

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERÍA**

**DISEÑO Y ELABORACIÓN DE PRÁCTICAS DE LABORATORIO
PARA ROBÓTICA INDUSTRIAL UTILIZANDO MANIPULADORES
ROBÓTICOS KUKA**

TANIA VANESSA UTRERAS AGUILAR

SANGOLQUÍ – ECUADOR

2013

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Ing. Alexander Ibarra

Ing. Víctor Proaño

Certificamos que el proyecto titulado “DISEÑO Y ELABORACIÓN DE PRÁCTICAS DE LABORATORIO PARA ROBÓTICA INDUSTRIAL UTILIZANDO MANIPULADORES ROBÓTICOS KUKA”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Tania Vanessa Utreras Aguilar que lo entreguen al Ingeniero Luis Orozco, en su calidad de Director de la Carrera.

Sangolquí, 19 de agosto del 2013

Ing. Alexander Ibarra
DIRECTOR

Ing. Victor Proaño
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL****DECLARACIÓN DE RESPONSABILIDAD**

TANIA VANESSA UTRERAS AGUILAR

DECLARO QUE:

El proyecto de grado denominado “DISEÑO Y ELABORACIÓN DE PRÁCTICAS DE LABORATORIO PARA ROBÓTICA INDUSTRIAL UTILIZANDO MANIPULADORES ROBÓTICOS KUKA”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas y fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

Sangolquí, 19 de agosto del 2013

TANIA VANESSA UTRERAS AGUILAR

DEDICATORIA

Esta tesis fruto de mi esfuerzo la dedico a mis padres por toda la gratitud que les tengo. A mis hermanos por su apoyo incondicional y palabras de aliento durante mi formación profesional.

Vanessa

AGRADECIMIENTO

Primeramente quiero agradecer a Dios por darme sabiduría para culminar satisfactoriamente esta etapa de mi vida.

A mi padre por ser un ejemplo de responsabilidad y perseverancia.

A mi madre que ha sido mi compañera de largas jornadas y es mi fortaleza en momentos de decline.

A mi hermana Mary que es mi ejemplo de vida que día a día me brinda su cariño, tiempo, experiencias y apoyo incondicional.

A mi hermano Jeffo por compartir conmigo este logro y exigirme ser mejor.

A esta prestigiosa Universidad especialmente a mi director Ing. Alexander Ibarra quien ha sido más que un profesor una guía, apoyo y amigo e Ing. Victor Proaño por su orientación y colaboración en el desarrollo de este proyecto.

A mis compañeros de aula por brindarme su amistad, compartir vivencias y conocimientos.

Vanessa Utreras

PRÓLOGO

Los robots industriales están concentrados en operaciones de producción por ser adaptables al entorno, de fácil manejo, con gran capacidad para realizar tareas repetitivas y de precisión. Estas características han llevado a la Robótica a su continuo crecimiento. Este proyecto solventa problemas en la realización de prácticas de laboratorio ya que el Departamento de Eléctrica y Electrónica en el área de Automática y Robótica posee un déficit de manipuladores robóticos CRS A255 funcionales; por tal motivo la utilización de manipuladores robóticos KUKA existentes en los laboratorios del Departamento de Ciencias de la Energía y Mecánica ayudarán a complementar la falta de prácticas aplicativas existentes actualmente en la materia.

El desarrollo de este proyecto favorecerá en la formación sólida del estudiante con métodos interactivos asegurando un buen desempeño e inserción en un futuro ámbito laboral. Para lo cual se diseñó prácticas que contienen trabajo preparatorio, marco teórico, procedimiento sistemático y trabajo complementario que permiten la identificación, programación y funcionamiento íntegro del hardware y software del manipulador robótico KUKA KR 16 y KR 5ARC.

INDICE DE CONTENIDO

CAPÍTULO 1	1
DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1. ANTECEDENTES.....	1
1.2. JUSTIFICACIÓN E IMPORTANCIA	2
1.3. ALCANCE DEL PROYECTO.....	2
1.4. OBJETIVOS	3
1.4.1 General.....	3
1.4.2 Específicos	3
CAPÍTULO 2	4
MANIPULADORES ROBÓTICOS KUKA	4
2.1 INTRODUCCIÓN A LA ROBÓTICA	4
2.2 HISTORIA DE LA ROBÓTICA INDUSTRIAL	5
2.3 ROBOT INDUSTRIAL.....	11
2.4 MANIPULADORES ROBÓTICOS	11
2.5 ORGANIZACIÓN FUNCIONAL	14
2.6 LAZO DE CONTROL.....	15
2.7 PARTES DE UN MANIPULADOR ROBÓTICO	16
2.8 ELEMENTOS DE UN ROBOT	17
2.8.1 UNIDAD DE CONTROL.....	20
2.8.2 DISPOSITIVOS DE ENTRADA Y SALIDA	20
2.8.3 UNIDAD MANUAL DE PROGRAMACIÓN	21
2.9 SISTEMAS DE COORDENADAS.....	27
2.10 CARACTERÍSTICAS DE LOS MANIPULADORES INDUSTRIALES	28
2.11 MANIPULADORES ROBOTICOS KUKA	30
2.12 MANIPULADOR ROBOTICO INDUSTRIAL KR 5 ARC.....	31
2.13 SEGURIDAD DEL ROBOT.....	33
2.14 SOFTWARE	34
2.15 COMUNICACIÓN KUKA OFFICE LITE CON KUKA SIM PRO 2.2	37

CAPÍTULO 3	41
GUÍAS DE LABORATORIO	41
3.1 IDENTIFICACIÓN DEL HARDWARE DEL MANIPULADOR ROBÓTICO KUKA KR 16 Y KR 5ARC	41
3.2 FUNCIONES DE LAS TECLAS DEL KCP Y MOVIMIENTO MANUAL EN LOS SISTEMAS DE COORDENADAS.....	58
3.3 CONFIGURACIÓN Y MEDICIÓN DE LA BASE Y HERRAMIENTA DEL ROBOT	66
3.2 PROGRAMACIÓN DE TRAYECTORIAS DEL MANIPULADOR ROBÓTICO EN MODO USUARIO	76
3.5 SIMULACIÓN DE MOVIMIENTOS DEL ROBOT EN EL ENTORNO VIRTUAL KUKASIM.PRO.....	93
3.6 PROGRAMACIÓN EN MODO EXPERTO USANDO ESTRUCTURAS EN KRL.....	108
3.7 TRAYECTORIAS CON SENTENCIAS CONDICIONALES.....	115
3.8 SUBPROGRAMAS Y FUNCIONES.....	124
3.9 CONFIGURACIÓN DEL MÓDULO DE ENTRADAS Y SALIDAS DIGITALES.....	129
3.10 PROCESO DE PALETIZACIÓN	142
3.11 MISCELANEA DE PROCESOS CON SENTENCIAS CONDICIONALES ..	151
3.12 SIMULACIÓN DE UN PROCESO DE ENSAMBLAJE EN KUKA SIMPRO160	
3.13 GUÍA PARA EL RESPALDO DE LA INFORMACIÓN.....	165
 CAPÍTULO 4	 168
CONCLUSIONES	168
4.1 CONCLUSIONES:.....	168
4.2 RECOMENDACIONES:.....	171
REFERENCIAS BIBLIOGRÁFICAS	172
ANEXOS	¡Error! Marcador no definido.

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROYECTO

1.1. ANTECEDENTES

La automatización nos exige la interacción entre el ser humano y los sistemas tecnológicos para desarrollar procesos más confiables y eficientes por lo que las distintas aplicaciones ofrecidas por los manipuladores robóticos KUKA constituyen una alternativa para lograr la optimización de recursos en procesos como: soldadura, plegado, proveedores de la industria automovilística, metalúrgico, plástico, industria alimentaria, etc.

En los últimos años Ecuador ha ido incursionando en el campo de la Robótica industrial, siendo esta una herramienta para realizar productos de excelente calidad y ser más competitivos en el mercado con la participación de profesionales que permitan cumplir este objetivo.

Por ello el Departamento de Eléctrica y Electrónica para contribuir con la filosofía de Escuela Politécnica del Ejército “camino a la excelencia” adquirió manipuladores robóticos para la cátedra de Robótica Industrial con el fin de afianzar los conocimientos adquiridos.

1.2. JUSTIFICACIÓN E IMPORTANCIA

Actualmente la influencia tecnológica sobre los procesos exige que el estudiante desarrolle nuevas competencias para enfrentar exitosamente nuevos retos en la industria.

Con este proyecto se solventarán problemas en la realización de prácticas ya que el Departamento de Eléctrica y Electrónica en el área de Automática y Robótica posee un déficit de manipuladores robóticos CRS255 funcionales; por tal motivo la utilización de manipuladores robóticos KUKA existentes en los laboratorios del Departamento de Ciencias de la Energía y Mecánica ayudarán a complementar la falta de prácticas aplicativas existentes actualmente en la materia.

Por lo que es importante contar con guías de laboratorio que formen sólidamente al estudiante con prácticas, métodos interactivos y operando los equipos, obteniendo mejores resultados; asegurando un buen desempeño y mejor inserción en un futuro ámbito laboral.

1.3. ALCANCE DEL PROYECTO

Este proyecto tiene como finalidad diseñar y elaborar prácticas de laboratorio utilizando manipuladores robóticos KUKA para la cátedra de Robótica Industrial. Cuenta con teoría básica de los dispositivos, funcionamiento de cada componente del manipulador robótico y medidas generales de seguridad; además se indica los procedimientos a seguir.

Se planteó una serie de prácticas aplicativas las cuales fueron previamente probadas para garantizar su óptimo funcionamiento. Para el desarrollo de lo planteado se diseñó un documento que contiene la siguiente información, tema, resultado de aprendizaje, marco teórico, materiales, procedimiento, planteamiento

de tareas.

Adicionalmente se diseñó guías preparatorias que cuentan con preguntas referentes al tema de la práctica que permiten ejercitar e involucrar al estudiante en el desarrollo de la misma.

1.4. OBJETIVOS

1.4.1. General

Diseñar y elaborar prácticas de laboratorio para Robótica Industrial utilizando manipuladores robóticos KUKA.

1.4.2. Específicos

- Conocer y utilizar todos los comandos que ofrece el KCP (KUKA CONTROL PANEL) que permita el correcto funcionamiento del manipulador.
- Diseñar preparatorios de laboratorio para cada práctica para la cátedra de Robótica Industrial, para ejercitar al estudiante previo al desarrollo de la práctica.
- Realizar las simulaciones de las prácticas, aprovechando las bondades que nos proporciona el software KUKA SIM PRO.
- Diseñar prácticas de laboratorio para la cátedra de Robótica Industrial.

CAPÍTULO 2

MANIPULADORES ROBÓTICOS KUKA

2.1 INTRODUCCIÓN A LA ROBÓTICA

Robótica proviene del término Esloveno “Robota” definido como: trabajo forzado, introducido por el checo Karel Capek en el año 1921.

La Robótica empieza su desarrollo en los siglos XVII y XVIII, en este período se construyeron mecanismos con características humanas para reproducir las acciones de los seres vivos, esta época se caracteriza por las constantes innovaciones tecnológicas inicialmente se trataba de robots mecánicos diseñados para un propósito específico dirigido principalmente para uso lúdico a medida de su desarrollo, se crearon androides, robots móviles, industriales, encargados de cumplir actividades específicas.

Los robots industriales están concentrados en operaciones de producción por ser adaptables al entorno, de fácil manejo, con gran capacidad para realizar tareas repetitivas y de precisión, siendo una alternativa en la mejora de productos, optimizando recursos, tiempos, disminuyendo accidentes en operadores sirviendo como herramientas de ayuda en esfuerzos físicos, estas características han llevado a la Robótica a su continuo crecimiento.

2.2 HISTORIA DE LA ROBÓTICA INDUSTRIAL

El nacimiento de la robótica industrial se debe a las exigencias en la producción, siendo la base para una automatización flexible fusionando los conocimientos de mecánica y electrónica.

- En 1962: Aparece el primer brazo robótico industrial, Unimate fue construido por General Motors se instaló a manera de prueba en una planta para funciones de manipulación como: fundición a presión, soldadura y ensamblaje.

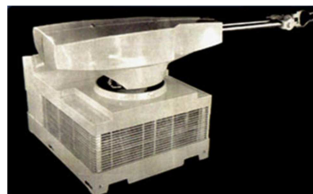


Figura 2.1: Brazo Industrial Robótico UNIMATE (GIZMODO)

- En 1966: Se crea ELIZA, el Primer Programa de Inteligencia Artificial.
- En 1968: Se desarrolló un robot móvil llamado 'Shakey' en Stanford Research Institute (SRI), dotado de sensores y una cámara de visión para desplazarse por el suelo. El control era mediante un computador del tamaño de una habitación.



Figura 2.2: Robot móvil SHARKEY (Artificial Intelligence Center)

- En 1973: SRI (Stanford Research Institute) desarrollo el primer lenguaje de programación de robots para la investigación denominado WAVE.
- En el mismo año KUKA construye el primer robot industrial denominado FAMULUS de seis ejes de accionamiento electromecánico su fin es la industria automotriz por su gran potencia y fiabilidad.

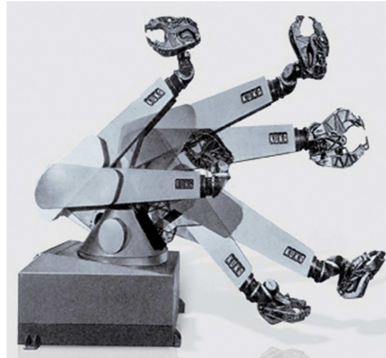


Figura 2.3: Robot Famulus (KUKA)

- En 1978: Se creó el robot PUMA (Máquina Programable Universal Para Ensamblar), dotado de seis articulaciones rotatorias que le proporcionan seis grados de libertad y permiten orientar su herramienta final, capaz de mover un objeto y colocarlo en cualquier ubicación. La programación del sistema del robot requería un lenguaje de alto nivel llamado VAL – II permite interactuar con señales externas.



Figura 2.4: Robot PUMA (INSTITUTO CULTURAL TEHUACAN)

- En 1979: Se desarrolló el SCARA (Selective Compliance Arm for Robotic Assembly) con el apoyo de dos universidades entre ellas Yamanashi Japón, para usar componentes de forma vertical, con un número reducido de grados de libertad 3 o 4 orientada al ensamblaje de piezas.
- En 1989: Genghis, El Primer Robot que puede caminar, es desarrollado en el Massachusetts Institute of Technology (EE.UU).

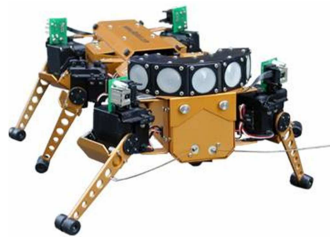


Figura 2.5: Robot móvil Genghis (TIME RIME)

- En 2000: Honda presenta a ASIMO el Robot Humanoide capaz de caminar y correr con gestos humanos cuyas características físicas son: 1,2 m de altura, 450 mm de ancho de hombros, 440 mm de profundo y 43 kg de peso, alcanza una velocidad de 3 km/h caminando, su alimenta con baterías de 38 voltios y 10 A, sus habilidades le permiten pasear por terreno llano, subir y bajar escaleras, abrir puertas, pulsar interruptores y empujar obstáculos.



Figura 2.6: Robot Humanoide ASIMO (HONDA)

- En 2005: KUKA desarrolló el Robot KUKA SAFE, diseñado para ofrecer seguridad operacional permitiéndole la interacción directa entre humanos y robots, de manera que el operario cuenta con seguridad cuando se encuentra en la zona de trabajo de un robot industrial porque el detiene su proceso si una persona o un obstáculo se cruza en su trayectoria de movimiento.



Figura 2.7: Robot KUKA SAFE (KUKA)

- En 2007: KUKA presenta el robot "TITAN " con carga útil de 1.000 kg de capacidad y un alcance de 3200 mm, con 6 ejes estas características le hacen el más grande y fuerte a nivel mundial lo que le hace parte del Libro de Récorde Guinness. Está dirigido a aplicaciones en vidrio, fundición e industria automotriz especialmente para trasladar carrocerías de automóviles.



Figura 2.8: Robot KUKA TITAN (KUKA)

- En este año KUKA presenta a Robot LWR de peso ligero construido de aluminio. Tiene una capacidad de carga de 7 kg, gracias a sus sensores integrados, es altamente sensible. Esto hace que sea ideal para tareas de manipulación y montaje. Debido a su reducido peso de sólo 16 kg, el robot es energéticamente eficiente y portátil.



Figura 2.9: Robot LWR (KUKA)

- En 2009: KUKA presenta la gama de robots Paletizadores KR 300 PA, KR 470 PA y KR 700 PA KUKA, rápidos y robustos cumplen funciones de: Paletizar, despaletizar, elevar, apilar, embalar, transportar, clasificar y etiquetar.

Poseen potentes motores que garantizan una producción rápida, con cargas de 300kg, 470kg y 700 kg y un trayecto de paletizado de 400/2000/400 mm respectivamente.



Figura 2.10: Robot KR 300 PA (KUKA)

- En 2010: KUKA presenta la nueva serie de robots QUANTEC, la primera y única gama de robots que cubre completamente el rango de capacidad de carga de 90 a 300 kg con trayectorias de hasta 3.10 m. Posee un nuevo dispositivo de manejo smartPAD en la plataforma WorkVisual facilitando el manejo y la programación. Su diseño permite minimizadas perturbadores y aumentar la accesibilidad en espacios reducidos.



Figura 2.11: Robot QUANTEC (KUKA)

- 2012: KUKA presenta la nueva gama de robots pequeños KR AGILUS. Posee 6 ejes, velocidades muy altas, tiempos de ciclos cortos, sistemas de potencia integrados, se instala en el piso, techo o pared. Con capacidad de carga de 6 kg a 10 kg y un alcance de aprox. 900 mm en su versión básica. De gran precisión y repetitividad, permiten ofrecer una calidad de producción.



Figura 2.12: Robot KR AGILUS (KUKA)

2.3 ROBOT INDUSTRIAL

Son máquinas constituidas por una estructura mecánica, dotadas de un sistema electrónico reprogramables con el fin de controlar movimientos e involucrarlos en procesos de producción.

Son sistemas robustos diseñados para mover cargas, materiales, piezas y herramientas con trayectorias previamente programadas, multifuncionales, tienen la capacidad de realizar actividades repetitivas, con un sistema flexible cuya principal característica es adaptar las líneas o células de fabricación a las diferentes tareas de producción, mejorando la velocidad y calidad de todos los procesos industriales y productivos. Ideales para trabajar en áreas de peligro o en ambientes nocivos.

2.4 MANIPULADORES ROBÓTICOS

"Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover material, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas" **(RIA, 2010)**

Este mecanismo permite reproducir capacidades de organismos vivos, denominado también brazo robótico por la analogía que presenta con las extremidades superiores del cuerpo humano, formado por elementos en serie y articulados entre sí.

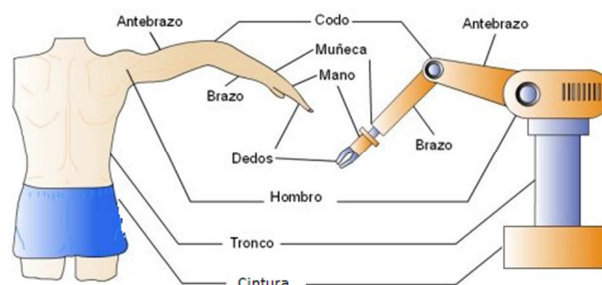


Figura 2.13: Analogía brazo humano vs brazo robótico **(Bueno)**

Entre las principales ventajas de los manipuladores robóticos se puede mencionar las siguientes:

- Mejorar la calidad de vida del operario.
- Reducción de errores humanos.
- Garantiza la seguridad, al realizar trabajos en zonas de alto riesgo para el ser humano.
- Incrementar la capacidad productiva con procesos continuos.
- Garantizar procesos más eficientes, minimizando desperdicios.
- Son reprogramables permitiendo flexibilidad para aceptar nuevos requerimientos.

Las desventajas de los manipuladores robóticos que se puede mencionar son las siguientes:

- No son aptos para la creatividad o la innovación.
- No se adaptan a los cambios en el entorno.
- Reduce el número de operarios en un proceso productivo.

ROBOT CARTESIANO: El robot de configuración cartesiano posee tres movimientos lineales X, Y y Z, sus articulaciones son de tipo prismático.

Los movimientos que cumple este robot están dados por interpolaciones lineales. Es decir la trayectoria se realiza en línea recta.

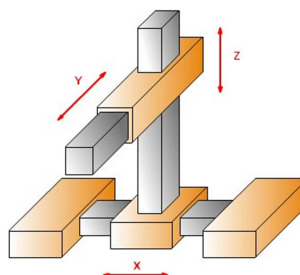


Figura 2.14: Robot Cartesiano (**Bueno**)

ROBOT POLAR: El robot de configuración polar realiza movimientos de tipo rotacional y prismático donde la primera y segunda articulación son de tipo

rotacional y la tercera es prismática. Se torna difícil controlar sus movimientos de traslación cuando trabaja con cargas de cierta magnitud, aun así son apropiados para mover cargas elevadas que no precisen mucha exactitud. Este robot utiliza la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión.

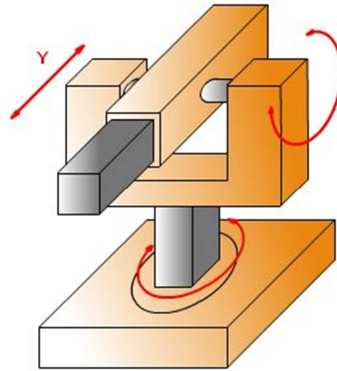


Figura 2.15: Robot Polar (Bueno)

ROBOT CILÍNDRICO: El robot de configuración cilíndrica, realiza dos movimientos: un prismático y uno rotacional. Este robot está diseñado para ejecutar los movimientos conocidos como interpolación lineal e interpolación por articulación. La interpolación por articulación se lleva a cabo en la primera articulación.

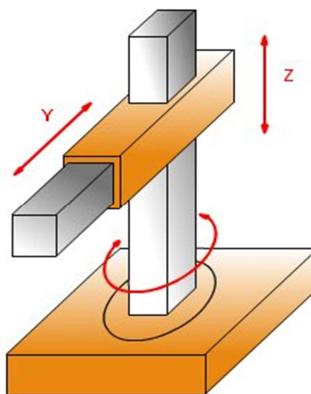


Figura 2.16: Robot Cilíndrico (Bueno)

ROBOT REVOLUTO O ANTROPOMÓRFICO: El robot de configuración antropomórfico simula los movimientos de un brazo humano, tiene tres articulaciones de tipo rotacional, sus articulaciones rotativas de hombro y de codo giran alrededor de ejes verticales, permitiendo que se efectúen giros en un plano horizontal.

Son caracterizados principalmente por la facilidad para acceder a zonas con obstáculos y accesibilidad limitada.

Presenta inconvenientes de precisión cuando trabaja con carga y a velocidades muy altas. Su precisión es inherentemente mayor en el entorno de la base. Sus grados de libertad permiten realizar aplicaciones industriales, desde la soldadura por puntos, hasta aplicaciones de pintado y sellado.

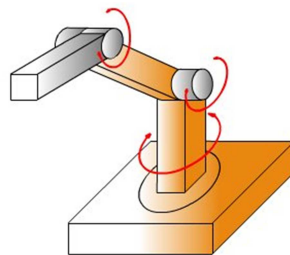


Figura 2.17: Robot Antropomórfico (Bueno)

2.5 ORGANIZACIÓN FUNCIONAL

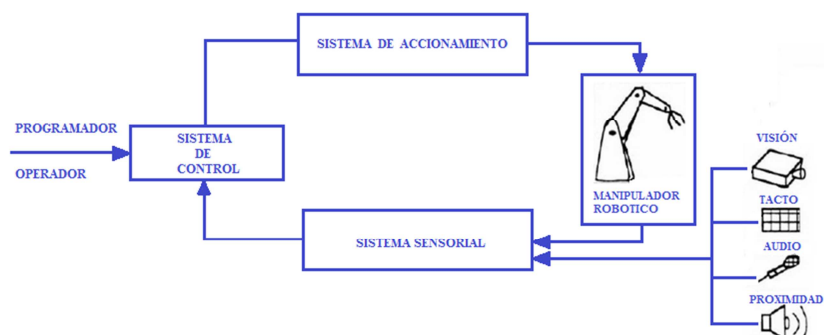


Figura 2.18: Organización funcional de un sistema robótico

La organización funcional está constituida sistemas que cumplen funciones específicas.

- **Sistema de control:** Es el centro de control de los movimientos que realizará el manipulador robótico con la información emitida por el programador u operador, a su vez procesa la información que proporcionan los sensores.
- **Sistema Sensorial:** Está compuesto por las señales o estímulos de tipo, auditivo, táctil, visual y proximidad que le llega del exterior como impulso eléctrico.
- **Sistema de Accionamiento:** Corresponde a los dispositivos de conversión de energía, que transforman la potencia eléctrica, hidráulica o neumática en una potencia mecánica, produciendo los movimientos del manipulador.
- **Manipulador Robótico:** Es la estructura mecánica o robot.

2.6 LAZO DE CONTROL

En un sistema de control en lazo cerrado, las señales de entrada y salida del sistema están dadas por coordenadas, el bloque de control está constituido por el software instalado en la PC, la planta es el manipulador robótico formado por motores, engranes y roscas. El sistema es realimentado por las señales que reciben los sensores de forma externa o interna.

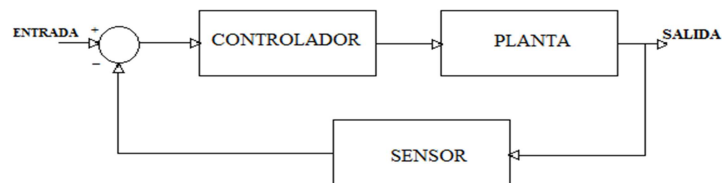


Figura 2.19: Lazo de control del manipulador robótico

2.7 PARTES DE UN MANIPULADOR ROBÓTICO

ESLABON: Son cuerpos estructurales rígidos del robot que proporcionan movimientos basados en un sistema de coordenadas a cada uno de ellos se les asocia un sistema de referencia para definir la posición y orientación.



Figura 2.20: Eslabón del Robot (Gonzalez)

ARTICULACIÓN: Su función es conectar los eslabones para permitir el movimiento, cada uno es controlado de manera independiente, al incorporar varias articulaciones con elementos rígidos llamados uniones permiten que sus movimientos sean combinaciones complejas de traslaciones y giros.



Figura 2.21: Articulaciones del Robot (Gonzales)

El conjunto de eslabones y articulaciones se denomina cadena cinemática. Una cadena cinemática es abierta cuando el eslabón se conecta mediante articulaciones directamente al extremo anterior y al siguiente, exceptuando el primero, que es fijado al soporte, y el último cuyo extremo final se conecta a un elemento terminal o actuador final. A continuación en la Tabla 2.1 se detallan los tipos de articulaciones.

Tabla 2.1: Tipos de articulaciones

ARTICULACIÓN	ESQUEMA	GRADOS DE LIBERTAD
ROTACIONAL		1
PRISMÁTICA		1
CILÍNDRICA		2
PLANAR		2
ESFÉRICA		3

2.8 ELEMENTOS DE UN ROBOT

ARMAZON: Es la estructura externa o cuerpo, define la forma física del robot o elementos de soporte.

SENSORES: Son dispositivos que miden en cantidades físicas. Su función es similar a la de nuestros sentidos, como herramienta de percepción permiten

apreciar lo que sucede en el medio ambiente, convierten un estímulo en una forma de energía.

Los sensores para robótica se clasifican en sensores externos enfocados en guiar al robot por el volumen de revolución y sensores internos para controlar y conocer el estado del robot. Entre ellos se pueden encontrar.

- **Sensores Táctiles:** Dispositivos que responden al contacto con objetos.
- **Sensores de Proximidad:** Dispositivo que usa un campo de detección electrónica para detectar la presencia de un objeto.
- **Sensor Electromecánico:** Dispositivos de seguridad que detienen al robot cuando entra en contacto con una sonda o barra de contacto.
- **Sensores Fotoeléctricos:** Dispositivo que detecta la presencia de un objeto por medio del uso de un haz de luz. Los sensores fotoeléctricos tienen un rango de detección extremadamente amplio.
- **Fin de Carrera:** Sensor mecánico que requiere contacto físico para detectar la presencia o ausencia de un objeto.

TRANSMISORES: Los transmisores pueden modificar o conservar la energía son los encargados de transmitir el movimiento desde los actuadores hasta las articulaciones, utilizados para convertir movimientos circulares en lineales o viceversa.

Se debe considerar que el sistema de transmisores no afecte al movimiento para ello no debe existir rozamiento, ni ser muy holgado y ser capaz de soportar un funcionamiento continuo.

REDUCTORES: Son los encargados de adaptar el par y la velocidad de la salida del actuador especialmente cuando el robot mueve su extremo con aceleraciones elevadas permitiendo minimizar el momento de inercia, ya que el trabajo de los robots implica continuos arranques y paradas.

ACTUADORES: Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en energía mecánica.

Los elementos que conforman un actuador son los siguientes:

- Sistema de accionamiento: es el encargado de producir el movimiento
- Sistema de transmisión: es el encargado de transmitir el movimiento del actuador a otros elementos.
- Sistema reductor: encargado de adecuar el torque y la velocidad del actuador a los valores requeridos.

Tabla 2.2: Cuadro comparativo entre los tipos de actuadores

TIPO DE ACTUADOR	CARACTERISTICA	EJEMPLO	VENTAJAS	DESVENTAJA
NEUMÁTICOS	La energía del aire comprimido se transforma por medio de cilindros en movimiento circular o movimiento rectilíneo.	Cilindros y motores neumáticos.	Bajo costo Rapidez Sencillo Robustos	Ruidosos Requiere de instalación especial.
HIDRÁULICOS	Obtienen su energía de un fluido a presión, generalmente algún tipo de aceite.	cilindros hidráulicos, motores hidráulicos y válvulas hidráulicas	Rápidos Alta capacidad de carga Estables	Requieren de instalación especial. Difícil mantenimiento.

ELÉCTRICOS	Los actuadores eléctricos transforman la energía eléctrica en energía mecánica rotacional.	Motores de corriente continua, motores de corriente alterna y de paso a paso.	Precisos Fiabiles Silenciosos Fácil instalación	Potencia limitada
-------------------	--	---	--	-------------------

2.8.1 UNIDAD DE CONTROL

Es el centro de control y modulación del movimiento del robot, generalmente son usados sistemas micro controlados; y para actuadores finales dependiendo de la aplicación electro neumática o electro hidráulica; la función del controlador es gobernar el trabajo de los actuadores o dispositivos que originan el movimiento y las transmisiones las cuales modifican el movimiento; la acción de control está determinada por medio de la comparación entre la referencia (posición requerida) y la señal de salida (posición actual) medida a través de encoders.

2.8.2 DISPOSITIVOS DE ENTRADA Y SALIDA

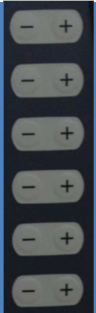

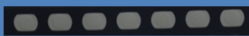
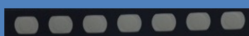
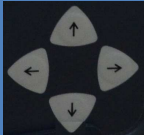



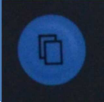
Principalmente la caja de comandos o teach pendant es un dispositivo que permite realizar el control del robot de forma remota, su función es dar las posiciones al manipulador del robot a través de la pantalla que permite editar comandos. Cuenta con un sistema de seguridad que permite el cierre de la operación inmediatamente en el caso de requerirlo.


2.8.3 UNIDAD MANUAL DE PROGRAMACIÓN

El KCP (KUKA Control Panel) es la unidad manual de programación del sistema KUKA, cuenta con los elementos de la parte frontal descritos en la Tabla 2.3

Tabla 2.3: Elementos de la parte de delante del KCP

ÍCONO	NOMBRE	DESCRIPCIÓN
	Selector de modos de servicio	Permite usar cuatro modos de trabajo, mediante la llave: T2 (Manual Velocidad alta) AUT (Automático) AUT EXT (Automático Externo) T1 (Manual Velocidad reducida)
	Accionamientos CONEXIÓN	Conecta los accionamientos del robot en modo automático.
	Accionamientos DESCONEXIÓN	Desconectar los accionamientos del robot en modo automático.
	Pulsador de PARADA DE EMERGENCIA	Es un componente de seguridad que permite detener y bloquear los circuitos de control del robot.
	Space Mouse	Permite mover los ejes del robot en todos los sistemas de coordenadas.

	Teclas de estado derecha	Son teclas multifunción programables. Activan y desactivan lo que indique el ícono.
	Tecla de entrada	Permite seleccionar un programa para su ejecución.
	Tecla de menú	Permiten abrir el menú correspondiente al ícono que se encuentre a lado de la tecla.
	Softkeys	Permite activar la función que indica el ícono, estos se modifican según la ventana y función.
	Tecla del cursor	Permite desplazar entre las líneas de un programa.
	Tecla de ejecución hacia adelante	Permite ejecutar las líneas de un programa.
	Tecla de ejecución hacia atrás	Permite retroceder las líneas de un programa.
	Tecla de STOP	Detiene el programa en ejecución. Produce una parada sobre la trayectoria en el modo de servicio automático.
	Tecla de selección de	Permite cambiar de una ventana a otra.

	ventana	
	Tecla ESC	Interrumpe una acción sobre la superficie de operación. También permite cerrar, paso a paso, los menús abiertos por equivocación

Teclado Alfabético: El teclado está dispuesto de diferentes partes:

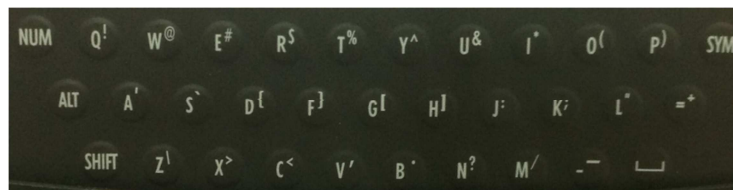


Figura 2.22: Teclado alfabético

- **NUM:** Activa el mando del bloque numérico. La barra de estados muestra la función que se encuentra activa.
- **ALT:** Es una tecla modificadora que se usa para acceder a menús tiene función de auto retención pulsándola una vez.
- **SHIFT:** Permite el cambio entre la escritura mayúscula y minúscula, SIM+SHIFT permite realizar escritura continua en mayúsculas. Tiene función de auto retención pulsándola una vez
- **SYM:** Activa los caracteres de la segunda asignación a las teclas de letras. Tiene función de auto retención pulsándola una vez.

Bloque numérico: Permite conmutar entre la función numérica y las funciones de mando.



Figura 2.23: Bloque numérico

- **INS (0)** Conmuta entre el modo de inserción y el de sobre escritura.
- **DEL (.)** Permite borrar los caracteres.
- **END (1)** Coloca el cursor al final de la línea en la cual se encuentra.
- **CTRL (2)** Se utiliza en combinaciones con otras teclas.
- **PG DN (3)** Pasar de una hoja a otra.
- **(4)** No tiene asignada ninguna función.
- **UNDO (5)** Anula la última entrada realizada.
- **TAB (6)** Coloca el curso sobre el próximo elemento de la superficie.
- **HOME (7)** Coloca el cursor al comienzo de la línea en la cual se encuentra.
- **LDEL (8)** Borra la línea en la que se encuentra el cursor.
- **PG DN (9)** Pasa hojas de la pantalla en dirección del comienzo del Archivo.

Tabla 2.4: Elementos de la parte posterior del KCP

ÍCONO	NOMBRE	DESCRIPCIÓN
	<p>Placa característica del KCP.</p>	<p>Contiene características de KCP y código de barras del producto.</p>
	<p>Tecla de arranque para programa.</p>	<p>Permite correr las líneas de programación.</p>
	<p>Pulsador de hombre muerto.</p>	<p>Posee tres pulsadores de hombre muertos. Permite accionar los ejes del robot junto a otros comandos.</p>

Barra de estado

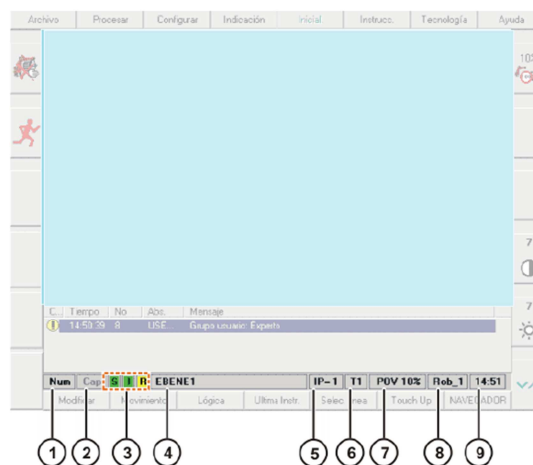


Figura 2.24: Operación de la barra de estado

- ① Estado del bloque numérico
- ② Estado de escritura minúsculas/mayúsculas
- ③ **S**: Estado del interpretador Submit
 - I: Estado de los accionamientos
 - R: Estado del programa
- ④ Nombre del programa seleccionado
- ⑤ Número del paso actual en el programa
- ⑥ Modo de servicio actual T1 / T2 /AUT /EXT
- ⑦ Override actual (POV override de programa / HOV override manual)
- ⑧ Nombre del robot
- ⑨ Horario del sistema

Tabla 2.5: Simbología del bloque numérico

SÍMBOLO	DESCRIPCIÓN
	Función de mando del bloque numérico activada.
	Función de mando del bloque numérico desactivada.
	Mayúsculas activadas.
	Minúsculas desactivadas.

Submit ejecuta pequeñas tareas de control de entradas y salidas, como circuitos de refrigeración y de seguridad. El estado del Submit indica la Tabla 2.6

Tabla 2.6: Simbología del Submit

SÍMBOLO	DESCRIPCIÓN
	El interpretador Submit desactivado.
	El interpretador Submit detenido.
	Interpretador Submit en ejecución.

Tabla 2.7: Simbología del Accionamiento








SÍMBOLO	DESCRIPCIÓN
	Accionamientos preparados
	Accionamientos no preparados

Tabla 2.8: Simbología del estado del programa

SÍMBOLO	DESCRIPCIÓN
	No se encuentra seleccionado ningún programa.
	El puntero está sobre la primera línea del programa seleccionado.
	El programa ha sido seleccionado y se encuentra en ejecución.
	El programa seleccionado y arrancado ha sido detenido.
	El puntero está sobre la última línea del programa seleccionado.

2.9 SISTEMAS DE COORDENADAS

El desplazamiento manual del robot depende del sistema de coordenadas respecto al cual se puedan hacer referencia los movimientos. En la unidad de control del robot se encuentran definidos los siguientes sistemas de coordenadas:

- **WORLD** : Sistema de coordenadas universales
- **ROBROOT** : Sistema de coordenadas en el pie del robot
- **BASE** : Sistema de coordenadas de base
- **FLANGE** : Sistema de coordenadas de brida
- **TOOL** : Sistema de coordenadas de herramienta

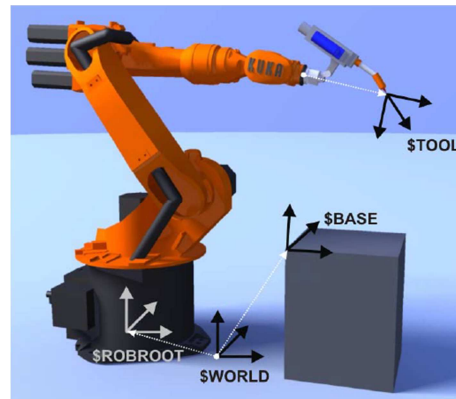


Figura 2.25: Sistema de coordenadas de robot (KUKA)

2.10 CARACTERÍSTICAS DE LOS MANIPULADORES INDUSTRIALES

Grados de libertad: Denominados DOF (degree of freedom) identifican el número y tipo de movimientos independientes con respecto a la articulación anterior, que puede realizar el manipulador en los diferentes ejes.

A más grados de libertad existe mayor flexibilidad en el posicionamiento y orientación del elemento terminal.

En los robots industriales se consideran seis grados de libertad: tres de ellos para definir la posición en el espacio y los otros tres para orientar la herramienta

Zona de trabajo: El robot cuenta con una geometría fija y limitada para determinar la zona de trabajo para lo cual se toma en cuenta únicamente el actuador, esta zona será el espacio dentro del cual puede desplazarse el extremo de su muñeca.

La zona de trabajo se ve restringida por los límites de giro y desplazamiento que existen en las articulaciones.

Capacidad de carga: Es la carga máxima en kg que el robot está en la capacidad de transportar a su velocidad nominal considerando su configuración y garantizando el posicionado. Es posible exceder la carga útil máxima sin embargo esto no se aconseja cuando el robot está acelerando. La carga útil debe ser menor que el valor de carga recomendada.

Resolución Espacial: Es el mínimo incremento en [mm] o variación de desplazamiento en el que el robot puede dividir su volumen de trabajo. La resolución espacial depende de las inexactitudes mecánicas que se encuentran estrechamente relacionadas con la calidad de los componentes que conforman las uniones y las articulaciones.

Exactitud: Es la capacidad que un robot presenta cuando su brazo opera cerca de la base, a medida que el brazo se aleja, la exactitud es menor. Un factor que afecta a la exactitud es el peso de la carga, mientras mayor sea la carga la exactitud se reduce.

Repetitividad: Esta característica hace referencia a la capacidad del robot de regresar al punto que se le programó las veces que sean necesarias. Indica la aproximación con la que el robot puede repetir una operación de ir a un punto.

Posición Home: Es un punto de referencia inicial para el robot que permite una repetición segura en sus movimientos.

El robot busca su posición home mediante los micros interruptores montados sobre cada una de las articulaciones del robot y monitorizados por el controlador.

La Position Home debe cumplir las siguientes condiciones:

- Posición de salida favorable para la ejecución del programa.
- Posición de parada favorable, esta no debe ser un obstáculo.

2.11 MANIPULADORES ROBOTICOS KUKA

Historia de KUKA

KUKA es una empresa Alemana fundada por Johann Josef Keller y Jakob Knappich, una de las más importantes en fabricación de robots industriales en el mundo. Sus principales competencias son el desarrollo, producción y venta de robots industriales, unidades de control, software y unidades lineales.

Caracterizada por ser líder innovador y tecnológico debido a que ofrece la mayor gama de aplicaciones del sector de la automatización.

KUKA ofrece mayor flexibilidad y un alto grado de seguridad debido al empleo de productos modulares. De este modo, es posible reequipar una instalación fácilmente para desempeñar nuevas tareas (sistemas de manufactura flexible).

KUKA provee aplicaciones completas dotando de células de trabajo que permiten aprovechar de forma eficaz las características de los manipuladores que ofertan para obtener un incremento más dinámico de la producción, con la consecuente reducción de los costos de fabricación y del tiempo invertido en diseño e ingeniería.

MANIPULADOR ROBOTICO INDUSTRIAL KUKA KR 16

El modelo KR 16 es un robot de carga ligera de gran versatilidad y flexibilidad, idóneo para instalaciones en las que se desea ahorrar espacio y costos. Por ello, es utilizado en casi todos los sectores, tanto en la industria automotriz como en áreas con un alto grado de suciedad y elevadas temperaturas.

Tabla 2.9: Nomenclatura del Robot

KR	ROBOT KUKA
----	------------

16	Identificación de carga en kg
-----------	-------------------------------



Figura 2.26: Robot KUKA KR 16 (KUKA, 2010)

MEDIDAS DEL ROBOT KR 16

La construcción de los manipuladores los convierte en aptos para aquellas instalaciones en las que se desea ahorrar espacio y costes, permitiéndoles poseer aplicaciones que se realizan en varios sectores.

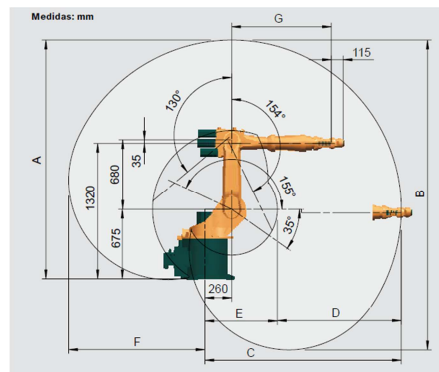


Figura 2.27: Dimensiones del Robot (Kuka Roboter GmbH, 2009)

2.12 MANIPULADOR ROBOTICO INDUSTRIAL KR 5 ARC

El manipulador KR 5 ARC cuya carga es de 5 kg, es ideal para las tareas de soldadura puede estar montado sobre el suelo o en el techo. El acople de la brida permite colocar herramientas para soldadura de arco.

KR 5 ARC cuenta con la mayor vida útil en su categoría con 40.000 horas de producción.

Ventajas:

El lubricante es de baja viscosidad hace que su mantenimiento sea más fácil todavía.

- Ahorro de espacio y mejor accesibilidad
- Posibilidades ilimitadas y máxima seguridad.
- Mayor productividad y mayor disponibilidad.

Tabla 2.10: Nomenclatura del Robot

KR	ROBOT KUKA
5	Identificación de carga en kg
ARC	Robot de soldadura de trayectoria con base plana y nuevo cableado de motor en eje 1



Figura 2.28: Robot KUKA KR 5 ARC (KUKA)

MEDIDAS DEL ROBOR KR5 ARC

Este modelo de manipulador es idóneo para instalaciones en las que se desea ahorrar espacio, debido a sus dimensiones compactas, el KR 5 ARC puede operar en espacios muy reducidos. La flexibilidad de este modelo de robot queda reflejada en las diversas opciones de montaje ya sea sobre el piso o techo.

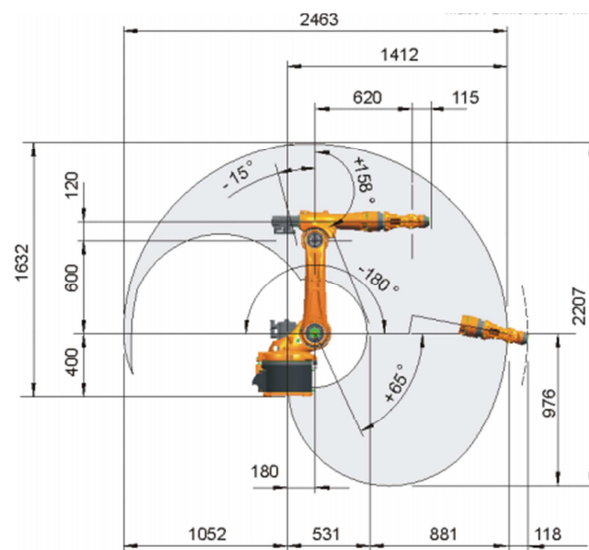


Figura 2.29: Dimensiones del Robot KR 5ARC (KUKA Roboter GmbH, 2009)

2.13 SEGURIDAD DEL ROBOT

Un sistema robótico debe contar con las características de seguridad pertinentes. Las celdas de los sistemas de seguridad están constituidas por:

- 1 Valla de protección
- 2 Topes finales mecánicos o limitaciones de los campos de los ejes 1, 2 y 3.
- 3 Puerta de protección con contacto para el control de la función del cierre.
- 4 Pulsador de parada de emergencia en un panel externo.
- 5 Pulsadores de parada de emergencia, interruptores de confirmación, interruptores con llave para abrir el gestor de conexiones.
- 6 Control de seguridad KR C2 integrado

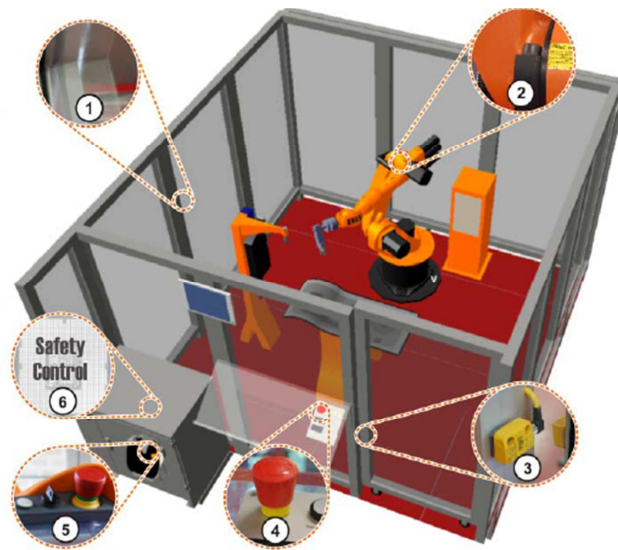


Figura 2.30: Célula de capacitación (KUKA Roboter GmbH, 2009)

2.14 SOFTWARE

KUKA posee una serie de módulos de programación con los que se podrá planificar y calcular previamente con total fiabilidad los procesos de producción, independientemente de que se planificar nuevas instalaciones u optimizar los sistemas existentes, con variedad de potentes herramientas de trabajo como visualización y simulación en 3D y la configuración de las cargas hasta la programación offline de sus robots e instalaciones. De esta forma se puede adaptar distintas tareas y ajustar su instalación sin necesidad de interrumpir la producción.

Windows XP embebido para KUKA

Windows XP Embebido elimina los inconvenientes y aprovecha las ventajas para el uso de nuevos programas. Al mismo tiempo, el usuario se beneficia de la facilidad de uso y las capacidades de visualización como la disponibilidad de numerosas aplicaciones. Este sistema es compatible con el controlador del robot KUKA y el KCP para proporcionar soporte en tiempo real mejorando la comunicación y capacidades gráficas.

El sistema presenta mayor flexibilidad, posee gran capacidad para adaptarse rápidamente a los cambios diarios en las necesidades de producción.

KUKA ROBOT LENGUAJE

Se ha desarrollado un programa en el lenguaje KRL (KUKA Robot Language), que es el que contiene las órdenes de movimiento del robot, para ello se ha utilizado el KRC Editor, programa facilitado por KUKA.

Es importante destacar que el lenguaje KRL trabaja directamente con el hardware del robot, facilitando al programador el control de los movimientos, ya que solo se especifica el punto destino al cual queremos que se desplace y el tipo de trayectoria a seguir para alcanzarlo.

Las características de velocidad, fiabilidad y robustez de los programas en KRL hacen aprovechar al máximo las capacidades del robot.

KUKA Sim Pro

Kuka Sim Pro ha sido desarrollado para la programación y simulación offline de robots KUKA. Este software no se limita a crear y presentar las simulaciones, sino que también es capaz de proyectar los movimientos programados en tiempo actual y evaluarlos con respecto a sus tiempos de ciclo. La unidad de control virtual del robot (VRC) está conectada al programa de simulación KUKA Sim Pro, con lo que el robot KUKA puede programarse directamente en KRL (KUKA Robot Language) sin tener que intervenir en la ejecución de los procesos. La biblioteca incluye una amplia gama de componentes eficientes. Al ser modular evita que los componentes tengan que ser dibujados de nuevo y supone por lo tanto un ahorro considerable de tiempo. Dispone además, un lenguaje abierto para crear archivos de órdenes, con el que puede ampliarse el comportamiento de los componentes de forma flexible. En KUKA.Sim Pro es posible crear y guardar bibliotecas de componentes propias para sus productos; de este modo podrá integrar dichos componentes en otros conceptos de instalación en cualquier momento.

KUKA. OfficeLite

KUKA.OfficeLite es el controlador virtual de robots KUKA y ha sido concebido principalmente para la programación y el desarrollo de aplicaciones. Permite crear y optimizar programas para robots de KUKA en cualquier PC. KUKA.OfficeLite es idéntico al software de sistema KUKA KR C2.

CARACTERÍSTICAS

- Disponibilidad de todo el repertorio de funciones de las respectivas ediciones del software de sistema, pero sin posibilidad de conectar hardware periférico, basado en Ethernet.
- Control completo de la ejecución de un programa de aplicación de robot en tiempo real. Ello permite optimizar la duración de los ciclos.
- Es posible simular señales de salida digitales para comprobar en el programa KRL la reacción a dichas señales.
- No es posible emplear KUKA.OfficeLite para controlar un robot en tiempo real.
- El programa simulado en KUKA.OfficeLite corre más lento que en el controlador real.



Figura 2.31: KUKA.OfficeLite V5.6 OL

2.15 COMUNICACIÓN KUKA OFFICE LITE CON KUKA SIM PRO 2.2

El software Kuka Office Lite permite comunicar el simulador de KUKA con el robot real. El objetivo es reproducir los movimientos ejecutados en KUKA Sim Pro en el robot real. Para lo cual se debe seguir los siguientes pasos:

Ejecutar los dos programas.

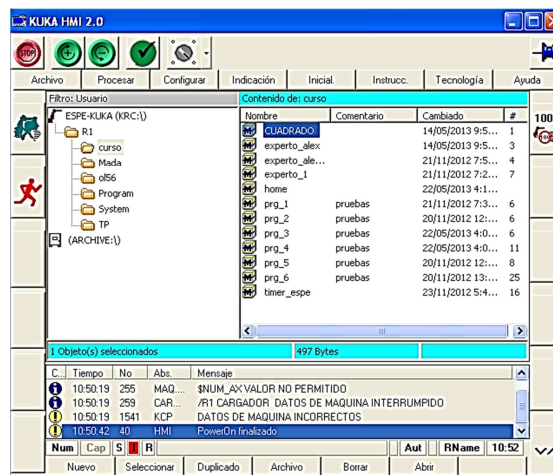


Figura 2.32: Interfaz de KUKA.OfficeLite V5.6 OL

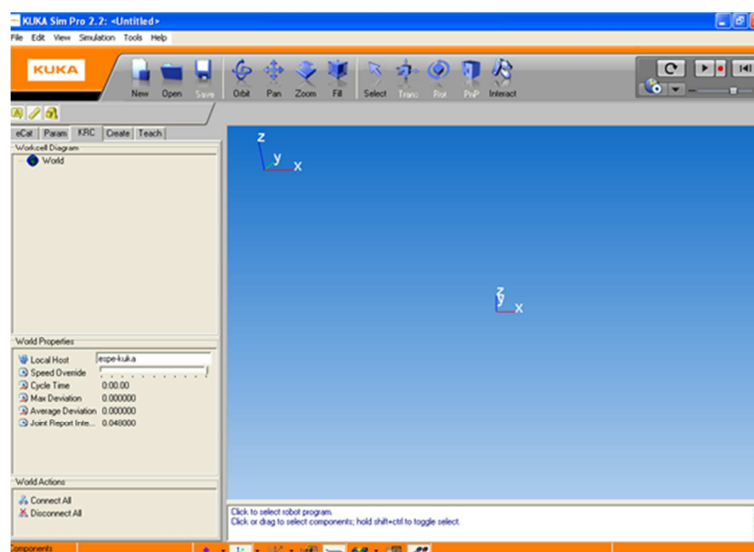


Figura 2.33: Interfaz KUKA SimPro

Seleccionar un manipulador robótico en KUKA Sim Pro 2.2

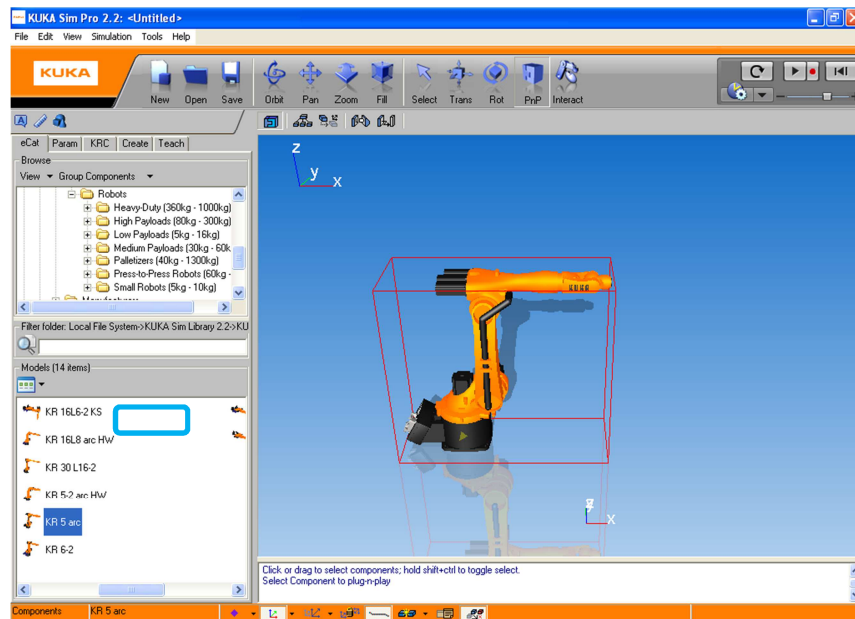


Figura 2.34: Manipulador robótico modelo KR 5arc

Seleccionar la pestaña KRC, seleccionar conectar.

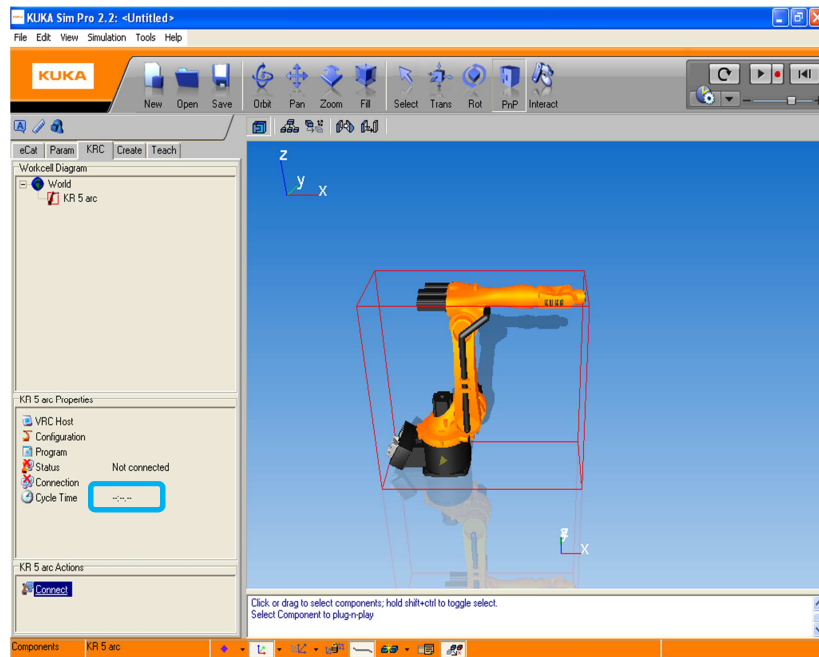


Figura 2.35: Comunicación entre software

Ingresar el nombre de la máquina “espe-kuka” (no el grupo de trabajo).

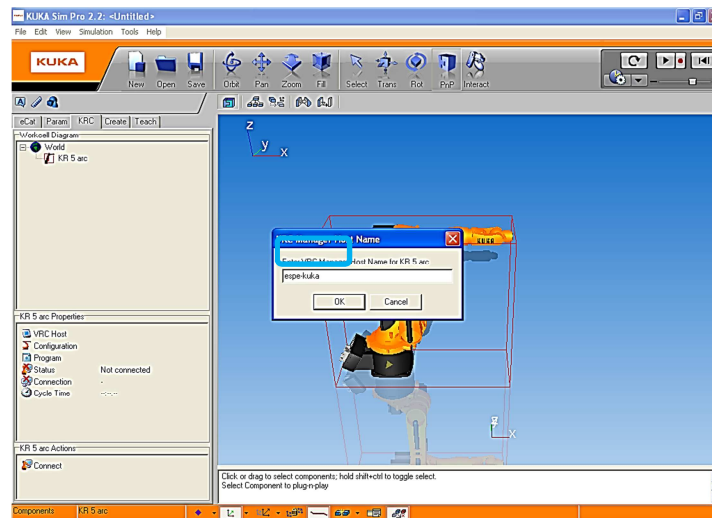


Figura 2.36: Ingreso del nombre de la máquina

Cancelar el mensaje.

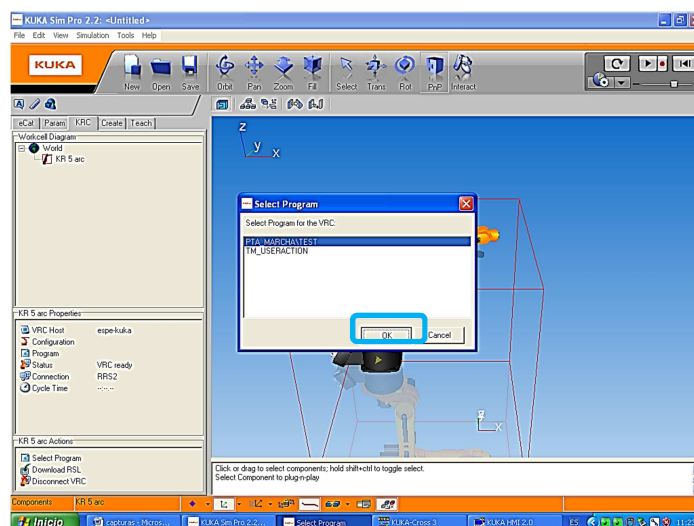


Figura 2.37: Mensaje de selección

Terminada la simulación al guardar el archivo se crearan el archivo.Dat y .SRC

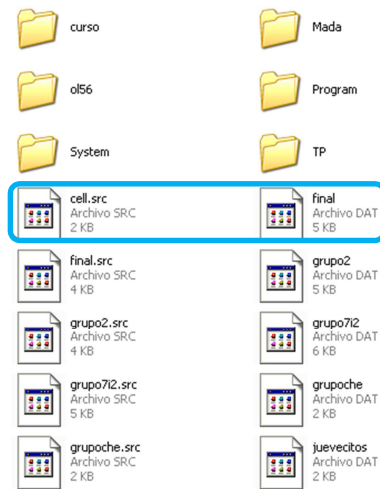


Figura 2.38: Archivos .Dat y .SRC

Tomar el archivo .Dat guardarlo en una flash memory y cargarlo en el robot real.

CAPÍTULO 3

GUÍAS DE LABORATORIO

3.1 IDENTIFICACIÓN DEL HARDWARE DEL MANIPULADOR ROBÓTICO KUKA KR 16 Y KR 5ARC

A. OBJETIVOS

Objetivo General:

- Realizar un primer acercamiento al manipulador robótico KUKA KR16 y KR5 ARC para afianzar los conocimientos teóricos.

Objetivos Específicos:

- Identificar los componentes que forman parte del hardware de manipulador robótico KUKA KR16 y KR5 ARC.
- Estudiar la parte mecánica del robot.

B. TRABAJO PREPARATORIO

1. Consultar sobre los manipuladores KR 16 y KR 5ARC.
2. Revisar y visualizar las características físicas y técnicas de los elementos que forman parte del sistema KUKA KR16 Y KAR ARC 5.

C. MARCO TEÓRICO

Manipulador Robótico KUKA KR 16

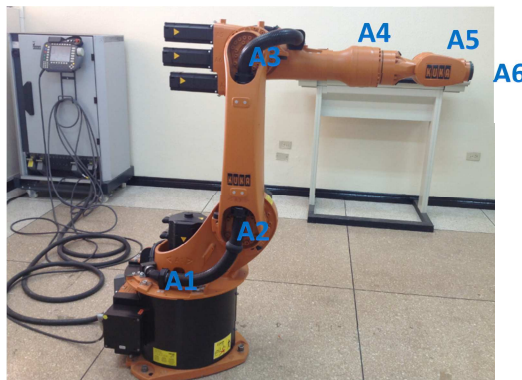


Figura 3.1: Ejes del robot KUKA KR 16

El robot KR 16 es ideal para transportar objetos para lo cual cuenta con las características presentadas en la Tabla 3.1

Tabla 3.1: Características del Robot

Carga útil	16 kg
Complementario carga útil	10 kg
Alcance máximo	1610 mm
Zona de trabajo	1911 mm
Número de ejes	6
Repetibilidad	$<\pm 0,05$ mm
Peso(sin unidad de control)	235 kg
Posiciones de montaje	Suelo, techo
Controlador	KR C2
Peso del controlador	aprox. 240 kg
Temperatura en servicio	+5 °C / +55 °C
NORMAS	CE – ISO

Superficie de colocación	500 mmx 500 mm
Conexión	7,3 kVA
Nivel de ruido	< 75 dB

Tabla 3.2 : Características de los ejes

Ejes	Campo Software	Velocidad con carga nominal 16 kg
Eje 1 (A1)	$\pm 185^\circ$	156°/s
Eje 2 (A2)	$+35^\circ/-155^\circ$	156°/s
Eje 3 (A3)	$+154^\circ/-130^\circ$	156°/s
Eje 4 (A4)	$\pm 350^\circ$	335°/s
Eje 5 (A5)	$\pm 130^\circ$	355°/s
Eje 6 (A6)	$\pm 350^\circ$	647°/s

Manipulador Robótico KUKA KR5 ARC



Figura 3.2: KUKA KR ARC5

El modelo KR5 ARC, es ideal para tareas de soldadura y cuenta con las características presentadas en la Tabla 3.3

Tabla 3.3: Características del Robot

Alcance máximo	1411 mm
Carga útil nominal	5 kg
Carga adicional	12–20 kg
Carga máx.	total 37 kg
Cantidad de ejes	6
Posición de montaje	Piso, techo
Repetibilidad de posición	$\pm 0,04$ mm
Unidad de control	KR C2
Peso (sin unidad de control)	127 kg
Temperatura en servicio	+10 °C hasta +55 °C
Tipo de protección	IP 54, IP 65 (Muñeca central)
Superficie de colocación robot	324 mm x 324 mm
Conexión	7,3 kVA
Nivel de ruido	< 75 dB

Tabla 3.4: Características de los ejes

Ejes	Campo Software	Velocidad con carga nominal 5 kg
Eje 1 (A1)	$\pm 155^\circ$	154°/s
Eje 2 (A2)	+65°/–180°	154°/s
Eje 3 (A3)	+158°/–15°	228°/s
Eje 4 (A4)	$\pm 350^\circ$	343°/s
Eje 5 (A5)	$\pm 130^\circ$	384°/s
Eje 6 (A6)	$\pm 350^\circ$	721°/s

Controlador KR C2

El controlador KR C2 integra todas las instalaciones de alimentación, seguridad y movimientos, con gran rendimiento, escalabilidad, flexibilidad, a la vez permite reducir los costes de integración y mantenimiento, aumenta la eficiencia. KUKA ha desarrollado una arquitectura de estándares abiertos de alto rendimiento en la que todos los controles tienen una base de datos y una arquitectura común.

Ventajas:

- Facilidad de manejo y mantenimiento.
- Rápido y fácil manejo gracias al empleo continuo de estándares.
- Comunicación en tiempo real entre los controles.
- Técnicas de seguridad integradas para campos de aplicación.
- Firewall de software integrado para mayor seguridad en red.
- Eficiencia energética optimizada.
- Alimentación Eléctrica 3x440V ~ (+10%).

CARACTERÍSTICAS FÍSICAS

- Carcasa compacta de acero inoxidable,
- Componentes duraderos que garantizan un trabajo fiable bajo largas jornadas de trabajo o condiciones de gran exigencia.
- El lenguaje KRL permite comunicarse con otros lenguajes como mecanizado CNC, controles PLC, unidades de control SIEMENS o Rockwell.
- El sistema de ventilación y refrigeración consigue reducir el consumo de energía en stand by adicionalmente logra que el sistema sea silencioso.



Figura 3.3: Unidad de control KR C2

CONSOLA KCP (KUKA Control Panel)

Es la interfaz de comunicación entre el usuario y el robot, ofrece una visión esquemática de los botones y símbolos. Contiene todas las funciones necesarias para el desplazamiento manual y programación de los movimientos. Controla todos los elementos de mando que se encuentran en el KR C2.

Cuenta con un diseño ergonómico que facilita su uso.

CARACTERISTICAS

- Tiene teclas de desplazamiento para el control manual.
- Movimientos y programación eficiente mediante ratón y teclado.
- La pantalla anti reflectante de alta resolución puede trabajarse sin forzar la visión.
- Indica el estado del sistema.
- Está integrado directamente con la PC.
- Controla los circuitos de seguridad.
- Precisa cursos de capacitación que permitan conocer las normas de seguridad.

KCP FRONTAL

1. Selector de modos de servicio
2. Accionamientos CONEXIÓN
3. Accionamientos DESCONEXIÓN
4. Pulsador de PARADA

EMERGENCIA

5. Space Mouse
6. Teclas de estado derecha
7. Tecla de entrada

Figura 3.4: KUKA Control Panel frontal

8. Teclas de dirección
9. Teclado Alfanumérico
10. Bloque numérico
11. Softkeys
12. Tecla de retroceso
13. Tecla de avance hacia adelante
14. Tecla de STOP
15. Tecla de selección de ventana
16. Tecla ESC
17. Teclas de estado izquierda
18. Teclas de menú

KCP POSTERIOR



1, 2 y 3 Pulsador de hombre muerto
4 Tecla de ejecución de programa

Figura 3.5: Parte posterior del KCP

INTERRUPTOR GENERAL

Permite energizar el sistema, este proceso se tarda alrededor de cuatro minutos igualmente al momento de apagar.

Sabremos que el sistema está listo cuando en la ventana de mensajes aparezca **DSE-Arranque finalizado.**



Figura 3.6: Interruptor de seguridad

SERVO CONVERTIDORES KSD1

Posee seis servos convertidores de frecuencia que permiten el control de velocidad, torque y posición de los servomotores de corriente alternada.

Los servos convertidores KSD1 son de alto desempeño y precisión en el control de movimiento de los ejes debido a la realimentación de posición que está dada por un sensor que se encuentra dentro del servomotor. Posee resistores de frenado que permiten tiempos de frenado muy reducidos. Las interfaces y protocolos de redes de comunicación permiten que su operación sea rápida y precisa, posibilitando su integración a diferentes sistemas de control y monitoreo.



Figura 3.7: Servo convertidores

FUENTE DE ALIMENTACIÓN DE POTENCIA (KPS)

La KUKA POWER SUPPLY, posee dos fuentes una de KPS 600V y una fuente de alimentación rectificadora a 27 VC, que alimenta a los frenos, reguladores de los motores, acumuladores y tarjetas electrónicas de seguridad.



Figura 3.8: Fuente de alimentación

TARJETA DE SEGURIDAD (KPS)

Integra todos los dispositivos periféricos mediante un bus, ofrece seguridad ante un mal funcionamiento, parada de emergencia, hombre muerto para lo cual se produce una desconexión de la alimentación de los accionamientos en la KPS y se origina una parada de los ejes.

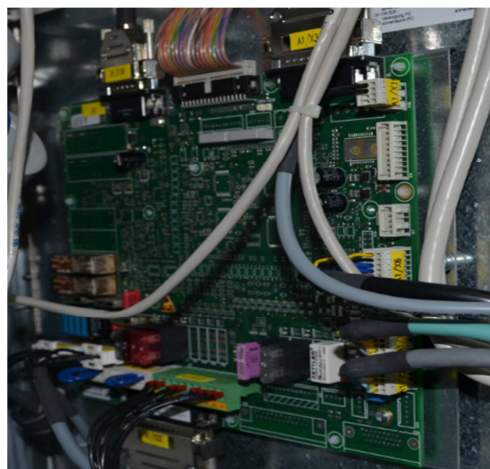


Figura 3.9: Tarjeta de seguridad

TARJETA MULTIFUNCIONAL DE INTERFACE (MFC)

Es una tarjeta multifunción que tiene el control de las entradas y salidas del robot mediante una conexión DeviceNet.

DeviceNet es un bus estándar de campo para comunicación entre dispositivos como sensores, actuadores y sistemas de automatización industriales.

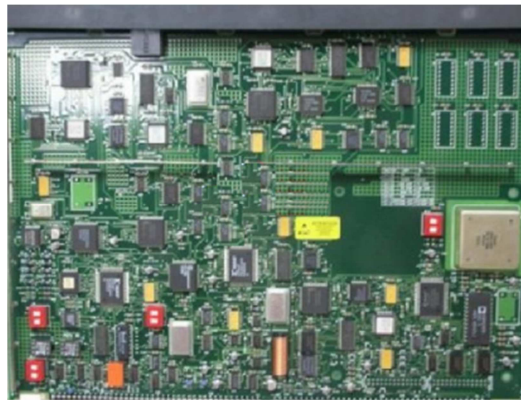


Figura 3.10: Tarjeta MFC

BATERIAS

Posee dos baterías que mantienen activa la memoria del sistema, suministran 19V. En caso de agotamiento se muestra un mensaje en la ventana de mensajes al arrancar el sistema. Para evitar una descarga completa de las baterías estas deben cargarse regularmente en función de la temperatura de almacenamiento.

Con una temperatura de almacenamiento de +20 °C o menos, las baterías deben cargarse cada 9 meses.

Con una temperatura de almacenamiento de +20 °C hasta +30 °C, las baterías deben cargarse cada 6 meses.

Con una temperatura de almacenamiento de +30 °C hasta +40 °C, las baterías deben cargarse cada 3 meses.

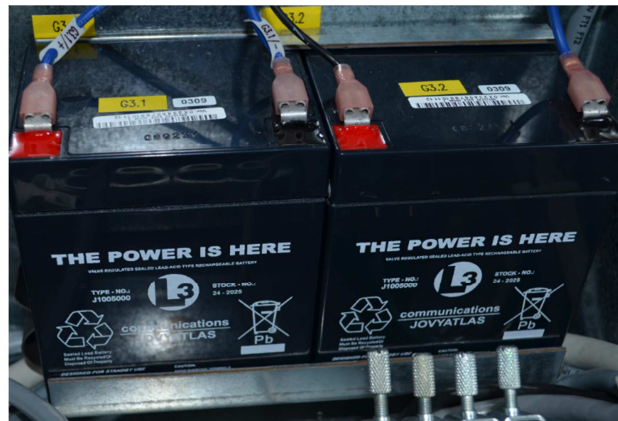


Figura 3.11: Baterías

MODULO I/O WAGO

El módulo WAGO de entradas y salidas digitales permite integrar al sistema dispositivos de campo como sensores, actuadores y lámparas, mediante un bus de campo Fieldbus que opera en tiempo real.

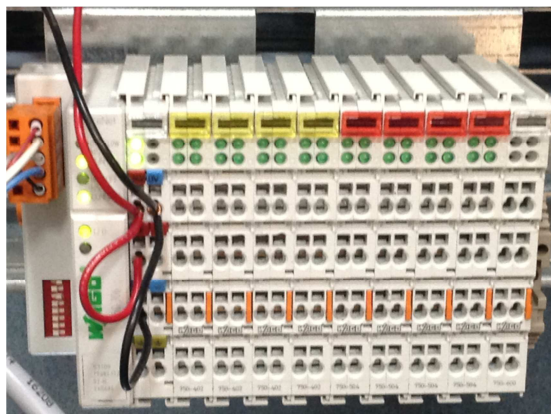


Figura 3.12: Módulo de entradas y salidas

CONECTOR DE SEGURIDAD (X11)

El conector X11 es un conector Harting macho. A través de esta interfaz se puede conectar dispositivos de seguridad, de protección del operario y bloqueo de dispositivos.

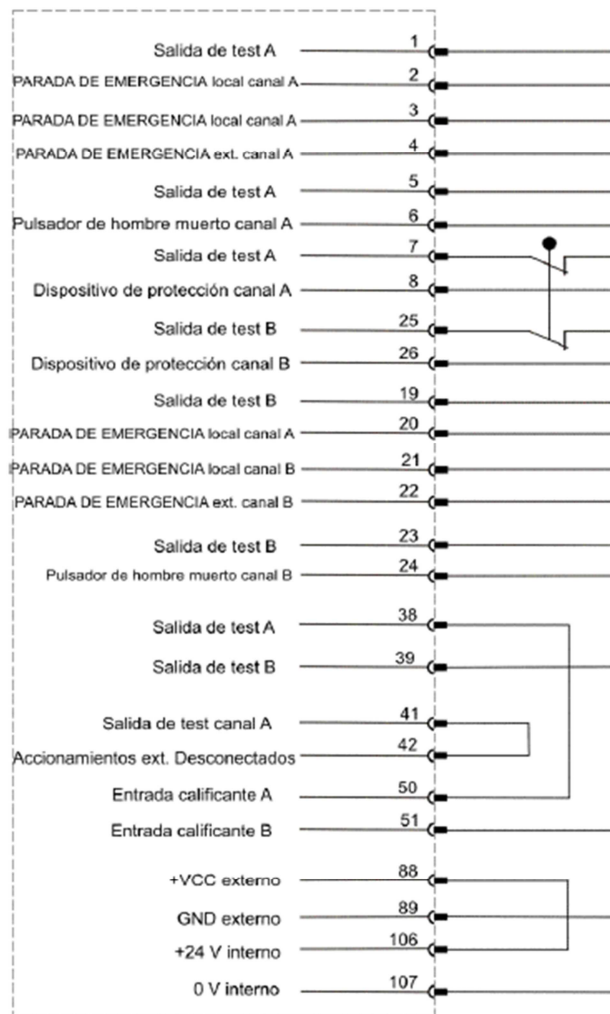


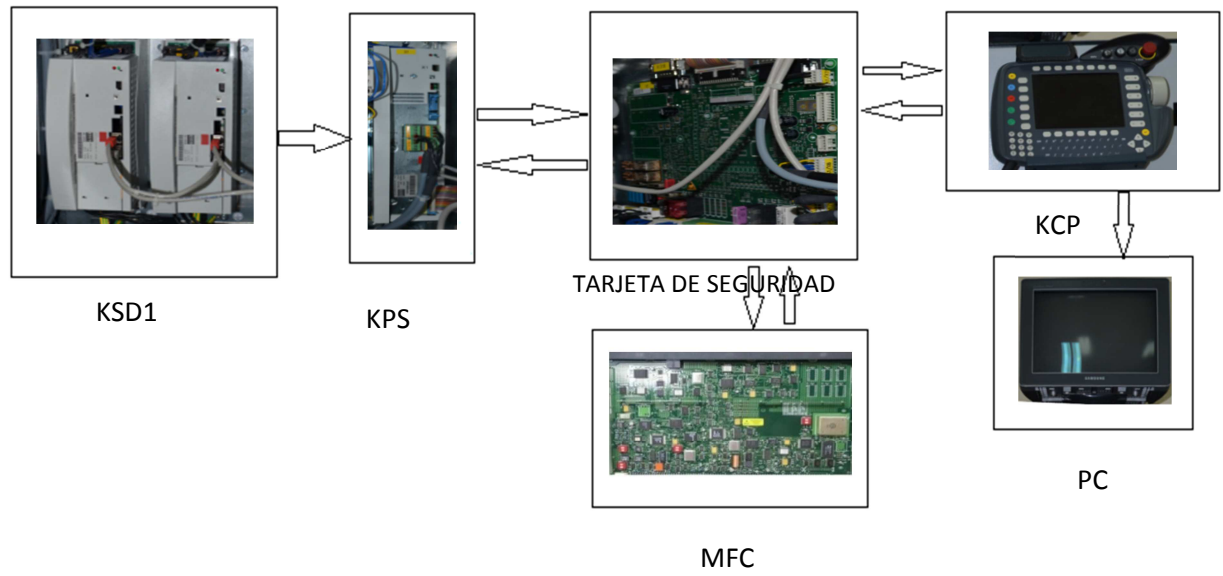
Figura 3.13: Diagrama de conectores X11

PC

La PC cumple con los siguientes requerimientos:

- Sistema operativo Windows embebido
- Memoria RAM de 256 MB
- Memoria HDD de 15 GB

DIAGRAMA DE CONEXIÓN DE DISPOSITIVOS



TERMINAL DE CONEXIÓN

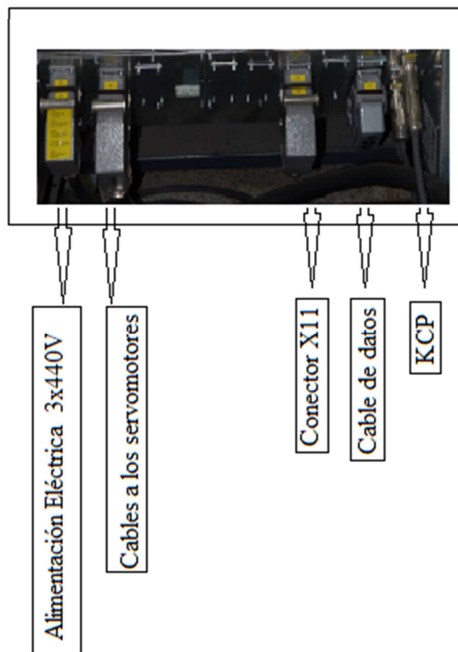


Figura 3.14: Conexión de dispositivos

MECÁNICA DEL ROBOT

Los componentes mecánicos del robot están fabricados en su mayor parte con aluminio y acero o fibra de carbono dependiendo de su aplicación.

La mecánica del robot de un sistema de KUKA tiene una estructura modular. Posee 6 ejes de rotación, los rangos de movimiento se delimitan por fines de carrera, en el caso de los ejes 1, 2, 3 y 5 poseen adicionalmente topes mecánicos con efecto amortiguado.

El diseño mecánico fue optimizado utilizando CAD (Diseño Asistido por Computadora) y FEM (Método de los Elementos Finitos), teniendo en cuenta una construcción liviana, de alta rigidez, torsión, flexibilidad y resistencia a las vibraciones.

Los ejes son accionados por servomotores de AC que constituyen la una unidad motriz que tiene integrada los frenos para mantener los ejes la posición deseada.

Grados de libertad

El robot posee seis grados de libertad, cada grado identifica el tipo de movimiento independiente que realiza cada eje.

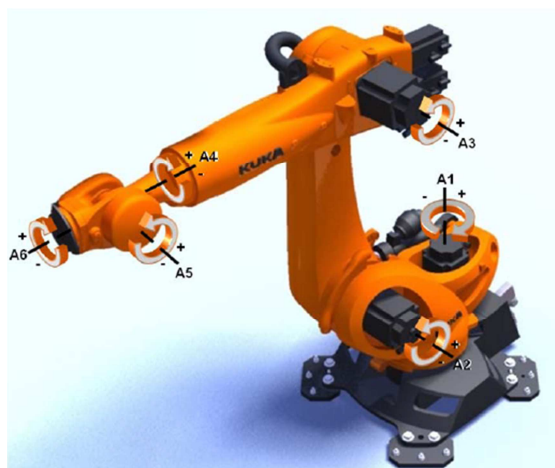


Figura 3.15: Grado de libertad del KUKA KR5 Y KR16 (Craig J. , 2008)

SEGURIDAD

Es fundamental que las personas que van a trabajar con el sistema, se instruyan sobre las posibles fuentes de peligro antes de asumir la responsabilidad de usar el robot.

El sistema del robot cumple con normativas de construcción de máquinas, certificación CE y normas de instalación de baja tensión.

Para evitar accidentes se debe tener en cuenta las siguientes recomendaciones.

- Trabajar a una distancia mínima de 2 metros del robot.
- Se debe vestir indumentaria de protección, especialmente, zapatos protectores y vestimenta ajustada al cuerpo.
- Se debe evitar el contacto con los motores porque alcanzan temperaturas altas durante el servicio y esto puede provocar quemaduras a la piel.
- Revisar que el override (velocidad) del programa que esté en un valor bajo (el máximo es 100%), especialmente cuando cambia de una subrutina a otra.
- Se debe iniciar un programa en T1(velocidad reducida), después de haberlo probado cambiar a otro modo de servicio.
- Para realizar trabajos de reemplazo, ajuste, mantenimiento, reparación, conexión del módulo WAGO debe colocar el interruptor general en la posición de DESCONECTADO.
- La desconexión, de los cables de alimentación del circuito intermedio dura hasta 5 minutos.
- Prestar atención a los textos y/o símbolos que se muestran en la ventana de mensajes.

El sistema del robot no admite:

- Transporte de personas o animales.
- Utilizar fuera de los límites de servicios permitidos.
- Utilización en ambientes con peligro de explosión.

En el caso de producirse peligro usar el **PARO DE EMERGENCIA y/o PULSADOR DE HOMBRE MUERTO**, según el modo de servicio en el que se trabaje.

D. PROCEDIMIENTO

1. Verifique las características físicas y las características técnicas de la placa del robot.
2. Identifique los ejes que determinan los grados de libertad y describa los movimientos que produce cada eje.
3. Ubique cada uno de los elementos del sistema del robot y encuentre características importantes adicionales al anexo.
4. Realizar una analogía referente al software, hardware y funcionamiento del sistema robótico KUKA y CRS255.

3.2 FUNCIONES DE LAS TECLAS DEL KCP Y MOVIMIENTO MANUAL EN LOS SISTEMAS DE COORDENADAS

A. OBJETIVOS

Objetivo General:

- Operar los manipuladores robóticos industriales y comprobar las funciones de las teclas del KCP.

Objetivos Específicos:

- Utilizar el KCP (KUKA Control Programmer) y el Space Mouse para el desplazamiento manual.
- Realizar movimientos incrementales con el manipulador en todos los sistemas de coordenadas.

B. TRABAJO PREPARATORIO

1. Investigue los sistemas de coordenadas en los que trabajan los manipuladores robóticos y genere un informe.
2. Revisar el marco teórico de esta práctica.

C. MARCO TEORICO

Arranque del sistema

El sistema presenta dos tipos de arranques, en caliente y frío. El arranque en caliente re establece las salidas e interface al punto en que se trabajó la última vez. Mientras que el arranque en frío re establece todo el sistema por completo. El arranque en Hibernación, tiene un comportamiento idéntico a un arranque en caliente.



Sistema de coordenadas

Para orientar los movimientos y la posición que realiza un manipulador robótico se utiliza un sistema de coordenadas cartesianas.

La posición de un punto en un sistema de coordenadas cartesianas, se define como la distancia del origen a las proyecciones ortogonales sobre cada uno de los ejes x,y,z del plano cartesiano. El sistema de coordenadas cartesiano es un sistema de referencia formado por rectas numéricas que se cortan perpendicularmente en un punto de origen.

KUKA ha definido los siguientes sistemas de referencias mencionados en la Tabla 3.5

Tabla 3.5: Sistema de Referencia

ÍCONO	SISTEMA DE REFERENCIA	DEFINICIÓN
	EJES ESPECÍFICOS	Desplaza en forma independiente cada uno de los ejes u articulación del robot. El sistema de coordenadas se encuentra en cada eje.
	WORLD	El sistema de coordenadas WORLD se encuentra en el pie del robot.
	TOOL	El origen de coordenadas se encuentra en la punta de la herramienta. Hace referencia al sistema de coordenadas BASE.
	BASE	Es un sistema de coordenadas cartesianas que describe la posición de la base.

Ejes del robot: Es un elemento destinado a guiar el movimiento de rotación de una pieza o de un conjunto de piezas, como una rueda o un engranaje. En el robot es el punto donde se describe los movimientos lineales o circulares sobre un sistema de coordenadas cartesianas.

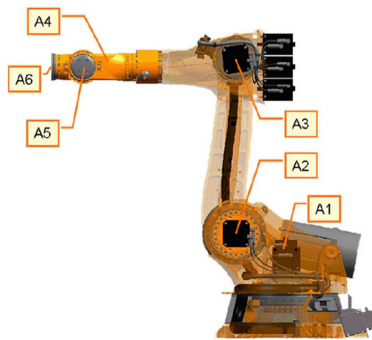


Figura 3.16: Ejes del Robot (KUKA GROUP, 2011)

Se puede realizar diferentes tipos de movimiento de los ejes según su configuración a continuación se detalla en la Tabla 3.6

Tabla 3.6: Sistema de Referencia

CONFIGURACIÓN DE EJES	DESCRIPCIÓN
XYZ	Movimientos de traslación en las direcciones X,Y, Z
ABC	Movimientos de rotación alrededor de los ejes X, Y, Z
5D	Movimientos de traslación en las direcciones X,Y,Z Movimientos de rotación alrededor de los ejes X, Y, Z

Los ejes pueden ser accionados simultáneamente eso depende de la configuración de mouse space. Los tipos de configuración son:

- **Dominante:** permite desplazar solo un eje con el Space- Mouse.
- **No dominante:** permite desplazar 3 o 6 ejes al mismo tiempo.

Orientación del Space Mouse

La función del Space Mouse puede ser adaptada a la posición del usuario, para que la dirección de desplazamiento del TCP (tool center point) corresponda a la desviación del Space Mouse.

La posición del usuario se indica en grados. El ajuste por defecto es 0° , esto corresponde a un usuario que se encuentra colocado frente a la caja de conexiones.

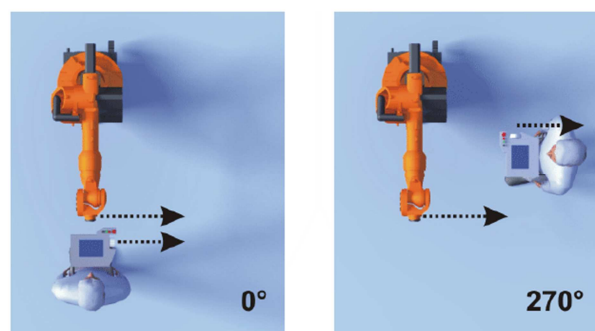


Figura 3.17: Space mouse en 0° y 270° (KUKA, 2009)

Interfaz de usuario KUKA.HMI

La interfaz KUKA.HMI está subdividida en varias zonas que desempeñan diferentes operaciones para el desarrollo de tareas.

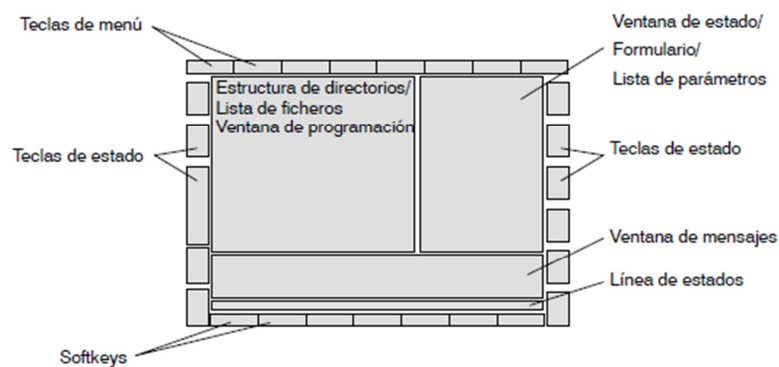


Figura 3.18: Interfaz KUKA.HMI

D. PROCEDIMIENTO

Configuración de las condiciones previas

1.1 Accionar el switch del gabinete de control KR C2, el sistema arrancará en modo de Hibernación.



Figura 3.19: Switch de alimentación principal

1.2 Seleccionar el modo de servicio T1 que permite el trabajo manual con velocidad reducida.



Figura 3.20: Modo de servicio T1

- Esperar que aparezca el mensaje **DSE-Arranque finalizado**.
- Verificar que el botón de paro de emergencia no este accionado, caso contrario desactivarlo.



Figura 3.21: Paro de emergencia

- Presionar softkey **Confirmar todos**, de esta manera se aceptan los mensajes de accionamientos preparados y paro de emergencia que se muestran en la **Ventana de mensajes**.

2. Desplazamiento con teclado

2.1 Seleccionar el **tipo de desplazamiento con teclado** con la primera tecla de estado del lado izquierdo y con la tecla de estado que se encuentran en el lado derecho seleccionar opción de desplazamiento de **eje específico**.

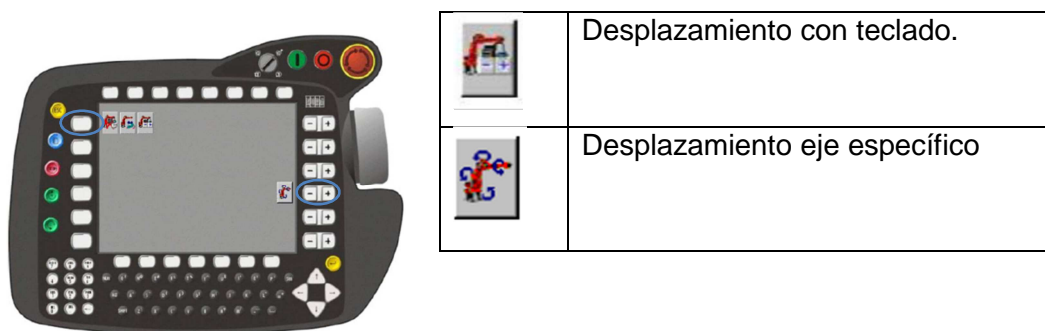


Figura 3.22: Selección desplazamiento eje específico con teclado

2.2 Presionar la tecla de hombre muerto y pulsar la tecla “+” y “-” para mover los ejes en dirección positiva o negativa respectivamente, realizar la acción para los seis ejes con x, y, z, a, b, c.



Figura 3.23: Teclas de accionamiento de ejes.

2.3 Cambiar la velocidad de los ejes a 10% con la tecla indicada en la Figura 3.24 y repetir el procedimiento 2.2.



Figura 3.24: Tecla de override

2.4 Realizar movimientos incrementales: Presionar la tecla de estado del lado derecho que indica la figura 3.25, seleccionar en incremento de 100 mm o 10 y repetir el procedimiento 2.2

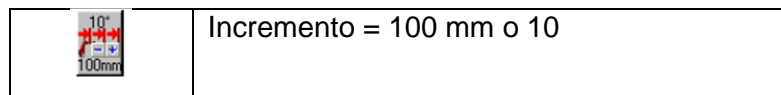


Figura 3.25: Tecla de incremento

3. Desplazamiento con mouse space



3.1 Mover cada eje con el space mouse: Escoger tipo de desplazamiento mouse space. Seleccionar en la barra de menú **Configuración > Movimiento manual > Configuración de puntero > 6D y modo dominante.**

Presionar el botón de hombre muerto y a la vez mover o empujar el Space Mouse para su desplazamiento según la Figura 3.26.

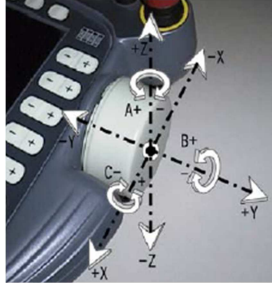


Figura 3.26: Ejes para movimiento de articulaciones.

E. TRABAJO COMPLEMENTARIO

1. Realizar el procedimiento 2.2 para los valores de 1%, 10%, 30%, 50%, 75%, 100%.
2. Realizar el procedimiento 2.2 para el incremento en 0,005 , 1 , 3 y continuo.
3. Seleccionar el modo de desplazamiento WORLD, BASE y TOOL y realice el procedimiento 2.2, 2.3 y 2.4
4. Accionar el **modo no dominante** y realizar el procedimiento 3.1, 3.2 y 3.3

3.3 CONFIGURACIÓN Y MEDICIÓN DE LA BASE Y HERRAMIENTA DEL ROBOT

A. OBJETIVOS

Objetivo General:

- Configurar el origen de coordenadas de la base de trabajo y herramienta del robot.

Objetivos Específicos:

- Utilizar los diferentes métodos de configuración de base y herramienta.

B. TRABAJO PREPARATORIO

1. Revisar el algoritmo de Denavit y Hartenberg.
2. Consultar ¿Qué es cinemática directa?
3. Consultar los tipos de herramientas terminales para robots.

C. MARCO TEORICO

Base: es un área específica de trabajo donde se crea un nuevo sistema de coordenadas. La configuración de la base consiste en definir el origen y los ejes cartesianos. Este será asignado por el operador según su requerimiento o proceso.

Se pueden guardar como máximo 32 sistemas de coordenadas BASE

Método de los tres puntos

1. Definición del origen
2. Definición de la dirección X positiva
3. Definición de la dirección Y positiva

Herramienta: O gripper es montada en la brida de acople del robot. El origen es un punto definido por el usuario, dicho punto se denomina TCP (Tool Center Point), este es el punto de trabajo de la herramienta.

Se pueden guardar como máximo 16 sistemas de coordenadas TOOL

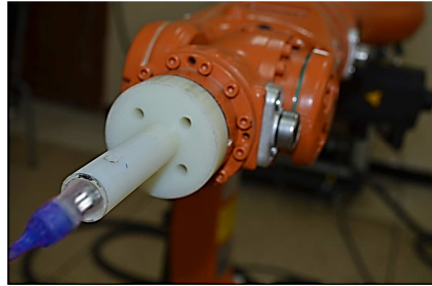


Figura 3.27: Herramienta del KUKA

Posición de la herramienta en un sistema de coordenadas

Medición XYZ de 4 puntos: Sirve para determinar el origen del sistema de coordenadas de la herramienta. Este método permite que el TCP de la herramienta con la que se va a trabajar se desplace a un punto de referencia desde 4 direcciones diferentes. El punto de referencia puede ser cualquiera.

En este método se efectúa la ubicación de una nueva herramienta mediante una herramienta auxiliar que define un punto preciso. La unidad de control del robot compara guarda las posiciones del TCP de la nueva herramienta

Orientación de la herramienta en un sistema de coordenadas

Método ABC World: Este método permite definir la orientación del sistema. Los ejes del sistema de coordenadas World se definen de forma paralela a los ejes del sistema de coordenadas TOOL. De este modo, la unidad de control del robot conoce la orientación del sistema de coordenadas TOOL.

Este método tiene 2 variantes:

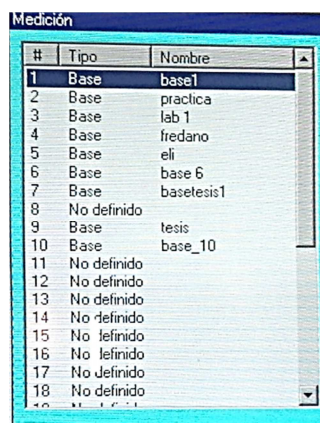
5D: A la unidad de control sólo se le declara la dirección de trabajo de la herramienta. Por defecto, la dirección de trabajo es el eje X. La dirección de los demás ejes la determina el sistema.

6D: A la unidad de control del robot se le comunican las direcciones de los 3 ejes.

D. PROCEDIMIENTO

1. Configuración de las condiciones previas

- Accionar el switch del gabinete de control KR C2, el sistema arrancará en modo de Hibernación.
- Seleccionar el modo de servicio T1 que permite el trabajo manual con velocidad reducida.
- Esperar que aparezca el mensaje DSE-Arranque finalizado.
- Verificar que el botón de paro de emergencia no este presionado, caso contrario deshabilitar y seleccionar el softkey Confirmar.
- Confirmar todos los mensajes que se muestran en la Ventana de mensajes.
- Revisar las bases disponibles. Seleccionar en las teclas de menú **Configurar>Definición de herramienta> Tipo de base.**



The screenshot shows a window titled 'Medición' with a table listing available bases. The table has three columns: '#', 'Tipo', and 'Nombre'. The rows are numbered 1 through 18. Rows 1-7 are of type 'Base' with various names. Rows 8-18 are of type 'No definido'.

#	Tipo	Nombre
1	Base	base1
2	Base	practica
3	Base	lab 1
4	Base	fredano
5	Base	eli
6	Base	base 6
7	Base	basetesis1
8	No definido	
9	Base	tesis
10	Base	base_10
11	No definido	
12	No definido	
13	No definido	
14	No definido	
15	No definido	
16	No definido	
17	No definido	
18	No definido	

Figura 3.28: Lista de bases disponibles

- Revisar las herramientas disponibles. Seleccionar en las teclas de menú **Configurar>Definición de herramienta> Tipo de herramientas**

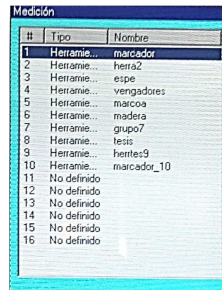


Figura 3.29: Lista de herramientas disponibles

2. Configuración de la Base

- Seleccionar en las teclas de menú la secuencia **Inicial. > Medición > Base > 3-Puntos**
- Indicar un número y un nombre para la nueva base y presionar el softkey **Continuar.**

Base Num: 7

Nombre del sistema base: basetesis1

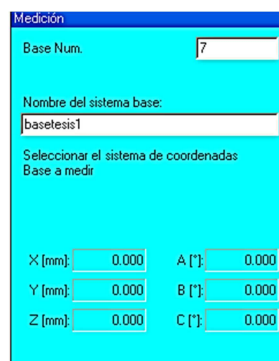
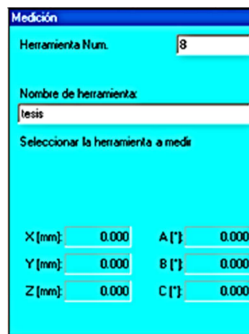


Figura 3.30: Número y nombre de la base

- Colocar el nombre de la herramienta y presionar el softkey **Continuar.**

Herramienta Num: 8

Nombre de la herramienta: tesis



Medición			
Herramienta Num.	8		
Nombre de herramienta:	tesis		
Seleccionar la herramienta a medir			
X [mm]:	0.000	A [°]:	0.000
Y [mm]:	0.000	B [°]:	0.000
Z [mm]:	0.000	C [°]:	0.000

Figura 3.31: Número y nombre de la herramienta

- Desplazar el TCP a la nueva base (mesa de trabajo), seleccionar el punto de origen del nuevo sistema. Pulsar el softkey **Medición** y **continuar**.

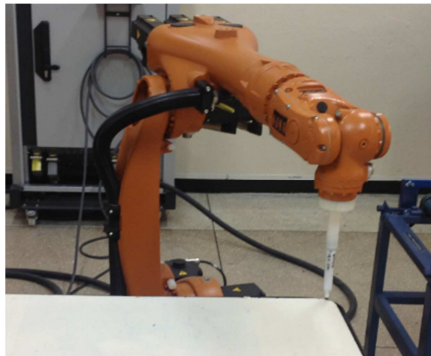


Figura 3.32: Punto de origen

- Desplazar TCP a un punto del eje X positivo del nuevo sistema. Pulsar **Medir** y **Continuar**.



Figura 3.33: Eje X positivo

- Desplazar a un punto del plano XY (positivo). Pulsar **Medir** y **Continuar**.

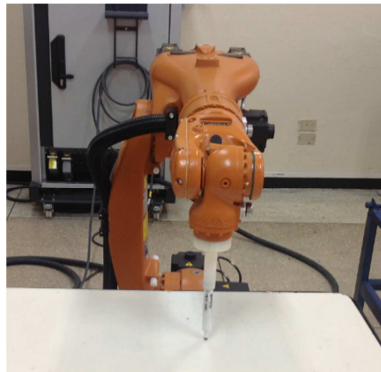


Figura 3.34: Eje XY positivo

- Realizar un movimiento y verificar las coordenadas del robot en la nueva base en **Indicación>Posición actual> Cartesiano**

Posición de robot			Posición de robot		
Nombre	Valor	Unidad	Eje	Pos. [grad, mm]	Incrementos
Herramienta/Base			A1	0,00	-17
tesis (8)	#BASE	Tool	A2	-90,00	-512013
basetesis1 (7)	#BASE	Base	A3	90,00	356360
Posición			A4	0,00	10
X	331,36	mm	A5	-0,01	-8
Y	237,38	mm	A6	0,00	2
Z	-172,27	mm			
Orientación					
A	1,98	gra			
B	3,48	gra			
C	27,66	gra			
Posición de robot					
S	110	bin			
T	010011	bin			

Figura 3.35: Posición del Robot en ejes Cartesiano y Especificos

3. Configuración de la Herramienta

- Seleccionar la secuencia en las teclas de menú **Inicial. > Medición > Herramienta > XYZ 4-Puntos**
- Asignar un número y un nombre para la herramienta que se desea medir. Confirmar con **OK**. Los demás casilleros se quedaran con los datos por default.

- Acercar el TCP al punto de referencia de la herramienta conocida (la punta del marcador ubicado en la base). Pulsar la tecla **Medir** y **Continuar**.

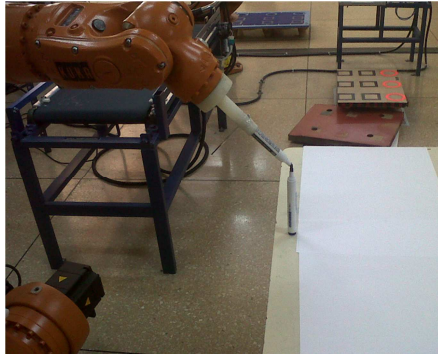


Figura 3.36: Primera medición

- Con el TCP desplazarse a un punto de referencia desde otra dirección. Pulsa de nuevo la tecla **Medir** y **Continuar**.

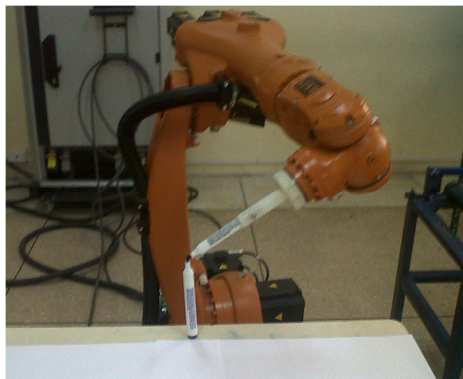


Figura 3.37: Segunda medición

- Con el TCP desplazarse a otro punto de referencia desde otra dirección. Pulsa de nuevo la tecla **Medir** y **Continuar**.

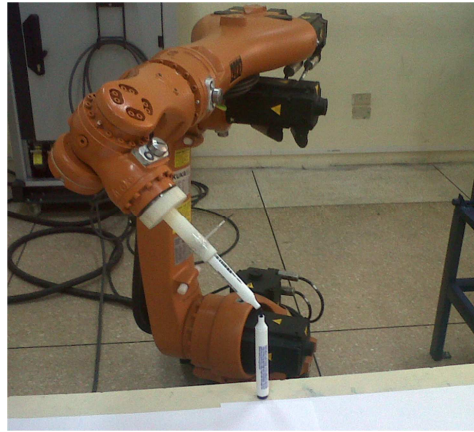


Figura 3.38: Tercera medición

- Con el TCP desplazarse al otro punto de referencia desde otra dirección. Pulsa de nuevo la tecla **Medir** y **Continuar**.

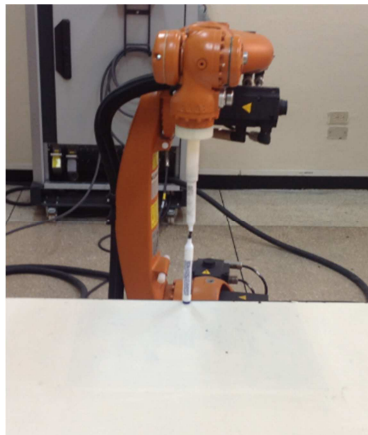


Figura 3.39: Cuarta medición

- Seleccionar la secuencia de menú **Inicial. > Medir > Herramienta > ABC World**.
- Colocar el número y un nombre de la herramienta que se midió. Confirmar con OK.
- Seleccionar el eje X y el campo 5D. Confirmar con OK.

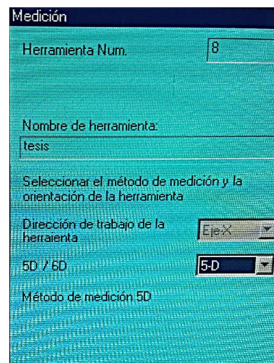


Figura 3.40: Campo 5D y eje X

- Realizar una medición paralela con la herramienta fija y la herramienta auxiliar.

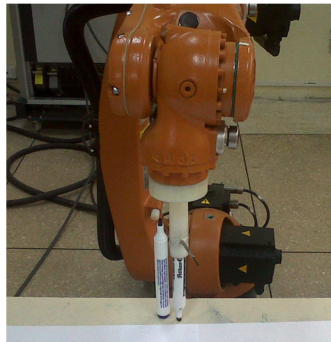


Figura 3.41: Posición paralela de las herramientas

- Actualizar la herramienta en **Configuración>Actual Herramienta/Base**.

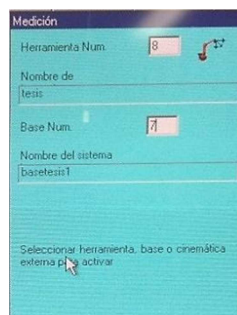



Figura 3.42: Actualización de base y herramienta

E. TRABAJO COMPLEMENTARIO

1. Realizar movimientos con la herramienta y verificar las coordenadas.

- Seleccionar el **Sistema de coordenadas TOOL** 
- Abrir la ventana Posición del robot en **Indicación>Posición actual>Cartesiano.**
- Mover el eje a, observar la posición y orientación.
- Mover el eje b, observar la posición y orientación.
- Mover el eje c, observar la posición y orientación.

3.2 PROGRAMACIÓN DE TRAYECTORIAS DEL MANIPULADOR ROBÓTICO EN MODO USUARIO

A. OBJETIVOS

Objetivo General:

- Crear programas mediante formularios de trayectorias que permitan cumplir con movimientos deseados en modo usuario.

Objetivos Específicos:

- Crear y modificar movimientos utilizando instrucciones y formularios.

B. MARCO TEÓRICO

MODO USUARIO

El modo usuario es sencillo facilita en la selección y creación de programas pero limita las modificaciones en el.

El programador controla los movimientos del robot con la ayuda de instrucciones de trayectorias.

NAVEGADOR PARA ADMINISTRACIÓN DE FICHEROS

El navegador en modo usuario permite administrar programas y todos los ficheros específicos del sistema, seleccionándoles según el tipo.

El navegador está compuesto de cuatro secciones. Dentro del navegador se utilizan los siguientes iconos o símbolos: como muestra la Figura 3.43.

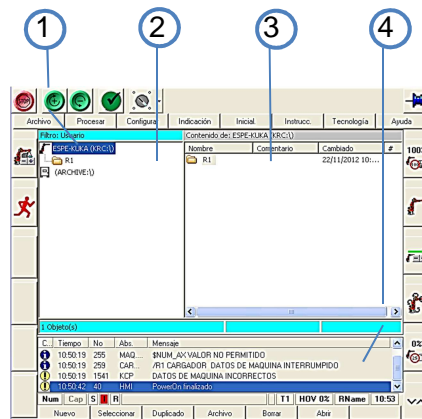


Figura 3.43: Navegador del Operario

- ① Encabezamiento : Se visualiza el directorio o la unidad de disco.
- ② Estructura de directorios : Vista general sobre directorios y unidades de discos estos son visualizados dependiendo del grupo de usuarios y de la configuración.
- ③ Lista de ficheros: Muestra los programas dependiendo de la carpeta seleccionada.
- ④ Línea de estados

Indica información como:

- Objetos marcados
- Acciones en ejecución
- Preguntas de seguridad

TIPOS DE ARCHIVOS

Un programa puede constar de uno o varios archivos de programa.

Archivo .SRC

- Permite declaraciones externas para subprogramas y funciones que pueden ser utilizadas en otro archivo SRC.
- Permite declaraciones de señales de entrada o salida, que pueden ser utilizadas en el archivo SRC.

Lista de datos .DAT

- Llevaran el mismo nombre que el archivo SRC y termina con la extensión “.DAT”.
- En una lista de datos sólo pueden encontrarse declaraciones e inicializaciones.
- No se acepta ninguna variable del sistema.

Tipo de plantillas para programas

Modul: Crea un archivo SRC y un archivo DAT, contiene una estructura base DEF y END.

Expert: Crea un archivo SRC y un archivo DAT, contienen solo el encabezamiento DEF y END.

Cell : Crea un archivo .SRC que contiene un programa base. Permite el mando del robot a través de un PLC.

Function: Crea una archivo SRC que contiene el encabezamiento DEF... y END. Permite el mando para un PLC.

Submit: El fichero Submit contiene instrucciones que puede ser utilizado para programación de grippers. El fichero Submit trabaja en forma paralela al robot y es ejecutado por la unidad de control.

Expert Submit: Crea un fichero SUB, adicionalmente contiene el encabezamiento DEF... y END.

TIPOS DE MOVIMIENTOS

Movimiento PTP

En el movimiento Point To Point la herramienta se desplaza describiendo un movimiento a un punto de destino, tomando la trayectoria más rápida. Los ejes se mueven de manera sincronizada.

Este movimiento tiene dos tipos de posicionamiento:

- Parada Exacta: El robot se desplaza al punto de destino con posicionamiento exacto.
- Con posicionamiento aproximado: Utiliza un entorno de aproximación alrededor del punto de destino.

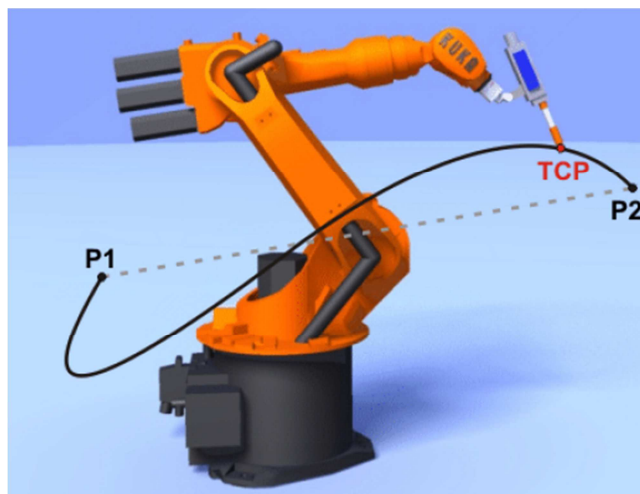


Figura 3.44: Movimiento PTP (KUKA, 2011)

Movimiento LIN

En un movimiento lineal los ejes del robot se coordinan entre sí de tal manera que el TCP, se mueve a lo largo de una recta desde un punto de origen hacia un punto de destino.

Este movimiento tiene dos tipos de posicionamiento:

- Movimiento LIN con parada exacta, el robot se desplaza a cada uno de los puntos de destino con posicionamiento exacto.
- En el posicionamiento aproximado en una instrucción LIN, la unidad de control genera un entorno de aproximación alrededor del punto. En cuanto se ha alcanzado ese entorno, los movimientos de los ejes pasan hacia el punto siguiente.

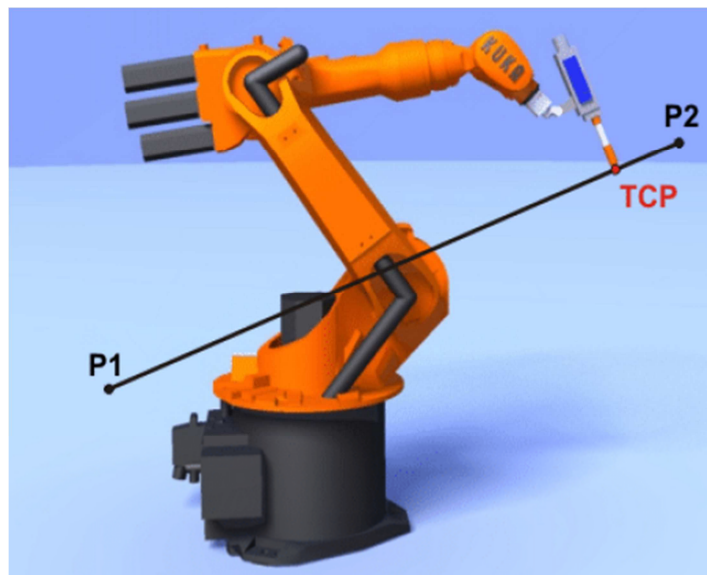


Figura 3.45: Movimiento LIN (KUKA, 2011)

Movimiento CIRC

El movimiento CIRC, se desplaza de un punto de referencia del TCP a un punto de destino con trayectoria circular. El recorrido se describe mediante un punto de inicio, un punto final. Como punto de inicio se puede tomar el punto de destino de la instrucción del movimiento anterior.

- Movimiento CIRC con parada exacta, el robot se desplaza a cada uno de los puntos de destino con posicionamiento exacto.
- Movimiento CIRC con posicionamiento aproximado, la unidad de control utiliza un entorno de aproximación alrededor del punto de destino.

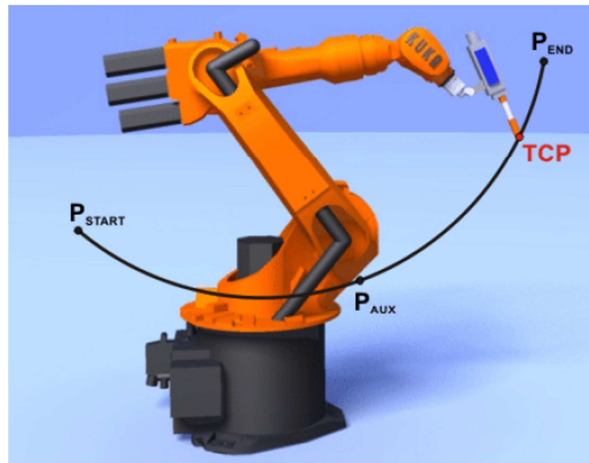


Figura 3.46: Movimiento CIR (KUKA, 2011)

Movimiento SPLINE

SPLINE describe movimiento apropiado para trayectorias curvas, se puede obtener resultados similares con las trayectorias LIN y CIRC aproximados. SPLINE solo trabaja en modo experto.

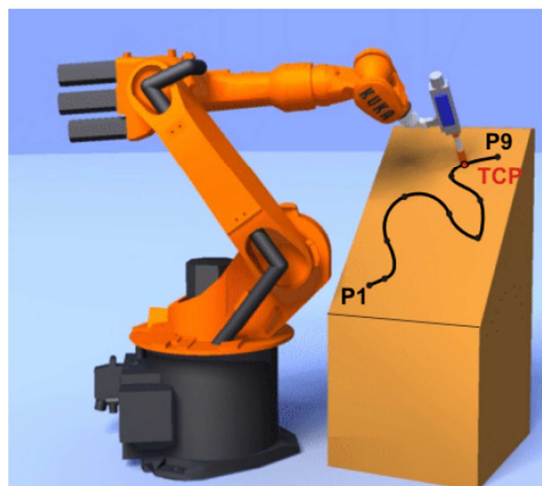


Figura 3.47: Movimiento SPLINE (KUKA, 2011)

Para determinadas trayectorias con el movimiento Spline debe cumplir la siguiente relación: $2/3 \leq a/b \leq 3/2$ o $0.666 \leq a/b \leq 1.5$

a = Es la distancia del punto inicial del segmento SPL al punto de intersección de los segmentos.

b = Es la distancia del punto de intersección de los segmentos SLIN al punto de destino del segmento.

Ejemplo:

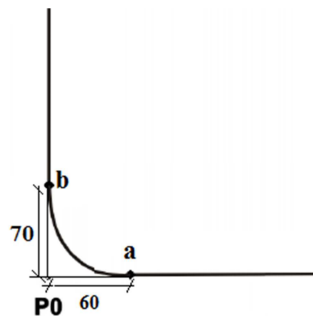


Figura 3.48: Movimiento SPLINE

— —

Posición HOME

La posición HOME por regla general, se la utiliza como primera y última posición en el programa.

La posición HOME está declarada por defecto en la unidad de control del robot con los siguientes valores.

Eje A1= 0

Eje A2= -90

Eje A3= +90

Eje A4= 0

Eje A5= 0

Eje A6= 0

C. ACTIVIDADES

1. Crear movimientos PTP, LIN, CIRC con la ayuda de formularios.
2. Realizar un cuadrado de dimensiones 10x10 cm sobre la base tomando como referencia los p1,p2,p3 y p4 inscritos en la Figura 3.49. Utilizar las trayectorias PTP y LIN, con una velocidad de 50% y 0.2 m/s respectivamente.

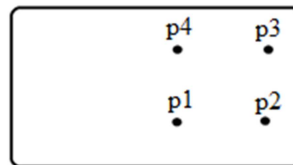


Figura 3.49: Puntos de referencia para el cuadrado

- Realizar las condiciones que indica el diagrama de flujo.

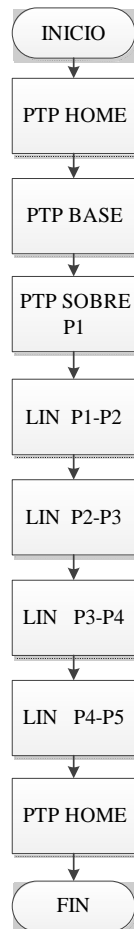


Figura 3.50: Diagrama de flujo cuadrado

3. Realizar un círculo de radio 6cm. Tomando como referencia los puntos p1,p2,p3,p4 que indica la figura 3.51. Utilizar las trayectorias PTP y CIRC, con una velocidad de 100% y 2 m/s respectivamente.

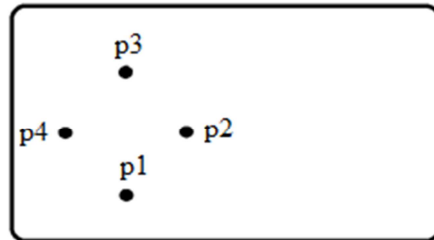


Figura 3.51: Puntos de referencia para la circunferencia

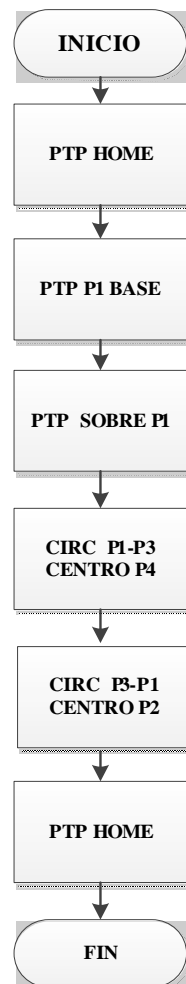


Figura 3.52: Diagrama de flujo círculo

D. PROCEDIMIENTO DE LA ACTIVIDAD 1

Configuración de las condiciones previas

- Accionar el switch del Gabinete de control KR C2, el sistema arrancará en modo de Hibernación.
- Seleccionar el modo de servicio T1 que permite el trabajo manual con velocidad reducido.
- Esperar que aparezca el mensaje **DSE-Arranque finalizado**.
- Verificar que el botón de paro de emergencia no este presionado, caso contrario deshabilitar y seleccionar el softkey Confirmar.
- Confirmar todos los mensajes que se muestran en la **Ventana de mensajes**.

Crear una nueva carpeta

- Seleccionar en el navegador, las teclas de menú **Archivo>Nuevo**
- Introducir un nombre de la carpeta y presionar **OK**.

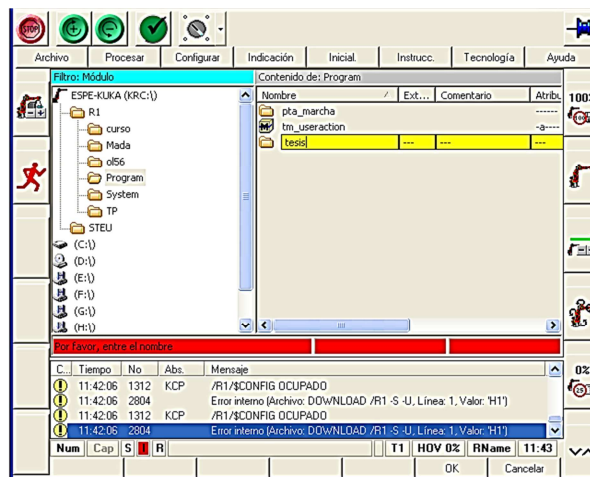


Figura 3.53: Nueva Carpeta

Crear un nuevo programa

- Abrir la carpeta creada

- Seleccionar en el navegador **Archivo>Nuevo**
- Introducir el nombre del programa y presionar **OK**. Tomar en cuenta las siguientes restricciones:
 - Longitud máxima 23 caracteres
 - No se permiten signos especiales excepto **_** y **\$**.
 - No está permitido colocar un número en el primer lugar.

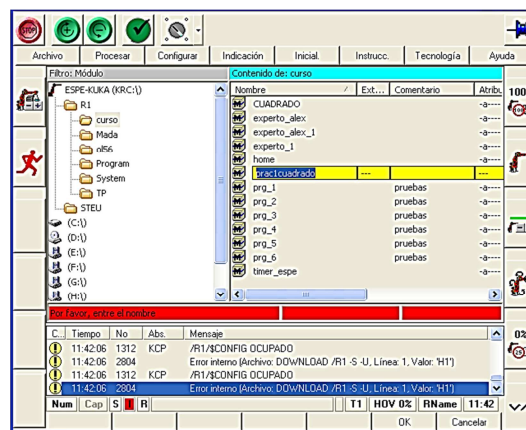


Figura 3.54: Nuevo Programa

Seleccionar el sofkey **abrir**. Aparece la pantalla que muestra la figura 3.55, la primera línea se presenta por default con la posición HOME.

```

1 DEF home ( )
2 INI
3
4 PTP HOME Ve1= 100 % DEFAULT
5
6 PTP HOME Ve1= 100 % DEFAULT
7
8 END

```

Figura 3.55: Posición HOME

Programar un movimiento PTP

- Mover el TCP a la posición que se programará como punto de destino.
- Seleccionar la secuencia **Instrucc. > Movimiento > PTP** o el sofkey **PTP** que se encuentra en la parte inferior izquierda.

- Declarar los parámetros en el formulario Inline.

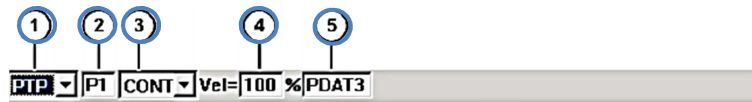


Figura 3.56: Formulario Inline configuración PTP

- ① Tipo de movimiento PTP
- ② Nombre del punto de destino (el sistema asigna automáticamente un nombre, sin embargo se puede renombrar)
- ③ CONT: Indica posicionamiento aproximado, cuando el campo está vacío indica el posicionamiento exacto.
- ④ Velocidad 1% ... 100%
- ⑤ Nombre archivo de destino (el sistema asigna automáticamente un nombre)

- Guardar la instrucción con **Instrucción OK**.

```

1 DEF home( )
2 INI
3
4 PTP HOME Vel= 100 % DEFAULT
5
6 PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
7 PTP HOME Vel= 100 % DEFAULT
8
9 END

```

Figura 3.57: Programación de la trayectoria PTP

- Con las teclas de dirección ubicarse en la línea de programación que se desea ejecutar, presionar el softkey **Seleccionar línea**.
- Mantener presionado el botón de hombre muerto y el botón de arranque



, ubicado en la parte izquierda del KCP.

Programar un movimiento LIN

- Mover el TCP a la posición a la que se programará como punto de destino.

- Seleccionar la secuencia **Instrucc. > Movimiento > LIN.**
- Declarar los parámetros en el formulario Inline. Velocidad 2 m/s



Figura 3.58: Formulario Inline trayectoria LIN

- Guardar la instrucción con el softkey **Instrucción OK**

```

1 DEF home( )
2  INI
3
4  PTP HOME  Vel= 100 % DEFAULT
5
6  [.IN P2 Vel=2 m/s CPDAT8 Tool[8]:tesis Base[7]:basetesis1
7  PTP HOME  Vel= 100 % DEFAULT
8
9  END

```

Figura 3.59: Programación de la trayectoria LIN

- Con las teclas de dirección ubicarse en la línea de programación que se desea ejecutar, seleccionar con el softkey **Seleccionar línea.**
- Mantener presionado el botón de hombre muerto y el botón de arranque



, ubicado en la parte izquierda del KCP.

Programar un movimiento CIR

- Definir con el softkey “**Touch PI**” el punto inicial de la trayectoria circular y con el sofkey “**Touch PF**” el punto de destino.
- Seleccionar la secuencia **Instrucc. > Movimiento > CIRC.**
- Declarar los parámetros en el formulario Inline.



Figura 3.60: Formulario Inline trayectoria CIR

- Guardar la instrucción con el softkey **Instrucción OK**.

```

1 DEF home( )
2 INI
3
4 PTP HOME Vel= 100 % DEFAULT
5
6 CIRC P3 P4 Vel=2 m/s CPDAT8 Tool[8]:tesis Base[7]:basetesis1
7 PTP HOME Vel= 100 % DEFAULT
8
9 END

```

Figura 3.61: Programación de la trayectoria CIRC

- Con las teclas de dirección ubicarse en la línea de programación que se desea ejecutar, seleccionar con el softkey **Seleccionar línea**.
- Mantener presionado el botón de hombre muerto y el botón de arranque



, ubicado en la parte izquierda del KCP.

PROCEDIMIENTO DE LA ACTIVIDAD 2

- Crear un programa con el nombre `prac_cuadrado`
- Realizar la programación KRL, con el código que se indica a continuación.

`DEF prac_cuadrado()`; nombre del programa, se crea por default con el nombre del módulo.

`INI`; comando de inicialización del programa

`PTP HOME Vel= 100 % DEFAULT`; posición Home aparece por default al crear el programa.

`PTP P1 Vel=50 % PDAT1 Tool[8]:tesis Base[7]:basetesis1`; primer movimiento PTP (en el espacio) desde la posición home hasta la mesa.

`PTP P2 Vel=50 % PDAT2 Tool[8]:tesis Base[7]:basetesis1`; movimiento PTP de ubicación del TCP en p1.

`LIN P3 Vel=0.2 m/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1`; movimiento LIN de p1 a p2. (usar coordenadas WORLD)

LIN P4 Vel=0.2 m/s CPDAT2 Tool[8]:tesis Base[7]:basetesis1;movimiento LIN de p2 a p3. (usar coordenadas WORLD)

LIN P5 Vel=0.2 m/s CPDAT3 Tool[8]:tesis Base[7]:basetesis1 ;movimiento LIN de p3 a p4. (usar coordenadas WORLD)

LIN P6 Vel=0.2 m/s CPDAT4 Tool[8]:tesis Base[7]:basetesis1;movimiento LIN de p4 a p1. (usar coordenadas WORLD)

PTP HOME Vel= 100 % DEFAULT; posición Home aparece por default al crear el programa.




END; Finalización del programa.

```

1 DEF prac_cuadrado( )
2  INI
3
4  PTP HOME  Vel= 100 % DEFAULT
5
6  PTP P1 Vel=50 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
7  PTP P2 Vel=50 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
8  LIN P3 Vel=0.2 n/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1
9  LIN P4 Vel=0.2 n/s CPDAT2 Tool[8]:tesis Base[7]:basetesis1
10 LIN P5 Vel=0.2 n/s CPDAT3 Tool[8]:tesis Base[7]:basetesis1
11 LIN P6 Vel=0.2 n/s CPDAT4 Tool[8]:tesis Base[7]:basetesis1
12 PTP HOME  Vel= 100 % DEFAULT
13
14  END

```

Figura 3.62: Programación del cuadrado

- Ejecutar en modo paso a paso .
- Seleccionar la línea desde donde se va a correr el programa, presionar el botón hombre muerto y .
- Con velocidad de programa de .

PROCEDIMIENTO DE LA ACTIVIDAD 3

- Crear un programa con el nombre prac_circulo
- Realizar la programación KRL, con el código que se indica a continuación.

DEF prac_circulo() ; nombre del programa

INI ; comando de inicialización del programa

PTP HOME Vel= 100 % DEFAULT; posición Home aparece por default

PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1; primer movimiento PTP (en el espacio) desde la posición home hasta la mesa.

PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1; movimiento PTP de ubicación del TCP en p1.

CIRC P3 P17 Vel=2 m/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1; movimiento CIRC de p1 a p3, con centro en p4 (usar coordenadas WORLD).

CIRC P4 P5 Vel=2 m/s CPDAT6 Tool[8]:tesis Base[7]:basetesis1; movimiento CIRC de p3 a p1, con centro en p2 (usar coordenadas WORLD).

PTP P7 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1; movimiento PTP de ubicación sobre en p3.

PTP HOME Vel=100 % DEFAULT; posición Home aparece por default al crear el programa



END


```

1  DEF prac_circulo( )
2  INI
3
4  PTP HOME Vel= 100 % DEFAULT
5
6  PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
7  PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
8  CIRC P3 P4 Vel=2 m/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1
9  CIRC P5 P6 Vel=2 m/s CPDAT2 Tool[8]:tesis Base[7]:basetesis1
10 PTP P7 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
11
12 PTP HOME Vel=100 % DEFAULT
13  END

```

Figura 3.63: Programación del círculo

- Ejecutar en modo paso a paso 
- Seleccionar la línea desde donde se va a correr el programa, presionar el hombre muerto y 

- Con velocidad de programa de 

E. RESULTADOS

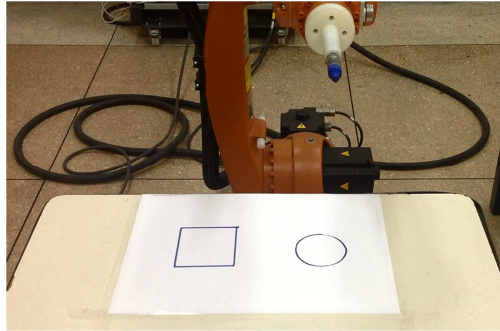


Figura 3.64: Resultado de las trayectorias LIN Y CIRC

F. TRABAJO COMPLEMENTARIO

Utilizar trayectorias punto a punto, circulares y lineales para obtener la figura que se indica a continuación.

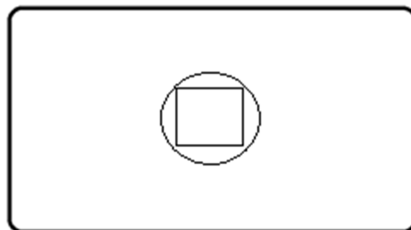


Figura 3.65: Figura determinada

3.5 SIMULACIÓN DE MOVIMIENTOS DEL ROBOT EN EL ENTORNO VIRTUAL KUKASIM.PRO

A. OBJETIVOS

Objetivo General:

- Diseñar y simular movimientos que permitan realizar un proceso en KUKA Sim.Pro

Objetivos Específicos:

- Utilizar las herramientas y componentes que ofrece el entorno virtual KUKASim.Pro
- Mover una pieza con el manipulador robótico de una mesa y ubicarlo en una banda transportadora.

B. MARCO TEÓRICO

KUKA SIM PRO

KUKA.Sim Pro es un programa específico para robots KUKA, permite elaborar diseños en tres dimensiones, cuenta con librerías que contienen una variedad de componentes como:

- Robots de diferentes modelos.
- Mesas de trabajo
- Gripper
- Elementos pasivos, no pueden ser simulados solo forman parte del entorno de trabajo.

Entorno de trabajo es el área donde están todos los componentes, y se desplaza el robot.

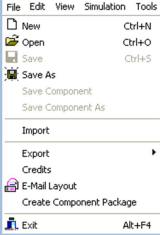
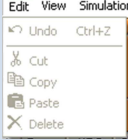

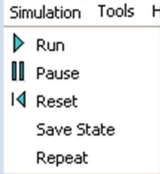
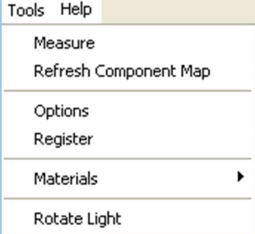
La **zona de trabajo** está determinada por los límites de los ejes y el desplazamiento de las articulaciones.

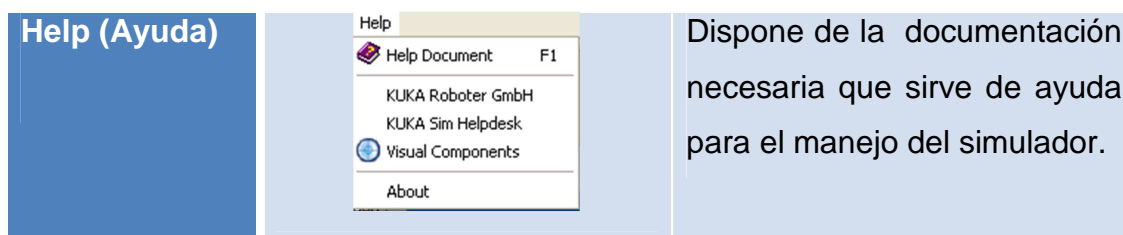
Volumen de trabajo es el espacio donde se desplaza el gripper.

BARRA DE MENÚ

La barra de menú dispone de las siguientes opciones.






Tabla 3.7: Funciones de navegación de KUKA Sim.Pro

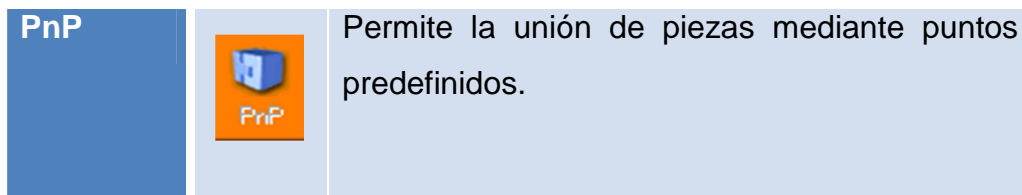
OPCIONES	ICONO	DETALLE
File (Archivo)		Contiene opciones generales como abrir, guardar, nuevo, etc.
Edit (Editar)		Contiene opciones como regresar, cortar, pegar o eliminar.
View (Ver)		Permite ocultar o mostrar diferentes barras o paneles en la interfaz de usuario.
Simulation (Simulación)		Permite poner en marcha la simulación: iniciar, parar, retroceder, etc.
Tools (Herramientas)		Muestra herramientas para agregar al sistema o interfaz de usuario.



BARRA DE NAVEGACIÓN

Tabla 3.8: Funciones de navegación de KUKA Sim.Pro

FUNCIÓN	ICONO	DETALLE
Orbit		Permite girar en tres dimensiones el entorno de trabajo.
Pan		Permite mover el entorno de trabajo en cualquier dirección.
Zoom		Permite ampliar la zona de trabajo para una mejor visualización.
Fill		Permite encuadrar dentro del entorno de trabajo todos los elementos que existen en ella sin necesidad del zoom.
Select		Permite seleccionar cualquier componente del entorno de trabajo.
Trans		Permite trasladar los elementos previamente seleccionados en un sistema de coordenadas.
Rot		Permite rotar los elementos previamente seleccionados con respecto a un sistema de coordenadas.



HERRAMIENTAS PARA EL CONTROL DE SIMULACIÓN

Dispone de tres teclas; **repeat** permite repetir el ciclo de simulación, **play** ejecuta la simulación y **reset** reinicia el programa e inicializa las variables.

El **sim speed** controla la velocidad de simulación, si se desplaza el cursor hacia la izquierda se reduce la velocidad, si se desplaza hacia la derecha se incrementa.



Figura 3.66: Control de simulación

En la parte inferior al campo de trabajo se encuentra el panel de mensajes que provee información adicional acerca de las operaciones que se están ejecutando.

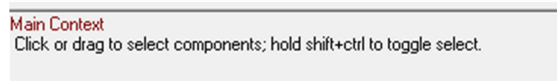


Figura 3.67: Panel de mensajes

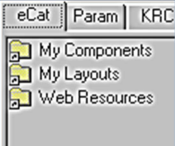
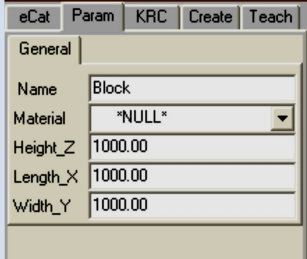

En la parte inferior también se encuentra la barra de estado que contiene características del área de trabajo.

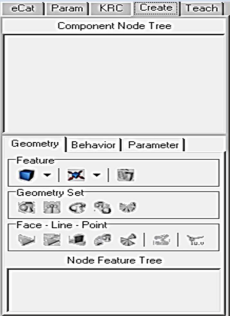
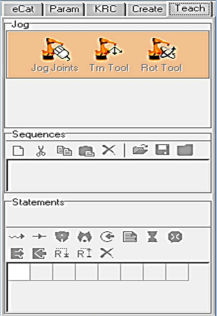


Figura 3.68: Barra de estado

CAMPOS Y PESTAÑAS

Tabla 3.9: Campos de KUKA Sim.Pro

FUNCIÓN	ICONO	DETALLE
eCAt		Muestra los componentes y las librerías.
Param		Muestra las características del componente actualmente seleccionado y permite modificar parámetros como altura, espesor, ancho.
KRC		KUKA Robot Controller (KRC) se usa para verificar el movimiento. KRC realiza la comunicación con KUKA Office.

Create		<p>Dispone las funciones para crear componentes. Posee un menú desplegable en el que se encuentran las herramientas para su creación.</p>
Teach		<p>Provee las funciones para mover y programar las trayectorias del robot.</p>

OPCIONES DE LA PESTAÑA TEACH



Figura 3.69: Opciones del Teach







- **Jog Joints:** Permite mover cada articulación del robot independientemente.
- **Tm Tool:** Permite mover el gripper del robot dentro del volumen de trabajo con coordenadas cartesianas.
- **Rot Tool:** Permite mover el gripper del robot dentro del volumen de trabajo con coordenadas cilíndricas.

Botones de programación



Figura 3.70: Herramientas de programación

Tabla 3.10: Iconos de programación de KUKA Sim.Pro

ICONO	DESCRIPCIÓN
	Graba posición.
	Graba posición, con trayectorias lineales.
	Abre y cierra el gripper.
	Llama a subrutina.
	Delay, tiempo de espera.
	Set/ reset banderas.
	Activan las rutinas

C. PROCEDIMIENTO

Realiza una simulación de un proceso como muestra la figura 3.71.

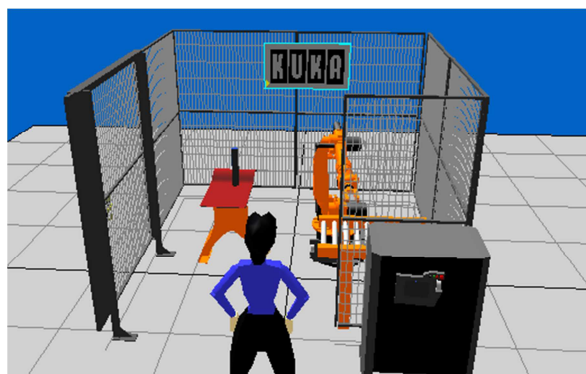


Figura 3.71: Proceso en KUKA Sim Pro

D. PROCEDIMIENTO DETALLADO

Ejecutar la aplicación KUKA SimPro1.1.



Figura 3.72: Aplicación KUKA SimPro1.1

Ejecutada la aplicación se presenta una pantalla como muestra la Figura 3.71. Esta es la interfaz donde trabajará el usuario.

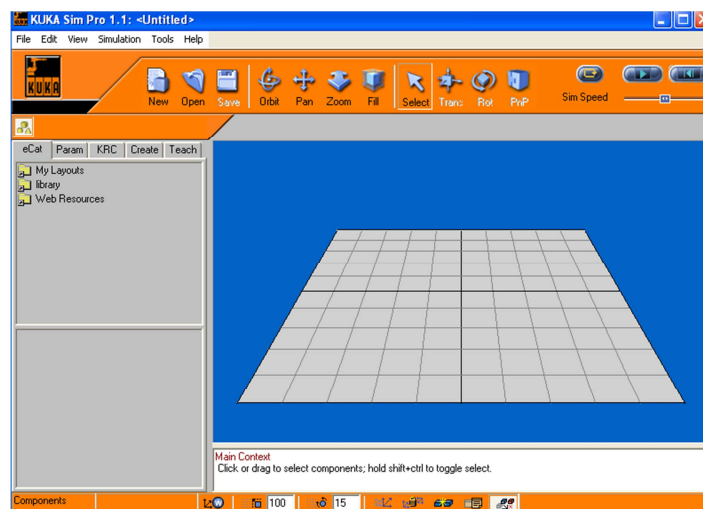


Figura 3.73: Interfaz de usuario

Agregar el modelo del brazo Robótico KR 16. Seleccionar **eCat> library > Kuka Roboter> Robots> Low Payloads >** Arrastrar hacia nuestra área de trabajo.

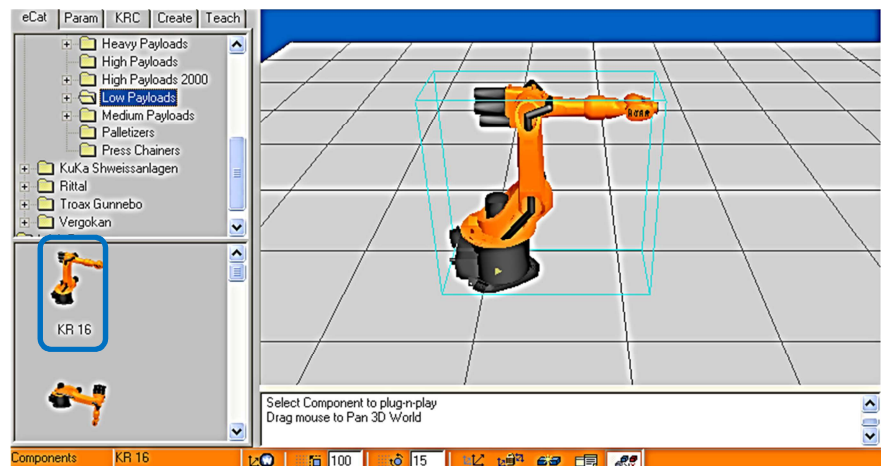


Figura 3.74: Manipulador KR16

Agregar un Gripper tipo pinza. **Library > General> Grippers> Generic Two Finger**. Arrastrar al área de trabajo.

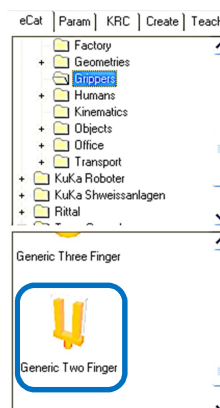


Figura 3.75: Gripper tipo pinza

Asociar el Gripper a la articulación. Colocar el gripper a la altura de la articulación del robot. Seleccionar en la **Barra de Navegación > PnP > Interactive Plug and Play**. Al dar un click sobre el gripper, aparecerá una línea que indica a que articulación se unirá.



Figura 3.76: Opción Interactive Plug and Play

Unir el gripper al robot. Seleccionar en la **Barra de Navegación** > **PnP** > **Set Parent Node**, dar click en la articulación, aparecerá una flecha, dar click en PNP, arrastrar el gripper a la articulación del robot. Finalmente dar doble click sobre el robot.



Figura 3.77: Opción Set parent node

Agregar una mesa. Seleccionar **eCat> library > Kuka Roboter > Cells > Training > Table**. Arrastrar al entorno de trabajo.

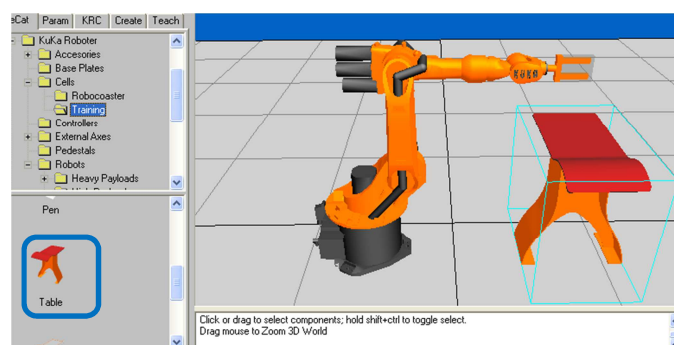


Figura 3.78: Mesa

Agregar una banda transportadora. Seleccionar la carpeta **eCat> library > General >Transport > Conveyors**. Arrastrar al entorno de trabajo.

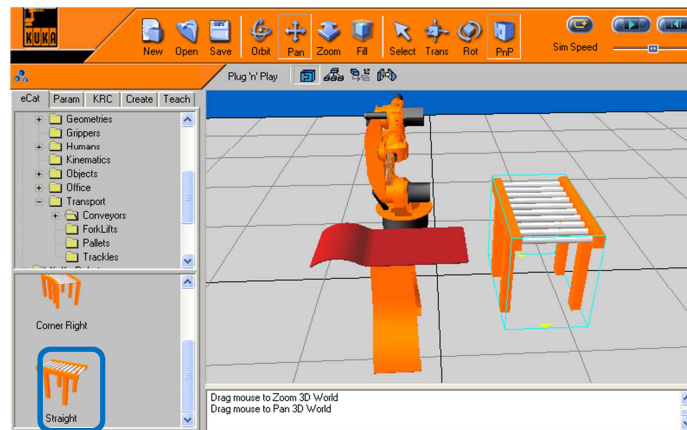


Figura 3.79: Banda transportadora

Agregar un objeto. Seleccionar la carpeta **eCat> library > General> Geometries > Primitives > Cylinder**, colocar el cilindro sobre la mesa. Modificar las propiedades del cilindro en la pestaña **Param**, seleccionar el tipo de material y las dimensiones como indica la figura 3.80.

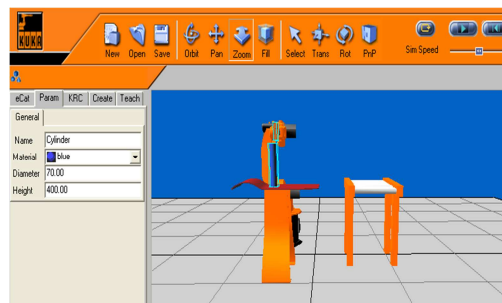


Figura 3.80: Cilindro

Programar el robot. Seleccionar la pestaña **Teach**, aparecen dos marcas de color gris; una en la base y otra en el gripper. Escoger la marca de la Base.

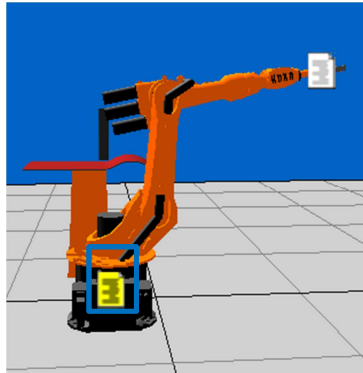


Figura 3.81: Hoja de la Base

Ubicar al robot en la posición Home. Seleccionar la pestaña **Teach> Jog Joints**, escoger la articulación a mover y desplazar a la posición indicada en la figura 3.82.

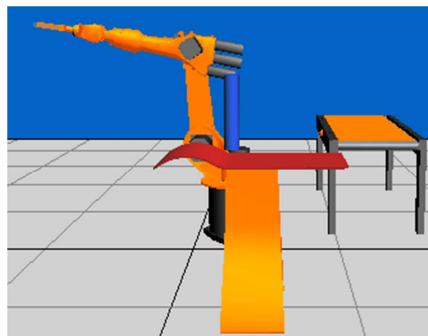



Figura 3.82: Posición Home

Guardar la posición. Seleccionar la opción **Add Point**  y automáticamente se configura la posición P1 (home).

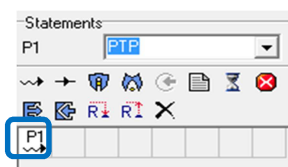


Figura 3.83: Posición Home P1

Mover el robot sobre del cilindro y guardar la posición.

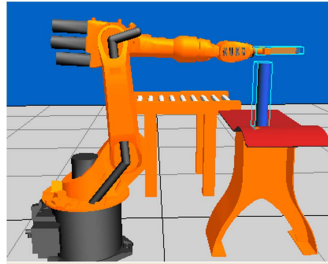





Figura 3.84: Posición P2

Abrir el gripper con **Grasp an Item with a Gripper**  , tomar el cilindro con **Add Point**  guardar la posición y cerrar el gripper con **Release grasped ítem**  .

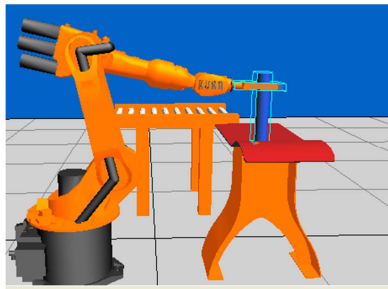


Figura 3.85: Posición P3- tomar el cilindro

Levantar el cilindro de la mesa y guardar la posición.

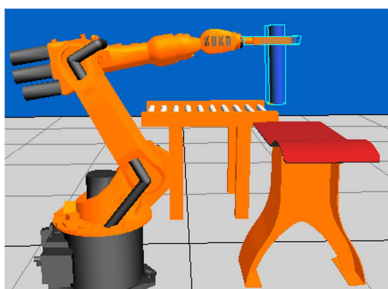


Figura 3.86: Posición P4- banda transportadora

Llevar el objeto a la banda transportadora y guardar la posición.

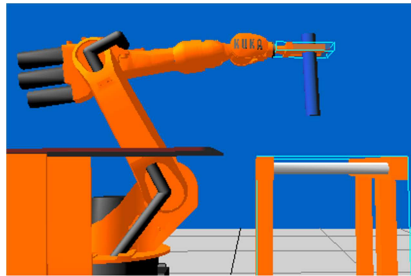



Figura 3.87: Posición P5

Ubicar el cilindro sobre la banda transportadora, dejar el cilindro  y levantar el brazo.

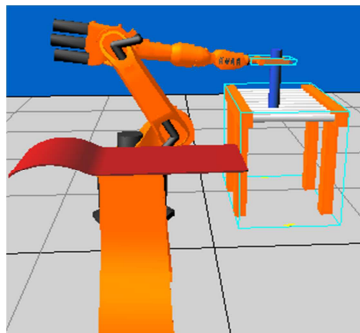


Figura 3.88: Posición P6-P7

Regresar a la posición home y guardar la posición.

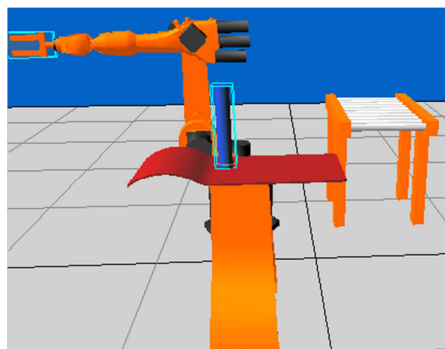


Figura 3.89: Posición P8- Regresar a posición home

La Figura 3.90 muestra la programación de los movimientos para el proceso.



Figura 3.90: Programación del proceso

Añadir una célula de trabajo, insertar los siguientes objetos:

Operador (General, Human), controlador (Kuka Roboter, controllers, KRC2), KCP (Kuka Roboter, Controller, KCP) y puertas (Troax Gunnebo, Grills, Splide Door Left)

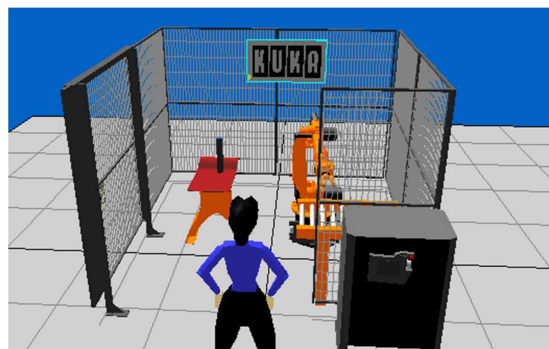


Figura 3.91: Célula del operador

Simular el proceso usar los botones **Play** y **Reset**. Realizar el proceso una y otra vez con opción **Loop Simulation** y variar la velocidad.

E. TRABAJO COMPLEMENTARIO

1. Realizar un proceso usando el manipulador robótico modelo KR-5 ARC y un gripper de tipo ventosa, para transportar vidrio.

3.6 PROGRAMACIÓN EN MODO EXPERTO USANDO ESTRUCTURAS EN KRL

A. OBJETIVOS

Objetivo General:

- Establecer posiciones y velocidades a los ejes del robot en modo experto.

Objetivos Específicos:

- Asignar valores de coordenadas y ángulos a los ejes para realizar movimientos con el robot.

B. MARCO TEORICO

MODO EXPERTO

El modo experto permite realizar modificaciones, abrir carpetas propias del sistema, usar instrucciones, colocar comentarios. En este modo se puede visualizar el contenido de los archivos .DAT.

ESTRUCTURAS

STRUC: permite agrupar diferentes tipos de datos definidos previamente o directamente.

POS: Identifica los tres valores de posición (X,Y y Z) y tres orientaciones (A, B y C)

STRUC POS (X, Y, Z, A, B, C)

VELOCIDAD

El robot permite una velocidad máxima de 250 mm/s, para programar se coloca de 0.....100 %, Este valor sólo hace referencia al segmento al cual pertenece, no

actúa sobre los subprogramas.

\$VEL_AXIS [número de eje]

ACELERACIÓN

El valor máximo depende del tipo de robot. Al programar se coloca de 0.....100%. El eje con el recorrido más largo, es llamado eje directriz, se desplaza con el valor de aceleración y velocidad programado. Todos los demás ejes se mueven solamente con las aceleraciones y velocidades necesarias para alcanzar el punto final del movimiento.

\$ACC_AXIS [número de eje]

La posición home tendrá los siguientes valores en sus ejes.

PTP {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}

Para producir un ángulo de rotación de una articulación usar: **PTP {"eje" "grados"}**

Ejemplo:

PTP {A3 45} ; el eje tres se mueve 45°

Si se requiere rotar diferentes articulaciones, manteniendo las otras en el valor actual, usar:

Ejemplo:

PTP_REL {A1 80, A4 35}

C. ACTIVIDAD

Realizar un programa que realice las operaciones definidas en el diagrama de flujo y dibujar un triángulo. Ubicando los tres puntos del triángulo sobre la base. Según la figura 3.92.

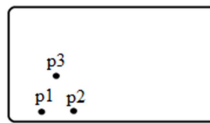


Figura 3.92: Puntos de referencia para un triángulo

1. Realizar las condiciones que indica el diagrama de flujo.

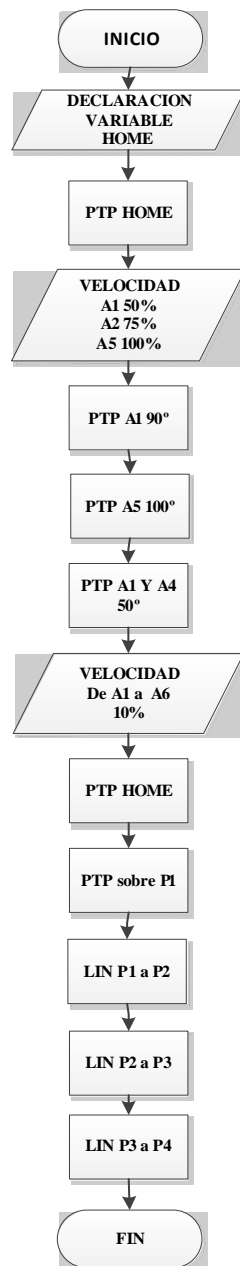


Figura 3.93: Diagrama de flujo

D. PROCEDIMIENTO

1. Crear un nuevo programa tipo módulo llamado **prac_estructuras()**
2. Seleccionar la base y herramienta.
3. Cambiar el grupo a experto.
 - Seleccionando en la barra de menú **Configurar > Grupo de usuario**.
 - Seleccionar en la parte inferior izquierda **conectar**.

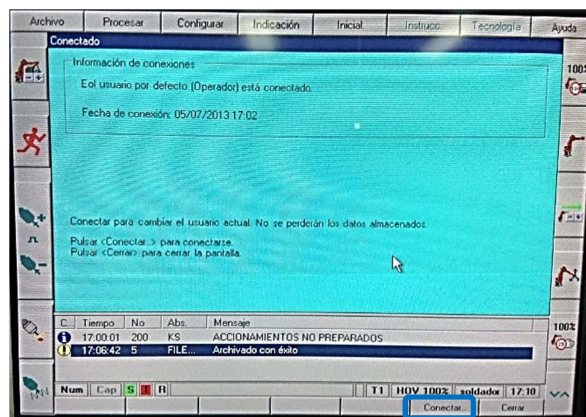


Figura 3.94: Conexión a grupo experto

- Ingresar la clave: **“kuka”** y seleccionar el softkey **conectar**

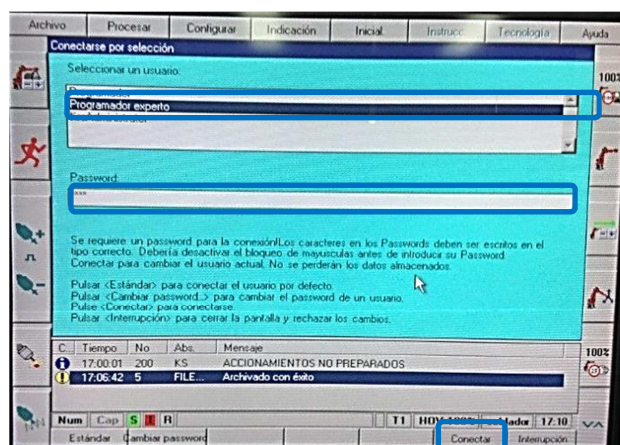


Figura 3.95: Ingreso de clave

4. Programar en lenguaje KRL

DEF prac_estructuras()

decl **AXIS** HOME ; declaración de la posición Home

INI; inicialización del programa

BAS(#INITMOV); movimientos básicos.

\$base=\$World ; tipo de coordenadas

\$tool=\$nullframe ; herramienta externa

HOME={**AXIS**: A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0} ; coordenadas para posición home.

ptp HOME ; movimiento punto a punto para posición home.

\$vel_axis[1]=50 ; velocidad 50% del eje 1

\$vel_axis[4]=75; velocidad 75% del eje 4

\$vel_axis[5]=100; velocidad 100% del eje 5

ptp {A1 90} ; movimiento punto a punto de articulación 1 en un ángulo de 90°

ptp {A5 100}; movimiento punto a punto del articulación 5 en un ángulo de 100°

ptp_rel{A4 50,A1 50}; movimiento punto a punto del eje 4 y 1 en un ángulo de 50°

\$vel_axis[1]=30 ; velocidad 30% del eje 1

\$vel_axis[2]= 30; velocidad 30% del eje 2

\$vel_axis[3]=30; velocidad 30% del eje 3

\$vel_axis[4]=30; velocidad 30% del eje 4

\$vel_axis[5]=30; velocidad 30% del eje 5

\$vel_axis[6]=30; velocidad 30% del eje 6

\$acc_axis[1]=50; aceleración 50% del eje 1

\$acc_axis[2]=50; aceleración 50% del eje 2

\$acc_axis[3]=50; aceleración 50% del eje 3

\$acc_axis[4]=50; aceleración 50% del eje 4

\$acc_axis[5]=50; aceleración 50% del eje 5

\$acc_axis[6]=50; aceleración 50% del eje 6

ptp HOME

ptp {**pos**: x 803.30,y 0.04, z 900,**a** 164.5,**b** 1.61,**c** 179.56,**s** 2,**t** 35} ;posición punto a punto

F. TRABAJO COMPLEMENTARIO

1. Realizar movimientos con el tipo de coordenadas tool y variar las velocidades.

3.7 TRAYECTORIAS CON SENTENCIAS CONDICIONALES

A. OBJETIVOS

Objetivo General:

- Realizar trayectorias y formar figuras con un número de repeticiones establecidas.

Objetivos Específicos:

- Trabajar en modo experto.
- Utilizar sentencias condicionales en la programación de los movimientos.

B. MARCO TEORICO

ESTRUCTURA DE UN PROGRAMA

Un programa o subprograma consta de tres partes básicas:

Sección de declaración: En esta sección se asigna el tipo de variables que se usa en el programa y pueden ser llamada desde cualquier parte del programa según convenga.

Sección de inicialización: Se encuentra el comando INI que da inicio al programa.

Sección de instrucciones: Esta sección cuenta con comandos, funciones o sentencias de control.

Ventajas:

Un subprograma o una función puede llamar a otros subprogramas.

- Todos los subprogramas se declaran exactamente igual que los programas principales, cuentan con la misma estructura.

- Son utilizados para reducir la programación cuando existen tareas que se repiten frecuentemente.

-

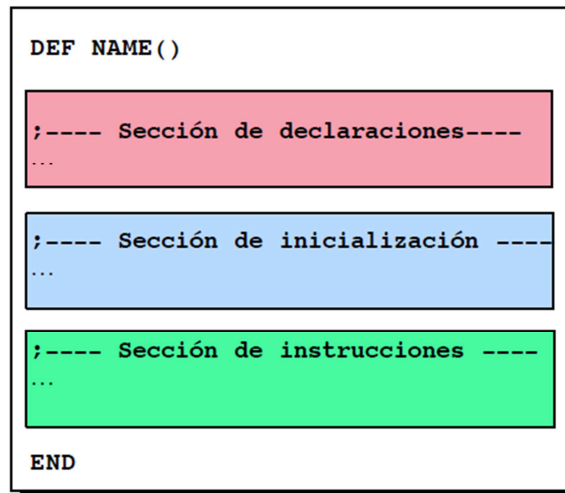


Figura 3.98: Estructura básica de un programa (KUKA, 2011)

FUNCIONES

LOOP - ENDLOOP

Permite repetir indefinidamente un bucle de un bloque de instrucciones, no se necesita inicialización. Se utiliza de la siguiente manera:

LOOP

Instrucciones

ENDLOOP

La ejecución repetida del bloque solo se interrumpe con EXIT.

LOOP

Instrucciones

IF condicion

EXIT

ENDLOOP

IF – ELSE- THEN – ENDIF

Se ejecuta en función de una condición. En caso de no cumplirla el programa continúa con las siguientes líneas de programación. Se pueden intercalar varias instrucciones IF entre sí. Se utiliza de la siguiente manera:

IF

Instrucciones

ENDIF

EXIT

Con la instrucción EXIT se puede terminar cualquier bucle instantáneamente. Llamando a EXIT dentro de un bloque.

SWITCH - CASE – ENDSWITCH

La instrucción SWITCH es para selección. Si el valor concuerda con una identificación de bloque, se procesa la acción en el programa o subprograma correspondiente, si no coincide ninguna identificación con el criterio de selección, se ejecutará el bloque DEFAULT. Si no hay ningún bloque DEFAULT, continua la ejecución del programa tras ENDSWITCH. Se utiliza de la siguiente manera:

SWITCH condición

case 1

subprograma

case 2

subprogram

.....

ENDSWITCH

FOR

Es un bucle de conteo en el cual se necesita declarar un dato entero que será el contador y el valor hasta el cual llegará.

Se utiliza de la siguiente manera:

DECL INT contador

FOR contador=1 **TO** número de veces que quiere que la instrucción se realice

Instrucciones

ENDFOR

TIPOS DE VARIABLES

INT

Los datos tipo Integer constituyen una parte de la cantidad total de números enteros. Representan un conjunto de enteros de 32 bits. El número 0 es considerado parte de los números positivos. Se usa para declarar variables.

REAL

Es una aproximación de un número entero. Este tipo de dato indica la precisión con que se pueden representar los números con decimales. Esta característica también está directamente relacionada con la cantidad de memoria disponible para almacenar un valor real. Se usa para identificar las coordenadas.

BOOL Y SIGNAL

Es un tipo de datos lógicos, permite usar variables que disponen sólo de dos posibles valores, cierto o falso.

Las variables tipo signal son usadas en señales de entrada y salida.

CHAR

El tipo de dato char se usa para representar caracteres, este es representado internamente por un entero.

OPERADORES DE COMPARACIÓN

Con los operadores de comparación se pueden formar expresiones lógicas. Por tanto, el resultado de una comparación es siempre un dato de tipo BOOL, ya que una comparación sólo puede ser verdadera (TRUE) o falsa (FALSE).

Tabla 3.11: Descripción de los operadores de comparación

OPERADOR	DESCRIPCIÓN
==	Igual Para tipos de datos INT, REAL, CHAR, BOOL
<>	desigual Para tipos de datos INT, REAL, CHAR, BOOL
>	mayor que Para tipos de datos INT, REAL, CHAR
<	menor que Para tipos de datos INT, REAL, CHAR
>=	mayor o igual que Para tipos de datos INT, REAL, CHAR
<=	menor o igual que Para tipos de datos INT, REAL, CHAR

C. ACTIVIDADES

1. Realizar las siguientes figuras, un cuadrado de 12cm y semicircunferencias de radio 6cm como indica la figura 3.99 en base al diagrama de flujo.

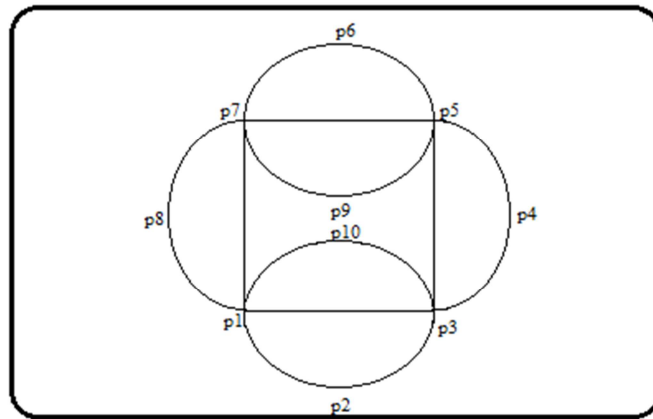


Figura 3.99: Figuras a realizar

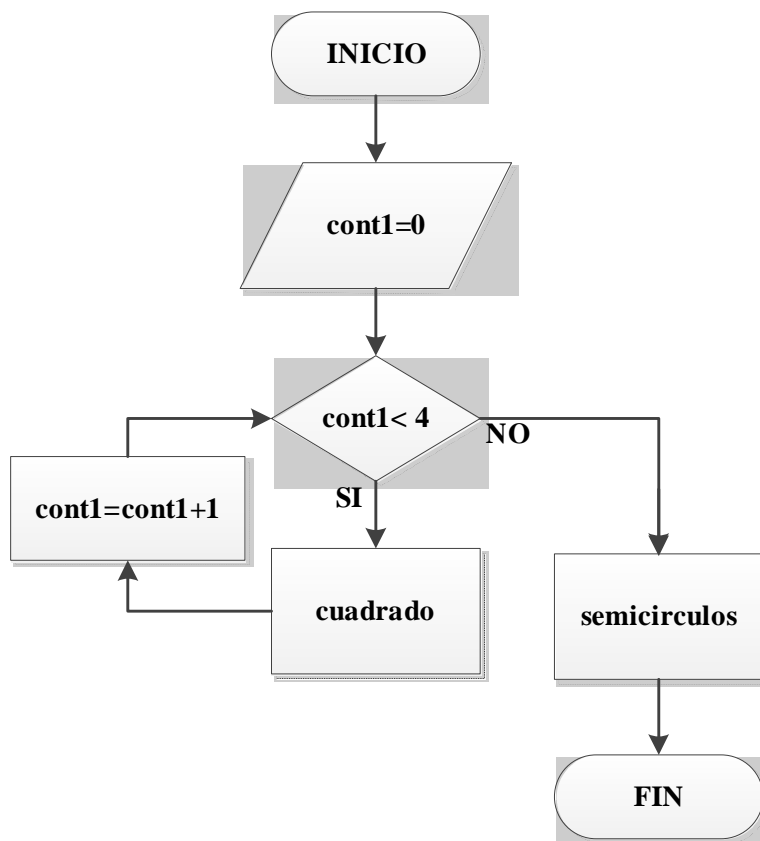


Figura 3.100: Diagrama de flujo

D. PROCEDIMIENTO

1. Trabajar en modo experto.
2. Programar en el lenguaje KRL. Utilizar los formularios de trayectorias para realizar los movimientos.

DEF prac_cuad_semic(); Nombre del programa.

decl int cont1; Declarar la variable cont1 como tipo entero.

INI ; comando de inicialización del programa.

cont1=0; inicialización de la variable.

PTP HOME Vel= 100 % DEFAULT DEFAULT ; posición Home

for cont1=1 to 3; repetición del lazo, de tres veces.

PTP P11 CONT Vel=50 % PDAT11 Tool[8]:tesis Base[7]:basetesis1 ; movimiento al primer punto del cuadrado.

LIN P1 Vel=0.5 m/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1;movimiento LIN de p1 a p2. (usar coordenadas WORLD) .

LIN P2 Vel=0.5 m/s CPDAT2 Tool[8]:tesis Base[7]:basetesis1;movimiento LIN de p2 a p3. (usar coordenadas WORLD) .

LIN P3 Vel=0.5 m/s CPDAT3 Tool[8]:tesis Base[7]:basetesis1;movimiento LIN de p3 a p4. (usar coordenadas WORLD) .

LIN P4 Vel=0.5 m/s CPDAT4 Tool[8]:tesis Base[7]:basetesis1 ;movimiento LIN de p4 a p1. (usar coordenadas WORLD)

endfor; Cerrar el lazo

CIRC P6 P7 CONT Vel=0.5 m/s CPDAT6 Tool[8]:tesis Base[7]:basetesis1; movimiento circular de p7 a p5, con centro en p6.

CIRC P12 P13 CONT Vel=0.5 m/s CPDAT10 Tool[8]:tesis Base[7]:basetesis1; movimiento circular de p5 a p3, con centro en p4.

CIRC P15 P17 CONT Vel=0.5 m/s CPDAT11 Tool[8]:tesis Base[7]:basetesis1; movimiento circular de p3 a p1, con centro en p2.

CIRC P18 P19 CONT Vel=0.5 m/s CPDAT12 Tool[8]:tesis Base[7]:basetesis1; movimiento circular de p1 a p7, con centro en p8.

CIRC P20 P21 CONT Vel=0.5 m/s CPDAT13 Tool[8]:tesis Base[7]:basetesis1; movimiento circular de p7 a p3, con centro en p10.

LIN P24 CONT Vel=0.5 m/s CPDAT15 Tool[8]:tesis Base[7]:basetesis1;
movimiento lineal de p5 a p3.

CIRC P22 P23 CONT Vel=0.5 m/s CPDAT14 Tool[8]:tesis Base[7]:basetesis1;
movimiento circular de p3 a p1, con centro en p10.

PTP HOME Vel= 100 % DEFAULT

END

E. RESULTADO

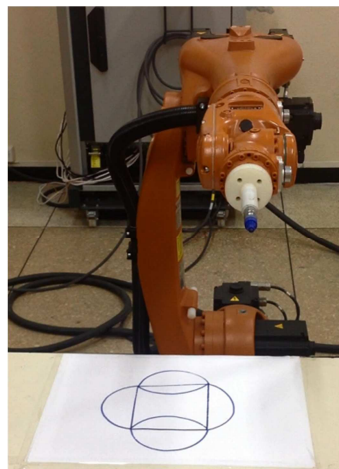


Figura 3.101: Resultado de la figura determinada

F. TRABAJO COMPLEMENTARIO

1. Realizar las trayectorias mostradas en la figura.

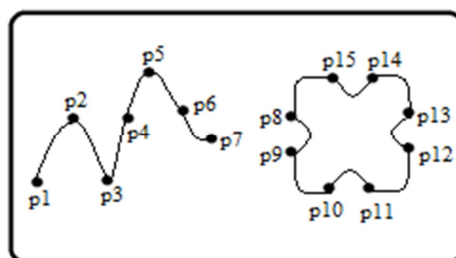



Figura 3.102: Figuras determinadas

2. Tomar como referencia los puntos de la figura 3.102
3. Crear un nuevo programa llamado **mov_spline**.
4. Cambiar de modo de trabajo a modo experto.
5. Seleccionar la secuencia de menú **Instrucc. > Movimiento > Bloque SPLINE**.
6. Declarar los parámetros en el formulario Inline.



SPLINE S1 Vel=2 m/s CPDAT1

Figura 3.103: Formulario Inline de trayectorias SPLINE

7. Guardar la instrucción con el softkey **Instrucción OK**.
8. Pulsar la tecla de funciones **Abrir/Cerrar Fold**. Ahora se pueden insertar líneas en el bloque spline.

3.8 SUBPROGRAMAS Y FUNCIONES

A. OBJETIVOS

Objetivo General:

- Ejecutar subrutinas y funciones para realizar un proceso que permita ubicar objetos en sitios específicos.

Objetivos Específicos:

- Utilizar instrucciones de espera en subrutinas.
- Realizar un programa principal que permita llamar a subrutinas.

B. MARCO TEORICO

SUBPROGRAMAS Y FUNCIONES

Permiten reducir la programación al evitar la escritura repetida de un programa. Como consecuencia se reduce la longitud del programa.

Un efecto de los subprogramas y de las funciones es la posibilidad de volver a utilizar en otros programas los algoritmos que ya han sido registrados. Se puede procesar tareas parciales.

COMENTARIOS

Los comentarios son importantes, con ellos se puede clarificar la estructura de un programa y hacer que otro usuario puedan comprenderlo. Los comentarios no influyen en la velocidad de procesamiento del programa, pueden colocarse en cualquier parte de un programa. Siempre comienzan con un punto y coma “;”

Instrucciones de espera

La instrucción **WAIT** espera hasta que se produzca un evento determinado. Con la instrucción **WAIT FOR** (condición), se puede detener el procesamiento del programa hasta que se produzca el evento específico.

La instrucción **WAIT SEC** sirve para programar los tiempos de espera en segundos: **WAIT SEC** (Tiempo). El tiempo es una expresión aritmética REAL, con la que se indica la cantidad de segundos dura la interrupción del programa. Si el valor es negativo, no se produce la espera.

C. ACTIVIDAD

1. Realizar un proceso que permita tomar un objeto de la mesa lo lleve a la banda transportadora, regrese a la mesa tome otro objeto y lo lleve a una repisa de almacenamiento.

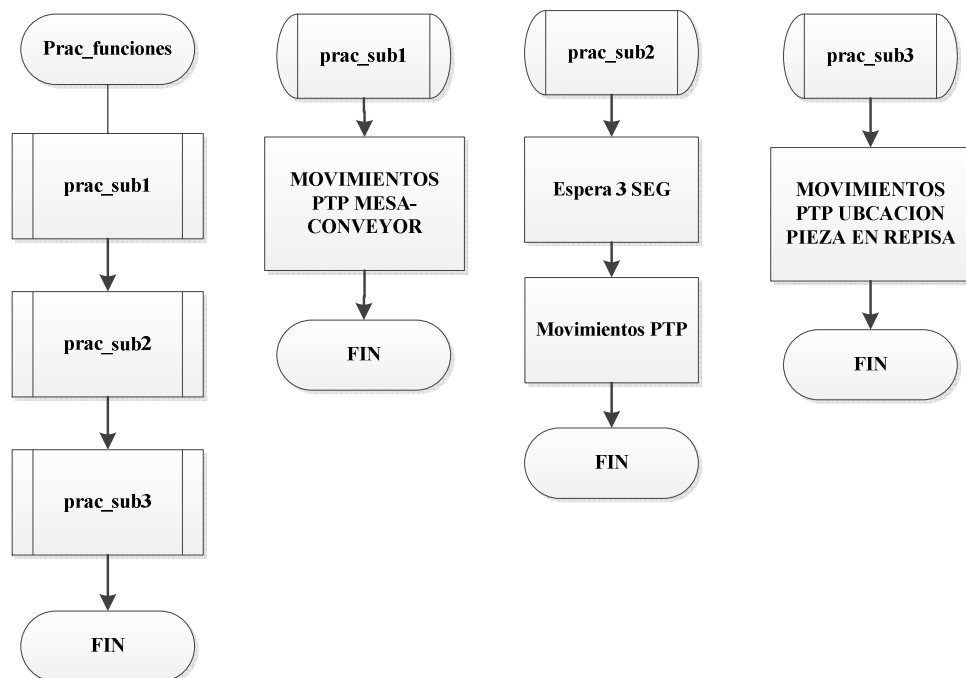


Figura 3.104: Diagrama de flujo del proceso

D. PROCEDIMIENTO

1. Crear un programa con el nombre **prac_funciones**, este será el programa principal, en el que se encontrarán las subrutinas y repetirá el proceso 5 veces.

```
DEF prac_funciones()  
decl int cont INI  
cont=0  
for cont=1 to 5  
prac_sub1()  
prac_sub2()  
prac_sub3()  
endfor  
END
```

2. Crear un programa con el nombre **prac_sub1** que cumpla los movimientos de las trayectorias de la mesa de trabajo a la banda transportadora y viceversa.

```
DEF prac_sub1()  
INI ; comando de inicialización del programa.  
PTP P1 Vel=70 % PDAT2 Tool[8]:tesis Base[7]:basetesis1; ubicación del TCP  
sobre el objeto  
que se encuentra en la mesa.  
PTP P2 Vel=70 % PDAT3 Tool[8]:tesis Base[7]:basetesis1;toma del objeto de la  
mesa.  
LIN P3 Vel=1 m/s CPDAT5 Tool[8]:tesis Base[7]:basetesis1; movimiento de la  
mesa a la banda transportadora.  
LIN P4 Vel=1 m/s CPDAT6 Tool[8]:tesis Base[7]:basetesis1 ;movimiento de la  
banda transportadora a la mesa.  
return; comando para regresar al programa principal "prac_funciones( )"
```

END

```

1 DEF prac_sub1( )
2   INI
3
4   PTP P1 Vel=70 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
5   PTP P2 Vel=70 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
6   LIN P3 Vel=1 m/s CPDAT5 Tool[8]:tesis Base[7]:basetesis1
7   LIN P4 Vel=1 m/s CPDAT6 Tool[8]:tesis Base[7]:basetesis1
8
9
10  return
11  end

```

Figura 3.105: Programa desplazamiento mesa-banda transportadora

3. Crear un programa con el nombre **prac_sub2()**, que espere dos segundos y tome un objeto.

DEF prac_sub2()

INI ; comando de inicialización del programa.

WAIT Time=5 sec ; sentencia de espera, tiempo en segundos.

PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1; deja el elemento en la mesa

PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1; toma el elemento de la mesa

return; comando para regresar al programa principal "prac_funciones()"

END;

```

3
4   WAIT Time=5 sec
5   PTP P1 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
6   PTP P2 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1
7   prac_sub2( )
8
9
10  END

```

Figura 3.106: Programa de espera y toma de objeto

4. Crear otro programa que lleve el nombre **prac_sub3**.

DEF prac_sub3() ; definir el programa

INI ; comando de inicialización del programa.

PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1; movimiento de toma del objeto de la mesa.

PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1; movimiento hacia la repisa de almacenamiento

PTP P3 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1; movimiento ubicación en repisa.

PTP P4 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1; movimiento ubicación en repisa.

PTP P5 Vel=100 % PDAT5 Tool[8]:tesis Base[7]:basetesis1; movimiento ubicación en repisa.

PTP P6 Vel=100 % PDAT6 Tool[8]:tesis Base[7]:basetesis1; movimiento ubicación en repisa.

PTP P7 Vel=100 % PDAT7 Tool[8]:tesis Base[7]:basetesis1; movimiento regreso a la mesa.

PTP P8 Vel=100 % PDAT8 Tool[8]:tesis Base[7]:basetesis1; movimiento regreso a la mesa.

return; comando para regresar al programa principal "prac_funciones()"

END

```
1 DEF prac_sub3( )
2 INI
3
4
5 PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
6 PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
7 PTP P3 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
8 PTP P4 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1
9 PTP P5 Vel=100 % PDAT5 Tool[8]:tesis Base[7]:basetesis1
10 PTP P6 Vel=100 % PDAT6 Tool[8]:tesis Base[7]:basetesis1
11 PTP P7 Vel=100 % PDAT7 Tool[8]:tesis Base[7]:basetesis1
12 PTP P8 Vel=100 % PDAT8 Tool[8]:tesis Base[7]:basetesis1
13
14 return
15
16
17 END
```

Figura 3.107: Programa de ubicación de objeto en repisa

3.9 CONFIGURACIÓN DEL MÓDULO DE ENTRADAS Y SALIDAS DIGITALES

A. OBJETIVOS

Objetivo General:

- Configurar y verificar la activación de las entradas y salidas digitales del módulo (WAGO).

Objetivos Específicos:

- Revisar las conexiones y características generales módulo de entradas y salidas.
- Programar en modo experto acciones que permitan validar las entradas y salidas.

B. MARCO TEORICO

MÓDULO DE ENTRADAS Y SALIDAS WAGO-I/O-SYSTEM 750

Es un sistema de bus versátil e independiente, escalable en potencia, de elevada densidad de integración, seguridad operativa, protección de equipo y sencilla planificación.

Ventajas de los módulos Interface:

-Reducción de tiempos de instalación.

-Adaptación rápida.

-Costos reducidos.

Elementos

Controlador Fieldbus permite vincular el módulo de entrada y salida con los dispositivos de campo como sensores, actuadores.

Es un sistema abierto, las declaraciones están disponibles sin necesidad de licencias.

Hay dos partes importantes de la arquitectura del sistema Fieldbus: la interconexión y aplicación. La interconexión se refiere a la transmisión de datos desde un dispositivo a otro, puede ser un dispositivo de campo, operador de consola o un configurador.

Esta es la parte del protocolo de comunicación de bus de campo.

La aplicación es la función de automatización que el sistema realiza.

Elementos del controlador fieldbus:

- Interfaz fieldbus para conexión del bus.
- Suministro de energía para el sistema.
- Jumpers de contacto para suministrar energía a los módulos.
- Led indicadores de estado.
- Switch para Mac ID y velocidad de transmisión.

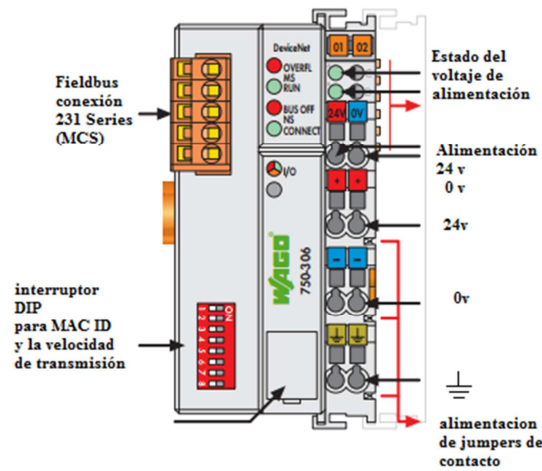


Figura 3.108: Controlador Fieldbus (WAGO, 2012)

Módulo de entradas

Los módulos de entradas digitales reciben señales de control desde los dispositivos de campo.

-Cada módulo de entrada tiene un filtro de ruido.

-Tiene un acoplador óptico que se utiliza para el aislamiento eléctrico entre el bus y los dispositivos del campo.

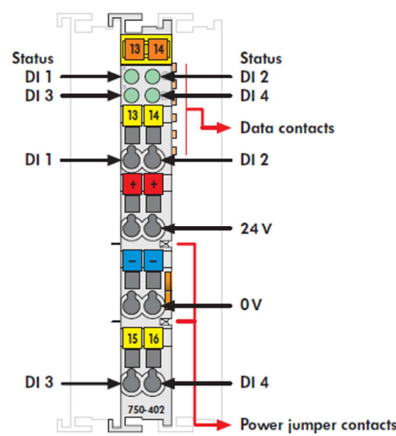


Figura 3.109: Módulo de entradas (WAGO, 2012)

Tabla 3.12: Datos técnicos del módulo de entradas

Cantidad de entradas	4
Configuración	vía PC o PLC
Consumo de corriente (interno)	7,5 mA
Voltaje de alimentación	a través de jumper
Tensión de señal	DC 15 V ... 30V
Filtro de entrada	3.0ms (750-402) 0,2 ms (750-403)
Aislamiento del sistema de alimentación	500V
Ancho de bits interno	4bits
Temperatura de operación	0 °C ... +55 °C

Módulo de salidas

-La carga conectada se conmuta a través de la salida digital del sistema de control.

-Todas las salidas están electrónicamente en corto circuito para su protección.

-Cada salida está aislada eléctricamente del bus mediante el uso de opto acopladores.

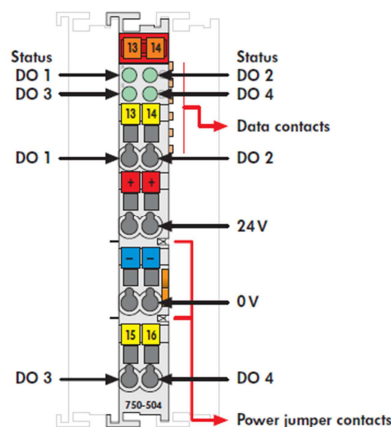


Figura 3.110: Módulo de salidas (WAGO, 2012)

Tabla 3.13: Datos técnicos del módulo de salidas

Cantidad de salidas	4
Configuración	vía PC o PLC
Consumo de corriente (interno)	7 mA
Voltaje de alimentación	Contactos DC 24 V (-25% +30%)
Tipo de cargas	resistivas, inductivas, lámparas
Frecuencia de conmutación máx.	1 kHz
Corriente de salida	0,5 A corto-circuito-protegido
Aislamiento del sistema / de alimentación	500V
Disipación de energía	0,3 J W max.
Ancho de bits interno	4 bits
Temperatura de operación	0 °C ... +55 °C

PROGRAMACIÓN DE ENTRADAS Y SALIDAS

La activación y desactivación de entradas y/o salidas es una asignación Booleana, donde true permite la activación y false la desactivación.

Ejemplos:

```
$in[1]==true
```

```
$in[1]==false
```

```
$out[1]=true
```

```
$out[1]=false
```

Variable signal: Permite la declaración de entradas y salidas son mencionadas al inicio del programa, también pueden declararse en el archivo **\$CONFIG.DAT** ahí se mantienen por tiempo indefinido y se puede leer desde cualquier programa o subprograma.

Tipo de variable: **SIGNAL**

Nombre: **pulso** (cualquier nombre)

Entrada: **\$in[5]** (número de entrada)

Ejemplo:

SIGNAL pulso \$in[5]

C. Configuración de condiciones previas

- Entrar a modo experto.
- Seleccionar **Configurar > Driver de E/S > Reconfiguración**.
- Revisar que el archivo que se abrió **IOSYS.INI**, tenga todas las líneas detalladas a continuación. Especialmente que no tenga comentado el DEVNET como se indica en el ANEXO1
- Guardar los cambios.
- Reiniciar el sistema.

D. ACTIVIDAD 1

1. Interconectar el panel didáctico auxiliar con el módulo de entradas y salidas.
 - 1.1 Realizar un programa que permita activar las salidas del módulo y encender las luces del panel didáctico auxiliar.



Figura 3.111: Panel didáctico auxiliar

- Conectar las salidas de la botonera en el módulo WAGO. Conectar la luz roja en la salida DO 1 y la luz verde en la salida DO 5.

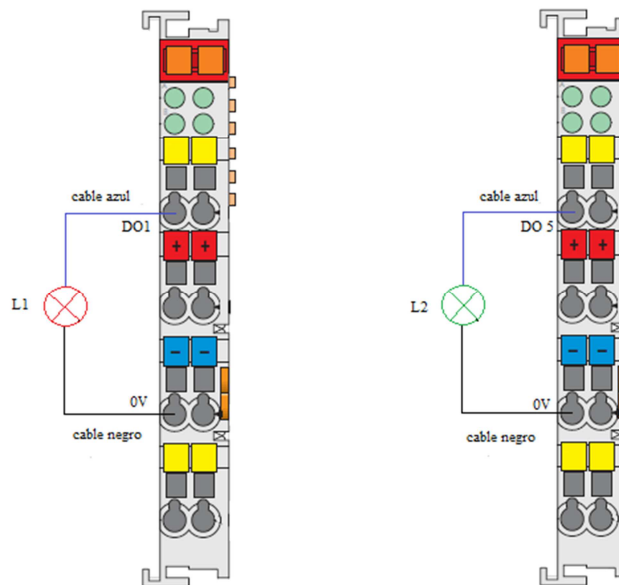


Figura 3.114: Diagrama de conexión de salidas

- Encender el KC 2, cambiar a modo experto.
- Verificar que exista conexión en **Configurar >Drivers >Reset** y **Configurar >Drivers >Reconfiguración E/S**

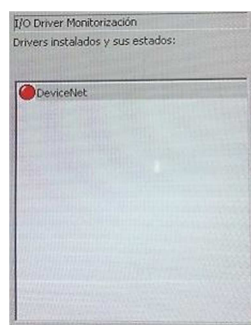


Figura 3.115: Conexión DeviceNet

- Seleccionar en el Menú **Indicación>Entradas/Salidas>Entrada Digital**. Revisar las entradas que se encuentran disponibles.

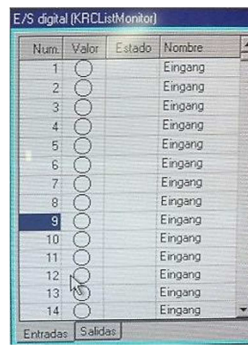


Figura 3.116: Entradas Digitales

8. Seleccionar la pestaña de salida, usar tab. Forzar la activación de las salidas con: **el botón de hombre muerto>seleccionar la salida>valor**. Observar y verificar que en el módulo que se enciendan las salidas seleccionadas.

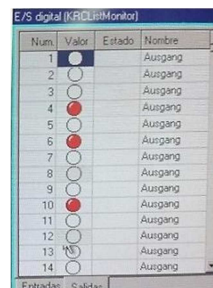


Figura 3.117: Salidas Digitales

ACTIVIDAD 2

1. Realizar un programa según el siguiente diagrama.

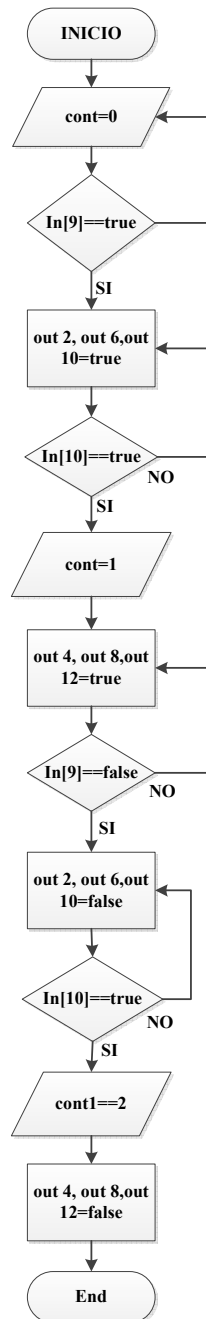


Figura 3.118: Diagrama de flujo para entradas y salidas

PROCEDIMIENTO ACTIVIDAD 2

1. Crear un nuevo programa llamado **prac_io**
2. Realizar el programa en KRL

```
DEF prac_io()  
decl int cont  
decl int cont1  
signal pulso $in[10] ; declarar la entrada 10, como el nombre pulso  
INI  
cont=0  
wait for $in[9]==true ; espera por entrada 9  
if $in[9]==true then; condicionar la entrada 9  
$out[2]=true ; enciende la salida 2  
$out[6]=true; enciende la salida 6  
$out[10]=true; enciende la salida 10  
endif; finalizar la sentencia  
wait for pulso==true; espera por la entrada 10  
if pulso==true then  
cont=1  
$out[4]=true; enciende la salida 4  
$out[8]=true; enciende la salida 8  
$out[12]=true; enciende la salida 12  
endif  
wait for $in[9]==false  
if $in[9]==false then  
$out[2]=false; apaga la salida 2  
$out[6]=false; apaga la salida 6  
$out[10]=false; apaga la salida 10  
Endif  
wait for pulso==true  
cont1=cont+1
```

if pulso==true and (cont1==2) then

\$out[4]=false; apaga la salida 4

\$out[8]=false; apaga la salida 8

\$out[12]=false; apaga la salida 12

Endif

```

1 DEF prac_io( )
2 decl int cont
3 decl int cont1
4 signal pulso $in[10]
5 INI
6 cont=0
7 wait for $in[9]==true
8 if $in[9]==true then
9   $out[2]=true
10  $out[6]=true
11  $out[10]=true
12 endif
13
14 wait for pulso==true
15 if pulso==true then
16   cont=1
17   $out[4]=true
18   $out[8]=true
19   $out[12]=true
20 endif
21
22 wait for $in[9]==false
23 if $in[9]==false then
24   $out[2]=false
25   $out[6]=false
26   $out[10]=false
27 endif
28
29 wait for pulso==true
30 cont1=cont+1
31 if pulso==true and (cont1==2) then
32   $out[4]=false
33   $out[8]=false
34   $out[12]=false
35 endif
36
37 |
38 END

```

Figura 3.119: Programa de secuencia de encendido y apagado de salidas

E. RESULTADOS

Switch on (IN9)

Num.	Valor	Estado	Nombre
1	<input type="radio"/>		Eingang
2	<input type="radio"/>		Eingang
3	<input type="radio"/>		Eingang
4	<input type="radio"/>		Eingang
5	<input type="radio"/>		Eingang
6	<input type="radio"/>		Eingang
7	<input type="radio"/>		Eingang
8	<input type="radio"/>		Eingang
9	<input checked="" type="radio"/>		Eingang
10	<input type="radio"/>		Eingang
11	<input type="radio"/>		Eingang
12	<input type="radio"/>		Eingang
13	<input type="radio"/>		Eingang
14	<input type="radio"/>		Eingang

Entradas Salidas

Num.	Valor	Estado	Nombre
1	<input type="radio"/>		Ausgang
2	<input checked="" type="radio"/>		Ausgang
3	<input type="radio"/>		Ausgang
4	<input type="radio"/>		Ausgang
5	<input type="radio"/>		Ausgang
6	<input checked="" type="radio"/>		Ausgang
7	<input type="radio"/>		Ausgang
8	<input type="radio"/>		Ausgang
9	<input type="radio"/>		Ausgang
10	<input checked="" type="radio"/>		Ausgang
11	<input type="radio"/>		Ausgang
12	<input type="radio"/>		Ausgang
13	<input type="radio"/>		Ausgang
14	<input type="radio"/>		Ausgang

Entradas Salidas

Figura 3.120: Activación de entrada 9 Figura 3.121: Activación de salidas 2, 6 y 10

Pulsador on (IN10)

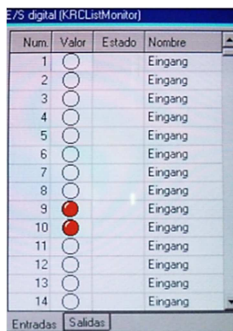


Figura 3.122: Activación de entrada 10

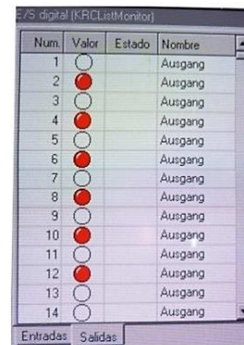


Figura 3.123: Activación de salidas 4,8,12

Switch off (IN9)

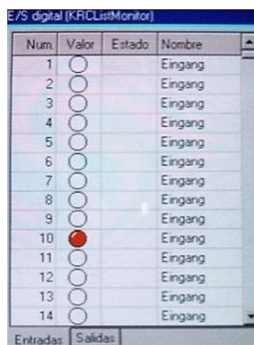


Figura 3.124: Desactivación de entrada 9



Figura 3.125: Desactivación de salidas 2,6 y 10

Pulsador off (IN10)

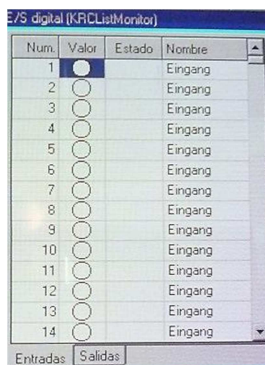


Figura 3.126: Desactivación de entradas

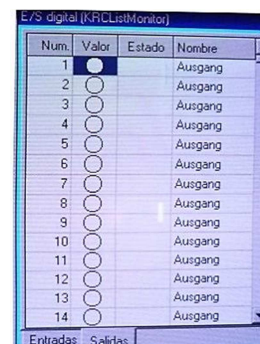


Figura 3.127: Desactivación de todas las salidas

3.10 PROCESO DE PALETIZACIÓN

A. OBJETIVOS

Objetivo General:

- Realizar un proceso de paletizado de un número determinado de cajas en modo de servicio automático.

Objetivos Específicos:

- Utilizar estructuras y formularios inline para obtener una ubicación exacta de las cajas.
- Tomar las cajas de la banda transportadora y ubicarla sobre la mesa de trabajo.

B. MARCO TEORICO

PROCESO DE PALETIZACIÓN

El proceso de paletizado consiste en agrupar sobre una superficie (pallet, tarima, paleta) cierta cantidad de objetos ubicados de forma individual cuyas características generalmente son frágiles, pesados o voluminosos, con la finalidad de formar una unidad de manejo que pueda ser transportada y almacenada con el mínimo esfuerzo, en una sola operación a su vez producir el aumento en la productividad, disminuyendo los tiempos de carga, descarga, almacenamiento y daños en los productos por la manipulación. Para lograr estos objetivos se debe cumplir las siguientes condiciones:

- La altura máxima a la que se coloca la carga debe ser de fácil acceso para su descarga.
- Una mala colocación puede provocar derrumbes.
- Para almacenar paletas se puede instalar estanterías.

C. ACTIVIDAD

Realizar un proceso de paletizado. que permita ubicar ocho cajas de 10X10 cm ordenando en dos filas y dos columnas, en modo de servicio automático. con las condiciones que indica diagrama de flujo de la figura 3.128.

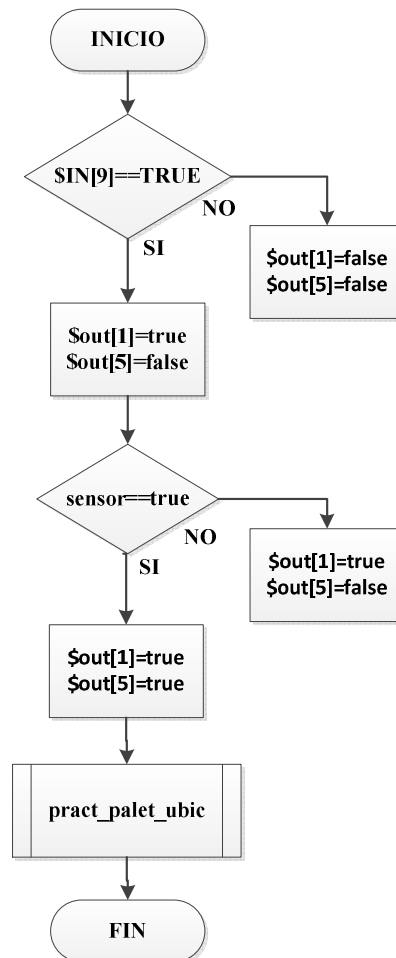


Figura 3.128: Diagrama de condiciones iniciales

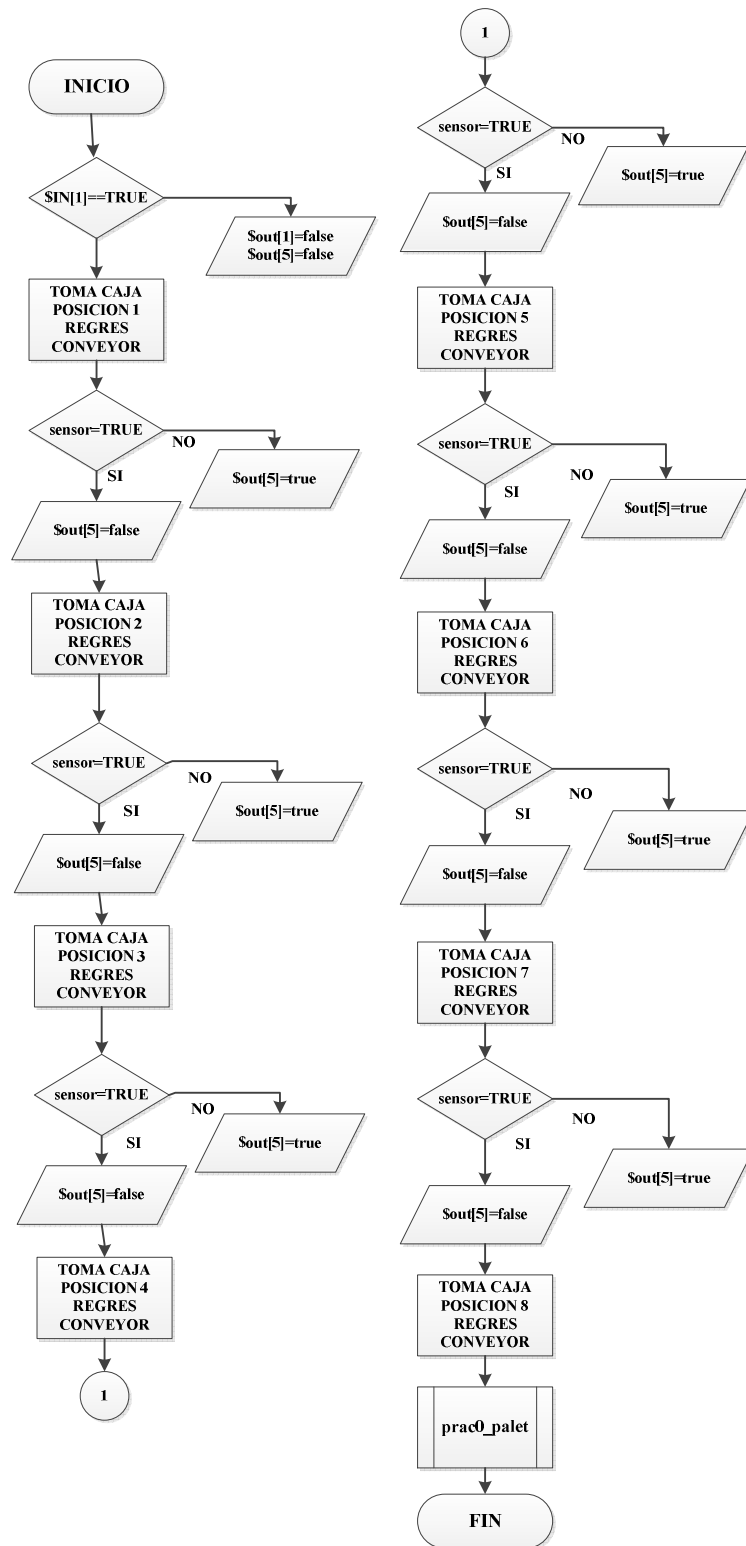


Figura 3.129. Diagrama de ubicación de cajas

D. PROCEDIMIENTO

1. Crear un programa llamado **prac_pallet**
2. Definir las señales en el **\$CONFIG.DAT**. Entrar a modo experto, abrir la carpeta **R1>System>\$config**. Definir las entradas como variables de tipo SIGNAL.

```

1  DEFDAT  $CONFIG
2  BASISTECH GLOBALS
3  AUTOEXT GLOBALS
4  ARCTECHDIGITAL GLOBALS
5  USER GLOBALS
6
7  [INT VALOR_GLOB=1554
8  INT conts1
9  SIGNAL onoff $in[9]
10 SIGNAL sensor $in[10]
11 ENDDAT

```

Figura 3.130: Entradas tipo signal en archivo Archivo \$Config

3. Crear un programa llamado **prac0_palet()**, que permitirá dar las condiciones iniciales del proceso.

```

DEF prac0_palet()
INI; inicio del programa
wait for $in[9]==true ; espera por entrada 9
if $in[9]==true then ; condiciona entrada 9, si es verdadera
$out[1]=true; enciende la salida 1
endif
wait for sensor==true ; espera por entrada 10
if sensor==true then; condiciona entrada 10, si es verdadera
$out[1]=false ; apaga la salida 1
$out[5]=true; enciende la salida 5
endif
prac0_palet_ubic() ; va a la subrutina prac0_palet_ubic
END

```

5. Crear un subprograma llamado `prac0_palet()_ubic()`, que permitirá ubicar la caja.

```
DEF prac0_palet_ubic( )
INI
wait for $in[9]==true
;toma la caja
PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
PTP P3 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
PTP P4 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1
;posicion1
PTP P30 Vel=100 % PDAT30 Tool[8]:tesis Base[7]:basetesis1
PTP {x 25, y 186, z 160, a -28, b -86,c 34}
PTP P32 Vel=100 % PDAT32 Tool[8]:tesis Base[7]:basetesis1
;regresa al conv
PTP P39 Vel=100 % PDAT39 Tool[8]:tesis Base[7]:basetesis1
PTP P40 Vel=100 % PDAT40 Tool[8]:tesis Base[7]:basetesis1
$out[1]=true
;espera caja
wait for sensor==true
if sensor==true then
$out[1]=false
;toma la caja
PTP P49 Vel=100 % PDAT49 Tool[8]:tesis Base[7]:basetesis1
PTP P50 Vel=100 % PDAT50 Tool[8]:tesis Base[7]:basetesis1
PTP P51 Vel=100 % PDAT51 Tool[8]:tesis Base[7]:basetesis1
PTP P52 Vel=100 % PDAT52 Tool[8]:tesis Base[7]:basetesis1
;posicion2
PTP P33 Vel=100 % PDAT33 Tool[8]:tesis Base[7]:basetesis1
PTP {x 125, y 186, z 160, a -28, b -86,c 34}
PTP P35 Vel=100 % PDAT35 Tool[8]:tesis Base[7]:basetesis1
```

```
;regresa conv
PTP P41 Vel=100 % PDAT41 Tool[8]:tesis Base[7]:basetesis1
PTP P42 Vel=100 % PDAT42 Tool[8]:tesis Base[7]:basetesis1
$out[1]=true
endif
;espera caja
wait for sensor==true
if sensor==true then
$out[1]=false
;toma caja
PTP P53 Vel=100 % PDAT53 Tool[8]:tesis Base[7]:basetesis1
PTP P54 Vel=100 % PDAT54 Tool[8]:tesis Base[7]:basetesis1
PTP P55 Vel=100 % PDAT55 Tool[8]:tesis Base[7]:basetesis1
PTP P56 Vel=100 % PDAT56 Tool[8]:tesis Base[7]:basetesis1
;posicion3
PTP P129 Vel=100 % PDAT129 Tool[8]:tesis Base[7]:basetesis1
PTP {x 125, y 286, z 160, a -28, b -86,c 34}
PTP P131 Vel=100 % PDAT131 Tool[8]:tesis Base[7]:basetesis1
;regresa conv
PTP P43 Vel=100 % PDAT43 Tool[8]:tesis Base[7]:basetesis1
PTP P44 Vel=100 % PDAT44 Tool[8]:tesis Base[7]:basetesis1
$out[1]=true
endif
;espera caja
wait for sensor==true
if sensor==true then
$out[1]=false
;toma la caja
PTP P78 Vel=100 % PDAT78 Tool[8]:tesis Base[7]:basetesis1
PTP P79 Vel=100 % PDAT79 Tool[8]:tesis Base[7]:basetesis1
PTP P80 Vel=100 % PDAT80 Tool[8]:tesis Base[7]:basetesis1
PTP P81 Vel=100 % PDAT81 Tool[8]:tesis Base[7]:basetesis1
```

```
;posicion4
```

```
PTP P132 Vel=100 % PDAT132 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP {x 25, y 286, z 160, a -28, b -86,c 34}
```

```
PTP P133 Vel=100 % PDAT133 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P134 Vel=100 % PDAT134 Tool[8]:tesis Base[7]:basetesis1
```

```
;regresa al conv
```

```
PTP P85 Vel=100 % PDAT85 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P86 Vel=100 % PDAT86 Tool[8]:tesis Base[7]:basetesis1
```

```
$out[1]=true
```

```
endif
```

```
;espera caja
```

```
wait for sensor==true
```

```
if sensor==true then
```

```
$out[1]=false
```

```
;toma la caja
```

```
PTP P87 Vel=100 % PDAT87 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P88 Vel=100 % PDAT88 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P89 Vel=100 % PDAT89 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P90 Vel=100 % PDAT90 Tool[8]:tesis Base[7]:basetesis1
```

```
;posicion5
```

```
PTP P135 Vel=100 % PDAT135 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP {x 25, y 186, z 60, a -28, b -86,c 34}
```

```
PTP P137 Vel=100 % PDAT137 Tool[8]:tesis Base[7]:basetesis1
```

```
;regresa al conv
```

```
PTP P94 Vel=100 % PDAT94 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P95 Vel=100 % PDAT95 Tool[8]:tesis Base[7]:basetesis1
```

```
$out[1]=true
```

```
endif
```

```
;espera caja
```

```
wait for sensor==true
```

```
if sensor==true then
```

```
$out[1]=false
```

```
;toma la caja
```

```
PTP P96 Vel=100 % PDAT96 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P97 Vel=100 % PDAT97 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P98 Vel=100 % PDAT98 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P99 Vel=100 % PDAT99 Tool[8]:tesis Base[7]:basetesis1
```

```
;posicion6
```

```
PTP P138 Vel=100 % PDAT138 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP {x 125, y 196, z 60, a -28, b -86,c 34}
```

```
PTP P140 Vel=100 % PDAT140 Tool[8]:tesis Base[7]:basetesis1
```

```
;regresa al conv
```

```
PTP P103 Vel=100 % PDAT103 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P104 Vel=100 % PDAT104 Tool[8]:tesis Base[7]:basetesis1
```

```
$out[1]=true
```

```
endif
```

```
;espera caja
```

```
wait for sensor==true
```

```
if sensor==true then
```

```
$out[1]=false
```

```
;toma la caja
```

```
PTP P105 Vel=100 % PDAT105 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P106 Vel=100 % PDAT106 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P107 Vel=100 % PDAT107 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P108 Vel=100 % PDAT108 Tool[8]:tesis Base[7]:basetesis1
```

```
;posicion7
```

```
PTP P141 Vel=100 % PDAT141 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP {x 125, y 286, z 60, a -28, b -86,c 34}
```

```
PTP P143 Vel=100 % PDAT143 Tool[8]:tesis Base[7]:basetesis1
```

```
;regresa al conv
```

```
PTP P112 Vel=100 % PDAT112 Tool[8]:tesis Base[7]:basetesis1
```

```
PTP P113 Vel=100 % PDAT113 Tool[8]:tesis Base[7]:basetesis1
```

```
$out[1]=true
```

```
endif
```

```
;espera caja
wait for sensor==true
if sensor==true then
$out[1]=false
;toma la caja
PTP P114 Vel=100 % PDAT114 Tool[8]:tesis Base[7]:basetesis1
PTP P115 Vel=100 % PDAT115 Tool[8]:tesis Base[7]:basetesis1
PTP P116 Vel=100 % PDAT116 Tool[8]:tesis Base[7]:basetesis1
PTP P117 Vel=100 % PDAT117 Tool[8]:tesis Base[7]:basetesis1
;posicion8
PTP P144 Vel=100 % PDAT144 Tool[8]:tesis Base[7]:basetesis1
PTP {x 25, y 296, z 60, a -28, b -86,c 34}
PTP P146 Vel=100 % PDAT146 Tool[8]:tesis Base[7]:basetesis1
;regresa al conv
PTP P121 Vel=100 % PDAT121 Tool[8]:tesis Base[7]:basetesis1
PTP P122 Vel=100 % PDAT122 Tool[8]:tesis Base[7]:basetesis1
$out[1]=true
endif
$out[5]=false
prac0_palet()
```

Realizar el proceso en modo automático, **Colocar la llave en modo automático> Seleccionar el programa> Presionar Accionamientos CON> Hombre muerto>Tecla de ejecución hacia adelante** (soltar después de iniciar la primera línea)

3.11 MISCELANEA DE PROCESOS CON SENTENCIAS CONDICIONALES

A. OBJETIVOS

Objetivo General:

- Presentar una serie de aplicaciones continuas y específicas.

Objetivos Específicos:

- Utilizar el panel didáctico auxiliar para validar la activación los procesos.
- Trabajar en modo de servicio en T2.

B. ACTIVIDAD

Utilizar los Sw1, Sw2, Sw3, para realizar combinaciones binarias que permitan activar procesos específicos según el diagrama de flujo

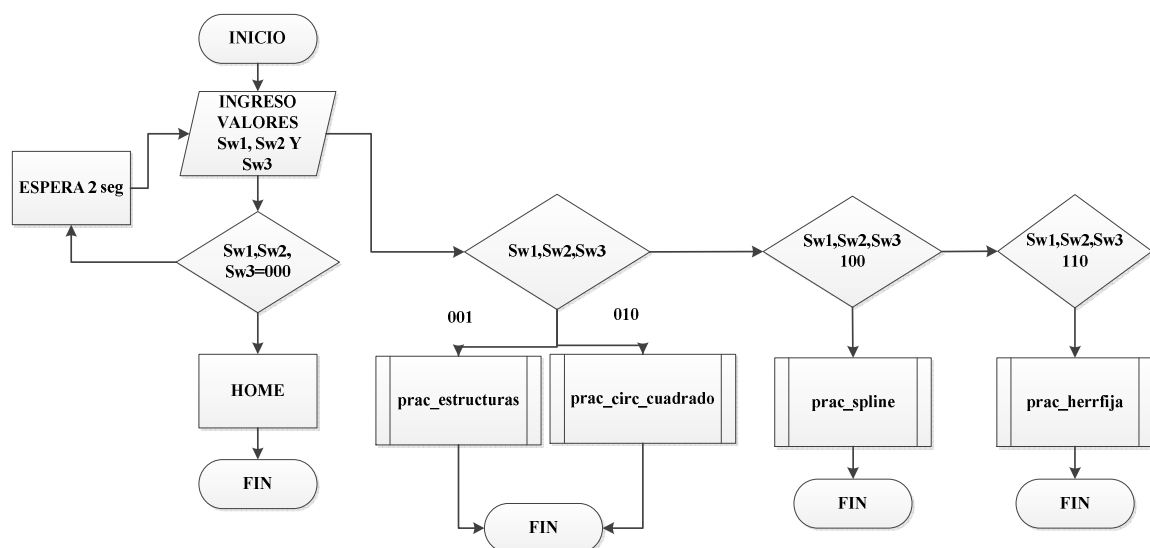


Figura 3.131: Diagrama de flujo del para el programa Miscelánea

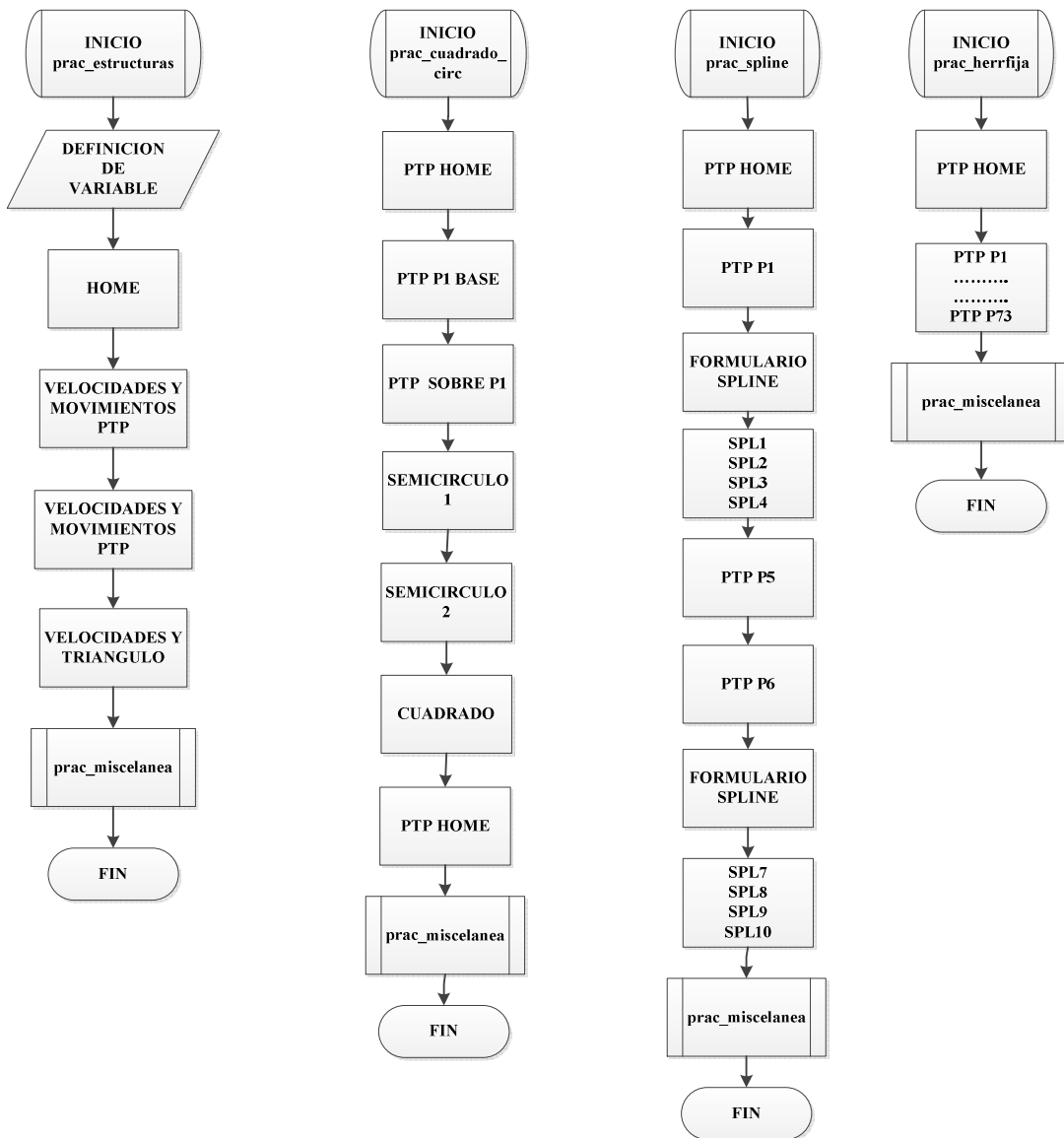


Figura 3.132: Subprogramas

1. Verificar el funcionamiento en la Tabla

Tabla 3.14: Combinaciones binarias para rutina miscelánea

COMBINACIÓN	SUBROUTINA
000	Home
001	prac_estructuras
010	prac_cir_cuadrado

011	Espera condición
100	prac_spline
101	espera condición
110	prac_herrfija
111	espera condición

C. PROCEDIMIENTO

1. Crear un programa llamado **prac_misclania**, según el diagrama de flujo de la figura

```

DEF prac_misclanea()
signal entradas $in[1] to $in[10] ; declaración de entradas
INI
loop
if not $in[1] and not $in[2] and not $in[5] then ; condición para 000
PTP HOME Vel=100 % DEFAULT
wait sec 2 ; espera 2 segundos para ingresar el dato
endif
switch entradas ; instrucción switch
case 1 ;condición 100
prac_spline_misc() ; va a la subrutina prac_spline
case 2; condición 010
prac_circ_cuadrado_misc() ; va a la subrutina prac_circ_cuadrado
endswitch
if not $in[1] and not $in[2] and $in[5] then ; condición para 001
prac_estructuras_misc()
endif
if $in[1] and $in[2] and not $in[5] then ; condición para 110
prac_herrfija_misc()

```

```

endif
endloop
END

```

2. Utilizar el programa realizado en la práctica estructuras “**prac_estructuras_misc**”

```

DEF prac_estructuras_misc()
decl AXIS HOME ; declaración para posición Home
INI
BAS(#INITMOV,0)
$base=$World ; tipo de coordenadas
$tool=$nullframe ; herramienta
HOME={AXIS: A1 0, A2 -90, A3 90, A4 0, A5 0, A6 0} ; coordenadas para
posiciónhome.
ptp HOME ; movimiento punto a punto para posición home.
$vel_axis[1]=50 ; velocidad 50% del eje 1
$vel_axis[4]=75; velocidad 75% del eje 4
$vel_axis[5]=100; velocidad 100% del eje 5
ptp {A1 90} ; movimiento punto a punto del eje 1 en un ángulo de 90°
ptp {A5 100}; movimiento punto a punto del eje 1 en un ángulo de 100°
ptp_rel{A4 50,A1 50}; movimiento punto a punto del eje 5 y 1 en un ángulo de 90°
$vel_axis[1]=30 ; velocidad 30% del eje 1
$vel_axis[2]= 30;velocidad 30% del eje 2
$vel_axis[3]=30; velocidad 30% del eje 3
$vel_axis[4]=30; velocidad 30% del eje 4
$vel_axis[5]=30; velocidad 30% del eje 5
$vel_axis[6]=30; velocidad 30% del eje 6
ptp HOME
ptp {pos: x 803.30,y 0.04, z 900,a 164.5,b 1.61,c 179.56,s 2,t 35} ;posición punto a
punto

```

```

lin {pos: x 803.29,y -54.29, z 899.99,a 164.50,b 1.61,c 179.56};posición punto a
punto
lin {pos: x 847.11,y -29.72, z 899.99,a 164.48,b 1.61,c 179.56};posición punto a
punto
lin {pos: x 803.00,y -0.87, z 899.99,a 164.48,b 1.61,c 179.56};posición punto a
punto
prac_misclanea()
END

```

3. Utilizar el programa “**prac_circ_cuadrado_misc**”.

```

DEF prac_circ_cuadrado_misc( )
INI
PTP HOME Vel= 100 % DEFAULT
PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
CIRC P3 P4 Vel=2 m/s CPDAT1 Tool[8]:tesis Base[7]:basetesis1
CIRC P5 P6 Vel=2 m/s CPDAT2 Tool[8]:tesis Base[7]:basetesis1
PTP P7 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
PTP P8 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1
LIN P9 Vel=2 m/s CPDAT3 Tool[8]:tesis Base[7]:basetesis1
LIN P10 Vel=2 m/s CPDAT4 Tool[8]:tesis Base[7]:basetesis1
LIN P11 Vel=2 m/s CPDAT5 Tool[8]:tesis Base[7]:basetesis1
LIN P12 Vel=2 m/s CPDAT6 Tool[8]:tesis Base[7]:basetesis1
prac_misclania()
END

```

4. Utilizar el programa “**prac_spline_misc**”

```

DEF prac_spline_misc( )
INI

PTP HOME Vel= 100 % DEFAULT

```

```
PTP P43 Vel=100 % PDAT6 Tool[8]:tesis Base[7]:basetesis1
SPLINE S2 Vel=2 m/s CPDAT5 Tool[8]:tesis Base[7]:basetesis1
SPL P3 Vel=2 m/s
SPL P4
SPL P6
SPL P7
SPL P8
PTP P44 Vel=100 % PDAT7 Tool[8]:tesis Base[7]:basetesis1
PTP P37 Vel=100 % PDAT5 Tool[8]:tesis Base[7]:basetesis1
SPLINE S6 Vel=2 m/s CPDAT10 Tool[8]:tesis Base[7]:basetesis1
SCIRC P40 P41
SPL P27
SCIRC P28 P29
SPL P30
SCIRC P31 P32
SPL P33
SCIRC P34 P35
SPL P36
prac_miscelanea()
END
```

5. Utilizar el programa “prac_herrfija_misc”

```
DEF prac_herrfija( )
INI
PTP HOME Vel= 100 % DEFAULT
PTP P1 Vel=100 % PDAT1 Tool[8]:tesis Base[7]:basetesis1
PTP P2 Vel=100 % PDAT2 Tool[8]:tesis Base[7]:basetesis1
PTP P3 Vel=100 % PDAT3 Tool[8]:tesis Base[7]:basetesis1
PTP P4 Vel=100 % PDAT4 Tool[8]:tesis Base[7]:basetesis1
PTP P5 Vel=100 % PDAT5 Tool[8]:tesis Base[7]:basetesis1
```

PTP P6 Vel=100 % PDAT6 Tool[8]:tesis Base[7]:basetesis1
PTP P7 Vel=100 % PDAT7 Tool[8]:tesis Base[7]:basetesis1
PTP P8 Vel=100 % PDAT8 Tool[8]:tesis Base[7]:basetesis1
PTP P9 Vel=100 % PDAT9 Tool[8]:tesis Base[7]:basetesis1
PTP P10 Vel=100 % PDAT10 Tool[8]:tesis Base[7]:basetesis1
PTP P11 Vel=100 % PDAT11 Tool[8]:tesis Base[7]:basetesis1
PTP P12 Vel=100 % PDAT12 Tool[8]:tesis Base[7]:basetesis1
PTP P13 Vel=100 % PDAT13 Tool[8]:tesis Base[7]:basetesis1
PTP P14 Vel=100 % PDAT14 Tool[8]:tesis Base[7]:basetesis1
PTP P15 Vel=100 % PDAT15 Tool[8]:tesis Base[7]:basetesis1
PTP P16 Vel=100 % PDAT16 Tool[8]:tesis Base[7]:basetesis1
PTP P17 Vel=100 % PDAT17 Tool[8]:tesis Base[7]:basetesis1
PTP P18 Vel=100 % PDAT18 Tool[8]:tesis Base[7]:basetesis1
PTP P19 Vel=100 % PDAT19 Tool[8]:tesis Base[7]:basetesis1
PTP P20 Vel=100 % PDAT20 Tool[8]:tesis Base[7]:basetesis1
PTP P21 Vel=100 % PDAT21 Tool[8]:tesis Base[7]:basetesis1
PTP P22 Vel=100 % PDAT22 Tool[8]:tesis Base[7]:basetesis1
PTP P23 Vel=100 % PDAT23 Tool[8]:tesis Base[7]:basetesis1
PTP P24 Vel=100 % PDAT24 Tool[8]:tesis Base[7]:basetesis1
PTP P25 Vel=100 % PDAT25 Tool[8]:tesis Base[7]:basetesis1
PTP P26 Vel=100 % PDAT26 Tool[8]:tesis Base[7]:basetesis1
PTP P27 Vel=100 % PDAT27 Tool[8]:tesis Base[7]:basetesis1
PTP P28 Vel=100 % PDAT28 Tool[8]:tesis Base[7]:basetesis1
PTP P29 Vel=100 % PDAT29 Tool[8]:tesis Base[7]:basetesis1
PTP P30 Vel=100 % PDAT30 Tool[8]:tesis Base[7]:basetesis1
PTP P31 Vel=100 % PDAT31 Tool[8]:tesis Base[7]:basetesis1
PTP P32 Vel=100 % PDAT32 Tool[8]:tesis Base[7]:basetesis1
PTP P33 Vel=100 % PDAT33 Tool[8]:tesis Base[7]:basetesis1
PTP P34 Vel=100 % PDAT34 Tool[8]:tesis Base[7]:basetesis1
PTP P35 Vel=100 % PDAT35 Tool[8]:tesis Base[7]:basetesis1
PTP P36 Vel=100 % PDAT36 Tool[8]:tesis Base[7]:basetesis1
PTP P37 Vel=100 % PDAT37 Tool[8]:tesis Base[7]:basetesis1

PTP P38 Vel=100 % PDAT38 Tool[8]:tesis Base[7]:basetesis1
PTP P39 Vel=100 % PDAT39 Tool[8]:tesis Base[7]:basetesis1
PTP P40 Vel=100 % PDAT40 Tool[8]:tesis Base[7]:basetesis1
PTP P41 Vel=100 % PDAT41 Tool[8]:tesis Base[7]:basetesis1
PTP P42 Vel=100 % PDAT42 Tool[8]:tesis Base[7]:basetesis1
PTP P43 Vel=100 % PDAT43 Tool[8]:tesis Base[7]:basetesis1
PTP P44 Vel=100 % PDAT44 Tool[8]:tesis Base[7]:basetesis1
PTP P45 Vel=100 % PDAT45 Tool[8]:tesis Base[7]:basetesis1
PTP P46 Vel=100 % PDAT46 Tool[8]:tesis Base[7]:basetesis1
PTP P47 Vel=100 % PDAT47 Tool[8]:tesis Base[7]:basetesis1
PTP P48 Vel=100 % PDAT48 Tool[8]:tesis Base[7]:basetesis1
PTP P49 Vel=100 % PDAT49 Tool[8]:tesis Base[7]:basetesis1
PTP P50 Vel=100 % PDAT50 Tool[8]:tesis Base[7]:basetesis1
PTP P51 Vel=100 % PDAT51 Tool[8]:tesis Base[7]:basetesis1
PTP P52 Vel=100 % PDAT52 Tool[8]:tesis Base[7]:basetesis1
PTP P53 Vel=100 % PDAT53 Tool[8]:tesis Base[7]:basetesis1
PTP P54 Vel=100 % PDAT54 Tool[8]:tesis Base[7]:basetesis1
PTP P55 Vel=100 % PDAT55 Tool[8]:tesis Base[7]:basetesis1
PTP P56 Vel=100 % PDAT56 Tool[8]:tesis Base[7]:basetesis1
PTP P57 Vel=100 % PDAT57 Tool[8]:tesis Base[7]:basetesis1
PTP P58 Vel=100 % PDAT58 Tool[8]:tesis Base[7]:basetesis1
PTP P59 Vel=100 % PDAT59 Tool[8]:tesis Base[7]:basetesis1
PTP P60 Vel=100 % PDAT60 Tool[8]:tesis Base[7]:basetesis1
PTP P61 Vel=100 % PDAT61 Tool[8]:tesis Base[7]:basetesis1
PTP P62 Vel=100 % PDAT62 Tool[8]:tesis Base[7]:basetesis1
PTP P63 Vel=100 % PDAT63 Tool[8]:tesis Base[7]:basetesis1
PTP P64 Vel=100 % PDAT64 Tool[8]:tesis Base[7]:basetesis1
PTP P65 Vel=100 % PDAT65 Tool[8]:tesis Base[7]:basetesis1
PTP P66 Vel=100 % PDAT66 Tool[8]:tesis Base[7]:basetesis1
PTP P67 Vel=100 % PDAT67 Tool[8]:tesis Base[7]:basetesis1
PTP P68 Vel=100 % PDAT68 Tool[8]:tesis Base[7]:basetesis1

PTP P69 Vel=100 % PDAT69 Tool[8]:tesis Base[7]:basetesis1
PTP P70 Vel=100 % PDAT70 Tool[8]:tesis Base[7]:basetesis1
PTP P71 Vel=100 % PDAT71 Tool[8]:tesis Base[7]:basetesis1
PTP P72 Vel=100 % PDAT72 Tool[8]:tesis Base[7]:basetesis1
PTP P73 Vel=100 % PDAT73 Tool[8]:tesis Base[7]:basetesis1
prac_miscelanea()

3.12 SIMULACIÓN DE UN PROCESO DE ENSAMBLAJE EN KUKA SIMPRO

A. OBJETIVOS

Objetivo General:

- Diseñar un proceso completo de ensamblaje de piezas en una planta automotriz.

Objetivos Específicos:

- Utilizar los modelos de robots estudiados.
- Usar subrutinas para el desarrollo del proceso.

B. PROCEDIMIENTO

Diseñar un proceso de ensamblaje con las siguientes especificaciones:

- Realizar una pieza a partir de una plancha metálica.
- Movilizarla en una banda transportadora al área de ensamblaje.
- Ensamblar la pieza en un auto.

C. PROCEDIMIENTO DETALLADO

- Crear un programa nuevo en KUKA simpro.
- Crear un entorno de trabajo

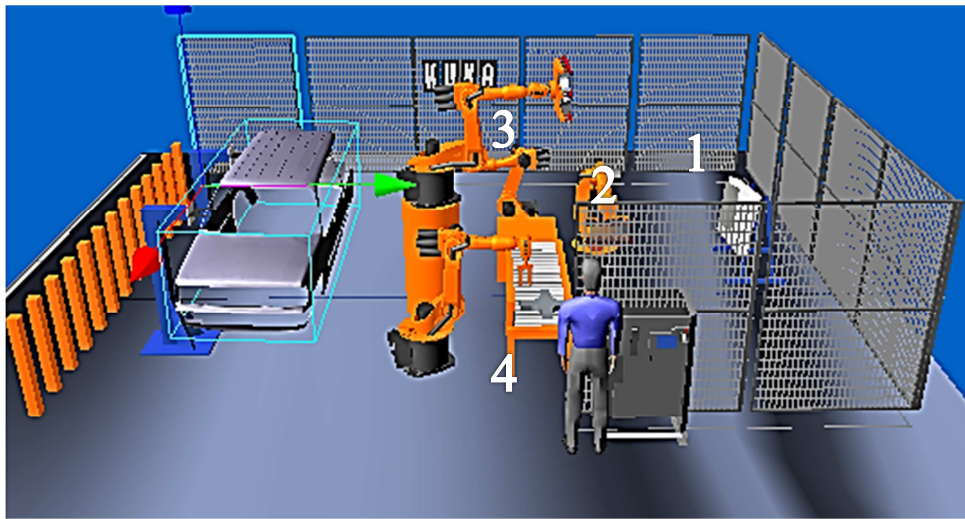


Figura 3.133: Objetos del entorno de trabajo

Programación Robot1: El robot1 toma la lámina de metal y la ubica sobre la mesa.



Figura 3.134: Programa robot 1

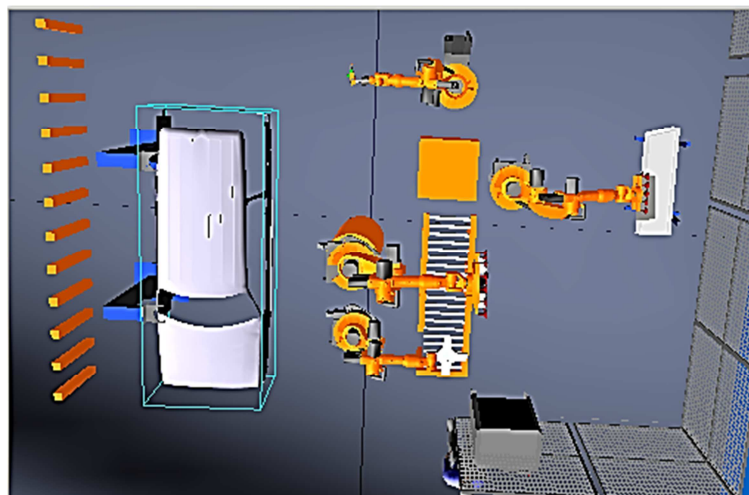


Figura 3.135: Resultado del robot1

Programación Robot2: El robot dos trabaja a partir de que tiene la lámina de metal, para ello se utiliza subrutinas y temporizadores.

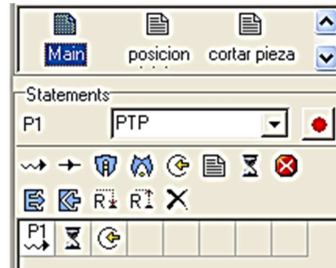


Figura 3.136: Rutina main del robot2

La subrutina posición ubica el robot sobre el área de trabajo y llama a la subrutina cortar pieza.

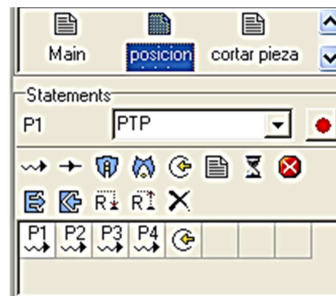


Figura 3.137: subrutina posición

La subrutina cortar pieza, realiza el corte de la pieza y termina el proceso del robot 2.

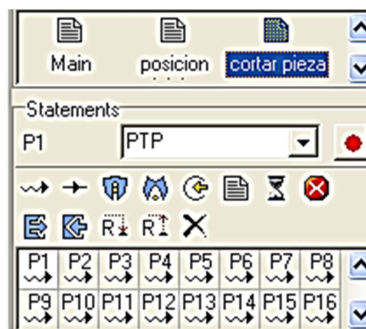


Figura 3.138: Subrutina cortar pieza

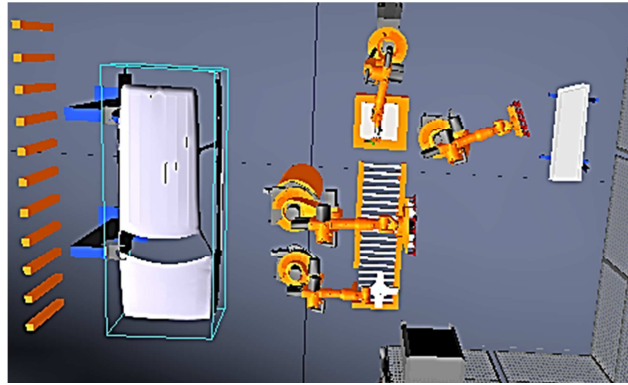


Figura 3.139: Resultado del robot2

Programación Robot3: El robot 3 toma la pieza terminada y la ubica en la banda transportadora. La rutina main, espera que la pieza esté terminada para empezar a trabajar.

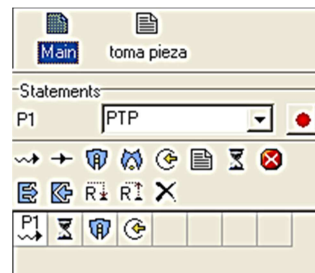


Figura 3.140: Rutina main del robot 3

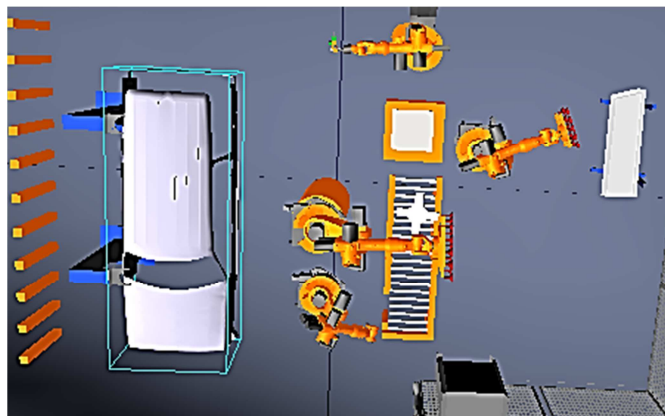


Figura 3.141: Resultado del robot 3

La subrutina toma pieza, indica los movimiento que realiza el robot para tomar la pieza y ubicarla en la banda transportadora.

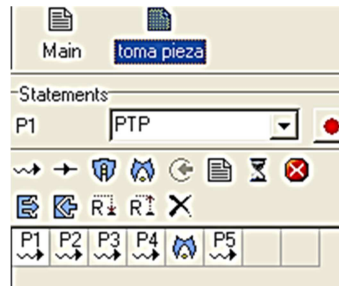


Figura 3.142: Subrutina toma pieza

Programación Robot4: Toma la pieza de la banda transportadora y la coloca en el auto.

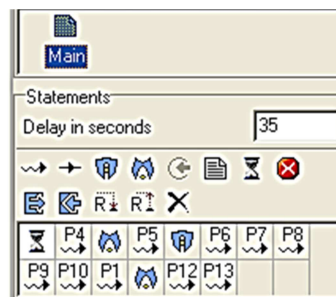


Figura 3.143: Programa del robot4

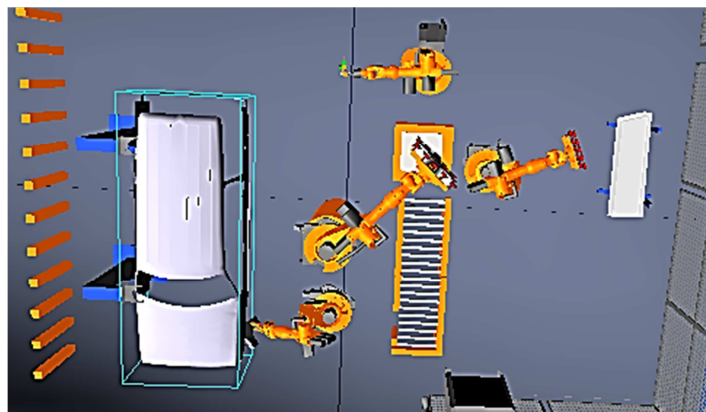


Figura 3.144: Resultado del robot

3.13 GUÍA PARA EL RESPALDO DE LA INFORMACIÓN

A. OBJETIVO

- Determinar el proceso para guardar la información de configuración y archivos de programación.

B. PROCEDIMIENTO

Guardar la información de los programas y parámetros configurados con los siguientes pasos.

1. Conectar un mouse y un flash en los puertos USB de la PC.

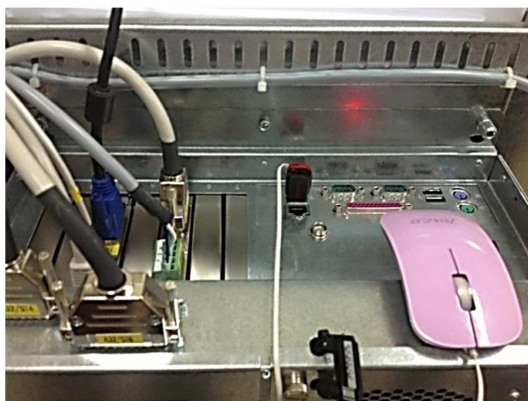


Figura 3.145: Conexión de flash memory y mouse

2. Mover el mouse a la posición que indica la figura y dar un click.

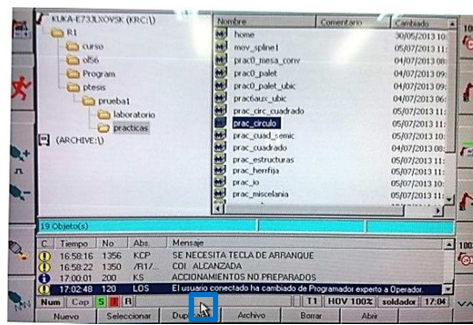


Figura 3.146: Entrar al sistema operativo Windows

3. Dar click en minimizar

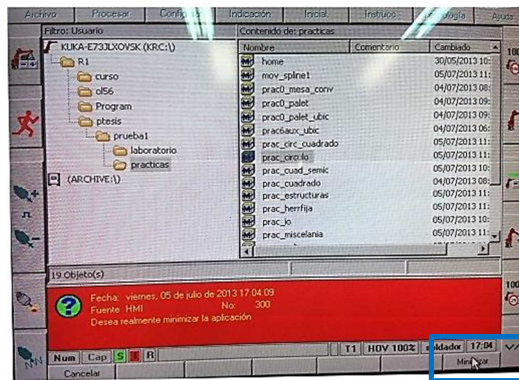


Figura 3.147: Minimizar la interfaz HMI KUKA

- 4. Verificar que la unidad en que se va a guardar sea U:
- 5. Regresar la interface, dar un click en KUKA HMI 2.0

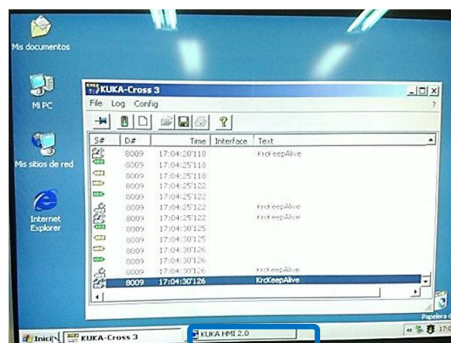


Figura 3.148: Ícono de interfaz HMI

6. Seleccionar Archivo>Archivo>Todo

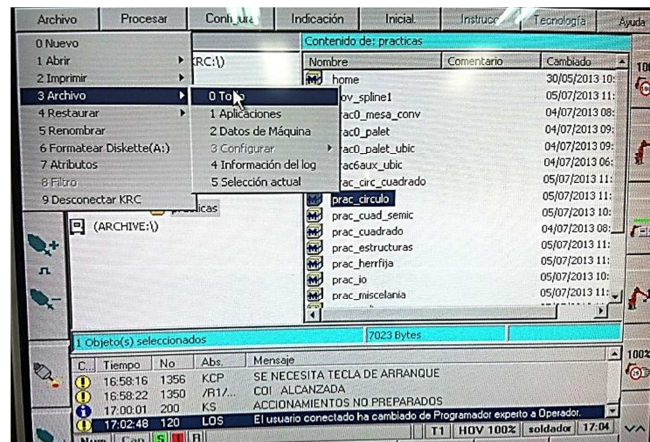


Figura 3.149: Guardar los archivos de configuración

7. Aceptar el mensaje, seleccionar **SI** y esperar hasta que termine el proceso de almacenamiento.

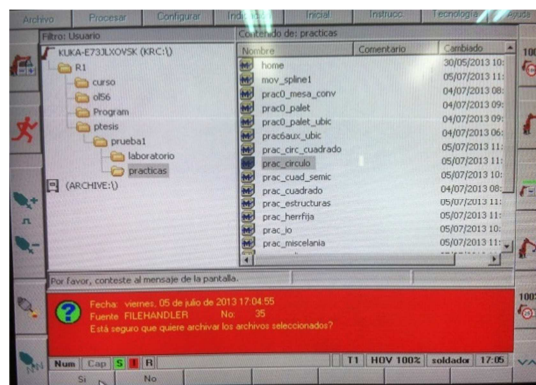


Figura 3.150: Confirmación de proceso de almacenamiento.

CAPÍTULO 4

CONCLUSIONES

4.1 CONCLUSIONES:

1. El diseño de las prácticas estaba basado en una estructura sistemática para afianzar el conocimiento teórico de la asignatura de Robótica Industrial para ser usadas a medida que avanza el proceso de enseñanza.
2. Las guías de laboratorio tienen las herramientas necesarias para el desarrollo de las mismas, están dotadas con una parte teórica, procedimiento detallado, diagramas de flujo, simulaciones, tareas propuestas y sus respectivas soluciones.
3. Se desarrolló prácticas en las que se involucra software y hardware afianzando conocimientos de conexión y configuración de equipos, al usar el módulo de entradas y salidas digitales que permite realizar aplicaciones reales como paletizado y soldadura.
4. Se realizó aplicaciones de procesos industriales como paletizado, con la ayuda de un panel didáctico auxiliar y los componentes que ofrece el lenguaje de programación de KUKA permitiendo realizar posicionamiento aproximado y exacto.

5. La gama de guías presentadas permiten que el estudiante tenga nuevas competencias para programar en modo usuario y experto, puesto que desarrollará prácticas básicas, intermedias y avanzadas.
6. El entorno de simulación KUKA SimPro permite tener una amplia visión de las funciones en cuanto a coordenadas y movimientos del robot, sin embargo la velocidad de los movimientos no es igual a la del robot real.
7. El desarrollo de estas prácticas nos hace competitivos en el campo laboral pues la ESPE dota a los estudiantes de equipos y conocimiento para ello ya que actualmente en el Ecuador existen varias empresas que utilizan en sus procesos manipuladores robóticos KUKA.

DEL SISTEMA:

8. El robot no va a operar aun cuando sus accionamientos estén activados, si no se confirma los mensajes que presenta la ventana mensajes.
9. El modo usuario no permite seleccionar ninguna plantilla, crea por defecto una plantilla del tipo Módulo y la vez crea un fichero SRC y un fichero DAT, que contiene un programa base para el mando del robot.
10. Es posible modificar las coordenadas de la posición HOME y automáticamente la nueva posición se cambia para todos los programas en los cuales se utilice dicha posición.
11. Un programa debe empezar con un movimiento PTP, esto permite que el desplazamiento COI (coincidente), que es un movimiento automático con velocidad reducida tome la posición actual y continúe con la trayectoria programada, caso contrario si los movimientos coinciden, el robot se quedara detenido.

12. Cuando se requiere hacer una modificación en el programa el robot no debe estar trabajando y el programa debe estar en modo de edición (no en ejecución) porque produce un salto a cualquier parte del programa y ocasiona peligro de colisión.
13. En el modo de servicio Automático los pulsadores de hombre muerto queda inhabilitado por lo tanto no permiten detener el programa. Para este efecto se debe usar el botón Accionamientos DESCONECTADOS, no el paro de emergencia, porque su uso excesivo causaría daños al sistema.
14. En el caso que los valores que se asigne \$ACC y \$VEL sean iguales y las longitudes de los pasos sean grandes la velocidad se reduce y realiza un posicionamiento exacto en el punto. Si las longitudes de los paso son pequeñas realiza un posicionamiento aproximado.
15. La declaración de varias entradas/salidas debe ser en orden ascendente y de tipo "signal", éstas están protegidas contra escritura.
16. Cuando existe un desajuste en el eje A4 también se produce en los ejes A5 y A6 porque se encuentran acoplados mecánicamente. Las causas de esto son los cambios de posiciones bruscas a altas velocidades.

4.2 RECOMENDACIONES:

1. El estudiante deberán tener conocimientos previos de las características de los equipos que va a operar.
2. Para realizar trabajos de ajuste o programación que requieran se debe trabajar en modo de funcionamiento manual a baja velocidad T1 nunca en T2.
3. Se recomienda la construcción de una celda y limitar el área donde el operario puede trabajar, para evitar accidentes y porque es una norma de seguridad.
4. No modificar las variables que empiezan con \$ porque se puede desconfigurar parte del sistema. Sin embargo se usan los archivos que tienen \$ para declarar variables.
5. Al programar los movimientos se debe prever que al ejecutar el programa los cables no se enrollen y ocasionen daños.
6. Se debe realizar el mantenimiento de cambio de aceite de 4 a 5 años cuando su uso es académico.
7. Se recomienda utilizar un brazalete antiestático para la conexión del módulo WAGO.
8. Apagar el sistema para realizar las conexiones y desconexiones del módulo WAGO, sino se trabaja bajo esa condición, el módulo se quema.
9. Verificar que la unidad para almacenar la configuración del sistema o los programas este direccionado a un flash memory Unidad U:, caso contrario estará direccionada al diskette.

REFERENCIAS BIBLIOGRÁFICAS

- RIA. (2010). Recuperado el 5 de Marzo de 2013, de [http:// RIA](http://RIA.com), Robotic Industry Association
- Artificial Intelligence Center. (s.f.). Obtenido de <http://www.ai.sri.com/shakey>
- Bueno, A. (s.f.). Recuperado el 20 de Marzo de 2013, de http://portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- Bueno, A. (s.f.). Recuperado el 15 de Marzo de 2013, de http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- Bueno, A. (s.f.). Recuperado el 15 de Marzo de 2013, de http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- Bueno, A. (s.f.). Recuperado el 15 de Marzo de 2013, de http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- Bueno, A. (s.f.). Recuperado el 15 de Marzo de 2013, de http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- Craig, J. (2008). Robotica manipuladores y robots móviles. España: Mrcrombo.
- GIZMODO. (s.f.). Obtenido de <http://www.gizmodo.fr/2011/12/22/en-1961-le-robot-unimate-etait-deja-fort-impressionnant.html>
- Gonzales, V. (s.f.). Recuperado el 2 de Abril de 2013, de http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm
- Gonzalez, V. (s.f.). Recuperado el 2 de Abril de 2013, de http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm
- HONDA. (s.f.). Obtenido de <http://world.honda.com/ASIMO>
- INSTITUTO CULTURAL TEHUACAN. (s.f.). Obtenido de <http://trvhgv.blogspot.com>
- KUKA. (s.f.). Obtenido de <http://www.kuka-robotics.com/es/company/group/milestones/1973.htm>

- KUKA. (s.f.). Obtenido de <http://www.kuka-robotics.com/es/company/group/milestones/2005.htm>
- KUKA. (s.f.). Recuperado el 15 de Febrero de 2013, de <http://www.kuka-robotics.com/es/company/group/milestones/2007.htm>
- KUKA. (s.f.). Recuperado el 17 de Febrero de 2013, de http://www.kuka-robotics.com/es/pressevents/news/nn_070629_robotcup.htm
- KUKA. (s.f.). Recuperado el 17 de Febrero de 2013, de <http://www.kuka-robotics.com/es/company/group/milestones/2009.htm>
- KUKA. (s.f.). Recuperado el 17 de Febrero de 2013, de http://www.kuka-robotics.com/es/products/industrial_robots/special/palletizer_robots/kr180_r3200
- KUKA. (s.f.). Recuperado el 17 de Febrero de 2013, de http://www.kuka-robotics.com/es/products/industrial_robots/small_robots/kr6_r900_sixx
- KUKA. (s.f.). Recuperado el 12 de Mayo de 2013, de <http://www.kuka-robotics.com/es/downloads/>
- KUKA. (s.f.). Recuperado el 15 de Mayo de 2013, de www.kuka-robotic.com
- KUKA. (2009). Kuka System Software 5.5. Alemania.
- KUKA. (2010). KUKA. (G. S. Center, Editor) Recuperado el 8 de Mayo de 2013, de <http://kuka-robotics.com>
- KUKA. (2011). Principios de la programación de robots. Alemania.
- KUKA. (2011). Principios de la programación de robots. Alemania.
- KUKA. (2011). Principios de la programación de robots (135 ed.). Alemania.
- KUKA. (2011). Principios de la programación de robots. Alemania.
- KUKA. (2011). Principios de la programación de robots. Alemania.
- KUKA GROUP. (2011). Robots KR 5ARC. Alemania.

- Kuka Roboter GmbH. (2009). KUKA System Software 5.5. Alemania.
- KUKA Roboter GmbH. (2009). KUKA System Software 5.5. Alemania.
- KUKA Roboter GmbH. (2009). KUKA System Software 5.5. Alemania.
- WAGO. (2012). Modulo digital. Estados Unidos.
- WAGO. (2012). Modulo entradas. Estados Unidos.
- WAGO. (2012). Módulo Salida. Estado Unidos.

INDICE DE FIGURAS

Figura 2.1: Brazo Industrial Robótico UNIMATE (GIZMODO).....	5
Figura 2.2: Robot móvil SHARKEY (Artificial Intelligence Center)	5
Figura 2.3: Robot Famulus (KUKA).....	6
Figura 2.4: Robot PUMA (INSTITUTO CULTURAL TEHUACAN).....	6
Figura 2.5: Robot móvil Genghis (TIME RIME)	7
Figura 2.6: Robot Humanoide ASIMO (HONDA).....	7
Figura 2.7: Robot KUKA SAFE (KUKA)	8
Figura 2.8: Robot KUKA TITAN (KUKA)	8
Figura 2.9: Robot LWR (KUKA).....	9
Figura 2.10: Robot KR 300 PA (KUKA).....	9
Figura 2.11: Robot QUANTEC (KUKA).....	10
Figura 2.12: Robot KR AGILUS (KUKA)	10
Figura 2.13: Analogía brazo humano vs brazo robótico (Bueno)	11
Figura 2.14: Robot Cartesiano (Bueno).....	12
Figura 2.15: Robot Polar (Bueno).....	13
Figura 2.16: Robot Cilíndrico (Bueno).....	13
Figura 2.17: Robot Antropomórfico (Bueno).....	14
Figura 2.18: Organización funcional de un sistema robótico.....	14
Figura 2.19: Lazo de control del manipulador robótico.....	15
Figura 2.20: Eslabón del Robot (Gonzalez).....	16
Figura 2.21: Articulaciones del Robot (Gonzales)	16
Figura 2.22: Teclado alfabético	23
Figura 2.23: Bloque numérico	24
Figura 2.24: Operación de la barra de estado	25
Figura 2.25: Sistema de coordenadas de robot (KUKA)	28
Figura 2.26: Robot KUKA KR 16 (KUKA, 2010).....	31
Figura 2.27: Dimensiones del Robot (Kuka Roboter GmbH, 2009)	31
Figura 2.28: Robot KUKA KR 5 ARC (KUKA)	32
Figura 2.29: Dimensiones del Robot KR 5ARC (KUKA Roboter GmbH, 2009)	33
Figura 2.30: Célula de capacitación (KUKA Roboter GmbH, 2009).....	34

Figura 2.31: KUKA.OfficeLite V5.6 OL	36
Figura 2.32: Interfaz de KUKA.OfficeLite V5.6 OL	37
Figura 2.33: Interfaz KUKA SimPro.....	37
Figura 2.34: Manipulador robótico modelo KR 5arc	38
Figura 2.35: Comunicación entre software.....	38
Figura 2.36: Ingreso del nombre de la máquina.....	39
Figura 2.37: Mensaje de selección.....	39
Figura 2.38: Archivos .Dat y .SRC.....	40
Figura 3.1: Ejes del robot KUKA KR 16.....	42
Figura 3.2: KUKA KR ARC5.....	43
Figura 3.3: Unidad de control KR C2.....	46
Figura 3.4: KUKA Control Panel frontal.....	47
Figura 3.5: Parte posterior del KCP.....	48
Figura 3.6: Interruptor de seguridad.....	48
Figura 3.7: Servo convertidores	49
Figura 3.8: Fuente de alimentación	50
Figura 3.9: Tarjeta de seguridad	50
Figura 3.10: Tarjeta MFC	51
Figura 3.11: Baterías.....	52
Figura 3.12: Módulo de entradas y salidas.....	52
Figura 3.13: Diagrama de conectores X11	53
Figura 3.14: Conexión de dispositivos.....	54
Figura 3.15: Grado de libertad del KUKA KR5 Y KR16 (Craig J. , 2008)	55
Figura 3.16: Ejes del Robot (KUKA GROUP, 2011).....	60
Figura 3.17: Space mouse en 0° y 270 (KUKA, 2009)	61
Figura 3.18: Interfaz KUKA.HMI	61
Figura 3.19: Switch de alimentación principal	62
Figura 3.20: Modo de servicio T1	62
Figura 3.21: Paro de emergencia.....	62
Figura 3.22: Selección desplazamiento eje específico con teclado.....	63
Figura 3.23: Teclas de accionamiento de ejes.	63
Figura 3.24: Tecla de override.....	64

Figura 3.25: Tecla de incremento.....	64
Figura 3.26: Ejes para movimiento de articulaciones.....	65
Figura 3.27: Herramienta del KUKA.....	67
Figura 3.28: Lista de bases disponibles.....	68
Figura 3.29: Lista de herramientas disponibles.....	69
Figura 3.30: Número y nombre de la base.....	69
Figura 3.31: Número y nombre de la herramienta.....	70
Figura 3.32: Punto de origen.....	70
Figura 3.33: Eje X positivo.....	70
Figura 3.34: Eje XY positivo.....	71
Figura 3.35: Posición del Robot en ejes Cartesiano y Específicos.....	71
Figura 3.36: Primera medición.....	72
Figura 3.37: Segunda medición.....	72
Figura 3.38: Tercera medición.....	73
Figura 3.39: Cuarta medición.....	73
Figura 3.40: Campo 5D y eje X.....	74
Figura 3.41: Posición paralela de las herramientas.....	74
Figura 3.42: Actualización de base y herramienta.....	74
Figura 3.43: Navegador del Operario.....	77
Figura 3.44: Movimiento PTP (KUKA, 2011).....	79
Figura 3.45: Movimiento LIN (KUKA, 2011).....	80
Figura 3.46: Movimiento CIR (KUKA, 2011).....	81
Figura 3.47: Movimiento SPLINE (KUKA, 2011).....	81
Figura 3.48: Movimiento SPLINE.....	82
Figura 3.49: Puntos de referencia para el cuadrado.....	83
Figura 3.50: Diagrama de flujo cuadrado.....	83
Figura 3.51: Puntos de referencia para la circunferencia.....	84
Figura 3.52: Diagrama de flujo círculo.....	84
Figura 3.53: Nueva Carpeta.....	85
Figura 3.54: Nuevo Programa.....	86
Figura 3.55: Posición HOME.....	86
Figura 3.56: Formulario Inline configuración PTP.....	87
Figura 3.57: Programación de la trayectoria PTP.....	87

Figura 3.58: Formulario Inline trayectoria LIN	88
Figura 3.59: Programación de la trayectoria LIN	88
Figura 3.60: Formulario Inline trayectoria CIR.....	88
Figura 3.61: Programación de la trayectoria CIRC.....	89
Figura 3.62: Programación del cuadrado	90
Figura 3.63: Programación del círculo.....	91
Figura 3.64: Resultado de las trayectorias LIN Y CIRC	92
Figura 3.65: Figura determinada	92
Figura 3.66: Control de simulación.....	96
Figura 3.67: Panel de mensajes.....	96
Figura 3.68: Barra de estado.....	96
Figura 3.69: Opciones del Teach	98
Figura 3.70: Herramientas de programación	99
Figura 3.71: Proceso en KUKA Sim Pro.....	99
Figura 3.72: Aplicación KUKA SimPro1.1	100
Figura 3.73: Interfaz de usuario.....	100
Figura 3.74: Manipulador KR16.....	101
Figura 3.75: Gripper tipo pinza	101
Figura 3.76: Opción Interactive Plug and Play	102
Figura 3.77: Opción Set parent node	102
Figura 3.78: Mesa	102
Figura 3.79: Banda transportadora.....	103
Figura 3.80: Cilindro	103
Figura 3.81: Hoja de la Base.....	104
Figura 3.82: Posición Home	104
Figura 3.83: Posición Home P1.....	104
Figura 3.84: Posición P2	105
Figura 3.85: Posición P3- tomar el cilindro.....	105
Figura 3.86: Posición P4- banda transportadora.....	105
Figura 3.87: Posición P5	106
Figura 3.88: Posición P6-P7.....	106
Figura 3.89: Posición P8- Regresar a posición home	106
Figura 3.90: Programación del proceso	107

Figura 3.91: Célula del operador	107
Figura 3.92: Puntos de referencia para un triángulo	110
Figura 3.93: Diagrama de flujo	110
Figura 3.94: Conexión a grupo experto	111
Figura 3.95: Ingreso de clave	111
Figura 3.96: Programación en KRL	113
Figura 3.97: Triángulo realizado con programación KRL	113
Figura 3.98: Estructura básica de un programa (KUKA, 2011)	116
Figura 3.99: Figuras a realizar.....	120
Figura 3.100: Diagrama de flujo	120
Figura 3.101: Resultado de la figura determinada.....	122
Figura 3.102: Figuras determinadas.....	122
Figura 3.103: Formulario Inline de trayectorias SPLINE	123
Figura 3.104: Diagrama de flujo del proceso.....	125
Figura 3.105: Programa desplazamiento mesa-banda transportadora	127
Figura 3.106: Programa de espera y toma de objeto	127
Figura 3.107: Programa de ubicación de objeto en repisa	128
Figura 3.108: Controlador Fieldbus (WAGO, 2012)	131
Figura 3.109: Módulo de entradas (WAGO, 2012)	131
Figura 3.110: Módulo de salidas (WAGO, 2012).....	132
Figura 3.111: Panel didáctico auxiliar.....	134
Figura 3.112: Módulo WAGO	135
Figura 3.113: Diagrama de conexión de entradas.....	135
Figura 3.114: Diagrama de conexión de salidas	136
Figura 3.115: Conexión DeviceNet.....	136
Figura 3.116: Entradas Digitales	137
Figura 3.117: Salidas Digitales.....	137
Figura 3.118: Diagrama de flujo para entradas y salidas	138
Figura 3.119: Programa de secuencia de encendido y apagado de salidas	140
Figura 3.120: Activación de entrada 9	140
Figura 3.121: Activación de salidas 2, 6 y 10	140
Figura 3.122: Activación de entrada 10.....	140
Figura 3.123: Activación de salidas 4,8,12.....	141

Figura 3.124: Desactivación de entrada 9.....	141
Figura 3.125: Desactivación de salidas 2,6 y 10.....	141
Figura 3.126: Desactivación de entradas.....	141
Figura 3.127: Desactivación de todas las salidas.....	141
Figura 3.128: Diagrama de condiciones iniciales	143
Figura 3.129. Diagrama de ubicación de cajas	144
Figura 3.130: Entradas tipo signal en archivo Archivo \$Config	145
Figura 3.131: Diagrama de flujo del para el programa Miscelánea	151
Figura 3.132: Subprogramas.....	152
Figura 3.133: Objetos del entorno de trabajo	161
Figura 3.134: Programa robot 1	161
Figura 3.135: Resultado del robot1	161
Figura 3.136: Rutina main del robot2	162
Figura 3.137: subrutina posición	162
Figura 3.138: Subrutina cortar pieza	162
Figura 3.139: Resultado del robot2	163
Figura 3.140: Rutina main del robot 3	163
Figura 3.141: Resultado del robot 3	163
Figura 3.142: Subrutina toma pieza	164
Figura 3.143: Programa del robot4.....	164
Figura 3.144: Resultado del robot	164
Figura 3.145: Conexión de flash memory y mouse	165
Figura 3.146: Entrar al sistema operativo Windows	166
Figura 3.147: Minimizar la interfaz HMI KUKA	166
Figura 3.148: Ícono de interfaz HMI	166
Figura 3.149: Guardar los archivos de configuración	167
Figura 3.150: Confirmación de proceso de almacenamiento.	167

INDICE DE TABLAS

Tabla 2.1: Tipos de articulaciones.....	17
Tabla 2.2: Cuadro comparativo entre los tipos de actuadores	19
Tabla 2.3: Elementos de la parte de delante del KCP	21
Tabla 2.4: Elementos de la parte posterior del KCP.....	25
Tabla 2.5: Simbología del bloque numérico	26
Tabla 2.6: Simbología del Submit.....	26
Tabla 2.7: Simbología del Accionamiento	27
Tabla 2.8: Simbología del estado del programa.....	27
Tabla 2.9: Nomenclatura del Robot.....	30
Tabla 2.10: Nomenclatura del Robot.....	32
Tabla 3.1: Características del Robot	42
Tabla 3.2 : Características de los ejes	43
Tabla 3.3: Características del Robot	44
Tabla 3.4: Características de los ejes	44
Tabla 3.5: Sistema de Referencia	59
Tabla 3.6: Sistema de Referencia	60
Tabla 3.7: Funciones de navegación de KUKA Sim.Pro	94
Tabla 3.8: Funciones de navegación de KUKA Sim.Pro	95
Tabla 3.9: Campos de KUKA Sim.Pro.....	97
Tabla 3.10: Iconos de programación de KUKA Sim.Pro.....	99
Tabla 3.11: Descripción de los ooperadores de comparación.....	119
Tabla 3.12: Datos técnicos del módulo de entradas.....	132
Tabla 3.13: Datos técnicos del módulo de salidas	133
Tabla 3.14: Combinaciones binarias para rutina miscelánea	152

GLOSARIO

C

- **CE:** Directiva de parlamento y consejo Europeo, legisla los estados miembros sobre el material eléctrico destinado a utilizarse con determinados límites de tensión.
- **COI:** (Coincidencia de paso) Coincidencia entre la posición del robot con las coordenadas del paso de movimiento.

D

- **Directorio:** Contenedor virtual, organiza y almacena ficheros.

E

- **Eje dominante:** Movimiento de un solo eje con el Space—Mouse
- **Ejes principales o básicos:** Son los ejes A1, A2 y A3 del robot, cuyos movimientos dependen uno del otro.
- **ESC:** (Electronic Safety Circuit) Sistema de seguridad, que permite un control permanente de la tarjetas de seguridad.

F

- **Fichero DAT:** Lista de datos, visible en el nivel del experto, contiene archivos de configuración del robot.
- **Fichero SRC:** Fichero del programa, visibles en el nivel del experto y usuario.

G

- **Grados de libertad:** Ejes que pueden moverse.

- **Grupos de usuarios:** “Usuario”, “Experto” dispone distintas funciones sobre la superficie de operación KUKA

H

- **Herramienta:** TOOL o gripper.
- **Herramienta externa:** Una herramienta que no está montada en el robot

I

- **Interpretador SUBMIT:** Un programa que se ejecuta cíclicamente en el segundo plano, que puede tener a su cargo el mando de operaciones de manejo y control en el entorno del robot.
- **Interrupción:** Un suceso asincrónico, que interrumpe la ejecución actual del programa, y que continúa con la ejecución de un programa asignado al suceso. Si este programa finalizó su ejecución, se reanuda la ejecución del programa antes interrumpido.

K

- **KRC:** (KUKA Robot Control) Denominación para la unidad de control del robot, compuesta de una sección de operación y un armario de control.
- **KCP:** (Kuka Control Panel) Panel de operación de la unidad de control del robot.
- **KRL:** (KUKA Robot Language) Idioma de programación especialmente desarrollado para la unidad de control del robot.
- **KSS:** (KUKA System Software) Conjunto de todo el software utilizado por la unidad de control del robot.

M

- **MFC** : (Tarjeta multifuncional) Se encuentra en la sección del ordenador del armario control y contiene una interfaz de conexión Ethernet así como para Device Net forma la interfaz entre PC y KCP.
- **Modo de ejecución del programa**: Es la forma de ejecución del programa. “Single-Step” (paso por paso), “Step” (por línea), “Go” (en forma corrida) y “en forma reversa”.
- **Modo de servicio**: Desplazamiento del robot en “Servicio de test/ servicio manual” (T1 o T2), “Automático” o “Automático Externo”
- **Modo de servicio automático**: El robot, en este modo de servicio, ejecuta su programa en forma automática.
- **Modo de servicio de Test/Manual**: El robot puede desplazarse en el Modo de servicio T1 o T2 de forma manual.

O

- **Override de programa (POV)**: La velocidad de proceso puede ser reducida.
- **Override manual (HOV)**: Puede reducirse la velocidad de desplazamiento manual

P

- **Parada de emergencia**: Parada inmediata del robot después de haber accionado el pulsador de parada de emergencia o si se abre una puerta de seguridad durante el servicio.
- **Parada exacta**: El robot se desplaza al punto y se detiene exactamente sobre el mismo

- **PTP:** Instrucción de movimiento para el desplazamiento más rápido entre dos puntos.
- **Pulsador de hombre muerto:** Pulsador que se encuentra en la parte trasera del KCP, que tiene que estar pulsado para ejecutar ciertas acciones como por ej. desplazamiento manual o arranque del programa.

S

- **Space-Mouse:** Elemento de operación para poder mover hasta seis ejes del robot a la vez.
- **Softkey:** Funciones que se activan con las teclas que corresponde al icono que se encuentre sobre ella.

T

- **TCP (Tool Center Point):** El punto de referencia de la herramienta activada (por ej. la punta de la herramienta).

X

- **XYZ- 4:** Puntos Programa de medición de la herramienta en el cual, el robot debe desplazarse con la herramienta a un punto de referencia desde cuatro direcciones distintas