



**Universidad
Zaragoza**

Trabajo Fin de Grado

Control de una línea de fabricación dependiente de
un sistema de reconocimiento de patrones.

Autor

Guillermo Perales Gran

Directores

Ana María López Torres
Eduardo Gil Herrando

Escuela Universitaria Politécnica de Teruel
2016



Control de una línea de producción dependiente de un sistema de reconocimiento de patrones

RESUMEN

En este Trabajo Fin de Grado se presenta un proceso automatizado capaz de distinguir distintos tipos de piezas mediante la utilización de un sistema detector de patrones basado en procesado digital de imágenes del sistema.

Para la implementación de dicho proceso se ha utilizado una maqueta disponible en la EUPT en la que se puede simular un proceso industrial automático controlado mediante un controlador lógico programable (PLC). En este proceso, se ha integrado una cámara USB que se utilizará para capturar las imágenes, en un determinado punto de la línea de producción, procesándolas y analizándolas posteriormente gracias a un sistema detector de patrones implementado en Labview. Una vez se tengan detectadas las piezas, el sistema indicará al PLC las siguientes acciones a llevarse a cabo en el proceso.

Además del proceso automático mencionado, el sistema tendrá un modo manual que permitirá el movimiento de los distintos elementos que intervienen en el proceso. En este modo, se permitirá la configuración de varios parámetros para el desarrollo del proceso de reconocimiento de patrones.

El control del sistema así como su supervisión, se llevarán a cabo a través del SCADA (Supervisory Control And Data Acquisition) implementado en Labview. El SCADA, será capaz en todo momento de indicar el estado y modo de funcionamiento en que se encuentra el sistema. Además, en caso de paradas por fallos o averías, proporcionará al usuario información al respecto.





Índice

1. Introducción.....	1
1.1. Antecedentes.	1
1.2. Objetivos.....	1
1.2.1. Objetivos generales.....	1
1.2.2. Objetivos específicos.....	2
1.3. Estructura de la memoria.....	2
2. Labview como entorno de desarrollo.	3
2.1. Motivos de elección de Labview.....	3
2.2. Instrumentos virtuales.....	3
2.3. Metodología de programación.....	4
3. Descripción general del sistema.	5
3.1. Introducción.....	5
3.2. Características del sistema.....	5
3.2.1. Proceso productivo.....	5
3.2.2. Panel de control.....	6
3.2.3. Modos de funcionamiento.....	7
3.2.4. Comportamiento ante fallos.....	8
3.3. Elementos del sistema.....	8
3.4. Diagrama de bloques.....	9
3.5. Parámetros de configuración del sistema detector de patrones.....	9
3.5.1. Modificar número de patrones.....	10
3.5.2. Crear nuevo patrón.....	10
3.5.3. Modificar Tmaq1 y Tmaq2.....	11
3.5.4. Ajuste de color y forma como parámetros de detección.....	11
3.5.5. Exactitud de detección.....	12
3.6. Archivos de registro.....	12
3.7. Factores externos.....	13
4. Sistema de proceso automático.....	14
4.1. Autómata programable.....	14
4.1.1. Introducción.....	14
4.1.2. Estructura de un autómata programable.....	14
4.1.3. Programación del autómata.....	15
4.2. Maqueta utilizada.....	16
4.3. PLC. TSX 3722.....	16



4.3.1.	Características.....	16
4.3.2.	Tarjetas de E/S utilizadas.....	17
4.3.3.	Fuente de alimentación.....	18
4.3.4.	Software de programación. PL7 Pro.....	18
4.3.5.	Funcionamiento. Procesador monotarea.....	18
4.4.	Control de Maqueta mediante PLC.....	19
4.4.1.	Definición de Entradas y Salidas en el PLC.....	19
4.4.2.	Estructura de la aplicación en PL7.....	20
4.4.3.	GRAFSET de Producción normal automática.....	20
4.4.3.1.	Grafos.....	20
4.4.3.2.	Habilitaciones.....	22
4.4.3.3.	Contador de piezas.....	22
4.4.3.4.	Tiempo de trabajo de Máquinas.....	23
4.4.4.	GRAFSET de conducción.....	23
4.4.5.	Implementación de guía GEMMA.....	24
4.4.6.	Paradas de emergencia.....	26
4.4.7.	Gestión de rearmes.....	26
4.4.8.	Programación de Salidas.....	27
5.	Comunicación de dispositivos.....	28
5.1.	OPC vs DAQ.....	28
5.2.	Prestaciones y limitaciones de comunicación.....	28
5.2.1.	Prestaciones de comunicación de la DAQ.....	28
5.2.2.	Prestaciones de comunicación del PLC.....	29
5.2.3.	Limitaciones en la comunicación PLC-DAQ.....	29
5.3.	Soluciones adoptadas en la comunicación PLC-DAQ.....	29
5.3.1.	Salidas analógicas de la DAQ (0_10v) – Entradas analógicas del PLC (0_10v).....	29
5.3.2.	Salidas analógicas del PLC (0_10v) – Entradas analógicas de la DAQ (0_10v).....	31
5.3.3.	Salidas digitales de la DAQ (5v) –Relés –Entradas digitalesdel PLC (24v).....	31
5.3.4.	Salidas digitales de la DAQ (5v) –Entradas analógicas del PLC (0_10v).....	31
5.3.5.	Salidas digitales del PLC (5v) –Entradas digitalesde la DAQ (5v).....	31
5.4.	Resumen de variables de comunicación.....	32
6.	Sistema detector de patrones.....	34
6.1.	Visión por computador.....	34
6.1.1.	Introducción.....	34
6.1.2.	Etapas de un sistema de visión por computador.....	34



6.1.3.	Muestreo y cuantificación. Pixeles.....	35
6.2.	Elemento de captación.....	35
6.2.1.	Elemento sensor.	35
6.2.2.	Distancia focal, profundidad de campo y diafragma.	36
6.2.3.	Iluminación.....	37
6.2.4.	Exposición.....	37
6.3.	Cámara utilizada: Creative Live! Cam Chat HD.....	38
6.3.1.	Características.....	38
6.3.2.	Limitaciones de la cámara en cuanto a exposición.....	39
6.4.	Algoritmos de detección de patrones.	39
6.4.1.	Diagrama de flujo de creación de nuevos patrones.....	39
6.4.2.	Rectángulo de trabajo.	40
6.4.3.	Diagrama de flujo de detección de patrones de forma automática.....	40
6.5.	Uso de Labview para programación del sistema detector de patrones.....	41
6.5.1.	Adquisición y procesamiento de imágenes a través del paquete NI VISION.	41
6.5.2.	Instrumentos virtuales de NI Vision utilizados.	42
6.5.3.	Subvis para la detección de patrones.	44
6.5.4.	Programación de algoritmos.....	45
6.5.4.1.	Creación de patrones. Creacion_patron.vi.....	45
6.5.4.2.	Detector de patrones. Detector_patrones.vi.....	46
7.	Conclusiones.	48
8.	Bibliografía.....	49





1. Introducción.

1.1. Antecedentes.

Las necesidades de las industrias de alcanzar los más altos niveles de control de calidad, y el mayor rendimiento de sus procesos productivos, mejorando la competitividad, han llevado a buscar mejoras en los diferentes sistemas empleados en dichos procesos, surgiendo como una solución alternativa o complementaria a la utilización de operarios, los sistemas de visión artificial.

La visión artificial llegó a España a finales de los años 90, y desde entonces, ha habido numerosos cambios e innovaciones con resultados muy positivos entorno al sector industrial, donde han sido más notables, pero la visión artificial se extiende en numerosos sectores.

En el campo industrial, con la visión artificial podemos:

- Automatizar tareas repetitivas de inspección realizadas por operarios.
- Evitar la presencia de operarios en entornos peligrosos (térmicos, nucleares, ruidosos).
- Realizar controles de calidad de productos que no es posible verificar por métodos tradicionales.
- Realizar inspección de objetos sin contacto físico.
- Realizar la inspección del 100% de la producción a gran velocidad (reducción de tiempos de ciclo).
- Realizar inspecciones en procesos donde existe diversidad de piezas con cambios frecuentes de producción.
- Incrementar de la fiabilidad. Eliminación de criterios subjetivos
- Mejorar la calidad de los productos y de los procesos involucrados. Detección de defectos más pequeños, manipulado más preciso de piezas, etc.
- Integración en el entorno automatizado. Dotación sensorial del entorno automatizado.

Como se observa, existe un gran número de ventajas que aporta el uso de la visión artificial en la industria. En este proyecto, se va a realizar la integración de un sistema de visión artificial en un proceso automático, que sin ayuda de la visión por computador, necesitaría de la acción de un operario para llevarse a cabo con similares prestaciones.

1.2. Objetivos.

El objetivo principal que se busca con este proyecto es realizar la integración de un sistema de visión por computador en un proceso controlado por PLC. Es decir, no se define un proceso concreto a desarrollar, sino la integración de forma útil de un sistema de visión en un posible proceso automático desarrollado en la maqueta disponible.

En cualquier proyecto de automatización, existe una primera fase de definición de objetivos y características del sistema en la que en la mayoría de las ocasiones, inicialmente no quedan completamente definidos. Esto es una dificultad que nos encontramos en este proyecto, ya que, aunque tenemos definido que el proceso se llevará a cabo en una sencilla maqueta disponible, se otorga por completo al diseñador la tarea inicial de definir las características del sistema a diseñar con los medios disponibles. Es por eso que se establece la diferencia entre objetivos generales (objetivos que se buscan con el desarrollo del proyecto) y objetivos específicos (objetivos específicos del sistema a desarrollar).

1.2.1. Objetivos generales.

Los objetivos generales que se buscan con la realización de este proyecto son:

- Desarrollo en una aplicación real de los conocimientos adquiridos en los campos de Visión por Computador y Automatización Industrial. Integración de ambos campos.
- Definición realista de objetivos específicos de un proceso industrial automático.
- Traslado de las características de un hipotético proceso real al proceso desarrollado sobre la maqueta disponible.



- Implementación de un SCADA para monitorizar y gestionar un proceso productivo.
- Comunicación de dispositivos de distintos fabricantes.
- Implementación del sistema utilizando Software y Hardware disponible en el centro.

1.2.2. Objetivos específicos.

Como ya se ha mencionado, son los objetivos que se han fijado para desarrollar el proceso:

- Integración de un sistema detector de patrones en un proceso automático desarrollado sobre una maqueta y controlado mediante PLC.
- Fiabilidad en la detección de las piezas.
- Versatilidad máxima del sistema. El sistema detector de patrones tendrá ciertos parámetros configurables para otorgar mayores posibilidades al usuario.
- Establecimiento de tiempos de ciclo mínimos.
- Desarrollo de distintos modos de funcionamiento: Manual, Automático, etc.
- Respuesta rápida y segura del sistema ante errores o averías durante los distintos modos de funcionamiento.
- Gobierno y monitorización del sistema por parte del usuario mediante el SCADA.

1.3. Estructura de la memoria.

En este apartado se detalla la estructura de la memoria, para intentar ayudar al lector a seguir el texto. Como es habitual en cualquier proyecto, en este primer punto se ha comenzado introduciendo y contextualizando el estado actual de la visión por computador en los entornos industriales automatizados y se han expuesto los objetivos marcados.

En el segundo punto se exponen los motivos para utilizar Labview como plataforma de desarrollo del sistema y la metodología de programación que se seguirá.

En el tercer punto, se hace una descripción de las características generales del sistema automático que se ha llevado a cabo, sin entrar en la programación que lleva implícita. Se comentan los factores externos que influyen en el sistema y se describen los elementos que lo componen.

En el cuarto punto se va a dejar de lado el sistema detector de patrones para centrarnos en el proceso automático que se ha implementado. Inicialmente se van a dar unas nociones básicas de automatización, necesarias para la comprensión del sistema desarrollado. Después se analizará la maqueta y el PLC utilizado, para finalmente, explicar la programación llevada a cabo en el PLC para el control de la maqueta.

En el quinto punto se detallará la forma en que se comunicarán el PLC con el SCADA. Se explicarán las distintas soluciones llevadas a cabo ante las limitaciones del hardware disponible.

El sexto punto se centrará en la explicación del sistema detector de patrones. Inicialmente se comentarán ciertos aspectos básicos del campo de la visión por computador. Una vez descritos estos aspectos, se detallarán las características de la cámara utilizada. Finalmente, se expondrán las características principales de los algoritmos implementados en Labview para poder realizar la detección de patrones.

En el séptimo punto se exponen las conclusiones finales a las que se llega tras la realización del proyecto.

Se concluirá con la Bibliografía utilizada.

En el CD que se entrega junto con esta memoria, se incluye la memoria completa y se añaden dos anexos Anexo A y Anexo B, donde se detallará la programación fundamental del PLC y de Labview respectivamente. Además, se incluye el archivo de programación del PLC, "**Control de línea de producción.stx**" y los archivos de programación en Labview, entre los que está el VI principal ("**SCADA.vi**"), el resto de subVIs incluidos en la carpeta llamada "**Creaciones propias**" y otros archivos necesarios para el funcionamiento del sistema como los incluidos en la carpeta "**Archivos SCADA**".



2. Labview como entorno de desarrollo.

2.1. Motivos de elección de Labview.

Antes de comenzar a explicar el sistema implementado, se van a detallar los motivos por los que se ha utilizado Labview como herramienta fundamental de desarrollo del sistema. Desde el principio, se pensó en la utilización de Labview debido a los motivos que se exponen a continuación:

- National Instruments (NI) es uno de los fabricantes líderes en el mercado en visión artificial y procesamiento de imágenes.
- Posee una extensa colección de drivers para periféricos. De forma rápida y sencilla, se puede utilizar la cámara disponible para realizar la captura de imágenes.
- Tiene gran número de funciones de visión en el Módulo Vision Development. Junto con la extensa biblioteca de funciones de visión artificial y procesamiento de imágenes, este módulo incluye el Vision Assistant, un entorno interactivo para generación de prototipos de aplicaciones de inspección que pueden generar código listo para ejecutar en cualquiera de los lenguajes soportados.
- Posibilidad de crear un sencillo SCADA mediante sus funciones básicas. Se pueden conseguir mayores prestaciones con las funciones integradas en el Módulo NI LabVIEW Datalogging and Supervisory Control (DSC).
- Posee las funciones necesarias para comunicarse con otros dispositivos usando una variedad de opciones de E/S y protocolos incluyendo E/S digitales, Modbus, Serial RS232, TCP/IP, EtherNet/IP y EtherCAT.
- Posibilidad de crear Servidor OPC para conectividad de hardware de distintos fabricantes.
- Disponibilidad de Tarjeta de Adquisición de datos.
- Conocimiento del entorno y programación de Labview y disponibilidad de licencia de estudiante para los módulos necesarios con opción de soporte del fabricante.

Resumiendo, Labview posee las capacidades para desarrollar el sistema detector de patrones, implementar el SCADA y la comunicación con el PLC. Además, el conocimiento de la plataforma y la disponibilidad de licencia para su uso y soporte, son motivos de sobra para justificar el uso de Labview como elemento de desarrollo del sistema.

2.2. Instrumentos virtuales.

Los ficheros generados con Labview se llaman Instrumentos Virtuales, VIs, llamados así porque su apariencia y comportamiento simula a instrumentos físicos. Los VIs utilizan funciones que manipulan las entradas del usuario o datos provenientes de otras fuentes y muestran la información procesada o la pasan a ficheros u otros ordenadores. Cada VI se compone de las siguientes partes [2]:

- **Panel frontal** (front panel) o interface con el usuario. Se construye mediante controles e indicadores, los cuales corresponden a las entradas y salidas del VI respectivamente. Los controles los constituyen diales, conmutadores, potenciómetros deslizantes y otros dispositivos de entrada, que suministran los datos de entrada al VI. Los indicadores son gráficos, leds y otros tipos de displays que simulan los dispositivos de salida y visualizan los datos adquiridos o generados por el VI.
- **Diagrama de bloques** (block diagram) o código fuente que define el comportamiento del VI. En el diagrama de bloques se pueden tener los siguientes elementos:
 - **Terminales:** Representan el tipo de dato del control o indicador. Los terminales son puertos de entrada y salida que intercambian información entre el panel frontal y el diagrama de bloques.
 - **Nodos:** Son objetos del diagrama de bloques que tienen entradas/salidas y realizan operaciones cuando el VI se ejecuta. Es equivalente a los estamentos, operadores, funciones y subrutinas en los lenguajes de programación basados en texto
 - **Cables:** Los datos se distribuyen por el diagrama de bloques mediante los cables. Son las conexiones entre los diferentes terminales y nodos del diagrama de bloques.



Cada cable contiene un único dato que puede ser leído por diferentes nodos o VIs. El aspecto del cable (grosor, color...) indica el tipo de dato que está presente en el mismo.

- **Estructuras:** Son representaciones gráficas de las estructuras de lazos, estamentos tipo case, etc de los lenguajes de programación basados en texto. Se utilizan las estructuras para repetir bloques de código o para ejecutar código de forma condicional o en un orden determinado
- **Icono** que identifica al propio VI, puesto que se pueden utilizar VIs en otros VIs. **A un VI llamado por otro se le denomina subVI**, esto es el equivalente a las subrutinas en la programación basada en texto. Se resalta esta nomenclatura ya que a lo largo del texto se hará referencia en multitud de ocasiones a los subVIs utilizados en el sistema.

2.3. Metodología de programación.

Normalmente, al diseñar una aplicación Labview, se comienza desde el VI más general definiendo las entradas y salidas de la aplicación. Después se crean subVIs que realizan tareas más sencillas dentro del VI general. Este método de diseño es una de las ventajas de Labview. Se pueden diseñar fácilmente aplicaciones complejas utilizando una estructura jerárquica y usando elementos comunes varias veces dentro de la aplicación.

Esta es la metodología que se ha seguido en nuestro sistema, donde el VI llamado "**SCADA.vi**", se ejecuta directamente cuando se hace doble click sobre él. Una vez en ejecución, contiene varios subVIs que serán llamados cuando así lo solicite el proceso. Se ha personalizado el comportamiento de estos subVIs, de forma que algunos de ellos presentan menús de interacción con el usuario y todos ellos, cuando son ejecutados por completo, vuelven a dar paso a la ejecución del VI que los había llamado.

Resumiendo, en nuestro sistema se aprovechan las características de la programación mediante subVIs, por dos motivos principalmente:

- Se crea estructura jerarquizada más sencilla y manejable de programar y fácilmente escalable.
- Se aprovecha la posibilidad de lanzar ventanas de opciones mediante la ejecución del correspondiente subVI, cuando el proceso lo precise.

Todos los SubVIs se han agrupado juntos en la librería Addons y han sido salvados, también juntos, en la ubicación "**C:\Archivos de programa\National Instruments\LabVIEW 2014\vi.lib\Creaciones propias**" del PC. Se han añadido estos archivos en el CD adjunto.

En el apartado 6.5.3 son explicados los diferentes subVIs (utilizados principalmente para las funciones de detección de patrones), mientras que su programación se detalla en el anexo B.

Es importante mencionar también el uso de un buen número de variables globales (16) para la comunicación directa entre subVIs, al hacer referencia a las mismas. Las variables globales son un tipo especial de VI, que únicamente dispone de Panel Frontal, en el cual se define el tipo de datos de la variable y el nombre de identificación imprescindible para después podernos referir a ella. Podemos crear un VI para cada variable global o definir las todas en el mismo, que es la opción que se ha elegido, de forma que todas ellas están contenidas en el VI llamado "**Parametros_globales.vi**". Pueden consultarse estas variables en el apartado B.3 del anexo B.

3. Descripción general del sistema.

3.1. Introducción.

Como se menciona en varias ocasiones a lo largo de esta memoria, se va a diseñar una línea de producción automática que contiene un sistema detector de patrones. En los siguientes puntos se van a describir las características del sistema final, así como los factores externos que influyen en la detección de patrones y los distintos elementos hardware que son necesarios para su realización.

3.2. Características del sistema.

A continuación se mencionan las principales características del sistema diseñado, cumpliendo los objetivos fijados en el apartado 1.2. En este apartado, se explicarán estas características sin entrar en detalle de la programación que se ha llevado a cabo para desarrollarlas. Dicha programación, se explicará con detalle en puntos posteriores de la memoria.

3.2.1. Proceso productivo.

Tenemos un proceso productivo emulado en una maqueta en el que determinadas piezas son transportadas por distintos elementos integrados en la misma. En dos puntos determinados del proceso, tenemos sendas máquinas herramienta que realizan su función durante un determinado tiempo, función del tipo de pieza que haya sido reconocido en el momento en que la pieza llega al Selector 1. La figura 3.1 muestra la maqueta del proceso con los distintos elementos.

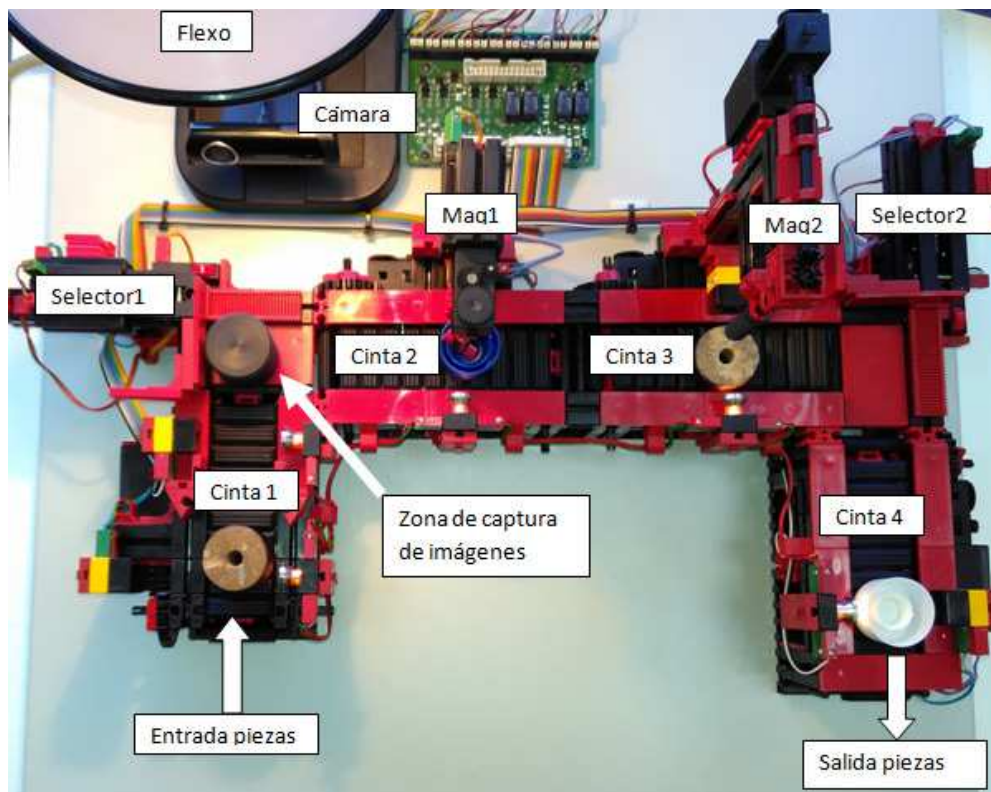


Figura 3.1. Maqueta del proceso productivo.

Los operarios introducirán piezas al inicio del proceso de forma que en la primera cinta podrá haber más de una pieza, aunque no es recomendable porque este hecho puede afectar a la detección de la pieza realizada en la zona de carga del Selector 1. En el resto de cintas, solamente podrá haber una pieza en cada momento.

El proceso es capaz de trabajar con distintas piezas, siempre que cumplan las siguientes características:

- Las dimensiones de las piezas deberán ser adecuadas para poder ser transportadas por todos los elementos del proceso de forma segura.

- Las piezas deberán tener forma y color regulares para poder ser detectadas correctamente, ya que solamente se dispone de una cámara que recoge la imagen de la pieza en dos dimensiones.

Todo este proceso es monitorizado y comandado desde un panel de control integrado en el SCADA, que permitirá al usuario una gestión cómoda y segura del proceso.

3.2.2. Panel de control.

Como ya se ha mencionado en varias ocasiones, se ha desarrollado un SCADA en Labview para control y supervisión del proceso. SCADA viene de las siglas de "Supervisory Control And Data Adquisition". Se trata de una aplicación software especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde la pantalla del ordenador. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios. [1]

El panel de control, va a ser la interface hombre –máquina (HMI) del SCADA, mediante la que el usuario va a poder interactuar en el proceso automático desarrollado. En la figura 3.2 se muestra el panel de control (Front Panel del archivo "SCADA.vi") separado en 4 partes bien diferenciadas que a continuación se explican. Esta imagen puede consultarse ampliada en el apartado B.1 del anexo B.

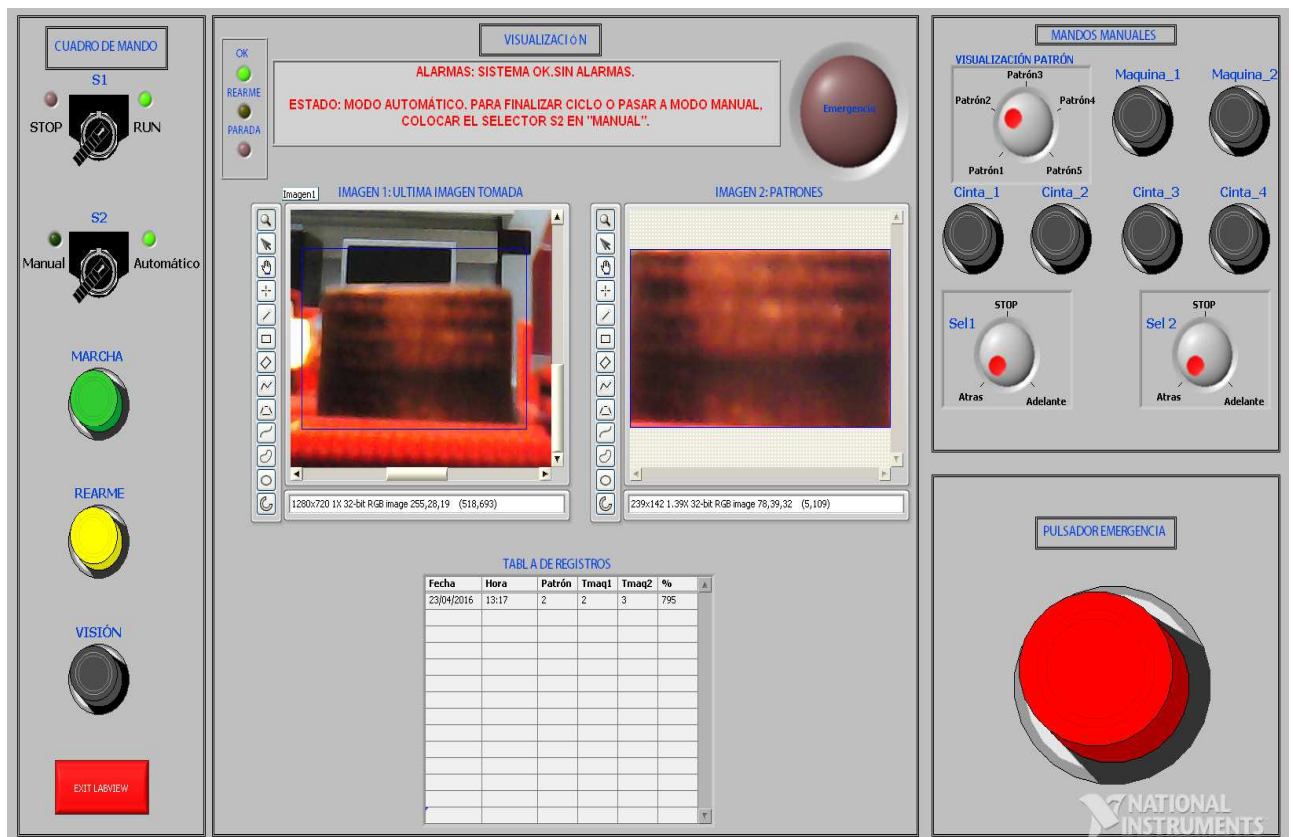


Figura 3.2. Panel de control del sistema.

Cuadro de mando.

En esta zona, se tienen los pulsadores y selectores necesarios para gobernar el sistema desarrollado. A continuación se detallan los mandos disponibles:

- S1:** Selector que establece el funcionamiento o no del sistema. Si está en la posición de "STOP", el sistema global queda desconectado y el PLC en particular queda en STOP. En la posición de "RUN", el sistema queda encendido y el PLC en RUN.
- S2:** Selector con dos posiciones ("Manual" y "Automático") que establece el modo de funcionamiento. En el apartado 3.2.3 se explican los posibles modos.



- **MARCHA:** Pulsador utilizado para dar marcha al proceso, ya sea por estar en su posición inicial o tras rearme por haber sufrido alguna parada de emergencia.
- **REARME:** Pulsador utilizado para rearmar el sistema cuando se produce una parada de emergencia.
- **VISIÓN.** Pulsador utilizado para acceder al menú de configuración de los parámetros del sistema detector de patrones. Veremos este menú con detalle en el apartado 3.5.
- **EXIT LABVIEW:** Es un pulsador utilizado para cerrar la ejecución del SCADA. Este pulsador solo tiene efecto cuando el sistema está parado (S1 en "STOP").

Visualización.

En esta zona, se observa que se tienen distintos elementos de visualización que proporcionan al usuario en todo momento información sobre el desarrollo del proceso.

- **Indicadores de estado:** Se tienen tres indicadores que muestran si el sistema está ok (OK), está rearmado o en proceso de rearme (REARME), o si se ha producido alguna alarma que haya generado parada (PARADA).
- **Pantalla de información:** A la derecha de los indicadores de estado, se tiene una pantalla que mostrará información referente al modo de trabajo en que se encuentra el sistema. En caso de parada, esta pantalla indicará el motivo de la misma. Servirá de guía al usuario para el gobierno del proceso en todo momento.
- **Led Emergencia:** A la derecha de la Pantalla de información, se tiene un gran led que indica si el sistema tiene algún error o emergencia activo.
- **IMAGEN 1:** Muestra la última imagen tomada por el sistema. Esta imagen incluye el rectángulo que contiene la parte de la imagen que se compara con los distintos patrones a la hora de la detección. Se detallará este proceso en el apartado 6.4.2.
- **IMAGEN 2:** Muestra el patrón reconocido si se está en modo Automático, o el patrón seleccionado mediante el selector "Visualización Patrón" si se está en modo Manual.
- **Tabla de Registros:** Muestra características obtenidas de las piezas detectadas como son fecha y hora de la detección, patrón detectado, % de acierto en la detección, etc. Esta tabla se reinicia cada vez que se arranca el sistema. No obstante, se almacenan todos estos registros en un archivo ubicado en la carpeta "**Archivos SCADA**" (ver apartado 3.6) por si se quiere analizar o contabilizar los resultados.

Mandos manuales.

Si se tiene el sistema trabajando en modo Manual, se podrán activar las cintas, selectores y máquinas mediante su correspondiente mando/selectores manual. También se tiene el selector de "Visualización de Patrón" ya comentado anteriormente.

Pulsador de emergencia

La seta de emergencia, es un pulsador cuya acción provoca una parada inmediata de los elementos del sistema. Con la seta de emergencia pulsada, no será posible ningún movimiento de los elementos del sistema, ni en modo Manual ni en modos automáticos.

3.2.3. Modos de funcionamiento.

En todo momento, el SCADA comunicará mediante la pantalla de información del modo de funcionamiento en que se encuentra el proceso. Dicha pantalla, proporcionará ayuda al usuario sobre las opciones disponibles para cada modo. Se tienen los siguientes modos:

- **Modo Inicio.** Es el estado inicial una vez se activa el sistema. Cuando es activado se produce un reinicio de las variables utilizadas en el modo Automático. Es un estado de espera a que se den las condiciones para pasar a Manual o Automático.
- **Modo Manual.** El usuario podrá activar los distintos elementos (cintas, selectores, máquinas) del proceso. El proceso se encontrará en este modo siempre que S2 esté en la posición



“Manual” y además no haya alarmas en el sistema ni el proceso se encuentre inmerso en una etapa de proceso Automático.

- **Modo Visión.** Modo de funcionamiento en el que se accede al menú de configuración de los parámetros del sistema detector de patrones. Se accede siempre desde el modo Manual pulsando el pulsador Visión.
- **Modo de Posicionamiento:** En realidad, no es un modo de funcionamiento, sino un paso previo para que comience el modo Automático. Se comprueba que los selectores estén en posición inicial y que no hay piezas en el circuito antes del comienzo del ciclo.
- **Modo Automático.** En este modo, se ejecutará el proceso establecido de forma automática. El usuario será el encargado de introducir las piezas al inicio del proceso y de sacarlas al final del mismo. Para que se dé este modo, siempre será necesario (no suficiente) que S2 esté en “Automático”.
- **Modo Fin de Ciclo:** Es un modo similar al Automático pero con la salvedad de que no deja introducir más piezas en el sistema. Se accede al mismo desde el modo Automático posicionando S2 en “Manual”. Es el paso previo necesario para pasar de modo Automático a modo Manual de forma segura.

A lo largo del texto, se hará referencia en varias ocasiones a modos automáticos, englobando en esta expresión los modos Automático, de Posicionamiento y Fin de Ciclo.

3.2.4. Comportamiento ante fallos.

En cualquier modo de funcionamiento, el sistema deberá ser capaz de detectar fallos y actuar ante los mismos priorizando la seguridad tanto de los operarios como de la instalación. De este modo, cuando detecte algún fallo, se producirá una parada inmediata y hasta que no se solucione el motivo de dicha parada, no se permitirá continuar con el proceso, en el caso de que estemos en un modo Automático, ni activar los elementos del proceso en caso de estar en el modo Manual.

Tras un fallo, siempre será necesario rearmar el sistema. Se ha diseñado de esta manera para aumentar la seguridad, siendo obligatorio que el usuario, tras conocimiento del motivo de la parada, sea el encargado de habilitar de forma activa el proceso.

Al igual que para los modos de funcionamiento, se informará mediante la pantalla de información y los leds de estado y emergencia cuando se produzca una situación de emergencia, dando información útil al usuario del motivo de la misma.

3.3. Elementos del sistema.

Para poder implementar el sistema, se ha necesitado utilizar varios elementos (algunos de ellos ya mencionados) y ha sido necesaria la interacción entre ellos.

- **PC:** Se utiliza un PC en el que corre el SCADA implementado en Labview y se almacenan las imágenes de patrones y otros archivos necesarios para el correcto funcionamiento de dicho SCADA. Por otro lado, el PC es la interfaz de conexión de la cámara al sistema, ya que ésta, va conectada al mismo mediante un puerto USB del PC. A su vez, el PC también hace de interfaz con la tarjeta de adquisición de datos a través de un puerto PCI. Mencionar también que el PLC es programado mediante puerto serie también desde el PC.
- **Sistema de supervisión, control y adquisición de datos (SCADA):** Servirá de interface con el operador mediante un conjunto de mandos manuales, indicadores y tablas, permitiendo al usuario interactuar en el proceso. También contendrá la programación necesaria para la comunicación con el PLC y la gestión del sistema detector de patrones.
- **Cámara de fotos:** será el elemento encargado de captar las imágenes para ser procesadas y analizarlas en busca de la coincidencia con los patrones establecidos para determinar la pieza que se está tratando.
- **Sistema de adquisición de datos (DAQ):** Servirá para comunicar las señales entre el PLC y el SCADA.

- **Maqueta:** Representa el proceso productivo. Consta de 4 cintas, 2 selectores y 2 máquinas herramienta que van trasladando las piezas con ayuda de 5 sensores de presencia, 4 finales de carrera y el sistema detector de patrones. Se detallan sus características en el apartado 4.2.

3.4. Diagrama de bloques.

La figura 3.3, muestra el diagrama de bloques que incluye la conexión de todos los elementos explicados en el apartado 3.3.

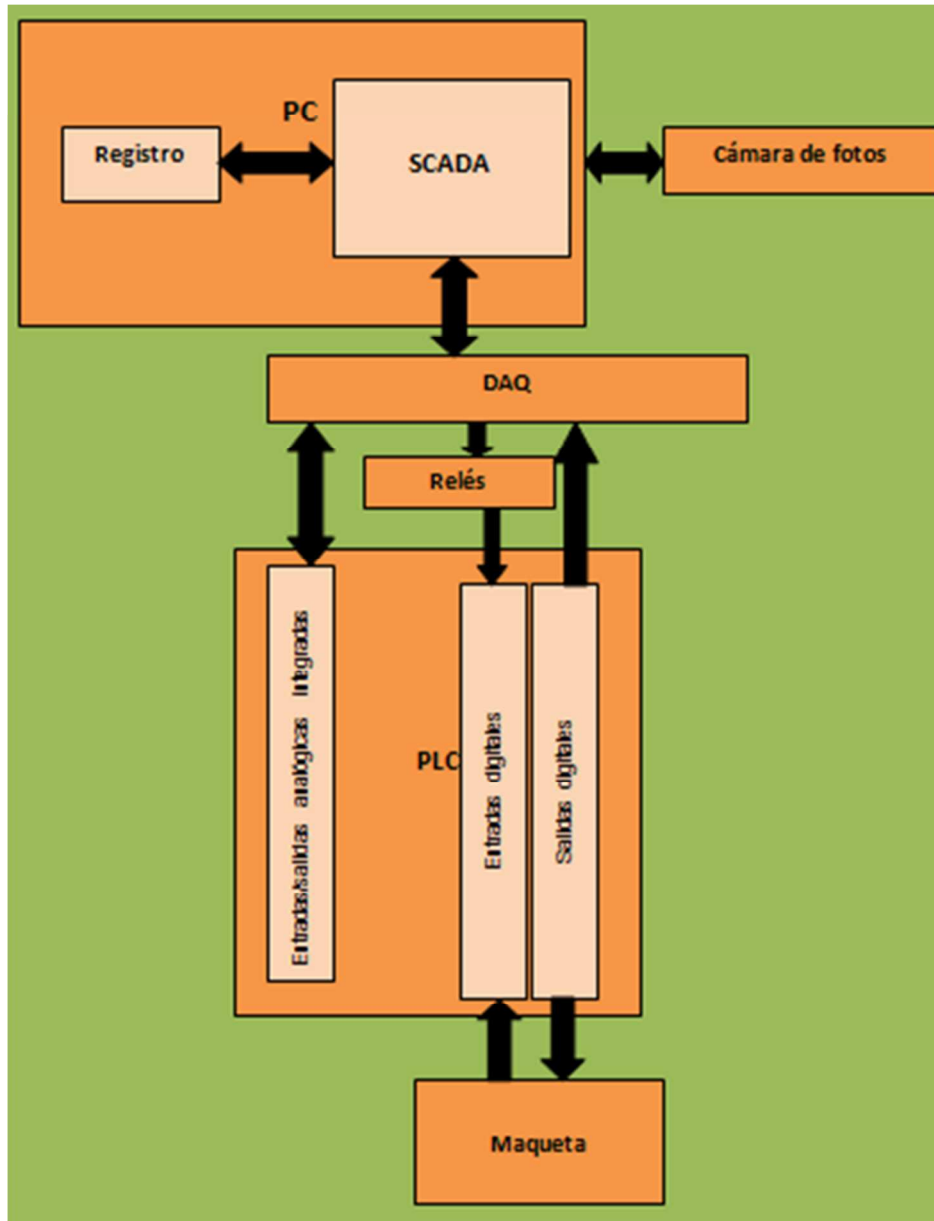


Figura 3.3. Diagrama de bloques del sistema.

3.5. Parámetros de configuración del sistema detector de patrones.

Se ha mencionado anteriormente que existe un modo de funcionamiento llamado Modo Visión, correspondiente a aquel modo en el que se pueden configurar ciertos parámetros del sistema de detección de patrones. Ya se ha visto que para acceder a este modo, simplemente se debe accionar el pulsador “**Visión**” cuando se está en modo Manual. Una vez se hace esto, se accede al menú mostrado en la figura 3.4, donde se aprecian 5 opciones de configuración.

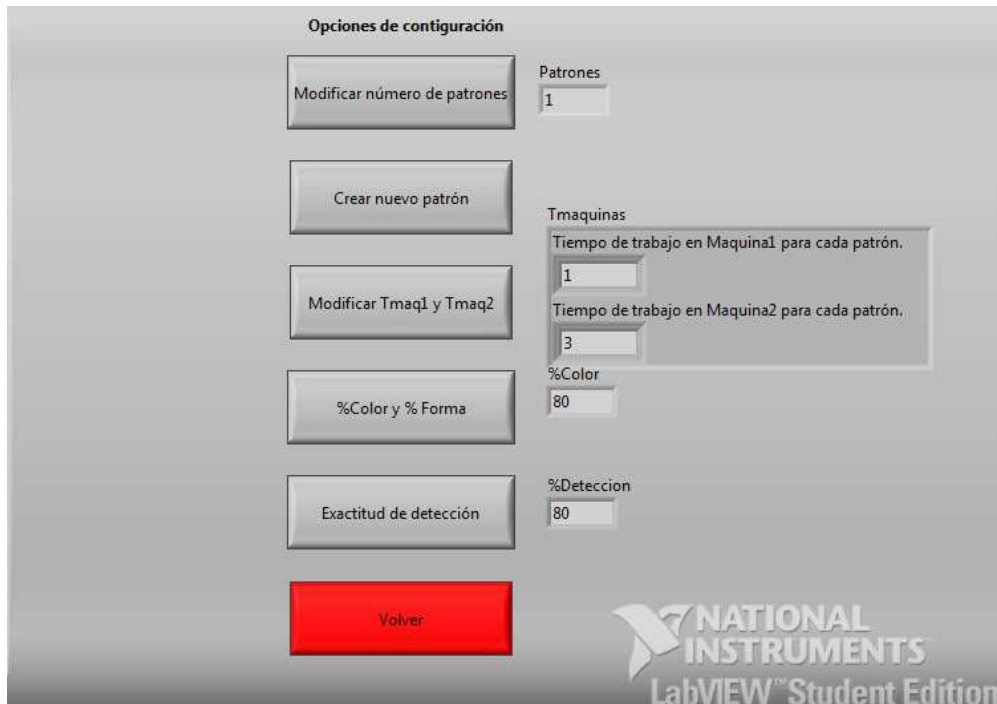


Figura 3.4. Menú de configuración de parámetros del sistema de detección.

3.5.1. Modificar número de patrones.

Esta opción permite modificar el número de patrones que se quiere que el sistema sea capaz de reconocer. Tal y como se observa en la figura 3.5, este número siempre deberá estar comprendido entre 1 y 5.

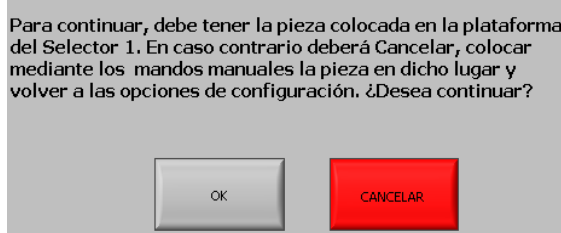


Figura 3.5. Menú de selección del número de patrones.

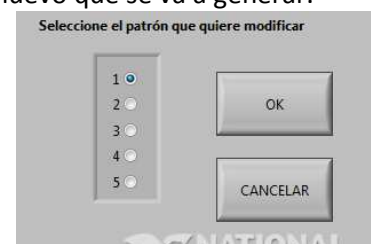
3.5.2. Crear nuevo patrón.

Esta opción es la más importante. El sistema permite el “aprendizaje” de nuevos patrones. Se realizan dos pasos previos antes de la toma de la imagen que se utilizará como patrón:

- Una vez se pulsa en “Crear nuevo patrón”, el sistema pregunta (ver figura 3.6.a) si se tiene la pieza a introducir en la posición de detección (Selector1).
- Si se tiene la pieza en dicha posición y se pulsa “OK”, el sistema pregunta por el número de patrón que se quiere sustituir (ver figura 3.6.b) por el nuevo que se va a generar.



(a)



(b)

Figura 3.6. Pasos previos a la creación del nuevo patrón. (a) Comprobar si la pieza está en el sistema de detección. (b) Seleccionar el patrón a sustituir

Si se vuelve a pulsar "Ok", se producirá el disparo de la cámara y se mostrará por pantalla la imagen tomada. En este momento el usuario, seleccionará un rectángulo que contendrá el patrón a utilizar para el reconocimiento. Si se acepta sin que se produzca ningún problema, se almacenará el patrón en la carpeta "Archivos SCADA", como se verá en el apartado 3.6. Esta ubicación será a la que va el sistema para recoger cada patrón a comparar en el proceso de detección. Cualquier fallo o cancelación durante el proceso, hará estéril la creación del nuevo patrón y se mantendrá el patrón disponible hasta ese momento.

3.5.3. Modificar Tmaq1 y Tmaq2.

El sistema permite asociar para cada patrón disponible el tiempo que se quiere que trabajen las máquinas 1 y 2 en un rango de 0 a 9 segundos. En la figura 3.7 se ve el proceso de guiado del sistema para modificar estos parámetros. También existirá la opción de reiniciar a 0 los tiempos para todos los patrones a la vez.

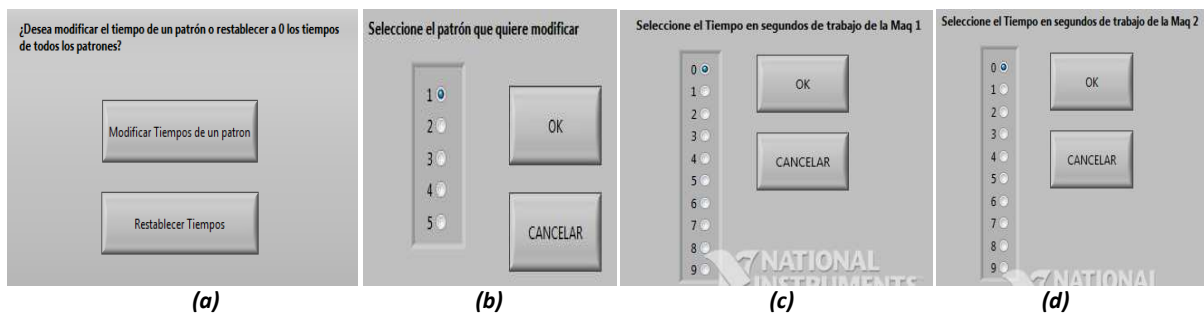


Figura 3.7. Pasos previos a la modificación de tiempos de trabajo de máquinas de un patrón. (a) Modificar tiempos de un patrón o reiniciar todos. (b) Selección del patrón cuyos tiempos se van a cambiar. (c) Selección del Tmaq1 de ese patrón. (d) Selección del Tmaq2 de ese patrón.

3.5.4. Ajuste de color y forma como parámetros de detección.

Mediante esta opción, el sistema permitirá al usuario definir el porcentaje que se tendrá en cuenta a la hora de realizar la detección de los dos parámetros en que se basa el algoritmo de detección: Color y Forma. Debido a la importancia de este parámetro, se ha restringido la posibilidad de modificación de este parámetro mediante contraseña. La figura 3.8 muestra el proceso de elección de este parámetro, considerando que el valor que se observa es el % en que se tiene en cuenta el color, mientras que la diferencia de este porcentaje, respecto de 100 es el % en que se tiene en cuenta el parámetro Forma. Tras las pruebas realizadas en el escenario disponible se establece como valor un 90% consiguiendo alta fiabilidad en los resultados. El algoritmo de detección que utiliza estos parámetros será explicado en el apartado 6.5.4.2.

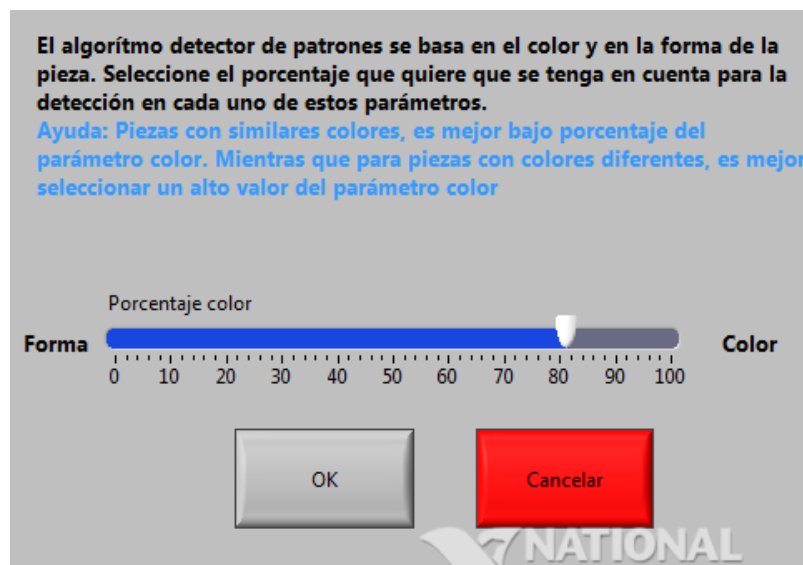


Figura 3.8. Proceso de elección de % de Color y % de Forma utilizado en la detección.

3.5.5. Exactitud de detección

Este parámetro establece el límite a partir del cual una comparación de una imagen con un patrón se considera como detección o como no detección. Este valor puede variar entre 50 y 100% como se aprecia en la figura 3.9. Un valor muy elevado, puede ocasionar que no se reconozcan las piezas, mientras que un valor muy bajo, puede provocar errores en la detección. Tras las pruebas realizadas en el escenario disponible se establece como valor un 75% consiguiendo alta fiabilidad en los resultados.

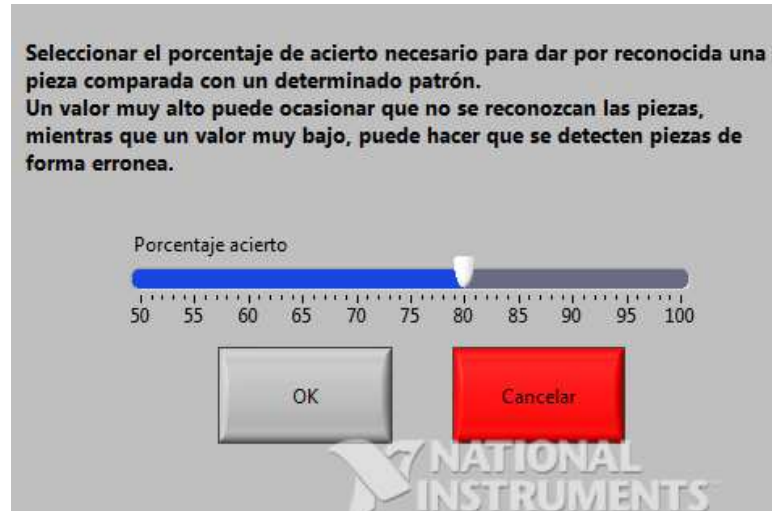


Figura 3.9. Proceso de elección de % de acierto.

3.6. Archivos de registro.

Se ha hecho referencia en un par de ocasiones a la existencia de una carpeta que contiene archivos como las imágenes de los patrones y la Tabla de Registros. Estos archivos y otros que se mencionan a continuación, son archivos que utiliza el SCADA para su correcto funcionamiento y son almacenados/accedidos en/desde la ubicación siguiente (también son incluidos en el CD anexo a esta memoria): **C:\Documents and Settings\alumno\Escritorio\Archivos SCADA.**

Cuando se produzca un cambio en algún parámetro del sistema, este deberá quedar almacenado para que en caso de pérdida de corriente, cuando se vuelva a arrancar, continúe trabajando con dicho parámetro. Los parámetros serán guardados en la ubicación mencionada siempre que se realice alguna modificación, mientras que el SCADA leerá estos archivos siempre que el proceso lo exija. Se tienen los siguientes archivos de registro:

- **Patrones:** patron1.bmp, patron2.bmp, patron3.bmp, patron4.bmp y patron5.bmp. Son los archivos que contienen las imágenes de los patrones que son comparadas por el sistema detector de patrones en el algoritmo de detección.
- **Dimensiones_patrones.xls:** Es un archivo que contiene las dimensiones de los rectángulos que contienen a los diferentes patrones. Este archivo se usa para obtener el rectángulo que se utiliza en la detección de patrones.
- **Rectangulo.xls:** Es el archivo que contiene los vértices del rectángulo utilizado en la detección de patrones, según los parámetros del sistema configurados. Se verá en detalle en el apartado 6.4.2.
- **Patrones.xls:** Es el número de patrones posibles a detectar configurados actualmente en el sistema.
- **Registro.xls:** Es el archivo en el que se guarda la Tabla de registros vista en el apartado 3.2.2.
- **Tmaq1:** Guarda los tiempos de trabajo en la Máquina 1 de los posibles patrones.
- **Tmaq2:** Guarda los tiempos de trabajo en la Máquina 2 de los posibles patrones.
- **Porcentaje_Color.xls:** Guarda el % de color (apartado 3.5.4) utilizado en la detección.
- **Porcentaje_Acierto.xls:** Guarda el % de acierto (apartado 3.5.5) utilizado en la detección.



3.7. Factores externos.

Debido a que el control del proceso va a depender de que un sistema detecte piezas mediante procesamiento de imágenes, va a ser fundamental conseguir que ciertos factores externos se mantengan constantes en todo momento, ya que la variabilidad de los mismos provocaría alteraciones en los parámetros definidos para la detección, dificultando el reconocimiento de la pieza. Principalmente a considerar 3 factores:

- **Iluminación:** Es fundamental conseguir que sea constante. Se colocará un foco sobre la zona de detección de las piezas, de forma que la imagen tomada en ese punto pueda ser captada, analizada y procesada determinando con gran fiabilidad la pieza contenida en dicha imagen.
- **Cámara en punto fijo.** La cámara deberá estar instalada en un punto fijo de modo que las piezas sean captadas en condiciones geométricas similares.
- **Fondo constante.** Se debe conseguir que el fondo que aparece en cada imagen sea lo más constante posible, variando únicamente la parte de la imagen donde se encuentra la pieza a reconocer. Será imperativo que no aparezcan en la imagen elementos indeseados que puedan interferir en la detección superponiéndose o creando sombras sobre la pieza a detectar.

El hecho de trabajar con una maqueta, hace difícil conseguir una estabilidad en estos aspectos. En caso de desarrollar el sistema de forma real, se deberían tomar estas tres consideraciones muy en cuenta para conseguir la máxima fiabilidad.

4. Sistema de proceso automático.

4.1. Autómata programable.

4.1.1. Introducción.

Llegados a este punto, nos vemos en la necesidad de definir brevemente lo que se conoce como autómata programable. Un autómata programable industrial es una máquina electrónica programable preparada (tanto en software como en hardware) para realizar, en ambiente industrial, automatismos combinatorios y secuenciales en tiempo real. Es entonces, el encargado de gestionar las acciones que se deben llevar a cabo en el proceso de forma automática.

4.1.2. Estructura de un autómata programable.

Un autómata programable consta (figura 4.1) de cuatro elementos fundamentales: Unidad de memoria, unidad de control, elementos de entrada y salida y buses de campo. [2]

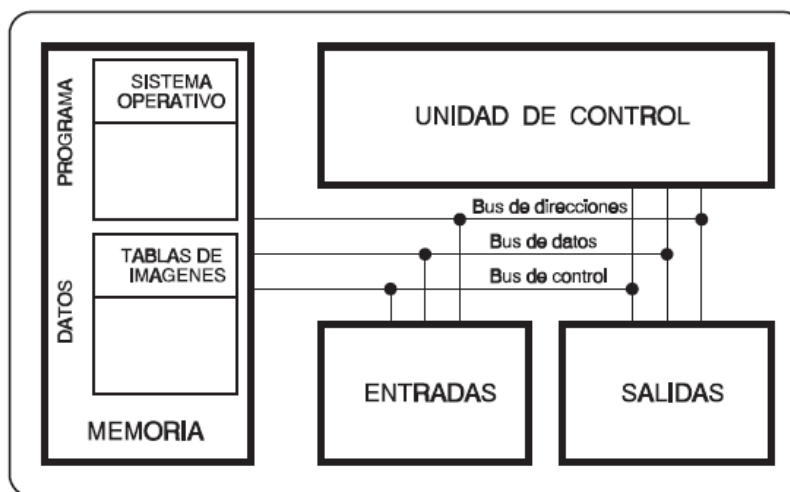


Figura 4.1. Elementos de un Autómata Programable.

Unidad de memoria: La memoria de un autómata programable sirve para almacenar el programa y los datos del proceso. Dentro de la memoria de datos tenemos una parte fija que es la tabla de imágenes de entradas y salidas; cuya medida viene ya definida mientras que el resto de la memoria de datos puede ser variable en función de las necesidades de cada programa. Pueden utilizar varios tipos de memoria que solo nombraremos: ROM, PROM, EPROM, EEPROM, etc.

Unidad de control: La unidad de control, también llamada CPU es la parte inteligente del autómata. Su función es ejecutar las instrucciones del programa. También se encarga de las comunicaciones con los equipos de programación y de la gestión de los estados de error. Su elemento base es el microprocesador.

La ejecución del programa sigue un ciclo llamado scan que consiste en:

0. Operaciones de Gestión del Sistema. Consiste en el tratamiento de las informaciones y de los bits de sistema. El tratamiento de las peticiones y las llamadas efectuadas por el terminal de programación y el envío de mensajes al terminal.
1. Lee las entradas y guarda sus estados en la tabla de imágenes de entrada.
2. Hace una ejecución del programa cogiendo los datos necesarios de la tabla de entradas, los contadores, los temporizadores, etc. y dejando lo que convenga en la tabla de salidas, contadores, etc.
3. Copia la tabla de imágenes de salida sobre las salidas.
4. Vuelve a empezar el ciclo leyendo las entradas.

Elementos de entrada y salida: Los elementos de entrada y salida son los que permiten comunicar el autómata con el proceso que está controlando y con el operador. Mediante los elementos de



entrada el autómata se entera del estado en que se encuentra el proceso a partir de los captadores que el diseñador ha situado para las señales que interesan. Los elementos de salida permiten que el PLC actúe sobre el proceso (electroválvulas, motores, pilotos, etc.).

Buses de comunicación

Son el medio físico a través del cual el procesador se comunica con el resto de elementos del sistema (entradas y salidas, memoria, periféricos). Hay normalmente tres buses:

- El bus de direcciones es por donde el procesador envía la dirección del elemento al que quiere enviar o que quiere que le envíe información. Esta dirección llegará a todos los elementos pero sólo tiene que haber un elemento que se identifique.
- El bus de datos es por donde todos los elementos enviarán los datos.
- El bus de control es aquel mediante el cual el procesador explica qué operación se está efectuando. Las operaciones más corrientes son leer y escribir.

4.1.3. Programación del autómata.

La programación del autómata consiste en el establecimiento de una sucesión ordenada de instrucciones, escritas en uno o varios lenguajes de programación entendidos por el PLC. Podríamos dividir el proceso de programación del autómata en varias etapas:

1. Definir el sistema de control (qué debe hacer, en qué orden, etc.): Redes de Petri
2. Identificar las señales de entrada y salida del autómata.
3. Asignar las direcciones de entrada/salida a las correspondientes del modelo.
4. Codificar la representación del modelo. Lenguaje de programación.
5. Cargar el programa en la memoria del autómata desde la unidad de programación.
6. Probar y depurar el programa

Existen varios tipos de lenguaje de programación **[3]**:

- Mnemónico o Lista de instrucciones (IL). Consiste en un conjunto de códigos simbólicos, cada uno de los cuales corresponde a una instrucción. Cada fabricante utiliza sus propios códigos y nomenclatura.
- Esquema de contactos (LD): Es un lenguaje gráfico, derivado del lenguaje de relés, que mediante símbolos representa contactos, solenoides... Su principal ventaja es que los símbolos básicos están normalizados según normas NEMA y son empleados por todos los fabricantes.
- Literal estructurado (ST): El lenguaje Literal estructurado permite realizar programas mediante la escritura de líneas de programación.
- Lenguajes de alto nivel
- GRAFCET: Permite representar gráficamente y de forma estructurada el funcionamiento de un automatismo secuencial mediante etapas y transiciones.

4.2. Maqueta utilizada.

Como se ha comentado, en varias ocasiones, vamos a simular un proceso industrial automatizado con una maqueta disponible en la EUPT. Podemos ver esta maqueta en la figura 4.2.

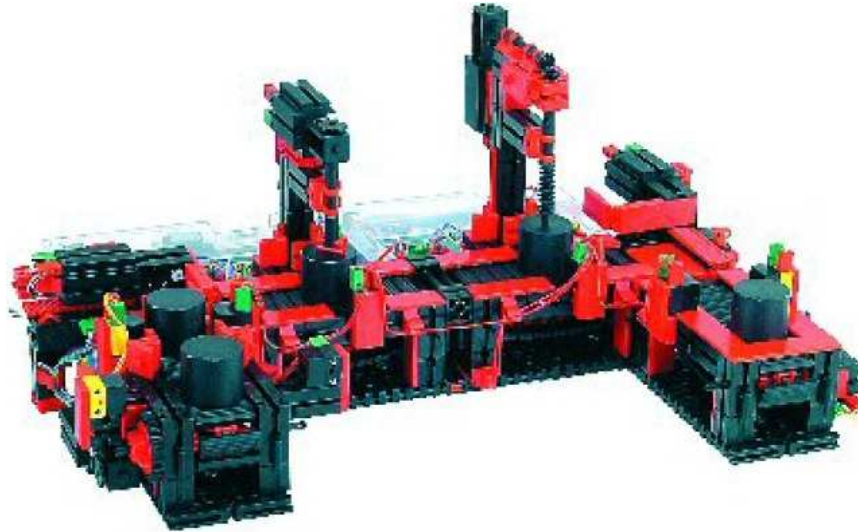


Figura 4.2. Maqueta utilizada.

Es una maqueta (dimensiones: 450x410x190m (LxWxH)) de marca Fischertechnik, que contiene una línea de fabricación con dos máquinas herramienta. Representa un sistema formado por varias cintas transportadoras, dispuestas en forma de U, para el transporte intermitente y para el mecanizado de piezas mediante dos máquinas. Concretamente, la maqueta contiene 2 estaciones de mecanizado, 4 cintas transportadoras, 8 motores DC (2 de ellos pueden invertir el sentido de giro), 4 finales de carrera y 5 detectores de presencia.

Tanto los sensores como los diferentes motores de la maqueta funcionan con tensión continua de 24v. Los sensores fotoeléctricos (PNP) son a 24v con tres conexiones, T1, T2 y T3. El sensor se alimenta entre los terminales T3 y T1, y cuando capta, se activa T2.

4.3. PLC. TSX 3722.

4.3.1. Características.

El modelo de PLC con el que se cuenta en la EUPT es el TSX 3722 de Telemecanique. Es un autómata modular, lo que le permite adaptarse a las necesidades de cada usuario, dentro de sus posibilidades, mediante la elección de los diferentes módulos con los que se puede configurar. Dispone de un rack con tres emplazamientos libres con alimentación integrada, un procesador con memoria RAM de 20Kpalabras, un reloj/calendario, admite ampliación de memoria y acoplador de comunicación, mediante tarjetas PCMCIA y puede ampliarse con un minirack de extensión.

Para su programación, previa instalación del driver en nuestro PC, dispone de dos conectores terminales RS 485, en formato mini-DIN 8 patillas, que permiten conectar respectivamente mediante la modalidad de comunicación UNI-TELWAY maestro de 9600 baudios, como se ve en la figura 4.3:

- TER: un terminal del tipo FTX o PC compatible, o conectar el autómata a los buses UNI-TELWAY o Modbus a través de la caja de aislamiento TSX P ACC 01.
- AUX: un terminal de diálogo con el operador o una impresora.

Esta información ha sido obtenida de [4] y [7]. Se recomienda su consulta para más información.

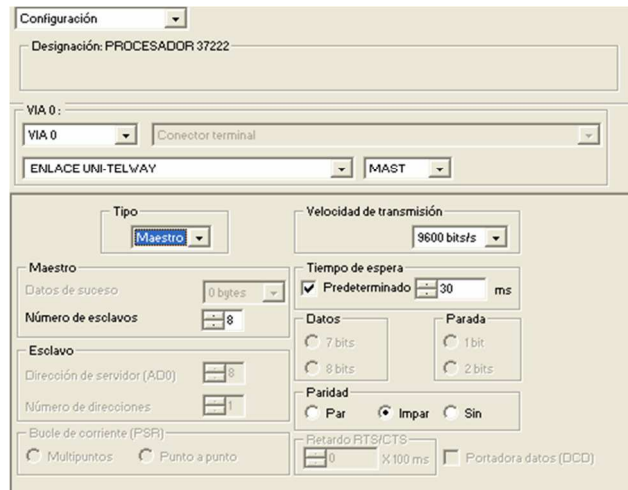


Figura 4.3. Configuración de Via0 para programación del PLC.

4.3.2. Tarjetas de E/S utilizadas.

Se utilizarán dos módulos digitales de E/S TSX DMZ 28DR cuya misión es la de transmitir, tanto las señales que recibe desde los sensores a la CPU del autómatas, como las señales que la CPU manda a los accionadores. Los módulos van ubicados en los racks 1 y 2 del autómatas, lo que hay que tener en cuenta a la hora de la configuración del hardware en el software de programación del autómatas.

El módulo TSX DMZ 28DR incluye 28 entradas/salidas repartidas como sigue:

- 16 entradas de 24 VCC, de lógica positiva tipo 1 o lógica negativa
- 12 salidas de relés.

Esta información ha sido obtenida de [4]. Se recomienda su consulta para más información.

En la figura 4.4 se tiene la configuración de entradas/salidas que se ha realizado para este proyecto.

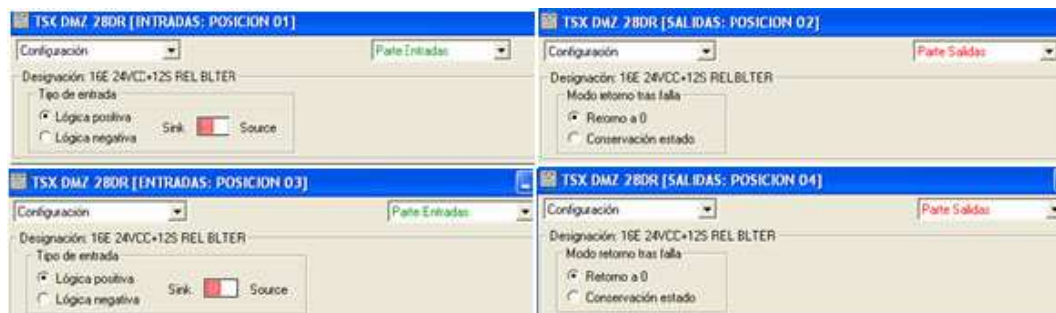


Figura 4.4. Configuración de tarjetas TSX DMZ 28DR.

Observando la figura 4.4, se observa que hay cuatro posiciones aunque solo se ha hablado de dos tarjetas. Esto es porque para este PLC, la modularidad de base es de medio formato, los módulos de formato estándar se direccionan como 2 módulos de medio formato superpuestos, como se observa en la figura 4.5.

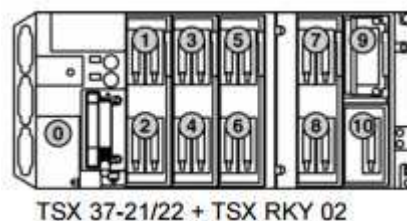


Figura 4.5. Configuración de módulos TSX 37-22.

Además de estos módulos, se utilizarán las Entradas/Salidas analógicas integradas en los autómatas TSX 37-22. Incluyen 8 entradas y 1 salida 0-10 V 8 bits, así como una salida de referencia de 10 V de tensión. El acceso a la interfaz analógica se efectúa a través de un conector SubD de 15 patillas con el esquema de conexión mostrado en la figura 4.6.

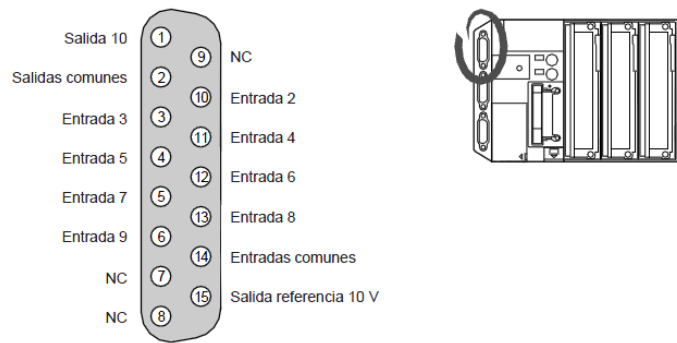


Figura 4.6. Esquema de conexión Entradas/Salidas analógicas integradas TSX 37-22.

Es importante tener en cuenta estos pines para la conexión con la DAQ y la programación de las distintas señales de comunicación que se verán en el apartado 5.4. Esta información ha sido obtenida de [4] y [7]. Se recomienda su consulta para más información.

4.3.3. Fuente de alimentación.

La fuente de alimentación interna del autómatas (24 VDC), es capaz de suministrar la corriente necesaria para alimentar a todos los captadores y los motores de la instalación. En el caso de que la instalación fuera real, deberíamos asegurarnos de que dicha fuente (400 mA) es suficiente para alimentar a los elementos de la instalación o utilizar en su defecto otra u otras fuentes de alimentación que si lo sean. Los bornes de la fuente de alimentación interna del autómatas, irán conectados a los bornes 18 (0VDC) y a los bornes 17 (+24VDC) de ambos módulos TSX DMZ 28DR, de modo que las entradas de ambas tarjetas y los sensores que a ellas se conectan, quedarán alimentadas.

Respecto a las salidas, los comunes de la primera tarjeta que alimentan los motores de la maqueta también irán conectados a dicha fuente de alimentación, mientras que los de la segunda tarjeta irán conectados a la alimentación de la DAQ como se verá en el apartado 5.3.5.

4.3.4. Software de programación. PL7 Pro.

El programa PL7 Pro es un software de programación, diseñado entre otros para los autómatas TSX 3722. Solo funciona en Windows XP y anteriores S.O. Este software, permite crear bloques DFB, pantallas de explotación y módulos funcionales. El programa PL7 dispone de un lenguaje gráfico (LD), un lenguaje booleano (LI), un lenguaje literal estructurado (ST) y un lenguaje GRAFCET. Para más detalle se puede consultar la documentación del fabricante [8].

4.3.5. Funcionamiento. Procesador monotarea.

EL TSX 3722 tiene la capacidad de funcionar mediante estructura monotarea o multitarea. En nuestro caso, se realizará la programación mediante estructura monotarea, como vemos en la configuración mostrada en la figura 4.7.a. Solo hay una tarea, la Tarea Maestra (MAST). Esta tarea se ejecutará de manera cíclica (funcionamiento predeterminado) de forma que las ejecuciones de la tarea se encadenan una tras otra, sin tiempo de espera como se ve en la figura 4.7.b

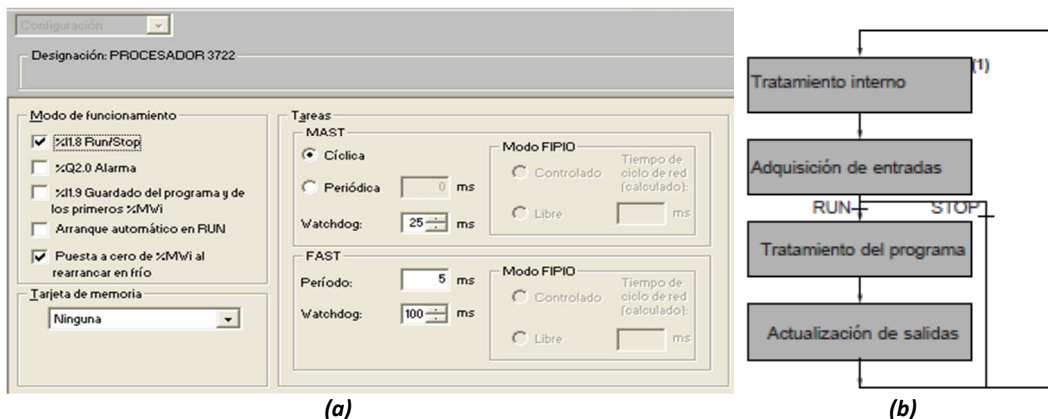


Figura 4.7 Procesador 3722. (a) Configuración procesador. (b) Ejecución de tarea MAST.

Destacar que en el Modo de funcionamiento se ha elegido la opción de poner el PLC en %I1.8 RUN/STOP. Tal y como indica la figura 4.7.b, el tratamiento del programa solo se ejecutará en el caso de que el PLC esté en RUN. Por ello, se va a poder controlar desde el SCADA la puesta a RUN/STOP del PLC mediante una entrada externa.

Las tareas de un programa PL7 se componen en varias partes denominadas secciones y subprogramas. Cada una de estas secciones puede programarse en el lenguaje apropiado al tratamiento que se desea realizar. En nuestro programa, se ha utilizado una única sección programada en Grafcet como se ve en la figura 4.8. Nuestra sección Grafcet comprende:

- El tratamiento preliminar (PRL) programado en ST. Se ejecuta antes de Grafcet.
- El Grafcet (CHART): en las páginas Grafcet, se programan las receptividades asociadas a las transiciones y acciones asociadas a las etapas y/o macroetapas.
- El tratamiento posterior (POST) programado en LD. Se ejecuta después de Grafcet,
- Un subprograma SRi (podría haber desde $i = 0$ a 253) programado en ST. Los módulos de subprograma, se programan como entidades separadas y las llamadas a los subprogramas se efectúan en el tratamiento preliminar (como es nuestro caso), posterior, en las acciones asociadas a las etapas o desde otros subprogramas. Los subprogramas se asignan a una tarea; sólo pueden llamarse desde una misma tarea.

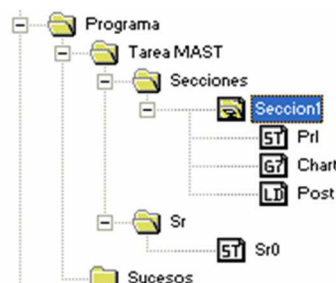


Figura 4.8. Árbol de configuración de nuestro programa.

Si se desea conocer más información sobre este software y sobre las distintas instrucciones y funciones, se aconseja leer el manual de referencia que proporciona el fabricante [8].

4.4. Control de Maqueta mediante PLC.

4.4.1. Definición de Entradas y Salidas en el PLC.

En la tabla 4.1 se detallan las entradas y salidas que se han cableado a los distintos captadores y actuadores de la maqueta:

VARIABLE	SÍMBOLO	COMENTARIO
%I1.0	Isel1fin	Selector 1 en posición final
%I1.1	Isensorc1	Sensor presencia final cinta 1
%I1.2	Isensorcarga	Sensor presencia de nuevo producto
%I1.3	Isensormaq1	Sensor presencia máquina 1
%I1.4	Isensormaq2	Sensor presencia máquina 2
%I1.5	Isel1ini	Selector 1 en posición inicial
%I1.6	Isensorc4	Sensor presencia final cinta 4
%I1.7	Isel2fin	Selector 2 en posición final
%I1.11	Isel2ini	Selector 2 en posición inicia
%Q2.0	Qsel1avanza	Selector 1 avanza
%Q2.1	Qmaq1	Maquina 1 trabaja
%Q2.2	Qcinta3	Cinta 3 avanza
%Q2.3	Qsel1retro	Selector 1 retrocede
%Q2.4	Qsel2avanza	Selector 2 avanza
%Q2.5	Qcinta1	Cinta 1 avanza
%Q2.6	Qcinta2	Cinta 2 avanza
%Q2.7	Qsel2retro	Selector 2 retrocede
%Q2.8	Qmaq2	Maquina 2 trabaja
%Q2.10	Qcinta4	Cinta 4 avanza

Tabla 4.1. Entradas y Salidas de la maqueta.

4.4.2. Estructura de la aplicación en PL7.

A continuación se explica la estructura que se ha seguido para la programación del sistema con el software PL7. El programa de la tarea maestra se divide en tres módulos:

Pri (ST): Se ha programado la gestión de paradas de emergencia y rearme. También contiene instrucciones necesarias para la comunicación entre el PLC y el SCADA.

Chart: Se ha utilizado Programación en GRAFCET para la implementación del modelo de Petri que representa el ciclo secuencial del proceso. En las diferentes etapas del Grafcet, se han podido ejecutar instrucciones programadas en lista de instrucciones o lenguaje de contactos. Como veremos en siguientes apartados, tendremos varios Grafcet que representarán las diferentes etapas que se van sucediendo en el proceso, todos ellos gobernados por un Grafcet de control y mando.

Post (LD): Se ha programado la activación de las salidas del autómeta, tanto las salidas que van a actuadores de la maqueta, como las señales de comunicación que manda el PLC al SCADA (se detallarán en el apartado 5.3.5).

Sr0 (ST): Se llama a este subprograma al inicio del Pri, para realizar una lectura de señales de entrada internas nada más realizar la lectura de señales procedentes de las señales externas de acuerdo a la ejecución de la tarea en el PLC mediante el diagrama visto en la figura 4.7.b. Todas estas señales, son señales de comunicación enviadas desde el SCADA al PLC. Se decide ejecutar este subprograma antes de tratar el resto del programa para que todas las entradas hayan sido actualizadas antes de comenzar a ejecutar el programa del proceso.

4.4.3. GRAFCET de Producción normal automática.

A continuación se va a pasar a explicar la programación del PLC para ejecutar de forma automática el proceso de tratamiento de las piezas.

4.4.3.1. Grafos.

El proceso se ha implementado mediante 6 Grafos comunicados entre sí, buscando la optimización de tiempos de tratado de piezas, de forma que los recursos (máquinas, cintas, etc) son liberados siempre que la situación lo permita para poder ser utilizados por otra pieza que tenga que hacer uso de ellos. A continuación se comentan estos grafos y las funciones que cada uno realiza.

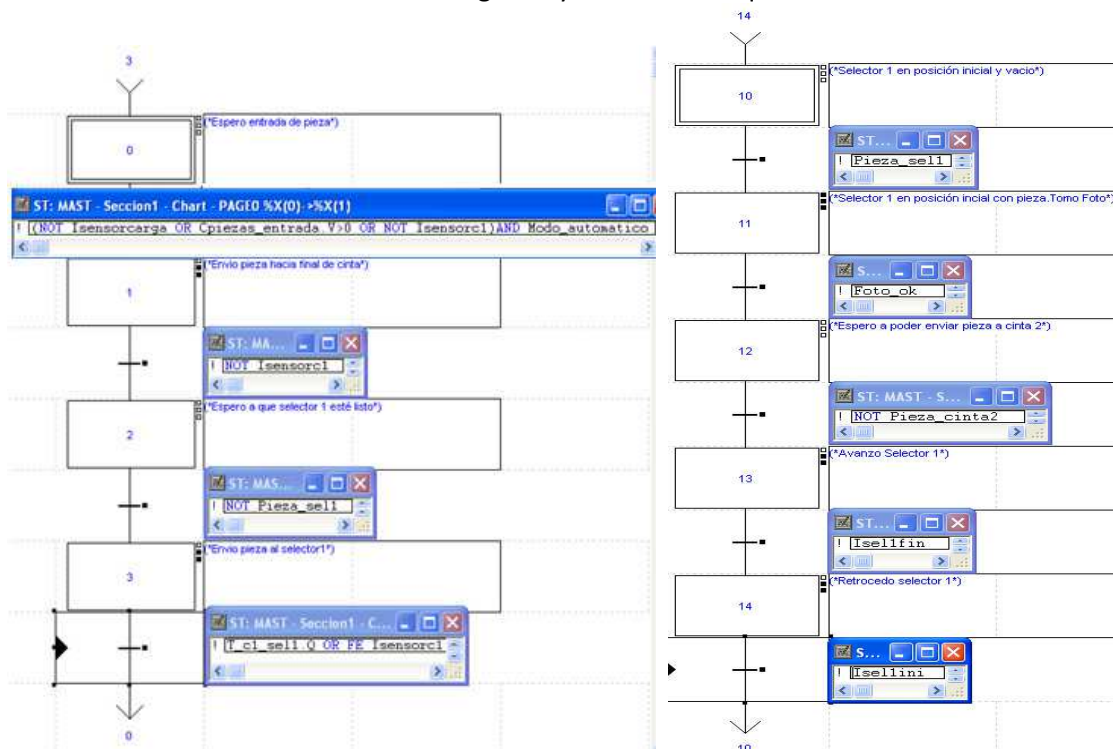


Figura 4.9. Grafos 1 y 2.

La figura 4.9 muestra los dos primeros grafos. En el primero de ellos se produce la entrada de la pieza en el proceso y su traslado hasta el Selector 1. El segundo de ellos, contempla el desplazamiento de la pieza por el Selector 1 hasta la Cinta 2. Destacamos la etapa 11 en la que se realiza la detección del patrón y se reciben del SCADA los tiempos que se utilizarán para el procesamiento en las Máquinas.

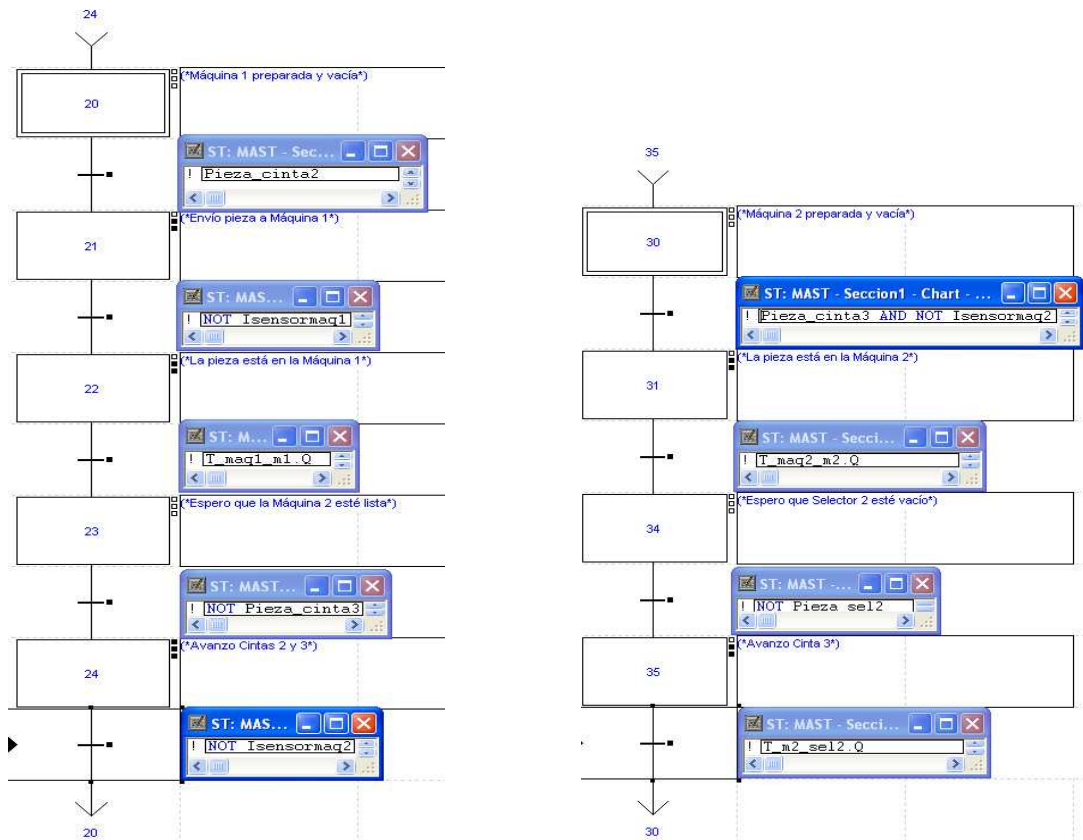


Figura 4.10. Grafos 3 y 4.

La figura 4.10 muestra el tercer y cuarto grafo. El tercero detalla el proceso desde que la pieza llega hasta la Cinta 2 hasta que es llevada a la Máquina 2 tras haber sido realizado el trabajo pertinente en la Máquina 1. El cuarto, implementa el proceso desde que la pieza llega a la Máquina 2 hasta que, una vez se realiza el trabajo pertinente en dicha máquina, se envía al Selector 2.

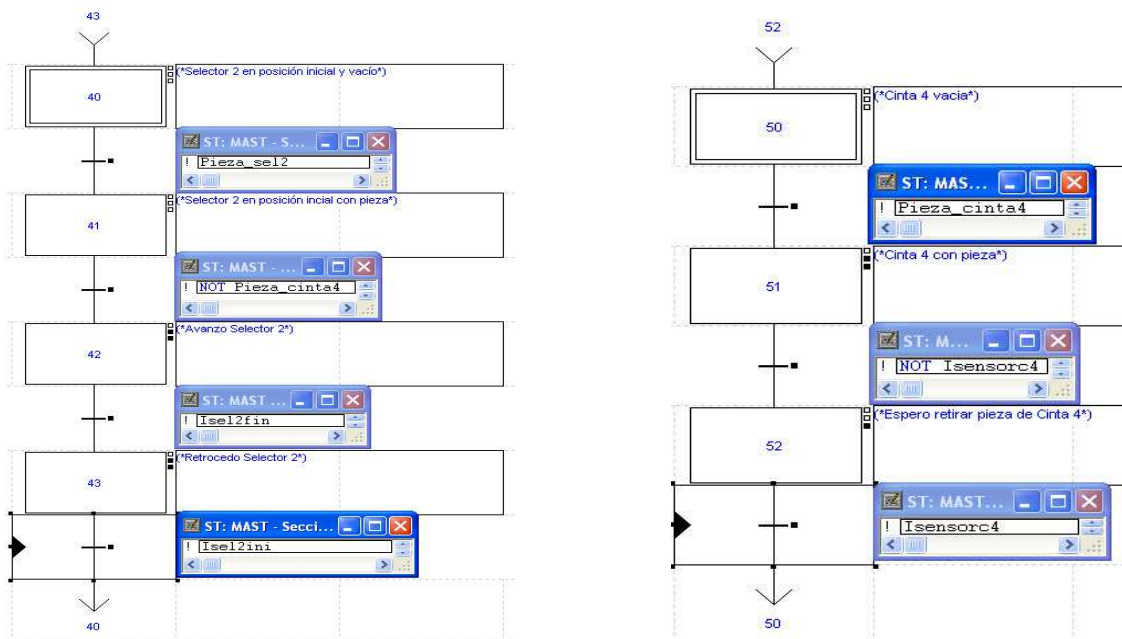


Figura 4.11. Grafos 5 y 6.

La figura 4.11 muestra el quinto y sexto grafo. El quinto implementa el proceso desde que la pieza llega al Selector 2 hasta que es enviada a la Cinta 4. Mientras que el sexto, detalla el proceso desde que la pieza llega a la Cinta 4, hasta que es sacada del sistema por el usuario cuando llega al final de dicha cinta.

4.4.3.2. Habilitaciones.

Para conseguir una utilización óptima de los recursos, se ha realizado un sistema de habilitaciones de forma que solo pueda mandarse la pieza al recurso siguiente cuando éste esté preparado para recibirla. A continuación se analizan las señales habilitantes utilizadas.

- “Modo_automático”. Esta señal puesta a 1 indica que el Grafset de conducción se encuentra en modo Automático. De esta forma, si el sistema no se encuentra en este modo, nunca se dará la transición entre etapas 0 y 1 que activa el Grafo 1.
- “Pieza_sel1”. Esta señal puesta a 0 indica que el Selector 1 está disponible para recoger una pieza enviada por la Cinta1. Se pone a 1 al finalizar la etapa 3 (la pieza está en el Selector1) y se vuelve a poner a 0 al finalizar la etapa 14 (el Selector 1 acaba de volver a su posición inicial tras entregar pieza a Cinta 2)
- “Pieza_cinta2”. Esta señal puesta a 0 indica que la Cinta 2 no tiene pieza. Se pone a 1 al finalizar la etapa 13 (el Selector 1 ha entregado la pieza a la Cinta 2) y se vuelve a poner a 0 al finalizar la etapa 24 (la pieza está en la Maquina 2).
- “Pieza_cinta3”. Esta señal puesta a 0 indica que la Maquina 2 no tiene pieza. Se pone a 1 al finalizar la etapa 24 (la pieza está en la Maquina 2) y se vuelve a poner a 0 al finalizar la etapa 35 (la pieza está en el Selector 2).
- “Pieza_sel2”. Esta señal puesta a 0 indica que el Selector 2 está disponible para recoger una pieza enviada por la Cinta3. Se pone a 1 al finalizar la etapa 35 (la pieza está en el Selector2) y se vuelve a poner a 0 al finalizar la etapa 43 (el Selector 2 acaba de volver a su posición inicial tras entregar pieza a Cinta 4).
- “Pieza_cinta4”. Esta señal puesta a 0 indica que la Cinta 4 no tiene pieza. Se pone a 1 al finalizar la etapa 42 (la pieza ha sido enviada por el Selector 2 a la Cinta 4) y se vuelve a poner a 0 al finalizar la etapa 52 (se retira pieza de la Cinta 4).

Para que una habilitación pueda ser activada, antes ha tenido que habilitarse la señal de habilitación previa, en el orden que se han descrito.

4.4.3.3. Contador de piezas.

Debido a que el sistema acepta la introducción de varias piezas a la vez, se ha configurado un contador que se incrementará en una unidad cada vez que se esté en las etapas 1 ó 3 (Cinta 1 activada) sin estar en situación de parada tras emergencia o rearme y se detecte un flanco negativo del detector “Isensorcarga”. En la figura 4.12 se contempla la programación en LD en las mencionadas etapas.



Figura 4.12. Programación en LD para contador de piezas en Cinta 1.

Este contador permite al sistema reconocer la existencia de piezas en la Cinta 1 introducidas mientras la Cinta 1 envía otra pieza al Selector 1. Cada vez que se entrega una pieza al Selector1, se decrementa el contador en una unidad.



No es necesaria la configuración de más contadores en el sistema, ya que en el resto de recursos solo se permite la existencia de una pieza en cada momento.

4.4.3.4. Tiempo de trabajo de Máquinas.

Un parámetro importante del sistema, es el tiempo que trabajará cada una de las máquinas en función del tipo de pieza detectada en la etapa 11. Es en esta etapa, donde se realiza la toma de la imagen y el procesado de la misma determinando el tiempo que tendrá que trabajar cada máquina sobre la pieza en particular. En la etapa 11 se tiene programado en ST las siguientes sentencias:

```
T_maq1_sel1.P:=Señal_mando1/1000;  
T_maq2_sel1.P:=Señal_mando2/1000;
```

Es decir, programamos el Preset de dos temporizadores con el valor de tiempo enviado por el SCADA en función de la pieza que ha sido detectada (parámetro establecido como se vio en el apartado 3.5.3). Vemos que las señales enviadas son divididas entre 1000 debido a que la señal del SCADA viene dada en mV, de forma que 1000 mV representarán un segundo.

Aunque a partir de esta etapa ya tenemos en el sistema los tiempos que trabajarán las máquinas para la pieza detectada en cuestión, es necesario hacer uso de variables auxiliares de forma que estas variables de tiempos no machaquen ni sean machacadas por otros tiempos de otras piezas anteriores o posteriores.

De este modo, cuando la pieza haya llegado a la Cinta 2 (Fin de la etapa 13), se realiza una transmisión de los tiempos a otros temporizadores como vemos a continuación:

```
T_maq1_m1.P:=T_maq1_sel1.P;  
T_maq2_m1.P:=T_maq2_sel1.P;
```

El primero de ellos, será el tiempo que se utilizará para trabajar en la Máquina 1, mientras que el segundo de ellos, volverá a transmitirse en la etapa 24 (una vez que me aseguro que no hay pieza en la Máquina 2) a otro temporizador $T_maq2_m2.P:=T_maq2_m1.P$ que finalmente será el que utilizaremos como variable de trabajo para la Máquina 2.

4.4.4. GRAFCET de conducción.

Además de los grafos explicados en el punto anterior, es necesario un Grafcet de conducción que gestionará los distintos modos de funcionamiento del proceso vistos en el apartado 3.2.3.

La figura 4.13 muestra este Grafcet de Conducción o Mando. En dicha figura se observa que cada modo de funcionamiento del sistema visto en el apartado 3.2.3 (salvo la Etapa 70 y 71 que se corresponden con modo Inicial) está representado por una etapa de este Grafcet de Conducción:

- Etapa 70 (%X70). Modo Inicio
- Etapa 71 (%X71). Estado inicial sistema (Estado previo al acceso a estado Automático)
- Etapa 72 (%X72). Modo Manual
- Etapa 73 (%X73). Modo Posicionamiento
- Etapa 74 (%X74). Modo Automático
- Etapa 75 (%X75). Modo Fin de Ciclo
- Etapa 76 (%X76). Modo Visión

Se ha añadido esta imagen ampliada para su consulta en la figura A.9 del apartado A.3 del anexo A.

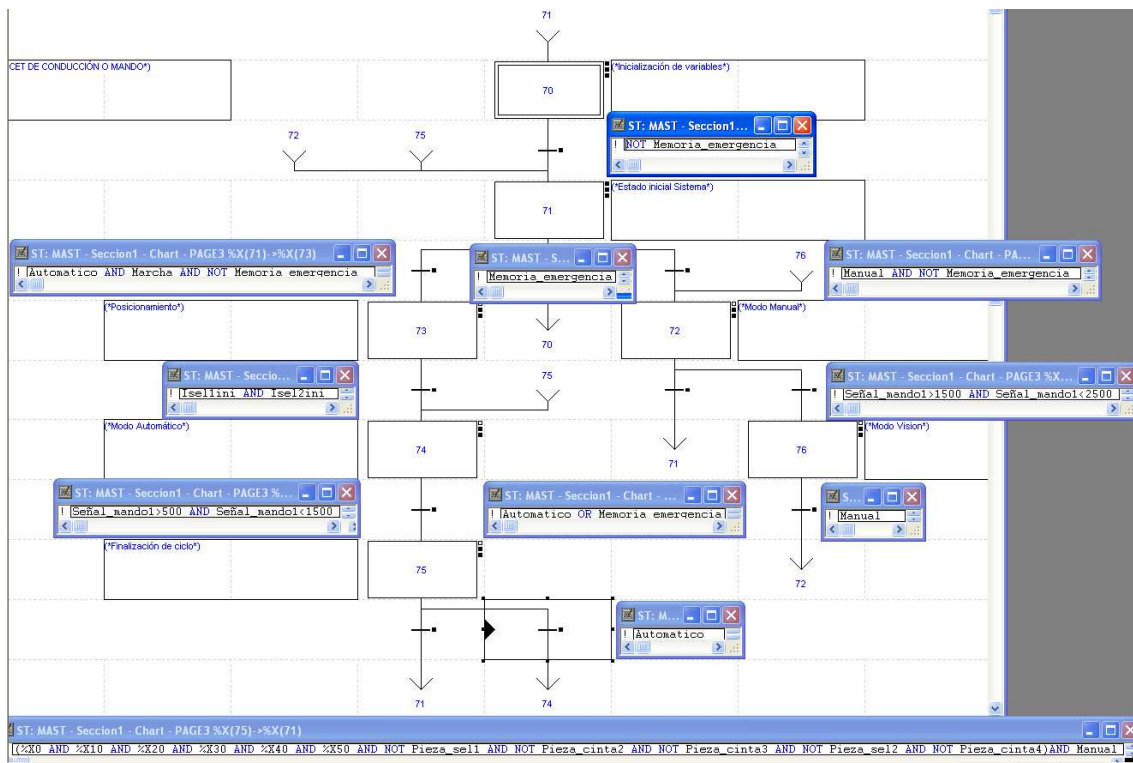


Figura 4.13. GRAFCET de Conducción o Mando.

4.4.5. Implementación de guía GEMMA.

La Guía GEMMA, Guía para el estudio de los modos de Marchas y Paradas, es una representación organizada de todos los modos o estados en que se puede encontrar un proceso de producción automatizado. También representa los saltos o transiciones que se dan de un estado a otro. Estos estados se organizan en tres grupos principales [3]:

- **Procesos de Parada y Puesta en Marcha (A):** Engloba los procesos de parada activados a petición del operador y los procesos conducentes a la puesta en marcha del proceso.
- **Procesos de fallo de la Parte Operativa (D):** Engloba los procesos de fallo, activados por un fallo propio en el proceso o a petición del operador al pulsar la seta de emergencia.
- **Procesos de funcionamiento (F):** Designa los procesos necesarios para la producción y obtención de productos.

Como bien dice su propio nombre, esto es una guía y para cada aplicación concreta se utilizan los estados necesarios. A continuación, se especifica una representación organizada como se aprecia en la figura 4.14 de todos los modos o estados en que se puede encontrar el proceso de producción automatizado que nos atañe.

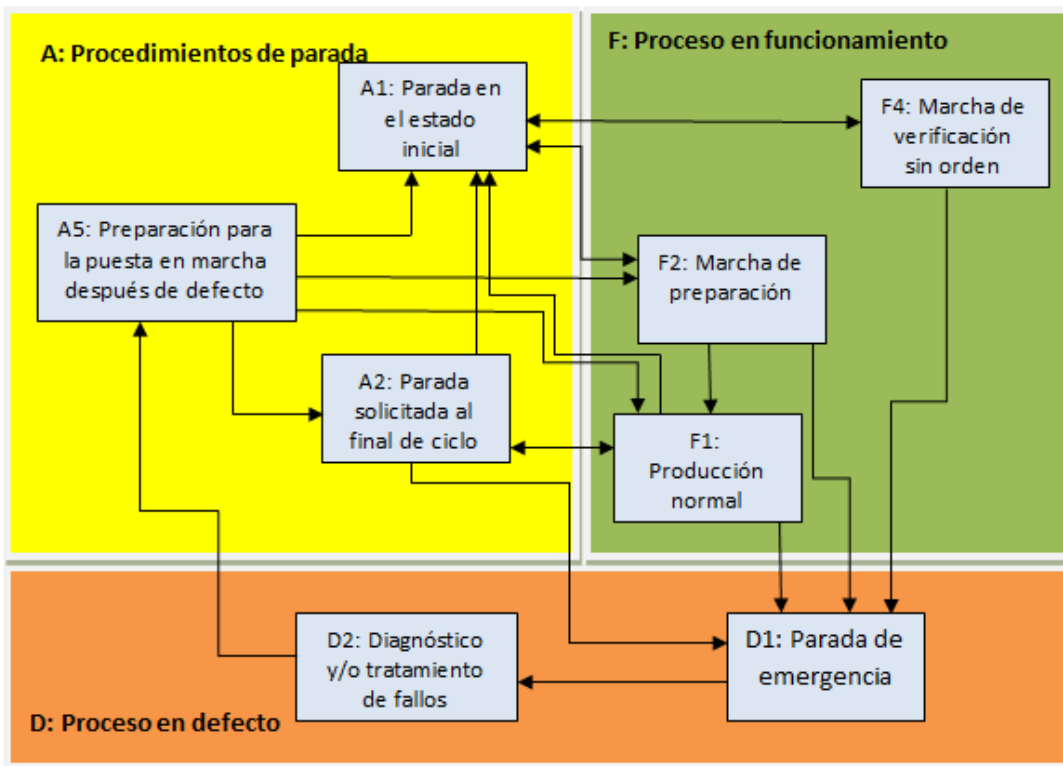


Figura 4.14. Guía GEMMA del proceso.

Parada en el estado inicial (A1): Es el estado normal de reposo del proceso. Se corresponde con la etapa inicial de los Grafsets. Vemos que desde aquí se puede ir a F4, que representa si el sistema está en modo Manual (%X72) y a F2, que representa la situación en que el sistema está en modo Posicionamiento (%X73), previo al modo Automático (%X74), F1.

Parada solicitada al final de ciclo (A2): Es un estado transitorio (%X75) en el que el proceso, que hasta ese momento estaba produciendo normalmente en modo Automático (%X74), produce sólo hasta acabar el ciclo. El proceso, se activará al colocar el selector S2 en "Manual". Se tratará hasta la última pieza que se introduzca en el sistema antes de solicitar la parada mencionada. Los Grafsets se irán colocando en sus posiciones iniciales a medida que vayan terminando sus respectivas funciones. Cuando todos estén en dichas posiciones, se pasará al estado A1 de la figura. Antes de que acabe el ciclo se podrá volver al estado F1 sin más que colocar de nuevo el selector de S1 en "Automático".

Preparación para la puesta en marcha después de defecto (A5): En este estado se deben realizar las acciones necesarias para corregir los fallos o defectos que han generado una parada de emergencia. El autómatá dará información sobre el motivo que ocasionó la parada. Una vez solucionado el motivo de la parada, el operador elegirá cómo se reinicia el proceso. En nuestro sistema, podremos continuar el proceso en el punto en qué se generó la parada, o rearmar el sistema a su posición inicial.

Parada de emergencia (D1): En este estado se lleva al proceso a una situación de seguridad (parada de accionadores) para la instalación y los operarios. Se producirá desde cualquier modo en que se encuentre el proceso siempre que se ocasione una situación de parada de emergencia. Existen varios motivos de parada que se explican en el apartado 4.4.6. Siempre que se produzca una parada de este tipo será necesario un rearme para continuar el proceso.

Diagnóstico y/o tratamiento de los defectos (D2): Se ha programado el sistema para que en todo momento, el autómatá indique el estado en que se encuentra el proceso y en caso de parada de emergencia, cual puede ser la causa de dicha parada.

Producción normal (F1): El estado de producción normal es el funcionamiento automático (%X74). El sistema se encontrará en el estado de producción normal cuando esté activado el modo Automático y el proceso transcurra sin paradas. Se introducen piezas por la cinta 1 y el sistema se encarga de realizar su procesamiento con las características programadas.



Marcha de preparación (F2): Son las acciones necesarias para que el sistema pase F1. En nuestro caso, antes del comienzo del ciclo automático, es necesario que los selectores se encuentren en posición inicial y no haya piezas ubicadas en el sistema (%X73).

Marchas de verificación sin orden (F4): El sistema está en modo Manual (%X72) y el operador por medio de mandos del panel de control, puede hacer mover los elementos del sistema.

4.4.6. Paradas de emergencia.

Como se ha mencionado en el apartado anterior, el sistema será capaz de realizar paradas controladas aportando seguridad para la instalación y para los usuarios, cuando se produzcan determinados eventos. Los posibles eventos que ocasionarán parada serán:

- F_tarjetas: Se detecta fallo en alguna tarjeta del PLC.
- F_magneto: Se detecta disparo de algún magnetotérmico de los motores. Éstos, han sido simulados mediante bits de memoria, pero en caso del proceso real, deberían ser cableados realmente al PLC.
- F_tiempos: Se produce un tiempo excesivo de funcionamiento de cintas o selectores.
- F_sitológicas: Se produce alguna situación incoherente en el proceso.
- F_comunicacion: Se detecta pérdida de comunicación entre el PLC y el SCADA.
- F_rearraque_caliente. La alimentación se recupera sin pérdida de contexto.
- F_Arraque_frio. La alimentación se recupera con pérdida de contexto.
- Seta_emergencia: Alguien pulsa la Seta de Emergencia del panel de control.
- Error_foto: Se genera algún fallo a la hora de realizar una fotografía.

Cada vez que se produzca alguna circunstancia de las mencionadas, se producirá la parada y se activará un bit ("**Memoria_emergencia**") que indicará que se ha producido una parada. Aunque el motivo de la parada haya desaparecido, hasta que no se produzca un rearme activo del sistema, no se desactivará este bit.

Es importante mencionar que si se produce una parada en uno de los estados automáticos (etapas 73, 74 ó 75 de apartado 4.4.4), se realizará una congelación de los GRAFCET mediante activación del bit de sistema %S23. De esta manera, una vez corregido el motivo de la parada, se podrá continuar con el proceso en el mismo punto que se encontraba el proceso cuando se produjo la parada si así se desea.

Por otro lado, el PLC proporcionará información al usuario sobre el motivo de la parada mediante una señal de control que comunicará con el SCADA. Se ha realizado la programación de forma que, en caso de darse varios motivos de parada, se indicará el que se haya programado como más prioritario. De esta manera, circunstancias de parada menos importantes no enmascaran otras más importantes, ya que el sistema solo es capaz de mostrar un motivo de parada en cada momento.

En el anexo A.1 se muestra la programación, realizada en PRL de todo lo referente a paradas de emergencia.

4.4.7. Gestión de rearmes.

Se podrá rearmar el sistema, siempre de forma activa por parte del usuario, de dos maneras distintas:

- En caso de estar trabajando en uno de los modos automáticos, se podrá volver a la situación previa al motivo de la parada realizando una pulsación del mando "Rearme" durante menos de 5 segundos, una vez resuelto el motivo de la parada. Esto, provocará el reseteo de la variable que almacenaba que se había producido una parada (Memoria_emergencia) y activará un bit que indicará que se ha rearmado el sistema. Cuando se ha producido el rearme, será necesario pulsar marcha para continuar con el proceso, momento en el que se descongelarán los Grafsets (RESET %S23).
- Si se realiza una pulsación larga (más de 5 segundos) de "Rearme" estando en modos automáticos, o se realiza cualquier tipo de pulsación en el resto de modos, se producirá un

rearme del sistema a la posición inicial. En este rearme, se reinician los GRAFCETs mediante el bit de sistema %S21 reseteando las variables utilizadas en los procesos automáticos.

La programación de todo lo correspondiente a rearmes, podrá consultarse en el apartado anexo A.1 del anexo A que contiene la programación del módulo PRL.

4.4.8. Programación de Salidas.

Como ya hemos comentado, las salidas se programarán en el POST. Se han programado en el lenguaje de diagrama de contactos (LD), debido a que con el gran número de condiciones que las habilitan, resulta mucho más sencillo y ordenado de controlar que si se hubieran hecho en ST mediante paréntesis que controlaran la lógica de las mismas.

Se utilizarán estados determinados del Grafcet para activar las salidas en cada caso. En el apartado A.4 del anexo A, se detalla la programación en el PLC de todas las salidas. Todas ellas han sido programadas siguiendo la metodología que se comenta a continuación para una de ellas (figura 4.15).

Para que se active una salida, existen dos posibilidades:

- Estar en la etapa concreta asociada a la activación de la salida (%X13), sin que se haya producido un rearme de memoria (se ha rearmado el sistema pero todavía no se ha pulsado marcha para proseguir).
- Estar en modo Manual (%X72) y pulsar el mando que mueve el dispositivo.

Se incorporan medidas de seguridad como cortar el circuito si se llega al final de carrera, si hay parada de emergencia (común en todas las salidas cableadas a la maqueta), si se pulsa otro mando manual incompatible con la acción que supone, etc.



Figura 4.15. Programación de una salida en el POST.

5. Comunicación de dispositivos.

5.1. OPC vs DAQ.

Como se ha visto en el diagrama de bloques del apartado 3.4, se tienen varios dispositivos a comunicar de forma eficaz para el funcionamiento correcto del sistema.

En un principio, se pensó en comunicar el PLC con Labview, mediante el uso de un servidor OPC gracias a las capacidades proporcionadas por el módulo DSC de Labview y la disponibilidad en el mismo de drivers de nuestro PLC para realizar esta conexión. Sin embargo, Telemecanique no permite esta conexión mediante el puerto de comunicación disponible en nuestro PLC (TER). Para esta conexión sería necesario el uso de algún módulo de comunicación que lo permitiera, como pudiera ser el TSX ETZ510.

Esto, hizo que la alternativa de comunicación desarrollada se fundamente en una tarjeta de adquisición de datos que hará de interfaz entre las E/S del PLC y el SCADA. Esta solución es mucho menos adecuada que la que se hubiera conseguido con el uso de un servidor OPC por los motivos que se exponen a continuación, en la tabla 5.1:

OPC	DAQ
Mediante un único cable de comunicaciones se transmiten todas las señales	Cada señal necesita una conexión física
Conexiones ilimitadas.	Limitación de número de conexiones
Un único punto de fallo	Múltiples puntos de fallo
Alta velocidad a través del cable de comunicación.	Baja velocidad de comunicación.
Simplicidad en la conexión.	Gran complejidad en la conexión

Tabla 5.1. Ventajas de comunicación mediante OPC respecto a DAQ.

En caso de desarrollar un proyecto de estas características en la vida real, sería imperativo el uso de una comunicación mediante OPC o similar.

5.2. Prestaciones y limitaciones de comunicación.

El uso de DAQ, obliga a buscar soluciones no adecuadas en el caso de implementar esta solución en un caso real. A continuación se resumen las prestaciones y limitaciones que se van a tener en la comunicación con los medios disponibles.

5.2.1. Prestaciones de comunicación de la DAQ.

Se utilizará una tarjeta de adquisición de datos (DAQ) que conectada al PC es utilizada para obtener señales analógicas y digitales de un elemento externo al ordenador (PLC) y transmitir señales analógicas y digitales generadas por el PC (SCADA) al exterior (PLC). La tarjeta a utilizar es una tarjeta Serie-M de National Instruments (PCI- 6221) y se utilizará el DAQ Assistant Express del Labview para configurar los canales virtuales a utilizar.

Se va a obviar la forma de configurar los diferentes puertos de la DAQ. Simplemente, comentar que esta tarjeta dispone de 2 salidas analógicas, 16 entradas analógicas (AI0...AI15) y 24 E/S digitales (Logiclow=0-0.8 v y Logic High=2-5 v), de las que se han utilizado las siguientes:

- Salidas Analógicas: 2 (AO0 y AO1). En el rango de 0 a 10v.
- Entradas Analógicas: 1 (AI0). En el rango de 0 a 10v.
- Entradas Digitales: 11 (P0.2,..., P1.4).
- Salidas Digitales: 6 (P0.0, P0.1, P1.5, P1.6, P1.7, P2.0).

Para más información se puede consultar la información del fabricante [9].



5.2.2. Prestaciones de comunicación del PLC.

Como se ha comentado en el apartado 4.3.2, se va a disponer de las E/S analógicas integradas en el TSX 3722, además de las E/S disponibles en las tarjetas. Para la comunicación con el SCADA a través de la DAQ, se van a utilizar:

- 7 Entradas analógicas. En el rango de 0 a 10v.
- 1 Salida analógica. En el rango de 0 a 10v.
- 2 Entradas digitales adaptadas mediante relés.
- 11 Salidas digitales. Señales de 0 (Low) y 5 v (High). El común conectado a 5v.

Para más información sobre los relés se puede consultar documentación del fabricante [10].

5.2.3. Limitaciones en la comunicación PLC-DAQ.

Analizando las posibilidades de comunicación ofrecidas por el PLC y la DAQ disponibles, se observan las siguientes limitaciones:

- El uso de numerosas conexiones cableadas (1 cable por señal), supone una compleja instalación (cableado elevado con numerosas posibilidades de fallo).
- Las entradas de la tarjeta TSX DMZ 28DR funcionan a 24 VDC, lo que imposibilita conectar directamente salidas digitales de la DAQ (a 5 VDC) con entradas digitales de esta tarjeta.
- Solamente se dispone de 2 salidas analógicas en la DAQ y de 1 en el PLC, con lo que la comunicación de forma directa mediante señales analógicas, también queda muy limitada.

Todas estas limitaciones, hacen que se deban aplicar diferentes soluciones para comunicar todas las variables necesarias para el proceso. En total se utilizan 5 soluciones distintas que se explican con detalle en el apartado siguiente.

Tal y como ya se mencionó, una solución basada en OPC facilitaría enormemente la comunicación, además de incrementar las prestaciones y la escalabilidad de la solución aportada.

5.3. Soluciones adoptadas en la comunicación PLC-DAQ.

Una vez detallados los problemas, se mencionan las soluciones aportadas para realizar la comunicación de todas las variables necesarias, un total de 21. En el apartado 5.4 se concretarán todas las señales comentadas a continuación y se mostrará un diagrama que ayudará al lector a comprender las conexiones mencionadas.

Mencionar que la programación correspondiente al PLC se verá en el anexo A, mientras que la correspondiente a Labview se verá en el anexo B.

5.3.1. Salidas analógicas de la DAQ (0_10v) – Entradas analógicas del PLC (0_10v).

Se ha configurado el mismo rango en ambos dispositivos, de forma que la conexión es directa.

- **Señal_Mando1:** Se utiliza para comunicar el tiempo de trabajo de la Máquina 1 tras la detección de la pieza en el Modo Automático. En Modo Manual, el valor de esta señal, establecerá la activación de los distintos mandos manuales (10). Obviamente, se debe crear una codificación en el PLC, para poder controlar todas estas acciones mediante una única señal de entrada (%IW0.2). Además, por comodidad y simplicidad en la programación, para los distintos mandos manuales se trasladará a entradas digitales el significado de la señal analógica. Se tiene la programación del PLC (módulo SR0) en la figura 5.1. Esta señal también controlará la entrada (2500>%IW0.2>1500) y la salida (1500>%IW0.2>500) del Menú de Configuración del sistema detector de patrones (Modo Visión), programada directamente sobre las transiciones de acceso y salida de dicha etapa.

```
(*En caso de estar en modo Manual, compruebo si se pulsa algún mando manual*)  
IF %X72 AND %IW0.2>2750 AND %IW0.2<=7750 THEN  
IF %IW0.2<3250 THEN (*Manual_C1*)  
SET %I3.4;  
ELSIF %IW0.2<3750 THEN (*Manual_sell_ad*)  
SET %I3.8;  
ELSIF %IW0.2<4250 THEN (*Manual_C2*)  
SET %I3.5;  
ELSIF %IW0.2<4750 THEN (*Manual_sell2_ad*)  
SET %I3.10;  
ELSIF %IW0.2<5250 THEN (*Manual_C3*)  
SET %I3.6;  
ELSIF %IW0.2<5750 THEN (*Manual_sell_atras*)  
SET %I3.9;  
ELSIF %IW0.2<6250 THEN (*Manual_C4*)  
SET %I3.7;  
ELSIF %IW0.2<6750 THEN (*Manual_sel2_atras*)  
SET %I3.11;  
ELSIF %IW0.2<7250 THEN (*Manual_m1*)  
SET %I3.2;  
ELSIF %IW0.2<7750 THEN (*Manual_m2*)  
SET %I3.3;  
END_IF;  
END_IF;
```

Figura 5.1. Programación de Señal_mando1 para mandos manuales en el PLC.

- **Señal_Mando2:** Se utiliza para comunicar el tiempo de trabajo de la Maquina 2 tras la detección de la pieza en el Modo Automático.

Ambas señales, siempre deben ser mayor de 0,5 v. Si el PLC detecta un valor menor, entenderá que se ha producido un error de comunicación entre el SCADA y el PLC y se generará una parada. La figura 5.2 muestra el algoritmo desarrollado en Labview para la transmisión de estas señales. Destacar que E_Detección hace referencia a la etapa donde se realiza la detección.

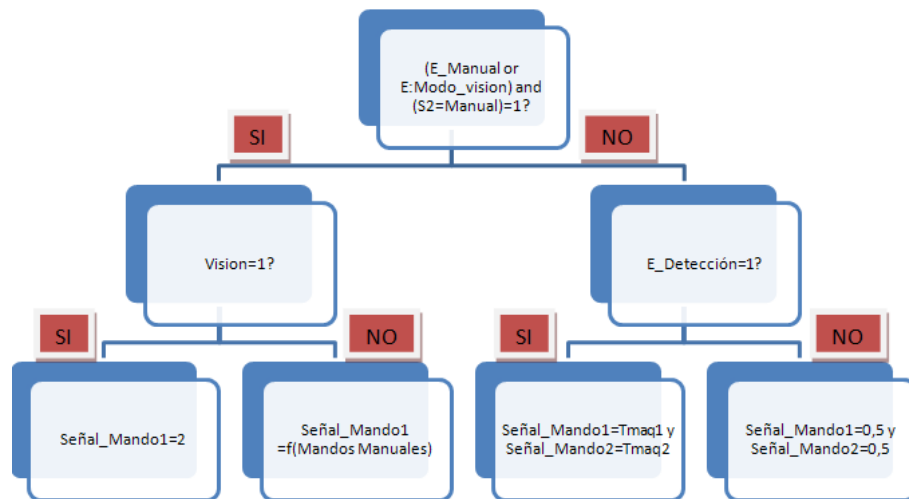


Figura 5.2. Algoritmo de funcionamiento de Señal_Mando1.

Por otro lado, se debe mencionar que la DAQ no está capacitada para enviar a la vez dos salidas analógicas, lo que hace necesario realizar el envío de ambas señales mediante una "flat Sequence Structure", realizando de esta forma el envío en ciclos distintos. Se observa esto en la figura 5.3.

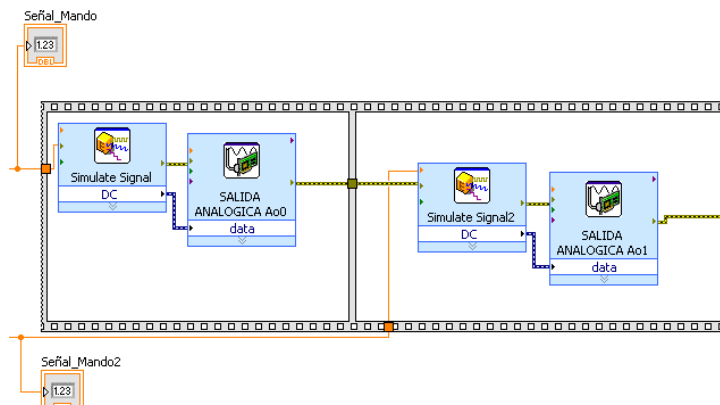


Figura 5.3. Envío de señales analógicas Señal_Mando1 y Señal_Mando2 desde el SCADA.

5.3.2. Salidas analógicas del PLC (0_10v) – Entradas analógicas de la DAQ (0_10v).

Al igual que en el caso anterior, la conexión es directa. Se establece la comunicación de la variable **Señal_Control**, desde la que el PLC informará al SCADA del motivo de la parada de emergencia (uno de los descritos en el apartado 4.4.6), cuando esta se produzca. De forma similar al caso anterior, continuamente se envía un valor mayor de 0,5 v, y en caso contrario, el sistema entenderá que se ha producido un error de comunicación entre el SCADA y el PLC y se generará una parada.

5.3.3. Salidas digitales de la DAQ (5v) –Relés –Entradas digitales del PLC (24v).

Para comunicar las variables de **RUN_STOP** y **SETA_EMERGENCIA**, es necesario utilizar sendos relés para adaptar salidas de 5v procedentes de la DAQ, con entradas de 24v en la tarjeta 28DR1 del PLC. Se realiza esta conexión debido a la necesidad de que estas variables puedan llegar al PLC como entradas externas para que el PLC pueda leerlas incluso cuando esté en modo STOP. La figura 5.4 muestra el esquema de conexión.

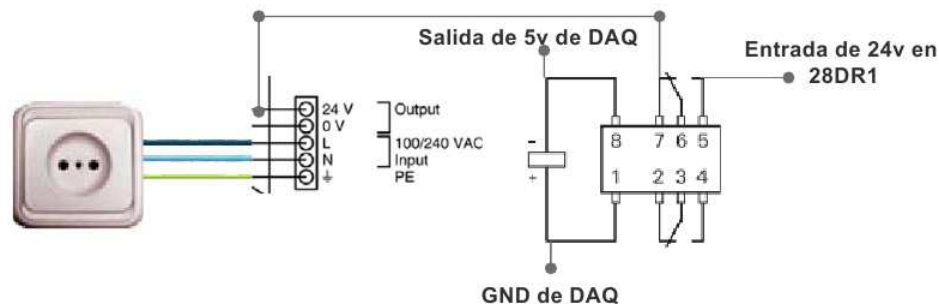


Figura 5.4. Esquema de conexión de relés para adecuar tensión entre PLC y DAQ.

Para más información sobre los relés se puede consultar documentación del fabricante [10].

5.3.4. Salidas digitales de la DAQ (5v) –Entradas analógicas del PLC (0_10v).

Para esta comunicación, se programa en el PLC que a partir de señales >2 v se entiende como un 1 lógico. La figura 5.5 muestra esta programación en el PLC.

```
(*Doy significado a las consignas recibidas como salidas digitales desde  
el SCADA a entradas Analógicas en el PLC*)  
IF %IW0.8>2000 THEN (*Automatico_Manual*)  
SET %I3.12;(*Automático*)  
ELSE SET %I3.13;(*Manual*)  
END_IF;  
IF %IW0.6>2000 THEN (*Marcha*)  
SET %I3.1;  
END_IF;  
IF %IW0.5>2000 THEN (*Rearme*)  
SET %I3.0;  
END_IF;  
IF %IW0.3>2000 THEN (*Patron detectado*)  
SET %I1.10;  
END_IF;  
IF %IW0.4>2000 THEN (*Error al tomar fotografia*)  
SET %I3.14;  
END_IF;
```

Figura 5.5. Programación de pulsadores y selectores de mando

Al igual que se vio para la **Señal_Mando1** en el apartado 5.3.1, se trasladará a entradas digitales el significado de las entradas analógicas para mayor comodidad y uniformidad en la programación. Es decir, cuando se quiera hacer referencia a una de estas variables en el programa, se hará referencia a la variable digital, que no es otra cosa que una codificación de la señal que verdaderamente hace la comunicación, que es la señal analógica.

5.3.5. Salidas digitales del PLC (5v) –Entradas digitales de la DAQ (5v).

Se establece la comunicación de 11 salidas digitales del PLC (28DR2), cuyos comunes se han alimentado a 5v procedentes de la DAQ, con otras tantas entradas digitales de la DAQ (5v). De esta forma, se consiguen conexiones digitales directas válidas a 5v. Estas señales informarán del estado del proceso al SCADA.

5.4. Resumen de variables de comunicación.

Mostrar gráficamente todas las conexiones establecidas, es complicado y su visualización no sería nada clara, por lo que se expone mediante la Tabla 5.2 el resumen de las variables de comunicación establecidas y el tipo de las mismas de acuerdo a lo explicado en el apartado anterior. Además se detallan en la figura 5.8 las interfaces de los elementos interconectados según lo indicado en esta tabla.

Variable *	SALIDAS					ENTRADAS				
	Tipo	Volt	Placa	Pto físico	Pto lógico	Tipo	Volt	Placa	Pto físico	Pto lógico
Run_stop	DIG	5	DAQ	52	P0.0	DIG	5	RELÉ1	8	-
Seta_Emergencia	DIG	5	DAQ	17	P0.1	DIG	5	RELÉ2	8	-
Run_stop	DIG	24	RELÉ1	5	-	DIG	24	DMZ 28DR1	9	%I1.8
Seta_Emergencia	DIG	24	RELÉ2	5	-	DIG	24	DMZ 28DR1	10	%I1.9
Auto_Manual	DIG	5	DAQ	6	P1.5	ANA	0_10	E/S Integra	13	%IW0.8
Marcha	DIG	5	DAQ	5	P1.6	ANA	0_10	E/S Integra	12	%IW0.6
Rearme	DIG	5	DAQ	38	P1.7	ANA	0_10	E/S Integra	4	%IW0.5
Error_Foto	DIG	5	DAQ	37	P2.0	ANA	0_10	E/S Integra	11	%IW0.4
Foto_ok	DIG	5	DAQ	3	P2.1	ANA	0_10	E/S Integra	3	%IW0.3
Señal_Mando1	ANA	0_10	DAQ	22	Ao0	ANA	0_10	E/S Integra	10	%IW0.2
Señal_Mando2	ANA	0_10	DAQ	21	Ao1	ANA	0_10	E/S Integra	5	%IW0.7
Señal_Control	ANA	0_10	E/S Integra	1	%QW0.10	ANA	0_10	DAQ	68	Ai0
Emergencia	DIG	5	DMZ 28DR2	20	%Q4.0	DIG	5	DAQ	49	P0.2
Mem_Emer	DIG	5	DMZ 28DR2	22	%Q4.1	DIG	5	DAQ	47	P0.3
E_Inicial	DIG	5	DMZ 28DR2	23	%Q4.2	DIG	5	DAQ	19	P0.4
E_Manual	DIG	5	DMZ 28DR2	24	%Q4.3	DIG	5	DAQ	51	P0.5
E_Posicionamiento	DIG	5	DMZ 28DR2	26	%Q4.4	DIG	5	DAQ	16	P0.6
E_Automatico	DIG	5	DMZ 28DR2	27	%Q4.5	DIG	5	DAQ	48	P0.7
E_Fin_Ciclo	DIG	5	DMZ 28DR2	28	%Q4.6	DIG	5	DAQ	11	P1.0
Rearmado	DIG	5	DMZ 28DR2	29	%Q4.7	DIG	5	DAQ	10	P1.1
E_Detección	DIG	5	DMZ 28DR2	31	%Q4.8	DIG	5	DAQ	43	P1.2
Rearmando	DIG	5	DMZ 28DR2	32	%Q4.9	DIG	5	DAQ	42	P1.3
E_Modo_Vision	DIG	5	DMZ 28DR2	33	%Q4.10	DIG	5	DAQ	41	P1.4

*El nombre de la variable puede variar ligeramente entre el programa del PLC y del SCADA.

Tabla 5.2. Resumen de conexión de variables de comunicación.

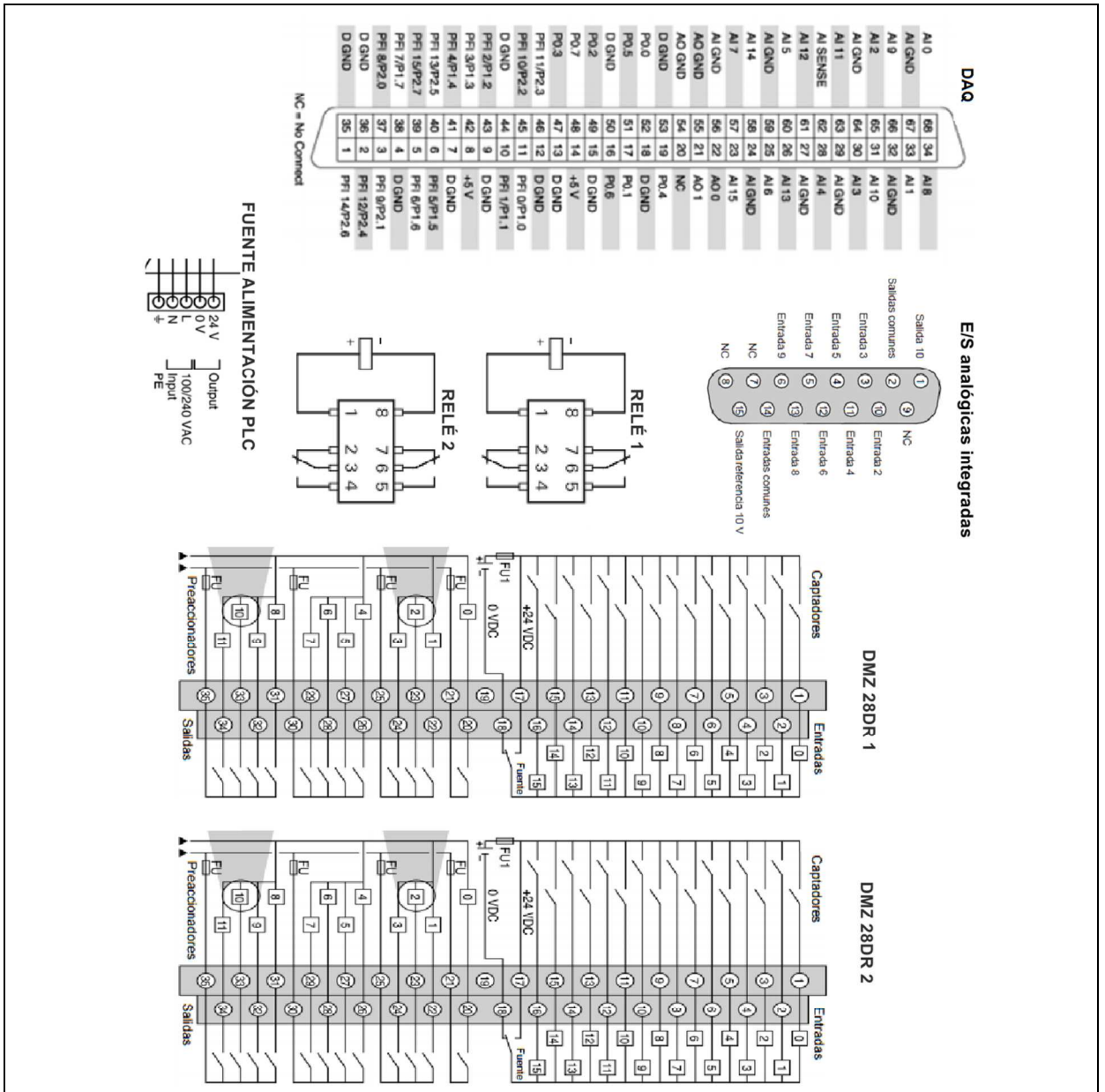


Figura 5.6. Interfaces físicas de los elementos a comunicar.

6. Sistema detector de patrones.

6.1. Visión por computador.

6.1.1. Introducción.

La visión por computador surge en la década de los 80 del siglo XX. La revolución de la Electrónica, con las cámaras de vídeo CCD y los microprocesadores, junto con la evolución de las Ciencias de la Computación hace que sea factible la Visión Artificial.

Podemos definir la Visión por computador como aquel campo de la Inteligencia Artificial enfocado a que las computadoras puedan extraer información del mundo físico a partir de imágenes, ofreciendo soluciones a problemas del mundo real.

Objetivos principales para los que se utilizan técnicas de Visión por computador son:

- Mejorar la información que obtiene un ser humano de una imagen digitalizada:
 - Para obtener imágenes de mejor calidad estética.
 - Para descubrir información que inicialmente no era visible.
- Conseguir la percepción automática (extracción automática de información) por parte de una máquina.

El número de aplicaciones relacionadas con la Visión Artificial aumenta cada día. En la Tabla 6.1 se citan algunos de los campos donde es empleada esta disciplina.

Área de Producción	Aplicación
Procesos industriales	Reconocimiento de piezas
Control de calidad	Inspección de productos (papel, aluminio, acero, etc), circuitos impresos, alimentos (naranjas)
Robótica	Control de soldaduras, guiado de robots, etc
Biomédicas	Análisis de imágenes de microscopio (virus, células, proteínas)
Astronomía	Exploración del espacio
Reconocimiento de caracteres	Inspección de textos
Control de tráfico	Matrículas de vehículos.
Militares	Vigilancia

Tabla 6.1. Campos de aplicación de la visión por computador

6.1.2. Etapas de un sistema de visión por computador.

El ser humano captura la luz a través de los ojos, y esta información circula a través del nervio óptico hasta el cerebro donde se procesa. La visión artificial, en un intento de reproducir este comportamiento, define tradicionalmente cuatro fases principales [18]:

- La primera fase, que es puramente sensorial, consiste en la captura o adquisición de las imágenes digitales mediante algún tipo de sensor (normalmente sensores CCD y CMOS incluidos en cámaras de fotos).
- La segunda etapa consiste en el tratamiento digital de las imágenes, con objeto de facilitar las etapas posteriores. En esta etapa de procesamiento previo es donde, mediante filtros y transformaciones geométricas, se eliminan partes indeseables de la imagen (suavizado) o se realzan partes interesantes de la misma.
- La siguiente fase se conoce como segmentación, y consiste en aislar los elementos que interesan de una escena para comprenderla.
- Por último se llega a la etapa de reconocimiento o clasificación. En ella se pretende distinguir los objetos segmentados, gracias al análisis de ciertas características que se establecen previamente para diferenciarlos.

Estas cuatro fases no se siguen siempre de manera secuencial, sino que en ocasiones deben realimentarse hacia atrás, como puede apreciarse en la figura 6.1. Así, es normal volver a la etapa de segmentación si falla la etapa de reconocimiento, o a la de preproceso, o incluso a la de captura, cuando falla alguna de las siguientes.

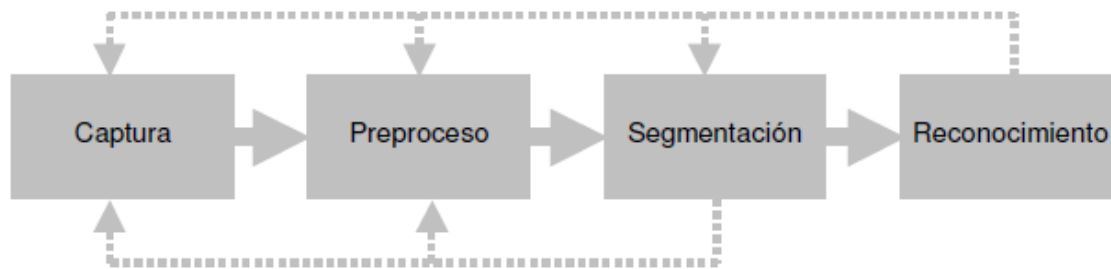


Figura 6.1. Etapas típicas en un sistema de visión artificial.

6.1.3. Muestreo y cuantificación. Píxeles.

Un sistema de Visión Artificial actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etcétera. Estas representaciones suelen darse en forma de imagen bidimensional. Una imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por ejemplo su brillo o su matiz). Se puede representar como se indica en la figura 6.3.

La información de la imagen digital se muestrea espacialmente y se cuantifica. Se conoce el valor de $V(x,y)$, como se ve en la figura 6.2 en un conjunto finito de puntos en el plano. M posiciones x y N posiciones y , es decir, $M \times N$ elementos.

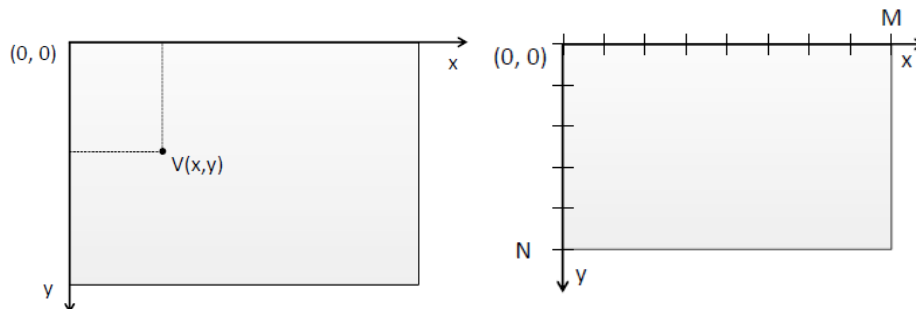


Figura 6.2. Representación de píxeles.

Cada uno de estos elementos recibe el nombre de píxel (picture element). La distancia entre píxeles determina la distancia de muestreo espacial. Esta puede ser distinta en la dirección x (Δx) y en la dirección y (Δy). Esta distancia de muestreo determina la resolución espacial que es mayor cuanto menor sea ésta.

El valor de la magnitud muestreada $V(n, k)$ puede tomar solamente un conjunto discreto de valores.

La distancia entre estos valores depende del número de bits utilizado para su cuantificación, del rango de la magnitud y del tipo de cuantificación (ej. lineal o exponencial). Cuanto mayor sea el número de píxeles y mayor el número de niveles de cuantificación mayor será la calidad de la imagen pero también mayor su tamaño.

Los píxeles van a ser los elementos fundamentales sobre los que vamos a trabajar en las técnicas de procesado de las imágenes [13].

6.2. Elemento de captación.

6.2.1. Elemento sensor.

La función de las cámaras de visión artificial es capturar la imagen proyectada en el sensor, para poder transferirla a un sistema electrónico y que pueda ser interpretada, almacenada y/o visualizada. Han tenido una rápida evolución en los últimos años. Actualmente se basan en tecnologías CCD (Charge Coupled Devide) o CMOS (Complementary Metal Oxide Semiconductor), pero la función de cada una de ellas es la misma, convertir los fotones en carga eléctrica y transformarla en una imagen. El sensor de la cámara cubre un área o matriz de píxeles donde llegan los fotones. Los tamaños más estándar de CCD están definidos en pulgadas (ver figura 6.3) siendo los más utilizados los de 1/3", 1/2" y 2/3".

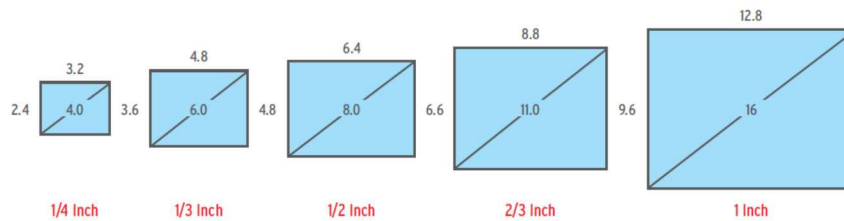


Figura 6.3. Tamaños estándar de CCD en pulgadas.

Los sensores de las cámaras de visión artificial actuales son mayoritariamente CCD. Están compuestos por millones de pequeños semiconductores de silicio (diodos), los cuales captan los fotones convirtiéndolos en señales eléctricas. A mayor intensidad de luz, más carga eléctrica existirá. El número de píxeles viene determinado por la cantidad de semiconductores del sensor que codifica esta información en un valor digital a través de su conversor analógico digital y da una salida digital en formato binario que en realidad es la imagen captada una vez digitalizada [14].

6.2.2. Distancia focal, profundidad de campo y diafragma.

Se presentan aquí ciertos conceptos a conocer a la hora de realizar captura de imágenes.

La distancia focal (F) es la medida en mm relativa a la distancia entre la lente y el elemento sensor (ver figura 6.4). La imagen sale invertida debido a la lente del objetivo, que invierte la imagen al recibir los rayos de luz.

Los objetivos de las cámaras tienen una distancia focal fija o variable, dependiendo del tipo de objetivo. Al variar la distancia focal conseguimos un menor o mayor acercamiento del objeto, comúnmente llamado zoom. Estando a la misma distancia de trabajo, lentes con diferente distancia focal muestran la misma imagen en distinto tamaño [14].

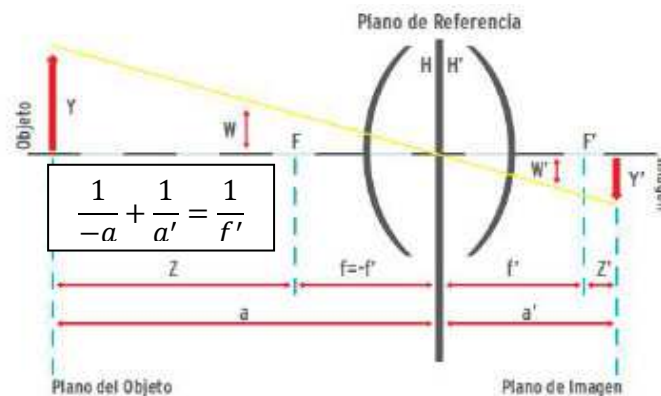


Figura 6.4. Distancia focal, F .

Normalmente las ópticas utilizadas en sistemas de visión artificial tienen una distancia focal variable, también denominada foco o enfoque. Es muy importante que el objeto a analizar esté perfectamente enfocado para su posterior análisis.

Otro parámetro a conocer es la profundidad de campo, que es la distancia en la que los objetos aparecen enfocados (ver figura 6.5). A menor apertura del diafragma, mayor será este parámetro.

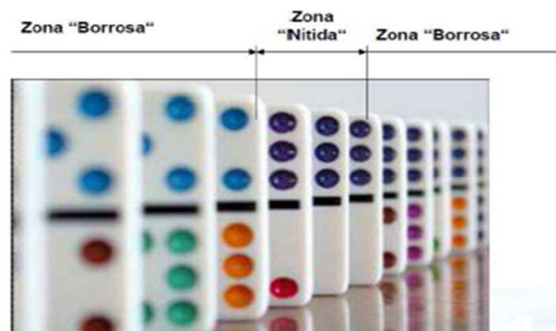


Figura 6.5. Enfoque.

El diafragma es una parte del objetivo que limita el nivel de luz que penetra en la cámara y a su vez al CCD. Funciona como el iris humano, abriéndose o cerrándose permitiendo que entre más o menos luz. La posición de apertura o cierre del objetivo (su diámetro es D) se denomina apertura del diafragma y se mide en números f (N), como vemos en la figura 6.6.

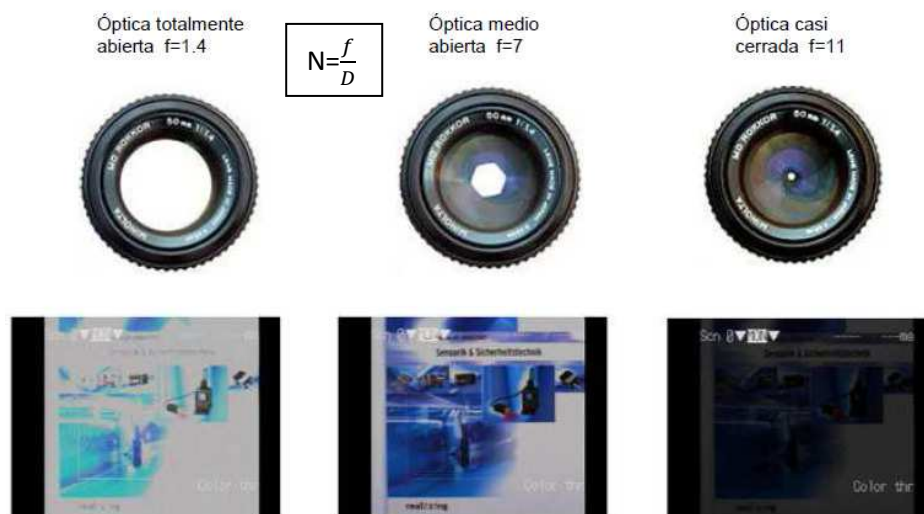


Figura 6.6. Apertura de diafragma.

6.2.3. Iluminación.

Para poder formar la imagen de un objeto es necesario capturar la intensidad luminosa que procede de él. En la mayor parte de los casos vamos a considerar que los objetos no emiten su propia luz, sino que son iluminados por una fuente externa. El éxito del procesado de imagen depende de elegir una iluminación adecuada. La uniformidad de luminosidad en la superficie, la estabilidad temporal y su respuesta espectral, así como las propiedades ópticas del entorno deben ser consideradas. Existen varios tipos de iluminación, de las que se deberá escoger la más apropiada para cada caso en concreto.

Para nuestro proyecto, el hecho de utilizar una maqueta, como ya se comentó en el punto 3.7 hace difícil conseguir unas condiciones ideales de iluminación, debido a la variación de la luz en el aula donde se ubica la maqueta y la geometría y dimensiones de la misma. Tampoco se ha visto necesario realizar un gasto en equipos de iluminación específicos, sino que se han utilizado equipos de iluminación rudimentarios (flexos, focos, linternas, etc) disponibles por cualquier persona.

En busca de conseguir la situación más favorable de iluminación, tras probar con distintos elementos y situaciones de iluminación, se ha determinado la utilización de un flexo cuya fuente de luz incide directamente sobre la pieza en el momento de la detección, además de contar con la iluminación del aula encendida.

En caso de que se implementara el sistema en un proceso real, distinto de nuestra maqueta, se debería conseguir una iluminación constante y adecuada. Sería necesario realizar un estudio y presupuesto de los elementos de iluminación a instalar. Se debería analizar también si la iluminación frontal sería la más adecuada, teniendo en cuenta el tipo de piezas a detectar.

6.2.4. Exposición.

La exposición es la acción de someter un elemento fotosensible (en cámaras digitales el sensor) a la acción de la luz, piedra angular de la fotografía. En función del grado de exposición de una foto podremos hablar de tres situaciones:

- **Subexposición:** La fotografía presenta una carencia considerable de luz frente a la de la escena original.
- **Exposición:** La fotografía recoge la cantidad de luz apropiada para representar fielmente la escena fotografiada)
- **Sobreexposición:** Se aprecia un exceso de luz en la fotografía frente a la escena retratada.

Existen tres factores que influyen en la exposición:

- **Apertura del diafragma.** Determina la cantidad de luz que se deja incidir sobre el sensor de nuestra cámara. Una mayor apertura supondrá una mayor cantidad de luz actuando sobre el sensor.
- **Velocidad de obturación o tiempo de exposición.** Marca el tiempo durante el que la luz incide sobre el sensor. Un mayor tiempo y, por tanto, una menor velocidad, darán lugar a que la luz incida durante un periodo más prolongado sobre el sensor.
- **Sensibilidad ISO.** Refleja lo receptivo que se muestra el sensor de nuestra cámara ante la luz que actúa sobre él. Una mayor sensibilidad hará que, a igual cantidad de luz y tiempo de incidencia, el sensor se haya excitado más y, por tanto, la fotografía tenga una mayor exposición [14].

Se han mencionado estos conceptos, sin entrar demasiado en detalle, debido a su importancia en el funcionamiento de nuestro sistema como se verá en el apartado 6.3.2.

6.3. Cámara utilizada: Creative Live! Cam Chat HD

6.3.1. Características.

La cámara que se va a utilizar como elemento captador de las imágenes será una webcam modelo Creative Live! Cam Chat HD, que podemos apreciar en la figura 6.7. De sus especificaciones técnicas destacamos las siguientes:

- Sensor: sensor de imagen HD 720p (1.280 x 720)
- Resolución de vídeo: HD 720p (1.280 x 720 píxeles)
- Resolución de foto: 5,7 megapíxeles
- Velocidad de fotogramas: hasta 30 fps de calidad HD 720p
- Micrófono integrado con cancelación de ruido
- Botón Snapshot / Live! Contacts
- Enfoque fijo
- Base adaptable
- Longitud del cable: 1,5 metros
- USB 2.0 de alta velocidad
- **Exposición automática** (Velocidad de obturación y apertura de diafragma automática)



Figura 6.7. Modelo de cámara utilizada.

Se ha marcado en negrita la especificación de exposición automática, debido a la importancia que va a tener este parámetro en nuestro sistema detector de patrones.

6.3.2. Limitaciones de la cámara en cuanto a exposición.

Una vez desarrollado el sistema detector de patrones que se explicará con detalle en el apartado 6.5, se sometió al mismo a una serie de pruebas de funcionamiento. Una de estas pruebas arrojó unos resultados, en un principio desconcertantes, hasta que se consiguió dar una explicación al fenómeno que se estaba produciendo.

Las pruebas consistían en lo siguiente: Se posicionaba una determinada pieza en la zona donde el sistema debía de capturar la imagen para realizar la detección. Una vez allí, se ejecutaba en bucle el algoritmo detector de patrones, de forma que se tomaba una foto, se procesaba y se obtenía el resultado de detección, volviendo seguidamente a comenzar el ciclo tomando una nueva foto. Lo que se buscaba con esta prueba era conseguir gran repetibilidad en los resultados obtenidos, ya que para una misma pieza en las mismas condiciones deberían obtenerse resultados prácticamente idénticos. Sin embargo, esto no era así. Existía cierta variación en los resultados, que en ocasiones se volvía inaceptable para las exigencias de funcionamiento que se estaban buscando. No era aceptable la confusión de unas piezas con otras.

Tras consultar con un experto en Fotografía, se determinó que el problema es debido a las prestaciones de la cámara utilizada. Es un problema de la variabilidad en la obtención de la **exposición** debido a que ésta, se obtiene **automáticamente**, como se ha visto en las características de la cámara. Esta cámara, y las webcam en general, son utilizadas para aplicaciones sencillas y sus prestaciones pueden ser suficientes para los usuarios en general, que normalmente las usan en condiciones de iluminación pobres y muy variadas. Sin embargo, esta limitación puede ser importante para un uso avanzado como es nuestro caso, ya que se necesita alta repetibilidad y exactitud en las características de la imagen tomada para realizar una adecuada detección. El problema es que ese tipo de cámaras ajustan automáticamente los niveles de exposición dependiendo de las masas de luz/sombra que detecten. Aunque la iluminación no varíe, tienden a sobreexponer para ajustar la iluminación a lo que la cámara entiende como parte más importante y esto conlleva que las imágenes tengan parámetros de exposición ligeramente diferentes ocasionando la variación mencionada en los resultados.

La solución para evitar esto, sería usar una cámara reflex o al menos una que permita establecer los ajustes de exposición, velocidad de obturación y diafragma de modo manual, para que si la iluminación es constante, no varíen entre disparos.

Resumiendo, esta cámara, nos sirve para el cometido del presente proyecto, simplemente académico. En caso de implementar el sistema en un proceso real se debería de buscar una cámara de mayores prestaciones, adecuada para conseguir mayor repetibilidad y menos variabilidad.

6.4. Algoritmos de detección de patrones.

En esta apartado se va a describir el funcionamiento de los algoritmos necesarios para el proceso de detección de patrones. Existen dos algoritmos fundamentales para el funcionamiento del sistema, como son el algoritmo de creación de nuevos patrones y el algoritmo de detección de patrones de forma automática. Ambos algoritmos, están programados en Labview, mediante subVIs que son llamados por el programa principal.

Antes de comenzar a describir la implementación en Labview de estos algoritmos, se van a describir los diagramas de flujo que siguen para ayudar al lector a seguir la explicación.

6.4.1. Diagrama de flujo de creación de nuevos patrones.

Como se vio en el apartado 3.5.2, existe la posibilidad de crear nuevos patrones en el sistema para utilizarlos posteriormente en la detección. La figura 6.8 muestra el diagrama del proceso llevado a cabo para la creación de un nuevo patrón para ser utilizado posteriormente en el sistema. En el apartado 6.5.4.1 se verá la programación que se realiza en Labview para conseguir este proceso.

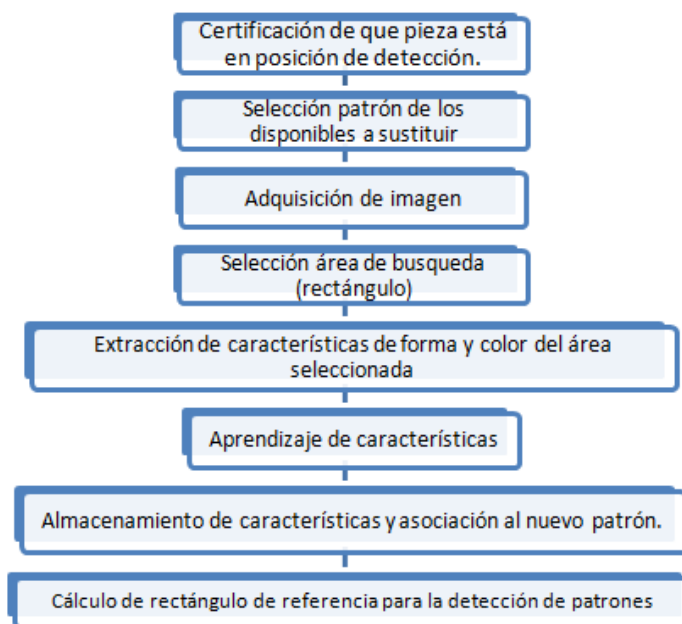


Figura 6.8. Diagrama de flujo de algoritmo de creación de nuevos patrones.

6.4.2. Rectángulo de trabajo.

Se ha hecho referencia en varias ocasiones al rectángulo de trabajo que se utilizará para la búsqueda de patrones. Este rectángulo, está formado por un array que define las coordenadas del rectángulo que contendrá la parte de la imagen que será analizada para determinar si la pieza contenida en dicha imagen se corresponde con alguno de los patrones aprendidos por el sistema.

Este rectángulo, es calculado en función de las dimensiones de los patrones y del número de estos con el subVI "**Rectangulo.vi**". Cada vez que en el sistema se introduce un nuevo patrón o se varía el número de patrones con los que se va a trabajar, se produce un nuevo cálculo de las coordenadas de este rectángulo. Dicho rectángulo, se calcula de modo que contenga el mínimo área posible que incluya a todos los posibles patrones del sistema. El uso de este rectángulo reduce en gran medida el área de búsqueda de patrones en la imagen capturada. Esta reducción incrementa la velocidad de funcionamiento del algoritmo, evitando búsquedas innecesarias en zonas donde es conocido que no se va a encontrar ningún patrón e incluso podrían provocar detecciones erróneas.

Como se adelantó en el apartado 3.6, los parámetros de este rectángulo son almacenados en el archivo de registro llamado "**Rectangulo.xls**".

La programación del cálculo de este rectángulo mediante "**Rectangulo.vi**", será vista con detalle en el anexo B, junto con el resto del programa de Labview.

6.4.3. Diagrama de flujo de detección de patrones de forma automática.

La figura 6.9 muestra el diagrama del proceso llevado a cabo para la detección de una pieza determinada que se encuentra en la zona de detección del proceso ($E_Detección=1$) durante el ciclo automático de ejecución.

Destacar de nuevo, la importancia del rectángulo de trabajo explicado en el apartado anterior, ya que para la detección de patrones, la comparación para la búsqueda de coincidencia con un patrón, se realiza entre el área de la imagen tomada comprendida entre dicho rectángulo y los distintos patrones.

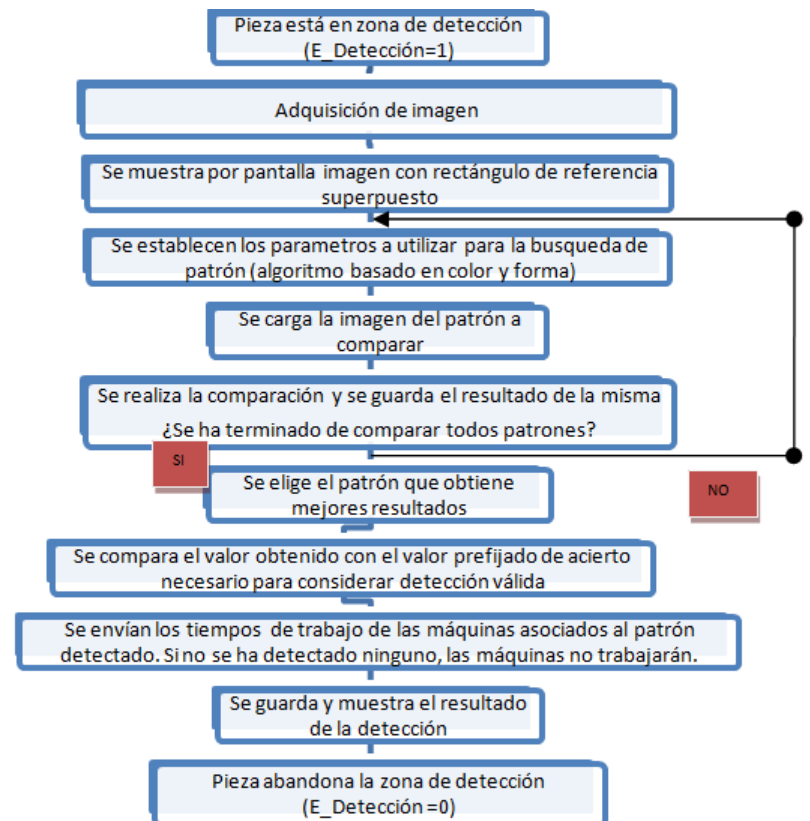


Figura 6.9. Diagrama de flujo de algoritmo de detección de patrones automática.

6.5. Uso de Labview para programación del sistema detector de patrones.

No se va a entrar en detalle, sobre las características de la programación en Labview. Simplemente se van a describir las principales características de las librerías contenidas en NI Vision y se van a detallar brevemente los Vis utilizados para nuestro sistema. Para mayor información sobre todo esto se deberá consultar la documentación del fabricante.

6.5.1. Adquisición y procesamiento de imágenes a través del paquete NI VISION.

A continuación se nombra el conjunto de librerías contenidas en NI Vision:

- **Ni-IMAQ:** Posee los VI (instrumentos Virtuales) que establecen el sistema de adquisición y captura de las imágenes
- **Vision Utilities:** Proporciona los VI que permiten crear y manipular imágenes, leer y escribir imágenes en diferentes formatos, establecer regiones de interés, manipular los píxeles, sobreponer líneas o rectángulos, calibrar para realizar conversiones de medidas de píxeles a medidas del mundo real, etc.
- **Image Processing:** Presenta un conjunto de VI que se utilizan para analizar, filtrar y procesar imágenes de NI Vision.
- **Machine Vision:** Posee VI que se utilizan en la visión artificial para realizar inspecciones de objetos, búsqueda de patrones, localización de bordes, análisis de partículas, selección dinámica de regiones de interés, etc.
- **IMAQ USB.** Proporciona los instrumentos virtuales necesarios para establecer el funcionamiento y adquisición de imágenes a través de cámaras USB de bajo coste.

Destacar además, la disponibilidad del **Vision Express** que Contiene Vis preparados para su funcionamiento inmediato como son el Visión Assistant (entorno interactivo para generación de prototipos de aplicaciones de inspección que pueden generar código listo para ejecutar en cualquiera de los lenguajes soportados) y el Vision Acquisition.

La captura de imágenes de nuestro sistema se realiza mediante **Vision Acquisition express**, que es un entorno interactivo que permite de forma rápida y cómoda la adquisición de imágenes mediante una cámara conectada a nuestro PC vía USB o similar. Ofrece las siguientes características:

- El sistema detecta automáticamente las cámaras (cuyos drivers reconoce) conectadas en el PC, permitiendo la elección de la cámara a utilizar.
- Permite varios tipos de adquisición: Simple, continua, intervalos, etc.
- Permite configurar las opciones disponibles de la cámara
- Permite almacenar, de forma transparente, las imágenes en una ubicación determinada.

6.5.2. Instrumentos virtuales de NI Vision utilizados.

Para los procesos de creación y detección de patrones, se utilizarán varios Vis, que se van a comentar a continuación. Son los únicos VIs que se van a explicar brevemente en este proyecto. Para una información más detallada de estos y otros Vis, será necesario consultar la ayuda de Labview o cualquier otra documentación o bibliografía al respecto. Se insiste en que estos Vis son explicados porque serán fundamentales en la programación de nuestros algoritmos que veremos en el apartado 6.5.4.

IMAQ Create (figura 6.10): Para que LabVIEW pueda manejar archivos gráficos o capturas debe reservar un espacio de memoria para su almacenamiento temporal.

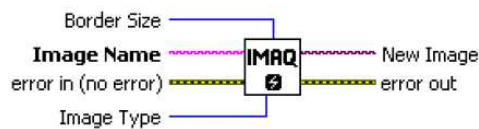


Figura 6.10. IMAQ Create.

- **Image Name:** Es la entrada más importante de este VI. En ella se proporciona el nombre asociado a la imagen, debe ser único.
- **Image Typ.** Se puede seleccionar entre escala de grises, color o compleja. En nuestro caso, al trabajar con imágenes de color se elegirá la opción RGB (U32).
- La salida **New Image**, nos proporciona la referencia a la imagen, que se proporciona como entrada a las muchas funciones usadas por NI Vision.

IMAQ Dispose (figura 6.11): Así como es importante reservar espacio de memoria, también lo es liberar este espacio de memoria cuando ya no se utilice más.

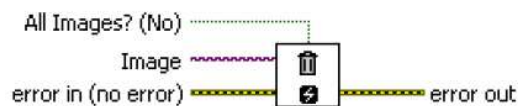


Figura 6.11. IMAQ Dispose.

IMAQ Select Rectangle(figura 6.12): Permite seleccionar un rectángulo de una imagen. Destacar la salida **Rectangle** que especifica las coordenadas del rectángulo elegido por el usuario.

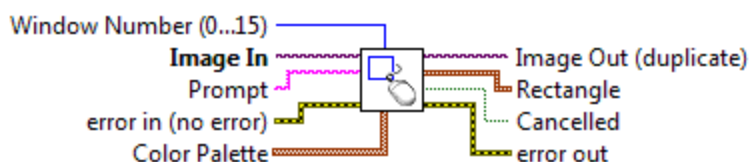


Figura 6.12. IMAQ Select Rectangle.

IMAQ Convert Rectangle to ROI (figura 6.13): Convierte un rectángulo en una Región de interés (ROI)



Figura 6.13. IMAQ Convert Rectangle to ROI.

IMAQ Extract Tetragon (figura 6.14): Permite extraer de una imagen un fragmento ajustando la resolución.

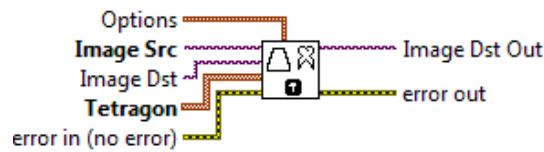


Figura 6.14. IMAQ Convert Rectangle to ROI.

IMAQ Setup Learn Color Pattern (figura 6.15): Define los parámetros a usar en la etapa de aprendizaje. Este VI se debe utilizar antes del IMAQ Learn Color Pattern, que se encarga de introducir las configuraciones para la búsqueda, estableciendo el modo de aprendizaje y la saturación a ser considerada. Destacamos las entradas que se han configurado en nuestro sistema de creación de patrones.

- **Learn Mode:** Especifica si se busca referencias tanto de posición como de rotación para el aprendizaje. En nuestro caso se deja en "All" (ambas).
- **Feature Mode:** Especifica si la función de aprendizaje incluye características de forma y/o color. En nuestro caso se ha dejado la opción que viene por defecto (ambas).

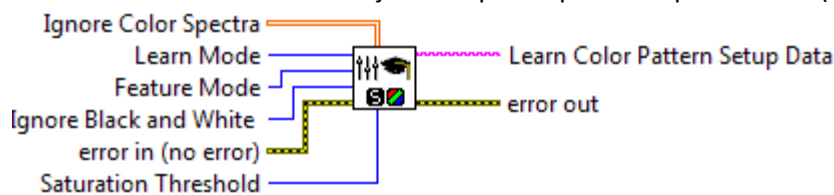


Figura 6.15 IMAQ Setup Learn Color Pattern.

IMAQ Learn Color Pattern (figura 6.16): Es una herramienta en la que se crea una descripción de los parámetros de la plantilla que se van a buscar durante la etapa de correlación de formas y color. Crea una plantilla en función de las opciones de configuración consideradas.

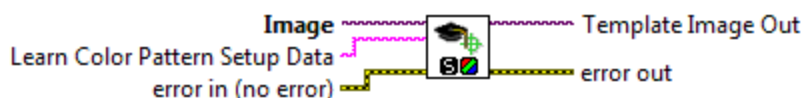


Figura 6.16. IMAQ Learn Color Pattern.

IMAQ Setup Match Color Pattern (figura 6.17): Establece los parámetros que son usados durante el proceso de reconocimiento. Destacar las siguientes opciones utilizadas en nuestro sistema de detección:

- **Match Feature Mode:** Especifica las características a buscar en la detección. En nuestro caso se considera la forma y el color
- **Color Score Weight:** Es el peso de 0 a 1000 que tienen las características del parámetro color en el proceso de reconocimiento. La diferencia de este valor respecto de 1000, es el peso que tienen las características del parámetro forma en el proceso de reconocimiento. En nuestro sistema de detección, existe la posibilidad de modificar este parámetro como se vio en el apartado 3.5.4.
- **Color Sensitivity:** Especifica la sensibilidad en cuanto a color que se tendrá en cuenta. En nuestro caso, se escogerá la opción de alta sensibilidad.
- **Search Strategy:** Especifica cómo se utilizan las características de color de la imagen durante la fase de búsqueda. Para nuestra detección se ha elegido la opción conservadora, en la que prima la precisión sobre la velocidad de reconocimiento.

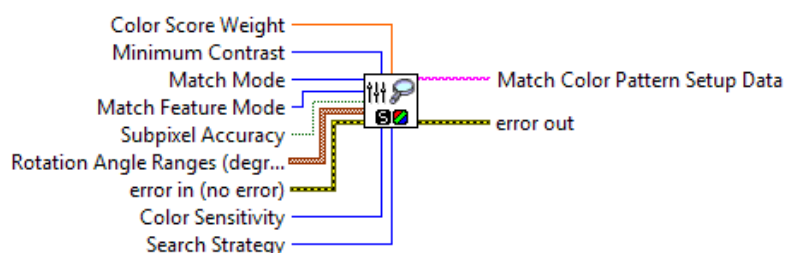


Figura 6.17. IMAQ Setup Match Color Pattern.

IMAQ Match Color Pattern (figura 6.18): Permite la búsqueda de un modelo en color, o la imagen de plantilla en color. Destacar las siguientes opciones utilizadas en nuestro sistema de detección:

- **Optional Rectangle:** Define una matriz de cuatro elementos que contiene las coordenadas de la región a procesar. Se indica el rectángulo contenido en la imagen a analizar donde se realizará la búsqueda de patrones.
- **Number of Matches Requested.** Es el número de resultados obtenidos. En nuestro caso es 1, ya que en cada imagen solo se espera encontrar un acierto.
- **Minimum Match Score.** Es el valor mínimo de puntuación (0-1000) con el que se considerará un resultado de detección válido. En nuestro caso se ha seleccionado 0 ya que el valor para obtener la detección será el score sacado de la salida Matches.
- **Number of Matches:** Indica el número de resultados positivos que se han encontrado en la imagen.
- **Matches:** Es un cluster que contiene parámetros de los resultados obtenidos. Entre otros parámetros, contiene el score, que será el parámetro que utilizaremos para establecer la detección en nuestro sistema como veremos en el apartado 6.5.4.

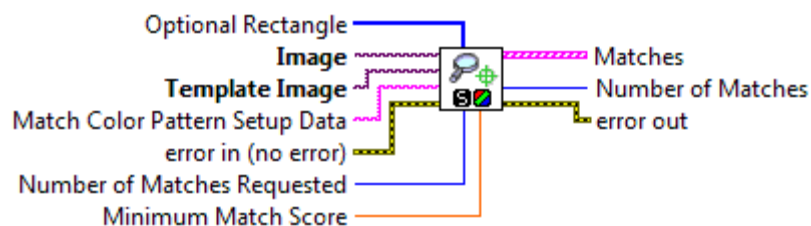


Figura 6.18. IMAQ Match Color Pattern.

IMAQ Write File (figura 6.19): Guarda una imagen junto con todos sus datos asociados de visión.

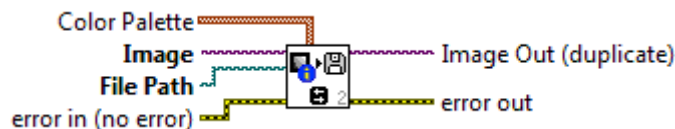


Figura 6.19. IMAQ Write File.

IMAQ Read Image And Vision (figura 6.20): Carga una imagen junto con todos sus datos asociados de visión.

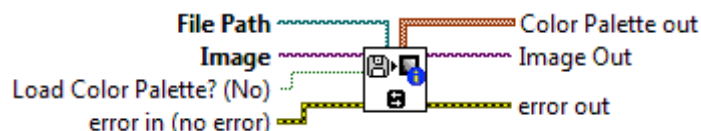


Figura 6.20. IMAQ Read Image.

6.5.3. Subvis para la detección de patrones.

Como se ha mencionado en varias ocasiones, se ha realizado una programación jerárquica en Labview basada en subVIs, que son utilizados para distintas funciones necesarias para procesos de obtención, modificación y detección de patrones. La tabla 6.2 muestra todos estos subVIs, dejando su programación para ser expuesta en el Anexo B.

Simplemente, destacar que el uso de esta forma de programar jerarquizada, ayuda a la hora de programar y facilita la escalabilidad.

SubVI	Función
Carga_Im_patrones.vi	Carga las imágenes de los patrones en el sistema.
Detector de patrones.vi	Realiza la detección de patrones en el proceso automático.
Menu_Configuracion.vi	Muestra al usuario el Menú Visión con las opciones de configuración disponibles.
Sel_patrones.vi	Configura el número de patrones con los que trabajar.
Rectangulo.vi	Algoritmo que calcula el rectángulo de trabajo donde se realiza la búsqueda.
Crear_patron.vi	Realiza el proceso de creación de un nuevo patrón.
Previa_crear_patron.vi	Se pregunta al usuario si la pieza está en la zona de detección para crear patrón.
Carga_2patrones.vi	Muestra al usuario la posibilidad de elegir entre los 2 patrones disponibles.
Carga_3patrones.vi	Muestra al usuario la posibilidad de elegir entre los 3 patrones disponibles.
Carga_4patrones.vi	Muestra al usuario la posibilidad de elegir entre los 4 patrones disponibles.
Carga_5patrones.vi	Muestra al usuario la posibilidad de elegir entre los 5 patrones disponibles.
Mod_Tmaq.vi	Configura los tiempos de trabajo para cada máquina en función de patrón.
Reinicio_tiempos.vi	Reinicia, para todos patrones, los tiempos de trabajo de máquinas.
Selección del tiempoM1.vi	Muestra al usuario la posibilidad de elegir el Tiempo de la Maq1 (entre 1 y 9).
Selección del tiempoM2.vi	Muestra al usuario la posibilidad de elegir el Tiempo de la Maq2 (entre 1 y 9).
Password.vi	Obliga al usuario a conocer la contraseña para poder acceder a ciertos menús.
Porcentaje_Color.vi	Permite escoger el % de ponderación del parámetro color en la detección.
Porcentaje_Acierto.vi	Permite escoger el % de puntuación necesaria para dar por válida una detección.
Carga_archivos.vi	Realiza la carga de los archivos vistos en el apartado 3.6

Tabla 6.2. SubVIs empleados para la detección.

6.5.4. Programación de algoritmos.

6.5.4.1. Creación de patrones. Creacion_patron.vi.

Este proceso se realizará mediante el subVI llamado “**Crear_patron.vi**”. La figura 6.21 muestra los pasos más importantes (del 3 al 7) de este subVI que se explican a continuación. En el anexo B se muestra detallada la programación completa de este subVI realizada en Labview.

Los pasos que se siguen para la creación de patrones concuerdan con los de la figura 6.8:

1. Se pregunta al usuario si la pieza está en la zona de detección para crear patrón (“**Previa_crear_patron.vi**”), como se vio en la figura 3.6.a.
2. Se da opción al usuario de elegir el patrón a modificar (“**Carga_xpatrones.vi**”) de entre los disponibles, como se vio en la figura 3.6.b.
3. Se captura la imagen mediante Vision Acquisition.
4. Selección por parte del usuario del rectángulo que contiene al patrón.
5. Extraer las características de visión contenidas en este rectángulo.
6. Aprendizaje de características de visión del área seleccionada en el rectángulo.
7. Almacenamiento de estas características (incluyendo la imagen).
8. Calculo de nuevo rectángulo de trabajo (“**Rectangulo.vi**”) y actualización en sistema.

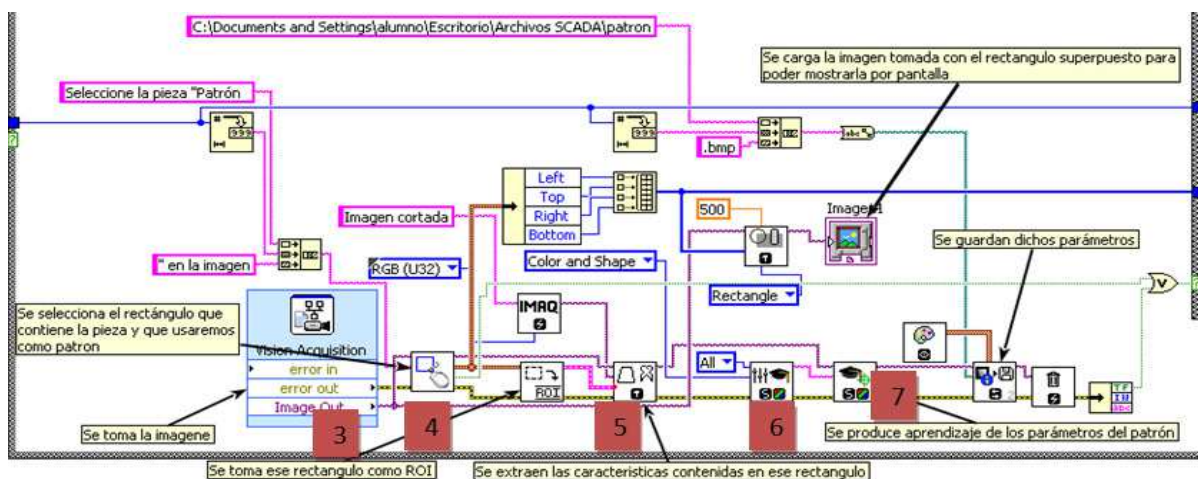


Figura 6.21. Creacion_patron.vi.

6.5.4.2. Detector de patrones. *Detector_patrones.vi*.

Como se vio en la figura 6.9, cuando se detecte que hay una pieza en situación de ser analizada (se active “E_Detección”), se ejecutará el subVI llamado “*Detector_patrones.vi*”.

Debido a la importancia de este algoritmo, parte fundamental del proyecto, se va a explicar a continuación, apoyándonos para ello de la imagen de su programa en Labview distribuido en cuatro figuras (6.22, 6.23, 6.24 y 6.25). En el anexo B se muestran estas imágenes con mayor detalle.

En la figura 6.22, se observa como al ejecutarse este algoritmo, lo primero que se hace es:

- Realizar una captura de imagen mediante Vision Acquisition.
- Se reinicializa “*Vector_Aciertos*” donde se incluirán los scores obtenidos para cada patrón.

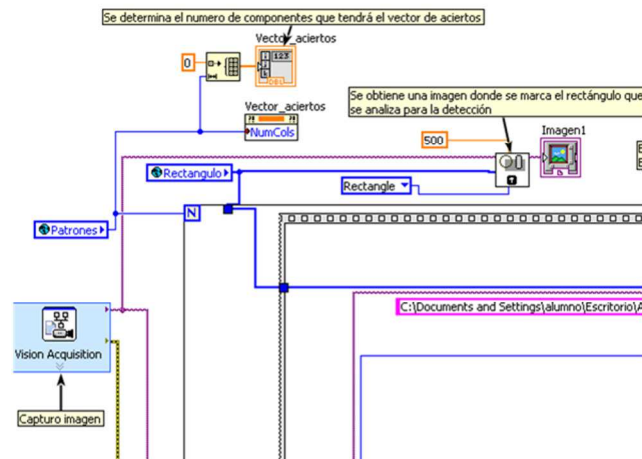


Figura 6.22. Algoritmo de detección. Parte 1.

Tras esto, se mostrará por pantalla la imagen capturada con el rectángulo de trabajo superpuesto. Por otro lado, vemos que se ejecuta un bucle for con número de iteraciones igual al número de patrones que tenemos en el sistema. Este bucle for sirve para ejecutar la comparación de la imagen capturada con cada uno de los patrones definidos en el sistema.

En la figura 6.23, se aprecia como la detección se realiza mediante la dupla de **Vis IMAQ Setup Match Color Pattern** y **IMAQ Match Color Pattern** mencionados en el apartado 6.5.2. Los parámetros configurados, que se aprecian en la imagen, ya se mencionaron en dicho apartado.

- La búsqueda de patrones se basará en los parámetros de color y forma.
- El % de ponderación de los parámetro color y forma serán definidos por el usuario.
- Se utilizará una estrategia conservadora, primando la fiabilidad sobre la velocidad de proceso.
- Se permite cierta rotación a la hora de realizar la búsqueda.
- Se selecciona, mediante los parámetros definidos en el rectángulo de trabajo, la zona de la imagen donde se buscarán los patrones.
- Se espera un único acierto en cada comparación.

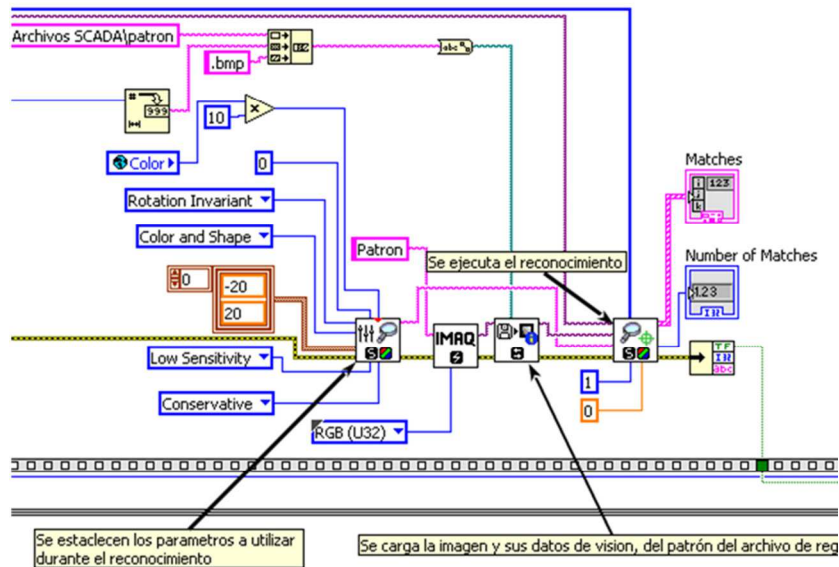


Figura 6.23. Algoritmo de detección. Parte 2.

En la figura 6.24 se observa cómo se recoge la información de la puntuación guardada tras la comparación para todos los patrones en "Vector_aciertos". De estos valores, se seleccionará el que tenga mayor valor (mayor score en la detección).

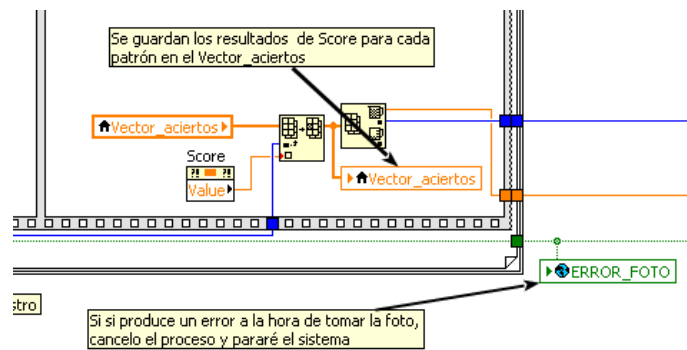


Figura 6.24. Algoritmo de detección. Parte 3.

En la figura 25, se ve como finalizado el bucle se selecciona el score mayor de todos y se compara con el parámetro de % de acierto predefinido para determinar si se puede considerar que se ha producido una detección de la pieza o no. Si se supera este valor, se producirá un envío de los tiempos de trabajo de las máquinas para ese patrón, y si no se supera se enviarán sendos 0 como valor de tiempos para que las piezas sean sacadas del sistema sin que las máquinas actúen sobre ellas. Tanto en caso de acierto, como en el caso de que no se haya reconocido ningún patrón, se almacenará el resultado en la Tabla de registro y en el archivo correspondiente.

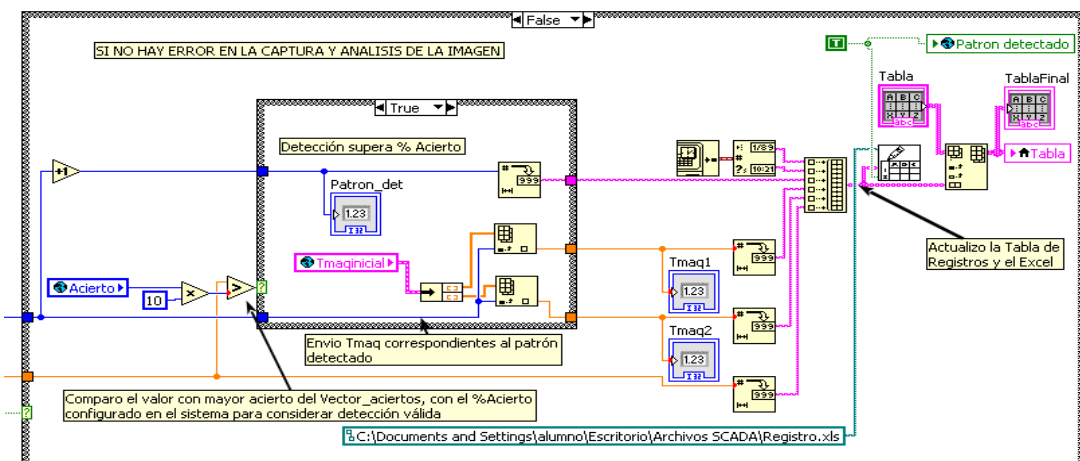


Figura 6.25. Algoritmo de detección. Parte 4.



7. Conclusiones.

Como se ha mencionado en varias ocasiones, para la implementación del sistema nos hemos encontrado con ciertas limitaciones, que se resumen a continuación:

- Trabajo sobre una maqueta.
- Disponibilidad limitada de recursos de comunicación de hardware (imposibilidad de comunicación mediante OPC).
- Características limitadas de la cámara web para la captura de imágenes.
- Material de iluminación rudimentario.

Obviamente, en caso de tratarse de una implementación profesional para un caso real, se debería prestar mucha mayor atención a la solución escogida en cuanto a la iluminación y características de la cámara. También, sería imperativo conseguir una comunicación rápida y eficaz entre el hardware utilizado mediante OPC o similar.

No obstante, mediante los medios disponibles, se ha podido implementar un sistema trasladable a un caso real donde se ha podido desarrollar una capacidad muy utilizada de la visión por computador en la industria como es la detección de patrones mediante procesado de imágenes.

La implementación de los sistemas de visión artificial en los procesos industriales, añaden a dichos procesos mayor rapidez, fiabilidad y seguridad. Aunque los dispositivos utilizados pueden obligar a una inversión inicial elevada, normalmente son amortizados en breve espacio de tiempo. Además, la gran cantidad de material que hay disponible en este campo, hace que existan multitud de posibilidades para elegir el hardware y software adecuados para las necesidades concretas de cada caso.

Destacar la capacidad de Labview para realizar trabajos de este tipo, sobre todo en el ámbito académico, debido a que además de su gran número de librerías de Visión, dispone de capacidad de conexión rápida y sencilla con hardware de distintos fabricantes, para integrar con PLCs, SCADAS, cámaras, etc. Este hecho, lo convierte en un software muy atractivo en el ámbito académico, donde los recursos de hardware suelen ser limitados, ya que mediante un único software se pueden crear implementaciones muy interesantes.



8. Bibliografía.

- [1] <http://www.automatas.org/redes/scadas.htm>
- [2] ftp://ftp.ehu.es/cidira/dptos/depjt/Instrumentacion/BK-ANGEL/10_LabVIEW/Instrumentos_virtuales.PDF
- [3] “**Ingeniería de la Automatización Industrial**”, Ramón Piedrafita Moreno. Ed. RA-MA S.A.
- [4] “**Autómatas Modicon Micro TSX 3705/ 3708/ 3710/ 3720 Manual de puesta en marcha**”, Tomo 1, Marzo 2005, *Schneider Electric*.
- [5] “**Autómatas Modicon Micro TSX 3705/ 3708/ 3710/ 3720 Manual de puesta en marcha**”, Tomo 2, Marzo 2005, *Schneider Electric*.
- [6] “**Autómatas Modicon Micro TSX 3705/ 3708/ 3710/ 3720 Manual de puesta en marcha**”, Tomo 3, Marzo 2005, *Schneider Electric*.
- [7] “**Configuración del Modicon TSX Micro**”, Manual de Curso Centro de Formación Técnica *Schneider Electric Argentina*
- [8] “**TSX37_Manual de referencia**”, Descripción del software PL7, *Schneider Electric*.
- [9]: “**NI 622x Specifications**”, *National Instruments*.
- [10] “**Datasheet Signal Relays**”, *TE Connectivity (TE)*.
- [11] “**Automatismos eléctricos programables**”, Oriol Boix Aragonés, Miquel A. Saigí Grau, Ferran Zabaleta Alañá.
- [12] <http://www.elai.upm.es/webantigua/>
- [13] **Visión por Computador**. Asignatura Grado Ingeniería Electrónica y Automática, EUPT (Universidad de Zaragoza).
- [14] <http://visionartificial.fpcat.cat/wp-content/uploads/Conocimientos.pdf>
- [15] <http://www.dzoom.org.es/la-exposicion-la-primera-clave-de-una-buena-fotografia-ahmf31-dia7/>
- [16] **NI Vision for LabVIEW™ User Manual**, *National Instruments*
- [17] “**Getting Started with the LabVIEW Datalogging and Supervisory Control Module**”, *National Instruments*.
- [18] <http://www.escet.urjc.es/~visionc/VisionPorComputador.pdf>

Otra Información en la Web:

<http://www.ni.com>

<http://relays.te.com/definitions>

<http://www.schneiderelectric.es>

<http://www.opcspain.org>

<http://www.elai.upm.es/webantigua/>

http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf

<http://www.visiononline.es/es/actualidad-en-vision-artificial/la-vision-artificial-y-su-influencia-en-espana>