

Prototipo de un proceso industrial de control de calidad con visión artificial

Trabajo Fin de Grado
Grado en Ingeniería Eléctrica

Autor: Misael R. Sandoval Salvatierra

Tutor: D. Juan Ramón Rufino Valor

Co-tutor: D. Jaime Masiá Vañó

Curso: 2016-2017

RESUMEN

El presente documento recoge información para la realización de un prototipo para el estudio de un proceso industrial de control de calidad, mediante el control automático y la tecnología de visión artificial. Realizando el diseño, fabricación y puesta en marcha de la misma, con el objetivo de desarrollar un entorno multidisciplinar, práctico y eficaz aplicable en la industria moderna o con fines educativos.

La importancia de este trabajo, radica en la flexibilidad que tiene el sistema en la adaptación del entorno industrial, así como el amplio abanico de disciplinas tecnológicas e ingenieriles.

El prototipo consta de una Estación, la cual, parte de una base de aluminio mecanizada para dicho fin sobre la que van situados diferentes módulos con diversos dispositivos y componentes, los cuales conforman una instalación automática y electroneumática.

El control de la instalación, se gestiona y desarrolla mediante el software LabVIEW, cuya lista de instrucciones y ejecución de órdenes programadas son procesadas por un dispositivo embebido de bajo coste (NI myRIO), ideal para control e instrumentación, la cual cuenta con un procesador programable que ejecuta un sistema operativo en tiempo real, así como un FPGA personalizado.

La visión artificial o visión por computador, está compuesta por dos plataformas diferentes, la primera de ellas mediante el software y la herramienta ya antes mencionada LabVIEW y NI myRIO, y la segunda comprendida por un computador de placa reducida (Raspberry Pi 3) empleando programación en C++ y librerías de OpenCV.

SUMMARY

This document shows the realization of a prototype for the study of an industrial process and quality with the technology of automatic control and artificial vision. The design, manufacture and start-up are all detailed, it has as a main aim a multidisciplinary, practical and effective environment applicable in modern industry.

The most relevant point in this work is the flexibility of the system in the adaptation of the industrial environment, as well as the range of technological and engineering disciplines applies for this purpose.

The prototype consists of making a station, which has a base made in aluminum on which are located different modules with different devices and components connected with an electro-pneumatic installation.

The control of the installation is developed by the LabVIEW software, whose list of instructions and execution of programmed orders are processed by an embedded device (NI myRIO), ideal for control and instrumentation, which has a processor Programmable system that runs a real-time operating system, as well as a custom FPGA.

Artificial vision or computer vision, is composed of two different platforms, the first one using the software previously mentioned LabVIEW and NI myRIO respectively and the second one is comprised by a reduced board computer (Raspberry Pi 3) with C++ and using the OpenCV libraries.

PALABRAS CLAVE

SolidWorks, KiCAD, myRIO, LabVIEW, Raspberry Pi 3, Opencv, C++

KEYWORDS

SolidWorks, KiCAD, myRIO, LabVIEW, Raspberry Pi 3, Opencv, C++

TABLA DE CONTENIDO

1.	Introducción	9
1.1	Descripción general	10
1.2	Objetivos.....	11
2.	Fundamentos Teóricos	12
2.1	Calidad	12
2.2	Control de calidad industrial	13
2.3	Automatización industrial.....	14
2.4	Introducción a la visión artificial	15
3.	Descripción de la Estación.....	16
3.1	Introducción.....	16
3.2	Sistema a implementar	16
3.2.1	Lógica de la Automatización mediante NI myRIO (Grafcet).....	20
3.2.2	Lógica de la Visión mediante NI myRIO	22
3.3	Diseño	23
3.3.1	Modelado 3D.....	23
3.3.2	Diseño de la Placa PCB.....	28
3.4	Piezas para el procesado	32
3.4.1	Piezas con Código QR.....	32
3.4.2	Piezas con Marcaje.....	33
3.5	Impresión 3D	34
4.	Montaje	35
4.1	Base	35
4.2	Unidad de mantenimiento y Manómetro.....	35

4.3 Electroválvulas	36
4.3.1 Electroválvulas de presión positiva	36
4.3.2 Electroválvula de vacío	37
4.4 Circuito de potencia.....	37
4.5 Eyector	38
4.6 Sensores	38
4.6.1 Sensores Magnético para el controlador neumático Lineal	39
4.6.2 Sensores Magnético para el controlador neumático Doble vástago	40
4.6.3 Sensores Magnético para el controlador neumático Rotativo	40
4.6.4 Sensores Magnético para el controlador neumático Paletas	41
4.7 Racores y tubo neumáticos	42
4.8 Instalación Eléctrica - Electrónica.....	43
4.8.1 Esquema eléctrico de alimentación y emisión de datos de los sensores.....	43
4.8.2 Esquema eléctrico de las válvulas y su circuito de activación	44
4.8.3 Esquema eléctrico de los switches y Leds de señalización	45
4.9 Instalación neumática.....	46
4.10 Actuadores del sistema de automatización	47
4.10.1 Actuador Neumático de doble vástago	47
4.10.2 Actuador Neumático lineal	48
4.10.3 Actuador Neumático Base rotacional	49
4.10.4 Actuador Neumático de Paletas.....	49
4.11 Componentes para la automatización y el tratamiento de imágenes	50
4.11.1 Sistema myRIO.....	50
4.11.2 Dispositivo Raspberry Pi 3	52
4.11.3 Cámara Logitech c270	54

4.11.4	Cámara RaspiCam	55
4.11.5	Iluminación.....	56
4.12	Otros dispositivos	59
4.12.1	Fuente de alimentación 24 v	59
4.12.2	Toma de aire a presión	60
5.	Herramientas empleadas.....	61
5.1	Para el Mecanizado.....	61
5.1.1	Taladradora.....	61
5.1.2	Sierra de Cinta horizontal.....	62
5.2	Eléctricas / Electrónicas.....	63
5.2.1	Estación soldadora	63
6.	Software.....	64
6.1	LabVIEW 2016	64
6.2	SolidWORKS 2016	68
6.3	KiCAD	69
7.	Programación	70
7.1	Programa de control myRIO, LabVIEW.....	70
7.1.1	Diagrama funcional en LabVIEW	73
7.1.2	Módulo de visión myRIO, LabVIEW.....	74
7.2	Programa visión Raspberry Pi, OpenCV y C++	79
7.2.1	Código de la programación en c++	80
8.	Presupuesto.....	85
9.	Pruebas en Laboratorio	86
9.1	Muestras del Procesamiento de imagen	86
9.1.1	tratamiento de la imagen con myRIO y LabVIEW	86

9.1.2 tratamiento de la imagen con Raspberry Pi 3 y C++	91
10. Conclusión	93
11. Bibliografía	94
12. Anexos	95

1. INTRODUCCIÓN

La evolución de la tecnología desde sus inicios, siempre se ha caracterizado por su desarrollo constante a un ritmo vertiginoso con el propósito de facilitar tareas abarcando prácticamente cualquier sector. Así como también la optimización de actividades, las cuales son comprendidas desde la ejecución de simples tareas repetitivas, hasta actividades que pueden ser perjudiciales para la salud.

Desde finales del siglo XX en el sector industrial, se han implantado sistemas que aporten eficiencia en el control de procesos con el objetivo de mejorar la rentabilidad y la producción del mismo.

La innovación por parte de empresas en el sector, están cada vez más a la orden del día, dado que es vital para mantener la competitividad en el mercado. Esto se traduce en que hoy en día es normal la aparición de nuevas formas de diseño y control, donde el objetivo principal es lograr una ventaja en situaciones donde la tecnología tradicional no consigue alcanzar o suplir las especificaciones requeridas.

Es por ello que en el presente documento se plantea la realización del diseño, programación y puesta en marcha de un prototipo para un proceso industrial de control de calidad mediante visión artificial.

1.1 DESCRIPCIÓN GENERAL

En presente documento recoge toda la información necesaria para el diseño, montaje y puesta en marcha de un prototipo de proceso industrial para el control de calidad mediante el sistema de visión por computador para la inspección de anomalías y almacenamiento de una pieza en cuestión. Dicho sistema está compuesto por dos tecnologías diferentes de procesamiento de imágenes, una de ellas es mediante el sistema embebido NI myRIO empleando programación en LabVIEW y la segunda está comprendida por el microordenador de placa única o placa reducida Raspberry Pi 3 empleando librerías OpenCV y programación en C++.

La pieza será posicionada en diferentes etapas de la estación para el procesamiento de imagen y almacenamiento, la cual será posible mediante la implantación de un sistema electro-neumático comprendido por sensores magnéticos y actuadores de diferentes tipos conectados a una Placa de circuito impreso (PCB) diseñada para cubrir esta necesidad. Así pues, todo lo relacionado con el control automático, será programado mediante el NI myRIO mencionado anteriormente. El proceso del sistema de inspección a grandes rasgos, será el siguiente:

La pieza a inspeccionar, partirá de una base diseñada en 3D, la cual será extraída mediante un sistema de acople a un actuador neumático de doble vástago encargado de la expulsión de la pieza. A continuación se procederá a realizar el primer procesamiento de imagen, la cual se realizará mediante una cámara Logitech C270 y el NI myRIO con la programación en LabVIEW.

En el caso de que haya un código QR impreso sobre la pieza, éste será decodificado y la pieza será almacenada mediante un actuador neumático de tipo lineal hasta el 'Depósito A'. Si por el contrario no existe código QR, ésta será posicionada para un segundo procesamiento de imagen, la cual será realizada mediante una cámara Raspicam y el uso de la Raspberry Pi 3 con programación en C++ comunicada mediante la señal sincronizada de Trigger con el NI myRIO. Tras la adquisición de la imagen ésta detectará si existe algún tipo de anomalía en la pieza inspeccionada y devolverá señales en consecuencia. En el caso de que la pieza tenga alguna anomalía, ésta será almacenada en el 'Deposito C'. Si por el contrario la pieza no tiene ninguna anomalía, será almacenada en el 'Depósito B'. Los almacenajes se realizan gracias al conjunto de, un eyector de vacío, un actuador neumático de tipo paletas y otro de tipo base rotativa.

El proceso de la estación, puede verse de forma detallada en la figura 3.1 representada en el apartado "3.2 Sistema a implementar" del presente documento.

1.2 OBJETIVOS

El objetivo principal es el llevar a cabo conceptos teóricos multidisciplinares a la práctica mediante un prototipo industrial flexible, para la mejora de producción y/o implantación de un sistema cuyo objetivo sea la optimización de procesos mediante el uso de la tecnología visión por computador o visión artificial. Así como también su posible uso con fines didácticos para futuros alumnos de la UPV Campus d'Alcoi, más concretamente en el ámbito de la mecánica y la automática

2. FUNDAMENTOS TEÓRICOS

2.1 CALIDAD

Para hablar sobre el concepto de calidad, tenemos que mencionar que los primeros estudios documentados sobre la calidad se hicieron en los años 30 antes de la Segunda guerra Mundial. En aquel entonces, la calidad no mejoró significativamente, pero a raíz de ello, se realizaron los primeros experimentos para lograr que ésta mejorase.

Así pues, podemos decir que los estudios sobre calidad se hicieron en Estados Unidos. En el año 1933 el Doctor Walter A. Shward, de los Bell Laboratories, aplicó el concepto de control estadístico de proceso por primera vez con propósitos industriales; su objetivo era mejorar en términos de costo-beneficio las líneas de producción, el resultado fue el uso de la estadística de manera eficiente para elevar la productividad y disminuir los errores, con la intención de mejorar la calidad y elevar la productividad.



FIGURA 2.1.1 *Inspección de munición segunda guerra mundial*

A raíz de ello, la calidad en el transcurso de los años ha experimentado un cambio significativo hasta llegar a convertirse en lo que hoy en día conocemos por calidad.

En concepto de organizaciones industriales, desde principios de este siglo, y a grandes rasgos desde mucho antes, se entendía calidad como:

"El grado en que un producto cumplía con las especificaciones técnicas que se habían establecido cuando fue diseñado".

En el transcurso del tiempo, este concepto fue evolucionando, el cual la norma UNE 66-001 define como:

"La adecuación al uso del producto o más detalladamente, el conjunto de propiedades y características de un producto o servicio que le refieren su aptitud para satisfacer las necesidades expresadas o implícitas".

Más recientemente, este concepto ha derivado a todos los ámbitos de la organización, llegándose a definir como:

"Todos los métodos a través de los cuales la organización satisface las necesidades y expectativas de sus clientes, empleados, entidades implícitas y todo lo relacionado con su entorno".

Como podemos ver, esta segunda definición comprende de la combinación de todas las definiciones que refieren a dicho concepto, ya que el objetivo principal de ésta es la evolución hacia la mejora enfatizando la eficacia.

2.2 CONTROL DE CALIDAD INDUSTRIAL

El control de calidad juega un rol de vital importancia para el desarrollo industrial, ya que es un factor imprescindible para el logro de objetivos en todos los ámbitos de su entorno.

Es por ello que podemos clasificar en concepto, la definición de control de calidad como el proceso de control que realiza o participa en la caracterización de los nuevos productos o servicios en sus diferentes fases de desarrollo y en el establecimiento de las especificaciones de calidad de los mismos. Del mismo modo que desarrolla, ejecuta o coordina la ejecución de los métodos de ensayo para determinar las características de calidad de las materias primas, materiales, productos intermedios y productos finales.



FIGRUA 2.2.1 *Inspección Control de Calidad de una pieza industrial*

Este concepto forma parte de la Gestión de Calidad, la cual es más mucho más amplia. En este documento nos centraremos más en este último concepto, ya que es un pilar fundamental desde el punto de vista industrial y práctico a efectos de competencia.

2.3 AUTOMATIZACIÓN INDUSTRIAL

El ser humano por instinto en aspectos ergonómicos, siempre ha buscado la creación de herramientas y en su defecto, máquinas que le faciliten la realización de tareas peligrosas, tediosas y repetitivas.

En los últimos tiempos, la aparición de máquinas altamente sofisticadas ha dado lugar a un gran desarrollo del campo de la automatización y el control de las tareas, aplicado ya en muchas máquinas que se manejan diariamente.

El principal objetivo de la Automatización industrial es generar la mayor cantidad de productos o de un conjunto de estos, en el menor tiempo posible, con el fin de reducir los costes y garantizar una uniformidad en la calidad.

La automatización industrial, está compuesta por distintas tecnologías, que en su conjunto la hacen posible, como por ejemplo la Instrumentación que nos ayuda a realizar mediciones de la materia, motores y servomotores que ayudan a la ejecución de movimientos y Controladores Lógicos Programables (PLC) encargados de la realización de secuencias, entre otros .

En cuanto al impacto social, es un pensamiento muy común que la automatización es una fuente directa de desempleo. Sin embargo, el desempleo es causa por políticas económicas de las empresas como por ejemplo, prescindir de los servicios de un operador en lugar de cambiar sus tareas, que quizás ya no sean de máxima concentración en el proceso sino tareas de supervisión del proceso. La realidad es que, los trabajadores son desplazados y por lo general son contratados para otras tareas dentro de la misma empresa, o bien en el mismo trabajo en otra organización que todavía no se ha automatizado. Esto se puede establecer porque existen muchos trabajos donde no existe riesgo inmediato de la automatización. Ningún dispositivo ha sido inventado que pueda competir contra el ojo humano para la precisión y certeza en muchas tareas; tampoco el oído humano. Los seres humanos pueden identificar y distinguir mayor cantidad de esencias que cualquier dispositivo automático.

2.4 INTRODUCCIÓN A LA VISIÓN ARTIFICIAL

El inicio de la visión artificial, desde el punto de vista práctico, fue marcado por Larry Roberts, el cual, en 1961 creó un programa que podía "ver" una estructura de bloques, analizar su contenido y reproducirla desde otra perspectiva, demostrando así a los espectadores que esa información visual que había sido mandada al ordenador por una cámara, había sido procesada adecuadamente por él.

En los últimos años, los sistemas de producción han evolucionado considerablemente buscando con ahínco la maximización de la productividad. Todas las nuevas filosofías y técnicas de fabricación tienen entre sus objetivos principales maximizar la calidad, un factor clave para la mejora de competitividad.

Por ello, se ha denominado la visión artificial como una herramienta importante en el desarrollo industrial, ya que con ella se pretende reemplazar o complementar las inspecciones manuales, mediante el uso de cámaras digitales y procesamientos de imágenes.

3. DESCRIPCIÓN DE LA ESTACIÓN

3.1 INTRODUCCIÓN

A lo largo de la historia, el ser humano siempre ha buscado un método o una herramienta capaz de mejorar eficazmente y sin un exceso de coste, la facilidad de la realización de diferentes tareas en distintos ámbitos, aunque con mucho mayor interés en el sector de la industria.

No es hasta la segunda mitad del siglo XVIII, después de la Revolución Industrial que se empieza a llevar a la práctica esta filosofía con el fin de maximizar ganancias, mediante la creación de prototipos, optimización de procesos y creación de los mismos.

Es por ello que los prototipos industriales, son representaciones de un objeto para el control y la supervisión de procesos, que permiten la comprobación de diferentes estudios en un entorno real, explorando su uso y permitiendo discutir y valorar la viabilidad de éste.

Para llevar a cabo, los objetivos planteados, procederemos al diseño, montaje y el control de un prototipo para un proceso industrial mediante el uso de la visión artificial.

3.2 SISTEMA A IMPLEMENTAR

A la hora de controlar e inspeccionar un proceso industrial, se necesita estudiar varios métodos para garantizar su efectividad y viabilidad.

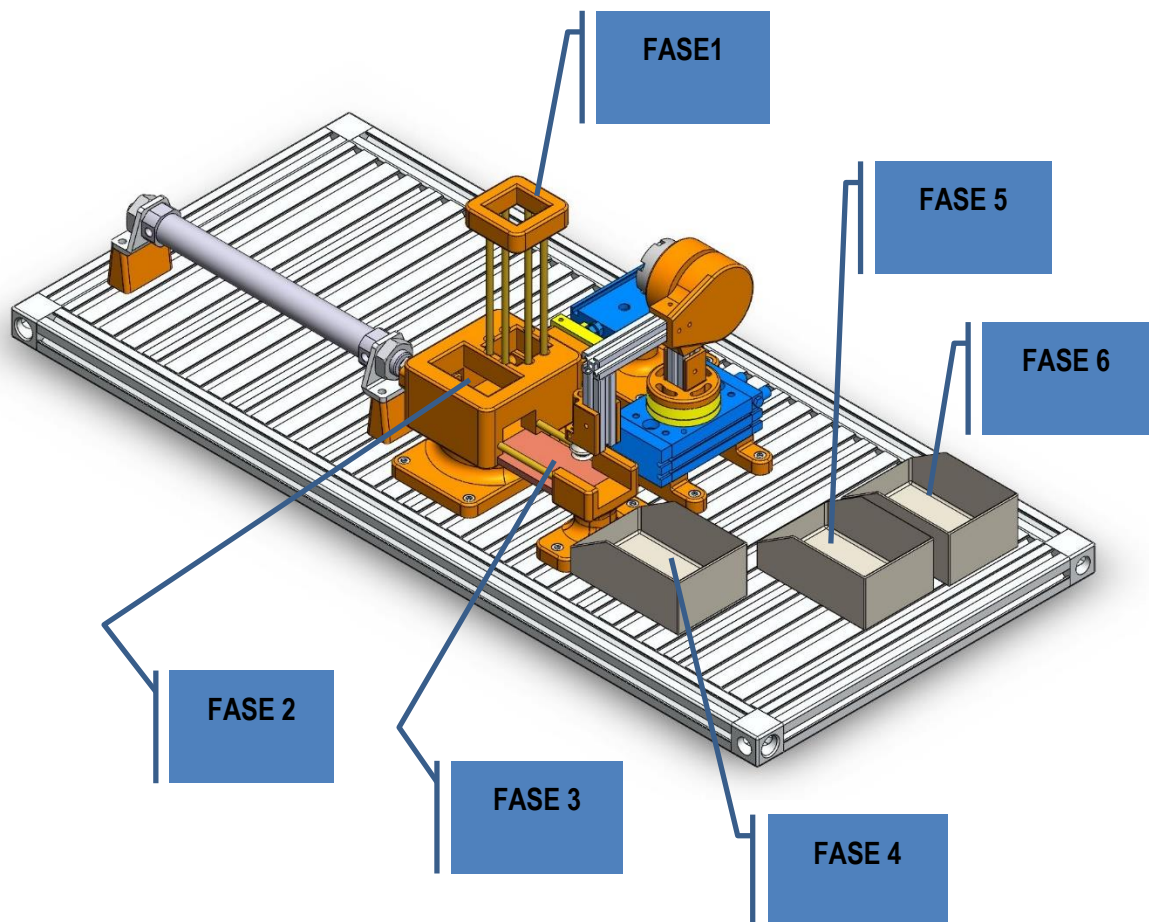
En este caso, hemos optado por el desarrollo de una plataforma en forma de estación para la evaluación de piezas y su almacenamiento, controlado mediante un sistema electroneumático e inspeccionado mediante visión artificial provenientes de dos tecnologías diferentes mencionadas anteriormente.

El manejo de actuadores es controlado por un fluido a presión, en este caso aire. Esto se ha conseguido mediante el desarrollo de una estación con dispositivos alimentados a +3.3V y +24V, mediante el diseño de una PCB diseñada y fabricada para tal fin, conectado a un sistema de hardware embebido (myRIO), mediante el que se manejará el software de control.

La estación, consta de una base sobre la cual es posible colocar diferentes actuadores neumáticos, eléctricos y electrónicos, con el fin de someter a las piezas a distintas inspecciones y distribuciones.

Durante el desarrollo de este capítulo se detallarán las especificaciones que caracterizan la forma y funcionamiento de esta estación. Se explicará el diseño que se ha llevado a cabo para la fabricación de la misma y se comentará uno a uno todos los dispositivos que lo integran, explicando su funcionamiento y la labor que desempeñan. Por último se dará una visión general sobre las instalaciones, electrónicas y electroneumática que forman los distintos elementos de la estación, así como los sistemas de visión, explicando sus respectivos funcionamientos.

El prototipo a diseñar, se trata de una estación, la cual está dividida en 4 diferentes Etapas comprendidas por diversas funciones y subrutinas para el procesamiento y la inspección de una pieza en concreto cuyas características, diseño y dimensiones se verán reflejados a lo largo de éste capítulo.



FIGRUA 3.2.1 Estación y sus fases generales

- **FASE 1**

Esta etapa es la inicial, ya que en ella se efectúa la alimentación de las piezas a procesar, dicha acción es realizada manualmente por el operario, que deposita la pieza por la parte superior de la base principal (Ver Figura 3.2.1), para ello se diseñó un mecanismo, el cual tiene la funcionalidad de guiado y posicionamiento de la pieza a procesar, permitiendo de forma efectiva la expulsión de ésta para su posterior procesamiento.

Para la expulsión de la pieza procesar hemos empleado un actuador neumático de doble vástago cuyas especificaciones técnicas se ven reflejadas en el apartado "4.10.1 Actuador Neumático de doble vástago"

La ejecución del movimiento eficaz de alimentación y expulsión de la pieza a procesar, se resolvió mediante el diseño un acople para poder realizar las tareas de contención de las piezas apiladas y la expulsión de la misma instalándola en el cabezal de los vástagos, cumpliendo así con su tarea asignada.

- **FASE 2**

Esta es la primera fase de procesamiento de la imagen, la cual se obtiene mediante una cámara Logitech 260C que captura la imagen y envía la información al myRIO, dispositivo mediante el cual realizamos todo el proceso automático y una parte del tratamiento de la imagen del cual hablaremos más específicamente en el apartado 3.4 de éste capítulo.

En este punto mediante la visión artificial, podremos decodificar un código QR y posicionar la pieza en la Etapa 4.

- **FASE 3**

En el caso de que la pieza en cuestión no disponga de código QR en el procesamiento de imagen en la Etapa 2. Se procederá a posicionar la pieza en ésta etapa, la cual tiene como misión procesar la imagen mediante el computador de placa reducida Raspberry Pi 3 cuyas características y especificaciones técnicas se encuentran en el apartado 3.5 de éste capítulo.

- **FASE 4**

Ésta etapa hace la función de Depósito A, en el cual son alojadas las piezas con códigos QR decodificadas previamente mediante el sistema embebido myRIO visto en detalle en el capítulo 3.5 de éste capítulo.

- **FASE 5**

Ésta etapa hace la función de Depósito B, en el cual van alojadas las No Defectuosas cuyo procesamiento de imagen se realiza mediante el computador de placa reducida Raspberry Pi 3

- **FASE 6**

Ésta etapa hace la función de Depósito C, en el cual van alojadas las Defectuosas procesadas mediante el computador de placa reducida Raspberry Pi 3

Para un mejor entendimiento de las etapas del proceso, se ha diseñado un diagrama de flujo para las etapas y sus transiciones correspondientes, ilustrado en el apartado “7.1 Programa de control myRIO, LabVIEW” de éste documento.

3.2.1 LÓGICA DE LA AUTOMATIZACIÓN MEDIANTE NI MYRIO (GRAFSET)

Para la realización de la lógica de la automatización se ha empleado este sistema de diagrama funcional, la cual está compuesta por diferentes acciones de cilindros pertenecientes a los actuadores neumáticos, así como también el uso de sus respectivos sensores magnéticos.

Para poder entender mejor la lógica, veremos el siguiente croquis mediante el cual se ven reflejados las posiciones de reposo y los accionamientos de los mismos.

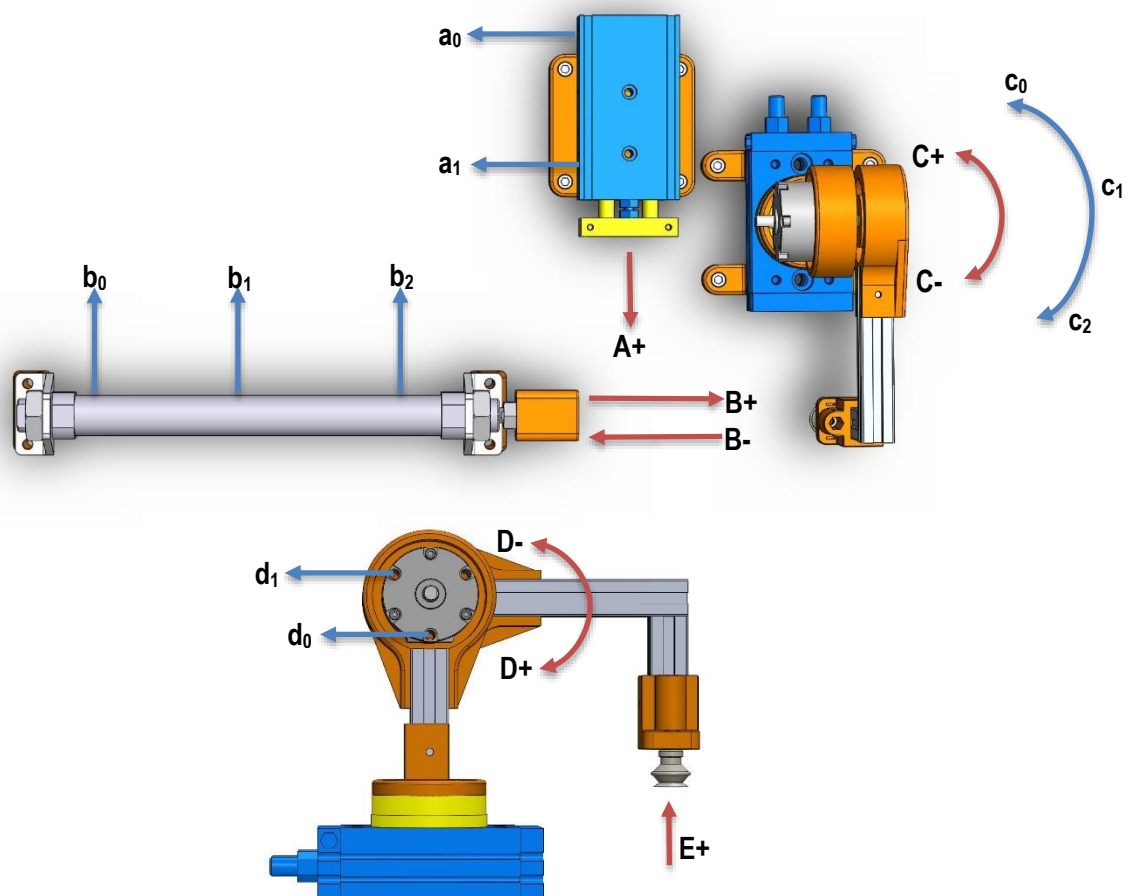


Figura 3.2 Movimiento y posicionamiento de los actuadores mediante los sensores

Electroválvula	Actuador	Tipo
A	Doble vástago	Monoestable
B	Lineal	Biestable
C	Base rotativa	Biestable
D	Paletas	Monoestable
E	Eyector de vacío	Monoestable

TABLA 3.1 Nomenclatura de los actuadores neumáticos

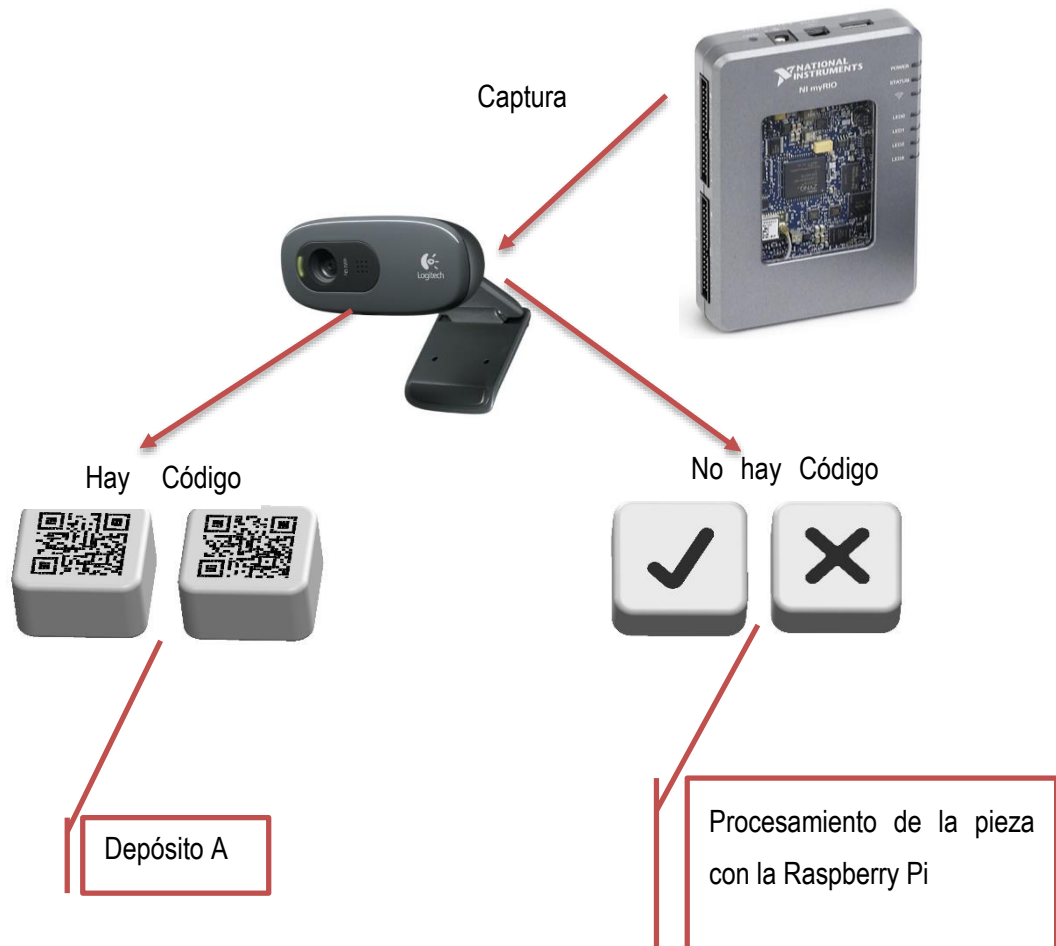
Sensor	Posicionamiento de los actuadores
a ₀	Doble vástago Posición Inicial
a ₁	Doble vástago Posición Expulsar pieza
b ₀	Lineal Posición Inicial
b ₁	Lineal Posición Pieza No Código
b ₂	Lineal Posición Pieza con Código
c ₀	Rotacional Posición Coger pieza No Código
c ₁	Rotacional Posición Pieza OK
c ₂	Rotacional Posición Pieza No Ok
d ₀	Paletas Posición a 0°
d ₁	Paletas Posición a 90°
M/P	Pulsadores de Marcha y Paro

TABLA 3.2 Nomenclatura de los sensores y posicionamiento para los actuadores

3.2.2 LÓGICA DE LA VISIÓN MEDIANTE NI MYRIO

Como se ha mencionado anteriormente, la lógica de la visión mediante en myRIO, decodifica todas las piezas que dispongan de código QR, las cuales serán almacenadas en el Depósito A. Si por el contrario, la pieza a inspeccionar no dispone de dicho código, ésta será posicionada para una segunda inspección mediante la Raspberry Pi.

A continuación vemos un croquis mediante el cual podemos observar a grandes rasgos, el proceso del tratamiento de visión mediante el myRIO.



3.3 DISEÑO

Se han diseñado piezas a medida para llevar a cabo con éxito este prototipo. A continuación se ven reflejadas todas ellas

3.3.1 MODELADO 3D

Esta pieza fue diseñada para cubrir la necesidad de la creación de una pieza desde un punto de vista práctico y eficiente para realizar el tratamiento de la pieza a procesar, partiendo desde el almacenamiento, la extracción y finalmente el procesamiento de la misma.



Figura 3.4 Varilla de aluminio para el calibrado

Esta pieza tiene como función la de facilitar la alimentación del almacén con las piezas a procesar alojadas en el almacén de la base principal, las cuales son unidas entre sí mediante varillas de Aluminio de 6 mm que facilitan su posicionamiento.

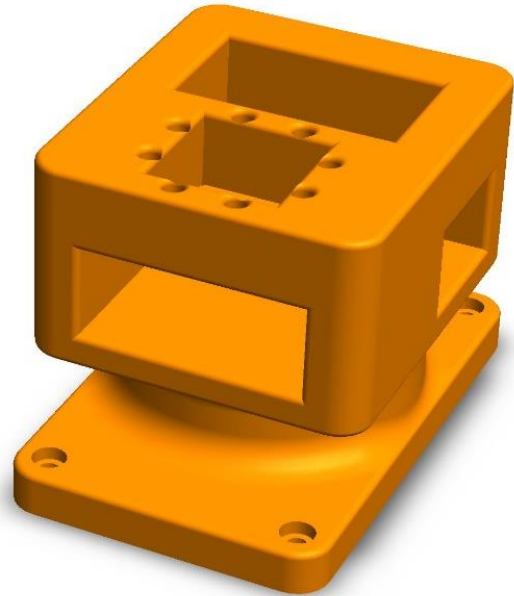


Figura 3.3 Base Principal

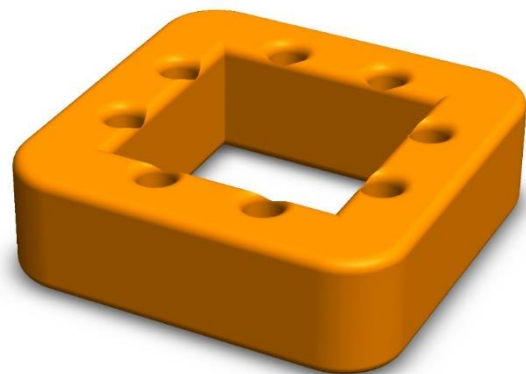


Figura 3.5 Unión calibrado alimentación de la Base Principal

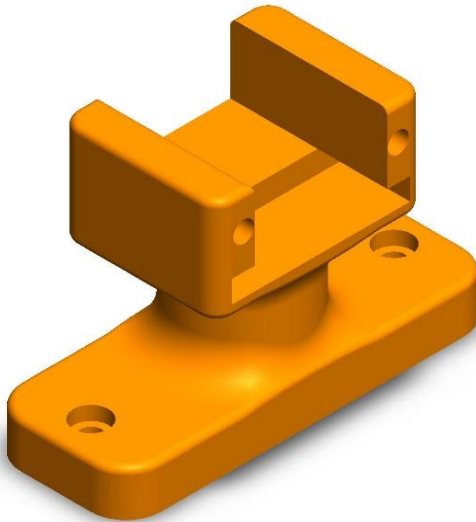


Figura 3.6 Unión calibrado alimentación de la Base Principal

Mediante esta pieza podemos comunicar la base principal mediante el calibrado sirviendo de guía para la pieza a procesar empleando varillas de 6mm para dicha unión y una pletina de Aluminio.

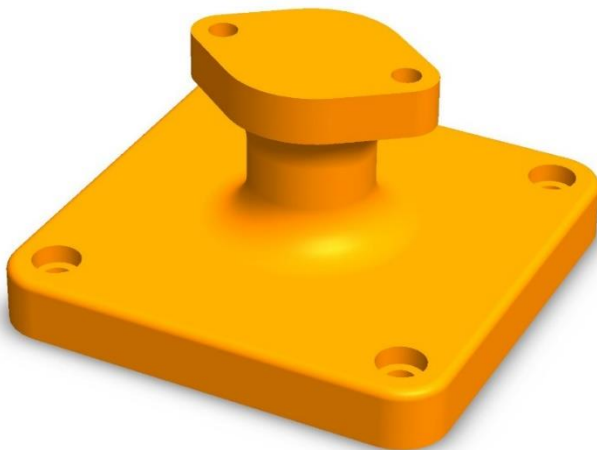


Figura 3.7 Soporte para el Cilindro de doble vástago

Esta pieza se diseñó para el soporte del Cilindro de doble vástago mediante el cual la pieza a procesar es expulsada desde el almacén para su posterior tratamiento.

Soporte A del Cilindro neumático de paletas, el cual permite la extensión desde la base del Cilindro automático rotacional empleando un perfil de aluminio de 20x20mm hasta la extensión del soporte de la ventosa mediante la comunicación con el Soporte B.

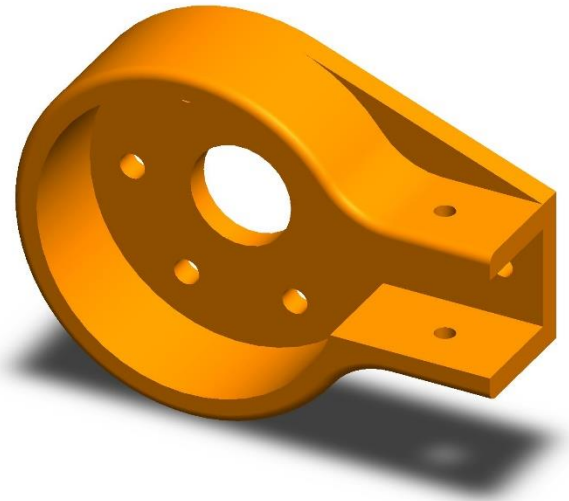
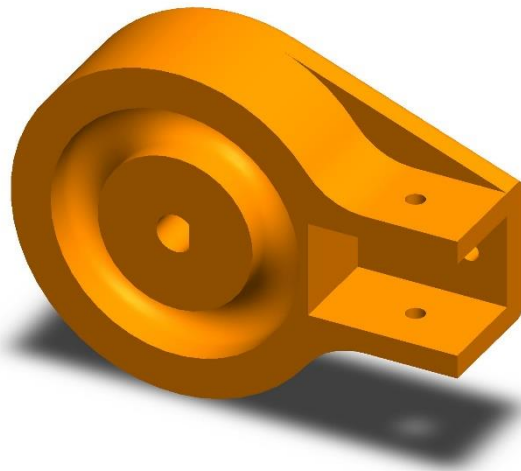
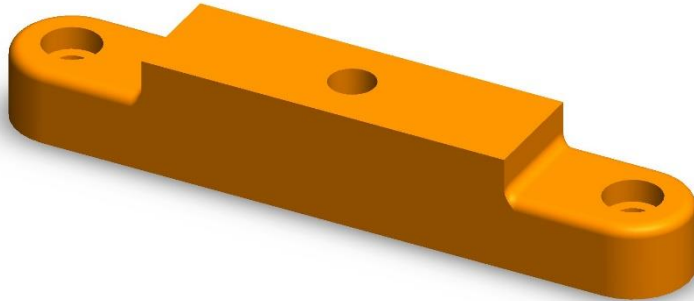


Figura 3.8 Soporte para la extensión del cilindro de Paletas



Soporte B del Cilindro neumático de paletas, el cual permite la ejecución del movimiento y de comunicación directa con el perfil de 20 x 20mm el cuál sirve de punto de sujeción del soporte para la ventosa

Figura 3.9 Soporte para la comunicación del Cilindro rotacional con el Cilindro de Paletas



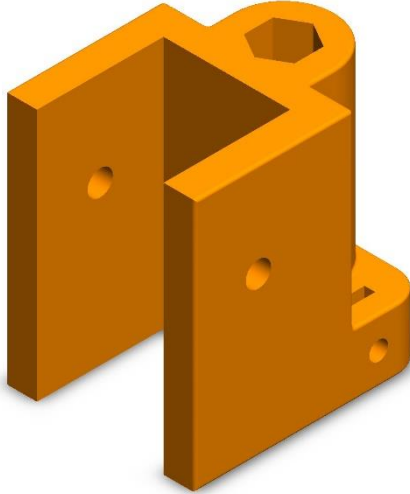
Pieza diseñada para el soporte del cilindro neumático rotacional, la cual permite la sujeción a la base de la estación

Figura 3.10 Soporte para la sujeción del Cilindro neumático rotacional



Esta pieza fue diseñada para servir de soporte al Cilindro neumático rotacional y para la extensión mediante el perfil de aluminio con los soportes del cilindro neumático de Paletas. Gracias a este diseño, el cilindro dispone de una mayor flexibilidad de posicionado y una versatilidad de ejecución de movimientos

Figura 3.11 Soporte para base del Cilindro Rotacional



Este diseo es para la sujeci3n de la Ventosa, la cual es encargada de manipular la pieza a procesar mediante un eyector y una v3lvula de vaci3

Figura 3.12 Soporte para la sujeci3n de la Ventosa

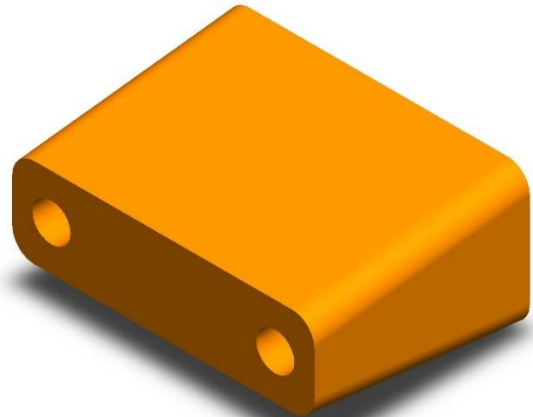


Figura 3.13 Soporte para el cilindro neum3tico lineal

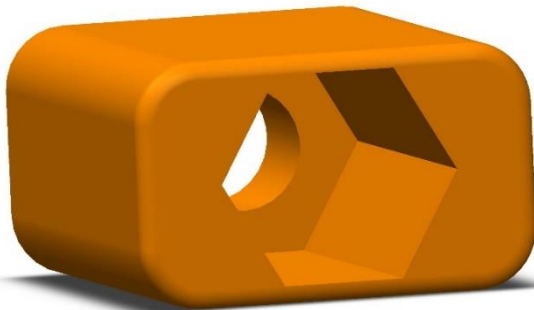


Figura 3.14 Acople para el cilindro neum3tico lineal*

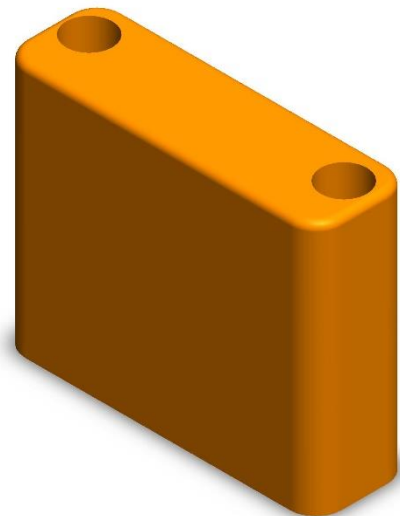


Figura 3.15 Acople para el cilindro neum3tico de doble v3stago*

[*] Acoples diseoados para solventar la problem3tica relacionada con la alimentaci3n y el posicionamiento de la pieza a procesar.

3.3.2 DISEÑO DE LA PLACA PCB

Los bloques de las válvulas, así como la alimentación de los sensores funcionan con una tensión de alimentación de 24 V, lo que supone una corriente elevada en comparación con el resto de componentes electrónicos de la estación, ya que los dispositivos de myRIO y Raspberry Pi disponen de una tensión de 3.3V y 5 V. Por lo tanto se optó por diseñar una Placa de Circuito Impreso (PCB) para realizar la alimentación de los diferentes sensores y actuadores, además de la comunicación entre los dispositivos.

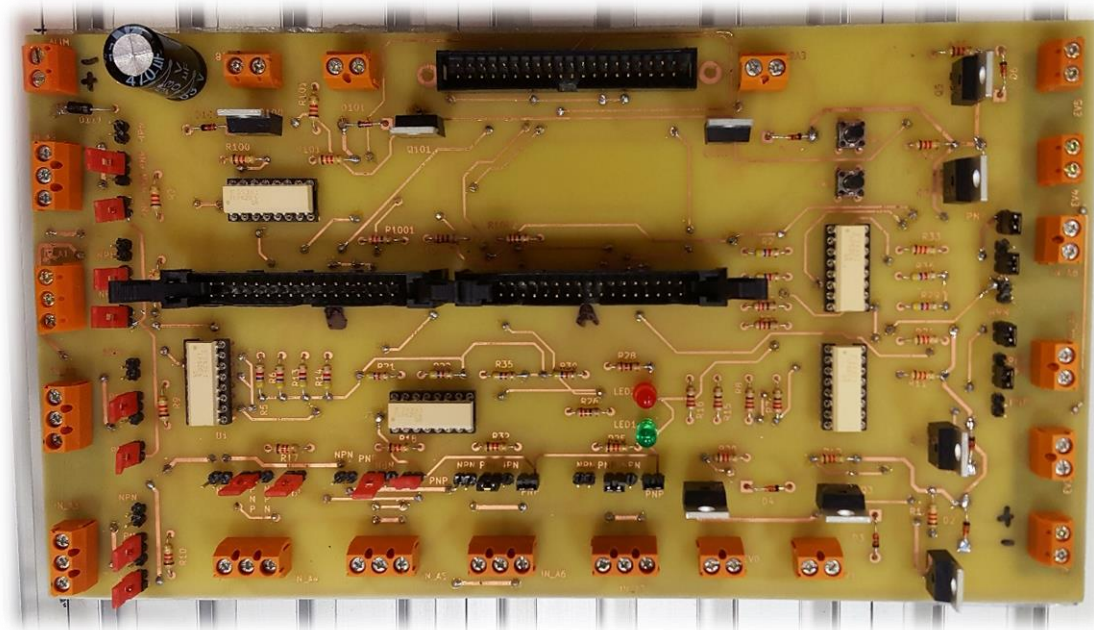


Figura 3.16 PCB con sus respectivos mecanismos conectados

Para el diseño de la PCB se ha empleado el Software open-source Kicad, cuyas características se ven reflejadas en el apartado 6.2.3 Kicad

Cabe destacar de que en el diseño del PCB, se ha previsto la posibilidad de la conexión de sensores tipo PNP y NPN indiferentemente, empleando jumpers para la conmutación, resultando así un sistema aún más flexible.

Además, se han instalado dos pulsadores, que servirán para la puesta en marcha y el paro del proceso. Así como también dos Leds de señalización.

En el apartado Anexo Esquema PCB se detalla todo el conexionado y la nomenclatura de las entradas y salidas que componen esta placa de circuito impreso.

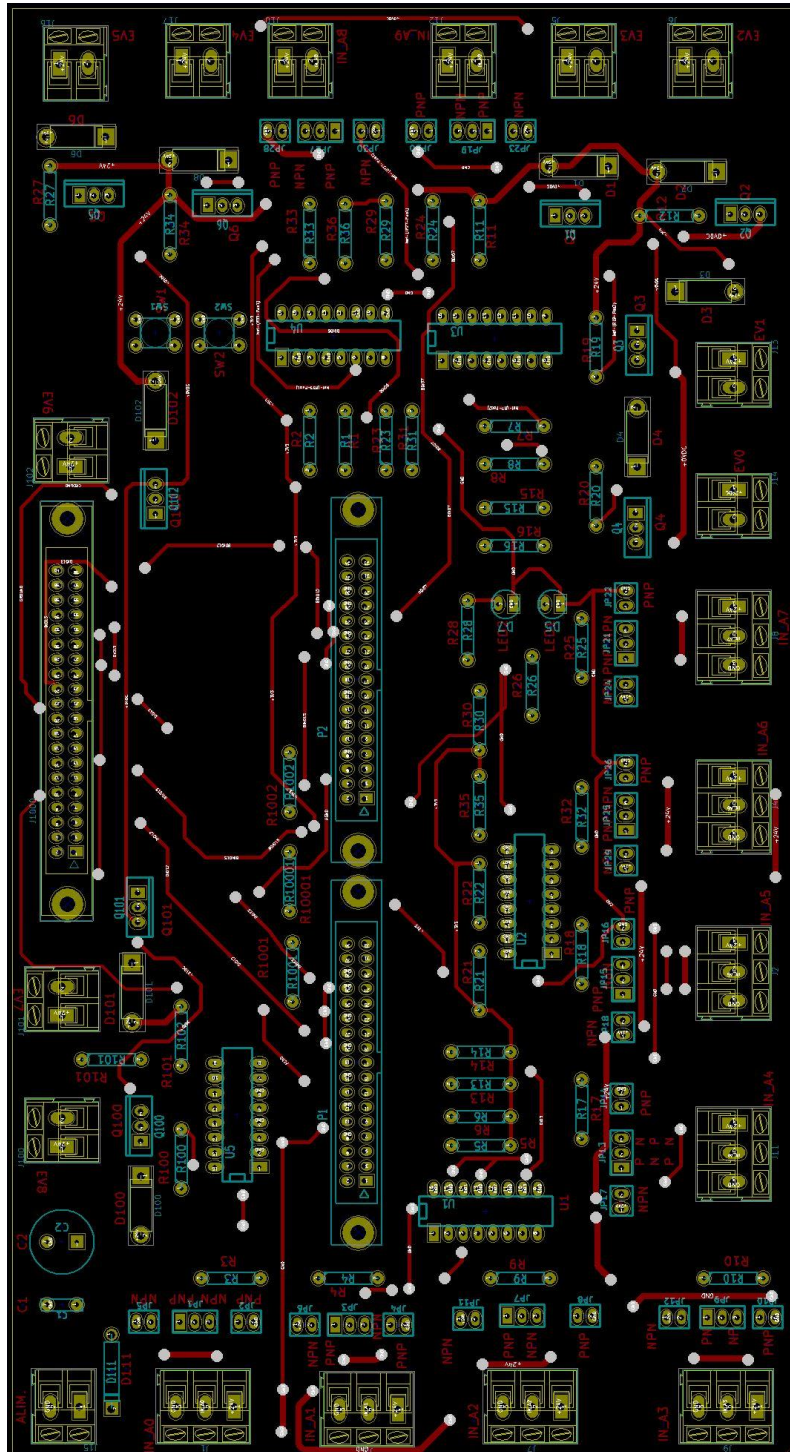


Figura 3.17 PCB conexiones en la cara de cobre superior

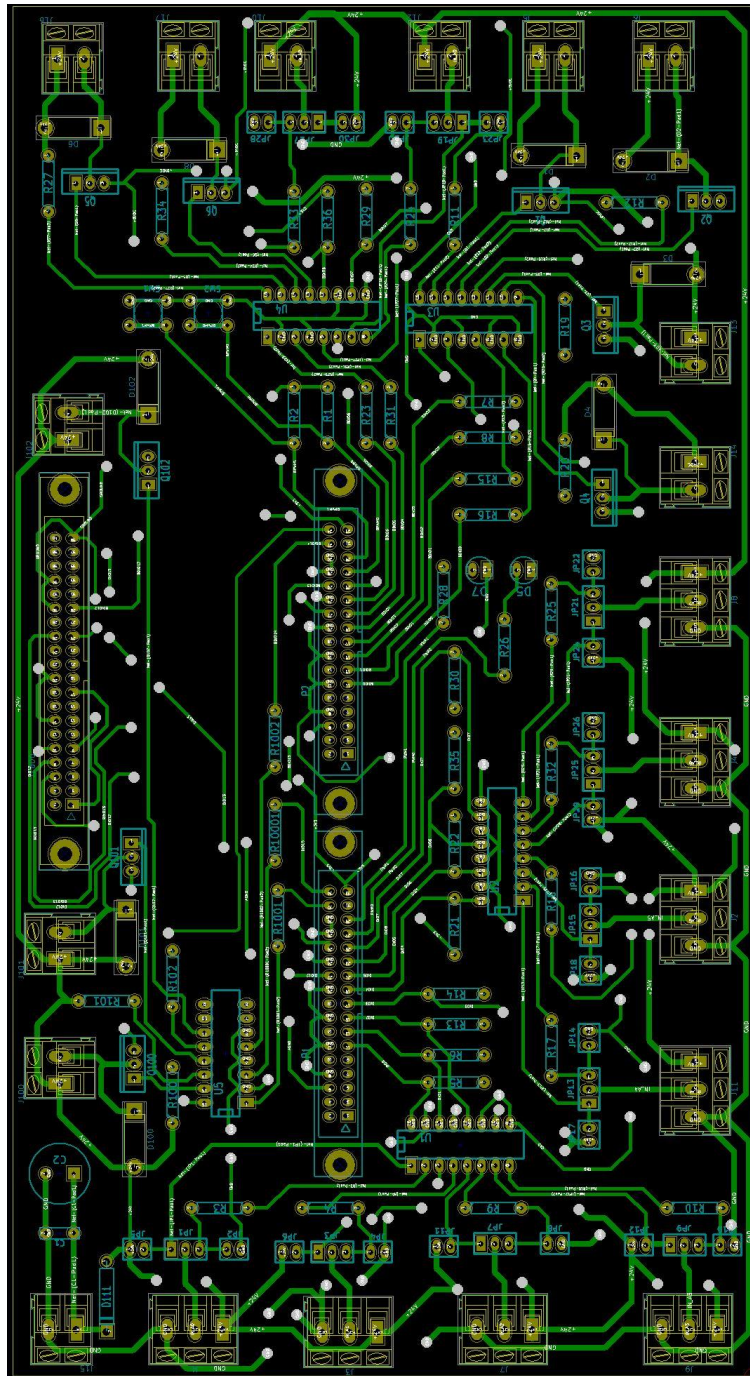


Figura 3.18 PCB conexiones en la cara de cobre inferior

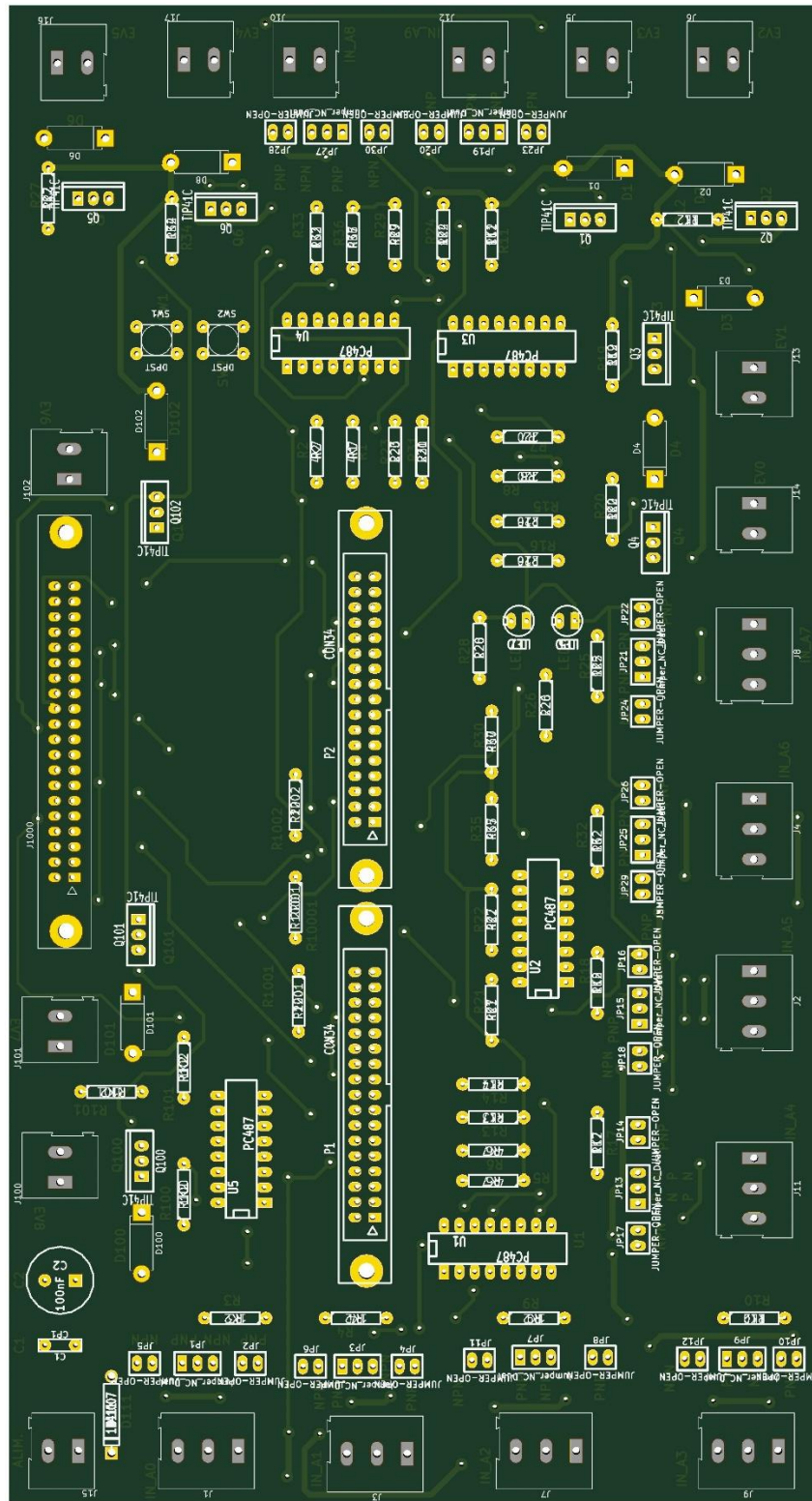


Figura 3.19 PCB en diseño 3D

3.4 PIEZAS PARA EL PROCESADO

Para llevar a cabo el procesado y la inspección de las piezas, se planteó el diseño de cuatro diferentes tipos de piezas, empleando dos métodos diferentes.

El primer método consta de códigos QR los cuales indican si se trata de Pieza A o de una Pieza B. El segundo, parte de una señalización, la cual indica si la pieza es Pieza correcta o Pieza defectuosa.

A continuación veremos cada una de ellas y de sus características.

3.4.1 PIEZAS CON CÓDIGO QR

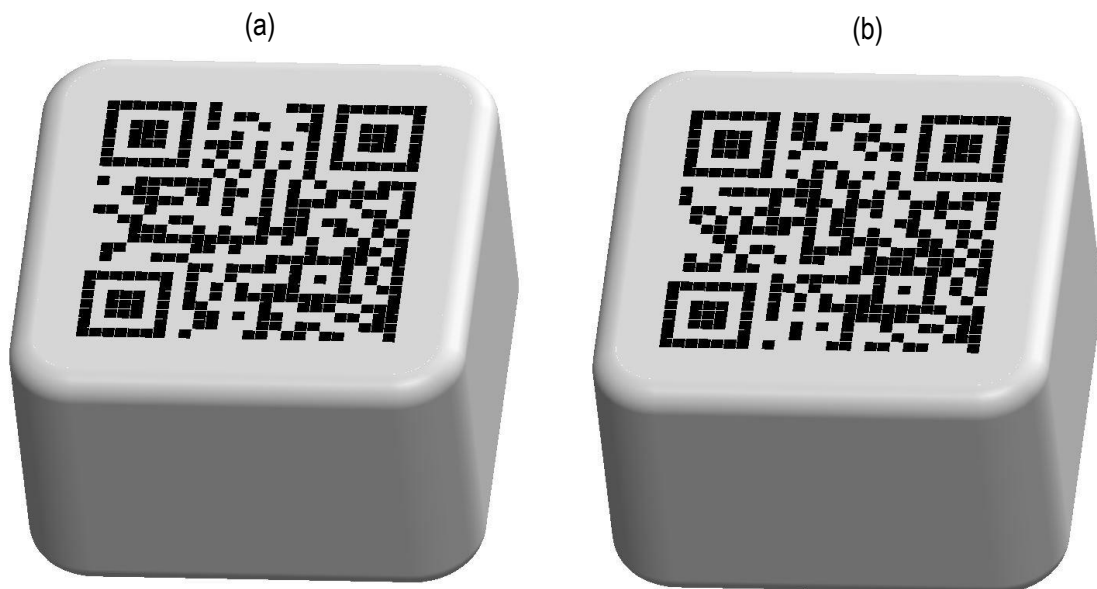


Figura 3.20 Pieza a procesar con Código QR (a) Pieza A; (b) Pieza B

Como se ha mencionado anteriormente, estas piezas están diseñadas con el fin de poder ser decodificadas mediante el empleo de visión artificial.

En éste caso, las piezas son decodificadas mediante el NI myRIO y LabVIEW

3.4.2 PIEZAS CON MARCAJE

Estas piezas fueron diseñadas para el procesamiento e inspección ejecutadas por la Raspberry Pi 3, la cual mediante el empleo de librerías OpenCV y la programación en C++, es capaz de diferenciar un tipo de pieza u otro y enviar la comunicación para la ejecución automática del proceso de la estación.

En esta etapa del procesamiento podemos encontrarnos con estos dos tipos de piezas, (a) que indica que la pieza está en buen estado y la (b) que indica que tiene algún defecto.

De esta manera simularíamos el proceso de identificación de defectos de la pieza, implementando la misma filosofía, pero evitando tener que romper o crear un defecto físico a la pieza en cuestión.



Figura 3.21 *Pieza a procesar Sin Defecto*

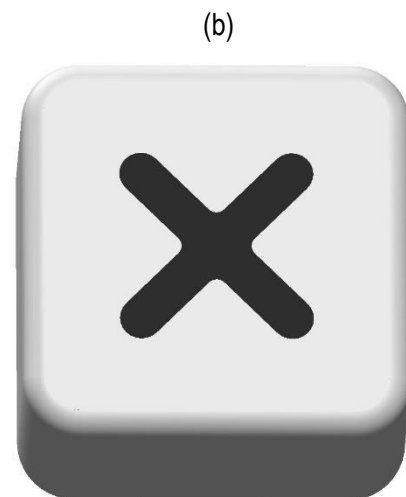


Figura 3.22 *Pieza a procesar Defectuosa*

3.5 IMPRESI3N 3D

Para la impresi3n de las piezas dise1adas mencionadas anteriormente, se emple3 una impresora 3D profesional, "Dimension Elite".



Figura 3.23 Impresora 3D Dimension Elite

Una de las ventajas a destacar, es la flexibilidad de imprimir piezas con geometrías complejas. Adem1s al disponer de un sistema SCA-1200, el cual contiene sosa en su interior creando un proceso de agitaci3n a una temperatura determinada que dan el resultado de una limpieza 3ptima de la pieza.

4. MONTAJE

4.1 BASE

La base principal sobre el que va montado todos los componentes de automática y visión está compuesta por perfiles de aluminio de 30 x 100 mm con una longitud de 330 mm encuadrados por perfiles de 30 x 30 mm con una longitud de 800 mm y 330 mm. Esta base presenta un mecanizado realizado especialmente para ésta función.

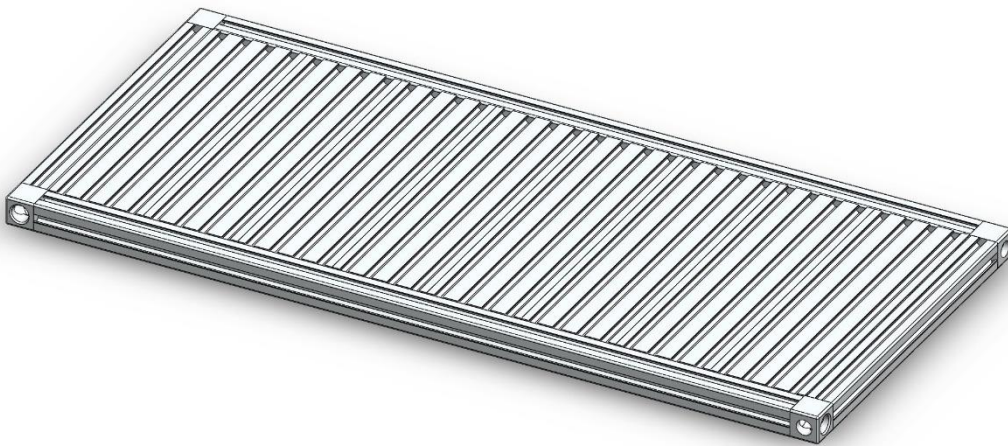


Figura 4.1 Base mecanizada

4.2 UNIDAD DE MANTENIMIENTO Y MANÓMETRO

El manómetro se encarga de regular la presión a la entrada de la estación. Consta de una llave de paso que abre o corta el flujo de aire, y una rueda que regula la presión a la que deja entrar en el circuito el aire a presión, de manera que ante posibles variaciones de presión en el exterior de la estación, el manómetro impide que haya picos de presión que puedan sobrepasar la presión deseada en el interior del circuito.

El manómetro admite una presión máxima de 230 psi / 1.6 MPa, superando de esta forma con creces la presión máxima con la que trabaja la estación.



Figura 4.2 Unidad de mantenimiento y *Manómetro*

4.3 ELECTROVÁLVULAS

Para la automatización de éste prototipo, emplearemos electroválvulas tanto de presión positiva, para realizar la automatización del prototipo.

El diseño neumático y electrónico del ésta estación, ha sido diseñado para activar 5 válvulas de forma parcial y/o simultánea.

4.3.1 ELECTROVÁLVULAS DE PRESIÓN POSITIVA

Estas electroválvulas se encargan de abrir y cerrar el flujo del aire que le llega al actuador neumático cuando se desea realizar un control de presión positiva. Las válvulas van conectadas directamente a un racor para la conexión de un total de 5 Electroválvulas que a su vez están conectadas al manómetro, el cual determina la presión a la que el aire atravesará las válvulas.

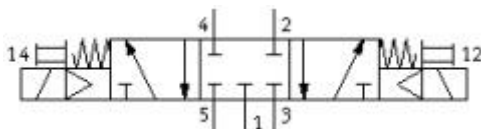


Figura 4.3 *Electroválvula SMC 5/3 centro cerrado SY5*

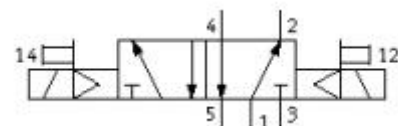


Figura 4.4 *Electroválvula SMC 5/2 SY5*

4.3.2 ELECTROVÁLVULA DE VACÍO

Las electroválvulas para el control de vacío no son necesarias para éste prototipo, ya que mediante su caudal, se pueden suplir las demandas de presión negativa requerida.

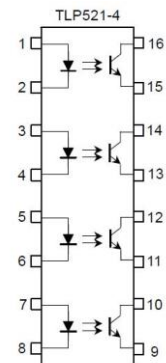
4.4 CIRCUITO DE POTENCIA

Los bloques de las válvulas funcionan con una tensión de alimentación de 24 V, lo que supone una corriente alta en comparación con el resto de componentes electrónicos de la estación. Para evitar posibles fugas de altas corrientes a elementos del circuito con corrientes de funcionamiento más bajas, se ha diseñado un circuito óptico de manera que el accionamiento de las válvulas quede aislado física y eléctricamente.

Esto se consigue mediante un circuito con optoacopladores. Un optoacoplador es un dispositivo que utiliza una corriente eléctrica para activar un diodo, el cual satura un fototransistor, por el que circulará una corriente colector - emisor mucho mayor. A su vez, la corriente que circule por este fototransistor servirá para excitar la puerta de un transistor de potencia, el cual, al saturarse, activará la electroválvula.



TO-220



1, 3, 5, 7 : Anode
 2, 4, 6, 8 : Cathode
 9, 11, 13, 15 : Emitter
 10, 12, 14, 16: Collector

El funcionamiento detallado de este circuito eléctrico será explicado más adelante en este mismo capítulo, en el apartado "4.8 Instalación eléctrica".

4.5 EYECTOR

Los eyectores son los dispositivos encargados de crear el vacío. Estos funcionan aplicando el principio de Venturi, el cual al circular aire por una tobera principal. Al llegar a una zona donde la tobera presenta un diámetro menor, se generará una zona de baja presión. Este punto, conectado a una segunda tobera, succionará el aire, generando así el vacío deseado.



Figura 4.5 Eyector de vacío SMC serie ZU

Los eyectores instalados en el banco de ensayos son el modelo SMC ZU07S, capaces de conseguir una presión de -84 kPa con respecto a la presión atmosférica, vacío suficiente para las aplicaciones de ésta estación.

4.6 SENSORES

Los sensores empleados en ésta estación son los recomendados por el fabricante, son de tipo magnéticos, de alimentación 24V, de 3 hilos PNP y de 2 hilos.

A continuación veremos con más detalle los tipos de sensores empleados para cada tipo de actuador neumático.

4.6.1 SENSORES MAGNÉTICO PARA EL CONTROLADOR NEUMÁTICO LINEAL



Figura 4.6 Sensor PNP D-H7A2 con abrazadera para el cilindro lineal

D-H7A2	
Tipo	Estado sólido
Indicador LED	Sí
Protección IP	IP67
Tensión de Alimentación	4.5 → 28V dc
Mínima Temperatura de Funcionamiento	-10°C
Máxima Temperatura de Funcionamiento	+60°C
Funcionamiento	PNP
Características técnicas	

Dispondremos de un total de tres sensores de éste tipo, que van instalados sobre el actuador lineal de tipo Lineal.

- b_0 : Indica que la posición del actuador está en Posición Inicial (Cilindro dentro)
- b_1 : Indica que la pieza ha sido posicionada para el segundo procesamiento de imagen
- b_2 : Indica que el actuador ha posicionado la pieza en el Depósito A

4.6.2 SENSORES MAGNÉTICO PARA EL CONTROLADOR NEUMÁTICO DOBLE VÁSTAGO



Figura 4.7 Sensor PNP D-H7PW para el cilindro de doble vástago

D-Y7P	
Compatible con	MGP, MH, Serie CXS
Tipo	Estado sólido
Funcionamiento	PNP
Indicador LED	Sí
Tensión Nominal	4.5 → 28V dc
Caída de Tensión	≤0.8V
Protección IP	IP67
Máxima Temperatura de Funcionamiento	+60°C
Mínima Temperatura de Funcionamiento	-10°C
Funcionamiento	PNP
Características técnicas	

Para este tipo de sensor dispondremos de un total de dos sensores, los cuales irán instalados en el actuador neumático de doble vástago.

- a_0 : Indica que la posición del actuador está en Posición Inicial (Cilindro dentro)
- a_1 : Indica que la pieza ha sido expulsada para el primer procesamiento de imagen

4.6.3 SENSORES MAGNÉTICO PARA EL CONTROLADOR NEUMÁTICO ROTATIVO



Figura 4.8 Sensor PNP D-M9PW para el cilindro de base rotativa

D-M9	
Tipo	Estado sólido
Funcionamiento	PNP
Indicador LED	Sí
Tensión Nominal	4.5 → 28V dc
Caída de Tensión	≤0.8 V @ 10 mA, ≤2 V @ 40 mA
Protección IP	IP67
Máxima Temperatura de Funcionamiento	+60°C
Mínima Temperatura de Funcionamiento	-10°C
Características técnicas	

Este tipo de sensor es empleado sobre el cilindro neumático de base rotativa con un total de tres sensores, los cuales indican

- c_0 : Indica que la pieza está a punto para su almacenaje en los depósitos B o C
- c_1 : Indica que la pieza está lista para ser almacenada en el depósito C
- c_2 : Indica que la pieza está lista para ser almacenada en el depósito B

4.6.4 SENSORES MAGNÉTICO PARA EL CONTROLADOR NEUMÁTICO PALETAS



Figura 4.9 Sensor 2 hilos D-T92L para el cilindro de paletas

D-T79	
Compatible con	NCRB
Tipo	Estado sólido
Indicador LED	Sí
Tensión Nominal	4.5 → 28V dc
Protección IP	IP67
Mínima Temperatura de Funcionamiento	-10°C

Máxima Temperatura de Funcionamiento	+60°C
Nº de hilos	2
Características técnicas	

4.7 RACORES Y TUBO NEUMÁTICOS

Los racores son las conexiones que utilizan los distintos dispositivos neumáticos para acoplar las tuberías. Ésta estación utiliza distintos tipos de racores y de diferentes medidas dependiendo del tubo a conectar.

Las medidas de los tubos utilizados son de 4, 6 y 8 mm de diámetro exterior.

4.8 INSTALACIÓN ELÉCTRICA - ELECTRÓNICA

La instalación eléctrica se encarga de alimentar a todos los dispositivos eléctricos de la estación, conectarlos al sistema embebido myRIO y al computador de placa reducida Raspberry Pi.

Los dispositivos eléctricos de la estación constan de una tensión de alimentación 5V y/o 3,3 V proporcionados por el myRIO y la Raspberry Pi.

Por otra parte, disponemos de 24 V proporcionados por la fuente de alimentación, para el conexionado de las válvulas y para la alimentación de los sensores magnéticos, de ésta manera podremos actuar sobre ellos o recibir su información.

A lo largo de este apartado se mostrará el esquema eléctrico/electrónico de los componentes de la estación, indicando cómo están alimentados y conectados.

4.8.1 ESQUEMA ELÉCTRICO DE ALIMENTACIÓN Y EMISIÓN DE DATOS DE LOS SENSORES

Para la activación de los sensores, se ha diseñado un circuito de activación con el fin de aislar físicamente al microcontrolador, ya que éstas funcionan a una tensión y corriente mayores. Con esto se consigue proteger al myRIO de algún pico de tensión o alguna posible fuga de corriente.

Para ello empleamos el optoacoplador TLP521 y resistencias de diferentes valores mediante el cual tenemos un total de 10 conectores destinados para los sensores para la alimentación y transmisión de datos de las mismas.

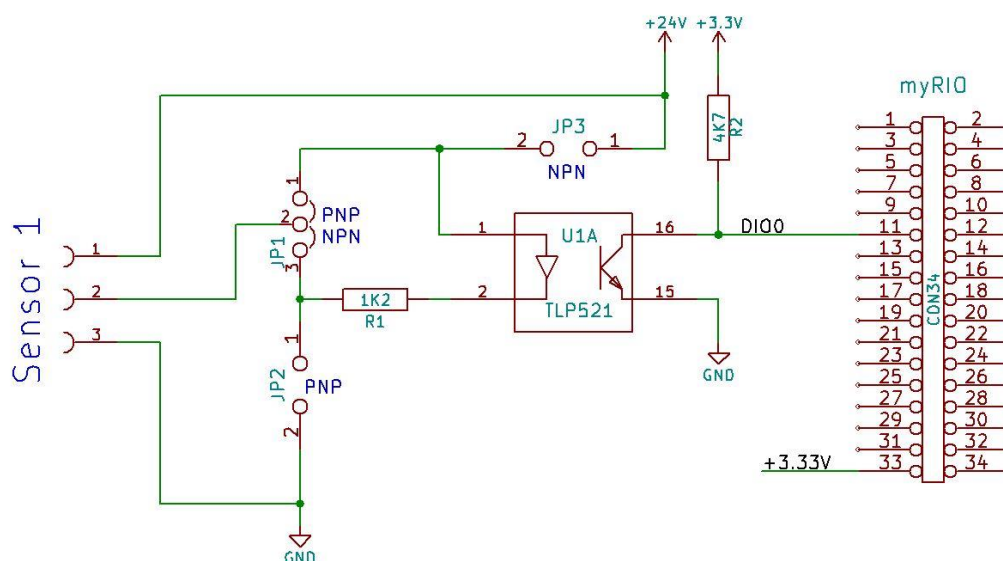


Figura 4.10 Circuito eléctrico para el conexionado de los sensores

- Cálculo de R1

$$R_1 = \frac{V_{CC} - V_F}{I_F} = \frac{24 - 1.15}{0.02} = 1142.5 \Omega$$

Se ha elegido una resistencia de 1k2

- Cálculo de R2

$$R_2 = \frac{V_+ - V_{CE(SAT)} - V_{EB}}{I_F} = \frac{3.3 - 0.2 - 1.18}{0.01} = 192 \Omega$$

El valor típico del transistor en saturación es de 0.2 V. Así que finalmente se ha optado por una resistencia de 4k7

4.8.2 ESQUEMA ELÉCTRICO DE LAS VÁLVULAS Y SU CIRCUITO DE ACTIVACIÓN

Al igual que el diseño de la conexión de los sensores, hemos procedido a emplear optoacopladores con el fin de controlar corrientes y tensiones mayores a las permitidas por el myRIO.

Además de lo anterior, el circuito de alimentación para cada electroválvula, consta de un optoacoplador TLP521, un transistor TIP41C y resistencias de distintos valores. En total se han instalado 5 circuitos como este para la activación simultánea de hasta 5 válvulas y 2 auxiliares para uso general. En la siguiente imagen se muestra el circuito eléctrico de activación para una de las electroválvulas con sus correspondientes cálculos:

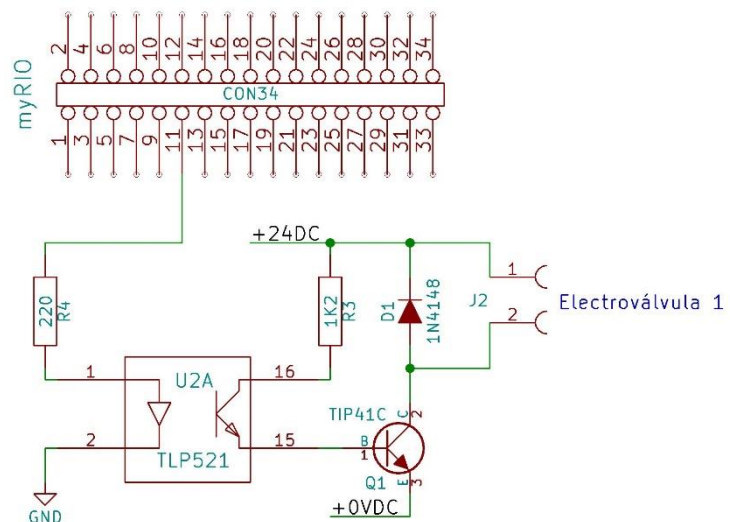


Figura 4.11 Circuito eléctrico del conexionado de la electroválvula

- Cálculo de R3

$$R_3 = \frac{V_{CC} - V_{CE(SAT)} - V_{EB}}{I_F} = \frac{24 - 0.4 - 1.2}{0.02} = 1120 \Omega$$

Finalmente, se optó por una resistencia de 1K2

- Cálculo de R4

$$R_4 = \frac{V_+ - V_F}{I_F} = \frac{3.3 - 1.18}{0.01} = 212 \Omega$$

4.8.3 ESQUEMA ELÉCTRICO DE LOS SWITCHES Y LEDS DE SEÑALIZACIÓN

Los 2 switches y los 2 LEDs de señalización van situados en el mismo PCB de la estación. Todos presentan una alimentación común de 3.3 V y un GND también común a todos ellos. A su vez, cada Switch va conectado a un pin configurado como entrada y cada LED a una salida digital del myRIO.

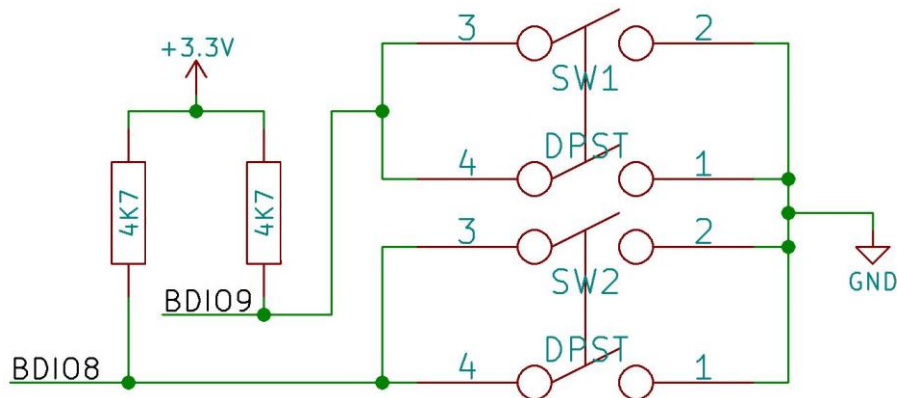


Figura 4.12 Circuito eléctrico del conexionado de los switches

Se han empleado valores de resistencias típicos de 4k7 para los pulsadores

4.9 INSTALACIÓN NEUMÁTICA

La estación está diseñada para regular y controlar hasta un máximo de 5 válvulas, las cuales son 2 de tipo doble efecto con 5 vías y 3 posiciones, y 3 de simple efecto con 5 vías y 2 posiciones, todas ellas montadas sobre un rack para una mayor optimización del espacio.

La estación recibe el aire proveniente del compresor, mediante una unidad de mantenimiento y un manómetro regulamos la presión con la que trabajará la estación. A continuación el aire se distribuye a las diferentes válvulas mediante el rack mencionado anteriormente.

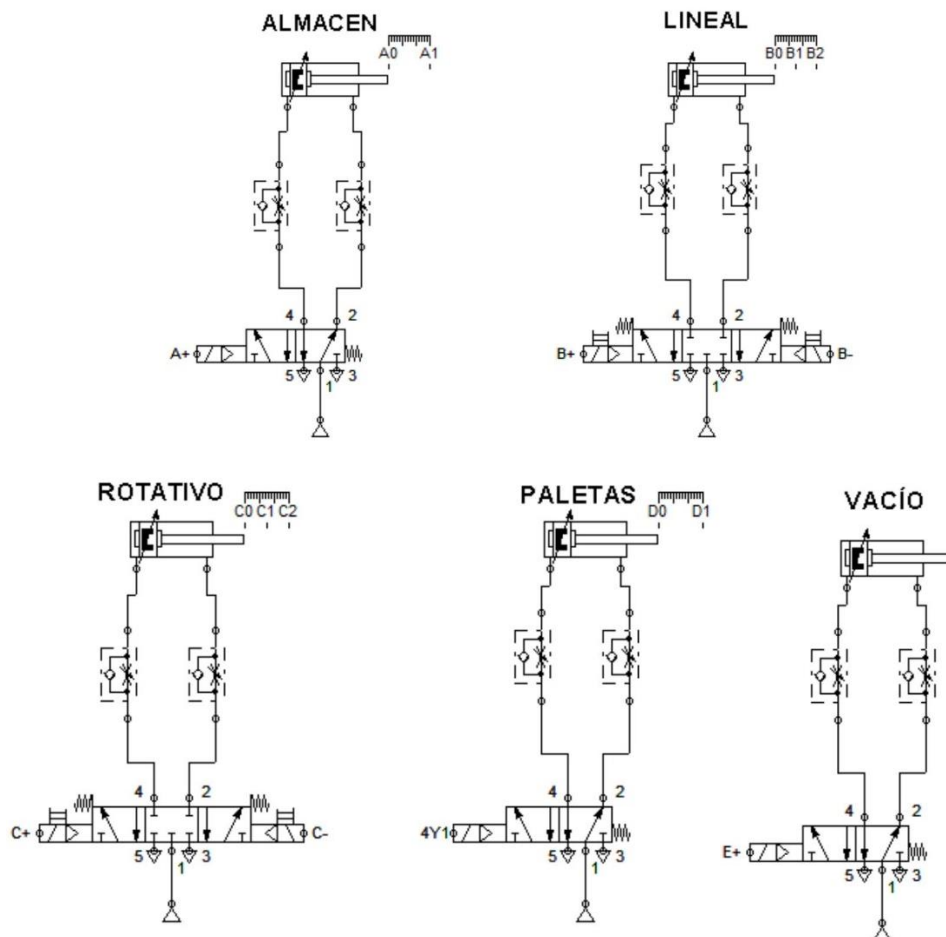


Figura 4.13 Instalación neumática general

Como podemos observar, ésta sería la instalación neumática general de la estación representada con sus diferentes sensores y accionamientos.

4.10 ACTUADORES DEL SISTEMA DE AUTOMATIZACIÓN

Para llevar a cabo la automatización de la estación, se ha empleado el sistema embebido myRIO del cual se detalla en el apartado 5 Herramientas y componentes empleados.

La ejecución de los movimientos de la automatización, es lograda gracias a los diferentes actuadores neumáticos,

4.10.1 ACTUADOR NEUMÁTICO DE DOBLE VÁSTAGO

Este cilindro como ya se ha mencionado anteriormente, será el encargado de realizar la expulsión de la pieza a procesar.



Figura 4.15 Actuador neumático de Doble Vástago CXSM20-50

Serie del Fabricante	CXSM
Varillas	Doble
Calibre	20mm
Recorrido	50mm
Máxima Presión de Funcionamiento	0,7 MPa
Tipo de Amortiguamiento	Caucho
Conexión de Puerto	M5 x 0.8
Máxima Temperatura de Funcionamiento	+60°C
Longitud	144mm
Anchura	25mm
Altura	64mm
Mínima Temperatura de Funcionamiento	-10°C

4.10.2 ACTUADOR NEUMÁTICO LINEAL

Este tipo de actuador es empleado para el posicionamiento y almacenamiento de la pieza a procesar.



Figura 4.16 Actuador neumático Lineal CD85N25-200-B

Serie del Fabricante	C85
Acción	Doble
Varillas	Único
Calibre	25mm
Recorrido	200mm
Máxima Presión de Funcionamiento	1 MPa
Presión Máxima Absoluta	1.5MPa
Tipo de Amortiguamiento	Caucho
Dimensiones del Cuerpo	33.5 (Dia.) x 337mm
Máxima Temperatura de Funcionamiento	+60°C
Mínima Temperatura de Funcionamiento	-10°C
Longitud	337mm
Diámetro	33.5mm

4.10.3 ACTUADOR NEUMÁTICO BASE ROTACIONAL

Este tipo de actuador es empleado para el almacenamiento de la pieza en los depósitos A y B.



Figura 4.17 Actuador neumático Rotativo CD85N25-200-B

Serie del Fabricante	MSQ
Calibre	18mm
Ángulo Giratorio	190°
Presión Máxima Absoluta	1MPa
Tipo de Amortiguamiento	Caucho
Conexión de Puerto	M5 x 0.8

4.10.4 ACTUADOR NEUMÁTICO DE PALETAS

Este cilindro se emplea junto al eyector de vacío para recoger la pieza del segundo procesamiento de la imagen y depositarla con ayuda del actuador de base rotativa en los depósitos B o C.



Figura 4.18 Actuador neumático de Paletas CRB2BW30-90SZ

Serie del Fabricante	CRB
Acción	Doble
Calibre	30mm
Ángulo Giratorio	90°
Máxima Presión de Funcionamiento	1 MPa
Tipo de Amortiguamiento	Caucho
Conexión de Puerto	M5 x 0.8

4.11 COMPONENTES PARA LA AUTOMATIZACIÓN Y EL TRATAMIENTO DE IMÀGENES

4.11.1 SISTEMA MYRIO

Para la automatización y parte de la visión artificial de la estación, se ha elegido una unidad central de procesamiento myRIO de National Instruments.

El NI myRIO es un dispositivo de hardware embebido enfocado mayoritariamente a la enseñanza, que permite diseñar sistemas de ingeniería complejos y reales de manera rápida y económica a través de sus componentes internos, acceso a software transparente y biblioteca de recursos y tutoriales.

NI myRIO es una herramienta ideal para ingeniería de control, automática, robótica, mecatrónica y sistemas embebidos.

NI myRIO está equipado con una FPGA Xilinx Zynq y procesador ARM Cortex-A9, con las cuales es posible la tarea de programar E/S personalizadas e implementar procesamiento en tiempo real en un dispositivo de manera rápida y eficaz.

Esta potente tecnología combinada con un acelerómetro interno, LEDs programables, E/S de audio, hasta 40 líneas de E/S digital, 10 canales de entrada analógica, 6 salidas analógicas, un puerto USB y WiFi interno para la conectividad de dispositivos USB ayudan a la creación de una gran variedad de proyectos



Figura 4.19 Controlador myRIO con carcasa y FPGA myRIO

Esta tecnología, al no tratarse de un tipo de lenguaje de programación tradicional, las cuales están basadas en un código con reglas sintácticas y semánticas que definen su estructura, ésta utiliza dos entornos relacionados entre sí para el desarrollo de sus programas:

- El panel frontal: este es la interfaz gráfica con la interacciona el usuario cuando el programa se ejecuta. De manera que se podría considerarse un tipo de programación orientada a objetos. En este panel el programador puede diseñar controles para manejar los actuadores del proyecto y colocar gráficas e indicadores que reciban los datos que registran los sensores.
- El diagrama de bloques: este entorno es el que lleva el peso de la programación. Aquí se desarrollaron los algoritmos y el código que hace funcionar el programa. Al igual que en otros lenguajes de programación, se pueden desarrollar, funciones (VI's), bucles y cualquier tipo de algoritmo para poder operar con los objetos del panel frontal.

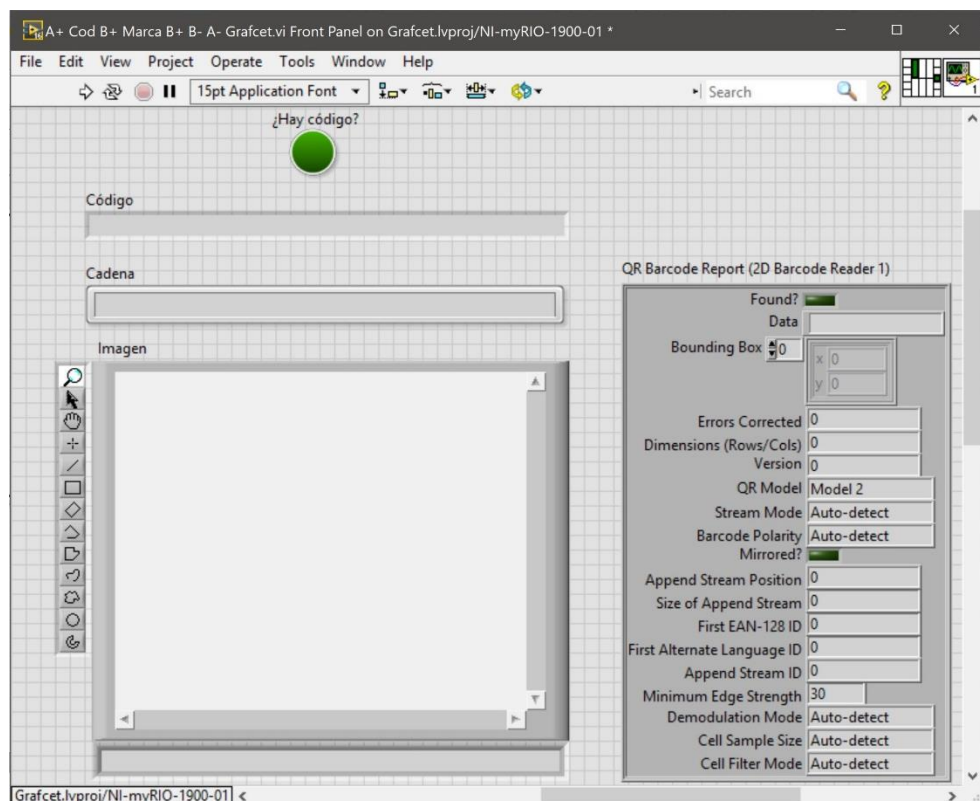


Figura 4.20 LabVIEW NI myRIO Panel frontal

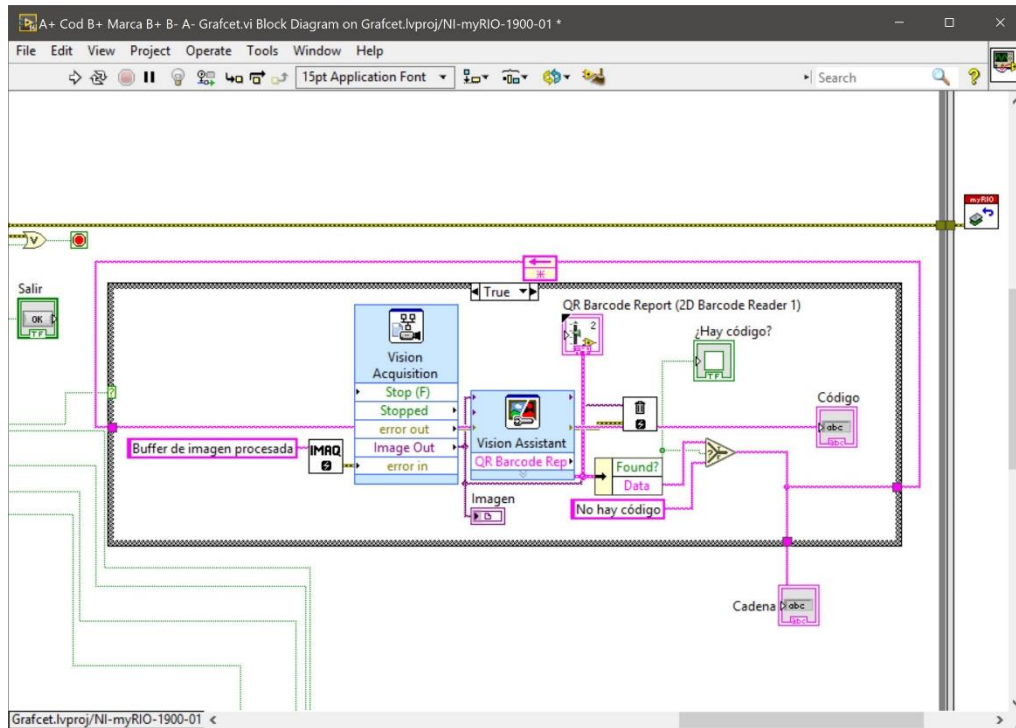


Figura 4.21 LabVIEW NI myRIO Diagrama de bloques

4.11.2 DISPOSITIVO RASPBERRY PI 3

Se ha empleado Raspberry Pi 3 en lugar de otras placas como por ejemplo Arduino, porque la Raspberry Pi es un microordenador y el Arduino es un microcontrolador, al tratarse de un procesamiento de imágenes era necesario emplear esta plataforma.

La Raspberry Pi tiene varias versiones, sin embargo esta última, dispone de mejoras significativas y es por eso que la Raspberry Pi 3 Model B es la versión que se ha empleado.

La Raspberry Pi 3, como se ha mencionado anteriormente, es un microordenador completo de bajo consumo en una única placa. Esta trae varias mejoras respecto a su anterior modelo, de las cuales destacaremos:

- Un procesador ARM Córtext A53
- Un procesador de cuatro núcleos a 1.2GHz de 64 bits
- Bluetooth 4.1 (de bajo consumo) y WiFi 802.11n integrado

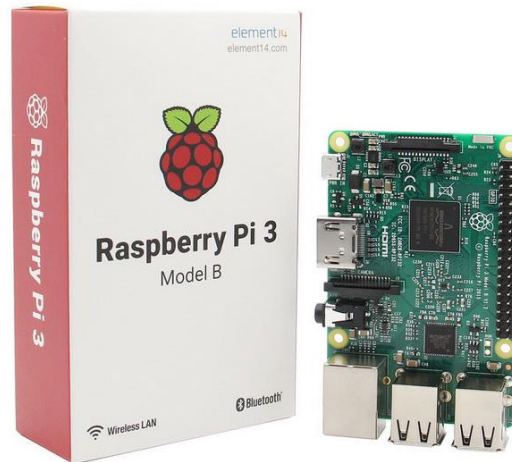


Figura 4.22 Raspberry Pi 3 Model B

Además podemos observar las especificaciones técnicas que trae consigo este dispositivo

Especificaciones técnicas de la Raspberry Pi 3 Model B

- Procesador:
 - Chipset Broadcom BCM2387.
 - 1,2 GHz de cuatro núcleos ARM Cortex-A53
- GPU
 - Dual Core VideoCore IV @ Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación.
- RAM: 1GB LPDDR2.
- Conectividad
 - Ethernet socket Ethernet 10/100 BaseT
 - 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)
 - Salida de vídeo
 - HDMI rev 1.3 y 1.4
 - RCA compuesto (PAL y NTSC)
 - Salida de audio
 - Jack de 3,5 mm de salida de audio, HDMI
 - USB 4 x Conector USB 2.0
 - Conector GPIO

- 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira
- Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro
 - Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
 - Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana Flex cable con dos carriles de datos y un carril de reloj
 - Ranura de tarjeta de memoria Empuje / tire Micro SDIO

4.11.3 CÁMARA LOGITECH C270

Mediante esta cámara, que es capaz de grabar imágenes de hasta 1280 x 720 a 30 FPS, realizaremos las capturas del primer procesamiento de la pieza realizado en NI myRIO y LabVIEW, mediante los cuales revisaremos si hay código QR o no lo hay.



Figura 4.23 Cámara Logitech C270

Marca	Logitech
Serie	C270
Peso del producto	150 g
Dimensiones del producto	21 x 7,6 x 15,2 cm
Color	Negro
Factor de forma	Webcam
Resolución de pantalla	3 MP
Máxima resolución de pantalla	720p
Fabricante del procesador	Intel
Número de procesadores	1
Voltaje	1.5 voltios
Plataforma de Hardware	PC / Mac

4.11.4 CÀMARA RASPICAM

La RaspiCam, no destaca por su robustez, su reducido tamaõ permite que sea un componente versàtil ocupando un espacio muy reducido.

Èsta es conectada con la Raspberry Pi a travÈs de un puerto de tipo CSI mediante una uni3n flexible que permite un mejor posicionamiento.

Tiene una resoluci3n de 5 megapìxeles y la capacidad de captura de vÌdeo a 1080p (Pìxeles) y 30 fps (Frames per second).



Figura 4.24 Raspberry Pi Camera "Raspicam"

4.11.5 ILUMINACIÓN

En este prototipo, no se ha especificado un sistema de iluminación específico, ya que será accionado desde un entorno controlado.

Sin embargo, la correcta iluminación es un pilar muy importante en un sistema de visión por computador para poder lograr la mayor optimización de la imagen, que a su vez mejora notablemente el procesado de la misma.

El objetivo principal de ésta, es la de aislar el elemento a inspeccionar de su entorno y así, resaltar los aspectos o rasgos de interés de una aplicación determinada.

Es por ello, que un estudio previo del sistema de iluminación en caso de no realizarse en un entorno controlado, es de vital importancia para un funcionamiento óptimo.

Existen múltiples métodos de iluminación que destacan diferentes características del objeto a inspeccionar. Estas se clasifican en:

- Iluminación direccional

A grandes rasgos, podemos decir que éste método dispone de una gran versatilidad, ya que la luz puede ser emitida desde uno o varios puntos y esto permite que se pueda incluir en diversas aplicaciones. Como principal defecto, provoca sombras y reflejos que en caso de no eliminarlos, pueden intervenir en el procesado del objeto.

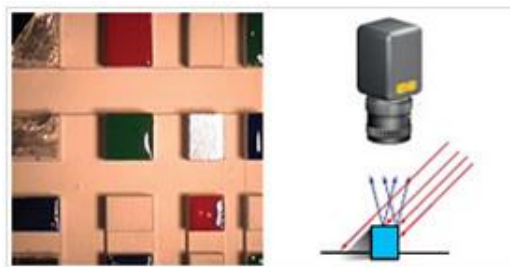


Figura 4.25 Iluminación direccional

- Iluminaci3n lateral

Al igual que la iluminaci3n direccional, este mètode es versàtil, ya que podemos variar diferentes àngulos de incidencia. Ademàs es un mètode mediante el cual es apto en aplicaciones cuya fuente de luz est3 en posici3n no central respecto a la càmara y en identificaci3n de objetos con profundidad

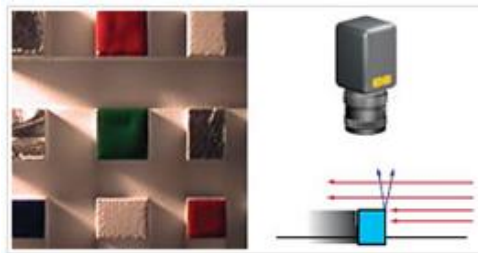


Figura 4.26 Iluminaci3n lateral

- Iluminaci3n difusa

Este tipo de mètode, sirve para solucionar la problemàtica de los mètodos mencionados anteriormente, ya que se emplea para eliminar las sombras y el deslumbramiento mediante un mltiple haz de luz uniforme desde diferentes àngulos hacia el objeto a inspeccionar. Pero, al disponer de mayores prestaciones, èsta afecta significativamente en el sistema, ya que se incrementa dificultando su instalaci3n en espacios reducidos.

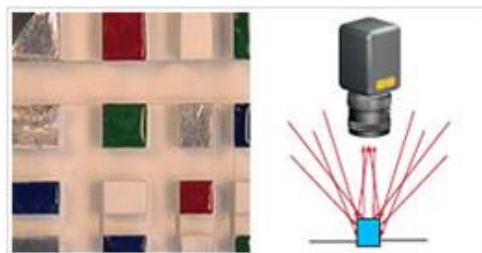


Figura 4.27 Iluminaci3n difusa

- Iluminación polarizada

Este tipo de método, consiste en un tipo de iluminación direccional teniendo las mismas ventajas de iluminación constante, pero con la peculiaridad de emplear luz polarizada para la reducción de los brillos y destellos.

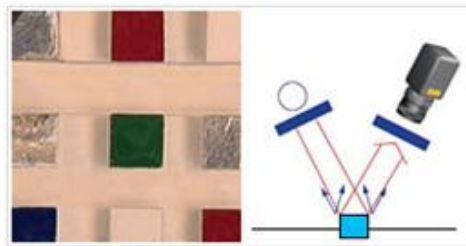


Figura 4.28 Iluminación polarizada

- Retroiluminación o backlight

Este tipo de método, consiste en disponer de una o varias fuentes lumínicas, pudiendo conseguir dos efectos diferentes dependiendo de la aplicación en objetos traslucidos u opacos. En el caso de un objeto no opaco la fuente lumínica se sitúa detrás de éste, permitiendo atravesarla y descubriendo imperfecciones. Para el caso de objetos opacos, este método se emplea para la detección de contornos.

Su principal desventaja es que no permite ver características superficiales y es sensible a los cambios lumínicos producidos en el ambiente.



Figura 4.29 Iluminación backlight

4.12 OTROS DISPOSITIVOS

4.12.1 FUENTE DE ALIMENTACIÓN 24 V

Los dispositivos de la estación tienen que ser alimentados a una determinada tensión, todos los dispositivos no funcionan a la misma tensión de alimentación, por ejemplo, la alimentación del myRIO lleva su propio transformador conectado a la red eléctrica, al igual que la Raspberry Pi. Sin embargo, el resto de componentes tienen una alimentación de tensiones de 3,3 V, proporcionada por el microcontrolador correspondiente.

La tensión de 24 V, que alimenta a las válvulas y a los sensores magnéticos, se necesita una fuente de alimentación.

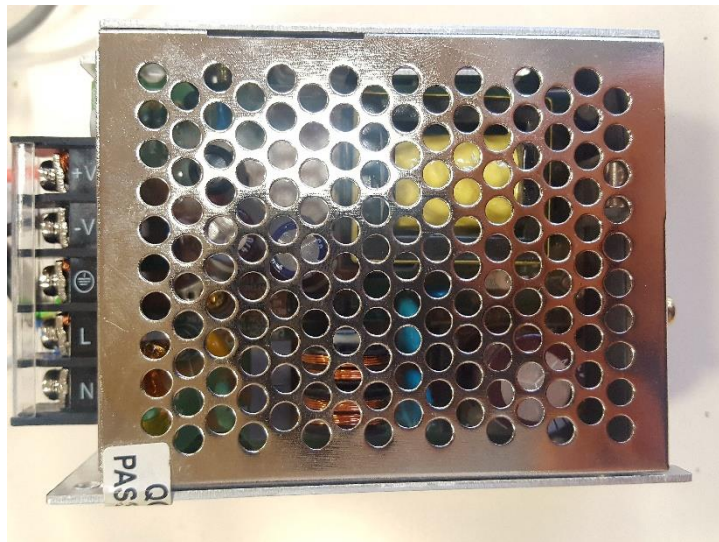


Figura 4.30 Fuente de alimentación 24VDC

4.12.2 TOMA DE AIRE A PRESIÒN

Un compresor proporciona aire a presiòn para el funcionamiento de la estaciòn. Este compresor genera presiòn y aire suficientes para el funcionamiento de èsta. La alimentaciòn de la estaciòn va conectada mediante la Unidad de Mantenimiento con un manòmetro instalado para poder ajustar la presiòn, la cual va instalada de forma directa al compresor



Figura 4.31 Compresor JUN-AIR de FESTO, 16bar

5. HERRAMIENTAS EMPLEADAS

5.1 PARA EL MECANIZADO

5.1.1 TALADRADORA

El taladro es una herramienta esencial para el mecanizado de la mayoría de piezas en talleres mecánicos. Una de sus ventajas es la sencillez y rapidez de su manejo.

Consta de dos movimientos. La primera es de rotación de la broca impresa por el motor eléctrico de la máquina mediante una transmisión de poleas y engranajes, y el de avance de penetración de la broca.

Para realizar la acción del taladrado a una pieza desde el punto de vista del mecanizado, hay que tener en cuenta los aspectos generales, como por ejemplo:

- Diámetro de la producción del agujero
- Calidad superficial y tolerancia
- Material de la pieza
- Material de la broca
- Longitud del agujero
- Sistema de fijación de la pieza en el taladro

5.1.2 SIERRA DE CINTA HORIZONTAL

La sierra de cinta, o tambi3n conocida como serrucho de banda, est3 constituida por una tira met3lica dentada, cuyas característic3s son de complej3n alargada, estrecha y flexible. El comportamiento de la tira es el de desplazamiento sobre dos puntos rodantes que se encuentran en el mismo plano vertical con un espacio entre ellas.

Para realizar el corte de materiales met3licos, se emplean sierras especiales que requieren de un refrigerante, la cual se va suministrando de manera constante sobre la cuchilla con el fin de evitar el sobrecalentamiento de la misma, que posiblemente causarí3 defectos en los cortes y un acortamiento prematuro de la vida ú3til de la cuchilla.



Figura 5.1 Sierra de Cinta horizontal

5.2 ELÉCTRICAS / ELECTRÓNICAS

5.2.1 ESTACIÓN SOLDADORA

Para la realización de las soldaduras de los componentes electrónicos en la placa de circuito impreso, se empleó una estación Soldadora "Weller" entre otros.

Las principales características de esta estación son:

- Display LCD de gran escala y fácil de leer
- Protección contra ESD
- ajustes variables de temperatura
- Bloqueo de temperatura, reducción y calibración, programables desde la unidad de alimentación
- Interfaz USB

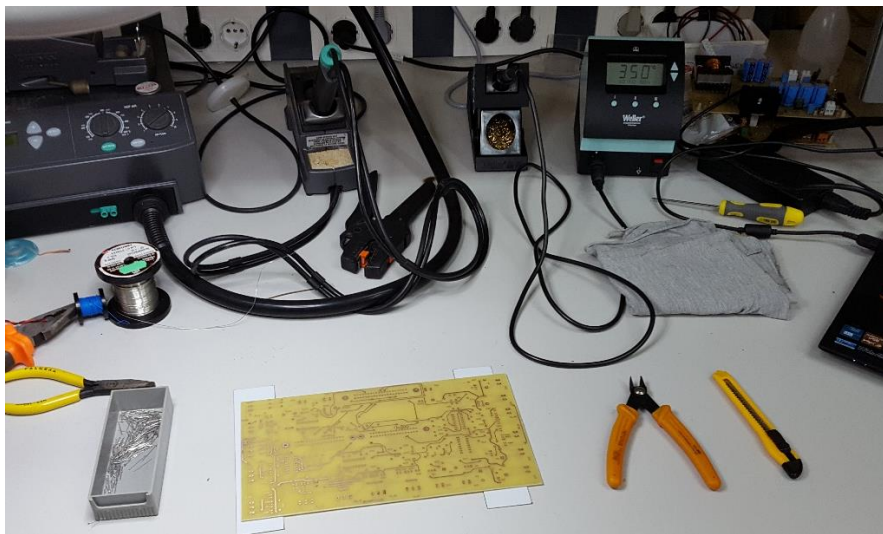


Figura 5.2 Estación soldadora y otros dispositivos empleados para soldar los componentes a la PCB

6. SOFTWARE

6.1 LABVIEW 2016



Figura 6.1 Ventana de inicio Software LabVIEW

LabVIEW (acr3nimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para dise1nar sistemas, con un lenguaje de programaci3n visual gr1fico. Recomendado para sistemas hardware y software de pruebas, control y dise1no, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gr1fico.

Este programa fue creado por National Instruments (1976) para funcionar sobre m1quinas MAC, sali3 al mercado por primera vez en 1986. Ahora est1 disponible para las plataformas Windows, UNIX, MAC y GNU/Linux. La 3ltima versi3n es la 2012, con la incre3ble demostraci3n de poderse usar simult1neamente para el dise1no del firmware de un instrumento RF de 3ltima generaci3n, a la programaci3n de alto nivel del mismo instrumento, todo ello con c3digo abierto.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen proven3a del control de instrumentos, aunque hoy en d3a se ha expandido ampliamente no s3lo al Control de todo tipo de electr3nica (Instrumentaci3n electr3nica) sino tambi3n a su programaci3n embebida, comunicaciones, matem1ticas, etc. Un lema tradicional de LabVIEW es: "La potencia est1 en el Software", que con la aparici3n de los sistemas multin3cleo se ha hecho a3n m1s potente. Entre

sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware- como de otros fabricantes.

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. Los programas en LabVIEW son llamados instrumentos virtuales (VIs) Para los amantes de lo complejo, con LabVIEW pueden crearse programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación. El LabVIEW 7.0 introduce un nuevo tipo de subVI llamado VIs Expreso (Express VIs). Estos son VIs interactivos que tienen una configuración de caja de diálogo que permite al usuario personalizar la funcionalidad del VI Expreso. El VIs estándar son VIs modulares y personalizables mediante cableado y funciones que son elementos fundamentales de operación de LabVIEW.

Presenta facilidades para el manejo de:

Interfaces de comunicaciones:

- Puerto serie
- Puerto paralelo
- GPIB
- PXI
- VXI
- TCP/IP, UDP, DataSocket
- Irda
- Bluetooth
- USB
- OPC...

Capacidad de interactuar con otros lenguajes y aplicaciones:

- DLL: librerías de funciones
- .NET
- ActiveX
- Multisim
- Matlab/Simulink
- AutoCAD, SolidWorks, etc.

Herramientas gráficas y textuales para el procesado digital de señales.

Visualización y manejo de gráficas con datos dinámicos.

Adquisición y tratamiento de imágenes.

Control de movimiento (combinado incluso con todo lo anterior).

Tiempo Real estrictamente hablando.

Programación de FPGAs para control o validación.

Sincronización entre dispositivos.

Como se ha dicho es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas:

- Panel Frontal: El Panel Frontal es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como van fluyendo los datos, un ejemplo sería una calculadora, donde tú le pones las entradas, y te pone el resultado en la salida). En esta interfaz se definen los controles (los usamos como entradas, pueden ser botones, marcadores etc...) e indicadores (los usamos como salidas, pueden ser gráficas...).
- Diagrama de Bloques: es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan (el código que controla el programa) --. Suele haber una tercera parte icono/conector que son los medios utilizados para conectar un VI con otros VIs.--

En el panel frontal, encontraremos todo tipos de controles o indicadores, donde cada uno de estos elementos tiene asignado en el diagrama de bloques una terminal, es decir el usuario podrá diseñar un proyecto en el panel frontal con controles e indicadores, donde estos elementos serán las entradas y salidas que interactuarán con la terminal del VI. Podemos observar en el diagrama de bloques, todos los valores de los controles e indicadores, como van fluyendo entre ellos cuando se está ejecutando un programa VI.

LabVIEW se puede usar para crear muchos tipos de programas sin estar enfocado a un ámbito en particular.

Las mismas funcionalidades de instrumentación, análisis y controles gráficos son también accesibles en Visual Basic, C++ o C# con Visual Studio gracias a Measurement Studio de National Instruments. Así, se puede disfrutar de la programación orientado a objetos y el framework .NET.

Las otras alternativas van desde lenguajes genéricos como C o Visual Basic pero sin la ayuda de la biblioteca de funcionalidades de National Instruments, a otras herramientas gráficas como HP-VEE, ahora Agilent-VEE.

Una de las alternativas en software libre es MyopenLab.

6.2 SOLIDWORKS 2016



Figura 6.2 Ventana de inicio Software SolidWorks

SolidWorks es un software CAD (diseño asistido por computadora) para modelado mecánico en 3D, desarrollado en la actualidad por SolidWorks Corp., una filial de Dassault Systèmes, S.A. (Suresnes, Francia), para el sistema operativo Microsoft Windows. Su primera versión fue lanzada al mercado en 1995 con el propósito de hacer la tecnología CAD más accesible.

SolidWorks Corporation fue fundada en diciembre de 1993 por Jon Hirschtick en el Instituto Tecnológico de Massachusetts con su sede en Concord, Massachusetts

SolidWorks lanzó su primer producto, SolidWorks 95, en noviembre de 1995

En 1997 Dassault Systèmes, mejor conocida por su software CAD CATIA, adquirió la compañía. Actualmente posee el 100% de sus acciones y es liderada por Jeff Ray.

Actualmente SolidWorks comercializa varias versiones del software CAD de SolidWorks, además de eDrawings, una herramienta de colaboración, y DraftSight, un producto CAD 2D.

6.3 KICAD



Figura 6.3. Logo del software Kicad

KiCad es una herramienta de software open-source para la creación de diagramas esquemáticos electrónicos y diseño de PCB.

KiCad incorpora un conjunto elegante de las siguientes herramientas de software independientes:

Nombre de la herramienta	Descripción	Extensión de Archivo
KiCad	Gestor de proyectos	*.pro
Eeschema	Editor de esquemas (esquemas y componentes)	*.sch, *.lib, *.net
CvPcb	Selector de huellas	*.net
Pcbnew	Editor de placa de circuito	*.kicad_pcb
GerbView	Visor de ficheros Gerber	Extensiones gerbers usuales
Bitmap2Component	Crea componentes o huellas a partir de imágenes bitmap	*.lib, *.kicad_mod, *.kicad_wks
PCB Calculator	Calculadora de componentes, tamaños de pistas, espaciado eléctrico, códigos de colores, etc.	Ninguno
PI Editor	Editor de formatos de página	*.kicad_wks

TABLA 6.1 Características de las herramientas del software Kicad

KiCad puede considerarse lo suficientemente maduro como para ser utilizado en el desarrollo exitoso y mantenimiento de tarjetas electrónicas complejas, es por ello que el diseño ha sido realizado con este software.

Y Al ser de código abierto (licencia GPL), KiCad representa la herramienta ideal para proyectos orientados a la creación de equipos electrónicos con estilo open-source.

7. PROGRAMACIÓN

7.1 PROGRAMA DE CONTROL MYRIO, LABVIEW

Para tener un completo control sobre la estación, se desarrolló un programa en LabVIEW. El programa consta de una pantalla principal en donde se da la posibilidad de la puesta en marcha o bien poder ejecutarla desde el pulsador de marcha físicamente instalado en la PCB o incluso se podrá poner en marcha desde un dispositivo móvil o Tablet.

Una primera funcionalidad del programa, es la de "MONITOR". Esta opción consiste en una pantalla de observación disponible en el Panel Frontal donde se muestran todos los dispositivos con los que cuenta la estación y están conectados a la PCB diseñada, la cual a su vez permite la comunicación entre la Raspberry Pi y el myRIO. Los dispositivos son las electroválvulas, los sensores, los Leds y los switches. El objetivo de esta pantalla es comprobar que todos estos dispositivos estén bien conectados en la estación y que el controlador pueda en todo momento leer la señal que recibe de ellos y activar o desactivar el dispositivo en cuestión si se trata de una salida.

Otra de las opciones que integra el programa, y a la que se puede acceder desde la pantalla principal del panel frontal, es la opción de control de la estación. Esta funcionalidad es la opción principal del programa y es donde reside todo el sistema de control para los actuadores. Esta sección del programa contiene al mismo tiempo en su interior varias pestañas donde se pueden ver los sensores magnéticos y la posición de los mismos. De esta manera se programa la secuencia para los canales de aire del circuito neumático que se va a emplear para el control pudiendo realizar un control de presión positiva.

A continuación veremos el diagrama de flujo (Grafcet) que se ha implantado para la automatización de la estación, la cual da una visión general sobre la lógica del algoritmo implantado en el sistema:

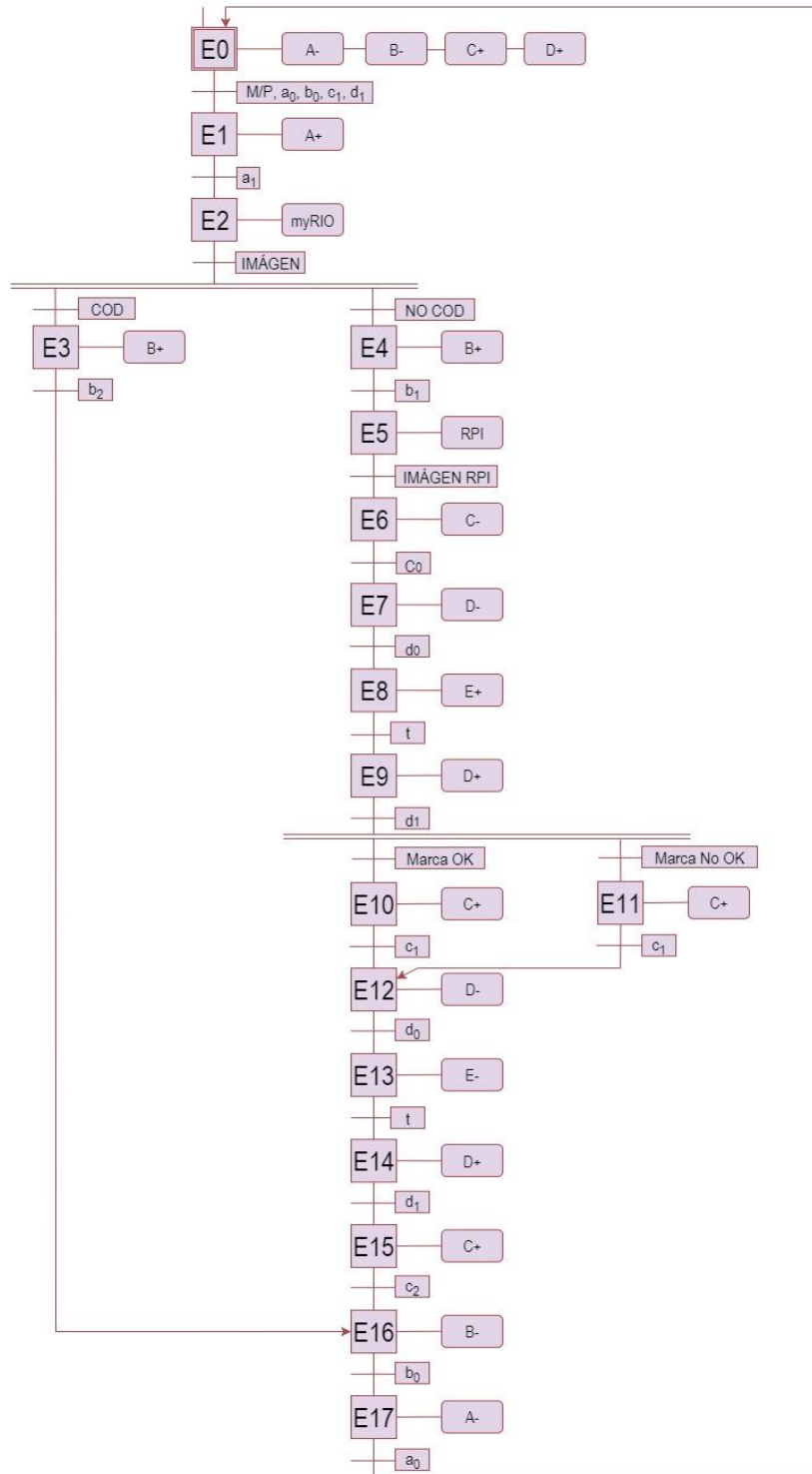


Figura 7.1 Diagrama funcional (Grafcet) para la automatización de la estación

Donde:

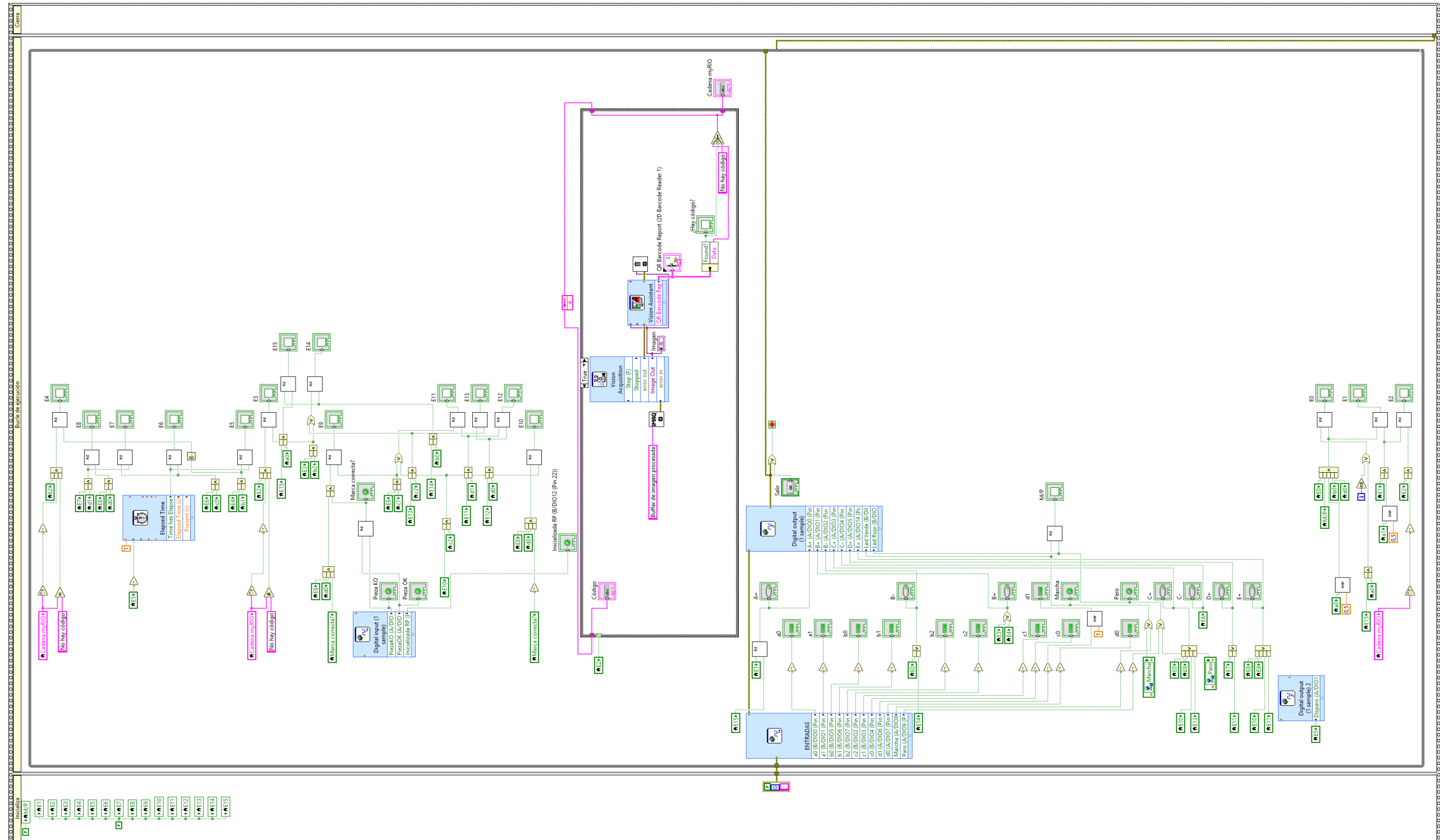
Electroválvula	Actuador	Tipo
A	Doble vástago	Monoestable
B	Lineal	Biestable
C	Base rotativa	Biestable
D	Paletas	Monoestable
E	Eyector de vacío	Monoestable

TABLA 7.1 Nomenclatura de los actuadores neumáticos

Sensor	Posicionamiento de los actuadores
a ₀	Doble vástago Posición Inicial
a ₁	Doble vástago Posición Expulsar pieza
b ₀	Lineal Posición Inicial
b ₁	Lineal Posición Pieza No Código
b ₂	Lineal Posición Pieza con Código
c ₀	Rotacional Posición Coger pieza No Código
c ₁	Rotacional Posición Pieza OK
c ₂	Rotacional Posición Pieza No Ok
d ₀	Paletas Posición a 0°
d ₁	Paletas Posición a 90°
M/P	Pulsadores de Marcha y Paro
t	Temporización

TABLA 7.2 Nomenclatura de los sensores y posicionamiento para los actuadores

7.1.1 DIAGRAMA FUNCIONAL EN LABVIEW



7.1.2 MÓDULO DE VISIÓN MYRIO, LABVIEW

En el presente apartado se detallan los módulos a implementar para el procesamiento de las piezas que disponen de código QR.

Al ser un sistema que adquiere imágenes y decodifica sin tener que emplear un guardado en memoria, lo primero que hay que hacer es la captura de la imagen y para ello emplearemos un módulo llamado:

- Ni Vision Acquisition Express

Mediante este módulo, procedemos a configurar parámetros, como el tipo de cámara que emplearemos que en nuestro caso es compatible con el módulo y algunas opciones de edición de brillo, ganancia, etc.

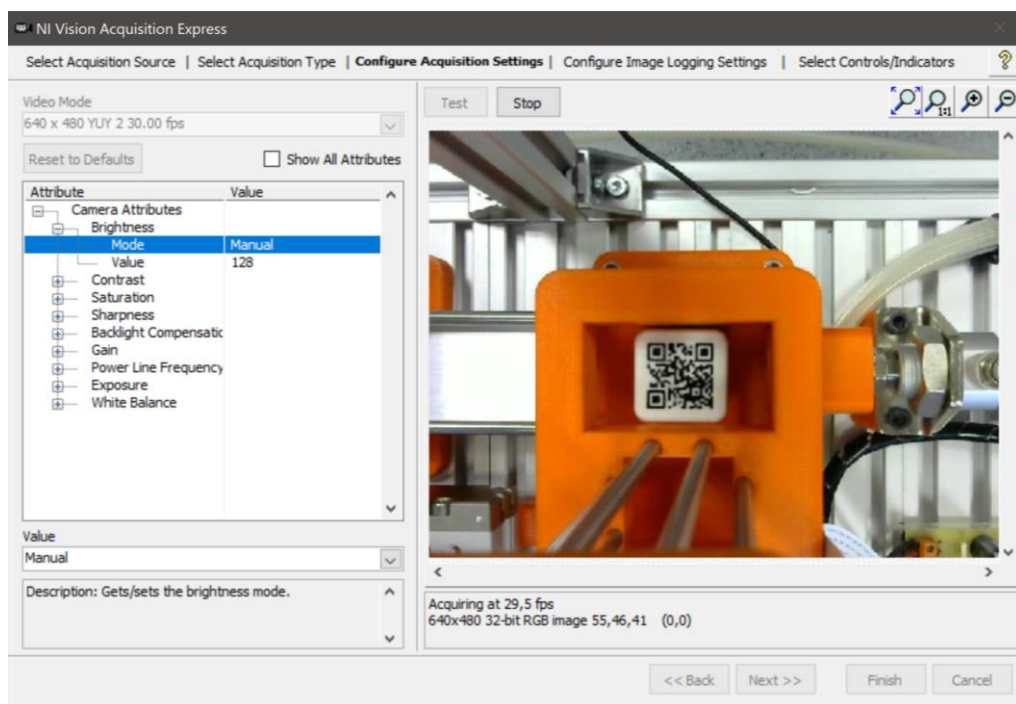


FIGURA 7.2 Configuración del asistente de adquisición de imagen

Este módulo es esencial para el buen funcionamiento de la cámara, ya que una mala configuración de ésta, supondría fallos en la captura de imágenes o de funcionamiento con anomalía. Sin embargo, es un módulo muy intuitivo y fácil de implementar.

Después de activar y configurar el módulo de captura de imagen, procedemos a configurar el módulo:

- NI Vision Assistant

Es en este módulo, donde aplicaremos los diferentes filtros para la optimización del procesamiento de la imagen.

En nuestro caso aplicaremos un modelo de color HSL (Hue Saturation Lightness) ya que define un modelo de color óptimo para la decodificación de nuestro Código QR teniendo en cuenta el color de la misma (Blanco) y el contraste entre ambos.

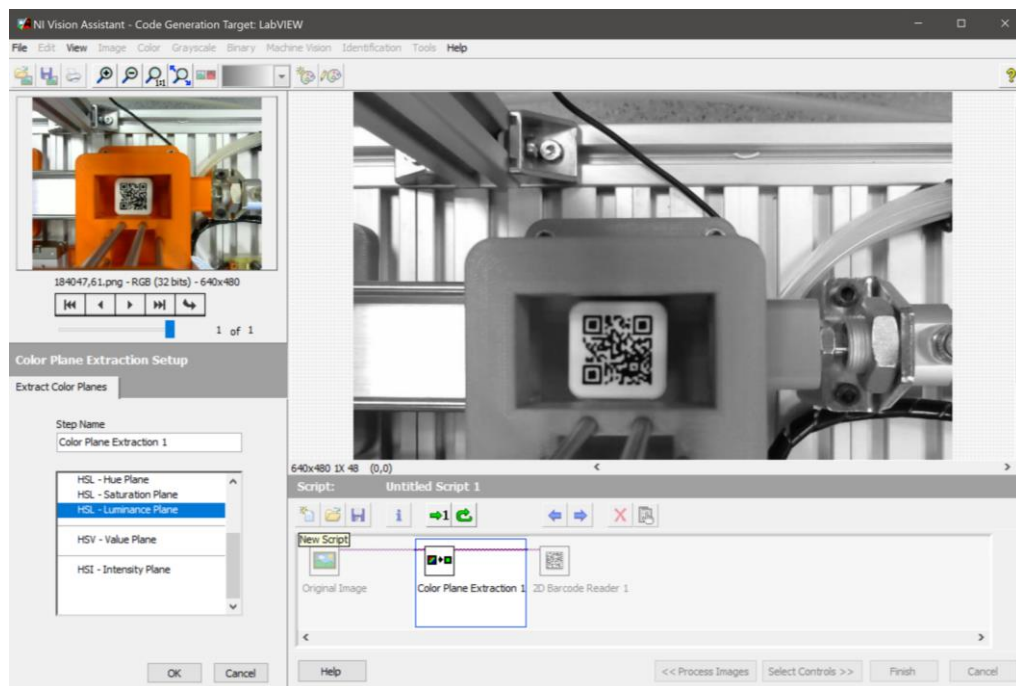


FIGURA 7.3 Configuración del modelo HSL mediante el módulo Vision Assistant

Por último, vamos a emplear el módulo para la decodificación de los códigos QR mediante un lector de dicha tecnología.

A continuación veremos las configuraciones para realizar la decodificación de manera eficaz.

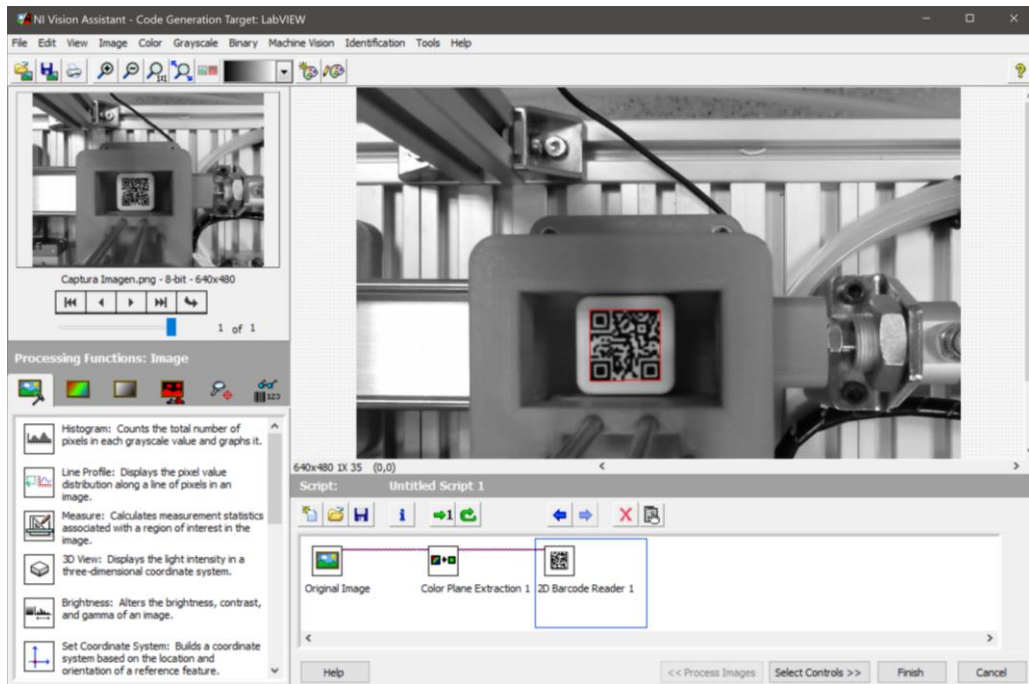


FIGURA 7.4 Configuración de parámetros del lector QR

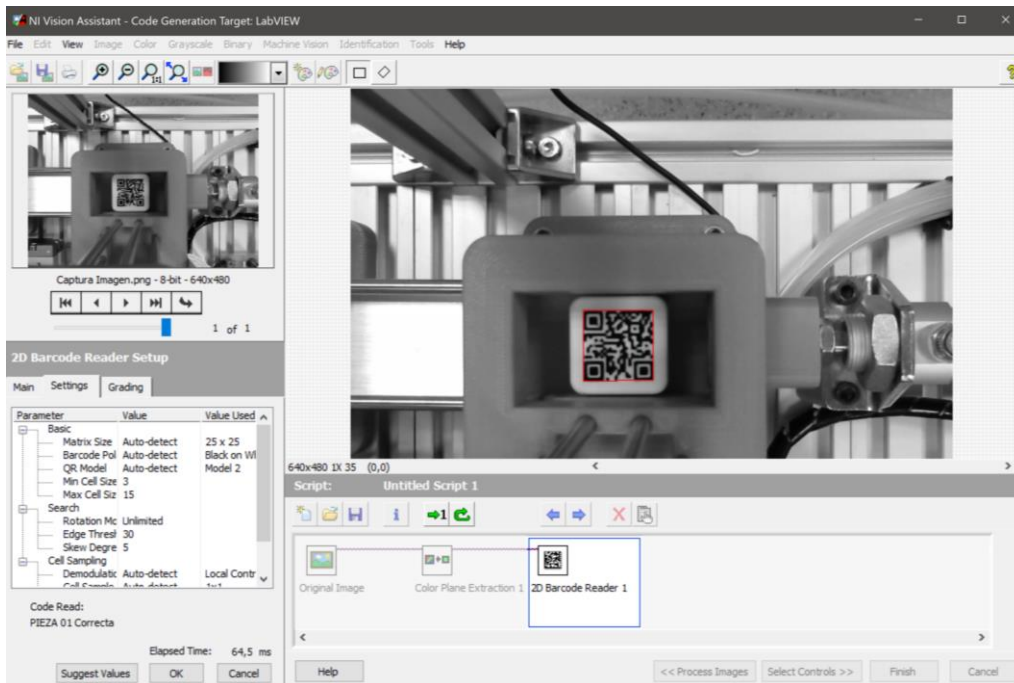


FIGURA 7.5 Parámetros de configuración del lector QR

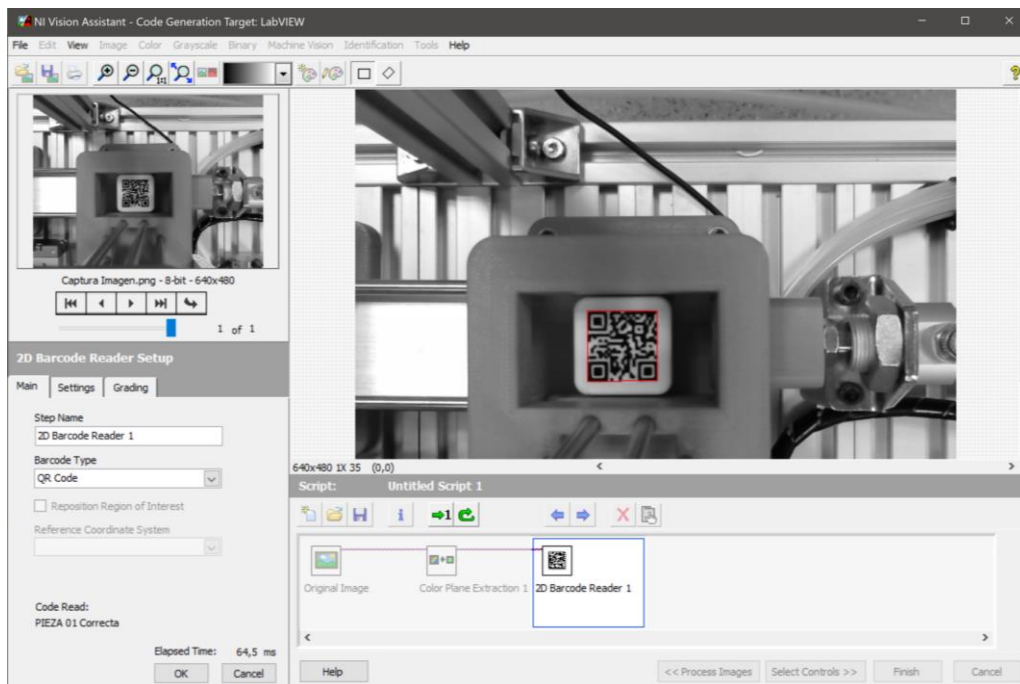


FIGURA 7.6 *Parámetros de configuración Principal del lector QR*

Cabe destacar que, es muy importante diferenciar entre un código matricial y un código QR, ya que la decodificación se realiza mediante diferentes tipos de lectores.

A continuación se ve la diferencia entre un código QR y otros de tipo matricial (Datamatrix y Código BIBI)

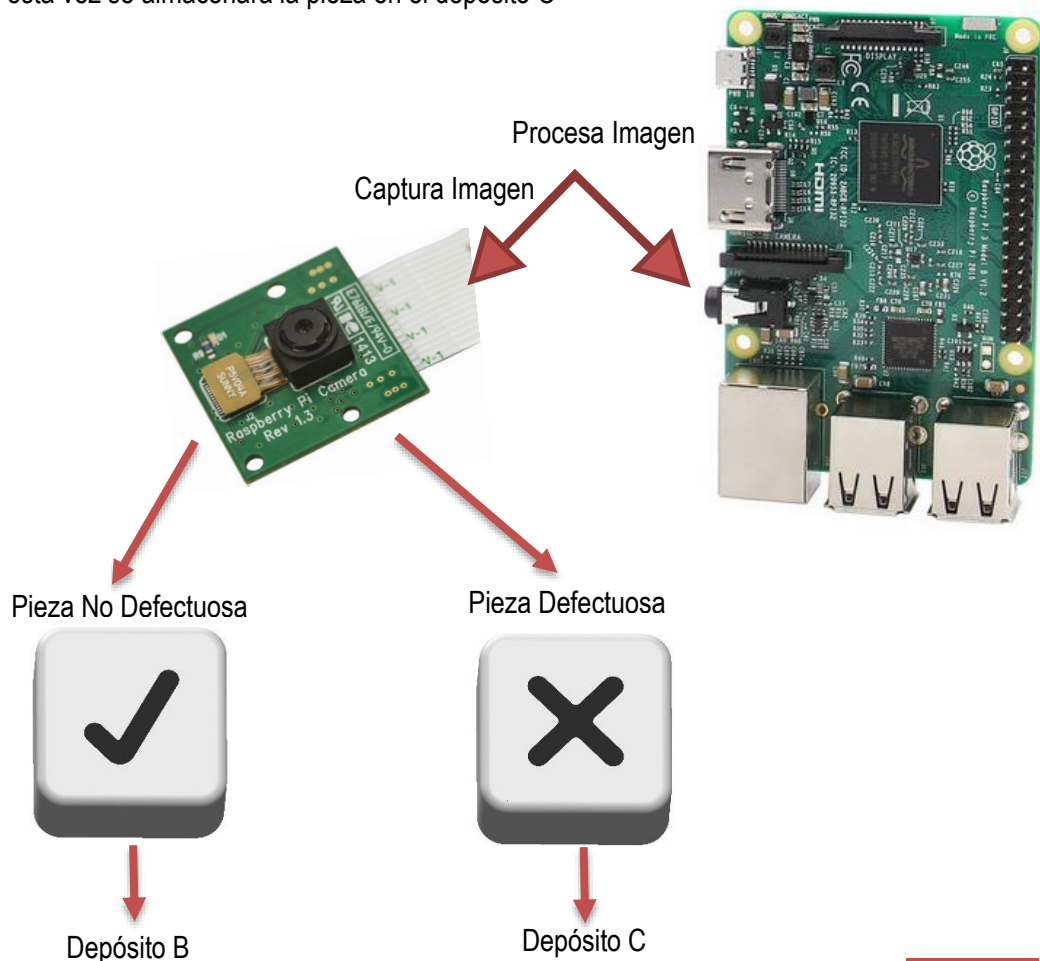


7.2 PROGRAMA VISIÓN RASPBERRY PI, OPENCV Y C++

Para la programación de la visión por computador de placa reducida Raspberry, se ha implementado una codificación en lenguaje C++ empleando librerías OpenCV.

Esta parte del procesamiento de la imagen es realizada para la inspección de piezas que no disponen de código QR Siguiendo los siguientes parámetros:

- Raspberry recibe señal de myRIO para poder capturar imagen
- Una vez recibe la señal del myRIO, la Raspberry inicializa la RaspiCam y Captura imagen
- Una vez capturada la imagen la Raspberry procesa la imagen e inspecciona la pieza
- Si la pieza es Correcta, envía una señal de control al myRIO para el almacenaje de la pieza en el Depósito B
- Si la pieza es Defectuosa, envía también una señal de control, pero con la diferencia de que esta vez se almacenará la pieza en el depósito C



7.2.1 CÓDIGO DE LA PROGRAMACIÓN EN C++

```
#include <ctime>
#include <iostream>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <raspicam/raspicam_cv.h>
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <opencv2/opencv.hpp>
#include <bcm2835.h>

using namespace std;
using namespace cv;

// Variables Capturar Imagen
time_t timer_begin, timer_end;
Raspicam_Cv Camera;
Mat imagen_RGB;
int nCount = 5;

int bandera = 0;
int noHay = 0;

/*****
/*      FUNCIONES DE ENCABEZADO      */
*****/

void CapturarImagen();

double getSimilarity( const Mat A, const Mat B );

int main()
{
    if (!bcm2835_init())
        return 1;

    // Definiendo los pines de entrada y salida
    bcm2835_gpio_fsel(RPI_GPIO_P1_10, BCM2835_GPIO_FSEL_OUTP); // Señal de salida de imagen No correcta
    //bcm2835_gpio_write(RPI_GPIO_P1_10, HIGH);

    bcm2835_gpio_fsel(RPI_GPIO_P1_08, BCM2835_GPIO_FSEL_OUTP); // Señal de salida de imagen Correcta
    //bcm2835_gpio_write(RPI_GPIO_P1_08, LOW);

    bcm2835_gpio_fsel(RPI_GPIO_P1_26, BCM2835_GPIO_FSEL_OUTP); // Señal para indicar que La Raspberry está lista
    //bcm2835_gpio_write(RPI_GPIO_P1_26, LOW);

    // sleep(20);

    // Variables de tipo doble para La comparación de Las piezas
    double parecido_equis;
    double parecido_bien;
    double parecido_bien2;
    double parecido_bien3;
    double parecido_bien4;
```



```
// Declaración de Las variables como Flags
int bandera_equis = 0;
int bandera_bien = 0;

.....

// Activa La salida BDIO12 del myRIO indicando que está preparada
bcm2835_gpio_write(RPI_V2_GPIO_P1_26, HIGH);

cout << "Raspberry iniciada" << endl;

cout << "\n*****" <<
      "\n*          INSPECCIÃN DE ANOMALÃ-AS PIEZAS          *" <<
      "\n*****\n" << endl;

// Configurar parametros de La camara
Camera.set(CV_CAP_PROP_FORMAT, CV_8UC3);
```

```
while(1){

// Importando comunicaci3n entre myRIO y Raspberry Pi
bcm2835_gpio_write(RPI_V2_GPIO_P1_26, HIGH);

cout<<"Esperando SeÃal myRIO ..."<<endl;
system("python envio.py");

cout<<"SeÃal Recibida!"<<endl;
bcm2835_gpio_write(RPI_V2_GPIO_P1_26, LOW);

cout<<"Abriendo camara..."<<endl;
if (!Camera.open())
{
cout<<"Error abriendo la camara"<<endl;
return -1;
}

// Llamar a La funcion Capturar Imagen
CapturarImagen();

// Parametros del tiempo
time ( &timer_end ); /* get current time; same as: timer = time(NULL) */
double secondsElapsed = difftime ( timer_end,timer_begin );

// Guardar imagen RGB
imwrite("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCapturadas/imagen_RGB.jpg",imagen_RGB);

cout << "\nParar camara..." << endl;
Camera.release();
```

```
// Creando Matriz para La Lectura de La imagen original que luego sera comparada
Mat src=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCapturadas/imagen_RGB.jpg",1);

// Creando matrices con todos Los posicionamientos posibles de La pieza No defectuosa
Mat bien=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCortadas/bien.jpg",1);
Mat bien2=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCortadas/bien2.jpg",1);
Mat bien3=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCortadas/bien3.jpg",1);
Mat bien4=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCortadas/bien4.jpg",1);

// Definiendo matriz para La Pieza Defectuosa
Mat equis=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCortadas/mal.jpg",1);

/* Creando Matriz foto y recortando La foto original para copiarla en ella, Los valores son:
(565,486,778-565,663-486)--> Pixel arriba izquierda (565,468) ; Pixel abajo derecha (778.663)
*/
Mat foto;
src(Rect(561,474,214,229)).copyTo(foto);

// Comparando similitudes entre La matriz foto y La imagen con La pieza No defectuosa en diferentes posicionamiento
parecido_bien = getSimilarity(foto,bien);
cout << "error parecido bien 1 " << parecido_bien << endl;
parecido_bien2 = getSimilarity(foto,bien2);
cout << "error parecido bien 2 " << parecido_bien2 << endl;
parecido_bien3 = getSimilarity(foto,bien3);
cout << "error parecido bien 3 " << parecido_bien3 << endl;
parecido_bien4 = getSimilarity(foto,bien4);
cout << "error parecido bien 4 " << parecido_bien4 << endl;

//Comparando similitudes entre La matriz foto y La imagen con con La pieza Defectuosa
parecido_equis = getSimilarity(foto,equis);
cout << "error parecido equis " << parecido_equis << endl;

//Guarda La imagen "foto"capturada
imwrite("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/FotoDeteccion/foto.jpg",foto);

// Errores entre La diferencia de matrices para distinguir entre una Pieza defectuosa de otr No defectuosa

if(parecido_bien<0.39){
    bandera_bien = 1;
    cout<<bandera_bien<< endl;
}
else if(parecido_bien2<0.39){
    bandera_bien = 1;
    cout<<bandera_bien<< endl;
}
else if(parecido_bien3<0.39){
    bandera_bien = 1;
    cout<<bandera_bien<< endl;
}
else if(parecido_bien4<0.39){
    bandera_bien = 1;
    cout<<bandera_bien<< endl;
}
else if(parecido_equis<0.39){
    bandera_equis = 1;
    cout<<bandera_equis<< endl;
}
else{
    noHay = 1;
}

if(bandera_bien ==1){
    bandera_bien = 0;
    cout << "Pieza Correcta!" << endl;
    bcm2835_gpio_write(RPI_V2_GPIO_P1_10, HIGH);
    bcm2835_gpio_write(RPI_V2_GPIO_P1_08, LOW);

    bcm2835_delay(3000);
}

if(bandera_equis == 1){
    bandera_equis = 0;
    cout << "Pieza Defectuosa!" << endl;
    bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_08, HIGH);
    bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_10, LOW);
}
```

```
}

if(bandera_equis == 1){
    bandera_equis = 0;
    cout << "Pieza Defectuosa!" << endl;
    bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_08, HIGH);
    bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_10, LOW);

    bcm2835_delay(3000);
}

if(noHay==1){
    noHay = 0;
    cout<<"No hay pieza" << endl;
    bcm2835_delay(3000);
}

bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_08, LOW);
bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_10, LOW);

}

bcm2835_gpio_write(RPI_BPLUS_GPIO_J8_26, LOW);
bcm2835_close();
return 0;
}

/*****
/*          FUNCION CAPTURAR IMAGEN          */
*****/

void CapturarImagen()
{
    //Capturar imagen
    // cout<<"Capturando "<<nCount<<" frames ..."<<endl;
    time ( &timer_begin );
    for ( int i=0; i<=nCount; i++ )
    {
        Camera.grab();
        Camera.retrieve (imagen_RGB);
        if ( i%5==0 ) cout<<"\r captured "<<i<<" images"<<std::flush;
        //La funcion de i%5==0 lo que hace es verificar si i es divisible por 5.
        //Si es divisible el resultado (residuo) de (i%5) sera 0.
        //Entonces la funcion compara este resultado y si es igual a cero es un numero divisible por 5.
    }
    cout<<endl;

    return;
}
}
```

```
/******  
/*      FUNCION DE SIMILITUD ENTRE MATRICES      */  
/******  
  
double getSimilarity( const Mat A, const Mat B )  
{  
    if ( A.rows > 0 && A.rows == B.rows && A.cols > 0 && A.cols == B.cols ) {  
        // Calculate the L2 relative error between images.  
        double errorL2 = norm( A, B, CV_L2 );  
        // Convert to a reasonable scale, since L2 error is summed across all pixels of the image.  
        double similarity = errorL2 / (double)( A.rows * A.cols );  
        return (similarity);  
    }  
    else {  
        //Images have a different size  
        return (10000000.0); // Return a bad value  
    }  
}
```

8. PRESUPUESTO

Artículo	Cantidad	Precio Unitario€	Total	Precio Grupo	
Neumática	Racor tubo recto de M8 a M4	2	1,78	3,56	12,44 €
	Silenciador Macho G1/4	4	2,22	8,88	
	* Racor hembra ø 8mm	2	2,13	4,26	
	* Tubo neumático ø 8mm	3	0,55	1,65	
	* Tubo neumático ø 4mm	6	0,34	2,04	
Dispositivos	* Sensor magnético PNP para Actuador Lineal	3	31,99	95,97	2.460,28 €
	* Sensor magnético estado sólido Actuador Paletas	2	35,72	71,44	
	Sensor PNP actuador de doble vástago	2	44,44	88,88	
	Rack de electroválvulas para 5 unidades	1	56,18	56,18	
	Cinta de montaje Interruptor 25 mm	3	2,81	8,43	
	Sensor PNP actuador de base rotativa	3	26,61	79,83	
	Válvula neumática de control manual	1	9,6	9,6	
	Carcasa auto-interruptor para el Actuador Paletas	1	83,2	83,2	
	Switch	2	0,75	1,5	
	Eyector	1	16,18	16,18	
	* myRIO	1	500	500	
	* Actuador neumático Lineal	1	63,06	63,06	
	* Actuador neumático doble vástago	1	242,38	242,38	
	* Actuador neumático Giratorio	1	469,27	469,27	
	* Actuador neumático Paletas	1	115,94	115,94	
	Ventosa neumática SMC ZPT16BS-B5, Fuelle Caucho de Silicona	1	14,98	14,98	
	Asa de agarre	2	4,71	9,42	
	Controlador de velocidad SMC AS2002F-04, 4mm x 4mm	8	15,72	125,76	
	Electroválvula neumática SMC, Serie SY5000, Rosca G 1/8, Función 5/3	2	88,23	176,46	
	* Raspberry Pi 3 y RaspiCam	1	55	55	
* Cámara Logitech C270	1	25	25		
Electroválvula SY3000 rosca G 1/8, función 5/2	3	50,6	151,8		
Electrónica e impresión 3D	Placa PCB	1	5	5	506,76 €
	Cable eléctrico	2	1,25	2,5	
	Cable conexión electroválvulas	7	9,8	68,6	
	Conectores hembra	20	0,75	15	
	Optoacopladores TLP521	5	2,43	12,15	
	Transistores	8	0,87	6,96	
	* Componentes de bajo coste (Resistencias, diodos, jumpers, etc.)	1	10	10	
	* Fuente de alimentación 24 v	1	50,55	50,55	
	Material de impresión 3D por horas	28	12	336	
Materiales	* Tornillería y tuercas de diferentes diámetros	1	20	20	710,74 €
	* Perfiles de aluminio 3 m 30 x 30	2	38,11	76,22	
	* Soporte ángulo	4	17,08	68,32	
	* Escuadras perfiles de aluminio	16	9,6	153,6	
	Varillas de acero de 6mm lote de 6	1	14,2	14,2	
	Pletina de aluminio lote de 3	1	25,18	25,18	
	* Perfiles de aluminio 1 m 20 x 20	1	25,56	25,56	
* Perfiles de aluminio 3 m 30 x 100	3	109,22	327,66		

* Estos artículos son reutilizados del departamento de mecánica y electrónica facilitados para este proyecto

TOTAL sin IVA 3.690,22 €

9. PRUEBAS EN LABORATORIO

9.1 MUESTRAS DEL PROCESAMIENTO DE IMAGEN

9.1.1 TRATAMIENTO DE LA IMAGEN CON MYRIO Y LABVIEW

Como hemos visto en el apartado "7.1.2 Diagrama funcional en LabVIEW" se ha programado un código en LabVIEW para el control de la estación y la inspección mediante el procesamiento de imágenes. Dichos procesamientos están abarcados por diferentes situaciones, las cuales actuarán en consecuencia en la parte de control

- **Inicialización de la cámara y a la espera de capturar imagen**

A continuación vemos el Panel Control, en el cual se pueden apreciar el funcionamiento de los sensores y los actuadores

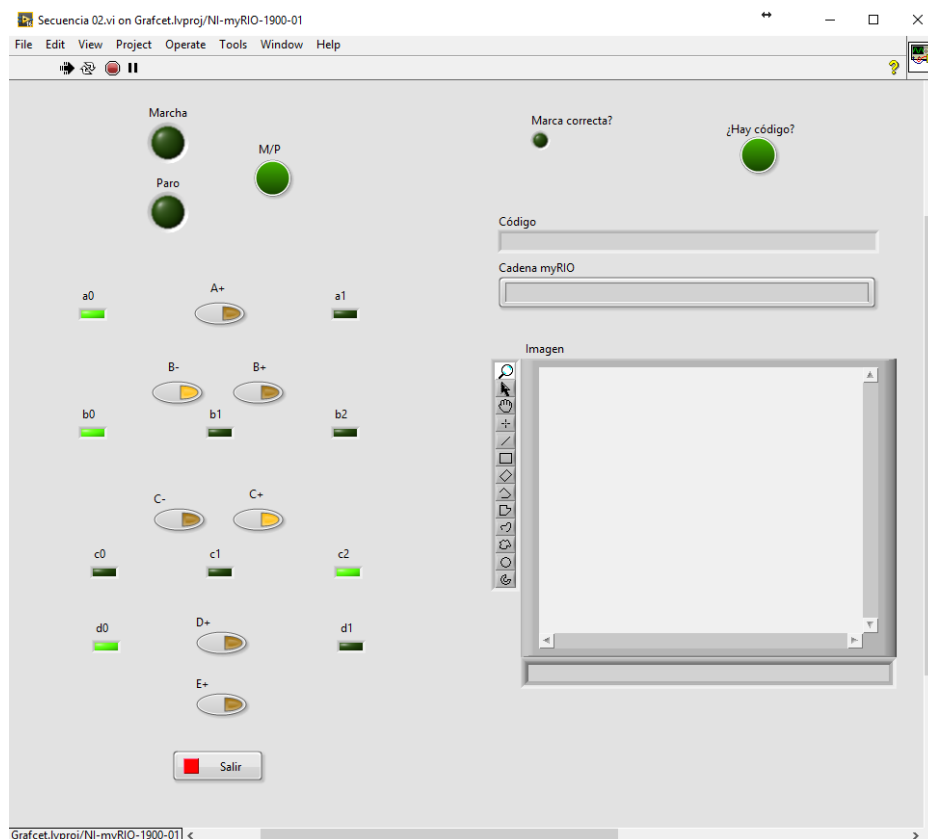


Figura 9.1 Panel frontal de inicialización del programa a la espera de captura de imagen

- **Imagen capturada, hay código QR Pieza 01 Correcta**

Esta situación es en la cual, se ha capturado una imagen y hay un código QR

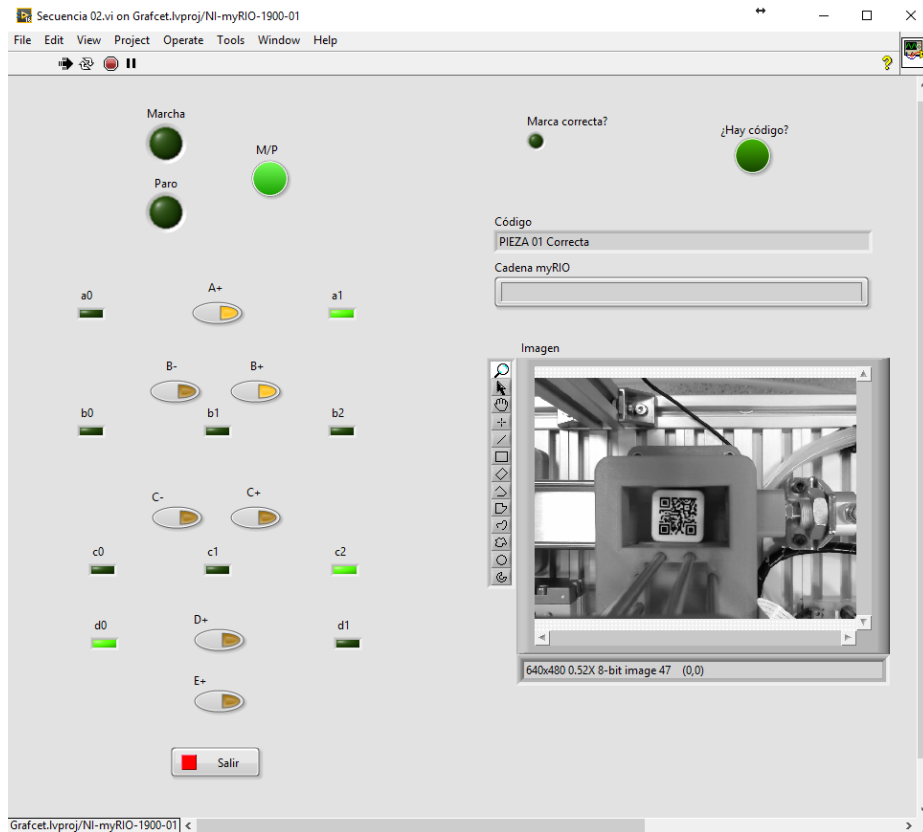


Figura 9.2 Decodificando código QR, Pieza 01 Correcta

Una vez se decodifica el código QR, ésta envía una señal al sistema de control para que la pieza sea almacenada en el Depósito A.

Cabe destacar, que hay dos tipos de piezas con códigos QR, la primera, es la que se muestra en la Figura anterior (Pieza 01 Correcta) y la segunda es la que se muestra a continuación.

- **Imagen capturada, hay código QR Pieza 02 Defectuosa**

Al igual que en el caso anterior, si hay código éste es decodificado y luego se envía una señal de control al sistema automatizado.

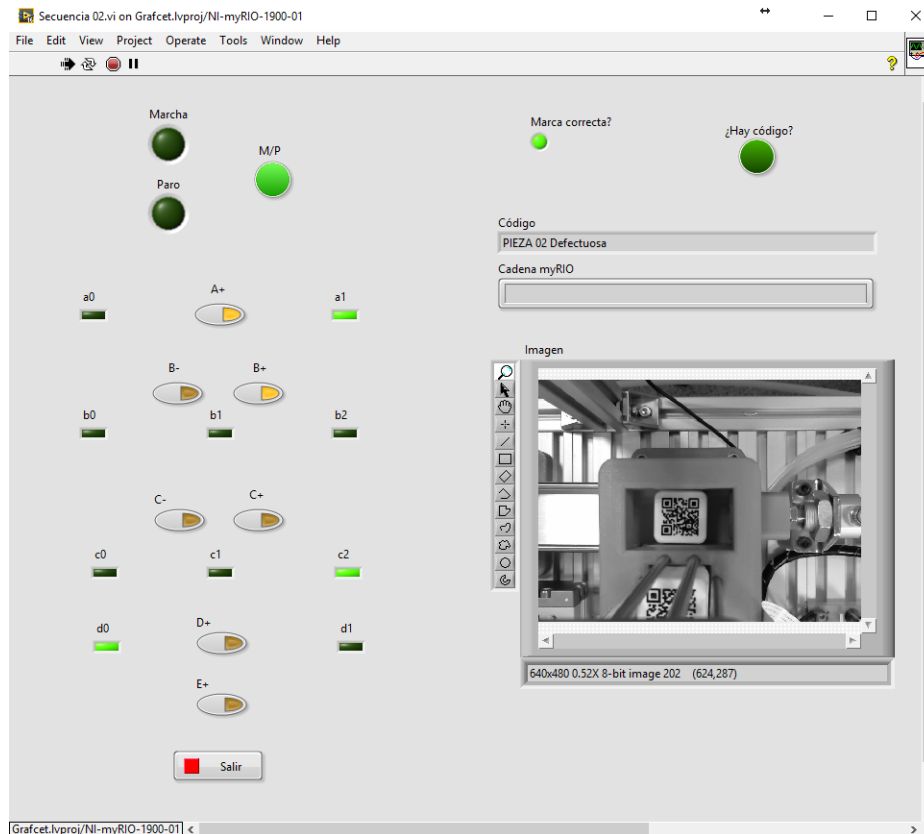


Figura 9.3 Decodificando código QR, Pieza 02 Defectuosa

Una vez decodificado, la pieza se almacenará en el depósito A que es donde se almacenan todas las piezas que disponen de Código QR.

- **Imagen capturada, No hay código QR**

En este caso, al capturar la imagen y no disponer de una pieza con código QR, ésta enviará una señal de control al sistema de automatización, la cual posicionará la pieza para una segunda inspección que se realiza mediante la Raspberry Pi y la Raspicam

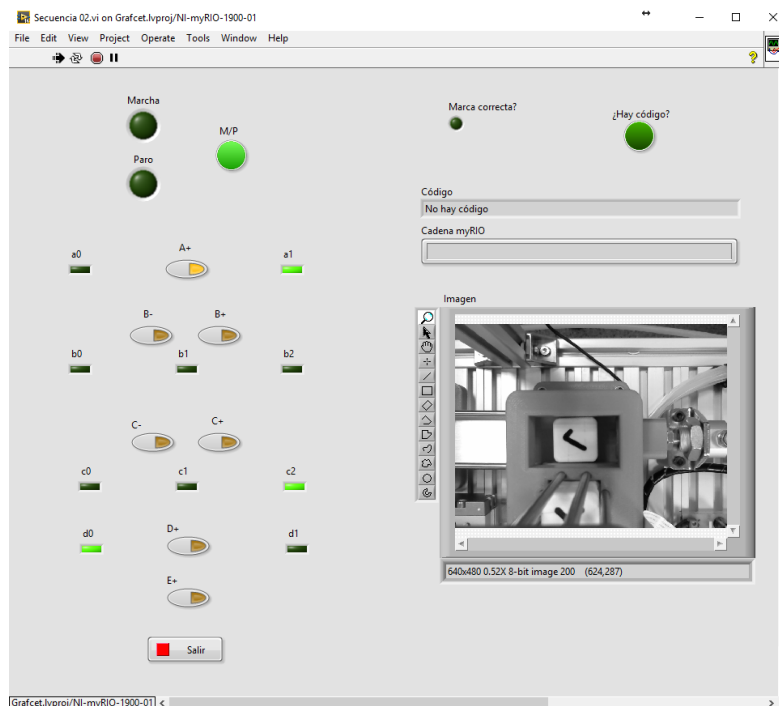


Figura 9.4 No encuentra código QR

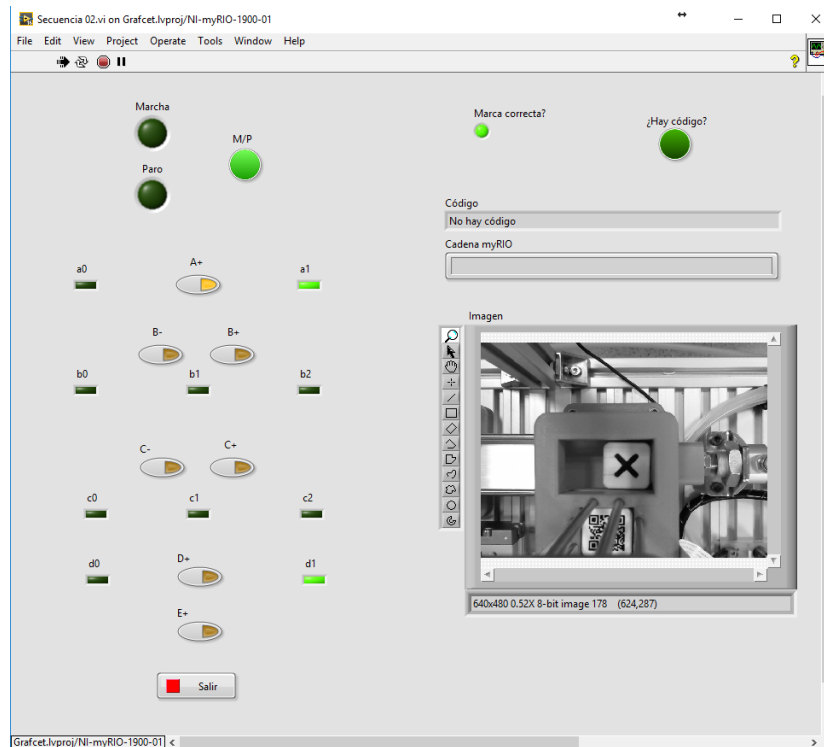


Figura 9.5 No encuentra código QR

Así pues, estas piezas serán procesadas por la Raspberry Pi y RaspiCam, las cuales procesarán la imagen para saber si se trata de una pieza Defectuosa, No defectuosa, o en su defecto si no hay pieza.

9.1.2 TRATAMIENTO DE LA IMAGEN CON RASPBERRY PI 3 Y C++

Para la discriminación de piezas, se realiza un muestreo de una imagen general, a la cual se le aplican filtros y funciones para el buen funcionamiento del programa, todo ello representado y comentado en el apartado de programación 7.2.1 Código de la programación en c++.



Figura 9.6 Imagen Original capturada por la RaspiCam

Cabe destacar, las funciones principales que da paso al algoritmo implementado para el procesamiento de la imagen, como por ejemplo, la captación de la imagen, que se hace mediante, primero el inicio de la cámara, captura de imagen y luego en la creación de una matriz:

```
// Creando Matriz para La lectura de La imagen original que luego sera comparada  
Mat src=imread("/home/pi/Desktop/TFGmisael/Misael/CapturaImagen/ImagenesCapturadas/imagen_RGB.jpg",1);
```

Para la realización de los recortes, se emplea el siguiente código creando nuevamente otra matriz, llamada foto:

```
Mat foto;  
src(Rect(561,474,214,229)).copyTo(foto);
```

A continuación se puede observar el resultado del recorte, tanto para piezas Defectuosas, como para No defectuosas.

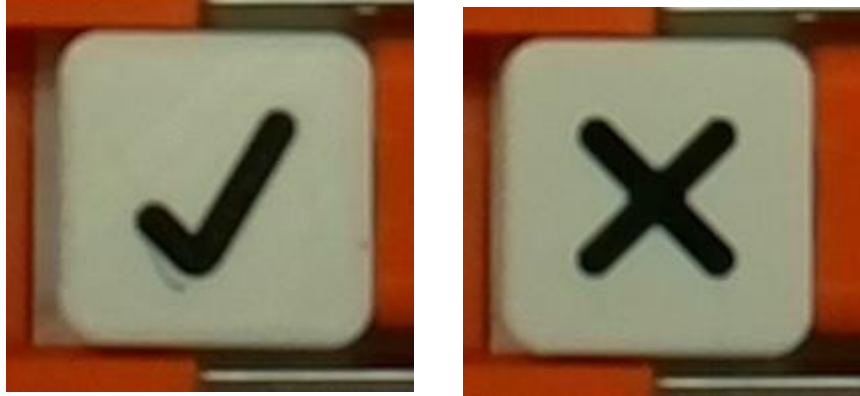


Figura 9.7 Muestra de imàgenes recortadas para su posterior procesamiento

La funci3n empleada para diferenciar unas piezas con otras, se realizan mediante la funci3n `getSimilarity()`; quien se encarga de obtener las diferencias entre dos matrices, la cual se declara como variable tipo "Double", ya que se dispone de valores con decimales.

```
double getSimilarity( const Mat A, const Mat B );
```

Para ver el procesamiento completo de la imagen, ver el c3digo completo en el apartado "7.2.1 C3digo de la programaci3n en c++ de la visi3n "

Es importante destacar que la seàal entre el LabVIEW y la Raspberry se realiza mediante un sistema de comunicaci3n en Python, el cual consiente en:

```
// Realiza la asignaci3n de los pines de la Raspberry
import RPi.GPIO as GPIO

//Nomina un Pull down al pin 29 de la Raspbeery
GPIO.setmode(GPIO.BOARD)
GPIO.setup(29,GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

// Declara una variable para indicar que el pin 29 de la Raspberry es una entrada
x = GPIO.input(29)

#print(x)

while x!=1:
    x= GPIO.input(29)
    #print(x)
```

10. CONCLUSIÓN

Mediante toda la documentación recogida en el presente proyecto, se ha demostrado que el prototipo cumple con todas las demandas exigidas para su buen funcionamiento. Así como también su flexibilidad a la hora de implementarlo, ya sea en un entorno industrial o académico.

El prototipo ha demostrado su efectividad en el sistema de visión artificial y en el control automático.

Un punto de mejora y ampliación de la estación, sería un estudio mucho más exhaustivo en el ámbito de la iluminación, ya que es un pilar fundamental para cualquier sistema de visión por computador, así como una actualización de las mejoras en dicha tecnología ya que siempre está sometida a constantes mejoras en optimización de funciones y parámetros que la convierten cada vez más en una herramienta imprescindible para la mejora de calidad en inspecciones industriales y en cualquier otro ámbito relacionado con el progreso de la tecnología en general.

En lo personal, el diseño de la estación como idea de una plataforma flexible, fue una elección acertada, ya que pude adquirir y mejorar conocimientos en diversos campos, como por ejemplo, en el modelado de piezas 3D, la automatización industrial, el diseño de Placas de Circuito Impreso y también en la programación orientada a la visión por computador.

11. BIBLIOGRAFÍA

- Learning Opencv3, Gary Bradski and Adrian Kaehler
- NI Vision Manual software Vision Assitant.
- Fundamentals of image precessing system, Niederjohann, D.B. 2014
- MIDOTP midwest optical systems 2012, Machine vision filters, USA.
- Asesor en visión artificial, Infaimon 2015, Catálogo.
- National Instruments - LabVIEW
- National Instruments - myRIO
- LabView: entorno gráfico de programación, J Pelegrí Sebastiá y José Rafael Lajara Vizcaíno
- Programación C++ , Escuela técnica superior de Ingeniería y Diseño Industrial
- Librerías OpenCV, opencv.org
- KiCad EDA, kicad-pcb.org
- SolidWorks Práctico I, Sergio Gómez González

12. ANEXOS

(Ver anexos adjuntos)