



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y AUTOMÁTICA

MÁSTER EN AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

Trabajo Final de Máster

**GUIADO DE UN COBOT CON DISPOSITIVO
EXTERNO PARA LA GRABACIÓN DE
TRAYECTORIAS**

Autor:

Iván Salvador Cambroneró

Director:

D. Martín Mellado Arteché

Valencia, Septiembre 2019

Agradecimientos

Dar las gracias a todas las personas que de una forma u otra han hecho posible la correcta realización de este proyecto, a mi familia y amigos por el apoyo brindado durante todo el tiempo dedicado al desarrollo del proyecto, y en especial a Sara Rabadán, por estar conmigo de principio a fin apoyándome y ayudándome en todo lo posible.

Agradecer especialmente a Martin Mellado como director y a Carlos Catalán por la ayuda ofrecida durante todo el desarrollo del proyecto siempre que la he necesitado, sobre todo en campos de conocimiento desconocidos para mí hasta este momento; así como por la confianza depositada en mí para llevar este proyecto adelante. Muchas gracias por el esfuerzo y la dedicación.

Resumen

El objetivo del actual trabajo es el diseño de un sistema externo compatible con el robot colaborativo UR5e, que permita el guiado completo del robot, así como la grabación de trayectorias y la posterior reproducción de las mismas.

Para la realización del proyecto se ha hecho uso de un robot colaborativo de la empresa Universal Robots, en concreto el modelo UR5e. Para el guiado y la grabación de trayectorias se ha utilizado un dispositivo externo en forma de guante, el cual se ha obtenido mediante impresión 3D. Este dispositivo está conectado directamente a las entradas digitales y analógicas que posee el propio robot y posee diferentes botones para las funciones de las que se ha dotado al dispositivo externo, y un joystick de dos ejes, que permite realizar el guiado del robot.

Para la programación de las funciones que debe realizar el robot se ha utilizado el controlador que proporciona la empresa junto al robot, ya que se trata de una programación sencilla y muy intuitiva para el usuario.

Por último, destacar la correcta realización del proyecto ya que se ha conseguido diseñar un sistema externo que permita guiar y grabar trayectorias en el robot, además de otras funciones añadidas.

Palabras clave: Robot colaborativo, guiado, dispositivo externo

Resum

L'objectiu de l'actual treball és el disseny d'un sistema extern compatible amb el robot col·laboratiu UR5e, que permeta el guiat complet del robot, així com la gravació de trajectòries i la posterior reproducció de les mateixes.

Per a la realització del projecte s'ha fet ús d'un robot col·laboratiu de l'empresa Universal Robots, en concret el model UR5e; i per al guiat i la gravació de trajectòries s'ha utilitzat un dispositiu extern en forma de guant, el qual s'ha obtingut per mitjà d'impressió 3D. Aquest dispositiu està connectat directament a les entrades digitals i analògiques que posseïx el propi robot i posseïx diferents botons per a les funcions de què s'ha dotat al dispositiu extern, i un joystick de dos eixos, que permet realitzar el guiat del robot.

Per a la programació de les funcions que ha de realitzar el robot s'ha utilitzat el controlador que proporciona l'empresa junt amb el robot, ja que es tracta d'una programació senzilla i molt intuïtiva per a l'usuari.

Finalment, destacar la correcta realització del projecte ja que s'ha aconseguit dissenyar un sistema extern que permeta guiar i gravar trajectòries en el robot, a més d'altres funcions afegides.

Paraules clau: Robot col·laboratiu, guiat, dispositiu extern

Abstract

The objective of the current work is the design of an external system compatible with the UR5e collaborative robot, which allows the complete guidance of the robot, as well as the recording of trajectories and the subsequent reproduction of them.

For the realization of the project has been made use of a collaborative robot of the company Universal Robots, in particular the model UR5e. A glove-shaped external device, which has been obtained by 3D printing, has been used for tracking and recording paths. This device is directly connected to the digital and analog inputs that the robot owns and has different buttons for the functions of which the external device has been equipped; and a two-axis joystick, which allows the robot to be guided.

For the programming of the functions to be performed by the robot has been used the controller provided by the company together with the robot, since it is a simple and very intuitive programming for the user.

Finally, highlight the correct realization of the project since it has been possible to design an external system that allows to guide and record trajectories in the robot, as well as other added functions.

Keywords: Collaborative robot, guidance, external device

Índice

Contenido

1.	Introducción	14
1.1.	Antecedentes	14
1.2.	Motivación.....	15
2.	Objeto del proyecto y especificaciones.....	16
3.	Contexto Teórico.....	18
3.1.	Robots tradicionales vs. Robots colaborativos.....	18
3.2.	Robots Colaborativos (Cobots)	21
3.2.1.	Seguridad en los cobots.....	23
3.3.	Normativa	25
4.	Estado del arte	26
4.1.	Robots colaborativos de un brazo.....	26
4.1.1.	ABB.....	26
4.1.2.	Rethink Robotics	27
4.1.3.	KUKA.....	28
4.1.4.	FANUC.....	29
4.1.5.	Universal Robots.....	30
4.2.	Guiado de robots	33
4.2.1.	Guiado mediante visión artificial	33
4.2.2.	Guiado robot UR mediante mando universal.....	34
4.2.3.	Control robot UR basado en visión con <i>OptiTrack Motion Tracking System</i> .35	
5.	Diseño del sistema	37
5.1.	Robot colaborativo UR5e	37
5.2.	Dispositivo externo	40
5.2.1.	Estructura guante de guiado.....	40
5.2.2.	Funcionalidad	42
5.2.3.	Esquema eléctrico	46

5.3.	Programación	48
5.3.1.	Tarea Principal	48
5.3.2.	Hilo composición vector	52
5.3.3.	Hilo lectura botones superiores	59
5.3.4.	Hilo guardar posiciones	61
5.3.5.	Hilo modo fuerza.....	63
6.	Validación del funcionamiento.....	65
7.	Conclusiones	68
8.	Referencias	70
1.	PRESUPUESTO	74
2.	PLANOS	76
3.	CÓDIGO	78

Índice de ilustraciones

Ilustración 1: Evolución de la revolución industrial.....	14
Ilustración 2: Robot industrial y robot colaborativo.....	15
Ilustración 3: Robots industriales en cadena.....	18
Ilustración 4: Robot colaborativo de la empresa KUKA.....	19
Ilustración 5: Niveles de colaboración (Fuente: IFR, 2016)	21
Ilustración 6: Nivel 1 - Coexistencia	21
Ilustración 7: Nivel 2 - Colaboración secuencial.....	22
Ilustración 8: Nivel 3 - Cooperación.....	22
Ilustración 9: Nivel 4 - Colaboración reactiva.....	22
Ilustración 10: Control de velocidad y separación.....	23
Ilustración 11: Guiado manual.....	24
Ilustración 12: Parada de seguridad.....	24
Ilustración 13: IRB 14050 Single-arm YuMi	26
Ilustración 14: Sawyer	27
Ilustración 15: LBR iiwa.....	28
Ilustración 16: CR-14iA/L	29
Ilustración 17: Robots colaborativos UR	31
Ilustración 18: Características robots UR	31
Ilustración 19: Robot Colaborativo YuMi.....	33
Ilustración 20: Marcadores Aruco	34
Ilustración 21: Mando controlador.....	34
Ilustración 22: Servidor MODBUS en el robot	35
Ilustración 23: Marcador y sistema de visión.....	36
Ilustración 24: Robot colaborativo UR5e.....	37
Ilustración 25: Consola de programación	39
Ilustración 26: Estructura guante de guiado.....	40
Ilustración 27: Vista izq. estructura guante.....	41
Ilustración 28: Entradas y salidas UR5e	42
Ilustración 29: Montaje guante guiado.....	43
Ilustración 30: Convertidor DC-DC Lm2596	45
Ilustración 31: Tarea principal.....	48
Ilustración 32: Movimiento del robot.....	49
Ilustración 33: Comando "speedl"	49
Ilustración 34: Parada de emergencia	50
Ilustración 35: Reproducir trayectoria	51

Ilustración 36: Hilo composición vector de movimiento	52
Ilustración 37: Conversión eje X.....	53
Ilustración 38: Conversión valores eje x joystick	53
Ilustración 39: Acondicionamiento valores eje x movimiento lineal	54
Ilustración 40: Acondicionamiento valores eje x movimiento angular.....	54
Ilustración 41: Conversión eje Y	55
Ilustración 42: Conversión valores eje y joystick	56
Ilustración 43: Acondicionamiento valores eje y movimiento lineal	56
Ilustración 44: Acondicionamiento valores eje y movimiento angular.....	57
Ilustración 45: Programación de los movimientos.....	58
Ilustración 46: Hilo lectura modo.....	59
Ilustración 47: Lectura entradas analógicas 4,5 y 6	59
Ilustración 48: Determinación modo funcionamiento	60
Ilustración 49: Hilo guardar posiciones	61
Ilustración 50: Determinación número máximo puntos trayectoria	61
Ilustración 51: Condiciones para guardar posiciones.....	62
Ilustración 52: Grabado de trayectoria	62
Ilustración 53: Hilo modo fuerza.....	63
Ilustración 54: Salida del modo fuerza	64
Ilustración 55: Parada de seguridad.....	64
Ilustración 56: Dispositivo externo de guiado	65
Ilustración 57: Realización de movimientos	66

Índice de tablas

Tabla 1: Comparativa robot industrial vs. robot colaborativo.....	20
Tabla 2: Características IRB 14050 Single-arm YuMi	27
Tabla 3: Características Sawyer	28
Tabla 4: Características LRB iiwa.....	29
Tabla 5: Características robots FANUC.....	30
Tabla 6: Características robots Universal Robots.....	32
Tabla 7: Especificaciones técnicas UR5e.....	39
Tabla 8: Conexiones guante-robot	44
Tabla 9: Conversión valores eje X	53
Tabla 10: Conversión valores eje Y	55
Tabla 11: Presupuesto hardware	74
Tabla 12: Presupuesto maquinaria	74
Tabla 13: Presupuesto mano de obra.....	75
Tabla 14: Resumen del presupuesto.....	75

Trabajo Final de Máster

MEMORIA

Iván Salvador Cambroneró

1. Introducción

En los siguientes apartados se van a exponer los motivos por los que se cree que es importante el trabajo de estudio y se propondrá un sistema externo que permite el guiado de un robot colaborativo.

1.1. Antecedentes

Desde hace unos años, la industria ha ido evolucionado de una forma tan rápida, que se está hablando de la cuarta revolución industrial, la llamada Industria 4.0. Ya no solo se habla solo de automatizar los procesos que forman parte de cada uno de las industrias, sino que la maquinaria que forma parte de esta automatización debe estar conectada entre sí, compartiendo datos en la nube, de forma que todas las máquinas de una misma industria formen una red que permita compartir información.



Ilustración 1: Evolución de la revolución industrial

Una de las partes fundamentales de esta cuarta revolución industrial son los robots que permiten la automatización de los diferentes procesos. Estos son los encargados de la realización de tareas repetitivas, peligrosas y principalmente precisas, de forma que liberan así a las personas encargadas de estas tareas y permitiendo reubicarlos en puestos de trabajo que sean de mayor valor para ellos. De esta forma los operarios que antes realizaban la tarea que con la automatización la realiza el robot, ahora se encargan de tareas como puede ser la programación del robot, la supervisión, el control de calidad, etc.

Estos robots también han sufrido una evolución importante, ya que se ha pasado de utilizar robots industriales pesados y peligrosos, los cuales debían estar enjaulados para evitar cualquier tipo de accidente con los operarios, a utilizar robots colaborativos, los cuales ha sido diseñados con sistemas de seguridad para poder trabajar junto a los humanos, y son en estos últimos en los que se hará especial enfoque en este proyecto.



Ilustración 2: Robot industrial y robot colaborativo

1.2. Motivación

La implantación de robots colaborativos en la industria, además de poder situar a los operarios en otros puestos de trabajo más entretenidos y menos peligrosos, también suponen una ventaja para la empresa. Una de las más importantes es la capacidad que tienen estos robots de trabajar juntos a personas, ya que mediante la implantación de diferentes sensores a lo largo del robot, son capaces de en caso de una colisión con una persona, pararse de inmediato de forma que la integridad del operario no se vea afectada. Otra ventaja principal de estos robots es la precisión que tienen y la repetitividad, ya que son capaces de realizar tareas repetitivas durante mucho tiempo sin perder la precisión.

Todas estas ventajas y más, han hecho que en los últimos años la venta de robots industriales se haya disparado de forma estratosférica llegando a crecer un 114% las ventas entre 2013 y 2017 (*IFR, 2018*). De esta forma se ha hecho necesaria la investigación dentro de los robots colaborativos ya que con el paso de los años las empresas están incorporando este tipo de robots a sus fábricas e invirtiendo en formación de sus operarios para poder trabajar con estos.

2. Objeto del proyecto y especificaciones

Debido al gran avance que se ha producido en la utilización de robots colaborativos para la automatización en las industrias, se plantea este proyecto ante la necesidad de poder utilizar el robot colaborativo UR5e de Universal Robots dentro de cualquier industria donde sea necesario que el robot realice trayectorias complicadas que no pueda “aprender” por sí mismo, de forma que sea el operario de forma intuitiva y sencilla el que “enseñe” al robot, mediante el dispositivo externo, la trayectoria que debe realizar. De esta forma se asegura que la trayectoria que realizará el robot para la tarea encomendada, sea segura y no suponga ningún peligro ni para los propios operarios que trabajan junto al robot ni para la maquinaria ni los productos que se manipulan.

Este proyecto se realiza gracias al Instituto de Automática e Informática Industrial de la UPV (ai2), el cual ha adquirido recientemente el robot colaborativo UR5e con el que se realizará este proyecto. Desde este instituto se está fomentando la utilización de robots colaborativos para diferentes usos tanto de la vida cotidiana de las personas como para tareas industriales.

El dispositivo externo desarrollado permite no tener que utilizar dispositivos externos comerciales los cuales pueden llegar a ser muy caros o incompatibles con el robot. Con este dispositivo se consigue disponer de un sistema de guiado del robot, el cual no tiene un coste elevado y puede desarrollar las mismas funciones que un dispositivo comercial.

Así pues, el dispositivo a desarrollar debe tener las siguientes especificaciones:

- Permitir al usuario la translación del robot en todos los ejes (X,Y,Z)
- Permitir al usuario la rotación del robot en todos los ejes (α,β,γ)
- Permitir al usuario la grabación de una trayectoria y la posterior reproducción de ella.

A parte de las especificaciones de diseño citadas anteriormente, el dispositivo debe cumplir los siguientes requisitos:

- Ser un dispositivo universal que pueda funcionar con cualquier robot que tenga entradas digitales y analógicas.
- Ser viable económicamente.
- Ser un sistema intuitivo y fácil de manejar para el operario
- Ser un sistema seguro para el operario
- Poder utilizar el dispositivo sin ningún sistema auxiliar.

3. Contexto Teórico

A continuación, se realizará un análisis de la evolución de la robótica dentro del mundo de la industria, así como la normativa que deben cumplir los robots.

3.1. Robots tradicionales vs. Robots colaborativos

Desde hace muchos años se llevan utilizando robots para realizar diferentes tareas dentro de la industria, las cuales antes estaban siendo realizadas por operarios, aunque pudieran llegar a ser aburridas o incluso peligrosas. Por esta razón surge la necesidad por parte de las empresas de incluir en sus instalaciones máquinas que puedan realizar estas tareas de forma repetitiva y sin perder precisión, recolocando a los operarios en puestos de trabajo más gratificantes para ellos. Ante esta necesidad surge el robot industrial, el cual es capaz de realizar las funciones que estaba realizando antes el operario de una forma más eficiente, de forma que la empresa se beneficia por el aumento en su productividad.

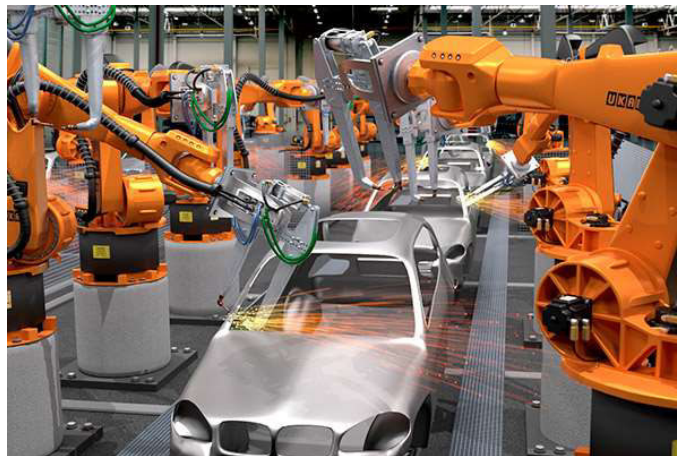


Ilustración 3: Robots industriales en cadena

Un robot industrial se define como:

*“Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”
(ISO/TC 299 Robotics, 2011).*

Es decir, se trata de un robot que se puede programar diversas veces para realizar diferentes tareas, sin necesidad de modificarlo físicamente y el cual posee un brazo mecánico encargado de la manipulación. Estos son los encargados de realizar tareas repetitivas y peligrosas dentro de las fábricas, como pueda ser la soldadura, la pintura o la manipulación de piezas, manipulación de sustancias peligrosas, etc. Es por eso que se trata de robots que poseen una gran precisión en sus movimientos. Sin embargo, se trata de robots muy pesados y muy difíciles de programar, por lo que son robots que son fabricados y diseñados para realizar una tarea en concreto ya que una vez han sido instalados en el puesto de trabajo que van a desempeñar, es muy complicado moverlos hacia otra parte de la fábrica y programarlos para que realicen otra tarea. Es por ello que estos robots se utilizan principalmente en cadenas de fabricación donde el robot no se moverá del puesto de trabajo; además este tipo de robots no pueden trabajar con personas a su alrededor ya que se trata de robots que no tienen ningún tipo de sistema de seguridad mientras realiza su tarea y por lo tanto debido a su intensa actividad y al gran pesaje que tienen, pueden suponer un peligro muy importante para la integridad física de los operarios. Es por ello que este tipo de robots deben estar enjaulados, en el caso de trabajar con operarios alrededor, o en lugares donde no pueda haber interacción con ninguna persona.

Por este motivo, ante la necesidad de puestos de trabajo en los que sea posible que los operarios puedan trabajar conjuntamente con los robots, es por lo que nace el concepto de robot colaborativo o cobot; los cuales han sido diseñados principalmente para trabajar junto a los humanos o incluso interactuando con ellos. Se trata de robots ligeros y fáciles de programar y que principalmente poseen sistemas de seguridad que permiten la parada del robot en el caso de que se produzca cualquier tipo de colisión con el humano.



Ilustración 4: Robot colaborativo de la empresa KUKA

Estos robots han supuesto un gran avance dentro de la automatización industrial y ha sido principalmente en la última década donde se han empezado a extender dentro de las industrias de todo el mundo. Empresas como KUKA, ABB o Universal Robots han invertido mucho dinero en investigación y desarrollo en el mundo de los cobots.

Este tipo de robots, los robots colaborativos, suponen una gran ventaja frente a los robots industriales tradicionales. La más significativa de estas es la que ya se ha comentado con anterioridad, que son capaces de poder trabajar conjuntamente con los humanos sin suponer ningún peligro para estos y sin tener que construir barreras para aislar el robot. Estas barreras no es necesario ponerlas debido a la reducida velocidad a la que trabajan los cobots en comparación con los robots industriales o debido a que, por sensores externos colocados alrededor del robot, se detecta la presencia de una persona y por lo tanto se reduce la velocidad de trabajo del robot hasta que se deja de detectar a la persona.

Otras de las principales ventajas que poseen los cobots con respecto a los robots industriales es la fácil instalación y programación de estos, ya que gracias a las facilidades que los fabricantes aportan a los clientes, estos pueden ser instalados e incluso programados por personas que no sean expertas en programación.

A continuación, se muestra una tabla comparativa entre los robots industriales y los robots colaborativos:

Tabla 1: Comparativa robot industrial vs. robot colaborativo

	Ventajas	Desventajas
Robot Industrial	<ul style="list-style-type: none"> • Gran capacidad de producción • Rápidos • Mueven grandes cargas 	<ul style="list-style-type: none"> • Necesario mucho espacio • Posición Fija • Peligroso • Debe estar aislado • No tiene sistema de seguridad • Muy caros
Robot Colaborativo	<ul style="list-style-type: none"> • Cooperación con humanos • Tiene sistemas de seguridad • Fáciles de instalar • Fácil Programación • Versátil 	<ul style="list-style-type: none"> • Pequeños • Lentos • No puede mover grandes cargas

3.2. Robots Colaborativos (Cobots)

En este apartado se hará una descripción más detallada de los robots colaborativos, ya que se trata del tipo de robot utilizado para este proyecto.

Fueron inventados en 1996 por J. Edward Colgate y Michael Peshkin, los cuales eran profesores de la Universidad de Northwestern. Sin embargo, no fue hasta 2004 donde la empresa alemana KUKA junto al Instituto del Centro Aeroespacial Alemán lanzaron al mercado el primer robot colaborativo, el *LRB 3*. (Wikipedia, 2019). Se trata de robots capaces de trabajar conjuntamente con los seres humanos, factor que ha hecho que en los últimos años se haya incrementado exponencialmente la venta de robots colaborativos sobre todo en pequeñas y medianas empresas donde no se necesita una gran masa de producción y las cuales no poseen grandes fortunas que permitan la adquisición de robots industriales. En la siguiente imagen se pueden observar los diferentes tipos de colaboración entre los robots y los humanos en la industria:

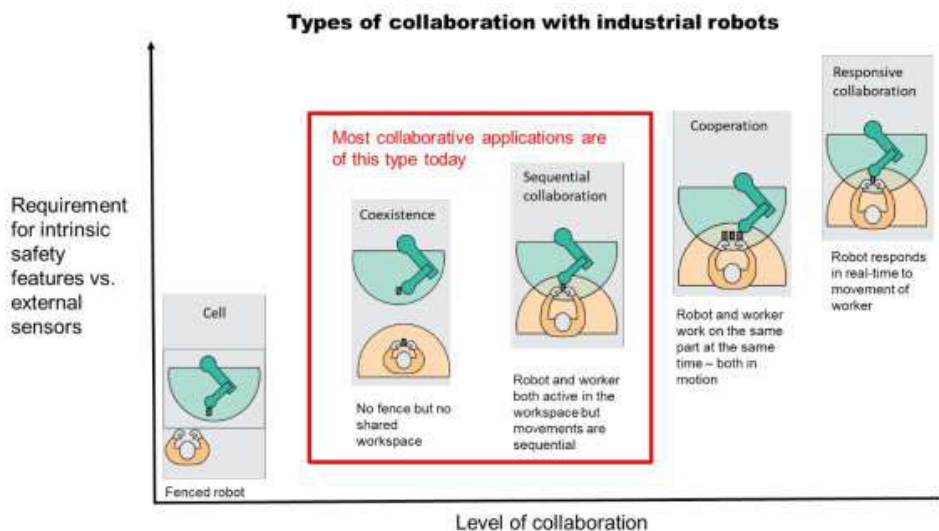


Ilustración 5: Niveles de colaboración (Fuente: IFR, 2016)

En la coexistencia de los cobots con los humanos se pueden diferenciar 4 niveles dependiendo del nivel de cooperación entre el robot y el humano:

- Nivel 1 – Coexistencia: En este nivel el robot y el humano comparten el mismo espacio, pero no entra ninguno de los dos en el área de trabajo del otro, sin necesidad de una valla que envuelva al robot.



Ilustración 6: Nivel 1 - Coexistencia

- Nivel 2 – Colaboración secuencial: En este nivel el robot y el humano trabajan los dos activamente en la misma área de trabajo, pero lo hacen de forma secuencial, es decir, cuando acaba uno de hacer su tarea, empieza el otro y viceversa.

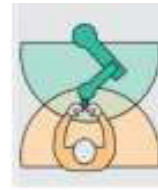


Ilustración 7: Nivel 2 - Colaboración secuencial

- Nivel 3 – Cooperación: En este nivel el robot y el humano trabajan los dos activamente en la misma área de trabajo simultáneamente. De forma que se coordinan realizando movimientos secuenciales para la tarea a realizar.



Ilustración 8: Nivel 3 - Cooperación

- Nivel 4 – Colaboración reactiva: En este nivel el robot y el humano trabajan los dos activamente en la misma área de trabajo, sin embargo, en este caso el robot es capaz de reaccionar a los movimientos del trabajador en tiempo real.



Ilustración 9: Nivel 4 - Colaboración reactiva

Una de las ventajas que proporciona la robótica colaborativa para las pequeñas y medianas empresas, es el acceso a la automatización industrial de las fábricas ya que se puede romper la barrera económica que suponen los robots industriales tradicionales. Por otro lado, la robótica colaborativa también ha penetrado en las empresas grandes, las cuales ya habían automatizado la producción mediante la incorporación de robots industriales a sus fábricas, de forma que estos nuevos robots colaborativos sirven de apoyo a los operarios en sus tareas de forma que permiten que estos no acaben lesionados debido a las tareas repetitivas y en ocasiones pesadas, que deben realizar.

A parte de las características citadas anteriormente, los robots colaborativos destacan por lo siguiente:

- Necesitan menos espacio para ser instalados, ya que las bases de estos robots son más pequeñas que las de los robots industriales, de forma que pueden ser instalados en mesas o incluso en superficies rodantes para poder ser colocados en diferentes partes.

- Se trata de robots de fácil instalación por parte del cliente debido a las facilidades de instalación que proporciona el fabricante, de forma que cualquier persona de la empresa, aunque no sea experta en robótica, puede instalar fácilmente el robot.
- La programación es muy sencilla e intuitiva. El fabricante proporciona junto al robot un dispositivo externo, el cual posee una interfaz muy intuitiva y permite la programación del robot de forma muy sencilla, permitiendo así que cualquier persona pueda realizar una puesta en marcha sencilla del robot sin necesidad de conocer a la perfección la programación de robots.
- Se trata de robots versátiles, ya que debido al poco peso que tienen y a la fácil programación, es muy sencillo cambiar la ubicación y la tarea a realizar por el robot, ya que se pueden volver a reprogramar sin ningún tipo de dificultad pudiendo incluso cambiar la herramienta del final del brazo para poder realizar una tarea específica u otra.

3.2.1. Seguridad en los cobots

Debido a que los robots colaborativos están diseñados para trabajar junto a personas, se deben tener en cuenta algunas medidas de seguridad que han de tener los cobots a la hora de realizar operaciones junto a los humanos:

- Control de velocidad y separación

Se debe reducir el riesgo manteniendo una distancia entre el robot y el humano, de forma que se ha de reducir la velocidad en caso de que se supere la distancia de seguridad establecida o se ha de modificar la trayectoria del robot.

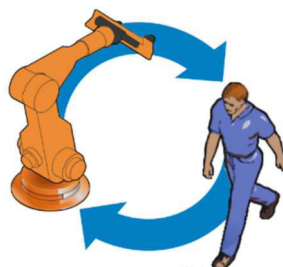


Ilustración 10: Control de velocidad y separación

- Limitación de fuerza

Los contactos entre los humanos y los robots se pueden producir siempre y cuando no supongan un peligro para la integridad física. De esta forma se realiza un análisis de riesgo determinando cuáles son las partes del cuerpo que se pueden ver afectadas y se estiman las fuerzas de contacto.

- Guiado manual

Para hacer posible el guiado manual del robot por parte del operario es necesario que el robot tenga una limitación de ejes y velocidades, y además este tipo de operaciones solo pueden llevarse a cabo presionando previamente un mando de validación.



Ilustración 11: Guiado manual

- Parada de seguridad

El robot debe tener una parada de seguridad de forma que cuando un humano entra en la zona de seguridad delimitada, el robot debe detenerse de forma segura. Una vez el humano haya abandonado la zona de seguridad, el robot debe seguir con su tarea.

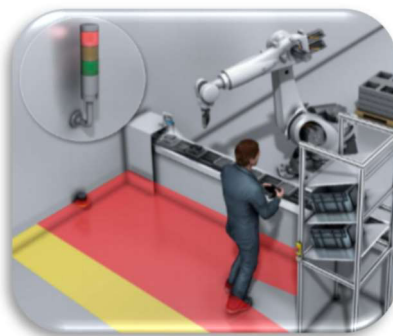


Ilustración 12: Parada de seguridad

3.3. Normativa

En este apartado se numerarán las normas ISO que rigen la interacción con robots industriales y por lo tanto serán las que se deben tener en cuenta a la hora de realizar este proyecto:

ISO 10218-1:2011: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 1: Robots.

En esta norma se especifican los requisitos y las pautas de seguridad aplicables en el momento de manipulación de los robots industriales de forma que se reducen los riesgos asociados a estos peligros. Además, esta norma no contempla el robot como una máquina completa y por lo tanto la emisión de ruido no es considerada como un peligro del robot.

ISO 10218-2:2011: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración.

Esta segunda norma sirve como complemento a la primera ya mencionada. Esta incluye la integración tanto del robot como de los sistemas robot y las células robotizadas, mencionando principalmente el diseño, la fabricación, la instalación, operación, mantenimiento y desmantelamiento de los sistemas robot, y proporciona la información necesaria para la integración. Por otra parte, describe los peligros y las situaciones peligrosas y proporciona requisitos para eliminar o reducir los riesgos.

Debido al gran auge que han tenido los robots colaborativos en los últimos años, ha sido necesario crear una nueva norma que se centre principalmente en este tipo de robots, de esta forma se crea la norma ISO/TC 15066:2016, en la cual se especifican cuáles deben ser los requisitos de seguridad para los robots colaborativos y el entorno de trabajo en el que desarrollan sus funciones dentro de los sistemas industriales.

Por último, mencionar otra norma general para todos los robots industriales y que complementa a las anteriormente mencionadas, la norma ISO 8373:2012, donde se definen los términos de uso en relación a los robots y los dispositivos robóticos que operan tanto en ámbitos industriales como no industriales.

4. Estado del arte

Debido al auge que han tenido los robots colaborativos en los últimos años, diversas empresas han empezado a desarrollar sus productos, por lo que en este apartado se desarrollarán diferentes robots colaborativos de un brazo, ya que es el utilizado para este proyecto, así como diferentes formas existentes de guiado de robots.

4.1. Robots colaborativos de un brazo

Dentro del mercado de los robots colaborativos empresas como ABB, KUKA, FANUC, Universal Robots o Rethink Robotics han sacado al mercado robots colaborativos de un brazo. De esta forma se ha producido una gran rivalidad entre las empresas por ser los primeros en el mercado de los robots colaborativos.

4.1.1. ABB

Se trata de una empresa pionera en cuanto a la fabricación de robots industriales, sin embargo, con la aparición de los cobots se ha abierto una nueva vía de mercado para ellos.

El IRB 14050 Single-arm YuMi se trata de un robot colaborativo que surge como evolución del YuMi de doble brazo presentado en 2015 por esta misma empresa. En este caso se trata de un robot de un brazo cuya característica destacable es que se trata de un robot de 7 ejes. De esta forma es capaz de imitar los movimientos que se realizarían con un brazo humano de una forma más precisa y con más agilidad que los robots de 6 ejes.



Ilustración 13: IRB 14050 Single-arm YuMi

Por otro lado, se trata de un brazo robot muy ligero ya que solamente pesa 9.5kg y que posee una base de dimensiones reducidas, factores que permiten que se trate de un robot muy versátil ya que se puede transportar de una parte a otra del espacio de trabajo sin dificultad alguna. Además, la herramienta está formada por una pinza, una cámara y una válvula de vacío de forma que el agarre y el reconocimiento de las piezas se puede realizar de diversas formas. Sin embargo,

este robot tiene la desventaja de únicamente poder manipular objetos con su herramienta de un peso máximo de 500 gramos, de forma que las aplicaciones industriales quedan muy limitadas.

Tabla 2: Características IRB 14050 Single-arm YuMi

	IRB 14050 Single-arm YuMi
Alcance	556mm
Ejes	7
Carga máxima	500g
Peso	9.5kg

4.1.2. Rethink Robotics

Se trata de una empresa alemana la cual únicamente se encarga de la fabricación de robots colaborativos entre los que destaca el Sawyer.

El Sawyer es el único robot colaborativo de esta empresa. Se trata de un robot presentado en 2015 como una evolución más pequeña y flexible que el primer robot presentado por esta empresa, el Baxter. Este robot destaca principalmente porque posee 7 grados de libertad, es decir que al igual que el IRB 14050 permite imitar los gestos de los humanos de una forma más ágil y precisa que los robots colaborativos de 6 ejes. Por otra parte, se trata de un robot bastante más pesado que el anterior, ya que en este caso el peso es de 19kg por lo que es bastante más complicado de mover que el anterior. Sin embargo, puede estar sujeto a un carrito proporcionado por el fabricante de forma que el transporte del robot se hace mucho más sencillo a pesar de su peso.



Ilustración 14: Sawyer

Por otra parte, posee en la herramienta una pinza para poder manipular diferentes objetos y una cámara que permite localizar estos objetos y manipularlos. En cuanto a la carga máxima de este robot, está por encima de la del robot IRB 14050 comentado anteriormente, ya que este tiene una carga máxima de 4kg. Pero sin duda el aspecto diferenciador de este robot es la pantalla que tiene anclada al eje de la base, la cual permite visualizar diferentes parámetros del robot e incluso se puede programar para que muestre información deseada para el usuario.

Tabla 3: Características Sawyer

	Sawyer
Alcance	1260mm
Ejes	7
Carga máxima	4kg
Peso	19kg

4.1.3. KUKA

Esta empresa al igual que ABB es especializada en la fabricación de robots industriales, pero que debido al auge de los robots colaborativos también se han interesado en la fabricación de estos.

El LBR iiwa es el robot colaborativo de esta empresa, el cual está disponible en dos versiones dependiendo de la capacidad de carga que es capaz de aguantar, 7 y 14kg, y del alcance máximo que posee el brazo, 800 y 820mm. Al igual que los dos robots anteriormente mencionados, se trata de un robot de 7 ejes por lo que es capaz de imitar los movimientos de un brazo humano de una forma más precisa y ágil que el resto de robots de 6 ejes.



Ilustración 15: LBR iiwa

Se trata de un robot cuyas principales características es que tiene capacidad de aprendizaje, es decir, el robot puede aprender trayectorias y coordenadas cuando se le muestra la posición de destino deseada. Además posee diferentes sensores a lo largo del robot lo que lo hacen un robot muy sensitivo que es capaz de adecuar la fuerza que realiza al tipo de pieza que se está manipulando.

En cuanto al peso de estos robots colaborativos, el LRB iiwa 7 tiene un peso aproximado de 24kg y el LRB iiwa 14 tiene un peso aproximado de 30kg, por lo que se tratan de robots pesados dentro del mundo de los robots colaborativos. Por otro lado la herramienta del LRB iiwa, al contrario de los dos robots comentados anteriormente, no tiene ninguna herramienta específica, sino que tiene un agarre para poder acoplar cualquier tipo de herramienta, ya sea una herramienta comercial o una herramienta fabricada por el cliente, de forma que puedes adaptar la herramienta necesaria al tipo de aplicación que se quiere llevar a cabo.

Tabla 4: Características LRB iiwa

	LRB iiwa	
	LRB iiwa 7	LRB iiwa 14
Alcance	800mm	820mm
Ejes	7	7
Carga máxima	7kg	14kg
Peso	24kg	30kg

4.1.4. FANUC

Esta empresa, al igual que ABB y KUKA, estaba especializada en la fabricación de robots industriales, pero en el momento en el que comenzaron a surgir los robots colaborativos, se interesaron también por este tipo de robots.

En la parte de los robots colaborativos, FANUC ha crecido mucho ya que con los años ha ido fabricando diversos robots, llegando a tener a día de hoy 5 modelos diferentes, muy por encima de la competencia. En este caso todos los robots fabricados por FANUC poseen únicamente 6 ejes, uno menos que los que ofrecen las empresas comentadas anteriormente.



Ilustración 16: CR-14iA/L

En cuanto a la carga que pueden soportar, los diferentes modelos que hay disponibles permiten poder elegir el robot que más se adecúe a las necesidades industriales ya que la carga soportada va desde los 4kg en el caso del robot más pequeño, hasta los 35kg en el caso del robot de más tamaño. Por otro lado, al igual que la carga máxima soportada, el alcance que tiene el robot también varía

en función del modelo del robot, el cual está comprendido entre los 550mm que ofrece el modelo más pequeño, hasta los 1813mm que ofrece el más grande.

En el caso de este robot, al igual que en el IRB iiwa de KUKA comentado anteriormente, el efecto final del robot no tiene ninguna herramienta adherida a él, sino que se trata de un efector final plano con el fin de poder colocar la herramienta que se desee, ya sea comercial o de fabricación propia del cliente, con el fin de hacerlo más versátil ya que se puede intercambiar una herramienta por otra dependiendo de la aplicación que vaya a realizar el robot. Por otro lado se trata de robots muy pesados ya que el peso de estos está comprendido entre los 48kg en el caso del robot más pequeño, y lo 990kg en el caso del robot más grande; por lo que se trata de robots muy poco versátiles a la hora de cambiar el puesto de trabajo donde están colocados.

Tabla 5: Características robots FANUC

	CR-4iA	CR-7iA	CR-14iA	CR-15iA	CR-35iA
Alcance	550mm	717/911mm	911mm	1441mm	1813mm
Ejes	6	6	6	6	6
Carga máxima	4kg	7kg	14kg	15kg	35kg
Peso	48kg	53/55kg	55kg	255kg	990kg

4.1.5. Universal Robots

Por último, se va a realizar un análisis de los robots colaborativos de Universal Robots ya que es la empresa que ha fabricado el robot que se va a utilizar en este proyecto. Universal Robots es una empresa danesa encargada de la fabricación únicamente de robots colaborativos siendo una de las punteras en el sector de los cobots.

Actualmente posee en el mercado tres modelos diferentes de robot colaborativo, el UR3e, el UR5e y el UR10e. Estos robots se diferencian principalmente en el radio de trabajo máximo que tienen y en la carga máxima que pueden soportar. Al igual que los robots de la empresa FANUC, estos también poseen únicamente 6 ejes, lo que los sitúa por debajo de la competencia en este aspecto.

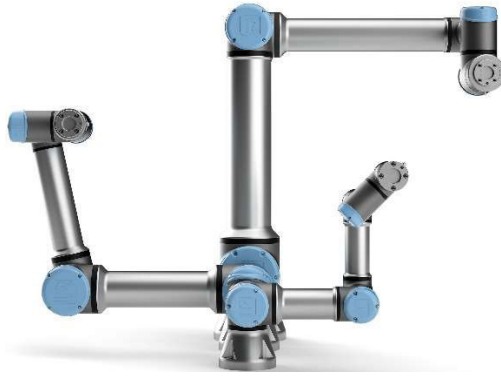


Ilustración 17: Robots colaborativos UR

Debido a la variedad de modelos que posee Universal Robots, se puede elegir el modelo que más se adecúe a la tarea que se desea realizar, ya que la carga máxima que puede soportar el robot está comprendida entre los 3kg que puede soportar el UR3e y los 10kg que puede soportar el UR10e, habiendo un robot intermedio que puede soportar 5kg, que es el caso del UR5e. Por otro lado, estos robots también se diferencian en el alcance máximo al que puede llegar el brazo robot, ya que para en el caso del robot más pequeño, el UR3e, este tiene un radio máximo de trabajo de 500mm; en el caso del modelo intermedio, el UR5e, el radio de trabajo máximo es de 850mm; y por último en el caso del modelo más grande, el UR10e, el radio de trabajo máximo es de 1300mm.

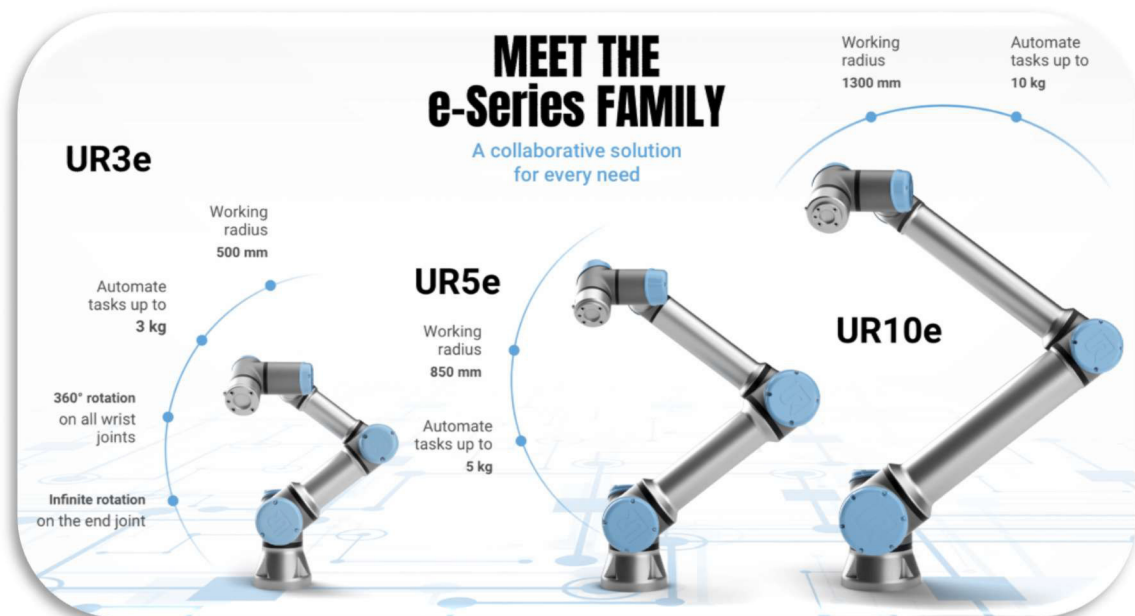


Ilustración 18: Características robots UR

El efector final de estos robots, al igual que los de FANUC y KUKA, no tienen herramienta de serie, sino que la empresa te proporciona diversas herramientas que son compatibles con el efector del robot, ya que posee una entrada de señales con la que puede comunicar el robot con la herramienta; además también se puede integrar herramientas fabricadas por el cliente de forma que se adapta la herramienta a la tarea que debe realizar el robot, haciendo así que el robot sea muy versátil.

Por otro lado, se trata de robots no especialmente pesados, ya que el UR3e pesa aproximadamente 11kg, el UR5e tiene un peso de 18.4kg y el UR10e pesa alrededor de 33.5kg, de forma que en el caso del UR3e y el UR5e son más manejables y por lo tanto pueden desempeñar tareas en diferentes espacios de trabajo, sin suponer un esfuerzo excesivo el traslado del robot.

Tabla 6: Características robots Universal Robots

	UR3e	UR5e	UR10e
Alcance	500mm	850mm	1300mm
Ejes	6	6	6
Carga máxima	3kg	5kg	10kg
Peso	11kg	18.4kg	33.5kg

Para el proyecto se ha optado por utilizar el robot UR5e ya que se trata de un robot no muy pesado, que puede manipular cargas de 5 kilos, lo que se supone suficiente para la aplicación que se va a desarrollar. Por otra parte, un factor que ha tenido bastante peso a la hora de elegir el robot, ha sido la gran variedad de herramientas que tiene disponibles Universal Robots para situar en el efector final del robot, de forma que la aplicación desarrollada se puede utilizar tanto para un pick and place donde haría falta una garra, hasta una tarea de manipulación de objetos frágiles donde se tenga que utilizar otra herramienta diferente a la garra.

4.2. Guiado de robots

La otra parte importante del proyecto que se está realizando, es el guiado del robot colaborativo, en mi caso, mediante un dispositivo externo en forma de guante. Sin embargo, en la actualidad se han realizado diversos estudios y proyectos en los que se ha realizado el guiado de formas diferentes.

4.2.1. Guiado mediante visión artificial

Este proyecto se trata del desarrollo de una aplicación para poder teleoperar con sistemas robóticos utilizando visión artificial. De este modo se realiza una interacción directa entre el operario y el robot, ya que lo que se busca en este proyecto es que el robot imite los gestos que realiza el operario con sus brazos.

En este caso se ha utilizado el robot colaborativo YuMi de la empresa ABB, que posee dos brazos articulados, cada uno con 7 ejes, de forma que se puede imitar el movimiento del brazo humano a la perfección, ya que posee en total 14 grados de libertad, el cual está disponible en los laboratorios del Instituto de Automática e Informática Industrial.



Ilustración 19: Robot Colaborativo YuMi

Para poder hacer efectiva la interacción natural entre el robot y el operario se ha utilizado un sistema de visión artificial. De esta forma la cámara detecta los movimientos y la posición de los brazos del humano y manda esta información al

robot, de forma que la unidad de control del robot interpreta la información que le llega procedente de la cámara, y la reproducirá en los brazos del robot.

Para la detección correcta de la posición de los brazos del operario se han utilizado unos marcadores que el operario se sitúan en ambas manos del operario, de forma que la cámara detecta la posición inicial de estos marcadores en un plano definido y sigue la trayectoria que van realizando, guardando los puntos por los que ha pasado y enviándoselos al robot para que vaya realizando esos mismos movimientos en tiempo real. (Jiva, 2019)

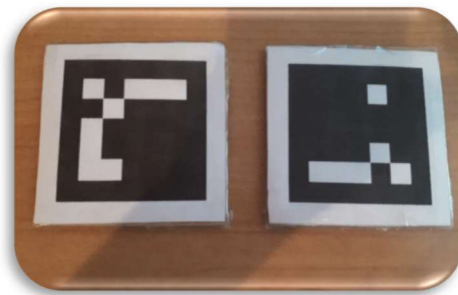


Ilustración 20: Marcadores Aruco

4.2.2. Guiado robot UR mediante mando universal

En este caso el proyecto realizado se trata del control de un robot UR mediante un mando externo universal conectado al ordenador.



Ilustración 21: Mando controlador

Para el control del robot se dispone de un programa en Python, el cual se está ejecutando en el ordenador y es el encargado de recibir la información enviada por el mando de todos los botones que se están pulsando y escribir esta información en un servidor MODBUS que permitirá el robot leer esta información.

Por otro lado, en el robot se está ejecutando un programa encargado de leer la información que le llega por el servidor MODBUS, interpretar esa información y de esta forma poder calcular la siguiente posición a la que debe ir el robot, la cual viene dada por la posición de los joysticks del mando y posteriormente realizar el movimiento a esa nueva posición.

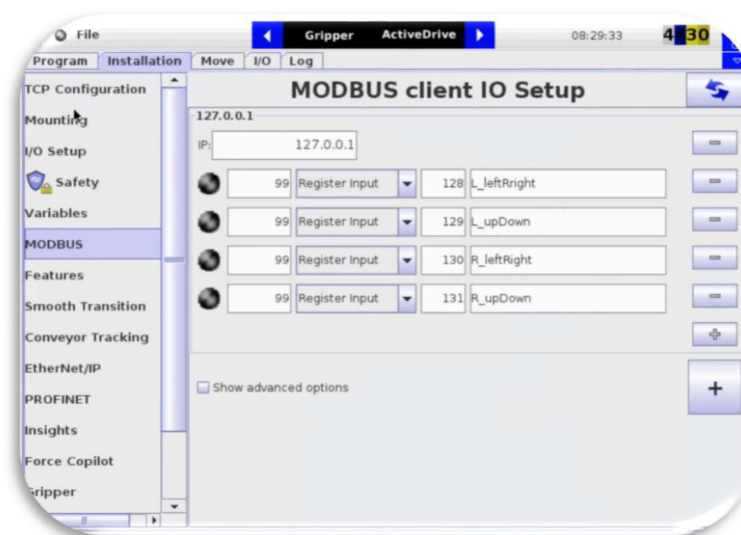


Ilustración 22: Servidor MODBUS en el robot

Tal y como se puede observar en el mando, este proyecto únicamente permite el movimiento de orientación de la herramienta, ya que solo es posible la modificación de cada uno de los ángulos de la herramienta y del movimiento de traslación del robot en el eje Z. Por otra parte, para la comunicación con el robot es necesaria la utilización de un ordenador, de forma que si no se dispone de ordenador en ese momento o se pierde la comunicación con el robot, la aplicación no funciona.

4.2.3. Control robot UR basado en visión con *OptiTrack Motion Tracking System*.

Este sistema de guiado se trata de un sistema de visión con un objetivo similar al proyecto de visión con el robot YuMi explicado anteriormente.

Para este proyecto se utiliza un sistema comercial de seguimiento de movimientos que está formado por 3 cámaras con iluminación infrarroja, un marcador que será el que llevará el operario y por lo tanto el encargado de realizar los movimientos

que posteriormente serán imitados por el robot, y un software propio del sistema que es el encargado de recibir la información de las cámaras.



Ilustración 23: Marcador y sistema de visión

Posteriormente la información obtenida por las cámaras infrarrojas es enviada al software del sistema el cual posteriormente lo envía al sistema de control del robot UR, que procesa la información y la reproduce mediante el movimiento del robot en tiempo real.

Debido a que se trata de un sistema comercial, es un sistema muy caro y además es necesario aprender a utilizar el software que la empresa proporciona junto al sistema de seguimiento de movimientos.

5. Diseño del sistema

En este apartado se describirá el robot utilizado para el proyecto a fondo, así como el dispositivo externo utilizado para su guiado y la programación necesaria para comunicar el dispositivo con el robot.

5.1. Robot colaborativo UR5e

Para la realización de este proyecto se ha utilizado el robot colaborativo UR5e de la empresa Universal Robots. La selección del robot se ha realizado después de haber comparado diferentes robots colaborativos de distintas empresas, comparando principalmente la carga máxima que pueden manipular, el peso del propio robot y la compatibilidad de diferentes herramientas para su efector final. Como resultado de la comparativa, se ha concluido que el robot que más de adecuía a las especificaciones del proyecto es el UR5e.



Ilustración 24: Robot colaborativo UR5e

Se trata de un robot colaborativo de Universal Robots. Esta empresa es una de las pioneras en el mundo de los robots colaborativos y se podría decir que la más importante dentro de este sector. Han trabajado mucho en el desarrollo de los robots colaborativos hasta conseguir que la nueva gama de robots (e-Series) sea polivalente, fácil de programar y fácilmente integrable, factores que han hecho que empresas que buscan la automatización de sus procesos, se hayan fijado en ellos.

Por otro lado, el robot utilizado para este proyecto es el de tamaño medio de la gama e-Series. Se trata de un robot muy flexible y ligero, aspectos que lo dotan de una versatilidad muy grande. A pesar de no ser el de mayor tamaño, este también puede desempeñar tareas de automatización de hasta 5kg, que puedan resultar ser peligrosas y repetitivas. Este robot supone un gran avance a la hora de realizar tareas como por ejemplo de pick and place, ya que de esta forma se libera a los operarios para que estos realicen otros procesos de la producción, ya que, según su director ejecutivo, el UR5e es capaz de realizar en apenas 4 horas tareas que, si estuvieran hechas por operarios, tardarían 2 o 3 días.

Además de lo comentando anteriormente, este robot posee 5 competencias diferenciadoras frente al resto de robots colaborativos de la competencia:

- **Fácil de programar:** Se trata de un robot que puede ser programado por cualquier operario sin ser necesarios conocimientos previos de programación de robots. Además, Universal Robots posee una la “*Universal Robots Academy*” en la que están disponibles de forma gratuita diferentes cursos tanto online como presenciales sobre la programación de sus robots, de forma que cualquier operario puede convertirse en programador de robots.
- **Instalación rápida:** Al igual que para la programación del robot, en este caso tampoco hace falta la presencia de un experto para la puesta en marcha del robot, sino que cualquier operario puede desembalar el robot y montarlo. Gracias a su intuitiva interfaz en pocos minutos se puede configurar el robot y realizar la primera tarea.
- **Flexible:** Debido al poco peso que tiene este robot y al poco espacio que ocupa, es realmente sencillo poder reubicar el robot dependiendo de las tareas a realizar necesarias. Además, gracias a la gran cantidad de efectores finales disponibles, se puede utilizar un mismo robot para diferentes tareas, únicamente cambiando el efector final.
- **Colaborativo y seguro:** Es capaz de realizar tareas repetitivas, peligrosas o incluso aburridas, llegando incluso a poder trabajar junto a los operarios sin ningún tipo de peligro para estos, debido a la gran cantidad de sistemas de seguridad que posee, desde el sensor de fuerza y par en la última articulación, hasta las diferentes funciones internas de seguridad que posee.

- **Amortización rápida:** Gracias a la posibilidad de que el robot pueda estar trabajando de forma continua constantemente sin ningún impedimento, aumentando así la productividad. Además, no es necesario ningún gasto en cuanto a vallado de seguridad o coste de programación.

Además de todas estas características generales que posee el robot, también cabe destacar algunas de las especificaciones técnicas más importantes, las cuales se recogen en la siguiente tabla:

Tabla 7: Especificaciones técnicas UR5e

	UR5e
Rango	50N - 10Nm
Resolución	2.5N – 0.04Nm
Precisión	4N – 0.3Nm
Carga	5Kg
Alcance	850mm
Grados de libertad	6 articulaciones giratorias
Repetibilidad	+/- 0.03mm

Como última característica a destacar de este robot, es que viene con una consola de programación con una pantalla táctil de 12 pulgadas con la interfaz de usuario *PolyScope* la cual es muy intuitiva y hace mucho más sencilla la tarea de programación al operario que no tiene conocimientos de programación.

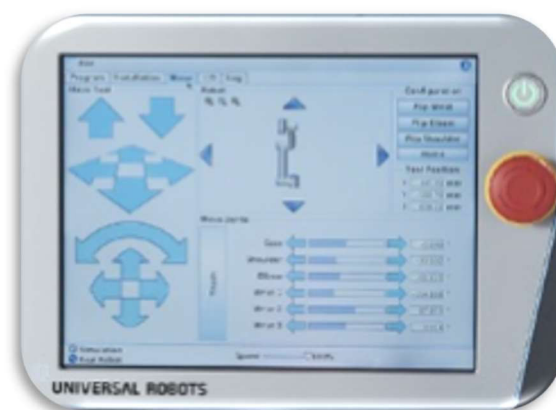


Ilustración 25: Consola de programación

Además, este robot posee tanto entradas y salidas digitales como analógicas, en concreto, 8 entradas digitales, 8 salidas digitales, 2 entradas analógicas y 2 salidas analógicas. De esta forma, la comunicación del robot con periféricos externos tanto de salida como de entrada es muy sencilla.

5.2. Dispositivo externo

La parte más importante del proyecto es el diseño y fabricación del dispositivo externo que se va a utilizar para el guiado del robot y las dos funciones auxiliares que son objetivo del trabajo.

Lo primero que se ha de comentar es que el sistema de guiado debe ser un sistema que no necesite de ningún equipo auxiliar, de forma que no dependa del buen funcionamiento de otro equipo, sino de sí mismo. De esta forma se ha optado por utilizar directamente las entradas analógicas y digitales de las que dispone el robot, de forma que el dispositivo externo irá directamente conectado a estas entradas.

5.2.1. Estructura guante de guiado

En el caso de este proyecto no se ha optado por el típico mando universal con dos joysticks para mover el robot, sino que se ha diseñado una pieza en forma de guante de forma que el operario la podrá adaptar a la perfección a su mano para así resultar mucho más cómoda a la hora de realizar el guiado del robot.



Ilustración 26: Estructura guante de guiado

Tal y como se puede observar en la *Ilustración 26* la estructura que forma el guante está formada por una estructura principal, la cual es la parte más ancha del guante, que permitirá que el guante una vez puesto no se mueva y quede totalmente sujeto a la mano del operario, ya que esta parte se moldeará a la medida del esta.

Esta estructura tiene una obertura en la parte izquierda de la misma, tal y como se puede observar en la *Ilustración 27*, la cual hace posible la introducción del dedo pulgar por ese hueco, permitiendo así que el guante se quede sujeto a la mano. Cabe destacar que el diseño realizado de la estructura del guante es para situar este en la mano derecha del operario.



Ilustración 27: Vista izq. estructura guante

Tal y como se puede observar en esta misma imagen, en la parte de la obertura para introducir el dedo pulgar, se sitúa un soporte rectangular que será el encargado de anclar el joystick que permitirá el guiado completo del robot.

Por otro lado, además de la estructura principal que se adapta tanto a la palma de la mano como a la parte superior de esta, se dispone de cuatro estructuras alargadas que van sujetas a la estructura principal, que corresponden a los cuatro dedos de la mano que permitirán realizar los diferentes movimientos programados.

Estas estructuras alargadas, al igual que la estructura principal, también se moldearán según las preferencias del operario, pudiendo dejarlas completamente rectas o doblándolas hasta que él desee.

Toda esta estructura ha sido diseñada mediante el software de diseño CAD SolidWorks, teniendo en cuenta que la pieza diseñada posteriormente se va a imprimir mediante material 3D.

La impresión 3D de las piezas se ha realizado en los laboratorios del ai2 (Instituto de Automática e Informática Industrial), utilizando la impresora BCN3D Sigma R17. Para ello se ha utilizado el material PLA, el cual se utiliza en muchas ocasiones para la impresión de piezas 3D debido a la gran variedad de colores que posee, así como a la poca cantidad de gases perjudiciales que desprende a la hora de imprimir.

5.2.2. Funcionalidad

Una vez diseñada la estructura en forma de guante, se procede a estudiar la forma de resolver el problema de comunicación entre el guante y el robot.

Tal y como se ha comentado anteriormente, en este caso se ha optado por una conexión directa desde el guante hasta el robot, utilizando las diferentes entradas analógicas y digitales que posee. De esta forma directamente manipulando el guante se obtendrá una respuesta en el robot, que posteriormente se utilizará para realizar los movimientos pertinentes del robot.

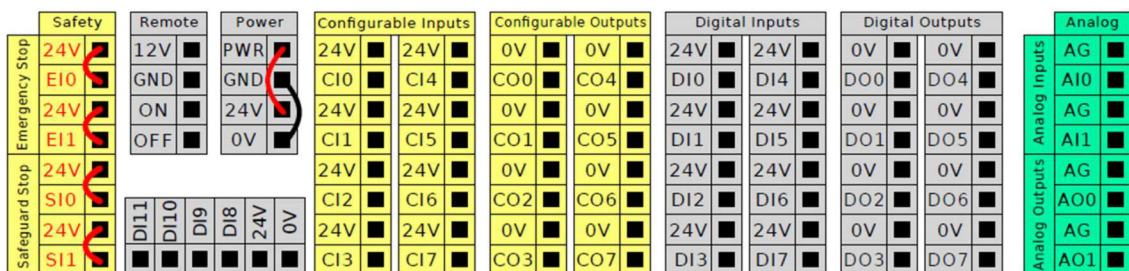


Ilustración 28: Entradas y salidas UR5e

Para conseguir comunicar el guante con el robot se ha decidido utilizar los siguientes componentes:

- 4 botones en los dedos: Cada uno de estos botones será el encargado de elegir el tipo de movimiento que se desea que el robot realice.
- 3 botones parte superior: Estos botones tienen cada uno una función: el primero de ellos es el encargado de comenzar a grabar la trayectoria, el segundo de ellos es el encargado de realizar la trayectoria previamente grabada, y el último de ellos es el encargado de desbloquear el robot cuando se produce una parada de emergencia.

- Joystick parte izquierda: Se trata de un joystick de dos ejes, el cual permite comunicar al robot en qué dirección debe moverse.

Todos estos componentes están situados en el guante tal y como se muestra en la *Ilustración 29*.

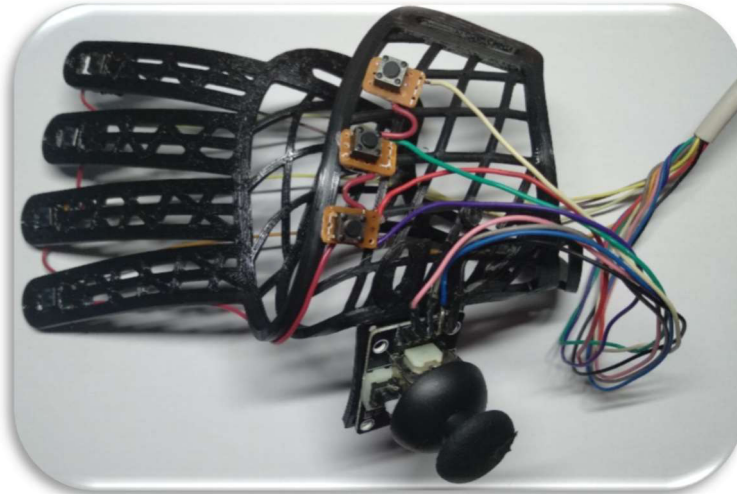


Ilustración 29: Montaje guante guiado

Cada uno de los botones está formado por dos terminales, uno de ellos irá conectado a la entrada digital correspondiente, y el otro terminal irá conectado a la alimentación, en este caso está conectado a 24V, los cuales se pueden extraer directamente del propio robot. De esta forma cuando el botón no esté pulsado, en la entrada digital habrá un 0 lógico y cuando esté pulsado habrá un 1 lógico, de forma que posteriormente mediante la programación en el robot se programarán cada uno de los comportamientos.

A continuación se muestra una tabla donde se pueden observar las distintas conexiones que existen entre el guante de guiado y el robot, especificando cada uno de los botones a que entrada digital está conectado, así como cada uno de los ejes en los que se puede mover el joystick, a que entradas analógicas está conectado.

Tabla 8: Conexiones guante-robot

Guante	Robot
Botón dedo índice	Entrada digital 0
Botón dedo corazón	Entrada digital 1
Botón dedo anular	Entrada digital 2
Botón dedo meñique	Entrada digital 3
Botón 1 parte superior	Entrada digital 4
Botón 2 parte superior	Entrada digital 5
Botón 3 parte superior	Entrada digital 6
Joystick eje X	Entrada analógica 0
Joystick eje Y	Entrada analógica 1

Por otra parte, se puede observar el joystick. Este está formado por 5 terminales, el primero de ellos es el común, el cual irá conectado a la tierra del robot, el siguiente de ellos es el de alimentación, que en este caso debe ser de 5 voltios, el siguiente se denomina *VRX* este terminal se trata de un terminal analógico, el cual irá conectado a la entrada analógica 0 del robot. Es el encargado de dar a conocer en qué posición está situado el joystick en el eje X, obteniendo como respuesta 0 voltios si el joystick está en la máxima posición negativa, 2.5 voltios si el joystick está situado en el centro y 5 voltios si el joystick está situado en la máxima posición positiva.

El siguiente terminal que se encuentra es el terminal *VRX*, el cual, al igual que *VRX*, se trata de un terminal analógico que en este caso irá conectado a la entrada analógica 1 del robot. Este es el encargado de dar a conocer la posición del joystick en el eje Y, obteniendo como resultado 0 voltios si el joystick está en la máxima posición negativa de este eje, 2.5 voltios si el joystick está en la posición central y 5 voltios si el joystick está en la máxima posición positiva de este eje.

Por último, se encuentra el botón *SW* que se trata de un botón que tiene el propio joystick que en este caso no se va a utilizar.

Debido a que el joystick se ha de alimentar a 5 voltios, y todas las salidas del robot son de 24 voltios, ha sido necesaria la utilización de un convertidor DC-DC de tipo Buck (*Ilustración 30*) para reducir los 24 voltios que se obtienen del robot, a los 5 voltios necesarios para alimentar el joystick. Este convertidor, se trata de

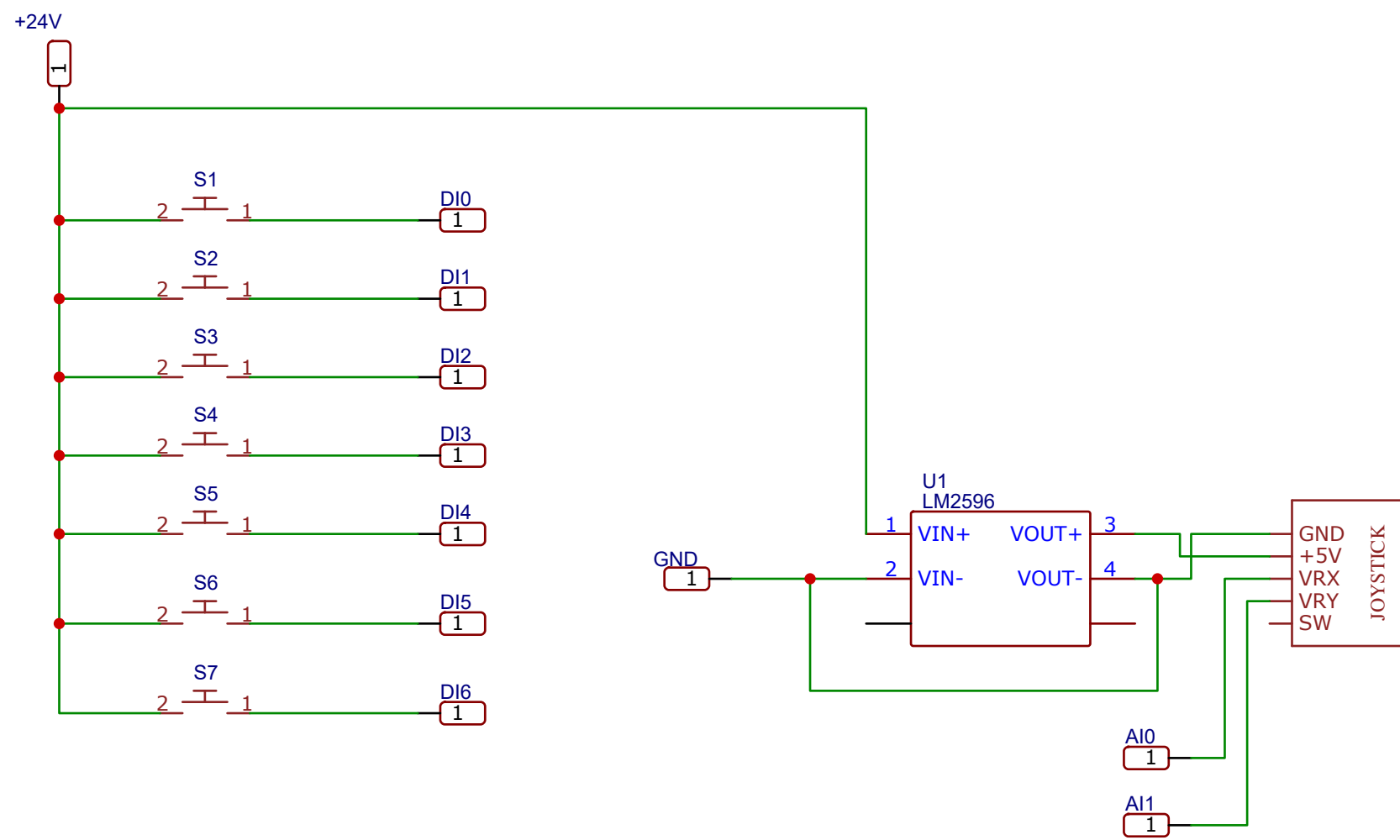
un reductor de voltaje variable de forma que mediante un potenciómetro que está integrado en la propia placa, se puede reducir el voltaje al valor deseado dentro de unos límites.




Ilustración 30: Convertidor DC-DC Lm2596

De esta forma la entrada del convertidor vendrá directamente de los 24 voltios del robot y del terminal común de este, y la salida, previamente ajustado el potenciómetro, irá conectada directamente a la alimentación del joystick y al terminal común de este.

5.2.3. Esquema eléctrico



TITLE: Esquemático guante de guiado		REV: 1.0
	Company: UPV	Sheet: 1/1
	Date: 2019-08-07	Drawn By: Ivan Salvador

5.3. Programación

La parte más importante del proyecto es la programación. Ya que mediante la programación del robot se podrá realizar una correcta interpretación de la información obtenida tanto de los botones como del joystick del guante, pudiendo posteriormente realizar los movimientos correctos del robot sin que se produzca ningún fallo.

Para ello se ha realizado el siguiente programa utilizando la interfaz de usuario *Polyscope*, la cual viene instalada en el controlador del robot y resulta muy sencilla e intuitiva.

5.3.1. Tarea Principal

Esta primera parte del código de programación corresponde a las líneas donde realmente se realizarán todos los comandos necesarios para el movimiento del robot.

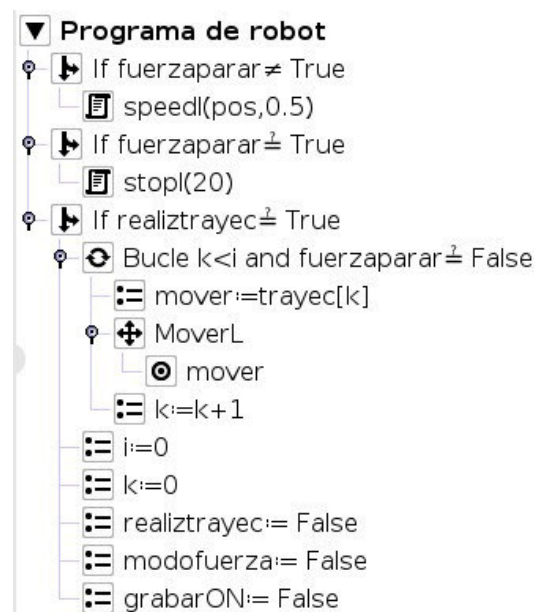


Ilustración 31: Tarea principal

Para explicar la funcionalidad que tiene esta primera tarea, se irá explicando línea a línea que se realiza en cada una, así como la explicación de los comandos utilizados.

Lo primero que se observa en la tarea es una condición. De esta forma, si la variable *fuerzaparar* no tiene un valor *True*, es decir, no ha sido activada en el hilo correspondiente (se explicará posteriormente), el robot ejecutará la siguiente línea.

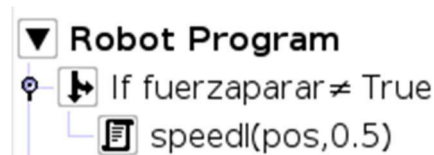


Ilustración 32: Movimiento del robot

Aquí se encuentra el primer comando propio del robot: *speedl*. Este comando se define como la velocidad de la herramienta, de forma que cuando se hace la llamada a esta función, el robot acelera linealmente hasta alcanzar la velocidad deseada de cada uno de los ejes.

speedl(xd, a, t, aRot=' a')

Ilustración 33: Comando "speedl"

Este comando necesita como parámetros los siguientes datos:

- **xd:** La velocidad de la herramienta en cada uno de los ejes. (m/s)
- **a:** La aceleración de la herramienta en cada posición. (m/s²)
- **t:** El tiempo antes de que la función se vuelva a ejecutar.
- **aRot:** La aceleración de la herramienta. (rad/s²)

En este caso la velocidad de la herramienta en cada uno de los ejes está definida en la variable *pos* la cual se actualiza constantemente gracias al hilo que se ejecuta en paralelo y que obtiene tanto los valores de los botones correspondientes a cada uno de los dedos, como el valor de cada uno de los ejes del joystick. Posteriormente se interpreta toda esta información y se colocan los valores correspondientes en cada una de las 6 posiciones del vector *pos*. El vector *pos* está formado por 6 posiciones ya que son las necesarias para definir completamente el movimiento del robot: translación en X, Y, Z y rotación en X, Y, Z. ([x, y, z, rx, ry, rz]).

En cuanto a la aceleración de la herramienta, ya que no se dispone de ninguna especificación que requiera una aceleración en concreto, se ha optado por utilizar la aceleración que viene propuesta en los manuales, la cual es de 0.5 m/s^2 .

Los dos últimos parámetros no son de obligada utilización, ya que el tiempo de retorno no hace falta ya que como constantemente se está modificando el valor del vector *pos* de forma que no se necesita que la herramienta alcance la velocidad que se define, sino que se utiliza este comando ya que los movimientos son mucho más suaves que en el caso de utilizar *MoveL* o *MoveJ*.

La siguiente comprobación que se realiza en el programa es si la variable *fuerzaparar* en este caso ha sido activada. En el caso de que haya sido activada, se hace una llamada a la función *stopl*, la cual decelera la velocidad de la herramienta hasta que es cero.

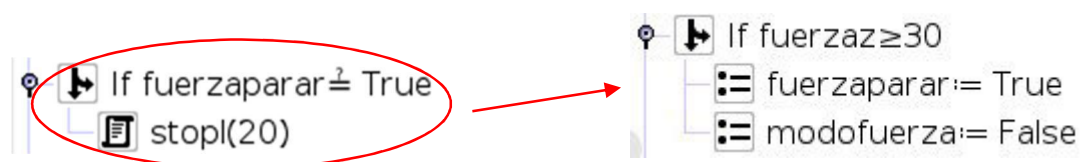


Ilustración 34: Parada de emergencia

En este caso el parámetro necesario para el correcto funcionamiento únicamente es la deceleración que se desea que tenga la herramienta. Al igual que en el caso de la anterior función, no hay ninguna especificación que requiera una deceleración en concreto, por lo que también se ha optado por utilizar el valor que viene dado en los manuales, el cual es 20 m/s^2 .

De esta forma el programa se quedará en esa instrucción enclavado, de forma que no se podrá hacer ningún movimiento con el robot hasta que se presione el botón correspondiente para desenclavar el robot del estado de paro en el que se encuentra.

A continuación, se vuelve a realizar otra comprobación; en este caso se comprueba si la variable *realiztrayec* ha sido activada. Esta variable se activará en el correspondiente hilo, que será el que lea las entradas digitales correspondientes a los botones situados en la parte superior de la mano.

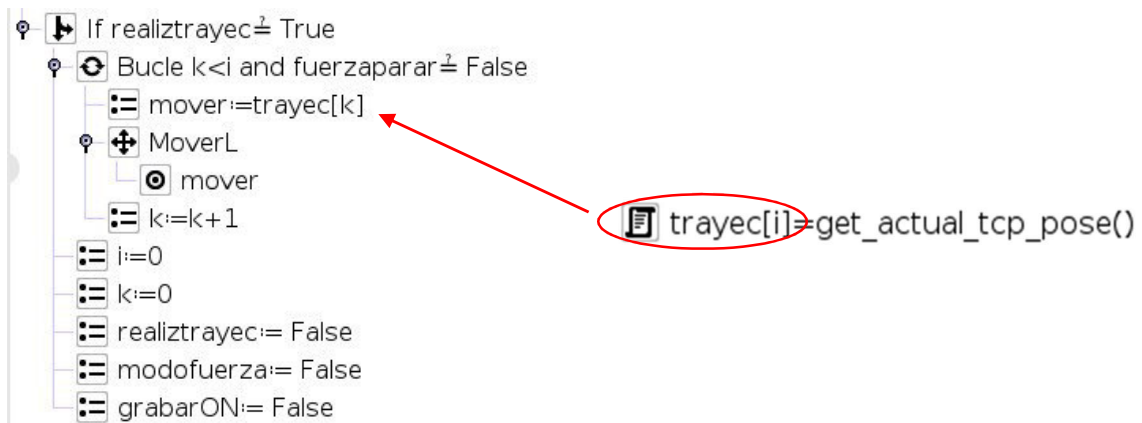


Ilustración 35: Reproducir trayectoria

Si ha sido activada, el programa entrará en un bucle, del cual no saldrá siempre y cuando la variable k sea menor que la variable i y la variable $fuerzaparar$ no esté activada. En este bucle se realizará el movimiento lineal (*MoveL*) de cada uno de los puntos guardados en la variable *trayec*. En esta variable se guardan los puntos por los que va pasando el robot en el caso de que se pulse el botón correspondiente a la grabación de trayectoria. Una vez ya se ha realizado el movimiento el movimiento correspondiente hasta el siguiente punto, se incrementa el valor de la variable k para indicar que se ha de pasar al siguiente punto de la variable *trayec* para realizar el movimiento lineal hasta este.

La variable k corresponde a las posiciones de las cuales ya se ha realizado el movimiento lineal, y la variable i corresponde al número de posiciones de la trayectoria que se han guardado. Una vez la variable k es igual o mayor a la variable i , es decir, ya se han realizado todos los movimientos lineales de la trayectoria que se había guardado, se procede a reestablecer tanto el valor de k como de i para poder realizar posteriormente otra grabación y reproducción de la trayectoria. A parte de estas dos variables, también se reestablecen las dos variables correspondientes a la grabación de trayectoria y la reproducción de esta.

5.3.2. Hilo composición vector

En este primer hilo del programa, se realiza la lectura de las dos entradas analógicas del robot, la cuales obtienen las señales del eje x y del eje y del joystick. También se realiza la lectura de las cuatro primeras entradas digitales, la cuales corresponden a los botones situados en cada uno de los dedos del guante.

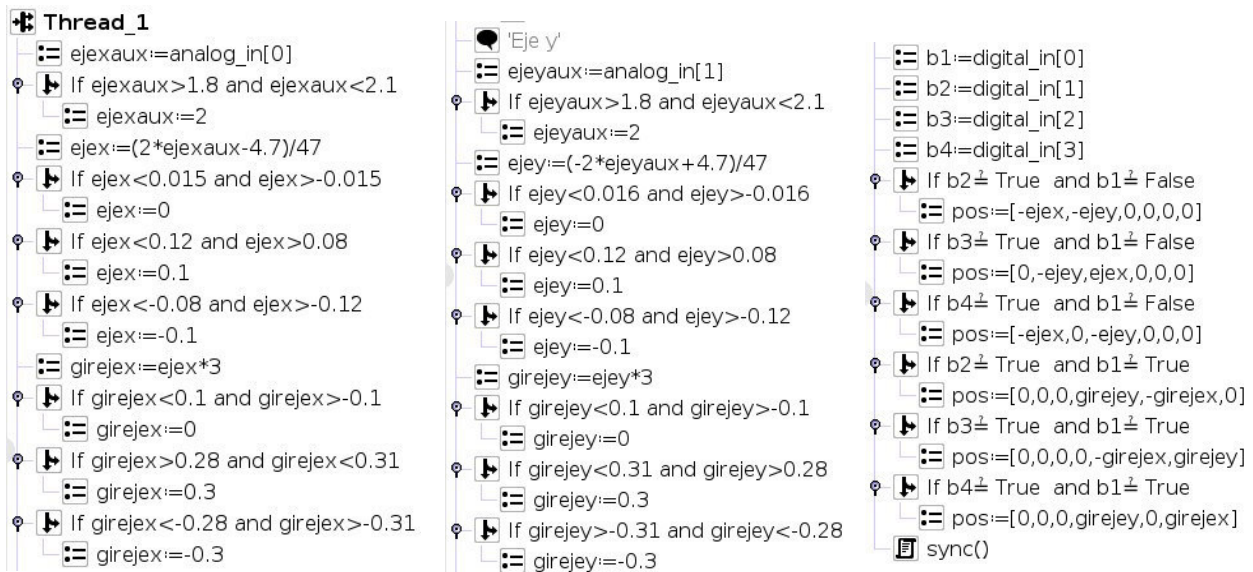


Ilustración 36: Hilo composición vector de movimiento

Las primeras instrucciones que se observan son las correspondientes al eje X del joystick. El joystick está alimentado a 5V, por lo que cuando está situado en la máxima posición positiva del eje X, se obtienen los 5V a los que está alimentado; si está situado en la máxima posición negativa, devuelve 0V; y si está situado en el centro, se obtiene 2.5V.

Para poder realizar un movimiento con el robot, utilizando los valores obtenidos del joystick se ha de realizar un factor de conversión, ya que se desea que cuando el joystick esté en la parte negativa del eje X, el robot se mueva en la dirección contraria a cuando el joystick está en la parte positiva; y además se desea que cuando el joystick esté en el centro, el robot no se mueva.

Para ello se ha hecho uso de las siguientes correspondencias:

Tabla 9: Conversión valores eje X

X (Reales)	Y (Deseadas)
0	-4,7
2,35	0
4,7	4,7

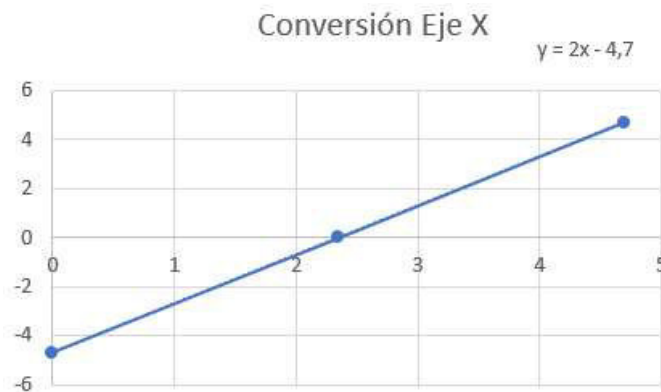


Ilustración 37: Conversión eje X

De esta forma las primeras líneas del código correspondiente a este hilo son las encargadas de leer el valor que hay en la entrada analógica 0, correspondiente al eje X del joystick, y realizar la conversión correspondiente, de forma que en la variable *ejex* se obtiene un valor entre -0.1 y 0.1.

```
ejexaux:=analog_in[0]
If ejexaux>1.8 and ejexaux<2.1
  ejexaux:=2
ejex:=(2*ejexaux-4.7)/47
```

Ilustración 38: Conversión valores eje x joystick

Idealmente, los valores que se debería obtener del eje X del joystick una vez realizada la conversión serían los siguientes:

- -0.1 → Joystick situado máxima posición negativa
- 0 → Joystick situado en el centro
- 0.1 → Joystick situado máxima posición positiva

Sin embargo, esto no ocurre ya que debido a la tolerancia de los componentes que forman el módulo del joystick, los valores analógicos obtenidos no son precisos, sino que tienen un pequeño error. Este hecho se ha podido solucionar mediante programación, ya que se han realizado unas asignaciones al valor de la variable *ejex* observando entre que valores variaba el valor de la entrada analógica en cada una de las posiciones deseadas del joystick, tal y como se muestra en el siguiente recorte del código.

```
☐ If ejex<0.015 and ejex>-0.015
  ☐ := ejex:=0
☐ If ejex<0.12 and ejex>0.08
  ☐ := ejex:=0.1
☐ If ejex<-0.08 and ejex>-0.12
  ☐ := ejex:=-0.1
```

Ilustración 39: Acondicionamiento valores eje x movimiento lineal

De esta forma ya se obtiene un valor deseado en cada una de las posiciones.

Se ha optado por utilizar un valor de 100 mm/s (valor de la variable *ejex*), ya que no se deseaba que los movimientos del robot fueran muy veloces ni bruscos.

Continuando con el código, en esta parte se realiza la asignación de la velocidad de giro deseada para el movimiento del robot. En este caso se ha optado por la utilización de una velocidad de 0.3 rad/s, la cual después de muchas pruebas se ha decidido que es la velocidad en la que más suaves son los movimientos de giro del robot.

Al igual que para la velocidad lineal, en este caso también se ha realizado la asignación del valor mediante unos rangos entre los que variaba la variable *girejex*, de forma que se obtiene el valor deseado para cada posición del joystick.

```
☐ := girejex:=ejex*3
☐ If girejex<0.1 and girejex>-0.1
  ☐ := girejex:=0
☐ If girejex>0.28 and girejex<0.31
  ☐ := girejex:=0.3
☐ If girejex<-0.28 and girejex>-0.31
  ☐ := girejex:=-0.3
```

Ilustración 40: Acondicionamiento valores eje x movimiento angular

De esta forma ya se han asignado los valores correspondientes tanto al movimiento lineal como angular obtenido del eje X del joystick.

La segunda parte del código es la correspondiente a la adecuación del valor obtenido en la entrada analógica 1, correspondiente al eje Y del joystick.

En este caso, cuando el joystick está situado en la máxima posición positiva del eje Y, se obtienen 0V; si está situado en la máxima posición negativa, devuelve los 5V a los que está alimentado; y si está situado en el centro, se obtiene 2.5V.

Para poder realizar un movimiento con el robot, utilizando los valores obtenidos del joystick se ha de realizar un factor de conversión, ya que se desea que cuando el joystick esté en la parte negativa del eje Y, el robot se mueva en la dirección contraria a cuando el joystick está en la parte positiva; y además se desea que cuando el joystick esté en el centro, el robot no se mueva.

Para ello se ha hecho uso de las siguientes correspondencias:

Tabla 10: Conversión valores eje Y

X (Reales)	Y (Deseadas)
0	4,7
2,35	0
4,7	-4,7



Ilustración 41: Conversión eje Y

De esta forma, las primeras líneas correspondientes al eje Y, son las que realizan la conversión del valor obtenido por la entrada analógica, obteniendo un valor entre -0.1 y 0.1. Este valor, al igual que en el eje X, será el utilizado para el movimiento lineal del robot.

```

'Eje y'
:= ejeyaux:=analog_in[1]
If ejeyaux>1.8 and ejeyaux<2.1
  := ejeyaux:=2
:= ejey:=(-2*ejeyaux+4.7)/47

```

Ilustración 42: Conversión valores eje y joystick

Idealmente, los valores que se debería obtener del eje X del joystick una vez realizada la conversión serían los siguientes:

- 0.1 → Joystick situado máxima posición positiva
- 0 → Joystick situado en el centro
- -0.1 → Joystick situado máxima posición negativa

Sin embargo, al igual que ocurría con el eje X, debido a la tolerancia de los elementos utilizados en el módulo del joystick, es necesario realizar la asignación del valor deseado mediante rangos. Esto es lo que se ha realizado en las siguientes líneas de código.

```

If ejey<0.016 and ejey>-0.016
  := ejey:=0
If ejey<0.12 and ejey>0.08
  := ejey:=0.1
If ejey<-0.08 and ejey>-0.12
  := ejey:=-0.1

```

Ilustración 43: Acondicionamiento valores eje y movimiento lineal

De esta forma ya se obtiene un valor deseado en cada una de las posiciones.

Se ha optado por utilizar un valor de 100 mm/s, al igual que el eje X, ya que no se deseaba que los movimientos del robot fueran muy veloces ni bruscos.

En el caso del valor correspondiente a los giros obtenidos del eje Y del joystick, se ha optado por la utilización de una velocidad de 0.3 rad/s, la cual después de muchas pruebas se ha decidido que es la velocidad en la que más suaves son los movimientos de giro del robot.

Al igual que para la velocidad lineal, en este caso también se ha realizado la asignación del valor mediante unos rangos entre los que variaba la variable *girejey*, de forma que se obtiene el valor deseado para cada posición del joystick.

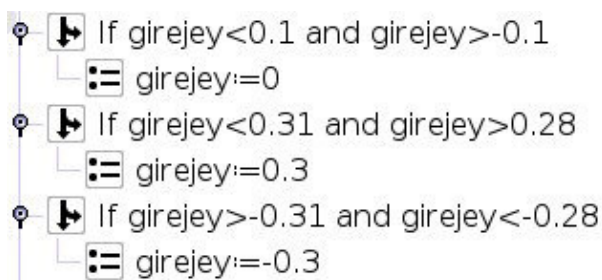


Ilustración 44: Acondicionamiento valores eje y movimiento angular

De esta forma ya se han asignado los valores correspondientes tanto al movimiento lineal como angular obtenido del eje Y del joystick.

La última parte del código del primer hilo es la correspondiente a la lectura de las cuatro primeras entradas digitales, las cuales corresponden a los botones situados en los dedos del guante.

Además de la lectura de las entradas digitales, se ha programado los movimientos, tanto lineales como angulares, en los diferentes ejes que realizará el robot dependiendo de qué botones se hayan pulsado.

- Dedo corazón pulsado → Movimiento lineal en los ejes XY
- Dedo anular pulsado → Movimiento lineal en los ejes YZ
- Dedo meñique pulsado → Movimiento lineal en los ejes XZ
- Dedos índice y corazón pulsados → Movimiento angular en los ejes XY
- Dedos índice y anular pulsados → Movimiento angular en los ejes YZ
- Dedos índice y meñique pulsados → Movimiento angular en los ejes XZ

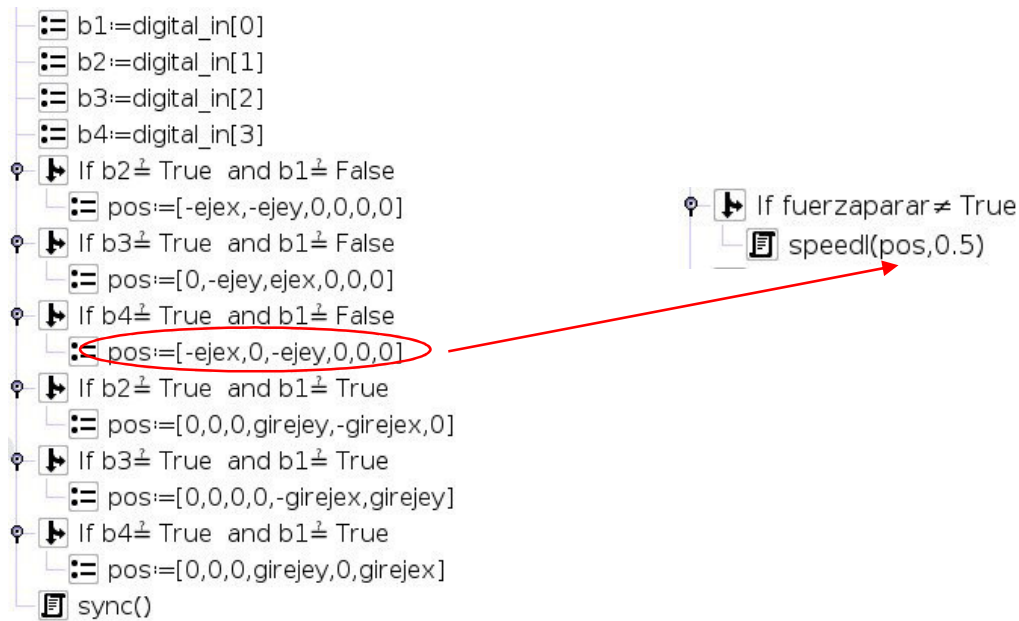


Ilustración 45: Programación de los movimientos

Cabe destacar que se han programado los movimientos del robot de forma que sea lo más intuitivo posible para la persona que utilice el sistema; de modo que los movimientos del robot coincidirán con el movimiento que se realice mediante el joystick.

Para el movimiento del robot, se han programado los movimientos teniendo en cuenta los ejes correspondientes a la base de este, ya que como no se va a utilizar una herramienta en concreto, lo más sencillo es realizar todos los movimientos con respecto a los ejes de la base.

Por último se realiza la llamada a la función *sync()*, la cual permite sincronizar todos los hilos que se están ejecutando de forma que se comiencen a ejecutar todos a la vez. La utilización de esta línea es muy importante ya que hay hilos más densos en cuanto a programación y que por lo tanto necesitan más tiempo para ejecutarse que otros, de forma que mediante la función *sync()*, todos los hilos se sincronizan y se ejecutan al mismo tiempo.

5.3.3. Hilo lectura botones superiores

Este segundo hilo es el encargado de leer constantemente el valor de las entradas digitales 4, 5 y 6, que son las correspondientes a los botones situados en la parte superior de la mano.

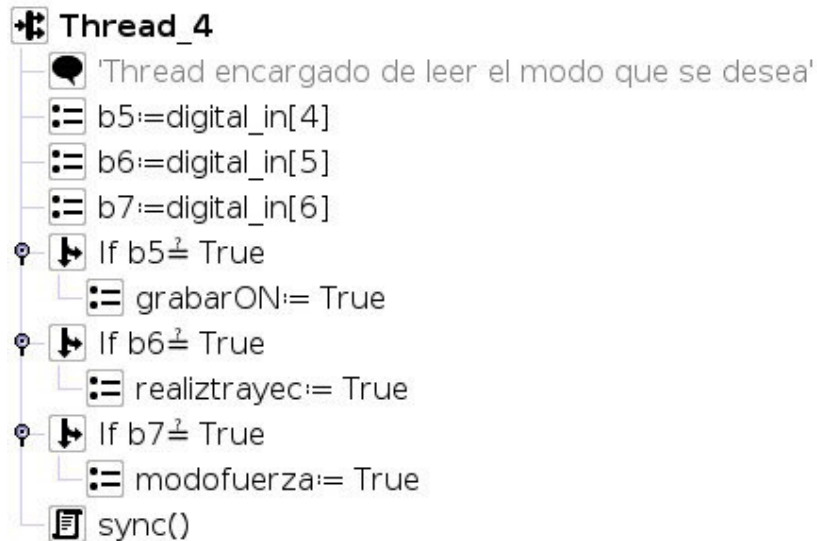


Ilustración 46: Hilo lectura modo

Las primeras instrucciones que se observan en la programación de este hilo son las correspondientes a la lectura de las entradas digitales. Para esta lectura se ha utilizado la función propia del robot *digital_in(port)*, la cual devuelve un valor de “False” en el caso en el que en la entrada haya un “0” lógico, o un valor de “True” en el caso de que en la entrada haya un “1” lógico. Debido a la continua ejecución de los hilos y a la rapidez de ejecución, cualquier cambio en cualquiera de las entradas se puede detectar en tiempo real.

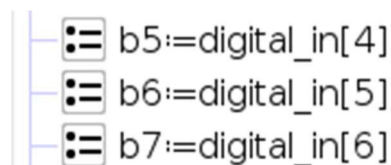


Ilustración 47: Lectura entradas analógicas 4,5 y 6

Cabe comentar que a la entrada digital 4 está conectado el botón correspondiente a la grabación de trayectoria, en la entrada digital 5 está conectado el botón correspondiente a la realización de la trayectoria, y en la entrada digital 6 está conectado el botón correspondiente a la liberación del robot de la situación de paro.

Mediante las siguientes instrucciones es posible conocer cuándo se ha pulsado cada uno de los botones correspondientes a este hilo, ya que cuando se pulsa uno de los botones, la variable correspondiente a ese botón cambia su valor y pasa a valer "True".

Para ello se han utilizado funciones condicionales, de forma que cuando la variable *b5*, es decir, el primer botón de la parte superior, cambie su valor a "True", se activará la variable *grabarON*, de forma que el hilo encargado de la grabación (hilo 3) comenzará a guardar las posiciones por las que se está moviendo la herramienta.

Por otro lado, si el que se ha pulsado es el segundo botón de la parte superior de la mano, *b6*, se activará la variable *realiztrayec* de forma que el programa principal entrará en el bucle para ir realizando punto a punto la trayectoria grabada anteriormente.

En el caso de que se haya pulsado el tercer botón de la parte superior de la mano, es decir el *b7*, se activará la variable *desbloqueo*, de forma que en el hilo encargado del modo fuerza, se desactivarán las variables *fuerzaparar* y *realiztrayec*, y se podrá seguir con el guiado del robot.

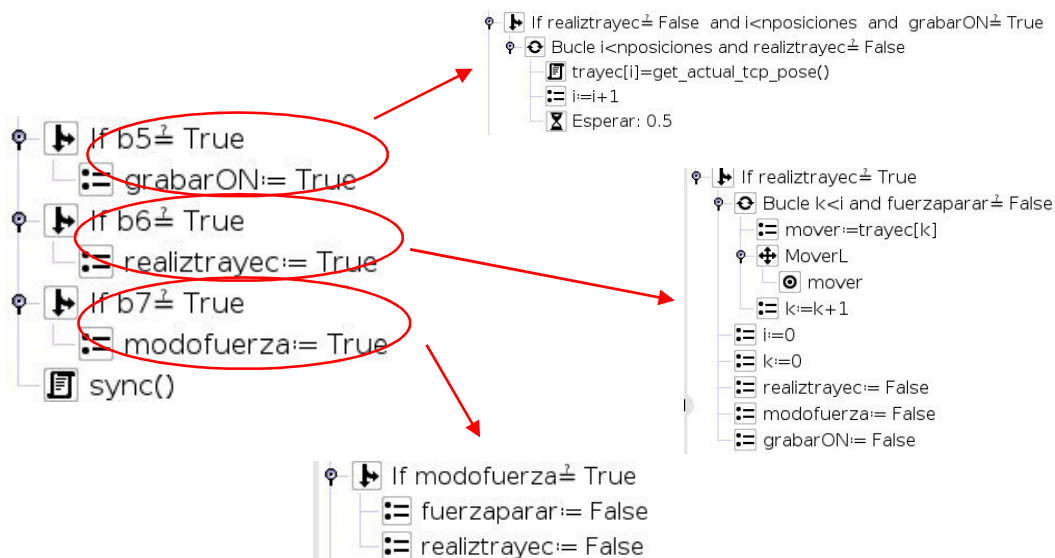


Ilustración 48: Determinación modo funcionamiento

Por último, al igual que en el primer hilo, se ha utilizado al final de este la función `sync()`, la cual permite hacer una espera hasta que el resto de hilos acaben de ejecutarse, de modo que se realiza una sincronización entre ellos permitiendo así que todos comiencen a ejecutarse al mismo tiempo.

5.3.4. Hilo guardar posiciones

Este tercer hilo es el encargado de realizar el guardado de cada una de las posiciones por las que va pasando el robot cuando se presiona el botón correspondiente al grabado de trayectoria.

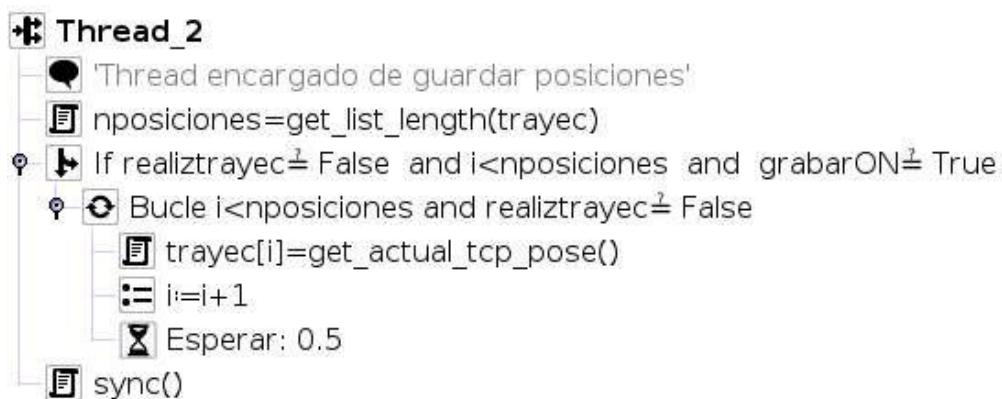


Ilustración 49: Hilo guardar posiciones

Lo primero que se realiza en este hilo es la definición de la variable `nposiciones` la cual obtendrá el valor que devuelva la función `get_list_length(trayec)`. Esta función devuelve la longitud que tiene una lista variable. En este caso devuelve la longitud que tiene el vector `trayec`, el cual está definido al principio del programa.

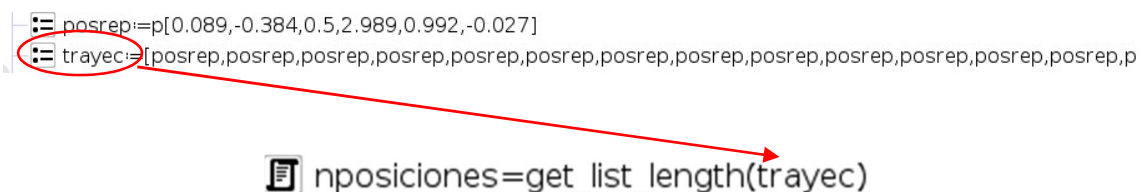


Ilustración 50: Determinación número máximo puntos trayectoria

El vector *trayec* se trata de un vector de 50 posiciones, que será el número máximo de posiciones que se pueden guardar para la trayectoria.

Este, inicialmente tiene unos valores de seguridad que vienen dados en la variable *posrep*. Estos valores son posiciones de las articulaciones que se suponen de seguridad.

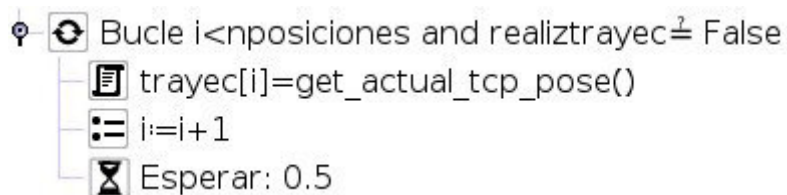
A continuación, la siguiente instrucción que aparece en el hilo es una condición, de forma que el programa seguirá a la siguiente línea siempre que la variable *realiztrayec* no se haya activado, la variable *i* sea menor que la variable *nposiciones* y que la variable *grabarON* se haya activado.



```
If realiztrayec == False and i < nposiciones and grabarON == True
```

Ilustración 51: Condiciones para guardar posiciones

Si todas estas condiciones se cumplen entonces se pasa a un bucle, del cual no se saldrá mientras la variable *i* sea menor que la variable *nposiciones* y la variable *realiztrayec* no se haya activado, es decir, no se saldrá hasta que se hayan guardado las 50 posiciones de la trayectoria o se haya pulsado el botón correspondiente a realizar la trayectoria.



```
Bucle i < nposiciones and realiztrayec == False
├── trayec[i]=get_actual_tcp_pose()
├── i:=i+1
└── Esperar: 0.5
```

Ilustración 52: Grabado de trayectoria

Una vez el programa está dentro del bucle, lo primero que se realiza es guardar la posición y orientación de la herramienta, mediante la función *get_actual_tcp_pose()*, en la posición *i* del vector *trayec*. Una vez se ha guardado la posición, se incrementa el valor de la variable *i* para poder guardar el siguiente punto en la siguiente posición del vector y por último se realiza una espera de 0.5 segundos. Esta espera se realiza ya que es necesario esperar entre el guardado de una posición y la siguiente, ya que la velocidad de procesamiento del robot es mucho mayor que la velocidad que puede tener el operario a la hora de mover el

joystick, por lo que si no se realiza una espera, las posiciones guardadas son muy similares y por lo tanto se pierde un número considerable de posiciones disponibles para guardar la trayectoria lo que conlleva en una disminución del tiempo máximo que se puede estar guardando la trayectoria.

Una vez se han completado el número máximo de puntos que se puede guardar o se ha decidido realizar la trayectoria, se sale del bucle previamente explicado y el programa pasa a la última línea del hilo. En esta última línea, al igual que los hilos anteriores, se ha utilizado la función *sync()*, la cual permite hacer una espera hasta que el resto de hilos acaben de ejecutarse, de modo que se realiza una sincronización entre ellos permitiendo así que todos comiencen a ejecutarse al mismo tiempo.

5.3.5. Hilo modo fuerza

Este hilo se trata del último del programa. En él se realizan todas las instrucciones correspondientes al modo fuerza que se ha programado; tanto el cálculo de la fuerza ejercida sobre la herramienta, como las instrucciones que liberan al robot del modo fuerza.

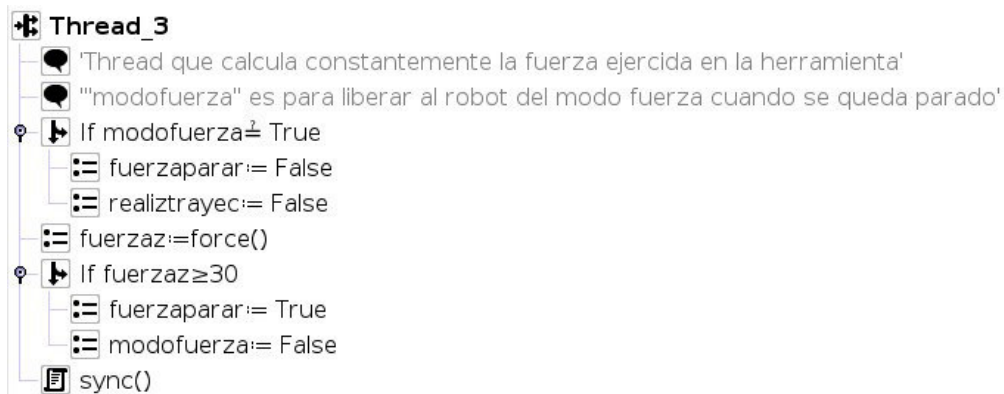


Ilustración 53: Hilo modo fuerza

Lo primero que se observa en el hilo es una condición mediante la cual, si el modo seleccionado es el 3, es decir, si se ha pulsado el botón correspondiente a la liberación del robot del modo fuerza; entonces tanto la variable *fuerzaparar* como la variable *realiztrayec* se ponen a un valor de “False”. De esta forma, en el programa principal ya se permite seguir realizando movimientos, y en el caso de que la fuerza se haya realizado durante la realización de la trayectoria

previamente grabada, esta se reseteará de forma que el operario deberá volver a grabar la trayectoria o volver a presionar el botón correspondiente para poder volver a ejecutar la trayectoria previamente grabada.

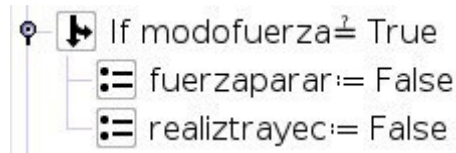


Ilustración 54: Salida del modo fuerza

Por último se hace uso de la función disponible en el robot *force()* la cual permite conocer la fuerza ejercida (Newtons) en la herramienta en todo momento. De esta forma se ha procedido a guardar este valor en una variable *fuerzaz*, de forma que en la posterior instrucción se vuelve a utilizar un condicional. Si la fuerza ejercida en la herramienta supera los 30 Newtons, la variable *fuerzaparar* cambiará su valor a “True” y por lo tanto el robot quedará en estado de paro hasta que se presione el botón para reanudar el movimiento.

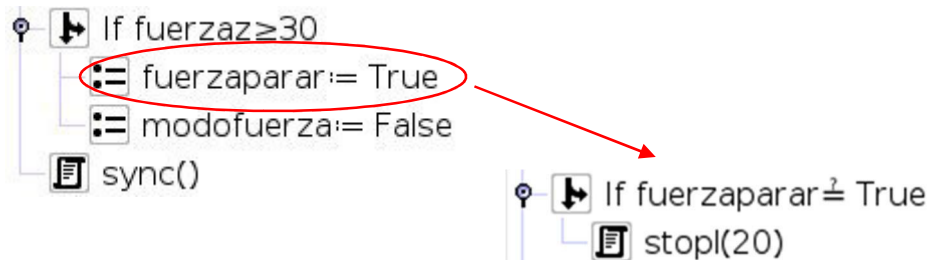


Ilustración 55: Parada de seguridad

Por último, al igual que en los anteriores hilos, se ha utilizado la función *sync()* de forma que todos los hilos quedan sincronizados, y por lo tanto se ejecutan todos a la vez.

6. Validación del funcionamiento

Por último, se ha realizado la validación del funcionamiento de la aplicación, de forma que se han realizado todos los movimientos que se han programado, así como cada uno de los modos de los que se ha dotado al dispositivo.

Para una correcta validación del proyecto, se ha procedido al montaje final del guante. Primero se ha realizado una comprobación de todas las conexiones que posee el guante, con el fin de evitar cualquier posible cortocircuito que pueda suponer un peligro para el propio robot; y posteriormente se ha procedido al anclaje de cada una de las partes, de forma que todo queda correctamente anclado a la estructura para así poder evitar cualquier fallo en el funcionamiento.

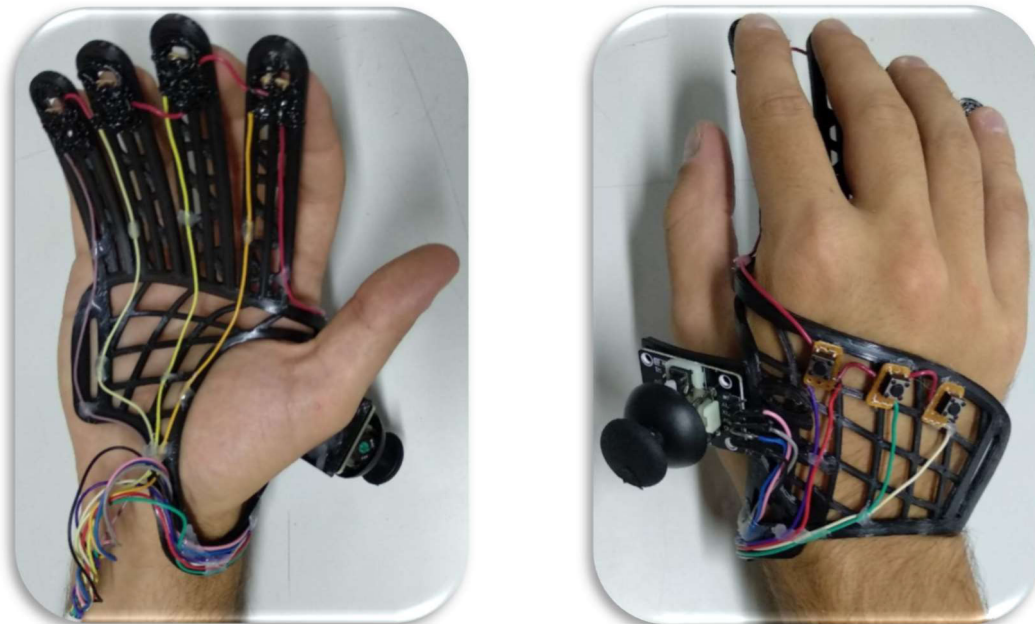


Ilustración 56: Dispositivo externo de guiado

La segunda parte de la validación es la correcta conexión a las entradas del robot, de forma que en el monitor del que dispone el robot se observa como la lectura de las entradas analógicas coinciden con los movimientos del joystick, así como la lectura de las entradas digitales, que coincide con la pulsación o no pulsación de cada uno de los botones del guante.

Por último, para validar el funcionamiento de la aplicación se ha puesto en marcha el programa que se ha desarrollado mediante el cual se pueden realizar todos los movimientos que se requerían en el inicio de este proyecto.

Después de numerosas pruebas se puede concluir con que el dispositivo cumple con los requerimientos funcionales que se plantearon al principio del proyecto, ya que se permite el movimiento del robot en todos los ejes tanto mediante movimientos lineales como movimientos angulares. Por otra parte, se permite la correcta grabación de trayectorias y su posterior reproducción; y por último también se puede validar la última función añadida, la cual realiza una parada de emergencia segura en el caso en el que el efector final del robot sufra una determinada fuerza.



Ilustración 57: Realización de movimientos

Por otra parte, también existen soluciones alternativas al presente proyecto y que pueden utilizarse para realizar las mismas funciones que este:

- Utilización de comunicación inalámbrica: En este caso la comunicación con el robot no se realizaría mediante la utilización de las entradas analógicas y digitales del robot, sino que mediante la comunicación TCP-IP se realiza la comunicación entre un ordenador y el robot. Para ello, tal y como se acaba de comentar, sería necesaria la utilización de un ordenador en el cual habría un programa ejecutándose en él, que sería el encargado de comunicarse con el dispositivo externo, y a la vez mandaría mediante un *socket* la información al robot. Este sistema tiene la principal desventaja de que si falla la comunicación TCP-IP o se detiene el programa del ordenador, la aplicación dejaría de funcionar hasta que se resolviera el problema.
- Por otra parte, el robot también permite comunicarse con dispositivos externos mediante comunicación MODBUS, por lo tanto, esta forma sería otra opción para la realización de este proyecto. Sin embargo tiene el mismo impedimento en cuanto a funcionamiento que el anterior caso.

7. Conclusiones

Tras la discusión, análisis y estudio de los resultados obtenidos en la realización de este trabajo de fin de máster, se concluye el presente proyecto de una forma satisfactoria y positiva.

En primer lugar, se ha conseguido diseñar y montar un dispositivo externo que permite realizar todos los movimientos que se plantearon como requisitos para el proyecto, así como las funciones de grabar trayectorias y repetirlas. Además, se ha añadido una función extra, de forma que, si se detecta que se aplica cierta fuerza en la herramienta del robot, este se para hasta que mediante un botón se desbloquee. De esta forma se ha implementado un sistema de seguridad para evitar posibles daños que se puedan producir en la herramienta.

Por otro lado se ha conseguido diseñar un dispositivo que permite realizar movimientos del robot sin la necesidad de utilizar la consola de programación que proporciona el fabricante, de forma que resulta más intuitivo para el operario a la hora de realizar los movimientos, y además dota de un grado de seguridad añadido ya que con el dispositivo diseñado no hace falta estar cerca del robot para manejarlo, debido a que el usuario puede disponer de una longitud de cable variable dependiendo de los requerimientos.

Por último, cabe destacar que el dispositivo diseñado es capaz de comunicarse con el robot por sí solo, por lo que no necesita de dispositivos adicionales para realizar la comunicación con el robot, como puedan ser ordenadores portátiles, microcontroladores o placas de desarrollo; ya que directamente se trabaja con las entradas digitales y analógicas que posee el propio robot.

En cuanto a las líneas futuras para este proyecto, se quedan abiertos muchos caminos en los que se puede investigar y obtener buenos resultados. Uno de ellos podría ser la ampliación de las funcionalidades que posee el dispositivo externo, ya sea realizando comunicación con un dispositivo colocado en el efector final, ya sea comercial o no comercial. También se podría realizar la sensorización de cada una de las articulaciones del robot de forma que se pueda detectar cualquier colisión con cualquier parte del robot, ya que de serie el robot únicamente posee un sensor de fuerza en la última articulación, es decir, en la herramienta.

Por otro lado, queda también abierta la investigación en otros dispositivos comerciales, como puede ser el *Myo Armband*, el cual permite conocer el

movimiento que se realiza con el brazo gracias a la lectura eléctrica de los músculos, así como del movimiento del brazo.

Por ultimo comentar que también se puede realizar una investigación para crear un dispositivo en forma de guante, tal y como se ha realizado en este proyecto, pero que sea ajustable a la mano del operario en cada momento de forma que lo puedan utilizar diferentes personas; ya que en este proyecto el guante se ha realizado a medida para la mano del autor.

8. Referencias

1. ABB. (s.f.). *IRB 14050 single-arm YuMi*. Obtenido de <https://new.abb.com/products/robotics/es/robots-industriales/irb-14050-single-arm-yumi>
2. AER Automation. (22 de Octubre de 2018). *Las ventas mundiales de robots industriales aumentan un 30% en 2017*. Obtenido de <https://www.aer-automation.com/las-ventas-mundiales-de-robots-industriales-aumentan-un-30-en-2017/>
3. Castets, B. (8 de Julio de 2019). *Control UR robot using a joystick connected on a PC*. Obtenido de <https://dof.robotiq.com/discussion/1681/control-ur-robot-using-a-joystick-connected-on-a-pc>
4. Colegio de Ingenieros Técnicos Industriales de Valencia. (10 de Junio de 2018). Obtenido de <https://www.copitival.es>
5. Energid Technologies. (15 de Octubre de 2015). *Actin Controlling a UR10 with OptiTrack Motion Tracking System*. Obtenido de <https://www.youtube.com/watch?v=NEy4PvG8Pe8>
6. FANUC. (s.f.). *CR-14iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%d1%80obots-colaborativos/collaborative-cr-14ial>
7. FANUC. (s.f.). *CR-15iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%d1%80obots-colaborativos/collaborative-cr15ia>
8. FANUC. (s.f.). *CR-35iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%d1%80obots-colaborativos/robot-colaborativo-cr35ia>
9. FANUC. (s.f.). *CR4-4iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%d1%80obots-colaborativos/collaborative-cr4ia>
10. FANUC. (s.f.). *CR-7iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%d1%80obots-colaborativos/collaborative-cr7ial>

11. González, V. R. (2003). *Definición de robot industrial*. Obtenido de http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/industrial.htm
12. IFR. (s.f.). *Robótica Colaborativa*. Obtenido de <https://www.aer-automation.com/mercados-emergentes/robotica-colaborativa/>
13. ISO/TC 299 Robotics. (Noviembre de 2000). *International Organization for Standardization*. Obtenido de ISO 14539:2000: <https://www.iso.org/standard/24062.html>
14. ISO/TC 299 Robotics. (Julio de 2011). *International Organization for Standardization*. Obtenido de ISO 10218-2:2011: <https://www.iso.org/standard/41571.html>
15. ISO/TC 299 Robotics. (Julio de 2011). *International Organization for Standardization*. Obtenido de ISO 10218-1:2011: <https://www.iso.org/standard/51330.html>
16. ISO/TC 299 Robotics. (Marzo de 2012). *International Organization for Standardization*. Obtenido de ISO 8373:2012: <https://www.iso.org/standard/55890.html>
17. ISO/TC 299 Robotics. (Febrero de 2016). *International Organization for Standardization*. Obtenido de ISO/TS 15066:2016: <https://www.iso.org/standard/62996.html>
18. Jiva, E. I. (Julio de 2019). Desarrollo de la teleoperación de robots industriales y colaborativos mediante técnicas avanzadas de visión artificial.
19. KUKA. (s.f.). *LRB iiwa*. Obtenido de <https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/lbr-iiwa>
20. Rethink Robotics. (s.f.). *Sawyer*. Obtenido de <https://www.rethinkrobotics.com/sawyer>
21. Universal Robots. (s.f.). *UR10e*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur10/>
22. Universal Robots. (s.f.). *UR3e*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur3/>
23. Universal Robots. (s.f.). *UR5e*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur5/>

24. Wikipedia. (26 de Junio de 2019). *Cobot*. Obtenido de <https://en.wikipedia.org/wiki/Cobot>

Trabajo Final de Master

ANEXOS

Iván Salvador Cambronero

1. PRESUPUESTO

En este apartado de los anexos de la memoria se va a explicar de forma desglosada el presupuesto necesario para la realización del presente proyecto, ya que es necesario conocer cuál sería el coste total que tendría el proyecto a la hora de ponerlo en marcha en la industria.

Lo primero que se debe conocer es el coste que tendrían los materiales necesarios para la construcción del guante de guiado necesario para la aplicación desarrollada.

Tabla 11: Presupuesto hardware

Capítulo 1: Hardware			
Circuito guante guiado			
Concepto	Unidades	Precio/Ud.	Subtotal
Módulo joystick para arduino	1 ud	1,78 €	1,78 €
Botón arduino	7 ud	0,10 €	0,70 €
Cable puente Macho-Macho	0,25 ud	1,77 €	0,44 €
Cable multihilo	3 m	55,04 €	1,65 €
PLA Impresión 3D	0,03 kg	18,00 €	0,54 €
Placa circuito impreso	0,1 ud	4,56 €	0,46 €
		Total	5,57 €

A continuación, se muestra el coste que ha tenido el robot adquirido para la realización de esta aplicación.

Tabla 12: Presupuesto maquinaria

Capítulo 2: Maquinaria			
Robot Colaborativo			
Concepto	Unidades	Precio/Ud.	Subtotal
UR5e	1 ud	27.500,00 €	27.500,00 €
		Total	27.500,00 €

Por último, se recoge un desglose de las horas de mano de obra dedicadas a cada una de las fases del proyecto. Cabe destacar que se ha tenido en cuenta el precio por hora de un ingeniero, el cual se ha obtenido del Colegio de Ingenieros Industriales de Valencia.

Tabla 13: Presupuesto mano de obra

Capítulo 3: Mano de obra			
Concepto	Unidades	Precio/Ud.	Subtotal
Búsqueda de soluciones	100 h	35,00 €	3.500,00 €
Diseño CAD	40 h	35,00 €	1.400,00 €
Programación Aplicación	130 h	35,00 €	4.550,00 €
Soldadura circuito electrónico	50 h	35,00 €	1.750,00 €
		Total	11.200,00 €

A continuación, se muestra un resumen de cada uno de los capítulos del presupuesto, donde el presupuesto total es la suma de los capítulos anteriores, a los cuales se le aplica un 7% de gastos generales y un 6% de beneficio industrial.

A esta primera suma, se le debe aplicar posteriormente los impuestos correspondientes al IVA, los cuales corresponden al 21%.

Tabla 14: Resumen del presupuesto

Resumen	
Capítulo	Importe
Capítulo 1	5,57 €
Capítulo 2	27.500,00 €
Capítulo 3	11.200,00 €
Presupuesto de ejecución	38.705,57 €
7% de gastos generales	2.709,39 €
6% de beneficio industrial	2.322,33 €
Suma	43.737,29 €
IVA 21%	9.184,83 €
Total	52.922,13 €

De esta forma el coste total de la realización del proyecto es de 52.922,13€.

2. PLANOS

3. CÓDIGO

```

def Progfinal():
    set_safety_mode_transition_hardness(1)
    set_gravity([0.0, 0.0, 9.82])
    set_tool_communication(False, 115200, 0, 1, 1.5, 3.5)
    set_tool_output_mode(0)
    set_tool_digital_output_mode(0, 1)
    set_tool_digital_output_mode(1, 1)
    set_tcp(p[0.0,0.0,0.0,0.0,0.0,0.0])
    set_payload(0.0)
    set_standard_analog_input_domain(0, 1)
    set_standard_analog_input_domain(1, 1)
    set_tool_analog_input_domain(0, 1)
    set_tool_analog_input_domain(1, 1)
    set_analog_outputdomain(0, 0)
    set_analog_outputdomain(1, 0)
    set_tool_voltage(0)
    set_input_actions_to_default()
    $ 1 "AntesDeIniciar"
    $ 2 "b1:= False "
    global b1= False
    $ 3 "b2:= False "
    global b2= False
    $ 4 "b3:= False "
    global b3= False
    $ 5 "b4:= False "
    global b4= False
    $ 6 "b5:= False "
    global b5= False
    $ 7 "q:= False "
    global q= False
    $ 8 "b6:= False "
    global b6= False
    $ 9 "b7:= False "
    global b7= False
    $ 10 "ejexaux:=0"
    global ejexaux=0
    $ 11 "ejex:=0"
    global ejex=0
    $ 12 "girejex:=0"
    global girejex=0
    $ 13 "ejey:=0"
    global ejey=0
    $ 14 "girejey:=0"
    global girejey=0
    $ 15 "ejeyaux:=0"
    global ejeyaux=0
    $ 16 "nposiciones:=0"
    global nposiciones=0
    $ 17 "grabarON:= False "
    global grabarON= False
    $ 18 "realiztrayec:= False "
    global realiztrayec= False
    $ 19 "fuerzaz:=0"
    global fuerzaz=0

```

```

$ 20 "fuerzaparar:= False "
global fuerzaparar= False
$ 21 "posrep:=p[0.089,-0.384,0.5,2.989,0.992,-0.027]"
global posrep=p[0.089,-0.384,0.5,2.989,0.992,-0.027]
$ 22
"trayec:=[posrep,posrep,posrep,posrep,posrep,posrep,posrep,posre
p,posrep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep]"
global
trayec=[posrep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,
posrep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep,posrep,posrep,posrep,po
srep,posrep,posrep,posrep,posrep]
$ 23 "pos:=[0,0,0,0,0,0]"
global pos=[0,0,0,0,0,0]
$ 24 "modofuerza:= False "
global modofuerza= False
$ 25 "i:=0"
global i=0
$ 26 "k:=0"
global k=0

$ 43 "Thread_1"
thread Thread_1():
  while (True):
    # 'Thread encargado de componer el vector de movimiento
del robot'
    global ejexaux=get_standard_analog_in(0)
    if (ejexaux>1.8 and ejexaux<2.1):
      global ejexaux=2
    end
    global ejex=(2*ejexaux-4.7)/47
    if (ejex<0.015 and ejex>-0.015):
      global ejex=0
    end
    if (ejex<0.12 and ejex>0.08):
      global ejex=0.1
    end
    if (ejex<-0.08 and ejex>-0.12):
      global ejex=-0.1
    end
    global girejex=ejex*3
    if (girejex<0.1 and girejex>-0.1):
      global girejex=0
    end
    if (girejex>0.28 and girejex<0.31):
      global girejex=0.3
    end
    if (girejex<-0.28 and girejex>-0.31):
      global girejex=-0.3

```



```

end
# 'Eje y'
global ejeyaux=get_standard_analog_in(1)
if (ejeyaux>1.8 and ejeyaux<2.1):
    global ejeyaux=2
end
global ejey=(-2*ejeyaux+4.7)/47
if (ejey<0.016 and ejey>-0.016):
    global ejey=0
end
if (ejey<0.12 and ejey>0.08):
    global ejey=0.1
end
if (ejey<-0.08 and ejey>-0.12):
    global ejey=-0.1
end
global girejey=ejey*3
if (girejey<0.1 and girejey>-0.1):
    global girejey=0
end
if (girejey<0.31 and girejey>0.28):
    global girejey=0.3
end
if (girejey>-0.31 and girejey<-0.28):
    global girejey=-0.3
end
global b1=get_standard_digital_in(0)
global b2=get_standard_digital_in(1)
global b3=get_standard_digital_in(2)
global b4=get_standard_digital_in(3)
if (b2 == True and b1 == False ):
    global pos=[-ejex,-ejey,0,0,0,0]
end
if (b3 == True and b1 == False ):
    global pos=[0,-ejey,ejex,0,0,0]
end
if (b4 == True and b1 == False ):
    global pos=[-ejex,0,-ejey,0,0,0]
end
if (b2 == True and b1 == True ):
    global pos=[0,0,0,girejey,-girejex,0]
end
if (b3 == True and b1 == True ):
    global pos=[0,0,0,0,-girejex,girejey]
end
if (b4 == True and b1 == True ):
    global pos=[0,0,0,girejey,0,girejex]
end
sync()
end
end
threadId_Thread_1 = run Thread_1()

```

```

$ 97 "Thread_2"
thread Thread_2():
    while (True):
        # 'Thread encargado de guardar posiciones'
        nposiciones=get_list_length(trayec)
        if (realiztrayec == False and i<nposiciones and
grabarON == True ):
            while (i<nposiciones and realiztrayec == False ):
                trayec[i]= get_actual_tcp_pose ()
                global i=i+1
                sleep(0.5)
            end
        end
        sync()
    end
end
threadId_Thread_2 = run Thread_2()

$ 106 "Thread_3"
thread Thread_3():
    while (True):
        # 'Thread que calcula constantemente la fuerza ejercida en
la herramienta'
        # '"modofuerza" es para liberar al robot del modo fuerza
cuando se queda parado'
        if (modofuerza == True ):
            global fuerzaparar= False
            global realiztrayec= False
        end
        global fuerzaz= force ()
        if (fuerzaz >= 30):
            global fuerzaparar= True
            global modofuerza= False
        end
        sync()
    end
end
threadId_Thread_3 = run Thread_3()

$ 117 "Thread_4"
thread Thread_4():
    while (True):
        # 'Thread encargado de leer el modo que se desea'
        global b5=get_standard_digital_in(4)
        global b6=get_standard_digital_in(5)
        global b7=get_standard_digital_in(6)
        if (b5 == True ):
            global grabarON= True
        end
        if (b6 == True ):
            global realiztrayec= True
        end
        if (b7 == True ):
            global modofuerza= True
        end
        sync()

```

```

end
end
threadId_Thread_4 = run Thread_4()

while (True):
    $ 27 "Programa de robot"
    $ 28 "If fuerzaparar≠ True "
    if (fuerzaparar != True ):
        $ 29 "speedl(pos,0.5)"
        speedl(pos,0.5)
    end
    $ 30 "If fuerzaparar≠ True "
    if (fuerzaparar == True ):
        $ 31 "stopl(20)"
        stopl(20)
    end
    $ 32 "If realiztrayec≠ True "
    if (realiztrayec == True ):
        $ 33 "Bucle k<i and fuerzaparar≠ False "
        while (k<i and fuerzaparar == False ):
            $ 34 "mover:=trayec[k]"
            global mover=trayec[k]
            $ 35 "MoverL"
            $ 36 "mover"
            movel(mover, a=0.5, v=0.1)
            $ 37 "k:=k+1"
            global k=k+1
        end
        $ 38 "i:=0"
        global i=0
        $ 39 "k:=0"
        global k=0
        $ 40 "realiztrayec:= False "
        global realiztrayec= False
        $ 41 "modofuerza:= False "
        global modofuerza= False
        $ 42 "grabaron:= False "
        global grabaron= False
    end
end
end
end

```