

CALIBRADOR DE COLOR PARA MONITORES LCD

CARLOS ERNESTO OMAÑA

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA, DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

BOGOTÁ

2012

I

CALIBRADOR DE COLOR PARA MONITORES LCD

CARLOS ERNESTO OMAÑA

Trabajo de grado para optar al título de
Ingeniería Electrónica

Director:
Camilo Otálora.

Ingeniero Electrónico, M.Sc.

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA, DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

BOGOTÁ

2012

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA

RECTOR MAGNÍFICO: R P. JOAQUÍN EMILIO SANCHEZ GARCÍA, S.J.

DECANO ACADÉMICO: ING. LUIS DAVID PRIETO, PhD.

DECANO DEL MEDIO UNIVERSITARIO: SERGIO BERNAL RESTREPO

DIRECTOR DE CARRERA: ING. JAIRO HURTADO LONDOÑO, PhD.

DIRECTOR DEL PROYECTO: ING. CAMILO OTÁLORA, M.Sc.

ARTÍCULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia”.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
2. MARCO TEÓRICO	2
2.1. PRINCIPIOS DEL COLOR	2
2.1.1. Colores Cromáticos	2
2.1.2. Tono	2
2.1.3. Valor.....	2
2.1.4. Intensidad	2
2.2. HISTORIA RGB	2
2.3. MODELO RGB PARA PROCESAMIENTO DE COLOR EN PANTALLAS ELECTRÓNICAS.....	3
2.4. MICROCONTROLADOR.....	5
2.4.1. Recursos básicos del Microcontrolador	6
2.4.2. Recursos especiales.....	9
2.4.3. Herramientas para desarrollo de aplicaciones	11
2.4.4. PIC24FJ64GB002	11
2.4.5. ICD3.....	12
2.5. SENSOR TCS3414	13
2.5.1. Descripción.....	13
2.5.2. Diagrama Funcional de Bloques	13
2.5.3. Principios de Operación	14
2.5.4. Responsividad Espectral.....	16
2.6. PROTOCOLOS	17
2.6.1. I2C	17
2.6.2. USB.....	19
3. CRITERIOS DE DISEÑO	23
3.1. CONEXIONES PIC24FJ64GB002.....	23
3.2. CONEXIONES TCS3414.....	23
3.3. ESQUEMÁTICO.....	24
4. DESARROLLOS.....	25
4.2. COMUNICACIÓN I2C.....	25
4.2. COMUNICACIÓN USB	28
5. PROTOCOLO DE PRUEBAS	30
5.1. PREPARACIÓN.....	30
5.1.1. Aislamiento del Sensor.....	30
5.1.2. Calibración del dispositivo	30
5.1.3. Tipos de pruebas	31
5.2. EJECUCIÓN.....	33
5.2.1. Ejecución de Prueba de aislamiento del sensor	33

5.2.2. Ejecución de Prueba de sin intensidad lumínica	33
5.2.3. Ejecución de Prueba del círculo cromático.....	34
5.2.4. Ejecución de Prueba sobre monitor LCD	34
5.3. ANÁLISIS DE RESULTADOS	35
5.3.1. Análisis prueba de aislamiento del sensor	35
5.3.2. Análisis prueba sin intensidad lumínica	35
5.3.3. Análisis Prueba del círculo cromático	36
5.3.3. Análisis Prueba sobre monitor LCD	40
6. CONCLUSIONES	41
7. BIBLIOGRAFÍA	43
8. ANEXOS	44

TABLA DE FIGURAS

FIGURA 1. PANTALLA LCD.....	4
FIGURA 2. REPRESENTACIÓN ESPACIAL DEL MODELO DE COLOR RGB. CUBO TRICROMÁTICO [4].....	4
FIGURA 3. REPRESENTACIÓN DE COLORES PRIMARIOS Y SECUNDARIOS	5
FIGURA 4. ARQUITECTURA HARVARD [5].....	7
FIGURA 5. PIC24FJ64GB002.....	12
FIGURA 6. ICD3	13
FIGURA 7. SENSOR TCS3414	13
FIGURA 8. DIAGRAMA FUNCIONAL DE BLOQUES DEL SENSOR TCS3414. [7].....	14
FIGURA 9. RESPONSABILIDAD ESPECTRAL[7].	16
FIGURA 10. SECUENCIA DE TRANSFERENCIA DEL BUS DE DATOS [8].	18
FIGURA 11. SÍMBOLO USB[7]	19
FIGURA 12. CONECTOR USB, TIPO A [7].	20
FIGURA 13. CONEXIONES MÍNIMAS PIC24FJ64GB002 [6].....	23
FIGURA 14. CONEXIÓN FÍSICA TCS3414 [7]	24
FIGURA 15. ESQUEMÁTICO FINAL.	24
FIGURA 16. PROTOCOLO I2C READ [7].	25
FIGURA 17. PROTOCOLO I2C, PRIMERA PARTE.	25
FIGURA 18. PROTOCOLO I2C, SEGUNDA PARTE	26
FIGURA 19. INICIALIZACIÓN I2C.....	26
FIGURA 20. MÁQUINA DE ESTADOS.	27
FIGURA 21. MÁQUINA DE ESTADOS DESCRITA EN CÓDIGO.....	27
FIGURA 22. INTERFAZ DE USUARIO.....	28
FIGURA 23. FUNCIÓN QUE ENVÍA DATOS AL PUERTO COM	29
FIGURA 24. COMUNICACIÓN USB	29
FIGURA 25. AISLAMIENTO DEL SENSOR.	30
FIGURA 26. EJEMPLO PRUEBA DE AISLAMIENTO DEL SENSOR.	31
FIGURA 27. PRUEBA SIN INTENSIDAD DE LUZ SOBRE EL SENSOR.	32
FIGURA 28. INSTRUCCIÓN PARA PRUEBAS.....	33
FIGURA 29. RESULTADO PRUEBA AISLAMIENTO DEL SENSOR.	33
FIGURA 30. RESULTADO PRUEBA SIN INTENSIDAD LUMÍNICA.....	34
FIGURA 31. INTERFAZ ADOBE PHOTOSHOP CS5.	36
FIGURA 32. ÁREA ABERTURA AISLAMIENTO DEL SENSOR	42

1. INTRODUCCIÓN

Calcular o conocer el valor de las componentes RGB proyectadas por una pantalla LCD es un proyecto que propone nuevos procesos de calidad sobre un producto. Quizás como saber la madurez de un alimento que se pueda determinar por su color, qué tanto está expuesto un aviso publicitario a la polución o medio ambiente (observando el deterioramiento del color del aviso), reproducción legal de prototipos gráficos de diferentes tipos de marcas (certificando el color original del prototipo gráfico), verificación del color real en colores, marcadores o pinturas, en calidad y durabilidad de las telas, cuando son sometidas al medio ambiente, a procesos de lavado, observar que tanto cambia su color. Uno de los objetivos del proyecto es saber exactamente cuánto cambió el color. Para poder realizar con éxito este tipo de mediciones es necesario tener mediciones muy exactas, mediciones que el ser humano no podría captar visualmente, aquí es donde se concluye que es necesario realizar un dispositivo que pueda captar estas variaciones o este tipo de mediciones. Siguiendo este objetivo es importante seleccionar un sensor que proporcione una resolución alta y que al transportar la frecuencia, voltaje o datos al microprocesador, no se pierda información de la señal emitida por el sensor.

Este proyecto busca diseñar un prototipo funcional de un sistema para poder medir las componentes RGB de la pantalla de un monitor LCD. Para la realización del proyecto es importante definir qué sensores y microprocesadores ofrece el mercado para brindar una solución óptima al problema. Es necesario analizar qué tipo de protocolo es necesario implementar y qué algoritmos pueden ser aplicados para brindar una medición más exacta.

Para descomponer un color en términos que tengan relación con algún medio electrónico es necesario analizar cada aspecto del mismo. Con esto se encuentra que cada color emite una frecuencia diferente, lo cual sería la relación más directa con algún fenómeno electrónico. Pero adaptándose a los sensores del mercado y para la aplicación del proyecto, las pantallas LCD emiten luz, la cual es proyectada sobre el sensor que en este caso es un fotodiodo. Este fotodiodo recibe la intensidad de luz emitida por la pantalla y la transforma, proporcionalmente lineal, en corriente. Según el tipo de sensor a utilizar convierte esta corriente en frecuencia, voltaje, o en una salida digital. Al seleccionar el microprocesador es importante seleccionarlo según el tipo de comunicación que utilice, que sea alimentado por USB y que transporte los datos de forma serial. Esto con el fin de poder analizar, ajustar y visualizar los datos en un computador.

OBJETIVO GENERAL:

Diseñar un prototipo funcional de un sistema para medir las componentes RGB de la pantalla de un monitor LCD.

OBJETIVOS ESPECÍFICOS:

- Diseñar un dispositivo que sea alimentado por USB, con una salida digital serial de 8 bits que representa cada componente RGB.
- Desarrollar un protocolo de pruebas para verificar las mediciones del dispositivo.

2. MARCO TEÓRICO

2.1. Principios del color

La percepción del color está asociada con la luz y con el modo en que ésta se refleja. La percepción del color cambia cuando se modifica una fuente luminosa, o cuando la superficie que refleja la luz está manchada o revestida de un pigmento diferente [1].

2.1.1. Colores Cromáticos

La idea común del color se refiere a los colores cromáticos, relacionados con el espectro que puede observarse en el arco iris. Los colores neutros no forman parte de esta categoría y pueden denominarse colores acromáticos. Todo color cromático puede describirse de tres modos.

2.1.2. Tono

El tono, que es el atributo que permite clasificar los colores como rojo, amarillo, azul, etc. La descripción de un tono será más precisa si se identifica la verdadera inclinación del color de un todo al siguiente [1]. Por ejemplo, un determinado rojo puede ser denominado, con más precisión, rojo anaranjado. Los diferentes sistemas de color utilizan códigos diferentes para describir los colores, recurriendo a letras, números o a una combinación de ambas cosas (terminología descrita en la sección 2.3).

2.1.3. Valor

El valor se refiere al grado de claridad o de oscuridad de un color. Un color de tono conocido puede describirse, más precisamente clasificándolo de claro u oscuro. Por ejemplo, se dice de un rojo que es claro cuando es más claro que nuestra idea de un rojo estándar [1].

2.1.4. Intensidad

La intensidad indica la pureza de un color. Los colores de fuerte intensidad son los más brillantes y vivos que pueden obtenerse. Los colores de intensidad débil son apagados y contienen una alta proporción de gris [1].

2.2. Historia RGB

Las siglas RGB (Red, Green, Blue) más que hacer referencia a los colores rojo, verde y azul, son la composición del color en términos de la intensidad de los colores primarios, esto es denominado como *Modelo del color RGB*. Este modelo permite representar un color mediante una mezcla de los tres colores primarios. Un medidor de estas componentes tiene un campo de aplicabilidad muy grande, especialmente en la industria, empleándose como control objetivo de calidad de colores para productos que se puedan evaluar por su color. En 1996, *Hewlett Packard* y *Microsoft* propusieron un soporte para un espacio de color estándar sRGB [2], esto para productos *HP* y sistemas operativos de *Microsoft*. El objetivo era complementar las estrategias de gestión de color, permitiendo un nuevo método de manejo de color en los sistemas operativos. Dos años después salía al mercado otro soporte llamado adobe RGB, este espacio de color RGB fue desarrollado por

Adobe Systems en 1998, este tuvo como objetivo reproducir de una forma más completa el espacio de color CMYK (acrónimo de Cyan, Magenta, Yellow y Key), utilizado en impresión, pero este desde un espacio de color RGB usado en monitores o en cámaras digitales. Este soporte mejoró la gama de color del espacio de color del sRGB, proporcionando cerca del 50% de los colores visibles especificados en el espacio de color Lab (CIE LAB y Hunter Lab). Más adelante la ISO (International Organization for Standardization) normalizó y reglamentó este problema con la norma ISO 15076-1:2005, la que especifica un formato de perfil de color y que describe la arquitectura dentro de la cual puede operar, con el fin de apoyar el intercambio de información que especifica el color de la imagen de proceso previsto de datos digitales, norma que ha sido actualizada como ISO 15076-1:2010[3]. Esto resalta la importancia de realizar mediciones sobre un color, cuestionar la calidad del mismo para obtener los resultados esperados en el momento de utilizarlos.

2.3. Modelo RGB para procesamiento de color en pantallas electrónicas

La interfaz con la que interactúa un ser humano en la mayoría de dispositivos electrónicos, para entender o percibir determinados eventos físicos, es el *display* (pantalla de visualización de dispositivos electrónicos). La sensación de color emitida por esta interfaz se consigue por la mezcla aditiva de los colores RGB en puntos minúsculos denominados píxeles.

La unidad de información más pequeña que puede procesar un computador es un *bit*. El píxel, es la unidad más pequeña que el hardware y el software de una pantalla puede manipular. Para tener una definición acertada del término píxel, hacemos referencias al libro “*Principios de color y holopintura*” de Antonio Valero Muñoz quien entrega la siguiente definición: “El término píxel deriva del acrónimo fonético inglés *picture element* (elemento de imagen), es la unidad más pequeña homogénea de color que forma parte de una imagen digital, ya sea fotografía, un fotograma de vídeo o un gráfico. El plural es píxeles.”[4]. Para poder almacenar la información de una imagen digital, el color de un píxel se representa mediante la cantidad de bits de información necesarios, denominada profundidad de color (bpp = bits por píxel).

Una profundidad de bit de n conlleva a que cada píxel de una imagen puede tener 2^n posibles valores distintos. En las imágenes de mapa de bits o en dispositivos con un enfoque de funcionalidad gráfica se suelen utilizar 3 bytes (24 bits), los que generarían $2^{24} = 16777216$ variaciones de color. Este tipo de imágenes se denomina *true color*.

Los píxeles se utilizan como unidad de resolución de una pantalla, de una imagen y de algunos dispositivos. Cada punto de la pantalla es un píxel compuesto por un conjunto de tres subpíxeles; rojo, verde y azul (Figura 1), cada uno de los cuales brilla con determinada intensidad.

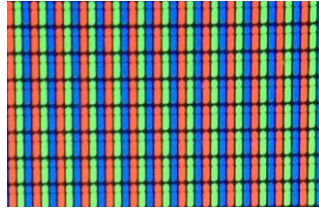


Figura 1. Pantalla LCD

El cubo cromático es un modelo de color el cual está configurado en el espacio RGB representado en la Figura 2. Cada coordenada o coeficiente cromático RGB tiene un rango $[0,1]$, en multimedia y procesamiento de imágenes está más extendida la especificación en cantidades presentes en el intervalo $[0,255]$.

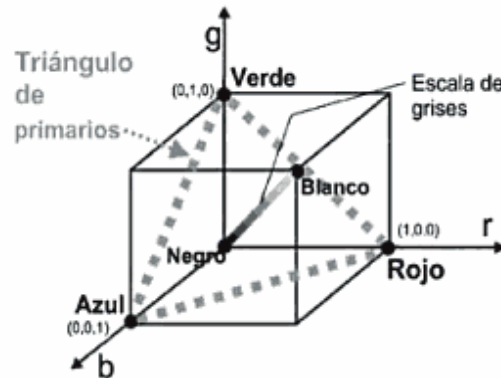


Figura 2. Representación espacial del modelo de color RGB. Cubo tricromático[4].

Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1), es frecuente que cada color primario se codifique con un byte (8 bits). La intensidad o proporción de cada una de las componentes de los primarios se mide según una escala que va del 0 a 255, donde el 0 indica un mínimo de intensidad y el 255 indica una intensidad máxima.

En el cubo RGB, el color rojo se obtiene con las siguiente coordenadas $(255,0,0)$, el verde con $(0,255,0)$ y el azul con $(0,0,255)$, en cada caso un color monocromático. La ausencia de color (lo que se conoce comúnmente como el color negro) se obtiene cuando todas las componentes son 0 $(0,0,0)$. La combinación de dos colores a nivel 255 con un tercio en nivel 0, genera tres colores intermedios (colores secundarios). Los colores generados son el amarillo $(255,255,0)$, el cian $(0,255,255)$ y el magenta $(255,0,255)$. La combinación de los tres colores primarios en su máximo nivel, genera el color blanco $(255,255,255)$.

El conjunto de todos los colores se puede representar en forma de cubo. Cada color es un punto de la en el espacio. El negro es generado por la superficie de la pantalla LCD donde se produce la síntesis aditiva de los primarios. Generalmente en los archivos gráficos, el sistema RGB se usa para

representar cada píxel con un triplete de la forma (R,G,B). Las imágenes en el modelo de color RGB están formadas por tres planos de imágenes independientes, cada uno correspondiente a un color primario. Cuando son introducidas en un monitor RGB, las tres imágenes se combinan en la pantalla para producir una imagen de color compuesta. En la separación, en canales RGB, de una imagen digital, es posible observar por medio de histogramas cómo el canal R posee valores altos, cuando la imagen en color original es bastante rojiza. Los histogramas de los canales R, G y B, con sus valores numéricos, nos informan del porcentaje de uso de cada valor numérico R, G y B. El empleo del modelo RGB para el procesamiento de imágenes es de gran utilidad cuando estas vienen expresadas en términos de los tres planos de colores. Alternativamente, la mayoría de las cámaras en color que se usan para adquirir imágenes digitales utilizan el formato RGB, lo que hace más interesante este formato. La figura 3 muestra cómo los colores primarios: rojo, verde y azul, pueden ser combinados para producir cuatro colores adicionales (amarillo, cian, magenta y blanco).



Figura 3. Representación de colores primarios y secundarios

2.4. Microcontrolador

Se denomina “controlador” al dispositivo a cargo del gobierno de uno o varios procesos. Aunque el concepto de controlador ha permanecido sin cambios a través del tiempo, su implementación física ha cambiado en distintas ocasiones. Años atrás, los controladores eran construidos exclusivamente con componentes de lógica discreta, el siguiente paso fue emplear los microprocesadores, los cuales eran circuitos impresos con chips de memoria y dispositivos de E/S. Actualmente todos los elementos de un controlador se han podido incluir en un chip, el cual se denomina microcontrolador.

Un microcontrolador es un circuito de alta escala de integración que incorpora en su gran mayoría parte de los elementos que configuran un controlador. A continuación se listan los componentes que contiene un microcontrolador:

- Procesador o CPU (Unidad Central de Procesamiento).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Módulos para control de periféricos (temporizadores, puertos serie y paralelo, ADC o conversores análogo/digital, DAC o conversores digital/análogo).

- Generador de pulsos de reloj que sincroniza el funcionamiento de todo el sistema.

Los productos que incorporan un microcontrolador para su regulación disponen de las siguientes ventajas:

- Prestaciones: mayor control sobre un elemento representa una mejora considerable en el mismo.
- Fiabilidad: Al utilizar un microcontrolador por un número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Tamaño: La integración de un microcontrolador en un chip disminuye el volumen y la mano de obra.
- Flexibilidad: Las características de control están programadas, por lo que su modificación sólo necesita cambios en el programa de instrucciones.

Como resumen o síntesis, el microcontrolador es un circuito integrado que incluye todos los componentes de un computador. Gracias a su pequeño tamaño, es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de “controlador empotrado” (Embedded controller).

Al estar integrados todos los “*microcontroladores*” en un solo chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales: procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. En las siguientes secciones se realiza un recorrido de cada característica del microcontrolador PIC24FJ64GB002 del fabricante Microchip.

2.4.1. Recursos básicos del Microcontrolador

2.4.1.1. Arquitectura Básica

Inicialmente todos los microcontroladores adoptaban la arquitectura clásica de Von Neumann. Esta arquitectura se caracteriza por disponer de una sola memoria principal, en esta memoria se almacenan datos e instrucciones de forma diferente. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

Actualmente, los microcontroladores presentan la arquitectura Harvard, esta arquitectura dispone de dos memorias independientes, una para las instrucciones y otra para sólo datos. Estas memorias disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones, de lectura o escritura, simultáneamente en ambas memorias (Figura 4).

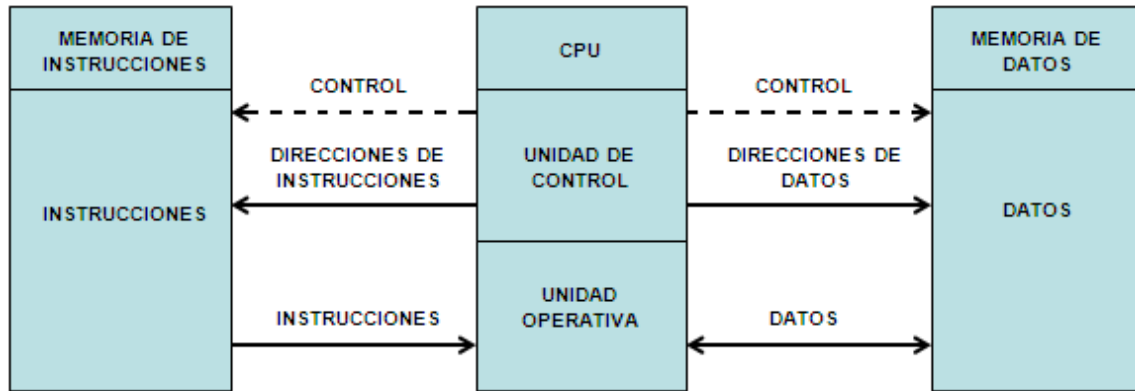


Figura 4. Arquitectura Harvard [5].

La arquitectura Harvard contiene dos memorias independientes, una para sólo instrucciones y otra para sólo datos. Cada una de estas memorias dispone de sistemas de buses de acceso y es posible realizar operaciones de acceso simultáneamente. Los microcontroladores PIC corresponden a la arquitectura Harvard mostrada en la Figura 4.

2.4.1.2. El procesador o CPU

Es la parte más importante del microcontrolador y determina sus principales características, a nivel de hardware y software. Es el encargado de direccionar la memoria de instrucciones, recibir el código de la instrucción, su decodificación y la ejecución de la operación que implica la instrucción, también se puede encargar del almacenamiento de resultados. Para los procesadores actuales existen tres tipos de orientaciones en cuanto a la arquitectura y funcionalidad.

- **CISC:** Su abreviación significa “*Computadores de Juego de Instrucciones Complejo*”, estos disponen de más de 80 instrucciones máquina en su repertorio, instrucciones sofisticadas y potentes, requiriendo gran número de ciclos para su ejecución. Presenta la ventaja ofrecer al programador instrucciones complejas que funcionan como macros.
- **RISC:** Su abreviación significa “*Computadores de Juego de Instrucciones Reducido*”. La industria de los computadores comerciales y la de los microcontroladores está migrando hacia la filosofía RISC. Para estos procesadores el repertorio de instrucciones máquina es muy reducido, con instrucciones más simples que se ejecutan en un ciclo. La sencillez y velocidad de las instrucciones permiten optimizar el hardware y el software del procesador.
- **SISC:** Su abreviación significa “*Computadores de Juego de Instrucciones Específico*”. Presentan un juego de instrucciones muy reducido y específico. Las instrucciones se adaptan a las necesidades de la aplicación a diseñar o implementar.

2.4.1.3. Memoria

La memoria de instrucciones y datos se encuentra integrada en el microcontrolador. Una parte es no volátil, tipo ROM, y se encarga del programa de instrucciones que gobierna la aplicación. La otra parte de memoria, tipo RAM, es volátil y es destinada para guardar las variables y los datos. En este

punto vale la pena mostrar dos diferencias importantes entre los microcontroladores y los computadores personales.

- No existen sistemas de almacenamiento masivo.
- Sólo hay que almacenar un único programa de trabajo, ya que el microcontrolador solo se destina a una tarea en la memoria ROM.

Para los microcontroladores la RAM es de poca capacidad, esta solo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Al existir únicamente un programa activo, no se requiere guardar una copia del mismo en la RAM, este se ejecuta directamente desde la ROM. Los computadores personales están destinados a manejar mega-bytes de memoria, pero los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8k bytes y de RAM comprendidas entre 20 y más bytes.

Existen cinco versiones de memorias no volátiles en el mercado. Es importante comprender cada una, ya que según el tipo de memoria ROM que contenga el microcontrolador, podría limitar la aplicación a diseñar.

- **ROM con Máscara:** Es una memoria no volátil de solo lectura, su contenido se graba durante la fabricación del chip.

- **OTP:** Su abreviación significa (One Time Programmable). Es una memoria volátil de solo lectura “*programable una sola vez*” por el usuario. Para este caso, es el usuario el encargado de escribir el programa en el chip mediante un programa que se maneja desde un PC. Esta versión es recomendable cuando es muy corto el ciclo de diseño del producto. Para este tipo de memoria se suele usar la encriptación mediante fusibles para proteger el código contenido.

- **EPROM:** Su abreviación hace referencia a “*Erasable Programmable Read Only Memory*”, en pocas palabras, puede borrarse y grabarse muchas veces. La grabación de estas memorias se realiza con un grabador gobernado desde un PC. Si se desea borrar el contenido, estas memorias tienen una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos.

- **EEPROM:** son memorias de sólo lectura, programables y borrables eléctricamente “*Electrical Erasable Programmable Read Only*”. La programación y el borrado, se realizan eléctricamente desde el grabador y bajo el control programado de un computador personal. Su programación y borrado es muy fácil, una vez instalados en un circuito pueden grabarse y borrarse una gran cantidad de veces sin ser retirados de dicho circuito. Para esto se usan “*grabadores en circuito*”, los cuales brindan una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo. El número de veces que se programa el microcontrolador es finito.

- **FLASH:** Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. A diferencia de la ROM esta memoria es programable en el circuito. La memoria de tipo FLASH está recomendada frente a la EEPROM cuando se necesita gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

2.4.1.4. Puertas de entrada y salida

La principal utilidad de los pines que posee el microcontrolador es soportar las líneas de Entrada/Salida (I/O) que comunican el computador interno con los periféricos externos. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas I/O se destinan a proporcionar el soporte a la señales de entrada, salida y control.

2.4.1.5. Reloj Principal

Todos los microcontroladores poseen un circuito oscilador que genera una onda cuadrada de alta frecuencia, la cual configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Por lo general, el circuito de reloj está incorporado en el microcontrolador y la frecuencia se activa con componentes externos. Estos componentes externos suelen utilizarse con un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red RC.

2.4.2. Recursos especiales

Los fabricantes ofrecen numerosas versiones de microcontroladores. En ocasiones amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. Lo importante es encontrar el modelo mínimo que satisfaga todos los requerimientos de la aplicación a implementar. Esto con el fin de minimizar costos, hardware y software. A continuación se muestra una breve explicación de los recursos especiales más comunes.

2.4.2.1. Temporizadores (Timers) o Contadores

Son empleados para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

2.4.2.2. Watchdog

Un microcontrolador funciona sin el control de un supervisor y de forma continua las 24 horas del día. El Watchdog consiste en un temporizador que provoca un reset automáticamente en el sistema.

2.4.2.3. Brownout

Resetea el microcontrolador cuando el voltaje de alimentación es inferior al voltaje mínimo requerido (Brownout). Mientras el voltaje de alimentación sea inferior al de Brownout el dispositivo se mantiene reseteado.

2.4.2.4. Estado de reposo o bajo consumo

Son muchas las ocasiones en que el microcontrolador debe esperar a que se produzca algún acontecimiento externo que lo ponga en funcionamiento. Para ahorrar energía, los microcontroladores disponen de una instrucción especial (SLEEP), estado de reposo o de bajo consumo, donde los requerimientos de potencia son mínimos.

2.4.2.5. Conversor de análogo a digital (A/D)

Los microcontroladores que contienen un conversor A/D (Analógico/Digital) pueden procesar señales analógicas de gran abundancia en la mayoría de aplicaciones. Estos disponen de un multiplexor que permite aplicar a la entrada diversas señales analógicas desde los pines del circuito integrado.

2.4.2.6. Conversor de digital a análogo (D/A)

Se encarga de transformar los datos digitales obtenidos del procesamiento de un dispositivo (computador, sensor, etc) en su correspondiente señal analógica, a la cual se puede acceder por los pines de salida del microcontrolador.

2.4.2.7. Comparador analógico

Los microcontroladores que disponen de este comparador, internamente tienen un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines de entrada. La salida del comparador proporciona un nivel lógico 1 ó 0 según la misma comparación.

2.4.2.8. Modulador de anchura de pulsos o PWM

Es un circuito que proporciona en su salida pulsos de anchura variable, a los cuales se puede acceder por los pines de salida.

2.4.2.9. Puertas de Entrada/Salida (I/O) Digitales

Los microcontroladores soportan entradas y salidas digitales, las cuales pueden ser necesarias para operar información de un sensor con salida de datos digitales o enviarlos a un computador para después ser manipulados.

2.4.2.10. Puertas de comunicación

Estas puertas tienen el objetivo de dotar al microcontrolador con posibilidad de comunicarse con otros dispositivos externos, otros buses de microcontroladores, buses de sistemas, buses de redes, esto por medio de protocolos específicos, a continuación se listan algunos de los más comunes:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie asíncrona y síncrona.
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (*Universal Serial Bus*) bus serial para PC.
- Bus I²C, que es una interfaz serial de dos hilos desarrollado por Philips
- CAN (*Controller Area Network*) permite la adaptación con redes de conexasión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

2.4.3. Herramientas para desarrollo de aplicaciones

Otro factor importante para la selección de un microcontrolador es el soporte tanto de Hardware como Software que dispone. Las principales herramientas a tener en cuenta en la elección, son la disponibilidad de compiladores, depuradores, simuladores y hardware con el uso de emuladores y grabadores.

2.4.3.1. Desarrollo del Software

2.4.3.1.1. Ensamblador: Este tipo de programación puede resultar difícil para un principiante, pero permite desarrollar programas muy eficientes. Este tipo de lenguaje entrega al programador el dominio absoluto del sistema. Por lo general los fabricantes suelen entregar el programa ensamblador de forma gratuita.

2.4.3.1.2. Compilador: Es programación en alto nivel (como el C o Basic) el cual permite disminuir el tiempo de desarrollo de un producto.

2.4.3.2. Depuración

Ya que los microcontroladores van a controlar dispositivos físicos, se necesitan herramientas que permitan comprobar el buen funcionamiento del dispositivo.

2.4.3.2.1. Simulador: son capaces de ejecutar en un computador personal programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideal para la depuración del mismo.

2.4.3.2.2. Placas de Evaluación: son pequeñas placas o sistemas con el microcontrolador montado, estas se conectan al PC y se realizan todo tipo de pruebas que pueda ejecutar el microcontrolador. Estas placas generalmente están acompañadas de visualizadores LCD, LEDs, teclados, esto para visualizar el comportamiento del microcontrolador al ser programado.

2.4.4. PIC24FJ64GB002

La función del microcontrolador en este proyecto es recibir los datos entregados por el sensor y comunicarlos al computador utilizando el puerto USB. Para realizar la elección del microcontrolador a utilizar en el proyecto se tuvo en cuenta que cumpliera las siguientes especificaciones:

- Comunicación digital de periféricos: 1-I²C
- Arquitectura: 16 bits
- USB
- Voltaje de Polarización: 0 V a 5 V

Las características requeridas para el proyecto se encuentran en gran cantidad de microcontroladores, por lo tanto existe gran oferta de componentes para la realización del dispositivo. En este caso, se realizó la elección entre los microcontroladores de la familia de PIC24FJ del fabricante Microchip®. Como se muestra en la Tabla 1, este componente PIC24FJ64GB002 cumple las especificaciones listadas anteriormente.

PIC24FJ Device	Pins	Program Memory (Bytes)	SRAM (Bytes)	Remappable Peripherals						I ² C™	10-Bit A/D (ch)	Comparators	PMP/PSP	RTCC	CTMU	USB OTG
				Remappable Pins	Timers 16-Bit	Capture Input	Compare/PWM Output	UART w/ IrDA®	SPI							
32GB002	28	32K	8K	15	5	5	5	2	2	2	9	3	Y	Y	Y	Y
64GB002	28	64K	8K	15	5	5	5	2	2	2	9	3	Y	Y	Y	Y
32GB004	44	32K	8K	25	5	5	5	2	2	2	13	3	Y	Y	Y	Y
64GB004	44	64K	8K	25	5	5	5	2	2	2	13	3	Y	Y	Y	Y

Tabla 1. Familia PIC24FJ de Microchip [6].

Las características de los componentes de esta familia son muy similares, se ha seleccionado el 64GB002 (Figura 5) por la cantidad de pines, con lo cual se logra disminuir el tamaño del circuito impreso a realizar. Otra característica importante para su elección fue la memoria de instrucciones, a diferencia del 32BG002, que presenta la misma cantidad de pines, la memoria del 64GB002 permite mayor almacenamiento.

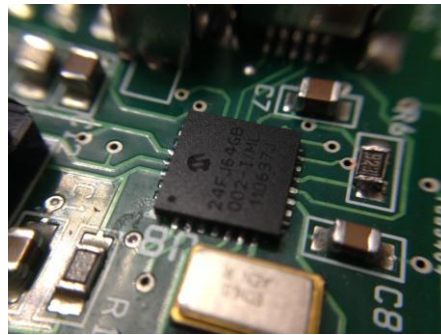


Figura 5. PIC24FJ64GB002

2.4.5. ICD3

El ICD3 (Figura 6) es un programador y depurador en tiempo real para microcontroladores. Permite programar microcontroladores con el IDE de Microchop por ICSP (In-Circuit Serial Programming) y revisar paso a paso el código realizado, esto con la finalidad de encontrar errores con mayor facilidad.

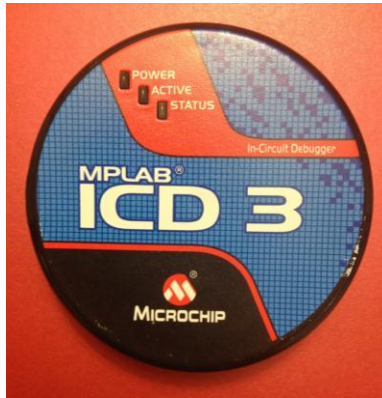


Figura 6. ICD3

2.5. Sensor TCS3414

2.5.1. Descripción

El sensor seleccionado para la implementación del dispositivo es el TCS3414 (Figura 7), fabricado por TAOS (Texas Advanced Optoelectronic Solutions). Este sensor convierte la intensidad de luz en una salida de 16 bits. Este tiene filtros internos que eliminan la fluctuación de la señal debido al parpadeo que produce la iluminación, este parpadeo se convierte en una componente AC que puede alterar el valor a medir. Tiene comunicación directa a un microcontrolador por medio del protocolo I²C y se comunica de forma serial. Posee cuatro conversores Análogo-Digital para cada color, rojo, verde, azul y *clear*, los cuales están conectados a cuatro registros que manejan la información de cada color. Este sensor se polariza entre 2.7 V y 3.6 V. Tiene un canal SYNC para poder realizar mediciones a fuentes de iluminación pulsadas.

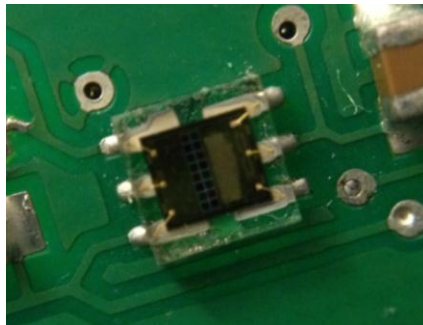


Figura 7. Sensor TCS3414

2.5.2. Diagrama Funcional de Bloques

El sensor recibe las emisiones de luz y filtra cada componente con los fotodiodos que contiene para cada color (rojo, verde, azul y limpio). El elemento clave del sensor a utilizar es el fotodiodo, este

transductor transforma la intensidad de luz en corriente, una vez se obtiene la intensidad de luz de cada componente en corriente (I), se convierte en una señal digital que es almacenada en cuatro registros paralelos para poder ser manipulada según la aplicación (Figura 8)[7].

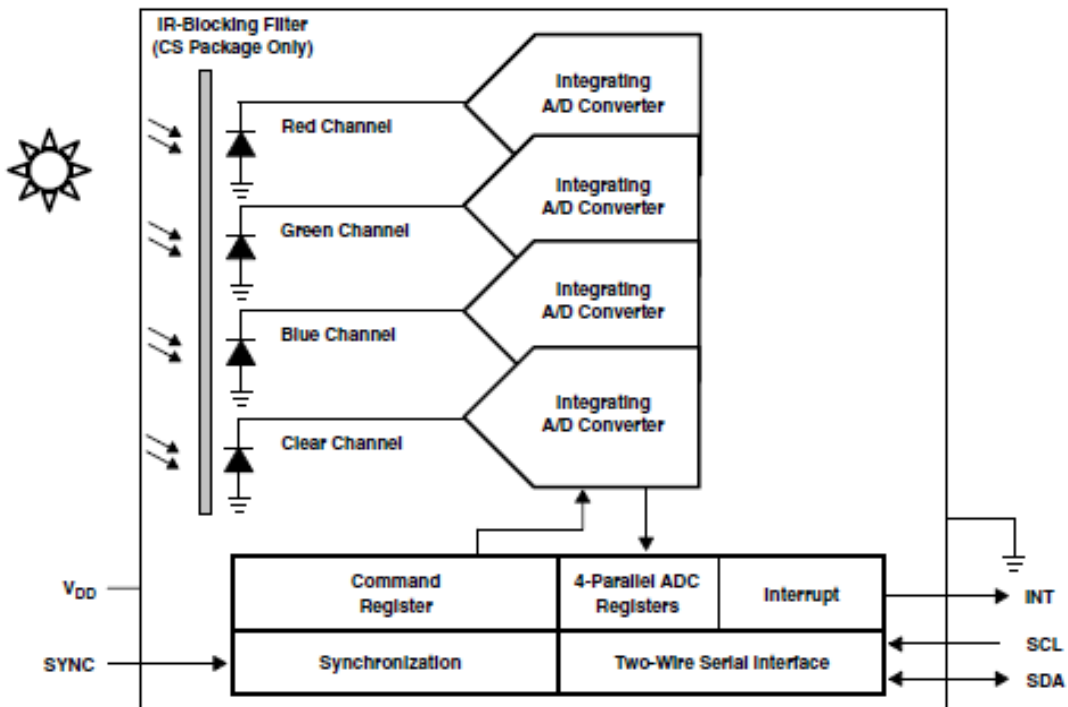


Figura 8. Diagrama Funcional de Bloques del sensor TCS3414 [7].

2.5.3. Principios de Operación

2.5.3.1. Conversores Análogo-Digital

El TCS3414 contiene cuatro conversores análogo-digital (ADC) que convierten la corriente de cada fotodiodo en una salida digital (Canal 1 al canal 4, cada canal corresponde a cada color). La conversión de los cuatro canales ocurre simultáneamente, y antes de completarse el ciclo de conversión los resultados son transferidos al registro de cada canal. Al terminar la transferencia, el dispositivo automáticamente inicia el siguiente ciclo de conversión.

Existen dos formas de controlar los ciclos de conversión: temporizado internamente y temporizado externamente. Los ciclos de conversión temporizados internamente pueden realizar continuamente conversiones una tras otra. Un ciclo de conversión también puede ser solicitado externamente como un único evento utilizando el sincronismo con el pin SYNC.

2.5.3.2. Interfaz Digital

Controlar el TCS3414 es posible a través de una interfaz serie de dos hilos, los cuales modifican un conjunto de registros que proporcionan acceso a las funciones de control y datos de salida del dispositivo. La interfaz serial es compatible con el bus I^2C *Fast Mode*. Este dispositivo soporta una única dirección en modo esclavo, 39h (0111001b).

2.5.3.2. Interrupt

Aun que los registros de los canales ADC pueden ser leídos en cualquier instante para obtener la conversión más reciente, en algunas aplicaciones el sondeo periódico puede no ser deseable. Para este tipo de aplicaciones, el dispositivo soporta una variedad de opciones para interrupciones permitiendo al usuario configurar el dispositivo para que indique cuando un cambio en la intensidad de luz ha ocurrido.

2.5.3.2. Registros

El TCS3414 es controlado y monitoreado por 18 registros de usuario y un registro de comando, el cual se accede a través de la interfaz serial. Estos registros proporcionan una variedad de funciones de control y pueden ser leídos para determinar los resultados de los conversores ADC. La Tabla 2 muestra el conjunto de registros del sensor [7].

ADDRESS	REGISTER NAME	REGISTER FUNCTION
--	COMMAND	Specifies register address
00h	CONTROL	Control of basic functions
01h	TIMING	Integration time/gain control
02h	INTERRUPT	Interrupt control
03h	INT SOURCE	Interrupt source
04h	ID	Part number/ Rev ID
07h	GAIN	ADC gain control
08h	LOW_THRESH_LOW_BYTE	Low byte of low interrupt threshold
09h	LOW_THRESH_HIGH_BYTE	High byte of low interrupt threshold
0Ah	HIGH_THRESH_LOW_BYTE	Low byte of high interrupt threshold
0Bh	HIGH_THRESH_HIGH_BYTE	High byte of high interrupt threshold
0Fh	--	SMBus block read (10h through 17h)
10h	DATA1LOW	Low byte of ADC green channel
11h	DATA1HIGH	High byte of ADC green channel
12h	DATA2LOW	Low byte of ADC red channel
13h	DATA2HIGH	High byte of ADC red channel
14h	DATA3LOW	Low byte of ADC blue channel
15h	DATA3HIGH	High byte of ADC blue channel
16h	DATA4LOW	Low byte of ADC clear channel
17h	DATA4HIGH	High byte of ADC clear channel

Tabla 2. Registros del sensor TCS3414 [7].

2.5.4. Responsividad Espectral

El espectro electromagnético representa los niveles de energía de la luz. La energía es equivalente a la longitud de onda, por lo tanto el espectro electromagnético abarca todas las longitudes de onda que la luz tiene. La porción de longitud de onda que el sensor es capaz de percibir se encuentra en el intervalo de 300nm a 1100nm (Figura 9). La responsividad espectral del sensor se puede analizar en la figura 9.

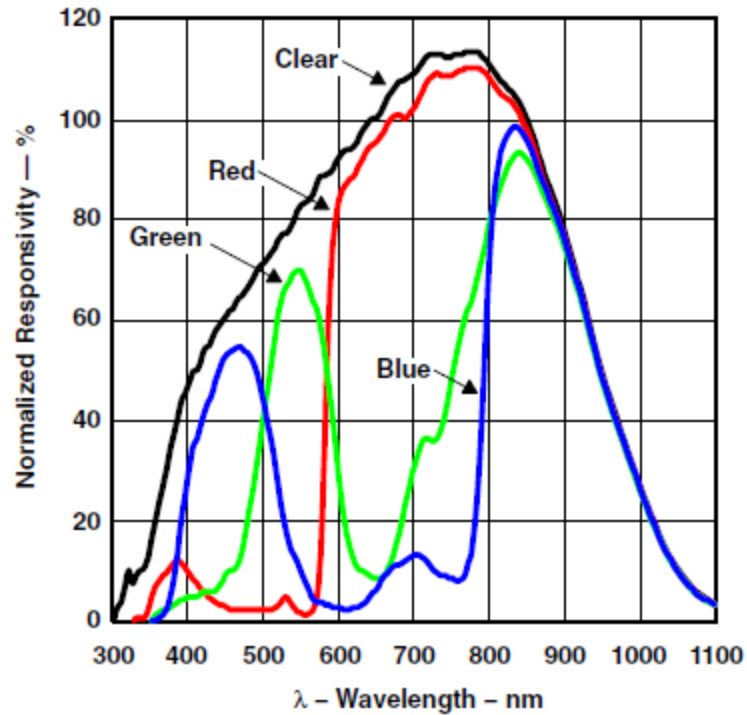


Figura 9. Responsividad espectral.[7]

La longitud de onda de cada color es distinta, para el color azul esta se encuentra en el intervalo de 450nm a 495nm, para el color verde entre 525nm a 560nm y para el color rojo entre 730nm y 790nm. La responsividad espectral está normalizada en una longitud de onda igual a 850nm, en esta longitud de onda el sensor percibe el color blanco (*Clear*).

En los rangos de longitud de onda que se encuentra cada color es importante analizar el comportamiento de las componentes restantes.

- Color Azul (450nm – 495nm): El porcentaje de su responsividad normalizada en aproximadamente el 50%, el color verde un 8% y el rojo un 4%.
- Color Verde (525nm – 560nm): El porcentaje de su responsividad normalizada en aproximadamente el 70%, el color azul un 10% y el rojo un 2%.
- Color Rojo (730nm – 790nm): El porcentaje de su responsividad normalizada en aproximadamente el 110%, el color azul un 10% y el verde un 50%.

2.6. Protocolos

La función del protocolo es definir los requerimientos de Hardware y especificar la forma de comunicación entre capas, además de sus requerimientos en Software para el funcionamiento apropiado de todas las capas.

2.6.1. I2C

Su nombre hace referencia a *Inter-Integrated Circuit* y es un bus de comunicaciones en serie. Diseñado por Philips en el año 1992, es un bus utilizado con frecuencia en la industria, principalmente para comunicar microcontroladores. Las principales características de este protocolo son:

- Bus bidireccional (Serial data y Serial Clock).
- Dos dispositivos solo para comunicación (*Master-Slave Communication*)

2.6.1.1. Terminología

Es importante realizar las definiciones importantes para entender y trabajar sobre este protocolo de comunicación.

2.6.1.1.1. Transmitter

Es el dispositivo que envía los datos al bus. Un transmisor puede ser un dispositivo que pone los datos en el bus por cuenta propia (*Master-Transmitter*) o en respuesta a una petición de datos de otros dispositivos (*Slave-Transmitter*).

2.6.1.1.2. Receiver

Es el dispositivo que recibe los datos del bus.

2.6.1.1.3. Master

Es el componente que inicia la transferencia, genera la señal de reloj y finaliza la transferencia. Un *master*, puede ser un transmisor o un receptor.

2.6.1.1.3. Slave

Es el dispositivo direccionado por el *master*, este puede ser transmisor o receptor.

2.6.1.1.4. Multi-master

La habilidad de que más de un *master* pueda coexistir en un bus sin perder datos o presentar una colisión.

2.6.1.1.5. Arbitration

El procedimiento preestablecido que autoriza solo a un *master* para tomar el control del bus.

2.6.1.1.6. Sync

El procedimiento preestablecido que sincroniza las señales de reloj dadas por dos o más *masters*.

2.6.1.1.7. SDA y SCL

Son los buses bidireccionales encargado de los datos (Serial Data) y de la señal de reloj (Serial Clock).

2.6.1.2. Características del bus

Sólo puede iniciarse una transferencia de datos cuando el bus este libre. Durante la transferencia de datos estos deben permanecer estables mientras la línea de reloj esté a nivel alto. Si se presentan cambios se interpretan como un START o un STOP.

A continuación se muestran las condiciones que se pueden presentar en protocolo I2C:

- Bus libre: Las líneas de datos y de reloj están en valor alto.
- START: Transición de alto a bajo mientras el reloj está alto.
- STOP: Transición de bajo a alto mientras el reloj está alto.
- Validación de datos: La línea de datos presenta un valor válido cuando después de la condición de STAR, la línea se mantiene estable mientras el reloj está alto. Los datos solo pueden cambiar mientras el reloj está en valor bajo (un pulso por dato). Cada transferencia es iniciada con un START y terminada por un STOP. El número de bytes entre ambos está determinado por el *master* y teóricamente no está limitado, a pesar de todo sólo los últimos 16 bytes se almacenan durante la escritura. Al producirse una sobre escritura se reemplazan los datos en un esquema FIFO (*First In First Out*).
- Reconocimiento: Cada receptor cuando se direcciona, está obligado a generar un ACK después de la recepción de cada byte. El *master* debe generar un pulso de reloj extra que se asocia al ACK. El ACK se da colocando un valor bajo sobre SDA durante el valor alto del reloj.

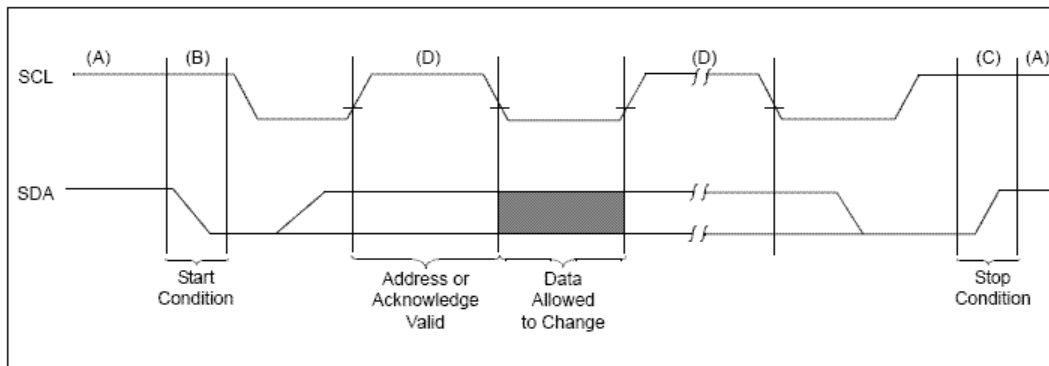


Figura 10. Secuencia de transferencia del bus de datos [8].

El *master* siempre genera las condiciones del STAR y STOP. El bus es considerado ocupado después de la condición de START y se considera libre un tiempo después de la condición de STOP. Este bus se considera ocupado si se repite una condición de STAR (Sr) en vez de una condición STOP. En este sentido, la condición START(S) y la condición repetida START (Sr) son funcionalmente idénticas. La detección de las condiciones de START y STOP de dispositivos conectados al bus, funcionan sin mayor complejidad si se incorpora la interfaz de hardware necesaria. Sin embargo los microcontroladores que no tienen esta interfaz, deben muestrear la línea SDA al menos dos veces por periodo para percibir la transición.

2.6.2. USB

Universal Serial Bus o USB es un puerto que sirve para conectar periféricos a un computador, fue creado por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC. El diseño de USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y así mejorar las capacidades de *plug and play* permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciarlo. A favor del bus USB es preciso decir que cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar (esto depende del sistema operativo).



Figura 11. Símbolo USB [7]

El USB se utiliza en diversos dispositivos como: *mouses*, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras, el USB se ha convertido el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha desplazado los puertos paralelos. Gran variedad de dispositivos requieren una potencia baja, así que pueden conectar varios sin necesitar fuentes de alimentación externas.

2.6.2.1. Interfaz Física

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de $90\Omega \pm 15\%$, cuyos hilos se denominan D+ y D-. Estos utilizan señalización diferencial en half dúplex para combatir los efectos del ruido electromagnético en enlaces largos. Los niveles de transmisión de la señal varían de 0 a 0,3V para bajos (0 lógico) y de 2,8V a 3,6V para altos (1 lógico) en las versiones 1.0 y 1.1, y $\pm 400\text{mA}$ en alta velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a tierra, pero en el modo de alta velocidad se tiene una

terminación de 45Ω a tierra o un diferencial de 90Ω para acoplar la impedancia de cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, dispositivos que tengan un consumo máximo de 500 mA. En la Figura 12 se muestra el conector USB y su enumeración de los pines que se describen a continuación.

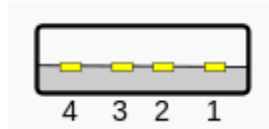


Figura 12. Conector USB, Tipo A [7].

- Pin 1 VCC: color rojo y es alimentación de 5V.
- Pin 2 D-: color blanco y es Data -.
- Pin 3 D+: color verde y es Data +.
- Pin 4 GND: color Negro y es tierra.

2.6.2.2. Velocidades del bus

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- *Low Speed*: Tasa de transferencia de hasta 1,5Mbps (192KB/s). Generalmente utilizado por dispositivos de interfaz humana, teclados, *mouses*, etcétera.
- *Full Speed*: Tasa de transferencia de hasta 12Mbps (1,5MB/s). Esta fue la fuente más rápida antes de la especificación USB 2.0, gran variedad de dispositivos en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO (*Last In First Out*).
- *High Speed*: Tasa de transferencia de hasta 480Mbps (60MB/s). Está presente casi en el 99% de los ordenadores actuales.
- *Super Speed*: Tasa de transferencia de hasta 4,8Gbps (600MB/S).

2.6.2.3. Transferencias

Una transferencia se puede medir como el conjunto global de los datos que forman una comunicación, para el caso USB, está formada por una o varias transacciones de diferentes paquetes de datos, los cuales contienen las tramas de una comunicación USB. La especificación USB permite cuatro tipos de transferencias:

- Control: Se utiliza para configurar y enviar comandos, por ejemplo en la enumeración del dispositivo.

- *Bulk*: Se utiliza para transferencias grandes de datos, es el más rápido pero no hay garantía de que la transmisión de datos se realice en un tiempo determinado. Este tipo contiene un sistema de corrección de datos que verifica que los datos hayan sido enviados correctamente, esta transferencia solo la pueden utilizar dispositivos que soporten las velocidades Full y High Speed.
- *Isócronas*: Es utilizada en dispositivos que transmiten señales de audio y de vídeo en tiempo real. Se garantiza una tasa de velocidad de transmisión determinada. Para utilizar este tipo de transferencia es necesario que los dispositivos soporten velocidades *Full Speed*.
- *Interrupción*: Latencia asegurada y verificación de que los datos se han transmitido correctamente.

2.6.2.4. Enumeración

El *Host* es el encargado de detectar el dispositivo que se conecta al bus. Cuando un dispositivo es detectado el *Host* necesita obtener información sobre él, a este proceso se denomina como enumeración. La información que necesita el Host se encuentra definida en el dispositivo en los descriptores. Los descriptores son datos que se deben guardar en la memoria no volátil del microcontrolador y contienen la siguiente información: El ID del vendedor (VID) y del producto (PID), consumo de corriente del dispositivo, tipo de transferencia, endpoint utilizados, versión USB, clase utilizada, etcétera.

El VID (Vendor ID) y el microcontrolador (Product ID) son dos números de 16 bits representados en Hexadecimal, si se utiliza la clase CDC (Communications Device Class) de CCS para la comunicación USB estos valores se pueden modificar en el archivo "*usb_desc_cdc.h*" y según el sistema operativo instalado en el *Host* deberemos modificarlos también en el archivo *.inf*.

Realmente el modificar estos valores solo es útil si queremos distribuir nuestro producto comercialmente, en ese caso existe una organización llamada *USB Implementers Forum* encarga de proporcionar, con previo pago correspondiente, un VID válido para que el dispositivo conviva sin problemas con el resto de dispositivos USB del mercado. Si no existiera una regulación y cada dispositivo tuviera el VID y PID que quisiera, en el caso en que coincidieran en un mismo sistema dos números iguales, se presentaría un conflicto hardware y ambos dispositivos dejarían de funcionar.

2.6.2.5. Endpoints

Los *Endpoints* son buffers de memoria RAM utilizados para el envío y recepción de datos o comandos de control durante una comunicación USB. Cada *Endpoint* puede ser de entrada, salida de datos o bidireccional, el *Endpoint* 0 está reservado para comandos de control, el proceso de enumeración se realiza a través del *Endpoint* número 0. Este concepto solo se aplica al dispositivo, en el *host* existen también buffers para el envío y recepción de datos, denominados de forma distinta.

2.6.2.6. Clases

Una clase es un modelo que describe el estado y el comportamiento de los objetos que la comparten. La clase provee de propiedades y métodos reusables por los objetos que comparten la clase. La especificación USB provee de propiedades y funciones que pueden ser utilizadas por los dispositivos que tengan características similares. Por ejemplo, un teclado y un *mouse*, por sus características pertenecen a la misma clase *Human Interface Device (HID)*. La especificación USB provee distintas clases para facilitar el desarrollo de dispositivos, a continuación se realiza una descripción de las más utilizadas en microcontroladores.

2.6.2.6.1. HID (Human Interface Device)

Esta clase soporta una velocidad *Low Speed* y tipos de transferencias de control e interrupción. Una característica interesante de esta clase, es que no existe la necesidad de instalar un driver específico en el sistema operativo.

2.6.2.6.2. MSD (Mass Storage Device Class)

Esta clase se enfoca en dispositivos de almacenamiento masivo. Soportada únicamente para dispositivos que manejen velocidades *Full* y *High Speed*. Utiliza transferencias de tipo *Bulk* o combinaciones entre transferencias de tipo control, *Bulk* e *Interrupt*.

2.6.2.6.3. CDC (Communications Device Class)

Un ejemplo de dispositivo que utiliza esta clase son los Modems, para este proyecto se utiliza esta clase para comunicar el PIC24FJ64GB002 con la aplicación de escritorio que se realice con un IDE multiplataforma, que permite ejecutar la aplicación en diferentes sistemas operativos, concretamente en Windows, Linux y MAC. La velocidad máxima al utilizar esta clase será de 80 kBytes/s y el tipo de transferencias soportadas son del tipo interrupción y Bulk. Utiliza también driver estándar incluidos ya en el sistema operativo, según el sistema operativo utilizado precisará o no de la instalación del archivo .INF, al utilizar esta clase en el PC se crea un puerto serie virtual y la comunicación entre el dispositivo y la aplicación de escritorio se hará a través de él, al igual que se haría con un puerto serie físico, esto supone una gran ventaja a la hora de diseñar la aplicación de escritorio, ya que cualquier IDE de programación sea del lenguaje que sea, dispone de un componente o librería que permite el acceso al puerto serie fácilmente.

2.6.2.7. GUI

La Interfaz Gráfica de Usuario fue desarrollada en Visual C# 2010 express, es un IDE para Windows basado en .NET *framework* y Visual Studio 2010. El Visual C# utiliza el lenguaje de programación orientada a objetos, tiene una interfaz gráfica con herramientas comunes en aplicaciones de Windows y aplicaciones Web como botones, cuadros de textos y ventanas. Este permite la integración con distintas librerías dinámicas de enlace (.dll) para interactuar con el hardware y con otros módulos de software como bases de datos u otras librerías. Además permite construir una aplicación ejecutable, que puede ser instalada en diferentes PCs.

3. CRITERIOS DE DISEÑO

Una vez definidos los dos componentes principales para el diseño del calibrador de color para pantallas LCD, es necesario definir qué componentes son necesarios para una correcta interacción entre el sensor (TCS3414) y el microcontrolador (PIC24FJ64GB002). Esta sección hace referencia a los criterios de diseño del *Hardware* de este proyecto.

3.1. Conexiones PIC24FJ64GB002

La hoja de especificaciones del microcontrolador documenta las conexiones mínimas que se debe realizar (Figura 13). Aparte de estas conexiones, se debe tener en cuenta los pines para conectar el ICD3 (Conector ICSP), el conector miniUSB, el regulador de voltaje de 3.3 V (Alimenta los dos componentes principales) y el cristal de 16 MHz, requerido por las conexiones.

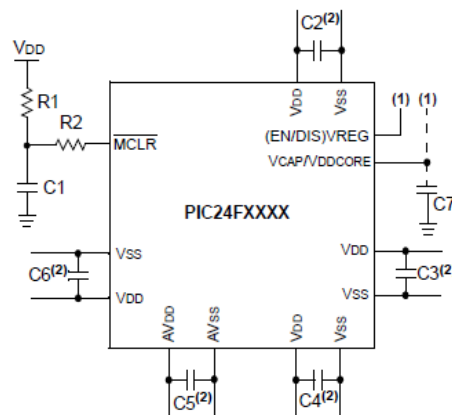


Figura 13. Conexiones mínimas PIC24FJ64GB002 [6].

Según la documentación los condensadores C1 hasta C6, deben tener un valor de 0.1uF, a 20V y de material cerámico. El condensador C7, de valor 10 uF, a 6.3 V o mayor, y su material puede ser cerámico o tantalio. Las resistencias R1 de valor 100Ω y la R2 de 470Ω.

3.2. Conexiones TCS3414

Las conexiones del sensor son muy básicas y vienen documentadas en la hoja de especificaciones. La siguiente figura muestra dicha conexión. Comprenden resistencias de *pull-up* para cada bus de datos y un condensador de desacople.

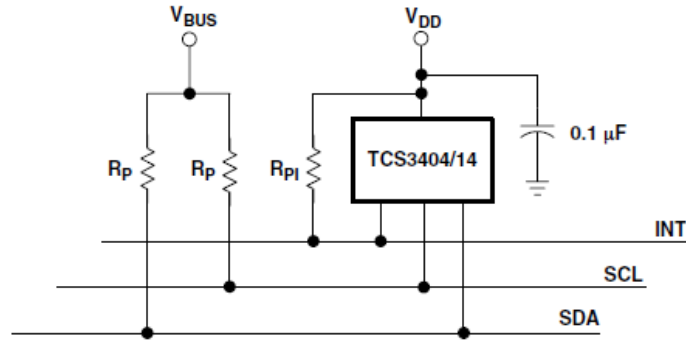


Figura 14. Conexión física TCS3414 [7]

3.3. Esquemático

A continuación se muestra en la Figura 15, el esquemático final del dispositivo, con las conexiones restantes.

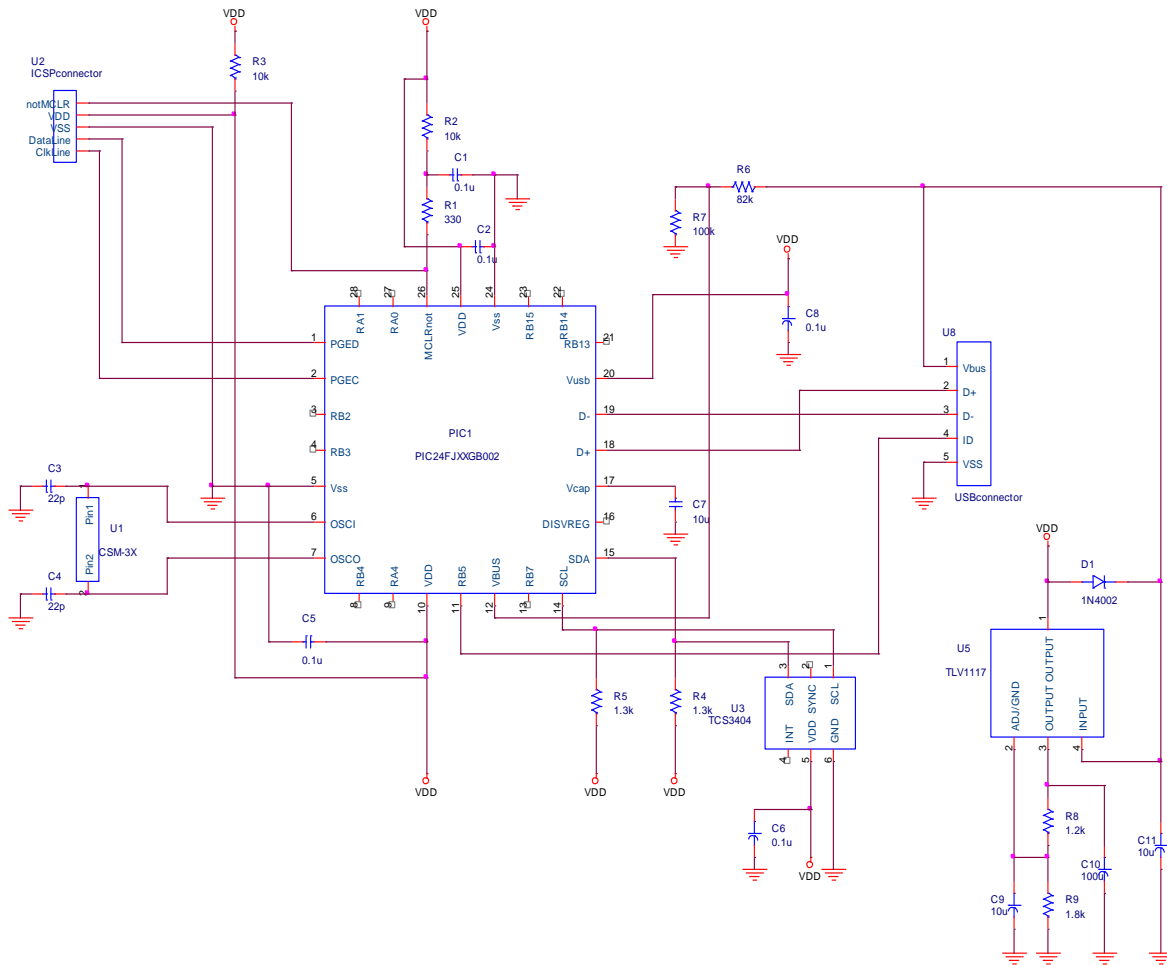


Figura 15. Esquemático final.

4. DESARROLLOS

La propuesta metodológica para el desarrollo de este proyecto, consiste en segmentar por partes la solución final del dispositivo. Primero se debe realizar la comunicación I2C (sensor-microcontrolador) y luego la comunicación USB con el PC.

4.1. Comunicación I2C

Para realizar la comunicación I2C entre el sensor TCS3414 y el microcontrolador PIC24FJ64GB002, fue necesario analizar la figura 16. Donde muestra el esquema para realizar una lectura de los registros del sensor.

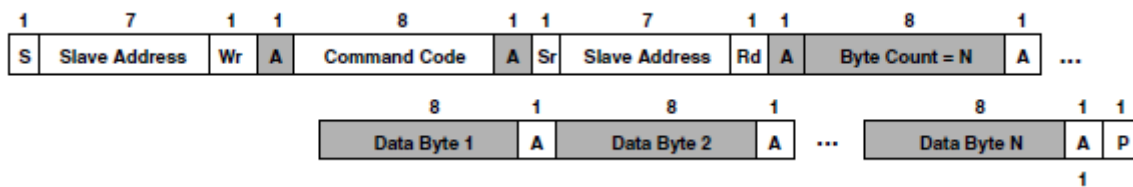


Figura 16. Protocolo I2C read [7].

La figura 17 muestra el código de la primera parte del esquema anterior, tomando como primera parte hasta el envío del primer ACK, mostrado con la letra A.

```
int I2C1read(void)
{
    int i=0;

    for (i;i<8;i++){
        // int err1 = I2C1init();
        I2C1CONbits.SEN = 1;           //Set start bit
        while( I2C1CONbits.SEN);
        I2C1TRN = (SLAVE << 1) & 0xFE; //Send Address WR
        while (I2C1STATbits.TRSTAT); //Wait for TX. TRSTAT=1 =>tx in progress
        I2C1STATbits.TRSTAT=0;
        if( I2C1STATbits.ACKSTAT )    //NACK=1 y ACK=0
        {
            I2C1STATbits.ACKSTAT=1;
            I2C1CONbits.PEN = 1;     // Set stop bit
            while(I2C1CONbits.PEN);
            I2CBuffer[8+i]= '0';     //err=0
            return 0;
            // ret 1 if NAK
        }
    }
}
```

Figura 17. Protocolo I2C, primera parte.

La Figura 18. Muestra el código de las partes restantes del esquema I2C descrito anteriormente.

```

I2C1TRN = 0x90+i; //0x90 + i; //Send Command
while (I2C1STATbits.TRSTAT); //Wait for TX. TRSTAT=1 =>tx in progress
I2C1STATbits.TRSTAT=0;
if( I2C1STATbits.ACKSTAT ) //NACK=1 y ACK=0
{
    I2C1CONbits.PEN = 1; // Set stop bit
    while(I2C1CONbits.PEN);
    I2C1STATbits.ACKSTAT=1;
    I2CBuffer[8+i]= '1';//
    return 1; //err=1 // ret 1 if NAK
}

I2C1CONbits.RSEN = 1;
while( I2C1CONbits.RSEN);

I2C1TRN = ((SLAVE << 1) | 0x01); // send (slave + READ bit ) (<< correrlo para poder enviar el read b.
while ( I2C1STATbits.TRSTAT); // wait for Transmit bit to be cleared 1=master bit is in progress

if( I2C1STATbits.ACKSTAT ) //NACK=1 y ACK=0
{
    I2C1CONbits.PEN = 1; // Set stop bit
    while(I2C1CONbits.PEN);
    I2CBuffer[8+i]= '2';//
    return 2; //err=2 // ret 1 if NAK
}

```

Figura 18. Protocolo I2C, segunda parte.

Para revisar el código más detalladamente, se puede ver el Anexo 1. Para que este protocolo funcione correctamente es necesario realizar una inicialización del mismo que corresponde a la figura 19.

```

int I2CInit(){

    I2C1MSK = MASTER; // set this controller's device address

    I2C1CONbits.I2CEN = 1; // enable I2C module
    I2C1CONbits.I2CSIDL = 0; // Continue module operation in Idle mode -----> Qué es Idle Mode?
    //I2C1CONbits.SCLREL = ;
    I2C1CONbits.IPMIEN = 0; // IPMI mode disabled -----> Qué es IPMI Mode?
    I2C1CONbits.A10M =0; // I2C1ADD is a 7-bit slave address
    I2C1CONbits.DISSLW = 1; // for 400 kHz up enable slew rate
    I2C1CONbits.SMEN = 1; // signal conditioning
    I2C1CONbits.GCEN = 0; // for slave mode only
    I2C1CONbits.STREN = 1; // Enable software or receive clock stretching, therefore clock must be re
    I2C1CONbits.ACKDT =0; //Send Ack
    I2C1CONbits.RCEN=0;
    I2C1BRG = 0x9D; // use 1MHz SCL frequency but apparently, its frequency is
    //0x9D
    IEC1bits.SI2C1IE = 0; // enable Slave I2C1 interrupt 0=disable
    IEC1bits.MI2C1IE = 0; // enable Master I2C1 interrupt 0=disable

    I2C1STATbits.BCL =0;
}

```

Figura 19. Inicialización I2C.

Después de realizar una breve descripción de cómo fue implementado el protocolo I2C, es importante explicar cómo funciona a un nivel más alto. La figura 21 ilustra cómo trabaja la máquina de estados de este protocolo.

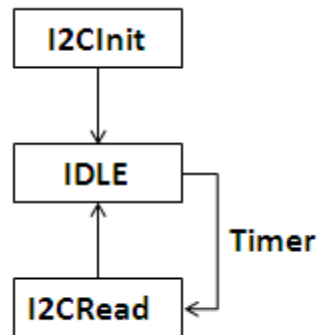


Figura 20. Máquina de estados.

Esta máquina de estados se puede ver descrita en el código desarrollado en la Figura 21.

```

/*-----Maquina de estados de IIC-----*/
void iic_process (int iic_state){

    int err;
    switch (iic_state){
        case IIC_INIT :{
            if(!I2CInit()){

                iic_state=IIC_IDLE;
            }
        };
        break;

        case IIC_READ :{

            err=I2Cread() ;

            iic_state=IIC_IDLE;

        };
        break;

        case IIC_IDLE :{

        };
        break;
    };
}
  
```

Figura 21. Máquina de estados descrita en código.

4.2. Comunicación USB

Se implementó un dispositivo que comunica vía USB (*polling*) los datos medidos por un sensor específico. Para poder analizar los datos, como usuario final del producto o cliente, se desarrolló una interfaz de usuario (GUI) que permite visualizar los datos con mayor facilidad (Figura 22). En el Anexo 1 se describe el Software utilizado.



Figura 22. Interfaz de usuario.

Con el desarrollo de este dispositivo se analizaron temas técnicos, como la instalación de un dispositivo USB en el sistema operativo Windows. También, la comunicación de una aplicación desarrollada en C# con el dispositivo para la visualización en tiempo real de los datos obtenidos por el sensor TCS3414.

Debido a la complejidad que presenta la comunicación USB, el fabricante del microcontrolador (Microchip®) proporciona librerías cuya finalidad es facilitar la programación del dispositivo. Por esta razón es necesario conocer qué funciones proporciona la librería para poder enviar y recibir datos a través del bus. En el caso de este proyecto se utiliza la clase CDC (Communications Device Class) y las funciones que se utilizan para enviar y recibir datos por USB son:

- `Usb_cdc_kbhit()`: Es una función que retorna un valor booleano según los caracteres que estén esperando en el buffer de recepción.
- `Usb_cdc_getc()`: Obtiene el carácter recibido en el buffer de recepción
- `Usb_cdc_putc(char c)`: Coloca el carácter que recibe como parámetro en el buffer de transmisión para ser enviado.

Estas librerías contienen otras funciones, las cuales se pueden analizar en el archivo `usb_cdc.h`. La comunicación USB es iniciada con la función de inicio, hay que tener en cuenta que para que la comunicación USB se mantenga deber ser realimentada constantemente, esto se realiza llamando la función *task* periódicamente, con esto se busca tener un ciclo infinito donde se incluye el llamado de esta función.

Para poder realizar la comunicación USB es necesario realizar o conseguir el firmware del microcontrolador. En la página del fabricante del microcontrolador, se pueden descargar el firmware del dispositivo y realizarle los ajustes necesarios para poder realizar la comunicación. A continuación, en la figura 23, se muestra la función que comunica los datos al PC.

```
Start Page x main.c x
138
139 unsigned char counter;
140 unsigned char I2CBuffer[20];
141 int ColorChannel[6];
142 int to;
143 int iic_state;
144
145 int tic;
146
147 /*Funcion de timer en software*/
148 void timer1 (void){
149
150     tic++;
151     if (tic==TIMER1){           //TIMER1 es un define del timer de software
152         iic_state=IIC_READ;    //Cambiar el estado de IIC para leer los datos
153
154         if (USBUSARTIsTxTrfReady())
155         {
156             putUSBUSART (I2CBuffer,20);    //Enviar los datos al puerto COM
157         }
158         tic=0;
159
160     }
161 };
162
```

Figura 23. Función en el microcontrolador que envía datos al puerto COM

Para entender en un nivel más alto, como se realiza el proceso completo de comunicación de datos del microcontrolador al PC, la figura 24 ilustrará este proceso. El esquema se puede revisar detenidamente el anexo del código (Anexo 1).

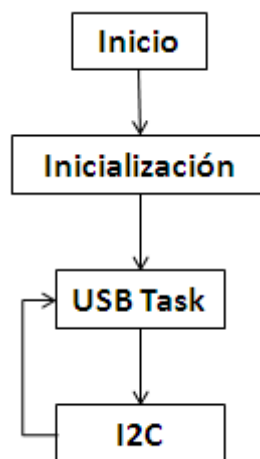


Figura 24. Comunicación USB

5. PROTOCOLO DE PRUEBAS

Es importante tener claridad sobre las fases que contemplan las pruebas de funcionamiento del dispositivo. El protocolo de pruebas está conformado por tres fases, preparación, ejecución y análisis de resultados. Es preciso indicar que las mediciones que entrega el dispositivo no tienen ningún tipo de unidad.

5.1. Preparación

El sensor TCS3414 mide la cromaticidad (color) y luminancia (intensidad) de la luz. Es necesario rectificar y aclarar qué mide el dispositivo, esto con la finalidad de realizar las pruebas necesarias para poder analizar su funcionamiento. La cromaticidad es la medida que identifica los colores que está emitiendo la pantalla LCD y la luminancia mide la intensidad de luz que entrega la pantalla.

5.1.1. Aislamiento del Sensor

Para poder tener mediciones que no sean afectadas por las emisiones de luz externas a la de la pantalla, es necesario aislar el sensor, dejando únicamente la abertura por donde se van a realizar las mediciones.

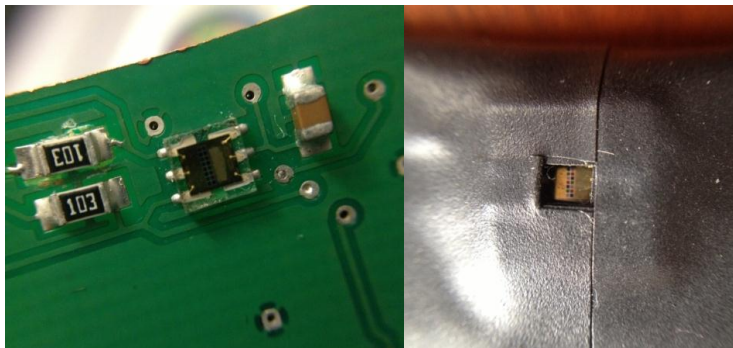


Figura 25. Aislamiento del sensor.

Para verificar que el empaque realmente está evitando que se filtre luz externa a la de la pantalla, se debe realizar una prueba cerrando la abertura y confirmando que el valor de todas las componentes, sean de valor “cero”.

5.1.2. Calibración del dispositivo

El TCS3414 es utilizado en diferentes tipos de aplicaciones, HDTVs, tablets, laptops, monitores, instrumentación médica, iluminación comercial, procesos industriales, etc. Por esta razón es importante identificar qué tipo de mediciones debe realizar el sensor. Para este proyecto, el sensor realiza las mediciones sobre monitores (pantallas) LCD, por esta razón se debe identificar la intensidad máxima emitida por las pantallas y ajustar la cromaticidad de cada componente respaldándose de un dispositivo certificado que contenga los colores correctamente calibrados.

Para realizar la “*calibración*” del dispositivo se realizan mediciones de intensidad y cromaticidad sobre la pantalla de un computador iMac Mid2010, equipo profesional especializado para trabajar en imagen y video. Este equipo contiene un calibrador de color que permite realizar mediciones sobre colores con medidas certificadas. Este equipo permite ajustar la intensidad de la pantalla, logrando igualar la intensidad de luz de una pantalla LCD.

Para calibrar la intensidad de luz máxima que debe medir el sensor, se utiliza la componente *clear*, realizando mediciones sobre el color blanco en ambas pantallas. Se utiliza esta componente, ya que es la que mayor luminosidad emite.

5.1.3. Tipos de pruebas

A continuación se realiza una breve descripción de las pruebas que se deben realizar sobre el dispositivo.

5.1.3.1. Prueba de aislamiento del sensor

Con la finalidad de realizar mediciones precisas, es necesario realizar una prueba sobre el empaque que aísla el sensor de recibir emisiones de luz distintas a la de la pantalla. Para esta prueba es necesario situar el sensor sobre una superficie plana que no sea la pantalla LCD. Al realizar la medición las componentes deben estar en 0, rectificando que no se emite ningún tipo de luz sobre el sensor al situarlo en una superficie plana.

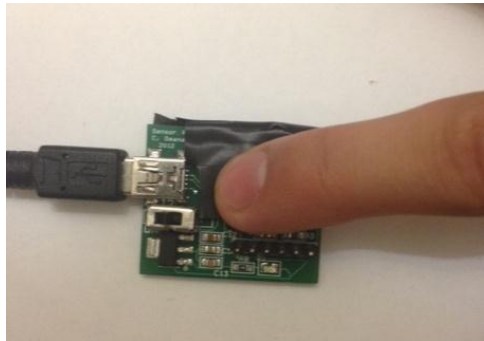


Figura 26. Ejemplo prueba de aislamiento del sensor.

5.1.3.2. Prueba de sin intensidad lumínica

Esta prueba se puede enfocar como una ratificación de la prueba anterior. El color de cada objeto varía según la iluminación que este tenga. Por ejemplo, un balón de fútbol puede presentar un determinado color si la luz que recibe en un día soleado, pero si a este mismo balón se le aplica una luz cálida de un bombillo, este presentaría un aspecto amarillo y cambiaría la percepción de color que tiene, cambiaría su color. En esta prueba se debe situar el sensor en un ambiente donde no se emita ningún tipo de luz, con esto se ratifica que el sensor responde correctamente cuando no hay emisiones de luz.

El sensor se debe situar sobre un estuche el cual se pueda cerrar completamente y aislar la luz en su totalidad, para poder realizar la medición.



Figura 27. Prueba sin intensidad de luz sobre el sensor.

5.1.3.3. Prueba del círculo cromático

La teoría de color RGB expone que los colores primarios son el rojo, verde y azul, y determinada combinación de los mismos, crean los colores secundarios (cian, amarillo y magenta). Esta prueba ratifica que cada componente responda según el color sobre el que se esté situando el sensor. Por esta razón se deben realizar mediciones sobre los colores primarios y luego sobre los colores secundarios, analizando la respuesta de cada componente. Esta prueba se debe realizar sobre el monitor que está calibrado (en este caso un iMac Mid2010) y sobre estos resultados se realiza la calibración del dispositivo. Para realizar las pruebas descritas en los numerales 5.1.3.3 y 5.1.3.4 se debe situar el sensor sobre la pantalla como lo indica la figura 28.



Figura 28. Instrucción para pruebas.

5.1.3.4. Prueba sobre monitor LCD

En esta prueba se realizan las mediciones de los colores primarios sobre una pantalla LCD, con la finalidad de analizar que ajustes se deben realizar sobre la pantalla para obtener colores cercanos a valores de pureza certificados. Esta prueba mide si la pantalla debe ser calibrada para tener los colores puros de RGB.

5.2. Ejecución

5.2.1. Ejecución de Prueba de aislamiento del sensor

Al seguir los procedimientos de la prueba listada en la sección 5.1.3.1, las mediciones entregadas por el sensor fueron las mostradas en la Figura 29.



Figura 5. Resultado prueba aislamiento del sensor.

5.2.2. Ejecución de Prueba sin intensidad lumínica

Al seguir los procedimientos de la prueba listada en la sección 5.1.3.2, las mediciones entregadas por el sensor fueron las mostradas en la Figura.



Figura 6. Resultado prueba sin intensidad lumínica.

5.2.3. Ejecución de Prueba del círculo cromático

Al seguir los procedimientos de la prueba listada en la sección 5.1.3.3, las mediciones entregadas por el sensor al situarlo sobre el monitor iMac Mid2010 con los colores primarios fueron los siguientes.

Componentes	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco
Rojo	13	2	0	15
Verde	2	10	1	15
Azul	0	3	5	7
Clear	16	15	6	37

Tabla 3. Resultados prueba colores primarios

Las mediciones entregadas por el sensor al situarlo sobre el monitor iMac Mid2010 con los colores secundarios fueron los siguientes.

Componentes	Medición sobre Amarillo	Medición sobre Magenta	Medición sobre Cian
Rojo	15	14	2
Verde	14	4	12
Azul	2	5	8
Clear	31	23	23

Tabla 4. Resultados prueba colores secundarios.

5.2.4. Ejecución de Prueba sobre monitor LCD

Al seguir los procedimientos de la prueba listada en la sección 5.1.3.4, las mediciones entregadas por el sensor fueron las mostradas en la Tabla 5.

Componentes	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco
Rojo	10	4	2	13
Verde	4	8	5	14
Azul	2	4	7	11
Clear	14	15	12	36

Tabla 5. Resultado prueba sobre monitor LCD.

5.3. Análisis de Resultados

A continuación se realiza un análisis de los resultados sobre las pruebas descritas en la etapa de ejecución. Se espera enfocar el análisis sobre la veracidad de las mediciones del sensor y la pantalla sobre cual se realizó la calibración del dispositivo.

5.3.1. Análisis prueba de aislamiento del sensor

Para poder realizar un análisis de los resultados obtenidos en esta prueba, es importante aclarar cuáles eran los resultados esperados de la prueba. Se estaba verificando que el empaque que aislaba el sensor funcionará correctamente, que no se filtrará ninguna emisión de luz. Los resultados esperados para esta prueba era tener todas las componentes (rojo, verde, azul y *Clear*) en valor cero. La siguiente tabla muestra los resultados esperados contra los resultados obtenidos.

Componentes	Resultados Esperados				Resultados Obtenidos			
	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco
Rojo	0	0	0	0	0	0	0	0
Verde	0	0	0	0	0	0	0	0
Azul	0	0	0	0	0	0	0	0
Clear	0	0	0	0	0	0	0	0

Tabla 6. Resultados pruebas de aislamiento del sensor.

Los resultados de esta prueba fueron los esperados, el empaque que está aislando el sensor está cumpliendo su función correctamente, según los datos se puede observar que no se está filtrando ninguna emisión de luz al ubicar el sensor sobre cualquier superficie.

5.3.2. Análisis prueba sin intensidad lumínica

La ejecución de esta prueba era una ratificación de la prueba anterior. Al exponer el sensor en un ambiente donde no se emita luz, las componentes del sensor deben ser de valor 0. Al realizar esta prueba se comprueba que el sensor funciona correctamente en ambientes donde no se emite luz. Los resultados son iguales a los mostrados en la tabla anterior.

5.3.3. Análisis Prueba del círculo cromático

Con esta prueba se analiza el comportamiento del sensor al realizar mediciones sobre los colores primarios y secundarios, emitidos por una pantalla LCD. Primero se realiza un análisis sobre los resultados de las componentes al realizar mediciones sobre los colores primarios.

5.3.3.1. Análisis colores primarios

La siguiente tabla muestra los resultados esperados y los obtenidos en la ejecución.

Componentes	Resultados Esperados				Resultados Obtenidos			
	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco
Rojo	15	0	0	15	13	2	0	15
Verde	0	15	0	15	2	10	1	15
Azul	0	0	7	7	0	3	5	7
Clear	16	15	7	37	16	15	6	37

Tabla 7. Tabla de resultados colores primarios

Como se puede observar en la tabla, los resultados obtenidos no son iguales a los resultados esperados. Es importante separar el análisis de esta prueba en dos partes, la primera es analizar la veracidad de la medida, y la segunda analizar si el sensor es capaz de diferenciar los colores.

La calibración del dispositivo se realizó ajustando todas las componentes en su valor RGB al máximo sobre el monitor que esta calibrado (Figura 31). La combinación de todas las componentes en su valor máximo entrega el color blanco, al realizar la calibración sobre este color, se puede conocer cuál es el valor máximo de cada componente, el cual sería el resultado esperado si se realiza la medición, de cada componente aislada.

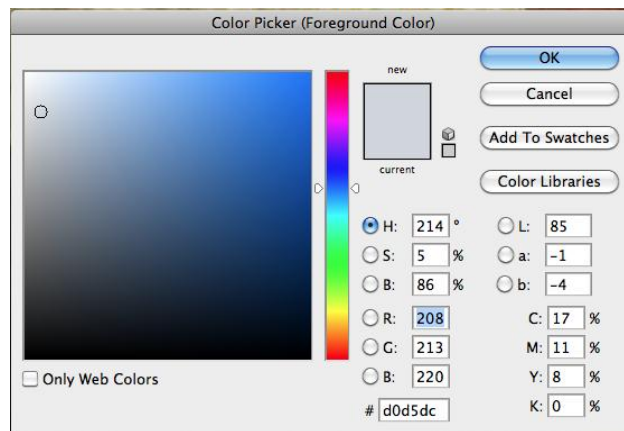


Figura 31. Interfaz Adobe Photoshop Cs5.

El porcentaje de error sobre cada componente es el siguiente:

- Componente Rojo: 13,33%
- Componente Verde: 33,33%
- Componente Azul: 28,57%

Estos porcentajes indican que al ingresar un color RGB de valor máximo (solo una componente al máximo) el sensor encuentra emisiones de luz de las otras dos componentes. A continuación se muestra el porcentaje de las componentes que se filtran a medir un color puro (Tabla 8).

Filtración de Componentes (%)	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul
Rojo		13,33	0
Verde	13,33		14,28
Azul	0	20	

Tabla 8. Filtración de Componentes.

La segunda parte del análisis debe mostrar si el sensor tiene la capacidad de distinguir (filtrar) los colores que la pantalla LCD está emitiendo. Al analizar los resultados de cada componente, en las situaciones que se ajusta un color RGB en su máximo valor, el porcentaje de color más alto siempre corresponde al color que se está midiendo. La siguiente tabla muestra los porcentajes de cada componente en las mediciones realizadas.

Componentes (%)	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul
Rojo	86,67	13,33	0
Verde	13,33	66,67	14,28
Azul	0	20	71,42

Tabla 9. Porcentajes de cada componente.

Analizando los porcentajes de la tabla anterior al configurar color en su valor máximo, el porcentaje de la componente que mayor valor tiene corresponde al color seleccionado para la medición. Los resultados de esta prueba muestran que la componente de color azul se comporta de forma distinta a las otras dos componentes, esta componente presenta un valor de casi la mitad respecto a las otras

componentes. Los resultados de la prueba sobre los colores secundarios confirmarán si el sensor está filtrando correctamente cada componente.

5.3.3.2. Análisis colores secundarios

La siguiente tabla muestra los resultados esperados y los obtenidos en la ejecución, con la cual se espera analizar la capacidad del sensor de reconocer cada componente.

Componentes	Resultados Esperados			Resultados Obtenidos		
	Medición sobre Amarillo	Medición sobre Magenta	Medición sobre Cian	Medición sobre Amarillo	Medición sobre Magenta	Medición sobre Cian
Rojo	15	15	0	15	13	2
Verde	15	0	15	14	4	13
Azul	0	7	7	3	5	7
Clear	30	22	22	32	22	22

Tabla 10. Resultados colores secundarios

Analizando los resultados obtenidos, estos no son iguales a los resultados esperados, pero en esta prueba analizaremos que el porcentaje de error, respecto al resultado esperado es bajo.

Esta prueba se realizó sobre colores secundarios, los cuales se obtienen a partir de la combinación de colores primarios, el análisis se realizó sobre cada combinación de colores.

5.3.3.2.1. Amarillo

El color amarillo se obtiene de la combinación de los colores rojo y verde, por esta razón al realizar la medición sobre este color, las componentes rojo y verde deben estar en su valor máximo. El siguiente listado muestra el porcentaje de error de cada componente. Para esta prueba es importante tener en cuenta la componente *clear*, ya que esta refleja el valor total de la medición.

- Rojo: 0%
- Verde: 6,66%
- Azul: 10% (Respecto al valor total *Clear*)
- Clear: 6,66%

Esta medición presentó errores sobre los colores verde y azul. La componente verde presentó un error del 6% y la componente azul, que esperaba un valor de 0 en su medición, presentó un porcentaje de error del 10%. La componente *clear*, que representa el total de todas las componentes,

presento un porcentaje de errores del 6%, lo que muestra que la medición sobre el color amarillo fue muy cercana a la esperada.

5.3.3.2.1. Magenta

El color magenta se obtiene de la combinación de los colores rojo y azul, por esta razón al realizar la medición sobre este color, las componentes rojo y azul deben estar en su valor máximo. El siguiente listado muestra el porcentaje de error de cada componente. Para esta prueba es importante tener en cuenta la componente *clear*, ya que esta refleja el valor total de la medición.

- Rojo: 13,33%
- Verde: 18,88% (Respecto al valor total *Clear*)
- Azul: 28,57%
- Clear: 0%

Esta medición presentó errores sobre las tres componentes principales. La componente de color rojo presentó un error del 13%, la verde 18% y la azul 28%, lo que son porcentajes de valor alto si se analiza componente por componente. La componente que representa un el valor total de la medición no presentó error, su medición cumplió el valor obtenido. Esto puede ocurrir ya que la combinación en porcentajes bajos de componentes que no debían aparecer, compensaron los valores de las componentes que no cumplieron el resultado esperado.

5.3.3.2.3. Cian

El color magenta se obtiene de la combinación de los colores verde y azul, por esta razón al realizar la medición sobre este color, las componentes verde y azul deben estar en su valor máximo. El siguiente listado muestra el porcentaje de error de cada componente. Para esta prueba es importante tener en cuenta la componente *clear*, ya que esta refleja el valor total de la medición.

- Rojo: 9,09% (Respecto al valor total *Clear*)
- Verde: 13,33%
- Azul: 0%
- Clear: 0%

Esta medición presentó errores sobre las componentes rojo y verde. La componente de color rojo presentó un error del 9%, la verde 13%. Este fue el color que presentó porcentajes más bajos al realizar mediciones, dos de las componentes fallaron el 10% aproximadamente y la componente que mide el valor total no presentó error. Existe la posibilidad, al igual que el caso anterior, que la aparición de componentes que no deben aparecer, nivelen las que no están presentando el valor esperado.

5.3.3. Análisis Prueba sobre monitor LCD

Los resultados de las pruebas anteriores muestran que en la calibración del dispositivo, no se lograron obtener colores RGB completamente puros. Por esa razón, esta prueba se realizará tomando como punto de referencia, las mediciones obtenidas en el proceso de calibración. La Tabla 11 muestra los resultados esperados y los resultados obtenidos al realizar mediciones sobre un monitor LCD.

Componentes	Resultados Esperados				Resultados Obtenidos			
	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco	Medición sobre Rojo	Medición sobre Verde	Medición sobre Azul	Medición sobre Blanco
Rojo	13	2	0	15	10	4	2	13
Verde	2	10	1	15	4	9	5	15
Azul	0	3	5	7	2	4	8	11
Clear	16	15	6	37	14	15	12	36

Tabla 11. Resultados pruebas sobre monitor LCD.

Los siguientes resultados muestran en que porcentaje está descalibrada la pantalla LCD respecto al el monitor iMac Mid2010. Este análisis se realizará componente a componente.

5.3.3.1. Componente Rojo

Los resultados de la Tabla 12 muestran que el monitor LCD está descalibrado respecto al monitor sobre el cual se calibró el dispositivo. El monitor LCD se encuentra aproximadamente un 15% descalibrado respecto al monitor iMac Mid2010.

Componentes	Resultado Esperado Medición Monitor Calibrado	Resultado Obtenido Medición LCD	Porcentaje de error
Rojo	13	10	23%
Verde	2	4	12,5%
Azul	0	2	12,5%
Clear	16	14	12,5%

Tabla 12. Mediciones LCD componente rojo.

5.3.3.2. Componente Verde

Los resultados de la tabla 13 muestran que el monitor LCD está descalibrado respecto al monitor sobre el cual se calibró el dispositivo. El monitor LCD se encuentra aproximadamente un 10% descalibrado respecto al monitor IMac Mid2010.

Verde	Resultado Esperado	Resultado Obtenido	
Componentes	Medición Monitor Calibrado	Medición LCD	Porcentaje de error
Rojo	2	4	13,33%
Verde	10	9	10%
Azul	3	4	6,66%
Clear	15	15	0%

Tabla 13. Mediciones LCD componente verde.

5.3.3.3. Componente Azul

Los resultados de la tabla 14 muestran que el monitor LCD está descalibrado respecto al monitor sobre el cual se calibró el dispositivo. El monitor LCD se encuentra aproximadamente un 60% descalibrado respecto al monitor IMac Mid2010.

Azul	Resultado Esperado	Resultado Obtenido	
Componentes	Medición Monitor Calibrado	Medición LCD	Porcentaje de error
Rojo	0	2	33,33%
Verde	1	5	83,33%
Azul	5	8	66,67%
Clear	6	12	50%

Tabla 14. Mediciones LCD componente Azul.

6. CONCLUSIONES

Este proyecto tenía como objetivo principal diseñar un prototipo funcional de un sistema para medir las componentes RGB de la pantalla de un monitor LCD. El cumplimiento de este objetivo se refleja con el protocolo de pruebas realizado sobre el dispositivo, el cual es uno de los objetivos específicos del proyecto. El protocolo de pruebas permitió analizar que las mediciones de las componentes RGB, deben ser normalizadas respecto a la grafica de Responsividad Espectral (Numeral 2.5.4), la participación de componentes de otros colores al realizar mediciones sobre colores puros, corresponde al comportamiento de la figura 10.

El sensor entrega dos componentes por cada color que mide (entrega un *high* y un *low*, de 8 bits cada uno). Las mediciones realizadas sobre los monitores siempre mostraban la participación de la componente *low*, pero nunca la de la componente *high*, lo que puede indicar que la configuración de software utilizada en el sensor no tenga un funcionamiento óptimo con la intensidad de luz de una pantalla LCD, para este caso sería necesario programar la ganancia y analizar nuevamente los resultados. Para descartar que la componente *high*, esté funcionando incorrectamente, se realizaron pruebas en otros ambientes de luz y el sensor respondía enviando datos de esta componente.

Una de las posibles causas de la diferencia de los resultados esperados contra los resultados obtenidos, es el aislamiento del sensor. Este aislamiento deja una abertura de 2mm por 2,5mm, en esta área se ubican 93,5 pixeles (Figura 32). Al desplazar el sensor medio pixel, la medición puede no estar teniendo en cuenta alguna componente RGB por lo menos de 11 pixeles, lo que afectaría la medición en un 12%.

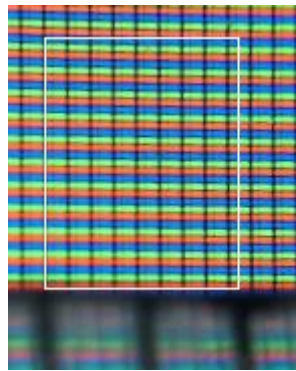


Figura 32. Área abertura aislamiento del sensor.

Las mediciones realizadas muestran que el monitor LCD del computador TOSHIBA Satellite Pro L300 no es adecuado para trabajar sobre imagen o video. Esta pantalla se encuentran aproximadamente un 30% descalibrada en comparación de una pantalla de un equipo especializado para trabajar en este campo. Para terminar de revisar todo el alcance del proyecto, enfocándose en la parte del Hardware, se logró construir con éxito un dispositivo alimentado por USB, con una salida digital serial de 8 bits que representa cada componente RGB.

7. BIBLIOGRAFÍA

- [1] Wucius Wong, “Principios del diseño de color”, 1990.
- [2] RGB. <http://www.w3.org/Graphics/Color/sRGB.html>. Consultada el 12 de Mayo del 2011.
- [3] ISO. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=54754. Consultada el 12 de Mayo del 2011.
- [4] Antonio Valero Muñoz, “Principios de Color y Holopintura”, 1980.
- [5] Omar Enrique Barra Zapata, “Microcontroladores PIC con Programación PBP”, 2011.
- [6] Microchip, “39940d”, 2010.
- [7] TAOS, “TCS3404 TCS3414-A”, 2011.
- [8] Microchip, “AN735”, 2000.

8. ANEXOS

Anexo 1. Main code

Anexo 2. Esquemático