencia de *pull-up*. Así mismo, el interruptor *SW3* se encarga de controlar la formación de las señales de video en el codificador *ADV7171* y del decodificador *ADV7183*, que se dirigen al puerto paralelo *PPI*. El interruptor también determina si el terminal *PF2* de las banderas programables controla la señal de salida del decodificador de video *ADV7183*.

Paso 4: Se realiza la conexión de la cámara fotográfica al *DSP* a través del puerto de video analógico *NTSC/PAL*, en la entrada de video *AVINI* Jack, el cual se encuentra en la parte inferior derecha de la tarjeta *ADSP-BF533 EZ-KIT Lite*, tal como se muestra en la figura 7.

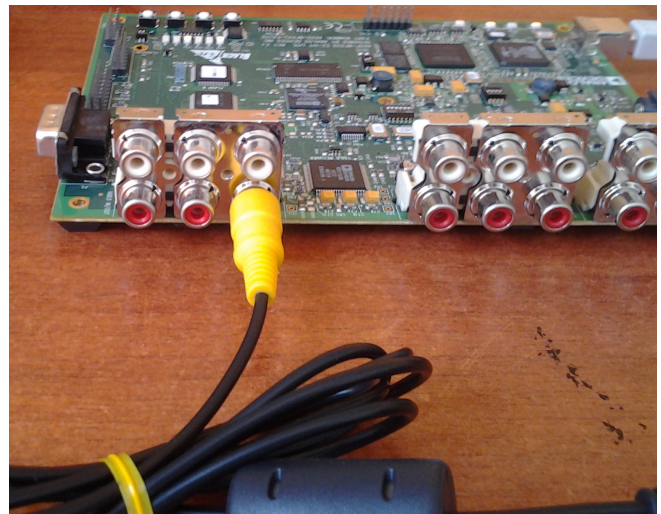


Figura 7. Conexión de cable RCA a la entrada *AVINI*.

Esta conexión permite la comunicación directa con el decodificador de video *ADV7183* del *DSP*, el cual convierte la señal analógica en una matriz de pixeles que es almacenada en la memoria de la *ADSP-BF533*. Una vez hecha realizada esta operación, y con el Software *VisualDSP++5.0* abierto, se realizará la ejecución por software para la adquisición de la imagen.

Paso 5. En este punto es necesario accionar el botón "Run" que se encuentra en el marco "Debug", pestaña ubicada en la barra del menú principal de *VisualDSP++5.0*. Este paso se ilustra claramente en la figura 8.

Cuando se ejecuta el código fuente, el procesador comienza a transformar cada una de las señales analógicas capturadas por la cámara en una matriz de pixeles a través de la *DMA*, almacenando cada uno de estos pixeles en la memoria interna *SDRAM*.

Paso 6: Para detener el procesador de la *ADSP-BF533*, es posible seleccionar la opción *“Halt”* que se encuentra en el marco *“Debug”*, pestaña ubicada en la barra del menú principal de *VisualDSP++5.0* (ver figura 9).

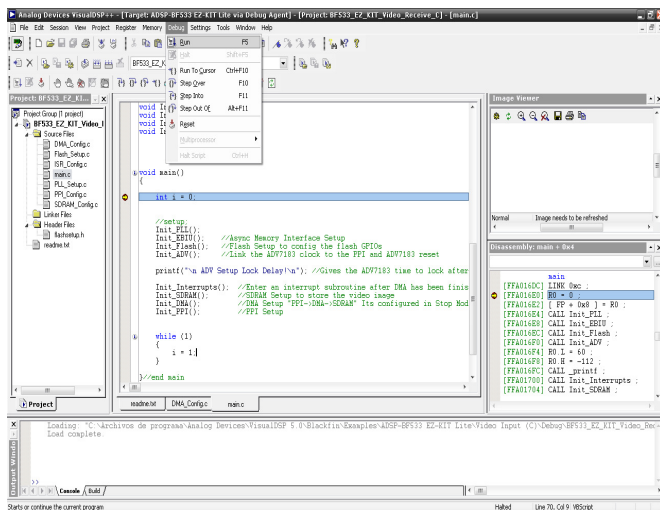


Figura 8. Procedimiento para la ejecución del programa principal.

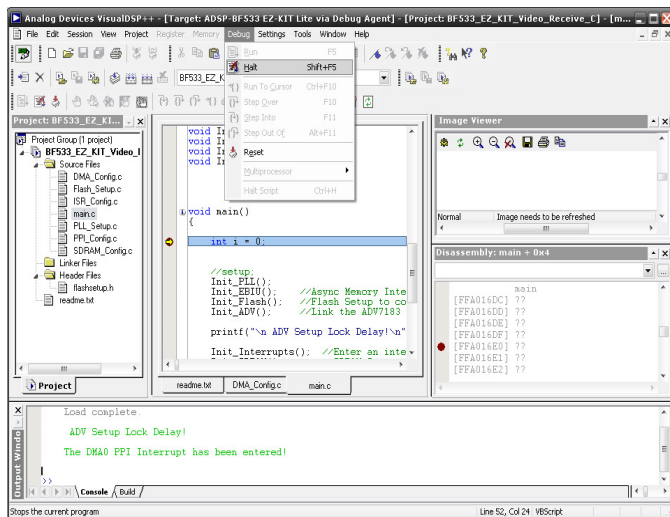


Figura 9. Botón de parada para detener el procesador.

Paso 7: Después de haber ejecutado de forma precisa cada una de las instrucciones anteriormente mencionadas, se procede a efectuar la visualización de la imagen que se encuentra digitalizada al interior del procesador, gracias al entorno de desarrollo de *VisualDSP++5.0*, el cual contiene la herramienta *“Image Viewer”*, de vital importancia para la implementación de aplicaciones de video. Esta herramienta se encuentra ubicada en el menú principal *“View/ Debug Windows/ Image Viewer”* (ver figura 10).

Esta herramienta facilita la visualización en el *PC* de la imagen que ingresa al decodificador de video *ADV7183*, ya que permite observarla tal como se vería en un televisor con entrada analógica de video.

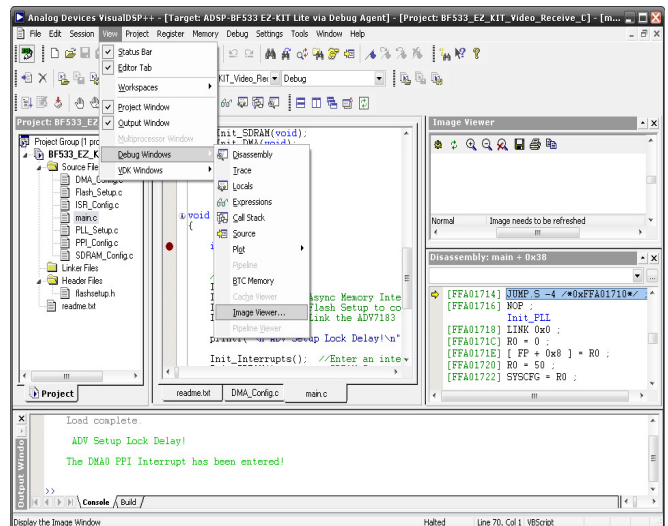


Figura 10. Visualización de la imagen capturada en el simulador.

Paso 8. Una vez accionado el botón *“Image Viewer”* descrito en el paso 7, se abrirá la ventana presentada en la figura 11. En esta ventana es necesario ajustar los siguientes parámetros [5], [6]:

- Start address: 0x1.
- Memory Stride: 2.
- Pixel Format: Gray Scale.
- Width (Pixels): 720.
- Height (Pixels): 525.

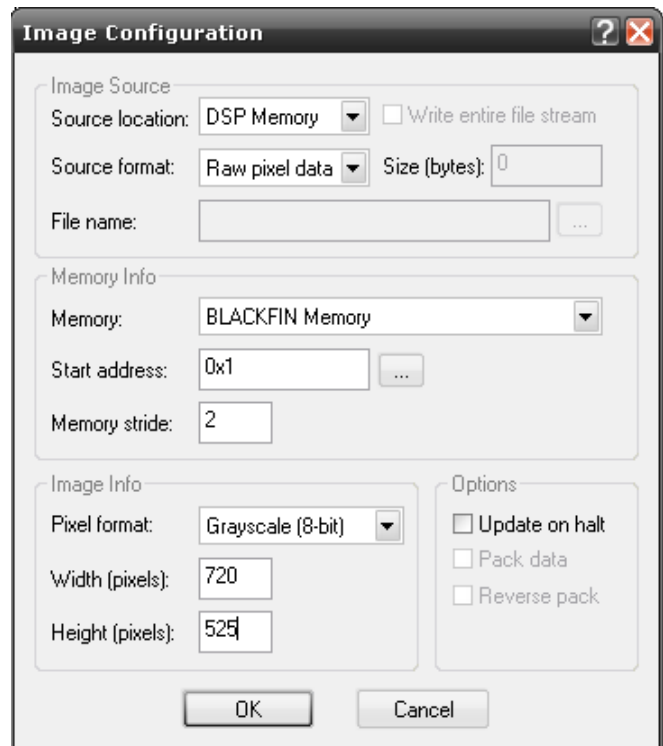


Figura 11. Ventana de configuración para la visualización de la imagen.

3. RESULTADOS

En la figura 12 es posible observar los resultados obtenidos al ejecutar el programa desarrollado en la herramienta de desarrollo *VisualDSP++*. Tal como se evidencia en la figura, el programa despliega la vista principal del visualizador de la cámara fotográfica, la cual es exportada al *DSP* a través del puerto analógico de video de conector *RCA*. Después de ser recibida por el decodificador de video, la imagen es transportada a la memoria interna de la tarjeta, quedando lista para un posterior procesamiento. Una vez allí, *VisualDSP++* accede a ella gracias a la conexión *USB* existente entre el *PC* y el procesador digital de señales.

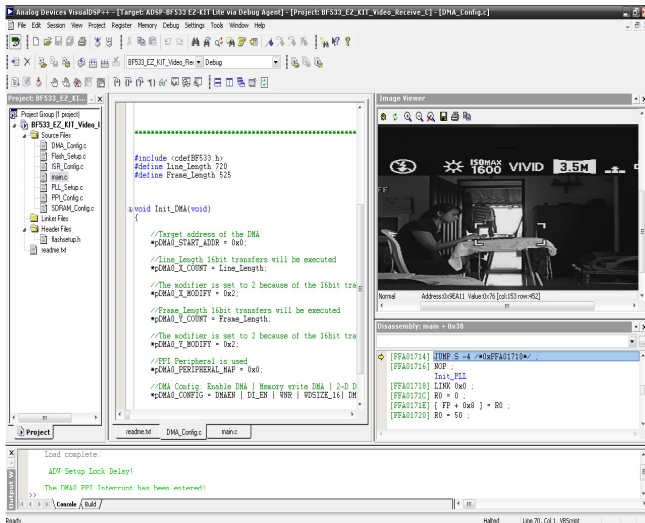


Figura 12. Resultado de la adquisición de la imagen.

Es necesario mencionar que originalmente el entorno de desarrollo del *DSP* presenta una imagen duplicada, debido a que los fotogramas son visualizados tal como son recibidos según el formato NTS, es decir, entrelazados (ver figura 13). Por esta razón, es necesario crear una rutina dentro del proyecto que permita desentrelazar las imágenes, con lo cual es posible obtener la captura presentada en la figura 12.



Figura 13. Imágenes entrelazadas según formato *NTSC*.

III. CONCLUSIONES

Gracias al desarrollo del presente proyecto es posible concluir que para optimizar el procesamiento de video en tiempo real sin pérdida de información, se hace necesario un conocimiento profundo del manejo de las estructuras de memoria interna del *DSP*, ya que se pueden presentar errores de latencia debido a la ausencia de sincronización. Una vez superada esta dificultad, la baraja de posibles aplicaciones del sistema de adquisición de imágenes crece de manera sustancial, ya que los fotogramas se encuentran listos para proceder a aplicar sobre ellos el procesamiento en tiempo real.

Es necesario mencionar que el tiempo de procesamiento del sistema desarrollado se estableció en 1.25 minutos aproximadamente. A primera vista, este tiempo se aleja bastante de lo esperado cuando se trata de sistemas de procesamiento en tiempo real. Sin embargo, es posible concluir que un altísimo porcentaje de este tiempo corresponde a la lectura de la imagen almacenada en las posiciones internas de memoria, la preparación para su transmisión al *PC* vía *USB*, y al procesamiento que realiza el software *Visual DSP++*, fundamental para la visualización del fotograma. Así pues, una vez sea innecesario proyectar la imagen en el *PC* debido probablemente a la urgencia de proceder con procesamientos posteriores dentro de la *DSP*, el sistema se vuelve supremamente rápido, lo cual concuerda con un principio básico de los sistemas embebidos, el cual introduce la necesidad de alejarse de los sistemas de cómputo tradicionales.

REFERENCIAS

- [1] C. Nagy, *Embedded Systems Design Using the TI MSP430 Series*, United States of America: Elsevier-Newton, 2003.
- [2] S. Pagani, *Procesamiento de Imágenes en Tiempo Real*, Facultad Regional Buenos Aires: Editorial Universidad Tecnológica Nacional, 2009.
- [3] ADSP-BF533 EZ-KIT Lite® *Evaluation System Manual, One Technology Way*, revision 3.1, September 2007. Part Number 82-000730-01. Analog Devices, Inc.
- [4] *ADSP-BF533 Blackfin Processor "Hardware Reference"*. Revision 3.4 Apr 2009. Analog Devices, Inc.
- [5] K. Singh, R. Babu, *VideoFramework Considerations for Image Processing on Blackfin Processors*. Revision 1 Sep 2005. Analog Devices Inc.
- [6] C. Lira. *Uma Solução para Captura, Compressão e Transmissão de Imagens Utilizando JPEG2000 em um Sistema Embarcado*, Pontificia Universidade Católica de Campinas. Campinas: 17 de dezembro de 2009.