



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**ARREGLO DE PANTALLAS OLED PARA  
ESTUDIO DE SUPERFICIES ASFÉRICAS**

**TESIS**

Que para obtener el título de

**Ingeniero eléctrico electrónico**

**P R E S E N T A**

Alejandro Flores Martínez

**DIRECTOR DE TESIS**

MI Juan Ricardo Damián Zamacona



Ciudad Universitaria, Cd. Mx., 2017

## Índice

<b>Planteamiento del problema</b>	<b>1</b>
<b>Objetivo</b>	<b>1</b>
<b>Antecedentes</b>	<b>2</b>
<b>Capítulo 1</b>	
<b>1.1 Tipos de pantallas comerciales</b>	<b>10</b>
<b>1.2 Ventajas sobre las pantallas LCD</b>	<b>10</b>
<b>1.3 Pantalla seleccionada</b>	<b>11</b>
<b>Capítulo 2</b>	
<b>2.1 Tipos de cámaras</b>	<b>13</b>
<b>2.2 Selección de cámara</b>	<b>16</b>
<b>Capítulo 3 Tarjetas de procesamiento</b>	
<b>3.1 Tipos de tarjetas</b>	<b>17</b>
<b>3.2 Tarjeta seleccionada</b>	<b>19</b>
<b>3.3 Ventajas de un Sistemas Operativos Linux</b>	<b>21</b>
<b>Capítulo 4 Procesamiento de Imágenes</b>	<b>22</b>
<b>Capítulo 5 Manejo de Archivos</b>	
<b>5.1 Despliegue en Pantalla</b>	<b>27</b>
<b>5.2 Captura de la foto</b>	<b>32</b>

<b>Capítulo 6 Sistema de comunicación Wi-Fi</b>	
<b>6.1 IEEE 802.11.b/g/n</b>	<b>33</b>
<b>6.2 Comunicación Servidor-Cliente usando Flask</b>	<b>34</b>
<b>6.3 Sistema de control del equipo</b>	<b>37</b>
<b>6.4 Fuente de alimentación</b>	<b>40</b>
<b>Resultados</b>	<b>44</b>
<b>Conclusiones y trabajo a futuro</b>	<b>45</b>
<b>Apéndice A</b>	<b>46</b>
<b>Bibliografía</b>	<b>48</b>



## **Agradecimientos**

A la UNAM por todos estos años de enseñanza, a su facultad de ingeniería que me permitió convertirme en profesionista.

Al Laboratorio de Electrónica del CECADET especialmente a mi asesor M.I. J. Ricardo Damián Zamacona y al M.I. Salvador Rodríguez Lozano quien fabrico los circuitos impresos necesarios para realizar el presente trabajo ya que sin la ayuda de ambos no hubiera sido posible realizarlo

Así mismo un agradecimiento al PAPIIT “desarrollos en topografía corneal” con clave IT101216 por los recursos otorgados.

## **Dedicatoria**

La tesis se la dedico a mi familia a mis padres: Joel Flores Rentería, Lucia Martínez Torres y hermano Ernesto Flores Martínez quienes siempre me han querido, apoyado y cuidado en cada momento y a mi niña Diana Luz, en donde quiera que estés.

## **Planteamiento del problema**

Para poder detectar deformaciones de la córnea del ojo humano existen en el mercado equipos muy caros y sofisticados, que trabajan mediante la proyección de imágenes predefinidas sobre la córnea, estos equipos son de banco y muy robustos, por lo mismo, son difíciles de transportar para realizar estudios en cierto tipo de pacientes. Por otro lado, en el CCADET se ha estado estudiando la forma de sustituir las imágenes que comercialmente sirven de patrón para el estudio del ojo por arreglos de manchas conocidas como pantallas nulas y de esta forma poder estudiar la topografía corneal por otro método, por ello es que en este trabajo se presenta una alternativa de solución al problema que se tiene de proyectar este tipo de imágenes sobre el ojo de una forma dinámica. Para ello se propone la utilización de pantallas OLED, este tipo de pantallas son parecidas a las típicas LCD pero con características más adecuadas para la presentación de las imágenes requeridas en este proyecto, el arreglo final debe de ser compacto para poder lograr un equipo de tamaño adecuado que pueda ser portátil. Esto está básicamente encaminado a mejorar un videoqueratómetro previamente diseñado y construido en el CCADET de la UNAM.

Como ya se mencionó, los equipos comerciales utilizados para realizar estudios de córnea son fijos y el paciente tiene que mantenerse quieto, cuando se realizan estos estudios a infantes se complica el poder tomar las imágenes de forma adecuada, por este motivo se busca tener un videoqueratómetro fácil de manipular y que sea portátil.

### **Objetivo**

Desarrollar un equipo que mediante el empleo de pantallas OLED pueda desplegar un arreglo de imágenes patrón sobre una superficie curva, de tal forma que estas se puedan proyectar sobre una superficie en forma estática o dinámica, las imágenes previamente generadas y almacenadas en un archivo se podrán desplegar con el arreglo de pantallas OLED y se podrá capturar la imagen reflejada sobre la superficie curva mediante una cámara CMOS para posteriormente ser almacenada en otro archivo para su posterior estudio. Este proceso de despliegue y captura de imágenes se podrá llevar a cabo desde una computadora conectada al equipo.

## Antecedentes

- **Problemas visuales**

La visión humana es uno de los sentido que más información proporciona al cerebro, el ojo es el órgano encargado de procesar la luz y transformarla en impulsos eléctricos que el cerebro recibe y procesa, podemos decir que es un transductor de radiación electromagnética a impulsos nerviosos, este transductor lo podemos dividir básicamente en dos secciones, la parte óptica que es un arreglo de lentes biológicas los cuales refractan los rayos de luz formando una imagen y la parte de sentido, donde se forma la imagen en la retina la cual está involucrada en la transformación de la radiación electromagnética en impulsos nerviosos.

La córnea es la lente por donde entra la radiación electromagnética y es donde se realiza la mayor parte de la refracción del ojo. En este componente del ojo pueden existir en algunas personas diversos problemas relacionados a la refracción, los más frecuentes son: Astigmatismo, Miopía, Hipermetropía y Queratocono

Astigmatismo:

En este problema la córnea se deforma achatándose y perdiendo su forma casi esférica para tener una forma más parecida a una elipse, teniendo como efecto que la convergencia de los rayos se encuentre antes o después de la retina, como resultado de dicha convergencia el paciente tiene problema para enfocar objetos tanto cercanos como lejanos. En la figura 1 se muestra como es este padecimiento.



a) Ojo con problema de astigmatismo

b) corrección con lente

Figura 1. Ojo con problema de astigmatismo

Podemos observar en la figura 1 a) como puede ser la refracción de las imágenes en un ojo con problemas de astigmatismo, mientras que en la figura 1 b) la corrección del problema con una lente cóncava-convexa.

Miopía:

Las personas con este padecimiento no pueden ver objetos lejanos esto se debe a que la convergencia de los rayos de luz se produce delante de la retina, este padecimiento no es por un problema en la córnea, se produce en pacientes que tienen un globo ocular más largo en comparación a las



personas que ven bien, en la figura 2 a) se observa de forma gráfica donde quedaría la imagen refractada en un ojo con miopía, también su corrección con una lente bicóncava en la figura 2 b).

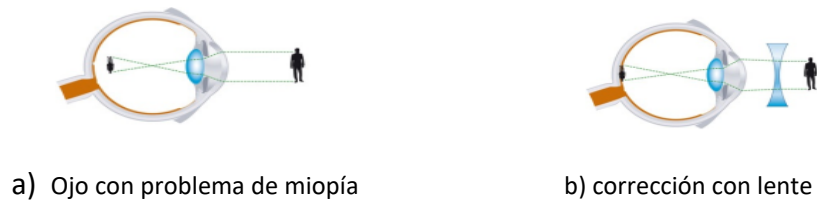


Figura 2 problema de miopía

### Hipermetropía:

En esta alteración del ojo el paciente tiene problemas de visión a distancias cortas, los rayos de luz convergen en la parte trasera de la retina, no es progresivo y se corrige con la utilización de lentes convergentes la causa de este problema radica en que el globo ocular es demasiado corto, como se aprecia en la figura 3 a) la imagen se forma afuera del ojo lo que produce una mala visión, para dar solución a éste tipo de problema se emplean lentes biconvexas en la figura 3 b) podemos observar los efectos de ellas

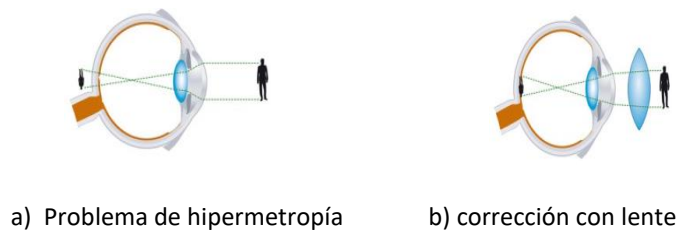


Figura 3 problema de miopía

### Queratocono:

Esta enfermedad de la córnea produce mala visión, en su etapa inicial puede mantenerse sin cambios bruscos, este padecimiento es debido a que la córnea adopta una forma de cono, a pesar de ser progresiva se puede detener usando diferentes métodos como lentes de contacto, gafas o en su fase final trasplante de córnea.

Los síntomas del Queratocono en su etapa inicial pueden confundirse fácilmente con una miopía o astigmatismo y son tratados con lentes normales después de un tiempo las imágenes se tornan borrosas y el astigmatismo aumenta al grado que los lentes convencionales dejan de ser una opción y solo es posible corregir con lentes de contacto en este punto es cuando se empiezan a realizar estudios para Queratocono, otro síntoma es la hipersensibilidad a la luz o picazón en el ojo los cuales no necesariamente se pueden presentar, se recomienda no frotarse el ojo vigorosamente si se

tiene este padecimiento ya que podría acelerar la enfermedad, en la figura 4 se muestra la forma que adopta la córnea cuando padece ésta enfermedad.

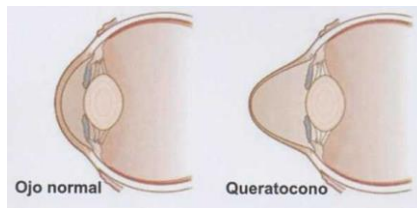


Figura 4. Deformación de la córnea con Queratocono

Para el diagnóstico del Queratocono se pueden utilizar diferentes exámenes entre los cuales están: examen de la córnea, con lámpara de hendidura en la figura 5 podemos observar el ojo de forma lateral y el haz de luz que lo recorre para realizar el examen y el más exacto es una topografía corneal el cual crea un mapa de la curvatura de la córnea.

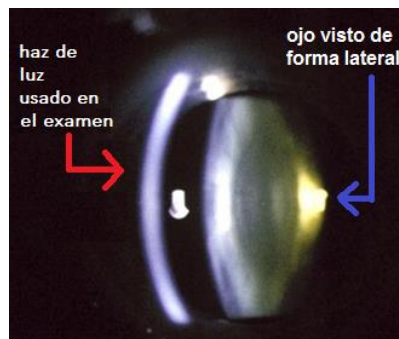


Figura 5. Examen con lámpara de hendidura

- **Equipos de diagnóstico**

- **Videoqueratómetros comerciales**

- Existen diferentes equipos para medir la forma de la córnea, los primeros equipos de este tipo, dejaban los resultados a la interpretación del médico que los usaba, después se cambió al uso de sistemas computarizados, los equipos más utilizados se muestran a continuación.

- **Queratoscopio**

- Este instrumento fue inventado por Antonio Plácido da Costa el cual consiste en una serie de círculos concéntricos alrededor de un agujero, estos círculos alternan los colores blanco y negro, en la figura 6 se muestra un ejemplo de éste dispositivo, dependiendo de la deformación de los círculos reflejados en el ojo se puede saber qué tipo de padecimiento se presenta.



Figura 6 Queratoscopio

### Videoqueratómetro

Es un instrumento computarizado que le permite al médico tomar medidas queratométricas de forma más exactas, funciona con un disco de Plácido y algunos también usan una lámpara de hendidura, en la figura 7 podemos apreciar un ejemplo de estos dispositivos.

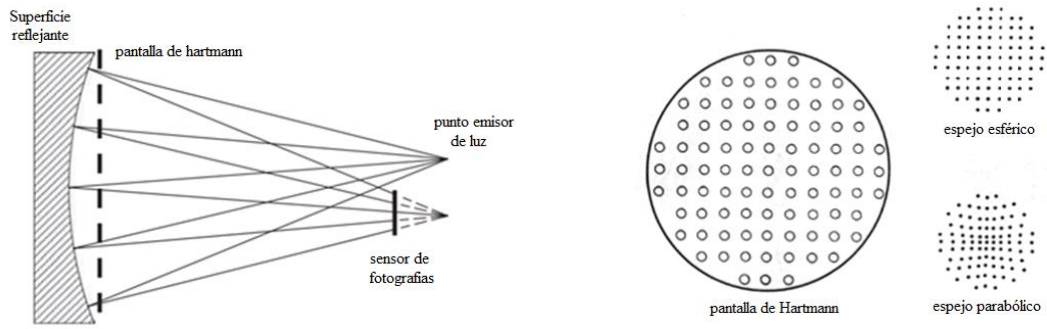


Figura 7. Videoqueratómetro comercial

- **Pantallas nulas**

Las pantallas nulas son un concepto que se basan en la prueba de Hartmann, la prueba clásica de Hartmann consiste en tener una máscara con pequeños agujeros y una fuente de luz que produce rayos discretos los cuales se utilizan para medir las

deformidades que se generan en la superficie que se desea analizar, en la figura 8 a) se muestra la forma en que se realiza la prueba de Hartmann mientras que en la figura 8 b) podemos observar la pantalla usada y los efectos que causa en dos tipos de espejos.

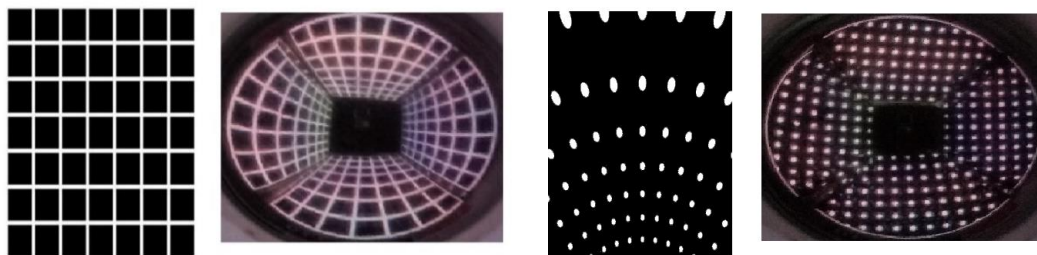


a) Prueba clásica de Hartmann

b) Pantalla usada y sus efectos en espejos

Figura 8. Método de Hartman para prueba de superficies

El funcionamiento de las pantallas nulas para el estudio de la curvatura de superficies ópticas es de la siguiente manera, si se proyecta una pantalla plana grabada con una figura conocida en una superficie plana bajo estudio no se generarán deformaciones en la imagen proyectada, pero por el contrario, si se proyecta esta misma pantalla en una superficie esférica la imagen proyectada se deformará debido a la curvatura de la superficie ver figura 9 a), tomando en cuenta este principio para saber si la superficie a analizar es perfecta se puede generar una pantalla con la forma específica de la superficie en estudio, ver figura 9 b) y si la imagen proyectada no presenta deformaciones se puede decir que la superficie no tiene errores, si por lo contrario, la proyección presenta deformaciones, entonces se puede sospechar que la superficie presenta deformaciones, entre más desviada sea la imagen proyectada mayores serán las deformaciones de la superficie, por este motivo esta técnica se llama método de pantallas nulas.



a) Pantalla plana en superficie curva

b) Pantalla curva en superficie curva

Figura 9 pantallas nulas aplicadas a una superficie esférica

- **Generación de Pantallas nulas**

En el laboratorio de óptica del CCADET se han venido desarrollando varias formas de generar pantallas nulas para pruebas ópticas, en un principio estas pantallas eran físicas y estaban construidas y grabadas en una pantalla translúcida tal como un rollo de papel o acetato el cual era iluminado por detrás, para tener la proyección sobre la superficie a analizar tal como se muestra en la figura 10, estas estaban construidas con una forma especial para cada objeto de estudio.

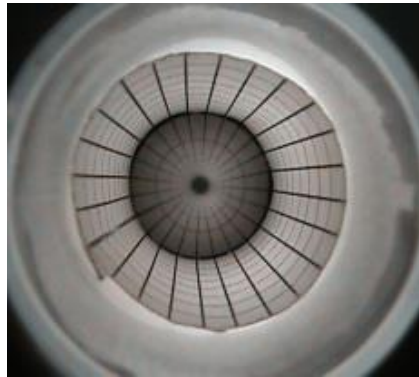


Figura 10. Pantallas nulas

Posteriormente se introdujo la idea de una pantalla nula dinámica con la ayuda de pantallas LCD en un arreglo geométrico. Con esta nueva técnica, la imagen generada se refleja en la superficie bajo estudio generando una pantalla nula, la cual puede modificarse de forma constante, al tener acceso a colores y no solo blanco y negro, con esta técnica se pudo generar una mayor densidad de puntos y tener imágenes con una resolución más alta. El inconveniente de esta técnica utilizando pantallas LCD es que el ángulo de vista de las pantallas es muy reducido, lo cual provoca que las imágenes proyectadas en algunas zonas se vean borrosas dificultando su posterior estudio con técnicas computacionales, por otro lado, el brillo que producen las pantallas es muy bajo para poder reflejar con nitidez la figura en el ojo, tal como se puede observar en la figura 11.

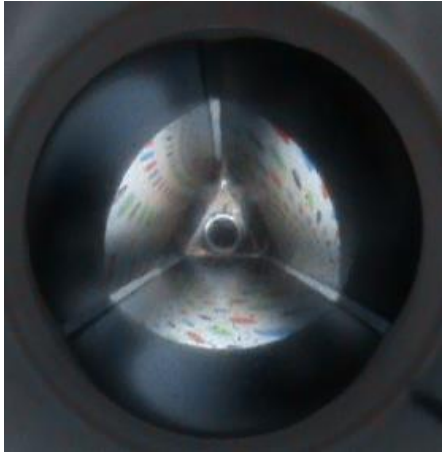


Figura 11. Prueba de pantalla nula con gotas de color

- **Sistema propuesto para la generación de pantallas nulas con pantallas OLED (diagrama a bloques)**

En la figura 12 se muestra un diagrama general del sistema empleado, como se puede observar los 4 bloques principales son:

1. La fuente de alimentación de DC  
Este módulo proporciona la energía necesaria para que el equipo funcione, se conecta a la línea de 120[V] y proporciona salidas de 5 [V], 4.5 [V], 3 [V] y -4.5 [V] de DC con una corriente máxima de 3 [A]
2. Las pantallas  
Son pantallas OLED de 2.8'' y su función es proyectar una imagen la cual se refleja en la superficie a analizar.
3. La cámara  
Tiene un sensor CCD de 8 mega PÍXELES y su función es tomar fotos y video de la superficie a analizar.
4. El modem  
Es el encargado de realizar las comunicaciones y ejecutar los programas requeridos para el funcionamiento del equipo.

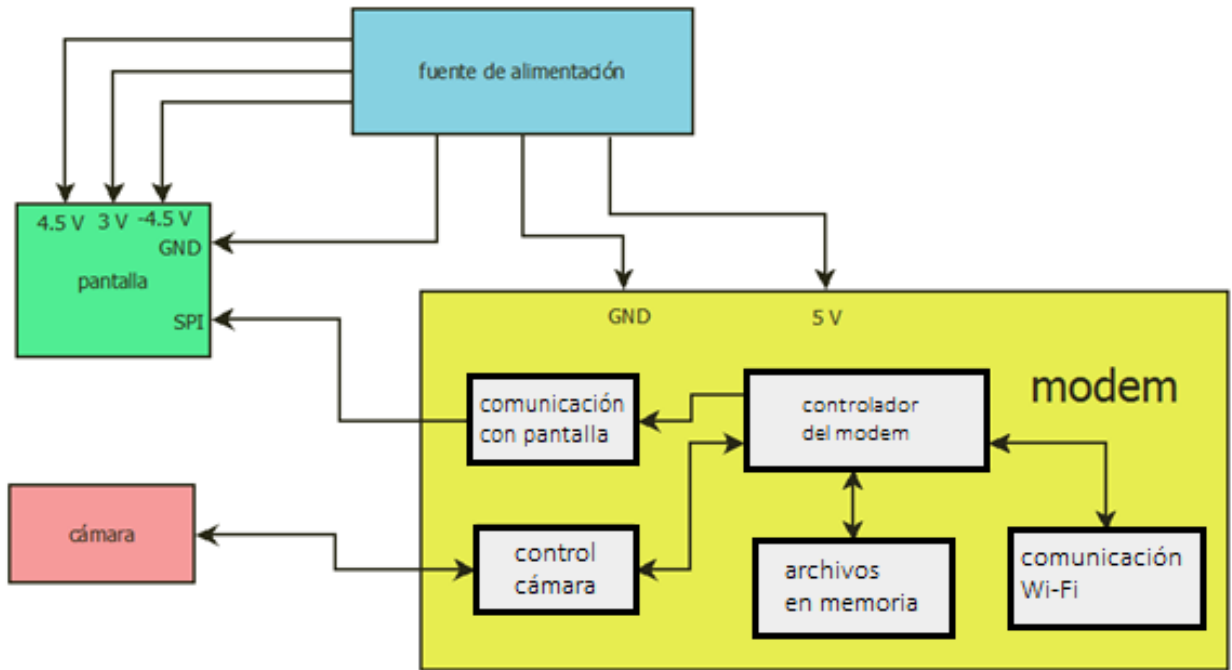


Figura 12. Diagrama a bloques del sistema propuesto para la generación de pantallas nulas dinámicas

## Capítulo 1

### Pantallas OLED

#### 1.1 Tipos de pantallas comerciales

Existen diferentes tipos de pantallas electrónicas comerciales para la generación o presentación de imágenes, algunas ya en fuera de uso como la CRT, pero en este apartado solo se abordarán dos tipos, que en la actualidad son las más usadas las LCD (pantallas de cristal líquido) y las OLED (diodo emisor de luz orgánico), el funcionamiento es muy diferente en cada una de ellas, por un lado las pantallas LCD están formadas por diferentes capas, una fuente de luz (Backlight), filtros polarizados y una capa de cristal líquido, de aquí su nombre, la cual se puede excitar permitiendo en menor o mayor medida el paso de luz y dando lugar a los pixeles. Por su parte, las pantallas OLED tienen diodos emisores de luz de tres colores rojo verde y azul, pueden tener un LED o varios del mismo color, los tres colores forman un pixel como se ve en la figura (1.1).

Las pantallas OLED son actualmente muy usadas en diversos equipos como teléfonos inteligentes, laptops, televisiones o tabletas, en general se venden en equipos ya terminados dedicados al área de entretenimiento, los cuales tienen un costo mayor a los equipos con tecnología LCD. Para los laboratorios de desarrollo electrónico que no pertenecen a las grandes transnacionales, las pantallas LCD han sido su principal opción, principalmente por la relación costo beneficio, por este motivo existen pocas opciones de pantallas OLED destinadas a desarrollo de equipos de investigación.

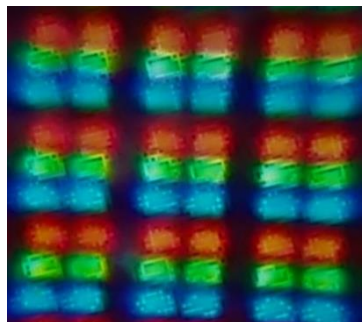


Figura 1.1 Imagen que muestra 6 pixeles de una pantalla OLED, cada pixel contiene dos LEDs de cada color

#### 1.2 Ventajas sobre la pantalla LCD

Para hablar de las ventajas de las pantallas OLED sobre las LCD es necesario poner en contexto el problema a resolver ya que para muchos casos las LCD resultan ser más adecuadas para algunas aplicaciones, especialmente si a costo se refiere, para lograr la optimización de imágenes en este proyecto se requiere un arreglo geométrico de cuatro pantallas que proyecten las imágenes de las pantallas a una superficie reflejante, como



podría ser la retina del ojo, con un alto nivel de contraste, un adecuado nivel de brillo y gran nitidez de color.

En estos casos las pantallas OLED seleccionadas presentan ventajas significativas sobre las LCD, su rango de visión es casi de 180°, en promedio 170°, el contraste es perfecto al tener control de encendido y apagado de cada LED, se pueden usar solo los pixeles requeridos. Como beneficio de esto, las imágenes no tienen un degradamiento de color en los bordes, además nos asegura que cuando se usa un color básico este no tendrá componentes de otro color.

Con la práctica se ha visto que las pantallas oled seleccionadas presentan un pequeño inconveniente que finalmente no es muy crítico dado que la posición de la superficie reflejante(ojo) dentro del arreglo óptico siempre será la misma, este inconveniente consiste en que a diferentes ángulos de vista, el color que se percibe de las imágenes puede variar, esto es debido a la geometría del arreglo usado en la matriz de LED, a pesar de este aparente error en las pantallas, además de su costo elevado y complejidad para operarlas, se optó por usar OLED en lugar de LCD ya que sus ventajas superan los problemas que pueden ocasionar.

### 1.3 Pantalla seleccionada

Las pantallas que finalmente se seleccionaron para este proyecto, se muestran en la figura 1.2, estas pantallas son de la empresa china CHI MEI EL CORPORATION, son pantallas OLED “full color” capaces de desplegar 262000 colores por pixel usando un formato RGB 666, el tamaño de las pantallas es 2.83 (in) y tiene una resolución de 240xRGBx320 con un ancho de pixel de 0.060x0.180 [mm], estas pantallas requieren para su operación de un circuito electrónico adicional que es necesario de fabricar por separado dado que no está disponible para su venta, sin este circuito las pantallas no se pueden operar adecuadamente, también es necesario incluir en este circuito un conector especial para tener acceso a las líneas de conexión de las pantallas. La circuitería interna de la pantalla tiene un banco de memoria y un controlador que se encarga de enviar las imágenes a la matriz de leds de la pantalla.

El controlador de la pantalla es el S6E63D6 de la empresa Samsung el cual cuenta con 3 interfaces para comunicarse las cuales son:

1. MPU i80, i68 -system 18-/16-/9-/8-bit bus interface
2. Serial data transfer interface
3. RGB 18-/16-/6-bit bus interface (DOTCLK, VSYNC, HSYNC, DE, DB17-0)

En la figura 1.2 se muestra una de las pantallas utilizada en la que se aprecia por un lado el conector de las pantallas junto con el cable flexible, por el otro lado, se muestra el área útil para desplegar las imágenes.

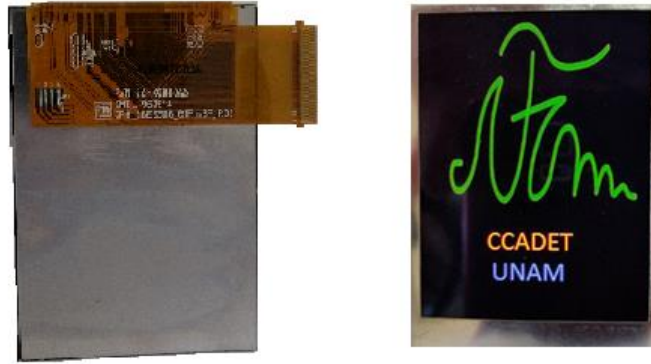


Figura 1.2 pantalla usada

Es importante resaltar que sin importar la interfaz que se quiera usar, es necesario inicializar la pantalla con el protocolo SPI, pero por otro lado el hardware adicional requerido para el envío de las imágenes hacia las pantallas es diferente y va de acuerdo a la interfaz seleccionada para la comunicación, es por ello que es de suma importancia seleccionar el tipo de comunicación desde el inicio, para este proyecto se seleccionó la comunicación SPI, en la figura 1.3 se muestra una imagen de las tarjetas electrónicas diseñadas y fabricadas en el CCADET para realizar la conexión de cada una de las pantallas, el circuito impreso es un circuito de doble cara por su densidad de pistas las cuales miden 0.3 [mm] con una separación entre pistas de 0.3 [mm] en la parte del conector.

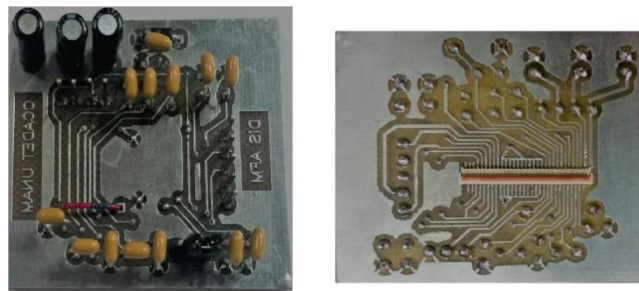


Figura 1.3 Electrónica y conector de pantallas OLED

Los niveles de tensión eléctrica que requieren las pantallas son tres diferentes valores, una polarización bipolar con valor máximo de 4.5 [V] y una referencia de 3[V] para la escala de grises.

## Capítulo 2

### Cámara para captura de imagen

#### 2.1 Tipos de cámaras

Para la captura de la imagen reflejada en la superficie bajo estudio una vez proyectada desde el arreglo de pantallas OLED, es necesario el uso de una cámara digital dado que la imagen se tiene que enviar a una computadora para su análisis, en el mercado existen diversos tipos de cámaras, es por ello que se requiere de un análisis de mercado para realizar una correcta selección de la más adecuada para el proyecto.

Las cámaras se dividen en dos grandes tipos, las analógicas que prácticamente están en desuso y las digitales que son las que usamos cotidianamente. El principio de funcionamiento óptico es el mismo en ambos casos, se tiene una lente que enfoca las imágenes, un obturador y un diafragma, este últimos, restringe el paso de luz a una zona donde es guardada la imagen, en el caso de las cámaras analógicas se tiene una película fotosensible que se encarga de guardar las imágenes las cuales son reveladas posteriormente en un cuarto oscuro, por su parte, las cámaras digitales sustituyen la película fotosensible por un sensor y se puede visualizar las fotos al momento de ser tomadas.

-Cámaras digitales

Podemos descomponer una cámara digital básicamente en cuatro componentes principales, sensor, visor, memoria y software.

- Sensor

Los tipos de sensores que ocupan las cámaras normalmente son dos, CCD y CMOS los dos sensores estas formados de celdas, éstas acumulan una carga eléctrica la cual es directamente proporcional a la intensidad de luz que incide sobre ella, las celdas son los llamados pixeles y están dispuestas en un arreglo matricial.

La diferencia entre las dos tecnologías radica principalmente en la forma en la cual procesan esta información, los sensores CCD necesitan de un convertidor analógico digital (ADC) externo, cuentan con registros de corrimiento verticales y uno horizontal, estos registros están formados por capacitores e interruptores interconectados a lo largo de una fila, en la figura (2.1) se muestra estos registros, para procesar la imagen, el registro horizontal envía los PIXELES (cargas eléctricas) uno por uno al ADC, una vez terminado este envió, los registros verticales alimentan al registro horizontal y se repite el proceso hasta recorrer toda la imagen.

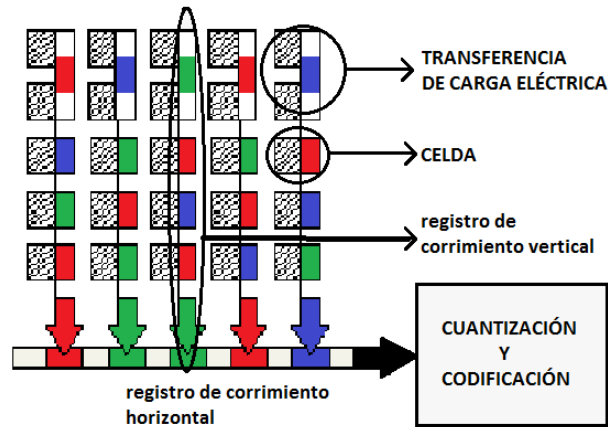


Figura 2.1 Sensor CCD

Por su parte los sensores CMOS cuantizan con un arreglo de transistores las cargas eléctricas dentro de las mismas celdas, por lo que no requieren un circuito externo, con los sensores CMOS se puede leer directamente el valor de cada PIXEL, esta tecnología es mucho más rápida que la CCD, adicionalmente prevé errores tales como el llamado “*blooming*” que se produce cuando la celda recibe una intensidad de luz muy grande, con lo cual se afectan los pixeles adyacentes. En contraparte, el problema principal de los sensores CMOS es que cada celda requiere electrónica adicional, con lo que se aumenta el tamaño de ésta, por tal motivo se reduce la resolución de la imagen comparada con un sensor CCD del mismo tamaño, en la figura (2.2) se muestra un esquema del sensor CMOS.

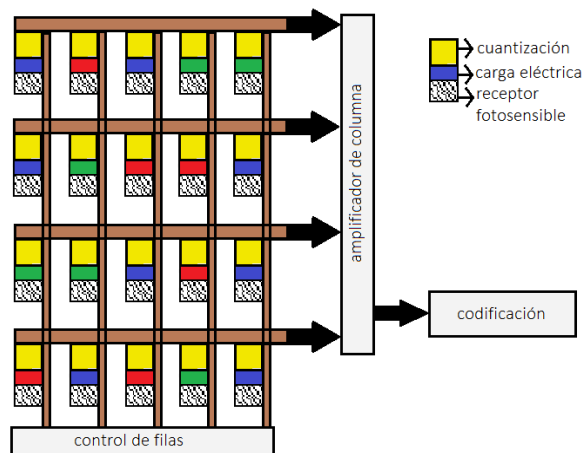


Figura 2.2 sensor CMOS

- Visor  
El visor es el modo por el cual el usuario de la cámara enfoca el objeto antes de tomar la foto, existen tres tipos de visores:

1. Ocular directo: Este tipo de visor ofrece un ángulo de visión paralelo al de la lente y está situado generalmente en la parte superior de las cámaras. Este tipo de visores puede presentar problemas de paralaje cuando el objeto a fotografiar se encuentra a una distancia cercana a la cámara, en la figura 2.3 podemos observar este efecto, a pesar de que el visor enfoca la letra 'a' el sensor se encuentra en la parte inferior y enfoca solo una parte de la letra a este efecto se le llama paralaje. El visor ocular directo se encuentra prácticamente en desuso.
2. Visor ocular réflex: Las cámaras profesionales de 35 [mm] usan comúnmente éste tipo de visores, los sistemas réflex ocupan un arreglo de espejos llamado pentaprisma para llevar la imagen al visor prácticamente igual que la que llega al sensor, su factor de cobertura oscila entre 90% y 99% de la imagen vista dependiendo de la cámara.
3. Visor de pantalla: este tipo de tecnología usa una pantalla LCD en la que se muestra la imagen real que el sensor ésta viendo, tiene la ventaja de poder incluir filtros o modificaciones a la imagen desde el momento de la captura, al no necesitar un arreglo de espejos se reduce las dimensiones y costos de las cámaras con este tipo de visor, en la actualidad el visor de pantalla está remplazando a los sistemas réflex.

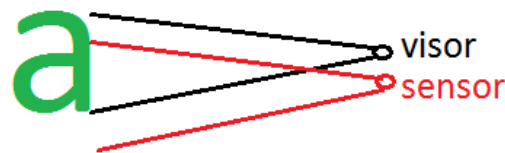


Figura 2.3 Efecto de paralaje

- Memoria

Las cámaras digitales ocupan dos tipos de memoria una interna y otra extraíble, la capacidad de almacenamiento y el tipo memoria varía dependiendo de la marca y principalmente el año de construcción de la cámara, la memoria interna también conocida como "buffer" es de gran importancia para realizar un procesamiento de la imagen antes de ser guardada en la memoria extraíble.

Las memorias extraíbles que se han utilizado en las cámaras digitales son:

**-Compact Flash**

Las memorias Compact Flash fueron famosas en los años 2000 a 2005 aproximadamente, aunque su tamaño puede parecer ahora grande, pero eran compactas en su tiempo, actualmente es raro encontrar una memoria de este tipo en algún dispositivo.

**-MMC y SDC**

Las MMC y las SDC son tarjetas de memoria más compactas, aunque el funcionamiento de estas dos memorias es similar entre sí, las MMC, por su parte no cuentan con el seguro de escritura de las SDC, las tarjetas SD, como mejor se les conoce, básicamente se diferencian por su capacidad de almacenamiento, su tamaño físico y el formato de archivos que pueden usar.

- Software

Aunque no se aprecie físicamente todas las cámaras cuentan con un programa para procesamiento de las imágenes con el cual también se puede eliminar posibles errores en la captura de estas, el procesamiento de corrección de errores, cambia de una cámara a otra y al ser secreto de industria no se tiene acceso a él, lo que sí se sabe es que algunas cámaras ocupan un detector de errores el cual hace un barrido de la imagen y revisa que no exista un pixel con un color exageradamente diferente a los que están a su alrededor, si existiera lo corrige igualando el color al resto de los PIXELES.

## 2.2 Selección de la cámara

Para este proyecto se seleccionó la cámara nativa de la Raspberry llamada “Pi NoIR V2.1” por poseer las siguientes características que se consideraron importantes para la captura de las imágenes, en la figura 2.4 se muestra una imagen de la cámara seleccionada.

- Sensor de imagen CMOS Sony IMX219 con longitud focal fija
- Resolución de 8 megapíxel
- Conector MIPI Camera Serial interfaces (CSI-2)
- Control de exposición automático
- Balance de blancos automático,
- Nivel de negros automático

Este tipo de cámaras no cuenta con un filtro para longitudes de onda infrarroja por lo que las imágenes tomadas bajo la luz solar, se obtienen de color rosado o rojizo, en contra parte, estas cámaras permiten tomar fotos de noche y con poca intensidad de luz, la Raspberry Pi 3 almacena las imágenes en su tarjeta SD y el visor que se ocupa es un “visor de pantalla”, de igual manera la Raspberry Pi 3 muestra la imagen tomada en el dispositivo del usuario.



Figura 2.4 Pi NoIR V2.1

## Capítulo 3

### Tarjetas de procesamiento

#### 3.1 Tipos de tarjetas

Contar con una tarjeta de desarrollo agiliza en gran medida la realización de un prototipo, estas tarjetas pueden ser comerciales o de diseño propio, las tarjetas de desarrollo comerciales son de uso general y se apoyan en otras tarjetas complementarias las cuales se ensamblan en los pines de conexión como se aprecia en la figura 3.1 b). Las tarjetas no comerciales son diseñadas de tal forma que se optimice la menor cantidad de tarjetas complementarias, en la figura 3.1 a) se presenta un ejemplo de estas tarjetas donde ya se incluye la pantalla LCD.



a) Tarjeta de desarrollo PIC 18Fxxxx DIS: JCH CCADET



b) Tarjeta Arduino UNO y tarjeta de expansión LCD

Figura 3.1 tarjetas de desarrollo

Sin importar si las tarjeas son comerciales o diseñadas personalmente, se pueden diferenciar por su capacidad de procesamiento de datos y velocidad de ciclos de operación.

Analizando las tarjetas de menor velocidad y capacidad de procesamiento, éstas solo cuentan con un microcontrolador (MCU) por sus siglas en inglés y se usan para proyectos que requieren un procesamiento de datos que no sobrepase el número de

las operaciones básicas, suma y multiplicación; en electrónica este tipo de tarjetas son de las más socorridas por su gran versatilidad.

Adicionalmente a las tarjetas que utilizan chips MCU se encuentran las tarjetas que usan chips con mayor velocidad y propósitos más específicos. Dentro de estas encontramos aquellas que usan los DSPIC y los DSP, estos circuitos son utilizados para tener un rápido procesamiento en tiempo real, son procesadores más rápidos que se apoyan con Hardware especial para realizar operaciones matemáticas en un solo ciclo de reloj, este tipo de tarjetas son usadas principalmente en procesamiento de señal como audio o video y son capaces de implementar algoritmos complejos como lo es la transformada rápida de Fourier, convolución, correlación, etc. En la figura 3.2 observamos un DSP en una tarjeta de la empresa Texas Instruments (TI).

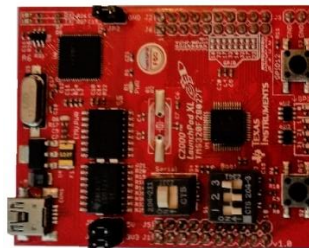


Figura 3.2 DSP, C2000 piccolo launchpad

En el mercado existen otras tarjetas para usos muy específicos, tales como aquellas que cuentan con los sistemas llamados “SoC” (sistemas dentro de un chip), a pesar de que todas estas tarjetas pueden utilizar sistemas operativos al igual que una tarjeta de computadora, el Hardware cambia significativamente entre ellas, esto se debe a que cada tarjeta está diseñada para desempeñar un mejor papel en ciertas áreas de trabajo, por ejemplo una tarjeta “Beaglebone Black” cuenta con varios puertos digitales, PWM y entradas analógicas lo cual la hace una buena opción para el control de sistemas electrónicos, por otra parte la Raspberry Pi 3 cuenta con una interconexión inalámbrica por Hardware además de tener una cámara digital y una pantalla táctil como Hardware complementario por lo que es una buena opción para sistemas de computación. También existen otras tarjetas que combinan los SoC con los arreglos de puertas lógicas programables (FPGA), un ejemplo de estas tarjetas es la llamada “ZYBO”, ésta puede funcionar como una mini computadora (SBC) o simplemente se le puede grabar una rutina de programación como si fueran un MCU solo que adicionalmente su hardware puede ser reconfigurable, por lo que con estas tarjetas se tiene una alternativa más completa ya que con ellas se puede tener Software, Drivers y Hardware a la medida del proyecto, en la figura 3.3 se muestra la “Beaglebone Black” y la “ZYBO”.



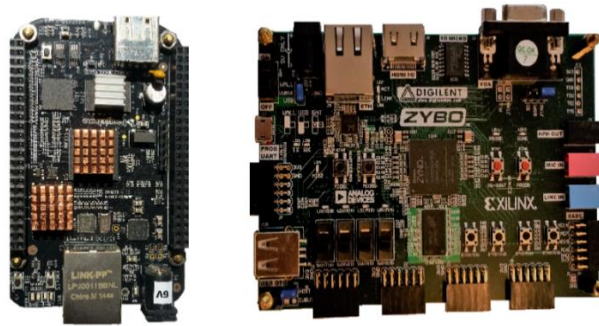


Figura 3.3 Tarjetas de desarrollo SoC

### 3.2 Tarjeta seleccionada

La tarjeta seleccionada para el control del sistema propuesto es la Raspberry Pi 3, una imagen de esta tarjeta se muestra en la figura 3.4, sus dimensiones son (8.5 x 5.4 cm), al ser una tarjeta SBC en un solo chip incluye el CPU la memoria RAM y la tarjeta gráfica. En el año de 2006 se desarrollaron los primeros diseños de este tipo de tarjetas, aunque no fueron comerciales hasta el año de 2012, la idea principal fue desarrollar una computadora para las escuelas que fuera barata, pequeña y con un bajo consumo de energía, para febrero del 2015 salió a la venta la Raspberry Pi 2 y un año después la versión 3 que incluye comunicación inalámbrica 802.11 b/g/n LAN mejor conocida como Wi-Fi y Bluetooth.



Figura 3.4 Raspberry Pi 3

Por el hecho de ser una computadora necesita tener un Sistema Operativo para su operación, en lugar de un disco duro la Raspberry puede utilizar una tarjeta micro SD como unidad de almacenamiento o en su defecto una memoria USB, existen diferentes Sistemas Operativos diseñados para esta tarjeta los cuales en su mayoría se basan en Linux, en este trabajo se utiliza el sistema llamado “Raspbian” que es una versión modificada de los sistemas “Debian”.

El Sistema Operativo se puede instalar de dos formas:

- 1) Se puede descargar la imagen ISO del Sistema Operativo directamente de la página oficial de la fundación Raspberry, la cual se guarda posteriormente en la memoria SD.
- 2) Descargar de la página oficial de la fundación Raspberry, un instalador llamado "NOOBS" el cual cuenta con diferentes sistemas y los paquetes básicos usados por la Raspberry Pi 3.

Al tener un Sistema Operativo propio es necesario contar con un teclado y un monitor, de otra forma, será necesario operarla remotamente mediante un intérprete de órdenes seguro (SSH) o por medio de un escritorio remoto, también existe otra manera adicional de acceder a la Raspberry, la cual es mediante el uso de una comunicación UART; la comunicación serial nos brinda un acceso directo a la termina del Sistema Operativo Raspbian.

Otro punto que requiere atención es el manejo de los puertos de entrada salida de propósito general, existen puertos que solo son de entrada-salida digital y otros que cuentan con una función extra la cual esta dedica a dos protocolos de comunicación "SPI" o "I2C", ver figura 3.5.

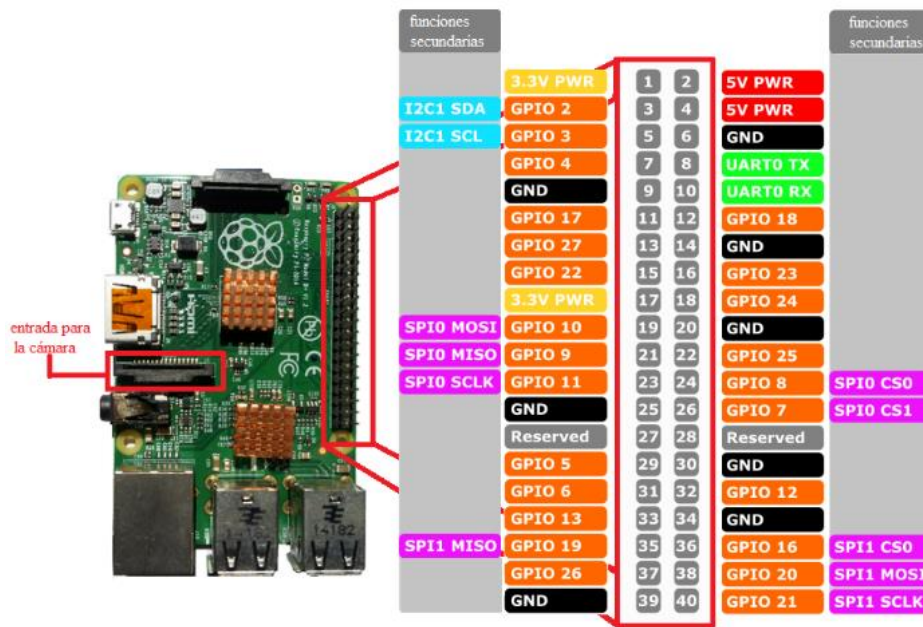


Figura 3.5 Pines de salida de la Raspberry Pi 3

Para el manejo de los puertos de manera que se pudieran controlar los dispositivos externos se utilizaron dos lenguajes de programación, el primero fue Python, para el control de la cámara y el segundo fue C para el control de las pantallas mediante protocolo SPI.

Hay que tener en cuenta que los programas que se ejecutan en la Raspberry no corren cuando se reinicia el Sistema Operativo, para que los programas inicien junto con el Sistema de arranque fue necesario crear un guion de “Bash” (interprete de comandos), donde se especifican los programas que iniciaran al encender el equipo y sus parámetros para configurar los programas.

### 3.3 Ventajas de un Sistema Operativo Linux

Cuando se trabaja con microcontroladores en lenguajes de medio o alto nivel, el lenguaje de programación dependerá en gran medida el tipo de microcontrolador y la empresa que lo fabrique, a pesar de que todos los microcontroladores aceptan el lenguaje C la sintaxis cambia de una empresa de microcontroladores a otra, es por eso que, a pesar de que la estructura básica del lenguaje C permanece igual no se puede utilizar un programa de un “MSP430” de Texas Instruments, en un “PIC” de la empresa Microchip, esto es debido al compilador que usa cada empresa, el cual busca optimizar las instrucciones y aprovechar al máximo las capacidades del microcontrolador, para ello utiliza palabras clave en la estructura del lenguaje C para realizar ciertas instrucciones, las cuales son propias de cada empresa, pero ¿Qué pasaría, si no buscamos aprovechar todo el microcontrolador y sacrificamos un poco de recursos (velocidad de procesamiento), a cambio de unificar todas las variantes de C en un solo “C para microcontroladores”? La respuesta es que se podría ocupar el mismo programa en cualquier microcontrolador sin importar que empresa lo fabrique.

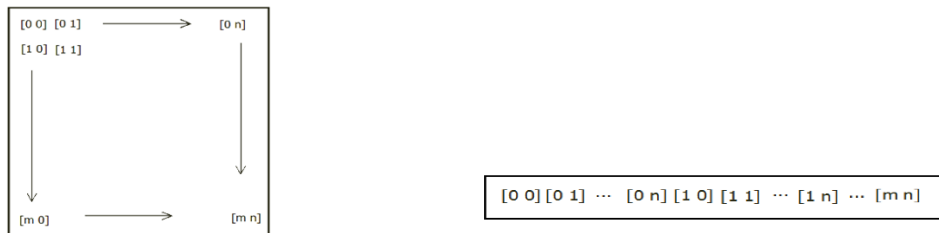
La idea de tener un Sistema Operativo en un equipo de cómputo es en principio la misma idea que el “C para microcontroladores”, todos los programas que se realicen dentro de éste Sistema Operativo pueden ser utilizados en otra computadora sin importar si cambia el procesador o la memoria RAM, siempre y cuando utilice el mismo Sistema Operativo, en este caso los programas que se ejecuten en ambas computadoras entregarán resultados iguales. Si extrapolamos estos planteamientos a los equipos SBC como la Raspberry, se entiende que podremos usar los programas diseñados en la Raspberry versión “Pi B+” con un Sistema Operativo “Raspbian Wheezy” en una versión Raspberry “Pi 3” con un Sistema Operativo “Raspbian Jessie”, sin tener que hacer alguna modificación en los programas, este tipo de migraciones se puede hacer en todos los Sistemas Operativos. Por otro lado, la importancia de tener un Sistema Linux radica en que cuando salieron al mercado, las SBC trabajaban con Software libre GNU/Linux y en la actualidad casi todas las tarjetas tienen una distribución Debian, con lo que se pueden migrar los proyectos de una tarjeta de desarrollo a otra, únicamente haciendo unas pequeñas modificaciones para adecuar el Hardware que cada tarjeta maneja, con esta posibilidad de migración los proyectos no quedan limitados a las plataformas Raspberry, pero si a los sistemas Debian.

## Capítulo 4

### Procesamiento de imágenes

Cualquier imagen digital se puede representar como un arreglo de tres dimensiones, alto, ancho y el valor del PIXEL, por ejemplo, el formato RGB, cada PIXEL está formado por tres componentes Rojo, Verde y Azul. Por otra parte, las pantallas en las que se despliegan las imágenes tales como las pantallas OLED se pueden representar de forma matricial como se muestra en la figura 4.1 a), esta matriz es de dos dimensiones y en cada localidad se encuentra el valor del PIXEL, la diferencia entre la pantalla y la imagen es la forma en que el valor del PIXEL está representado, en las imágenes RGB cada color tiene un valor entero independiente de los otros dos, mientras que en las pantallas los tres colores RGB se representan con un único valor numérico.

Al asignar los elementos de la matriz para desplegar una imagen en una pantalla en una forma individual y ordenada podemos representar la matriz como un arreglo unidimensional, la forma de este arreglo se observa en la figura 4.1 b).



a) Matriz de dos dimensiones

b) Arreglo unidimensional

Figura 4.1 Representaciones de una pantalla OLED

Pero las computadoras no solo pueden codificar imágenes en formato RGB, también pueden utilizar otro tipo de formatos algunos ejemplos de estos son: PNG, JPG y BITMAP, cada formato ordena en archivos de forma diferente la información de la imagen generada, además de agregarle otros datos que son importantes para que el Sistema Operativo haga una correcta generación de la imagen, estos datos adicionales resultan ser inútiles exclusivamente para cuando la imagen es desplegada en un cierto tipo de pantallas además algunos de estos formatos no permiten ser convertidos a un arreglo de arreglos que es la forma en que Python maneja las matrices, para solucionar este problema, en este trabajo primero se convirtió el archivo de la imagen que se quiere desplegar, previamente almacenado en cualquier formato a un formato RGB 888 y posteriormente a un arreglo de números enteros.

Una vez que se convirtió la imagen a un arreglo, se tiene una matriz tridimensional de números enteros con valor entre 0 y 255 esto nos indica que cada componente del PIXEL tiene 8 bits, la pantalla utilizada requiere 5 bits en rojo, 6 bits en verde y 5 bits en azul,

con lo que se logra un formato RGB 565 por lo cual es necesario reducir el tamaño de las componentes de los PIXELES originales.

Al estar trabajando con un lenguaje de alto nivel como Python, las operaciones se pueden realizar en números decimales sin embargo no hay que perder de vista los efectos que ocasionan estas operaciones a nivel de bits, nuestro objetivo es transformar tres números binarios en un arreglo de dos números binarios, en ambos casos la dimensión de los números es de 8 bits en la figura 4.2 podemos observar esta conversión.

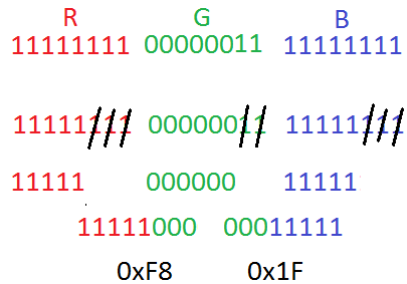


Figura 4.2 Conversión de RGB 888 a RGB 565

En un sistema binario, las operaciones básicas son la suma y multiplicación, la resta y la división son un caso particular de las operaciones básicas. Sin embargo, en los números binarios eliminar el bit menos significativo (LSB), equivale a realizar una división entre dos del número original en formato decimal, esta forma de dividir está representada en la Ecuación 1.

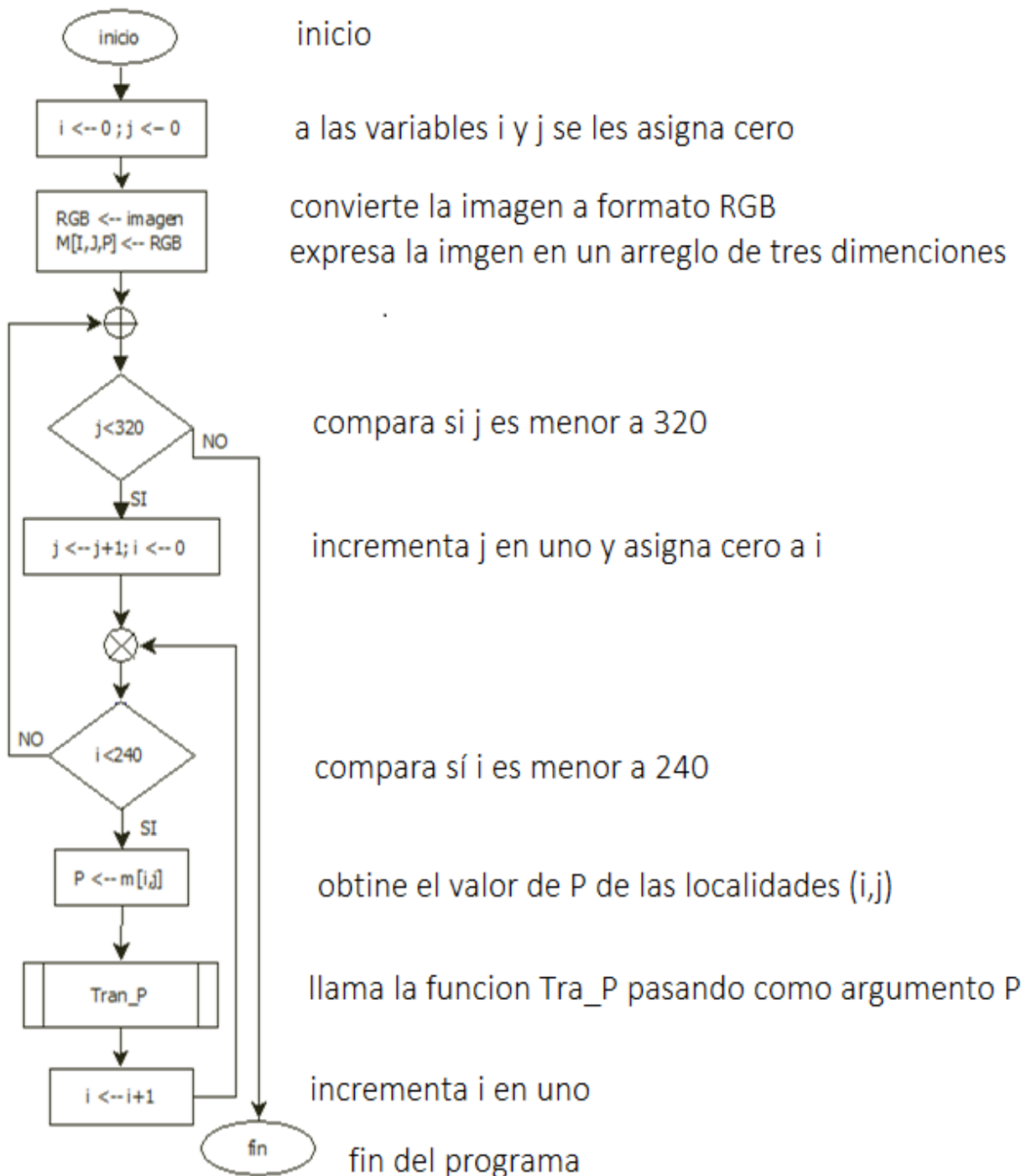
$$n = \frac{N}{2^k} \text{----- (EC 4.1)}$$

donde: "N" es el numero en formato decimal sin dividir  
 "n" es el numero ya dividido  
 "k" es el numero de bit que se desea eliminar

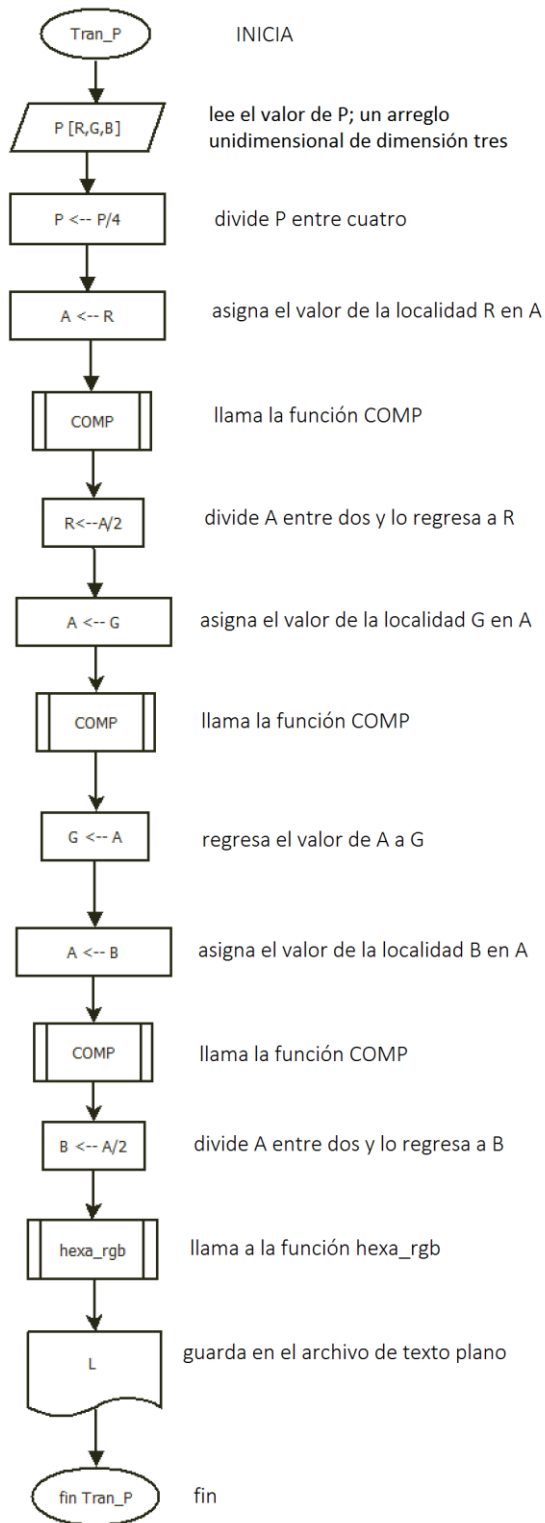
De esta manera se pueden eliminar los bits necesarios, pero se tiene una pérdida de información importante en los tonos de colores que tienen la imagen, dado que en este proyecto, para el uso de la técnica de "pantallas nulas dinámicas con gotas de color" solo se utilizan los colores básicos que son: Rojo, Verde, Azul, Blanco y Negro por tal motivo la perdida de bits no representa un problema significativo ya que se puede comparar cada componente de color y saturarlo o eliminarlo dependiendo del caso y recuperar los colores originales.

A pesar de que las pantallas OLED utilizadas necesitan un valor de 16 bits por PIXEL, y dado que el protocolo de comunicación usado SPI en la Raspberry Pi tiene un "buffer" de 8 bits es necesario dividir en dos partes este valor para poder transmitirlo. Por otro

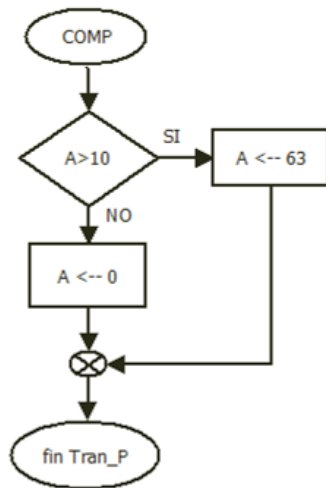
lado, para poder guardar la imagen en su nueva forma se utiliza un archivo de texto plano. El diagrama de flujo donde se representa este algoritmo se observa en la figura 4.3.



a) Programa principal



b) Función Tran\_P

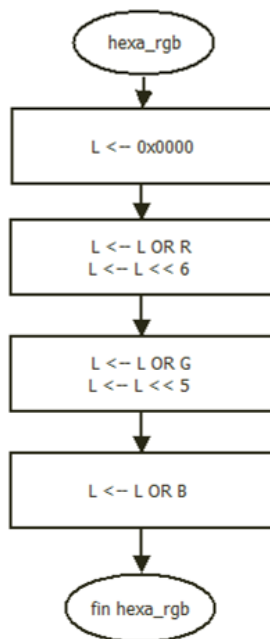


inicio

compara el valor de A con diez  
si es mayor le asigna sesenta y tres  
si es menor le asigna cero

fin

c) Función COMP



inicio

asigna el valor de cero a L

aplica a L operacion OR con R  
recorre L seis localidades a la izquierda

aplica a L operacion OR con G  
recorre L cinco localidades a la izquierda

aplica a L operacion OR con R

fin

d) Función hexa\_rgb

Figura 4.3 Diagrama de flujo conversión de imagen a texto plano



## Capítulo 5

### Manejo de archivos

#### 5.1 Despliegue en pantalla

Las pantallas usadas en este proyecto cuentan con tres formas de comunicarse con el hardware externo estas formas son: interfaz de CPU, SPI y RGB en cualquiera de los tres casos se necesita configurar los parámetros iniciales por medio de la comunicación SPI, para realizar esta comunicación la pantalla cuenta con un “buffer” de 24 bits, al cual es necesario mandarle primero el bit más significativo (MSB) y después el resto de la información, cabe mencionar que el flanco de la señal con la que la pantalla leerá los datos es configurable, por lo que en este caso se seleccionó el flanco de subida para la lectura de señales.

Las pantallas seleccionadas utilizan un sistema de apuntador para guardar los datos en memoria, esto quiere decir que primero se le da la dirección de memoria a la que se desea acceder y posteriormente los datos a guarda, en la figura 5.1 se muestra la forma correcta de estructurar la comunicación, se puede ver en la misma figura que es necesario enviar dos paquetes de 24 bits para configurar un solo parámetro. Como las pantallas utilizan el mismo “buffer” para control y datos, es necesario especificarle si el paquete es una dirección de memoria o un dato, el primer byte de cada transmisión es usado como palabra de control, si este byte de control es un 0x70 (INDEX) el paquete se refiere a una dirección de memoria, pero si es un 0x72 (DATA) La información que le sigue será un paquete de datos el cual se guardará en la localidad de memoria que se le asignó previamente.

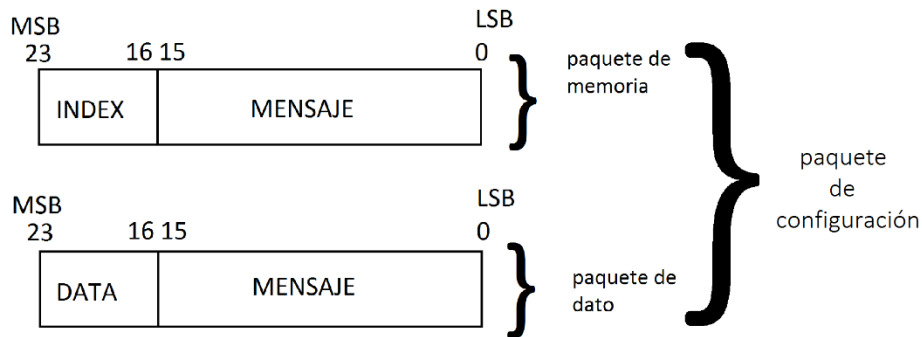


Figura 5.1 estructura del mensaje

El sistema de control de las pantallas seleccionadas adicionalmente, cuentan con una memoria interna donde se puede almacenar una imagen, para su posterior despliegue, también cuenta con un controlador de video que se encarga de refrescar las imágenes

guardadas en la memoria de las pantallas, al momento de guardar en la memoria interna la imagen que se desea desplegar, no es necesario especificarle la dirección donde se guarda cada uno de los pixeles de la imagen, la pantalla empieza a almacenarla en la localidad cero y recorre a la siguiente localidad de forma automática en el momento que guarda el valor, lo que sí es necesario especificarle a las pantallas, es la forma en la que se realizará el barrido de despliegue de la información sobre sus superficies, este barrido puede ser de forma horizontal o vertical, además cuentan con una función de espejo tanto horizontal como vertical para el manipuleo de la imagen, en la figura 5.2 se muestra como ejemplo una imagen de una letra “F”, así como los efectos que causaría usar las funciones de espejo, también se muestra cómo se utiliza el número de “PIXEL” para indicar el inicio y fin del barrido.

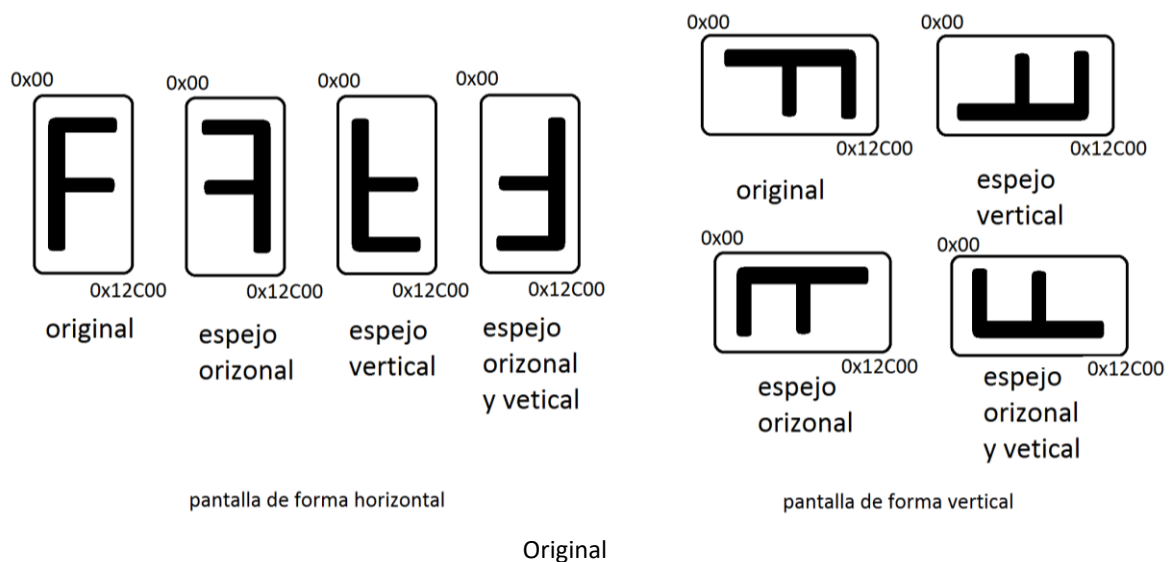


Figura 5.2 Formas de barrido de las pantallas

En la figura 5.3 se muestra la forma en la que se conecta una de las pantallas con el control externo (host) para el manejo de las imágenes, básicamente se tienen tres líneas, MOSI, CLK y CS, en la misma figura se muestran las formas de onda de estas señales, vistas en un osciloscopio en donde el canal D0 es el valor del dato que se está mandando, el canal D1 es el ciclo de reloj y el canal D2 es la activación del chip, el valor enviado, en este ejemplo corresponde a un paquete de memoria con información 0x720F10:

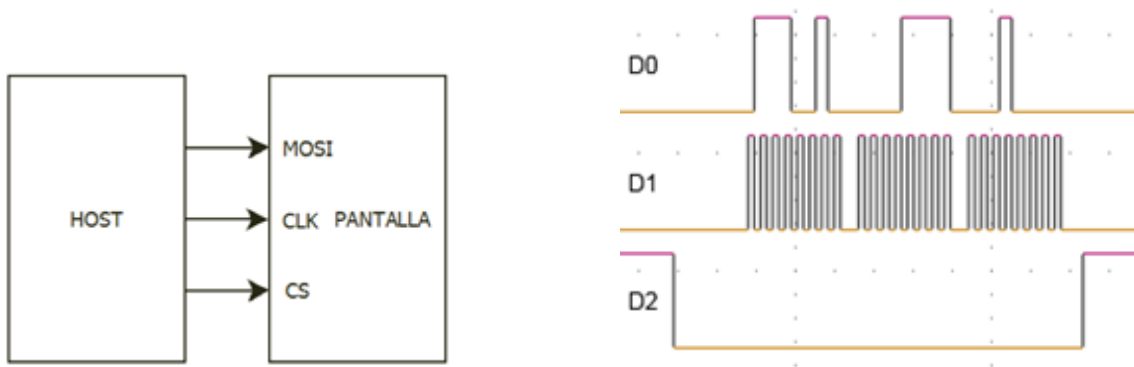
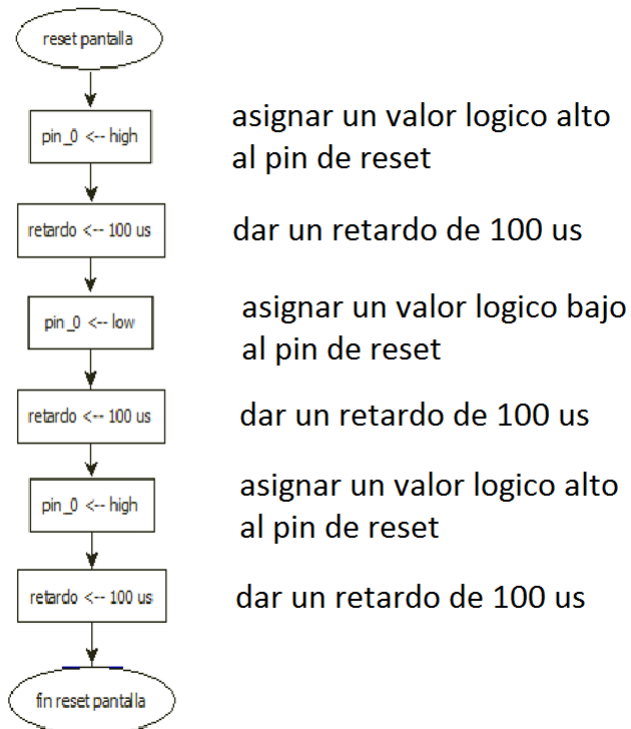
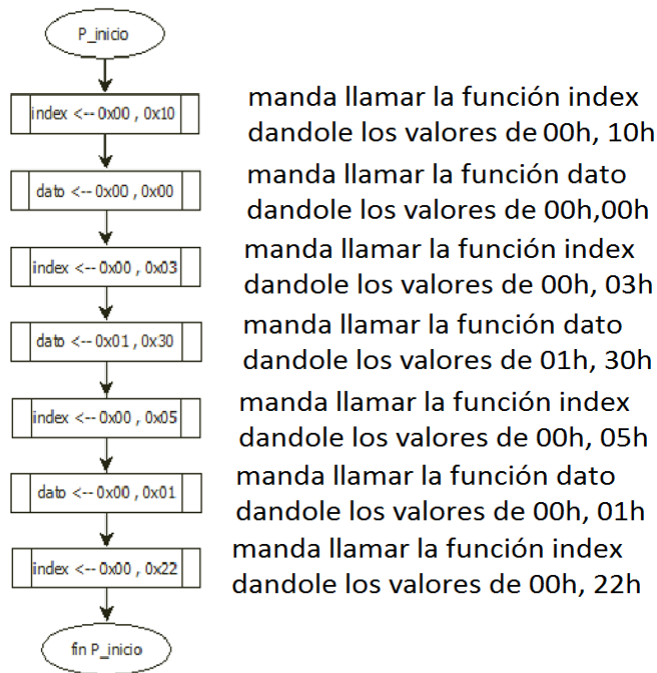


Figura 5.3 comunicación SPI

Para el envío de imágenes a las pantallas, se desarrolló un programa en lenguaje C el cual se encarga de leer el archivo desde donde se tiene guardada la imagen previamente diseñada y lo carga a la memoria de las pantallas, en la figura 5.4 se muestra el diagrama de flujo.

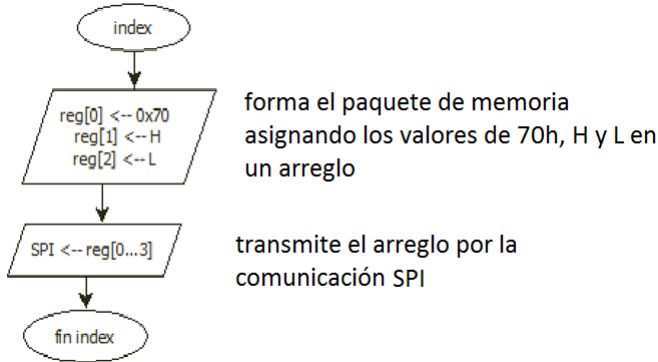


a) Subrutina para reiniciar la pantalla



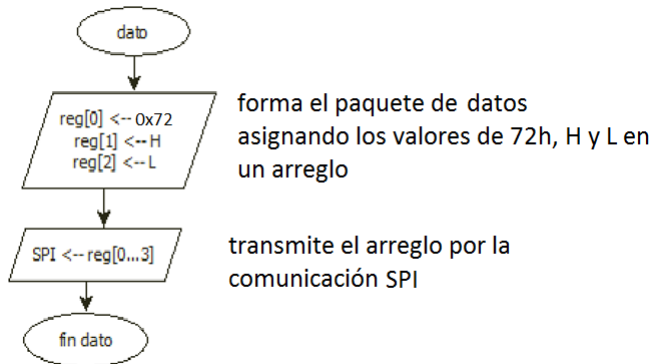
manda llamar la función index dandole los valores de 00h, 10h  
 manda llamar la función dato dandole los valores de 00h,00h  
 manda llamar la función index dandole los valores de 00h, 03h  
 manda llamar la función dato dandole los valores de 01h, 30h  
 manda llamar la función index dandole los valores de 00h, 05h  
 manda llamar la función dato dandole los valores de 00h, 01h  
 manda llamar la función index dandole los valores de 00h, 22h

b) Subrutina para asignar valores de inicio



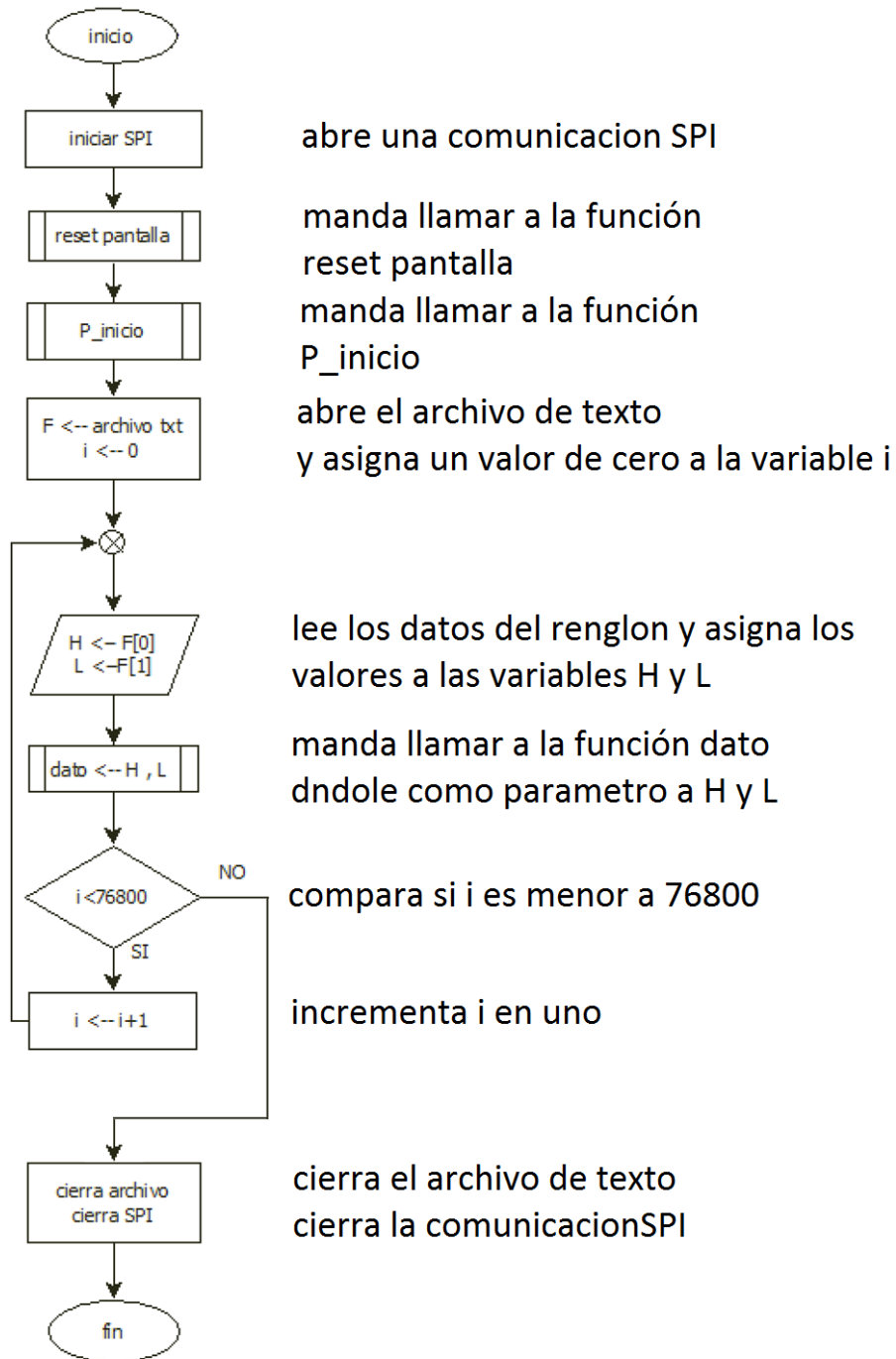
forma el paquete de memoria asignando los valores de 70h, H y L en un arreglo  
 transmite el arreglo por la comunicación SPI

c) Subrutina para formar paquete de memoria



forma el paquete de datos asignando los valores de 72h, H y L en un arreglo  
 transmite el arreglo por la comunicación SPI

d) Subrutina para formar paquete de datos



e) Rutina principal del programa

Figura 5.4 diagrama de flujo control de pantalla

## 5.2 Captura de la foto

La cámara seleccionada, para la captura de la imagen reflejada sobre la superficie en estudio una vez proyectada desde el arreglo de pantallas OLED, es la “Raspberry Pi NoIR Camera V2” la cual es nativa de la tarjeta Raspberry Pi. Para el manejo de tales imágenes capturadas tanto fijas como de video se utilizó el lenguaje Python este lenguaje cuenta con una biblioteca dedicada al manejo de esta cámara, por lo que en este trabajo no se tratará la forma en la que se aplicó este software para este fin, sin embargo, existen otros factores físicos que se tienen que tomar en cuenta a la hora de utilizar cámaras como esta, uno de estos puntos, es que este tipo de cámaras no cuentan con un filtro infrarrojo y por lo mismo, el ángulo de incidencia de la luz resulta ser de suma importancia a la hora de capturar una imagen, la cámara de la Raspberry PI en su versión 2 puede capturar imágenes a plena luz de día y en condiciones de baja intensidad de luz con buenos resultados para la aplicación que se presenta en este trabajo sin embargo en el primer caso, cuando se toman imágenes a plena luz de día la calidad de estas dependerá del ángulo de incidencia, ya que, si la luz incide de forma normal al sensor el procesamiento que realiza la cámara eliminara las componentes infrarrojas pero, por el contrario si la incidencia de luz resulta ser en un ángulo mayor a 15 grados con respecto a la normal las imágenes tendrán un tono rosado ; si este ángulo aumenta, el tono de la imagen subirá de tono hasta llegar a ser rojizo. Al no tenerse acceso al procesamiento interno de imágenes que realiza la cámara para poder modificarlo, se empleó una solución física, como se muestra en la figura 5.5, en esta figura se observa una lente externa a la cámara la cual sirve tanto para enfocar los objetos cercanos como para realizar una incidencia de luz de forma normal al sensor CMOS.

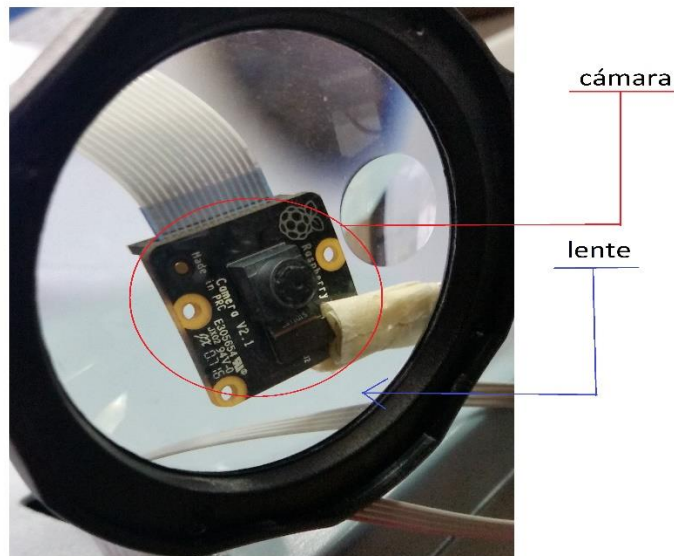


Figura 5.5 Arreglo de lente y cámara

## Capítulo 6

### Sistema de Comunicación Wi-Fi

#### 6.1 IEEE 802.11.b/g/n

El estándar 802.11 del IEEE (Institute of Electrical and Electronics Engineers) es mejor conocido como Wi-Fi, cuenta con velocidades de transmisión entre 5.4 y 600 Mbits/s dependiendo de la actualización que se esté usando, el alcance de la transmisión entre los equipos que utilizan este tipo de comunicación, se encuentra entre 50 y 300 metros y trabaja en la banda libre de 2.4 GHz como se muestra en la figura 6.1. En esta figura se observa un barrido en frecuencia utilizando un analizador de espectro modelo RSA306, marca tektronix, en su pantalla se puede observar un barrido en frecuencia de la transmisión realizada entre el equipo desarrollado y una computadora, el estándar 802.11 mencionado, surge de la necesidad de establecer una comunicación inalámbrica entre dispositivos de distintas marcas, con el estándar mencionado se unificó la transmisión de tal forma que la mayoría de los equipos fueran compatibles.

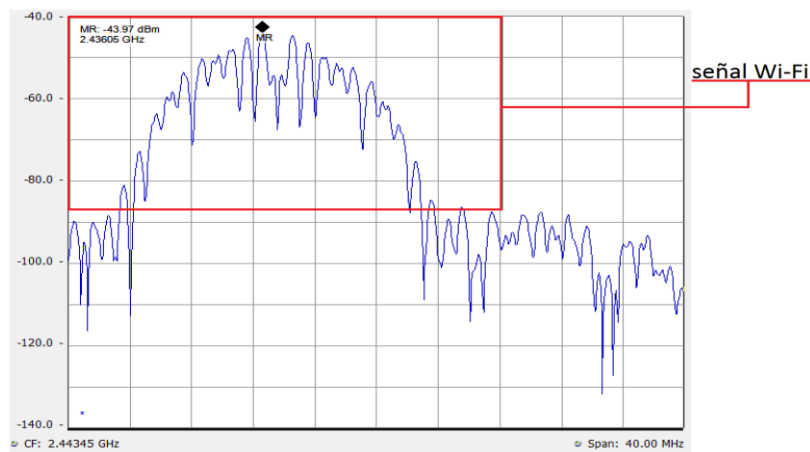


Figura 6.1 Señal Wi-Fi en el dominio de la frecuencia

Resulta ser muy útil contar con algún tipo de comunicación inalámbrica que sea compatible con diferentes dispositivos, por obvias razones la principal ventaja es la no utilización de cables, los cuales limitan el espacio de trabajo e incremental el tamaño de los dispositivos usados al necesitar un puerto de conexión. En la actualidad la conexión Wi-Fi se encuentra incluso en electrodomésticos como refrigeradores, lavadoras de ropa o televisores, entre otros. Por otro lado, y contrariamente a las ventajas, la comunicación Wi-Fi también presenta algunos problemas, tales como la interferencia de la comunicación por ruido, la saturación del espectro electromagnético y especialmente la inseguridad de la red de comunicación debido a su bajo nivel en el cifrado de los datos, es por ello que al ser una red abierta que se encuentra en el aire,

cualquier persona puede tener acceso a esta, a pesar de que el estándar cuenta con un cifrado llamado “WAP2”, este cifrado resulta ser seguro contra la mayoría de los usuarios, pero no necesariamente contra los llamados “hackers” que son personas que cuentan con una elevada experiencia en informática, la cual les permite tener acceso de forma ilegal en este tipo de comunicaciones dejando expuesta la información que se transmite.

En la figura 6.2 se muestra un diagrama de bloques de la forma en la que se realiza la comunicación Wi-Fi en su forma más general, el diagrama cuenta con tres bloques los cuales son el Modem AP el cual tiene dos funciones, una es la de modular y demodular (modem) la información que se transmite a la Internet y la otra es crear un punto de acceso (AP) el cual permite interconectar diferentes equipos electrónicos en una red inalámbrica. El servidor es un programa que atiende las peticiones de otro programa (cliente) y le devuelve una respuesta a tal petición, el servidor se encuentra conectado con el modem mediante una terminal alámbrica ya sea de cable telefónico o de fibra óptica. El cliente es un programa que utiliza a un servidor para realizar alguna acción.

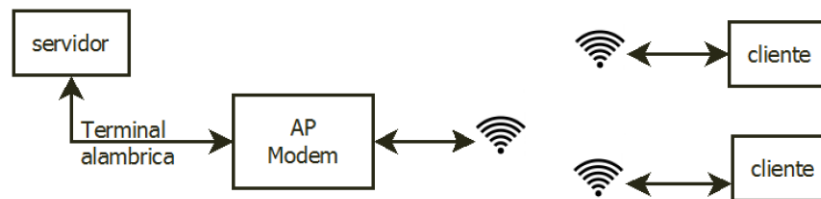


Figura 6.2 Esquema general del estándar 802.11

## 6.2 Comunicación Servidor-Cliente usando Flask

Existen diferentes formas de interconectar dos equipos de cómputo que se encuentren dentro de una red de área local, la forma más sencilla es utilizar una comunicación segura (SSH), para la realización de esta forma de comunicación entre dos computadoras, es necesario tener una conexión como la mostrada en la figura 6.3, en la cual los dos clientes que se quieren comunicar, necesitan estar conectados al mismo punto de acceso (AP), bajo esta topología, pueden compartir información entre ellos y aunque resulta ser la forma más fácil de comunicar una computadora de forma remota y por consecuencia, también de controlarla, es necesario contar con cierta experiencia manejando los comandos básicos de Linux.



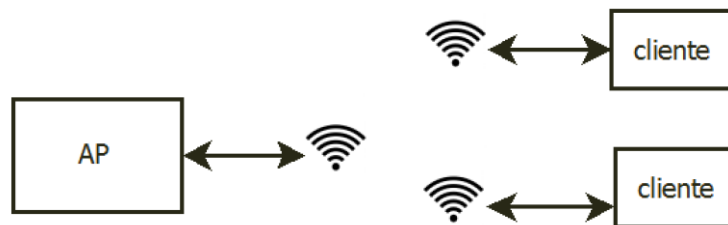


Figura 6.3 Conexión de computadoras usando un punto de acceso

En la actualidad es difícil encontrar personas que siga manejando una computadora mediante una terminal de comandos, por lo general se utiliza un entorno gráfico donde se sustituyen los nemónicos por botones, los cuales facilitan al usuario la utilización del equipo, por este motivo es necesario crear una interfaz gráfica para comunicarse con el equipo, dado que la comunicación por SSH no cuenta con dicha interfaz esta opción deja de ser la más viable para el proyecto aquí presentado.

Por otro lado, la estructura de comunicación que más se asemeja a la anteriormente expuesta. En esta estructura se puede prescindir de la Internet y realizar la comunicación mediante la comunicación Wi-Fi. Bajo esta topología, se tendrá un servidor que atenderá las peticiones de los clientes (equipos conectados a este). En este trabajo finalmente se utilizó este esquema de cliente-servidor, en el cual las peticiones de los clientes son empleadas para el manejo de archivos internos (imágenes y fotos) y para la manipulación de equipos periféricos (cámaras y pantallas) conectados al servidor. En la figura 6.4 se muestra un diagrama a bloques de la topología cliente servidor utilizada en el desarrollo de este proyecto.

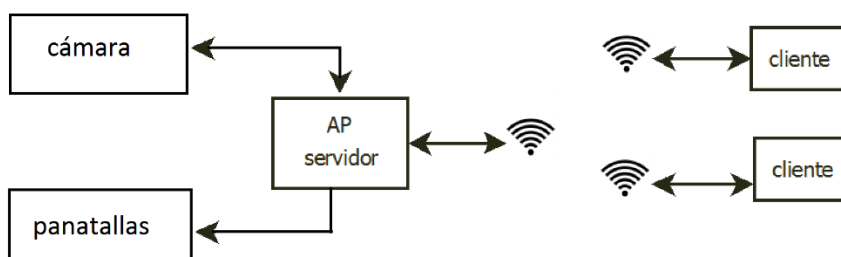


Figura 6.4 Estructura cliente-servidor donde el servidor controla el Hardware

La configuración del servidor, empleado para integrar la red utilizada en el proyecto, se programó en lenguaje de programación conocido como Python, para ello fue necesario utilizar una infraestructura digital (Framework) llamada "Flask" la cual permite crear aplicaciones web utilizando pocas líneas de código, este Framework realiza el control de flujo del servidor, sin embargo, no se pueden realizar interfaces gráficas amigables para el usuario. Dado que Flask es una aplicación Web, tiene la forma de acceder a los diferentes campos del servidor, para ello utiliza un explorador de internet tal como Firefox o Chrome. Es debido a lo anteriormente expuesto que la interfaz gráfica se realizó mediante el uso de páginas Web escritas en lenguaje HTML, en la figura 6.5 se

muestra el menú principal de las opciones programadas en el servidor, también se puede apreciar que en la barra de direcciones se muestra la dirección IP que se utiliza para la conexión, de igual forma se pueden observar las diferentes opciones que se tienen y los botones para acceder a ellas.

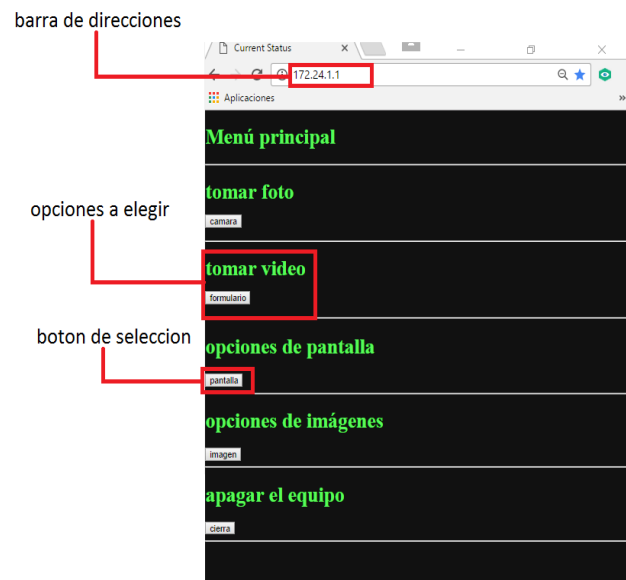


Figura 6.5 Página principal del modem vista en una computadora

Una de las ventajas de ocupar paginar Web para la realización de la interfaz gráfica, es que se tiene la posibilidad de utilizar diversos dispositivos, incluso teléfonos inteligentes, para comunicarse con el servidor y poder ordenar las diversas tareas para las cuales fue programado. Con esta topología de red (cliente-servidor) que cuente con un explorador de internet, se puede atender cualquier dispositivo que este equipado con la tecnología Wi-Fi, en la figura 6.6 se muestra una prueba en un Sistema Operativo Android, donde en la barra de direcciones se pueden apreciar las opciones que fueron seleccionadas en los menús anteriores.



Figura 6.6 Prueba en sistema Operativo Android

En la topología cliente servidor empleada, el bloque correspondiente al servidor-AP, está conformado a su vez por tres bloques principales los cuales son:

- Sistema de control: El cual se encarga de recibir y atender peticiones de los clientes, además de controlar la cámara y las pantallas.

- Interfaz gráfica: Son páginas web que sirven para que el cliente pueda realizar las peticiones al servidor y también despliega los resultados que este entrega la respuesta del servidor puede ser la entrada a otro menú o la descarga de un video o imagen previamente obtenida por el sistema de control.
- AP: Es el punto de acceso para realizar la comunicación Wi-Fi el cual es proporcionado en este trabajo por la Raspberry Pi 3, el Hardware de punto de acceso se encuentra instalado de fábrica.

Un esquema a bloques más detallado del sistema cliente-servidor se muestra en la figura 6.7 donde se aprecia la forma en que opera este sistema.

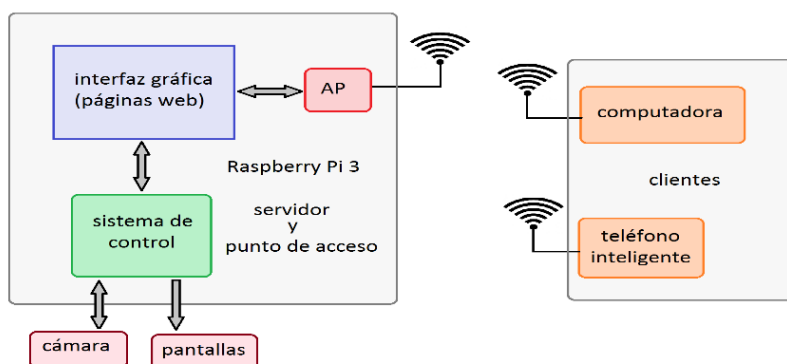


Figura 6.7 Diagrama a bloques del sistema diseñado

### 6.3 Sistema de control del equipo

La función principal del sistema de control diseñado es administrar la información que se recibe del cliente, atiende las peticiones de este, así como el control de las pantallas y la cámara, para realizar estas tareas el sistema de control está compuesto por diferentes programas, dividiéndose en cuatro partes las cuales son: control de cámara, programa servidor, control de pantalla e interfaz gráfica, los lenguajes de programación usados fueron seleccionados dependiendo de la función que realiza esa parte del programa, para las partes que necesitan una respuesta rápida el lenguaje de programación seleccionado fue "C", para las interfaces graficas se utilizó HTML, para el procesamiento de imágenes así como para el control de la cámara y el programa principal que realiza el control del equipo (servidor) se seleccionó el lenguaje Python. En la figura 6.8 se puede observar el diagrama de bloques del sistema de control el cual se encuentra dividido por colores, usados para diferenciar los lenguajes de programación que fueron empleados.

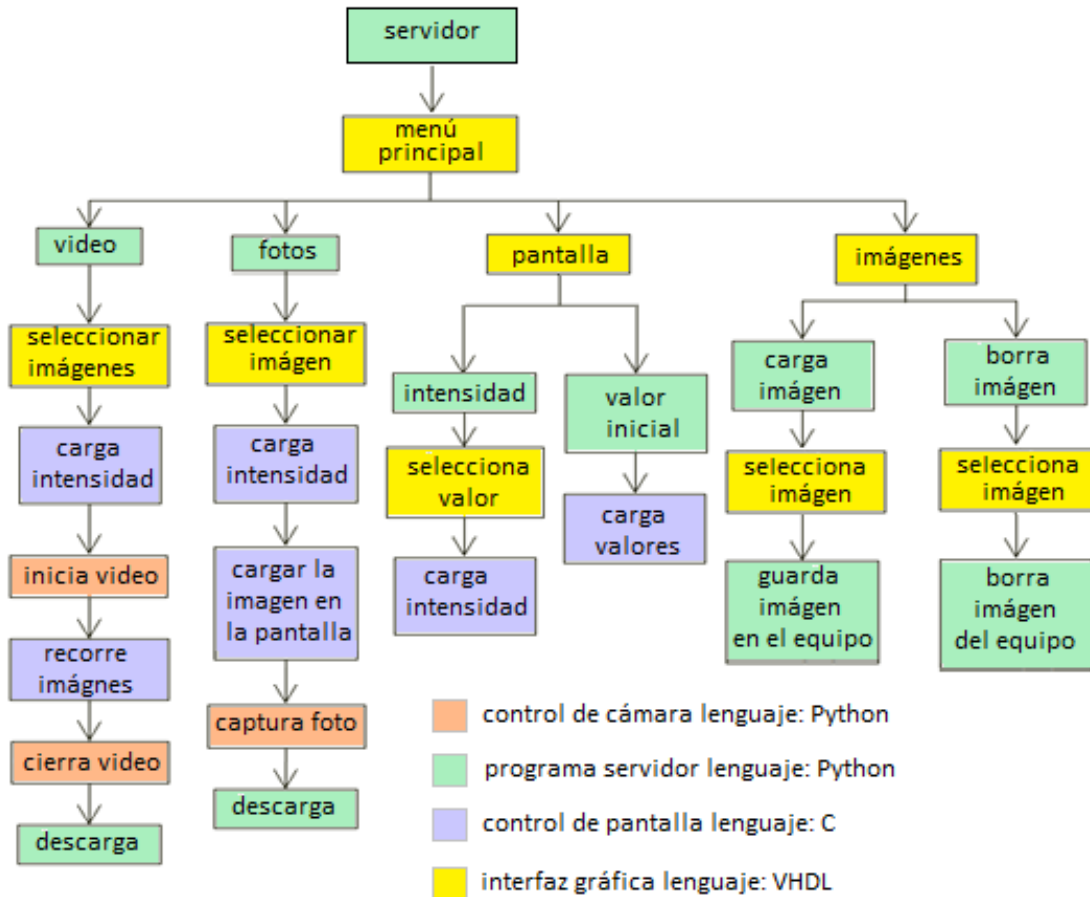


Figura 6.8 Diagrama de bloques del sistema de control

El programa “servidor” es el encargado de atender al cliente que se conecte a la red Wi-Fi, este programa se apoya en los programas complementarios para realizar las acciones necesarias y ofrecer una respuesta a cada petición del cliente.

En este tipo de topologías no podemos hablar de un programa de control, sino de un sistema, esto se refiere a que se tienen pequeños programas los cuales son entrelazados por la interfaz gráfica y el programa “servidor”. Para dejar más clara esta idea, es necesario recordar que los sistemas telefónicos usados en la segunda guerra mundial, donde para realizar una llamada contestaba una persona a la cual se le llamaba operadora y su trabajo consistía en conectar la línea telefónica de la llamada entrante con la línea destino, esta forma de atender llamadas telefónicas es en principio el sistema cliente-servidor, el explorador de internet realiza la función de la operadora tomando de la barra de direcciones el destino a interconectar, la interfaz gráfica es la llamada entrante y los programas complementarios son las líneas destino, en la figura 6.9 se observa la forma de operar del sistema cliente-servidor.

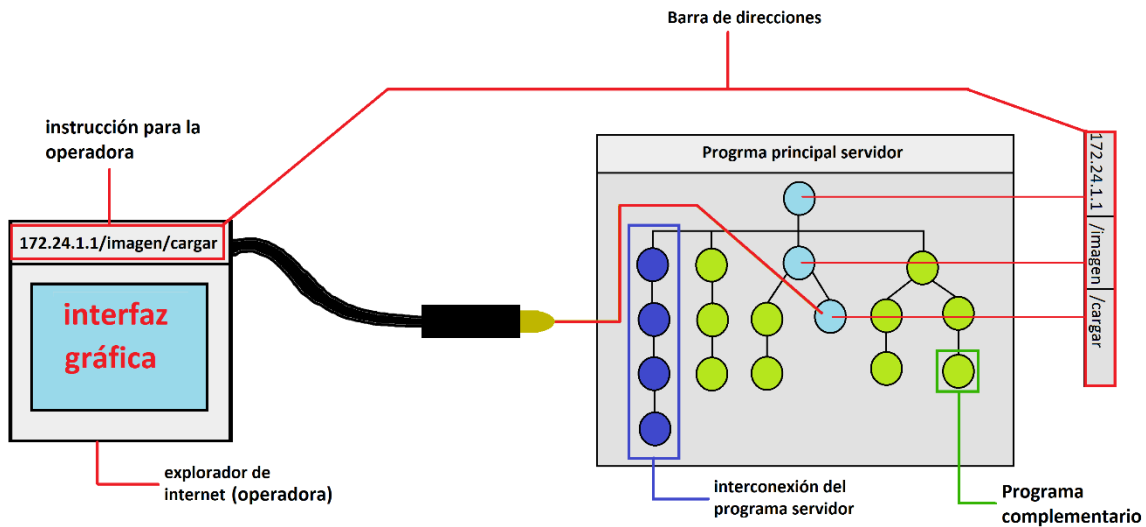


Figura 6.9 Forma de operar del sistema cliente-servidor

El programa “servidor” define la forma en la que se interconectan los programas complementarios como se aprecia en el recuadro azul de la figura 6.8. El cliente tiene la opción de elegir el camino de programas a realizar en la dirección principal (172.24.1.1), algunos de estos programas se ejecutan en segundo plano sin que el cliente lo sepa, otros, modifican la interfaz gráfica y se quedan en espera de recibir un nuevo comando, la principal característica que tiene el sistema cliente-servidor radica en que no importa en qué programa secundario se encuentre la conexión con el explorador de internet, lo que importa es la ruta de la barra de direcciones (la petición del cliente), el servidor puede desconectar el programa y reconectar en cualquier otro programa sin importar si están interconectados en el programa principal, en otras palabras, el servidor siempre le dará prioridad a la petición del cliente y no a la secuencia del programa principal.

En la figura 6.10 se observa mediante un diagrama de bloques las pruebas del sistema descrito anteriormente, en el diagrama se encuentran 4 bloques, el bloque llamado “sistema diseñado” muestra las pantallas y la cámara conectadas a la Raspberry Pi 3 las pantallas se conectan mediante una tarjeta de pruebas mientras que la cámara se conecta directamente, en el bloque “monitor,” se observa el proceso del programa servidor atendiendo a un cliente que en este caso es un teléfono inteligente con sistema operativo Android Lollipop, el cual, se muestra en el bloque “teléfono celular” por último se observa la fuente de alimentación usada para las pruebas.

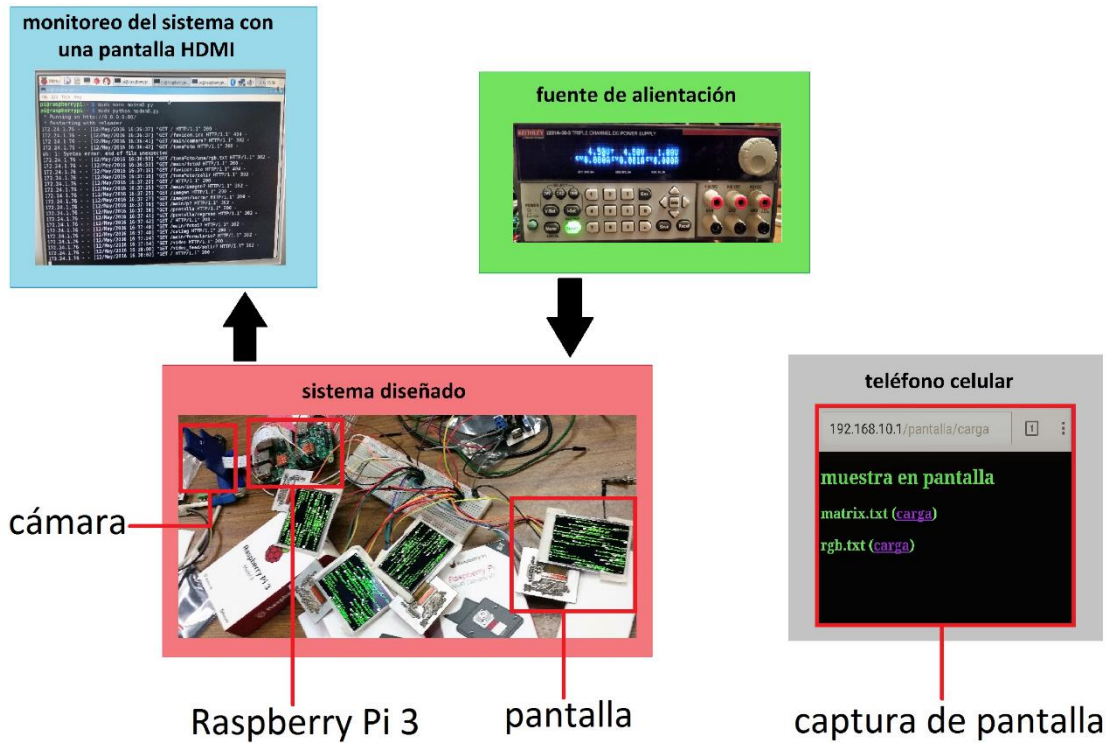


Figura 6.10 Pruebas del sistema diseñado

#### 6.4 Fuente de alimentación

Todo sistema electrónico necesita una fuente de alimentación para poder funcionar, en este trabajo la fuente de alimentación es un convertidor de AC-DC, su función es mantener un cierto nivel de tensión eléctrica en DC a una corriente determinada por la carga, la fuente de alimentación diseñada es una fuente lineal, este tipo de fuentes utiliza un transformador para reducir el voltaje, un rectificador de onda completa, un filtro para el voltaje de rizo y los reguladores de voltaje, el diagrama general se puede observar en la Figura 6.11 Como se puede observar, la fuente entrega diferentes niveles de voltaje, necesarios para alimentar los diversos componentes involucrados en el sistema, tales como pantallas, ventilador del sistema de enfriamiento de la misma fuente así como la Raspberry Pi 3.

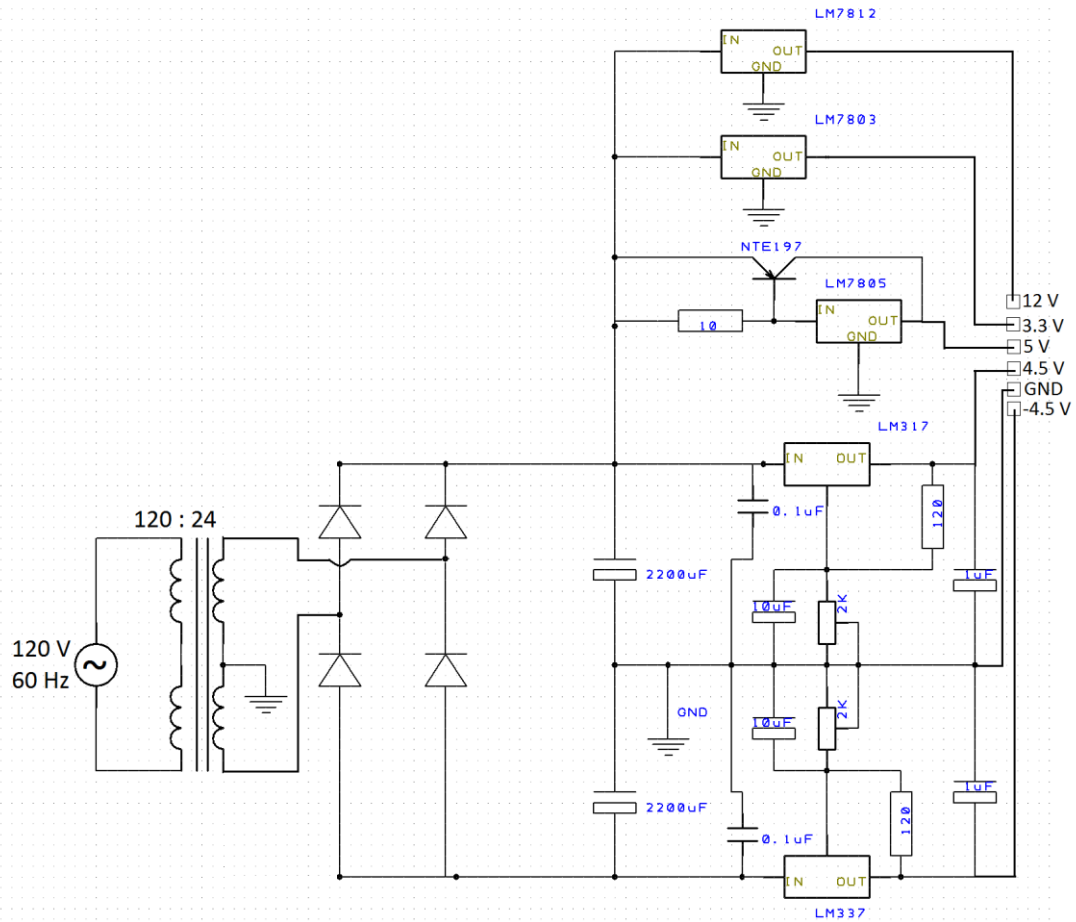


Figura 6.11 diagrama de la fuente de alimentación

Dado que la fuente de alimentación entrega diferentes valores de voltaje a diferentes corrientes es necesario analizar cada regulador por separado, la familia de reguladores LM78XX entregan 1.5 Amper en corriente de salida, además requiere un voltaje de encendido de tres volts por arriba de su voltaje de salida, si la carga demanda más de un Amper es necesario utilizar un circuito de Bypass con un transistor para obtener la ganancia de corriente, este es el caso del regulador de cinco volts que alimenta la Raspberry Pi 3 la cual demanda entre 500 [mA] y 1.5 [A] aunque puede presentar picos de 2 [A] en condiciones de pleno uso, el circuito de Bypass necesita una resistencia entre la base y emisor del transistor la cual se calcula con la fórmula 6.1, donde  $R_{on}$  es el valor de la resistencia de encendido,  $I_{reg}$  la corriente que pasa por el regulador y  $v_{on}$  el voltaje necesario para encender el transistor, para obtener el valor de resistencia necesario fijamos la corriente que queremos que pase por el regulador en 100 [mA] y el voltaje de encendido lo consideramos de 1 [V] tomando en cuenta estas consideraciones la resistencia necesaria es de 10 [ $\Omega$ ].

$$R_{on} = \frac{v_{on}}{I_{reg}} \dots\dots\dots(6.1)$$

$$10 = \frac{1}{100 \times 10^{-3}} \dots\dots\dots(6.2)$$

Por su parte, los demás reguladores no necesitan un circuito complementario que les ayude con la demanda de corriente, por ejemplo, el regulador de doce volts solamente alimenta un ventilador de 200 [mA] y el regulador de tres volts funciona únicamente como un nivel de referencia para la escala de grises de las pantallas y la corriente que se le demanda tiende a ser cero, los otros dos reguladores restantes, están conectados de tal forma que su voltaje de salida es ajustable y alimentan a las pantallas con un voltaje bipolar de 4.5 volts tanto positivos como negativos, tanto el regulador LM317 como el regulador LM337 empleados para estos últimos niveles de voltaje pueden suministrar una corriente máxima de 800[mA] sin modificar su voltaje de salida, las pantallas seleccionadas demandan en conjunto 600 [mA] de corriente máxima, tomando en cuenta estos datos, no es necesario en este caso utilizar un circuito de Bypass con transistores.

Antes de los reguladores se colocaron capacitores de filtrado los cuales se utilizan para reducir la variación de la señal rectificada a un nivel adecuado de tal forma que los reguladores puedan operar de forma apropiada en la figura 6.12 se puede observar una simulación de la de la forma de onda obtenida a la salida del puente rectificador con filtro y sin filtro, en la figura mencionada, se presenta en la parte inferior la forma de onda obtenida sin filtrado y en la parte superior la forma de onda después de ser filtrada.

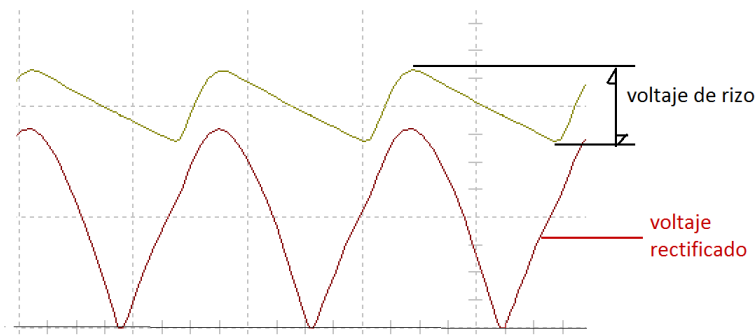


Figura 6.12 voltaje de rizo

Para calcular el valor del capacitor, es necesario definir el valor mínimo que puede tomar el voltaje de rizo, este valor está en función de los valores que necesitan los reguladores para su correcto funcionamiento, como ejemplo se presenta el cálculo del filtro para la fuente de 12V, dado que es el más crítico ya que es el que requiere el mayor voltaje a su entrada tomando en cuenta que el voltaje de dropout de este tipo de reguladores empleados es de 3V entonces, si el voltaje que uno de ellos entrega a su salida es de 12V, por consecuencia el valor mínimo del voltaje de rizo ( $V_{cdFL}$ ) debe de ser 15V para lograr una operación correcta, por otro lado se sabe que la corriente máxima ( $I_{max}$ ) demandada al transformador por todos los componentes conectados es de 2.8 [A], para calcular el valor del capacitor se utilizó la fórmula 6.3, cabe



mencionar que el valor de capacitancia esta dado en micro Faradios y el % de rizo se tomó del 10%.

$$C = \frac{I_{max}}{200[3.5(\frac{\%de\ rizo * V_{cdFL}}{100})]} \dots\dots\dots (6.3)$$

$$2666 = \frac{2.8}{200[3.5(\frac{10*15}{100})]} \dots\dots\dots (6.4)$$

Con base en el cálculo anterior se seleccionó el capacitor comercial más aproximado, el cual es de 2200[uF], el transformador utilizado necesita entregar tanto la corriente como el voltaje calculado anteriormente, recordemos que se necesita el doble del valor de  $V_{max}$  ya que este valor es tomado desde la derivación central, la corriente máxima calculada es de 2.8 [A] y el valor de la línea domestico es de 120  $V_{rms}$ , por lo tanto es necesario un transformador reductor de 120 a 24 [ $V_{rms}$ ] que entregue una corriente de 3 [A] como mínimo.

En la figura 6.13 se observa la fuente diseñada



Figura 6.13 Fuente diseñada

## Resultados

El equipo desarrollado se muestra en la figura R.1, en dicha figura se puede observar una prueba del equipo, la prueba consistía en mandar una imagen a las pantallas OLED.

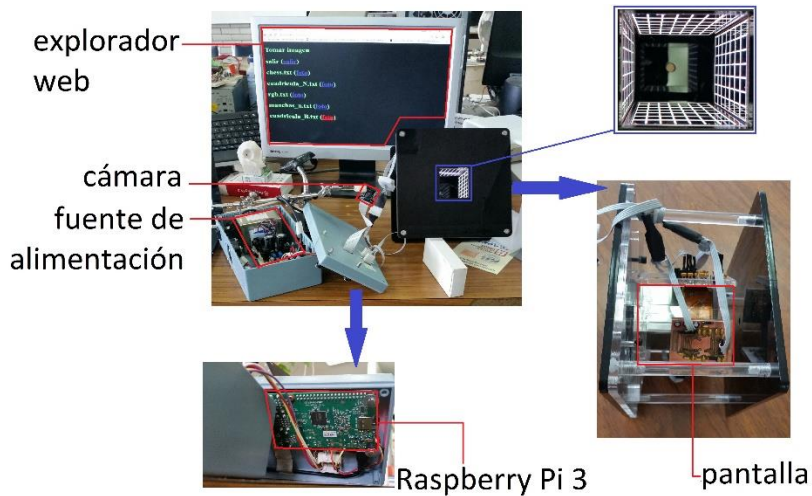
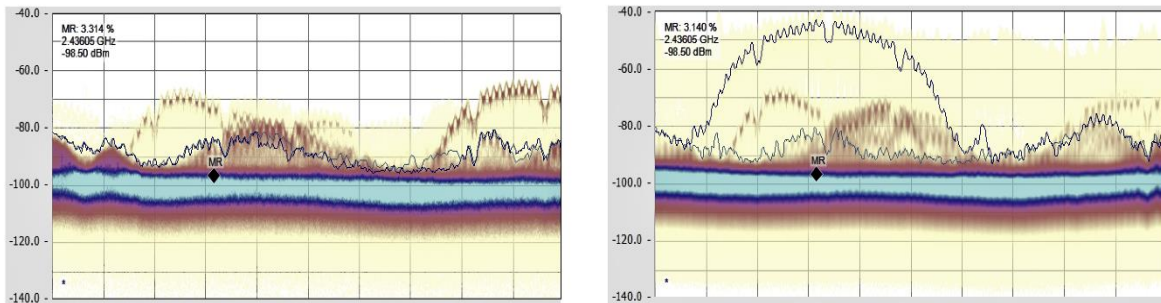


Figura R.1 prueba de equipo

Utilizando un analizador de espectro RSA306 se observó la trasmisión del equipo desarrollado en la figura R2, se puede apreciar en la imagen a) el ruido en el ambiente, mientras que en la imagen b) se observa la comunicación entre la Raspberry Pi 3 y una computadora.



a) Señal de ruido en el ambiente

b) transmisión Wi-Fi

Figura R.2 Análisis de espectro en la banda libre de 2.4 GHz.

## Conclusiones y trabajo a futuro

Las pruebas realizadas en el laboratorio mostraron que se redujo de forma considerable las zonas oscuras indeseables en la superficie de prueba, en la figura C.1 a) se muestra una imagen tomada con el equipo de pantallas LCD mientras que en la figura C.1 b) una imagen tomada con el equipo diseñado con pantallas OLED, en ambos casos se enmarco en color rojo las zonas de color oscuro donde no existe imagen reflejada o presenta una pérdida de dicha imagen, como se puede apreciar en la comparación, se redujo de forma significativa este tipo de zonas.

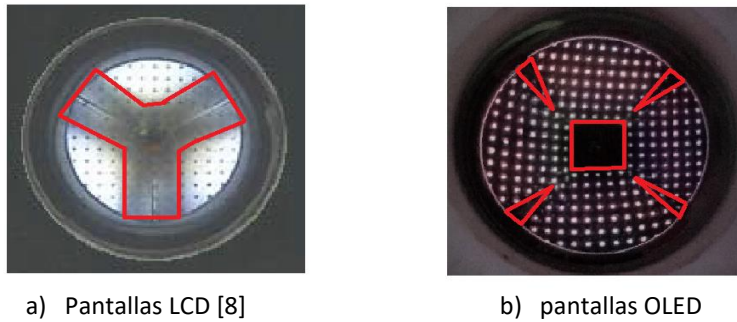


Figura C.1 comparación de imágenes obtenidas con pantallas LCD y OLED

Por otra parte, y a pesar de que se redujeron las zonas oscuras aún se tiene una gran parte al centro de la imagen que causa una pérdida de información significativa por este motivo en un trabajo futuro se reducirá esta zona mediante el uso de un arreglo geométrico diferente de pantallas OLED. Adicionalmente, se podría también aumentar 30 veces la velocidad de carga de las imágenes en las pantallas con la finalidad de reproducir video y de esta forma poder realizar pruebas dinámicas de la superficie en estudio.

Por otro lado, la comunicación inalámbrica resulto ser exitosa como se muestra en la figura C.2 donde se puede apreciar en la imagen a) la señal Wi-Fi instantánea en una descarga de archivos mientras que, en la imagen b), se muestra el historial de la comunicación en el tiempo.

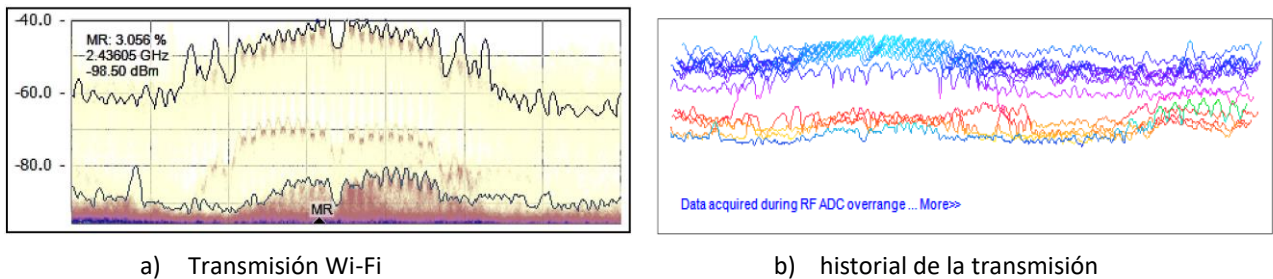


Figura C.2 señal de Wi-Fi del equipo en plena transmisión

Tomando en cuenta los dos puntos anteriores se concluye que el equipo diseñado cumplió con los objetivos propuestos en el presente trabajo.

## Apéndice A

### Protocolo de comunicación SPI (Serial Peripheral Interface)

El protocolo SPI es una interconexión de 4 hilos, bidireccional (Full dúplex) capaz de enviar y recibir información al mismo tiempo, síncrona y ocupa el esquema de maestro esclavo, los 4 hilos son "SCLK" en el cual se transmite el ciclo de reloj, "MOSI" la línea por la cual el maestro le envía información al esclavo, "MISO" en esta línea el esclavo envía información al maestro y "CS" la línea de habilitación, en la figura A1 se aprecia un esquema de esta conexión.

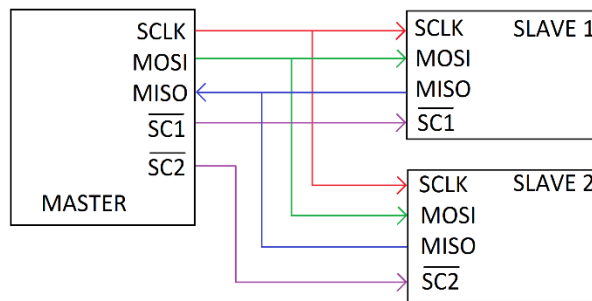


Figura A1 conexión de los módulos SPI

Para iniciar la comunicación es necesario mandar la línea  $\overline{SC}$  a un valor lógico bajo, después el maestro mediante un ciclo de reloj le manda o recibe la información del esclavo, cada ciclo de reloj representa 1 bit en la transmisión, generalmente los MCU ocupan buffer de 8 bits, aunque no es una regla y dependerá del esclavo el tamaño de la transmisión a realizar en la figura A2 se muestra un ejemplo de la transmisión descrita anteriormente.

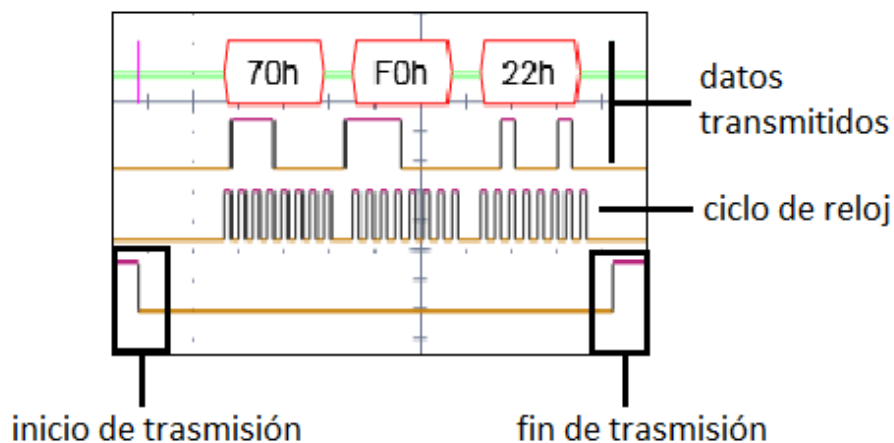


Figura A2 ejemplo de las señales SPI

En este tipo de comunicaciones se puede leer los datos en dos momentos flanco de subida o flanco de bajada este es un parámetro de configuración de la transmisión, en la figura A3 se muestran las dos formas de leer datos.

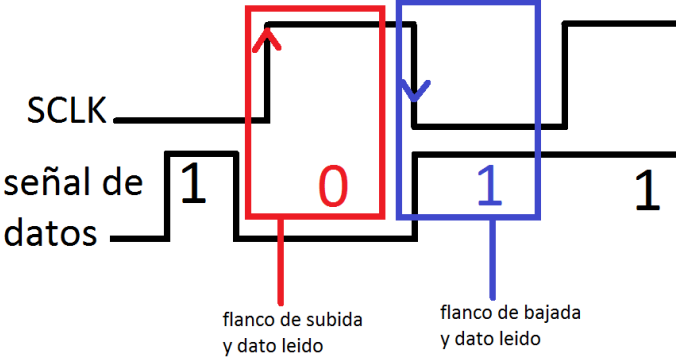


Figura A3 flancos de la señal de reloj

En la comunicación SPI se puede enviar la cadena de bits de dos formas, iniciando primero con el bit más significativo (MSB) o el caso contrario, iniciando con el bit menos significativo (LSB) este también es un parámetro de configuración. En la figura A4 se muestra la forma en que envían los datos de las dos maneras; el dato está representado en forma hexadecimal "0x79" y en binario "01111001".

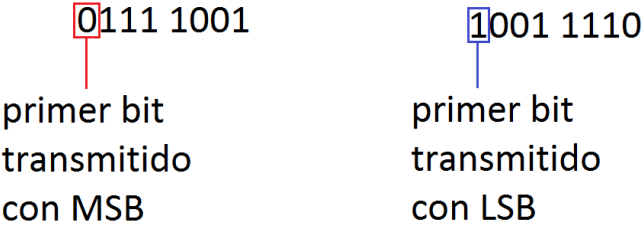


Figura A4 ejemplos de inicio de bits en transmisión

Nótese que el bit inicial sirve para indicar donde inicia la transmisión, como se muestra en la figura A4 el orden de del arreglo se invierte de un modo al otro.

## Bibliografía

Orden alfabético autores apellido nombre año publicación titulo país editorial  
Autor nombre disponible en www...

- [1] Cairó Osvaldo, 2005, Metodología de la programación, México, Alfaomega
- [2] Coughlin Robert, Driscoll Frederick, 1999, Amplificadores operacionales y circuitos integrados lineales, México, Prentice Hall
- [3] Huidobro José, Luque Javier, 2014, Comunicaciones por Radio. Tecnologías, Redes y Servicios de Radiocomunicaciones. El espectro electromagnético, México, Alfaomega
- [4] Mansoor Syed, Koretsky Robert, 2003, El libro de Linux, España, PEARSON
- [5] Punch William, Enbody Richard, The Practice of computing using Python, USA, PEARSON
- [6] Raspberry Pi HQ, How-To: Turn a Raspberry Pi into a WiFi router disponible en <http://raspberrypi.com/how-to-turn-a-raspberry-pi-into-a-wifi-router/>
- [7] Rodríguez Roberto, Sossa Juan, 2012, Procesamiento y Análisis Digital de Imágenes, México, Alfaomega
- [8] Rodríguez Martin, et al, 2015, Dynamic point shifting with null screens using three LCDs as targets for corneal topography, applied optics
- [9] Tojeiro German, 2016, Raspberry Pi 2 para electrónicos, Barcelona, España, Alfaomega
- [10] Uploading Files, disponible en <http://flask.pocoo.org/docs/0.11/patterns/fileuploads/>
- [11] Welcom to Flask, disponible en <http://flask.pocoo.org/docs/0.11/>
- [12] 2006, data sheet S6E63D6, SAMSUNG
- [13] 2007, Application Note for am2.2 2.4 2.8 20070926, CHI MEI EL CORPORATION