

CARACTERIZACIÓN DE IMÁGENES DE BAJA RESOLUCIÓN EN MATRICES LED RGB. UN MODELADO ELECTRÓNICO BAJO HERRAMIENTAS LIBRES

Luis David Narváez ^{1*}, Stalin Marcelo Arciniegas ¹, Cristian Javier Cuaspa ¹

¹ Pontificia Universidad Católica del Ecuador Sede Ibarra, Escuela de Ingeniería, Ibarra-Ecuador

*Autor para correspondencia: e-mail: ldnarvaez@pucesi.edu.ec

Recibido: 2018/03/31

Aprobado: 2018/11/05

DOI: <https://doi.org/10.26621/XV19.2018.12.A07.PUCESI.2550.6684>

RESUMEN

El procesamiento de imágenes bidimensionales cromáticas de baja resolución se basa en principios de representación matemática de coordenadas en un plano cartesiano, donde se usan funciones con variables de posición (x e y) para modelar la intensidad de iluminación de cada pixel en base a la combinación de cada color primario: rojo, verde y azul (RGB, por sus siglas en inglés). Cabe mencionar que la resolución depende de manera directa de la cantidad de pixeles procesados, motivo por el cual para la ejecución de la presente investigación se han usado imágenes de prueba de baja resolución, con la finalidad de ser representadas en un tamaño de 64 x 64 pixeles.

En este sentido, el presente trabajo abarca en un principio, los procesos de conceptualización del modelo electrónico basado en un análisis bibliográfico de las plataformas de hardware libre, principalmente del sistema embebido Arduino y sus diferentes escudos o elementos complementarios, luego se establece el proceso de representación matemática de cada uno de los pixeles de las imágenes seleccionadas mediante las escalas de intensidad para cada color primario (rojo, verde y azul), mismas que luego son incorporadas en la codificación del sistema electrónico bajo Lenguaje C; finalmente se reproducen las imágenes en matrices Led RGB, donde cada pixel es un Led (64x64).

Palabras clave: Led RGB, Arduino, Lenguaje C, Código Abierto

ABSTRACT

The processing of two-dimensional chromatic images of low resolution is based on principles of mathematical representation of coordinates in a Cartesian plane, where functions with position variables (x and y) are used to model the illumination intensity of each pixel, based on the combination of each primary color: red, green and blue (RGB). It should be mentioned that the resolution depends directly on the number of pixels processed, which is why, for the execution of the present investigation, low resolution test images have been used in order to be represented in a size of 64 x 64 pixels.

In this sense, the present work covers initially the processes of conceptualization of the electronic model based on a bibliographic analysis of the free hardware platforms, mainly of the Arduino embedded system and its different shields or complementary elements, then the process of mathematical representation of each of the pixels of the images selected by the intensity scales for each primary color (Red, Green and Blue), which are then incorporated into the coding of the electronic system under Language C; Finally,

the images are reproduced in RGB Led arrays, where each pixel is a Led (64x64).

Keywords: RGB Led, Arduino, C Language, Open Source

INTRODUCCIÓN

El desarrollo de los sistemas embebidos de visualización de imágenes ha evolucionado en formas muy diversas, principalmente en el área de publicidad y propaganda con paneles electrónicos de grandes dimensiones, cuyo componente fundamental son Leds RGB y controladores programables de código fuente no libre o comúnmente llamados hardware y software privativo.

El limitante del acceso al código fuente de los controladores privativos ha motivado la presente investigación aplicada, con el afán de crear un sistema de visualización 2D con base a modelos matemáticos de procesamiento de imágenes bidimensionales, bajo el paradigma de libertad del código y el hardware, que por lo general es conocido como open source (Gibb, 2014).

En un inicio se establecieron los fundamentos teórico-científicos del procesamiento de imágenes bidimensional es con el uso de técnicas de investigación documental, para entender la actualidad de los sistemas de visualización, herramientas de desarrollo open source, programación de sistemas microelectrónicos y manejo de elementos de representación LED en formato RGB. Esta parte del proceso investigativo permitió determinar que en el mercado existen diferentes propuestas, marcas y tamaños; pero con un alto costo económico y de nulidad en cuanto al acceso al algoritmo y código fuente (Beltrán, 2011).

A continuación, se estableció un algoritmo con el afán de procesar las imágenes de baja resolución, condensado en un flujograma de procesos para luego ser codificado en Lenguaje C (Muñoz, 2012) bajo el IDE propio de Arduino (Pan y Zhu, 2018). En base al algoritmo se diseñó un diagrama de componentes del controlador electrónico. Como punto final se construyó el circuito electrónico que incluye solamente el aspecto de visualización de las imágenes en los paneles Led RGB (Yamanoor y Yamanoor, 2015), que es el objetivo principal de la investigación

FUNDAMENTACIÓN TÉCNICA

Visión humana y visión artificial

La visión del ser humano dispone de una extraordinaria capacidad sensorial que puede ser simulada por elementos electrónicos y computacionales actuales, pero no puede ser igualada por los equipos de visión artificial (Kremers, Baraas, y Marshall, 2016). La principal diferencia radica en la toma y procesamiento de las imágenes ya que el ser humano lo realiza de forma análoga y el mundo computacional en cambio es en forma digital. Esto repercute indudablemente en la calidad (resolución en píxeles) de las imágenes. (Tian, et. al. 2017)

Una imagen cromática por lo general puede ser representada por tres funciones esenciales de dos variables, $f_r(x,y)$, $f_g(x,y)$, $f_b(x,y)$, que expresarían la intensidad de iluminación de un punto coordinado (pixel). Estas funciones representarían los colores cromáticos primarios rojo (R), verde (G) y azul (B) (Ozcelik, et. al. 2017,). El tamaño de una imagen digital ronda por los 5 megapíxeles en promedio, que representa una cantidad importante de información a ser procesada. En la actualidad existen potentes equipos y sistemas para el procesamiento matemático de estas funciones, pero requieren un alto componente computacional lo que indudablemente representa un elevado costo económico, además de ser soluciones privativas cerradas que dificultan el desarrollo de la investigación en este campo.

Desde un punto de vista fisiológico, el ojo del ser humano es un sistema que recrea la realidad mediante el procesamiento de la información que recibe del entorno, en forma de haces de luz que se reflejan en todos

los objetos de la naturaleza. La luz al atravesar el cristalino del ojo se proyecta en la retina, formando una imagen invertida del objeto, luego esta es enviada al cerebro por medio del nervio óptico, donde es girada y presentada de forma correcta. Este proceso de recreación de la imagen de un objeto se aprecia en la figura 1. (Mesa y Rivero, 2015)

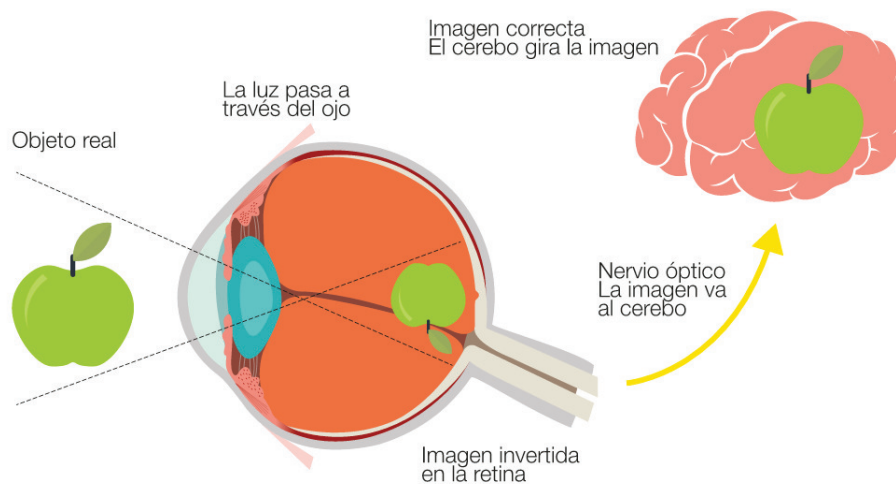
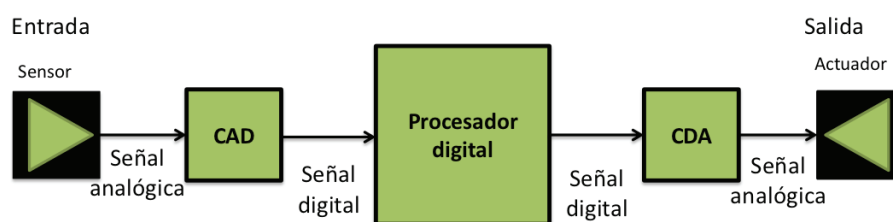


Figura 1. Funcionamiento del ojo humano

Fuente: Aurich, (2015).

Un controlador electrónico de procesamiento digital de señales, emula el comportamiento fisiológico de un sentido del ser humano; para este caso en específico, la recreación de las imágenes del entorno, son a través del manejo de señales transmitidas en forma de luz. En este sentido, un sistema de procesamiento permite capturar las señales de un sensor análogo que transforma la intensidad o color en niveles de voltaje para luego ser enviados al convertor análogo digital (CAD, por sus siglas en inglés); con los datos digitales la unidad central de proceso intenta recrear la imagen en base a su capacidad y la cantidad de información, luego del tratamiento son enviadas las señales de respuesta al convertor digital análogo (DAC, por sus siglas en inglés) y finalmente a los actuadores que en este caso es un sistema de visualización Led. El proceso artificial de procesamiento digital de la imagen de un objeto se aprecia en la figura 2. (Marković y Brodersen, 2012)



En el procesamiento digital de señales hay 3 etapas: la conversión a señal analógica digital (CAD), el procesamiento de la señal digital y la conversión de la señal procesada a una forma analógica (CDA).

Figura 2. Funcionamiento de un procesador digital de señales

Fuente: Urbina, (2012).

Sistema embebido Arduino

El sistema microelectronico Arduino es un conjunto de hardware y software libre, basado principalmente en microcontroladores ATMEL, disenado con el objetivo de facilitar la construccion de prototipos a nivel acadmico y profesional, ya que su presentacion fsica es una placa que requiere poco cableado para comunicarse con entradas (sensores) y salidas (actuadores). Dicha facilidad ha permitido que miles de proyectos e iniciativas alrededor del mundo se desarrollen, en campos de la educacion, la ciencia, la publicidad, la robtica, entre muchas ms (Russell, 2010).

Arduino como plataforma electronica de desarrollo es de muy bajo costo en comparacion con otros controladores programables como Parallax Basic Stamp, Phidgets, Raspberry pi, Microchip. Adems vale destacar la diversidad de modelos con mltiples capacidades y funcionalidades, por su puesto todo bajo un misma placa. Su entorno de programacion se basa en bibliotecas y libreras de lenguaje C y C++, denominado entorno integrado desarrollo Arduino (IDE, por sus siglas en ingls) (Purdum, 2012).

El licenciamiento de las guas de manejo del sistema Arduino es del tipo abierta, en especfico posee la licencia Creative Commons Attribution ShareAlike 3.0 license (CC-BY-SA), la cual establece la libertad de compartir la informacion en cualquier medio o formato, adems de la libertad de adaptacion del material para cualquier propsito inclusive de tipo comercial, caractersticas que son idneas para el desarrollo de la presente investigacion.

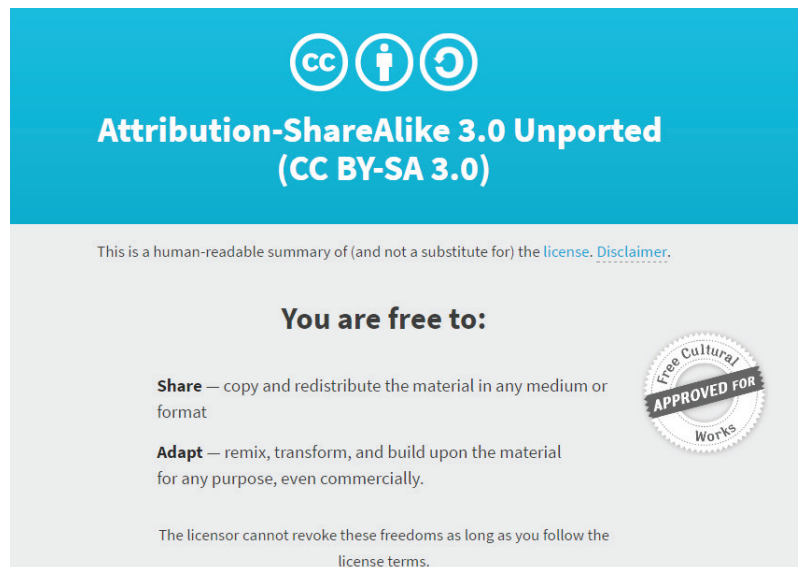


Figura 3. Creative Commons Attribution ShareAlike 3.0 license
Fuente: Creative Commons, (2014).

El desarrollo de la gran variedad de placas Arduino ha sido exponencial en los ltimos aos, cada version es creada con el objetivo de dotar funciones ms potentes a los procesos en que se integren hardware programable. En la categora de las placas Arduino USB, la version Arduino UNO es la ms actualizada en relacion a sus anteriores entregas, tales como: Arduino Duemilanove, Arduino Diecimila, entre otras. Dicho Arduino UNO es de tamao pequeo (12.5 cm x 5.3 cm), pero de gran potencial, capaz de usar sus pines como entrada o salida de seales, 14 pines para la entrada y salida de seales digitales, entre los cuales 6 pueden ser utilizados como salida PWM y 2 pines para el envo y transmision de datos en serie, 6 pines para la entrada de seal analgica, de manera que es fcil conectar una gran variedad de sensores y actuadores a sus pines para conectarse con el exterior. («Arduino - Home», 2018)

Acorde a lo descrito en la página oficial de Arduino, se presenta un resumen de las características de la placa Arduino UNO:

Tabla 1. Características Arduino Uno

Característica	Descripción
Microcontrolador	Atmega328
Tensión de operación	5V
Tensión de entrada(recomendada)	7 – 12 V
Pines digitales de E/S	14(de los cuales 6 proveen salidas PWM)
Pines de entrada analógicos	6
Corriente DC por pin E/S	40 mA
Corriente DC para pin	3.3 V 50 mA
Memoria Flash	32 kB (de los cuales 0,5 kB usados para bootloader)
SRAM	2 kB
EEPROM	1 kB
Frecuencia de reloj	16 MHz

Fuente: («Arduino - Home», 2018)

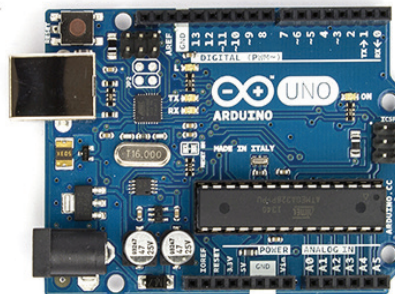


Figura 4. Placa Arduino UNO

Fuente: Banzi y Shiloh, (2014).

En igual sentido se analizó para el desarrollo de la investigación el modelo Arduino 101, esta placa contiene las prestaciones necesarias para ejecutar tareas que requieren un nivel alto de procesamiento con mayor rapidez.

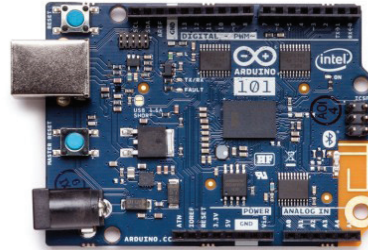


Figura 5. Placa Arduino UNO
Fuente: Arduino - Home, (2018)

Esta version de Arduino, mantiene el modelo del Arduino uno como se puede apreciar en la Figura 1, pero con mejores caractersticas de hardware las cuales se detallan en la Tabla 2.

Tabla 2. Especificaciones tecnicas de Arduino 101

Caracterstica	Descripcion
Microcontrolador	Intel Curie
Tension de funcionamiento	3.3V (E / S tolerante a 5V)
Voltaje de entrada (recomendado)	7-12V
Pines de E / S digitales	14 (de los cuales 4 proporcionan salida PWM)
Pines de E / S digitales de PWM	4
Clavijas de entrada analgica	6
Corriente DC por Pin E / S	20 mA
Memoria flash	196 kB
SRAM	24 kB
Velocidad de reloj	32 MHz
LED_BUILTIN	13
Caractersticas Especiales	Bluetooth LE, acelermetro / giroscopio de 6 ejes

Fuente: Arduino - Home, (2018)

Un modelo que igual se emple en la investigacion es el Arduino Mega, que es una placa que tiene todo lo necesario para soportar la programacion capaz de realizar muchas tareas por su cantidad de perifricos de entrada y salida, basado en microcontrolador Atmega1280, el cual se puede apreciar en la Figura 6.

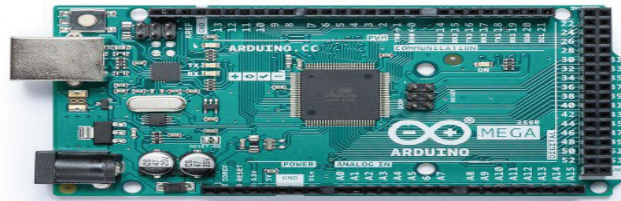


Figura 6. Placa Arduino MEGA
Fuente: («Arduino - Home», 2018)

Las especificaciones técnicas que posee Arduino Mega 2560 lo convierten en un dispositivo estable y robusto que soporta un uso prolongado en la ejecución de tareas que requieren un nivel moderado de procesamiento, dichas especificaciones se detallan en la Tabla 3.

Tabla 3: Especificaciones técnicas de Arduino Mega

Característica	Descripción
Microcontrolador	ATmega1280
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Pines de E / S digitales	54 (de los cuales 15 proporcionan salida de PWM)
Clavijas de entrada analógica	dieciséis
Corriente DC por Pin E / S	40 mA
Corriente DC para 3.3V Pin	50 mA
Memoria flash	128 KB de los cuales 4 KB utilizados por el bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Fuente:Arduino - Home, (2018)

¿Qué es un Led RGB?

Un Led es un diodo que emite luz cuando es electrificado por una corriente pequeña, por lo general compuesto por una serie de capas de materiales semiconductores. La eficiencia del Led se mide por la cantidad de electricidad que es convertida directamente en partículas de luz (fotones), que, en comparación con otras fuentes de luz, la mayor parte de la electricidad se convierte en calor y un porcentaje relativamente bajo en luz, ejemplo de ello son las lámparas incandescentes (Lasance y Poppe, 2013).

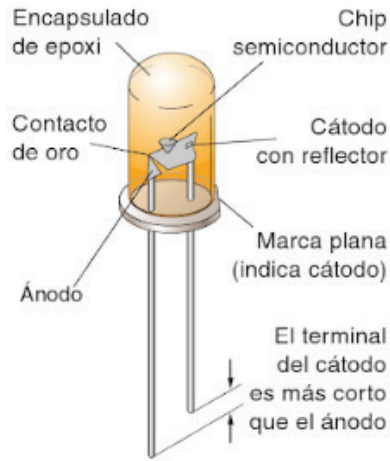


Figura 7. Elementos de un Led
Fuente: Montero, (2009)

Un Led RGB es la combinación de 3 diodos leds normales de colores rojo, verde y azul, comprimidos en un solo. La mezcla de estos tres colores primarios permite obtener cualquier color dentro del espectro electromagnético visible. Existen dos tipos, el ánodo común y cátodo común, donde la única diferencia es la forma de conexión de alimentación (Voltaje positivo) y tierra (Voltaje cero), por supuesto existe variedad de encapsulados, intensidad y tamaño.

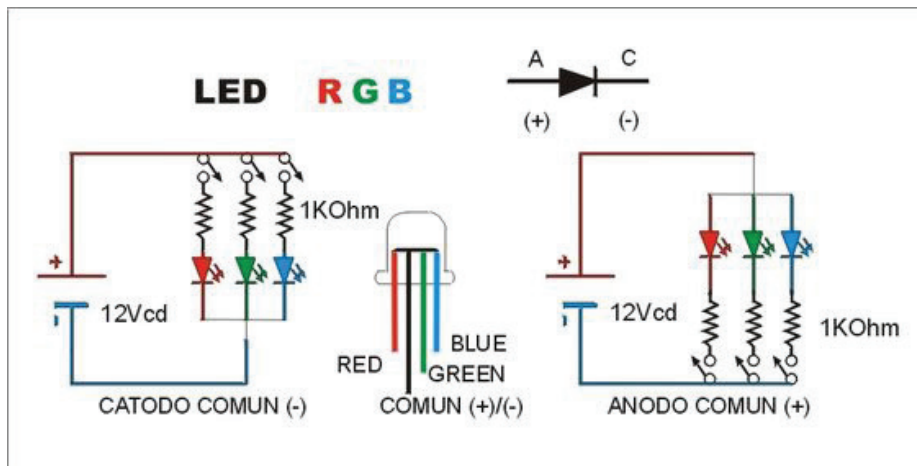


Figura 8. Conexión y esquema de un Led RGB
Fuente: Calderón M., (2017)

Matriz de Leds RGB

Una matriz de Leds RGB es un conjunto de Leds RGB dispuestos en filas y columnas a modo de puntos en un plano cartesiano, esta disposición es con el objetivo de minimizar el número de puntos de conexión y control, por lo que general estos paneles son manejados por técnicas de multiplexación en el tiempo con el uso de elementos electrónicos llamados registros de desplazamiento.

Es importante resaltar que a mayor número de Leds mayor es la resolución de las imágenes que se pueden representar. El modelo empleado para la presente investigación es el modelo 32x32 RGB LED MATRIX PANEL de 4 milímetros por pixel (1 Led equivale a 1 pixel), de la industria Adafruit. (Adafruit Industries, 2018)

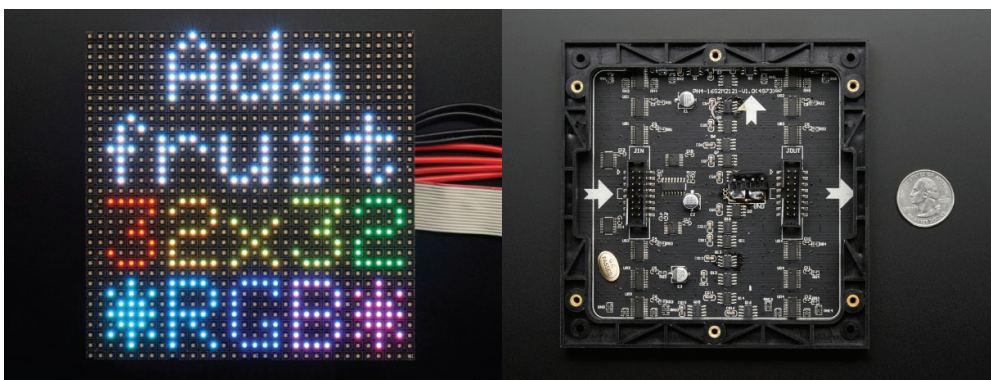


Figura 9. Matriz Led RGB 32x32

Fuente: Adafruit Industries, (2018)

Esta matriz ofrece una librería libre que hace posible su interconexión con el sistema Arduino, tiene un total de 1024 Leds RGB que significa un igual número de pixeles a ser representados. Un tamaño de soporte físico de 5 pulgadas que requiere de 13 pines digitales (6 bits para los datos y 7 bits para el control) y un suministro de energía eléctrica de 5 voltios y hasta 4 amperios de corriente continua. El espacio de memoria necesario que se debe disponer es de 1600 bytes de RAM para almacenar una imagen en color de 12 bits de resolución.

DISEÑO DEL CONTROLADOR ELECTRÓNICO

El objetivo principal de la presente investigación es la representación de imágenes de baja resolución en un panel Led RGB (Matriz de 32x32 pixeles), por lo que se ha diseñado un controlador que permite cumplir esta función, junto a otras funcionalidades para futuras aplicaciones e investigaciones. Los elementos que se han considerado son los siguientes:

- Almacenamiento y acceso a archivos: El controlador permite almacenar los archivos de imagen en un módulo memoria externa, SD por ejemplo, y provee los métodos para el acceso y lectura de la información de las imágenes.
- Administración del controlador: Esta parte del controlador es la comunicación con el usuario, ya que permite utilizar todas las funciones que da el controlador, es decir, la administración de archivos, procesar imágenes, y dar paso a la visualización.

- Visualización de imágenes: El controlador permite la transmisión de la información de las imágenes procesadas desde la unidad central hacia los periféricos de salida, que para este caso es la matriz Led RGB.

Estas características son consideradas como requerimientos funcionales del modelo electrónico programable, mismos que constituyen los componentes del sistema de visualización en matrices Led RGB, el detalle se muestra en la siguiente figura:

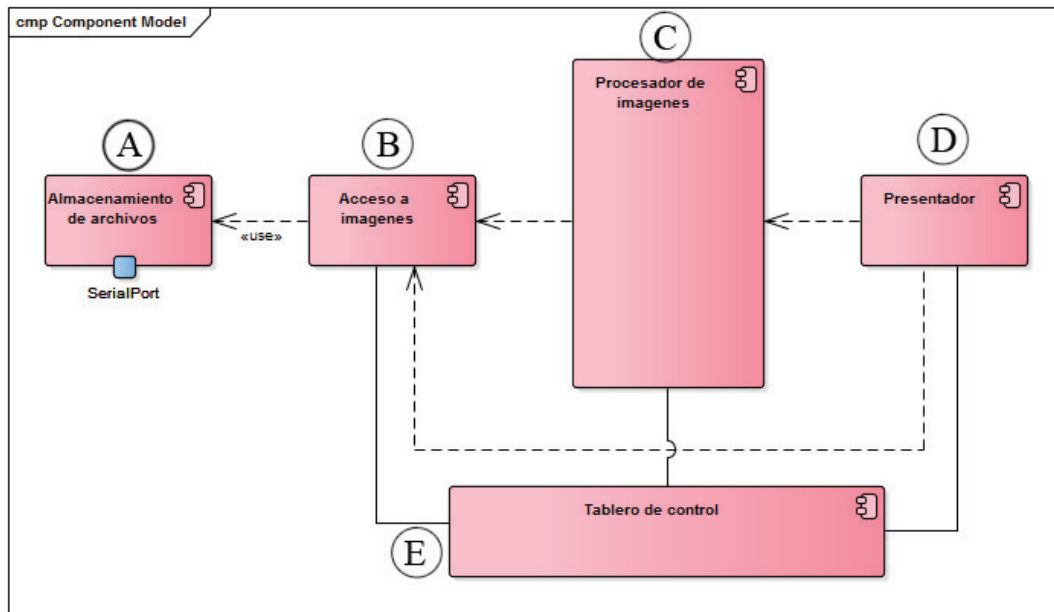


Figura 10. Diagrama de componentes

Fuente: Elaboración propia

El diseño propuesto contiene los siguientes componentes:

A. Almacenamiento de archivos: componente destinado a contener las imágenes de baja resolución que a posterior serán procesados por el controlador programable (sistema Arduino).

B. Acceso a imágenes: elemento que posibilita los mecanismos de comunicación con el sistema de archivos del componente de almacenamiento de archivos, para su posterior manipulación por el resto de componentes.

C. Procesador de imágenes: es el componente principal del controlador electrónico, su función principal es el procesamiento de cada uno de los píxeles de las imágenes almacenadas, para luego la información resultante ser enviada a los elementos de visualización (matriz 32x32), en un formato que permita la interpretación de un Led RGB por píxel, con la combinación precisa de cada nivel de intensidad de cada color primario.

D. Presentador: compuesto principalmente por la matriz de Led RGB y los diferentes componentes electrónicos que permiten la multiplexación en el tiempo (registros de desplazamiento) de las señales de control y datos provenientes del procesador central.

E. Tablero de control: componente que permite la administración de los demás elementos, principalmente en actividades de selección y apertura de los archivos de imágenes.

MODELO DE DESPLIEGUE

El modelo de despliegue del sistema de visualización Led RGB se basa principalmente en elementos de hardware y software con características open source, en este caso específico la plataforma Arduino, con el objetivo de brindar una alternativa a las opciones privativas existentes en el mercado.

En la siguiente figura se observa el modelo de despliegue propuesto:

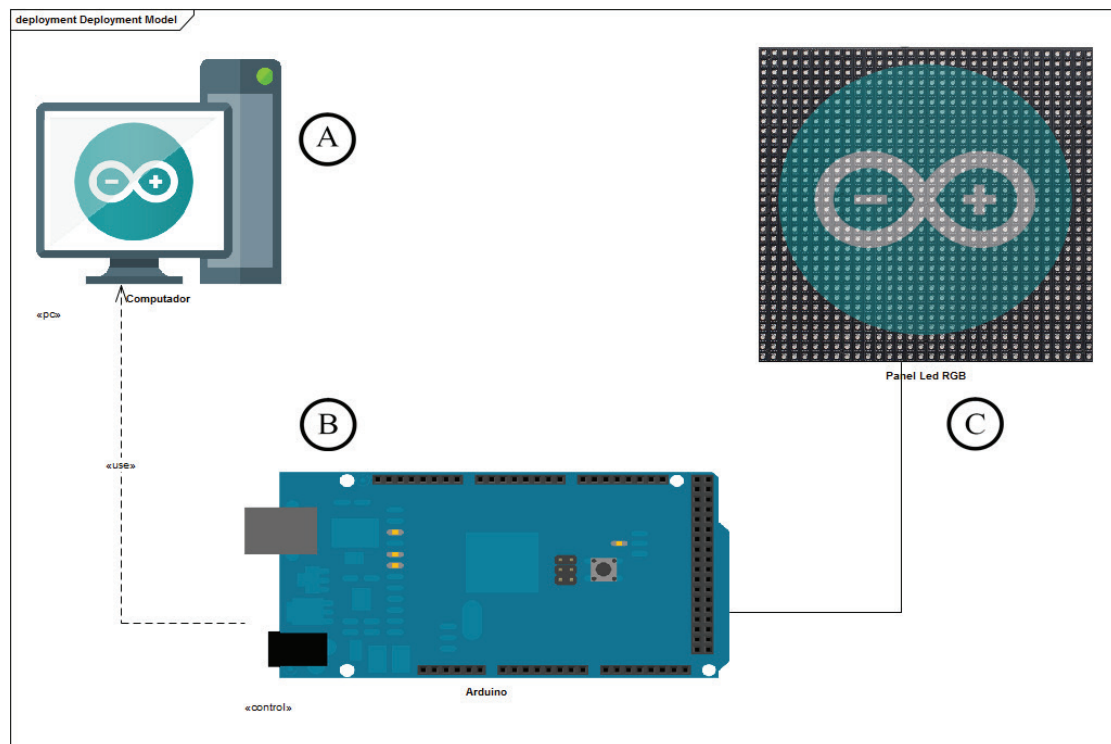


Figura 11. Modelo de despliegue

El detalle de los elementos del modelo de despliegue se los define de la siguiente forma:

A. Computador: permite la carga de imágenes al componente de almacenamiento y las actividades de administración del sistema.

B. Sistema Arduino: es el cerebro artificial de todo el modelo electrónico, donde se alojan las subrutinas de programación preconcebidas para la ejecución de tareas de procesamiento, apertura de imágenes y manejo de la comunicación con las matrices de visualización RGB.

C. Panel Led RGB: es elemento de salida de la representación de las imágenes de baja resolución procesadas, para este caso en específico es una matriz Led RGB con una densidad 32 x 32 pixeles.

ALGORITMO DEL PROCESO DE VISUALIZACIÓN

El algoritmo inicia con la carga de imágenes a través de la creación de una instancia que contenga toda la información sobre la imagen, para este caso en específico esto significa reservar un espacio en memoria

temporal para almacenar esta información. A paso seguido se procede a obtener la combinación RGB de cada píxel de la imagen, en otras palabras la intensidad de cada color primario (rojo, verde y azul).

El proceso en la plataforma Java NetBeans se muestra en la figura siguiente:

```
File archivo = new File("C:\\Users\\Alvaro\\Desktop\\ejemplo.png");
BufferedImage imagen = ImageIO.read(archivo);
int rgb = imagen.getRGB(0, 1); //always returns TYPE_INT_ARGB
int red = (rgb >> 16) & 0xFF;
int green = (rgb >> 8) & 0xFF;
int blue = (rgb) & 0xFF;
String salida = "R" + red + "G" + green + "B" + blue;
System.out.println(salida);
```

Figura 12. Extracto de código. Proceso de obtención de la combinación RGB

Fuente: Elaboración propia

El resultado del proceso contiene los valores de cada color en un formato de 8 bits en representación decimal. Para enviar esta información al controlador electrónico se debe escalar a la resolución límite de las funciones codificadas en el sistema Arduino, que por lo general son a 7, 4 y 3 bits de profundidad.

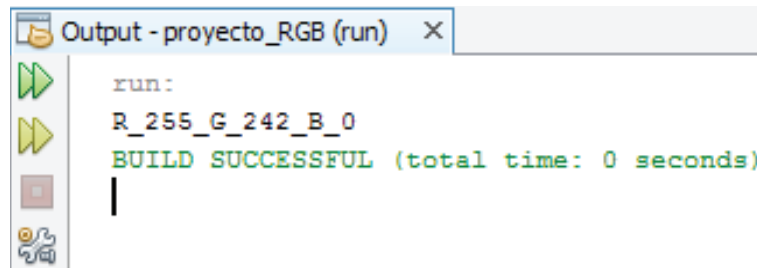


Figura 13. Extracto de código. Salida del proceso de obtención de la combinación RGB

Fuente: Elaboración propia

Finalmente, el proceso de visualización de una imagen de baja resolución en el panel RGB de 32 x 32 píxeles, se basa en el principio de multiplexación en el tiempo, donde se enciende línea por línea de píxeles (32 a la vez).

El formato de emisión de datos que se ha propuesto es de la siguiente forma: "1R2G3B4X5Y", en la cual 1 representa el valor de intensidad para el color rojo seguido del separador "R", 2 representa el valor de intensidad para el color verde seguido del separador "G", el 3 representa el valor de intensidad para el color azul seguido del separador "B", el 4 representa la coordenada horizontal seguido del separador "X", el 5 representa la coordenada vertical seguido del separador "Y".

Este proceso se muestra en el siguiente diagrama de flujo:

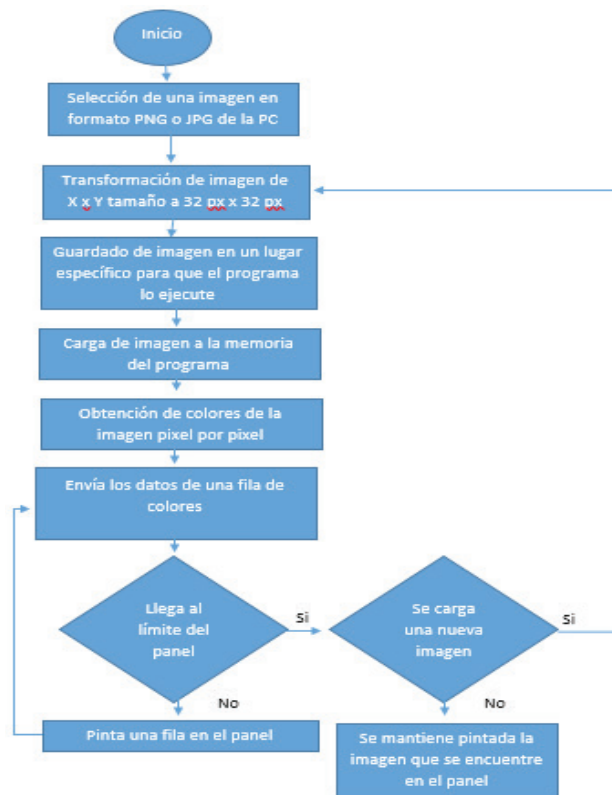


Figura 14. Flujograma del proceso de visualización

INTERFAZ DE CONTROL

Para la operatividad de los controles se creó una aplicación desarrollada en Python, con base a la secuencia de la función NewImage() que trabaja netamente con comunicación serial RS232, dicha interfaz se muestra en la figura 15.

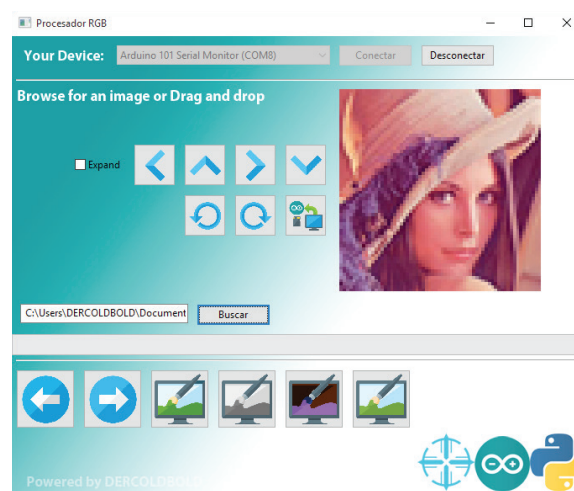


Figura 15. Flujograma del proceso de visualización

RESULTADOS

Los resultados de la presente investigación se engloban en la obtención del modelo y proceso para la representación de una imagen en baja resolución, tomando en cuenta que se manejó imágenes de 1024 pixeles (32 x 32), si esto se compara con una fotografía actual cuya resolución está en el orden los megapíxeles (millones de pixeles).

El principio de obtención de la intensidad de cada color primario por pixel se logró satisfactoriamente con una profundidad 8 bits, esto significa una escala de 0 a 255 por cada color, por supuesto esto es aún muy bajo si se compara con la profundidad con la que el ojo humano trabaja (1 millón de tonalidades) (Hadhazy A. 2015), pero es un inicio alentador para representar una gran gama de colores derivados.

En este sentido se diseñó un modelo electrónico funcional, que permite el control del procesamiento de imágenes en base la construcción una matriz bidimensional de pixeles; donde cada pixel contiene la información de la intensidad de cada color primario (rojo, verde y azul) además de la posición horizontal y vertical, la interacción alcanzada ente hardware y software se observa en la Figura 16.

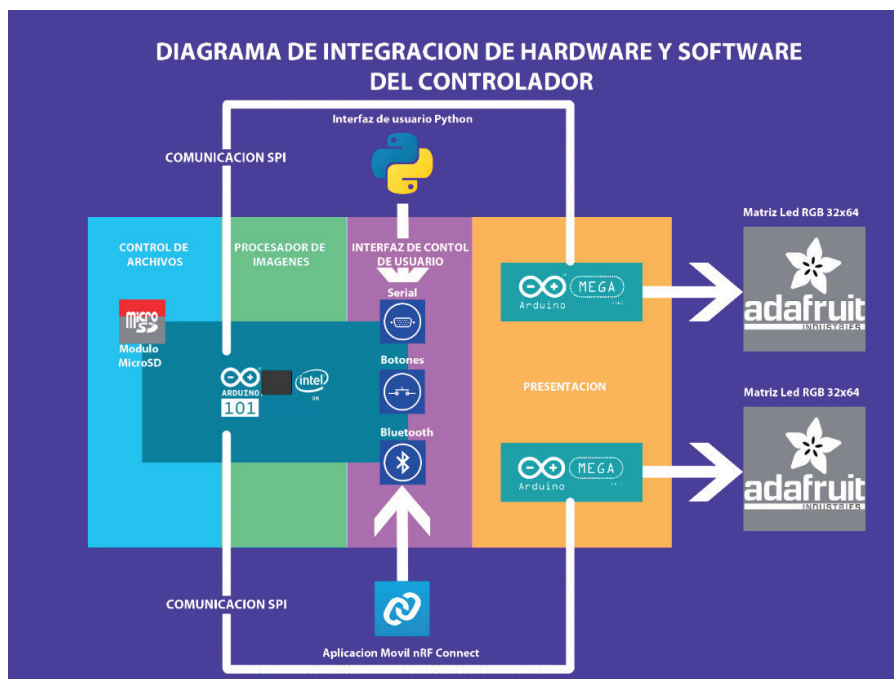


Figura 16. Interacción entre hardware y software.

El principio para la optimización de las conexiones físicas electrónicas fue la multiplexación en el tiempo para el barrido de encendido y apagado de los Leds, fila por fila de pixeles de la imagen a una frecuencia que el ojo del ser humano no detecte dicho encendido y apagado.

En las siguientes figuras se puede observar los resultados de la representación de imágenes de baja resolución tomadas como ejemplo.

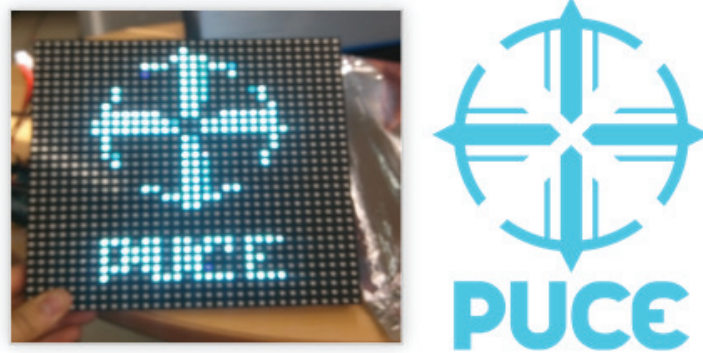


Figura 17. Representación del logo PUCE



Figura 18. Representación del logo Arduino

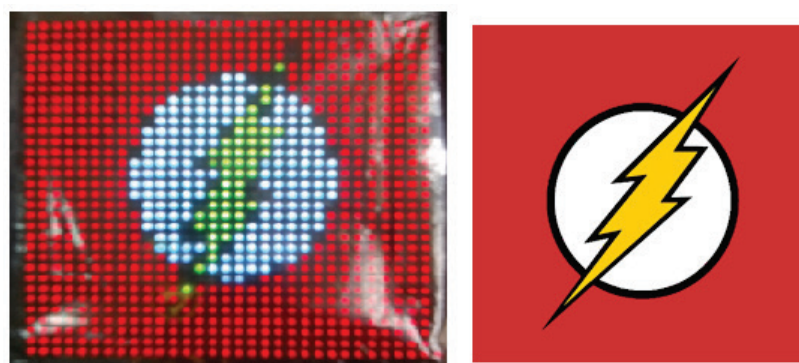


Figura 19. Representación del logo Flash

CONCLUSIONES

Una vez que se ha finalizado las etapas del modelado electrónico bajo herramientas libres del proceso de visualización de imágenes se pueden enunciar las siguientes conclusiones:

Las industrias y empresas invierten grandes cantidades de dinero o recursos para crear algoritmos, hardware y código fuente, que son insumos para crear aplicaciones de visualización a gran escala, estos desarrollos son protegidos con patentes y licencias de uso privativas. Esto motivó el desarrollo de la presente investigación, cuyo objetivo principal es dar el punto de partida para la construcción de una solución de visualización RGB bajo los paradigmas open source tanto en hardware como en software; alcanzando dentro de esto la construcción del código fuente bajo Lenguaje C y como plataforma de hardware Arduino, con la funcionalidad de ser un código reciclable.

La obtención de la intensidad de los colores primarios se logró a un nivel de profundidad de 8 bits, pero al momento de ser procesados en el sistema embebido Arduino lo máximo que se logró fue 7 bits, por lo que se pierde un rango importante de matices. En futuros desarrollos sería importante al menos llegar a una profundidad de 16 bits por color con la mejora de la codificación en lenguajes C o el uso de más atributos de las librerías que conforman la biblioteca OpenCV. (Chung B., 2017)

La elección de conjugar los dispositivos que intervienen en el controlador con comunicación SPI (de sus siglas en inglés, Serial Peripheral Interface) es debido a la eficiencia que brinda al comunicar con varios módulos de forma simultánea, sin impedir la transferencia de información al módulo de almacenamiento (Micro SD), como también para comunicarse con los sistemas Arduino maestros y esclavo.

Las imágenes de baja resolución ocupan menor poder computacional, debido a la menor cantidad de información que se debe procesar, para este caso en específico son 1024 píxeles con una profundidad de 8 bits por cada color primario (rojo, verde y azul), esto es por cada imagen que se desee representar en el panel de Leds. Esta conclusión significa que si en un futuro se desea recrear imágenes de mayor tamaño y profundidad de color se debe usar un procesador microelectrónico más avanzado (otra variante de Arduino), o en su defecto distribuir el procesamiento en redes de controladores.

REFERENCIAS BIBLIOGRÁFICAS

- Adafruit Industries. (2018). 32x32 RGB LED Matrix Panel - 4mm Pitch. Recuperado 28 de marzo de 2018, a partir de <https://www.adafruit.com/product/607>
- Arduino - Home. (2018). Recuperado 1 de abril de 2018, a partir de <https://www.arduino.cc/>
- Aurich, pixel smart A. fuer neue M. (2015, septiembre 6). • Ojos y visión | gafasyvision.com. Recuperado 30 de marzo de 2018, a partir de <https://www.gafasyvision.com/ojos-y-vision/>
- Banzi, M., y Shiloh, M. (2014). Getting Started with Arduino: The Open Source Electronics Prototyping Platform. Maker Media, Inc.
- Beltrán, F. F. (2011). El copyright en cuestión: Diálogos sobre la propiedad intelectual. Universidad de Deusto.
- Bimbo, A. (Ed.). (1997). Image Analysis and Processing: 9th International Conference, ICIAP'97, Florence, Italy, September 17-19, 1997, Proceedings, Volume II. Berlin Heidelberg: Springer-Verlag. Recuperado a partir de [//www.springer.com/us/book/9783540695868](http://www.springer.com/us/book/9783540695868)
- Calderón M. (2017). Pinterest. Recuperado 15 de marzo de 2018, a partir de <https://www.pinterest.com.mx/pin/540924605227198239/>
- Chung B. (2017). Pro Processing for Images and Computer Vision with OpenCV - Solutions for Media Artists and Creative Coders | Bryan WC Chung | Springer. Recuperado 1 de abril de 2018, a partir de <https://www.springer.com/us/book/9781484227749>
- Creative Commons — Attribution-ShareAlike 3.0 Unported — CC BY-SA 3.0. (s. f.). Recuperado 5 de marzo de 2018, a partir de <https://creativecommons.org/licenses/by-sa/3.0/>
- Gibb, A. (2014). Building Open Source Hardware: DIY Manufacturing for Hackers and Makers. Addison-Wesley Professional.
- Hadhazy, Adam. (2015). ¿Cuáles son los límites de la visión humana? Recuperado 31 de marzo de 2018, a partir de http://www.bbc.com/mundo/noticias/2015/08/150810_vert_fut_vision_limites_lp

- Kremers, J., Baraas, R. C., y Marshall, N. J. (Eds.). (2016). *Human Color Vision*. Springer International Publishing. Recuperado a partir de [//www.springer.com/la/book/9783319449760](http://www.springer.com/la/book/9783319449760)
- Lasance, C. J. M., y Poppe, A. (2013). *Thermal Management for LED Applications*. Springer Science and Business Media.
- Markovič, D., y Brodersen, R. W. (2012). *DSP Architecture Design Essentials*. Springer US. Recuperado a partir de [//www.springer.com/us/book/9781441996596](http://www.springer.com/us/book/9781441996596)
- Mesa, R. R., y Rivero, P. T. (2015). *Óptica para el cirujano faco-refractivo: Monografías SECOIR*. Elsevier España.
- Montero, A. C. (2009). *Electrónica*. Editex.
- Muñoz, V. J. E. (2012). *Aprendiendo a programar paso a paso con C*. Vicente Javier Eslava Muñoz.
- Ozcelik, Z., Tastimur, C., Karakose, M., y Akin, E. (2017). A vision based traffic light detection and recognition approach for intelligent vehicles. En *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 424-429). <https://doi.org/10.1109/UBMK.2017.8093430>
- Paco (2016). Coparoman: Led RGB. Recuperado a partir de <http://coparoman.blogspot.com/2016/07/led-rgb.html>
- Pan, T., y Zhu, Y. (2018). *Designing Embedded Systems with Arduino: A Fundamental Technology for Makers*. Springer Singapore. Recuperado a partir de [//www.springer.com/la/book/9789811044175](http://www.springer.com/la/book/9789811044175)
- Purdum, J. (2012). *Beginning C for Arduino: Learn C Programming for the Arduino*. Apress. Recuperado a partir de [//www.springer.com/us/book/9781430247760](http://www.springer.com/us/book/9781430247760)
- Russell, D. (2010). *Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment*. Morgan y Claypool. Recuperado a partir de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6813318>
- Tian, L., Li, M., Zhang, G., Zhao, J., y Chen, Y. Q. (2017). Robust human detection with super-pixel segmentation and random ferns classification using RGB-D camera. En *2017 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1542-1547). <https://doi.org/10.1109/ICME.2017.8019303>
- Urbina, C. (2012). Cecilia Urbina: Procesadores Digitales de Señales. Recuperado a partir de <http://cecilia-urbina.blogspot.com/2012/02/controladores-digitales-de-senales.html>
- Yamanoor, S., y Yamanoor, S. (2015). *Raspberry Pi Mechatronics Projects HOTSHOT*. Packt Publishing Ltd.