



Puesta en funcionamiento de visión integrada IRC5C con COGNEX 7000



Alumno: Borja Pascual Corchero.
Instituto: IES Javier Garcia Tellez.

17 de junio del 2022

1. ÍNDICE.

1. ÍNDICE.	1
2. INTRODUCCIÓN-JUSTIFICACIÓN.	2
3. OBJETIVOS.	3
3.1. Verificar opción de visión integrada en IRC5.	3
3.2. Puesta en marcha visión integrada con cámara COGNEX 7000.	3
3.3. Calibrar cámara y creación de trabajo.	3
3.4. Calibrar herramienta del robot.	3
3.5. Colocar cinta transportadora y captar robttargets.	3
3.6. Realizar programa en Rapid.	3
3.7. Verificar el correcto funcionamiento del programa.	3
3.8. Mejorar y optimizar el programa.	3
4. PROCESO.	4
4.1. Planteamiento del proyecto.	4
4.2. Instalación de hardware y software.	7
4.2.1. Verificación Visión integrada.	7
4.2.3. Colocación de la cinta transportadora.	13
4.2.4. Configuración IP y nombre de la Cámara.	14
4.2.5. Calibración de la cámara.	16
4.2.6. Calibración de la herramienta.	19
4.2.7. Captación de Robtarget.	20
4.3. Programación.	22
4.3.1. Programación trabajo de la cámara.	22
4.3.2 Programación en RAPID.	27
4.4. Verificación del correcto funcionamiento y ajustes necesarios.	32
4.5. Optimización del programa.	34
4.6. Funcionamiento final.	35
5. CONCLUSIONES	37
6. Planos y Presupuesto	38
6.1. Planos.	38
6.2. Presupuesto.	39
7. ANEXOS	41
7.1. Material utilizado.	41
7.2. Índice Figuras / Planos / Presupuesto.	41
7.3. Código RAPID.	42

2. INTRODUCCIÓN-JUSTIFICACIÓN.

El proyecto está planteado para adquirir el conocimiento y experiencia usando la visión integrada en una controladora *IRC5* de *ABB*, con la ayuda de la cámara *COGNEX 7000*, que es una cámara pequeñita de una gama Alta/Media que para el proyecto que vamos a realizar nos viene genial puesto que por su tamaño nos facilitara a la hora de colocarla en su posición de trabajo. La controladora *IRC5* es bastante sencilla de utilizar y versátil para ser una de las más pequeñas de la gama que *ABB* proporciona a su vez pasa con el *ROBOT IRB120* que también es de los más pequeños que *ABB* tiene en catálogo pero para nuestro proyecto tendremos suficiente.

Siempre he querido formarme en esta clase de proyectos relacionados con robots industriales, coger experiencia con los que tenemos en clase para utilizarlos y programarlos lo mejor posible y en este caso al tener que utilizar visión integrada lo hace más interesante para entender el comportamiento y la obtención de datos que el robot precisa para realizar las tareas deseadas.

En el futuro cercano me gustaría seguir estudiando en base a robots industriales y acabar trabajando con ellos y creo que la elección de este proyecto es la mejor que podía realizar ya que me parece un buen punto de partida para entender el funcionamiento de estas poderosas máquinas de la industria, comprender su alta capacidad de realizar cualquier tarea que se necesite hacer en la industria, como por ejemplo tareas en la industria del automóvil como puede ser pintar dichos coches, montar su interior, incluso soldar las partes del chasis.

Sobre el proyecto a realizar me resulta muy interesante el plantear dicha clasificación de piezas con la visión integrada para aprender a obtener los datos que dicha cámara proporciona y explotar al máximo sus posibilidades.

3. OBJETIVOS.

3.1. Verificar opción de visión integrada en IRC5.

Ir a las opciones de la controladora para verificar que la opción de visión integrada está instalada.

3.2. Puesta en marcha visión integrada con cámara COGNEX 7000.

Configurar la cámara en la controladora asignando una IP y un nombre a dicha cámara.

3.3. Calibrar cámara y creación de trabajo.

Ya configurada la cámara pasaremos a realizar la calibración y la creación del trabajo de la cámara, donde se encontrará los filtros de las imágenes y la captación de los datos.

3.4. Calibrar herramienta del robot.

Usaremos una ventosa para captar las piezas y tendremos que realizar la calibración de la misma.

3.5. Colocar cinta transportadora y captar robttargets.

Colocar la cinta transportadora en la posición deseada para la prueba y realizar captura de los puntos donde se depositarán las piezas, así como el punto de Home y para mover el robot hasta la posición de capturar las piezas en la cinta.

3.6. Realizar programa en Rapid.

Programar la controladora con el código en *Rapid*.

3.7. Verificar el correcto funcionamiento del programa.

Una vez acabado el programa, verificar que no tiene errores y que tanto la cámara como el robot hacen correctamente lo que tienen que hacer para que el trabajo se cumpla en su totalidad.

3.8. Mejorar y optimizar el programa.

Optimizar en lo máximo posible el programa para que lo realice de la manera más rápida y segura posible.

4. PROCESO.

4.1. Planteamiento del proyecto.

Para este proyecto tenemos el objetivo de entender mejor las funciones de la visión integrada que tiene nuestro controlador y para ello vamos a exprimir estas funciones al máximo posible.

Tenemos una cámara COGNEX 7000 (Figura 1) monocromo la cual es bastante versátil y que podemos poner en cualquier posición, puesto que el objetivo que estamos utilizando tiene mucho zoom y no podemos modificarlo tenemos que buscar una altura mínima para hacer una buena zona de captura de imágenes.



Figura 1. Cámara COGNEX 7000.

Una vez tengamos estudiada la altura necesaria para la cámara, buscaremos una posición óptima para nuestro hardware que en este caso es una cinta transportadora que funciona de forma manual mediante un panel de mando en uno de sus laterales (Figura 2).



Figura 2. Panel de control cinta transportadora.

Para la posición de la cinta transportadora podríamos poner que venga en el eje -X del robot o en el eje Y, pero sabiendo donde va colocada la cámara para este proyecto nos viene mejor posicionar la cinta transportadora al lateral derecho del robot para que las piezas lleguen desde el eje -X. El robot articulado *IRB120* (Figura 3) es muy pequeño y por eso tiene poco alcance de maniobra, es por eso que necesitamos buscar la forma de posicionar la cinta transportadora lo más cerca posible para que el robot tenga alcance a todo el ancho de la cinta, es por eso que la solución fue situar esta cinta sobre unos soportes para aumentar su altura y poder aproximarse hacia el eje del robot (Figura 4).



Figura 3. Robot IRB120_3_58_G_01.

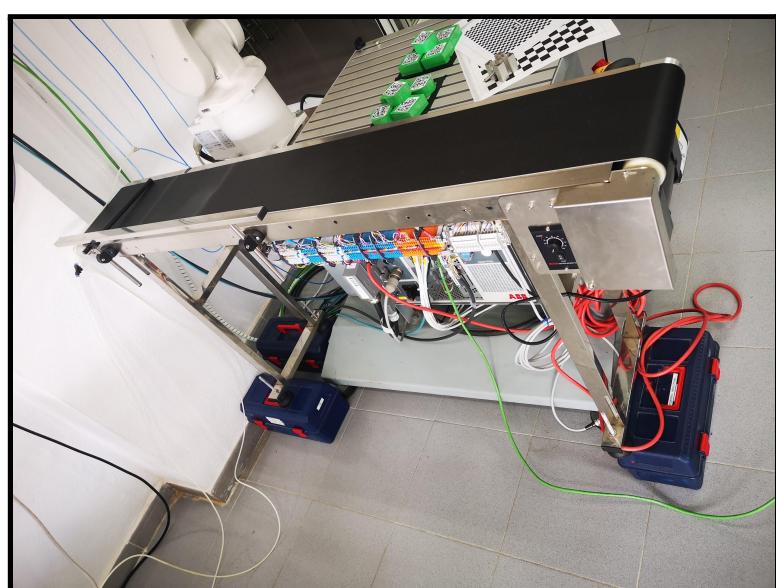


Figura 4. Posición cinta transportadora.

Colocado ya el hardware de nuestro proyecto avanzamos hacia el planteamiento de las funciones que le podemos utilizar a la cámara COGNEX 7000 con la visión integrada de nuestra controladora, esta visión integrada nos deja realizar muchas funciones como la detección de piezas según su forma, decirnos la posición de estas piezas a partir del punto de coordenadas de la cámara, leer números, letras, símbolos, códigos de barras, códigos QR, etc... La propia visión integrada puede también detectar por colores pero la cámara que utilizamos es monocromo así que esa función no podemos utilizarla.

Observando nuestras capacidades llegamos a la conclusión de que lo mejor es hacer una detección de piezas por forma, las cuales serán cuadradas y redondas (Figura 5) y llevarán un QR pegado en la parte superior para definir un color a estas piezas (Figura 5).



Figura 5. Piezas utilizadas para el proyecto.

4.2. Instalación de hardware y software.

4.2.1. Verificación Visión integrada.

Con el planteamiento finalizado comienza la instalación de lo que necesitamos para realizarlo.

Empezamos encendiendo nuestra controladora ya que debemos mirar si tiene la opción de visión integrada que necesitamos para el proyecto. Una vez arrancada nos conectaremos vía red a la controladora por el puerto de servicio, ejecutaremos en nuestro equipo informático el programa *RobotStudio* y crearemos

una estación vacía sin controladora virtual ya que trabajaremos directamente con la física, en el proyecto nos dirigimos al apartado de controladora y en la parte superior izquierda veremos la opción de “*añadir controlador*”, pinchamos en esta opción y nos saldrá un desplegable (Figura 6) donde nos mostrara varias opciones y deberemos darle a la primera opción que pone “*conexión con un clic...*” (Figura 6), esta opción nos conectará directamente con la controladora, posiblemente nos pida usuario y contraseña (Figura 7), le daremos a la opción de conectarse como usuario predeterminado.

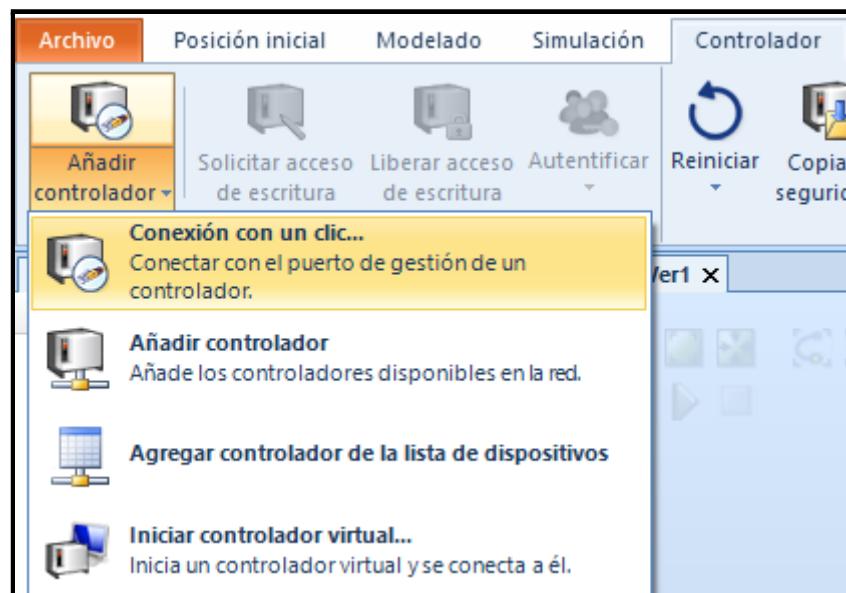


Figura 6. Panel desplegable añadir controladora.

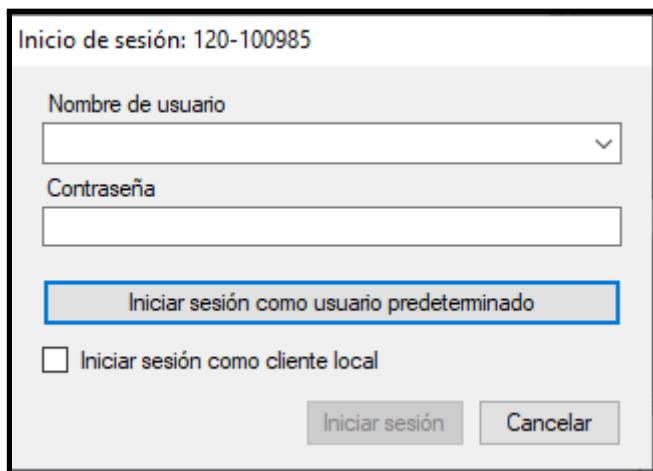


Figura 7. Ventana usuario y contraseña controladora.

Ya estaremos conectados a la controladora y tendremos que verificar que tenga la opción de visión integrada instalada, para ello tenemos que dar clic derecho sobre el nombre de la controladora y nos desplegará un menú, en este menú saldrá la opción de “*Cambiar opciones...*” (Figura 8) le damos clic a esta pestaña.

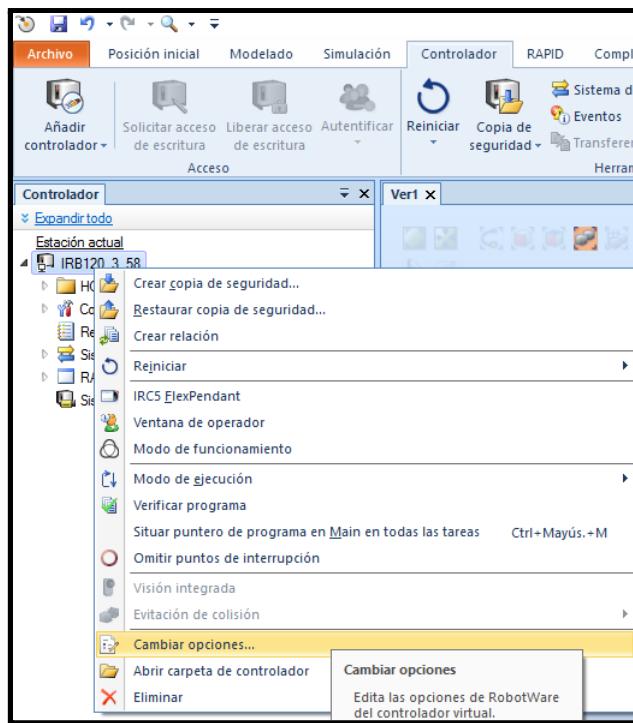


Figura 8. Menú desplegable controladora.

Nos abrirá una ventana donde tendremos todas las opciones posibles para la controladora (Figura 9) y tendremos que buscar la opción llamada “1341-1/1520-1 *Integrated Vision Interface*” en el apartado de “System Options” (Figura 10).

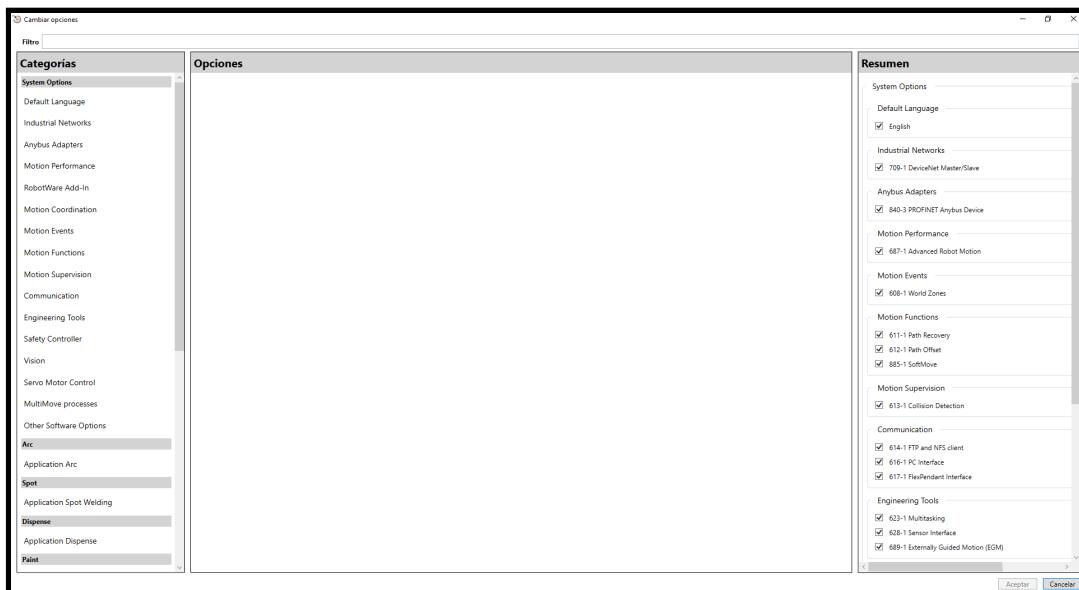


Figura 9. Ventana opciones controladora.

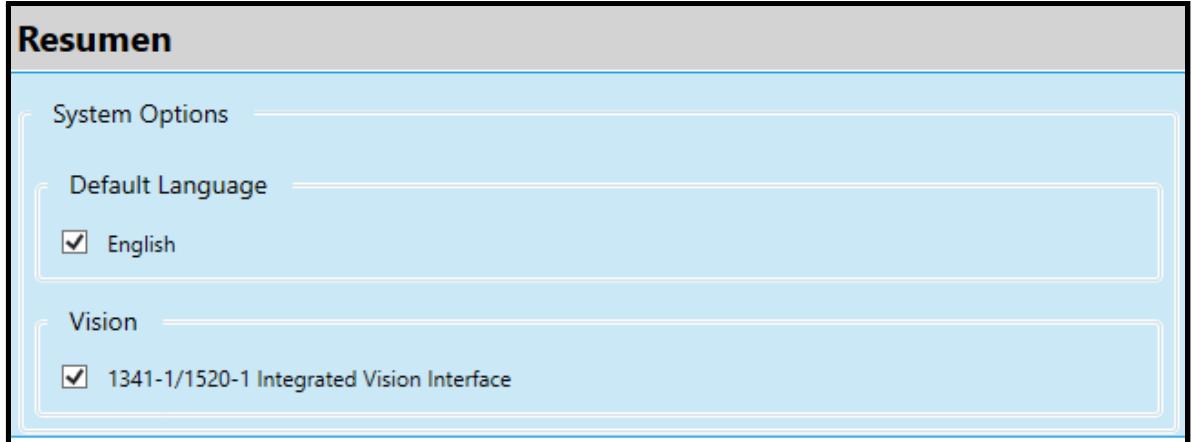


Figura 10. Opción visión integrada controladora.

4.2.2. Colocación de la cámara.

Con la verificación de que tenemos la opción de visión integrada instalada pasamos a la parte de colocar la cámara en su posición final donde estará ubicada para realizar las capturas de imagen de la cinta transportadora. Como comentamos en el planteamiento, necesitamos una gran altura de la cámara para que esta tenga buena visión de la cinta transportadora y sea capaz de captar todo su ancho para la detención de piezas.

La cámara irá situada en un soporte en forma de L invertida que irá acoplado a la pared él cual nos dará algo de maniobra en el eje vertical como horizontal (Figura 11), la cámara a su vez irá acoplada en un soporte impreso en 3D (Figura 12 y 13) que es el que nos facilita que la cámara se mueva de forma horizontal.



Figura 11. Soporte de pared para la cámara.

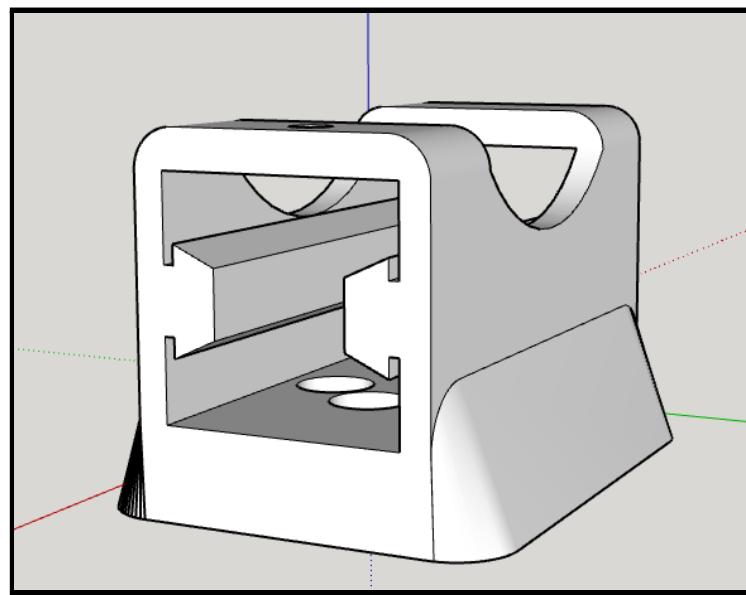


Figura 12. Soporte de la cámara.

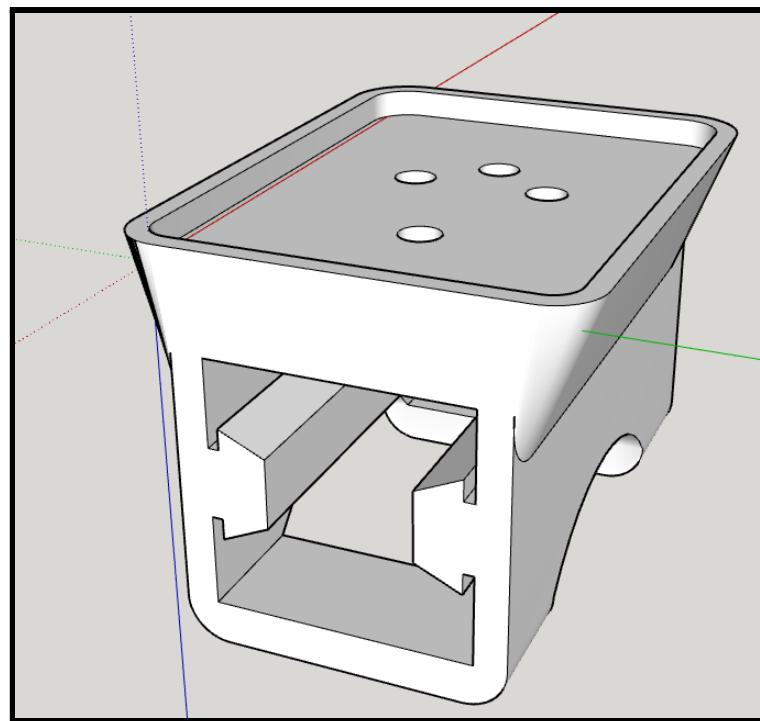


Figura 13. Soporte de la cámara.

Una vez colocada la cámara en su posición pasaremos los cables de forma segura desde la cámara hasta la controladora sin dejarlos muy tensos para que tengan juego por si hay que hacer cambios en la posición de la cámara (Figura 14 y 15).

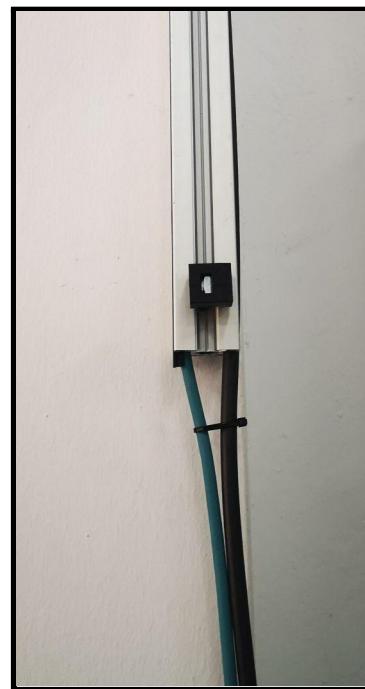


Figura 14. Cables desde la cámara hasta el controlador.

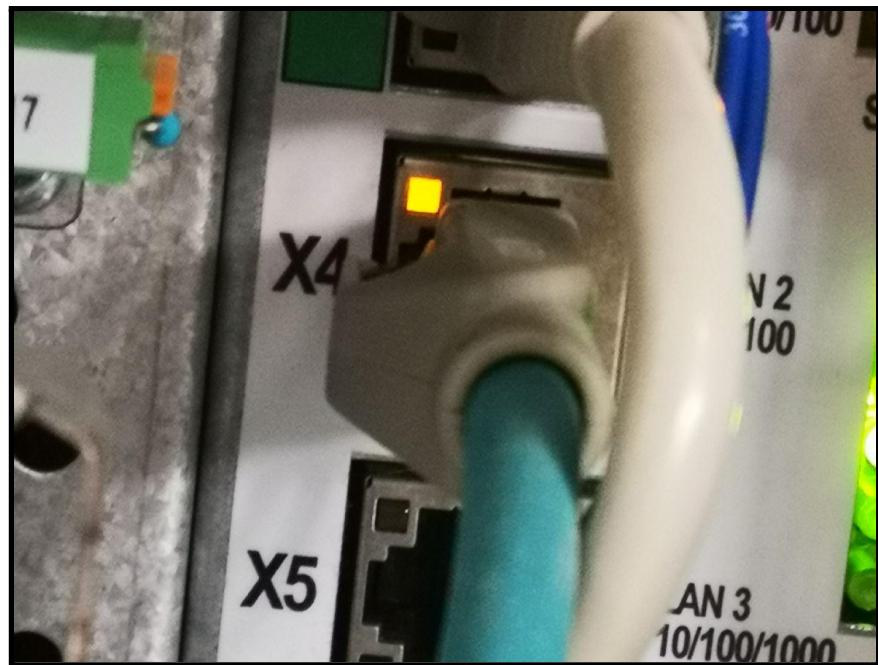


Figura 15. Conexionado de la cámara a la controladora.

4.2.3. Colocación de la cinta transportadora.

Con la cámara ya en su posición podemos situar la cinta transportadora en su lugar, para ello como ya comentamos en el planteamiento, necesitamos que esté lo más próximo posible al eje del robot para que este pueda llegar a lo ancho de la cinta y la cámara pueda ver también esa zona de la cinta. Para ello situaremos la cinta sobre unos soportes estables para aumentar su altura y no choque con la estructura del robot (Figura 16), con la cinta en el soporte, usaremos un folio para visualizar el ancho de la cinta en la cámara.



Figura 16. Soportes de la cinta transportadora.

4.2.4. Configuración IP y nombre de la Cámara.

Para poder observar el estado de la cámara nos tendremos que conectar a la controladora como hicimos anteriormente para observar sus opciones. Una vez conectados a la controladora nos saldrán en el apartado de estación la opción “*Sistema de visión*” (Figura 17), le daremos clic derecho y pinchamos en la única opción que nos sale que es “*Visión integrada*”, esta opción nos abrirá una nueva pestaña que antes no era visible llamada “*Visión*” en la parte superior (Figura 18).

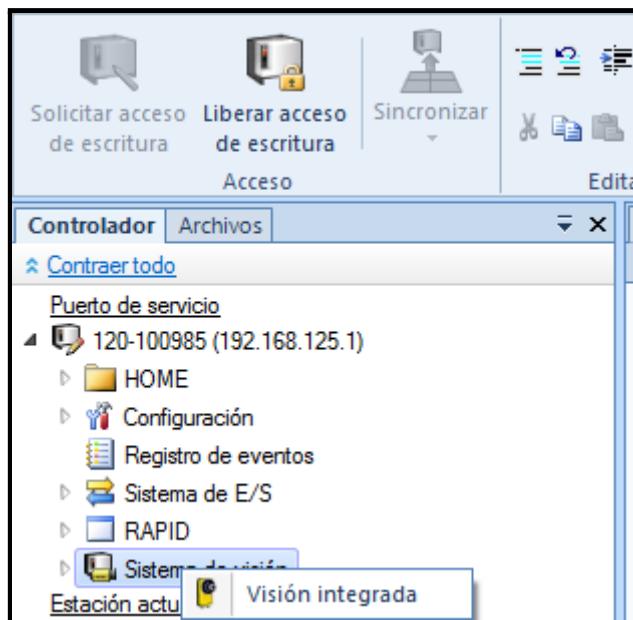


Figura 17. Opción de sistema de visión en estación.

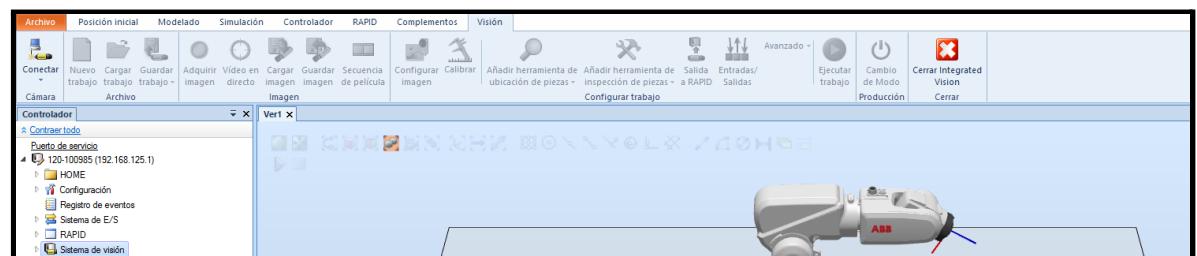


Figura 18. Ventana Visión en RobotStudio.

Una vez estemos en esta nueva ventana, de la misma forma que entramos en esta ventana dando clic derecho en Sistema de visión, le volveremos a dar y ahora nos saldrán mas opciones y si el conexionado de la cámara está bien realizado nos saldrá la MAC de la cámara, tendremos que configurar clicando en la cámara para ponerle un nombre de red ya que la IP siempre se la pondrá la controladora aunque podemos ponerle una manualmente. Para nuestro caso llamaremos a la cámara “VISION_TELLEZ” (Figura 19) y la IP dejaremos que se la ponga la controladora.

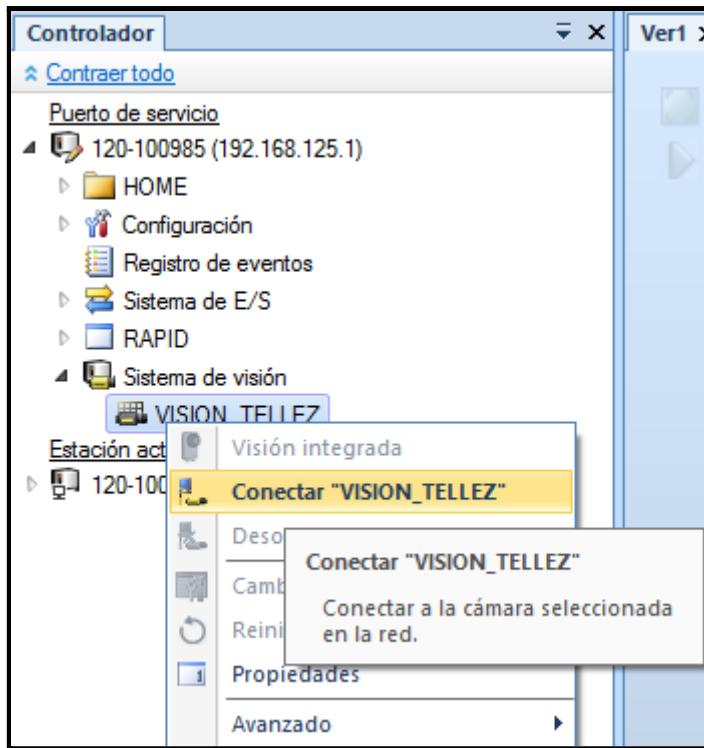


Figura 19. Cámara configurada en RobotStudio.

Ya configurada la cámara nos podremos conectar con ella, al hacerlo se nos mostrará una imagen captada por la cámara si todo está correcto, en la parte superior de la ventana tendremos diversas opciones para configurar y calibrar la cámara a nuestro gusto (Figura 20).

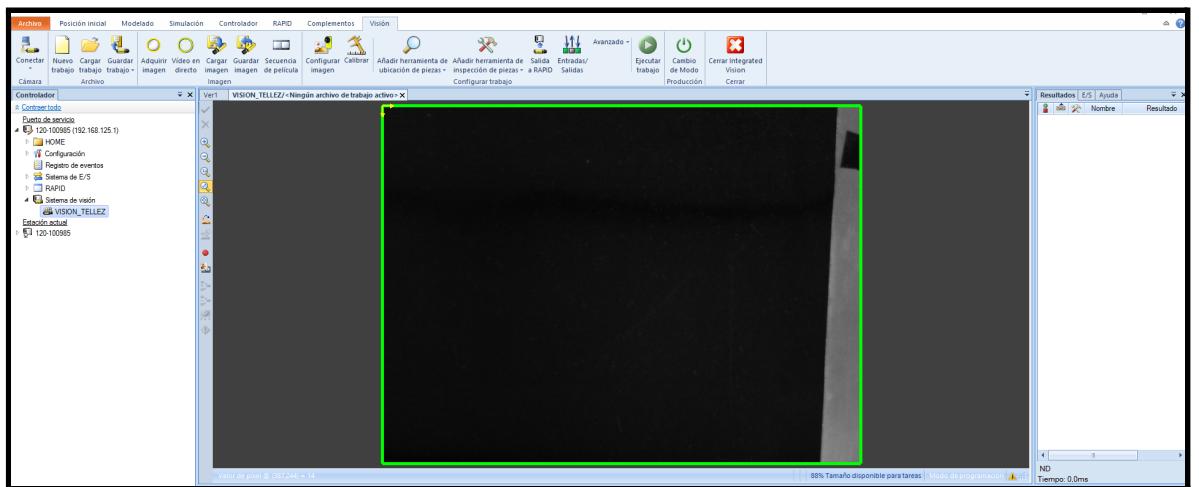


Figura 20. Opciones de la ventana Visión.

Comentar algunos datos importantes sobre estas opciones de la ventana (Figura 20), En la parte izquierda tenemos la opción de “Conectar” el cual es el que usaremos cada vez que nos queramos conectar a la cámara ya que ya la tenemos definida en la red con IP y nombre, a su derecha encontramos el apartado de “Archivo” donde situaremos las opciones del trabajo, el trabajo para entender su funcionamiento es lo que la cámara va a realizar cada vez que nosotros

ejecutemos dicho trabajo, resumiendo es la programación de la cámara. Pasamos al apartado de “Imagen” en el encontramos opciones como la de “Adquirir imagen” y “Vídeo en directo” estas opciones nos permitirán capturar imágenes o visualizar en directo lo que está viendo en ese momento la cámara, esto facilita su calibración y colocación, también encontramos opciones de cargar y guardar imágenes las cuales como su nombre indica sirven para guardarnos imágenes ya realizadas de la cámara o cargar imágenes ya guardadas, no es estrictamente necesario que sean solo imágenes de la cámara pero para un mejor calibrado se recomienda. Continuamos con el apartado de “Configurar trabajo” en este apartado esta lo mas importante de la configuración de la visión integrada, en el encontramos la configuración de la imagen que sirve para editar la imagen de la cámara con diversas opciones, el apartado de calibrado que es el que usaremos más adelante para calibrar nuestra cámara, tiene varias formas de calibrado (Figura 21) en nuestro caso utilizaremos la forma “GRID”. Los siguientes apartados se utilizarán para definir regiones de búsqueda y especificar según las herramientas que te proporciona la visión integrada lo que debe buscar, las salidas a RAPID se usará para definir dónde irán los datos recogidos con las herramientas de la visión integrada para exportar los datos al código RAPID. Para terminar encontramos 3 opciones, “Ejecutar trabajo” se usará para simular la programación implementada a la cámara con las herramientas que utilicemos, la opción “Cambio de modo” se usará para cambiar la cámara en modo “Producción” que este modo no nos dejará configurar nada de la cámara ya que estará activa realizando el trabajo cargado cada vez que se solicite y el modo “Programación” el cual se utilizará para configurar y programar la cámara. Para terminar tenemos el botón de cerrar la ventana de visión.

4.2.5. Calibración de la cámara.

Ahora pasaremos a la calibración de la cámara así como la creación del trabajo que realizará, en la ventana de “Visión” y ya conectados a la cámara, crearemos un nuevo trabajo, con este empezaremos a trabajar de cero, lo primero que tenemos que hacer es el calibrado de la herramienta y para ello clicamos en la opción de “Calibrar” y nos saldrá en la parte inferior una ventana donde encontraremos un desplegable que al abrirlo saldrán las opciones de la Figura 21.

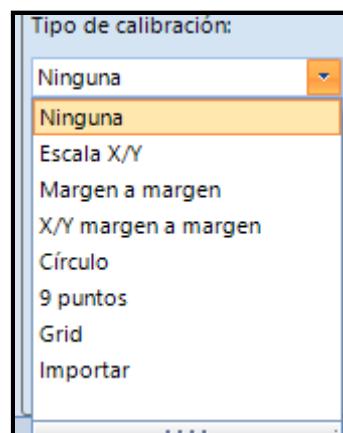


Figura 21. Opciones de calibrado.

Le daremos a la opción de “Grid” se nos abrirá en la misma ventana una nueva pantalla con opciones de calibrado, esto lo tendremos que configurar a nuestra necesidad, en nuestro caso lo pondremos como se muestra en la Figura 22, y le daremos a “Print grid...” esto nos creará un archivo PDF con la cuadrícula para imprimir (Figura 23).

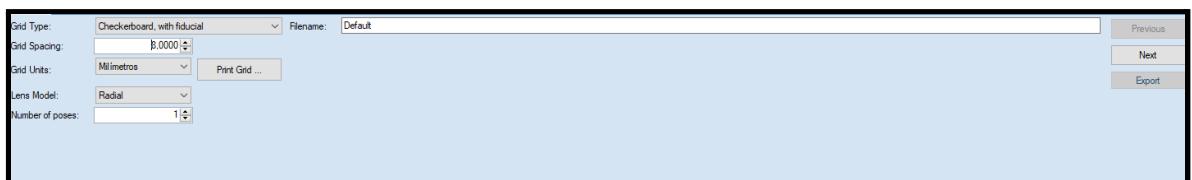


Figura 22. Opciones de plantilla del calibrado en Grid.

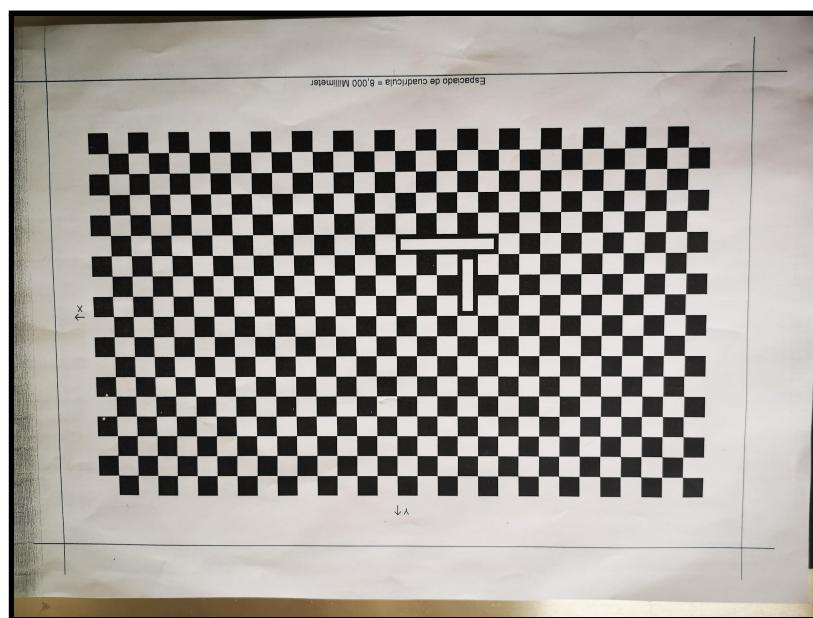


Figura 23. Plantilla grid impresa.

Una vez tengamos la plantilla impresa la colocaremos en la cinta en la posición donde la cámara capta las imágenes y clicamos “Next”, nos pasara a otro menú donde podremos ver los puntos de calibrado, esta ventana a su vez nos ayudará a colocar la plantilla lo mejor posible bajo la cámara, le daremos a “Acquire image” para coger una imagen nueva y ver la posición de la plantilla (Figura 24).

Un dato importante a tener en cuenta es que el punto de coordenadas de la cámara se creará en base a las marcas que están en la plantilla (Figura 25), por ello tenemos que tener en cuenta dónde queremos crear el punto de la cámara y en qué dirección antes de ejecutar el calibrado final.

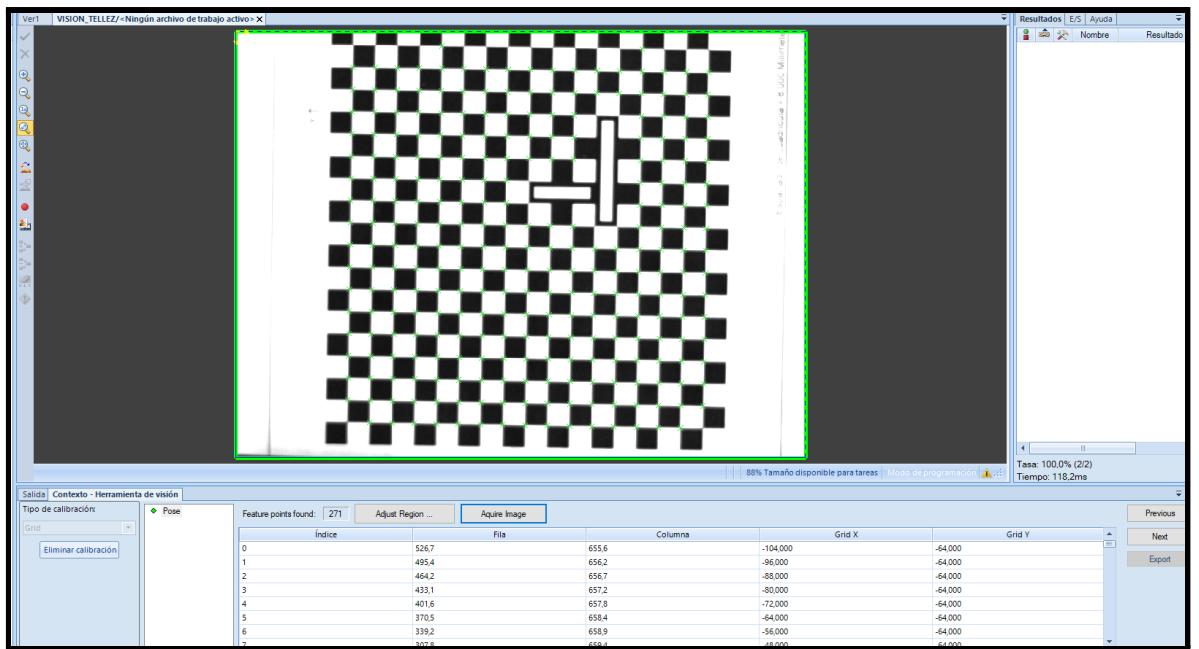


Figura 24. Ajuste previo al calibrado de la cámara.

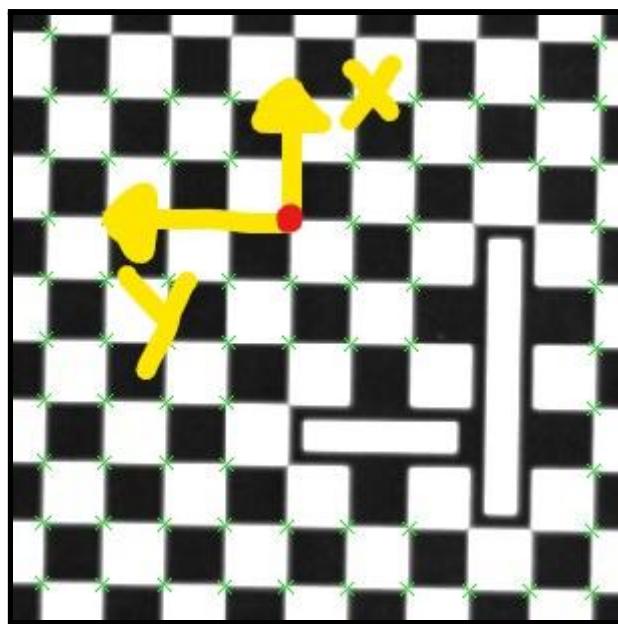


Figura 25. Eje de coordenadas según plantilla.

Cuando hayamos colocado en la posición deseada nuestra cuadrícula, pasaremos al siguiente paso que es verificar que el calibrado sea aceptable, para ellos le daremos a “Next” y nos mostrará una barra de puntuación donde lo más próximo a 0 será un buen resultado y cercano al 5 será un mal resultado, en la Figura 26 se puede observar que el resultado fue de excelente y daremos por finalizado el calibrado clicando en “Finish”. Guardaremos esta calibración en el trabajo dando a “Guardar trabajo” y le pondremos un nombre que nos pedirá en un futuro en RAPID, con esto damos por terminada la calibración de la cámara.

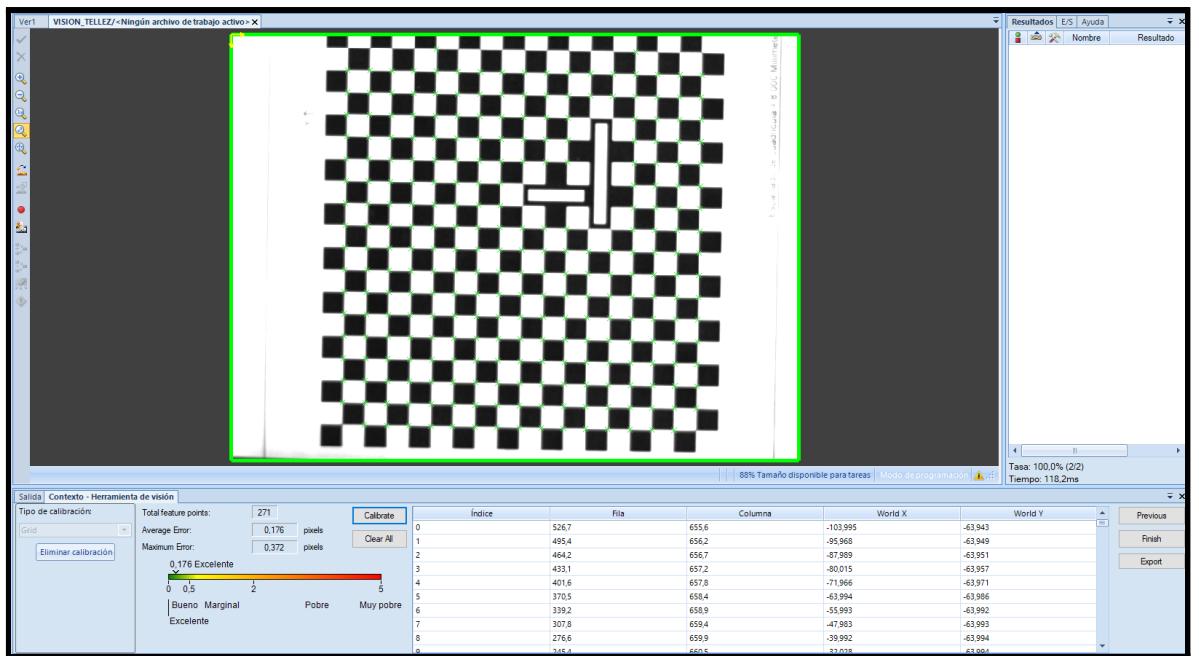


Figura 26. Resultado de calidad del calibrado.

4.2.6. Calibración de la herramienta.

A continuación pasaremos a la calibración de la herramienta que llevará el robot, en nuestro caso usaremos una herramienta con 4 ventosas (Figura 27) de las cuales solo usaremos una, es por eso que a la hora de calibrar la herramienta solo tendremos en cuenta una de estas ventosas y no las otras 4.

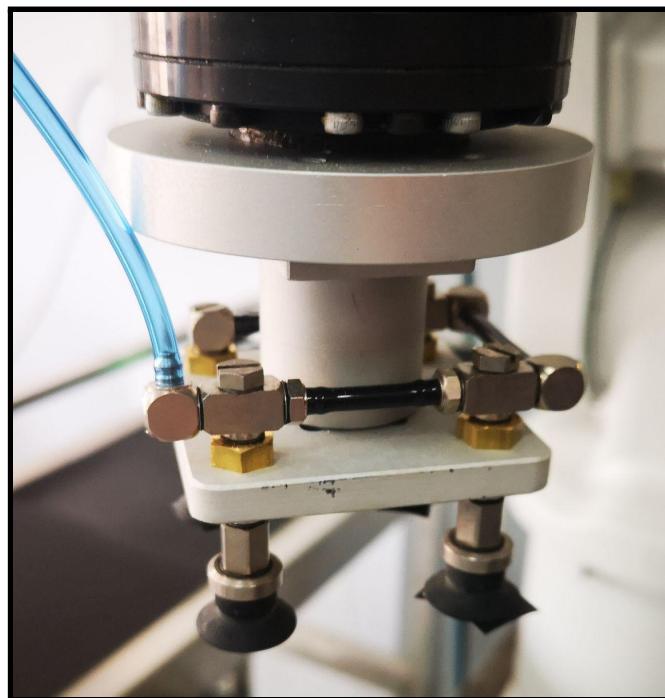


Figura 27. Herramienta del robot.

Para calibrar la herramienta necesitamos un punto de referencia físico y con una punta lo más fina posible para realizar un buen calibrado de la herramienta, para ello tenemos un soporte con un tornillo afilado que nos ayudará a la calibración de la herramienta (Figura 28).



Figura 28. Objeto de calibración.

Es recomendable fijarlo en la mesa de trabajo para que en caso de golpearlo levemente con el robot, no se mueva.

Empezamos creando dicha herramienta en el *FlexPendant* de nuestro robot y la celebraremos en 3 puntos, estos 3 puntos se tomarán en referencia de la punta de nuestro objeto de calibrar que nos facilitara esta tarea, una vez calibrada con su TCP creado correctamente, le tenemos que poner una masa a nuestra herramienta y estamos listos para captar puntos o robtarget por donde nuestro robot se moverá o utilizará de referencia para captar las piezas o depositarla en su posición.

4.2.7. Captación de Robtarget.

Para la captación de robtarget deberemos haber hecho antes la calibración de la herramienta y la calibración de la cámara ya que para esta última necesitaremos saber donde se encuentra el punto de referencia de la cámara para captarlo como robtarget y así el robot tener la misma referencia que la cámara a la hora de dar las coordenadas de la posición de las piezas.

Para crear robtarget cogeremos el *FlexPendant* y nos dirigimos al menú de este, entraremos en la pestaña de “*Datos de programa*” (Figura 29), esto nos abrirá otro menu con diferentes opciones (Figura 30) y algunas están ocultas, en la parte

inferior derecha podremos hacer que se muestren todas hasta que encontremos la que queremos que es robtarget (Figura 30).

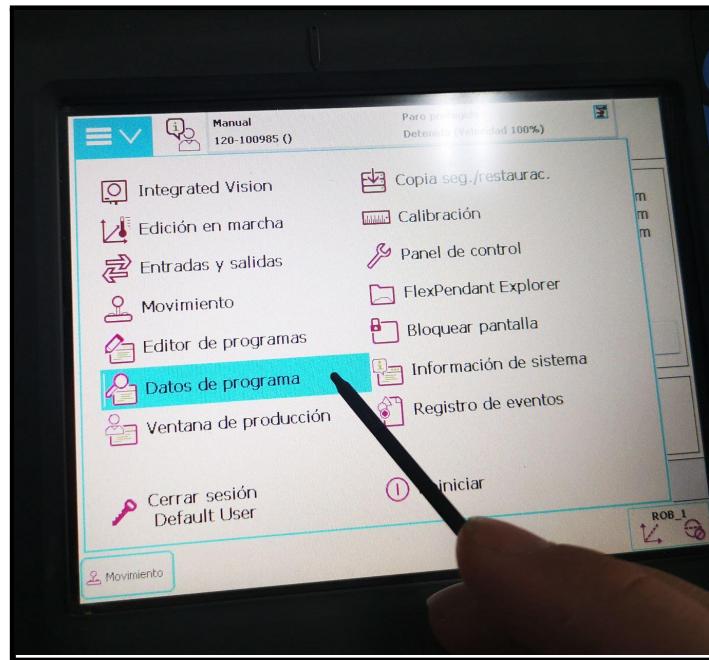


Figura 29. Opciones menú FlexPendant.

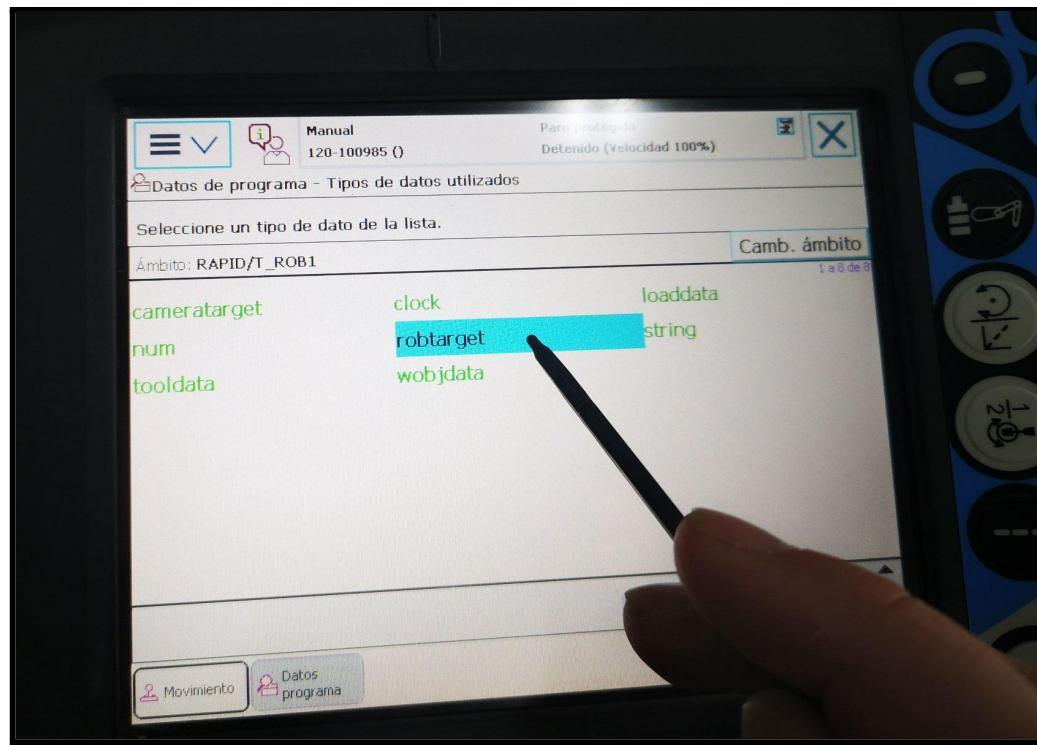


Figura 30. Opciones menú Datos del programa.

Nos saldrá la ventana vacía, ahora tendremos que mover el robot a las posiciones deseadas para capturar estos puntos y cada vez que tengamos el robot en posición ir a la ventana de “Robtarget” y pinchar en nuevo, nos abrirá una ventana donde tendremos que poner el nombre al punto y si no es necesario no tocar nada más.

Para la creación del punto que se utilizará como referencia del punto de coordenadas de la cámara es recomendable colocar una pieza, el punto lo más centrado posible sobre la pieza y con el robot colocar la ventosa que tiene el TCP encima de esta para captar el punto ahí y así tendremos la altura de las piezas incluida.

Una vez creados todos los puntos deberemos verlos en la pantalla igual que la Figura 31, y habremos terminado con la captación de robtarget para tenerlos disponibles en *RAPID*.

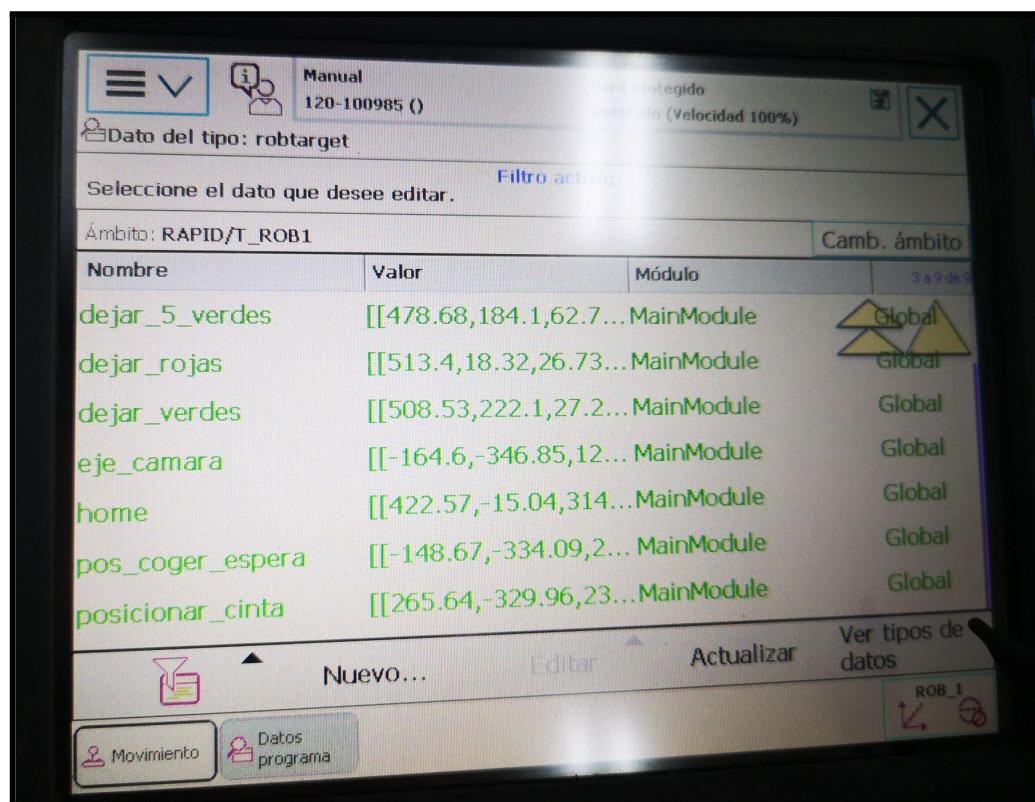


Figura 31. Ventana de Robtarget en FlexPendant.

4.3. Programación.

4.3.1. Programación trabajo de la cámara.

Para la programación de la cámara nos tendremos que dirigir a la ventana de “Vision” y estar conectados a la cámara, cargar el trabajo donde ya calibraremos dicha cámara y realizar la programación sobre este trabajo que será el que utilicemos en *RAPID*.

Para que la cámara sea capaz de reconocer formas tendremos que añadirle una herramienta de ubicación de piezas, estas se encuentran en la parte superior, al clicar en la opción nos saldrá una ventana emergente con diversas opciones, ahora solo nos interesa la herramienta de “*Patrón PatMax*” (Figura 32) que también lo encontraremos en 2 versiones diferentes, para capturar de 1 a 10 piezas simultáneas iguales o de 1 a 50, en nuestro caso solo veremos una pieza simultánea por ello solo usaremos la herramienta que detecta una pieza aunque no pasa nada si utilizamos cualquier opción de las mencionadas anteriormente.

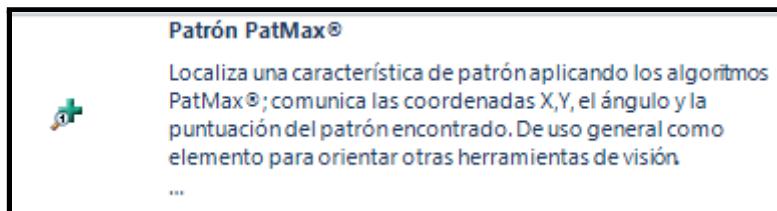


Figura 32. Herramienta Patrón PatMax.

Una vez elegida esta herramienta nos saldrá ya en la imagen 2 rectángulos, uno llamado “*Buscar*” y el otro “*Modelo*”, para entender esto, el color morado será el rectángulo seleccionado que podemos editar. En caso de que no haya colocado las piezas aún bajo la cámara ponga una redonda y una cuadrada y adquiera una imagen nueva con la opción que se encuentra en la parte superior izquierda de la ventana “*Vision*”, el modelo es lo que la cámara buscará dentro del rectángulo que usted le marque con el rectángulo de buscar, tanto la zona de búsqueda como el modelo se pueden cambiar de forma en la parte inferior aunque el que no debería variar es el de búsqueda ya que debería captar por toda la zona de la imagen y es por eso que deberemos hacerlo más grande hasta ocupar toda la imagen, en el caso del modelo deberemos transformarlo en forma rectangular/cuadrada para crear el modelo de la pieza cuadrada (Figura 33).

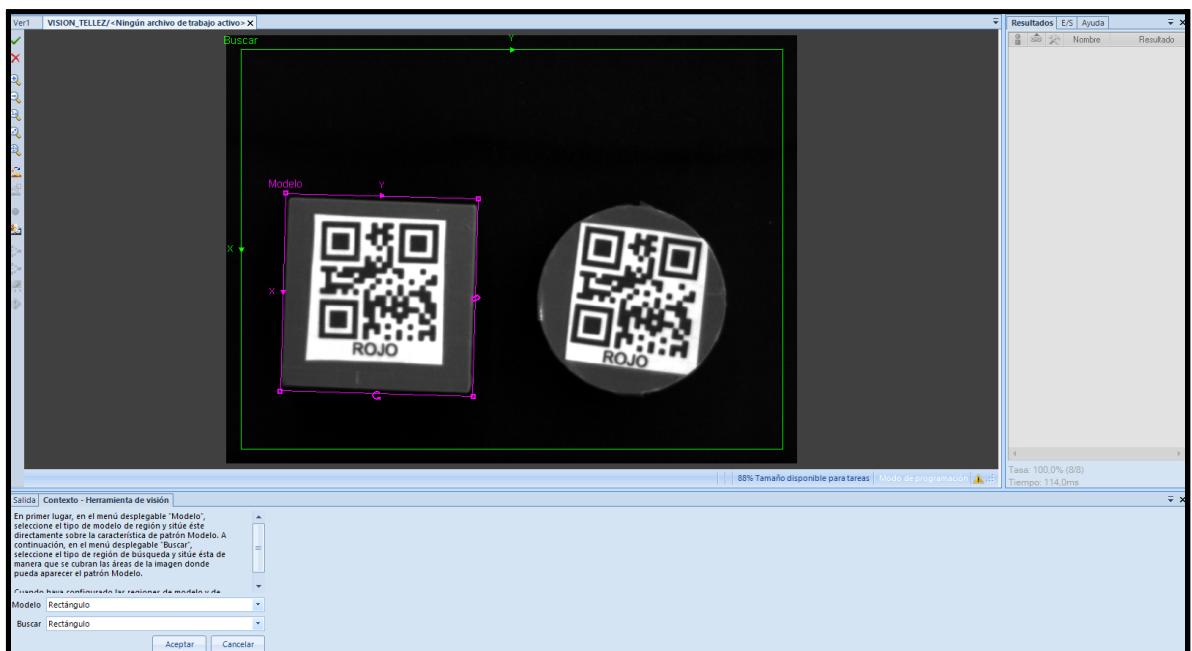


Figura 33. Ajuste zona búsqueda y modelado.

Una vez tengamos ya las zonas definidas clicamos en “Aceptar” y nos creará la herramienta de búsqueda (Figura 34), le pondremos un nombre a esta herramienta que en este caso será “Cuadradas” para tener la herramienta ubicada, entramos en la pestaña de ajuste y en “Tolerancia de rotación” le pondremos 180, esto es el ángulo de rotación que la cámara aceptara para captar la pieza, al poner 180º capturara la pieza en cualquier posición.

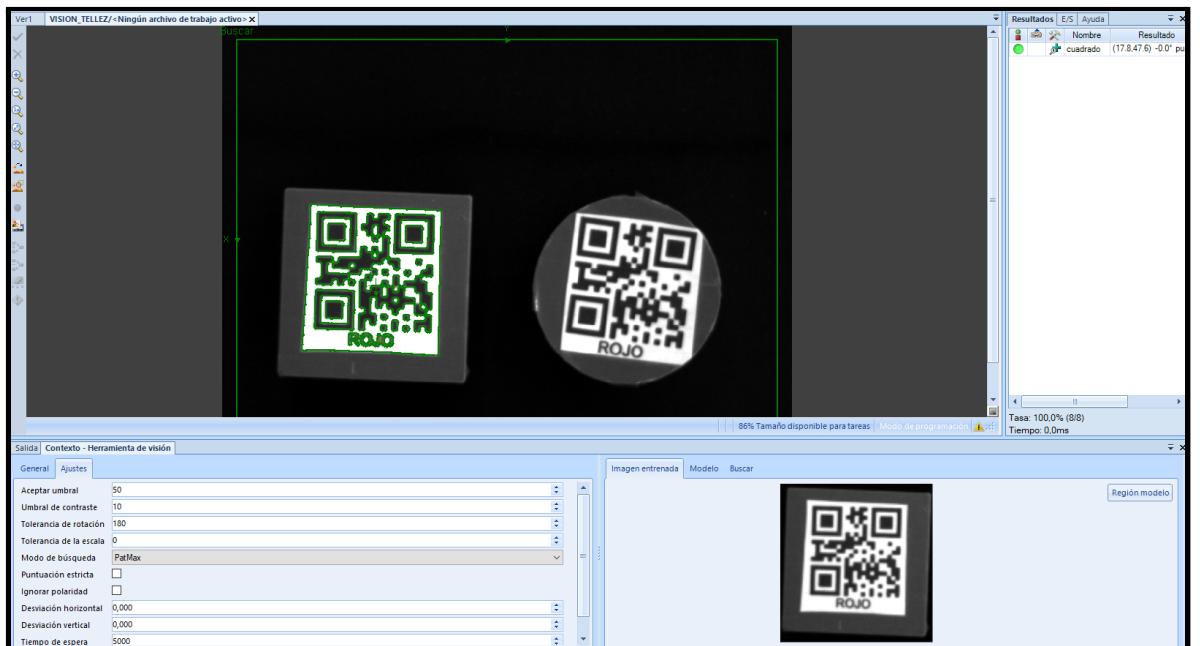


Figura 34. Herramienta creada y ajustada.

Volveremos a realizar el proceso de elección de herramienta de búsqueda sólo que en este caso cambiaremos la forma del modelo a un modelo redondo, esto nos facilitara a la hora de detectar la pieza redonda (Figura 35).



Figura 35. Ajuste zona búsqueda y modelo redondo.

Una vez tengamos las zonas en su posición, pasaremos al siguiente paso que es poner un nombre a la herramienta en nuestro caso será “Circular” (Figura 36) y como hicimos con las cuadradas en la tolerancia de rotación le pondremos un 180.

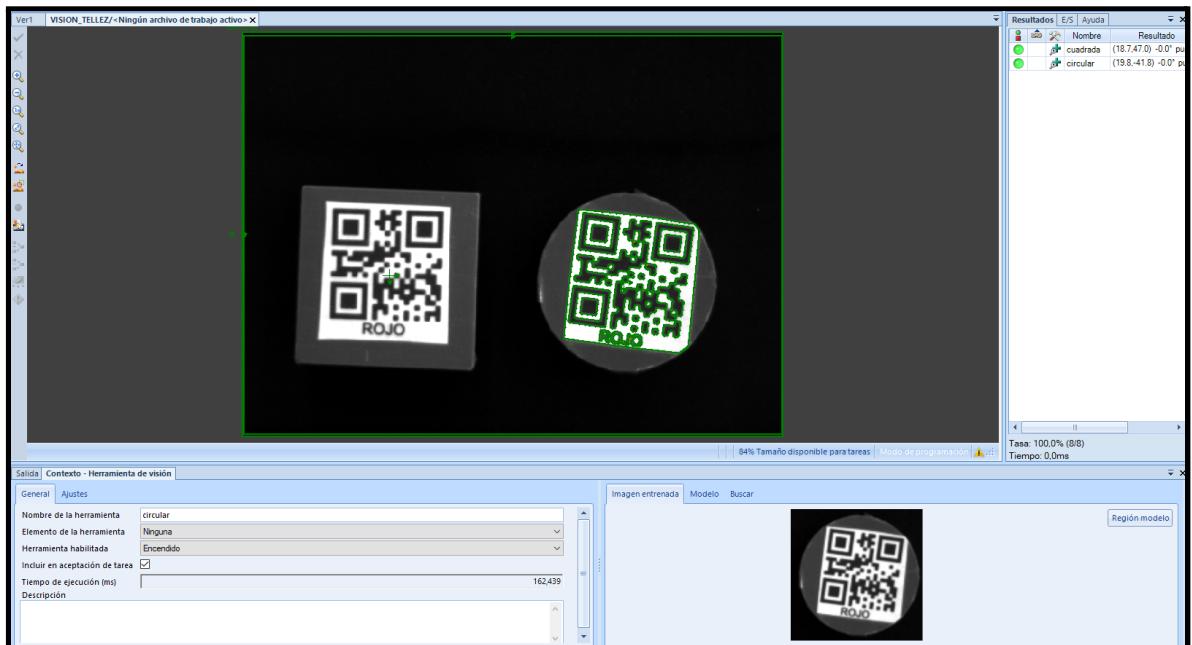


Figura 36. Herramienta creada y ajustada.

Ahora toca la parte de la lectura del QR que nos dirá un color puesto que la cámara es monocromo y no podemos detectar por colores, podemos poner cualquier texto en el QR, se recomienda que no sea muy largo.

Para esto iremos a la opción de herramienta para inspección de piezas donde encontraremos la herramienta “Leer código 2D” (Figura 37).

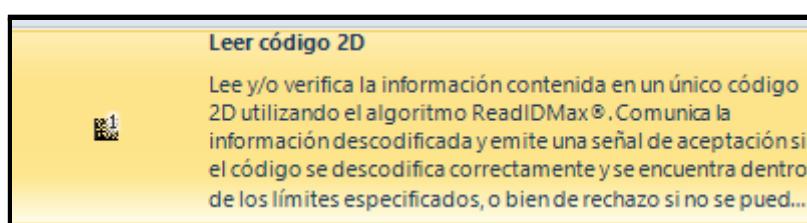


Figura 37. Herramienta de inspección Leer código 2D.

Una vez la tengamos seleccionada nos saldrá en la visión de la imagen para posicionar la zona de búsqueda de dicha herramienta, esta herramienta no precisa de ningún modelo solo una zona de búsqueda, para que siempre nos detecte una pieza en la imagen pondremos esta zona lo más grande posible cubriendo toda la imagen de la cámara (Figura 38).

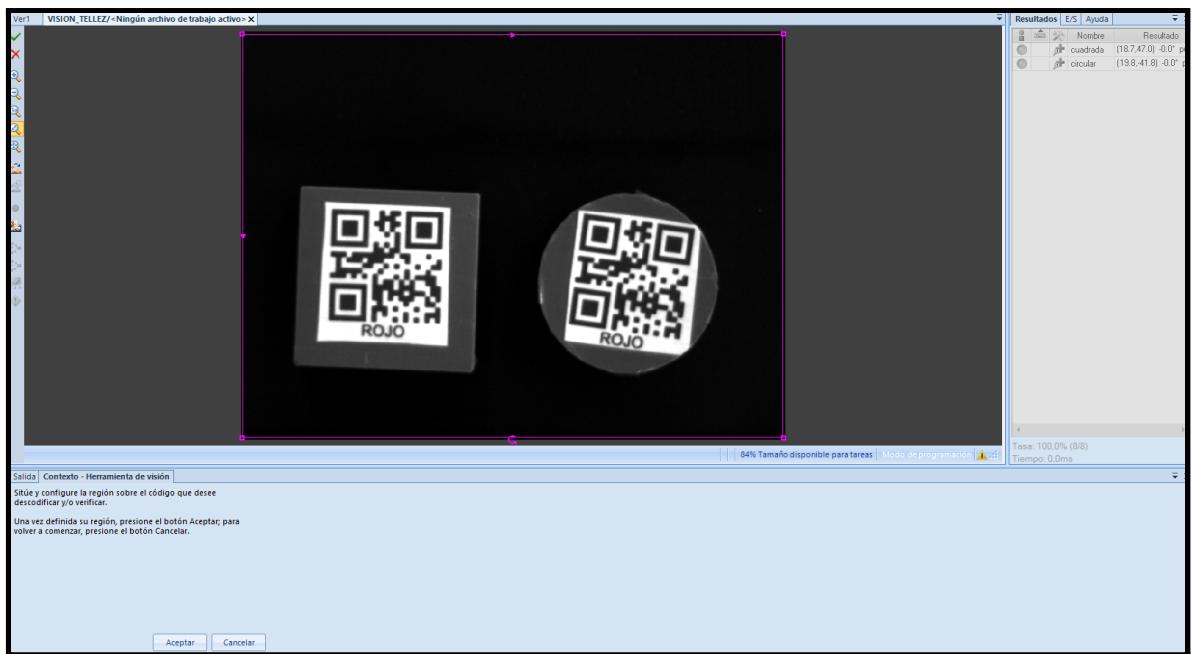


Figura 38. Zona de búsqueda código 2D.

Una vez colocada la zona de búsqueda tendremos que ajustar esta herramienta, por defecto viene para buscar un tipo de simbología llamado “*Data Matrix*”, en ajustes de la herramienta podremos cambiar esta opción a “*QR Code*” que es la opción que nos interesa para esta herramienta (Figura 39).



Figura 39. Ajustes de herramienta de inspección.

Ya terminado este paso tendremos todas las herramientas necesarias para el funcionamiento de la cámara, solo nos falta configurar las salidas a *RAPID* que es donde queremos que los datos de la cámara lleguen, para ello nos iremos a la opción en la parte superior “*Salidas a RAPID*”, esto nos abrirá una ventana donde podemos crear varias piezas para gestionar datos, en nuestro caso haremos 2 piezas, “*Circulo*” y “*Cuadrado*” en estas piezas tendremos que meter los datos que queremos en *rapid*, para ambas piezas serán iguales los datos solo que vendrán de diferentes piezas.

Necesitaremos la posición en Y de las piezas, pondremos el grupo del que queremos esa posición, el grupo se refiere a la herramienta creada en caso de redondas ira para círculo y en cuadradas para cuadrado, le pediremos que nos suelte de esta herramienta la posición en Y que en dicho apartado se llama

“Fixture.Y”, en la misma fila nos mostrará el nombre de la ruta de *RAPID* que deberemos seguir para leer este dato.

Pasamos a los valores de calidad de captura y el número que define cuando es redonda o cuadrada la pieza, para la calidad la añadiremos en el “value 1” y pondremos que queremos recoger en ese dato el valor de “Fixture.Score” y en el dato “Value 2” captamos el número de aceptación límite que debemos definir manualmente en cada herramienta para poder distinguir entre cuadradas y redondas, en este caso las redondas tendrán un valor de 50 y las cuadradas de 49, este número nunca varía y nos ayudará para distinguir entre formas, el valor que queremos añadir a este dato será el que nos suelte con “Accept_Threshold” y para terminar nos falta el valor del QR, para esto necesitamos un tipo de dato *STRING*, es por eso que iremos a los datos “String 1”, marcaremos la herramienta de del QR para pasar el dato y simplemente tendremos que decir que queremos pasar el valor del string, en este caso nos pasara “ROJO” o “VERDE” (Figura 40).

Figura 40. Ventana Salidas a RAPID.

Con esto último habremos terminado con la programación de la cámara, tendremos que guardar el trabajo para que este quede grabado con todo ya creado, herramientas y salidas de *RAPID* y podremos pasar al siguiente paso.

4.3.2 Programación en RAPID.

Para la programación de *RAPID* programaremos directamente sobre la controladora por lo que se recomienda realizar copias de seguridad cada un cierto tiempo para tener el progreso guardado, programaremos directamente en la controladora porque nos saldrán automáticamente los *RobTarget* y si necesitamos ajustarlos más no tendremos que cambiar nada porque se ajustarán solos, a la hora de probar el proyecto no tendremos que pasarlo en pendrive a la controladora.

Puesto que es la controladora real ya tendremos una tarea creada previamente donde se han generado tanto la definición de la herramienta como los *RobTarget*, después de estos datos definiremos datos necesarios para el programa como para la visión de la cámara (Figura 41).

```

ROB1/MainModule.x
1 MODULE MainModule
2   TASK PERS toodata T_valvula:=[TRUE, [[-21.3786, -23.6601, 76.4411], [1, 0, 0, 0]], [1, [0.1, 0, 0], [1, 0, 0, 0], 0, 0, 0]];
3     CONST robtarget pos_coger_espresa:=[[148.67, -334.09, 237.40], [0.0221112, 0.379172, -0.925061, -0.0013214], [-2, -1, -2, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
4     CONST robtarget home:=[[422.57, -15.04, 314.73], [0.0228078, 0.379122, -0.925082, -0.00129255], [-1, -1, -1, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
5     CONST robtarget eje_camara:=[[164.60, -346.85, 129.74], [0.0221252, 0.379112, -0.925085, -0.00127603], [-2, -1, -2, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
6     CONST robtarget posicionar_cinta:=[[265.64, -329.96, 237.40], [0.0221105, 0.379148, -0.925071, -0.00130637], [-1, -1, -2, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
7     CONST robtarget dejar_verdes:=[[508.53, 222.18, 27.27], [0.0218995, 0.379168, 0.925068, -0.00125911], [0, 0, -1, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
8     CONST robtarget dejar_rojas:=[[513.40, 18.32, 26.73], [0.0217502, 0.37925, -0.925038, -0.00111166], [0, -1, -1, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
9     CONST robtarget dejar_5_rojas:=[[478.69, -19.33, 62.78], [0.022107, 0.379084, -0.925097, -0.00128542], [-1, -1, -1, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
10    CONST robtarget dejar_5_verdes:=[[478.68, 184.10, 62.77], [0.0221189, 0.379096, -0.925092, -0.00128612], [0, 0, -1, 0], [9E+09, 9E+09, 9E+09, 9E+09, 9E+09]];
11
12    CONST string myjob := "vision_26042022"; !Creamos el nombre del trabajo de la cámara y definimos con el archivo guardado.
13    VAR cameratarget mycameratarget; !se utiliza para intercambiar datos de visión entre la imagen de cámara y el programa de RAPID.
14    VAR num offfs_pieza; !valor que tendrá la posición en Y de la pieza captada por la cámara.
15    VAR num estado_cam; !valor para el arranque de la cámara.
16    VAR num piezas_rojas; !valor para memorizar la cantidad de piezas rojas paletizadas
17    VAR num piezas_verdes; !valor para memorizar la cantidad de piezas verdes paletizadas
18    VAR num offset_dejar_rojas_y; !valor para ajustar el offset al dejar piezas rojas en Y
19    VAR num offset_dejar_rojas_x; !valor para ajustar el offset al dejar piezas rojas en X
20    VAR num offset_dejar_verdes_y; !valor para ajustar el offset al dejar piezas verdes en Y
21    VAR num offset_dejar_verdes_x; !valor para ajustar el offset al dejar piezas verdes en X

```

Figura 41. Definición de datos usados en programa.

A continuación crearemos el procedimiento “Main” que es donde se encontrará todo nuestro programa.

Lo primero que haremos al iniciar el procedimiento será activar la cámara, esto solo será necesario hacerlo una vez por ello lo meteremos en una instrucción *IF* y compararemos el valor de “estado_cam” que en caso de ser 1 no volverá a ejecutar el *IF* (Figura 42).

```

25
26   PROC main()
27
28   IF estado_cam < 1 THEN !PONER EN MARCHA LA CÁMARA
29     CamSetProgramMode VISION_TELLEZ;
30     CamLoadJob VISION_TELLEZ, myjob;
31     CamSetRunMode VISION_TELLEZ;
32     estado_cam := 1;
33   ENDIF !MARCHA CÁMARA

```

Figura 42. Activación de la cámara.

El funcionamiento de estas líneas de código es el siguiente, “*CamSetProgramMode*” pondrá la cámara que en este caso es “*VISION_TELLEZ*” en modo programación para poder cargar el trabajo de la cámara con la función “*CamLoadJob*” junto a la cámara que le va a cargar el trabajo y después el trabajo que le va a cargar que definimos anteriormente en “*myjob*”, una vez cargado el trabajo en la cámara usamos la función “*CamSetRunMode*” para poner la cámara en modo ejecución y definimos que cámara queremos poner en este modo, una vez realizado estos procesos ponemos la variable “*estado_cam*” a 1 para no volver a realizar este proceso de activación de la cámara.

Después mandaremos el *ROBOT* a posición *HOME* en caso de que no esté en ella, en caso de que ya haya realizado un paletizado completo y por tanto todas las piezas estén en la posición de paletizado realizamos unas preguntas para poder repetir el programa sin tener que pasar este a *STOP* y esperar de nuevo a la iniciación de la cámara, estas preguntas cómo se puede observar en la Figura 43 no son infinitas y en caso de que no quieras seguir con el proceso o no retires las piezas después de preguntar 2 veces, el programa pasará a *STOP* automáticamente.

```

35      MoveJ home,v200,z0,T_valvula\WObj:=wobj0;
36
37      IF piezas_rojas = 5 AND piezas_verdes = 5 THEN !INICIO PREGUNTAS
38          TPReadNum reg1,"¿QUIERE EJECUTAR OTRA VEZ EL PROCESO? 1= SI / 2= NO"; !PRIMERA PREGUNTA PROCESO
39          TPWrite" ";
40          IF reg1 = 1 THEN !RESPUESTA 1 SEGUIR PROCESO
41              TPReadNum reg2,"¿Ha recogido las piezas de la zona de colocación? 1= SI / 2= NO"; !SEGUNDA PREGUNTA PIEZA A
42              TPWrite" ";
43              IF reg2 = 1 THEN !RESPUESTA 1 PIEZAS A
44                  TPWrite "Iniciando proceso...";
45                  TPWrite"3...";
46                  WaitTime 1;
47                  TPWrite"2...";
48                  WaitTime 1;
49                  TPWrite"1...";
50                  WaitTime 1;
51                  TPErase;
52                  piezas_verdes := 0;
53                  piezas_rojas := 0;
54              ELSEIF reg2 = 2 THEN !RESPUESTA 2 PIEZAS A
55                  TPReadNum reg3,"¿Has terminado de recoger las piezas? 1= SI / 2= NO"; !TERCERA PREGUNTA PIEZA B
56                  IF reg3 = 1 THEN
57                      TPWrite "Iniciando proceso..."; !RESPUESTA 1 PIEZAS B
58                      TPWrite"3...";
59                      WaitTime 1;
60                      TPWrite"2...";
61                      WaitTime 1;
62                      TPWrite"1...";
63                      WaitTime 1;
64                      TPErase;
65                      piezas_verdes := 0;
66                      piezas_rojas := 0;
67                  ELSEIF reg3 = 2 THEN
68                      TPWrite "Cancelando proceso..."; !RESPUESTA 2 PIEZAS B
69                      piezas_verdes := 0;
70                      piezas_rojas := 0;
71                      TPWrite"5...";
72                      WaitTime 1;
73                      TPWrite"4...";
74                      WaitTime 1;
75                      TPWrite"3...";
76                      WaitTime 1;
77                      TPWrite"2...";
78                      WaitTime 1;
79                      TPWrite"1...";
80                      WaitTime 1;
81                      TPErase;
82                      Break;
83                  ELSE
84                      ENDIF
85                  ENDIF
86                  ELSE !RESPUESTA 2 SEGUIR PROCESO
87                      TPWrite "Hasta la próxima...";
88                      piezas_verdes := 0;
89                      piezas_rojas := 0;
90                      WaitTime 5;
91                      TPErase;
92                      Break;
93                  ENDIF
94              ENDIF !FIN PREGUNTAS

```

Figura 43. Preguntas para seguir el proceso de paletizado o acabar con dicho proceso.

Estas preguntas sólo saldrán si ya realizó el primer paletizado, en caso de no hacerlo pasará directamente a la iniciación del proceso el cual inicia pidiendo una captura de imagen de la cámara con la función “*CamReqImage*” este sera el que active el disparador de la cámara, para asegurarnos de que la cámara siempre de una imagen tendremos que dirigirnos a la ventana de “*Vision*” y en “*Configurar imagen*” tendremos que poner el disparador en continuo para que este constantemente mandando imágenes así que caso de que la función se active antes de haber una pieza bajo la cámara no de un error en la siguiente función. La siguiente función sera “*CamGetResult*” y se encargará de guardar los datos de la cámara en el tipo de dato definido anteriormente llamado “*mycameratarget*”, esta función tiene un tiempo límite de espera hasta captar los datos que por defecto tiene de máximo 120s, en caso de pasar este tiempo da un error y pasa a

STOP la tarea. En todo momento irán apareciendo mensajes en el *FlexPendant* de como va el estado del proceso y que está realizando el *ROBOT* en ese instante. Una vez captados los datos colocamos el robot en posición de espera para recoger la pieza detectada (Figura 44).

```
TPWrite "INICIANDO LECTURA DE LA PIEZA";  
  
CamReqImage VISION_TELLEZ;  
CamGetResult VISION_TELLEZ, mycameratarget;  
  
MoveJ posicionar_cinta,v300,z200,T_valvula\WObj:=wobj0;  
MoveJ pos_coger_espera,v300,fine,T_valvula\WObj:=wobj0;  
TPWrite "MOVIENDO HACIA RECOGIDA DE PIEZA";
```

Figura 44. Disparador cámara y captura de datos.

Con los datos captados y el *ROBOT* en posición empezamos el proceso de identificación de la pieza, su posición en la cinta transportadora, su forma y el texto de su QR. Para explicar este proceso tomaremos sólo la parte de lectura de piezas redondas rojas puesto que para las otras 3 variantes el código es exactamente el mismo, el código se adjuntará completamente en la parte de Anexos.

El primer dato a comparar será el string que nos indicará si la pieza es “ROJA” o si es “VERDE” ya que esto es lo que definirá su posicionamiento de paletizado y no su forma, estos datos los tenemos guardados en la variable “mycameratarget” que es un array de datos pasados por la cámara, solo tenemos que saber la ruta de los datos que nos interesa, para los datos en string tendremos que utilizar la ruta “mycameratarget.string1” una vez comparado el string y sepamos de qué color es cada pieza comparamos la forma de la pieza con la ruta “mycameratarget.val2” en caso de ser 50 sabremos que es redonda y si es 49 será cuadrada. Continuamos con la medición de la calidad de la captura de la pieza, esto es meramente un trámite puesto que es muy complicado tener una captura por debajo del límite que nosotros pedimos en código, la ruta de este valor es “mycameratarget.val1”.

Si todo está correcto continuamos a pasar los datos de la posición en Y a nuestra variable “offs_pieza”, la ruta de este valor se encuentra en “mycameratarget.cframe.trans.y”, en mi caso “offs_pieza” tengo que compararlo puesto que calibre la cámara de forma errónea y el eje Y de la cámara es el invertido del eje Y del robot y tengo que cambiar el símbolo de este dato para que el robot vaya a la posición correcta de la pieza, si la calibración de tu camara esta bien este paso no es necesario y pasaremos a los movimientos de captura de la pieza, en estos movimientos siempre usaremos offset puesto que tomaremos como referencia en *RobTarget* que utilizamos para tomarlo como referencia del eje de coordenadas de la cámara.

Cuando estemos sobre la pieza activaremos la salida que controla la electroválvula de la ventosa y añadimos una espera de 0.5s para asegurarnos que la captura se realiza correctamente (Figura 45).

```

!inicio rojas
IF mycameratarget.string1 = "ROJO" THEN

    IF mycameratarget.val2 = 50 THEN !piezas redondas rojas
        TPWrite "ES UNA PIEZA REDONDA ROJA";
        IF mycameratarget.val1 > 50 THEN !medimos el valor de medicion

            offs_pieza := mycameratarget.cframe.trans.y; !cogemos pieza en cinta

            IF offs_pieza < 0 THEN !correcion de simbolo por mal calibrado de la cámara.
                offs_pieza := abs(offs_pieza);
            ELSE IF offs_pieza > 0
                offs_pieza := 0 - offs_pieza;
            ENDIF
            !movimiento para capturar pieza.
            MoveL offs (eje_camara,0,offs_pieza,40),v100,fine,T_valvula\WObj:=wobj0;
            MoveLdo offs (eje_camara,0,offs_pieza,-0.7),v50,fine,T_valvula\WObj:=wobj0,do7_DSQC652,1;
            WaitTime 0.5;
            TPWrite "PIEZA REDONDA ROJA CAPTURADA";
            MoveL offs (eje_camara,0,offs_pieza,40),v100,fine,T_valvula\WObj:=wobj0;
        ENDIF

```

Figura 45. Comparación de datos de la cámara y captura de la pieza.

Con la pieza ya capturada con la ventosa pasamos al paletizado de la misma, para ir a estos puntos haremos un *movimiento J* sobre la cinta y iniciamos comparación para saber en qué posición deberemos dejar la pieza en caso de la cantidad de piezas ya colocadas, para colocar las piezas en diferentes posiciones jugaremos con un offset tanto en Y como en X con las variables anteriormente creadas llamadas “*offset_dejar_rojas_x*” y “*offset_dejar_rojas_y*”, a estos datos como se puede observar en la Figura 46, le añadiremos un valor de 75, este valor es en *mm*, y será la distancia que dejaremos entre el centro de una pieza y la siguiente, cada pieza mide 6cm de lado o en caso de las redondas 6cm de diámetro y con este valor de offset nos aseguramos que dejen una distancia aproximada entre cada una de las piezas de 1,5cm. Una vez tengamos 5 piezas rojas colocadas, los offset volverán a un valor de 0.

Cada vez que coloque una pieza en la posición de paletizado hará un *movimiento J* a *HOME* para esperar a la próxima toma de datos de la pieza siguiente o para esperar a la orden de acabar el proceso o repetirlo (Figura 46).

```

127      !ya hemos cogido la pieza de la cinta
128      MoveJ posiciona_cinta,v300,z200,T_valvula\WObj:=wobj0;
129      TPWrite "MOVIMIENTO HACIA LUGAR DE DEPOSITO DE PIEZA";
130
131      !iniciamos paletizacion de las piezas redondas rojas
132      IF piezas_rojas <=1 THEN
133          MoveL offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,40),v200,z0,T_valvula\WObj:=wobj0;
134          MoveLDO offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,-0.5),v50,fine,T_valvula\WObj:=wobj0,do7_DSQC652,0;
135          WaitTime 1;
136          MoveL offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,40),v200,z0,T_valvula\WObj:=wobj0;
137          offset_dejar_rojas_y := -75;
138          TPWrite "PIEZA DEPOSITADA";
139      ENDIF
140      IF piezas_rojas = 2 or piezas_rojas = 3 THEN
141          offset_dejar_rojas_x := -70;
142          MoveL offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,40),v200,z0,T_valvula\WObj:=wobj0;
143          MoveLDO offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,-0.5),v50,fine,T_valvula\WObj:=wobj0,do7_DSQC652,0;
144          WaitTime 1;
145          MoveL offs (dejar_rojas,offset_dejar_rojas_x,offset_dejar_rojas_y,40),v200,z0,T_valvula\WObj:=wobj0;
146          offset_dejar_rojas_y := -75;
147          TPWrite "PIEZA DEPOSITADA";
148      ENDIF
149      IF piezas_rojas = 4 THEN
150          MoveL offs (dejar_5_rojas,0,0,40),v200,z0,T_valvula\WObj:=wobj0;
151          MoveLDO offs (dejar_5_rojas,0,0,0),v50,fine,T_valvula\WObj:=wobj0,do7_DSQC652,0;
152          WaitTime 1;
153          MoveL offs (dejar_5_rojas,0,0,40),v200,z0,T_valvula\WObj:=wobj0;
154          TPWrite "PIEZA DEPOSITADA";
155      ENDIF
156
157      piezas_rojas := piezas_rojas + 1; !sumamos 1 en el palet hasta un total de 5
158      IF piezas_rojas = 2 THEN
159          offset_dejar_rojas_y := 0;
160      ENDIF
161      IF piezas_rojas = 5 THEN
162          offset_dejar_rojas_y := 0;
163          offset_dejar_rojas_x := 0;
164          !ya se han paletizado 5 piezas
165      ENDIF
166      TPWrite "MOVIENDO A HOME";
167      MoveJ home,v200,fine,T_valvula\WObj:=wobj0;
168      TPErase;
169  ENDIF !fin piezas redondas rojas

```

Figura 46. Paletizado de piezas Redondas rojas.

Con esto habría terminado el paletizado de las piezas tanto redondas como cuadradas y empezaron a salir en el *FlexPendant* las preguntas de la Figura 43. Como ya comente anteriormente el resto del código solo varia en la comparación de datos de la cámara para definir si la pieza es redonda verde, cuadrada roja o cuadrada verde, podemos copiar el código ya realizado y cambiar con mucho atención al detalle los cambios necesarios en cada parte del código donde sea necesario.

4.4. Verificación del correcto funcionamiento y ajustes necesarios.

Con el programa ya acabado en *RAPID* lo aplicamos a la controladora y ponemos esta en modo automático. Cuando estemos listos ejecutaremos la tarea y prestaremos atención en cada detalle para verificar posibles errores o mejoras del programa para optimizarlo.

Observaremos que los mensajes que hemos escrito en el código que deben salir en el *FlexPendant* salen correctamente sin errores y en su momento (Figura 47).

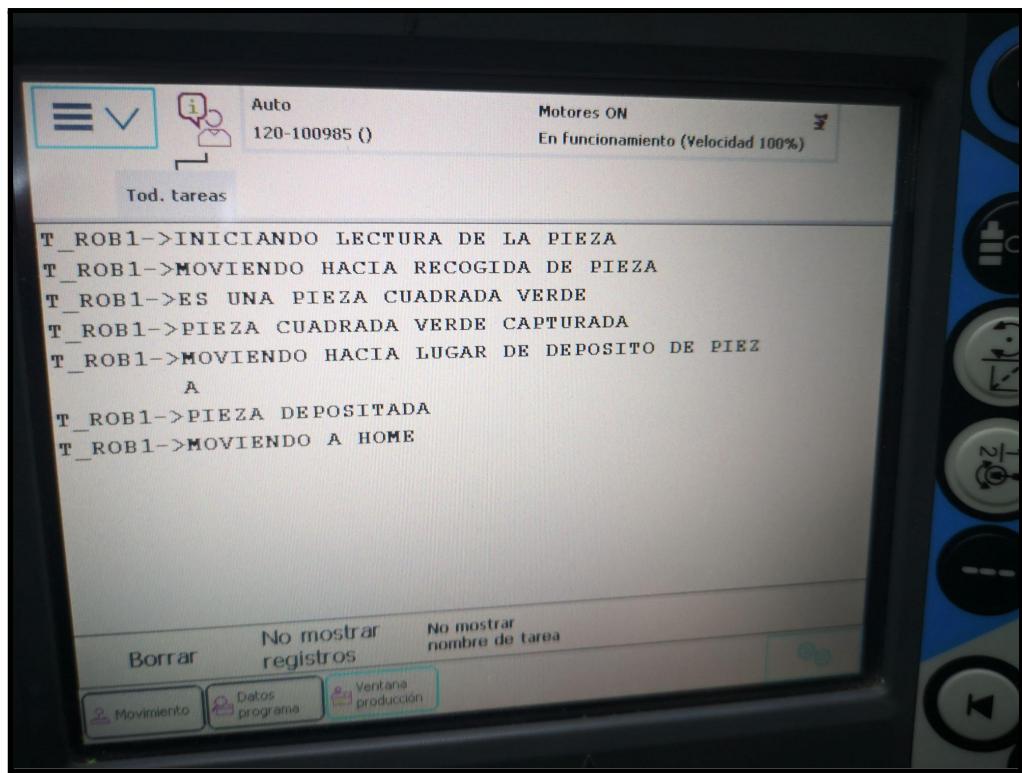


Figura 47. Mensajes del proceso en FlexPendant.

Verificamos que el robot hace unos movimientos cómodos sin hacer cosas extrañas o que puedan dañar al mismo y que captura correctamente las piezas con la ventosa (Figura 48).

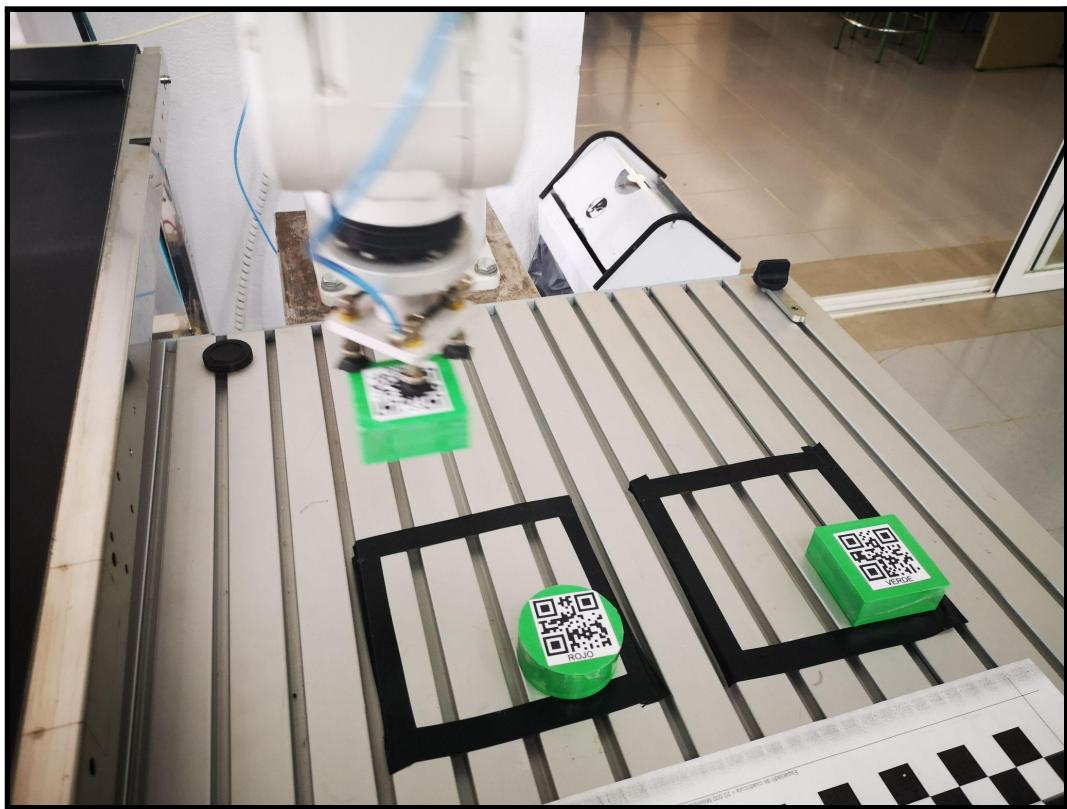


Figura 48. Movimiento robot y captura de pieza con ventosa.

Una vez terminado el primer paletizado verificamos que las preguntas están bien programadas en el *FlexPendant* y que el proceso se puede repetir o acabar a voluntad (Figura 49).

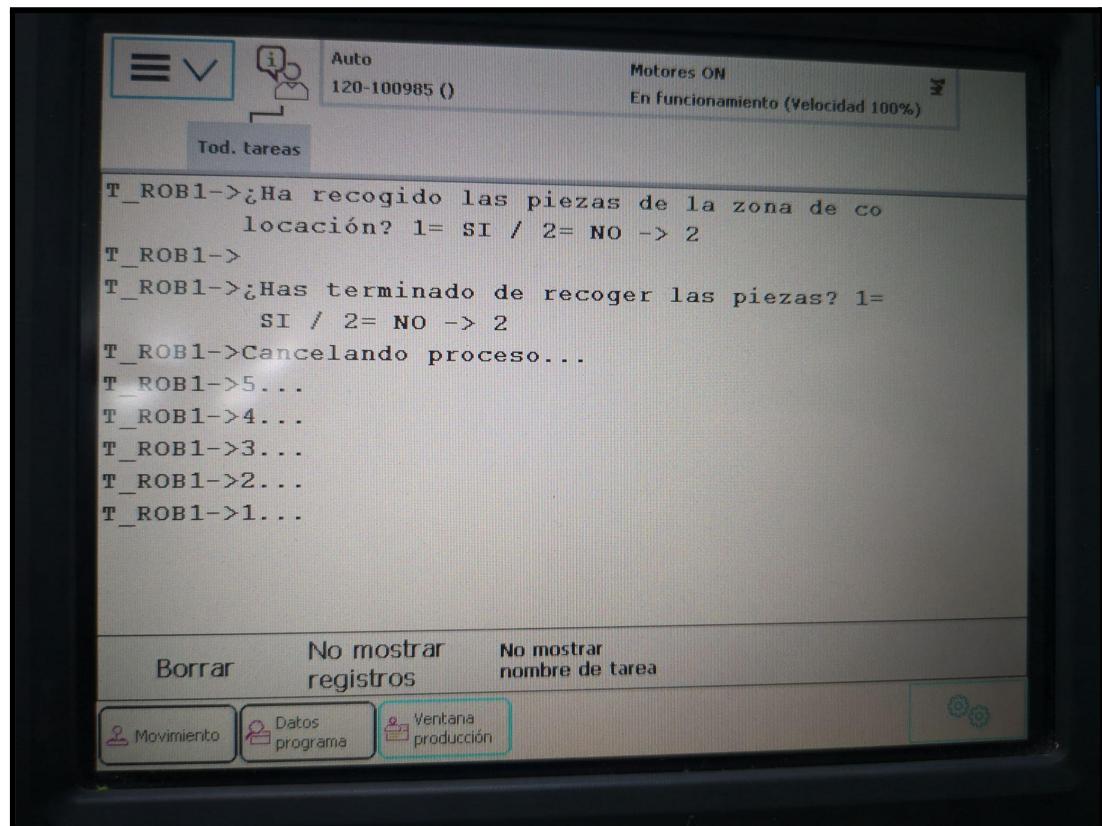


Figura 49. Preguntas sobre continuidad del proceso en FlexPendant.

Por último detallar algunos ajustes realizados que posiblemente surjan, el robot en ocasiones no realiza la captura de la pieza, esto se debe a varias opciones, la primera es que posiblemente el compresor no tenga aire comprimido suficiente para que la ventosa actúe correctamente, también posiblemente es que la cinta no esté a nivel o el robot y por ello en ciertas partes de la cinta la pieza se encuentre más abajo de lo que está el punto de referencia, para ello añadiremos un offset en el eje Z de -0.4 o superior hasta un máximo de -0.7 por razones de seguridad, para que la ventosa baje un poco más y pueda realizar la succión correctamente.

4.5. Optimización del programa.

Con el programa ya acabado y verificado su correcto funcionamiento podemos buscar siempre formas de hacer que este proceso sea más rápido y seguro para el robot.

Las primeras pruebas que realizamos mostraron que en la activación de la cámara tardaba mucho entre paletizados, es por eso que a parte de añadir la activación de la cámara en un condicionador añadimos las preguntas para saber en todo momento si se seguirá ejecutando el proceso o se detendrá. Esto nos permite saltarnos el proceso de activación de la cámara en caso de querer repetir el

proceso sin tener que parar la tarea e iniciarla nuevamente lo que necesitaría activar la cámara nuevamente.

Por otro lado inicialmente se realizaban *movimientos L* para posicionar el robot en la cinta transportadora esto suponía que el robot realizaba un movimiento lineal bastante largo para llegar a la posición deseada de espera lo que suponía una pérdida de tiempo, esto se cambió por *movimientos J* con un *zonedata* de 200 lo que dejaba al robot con un movimiento más suave y cómodo para llegar a la posición de espera.

Otros cambios que se realizaron con diversas pruebas fueron aumentar la velocidad de algunos movimientos para disminuir el tiempo de la tarea pero sin perder de lado la seguridad del robot y de las personas que puedan estar cerca del mismo, después de dichas pruebas se cambió en algunos movimientos que se realizaban a una velocidad de v300 a la velocidad final de v1000.

4.6. Funcionamiento final.

El funcionamiento del proceso es para comprobar y utilizar funciones de la visión integrada, por ello tenemos una cámara sobre una cinta transportadora la cual hará que las piezas lleguen a la zona de visión donde está situada la cámara, esta cámara se encarga de capturar imágenes y observa las piezas para detectar su posición sobre la cinta, su forma y leerá un pequeño QR que se encuentra en las piezas.

Una vez la cámara obtenga los datos de dichas piezas, el robot se moverá con un *movimiento J* hacia dichas piezas y la capturará con la ayuda de una ventosa que el mismo accionará para transportarlas a su lugar de depósito ya predefinido.

Realiza este proceso de detección de piezas, captura y paletizado de las mismas unas diez veces, cinco en caso de las piezas Rojas y otras cinco para piezas Verdes, una vez tenga las diez piezas en posición, se moverá a su posición de *HOME* y espera nuevas órdenes que se darán mediante el *FlexPendant* (Figura 50).



Figura 50. Resultado final con robot en posición HOME.

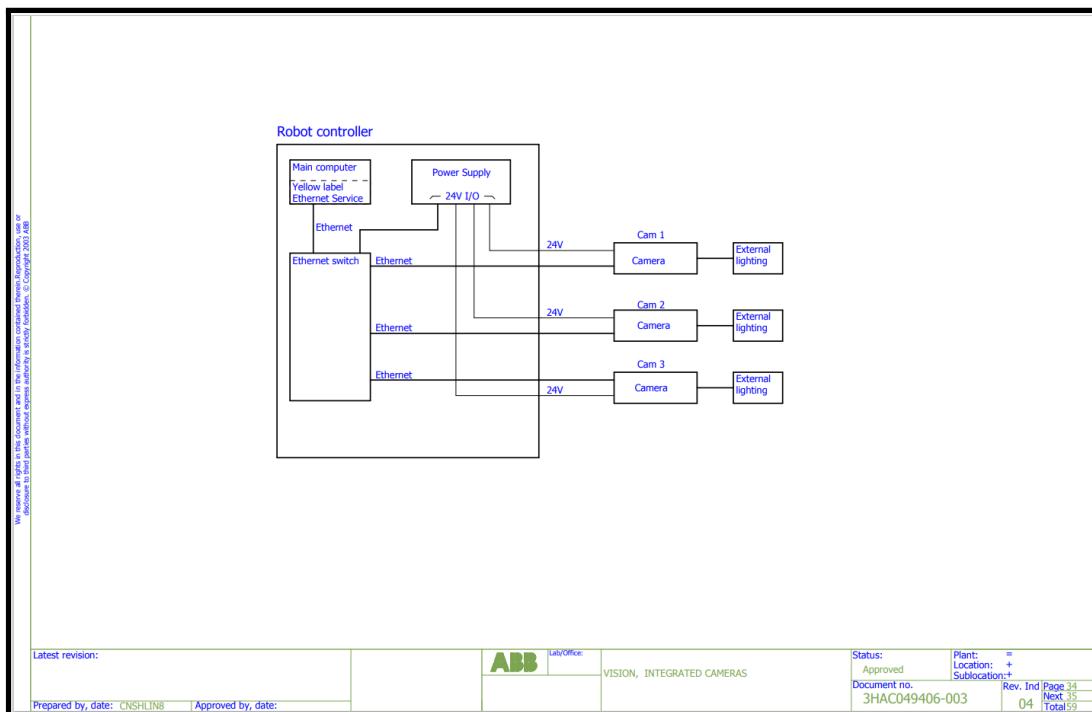
5. CONCLUSIONES

En resumen la dificultad de configurar, programar y utilizar la visión integrada de la controladora IRC5 es media tirando a alta, puesto que tienes que tener muchos factores que sin experiencia previa no conoces, es muy entretenido meterse en esta clase de funciones que nos aporta la controladora. Su capacidad para cualquier tarea a realizar con la ayuda de todas las herramientas que proporciona es para sorprenderse puesto que puedes realizar casi cualquier tarea con la imaginación y capacidad de entendimiento de las herramientas necesarias.

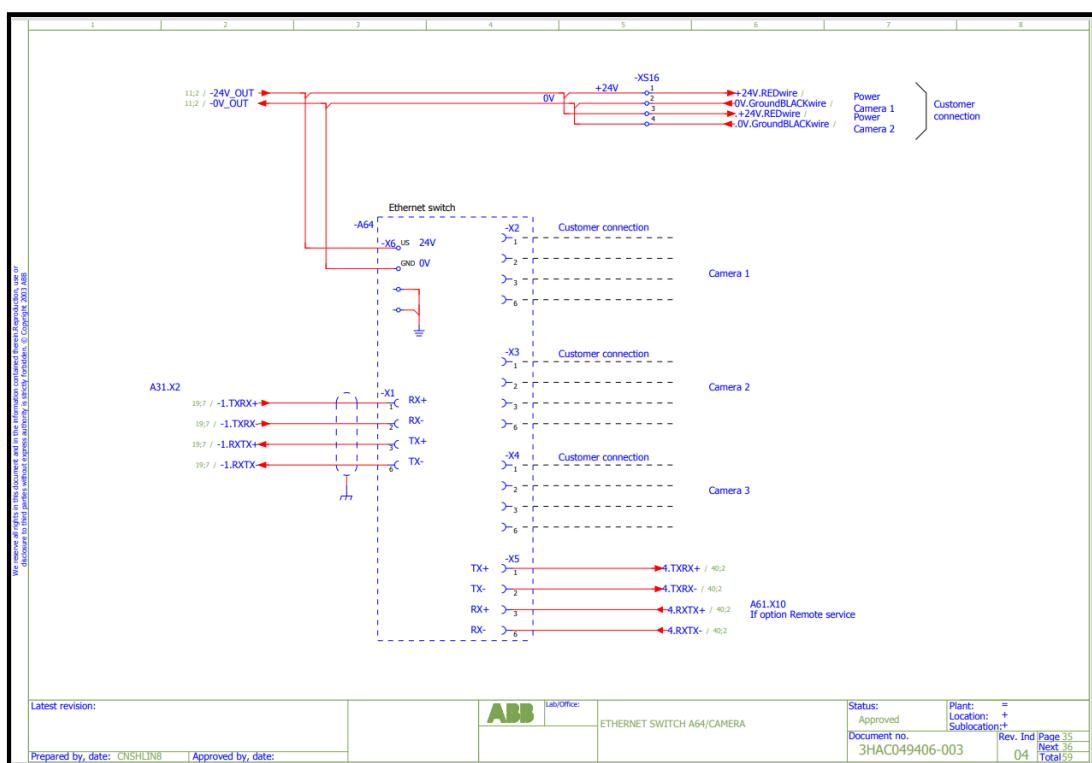
La parte de más dificultad sería la parte de pasar los datos de la cámara al robot para realizar los movimientos pero una vez lo entiendes es bastante sencillo de realizar e intuitivo.

6. Planos y Presupuesto

6.1. Planos.



Plano 1. Alimentación y conexiónado cámaras en IRC5.



Plano 2. Conexiones de red cámaras a IRC5.

6.2. Presupuesto.

El presupuesto de este proyecto es bastante elevado teniendo en cuenta los equipos que vamos a necesitar, por comentar por encima un poco aunque ya deberíamos poseer este equipo hablamos del precio de la controladora más el robot, estos equipos hablando de una controladora *IRC5* y un robot *IRB120* como es en nuestro caso suelen tener un precio aproximado a 13.550€ + IVA en el mejor de los casos, pero en el presupuesto nos centraremos en el precio de la visión integrada.

Para la vision integrada necesitaremos obviamente la camara COGNEX 7000 o una similar a esta que la controladora acepte, esta cámara tiene un precio de 2.845,00 €/unidad. pero no solo necesitaremos la camara, tambien necesitaremos el lente de esta, los cables de alimentación y comunicación, y lo más importante la KEY para la controladora (Presupuesto 1).

DESCRIPCION	CANTIDAD	PRECIO	P. TOTAL
DSQC1020 Integr Vision camera med. res	1	2.845,00	2.845,00 €
8 mm camera lens, LTC-08F	1	420,00	420,00 €
Int. Vis. eth. cable 10m	1	440,00	440,00 €
Int. Vis. pow. cable 10m	1	415,00	415,00 €
Nueva KEY para adaptación de cámara	1	484,00	484,00 €
IMPORTE TOTAL DE LA OFERTA			4.604,00 €

Presupuesto 1. Precios según ABB.

Estos precios como podemos observar son bastante elevados pero lo podemos abaratar en algunos componentes, obviamente la cámara deberá ser una que la controladora acepte y por ello dejaremos esa en el presupuesto, por otro lado si o si necesitaremos la KEY de la controladora, pero podemos rebajar los costes adquiriendo una lente más barata de otro suministrador así como el cable de alimentación y de comunicación. buscando estos objetos a la mejor calidad precio nos quedará un mejor precio (Presupuesto 2).

DESCRIPCION	CANTIDAD	PRECIO	P. TOTAL
DSQC1020 Integr Vision camera med. res	1	2.845,00	2.845,00 €
Lente para camaras 25mm 10MP seeed	1	33,03	33,03 €
cable ethernet NBC-MS, Phoenix Contact	1	79,66	79,66 €
Cable de sensor SAC-12P, Phoenix Contact	1	98,29	98,29 €
Nueva KEY para adaptación de cámara	1	484,00	484,00 €
IMPORTE TOTAL DE LA OFERTA			3.539,98 €

Presupuesto 2. Precios alternativos.

Con estos precios alternativos somos capaces de abaratar el presupuesto 1.064,02€ en comparación a los precios suministrados por ABB, esto es una bajada significativa puesto que en nuestro caso es puramente para enseñanza y no prescindimos de equipos de buena calidad para la industria, no obstante en caso de querer el equipamiento para la industria recomendamos encarecidamente la obtención de los equipos oficiales propuestos por ABB para su mejor funcionamiento y un buen servicio técnico en caso de avería.

7. ANEXOS

7.1. Material utilizado.

Para este proyecto hemos utilizado muchos materiales y equipos informáticos para la realización del mismo. A continuación se proporciona una lista de los materiales y equipos utilizados:

- Ordenador portátil, gama Alta/Media.
- Impresora 3D, gama Alta/Media.
- Cámara COGNEX 7000, gama Alta/Media.
- Controladora IRC5 ABB, gama Alta/Media.
- Robot IRB120, gama Alta/Media.
- Cinta transportadora, gama Media/Baja.
- Compresor de aire pequeño, gama Media/Baja.
- Electroválvula FESTO, gama Alta/Media.
- Tubing neumático, gama Media.
- Filamento PLA, gama Media/Baja.

7.2. Índice Figuras / Planos / Presupuesto.

Se proporcionará a continuación un índice para situar las figuras, planos y presupuestos proporcionados en el proyecto:

- ◆ Figura 1: Cámara COGNEX 7000. (Page, 4)
- ◆ Figura 2: Panel de control cinta transportadora. (Page, 5)
- ◆ Figura 3: Robot IRB120_3_58_G_01. (Page, 6)
- ◆ Figura 4: Posición cinta transportadora. (Page, 6)
- ◆ Figura 5: Piezas utilizadas para el proyecto. (Page, 7)
- ◆ Figura 6: Panel desplegable añadir controladora. (Page, 8)
- ◆ Figura 7: Ventana usuario y contraseña controladora. (Page, 8)
- ◆ Figura 8: Menú desplegable controladora. (Page, 9)
- ◆ Figura 9: Ventana opciones controladora. (Page, 9)
- ◆ Figura 10: Opción visión integrada controladora. (Page, 10)
- ◆ Figura 11: Soporte de pared para la cámara. (Page, 11)
- ◆ Figura 12: Soporte de la cámara. (Page, 11)
- ◆ Figura 13: Soporte de la cámara. (Page, 12)
- ◆ Figura 14: Cables desde la cámara hasta el controlador. (Page, 12)
- ◆ Figura 15: Conexionado de la cámara a la controladora. (Page, 13)
- ◆ Figura 16: Soportes de la cinta transportadora. (Page, 13)
- ◆ Figura 17: Opción de sistema de visión en estación. (Page, 14)
- ◆ Figura 18: Ventana Visión en RobotStudio. (Page, 14)
- ◆ Figura 19: Cámara configurada en RobotStudio. (Page, 15)
- ◆ Figura 20: Opciones de la ventana Visión. (Page, 15)
- ◆ Figura 21: Opciones de calibrado. (Page, 16)
- ◆ Figura 22: Opciones de plantilla del calibrado en Grid. (Page, 17)
- ◆ Figura 23: Plantilla grid impresa. (Page, 17)

- ◆ Figura 24: Ajuste previo al calibrado de la cámara. (Page, 18)
- ◆ Figura 25: Eje de coordenadas según plantilla. (Page, 18)
- Figura 26: Resultado de calidad del calibrado. (Page, 19)
- Figura 27: Herramienta del robot. (Page, 19)
- Figura 28: Objeto de calibración. (Page, 20)
- Figura 29: Opciones menú FlexPendant. (Page, 21)
- Figura 30: Opciones menú Datos del programa. (Page, 21)
- Figura 31: Ventana de Robtarget en FlexPendant. (Page, 22)
- Figura 32: Herramienta Patrón PatMax. (Page, 23)
- Figura 33: Ajuste zona búsqueda y modelado. (Page, 23)
- Figura 34: Herramienta creada y ajustada. (Page, 24)
- Figura 35: Ajuste zona búsqueda y modelo redondo. (Page, 24)
- Figura 36: Herramienta creada y ajustada. (Page, 25)
- Figura 37: Herramienta de inspección Leer código 2D. (Page, 25)
- Figura 38: Zona de búsqueda código 2D. (Page, 26)
- Figura 39: Ajustes de herramienta de inspección. (Page, 26)
- Figura 40: Ventana Salidas a RAPID. (Page, 27)
- Figura 41: Definición de datos usados en programa. (Page, 28)
- Figura 42: Activación de la cámara. (Page, 28)
- Figura 43: Preguntas para seguir el proceso de paletizado o acabar con dicho proceso. (Page, 29)
- Figura 44: Disparador cámara y captura de datos. (Page, 30)
- Figura 45: Comparación de datos de la cámara y captura de la pieza. (Page, 31)
- Figura 46: Paletizado de piezas Redondas rojas. (Page, 32)
- Figura 47: Mensajes del proceso en FlexPendant. (Page, 33)
- Figura 48: Movimiento robot y captura de pieza con ventosa. (Page, 33)
- Figura 49: Preguntas sobre continuidad del proceso en FlexPendant. (Page, 34)
- Figura 50: Resultado final con robot en posición HOME. (Page, 36)

- Plano 1: Alimentación y conexiónado cámaras en IRC5. (Page, 38)
- Plano 2: Conexiones de red cámaras a IRC5. (Page, 38)

- Presupuesto 1: Precios según ABB. (Page, 39)
- Presupuesto 2: Precios alternativos. (Page, 39)

7.3. Código RAPID.

El código Rapid y la última copia de la controladora estará compartido mediante una carpeta de Drive que se proporciona a continuación:

<https://drive.google.com/drive/folders/1BQoX3P569cK4VERZwbwMRvCtGHd4ongw?usp=sharing>