UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

CENTRO DE INVESTIGACIÓN EN INGENIERÍA Y CIENCIAS APLICADAS

"DISEÑO Y CONTROL ELECTRÓNICO DE UNA MATRIZ DE LEDS RGB PARA LA PROYECCIÓN DE IMÁGENES Y TEXTO ALFANUMÉRICO"

TESIS PARA OBTENER EL GRADO DE:
MAESTRÍA EN INGENIERÍA Y CIENCIAS APLICADAS

PRESENTA:
MARTIN ALBERTO VAZQUEZ CASTREJON

DIRECTOR: Dr. ÁLVARO ZAMUDIO LARA
CODIRECTOR: Dr. J. JESÚS ESCOBEDO ALATORRE
SINODALES: Dr. OMAR PALILLERO SANDOVAL
Dr. JOSÉ ANTONIO MARBÁN SALGADO
Dr. JOSÉ GUADALUPE VELASQUEZ AGUILAR

CUERNAVACA, MORELOS **OCTUBRE 2019**

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Instituto de Investigación en Ciencias Básicas y Aplicadas

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

Jefatura de Posgrado en Ingeniería y Ciencias Aplicadas

"1919-2019: en memoria del General Emiliano Zapata Salazar"

Cuernavaca, Morelos, a 24 de octubre de 2019.

DR. ROSENBERG JAVIER ROMERO DOMÍNGUEZ
COORDINADOR DEL POSGRADO EN
INGENIERÍA Y CIENCIAS APLICADAS
P R E S E N T E

Atendiendo a la solicitud para emitir DICTAMEN sobre la revisión de la TESIS titulada "Diseño y control electrónico de una matriz de leds RGB para la proyección de imágenes y texto alfanumérico" que presenta el alumno **Martín Alberto Vazquez Castrejon,** para obtener el título de **Maestría en Ingeniería y Ciencias Aplicadas**.

Nos permitimos informarle que nuestro voto es:

NOMBRE	DICTAMEN	FIRMA
DR. OMAR PALILLERO SANDOVAL	Aprobado	
DR. JOSÉ ANTONIO MARBÁN SALGADO	Aprobado	Jose Antonio MS.
DR. J. GUADALUPE VELÁSQUEZ AGUILAR	Aprobado	
DR. J JESÚS ESCOBEDO ALATORRE	Aprobado	
DR. ÁLVARO ZAMUDIO LARA	Aprobado	

PLAZO PARA LA REVISIÓN 20 DÍAS HÁBILES (A PARTIR DE LA FECHA DE RECEPCIÓN DEL DOCUMENTO)

NOTA. POR CUESTION DE REGLAMENTACIÓN LE SOLICITAMOS NO EXCEDER EL PLAZO SEÑALADO, DE LO CONTRARIO LE AGRADECEMOS SU ATENCIÓN Y NUESTRA INVITACIÓN SERÁ CANCELADA.

RJRD*RSU/nmc

RESUMEN

El uso de pantallas LED se ha extendido en los últimos años. Se destacan por mostrar información y publicidad, en comparación con los carteles publicitarios, las pantallas LED ofrecen un medio para desplegar información de una manera más rápida y fácil de sustituir. La reducción en los costos, el uso generalizado de los componentes y los más recientes avances en tecnología LED permiten la investigación de nuevas formas de construir y controlar pantallas basadas en módulos de matriz LED.

Este trabajo de tesis describe el diseño y construcción de una matriz de LEDs RGB para la proyección de imágenes y texto alfanumérico utilizando los LEDs inteligentes con driver WS2812B y un microcontrolador como interfaz de control. Se desarrolló un diseño de matriz de LEDs con 30 filas y 33 columnas, utilizando tiras de LEDs RGB WS2812B, papel cascaron y un cajón de madera. Se presenta la manufactura de una placa de desarrollo que incluye un microcontrolador de 32 bits para la interfaz de control de la matriz de LEDs. La lógica de los algoritmos desarrollados para la transmisión de datos y control de los pixeles de la matriz logran la proyección de imágenes y texto alfanumérico. Los resultados muestran nuevas formas de armar y construir matrices LED, también evidencian que se pueden controlar matrices LED sin la necesidad de aplicar técnicas de barrido, obtenido mejores niveles de brillo y color, pero en consecuencia, un mayor consumo de corriente.

ABSTRCAT

The use of LED screens has been extended in recent years. They stand out for displaying information and advertising, compared to advertising posters, LED screens offer a means to display information in a faster and easier way to replace. The reduction in costs, the widespread use of the components and the most recent advances in LED technology allow the investigation of new ways to build and control screens based on LED matrix modules.

This thesis paper describes the design and construction of a matrix array of RGB LEDs for the projection of images and alphanumeric text using smart LEDs with WS2812B driver and a microcontroller as a control interface. A matrix array of LEDs with 30 rows and 33 columns was developed, using WS2812B RGB LED strips, shell paper and a wooden drawer. The manufacture of a development board that includes a 32-bit microcontroller for the control interface of the LED array is presented. The logic of the algorithms developed for the transmission of data and control of the pixels of the matrix achieve the projection of images and alphanumeric text. The results show new ways of assembling and constructing LED matrices, they also show that LED matrices can be controlled without the need to apply scanning techniques, obtained better levels of brightness and color, but consequently, a greater current consumption.

AGRADECIMIENTOS

Doy gracias a Dios por darme la vida, la salud y llenarme de bendiciones todos los días. Gracias por permitirme concluir esta etapa de mi vida, por ser mi guía y darme fortaleza en los momentos difíciles.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado durante la realización de este proyecto.

A mis abuelos, Elizabeth y Rosalino, por siempre creer en mí, por inculcarme buenos valores y apoyarme incondicionalmente en cada etapa de mi vida. No cabe duda de que gracias a ustedes y a Dios he llegado hasta este punto.

A mi esposa, por estar a mi lado en los buenos y en los malos momentos, siempre teniendo las palabras correctas para hacerme sentir mejor. Gracias por creer en mí y todo el apoyo incondicional que me brindas. Siempre me animas a enfrentar nuevos retos. Te amo.

A mis padres y hermanas, por apoyarme y protegerme siempre que lo necesito.

A mi asesor, el doctor Álvaro Zamudio Lara por la confianza depositada.

Al doctor J. Jesús Escobedo Alatorre por los consejos brindados para la elaboración de este proyecto.

A mis sinodales, Dr. Omar Palillero, Dr. José Antonio Marban y Dr. José Guadalupe Velásquez, por su tiempo, sus consejos y sus conocimientos transmitidos.

Al Centro de investigación en ingeniería y ciencias aplicadas (CIICAp) y a la Universidad autónoma del estado de Morelos (UAEM) por permitirme realizar mis estudios de maestría en sus instalaciones.

ÍNDICE GENERAL

ÍNDICE GENERAL i
ÍNDICE DE TABLAS iii
ÍNDICE DE FIGURAS iv
CAPITULO 1 INTRODUCCIÓN 1
CAPITULO 2 MARCO TEÓRICO 7
2.1 MICROCONTROLADOR 7
2.2 CPU DEL MICROCONTROLADOR 7
2.2.1 UNIDAD DE CONTROL 8
2.2.2 UNIDAD ARITMETICA/LOGICA 8
2.2.3 REGISTROS 8
2.3 MEMORIAS EN MICROCONTROLADORES 9
2.3.1 MEMORIA DE PROGRAMA 9
2.3.2 MEMORIA DE DATOS 9
2.3.3 MEMORIA DE VARIABLES 9
2.4 PERIFERICOS DE LOS MICROCONTROLADORES 10
2.5 MICROCONTROLADORES PIC 10
2.6 PROGRAMACION DE MICROCONTROLADORES 11
2.7 SELECCION DE UN MICROCONTROLADOR 12
2.8 DIODO EMISOR DE LUZ 14
2.9 LED WS2812B 14
2.10 MATLAB 17
2.10.1 INTERFAZ GRAFICA DE USUARIO CON MATLAB 17
2.10.2 PROCESAMIENTO DIGITAL DE IMÁGENES EN MATLAB 19
CAPITULO 3 METODOLOGÍA 20
3.1 MANUFACTURA DE PLACA DE DESARROLLO 20
3.2 TIRA LED WS2812B 23
3.2.1 ALGORITMOS DE CONTROL DE TIRA LED WS2812B 24
3.3 MATRIZ DE LEDs WS2812B DE 8X5 PIXELES 27
3.3.1 PRUEBA DE MATRIZ DE 8X5 PIXELES 28
3.3.2 GRÁFICOS EN MATRIZ DE 8X5 PIXELES 28

3.3.3 DESPLAZAMIENTO DE CARACTERES EN MATRIZ LED 30
3.4 CONCTRUCCION DE LA MATRIZ DE LEDS RGB 34
3.5 PRUEBAS DE TEXTO EN MATRIZ LED DE 30X33 PIXELES 38
3.5.1 DESPLAZAMIENTO DE CARCATERES 38
3.5.2 CARACTERES ESTATICOS 38
3.6 MÓDULO UART DEL MICROCONTROLADOR 41
3.7 RECEPCIÓN DE CADENA DE CACRACTERES POR UART 43
3.8 IMÁGENES EN MATRIZ DE 30X33 PIXELES 45
3.8.1 INTERFAZ GRAFICA DE USUARIO 46
3.8.2 FUNCIONAMIENTO DE INTERFAZ GRÁFICA 48
CAPITULO 4 RESULTADOS 50
4.1 CONTROL DE TIRA LED WS2812B. 50
4.2 MATRIZ DE LEDs WS2812B DE 8x5 PIXELES. 51
4.2.1 GRÁFICOS 51
4.2.2 DESPLAZAMIENTO DE CARACTERES DE 8X5 PIXELES 52
4.3 MATRIZ DE LEDs WS2812B DE 30x33 PIXELES 53
4.3.1 DESPLAZAMIENTO DE CARACTERES 53
4.3.2 CARACTERES ESTATICOS 54
4.3.3 RECEPCIÓN DE TEXTO POR MÓDULO UART 55
4.4 INTERFAZ GRÁFICA PARA TRANSMISIÓN DE IMÁGENES 56
CAPITULO 5 CONCLUSIONES Y TRABAJO FUTURO 58
5.1 CONCLUSIONES 58
5.2 TRABAJO FUTURO 60
Bibliografía 61

ÍNDICE DE TABLAS

Tabla 1 Partes principales de un microcontrolador 7
Tabla 2 Componentes principales de una CPU 7
Tabla 3 Periféricos de los microcontroladores 10
Tabla 4 Características de los microcontroladores PIC 11
Tabla 5 Ventajes de los lenguajes de alto nivel 11
Tabla 6 Modelos y características de microcontroladores PIC 12
Tabla 7 Caracteristicas del LED WS2812B 14
Tabla 8 Tiempos de protocolo WS2812B 16
Tabla 9 Usos de MATLAB 17
Tabla 10 Lista de materiales para placa de desarrollo 22
Tabla 11 Materiales para matriz de 8x5 pixeles 27
Tabla 12 Materiales para construir una matriz de LEDs WS2812B de 30x33 pixeles 35

ÍNDICE DE FIGURAS

Fig. 1 Ejemplo de primeros visualizadores. 1
Fig. 2 Display de 7 segmentos..... 2
Fig. 3 Display de matriz de puntos LED. 3
Fig. 4 Publicidad y anuncios en pantalla LED. 4
Fig. 5 Módulo matriz LED 16x32. 5
Fig. 6 Tira de LEDs WS2812B 14
Fig. 7 LED WS2812B..... 15
Fig. 8 Cadena de datos para LED WS2812B 15
Fig. 9 Secuencias de tiempos para protocolo de transferencia del LED WS2812B. a) secuencia de tiempos para código 0, b) secuencia de tiempos para código 1, c) secuencia de tiempo para RESET (Treset≥50µS). 16
Fig. 10 Diseño electrónico de placa de desarrollo 20
Fig. 11 Diseño PCB de placa de desarrollo..... 21
Fig. 12 Placa de desarrollo manufacturada 21
Fig. 13 Placa de desarrollo armada 22
Fig. 14 PIC18F4550 y tira de LEDs WS2812B 23
Fig. 15 Diagrama de flujo de función NeoBit 24
Fig. 16 Diagrama de flujo de función NeoDraw 25
Fig. 17 Diagrama de flujo para control de tira de LEDs WS2812B 26
Fig. 18 Conexiones de tiras LED WS2812B 27
Fig. 19 Matriz de LEDs WS2812B de 8x5 pixeles..... 28
Fig. 20 Diagrama de flujo ara gráficos en matriz de 8x5 pixeles..... 29
Fig. 21 Tabla de valores binarios (8 filas, 5 columnas) 30
Fig. 22 Valor de 8 bits de una columna de la tabla binaria..... 31
Fig. 23 Carácter "#" de 8x5 pixeles..... 31
Fig. 24 Efecto de desplazamiento de caracteres 32
Fig. 25 Diagrama de flujo para algoritmo de desplazamiento de caracteres 33
Fig. 26 Tiras LED WS2812B 35
Fig. 27 Matriz de LEDs WS2812B de 30x33 pixeles..... 36
Fig. 28 Tiras de papel cascaron 36
Fig. 29 Reílla hecha con tiras de papel cascaron..... 37
Fig. 30 Cajón de madera con matriz de LEDs en su interior 38
Fig. 31 Texto en matriz de 30x33 pixeles 39
Fig. 32 Renglón con cadena de caracteres 39
Fig. 33 Diagrama de flujo para texto estático en matriz LED de 30x33 pixeles 40
Fig. 34 Conexión tipo DB9 de puerto serie 42
Fig. 35 Módulo conversor de USB a TTL ft232rl..... 43
Fig. 36 Diagrama de flujo para recepción de datos por módulo UART 44
Fig. 37 Entorno de desarrollo GUIDE 47
Fig. 38 Diseño de GUI para transmitir datos de imagen digital 47
Fig. 39 Características de la interfaz gráfica 47

Fig. 40 Diagrama de flujo para el funcionamiento de interfaz gráfica 48
Fig. 41 Diagrama de flujo para algoritmo de recepción de datos de imagen 49
Fig. 42 Tira LED WS2812B y PIC18F4550 50
Fig. 43 Tira LED WS2812B y placa de desarrollo 50
Fig. 44 Gráficos mostrados en matriz de LEDs WS2812B de 8x5 pixeles 51
Fig. 45 Caracteres tipo ASCII en matriz LED de 8x5 pixeles. 52
Fig. 46 Desplazamiento de caracteres en matriz de LEDs de 30x33 pixeles. 53
Fig. 47 Texto en matriz de LEDs de 30x33 pixeles 54
Fig. 48 Matriz de LEDs con recepción de datos por puerto serie. 55
Fig. 49 Interfaz gráfica para la transmisión de imágenes por puerto serie. 56
Fig. 50 Imágenes proyectadas en matriz de LEDs WS2812B de 30x33 pixeles. a) Logotipo CIICAp. b) Mandril. c) Lena. d) Árabe. 57

CAPITULO 1
INTRODUCCIÓN

En vez de los ya tradicionales letreros de madera o de metal, la bombilla hizo posible por primera vez el diseño de letreros y formas iluminadas. La bombilla incandescente fue reemplazada en gran medida por los tubos fluorescentes, que iluminaban más y a los que más adelante también se les podría dar forma. Desde ese momento, la tecnología de luz como la publicidad luminosa se han desarrollado continuamente. En un principio, los tubos fluorescentes fueron cada vez más eficientes y pequeños, siendo finalmente desplazados por los diodos emisores de luz (LED) (Susan Walsh Sanderson, 2014).

Se le llama visualizador (display) a un tipo de dispositivo de salida que permite mostrar información de manera visual. Los primeros visualizadores consistían en una bombilla incandescente iluminando por la parte trasera a letras o números. Con el avance tecnológico, los displays mejoraron considerablemente en lo que conocemos como display de 7 segmentos y el display de matriz de puntos LED.

Fig. 1 Ejemplo de primeros visualizadores.

Uno de los visualizadores más importantes hoy en día es el display de 7 segmentos. El display de 7 segmentos es un componente que se utiliza para representar caracteres (normalmente números) en muchos dispositivos electrónicos. Internamente están constituidos por una serie de LEDs con determinadas conexiones y estratégicamente colocados de tal manera que se forma un “8”. Hoy en día, este tipo de visualizadores parecen obsoletos debido al uso de nuevas tecnologías como los displays de matrices de punto o pantallas lcd, aun así el display de 7 segmentos sigue siendo una excelente opción debido a su potencia lumínica, facilidad de implementación, bajo costo y robustez.

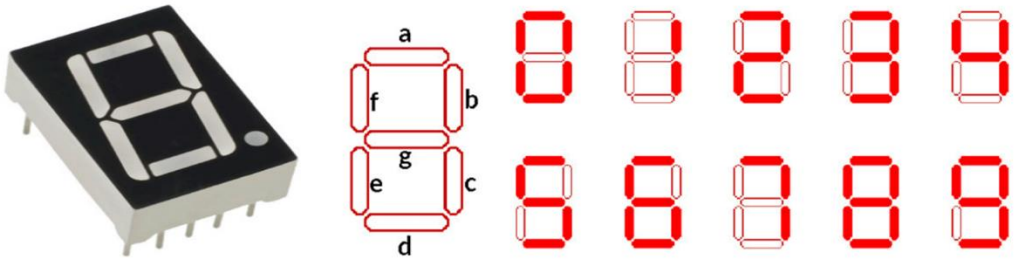

Fig. 2 Display de 7 segmentos

Una matriz de puntos es una técnica que se utiliza para representar caracteres, símbolos e imágenes en un patrón sobre una matriz. Una de las aplicaciones de la técnica de matriz de puntos se puede encontrar en el visualizador de matriz de puntos.

El visualizador de matriz es un dispositivo utilizado para mostrar información en aplicaciones industriales o comerciales, así como para las interfaces hombre-máquina. Consiste en una matriz de puntos luminosos (LEDs) con sus cátodos unidos en columnas y sus ánodos en filas (o viceversa). Controlando el flujo de corriente a través de cada par fila-columna, es posible controlar cada LED de forma individual. Utilizando la técnica de multiplexación, se pueden crear caracteres o imágenes que muestren la información al usuario.

La frecuencia con la que los LEDs son encendidos debe ser lo suficientemente rápida para evitar que el ojo humano detecte el parpadeo (Wahyono, 2015).

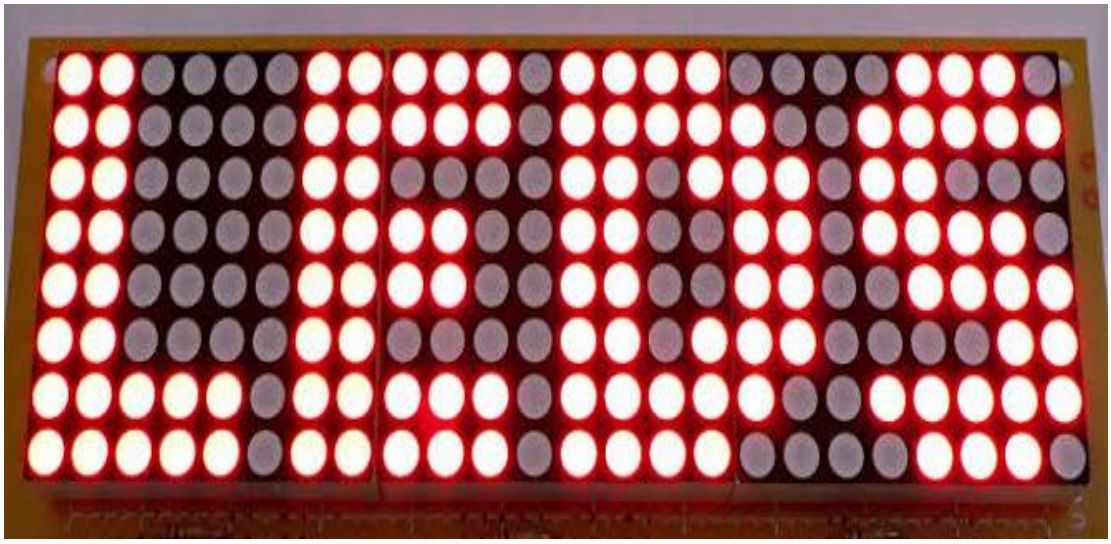

Fig. 3 Display de matriz de puntos LED.

Una pantalla LED es un dispositivo electrónico que se caracteriza por estar conformado por diodos emisores de luz (LED), y que puede desplegar datos, información, imágenes, vídeos, etcétera (Ahn, 2013).

Las pantallas LED se componen de módulos o paneles LED basados en los displays de matriz de puntos, ya sean monocromáticos (un solo color), bicolor (dos colores) o policromáticos. Estos últimos se conforman a su vez por LEDs RGB (rojo, verde y azul, los colores primarios de la platea de colores para pantallas). Los LEDs forman pixeles, lo que permite formar caracteres, textos, imágenes y hasta vídeo, dependiendo de la complejidad de la pantalla y el dispositivo de control (Kurdthongmee, 2005).

En pantallas LED con pixeles RGB se obtiene una paleta de colores a través de la mezcla o combinación de los tres colores primarios, y así poder obtener un despliegue de millones de colores como al que puede llegar un monitor convencional (Zalesinska, 2018).

Las pantallas LED se están volviendo cada vez más comunes a medida que sus precios van cayendo significativamente en los últimos años. Con el uso generalizado de componentes baratos, entusiastas de la electrónica han intentado crear versiones baratas de los costosos productos comerciales (Petr Olivka, 2015).

Fig. 4 Publicidad y anuncios en pantalla LED.

Los LEDs son dispositivos manejados por corriente. Manejar varios LEDs individualmente es relativamente simple. Sin embargo, cuando el número de LEDs incrementa, los recursos que se necesitan para operarlos resultan bastante extensos. Un método utilizado para manejar matrices de LEDs es el multiplexado (AVAGO, 2013).

La multiplexación es una técnica empleada para operar matrices de LEDs. Cada LED puede ser controlado individualmente en modo multiplexado. Esto se lo hace dividiendo la secuencia de manejo del LED en niveles en el dominio del tiempo. Por multiplexación, solo una fila de la matriz de LEDs es activada

en un intervalo de tiempo. Este método se aplica porque un terminal del LED (sea el ánodo o el cátodo) está unido a una sola fila.

La multiplexación por división de tiempo se la realiza en orden secuencial. Solamente una fila es energizada en un único instante de tiempo. Durante el período en el cual una fila es energizada, los LEDs deseados son encendidos, energizando las columnas apropiadas y respectivas de acuerdo con los datos que se desean mostrar. Este proceso es conocido también como barrido.

El mercado de las pantallas modulares de matriz LED está dominado por el modelo de la Fig. 5. Este tipo de matrices LED requieren de 12 pines digitales (6 pines de datos y 6 pines de control), además de 2 cables más para la alimentación, a esta interfaz de control se le conoce como: HUB-75.

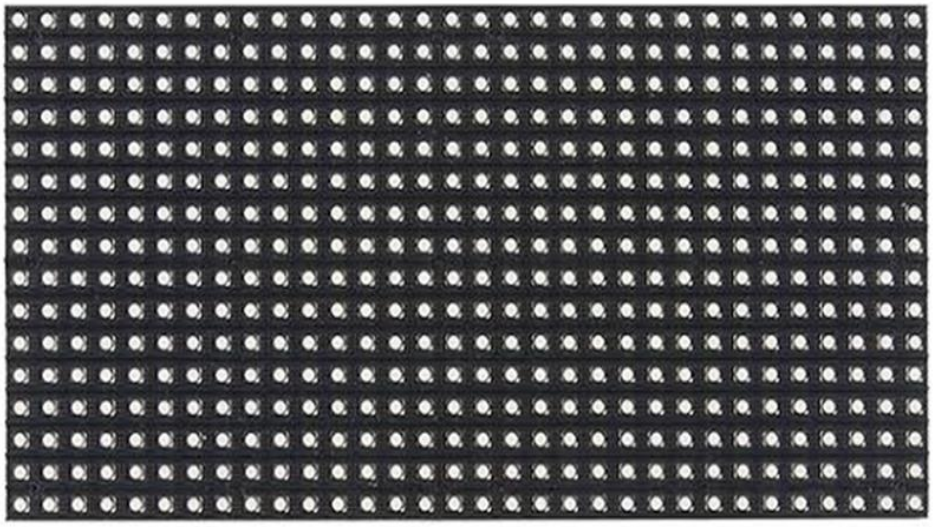

Fig. 5 Módulo matriz LED 16x32.

La cantidad de pines necesarios para controlar los módulos de matriz LED es bastante ineficiente, aparte de que también se tiene que ocupar la técnica de la multiplexación. Por esta razón, ha aumentado la investigación para encontrar nuevas formas de controlar matrices LED y también nuevas formas de construir matrices LED.

La reciente aparición de la tecnología de los LEDs inteligentes permite innovar en la construcción de matrices LED, y también en la investigación y creación de controladores. Los LEDs inteligentes incorporan protocolos de transferencia de datos de manera serial, posibilitando el control de múltiples LEDs conectados en serie a través de un solo cable de datos.

OBJETIVO

El objetivo de este proyecto es construir una matriz de LEDs RGB con el fin de proyectar imágenes y texto alfanumérico utilizando un microcontrolador como interfaz de control.

ORGANIZACIÓN DE LA TESIS

Este trabajo se compone de 5 capítulos bajo la siguiente estructura:

- Capítulo 1: Se presenta la introducción y el objetivo de la tesis.
- Capítulo 2: Se describe el marco teórico, se detallan los conceptos básicos de los elementos necesarios para construir y controlar una matriz de LEDs, también se presenta la selección de los componentes como son el microcontrolador y el modelo de los LEDs.
- Capítulo 3: Se muestra la metodología utilizada en el proyecto, se describe la lógica de los algoritmos de control de LEDs WS2812B, se presentan las acciones realizadas para armar una matriz de LEDs y las pruebas realizadas.
- Capítulo 4: Se describen los resultados de las pruebas realizadas a la matriz de LEDs.
- Capítulo 5: Se presentan las conclusiones del proyecto y se discuten brevemente algunas ideas para determinar posibles mejoras y trabajo futuro.

CAPITULO 2

MARCO TEÓRICO

2.1 MICROCONTROLADOR

El microcontrolador es un circuito integrado programable que contiene todos los componentes necesarios para controlar el funcionamiento de una tarea determinada y es capaz de ejecutar las ordenes grabadas en su memoria (Palacios, Remiro, & López, 2004). Estos dispositivos son de muy bajo coste y se pueden utilizar para aplicaciones de control digital que van desde hacer una sencilla secuencia de luces hasta el control total de un proceso de una planta industrial (Ibrahim, 2006).

Los microcontroladores se componen principalmente por los siguientes elementos:

Tabla 1 Partes principales de un microcontrolador

1	Unidad de procesamiento central (CPU).
2	Memoria.
3	Periféricos.

2.2 CPU DEL MICROCONTROLADOR

Es el dispositivo que se encarga de procesar la información y de interpretar las instrucciones que se encuentran en la memoria de programa mediante operaciones aritméticas y lógicas. En la CPU destacan tres componentes principales (Jesús Javier Rodriguez Sala, 2003):

Tabla 2 Componentes principales de una CPU

1	Unidad de control.
2	Unidad aritmética/lógica (ALU).
3	Registros.

2.2.1 UNIDAD DE CONTROL

Es uno de los componentes más importantes, debido a que sin ella simplemente no se podría ejecutar e inclusive ni siquiera decodificar las instrucciones programadas para que trabaje la CPU, y por ende el microcontrolador completo con cada uno de sus componentes. La unidad de control determina parámetros como el conjunto de instrucciones, la velocidad de ejecución, el tiempo de ciclo máquina, los tipos de buses, las interrupciones y su configuración general.

2.2.2 UNIDAD ARITMETICA/LOGICA

También conocida por el nombre de coprocesador o coprocesador matemático, es la encargada las operaciones lógicas (booleanas) y matemáticas (sumas y restas) que se requieran, no obstante, cabe mencionar que está enfocada hacia operaciones básicas, en el caso de operaciones más complejas se generan ciclos de operaciones sencillas utilizando mayor cantidad de recursos. En el caso de los microcontroladores más actuales se tiene un coprocesador de mayor potencia para la ejecución tanto de operaciones como de instrucciones de mayor peso.

2.2.3 REGISTROS

Los registros son las memorias principales de la CPU las cuales tienen un espacio del mismo tamaño que los buses del microcontrolador y pueden funcionar a la misma velocidad que la CPU. Los registros los podemos encontrar desde los más pequeños de 4 bits hasta los más grandes de 64 bits, estos son los que determinan el tamaño de las operaciones a ejecutar.

2.3 MEMORIAS EN MICROCONTROLADORES

Las memorias son utilizadas para almacenar información dentro de cualquier sistema que así lo necesite. Para el caso de los microcontroladores, almacena instrucciones, direcciones, datos y estados, así como resultados de las operaciones que realiza la CPU.

En los microcontroladores, la memoria está dividida en tres tipos diferentes: la memoria de programa, la memoria de datos y la memoria de variables.

2.3.1 MEMORIA DE PROGRAMA

La memoria de programa es básicamente de solo lectura de tipo no volátil, en ella se almacenan todas las instrucciones del programa de control, como este siempre es el mismo, debe de estar grabado de forma permanente. Generalmente, los microcontroladores usan memorias de tipo FLASH para la memoria de programa. La memoria de programa se puede leer desde el programa principal que se está ejecutando.

2.3.2 MEMORIA DE DATOS

La memoria de datos es una memoria de tipo no volátil que permite lectura y escritura de datos. Generalmente, la memoria de datos en los microcontroladores una memoria EEPROM, sirve para almacenar datos de suma importancia mismos que podrán ser retenidos en el sistema por unos 50 años aproximadamente.

2.3.3 MEMORIA DE VARIABLES

La memoria de variables está destinada a guardar los valores de las variables que se están ejecutando en el programa principal. La memoria de variables es de tipo volátil y que permite la lectura y escritura de datos. Generalmente, los microcontroladores usan una memora de tipo SRAM (Static RAM) para evitar la necesidad de estar refrescando los datos cada cierto tiempo.

2.4 PERIFERICOS DE LOS MICROCONTROLADORES

Dentro de la organización básica de los microcontroladores se ubican un conjunto de periféricos. Los periféricos son componentes internos de un microcontrolador, auxiliares e independientes pero que pueden interactuar con el procesador directamente y con las líneas de entrada y salida del microcontrolador. Los periféricos más importantes de los microcontroladores son los siguientes:

Tabla 3 Periféricos de los microcontroladores

1	Puertos de entrada y salida de propósito general.
2	Temporizadores y contadores.
3	Conversor Analógico/Digital.
4	Puertos de comunicación.
5	USART.
6	Comparadores.
7	Modulación por ancho de pulso.

2.5 MICROCONTROLADORES PIC

Los PICs son una familia de microcontroladores que cuentan con un procesador tipo RISC y arquitectura HARVARD, son fabricados por Microchip Tecchnology Inc. El nombre completo de estos microcontroladores es PICmicro, aunque generalmente se utiliza el termino PIC (Peripheral Interface Controller).

Los microcontroladores PIC han tenido un gran éxito en los últimos años debido a (Enrique Mandado Pérez, 2007):

Tabla 4 Características de los microcontroladores PIC

1	Su buena relación precio/prestaciones.
2	La facilidad para desarrollar aplicaciones, debido a su reducido repertorio de instrucciones.
3	Su facilidad de reprogramación, ya que cuentan con memoria no volátil de tipo FLASH, que es borrable y programable eléctricamente.
4	Documentación y herramientas de programación de libre distribución, muchas de las cuales son proporcionadas directamente por Microchip.
5	Su disponibilidad como circuitos normalizados y la existencia de numerosos desarrolladores de sistemas que lo utilizan.

2.6 PROGRAMACION DE MICROCONTROLADORES

Tradicionalmente, los microcontroladores se programan utilizando el lenguaje ensamblador con el propio repertorio de instrucciones del microcontrolador. Como resultado, el lenguaje ensamblador de los microcontroladores fabricados por diferentes empresas es distinto e incompatible, por lo que el usuario tiene que aprender nuevas instrucciones en lenguaje ensamblador antes de programar un nuevo microcontrolador. Hoy en día, los microcontroladores se pueden programar utilizando lenguajes de alto nivel como BASIC, PASCAL, FORTRAN o C (Ibrahim, 2006).

Los lenguajes de alto nivel ofrecen varias ventajas en comparación al lenguaje ensamblado, por ejemplo:

Tabla 5 Ventajas de los lenguajes de alto nivel

1	Código más sencillo y comprensible.
2	El algoritmo puede funcionar para distintos microcontroladores.
3	Permite crear algoritmos complejos con pocas líneas de código.

La mayor desventaja del lenguaje de alto nivel es que los programas se ejecutan más lento que los hechos en lenguaje ensamblador debido a que el lenguaje de alto nivel genera más líneas de código.

2.7 SELECCION DE UN MICROCONTROLADOR

Para seleccionar un microcontrolador se consideraron los siguientes puntos: conocimientos previos, fabricante, tipo de procesador, velocidad de procesador, MIPS (millones de instrucciones por segundo), tamaño de memoria interna y lenguaje de programación compatible.

Se decidió trabajar con los microcontroladores PIC de Microchip, ya que disponen de una gran variedad de modelos, módulos y periféricos internos y son compatibles con lenguaje de programación en C.

En la siguiente tabla se comparan algunos modelos de microcontroladores PIC así como sus datos técnicos correspondientes:

Tabla 6 Modelos y características de microcontroladores PIC

Microcontrolador	ROM	RAM	Tipo CPU	Velocidad CPU	MIPS
PIC18F4550	32kB	2kB	8bits	48MHz	12
dsPIC30F2010	12kB	512B	16bits	120MHz	30
dsPIC33EP64MC202	64kB	8kB	16bits	140MHz	70
PIC32MZ1024EFK144	1MB	262kB	32bits	200MHz	330

Para programar estos microcontroladores se usó el software MPLAB X v5.25 proporcionado por Microchip de manera gratuita, junto con los compiladores XC16 y XC32 que permiten una programación en lenguaje C y también en lenguaje ensamblador.

Se comenzó el estudio de los microcontroladores de 16 bits y con el entorno de desarrollo de MPLAB X v5.25 con el modelo dsPIC30F2010 (MICROCHIP, dsPIC30F2010 Datasheet, 2005), con el que se realizaron algunas prácticas

como uso de los puertos de entrada y salida, uso de retardos, código binario y hexadecimal, uso de pantalla de cristal líquido, uso de teclado y display de 7 segmentos y un reloj/calendario configurable con botones. Posteriormente se usó un modelo más avanzado, el dsPIC33EP64MC202, con el que realizamos algunas prácticas más avanzadas como, uso de los módulos ADC, uso del módulo UART y uso del módulo PWM (MICROCHIP, dsPIC33EP64MC202 Datasheet, 2013).

Se tomó la decisión de usar el microcontrolador PIC32MZ1024EFK144, ya que cuenta con acceso directo a memoria (DMA), así como también tiene disponibles módulos USART, ADC, I2C, SPI y PWM (MICROCHIP, PIC32MZ1024EFK144 Datasheet, 2016). Se realizaron algunas prácticas sencillas como uso de los puertos de entrada/salida, uso de retardos, uso de USART, configuración de puertos, configuración de registros, configuración de módulos y configuración de frecuencia de reloj.

2.8 DIODO EMISOR DE LUZ

El diodo emisor de luz o LED (Light Emitting Diode) es un dispositivo semiconductor que emite luz con una longitud de onda monocromática especifica muy bien definida cuando se polariza de forma directa pasando una corriente eléctrica entre sus extremos (Alfonso Gago Calderón, 2012).

2.9 LED WS2812B

El LED WS2812B es un LED inteligente que contiene un circuito de control y un LED RGB integrados en un solo empaquetado. El circuito de control incluye una memoria tipo LATCH inteligente, un módulo de acondicionamiento y amplificado de señal, un oscilador de precisión interno y un circuito regulador de voltaje para asegurar el color y el brillo del LED. Además, los LED WS2812B pueden ser conectados en serie (Worldsemi, 2019).

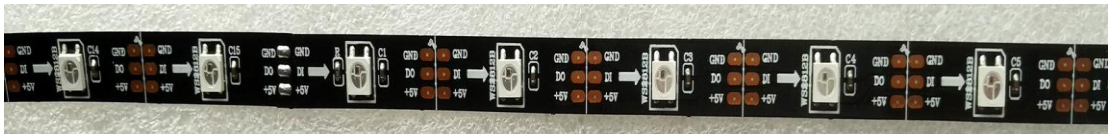

Fig. 6 Tira de LEDs WS2812B

Las características y beneficios de LED WS2812B son las siguientes:

Tabla 7 Caracteristicas del LED WS2812B

1	Protección inteligente contra conexión inversa.
2	El circuito de control y el LED comparten la fuente de alimentación.
3	El circuito de control y el LED están integrados en un solo componente.
4	Circuito de acondicionamiento y amplificación de señal incorporado.
5	Un LED RGB es capaz de mostrar 16777216 colores.
6	Transmisión de datos en serie.
7	Velocidad de transferencia de datos de 800Kbps

2.9.1 TRANSFERENCIA DE DATOS DEL LED WS2812B

El LED WS2812B utiliza un modo único de NZR (no zero return) como protocolo de comunicación para la transferencia de datos. Después de que el LED enciende, el pin DIN recibe datos del microcontrolador, el primer LED almacena sus respectivos datos y los envía a la memoria LATCH interna, los demás datos pasan al circuito de acondicionamiento y amplificación de señal y son enviados al siguiente LED de la serie a través del pin DO (ver Fig. 7).

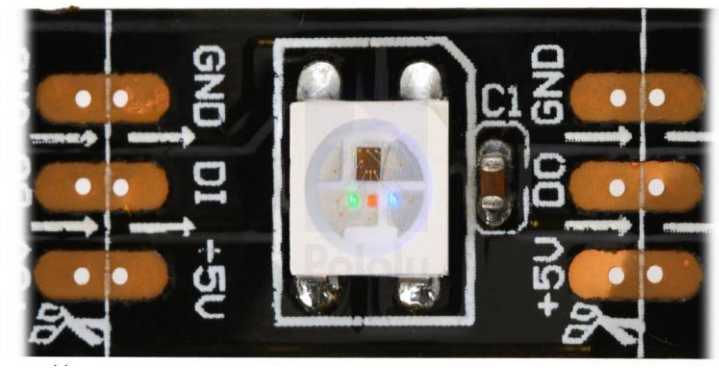

Fig. 7 LED WS2812B.

Para encender un LED WS2812B son necesarios 24 bits de datos. Estos 24 bits contiene los niveles de brillo que tendrá el LED RGB (8 bits para el rojo, 8 bits para el verde y 8bits para el azul), logrando que el LED RGB sea capaz de mostrar más de 16 millones de colores. La cadena de 24 bits de datos empieza por el bit más significativo para el nivel de brillo correspondiente al color verde y termina con el bit menos significativo del nivel de brillo correspondiente al color azul.

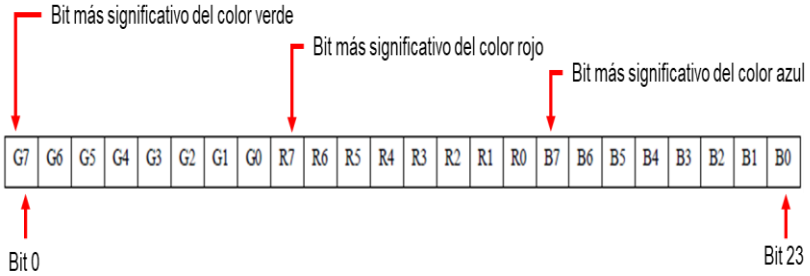

Fig. 8 Cadena de datos para LED WS2812B

Cada bit de control se rige por estados lógicos con diferentes tiempos para representar un 1 o un 0 de acuerdo con la siguiente tabla:

Tabla 8 Tiempos de protocolo WS2812B

Valor	Tiempo en alto	Tiempo en bajo	Tolerancia
1	0.8µS	0.45µS	±150nS
0	0.4µS	0.85µS	±150nS
RESET	Tiempo en bajo ≥ 50µS		

Si el LED WS2812B deja de recibir datos por un tiempo mayor o igual a 50 microsegundos, entonces el circuito de control se reinicia y se prepara para recibir nuevos datos de control.

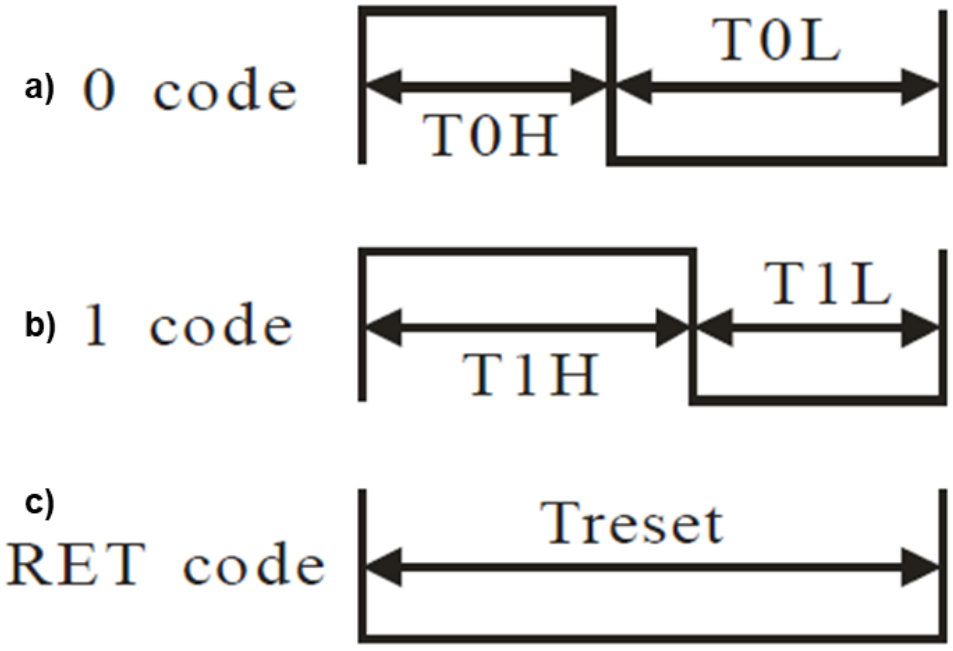

Fig. 9 Secuencias de tiempos para protocolo de transferencia del LED WS2812B. a) secuencia de tiempos para código 0, b) secuencia de tiempos para código 1, c) secuencia de tiempo para RESET (Treset≥50µS).

2.10 MATLAB

MATLAB es un lenguaje de programación de alto nivel basado en matrices/arreglos que posee sentencias de control de flujo, funciones, estructuras de datos y características de programación orientado al cálculo técnico. El nombre MATLAB proviene de Matrix Laboratory (Laboratorio de Matrices) dado que en sus orígenes fue escrito para facilitar el desarrollo de software matricial. (Arellano, 2013). MATLAB generalmente es utilizado en:

Tabla 9 Usos de MATLAB

1	Cálculo y matemática.
2	Desarrollo de algoritmos.
3	Modelamiento, simulación y prototipado.
4	Análisis, exploración y visualización de datos.
5	Gráficos científicos.
6	Desarrollo de aplicaciones con interfaces graficas (GUI).
7	Procesamiento digital de imágenes.

2.10.1 INTERFAZ GRAFICA DE USUARIO CON MATLAB

La interfaz gráfica de usuario, conocida también como GUI (del inglés Graphical User Interface) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Como en una GUI las acciones se realizan mediante manipulación directa, el usuario no tiene que crear un script, digitar algún comando o comprender los detalles de cómo se realizan las tareas para poder hacer alguna actividad con la aplicación. Las GUIs surgen como evolución de la línea de comandos de los primeros sistemas operativos y es pieza fundamental en un entorno gráfico (Arellano, 2013).

Desde el punto de vista de la programación en MATLAB, una GUI es una visualización gráfica de una o más ventanas que contienen controles, llamados componentes, que permiten a un usuario realizar tareas en forma interactiva.

Una GUI MATLAB es una ventana figura (figure) en la cual se añaden los controles operados por el usuario (componentes). A través de devoluciones de llamada (callbacks) se puede hacer que los componentes hagan lo que se desea cuando el usuario le da clic o los manipula con pulsaciones del teclado (keystrokes).

Se puede crear una GUI en MATLAB de dos maneras:

1) Usando GUIDE (GUI Development Environment). GUIDE es una herramienta interactiva para la construcción de GUIs. Se inicia con una ventana figura en la cual se colocan los componentes desde un editor de diseño. GUIDE crea los archivos M (archivos de MATLAB) asociados que contienes las callbacks de la GUI y sus componentes. GUIDE trabaja con dos tipos de archivos: Archivo para almacenar el diseño de la ventana figura (archivo .fig) y Archivo para almacenar el código fuente de la aplicación (archivo .m).
2) Usando solo archivos M (funciones o script) que generen los GUIs o construcciones programáticas de GUIs. Aquí, se codifica un archivo M que define todas las propiedades y comportamientos de los componentes; cuando un usuario ejecuta el archivo M, se crea una ventana figura con los componentes y manipuladores interactivos para el usuario.

2.10.2 PROCESAMIENTO DIGITAL DE IMÁGENES EN MATLAB

Una imagen puede ser definida matemáticamente como una función bidimensional f(x,y), donde x e y son coordenadas espaciales (en un plano), y f en cualquier par de coordenadas es la intensidad o nivel de gris de la imagen en esa coordenada (Rafael C. Gonzalez, 2002).

Cuando x, y y los valores de f son todas cantidades finitas, discretas decimos que la imagen es una imagen digital. Una imagen digital se compone de un numero finito de elementos, cada uno con lugar y valor específicos. Estos elementos son llamados pixeles.

En MATLAB, una imagen a escala de grises es representada por medio de una matriz bidimensional de m por n elementos en donde n representa el número de pixeles de ancho (columnas) y m representa el número de pixeles de largo (filas). Cada elemento de la matriz de la imagen tiene un valor 0 (negro) y 255 (blanco). Por otro lado, una imagen de color RGB es representada por una matriz tridimensional m por n por p, donde m y n significan lo mismo que en el caso anterior y p representa el plano, que para RGB puede ser 1 para el rojo, 2 para el verde y 3 para el azul.

El procesamiento de imágenes tiene como objetivo el mejoramiento de estas y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. El procesamiento de imágenes se puede hacer por métodos ópticos o bien por métodos digitales.

CAPITULO 3

METODOLOGÍA

3.1 MANUFACTURA DE PLACA DE DESARROLLO

El microcontrolador PIC32MZ1024EFK144 solo se encuentra disponible con un encapsulado de tipo LQFP (encapsulado plano de perfil bajo). Este tipo de encapsulado es de tipo montaje superficial, por lo que es imposible hacer conexiones y pruebas con ayuda de un protoboard. Debido a esto, se decidió hacer un diseño electrónico para fabricar una placa de desarrollo que nos facilite las conexiones del microcontrolador.

Para realizar el diseño electrónico de la placa utilizamos una versión estudiantil del software EAGLE de la empresa Autodesk. El software brinda todo un entorno para el diseño de las placas en PCB. Como primer paso, se hizo el diseño en esquemático de las conexiones básicas y los puertos de entrada/salida del microcontrolador.

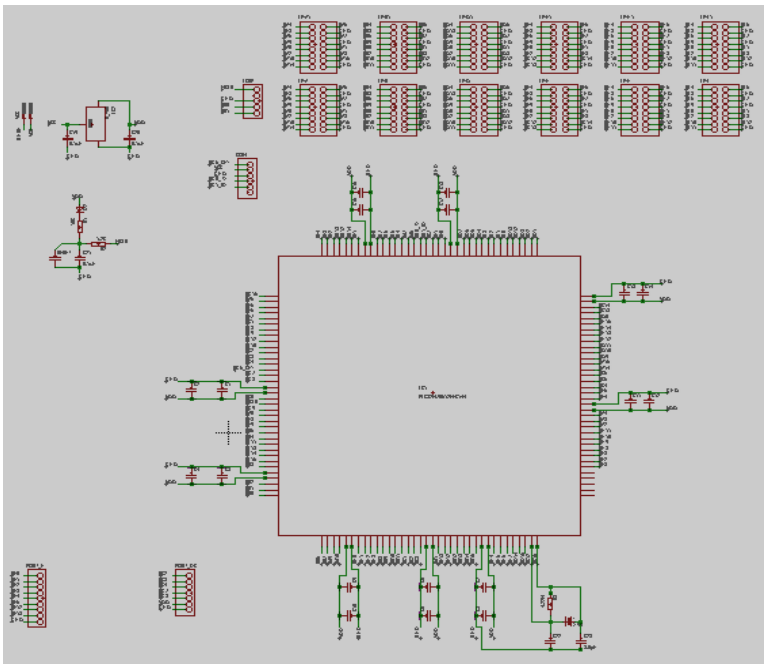

Fig. 10 Diseño electrónico de placa de desarrollo

Posteriormente se realizó el diseño electrónico de tarjeta PCB. Se diseñó una placa de desarrollo con un área de 10 x 10 cm en la que se intentó aprovechar de todos los pines de entrada/salida con los que cuenta el microcontrolador.

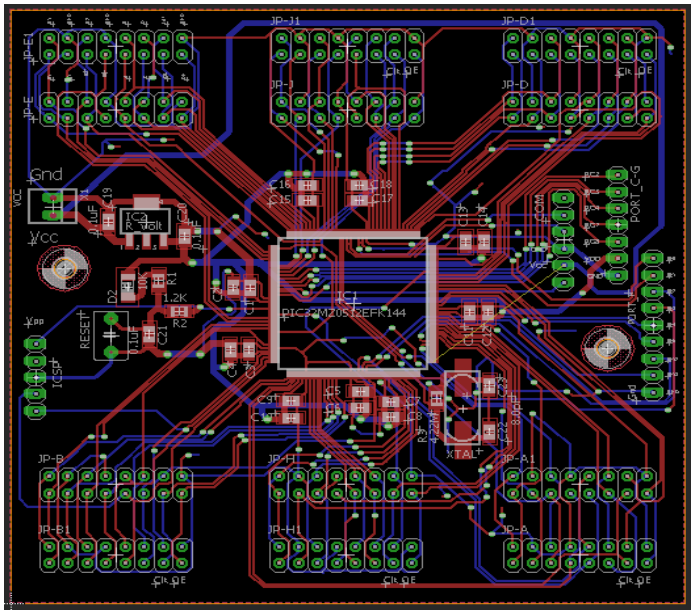

Fig. 11 Diseño PCB de placa de desarrollo

Al tener listo el diseño de la placa de desarrollo, se mandó a manufacturar a una empresa de China, por todas las facilidades que esta ofrece. El proceso de manufactura y envío de la placa tardo un periodo de dos semanas.

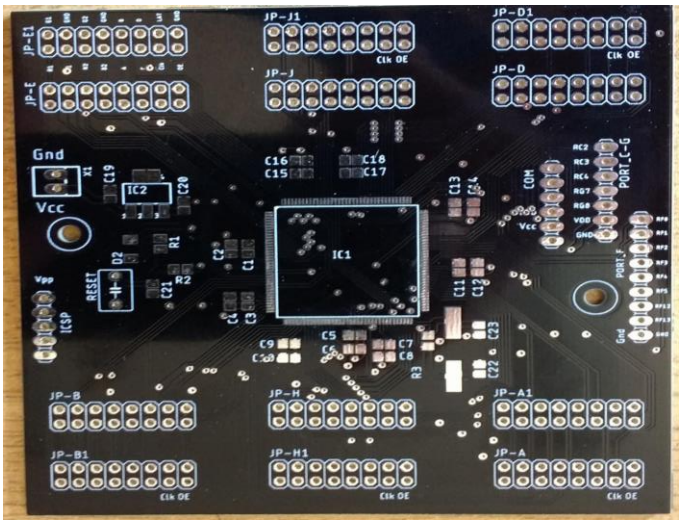

Fig. 12 Placa de desarrollo manufacturada

Por último, se procedió a colocar y a soldar todos los materiales necesarios para el funcionamiento de la placa. Los materiales son los siguientes:

Tabla 10 Lista de materiales para placa de desarrollo

1	Microcontrolador PIC32MZ1024EFK144.
2	Oscilador de cristal de alta velocidad de 24 MHz tipo montaje superficial.
3	2 tiras de pines tipo hembra de doble fila.
4	1 tira de pines tipo hembra de fila única.
5	23 capacitores de varios valores de tipo montaje superficial.
6	Regulador lineal de voltaje 78M33 de tipo montaje superficial.
7	2 resistencias de tipo montaje superficial.
8	1 diodo de tipo montaje superficial.
9	1 conector tipo molex de dos pines tipo macho.
10	1 tira de pines tipo macho en ángulo de 90 grados.
11	1 botón pulsador normalmente abierto.

Cabe mencionar que el microcontrolador PIC32MZ1024EFK144 fue proporcionado de manera gratuita por Microchip, el único pago que se cubrió fueron los gastos por el envío a Cuernavaca Morelos México.

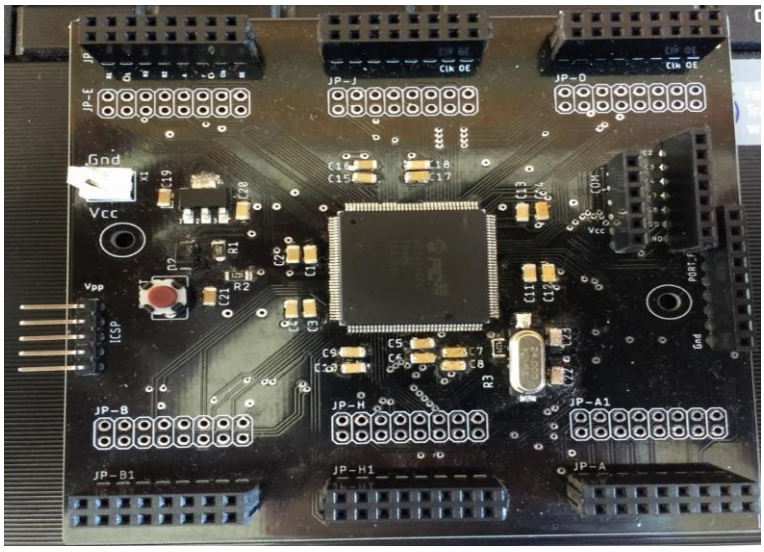

Fig. 13 Placa de desarrollo armada

3.2 TIRA LED WS2812B

El primer acercamiento que tuvimos con los LEDs RGB WS2812B fue con una tira de tan solo 5 LEDs, y para el control se usó el microcontrolador PIC18F4550, con arquitectura de 8 bits, funcionando a una frecuencia de trabajo de 12 MHz.

Se decidió hacer esta prueba con un microcontrolador de 8 bits debido a que ya se tenían conocimientos previos para el uso y la programación de estos microcontroladores, además el PIC18F4550 es de una gama avanzada por lo que puede trabajar a un ciclo máquina de 83.3 nano segundos.

Con esta primera prueba se pretendía comprender el protocolo NRZ, con el que se controlan los LEDs WS2812B, para lograr programar un algoritmo que controle el estado lógico de uno de los pines del microcontrolador de acuerdo con los tiempos establecidos en la Tabla 8.

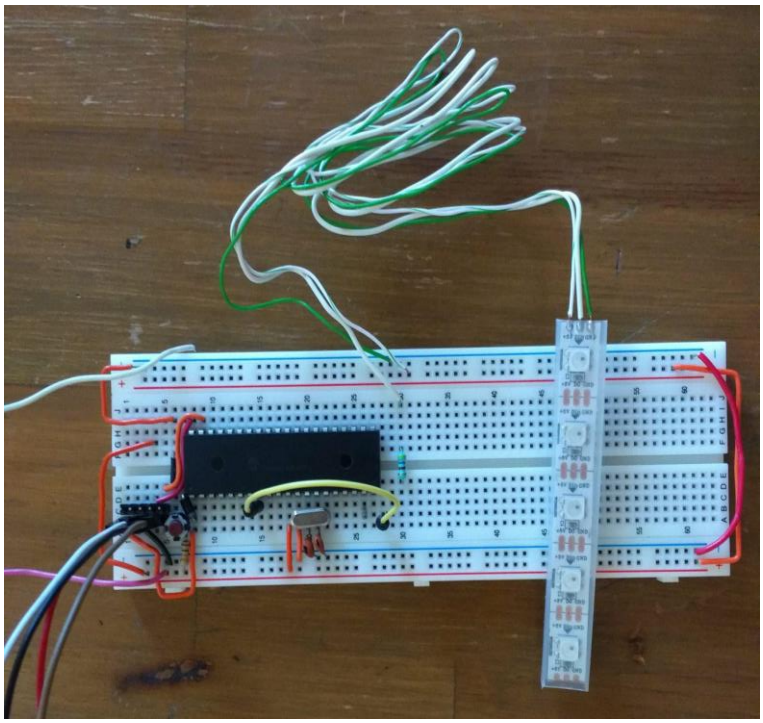

Fig. 14 PIC18F4550 y tira de LEDs WS2812B

3.2.1 ALGORITMOS DE CONTROL DE TIRA LED WS2812B

Para programar el protocolo de comunicación del LED WS2812B, es necesario que el microcontrolador reconozca el dato, ya sea 1 o 0, que se enviará al LED WS2812B e identifique que instrucciones debe seguir. Para resolver esto, implementamos una estructura de toma de decisiones. De esta manera el microcontrolador sabrá que instrucciones deberá seguir según la decisión tomada. A este algoritmo lo hemos nombrado “NeoBit” y el diagrama de flujo que lo representa es el siguiente:

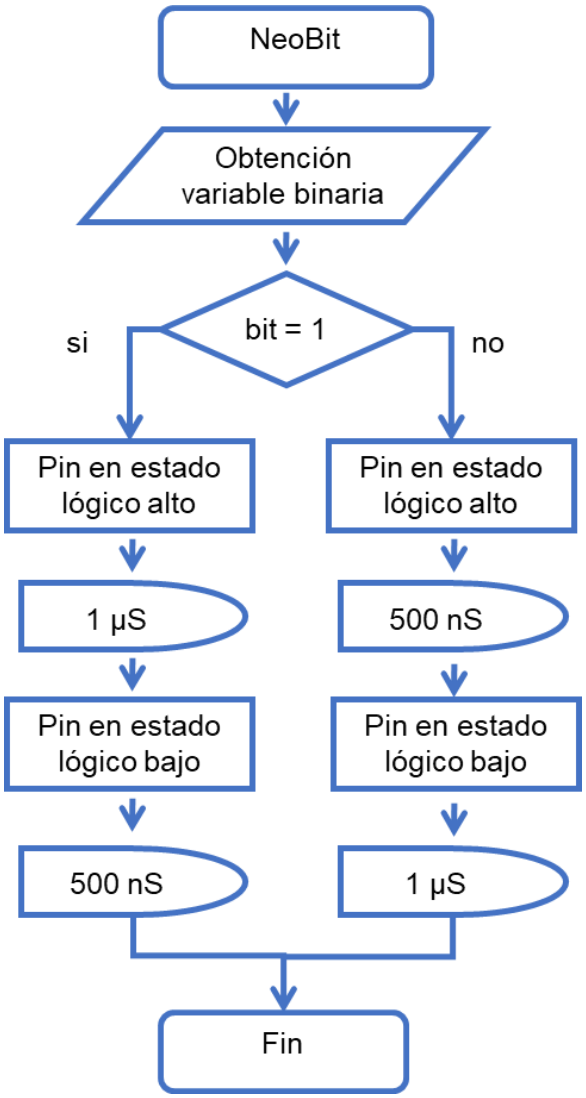

Fig. 15 Diagrama de flujo de función NeoBit

El protocolo de comunicación necesita de un total de 24 bits de datos, el algoritmo NeoBit solo realiza el envío de un bit a la vez y solo reconoce variables de tipo binario (1 o 0) para hacer la toma de decisiones. Se necesita crear otro algoritmo que nos permita ingresar en formato decimal y de forma separada los niveles de brillo que tendrá cada color del LED RGB, es decir, el nivel del color rojo, del verde y del azul, y que este sea capaz de convertirlos a formato binario y, mediante el algoritmo NeoBit, enviar la información al LED WS2812B. El nombre de este algoritmo es NeoDraw y su diagrama de flujo es el siguiente:

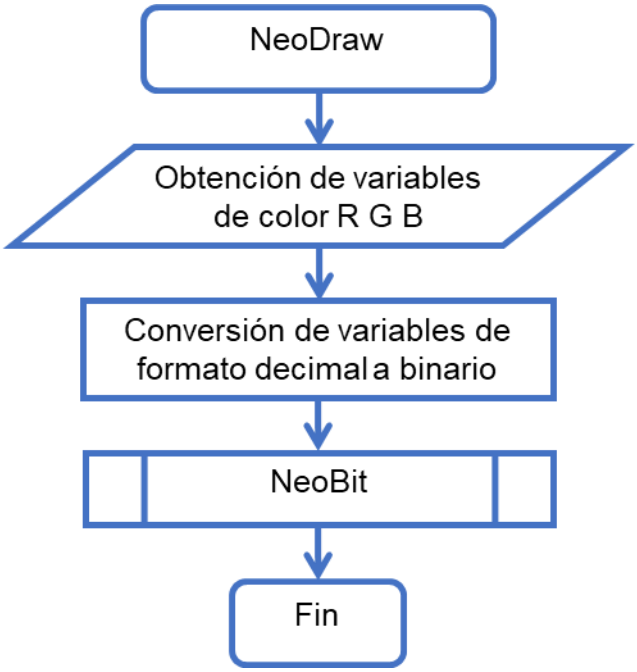

Fig. 16 Diagrama de flujo de función NeoDraw

Por último, el diagrama de flujo de flujo que representa al algoritmo para el control de una tira de LEDs con 5 LEDs RGB WS2812B es el siguiente:

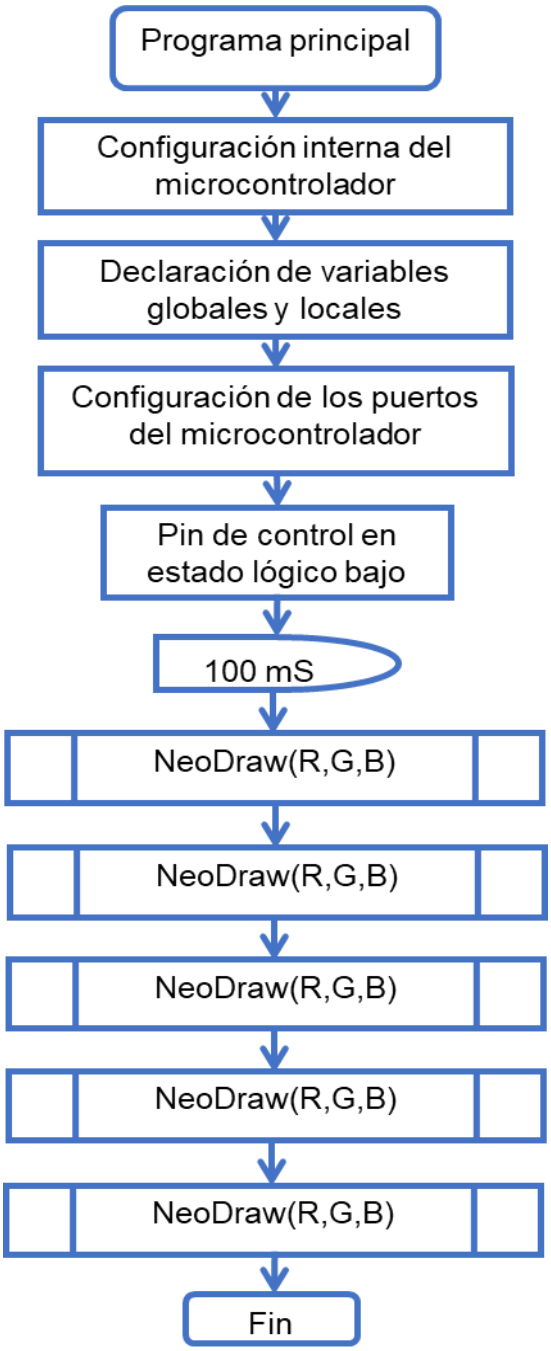

Fig. 17 Diagrama de flujo para control de tira de LEDs WS2812B

Posteriormente, este mismo algoritmo se implementó en nuestra placa de desarrollo (Fig. 13). Aunque este algoritmo fue implementado en dos microcontroladores totalmente diferentes, cabe mencionar que el propósito del algoritmo es exactamente el mismo, es decir, controlar la tira con 5 LEDs RGB.

3.3 MATRIZ DE LEDs WS2812B DE 8X5 PIXELES

Antes de comenzar la construcción de la matriz de LEDs que se utilizó para este proyecto, primero se construyó una pequeña matriz para realizar las primeras pruebas de programación. Esta matriz consiste en 8 filas y 5 columnas, ya que, con esta configuración es posible mostrar caracteres como son las letras en minúsculas y mayúsculas, los números, signos de puntuación y símbolos como @, #, $, %, &, etc.

Para construir esta matriz de 8x5 pixeles fueron necesarios los siguientes materiales:

Tabla 11 Materiales para matriz de 8x5 pixeles

1	8 tiras con 5 LEDs RGB WS2812B cada una.
2	Un tramo de cartón.
3	Cables de conexión.

Comenzamos colocando las 8 tiras de 5 LEDs RGB sobre el tramo de cartón, esto es para darle un soporte a esta pequeña matriz. La forma en que se colocaron las tiras LED es muy importante, ya que para que la matriz funcione correctamente, estas deben de ser colocadas en forma de zigzag, es decir, la siguiente tira a colocar debe ir en sentido contrario a la anterior.

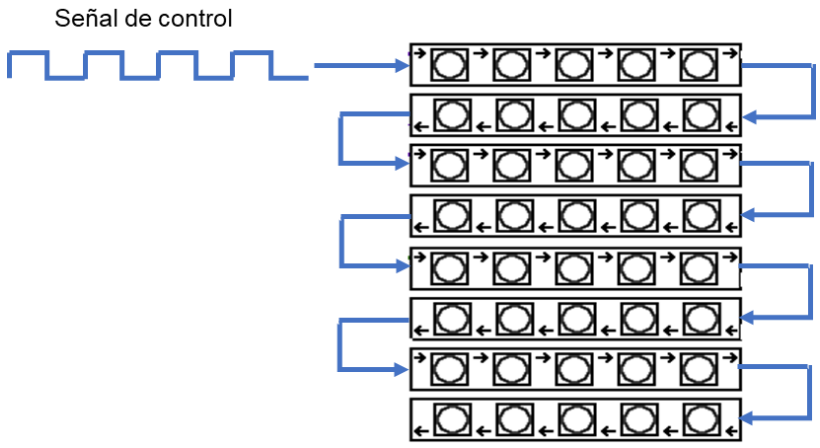

Fig. 18 Conexiones de tiras LED WS2812B

Por último, se soldaron cables para las conexiones de alimentación y para la señal de datos, así también se soldaron puentes entre las tiras LED para la alimentación y la señal de datos.

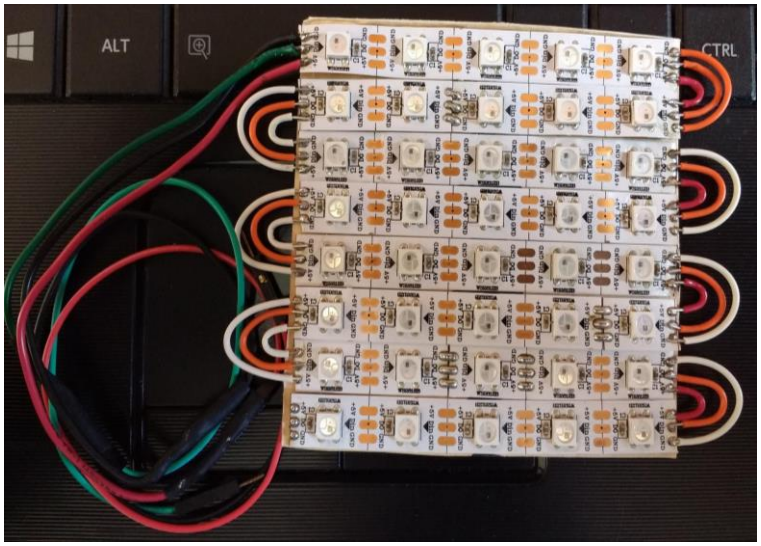

Fig. 19 Matriz de LEDs WS2812B de 8x5 pixeles

3.3.1 PRUEBA DE MATRIZ DE 8X5 PIXELES

La primera prueba que se le realizó a la matriz de LEDs de 8x5 pixeles fue comprobar que los LEDs de la primera fila funcionaran correctamente, para esto se ocupó el algoritmo de la Fig. 17, el cual ya se encontraba grabado en la memoria de programa del microcontrolador.

3.3.2 GRÁFICOS EN MATRIZ DE 8X5 PIXELES

Para lograr que la matriz de LEDs imprima gráficos, se necesita almacenar los datos necesarios para encender los 40 LEDs que componen a la matriz de 8x5 pixeles. Para esto, implementamos un arreglo unidimensional (vector) de 40 elementos, donde cada elemento del vector representan el color al que debe de encender un LED RGB de la matriz. Para almacenar un gráfico, se debe de colocar el valor del color al que deseamos que encienda el LED que forma el grafico y colocar 0 para el LED que debe permanecer apagado, los elementos del vector deben ser acomodados respetando el orden de la Fig. 18.

Mediante una estructura de ciclo for y una variable de tipo entero que sirva como contador se realiza la llamada a cada uno de los elementos del vector y utilizando los algoritmos NeoBit (Fig. 15) y NeoDraw (Fig. 16) se realiza el envío de datos hacia la matriz de LEDs.

El diagrama de flujo para imprimir gráficos en la matriz de LEDs de 8x5 pixeles es el siguiente:

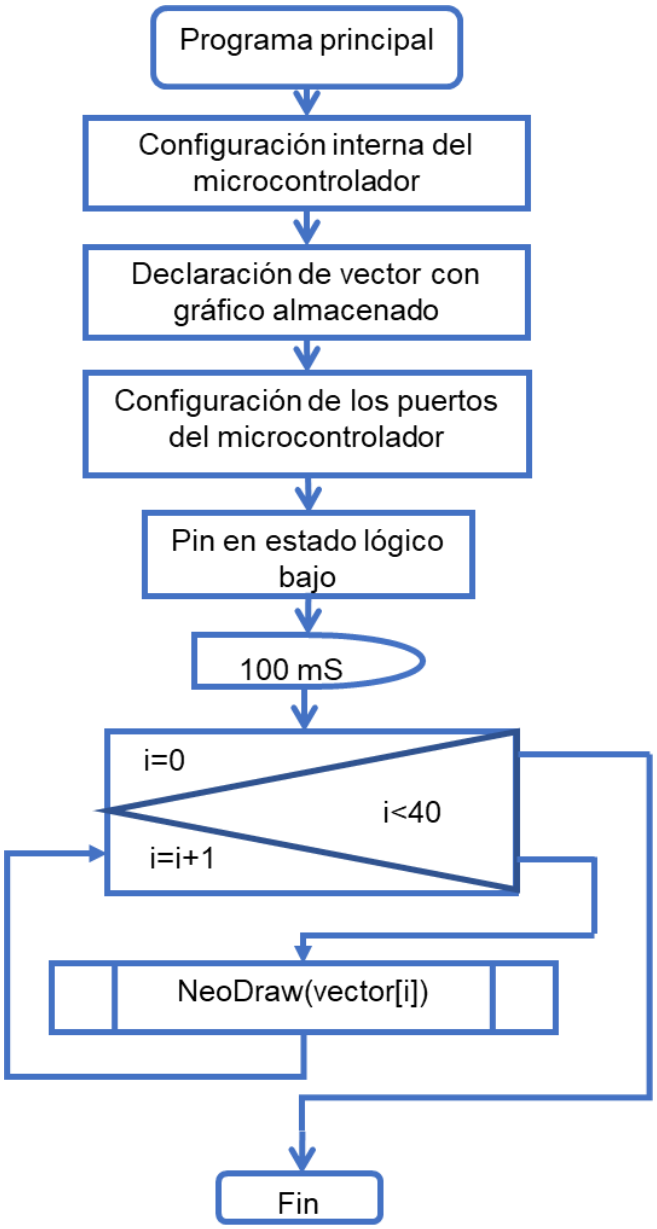

Fig. 20 Diagrama de flujo ara gráficos en matriz de 8x5 pixeles

3.3.3 DESPLAZAMIENTO DE CARACTERES EN MATRIZ LED

La siguiente prueba que se realizó con la matriz LED de 8x5 pixeles fue la de mostrar caracteres ASCII. Debido al reducido tamaño de la matriz de LEDs, solo se podrá visualizar un carácter a la vez. Para mostrar una cadena de caracteres se tiene que lograr reproducir un efecto de desplazamiento.

Como primero paso para programar un algoritmo que muestre un carácter ASCII en la matriz LED, se tiene que establecer la manera en que se representaran y se almacenaran los caracteres ASCII.

Imaginemos que tenemos una tabla de 8 filas por 5 columnas. Cada una de las celdas de la tabla puede tomar un valor binario, es decir, 1 o 0. Si una celda toma el valor de 1, esta se torna de color negro. Si la celda toma el valor de 0, entonces la celda se queda de color blanco.

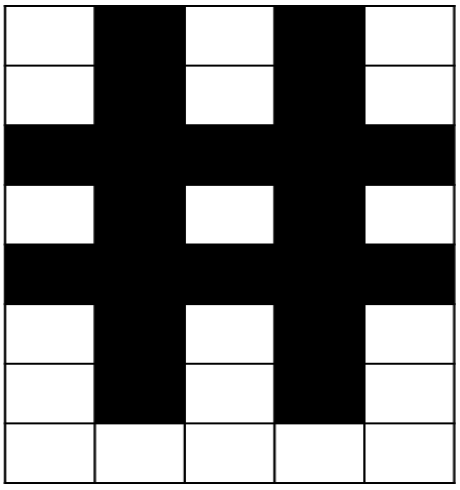

Fig. 21 Tabla de valores binarios (8 filas, 5 columnas)

Si tomamos los valores binarios de las celdas de una fila, podemos formar un numero binario de 8 bits, donde el bit menos significativo corresponde a la celda de la fila 1 y el bit más significativo corresponde a la celda de la fila 8.

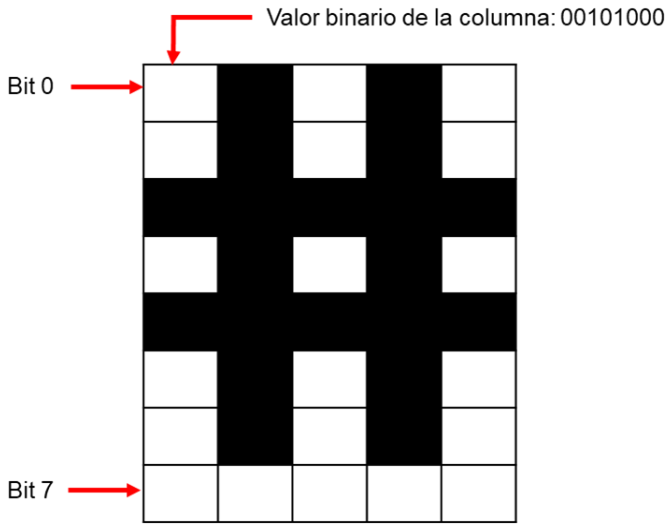

Fig. 22 Valor de 8 bits de una columna de la tabla binaria

Si realizamos la misma operación con las demás columnas de la tabla binaria, y además usamos el formato hexadecimal para representar los valores de las columnas, podemos representar caracteres ASCII de una manera muy práctica.

0x14	0x7F	0x14	0x7F	0x14
0	1	0	1	0
0	1	0	1	0
1	1	1	1	1
0	1	0	1	0
1	1	1	1	1
0	1	0	1	0
0	1	0	1	0
0	0	0	0	0

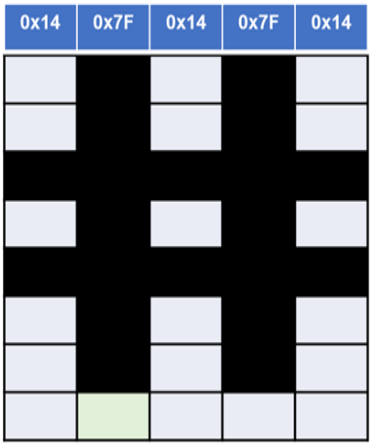

Fig. 23 Carácter "#" de 8x5 pixeles

Los caracteres los almacenaremos en la memoria del microcontrolador mediante la implementación de un arreglo bidimensional (matriz) de 100 filas por 5 columnas, donde cada fila de la matriz corresponde a un carácter ASCII de 8x5 pixeles, lo que significa que hemos almacenado un total de 100 caracteres diferentes.

El siguiente paso en la programación del algoritmo es reproducir un efecto de desplazamiento que nos permita visualizar una cadena de caracteres en la matriz LED de 8x5 pixeles.

De la misma manera que en el caso anterior, supongamos que tenemos una tabla binaria de 8 filas por 255 columnas. Las celdas de la tabla binaria toman los valores de la siguiente cadena de caracteres “HOLA MUNDO”. La tabla binaria está bloqueada y solo es posible ver su contenido a través de una ventana de 8 filas por 5 columnas. Esta ventana puede hacer un movimiento de desplazamiento de izquierda a derecha y así poder ver todo el contenido de la tabla binaria.

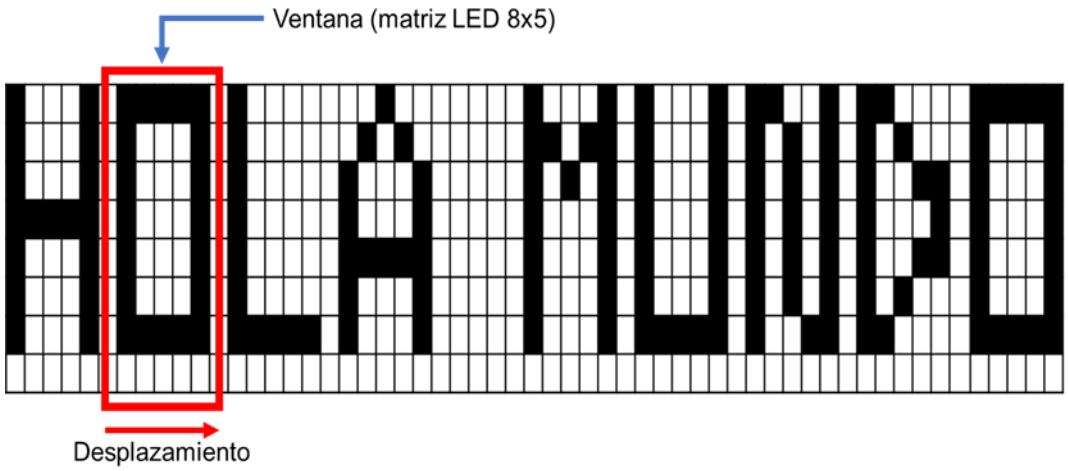

Fig. 24 Efecto de desplazamiento de caracteres

Declarando una matriz de 8 filas por 255 columnas, podemos almacenar una cadena con un total de 51 caracteres. Usando un vector de 40 elementos se logra simular la ventana que permite visualizar cada uno de los elementos de la cadena de caracteres.

Implementando algunos ciclos for que nos ayude a ubicar los elementos de la ventana, podemos reproducir el efecto de desplazamiento de los caracteres. Se debe de tener precaución en cómo se acomodarán los elementos del vector de la ventana, debido a la manera en que la matriz de LEDs se ensamblo.

El diagrama de flujo para el algoritmo de desplazamiento de caracteres en matriz de LEDs de 8x5 pixeles es el siguiente:

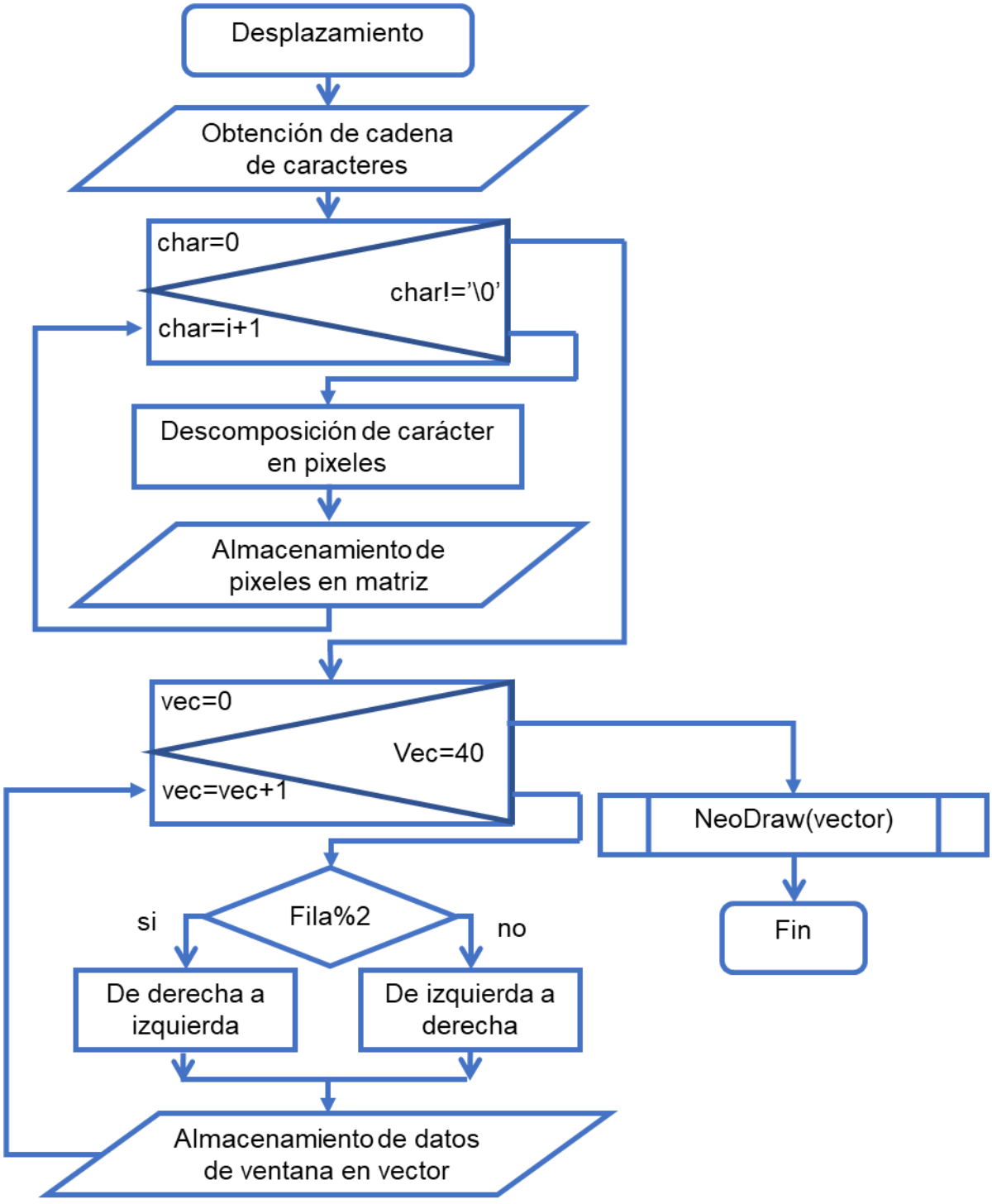

Fig. 25 Diagrama de flujo para algoritmo de desplazamiento de caracteres

3.4 CONCTRUCCION DE LA MATRIZ DE LEDS RGB

Para iniciar a construir la matriz de LEDs, se hizo una lista de todos los materiales electrónicos y misceláneos que se ocuparían, así como también algunos sencillos cálculos para determinar la cantidad de LEDs necesarios para trabajar 30 cuadros por segundo de actualización de pantalla, también para calcular la cantidad de corriente que la matriz de LEDs consumirá.

Como se vio en el capítulo anterior en la sección 3.9.1, el LED WS2812B trabaja con un protocolo NRZ, con el que se logra una transferencia de datos en un tiempo bastante reducido.

Con los valores de la Tabla 8, podemos calcular la cantidad de LEDs necesarios para construir una matriz de LEDs que tenga una tasa de actualización de 30 cuadros por segundo. Se hicieron los siguientes cálculos:

Tiempo para encender un LED: $T_L = (1.4uS) \times (24) = 33.6uS$

Tiempo para reset: $T_r = 50uS$

Imágenes por segundo: 30FPS = 1/30 = 33.33mS

Cantidad de LEDs a 30FPS: $C_L = (33.33mS – 50nS) / 33.6uS = 990.5753$ LEDs

Tiempo de actualización: $T_A = (33.6uS * 990LEDs) + 50uS = 30.29mS$

Corriente nominal por LED: $C_L = 60mA$

Corriente total: $C_T = 60mA * 990LEDs = 59.4$ A

Con los cálculos anteriores, ahora sabemos la cantidad necesarios para construir una matriz de LEDs de 990 LEDs, que estará formada por 30 filas y 33 columnas, así también conocemos la corriente total necesaria para el correcto funcionamiento de cada LED, la cual es de 59.4 A. Los componentes necesarios para armar la matriz de LEDs de 30 filas y 33 columnas se muestran en la siguiente tabla:

Tabla 12 Materiales para construir una matriz de LEDs WS2812B de 30x33 pixeles

1	17m de tira LED RGB con circuito integrado WS2812B.
2	Fuente de poder de 5 V, 70 A y 350 W.
3	50cm cable dúplex AWG calibre 12.
4	Clavija.
5	Cables de conexión.
6	6 terminales.
7	1 pieza de papel cascaron.
8	Tinte negro para zapatos.

Después de contar con todo el material necesario para construir la matriz de LEDS, iniciamos preparando los LEDs que conformaran la matriz, para esto se hicieron 30 tiras con 33 LEDs cada una.

Fig. 26 Tiras LED WS2812B

Posteriormente, se recorta el papel cascaron a una medida de 55cm de largo por 50cm de alto, y con el tinte negro para zapatos se pintó la superficie del papel cascaron, esto fue para colocar las 30 tiras con 33 LEDs cada una sobre el papel el papel cascaron y así darle un soporte a la matriz LED. Por último, se hacen las conexiones necesarias para interconectar las tiras LED en forma de zigzag, para la trasmisión de datos, así como también de la alimentación, siguiendo el ejemplo de las conexiones vistas en la Fig. 18.

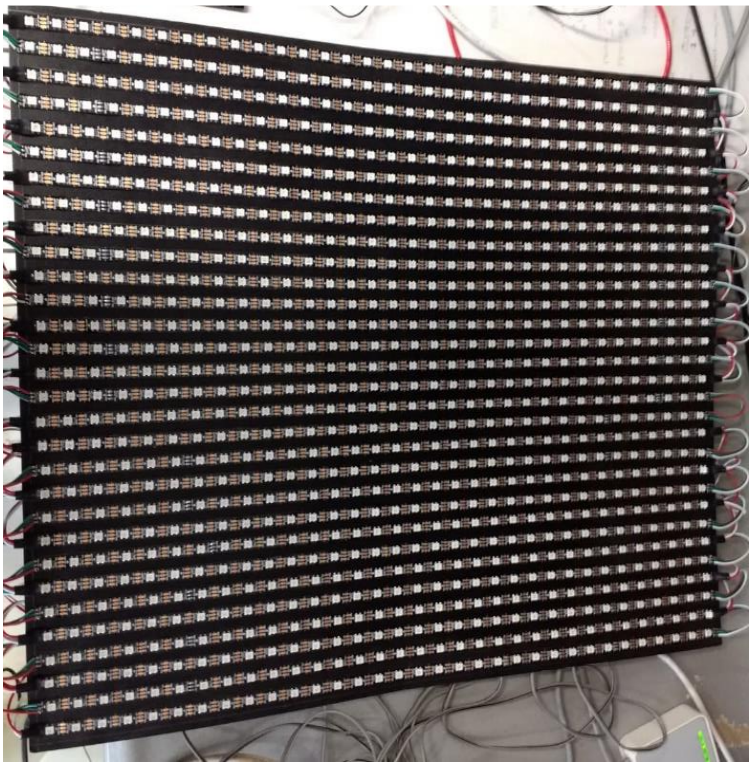

Fig. 27 Matriz de LEDs WS2812B de 30x33 pixeles

Para darle una mejor definición a las imágenes proyectadas en la matriz de LEDs, son construyo una rejilla echa con tiras de papel cascaron, para separar cada LED de forma individual.

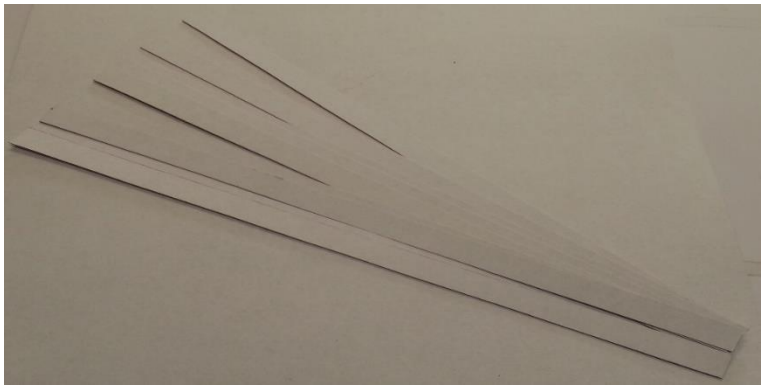

Fig. 28 Tiras de papel cascaron

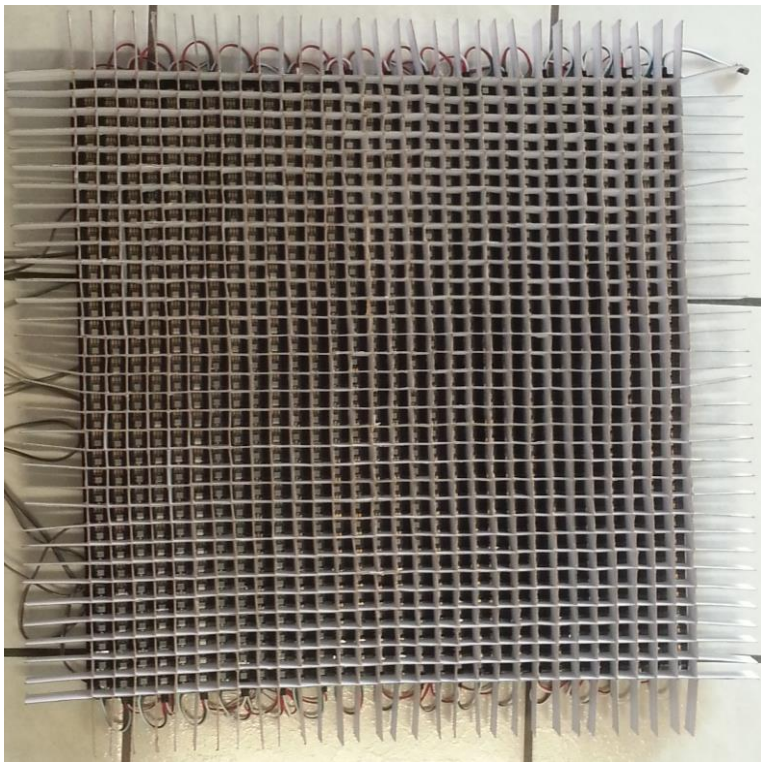

Fig. 29 Reílla hecha con tiras de papel cascaron

Aunque la matriz de LEDs ya estaba armada, no era correcto, hablando en términos técnicos y estéticos, dejarla de esa manera. Por lo que se construyó un cajón de madera para colocar en su interior a la matriz de LEDs.

Por último, también se adquirió un tramo de acrílico transparente, que se recortó a las medidas de la matriz de LEDs, después se le aplicó un trabajo de esmerilado sobre la superficie del acrílico.

Ya con todos estos materiales listos, se introdujo en el interior del cajón, el acrílico esmerilado, la rejilla de papel cascaron, la matriz de LEDs y la fuente de alimentación, todo en ese orden.

Fig. 30 Cajón de madera con matriz de LEDs en su interior

3.5 PRUEBAS DE TEXTO EN MATRIZ LED DE 30X33 PIXELES

3.5.1 DESPLAZAMIENTO DE CARCATERES

La primera prueba realizada a la matriz de LEDs de 30x33 pixeles consistió en el desplazamiento de caracteres. Utilizando el mismo algoritmo visto en la Fig. 25 y realizando algunas pequeñas modificaciones para que este funcione en la matriz de 30x33 pixeles.

3.5.2 CARACTERES ESTATICOS

Para esta prueba, se tiene que programar un algoritmo que permita mostrar un texto en la matriz de LEDs. Tomando como referencia el algoritmo que se muestra en la Fig. 25, se puede obtener un algoritmo para mostrar texto de manera estática en la matriz de LEDs.

Debido al tamaño de la matriz de LEDs y al número de pixeles necesarios para imprimir un carácter alfanumérico, solo se permite visualizar una cadena de caracteres de 18 elementos en 3 renglones como se observa en la Fig. 31.

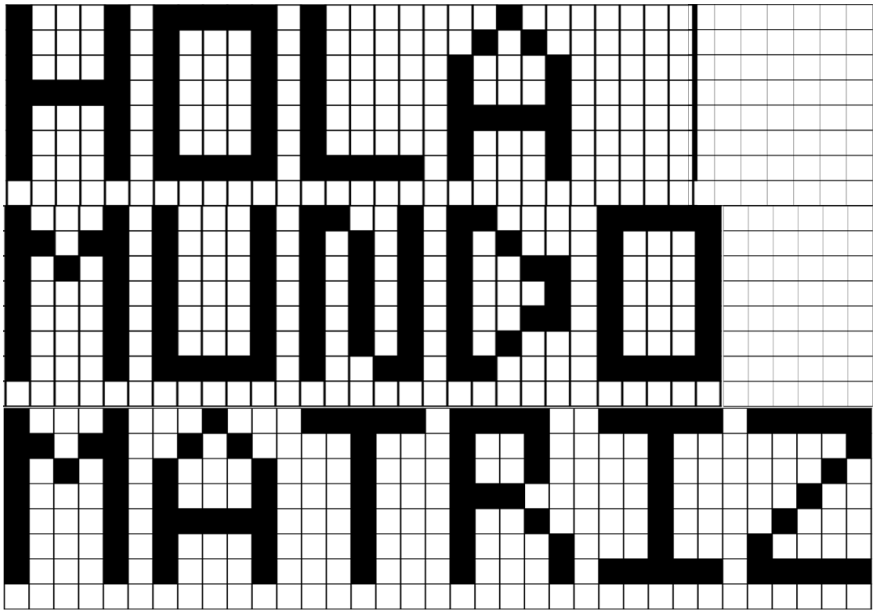

Fig. 31 Texto en matriz de 30x33 pixeles

Para mostrar el texto de una cadena de caracteres, los pixeles de cada uno de los caracteres deben acomodarse como si solo existiera un solo reglón como se muestra en la Fig. 32.

Fig. 32 Renglón con cadena de caracteres

Al igual que en el algoritmo de la Fig. 25, almacenamos la cadena de caracteres que deseamos visualizar en un arreglo bidimensional (matriz) de 8 filas y 90 columnas, el cual permite almacenar un total de 18 caracteres. También es necesario utilizar un arreglo unidimensional (vector) para almacenar los pixeles de los caracteres que se visualizaran en la matriz, este vector tendrá un tamaño total de 990 elementos, que es el número de pixeles con los que cuenta la matriz de LEDs RGB.

El algoritmo se encarga de almacenar y acomodar los pixeles de la cadena de caracteres que deseamos mostrar en la matriz de LEDs RGB, también se encarga de almacenar correctamente los elementes del vector que es el encargado de mandar la información hacia la matriz de LEDs.

El diagrama de flujo que representa al algoritmo para imprimir una cadena de caracteres en la de LEDs RGB de 30x33 pixeles es el siguiente:

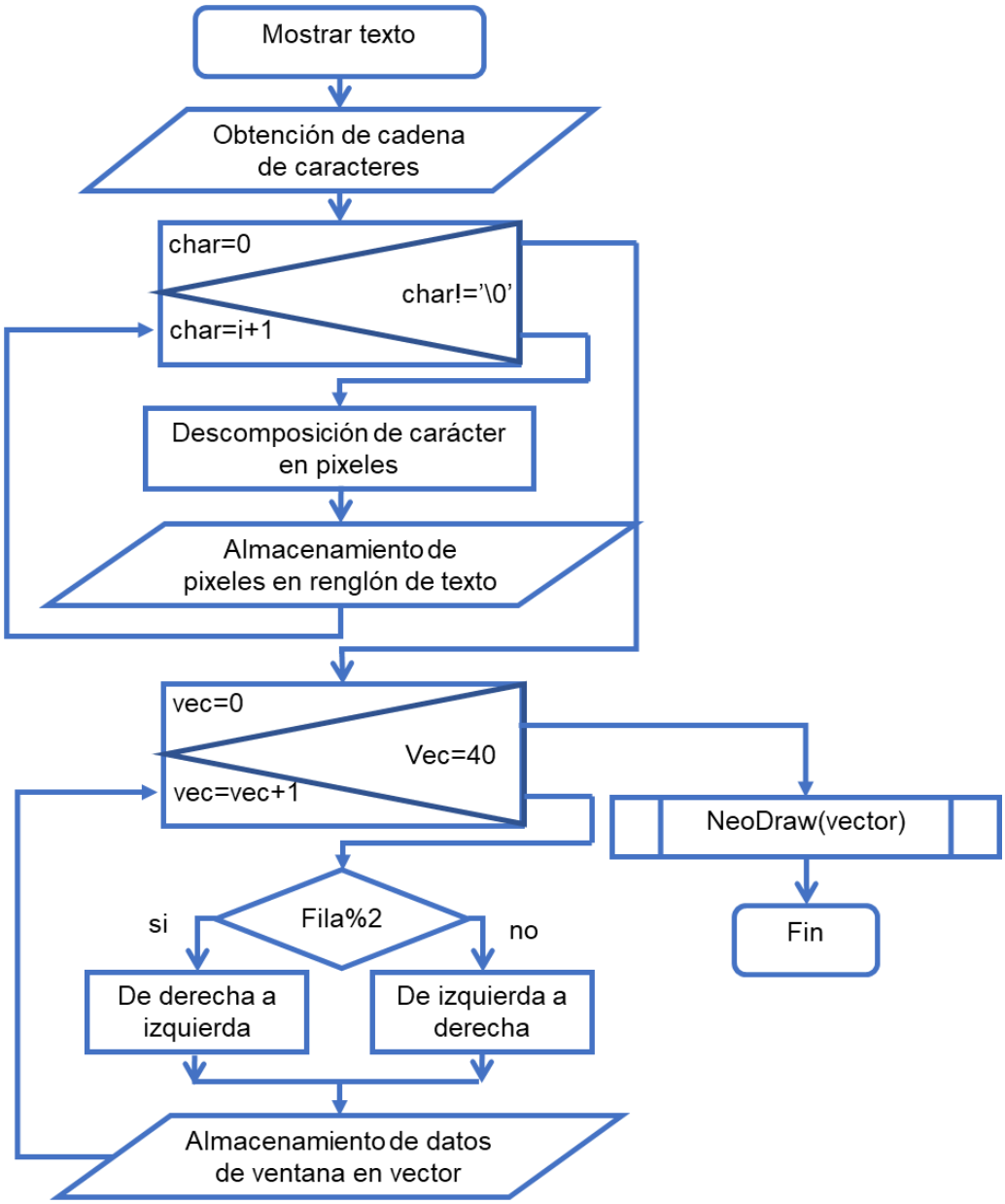

Fig. 33 Diagrama de flujo para texto estático en matriz LED de 30x33 pixeles

3.6 MÓDULO UART DEL MICROCONTROLADOR

Uno de los retos de este proyecto es lograr una comunicación serie entre una PC y la matriz de LEDs RGB, para recibir una cadena de caracteres y también para recibir una imagen y poder visualizarla en la matriz de LEDs. Por esta razón, tuvimos que realizar la configuración del módulo UART del microcontrolador PIC32MZ1024EFK144.

Para configurar el módulo UART del microcontrolador es necesario conocer cuáles son los registros de control y configuración correspondientes al módulo, también es necesario calcular la tasa de baudios para establecer la velocidad de transmisión de datos de la comunicación serie bajo el protocolo RS232.

Las ecuaciones necesarias para el cálculo de la tasa de baudios las podemos en la hoja de datos del microcontrolador, la ecuación es la siguiente:

$UxBRG = (F_{pb} / 16 * Baud\ rate) – 1;$

$F_{pb} = F_{sys} / 2;$

Donde:

UxBRG es un registro del microcontrolador que se encarga de manejar la velocidad de transmisión de datos de la comunicación serie.

Baud rate es la velocidad de transmisión que nosotros a la que nosotros deseamos trabajar. En este caso, manejaremos una tasa de baudios de 115200 bits por segundo.

F_{pb} es la frecuencia a la que trabajará el módulo UART del microcontrolador.

F_{sys} es la frecuencia a la que se encuentra trabajando la CPU del microcontrolador. Trabajamos con una frecuencia de 240 MHz.

Sustituyendo valores en la ecuación del registro UxBRG obtenemos:

Fpb = 240 MHz / 2 = 120 MHz

UxBRG = (120 MHz / 16 * 115200) – 1 = 64.104 ≈ 64

El registro UxMODE nos ayuda a configurar otras opciones de la comunicación serie bajo protocolo RS232, por ejemplo, activar o desactivar el módulo UART del microcontrolador, manejar de datos de 8 bits, el tipo de paridad y si habrá uno o dos de parada.

Para que el módulo UART funcione, se tiene que configurar los pines del microcontrolador que servirán para la recepción (RX) y transmisión (TX) de los datos. El microcontrolador no tiene definidos los pines RX y TX de fábrica, ya que una de las características de este es que los pines asociados a los módulos del microcontrolador pueden ser configurados por software. Los pines del módulo UART se configuran mediante los registros RPG0R y U1RXR, para los pines TX y RX respectivamente.

Una vez configurado el módulo UART del microcontrolador, ya podemos recibir una cadena de caracteres a través del puerto serie de un PC. Para identificar que un nuevo dato ha sido recibido en el pin RX usamos el registro U1STA el cual mediante el bit RXDA nos informa de un nuevo dato recibido.

El módulo UART trabaja bajo los protocolos de la comunicación serie, este tipo de comunicación necesita que una PC disponga de una conexión de tipo serie. La conexión de tipo serie es muy antigua, usa conectores tipo DB9 de puerto serie RS-232, este tipo de conectores ya no soy comunes en las maquinas actuales, ya que estas solo disponen de puertos USB.

Fig. 34 Conexión tipo DB9 de puerto serie

Afortunadamente, en el mercado existen un gran número de adaptadores y módulos que convierten el protocolo de comunicación serie RS232 en señales compatibles con la conexión USB.

En nuestro caso, se seleccionó el módulo YP-05 el cual contiene un circuito integrado ftdi ft232rl que es un circuito especializado en la conversión de USB a niveles TTL.

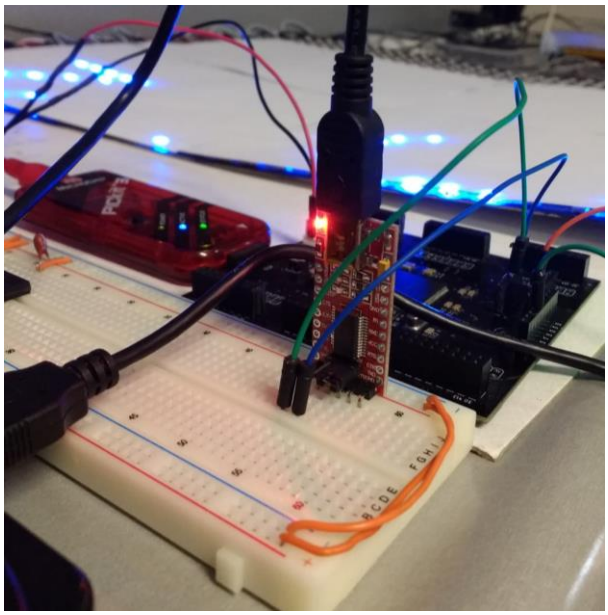

Fig. 35 Módulo conversor de USB a TTL ft232rl

3.7 RECEPCIÓN DE CADENA DE CACRACTERES POR UART

Para realizar la recepción de una cadena de caracteres en el microcontrolador, se tiene que definir el número de elementos que serán recibidos. Como se vio en la sección 4.5.2, la matriz solo es capaz de mostrar un total de 18 caracteres, entonces el número total de caracteres que se podrán recibir por el módulo UART es de 18 datos.

Para almacenar los datos recibidos a través del módulo UART usaremos un arreglo unidimensional (vector). El microcontrolador debe de verificar de manera continua que un dato serie se encuentra disponible para ser almacenado en el vector. Cuando el número total de elementos que se han

almacenado en el vector sea igual a 18, significa que se ha recibido una cadena de caracteres que debe ser mostrada en la matriz de LEDs. Para mostrar la cadena de caracteres en la matriz de LEDs usamos el algoritmo visto en la Fig. 33.

El diagrama de flujo para el algoritmo de recepción de datos a través del módulo UART del microcontrolador es el siguiente:

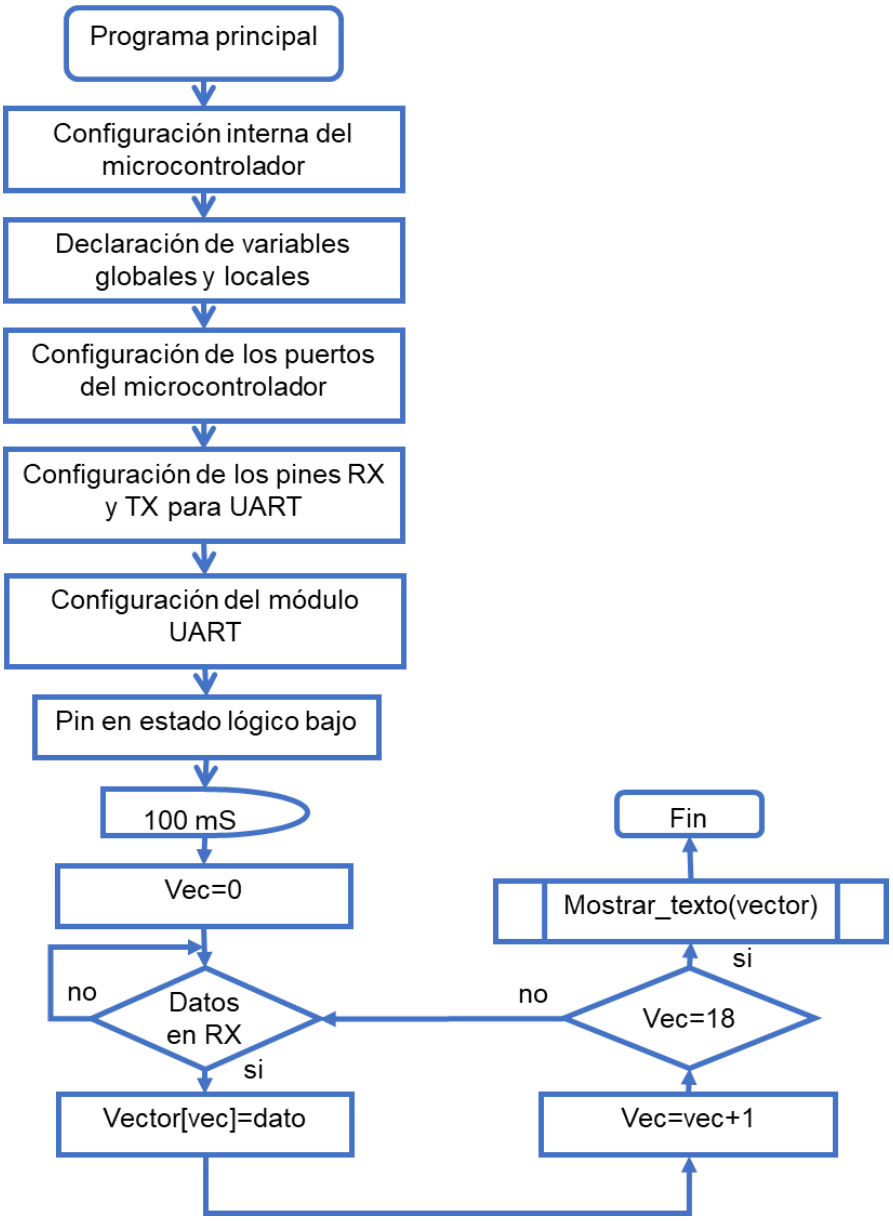

Fig. 36 Diagrama de flujo para recepción de datos por módulo UART

3.8 IMÁGENES EN MATRIZ DE 30X33 PIXELES

En la sección 4.3.2 hablamos de como mostrar pequeños gráficos en la matriz de LEDs de 8x5 pixeles, mediante el almacenamiento de los pixeles del gráfico en la memoria del microcontrolador. Esta no es la mejor manera para almacenar imágenes, debido a que esta vez el número de pixeles es considerablemente superior. Almacenar la imagen en la memoria del microcontrolador tampoco es muy buena idea, ya que, si se desea mostrar una imagen diferente implica que el microcontrolador debe ser programado nuevamente.

Una imagen digital RGB se puede definir como la combinación de tres matrices con el mismo número de filas y columnas que los pixeles de los que se compone dicha imagen. Cada matriz representa un color primario (rojo, verde, azul), los elementos de las matrices representan los niveles de brillo y color que deben contener los pixeles para mostrar la imagen digital.

Se necesita implementar una herramienta que tenga la habilidad de leer los datos correspondientes a una imagen digital RGB de una manera sencilla. Esta herramienta debe ser capaz de escalar la imagen de su escala original a una escala menor que sea compatible con la matriz de LEDs de 30x33 pixeles, y debe de realizar la transmisión de los datos de la imagen al microcontrolador a través de una comunicación serie.

Una interfaz gráfica de usuario (GUI) otorga las habilidades que necesitamos para leer una imagen y después aplicar una técnica de escalamiento para reducir la imagen, también nos ayuda a transferir los datos de una imagen de una PC hacia el microcontrolador para que este pueda enviar la señal de control y que la matriz logre mostrar la imagen deseada.

3.8.1 INTERFAZ GRAFICA DE USUARIO

La interfaz gráfica debe contar con un botón que permita seleccionar la imagen que deseamos enviar hacia la matriz de LEDs RGB. Al seleccionar la imagen deseada, la interfaz debe de mostrar la imagen con su tamaño original y también debe mostrar la imagen escalada a la resolución de la matriz de LEDs (30 filas y 33 columnas).

La interfaz también debe disponer de botón para enviar los datos de la imagen escalada al microcontrolador, así mismo, tendrá un botón que le indique al microcontrolador que la matriz de LEDs debe ser limpiada (apagar todos los pixeles).

Por último, la interfaz debe ser capaz de realizar una conexión entre la PC y el microcontrolador a través de una comunicación por puerto serie, para esto necesitamos que la GUI cuente con un elemento que nos permita escoger la velocidad de transmisión, seleccionar el puerto serie a utilizar y un botón que establezca la conexión.

Para la programación de la interfaz gráfica de usuario se eligió usar el software MATLAB, ya que este cuenta con una herramienta para el diseño y desarrollo interactivo de interfaces graficas de usuario llamada GUIDE.

El editor GUIDE permite construir interfaces arrastrando y soltando componentes en el área de diseño de la GUI. Posteriormente, el usuario debe programar las instrucciones que deben realizar cada uno de los componentes que forman parte de la interfaz gráfica de usuario mediante la escritura de código en el compilador de MATLAB.

El entorno de desarrollo para interfaces gráficas GUIDE y sus elementos se observan en la Fig. 37.

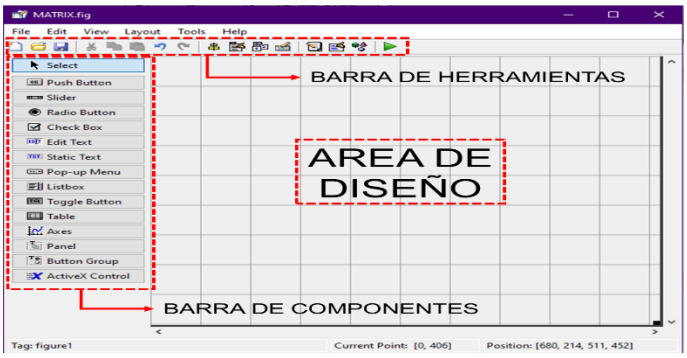

Fig. 37 Entorno de desarrollo GUIDE

Después de trabajar en el diseño de la interfaz gráfica, se presenta el siguiente diseño:

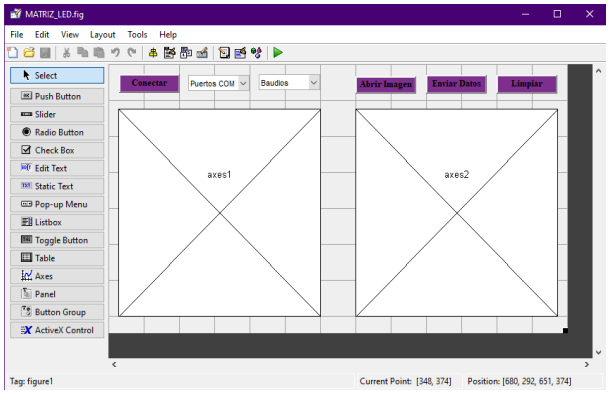

Fig. 38 Diseño de GUI para transmitir datos de imagen digital

Las características de la interfaz gráfica se muestran en la Fig. 1Fig. 39.

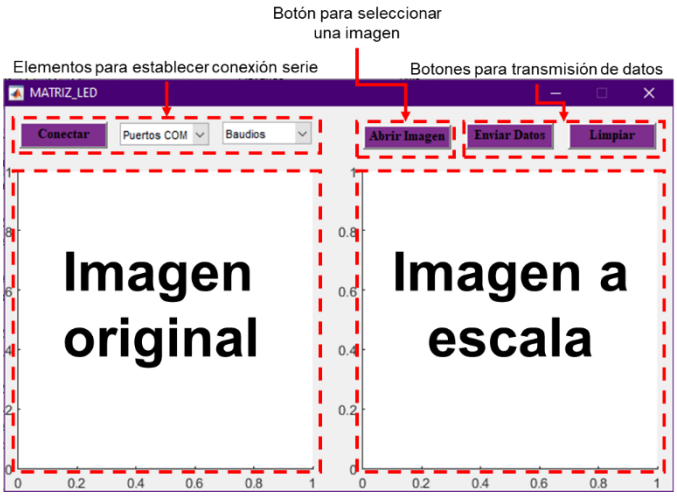

Fig. 39 Características de la interfaz gráfica

3.8.2 FUNCIONAMIENTO DE INTERFAZ GRÁFICA

El algoritmo de la interfaz gráfica se muestra en el siguiente diagrama de flujo:

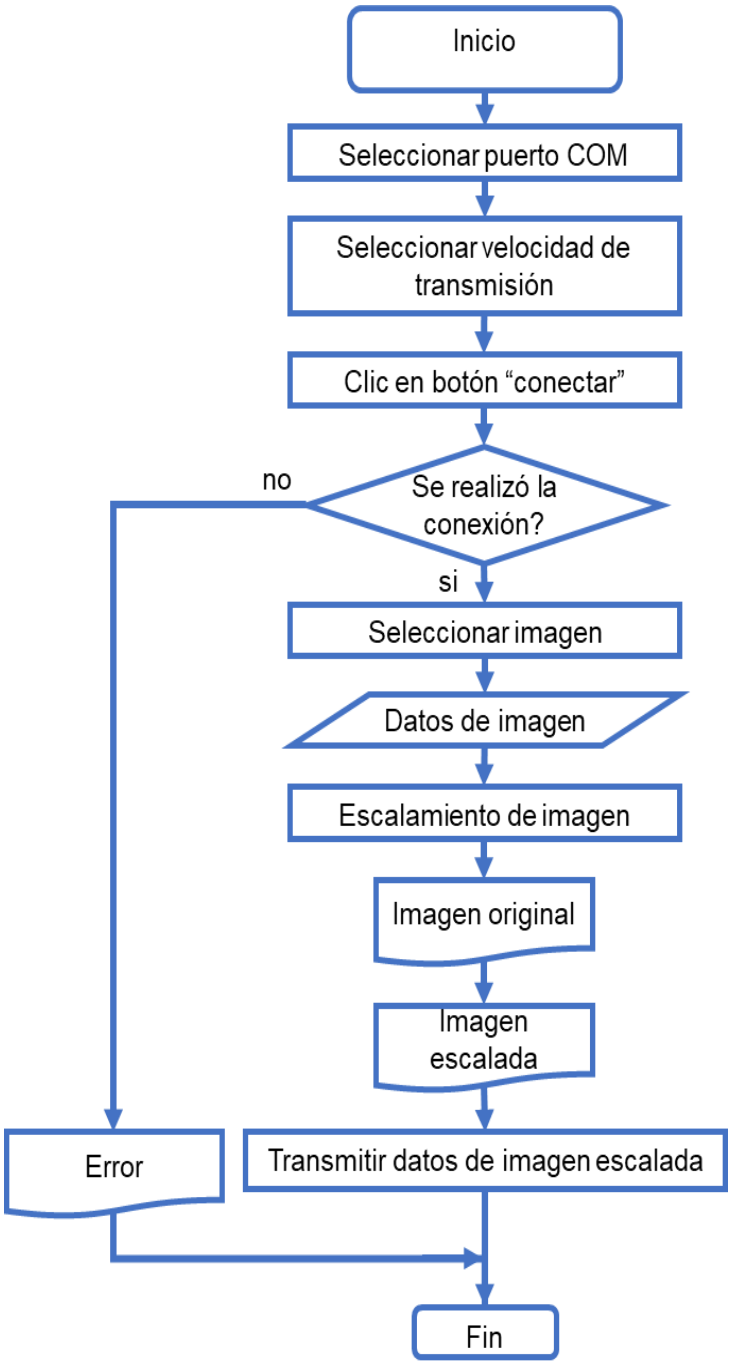

Fig. 40 Diagrama de flujo para el funcionamiento de interfaz gráfica

3.8.3 ALGORITMO PARA RECEPCIÓN DE DATOS DE IMAGEN

Para que el microcontrolador pueda recibir los datos correspondientes a una imagen, primero se tiene que realizar las configuraciones internas y la configuración del módulo UART. Después, el microcontrolador debe de monitorear continuamente se existen datos por recibir a través del pin RX del módulo UART. Si el primer dato recibido es igual a 1 entonces significa que se tiene que limpiar la matriz de LEDs (apagar todos los LEDs), si el dato recibido es diferente de 1 significa que se recibirán los datos de una imagen y deben ser almacenados en un vector, después se realiza la conversión y envío de datos a la matriz de LEDs con ayuda del algoritmo NeoDraw (Fig. 16). En el siguiente diagrama de flujo se muestra el algoritmo para la recepción de los datos de una imagen en el microcontrolador.

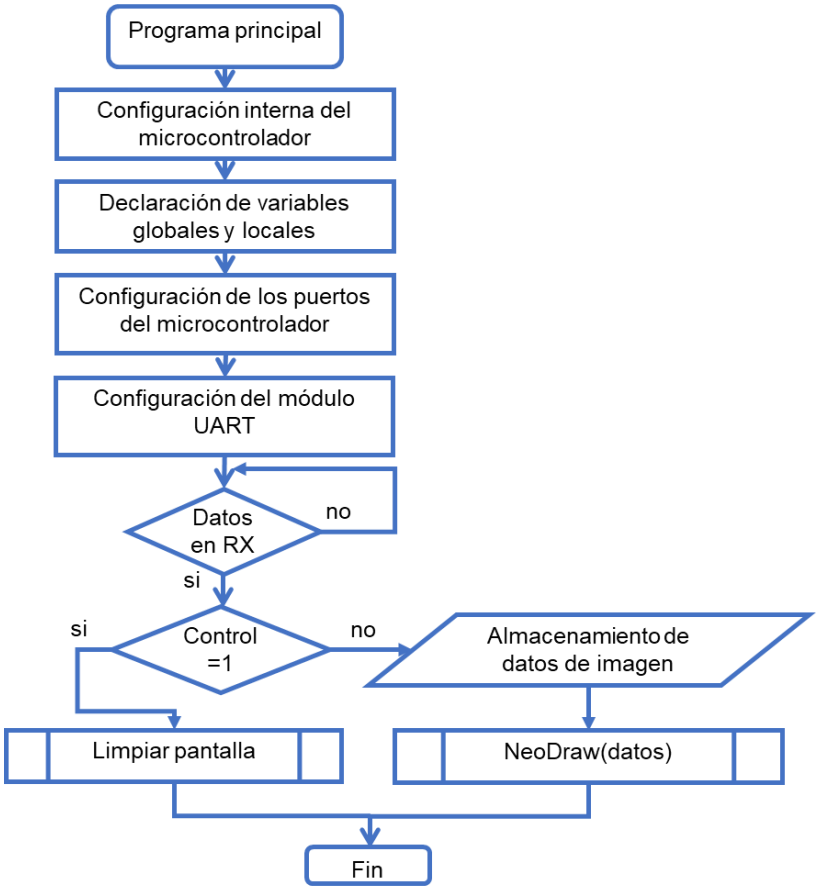

Fig. 41 Diagrama de flujo para algoritmo de recepción de datos de imagen

CAPITULO 4

RESULTADOS

4.1 CONTROL DE TIRA LED WS2812B.

Como resultado de las pruebas realizadas a la tira con tan solo 5 LEDs WS2812B presento las siguientes imágenes.

En la Fig. 42 se observa al PIC18F4550 realizando el control de cada uno de los LEDs WS2812B que forman parte de la tira LED. Los algoritmos Neobit (Fig. 15) y NeoDraw (Fig. 16) funcionan de manera correcta, lo que nos permite que cada LED de la tira tenga diferente color.

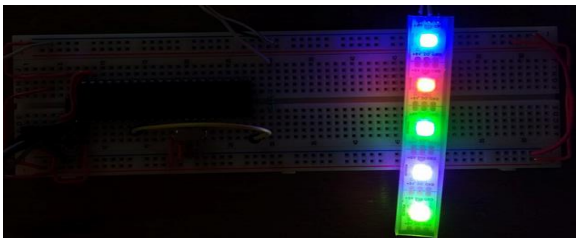

Fig. 42 Tira LED WS2812B y PIC18F4550

En la Fig. 43 se observa nuestra placa de desarrollo (Fig. 13) y la tira de LEDs WS2812B. Aunque es un microcontrolador totalmente diferente, la lógica de los algoritmos NeoBit y NeoDraw es examante la misma, lo que permite obtener los resultados del caso anterior al controlar la tira de LEDs WS2812B.

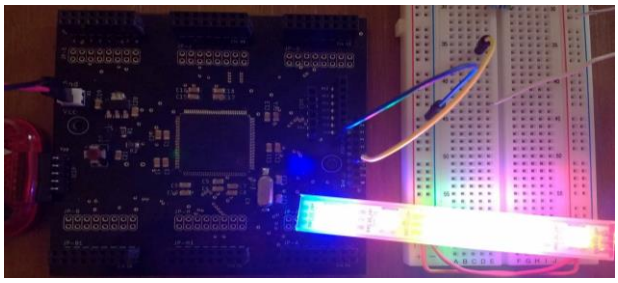

Fig. 43 Tira LED WS2812B y placa de desarrollo

Debido a que el sensor de la cámara con la que fueron tomadas las fotografías se satura, el centro de los LEDs se visualiza de color blanco.

4.2 MATRIZ DE LEDs WS2812B DE 8x5 PIXELES.

4.2.1 GRÁFICOS

Como resultado de la prueba realizada en la matriz de LEDs de 8x5 pixeles para la proyección de pequeños gráficos, presento la Fig. 44, la cual contiene 6 gráficos proyectados en la matriz de LEDs de 8x5 pixeles. Los gráficos con incisos a, b, c y d dan la impresión de ser caracteres tipo ASCII (0, 1, 2, A), pero en realidad son solo elementos de un arreglo unidimensional (vector) colocados de forma estratégica. En el gráfico del inciso c se muestra el gráfico de la bandera de México y en el inciso d el gráfico de una persona (muñeco).

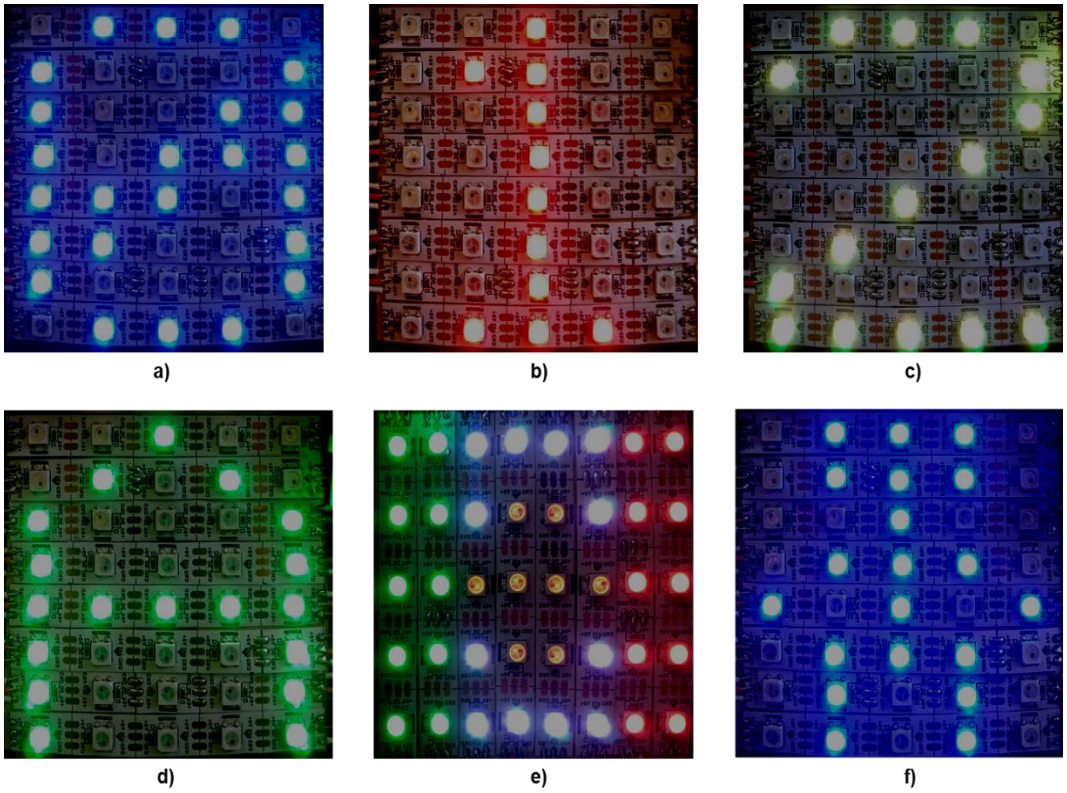

Fig. 44 Gráficos mostrados en matriz de LEDs WS2812B de 8x5 pixeles

4.2.2 DESPLAZAMIENTO DE CARACTERES DE 8X5 PIXELES

Como resultado de la implementación del algoritmo para la generación y desplazamiento de caracteres ASCII (Fig. 25) en la matriz de LEDs de 8x5 pixeles, presento la Fig. 45.

En el inciso a) de la Fig. 45 se observa el carácter “M”, en el inciso b) se muestra el carácter “o”. Este par de caracteres forman parte de la cadena “Hola Mundo”, que se encuentra realizando un efecto de desplazamiento de caracteres.

Las fotografías de la Fig. 45 fueron tomadas en el momento en que el efecto de desplazamiento de los caracteres los colocaba en el centro de la matriz de LEDs de 8x5 pixeles.

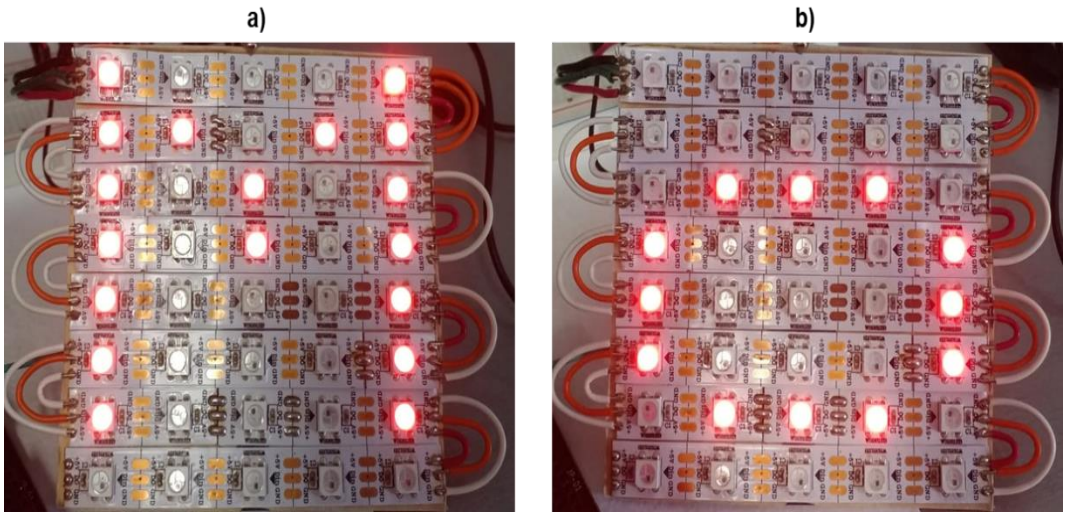

Fig. 45 Caracteres tipo ASCII en matriz LED de 8x5 pixeles.

4.3 MATRIZ DE LEDs WS2812B DE 30x33 PIXELES

4.3.1 DESPLAZAMIENTO DE CARACTERES

Como resultado de la primera prueba realizada a la matriz de LEDs de 30x33 pixeles, la cual consistió en el desplazamiento de una cadena de caracteres en la que se utilizó el algoritmo de la Fig. 25 al que se le realizaron modificaciones para que funcionara en el formato de 30x33 pixeles, presento la Fig. 46.

Los incisos a, b y c de la Fig. 46 muestran el desplazamiento de una cadena de caracteres que está formada solo por los números del 0 al 9. Los incisos d, e y f corresponden al desplazamiento de una cadena de caracteres formada por todas las letras del abecedario (de la “a” a la “z”).

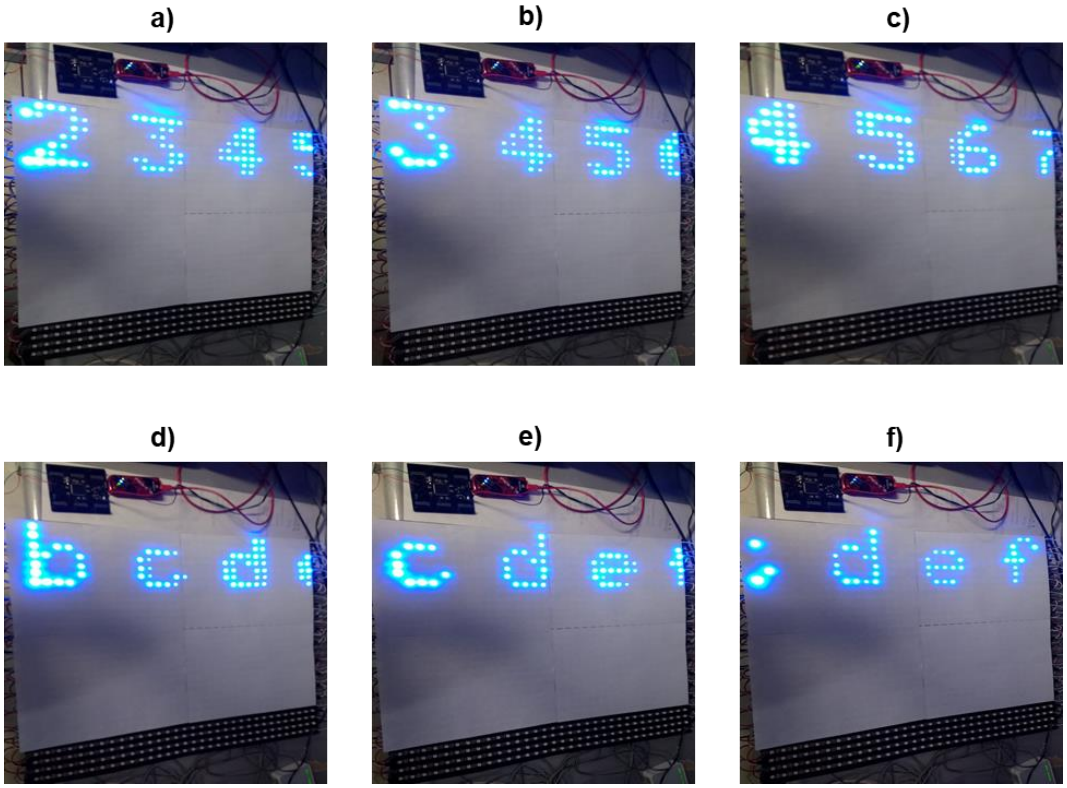

Fig. 46 Desplazamiento de caracteres en matriz de LEDs de 30x33 pixeles.

4.3.2 CARACTERES ESTATICOS

Como resultado de las pruebas realizadas para la proyección de caracteres estáticos en la matriz de LEDs de 30x33 pixeles utilizando el algoritmo de la Fig. 33, presento la Fig. 47.

En la Fig. 47 se observa el texto “CIICAP MICA UAEM” que se encuentra separado en tres renglones “virtuales” pero en realidad los caracteres están almacenados en la memoria del microcontrolador como un arreglo bidimensional (matriz) que simula un solo reglón “virtual” como se muestra en la Fig. 32.

Recordemos que el algoritmo generador de caracteres (Fig. 33), genera los caracteres con un tamaño de 8x5 pixeles, lo que significa que incluso el carácter correspondiente al “espacio” también posee esas dimensiones lo que explica por qué las letras “MICA” y “UAEM” no se encuentran alineadas con las letras “CIICAp”, ya que el algoritmo proyecta un renglón pero estructurado en tres renglones virtuales.

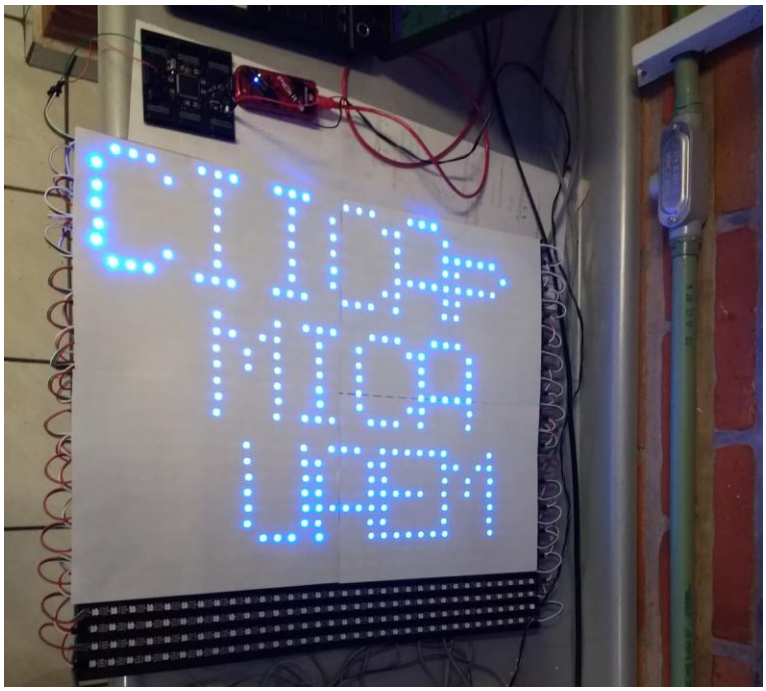

Fig. 47 Texto en matriz de LEDs de 30x33 pixeles

4.3.3 RECEPCIÓN DE TEXTO POR MÓDULO UART

Como resultado de las pruebas para recepción de una cadena de caracteres a través del módulo UART del microcontrolador implementando el algoritmo de la Fig. 33, presento la Fig. 48.

En la Fig. 48 se observa marcada con un recuadro azul la matriz de LEDs de 30x33 pixeles con el mensaje “17:00H NOV 29 2018” que corresponde al día y la hora en la que fue realizada la prueba correspondiente. El recuadro de color verde contiene un error causado por una falla del algoritmo para la recepción de datos seriales. El recuadro color rosa encierra a la placa de desarrollo junto con un programador de microcontroladores PIC PICkit3. En el recuadro color amarillo se muestra el circuito conversor de conexión USB a niveles de voltaje TTL para comunicación serial. El recuadro color rojo señala el computador con un software monitor serie.

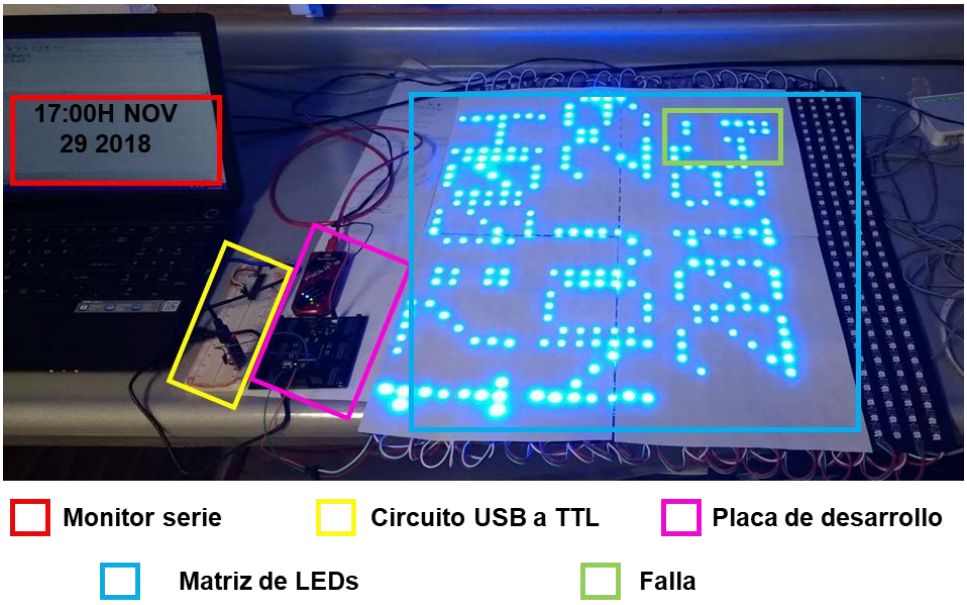

Fig. 48 Matriz de LEDs con recepción de datos por puerto serie.

4.4 INTERFAZ GRÁFICA PARA TRANSMISIÓN DE IMÁGENES

Como resultado de las pruebas realizadas con la interfaz gráfica de usuario hecha con la herramienta GUIDE del software MATLAB para la transmisión de los datos de una imagen escalada y el algoritmo de la Fig. 41 para la recepción de los datos transmitidos por la interfaz gráfica, presento la Fig. 49 y la Fig. 50.

En la Fig. 49 se observa la interfaz gráfica de usuario, se ha seleccionado el puerto serie COM4, una velocidad de transmisión de 1,250,000 baudios y se ha realizado la conexión de manera correcta. Del lado izquierdo se observa el logotipo del CIICAp en su tamaño original y de lado derecho se muestra también el logotipo del CIICAp pero escalado a unas dimensiones de 30x33 pixeles.

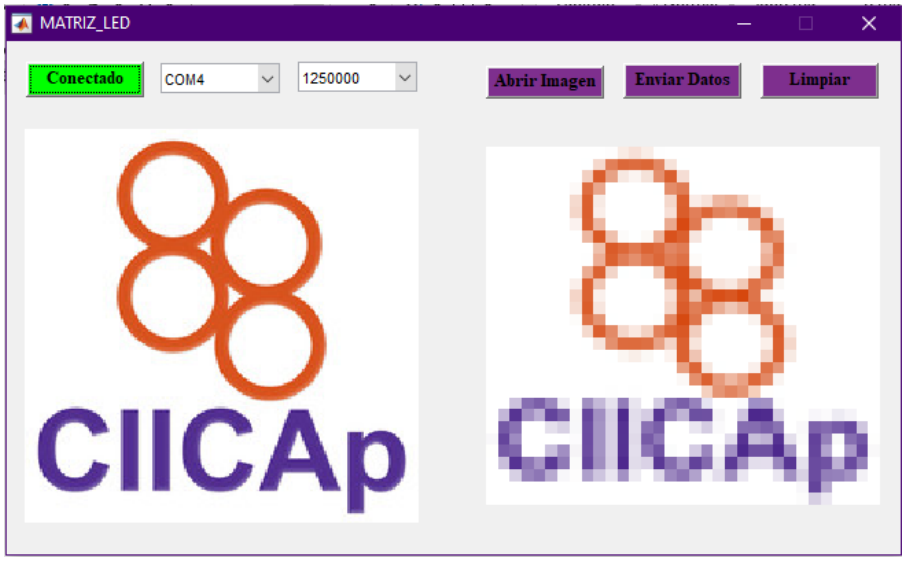

Fig. 49 Interfaz gráfica para la transmisión de imágenes por puerto serie.

La Fig. 50 contiene 4 imágenes proyectadas en la matriz de LEDs de 30x33 pixeles. Los datos de las imágenes son transmitidos por la interfaz gráfica de la Fig. 49, la recepción de los datos y la proyección de las imágenes en la matriz es gracias al algoritmo de la Fig. 41.

El inciso a) de la Fig. 50 corresponde al logotipo del Centro de investigación en ingeniería y ciencias aplicadas (CIICAp).

Los incisos b), c) y d) son tres imágenes muy utilizadas en el estudio del procesamiento digital de imágenes.

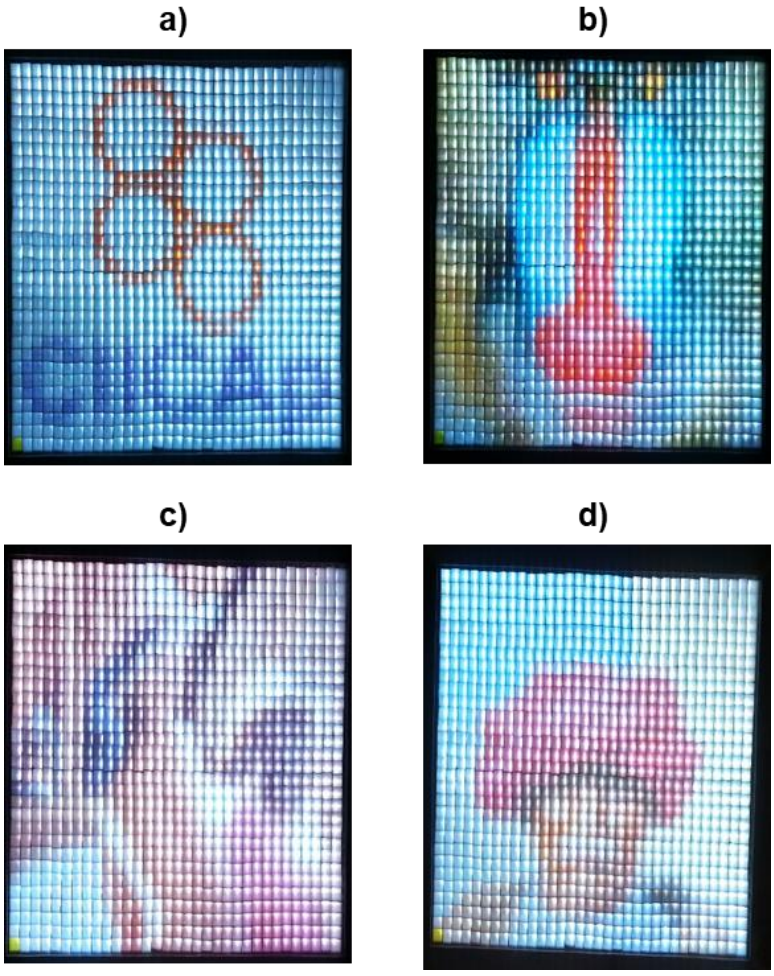

Fig. 50 Imágenes proyectadas en matriz de LEDs WS2812B de 30x33 pixeles. a) Logotipo CIICAp. b) Mandril. c) Lena. d) Árabe.

CAPITULO 5

CONCLUSIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES

En esta tesis, se ha realizado el diseño y construcción de una matriz de LEDs RGB con driver WS2812B de 30x33 pixeles para la proyección de imágenes y texto alfanumérico utilizando un microcontrolador PIC32MZ1024EFK144 como interfaz de control.

En comparación con los módulos matriz LED RGB que dominan el mercado, la matriz de LEDs WS2812B tiene las siguientes ventajas:

- La interfaz de control es de solo 3 conductores, 1 para señal de datos y 2 para alimentación (5V y GND).
- Los LEDs WS2812B tienen la capacidad de retener los datos transmitidos, lo que evita la necesidad de usar técnicas de barrido para contener los datos de la matriz de LEDs.
- Al no usar técnicas de barrido, como la multiplexación, los pixeles de la matriz de LEDs pueden alcanzar mayores niveles de brillo y de color. También evita que el microcontrolador se encuentre trabajando en el barrido de los datos.
- Es posible construir una pantalla flexible, es decir, dependiendo de la base de los LEDs, se puede construir una pantalla flexible, enrollable o curva.

Continuando con la misma comparación, las desventajas son las siguientes:

- Al no usar técnicas de barrido y alcanzar mayores niveles de brillo, el consumo de corriente es 10 veces superior.
- Debido a los tiempos requeridos por el protocolo de transmisión de datos para los LEDs WS2812B, solo se pueden controlar 990 LEDs para reproducir animaciones con una tasa de 30 cuadros por segundo.

- Ya que la transmisión de los datos entre los LEDs WS2812B es en serie, si uno de los LEDs de la matriz se daña, la comunicación se interrumpe provocando que la matriz resulte inoperable, hasta el momento en que el LED dañado sea reemplazado.

Las medidas de la matriz de LEDs (30x33 pixeles) se pensaron para que se proyectaran animaciones o videos a 30 cuadros por segundo. Se realizaron pruebas de animaciones usando el software “JINX”, que es un software especializado para el control de matrices LED, con resultados poco favorables debido a una falla en la sincronía de comunicación serial entre el software “JINX” y el microcontrolador.

Si el diseño de la matriz de LEDs sea solo para proyectar texto alfanumérico, la estructura tendría otra configuración, por ejemplo 8x120 pixeles, lo que permitiría proyectar una cadena de 24 caracteres.

Si la matriz de LEDs se usara solo para proyectar imágenes, es posible armar una matriz con una resolución de 150x198 pixeles, que seria el equivalente a tener 30 matrices LED de 30x33 pixeles. Con estas dimensiones, una imagen tardaría 1 segundo en proyectar los datos correspondientes a 29700 LEDs.

5.2 TRABAJO FUTURO

Implementar un módulo lector de tarjetas SD (o microSD), para almacenar las imágenes en el interior de la memoria SD y que el microcontrolador solo realice la lectura de los datos de las imágenes previamente ordenados según el protocolo del LED WS2812B, así mismo, almacenar un conjunto de imágenes correspondientes a los cuadros por segundo de una animación para que estos sean proyectados en la matriz de LEDs.

Desarrollar mejores algoritmos del microcontrolador para el control y transmisión de los datos de los LEDs WS2812B, tanto para manejo de tiempos como para mejorar la sincronía en la comunicación serial con software especializado para el control de matrices LED, como puede ser el software “JINX”.

Desarrollar algoritmos para controlar una matriz de LEDs de mejor resolución tanto para caracteres, imágenes, animaciones y videos.

Bibliografía

Ahn, J.-H. (2013). Implementationof an LED tile controller for high-quality image display. *Displays*, 17-26.

Alfonso Gago Calderón, J. F. (2012). *Iluminación con tecnología LED.* Paraninfo.

Arellano, M. A. (2013). *MATLAB & SIMULINK para ingeniería.*

AVAGO, T. (2013). *Introduction to driving LED matrices.*

Breijo, E. G. (2008). *Compilador C CCS y simulador PROTEUS para microcontroladores PIC.* Alfaomega.

Enrique Mandado Pérez, L. M. (2007). *Microcontroladores PIC. Sistema integrado para el autoaprendizaje.* Barcelona: MARCOMBO.

Ibrahim, D. (2006). *Microcontroller based applied digital control.* John Wiley & Sons, Ltd.

Jesús Javier Rodriguez Sala, L. S. (2003). *Introducción a la programación. Teoría y práctica.* Club universitario.

Kurdthongmee, W. (2005). Desing and implementation of a FPGA-based multiple-colour LED display board. *Microprocessors and Microsystems*, 327-336.

MICROCHIP. (2005). *dsPIC30F2010 Datasheet.*

MICROCHIP. (2013). *dsPIC33EP64MC202 Datasheet.*

MICROCHIP. (2016). *PIC32MZ1024EFK144 Datasheet.*

Palacios, E., Remiro, F., & López, L. (2004). *Microcontrolador PIC16F84 Desarrollo de proyectos.* ALFAOMEGA.

Petr Olivka, J. H. (2015). Microcontroller peripheral mapping used to control RGB LED panel. *IFAC*, 436-441.

Rafael C. Gonzalez, R. E. (2002). *Digital Image Processing Second edition.* Prentice Hall.

Susan Walsh Sanderson, K. L. (2014). Light emitting diodes and the lighting revolution: The emergence of a solid-state lighting industry. *Research policy*, 1730-1746.

Wahyono, K. J. (2015). LED dot matrix text recognition method in natural scene. *Neurocomputing*, 1033-1041.

Worldsemi. (2019). *WS2812B Datasheet.*

Zalesinska, M. (2018). The impact of luminace, size and location of LED billboards on drivers visual performance. *Accident Analysis and Prevention*, 439-448.