



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

SARA RICO RIVERO

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

**AUTOMATIZACIÓN DEL PROCESO DE MONTAJE
Y TEST DE UNA LÁMPARA LED PARA
ILUMINACIÓN MEDIANTE ROBOT INDUSTRIAL Y
VISIÓN ARTIFICIAL**

Febrero del 2021

RESUMEN

El proyecto fin de máster surge de las prácticas de empresa que se están realizando en la empresa Normagrup Technology S.A. El objetivo es automatizar el montaje de una lámpara de iluminación LED mediante un robot industrial, concretamente el proceso de colocación de las placas de LED en la base y soldadura de las mismas. Posteriormente, se realiza un test de funcionamiento, alimentando el conjunto de la lámpara y verificando mediante visión artificial la correcta operación de cada diodo LED de manera individual. Se construye un demostrador basado en un robot de ABB que se programa en RAPID. La cámara de visión artificial, utilizada para detectar la correcta operación de los LEDs, se programa en C++. La pantalla de explotación del sistema y los algoritmos de procesado de visión artificial se programan en Python. El prototipo desarrollado se integra en una de las líneas de fabricación automatizada de la empresa.

Índice

1-	Hipótesis de partida y alcance	5
1.1-	La empresa: NormaGrup Technology S.A.	5
1.2-	La importancia de la automatización de los sistemas de iluminación ...	6
2-	Objetivos	7
2.1-	Objetivos concretos.....	7
2.2-	Herramientas hardware disponibles.....	9
2.2.1-	Robot IRB2600 y controlador IRC5.....	10
2.2.2-	Unidad de soldadura USP5-093A	11
2.2.3-	Limpiador automático FA-1000BC	13
2.2.4-	Sistema de agarre y fijación.....	14
2.2.5-	Cámara Nano-C1920-1	15
3-	Descripción del proceso.....	15
3.1-	Entorno de trabajo.....	15
3.1.1-	Campo de trabajo 1 o automático	18
3.1.2-	Campo de trabajo 2 o manual.....	18
3.2-	Paso 1: Inicio del proceso y caracterización de la lámpara	19
3.3-	Paso 2: Automatización del proceso de colocación de las tiras de LED sobre la base de montaje.....	21
3.4-	Paso 3: Automatización del proceso de soldadura.....	22
3.5-	Paso 4: Inicio del proceso de test: Alimentación de la lámpara.....	24
3.6-	Paso 5: Test de la lámpara mediante visión artificial.....	25
3.7-	Paso 6: Validación del producto	26
4-	Trabajo realizado y resultados obtenidos.....	27
4.1-	Descripción de las comunicaciones digitales entre todos los elementos del sistema	27
4.1.2-	Sockets TCP/IP.....	27
4.1.3-	Ethernet	28
4.2-	Descripción del programa RAPID para manejo del robot	29
4.2.1-	Conexiones del robot	32
4.2.2-	Detalle de implementación del paso 1 en RAPID	33
4.2.3-	Detalle de implementación del paso 2 en RAPID	33
4.2.4-	Detalle de implementación del paso 3 en RAPID	34
4.2.5-	Detalle de implementación del paso 4 en RAPID	38
4.1.6-	Detalle de implementación del paso 5 en RAPID	38

4.2.7- Detalle de implementación del paso 6 en RAPID	39
4.3- Descripción de la configuración de la cámara y el programa en C++ para la captura de imágenes	39
4.3.1- Archivo de configuración de la cámara	39
4.3.2- Descripción del programa adquisición de imágenes	39
4.4- Descripción del programa de Python.....	40
4.4.1- Archivo de configuración de Python.....	40
4.4.2- Descripción del algoritmo de visión artificial e implementación en Python	41
4.4.3- Cliente TCP socket	43
4.4.4- Servidor TCP socket	44
4.5- Pantallas de explotación e interface con el usuario.....	44
4.5.1- Diseño de la pantalla de explotación.....	44
4.5.2- Uso de la pantalla de explotación	46
4.6- Implementación de estrategias de seguridad	47
4- Conclusiones y futuros trabajos	49
5- Bibliografía	50

1- Hipótesis de partida y alcance

1.1- La empresa: NormaGrup Technology S.A.

NormaGrup Technology S.A es una empresa tecnológica asturiana dedicada al sector de la iluminación. Está situada en el parque tecnológico de Llanera en Asturias, contando con cuatro naves dentro de dicho parque, en las que se desarrollan las distintas tareas.

La empresa se formó en el año 1971. En sus comienzos se dedicaba a la fabricación prácticamente artesanal de una lámpara de emergencia. Sin embargo, en la actualidad se ha convertido en una gran empresa, con un gran número de productos y líneas de fabricación. Además, actualmente la empresa tiene presencia a nivel internacional, distribuyendo su tecnología por distintos países.

Una característica destacable de NormaGrup es que está muy dedicada a la innovación, contando incluso sus propios laboratorios de investigación y experimentación. Gracias a esto en 2018, Idepa le concedió a la empresa un premio por su apuesta en crecimiento, su gran inversión en innovación y desarrollo y por su gran apertura hacia nuevos mercados.

Otra de las características destacable de la empresa es su alto nivel de automatización dentro de la fabricación, almacenamiento y montaje de sus distintos productos, incorporando constantemente tecnologías de vanguardia que les permite tanto una alta velocidad de producción como una gran fiabilidad. Con todo esto la empresa de ha vuelto líder de venta de iluminación de emergencia y está quinto en el sector a un nivel europeo.[1]

Debido a su gran tamaño, en la actualidad la empresa cuenta con cuatro divisiones dentro de la misma. Cada una de ellas tiene un área de especialización. Las divisiones que forman en su conjunto la empresa NormaGrup son las siguientes:

- NormaLux: especializada en iluminación de emergencia.
- NormaLit: especializada en sistema de luz tanto a nivel técnico y arquitectónico.
- NormaDet: especializada en detectores de humo autónomos.
- NormaClinic: especializada en accesorios destinados a hospitales, residencias o lugares para cuidados intensivos.

Por otro lado, dentro de la empresa NormaGrup Technology S.A existen distintos departamentos encargados de las distintas tareas dentro de la empresa. Los 4 departamentos principales son el departamento de la planta productiva, encargado de las taras de producción, el departamento de I+D+I, para las tareas de desarrollo de nuevos productos y sistemas, el departamento de diseño y el departamento de mecanizado. Las tareas realizadas para la asignatura de las practicas, fueron realizadas para el departamento de producción. Dicho departamento se encuentra situado en la nave 1 del parque tecnológico de Llanera. El trabajo realizado fue implementado dentro de una de las líneas de fabricación de la empresa en dicha nave. En la Figura 1.1 se pueden ver la apariencia de las naves que forman la empresa NormaGrup Technology S.A.



Figura 1.1.- Naves que forman la empresa NormaGrup Technology S.A. Cortesía de [1]

1.2- La importancia de la automatización de los sistemas de iluminación

La automatización industrial se ha implementado en los últimos años de manera progresiva. En las industrias de todo el mundo se han sustituido tareas realizadas por personas, para que fueran realizadas por máquinas. Un sistema de producción automatizada, es un sistema en el que no es necesario la intervención de personas para la producción. Para este tipo de sistemas se utilizan componentes como sensores, actuadores, tecnologías software, robots industriales, procesadores y otras tecnologías de más reciente utilización como son la visión artificial, inteligencia artificial o aprendizaje automático. En conjunto, la automatización industrial puede definirse como el uso de máquinas electromecánicas de robótica industrial o por sistemas por computadora que realizan de manera automática y autónoma los procesos que se ejecutan en una empresa. Las principales razones que han llevado a la industria a automatizarse son que mediante estas medidas se consigue una gran efectividad, velocidad de producción, mejora el control logístico y la organización e incluso mejora la calidad en los productos fabricados. Además, se evita que las personas realicen trabajos que pongan en peligro su seguridad, mejorándose la calidad del trabajo de las personas.[2][3]

En los sistemas de iluminación creados por la empresa NormaGrup Technology S.A, la automatización de la fabricación de dichos sistemas permite la obtención de grandes velocidades de producción, así como una alta calidad en sus productos. Mezclando el uso de robots industriales y otros dispositivos y técnicas de automatización, con el trabajo elaborado por personal cualificado, se

consigue una fabricación bien organizada, con flexibilidad en los productos y sin desperdicios de materiales.

2- Objetivos

2.1- Objetivos concretos

El objetivo del trabajo es automatizar parte del proceso de fabricación de lámparas LED para la empresa NormaGrup Technology S.A. Más concretamente, se quiere automatizar el montaje de tiras de LED sobre la base de la lámpara, la soldadura de las tiras de LED, y la verificación del correcto montaje de la lámpara. Para ello se cuenta con un robot industrial equipado con las herramientas necesarias, y se utilizarán conjuntamente, técnicas de visión artificial.

Existe variedad en cuanto a las lámparas de LED que puede fabricar el sistema, aunque sus características en común es que estarán formadas por placas de LED alargadas, las cuales se sitúan en una base para la lámpara, una detrás de la otra. En la Figura 2.1 se pueden ver 9 tiras de LED de la empresa NormaGrup Technology S.A., las cuales son un ejemplo de tiras de LED que podrían utilizarse para el proceso de fabricación. El sistema cuenta entonces con versatilidad para que sea posible su utilización para distintos modelos de estas lámparas. Por ejemplo, con tiras de LED con distintas longitudes, con distinto circuito interno, y requiriendo distintas soldaduras, y por supuesto para para lámparas con mayor o menor número de tiras de LED.



Figura 2.1.- Ejemplo de tiras de LED de la empresa NormaGrup Technology S.A.

En la se Figura 2.2 se muestra como son las bases de las lámparas y como se sitúan las tiras de LED sobre las mismas. Las tiras de LED quedan fijas sobre la base al introducir los enganches en los agujeros de la base. Además, las tiras de LED se sueldan unas a otras por sus extremos. La longitud de las bases de la lámpara depende del número de tiras de LED que lleven.

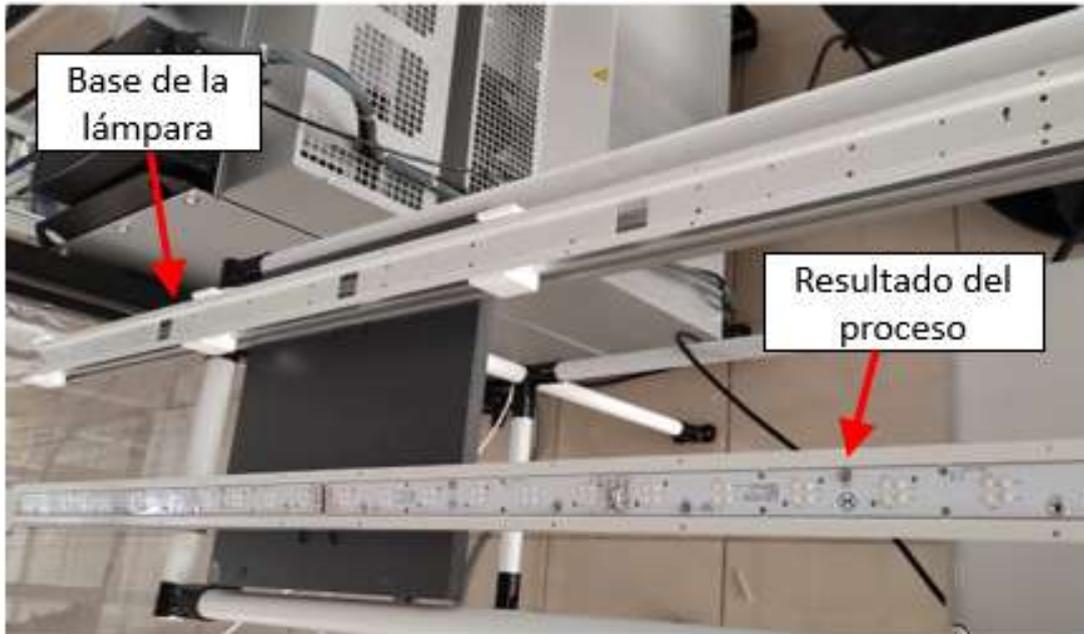


Figura 2.2.- Imagen de la base de la lámpara y el resultado tras el proceso.

Como se ha dicho, cada tira de LED puede tener un circuito interno diferente, y requerir distintas soldaduras. Como ejemplo podemos observar la Figura 2.3 donde vemos un ejemplo circuito de una lámpara formada por tres tiras de LED. El circuito interno de los LED que componen las placas son en realidad dos circuitos, que se deben unir y alimentar como se muestra en la figura. Para el montaje y verificación de dicha lámpara habrá que realizar las soldaduras y alimentación mostrada, pero para otra lámpara requerirá otras circunstancias distintas. De esta manera, el sistema debe ser capaz de amoldarse a las necesidades de fabricación de distintos modelos de lámparas LED.

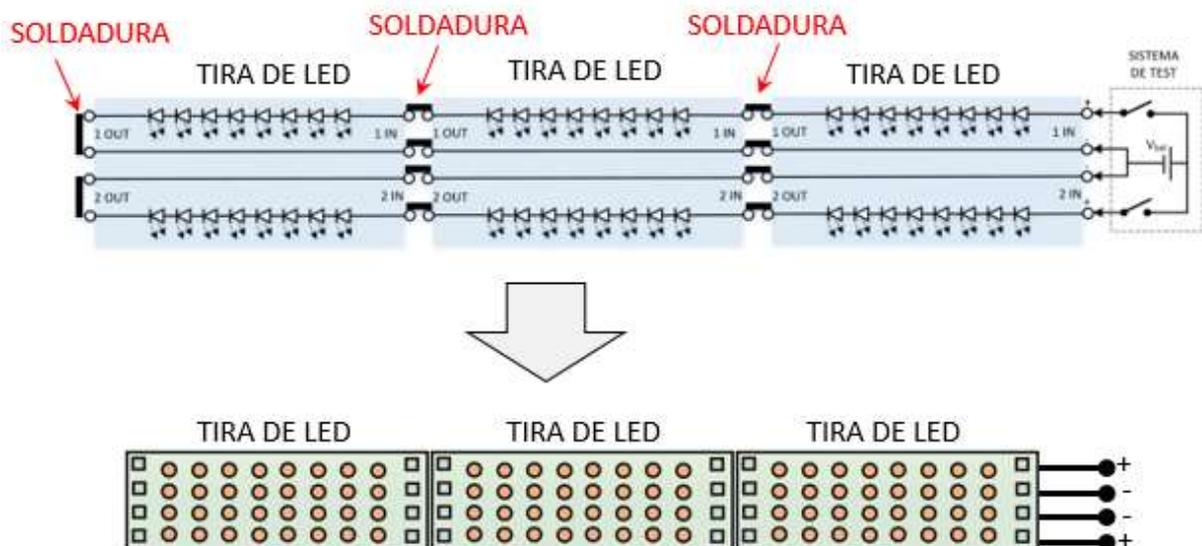


Figura 2.3.- Tira de LED con dos circuitos en serie.

Todo el sistema formará parte de una línea de producción de la empresa NormaGrup Technology S.A. El proceso que lleva a cabo el sistema deberá poder realizarse en dos campos de trabajo distintos. El primero como parte de una de las líneas automatizadas de la empresa, siendo el sistema una de las estaciones de trabajo de la misma. Y, el segundo, en un entorno manual para producciones de menos tamaño cuando la línea automatizada esté ocupándose de otras producciones. El sistema debe estar situado y preparado, por lo tanto, de manera que trabaje en el área de trabajo que se requiera.

En cuanto a la situación en la que se encontrará el sistema, se debe tener también en cuenta que estará trabajando con personas a su alrededor. Es por esto que se deberán integrar todas las medidas de seguridad convenientes, tanto de prevención como de protección, necesarias para cumplir la normativa correspondiente a dicha situación.

Para cumplir todos estos objetivos, se utiliza un robot industrial que utiliza distintas herramientas, y al que se comunica por medio de Socket TCP/IP con el sistema de visión artificial. El sistema de visión artificial esta formado por una cámara y un sistema informático. En Figura 2.4 podemos ver los elementos más importantes del sistema y como se comunican entre ellos.

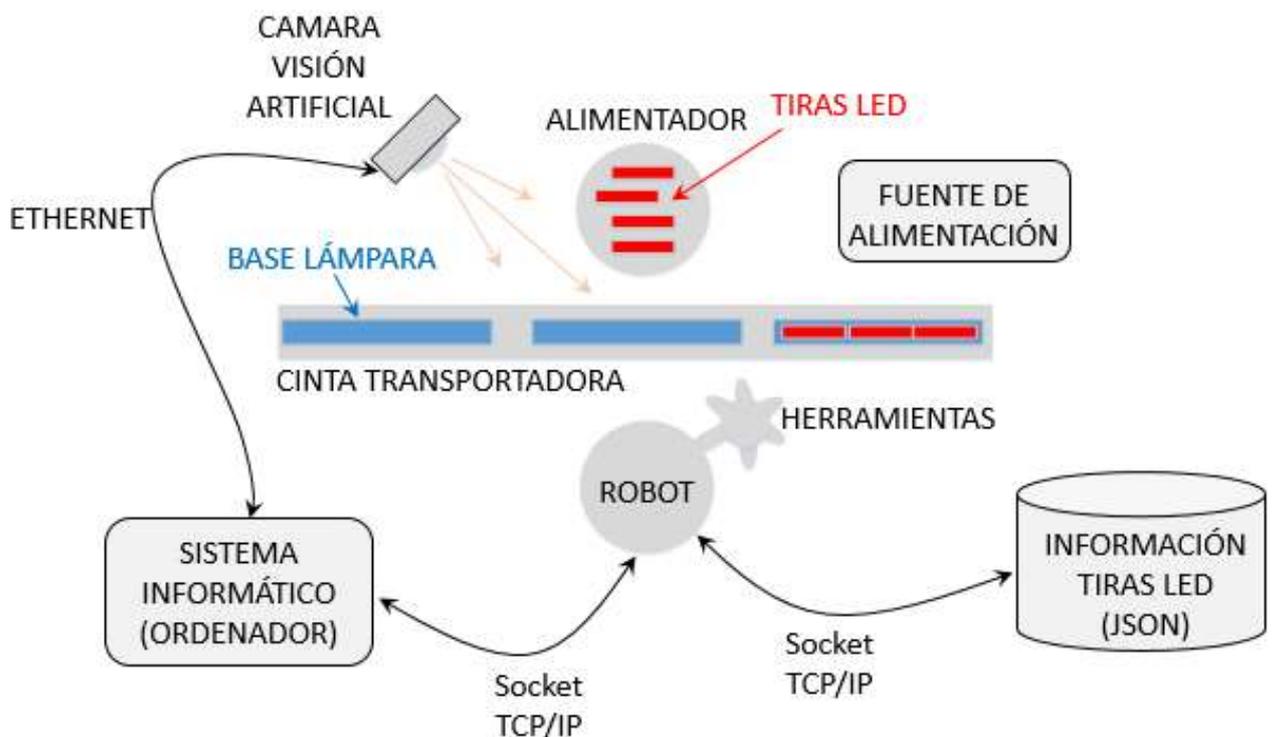


Figura 2.4.- Esquema de componentes del sistema.

2.2- Herramientas hardware disponibles

A continuación, se explican detalladamente las herramientas hardware y sus características principales, que la empresa NormaGrup Technology S.A. puso a disposición para la realización del proyecto.

2.2.1- Robot IRB2600 y controlador IRC5

El robot utilizado es el IRB 2600 de la compañía ABB. Este robot es de gran tamaño, permitiendo un alcance de hasta 1.65 m y con 6 ejes permitiendo gran movilidad. Además, permite ser utilizado para grandes cargas de hasta 20 kg. El robot permite distintas opciones de montaje, siendo posible montarlo en pared, inclinado, sobre estanterías, etc.[4]

El robot industrial IRB2600 ha de ser utilizado con un controlador externo. En este caso, el controlador IRC5 de ABB que es su controlador de 5º generación con un sistema de movimiento del robot optimizado y con menor número de desviaciones y una mayor precisión. El IRC5 tiene como módulo de control el RobotWare base. Entre características importantes que usaremos en nuestra aplicación se encuentran la posibilidad de tener multitarea. Esto es para usar más de una tarea de funcionamiento simultáneo. Mediante el controlador se podrían ejecutar hasta 14 tareas simultáneamente. Además, el controlador cuenta con PC interface que permite la comunicación con un ordenador en la red.

El controlador viene con FlexPendant que cuenta con pantalla táctil y joystick. Permite el movimiento manual del robot y facilita la depuración de código. Su pantalla se puede utilizar además para comunicarse con el usuario ya sea pidiendo información en la pantalla táctil o mostrándola en la misma. [5]



Figura 2.5.- Imagen del robot IRB2600 y el controlador IRC5. Cortesía de [6].

El controlador, como parte inteligente del sistema, estará comunicado con el robot y con sus herramientas para controlar el funcionamiento de las mismas. En el controlador están también implementadas las comunicaciones del sistema

robótico con el sistema de visión artificial y con el ordenador de la línea de producción mediante TCP sockets.

2.2.2- Unidad de soldadura USP5-093A

Para realizar las soldaduras de las placas de LED, se le añade al robot la unidad de soldadura USP5-093A de la empresa Japan-Unix. Esta unidad está diseñada para su uso en sistemas robóticos, incorporándola dentro del proceso de fabricación para la realización de soldaduras de manera automática. El sistema de soldadura está formado por tres componentes: el alimentador del soldador UPM-023, la cabeza del soldador UMC-093A y el controlador del soldador USC-871. Este último es el encargado de controlar los demás elementos durante el proceso de soldadura.[7]

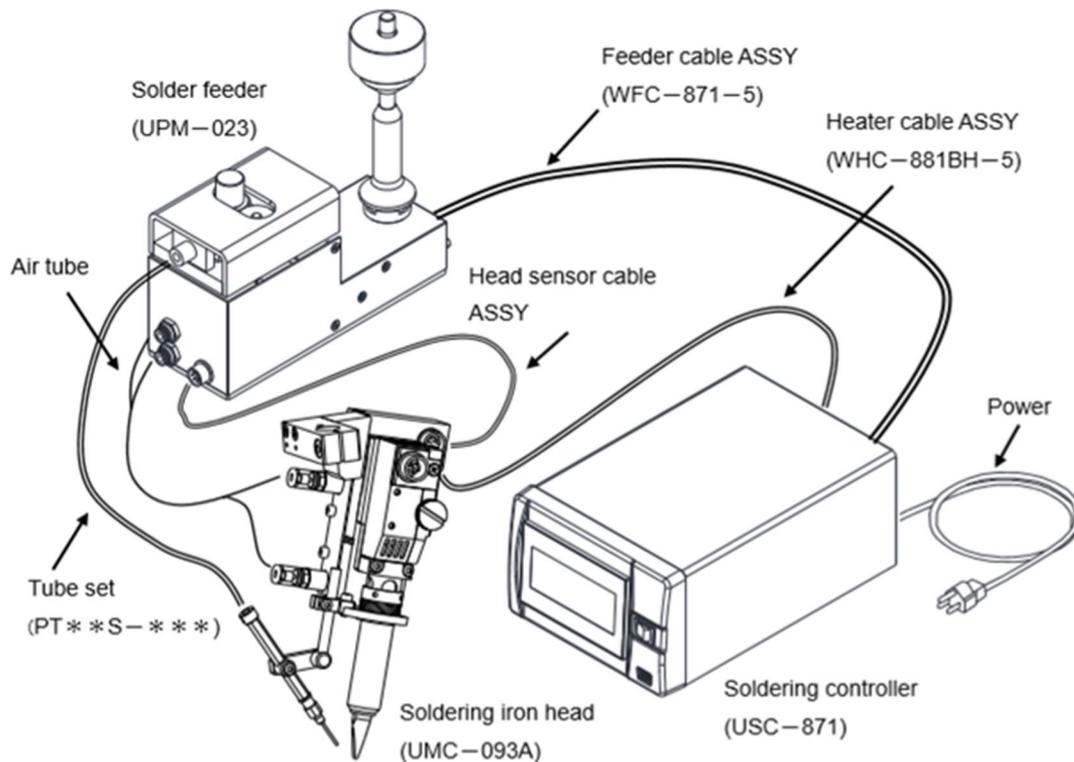


Figura 2.6.- Esquema de la unidad de soldadura USP5 de Japan Unix. Cortesía de [7].

- Alimentador del soldador UPM-023

El equipo UPM-023 es el encargado de alimentar la cabeza del soldador con estaño. Para ello utiliza un motor paso a paso de dos fases. Incluye un encoder que le permite detectar errores durante el proceso de alimentación. El diámetro del material de soldadura debe ser de 1 mm. Como se ve en la figura anterior el material de soldadura sale del equipo para ir directamente a la punta del soldador. Además, el equipo proporciona el aire al soldador para que este pueda subir y bajar la punta y recibe la señal del sensor de final de carrera sobre la posición de la misma. Tiene por tanto una entrada de aire a presión. Para el correcto funcionamiento se debe alimentar entre 0,4 y 0,5Mpa. Por último, está unido al controlador USC-871 mediante el cable de control. Mediante las señales que él mande el controlador el equipo sabrá como deberá comportarse. [7]



Figura 2.7.- Alimentador del soldador UPM-023. Cortesía de[7].

- Cabeza del soldador UMC-093A

La cabeza de soldadura UMC-093A permite realizar soldaduras de manera precisa gracias a la capacidad de desplazar su punta hacia arriba y abajo por medio de aire a presión. Este aire se lo proporciona el alimentador del soldador UPM-023. Dicho movimiento está controlado por finales de carrera, cuyo estado será enviada también al alimentador. Cuenta con un calentador alimentado por el controlador del soldador USC-871, quien conseguirá y mantendrá la temperatura adecuada en cada momento. Tiene una potencia de 250W.

El material de soldadura que llega desde el alimentador UPM-023 a una aguja cuya posición se puede desplazar, ya que está conectada a un cilindro neumático. Las razones por las que se pueda desplazar la aguja es principalmente para su limpieza, ya que al estar está situada tan cercana a la punta de soldadura y le llegan los humos y calor durante el proceso de soldadura. También ha de desplazarse para limpiar el soldador con el limpiador automático. El control de esta aguja debe realizar de manera externa al sistema, en nuestro caso lo se realiza con una electroválvula 3/2 controlada por una salida digital del IRC5. [7]



Figura 2.8.- Cabeza del soldador UMC-093ª. Cortesía de[7].

- Controlador del soldador USC-871

El controlador del soldador USC-871 tiene como función el control de la cabeza del soldador UMC-093A y el alimentador del soldador UPM-023 para que se produzca la soldadura. Cuenta con una pantalla táctil en la que se pueden ajustar sus parámetros principales y donde se muestra el estado del soldador, temperatura y otras condiciones, así como mensajes de error en caso de producirse.

En el controlador se pueden configurar hasta 255 condiciones de soldadura. Estas condiciones se refieren a la velocidad de soldadura, el número de aporte de material, la temperatura, etc. Las condiciones de soldadura se pueden elegir mediante entradas externas o mediante la pantalla táctil. En nuestro caso utilizaremos las entradas externas que están conectadas al IRC5. Hay 8 entradas digitales en el controlador USC-871, correspondientes a los bits para poder elegir entre las 255 condiciones de soldadura.[7]



Figura 2.9.- Controlador del soldador USC-871. Cortesía de[7].

2.2.3- Limpiador automático FA-1000BC

Para limpiar la punta del soldador de manera automática, sin parar la producción se utiliza el limpiador automático FA-1000BC. Este limpiador es de la empresa Taiyo Electric IND, que está dedicada a equipos y accesorios de soldadura. Para la limpieza utiliza rodillos giratorios cuyo movimiento se consigue gracias a un motor de continua de 24 V.[8]



Figura 2.10.-Limpiador automático FA-100BC. Cortesía de Cortesía de [8].

2.2.4- Sistema de agarre y fijación

El robot cuenta con un sistema para agarrar las tiras de LED y posicionarlas en la base de la lámpara. Se tomó la decisión de utilizar ventosas para esta función. Las ventosas se activan mediante válvulas neumática. Se utilizan 5 bares para asegurar una correcta sujeción de las tiras de LED. Las ventosas utilizadas son de la empresa Piab y cuentan con el revestimiento de espuma para superficies rugosas de su línea piGRI. Esto permite el agarre de las tiras de LED a pesar de que tengan los LED sobresalientes.[9]



Figura 2.11.- Ventosas con revestimiento de espuma.

Por otro lado, para sujetar las tiras de LED sobre la base se requieren cilindros neumáticos. Se utilizan los cilindros de simple efecto CUJB6-15D de la empresa SMC Pneumatics. Estos cilindros permiten ahorrar espacio debido a su pequeño tamaño.[10]



Figura 2.12.- Cilindro neumático CUJB6-15D. Cortesía de [10]

2.2.5- Cámara Nano-C1920-1

Para realizar la verificación mediante visión artificial, es necesario el uso de una cámara. En el proyecto se ha utilizada la cámara Nano-C1920-1 de Dalsa Teledyne Dalsa. Esta cámara está especializada para su uso en aplicaciones industriales de visión artificial. Forma además parte de la serie Genie Nano proporcionando una rapidez de Gigabit Ethernet. Permite ajustar todos sus parámetros para darle flexibilidad y que se ajuste a las distintas aplicaciones. Tiene además un tamaño pequeño de 44x29x21 mm.

La cámara Nano-C1920-1 tiene 2 entradas y 2 salidas digitales programables, contando con la posibilidad de activación por medio de software, pero también por hardware. Esto será muy útil para el proyecto conectándose con facilidad a las salidas del controlador IRC5. Tiene además una entrada para su conexión a red por cable Gigabit Ethernet. Cuenta además con software exclusivo para facilitar su uso y su programación. Permitiendo incluso la creación de programas con ayuda de sus herramientas, librerías y clases personalizables. [11]



Figura 2.13.- Cámara Nano-C1920-1. Cortesía de [11].

3- Descripción del proceso

3.1- Entorno de trabajo

Como se ha mencionado en los objetivos, el sistema debe ser capaz de trabajar en dos campos de trabajo distinto. Para esto, el robot se ha situado en el medio de ambos como se puede ver en la Figura 3.1 A un lado del robot está el campo de trabajo automático, este es la línea de producción, y al otro está el campo de trabajo manual.

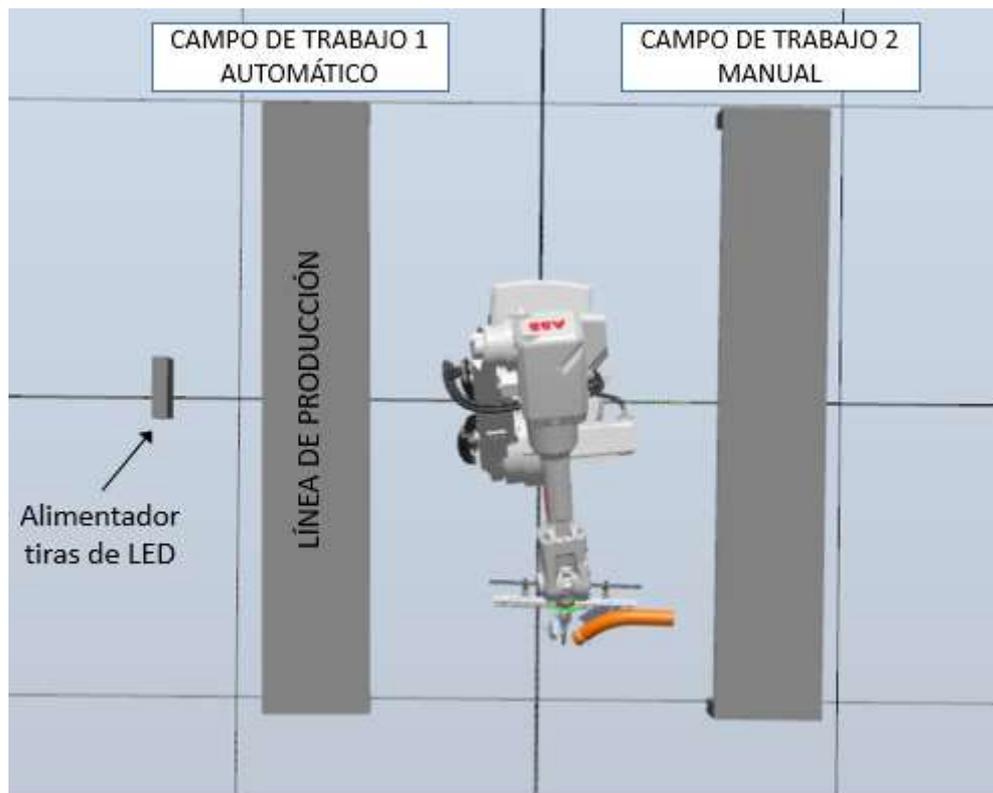


Figura 3.1.- Esquema de los campos de trabajo del robot.

En cuanto a las herramientas del robot, en la Figura 3.2 podemos ver como se posicionan en el mismo. El soldador está a 90° aproximadamente de las ventosas y los cilindros neumáticos. De esta manera se pueden usar dichas herramientas independientemente sin que se molesten unas a otras. Las puntas de test, se sitúan en paralelo al soldador, ya que tras varias pruebas se observó que era la posición más cómoda para encender la lámpara.

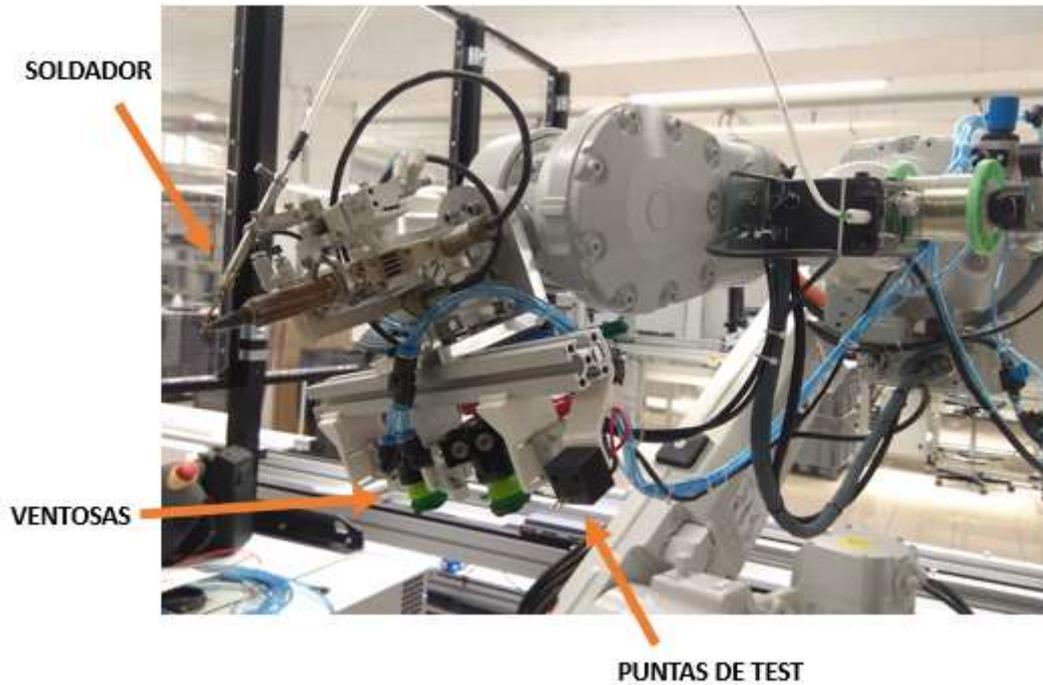


Figura 3.2.- Fotografía de las herramientas del robot.

La colocación del FlexPendant y el controlador del soldador USC-871 está de manera que el usuario pueda alcanzarlos para utilizarlos en caso de necesidad. Su posición es encima del controlador IRC5, que también está a disposición del usuario para poder utilizar la seta de emergencia, cambiar a modo automático y manual o para otras tareas de mantenimiento.

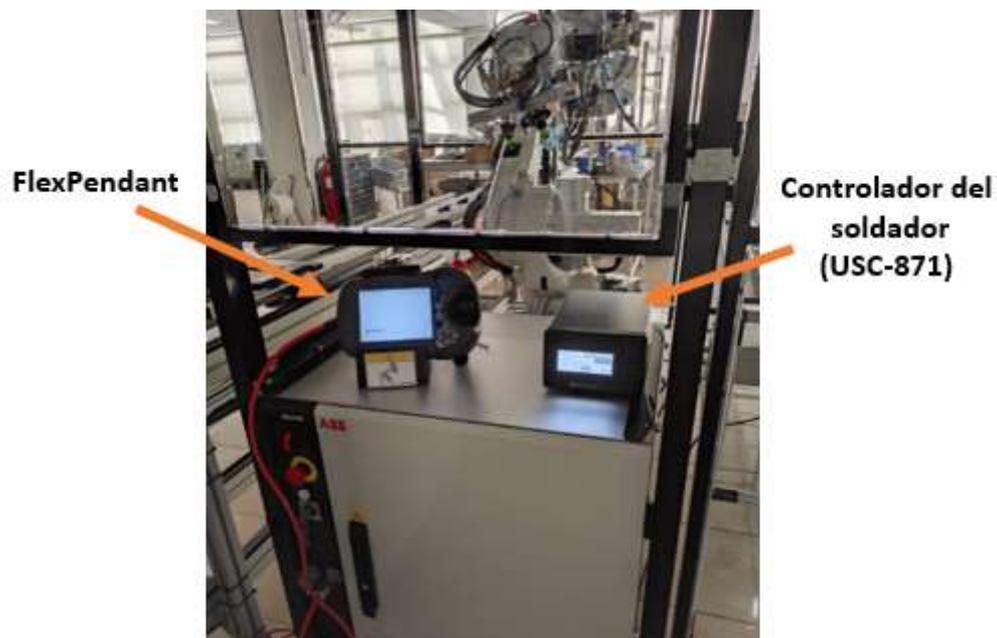


Figura 3.3.- Fotografía del FlexPendant y controlador del soldador.

3.1.1- Campo de trabajo 1 o automático

El campo de trabajo 1 se refiere a la línea de producción de la empresa. El robot es uno de los puestos de trabajo de dicha línea. Anterior y posterior al robot, hay puestos de trabajo llevados a cabo por personas. La base llega a la línea desde el puesto anterior por medio de una cinta transportadora. En la Figura 3.4 vemos una fotografía desde el campo de trabajo 1. El alimentador de tiras de LED está situado en este campo de trabajo. También está en este campo el limpiador del soldador. Al ser una producción automática se ha tratado de minimizar las distancias recorridas por el robot y por tanto el tiempo de duración del proceso.

La cámara de visión está situada en el campo 1. En el campo 2, al ser manual, hay un operario que podrá comprobar si se enciende la lámpara o no sin necesidad del proceso automático. Esta cámara está situada en el techo, para poder alcanzar el ángulo de visión necesario de todo el campo de trabajo.



Figura 3.4.- Fotografía del campo de trabajo 1.

3.1.2- Campo de trabajo 2 o manual

El campo de trabajo 2, a diferencia del anterior no está cuenta con una línea transportadora, ni es un puesto de la línea de producción. Este campo es un puesto fijo, sobre el que el robot realizará sus tareas. En la Figura 3.5 se puede ver una fotografía del campo de trabajo 2. La idea es utilizar el campo 2 para hacer pruebas o coger nuevas posiciones sin interrumpir otros trabajos en la línea de producción. También podrá usarse para producciones pequeñas, cuando la línea esté ocupada con otras tareas. Las bases de la lámpara deben ser posicionadas y retiradas por el usuario manualmente.



Figura 3.5.- Fotografía del campo de trabajo 2.

3.2-Paso 1: Inicio del proceso y caracterización de la lámpara

Debido a que el sistema puede encargarse de la fabricación de distintos tipos de lámparas, se le deben decir cuáles son las características de la fabricación. Para esta tarea se han diseñado recetas se especifican cuáles son los datos de la lámpara, que el robot necesita saber para poder fabricarla y verificarla correctamente. Estas recetas son archivos JavaScript Object Notation (JSON). Esto es, es un formato de texto ligero para el intercambio de información, muy popular por ser de fácil escritura y compresión por los humanos. Además, su estructura es familiar para cualquiera acostumbrado a la programación en lenguajes C (C, C++, C#) o Java, JavaScript, Python entre otros. Nació basado en el lenguaje JavaScript Programming Language Standard ECMA-2623 3rd Edition en diciembre de 1999, pero su uso es independiente del lenguaje de programación utilizado.[12]

Los archivos JSON tiene de nombre el nombre de la receta y de contenido tiene 6 datos. Los nombres de dichos datos, así como su significado y tipo son los siguientes:

- 1- Longitud: Longitud de las tiras de LED que forman la lámpara. Es un número entero.
- 2- NombrePlaca: Nombre de las tiras de LED que forman la lámpara. Es un nombre tipo string. El nombre de las tiras de LED está implementado dentro de la empresa NormaGrup Technology S.A.
- 3- InspecciónVisual: Valor que indica la necesidad de realizar la inspección visual de todos los LED una vez terminada la fabricación de la lámpara. Es un valor booleano.

- 4- NumeroLeds: Número de LED en cada una de las tiras de LED que conforman la lámpara. Es un número entero.
- 5- NumeroPlacas: Número de tiras de LED que hay que poner en la lámpara para su fabricación. Es un número entero.
- 6- VectorInicio: Posiciones de inicio de cada una de las placas de LED. Esto es la posición en la que hay que colocar la siguiente placa respecto a un inicio. Esto se utiliza para poder realizar separaciones entre las placas, que no lleven soldadura. Es un vector de número enteros.

El nombre de la receta puede ser elegido libremente por el usuario. Para facilitar la organización se ha utilizado como nombre de la receta, el nombre de la placa de LED y le número de ellas que lleva la lámpara. Por ejemplo, si tenemos la receta SC1311F1-4.JSON dentro tendría una estructura similar a la siguiente:

```
{"Longitud": 279, "NombrePlaca": "SC1311F1", "InspeccionVisual": true, "NumeroLeds": 96, "NumeroPlacas": 4, "VectorInicio": [0, 279, 562, 841]}
```

El sistema puede recibir las recetas en cualquier momento, independientemente de en qué estado se encuentre el proceso. Sin embargo, en el caso de que es sistema se encuentre en medio de la fabricación de una lámpara y le llegue una receta nueva, no tendrá en cuenta los nuevos datos recibidos hasta que no se termine la fabricación de la lámpara ya comenzada. El primero paso es, por tanto, comprobar si se ha recibido una nueva receta de fabricación o se debe continuar fabricando igual. Para ello se utiliza un servidor de TCP Socket. En la Figura 3.6 se pueden ver los principales elementos y las conexiones correspondientes al paso 1 del proceso.

En modo automático, cuando el sistema este fabricando en el campo de trabajo 1, el encargado de enviar las recetas será el ordenador de la línea de producción correspondiente, que será el que conozca los datos sobre el proceso de producción que se desea realizar. Sin embargo, en caso de querer utilizar el sistema en el modo manual en el campo de trabajo 2 se deberá utilizar la pantalla de usuario para poder introducir la receta que se desee. Desde la misma pantalla de usuario, se pueden también modificar y crear recetas en el momento. Para esto el sistema debe recibir información sobre las características que tiene la siguiente lámpara a fabricar. Esta información la recibe del ordenador de la línea de fabricación o desde la pantalla de usuario. Este ordenador se comunica con un cliente TCP socket. Esta comunicación puede ocurrir en cualquier momento, por ello, el sistema debe estar siempre atento a la comunicación entrante, independientemente de en qué estado se encuentre el proceso de fabricación.

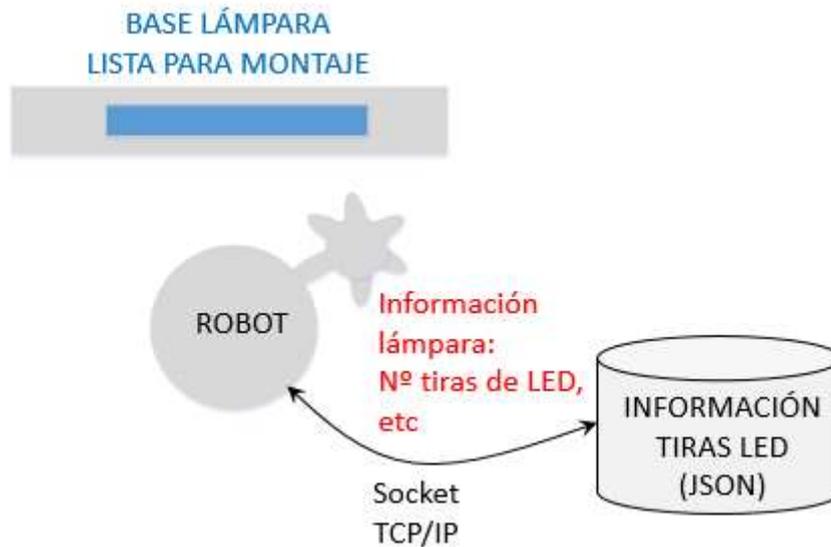


Figura 3.6.- Esquema del paso 1.

3.3- Paso 2: Automatización del proceso de colocación de las tiras de LED sobre la base de montaje

La colocación y fijación de las placas de LED en la base de la lámpara es el comienzo de la fabricación. Esto ocurrirá en el momento que se dé la señal de inicio de proceso. Esta señal es distinta dependiendo del campo de trabajo elegido:

- Campo 1: Campo automático. Se trabaja sobre la cinta transportadora, por lo tanto, la señal de inicio proviene de dicha cinta e indica que ya hay una base preparada para el montaje.
- Campo 2: Campo manual. El operario es el encargado de colocar la base en la posición adecuada y será este por lo tanto el que por medio de un pulsador indique la señal de inicio de proceso.

Cabe destacar que puede haber ocasiones en que este proceso de colocar las piezas no se quiera realizar. El usuario podrá elegir si realizarlo o no por medio del FlexPendant al comienzo del programa y quedará elegido hasta que no se reinicie el mismo. La razón de permitir que no se colocan las piezas viene sobre todo de cara a los procesos realizados en el campo de trabajo 2 de forma manual, donde puede ser más eficiente si el usuario además de colocar la base, coloque las placas de LED sobre la misma. Además, hay que tener en cuenta que el alimentador de placas de LED se encuentra situado en el campo 1, por lo que le lleva al robot algo más de tiempo realizar este paso en el campo 2.



Figura 3.7.- Fotografía de las ventosas del robot sujetando una tira de LED.

En este paso 2, el robot coge cada una de las tiras y las coloca en la posición adecuada dentro de la base de la lámpara. Posteriormente, las fija a la misma haciendo presión sobre los enganches con los que cuentan las tiras de LED y que se introducen en los agujeros de la base. Para coger las tiras de LED se utilizan las ventosas y para hacer presión sobre los enganches de fijación cilindros neumáticos. Toda la programación se realiza en el robot en lenguaje RAPID. En la Figura 3.8 se puede ver el esquema de las partes implicadas en el paso 2.

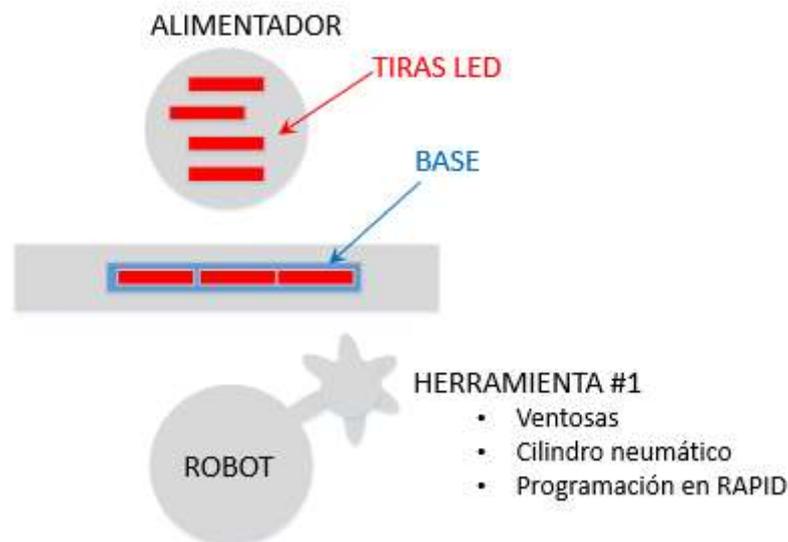


Figura 3.8.- Esquema del paso 2.

3.4- Paso 3: Automatización del proceso de soldadura

Una vez las tiras de leds están situadas y fijadas en la base de la lámpara, hay que soldarlas unas a otras para conseguir continuidad de corriente a lo largo de la lámpara. El proceso debe ser también totalmente automático y de gran precisión, por lo que el robot debe contar también con una herramienta de soldador para dicho trabajo. Para ello se cuenta con el soldador de Japan Unix. En la Figura 3.10 se puede ver el esquema del paso 3.



Figura 3.9.- Fotografía del soldador de Japan Unix realizando una soldadura a las tiras de LED.

El proceso de soldadura de las placas de LED ocurre posteriormente al paso 1 de colocación, o en caso de que este no se realice, será el primer proceso tras la señal de inicio. El orden en que se sueldan las tiras de LED es contrario al orden en que se colocan estas en la base de la lámpara. Esto es así para que, al realizar los procesos seguidos se empiece uno donde ha terminado el anterior y el robot no tenga que recorrer más de lo necesario. Se empieza por lo tanto soldado la última de las placas de LED.

Este proceso se programa en RAPID. En el programa se tiene la información respecto a que soldaduras realizar en cada una de las tiras de LED. Con la información recibida en el paso 1 se sabe el nombre de la tira de LED.

Cada una de las tiras de LED tiene soldaduras en lugares y sitios distintos. También dependiendo de la soldadura a realizar se utiliza una condición de soldadura distinta. Las condiciones de soldadura se programan en el controlador del soldador USC-871 y se escogen por medio de salidas digitales del robot. En las condiciones de soldadura se escoge la cantidad de estaño y velocidad de salida, el tiempo de precalentamiento y si se quiere un único aporte de estaño o dividirlo en varios. Para algunas soldaduras en lugares complicado, se puede requerir también inclinar la punta del soldador, en vez de soldar totalmente recta como es el caso en la Figura 3.9.

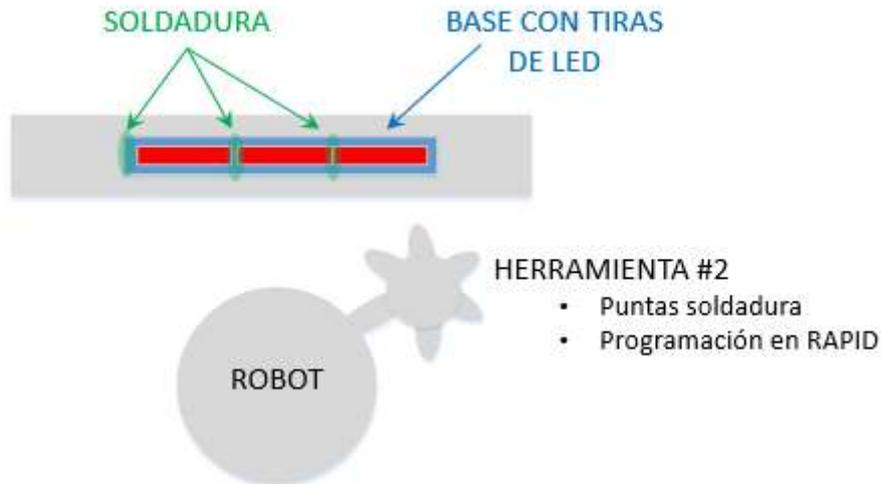


Figura 3.10.- Esquema del paso 3.

3.5- Paso 4: Inicio del proceso de test: Alimentación de la lámpara

Cuando se terminan de realizar las soldaduras el proceso de fabricación habrá finalizado y quedará únicamente la inspección visual. En caso de que haya que realizarla, ya que recordemos que se puede elegir en el paso 1 mediante las recetas si se desea o no realizar una inspección visual, habrá que encender previamente la lámpara. Para el encendido de las lámparas se utilizan puntas de TEST. Estas puntas están conectadas a una fuente de alimentación mediante un driver para el encendido de LED. Este driver mantiene la corriente constante, permitiendo alimentar independientemente del número de tiras de LED que se sitúen en la lámpara, dentro de unos límites. Tiene además protecciones contra cortocircuitos, conexión sin carga y exceso de carga.



Figura 3.11.- Fotografía de las puntas de TEST alimentando la lámpara.

En la Figura 3.12 se puede observar el esquema del paso 4. El encendido de los LED se realiza posteriormente a la soldadura, aprovechando que el robot está situado en la primera tira de LED. Al realizar el encendido es importante tener en cuenta algunas cosas. Lo primero es que las puntas deben estar totalmente en contacto con la placa antes de encender la tensión. Además, deben apagarse y esperar unos segundos antes de separarlas. De esta manera se evitan sobre corrientes que pueden estropear los LED.

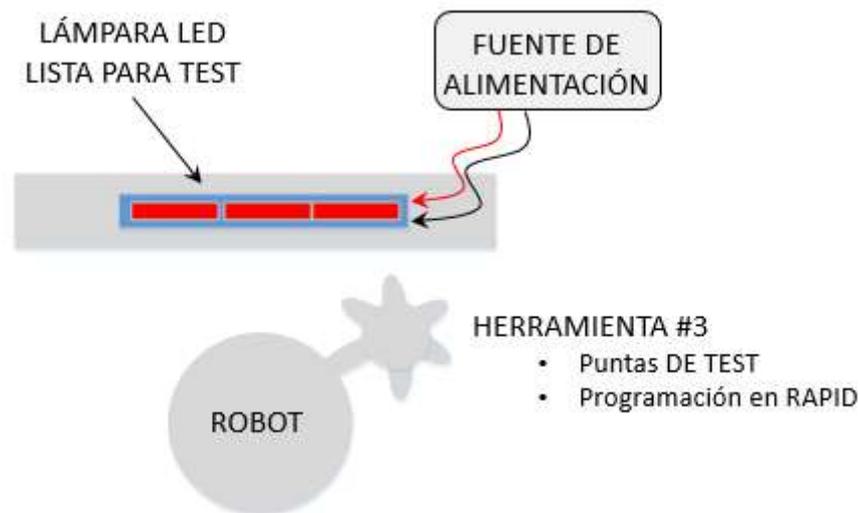


Figura 3.12.- Esquema del paso 4.

3.6- Paso 5: Test de la lámpara mediante visión artificial

Este paso solo se realiza si se ha hecho el anterior, es decir, si es necesario realizar la inspección visual de la lámpara. A diferencia de los pasos anteriores en este caso no es RAPID el lenguaje en que se ha realizado el proceso, ya que no es el robot el que lo lleva a cabo. La programación de la inspección visual se realiza en dos lenguajes, Python para el algoritmo de visión artificial y C++ para la programación de la cámara.

El robot solo se encargará de informar que ha de realizarse la inspección visual. Por un lado, ha de avisar a la cámara para que haga una foto, y, por otro lado, al proceso de visión artificial de Python. Para ello tiene la cámara estará conectada a una de las salidas del robot y utiliza un cliente TCP socket para informar al programa de Python, que hay que realice el algoritmo de visión artificial sobre las fotos tomadas por la cámara. En el programa de Python hay un servidor TCP que está esperando la conexión de dicho cliente. En este programa se programa de Python también está la pantalla de operador que permite cambiar los parámetros de algoritmo de visión artificial, así como enviar las recetas en el campo de trabajo manual, como se explica en el paso 1. En la Figura 3.13 se puede ver el esquema de todos los elementos del paso 5, así como las conexiones entre ellos.

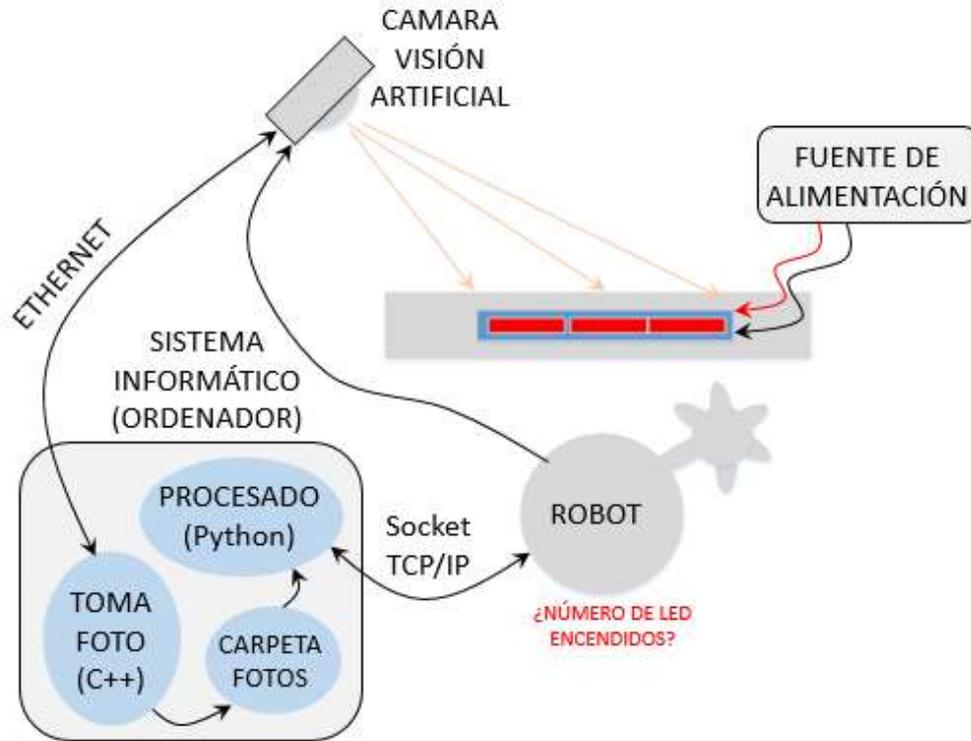


Figura 3.13.- Esquema del paso 5.

3.7-Paso 6: Validación del producto

Para finalizar, una vez realizada la inspección visual, el sistema debe indicar al usuario si el resultado era el esperado y la lámpara funciona correctamente, o si se ha detectado alguna anomalía durante la inspección visual. Además, hay que mostrar al usuario la imagen adquirida, señalando los leds encontrados en ella, la hora y día de adquisición, para que el usuario puede encontrar con facilidad el fallo.

Para decidir si la inspección visual ha sido correcta el robot compara el resultado obtenido por TCP socket del programa de visión artificial realizada en Python, con el valor que había recibido en el paso 1 de la receta. En caso de que dichos valores coincidan es que todos los LED se han encendido correctamente, en caso contrario hay algún problema en la lámpara fabricada. Para avisar al usuario del resultado se mostrará por el FlexPendant un mensaje. Además, se utilizarán dos salidas digitales para que el robot avise a la línea de producción. Dichas señales se denominan 'PASS', cuando la inspección visual ha tenido un resultado correcto, y 'NOT PASS' en caso de que ocurra algún fallo. En la Figura 3.14 se ve el esquema del paso 6.

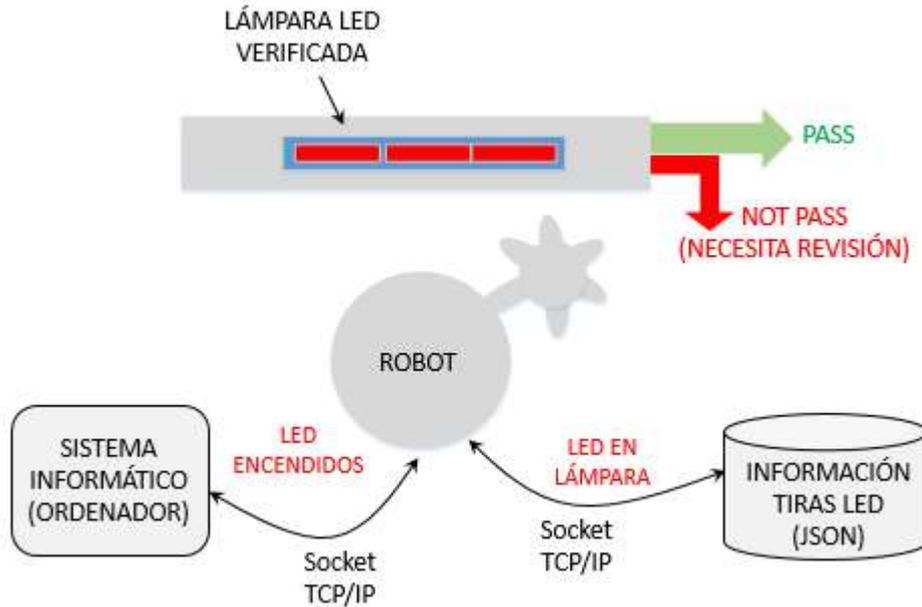


Figura 3.14.- Esquema del paso 6.

4- Trabajo realizado y resultados obtenidos

4.1- Descripción de las comunicaciones digitales entre todos los elementos del sistema

Para las comunicaciones entre procesos en el sistema, se utilizan dos formas. La principal son los sockets TCP/IP. Estos se utilizan para comunicar el robot con el sistema de visión artificial y para que el robot reciba las recetas en el paso 1 del proceso. La programación de los sockets se realiza en Python y en RAPID.

Por el otro lado, el sistema también utiliza Ethernet para la comunicación con la cámara de visión artificial. Más concretamente la cámara permite comunicación 'Gigabit Ethernet'. La comunicación con la cámara se programa en C++.

4.1.2- Sockets TCP/IP

Los sockets son medios de comunicaciones y transmisión de datos entre dos programas. Estos programas, sockets, pueden estar en el mismo equipo o en distintos equipos. Una vez establecida la comunicación entre dos sockets se podrá enviar y recibir información entre ellos con gran facilidad. Algunas de las características más importantes de los sockets, son que permiten una comunicación fiable, sin pérdida de información. También, que se conserva el orden de llegada de los datos al socket, y que no duplica los mensajes, asegurándose siempre un mensaje por cada recepción. Además, es un tipo de comunicación conectada punto a punto. Esto quiere decir que la comunicación se establece antes de cualquier envío de datos.

Los sockets utilizan una arquitectura cliente-servidor. No hay mucha diferencia entre ellos un cliente y un servidor, ya que ambos pueden enviar y recibir mensajes. La diferencia principal es que es el cliente el que debe comenzar la comunicación. El cliente solicita conectarse con el servidor. Para ello debe

conocer la dirección en que este está escuchando dicho servidor. El servidor solo debe estar abierto y aceptar las conexiones entrantes.

Para caracterizar un socket debemos conocer el su dominio y su protocolo y su tipo. Los dominios de las comunicaciones de los sockets indican los protocolos a tomar. También limitan las direcciones en que se puedan crear los sockets. Existen múltiples dominios, dos de los más importantes son:

- AF_UNIF (Address Family Unix): En este dominio tanto el cliente como el servidor están en el mismo dispositivo y cada socket debe tener una dirección única.
- AF_INET (Adress Family Inet): En este dominio el cliente y el servidor pueden estar en máquinas distintas y la comunicación se basa en TCP/IP. Este es el dominio utilizado para las comunicaciones mediante sockets en el trabajo. La dirección del socket se realiza utilizan un número de IP y un puerto. Este es el dominio más habitual para los sockets.

Por el otro lado, los tipos de sockets son principalmente dos:

- SOCK_STREAM: Sockets para flujo de datos. Son implementados en el dominio AF_INET por conexiones TCP/IP. Este es por tanto el tipo implementado en el proyecto.
- SOCK_DGRAM: Sockets para datagramas. Tiene límites en cuanto al tamaño.

Para el funcionamiento del sistema se utilizan entonces Sockets TCP/IP de dominio AF_INET, para conexión entre distintos dispositivos, ordenador y robot, y de tipo SOCK_STREAM. Tanto en Python como en Rapid se programan un cliente y un servidor, ya que existen dos comunicaciones independientes entre ambos dispositivos. Un cliente en Python se conecta con un servidor en RAPID para enviarle al robot las recetas. Y, un cliente en RAPID se conecta a un servidor en Python para pedirle que haga el proceso de inspección visual por medio de visión artificial, y que le devuelva el resultado. Se han elegido así los clientes y servidor por quien comenzaba la comunicación en cada caso. [13] [14]

4.1.3- Ethernet

Ethernet es uno de los protocolos para intercambio de datos más utilizado. El Institute of Electrical and Electronic Engineers(IEEE) define Ethernet como el protocolo 802.3. Desde su creación han existido distintas versiones del protocolo, cada una de ellas proporcionando mayores velocidades que la anterior. Inicialmente la velocidad de Ethernet era 10 megabits por segundo, evolucionado rápidamente hasta los 100 megabits por segundo lo que se denominó 'Fast Ethernet'. En la actualidad, y lo que se ha utilizado en este proyecto para la comunicación con la cámara de visión artificial es el denominado 'Gigabit Ethernet' que alcanza una velocidad de hasta 1000 megabits por segundo.[15]

4.2- Descripción del programa RAPID para manejo del robot

La realización del programa del controlador IRC5 se realiza con ayuda del programa RobotStudio que proporciona ABB. Este programa permite la simulación de todas las trayectorias y posiciones de herramientas del robot sin necesidad de disponer o utilizar el robot ni controlador físicamente. El lenguaje de programación del controlador el RAPID.

El programa realizado en RobotStudio cuenta con dos tareas, T_ROB1 y T_ROB2. Estas tareas se ejecutan de manera simultánea funcionando independientemente una de la otra. La tarea T_ROB1 permite programar movimientos del robot, la otra se utiliza exclusivamente para la comunicación del sistema robótico con el ordenador de la línea de producción mediante un servidor TCP socket. El código correspondiente a las tareas T_ROB1 y T_ROB2 se puede ver en el Anexo 1.

Para el desarrollo del programa y sobre todo para simplificarlo, fue muy importante el uso de los objetos de trabajo (WorkObjects). Los objetos de trabajos son las bases de coordenadas sobre las cuales se definen las distintas posiciones del robot. Gracias a esto simplemente hizo falta definir las posiciones de soldadura y de colocar las tiras de LED en una de las tiras y en un campo de trabajo, y luego variar simplemente el objeto de trabajo respecto a la que se referencian. Para todo esto utilizamos 3 objetos de trabajo:

- 1- Objeto de trabajo fijo (Wo_Fijo): Se definen respecto a esta base de coordenadas los puntos que no varía su posición. Esto es por ejemplo el alimentador de tiras de LED cuya posición nunca cambia, la posición inicial del robot o el limpiador del soldador.
- 2- Objeto de trabajo del campo 1 (Wo_placa1): Esta base de coordenadas está en la posición de la primera tira de LED en el campo de trabajo 1. Sobre este campo de trabajo se utilizan los puntos de soldadura y el punto de colocar la placa en este campo de trabajo.
- 3- Objeto de trabajo del campo 2 (Wo_placa1_2): Esta base de coordenadas está en la posición de la primera tira de LED en el campo de trabajo 2. Sobre este campo de trabajo se utilizan los puntos de soldadura y el punto de colocar la placa en este campo de trabajo.

En la Figura 4.1 se pueden ver cómo están definidos los objetos de trabajo de los campos de trabajo 1 y 2. El objeto de trabajo fijo está situado en el origen de coordenadas del robot, debajo del mismo en el centro. Para que las mismas posiciones del robot sirvieran en ambos campos de trabajo se tomaron los objetos de trabajo en posiciones contrarias como se ve en la Figura 4.1. Se utilizan por tanto las mismas posiciones de soldadura y posicionado de las tiras de LED para ambos campos de trabajo, solo que en un caso o en otro se utilizan refiriéndolas a un objeto de trabajo o a otro.

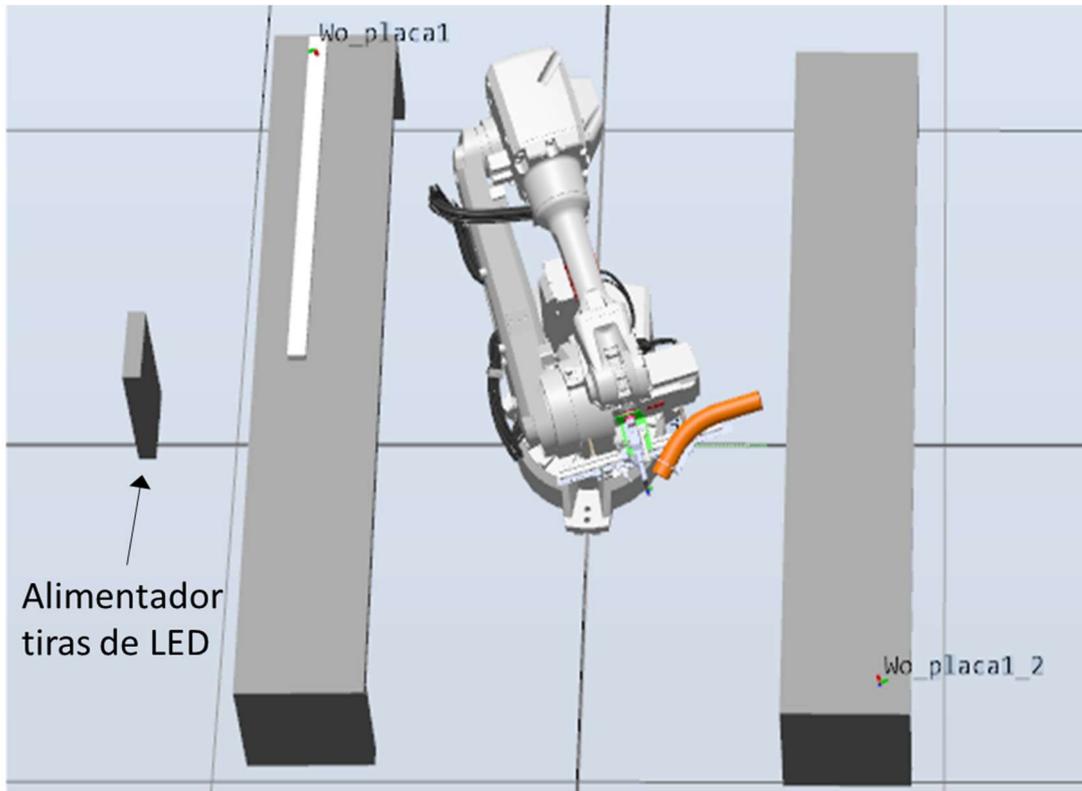


Figura 4.1.- Esquema de los objetos de trabajo para los campo de trabajo 1 y 2.

Además de utilizar las mismas posiciones para ambos campos de trabajo, también se utilizan las mismas posiciones para cada una de las tiras de LED. Es decir, se definen las posiciones de colocar las tiras de LED y soldarlas en la primera de las placas y luego se utilizan para todas. Para esto se trasladan los objetos de trabajo del campo 1 y 2, vistos en la Figura 4.1, a lo largo de la base de la lámpara como se puede ver en la Figura 4.2. La siguiente posición a la que se traslada el objeto de trabajo se calcula utilizando las posiciones de inicio que se reciben de las recetas del paso1.

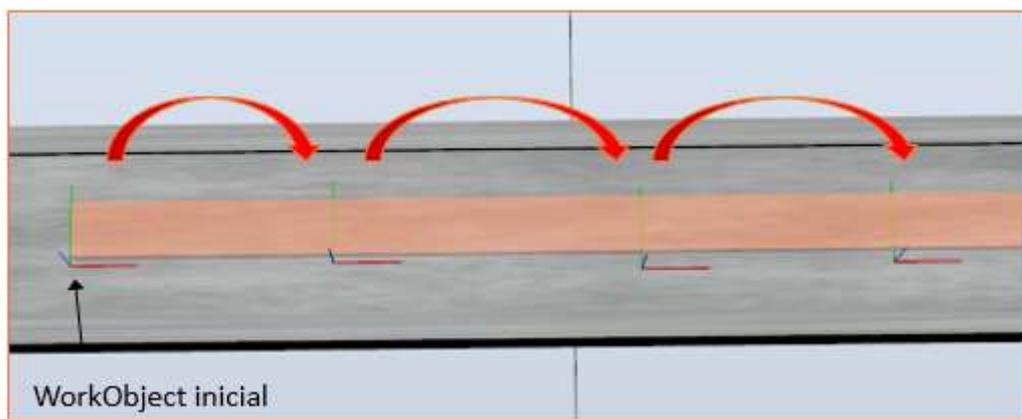
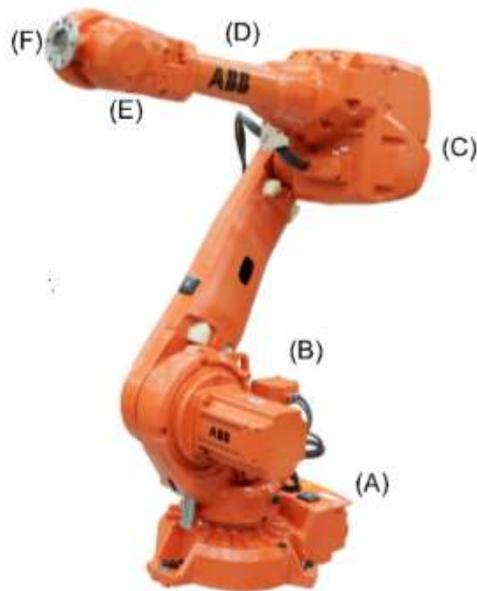


Figura 4.2.- Representación del desplazamiento de los objetos de trabajo por las tiras de LED.

Debido a que se utilizan las mismas configuraciones de las posiciones de cada uno de los ejes del robot en ambos campos de trabajo, hay posiciones a las que el robot se movería con 360° de diferencia entre el campo de trabajo 1 y el campo

de trabajo 2. Esto no se puede permitir en el eje 1, ya que está limitado y no gira 360°, por lo que provocaría un error si el robot lo intentara. Tampoco se puede permitir en el eje 6, en el que están las herramientas. Esto es debido a que a las herramientas llegan múltiples cables, tubos de neumática, conexiones, etc. y estas tienen una longitud suficiente para realizar el giro en un sentido, pero no en el contrario. Por todo esto se ha creado una función que comprueba cada posición del robot antes de que se mueva hacia ella. En caso de que con la configuración de esa posición el eje 1 gire más de 170° en algún sentido o que el eje 6 lo haga más de 185°, cambiará la configuración de la posición sumado o restando 360° al eje que esté dando el problema. La numeración de los ejes del robot se pueden ver en la Figura 4.3.



xx080000437

Pos	Descripción	Pos	Descripción
A	Eje 1	B	Eje 2
C	Eje 3	D	Eje 4
E	Eje 5	F	Eje 6

Figura 4.3.- Esquema de la numeración de los ejes del robot IRB 2600. Cortesía de [22].

En el programa se definen dos herramientas del robot. Por un lado, las ventosas denominadas 'tool_ventosa', y por el otro el soldador denominado 'tool_soldador'. A cada posición el robot acudirá con una u otra herramienta según se lo indique la instrucción de movimiento.

Por otro lado, también ha sido importante en el programa de RAPID el uso del parámetro 'RelTool' en las instrucciones de movimiento (MoveJ, MoveL). Este parámetro nos permite definir un punto respecto a otro. Para ello hay que indicar la distancia o la rotación en los ejes x,y,z. De esta manera se pueden definir puntos de aproximación importante por ejemplo para recoger y colocar las tiras de leds ya que se debe aproximar a estos puntos de manera lineal (MoveL).

4.2.1- Conexiones del robot

El controlador IRC5 utiliza un módulo de entradas y salidas digitales con 16 entradas y 16 salidas. Mediante las salidas realiza el control de las herramientas y se comunica con otros dispositivos del sistema. Las entradas le ayudan a recibir información sobre el entorno del sistema. En la Tabla 1 y Tabla 2 vemos un listado de las entradas y salidas, con una pequeña descripción sobre su cometido.

SALIDAS	
Número	Descripción
OUT 1	Subir tubo de aporte de estaño
OUT 2	PASS (Inspección visual correcta)
OUT 3	Start soldador
OUT 4	NOT PASS (Inspección visual con errores)
OUT 5	Bajarpunta del soldador
OUT 6	Activar modo 'sleep' soldador
OUT 7	Activar calentador del soldador
OUT 8	Reset soldador
OUT 9	Cámara de visión artificial
OUT 10	Activar limpiador del soldador
OUT 11	Activar ventosas
OUT 12	Subir cilindros neumáticos para fijar pieza
OUT 13	Encender fuente de alimentación
OUT 14	Bit 1 Condición de soldadura
OUT 15	Bit 2 Condición de soldadura
OUT 16	Bit 3 Condición de soldadura

Tabla 1.- Conexiones digitales de salidas del controlador IRC5.

ENTRADAS	
Número	Descripción
IN 1	Final de carrera arriba tubo aporte de estaño
IN 2	Final de carrera abajo tubo aporte de estaño
IN 3	Ready soldador
IN 4	Alarma del calentador del soldador
IN 5	Alarma del alimentador del soldador
IN 6	Señal de inicio del campo 1
IN 7	Señal de inicio del campo2

Tabla 2.- Conexiones digitales de entradas del controlador IRC5.

4.2.2- Detalle de implementación del paso 1 en RAPID

El paso 1 corresponde a obtener la información del producto por medio de recetas como se explica en el apartado 3.2. Esto se programa en RAPID en la tarea secundaria del robot (T_ROB2). El objetivo de esta tarea es recibir información sobre el producto a fabricar. Para ello utiliza un servidor socket TCP. Este servidor está continuamente escuchando, por lo que se puede recibir información en cualquier momento, independientemente de lo que esté pasando en la otra tarea y donde este el robot. La información recibida la va guardando en variables que son compartidas entre las tareas T_ROB1 y T_ROB2. La tarea TROB_1, que es donde esta programadas las tareas principales para el movimiento del robot, leerá las variables antes de comenzar cada proceso, después de recibir la señal de inicio y guardará los valores para realizar la fabricación. Todo esto significa que los datos pueden ser recibidos en cualquier momento, pero solo afectarán a la fabricación en su comienzo, nunca a una fabricación que ya haya comenzado.

Como se ha dicho la información recibida son recetas en formato JSON. Las recetas tienen la información del nombre de las tiras de LED, el número de LED de cada una, longitud de las tiras de LED y posiciones de inicio (ver apartado 3.2). El sistema recibe estos valores de JSON como una variable de tipo string. Conociendo la estructura del JSON se extrae los valores de las variables. Para ello se hacen búsquedas de caracteres en el string, buscando las palabras clave de los nombres de variables o las ‘,’ y cambio de línea que indican que se pasa de un valor a otro.

Según el manual de referencia técnica de RAPID de ABB el límite máximo para el tamaño que puede tener un string es de 80 caracteres. Por esta razón no puede ser posible mandar toda la información de las recetas en conjunto. Sin embargo, esto no es un problema basta con enviar varios JSON cada uno parte de la información. El programa comprueba en cada recepción si aparecen todos los parámetros.

4.2.3- Detalle de implementación del paso 2 en RAPID

Este paso se realiza tras la señal de inicio. Recordemos que puede que no sea necesario realizarlo si se ha elegido el campo de trabajo 2 y el usuario ha elegido colocar las tiras de LED manualmente. Por lo tanto, esto es lo primero que se comprueba para saltarse o no el paso.

Para la colocación y sujeción de las tiras de LED se utiliza un proceso de RAPID denominado ‘Proceso_colocar’. Este proceso se encarga de las acciones necesarias para llevar una tira de LED desde el alimentador hasta su posición en la base de la lámpara, y fijarla en la misma. Esta función tiene como parámetro de entrada un objeto de trabajo. Se llamará a la misma tantas veces como tiras de LED se deban colocar en la lámpara. Como se explicó se va desplazando el objeto de trabajo a lo largo de la lámpara como en la Figura 4.2. Se comenzará llamando al proceso con el objeto de trabajo ‘Wo_placa1’ o Wo_placa1_2, que es el objeto de trabajo inicial de los campos de trabajo 1 y 2, y después se desplazará dicho objeto de trabajo en su eje X. Se repite el proceso hasta terminar de colocar todas las tiras de LED que lleve la lámpara.

Dentro del proceso ‘Proceso_colocar’, cada vez que se llame se realizan las siguientes acciones:

- 1- Mover el robot hasta alimentador de tiras de LED con la herramienta 'tool_ventosas' y el objeto de trabajo 'Wo_Fijo'. Ya que el punto de recogida de las piezas es siempre el mismo.
- 2- Encender la salida correspondiente a las ventosas y esperar 0.5 segundos para garantizar el agarre de la tira de LED.
- 3- Moverse hasta el Target_dejada donde dejar la pieza sobre la base, utilizando la herramienta 'tool_ventosas' y el objeto de trabajo recibido como entrada de la función.
- 4- Esperar 0.5 segundos y desactivar la salida correspondiente a las ventosas.
- 5- Subir la herramienta 4 mm por encima del punto anterior para dejar la placa de LED posicionarse bien en su sitio gracias a la gravedad.
- 6- Accionar la salida correspondiente a los cilindros neumáticos y fijar de esta manera la placa por sus enganches de fijación.
- 7- Esperar 0.5 segundos y desactivar la salida de los cilindros neumáticos.
- 8- Subir 100 mm por encima del punto de dejada, asegurándose así de que el robot tendrá el espacio suficiente para poder realizar el siguiente movimiento, sea cual sea este.

4.2.4- Detalle de implementación del paso 3 en RAPID

Una vez se hayan terminado de colocar todas las tiras de LED en la base de la lámpara, se podrá comenzar la soldadura. Se empieza, por lo tanto, soldado la última tira de LED, que es también la última colocada por el robot en el paso 2. De esta manera, se ahorra recorrido y además se termina la soldadura al principio, donde se enciende la lámpara.

Existen múltiples tiras de LED, cada una de ellas con un circuito interno y, por lo tanto, cada una necesitando unas soldadoras en sitios y con posiciones diferentes. El programa lo que recibe en el paso 1 es el nombre de la tira de LED. Con este nombre consulta las denominadas recetas de soldadura. Estas recetas relacionan dicho nombre con las posiciones de las soldaduras que necesita, el ángulo en que realizar dicha soldadura y la condición. Para estas recetas de soldaduras utiliza vectores de 'strings' como el siguiente:

```
CONST string SC1311F1{10,3} := [ ["SC1311_Punto2","1","0"],
["SC1311_PuntoUnion1","1","0"],
["SC1311_PuntoUnion2","2","0"],["SC1311_Punto1","1","0"],["SC1311_PuntoFinal3","1","0"],
["SC1311_PuntoFinal2","1","0"],["SC1311_PuntoFinal1","1","0"],["PuntoTest1","null","null"], ["" , "" , ""],["" , "" , ""]];
```

Una característica importante de estos vectores de 'string' que son las recetas de soldadura es que todos son de longitud 10 a pesar de no necesitarlo. Esto es necesario ya que en lenguaje RAPID no es posible el cambio de tamaño dinámico de vectores y fue necesario el uso de vectores con la misma longitud para todas las recetas. Simplemente se guardará el último valor válido en cada vector y las posiciones a partir de esa no se tendrán en cuenta. Por ejemplo, en el ejemplo anterior las últimas dos posiciones no son válidas y están vacías. Además, la última posición del vector (PuntoTest1 en el ejemplo) se refiere al punto donde debe ir el robot para encender la lámpara, como se explica en el siguiente apartado. Es por eso, no se rellenan los valores de ángulo de soldadura y condición de soldadura, ya que no se refiere a una soldadura.

Los nombres de las posiciones que se incluyen en las recetas de soldadura no pueden escogerse al azar. Hay cuatro tipos de posiciones:

- Posiciones de soldadura que incluyen una única tira de LED: Se realizan en todas las tiras de LED. El nombre de estas posiciones no puede incluir ninguna de las palabras 'Test', 'Union' o 'Final'. Estas soldaduras se realizan en las tiras de LED sobre sí mismas.
- Posiciones de unión de una tira de LED con la siguiente: Estas posiciones deben llevar la palabra 'Union' en su nombre. Solo se realiza cuando no haya separación entre placas. La separación entre placas se puede ver en la Figura 4.4. Estas separaciones ocurren ya que a veces la empresa utiliza las mismas bases de lámparas para distintas tiras de LED y sobra espacio por la diferencia de longitud.
- Posiciones de la placa final: Se realizan únicamente en la tira de LED final. Esta es distinta a las otras ya que lleva soldaduras para cerrar el circuito y por tanto deben diferenciarse de las demás. Deben llevar en su nombre la palabra 'Final'.
- Posiciones de test: Se utilizan para encender la lámpara. Llevan la palabra 'Test'. Se explica con más detalle en el siguiente apartado.



Figura 4.4.- Fotografía de una soldadura con las placas separadas frente a una con ellas juntas.

Lo que hace el programa de RAPID es buscar primero las posiciones en el vector de soldadura que contengan la palabra 'Final' y realizarlas en la última placa. Posteriormente, va trasladando el objeto de trabajo a la posición de inicio correspondiente y va realizando las soldaduras que no tengan ni la palabra 'Final' ni 'Union' ni 'Test'. Para las que tengan la palabra 'Union' comprobará si la posición de inicio de esa placa menos la de la anterior es mayor que la longitud de la tira de LED, y por lo tanto no están separadas. Solo las realizará si no hay esa separación.

Al igual que en la fase 2 se utiliza un único proceso para realizar las acciones correspondientes a casa soldadura. En este caso llamado 'Proceso_soldadura' que se llamará a la misma tantas veces como soldaduras se quieras realizar. El proceso 'Proceso_soldadura' tiene cuatro parámetros de entrada: un objeto de trabajo, una posición a la que deba ir el robot, una condición de soldadura y un ángulo de soldadura. El proceso necesita tantos parámetros para poder realizar bien las soldaduras. Los pasos que se realizan en este proceso son:

- 1- Para indicarle al controlador del soldador que condición de soldadura se desea utilizar se utilizan tres salidas del IRC5 correspondientes a tres bits. El primer paso es por tanto indicar el valor de dichos bits en función del byte recibido como condición de soldadura en el parámetro de entrada de la función. Dependiendo de la condición de soldadura, se elegirá cuanto tiempo el soldador deba estar apoyado antes de levantarse. Se elige también esta variable llamada 'TimeSolderDown'.
- 2- Mover el robot al 'Target' indicado en el parámetro de entrada, con el objeto de trabajo también indicado en el parámetro de entrada y con la herramienta 'tool_soldador'.

- 3- Esperar que el soldador está 'READY' comprobando la entrada del IRC5 correspondiente.
- 4- Iniciar la soldadura realizando un pulso en la salida digital correspondiente.
- 5- Esperar el tiempo indicado por la variable 'TimeSolderDown' y después levantar el soldador 7 mm hacia arriba.
- 6- Esperar a que termine la soldadura mirando que la entrada 'READY' vuelva a estar lista.
- 7- Levantar el soldador 20 mm hacia arriba, preparándolo así para cualquier movimiento que tenga que realizar a continuación.

Cada vez que se realiza una soldadura se incrementa un contador. Al terminar el proceso, incluido la inspección visual se comprobará si ese contador ha superado el valor 50. Es decir, si se han realizado más de 50 soldaduras desde la última limpieza. En caso de ser así se deberá limpiar la punta del soldador llamando al proceso 'LimpiezaSoldador'. Esta función se encarga de:

- 1- Separar el tubo de aporte de estaño del soldador, activando la salida del IRC5 correspondiente. En la Figura 4.5 se puede ver el soldadora con el tubo de estaño separado. Esto se hace para que no se estropee al limpiar la punta del soldador.
- 2- Comprobar con el final de carrera del mismo que se ha separado correctamente.
- 3- Llevar al robot a la posición donde se encuentra el limpiador con la herramienta 'tool_soldador', el objeto de trabajo 'Wo_Fijo', ya que es un punto que no cambia nunca independientemente de la lámpara que se fabrique.
- 4- Encender el motor del soldador con la salida digital del IRC5 correspondiente.
- 5- Esperar 3 segundos para que se limpie bien el soldador.
- 6- Apagar el motor del soldador desactivando la salida anteriormente activada.
- 7- Volver a la posición 'Home', que es la posición inicial.
- 8- Volver el tubo de aporte de estaño a su posición normal.
- 9- Comprobar que está bien colocado mirando la entrada digital del IRC5 correspondiente al final de carrera cerrado.



Figura 4.5.- Fotografía del soldadora con el tubo de estaño separado.

4.2.5- Detalle de implementación del paso 4 en RAPID

En caso de tener que realizar la inspección visual, el primer paso será encender la lámpara. Para comprobar de que haya que realizarlo se utilizan se comprueba la variable 'InspeccionVisual'. La posición a la que se debe acudir el robot para encender la lámpara, viene dada también por la receta de soldadura, en este caso por las posiciones que contengan la palabra 'Test'. Se utilizan entonces a el proceso 'ProcesoInspeccionVisual' con el target como parámetro de entrada. En esta función se realizan las demás acciones relacionadas con la inspección visual.

Lo primero que hace el proceso 'ProcesoInspeccionVisual' es llevar el robot hasta el punto de encendido. Para encender se utilizas las puntas de TEST que están situadas en paralelo con el soldador, por lo que dicha posición se ha guardado utilizando la herramienta 'tool_soldador'. Las lámparas se encienden en su punto inicial siempre por lo que se podrá utilizar el 'Wo_placa1'.

Una vez se ha alcanzado la posición de encendido se activará la señal que da corriente a la lámpara. Es importante esperar a que se haya alcanzado la posición antes de encender para que no se dañen los LED y esperar un poco tras apagar la alimentación para separar las puntas de TEST.

4.1.6- Detalle de implementación del paso 5 en RAPID

Una vez y en caso de que se encendida la lámpara el siguiente paso es realizar la inspección visual. Esa parte está programada en Python, sin embargo, el robot ha de avisar de que se realiza.

Para avisar a la cámara utiliza una de sus salidas digitales. Por el otro lado para avisar al sistema de visión artificial creado en Python, se realiza un cliente TCP socket. El cliente se conectará con el servidor TCP Socket creado en Python. Para hacer esta conexión se utiliza una función denominada

‘Establecer_conexión. Esta función tiene como parámetro de entrada el valor que se quiere enviar al servidor, y como parámetro de salida devuelve el valor recibido por dicho servidor. El valor que se le manda el servidor es ‘1’, para recibir de vuelta el número de LED encendidos. Los detalles sobre esta comunicación se pueden ver en el apartado 4.4.2.

4.2.7- Detalle de implementación del paso 6 en RAPID

El último paso a realizar tras la inspección visual, es tomar decisiones sobre si la fabricación ha sido exitosa o si ha habido algún problema y el producto necesita una revisión posterior. Para esto, simplemente se compara el valor recibido de Python por TCP socket en el paso anterior, con el valor que se obtiene de la receta en el paso 1. Si estos valores son iguales, hemos tenido una fabricación exitosa. Si estos valores son distintos, se necesitará algún tipo de revisión.

Para avisar del resultado se utilizan las señales PASS y NOT PASS, siendo la primera activada cuando la fabricación sea exitosa y la segunda en caso contrario. Estas son salidas digitales del IRC5. Además, se mostrará un mensaje por el FlexPendant indicando el número de LED encendidos y si era el número esperado o no. También se podrá observar el resultado en la pantalla de usuario como se explica en los siguientes apartados.

4.3- Descripción de la configuración de la cámara y el programa en C++ para la captura de imágenes

4.3.1- Archivo de configuración de la cámara

El archivo de configuración de la cámara sirve para configurar las características más importantes. Estas van desde las propiedades de la imagen tomada hasta la configuración de sus entradas y salidas digitales. Para realizar el archivo de configuración se utilizó el programa CamExpert. El programa de configuración completo se puede ver en el Anexo 3. Las características más importantes del archivo de configuración son:

- Valor de ganancia 1
- Valor de tiempo de exposición 100
- Adquisición de imagen por modo trigger activado
- Flanco de subida en la entrada 1 como fuente de trigger
- Contador y temporizador desactivados
- Formato de imagen: BayerRG 8bit, tamaño 1936x1216 pixeles (tamaño máximo)
- Periodo de rebotes 100ms. Este es el tiempo tras el cual un flanco de subida se considera un pulso. En RAPID se utilizan pulsos de 200ms, por lo que se cogió el valor medio.

4.3.2- Descripción del programa adquisición de imágenes

Utilizando la información y los programas de ejemplo disponibles para C++ y con ayuda del programa Visual Studio 2017 se ha creado el programa para el uso de la cámara Teledyne DALSA Genie Nano-GigE. Este programa inicializa la cámara con un archivo de configuración específico. Este archivo de configuración lo busca dentro de la carpeta donde se encuentre el ejecutable. En caso de no haber archivo de configuración en dicha carpeta el programa finaliza.

En caso de que haya más de un archivo de configuración elegirá el primero por orden de lista del nombre del archivo. Una vez inicializada la cámara y el buffer donde se adquieren las imágenes se registra una función (callback) que se ejecute cada vez que se produzca el evento de adquisición de una imagen (Valid Frame Trigger).

Si elegimos el archivo de configuración generado con CamExpert que asociaba el flanco de subida de la entrada 1 a la adquisición de una imagen, el evento se produce cada vez que ocurra dicho flanco. La programación comienza entonces la grabación y espera en un bucle infinito a que se produzca dicho evento.

La función callback a la que va el programa cada vez que ocurre el evento, guarda la información dentro del buffer en tres archivos de imágenes en formato bmp. Los archivos donde se guardan las imágenes se encuentra en la misma carpeta que el ejecutable y tienen los nombres 'FotoSaperaCamara_ 1.bmp', 'FotoSaperaCamara_ 2.bmp' y 'FotoSaperaCamara_ 3.bmp'. Conociendo la situación y nombre de dichas imágenes el programa de visión artificial en Python las utilizará para detectar el número de LED encendidos.

4.4- Descripción del programa de Python

En el programa de Python se ha realizado toda la programación relacionada con la pantalla de usuario y con la inspección visual por medio de un algoritmo de visión artificial. Todo esto se divide en cuatro tareas principales. Las tareas que se realizan en el programa de Python son:

- 1- Realizar el algoritmo de visión artificial para la correcta detección de los LED encendidos en la lámpara.
- 2- Permitir en la pantalla de usuario modificar los parámetros del algoritmo de visión artificial, así como observar los LED detectados en cada uno de los procesos.
- 3- Manejar la comunicación con el robot para informarle del resultado en el proceso de visión artificial y que este le avise de cuando ha de realizarse dicho proceso.
- 4- Poder enviar recetas al robot desde la pantalla de usuario, así como crearlas y modificarlas. Esto es para el manejo en el campo de trabajo manual.

Para realizar estas tareas, el programa utiliza una pantalla de usuario creada con PyQt5 y dos hilos de la clase Qthread, que permite el manejo de 'signals' y 'slots' para la comunicación entre hilos.

4.4.1- Archivo de configuración de Python

En el programa de Python se utiliza un archivo de configuración. Este archivo contiene datos que se quieren guardar tras cerrar el programa para que ya estén las cosas configuradas la próxima vez que se ejecute. El archivo de configuración se realiza utilizando el módulo 'configparse' para archivos de configuración de Python.[16]

El archivo de configuración tiene dentro los siguientes valores:

- Threshold máximo: valor del threshold máximo utilizado en el algoritmo de visión artificial blob detector.
- Threshold mínimo: valor del threshold mínimo utilizado en el algoritmo de visión artificial blob detector.
- Área máxima: valor del área de blob máxima utilizada en el algoritmo de visión artificial blob detector.
- Área mínima: valor del área de blob mínima utilizada en el algoritmo de visión artificial blob detector.
- ROI (Región de interés): La región de interés es la zona en la que el algoritmo de visión artificial buscará los blobs. Se concentra por tanto en la zona de la imagen donde está la lámpara.
- IP: IP del ordenador donde se está ejecutando el programa, y donde se debe, por lo tanto, abrir el servidor de TCP socket.

Los valores de threshold máximo y mínimo y área máxima y mínima, pueden ser cambiados por el usuario durante la ejecución por medio de la pantalla de usuario, como se ve en el apartado 4.5. Los cambios en estos valores se deben a cambios principalmente en la luz exterior. La región de interés también puede ser modificada por el usuario, en caso de que se modifica la posición de la cámara o haya cambios en la posición de la línea de producción.

La lectura del archivo de configuración se realiza nada más ejecutar el programa. Los datos del archivo se guardan cada vez que se cierre la pantalla de usuario, que finaliza el programa. Si cuando ejecutamos el programa no existe ningún archivo de configuración, se creará uno con valores por defecto. Para la IP, se preguntará por ventana de comandos cual es la IP del ordenador. Después el usuario podrá modificar los valores en la pantalla de usuario o incluso directamente modificar el archivo de configuración creado.

4.4.2- Descripción del algoritmo de visión artificial e implementación en Python

Como algoritmo de visión artificial se utiliza un Blob detector (detector de Blobs). Este algoritmo es muy adecuado para nuestro caso, en que tratamos de distinguir LED que se hayan encendido, ya que el algoritmo es capaz de distinguir partes de una imagen que tenga alguna característica en común. Los pasos del algoritmo Blob detection son los siguientes:[17]

- 1- Umbralizar (Thresholding): La imagen original se convierte en varias imágenes binarias. Las imágenes son creadas desde un umbral mínimo hasta un umbral máximo, avanzando una cantidad en concreto. Todos estos parámetros pueden ser elegidos por el usuario para parametrizar el algoritmo.

- 2- Agrupar: Todos los píxeles blancos se agrupan juntos formando los denominados Blobs binarios. Esto se realiza en todas las imágenes binarias.
- 3- Fusión de los Blobs binarios: para unir los Blobs binarios se calcula el centro de cada uno. Se considerarán del mismo Blob, los centros de los Blobs binarios que estén a una distancia inferior a la mínima distancia entre Blobs definida por el usuario.
- 4- Cálculo de centro y radio: una vez se han fusionado los distintos Blobs binarios en los Blobs finales, se calcula el radio y el centro de estos últimos. Teniendo así la posición y tamaño de los Blobs.

Utilizando un Blob detector se pueden elegir los parámetros según los cuales se quieren distinguir los Blobs del resto de la imagen. Estos parámetros según los cuales se pueden filtrar los Blobs son: color, tamaño en píxeles y forma. Dentro de los filtros de forma, puede ser por circularidad, convexidad y relación de inercia. En la Figura 4.6 se pueden observar visualmente a que se refieren cada uno de los parámetros mencionados.

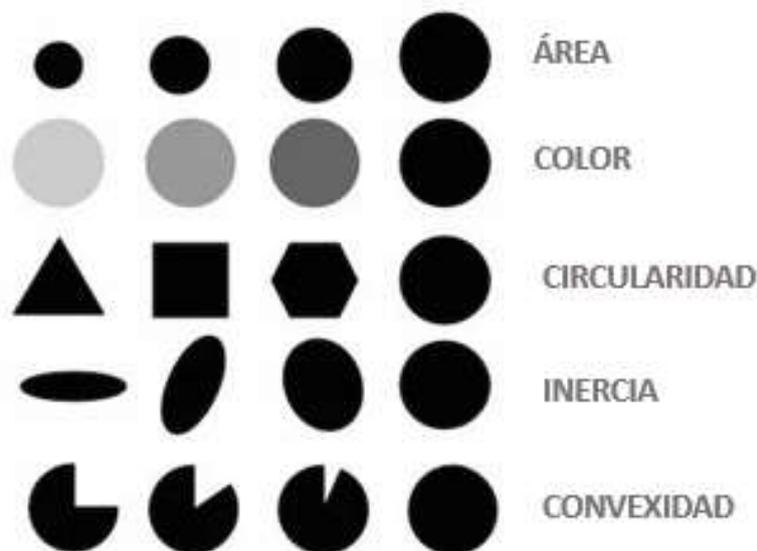


Figura 4.6.- Esquema de los parámetros a elegir en el Blob detector. Cortesía de [17]

La implementación del blob detector se ha realizado mediante la librería OpenCV (Open Source Computer Vision Library) que incluye múltiples algoritmos de visión artificial, así como funciones para el procesamiento, lectura y escritura de imágenes. Esta librería está disponible para su uso en los lenguajes de programación C++ y Python. La aplicación de este trabajo está realizada en Python mediante el editor Anaconda.[18]

La cámara adquiere 3 imágenes, muy oscuras, según los parámetros indicados en el archivo de configuración de la cámara. Se utiliza el Blob detector sobre las 3 imágenes para obtener así varios resultados. Posteriormente, se busca cual es el resultado más veces obtenido, es decir la moda, y se escoge como resultado del proceso.

4.4.3- Cliente TCP socket

Uno de los hilos creados en Python es para crear clientes, que se conecten con el robot y envían las recetas explicadas en el paso 1, apartado 3.2. Para hacer esto el usuario debe utilizar la pantalla de usuario, en la que escogerá los valores de las recetas y cuando enviarlas. La explicación más detallada de esta pantalla se puede ver en el apartado 4.5.

El hilo mandará la receta cuando el usuario se lo indique mediante la pantalla de usuario. Para ello se utiliza una función (slot) y una señal que refleje dicha situación. Como parámetro de entrada de la función slot se le manda el nombre de la receta que el usuario haya seleccionado. Los valores de la receta los saca de la carpeta Receta, leyendo el archivo. JSON con el nombre de dicha receta.

Es importante el hecho de que a RAPID se no se le puede enviar el JSON entero, ya que como se ha explicado antes no se permiten strings de tanto tamaño. Es por esto que se dividen los datos, y se van enviando JSON más pequeños, poniendo dos variables en cada uno. Cada vez que se mandan datos se recibe el mensaje 'ACK' del cliente, como se puede ver en la Figura 4.7. Una vez enviada toda la información, el cliente se desconecta del servidor y así él sabe que ha terminado de enviarle toda la información. No es necesario enviar los valores de la receta en el orden mostrado en la figura. Cualquier orden sería válido para el servidor.

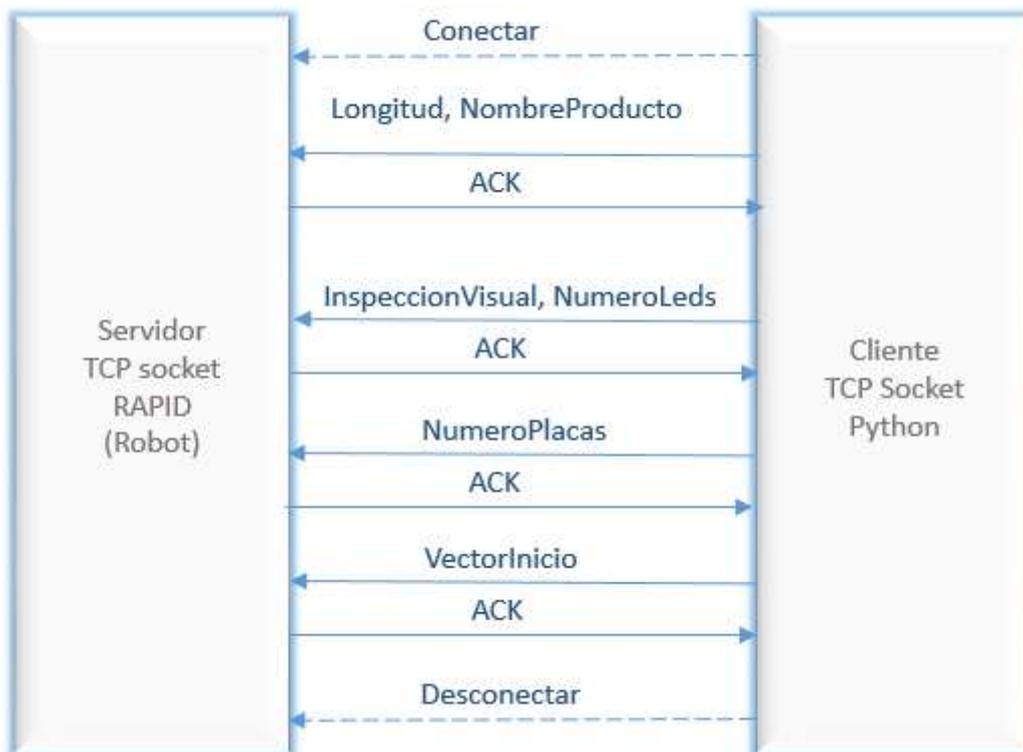


Figura 4.7.- Esquema de la comunicación del cliente TCP socket de Python.

4.4.4- Servidor TCP socket

El otro hilo programado en Python, es un servidor TCP socket abierto constantemente y a la espera de la conexión de un cliente. Este hilo es el encargado de esperar por la señal de que se realice la inspección visual y de enviar el resultado. La IP en la que se cree dicho servidor ha de ser la IP del ordenador utilizado, es por esto que el hilo recibe el valor de la IP del archivo de configuración. Se debe comprobar que en dicho archivo se encuentra el valor de IP correcto en cada caso.

Cuando el servidor reciba una conexión entrante, realiza una comprobación, enviéndoles el mensaje ‘comenzar?’. Si la respuesta del cliente es un 1 sabrá que ha de avisar de que se ha conectado el robot y se debe realizar la inspección visual. La idea es que, si se quiere utilizar el servidor en un futuro para otras aplicaciones, solo haya que enviarle al servidor otros valores.

Cuando se conecte el robot al servidor, este último avisará mediante ‘signal’ para que se realicen las funciones de visión artificial, y se quedará a la espera del resultado. Cuando se termine el proceso de visión y se haya mostrado el resultado, también mediante un ‘signal’ se le devuelve el resultado para que el servidor se lo envíe a su vez al cliente robot. Una vez terminado esto se desconectan ambos cliente y servidor cerrándose la conexión y el cliente vuelve a la espera. El esquema con la conexión se ve en la Figura 4.8.

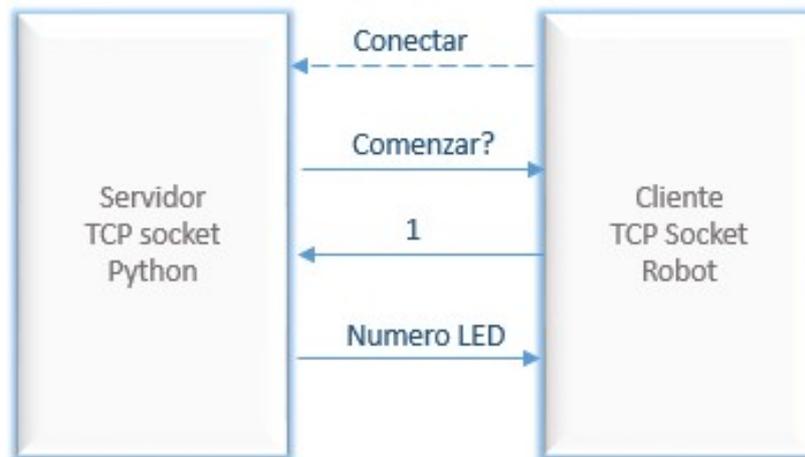


Figura 4.8.- Esquema de la comunicación del servidor TCP socket de Python.

4.5- Pantallas de explotación e interface con el usuario

4.5.1- Diseño de la pantalla de explotación

La pantalla de usuario se realiza en Python. ‘GUIcolor5.py’ es el archivo encargado de la definición de la clase llamada ‘Ui_MainWindow’, que define la pantalla de usuario creada para el trabajo. Este archivo al completo se encuentra en el Anexo 2. Esta pantalla de explotación está basada en PyQt5, que permite la implementación de basada en las librerías Qt para C++ en Python.[19]

Sin embargo, el archivo ‘GUIcolor5.py’ no ha sido creado directamente en Python sino basándose un archivo previo de extensión ‘.ui’ creado con ayuda del programa Qt Designer. Con este programa se definen las características de

aparición de la pantalla de usuario, así como algunas de las interacciones entre los objetos que la componen mediante signal y slots. Para la conversión del documento 'Prueba1.ui' a 'GUIcolor3.py' se utiliza la herramienta de desarrollo pyuic5.[20]

Tras la conversión a un programa de Python, se añadieron a la pantalla las adaptaciones necesarias para su comunicación correcta con el resto del programa y para realizar las interacciones con el usuario requeridas. Se añadieron signal y slots y librerías necesarias para las acciones realizadas en los slots. El resultado de la pantalla se puede ver en la Figura 4.9.

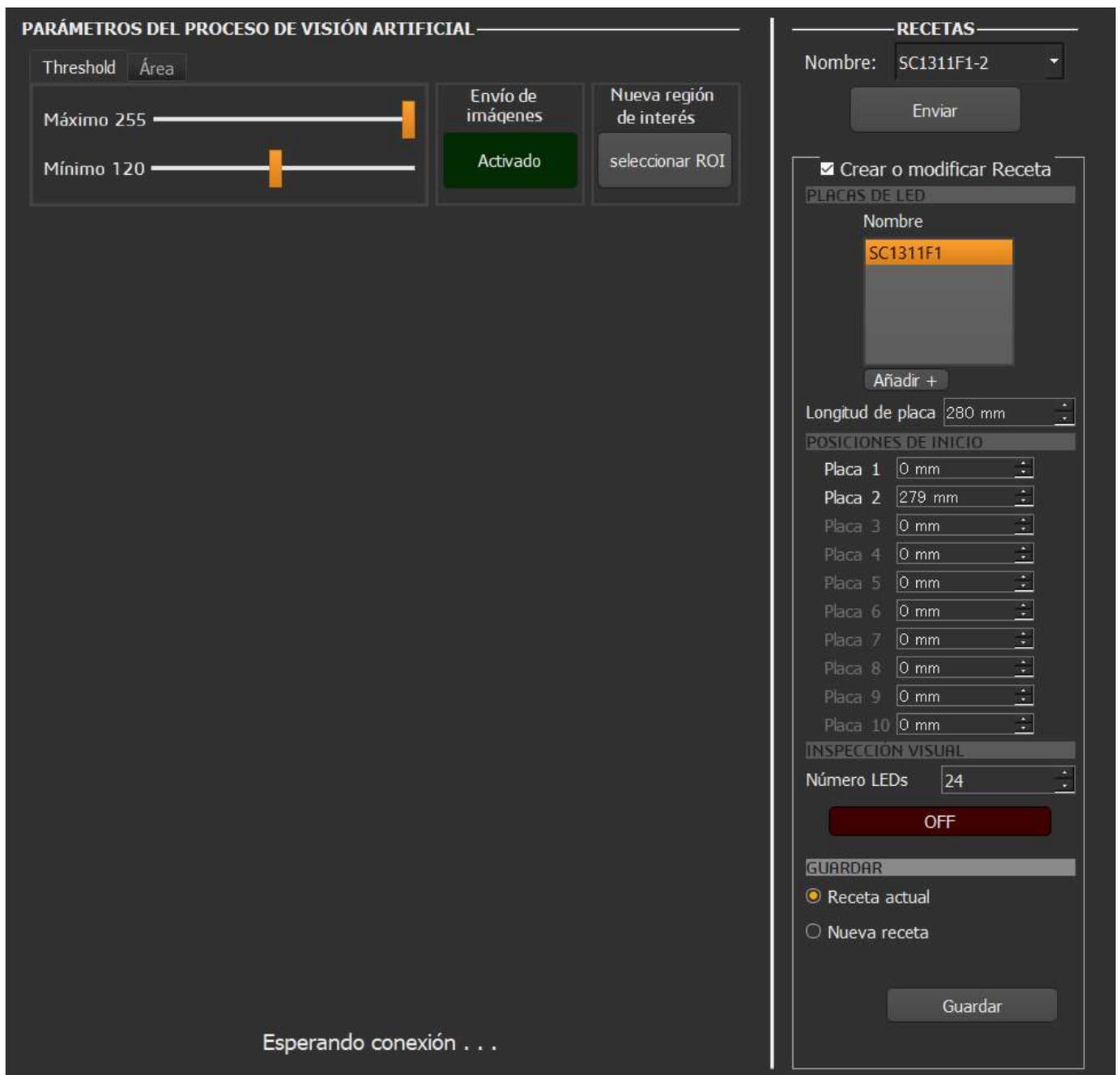


Figura 4.9.- Pantalla de explotación del trabajo.

5.4.2- Uso de la pantalla de explotación

Mediante la pantalla de explotación el sistema es capaz de interactuar con las personas para mostrarles información sobre lo que está ocurriendo en el proceso. Además, las personas son capaces de dar instrucciones e información al sistema. Para explicar las distintas aplicaciones de la pantalla se ha dividido en tres partes fundamentales. En la Figura 4.10 se pueden ver la pantalla de explotación del sistema. Las distintas partes que nos permiten utilizar distintas aplicaciones dentro del sistema, son las siguientes:

- Control del algoritmo de visión artificial

En la parte superior de la pantalla tenemos los pulsadores y barras deslizantes que nos permiten cambiar los parámetros del algoritmo de visión artificial. Las barras deslizantes permiten cambiar los valores de los *threshold* y áreas. Además, contamos con un pulsador para cambiar la región de interés, así como para cancelar el envío del resultado hasta tener el algoritmo completamente parametrizado.

- Envío de recetas

Desde la pantalla se pueden enviar recetas sobre la fabricación del sistema. Los elementos para realizar este envío de recetas están en la parte derecha de la pantalla. En caso de que ya haya recetas creadas, aparecerán en el menú desplegable, estas se pueden modificar con facilidad. Además, se pueden crear nuevas recetas que serán guardadas con las demás. Tras cerrar el programa se guardarán todas las recetas creadas en formato *.JSON* y volverán a aparecer en el menú desplegable una vez se vuelva a ejecutar el programa.

- Observación del resultado

En el centro de la pantalla nos aparecerán las imágenes tomadas por la cámara. Estas están tomadas muy oscuras, para poder distinguir con facilidad los LED encendidos. Es por esto, que lo que realmente vemos son los puntos de luz encendidos. En esta imagen, aparecerán rodeados en rojo los LED que hayan sido detectados por el algoritmo de visión artificial.

Además, aparecerá mensaje tanto en la esquina superior derecha, como en la parte inferior de la pantalla. Estos mensajes nos informan de cómo se encuentra el sistema o del resultado de la inspección visual y la fecha y hora en que fue realizada.

En la esquina superior derecha aparece el resultado de la inspección visual junto a la fecha y hora en que fue realizado. Además, aparecerá si se ha enviado o no el resultado al cliente. Por el otro lado, en la parte de debajo de la pantalla podremos ver en qué estado se encuentra el sistema. Normalmente el sistema está esperando conexión. Esto significa que el servidor *TCP socket* está abierto y listo para que se le conecte un cliente. Tras recibir la conexión y procesarla el sistema volverá a esperar la conexión del siguiente cliente.

Si existe algún problema y no se pueden encontrar las imágenes. Nos aparecerá un mensaje en el centro de la pantalla informándonos de esta situación, y se informará de que se han encontrado 0 LED encendidos ya que no había imágenes disponibles en la que procesarlos.

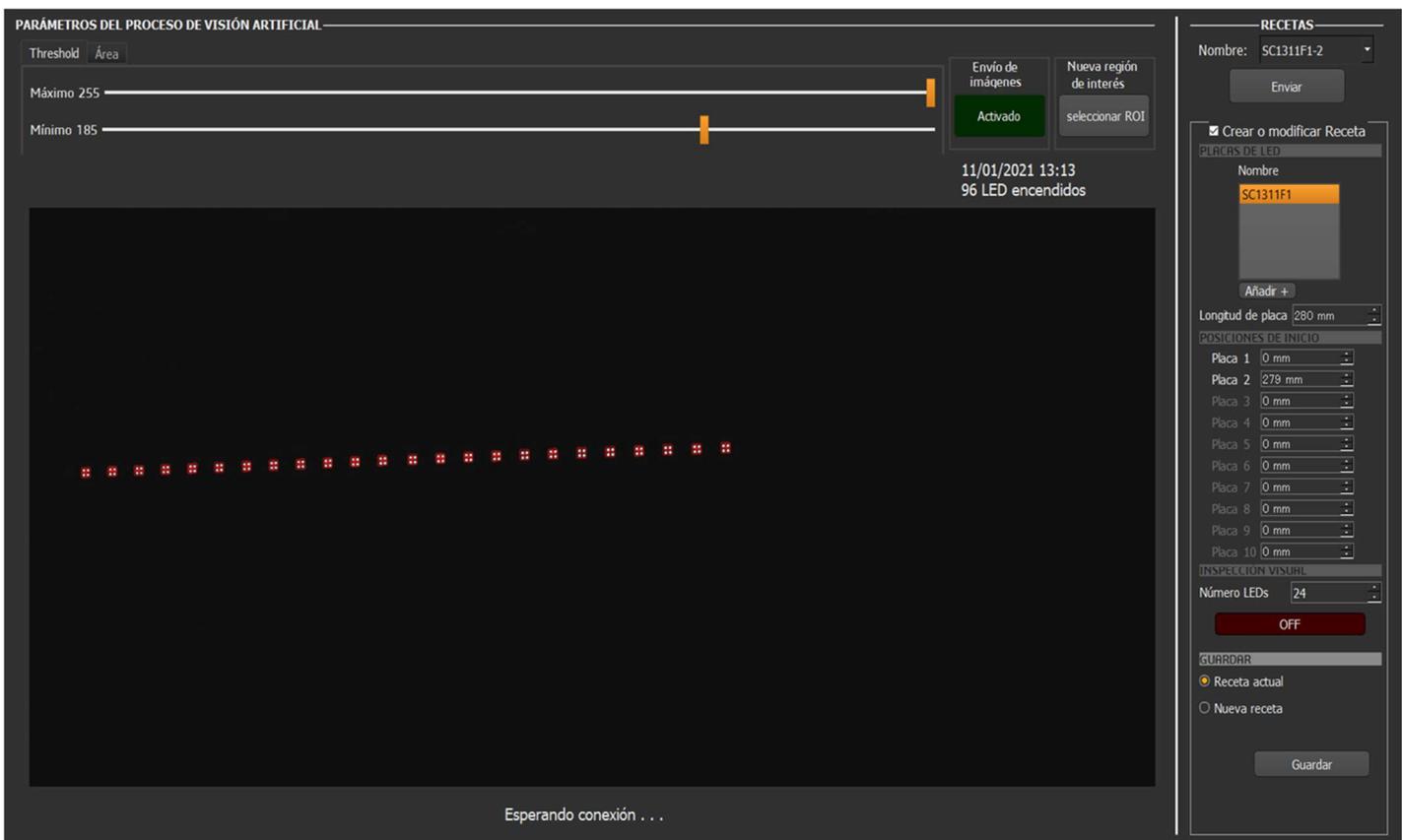


Figura 4.10.- Pantalla de explotación tras realizar el algoritmo de visión artificial.

4.6- Implementación de estrategias de seguridad

Los robots industriales requieren medidas de seguridad para evitar accidentes. Las medidas dependen de según el tamaño del robot, el entorno en que este trabajando, la interacción con personas, entre muchos otros factores. En la empresa NormaGrup Technology S.A ya se contaba con robots industriales trabajando en sus instalaciones. Es por tanto que se utilizaron los procedimientos ya conocidos por la empresa para asegurar el trabajo con el robot industrial.

Es muy importante el hecho de que en los alrededores del robot hay personas trabajando. Recordemos que el robot forma parte de una línea de producción en la que en los puestos anteriores son personas las que realiza en trabajo. Pero no solo esto, sino que además hay personas circulando constantemente al rededor del robot ya que está en medio de una instalación por la circulan personas. Teniendo en cuenta todo esto se implementaron medidas de protección y de prevención de accidentes. Además, el robot y controlador utilizado ya están diseñados para respetar las medidas de seguridad establecidas por las normas y reglamentos.

Como medida de protección contamos con las setas de emergencia. Estas son obligatorias en todo sistema automatizado.[21] Por normativa las setas de emergencia tienen que tener algunas características, como ser interruptores normalmente cerrado o tener un color rojo y forma distintiva. Además, deben estar en una posición de alcance de facilidad para el usuario. Se tienen setas de

emergencia en el FlexPendant y el controlador IRC5 como se puede ver en la Figura 4.11.



Figura 4.11.- Setas de emergencia del sistema robótico.

También como medida de protección se impide al usuario el acceso al área de trabajo del robot. Para ello se utilizan paneles de metacrilato resistentes a golpes. Sin embargo, existen áreas del entorno del robot en la que es necesario que el usuario pueda acceder, ya que lo necesita para el funcionamiento del sistema o para tareas de mantenimiento. Por el área de trabajo 1, el usuario ha de acceder al alimentador de tiras de LED, para rellenar o cambiar las tiras de LED. Así como, para acceder en caso de atascos. Por el otro lado, en el campo de trabajo 2 el usuario ha de colocar las bases de las lámparas en la posición adecuada para que acceda el robot. Recordemos que este campo de trabajo es manual y no cuenta con cinta transportadora. Estos lugares en los que no hay más remedio que permitir al usuario acceder al área de trabajo del robot fueron protegidos por medio de barreras de seguridad luminosas. Estas barreras detectan al usuario y paran completamente al robot, impidiendo que realice ningún movimiento. En la Figura 4.12 vemos el esquema de las barreras protectoras alrededor del robot.

También se limitan los movimientos de los ejes del robot. Esto significa que si el robot trata de ir a una posición que requiera que alguno de sus ejes gire más del límite establecido, dará un error y se parará. Con esto se evitan colisiones con las paredes, y que el robot salga del área de trabajo utilizada.

Adicionalmente a todo lo anterior, se implementan medidas para reducir la probabilidad de accidentes. Los usuarios que manejan el robot son usuario con formación y conocimientos para el manejo y control de robots industriales. Los cuales han recibido además una formación de seguridad para saber cómo actuar en caso de accidentes, y para saber las medidas a tomar para reducir la probabilidad de que ocurran.

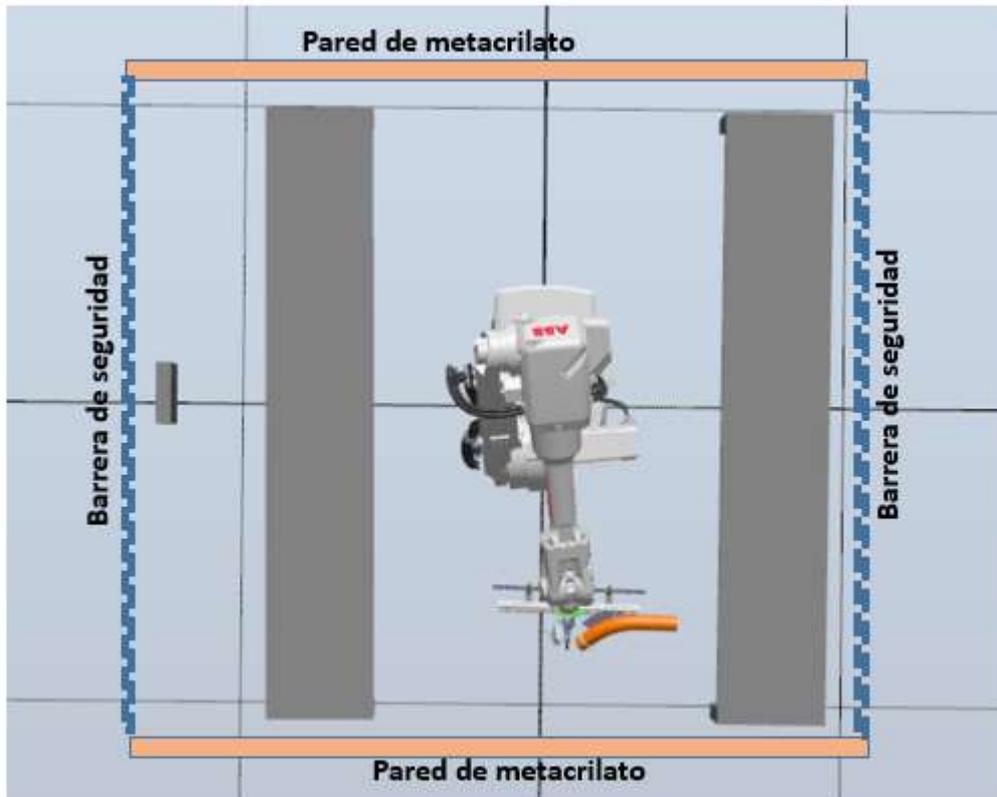


Figura 4.12.- Esquema de las protecciones al rededor del robot.

4- Conclusiones y futuros trabajos

El sistema realizado cumple los objetivos principales marcados por la empresa NormaGrup Technology SA. Se han conseguido resultados satisfactorios para la fabricación de lámparas en serie, utilizando distintos tipos de tiras de LED. El sistema se adapta a las necesidades de usuario siendo fácil su uso y la detección de fallos en el mismo.

Sin embargo, el sistema es algo lento. Es por esto que se trataron de buscar mejoras de cara a realizar las acciones de colocar, colocar, soldar y verificar las tiras de LED. La parte del proceso más lenta es la soldadura. Para mejorar esta velocidad se plantea utilizar placas pre soldadas. Esto significa que los puntos donde el soldador automático vaya a realizar la soldadura traigan previamente una capa de estaño. Con esto se reduciría el tiempo de pre-calentamiento e incluso se prevé una mayor eficiencia en el proceso de soldadura, reduciéndose los errores en el sistema automático de soldado.

Por otra parte, la otra línea de mejora del sistema va de cara a no necesitar el uso de las recetas tan repetidamente. El cambio de unas tiras de LED a otras siempre va a requerir el uso de recetas, esto se considera aceptable ya que cambiarlas lleva consigo también otras tareas. Como cambiar las tiras del alimentador. Sin embargo, cambiar la longitud de la lámpara a fabricar podría realizar en continuo sin parar la línea simplemente cambiando el tamaño de la base de lámpara. Para que esto se pueda hacer se prevé utilizar un sistema de visión artificial, aprovechando la cámara ya existente en el sistema y el programa de Python ya realizado. Este nuevo sistema tomaría fotografías y calcularía el tamaño de la base de la lámpara y cuantas tiras de LED se debe utilizar.

5- Bibliografía

- [1] “Home - Normagrup.” [Online]. Available: <https://www.normagrup.com/>. [Accessed: 14-Jan-2021].
- [2] “ Qué es la Automatización Industrial, cómo funciona y ejemplos.” [Online]. Available: <https://revistaderobots.com/industria/automatizacion-industrial/>. [Accessed: 13-Jan-2021].
- [3] “Automatización y robótica industrial.” [Online]. Available: <https://blog.infaimon.com/automatizacion-y-robotica-industrial/>. [Accessed: 13-Jan-2021].
- [4] “IRB 2600 - Robots industriales | ABB.” [Online]. Available: <https://new.abb.com/products/robotics/es/robots-industriales/irb-2600>. [Accessed: 03-Jan-2021].
- [5] “IRC5 - Controladores | ABB.” [Online]. Available: <https://new.abb.com/products/robotics/es/controladores/irc5>. [Accessed: 03-Jan-2021].
- [6] “ABB Robótica.” [Online]. Available: <https://new.abb.com/products/robotics/es/>. [Accessed: 03-Jan-2021].
- [7] “USP5 | japanunix.” [Online]. Available: <https://www.japanunix.com/es/products/automation/unit/usp5/>. [Accessed: 03-Jan-2021].
- [8] “FA-1000BC | TAIYO ELECTRIC IND.CO.,LTD.” [Online]. Available: <http://www.goot.jp/en/option/fa-1000bc/>. [Accessed: 03-Jan-2021].
- [9] “La ventosa de espuma manipula con facilidad los objetos difíciles – Piab.” [Online]. Available: <https://www.piab.com/es-ES/news/press-releases/foam-suction-cup-handles-difficult-objects-with-ease/>. [Accessed: 07-Jan-2021].
- [10] “SMC CUJB6-15D cyl, free mt, dbl acting, CUJ COMPACT MINI FREE-MOUNT***.” [Online]. Available: <https://www.smc-pneumatics.com/CUJB6-15D.html>. [Accessed: 07-Jan-2021].
- [11] “Genie Nano-1GigE | Teledyne DALSA.” [Online]. Available: <https://www.teledynedalsa.com/en/products/imaging/cameras/genie-nano-1gige/>. [Accessed: 03-Jan-2021].
- [12] “JSON.” [Online]. Available: <https://json.org/json-es.html>. [Accessed: 03-Jan-2021].
- [13] O. Déniz, S. Alexis, Q. Arencibia, and F. J. Santana Pérez, “Sistemas Operativos: Programación de Sistemas Curso 2006-07.”
- [14] “Redes 3: 2.1 Puertos / Sockets TCP/IP.” [Online]. Available: <http://hindel-redes3.blogspot.com/2017/04/21-puertos-sockets-tcpip.html>. [Accessed: 11-Jan-2021].
- [15] “¿Qué es Ethernet?” [Online]. Available: <https://www.linksys.com/es/r/resource-center/que-es-ethernet/>. [Accessed: 11-Jan-2021].
- [16] “configparser — Parser para archivos de configuración — documentación de Python - 3.10.0a4.” [Online]. Available: <https://docs.python.org/es/3.10/library/configparser.html>. [Accessed: 14-Jan-2021].
- [17] “Blob Detection Using OpenCV (Python, C++) | Learn OpenCV.” [Online]. Available: <https://www.learnopencv.com/blob-detection-using-opencv-python->

- c/. [Accessed: 10-Jan-2021].
- [18] “Home - OpenCV.” [Online]. Available: <https://opencv.org/>. [Accessed: 11-Jan-2021].
- [19] “PyQt5 · PyPI.” [Online]. Available: <https://pypi.org/project/PyQt5/>. [Accessed: 14-Jan-2021].
- [20] “pyuic5-tool · PyPI.” [Online]. Available: <https://pypi.org/project/pyuic5-tool/>. [Accessed: 14-Jan-2021].
- [21] A. Robotics, “Manual del producto - IRC5,” 2004.
- [22] “IRB 2600, M2004, Especificaciones del producto.” [Online]. Available: <https://search.abb.com/library/Download.aspx?DocumentID=3HAC035959-005&LanguageCode=es&DocumentPartId=&Action=Launch>. [Accessed: 13-Jan-2021].