



CENTRO DE INGENIERIA Y DESARROLLO INDUSTRIAL

CENTRO DE INGENIERIA Y DESARROLLO INDUSTRIAL

Proyecto Industrial Terminal

Control de una cámara fireware en ambiente LabVIEW

PARA OBTENER LA ESPECIALIDAD EN

005623

“TECNOLOGO EN MECATRONICA”

PRESENTA

Alumno: Ing. Carlos Alberto Paredes Orta

Tutor de Planta: Dr. Jesús Carlos Pedraza Ortega

Tutor Académico: Dr. Tomas Salgado Jiménez

QUERETARO, QRO. 2008

15/sep/2008

Vo. Bo.  
Dr. Tomas Salgado J



## **Resumen**

En este documento se presenta como se realizó el proyecto de configurar, caracterizar y desarrollar una aplicación en LabVIEW para la cámara Basler Scout cuyo protocolo de comunicación es vía FireWire. Esta cámara cuenta con 2 lentes: uno lente Fujinon TV HF9HA-1B de 1.4 / 9 mm este se ajusta manualmente y el lente Rainbow H6 X8M-II de 8 a 48 que es controlado por un servo controlador modelo V3LC. También se explica como se llegó a realizar dentro de la aplicación la configuración y la programación del servo controlador puesto que este se comunica con la computadora por el puerto serial, lo que nos permite tener un control más preciso y se puede ajustar el zoom, foco e iris desde la aplicación en LabVIEW.

Adicionalmente se realizó procesamiento de imágenes usando Vision Assistant 8.2.1 de LabVIEW. Las rutinas creadas fueron calibrar la imagen, rutina para la identificación de figuras y poder calcular su área, otra para determinar su posición y finalmente su color.

# INDICE

Resumen .....	2
1.-Antecedentes .....	8
2.-Definición del proyecto.....	8
3.-Justificación.....	8
4.-Objetivos .....	9
5.- MARCO TEORICO .....	9
5.1.-FireWire .....	9
5.2.-LabVIEW .....	10
5.3.-NI Vision Assistant .....	10
5.4.-Puerto serial .....	11
5.5.-Las Lentes de Cámara.....	12
5.6.-Concepto y modelización de la calibración .....	14
5.7.-Procedimientos de calibración.....	16
5.8.-Cámaras bajo el punto de vista de la calibración.....	16
5.9.-Relacion entre el equipo .....	17
6.-Descripción de los VI'S utilizados en este trabajo .....	20
6.1.-PUERTO SERIAL.....	20
6.2.-MANEJO DE CARACTERES .....	20
6.3.-OPERACIONES MATEMATICAS .....	21
6.4.-ESTRUCTURAS CASE .....	21
6.5.-VI DE VISION.....	22
7.-Programa de control de lente .....	26
7.1.- calibración del Zoom .....	33
7.2.-Calibracion del Focus.....	35

7.3.-Calibracion del Iris.....	36
7.4.- Ajuste Manual.....	37
7.5.-retroceso del zoom<.....	39
7.6.-retroceso del Focus <.....	40
7.7.-retroceso del Iris <.....	41
7.8.-adelanto del zoom >.....	41
7.9.-adelanto del Focus >.....	42
7.10.-adelanto del Iris >.....	42
7.11.-retroceso largo del zoom << .....	42
7.12.-retroceso largo del Focus << .....	43
7.13.-retroceso largo del Iris << .....	43
7.14.-adelanto largo del zoom >> .....	44
7.15.- adelanto largo del Focus >> .....	44
7.16.- adelanto largo del Iris >> .....	44
8.-Desarrollo de la programación de la calibración de la cámara y procesamiento de imagen.....	45
9.-Resultados .....	52
10.-Conclusiones y trabajo futuro .....	53
11.-Bibliografía .....	53

## INDICE DE FIGURAS

Fig 1.- Logotipo de FireWire.

Fig 2.- conector FireWire.

Fig 3.- Ejemplos de Aplicaciones de Visión Artificial

Fig 4.-Puerto en serie ATX

Fig 5.-Conector RS-23 hembra.

Fig 6.- lente comun

Fig 7.- Connection to computer and lens

Fig 8.- Cámara Scout

Fig 9.- Lente Fujinon

Fig 10.- Lente Rainbow

Fig 11.- Pestaña de Configuración de la cámara

Fig 12.- Pestaña de Configuración del puerto

Fig 13.- Pestaña de Calibración

Fig 14.- Pestaña de Velocidad del lente

Fig 15.- Ajuste del lente manual

Fig 16.- Programa Lens Controller

Fig 17.- ventana de propiedades del puerto serial

Fig 18.- ciclo for y escritura

Fig 19.- Extracción de los datos

Fig 20.- llenado de la barra de proceso

Fig 21.- convertir los valores a cadenas

Fig 22.- leer dos datos

Fig 23.-Adquisición de la posición actual

Fig 24.-adquisición de la velocidad

Fig 25.-secuencia de retroceso

Fig 26.-reenvió

Fig 27.- secuencia retroceso 2

Fig 28.-convertirlo a cadena  
Fig 29.-restarle 100  
Fig 30.- detección de la cámara  
Fig 31.- script creado en vision assistant  
Fig 32.conexión con la cámara  
Fig 33.- Desplegado de la imagen  
Fig 34.- imagen original  
Fig 35.ajuste brillo  
Fig 36.- imagen procesada  
Fig 36. Resultado de las medidas en cm  
Fig 37. Figuras encontradas  
Fig 38. finalizar la sesión de la cámara  
Fig 39. finalizar la sesión de la cámara  
Fig 40 Imagen original  
Fig 41. Imagen Resultado

005623

#### INDICE DE DIAGRAMAS

Diagrama 1.- Diagrama de conexión  
Diagrama 2: diagrama de flujo del lente  
Diagrama 3: diagrama de bloque de la conexión serial  
Diagrama 3: Diagrama de bloques del programa  
Diagrama 5: diagrama de bloques de la calibración del focus  
Diagrama 6: diagrama de bloques de la calibración del iris  
Diagrama 7: diagrama de bloques del ajuste manual  
Diagrama 8 diagrama de bloques de Pedir la posición  
Diagrama 9 diagrama de bloques de Decrementar la posición  
Diagrama 10 Diagrama de bloques del retroceso del focus  
Diagrama 11 Diagrama de bloques del retroceso del Iris  
Diagrama 12 Diagrama de bloques del adelanto del zoom

- Diagrama 13 Diagrama de bloques del adelanto del focus
- Diagrama 14 Diagrama de bloques del adelanto del Iris
- Diagrama 15 Diagrama de bloques del retroceso largo del zoom
- Diagrama 16 Diagrama de bloques del retroceso largo del focus
- Diagrama 17 Diagrama de bloques del retroceso largo del Iris
- Diagrama 18 Diagrama de bloques del adelanto largo del zoom
- Diagrama 19 Diagrama de bloques del retroceso largo del focus
- Diagrama 20 Diagrama de bloques del retroceso largo del Iris
- Diagrama 21 Diagrama de bloques de la identificación del color azul
- Diagrama 22 Diagrama de bloques de la identificación del color Rojo
- Diagrama 23 Diagrama de bloques de la identificación del color verde
- Diagrama 24 Diagrama de bloques de la separacion de los objetos
- Diagrama 25 Diagrama de bloques de la calibración y presentación de datos

## **INCIDE DE TABLAS**

- Tabla 1 configuración del puerto serial
- Tabla 2.- Tabla de comandos de entrada
- Tabla 3 .- Tabla de comandos de salida
- Tabla 4."resultados

## **1.-Antecedentes**

En la actualidad el CIDESI esta desarrollando un submarino con recursos de CONACYT y CIDESI el cual contiene una cámara LED Multi SeaCam con recubrimiento de titanium, en esta cámara se planea desarrollar la capacidad de medir piezas.

Se pretende que esta rutina pueda ayudar a los usuarios del Robot en la determinación del tamaño de los daños inspeccionados, para ello se tiene que calibrar y caracterizar la cámara.

Por el momento se cuenta con la cámara BASLER Scout y dos tipos de lentes el primero un lente Fujonon HF9HA -1B 1:1.4/9mm y el segundo un Rainbow H6 X 8M, II que esta controlado por un controlador VCL3 que se conecta a la computadora por el puerto serial, para poder controlar 3 motores y poder ajustar el zoom, contraste e iris. Esta cámara nos puede servir perfecta mente para diseñar un programa que cumpla con esas características siendo un base y marco de referencia para la programación de la cámara LED Multi SeaCam.

Aparte de que puede ser reutilizada esta cámara para otros fines investigación en procesamiento de imágenes ya que está caracterizada y calibrada.

## **2.-Definición del proyecto**

El proyecto consiste en caracterizar la cámara Basler Scout que se conecta a la computadora vía fireware desarrollado una aplicación en LabVIEW para su manejo, también en esa aplicación caracterizar el controlador VCL3 que es el que maneja el lente Rainbow H6 X 8M II haciendo que en el mismo programa se pueda manipular el lente y la cámara al mismo tiempo. También se realizo procesamiento de imágenes; se creo una rutina para identificar figuras y poder calcular su área, su posición y su color.

## **3.-Justificación**

En la actualidad el CIDESI esta desarrollando un submarino con recursos de CONACYT y CIDESI el cual contiene una cámara LED Multi SeaCam de titanium con esta cámara se planea medir superficies y piezas. Para ello el desarrollo de este proyecto sirve muy bien como referencia y soporte para poder realizar esta tarea de medición en un futuro.

Las técnicas de visión por computador tienen como fin último extraer propiedades del mundo a partir de un conjunto de imágenes. Estas nos sirven de muchas maneras, en este caso la medición de figuras y poder llegar a una exactitud que muchas veces por el ambiente (el agua) es difícil conocer o darse una idea del tamaño real de un objeto o falla por medio de una simple imagen.

Con este trabajo se busca sentar bases para implementar un algoritmo de medición en un futuro cercano.



## 4.-Objetivos

Caracterizar la cámara BASLER Scout y sus dos lentes el primero un lente manual marca Fujinon HF9HA -1B 1:1.4/9mm y el segundo un lente automático Rainbow H6 X 8M, II.

Crear un programa para el control del lente Rainbow H6 X 8M, II que se controla vía RS-232 con el servo controlador VCL3.

Diseñar un programa en LabVIEW que obtenga los parámetros geométricos y la localización de figuras definidas (círculos, cuadrados, rectángulos) y determinar las medidas (con un error aceptable) de dichas figuras.

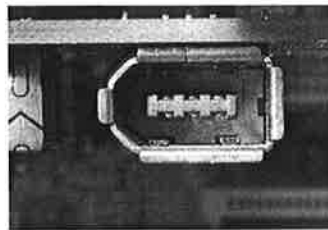
## 5.- MARCO TEORICO

En esta sección se realizaron definiciones que serán usadas en este trabajo.

### 5.1.-FireWire



**Fig 1** Logotipo de FireWire.



**Fig 2** .- conector FireWire.

FireWire es uno de los estándares de periféricos más rápidos desarrollados hasta el momento, y resulta ideal para el uso de periféricos multimedia como las videocámaras y otros dispositivos de alta velocidad, por ejemplo las unidades de disco y las impresoras de última generación.

FireWire es ya la interfaz más utilizada para audio y video, y entre sus ventajas se cuentan la alta velocidad, una amplia conectividad y que admite la conexión de hasta 63 dispositivos.

#### 5.1.1.-Ventajas de FireWire

Las ventajas de FireWire pueden resumirse en tres palabras: velocidad, velocidad y más velocidad. A 400 Mbps, tiene un ancho de banda 30 veces superior al de USB, lo que lo convierte en la opción ideal para el almacenamiento de alta velocidad y la captación de video profesional. Estas son otras de sus ventajas:

Admite un máximo de 63 dispositivos con cables de hasta 4,25 metros.  
▪ FireWire es "(des)conectable en uso", lo que significa que no necesitas desactivar un escáner o

una unidad de CD para conectarlo o desconectarlo, y que no necesitas reiniciar la computadora.

- Los cables de FireWire son fáciles de conectar: no necesita identificadores del dispositivo, puentes, interruptores DIP, tornillos, cierres ni terminadores.

Tomado de : <http://www.apple.com/la/firewire/>

### 5.2.-LabVIEW

LabVIEW es una herramienta gráfica para pruebas, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje gráfico G.

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, Mac y Linux y va por la versión 8.20 y 8.21 con soporte para Windows Vista.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, lo que da una idea de su uso en origen: el control de instrumentos. El lema de LabVIEW es: *"La potencia está en el Software"*. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a programadores no expertos. Esto no significa que la empresa haga únicamente software, sino que busca combinar este software con todo tipo de hardware, tanto propio -tarjetas de adquisición de datos, PAC, Visión, y otro Hardware- como de terceras empresas.

Tomado de : <http://es.wikipedia.org/wiki/LabVIEW>

### 5.3.-NI Vision Assistant

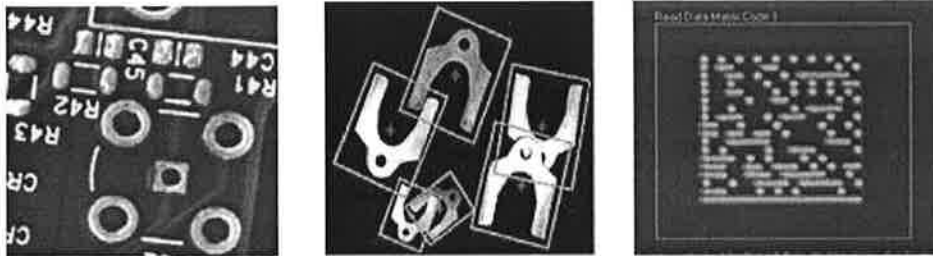
Por más de 30 años, National Instruments ha sido líder en medición y automatización, proporcionando poderosas plataformas flexibles de hardware y software para ingenieros y científicos alrededor del mundo. Hace más de una década, NI agregó visión artificial a su línea de productos. La plataforma de visión artificial de NI incluye hardware desde dispositivos conectados a sistemas PCI y PXI a procesamiento de imágenes en el sensor mismo con la nueva NI Smart Camera. Las opciones de software incluyen software de adquisición de imágenes para adquirir imágenes desde miles de cámaras, una librería de procesamiento de imagen de clase mundial, y una interfaz configurable para aplicaciones industriales de visión artificial.

¿Por qué Visión Artificial?

Ingenieros y científicos utilizan visión artificial en lugar de otros sensores tradicionales y herramientas de medición ya que les ofrece habilidades únicas no encontradas en muchas de las herramientas tradicionales. Considerando, por ejemplo, una banda transportadora con objetos que se mueven a 300 partes por minuto, y usted desea tomar una simple medición de ancho de los objetos transportados en la línea de producción. Con las herramientas tradicionales usted necesitará utilizar un calibrador Vernier para obtener mediciones precisas. Sin embargo, a usted no le será posible tomar las mediciones de una manera efectiva para cada pieza, por lo que necesitará muestrear y medir una pieza cada 100 unidades, dándole a su sistema 20 segundos por pieza para remover el objeto de la banda transportadora, medirla y reemplazar el artículo. También existe el riesgo de dañar el objeto mientras que el operador o el robot trate de remover y reemplazar esta pieza en la banda transportadora.

Con visión artificial, usted puede inspeccionar cada pieza en lugar de muestrear en forma aleatoria y esperando que el resto de las piezas estén bien. Debido a que las herramientas de visión artificial

no son destructivas, usted puede asegurar que ninguna de sus piezas sea dañada en el proceso de medición mientras que logra mediciones de alta precisión.

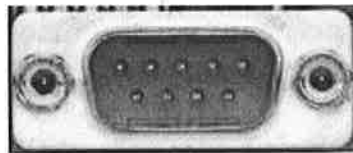


**Fig 3.- Ejemplos de Aplicaciones de Visión Artificial**

Muchas otras tareas no pueden ser realizadas efectivamente o no pueden ser realizadas del todo con sensores tradicionales. Verificar etiquetas, contar píldoras que entran a una botella, monitoreo de velocidad de balas, leer o verificar textos en empaques, leer códigos de barra de una y dos dimensiones, y muchas otras aplicaciones son tareas simples si usted cuenta con software y hardware de visión artificial lo suficientemente poderosos para realizar las mediciones.

Tomado de: <http://zone.ni.com/devzone/cda/tut/p/id/6908>

#### 5.4.-Puerto serial



**Fig 4.- Puerto en serie ATX**

Un puerto de serie es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, en donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente. La comparación entre la transmisión en serie y en paralelo se puede explicar con analogía con la carreteras. Una carretera tradicional de un sólo carril por sentido sería la como la transmisión en serie y una autovía con varios carriles por sentido sería la transmisión en paralelo, siendo los coches los bits.

RS-232 (también conocido como Electronic Industries Alliance RS-232C) es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (*Data Communication Equipment*, Equipo de Comunicación de datos), aunque existen otras situaciones en las que también se utiliza la interfaz RS-232.



Fig 5.-Conector RS-23 hembra.

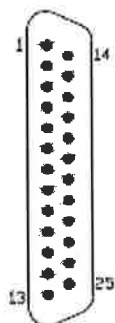
En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos, como pueden ser computadores. Evidentemente, en el caso de interconexión entre los mismos, se requerirá la conexión de un DTE (*Data Terminal Equipment*) con otro DTE.

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DB-9), más barato e incluso más extendido para cierto tipo de periféricos.

Tomado de : [http://es.wikipedia.org/wiki/Puerto\\_serie](http://es.wikipedia.org/wiki/Puerto_serie)

Para el programa esta relacionado el puerto serial de esta manera :

### Connectors DB25S



Pin	Signal Type	Signal Name - Left Connector	Signal Name - Right Connector
1	6, 8 or 12 VDC	Motor 0 (zoom) Command Voltage +	Motor 3 (zoom) Command Voltage +
14	6, 8 or 12 VDC	Motor 0 (zoom) Command Voltage -	Motor 3 (zoom) Command Voltage -
2	6, 8 or 12 VDC	Motor 1 (focus) Command Voltage +	Motor 4 (focus) Command Voltage +
15	6, 8 or 12 VDC	Motor 1 (focus) Command Voltage -	Motor 4 (focus) Command Voltage -
3	6, 8 or 12 VDC	Motor 2 (iris) Command Voltage +	Motor 5 (iris) Command Voltage +
16	6, 8 or 12 VDC	Motor 2 (iris) Command Voltage -	Motor 5 (iris) Command Voltage -
4	User signal	Auxiliary 0 (Left) Relay Normally Open	Auxiliary 1 (Down) Relay Normally Open
17	User signal	Auxiliary 0 Relay Common	Auxiliary 1 Relay Common
5	DAC	Future Use (Analog Voltage Output)	Future Use (Analog Voltage Output)
18	User signal	Auxiliary 0 (Right) Relay Normally Closed	Auxiliary 1 (Up) Relay Normally Closed
6	0 - 5 VDC	Iris potentiometer input	Iris potentiometer input
19	GND	Ground	Ground
7	0 - 5 VDC	Focus potentiometer input	Focus potentiometer input
20	GND	Ground	Ground
8	0 - 5 VDC	Zoom potentiometer input	Zoom potentiometer input
21	0 - 5 VDC	VREF; voltage to one side of pots	VREF; voltage to one side of pots
9	0 - 5 VDC	VREF; voltage to one side of pots	VREF; voltage to one side of pots
22	0 - 5 VDC	VREF; voltage to one side of pots	VREF; voltage to one side of pots
10	6-12 VDC	VLENS (may be used for autoiris lenses)	VLENS (not normally used)
23	GND	Ground	Ground
11	VFAN / VTILT	Auxiliary 0 (Pan) Source Voltage	Auxiliary 1 (Tilt) Source Voltage
24	GND	Ground	Ground
12	0 - 5 VDC	Aux potentiometer input - pan	Aux potentiometer input - tilt
25	GND	Ground	Ground
13	N/C	Future Use	Future Use

Tabla 1 configuración del puerto serial

### 5.5.-Las Lentes de Cámara

El propósito de una lente de cámara está transmitir rayos de luz en la cámara y los enfocarlos para formar una imagen intensa sobre la película. Con las cámaras más simples no se requiere ningún ajuste de enfoque manual. Todos los objetos, desde unos pies en la frente de la cámara a una gran distancia, se registrarán en una imagen razonablemente intensa. Las cámaras más avanzadas, sin

embargo, tienen un control de enfoque que ajusta la lente para que los objetos a una distancia dada desde la cámara estén en el foco en la película.

Con algunas cámaras los focos se toman simplemente por estimación del fotógrafo de la distancia al objeto y colocando el enfoque en una escala en la cámara marcada en números o distancias. Con cámaras de telémetro el fotógrafo ajusta una aguja que se encuentra en la parte de arriba con respecto a otras dos fuera del objeto. Con cámaras reflex la lente se ajusta hasta que la imagen sobre la pantalla del ocular aparezca en el foco con respecto a un pequeño círculo.

Algunas cámaras se equipan con un sistema de microchip, que enfoca utilizando señales ópticas, infrarrojas, o ultrasónicas. El fotógrafo activa el sistema al poner una leve presión sobre el disparador, y centrar el objeto mediante el ocular. El sistema de la cámara mide la distancia al objeto y automáticamente ajusta consiguientemente la lente.

Tipos y propiedades de lentes.

El tipo más simple de lente es el menisco, un simple pedazo curvado de vidrio óptico o plástico moldeado. Todas las lentes sufren de aberraciones, o imperfecciones. Estas pueden ser su alcance y nitidez de imagen. La calidad mejora apreciablemente, al montar dos o más elementos de lente individuales como una unidad. Las lentes más complejas pueden tener una docena o más de tales elementos. En la mayoría de las lentes de buena calidad, estos elementos se revisten con una capa delgada del fluoruro de magnesio o alguna otra sustancia para reducir reflejos internos.



**Fig 6.- lente comun**

Las lentes tienen longitudes focales fijas. La longitud focal de una lente es la distancia desde el centro óptico de la lente al plano (en este caso, la película) detrás de la cual la imagen se forma más intensamente cuando la lente se enfoca en un objeto distante. Un objeto distante se dice que esta en el infinito. La longitud focal de una lente se da a veces en pulgadas, pero usualmente se da en milímetros (mm) o centímetros (cm).

Para el trabajo fotográfico ordinario, las cámaras se equipan con una lente de longitud focal de "norma" o "normal" . Esta longitud es aproximadamente el valor de la diagonal del marco de la película. Las lentes normales no equiparan comúnmente a la diagonal de la película exactamente. Por ejemplo, en la diagonal de una norma 35 - mm el marco del negativo es 43.2 mm, pero las lentes con una longitud focal desde 38 mm a 58 mm son normales para este formato.

Una lente que tiene una longitud focal considerablemente más corta que el de una lente normal se llama un objetivo gran angular. Provee una imagen menor del objeto pero cubre un campo de vista más amplio. Se usa frecuentemente cuando el fotógrafo quiere incluir más de la escena original en el cuadro pero es incapaz de ir más lejos del objeto, como, por ejemplo, cuando tomando un cuadro en una sala pequeña. Un objetivo gran angular puede dar también un cuadro con un efecto de perspectiva exagerada y de gran profundidad. Para cámaras de 35 - mm, los objetivos gran angular se encuentran disponibles con longitudes focales que van desde 35 mm a 15 mm.

Las lentes con una longitud focal que es considerablemente más larga que la normal se llaman de foco largo u objetivos de acercamiento (teleobjetivos). Técnicamente estos no son idénticos. Las lentes de foco largo son abultadas; los objetivos de acercamiento se construyen para ser más compactas. En la práctica, sin embargo, las lentes de foco largo de todos los tipos son indiscriminadamente llamadas telefoto. Tales lentes proveen una imagen más grande del objeto pero cubren un campo más estrecho de visión de lo que lo hacen las lentes de longitud focal más corta. Los objetivos de acercamiento pueden usarse para magnificar un objeto distante, por esto se usan frecuentemente en la fotografía de fauna silvestre. En el retrato, un objetivo de acercamiento intermedio puede minimizar distorsión de perspectiva, y en la fotografía de paisaje un objetivo de acercamiento puede aplanar una escena para lograr un efecto más bidimensional.

Con una lente convencional la longitud focal es fija, pero una lente de zoom provee al fotógrafo una gama continua de longitudes focales. Hay zoom de ángulo ancho, zoom telefoto, y zooms versátiles que pasan de ángulos anchos, longitudes focales normales, y la gama del telefoto.

Las macro y micro lentes están especialmente diseñadas para tomar fotografías de aproximación de objetos. Las lentes ojo pescado son ultra anchas, son lentes angulares que dan una imagen de la escena deformada como si se encontrara en una "pecera".

Tomado de: <http://www.textoscientificos.com/fotografia/camara>

### *5.6.-Concepto y modelización de la calibración*

En la formación de la imagen intervienen tanto la cámara como las lentes, por tanto como primer paso debemos establecer un modelo matemático de la cámara -su geometría- y como segundo modelizaremos el efecto distorsionador de las lentes.

a) Modelo geométrico de la cámara la conversión geométrica del objeto tridimensional en un objeto 2D en el plano imagen de la cámara obedece las leyes de la proyección perspectiva o proyección central, en la que los rayos de luz procedentes del objeto pasan a través del centro de proyección o centro óptico de la cámara y se plasman en la película o el sensor. Esta proyección es el modelo de la cámara más básica, "pinhole", o cámara oscura.

La formulación matemática, básicamente, trata de relacionar linealmente las coordenadas tridimensionales del punto en el mundo real  $(X,Y,Z)$  con sus coordenadas 2D en el plano imagen  $(x,y)$ . Si usamos coordenadas homogéneas se define la matriz de transformación de perspectiva en la notación matricial como:

Los parámetros a calibrar según este modelo serían las coordenadas del punto principal y la focal. En algunas notaciones el primero se calcula en función del formato del sensor o el negativo y en otras se toma respecto al teórico origen (0,0) del punto principal.

b) Modelo de la distorsión

Si los componentes ópticos de la cámara fueran perfectos, la transformación entre la imagen bidimensional y el objeto tridimensional en el espacio sería perfectamente lineal y fácil de resolver. Pero las lentes del sensor producen distorsiones que no son lineales y que afectan a la precisión de la transformación.

La distorsión radial desplaza los puntos de la imagen radialmente a partir del centro, mientras que la tangencial o descentrada los desplaza perpendicularmente a la línea radial. La causa de la primera es el incorrecto pulido de la lente, mientras que de la segunda es la falta de alineación entre los componentes ópticos.

A éstas hay que añadir la falta de planaridad de dos planos: el de la propia imagen y el del plano focal. Estos dos problemas desaparecen en el caso de cámaras digitales, y en la práctica, la única distorsión que produce verdaderos problemas es la radial. Veamos algunas formulaciones de ésta:

La ISPRS propone el llamado modelo "balanceado":

$$dr = a_1 r(r^2 - r_0^2) + a_2 r(r^4 - r_0^4) \quad (1)$$

dr es la distorsión radial considerada como el desplazamiento, medido a lo largo de un radio del plano imagen, entre la posición de un punto en la imagen y su posición ideal según el modelo matemático; r es el radio (también llamado distancia radial) desde el punto al punto principal y por último,  $r_0$ , el valor escogido de dicho radio para el cual la distorsión se anula. El valor de  $r_0$  es arbitrario, y obedece a un criterio según el cual se asume que para algún punto de la imagen la posición real coincide con la posición matemática. De esta forma la ecuación de la distorsión (1) adopta una forma más suavizada y unos valores menos comprometidos. Esta ecuación proviene de:

$$dr = a_0 r + a_1 r^3 + a_2 r^5 \quad (2)$$

dr = en la que se define ese valor  $r_0$  para el que la distorsión se anula,  $dr=0$ , de forma que:

$$a_0 = -a_1 r_0^2 - a_2 r_0^4 \quad (3)$$

Frente a este modelo se utiliza el modelo "gaussiano" definido por:

$$dr = K_1 r^3 + K_2 r^5 \quad (4)$$

Ambos son modelos polinomiales que dependen de la distancia focal, que a su vez es desconocida, así que formulamos el modelo restringiendo los valores de la distorsión para un valor prefijado de aquella, por ejemplo el ya citado de que una cierta distancia focal ya calibrada (distancia principal) anule la distorsión radial para una distancia radial convenida; o que minimice la suma de los valores al cuadrado de la distorsión, etc.

### *5.7.-Procedimientos de calibración*

Desde el punto de vista operativo se recurre a distintas alternativas:

1. Establecer una transformación entre las coordenadas imagen 2D y las coordenadas del mundo real 3D. A este concepto responden los conocidos métodos de Tsai, Faugueras y Toscani, Song de Ma, etc. Ello da pie a numerosos procedimientos que utilizan, en definitiva, puntos de coordenadas reales conocidas mediante técnicas ajenas a la fotografía (topografía, geodesia) y cuyas coordenadas imagen también lo son.

2. Fotografiar patrones-objeto con características o restricciones geométricas tales como colinealidad, coplanaridad, ortogonalidad, puntos de fuga, etc. que facilitan la calibración sin necesidad de realizar mediciones auxiliares. Este método también se cita como el método "test field" de calibración.

3. Utilizar haces proyectivos conocidos en su forma y en su posición. La más popular aplicación de este método conceptual es el procedimiento del multicolimador, utilizado sobre todo en calibración de cámaras aéreas en laboratorio.

### *5.8.-Cámaras bajo el punto de vista de la calibración*

El proceso de calibración con respecto a las cámaras fotográficas determina una clasificación inmediata de las mismas:

**Cámaras métricas:** aquellas en las que el fabricante realiza una calibración de precisión en laboratorio (normalmente por el método del multicolimador), y ofrece un certificado de los valores de la distorsión para unos ciertos valores de focal. Ello conlleva, además, la estabilidad de los elementos ópticos de la cámara. Ninguna cámara "autofoco" es una cámara métrica.

**Cámaras semimétricas:** se sitúan entre las dos anteriores. Son cámaras más estables que las anteriores, a las que se le añade, entre las lentes y el plano de la película, una retícula (réseau) con unas marcas fiduciales perfectamente conocidas. De manera que cada vez que se use la cámara se pueden utilizar estas marcas impresas en la película para determinar la posición del punto principal y la deformación del plano de la película a la vez que se realiza el resto del proceso fotogramétrico. En el caso de cámaras digitales, la placa del sensor, si tiene la suficiente garantía de fabricación, puede interpretarse como una retícula formada por la matriz de píxeles de coordenadas imagen dadas por la fila y la columna de cada píxel.



Y a la inversa, según el tipo de cámara podemos recurrir a distintos métodos de calibración. Podemos encontrar una comparativa entre cámaras y métodos de calibración en García y Cuartero (2002). Por lo que respecta al software de calibración, encontramos un estudio de varias aplicaciones en Wiedemann et al. (2001).

### 5.9.-Relacion entre el equipo

El equipo está relacionada de la siguiente manera, el controlado VCL3 va conectado al lente y el lente esta fijo en la cámara, como se muestra en el diagrama siguiente :



Diagrama de conexión con la PC:

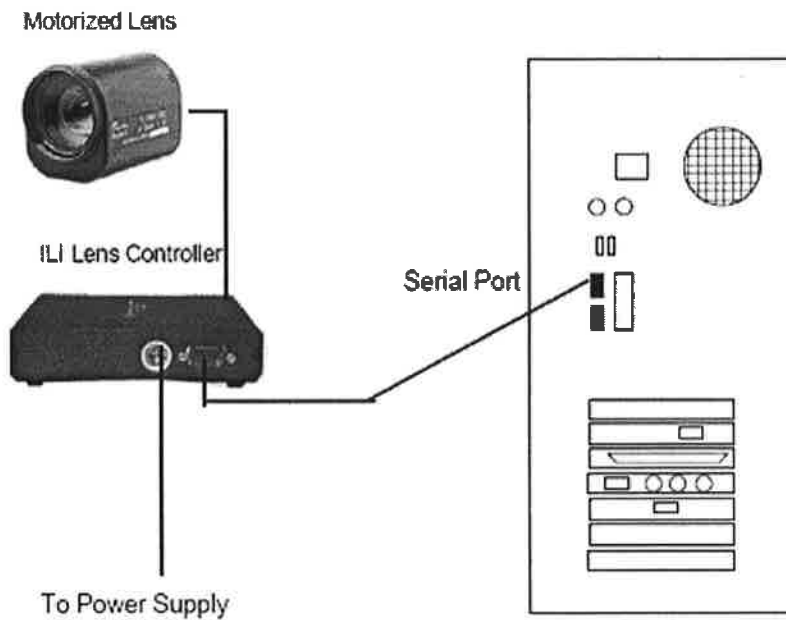


Figure 7: Connections to computer and lens

### Carateristicas Camara Scout :

Modelo : scA640-70fc

Sensor Size : 658 x 492

Max Frame Rate(at full resolution) : 71

Camara a color

Optical Size: 1/2"

Píxel Size (H x V): 9.9 x 9.9

Image Data Output Type: IEEE 1394b

Min Setting the gain: 280

Max Setting the gain (8bit depth): 1023

Max Setting the gain (12 depth): 511

For Gain settings from 280 to 511:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10} \left( \frac{658 + \text{Gain Setting}}{658 - \text{Gain Setting}} \right) - G_c$$



Fig 8.- Cámara Scout

**Carateristicas Lente Fujinon:**

Focal Length : 6mm

Iris Range : F1.2 ~ F16

Focus: manual

Iris : manual

Angle of view:

1/2 " 56"09' X 43"36'

1/3" 43"36' X 33"24'

1/4" 33"24' X 25"22'

Back Focal Distance: 11.44



**Fig 9.- Lente Fujinon**

**Carateristicas Lente Rainbow H6X 8M- II:**

Focal Length : 8~ 48 mm

Max. Relative Apertuare: 1:1.0

Iris: F1.0~Close

Zoom Ratio: 6X

Angular Field of view:

1/2": 43.6° x 33.4° at 8mm  
7.7° x 5.7° at 48mm

1/3": 33.4° x 25.4° at 8mm  
5.7° x 4.3° at 48mm

**Motorized Lens**

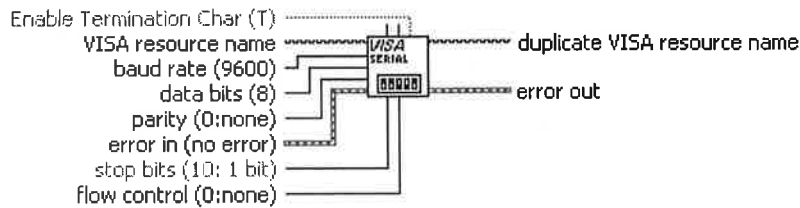


**Fig 10.- Lente Rainbow**

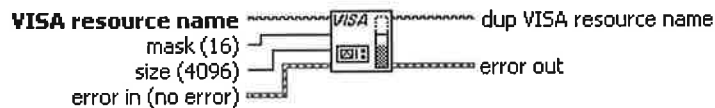
## 6.-Descripción de los VI'S utilizados en este trabajo

### 6.1.-PUERTO SERIAL

**VISA Configure Serial Port:** (Configurar Puerto Serie VISA).- Para utilizar el puerto serie es necesario primero configurarlo con los diferentes parámetros que hacen que funcione correctamente.



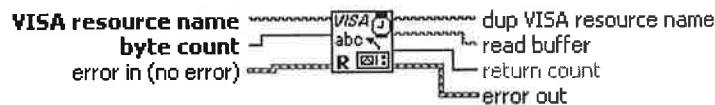
**VISA Set I/O Buffer Size:** da un tamaño del buffer



**VISA Write:** (Escribir en el puerto VISA).- Escribe un dato de tipo string en el puerto serie que previamente fue configuro en el puerto serie VISA.



**VISA Read:** (lee del puerto VISA).- lee un dato de tipo string del puerto serie ya que previamente se configuro el puerto serie VISA.

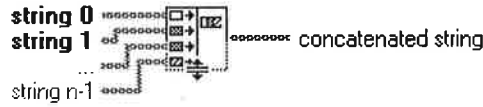


**VISA Close:** (Cerrar VISA).- Cierra el puerto VISA, se recomienda cerrarlo después de usarlo para evitar errores.

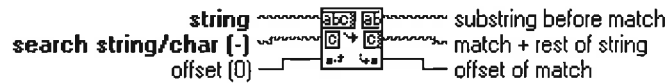


### 6.2.-MANEJO DE CARACTERES

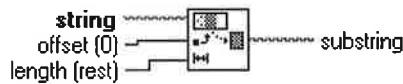
**Concatenate Strings:** concatena una serie de cadenas



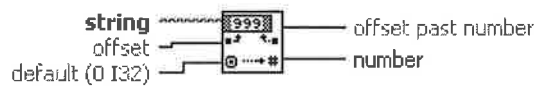
**Search / Split String :** busca valores en una cadena y si los encuentra manda su posición .



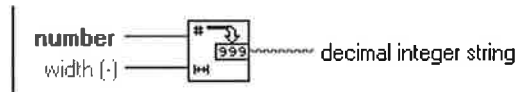
**String Subset:** crea una sub cadena del tamaño que uno quiera y empezando de la posición que uno dese.



**Decimal String To Number :** Convierte una cadena en número

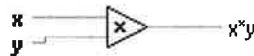


**Number To Decimal String :** Convierte un número en una cadena .



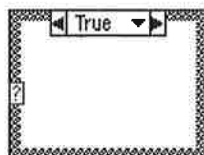
### 6.3.-OPERACIONES MATEMATICAS

**Multiply :** multiplica 2 números

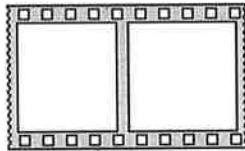


### 6.4.-ESTRUCTURAS CASE

**Case Structure:** (Estructura en caso de).- Es una estructura en la que si la condición es verdadera se ejecuta el programa que este en la ventana de TRUE pero si es falsa se ejecutara lo que se encuentre en la ventana de FALSE.



**Flat Sequence Structure:** Consiste de una o más subdiagramas, o marcos, que ejecutan secuencialmente operaciones.

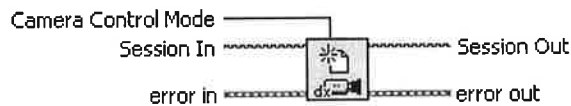


**Feedback Node:** transferencia de un valor de una cluster a, for, case a otro.

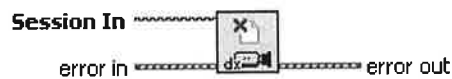


### 6.5.-VI DE VISION

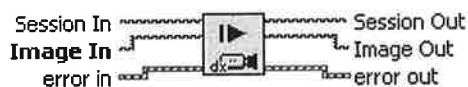
**IMAQdx Open Camera:** Se crea una sesión con la cámara.



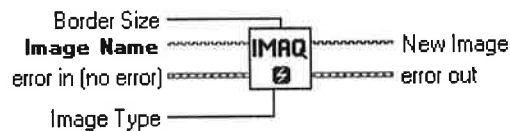
**IMAQdx Close Camera:** Finaliza la sesión de la cámara



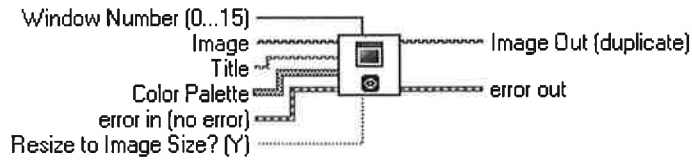
**IMAQdx Snap:** Toma una imagen con la cámara



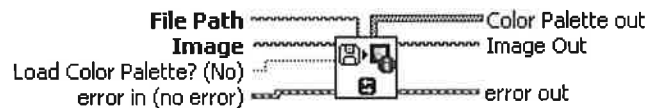
**IMAQ Create:** Crea una imagen



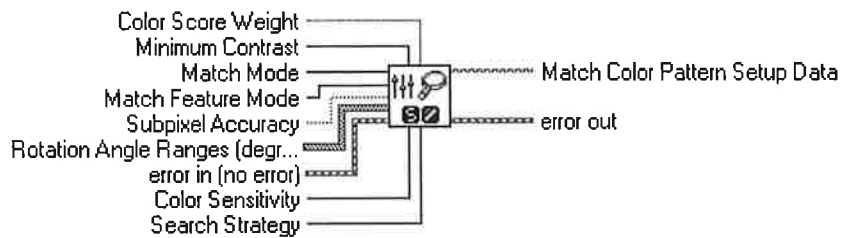
**IMAQ WindDraw:** muestra una imagen en una ventana de imagen. La imagen de la ventana aparece automáticamente cuando el VI se ejecuta.



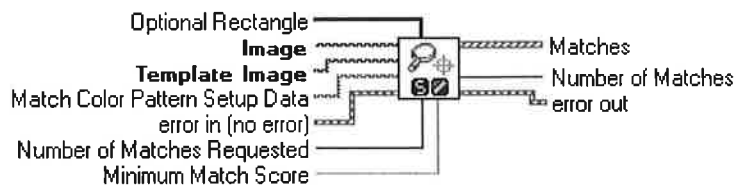
**IMAQ Read Image And Vision Info** : Lee un archivo de imagen, incluyendo cualquier información adicional visión guardado con la imagen. Esto incluye la superposición de información, patrones información plantilla, y la calibración de información, como está escrito de IMAQ Write Imagen y Visión Info.



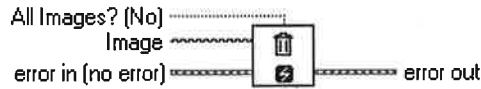
**IMAQ Setup Match Color Pattern** : Establece los parámetros que se utilizan durante el proceso de correlación de colores. Ejecutar este VI antes de usar el IMAQ Match Color Patrón VI.



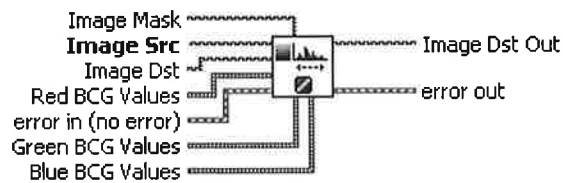
**IMAQ Match Color Pattern**: La búsqueda de un patrón de color, color o imagen plantilla, en la imagen de entrada. Utilice el color IMAQ Aprenda Plan VI antes de usar este VI para garantizar que la plantilla de imagen en color ha sido configurado para coincidir con el color.



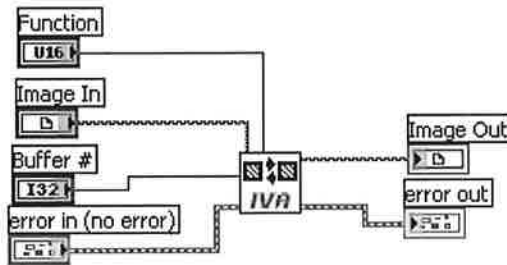
**IMAQ Dispose** : Destruye una imagen y libera el espacio que ocupa en la memoria. Este VI es necesario que cada imagen creada en una solicitud para liberar la memoria asignada a IMAQ Crear.



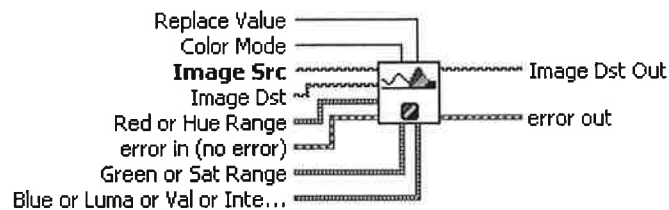
**IMAQ ColorBCGLookup**: Aplica un brillo, contraste, gamma y la corrección de color a cada plano por separado.



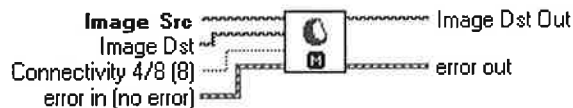
**IVA IMAGE BUFFER** : Es una aplicación que nos permite guardar en el buffer o sacar una imagen del buffer.



**IMAQ ColorThreshold** : Se aplica un umbral para los tres planos de RGB o HSL imagen y los sustituye en una imagen de 8 bits.

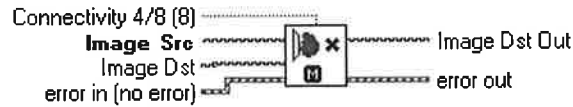


**IMAQ Convex Hull**: Señala el casco convexo para cada partícula en la imagen.

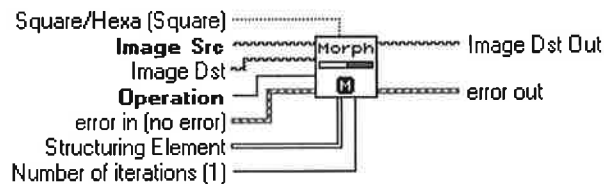


**IMAQ RejectBorder**: Elimina las partículas que tocan el borde de una imagen. La fuente de imagen debe ser una de 8 bits binarios imagen.

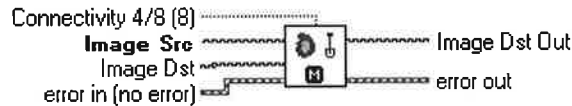




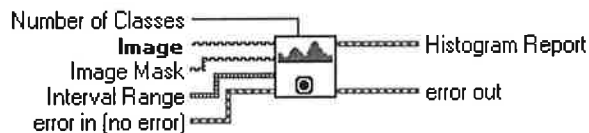
**IMAQ Morphology:** Realiza primaria transformaciones morfológicas primarias. Todos de origen y destino, las imágenes deben ser de una imágenes de 8 bits. La fuente conectada imagen para una transformación morfológica debe haber sido creado con una frontera capaz de soportar el tamaño del elemento estructurante. A  $3 \times 3$  estructuración elemento requiere un mínimo de 1 frontera, a  $5 \times 5$  estructuración elemento requiere un mínimo de 2 de frontera, y así sucesivamente. La frontera tamaño de la imagen de destino no es importante.



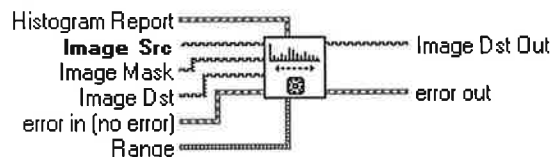
**IMAQ Fill Hole :** Llena los agujeros se encuentran en una partícula. Los agujeros se rellenan con un valor de píxel 1. La fuente de imagen debe ser una de 8 bits binarios imagen. Los agujeros se encuentran en contacto con la imagen de la frontera nunca se llena porque es imposible determinar si estos agujeros son parte de una partícula.



**IMAQ Histogram:** Calcula el histograma de la imagen.

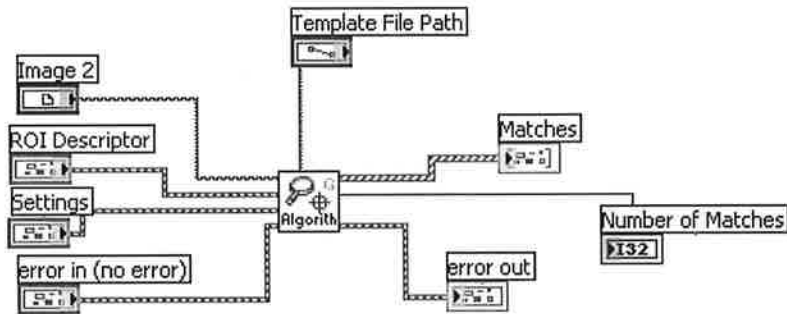


**IMAQ Equalize:** Produce una ecuilización del histograma de la imagen. Este VI redistribuye los valores de píxel de una imagen a línea rizar el histograma acumulado. La precisión del VI depende de la precisión histograma, que a su vez depende del número de clases utilizado en el histograma.

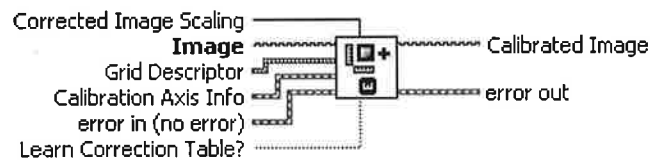


**IVA Match Geometric Pattern Algorithm :** Este VI es generado por visión assistant y lo que hace es buscar una figura que se le da en rango de la imagen y localizarla sin importar que este

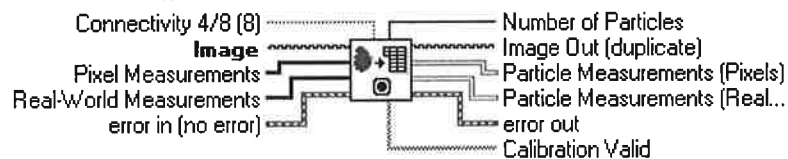
volteada o a diferente escala, después de localizarla la guarda en un arreglo con el varios datos : posición , número de veces que se encontró, tamaño de pixeles.



**IMAQ Set Simple Calibration :** Establece una simple calibración de una imagen o restablece el mundo real sistema de coordenadas en una imagen calibrada. Cuando se utiliza para restablecer un sistema de coordenadas, si la corrección de mesa sigue siendo necesaria, la tabla debe ser retroalimentada. Cuando el ángulo relativo a la horizontal se ajusta a 0, no es necesario hacer una corrección.



**IMAQ Particle Analysis:** Devuelve el número de partículas detectadas en una imagen binaria 2D y una serie de mediciones sobre pidió a la partícula.



## 7.-Programa de control de lente

Este programa se desarrollo en 2 partes; la primera fue crear la funcion del manejo del lente y la otra fue desarrollar las funciones de procesamiento de imagen y calibración de la cámara.

**Descripción del programa para manejar el lente:**

**Pantallas:** las figuras siguientes muestra la pantalla de control del programa desarrollado.



**Fig 11 .-** Pestaña de Configuración de la cámara

En esta pestaña se puede configurar lo que es el puerto serial a que velocidad va a recibir datos y que tamaño queremos el buffer

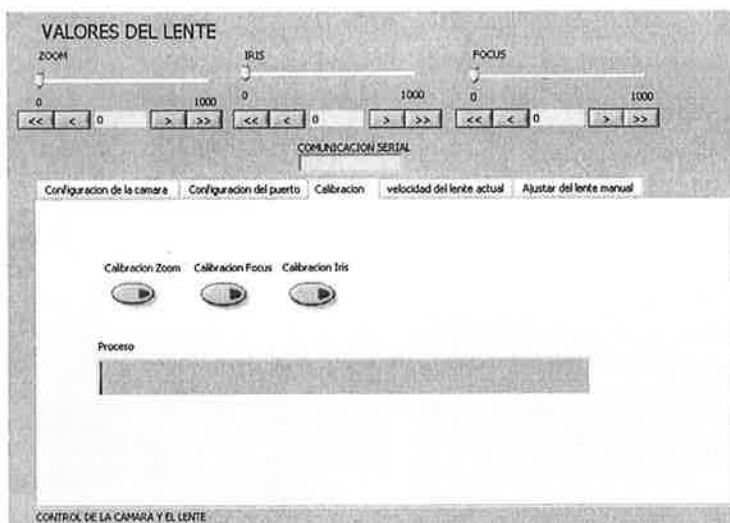
Como podemos ver lo primero que se muestra son los valores del lente en que posición se encuentra y la opción de moverlo.

En esta pestaña se controla la cámara se puede ajustar la cámara también tiene 3 perillas las cuales son para ajustar el contraste, brillo y gamma.



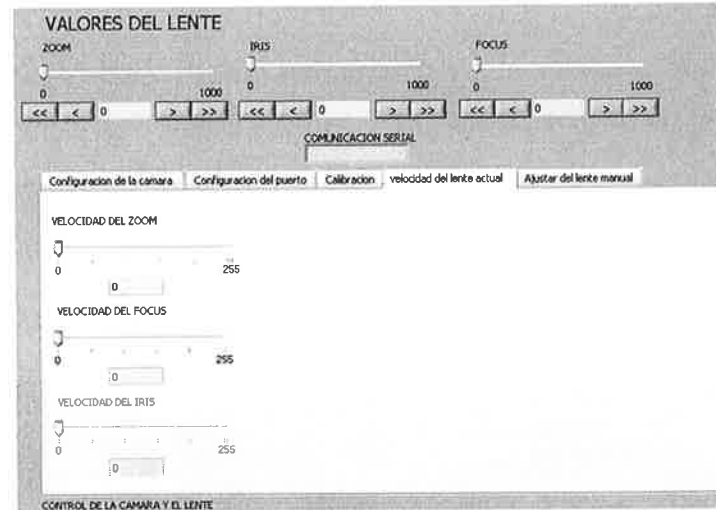
**Fig 12.-** Pestaña de Configuración del puerto

En esta pestaña cuenta con 3 botones los cuales con ellos se puede calibrar el zoom, focos y iris.



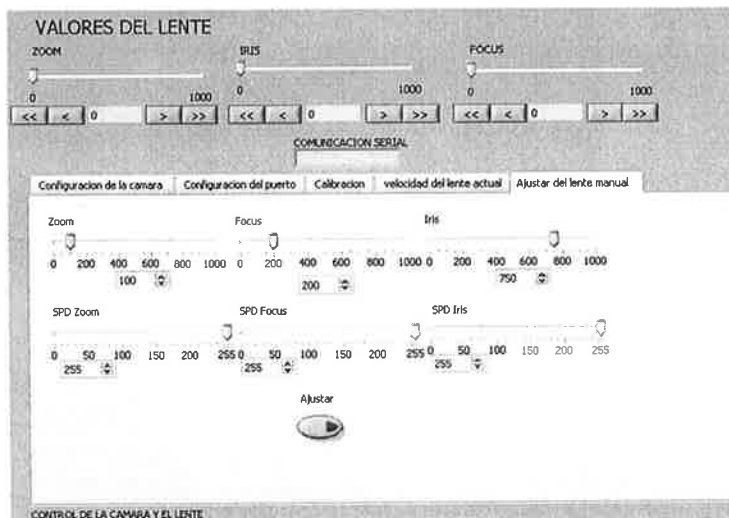
**Fig 13.-** Pestaña de Calibración

En esta pestaña se puede visualizar a que velocidad se están moviendo los motores del zoom, focus y iris.



**Fig 14.- Pestaña de Velocidad del lente**

En esta pestaña podemos ajustar el zoom, focus y iris a una posición determinada y también a que velocidad lo queremos hacer.



**Fig 15.- Ajuste del lente manual**

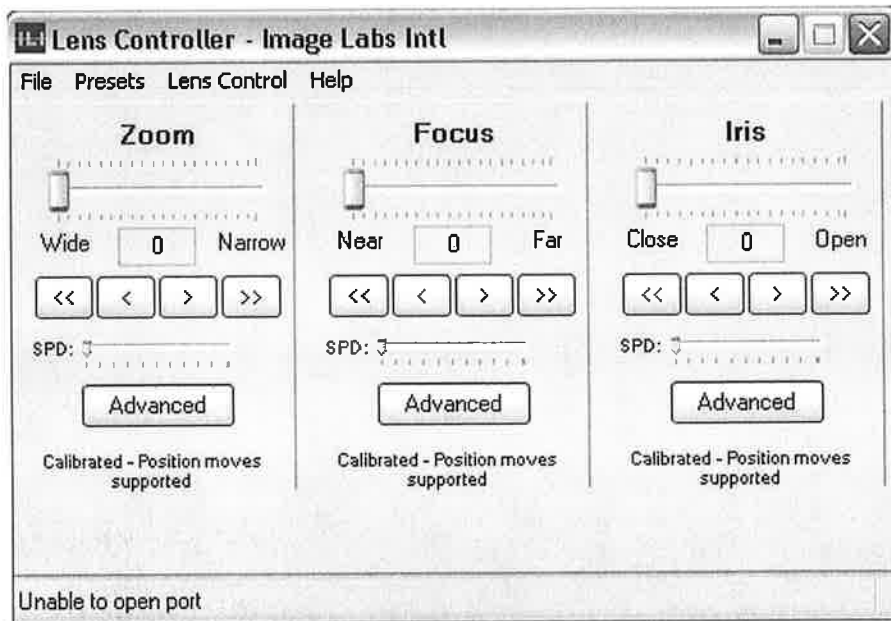
Con este programa es posible manipular tres aspectos importantes del lente: Nosotros podemos manipular 3 cosas importantes del lente lo que es:

**Zoom:** Capacidad de aumentar o reducir el tamaño de la figura visualizada en la pantalla.

**Focus:** función que hace que la imagen de un objeto producida en el foco de una lente se visualice con claridad sobre un plano determinado

**Iris:** Dispositivo colocado en el interior del lente de la cámara que regula la cantidad de luz que entra a la misma.

Para tener un control preciso de estas funciones se creo un programa que esta basado en el diseño que ya estaba realizado por el fabricante el cual se ve de esta manera:



**Fig 16.- Programa Lens Controller**

Basándonos en este programa se analizó las partes que lo componía y su funcionalidad, de esta forma se desarrolló el programa para el control del lente el cual se realizó de la siguiente forma:

## DIAGRAMA DE FLUJO Del lente

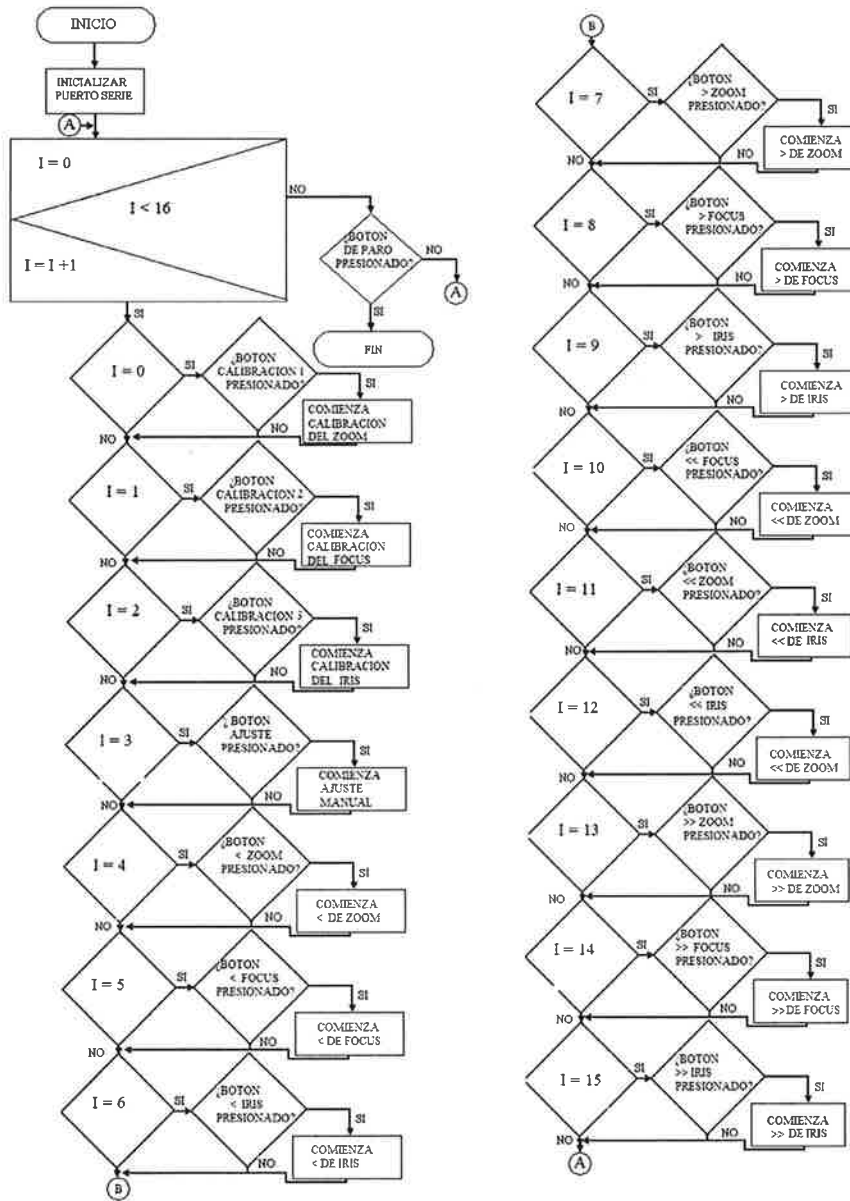


Diagrama 2: diagrama de flujo del lente

Como podemos ver en el diagrama primero se realiza la comunicación serial y después entra en un ciclo con 16 intrusiones las que se tiene relacionadas cada una con su botón. Al momento en que se activa el botón se realiza la operación indicada. Enseguida se explicara con detalle como se realizaron cada una de las instrucciones.

Primeramente se define el puerto serial, el cual se configura de esta manera:

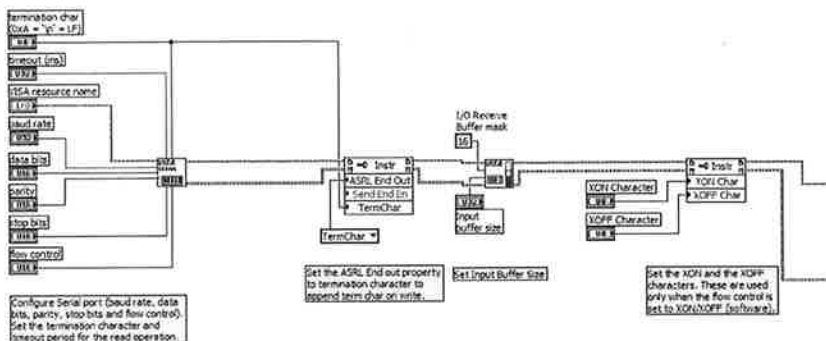


Diagrama 3 conexión con el puerto serial

Esta configuración tiene 5 características que están definidas por el fabricante del controlador y son muy importantes ya que sin ellas no se logra la comunicación:

Baud Rate	19200
Data Bits	8
Parity	None
Stop Bit	1
Flow Control	None

Tiene que estar configurado de igual manera en la configuración del puerto com:

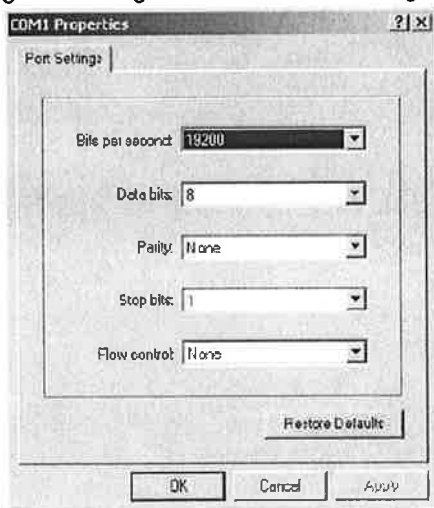


Fig 17.- ventana de propiedades del puerto serial

Después de establecida la comunicación el programa entra a un ciclo while que se va a estar repitiendo todo el tiempo que se este ejecutando el programa, para el controlador del lente se utilizan diferentes comandos de entradas y salidas.

Los datos de entrada son estos:

ENTRADAS	Descripción	Ejemplo	Función
TIM*#####→TIM05000	establece el tiempo de del motor * de reacción en #####	TIM05000	Hace que el tiempo de reacción de el zoom por que es 0 de las instrucciones sea en 5000 ms
SPD*#####→SPD00255	establece la velocidad de un motor * a #####	SPD00255	Hace que la velocidad del motor del zoom sea 255
DIR*#→DIR00	llama a la dirección del motor * en memoria	DIR00	Guarda las posiciones del zoom en el espacio de memoria 0
MVP*#####→MVP0045	mueve el motor * a una posición #####	MVP0045	mueve el zoom en una posición de 045
POS*→POS0	pedir la posición del motor *	POS1	pedir la posición actual del focus
SPD*→SPD0	pedir la velocidad del motor *	SPD2	pedir la velocidad actual del iris

**Tabla 2.- Tabla de comandos de entrada**

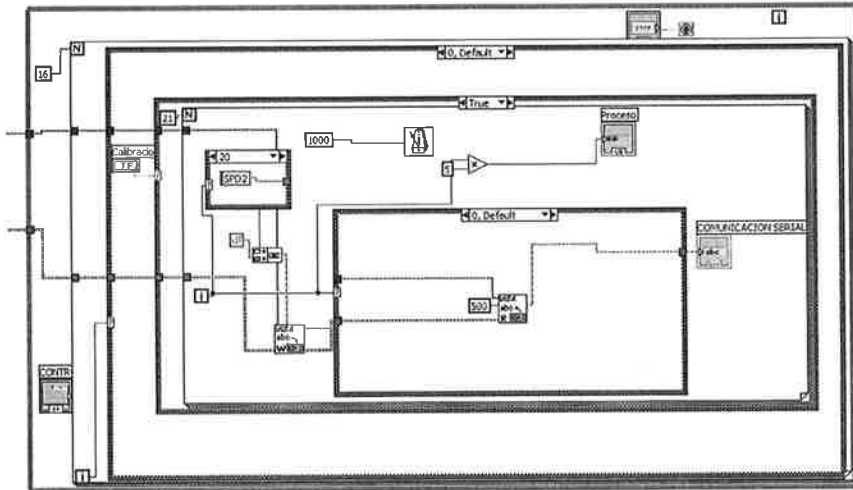
SALIDAS	Descripción	Ejemplo	Que información manda
T*@#####→T0@05000 *S	Regresa el tiempo establecido al ser T	T0@05000 *S	esta regresando que el tiempo de respuesta del zoom es de 5000
S*@#####→S0@00255 *S	Regresa la velocidad establecida al ser S	S0@00255 *S	Esta regresando el valor de la velocidad actual del zoom que es 0255
P*@#####→P0@00603 *S	Regresa la posición que esta actualmente el motor	P0@00603 *S	Esta regresando el valor de la posición actual del zoom que es 0603
P*@#####→P0@00978 *F	Regresa la posición a la cual pudo llegar el motor exactamente como no pudo llegar a la posición deseada manda un *F	P0@00978 *F	Para este caso se le pudo mandar un MVP01000 y por seguridad regresa el valor de la posición actual del zoom que es de 0978 y no pudo llegar a 1000 por seguridad del motor

**Tabla 3.- Tabla de comandos de salida**



Al saber esto entonces lo siguiente es general el ciclo de envío de los datos.

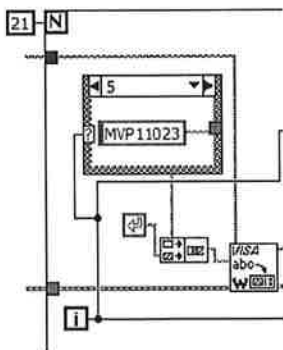
Para ello esta entramos a otro ciclo for de 16 casos que son nuestras instrucciones. las cuales a su vez entran a un case que se activara con los botones que le corresponde a cada una de las instrucciones. El diagrama de bloques queda así:



**Diagrama 4: Diagrama de bloques del programa**

A continuación veremos cada una de las 16 instrucciones las cuales son del 1 al 3 calibración los 3 motores, el 4 ajustar manualmente los 3 motores, del 5 al 7 un pequeño adelanto al zoom, focus y iris, del 8 al 10 un pequeño retroceso al zoom, focus y iris, del 11 al 13 un largo adelanto al zoom, focus y iris y del 14 al 16 un largo retroceso al zoom, focus y iris.

### 7.1.- calibración del Zoom



Si se activa el boton de calibración del zoom entra a un ciclo for en este caso de 21 interacciones que son los pasos para llegar a calibrar el zoom, a continuación creamos un case que esta relacionado con el ciclo y manda 21 ordenes por el puesto serial al servo motor para llegar a realizar la calibración como se puede ver en la imagen:

Si es el caso 1 mandamos : TIM0 → establece el tiempo de reacción

Si es el caso 2 mandamos : TIM0 → establece el tiempo de reacción

Si es el caso 3 mandamos : SPD0 → establece la velocidad en 0 del zoom

**Fig 18.- ciclo for y escritura**

Si es el caso 4 mandamos : SPD0255→ establece la velocidad en 255 del zoom

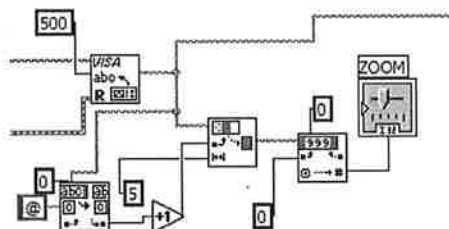
Si es el caso 5 mandamos : DIR00 → llama a la dirección 0

Si es el caso 6 mandamos : MVP01023 → mueve el zoom hasta la posición 1023

Si es el caso 7 mandamos : TIM01000→ establece el tiempo de reacción con 1000

- Si es el caso 8 mandamos : MVP01023→ mueve el zoom hasta la posición 1023
- Si es el caso 9 mandamos : TIM05000→ establece el tiempo de reacción 5000
- Si es el caso 10 mandamos : MVP00 → mueve el zoom hasta la posición 0
- Si es el caso 11 mandamos : TIM01000 → establece el tiempo de reacción con 1000
- Si es el caso 12 mandamos : MVP00 → mueve el zoom hasta la posición 0
- Si es el caso 13 mandamos : TIM05000→ establece el tiempo de reacción 5000
- Si es el caso 14 mandamos : DIR00→ llama a la dirección 0
- Si es el caso 15 mandamos : SPD0255→ establece la velocidad en 255 del zoom
- Si es el caso 16 mandamos : POS0→ pide la posición actual del zoom
- Si es el caso 17 mandamos : SPD0→ pide la velocidad del zoom
- Si es el caso 18 mandamos : POS1→ pide la posición actual del focus
- Si es el caso 19 mandamos : SPD1→ pide la velocidad del focus
- Si es el caso 20 mandamos : POS2→ pide la posición actual del iris
- Si es el caso 21 mandamos : SPD2→ pide la velocidad del iris

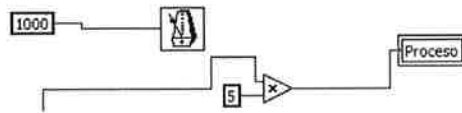
Después de estar enviándolos cada uno de ellos nos centramos particular lo que nos reenvía el servomotor como respuestas de nuestras instrucciones en las primeras instrucciones no regresa nada el controlador. En la instrucción 16,17,18,19,20 y 21 en ellas nos manda una respuesta que son las posiciones reales y velocidades reales del lente en la que se encuentra. Claro que los datos hay que darle su debida interpretación para ello se creó un proceso en el cual se extraen los datos de esta manera :



**Fig 19.- Extracción de los datos**

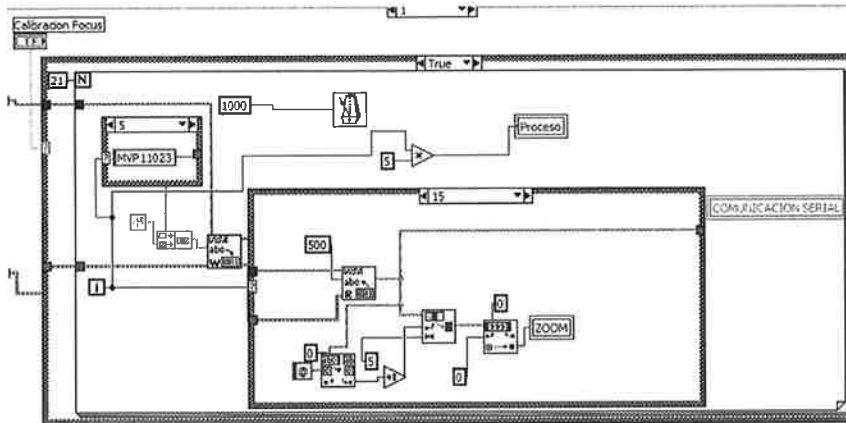
Por que el controlador del lente nos manda datos de esta manera P0@00044\*S entonces primero de esa cadena buscamos la @ del valor que tiene esa posición la incrementamos y dividimos la cadena en un nueva sub cadena con solo 5 valores que son los número que queremos en este caso el 00044 después lo transformamos a un valor numérico y lo colocamos en la variable que le corresponde para poder verlo en la pantalla. Esto se repite en los demás casos. Y también se le agregó una barra de procesos para ver en que proceso va y cuanto le falta para terminar.

Cabe destacar que se configuro a un tiempo determinado para calibrar.



**Fig 20.- llenado de la barra de proceso**

## 7.2.-Calibracion del Focus



**Diagrama 5: diagrama de bloques de la calibración del focus**

Esta secuencia es muy semejante a la calibración del zoom lo que hacemos es otra vez usar un ciclo for con 21 pasos para calibrar el focus de la siguiente manera :

- Si es 1 mandamos : TIM1 → establece el tiempo de reacción
- Si es 2 mandamos : TIM1 → establece el tiempo de reacción
- Si es 3 mandamos : SPD1 → establece la velocidad en 0 del focus
- Si es 4 mandamos : SPD1255→ establece la velocidad en 255 del focus
- Si es 5 mandamos : DIR10 → llama a la dirección 1
- Si es 6 mandamos : MVP11023 → mueve el focus hasta la posición 1023
- Si es 7 mandamos : TIM11000→ establece el tiempo de reacción con 1000
- Si es 8 mandamos : MVP11023→ mueve el focus hasta la posición 1023
- Si es 9 mandamos : TIM15000→ establece el tiempo de reacción 5000
- Si es 10 mandamos : MVP10 → mueve el focus hasta la posición 0
- Si es 11 mandamos : TIM11000 → establece el tiempo de reacción con 1000
- Si es 12 mandamos : MVP10 → mueve el focus hasta la posición 0
- Si es 13 mandamos : TIM15000→ establece el tiempo de reacción 5000
- Si es 14 mandamos : DIR10→ llama a la dirección 1

Si es 15 mandamos : SPD1255→ establece la velocidad en 255 del focus

Si es 16 mandamos : POS0→ pide la posición actual del zoom

Si es 17 mandamos : SPD0→ pide la velocidad del zoom

Si es 18 mandamos : POS1→ pide la posición actual del focus

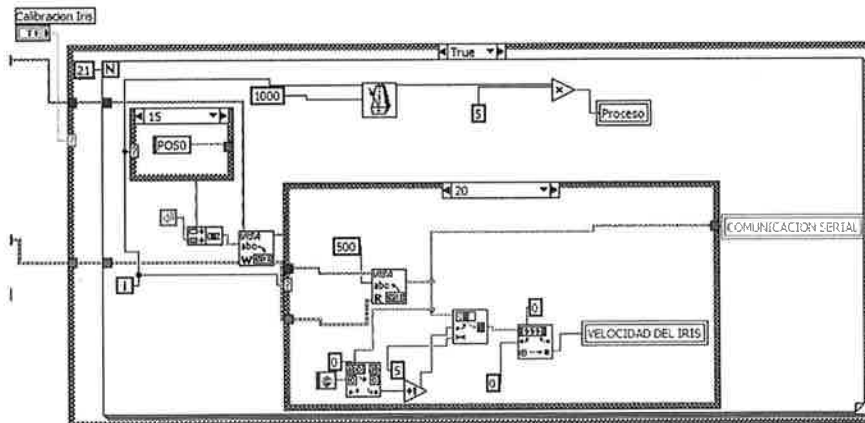
Si es 19 mandamos : SPD1→ pide la velocidad del focus

Si es 20 mandamos : POS2→ pide la posición actual del iris

Si es 21 mandamos : SPD2→ pide la velocidad del iris

Como podemos ver la única diferencia es en los primeros pasos en lugar de mandar un 0 se cambio por un 1 pero por lo demás es igual. También de la misma manera recibimos y enviamos los datos a una variable local que le corresponde.

### 7.3.-Calibración del Iris



**Diagrama 6: diagrama de bloques de la calibración del iris**

Esta secuencia es muy semejante a la calibración del zoom lo que hacemos es otra vez empezar un ciclo for de 21 pasos para calibrar el iris de la siguiente manera :

Si es 1 mandamos : TIM2 → establece el tiempo de reacción

Si es 2 mandamos : TIM2 → establece el tiempo de reacción

Si es 3 mandamos : SPD2 → establece la velocidad en 0 del iris

Si es 4 mandamos : SPD2255→ establece la velocidad en 255 del iris

Si es 5 mandamos : DIR20 → llama a la dirección 2

Si es 6 mandamos : MVP21023 → mueve el iris hasta la posición 1023

Si es 7 mandamos : TIM21000→ establece el tiempo de reacción con 1000

- Si es 8 mandamos : MVP21023→ mueve el iris hasta la posición 1023
- Si es 9 mandamos : TIM25000→ establece el tiempo de reacción 5000
- Si es 10 mandamos : MVP20 → mueve el iris hasta la posición 0
- Si es 11 mandamos : TIM21000 → establece el tiempo de reacción con 1000
- Si es 12 mandamos : MVP20 → mueve el iris hasta la posición 0
- Si es 13 mandamos : TIM25000→ establece el tiempo de reacción 5000
- Si es 14 mandamos : DIR20→ llama a la dirección 2
- Si es 15 mandamos : SPD2255→ establece la velocidad en 255 del iris
- Si es 16 mandamos : POS0→ pide la posición actual del zoom
- Si es 17 mandamos : SPD0→ pide la velocidad del zoom
- Si es 18 mandamos : POS1→ pide la posición actual del focus
- Si es 19 mandamos : SPD1→ pide la velocidad del focus
- Si es 20 mandamos : POS2→ pide la posición actual del iris
- Si es 21 mandamos : SPD2→ pide la velocidad del iris

Como podemos ver la única diferencia es en los primeros pasos en ves de mandar un 0 se cambio por un 2 pero por lo demás es igual. También de la misma manera recibimos y enviamos los datos a una variable local que le corresponde.

#### 7.4.- Ajuste Manual

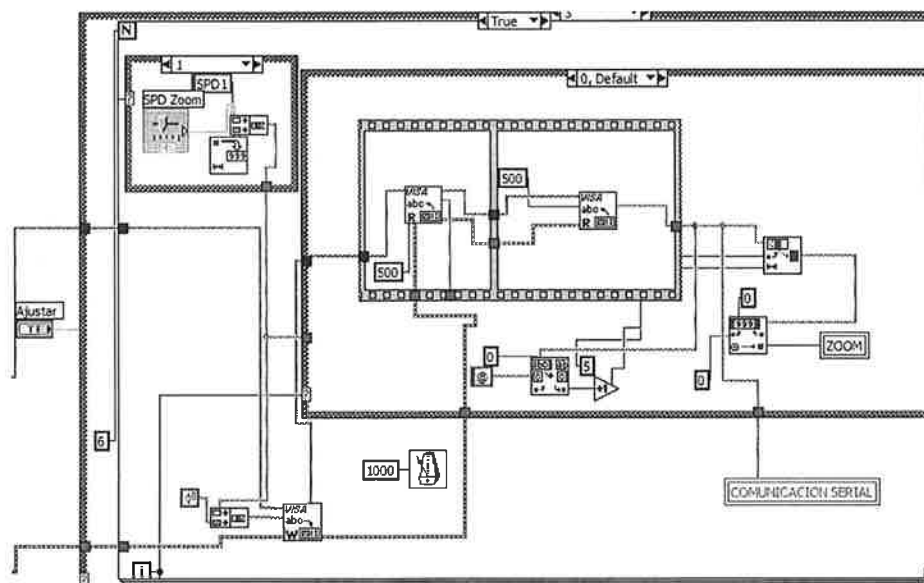
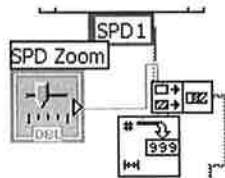


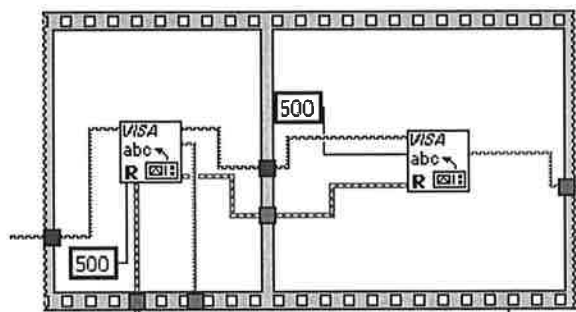
Diagrama 7: diagrama de bloques del ajuste manual

En esta sección tenemos otro ciclo for con 6 instrucciones que mandamos al controlador del lente las cuales son zoom , velocidad del zoom, focus, velocidad focus, iris y velocidad de iris, para ello primero hacemos un procedimiento para conseguir los valores que nos dio el usuario usando los controladores en el anel de control del programa y transformamos estos valores a cadenas :



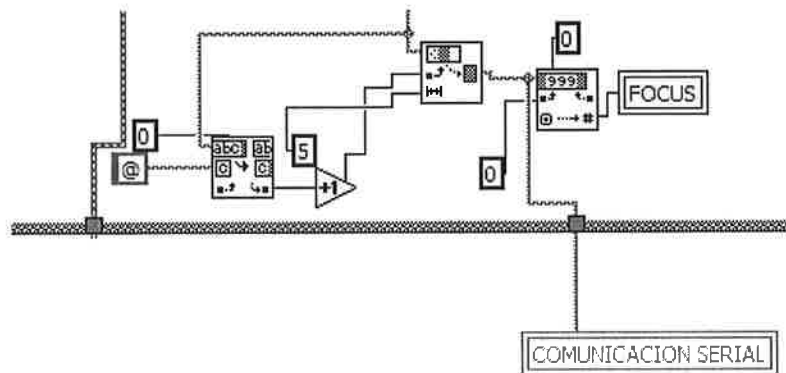
**Fig 21.- convertir los valores a cadenas**

Primero convertimos el número en una cadena y después lo concatenamos con un “MVP” o “SPD” dependiendo de la instrucción “MVP” para ajustar el motor o “SPD” para ajustar la velocidad y el número del motor que queremos controlar para zoom 0 , focus 1 y iris 2, finalmente esa instrucciones son mandadas al puerto serial.



**Fig 22.- leer dos datos**

Después recibimos 2 mensajes del puerto, el primero valor es el enviado el cual puede que sea el valido de la posición actual o no debido a la limitancia del lente, así que no lo utilizamos y la segunda si ya es la posición exacta del lente entonces la mandamos al siguiente procedimiento.



**Fig 23.-Adquisición de la posición actual**

Este procedimiento es el mismo ya visto en la calibración para separar el valor del mensaje recibido buscando el @ y a partir de este carácter creamos una sub cadena y transformando esa cadena en un número, para después mandarlo a su variable correspondiente que se indicara en la pantalla.

Para las velocidades es diferente solamente se toma el valor que se le coloco en la pantalla y se vuelve enviar directamente a la variable ya que no tiene que pasar por una validaciones ese valor  
Ejemplo:

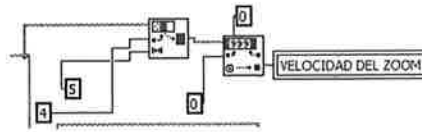


Fig 24.-adquisición de la velocidad

### 7.5.-retroceso del zoom <

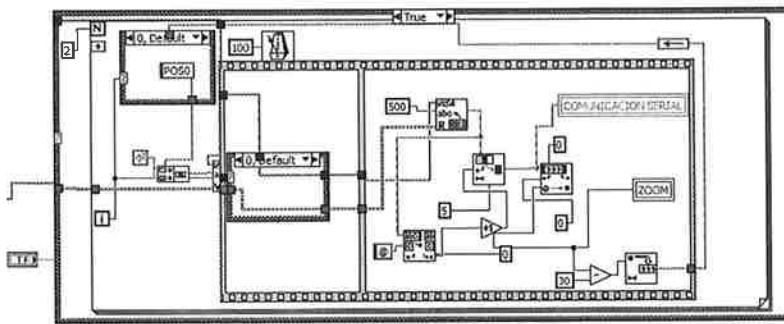


Diagrama 8 Diagrama de bloques de Pedir la posición

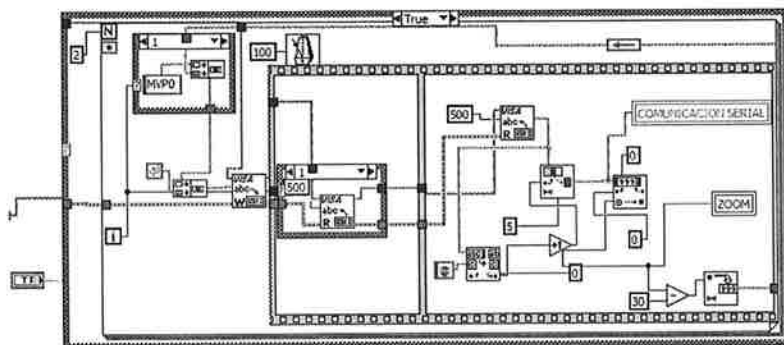

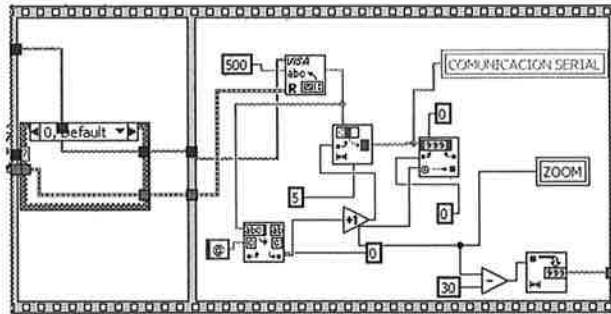


Diagrama 9 Diagrama de bloques de Decrementar la posición

Como en los casos anteriores si el botón retroceso zoom  es presionado entra a un ciclo for en el cual solo se repetirá 2 veces, en el cual la primera vez se le va a mandar un "POS" con el valor del motor 0 que es zoom para obtener el valor en que se encuentra el motor actualmente, después le restamos a ese valor 30 (este valor se pone ya que tiene un rango los motores de aproximadamente de 0 a 1000 entonces para que sea visible se selecciona el valor de 30) y después en la repetición se le manda una "MVP" con un 0 ya que es el zoom y el nuevo valor creado al para que se mueva a esa posición actual.

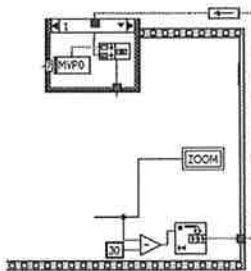
Ejemplo : primero preguntamos POS0 y nos llega el valor P0@00300 \*S entonces sabemos que esta en la posición 300 le restamos 30 tenemos un valor de 270 entonces ese valor lo mandamos al puerto serial con un MVP0270 para que se mueva.

Entonces le mandamos el POS0 otra vez para asegurarnos de que sea el valor actual y lo metemos a un sequence que queda de esta manera:



**Fig 25.-secuencia de retroceso**

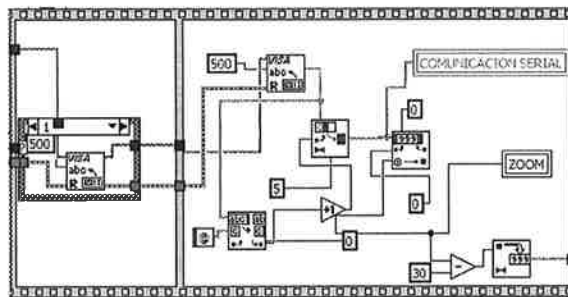
en la secuencia leemos el dato, lo transformamos a un número y lo metemos en la variable correspondiente zoom.



**Fig 26.-reenvió**

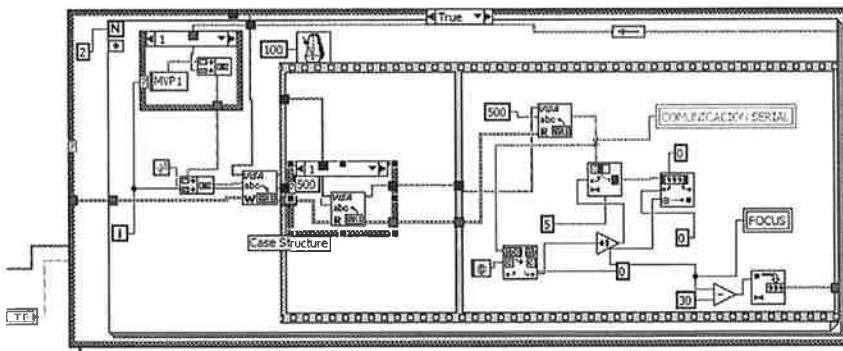
Después se le resta la variable y la enviamos otra vez para el siguiente paso adonde le decimos al motor que se mueva a la posición restada .

Después entra otra vez al Sequence primero le el valor enviado y después leemos el otro valor el cual es la posición actual esa la mandamos a la variable correspondiente.



**Fig 27.- secuencia retroceso 2**

7.6.-retroceso del Focus <



**Diagrama 10 Diagrama de bloques del retroceso del focus**

esta secuencia es idéntica a la pasada el único cambio que se le realiza es en vez de manejar el motor 0 manejamos el motor 1 y mandamos los valores a la variable FOCUS



7.7.-retroceso del Iris <

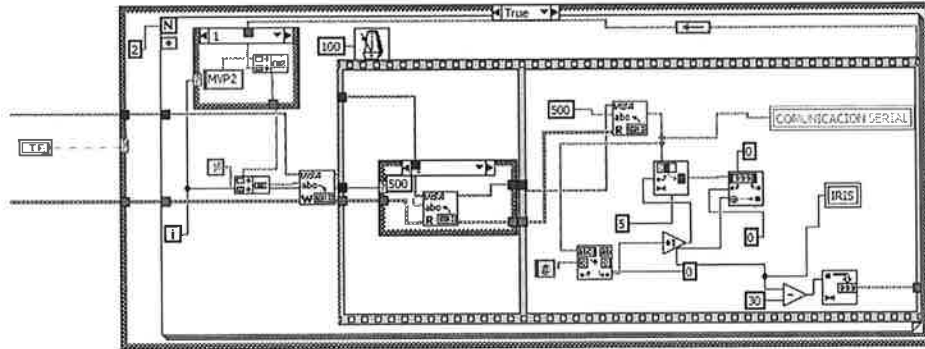


Diagrama 11 Diagrama de bloques del retroceso del Iris

Esta secuencia es idéntica a la pasada el único cambio que en lugar de manejar el motor 1 manejamos el motor 2 y mandamos los valores a la variable IRIS

7.8.-adelanto del zoom >

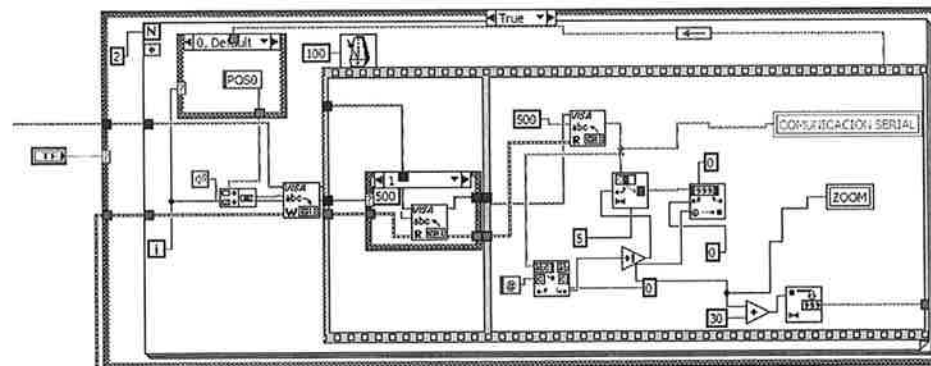


Diagrama 12 Diagrama de bloques del adelanto del zoom

Esta secuencia es totalmente idéntica a la 5 la única diferencia que se presenta es que en vez de restarle 30 aquí se le suma 30 (este valor se pone ya que tiene un rango los motores de aproximadamente de 0 a 1000 entonces para que sea visible se selecciona el valor de 30) a la variable.

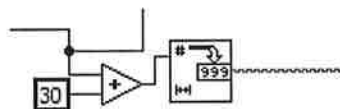


Fig 28.-convertirlo a cadena

### 7.9.-adelanto del Focus >

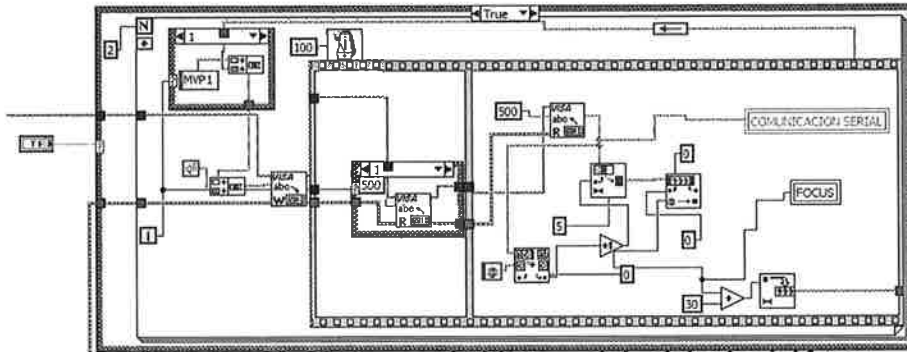


Diagrama 13 Diagrama de bloques del adelanto del focus

Esta secuencia es totalmente idéntica a la 6 la única diferencia que se presenta es que en vez de restarle 30 aquí se le suma 30 a la variable.

### 7.10.-adelanto del Iris >

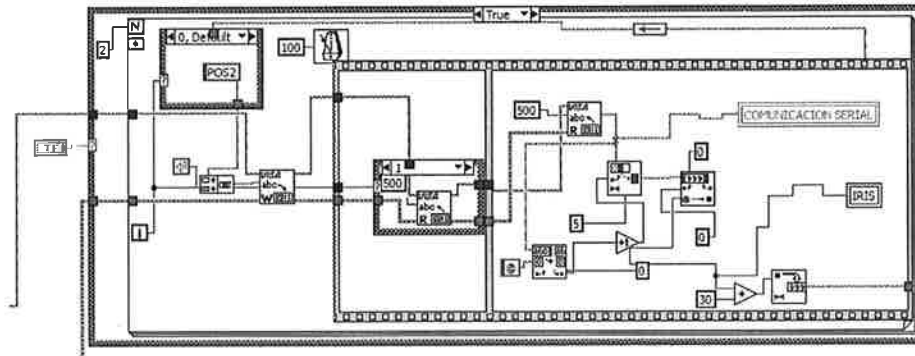


Diagrama 14 Diagrama de bloques del adelanto del Iris

Esta secuencia es totalmente idéntica a la 7 la única diferencia que se presenta es que en lugar de restarle 30 aquí se le suma 30 a la variable.

### 7.11.-retroceso largo del zoom <<

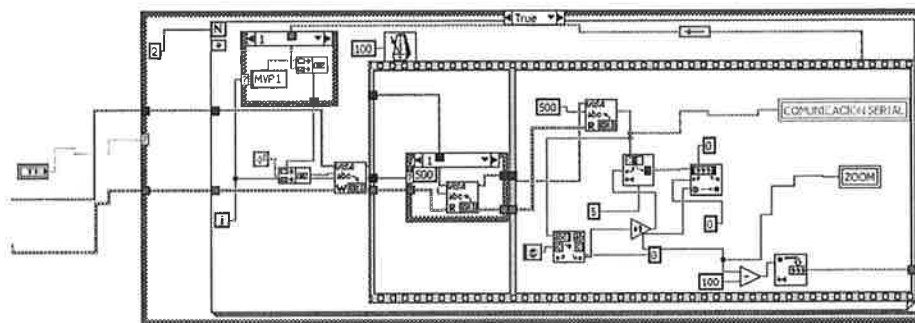


Diagrama 15 Diagrama de bloques del retroceso largo del zoom

Esta secuencia es totalmente idéntica a la 5 la única diferencia que se presenta es que en vez de restarle 100 este valor se elige para que sea un salto más grande y uno se pueda avanzar más rápido que a diferencia de 30.

Esta secuencia es totalmente idéntica a la 7 la única diferencia que se presenta es que en vez de suma 100 (este valor se pone ya que tiene un rango los motores de aproximadamente de 0 a 1000).

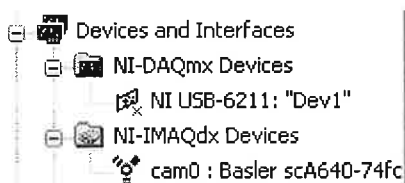
Con ello se finaliza la parte del programa correspondiente al control del lente Rainbow H6 X 8M II y el controlador VCL3.

## 8.-Desarrollo de la programación de la calibración de la cámara y procesamiento de imagen.

Para esta parte se analizaron las características de la cámara y de sus lentes para tener información y exacta del equipo que tenemos, revisando la hoja de datos de cada uno de estos sistemas.

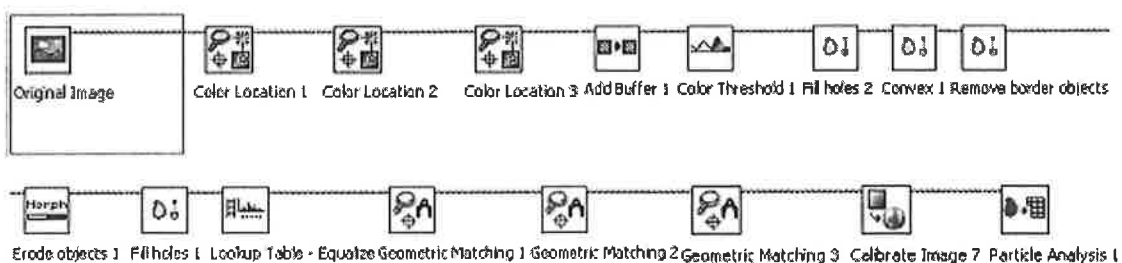
Después se configuró la cámara en el Measurement & Automation Explorer, adonde ha sido reconocida la cámara y podemos localizarla de esta manera :

El Measurement & Automation Explorer es una aplicación de LabVIEW que permite la detección y configuración del hardware para usarlo de una manera más fácil.



**Fig 30.- detección de la cámara**

Posteriormente usando NI Vision Assistant creamos este script (código del programa) para la medición de las figuras establecidas: NI Vision Assistant es una aplicación de LabVIEW para facilitar las aplicaciones con Vision.

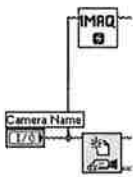


**Fig 31.- script creado en vision assistant**

El cual consiste en estas actividades:

Abre una imagen (VI Original Image)

Identificar color Azul(VI Color Location 1)  
 Identificar color Rojo(VI Color Location 2)  
 Identificar color Verde(VI Color Location 3)  
 Guardar en el buffer la imagen original(VI Add Buffer)  
 Aplicar un umbral a los 3 colores para distinguirlos de los demás(VI Color Threshold 1)  
 rellena los huecos (VI Fill holes 1 )  
 Eliminamos todas las partículas que tocan el borde de la imagen(VI Convex 1)  
 Removemos los bordes de los objetos (VI Remove border objects)  
 Aplicamos un filtro de erode (VI Erode objects 1)  
 Otra vez rellenamos los huecos(VI Fill holes 2)  
 Calculamos su histograma (VI Lookup Table)  
 Calibramos la imagen (VI Calibrate Image 7)  
 Buscamos y calculamos círculos(VI Geometric Matching 1)  
 Buscamos y calculamos cuadrados(VI Geometric Matching 2)  
 Buscamos y calculamos triángulos(VI Geometric Matching 3)  
 Desplegarlos los datos (VI Particle Análisis 1)

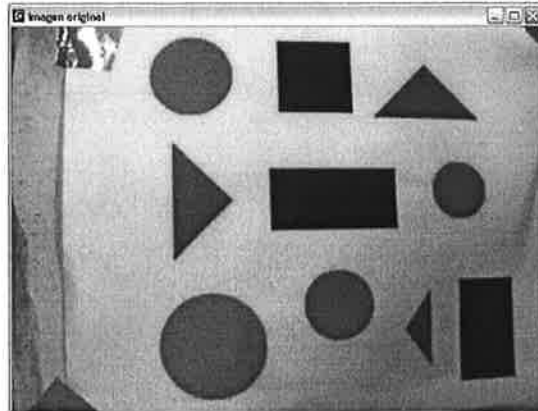
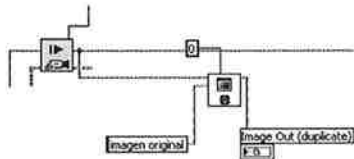


El siguiente paso fue pasarlo a nuestro programa principal.

En este programa generamos una comunicación con la cámara fireware y lo pasaremos un IMAQ para adquirir la imagen se ve de la siguiente manera:

**Fig 32.conexión con la cámara**

En seguida entra en un ciclo while infinito que va a tomar video todo el tiempo en el que estemos corriendo el programa, en esta figura 34 se despliega la imagen original en la primera ventana:



**Fig 33.- Despliegado de la imagen**

**Fig 34.- imagen original**

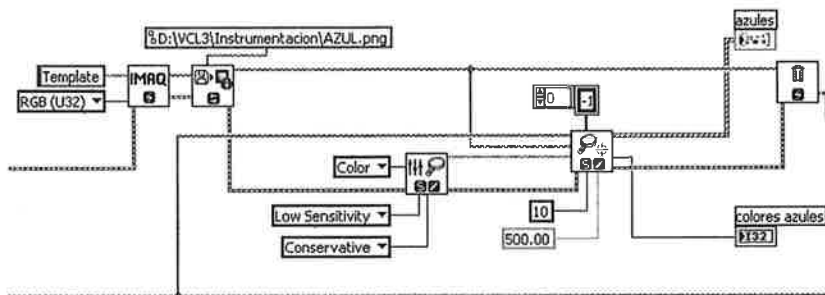
Al adquirir la imagen es posible ajustar el brillo, gamma y contraste de la imagen en la pantalla de configuración de la cámara hay se pusieron 3 perillas para poder ajustarlo.




**Fig 35.ajuste brillo**

Después aremos el primer paso que es la identificar de los colores en la imagen y desplegarlo.

1.- Identificar color Azul



**Diagrama 21 Diagrama de bloques de la identificación del color azul**

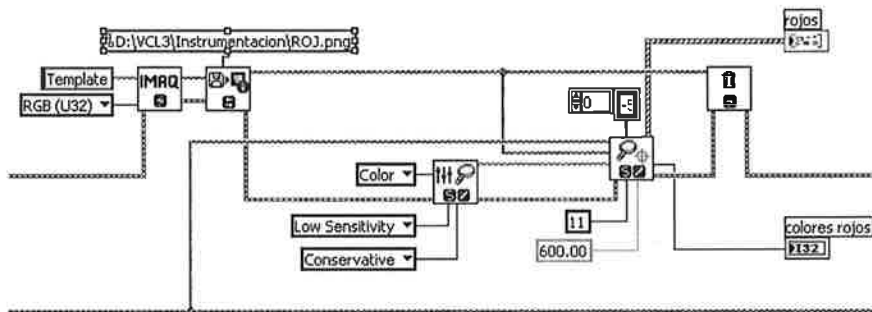
La cual consistes que creamos otra IMAQ con la imagen ya de finida del color azul la cual es esta:  un color azul con esta tonalidad.

Ya después aemos una búsqueda del color en la imagen adquirida por la cámara, para ello se utiliza el algoritmo IMAQ Match color Pattern de NI vision desplegamos el resultado de la búsqueda, cuantas veces se encontró exactamente el color y en que posición se encontro.


Y para finalizar desechamos la imagen del color para que no gastemos memoria.

Después repetimos otra vez el mismo procedimiento para el color rojo:

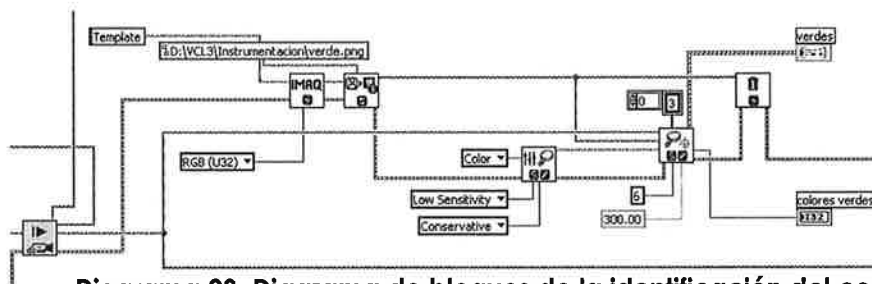
2.- Identificar color Rojo




**Diagrama 22 Diagrama de bloques de la identificación del color Rojo**

Definimos el color rojo como una imagen la cual es esta: 

De la misma manera se repite para procedimiento del color verde:



**Diagrama 23 Diagrama de bloques de la identificación del color verde**

Definimos el color verde con la siguiente tonalidad: 

Nota: se define el color de esta manera por que como existe una serie colores diferentes tonalidades queremos que identifique ese color exactamente con esa tonalidad y no se confunda con otros azules o rojos o verdes.

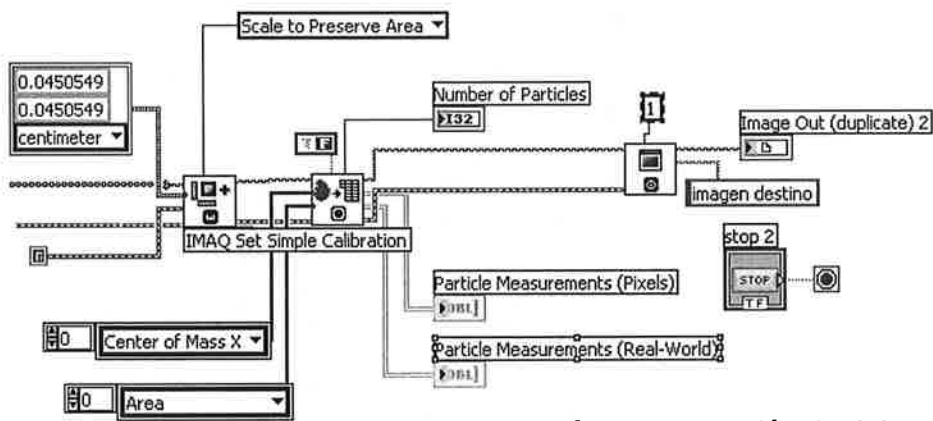
El siguiente paso Aplicar un umbral a los 3 colores para distinguirlos de los demás, en este caso se utilizo el rango de los mismo 3 colores ya definidos para distinguirlos genera de una imagen de colores una imagen binaria donde los colores tienen un valor de 255 o sea blanco.

Después aplicamos rellena los huecos ,eliminamos todas las partículas que tocan el borde de la imagen, removemos los bordes de los objetos, aplicamos un filtro de erosión ,otra vez rellenamos

En la tabla 4 de la sección de resultados se muestran las medidas obtenidas y su diferencia con las medidas de la figura real.

El programa de calibración y de determinación del número de figuras se muestra en la figura siguiente.

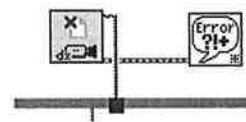
Donde se puede ver el VI de Simple calibration que es el encargado de la calibración de la imagen el cual usa un algoritmo generado en Vision el cual consiste en una proyectando un conjunto de puntos de posición conocida. En este caso una hoja solo se le da el tamaño de la hoja y el calibra con respecto a la hoja.



**Diagrama 25 Diagrama de bloques de la calibración y presentación de datos**

También vemos como se despliegan los datos en la pantalla y la imagen en otra ventana llamada imagen destino.

Por último hacemos que se finalice la sesión de la cámara usando el VI:



**Fig 39. finalizar la sesión de la cámara**

Se puede ver que el algoritmo no falla en determinar la forma de la figura y que la diferencia de medidas es pequeña en el orden de cm.

## **10.-Conclusiones y trabajo futuro**

El proyecto fue muy interesante ya que probamos como la implementación de teorías básicas nos puede llevar al desarrollo de tecnologías avanzadas. Se lograron los objetivos los cuales eran caracterizar la cámara BASLER Scout y sus dos tipos de lentes: el primero un lente Fujinon HF9HA -1B 1:1.4/9mm y el segundo un Rainbow H6 X 8M, II. Se creó un programa para el controlador VCL3, también se logró diseñar un programa que obtenga los parámetros geométricos y la localización de figuras definidas para la medición de dichas figuras. El cual tiene un buen acercamiento a la realidad, pero tiene que ser mejorado si se quiere usar en un robot.

También puede medir cualquier otro tipo de figura que se le defina.

Finalmente este trabajo permitió conocer uno de los aspectos más importantes del desarrollo secuencial para el mejoramiento y perfección de tecnologías ya existentes.

Como trabajo futuro se planea realizar pruebas con la cámara en ambiente submarino, para ello se planea poner dentro de una pecera llena de agua y ver las características que nos muestra en este ambiente. Como técnica de medición submarina se plantea colocar unos Led's con ángulos previamente determinados, para calcular de esta manera el tamaño aproximado de figuras y objetos. Con esta técnica la cámara podrá estar en movimiento y esta es la idea principal, puesto que la cámara estará montada en el robot submarino.

## **11.-Bibliografía**

- [1] Labview Intermediate 1 y 2 , editorial Nacional instruments.(2005)
- [2] Image Acquisition and processing with Labview . Christopher G.Relf. ed.VRC PRESS(2006)
- [3] Digital Image Processing, segunda edición, Rafael C.Gonzalez y Richard E. Woods. (2001)
- [4] Fundamentals of image processing , Ian T. Young, Jan J.Gerbrands y Lucas J. van Vliet (1998)
- [5] comparativa entre cámaras y métodos de calibración en García y Cuartero (2002).
- [6] software de calibración, encontramos un estudio de varias aplicaciones en Wiedemann et al. (2001).
- [7] <http://www.apple.com/la/firewire/>
- [8] <http://es.wikipedia.org/wiki/LabVIEW>
- [9] <http://zone.ni.com/devzone/cda/tut/p/id/6908>
- [10] [http://es.wikipedia.org/wiki/Puerto\\_serie](http://es.wikipedia.org/wiki/Puerto_serie)
- [11] <http://www.textoscientificos.com/fotografia/camara>