

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“Diseño y desarrollo de un prototipo para inspeccionar faltantes de
botellas en cajas usando el software Labview 8.2”**

TESIS DE GRADO

Previo a la obtención del Título de:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN
ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL.**

Presentada por:

Christian Alfonso Espinoza Villao

Jorge Enrique Mazzini Moscoso

GUAYAQUIL - ECUADOR

Año: 2009

AGRADECIMIENTO

Al Ing. Holger Cevallos, profesor de la FIEC y director de nuestra tesis, por su colaboración y ayuda incondicional en la realización del mismo. A nuestras familias, por su apoyo integral.

DEDICATORIA

Dedico principalmente a Dios y a mis padres que en el amor me enseñan a seguir adelante en mis sueños.

A mis amigos, profesores y compañeros quienes incondicionalmente siempre me apoyaron hasta las últimas circunstancias.

Jorge Enrique Mazzini Moscoso

GRACIAS

DEDICATORIA

Es mi anhelo dedicar este trabajo a Dios que es mi fortaleza diaria y aquel que me inspira a ser mejor cada día, a mis padres por apoyarme y ser mis guías a lo largo de mi vida.

Christian Alfonso Espinoza Villao

GRACIAS

TRIBUNAL DE GRADUACIÓN

Ing. Jorge Aragundi
SUBDECANO FIEC

Ing. Holger Cevallos
DIRECTOR DE TESIS

Ing. Alberto Larco
VOCAL PRINCIPAL

Ing. Efrén Herrera
VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de ésta Tesis de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

Jorge Enrique Mazzini Moscoso

Christian Alfonso Espinoza Viilao

RESUMEN

La tesis que desarrollamos analiza y propone una solución al problema de faltante de botellas en una jaba causada por el proceso de encajonamiento de éstas, en una planta embotelladora, previa al almacenamiento y distribución de las jabas.

El “Inspector de cajas” es un sistema cuya función es verificar que las jabas se transporten completas para obtener una mejor calidad en el proceso. Esta etapa la realizan los operadores, que tradicionalmente verifican y corrigen las jabas incompletas en el proceso de post encajonamiento; el “Inspector de cajas” hará ésta verificación de forma automatizada.

El inspector de cajas captura imágenes de la jaba con una cámara verificando si esta se encuentra completa. Luego, las imágenes tomadas son comparadas dentro de un software con imágenes de referencia, y después de este proceso el software decide si la jaba se acepta o se rechaza, de acuerdo al número de botellas encontradas. Una jaba aceptada debe estar completamente llena, mientras que una jaba rechazada debe tener un faltante de al menos una botella.

Las herramientas utilizadas para el desarrollo del “Inspector de cajas” fueron Vision Assistant, que es el software encargado de procesar las imágenes; y

Labview que es el software que se utilizó para programar gráficamente el algoritmo de solución; ambas pertenecientes al módulo de desarrollo de Vision NI de Labview.

INDICE GENERAL

RESUMEN.....	VI
INDICE GENERAL.....	VIII
INDICE.....	VIII
ABREVIATURAS.....	XI
INDICE DE FIGURAS.....	XII
INDICE DE TABLAS.....	XVI
INTRODUCCIÓN.....	XVII

INDICE

CAPÍTULO 1

1. GENERALIDADES.....	3
1.1. ¿Qué es un inspector de cajas?.....	3
1.2. Descripción del funcionamiento del inspector de cajas.....	8
1.3. Aplicaciones del inspector de cajas.....	11
1.4. Alcance del proyecto.....	12

CAPÍTULO 2

2. DISEÑO Y ARQUITECTURA DEL SISTEMA INSPECTOR DE CAJAS.....	15
2.1. Consideraciones en el diseño.....	15
2.2. Tecnología empleada.....	17
2.2.1. Tipos de software.....	17
2.2.2. Tipos de cámara.....	22
2.2.3. Consideraciones mecánicas.....	26
2.3. Unidad inspectora de cajas.....	27
2.4. Hardware del sistema.....	35
2.4.1. Cámara USB.....	35
2.4.2. Computador.....	37
2.4.3. Interfaz de comunicación del sistema.....	38
2.5. Software de aplicación.....	47
2.5.1. Vision Assistant.....	47
2.5.2. Labview 8.2.....	48

CAPÍTULO 3

3. PROGRAMACIÓN DEL SISTEMA INSPECTOR DE CAJAS.....	52
3.1. Vision Assistant.....	53
3.1.1. Imagen patrón.....	53
3.1.2. Pruebas de evaluación.....	58

3.1.3. Sub VI.....	62
3.2. Elaboración del sub VI del inspector de cajas.....	65
3.2.1. Diagrama de bloques.....	65
3.2.2. Panel frontal.....	77
3.3. Pruebas de inspección.....	86
3.3.1. Variando el punto mínimo de encuentro.....	87
3.3.2. Variando los atributos.....	88
3.3.3. Variando la estrategia de búsqueda.....	91
3.4. Análisis de resultados.....	93

CAPÍTULO 4

4. ANÁLISIS DE COSTOS DEL SISTEMA INSPECTOR DE CAJAS.....	96
4.1. Costos del hardware.....	97
4.2. Costos del software.....	100
4.3. Análisis.....	101

CONCLUSIONES Y RECOMENDACIONES

BIBLIOGRAFÍA

ANEXOS

ABREVIATURAS

Bit	Es la unidad más pequeña de información que utiliza un ordenador
Bps	Bits por segundo
Byte	Conjunto de 8 bits
cm.	Centímetro
CMOS	Semiconductor complementario del óxido de metal
DB9	Conector para puerto serie
Frames	Número de imágenes
FPS	Frames por segundo
Hz	Hertzios
ICSP	Programación serial en circuito
Labview	Laboratory Virtual Instruments Electronics Workbench
MAX232	Integrado de conversión de TTL a RS232
MHz	Megahertzios
NI	National Instruments
PCB	Circuito Impreso
PIC	Microcontrolador de Microchips
RS232	Estándar de comunicación serial
Rx	Recepción
SubVI	Instrumento Virtual de bajo nivel
TTL	Lógica transistor-transistor
Tx	Transmisión
USB	Bus Serial Universal
V	Voltio
VI	Instrumento Virtual

INDICE DE FIGURAS

Figura 1.1.1 Inspector de cajas.....	5
Figura 1.1.2 Iluminación del inspector de cajas.....	8
Figura 1.2.1 Cámara de Visión artificial.....	9
Figura 1.2.2 Software grafico de aplicación.....	10
Figura 1.3.1.Caja de botellas de coca cola.....	12
Figura 1.4.1. Implementación prototipo inspector de cajas.....	14
Figura 2.2.1.1 Imagen del software patmax.....	19
Figura 2.2.1.2 Imagen del software MVImpact.....	20
Figura 2.2.1.3 Ventana del software I+Solex.....	19
Figura 2.2.2.1 Cámara de Visión Sick.....	23
Figura 2.2.2.2 Cámara CMOS BCi4.....	24
Figura 2.2.2.3 Cámara CMOS FCi4-14000.....	25
Figura 2.3.1. Caja de inspección.....	28
Figura 2.3.2. Esquema General del Circuito de Control.....	29
Figura 2.3.3 Esquemático del circuito de control en Proteus en ISIS.....	30
Figura 2.3.4 Pistas del circuito de control en Proteus ARES.....	30
Figura 2.3.5. PCB del circuito de control en Proteus ARES.....	31
Figura 2.3.6. Parte superior de la baquelita del circuito de control.....	34
Figura 2.3.7. Parte inferior de la baquelita del circuito de control.....	34
Figura 2.4.1.1 Cámara Genius USB.....	36
Figura 2.4.2.1 Laptop y cable USB-RS232.....	38

Figura 2.4.3.1 Capas del USB.....	40
Figura 2.4.3.2 Periféricos USB conectados a la PC.....	44
Figura 2.4.3.3 Grafico del puerto serie RS-232 de 9 y 25 pines macho.....	46
Figura 2.5.1.1 Ventana de Vision Assistant.....	48
Figura 2.5.2.1 Panel frontal y diagrama de bloques Labview.....	50
Figura 3.1.1.1 Abrir imagen Vision Assistant.....	54
Figura 3.1.1.2 Procesador de funciones para imágenes de color.....	55
Figura 3.1.1.3 Icono de coincidencia del color patrón.....	55
Figura 3.1.1.4. Creación de la imagen patrón de la caja.....	56
Figura 3.1.1.5 Configuración del template.....	57
Figura 3.1.2.1 Imagen patrón.....	58
Figura 3.1.2.2 Evaluación con estrategia conservativa.....	60
Figura 3.1.2.3 Evaluación con estrategia muy agresiva.....	61
Figura 3.1.3.1 Script o código Fuente en Vision Assistant.....	63
Figura 3.1.3.2 Creación de un sub VI dado el script.....	63
Figura 3.1.3.2 Panel Frontal del éxito.VI.....	64
Figura 3.1.3.4 Diagrama de Bloques del éxito.VI.....	64
Figura 3.2.1.1 Cámara ON.VI.....	66
Figura 3.2.1.2 Imaq USB Grab Acquire.VI.....	67
Figura 3.2.1.3 Imaq USB close.VI.....	67
Figura 3.2.1.4 VISA Configure Serial Port.VI.....	68
Figura 3.2.1.5 VISA Open.....	68

Figura 3.2.1.6 VISA Read	69
Figura 3.2.1.7 VISA Write.....	69
Figura 3.2.1.8 VISA Close.....	69
Figura 3.2.1.9 Imaq Create.VI.....	70
Figura 3.2.1.10 Imaq read and image and Vision info.....	72
Figura 3.2.1.11 Imaq Setup Match Color Pattern.....	73
Figura 3.2.1.12 Imaq Match Color Pattern.....	74
Figura 3.2.1.13 IVA Add offset to matches.....	76
Figura 3.2.1.14 Overlay Blister Pack Match Position.VI.....	76
Figura 3.2.1.15 Contador en milisegundos.....	77
Figura 3.2.1.16 Dispositivo IMAQ.....	77
Figura 3.2.2.1 Nombre Cámara USB.....	78
Figura 3.2.2.2 Puerto Serial.....	78
Figura 3.2.2.3 Display de estado.....	78
Figura 3.2.2.4 Imagen patrón.....	79
Figura 3.2.2.5 Imagen Principal.....	79
Figura 3.2.2.6 Lectura desde el puerto serial.....	80
Figura 3.2.2.7 Tiempo (ms).....	80
Figura 3.2.2.8 Diagrama de Bloques en el estado Iniciando.....	81
Figura 3.2.2.9 Panel Frontal en el estado “Esperando botón”.....	82
Figura 3.2.2.10 Diagrama de Bloques “Esperando botón”.....	83
Figura 3.2.2.11 Diagrama de bloques en estado “Procesando imagen”.....	83

Figura 3.2.2.12 Diagrama de bloques del patrón de encuentros a color.....	84
Figura 3.2.2.13 Panel frontal de una caja aceptada.....	85
Figura 3.2.2.14 Panel frontal de una caja rechazada.....	85
Figura 3.2.2.15. Diagrama de bloque "Siguiete jaba".....	86
Figura 3.3.1.1. Prueba de Inspección con mínimo punto de encuentro 800.....	88
Figura 3.3.2.1. Prueba de inspección con atributo color.....	90
Figura 3.3.2.2. Atributo color desde punto mínimo de encuentro 875.....	91
Figura 3.3.3.1. Prueba de inspección con estrategia de búsqueda muy agresiva.....	92
Figura 3.3.3.2. Estrategia de búsqueda muy agresiva desde punto mínimo de encuentro 850.....	93

INDICE DE TABLAS

Tabla 2.3.1. Descripción de los parámetros de envío y recepción.....	33
Tabla 2.4.3.1 Señales comunes según pines asignados.....	46
Tabla 3.1.2.1 Prueba de evaluación con estrategia conservativa.....	59
Tabla 3.1.2.2 Prueba de evaluación con estrategia muy agresivo.....	61
Tabla 3.3.1.1. Variando el Mínimo Punto de encuentro.....	87
Tabla 3.3.1.2. Variando los atributos.....	89
Tabla 3.3.1.2 Variando la estrategia de búsqueda.....	92
Tabla 4.1.1 Costos de la planta.....	98
Tabla 4.1.2 Costos de la placa electrónica.....	99
Tabla 4.1.3 Costos de equipos utilizados.....	99
Tabla 4.1.4 Costos total del hardware.....	100
Tabla 4.2.1 Costos de los software usados.....	100
Tabla 4.3.1 Total de horas trabajadas.....	102

INTRODUCCIÓN

El inspector de cajas es un sistema que se propone como solución al problema de las jabas incompletas debido a fallas en equipos del proceso de encajonamiento en una planta embotelladora. En la actualidad la verificación de las jabas en el área de post encajonamiento es realizada por operadores que se encargan visualmente de identificar los faltantes. El inspector de cajas es un sistema que verifica y optimiza la calidad del proceso, por una computadora con un software especial y una cámara, ambos harán el mismo trabajo de forma automatizada. Esto permitirá el ahorro de tiempo en el proceso mejorando la condición del mismo.

La solución esta basada en la herramienta de programación grafica Labview y el procesador de imágenes Vision Assistant, los cuales interactúan con una cámara de fotos que se encarga de tomar imágenes de las jabas luego del proceso de encajonamiento en la línea de producción. Las imágenes son almacenadas en una base de datos para luego ser comparadas con modelos patrón a través del software creado. El modelo de imágenes patrón fue elaborado usando la herramienta Vision Assistant y el algoritmo de comparación de imágenes fue programado en Labview como núcleo del proyecto. Este algoritmo decide si una jaba es aceptada o rechazada de acuerdo al número de botellas encontradas. Si la jaba esta completa es

aceptada y si tiene al menos una botella faltante, alguna botella rota o algún objeto extraño, etc. la jaba es rechazada.

El Inspector de cajas se presenta como una solución para las empresas embotelladoras que requieren un proceso de inspección eficiente y automatizada.

CAPÍTULO 1

1. GENERALIDADES

El presente capítulo explica la solución de uno de los problemas que se presentan en las industrias embotelladoras; este es el faltante de botellas en las cajas o jabs después del proceso de encajonamiento de las mismas en una línea de producción.

Se presenta una de las soluciones del problema anteriormente descrito acercándonos lo más real posible y académicamente permitido a un sistema de inspección visual industrial.

1.1. ¿Qué es un inspector de cajas?

En el campo industrial existe diversidad de procesos, en el cual se realizan diferentes formas de inspección, determinadas por el enfoque que se necesita verificar.

En la línea de producción de las industrias embotelladoras existen diferentes tipos de anomalías causadas en el proceso de encajonamiento (una máquina que ubica las botellas en las cajas), que al final de éste, presenta problemas como botellas rotas, objetos extraños, plásticos, botellas faltantes en las jabas, etc.

El problema observado es la falta de un sistema adecuado de detección de faltantes de botellas. Para solucionar éste, se ha desarrollado un sistema visual denominado inspector de cajas que contiene una cámara, integrada mediante visión artificial, observando el 100% de la caja la cual contiene botellas distribuidas matricialmente; éste encierra la jaba que es iluminada mediante focos colocados estratégicamente para mantener constante la luz. También contiene un interfaz de comunicación entre el hardware anteriormente descrito y el software de inspección que es el que analizará la imagen de la cámara previamente adquirida, proporcionando una salida indicadora de correcto/incorrecto (ON/OFF) que garantiza que solamente pase el producto adecuado.

Este sistema se adapta a diversas formas de inspección y parámetros previamente configurados, según los formatos que presenten las botellas.

Aunque se trata de un sistema de bajo costo, tiene prestaciones de mucha variedad, incorporando los últimos avances de la tecnología.



Figura 1.1.1 Inspector de cajas

Ventajas del Inspector de Cajas

En la línea de producción podemos encontrar diversos problemas que diariamente en forma manual lo resuelven los operadores de la misma. Por ejemplo, cuando se llenan las cajas de botellas en la encajonadora, muchas botellas faltan y el operador tiene que agregarlas para completar las cajas y así continuar con el proceso.

Para solucionar éste inconveniente y otros, existe el inspector de cajas, el cual detallaremos algunas ventajas para su aplicación.

Ventajas para el Departamento de Producción

- Obtiene la producción diaria dada por el software, que mejora el nivel de calidad para cumplir las metas requeridas en la planta.

- Ahorra recursos económicos, al disminuir el porcentaje de cajas faltantes de botellas dirigidas a la etapa de almacenamiento y al evitar que el operador continuamente coloque las botellas dentro de la caja.

- Agilita el traslado de las cajas a su lugar de destino, es decir, las cajas completas se dirigen a la paletizadora o zona de recolección de cajas, mientras que las incompletas se las direcciona a un lugar de almacenamiento de producto incompleto.

Ventajas para los usuarios

Como las empresas son las que usarían el inspector de cajas en sus líneas de producción, son ellas las encargadas en darle una buena operación con un buen mantenimiento a éste equipo. Sus principales ventajas son:

- Fácil acceso para el operador del equipo, porque el inspector de cajas contiene un monitor de visualización para su observación técnica. Por ejemplo: Observamos el formato de la botella que se está inspeccionando.

- Facilidad en la calibración del equipo, dependiendo de las características en el análisis de la botella como color, forma y brillo. Esto requiere que la empresa no solicite el servicio técnico porque el operador del equipo puede hacer la calibración respectiva.



Figura 1.1.2 Iluminación del inspector de cajas

1.2. Descripción del funcionamiento del inspector de cajas

El inspector de cajas fue diseñado con el objetivo de determinar si el número de botellas se encuentra completo en la caja, que se da a la salida de la encajonadora dentro de la línea de producción.

La inspección comienza cuando la caja se encuentra en posición ideal para que se adquiera la imagen, es decir, el 100% de la caja capturada. Después, con el mejor ángulo de visión la cámara espera una señal para capturar la imagen de la caja.



Figura 1.2.1. Cámara de visión artificial

La señal que se espera es proveniente de un sensor de entrada de la jaba al inspector de cajas, la cual es leída por una tarjeta que adquiere datos al software del computador; éste a su vez manda una información hacia el programa por medio de un interfaz de comunicación, cuya información es receptada por el software de inspección, y esta habilita la cámara para adquirir la foto.

Una vez adquirida la foto, ésta pasa a ser analizada por el software de inspección, comparándose con una imagen patrón previamente adquirida.

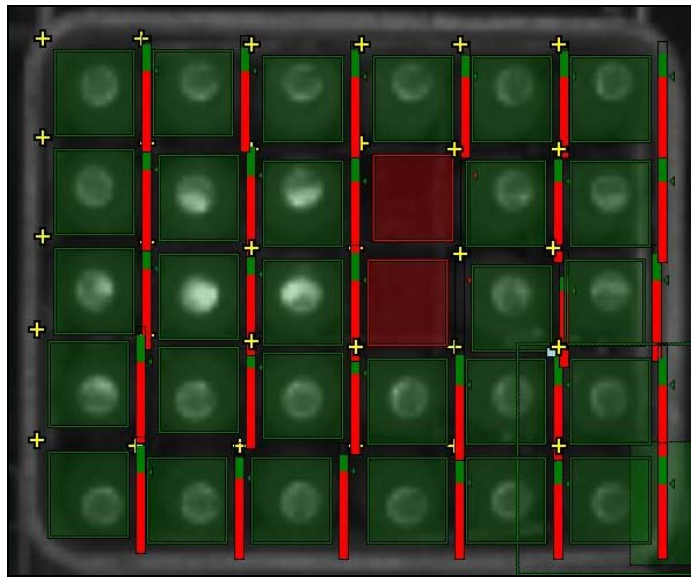


Figura 1.2.2. Software gráfico de aplicación

Cuando la imagen es analizada, el software muestra si la caja es aceptada o rechazada; y además emite una señal de salida para que en hardware, si la jaba es rechazada, lo saque de curso de producción o de jabas aceptadas. Ésta señal la manda una información desde el programa por medio de un interfaz de comunicación, cuya información es enviada por el software de inspección. Una vez terminado este proceso la cámara se encuentra lista esperando la señal de entrada del sensor para adquirir la imagen de la nueva caja en la línea de producción.

1.3. Aplicaciones del inspector de cajas

En el Ecuador las empresas han planteado soluciones dentro de su producción que ha mejorado la misma, gracias al auge de la tecnología y a diferentes tipos de sistemas, como los de visión artificial.

El sistema inspector de cajas es desarrollado mediante visión artificial, aunque el campo de acción es muy variado, lo utilizamos para la adquisición y verificación de imágenes en el área industrial.

El sistema inspector de cajas se lo puede adaptar para que realice inspecciones como el de verificar etiquetas, inspección de botellas, etc.

También el sistema inspector de cajas trabaja para varios formatos de botella como los de 200 CC, 300 CC, 500 CC, 1000 CC, 1500 CC, 2000 CC, 2500 CC, 3000 CC, además con diferentes tipos de cajas como las de 6 envases, 12 envases, 24 envases, etc; esto se puede realizar comparándolas con la imagen patrón que serán las tapillas de los mismos formatos, siempre y cuando las tapillas de los envases sean similares.



Figura 1.3.1 Caja de botellas de coca cola

Este sistema se adapta dependiendo al tipo de formato que se desee inspeccionar y también a que elemento se desee inspeccionar.

1.4. Alcance del proyecto

Este proyecto está enfocado para realizar la inspección de cajas o jabas, se trata de un prototipo que por medio de una cámara USB, capta una imagen de una caja o jaba para ser comparada con una imagen patrón en su inspección. Definiendo al prototipo como una representación limitada en el diseño de un sistema inspector de cajas que nos permite experimentar y probar botellas similares en pequeña escala.

La jaba es rectangular y mide 9,5cm X 8cm está distribuida matricialmente en 3 X 2 donde 3 es el número de columnas y 2 de filas y encerrada por una caja de madera cuadrada de 30 cm de lado

y con una de sus tapas de fácil apertura para poder observar la jaba con las pequeñas botellas.

Las botellas del prototipo son pequeñas y de vidrio con características físicas similares cuyas tapillas blancas tienen un logotipo en forma de T y tienen 8 cm. de altura.

La cámara está colocada en el centro de la caja de madera en la parte superior, de tal manera que se puedan observar todas las tapillas que luego son comparadas con la imagen patrón para la inspección en el software Labview.

La imagen patrón es adquirida y analizada por el software Vision Assistant y aprobada mediante pruebas de inspección.

La caja de madera encierra la jaba de botellas con el objetivo que la luz artificial, puesta por dos pequeños focos a los lados de la misma, se mantenga constante; y así obtener una imagen apropiada para la inspección.

El interfaz de comunicación entre el hardware y el software es el USB-DB9 cuyas características se explicarán detalladamente en el capítulo posterior.



Figura 1.4.1. Implementación prototipo inspector de cajas

Una alternativa para la aplicación es que pueda ser acoplada a cualquier tipo de industria embotelladora para poder llevar acabo esto se necesita realizar las variaciones y análisis tanto hardware como en software.

CAPÍTULO 2

2. DISEÑO Y ARQUITECTURA DEL SISTEMA INSPECTOR DE CAJAS

En el presente capítulo se detallan los requerimientos de diseño y arquitectura, el software de aplicación, además se explica las consideraciones físicas para la mejor captura de imágenes, como es la selección de la luz y los diferentes tipos de variantes que permiten la construcción del sistema inspector de cajas.

2.1. Consideraciones en el diseño

Comprendiendo los requerimientos del diseño, se trata de crear un sistema de visión artificial para la detección de cajas incompletas, las cuales pasaran en una banda transportadora a una velocidad

constante y debidamente posicionadas entre dos rieles que representa en nuestro prototipo una jaba estática a la cual capturaremos imágenes de la cámara USB, luego que entre la jaba por nuestro inspector de cajas representado por una botonera de entrada digital y esperando un tiempo debidamente programado en Labview.

La cámara deberá contar con un ángulo de visión concentrando el área de análisis de la jaba por lo que se fijará en la parte superior central de la caja de madera a una altura de 20 cm. de las tapillas a ser inspeccionadas.

El sistema inspector de cajas debe ser capaz de detectar la caja cuando yo presione una botonera que indica que la caja ya se encuentra en el área de análisis en representación del sensor de la señal de entrada comúnmente visto en las industrias. Además que las cajas rechazadas se mostraran a la salida del hardware con un led en representación de un pistón que comúnmente lo saca de curso del trayecto de jabas aceptadas que sirven para la producción.

Cabe resaltar que para la detección de las botellas en la caja tiene que ser todas las tapillas del mismo formato, aunque si no lo tienen ésta botella no será inspeccionada.

También se considera no utilizar la tarjeta de adquisición de datos, diseñando una alternativa diferente con un pequeño circuito que nos permite enviar y recibir señales para comunicarnos entre el software y hardware del sistema mencionado anteriormente.

2.2. Tecnología empleada

Al momento de escoger un software de visión, lo más importante que se debe considerar es que interactúe con la cámara, que mejor se ajusta a la aplicación y las condiciones económicas de la empresa.

En este caso se utilizaron herramientas de tipo académicas y elementos de bajo costo para realizar el proyecto.

2.2.1. Tipos de software

Después que se ha detallado las consideraciones de diseño del proyecto y sus requerimientos se procede a mencionar y

describir de manera general varios software que existen en el mercado.

PatMax

Es un sistema de Visión que puede localizar piezas de forma repetitiva, que sufra grandes variaciones en la orientación, cambio de tamaño o el aspecto de las piezas, que incrementa significativamente el rendimiento y la fiabilidad de la inspección de Visión.

Estas variaciones incluyen:

- Contraste de las piezas
- Piezas múltiples
- Cambios en la iluminación
- Enfoque de las imágenes
- Aspecto degradado
- Piezas parcialmente ocultas

Patmax es el estándar de rendimiento de la industria para el software de localización de características y piezas, utiliza una

tecnología avanzada de correspondencia de patrones geométricos para localizar piezas de manera fiable y precisa. Incluso en las condiciones más desagradables, esta herramienta puede reducir significativamente o eliminar los costes y los requisitos de fijación para localizar piezas o características, además PatMax ofrece el máximo rendimiento y fiabilidad posible de inspección de Visión en un sistema de Visión.



Figura 2.2.1.1 Imagen del Software Patmax

MVImpact

Es un software modular para procesamiento de imagen de última generación para integradores de Visión y fabricantes de equipos, compatibles con las cámaras y tarjetas Matriz visión.

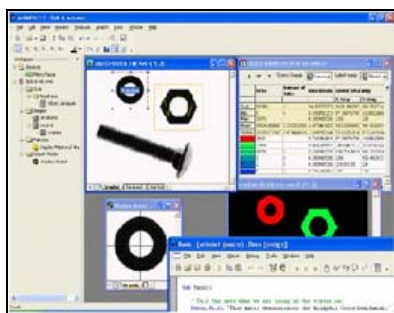


Figura 2.2.1.2 Imagen del software MVImpact

OpenCV

OpenCv es la abreviación de The Open Computer Vision Library, estas es una librería de Visión artificial desarrollada por Intel en el 2003, esta librería, que es código abierto desarrollado en C, presenta las siguientes características: estabilidad, escalabilidad, rápido crecimiento y desarrollo, lo que garantiza su continuidad y la hacen especialmente útil en el ambiente universitario.

I+Solex

Es un software de procesamiento y análisis de imágenes desarrollado en Java. Permite capturar, mostrar editar, analizar imágenes, crear y configurar videos. Esta especialmente creado para las aplicaciones científicas.

Una característica de I+Solex permite el manejo de diversos tipos de cámaras, tanto digitales, vía firewire (IEEE 1394), USB, Camera Link, como analógicas, vía tarjeta digitalizadora (Frame Grabber).

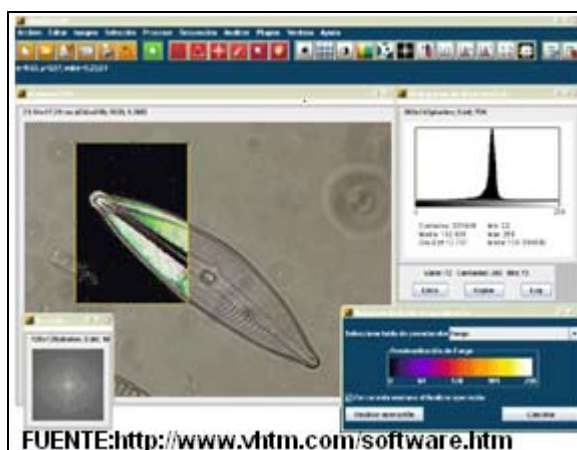


Figura 2.2.1.3 Ventana del software I+Solex

PyVison

Es un software que tiene como objetivo brindar soluciones de Visión artificial para el sector industrial, se caracteriza por su simplicidad, flexibilidad, su eficacia, su software es configurable para cualquier aplicación.

Vision Assistant

Es una herramienta que nos sirve para analizar el proceso de imágenes y crear prototipos para diferentes tipos de aplicaciones de Visión artificial, además en conjunto con el software Labview puede crear algoritmos de solución.

Más adelante definiremos detalladamente las características de este software que se utilizará para nuestro prototipo.

2.2.2. Tipos de cámara

Una vez establecido el diseño del sistema y el software a emplear en el mismo el siguiente paso es elegir una cámara que vaya de acuerdo a las necesidades que se requiere.

Aquí detallamos a continuación algunas de las cámaras que nos ayuda con el proceso de análisis de inspección.

IVC3D

La cámara inteligente de 3 dimensiones IVC-3D proporciona una nueva dimensión en los sistemas de Visión, dotándolos de inteligencia y mejorando la flexibilidad de los procesos de automatización.

La cámara IVC-3D es el primer sistema en el mundo que combina la flexibilidad de una cámara inteligente y la potencia del 3D a alta velocidad en un sistema industrial robusto.



FUENTE: <http://www.oxfordsensor.com/ivc3d.ht>
Figura 2.2.2.1 Cámara de Visión Sick

BCi4

La BCi4 es una cámara es muy compacta de 50*50*50 mm, tiene una alta resolución de cámara CMOS con un sensor de

1280*1024 color y monocromo que tiene una salida digital de 12 bit a 20 MHz píxel que permite un excelente contraste.

También posee una memoria de 8 Mbytes que es usada como imagen FIFO en USB 2.0, esta cámara también puede ser usada como calibración de datos y otros propósitos.



Figura 2.2.2.2 Cámara Cmos BCi4

FCi4-14000

Esta cámara es muy dinámica al momento de su uso, presenta un sensor de 14 Mega píxeles, tiene un formato de salida digital de 12 bits a 60 MHz y 3.25 frames por segundo, se usa para calibrar e inspeccionar formas y datos.



Figura 2.2.2.3 Cámara CMOS FCi4-14000

USB

La cámara USB es un transductor óptico que captura las imágenes y las convierte en una señal eléctrica, para luego con los demás elementos del sistema de adquisición y un PC se procesa y analiza.

Más adelante se detallará específicamente la función que tiene esta cámara en nuestro prototipo.

2.2.3. Consideraciones mecánicas

Entendido los requerimientos para la implementación del proyecto, el sistema debe ser capaz de detectar la caja que se encuentra inhabilitada para seguir a la siguiente etapa del proceso.

Para que luego, un actuador empujaría la caja que se encuentre incorrecta como una solución que la industria le da para rechazar la caja; en nuestro prototipo el actuador es reemplazado por un diodo led de salida que indica que la jaba se rechaza y éste a su vez es controlado por el VISA del software Labview en conjunto con el PIC16F628A que se lo programa previamente.

También el sistema inspector de cajas común en las industrias envía una señal hacia el software mediante un sensor el cual puede ser fotoeléctrico o réflex. En nuestro prototipo reemplazamos la función del sensor por una botonera.

2.3. Unidad inspectora de cajas

La unidad inspectora de cajas se lo puede definir como un sistema de visión que utiliza diferentes tipos de identificación de cajas porque tiene una amplia gama de resultados. Este es el encargado de la detección de errores de las tapillas de las botellas contenidas en la caja mediante la adquisición de imágenes a través de la cámara.

La unidad inspectora de cajas esta conformado por:

- Caja de inspección
- Circuito de control
- Cámara USB
- El computador
- Interfaz de comunicación
- Software de aplicación

La caja de inspección es el lugar donde se va a realizar todo la operación de análisis y detección de las cajas de la línea de producción es el centro de nuestro prototipo y se lo muestra a continuación.



Figura 2.3.1. Caja de inspección

El circuito de control es el que une el hardware controlado (la botonera y el diodo led) con el software de control Labview en comunicación serial.

El circuito está conformado principalmente por el PIC16F628A y el PIC MAX232. El primero es programado por el software mikroBasic y el segundo es que realiza la recepción y el envío de datos al software por medio de la comunicación serial con DB9.

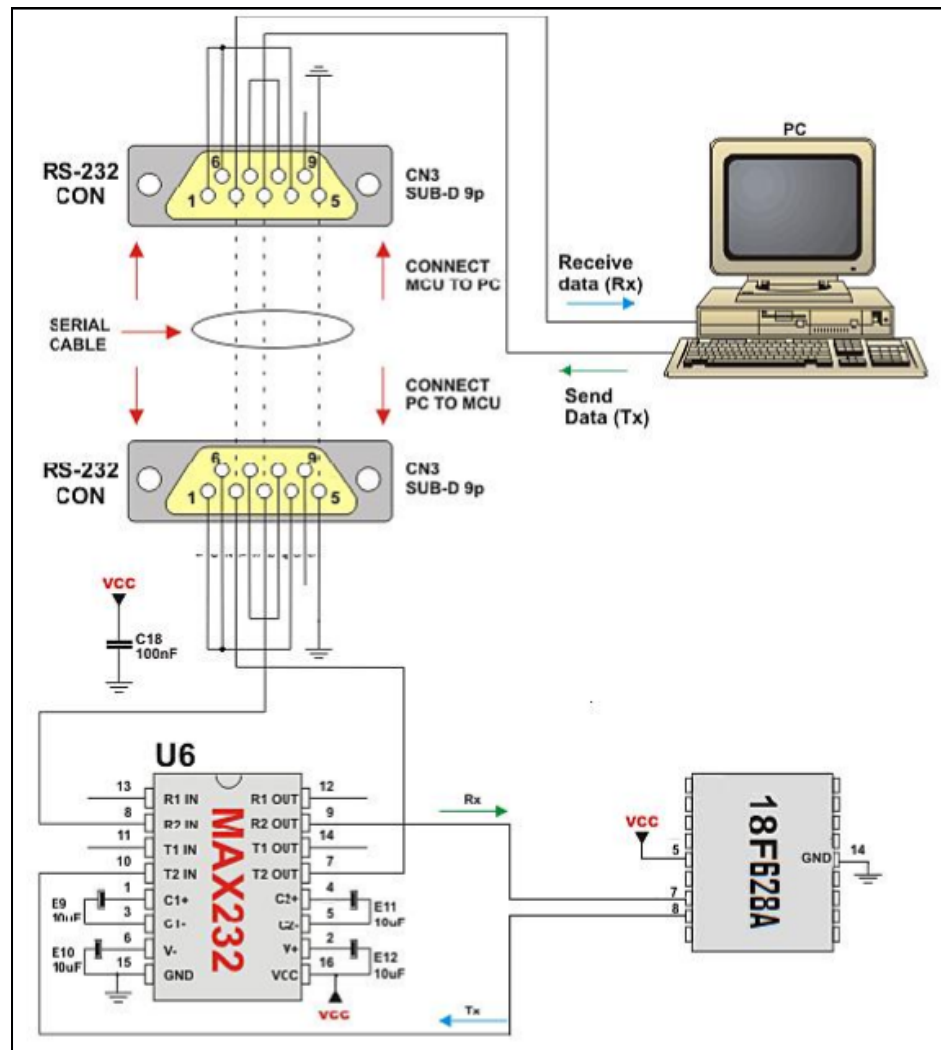


Figura 2.3.2. Esquema General del Circuito de Control

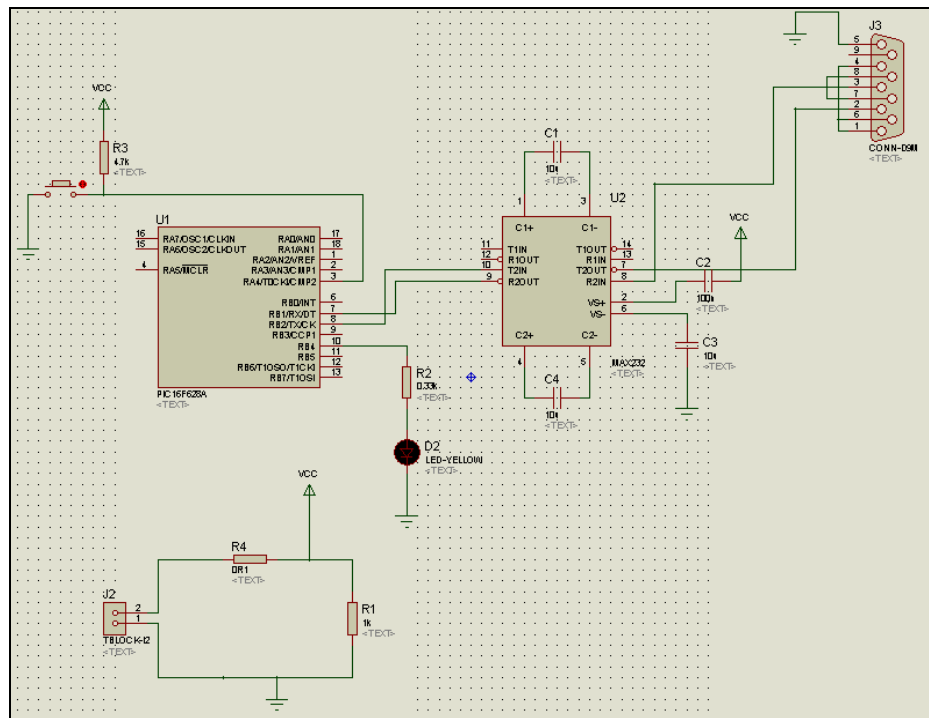
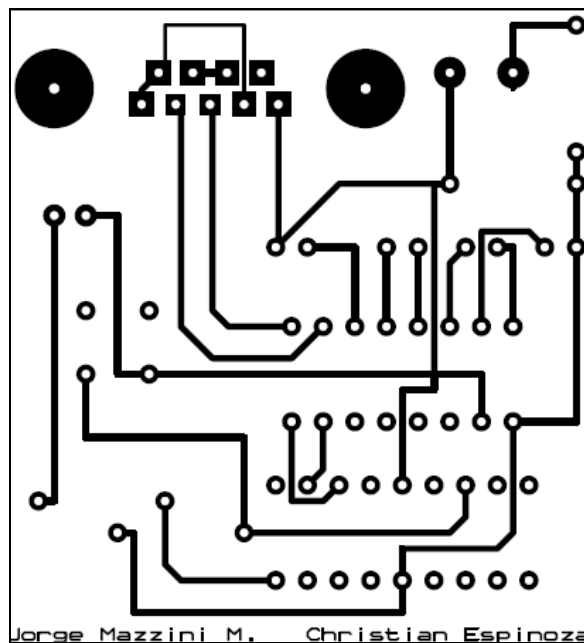


Figura 2.3.3 Esquemático del circuito de control en Proteus en ISIS



Jorge Mazzini M., Christian Espinoza
 Figura 2.3.4 Pistas del circuito de control en Proteus ARES

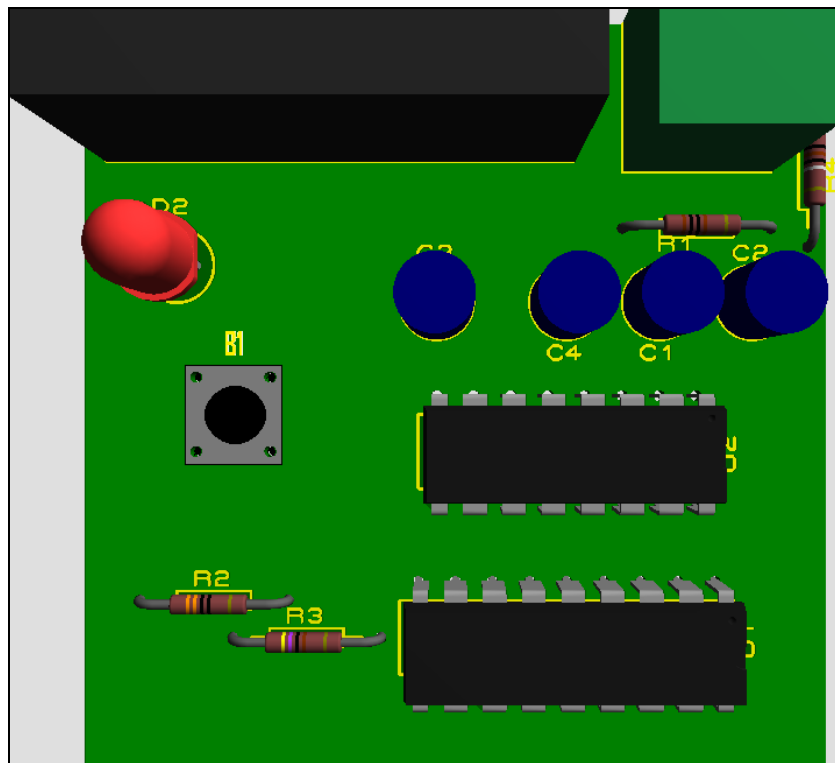


Figura 2.3.5. PCB del circuito de control en Proteus ARES

A continuación la programación en el PIC16F628A que se hace a una tasa de transferencia a 9600 Bps y con un reloj interno de 4 MHz.

```
program Inspector Box
```

```
dim Recibido as byte
```

```
dim i as byte
```

```
main:
```

```
    CMCON = 7    'Puerto A digital
```

```
    Usart_Init(9600)    'Inicia el puerto serial
```

```
    TRISB.4 = 0    'Pin 4 del puerto B como salida
```

```
    TRISA.4 = 1    'Pin 4 del puerto A como entradas
```

```
    PORTB.4 = 0    'Al principio el led está apagado
```

```

for i = 1 to 3
    PORTB.4 = 1
    delay_ms(100)
    PORTB.4 = 0
    delay_ms(100)
next i

Usart_Write("R")

while true

    'Verificamos si ha activado el pin 4
    if PORTA.4 = 0 then
        while PORTA.4 = 0 wend
        Usart_Write("B")
        delay_ms(100)
    end if

    'Verificamos si no ordenan prender el LED
    if Usart_Data_Ready = 1 then
        Recibido = Usart_Read
        select case Recibido
        case "L"
            PORTB.4 = 1
            delay_ms(500)
            PORTB.4 = 0
        end select
    end if

wend

end.

```

- La lógica para recepción y envío de datos se describe a continuación:
- Inicializamos el puerto serial y los puertos A como entradas y B como salidas en el pin 4.

- Luego verificamos si se activa el pin 4 de entrada pulsando la botonera; si es verdadero escribe una letra “B” que será leído a su vez por el software Labview para proceder la inspección de las botellas en la caja, sino hay que esperar que se active el pin 4 del puerto A de entrada.
- Luego verificamos si nos ordenan prender el LED del pin 4 de salida, si es verdadero lee una letra “L” que será leído a su vez por el software Labview para proceder al rechazo de la jaba incompleta, sino hay que esperar que se active el pin 4 del puerto B de salida.

Más información sobre estos parámetros se puede encontrar en la tabla 2.3.1.

ASCII	Decimal	Descripción
B	66	Espera la activación de la botonera
L	76	Activa el diodo LED
R	82	Verificación de la activación del pin 4

Tabla 2.3.1. Descripción de los parámetros de envío y recepción

A continuación detallaremos la construcción de la baquelita con todos sus elementos anteriormente simulados en Proteus:



Figura 2.3.6. Parte superior de la baquelita del circuito de control

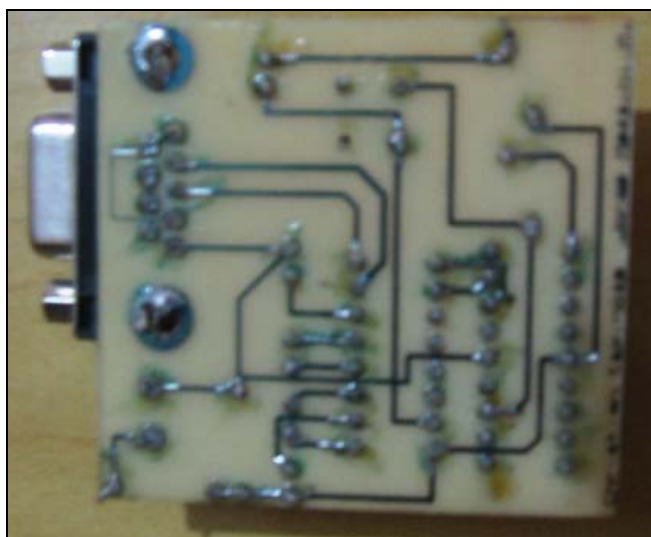


Figura 2.3.7. Parte inferior de la baquelita del circuito de control

Las otras partes de la unidad inspectora de cajas se las describirá en los siguientes subcapítulos.

2.4. Hardware del sistema

Hoy en día existen gran variedad de hardware utilizados en visión artificial, de esta manera hemos querido dar una solución a las exigencias que el mundo real presenta en las industrias embotelladoras.

2.4.1. Cámara USB

Partimos del hecho que una cámara es un transductor óptico que captura las imágenes y las convierte en una señal eléctrica, para luego con los demás elementos del sistema de adquisición y un PC se procesa y analiza. La cámara que nosotros hemos utilizado es la DSB-C120 de la marca Genius, la cual es una cámara de video digital USB, y esta diseñada para ser conectada a la computadora, sus características son:

- Resolución 352*288 Píxeles.
- Capacidad de transmisión de imágenes: 30 frames por segundo.
- Interfaz: USB.
- Sensor de imágenes CMOS.

- Conexión plug & play.

Cabe indicar que las CMOS son pocos sensibles a la luz, de menor calidad en resolución pero mas baratos que otros tipos.



Figura 2.4.1.1 Cámara Genius USB

Para la implementación se eligió la cámara mencionada anteriormente debido a que la banda transportadora detendrá las cajas en frente de la cámara debido a que esta webcam que es muy lenta en el refrescamiento de la imagen y al envío de datos a la computadora, y no industrial debido a su alto costo.

Además de que es un proyecto tipo académico debido a que no se están considerando las variables reales de un proceso en una industria como es el caso de la velocidad de las cajas, entre otros.

La característica plug & play, es que con solo conectar el dispositivo al CPU este sea capaz de interpretar la información y reproducirla inmediatamente.

La cámara DSB-C120 es más que una simple Webcam, es un completo sistema de comunicación visual ya que provee lo último en tecnología óptica. Adicionalmente contiene un software de edición versátil de video.

2.4.2. Computador

La computadora es la unidad principal del sistema, esta es una máquina electrónica que recibe y procesa de datos para convertirlos en información útil. La PC se encarga de ejecutar el programa que controla todo el proceso. El programa que se desarrolló funciona bajo Windows XP. La PC en que se trabajo durante todo el proyecto consta de los siguientes requisitos:

- Pentium 4 2.66GHz, Service Pack 2
- Memória RAM de 496MB
- Monitor para visualizar el proceso
- Disco duro de 77 GB

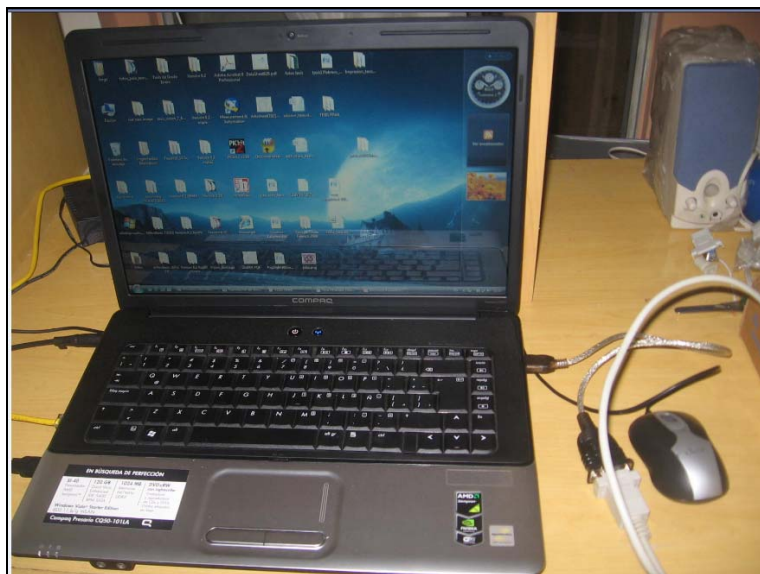


Figura 2.4.2.1 Laptop y cable USB-RS232

2.4.3. Interfaz de comunicación del sistema

La conexión física y funcional del sistema es una parte muy importante en cualquier proyecto de este tipo, la comunicación y el intercambio de datos entre si se denomina protocolo de comunicación.

Hay diferentes tipos de protocolo pero en este tema se va a explicar la comunicación por el puerto USB y la comunicación serial por el puerto RS-232 que se utilizan para el desarrollo del proyecto.

Funcionamiento y Estructura del USB

Trabaja como interfaz para transmisión de datos y distribución de energía, que ha sido introducida en el mercado de PC's y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Esta interfaz de 4 hilos, 12 Mbps y "plug and play", distribuye 5 v para alimentación, además que contribuye de forma notable al desarrollo de la transmisión de datos mediante PC.

Emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, esta el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran mas abajo en la pirámide sufran retardo. A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores.

El sistema de bus serie universal USB consta de tres componentes:

- Controlador
- Hubs o concentradores
- Periféricos

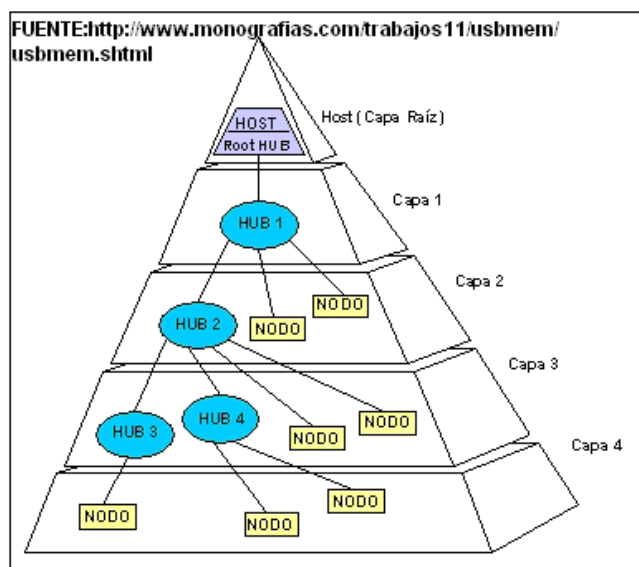


Figura 2.4.3.1 Capas del USB

Controlador

Reside dentro del PC y es responsable de las comunicaciones entre los periféricos USB y la CPU del PC. Es también responsable de la admisión de los periféricos dentro del bus, tanto si se detecta una conexión como una desconexión. Para cada periférico añadido, el controlador determina su tipo y le

asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo. Si se producen errores durante la conexión, el controlador lo comunica a la CPU, que, a su vez, lo transmite al usuario.

Una vez se ha producido la conexión correctamente, el controlador asigna los periféricos los recursos del sistema que este precise para su funcionamiento.

El controlador también es responsable del control de flujo de datos entre el periférico y la CPU.

Concentradores o Hubs

Son distribuidores inteligentes de datos y alimentación, y hacen posible la conexión a un único puerto USB de 127 dispositivos. De una forma selectiva reparten datos y alimentación hacia sus puertas descendentes y permiten la comunicación hacia su puerta de retorno o ascendente. Un hub de 4 puertos, por ejemplo, acepta datos del PC para un periférico por su puerta de retorno o ascendente y los distribuye a las 4 puertas descendentes si fuera necesario.

Los concentradores también permiten las comunicaciones desde el periférico hacia el PC, aceptando datos en las 4 puertas descendentes y enviándolos hacia el PC por la puerta del retorno.

Además del controlador, el PC también contiene el concentrador de raíz. Este es el primer concentrador de toda la cadena que permite a los datos y a la energía pasar a uno o dos conectores USB del PC, y de allí a los 127 periféricos que, como máximo, puede soportar el sistema. Esto es posible añadiendo concentradores adicionales. Por ejemplo, si el PC tiene una única puerta USB y a ella le conectamos un hub o concentrador de cuatro puertas, el PC se queda sin más puertas disponibles. Sin embargo, el hub de 4 puertas permite realizar 4 conexiones descendentes. Conectando otro hub de 4 puertas a una de las 4 del primero, habremos creado un total de 7 puertas a partir de una puerta del PC. De esta forma, es decir, añadiendo concentradores, el PC puede soportar hasta 127 periféricos USB.

La mayoría de los concentradores se encontrarán incorporados en los periféricos. Por ejemplo, un monitor USB puede contener

un concentrador de 7 puertas incluido dentro de su chasis. El monitor utilizara una de ellas para sus datos y control y le quedaran 6 para conectar allí otros periféricos.

Periféricos

USB soporta periféricos de baja y media velocidad. Empleando dos velocidades para la transmisión de datos de 1.5 y 12 Mbps se consigue una utilización más eficiente de sus recursos. Los periféricos de baja velocidad tales como teclados, ratones, joysticks y otros periféricos para juegos, no requieren 12 Mbps. Empleando para ellos 1.5 Mbps, se pueden dedicar más recursos del sistema a periféricos tales como monitores, impresoras, módems, scanner, equipos de audio, que precisan de velocidades más altas para transmitir mayor volumen de datos cuya dependencia temporal es más estricta.

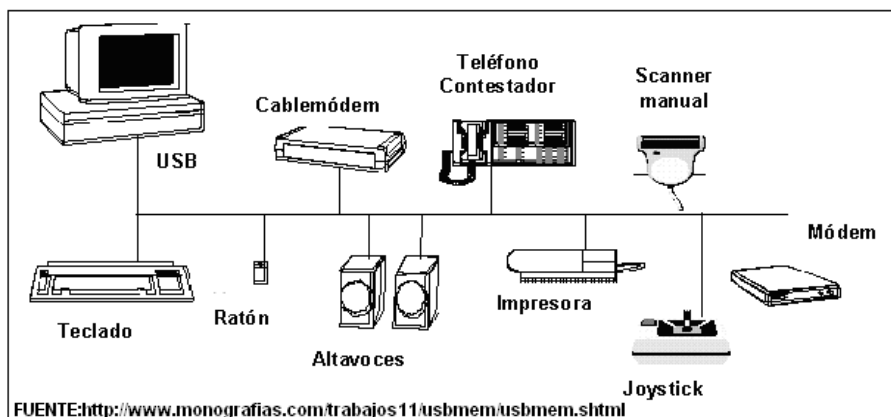


Figura 2.4.3.2 Periféricos USB conectados a la PC

Comunicación serial

La comunicación serie consiste en la transmisión y recepción de pulsos digitales que representan un dato binario a una misma velocidad.

La interfaz física para esta comunicación es el puerto serie donde la transmisión es de bit a bit de forma ordenada. Se ha usado para conectar las computadoras a dispositivos externos como los Mouse, teclados y otros periféricos.

Los bits se representan por códigos que se transmiten en un periodo de tiempo fijo. Este periodo de tiempo es usado para transmitir cada código y se conoce como periodo de baud.

Si se tiene que comunicar dos dispositivos a una velocidad de 2400 bits por segundo (Baudios), el receptor capturará las señales de voltaje que le está enviando el transmisor, y por cada $1/2400$ de un segundo, interpreta dicha señal como un nuevo bits de datos. Si el voltaje es bajo (0 voltios), interpretará el dato como 0. Así el receptor puede obtener un mensaje interpretando una secuencia de datos.

Puerto RS-232

Generalmente se utiliza el protocolo RS-232 para estas comunicaciones que es una interfaz que define un estándar para el intercambio serie de estos datos utilizando un cableado simple desde 3 hilos hasta 25 hilos entre transmisor y un receptor a una distancia máxima de 15 metros.

A inicios el RS-232 original tenía un conector tipo D de 25 pines de los cuales muchos no se utilizaban por lo que se incorporó un conector más pequeño de 9 pines que actualmente se lo utiliza y se lo conoce como DB9.

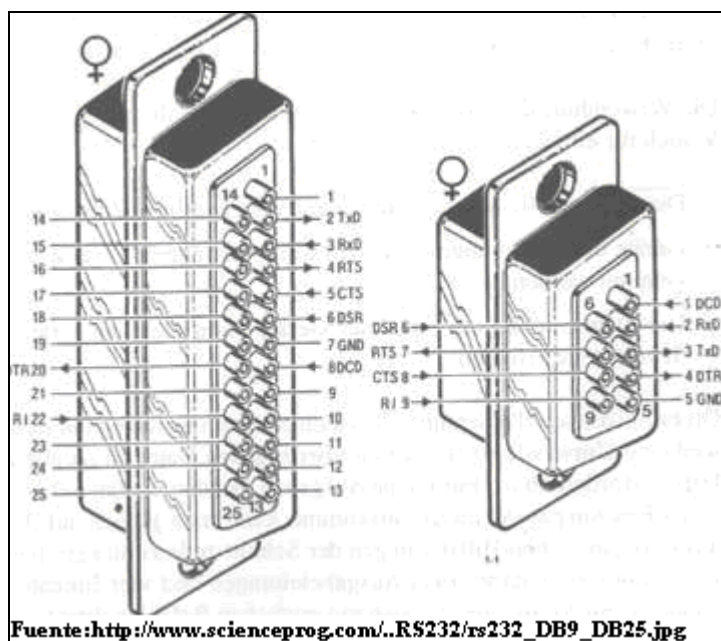


Figura 2.4.3.3 Grafico del puerto serie RS-232 de 9 y 25 pines macho.

Señal		DB-25	DB-9
Common Ground	G	7	5
Transmitted Data	TD	2	3
Received Data	RD	3	2
Data Terminal Ready	DTR	20	4
Data Set Ready	DSR	6	6
Request To Send	RTS	4	7
Clear To Send	CTS	5	8
Carrier Detect	DCD	8	1
Ring Indicator	RI	22	9

Tabla 2.4.3.1 Señales comunes según pines asignados

La tabla anterior muestra la asignación de pines para los dos tipos de conectores.

2.5. Software de aplicación

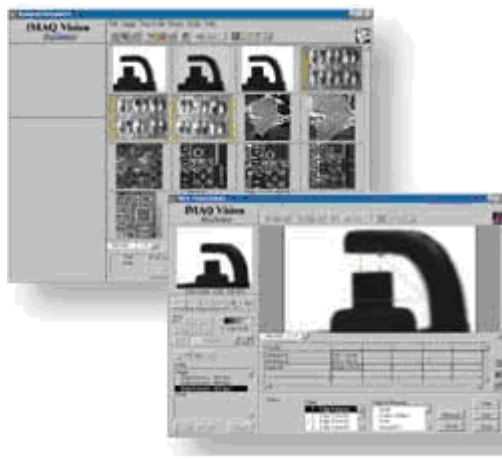
El enfoque de todo software de Visión es tomar decisiones basándose en los objetos de una escena tomada de una imagen. Detallaremos a continuación los softwares que utilizaremos para el desarrollo de nuestro tema y las principales características que estos poseen, su área y su entorno de aplicación.

2.5.1. Vision Assistant

El NI Vision Assistant es una herramienta de National Instruments que nos sirve para analizar procesamiento de imágenes y crear prototipos para aplicaciones, además de crear algoritmos con el asistente de visión que es una característica de la secuencia de comandos.

El Vision Assistant también puede crear un VI en Labview que es realizado por el prototipo de una imagen creado

anteriormente en el programa Vision Assistant oferta tres tipos de adquisiciones de imágenes: Snap, grab and sequence.



Fuente:http://www.ni.com/vision_assistant
Figura 2.5.1.1 Ventana de Vision Assistant

2.5.2. Labview 8.2

El software LABVIEW (Laboratory Virtual Instrument Engineering Wordbench) fue creado por Nacional Instruments en 1976, es una herramienta de programación grafica destinado al desarrollo de aplicaciones que involucren sistemas de adquisición de datos, instrumentación, control, análisis y presentación de datos. Labview da la facilidad de crear una interfaz de usuario que permite estar interactuando con el sistema, valido para cualquier clase de programador incluyendo

a personas con pocos conocimientos en programación debido a su fácil programación gráfica o lenguaje G para crear programas basados en diagramas de bloques llamados instrumentos virtuales o VI's.

La instrumentación virtual se puede describir como un sistema de medición, precisión, análisis y control de señales físicas, con la ayuda de instrumentos virtuales desarrollados desde un computador.

Entre sus principales ventajas se puede citar:

- Es un lenguaje gráfico intuitivo por su facilidad de programación.
- Puede trabajar con programas de otra área de aplicación, como por ejemplo Vision Assistant, Matlab/Simulink, AutoCAD, entre otras.
- Facilidades en el manejo de interfaces de comunicación. Como por ejemplo Puerto Serie, Puerto Paralelo, USB, etc.

- Fácil integración con el Hardware, por ejemplo: tarjetas de adquisición, medición y procesamiento de datos (incluyendo adquisición de imágenes).

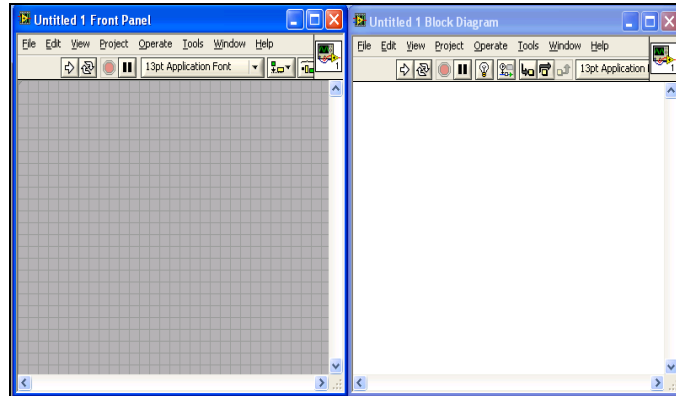


Figura 2.5.2.1 Panel Frontal y diagrama de bloques Labview

Área de aplicación

Su mayor aplicación es en sistemas de medición, monitoreo de procesos y aplicaciones de control. Por ejemplo:

- Automatización de máquinas
- Sistemas de monitoreo de condiciones de máquina
- Visión artificial
- Adquisición y tratamientos de imágenes y datos
- Laboratorios para clases en universidades
- Procesos de control industrial

- Diseño y análisis de sistemas empotrados como las FPGAs y Microprocesadores

Labview también es utilizado en el procesamiento digital de señales, procesamiento en tiempo real de aplicaciones biomédicas, automatización, manipulación de imágenes y audio, diseño de filtros digitales, generación de señales, etc.

CAPITULO 3

3. PROGRAMACIÓN DEL SISTEMA INSPECTOR DE CAJAS

Para la programación del sistema inspector de cajas hemos dividido en cuatro partes éste capítulo. Primero la programación en Vision Assistant 8.5 para la obtención de la mejor imagen patrón. Segundo la obtención del sub-VI inspector de cajas en Labview 8.2. Tercero las pruebas de inspección de las botellas en la jaba. Y por último el análisis de los resultados indicando cuantas cajas se aceptan y se rechazan para la producción de la línea por minuto.

3.1. Vision Assistant

Vision Assistant es una programación muy sencilla de manejar, puesto que es gráfica e interactivo en lo que respecta a aplicaciones de Visión como lo indicamos en el capítulo anterior.

Abrimos el software de programación grafica NI Vision Assistant 8.5, luego adquirimos la imagen por la cámara en Acquire Image(USB), y al guardarlas encontramos que las imágenes son de dimensiones 352 X 288 pixeles y tipo imagen PNG.

El patrón (template) requerido es la tapilla de una de las botellas de la jaba. Si en alguna botella falta una tapilla, la jaba igual se procede a rechazar lo mismo como que si hubiere objetos extraños en las botellas.

3.1.1. Imagen patrón

Para hallar el patrón de la jaba realizamos los siguientes pasos en Vision Assistant 8.5:

- 1.- Abrimos la imagen (open image) que en nuestro caso es una jaba pequeña de 6 botellas.

Ésta imagen es adquirida por la cámara Genius para obtener el modelo de patrón que inspeccionaran las demás botellas.



Figura 3.1.1.1 Abrir imagen Vision Assistant

2.- Nos dirigimos al procesador de imágenes (Process Images) del mismo Vision Assistant, como es una foto a color utilizamos las funciones para color (Processing Functions: Color).

Este grupo de paletas analiza las funciones de imágenes en color.

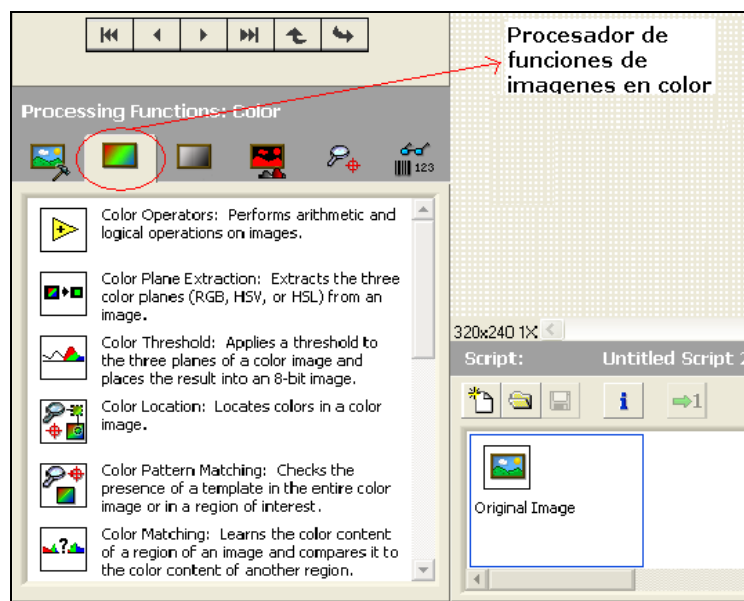


Figura 3.1.1.2. Procesador de funciones para imágenes de color

3.- Seleccionamos el bloque de coincidencia del color de patrón (Color Pattern Matching).

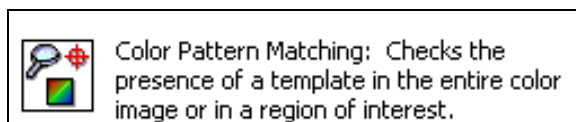


Figura 3.1.1.3 Icono de coincidencia del color de patrón

Éste icono examina la presencia de una plantilla de toda la imagen en color o en una región de interés.

4.- Al seleccionar el Color Pattern Matching creamos el patrón de la imagen (create template) que debe ser menor o igual al área de interés seleccionada que es el de una caja. El patrón es el de una tapilla en la que está un símbolo la letra T.

Luego procedemos a guardar la imagen patrón en la base de datos de la computadora para una posterior evaluación.

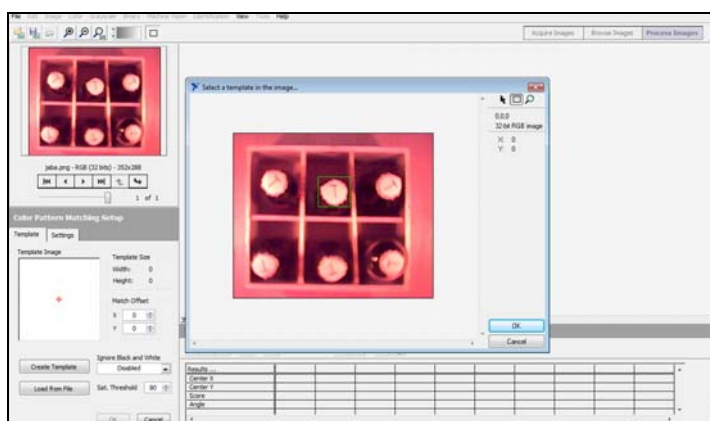


Figura 3.1.1.4 Creación de la imagen patrón de la caja

5.- Luego escogemos la imagen patrón (template) más apropiada, es decir, la que contenga el mayor número de botellas encontradas con el menor porcentaje de aceptación (mínimum score).

Para obtener el mejor modelo damos clic en Configuración (Settings) donde calibramos el número de objetivos a examinar y el porcentaje mínimo en que los objetivos se asemejen al patrón antes adquirido.

Este porcentaje mínimo debe reconocer la mayor cantidad de objetivos (6 botellas). En éste reconocimiento cuenta un papel

importante los parámetros de la sensibilidad del color (color sensitivity) y la estrategia de búsqueda (search strategy) que en las pruebas de evaluación se explicarán más detalladamente.

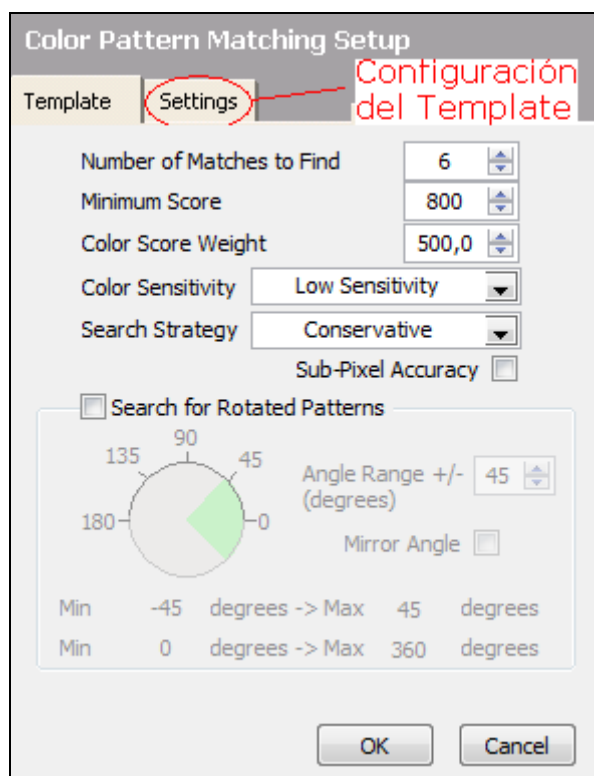


Figura 3.1.1.5 Configuración del Template

Para que el sistema inspector de cajas tenga una programación eficiente debemos elegir el menor número de patrones que reconozca el número total de objetivos antes determinados (6 botellas).

3.1.2. Pruebas de evaluación

Con el patrón creado para análisis tenemos que el número de botellas esperadas es igual al número de botellas encontradas, por consiguiente, la caja se acepta con un porcentaje de aceptación del 80% por consiguiente, puede ser la mejor imagen patrón obtenida para la inspección del sistema inspector de cajas.



Figura 3.1.2.1 Imagen Patrón

Para saber si contamos con la mejor imagen patrón realizamos algunas pruebas de evaluación en Vision Assistant configurando la estrategia de búsqueda, mínimo porcentaje de aceptación y la sensibilidad del color.

Estrategia de Búsqueda (Search strategy)	Sensibilidad de Color (Color Sensitivity)		
Conservativa (Conservative)	Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	6	0
825	6	6	0
850	6	6	0
875	6	6	0

Tabla 3.1.2.1. Prueba de Evaluación con estrategia Conservativa

En la tabla 3.1.2.1, encontramos los valores para la imagen patrón con una estrategia de búsqueda conservativa y una sensibilidad de color baja y con un mínimo puntaje de encuentros cuyos valores van desde un rango de 650 hasta 875. Observamos que en todo ese rango el número de botellas encontradas es igual al número de botellas esperadas dentro del área de interés (jaba) que no existen otras inspecciones fuera de las imágenes parecidas a la imagen patrón (bloques out).

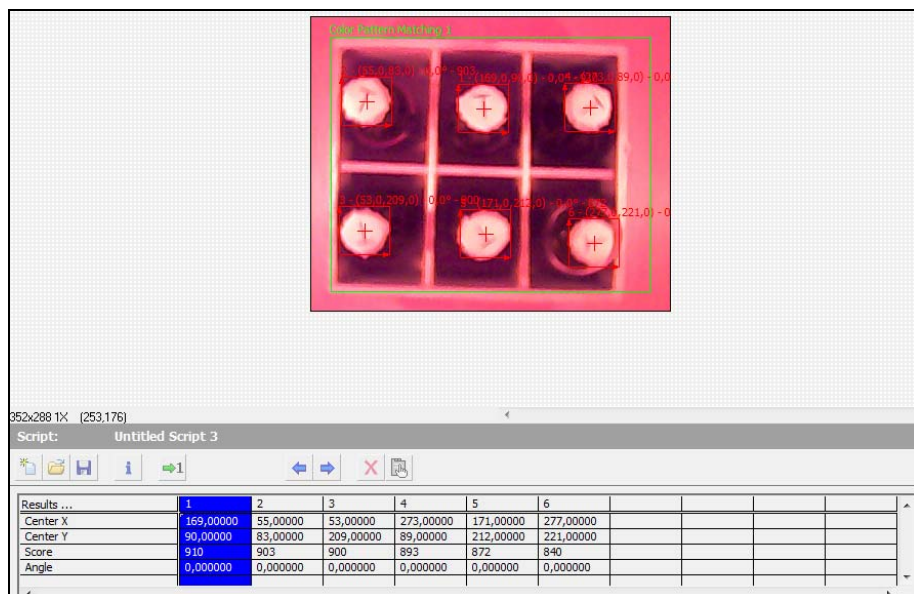


Figura 3.1.2.2 Evaluación con estrategia conservativa

Mientras que en la tabla 3.1.2.2, cuya estrategia de búsqueda es muy agresiva a diferencia de la tabla 1, el número de botellas encontradas nunca es igual al número de botellas esperadas por lo que se proceden a rechazar las cajas.

Si observamos que en el rango del mínimo porcentaje de encuentros de 650 a 750 tenemos algunas irregularidades en la búsqueda por lo que la imagen inspeccionada no es la imagen patrón sino cualquier otra sección del área de interés.

Estrategia de Búsqueda (Search strategy)	Sensibilidad de Color (Color Sensitivity)		
Muy Agresivo (Very Aggressive)	Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Número de Botellas Encontradas (Match)		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	4	2
675	6	4	2
700	6	4	2
725	6	4	1
750	6	4	1
775	6	4	0
800	6	4	0
825	6	4	0
850	6	4	0
875	6	3	0

Tabla 3.1.2.2. Prueba de evaluación con estrategia muy agresivo

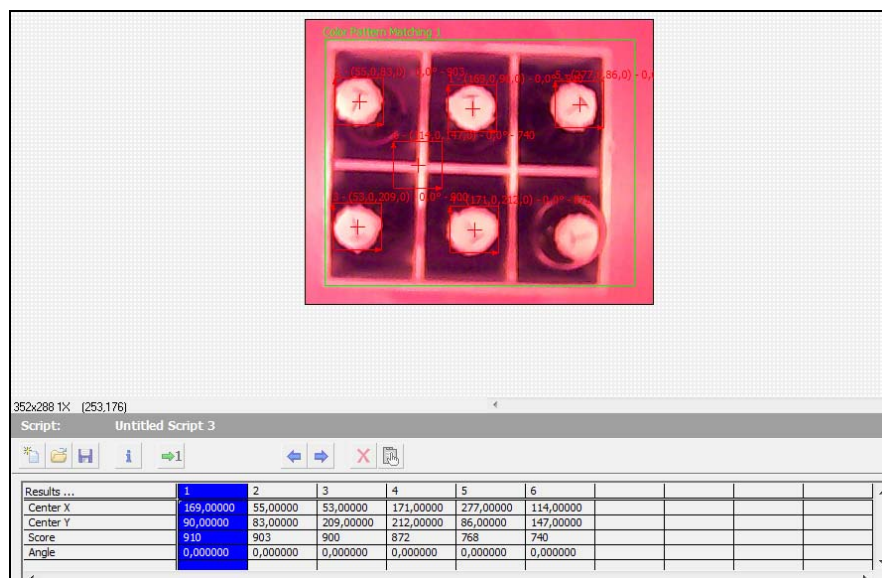


Figura 3.1.2.3 Evaluación con estrategia muy agresiva

De todas las pruebas hechas y adjuntadas en la parte de ANEXOS de la tesis decidimos dejar la configuración de la imagen patrón de la tabla 1 porque en su mínimo porcentaje de encuentros satisface que el número de botellas encontradas es igual al número de botellas esperadas y no tiene inspecciones de la imagen fuera de la imagen patrón (template).

3.1.3. SubVI

Una vez hecho las pruebas de evaluación y seleccionado el template más apropiado para la inspección de la jaba, procedemos a crear un subVI en Labview 8.2 desde Vision Assistant.

Los pasos para crear un subVI del patrón de encuentros a color (color pattern matching) en Vision Assistant son:

- 1.- Una vez aceptado la imagen patrón más apropiado se crea automáticamente un script o código fuente en Vision Assistant que contiene el patrón a color de encuentros.



Figura 3.1.3.1 Script o Código Fuente en Vision Assistant

2.- Procedemos a crear un subVI de Labview 8.2 desde Vision Assistant 8.5. (Labview VI Creation Wizard).

En Tools de la barra de herramientas elegimos create Labview VI por lo que nos aparecerá una ventana en la que se elegirá un archivo o carpeta para guardar el nuevo subVI según la figura 3.1.3.2 que se muestra a continuación.

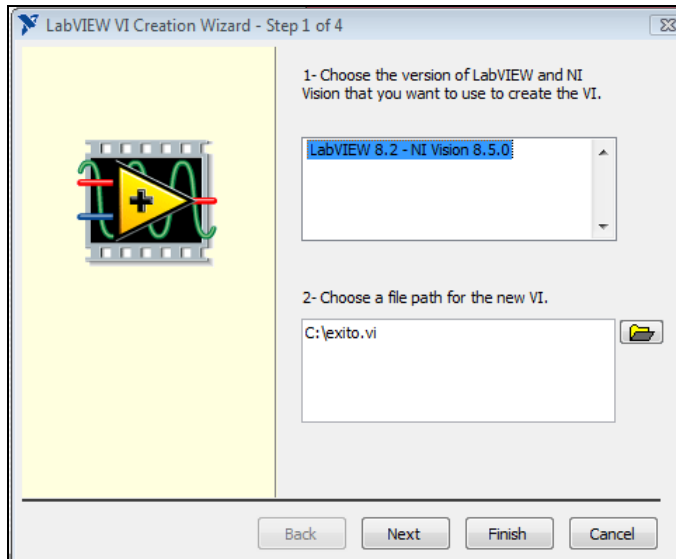


Figura 3.1.3.2 Creación de un sub VI dado el script

3.- Al finalizar el paso anterior automáticamente se crea un sub VI llamado éxito.VI que es el nombre que le dimos en el proceso de creación del mismo.

Creamos el Panel Frontal y el Diagrama de Bloques de la programación gráfica en Labview 8.2 desde Vision Assistant 8.5 para proceder a trabajar en el sistema de inspección de cajas basándonos en éste sub VI.

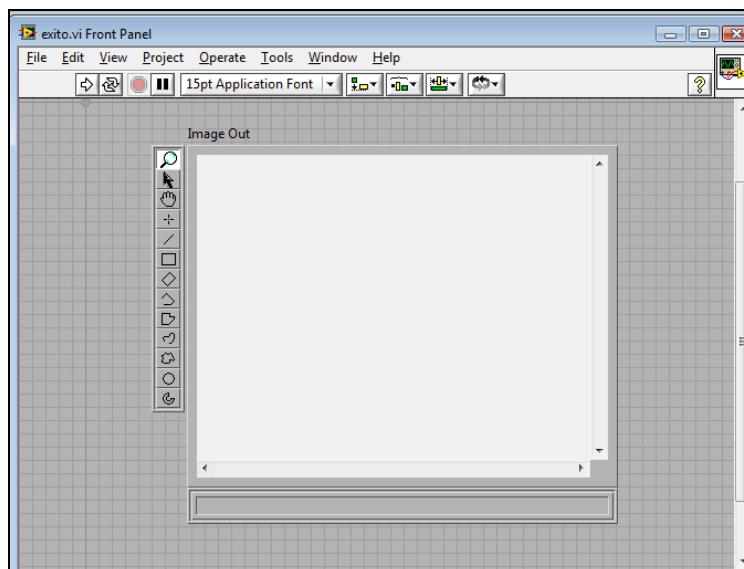


Figura 3.1.3.3 Panel Frontal del éxito.VI

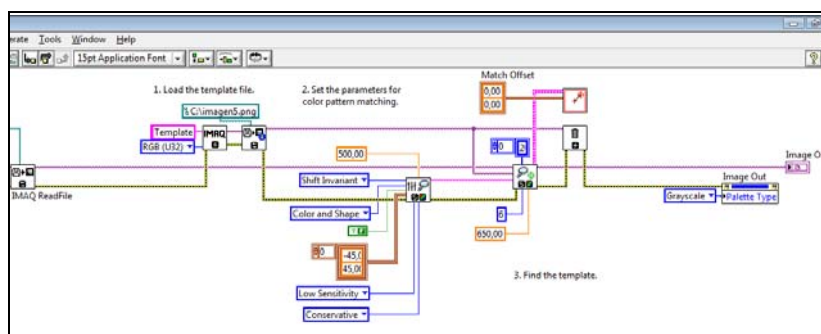


Figura 3.1.3.4 Diagrama de Bloques del éxito.VI

3.2. Elaboración del sub VI del inspector de cajas

La programación de Labview 8.2 se hace menos laboriosa con la complicitad de Vision Assistant 8.5 puesto que disminuye la complejidad en la programación gráfica.

Para programar gráficamente el sub VI del inspector de cajas en Labview debemos conocer las características de los VI a los que Vision Assistant y la necesidad del prototipo inspector de cajas nos conduce a usar y así, comprender mejor su contenido.

3.2.1. Diagrama de bloques

Para un análisis más completo de las características de los VI's en el diagrama de bloques del sistema inspector de cajas en Labview 8.2 se lo divide en 3 partes: La apertura y cierre de la cámara, apertura y cierre del puerto serial y la inspección de la imagen patrón.

La apertura y cierre de la cámara

Debido a que nuestro prototipo necesita una cámara USB para la captura de la caja con sus botellas, debemos conocer los VI a utilizar para la configuración y programación de la cámara USB.

Los VI para la programación de la cámara USB son:

1.- Cámara ON: Procede la cámara abrir para la captura de las imágenes. También guarda un espacio en memoria temporal para una nueva imagen a color. Éste VI a su vez contiene algunos sub VI's que se detallarán en los ANEXOS posteriormente.

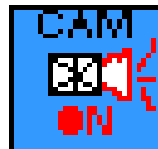


Figura 3.2.1.1 Cámara ON.VI

2.- Imaq USB Grab Acquire: Adquiere una imagen durante una continua adquisición.



Figura 3.2.1.2 Imaq USB Grab Acquire.VI

3.- Imaq USB Close: Cierra una sesión de una cámara USB que ha sido anteriormente abierta.



Figura 3.2.1.3. Imaq USB close.VI

La apertura y cierre del puerto serial

Debido a que nuestro prototipo necesita un puerto serial para simular las entradas y salidas del inspector de cajas, debemos conocer los VI a utilizar para la configuración y programación del puerto serial VISA en el diagrama de bloques.

Los VI para la programación del puerto serial son:

1.- VISA Configure Serial Port: Inicializa el puerto serial especificado por el nombre VISA de la instancia a usar (COM5). Si no contienen valores a la entrada, éstos por default son: Velocidad de transmisión de datos 9600, el número de bits al ingresar datos es 8, la paridad usada para los esquemas de transmisión o recepción es 0 o que no tiene paridad.



Figura 3.2.1.4. VISA Configure Serial Port.VI

2.- VISA Open: Abre una sesión de un dispositivo especificada por el nombre VISA de la instancia a usar (COM 5) y retorna una sesión especificada que puede se usada llamando a cualquier otra operación de ese dispositivo.



Figura 3.2.1.5 Visa Open

3.- VISA Read: Lee un número específico de bytes desde un específico dispositivo o interfaz por el nombre de instancia VISA a usar (COM 5) y retorna los datos en búfer de lectura.

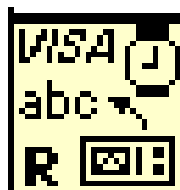


Figura 3.2.1.6. VISA Read

4.- **VISA Write:** Escribe los datos de un búfer de escritura a un dispositivo o interfaz especificado por el nombre de instancia VISA a usar.

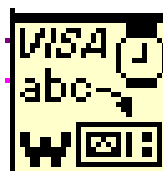


Figura 3.2.1.7. VISA Write

5.- **VISA Close:** Cierra la sesión de un dispositivo o evento de comunicación especificado por el nombre de la instancia VISA a usar.



Figura 3.2.1.8. VISA Close

La inspección de la imagen patrón

El script o código fuente en Vision Assistant, crea los siguientes VI en Labview dentro de un sub VI para su programación gráfica:

1.- Imaq Create: Crea un lugar de memoria temporal para una imagen. Usamos IMAQ Crear, en relación con IMAQ Deseche VI para crear o disponer de Visión NI Labview en imágenes.

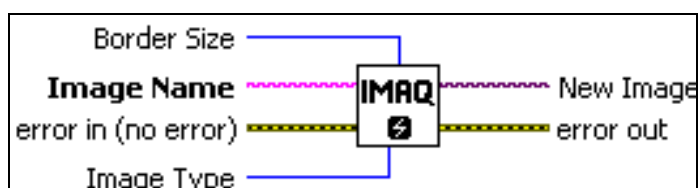


Figura 3.2.1.9. Imaq Create.VI

El **border size** determina la anchura, en píxeles, de la frontera para crear un entorno a una imagen. Crea una frontera al inicio de su aplicación.

El valor por default de la frontera es de 3. Con una frontera de tres píxeles, puede utilizar los núcleos (centro) hasta 7×7 . Si se planea utilizar los núcleos de más de 7×7 en su proceso,

tiene que especificar una mayor frontera al crear su imagen.

Cada **imagen creada** debe tener un nombre único.

El **error** (no error) describe el estado de error antes de que este VI se ejecute. El valor por default es ningún error.

El **tipo de imagen** se puede elegir entre los siguientes valores:

- 1) Escala de grises (U8) (0) 8 bits por píxel.
- 2) Escalo de grises (16) (1) 16 bits por píxel.
- 3) Escala de grises (SGL) (2) 32 bits por píxel.
- 4) Complejo (CSG) (3) 2 × 32 bits por píxel.
- 5) RGB (U32) (4) 32 bits por píxel (rojo, verde, azul, alfa).
- 6) HSL (U32) (5) 32 bits por píxel (matiz, saturación, luminancia, alfa).
- 7) RGB (U64) (6) 64 bits por píxel (rojo, verde, azul, alfa).

2.- Imaq read and image and vision info: Lee un archive de una imagen, incluyendo cualquier información extra de Visión guardada con la imagen.



Figura 3.2.1.10. Imaq read and image and Vision info

File Path es la ruta completa, que incluye el directorio, el nombre del archivo y el driver del archivo para la lectura.

Image es una referencia a la estructura de la imagen para que los datos del archivo de imagen se apliquen.

Imagen Out es una referencia a la imagen de destino. Si la imagen está conectado Dst, Imagen Dst salida es la misma que la imagen Dst. De lo contrario, la imagen Dst Out se refiere a la imagen de referencia de la imagen Src.

3.- Imaq Setup Match Color Pattern: Fija los parámetros que son usados durante el proceso de encuentro de imagen patrón a color.

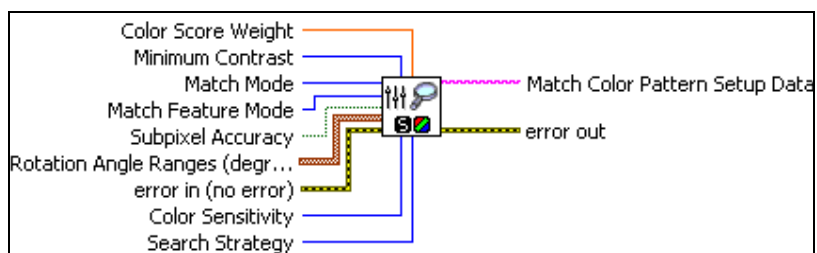


Figura 3.2.1.11. Imaq Setup Match Color Pattern

Color Score Weight determina el porcentaje de contribución del color al final del puntaje de encuentros del patrón. Tiene un valor entre 0 y 1000. Por default es 500 que indica que el puntaje de encuentro usa una igual combinación en color y forma.

Minimum Contrast especifica la mínima diferencia entre el valor máximo de luminosidad y el valor mínimo de luminosidad de los píxeles de la inspección de imagen a color, por default es 0.

Match Mode especifica el modo de no variar para el desplazamiento y la rotación de la imagen patrón en la imagen inspeccionada.

Match Feature Mode especifica las características a usar cuando se inspecciona el patrón a color de la imagen de color.

La característica por default es Color y Forma (Color and Shape).

Color Sensitivity especifica el nivel de la sensibilidad del color de la imagen (bajo, mediano y alto) que afecta el número de características en el color del patrón.

Search Strategy especifica como las características de color son usadas durante una fase de búsqueda. Éstas son conservativas, equilibradas, agresivas y muy agresivas.

4.- Imaq Match Color Pattern: A una imagen de entrada le busca un patrón a color o un template a color.

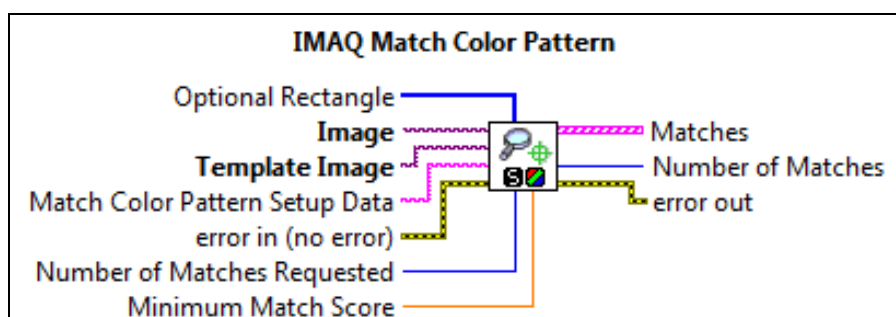


Figura 3.2.1.12. Imaq Match Color Pattern

Optional Rectangle define un período de cuatro elementos matrices que contiene la izquierda, arriba, derecha, abajo y las coordenadas de la región para procesar. El VI se aplica la

operación a toda la imagen si el facultativo rectángulo está vacío o no conectado.

Image es una referencia a la imagen en color en el que se quiere localizar el patrón de imagen en color.

Template Image es una referencia a la imagen patrón de color para el que desea buscar durante la fase de adecuación.

Match color Pattern Setup Data es una cadena que contiene la información obtenida de IMAQ coincidencias de patrones de color. Si esta entrada no está conectada, el VI utiliza parámetros por default durante la fase de adecuación.

Number of Matches Requested es el número esperado de encuentros válidos.

Minimum Match Score es el menor porcentaje de encuentros que puede tener para ser considerado válido. La gama de datos para el porcentaje de encuentros entre 0 y 1000.

Matches es un arreglo grupal de encuentros.

Number of Matches es el número de imágenes patrón de coincidencias encontradas en la inspección de imágenes basada en la configuración de entrada.

5.- IVA Add offset to matches: Es el desplazamiento de encuentros en una posición indicada.

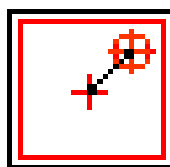


Figura 3.2.1.13. IVA Add offset to matches

6.- Overlay Blister Pack Match Position: Coloca en la posición de encuentros de la imagen patrón un recuadro de inspección.

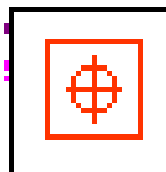


Figura 3.2.1.14. Overlay Blister Pack Match Position.VI

7.- Tick Count: Retorna el valor del tiempo en milisegundo. La base de referencia de tiempo (cero milisegundo) es indefinido.

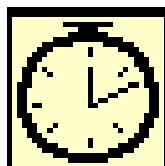


Figura 3.2.1.15. Contador en milisegundos

8.- Imaq Dispose: Destruye una imagen y libera el espacio ocupado en memoria. Este VI es requerido para cada imagen creada en una aplicación para liberar la memoria asignada al imaq create VI. Se ejecuta imaq dispose solo cuando la imagen no es tan larga en una aplicación. Se puede usar Imaq Dispose para cada llamada a Imaq create o solo una vez para todas las imágenes creadas con Imaq create.

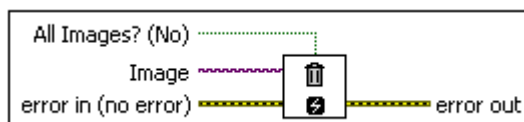


Figura 3.2.1.16. Dispositivo IMAQ

3.2.2. Panel frontal

Los componentes del panel frontal del sistema inspector de cajas en Labview 8.2 son los siguientes:

1.- Nombre Cámara USB: Se introduce el nombre en el que direcciona la cámara USB. El nombre es Dlink DSB-C120.

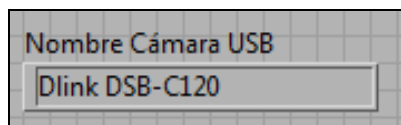


Figura 3.2.2.1. Nombre Cámara USB

2.- Puerto Serial: Se configura la instancia en la laptop en el que se encuentra el convertidor USB-Serial. El nombre configurado es el COM 5.

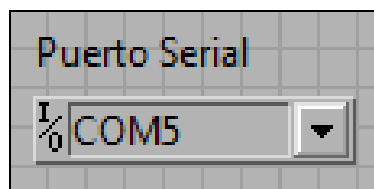


Figura 3.2.2.2. Puerto Serial

3.- Display Estado: Se evalúa en cada parte la que se consideraron más importantes en el diagrama de bloques. Las partes evaluadas son: "Iniciando" al inicio del programa, "Esperando botón" en espera de la botonera, "Procesando Imagen" que carga una nueva imagen, y "Siguiendo Jaba" cuando enviamos la L por el puerto serial para encender el led.

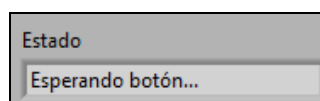


Figura 3.2.2.3. Estado

4.- Imagen Patrón: Aparece el template o patrón de la caja que es una de las tapillas en una pantallita del panel frontal. Éste patrón está guardado en la base de datos de la computadora y tiene como dirección C:\Imagen5.png.



Figura 3.2.2.4. Imagen Patrón

5.- Imagen Principal: Se presenta la imagen principal a la que se le hace la inspección (jaba).

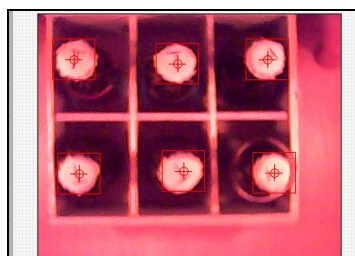


Figura 3.2.2.5. Imagen Principal

6.- Leído desde el puerto serial: Se lee desde el puerto serial la letra B que previamente se programó como entrada del PIC16F628A.

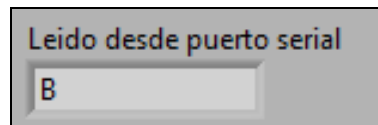


Figura 3.2.2.6. Lectura desde el puerto serial

7.- Tiempo (ms): es el tiempo que transcurre entre la lectura de una caja y la siguiente.

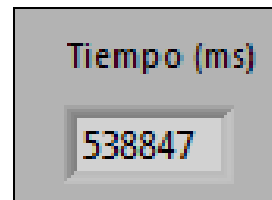


Figura 3.2.2.7. Tiempo (ms)

Una vez terminado el análisis del sub VI tanto en el diagrama de bloques como en el panel frontal procedemos a describir la programación gráfica del VI inspector faltante de botellas en Labview 8.2.

La programación del inspector de cajas comienza con la apertura de la cámara USB y la habilitación del puerto serial VISA.

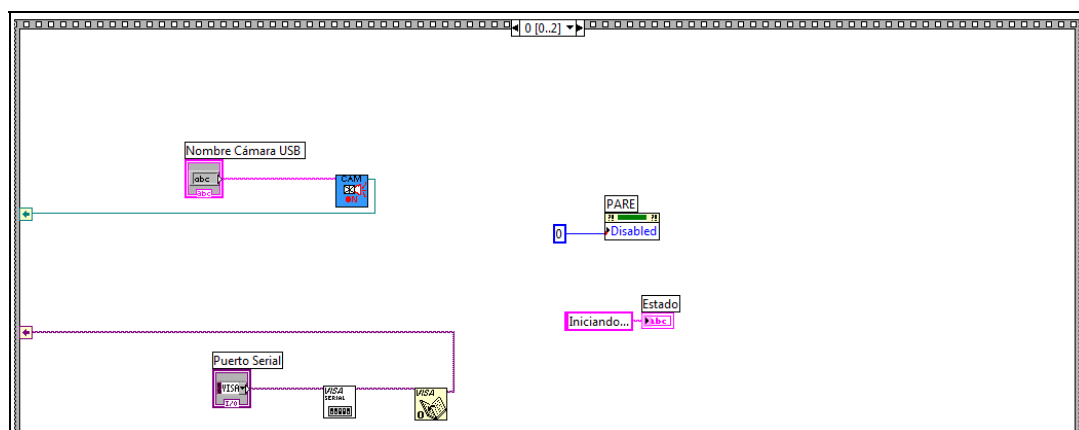


Figura 3.2.2.8. Diagrama de Bloques en el estado Iniciando

En la primera ventana del lazo secuencial de la programación en el diagrama de bloques, la cámara USB espera el nombre Dlink DSB-C120 para prender y el puerto serial VISA espera una instancia COM 5 para habilitarse y abrir consecuentemente.

Estamos en un estado de iniciando, luego se procede a cargar la imagen patrón o template guardado en la memoria del disco duro C como imagen5.png.

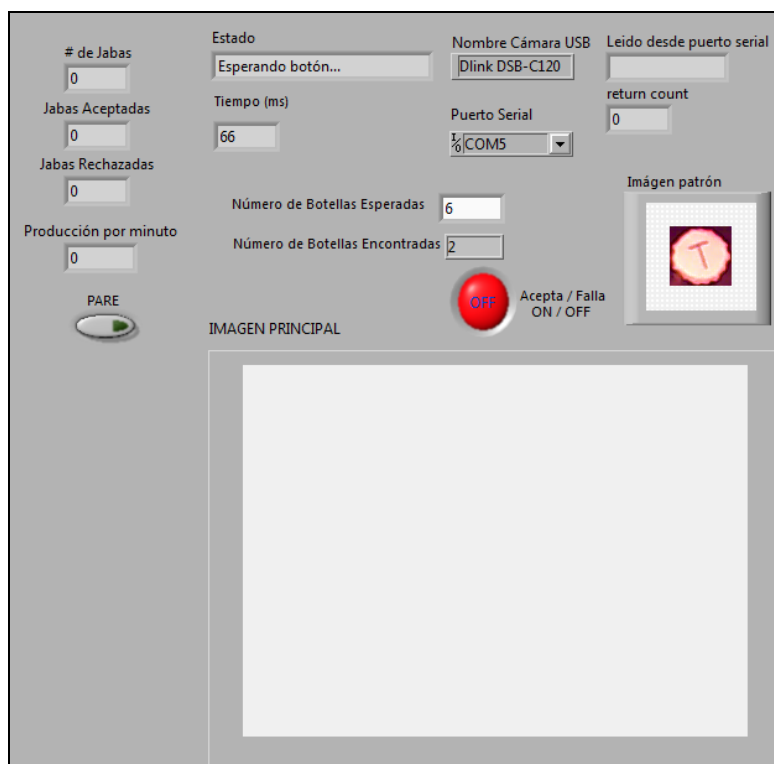


Figura 3.2.2.9 Panel Frontal en el estado “Esperando botón”

Ahora estamos en un estado de “Esperando botón” que espera una botonera como entrada digital de la placa, que al pulsarla envía una señal al software de la computadora para cargar una imagen de una caja (jaba de 6 botellas) para ser inspeccionada. Una vez pulsada la botonera el dato del puerto serial sale del lazo “while” para ser comparada con la letra B como observamos en la figura 3.2.2.10 y dirigirse a un estado de “Dato Recibido”.

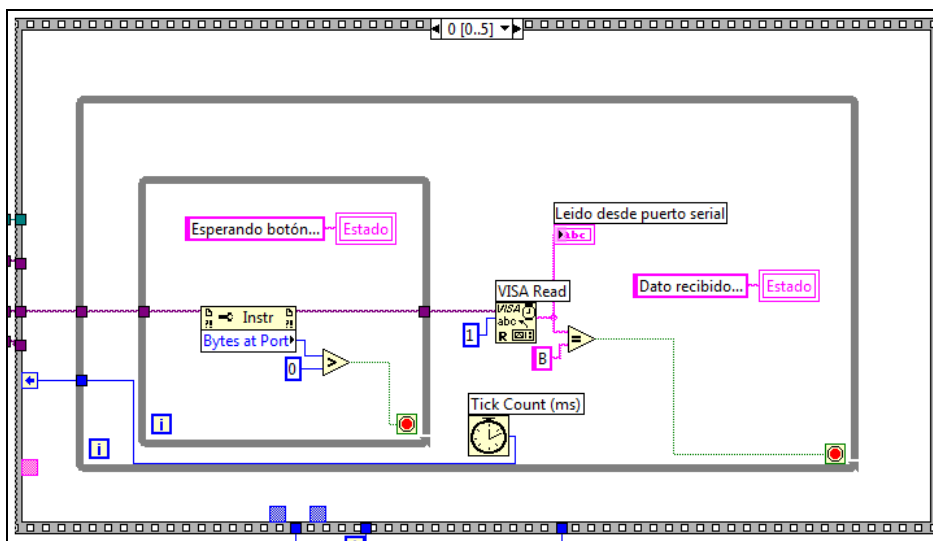


Figura 3.2.2.10. Diagrama de Bloques en estado “Esperando botón”

Una vez confirmado la igualdad de datos nos dirigimos al siguiente lazo secuencial que nos lleva al estado “Procesando imagen”.

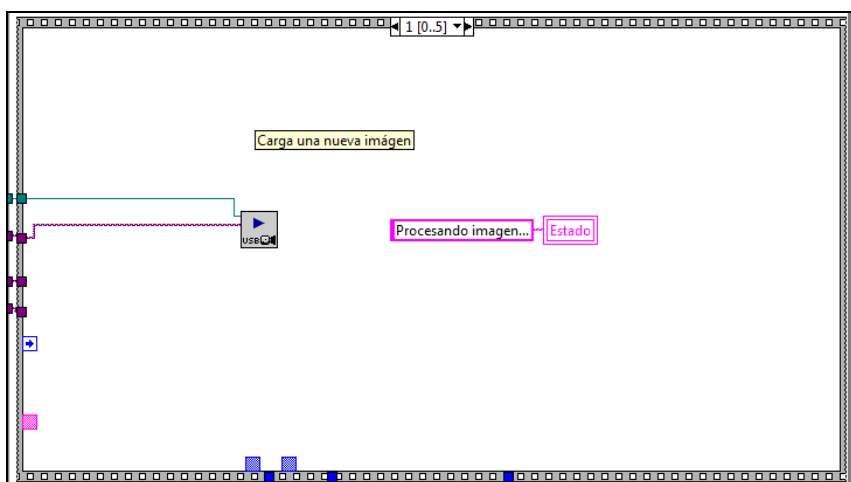


Figura 3.2.2.11. Diagrama de bloques en estado “Procesando imagen”

En éste estado permite procesar la imagen y compararla con la imagen patrón para proceder a su inspección, sabiendo que el total de imágenes parecidas a la imagen patrón en la caja es 6 que son las botellas esperadas.

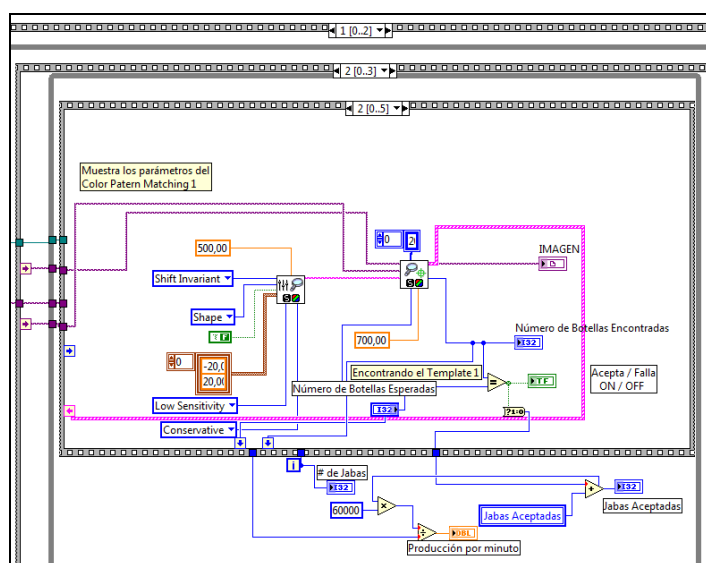


Figura 3.2.2.12. Diagrama de bloques del patrón de encuentros a color

Si el número de botellas esperadas es igual al número de botellas encontradas e igual a 6 botellas, entonces la caja se acepta por lo que habilitará una señal led de color verde que a su vez se contará para la producción diaria.

La producción diaria viene dado en número de cajas aceptadas por minutos. En el diagrama de bloques el tiempo viene dado en

milisegundos por lo que hubo la necesidad de multiplicarla por 60000 para convertirla a minutos.



Figura 3.2.2.13. Panel frontal de una caja aceptada

Mientras que si el número de botellas encontradas es menor al número de botellas esperadas, entonces la caja se rechaza por lo que habilitará una señal led de color rojo en el Labview la cual no se contará para la producción diaria.

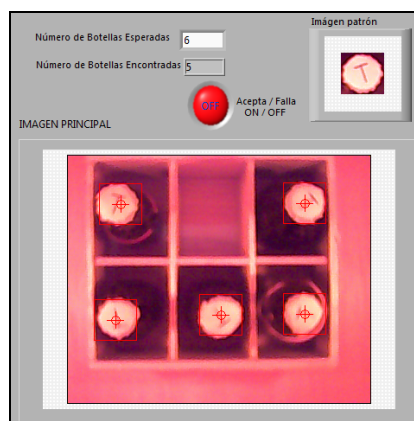


Figura 3.2.2.14. Panel frontal de una caja rechazada

En el siguiente estado “Siguiete jaba” del diagrama de bloques el dato recibido en VISA write debe ser igual a la letra “L” que es enviada cuando la jaba se rechaza, enviando una señal por el puerto serial a la placa donde se habilitará una señal de salida digital representado por un led rojo común.

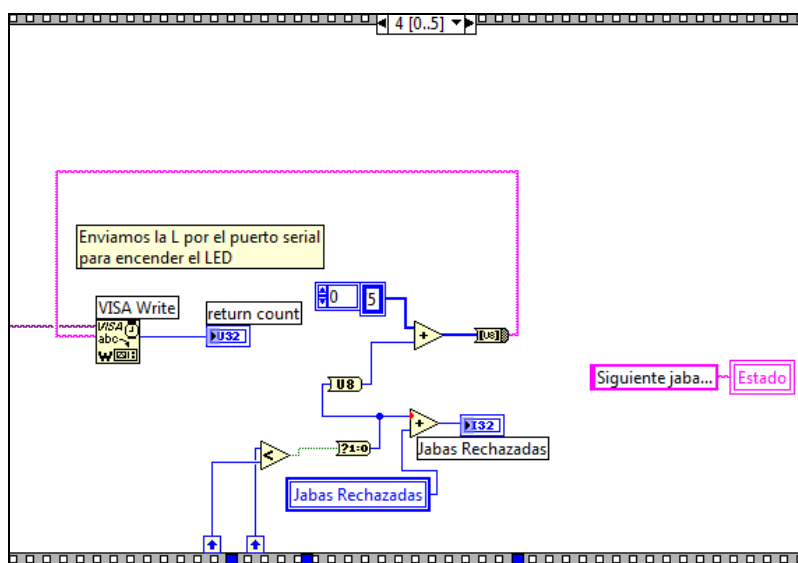


Figura 3.2.2.15. Diagrama de bloque del estado “Siguiete jaba”

3.3. Pruebas de Inspección

Las pruebas de inspección se realizaron en el diagrama de bloques de patrón de encuentros a color (match color pattern) en los que evaluaremos cual es el mejor patrón encontrado en la inspección de la caja aceptada en Labview.

3.3.1. Variando el punto mínimo de encuentro

El punto mínimo de encuentro (mínimum match score) lo variamos del IMAQ match color pattern en un rango de 650 a 875 donde 1000 es igual a la imagen patrón que la elegimos como la más óptima para la inspección en Vision Assistant.

Recalamos que solo se va a variar el punto mínimo de encuentro dejando constante las demás variables tanto los atributos en forma (shape) como la estrategia de búsqueda en conservativa (conservative).

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color SensitiVlty) Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	5	0
825	6	4	0
850	6	3	0
875	6	1	0

Tabla 3.3.1.1. Variando el Mínimo Punto de encuentro

Cuando el mínimo punto de encuentro es 800 observamos que el número de botellas inspeccionadas es 5, a pesar que se encuentran en la imagen principal las 6 botellas como observamos en la figura 3.3.1.1, por consiguiente no es igual al número de botellas esperadas. Lo mismo sucede con los valores superiores a 800 del mínimo punto de encuentro.

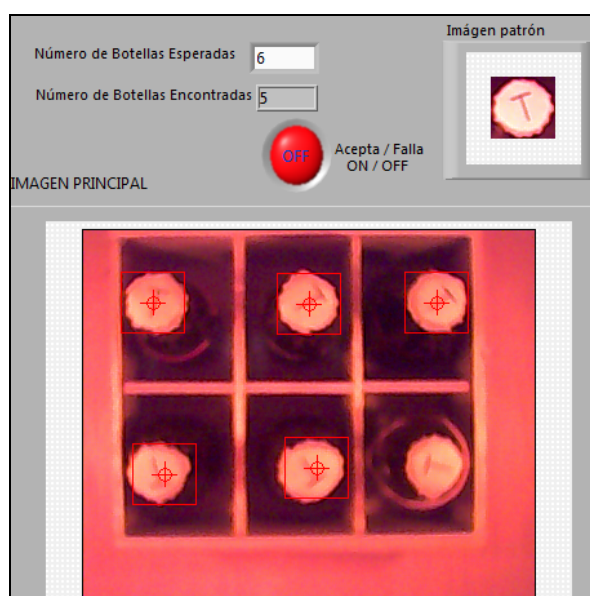


Figura 3.3.1.1. Prueba de Inspección con mínimo punto de encuentro 800.

3.3.2. Variando los atributos

Los atributos (match feature mode) lo variamos del IMAQ Setup match color pattern en forma, color y forma/color donde la forma

es el atributo que se eligió para la imagen patrón en la inspección de Vision Assistant.

Recalcamos que solo se va a variar los atributos dejando constante las demás variables tanto el punto mínimo de encuentro en 700 como la estrategia de búsqueda en conservativa (conservative).

Cuando el atributo es solo color, forma y color/forma observamos que el número de botellas inspeccionadas es igual al número de botellas esperadas.

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		Mínimo Punto de Encuentro = 700
Atributos	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
Forma	6	6	0
Color	6	6	0
Forma y Color	6	6	0

Tabla 3.3.1.2. Variando los atributos

Si variamos los atributos de color observamos que igual el número de botellas encontradas es igual al número de botellas

esperadas por lo que el mínimo punto de encuentro llega hasta 850.

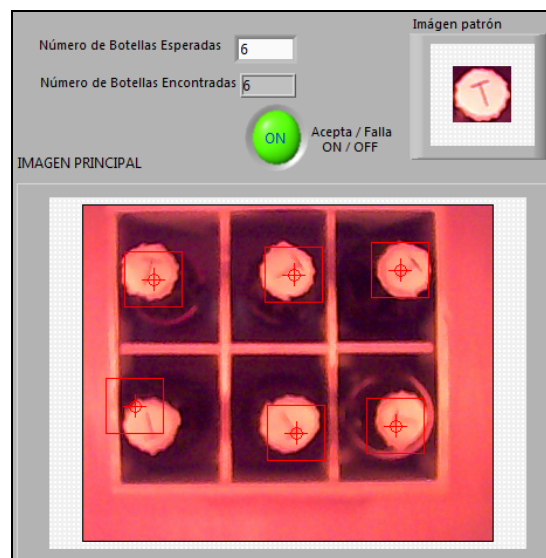


Figura 3.3.2.1. Prueba de inspección con atributo color.

Pero si variamos los atributos de color con el punto mínimo de encuentro desde 875 observamos que el número de botellas encontradas es menor al número de botellas esperadas, como lo muestra la siguiente figura 3.3.2.2

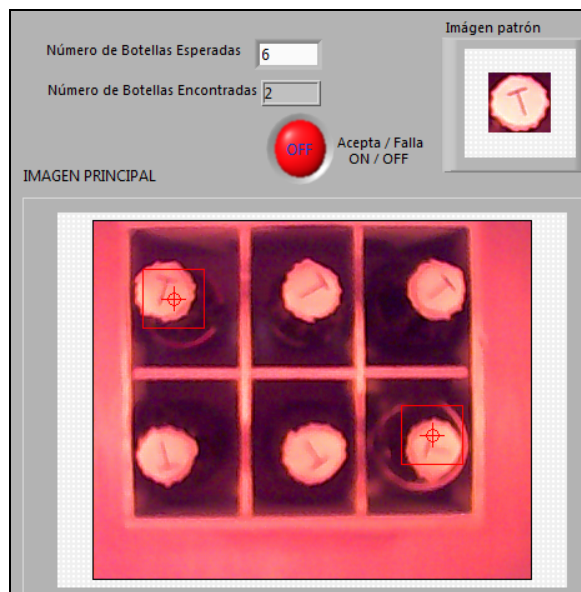


Figura 3.3.2.2. Atributo color desde punto mínimo de encuentro 875

3.3.3. Variando los estrategia de búsqueda

La estrategia de búsqueda (Search strategy) lo variamos del IMAQ Setup match color pattern en conservativa (conservative), equilibrada (balance), agresiva (aggressive), muy agresiva (very aggressive) donde la conservativa es la estrategia de búsqueda que se eligió para la imagen patrón en la inspección de Vision Assistant.

Recalcamos que solo se va a variar la estrategia de búsqueda dejando constante las demás variables tanto el punto mínimo de encuentro en 700 como los atributos en color y forma (color and shape).

Atributos Color y Forma	Sensibilidad de Color (Color Sensitivity) Baja (Low)		Mínimo Punto de Encuentro = 700
Estrategia de Búsqueda (Search strategy)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
Conservativa (Conservative)	6	6	0
Equilibrada (Balance)	6	6	0
Agresiva (Aggressive)	6	6	0
Muy agresiva (Very aggressive)	6	6	0

Tabla 3.3.1.2. Variando la estrategia de búsqueda

Si variamos la estrategia de búsqueda de conservativa a muy agresiva el número de botellas esperadas es igual al número de botellas encontradas porque el mínimo punto de encuentro llega hasta 825.

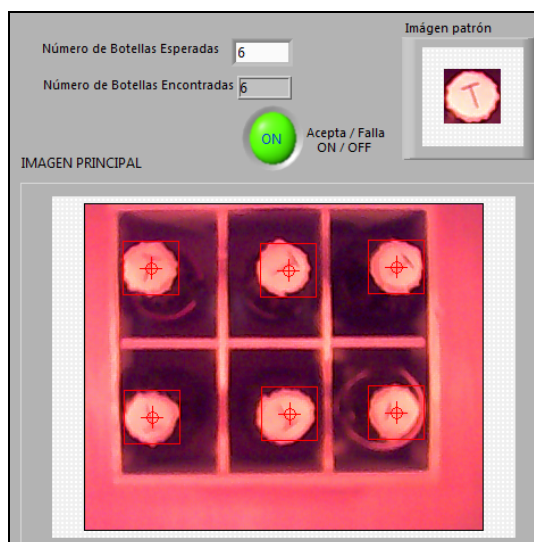


Figura 3.3.3.1. Prueba de inspección con estrategia de búsqueda muy agresiva

Pero si variamos la estrategia de búsqueda muy agresiva con el punto mínimo de encuentro desde 850 observamos que el número de botellas encontradas es menor al número de botellas esperadas, como lo muestra la siguiente figura 3.3.3.2

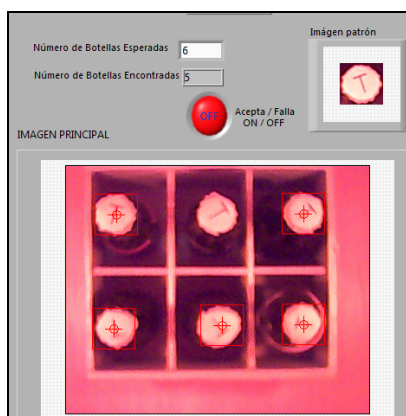


Figura 3.3.3.2. Estrategia de búsqueda muy agresiva desde punto mínimo de encuentro 850.

3.4. Análisis de resultados

Analizando las tablas en las pruebas de inspección concluimos lo siguiente:

- El mínimo punto de encuentro es la característica más prioritaria para una excelente inspección por lo que, mientras mas se acerca al valor de 1000 van reduciendo el número de botellas inspeccionadas, tal es el caso de 875 se redujeron a 1 el número de botellas encontradas. También observamos que

cuando el mínimo punto de encuentro llega a 800 comienza a reducir el número de botellas encontradas a 5.

- Los atributos juegan un papel importante puesto que cuando está en atributo forma con el mínimo punto de encuentro hasta 775 y cuando está en atributo color con el mínimo punto de encuentro hasta 850 el número de las botellas encontradas es igual al número de botellas esperadas. Si es superior a los dos mínimos puntos de encuentros anteriores entonces el número de botellas encontradas es menor al número de botellas esperadas. Debido a lo anteriormente dicho, el atributo más óptimo para la inspección es la unión del color y la forma.
- La estrategia de búsqueda también juega un papel importante puesto que en estrategia de búsqueda conservativa y muy agresiva con el mínimo punto de encuentro hasta 825 el número de botellas encontradas es igual al número de botellas esperadas. Si es superior al mínimo punto de encuentro anterior entonces el número de botellas encontradas es menor al número de botellas esperadas; esto sucede con las demás características de la estrategia de búsqueda siempre y cuando el atributo sea de forma y color.

- Los bloques out, las inspecciones de las imágenes parecidas a la imagen patrón, son cero debido a las acertadas pruebas de evaluación en Vision Assistant.

CAPÍTULO 4

4. ANALISIS DE COSTOS DEL SISTEMA INPSECTOR DE CAJAS

En este capítulo de análisis se considera de manera detallada los equipos y elementos que se utilizan para realizar el proyecto, además de las interfaces requeridas por cada uno de ellos.

La metodología usada para obtener las proyecciones económicas sobre el desarrollo del prototipo inspector de cajas está basada en diferentes análisis técnicos económicos, los cuales se describen en los próximos párrafos.

4.1. Costos del hardware

En la instalación de los instrumentos de campo se hace uso de diferentes conectores y medios de comunicación, así como materiales de protección de los mismos.

Puesto que nuestro proyecto es un prototipo nosotros hemos bajado costos, aquí detallaremos nuestro presupuesto en dólares norteamericanos de los elementos para el hardware del mismo.

Para detallar mejor los costos totales del hardware se lo ha dividido en tres secciones:

- Costos de la planta
- Costos de la placa
- Costos de equipo

Para el análisis correspondiente de los **costos de la planta**, implementamos un pequeño prototipo el cual contiene los elementos detallados a continuación.

Ítem	Cantidad	Descripción	Valor Unitario	Valor Total
1	1	jaba	1,5	1,5
2	1	Cámara USB	17	17
3	2	Molde cerámico	0,3	0,6
4	2	focos de 15w	0,3	0,6
5	2	enchufes	0,3	0,6
6	2	Cable AWG 2*18	0,3	0,6
7	1	Cinta doble empaque (metro)	2,5	2,5
8	2	Frasco de silicona	0,5	1
9	1	Soporte de tapa	0,3	0,3
			Total *	24,7

Tabla 4.1.1 Costos de la planta

A continuación detallaremos todos los **costos de** los componentes que se usaron para la implementación de **la placa**.

Ítem	Cantidad	Descripción	Valor Unitario	Valor Total
1	1	Resistencia de 160 1/4w	0,02	0,02
2	1	Resistencia de 0,33k 1/4w	0,02	0,02
3	1	Resistencia de 4,7k 1/4w	0,02	0,02
4	1	Resistencia de 1k 1w	0,05	0,05
5	4	Capacitor electrolítico de 10 uf	0,05	0,2

6	2	zócalos integrados	0,4	0,8
7	1	Bornera doble	0,2	0,2
8	1	PIC 16F628A	4,4	4,4
9	1	Botonera	0,3	0,3
10	1	Socket hembra	0,5	0,5
11	1	MAX 232CPE	2,2	2,2
12	1	Baquelita	6	6
13	1	Diodo Led	0,1	0,1
			Total *	14,81

Tabla 4.1.2 Costos de la placa electrónica

En la siguiente tabla detalla todos los **costos de los equipos** que se utilizaron en la elaboración del prototipo.

Ítem	Cantidad	Descripción	Valor Unitario	Valor Total
1	1	Multímetro	4,5	4,5
2	1	Cautín	2,2	2,2
3	1	Pinzas cortacables	1,5	1,5
4	1	Estilete	0,3	0,3
			Total *	8,5

Tabla 4.1.3 Costos de equipos utilizados

Los costos totales del hardware del prototipo inspector faltante de botellas en cajas se mencionan a continuación:

Ítem	Cantidad	Descripción	Valor Unitario	Valor Total
1	1	Cable UTP hembra-hembra	0,3	0,3
2	2	Caja para socket RS-232	0,5	1
3	1	Planta	24,7	24,7
4	1	Placa	14,81	14,81
5	1	Cargador de 5 voltios	2,5	2,5
6	1	Cable convertidor USB-RS232	17	17
7	1	Laptop	920	920
8	1	Equipos utilizados	8,5	8,5
			Total *	988,81

Tabla 4.1.4 Costo total del hardware

4.2. Costos del software

Para el desarrollo del inspector faltante de botellas en cajas se necesitan de los siguientes software de aplicación cuyos costos están detallados a continuación.

Ítem	Cantidad	Descripción	Valor Unitario	Valor Total
1	1	Sistema de desarrollo Labview	4730	4730
2	1	Mikro Basic	99	99
3	1	Vision acquisition software	460	460
			Total	5289

Tabla 4.2.1 Costos de Software usados

4.3. Análisis

Observando los datos de los costos totales del hardware que es \$988,81 y los costos totales del software que es \$5289, el costo total del proyecto es:

Costo total del hardware \$ 988,81

Costo total del software \$5289,00

Total **\$6277,81**

Se debe considerar además de los gastos monetarios, el tiempo invertido en la construcción y diseño, los cuales se detallan a continuación:

Detalle de trabajo	Tiempo (horas)
Cambio del sistema operativo de laptop	10
Diseño de la placa electrónica	15
Ensamble de la Planta	20
Programación del PIC16F628A	10
Diseño del programa en Labview	210
Diseño del template en Vision Assistant	15

Pruebas de evaluación en Vision Assistant	15
Pruebas de inspección en Labview	30
Total de horas	325

Tabla 4.3.1 Total de horas trabajadas

Considerando que las horas laborables en este proyecto fueron 325 y las horas laborables en un mes fueron 160 se podrían decir que este prototipo equivale a un poco más de 2 meses de trabajo.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

La construcción de un prototipo inspector de cajas a pequeña escala es muy útil para mejorar la comprensión del mismo a gran escala dentro de los parámetros establecidos, esto nos ayuda a investigar diferentes tipos de proyectos de Visión con fines didácticos en laboratorios de la Espol.

Nuestro prototipo cumple las expectativas del proceso de Visión con Labview y otros softwares de simulación para su desarrollo. El PIC16F628A fue programado de tal manera que en conjunto con la programación con Labview pone en comunicación el hardware con el software para interactuar convenientemente; tal es el caso en el desarrollo de la tesis, cuando la caja se rechaza por incompleta observamos un led rojo prenderse tanto en el software como en el hardware.

Para el reconocimiento de imágenes en nuestro prototipo la luminosidad constante es un factor primordial debido a que las imágenes capturadas deben tener la misma luminosidad que la o las imágenes guardadas en la base de datos del computador como imagen patrón para proceder a una inspección

óptima; para el cual desarrollamos una caja de madera que encierra luz artificial continua y debidamente distribuida, colocando focos estratégicamente ubicados, para la captura de las imágenes de las botellas en la caja.

Con las pruebas de evaluación y de inspección, de Vision Assistant y Labview respectivamente concluimos que la mejor imagen patrón es aquella que tiene el mayor porcentaje de similitud en color, forma y brillo. Nuestra mejor imagen patrón resultó con un mínimo punto de encuentro de 710 que representa el 71% de similitud entre la imagen patrón y la imagen principal inspeccionada; y con una estrategia de búsqueda conservativa y atributo en color y forma.

Recomendaciones:

El inspector de cajas es recomendable para procesos en los cuales no se requiera extrema rapidez en la captura de imágenes, ya que el diseño no lo permite; y si aún se pondría en marcha, lo que obtendremos en la mayoría de los casos son cajas rechazadas. Para proyectos que requieran una mayor producción o jabas aceptadas en menor tiempo se recomienda una cámara con mayor velocidad de captura de imágenes.

Para futuros proyectos, se le puede añadir un módulo mecánico para transportar las jabas como una banda electromecánica que complemente al

inspector de cajas, de ésta manera capturaremos imágenes con una jaba en movimiento.

Para mejorar la coincidencia de la imagen patrón con las imágenes capturadas de la cámara recomendamos tomar un segundo patrón, en el cual se identifiquen la mayor cantidad de botellas, así, con la suma de dos patrones obtenemos la mayor cantidad de botellas; pues nuestro prototipo con las pruebas respectivas solo necesitó una imagen patrón.

BIBLIOGRAFIA

- (1) <http://www.coca-cola.com.ec>
Proceso en una línea de producción en una empresa embotelladora.
- (2) <http://www.cervecerianacional.com.ec/Desktop.aspx?Id=247&e=20>
Formatos múltiples de botellas en vidrio.
- (3) <http://www.dlink.com>
Configuración de la cámara USB.
- (4) www.vhtm.com/software.htm
Tipos de software de Visión.
- (5) www.vhtm.com/c-cambci_4
Tipos de cámaras.
- (6) www.monografias.com/trabajo11/usbmem
Comunicación en puerto serial
- (7) Mikrobasic Help, usart library, hw connection.
Conexión del circuito de control con DB9 y PIC16F628A.
- (8) Hoja de datos de la familia MAXIM, Multichannel & Drivers/Receivers, MAX232, pp, 17.
- (9) Hoja de datos de la familia PIC16F627A/628A/648A, www.microchip.com, pp., 10.
- (10) <http://www.ni.com/labVlew/esa/>
Información General sobre Labview.

- (11) http://cnx.org/content/m14767/latest/lvt_subVI.html
Creación de un subVI en Labview.
- (12) http://www.allianceVIsion.com/us/products/software_ni/VIsion_assistant.htm
Información General sobre Vision Assistant.
- (13) National Instruments, "Labview 8.2".
- (14) National Instruments, "NI-IMAQ for USB Cameras User Manual", 2000.
- (15) National Instruments, "NI-IMAQ Function Reference Manual", 2000.
- (16) National Instruments, "IMAQ Vision for User Manual", 2000.
- (17) National Instruments, "IMAQ Vision Concepts Manual", 2000.
- (18) National Instruments, "IMAQ Vision Builder Tutorial", 2000.
- (19) National Instruments, "Vision Assistant 8.5".
- (20) <http://www.wordreference.com>
Definiciones, sinónimos y traducción de palabras.

ANEXOS

A. VI's de Labview

- Encendido de cámara (Cámara ON)

Panel Frontal

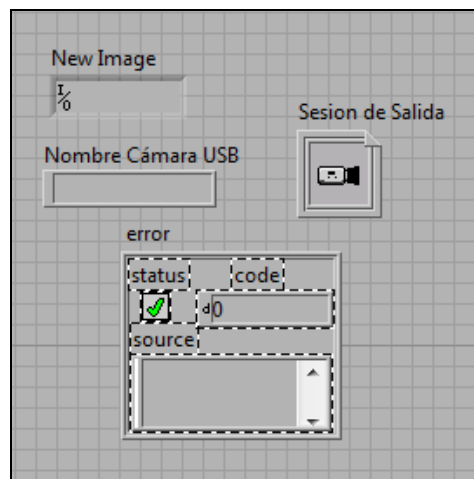
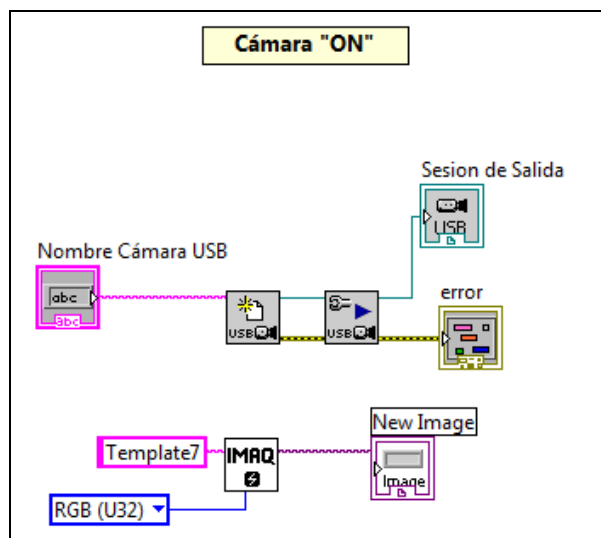


Diagrama de Bloques



- **Adquisición de Imágenes (IMAQ USB Grab Acquire)**

Panel Frontal

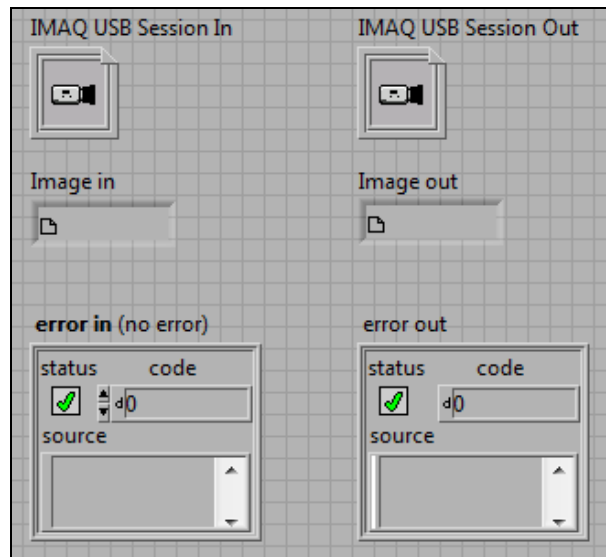
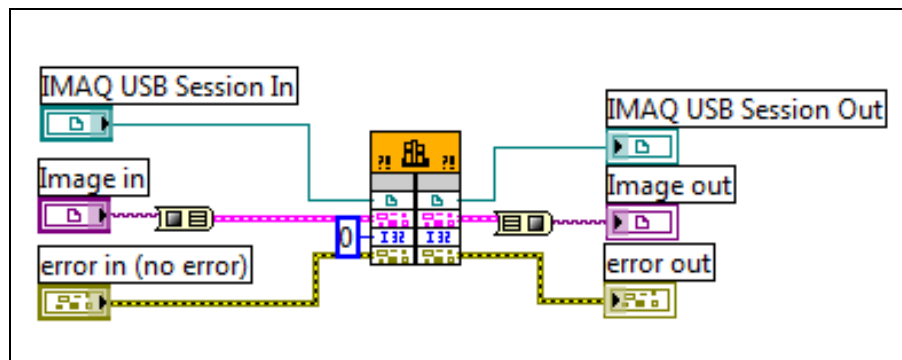


Diagrama de Bloques



- Cierre de la cámara (IMAQ USB Close)

Panel Frontal

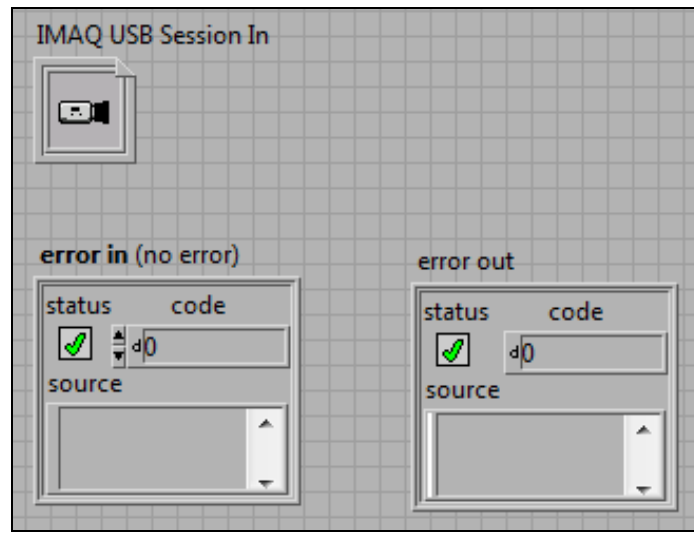
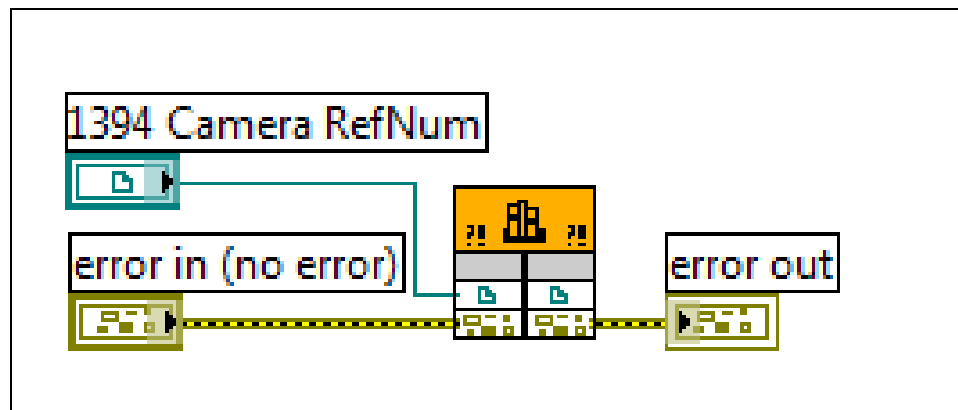


Diagrama de Bloques



- **Configuración de Puerto Serial (VISA)**

Panel Frontal

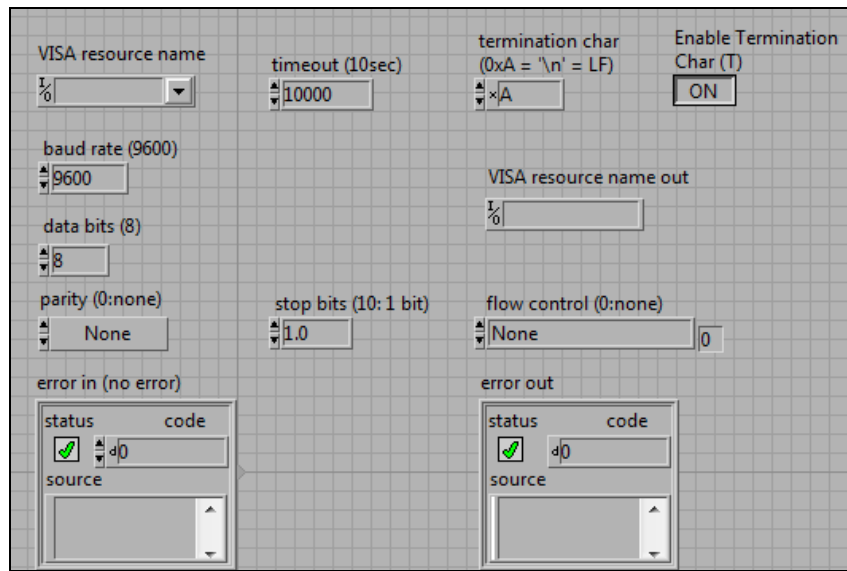
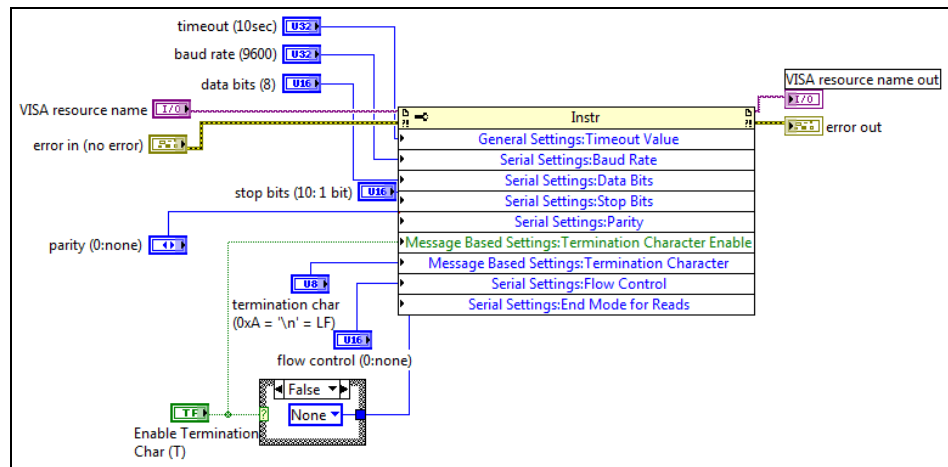


Diagrama de Bloques



- Creación de imagen (IMAQ Create)

Panel Frontal

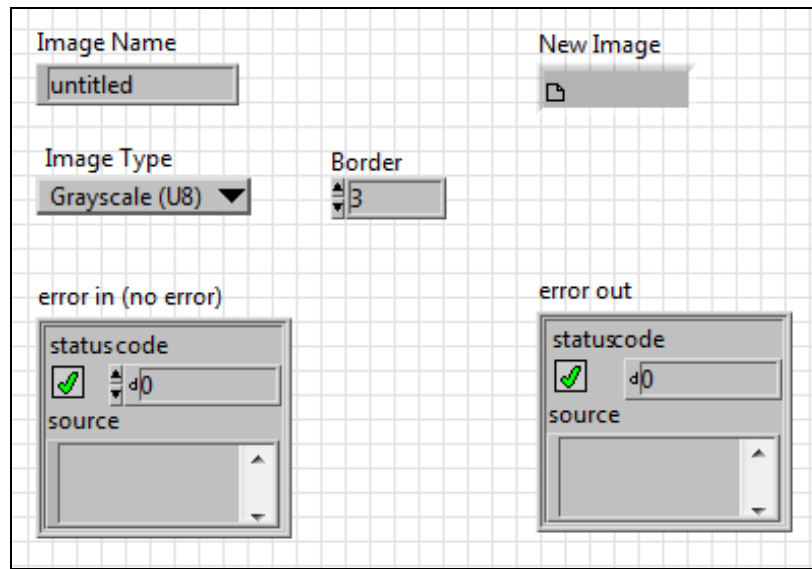
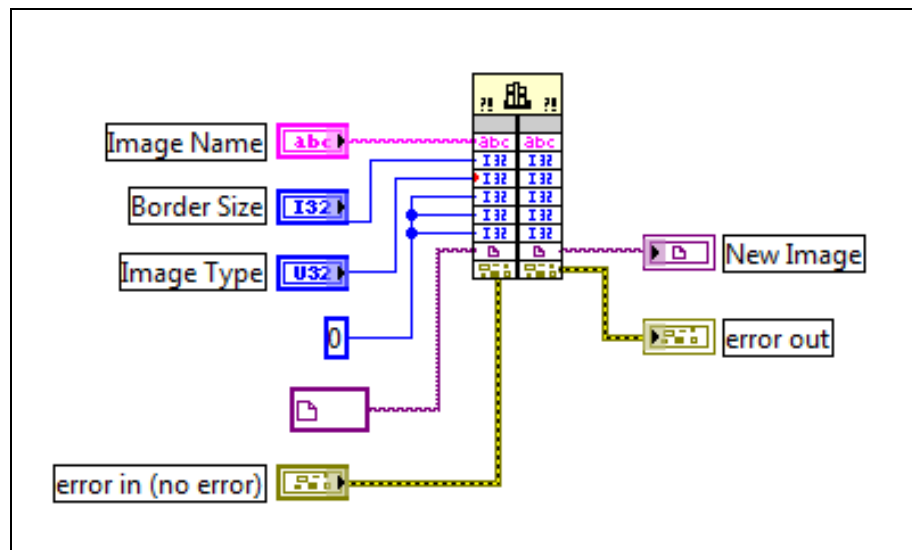


Diagrama de Bloques



- **Lectura e información de la imagen (IMAQ Read Image and Vision Info)**

Panel Frontal

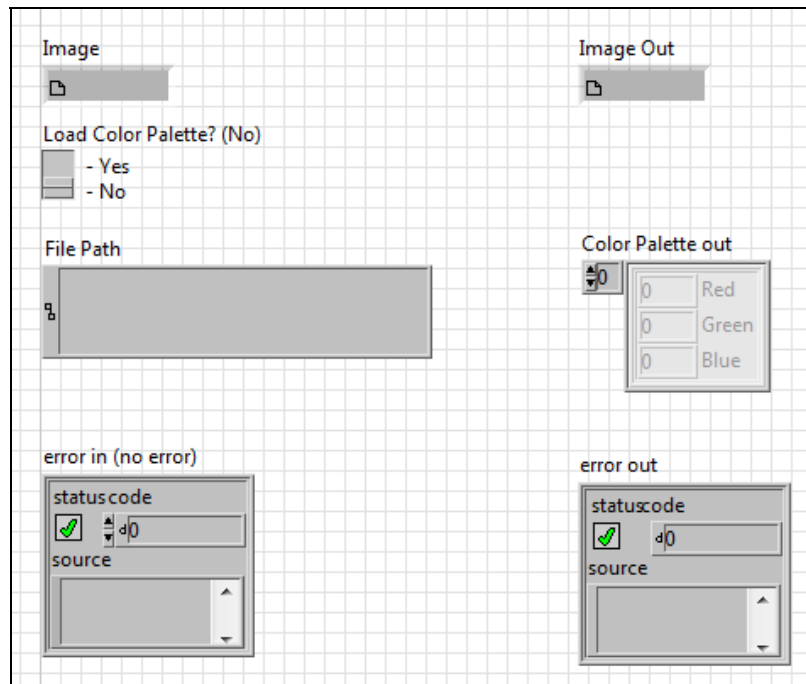
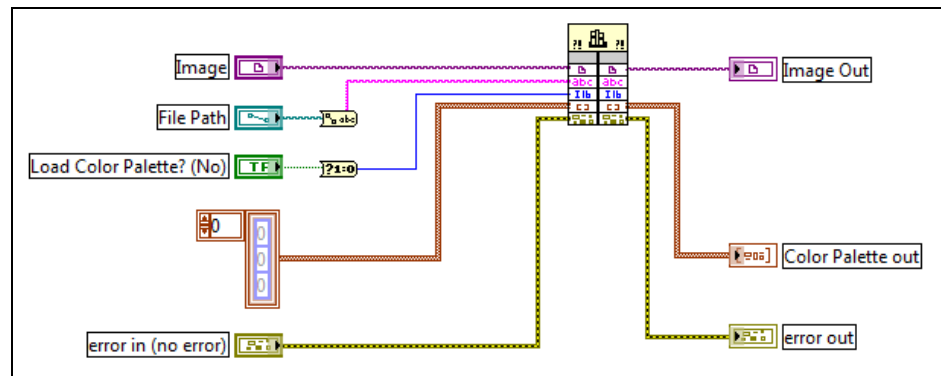


Diagrama de Bloques



- Montaje del patrón color de imágenes (IMAQ Setup Match Color Pattern)

Panel Frontal

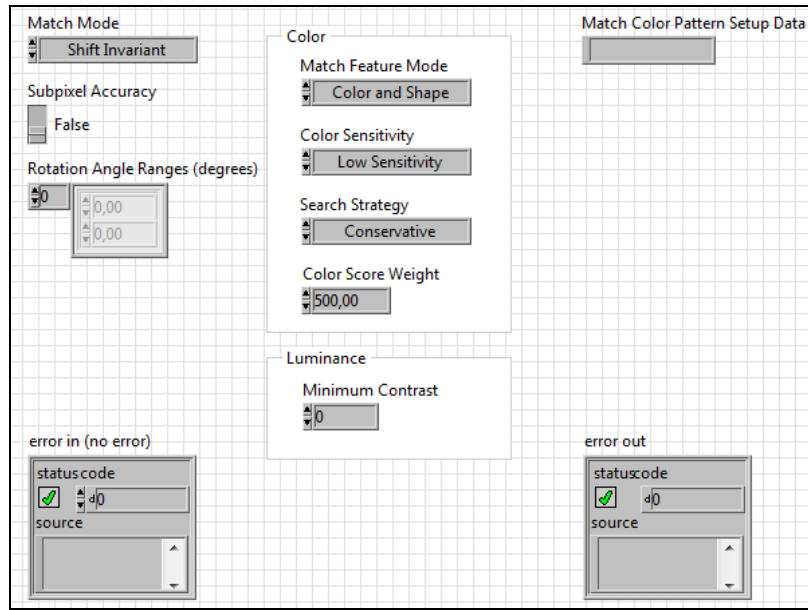
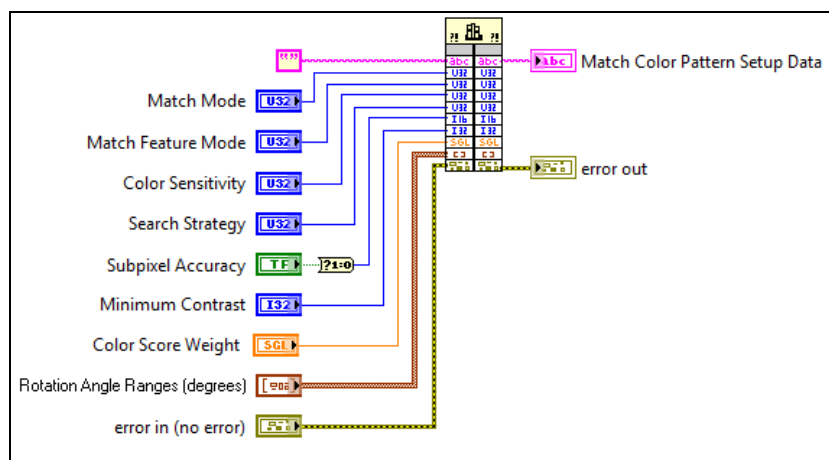


Diagrama de Bloques



- Patrón color de imágenes (IMAQ Match Color Pattern)

Panel Frontal

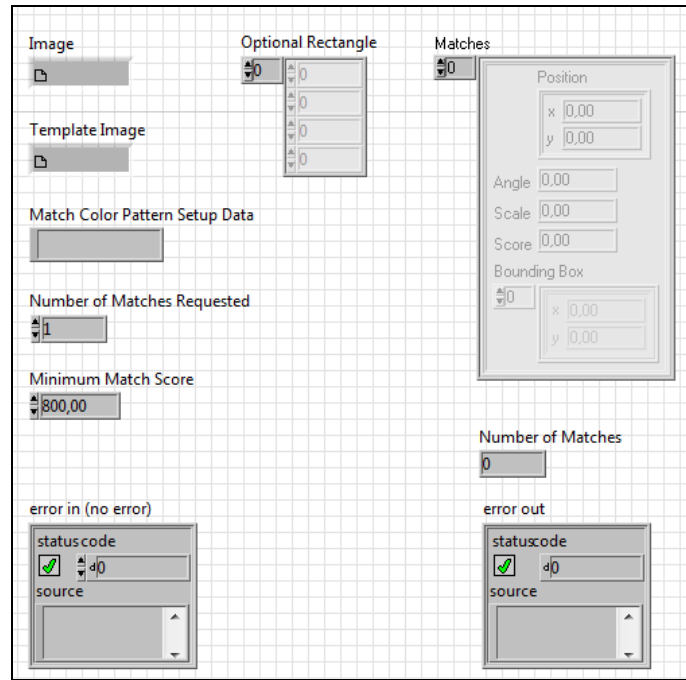
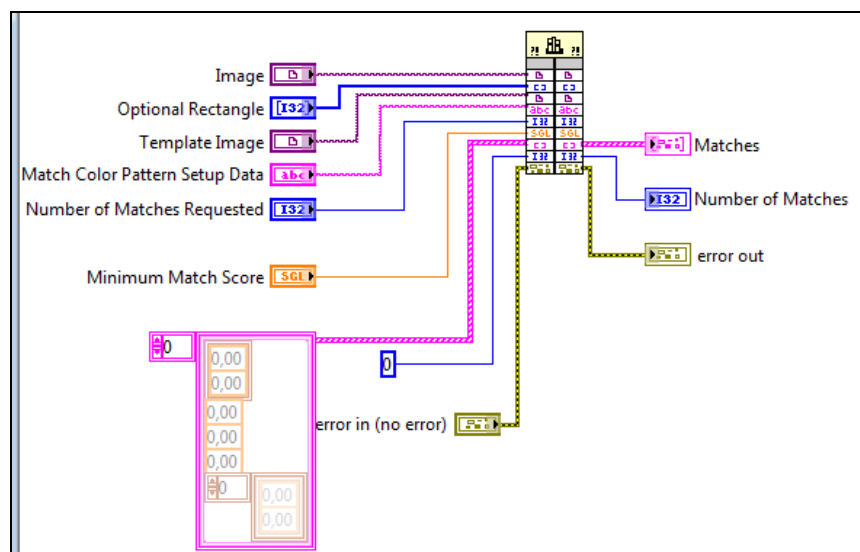


Diagrama de Bloques



- **Posición de las botellas encontradas (Overlay Blister Pack Matches Position)**

Panel Frontal

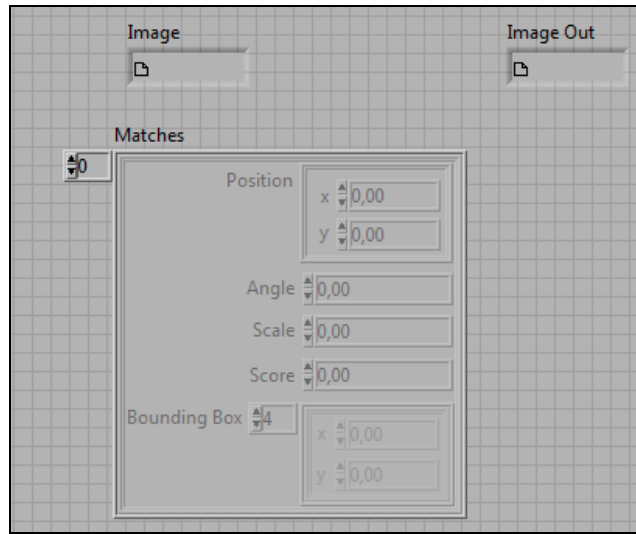
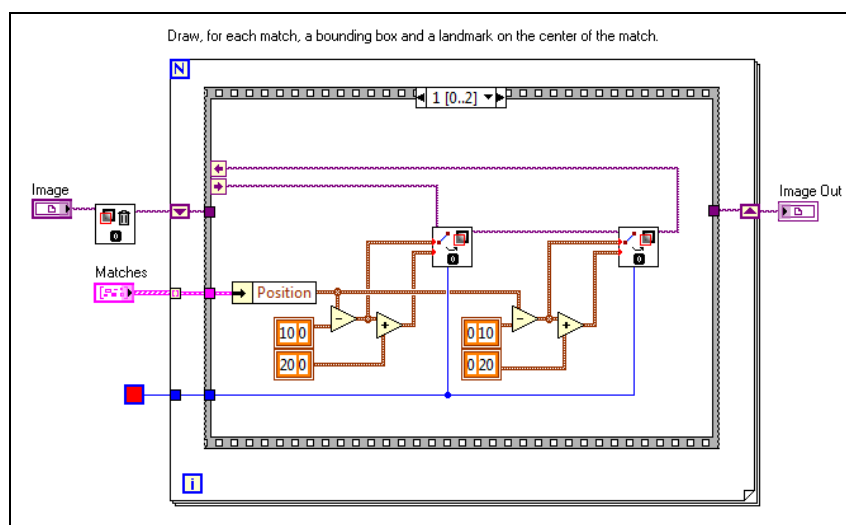


Diagrama de Bloques



- **Añade las nuevas botellas encontradas (IVA Add Offset to Matches)**

Panel Frontal

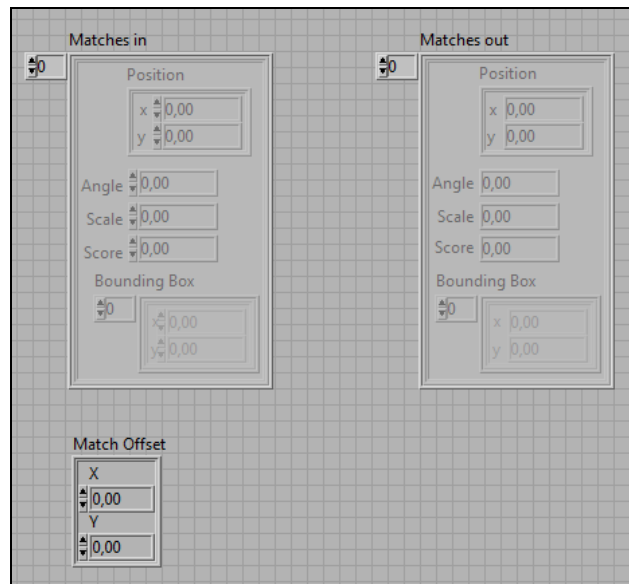
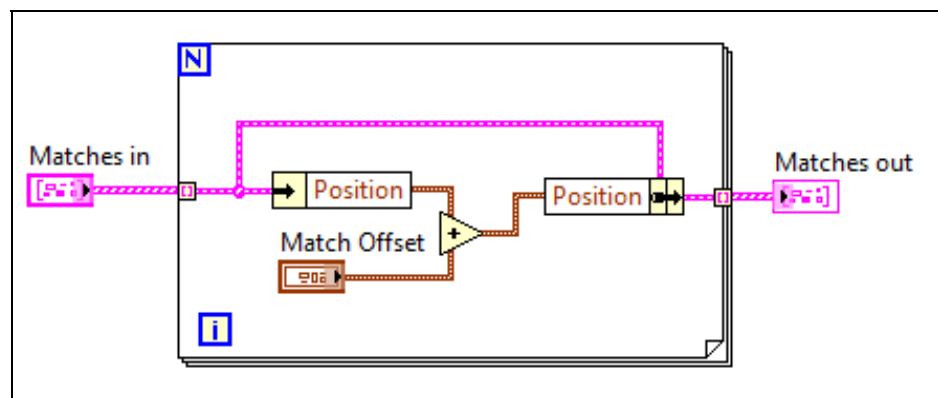


Diagrama de Bloques



- **Eliminación de imágenes (IMAQ Dispose)**

Panel Frontal

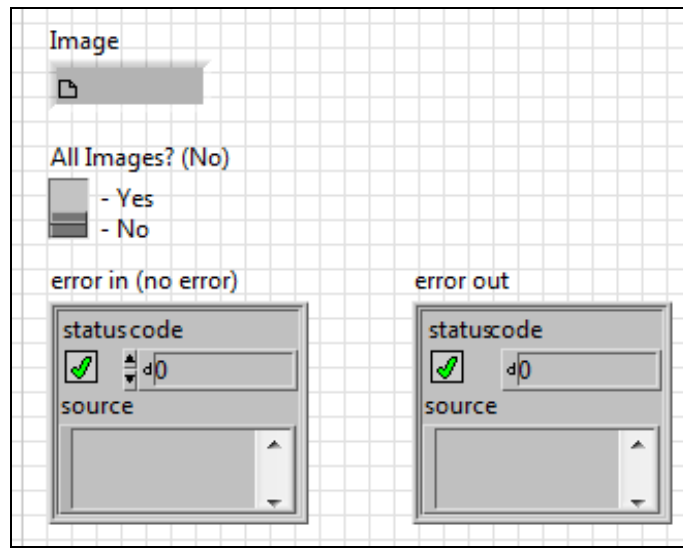
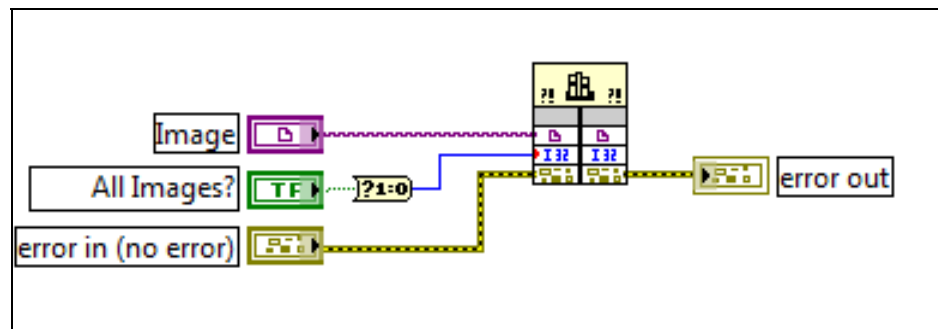


Diagrama Frontal



B. Pruebas de Evaluación en Vision Assistant

- **Sensibilidad de Color Baja**

Estrategia de búsqueda Conservativa

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	6	0
825	6	6	0
850	6	6	0
875	6	6	0

Estrategia de búsqueda Equilibrada

Estrategia de Búsqueda (Search strategy) Equilibrada (Balance)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	5	1
675	6	5	1
700	6	5	1
725	6	5	1
750	6	5	0
775	6	5	0
800	6	5	0
825	6	4	0
850	6	4	0
875	6	3	0

Estrategia de búsqueda Agresiva

Estrategia de Búsqueda (Search strategy) Agresiva (Agressive)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	5	0
825	6	5	0
850	6	5	0
875	6	4	0

Estrategia de búsqueda Muy Agresiva

Estrategia de Búsqueda (Search strategy) Muy Agresivo (Very Agressive)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	4	2
675	6	4	2
700	6	4	2
725	6	4	1
750	6	4	1
775	6	4	0
800	6	4	0
825	6	4	0
850	6	4	0
875	6	3	0

- **Sensibilidad de Color Alta**

Estrategia de búsqueda Conservativa

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Alta (High)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	6	0
825	6	6	0
850	6	5	0
875	6	3	0

Estrategia de búsqueda Equilibrada

Estrategia de Búsqueda (Search strategy) Equilibrada (Balance)	Sensibilidad de Color (Color Sensitivity) Alta (High)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	5	0
800	6	5	0
825	6	5	0
850	6	4	0
875	6	2	0

Estrategia de búsqueda Agresiva

Estrategia de Búsqueda (Search strategy) Agresiva (Aggressive)	Sensibilidad de Color (Color Sensitivity) Alta (High)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	5	1
675	6	5	1
700	6	5	1
725	6	5	1
750	6	5	0
775	6	5	0
800	6	5	0
825	6	5	0
850	6	4	0
875	6	3	0

Estrategia de búsqueda Muy Agresiva

Estrategia de Búsqueda (Search strategy) Muy Agresiva (Very Aggressive)	Sensibilidad de Color (Color Sensitivity) Alta (High)		
Mínimo Puntaje de Encuentros (Minimum Match Score)	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
650	6	6	0
675	6	6	0
700	6	6	0
725	6	6	0
750	6	6	0
775	6	6	0
800	6	6	0
825	6	6	0
850	6	5	0
875	6	3	0

C. Pruebas de Inspección en Labview

- Variando los Atributos

Punto Mínimo de Encuentro 650

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		Mínimo Punto de Encuentro = 650
Atributos	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
Forma	6	6	0
Color	6	6	0
Forma y Color	6	6	0

Punto Mínimo de encuentro 700

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		Mínimo Punto de Encuentro = 700
Atributos	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
Forma	6	6	0
Color	6	6	0
Forma y Color	6	6	0

Punto Mínimo de encuentro 775

Estrategia de Búsqueda (Search strategy) Conservativa (Conservative)	Sensibilidad de Color (Color Sensitivity) Baja (Low)		Mínimo Punto de Encuentro = 775
Atributos	Color Score Weight = 500		
Patrón 1	Botellas Esperadas	Botellas Encontradas	Bloques Out
Forma	6	5	0
Color	6	5	0
Forma y Color	6	6	0