

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

**FACULTAD DE INGENIERÍA MECÁNICA, ELECTRÓNICA Y
AMBIENTAL**

**CARRERA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES**



**“DISEÑO, PROGRAMACIÓN E IMPLEMENTACIÓN DE UN
PROTOTIPO DE PANEL LED RGB PARA LA TRANSMISIÓN DE
MENSAJES Y DATOS UTILIZANDO LA PLACA RASPBERRY PI 3
PARA APLICARLO A UN SISTEMA DE INFORMACIÓN EN LA
UNTELS”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES

PRESENTADO POR EL BACHILLER

MAYTA CASAS, JUAN MIGUEL

**Villa El Salvador
2016**

DEDICATORIA

A mis padres por su apoyo
Incondicional en mi formación
profesional y por todo su cariño
y amor que siempre me han brindado

AGRADECIMIENTO

Agradezco a mis padres que me dieron todo su apoyo tanto económico como moral en la culminación de este proyecto, a mi familia que siempre estuvo dándome una mano cuando mas lo necesitaba, a mis profesores que con sus enseñanzas me permitieron desarrollarme como persona y como profesional

ÍNDICE

	Pág.
INTRODUCCIÓN.....	xi
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	
1.1. Descripción de la Realidad Problemática.....	14
1.2. Justificación del Proyecto.....	16
1.3. Delimitación del Proyecto.....	17
1.4. Formulación del Problema.....	18
1.5. Objetivos.....	19
1.5.1. Objetivo General.....	19
1.5.2. Objetivos Específicos.....	19
CAPÍTULO II: MARCO TEÓRICO	
2.1 Antecedentes de la Investigación.....	22
2.1.1 Ámbito Nacional.....	22
2.1.2 Ámbito Internacional.....	22
2.2 Bases Teóricas.....	25
2.2.1 PWM (Modulación por Ancho de Pulso)	25
2.2.2 Modelo de color RGB.....	26
2.2.2.1 Percepción y sensación de color.....	27
2.2.2.2 Tratamiento de la señal de video RGB.....	28
2.2.3 Profundidad de color.....	29
2.2.3.1 Color real o true color.....	29
2.2.4 Matriz LED RGB 8x8 ánodo común (D23088A/B)	30
2.2.5 Raspberry Pi 3 modelo B.....	32
2.2.6 IC 74HC245.....	35
2.2.7 IC 74HC595.....	36

2.2.8 IC 74HC138.....	38
2.2.9 IC 74HC04.....	40
2.2.10 Transistor BD135.....	40
2.2.11 IC ULN2803.....	41
2.3 Marco Conceptual.....	44
2.3.1 Frecuencia de refresco.....	44
2.3.2 Pixel Pitch	44
2.3.3 Pixel	44
2.3.4 Scanning mode	45
2.3.5 Decodificador / Demultiplexor.....	45

CAPÍTULO III: DISEÑO Y DESCRIPCIÓN DEL SISTEMA

3.1 ANÁLISIS DEL SISTEMA.....	48
3.2 DISEÑO DEL SISTEMA.....	53
3.2.1 Descripción del diseño.....	53
3.2.2 Funcionamiento del prototipo de panel LED RGB 16x32.....	60
3.2.3 Diseño alternativo de prototipo de panel LED RGB 16x32.....	73
3.2.4 Programación del panel LED RGB 16x32.....	79
3.2.5 Implementación del panel LED RGB 16x32.....	85
3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS.....	88
3.3.1 Resultados de la implementación del prototipo de panel LED.....	88
3.3.2 Aplicación a un sistema de información en la UNTELS.....	94
3.3.3 Análisis económico de un sist. de información en la UNTELS.....	96
CONCLUSIONES.....	101

RECOMENDACIONES.....	102
REFERENCIAS.....	104
BIBLIOGRAFÍA.....	107
ANEXOS.....	110

LISTADO DE FIGURAS

	Pág.
FIGURA 2.1: Generación de una señal PWM	25
FIGURA 2.2: Modelo de color RGB	26
FIGURA 2.3: Imagen en diferentes profundidades de color	30
FIGURA 2.4: Matriz LED RGB 8X8 ánodo común	31
FIGURA 2.5: Patillaje del módulo de matriz LED RGB 8X8.....	31
FIGURA 2.6: Microcomputadora RASPBERRY PI 3 modelo B	32
FIGURA 2.7: Chip BCM2837 ARMv8 Cortex-A53 de 64 bit	33
FIGURA 2.8: Pinout y diagrama de conexión del 74HC245.....	35
FIGURA 2.9: Diagrama funcional del 74HC595.....	37
FIGURA 2.10: Diagrama lógico del 74HC595.....	37
FIGURA 2.11: Pinout del 74HC595.....	38
FIGURA 2.12: Diagrama lógico y pinout del 74HC138.....	39
FIGURA 2.13: Pinout del 74HC04.....	40
FIGURA 2.14: Transistor BD135.....	41
FIGURA 2.15: Diagrama funcional del ULN2803.....	42
FIGURA 2.16: Diagrama lógico y pinout del ULN2803.....	43
FIGURA 2.17: Decodificador 2 x 4.....	46
FIGURA 3.1: Arquitectura de un sistema de paneles LED RGB	48
FIGURA 3.2: Paneles LED RGB unidos formando una pantalla más grande...49	
FIGURA 3.3: Display LED RGB formado por paneles LED RGB modulares ...50	
FIGURA 3.4: Panel con y sin la tecnología Virtual Pixel	51
FIGURA 3.5: Prototipo propuesto de panel LED RGB 16x32.....	52
FIGURA 3.6: Resultados finales obtenidos en paneles LED RGB	52
FIGURA 3.7: Generación de colores en el panel LED RGB 16x32.....	53
FIGURA 3.8: Ejemplo de panel LED RGB de mayores dimensiones.....	54
FIGURA 3.9: Microcomputadora Raspberry Pi 3.....	56
FIGURA 3.10: Escritorio del S.O. Raspbian instalado en la Raspberry Pi 3.....	57

FIGURA 3.11: Multiplexación de las líneas de dirección (filas).....	58
FIGURA 3.12: FPGA SpartanIIe de Xilinx.....	59
FIGURA 3.13: Esquema del 74HC138 con salidas hacia los MOSFET's.....	61
FIGURA 3.14: Panel LED RGB 16x32, mostrando los 4 cuadrantes.....	63
FIGURA 3.15: Panel LED RGB 16x32, compuesto por 8 matrices LED 8x8....	64
FIGURA 3.16: Diagrama indicando la operación de los registros.....	65
FIGURA 3.17: Conector HUB75 mostrando las señales de entrada.....	67
FIGURA 3.18: Conexión de los pines GPIO de la placa Raspberry Pi 3....	68
FIGURA 3.19: Esquema del HUB75 conectado hacia los IC's 74HC245.....	68
FIGURA 3.20: Señales R1, G1, B1 ingresando hacia los 3 primeros IC's.....	69
FIGURA 3.21: IC MBI5026 mostrando su conexión con las columnas 1-16....	70
FIGURA 3.22: Distribución de las señales R1, G1, B1, CLK, ST	70
FIGURA 3.23: Esquema del conexionado de las entradas de direcciones.....	71
FIGURA 3.24: Salida digital de tipo colector abierto y sinking current.....	73
FIGURA 3.25: Esquema del conexionado de los IC's 74HC595.....	74
FIGURA 3.26: Esquema del conexionado de los 24 IC's 74HC595.....	75
FIGURA 3.27: Señales de data (R1, G1, B1, R2, G2, B2) invertidas	76
FIGURA 3.28: Esquema del conexionado de los transistores BJT BD135.....	77
FIGURA 3.29: Esquema del conexionado del IC 74HC138.....	78
FIGURA 3.30: Conexión necesario para un solo módulo de matrix LED....	79
FIGURA 3.31: Pi Recovery con distintas opciones de sistemas operativos ...	80
FIGURA 3.32: Menú principal del S.O. que muestra las opciones de setup.....	81
FIGURA 3.33: Escritorio de Raspbian en modo gráfico y modo comando.....	82
FIGURA 3.34: Creación del directorio display 16x32 en el terminal LX.....	83
FIGURA 3.35: Creación de una imagen utilizando el software GIMP	84
FIGURA 3.36: Fase inicial de la implementación del prototipo.....	85
FIGURA 3.37: Pruebas iniciales en el prototipo de panel LED RGB 16x32.....	86
FIGURA 3.38: Mi persona trabajando en la programación del prototipo.....	87
FIGURA 3.39: Resultados finales del prototipo de panel LED RGB 16x32.....	88
FIGURA 3.40: Mensajes en movimiento en el prototipo de panel LED RGB....	89

FIGURA 3.41: Logotipo de la universidad UNTELS mostrado en el panel.....	89
FIGURA 3.42: Logotipo de la placa Raspberry Pi mostrado en el panel.....	90
FIGURA 3.43: Texto de 16x150 pixeles creado usando el programa GIMP.....	91
FIGURA 3.44: El mismo mensaje anterior siendo desplazado en el prototipo..	91
FIGURA 3.45: Visualización de la pantalla del monitor en el prototipo.....	92
FIGURA 3.46: Visualización de diferentes imágenes en el prototipo.....	93
FIGURA 3.47: Esquema que muestra la estructura de una pantalla LED.....	94
FIGURA 3.48: Esquema de un sistema de información para la UNTELS.....	95
FIGURA 3.49: Diferentes tamaños propuestos de sist. de información.....	100

LISTADO DE TABLAS

	Pág.
TABLA 2.1: Ocho colores y sus valores HSL y RGB asociados	27
TABLA 2.2: Especificaciones técnicas y comparación de Raspberry pi.....	34
TABLA 2.3: Tabla de verdad del 74HC138.....	39
TABLA 3.1: Especificaciones Técnicas de la placa Raspberry Pi 3.....	56
TABLA 3.2: Tabla de verdad del 74HC138.....	60
TABLA 3.3: Descripción del pinout del conector de entrada HUB75.....	66
TABLA 3.4: Análisis económico de un panel LED RGB de 16x32.....	96
TABLA 3.5: Análisis económico de un sist. de información de 1m x 0.5m.....	97
TABLA 3.6: Análisis económico de un sist. de información de 2m x 1m.....	98
TABLA 3.7: Análisis económico de un sist. de información de 4m x 2m.....	99

INTRODUCCIÓN

En la actualidad, se mueven grandes cantidades de dinero debido a la economía proveniente del comercio de diversos productos o servicios, es así que las empresas proveedoras de éstos, recurren a la publicidad como medio de llegada a los consumidores para poder comercializar la mayor cantidad de sus productos.

Dentro de todas las formas de publicidad, una de las más interesantes es la publicidad visual, puesto que permanece las 24 horas del día y los 365 días del año, no contamina el medio ambiente y se puede personalizar orientándolo hacia el público objetivo. Sin embargo, para lograr una buena percepción de la publicidad, es necesaria una combinación de los siguientes factores: buen diseño del anuncio, una correcta iluminación y finalmente una interacción de este sobre el primero.

Así, en el presente trabajo se pretende diseñar un panel led RGB que sirva como sistema de información para la UNTELS, de tal manera que se pueda desplegar información relacionada a asuntos académicos de la universidad tales como horarios, anuncios, avisos importantes, reuniones, fechas de exámenes programadas, eventos, etc. Además, mediante la utilización de los LED's RGB se pueden generar una gran gama de colores (más de 16,7 millones de colores diferentes con una profundidad de color de 24 bits) y será mediante estos que se crearán efectos de movimiento y degradé de colores; estos efectos podrán

ser seleccionados por el usuario, así como los colores entre los cuales se desea obtener el degradé y efectos; todo esto con el objetivo de crear un mayor dinamismo en la transmisión de datos, así como atracción de parte del público hacia la información ubicada en el panel.

De esta manera en el capítulo I se abordará el planteamiento del problema, se hará una breve descripción de la realidad problemática, así como la justificación del proyecto sus delimitaciones, la formulación del problema, el objetivo general y específicos.

En el capítulo II se tratará sobre el marco teórico, los antecedentes de la investigación, bases teóricas y marco conceptual

Finalmente, en el capítulo III se tratará sobre el diseño, programación e implementación del prototipo de panel LED RGB para la transmisión de mensajes y datos utilizando la placa Raspberry pi 3 para aplicarlo a un sistema de información en la UNTELS.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

En este primer capítulo, se plantearán los problemas que motivaron para la realización de este proyecto. Primero, se presentará una descripción de la realidad problemática del lugar designado para la ejecución de este proyecto; segundo, se plantearán los diversos objetivos que se quieren alcanzar con la implementación de este proyecto; posteriormente, se pasará a presentar la justificación del proyecto en estudio; para finalmente, evaluar ciertas limitaciones que se presentaron en la implementación del proyecto.

1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA

Actualmente la Universidad Nacional Tecnológica de Lima Sur (UNTELS) se encuentra en proceso de crecimiento continuo, recientemente se han construido y habilitado nuevas áreas de infraestructura para los alumnos tales como un nuevo anfiteatro, un nuevo auditorio, así como nuevos pabellones y laboratorios y mejoras en otros servicios como el comedor, biblioteca etc. Es decir, la universidad se encuentra en constante crecimiento en beneficio de la comunidad estudiantil, profesores, personal administrativo, etc.

En este proceso de mejora continua es que se identifican ciertos problemas que se han venido solucionando como la necesidad de nuevos laboratorios, más salones de clases ante el constante crecimiento de la población estudiantil es decir más infraestructura la cual ya está siendo finalizada y pronto se tendrá la solución a estos problemas pero sin embargo dentro de todos estos problemas se identifica que la UNTELS no cuenta con un sistema de información eficiente para transmisión de información en tiempo real, actualmente los alumnos cuando necesitan alguna información ya sea ubicación de sus aulas, horarios, etc. deben preguntar al personal de seguridad que se encuentra en las instalaciones de la universidad pero muchas veces este personal no se encuentra disponible y los alumnos no saben a quién consultar, sobre todo los nuevos ingresantes (cachimbos) que están recién adaptándose.

A parte de ello si se desea transmitir un comunicado, a la comunidad estudiantil, de diversa índole (fechas de matrícula, horarios, eventos,

ponencias, fechas de exámenes, etc.) estos generalmente se hace con hojas pegadas o fijadas en las vitrinas de informes que se tienen en el campus, pero muchas veces esta manera de mostrar la información no es eficiente ya que el alumno debe desplazarse todos los días o constantemente hasta la vitrina de informes para saber si “hay algo nuevo”, lo cual muchos alumnos no lo hacen ya que están preocupados más en sus estudios y sus clases con lo cual muchos de los mensajes que se quieren transmitir pasan desapercibidos o solo una pequeña parte de la población estudiantil se llega a enterar.

Así mismo también se hace uso de la página web de la UNTELS para la transmisión de todo tipo de información relacionada a la universidad sin embargo no todos los alumnos ingresan a la página web todos los días o muy a menudo para informarse.

Es por ello que se propone un sistema de información eficiente, programable, llamativo (a todo color) y que se encuentre visible a la comunidad estudiantil, que sea visible tanto en el horario de día como de noche y que permita la transmisión de información importante, urgente o relevante a aspectos académicos de la universidad. Este sistema de información son los paneles LED RGB los cuales están formados por matrices de LED's RGB (Red, Green, Blue) los cuales son muy usados para transmitir imágenes y video en espacios publicitarios. Estos paneles son muy costosos inclusive con tamaños pequeños ya que su tecnología (hardware y software) es cerrada es decir no se encuentra disponible al público, muchos

llevan patentes y son implementados por grandes empresas. sin embargo, se diseñará un prototipo (un sistema a escala pequeño) de panel LED RGB que permitirá la transmisión de información y el cual se puede escalar y hacerlo más grande (a las necesidades de la universidad) tan solo con la adición de más módulos de matrices y cambios en el código de programación, así como una fuente de alimentación de más amperaje, etc.)

1.2 JUSTIFICACIÓN DEL PROYECTO

Implementando un sistema de información con paneles LED RGB permitirá mejorar el actual sistema de información que generalmente se hace con hojas pegadas o fijadas en las vitrinas de informes que se tienen en el campus, pero no es eficiente ya que el alumno debe desplazarse todos los días o constantemente hasta la vitrina de informes para saber si “hay algo nuevo”, lo cual muchos alumnos no lo hacen ya que están preocupados más en sus estudios y sus clases con lo cual muchos de los mensajes que se quieren transmitir pasan desapercibidos o solo una pequeña parte de la población estudiantil se llega a enterar.

El sistema con paneles LED RGB permite la transmisión de información de datos (fechas de matrícula, horarios, eventos, ponencias, fechas de exámenes, etc.) en tiempo real y, además, este sistema de información es eficiente, programable, llamativo (a todo color) y se puede ubicar en un lugar visible a la comunidad estudiantil.

Sera visible tanto en el horario de día como de noche y permitirá la transmisión de información importante, urgente o relevante a aspectos académicos de la universidad.

1.3 DELIMITACIONES DEL PROYECTO

1.3.1 CONCEPTUAL Y TEÓRICO

Este prototipo se ha diseñado con la intención de servir como introducción al campo de los sistemas de pantallas LED RGB. Su estructura debería poder permitir desarrollar sistemas comerciales a partir de este prototipo en investigaciones futuras. Si se unen varios de estos prototipos de panel LED RGB en conexión cascada podría llegar a tener la capacidad de mostrar imágenes y video de baja resolución, ya que se trata de un prototipo de panel LED RGB similar a los sistemas de pantallas LED comerciales, si se desea implementar un panel más grande la placa Raspberry pi 3 y el código con su respectiva programación lo soporta.

Este proyecto se basa en la teoría de la modulación por ancho de pulso (PWM), multiplexación y demultiplexación, microcontroladores, sistemas embebidos, etc. El proyecto se delimitará a la realización de un prototipo de panel RGB (exactamente un panel de 16x32 pixeles) de 512 leds RGB (1536 LED's en total) y abarcará desde el conexionado de las matrices LED RGB hasta la configuración, programación de la placa Raspberry pi 3 para que pueda manejar y distribuir los datos en el prototipo de panel LED RGB. El prototipo será capaz de mostrar texto en movimiento en diferentes colores, figuras, formas, etc.

El prototipo se desarrolló enfocándose básicamente en la accesibilidad, flexibilidad y versatilidad. Por ello, se trabajó con especial cuidado en el diseño y minimizar el costo de los componentes, en un esfuerzo de incentivar las futuras investigaciones.

1.3.2 ESPACIAL

Este proyecto se desarrolla en la Universidad Nacional Tecnológica del Cono Sur de Lima (UNTELS) distrito de Villa el Salvador, Lima - Perú.

1.3.3 TEMPORAL

Este proyecto de ingeniería se desarrolló en el periodo de febrero del 2016 hasta julio del 2016.

1.4 FORMULACIÓN DEL PROBLEMA

1.4.1 PROBLEMA GENERAL

¿Cómo se diseña, programa e implementa un prototipo de panel LED RGB para la transmisión de mensajes y datos utilizando la placa Raspberry pi 3 para aplicarlo a un sistema de información en la UNTELS?

1.4.2 PROBLEMAS ESPECÍFICOS

- ¿Cómo se diseña el circuito electrónico con los IC's que harán la función de driver de las matrices LED RGB utilizando el programa PROTEUS 8.4?
- ¿De qué manera se arma e implementa la parte hardware del circuito, de acuerdo al diseño realizado, con los IC's drivers de LED's realizando el conexionado con cada una de las matrices 8x8 y

realizando el montaje de todos los componentes necesarios para el funcionamiento del panel LED RGB?

- ¿De qué manera se programa, compila y depura el código fuente que manejará el funcionamiento y distribución de la data hacia el panel LED RGB con el lenguaje C++ y PYTHON y bajo el entorno del sistema operativo RASPBIAN?
- ¿Cómo se realiza la configuración de la placa RASPBERRY PI 3 con el sistema operativo RASPBIAN e implementa la conexión al circuito controlador y hacia las matrices LED's y prueba el funcionamiento del sistema en conjunto?

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Diseñar, programar e implementar un prototipo de panel LED RGB para la transmisión de mensajes y datos utilizando la placa Raspberry pi 3 para aplicarlo a un sistema de información en la UNTELS.

1.5.2 OBJETIVOS ESPECÍFICOS

- Diseñar el circuito electrónico con los IC's que harán la función de driver de las matrices LED RGB utilizando el programa PROTEUS 8.4.
- Armar e implementar la parte hardware del circuito, de acuerdo al diseño realizado, con los IC's drivers de LED's realizando el conexionado con cada una de las matrices 8x8 y realizar el montaje de todos los componentes necesarios para el funcionamiento del panel LED RGB.

- Programar, compilar y depurar el código fuente que manejara el funcionamiento y distribución de la data hacia el panel LED RGB con el lenguaje C++ y PYTHON y bajo el entorno del sistema operativo RASPBIAN.
- Realizar la configuración de la placa RASPBERRY PI 3 con el sistema operativo RASPBIAN e implementar la conexión al circuito controlador y hacia las matrices LED's y probar el funcionamiento del sistema en conjunto.

CAPÍTULO II

MARCO TEÓRICO

En este segundo capítulo, se mencionarán los antecedentes de la investigación, tanto en el ámbito nacional como en el ámbito internacional, para luego mencionar las bases teóricas en las que se basa este proyecto y para finalmente definir el marco conceptual de cada uno de los componentes y elementos utilizados en el diseño, programación e implementación de este proyecto.

2.1 ANTECEDENTES DEL PROYECTO

2.1.1 Ámbito Nacional

Baldarrago, J.A. (2010). *Diseño de efectos y variación de colores mediante degradé en leds de potencia RGB aplicados a paneles publicitarios*. Tesis. Pontificia Universidad Católica. Perú

En el presente trabajo se realizó el diseño de efectos con colores mediante degradés en una matriz de LED's de potencia RGB orientado a paneles publicitarios, con el objetivo de implementar este sistema de iluminación como una opción frente a los métodos tradicionales, pues estos últimos ofrecen muchas desventajas, tanto en la parte visual como en la parte económica. Además de incluir en el diseño, una interfaz para ser operada por un usuario, de manera fácil e intuitiva, la cual actualmente no es brindada por los equipos comerciales que ofrecen este tipo de iluminación, pues estos ya vienen predeterminados desde su fabricación.

El diseño realizado consta de 2 partes. La primera parte es el desarrollo del software, es decir, el programa cuyo fin es lograr el degradés de los colores, así como los efectos realizados con estos; mientras que la segunda parte es el hardware, lo que incluye tanto los controladores de potencia para los LED's, como los LED's mismos y la tarjeta de control de todo el sistema.

2.1.2 Ámbito Internacional

De Jong, S. (2010). *Investigative development of an accesible LED matrix display*. Tesis. Massey University. Nueva Zelanda.

El principal objetivo de este proyecto es investigar los fundamentos del desarrollo de pantallas o displays LED RGB a través de un prototipo. El prototipo desarrollado en este proyecto también sirve para familiarizarse con las tecnologías relacionadas con las pantallas LED RGB.

El prototipo se desarrolló enfocándose básicamente en la accesibilidad, flexibilidad y versatilidad. Por ello, se trabajó con especial cuidado en mantener el diseño simple y minimizar el costo de los componentes, en un esfuerzo de incentivar las futuras investigaciones.

Este prototipo se ha diseñado con la intención de servir como introducción al campo de los sistemas de pantallas LED RGB. Este prototipo no se ha diseñado con la intención de que rivalice en desempeño con sus similares comerciales, sin embargo, su estructura debería poder permitir desarrollar sistemas comerciales a partir de este prototipo en investigaciones futuras.

Mohd, S. (2014). *Full Color of LED Display System using ARM microcontroller*. Tesis. Universidad Tecnológica de Malasia. Malasia.

En este proyecto, se propone un diseño de pantalla LED usando un microcontrolador ARM de acuerdo con la estructura de las pantallas RGB LED. Este sistema utiliza un microcontrolador ARM LPC1769 que es parte de la familia de microcontroladores LPC17xx ARM Cortex M3 y que se programó con la ayuda del software LPCXpresso de NXP el cual se usó como el IDE (Integrated Development Environment) del microcontrolador para desarrollar el código de programa para el sistema.

El sistema propuesto se implementa con el microcontrolador ARM LPC1769. Además, el diseño del panel LED RGB se utiliza con el fin de

comprobar la funcionalidad del sistema. Este proyecto se implementa utilizando el lenguaje de programación C. Este proyecto ofrece un sistema de pantalla LED con baja disipación de potencia, así como de bajo coste de desarrollo. Por lo tanto, el costo de producción puede ser minimizado.

Agonoy, A. y Cuenco A. (2009). *Programmable 24-bit RGB LED Color Panel via Bluetooth Technology*. Tesis. Mapua Institute of technology. Filipinas.

En este proyecto, se diseña un panel LED RGB basado en microcontrolador el cual es muy útil como una forma de publicidad para las pequeñas empresas. El diseño utiliza la tecnología Bluetooth como medio de comunicación para el usuario y para poder controlar el panel LED RGB. El objetivo principal del proyecto es el diseño de un panel LED RGB que pueda ayudar a las pequeñas empresas en su lucha contra el alto costo de las técnicas de publicidad. Se utiliza microcontroladores y controladores de LED para manipular los colores producidos por el panel LED RGB. El diseño es alimentado por una fuente de 12 voltios y es controlado por un microcontrolador PIC. El diseño utiliza un módulo de conexión Bluetooth para recibir los datos procedentes de los teléfonos móviles, ordenadores portátiles y de ordenadores de escritorio.

El panel se puede conectar con computadoras personales de escritorio a través de un hiper - terminal y que utilice el conector telefónico RJ11.

2.2 BASES TEÓRICAS

2.2.1 PWM (Modulación por Ancho de Pulso)

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

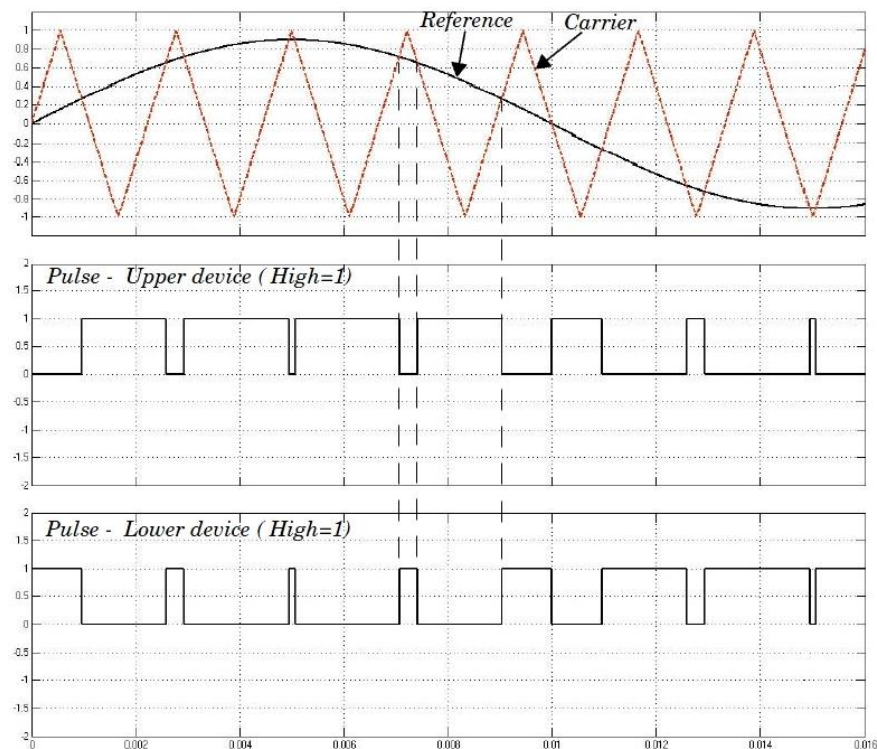


FIGURA 2.1: Generación de una señal PWM

FUENTE: Página Web <http://www.mathworks.com>. Consultada en junio 2016

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período, como se observa en la figura 2.1. La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida. Una de las entradas se conecta

a un oscilador de onda dientes de sierra, mientras que la otra queda disponible para la señal moduladora.

La principal desventaja que presentan los circuitos PWM es la posibilidad de que haya interferencias generadas por radiofrecuencia. Éstas pueden minimizarse ubicando el controlador cerca de la carga y realizando un filtrado de la fuente de alimentación (Malvino y Bates, 2007).

2.2.2 MODELO DE COLOR RGB

RGB es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios. El modelo de color RGB, que se muestra en la figura 2.2 y en la tabla 2.1, no define por sí mismo lo que significa exactamente rojo, verde o azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en distintos dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente (Mandado, 2007).

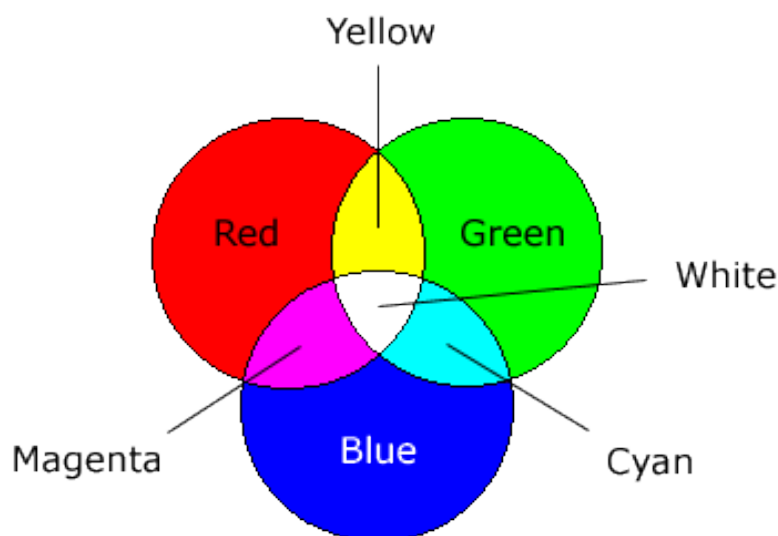


FIGURA 2.2: Modelo de color RGB

FUENTE: Página Web <https://www.dfly.com/esp/hslrgb.html>. Consultada en junio 2016

TABLA 2.1: Ocho colores y sus valores HSL y RGB asociados

Color	Matiz	Sat	Lum	Rojo	Verde	Azul
Rojo	0	240	120	255	0	0
Amarillo	40	240	120	255	255	0
Verde	80	240	120	0	255	0
Cian	120	240	120	0	255	255
Azul	160	240	120	0	0	255
Fucsia	200	240	120	255	0	255
Blanco	0	0	240	255	255	255
Negro	0	0	0	0	0	0

FUENTE: Página Web <https://www.dfly.com/esp/hslrgb.html>. Consultada en junio 2016

2.2.2.1 Percepción y sensación de color

Los ojos humanos tienen dos tipos de células sensibles a la luz o fotorreceptores: los bastones y los conos. Estos últimos son los encargados de aportar la información de color.

Para saber cómo es percibido un color, hay que tener en cuenta que existen tres tipos de conos con respuestas frecuenciales diferentes, y que tienen máxima sensibilidad a los colores que forman la terna RGB. Aunque los conos, que reciben información del verde y el rojo, tienen una curva de sensibilidad similar, la respuesta al color azul es una veintava (1/20) parte de la respuesta a los otros dos colores. Este hecho lo aprovechan algunos sistemas de codificación de imagen y vídeo, como el JPEG o el MPEG, "perdiendo" de manera consciente más información de la componente azul, ya que el ser humano no percibe esta pérdida.

La sensación de color se puede definir como la respuesta de cada una de las curvas de sensibilidad al espectro radiado por el objeto observado.

De esta manera, obtenemos tres respuestas diferentes, una por cada color (Mandado, 2007).

El hecho de que la sensación de color se obtenga de este modo, hace que dos objetos observados, radiando un espectro diferente, puedan producir la misma sensación. Y en esta limitación de la visión humana se basa el modelo de síntesis del color, mediante el cual podemos obtener a partir de estímulos visuales estudiados y con una mezcla de los tres colores primarios, el color de un objeto con un espectro determinado.

2.2.2.2 Tratamiento de la señal de video RGB

RGB es el tratamiento de la señal de vídeo que trata por separado las señales de los tres colores rojo, verde y azul. Al usarlo independientemente, proporciona mayor calidad y reproducción más fiel del color. El modelo de color llamado RGB es el que se utiliza en todos los sistemas que forman imágenes a través de rayos luminosos, ya sea emitiéndolos o recibiendo los.

El modelo RGB está formado por los tres componentes de colores primarios aditivos y como mínimo un componente de sincronismo. Los componentes de color son las señales rojo, verde y azul; siendo transmitidos cada uno independiente y aislado del resto.

De esta forma no hay pérdidas en el tratamiento de la imagen puesto que los colores primarios siguen existiendo como tal en su transmisión. Por el contrario, mediante este sistema hay mucha información redundante, con el consiguiente aumento del ancho de banda necesario respecto a otros métodos de transmisión. Por ejemplo, cada color lleva el

valor de brillo de toda la imagen, de forma que esta información está por triplicado (Mandado, 2007).

2.2.3 PROFUNDIDAD DE COLOR

La profundidad de color o bits por pixel (bpp) es un concepto de la computación gráfica que se refiere a la cantidad de bits de información necesarios para representar el color de un píxel en una imagen digital o en un framebuffer. Debido a la naturaleza del sistema binario de numeración, una profundidad de bits de n implica que cada píxel de la imagen puede tener 2^n posibles valores y, por lo tanto, representar 2^n colores distintos.

Debido a la aceptación prácticamente universal de los octetos de 8 bits como unidades básicas de información en los dispositivos de almacenamiento, los valores de profundidad de color suelen ser divisores o múltiplos de 8, a saber 1, 2, 4, 8, 16, 24 y 32, con la excepción de la profundidad de color de 10 o 15, usada por ciertos dispositivos gráficos (Angulo, 1996).

2.2.3.1 Color Real o True color

Para la profundidad de color de 24 bits por pixel, se habla de color verdadero debido a que la policromía se acerca a lo que el ojo humano puede encontrar en el mundo real, y a que dicho ojo humano se torna incapaz de diferenciar entre un tono y otro, si la diferencia se mantiene en un cierto rango mínimo. En la profundidad de color de 24 bits por pixel, figura 2.3, se dedica un octeto entero a representar la intensidad luminosa de cada uno de los tres tonos primarios de rojo, verde y azul, lo cual

permite que cada pixel pueda tomar $2^{24} = 256 \times 256 \times 256 = 16.777.216$ colores distintos.



FIGURA 2.3: Imagen en diferentes profundidades de color

FUENTE: Página Web <http://pixelnauta.com.ar/wp/?p=691>. Consultada en junio 2016

Cuando se utilizan 32 bits para representar un color se agrega al esquema de 256 valores para cada tono primario un cuarto canal denominado alfa que representa la transparencia. Este valor se utiliza cuando se deben superponer dos imágenes (Angulo, 1996).

2.2.4 MATRIZ LED RGB 8X8 ÁNODO COMÚN (D23088A/B)

Para este proyecto se está utilizando el módulo de matriz LED RGB 8X8 ánodo común con código D23088A/B, este módulo consta de 64 LED's RGB, 192 LED's en total, el módulo se muestra en la figura 2.4 y el patillaje se muestra en la figura 2.5 (Betlux, 2016).



FIGURA 2.4: Matriz LED RGB 8X8 ánodo común

FUENTE: Página Web www.amazon.com. Consultada en junio 2016

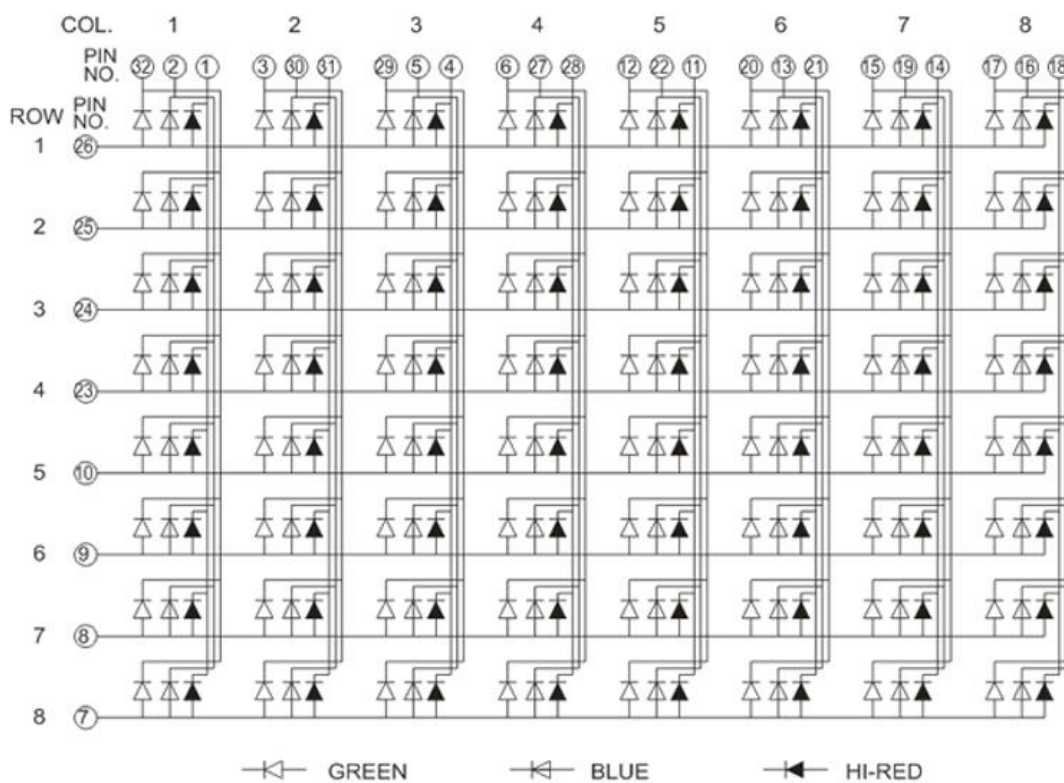


FIGURA 2.5: Patillaje del módulo de matriz LED RGB 8X8

FUENTE: Datasheet en página Web <http://www.betlux.com>. Consultada en junio 2016

2.2.5 RASPBERRY PI 3 MODELO B

Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

Para este proyecto se está utilizando la microcomputadora RASPBERRY PI 3 modelo B, como se muestra en la figura 2.9, que es su versión más reciente.

En su última versión la placa Raspberry Pi ha aumentado entre un 50 y un 60 por ciento el rendimiento respecto a su antecesor.



FIGURA 2.6: Microcomputadora RASPBERRY PI 3 modelo B

FUENTE: Página Web www.raspberrypi.org. Consultada en junio 2016

En este nuevo modelo, como se observa en la figura 2.10, se ha substituido el chip BCM 2836 ARMv7 Cortex a 900MHz por el BCM 2837, un

procesador de cuatro núcleos ARMv8 Cortex-A53 de 64 bit a 1,2GHz; pero cabe destacar que mantiene el bus compartido entre los puertos USB y su controladora ethernet 10/100.

Incorpora conectividad inalámbrica Wifi 802.11n y Bluetooth 4.1, manteniendo la compatibilidad del software y la distribución de los pines GPIO para con las anteriores versiones.

Este nuevo modelo, como se muestra en la tabla 2.2, tiene un consumo energético máximo de 2,5 Amperios por lo que cada vez se hace más presente la necesidad de utilizar fuentes específicas, siendo ya insuficientes los 500mA que ofrecen la mayoría de los puertos USB convencionales para tareas de alto rendimiento (Schmidt Y Carter, 2012).

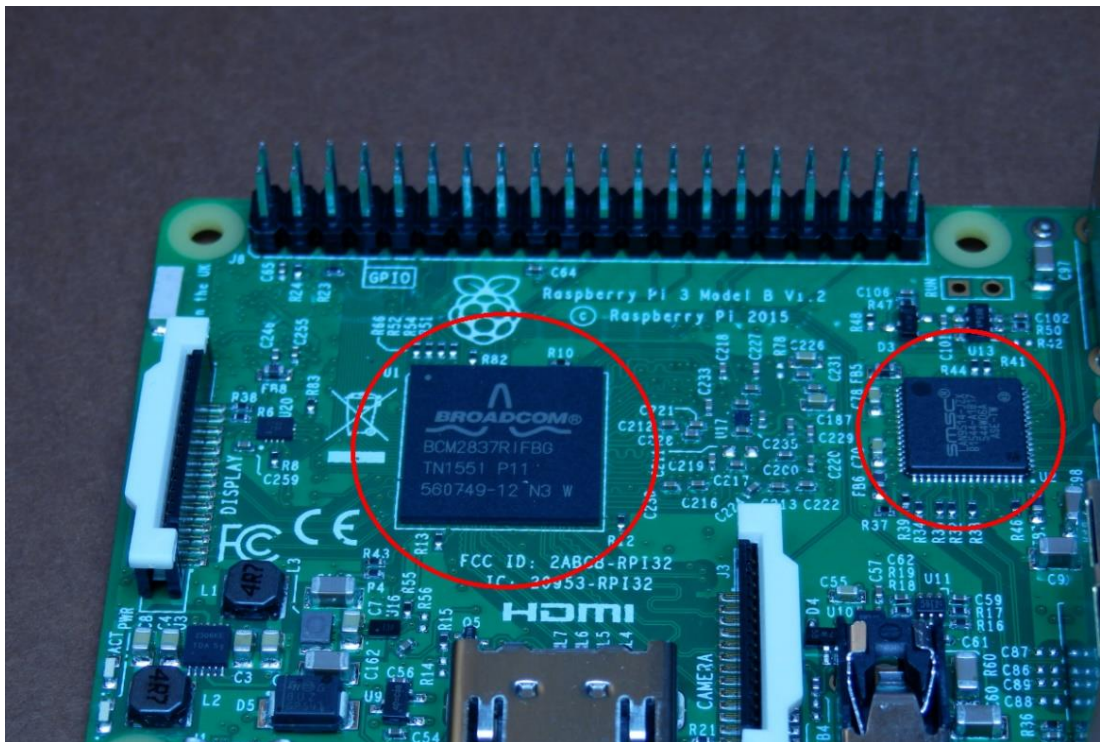


FIGURA 2.7: Chip BCM2837 ARMv8 Cortex-A53 de 64 bit, y memoria RAM de 1GB en la placa Raspberry Pi 3.

FUENTE: Página Web <http://i2.wp.com>. Consultada en junio 2016

TABLA 2.2: Especificaciones técnicas y comparación entre las diferentes versiones de Raspberry Pi

VERSIONES	Raspberry PI B	Raspberry PI B+	Raspberry PI 2	Raspberry PI 3
CPU	ARMv6 700Mhz	ARMv6 700Mhz	ARMv7 Cortex Quadcore 900Mhz	ARMv8 Cortex Quadcore 1,2GHz
GPU	VideoCore IV Dual Core 250MHz	VideoCore IV Dual Core 250MHz	VideoCore IV Dual Core 250MHz	VideoCore IV Dual Core 400MHz
RAM	512MB	512MB	1024MB (1GB)	1024MB (1GB)
Ethernet/WiFi	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100 – WiFi 802.11 b/g/n
Alimentación	5V 1A	5V 1A	5V 2A	5.1V 2.5A
Temperatura mínima	33.6° C	31.5° C	34.2° C	33.1° C
Temperatura máxima	50.8° C	50.8° C	50.8° C	48.7° C
Tasa de lectura	10.46 MB/s	10.03 MB/s	10.46 MB/s	10.46 MB/s
Tasa de escritura	3.34 MB/s	3.34 MB/s	3.34 MB/s	3.34 MB/s
Velocidad de subida	2.21 MB/s	2.21 MB/s	2.21 MB/s	2.21 MB/s
Velocidad de bajada	2.84 MB/s	2.84 MB/s	2.84 MB/s	2.84 MB/s

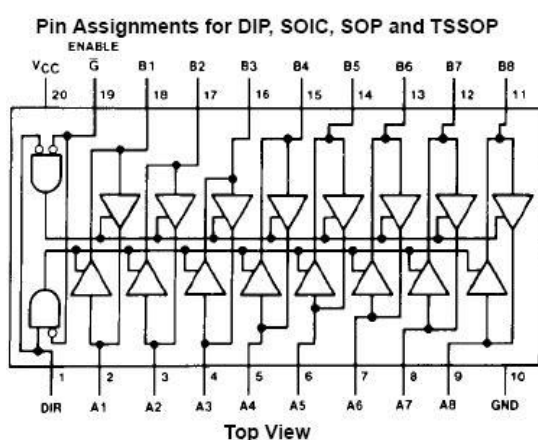
Mflops	45.43	45.19	119.79	180.00
VAX Mips	855.06	851.07	1536.95	2469.38
Test de estabilidad	Superado	Superado	Superado	Superado

FUENTE: Página Web <http://i2.wp.com>. Consultada en junio 2016

2.2.6 IC 74HC245

El circuito integrado 74HC245 es un transceptor octal que posee salidas 3 estados (tri-state) no inversoras, en ambas direcciones, tanto para enviar como para recibir. El circuito integrado 74HC245 cuenta con una entrada de habilitación de salidas (OE) para una fácil conexión en cascada y una entrada de envío / recepción (DIR) para el control de la dirección, como se observa en la figura 2.11. La entrada OE controla la habilitación de las salidas de manera que se aíslan con eficacia de los buses (NXP Semiconductors, 2015).

Connection Diagram



Truth Table

Control Inputs		Operation
G	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = HIGH Level
L = LOW Level
X = Irrelevant

FIGURA 2.8: Pinout y diagrama de conexión del 74HC245

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

2.2.7 IC 74HC595

El circuito integrado 74HC595 es un registro de desplazamiento de 8 bits con entrada serie / salida serie o paralelo con latch ("registro de almacenamiento") y salidas 3 estados (tri-state).

Tanto el registro de desplazamiento como el registro de almacenamiento tiene relojes separados, como se observan en las figuras 2.12 y 2.13. El dispositivo tiene una entrada serial (DS) y una salida serial (Q7S) para permitir la conexión en cascada y una entrada de reset asíncrono (MR). Un nivel bajo en la entrada MR resetea el registro de desplazamiento.

Los datos se desplazan en las transiciones de bajo a alto de la entrada de reloj SHCP. Los datos en el registro de desplazamiento se transfieren al registro de almacenamiento en una transición de bajo a alto de la entrada STCP. Si ambos relojes están conectados entre sí, el registro de desplazamiento siempre estará un pulso de reloj delante del registro de almacenamiento. Los datos en el registro de almacenamiento aparecen en la salida siempre que la entrada de habilitación de salidas (OE) es bajo, como se observa en la figura 2.14. Un nivel alto en OE hace que las salidas asuman un estado de alta impedancia. El funcionamiento de la entrada OE no afecta el estado de los registros. Las entradas incluyen diodos de fijación. Esto permite el uso de resistencias limitadoras de corriente para interconectar entradas a tensiones superiores a VCC (NXP Semiconductors, 2015).

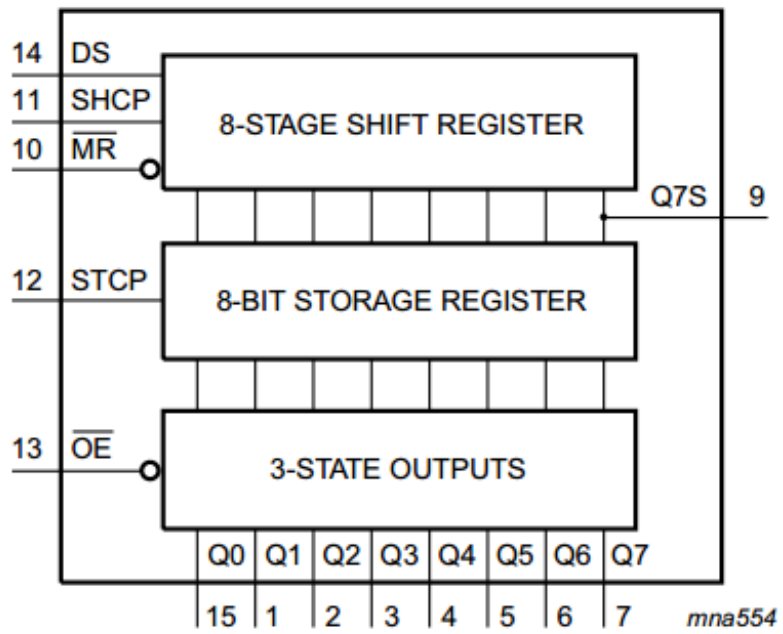


FIGURA 2.9: Diagrama funcional del 74HC595

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

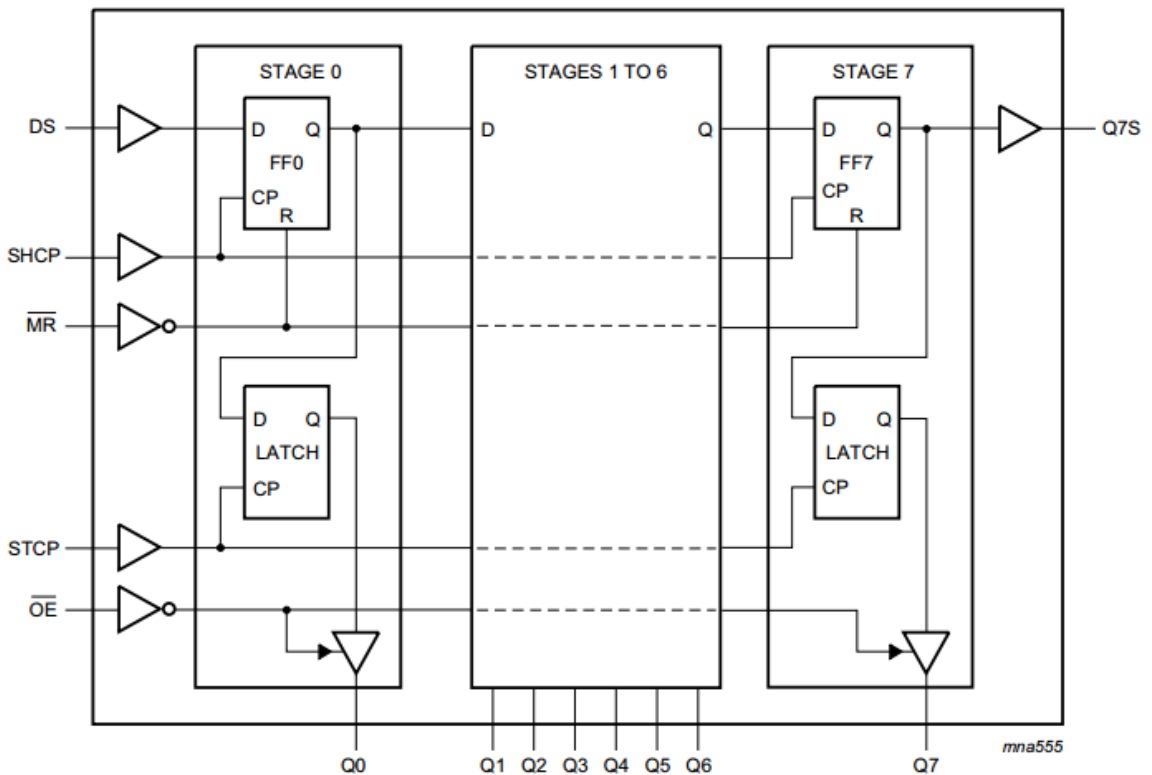


FIGURA 2.10: Diagrama lógico del 74HC595

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

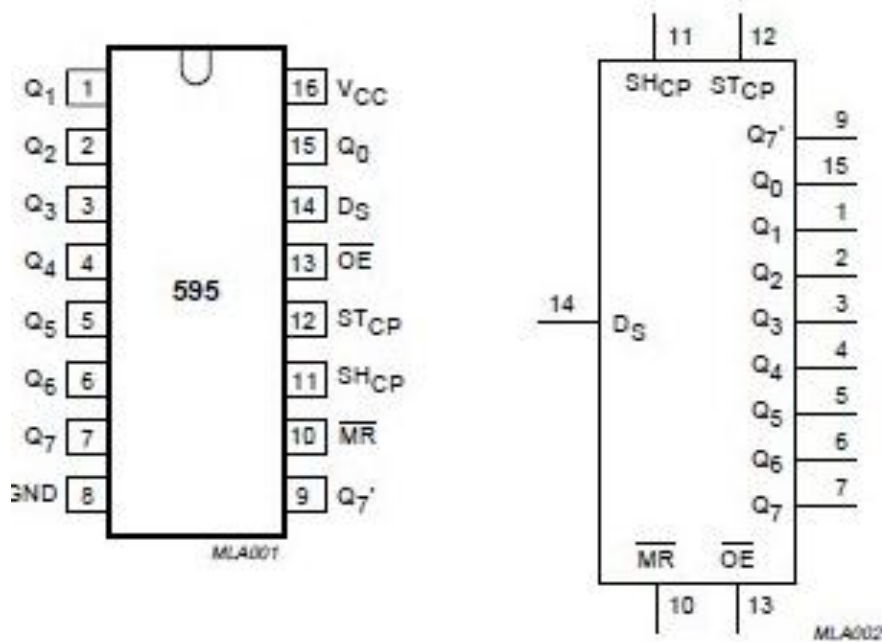


FIGURA 2.11: Pinout del 74HC595

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

2.2.8 IC 74HC138

El circuito integrado 74HC138, como se muestra en la figura 2.15, es un circuito integrado que tiene la función de decodificador / demultiplexor binario de 3 bits (1:8).

Con las tres entradas que posee el circuito podemos realizar 8 combinaciones diferentes, de 000 a 111 que nos activaran una de las salidas Y_n . Este circuito integrado se utiliza mucho para seleccionar memorias y periféricos en el espacio de memoria de los sistemas con microprocesadores.

La habilitación del 74HC138, como se observa en la tabla 2.3, se activa sólo cuando se cumple la siguiente ecuación de las patillas de entrada: $E = G1 * G2A * G2B$.

Utilizando la formula anterior podemos hacer decodificaciones de más salidas, activando o desactivando la habilitación se pueden conectar en

cascada más circuitos para realizar decodificaciones mayores. Aunque si queremos un decodificador que tenga una entrada más y el doble de salidas ya tenemos el circuito 74HC154. El tiempo de retardo o propagación del 74HC138 es de unos 12nS y las salidas son del tipo Totem pole (NXP Semiconductors, 2015).

TABLA 2.3: Tabla de verdad del 74HC138

Control			Input			Output							
E1	E2	E3	A2	A1	A0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X											
X	X	L											
L	L	H	L	L	L	H	H	H	H	H	H	H	L
			L	L	H	H	H	H	H	H	H	L	H
			L	H	L	H	H	H	H	H	L	H	H
			L	H	H	H	H	H	H	L	H	H	H
			H	L	L	H	H	H	L	H	H	H	H
			H	L	H	H	H	L	H	H	H	H	H
			H	H	L	H	L	H	H	H	H	H	H
			H	H	H	L	H	H	H	H	H	H	H

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

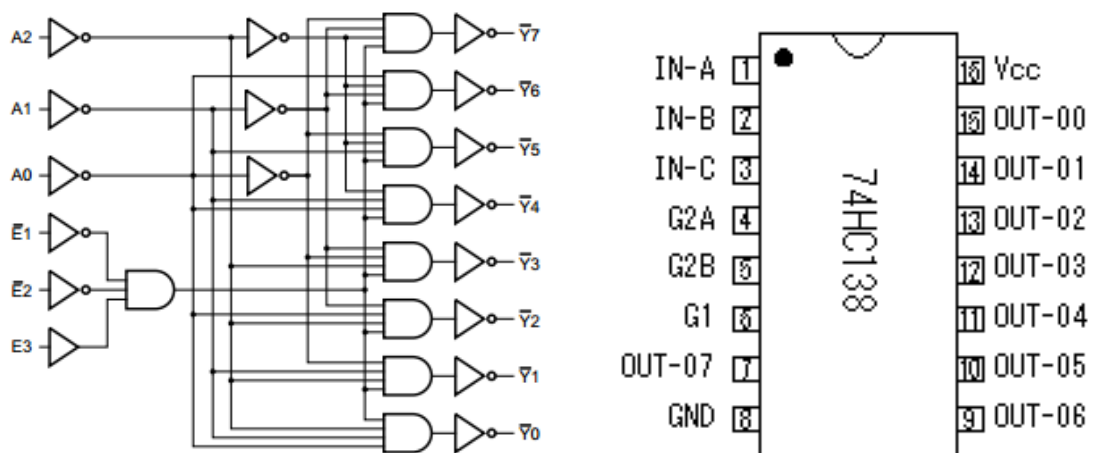


FIGURA 2.12: Diagrama lógico y pinout del 74HC138

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

2.2.9 IC 74HC04

El circuito integrado 74HC04 cuenta con 6 inversores independientes con tecnología HC, como se observa en la figura 2.16, que es la versión rápida del circuito integrado. HC del inglés “High commutation” alta conmutación. Se le utiliza en equipos digitales que trabajan a altas frecuencias. Cada inversor puede ser usado sin la necesidad de conectar los demás.

Su salida es el estado inverso a su entrada, la cual no debe ser superior al voltaje de alimentación del circuito integrado.

Se debe alimentar con una fuente de alimentación de 5v de corriente continua, para mantener los estados lógicos digitales en sus valores correctos. El 7404 puede funcionar en varios tipos de conexiones, siendo las más comunes: inversor y oscilador (NXP Semiconductors, 2015).

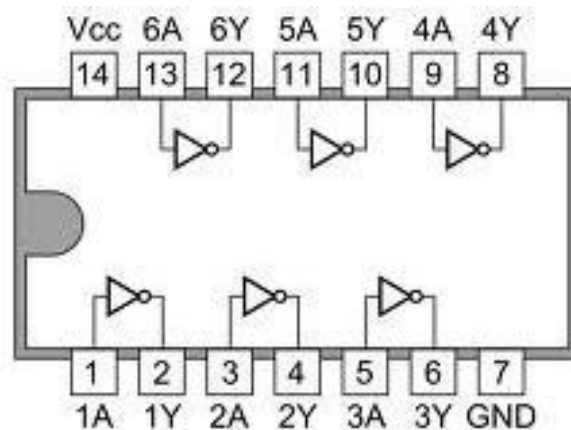


FIGURA 2.13: Pinout del 74HC04

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

2.2.10 Transistor BD135

Este componente es un transistor BJT NPN, figura 2.17, de IC (nominal) = 2A, tiene una ganancia típica $h_{fe} = 100$ y una disipación de potencia de 25w y frecuencia de operación típica de 8 Mhz, dentro de sus funciones es

utilizado como amplificador y en aplicaciones de conmutación, por su amplio uso se pueden encontrar en una variada gama de equipos electrónicos, fundamentalmente en aquellos que se utilizan como amplificadores de señal de audio (Fairchild Semiconductor, 2013).



FIGURA 2.14: Transistor BD135

FUENTE: Datasheet en página Web <http://projectshopbd.com>. Consultada en junio 2016

2.2.11 IC ULN2803

El circuito integrado ULN2803 es un arreglo de transistores Darlington que ofrecen 8 salidas de alta corriente, para impulsar dispositivos como: relays, motores, solenoides, bombillas de filamento, series de leds, displays, etc.

El ULN2803 lo podemos encontrar en encapsulados PDIP y SOIC de 18 pines, tal como se muestra en la figura 2.19. Otras de sus características principales son:

-Voltaje de alimentación: Típicamente el ULN2803 se puede utilizar con voltajes de 5, 9 y 12 voltios, pero también puede soportar voltajes de hasta 50v.

El “voltaje de alimentación” se hace por el pin 10 (COM), que más bien se trata de un punto común para todos los cátodos de los diodos clamp, como

se observa en la figura 2.18. La conexión a tierra (GND) la encontramos en el pin 9.

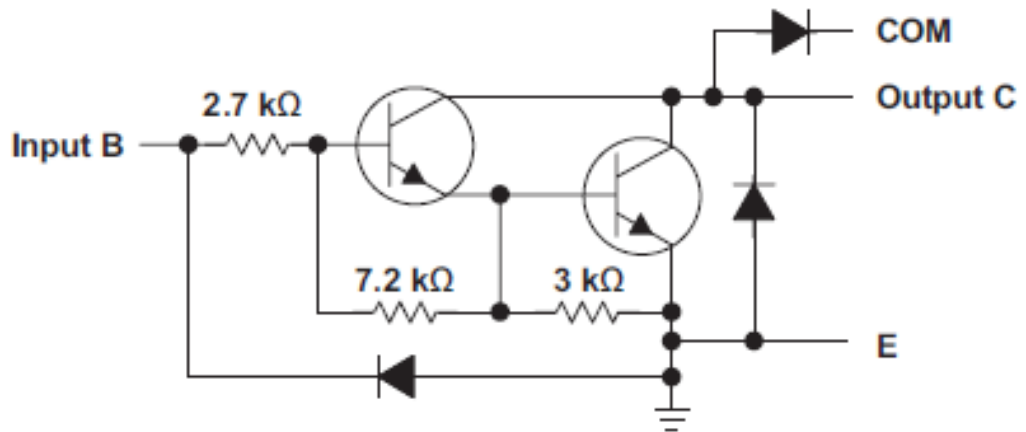


FIGURA 2.15: Diagrama funcional del ULN2803

FUENTE: Datasheet en página Web <http://www.ti.com>. Consultada en junio 2016

-Corriente: La corriente máxima que podemos obtener por salida del ULN2803 es de 500ma, pero la misma se puede incrementar si utilizamos las salidas en paralelo.

- Entradas: Responden a niveles lógicos C-MOS y TTL; soportan un voltaje máximo de 30v. Las entradas se encuentran desde el pin 1 al 8.

-Salidas: Se encuentran desde el pin 11 al 18. Se pueden conectar en paralelo para aumentar la corriente máxima de 500ma.

-Diodos clamp: El ULN2803 posee un diodo polarizado inversamente entre cada salida y el "voltaje de alimentación" (pin 10 (COM)) con el objetivo de eliminar las corrientes inversas, que se producen cuando se conectan cargas inductivas como: relays, solenoides, motores y transformadores. Esto evita que los transistores se dañen (Texas Instruments, 2015).

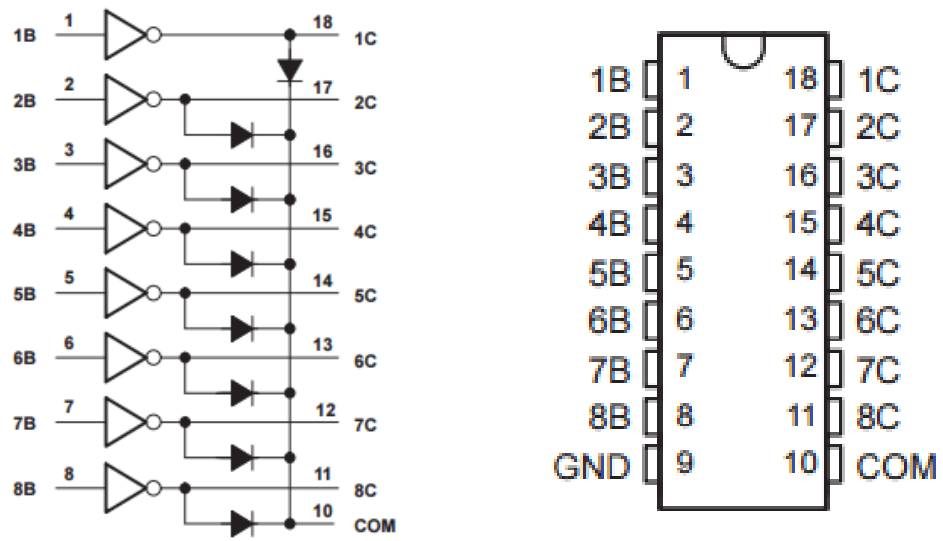


FIGURA 2.16: Diagrama lógico y pinout del ULN2803

FUENTE: Datasheet en página Web <http://www.ti.com>. Consultada en junio 2016

2.3 MARCO CONCEPTUAL

2.3.1 FRECUENCIA DE REFRESCO

La frecuencia de refresco de una pantalla de LED es el número de veces en un segundo que el hardware de la pantalla LED dibuja los datos. Si la pantalla muestra sólo una trama, esa trama se vuelve a dibujar en la pantalla a la frecuencia de refresco. La frecuencia de refresco de las pantallas LED es de unos 400 Hz como mínimo. Pero alrededor de este valor, las cámaras de vídeo pueden capturar la imagen parpadeante en la pantalla LED. Para imágenes sin parpadeo, la frecuencia de refresco debe ser superior a 1000 Hz. En las modernas pantallas de LED, dependiendo del tamaño de la pantalla y del tipo de escaneo, las tasas de refresco de hasta 9000 Hz son fácilmente alcanzables. Para altas tasas de refresco, el hardware de la pantalla LED debe ser capaz de manejar altas frecuencias. Si no, puede haber líneas o sombras indeseadas en la imagen (Hambley, 2001).

2.3.2 PIXEL PITCH

Pixel pitch es la distancia desde el centro de un grupo de LEDs (o píxel) al centro del siguiente grupo / píxel de LEDs medido en milímetros. El pixel pitch varía típicamente de 4 mm a 20 mm para pantallas LED interiores; Para pantallas LED exteriores, el pixel pitch puede oscilar entre 10mm a 34mm o más alto (Tocci, 2003).

2.3.3 PIXEL

Un pixel es un punto de luz y el elemento más pequeño de una pantalla LED. Una pantalla LED es en realidad una matriz de píxeles. Los píxeles son capaces de representar letras, gráficos, animaciones y videos.

En las pantallas LED exteriores, éstas tienden a estar compuestas por 2 LEDs rojos, 1 verde y 1 azul (o a veces 2R, 1G, 1B). Los LEDs de cada píxel se iluminan hasta diferentes intensidades para crear un color específico, actualizándose muchas veces por segundo, y cada pantalla está compuesta por cientos de miles de píxeles, creando una imagen perfecta.

Es el mismo principio de cómo funciona un televisor o pantalla de computadora; Sin embargo, debido a que el visor está mucho más lejos, los píxeles no necesitan ser tan pequeños (Tocci, 2003).

2.3.4 SCANNING MODE

En el interior de una pantalla LED, la relación de las líneas que se encienden al mismo tiempo y las líneas de toda la señalización digital, se llama scanning mode y esta técnica se puede dividir en 1/2 scan, 1/4 scan, 1/8 scan y 1/16 scan. Esto significa que el método de envío de los datos es diferente, lo que puede influir en el ajuste de las tarjetas de recepción. Al emplear el scanning mode se reducirá el brillo de la pantalla LED y en la mayoría de los casos se utilizará una resolución más alta para la compensación de brillo (Tokheim, 2016).

2.3.5 DECODIFICADOR / DEMULTIPLEXOR

Un decodificador o descodificador es un circuito combinacional, cuya función es inversa a la del codificador, esto es, convierte un código binario de entrada (natural, BCD, etc.) de N bits de entrada y M líneas de salida (N puede ser cualquier entero y M es un entero menor o igual a 2^N), tales que cada línea de salida será activada para una sola de las combinaciones posibles de entrada. Normalmente, estos circuitos suelen encontrarse como

decodificador / demultiplexor. Esto es debido a que un demultiplexor puede comportarse como un decodificador.

En electrónica digital, un demultiplexor es un circuito combinacional que tiene una entrada de información de datos d y n entradas de control que sirven para seleccionar una de las 2^n salidas, por la que ha de salir el dato que presente en la entrada. Esto se consigue aplicando a las entradas de control la combinación binaria correspondiente a la salida que se desea seleccionar (NXP Semiconductors, 2015).

Si por ejemplo se tiene un decodificador de 2 entradas con $2^2=4$ salidas, su funcionamiento sería el que se indica en la figura 2.17, donde se ha considerado que las salidas se activen con un "uno" lógico:

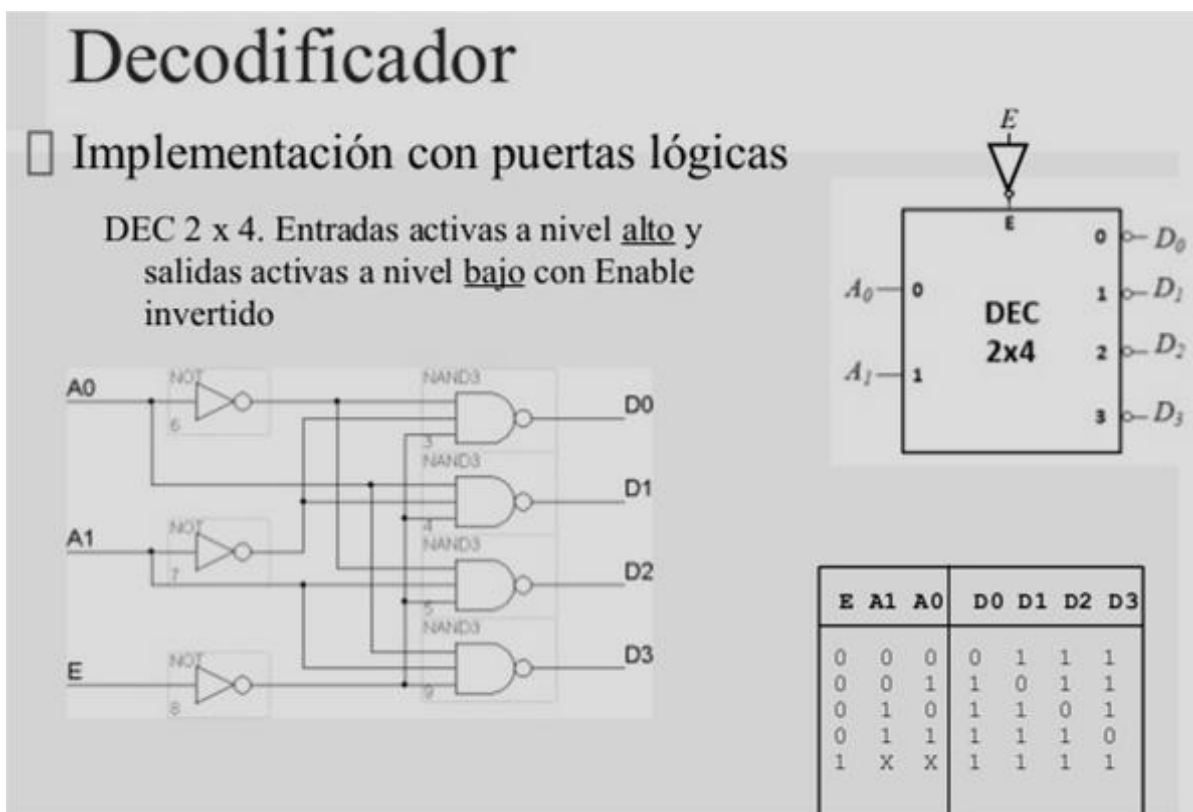


FIGURA 2.17: Decodificador 2 x 4

FUENTE: Página Web es.slideshare.net. Consultada en junio 2016

CAPÍTULO III

DISEÑO Y DESCRIPCIÓN DEL SISTEMA

En este tercer capítulo, se realizará un análisis del sistema, es decir se analizará el funcionamiento de las pantallas gigantes comerciales de los grandes fabricantes como referencia para luego plantear el diseño del sistema, tanto el primer como el segundo diseño de este proyecto serán explicados y detallados. Finalmente se realizará una revisión y consolidación final de los resultados obtenidos tras la implementación de este proyecto.

3.1 ANÁLISIS DEL SISTEMA

En la actualidad, todos los principales fabricantes de placas de video-pantalla LED tienen arquitecturas similares, pero patentadas usando módulos de paneles de vídeo LED RGB de diferentes tamaños (P5, P6, P7, P10, P20, P25, P32), colores (RGB, WRGB, etc.), tecnologías (virtual Pixel), etc. Generalmente se utilizan FPGA's junto con microprocesadores para poder manejar y distribuir y procesar toda la data y poder mostrarlo en cada uno de los LED's.

Se calcula que decenas de miles de paneles de gran tamaño de visualización de vídeo LED RGB han sido instalados en todo el mundo. La adopción de visualización de pantallas de video LED puede llegar a ser aún más extendida si el precio global del sistema se pudiera reducir de manera significativa y el procedimiento de funcionamiento de este tipo de tableros se pudiera simplificar. El presente proyecto de investigación representa un diseño de referencia para un prototipo de panel LED RGB de bajo costo, básico y modular.

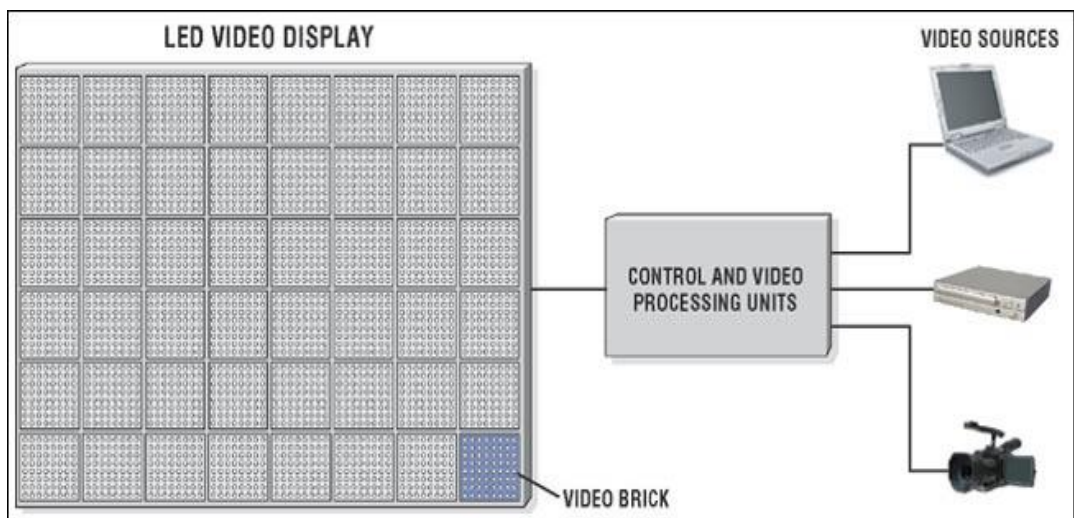


FIGURA 3.1: Arquitectura de un sistema de paneles LED RGB

FUENTE: Página Web maximintegrated.com. Consultada en junio 2016

Actualmente, Los más grandes fabricantes de paneles de visualización de vídeo LED RGB tienen arquitecturas similares, pero patentadas usando módulos de paneles de vídeo LED de diferentes tamaños, como se observa en la figura 3.1. La cantidad de píxeles de un módulo de panel LED RGB oscila desde 256 hasta 15.552, dependiendo del fabricante. Los paneles LED se pueden poner uno al lado del otro para montar una pantalla de vídeo de varios metros de tamaño de lado a lado, como se muestra en la figura 3.2.



FIGURA 3.2: Paneles LED RGB unidos formando una pantalla más grande

FUENTE: Página Web maximintegrated.com. Consultada en junio 2016

Los LED's y sus drivers están montados cerca unos de otros en diferentes PCB's dentro del módulo de panel LED. También se incluye en el PCB-driver de los LED, un FPGA y buffers de memoria de video. Para una pantalla gigante de vídeo LED, como se muestra en la figura 3.3, las interconexiones

entre los módulos de paneles LED se realizan normalmente utilizando cables coaxiales. La pantalla de vídeo LED se vincula al control y las unidades de procesamiento de vídeo por fibra óptica. La unidad de control se utiliza para configurar estos módulos de paneles de vídeo LED y para seleccionar la fuente de vídeo correcta. El procesador de vídeo dedicado recibe la señal de vídeo seleccionada, convierte su formato, y envía la información de datos correcta a la posición del pixel correspondiente. El almacenamiento temporal de datos y el escalamiento son tareas también realizadas por el procesador de vídeo dedicado. Las unidades de control y las unidades tratamiento y procesamiento de vídeo son equipos profesionales especializados que pueden llegar ser muy costosos.

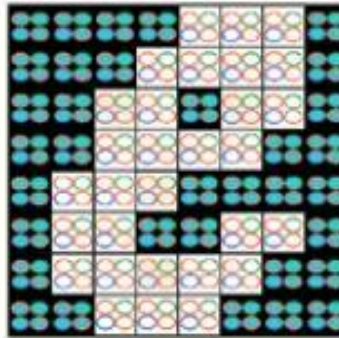


FIGURA 3.3: Display LED RGB formado por paneles LED RGB modulares

FUENTE: Página Web github.com. Consultada en junio 2016

Se debe tener presente que cada LED RGB consta de 3 LEDs individuales (Red, Green, Blue) y que cada LED RGB es un pixel. Sin embargo, existen fabricantes que utilizan en sus pantallas 4 LEDs para formar un pixel, 2 rojos, 1 verde, 1 azul, tal como se observa en la figura 3.4. Esta configuración se denomina R2G1B1 a diferencia de la tradicional que es la denominada R1G1B1, Esta configuración (R2G1B1) es la correcta para poder utilizar la tecnología llamada "Virtual Pixel" y claro el hardware también tiene que soportarlo.

1. The display effect without virtual technology



2. The display effect with virtual technology

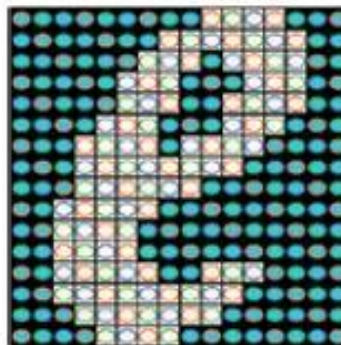


FIGURA 3.4: Arriba: Panel sin la tecnología Virtual Pixel; Abajo: Panel con la tecnología Virtual Pixel

FUENTE: Página Web maximintegrated.com. Consultada en junio 2016

Se ha realizado este prototipo de panel LED RGB de 16x32 (16 filas x 32 columnas), uniendo 8 módulos individuales más pequeños de matrices LED RGB de 8x8 de ánodo común como se aprecia en las figuras 3.5 y 3.6.

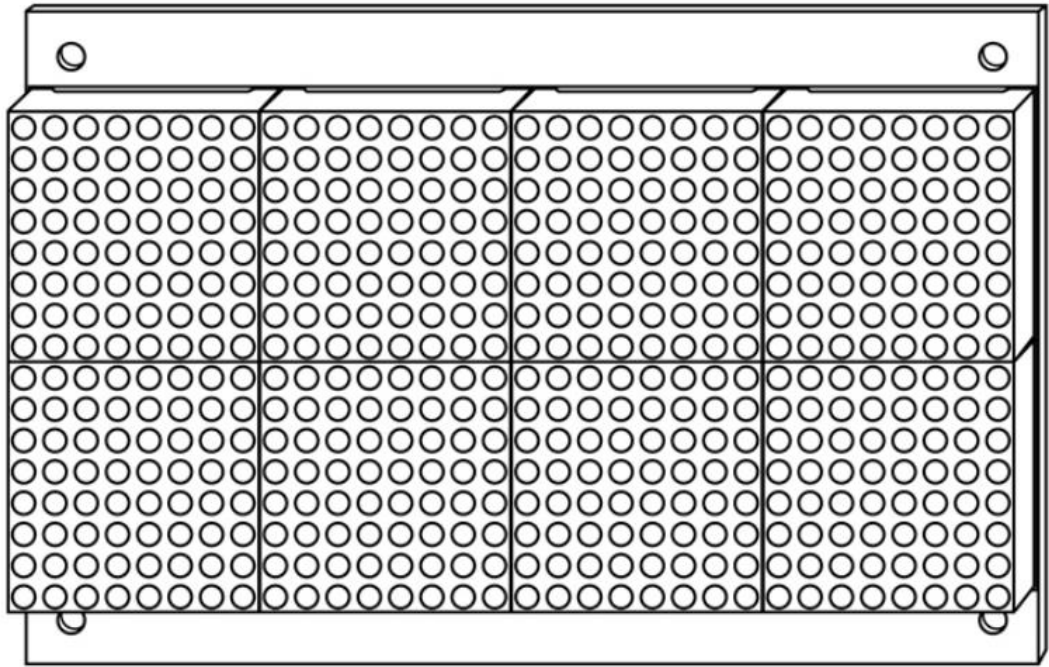


FIGURA 3.5: Prototipo propuesto de panel LED RGB 16x32 formado por módulos de matrices LED RGB 8X8 de ánodo común.

FUENTE: Página Web ackspace.nl. Consultada en junio 2016



FIGURA 3.6: Resultados finales obtenidos en paneles LED RGB

FUENTE: Página Web electronicproducts.com. Consultada en junio 2016

Este prototipo de panel RGB genera los diferentes colores gracias a la teoría del modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios, como se aprecia en la figura 3.7.

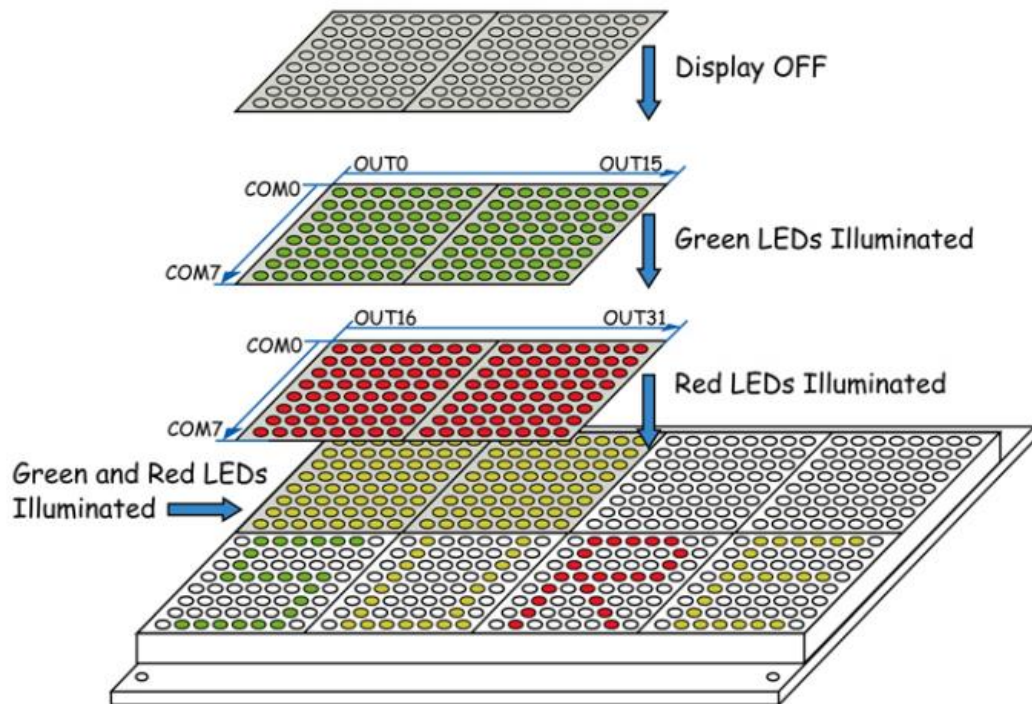


FIGURA 3.7: Generación de colores en el panel LED RGB 16x32

FUENTE: Página Web ackspace.nl. Consultada en junio 2016

El modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde o azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en distintos dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente.

3.2 DISEÑO DEL SISTEMA

3.2.1 DESCRIPCIÓN DEL DISEÑO

En este proyecto se ha planteado 2 diseños de prototipo de panel LED RGB de 16x32 (16 filas, 32 columnas), el primero es un diseño ideal con circuitos integrados específicos para la aplicación de matrices LED RGB, pero sin considerar la disponibilidad de dichos componentes (circuitos integrados, transistores, etc.) en Lima, así como también el costo elevado que tienen estos componentes de encontrarse en algún proveedor en Lima.

El segundo diseño es un diseño equivalente y alternativo con respecto al primero, y se obtienen los mismos resultados, con algunas ventajas (todos los componentes son fácilmente accesibles, menores costos, etc.) y ciertas desventajas (mayor cantidad de componentes utilizados, mayor cableado, circuito más complejo, etc.)



FIGURA 3.8: Ejemplo de panel LED RGB de mayores dimensiones

FUENTE: Página Web enviral-design.com. Consultada en junio 2016

Cabe resaltar que ambos diseños del prototipo de panel LED RGB de 16x32 (16 filas, 32 columnas) son totalmente funcionales y se recomienda

utilizar el primero si es que se cuenta con la disponibilidad de los componentes o si se implementa un panel RGB de mayores dimensiones, como se muestra en la figura 3.8, ya que esta optimizado para ello. Si no se cuenta con los componentes se puede optar por el segundo diseño que es totalmente funcional y también es escalable a mayores dimensiones, solo que la cantidad de componentes será mayor, así como el diseño del circuito.

Este prototipo de panel LED RGB utiliza lo que se conoce como PWM generado por software, es decir que todas las señales PWM que se ingresan a la matriz de LEDs son generadas por el controlador, en este caso la placa Raspberry Pi 3. Cabe destacar que esta es la técnica más difícil y compleja ya que el código fuente o la programación se hace muy extensa y por ende se satura al controlador por la cantidad de procesamiento que tiene que realizar llegando a utilizar toda la capacidad del microprocesador, así como casi toda la memoria RAM y FLASH del controlador, etc. Esto es así ya que los circuitos integrados (IC's drivers) que controlan a cada módulo de las matrices LED RGB son registros de desplazamiento sin ninguna clase de PWM incorporado, lo que hace muy barato el diseño del prototipo de panel LED RGB pero por contraparte deja toda la tarea de la generación de PWM al controlador, esta es la razón por la que se eligió la placa Raspberry Pi 3, figura 3.9, ya que esta microcomputadora, del tamaño de una tarjeta de crédito, cuenta con un microprocesador de 4 núcleos de 64 bits a 1.2 GHz, 1 Gb de RAM y una GPU a 400 MHz (Schmidt Y Carter, 2012), como se observa en la tabla 3.1, lo que hace que sea lo suficientemente potente para manejar varios paneles LED RGB de 32x32 en cascada, lográndose así un panel de dimensiones mayores.

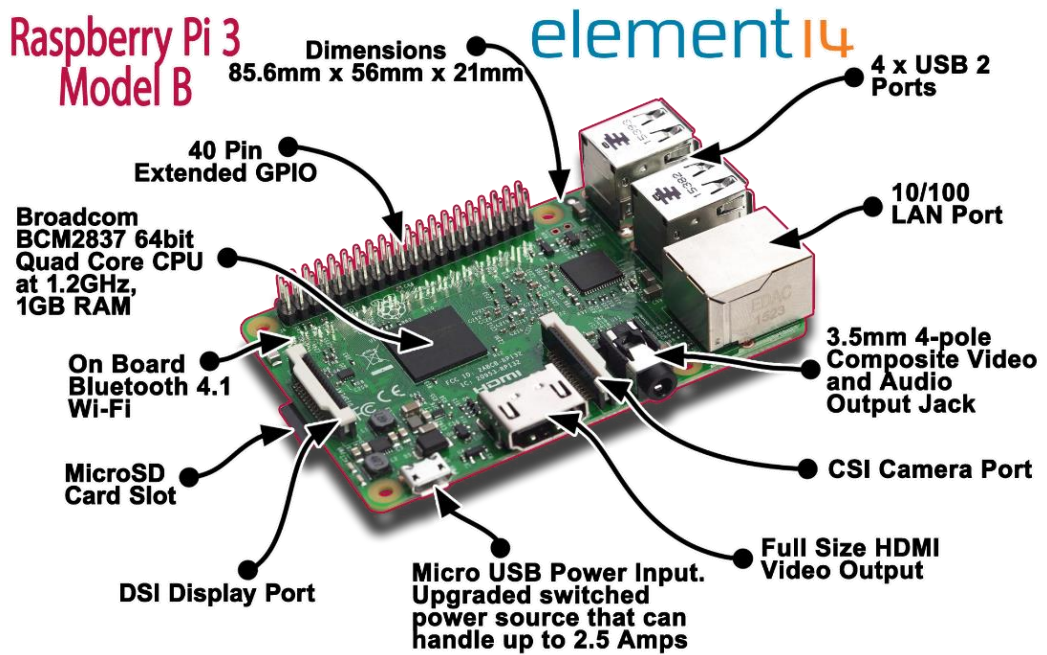
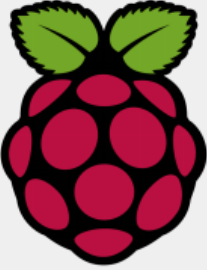


FIGURA 3.9: Microcomputadora Raspberry Pi 3

FUENTE: Página Web www.element14.com. Consultada en junio 2016

TABLA 3.1: Especificaciones Técnicas de la placa Raspberry Pi 3

	
Raspberry Pi 3 Model B	
Introduction Date	2/29/2016
SoC	BCM2837
CPU	Quad Cortex A53 @ 1.2GHz
Instruction set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40

FUENTE: Página Web www.hackaday.com. Consultada en junio 2016

Existen circuitos integrados con PWM incorporado tal como el IC TLC5947, de Texas Instruments, pero se necesitan muchos de estos IC's y su muy alto precio en comparación con los 74HC595 que se utilizaran en este proyecto, hace que sea una opción demasiado costosa.

Este prototipo de panel LED RGB necesita que se refresque la información a cada instante debido a su diseño. Es por ello que no se recomienda utilizar una placa controladora de prestaciones inferiores tal como Arduino ya que su capacidad de procesamiento va a ser insuficiente si se quiere desplegar muchos colores y especialmente si se desea agregar en cascada muchos de estos paneles, simplemente la placa Arduino no lo va a soportar, Cabe resaltar que la placa Raspberry Pi 3 trabajara bajo una distribución del Sistema Operativo Linux, específicamente el S.O. Raspbian, figura 3.10, para manejar el panel LED RGB.

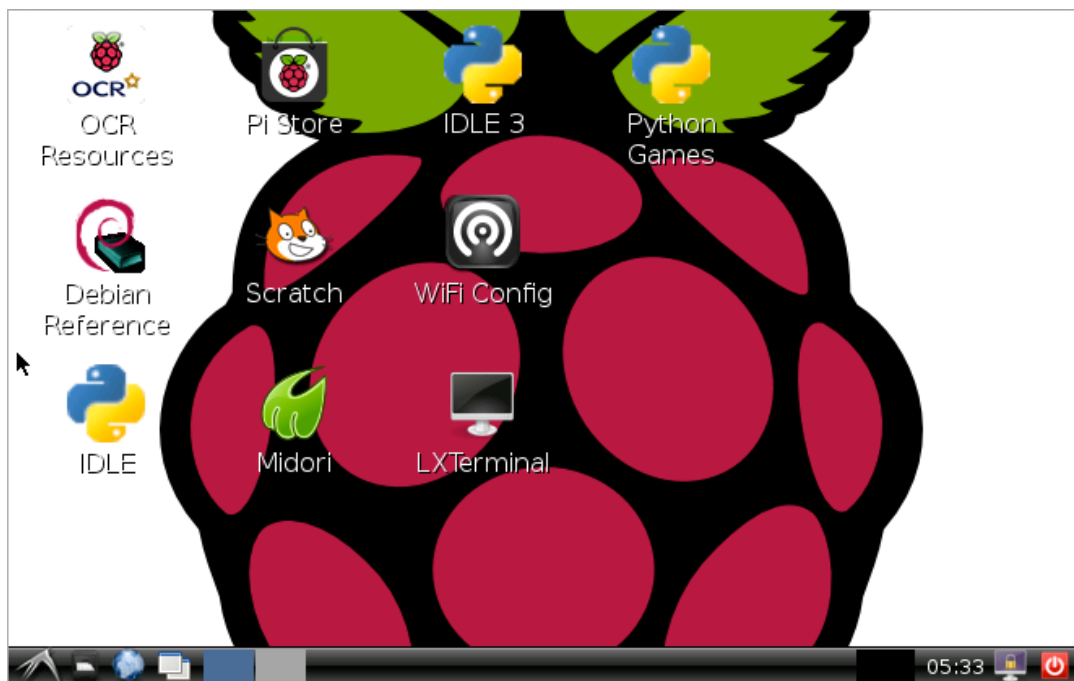


FIGURA 3.10: Escritorio del S.O. Raspbian instalado en la Raspberry Pi 3

FUENTE: Página Web dennistt.net. Consultada en junio 2016

En el numeral 3.2.2 Se hará una explicación detallada sobre el funcionamiento del prototipo de panel LED RGB de 16x32 (16 filas, 32 columnas), y el cual se puede extrapolar a un panel de 32x32 (32 filas, 32 columnas), que, básicamente, sólo añade otra línea de dirección. Básicamente, se puede controlar un panel tan grande como se desee, siempre y cuando se pueda enviar la data en forma paralela, y multiplexar a la vez las líneas de dirección lo suficientemente rápido, como se muestra en la figura 3.11.

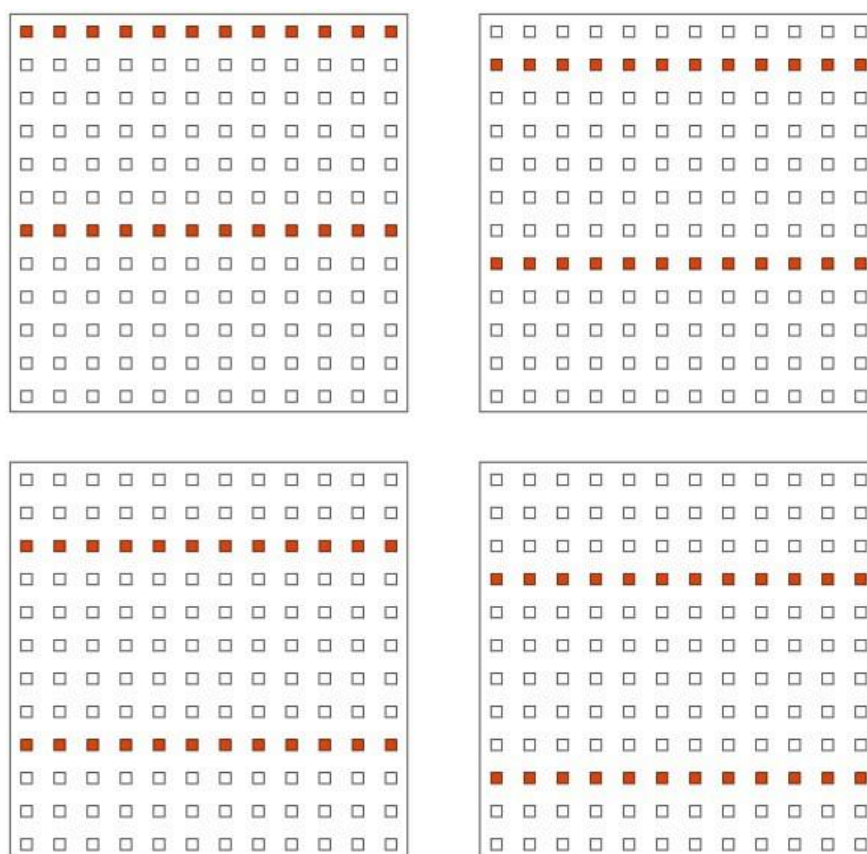


FIGURA 3.11: Multiplexación de las líneas de dirección (filas) con un scan rate de 1/6. Se puede ver que se multiplexan 2 filas a la vez.

FUENTE: Página Web www.lednews.org. Consultada en junio 2016

Esta es la razón por la cual las grandes compañías fabricantes de paneles LED RGB utilizan FPGA's + microprocesadores o controladores dedicados

para el control de sus módulos, estos son usados especialmente para recibir los datos de la PC y redistribuirlos. Las pantallas de fabricantes chinos suelen traer drivers sin generación de PWM incorporada, al igual que el diseño del presente prototipo de panel LED RGB. Hay un FPGA por cada cierto número de módulos de paneles LED RGB encargándose de llenar los drivers y recibir la trama de datos por par diferencial, como se muestra en la figura 3.12.

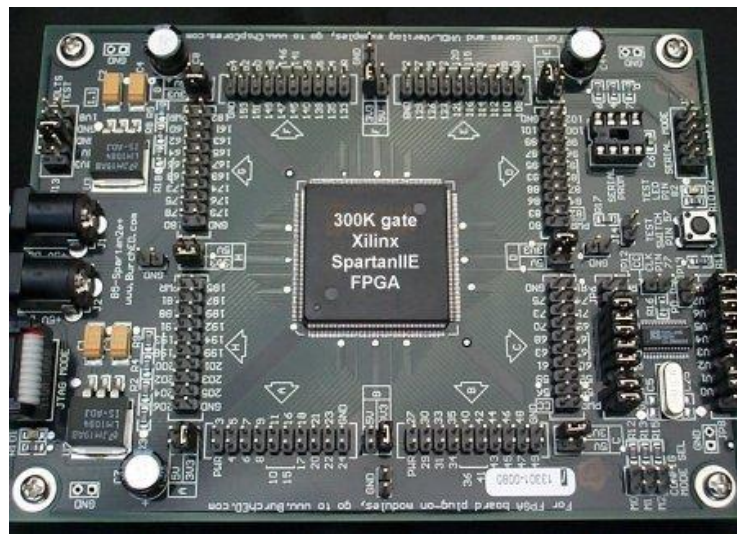


FIGURA 3.12: FPGA SpartanIIIE de Xilinx

FUENTE: Página Web members.optushome.com. Consultada en junio 2016

Los FPGA's son muy usados ya que se adaptan perfectamente con sincronización rápida y salidas de datos en paralelo. La desventaja es que son costosos y generalmente son usados varios FPGA's de alto rendimiento para manejar pantallas gigantes y adicionalmente se necesita de un controlador dedicado (microprocesador). En este proyecto se realiza el manejo de un prototipo de panel LED RGB (de diseño propio) para un sistema de información de la UNTELS, sin la necesidad de utilizar ningún FPGA, la placa Raspberry Pi 3 (que fue lanzada en febrero del 2016 y duplica la potencia de la anterior versión) se encarga de todo el procesamiento y

distribución de la data, y claro también se puede conectar varios de estos paneles en cascada gracias a la potencia de procesamiento de la placa Raspberry PI 3.

3.2.2 FUNCIONAMIENTO DEL PROTOTIPO DE PANEL RGB 16X32

Este panel muestra la información a través de la técnica de multiplexación, que básicamente significa actualizar la data de las filas y columnas del panel más rápido de lo que el ojo humano puede procesar. Si la actualización es lo suficientemente rápido, el ojo ve una imagen completa, sin parpadeo o flickering. En primer lugar, la fila a actualizar debe estar seleccionada. En este panel, esto se hace mediante el uso de un decodificador de direcciones de 3-a-8 (74HC138). Hay tres entradas de dirección al panel marcados como A, B y C. En base a la tabla 3.2, se puede ver que sólo una entrada está activa (nivel bajo) a la vez.

TABLA 3.2: Tabla de verdad del 74HC138

Input			Output							
A2	A1	A0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
X	X	X	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	L
L	L	H	H	H	H	H	H	H	L	H
L	H	L	H	H	H	H	H	L	H	H
L	H	H	H	H	H	H	L	H	H	H
H	L	L	H	H	H	L	H	H	H	H
H	L	H	H	H	L	H	H	H	H	H
H	H	L	H	L	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H

FUENTE: Datasheet en página Web <http://www.nxp.com>. Consultada en junio 2016

Aquí surge la primera interrogante ¿cómo los 8 bits del decodificador se pueden utilizar para seleccionar 16 filas totales? (panel LED RGB de 16x32). La técnica aquí es seleccionar filas en paralelo. Por ejemplo, en la figura 3.13, cuando se selecciona la fila 1 (fila 0), se selecciona también la 9ª fila (fila 8).

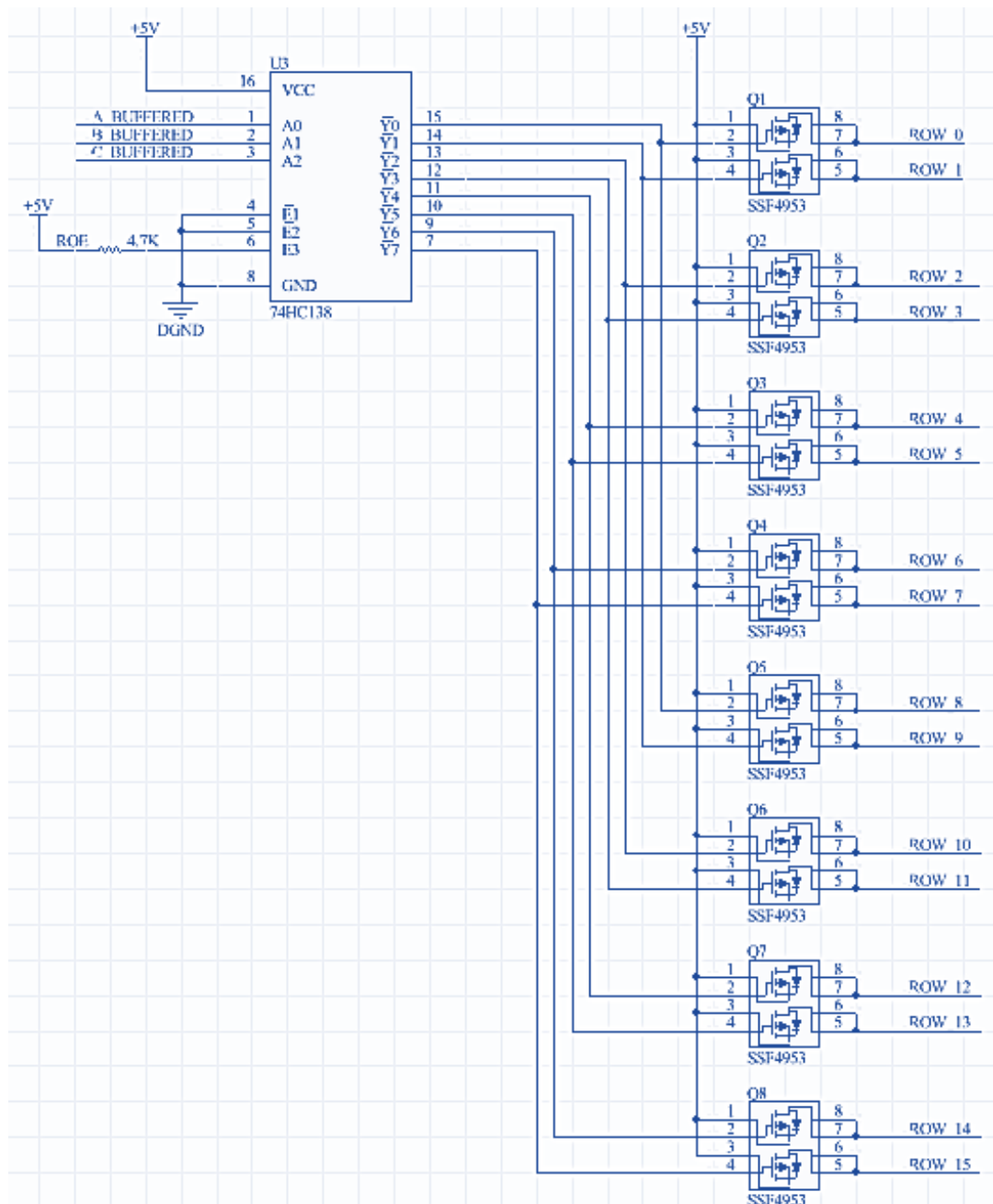


FIGURA 3.13: Esquema del 74HC138 con salidas hacia los MOSFET's

FUENTE: Página Web <http://i.stack.imgur.com/yKsPu.png>. Consultada en junio 2016

La ventaja del uso de esta técnica implica que solo hay que enviar los datos de todas las columnas para dos únicas filas a la vez en un mismo instante de tiempo. Sin embargo, las salidas del decodificador (74HC138) sólo puede manejar corrientes bajas y no pueden manejar una fila de LEDs directamente (NXP Semiconductors, 2015). Para remediar esto, 16 transistores MOSFET de canal P se utilizan como interruptores, que nos proporcionan la alta corriente que se necesita para alimentar una fila de LEDs. En el esquema anterior se puede ver que las salidas del decodificador se comparten entre cada 8 salidas (0 y 8, 1 y 9, repitiendo ese patrón hasta la fila 7 y 15).

Una vez seleccionadas las filas a enviar la data por medio del 74HC138, se debe almacenar los datos de las columnas. Los datos de las columnas se almacenan en un registro de desplazamiento de 16 bits de entrada serie / salida paralela. Puesto que el panel es de 32 píxeles de ancho, dos registros de desplazamiento deben ser encadenados juntos (conexión cascada). Este registro de desplazamiento (MBI5026), del fabricante MACROBLOCK, es un registro de desplazamiento que está diseñado especialmente para trabajar con LED's e implementa un sistema de corriente constante que garantiza el brillo de los LED's de manera permanente y uniforme (Macroblock Inc, 2004). Cabe resaltar que existen IC's equivalentes a este integrado tales como el TB62726 de TOSHIBA, el TLC5926 de Texas Instruments, el MAX6971 de MAXIM (Todos estos IC' son caros) y otros más genéricos de fabricantes chinos como el SM16162D, CYT62726B, STP16C596, los cuales son algo más baratos, pero mucho más difíciles de conseguir en Lima.

Hay 12 de estos registros de desplazamiento en total en el panel. cada 3 registros se utilizan para guardar la información de los datos de los colores rojo, verde y azul de 1 de 4 cuadrantes del panel, como se muestra en la figura 3.14. (3 colores * 4 cuadrantes = 12 IC's).

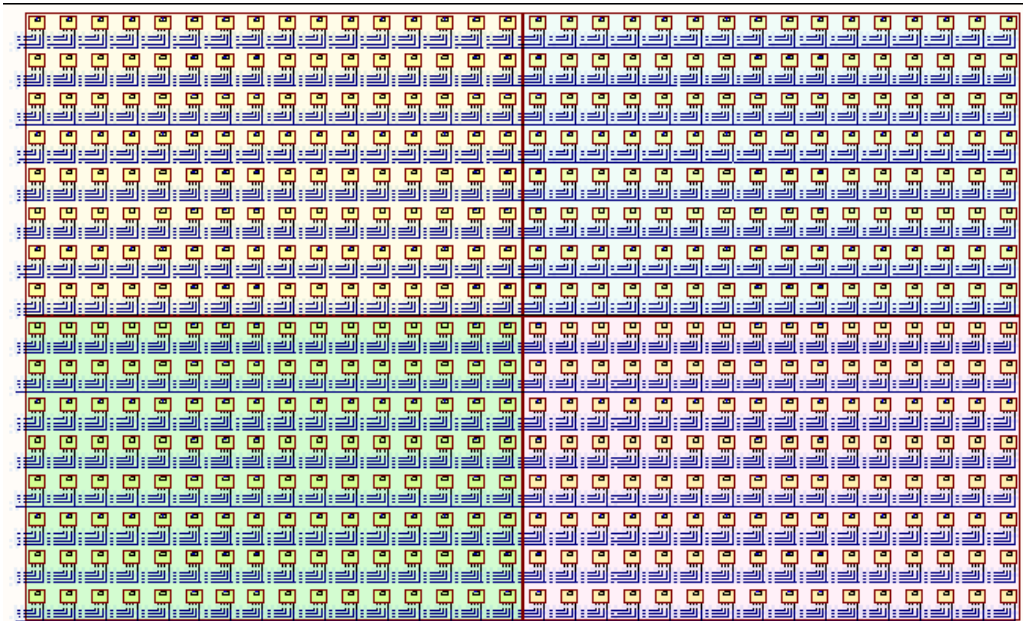


FIGURA 3.14: Panel LED RGB 16x32, mostrando los 4 cuadrantes

FUENTE: Página Web <http://i.stack.imgur.com/kbgVY.png>. Consultada en junio 2016

Aquí surge la segunda interrogante ¿De dónde vienen los cuadrantes? Como se mencionó anteriormente, se utilizan las 8 salidas del decodificador para seleccionar 16 filas seleccionando 2 filas en paralelo a la vez. Esto significa que debemos tener data única para cada fila de cada par de filas seleccionadas. Cuando la fila 0 y fila 8 son seleccionadas, hay que proporcionar a cada fila con datos exclusivos. Esto nos obliga a utilizar dos registros de desplazamiento diferentes para cada par de filas, un registro superior y un registro inferior. Debido a esto, el panel se divide en una mitad superior y una mitad inferior, como se observa en la figura 3.15.

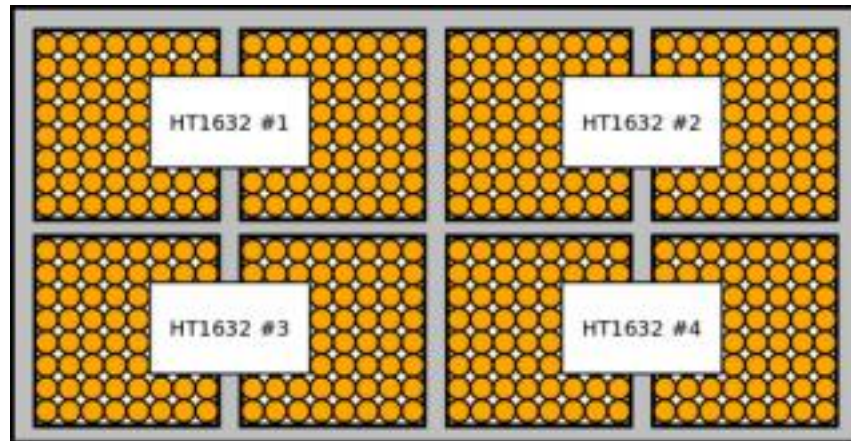


FIGURA 3.15: Panel LED RGB 16x32, compuesto por 8 matrices LED 8x8
FUENTE: Página Web statusbits.wordpress.com. Consultada en junio 2016

Los datos son desplazados en la parte superior a través de las señales de B1 R1, G1 que son las 3 primeras entradas de data en el conector. Los datos de la parte inferior son suministrados por las señales de B2 R2, G2 que son las segundas 3 entradas de data en el conector. En total son 6 entradas de data, como se observa en la figura 3.16. Entonces, dado que el panel es de 32 píxeles de ancho, debemos usar 2 registros de desplazamiento para mantener los 32 bits de datos de 32 píxeles para una sola fila. Esto crea las mitades izquierda y derecha de la pantalla. Cabe señalar que, esto se refiere a un solo color, para simplificar. En realidad, cada color tiene su propio registro de desplazamiento, así que de hecho se necesita 6 registros de desplazamiento para una sola fila.

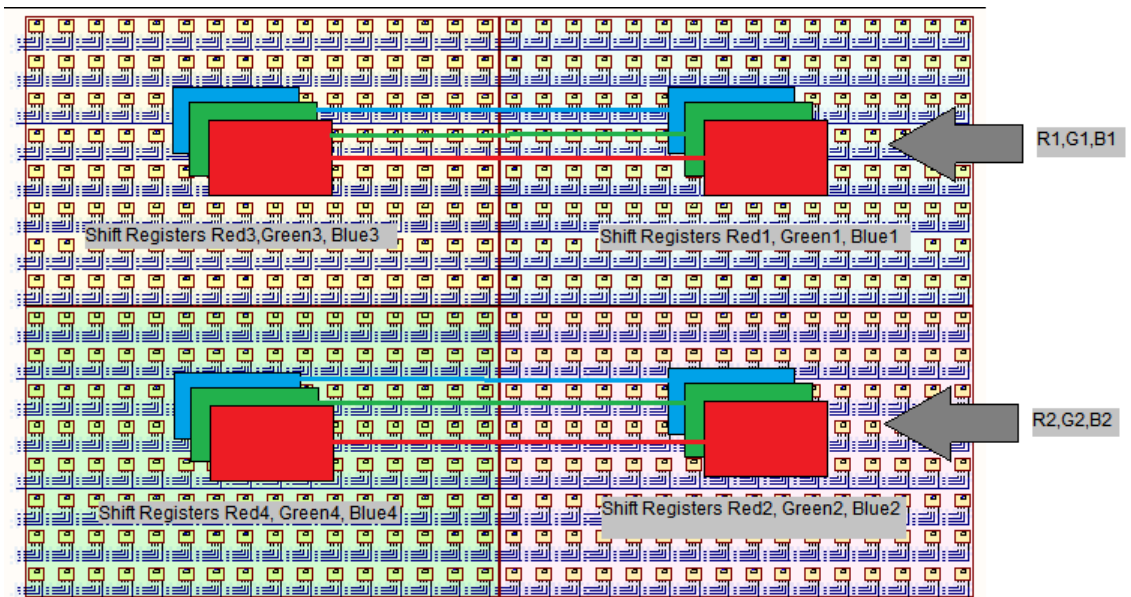


FIGURA 3.16: Diagrama indicando la operación de los registros en cada cuadrante del panel LED RGB

FUENTE: Página Web <http://i.stack.imgur.com/2HjZF.png>. Consultada en junio 2016

De todo esto se puede inferir que, si se quisiera tener un registro de desplazamiento para cada fila, eso requeriría 16 filas * 2 mitades de pantalla = 32 registros de desplazamiento, para un solo color. Se necesitaría 96 registros de desplazamiento para los 3 colores si se utilizara este método. Obviamente, esto no es muy factible, por 2 razones muy importantes; por la complejidad del diseño (96 IC's frente a 12 IC's) y por el tremendo costo que este método implicaría. Es por ello que para remediar esto, básicamente, se utiliza la técnica de multiplexación, si se hace lo suficientemente rápido, la pantalla aparecerá sin parpadeos a la vista. Cada cuadrante del panel es controlado por 3 registros de desplazamiento, uno para cada color. En primer lugar, la fila 0 y fila 8 se seleccionan. 32 bits de datos son desplazados en el registro de desplazamiento de cada color (R1, G1, B1), y después estos datos son almacenados en los latches de los registros. Al mismo tiempo, 32 bits de datos también se desplazan en los registros de desplazamiento de cada color

para la parte media inferior (R2, G2, B2), y después también son almacenados en los latches de estos registros. El proceso se repite 7 veces más, y cada vez seleccionando el par de filas siguientes, hasta que cada fila ha sido actualizada.

TABLA 3.3: Descripción del pinout del conector de entrada HUB75

Signal name	Pin number	Description
A	9	The lowest bit of row address
B	10	The second lowest bit of row address
C	11	The highest bit of row address
LE	14	Latch of row
CLK	13	Clock
EN	15	Enable
R1	1	Red data 1
R2	5	Red data 2
G1	2	Green data 1
G2	6	Green data 2
B1	3	Blue data 1
B2	7	Blue data 2
GND	4,8,12,16	Grounding

FUENTE: Página Web <http://www.rayslogic.com>. Consultada en junio 2016

Cada cuadrante se controla mediante 3 registros de desplazamiento, uno para cada color R, G, B. Este panel se describe como un panel LED RGB de 1/8 de scan rate o un 1/8 de velocidad de barrido, (el cual es un estándar en la industria, existen otros paneles que utilizan 1/2, 1/4, 1/16, 1/32, de scan rate) ya que se puede ver que se tarda 8 actualizaciones muy rápidas para dibujar el panel completo o un frame de datos.

En este diseño del prototipo de panel LED RGB se está utilizando un conector de entrada estándar que es conocido como conector HUB75, como se muestra en la figura 3.17, el cual es usado para transmitir las señales de control y data de los paneles comerciales usados en publicidad y eventos.

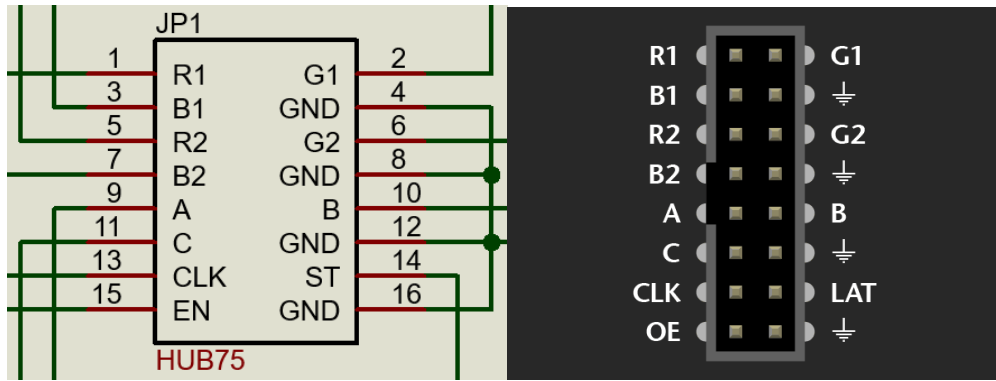


FIGURA 3.17: Conector HUB75 mostrando las señales de entrada

FUENTE: Elaboración propia

Como se observa en el esquema del conector de entrada del panel LED RGB de 16x32, existen 12 señales de entrada para manejar este panel (R1, G1, B1, R2, G2, B2, CLK, ST, EN, A, B, C) y 13 señales si es que se extrapola a un panel de 32x32, como se muestra en la tabla 3.3. (se agrega la señal D, que es una línea de dirección hacia el decodificador).

Estas señales provienen de la placa controladora, como se observa en la figura 3.18, en este caso la micro computadora Raspberry Pi 3, y todas estas señales se dirigen, en primera instancia, hacia 2 IC's 74HC245, el cual es un transceptor octal de bus paralelo que posee salidas 3 estados (tri-state) no inversoras, en ambas direcciones, tanto para enviar como para recibir, por lo que los datos presentes en el lado izquierdo son transmitidos a el lado derecho o viceversa, como se muestra en la figura 3.19.

En este prototipo se utiliza estos transceptores para aumentar o impulsar las señales provenientes de la placa Raspberry Pi 3 (ya que esta trabaja con salidas a 3.3V), así como para brindar el aislamiento necesario y proporcionar una distribución optima de las señales en diferentes lugares del circuito.

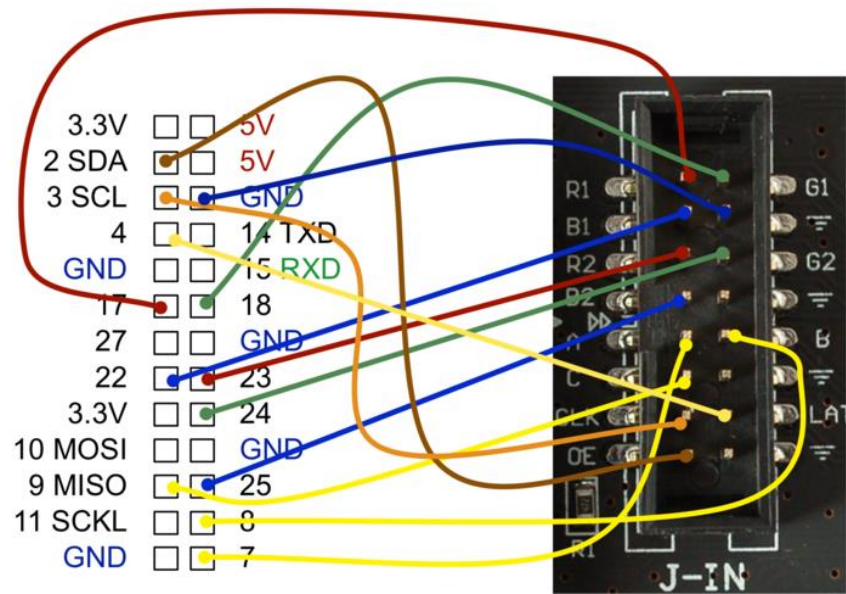


FIGURA 3.18: Conexión de los pines GPIO de la placa Raspberry Pi 3 hacia el Conector de entrada HUB75 del prototipo de panel LED RGB

FUENTE: Elaboración propia

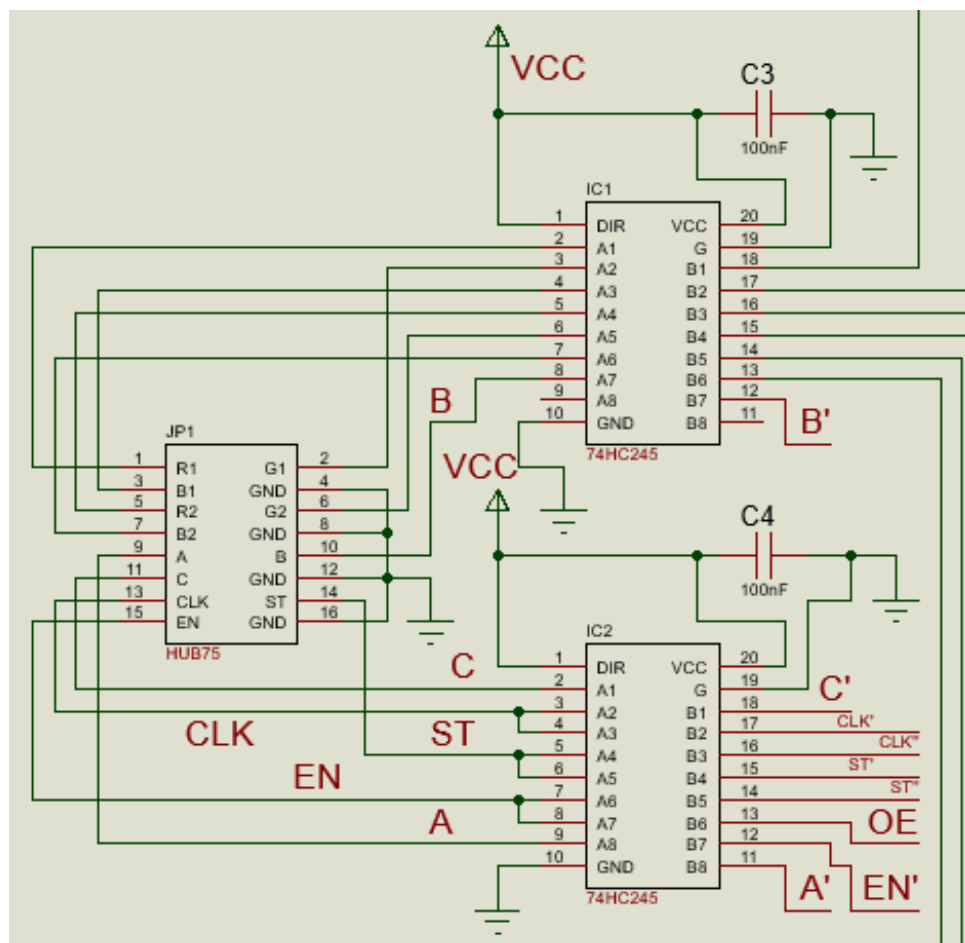


FIGURA 3.19: Esquema del HUB75 conectado hacia los IC's 74HC245

FUENTE: Elaboración propia

Después de pasar por el 74HC245 las señales han sido impulsadas y se distribuyen uniformemente por todo el circuito, básicamente las 6 señales de data (R1, G1, B1, R2, G2, B2), son enviadas hacia las entradas seriales de cada uno de los 6 primeros registros de desplazamiento (3 en la mitad superior y 3 en la mitad inferior), (1 registro de desplazamiento por cada color) como se observa en las figuras 3.20, 3.21 y 3.22.

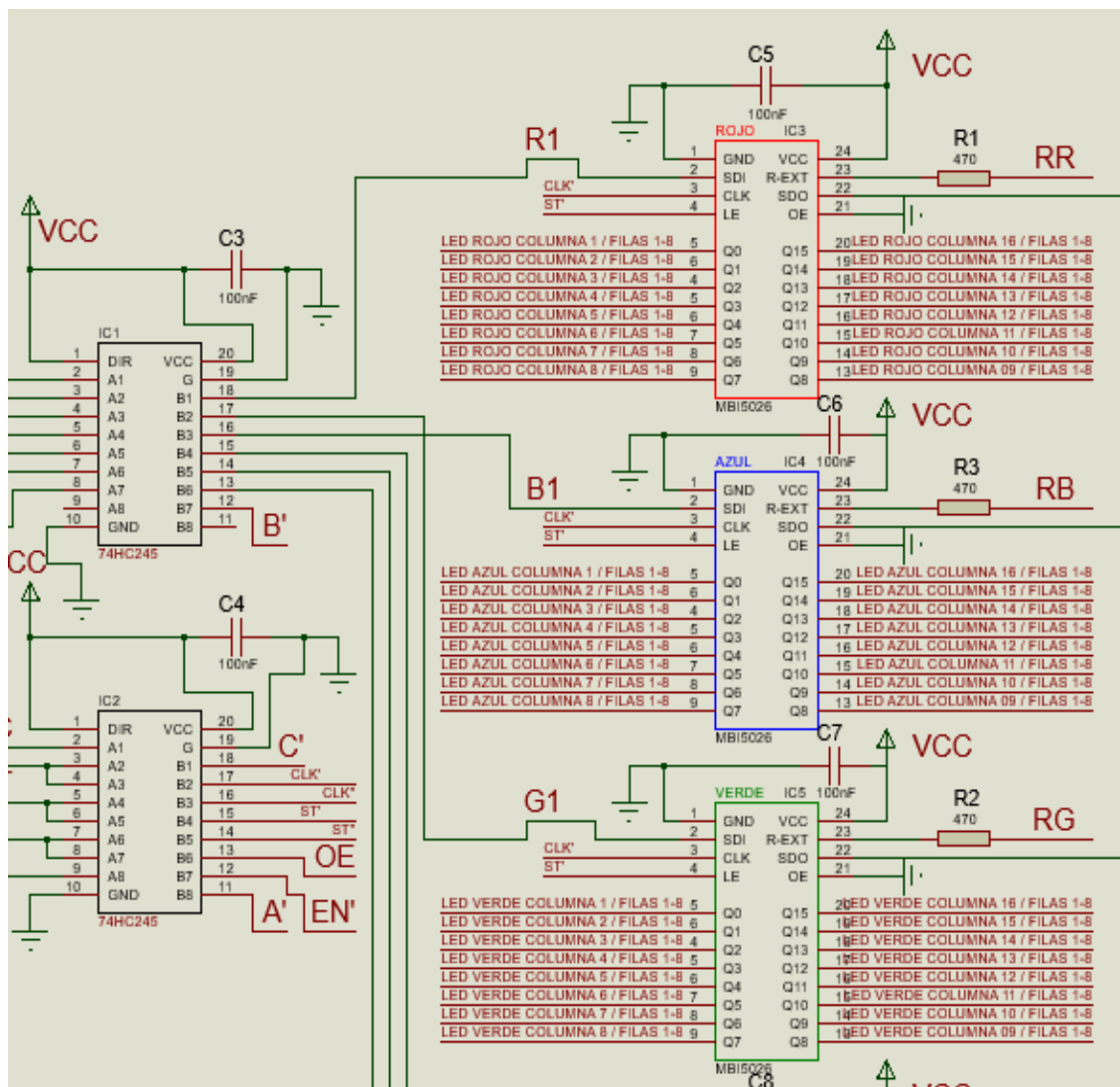


FIGURA 3.20: Señales R1, G1, B1 ingresando hacia los 3 primeros IC's MBI5026 de la mitad superior del panel LED (filas 1-8)

FUENTE: Elaboración propia

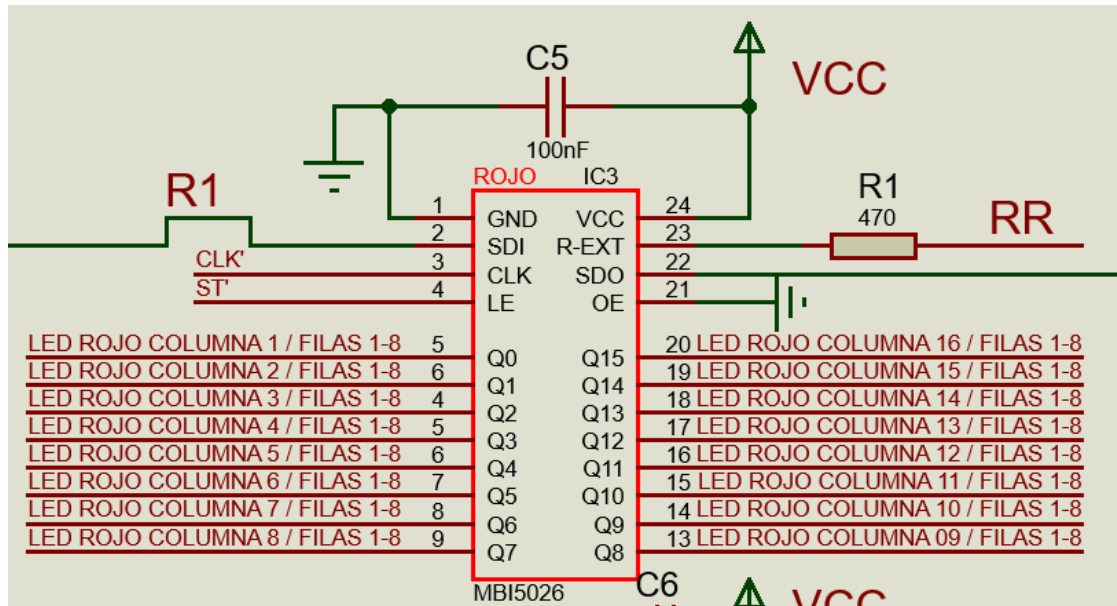


FIGURA 3.21: IC MBI5026 mostrando su conexión con las columnas 1-16

FUENTE: Elaboración propia

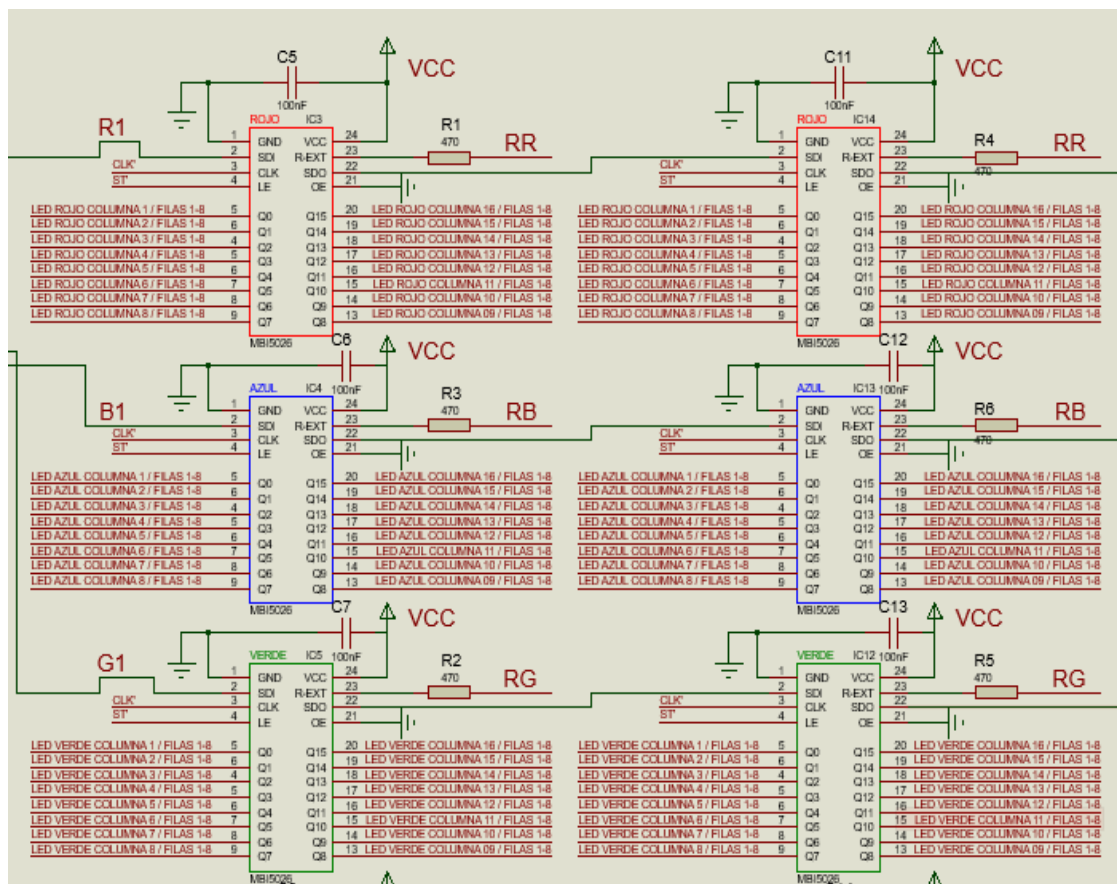


FIGURA 3.22: Distribución de las señales R1, G1, B1, CLK, ST ingresando hacia los IC's MBI5026 de la mitad superior del panel LED (filas 1-8)

FUENTE: Elaboración propia

También son enviadas a estos registros las señales CLK (reloj), ST (strobe o latch), pero a diferencia de las señales de data estas últimas se envían a todos los registros de desplazamiento del circuito. La señal EN (Enable), es una señal importante en el diseño de este prototipo y si no se presta especial atención a su habilitación y en su conexionado hará que el panel LED RGB no funcione en absoluto.

La señal EN (también llamada OE) necesita ser puesta a nivel bajo para que el panel pueda estar habilitado para desplegar algún tipo de información. Básicamente cuando la placa Raspberry Pi envía la señal EN (OE), la señal pasa a través del 74HC245 de A6 hacia B6 y se conecta a la patilla 4 y 5 (G2A / G2B) del 74HC138 que es la función de activación. cuando G2 es puesto a nivel alto, se apagan todas las salidas del decodificador 74HC138 lo que significa que ninguna de las filas está activa.

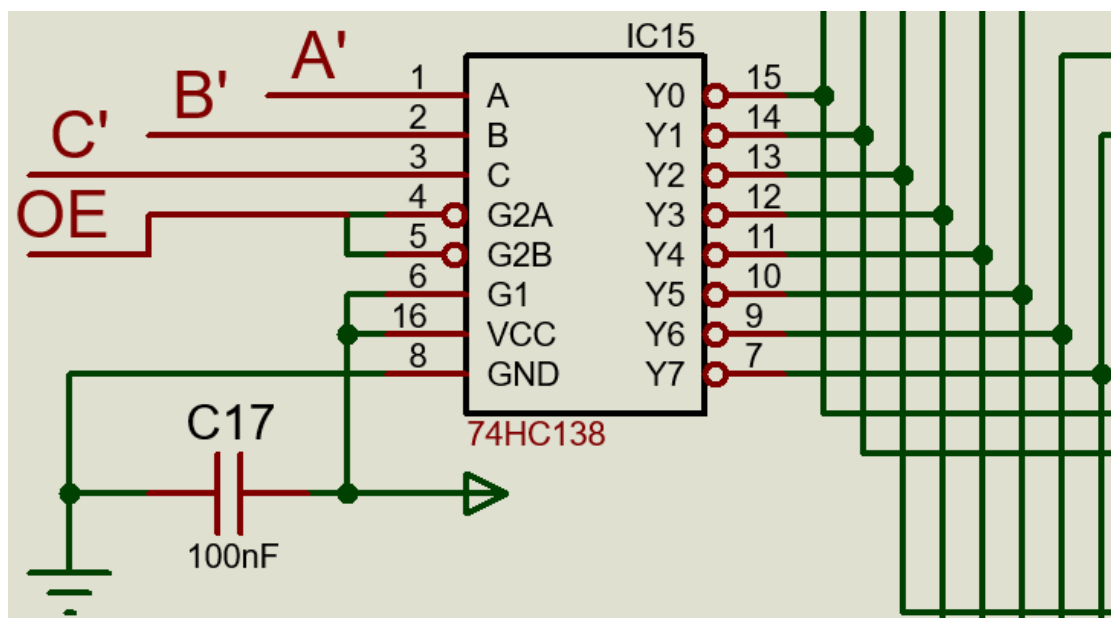


FIGURA 3.23: Esquema del conexionado de las entradas de direcciones A, B, C del IC 74HC138 y la entrada de habilitación OE.

FUENTE: Elaboración propia

Como se observa en la figura 3.23, las señales de direccionamiento de filas A, B, C son enviadas a las entradas de control del 74HC138 para la respectiva selección de filas, cada fila está controlada por un transistor MOSFET APM4953 que actúa como interruptor según qué fila se seleccione (Artschip Electronics, 2016). A la salida de la patilla de drenador de cada transistor MOSFET se conecta cada uno de los respectivos ánodos de las matrices que conforman el presente panel LED RGB, (todos los módulos de matrices LED RGB 8x8 usados en este diseño son de ánodo común) y los cátodos de los LED's van conectados hacia los registros de desplazamiento. Para este primer diseño las resistencias limitadoras no son necesarias ya que el IC MBI5026 (registro de desplazamiento) regula y limita la corriente que consumen los LED's gracias a una sola resistencia externa que se configura junto con el MBI5026 que determina el nivel de corriente deseado para todos los LED's conectados al registro (Macroblock Inc, 2004). Sin embargo, como se verá más adelante, en el segundo diseño, si será necesario la incorporación de resistencias limitadoras a las salidas de los registros de desplazamiento ya que estos registros (74HC595) no poseen dicha capacidad de regular y limitar la corriente por si solos. Cabe resaltar que en cualquier caso las salidas actúan como salidas tipo "sinking current", como se muestra en la figura 3.24. Como se sabe por teoría de circuitos electrónicos una salida digital de tipo "sinking" provee una referencia (tierra), mientras que una entrada o salida digital de tipo "sourcing" genera un voltaje fuente. En este diseño lo que se necesita a la salida de los registros de desplazamiento, para encender un LED, es una referencia a tierra, ya que lo que manejan los registros de desplazamiento son cátodos de LED's. Por esta

razón, en el segundo diseño, se utiliza un IC 74HC04 para invertir las 6 señales de data (R1, G1, B1, R2, G2, B2) antes que pasen hacia las entradas seriales de cada uno de los 6 primeros registros de desplazamiento (3 en la mitad superior y 3 en la mitad inferior), (1 registro de desplazamiento por cada color) como se observa en el esquema del segundo diseño (ver pág. 74). En el primer diseño no se necesita este IC 74HC04, ya que el MBI5026 invierte la señal internamente ya que sus salidas son de tipo colector abierto (open collector output).

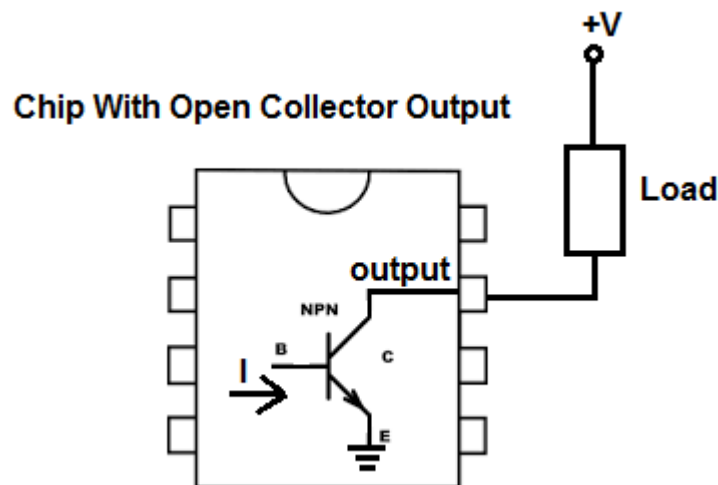


FIGURA 3.24: Salida digital de tipo colector abierto y sinking current.

FUENTE: Página Web learningaboutelectronics.com. Consultada en junio 2016

3.2.3 DISEÑO ALTERNATIVO DE PROTOTIPO DE PANEL RGB 16X32

En este segundo diseño (diseño alternativo) todas las funciones son iguales al primer diseño solo cambian algunos componentes tales como los registros de desplazamiento MBI5026 de 16 bits que serán reemplazados por los registros de desplazamiento 74HC595 de 8 bits, como se observa en la figura 3.25, por lo cual, en vez de 12 ahora se necesitarán 24 de estos IC's,

en esencia la función es la misma con la diferencia que el IC MBI5026 tiene mayores prestaciones.

El conexionado es prácticamente el mismo como se aprecia en la figura 3.26 de este diseño final.

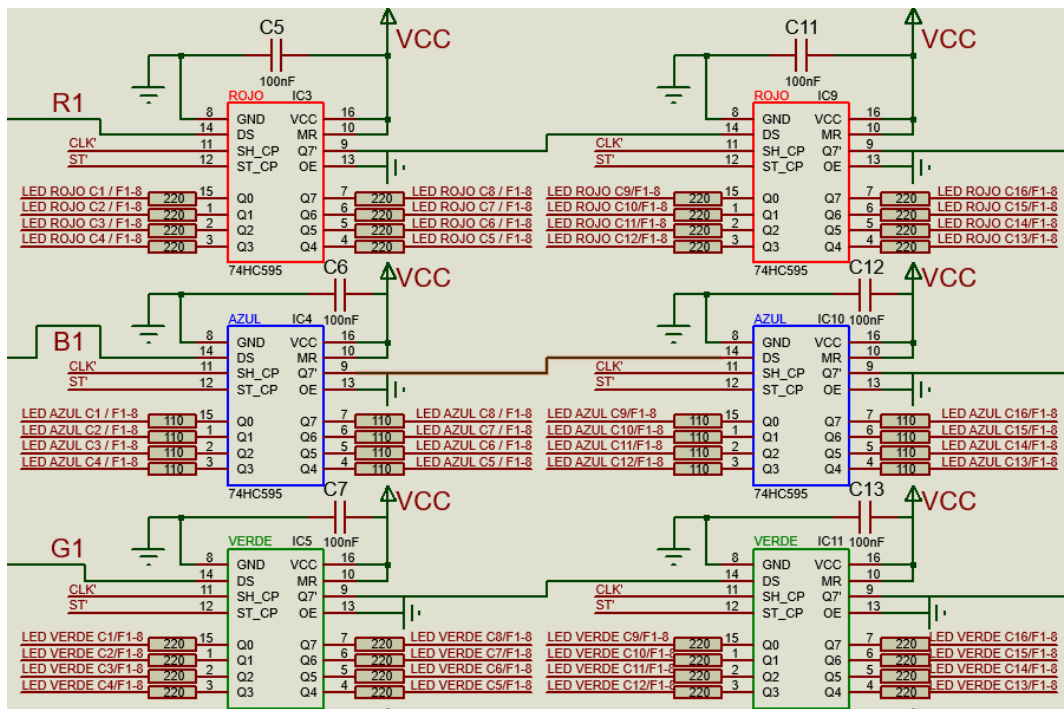


FIGURA 3.25: Esquema del conexionado de los IC's 74HC595 en reemplazo de los IC's MBI5026.

FUENTE: Elaboración propia

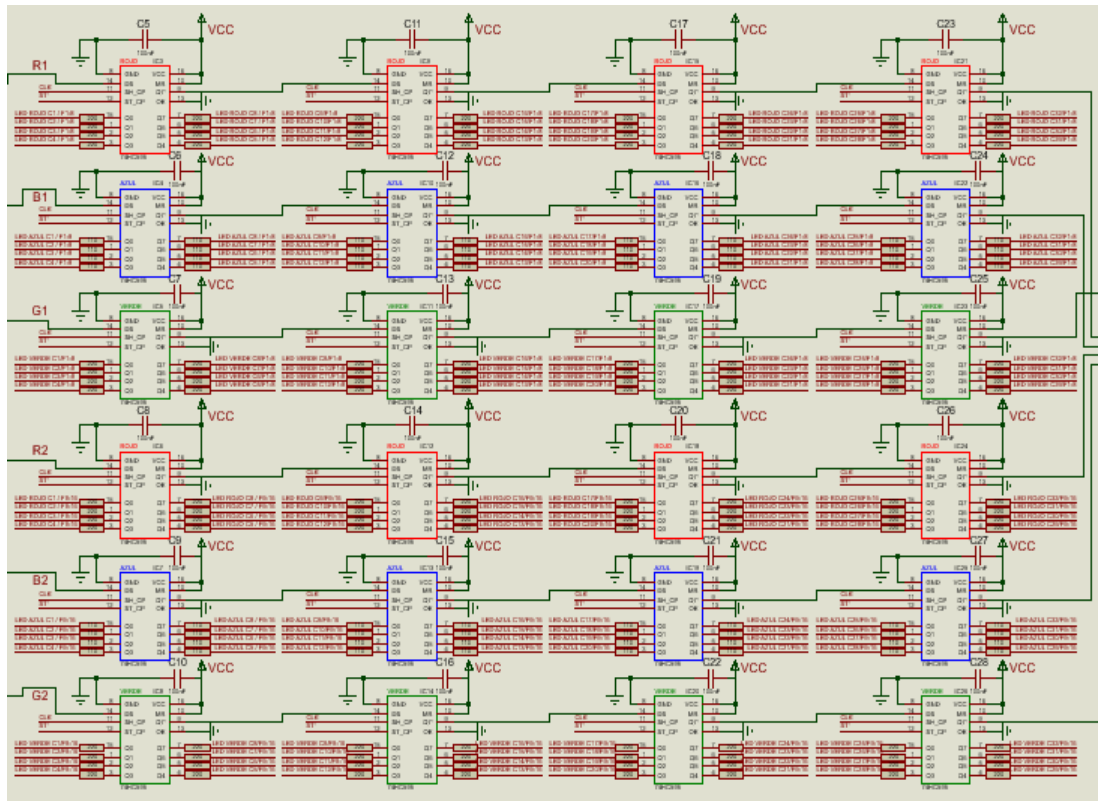


FIGURA 3.26: Esquema del conexionado de los 24 IC's 74HC595

FUENTE: Elaboración propia

Como se observa en la figura 3.26, se necesitó la incorporación de resistencias limitadoras a las salidas de los registros de desplazamiento ya que estos registros (74HC595) no poseen la capacidad de regular y limitar la corriente por si solos como la posee el IC MBI5026. Cabe resaltar que en cualquier caso las salidas actúan como salidas tipo “sinking current”.

En ambos diseños, como se mencionó anteriormente, lo que se necesita a la salida de los registros de desplazamiento, para encender un LED, es una referencia a tierra, ya que lo que manejan los registros de desplazamiento son cátodos de LED's. Por esta razón, en este presente diseño, se utiliza un IC 74HC04 para invertir las 6 señales de data (R1, G1, B1, R2, G2, B2) antes que pasen hacia las entradas seriales de cada uno de los 6 primeros registros

de desplazamiento (3 en la mitad superior y 3 en la mitad inferior), (1 registro de desplazamiento por cada color) como se observa en la figura 3.27.

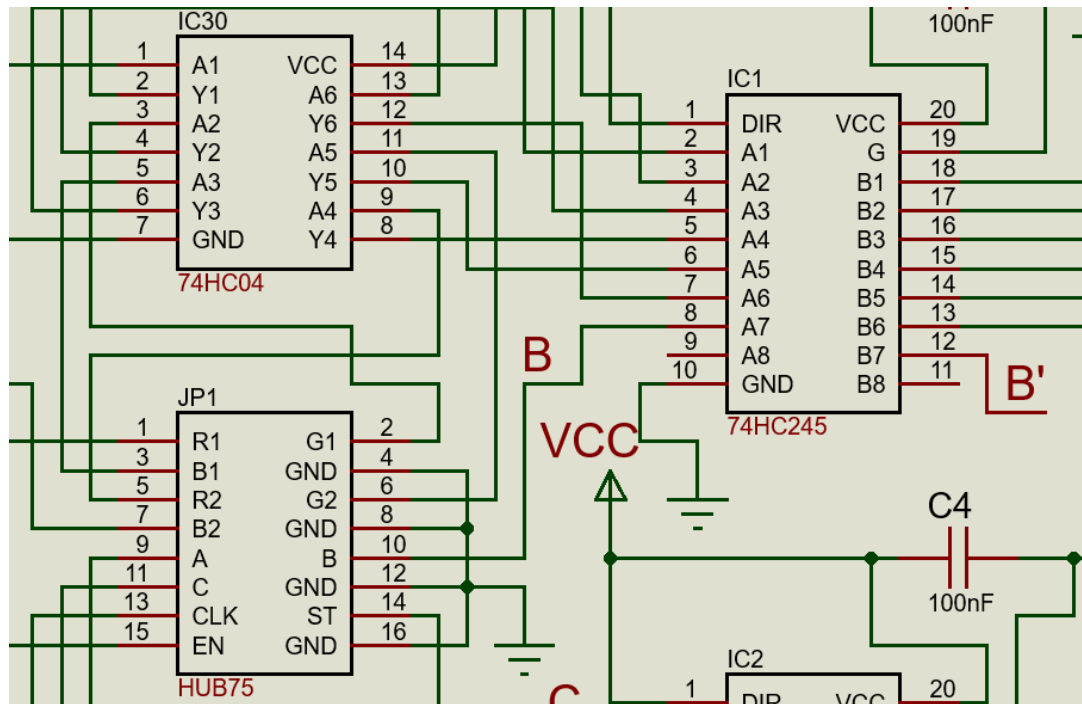


FIGURA 3.27: Esquema del conector de entrada con las señales de data (R1, G1, B1, R2, G2, B2) invertidas a través del IC 74HC04

FUENTE: Elaboración propia

En este diseño se necesita el IC 74HC04, ya que el 74HC595 no invierte la señal internamente como lo hace el MBI5026 ya que sus salidas son de tipo colector abierto (open collector output).

Como se observa en la figura 3.28, se están cambiando los 16 transistores MOSFET APM4953 por 16 transistores BJT NPN BD135, como se aprecia el número de transistores es el mismo, pero en el primer diseño aparecen 8 debido a que en cada uno hay 2 transistores MOSFET encapsulados.

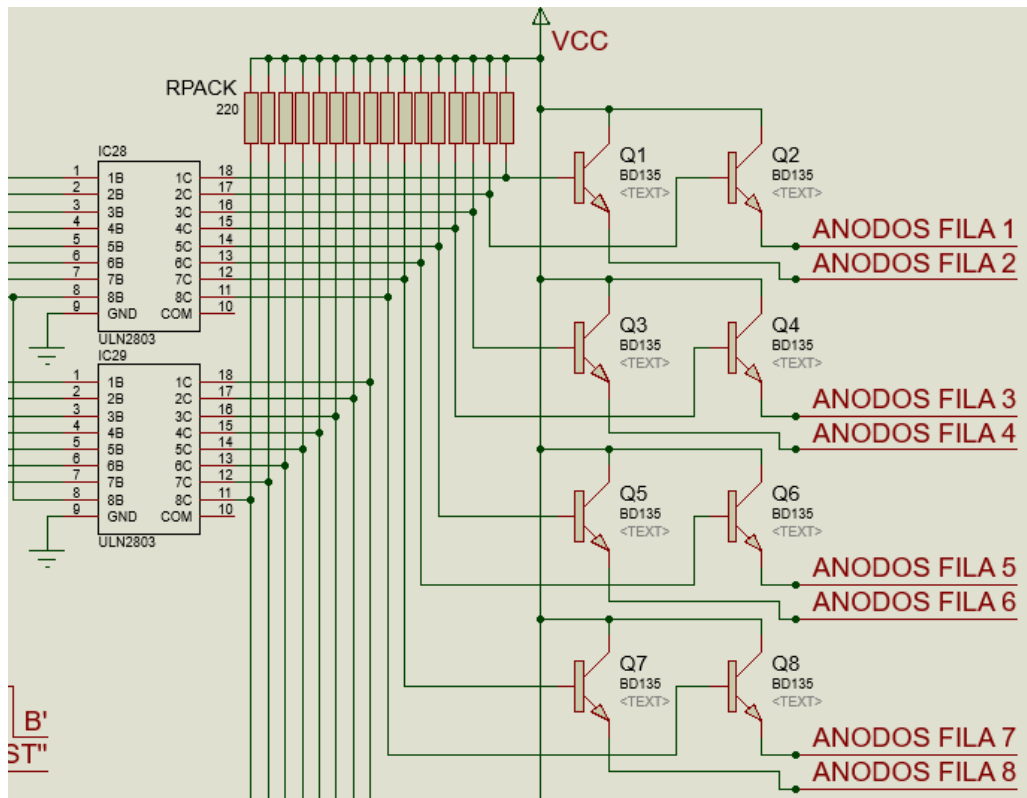


FIGURA 3.28: Esquema del conexionado de los transistores BJT BD135 en reemplazo de los transistores APM4359

FUENTE: Elaboración propia

Para poder hacer la equivalencia exacta con los transistores MOSFET se necesitó agregar un circuito integrado a la salida del decodificador 74HC138, como se observa en la figura 3.29, este es el IC ULN2803, con el cual y adicionando resistencias de pull-up en las salidas de este IC, se logró hacer compatibles las salidas del decodificador con las bases de los transistores BD135. Como se sabe, por teoría de circuitos electrónicos, las bases de los transistores BJT trabajan con corriente y los gates de los transistores MOSFET trabajan con tensión, es por ello la necesidad de IC's y componentes que hagan de interface para que sean totalmente compatibles.

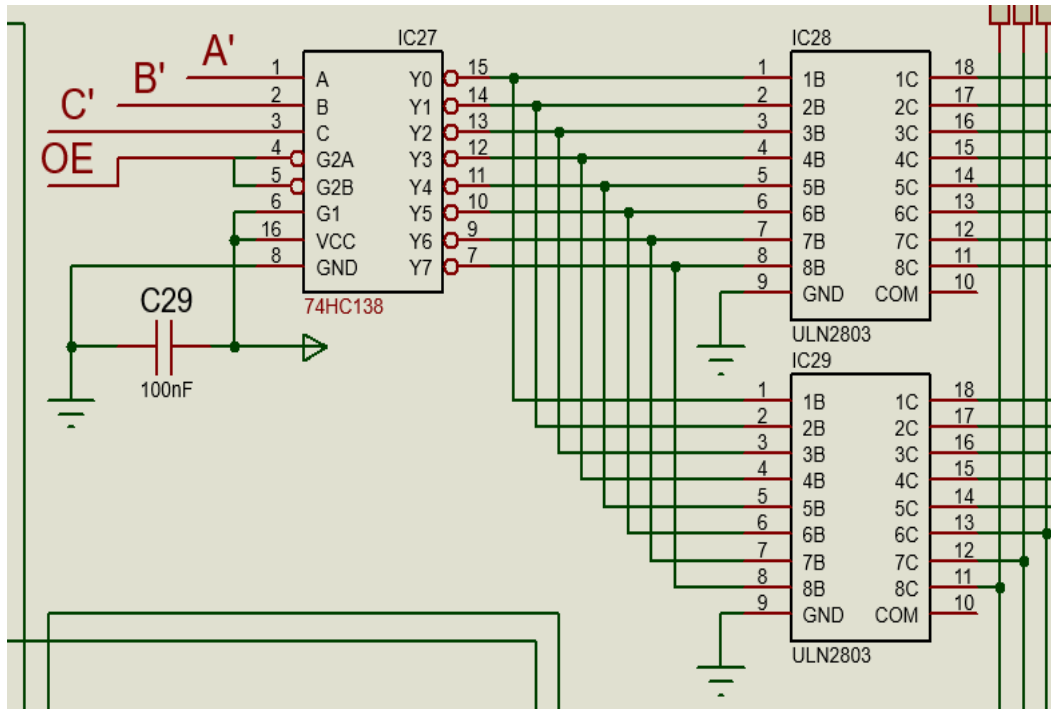


FIGURA 3.29: Esquema del conexionado del IC 74HC138 con sus salidas hacia los IC's ULN2803

FUENTE: Elaboración propia

Como se observa se debe poner condensadores de desacoplo de 100nF en cada IC que se utilice en este proyecto, así como también condensadores electrolíticos de filtrado de 1000uF en la entrada de alimentación Vcc y GND.

Para un solo módulo de matrix LED RGB de 8x8 (ánodo común) el conexionado utilizando protoboards se muestra en la figura 3.30.

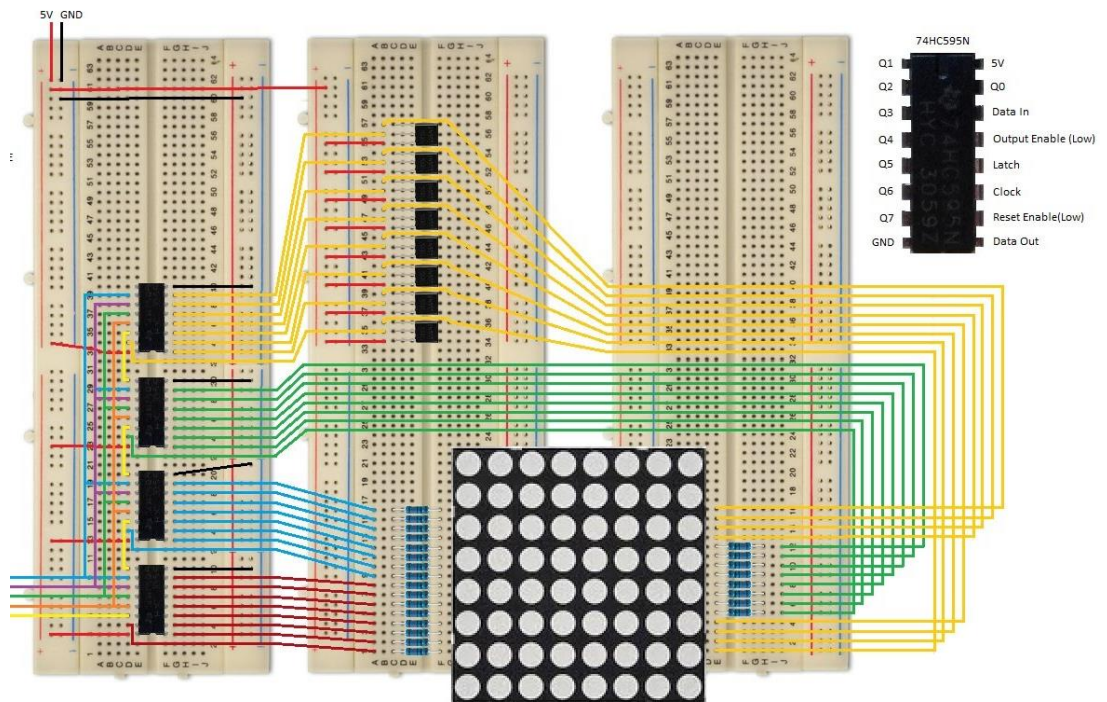


FIGURA 3.30: Conexión necesario para un solo módulo de matrix LED RGB 8x8 con sus respectivos IC's, resistencias y transistores.

FUENTE: www.i.ytimg.com. Consultada en junio del 2016

A este conexionado se debe agregar 7 módulos más para formar el presente prototipo de panel LED RGB de 16x32.

Básicamente estos serían todos los cambios que se aplican en este segundo diseño, el cual como ya se mencionó cumple con todas funcionalidades del primero, y con las ventajas de la fácil disponibilidad de los componentes y menores costos.

3.2.4 PROGRAMACIÓN DEL PANEL LED RGB 16X32

Para empezar a utilizar Raspberry Pi 3, se necesita un sistema operativo. NOOBS (New Out Of Box Software) es un gestor e instalador de sistemas operativos para la placa Raspberry Pi 3, se recomienda utilizar una tarjeta micro SD de al menos 8GB.

Una vez la tarjeta ha sido preparada y NOOBS ha sido copiado, la placa Raspberry Pi se reinicia y el primer programa que se ejecuta es Pi Recovery, como se observa en la figura 3.31, este programa sirve para instalar la versión del sistema operativo deseada. El sistema operativo con el cual se trabajó en este proyecto es el S.O. Raspbian Jessie (Versión del Kernel 4.4).

En la primera pantalla inicial, se selecciona Raspbian para instalar el sistema operativo. Este proceso tomara entre 15-20 minutos.

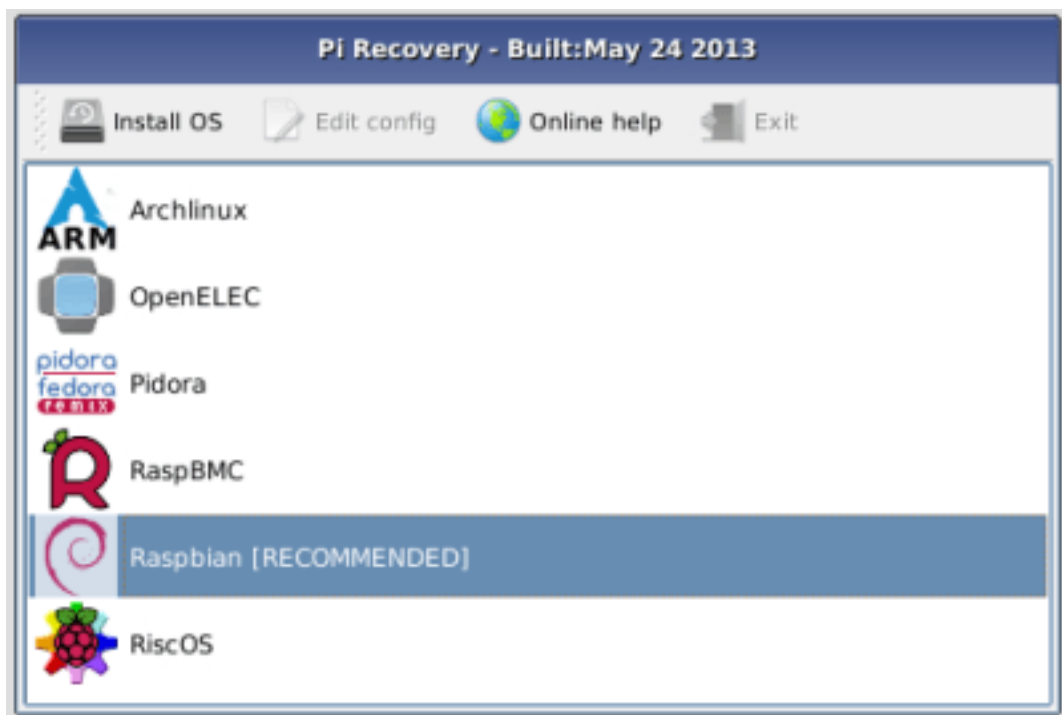


FIGURA 3.31: Pi Recovery con distintas opciones de sistemas operativos

FUENTE: www.frambuesapi.co. Consultada en junio del 2016

Después que Raspbian “Jessie” ha sido instalado, toca configurar el Sistema operativo para que funcione en español. Los siguientes pasos muestran las diferentes opciones disponibles.

Al reiniciar el dispositivo el primer programa que se ejecuta se llama “raspi-config”, este programa solo se ejecuta en inglés. Más adelante, cuando ya se

haya instalado el sistema operativo y se desee realizar alguna de estas modificaciones, se puede hacer ejecutando el siguiente comando desde la terminal: `sudo raspi-config`.

El primer menú, que se muestra en la figura 3.32, contiene 9 diferentes opciones disponibles, a continuación, se revisara 2 de las opciones más importantes:

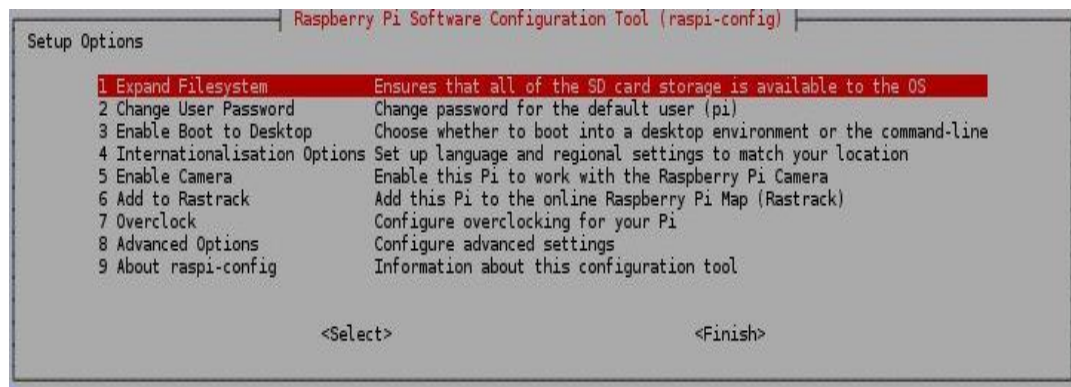


FIGURA 3.32: Menú principal del S.O. que muestra las opciones de setup.

FUENTE: www.frambuesapi.co. Consultada en junio del 2016

La opción 1, expandir el sistema de archivos (Expand Filesystem) permite expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta. Cuando se instala Raspbian "Jessie" la imagen copiada en la tarjeta solo ocupa 3 GB, por lo tanto, es necesario ejecutar esta opción para que todo el espacio de la tarjeta SD sea utilizado.

Si el sistema operativo fue instalado utilizando NOOBS, no es necesario ejecutar esta opción. Ya que el sistema operativo ya ha sido expandido.

La opción 3, activar el escritorio al iniciar (Enable Boot to Desktop) permite que la placa Raspberry Pi, después de iniciar el sistema, comience inmediatamente el escritorio en modo gráfico o en línea de comando. Como se muestra en la figura 3.33. En caso se inicie en modo de comando y

después se desea ingresar al modo gráfico solo se ingresa el siguiente código: startx.

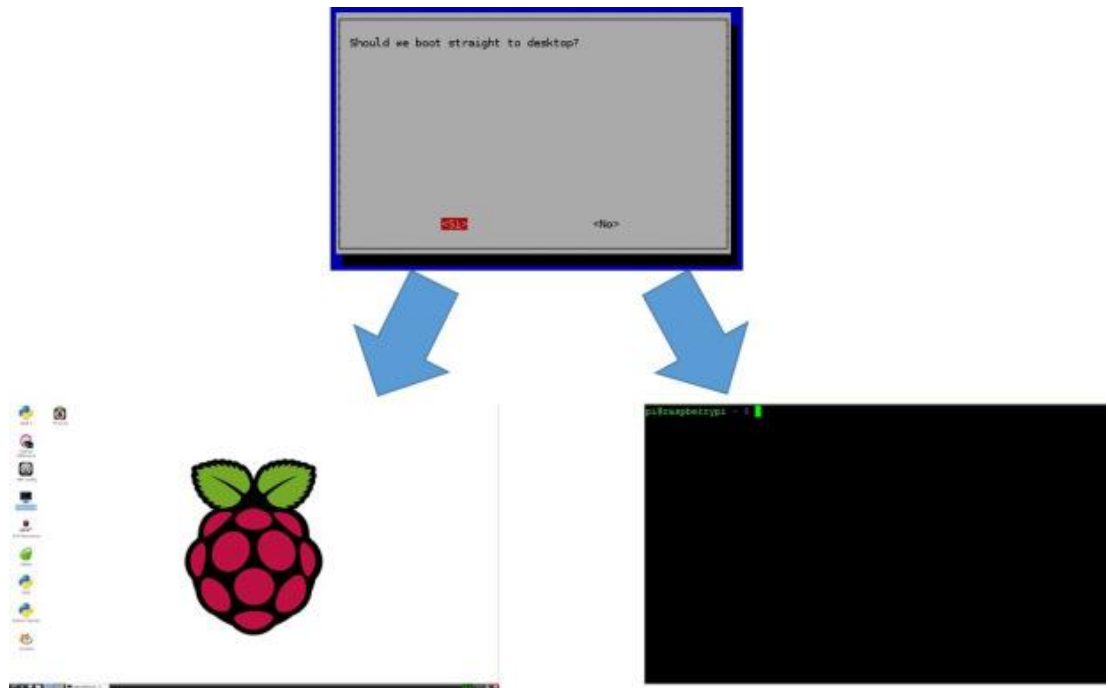


FIGURA 3.33: A la izquierda: escritorio de Raspbian en modo gráfico; a la derecha en modo línea de comandos.

FUENTE: www.frambuesapi.co. Consultada en junio del 2016

Luego de configurar la placa Raspberry Pi y conectarla a la red se debe empezar por la creación de un directorio en donde se guarde el código. Por lo cual se debe abrir una sesión en el terminal LX, como se muestra en la figura 3.34, (se puede usar SSH si se prefiere) y se escribe los siguientes comandos.

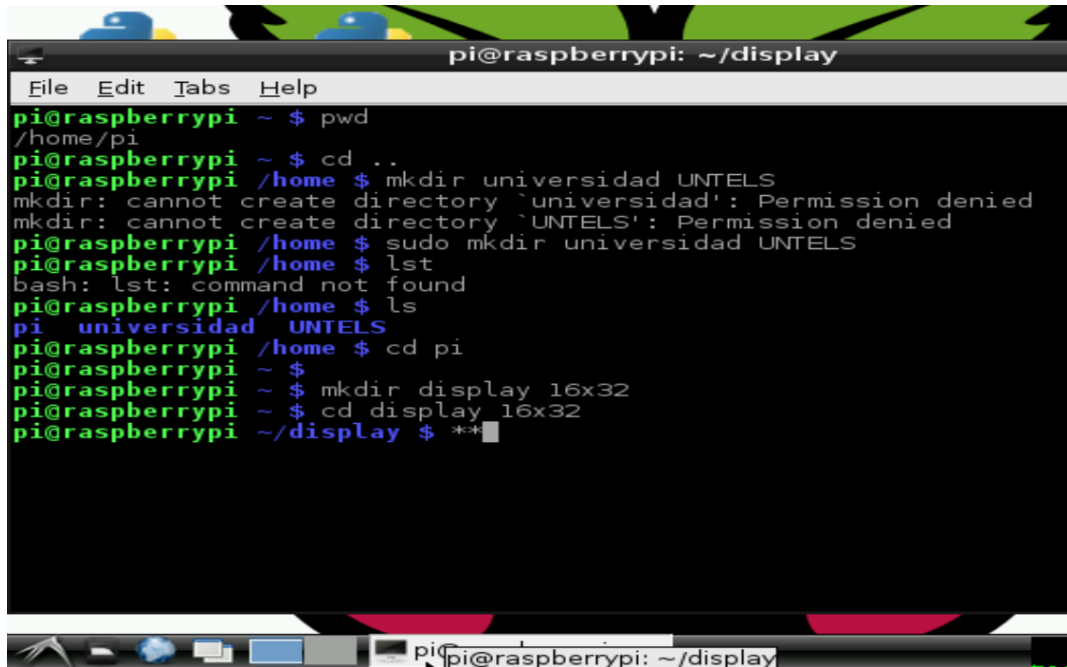
```
$ mkdir display16x32
```

```
$ cd display16x32
```

Ahora se debe buscar y descargar el código del proyecto que se ha subido y se encuentra alojado en mi repositorio de la página web de GitHub que se

encuentra con el nombre de “juanmayta/PROTOTIPO-PANEL-LED-RGB” esto se hace a través del comando:

```
$ git clone https://github.com/juanmayta/PROTOTIPO-PANEL-LED-RGB
```



```
pi@raspberrypi: ~/display
File Edit Tabs Help
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $ cd ..
pi@raspberrypi /home $ mkdir universidad UNTELS
mkdir: cannot create directory `universidad': Permission denied
mkdir: cannot create directory `UNTELS': Permission denied
pi@raspberrypi /home $ sudo mkdir universidad UNTELS
pi@raspberrypi /home $ lst
bash: lst: command not found
pi@raspberrypi /home $ ls
pi  universidad  UNTELS
pi@raspberrypi /home $ cd pi
pi@raspberrypi ~ $ mkdir display 16x32
pi@raspberrypi ~ $ cd display 16x32
pi@raspberrypi ~/display $ **
```

FIGURA 3.34: Creación del directorio “display 16x32” usando el terminal LX

FUENTE: Elaboración propia

Este código está en código fuente C, por lo que se necesita compilar antes de que se pueda ejecutar, además también se debe habilitar una línea de código que hace referencia al pinout utilizado en la raspberry pi (conexión de pines GPIO), esto se hace con los siguientes comandos:

```
$ cd rpi-rgb-led-matrix/lib
```

```
$ nano Makefile
```

En la última línea de comando se está utilizando el editor de texto “nano” pero se puede utilizar cualquier otro editor de texto para habilitar la siguiente línea de código que se encuentra en el archivo “makefile”:

```
DEFINES+=-DRGB_CLASSIC_PINOUT
```

Una vez hecho esto se guardan los cambios, salimos del editor y escribimos los siguientes comandos para compilar el código e iniciar el programa.

```
$ cd ..
```

```
$ make
```

Ahora para correr y probar el programa se ingresa el siguiente comando:

```
$ sudo ./led-matrix
```

El programa de prueba puede mostrar mensajes en movimiento, figuras, formas, etc. Se recomienda leer el archivo de información que se acompaña al código y a las librerías, las cuales se encuentran en mi repositorio personal en GitHub, en el link mostrado anteriormente. Se puede crear mensajes en movimiento e imágenes para desplegar en el panel usando el programa GIMP v2.8, como se observa en la figura 3.35, el cual es un programa de código abierto.

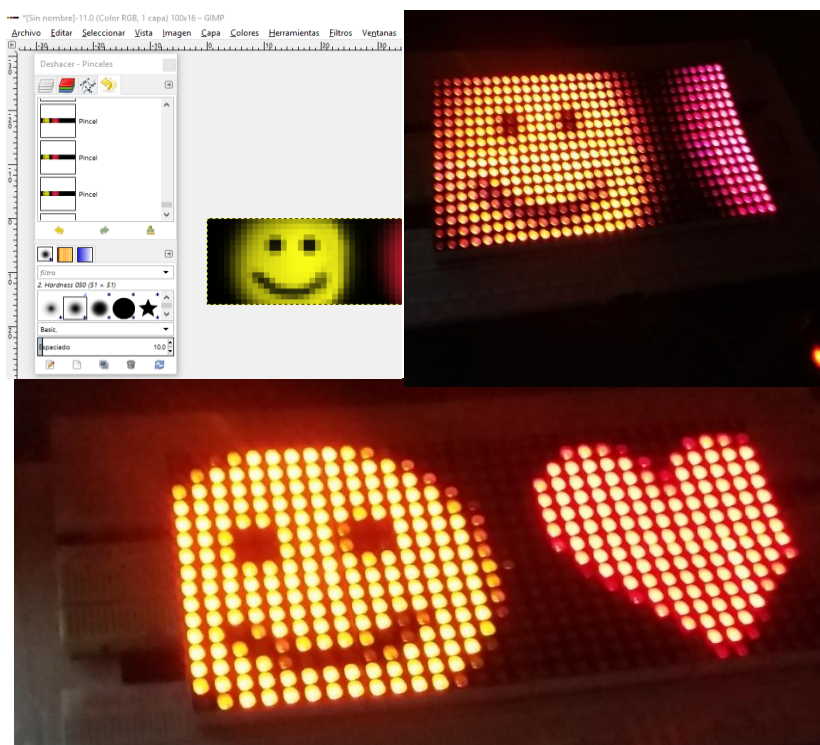


FIGURA 3.35: A la izquierda: creación de una imagen utilizando el software GIMP; A la derecha: la misma imagen en el prototipo de panel LED RGB

FUENTE: Elaboración propia

3.2.5 IMPLEMENTACIÓN DEL PANEL LED RGB 16X32

Se implementó el prototipo de panel LED RGB 16x32 de acuerdo al segundo diseño planteado. Se empezó por probar el correcto funcionamiento de los 24 IC's 74HC595 haciendo pruebas individuales a estos y a cada módulo de matriz led 8x8, luego se realizó el conexionado de cada IC con sus respectivas resistencias y capacitores, tal como se observa en la figura 3.36. Después se realizó el cableado de los 8 módulos de matriz LED RGB 8x8 hacia los registros de desplazamiento y los ánodos hacia los 16 transistores BD135.



FIGURA 3.36: Fase inicial de la implementación del prototipo, conexionado y ensamblado de los módulos 8x8 de las matrices LED RGB

FUENTE: Elaboración propia

Se realizó pruebas preliminares conforme se fue ensamblando el prototipo, en las cuales se corrigieron fallos y se aseguró un correcto funcionamiento de todo el sistema, tal como se observa en la figura 3.37.

Luego de esto, se realizó una interface de entrada para la conexión de los pines GPIO de la placa Raspberry Pi, ya que estos trabajan con 3.3V y el prototipo de panel LED RGB trabaja con 5V. Para esta interface se utilizó los IC's 74HCT245. Seguidamente, se realizó el conexionado de los pines GPIO hacia el prototipo de panel LED RGB. Se utilizó una fuente de 3A para la alimentación del prototipo de panel LED RGB 16x32.



FIGURA 3.37: Pruebas iniciales en el prototipo de panel LED RGB con 2 matrices 8x8 ensambladas.

FUENTE: Elaboración propia

Luego de ensamblar todos los módulos de matriz LED RGB 8x8 y conectarlos hacia sus respectivos 24 drivers IC's 74HC595, así como hacia sus respectivos transistores BD135 y, estos a su vez, conectados hacia sus

respectivos drivers ULN2803 y al driver IC 74HC138, tal como se muestra en la figura 3.38, se pasó a probar el funcionamiento de todo el sistema en conjunto, para ellos previamente se realizó la configuración de la placa raspberry pi 3, y se cargó un programa de prueba básico para iluminar el panel por primera vez.

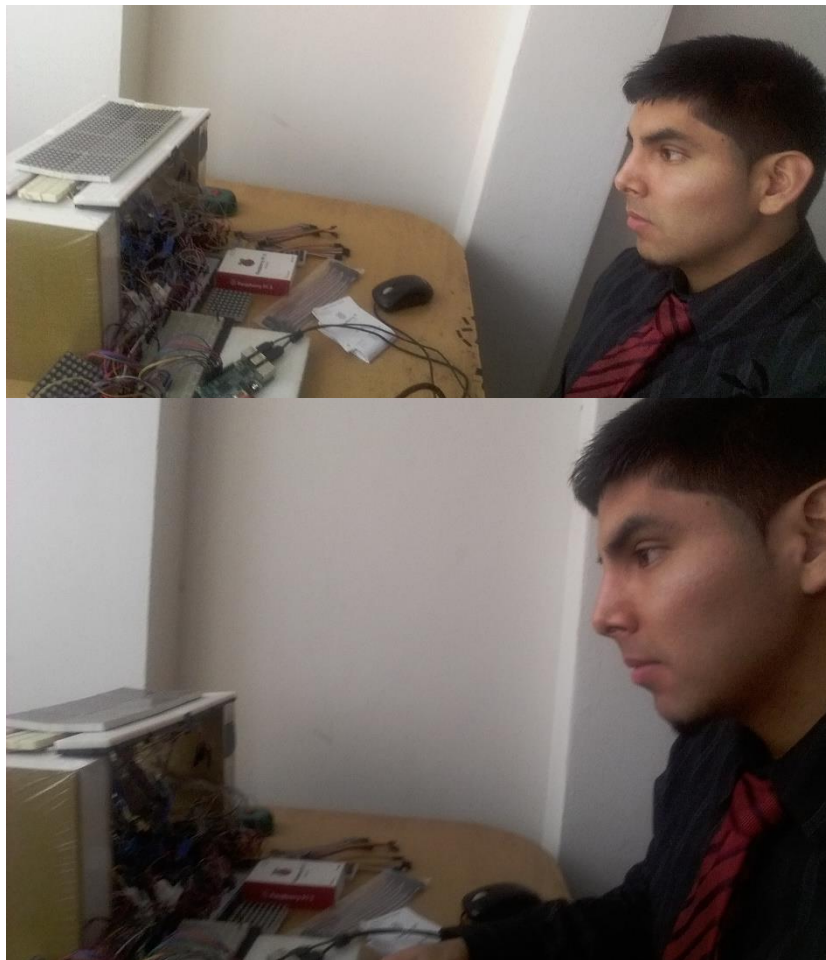


FIGURA 3.38: Mi persona trabajando en la implementación del prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

Luego de la corrección de errores y de asegurar el correcto conexionado del prototipo de panel y la placa Raspberry Pi en conjunto, se finalizó la implementación del prototipo de panel LED RGB y el resultado terminado se

muestra en la figura 3.39. En dicha figura se muestra el prototipo desplegando una imagen previamente cargada en el fichero donde se encuentra el código fuente que maneja este prototipo. En la parte superior se encuentra el panel montado adecuadamente en protoboard's y en la parte inferior se encuentra todo el circuito de control y los drivers IC's que manejan el prototipo.

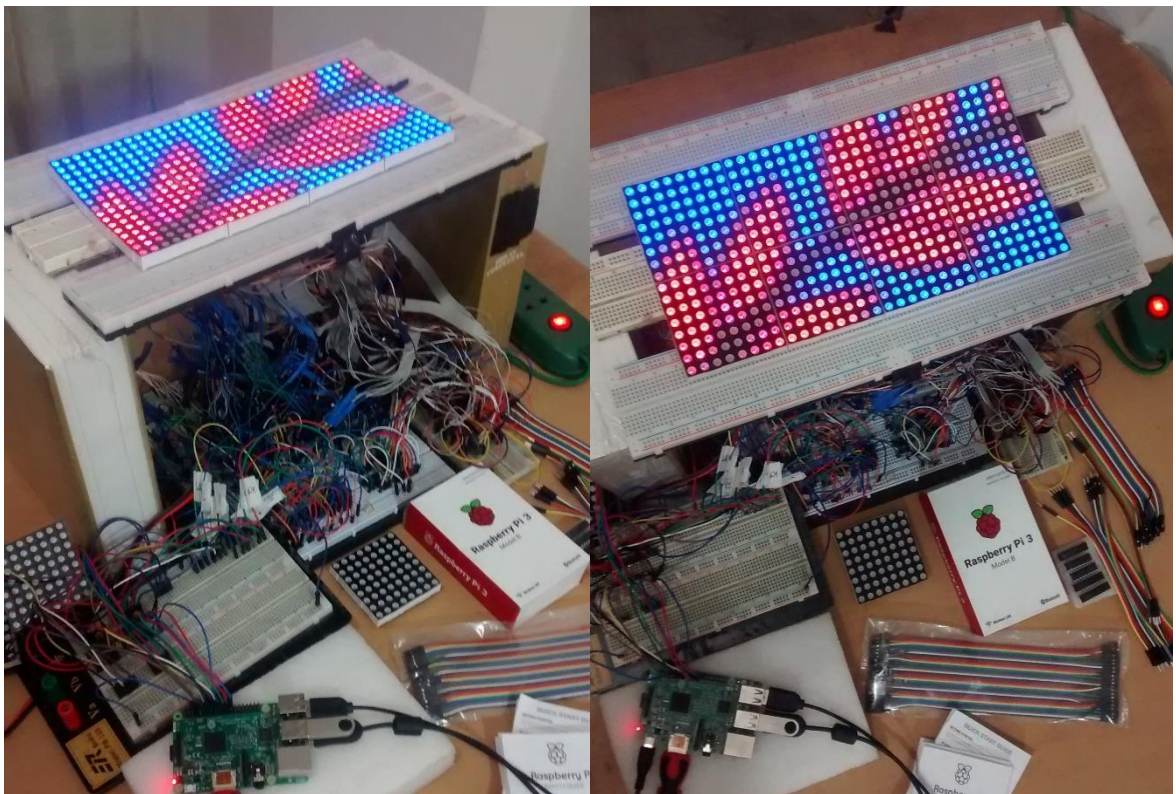


FIGURA 3.39: Resultados finales del prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS

3.3.1 Resultados de la implementación del prototipo de panel LED RGB

Luego de implementar el prototipo de panel LED RGB 16x32, se verificó que funciona tal como se esperaba y de acuerdo al diseño que se planteó. Se comprueba que este prototipo permite desplegar toda clase de información, formas, figuras, texto estático y en movimiento, etc. Así como

también permite la lectura de imágenes en diferentes formatos, aunque está limitado al tamaño de 16x32 pixeles ya que este es el tamaño del prototipo. En las figuras 3.40, 3.41 y 3.42, se muestran algunos de los resultados obtenidos.

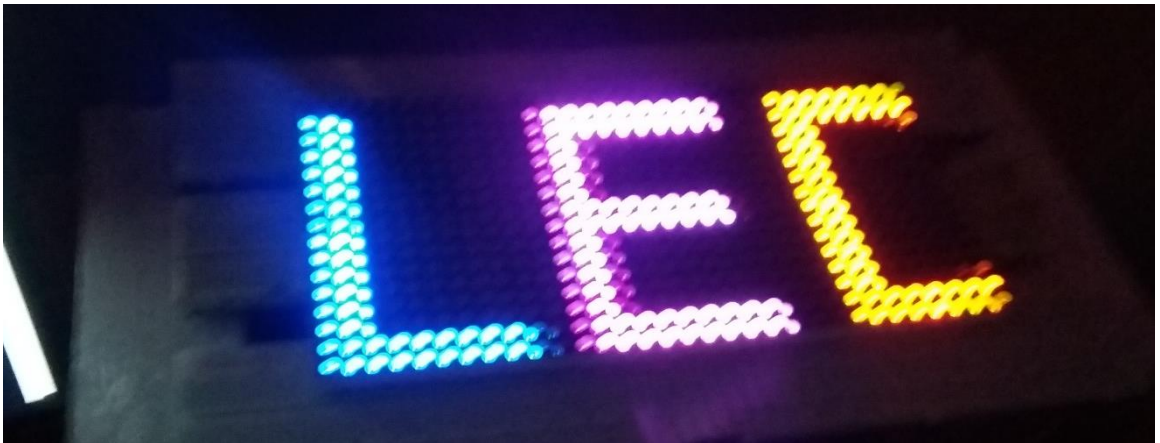


FIGURA 3.40: Mensajes en movimiento en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

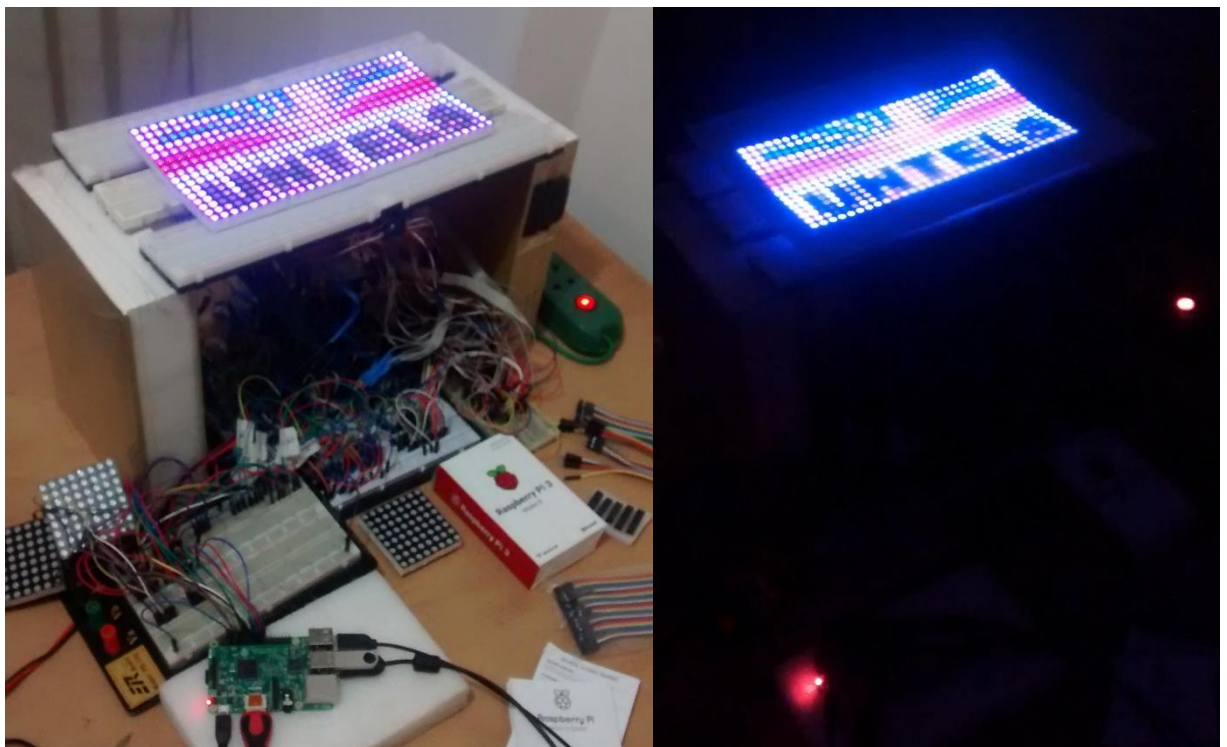


FIGURA 3.41: Parte del logotipo de la universidad UNTELS mostrado en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia



FIGURA 3.42: Logotipo de la placa Raspberry Pi mostrado en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

Como se aprecia, se puede desplegar cualquier tipo de información o data en el prototipo de panel LED RGB. Si se desea mostrar una imagen y esta es muy grande, se puede mostrar parte de ella en el panel o se puede escalar al tamaño del panel, pero perdiendo resolución. La idea es que en base a este prototipo se implemente un sistema de información en la UNTELS con un panel de mayores dimensiones, que estaría formado por varios de estos prototipos de panel LED RGB unidos en paralelo y en conexión cascada.

Una de las formas para transmitir información es a través de mensajes en movimiento, esto quiere decir que se puede crear un mensaje muy grande y hacer que se desplace en el panel mientras se va visualizando. Se puede crear el mensaje con letras, números, animaciones y toda clase de efectos para un mayor impacto visual, tal como se muestra en la figura 3.43 y en la figura 3.44.

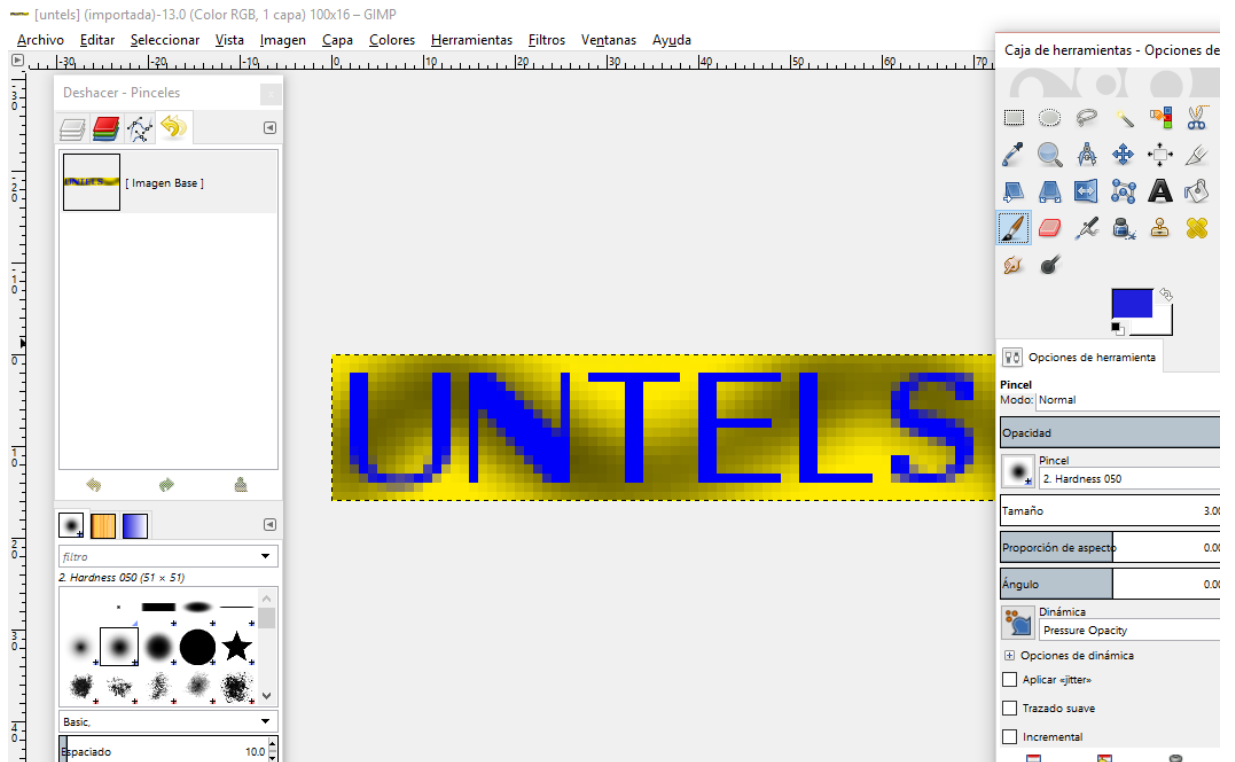


FIGURA 3.43: Texto de 16x150 pixeles creado usando el programa GIMP v2.8, para mostrarlo como mensaje en movimiento en el panel LED RGB 16x32

FUENTE: Elaboración propia

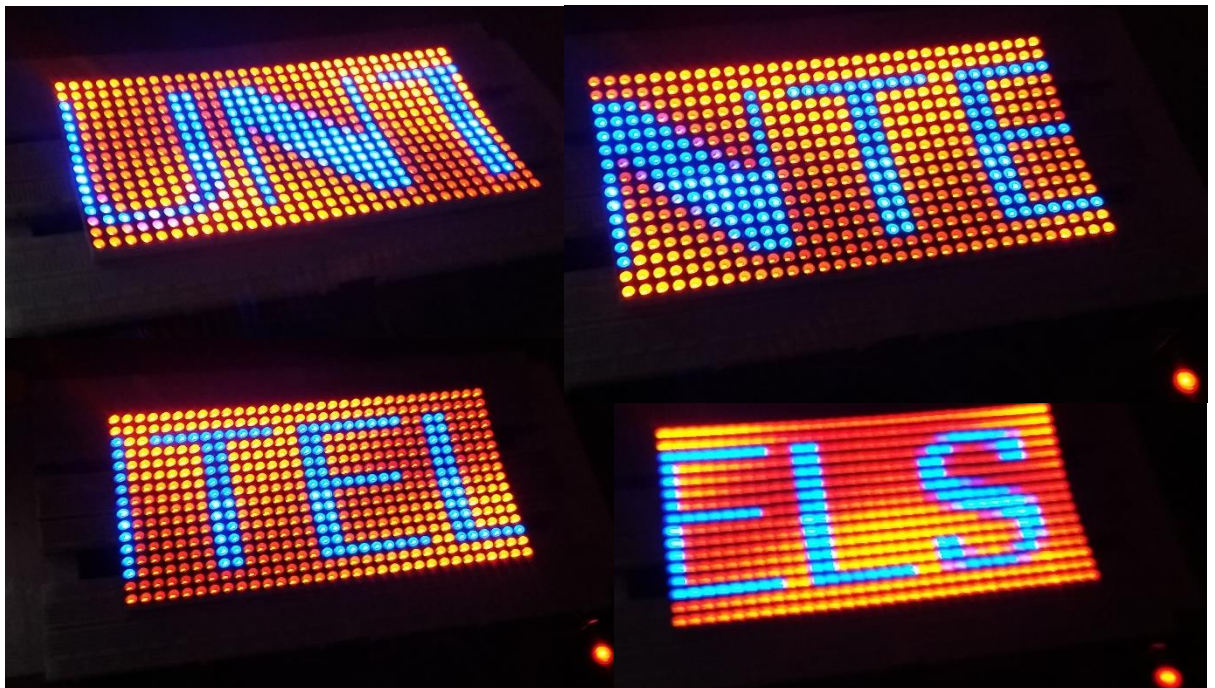


FIGURA 3.44: El mismo mensaje anterior siendo desplazado en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

Para ello se utilizó el programa de código abierto GIMP v2.8, en el cual se creó un texto de prueba de un tamaño de 16 pixeles de alto por 150 pixeles de ancho. Se puede crear cualquier texto con un ancho mayor, solo se debe mantener los 16 pixeles de alto. Si se quiere aumentar el tamaño se tiene que unir más paneles en paralelo y en conexión cascada.

Una de las características del programa empleado en el panel led RGB es que permite la lectura de imágenes con extensión ppm, como se aprecia en la figura 3.45.

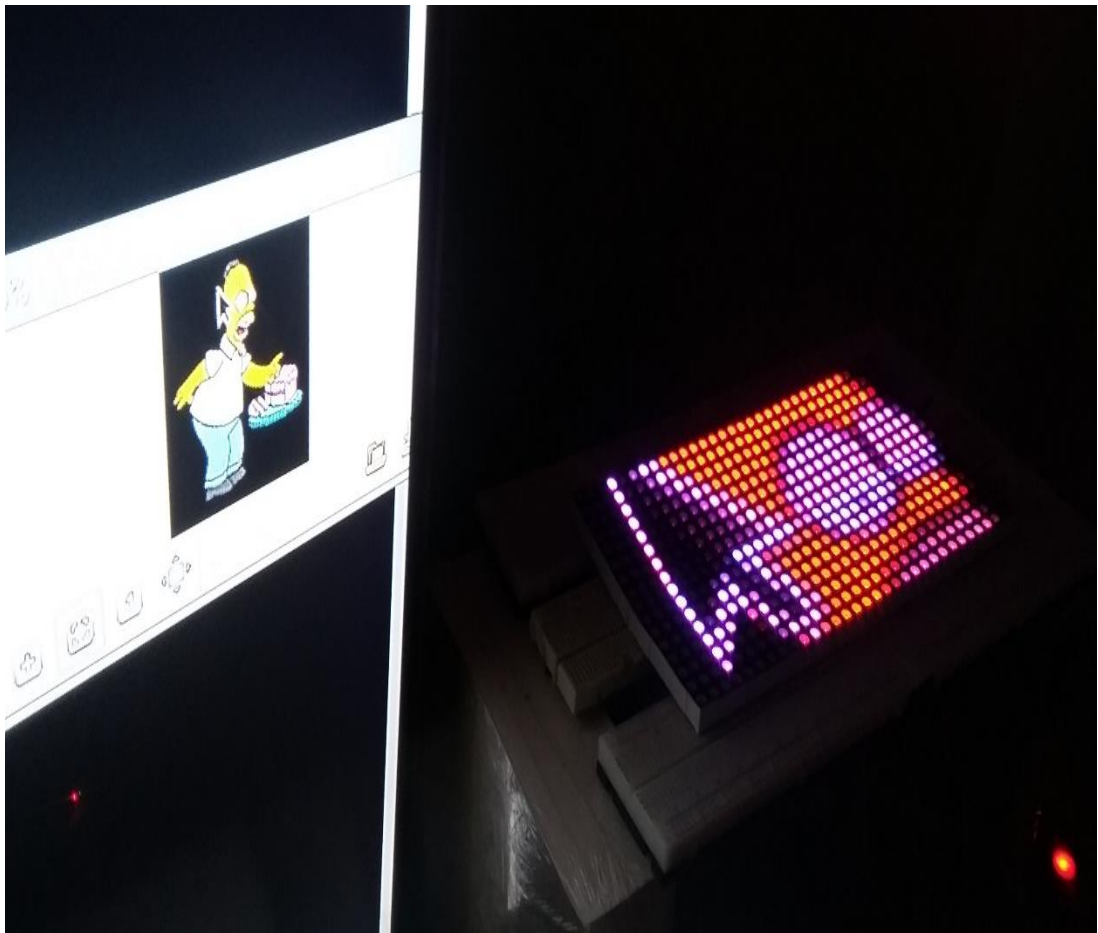


FIGURA 3.45: Visualización de parte de la pantalla del monitor en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia



FIGURA 3.46: Visualización de parte de la pantalla del monitor de diferentes imágenes en el prototipo de panel LED RGB 16x32.

FUENTE: Elaboración propia

Se puede desplegar imágenes y video de baja resolución también, como se observa en la figura 3.46, pero por ser este un prototipo de panel LED RGB de solo 16x32 pixeles, resulta muy escaso para visualizar una imagen completa o un video, inclusive con una calidad baja, pero es posible si se aumenta el tamaño del panel conectando en cascada varios de estos paneles de 16x32, ya que están diseñados para ser unidos en esta configuración. La placa Raspberry Pi 3 soporta la conexión de muchos paneles en cascada y en paralelo por lo cual solo se necesita adaptar el código y conectar dichos

paneles, claro está cada panel deberá ser idéntico para que sean compatibles. En la figura 3.47 se muestra la estructura de una pantalla LED.

DISPLAY STRUCTURE

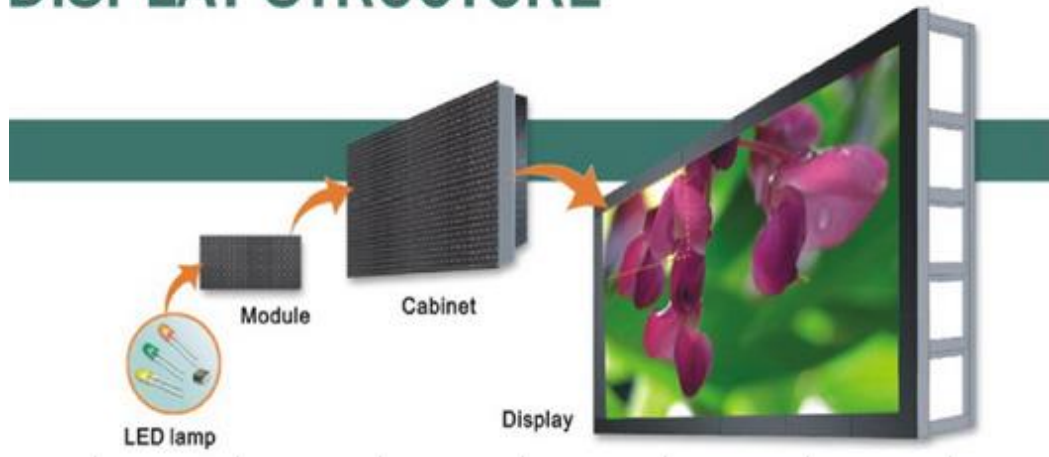


FIGURA 3.47: Esquema que muestra la estructura de una pantalla LED

FUENTE: Página Web www.leddisplayscreen.biz. Consultada en junio 2016

3.3.2 Aplicación a un sistema de información en la UNTELS

Debido a los buenos resultados obtenidos con la implementación del prototipo de panel LED RGB de 16x32 y gracias a la potencia de la placa Raspberry Pi 3, es totalmente factible la conexión de múltiples de estos paneles para formar una pantalla LED RGB más grande y aplicarlo a un sistema de información en la UNTELS. Ya que esta placa soporta no solamente la conexión en cascada de varios paneles, sino también la conexión en paralelo de estos, formando así, un display que puede transmitir todo tipo de información, hasta video en baja resolución. El sistema quedaría implementado tal como se muestra en la figura 3.48:

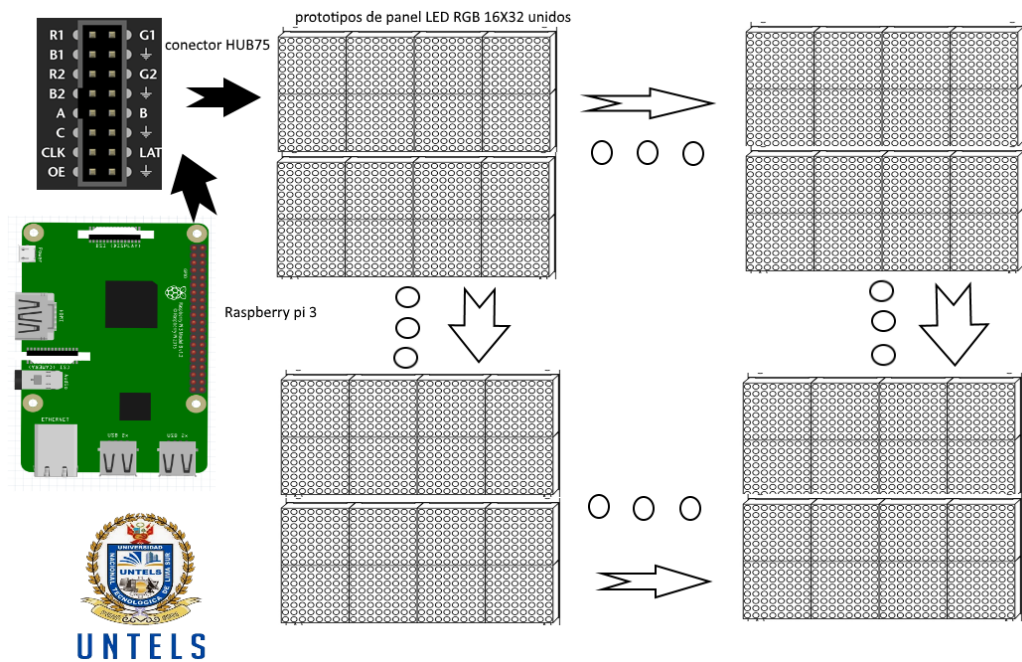


FIGURA 3.48: Esquema de un sistema de información para la UNTELS formado con módulos de paneles LED RGB de 16x32

FUENTE: Elaboración propia

El sistema soporta fácilmente 3 conexiones en paralelo con varios paneles LED RGB de 32x32 (2 paneles de 16x32) cada uno. La placa Raspberry Pi 3 puede conectar fácilmente 12 paneles por cada conexión en paralelo (36 paneles en total), pero esto se puede incrementar hasta un total de 96 paneles de 32x32 pixeles (32 en cada conexión en paralelo) y todavía llegar a tener alrededor de una frecuencia de actualización de 100 Hz con profundidad color de 24 bits (profundidad completa). Con un menor número de colores se puede controlar aún más rápido y con más paneles conectados en cascada. Se necesita una fuente de alimentación independiente para el panel. Aproximadamente 3,5 amperios por cada panel de 32x32 pixeles. Como se mencionó anteriormente, el conector en los paneles RGB se llama HUB75 y cada panel debe tener dos puertos, uno para la entrada y el otro para la salida de datos que irán a los siguientes paneles.

3.3.3 Análisis económico de un sistema de información en la UNTELS

El costo de implementación del sistema de información planteado puede variar dependiendo principalmente del tamaño final del panel LED RGB a montar. Este tamaño final deberá ser elegido de acuerdo a las necesidades de la universidad y los fines específicos que sus autoridades le quieran dar, por lo cual un análisis de costos definitivo no es posible, pero si se puede establecer un análisis para un panel propuesto de un tamaño estándar para exteriores en formato 4:2 de 1m x 0.5m, el cual tiene el tamaño suficiente para desplegar información básica, imágenes y video de baja resolución, salida de información de algún proceso, etc. Para este tamaño planteado se utilizarán 16 módulos del prototipo de panel LED RGB unidos todos en conexión serie y paralelo a la placa controladora Raspberry Pi 3. Los costos de los materiales por cada módulo del prototipo de panel LED RGB se muestran en la tabla 3.4:

TABLA 3.4: Análisis económico por panel LED RGB de 16x32 de un sistema de información para la UNTELS

Materiales por módulo de panel LED RGB	Cantidad	Costo \$/.
Resistencia 220 Ω 1/4 w	200 unidades	4.00
Resistencia 100 Ω 1/4 w	100 unidades	2.00
Condesador cerámico 100nF	50 unidades	4.00
Condesador electrolítico 1000uF	4 unidades	2.00
Matrix LED RGB 8x8 ánodo común	8 unidades	160.00
IC 74HC595	24 unidades	24.00
IC 74HC04	2 unidades	2.00
IC 74HC245	4 unidades	8.00
IC 74HC138	1 unidad	1.00
IC ULN2803	2 unidades	4.00
Transistor BD135	16 unidades	8.00
Placa PCB elaborada	1 unidad	25.00
Cables hembra-hembra	40 unidades	10.00
Costo de elaboración por módulo	1 unidad	25.00
Otros	1 unidad	20.00
	Costo Total	299.00

FUENTE: Elaboración propia

El costo total por 16 módulos que componen el tamaño de panel LED RGB planteado más la placa controladora Raspberry pi 3, su respectiva estructura de soporte, así como los demás elementos necesarios para su funcionamiento se muestra en la tabla 3.5

TABLA 3.5: Análisis económico de un sistema de información de 1m x 0.5m formado con 16 módulos de paneles LED RGB de 16x32

Materiales y componentes generales	Cantidad	Costo S/.
Módulo prototipo de panel LED RGB 16x32	16 unidades	4,784.00
Raspberry pi 3 modelo B	1 unidad	250.00
Fuente para raspberry pi 3 - 2.5A 5V	1 unidad	20.00
Memoria SD para raspberry pi 3 8Gb	1 unidad	30.00
Costo de ensamblaje de los módulos	1 unidad	100.00
Estructura metálica de soporte	1 unidad	200.00
Fuente de 20A 5V conmutada	1 unidad	250.00
Cables de datos y conexionado Hub 75	16 unidades	48.00
Cables de alimentación	16 unidades	32.00
Otros	1 unidad	200.00
	Costo Total	5,914.00

FUENTE: Elaboración propia

Como se observa después del análisis económico, el monto final para un panel propuesto de un tamaño estándar para exteriores en formato 4:2 de 1m x 0.5m es de S/. 5,914.00 soles. Esto es un análisis económico con respecto a materiales, componentes y mano de obra.

Para un tamaño estándar de 2m x 1m en formato 4:2 para exteriores se necesitarían 64 módulos del prototipo de panel LED RGB el análisis económico se muestra en la tabla 3.6:

TABLA 3.6: Análisis económico de un sistema de información de 2m x 1m formado con 64 módulos de paneles LED RGB de 16x32

Materiales y componentes generales	Cantidad	Costo S/.
Módulo prototipo de panel LED RGB 16x32	64 unidades	19,136.00
Raspberry pi 3 modelo B	1 unidad	250.00
Fuente para raspberry pi 3 - 2.5A 5V	1 unidad	20.00
Memoria SD para raspberry pi 3 8Gb	1 unidad	30.00
Costo de ensamblaje de los módulos	4 unidades	400.00
Estructura metálica de soporte	4 unidades	800.00
Fuente de 20A 5V conmutada	4 unidades	1,000.00
Cables de datos y conexionado Hub 75	64 unidades	192.00
Cables de alimentación	64 unidades	128.00
Otros	1 unidad	800.00
	Costo Total	22,756.00

FUENTE: Elaboración propia

Como se observa después del análisis económico, el monto final para un panel propuesto de un tamaño estándar para exteriores en formato 4:2 de 2m x 1m es de S/. 22,756.00 soles. Esto es un análisis económico con respecto a materiales, componentes y mano de obra.

Por último, se propone un tamaño mayor de panel LED de 4m x 2m en formato 4:2 para exteriores, para ello se necesitarían 256 módulos del prototipo de panel LED RGB. El análisis económico se muestra en la tabla 3.7:

TABLA 3.7: Análisis económico de un sistema de información de 4m x 2m formado con 256 módulos de paneles LED RGB de 16x32

Materiales y componentes generales	Cantidad	Costo S/.
Módulo prototipo de panel LED RGB 16x32	256 unidades	76,544.00
Raspberry pi 3 modelo B	1 unidad	250.00
Fuente para raspberry pi 3 - 2.5A 5V	1 unidad	20.00
Memoria SD para raspberry pi 3 8Gb	1 unidad	30.00
Costo de ensamblaje de los módulos	16 unidades	1,600.00
Estructura metálica de soporte	16 unidades	3,200.00
Fuente de 20A 5V conmutada	16 unidades	4,000.00
Cables de datos y conexionado Hub 75	256 unidades	768.00
Cables de alimentación	256 unidades	512.00
Otros	1 unidad	3,200.00
	Costo Total	90,124.00

FUENTE: Elaboración propia

Como se observa después del análisis económico, el monto final para un panel propuesto de un tamaño estándar para exteriores en formato 4:2 de 4m x 2m es de S/. 90,124.00 soles. Esto es un análisis económico con respecto a materiales, componentes y mano de obra.

En la figura 3.49 se muestra la relación en dimensiones de los diferentes tamaños de sistemas de información de panel LED RGB propuestos y la cantidad de módulos LED que los componen. Como se observa, el primer tamaño propuesto se compone de 16 módulos de panel LED RGB, que unidos hacen un display de 1m x 0.5m. El siguiente tamaño propuesto es 4 veces más grande que el anterior, ya que se compone de 64 paneles LED RGB y tiene unas dimensiones de 2m x 1m en formato 4:2, el precio de este display también es aproximadamente 4 veces mayor con respecto al anterior tamaño propuesto. El tercer tamaño propuesto, a su vez, también es 4 veces más grande que el panel anterior ya que se compone de 256 módulos de

panel LED RGB y tiene unas dimensiones de 4m x 2m también en formato 4:2.

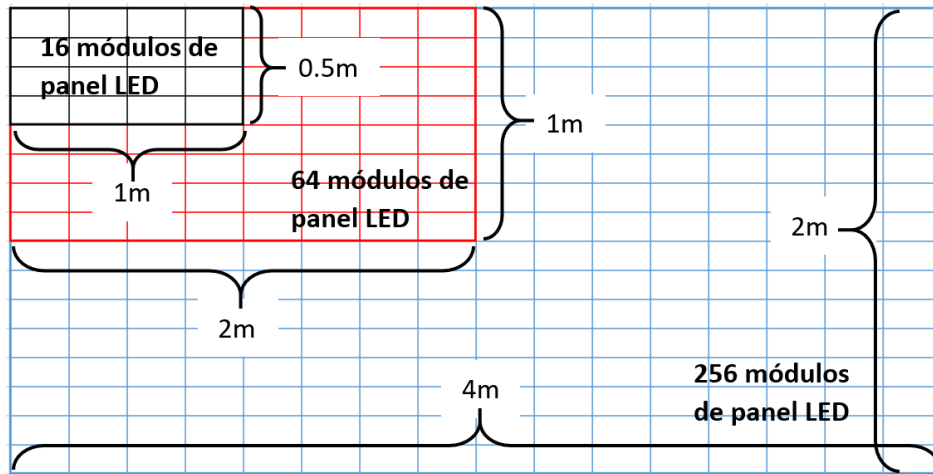


FIGURA 3.49: Diferentes tamaños propuestos de sistemas de información formados con 16, 64 y 256 módulos de paneles LED RGB de 16x32

FUENTE: Elaboración propia

CONCLUSIONES

- Se concluyó el diseño del circuito electrónico con los IC's que hicieron de función de drivers de las matrices LED RGB utilizando el programa PROTEUS 8.4.
- Se logró armar e implementar la parte hardware del circuito, de acuerdo al diseño realizado, con los IC's drivers de LED's realizando el conexionado con cada una de las matrices 8x8 y realizando el montaje de todos los componentes necesarios para el funcionamiento del panel LED RGB.
- Se logró programar, compilar y depurar el código fuente que manejó el funcionamiento y distribución de la data hacia el panel LED RGB con el lenguaje C++ y PYTHON y bajo el entorno del sistema operativo RASPBIAN.
- La configuración de la placa RASPBERRY PI 3 con el sistema operativo RASPBIAN y la conexión al circuito controlador y hacia las matrices LED's fue hecha de manera exitosa ya que, al probar el funcionamiento del sistema en conjunto, el sistema funciono de acuerdo a lo previsto en el diseño.

RECOMENDACIONES

- Se recomienda utilizar el primer diseño si se desea agregar una línea de dirección más a las 3 líneas ya existentes (A, B, C). para formar un panel de 32x32 pixeles. Si se realiza esto se deberá cambiar el decodificador de 3 - 8 por un decodificador de 4 - 16. Con este cambio también cambia el scan rate (Velocidad de barrido) del panel de 1/8 a 1/16. Pero si lo que se desea es implementar un panel más grande bastaría con unir en cascada varios paneles de 16x32, sin tener que pasar a módulos más grandes como 32x32, 32x64, etc. La placa Raspberry Pi 3 soporta la conexión de varios paneles LED RGB de 16x32 tanto en cascada y en paralelo.
- Se recomienda realizar un PCB que contenga todos los drivers de las matrices LED por un lado y por el otro lado todas las matrices LED RGB 8x8 correctamente posicionadas y con las pistas correctamente enrutadas hacia estos. De esta manera se evitará el enorme trabajo de soldar cable por cable si se utiliza otro soporte para las matrices o la gran cantidad de cables y el tiempo que se empleará si se utiliza protoboards para la conexión con las matrices, en todo caso se recomienda utilizar cable ribbon para la conexión con las matrices si se utiliza protoboards.
- Se recomienda utilizar las librerías empleadas en el código fuente para programar y compilar el código. Se puede cambiar el código y adaptarlo a las necesidades propias de la aplicación que se desee darle al panel LED RGB.

- Existen varios sistemas operativos basados en Linux o distribuciones de Linux que se pueden instalar en la placa Raspberry Pi 3, incluso se puede instalar una versión especial de Windows llamada Windows 10 IOT que fue lanzada recientemente, sin embargo, se recomienda instalar el sistema operativo RASPBIAN o RASPBIAN LITE en la placa Raspberry Pi 3 para poder configurar la placa según el diseño planteado y realizar la conexión al circuito controlador y hacia las matrices LED's.

REFERENCIAS

Libros:

- 1) Angulo, J. (1996). *Electrónica digital moderna: teoría y práctica* (16va Ed.). España: Paraninfo.
- 2) Hambley, A. (2001). *Electrónica* (2da Ed.). México: Pearson.
- 3) Malvino, A. Y Bates, d. (2007). *Principios de electrónica*. (7ma Ed.). México: McGraw Hill.
- 4) Mandado, E. (2007). *Sistemas electrónicos digitales* (9na Ed.). España: Marcombo.
- 5) Schmidt, M. Y Carter, J. (2012) *Raspberry Pi: A quick start guide*. (1ra Ed.). EEUU: The Pragmatic Programmers, LLC.
- 6) Tocci, R. (2003). *Sistemas digitales: Principios y Aplicaciones*. (8va Ed.). México: Pearson.
- 7) Tokheim, R. (2016). *Electrónica digital: principios y aplicaciones* (7ma. Ed.). España: McGraw-Hill.

Tesis consultadas:

- 1) Agonoy, A. y Cuenco A. (2009). *Programmable 24-bit RGB LED Color Panel via Bluetooth Technology*. Tesis. Mapua Institute of technology. Filipinas.
- 2) Baldarrago, J.A. (2010). *Diseño de efectos y variación de colores mediante degradé en leds de potencia RGB aplicados a paneles publicitarios*. Tesis. Pontificia Universidad Católica. Perú
- 3) De Jong, S. (2010). *Investigative development of an accesible LED matrix display*. Tesis. Massey University. Nueva Zelanda.

- 4) Mohd, S. (2014). *Full Color of LED Display System using ARM microcontroller*. Tesis. Universidad Tecnológica de Malasia. Malasia.

Documentos electrónicos:

- 1) Artschip electronics. (marzo de 2016). *Hoja técnica del transistor APM4953*. Recuperado el 15 de junio de 2016, de:
<http://radioremont.com/datasheet/APM4953.pdf>
- 2) Betlux. (enero de 2016). *Hoja técnica del módulo de matriz LED RGB 8X8 D23088A/B*. Recuperado el 15 de junio de 2016, de:
http://www.betlux.com/product/led_dot_matrix/BL-M23A881XXX.PDF
- 3) Fairchild Semiconductor. (agosto de 2013). *Hoja técnica del transistor BD135*. Recuperado el 15 junio de 2016, de:
<https://www.fairchildsemi.com/datasheets/BD/BD135.pdf>
- 4) Macroblock Inc. (marzo de 2004). *Hoja técnica del IC MBI5026*. Recuperado el 15 de junio de 2016, de:
<http://www.vabolis.it/stuff/MBI5026.pdf>
- 5) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC245*. Recuperado el 15 de junio de 2016, de:
http://www.nxp.com/documents/data_sheet/74HC_HCT245.pdf
- 6) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC138*. Recuperado el 15 de junio de 2016, de:
http://www.nxp.com/documents/data_sheet/74HC_HCT138.pdf
- 7) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC595*. Recuperado el 15 de junio de 2016, de:
https://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf

8) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC04*. Recuperado el 15 de junio de 2016, de:

https://www.nxp.com/documents/data_sheet/74HC_HCT04.pdf

9) Texas Instruments. (enero de 2015). *Hoja técnica del IC ULN2803*. Recuperado el 15 de junio de 2016, de:

<http://www.ti.com/lit/ds/symlink/uln2803a.pdf>

BIBLIOGRAFIA

Libros:

- 1) Angulo, J. (1996). *Electrónica digital moderna: teoría y práctica* (16va Ed.). España: Paraninfo.
- 2) Boylestad, R Y Nashelsky, L. (2009). *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos* (10ma Ed.). México: Pearson.
- 3) Hambley, A. (2001). *Electrónica* (2da Ed.). México: Pearson.
- 4) Lutz, M. (2011). *Programming Phyton*. (4ta Ed.). EEUU: O'Reilly
- 5) Malvino, A. Y Bates, d. (2007). *Principios de electrónica*. (7ma Ed.). México: McGraw Hill.
- 6) Mandado, E. (2007). *Sistemas electrónicos digitales* (9na Ed.). España: Marcombo.
- 7) Membrey, P. Y Hows, D. (2013). *Learn Raspberry Pi with Linux*. (1ra Ed.). EEUU: Technology in action.
- 8) Schmidt, M. Y Carter, J. (2012) *Raspberry Pi: A quick start guide*. (1ra Ed.). EEUU: The Pragmatic Programmers, LLC.
- 9) Tocci, R. (2003). *Sistemas digitales: Principios y Aplicaciones*. (8va Ed.). México: Pearson.
- 10) Tokheim, R. (2016). *Electrónica digital: principios y aplicaciones* (7ma. Ed.). España: McGraw-Hill.
- 11) Upton, E. Y Halfacree, G. (2012). *Raspberry Pi User Guide*. (1ra Ed.). Reino Unido: Wiley.

Tesis consultadas:

- 1) Agonoy, A. y Cuenco A. (2009). *Programmable 24-bit RGB LED Color Panel via Bluetooth Technology*. Tesis. Mapua Institute of technology. Filipinas.
- 2) Baldarrago, J.A. (2010). *Diseño de efectos y variación de colores mediante degradé en leds de potencia RGB aplicados a paneles publicitarios*. Tesis. Pontificia Universidad Católica. Perú
- 3) De Jong, S. (2010). *Investigative development of an accesible LED matrix display*. Tesis. Massey University. Nueva Zelanda.
- 4) Mohd, S. (2014). *Full Color of LED Display System using ARM microcontroller*. Tesis. Universidad Tecnológica de Malasia. Malasia.

Documentos electrónicos:

- 1) Artschip electronics. (marzo de 2016). *Hoja técnica del transistor APM4953*. Recuperado el 15 de junio de 2016, de:
<http://radioremont.com/datasheet/APM4953.pdf>
- 2) Betlux. (enero de 2016). *Hoja técnica del módulo de matriz LED RGB 8X8 D23088A/B*. Recuperado el 15 de junio de 2016, de:
http://www.betlux.com/product/led_dot_matrix/BL-M23A881XXX.PDF
- 3) Fairchild Semiconductor. (1984). *An Introduction to and comparison of 74HCT TTL compatible CMOS Logic*. Recuperado el 15 de junio de 2016, de: <https://www.fairchildsemi.com/application-notes/AN/AN-368.pdf>
- 4) Fairchild Semiconductor. (agosto de 2013). *Hoja técnica del transistor BD135*. Recuperado el 15 de junio de 2016, de:
<https://www.fairchildsemi.com/datasheets/BD/BD135.pdf>

- 5) Geekytheory (2016). *Tutoriales Raspberry Pi*. Recuperado el 15 de junio de 2016, de: <https://geekytheory.com/category/geeky-theory2/tutoriales2/raspberry-pi/>
- 6) Macroblock Inc. (marzo de 2004). *Hoja técnica del IC MBI5026*. Recuperado el 15 de junio de 2016, de: <http://www.vabolis.it/stuff/MBI5026.pdf>
- 7) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC245*. Recuperado el 15 de junio de 2016, de: http://www.nxp.com/documents/data_sheet/74HC_HCT245.pdf
- 8) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC138*. Recuperado el 15 de junio de 2016, de: http://www.nxp.com/documents/data_sheet/74HC_HCT138.pdf
- 9) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC595*. Recuperado el 15 de junio de 2016, de: https://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf
- 10) NXP Semiconductors. (noviembre de 2015). *Hoja técnica del IC 74HC04*. Recuperado el 15 de junio de 2016, de: https://www.nxp.com/documents/data_sheet/74HC_HCT04.pdf
- 11) Texas Instruments. (enero de 2015). *Hoja técnica del IC ULN2803*. Recuperado el 15 de junio de 2016, de: <http://www.ti.com/lit/ds/symlink/uln2803a.pdf>
- 12) Texas Instruments. (1993) *Implications of Slow or Floating CMOS Inputs*. Recuperado el 15 de junio de 2016, de: <http://www.ti.com/lit/an/scba004c/scba004c.pdf>

ANEXOS

ANEXO 1: DATASHEET MATRIZ LED 8X8 D23088A/B

LED DOT MATRIX

BL-M23A881XXX

Features:

- Ø 60.20mm (2.3") 1/5.0 dot matrix LED display, RGB COLOR
- Ø Low current operation.
- Ø Excellent character appearance.
- Ø Easy mounting on P.C. Boards or sockets.
- Ø I.C. Compatible.
- Ø ROHS Compliance.



Electrical-optical characteristics: (Ta=25°C) (Test Condition: IF=20mA)

Part No		Chip			VF Unit:V		Iv
Row Cathode Column Anode	Row Anode Column Cathode	Emitted Color	Material	λ _P (nm)	Typ	Max	TYP.(mcd)
BL-M23A881RGB- XX	BL-M23B881RGB- XX	Super Red	GaAlAs/GaAs,DH	660	1.85	2.20	280
		Green	GaP/GaP	570	2.20	2.50	250
		Ultra Blue	InGaN	470	2.70	4.20	150
BL-M23A881DUGU B-XX	BL-M23B881DUGU B-XX	Ultra Red	GaAlAs/GaAs,DDH	660	1.85	2.20	310
		Ultra Green	AlGaInP	574	2.20	2.50	380
		Ultra Blue	InGaN	470	2.70	4.20	270

--XX: Surface / Lens color£°

Number	0	1	2	3	4	5
Ref Surface Color	White	Black	Gray	Red	Green	
Epoxy Color	Water clear	White diffused	Red Diffused	Green Diffused	Yellow Diffused	

Absolute maximum ratings (Ta=25°C)

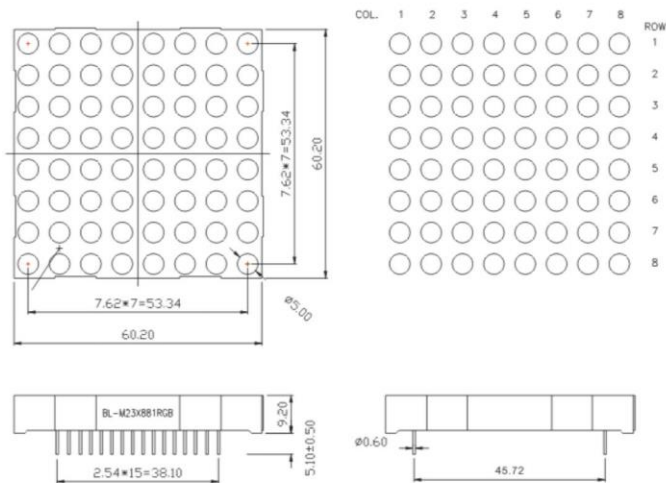
Parameter	S	G	B	D	UG	UB	Unit
Forward Current I _F	25	30	30	25	30	30	mA
Power Dissipation P _d	60	65	120	60	75	120	mW
Reverse Voltage V _R	5	5	5	5	5	5	V
Peak Forward Current I _{PF} (Duty 1/10 @1KHZ)	150	150	100	150	150	100	mA
Operation Temperature T _{OPR}	-40 to +80						°C
Storage Temperature T _{STG}	-40 to +85						°C
Lead Soldering Temperature T _{SOL}	Max.260±5°C for 3 sec Max. (1.6mm from the base of the epoxy bulb)						°C

LED DOT MATRIX

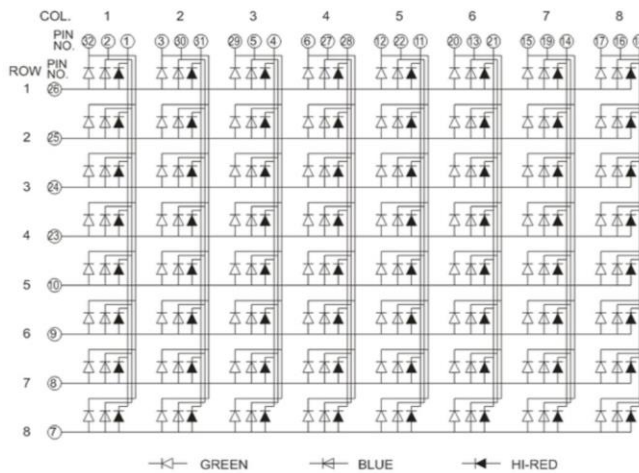
BL-M23A881XXX

Package configuration & Internal circuit diagram

BL-M23X881 Series



BL-M23B881RGB (BL-M23A881RGB C.C.)



Notes:

1. All dimensions are in millimeters (inches)
2. Tolerance is $\pm 0.25(0.01)$ unless otherwise noted.
3. Specifications are subject to change without notice.

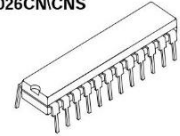
ANEXO 2: DATASHEET IC MBI5026



Features

- | 16 constant-current output channels
- | Constant output current invariant to load voltage change
- | Excellent output current accuracy:
 - between channels: $<\pm 3\%$ (max.), and
 - between ICs: $<\pm 6\%$ (max.)
- | Output current adjusted through an external resistor
- | Constant output current range: 5-90 mA
- | Fast response of output current, \overline{OE} (min.): 200 ns
- | 25MHz clock frequency
- | Schmitt trigger input
- | 5V supply voltage

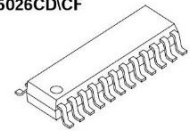
MBI5026CN\CNS



CN: P-DIP24-300-2.54

CNS: SP-DIP24-300-1.78

MBI5026CD\CF



CD: SOP24-300-1.27

CF: SOP24-300-1.00

MBI5026CP\CPA



CP\CPA: SSOP24-150-0.64

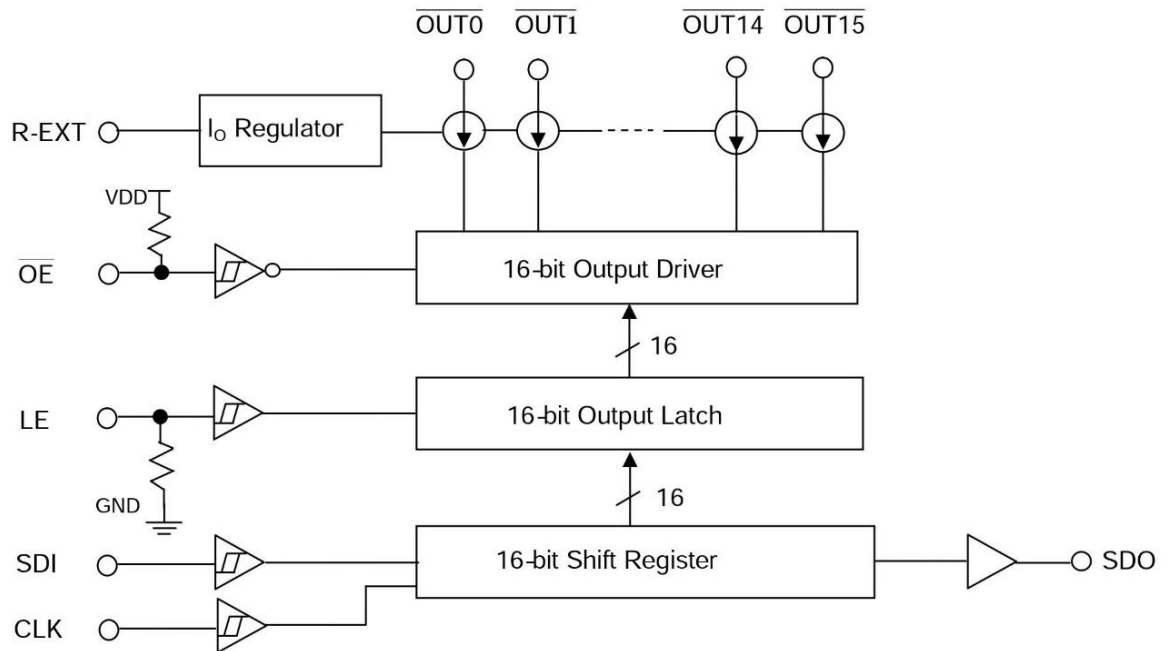
Current Accuracy		Conditions
Between Channels	Between ICs	
$< \pm 3\%$	$< \pm 6\%$	$I_{OUT} = 10 \sim 60 \text{ mA}$

Product Description

MBI5026 is designed for LED displays. As an enhancement of its predecessor, MBI5016, MBI5026 exploits PrecisionDrive™ technology to enhance its output characteristics. MBI5026 contains a serial buffer and data latches which convert serial input data into parallel output format. At MBI5026 output stage, sixteen regulated current ports are designed to provide uniform and constant current sinks for driving LEDs within a large range of Vf variations.

MBI5026 provides users with great flexibility and device performance while using MBI5026 in their system design for LED display applications, e.g. LED panels. Users may adjust the output current from 5 mA to 90 mA through an external resistor, R_{ext} , which gives users flexibility in controlling the light intensity of LEDs. MBI5026 guarantees to endure maximum 17V at the output port. The high clock frequency, 25 MHz, also satisfies the system requirements of high volume data transmission.

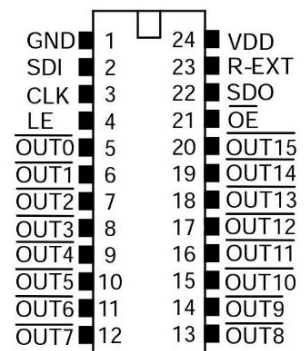
Block Diagram



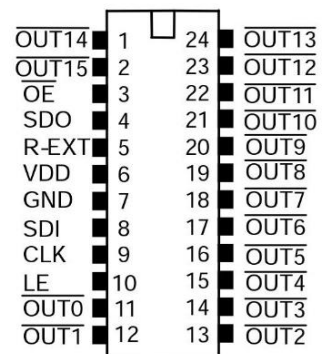
Terminal Description

Pin Name	Function
GND	Ground terminal for control logic and current sink
SDI	Serial-data input to the shift register
CLK	Clock input terminal for data shift on rising edge
LE	Data strobe input terminal Serial data is transferred to the output latch when LE is high. The data is latched when LE goes low.
$\overline{\text{OUT0}} - \overline{\text{OUT15}}$	Constant current output terminals
$\overline{\text{OE}}$	Output enable terminal When $\overline{\text{OE}}$ (active) low, the output drivers are enabled; when $\overline{\text{OE}}$ high, all output drivers are turned OFF (blanked).
SDO	Serial-data output to the following SDI of next driver IC
R-EXT	Input terminal used to connect an external resistor for setting up output current for all output channels
VDD	5V supply voltage terminal

Pin Configuration

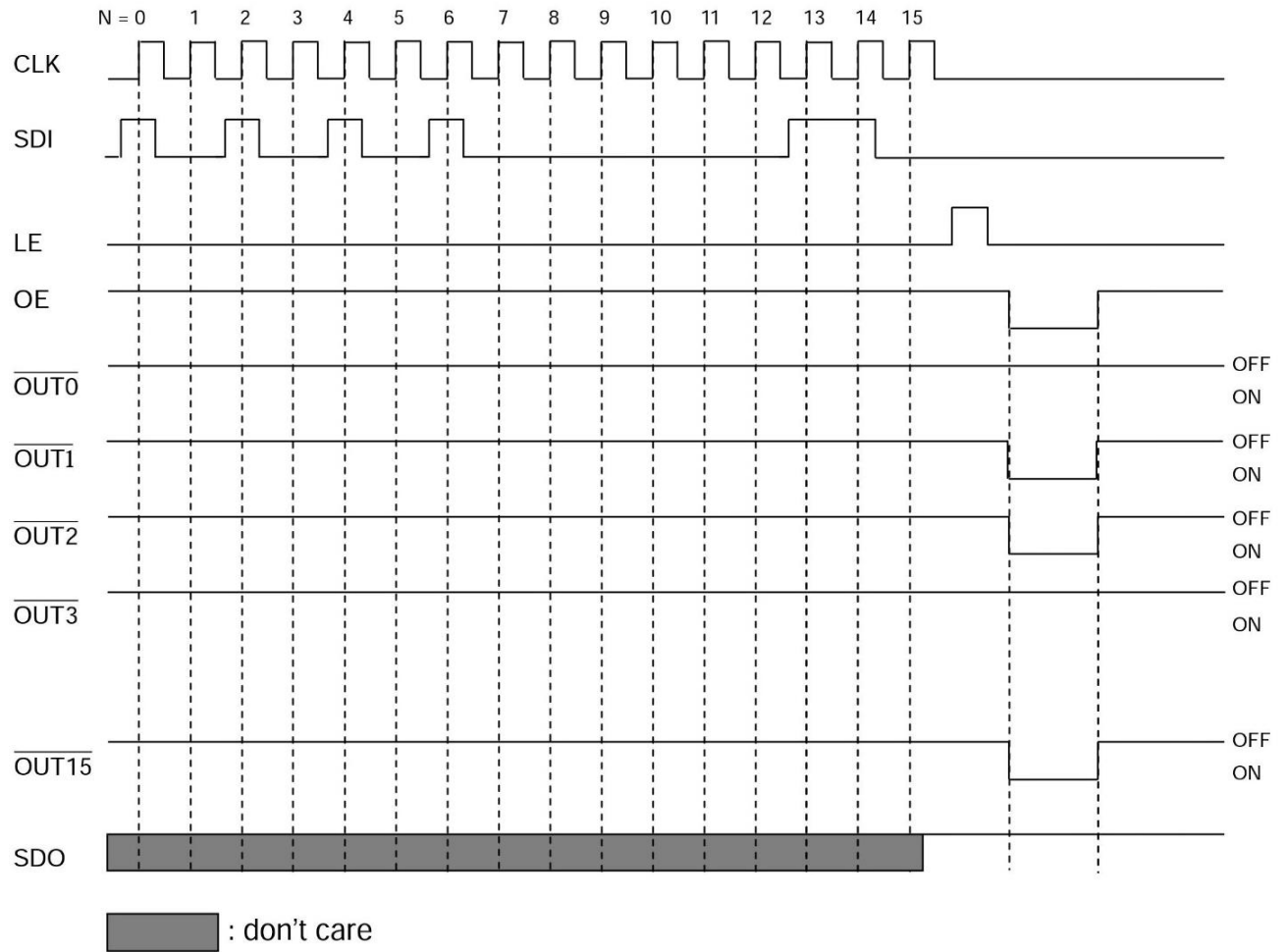


MBI5026CN\CS\CD\CF\CP



MBI5026CPA

Timing Diagram



Truth Table

CLK	LE	$\overline{\text{OE}}$	SDI	$\overline{\text{OUT0}} \dots \overline{\text{OUT7}} \dots \overline{\text{OUT15}}$	SDO
\uparrow	H	L	D_n	$\overline{D_{n-15}} \dots \overline{D_{n-7}} \dots \overline{D_{n-1}}$	D_{n-15}
\uparrow	L	L	D_{n+1}	No Change	D_{n-14}
\uparrow	H	L	D_{n+2}	$\overline{D_{n-13}} \dots \overline{D_{n-5}} \dots \overline{D_{n-2}}$	D_{n-13}
\downarrow	X	L	D_{n+3}	$\overline{D_{n-13}} \dots \overline{D_{n-5}} \dots \overline{D_{n-2}}$	D_{n-13}
\downarrow	X	H	D_{n+3}	Off	D_{n-13}

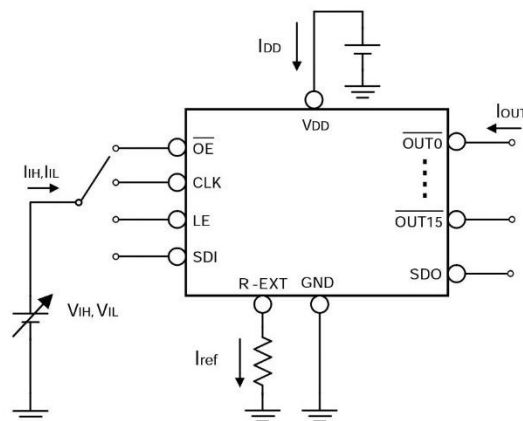
Maximum Ratings

Characteristic		Symbol	Rating	Unit
Supply Voltage		V_{DD}	0~7.0	V
Input Voltage		V_{IN}	-0.4~ $V_{DD} + 0.4$	V
Output Current		I_{OUT}	+90	mA
Output Voltage		V_{DS}	-0.5~+17.0	V
Clock Frequency		F_{CLK}	25	MHz
GND Terminal Current		I_{GND}	1440	mA
Power Dissipation (On PCB, $T_a=25^\circ\text{C}$)	CN – type	P_D	2.32	W
	CNS – type		1.87	
	CD – type		2.51	
	CF – type		2.12	
	CP – type		1.73	
	CPA – type		1.73	
Thermal Resistance (On PCB, $T_a=25^\circ\text{C}$)	CN – type	$R_{th(j-a)}$	53.82	$^\circ\text{C/W}$
	CNS – type		66.74	
	CD – type		49.81	
	CF – type		59.01	
	CP – type		72.43	
	CPA – type		72.43	
Operating Temperature		T_{opr}	-40~+85	$^\circ\text{C}$
Storage Temperature		T_{stg}	-55~+150	$^\circ\text{C}$

Electrical Characteristics

Characteristic		Symbol	Condition	Min.	Typ.	Max.	Unit
Supply Voltage		V_{DD}	-	4.5	5.0	5.5	V
Output Voltage		V_{DS}	$\overline{OUT0} \sim \overline{OUT15}$	-	-	17.0	V
Output Current		I_{OUT}	DC Test Circuit	5	-	90	mA
		I_{OH}	SDO	-	-	-1.0	mA
		I_{OL}	SDO	-	-	1.0	mA
Input Voltage	“H” level	V_{IH}	$T_a = -40 \sim 85^\circ\text{C}$	$0.8V_{DD}$	-	V_{DD}	V
	“L” level	V_{IL}	$T_a = -40 \sim 85^\circ\text{C}$	GND	-	$0.3V_{DD}$	V
Output Leakage Current		I_{OH}	$V_{OH}=17.0\text{V}$	-	-	0.5	μA
Output Voltage	SDO	V_{OL}	$I_{OL}=+1.0\text{mA}$	-	-	0.4	V
		V_{OH}	$I_{OH}=-1.0\text{mA}$	4.6	-	-	V
Output Current 1		I_{OUT1}	$V_{DS}=0.6\text{V}$ $R_{ext}=720\ \Omega$	-	26.25	-	mA
Current Skew		dI_{OUT1}	$I_{OL}=26.25\text{mA}$ $V_{DS}=0.6\text{V}$ $R_{ext}=720\ \Omega$	-	± 1	± 3	%
Output Current 2		I_{OUT2}	$V_{DS}=0.8\text{V}$ $R_{ext}=360\ \Omega$	-	52.5	-	mA
Current Skew		dI_{OUT2}	$I_{OL}=52.5\text{mA}$ $V_{DS}=0.8\text{V}$ $R_{ext}=360\ \Omega$	-	± 1	± 3	%
Output Current vs. Output Voltage Regulation		$\%/dV_{DS}$	V_{DS} within 1.0V and 3.0V	-	± 0.1	-	% / V
Output Current vs. Supply Voltage Regulation		$\%/dV_{DD}$	V_{DD} within 4.5V and 5.5V	-	± 1	-	% / V
Pull-up Resistor		$R_{IN(up)}$	\overline{OE}	250	500	800	K Ω
Pull-down Resistor		$R_{IN(down)}$	LE	250	500	800	K Ω
Supply Current	“OFF”	$I_{DD(off) 1}$	$R_{ext}=\text{Open}, \overline{OUT0} \sim \overline{OUT15} =\text{Off}$	-	7	12	mA
		$I_{DD(off) 2}$	$R_{ext}=720\ \Omega, \overline{OUT0} \sim \overline{OUT15} =\text{Off}$	-	10	12	
		$I_{DD(off) 3}$	$R_{ext}=360\ \Omega, \overline{OUT0} \sim \overline{OUT15} =\text{Off}$	-	12	15	
	“ON”	$I_{DD(on) 1}$	$R_{ext}=720\ \Omega, \overline{OUT0} \sim \overline{OUT15} =\text{On}$	-	10	18	
		$I_{DD(on) 2}$	$R_{ext}=360\ \Omega, \overline{OUT0} \sim \overline{OUT15} =\text{On}$	-	12	20	

Test Circuit for Electrical Characteristics



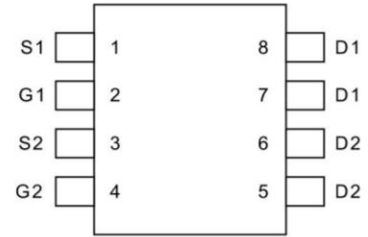
ANEXO 3: TRANSISTOR MOSFET APM4953

Dual P-Channel Enhancement Mode MOSFET

Features

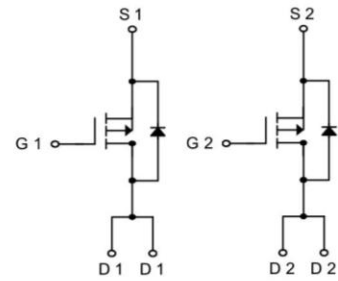
- -30V/-4.9A, $R_{DS(ON)} = 53\text{me (typ.) @ } V_{GS} = -10\text{V}$
 $R_{DS(ON)} = 80\text{me (typ.) @ } V_{GS} = -4.5\text{V}$
- Super High Density Cell Design
- Reliable and Rugged
- SO-8 Package

Pin Description



Applications

Power Management in Notebook Computer,
 Portable Equipment and Battery Powered
 Systems



P-Channel MOSFET

Ordering and Marking Information

<p>APM4953 □□-□□</p> <div style="margin-left: 20px;"> <p>□□ - Handling Code</p> <p>□ - Temp. Range</p> <p>□ - Package Code</p> </div>	<p>Package Code K : SO-8 Operation Junction Temp. Range C : -55 to 150°C Handling Code TU : Tube TR : Tape & Reel</p>
<p>APM4953 K : APM4953 XXXXX</p>	<p>XXXXX - Date Code</p>

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Symbol	Parameter	Rating	Unit
V_{DSS}	Drain-Source Voltage	-30	V
V_{GSS}	Gate-Source Voltage	±25	
I_D^*	Maximum Drain Current – Continuous	$T_A = 25^\circ\text{C}$ -4.9	A
I_{DM}	Maximum Drain Current – Pulsed	-30	

* Surface Mounted on FR4 Board, t M10 sec.

ANPEC reserves the right to make changes to improve reliability or manufacturability without notice, and advise customers to obtain the latest version of relevant information to verify before placing orders.

Absolute Maximum Ratings (Cont.) ($T_A = 25rC$ unless otherwise noted)

Symbol	Parameter	Rating	Unit
P_D	Maximum Power Dissipation	$T_A = 25rC$	2.5
		$T_A = 100rC$	1.0
T_J	Maximum Junction Temperature	150	rC
T_{STG}	Storage Temperature Range	-55 to 150	
R_{aJA}^*	Thermal Resistance - Junction to Ambient	50	$^{\circ}C/W$

Electrical Characteristics ($T_A=25^{\circ}C$ unless otherwise noted)

Symbol	Parameter	Test Condition	APM4953			Unit
			Min.	Typ ^a .	Max.	
Static						
BV_{DSS}	Drain-Source Breakdown Voltage	$V_{GS}=0V, I_{DS}=-250- A$	-30			V
I_{DSS}	Zero Gate Voltage Drain Current	$V_{DS}=-24V, V_{GS}=0V$			-1	- A
$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS}=V_{GS}, I_{DS}=-250- A$	-1	-1.5	-2	V
I_{GSS}	Gate Leakage Current	$V_{GS}=S25V, V_{DS}=0V$			S100	nA
$R_{DS(ON)}$	Drain-Source On-state Resistance ^b	$V_{GS}=-10V, I_{DS}=-4.9A$		53	60	mr
		$V_{GS}=-4.5V, I_{DS}=-3.6A$		80	95	
V_{SD}	Diode Forward Voltage ^b	$I_{SD}=-1.7A, V_{GS}=0V$		-0.7	-1.3	V
Dynamic^a						
Q_g	Total Gate Charge	$V_{DS}=-15V, I_{GS}=-10V$ $I_D=-4.6A$		22.3	29	nC
Q_{gs}	Gate-Source Charge			4.65		
Q_{gd}	Gate-Drain Charge			2		
$t_{d(ON)}$	Turn-on Delay Time	$V_{DD}=-15V, I_D=-2A,$ $V_{GEN}=-10V, R_G=6r$ $R_L=7.5r$		10	18	ns
T_r	Turn-on Rise Time			15	20	
$t_{d(OFF)}$	Turn-off Delay Time			22	38	
T_f	Turn-off Fall Time			15	25	
C_{iss}	Input Capacitance	$V_{GS}=0V$		1260		pF
C_{oss}	Output Capacitance	$V_{DS}=-25V$		340		
C_{rss}	Reverse Transfer Capacitance	Frequency=1.0MHz		220		

Notes

- ^a : Pulse test ; pulse width M800ns, duty cycle M2m
- ^b : Guaranteed by design, not subject to production testing

ANEXO 4: NIVELES LÓGICOS DE DISPOSITIVOS TTL Y CMOS

Logic Overview

IC Basics: Comparison of Switching Standards

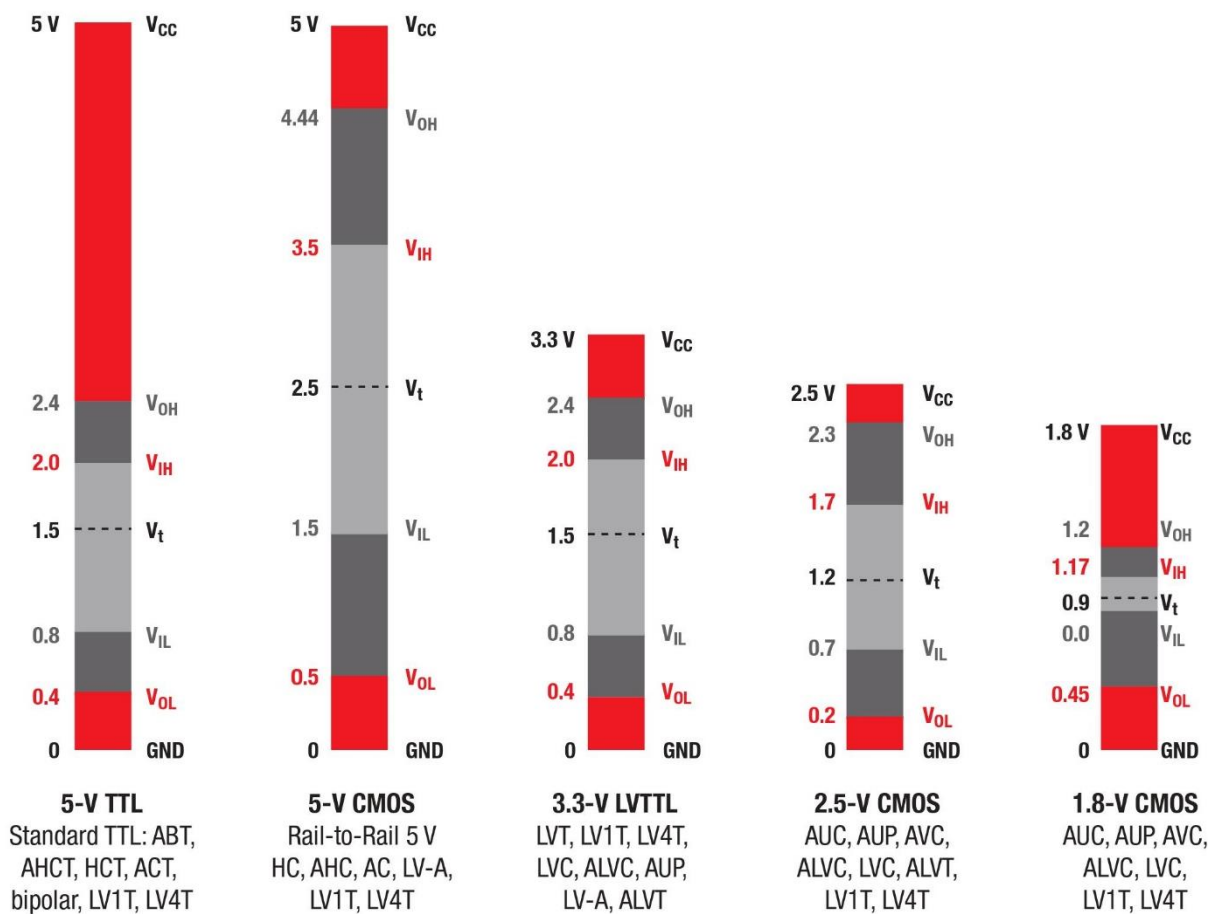
Shown below are the switching input/output comparison table and graphic that illustrate V_{IH} and V_{IL} , which are the minimum switching levels for guaranteed operation. V_t is the approximate switching level and the V_{OH} and V_{OL} levels are the guaranteed outputs for the V_{CC} specified.

Is V_{OH} higher than V_{IH} ?
Is V_{OL} less than V_{IL} ?



D \ R	5 TTL	5 CMOS	3 LVTTTL	2.5 CMOS	1.8 CMOS
5 TTL	Yes	No	Yes*	Yes*	Yes*
5 CMOS	Yes	Yes	Yes*	Yes*	Yes*
3 LVTTTL	Yes	No	Yes	Yes*	Yes*
2.5 CMOS	Yes	No	Yes	Yes	Yes*
1.8 CMOS	No	No	No	No	Yes*

* Requires V_{IH} Tolerance



**ANEXO 5: COMPARACIÓN Y COMPATIBILIDAD DE LA FAMILIA LÓGICA
HCT VS OTRAS FAMILIAS TTL Y CMOS**

An Introduction to and Comparison of 74HCT TTL Compatible CMOS Logic

Fairchild Semiconductor
Application Note 368
March 1984



The 54HC/74HC series of high speed CMOS logic is unique in that it has a sub-family of components, designated 54HCT/74HCT. Generally, when one encounters a 54/74 series number, the following letters designate some speed and power performance, usually determined by the technology used. Of course, the letters HC designate high speed CMOS with the same pinouts and functions as the 54LS/74LS series. The sub-family of HC, called HCT, is nearly identical to HC with the exception that its input levels are compatible with TTL logic levels.

This simple difference can, however, lead to some confusion as to why HCT is needed; how HCT should be used; how it is implemented; when it should be used; and how its performance compares to HC or LS. This paper will attempt to answer these questions.

It should also be noted that not all HCTs are the same. That is, HCTs from other vendors may have some characteristics that are different. Thus, when discussing general characteristics this paper will directly address Fairchild Semiconductor's 54HCT/74HCT which is compatible with **JEDEC standard 7**. Other vendors' ICs which also meet this standard will probably have similar characteristics.

WHY DOES HCT EXIST?

Ideally, when a designer sits down to design a low power high speed system, he would like to use 54HC/74HC, and CMOS LSI components. Unfortunately, due to system requirements he may have to use NMOS microprocessors and their NMOS or bipolar peripherals or bipolar logic (54S/74S,

54F/74F, 54ALS/74ALS, or 54AS/74AS) because either the specific function does not exist in CMOS or the CMOS device may not have adequate performance. Since the system designer still desires to use HC where possible, he will mix HC with these products. If these devices are specified to be TTL compatible, incompatibilities may result at the interface between the TTL, NMOS, etc. and HC.

More specifically, in the case of where a TTL or NMOS output may drive an HC input, a specification incompatibility results. *Table 1* lists the output drive specifications of TTL compatible outputs with the input specifications of 54HC/74HC. Notice that the output high level of a TTL specified device will not be guaranteed to have a logic high output voltage level that will be guaranteed to be recognized as a valid logic high input level by HC. A TTL output will be equal to or greater than 2.4V, but an HCMOS input needs at least 3.15V. It should be noted that in an actual application the TTL output will pull-up probably to about V_{CC} minus 2 diode voltages, and HC will accept voltages as low as 3V as a valid one level so that in almost all cases there is no problem driving HC with TTL.

Even with the specified incompatibility, it is possible to improve the TTL-CMOS interface without using HCT. *Figure 1* illustrates this solution. By merely tying a pull-up resistor from the TTL output to V_{CC} , this will force the output high voltage to go to V_{CC} . Thus, HC can be directly interfaced very easily to TTL. This works very well for systems with a few lines requiring pull-ups, but for many interfacing lines, HCT will be a better solution.

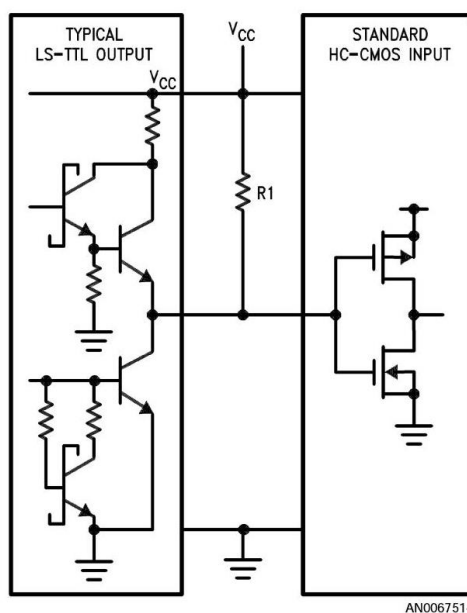


FIGURE 1. Interfacing LS-TTL Outputs to Standard CMOS Inputs Using a Pull-Up Resistor

The input high logic level of HC is the only source of incompatibility. 54HC/74HC can drive TTL easily and its input low level is TTL compatible. Again referring to *Table 1*, the logic output of the TTL type device will be recognized to be a valid logic low (0) level, so there is no incompatibility here. *Table 2* shows that the specified output drive of HC is capable of driving many LS-TTL inputs, so there is no incompatibility here either (although one should be aware of possible fanout restrictions similar to that encountered when designing with TTL).

The question then arises: since only the input high level must be altered, why not design CMOS logic to be TTL compatible? 54HC/74HC was designed to optimize performance in all areas, and making a completely TTL compatible logic family would sacrifice significant performance. Most importantly, there is a large loss of AC noise immunity, and there are speed and/or die size penalties when trying to design for TTL input levels.

Thus, since it is obvious that there is a need to interface with TTL and TTL compatible logic, yet optimum performance would be sacrificed, a limited sub-family of HCT devices was created. It is completely TTL input compatible, which enables guaranteed direct connection of TTL outputs to its inputs. In addition, HCT still provides many of the other advantages of 54HC/74HC.

WHEN TO USE 54HCT/74HCT LOGIC

The 54HCT/74HCT devices are primarily intended to be used to provide an easy method of interfacing between TTL compatible microprocessor and associated peripherals and bipolar TTL logic to 54HC/74HC. There are essentially two application areas where a designer will want to perform this interface.

1. The first case is illustrated in *Figure 2*. In this case the system is a TTL compatible microprocessor. This figure shows an NS16XXX (any NMOS μ P may be substituted) that is in a typical system and therefore must be interfaced to 54HC/74HC. In this instance, the popular gate, buffer, decoder, and flip-flop functions provided in the 54HCT/74HCT sub-family can be used to interface the many lines that come from TTL compatible outputs. It is also easy to upgrade this configuration to an *all* CMOS system once the CMOS version of the microprocessor is available by replacing the HCT with HC.
2. A second application is, when in speed-critical situations a faster logic element than HC, probably ALS or AS, must be used in a predominantly 54HC/74HC system, or a specific logic function unique to TTL is placed into an HC design. This situation is illustrated in *Figure 3*. In this case, pull-up resistors on an HC input may be sufficient, but if not, then an HCT can be used to provide the guaranteed interface.

ANEXO 6: PROGRAMA EN LENGUAJE PYTHON DEL PROTOTIPO DE PANEL LED RGB 16X32

```
1.  import RPi.GPIO as GPIO
2.  import time
3.
4.  delay = 0.000001
5.
6.  GPIO.setmode(GPIO.BCM)
7.  red1_pin = 17
8.  green1_pin = 18
9.  blue1_pin = 22
10. red2_pin = 23
11. green2_pin = 24
12. blue2_pin = 25
13. clock_pin = 3
14. a_pin = 7
15. b_pin = 8
16. c_pin = 9
17. latch_pin = 4
18. oe_pin = 2
19.
20. GPIO.setup(red1_pin, GPIO.OUT)
21. GPIO.setup(green1_pin, GPIO.OUT)
22. GPIO.setup(blue1_pin, GPIO.OUT)
23. GPIO.setup(red2_pin, GPIO.OUT)
24. GPIO.setup(green2_pin, GPIO.OUT)
25. GPIO.setup(blue2_pin, GPIO.OUT)
26. GPIO.setup(clock_pin, GPIO.OUT)
27. GPIO.setup(a_pin, GPIO.OUT)
28. GPIO.setup(b_pin, GPIO.OUT)
29. GPIO.setup(c_pin, GPIO.OUT)
30. GPIO.setup(latch_pin, GPIO.OUT)
31. GPIO.setup(oe_pin, GPIO.OUT)
32.
33. screen = [[0 for x in xrange(32)] for x in xrange(16)]
34.
35. def clock():
36.     GPIO.output(clock_pin, 1)
37.     GPIO.output(clock_pin, 0)
38.
39. def latch():
40.     GPIO.output(latch_pin, 1)
41.     GPIO.output(latch_pin, 0)
42.
43. def bits_from_int(x):
```

```

44.     a_bit = x & 1
45.     b_bit = x & 2
46.     c_bit = x & 4
47.     return (a_bit, b_bit, c_bit)
48.
49. def set_row(row):
50.     #time.sleep(delay)
51.     a_bit, b_bit, c_bit = bits_from_int(row)
52.     GPIO.output(a_pin, a_bit)
53.     GPIO.output(b_pin, b_bit)
54.     GPIO.output(c_pin, c_bit)
55.     #time.sleep(delay)
56.
57. def set_color_top(color):
58.     #time.sleep(delay)
59.     red, green, blue = bits_from_int(color)
60.     GPIO.output(red1_pin, red)
61.     GPIO.output(green1_pin, green)
62.     GPIO.output(blue1_pin, blue)
63.     #time.sleep(delay)
64.
65. def set_color_bottom(color):
66.     #time.sleep(delay)
67.     red, green, blue = bits_from_int(color)
68.     GPIO.output(red2_pin, red)
69.     GPIO.output(green2_pin, green)
70.     GPIO.output(blue2_pin, blue)
71.     #time.sleep(delay)
72.
73. def refresh():
74.     for row in range(8):
75.         GPIO.output(oe_pin, 1)
76.         set_color_top(0)
77.         set_row(row)
78.         #time.sleep(delay)
79.         for col in range(32):
80.             set_color_top(screen[row][col])
81.             set_color_bottom(screen[row+8][col])
82.             clock()
83.         #GPIO.output(oe_pin, 0)
84.         latch()
85.         GPIO.output(oe_pin, 0)
86.         time.sleep(delay)
87.
88. def fill_rectangle(x1, y1, x2, y2, color):
89.     for x in range(x1, x2):

```



```
90.         for y in range(y1, y2):
91.             screen[y][x] = color
92.
93.
94. def set_pixel(x, y, color):
95.     screen[y][x] = color
96.
97. fill_rectangle(0, 0, 12, 12, 1)
98. fill_rectangle(20, 4, 30, 15, 2)
99. fill_rectangle(15, 0, 19, 7, 7)
100.
101. while True:
102.     refresh()
```