



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA AUTÓNOMO DE DETECCIÓN Y APUNTAMIENTO DE OBJETOS POR COLOR MEDIANTE EL USO DE UNA CÁMARA DE VIDEO INTEGRADA A UN MICROCONTROLADOR

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y CONTROL**

DIEGO ESTEBAN AYALA CAJAS

diegomess89@hotmail.com

DIRECTOR: DR. GEOVANNY DANILO CHÁVEZ GARCÍA

danilochavezpn@gmail.com

DECLARACIÓN

Yo, Diego Esteban Ayala Cajas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

DIEGO ESTEBAN AYALA CAJAS

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Diego Esteban Ayala Cajas, bajo mi supervisión.

Dr. Danilo Chávez
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

A la Escuela Politécnica Nacional y sus docentes por su formación profesional y por brindarme los conocimientos necesarios para desarrollar este proyecto.

A mis padres por su esfuerzo y sacrificio durante todas las etapas de mi vida, por su constante preocupación por mi bienestar y educación.

A mi familia y amigos por su comprensión y apoyo en esta etapa de mi vida.

Al Dr. Danilo Chávez por su interés en este proyecto, por ayudarme a desarrollarlo con paciencia y profesionalismo, y por su amistad.

Al Dr. Andrés Rosales por guiarme en la culminación de mis estudios de pregrado y por confiar en mis capacidades.

Al Dr. Eduardo Ávalos cuya dedicación en la investigación y el desarrollo del país hizo posible la realización de este proyecto.

Al MSc. Ramiro Valenzuela y la MSc. Ana Rodas por dedicar su tiempo e interés en el presente trabajo, y por guiarme en progreso del mismo.

A todas las personas que estuvieron involucradas y me apoyaron en este desarrollo.

Diego Ayala

DEDICATORIA

El presente trabajo está dedicado a mis padres César y Olga, cuyo amor, apoyo y esmero, han sido fundamentales para ser una persona de bien y dedicada a dar lo mejor de mí en cada paso de la vida. A mi familia y hermanos, en especial a Andrés que desde pequeño me ha inculcado curiosidad por la tecnología y por averiguar el funcionamiento de las cosas. A mis amigos por acompañarme y compartir los buenos y malos momentos. Y finalmente al amor de mi vida Kathy por ayudarme a encontrar la felicidad junto a ella, por ser un apoyo incondicional, por escucharme, y siempre confiar en mí.

Diego Ayala

CONTENIDO

CONTENIDO	V
RESUMEN	VIII
PRESENTACIÓN	IX
CAPÍTULO 1	1
MARCO TEÓRICO.....	1
1.1 VISION COMPUTARIZADA.....	1
1.1.1 CAPTURA DIGITAL DE LA IMAGEN	1
1.1.1.1 Foto-sensores	1
1.1.1.1.1 <i>Dispositivo de carga acoplada (CCD)</i>	2
1.1.1.1.2 <i>Sensor de pixel activo (APS)</i>	3
1.1.1.2 Detección de colores	4
1.1.1.2.1 <i>Espacio de colores</i>	4
1.1.1.3 Formatos de color	6
1.1.1.4 Formación de la imagen	7
1.1.2 PROCESAMIENTO	8
1.1.2.1 Pre-procesamiento.....	8
1.1.2.2 Segmentación	9
1.1.2.2.1 <i>Detección de bordes</i>	9
1.1.2.2.2 <i>Detección de regiones de interés</i>	10
1.1.2.3 Descripción y Reconocimiento	11
1.1.2.4 Interpretación	11
1.2 USO DE MICROCONTROLADORES PARA PROCESAMIENTO DE IMAGEN	11
1.2.1 FAMILIA DE MICROCONTROLADORES ARM CORTEX-M.....	12
1.2.1.1 Ventajas frente a los microcontroladores de 16, y 8 bits.	12
1.2.1.2 Microcontrolador empleado.....	13
1.2.1.2.1 <i>Tarjeta de desarrollo</i>	15
1.2.1.3 Software de Programación	15
1.2.1.3.1 <i>Depuración de programas para el microcontrolador empleado</i>	16

CAPÍTULO 2	17
DESARROLLO DEL SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE IMAGEN	17
2.1 METODOLOGÍA DEL SISTEMA DE ADQUISICIÓN Y VISUALIZACIÓN DE LAS IMÁGENES	17
2.1.1 DISPOSITIVO DE ADQUISICIÓN DE IMAGEN EMPLEADO	17
2.1.1.1 Formato elegido	18
2.1.1.2 Resolución elegida	19
2.1.1.3 Temporización	20
2.1.1.3.1 Manejo de los datos mediante memoria FIFO	22
2.1.2 DISPOSITIVO DE VISUALIZACIÓN DE IMAGEN EMPLEADO	25
2.1.2.1 Interfaz 8080 paralela.....	25
2.1.2.1.1 Manejo de la interfaz 8080 mediante FSMC.....	26
2.1.2.2 Configuraciones del integrado SSD1289.....	28
2.1.2.3 Lámina táctil	29
2.1.2.3.1 Manejo de la lámina táctil mediante el circuito integrado XPT2046	30
2.2 PROCESAMIENTO DE LA IMAGEN	31
2.2.1 SEGMENTACIÓN DE UNA IMAGEN DE COLOR	31
2.2.1.1 Umbral de un color en el espacio RGB	32
2.2.1.1.1 Espacio RGB normalizado	33
2.2.2 DESCRIPCIÓN DE LA REGIÓN DE INTERÉS	35
CAPÍTULO 3	38
PROGRAMACIÓN DEL APUNTAMIENTO Y DE LOS ALGORITMOS DE CONTROL	38
3.1 CONFIGURACIÓN FÍSICA DE LA PLATAFORMA	38
3.1.1 CONFIGURACIÓN DE LOS SERVOMOTORES	39
3.2 LAZO DE CONTROL	39
3.2.1 IMPLEMENTACIÓN DIGITAL DE UN CONTROLADOR PI	40
3.2.1.1 Cálculo de las constantes Kp y Ki	41
3.2.1.1.1 Análisis a lazo cerrado	41
3.2.1.2 Implementación del controlador en tiempo discreto.....	43
3.2.2 IMPLEMENTACIÓN DIGITAL DE FILTROS	44

CAPÍTULO 4	47
PRUEBAS Y RESULTADOS	47
4.1 TRANSMISIÓN SERIAL DE UNA IMAGEN	47
4.2 PRUEBAS DE DETECCIÓN Y LOCALIZACIÓN	48
4.2.1 Segmentación de la imagen ante distintos niveles de iluminación	48
4.2.2 Detección de objetos de distintos colores.....	50
4.2.3 Pruebas de localización	51
4.2.4 Pruebas del controlador y del apuntamiento.....	54
CAPÍTULO 5	57
CONCLUSIONES Y RECOMENDACIONES	57
5.1 CONCLUSIONES Y RECOMENDACIONES	57
REFERENCIAS BIBLIOGRÁFICAS	59
ANEXOS	62
MANUAL DE USUARIO	62
CARACTERÍSTICAS DEL EQUIPO	67

RESUMEN

En los últimos años en diversos ámbitos como el industrial, militar, científico y civil han optado por hacer uso de sistemas de visión artificial en los cuales se pueda reconocer la existencia de objetos, su localización, entre otras características; la mayor parte de estos sistemas necesitan de un computador personal donde se ejecuta el software que procesa los datos de imagen, lo cual limita las capacidades del sistema. Aplicaciones tales como sistemas vehiculares no tripulados, o robots autónomos presentan limitaciones tanto de: espacio, consumo, robustez y de peso; haciendo que el uso de un computador personal sea impráctico, o requiriendo complejos y costosos métodos para transmitir la imagen hacia una estación fija que procese los datos de visión y retransmita los datos interpretados.

El desarrollo propuesto pretende dar solución a estas dificultades mediante un sistema compacto, ligero, de bajo consumo, en el cual ya esté incluido: la etapa de adquisición de imagen, interfaz de usuario y el procesamiento de los datos de imagen. Está destinado a aplicaciones que necesiten implementar visión artificial para detección, localización y apuntamiento de un objeto de color y que tengan fuertes limitantes en cuanto al tamaño, consumo y costo.

PRESENTACIÓN

El presente proyecto hace uso de un microprocesador que se comunica con una cámara de video, las imágenes procedentes son procesadas con el fin de detectar, y localizar un objeto de color que es elegido mediante la ayuda de una pantalla táctil, posteriormente mediante la acción de dos servomotores se realiza el apuntamiento, proceso en el cual la cámara ubica al objeto en el centro de su rango de visión.

Capítulo 1, se describe la composición de imágenes, los dispositivos que se usan para su captura digital. Se explica también las etapas de procesamiento por las que deben pasar para poder extraer datos relevantes en las mismas. De esta manera se pretende dar a entender el proceso de digitalización de la imagen. Finalmente se menciona el microcontrolador que se va a emplear de acuerdo a las capacidades y limitantes que se presentan.

Capítulo 2, se explica la metodología para realizar la adquisición y visualización de la imagen, los dispositivos que están involucrados en dicho proceso, como se comunican con el microcontrolador, sus características y las configuraciones que se les da para realizar este propósito.

Capítulo 3, en este capítulo se detalla la ubicación física de la cámara, la configuración de la plataforma, y principalmente la implementación del controlador que comanda los servomotores.

Capítulo 4, se presenta los resultados obtenidos a través de distintas pruebas prácticas ejecutadas.

Capítulo 5, se plantean conclusiones y recomendaciones basadas en los resultados y las experiencias conseguidas en el desarrollo de este proyecto.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se describe la composición de imágenes, los dispositivos que se usan para su captura digital, como se representa el color, y la formación de una imagen. Además se explica las etapas de procesamiento por las que deben pasar para poder extraer datos relevantes en las mismas. De esta manera se pretende dar a entender el proceso de digitalización de la imagen, y tener una mejor noción de los requerimientos necesarios para que un microcontrolador pueda procesar imágenes. Finalmente se menciona el microcontrolador que se va a emplear de acuerdo a las capacidades y limitantes que se presentan.

1.1 VISIÓN COMPUTARIZADA

La visión computarizada tiene como objetivo el extraer e interpretar información a partir de imágenes digitalizadas las cuales pueden ser proporcionadas en tiempo real, o bien pueden ser obtenidas de una base de datos; no obstante es común que un sistema básico de visión artificial se componga por dos etapas: obtención de imágenes y procesamiento de la imagen. Las mismas se describen a continuación.

1.1.1 CAPTURA DIGITAL DE LA IMAGEN

El dispositivo más básico para obtener información lumínica es llamado fotosito o foto-sensor, éste modifica una propiedad eléctrica de acuerdo al nivel de iluminación que incide sobre su superficie.

En el caso de una cámara digital se emplea un lente convergente que proyecta la luz sobre una superficie rectangular conformada por una matriz de foto-sensores.

1.1.1.1 Foto-sensores

Los foto-sensores están ordenados en filas y columnas de manera que cada uno representa un pixel que tendrá la información de intensidad lumínica. En cada

fotosito incide una distinta iluminación, en conjunto forman una imagen en una escala de grises.

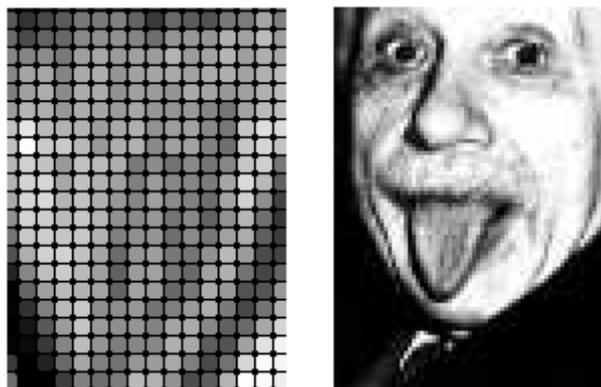


Figura 1.1 (a) Representación de la intensidad de iluminación en cada fotosito, (b) Imagen completa [1].

La fotografía de la Figura 1.1(a) es una composición de varios pixeles, en esta vemos unos pocos puntos con distintas intensidades que parecen formar una imagen, a la derecha en la Figura 1.1(b) apreciamos la imagen completa y como esta escala de grises va tomando una forma más coherente.

Existen dos clases principales de sensores empleados en cámaras digitales según el tipo de elemento fotosensible, las CCD y las APS.

1.1.1.1.1 Dispositivo de carga acoplada (CCD)

Consiste en un circuito integrado en cuyo interior reside un grupo de capacitores (ver la Figura 1.2) los cuales son descargados antes de empezar la adquisición, al recibir luz en la capa de semiconductor-óxido-metal (MOS) se produce una conversión de fotones en carga debido al efecto fotoeléctrico. Posterior a la exposición un circuito de control transfiere la carga de un capacitor a otro (de manera que operan como un registro de desplazamiento); el último capacitor está conectado a un amplificador de voltaje conectado al final de la fila de capacitores. La señal de voltaje es almacenada en una memoria para su posterior utilización.

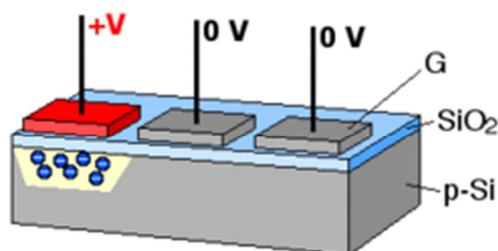


Figura 1.2 Representación de la carga y de los capacitores adyacentes en la celda CCD [2].

1.1.1.1.2 Sensor de pixel activo (APS)

Mayormente conocidos como tipo CMOS debido a que los transistores empleados en el sensor son de dicha tecnología. Se conforma por un circuito integrado en el cual se ubica un arreglo de pixeles bidimensional en filas y columnas, cada pixel incluye un foto-diodo el cual se polariza en reversa y por tanto presenta una corriente de fuga; al incidir un fotón en el semiconductor esta corriente aumenta en magnitud; un transistor actúa como buffer permitiendo la lectura del dato de voltaje sin afectar la carga eléctrica en el foto-diodo, otro transistor conecta este terminal a un bus correspondiente a la columna, esto permite que sólo una fila a la vez acceda al bus de columna; posteriormente un último transistor de reset vuelve a polarizar inversamente al foto-diodo, dejándolo así listo para una nueva captura de luz. Adicionalmente difiere de un sensor CCD al poseer un amplificador (buffer) ubicado en cada uno de los pixeles.

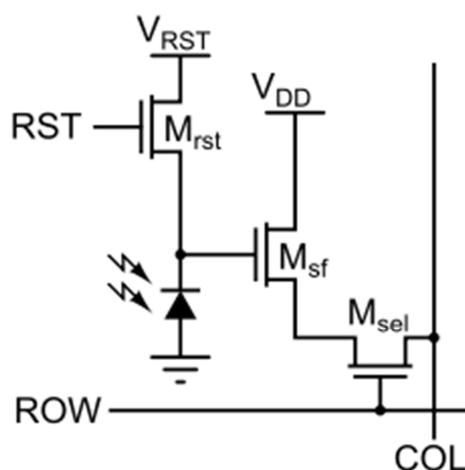


Figura 1.3 Bloque 3T muestra el fotodiodo como elemento fotosensible, y 3 transistores que realizan las funciones descritas anteriormente [3].

La configuración descrita se conoce como celda 3T y está representada en la Figura 1.3, pero existen otras configuraciones como la 4T, 5T y 6T que añaden más transistores de tipo CMOS.

1.1.1.2 Detección de colores

Los elementos fotosensibles actúan únicamente ante la intensidad lumínica, para discernir datos de color es necesario que la luz atraviese un filtro el cual selecciona que color llega a la superficie del fotosito; el más común es llamado filtro de Bayer ilustrado en la Figura 1.4(a), en la Figura 1.4(c) se aprecia que por cada pixel del mismo se deja pasar los colores rojo en un 25%, azul en un 25%, y verde en un 50% (debido a que el ojo humano percibe de mayor manera este color). Ubicado directamente bajo el color filtrado se encuentra el fotosito (ver Figura 1.4(b)), 4 de los mismos conforman un pixel con la información del color compuesto por los 3 colores básicos rojo, verde, y azul.

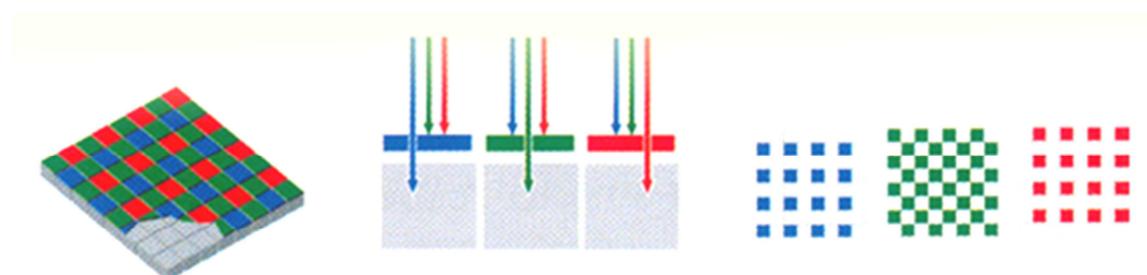


Figura 1.4 (a) Filtro de Bayer sobre una matriz detectora, (b) Representación de cómo se filtra cada color hacia el elemento detector, (c) Representación de los colores azul, verde, y rojo, que inciden en la matriz de detección [4].

1.1.1.2.1 Espacio de colores

El ojo humano registra la información de colores basándose en tres características: la intensidad lumínica, la proporción entre rojo-verde, y la proporción entre azul-amarillo. A partir de esta información nuestro cerebro interpreta los colores. Debido a que se emplea 3 variables para obtener toda la información para representar un color, a menudo se emplea cada variable como un vector formándose así un espacio tridimensional llamado espacio de colores. Existen diversas representaciones matemáticas de tres dimensiones para el color como son: RGB, LAB, HSV, HSL.

RGB: en este espacio el color es resultado de la adición de tres colores básicos que son el rojo, verde y azul. La intensidad de cada uno determina cuán oscuro o claro es el color resultante.

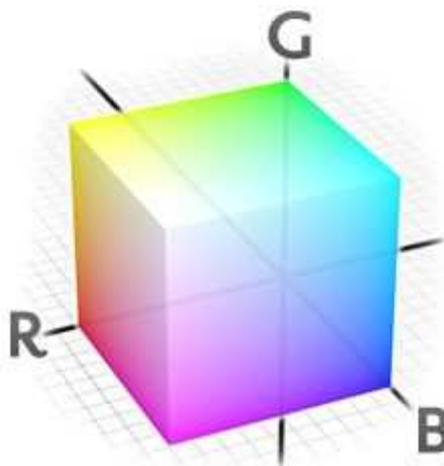


Figura 1.5 Representación del espacio de colores RGB[5].

Cada uno de los colores básicos es representado como un vector en un subespacio vectorial ortogonal representado gráficamente en la Figura 1.5. Se puede apreciar que si los 3 vectores están cercanos a su máximo valor tendremos un color más claro, si los valores son más bajos se tendrá un color más oscuro. Si una componente es predominante se tendrá un color más saturado, por ejemplo el vector rojo tiene su valor máximo y el resto el mínimo, esto da como resultado un color rojo bien definido.

El espacio RGB es el usado en las cámaras digitales las cuales poseen en cada pixel un sensor para rojo, otro sensible al verde y un último destinado al color azul, de esta manera por cada uno de los 3 colores básicos, se puede representar todos los colores.

De igual manera se usa este espacio de color para pantallas LCD, como apreciamos es un color aditivo, por cada pixel se tiene un emisor de luz rojo, otro verde y uno azul, como resultado se tiene una luz que es la suma de los colores.

HSV: el espacio RGB tiene ciertas limitaciones, no hace diferenciación entre el dato de iluminación, ni de matiz de color. El espacio HSV (ver la Figura 1.6) se desarrolló precisamente para obtener esos datos, en éste se representa el color en tres términos: matiz (H), saturación(S), y valor (V).

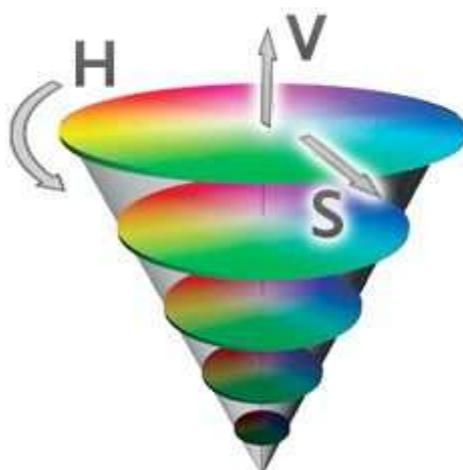


Figura 1.6 Representación del espacio HSV [5].

El matiz es un valor de ángulo que aumenta en sentido anti horario, empezando por el rojo, pasando por el verde, luego el azul hasta volver al rojo; el matiz determina el tono del color indiferente de la iluminación y la saturación del color, es decir si un color es amarillo claro, el dato de matiz es el mismo que si fuera amarillo oscuro.

El valor determina cuan oscuro es el color, y la saturación cuan puro es el color.

1.1.1.3 Formatos de color

Para la representación digital de un color primero se debe definir en uno de los espacios de color. En el caso de el espacio RGB se asigna una cantidad de bits por cada uno de los tres colores, existen varias representaciones del espacio RGB, algunas de ellas se ilustran en la Figura 1.7.

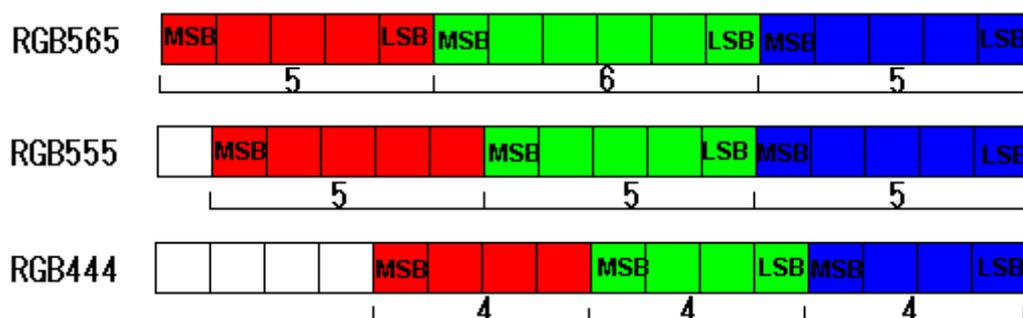


Figura 1.7 Varios formatos de color RGB, se aprecia que la numeración corresponde a la cantidad de bits usadas respectivamente.

Cada una de las combinaciones posibles de los bits nos dan la cantidad de colores que se pueden representar, de acuerdo a la fórmula:

$$C = 2^{nR+nG+nB}$$

De esta manera el formato RGB565 es de 2^{5+6+5} colores, razón por la que también es referido como formato de 65k (65536 colores posibles); además se observa que emplea totalmente los 16bits por lo cual es comúnmente llamado de 16bits. Es notable que RGB565 usa mayor cantidad de bits para definir el color verde, esto se debe a que el ojo humano es más sensitivo a este color.

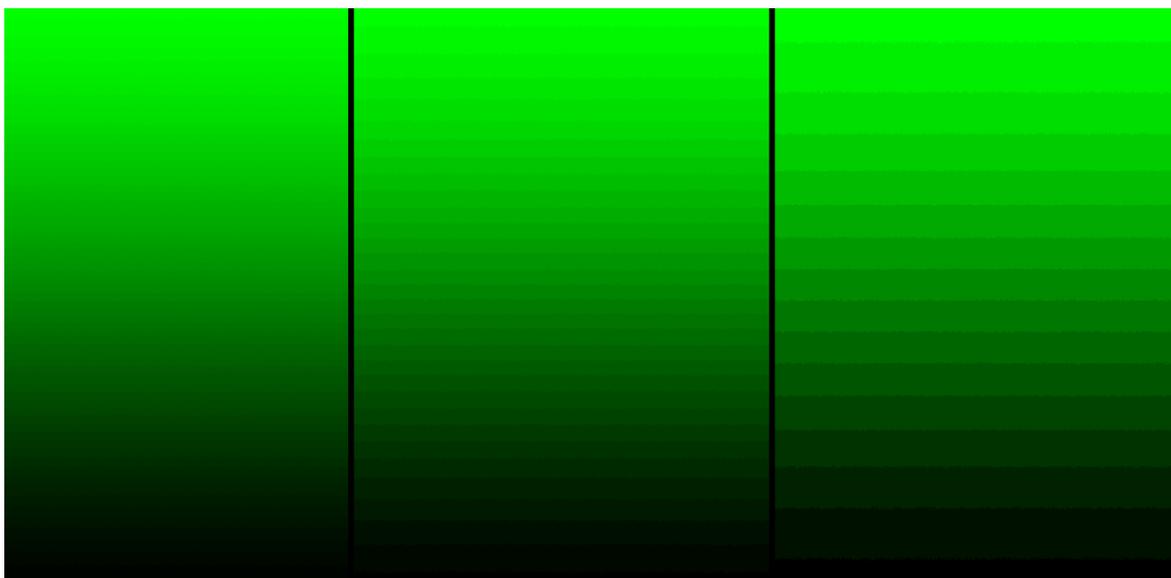


Figura 1.8 Gradiente del color verde representado en RGB565 (a la izquierda), RGB555 (en el centro), RGB444 (a la derecha).

En la Figura 1.8 se puede apreciar las diferencias de los distintos gradientes de color, por ejemplo cuando se tiene 6 bits para representar el color verde los cambios del gradiente son mucho más imperceptibles que en el caso de la representación de 4 bits.

1.1.1.4 Formación de la imagen

La representación digital de una imagen puede ser definida como un conjunto de varios pixeles, los mismos son la unidad de espacio más pequeña en la imagen y cada uno de ellos tiene la información de color en ese punto de un plano bidimensional. Los pixeles están ordenados en filas y columnas, a manera de una matriz rectangular. El número de columnas multiplicado por el número de filas nos da como resultado la resolución de la imagen.

Se conforma de esta manera un rectángulo donde se forma la imagen vista en ese instante de tiempo en un término conocido como fotograma o *frame*. La presentación de varios *frames* consecutivos representan un video.

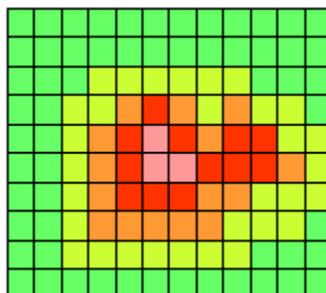


Figura 1.9 Imagen conformada por 12 columnas y 10 filas.

Cada uno de los pixeles mostrados en la Figura 1.9 contienen su propia información de color y están encerrados en un recuadro, como se puede apreciar forman una matriz de 12×10 , en este caso tenemos una imagen de 120 pixeles de resolución.

1.1.2 PROCESAMIENTO

Una vez obtenida la imagen es necesario extraer los datos relevantes de la misma, la imagen normalmente posee una cantidad muy grande de información y se debe extraer características tales como la detección de un objeto, ubicación del mismo, cantidad de objetos, entre otros dependiendo de los requerimientos del sistema de visión artificial. El proceso se puede dividir en: pre-procesamiento, segmentación, descripción, reconocimiento e interpretación.

1.1.2.1 Pre-procesamiento

Las imágenes adquiridas digitalmente están expuestas a ruido, a variaciones de iluminación, a bajo contraste, entre otras características no deseadas.

El pre-procesamiento puede corregir en parte estos problemas, a menudo las cámaras digitales vienen incorporadas con sistemas de pre-procesamiento para mejorar la imagen y resaltarla para lo cual se aumenta el contraste, se reduce el ruido y también tienen sistemas para corregir problemas asociados con su propia óptica, por ejemplo un sistema automático corrige los problemas introducidos por

el lente de la cámara como son la distorsión geométrica que produce una imagen cuyos bordes están curvados (visto en la Figura 1.10(b)), o el efecto viñeta que es el oscurecimiento de la imagen en las esquinas (visto en la Figura 1.10(a)).



Figura 1.10 (a) Imagen con efecto viñeta, (b) Imagen con distorsión geométrica [6].

1.1.2.2 Segmentación

En esta etapa se divide la imagen en secciones que tienen diferenciación con el entorno. Estas secciones agrupan características de interés de la imagen, y descartan el procesamiento de datos que no son necesarios.

Para realizar la segmentación se puede emplear tres criterios:

- Similitud: los píxeles son similares en color, iluminación, matiz, etc.
- Conectividad: los píxeles similares se encuentran adyacentes el uno del otro, y hay regiones enteras de éstos.
- Discontinuidad: los píxeles adyacentes que tienen distinto color, nivel de gris, etc. determinan una forma geométrica u objeto definido.

La segmentación nos da como resultado un contorno o una región de píxeles con las características deseadas.

1.1.2.2.1 Detección de bordes

Para detectar bordes a menudo se emplean algoritmos sobre un píxel y sus adyacentes, si el nivel de uno de éstos tiene una variación muy alta respecto al del centro entonces éste puede ser un posible borde.



Figura 1.11 (a) Imagen original, (b) Imagen procesada para detectar bordes.

En la Figura 1.11 se aprecia la detección de bordes realizada por el software Adobe Photoshop CS3 y su posterior filtro para obtener una imagen en escala de grises.

1.1.2.2.2 Detección de regiones de interés

Las regiones de interés de una imagen se pueden detectar mediante el uso de umbrales: se define un umbral, ya sea de nivel de gris, o de color. Los píxeles que entren dentro del umbral tendrán una información de uno lógico, los que estén fuera del umbral serán un cero lógico.



Figura 1.12 (a) Imagen original, (b) Imagen binarizada, (c) Nivel de umbral empleado.

Se aprecia en la Figura 1.12 que la binarización asigna un uno lógico a los píxeles cuyo nivel de gris es menor a 122 (en una escala de 0 a 255), estos mismos son representados en verde.

1.1.2.3 Descripción y Reconocimiento

Realizada la segmentación, se debe determinar si el objeto detectado en la segmentación es el objeto de interés. Para esto se puede usar criterios tales como: forma, dimensiones, similitud con patrones en la base de datos, separación de regiones con poco interés o de formas no concernientes.

La descripción etiqueta al objeto con las principales características que se necesiten en el estudio, como puede ser ubicación, centro de masa, forma, y características específicas sujetas al estudio como por ejemplo si el objeto es un engrane se puede identificar número de dientes, radio interno, externo, radio del eje, etc.

El reconocimiento clasifica al objeto en base a lo que se espera encontrar; citando el ejemplo del engrane, el reconocimiento identifica al objeto como engrane defectuoso, engrane aceptable, o no es un engrane.

1.1.2.4 Interpretación

Se usa los datos del objeto reconocido para efectuar acciones por ejemplo emplear un actuador lineal para empujar a una fruta clasificada como defectuosa fuera de la banda transportadora.

1.2 USO DE MICROCONTROLADORES PARA PROCESAMIENTO DE IMAGEN

Si se pretende hacer uso de un microcontrolador para realizar las tareas requeridas por el sistema de visión artificial necesitamos que éste sea capaz de:

- Comunicarse con el equipo que adquiere la imagen.
- Disponer de una memoria para almacenar las grandes cantidades de información que representan las matrices de píxeles.
- Poder acceder de manera rápida a esta memoria para procesar cada píxel.

- Realizar las acciones a partir de los datos interpretados.

Y de manera opcional en un sistema de visión el usuario desea inspeccionar la imagen adquirida, los datos procesados, entre otras tareas por lo cual el microcontrolador también puede incorporar las siguientes etapas:

- Poder manejar una interfaz de usuario y comunicarse con una pantalla para el despliegue de imagen y datos.
- Realizar todas las tareas pertinentes en un tiempo apropiado para el sistema y para el usuario.

1.2.1 FAMILIA DE MICROCONTROLADORES ARM CORTEX-M

Estos microcontroladores emplean procesadores diseñados por ARM Holdings los cuales ofrecen características de alto desempeño; ARM Holdings no manufactura microcontroladores físicamente, en lugar de ello vende licencias para fabricación a empresas de manufactura de circuitos integrados (tales como ST, Atmel, Texas Instruments, entre otras) a las cuales provee del procesador ARM mediante diseños integrables en hardware y los derechos de venta física del CPU fabricado.

Todos los microcontroladores de este tipo tienen una arquitectura de 32 bit RISC (Reduced Instruction Set Computing) la cual se basa en emplear un número reducido de instrucciones de alta eficiencia en lugar de una gran cantidad de instrucciones especializadas. De esta manera la mayoría de instrucciones se realiza en uno o dos ciclos de máquina.

1.2.1.1 Ventajas frente a los microcontroladores de 16, y 8 bits.

De manera general podemos mencionar que las nuevas arquitecturas de microcontroladores de 32 bits permiten manejar de una mejor manera los distintos tipos de datos de mayor longitud (que son ampliamente usados en programas complejos) como son el entero corto (16 bits), largo (32 bits), número flotante de simple precisión (32 bits), de doble precisión (64 bits) entre otros.

Si tenemos una arquitectura de 8 bits podemos tener por ejemplo un microcontrolador Atmega de la familia Atmel realizando una simple multiplicación de dos números enteros (de 16bits) con signo, la ALU de este microcontrolador no

puede realizar operaciones aritméticas directamente para estos números ya que su bus está limitado a 8 bits, por lo tanto para realizar la operación se necesitan algoritmos que realicen multiplicaciones y sumas parciales sucesivas (todas de 8 bits) hasta obtener el resultado, para este caso se necesitan de 15 a 19 ciclos de reloj.[7]

En comparación una arquitectura de 32 bits puede realizar operaciones lógicas y aritméticas ya sea de 8, 16 o 32 bits de manera directa en la ALU (Arithmetic Logic Unit), lo que en conjunto con frecuencias de reloj mucho más altas (desde 48MHz hasta los 180MHz) les da una velocidad de procesamiento mucho mayor a los microcontroladores de 8 bits.

Además cabe señalar que la mayoría de microcontroladores de 32 bits poseen hardware especializado en operaciones matemáticas de multiplicación, división, e inclusive operaciones para números de punto flotante.

En cuanto a memoria los microcontroladores de 8 bits y 16 bits poseen hasta 64Kbytes de memoria direccionable, de manera que si el código sobrepasa los 64Kbytes algunos microcontroladores implementan una segmentación de memoria por lo que siempre tienen que ejecutar un algoritmo de direccionamiento especial si el código ha sobrepasado los 64kbytes lo cual reduce la eficiencia [8].

En contraste un microcontrolador de 32 bits usualmente posee tamaños de memoria que empiezan desde los 16Kbytes hasta los 2Mbytes [9], esto permite el alojamiento de programas mucho más grandes y complejos como por ejemplo comunicarse por Ethernet, mostrar datos en una pantalla a color, comunicarse con memorias externas de gran capacidad, controlar motores, etc. Todo esto ha provocado que un microcontrolador de 32 bits usualmente esté equipado con una gran cantidad de periféricos para manejar dispositivos externos de manera eficiente (pantallas LCD, memorias externas, etc.).

1.2.1.2 Microcontrolador empleado

De manera concreta se va a utilizar el microcontrolador STM32F103VCT que es de arquitectura Cortex-M3 fabricado por STMicroelectronics, el mismo que posee muchas características; se listan las que tienen mayor impacto en este proyecto en particular [10]:

- Frecuencia máxima de operación: 72MHz.
- Multiplicación en un sólo ciclo.
- División mediante hardware.
- 256Kbytes de memoria Flash, y 64Kbytes de SRAM.
- Interfaces de comunicación 3xSPI, 2xI2C, 5xUSART, USB, CAN, SDIO.
- 4xTimers de 16 bits cada uno con 4 PWM/OC/IC.
- Interfaz paralela para pantallas LCD (8080/6800).

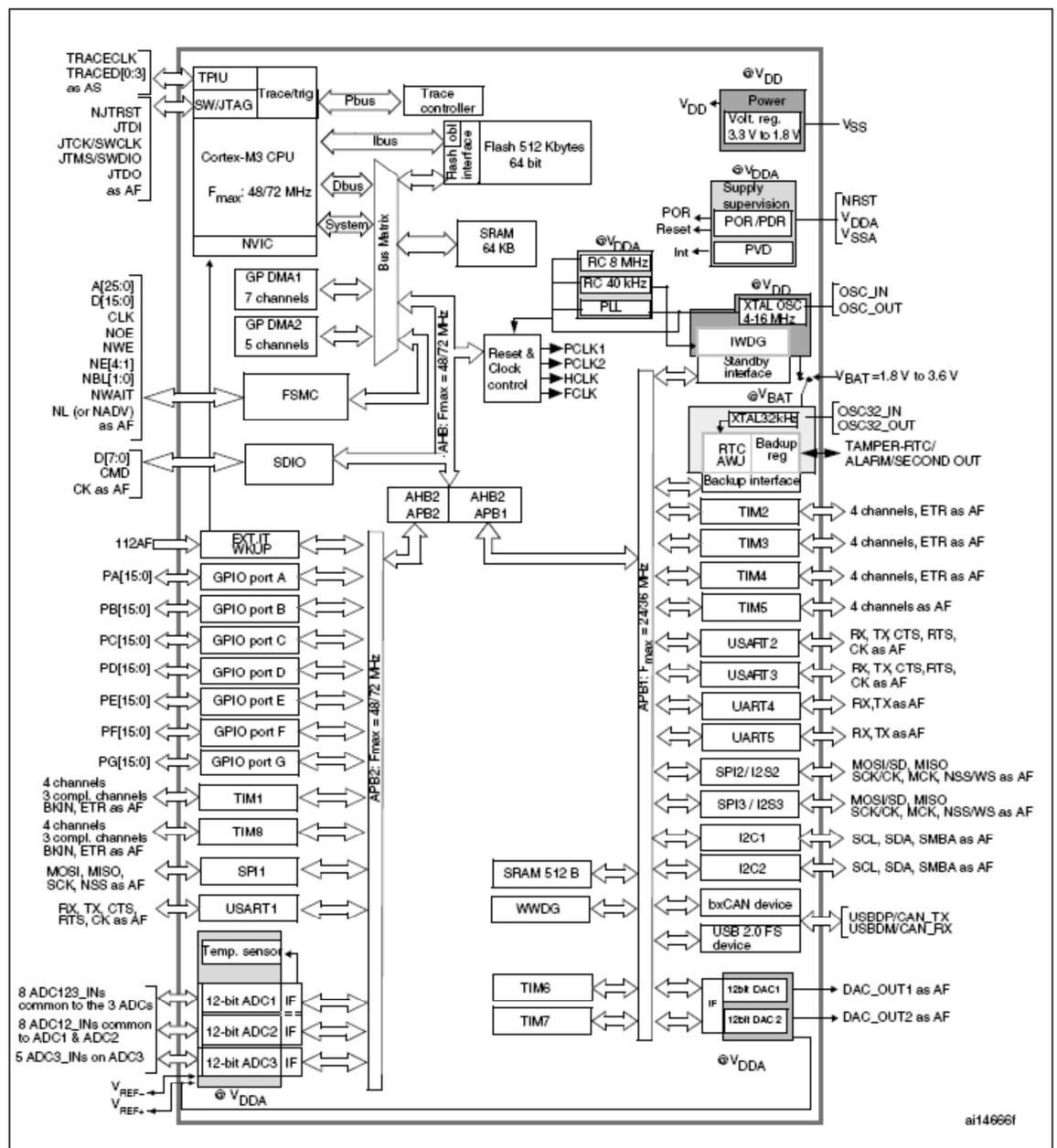


Figura 1.13 Diagrama de bloques de la línea STM32F103xC, STM32F103xD Y STM32F103xE [10].

De manera general se puede decir que este microcontrolador puede realizar las tareas básicas y necesarias como comunicarse con una cámara, procesar datos de manera rápida, y mostrarlos mediante una pantalla LCD. También posee Timers suficientes para controlar los 2 servomotores que se emplearán en la plataforma para seguir el objeto deseado.

1.2.1.2.1 Tarjeta de desarrollo

El microcontrolador mencionado viene incorporado en la tarjeta de desarrollo HY-Smart STM32 (ver la Figura 1.14) la cual además posee zócalos para conectar directamente una pantalla LCD y una cámara OV7725, dispositivos cuyo funcionamiento se detallará en el Capítulo 2.



Figura 1.14 Tarjeta de desarrollo HY-Smart STM32 con pantalla y cámara incorporadas.

1.2.1.3 Software de Programación

La compañía ARM posee el software KEIL MDK-ARM el cual es un entorno de desarrollo de dispositivos basados en procesadores Cortex-M, Cortex-R4, ARM7 y ARM9. Las herramientas incluyen un compilador de C y de C++, y IDE (entorno integrado de desarrollo) llamado uVison4 el cual también sirve para depurar los programas (debugging) y se muestra a continuación en la Figura 1.15.

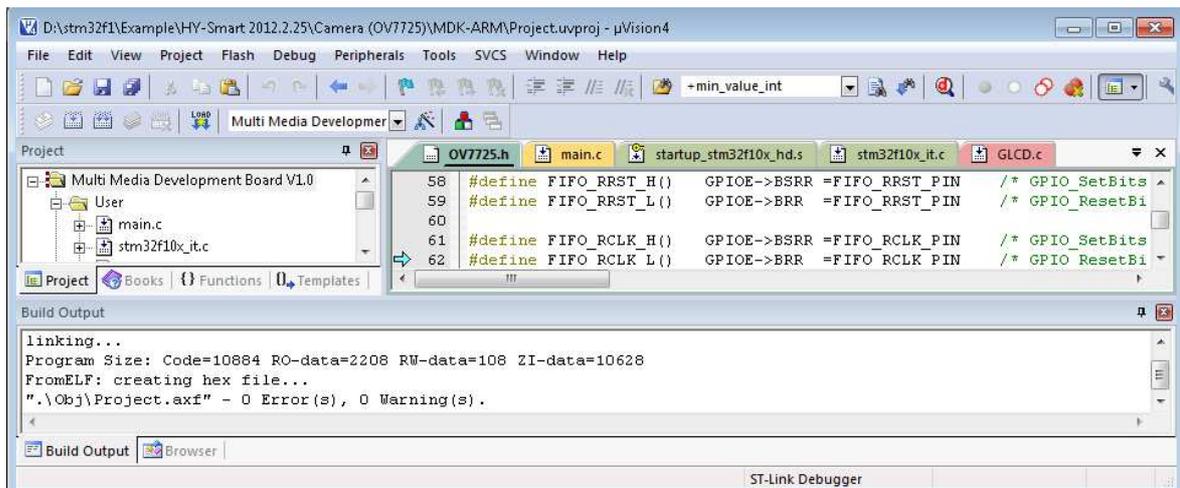


Figura 1.15 Entorno integrado de desarrollo uVision4.

1.2.1.3.1 Depuración de programas para el microcontrolador empleado

La depuración de programas es de gran utilidad, se puede detener el programa en un punto específico, realizar la ejecución paso a paso, e incluso se puede observar el valor de distintas direcciones de memoria en tiempo real. En el caso específico de los microcontroladores fabricados por ST se tiene una herramienta llamada ST-LINK/V2 la cual sirve tanto para programar como para depurar programas y es compatible con el software KEIL MDK-ARM.

CAPÍTULO 2

DESARROLLO DEL SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE IMAGEN

En este capítulo se explica la metodología para realizar la adquisición y visualización de la imagen, los dispositivos que están involucrados en dicho proceso, cómo se comunican con el microcontrolador, sus características y las configuraciones que se les da para realizar este propósito.

2.1 METODOLOGÍA DEL SISTEMA DE ADQUISICIÓN Y VISUALIZACIÓN DE LAS IMÁGENES

La adquisición de la imagen se realiza a través de una cámara de video, ésta envía constantemente datos de imagen de manera paralela hacia una memoria FIFO externa, la cual almacena estos datos. El microcontrolador puede leer los datos en la memoria FIFO bajo demanda, lo que le permite acceder a ellos en el momento que lo requiera.

A medida que el microcontrolador lee cada pixel, también lo muestra en la pantalla LCD, tanto la memoria FIFO así como el LCD poseen autoincremento, razón por la cual se puede repetir el proceso de lectura de un pixel y mostrarlo en la pantalla durante el número de pixeles que posea la imagen.

2.1.1 DISPOSITIVO DE ADQUISICIÓN DE IMAGEN EMPLEADO

Se emplea la cámara de video OV7725 (ver la Figura 2.1), la misma que es de tecnología CMOS y puede capturar hasta 60 imágenes por segundo, las principales características se listan a continuación:

- Voltaje de operación de 3.3V.
- Consumo activo de 120mA.

- Interfaz SCCB (Serial Camera Control Bus): esta interfaz es compatible con la interfaz I2C, llegando a emplear los mismos hilos SCL y SDA, mediante estos se envían los comandos de control de la cámara, se programa el tiempo de exposición, el formato de salida de la imagen, su resolución, entre otras características.
- Varios formatos de salida: GRB 4:2:2, RGB 565, RGB 555, RGB 444, YCbCr 4:2:2
- Varias resoluciones: VGA, QVGA, y cualquier tamaño elegible hasta 40x30
- Funciones de control de imagen automática: control de exposición automática (AEC), de ganancia (AGC), de balance (AWB), de nivel de negro (ABLC).
- Corrección de ruido, y de obscurecimiento por los lentes.



Figura 2.1 Cámara de video OV7725.

2.1.1.1 Formato elegido

La cámara usada tiene varios formatos de salida, la elección del formato con mayor cantidad de información es el RGB 565 y además éste puede ser reconocido directamente por la pantalla LCD empleada en este proyecto.

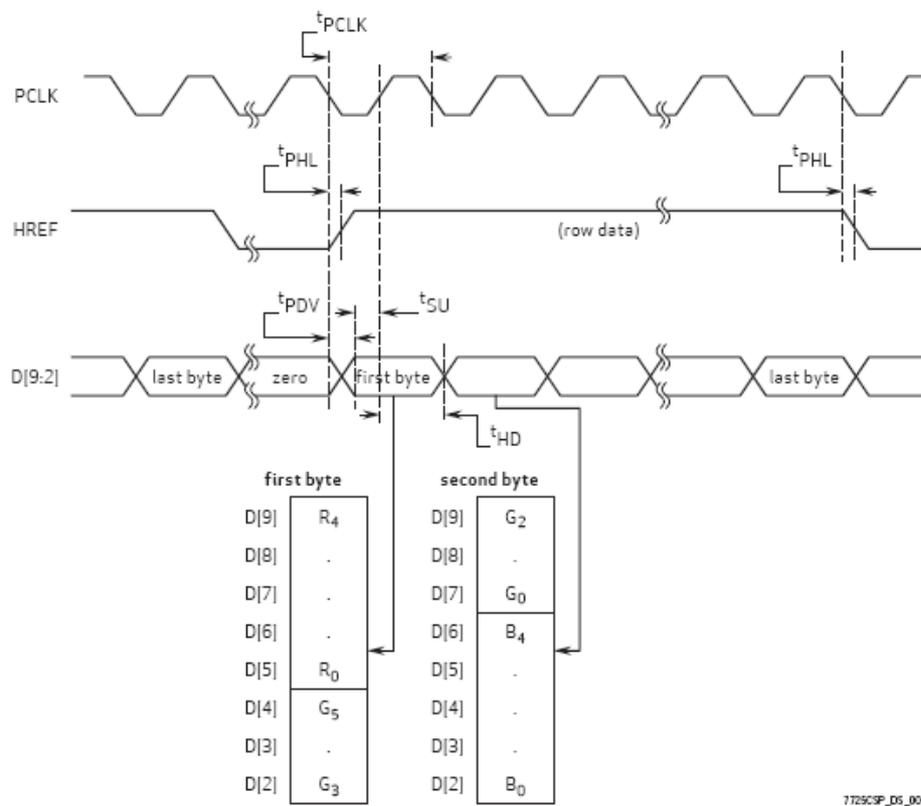


Figura 2.2 Temporización de la transmisión de datos para una fila completa en el formato RGB 565 [11].

Como se ilustra en la Figura 2.2 este formato envía 2 bytes de información por cada pixel, para que la información sea válida HREF tiene que estar en alto, esto indica el inicio de una fila de la imagen, a continuación se lee un byte en D[9:2] cada vez que PCLK pasa a uno lógico. En el segundo ciclo de PLCK se lee el segundo byte completando la información correspondiente a un pixel. Es de importancia indicar que PCLK siempre está activo aun cuando HREF está en bajo, pero no se debe tomar datos en esta condición.

Si el formato de salida de la cámara fuera YCbCr 4:2:2 se tendría que realizar una conversión hacia RGB565 por cada pixel para poder mostrarlo en la pantalla lo cual implica un indeseable incremento de tiempo en el procesamiento.

2.1.1.2 Resolución elegida

La mayor resolución posible por esta cámara es conocida como VGA (640x480), sin embargo la pantalla empleada es de 320x240 razón por la cual la resolución de salida de la cámara es QVGA (320x240), esto nos da como resultado 76800

pixeles por fotograma; una imagen de menor resolución implicaría una menor área para la detección del objeto, así como también una menor superficie de imagen mostrada en la pantalla.

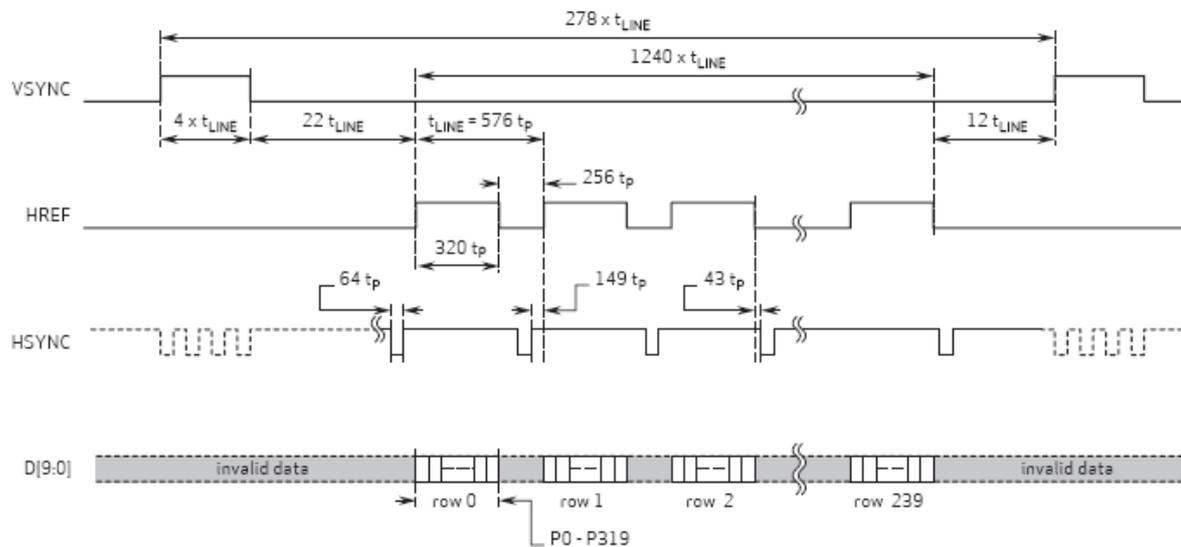


Figura 2.3 Temporización de la transmisión de datos para una imagen completa el formato QVGA (320x240), donde $t_p = 2 \times t_{pCLK}$ [11].

En la Figura 2.3 se indica la temporización que se emplea en el formato QVGA para enviar un fotograma completo; un cambio de VSYNC hacia uno lógico marca el inicio de un nuevo fotograma, posteriormente se produce su transición a cero lógico, la primera línea se transmite cuando HREF está en alto, una vez transmitidos los 320 pixeles correspondientes a una línea, HREF pasa a cero lógico y luego vuelve a su estado en alto para el inicio de la siguiente línea. El proceso se repite hasta la transmisión de las 240 líneas que comprenden un fotograma del formato QVGA.

2.1.1.3 Temporización

La cámara no posee osciladores internos, en lugar de ello recibe una señal externa de reloj a través de una entrada XCLK, a partir de esta señal, un prescalador interno y de un PLL (del inglés *phase locked loop*) se obtiene una señal de reloj de 10 a 48 MHz según la ecuación(2-1) [11].

$$fp = \frac{fx \times PLL}{(PRESC + 1) \times 2} = \frac{1}{tp_{CLK}} \quad (2-1)$$

Donde f_x es la frecuencia del reloj externo, PLL y PRESC es el valor asignado en los registros H0D y H11 respectivamente.

Para obtener una frecuencia máxima de trabajo de la cámara (de 48MHz) se ha optado por emplear una función especial del microcontrolador que permite asignar un pin como salida de reloj externo (MCO) que entrega una señal de 72MHz.

Dada esta configuración el PLL de la cámara debe tener un valor de multiplicación de 4 y el prescalador debe ser 2.

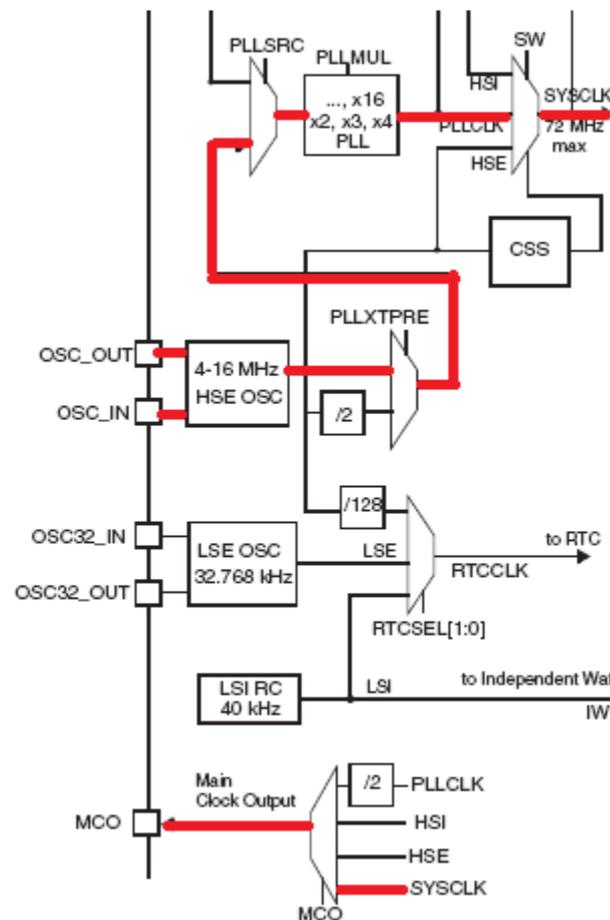


Figura 2.4 Diagrama parcial del microcontrolador STM32F103VCT, indicando la configuración del reloj del sistema (SYSCLK) y el reloj de salida (MCO) [10].

En la Figura 2.4 se puede observar la configuración del reloj SYSCLK del microcontrolador, para obtener la frecuencia máxima de 72MHz se debe emplear un oscilador externo (HSE) de 8MHz, éste pasa por un PLL que multiplica la frecuencia por 9; a través de un selector podemos dirigir la señal SYSCLK hacia una salida conectada a MCO.

Mientras exista una señal de reloj XCLK se tendrá una señal interna PCLK, es decir que el envío de bytes de datos es continuo; para que el microcontrolador reciba estos datos hay dos opciones:

- Esperar a que la señal VSYNC esté en nivel bajo y posteriormente recolectar los bytes de la fila cuando HREF esté en alto en cada periodo de PCLK por cada una de las 320 filas.
- Emplear una memoria externa que recolecte los datos mediante las señales de sincronización y esperar a que el microcontrolador acceda a cada uno de los datos.

Se aprecia que en la primera opción el microcontrolador está obligado a leer los datos de imagen cada vez que la cámara envía una línea de imagen; en la segunda opción la cámara trabaja normalmente enviando datos a una memoria externa, el microcontrolador accede a está en cualquier momento que lo requiera, éste es un mejor método para manejar las grandes cantidades de información que se tiene en un fotograma, además de no interrumpir periódicamente al microprocesador.

2.1.1.3.1 Manejo de los datos mediante memoria FIFO

Se emplea la memoria FIFO AL422B la cual posee 393216 bytes disponibles para almacenar datos. El espacio de memoria es suficiente para almacenar más de un fotograma ya que el formato RGB565 emplea 2 bytes de información de color por cada pixel, si se tiene en cuenta que un fotograma se representa mediante 76800 pixeles entonces podemos establecer que por cada fotograma se debe transmitir 153600 bytes de información.

Los tiempos de lectura y escritura de la memoria son de 20ns, es decir que puede operar a una frecuencia de 50MHz, la cámara está configurada para trabajar a 48MHz por lo que se tiene suficiente tiempo para realizar de manera íntegra la escritura y lectura de datos.

La memoria FIFO es la que se comunica directamente con la cámara como se indica en la Figura 2.5.

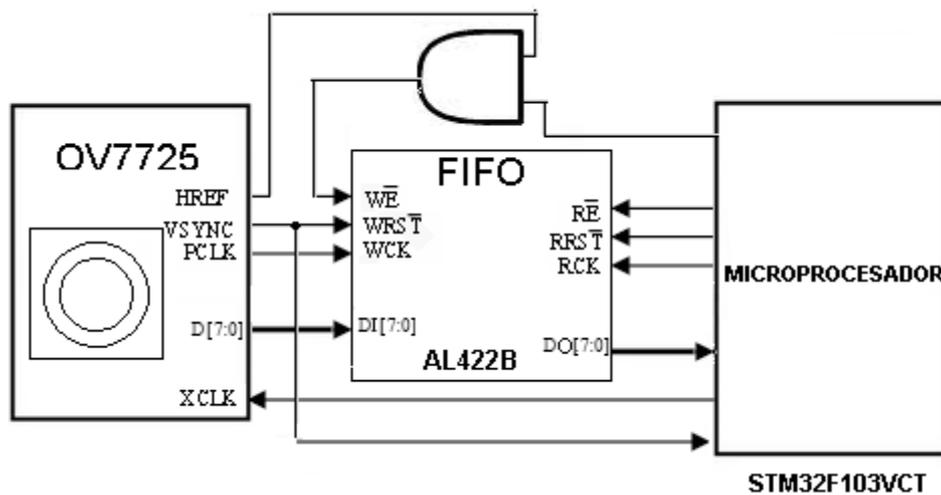


Figura 2.5 Diagrama de la comunicación entre la cámara OV7725 y el microcontrolador.

Escritura de datos: para escribir datos en la memoria se tiene 8 líneas de datos de entrada DI[7:0], y tres líneas de control: WCK, /WRST y /WE.

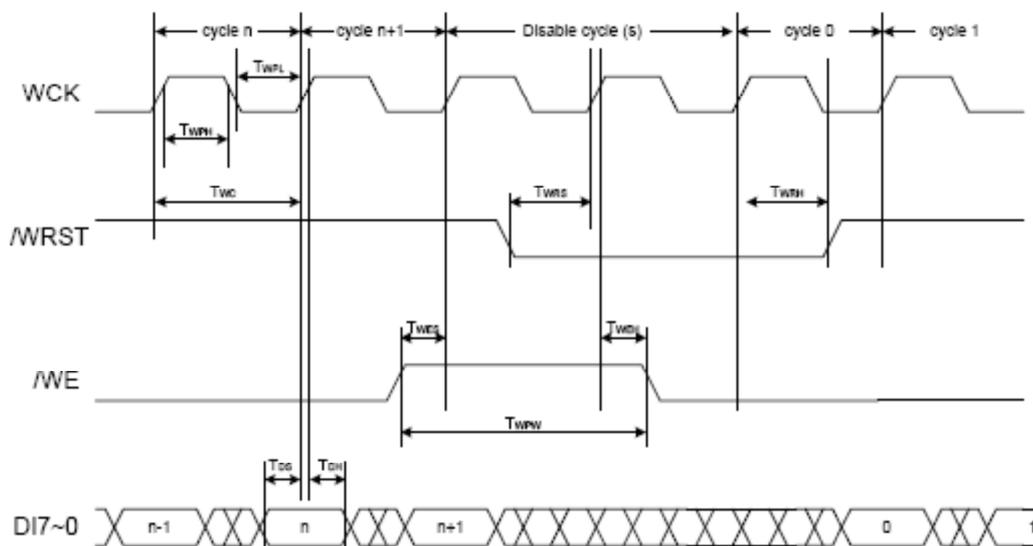


Figura 2.6 Temporización para realizar la escritura de datos en la memoria AL422B [12].

En la Figura 2.6 se indica las señales de control que se utilizan para la escritura de datos. La línea /WRST cambia a nivel alto para iniciar la escritura en la dirección cero (esto se realiza cuando VSYNC en la cámara cambia a nivel bajo, indicando el inicio del fotograma). Sólo se almacena datos cuando /WE está en

bajo, esta línea está en bajo cuando el microcontrolador lo ordena, con la ayuda de una compuerta lógica AND; también está en bajo cuando HREF está en cero lógico, de esta manera se evita capturar datos cuando HREF está en alto. Si /WE está en alto se almacena un byte cada que WCK cambia a nivel alto, la línea PCLK se conecta directamente a WCK y la dirección de la memoria tiene autoincremento.

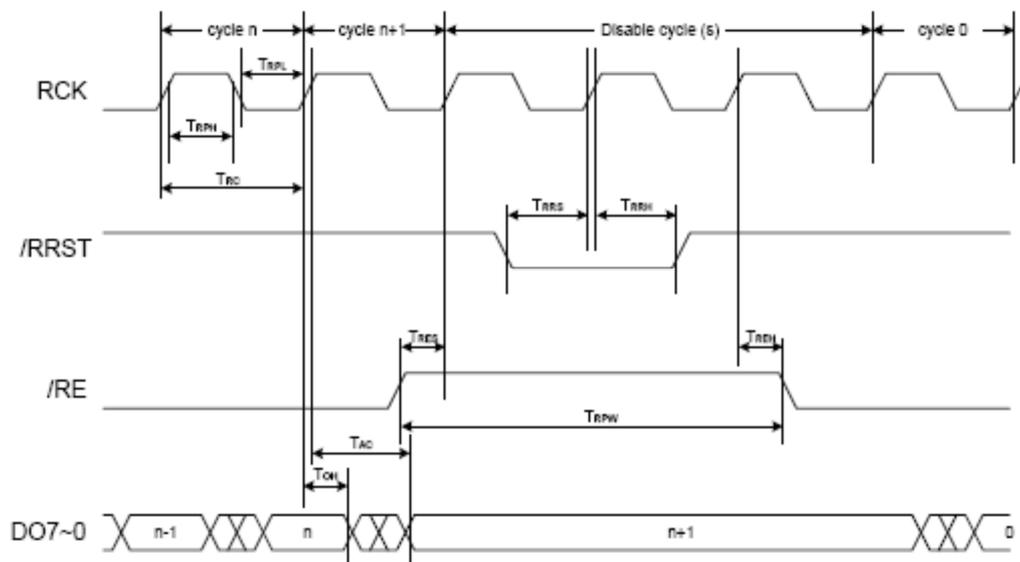


Figura 2.7 Temporización para realizar la lectura de datos en la memoria AL422B [12].

Lectura de datos: similar a la escritura, la memoria posee 8 líneas de datos de salida DO[7:0], y tres líneas de control: RCK, /RRST y /RE.

Las secuencias para la lectura se ilustran en la Figura 2.7. Cuando el microcontrolador lo requiera inicia la lectura del fotograma haciendo una transición hacia alto de la línea /RRST, el puntero del dato a leerse se ubica en la posición cero de la memoria FIFO; posteriormente pone en bajo la línea /RE y envía una transición hacia alto RCK cuando quiera leer un byte de datos, la dirección se autoincrementa hasta que se lee todos los bytes del fotograma (en este caso 153600 bytes).

Es de importancia señalar que el valor de las direcciones de escritura y de lectura son independientes por lo cual la lectura de datos se puede realizar en un tiempo distinto al de la escritura sin que ambas acciones tengan que estar sincronizadas.

2.1.2 DISPOSITIVO DE VISUALIZACIÓN DE IMAGEN EMPLEADO

Para la visualización de imagen se emplea una pantalla TFT LCD de 320x240 pixeles la misma que es controlada por un circuito integrado SSD1289 con 172800 bytes de GDDRAM (Graphic Display Data RAM), éste es el que alimenta a la pantalla LCD a través de la línea VCOM, opera 320 líneas correspondientes a las puertas (Gate) de cada uno de los transistores y 720 (son 3 colores por cada pixel de una fila de 240) para las compuertas fuente (Source).

El integrado también se comunica con el microcontrolador (ver la Figura 2.8), lo hace mediante la interfaz 8080 paralela. Además es de importancia señalar que la pantalla viene incorporada con una lámina táctil tipo resistiva de 4 líneas conectada al circuito integrado XPT2046 que le permite ubicar la posición de un punto único de presión en la pantalla y comunicarlo al microcontrolador mediante interfaz SPI.

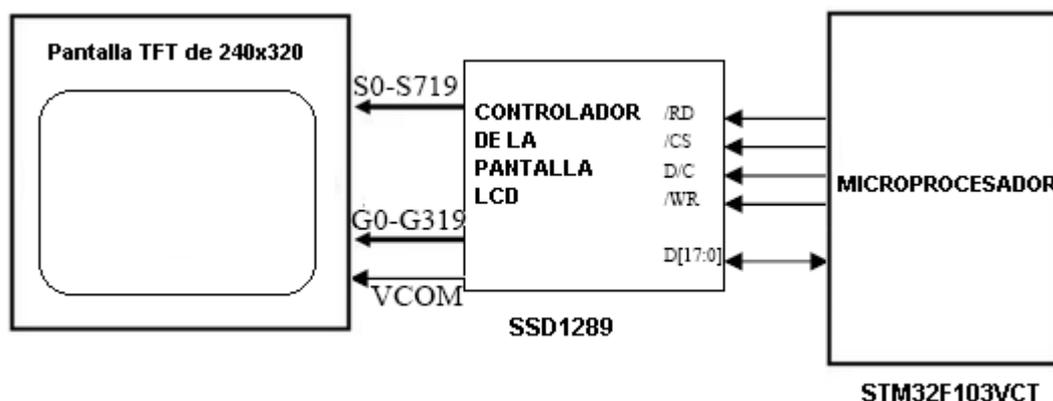


Figura 2.8 Diagrama de la comunicación entre la pantalla y el microcontrolador.

2.1.2.1 Interfaz 8080 paralela

La interfaz es también conocida como Intel 8080, consiste en 16 líneas bidireccionales de datos y 4 líneas de control: /WR, /RD, D/C y /CS. Las líneas de control tienen la función de señal de escritura (/WR), señal de lectura (/RD), realizar acciones ya sea sobre los datos o en el control de registros (D/C), la

memoria que se va a usar, tiempos de espera, de activación de escritura, lectura, entre otras características.

Una vez hecho esto el microcontrolador escribirá y leerá automáticamente la memoria externa cada vez que se escriba o se lea una región del banco de memoria asignado.

Las configuraciones para este caso en particular son de: ancho de bus de 16bits, para memoria SRAM (banco de memoria desde 0x60000000 hasta 0x6FFFFFFF), modo A y hace uso de las líneas de control NOE, NWE, NE1, A16 del microcontrolador de manera que se conectan con /RD, /WR, /CS y D/C del integrado SSD1289 respectivamente.

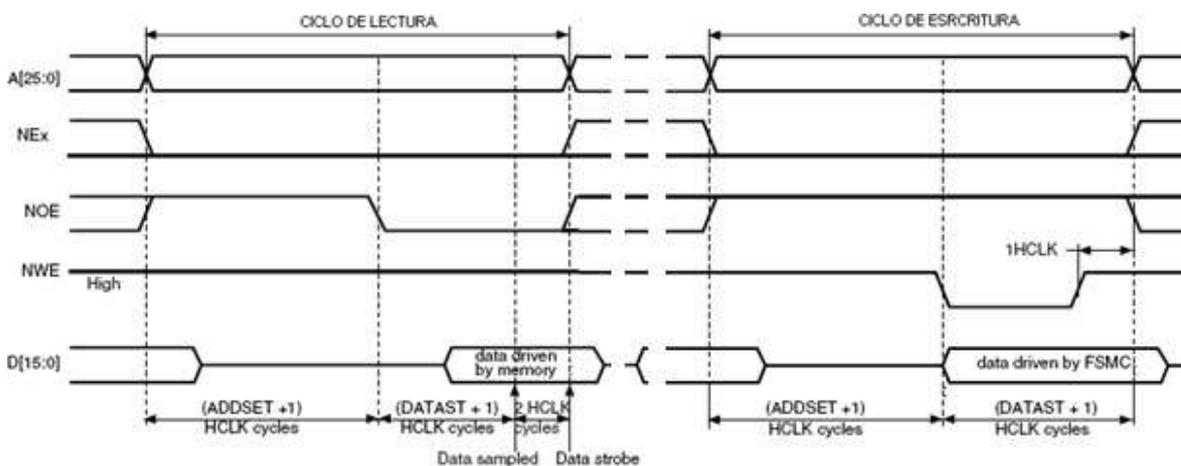


Figura 2.10 Operación del periférico FSMC según las configuraciones en este caso en particular (modo A, ancho de bus de 16bits, memoria SRAM) [14].

En las patillas del bus A[25:0] (address) se presenta automáticamente el valor de la dirección de memoria, un ejemplo de la implementación en código se muestra a continuación:

```
0x60000000=0x0023
```

Figura 2.11 Escritura del valor 0x0023 en la dirección de memoria 0x60000000.

En la Figura 2.11 se indica una única línea de código, cuando ésta es ejecutada por el microcontrolador, reconociendo que la dirección es igual o mayor a 0x60000000 (banco de memoria empleado por el bloque FSMC), de manera

automática el periférico FSMC realiza las operaciones de escritura respectivas (ver la Figura 2.10). En el pin A[16] se escribe cero lógico (0x60000000), esto indica una acción en el control de registros en el integrado SSD1289 (ver Figura 2.9). Cuando la línea /CS cambia a cero lógico en el bus de datos D[15:0] se escribe el valor 0x0023. Una vez terminado el ciclo de escritura tendremos como resultado que el registro 0x0023 del integrado SSD1289 está listo ya sea para una acción de lectura o de escritura de datos. Si quisiéramos escribir en ese registro el dato 0xFFFF debemos realizar una acción de escritura de datos y sería ejecutada mediante la línea de código ilustrada en la Figura 2.12 (en este caso el pin A[16] toma el valor de uno lógico):

0x600F0000=0xFFFF

Figura 2.12 Escritura del valor 0xFFFF en la dirección de memoria 0x600F0000.

2.1.2.2 Configuraciones del integrado SSD1289

El circuito integrado SSD1289 es capaz de representar colores en dos formatos RGB666 (262k colores) y RGB565 (65k colores), debido a que la cámara empleada transmite imágenes en un formato RGB565 éste es el elegido para la pantalla LCD. Al hacer uso de la interfaz 8080 de 16 bits podemos enviar o recibir toda la información de un pixel en tan sólo un ciclo de escritura o lectura respectivamente.

			Hardware pins																	
Interface	Color mode	Cycle	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
16 bits	65k		R4	R3	R2	R1	R0	G5	G4	G3		G2	G1	G0	B4	B3	B2	B1	B0	

Figura 2.13 Pines empleados por el integrado SSD1289 para la transmisión de un pixel en el formato RGB565 con interfaz de 16 bits [13].

En la Figura 2.13 se aprecia que sólo un ciclo es necesario para la comunicación de toda la información de color necesaria para el pixel. Además se indica la correspondiente conexión física de los pines del integrado SSD1289 con el bus de datos de 16 bits.

Otra configuración que se le asigna es la de autoincremento cuando se escribe o se realiza lectura en el registro 22. De esta manera al momento de escribir o leer

un dato por el bus de 16 bits el cursor se incrementa automáticamente para ser escrito o leído nuevamente.

2.1.2.3 Lámina táctil

La pantalla LCD lleva sobrepuesta una lámina táctil tipo resistiva de 4 líneas. La lámina táctil consta de un grupo de capas rectangulares (ilustradas en la Figura 2.14), la primera es transparente y flexible, ofrece protección al resto de capas y es la que tiene contacto físico con los dedos o un puntero usado para aplicar presión. Bajo esta capa se encuentra otra que tiene depositada un material conductor transparente (usualmente Oxido de Indio y Estaño, también conocido como ITO), le sigue una zona de aire y espaciadores, luego otra capa conductora y finalmente una capa sólida estable.

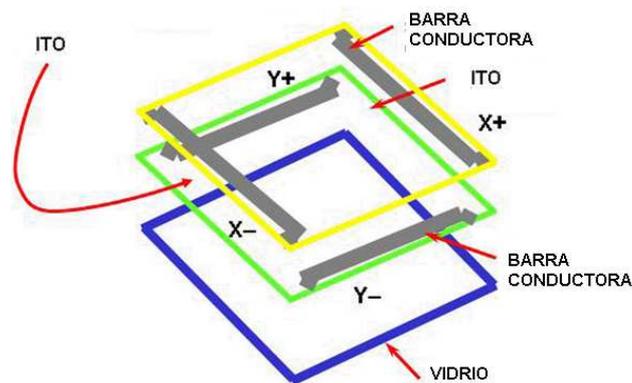


Figura 2.14 Capas que constituyen la lámina táctil (no se muestra la capa flexible de protección)

[15].

Las dos láminas conductoras están separadas y sólo hacen contacto entre ellas si se aplica presión sobre la capa flexible protectora mediante un puntero o un dedo, un voltaje se aplica en las barras conductoras ubicadas en una de las dos capas, por ejemplo en Y+ y Y-. El punto donde se hizo presión hace contacto con la lámina que no tiene alimentación de voltaje (ver la Figura 2.15), por lo tanto el voltaje en los terminales X+ y X- debe ser el resultante del divisor de voltaje formado por la resistencia entre Y+, el punto de presión y Y-. Durante este instante los terminales X+ y X- están conectados a un conversor analógico-digital de alta impedancia a fin de medir el voltaje en el punto de presión. Esta operación

se repite nuevamente pero alternando la capa en la que se aplica el voltaje, de esta manera se mide dos voltajes correspondientes a la posición X y Y.

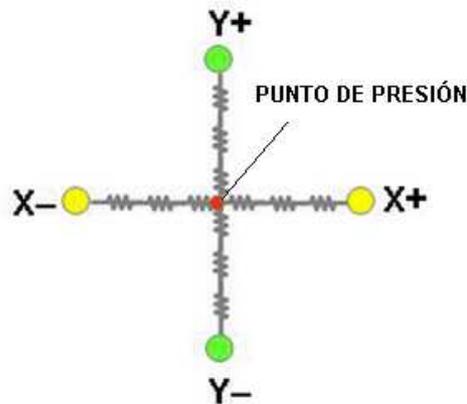


Figura 2.15 Representación eléctrica de las dos capas conductoras en contacto [15].

2.1.2.3.1 Manejo de la lámina táctil mediante el circuito integrado XPT2046

El circuito integrado XPT2046 tiene un bloque de control que realiza las acciones descritas para poder obtener los dos datos que ubican el punto de presión en la pantalla. Posee para esto un conversor ADC de 12 bits que muestrea el dato cada 125kHz. Como se aprecia en la Figura 2.16 el integrado posee terminales para conectar las 4 líneas de la pantalla directamente, además posee terminales adicionales que le sirven para comunicarse con el microcontrolador mediante interfaz SPI y una línea adicional cuyo nivel lógico es cero cuando se detecta un punto de presión sobre la lámina.

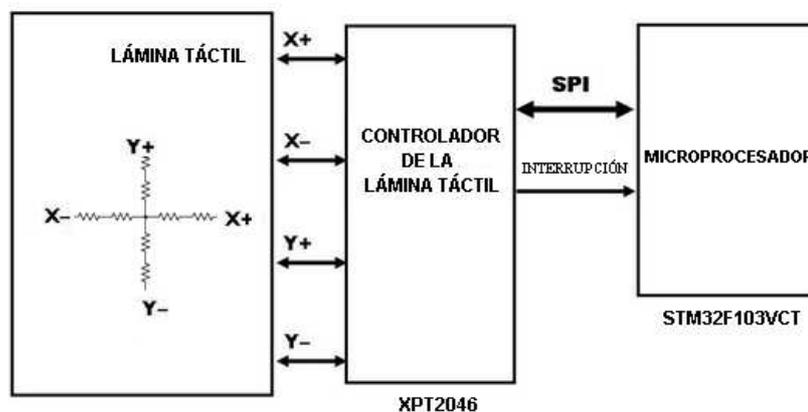


Figura 2.16 Diagrama de la comunicación entre la lámina táctil y el microcontrolador.

2.2 PROCESAMIENTO DE LA IMAGEN

La segmentación de la imagen se realiza durante la etapa de adquisición de la misma, el objetivo de esto es analizar uno de los pixeles de manera individual a medida que se leen en el microcontrolador. La segmentación tiene el fin de separar las regiones de interés de la imagen, considerándose como regiones de interés los pixeles que estén dentro de un umbral de color que se quiere detectar, este color se puede seleccionar mediante la pantalla táctil. Posteriormente un algoritmo analiza la región de interés y determina la existencia y ubicación de un objeto que contenga varios pixeles contiguos.

2.2.1 SEGMENTACIÓN DE UNA IMAGEN DE COLOR

La segmentación para este caso tiene la misión de separar una región de interés de la imagen basándose en el color que se elija. A medida que se obtiene cada pixel proveniente de la cámara, se lo compara con los valores de umbral, y se guarda el resultado en la imagen binaria. En la misma los pixeles que estén dentro del umbral serán representados como “uno lógico”, mientras los que estén fuera de este umbral serán “cero lógico”. La imagen binaria ocupa un espacio en el microcontrolador correspondiente a un arreglo de 2400 números de 32bits donde cada bit es un pixel de la imagen binaria, llegando a almacenar los 76800 pixeles correspondientes a un fotograma. A continuación en la Figura 2.17 un ejemplo de la imagen binaria siendo el color verde el color buscado.



Figura 2.17 Imagen original a la izquierda, Imagen binaria a la derecha.

2.2.1.1 Umbral de un color en el espacio RGB

En el formato RGB565 se posee 3 datos de color por pixel, el correspondiente al componente rojo toma valores de entre 0 y 32, el verde entre 0 y 64, finalmente el azul puede variar entre 0 y 32. La separación de los 3 componentes del color se realiza mediante la ayuda de 3 variables de 8 bits, de esta manera se toma los bits correspondientes del formato RGB565 para procesar de manera aislada cada componente.

Debido a esto el umbral del color deseado se escoge tomando una serie de muestras de los pixeles adyacentes al pixel escogido, a partir de éstos se determinan los valores máximos y mínimos de cada uno de los 3 componentes. El color sólo es admisible si está dentro de estos 3 rangos; como ejemplo en la Figura 2.18 hay dos variantes de color anaranjado, y un umbral cuyo valor es 32, cada uno de los tres componentes están dentro de los valores admisibles.

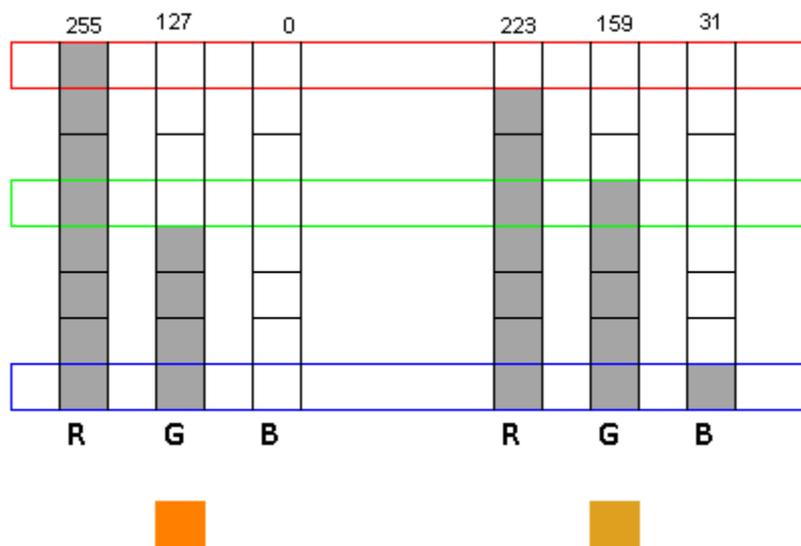


Figura 2.18 Dos variantes del color anaranjado, los valores de sus componentes individuales se muestran en la parte superior y se observa los umbrales permisibles para los componentes rojo, verde, y azul.

Los colores que vemos provenientes de un objeto son el resultado de la interacción entre iluminación y la reflectancia[16]. Un objeto posee distintas características que pueden afectar la manera en que refleja luz, ya sea por sus distintas geometrías, texturas, sombras, etc. A su vez una fuente de iluminación puede ser natural, artificial, frontal, lateral, a contraluz, etc.

La selección de un umbral de color en el espacio RGB presenta un problema debido a que en éste no están separados los datos de matiz y datos de iluminación. En el caso de existir un cambio de iluminación los valores individuales de rojo, azul y verde varían proporcionalmente a este cambio, lo cual se ilustra en Figura 2.20.

Una solución a este problema consiste en realizar una transformación del espacio RGB a otro espacio que tenga la información de matiz separada de la información de iluminación. Los espacios HSV, LAB, entre otros son posibles candidatos, pero la transformación debe ser aplicada a cada uno de los 78600 pixeles, implicando un tiempo considerable por cada operación matemática que requiera la transformación. Por esta razón se emplea el espacio RGB normalizado.

2.2.1.1.1 Espacio RGB normalizado

También conocido como el espacio *rg de cromaticidad*, el mismo que es bidimensional y que no contiene información de intensidad lumínica. En lugar de representar la intensidad de cada componente de color (RGB), se representa la proporción de cada uno (rgb) respecto a la iluminación total (I).

$$R + G + B = I$$

$$r = \frac{R}{I}, \quad g = \frac{G}{I}, \quad b = \frac{B}{I}$$

Es de importancia señalar que la sumatoria $r+g+b=1$, por lo tanto si se conoce r y g se tiene suficiente información para calcular b . Debido a esto se emplea tan sólo las variables r y g para obtener la información de cromaticidad, no se procesa el valor de b .

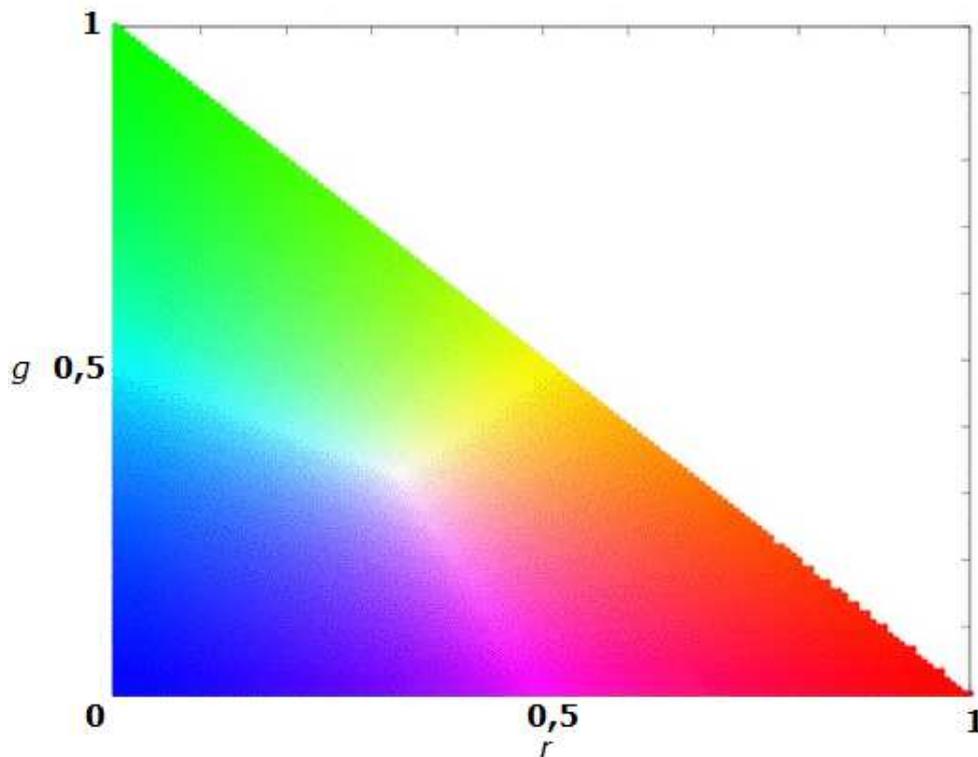


Figura 2.19 Espacio rg de cromaticidad [17].

En la Figura 2.19 se muestra el espacio de cromaticidad normalizado bidimensional en función de los valores r y g ; se aprecia que sólo se necesita estos 2 datos, para representar cualquier matiz.

Al aplicar la transformación a este espacio se emplea un número mínimo de operaciones matemáticas para obtener los valores r y g aislando así la información de cromaticidad.

En la siguiente figura se aprecia una esfera anaranjada donde debido al sombreado en el mismo cuerpo se tiene pixeles con componentes RGB que varían dramáticamente, aun cuando el matiz no ha cambiado.

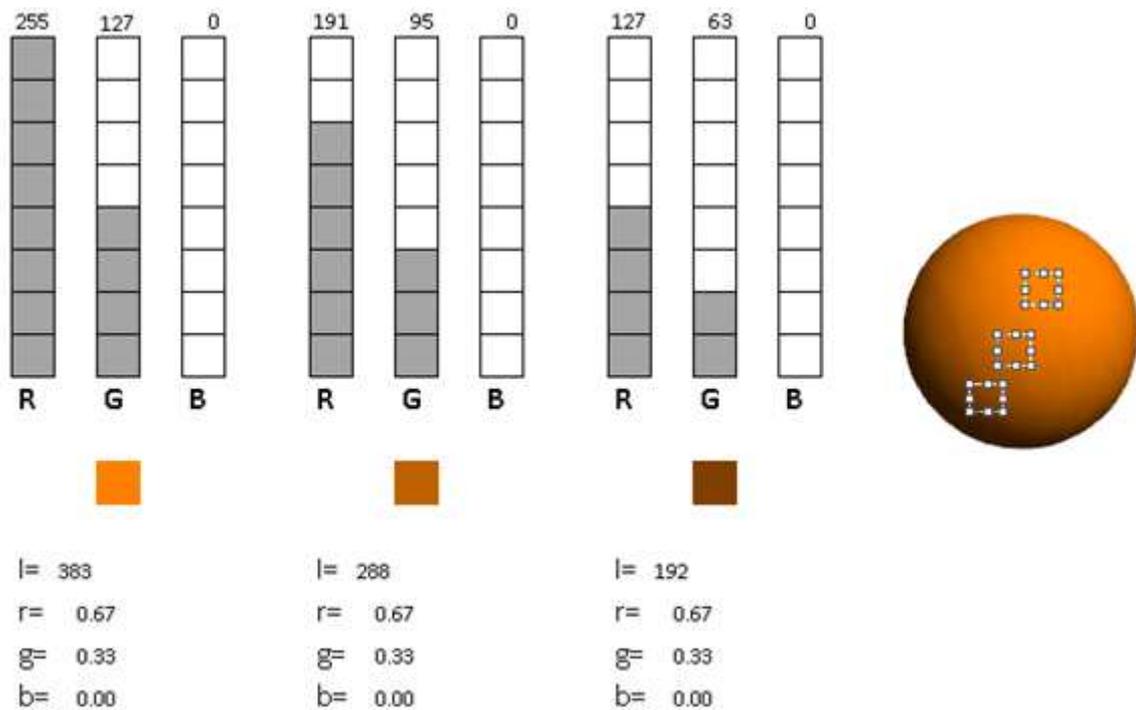


Figura 2.20 Tres pixeles de color distintos se escogen de una esfera anaranjada para analizar sus propiedades.

Claramente el componente rojo ha variado en casi la mitad de su valor, por lo que seleccionar un umbral en el espacio RGB donde se encuentre los posibles valores de rojo (R) es impráctico debido a la gran magnitud de la variación. Sin embargo, mientras el valor de iluminación cambia, los componentes guardan la misma proporción de r , g , y b . De esta manera se aplica la transformación a cada pixel y se lo compara con un umbral de color en el espacio rg de cromaticidad. Se logra así cierta tolerancia a las variaciones de iluminación (ver Figura 4.1).

2.2.2 DESCRIPCIÓN DE LA REGIÓN DE INTERÉS

Para analizar los datos provenientes de la etapa de segmentación se emplea un algoritmo sobre la imagen binaria. Para esto se asume que la imagen consiste de unos y ceros, siendo los pixeles del color de interés marcadas por uno lógico.

El algoritmo desarrollado tiene como objetivo delimitar la región de interés marcando el contorno de la agrupación de pixeles contiguos presentes en la imagen binaria (ver Figura 2.21), una vez realizado esto se entrega como

resultado los límites superior, inferior, lateral derecho, lateral izquierdo, y la ubicación tanto horizontal como vertical del centro del objeto.

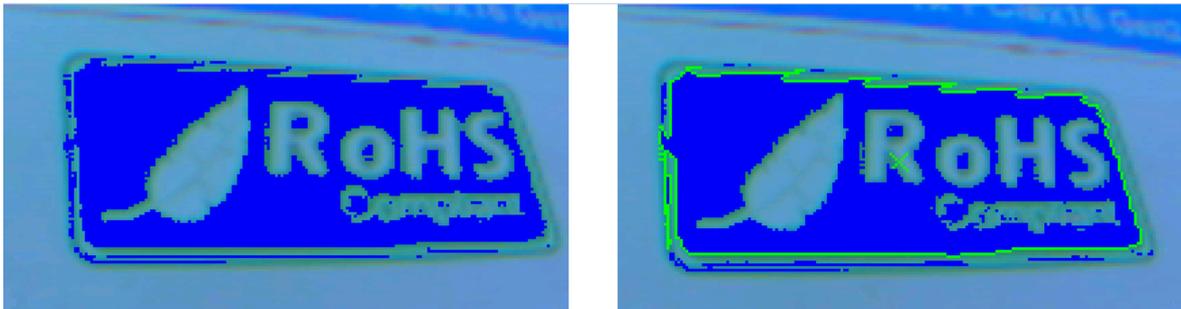


Figura 2.21 Reconocimiento del contorno de una imagen; a la izquierda la imagen original con los pixeles de la etapa de segmentación superpuestos, a la derecha el contorno reconocido en la imagen.

Para lograr este fin se inspecciona cada una de las filas de la imagen binaria desde la izquierda hacia la derecha, empezando de arriba hacia abajo. Si la agrupación de pixeles contiguas supera un ancho preestablecido entonces se empieza a recorrer el contorno a partir del primer pixel de esa línea.

La Figura 2.22 muestra como el algoritmo recorre el contorno de un sector de pixeles detectados, el proceso empieza una vez que se detecta la línea inicial (en este caso la línea inicial es de 1 pixel de ancho, y se muestra el inicio en color rojo), es desde aquí donde se inicia el recorrido del contorno, como regla el algoritmo empieza a buscar el siguiente pixel válido en sentido anti horario en una matriz de 3x3, iniciando la inspección desde un paso posterior al último pixel detectado. Para continuar la detección del contorno, el centro de la siguiente matriz de inspección se sitúa en el pixel detectado anteriormente. Cada vez que se detecta un pixel del contorno se compara si su valor de posición es menor al de la posición mínima, si este es el caso se actualiza el valor de posición mínima, lo mismo se realiza para la posición máxima. De esta manera al terminar de recorrer todo el contorno, se obtiene los máximos y mínimos de las posiciones en vertical y en horizontal. La inspección de contorno se detiene una vez que el pixel detectado vuelve a ser el correspondiente al de la posición inicial.

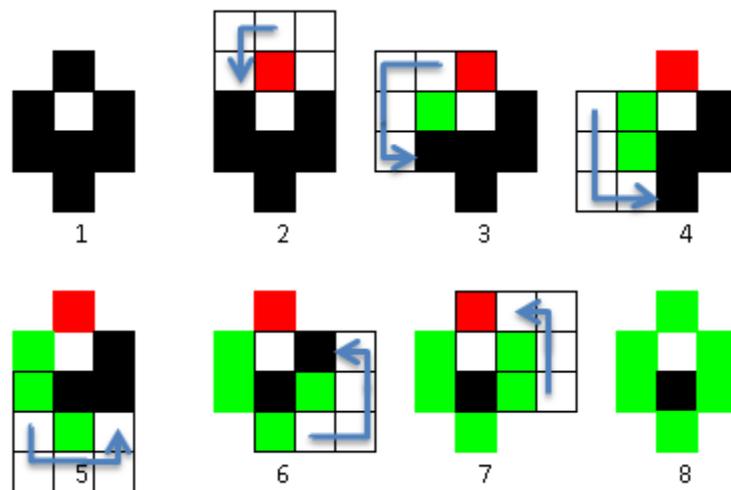


Figura 2.22 Ejemplo de reconocimiento del contorno de un grupo de píxeles.

Al emplear este algoritmo se tiene la seguridad que sin importar lo intrincada que sea la agrupación de píxeles, siempre se recorrerá todo el contorno y se retorna a la posición inicial; pero se debe tener en consideración que si la matriz 3x3 de evaluación está en los límites de la imagen, el centro estará sobre la imagen binaria, pero habrá píxeles evaluados fuera del cuadro de la imagen; se debe tener la precaución de que estos píxeles evaluados fuera del recuadro de la imagen sean igual a cero, es decir estén en blanco, de no ser así, el centro de la matriz de evaluación puede salir del recuadro de la imagen y evaluar sectores de memoria que no corresponden a la imagen produciendo resultados indeseables.

CAPÍTULO 3

PROGRAMACIÓN DEL APUNTAMIENTO Y DE LOS ALGORITMOS DE CONTROL

En este capítulo se describe la ubicación física de la cámara, los motores y la plataforma encargados de su movimiento, y los algoritmos que se emplean con la finalidad de ubicar a un objeto buscado en el centro del rango de visión de la cámara, lo cual se conoce como apuntamiento.

3.1 CONFIGURACIÓN FÍSICA DE LA PLATAFORMA

La plataforma donde se ubica la cámara con el fin de realizar el apuntamiento es controlada por dos servomotores, por lo que tiene dos grados de libertad (ver la Figura 3.1) para realizar los movimientos de rotación y guiñada (*Pan & Tilt*). El control de cada servomotor es realizado mediante modulación de ancho de pulso (PWM). Para esto se emplean los pines disponibles en la tarjeta de desarrollo HY-Smart STM32, existen 2 salidas PWM que se pueden emplear, son el Timer2 (canal 2) y Timer3 (canal 1). Estos controlan el servomotor HS-785HB (ubicado en la parte inferior de la plataforma) y el HS-645MG (ubicado en la parte superior).

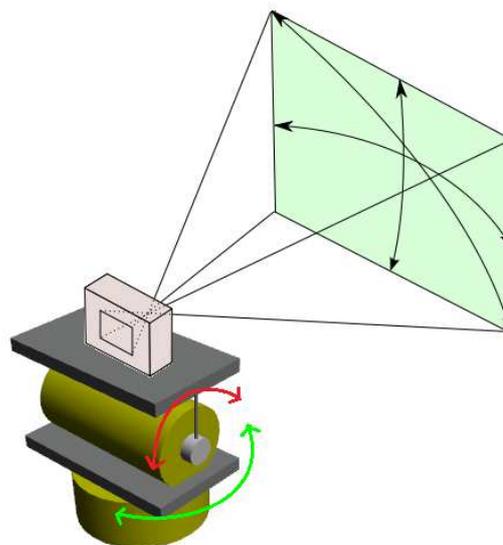


Figura 3.1 Representación de la plataforma que sostiene la cámara.

Si los servomotores se mueven varía la posición angular de los mismos, a su vez los objetos vistos en la pantalla se encuentran en una posición angular con respecto a la cámara. En la Figura 3.1 se representa a los servomotores en color amarillo, las flechas rojas indican como la cámara varía en posición vertical, y las verdes la variación horizontal.

3.1.1 CONFIGURACIÓN DE LOS SERVOMOTORES

Los 2 servomotores se controlan con pulsos cada 20ms, el ancho de pulso determina la posición que éstos tendrán, teniendo una resolución de 16 bits entonces se debe encontrar una configuración adecuada a fin de que el PWM tenga un periodo de 20ms. El microcontrolador permite emplear un pre-escalador y también fijar el límite del Timer con el que se actualiza el conteo del mismo. De esta manera el periodo de cada pulso está dado por la ecuación(3-1):

$$Periodo = \frac{2}{F_{sys}} \times (1 + TIM_PRESC) \times (1 + TIM_LIM) \quad (3-1)$$

Donde F_{sys} es la frecuencia del sistema (72MHz en este caso), TIM_PRESC es el valor del pre-escalador, y TIM_LIM es el límite, ambos pueden tomar valores entre 0 y 65535. Bajo estas configuraciones obtenemos el periodo deseado asignando un valor de 11 para el pre-escalador y de 59999 para el límite.

De esta manera ambas salidas PWM poseen igual configuración y la mayor resolución posible para un timer de 16 bits.

3.2 LAZO DE CONTROL

El sistema de visión obtiene realimentación a través de lo que observa en la cámara, el apuntamiento siempre pretenderá rotar la cámara de manera que el objeto de interés esté en el centro. Razón por la cual la señal de error estará dada por la posición detectada del objeto menos la posición central. A su vez la posición detectada del objeto no tiene correlación con la posición angular actual de los servomotores, por este motivo la medición de la posición detectada del objeto no es absoluta, es relativa a la ubicación angular de la cámara en ese instante de tiempo. Este hecho no afecta a la señal de error de manera alguna

debido a que la posición central de la cámara obviamente también es relativa a la orientación de la cámara.

3.2.1 IMPLEMENTACIÓN DIGITAL DE UN CONTROLADOR PI

El algoritmo de control empleado para ambos servos es un control tipo proporcional-integral (PI por sus siglas). Primeramente se debe determinar los efectos que tendrán las acciones de control sobre la variable de salida para obtener el modelo aproximado de la planta.

La acción de control es el tiempo de pulso en los servomotores, la variable que se mide es la posición detectada del objeto sobre la pantalla. Se debe establecer la razón entre el tiempo de pulso enviado a los servomotores y el cambio de la posición detectada en la pantalla, hacemos esto midiendo cuanto ha cambiado el valor del timer cuando el objeto en la pantalla se ha desplazado un número determinado de pixeles, este valor será la ganancia del sistema (es de importancia señalar que la ganancia para ambos servomotores no necesariamente es la misma).

Si bien el sistema se compone de servomotores, podemos hacer una modelación aproximada de la planta en base a su ganancia y a su respuesta dinámica.

Para esto se ha determinado representar el sistema mediante una función de transferencia de primer orden (3-2) cuya ganancia (K) ha sido determinada en base al método explicado previamente.

$$Gp(s) = \frac{K}{\tau \times s + 1} \quad (3-2)$$

La constante de tiempo τ se estima en base a la respuesta dinámica a la entrada paso unitaria. En la Figura 3.2 se aprecia la respuesta del sistema en función del tiempo, para un sistema de primer orden el tiempo de establecimiento es alcanzado en un tiempo de n veces τ .

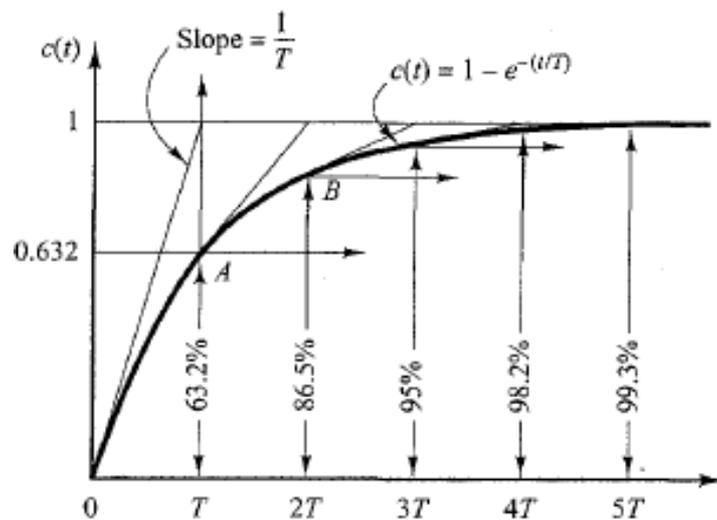


Figura 3.2 Respuesta de un sistema de primer orden ante la entrada paso unitaria [18].

Experimentalmente se determina que el sistema alcanzó el 98.2% de su valor final en aproximadamente 500ms lo cual viene a ser 4 veces el valor de la constante de tiempo τ .

3.2.1.1 Cálculo de las constantes K_p y K_i

Se decide usar un controlador PI ya que la acción proporcional, al requerir constantemente una magnitud de error distinta de cero, presenta constantemente un error de posición; esto se soluciona añadiendo la acción integral. A continuación se muestra la función de transferencia de un controlador PI.

$$G_c(s) = K_p + \frac{K_i}{s} = \frac{u(s)}{e(s)} \quad (3-3)$$

3.2.1.1.1 Análisis a lazo cerrado

Para facilitar el análisis se representa el sistema de un servomotor cuya acción de control es regida por un controlador en el sistema de lazo cerrado para tiempo continuo.

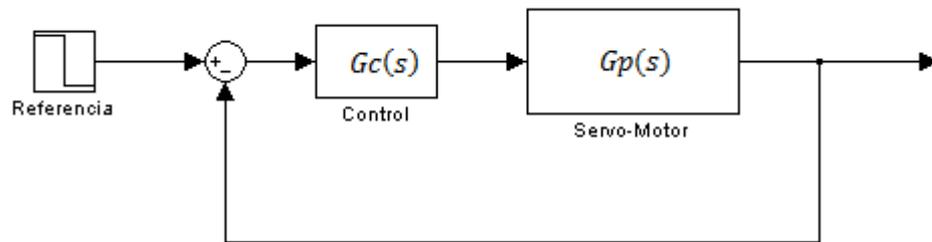


Figura 3.3 Diagrama de bloques simplificado del sistema en lazo cerrado.

El diagrama de bloques se puede simplificar en un sólo bloque obteniendo la siguiente función de transferencia que representa todo el sistema:

$$Tf(s) = \frac{Gp(s) * Gc(s)}{Gp(s) * Gc(s) + 1}$$

Los polos a lazo cerrado del sistema vendrán dados por las raíces del polinomio que se encuentra en el denominador. A continuación desarrollamos la expresión hasta encontrar dicho polinomio (3-4):

$$Tf(s) = \frac{(Kp + Ki/s) * (\frac{K}{\tau \times s + 1})}{(Kp + Ki/s) * (\frac{K}{\tau \times s + 1}) + 1}$$

$$Tf(s) = \frac{s \times (Kp \times K) + Ki \times K}{s^2 \times \tau + s \times (1 + Kp \times K) + Ki \times K}$$

$$p(s) = s^2 + s \times (1 + Kp \times K)/\tau + Ki \times K/\tau \quad (3-4)$$

Se aprecia una función de transferencia de segundo orden cuyo denominador se puede representar de la siguiente manera [18]:

$$p(s) = s^2 + s \times (2 \times \xi \times \omega_n) + \omega_n^2 \quad (3-5)$$

Donde los valores del coeficiente de amortiguamiento (ξ) y la frecuencia natural (ω_n) determinan en el sistema el porcentaje de sobre pico (PS) y también el tiempo de establecimiento (t_s) presente en la respuesta a una entrada escalón:

$$PS = 100\% \times e^{\frac{-\xi \times \pi}{\sqrt{1-\xi^2}}} \quad (3-6)$$

$$ts = \frac{4}{\xi \times \omega_n} \quad (3-7)$$

Realizando los reemplazos necesarios en las ecuaciones: (3-4), (3-5), (3-6), y (3-7) se despejan los valores Kp y Ki en función del tiempo de establecimiento y del sobre pico.

$$Kp = \frac{1}{K} \times \left(\frac{8 \times \tau}{ts} - 1 \right) \quad (3-8)$$

$$Ki = \frac{1}{K} \times \left(\frac{16 \times \tau}{ts^2 \times \frac{(\ln \frac{PS}{100\%})^2}{\pi^2 + (\ln \frac{PS}{100\%})^2}} \right) \quad (3-9)$$

3.2.1.2 Implementación del controlador en tiempo discreto

El controlador PI descrito en la ecuación (3-3) está expresado en tiempo continuo en el dominio s . Para implementar un controlador PI en el microcontrolador debemos discretizar esta función de transferencia y expresarla en el dominio del tiempo discreto. Esto se logra mediante la transformación bilinear (también llamada integración trapezoidal) la cual convierte la función de transferencia del dominio de s al tiempo discreto en el dominio z .

$$G(z) = G(s) \Big|_{s=\frac{2(z-1)}{T(z+1)}}$$

Donde T es el tiempo de muestreo. Aplicando la transformación al controlador $Gc(s)$, obtenemos la siguiente función de transferencia:

$$Gc(z) = Kp + \frac{Ki \times T(z+1)}{2(z-1)} = \frac{U(z)}{E(z)}$$

El controlador expresado en el dominio de z puede expresarse en el dominio del tiempo discreto, lo cual se realiza a continuación:

$$E(z) \times \left[Kp \times (1 - z^{-1}) + Ki \times \frac{T}{2} (1 + z^{-1}) \right] = U(z) \times (1 - z^{-1})$$

$$\left[Kp \times (E_{[k]} - E_{[k-1]}) + Ki \times \frac{T}{2} (E_{[k]} + E_{[k-1]}) \right] = U_{[k]} - U_{[k-1]}$$

$$U_{[k]} = U_{[k-1]} + E_{[k-1]} \times \left(Ki \times \frac{T}{2} - Kp \right) + E_{[k]} \times \left(Ki \times \frac{T}{2} + Kp \right) \quad (3-10)$$

Donde $U_{[k]}$ es el valor de la acción de control (valor del pulso PWM del servomotor) en ese instante de tiempo, el controlador actualiza su valor cada vez que se reconoce un fotograma, esto ocurre cada 115ms y viene a ser equivalente al tiempo de muestreo T ; Kp y Ki son determinados con las ecuaciones (3-8) y (3-9) respectivamente.

3.2.2 IMPLEMENTACIÓN DIGITAL DE FILTROS

La posición detectada por el algoritmo de reconocimiento está sujeta al ruido que se produce en cada fotograma, y además un movimiento rápido de los servos provoca una imagen borrosa lo cual empeora la calidad de la imagen y dificulta el reconocimiento del objeto. Para evitar estos problemas significativos se ha optado por implementar un filtro digital cuya finalidad es suavizar el movimiento de los servomotores.

El filtro que se ha empleado es de *Suavizamiento Exponencial*, el mismo presenta una respuesta infinita al impulso por lo que se clasifica como tipo IIR (*Infinite Impulse Response*).

$$Y_{[k]} = \alpha \times X_{[k]} + (1 - \alpha) \times Y_{[k-1]} \quad (3-11)$$

En la ecuación (3-11) se muestra la implementación digital del filtro, en la misma $Y_{[k]}$ es la señal filtrada, $X_{[k]}$ es la señal sin filtrar, y α determina el peso que tiene la señal sin filtrar. Es de importancia que el valor del peso sea comprendido entre $0 < \alpha < 1$, si α tiene valores cercanos a 1 la señal tendrá poca suavización; si α es cercano a 0 los aportes $X_{[k]}$ serán mínimos.

Para comprender como actúa el filtro a continuación se muestra un caso específico en el que la señal filtrada es equivalente a la representación del voltaje en el capacitor en un circuito RC en serie, más conocido como filtro pasa-bajo, a continuación se muestra una ecuación con su dinámica:

$$v_{in}(t) - v_c(t) = RC \times \frac{dv_c(t)}{dt}$$

Aproximando la ecuación a intervalos de tiempo discreto tenemos:

$$Vin_{[k]} - Vc_{[k]} = RC \times \frac{Vc_{[k]} - Vc_{[k-1]}}{T}$$

$$Vin_{[k]} + Vc_{[k-1]} \times \frac{RC}{T} = Vc_{[k]} \left(1 + \frac{RC}{T}\right)$$

$$Vin_{[k]} - Vc_{[k-1]} + Vc_{[k-1]} \left(1 + \frac{RC}{T}\right) = Vc_{[k]} \left(1 + \frac{RC}{T}\right)$$

$$\alpha = \left(1 + \frac{RC}{T}\right)^{-1}$$

$$Vin_{[k]} - Vc_{[k-1]} + Vc_{[k-1]} \times \alpha^{-1} = Vc_{[k]} \times \alpha^{-1}$$

$$Vc_{[k]} = Vin_{[k]} \times \alpha + Vc_{[k-1]} \times (1 - \alpha)$$

A su vez las constantes R y C determinan la respuesta dinámica del sistema de primer orden viniendo a ser la constante de tiempo igual a la multiplicación de estos términos. De esta manera α queda en función de la constante de tiempo que le queramos dar al filtro y del tiempo de muestreo

$$\alpha = \frac{T}{\tau + T}$$

El filtro se actualiza durante una interrupción producida por el mismo timer que maneja el servomotor, esto sucede cada 20ms.

Para que el controlador no se vea afectado por un tiempo de respuesta más lento de los servomotores, el valor del tiempo de establecimiento del filtro será el mismo que el tiempo de actualización de el controlador, esto es cada 115ms. Es decir el filtro alcanza el valor de 98.2% por cada vez que se actualiza la acción de control, la respuesta se aprecia en la Figura 3.4.

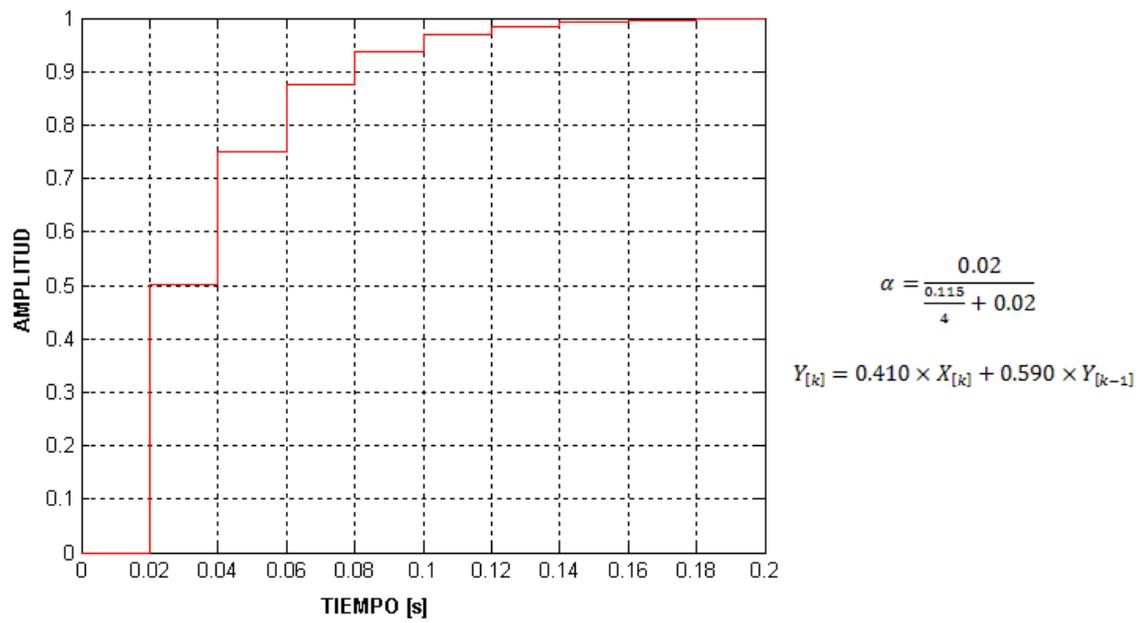


Figura 3.4 Simulación de la respuesta del filtro a una entrada paso unitaria, a la derecha se indica el valor de las constantes empleadas.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En el presente capítulo se realizan una serie de pruebas prácticas para demostrar el funcionamiento del sistema de visión desarrollado y el comportamiento que presenta ante distintas variaciones. Para esto es de importancia la documentación digital de las imágenes presentes en la pantalla y los datos de posición del objeto detectado, por lo cual se describe un método de obtención de estos datos que hace uso del pórtico serial.

4.1 TRANSMISIÓN SERIAL DE UNA IMAGEN

Con el objetivo de documentar las imágenes mostradas en la pantalla LCD, se implementó un algoritmo que recibe los datos de imagen por el pórtico serial, posteriormente los dibuja y guarda un archivo .bmp.

El tiempo que demora la transmisión serial de un fotograma depende de los parámetros empleados para la transmisión serial. En este caso en particular se emplea un pórtico UART configurado de la siguiente manera: 8 bits de datos, 1 bit de parada, sin paridad, velocidad de transmisión de 921600 bits por segundo. La transmisión de cada byte de datos emplea 1 bit de inicio y uno de parada por lo tanto se emplea un total de 10 bits por cada byte transmitido, demorándose un tiempo dado por la ecuación (4-1).

$$T_{byte} = \frac{10}{\text{Baud Rate}} \quad (4-1)$$

El número total de bytes de información contenida por cada fotograma determinará el tiempo que demora la transmisión del mismo. Esto a su vez dependerá del número de píxeles así como también del número de bits que emplea cada pixel.

Mediante la ecuación (4-2) se puede calcular el tiempo total para la transmisión de un fotograma sin ningún tipo de compresión y asumiendo que se tiene un número de píxeles igual al producto del número de líneas por columnas.

$$T_{frame} = nRows \times nLines \times T_{byte} \times \left(\frac{bpp}{8}\right) \quad (4-2)$$

Donde $nRows$ es el número de columnas en el fotograma, $nLines$ es el número de filas en el fotograma, T_{byte} está dado por la ecuación (4-1), y bpp es el número de bits empleado por pixel (para una tasa de transferencia máxima cada pixel debe estar contiguo de manera que un byte no tenga bits “vacíos” sin información del pixel).

El algoritmo es ejecutado mediante Matlab, adquiriendo cada uno de los 76800 pixeles, ubicándolos en una matriz, dibujándolos y guardándolos, para lo cual se demora 1666mS aproximadamente. La transmisión es lo suficientemente lenta para hacer impráctico al apuntamiento del sistema de visión, por lo cual sólo se transmite datos bajo demanda.

4.2 PRUEBAS DE DETECCIÓN Y LOCALIZACIÓN

4.2.1 Segmentación de la imagen ante distintos niveles de iluminación

La importancia de emplear la transformación del espacio RGB normalizado se aprecia en la Figura 4.1; a pesar de requerir una transformación matemática por cada pixel, se aprecia una mejor segmentación de la imagen. Para esta prueba se empleo 2 luminarias LED laterales cuya intensidad es controlada por modulación de ancho de pulso (PWM), para los dos tipos de segmentación el color a detectar se tomó durante el mayor nivel de iluminación, y posteriormente se redujo la iluminación hasta obtener 4 muestras por cada tipo de segmentación.

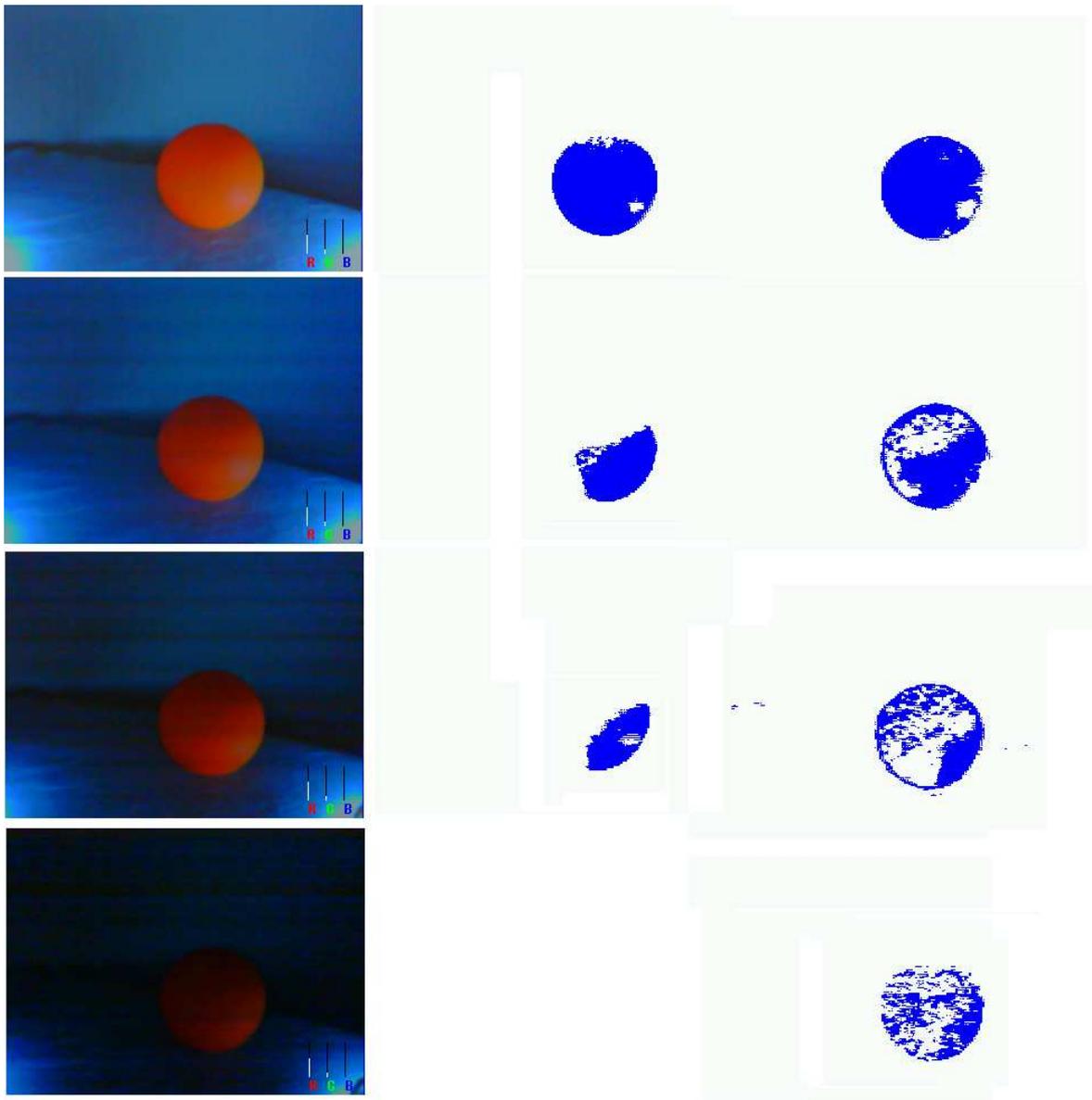


Figura 4.1 A la izquierda una esfera es iluminada con 4 niveles de iluminación; en el centro la segmentación empleando el espacio de color RGB565; a la derecha la segmentación resultante del espacio de color RGB normalizado.

El sistema de visión de este proyecto puede realizar la segmentación tanto en el espacio RGB565 como en el espacio RGB normalizado, siendo la última opción la predeterminada por su mayor confiabilidad; el uso del espacio RGB565 sería más apropiado en ambientes sumamente controlados donde los objetos que se esperen detectar no presenten sombreado ni variaciones representativas de iluminación dentro de su cuerpo.

4.2.2 Detección de objetos de distintos colores

En la Figura 4.2 se observa distintos objetos cuyo color no difiere mucho y por lo tanto sirven para demostrar la capacidad del sistema de visión de discernir colores; se tiene una tela roja, una esfera pequeña amarilla, una esfera grande anaranjada, y un sobre de color amarillo pálido. En cada uno de los 4 experimentos se seleccionó el color respectivo, y la localización de los objetos se realizó de manera apropiada.



Figura 4.2 Diferentes objetos de color similar son reconocidos de manera separada.

Un inconveniente se puede apreciar en la detección tanto de la tela roja, y más notablemente en la esfera amarilla pequeña, su sombreado no es reconocido por la etapa de segmentación, como consecuencia el contorno marcado del objeto no es ideal en todos los casos.

4.2.3 Pruebas de localización

Para esta prueba se emplea un objeto anaranjado unido a una estructura acoplada al eje de un motor DC, por lo cual se mueve de manera circular, similar a un reloj (ver Figura 4.3). La velocidad del motor es controlada por modulación de ancho de pulso (PWM), esto permite realizar pruebas de localización a distintas velocidades angulares y observar el comportamiento del sistema.



Figura 4.3 El triángulo anaranjado se mueve de manera circular en sentido horario (semejante a un reloj). Se aprecia la detección del contorno del objeto y la localización por parte del sistema.

A continuación se muestran las gráficas (Figura 4.4, Figura 4.5, y Figura 4.6) para distintas velocidades de revolución indicando posición-tiempo en el eje X, posición-tiempo en el eje Y, y de posición respecto a los ejes XY.

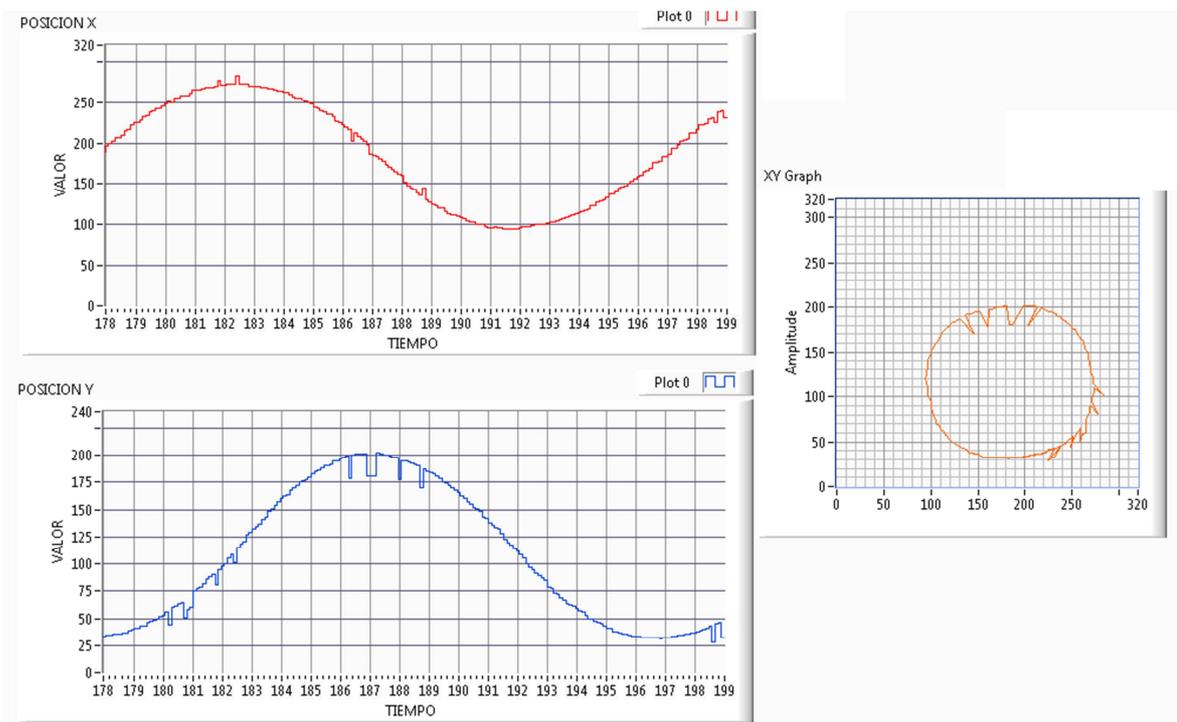


Figura 4.4 Localización del objeto realizando una revolución cada 21170ms.

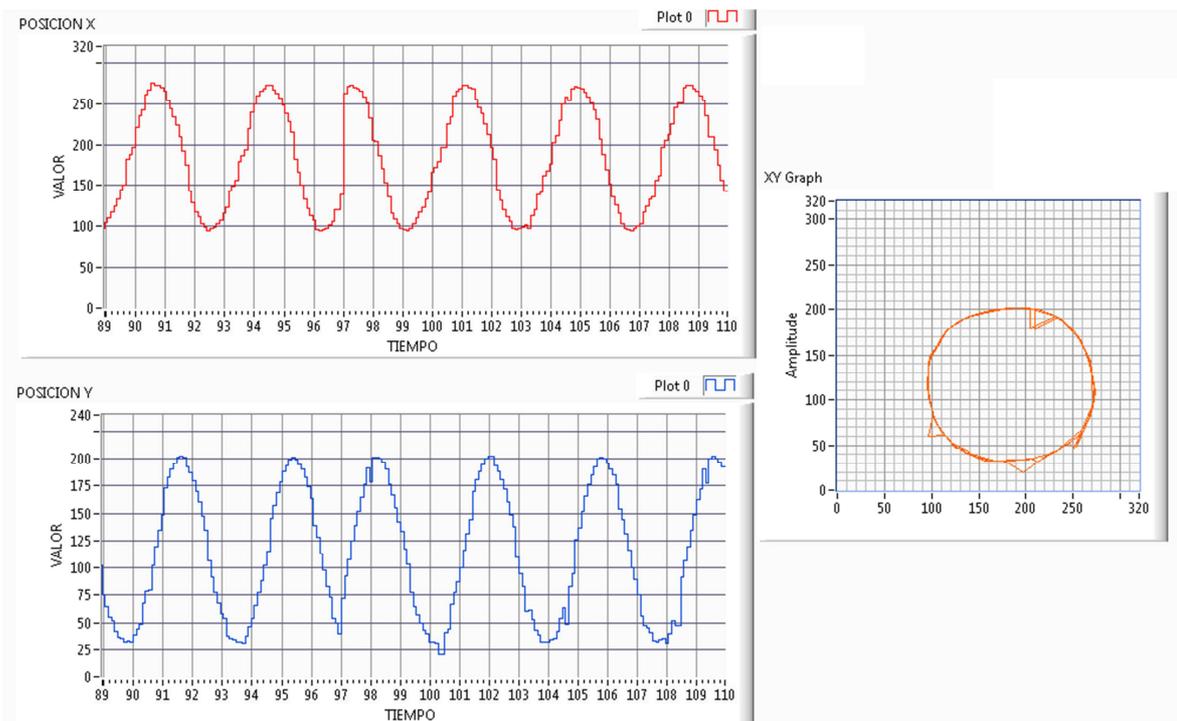


Figura 4.5 Localización del objeto realizando una revolución cada 3820ms.

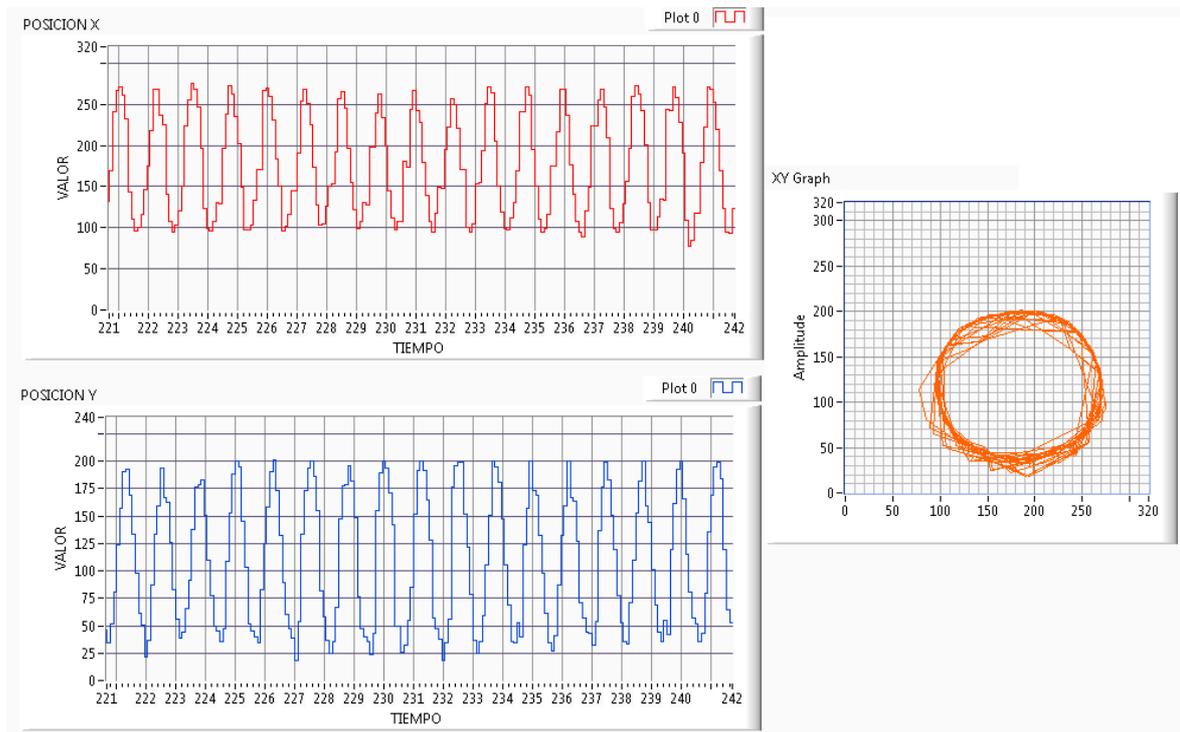


Figura 4.6 Localización del objeto realizando una revolución cada 1108ms.

Se puede apreciar en la Figura 4.4 que se realiza una localización bastante aceptable para el objeto moviéndose muy lentamente, en la Figura 4.5 el objeto se mueve más rápido, la localización aun es aceptable; finalmente en la Figura 4.6 el movimiento del objeto es lo suficientemente rápido para que la localización presente variaciones considerables.

Un hecho desfavorable en estas pruebas se aprecia en tiempos distintos tanto en el eje X como en el eje Y, se presentan ciertos "saltos" en la magnitud. Esto se debe al ruido, de manera aleatoria pixeles dentro del mismo cuerpo tendrán un color "no válido" y por lo tanto la imagen binaria no es ideal, tendrá ciertos pixeles "huecos" como se ilustra en la Figura 4.7 b). Este efecto será más notable principalmente en los bordes del cuerpo del objeto, si lo analizamos como una imagen bidimensional es en los bordes donde el color del entorno hace una transición hacia el color del objeto, la imagen está sujeta al ruido y los colores del borde son una mezcla entre el color de fondo y el del objeto.

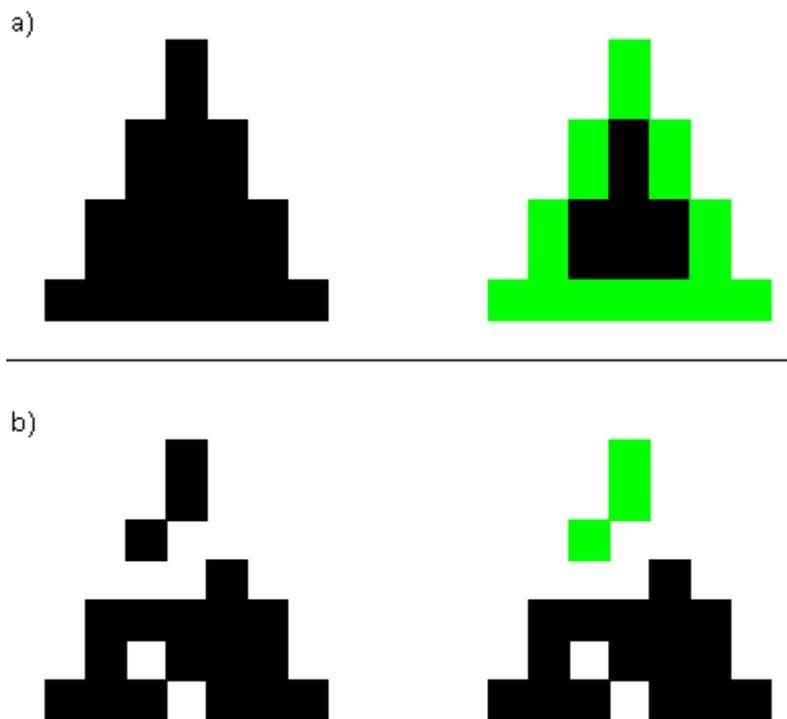


Figura 4.7 a) Imagen binaria segmentada de manera ideal, b) Imagen binaria segmentada con pixeles huecos.

En la Figura 4.7 b) se presenta un caso donde los pixeles huecos por casualidad resultan estar contiguos, llegando a "recortar" el cuerpo del objeto; el sistema simplemente detecta el primer objeto (desde la parte superior a la izquierda); y por lo tanto el centro de este objeto será distinto que el del cuerpo entero.

Una posible solución a este inconveniente consiste en realizar una "erosión" en la imagen binaria, la cual es una operación que remueve los pixeles de la parte externa de una agrupación, pero emplear este método implica procesar nuevamente todos los pixeles de la imagen lo cual puede consumir mucho tiempo de procesamiento; se opta por no implementar esta solución dado que los efectos causados por este inconveniente son tolerables y preferibles ante un incremento en el tiempo de procesamiento.

4.2.4 Pruebas del controlador y del apuntamiento

Para probar el comportamiento del controlador se trató de recrear una entrada paso para ambos ejes de movimiento, para ello se eligió un objeto de color a detectar ubicado en una esquina del rango de visión de la cámara y

posteriormente se activó el apuntamiento con lo cual los motores ubicaron el centro de la cámara en la ubicación del objeto.

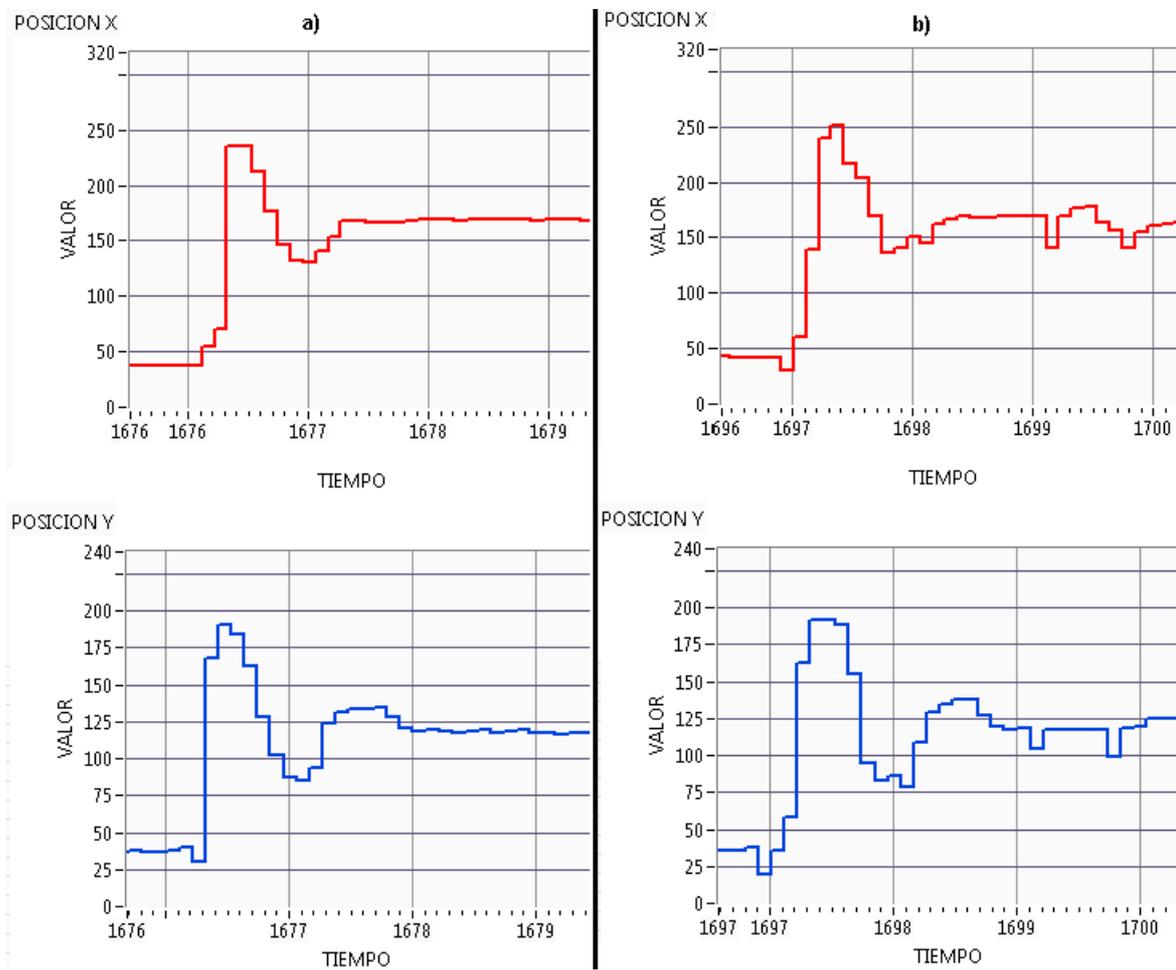


Figura 4.8 a) Respuesta del sistema para apuntar un objeto inicialmente ubicado en la esquina superior izquierda; b) Repetición de la prueba.

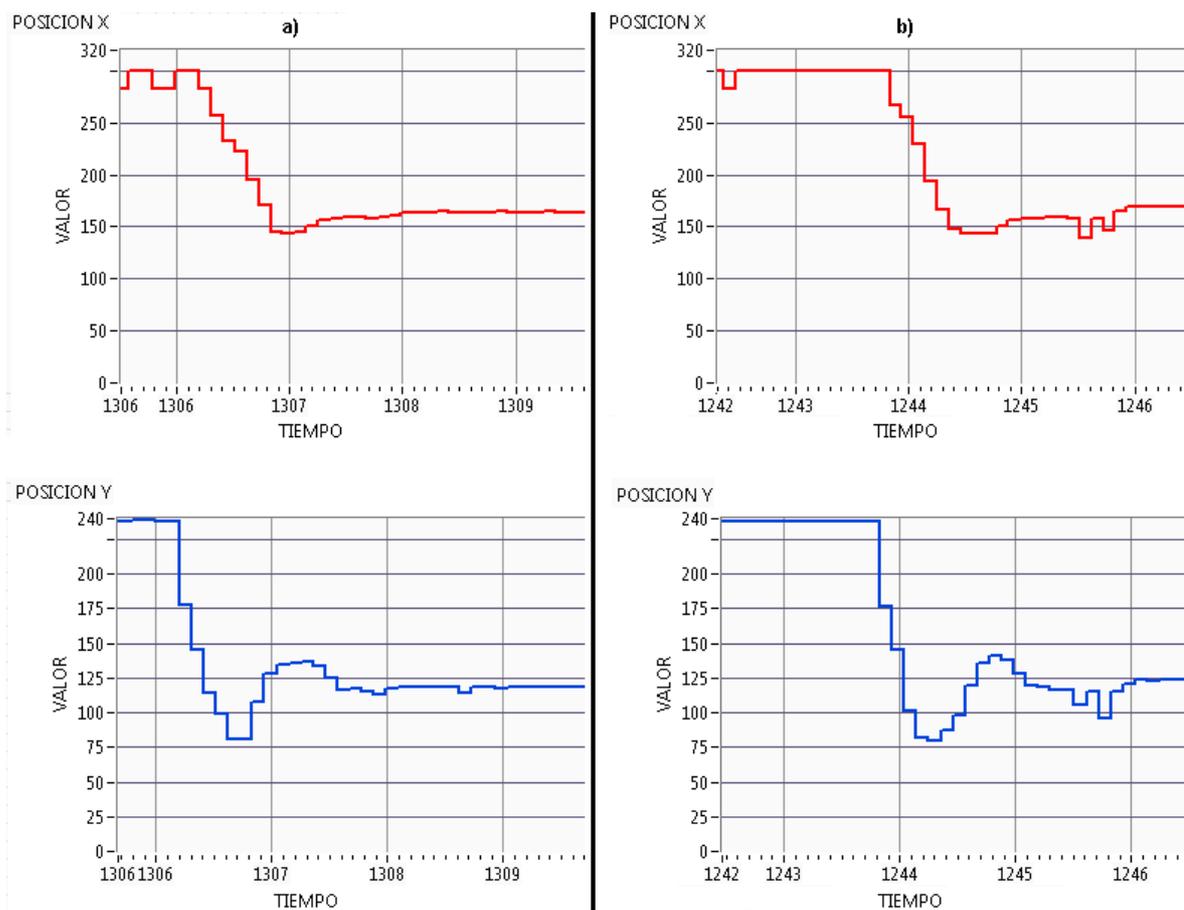


Figura 4.9 a) Respuesta del sistema para apuntar un objeto inicialmente ubicado en la esquina inferior derecha; b) Repetición de la prueba.

Tanto en la Figura 4.9 como en la Figura 4.8 se aprecia la respuesta del sistema con el controlador empleado. En poco tiempo el controlador apunta la cámara hacia el objeto. Cuando el valor absoluto del error llega ser menor a 12 se considera que el error fue corregido (se le da el valor de cero), esto se hace con el fin de que los servomotores no produzcan pequeñas oscilaciones, y se aprecia en las gráficas donde el valor de establecimiento no es exactamente el del centro de la imagen sino que la ubicación relativa del objeto es alrededor de 160 en el eje X, y de 120 en el eje Y.

El funcionamiento del controlador es satisfactorio, si bien es notable el sobre-pico en la respuesta, se aprecia un tiempo de establecimiento bastante rápido de aproximadamente 1.2 segundos. Aparentemente el sobre-pico es ligeramente más evidente en el eje Y; también se aprecia "saltos" en magnitud cuyo origen se explicó anteriormente en las pruebas de localización.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

A continuación se presentan las conclusiones y recomendaciones basadas en los resultados y las experiencias conseguidas en el desarrollo de este proyecto.

5.1 CONCLUSIONES Y RECOMENDACIONES

- El sistema de visión desarrollado en este proyecto tiene la capacidad de detectar, localizar, y realizar el apuntamiento de un objeto de color de manera satisfactoria mediante un microcontrolador, sin hacer uso de un computador personal.
- El proyecto desarrollado cubre las etapas básicas de un sistema de visión artificial (pre-procesamiento, segmentación, descripción, reconocimiento e interpretación), de ser necesario se puede evaluar el sistema en este proyecto y modificarlo para cumplir con requerimientos más específicos.
- Varios tipos de reconocimiento de imagen, el de objetos por color incluido, obtienen mejores resultados en un ambiente controlado en cuanto al tipo de fuente de iluminación y su ubicación, ya que los colores que vemos provenientes de un objeto son el resultado de la interacción entre iluminación y la reflectancia, a su vez dicho objeto posee distintas características que pueden afectar la manera en que refleja luz, ya sea por sus distintas geometrías, texturas, sombras, etc. Una recomendación para lograr un sistema más fiable es que, de ser posible, el objeto emita luz de determinado color, reduciendo así los problemas antes mencionados.
- Un sistema de visión debe empezar por diseñarse alrededor de una aplicación específica, en una etapa preliminar se debe establecer la clase de objetos a detectarse, la cantidad de objetos presentes en un instante de tiempo, la localización de la fuente de iluminación, en general se debe diseñar el sistema con un claro conocimiento de las imágenes que se esperan obtener; un sistema general y cuyo uso es pretendido en ambientes no controlados requiere de mayor complejidad y de inteligencia

capaz de emplear varios tipos de reconocimiento de objetos dependiendo de la imagen detectada.

- Si bien un ambiente controlado no es imprescindible para el buen funcionamiento de este sistema de visión, al estar basado en la detección de un objeto de color, es recomendable usar un objeto de color vívido único en la escena y una fuente de iluminación homogénea que no deslumbre a la cámara cuando ésta la enfoque, la carencia de estos requerimientos dará como resultado un apuntamiento poco fiable.
- El sistema de visión desarrollado es una base para aplicaciones más específicas como por ejemplo conteo de objetos, clasificación por color, apuntamiento de cámaras de una resolución más alta, localización de vehículos no tripulados en un ambiente controlado y determinación de sus trayectorias, entre otras.
- Para mejorar la rapidez de este desarrollo en particular, se puede migrar hacia la familia de microcontroladores STM32F4xx, los cuales poseen mejores características, principalmente un reloj mucho más rápido, y hardware especializado para operaciones con números de punto flotante (floating point); para aumentar la resolución, la cantidad de colores y la calidad de imagen se puede emplear cámaras con mejores capacidades y de ser el caso diseñar el hardware para comunicarlas con el microprocesador.

REFERENCIAS BIBLIOGRÁFICAS

- [1] University of Colorado; **Physics 2000 (TV screens)**
http://www.colorado.edu/physics/2000/tv/big_picture.html
- [2] Wikipedia; **Charged Coupled Device**
http://en.wikipedia.org/wiki/Charge-coupled_device
- [3] Wikipedia; **Active Pixel Sensor**
http://en.wikipedia.org/wiki/Active_pixel_sensor
- [4] Universidad de Sevilla; **Fotografía Digital**
http://www.aloj.us.es/galba/DIGITAL/CUATRIMESTRE_I/DIGITAL/lacamara1.htm
- [5] Colorotate; **Color Theory**
<http://learn.colorotate.org/color-models/#.Uffnd21s9h4>
- [6] Cambridge InColour; **Camera Lens Corrections**
<http://www.cambridgeincolour.com/tutorials/lens-corrections.htm>
- [7] Atmel; **Using the AVR Hardware Multiplier**
<http://www.atmel.com/Images/doc1631.pdf>
- [8] ARM; **32-Bit Microcontroller Size Analysis**
http://www.arm.com/files/pdf/ARM_Microcontroller_Code_Size_%28full%29.pdf
- [9] STMicroelectronics; **STM32 32-bit ARM Cortex MCUs**
<http://www.st.com/web/en/catalog/mmc/FM141/SC1169>
- [10] STMicroelectronics; **Data Sheet DS5792: High-density performance line ARM-based 32-bit MCU with 256 to 512KB Flash, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces**

<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00191185.pdf>

[11] OmniVision; **OV7725 Advanced Information Preliminary Datasheet**

<http://www.haoyuelectronics.com/Attachment/OV7725%20+%20AL422B%28FIFO%29%20Camera%20Module%28V1.0%29/OV7725%20datasheet%28V1.2%29.pdf>

[12] AverLogic; **AL422B Data Sheets (Revision V1.01)**

<http://pdf1.alldatasheet.com/datasheet-pdf/view/140798/ETC1/AL422.html>

[13] Sólomon Systech; **Advance Information240 RGB x 320 TFT LCD Controller Driverintegrated Power Circuit, Gate and Source Driverwith built-in RAM**

http://aitendo3.sakura.ne.jp/aitendo_data/product_img/lcd/tft2/M032C1289TP/3.2-SSD1289.pdf

[14] STMicroelectronics; **Reference Manual RM008: STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs**

http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference_manual/CD00171190.pdf

[15] Planet Analog; **Powering resistive touch screens efficiently**

http://www.planetanalog.com/document.asp?doc_id=527515&site=planetanalog

[16] Ming Xu and Tim Ellis; **Color-Invariant Motion Detection under Fast Illumination Changes**

http://www.bmva.org/bmvc/2001/papers/74/accepted_74.pdf

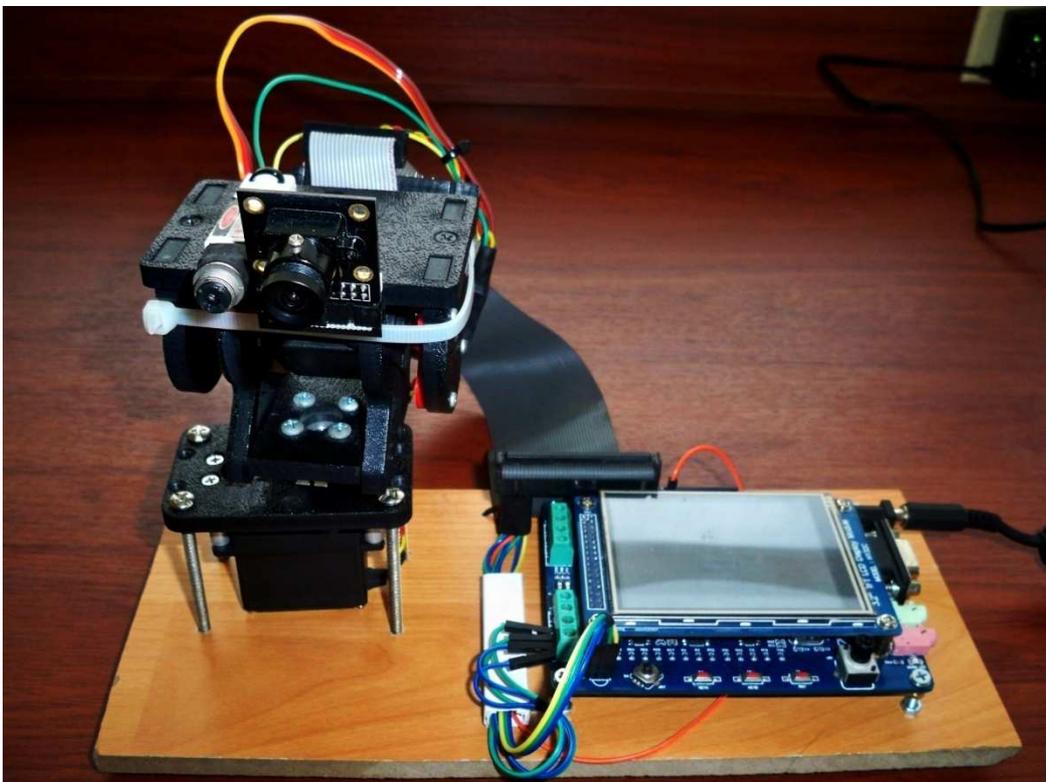
[17] C. Balkenius and B. Johansson, **Finding Colored Objects in a Scene**, Lund University
<http://www.lucs.lu.se/LUCS/M012/Minor12.pdf>

[18] Katsuhiko Ogata; **Modern Control Engineering**, University of Minnesota, 2002

ANEXOS

MANUAL DE USUARIO

El sistema de visión está orientado a detectar, localizar, y apuntar un objeto de color. Para esto hace uso de una cámara montada sobre una plataforma con dos servomotores los cuales realizan movimientos de rotación y guiñada (*Pan & Tilt*) a fin de satisfacer el apuntamiento. Mediante una pantalla LCD táctil el usuario puede seleccionar el objeto a ser detectado, configurar ciertos parámetros, así como también ver la imagen en tiempo real.



Gráfica 1 Sistema de visión en funcionamiento.

Componentes del sistema de visión

El sistema de visión básicamente hace uso de una cámara, la plataforma en la que se ubica, una pantalla LCD, y la tarjeta electrónica donde se encuentran el procesador y una sección con botones (ver Gráfica 1 y Gráfica 2).



Gráfica 2 Tarjeta electrónica del sistema de visión.

Botonera externa "JOY": este dispositivo contiene botones de dirección: arriba, abajo, izquierda, derecha, y centro, a manera de un "joystick"; se puede emplear este dispositivo para mover el cursor en la interfaz de usuario, así como también se puede emplear la pantalla táctil.

Botón externo "KEYA": este botón tiene la única función de detener el sistema de visión y volver a la página de inicio.

Botón externo "KEYB": este botón transmite la imagen presente en la pantalla mediante el puerto serial a la velocidad de transmisión de 921600 bits por segundo. La trama inicia con las letras: "IM:" y continúa con la transmisión de 153600 bytes correspondiente a los 320x240 píxeles en formato RGB565.

Botón externo "RST": al pulsar este botón el sistema se reinicia.

Interfaz de usuario

Al iniciar el sistema la pantalla LCD mostrará una página de inicio (ver Gráfica 3) en la cual se puede elegir entre: iniciar la cámara de video (Botón "**INICIAR**") o configurar las distintas opciones (Botón "**OPCIONES**").



Gráfica 3 Página de inicio.

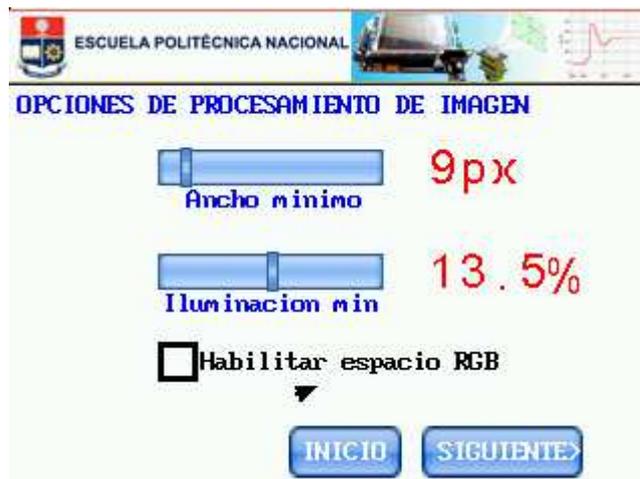
Configuraciones

Si se elige el Botón "**OPCIONES**" se despliega una página con configuraciones que afectarán la etapa de segmentación (el cómo reconocerá el sistema al objeto de color). Las configuraciones posibles son (ver Gráfica 4):

Ancho mínimo: determinará el ancho mínimo del objeto que se quiere reconocer, si el objeto de color no supera este ancho, simplemente se lo ignora. El ancho está dado en pixeles y su rango es de 3 hasta 103 pixeles, el valor predeterminado es de 9 pixeles.

Iluminación mínima: determinará la iluminación mínima en el objeto que se quiere reconocer, esto ayuda a descartar objetos oscuros no deseados que se encuentren en la imagen. El rango es de 1% hasta 26% del nivel de iluminación total (blanco), el valor predeterminado es del 13.5%.

Habilitación del espacio RGB: Habilita la búsqueda de color en espacio RGB, esto da como resultado un menor tiempo de procesamiento, y es recomendable usarlo en situaciones controladas de iluminación; se deshabilita esta opción, se emplea el espacio rg de cromaticidad (RGB normalizado), el cual tiene más robustez ante variaciones de luminosidad en el objeto buscado, pero introduce un mayor retardo de procesamiento. El valor predeterminado es deshabilitado (espacio rg de cromaticidad activado).



Gráfica 4 Página de opciones de procesamiento de la imagen.

Si se elige el Botón "**INICIO**" regresará a la página inicial, Botón "**SIGUIENTE**" abrirá otra página de configuraciones concernientes a la interpretación de la imagen (ver Gráfica 5).

Habilitar el apuntamiento: Habilita a los servomotores para que realicen el apuntamiento de la imagen. El valor predeterminado es deshabilitado.

Visualizar segmentación: Permite ver en tiempo real la imagen binaria obtenida de la segmentación, esto sirve de ayuda para visualizar que objetos en la imagen poseen el color que se quiere detectar. El valor predeterminado es habilitado.



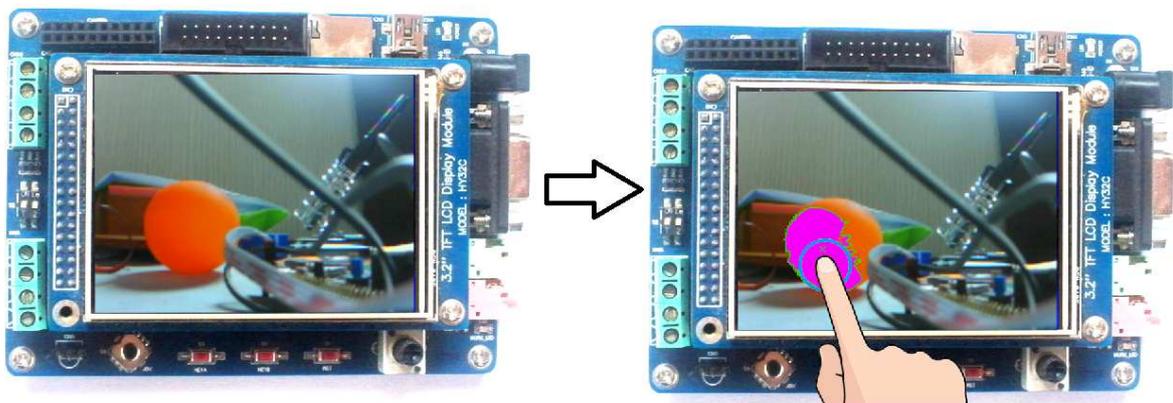
Gráfica 5 Página de opciones de interpretación de la imagen.

Funcionamiento

El sistema se pone en marcha si se elige el Botón "**INICIAR**" en la página inicial. Si se quiere detener el sistema y volver a la página inicial se lo puede hacer con el botón externo "**KEYA**".

Una vez inicializada la cámara de video el usuario debe ubicar en la pantalla táctil el objeto de color que se quiere detectar, como se ilustra en la Gráfica 6.

Establecido el contacto en la pantalla táctil se puede seguir haciendo contacto en la pantalla para seleccionar más variantes del color en el mismo objeto. Cuando se termine la selección inmediatamente el sistema de apuntamiento entra en acción; si se quiere seleccionar otro objeto se puede repetir el proceso.



Gráfica 6 Se selecciona el objeto de color anaranjado mediante la pantalla táctil.

CARACTERÍSTICAS DEL EQUIPO

Características eléctricas

Parámetro	Promedio	Máximo	Unidad
Voltaje de alimentación	5	5	V
Corriente de alimentación	200	*1100	mA
Frecuencia de operación	72	-	MHz
Tiempo de encendido	500	-	mS
Velocidad del puerto RS-232	921600	-	bps

*El consumo de corriente presenta picos debido al movimiento de los servomotores, el consumo será mayor si se encuentran obstruidos físicamente.

Dimensiones físicas

Parámetro	Valor	Unidad
Largo	30.0	cm
Ancho	13.0	cm
Altura	19.0	cm

Lista de costos

Elemento	Valor	Unidad
Tarjeta de desarrollo HY-Smart STM32 (con pantalla incluida)*	59.00	USD
Cámara OV7725 con memoria FIFO incluida*	30.00	USD
SPT200 Direct Drive Pan & Tilt System (servomotores incluidos)*	45.99	USD
Adaptador de 5V/1200mA	10.00	USD
Laser de apuntamiento 3.3V	2.00	USD
Cableado, base de madera, sujetadores, tornillos, tuercas, etc.	9.00	USD
TOTAL	155.99	USD

*No se listan los costos de transporte que pueden diferir de acuerdo a: el fabricante "ServoCity" para la plataforma de apuntamiento *SPT200*, y el fabricante "Micro4You" para la Tarjeta de desarrollo *HY-Smart STM32* y la Cámara *OV7725*.