

# KUKA

Training

KUKA Roboter GmbH

## Programación de robots 1

**KUKA System Software 8**

Documentación para la formación



Edición: 10.12.2013

Versión: P1KSS8 robot programming 1 (R2) V3



© Copyright 2013

KUKA Roboter GmbH  
Zugspitzstraße 140  
D-86165 Augsburg  
Alemania

La reproducción de esta documentación – o parte de ella – o su facilitación a terceros solamente está permitida con expresa autorización del KUKA Roboter GmbH.

Además del volumen descrito en esta documentación, pueden existir funciones en condiciones de funcionamiento. El usuario no adquiere el derecho sobre estas funciones en la entrega de un aparato nuevo, ni en casos de servicio.

Hemos controlado el contenido del presente escrito en cuanto a la concordancia con la descripción del hardware y el software. Aún así, no pueden excluirse totalmente todas las divergencias, de modo tal, que no aceptamos responsabilidades respecto a la concordancia total. Pero el contenido de estos escritos es controlado periódicamente, y en casos de divergencia, éstas son enmendadas y presentadas correctamente en la edición siguiente.

Reservados los derechos a modificaciones técnicas que no tengan influencia en el funcionamiento.

Traducción de la documentación original

KIM-PS5-DOC

Publicación:	Pub COLLEGE P1KSS8 Roboterprogrammierung 1 (R2) (PDF-COL) es
Estructura de libro:	P1KSS8 Roboterprogrammierung 1 (R2) V1.8
Versión:	P1KSS8 robot programming 1 (R2) V3

# Índice

<b>1 Estructura y función de un sistema de robot KUKA .....</b>	7
1.1 Vista general .....	7
1.2 Nociones básicas sobre robots .....	7
1.3 Mecánica de un robot KUKA .....	8
1.4 Unidad de control del robot (V)KR C4 .....	10
1.5 El KUKA smartPAD .....	12
1.6 Resumen smartPAD .....	14
1.7 Programación del robot .....	15
1.8 Seguridad del robot .....	16
<b>2 Mover un robot .....</b>	21
2.1 Vista general .....	21
2.2 Leer e interpretar los mensajes de la unidad de control del robot .....	21
2.3 Seleccionar y ajustar el modo de servicio .....	23
2.4 Desenchufar el smartPAD .....	27
2.5 Mover ejes del robot individualmente .....	30
2.5.1 Ejercicio: Operación y desplazamiento manual específico del eje .....	36
2.6 Sistemas de coordenadas en relación con los robots .....	37
2.7 Mover el robot en el sistema de coordenadas universales .....	38
2.7.1 Ejercicio: Operación y desplazamiento manual en el sistema de coordenadas universales	
43	
<b>3 Puesta en servicio del robot .....</b>	45
3.1 Resumen .....	45
3.2 Modo de puesta en servicio .....	45
3.3 Principio de ajuste .....	47
3.4 Ajustar el robot .....	49
3.4.1 Ejercicio: Ajuste de robot .....	56
3.5 Cargas sobre el robot .....	58
3.6 Datos de carga de la herramienta .....	58
3.7 Supervisar los datos de carga de la herramienta .....	59
3.8 Datos de carga adicional en el robot .....	60
3.9 Mover el robot en el sistema de coordenadas de la herramienta .....	62
3.9.1 Ejercicio: Desplazamiento manual en el sistema de coordenadas de la herramienta	66
3.10 Medición de una herramienta .....	67
3.10.1 Ejercicio: Medición de la herramienta clavija .....	77
3.10.2 Ejercicio: Medición de la herramienta garra, método de 2 puntos .....	80
3.11 Mover el robot en el sistema de coordenadas de base .....	84
3.11.1 Ejercicio: Procesos manuales en el sistema de coordenadas de base .....	88
3.12 Medición de una base .....	89
3.12.1 Ejercicio: Medición de la base mesa, método de 3 puntos .....	94
3.13 Consulta de la posición actual del robot .....	96
<b>4 Ejecutar los programas del robot .....</b>	99
4.1 Vista general .....	99
4.2 Realizar el desplazamiento de inicialización .....	99
4.3 Seleccionar e iniciar programas del robot .....	100
4.4 Ejercicio: Ejecutar programas del robot .....	107

<b>5 Uso de archivos de programas .....</b>	109
5.1 Vista general .....	109
5.2 Crear módulos de programa .....	109
5.3 Editar módulos de programa .....	110
5.4 Archivar y restaurar programas de robot .....	111
5.5 Registrar los cambios de programa y de estado con el listado LOG .....	113
<b>6 Crear y modificar movimientos programados .....</b>	117
6.1 Resumen .....	117
6.2 Creación de nuevas instrucciones de movimiento .....	117
6.3 Crear movimientos con optimización del tiempo de ciclo (movimiento del eje) .....	119
6.4 Ejercicio: Programa en el aire - Tratamiento de programa y movimientos PTP .....	125
6.5 Crear movimientos de trayectoria .....	127
6.6 Modificación de las instrucciones de movimiento .....	135
6.7 Ejercicio: Desplazamiento de trayectoria y posicionamiento aproximado .....	140
6.8 Programar movimientos con detección de colisión .....	143
<b>7 Usar funciones lógicas en el programa del robot .....</b>	149
7.1 Vista general .....	149
7.2 Introducción a la programación lógica .....	149
7.3 Programación de funciones de espera .....	150
7.4 Programación de funciones de conmutación simples .....	154
7.5 Programación de funciones de conmutación de trayectoria .....	157
7.6 Ejercicio: Instrucciones lógicas y funciones de conmutación .....	165
<b>8 Utilizar paquetes tecnológicos .....</b>	167
8.1 Vista general .....	167
8.2 Manejo de la garra con KUKA.GripperTech .....	167
8.3 Programación de garra con KUKA.GripperTech .....	168
8.4 Configuración de KUKA.GripperTech .....	171
8.5 Ejercicio: Programación de la garra, placa .....	174
8.6 Ejercicio: Programación de la garra, clavija .....	176
<b>9 Configuración y programación de herramientas externas .....</b>	179
9.1 Resumen .....	179
9.2 Mover el robot .....	179
9.2.1 Procesos manuales con una herramienta fija .....	179
9.2.2 Ejercicio: Procesos manuales con herramienta fija .....	181
9.3 Puesta en servicio del robot .....	182
9.3.1 Medición de una herramienta fija .....	182
9.3.2 Medición de una pieza guiada por robot .....	184
9.3.3 Ejercicio: Medir herramienta externa y pieza guiada por robot .....	186
9.4 Crear y modificar movimiento programado .....	191
9.4.1 Programación de movimiento con TCP externo .....	191
9.4.2 Ejercicio: Programación de movimientos con TCP externo .....	192
<b>10 Introducción al nivel del experto .....</b>	193
10.1 Resumen .....	193
10.2 Utilizar el nivel experto .....	193
10.3 Estructurar programas de robot .....	195

10.4	Concatenar programas de robot .....	198
10.5	Ejercicio: Programar una llamada a subprograma .....	201
<b>11</b>	<b>Variables y declaraciones .....</b>	<b>203</b>
11.1	Resumen .....	203
11.2	Gestión de datos en KRL .....	203
11.3	Trabajar con tipos de datos simples .....	206
11.3.1	Declaración de variables .....	206
11.3.2	Inicialización de variables con tipos de datos simples .....	208
11.3.3	Manipulación de valores de variables de tipos de datos simples con KRL .....	210
11.4	Visualizar variables .....	213
11.5	Ejercicio: Tipos de datos simples .....	215
<b>12</b>	<b>Utilización de controles de ejecución de programa .....</b>	<b>217</b>
12.1	Resumen .....	217
12.2	Programar bucles .....	217
12.2.1	Programar un bucle sinfín .....	217
12.2.2	Programar un bucle de conteo .....	219
12.2.3	Programar un bucle finito .....	221
12.2.4	Programar un bucle infinito .....	222
12.3	Programar consultas o ramificaciones .....	223
12.4	Programar distribuidores (SWITCH- CASE) .....	225
12.5	Programar instrucción de salto .....	227
12.6	Programar funciones de espera en KRL .....	229
12.6.1	Función de espera dependiente del tiempo .....	229
12.6.2	Función de espera dependiente de una señal .....	230
12.7	Ejercicio: Técnicas de bucles .....	233
<b>13</b>	<b>Trabajar con un control superior .....</b>	<b>235</b>
13.1	Vista general .....	235
13.2	Preparación para el inicio de programa del PLC .....	235
13.3	Adaptar el enlace PLC (Cell.src) .....	237
13.4	Preguntas: Trabajos en una unidad de control superior .....	239
<b>14</b>	<b>Anexo .....</b>	<b>241</b>
14.1	Configuración y aplicación del modo automático externo .....	241
14.2	Ejercicio: El modo Automático externo .....	250
14.3	Abreviaturas .....	251
<b>Índice .....</b>		<b>253</b>



# 1 Estructura y función de un sistema de robot KUKA

## 1.1 Vista general

Esta unidad incluye los siguientes contenidos:

- Introducción a la robótica
- Mecánica de un robot KUKA
- Unidad de control del robot KR C4
- KUKA smartPAD
- Programación del robot
- Seguridad en el robot

## 1.2 Nociones básicas sobre robots

### ¿Qué es un robot?

- El vocablo *robot* procede de la palabra eslava *robota*, que significa *trabajo duro*.
- La definición oficial del robot industrial es la siguiente: "Un robot es un manipulador controlado por un software que se puede programar libremente."
- También forman parte del sistema de robot la unidad de control y la unidad de operación, así como sus correspondientes cables de conexión y el software.



Fig. 1-1: Robot industrial

- 1 Control (armario de control (V)KR C4)
- 2 Manipulador (mecánica del robot)
- 3 Unidad manual de programación y de operación (KUKA smartPAD)

Todo aquello fuera de los límites del propio robot recibe el nombre de *Periféricos*:

- Herramientas (efector/herramienta)
- Dispositivo de protección
- Cintas transportadoras
- Sensores
- Máquinas
- etc.

### 1.3 Mecánica de un robot KUKA

#### ¿Qué es un manipulador?

El manipulador es la mecánica del robot propiamente dicha. Consta de un número de elementos (ejes) móviles encadenados entre sí. También recibe el nombre de cadena cinemática.

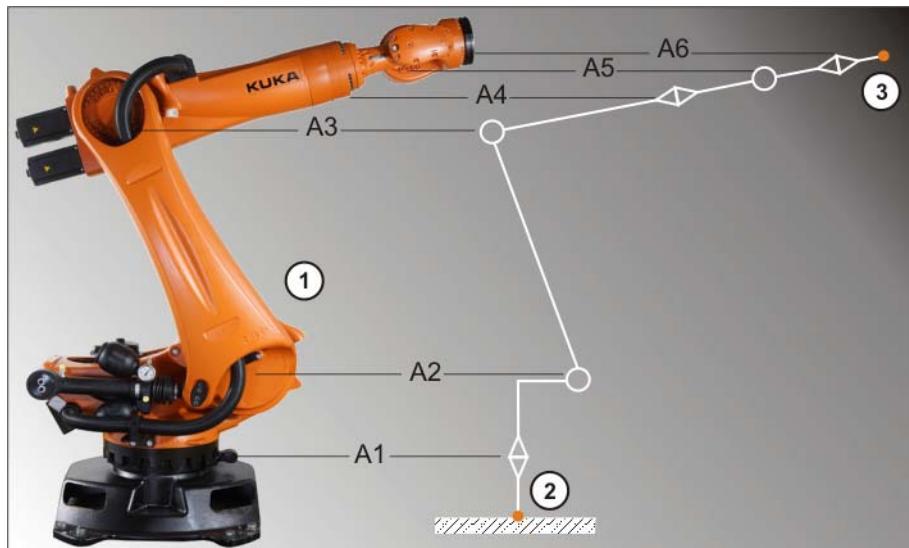
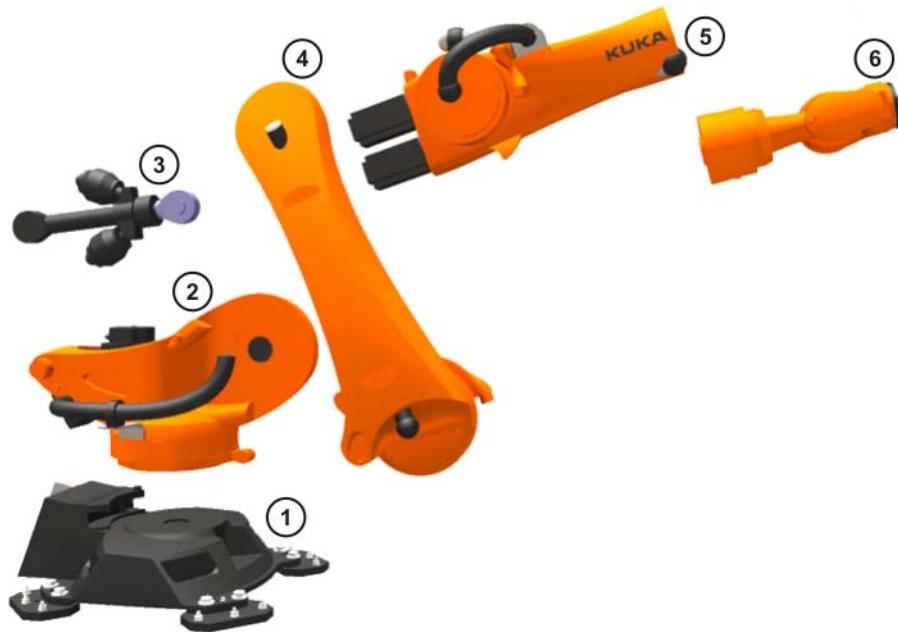


Fig. 1-2: Manipulador

- 1 Manipulador (mecánica del robot)
  - 2 Inicio de la cadena cinemática: Pie del robot (ROBROOT)
  - 3 Final libre de la cadena cinemática: Brida (FLANGE)
- A1...A6 Ejes del robot 1 al 6

El movimiento de los distintos ejes es ejecutado por la regulación selectiva de los servomotores. Éstos están unidos a los distintos componentes del manipulador por medio de reductores.

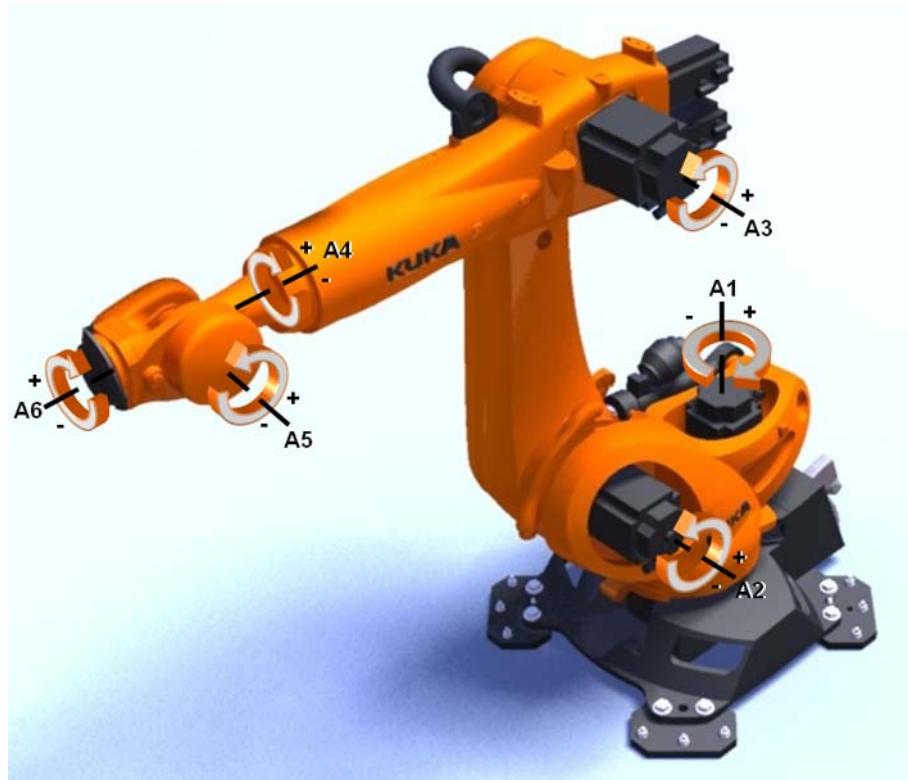


**Fig. 1-3: Resumen de los componentes mecánicos del robot**

- |                        |                       |
|------------------------|-----------------------|
| 1 Base                 | 4 Brazo de oscilación |
| 2 Columna giratoria    | 5 Brazo               |
| 3 Compensación de peso | 6 Muñeca              |

Los componentes mecánicos del robot están fabricados en su mayor parte con aluminio y fundición de acero. En casos aislados se utilizan también componentes de fibras de carbono.

Los ejes están numerados de abajo (pie del robot) a arriba (brida del robot):



**Fig. 1-4: Ejes de un robot KUKA**

Extracto de los datos técnicos de los manipuladores en los productos KUKA:

- **Número de ejes:**
  - 4 ejes en robots SCARA y de paralelogramo
  - 6 ejes en robots de brazo articulado vertical estándar
  - 7 ejes en robots de construcción liviana
- **Alcance:** de 0,35 m (KR 5 scara) a 3,9 m (KR 120 R3900 ultra K)
- **Peso propio:** de 20 kg a 4700 kg.
- **Exactitud:** repetibilidad de 0,015 mm a 0,2 mm.

Los rangos de movimiento de los ejes A1 hasta A3 y el eje de la muñeca A5 del robot se encuentran limitados por medio de topes finales mecánicos con amortiguadores.



Adicionalmente, pueden montarse topes finales mecánicos en los ejes adicionales.

#### AVISO

Si un robot o un eje adicional choca contra un obstáculo o bien un amortiguador en el tope mecánico o bien la limitación del campo de trabajo, puede ocasionar daños al sistema del robot. Antes de poner en marcha de nuevo el sistema del robot es obligatorio consultar con KUKA Roboter GmbH. Debe reemplazarse inmediatamente el amortiguador afectado por uno nuevo antes de reanudar la operación del robot. Si un robot (el eje adicional) choca contra un amortiguador con una velocidad mayor de 250 mm/s, debe cambiarse el robot (el eje adicional) o bien efectuarse una repuesta en marcha por KUKA Roboter GmbH.

## 1.4 Unidad de control del robot (V)KR C4

**¿Quién ejecuta el movimiento?** La mecánica del robot se mueve por medio de servomotores regulados por la unidad de control (V)KR C4.



**Fig. 1-5: Armario de control (V)KR C4**

**Características del control (V)KR C4)**

- Unidad de control del robot (cálculo de trayectoria): regulación de seis ejes de robot y, adicionalmente, de hasta dos ejes externos mediante la **unidad de control (V)KR C4**



**Fig. 1-6: (V)KR C4 Regulación de ejes**

- Unidad de control del robot (cálculo de trayectoria): regulación de seis ejes de robot y, adicionalmente, de hasta seis ejes externos mediante la **unidad de control (V)KR C4 extended**



**Fig. 1-7: (V)KR C4 extended - Regulación de ejes**

- Control de proceso opcional: Soft PLC integrado según IEC61131
- Control de seguridad
- Control de movimiento
- Opciones de comunicación por medio de sistemas de bus (p. ej. ProfiNet, Ethernet IP, Interbus):
  - Unidades de control con memoria programable (PLC)
  - Otras unidades de control
  - Sensores y actuadores
- Opciones de comunicación por red:
  - Ordenador principal
  - Otras unidades de control
  - Ordenador portátil de servicio



Fig. 1-8: Opciones de comunicación (V)KR C4

## 1.5 El KUKA smartPAD

**¿Cómo se opera con un robot KUKA?** Un robot KUKA se controla con un dispositivo de mando manual: el KUKA smartPAD.



**Fig. 1-9**

**Características del KUKA smartPAD:**

- Pantalla táctil (interfaz de usuario táctil) para operar con la mano o el lápiz que lleva integrado
- Display de gran formato y dimensiones
- Tecla de menú KUKA
- Teclas +/- de uso flexible (p. ej., como teclas de desplazamiento)
- Teclas para la operación de los paquetes tecnológicos
- Teclas para ejecutar el programa (paro/atrás/adelante)
- Tecla para mostrar el teclado
- Comutador de llave para cambiar el modo de servicio
- Pulsador de parada de emergencia
- Space Mouse
- Se puede desenchufar
- Conexión USB

## 1.6 Resumen smartPAD



Fig. 1-10

Pos.	Descripción
1	Botón para desenchufar el smartPAD
2	Interruptor de llave para acceder al gestor de conexiones. El conmutador únicamente se puede cambiar cuando está insertada la llave. El gestor de conexiones permite cambiar el modo de servicio.
3	Dispositivo de PARADA DE EMERGENCIA. Para detener el robot en situaciones de peligro. El dispositivo de PARADA DE EMERGENCIA se bloquea cuando se acciona.
4	Space Mouse: para el desplazamiento manual del robot.
5	Teclas de desplazamiento: para el desplazamiento manual del robot.
6	Tecla para ajustar el override de programa.
7	Tecla para ajustar el override manual.

Pos.	Descripción
8	Tecla de menú principal: muestra las opciones de menú en el smartHMI.
9	Teclas de estado. Las teclas de estado sirven principalmente para ajustar los parámetros de paquetes tecnológicos. Su función exacta depende del paquete tecnológico instalado.
10	Tecla de arranque: con la tecla de arranque se inicia un programa.
11	Tecla de arranque hacia atrás: con la tecla de arranque hacia atrás se inicia un programa en sentido inverso. El programa se ejecuta paso a paso.
12	Tecla STOP: con la tecla STOP se detiene un programa en ejecución.
13	Tecla del teclado:  Muestra el teclado. Generalmente no es necesario mostrar el teclado porque el smartHMI detecta cuándo es necesario introducir datos con el teclado y lo abre automáticamente.

## 1.7 Programación del robot

Con la programación del robot se consigue que los movimientos y procesos se ejecuten y repitan automáticamente. Para ello el control necesita gran cantidad de información:

- Posición actual del robot = posición de la herramienta (Tool) actual en el espacio (Base) actual
- Tipo de movimiento
- Velocidad/aceleración
- Informaciones de señales para condiciones de espera, ramificaciones, dependencias,...

**¿Qué lenguaje usa la unidad de control?**

El lenguaje de programación es el **KRL - KUKA Robot Language**.

Los programas sencillos se crean mediante formularios predefinidos (formularios inline). Para bucles se usa KRL.

Programa de ejemplo:

```

LOOP
PTP P1 Vel=100% PDAT1 Tool[2] Base[4]
PTP P2 Vel=100% PDAT2 Tool[2] Base[4]
WAIT FOR IN 10 'Part in Position'
PTP P3 Vel=100% PDAT3 Tool[2] Base[4]
LIN P4 Vel=2m/s CPDAT4 Tool[2] Base[4]
...
ENDLOOP

```

**¿Cómo se programa un robot KUKA?**

Para programar un robot KUKA existen varios métodos de programación:

- **Programación online** con el proceso de aprendizaje.



Fig. 1-11: Programación de robots con el KUKA smartPAD

- **Programación offline**
  - **Programación gráfica interactiva:** Simulación del proceso del robot

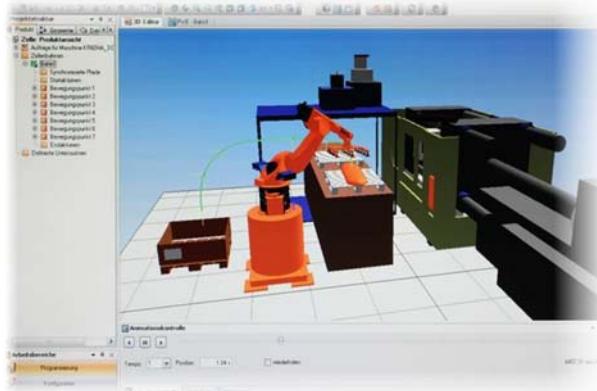


Fig. 1-12: Simulación con KUKA Sim

- **Programación textual:** Programación con ayuda de la superficie del smartPAD en un PC de control superpuesto (también para diagnósticos; la adaptación online se efectúa automáticamente con los programas en marcha).



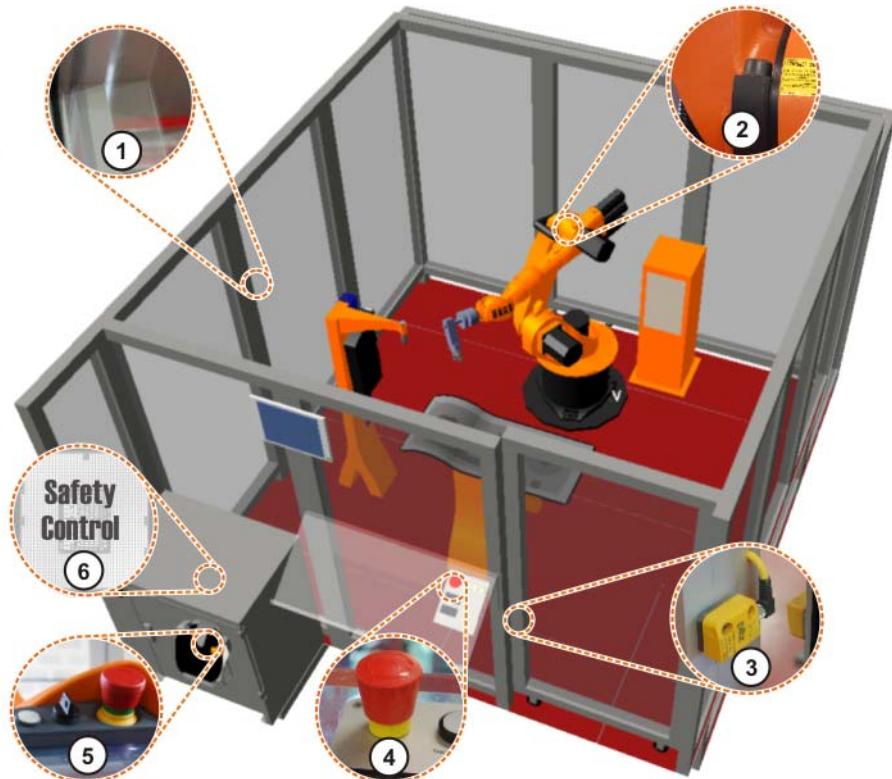
Fig. 1-13: Programación de robots con KUKA OfficeLite

## 1.8 Seguridad del robot

Un sistema robótico siempre debe contar con las características de seguridad pertinentes. Entre ellas se encuentran p. ej. los dispositivos de seguridad se-

paradores (vallas, puertas, etc.), pulsadores de parada de emergencia, dispositivos de verificación de hombre muerto, limitaciones de los campos de los ejes, entre otros.

### Ejemplo: Celda de formación College



**Fig. 1-14: Célula de capacitación**

- 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 (externo)
- 5 Pulsadores de parada de emergencia, pulsador de validación, interruptores con llave para abrir el gestor de conexiones
- 6 Control de seguridad (V)KR C4 integrado

**PELIGRO**

El sistema de robot puede causar daños personales o materiales si los dispositivos de seguridad no están en servicio. En caso de que se hayan desmontado o desactivado los dispositivos de seguridad, no se debe operar el sistema de robot.

### Dispositivo de parada de emergencia

El dispositivo de PARADA DE EMERGENCIA del robot industrial es el pulsador de PARADA DE EMERGENCIA de la KCP. El pulsador debe pulsarse en situaciones de peligro o en caso de emergencia.

Reacciones del robot industrial al pulsarse el pulsador de PARADA DE EMERGENCIA:

- El manipulador y los ejes adicionales (opcionales) se detienen con una parada de seguridad 1.

Para poder reanudar el servicio, debe desenclavarse el pulsador de PARADA DE EMERGENCIA por medio de un giro y, a continuación, confirmar el mensaje de error.

**⚠ ADVERTENCIA**

Las herramientas y otras dispositivos unidos al manipulador que puedan suponer algún peligro deben estar conectados desde la instalación al circuito de PARADA DE EMERGENCIA. Si no se respeta esta advertencia, pueden ocurrir importantes daños materiales, lesiones graves e incluso la muerte.

Como mínimo debe haber instalado un dispositivo externo de PARADA DE EMERGENCIA. Esto garantiza que se puede contar con un dispositivo de PARADA DE EMERGENCIA aún estando la KCP conectada.

**Parada de emergencia externa**

En todas las estaciones de operación que puedan accionar un movimiento del robot o crear una situación susceptible de ser peligrosa, se debe disponer de dispositivos de PARADA DE EMERGENCIA. El integrador de sistemas debe velar por ello.

Como mínimo debe haber instalado un dispositivo externo de PARADA DE EMERGENCIA. Esto garantiza que se puede contar con un dispositivo de PARADA DE EMERGENCIA aún estando la KCP conectada.

Los dispositivos externos de PARADA DE EMERGENCIA se conectan por medio de las interfaces del cliente. Los dispositivos externos de PARADA DE EMERGENCIA no se incluyen en el contenido de entrega del robot industrial.

**Protección del operario**

La señal **Protección del operario** sirve para bloquear dispositivos de protección separadores, p. ej. puertas de protección. Sin esta señal no es posible el modo de servicio automático. En caso de pérdida de señal durante el servicio automático (p. ej., se abre una puerta de protección), el manipulador realiza una parada de seguridad 1.

Para los modos de servicio de test Manual Velocidad reducida (T1) y Manual Velocidad alta (T2), la protección del operario no se encuentra activada.

**⚠ ADVERTENCIA**

Tras una pérdida de señal solo se podrá continuar el modo de servicio automático si el dispositivo de seguridad se ha cerrado de nuevo y si dicho cierre se ha confirmado. La confirmación debe evitar una reanudación del modo de servicio automático no intencionada hallándose personas dentro de la zona de peligro, como p. ej., en caso de una puerta de protección cerrada equivocadamente. La confirmación se debe implementar de forma que primero se pueda comprobar realmente la zona de peligro. Otras confirmaciones (p. ej. una confirmación que siga automáticamente al cierre del dispositivo de seguridad) no están permitidas. El integrador de sistemas es el responsable de que se cumplan estos requisitos. Si no se cumplen, pueden producirse daños materiales, lesiones graves o incluso la muerte.

**Parada de servicio segura**

La parada de servicio segura también se puede accionar a través de una entrada de la interfaz de cliente. El estado se mantiene mientras la señal externa permanezca en FALSE. Cuando la señal externa cambie a TRUE, se puede volver a desplazar el manipulador. No es necesario ninguna confirmación.

**Parada de seguridad externa 1 y parada de seguridad externa 2**

---

La parada de seguridad 1 y la parada de seguridad 2 se pueden accionar a través de una entrada de la interfaz de cliente. El estado se mantiene mientras la señal externa permanezca en FALSE. Cuando la señal externa cambie a TRUE, se puede volver a desplazar el manipulador. No es necesario ninguna confirmación.

---



## 2 Mover un robot

### 2.1 Vista general

Esta unidad incluye los siguientes contenidos:

- Lectura e interpretación de los mensajes
- Seleccionar y ajustar los modos de servicio
- Movimiento individual de los ejes del robot
- Movimiento del robot en el sistema de coordenadas universal
- Mover el robot en el sistema de coordenadas de herramienta
- Movimiento del robot en el sistema de coordenadas de base
- Desplazamiento manual con herramienta fija

### 2.2 Leer e interpretar los mensajes de la unidad de control del robot

Resumen de mensajes



Fig. 2-1: Ventana y contador de mensajes

- 1 Ventana de mensajes: aparece en pantalla el mensaje actual.
- 2 Contador de mensajes: cantidad de mensajes clasificados por tipo.

La unidad de control se comunica con el usuario a través de la ventana de mensajes. Se dispone de cinco tipos de mensajes distintos:

**Resumen de tipos de mensaje:**

Símbolo	Tipo
	<b>Mensaje de acuse de recibo</b> <ul style="list-style-type: none"> <li>■ Para mostrar los estados en los que se requiera la confirmación por parte del operador para que el programa siga ejecutándose (p. ej., "Conf. PARADA DE EMERGENCIA").</li> <li>■ Un mensaje de confirmación siempre provoca que el robot pare o no arranque.</li> </ul>
	<b>Mensaje de estado</b> <ul style="list-style-type: none"> <li>■ Los mensajes de estado informan de los estados actuales del control (p. ej., "PARADA DE EMERGENCIA")</li> <li>■ Los mensajes prestados no se pueden confirmar mientras el estado está pendiente.</li> </ul>

Símbolo	Tipo
	<p><b>Mensaje de observación</b></p> <ul style="list-style-type: none"> <li>Los mensajes de observación aportan información para la correcta operación del robot (p. ej., "Se necesita tecla de arranque").</li> <li>Los mensajes de observación se puede confirmar. No obstante, no se deben confirmar porque no detienen la unidad de control.</li> </ul>
	<p><b>Mensaje de espera</b></p> <ul style="list-style-type: none"> <li>Los mensajes de espera indican el suceso al que está esperando la unidad de control (estado, señal o tiempo).</li> <li>Los mensajes de espera se pueden cancelar manualmente pulsando el botón "Simular".</li> </ul>

**AVISO**

La instrucción "Simular" sólo se debe usar si no hay posibilidad alguna de colisión u otro peligro.

	<p><b>Mensaje de diálogo</b></p> <ul style="list-style-type: none"> <li>Los mensajes de diálogo se usan como comunicación o consulta directa con el operador.</li> <li>Aparece una ventana con botones con las distintas opciones disponibles como respuesta.</li> </ul>
--	--



Con "OK" puede confirmarse un mensaje (que pueda confirmarse). Con "Todo OK" pueden confirmarse todos los mensajes (que puedan confirmarse).

**Efecto de los mensajes**

Los mensajes influyen la funcionalidad del robot. Un mensaje de confirmación siempre provoca que el robot pare o no arranque. Primero debe confirmarse el mensaje para que se pueda mover el robot.

El comando "OK" (confirmar) exige al operador que reaccione activamente ante el mensaje.

**Consejos para actuar ante los mensajes:**

- Leer el mensaje detenidamente.
- Primero leer los mensajes más antiguos. El mensaje nuevo podría ser solo una consecuencia de un mensaje anterior.
- No pulsar simplemente "Todo OK".
- En especial después de haber efectuado el arranque: Revisar los mensajes. Visualizar todos los mensajes. Pulsando en la ventana de mensajes se expande la lista de mensajes.

**Tratamiento de los mensajes**

Los mensajes siempre aparecen con fecha y hora para poder saber el punto exacto de la incidencia.



Fig. 2-2: Confirmar mensajes

### **Procedimiento para examinar y confirmar mensajes.**

1. Tocar la ventana de mensajes para ampliar la lista de mensajes.
2. Confirmar:
  - Confirmar cada mensaje pulsando "OK".
  - Alternativa: Confirmar todos los mensajes pulsando "Todo OK".
3. Volviendo a tocar el mensaje situado más arriba o tocando sobre la "X" del extremo izquierdo de la pantalla se vuelve cerrar la lista de mensajes.

## **2.3 Seleccionar y ajustar el modo de servicio**

### **Modos de servicio de un robot KUKA**

- T1 (Manual Velocidad reducida)
  - Para el modo de prueba, programación y programación por aprendizaje
  - Velocidad en el modo de programación máx. 250 mm/s
  - Velocidad en el modo manual máx. 250 mm/s
- T2 (Manual Velocidad alta)
  - Para el modo de prueba
  - ¡Velocidad en el modo de programación según la velocidad programada!
  - Modo manual: no es posible
- AUT (Automático)
  - Para robots industriales sin unidad de control superior
  - ¡Velocidad en el modo de programación según la velocidad programada!
  - No es posible el desplazamiento manual mediante teclas de desplazamiento o Space-Mouse
- AUT EXT (Automático externo)
  - Para robots industriales con unidad de control superior (PLC)
  - ¡Velocidad en el modo de programación según la velocidad programada!
  - Modo manual: no es posible

### **Advertencias de seguridad de los modos de servicio**

#### **Servicio manual T1 y T2**

El servicio manual sirve para realizar los trabajos de ajuste. Se consideran trabajos de ajuste todos los trabajos que deban llevarse a cabo en el robot para poder ser operado en servicio automático. Entre ellos se encuentran:

- Aprendizaje / programación
- Ejecutar programa en modo tecleado (comprobación / verificación)

Los programas nuevos o modificados siempre se deben probar primero en el modo de servicio **Manual Velocidad reducida (T1)**.

En el modo de servicio **Manual Velocidad reducida (T1)**:

- La protección del operario (puerta de protección) no se controla.
- Se debe reducir al mínimo el número de personas dentro de la zona delimitada por los dispositivos de seguridad.

Si es imprescindible que varias personas permanezcan dentro de la zona delimitada por los dispositivos de seguridad, se debe tener en cuenta lo siguiente:

- Todas las personas deben tener un contacto visual sin obstáculos con el sistema de robot.
- Siempre debe existir contacto visual entre todas las personas implicadas.

- El operario debe situarse en una posición desde la cual pueda visualizar la zona de peligro y, así, poder evitar posibles peligros.

En el modo de servicio **Manual Velocidad alta (T2)**:

- La protección del operario (puerta de protección) no se controla.



En las celdas de formación de la escuela KUKA, sin embargo, la puerta de protección se controla y debe estar cerrada, a diferencia del estándar.

- Este modo de servicio sólo puede utilizarse cuando se requiere la aplicación de una prueba con velocidad más elevada que la del servicio Manual velocidad reducida.
- Este modo de servicio no es posible el modo de aprendizaje.
- Antes de iniciar la prueba, el operario debe asegurarse de que los dispositivos de validación están en condiciones de funcionamiento.
- Al ejecutar los programas en el servicio T2 se alcanza la velocidad programada.
- El operario, así como otras personas, deben colocarse fuera de la zona de peligro.

#### Modos de servicio Automático y Automático externo

- Todos los dispositivos de seguridad y protección deben estar debidamente montados y en condiciones de funcionamiento.
- Todas las personas se encuentran fuera de la zona delimitada por dispositivos de protección.

#### Reacciones de parada

El robot industrial tiene reacciones de parada debido a operaciones realizadas o como reacción ante controles y mensajes de error. Las siguientes tablas muestran las reacciones de parada en función del modo de servicio seleccionado.

Causa	T1, T2	AUT, AUT EXT
Soltar la tecla de arranque	STOP 2	-
Pulsar la tecla STOP	STOP 2	
Accionamientos DESC.	STOP 1	
La entrada "Validación de marcha" se desactiva	STOP 2	
Desconectar la unidad de control del robot (corte de corriente)	STOP 0	
Error interno en la sección de la unidad de control del robot sin función de seguridad	STOP 0 o STOP 1 (depende de la causa del error)	
Cambiar el modo de servicio durante el servicio	Parada de seguridad 2	
Abrir la puerta de protección (protección del operario)	-	Parada de seguridad 1
Soltar el pulsador de validación	Parada de seguridad 2	-
Pulsar el pulsador de validación o error	Parada de seguridad 1	-

Causa	T1, T2	AUT, AUT EXT
Pulsar PARADA DE EMERGENCIA		Parada de seguridad 1
Error en el control de seguridad o en los periféricos del control de seguridad		Parada de seguridad 0

Término	Descripción
Parada de servicio segura	<p>La parada de servicio segura es un control de parada. No detiene el movimiento del robot, sino que controla si los ejes del robot se detienen. En caso de que se muevan durante la parada de servicio segura, se activa una parada de seguridad STOP 0.</p> <p>La parada de servicio segura también se puede accionar desde el exterior.</p> <p>Cuando se activa una parada de servicio segura, la unidad de control del robot lo comunica a una unidad de control superior a través de una salida segura (X13, SIB extended) o a través de un protocolo de bus de campo seguro. Esta salida también se establece si en el momento en el que se acciona la parada de servicio segura no todos los ejes están parados y, por tanto, se activa una parada de seguridad STOP 0.</p>
Parada de seguridad STOP 0	<p>Una parada que se acciona y ejecuta desde el control de seguridad. El control de seguridad desconecta de inmediato los accionamientos y la alimentación de tensión de los frenos.</p> <p><b>Indicación:</b> en la presente documentación, esta parada recibe el nombre de parada de seguridad 0.</p>
Parada de seguridad STOP 1	<p>Una parada que se acciona y controla desde el control de seguridad. El procedimiento de frenado se ejecuta con un componente de la unidad de control del robot no destinado a la seguridad y controlado a través del control de seguridad. En el momento en que el manipulador se para, el control de seguridad desconecta los accionamientos y la alimentación de tensión de los frenos.</p> <p>Cuando se acciona una parada de seguridad STOP 1, la unidad de control del robot establece una salida a través del bus de campo.</p> <p>La parada de seguridad STOP 1 también se puede accionar de forma externa.</p> <p><b>Indicación:</b> en la presente documentación, esta parada recibe el nombre de parada de seguridad 1.</p>
Parada de seguridad STOP 2	<p>Una parada que se acciona y controla desde el control de seguridad. El procedimiento de frenado se ejecuta con un componente de la unidad de control del robot no destinado a la seguridad y controlado a través del control de seguridad. Los accionamientos se mantienen conectados y los frenos abiertos. En el momento en que el manipulador se para, se activa una parada de servicio segura.</p> <p>Cuando se acciona una parada de seguridad STOP 2, la unidad de control del robot establece una salida a través del bus de campo.</p> <p>La parada de seguridad STOP 2 también se puede accionar de forma externa.</p> <p><b>Indicación:</b> en la presente documentación, esta parada recibe el nombre de parada de seguridad 2.</p>

Término	Descripción
Categoría de parada 0	Los accionamientos se desconectan de inmediato y se activan los frenos. El manipulador y los ejes adicionales (opcional) frenan cerca de la trayectoria.  <b>Indicación:</b> esta categoría de parada recibe en el documento el nombre de STOP 0.
Categoría de parada 1	Transcurrido 1 s se desconectan los accionamientos y se activan los frenos.  El manipulador y los ejes adicionales (opcionales) frenan sobre la trayectoria.  <b>Indicación:</b> esta categoría de parada recibe en el documento el nombre de STOP 1.
Categoría de parada 2	Los accionamientos no se desconectan y no se activan los frenos. El manipulador y los ejes adicionales (opcional) frenan con una rampa de frenado sobre la trayectoria.  <b>Indicación:</b> esta categoría de parada recibe en el documento el nombre de STOP 2.

### Procedimiento



Si durante el servicio se cambia el modo de servicio, los accionamientos son inmediatamente desconectados. El robot industrial se para con una parada de seguridad 2

1. Mover el interruptor del KCP para el gestor de conexiones. Se visualiza el gestor de conexiones.

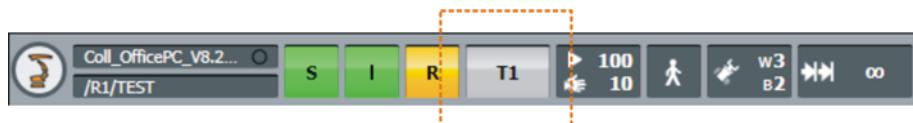


2. Seleccionar el modo de servicio.



3. Volver a colocar el interruptor para el gestor de conexiones en su posición original.

El modo de servicio seleccionado se muestra en la barra de estado del smartPAD.



## 2.4 Desenchufar el smartPAD

### Descripción para desenchufar el smartPAD

- El smartPAD puede desenchufarse aunque esté funcionando la unidad de control del robot.
- En todo momento puede enchufarse un smartPAD.
- El smartPAD conectado asume el modo de servicio actual de la unidad de control del robot.
- Para la conexión, la variante de smartPAD (estado del firmware) no es relevante, ya que se actualiza automáticamente.
- Transcurridos 30 s después de conectarlo, la PARADA DE EMERGENCIA y el pulsador de validación vuelven a estar en condiciones de funcionamiento.
- La smartHMI (interfaz de usuario) vuelve a visualizarse automáticamente (no tarda más de 15 s).

### Función para desenchufar el smartPAD

- 

**⚠ ADVERTENCIA** Si el smartPAD está desenchufado, la instalación no se puede desconectar a través del pulsador de PARADA DE EMERGENCIA del smartPAD. Por esta razón, la unidad de control del robot debe tener conectada una PARADA DE EMERGENCIA externa.

- 

**⚠ ADVERTENCIA** El explotador debe encargarse de retirar inmediatamente de la instalación el smartPAD desenchufado y mantenerlo fuera del alcance del personal que está trabajando en el robot industrial. De este modo se consigue evitar cualquier confusión entre los dispositivos de PARADA DE EMERGENCIA efectivos y los no efectivos.

- 

**⚠ ADVERTENCIA** Si no se respeta esta medida, pueden producirse importantes daños materiales y personales o incluso la muerte.

- 

**⚠ ADVERTENCIA** El usuario que conecte un smartPAD a la unidad de control del robot, luego deberá esperar como mínimo 30 s hasta que la PARADA DE EMERGENCIA y el pulsador de validación vuelvan a estar en condiciones de funcionamiento. De esta manera se evita, p. ej., que otro usuario se encuentre en una situación de emergencia y la PARADA DE EMERGENCIA no esté activa.

### Procedimiento para desenchufar un smartPAD

#### Retirar:

1. Pulsar el botón para retirar del smartPAD.

En la smartHMI se visualiza un mensaje y un contador. El contador controliza 25 s. Durante este tiempo puede retirarse el smartPAD de la unidad de control del robot.

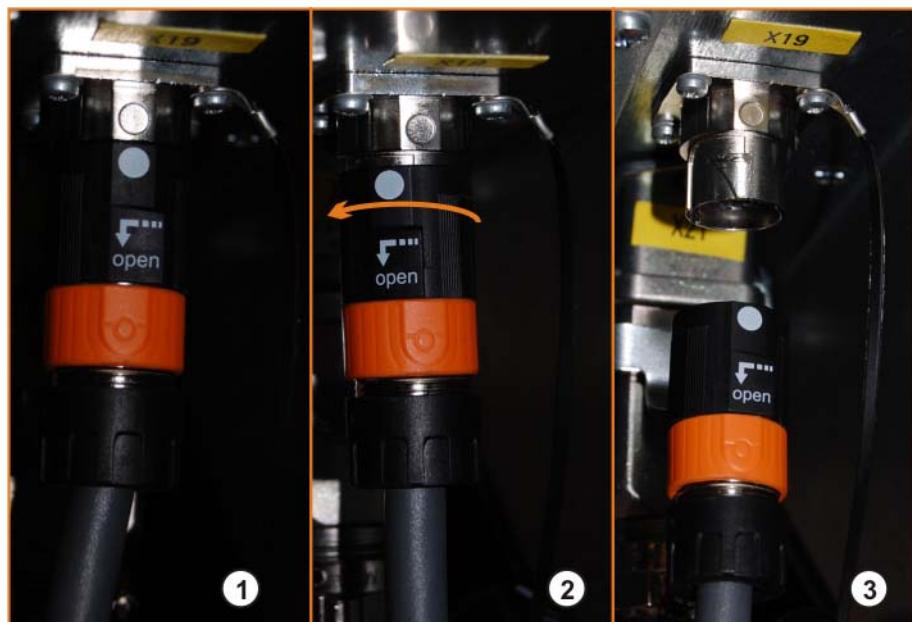


**Fig. 2-3: Desacoplar pulsador smartPAD**

**AVISO**

Si se extrae el smartPAD sin que corra el contador, se provoca una PARADA DE EMERGENCIA. La PARADA DE EMERGENCIA sólo puede anularse fijando de nuevo el smartPAD.

2. Abrir las puertas del armario de distribución (V)KR C4.
3. Desconectar el smartPAD de la unidad de control del robot.



**Fig. 2-4: Desenchufar smartPAD**

1	Conejero enchufado
2	Girar la parte superior negra unos 25° en la dirección de la flecha.
3	Retirar el conector hacia abajo.

4. Cerrar las puertas del armario de distribución (V)KR C4.

Si el contador llega hasta el final y no se retira el smartPAD, no pasa nada. El botón de enclavamiento se puede pulsar cuantas veces se quiera para visualizar el contador.

**Fijar:**

1. Asegurarse de que se vuelva a usar la misma variante de smartPAD.
2. Abrir las puertas del armario de distribución (V)KR C4.

3. Enchufar el conector del smartPAD

**AVISO**

Tener en cuenta la marca del casquillo y el conector del smartPAD.

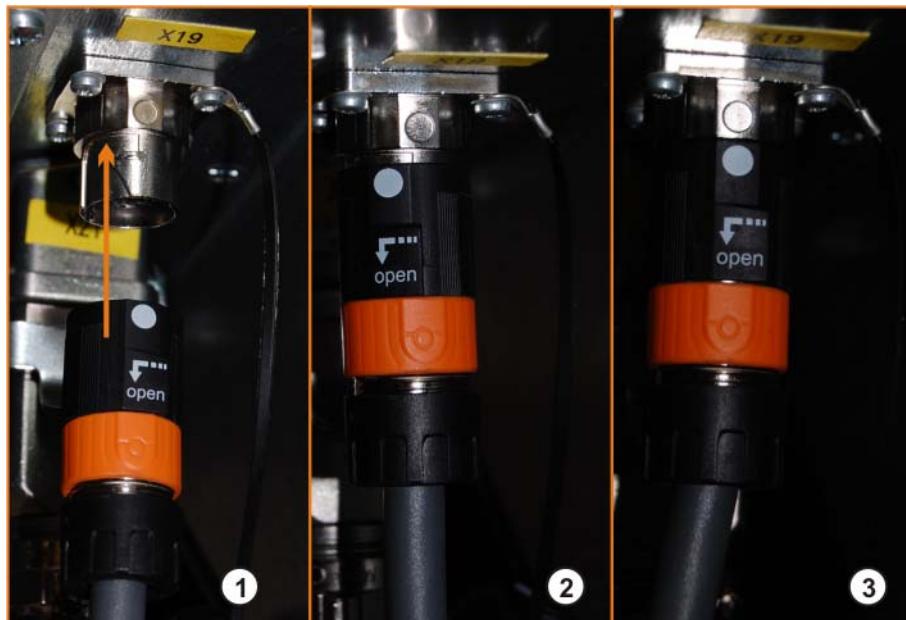


Fig. 2-5: Conexión del smartPAD

1	Conector desenchufado (tener en cuenta la marca).
2	Empujar el conector hacia arriba. La parte superior negra gira por sí sola unos 25° al empujar el conector hacia arriba.
3	El conector encaja por sí solo, es decir, las marcas quedan alineadas.

**⚠ ADVERTENCIA**

El usuario que fije un smartPAD a la unidad de control del robot, luego deberá esperar como mínimo 30 s hasta que la PARADA DE EMERGENCIA y el pulsador de validación vuelvan a estar en condiciones de funcionamiento. De esta manera se evita, p. ej., que otro usuario se encuentre en una situación de emergencia y la PARADA DE EMERGENCIA no esté activa.

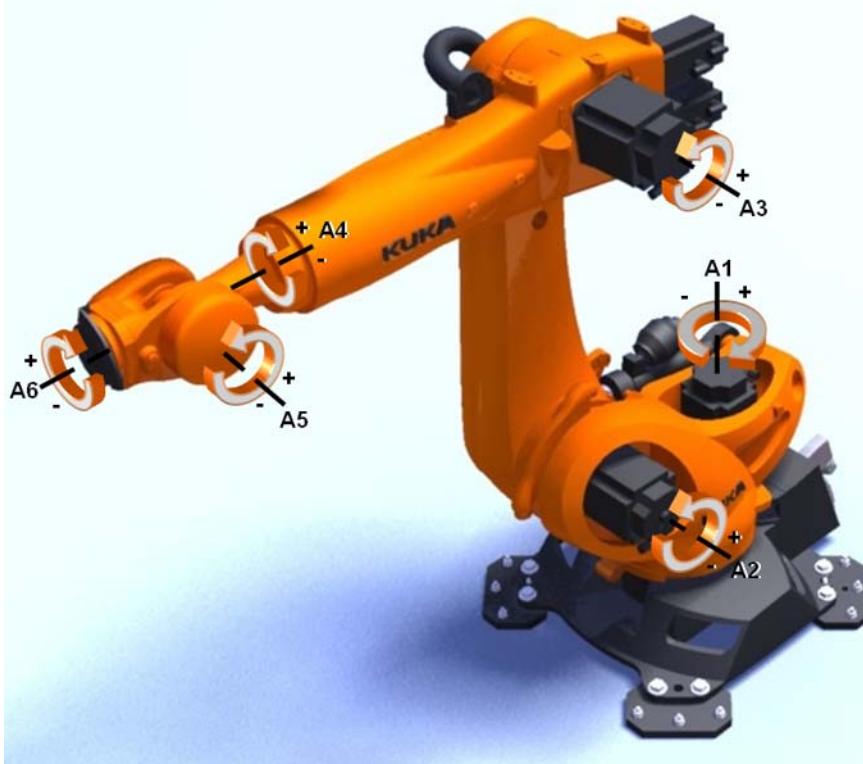
4. Cerrar las puertas del armario de distribución (V)KR C4.

## 2.5 Mover ejes del robot individualmente

**Descripción:**

**Movimiento**

**específico del eje**



**Fig. 2-6: Ejes de un robot KUKA**

### Movimiento de los ejes del robot

- Desplazar individualmente cada eje en dirección positiva y negativa.
- Para ello se utilizan las teclas de desplazamiento o el Space Mouse del KUKA smartPAD.
- La velocidad puede cambiarse (override manual: HOV).
- Los procesos manuales sólo se pueden efectuar en el modo de servicio T1.
- Debe estar pulsado el pulsador de validación.
  
- Al pulsar el pulsador de validación se activan los accionamientos. Cuando se dispone de la habilitación de accionamientos, los textos de las teclas de desplazamiento cambian a color verde. En el momento en que se acciona una tecla de desplazamiento o el Space Mouse, se inicia la regulación los ejes del robot y se ejecuta el movimiento deseado.
- Se puede elegir entre un movimiento continuo o un movimiento incremental. En la barra de estado se deberá seleccionar la medida de paso (valor del incremento). (>>> "Procedimiento" Página 32)

**Los siguientes mensajes ejercen influencia sobre el servicio manual:**

Mensaje	Causa	Solución
"Comandos activos bloqueados"	Existe un mensaje (de PARADA) o un estado que provoca el bloqueo de los comandos activos (p. ej., PARADA DE EMERGENCIA pulsada o accionamientos aún no preparados).	Desbloquear la parada de emergencia y/o confirmar los mensajes de la ventana de mensajes. Al pulsar el pulsador de validación se activan de inmediato los accionamientos.
"Interruptor de final de carrera de software A5"	El interruptor de final de carrera de software del eje mostrado (p. ej. A5) se ha aproximado en la dirección indicada (+ o -).	Desplazar el eje indicado en la dirección contraria.

## Advertencias de seguridad para movimientos manuales específicos de los ejes

### Modo de servicio

El modo manual del robot solo es posible en el modo de servicio T1 (Manual Velocidad reducida). En el modo T1, la velocidad de desplazamiento manual es de 250 mm/s como máximo. El modo de servicio se ajusta por medio del gestor de conexiones.

### Pulsador de validación

Para poder desplazar el robot se debe accionar un pulsador de validación. En el smartPAD se encuentran instalados tres pulsadores de validación. Los pulsadores de validación tienen tres posiciones:

- No pulsado
- Posición intermedia
- Pulsado a fondo (posición de pánico)

### Interruptores de final de carrera de software

El movimiento del robot está limitado por los valores máximos positivo y negativo de los interruptores de final de carrera de software, incluso en los movimientos manuales específicos de los ejes.

#### AVISO

Si en la ventana de mensajes aparece "Realizar ajuste", significa que también se pueden traspasar estos límites. En este caso se debe tener en cuenta que el sistema del robot puede sufrir daños.

## Procedimiento: Realizar movimiento específico del eje

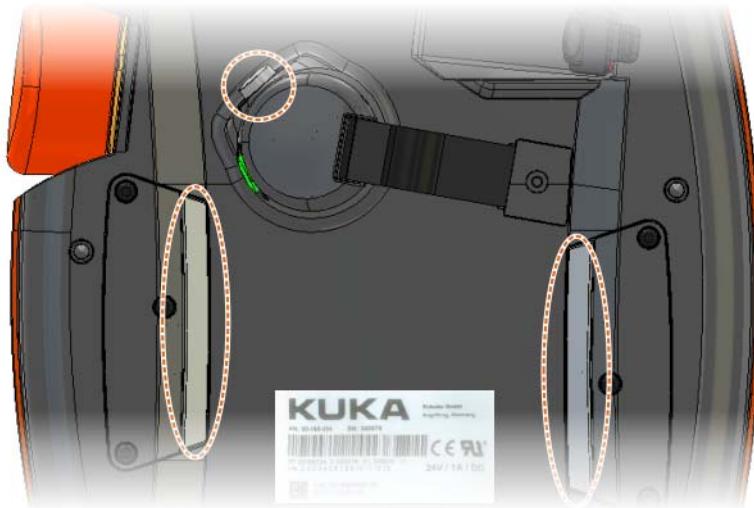
1. Como alternativa a las teclas de desplazamiento, seleccionar **Ejes**.



2. Ajustar el override manual



3. Pulsar y mantener pulsado el pulsador de validación en la posición intermedia.



4. Junto a las teclas de desplazamiento se muestran los ejes A1 a A6. Pulsar la tecla de desplazamiento + ó - para mover un eje en dirección positiva o negativa.



### Movimiento manual incremental

#### Descripción

El movimiento manual incremental permite que el robot se mueva en una distancia definida como, por ej., 10 mm o 3°. A continuación el robot se detiene automáticamente.

El desplazamiento manual incremental puede activarse durante el movimiento con las teclas de desplazamiento. En el desplazamiento con el Space Mouse, no es posible aplicar el desplazamiento manual incremental.

Ámbitos de aplicación:

- Posicionamiento de puntos a distancias iguales.
- Movimiento de separación de una posición en una distancia definida, por ej., en caso de fallo.
- Ajuste con reloj comparador.

#### Condición previa

- El tipo de desplazamiento "Teclas de desplazamiento" está activo.
- Modo de servicio T1

#### Procedimiento

1. Seleccionar el valor del incremento en la barra de estado:



**Fig. 2-7: Movimiento manual incremental**

- Mover el robot con las teclas de desplazamiento. Este se puede desplazar de forma cartesiana o específica del eje.

El robot detiene la marcha cuando se ha alcanzado el valor del incremento ajustado.

Cuando se interrumpe el movimiento del robot como, por ej., al soltar el pulsador de validación, en el próximo movimiento no se continúa con el incremento interrumpido, sino que se comienza con un incremento nuevo.

Se puede elegir entre las siguientes opciones:

Ajuste	Descripción
<b>Continuo</b>	El movimiento manual incremental está desconectado.
<b>100mm/10°</b>	1 incremento = 100 mm o 10°
<b>10mm/3°</b>	1 incremento = 10 mm o 3°
<b>1mm/1°</b>	1 incremento = 1 mm o 1°
<b>0,1mm/0,005°</b>	1 incremento = 0,1 mm o 0,005°

Incrementos en mm:

- Válido en desplazamientos cartesianos en direcciones X, Y y Z.

Incrementos en grados:

- Válido en desplazamientos cartesianos en direcciones A, B o C.
- Válido para movimientos específicos del eje.



**Mover el robot sin la unidad de control en casos de emergencia**

**Fig. 2-8: Dispositivo de liberación**

#### Descripción

- El dispositivo de liberación permite mover el robot mecánicamente en caso de accidente o avería.
- El dispositivo de liberación puede utilizarse para los motores de accionamiento de los ejes principales y, dependiendo de la variante del robot, también para los motores de accionamiento del eje de la muñeca.

- Sólo se debe utilizar en situaciones excepcionales y casos de emergencia como, p. ej., para liberar personas.
- Cuando se vaya a usar el dispositivo de liberación, se debe garantizar el correcto funcionamiento de los frenos.
  - Para ello, se debe realizar una prueba de frenos. Si esta resultara negativa, se deberán sustituir los motores.
  - Si la prueba de frenos no está disponible en la unidad de control o no se puede realizar, se deberán sustituir los motores afectados.

**⚠ ADVERTENCIA**

Durante el servicio, los motores alcanzan temperaturas que pueden causar quemaduras en la piel. Debe evitarse cualquier contacto. Deben aplicarse medidas de protección adecuadas como, p. ej., llevar guantes de protección.

#### Procedimiento

1. Desconectar la unidad de control del robot y asegurarla contra una reconnexión indebida (p. ej., con un candado).
2. Quitar la tapa protectora del motor.
3. Posicionar el dispositivo de liberación en el motor correspondiente y mover el eje en la dirección deseada.

Opcionalmente se puede adquirir una identificación de las direcciones sobre los motores por medio de flechas. Debe vencerse la resistencia mecánica del freno por motor y, en caso necesario, también las posibles cargas adicionales de los ejes.

#### Ejemplo con el motor del eje 2:

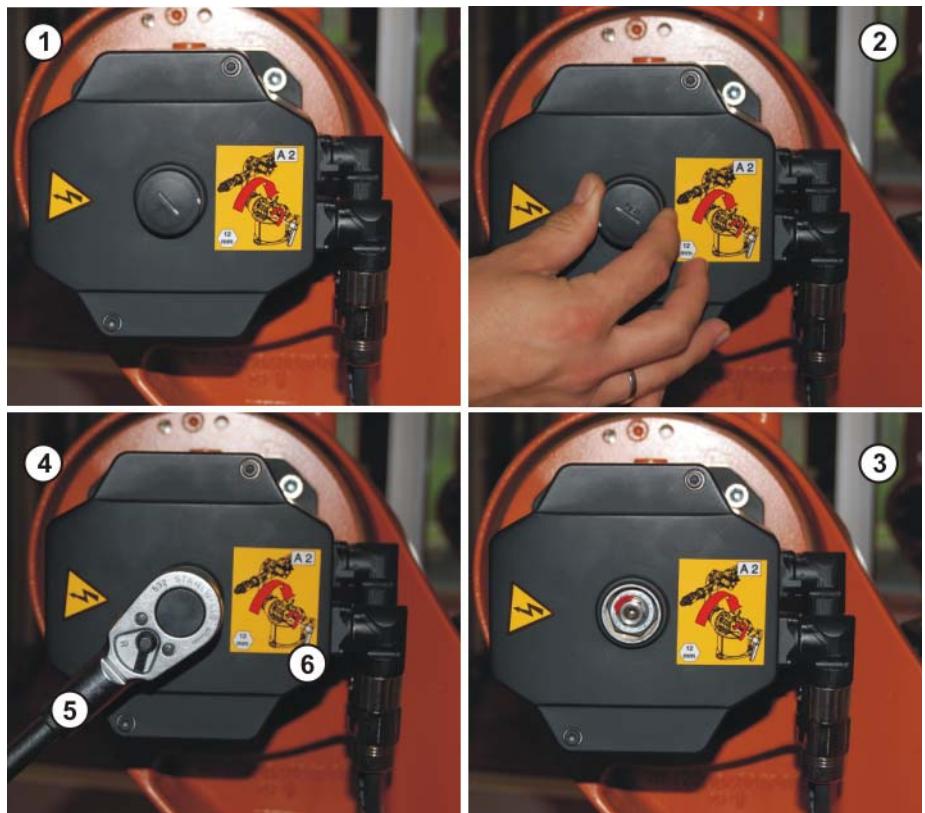


Fig. 2-9: Procedimiento con el dispositivo de liberación

Pos.	Descripción
1	Motor A2 con tapa protectora cerrada
2	Abrir la tapa protectora del motor A2
3	Motor A2 con tapa protectora retirada

Pos.	Descripción
4	Colocar el dispositivo de liberación sobre el motor A2
5	Dispositivo de liberación
6	Placa (opcional) con descripción de la dirección de giro

**⚠ ATENCIÓN**

Al desplazar un eje con el dispositivo de liberación, el freno por motor puede sufrir daños. Pueden producirse daños personales y materiales. Después de utilizar el dispositivo de liberación debe sustituirse el motor afectado.



En las instrucciones de servicio y de montaje del robot se encuentra información adicional.

## 2.5.1 Ejercicio: Operación y desplazamiento manual específico del eje

### Objetivo del ejercicio

Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Conectar/desconectar la unidad de control del robot
- Servicio básico del robot con el smartPad
- Desplazamiento manual específico del eje del robot con las teclas de desplazamiento y el Space Mouse
- Interpretar y eliminar los primeros mensajes de sistema sencillos

### Requisitos

Los siguientes requisitos son imprescindibles para superar este ejercicio con éxito:

- Participación en las instrucciones de seguridad



#### Aviso

Antes de comenzar el ejercicio se debe participar en las instrucciones de seguridad con documentación correspondiente.

- Conocimientos teóricos acerca del servicio general de un sistema de robot industrial KUKA
- Conocimientos teóricos del desplazamiento manual específico del eje

### Formulación de tarea

Ejecutar las siguientes tareas:

1. Conectar el armario de control y esperar hasta terminar la fase de arranque.
2. Desbloquear y confirmar la parada de EMERGENCIA.
3. Asegurarse de que esté ajustado el modo de servicio T1.
4. Activar el desplazamiento manual específico del eje.
5. Desplazar el robot de modo específico del eje con diversos ajustes del override manual (HOV) mediante las teclas de desplazamiento manual y el Space Mouse.
6. Determinar el área de desplazamiento de cada uno de los ejes, tener en cuenta obstáculos existentes, como p. ej. una mesa o un depósito de cubos con herramienta fija (análisis de accesibilidad).
7. Tener en cuenta la ventana de mensajes al alcanzar los interruptores de final de carrera de software.

## 2.6 Sistemas de coordenadas en relación con los robots

A la hora de programar, poner en servicio y operar con robots industriales, los sistemas de coordenadas juegan un papel importante. En la unidad de control del robot se encuentran definidos los siguientes sistemas de coordenadas:

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

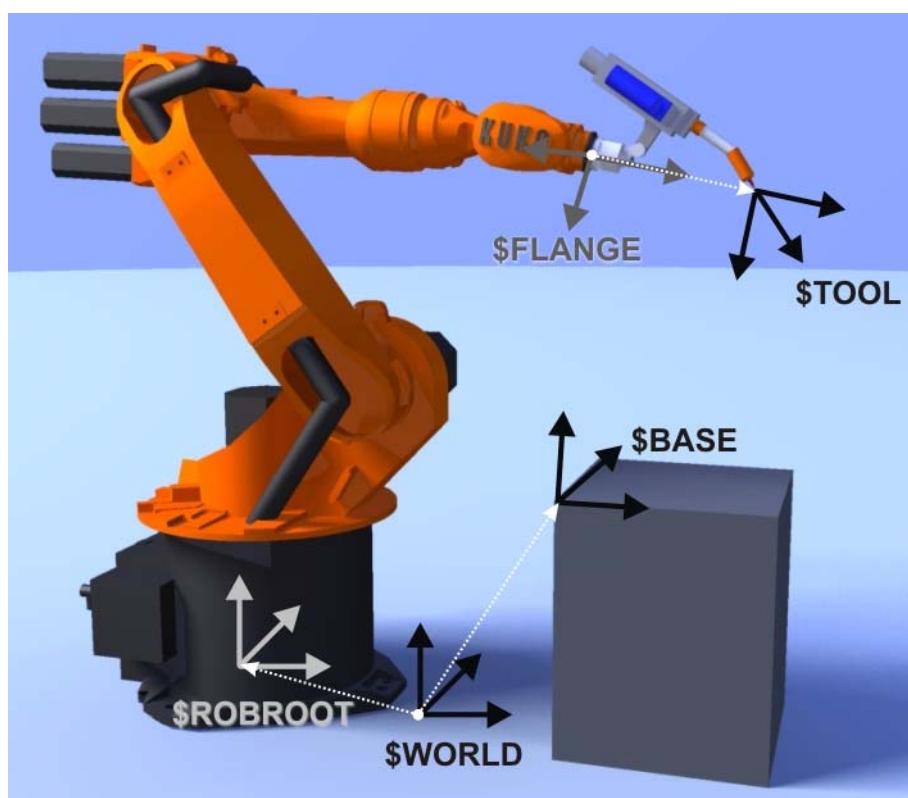


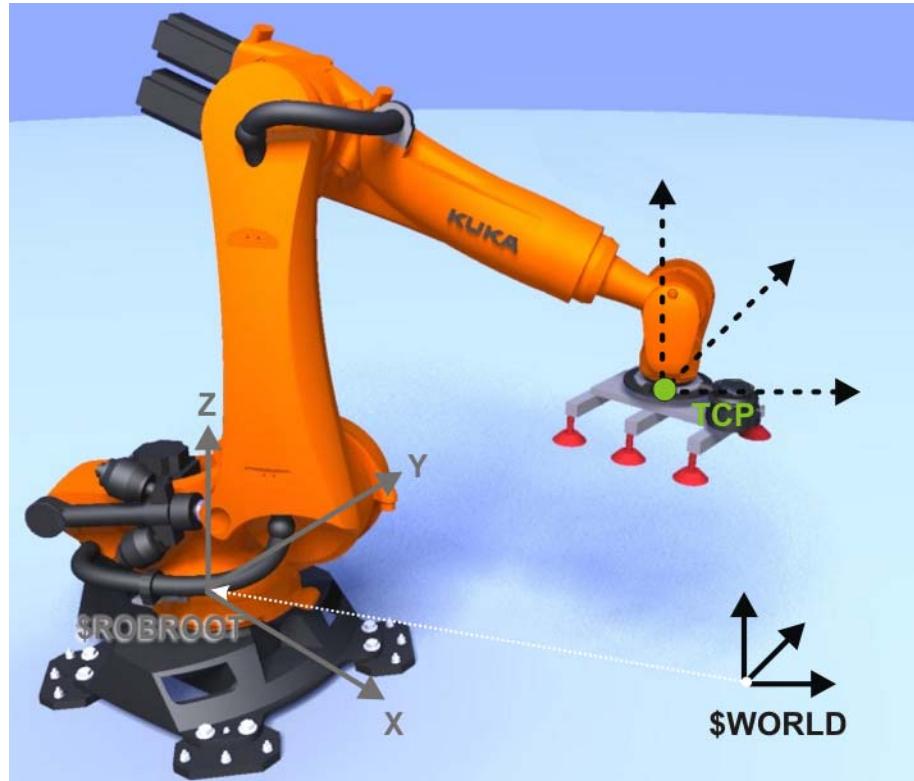
Fig. 2-10: Sistemas de coordenadas en el robot KUKA

- **ROBROOT**
  - Se encuentra anclado en el pie del robot.
  - Es el origen real del robot.
  - Es el punto de referencia para el sistema de coordenadas WORLD.
- **WORLD**
  - En el estado de suministro, coincide con el sistema de coordenadas ROBROOT.
  - Se puede "sacar" del pie del robot.
  - Describe la posición del sistema de coordenadas WORLD en relación con el sistema de coordenadas ROBROOT.
  - Se usa, entre otros, en sistemas de robot de montaje en pared o techo.
- **BASE**

- Es un sistema de coordenadas de libre definición específico del cliente
- Describe la posición de la base con relación a WORLD.
- Se usa para la medición de piezas y útiles.
- **FLANGE**
  - El sistema de coordenadas FLANGE se encuentra anclado en la brida del robot.
  - El origen es el centro de la brida del robot.
  - Es el punto de referencia para el sistema de coordenadas TOOL.
- **TOOL**
  - Es un sistema de coordenadas de libre definición específico del cliente.
  - El origen del sistema de coordenadas TOOL se denomina **TCP** - Tool Center Point.
  - Se usa para la medición de herramientas.

## 2.7 Mover el robot en el sistema de coordenadas universales

**Movimiento del sistema de coordenadas universales**



**Fig. 2-11: Principio del proceso manual del sistema de coordenadas de herramienta**

- La herramienta del robot puede moverse de acuerdo con las direcciones del sistema de coordenadas universales.
- **Todos** los ejes del robot se mueven.
- Para ello se utilizan las teclas de desplazamiento o el Space Mouse del KUKA smartPAD.
- En el ajuste estándar el sistema de coordenadas universales descansa sobre el pie del robot (Robroot).
- La velocidad puede cambiarse (override manual: HOV)
- Los procesos manuales sólo se pueden efectuar en el modo de servicio T1.

- El pulsador de validación debe estar pulsado.

#### Space Mouse

- El Space Mouse permite mover de forma intuitiva el robot y resulta la elección ideal para los procesos manuales en el sistema de coordenadas universales.
- La posición del puntero y el grado de libertad se pueden cambiar.

**Principio del proceso manual en el sistema de coordenadas universales**

#### Un robot se puede mover de dos maneras distintas en un sistema de coordenadas:

- Traslatoria (rectilínea) a lo largo de la dirección de orientación del sistema de coordenadas: X, Y, Z.
- Rotacional (giratoria / viratoria) en la dirección de orientación del sistema de coordenadas: ángulos A, B y C.

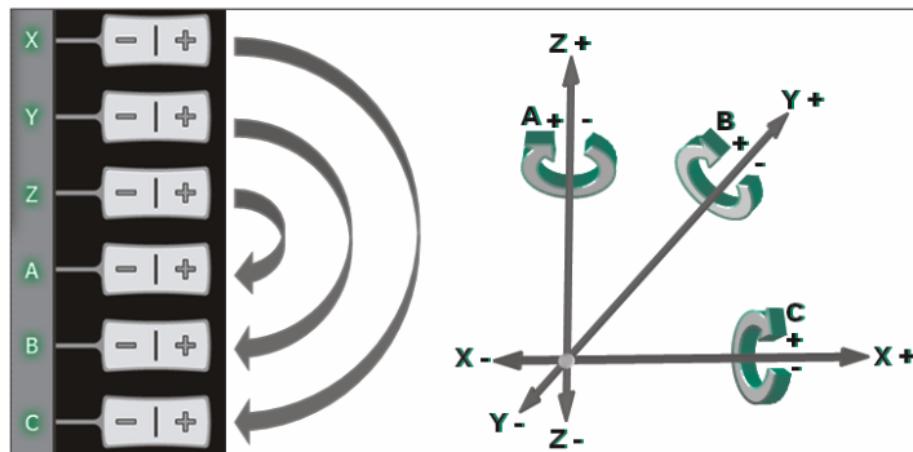


Fig. 2-12: Sistema de coordenadas cartesiano

En un comando de desplazamiento (p. ej., al pulsar una tecla de desplazamiento), el control primero calcula una distancia. El punto de inicio del recorrido es el punto de referencia de la herramienta (TCP). La dirección del recorrido lo fija el sistema de coordenadas universales. El control entonces regula todos los ejes de modo que la herramienta se conduce por este recorrido (traslación) o se gira (rotación) en este recorrido.

#### Ventajas de usar el sistema de coordenadas universales:

- El movimiento del robot siempre es predecible.
- Los movimientos siempre son únicos ya que el origen y la dirección de las coordenadas siempre se conocen de antemano.
- El sistema de coordenadas de la herramienta siempre se puede utilizar con el robot ajustado.
- El Space Mouse permite un manejo intuitivo del equipo.

#### Uso de Space-Mouse

- Todos los tipos de movimiento se pueden realizar con el Space Mouse:
  - Translatorio: pulsando y arrastrando el Space Mouse

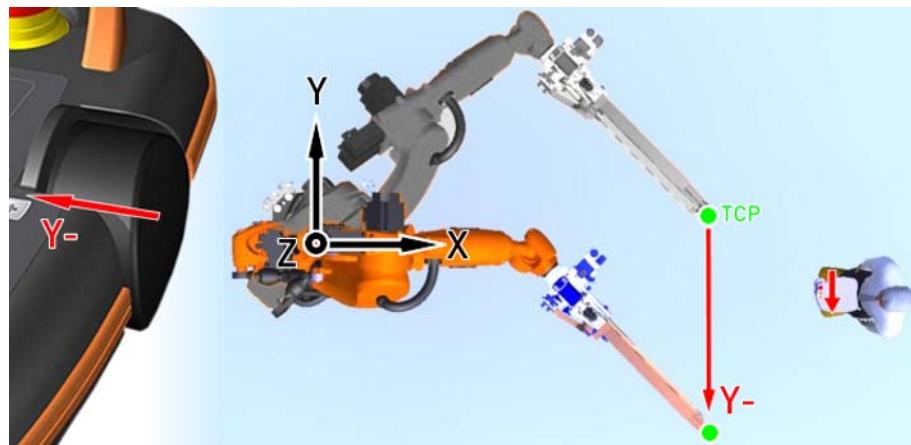


Fig. 2-13: Ejemplo: Movimiento a la izquierda

- Rotacional: girando y virando el Space Mouse

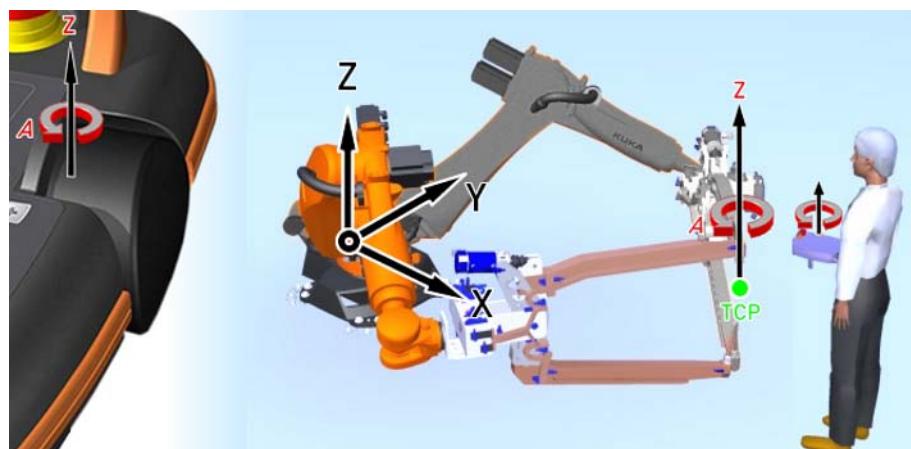


Fig. 2-14: Ejemplo: Movimiento rotacional en Z: Ángulo A

- Dependiendo de la posición de la persona y del robot puede adaptarse la posición del Space Mouse.

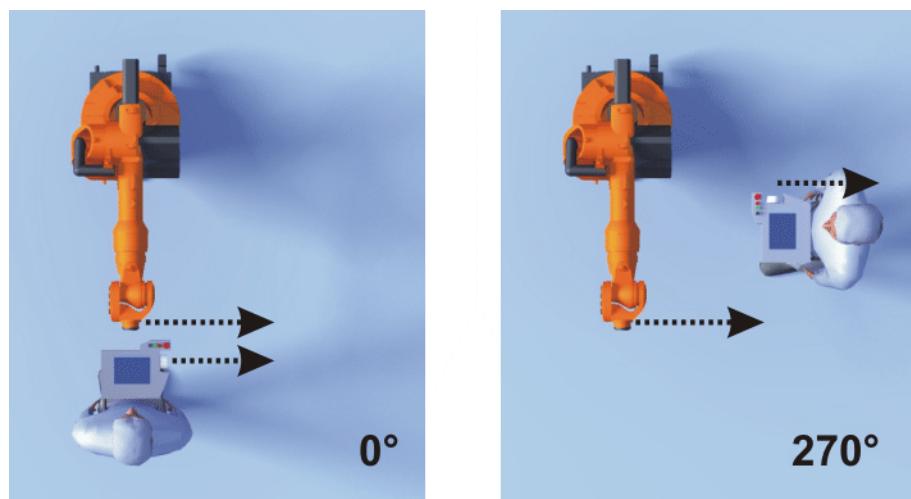


Fig. 2-15: Space Mouse: 0° y 270°

## Ejecutar un movimiento translatorio (Mundo)

- Ajustar la posición KCP desplazando el regulador (1)



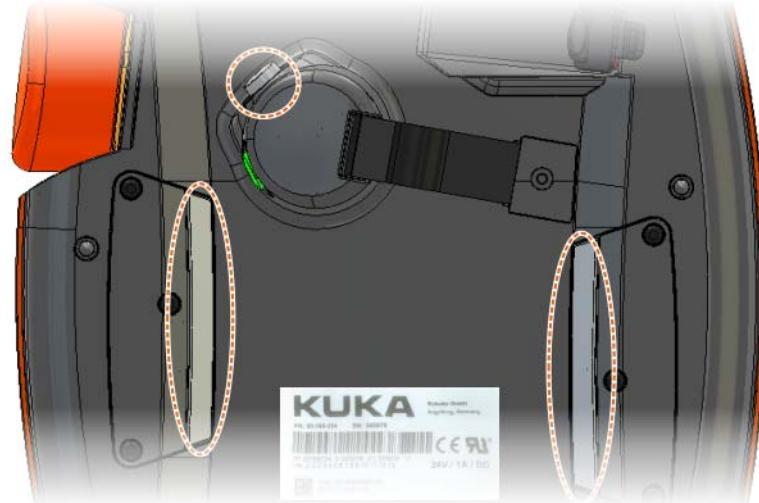
- Como alternativa al Space Mouse, seleccionar **Mundo**.



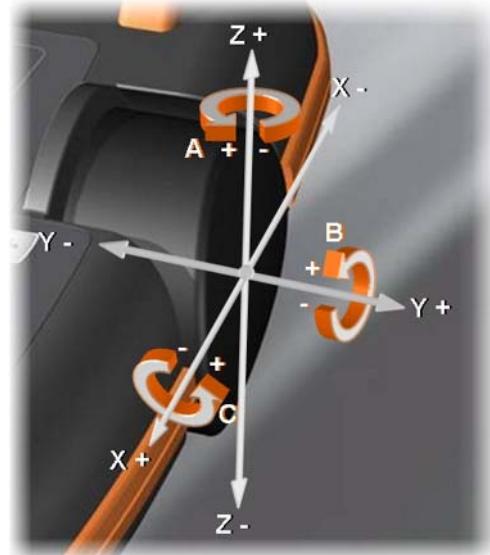
- Ajustar el override manual



4. Pulsar el pulsador de validación en la posición intermedia y mantenerlo pulsado.



5. Con el Space Mouse mover el robot en la dirección correspondiente.



6. Como alternativa también se pueden utilizar las teclas de desplazamiento



## 2.7.1 Ejercicio: Operación y desplazamiento manual en el sistema de coordenadas universales

**Objetivo del ejercicio** Despu s de completar correctamente este ejercicio, se dispondr  de la competencia necesaria para efectuar las siguientes tareas:

- Conectar/desconectar la unidad de control del robot
- Servicio b sico del robot con el smartPad
- Desplazamiento manual del robot en el sistema de coordenadas universales, con las teclas de desplazamiento y el Space Mouse
- Interpretar y eliminar los primeros mensajes de sistema sencillos

**Requisitos** Los siguientes requisitos son imprescindibles para superar este ejercicio con éxito:

- Participaci n en las instrucciones de seguridad



### Aviso

Antes de comenzar el ejercicio se debe participar en las instrucciones de seguridad con documentaci n correspondiente.

- Conocimientos te ricos acerca del servicio general de un sistema de robot industrial KUKA
- Conocimientos te ricos del desplazamiento en el sistema de coordenadas universales

- Formulaci n de tarea**
1. Desbloquear y confirmar la parada de EMERGENCIA.
  2. Asegurarse de que est  ajustado el modo de servicio T1.
  3. Activar el sistema de coordenadas universales
  4. Sacar un cubo del cargador y colocarlo sobre la mesa.
  5. Intentar orientar la garra del robot sobre el cubo.
  6. Cerrar la garra. Al cerrar la garra, el cubo no se debe mover.
  7. Desplazar el cubo de la mesa y depositarlo en otro lugar cualquiera.
  8. Cambiar entre teclas de desplazamiento y manejo del rat n.

### Lo que se debe saber tras el ejercicio:

1. ¿C mo se pueden confirmar los mensajes?

.....  
.....

2. ¿Cu l de los pictogramas representa el sistema de coordenadas universal?



3. ¿Cu l es la denominaci n del ajuste de velocidad para el desplazamiento manual?

.....  
.....

4. ¿Cu les son los modos de servicio?



## 3 Puesta en servicio del robot

### 3.1 Resumen

**Esta unidad incluye los siguientes contenidos:**

- Modo de puesta en servicio
- Ajuste de robot
- Medición de herramienta
- Medición de base
- Medición de una herramienta fija
- Desenchufar el smartPAD

### 3.2 Modo de puesta en servicio

<b>Descripción</b>	<ul style="list-style-type: none"> <li>■ No es posible desplazar el robot si, por ejemplo, durante una nueva puesta en servicio todavía no se dispone de periféricos de seguridad (p. ej., una PARADA DE EMERGENCIA externa).</li> <li>■ Esto se puede solucionar con el <i>modo de puesta en servicio</i>, el cual permite el movimiento del robot en la velocidad reducida T1.</li> <li>■ Gracias a ello, se pueden realizar tareas de puesta en servicio, como p. ej., el ajuste del robot.</li> <li>■ El robot industrial se puede poner en modo de puesta en servicio a través de una opción de menú de la interfaz de usuario smartHMI.</li> <li>■ En el caso de una conexión activa con un sistema de seguridad, se impide o finaliza el modo de puesta en servicio.</li> </ul>
--------------------	--

Al utilizar el modo de puesta en servicio, todos los dispositivos de seguridad externos se encuentran fuera de servicio.

#### **Possibles peligros y riesgos durante la utilización del modo de puesta en servicio:**

- Una persona entra en la zona de peligro del manipulador.
- Una persona no autorizada mueve el manipulador.
- En caso de peligro, se acciona un dispositivo externo de PARADA DE EMERGENCIA inactivo y el manipulador no se desconecta.

#### **Medidas adicionales para la prevención de riesgos en el modo de puesta en servicio:**

- No cubrir los dispositivos de PARADA DE EMERGENCIA que no estén operativos o indicar mediante un cartel de advertencia qué dispositivo de PARADA DE EMERGENCIA no está operativo.
- Si no se dispone de valla de seguridad, se debe evitar el acceso de personas a la zona de peligro del manipulador aplicando otras medidas, p. ej., con una cinta.
- La utilización del modo de puesta en servicio se debe limitar o evitar en la medida de lo posible aplicando medidas organizativas.
- El personal del servicio técnico debe asegurarse de que no hay nadie dentro o en las inmediaciones de la zona de peligro del manipulador mientras los dispositivos de seguridad estén fuera de servicio.

#### **Utilización del modo puesta en servicio conforme a los fines previstos:**

- Únicamente el personal del servicio técnico debidamente formado en materia de seguridad puede utilizar el modo de puesta en servicio.
- Para delimitar un error (error en los periféricos).

- Para la puesta en servicio

**Uso incorrecto**

- Todas las utilizaciones que difieran de la utilización conforme a los fines previstos se consideran incorrectas. Se incluye, p. ej., el uso por parte de otro grupo de personas.
- KUKA Roboter GmbH no se hace responsable de los daños que pudieran ocasionarse. El riesgo lo asume única y exclusivamente el explotador.

**Requisito previo**
**Requisito para la configuración PROFIsafe:**

- PROFIsafe está activado a través de una configuración de seguridad.
- Es necesario el grupo de usuarios experto.
- Ruta:** botón KUKA > Configuración > Grupo de usuarios
- La unidad de control del robot impide o finaliza el modo de puesta en servicio, si se establece o existe una conexión con un PLC de seguridad superior.

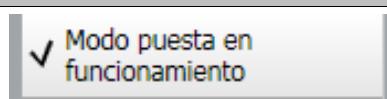
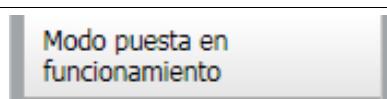
**Requisito para la configuración X11 / SIB**

- Es necesario el grupo de usuarios experto.
- Ruta:** botón KUKA > Configuración > Grupo de usuarios
- KR C4: X311 debe estar conectado como conector puente en la CCU.  
KR C4 compact: no es necesario, conector puente no disponible.
- **System Software 8.2 y anterior:**  
El modo de puesta en servicio solo es posible si todas las señales de entrada de la interfaz de seguridad X11 tienen el estado "cero lógico". De lo contrario, la unidad de control del robot impide o finaliza el modo de puesta en servicio.
- **System Software 8.3:**  
El modo de puesta en servicio es posible siempre. Esto significa también, que es independiente del estado de las entradas de la interfaz de seguridad X11.

**Procedimiento**

- Activación del modo de puesta en servicio:

**Ruta:** botón KUKA > Puesta en servicio > Servicio > Modo de puesta en servicio

Menú	Descripción
	El modo de puesta en funcionamiento está activo. Al tocar la opción de menú se desactiva el modo.
	El modo de puesta en funcionamiento no está activo. Al tocar la opción de menú se activa el modo.

- El modo de puesta en servicio activo se señaliza mediante un indicador intermitente amarillo **IBN** en el HMI.

Se muestra el mensaje *Modo de puesta en servicio activo, PARADA DE EMERGENCIA tiene efecto SOLO LOCAL*.



Fig. 3-1: Modo puesta en funcionamiento

### 3.3 Principio de ajuste

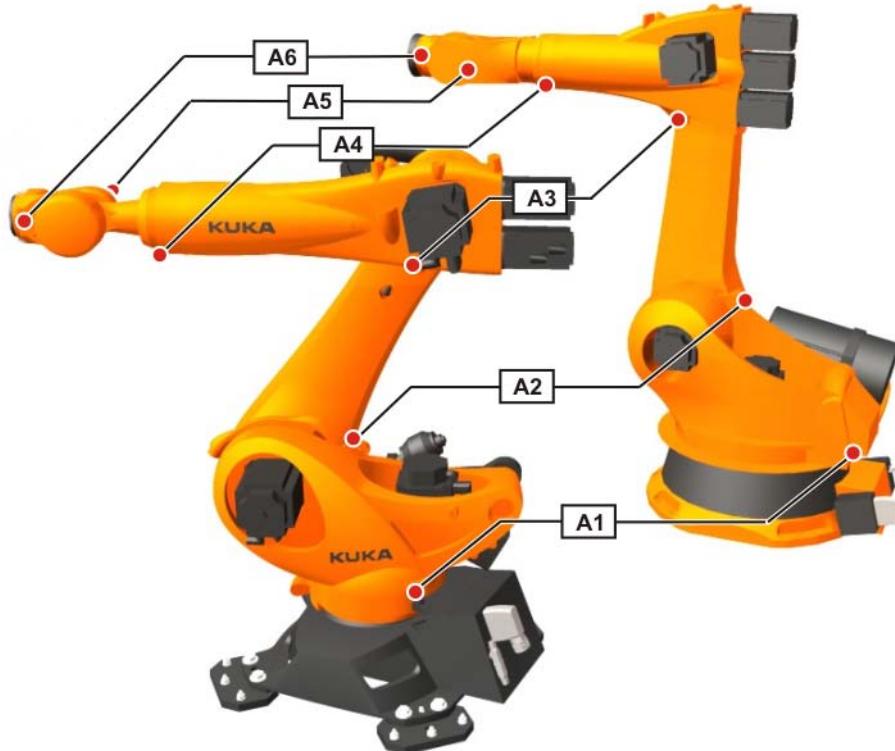
**¿Por qué ajustar?** Sólo un robot industrial perfecta y completamente ajustado puede funcionar de manera óptima. Porque solo entonces ofrece una total precisión en los puntos y la trayectoria y, sobre todo, puede desplazarse con movimientos programados.



Durante el ajuste a cada eje del robot se le asigna un valor de referencia. De esta forma, la unidad de control del robot sabe donde se encuentra el eje.

El proceso completo de ajuste incluye el ajuste de cada uno de los ejes. Con un medio auxiliar técnico (**EMD** = *Electronic Mastering Device*) se asigna un valor de referencia a cada eje en su **posición mecánica cero** (p. ej. 0°). Como con este proceso se hace coincidir la posición mecánica y eléctrica del eje, cada eje recibe un valor angular único. Para el ajuste del robot pequeño Agilus se usa el **MEMD - microEMD**.

Para todos los robots la posición de ajuste es similar, pero no idéntica. Las posiciones exactas pueden diferir también entre robots individuales de un mismo tipo de robot.



**Fig. 3-2: Posiciones de los cartuchos de ajuste**

Valores angulares de la posición mecánica cero (= valores de referencia)

Eje	Generación de robots "Quantec"	Otros tipos de robot (p. ej. la serie 2000, KR 16, etc.)
A1	-20°	0°
A2	-120°	-90°
A3	+110°	+90°
A4	0°	0°
A5	0°	0°
A6	0°	0°

## ¿Cuándo se realiza un ajuste?

- En principio, un robot siempre debe estar ajustado. Se debe realizar un ajuste en los siguientes casos:
- En la puesta en servicio
  - Después de haber efectuado trabajos de mantenimiento en los componentes que influyen sobre el registro de valores de posición (p. ej. el motor con resolver o RDC).
  - Cuando se hayan movido los ejes del robot sin el control, p. ej., por medio de un dispositivo de liberación.
  - Despues de haber efectuado reparaciones o solucionado problemas mecánicos, antes de que se pueda realizar el ajuste primero hay que desajustar el robot:
    - Despues de haber cambiado un engranaje.
    - Despues de una colisión contra un tope final con una velocidad mayor de 250 mm/s
    - Tras una colisión.



Antes de realizar cualquier trabajo de mantenimiento, por lo general resulta útil comprobar el ajuste actual.

## Advertencias de seguridad para el ajuste

En robots sin ajustar, su funcionamiento queda considerablemente limitado:

- No se puede utilizar el modo de programación: el robot no puede desplazarse a los puntos programados.
- No se puede realizar ningún movimiento manual cartesiano: no es posible efectuar movimiento en los sistemas de coordenadas.
- Los interruptores de final de carrera de software están desactivados.

### AVISO

En un robot desajustado, los interruptores de final de carrera de software están desactivados. El robot puede desplazarse y chocar contra los amortiguadores de los topes finales, por lo que podría sufrir daños y tener que cambiar los amortiguadores. En la medida de lo posible, no desplazar un robot desajustado o reducir al máximo el override manual.

## Realizar el ajuste



Fig. 3-3: EMD en acción

Se ajusta determinando el punto cero mecánico del eje. El eje se mueve hasta que se alcanza el punto cero mecánico. Este es el caso cuando el palpador llega al punto más profundo de la entalladura de medición. Por ello cada eje dispone de un cartucho y una marca de ajuste.

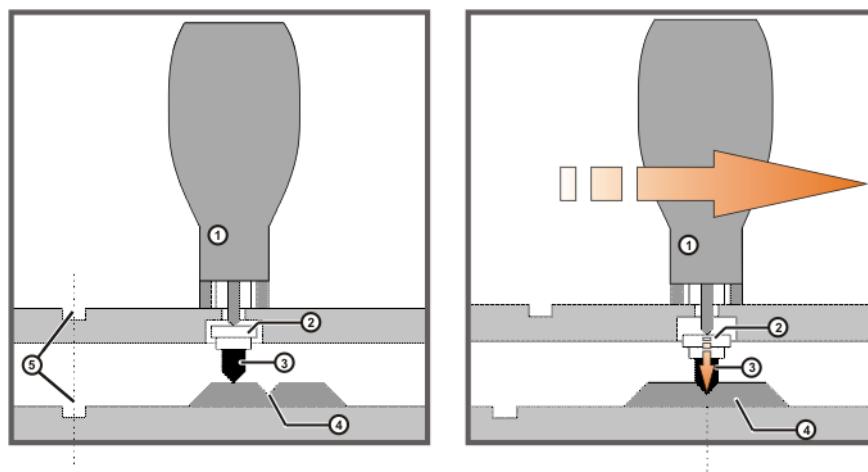


Fig. 3-4: Ejecución del ajuste EMD

- |                                     |                           |
|-------------------------------------|---------------------------|
| 1 EMD (Electronic Mastering Device) | 4 Entalladura de medición |
| 2 Cartucho de medición              | 5 Marca de preajuste      |
| 3 Palpador de medición              |                           |

### 3.4 Ajustar el robot

#### Opciones de ajuste de robot



Fig. 3-5: Opciones de ajuste

#### ■ Ajuste estándar

Columna de la figura: *Exactitud y situación de carga escasas*

Este tipo de ajuste se usa, cuando:

- El robot lleva, en su entorno de aplicación, una herramienta fija con un peso constante, p. ej., una boquilla de aplicación de pegamento
- Cuando la aplicación requiere una exactitud escasa, p. ej., paletizar paquetes

■ **Ajuste con corrección de peso**

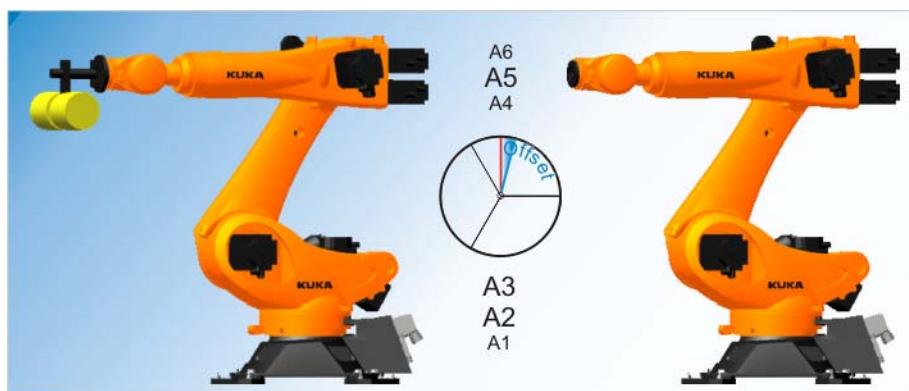
Columna de la figura: *Alta exigencia de exactitud o varias situaciones de carga*

Este tipo de ajuste se usa, cuando:

- Al mismo tiempo, se requiere una alta exactitud, p. ej. soldadura por láser
- El robot, en su entorno de aplicación, trabaja con cargas cambiantes, p. ej. garra con o sin carga

**¿Para qué sirve memorizar el offset?**

Debido al peso de la herramienta que está sujetada a la brida, el robot está sometido a una carga estática. A consecuencia de la elasticidad (en función del material) de los componentes y accionamientos puede haber diferencias en la posición de un robot cargado y otro sin carga. Estas diferencias de escasos incrementos repercuten en la precisión del robot.



**Fig. 3-6: Memorizar offset**

La función "Memorizar offset" se efectúa con carga. Entonces se guarda la diferencia con respecto al ajuste inicial (sin carga).

Cuando el robot trabaja con distintas cargas, debe ejecutarse la función de "Memorizar offset" para cada una de las cargas. En garras que recogen piezas pesadas, la función "Memorizar offset" se debe realizar para la garra sin pieza y para la garra con pieza.

Únicamente un robot ajustado con corrección de carga es capaz de ofrecer el nivel de precisión que se le exige. Por ello, para cada situación de carga se debe memorizar el offset. Como requisito es necesario que ya se haya efectuado la medición geométrica de la herramienta y, por tanto, que ya se haya asignado un número de herramienta.

**Ajustes en archivos log**

- Cuando se ajusta el robot, determinados datos específicos se protocolizan en un archivo log.
- Los Offset determinados se guardan, en grados, en el archivo C:\KRCIRO-BOTER\LOG\Mastery.log
- Los siguientes datos específicos de ajuste se guardan en el archivo **Mastery.log**:
  - Marca de tiempo (Fecha, hora)
  - Eje
  - Número de serie del robot
  - Valor de ajuste (*FirstEncoderValue*)
  - Número de herramienta
  - Valor offset (*Encoder Difference*) en grados en el motor

■ **Ejemplo de Mastery.log:**

```
Date: 01.09.11 Time: 13:41:07 Axis 1 Serialno.: 864585 First
Mastering (FirstEncoderValue: 1.138909)
Date: 01.09.11 Time: 13:42:07 Axis 2 Serialno.: 864585
First Mastering (FirstEncoderValue: 0.644334)
Date: 01.09.11 Time: 13:42:56 Axis 3 Serialno.: 864585
First Mastering (FirstEncoderValue: 0.745757)
Date: 01.09.11 Time: 13:43:29 Axis 4 Serialno.: 864585
First Mastering (FirstEncoderValue: 1.450234)
Date: 01.09.11 Time: 13:44:03 Axis 5 Serialno.: 864585
First Mastering (FirstEncoderValue: 0.686983)
Date: 01.09.11 Time: 13:44:30 Axis 6 Serialno.: 864585
First Mastering (FirstEncoderValue: 0.901439)
...
Date: 01.09.11 Time: 14:07:10
Axis 1 Serialno.: 864585
Tool Teaching for Tool No 1 (Encoder Difference: -0.0001209)
Date: 01.09.11 Time: 14:08:44
Axis 2 Serialno.: 864585
Tool Teaching for Tool No 1 (Encoder Difference: 0.005954)
...
```

**Procedimiento  
del ajuste inicial**

**AVISO**

El ajuste inicial sólo se debe realizar cuando el robot no está cargado. No se debe montar ninguna herramienta ni carga adicional.

1. Llevar el robot a la posición de ajuste inicial.



**Fig. 3-7: Ejemplos de posición de preajuste**

2. Seleccionar en el menú principal la secuencia **Puesta en servicio > Ajustar > EMD > Con corrección de peso > Ajuste inicial**.
- Se abre una ventana. Se visualizan todos los ejes que se deben ajustar. El eje con el número más bajo está marcado.
3. En el eje marcado en la ventana, retirar la tapa protectora del cartucho de medición. Si se le da la vuelta al EMD, puede utilizarse como destornillador. Enroscar el EMD en el cartucho de medición.



**Fig. 3-8: EMD enroscado en el cartucho de medición**

4. Colocar el cable de medición en el EMD y conectarlo en el conector X32 de la caja de conexiones del robot.



Fig. 3-9: Cable EMD, conectado

**⚠ ATENCIÓN**

Enroscar el EMD en el cartucho de medición siempre sin cable de medición. Montar entonces el cable de medición en el EMD. En caso contrario, se puede dañar el cable de medición.  
Retirar también siempre el cable de medición del EMD antes de retirar el EMD. Solo entonces se puede retirar el EMD del cartucho de medición.  
Después del ajuste, desmontar el cable de medición de la conexión X32. En caso contrario, se pueden producir señales parásitas o causar daños materiales.

5. Pulsar **Ajustar**.
6. Situar el pulsador de validación en la posición intermedia, pulsar la tecla de arranque y mantenerla pulsada.



Fig. 3-10: Tecla de inicio y pulsador de confirmación

Cuando el EMD ha recorrido el punto más bajo de la entalladura de medición, quiere decir que se ha alcanzado la posición de ajuste. El robot se detiene automáticamente. Los valores se guardan. En la ventana, el eje queda oculto.

7. Retirar el cable de medición del EMD. A continuación, retirar el EMD del cartucho de medición y volver a colocar la tapa protectora.
8. Repetir los pasos 2 a 5 en todos los ejes que se deseen ajustar.
9. Cerrar la ventana.
10. Retirar de la conexión X32 el cable de medición.

**Procedimiento para memorizar Offset**

La función "Memorizar offset" se efectúa con carga. Se guarda la diferencia respecto al ajuste inicial.

1. Llevar el robot a la posición de preajuste.
2. Seleccionar en el menú principal la secuencia **Puesta en servicio > Ajustar > EMD > Con corrección de peso >** seleccionar **memorizar offset**.

3. Introducir el número de herramienta. Confirmar con **Herram. OK**.  
Se abre una ventana. Se muestran todos los ejes que aún no se han memorizado para la herramienta. El eje con el número más bajo está marcado.
4. En el eje marcado en la ventana, retirar la tapa protectora del cartucho de medición. Enroscar el EMD en el cartucho de medición. Colocar el cable de medición en el EMD y conectarlo en la conexión X32 de la caja de conexiones situada en la base del robot.
5. Pulsar en **Memorizar**.
6. Pulsar el pulsador de validación y la tecla de arranque.  
Cuando el EMD detecta el punto más bajo de la entalladura de medición, se ha alcanzado la posición de ajuste. El robot se detiene automáticamente. Se abre una ventana. La diferencia de este eje respecto al ajuste inicial se muestra en incrementos y grados.
7. Confirmar con **OK**. En la ventana, el eje queda oculto.
8. Retirar el cable de medición del EMD. A continuación, retirar el EMD del cartucho de medición y volver a colocar la tapa protectora.
9. Repetir los pasos 3 a 7 en todos los ejes que se deseen ajustar.
10. Retirar de la conexión X32 el cable de medición.
11. Abandonar la ventana pulsando en **Cerrar**.

**Procedimiento para controlar/ establecer el ajuste de carga con offset**

- El ajuste de carga con offset se realiza con carga. Se calcula el ajuste inicial.
1. Llevar el robot a la posición de ajuste inicial.
  2. Seleccionar en el menú principal la secuencia **Puesta en servicio > Ajustar > EMD > Con corrección de carga > Ajuste de carga > con offset**.
  3. Introducir el número de herramienta. Confirmar con **Herram. OK**.
  4. Retirar la tapa al conector X32 y conectar el cable de medición.
  5. En el eje marcado en la ventana, retirar la tapa protectora del cartucho de medición. (Si se le da la vuelta al EMD, puede utilizarse como destornillador).
  6. Enroscar el EMD en el cartucho de medición.
  7. Montar el cable de medición en el EMD. Para ello alinear el punto rojo del conector con la ranura del EMD.
  8. Pulsar **Probar**.
  9. Mantener pulsado el pulsador de validación y pulsar la tecla de arranque.
  10. En caso necesario, guardar los valores con **Guardar**. Con ello se borran los datos de ajuste antiguos. Para poder restaurar un ajuste inicial perdido, guardar siempre los valores.
  11. Retirar el cable de medición del EMD. A continuación, retirar el EMD del cartucho de medición y volver a colocar la tapa protectora.
  12. Repetir los pasos 4 a 10 en todos los ejes que se deseen ajustar.
  13. Cerrar la ventana.
  14. Retirar de la conexión X32 el cable de medición.



El tipo de ajuste a usar (estándar o con corrección de peso) en instalaciones existentes debe corresponder al tipo de ajuste usado durante la puesta en servicio. Si se usa un tipo de ajuste incorrecto, puede conllevar a un robot mal ajustado.

**MEMD - Micro Electronic Mastering Device**

Para los robots pequeños de KUKA se ha desarrollado, conjuntamente con el KR C4 compact, un **MEMD - Micro Electronic Mastering Device**.

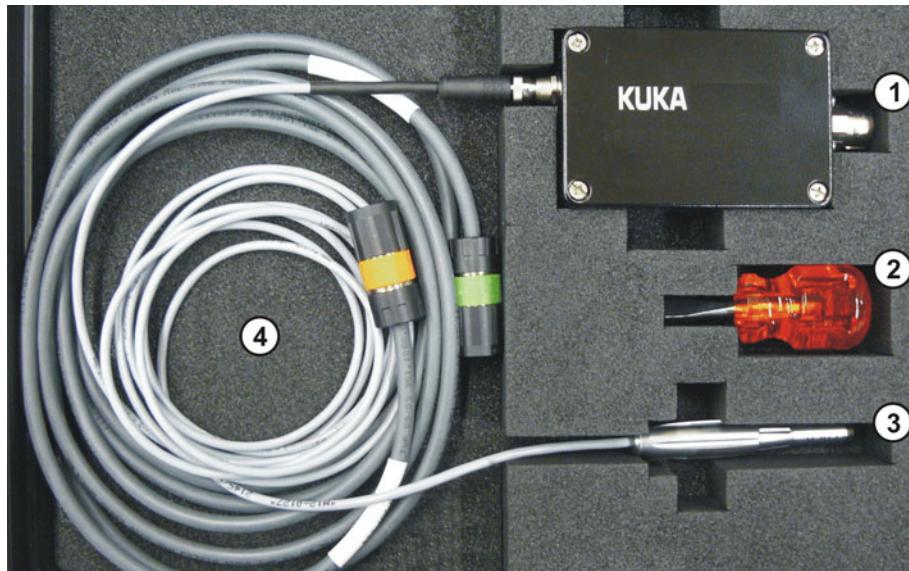


Fig. 3-11: Maletín MEMD

- |                  |          |
|------------------|----------|
| 1 Caja MEMD      | 3 MEMD   |
| 2 Destornillador | 4 Cables |

- El cable más fino es el cable de medición. Conecta el MEMD con la caja MEMD.
- El cable más grueso es el cable EtherCAT. Se conecta a la caja MEMD y al robot en X32.

#### AVISO

Dejar conectado el cable de medición en la caja MEMD y desconectarlo con la menor frecuencia posible. La posibilidad de conexión del conector enchufable del sensor M8 está limitada. En caso de conexión/desconexión frecuente en el conector enchufable, pueden producirse daños.

#### Ajuste A6, robot pequeño AGILUS

El A6 se ajusta sin MEMD.

Antes del ajuste, se debe desplazar el A6 a su posición de ajuste (es decir, antes del proceso de ajuste completo, no justo antes del ajuste propiamente dicho del A6). A este fin, en el A6 se encuentran dos marcas finas en el metal.

- Para colocar el A6 en la posición de ajuste, alinear ambas marcas de forma exacta.



Para el desplazamiento a la posición de ajuste es importante mirar en línea recta desde delante sobre la raya fija. Si la raya se observa desde el lado, la línea móvil no se podrá alinear con suficiente exactitud. La consecuencia es un ajuste incorrecto.



**Fig. 3-12: Posición de ajuste A6 – Vista desde delante**

- Seleccionar en el menú principal **Puesta en servicio > Ajustar > Referencia**.  
Se abre la ventana de opción **Ajuste de referencia**. El eje A6 se visualiza y está marcado.
- Pulsar **Ajustar**. El eje A6 se ajusta y se oculta en la ventana de opciones.
- Cerrar la ventana.
- Retirar el cable EtherCAT de la conexión X32 y de la caja MEMD.

### 3.4.1 Ejercicio: Ajuste de robot

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"> <li>■ Desplazar a posición de preajuste</li> <li>■ Selección del tipo de ajuste correcto</li> <li>■ Uso del "Electronic Mastering Device" (EMD)</li> <li>■ Ajuste de todos los ejes mediante EMD</li> </ul>
<b>Requisitos</b>	Los siguientes requisitos son imprescindibles para superar este ejercicio con éxito:
	<ul style="list-style-type: none"> <li>■ Conocimientos teóricos sobre el desarrollo general de un ajuste</li> <li>■ Conocimientos teóricos de la posición de preajuste</li> </ul>
	 
	<p>1      Eje no en posición de preajuste <span style="color: red;">✗</span></p> <p>2      Eje en posición de preajuste <span style="color: green;">✓</span></p>
	<ul style="list-style-type: none"> <li>■ Conexión correcta del EMD al robot</li> <li>■ Ajuste a través del menú de puesta en servicio</li> </ul>
<b>Formulación de tarea</b>	Ejecutar las siguientes tareas:
	<ol style="list-style-type: none"> <li>1. Desajustar todos los ejes del robot.</li> <li>2. Desplazar todos los ejes de robot de forma específica según el eje a la posición de preajuste.</li> <li>3. Efectuar un ajuste de carga con Offset en todos los ejes con el EMD/EMD-MD.</li> <li>4. Utilizar para ello la herramienta 3.</li> <li>5. Visualizar la posición real específica del eje.</li> </ol>
	<b>Lo que se debe saber tras el ejercicio:</b>
	<ol style="list-style-type: none"> <li>1. ¿Para qué se realiza un ajuste?</li> </ol> <p>.....</p> <ol style="list-style-type: none"> <li>2. Indicar los ángulos de los 6 ejes de la posición mecánica cero.</li> </ol>
	A1: .....                          A2: .....
	A3: .....                          A4: .....
	A5: .....                          A6: .....
	3. ¿Qué debe tenerse en cuenta en caso de un robot desajustado?

4. ¿Qué medio de ajuste se debe utilizar preferentemente?

.....

.....

5. ¿Cuáles son los peligros si se desplaza un robot con el EMD (comparador) conectado?

### 3.5 Cargas sobre el robot

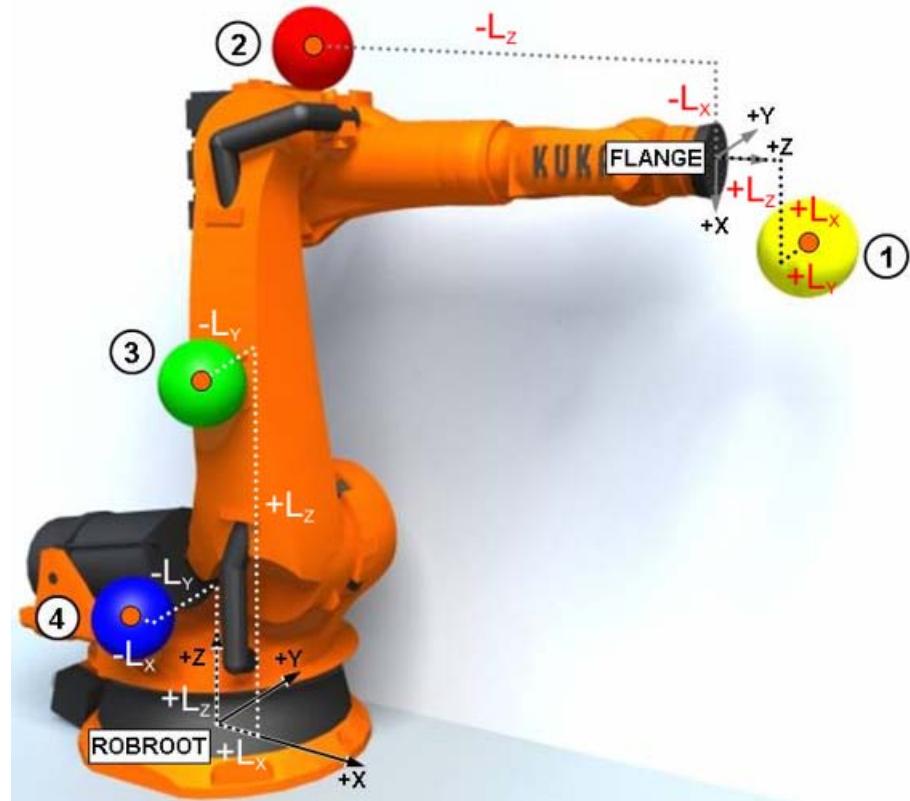


Fig. 3-13: Cargas en el robot

- |                                  |                                  |
|----------------------------------|----------------------------------|
| 1 Capacidad de carga             | 3 Carga adicional sobre el eje 2 |
| 2 Carga adicional sobre el eje 3 | 4 Carga adicional sobre el eje 1 |

### 3.6 Datos de carga de la herramienta

#### ¿Qué son los datos de carga de la herramienta?

Los datos de carga de la herramienta son todas aquellas cargas montadas sobre la brida del robot. Forman una masa adicional sobre el robot que debe moverse con él.

Los valores que deben introducirse son la masa, la posición del centro de gravedad (punto en el que incide la masa) y los momentos de inercia de la masa con los ejes principales de inercia correspondientes.

Los datos de carga **deben** introducirse en la unidad de control del robot y asignarse a la herramienta pertinente.

Excepción: Si los datos de carga ya se han transferido a la unidad de control del robot con KUKA.LoadDataDetermination, ya no es necesario introducirlos manualmente.

**Los datos de carga de la herramienta se pueden consultar en las fuentes siguientes:**

- Opción de software KUKA.LoadDetermination (sólo para cargas)
- Datos específicos del fabricante
- Cálculo manual
- Programas CAD

#### Efectos de los datos de carga

Los datos de carga introducidos ejercen su influencia sobre numerosos procesos de control. Entre ellas, las siguientes:

- Algoritmos de control ((cálculo de la aceleración)

- Monitorización de velocidades y aceleraciones
- Momento esfuerzo
- Control contra colisiones
- Control de energía
- entre otros muchos

Por ello, es de vital importancia que los datos de carga se introduzcan correctamente. Si el robot ejecuta sus movimientos con los datos de carga correctamente introducidos...

- se puede aprovechar su alta precisión,
- las secuencias de movimientos se pueden ejecutar con tiempos de ciclo óptimos,
- el robot alcanza una larga durabilidad (debido al poco desgaste).

### **Procedimiento**

1. En el menú principal, seleccionar **Puesta en servicio > Medir > Herramienta > Datos de carga de la herramienta.**
2. En el campo **Herramienta Nº** introducir el número de la herramienta. Confirmar pulsando **Continuar**.
3. Declarar los datos de carga:
  - Campo **M**: Masa
  - Campos **X, Y, Z**: Situación del centro de gravedad relativo a la brida
  - Campos **A, B, C**: Orientación de los ejes principales de inercia relativos a la brida
  - Campos **JX, JY, JZ**: Momentos de inercia de la masa  
(JX es la inercia alrededor del eje X del sistema de coordenadas que está torcido por A, B y C en relación a la brida. De forma análoga, JY y JZ son las inercias alrededor de los ejes Y y Z.)
4. Confirmar pulsando **Continuar**.
5. Pulsar **Guardar**.

### **3.7 Supervisar los datos de carga de la herramienta**

#### **Descripción**

En numerosos tipos de robots, la unidad de control del robot controla durante el servicio si existe sobrecarga o carga reducida. Este control se denomina "Control online de datos de carga" (= "OLDC" / "Online Load Data Check").

Si el OLDC, p. ej., determina una carga reducida, la unidad de control del robot muestra como reacción, p. ej., una mensaje. Las reacciones se pueden configurar.

El resultado de la comprobación también se puede consultar mediante la variable del sistema \$LDC\_RESULT.

El OLDC está disponible para los tipos de robots para los que también se puede utilizar KUKA.LoadDataDetermination. Para saber si el OLDC está disponible para el tipo de robot actual, se podrá consultar a través \$LDC\_LOADED (TRUE = sí).

Sobrecarga	Existirá una sobrecarga cuando la carga real sea superior a la carga configurada.
Carga reducida	Existirá una carga reducida cuando la carga real sea inferior a la carga configurada.

#### **Activación y configuración**

El OLDC se puede configurar en los siguientes puntos:

- Durante la introducción manual de los datos de herramienta

- Durante la introducción separada de los datos de carga

En la misma ventana en la que también se han introducido los datos de carga, se visualizan los siguientes campos:

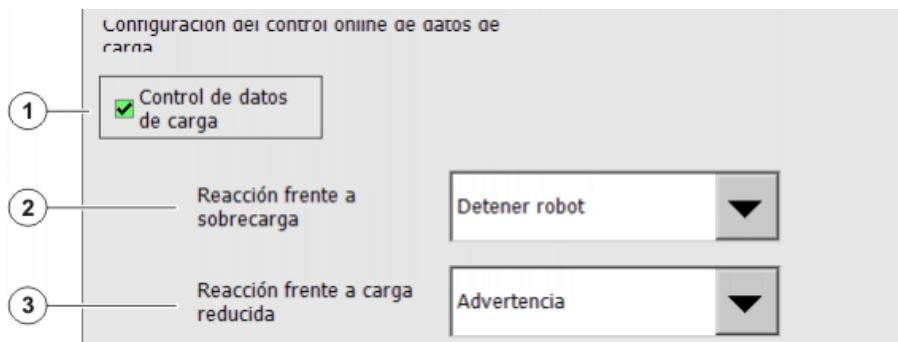


Fig. 3-14: Control online de datos de carga

Pos.	Descripción
1	<p><b>TRUE:</b> El OLCD está activo para la herramienta visualizada en la misma ventana. En caso de sobrecarga o de carga reducida se producirán las reacciones definidas.</p> <p><b>FALSE:</b> El OLCD está inactivo para la herramienta visualizada en la misma ventana. En caso de sobrecarga o de carga reducida no se producirá ninguna reacción.</p>
2	<p>Aquí se puede definir la reacción que debe producirse en caso de sobrecarga.</p> <ul style="list-style-type: none"> <li>■ <b>Ninguna:</b> sin reacción.</li> <li>■ <b>Advertencia:</b> La unidad de control del robot envía el siguiente mensaje de estado: <i>Al controlar la carga de robot (Tool {N.º}) se ha determinado sobrecarga.</i></li> <li>■ <b>Detener el robot:</b> La unidad de control del robot envía un mensaje de acuse de recibo con el mismo contenido que para una <b>Advertencia</b>. El robot se detiene con STOP 2.</li> </ul>
3	<p>Aquí se puede definir la reacción que debe producirse en caso de carga reducida. Las posibles reacciones son análogas a la sobrecarga.</p>

## NULLFRAME

Para los movimientos a los que esté asignada la herramienta **NULLFRAME** no se puede el OLCD. Las reacciones para este caso están establecidas y el usuario no puede influir en las mismas.

- Reacción frente a sobrecarga: **Detener robot**

Se emite el siguiente mensaje de confirmación: *Al verificar la carga de robot (no hay herramienta definida) y los datos de carga ajustados se ha detectado sobrecarga.* El robot se detiene con STOP 2.

- Reacción frente a carga reducida: **Advertencia**

Se emite el siguiente mensaje de estado: *Al verificar la carga de robot (no hay herramienta definida) y los datos de carga ajustados se ha detectado carga reducida.*

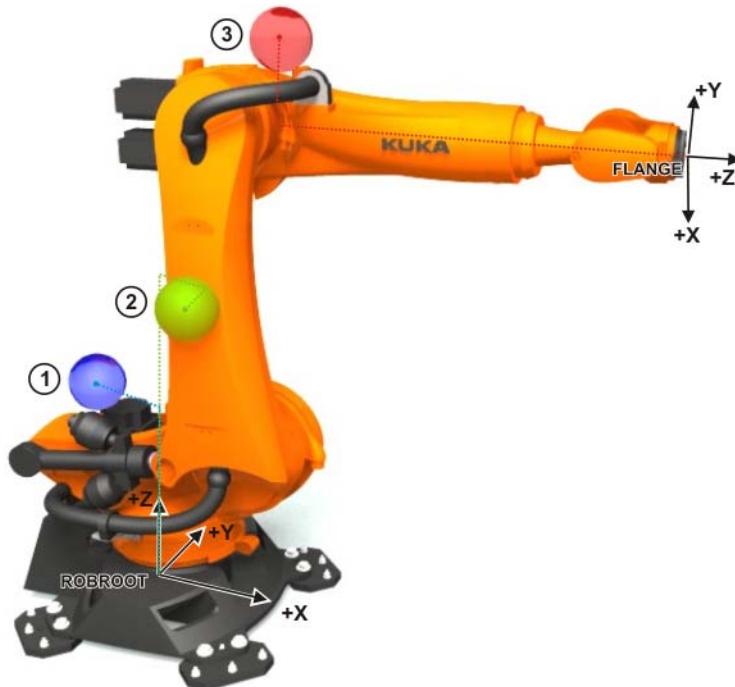
## 3.8 Datos de carga adicional en el robot

### Datos de carga adicional en el robot

Las cargas adicionales son componentes instalados adicionalmente en la base, el brazo de oscilación o el brazo. Por ejemplo:

- Alimentación de energía

- Válvulas
- Alimentación de materia
- Provisión de material



**Fig. 3-15: Datos de carga adicional en el robot**

Los datos de carga adicionales se deben introducir en la unidad de control del robot. Entre otros, es necesario indicar los datos siguientes:

- Masa (m) en kg
- Distancia del centro de gravedad de la masa al sistema de referencia (X, Y y Z) en mm.
- Orientación de los ejes de inercia principales al sistema de referencia (A, B y C) en grados ( $^{\circ}$ ).
- Momentos de inercia de la masa alrededor de los ejes de inercia ( $J_x$ ,  $J_y$  y  $J_z$ ) en  $\text{kgm}^2$ .

Sistemas de referencia para los valores X, Y y Z por cada carga adicional:

Carga	Sistema de referencia
Carga adicional A1	Sistema de coordenadas ROBROOT A1 = 0°
Carga adicional A2	Sistema de coordenadas ROBROOT A2 = -90°
Carga adicional A3	Sistema de coordenadas FLANGE A4 = 0°, A5 = 0°, A6 = 0°

Los datos de carga pueden ser consultados en las siguientes fuentes:

- Opción de software KUKA.LoadDetect (sólo para cargas)
- Datos específicos del fabricante
- Cálculo manual
- Programas CAD

### Influencia de las cargas adicionales sobre el movimiento del robot

Los datos que carga influyen sobre el movimiento del robot de las formas más diversas:

- Proyecto de trayectoria
- Aceleraciones
- Duración del ciclo
- Desgaste

#### **ADVERTENCIA**

Si un robot se hace funcionar con unos datos de carga erróneos o incorrectos, existe peligro de muerte, de lesiones o de que se produzcan daños materiales considerables.

### Procedimiento

1. En el menú principal, seleccionar **Puesta en servicio > Medir > Datos de carga adicional**.
2. Indicar el número del eje sobre el cual va montada la carga adicional. Confirmar pulsando **Continuar**.
3. Declarar los datos de carga. Confirmar pulsando **Continuar**.
4. Pulsar **Guardar**.

### 3.9 Mover el robot en el sistema de coordenadas de la herramienta

#### Movimientos manuales en el sistema de coordenadas de herramienta

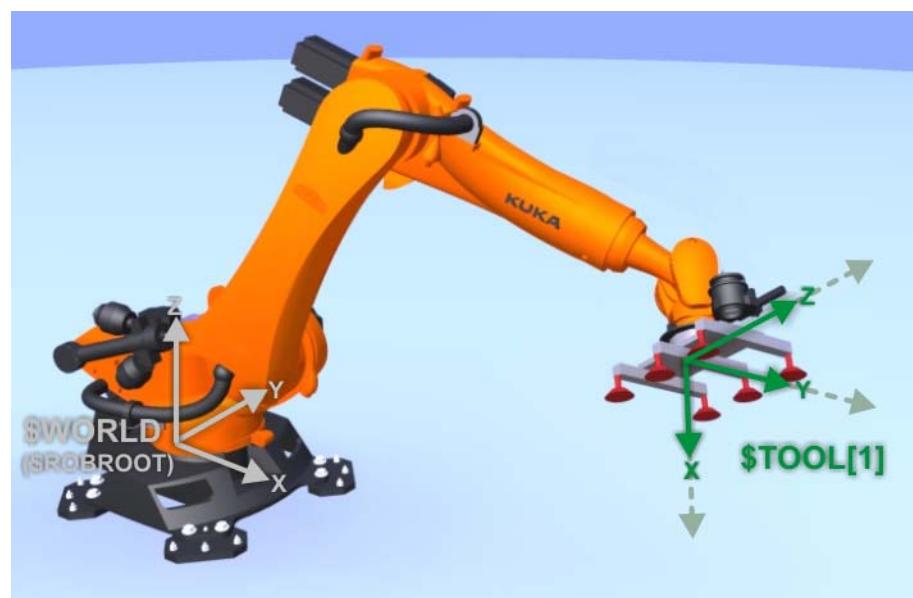


Fig. 3-16: Sistema de coordenadas de herramienta de robot

- En los movimientos manuales dentro del sistema de coordenadas de herramienta es posible desplazar el robot en base a las direcciones de coordenadas de una herramienta previamente medida.

Por lo tanto, el sistema de coordenadas no está sujeto a un lugar fijo (comp. sistema de coordenadas universales/base), sino que lo dirige el propio robot.

**Todos** los ejes del robot necesarios se mueven. El mismo sistema decide los ejes que deben moverse dependiendo del movimiento.

El origen del sistema de coordenadas de herramienta se denomina **TCP - Tool Center Point** - y se corresponde con el punto de trabajo de la herramienta, p. ej., la punta de una boquilla de aplicación de pegamento.

- Para ello se utilizan las teclas de desplazamiento o el Space Mouse del KUKA smartPAD.
- Hay disponibles 16 sistemas de coordenadas de herramienta distintos.
- La velocidad puede cambiarse (override manual: HOV).

- Los procesos manuales sólo se pueden efectuar en el modo de servicio T1.
- El pulsador de validación debe estar pulsado.



### Principio del proceso manual de herramienta

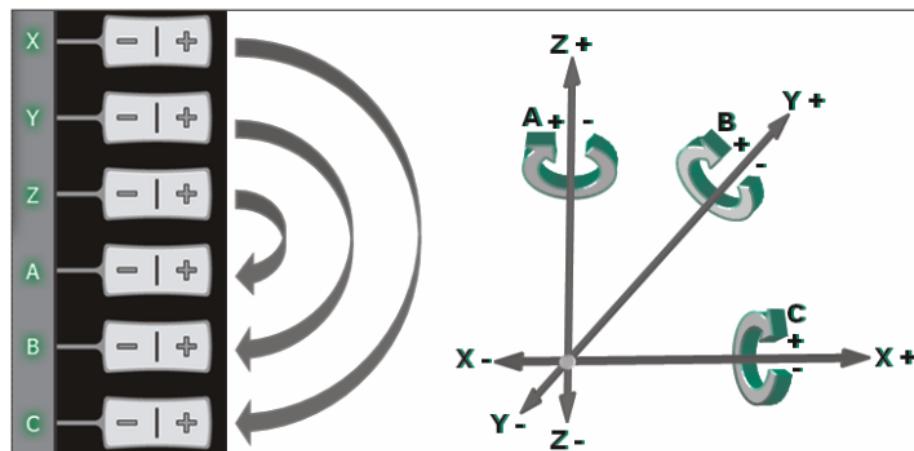


Fig. 3-17: Sistema de coordenadas cartesiano

**Un robot se puede mover de dos maneras distintas en un sistema de coordenadas:**

- Traslatoria (rectilínea) a lo largo de la dirección de orientación del sistema de coordenadas: X, Y, Z
- Rotacional (giratoria / viratoria) en la dirección de orientación del sistema de coordenadas: ángulos A, B y C

**Ventajas de usar el sistema de coordenadas de herramienta:**

- El movimiento del robot siempre es predecible mientras se conozca el sistema de coordenadas de herramienta.
- Cabe la posibilidad de desplazar en la dirección de avance de la herramienta o de orientarse con relación al TCP.

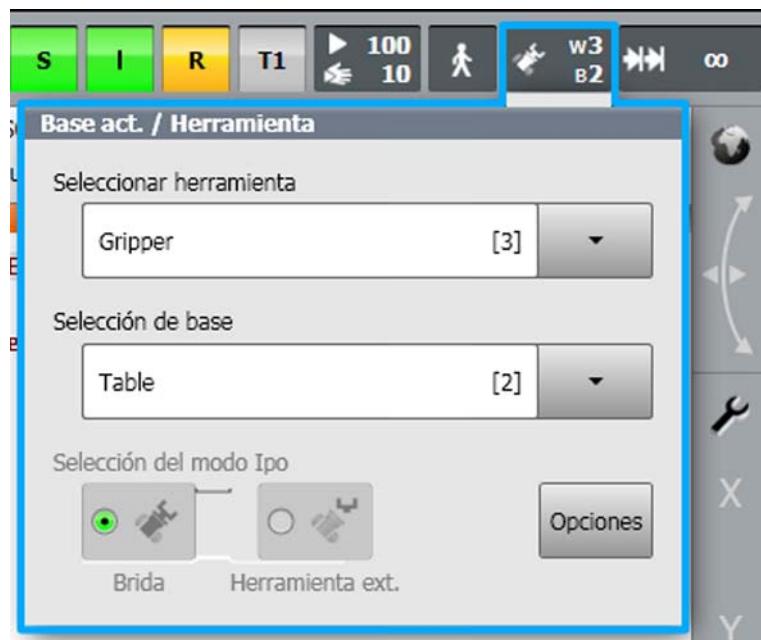
Por *dirección de impacto de la herramienta* se entiende la dirección de trabajo y de proceso de la misma, como por ejemplo, la dirección de salida del pegamento en una boquilla o la dirección de agarre de las garras al coger un componente.

**Procedimiento**

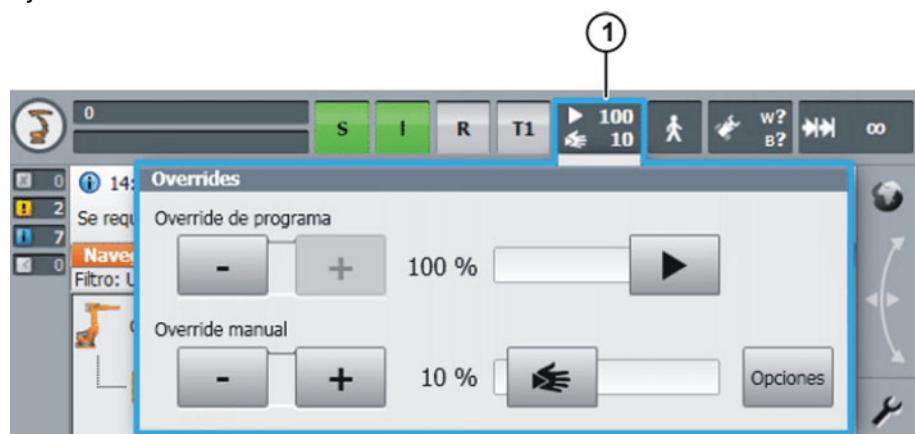
1. Seleccionar **herramienta** como el sistema de coordenadas a utilizar.



2. Seleccionar número de herramienta



3. Ajustar el override manual



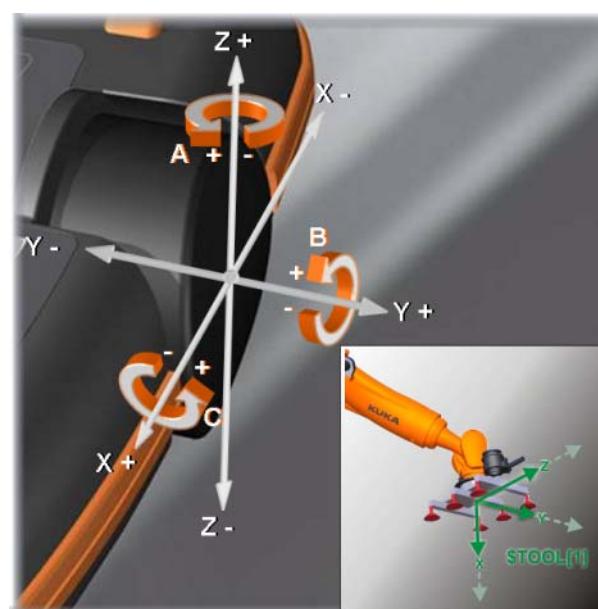
4. Pulsar el pulsador de validación en la posición intermedia y mantenerlo pulsado.



5. Mover el robot con las teclas de desplazamiento.



6. Alternativa: Con el Space Mouse mover el robot en la dirección correspondiente.



### 3.9.1 Ejercicio: Desplazamiento manual en el sistema de coordenadas de la herramienta

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"><li>■ Desplazamiento manual del robot en el sistema de coordenadas de la herramienta con las teclas de desplazamiento y el Space Mouse</li><li>■ Desplazamiento manual del robot en la dirección de impacto de la herramienta</li></ul>
<b>Requisitos</b>	Los siguientes requisitos son necesarios para completar este ejercicio correctamente: <ul style="list-style-type: none"><li>■ Participación en las instrucciones de seguridad</li></ul>
<b>Formulación de tarea</b>	<p><b>Aviso</b> Antes de comenzar el ejercicio se debe participar en las instrucciones de seguridad con documentación correspondiente.</p> <ul style="list-style-type: none"><li>■ Ejecutar las siguientes tareas:<ol style="list-style-type: none"><li>1. Desbloquear y confirmar la PARADA DE EMERGENCIA</li><li>2. Asegurarse de que esté ajustado el modo de servicio T1</li><li>3. Activar el sistema de coordenadas de la herramienta</li><li>4. Desplazar el robot en el sistema de coordenadas de la herramienta con diversos ajustes del override manual (HOV) mediante las teclas de desplazamiento manual y el Space Mouse. Comprobar el desplazamiento en la dirección de impacto y la reorientación alrededor del TCP.</li><li>5. Con ayuda de la herramienta "Garra", sacar la clavija del soporte</li></ol></li></ul>

### 3.10 Medición de una herramienta

#### Descripción

Mediante la medición de la herramienta (sistema de coordenadas **TOOL**), la unidad de control del robot sabe dónde se encuentra la punta de la herramienta (**TCP - Tool Center Point**) en relación al punto central de la brida y cómo está orientada.

La medición de la herramienta contiene entonces la medición de:

- el TCP (origen del sistema de coordenadas)
- la orientación del sistema de coordenadas



**Fig. 3-18: Ejemplos de herramientas medidas**



Se pueden guardar como máximo 16 sistemas de coordenadas de herramienta. (Variable: **TOOL\_DATA[1...16]**).

Durante la medición se guarda la distancia del sistema de coordenadas de la herramienta (en X, Y y Z) con respecto al sistema de coordenadas de la brida, así como el giro de este sistema de coordenadas (ángulos A, B y C).

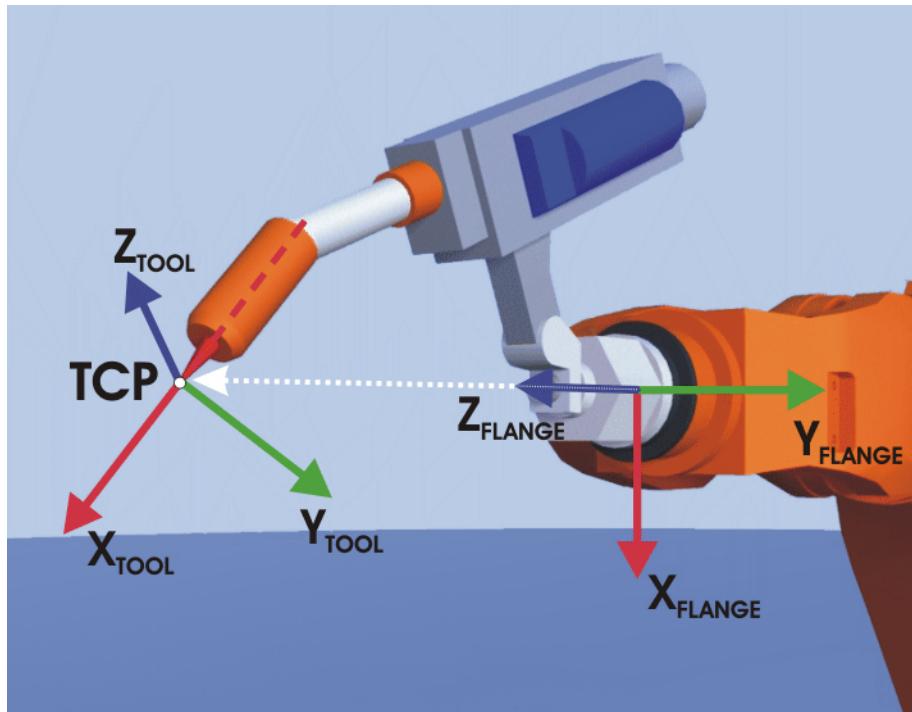


Fig. 3-19: Principio de la medición TCP

#### Ventajas

Cuando una herramienta ha sido medida con exactitud, el personal de operación y de programación cuenta con las ventajas siguientes:

- Procesos manuales perfeccionados
  - Se puede efectuar la reorientación del TCP (por ejemplo, punta de la herramienta).

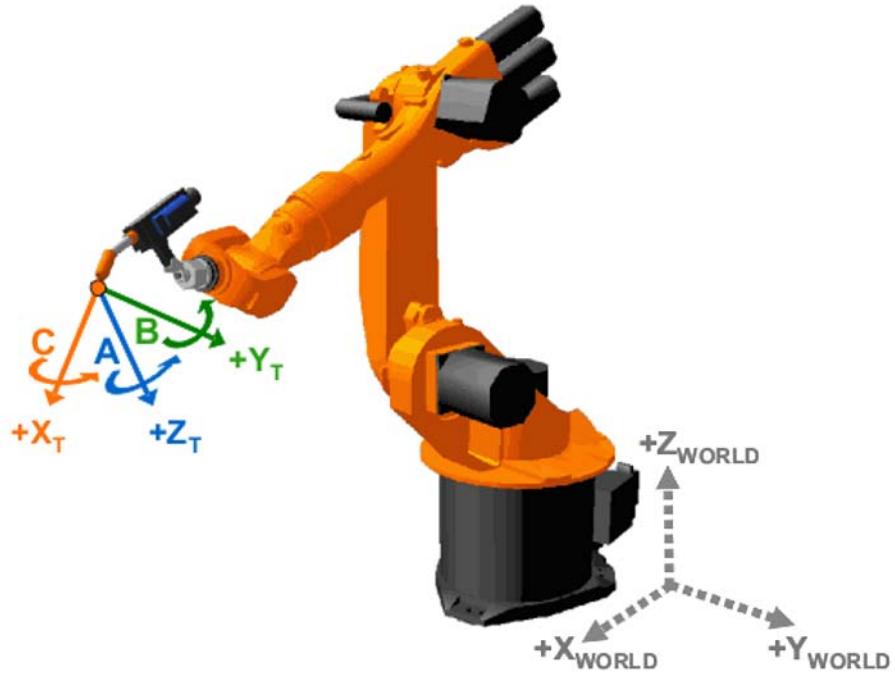
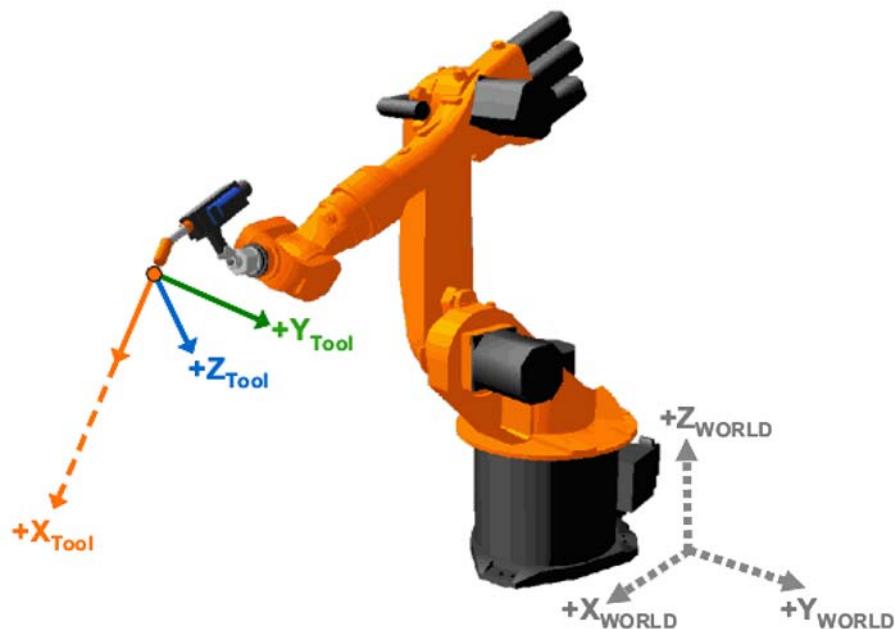


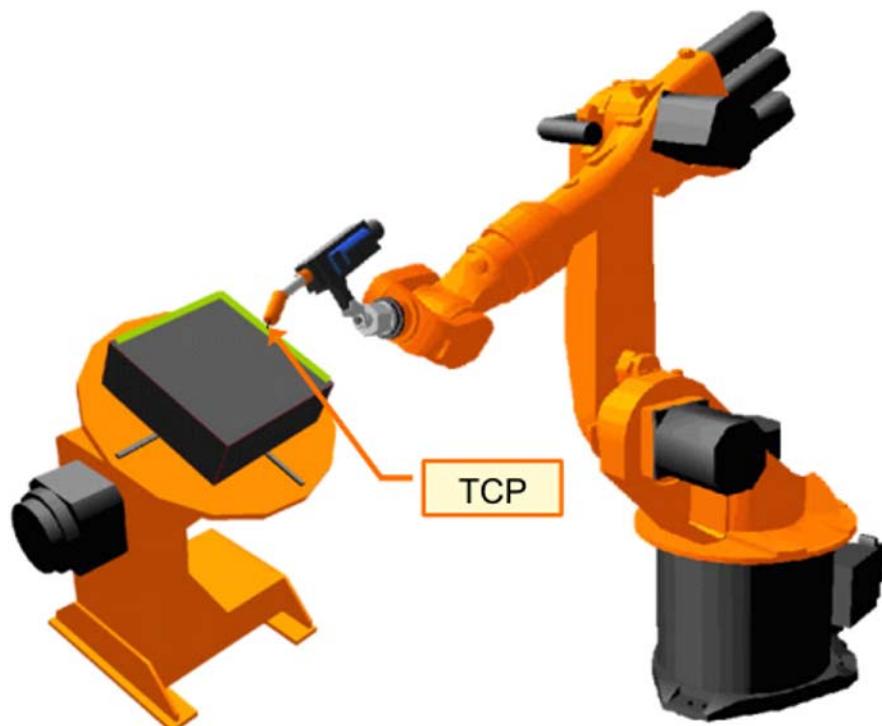
Fig. 3-20: Reorientación alrededor del TCP

- Desplazamiento en dirección de avance de la herramienta



**Fig. 3-21: Dirección avance TCP**

- Utilidad en la programación de movimientos de trayectoria (movimiento lineal o circular)
  - La velocidad programada se mantiene en el TCP a lo largo de toda la trayectoria.



**Fig. 3-22: Servicio de programa con TCP**

- Además es posible un control de la orientación definido a lo largo de la trayectoria.

#### Opciones de medición de la herramienta

La medición de la herramienta consta de 2 pasos:

Paso	Descripción
1	<p><b>Definir el origen del sistema de coordenadas TOOL</b></p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> <li>■ <i>Punto XYZ-4</i></li> <li>■ <i>Referencia XYZ</i></li> </ul>
2	<p><b>Definir la orientación del sistema de coordenadas TOOL</b></p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> <li>■ <i>ABC World</i></li> <li>■ <i>Punto ABC-2</i></li> </ul>
Alternativa:	<p>Entrada directa de valores para la distancia al punto central de la brida (X, Y, Z) y al giro (A, B, C).</p> <ul style="list-style-type: none"> <li>■ <i>Entrada numérica</i></li> </ul>

### Medición TCP, método XYZ 4 puntos

Con el TCP de la herramienta que se desea medir puede alcanzarse un punto de referencia desde 4 direcciones diferentes. El punto de referencia puede ser cualquiera. La unidad de control del robot calcula el TCP a partir de las distintas posiciones de la brida.



Las 4 posiciones de la brida con las cuales el robot se desplaza al punto de referencia deben estar suficientemente separadas.

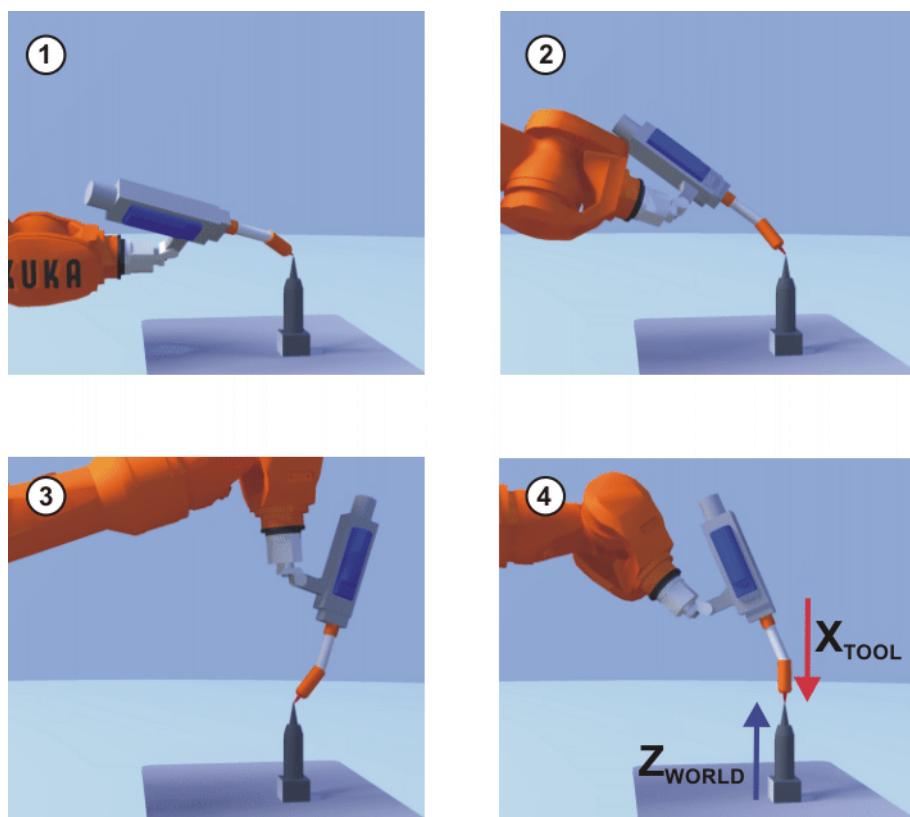
#### Procedimiento del método XYZ de 4 puntos:

1. Seleccionar la secuencia de menú **Puesta en servicio > Medición > Herramienta > XYZ 4 puntos**.
2. Indicar un número y un nombre para la herramienta que se quiere medir. Confirmar con **OK**.



Se puede escoger entre los números del 1 al 16.

3. Acercar un punto de referencia con el TCP. Confirmar con **OK**.
4. Con el TCP desplazarse al punto de referencia desde otra dirección. Confirmar con **OK**.



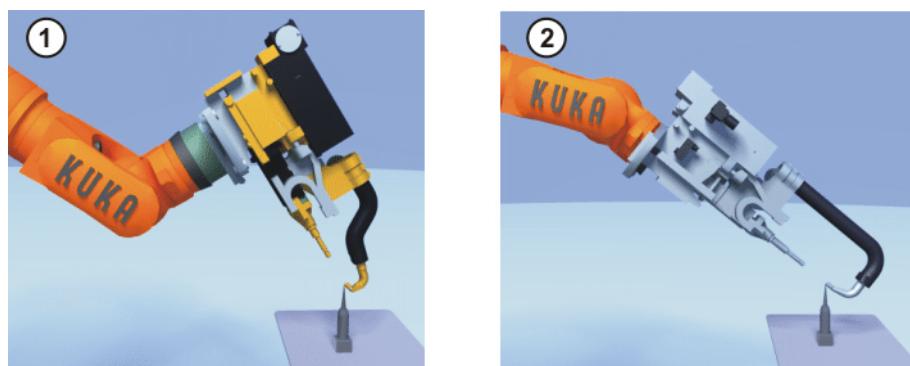
**Fig. 3-23: Método XYZ 4-Puntos**

5. Repetir dos veces el paso 4.
6. Pulsar **Guardar**.

### Medición TCP con método de referencia XYZ

En el *método de referencia XYZ*, la medición de una nueva herramienta se efectúa con una herramienta ya medida. La unidad de control del robot compara las posiciones de la brida y calcula el TCP de la nueva herramienta.

El *método de referencia XYZ* se usa para el memorizado de varias herramientas del mismo tipo con geometrías similares. El número de desplazamientos de medición se puede reducir así a dos, comparado con el *método XYZ de 4-puntos*.



**Fig. 3-24: Método XYZ - Referencia**

### Procedimiento

1. Requisito para este procedimiento es que la herramienta ya medida debe estar montada en la brida de acople y que ya se conozcan los datos del TCP.
2. Seleccionar en el menú principal la secuencia **Puesta en servicio > Me-dir > Herramienta > Referencia XYZ**.

3. Asignar un número y un nombre para la nueva herramienta. Confirmar pulsando **Continuar**.
4. Introducir los datos del TCP de la herramienta ya medida. Confirmar pulsando **Continuar**.
5. Acercar un punto de referencia con el TCP. Pulsar en **Medir**. Confirmar pulsando **Continuar**.
6. Liberar la herramienta y desmontarla. Montar la nueva herramienta.
7. Acercar el punto de referencia con el TCP de la nueva herramienta. Pulsar en **Medir**. Confirmar pulsando **Continuar**.
8. Pulsar **Guardar**. Los datos se guardan y la ventana se cierra.

O pulsar **Datos de carga**. Los datos se guardan y se abre una ventana en la que se pueden introducir los datos de carga.

### Medición de la orientación, método ABC- World

Los ejes del sistema de coordenadas de herramienta se alinean de forma paralela a los ejes del sistema de coordenadas de herramienta. De este modo, la unidad de control del robot conoce la orientación del sistema de coordenadas de herramienta.

**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 avance es el eje X. La dirección de los demás ejes la determina el sistema, y en general no puede ser reconocida fácilmente por el usuario.  
Casos de aplicación: por ej. Soldadura MIG/MAG, Laser o corte por chorro de agua
- **6D:** A la unidad de control del robot se le comunican las direcciones de los 3 ejes.  
Casos de aplicación: por ej. pinzas de soldadura, garras o boquillas de aplicación de pegamentos

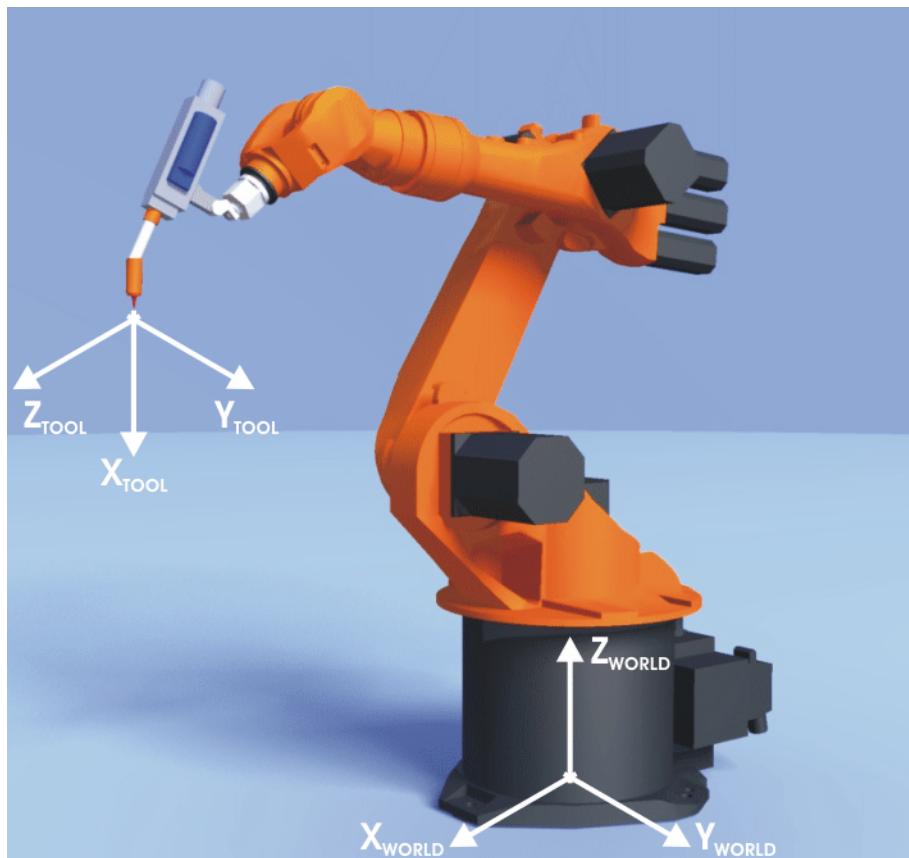


Fig. 3-25: Método ABC World

### Procedimiento del método ABC-World 5D

- Seleccionar la secuencia de menú **Puesta en servicio > Medición > Herramienta > ABC-World**.
- Introducir el número de herramienta. Confirmar con **OK**.
- En el campo **5D/6D** seleccionar una variante. Confirmar con **OK**.
- Si se ha seleccionado **5D**:  
Alinear  $+X_{TOOL}$  de forma paralela a  $-Z_{WORLD}$  ( $+X_{TOOL}$  = dirección de avance)  
Si se ha seleccionado **6D**:  
Alinear los ejes del sistema de coordenadas TOOL del siguiente modo:
  - $+X_{TOOL}$  de forma paralela a  $-Z_{WORLD}$  ( $+X_{TOOL}$  = dirección de avance)
  - $+Y_{TOOL}$  de forma paralela a  $+Y_{WORLD}$
  - $+Z_{TOOL}$  de forma paralela a  $+X_{WORLD}$
- Confirmar con **OK**.
- Pulsar **Guardar**.

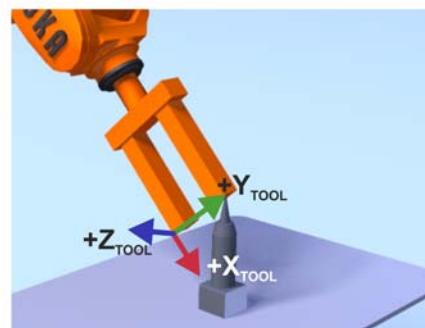
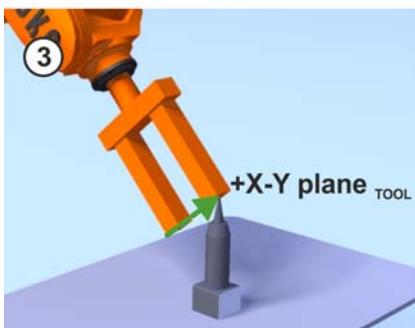
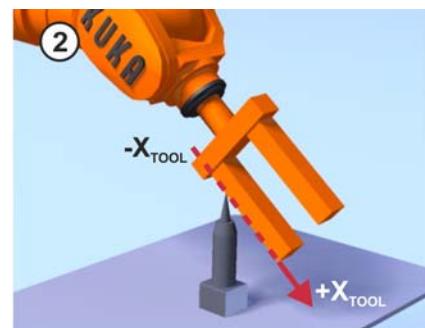
**Medición de la orientación, método ABC 2 puntos**

A la unidad de control del robot se le comunican los ejes del sistema de coordenadas de herramienta desplazando el robot a un punto del eje X y un punto en el plano XY.

Este método se utiliza cuando las direcciones de los ejes deben establecerse con la mayor exactitud posible.



El siguiente procedimiento es válido cuando la dirección de impacto de la herramienta es la dirección de impacto por defecto (= dirección X). Si la dirección de impacto se cambia a Y o Z, el procedimiento también debe cambiarse.



**Fig. 3-26: Método ABC 2-Puntos**

- Una condición es que el TCP ya se haya medido con un método XYZ.

2. Seleccionar en el menú principal la secuencia **Puesta en servicio > Medir > Herramienta > ABC 2 puntos**.
3. Indicar el número de la herramienta montada. Confirmar pulsando **Continuar**.
4. Con el TCP desplazarse a un punto de referencia cualquiera. Pulsar en **Medir**. Confirmar pulsando **Continuar**.
5. Desplazar la herramienta de tal modo que el punto de referencia sobre el eje X se encuentre sobre un punto de valor X negativo (es decir, en contra de la dirección de trabajo). Pulsar en **Medir**. Confirmar pulsando **Continuar**.
6. Desplazar la herramienta de modo tal que el punto de referencia sobre el plano XY se encuentre sobre un valor Y positivo. Pulsar en **Medir**. Confirmar pulsando **Continuar**.
7. O bien pulsar **Guardar**. Los datos se guardan y la ventana se cierra.  
O pulsar **Datos de carga**. Los datos se guardan y se abre una ventana en la que se pueden introducir los datos de carga.

**Advertencias de seguridad de las garras en el modo de formación**

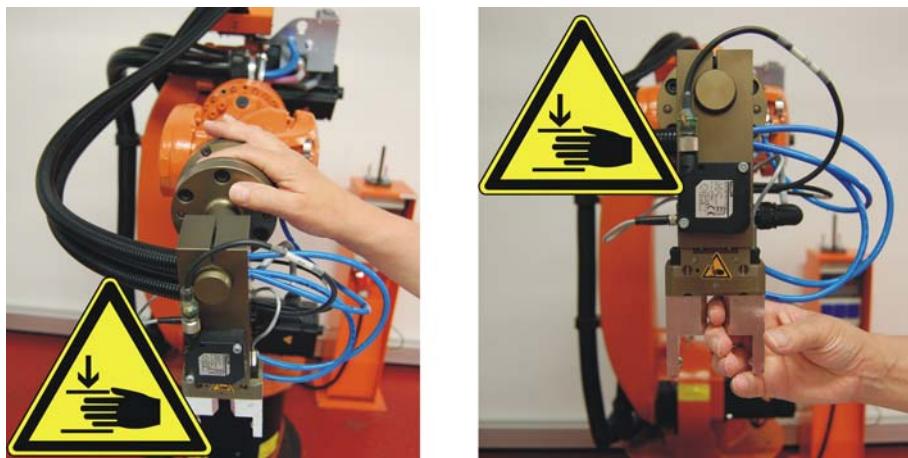


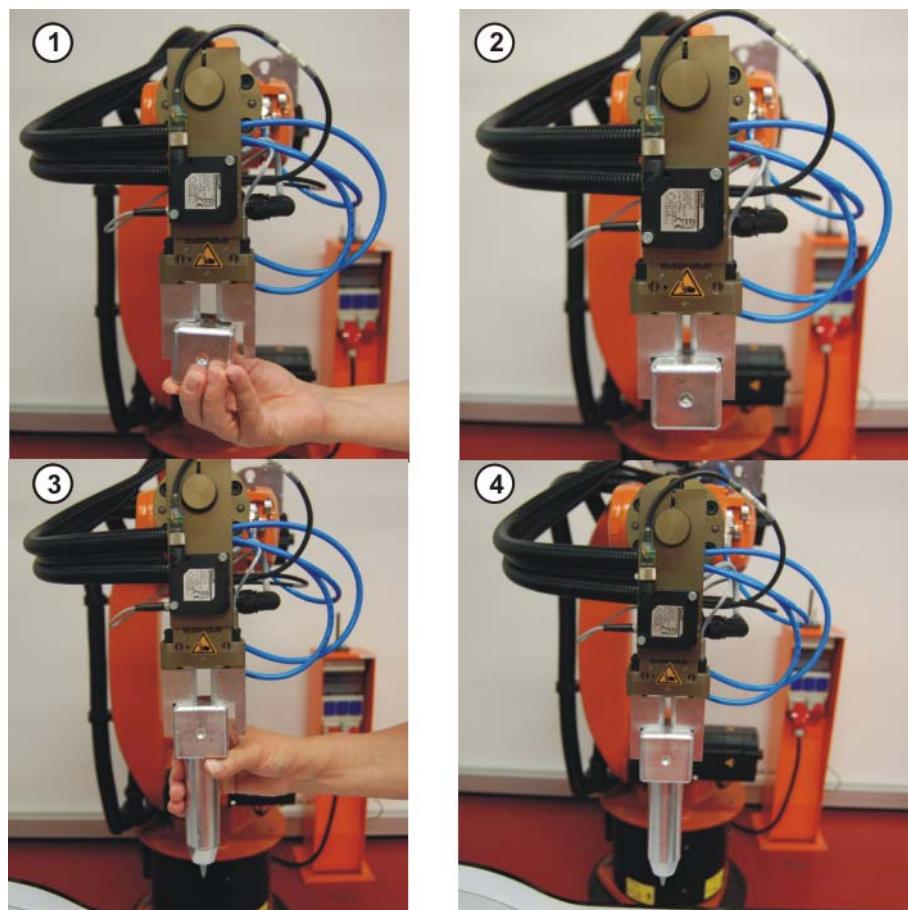
Fig. 3-27: Peligro de contusiones en la garra de formación

**⚠ ADVERTENCIA**

**Advertencia**

Al utilizar el sistema de garras existe riesgo de aplastamiento y corte. Aquellos que manejen las garras deben asegurarse de que no puedan quedar atrapados.

A la hora de sujetar los componentes (cubo, clavija) debe procederse con el máximo cuidado.



**Fig. 3-28: Amarrar objetos en la garra de formación**

Pos.	Observación
1	Sujeción del dado
2	Dado sujeto
3	Sujeción de una clavija
4	Clavija sujetada

**En caso de colisión se activa el seguro contra colisiones.**

La liberación del robot se produce tras activarse el seguro contra colisiones. Un operario pulsa el interruptor (1), al tiempo que mantiene cualquier parte del cuerpo alejada del robot, del seguro contra colisiones y de la garra. Un segundo operario se asegura, antes de la liberación del robot, de que no se pueda poner a ninguna persona en peligro por un movimiento del robot.

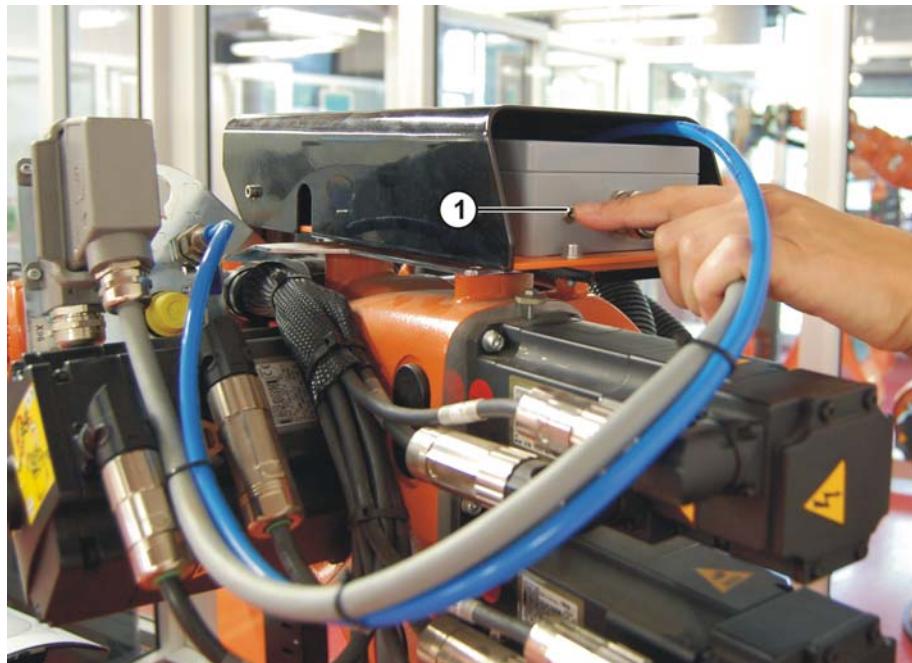


Fig. 3-29: Pulsador para la liberación del seguro contra colisiones

### 3.10.1 Ejercicio: Medición de la herramienta clavija

#### Objetivo del ejercicio

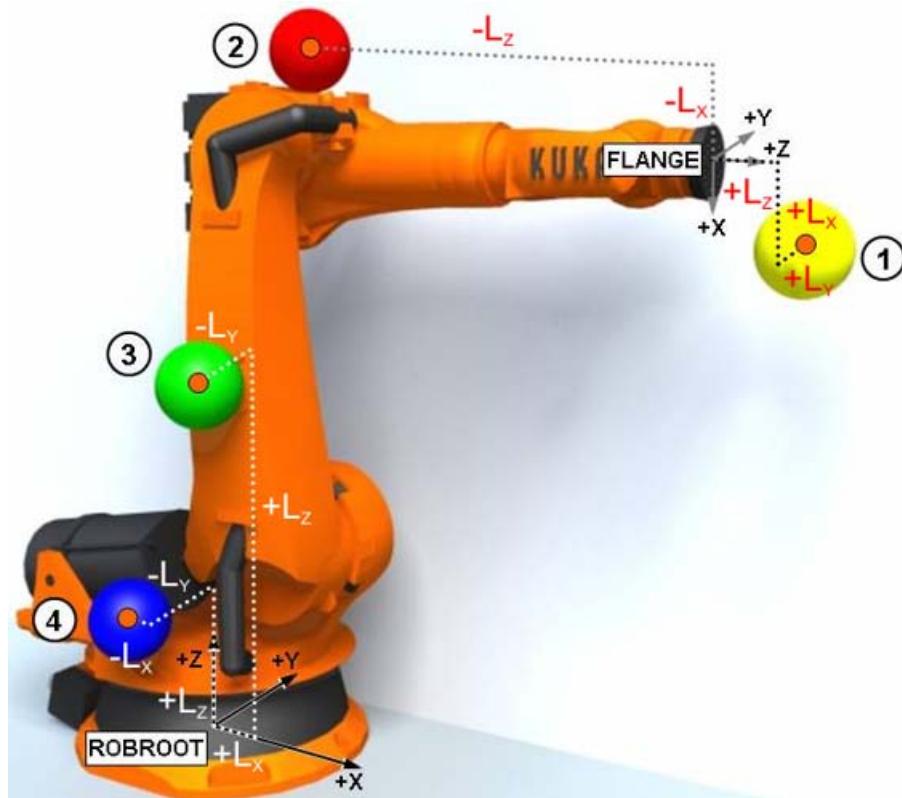
Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Medición de una herramienta con el método XYZ-4-puntos y el método ABC-World
- Activar la herramienta medida
- Desplazamiento en el sistema de coordenadas de la herramienta
- Desplazamiento en dirección de avance de la herramienta
- Reorientación de la herramienta alrededor del Tool Center Point (TCP)

#### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos de los diferentes métodos de medición del punto de trabajo de la herramienta, especialmente del método de 4 puntos XYZ.
- Conocimientos teóricos de los diversos métodos de medición de la orientación de la herramienta, especialmente del método ABC-World
- Conocimientos teóricos de los datos de carga de robot y de su correspondiente entrada



- |                                  |                                  |
|----------------------------------|----------------------------------|
| 1 Carga                          | 3 Carga adicional sobre el eje 2 |
| 2 Carga adicional sobre el eje 3 | 4 Carga adicional sobre el eje 1 |

#### Formulación de tarea

Ejecutar las siguientes tareas: Medición clavija

1. Medir el TCP de la clavija mediante el método de 4 puntos XYZ. Utilizar como punta de referencia la punta de metal negro. Sacar la clavija superior del depósito de clavijas y sujetarla en la garra. Utilizar el **número de herramienta 2** y darle el nombre de **Clavija1**.
2. Asegurar los datos de herramienta.
3. Medir la orientación de herramienta mediante el método ABC-World-5D.

4. La tolerancia no debería ser mayor de 0,95 mm. En la práctica, este valor no es suficiente. Lo mejor sería alcanzar tolerancias de 0,5mm o, incluso, 0,3mm.
5. Introducir los datos de carga para la garra con clavija como herramienta número 2.  
Usar para ello la garra con clavija correcta (ver más abajo).
6. Guardar los datos de herramienta y comprobar el desplazamiento con la clavija en el sistema de coordenadas de herramienta

**Datos de carga de la herramienta garra con clavija**

■ Garra de formación KR 16

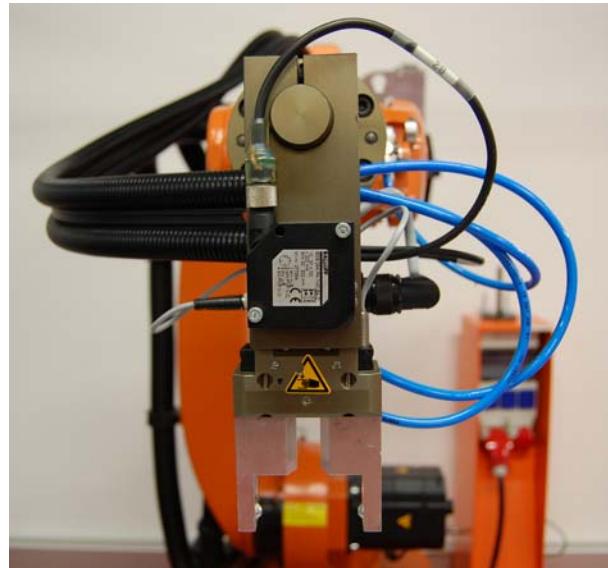


Fig. 3-30: Garra de formación KR 16

<b>Masa:</b>		
$M = 4,9 \text{ kg}$		
<b>Centro de masa:</b>		
$X = 53 \text{ mm}$	$Y = 49 \text{ mm}$	$Z = 65 \text{ mm}$
<b>Orientación:</b>		
$A = 0^\circ$	$B = 0^\circ$	$C = 0^\circ$
<b>Momentos de inercia:</b>		
$J_X = 0,02 \text{ kgm}^2$	$J_Y = 0,03 \text{ kgm}^2$	$J_Z = 0,15 \text{ kgm}^2$

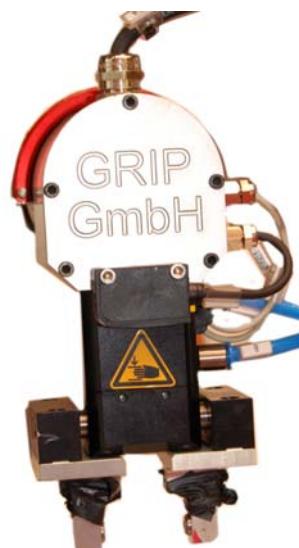
■ Garra de formación celda modular



Fig. 3-31: Garra de formación celda modular

<b>Masa:</b>		
M = 5 kg		
<b>Centro de masa:</b>		
X = 23 mm	Y = -38 mm	Z = 84 mm
<b>Orientación:</b>		
A = 0°	B = 0°	C = 0°
<b>Momentos de inercia:</b>		
J <sub>X</sub> = 0,02 kgm <sup>2</sup>	J <sub>Y</sub> = 0,07 kgm <sup>2</sup>	J <sub>Z</sub> = 0,14 kgm <sup>2</sup>

- Garra de formación celda móvil



**Fig. 3-32: Garra de formación celda móvil**

<b>Masa:</b>		
M = 2,14 kg		
<b>Centro de masa:</b>		
X = 30 mm	Y = 0 mm	Z = 64 mm
<b>Orientación:</b>		
A = 0°	B = 0°	C = 0°
<b>Momentos de inercia:</b>		
J <sub>X</sub> = 0,002 kgm <sup>2</sup>	J <sub>Y</sub> = 0,004 kgm <sup>2</sup>	J <sub>Z</sub> = 0,003 kgm <sup>2</sup>

#### **Lo que se debe saber tras el ejercicio:**

1. ¿Qué ventajas ofrece la medición de herramienta?

.....  
.....  
.....

2. Cuáles son los métodos de la medición de la herramienta?

.....  
.....  
.....

3. Cuáles puntos se determinan con el método XYZ 4-Puntos?

.....  
.....

### 3.10.2 Ejercicio: Medición de la herramienta garra, método de 2 puntos

**Objetivo del ejercicio** Despues de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Medición de una herramienta con ayuda del método XYZ de 4 puntos y el método ABC de 2 puntos
- Activar la herramienta medida
- Desplazamiento en el sistema de coordenadas de la herramienta
- Desplazamiento en dirección de avance de la herramienta
- Reorientación de la herramienta alrededor del Tool Center Point (TCP)

**Requisitos** Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos de los diversos métodos de medición del punto de trabajo de la herramienta, especialmente acerca del método de 2 puntos
- Conocimientos teóricos de los datos de carga de robot y la entrada correspondiente

**Formulación de tarea** Ejecutar las siguientes tareas

1. Usar para la medición de herramienta de la garra el nombre "Garra\_nueva" y el número de herramienta "3".
2. Medir el TCP de la garra con ayuda del método XYZ de 4 puntos como se indica en la figura:

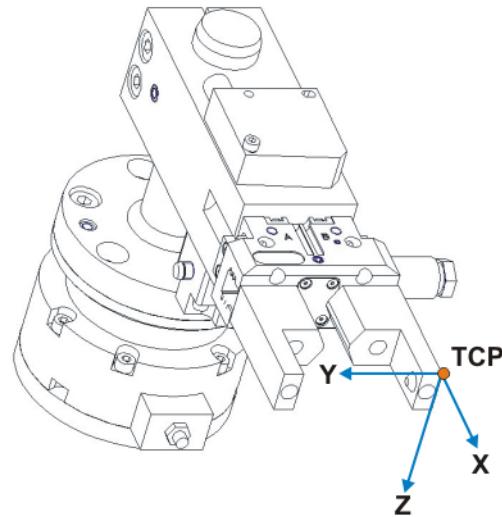
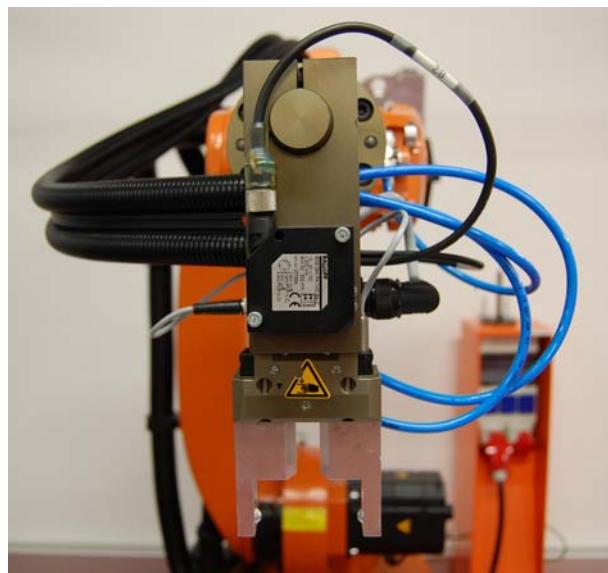


Fig. 3-33: College garra: Posición del TCP

3. Medir la orientación del sistema de coordenadas de garra con ayuda del método ABC de 2 puntos.
4. Introducir los datos de carga para la garra de la herramienta número 3. Usar para ello la garra correcta (ver más abajo).
5. Guardar los datos de herramienta y comprobar el proceso manual con la garra en el sistema de coordenadas de herramienta.

**Datos de carga de la herramienta - garra**

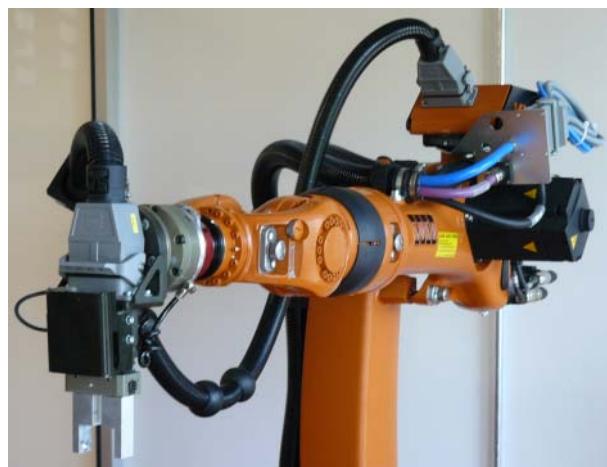
- Garra de formación KR 16



**Fig. 3-34: Garra de formación KR 16**

<b>Masa:</b>		
$M = 4,3 \text{ kg}$		
<b>Centro de masa:</b>		
$X = 41 \text{ mm}$	$Y = 35 \text{ mm}$	$Z = 62 \text{ mm}$
<b>Orientación:</b>		
$A = 0^\circ$	$B = 0^\circ$	$C = 0^\circ$
<b>Momentos de inercia:</b>		
$J_X = 0,03 \text{ kgm}^2$	$J_Y = 0,04 \text{ kgm}^2$	$J_Z = 0,12 \text{ kgm}^2$

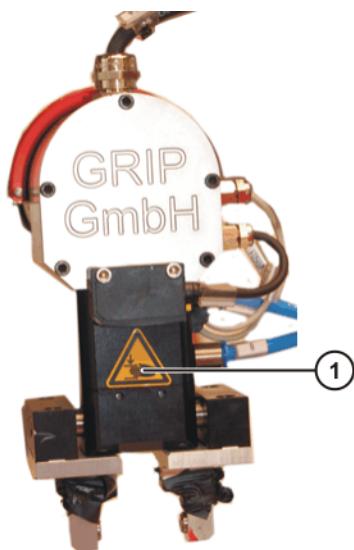
- Garra de formación celda modular



**Fig. 3-35: Garra de formación celda modular**

<b>Masa:</b>		
$M = 4,4 \text{ kg}$		
<b>Centro de masa:</b>		
$X = 7 \text{ mm}$	$Y = -24 \text{ mm}$	$Z = 80 \text{ mm}$
<b>Orientación:</b>		
$A = 0^\circ$	$B = 0^\circ$	$C = 0^\circ$
<b>Momentos de inercia:</b>		
$J_X = 0,02 \text{ kgm}^2$	$J_Y = 0,05 \text{ kgm}^2$	$J_Z = 0,11 \text{ kgm}^2$

- Garra de formación celda móvil



**Fig. 3-36: Garra placa**

<b>Masa:</b>		
$M = 2 \text{ kg}$		
<b>Centro de masa:</b>		
$X = 23 \text{ mm}$	$Y = 0 \text{ mm}$	$Z = 61 \text{ mm}$
<b>Orientación:</b>		
$A = 0^\circ$	$B = 0^\circ$	$C = 0^\circ$
<b>Momentos de inercia:</b>		
$J_X = 0,002 \text{ kgm}^2$	$J_Y = 0,004 \text{ kgm}^2$	$J_Z = 0,003 \text{ kgm}^2$

#### Tarea alternativa

De forma alternativa, la garra también se puede medir con una entrada numérica:

Ejecutar las siguientes tareas

1. Usar para la medición de herramienta de la garra el nombre "Garra\_nueva" y el número de herramienta "3".
2. Medir el TCP de la garra con la ayuda de la entrada numérica:

Garra de formación KR 16					
X	Y	Z	A	B	C
122,64 mm	177,73 mm	172,49 mm	45°	0°	180°

Garra de formación celda modular					
X	Y	Z	A	B	C
175,38 mm	-123,97 mm	172,71 mm	-45°	0°	-180°

Garra de formación celda móvil					
X	Y	Z	A	B	C
12,0 mm	0 mm	77,0 mm	0°	0°	0°

#### Lo que se debe saber tras el ejercicio:

1. Cuál pictograma representa el sistema de coordenadas de la herramienta?

a)



b)



c)



d)



2. Cuántos herramientas máx. puede administrar la unidad de control?

.....

3. Qué significa el valor -1 en los datos de carga de la herramienta?

.....

### 3.11 Mover el robot en el sistema de coordenadas de base

Movimiento en el sistema de coordenadas de base

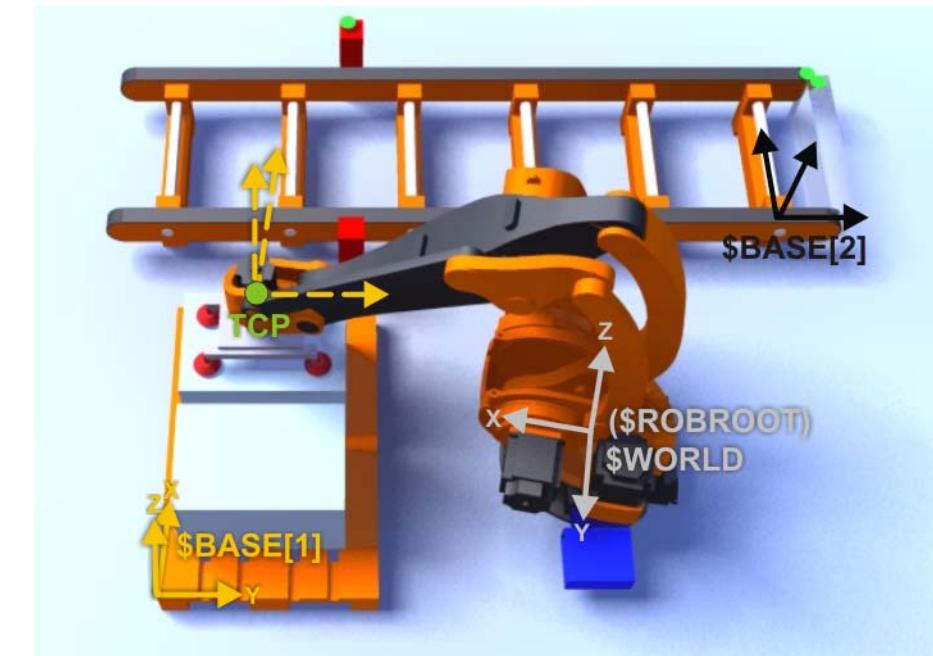


Fig. 3-37: Procesos manuales en el sistema de coordenadas de base

#### Descripción de base

- La herramienta del robot puede moverse de acuerdo con las direcciones del sistema de coordenadas de base. Los sistemas de coordenadas de base se pueden medir individualmente y a menudo están orientados a lo largo de los cantos de la pieza, los alojamientos de la pieza o los palets. Por ello los procesos manuales son muy cómodos.
- **Todos** los ejes del robot necesarios se mueven. El mismo sistema decide los ejes que deben moverse dependiendo del movimiento.
- Para ello se utilizan las teclas de desplazamiento o el Space Mouse del KUKA smartPAD.
- Hay 32 sistemas de coordenadas de base disponibles.
- La velocidad puede cambiarse (override manual: HOV).
- Los procesos manuales sólo se pueden efectuar en el modo de servicio T1.
- Debe estar pulsado el pulsador de validación.

Principio del proceso manual de base

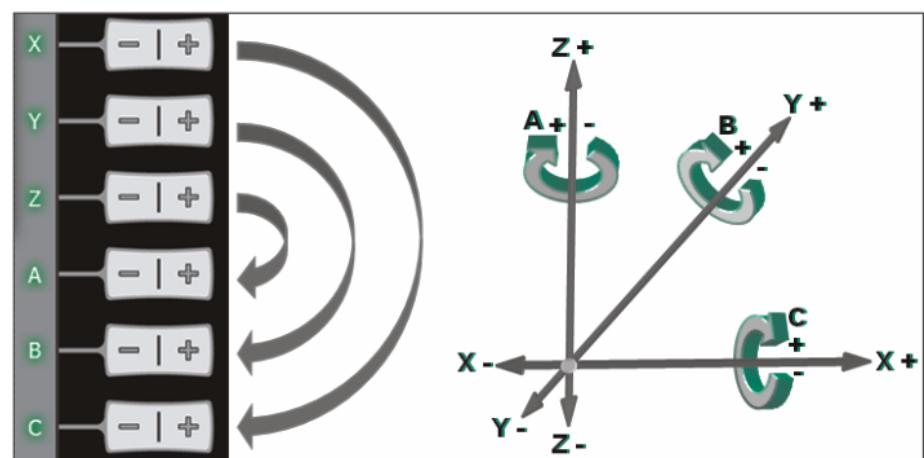


Fig. 3-38: Sistema de coordenadas cartesiano

**Un robot se puede mover de dos maneras distintas en un sistema de coordenadas:**

- Traslatoria (rectilínea) a lo largo de las direcciones de orientación del sistema de coordenadas: X, Y, Z
- Rotacional (giratoria / viratoria) en la dirección de orientación del sistema de coordenadas: ángulos A, B y C

En un comando de desplazamiento (p. ej., al pulsar una tecla de desplazamiento), el control primero calcula una distancia. El punto de inicio del recorrido es el punto de referencia de la herramienta (TCP). La dirección del recorrido lo fija el sistema de coordenadas universales. El control entonces regula todos los ejes de modo que la herramienta se conduce por este recorrido (traslación) o se gira (rotación) en este recorrido.

#### **Uso del sistema de coordenadas base:**

- El movimiento del robot siempre es predecible mientras se conozca el sistema de coordenadas de base.
- En este caso el Space Mouse también permite un manejo intuitivo del equipo. Como requisito es necesario que el operador esté colocado correctamente con respecto al robot o al sistema de coordenadas de base. Para ello, seleccionar el eje X como eje de referencia en caso necesario.
- Permite el desplazamiento paralelo, p. ej. hacia una pieza o hacia el borde de un palé.



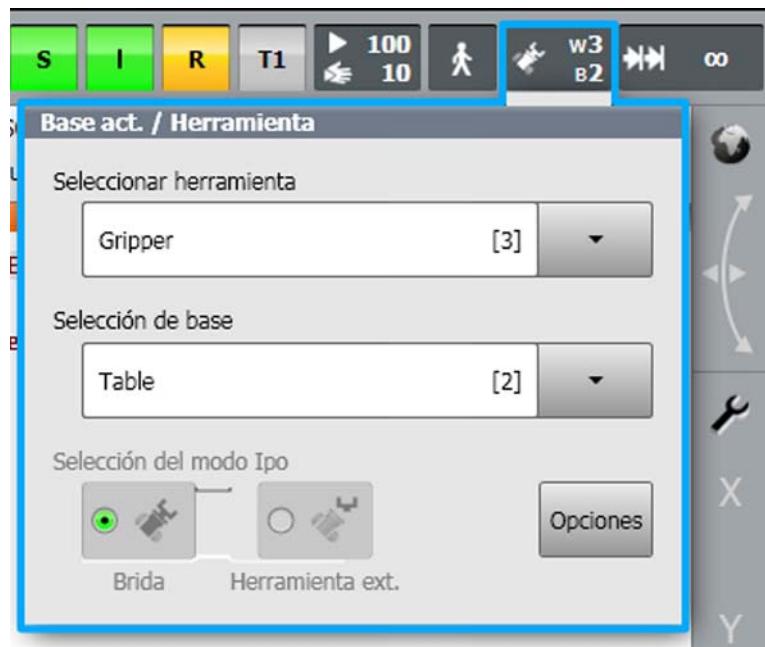
Si además se ajusta el sistema de coordenadas de herramienta correcto, en el sistema de coordenadas de base se puede reorientar con relación al TCP.

#### **Procedimiento**

1. Como alternativa a las teclas de desplazamiento, seleccionar **Base**.



2. Seleccionar herramienta y base



3. Ajustar el override manual



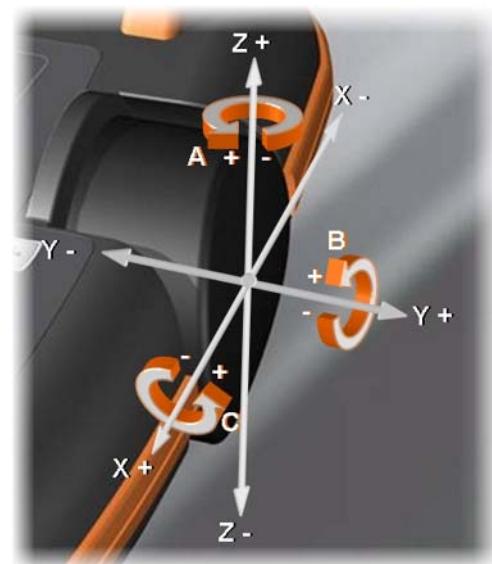
4. Pulsar el pulsador de validación en la posición intermedia y mantenerlo pulsado.



5. Desplazar en la dirección deseada con las teclas de desplazamiento



6. Otra alternativa consiste en desplazar con el Space Mouse



### 3.11.1 Ejercicio: Procesos manuales en el sistema de coordenadas de base

#### Objetivo del ejercicio

Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Desplazamiento manual del robot en el sistema de coordenadas de la herramienta, con las teclas de desplazamiento y el Space Mouse
- Desplazamiento manual a lo largo de cantos de pieza predefinidos

#### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Participación en las instrucciones de seguridad



#### Aviso

Antes de comenzar el ejercicio se debe participar en las instrucciones de seguridad con documentación correspondiente.

- Conocimientos teóricos del desplazamiento en el sistema de coordenadas de la pieza

#### Formulación de tarea

Ejecutar las siguientes tareas:

1. Desbloquear y confirmar la PARADA DE EMERGENCIA
2. Asegurarse de que esté ajustado el modo de servicio T1
3. Activar el sistema de coordenadas base *D\_Red Base Straight*
4. Sujetar la clavija en la garra y seleccionar *D\_Pen165* como sistema de coordenadas de la herramienta.
5. Desplazar el robot en el sistema de coordenadas base, con distintos ajustes de override manual (HOV), mediante las teclas de desplazamiento manual y el Space Mouse
6. Mover la clavija a lo largo del contorno exterior sobre la mesa de trabajo

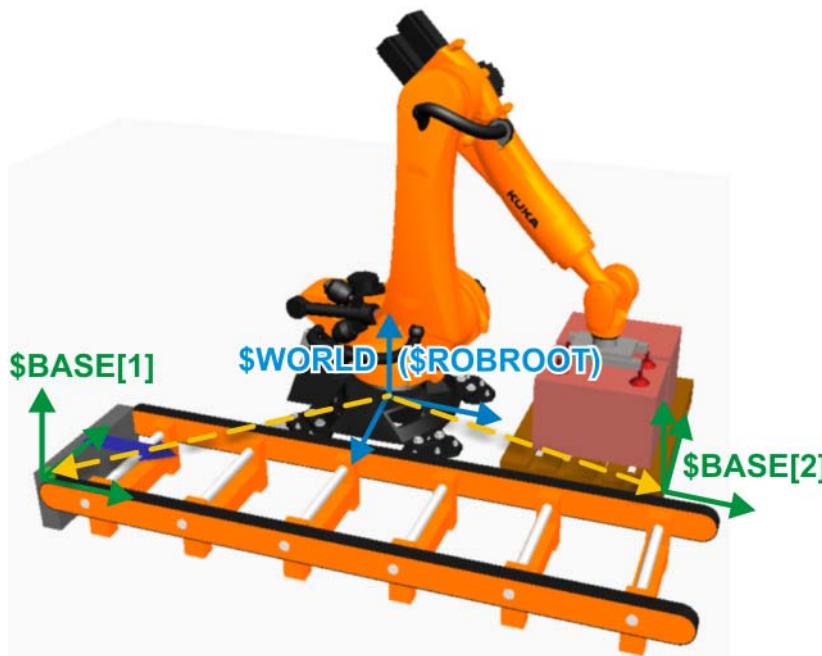
### 3.12 Medición de una base

#### Descripción

Medir una base significa crear un sistema de coordenadas en un determinado punto del entorno del robot a partir del sistema universal de coordenadas. El objetivo consiste en aplicar los movimientos manuales y las posiciones programadas del robot a este sistema de coordenadas. Los cantos definidos de alojamientos de piezas, compartimentos, palés o máquinas son, por tanto, puntos de referencia muy útiles para un sistema de coordenadas base.

La medición de una base se realiza en dos pasos:

1. Determinación del origen de las coordenadas
2. Definición de la dirección de las coordenadas

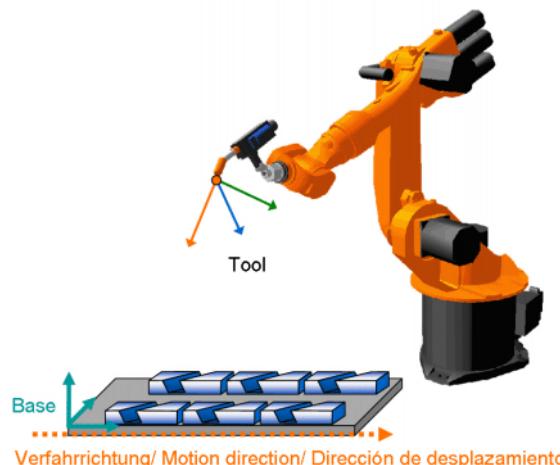


**Fig. 3-39: Medición de base**

#### Ventajas

Una vez medida con éxito una base, se dispone de las siguientes ventajas:

- Desplazamiento a lo largo de los cantos de la pieza:  
El TCP se puede mover de forma manual a lo largo de los cantos de la superficie de trabajo o de la pieza.



**Fig. 3-40: Ventajas de la medición BASE: Dirección de desplazamiento**

- Sistema de coordenadas de referencia:  
Los puntos aprendidos hacen referencia al sistema de coordenadas seleccionado.

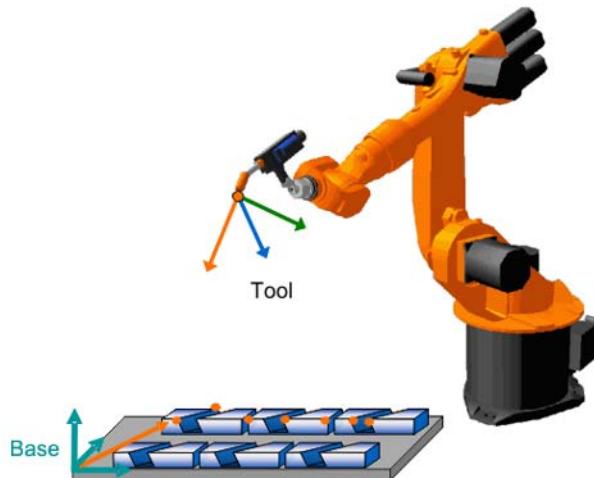


Fig. 3-41: Ventajas de la medición BASE: Referencia al sistema de coordenadas deseado

- Corrección / corrimiento del sistema de coordenadas:  
Puntos pueden ser programados por aprendizaje en relación a la base. Si la base debe ser desplazada, p. ej., porque la superficie de trabajo ha sido desplazada, los puntos se desplazan también y no tienen que ser programados nuevamente.

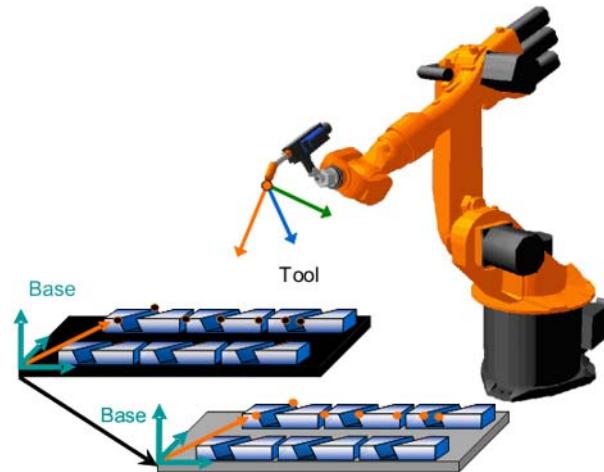
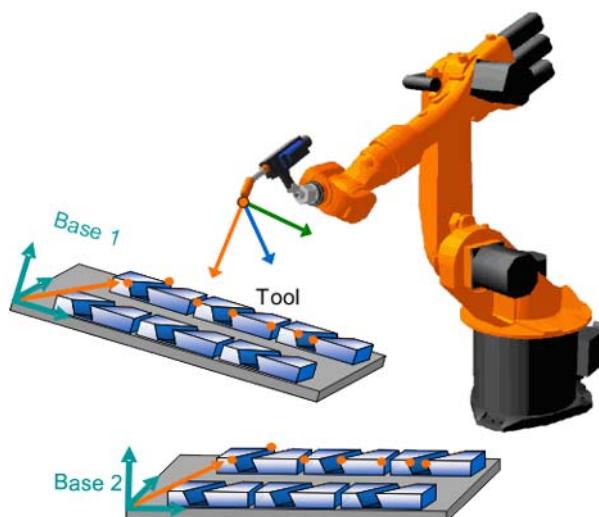


Fig. 3-42: Ventajas de la medición BASE: Corrimiento del sistema de coordenadas de base

- Utilización de varios sistemas de coordenadas de base:  
Se pueden generar hasta 32 sistemas de coordenadas diferentes y utilizarlos de acuerdo con el paso de programa.



**Fig. 3-43: Ventajas de la medición BASE: Utilización de varios sistemas de coordenadas de base**

### Opciones de la medición de base

Para la medición de base se dispone de los métodos siguientes:

Métodos	Descripción
<b>Método de 3 puntos</b>	1. Definición del origen 2. Definición de la dirección X positiva 3. Definición de la dirección Y positiva (plano XY)
<b>Método indirecto</b>	El método indirecto se utiliza cuando no se puede llegar con el robot al origen de la base, por ej. porque se encuentra en el interior de una pieza o fuera del campo de trabajo del robot.  Se realiza el desplazamiento a 4 puntos con referencia conocida a la base que se va a medir, cuyas coordenadas deben ser conocidas (datos CAD). La unidad de control del robot calcula la base utilizando estos puntos.
<b>Entrada numérica</b>	Entrada directa de valores para la distancia al sistema de coordenadas universales (X, Y, Z) y del giro (A, B, C).

### Procedimiento del método de 3 puntos

**i** La medición de base solo puede realizarse con una herramienta previamente medida (el TCP se debe conocer).

1. En el menú principal, seleccionar **Puesta en servicio > Medir > Base > 3 puntos**.
2. Introducir un número y un nombre para la base. Confirmar pulsando **Continuar**.
3. Introducir el número de herramienta cuyo TCP se utilice para medir la base. Confirmar pulsando **Continuar**.
4. Con el TCP mover el robot a la nueva base. Pulsar la tecla **Medir** y confirmar la posición pulsando **Sí**.

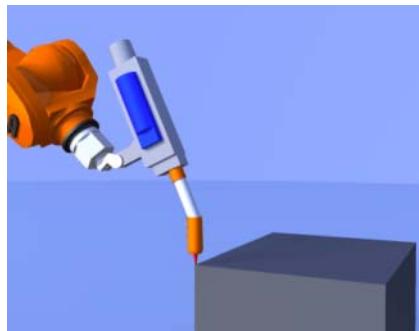


Fig. 3-44: Primer punto: Origen

5. Con el TCP desplazar el robot a un punto del eje X positivo de la nueva base. Pulsar **Medir** y confirmar la posición pulsando **Sí**.

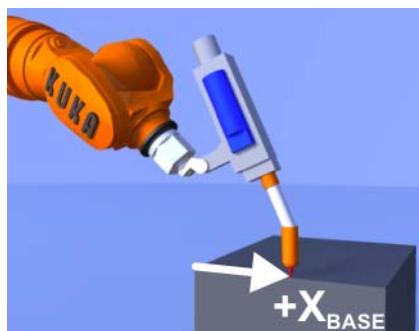


Fig. 3-45: Segundo punto: dirección X

6. Con el TCP desplazar el robot a un punto del plano XY con valor Y positivo Pulsar **Medir** y confirmar la posición pulsando **Sí**.

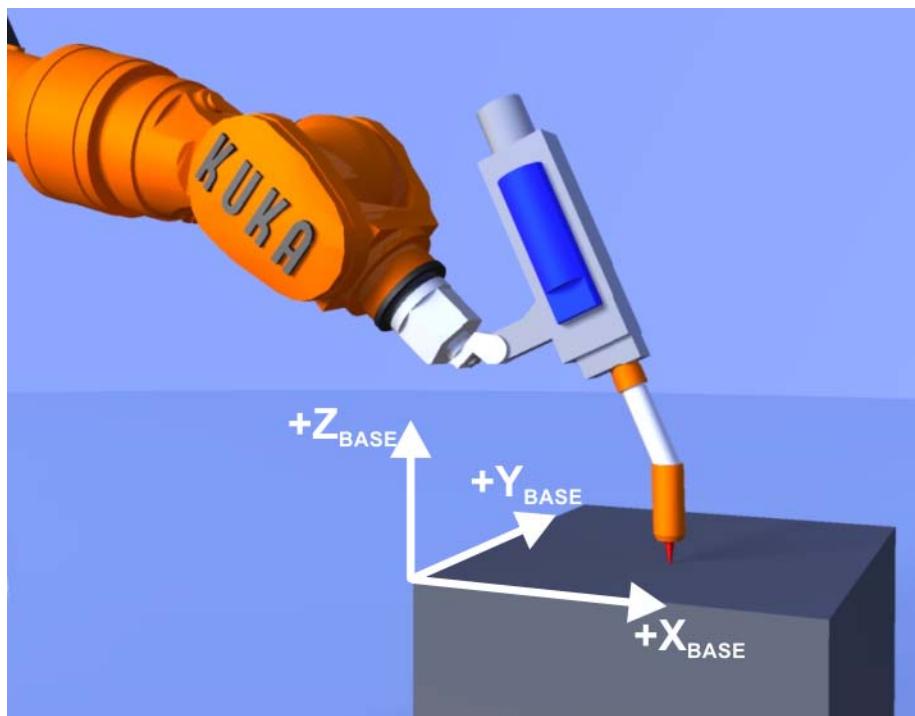


Fig. 3-46: Tercer punto: plano XY

7. Pulsar **Guardar**.
8. Cerrar el menú



Los tres puntos de medición no deben encontrarse en una línea recta. Debe existir un mínimo de ángulo entre los puntos (ajustes estandar 2,5°).

### 3.12.1 Ejercicio: Medición de la base mesa, método de 3 puntos

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"><li>■ Determinación de una base cualquiera</li><li>■ Medición de una base</li><li>■ Activación de una base medida para un movimiento manual</li><li>■ Movimiento en el sistema de coordenadas base</li></ul>
<b>Requisitos</b>	Los siguientes requisitos son necesarios para completar este ejercicio correctamente: <ul style="list-style-type: none"><li>■ Conocimientos teóricos de los métodos para la medición de base, especialmente acerca del método de 3 puntos</li></ul>
<b>Formulación de tarea</b>	Ejecutar las siguientes tareas: <ol style="list-style-type: none"><li>1. Medir la base azul sobre la mesa con el método de 3 puntos. Asignar el <b>número de base 1</b> con la denominación <b>azul</b>. Utilizar la clavija 1 ya medida (número de herramienta 2) como herramienta de medición.</li><li>2. Guardar los datos de la base medida.</li><li>3. Medir la base roja sobre la mesa con el método de 3 puntos. Asignar el <b>número de base 2</b> con la denominación <b>roja</b>. Utilizar la clavija 1 ya medida (número de herramienta 2) como herramienta de medición.</li><li>4. Guardar los datos de la base medida.</li><li>5. Desplazar la herramienta al origen del sistema de coordenadas de base azul y mostrar la posición cartesiana real.</li></ol>

X                    Y                    Z                    A                    B                    C

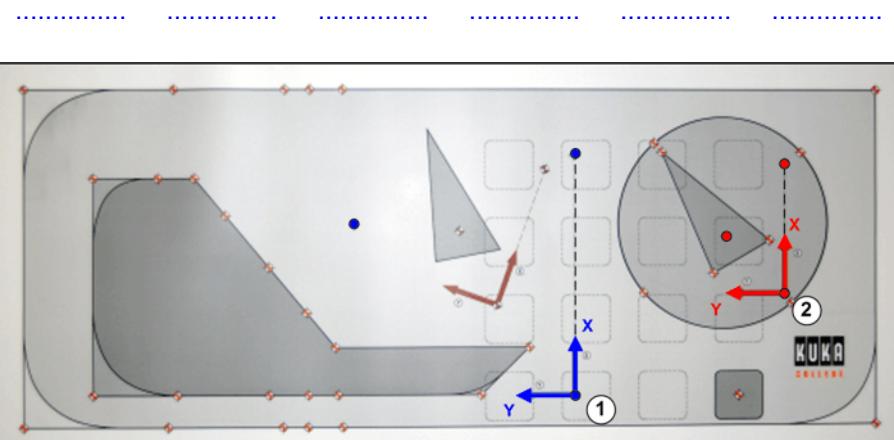


Fig. 3-47: Medición de base sobre la mesa

Lo que se debe saber tras el ejercicio:

1. ¿Porqué se debe medir una base?

2. ¿Qué pictograma representa el sistema de coordenadas base?

a)



b)



c)



d)



3. ¿Cuáles son métodos de la medición de base que existen?

.....  
.....  
.....

4. ¿Cuántos sistemas de base como máximo puede administrar la unidad de control?

.....  
.....  
.....

5. Describir la medición basada en el método de 3 puntos.

.....  
.....  
.....

### 3.13 Consulta de la posición actual del robot

#### Opciones de visualización de posiciones de robot

La posición actual del robot se puede mostrar de dos formas distintas:

- **Específica del eje:**

`$AXIS_ACT={A1...,A2...,A3...,A4...,A5...,A6...,E1.....,E6...}`

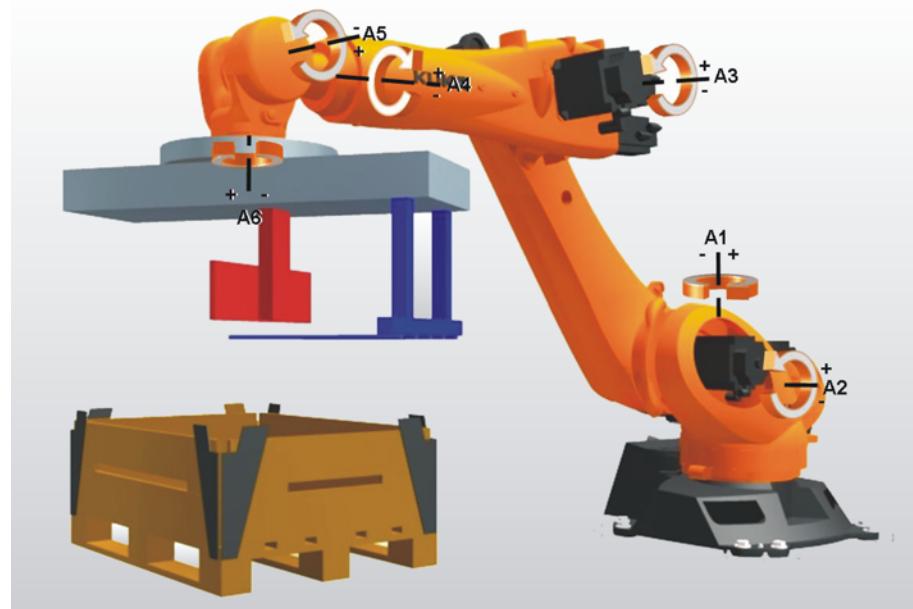
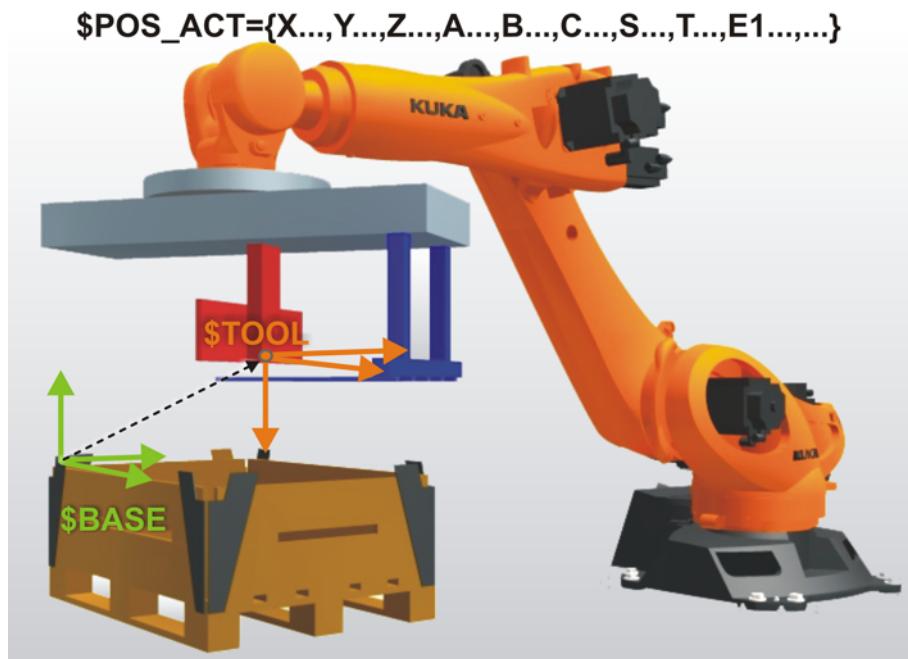


Fig. 3-48: Posición del robot específica del eje

Se muestra el ángulo de cada eje: equivale al valor angular absoluto partiendo de la posición cero del eje.

- **Cartesiano:**



**Fig. 3-49: Posición cartesiana**

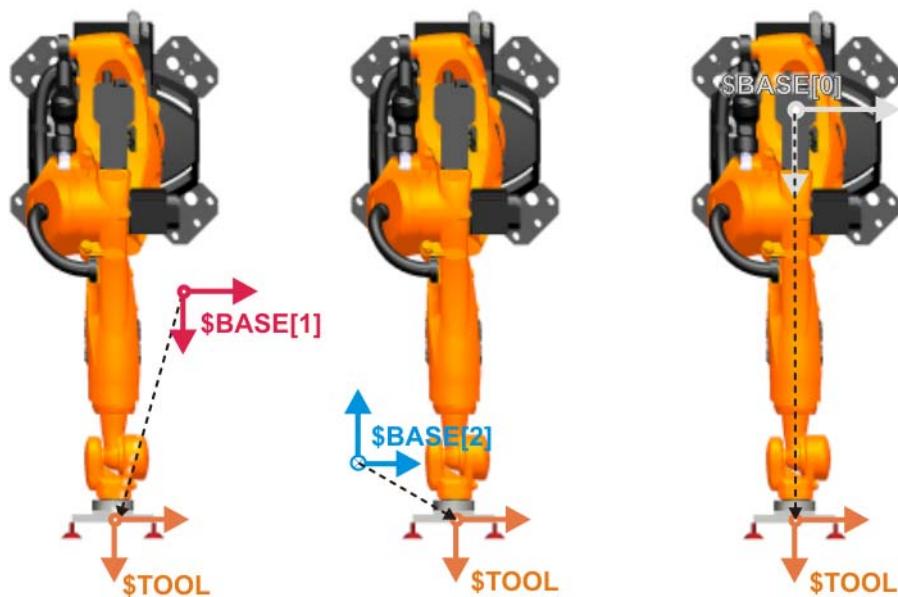
Se muestra la posición actual del TCP actual (sistema de coordenadas de herramienta) con relación al sistema de coordenadas de base actualmente seleccionado.

Si no hay ningún sistema de coordenadas de herramientas seleccionado, se aplica el sistema de coordenadas de brida.

Si no hay ningún sistema de coordenadas de base seleccionado, se aplica el sistema de coordenadas universales.

**Posición cartesiana con distintos sistemas de coordenadas de base**

Si se observa la figura inferior, se aprecia enseguida que el robot ocupa tres veces la misma posición. La indicación de posición, no obstante, muestra valores distintos en cada uno de estos tres casos:



**Fig. 3-50: Tres posiciones de herramientas, una posición de robot**

La posición del sistema de coordenadas de herramienta / TCP se muestra en el sistema de coordenadas base correspondiente:

- para la base 1

- para la base **2**
- para la base **0** corresponde al sistema de coordenadas del pie del robot  
(en la mayoría de los casos también el sistema de coordenadas universales)



Fig. 3-51: Seleccionar Tool y Base

Solo cuando se seleccionan la base y la herramienta correctas, el indicador de la posición cartesiana actual muestra valores razonables.

#### Consultar la posición del robot

##### Procedimiento:

- En el menú, seleccionar **Indicador > Posición real**. Se visualiza la posición real cartesiana.
- Pulsar **Específico del eje** para visualizar la posición real específica del eje.
- Pulsar **Cartesiano** para volver a visualizar la posición real cartesiana.

## 4 Ejecutar los programas del robot

### 4.1 Vista general

**Esta unidad incluye los siguientes contenidos:**

- Realizar el desplazamiento de inicialización
- Seleccionar e iniciar programas del robot

### 4.2 Realizar el desplazamiento de inicialización

**Desplazamiento COI** El desplazamiento de inicialización de un robot KUKA recibe el nombre de desplazamiento COI.

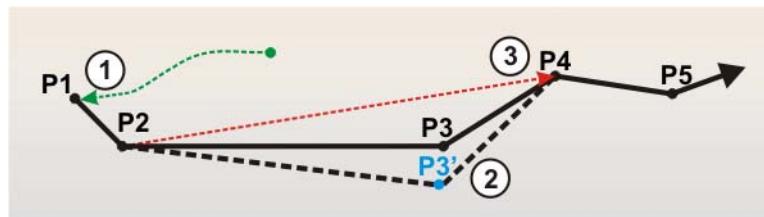


COI significa **Coincidencia de pasos**. La coincidencia significa "conformidad" y "coincidencia de sucesos temporales o espaciales".

**El sistema efectúa un desplazamiento COI en los casos siguientes:**

- Seleccionar un programa
- Resetear programa (restablecer)
- Desplazamiento manual durante el modo de programación
- Modificación en el programa
- Selección de paso

**Ejemplos de realización de un desplazamiento COI:**



**Fig. 4-1: Ejemplos de motivos para un desplazamiento COI**

- 1 Desplazamiento COI hasta la posición HOME tras preseleccionar o resetear un programa
- 2 Desplazamiento COI tras modificar una instrucción de movimiento: Borrar, programar por aprendizaje el punto, etc.
- 3 Desplazamiento COI tras seleccionar paso

**Motivos para un desplazamiento COI**

Un desplazamiento COI es necesario para hacer coincidir la posición actual del robot con las coordenadas del punto actual del programa.

Hasta que la posición actual de robot no coincide con una posición programada, no se puede realizar el cálculo de trayectoria. En primer lugar siempre se debe llevar el TCP a la trayectoria.

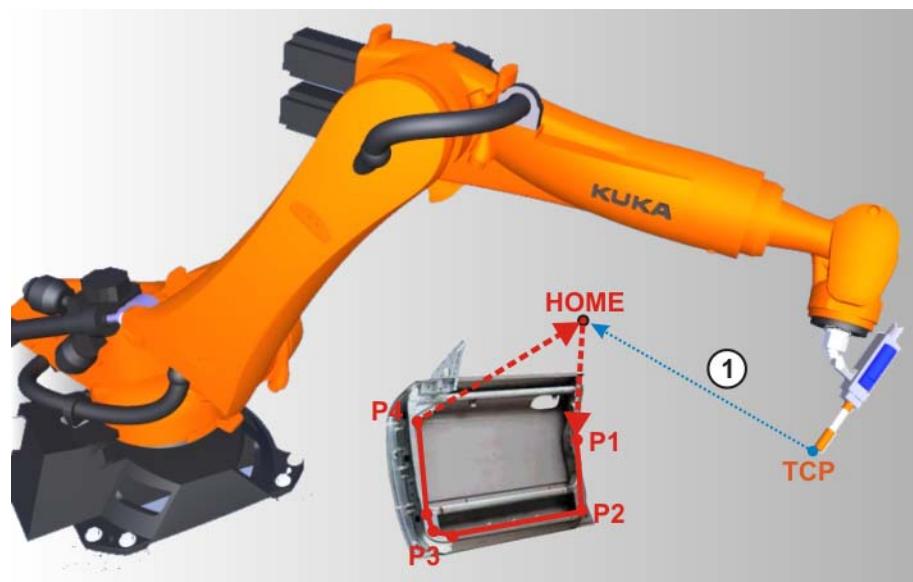


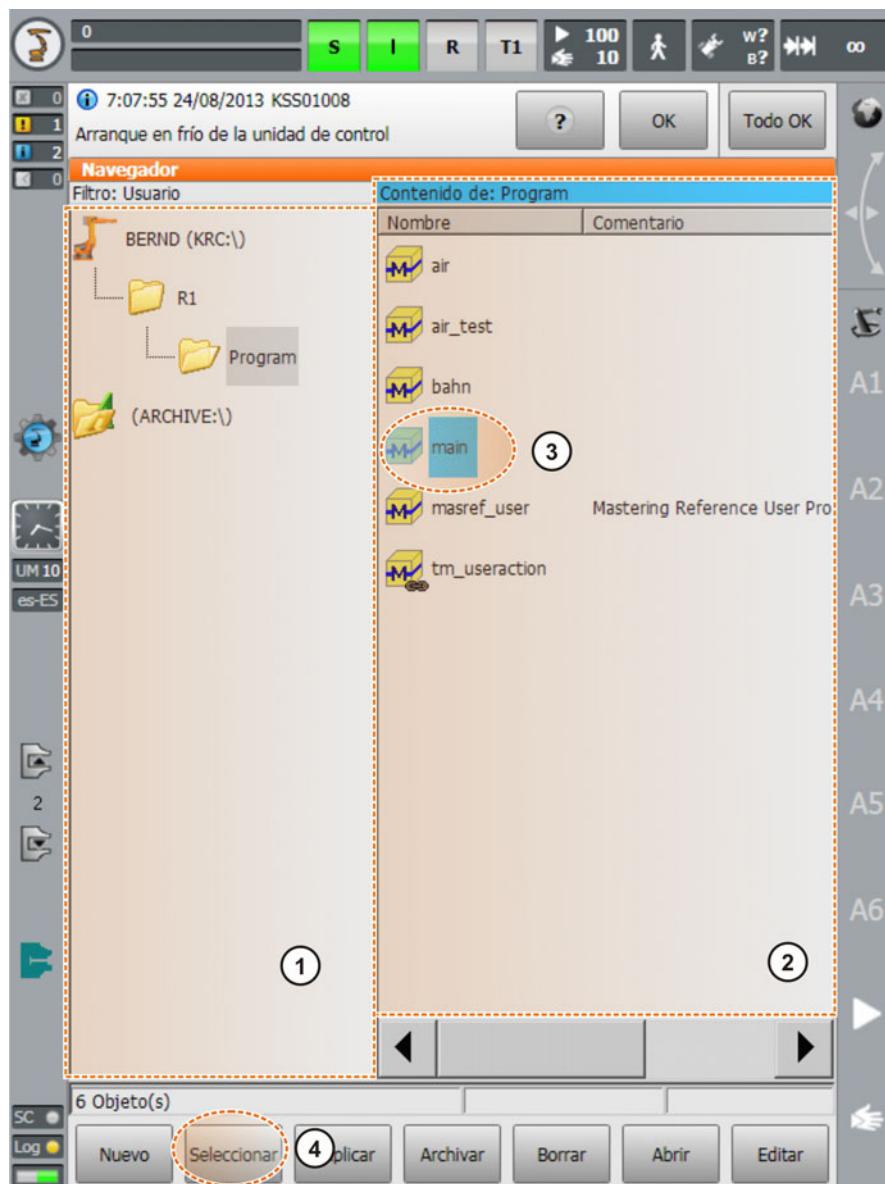
Fig. 4-2: Ejemplo de un desplazamiento COI en selección de programa

- 1 Desplazamiento COI hasta la posición HOME tras preseleccionar o resetear un programa

#### 4.3 Seleccionar e iniciar programas del robot

##### Seleccionar e iniciar programas de robots

Si se debe ejecutar un programa, hay que seleccionarlo. Los programas de robot se encuentran disponibles en el navegador de la interfaz de usuario. Normalmente los programas de desplazamiento encuentran en carpetas. El programa Cell (programa de administración para el control del robot desde un PLC) está siempre en la carpeta "R1".



**Fig. 4-3: Navegador**

- 1 Navegador: Estructura del directorio/unidad
- 2 Navegador: Lista de directorios/datos
- 3 Programa seleccionado
- 4 Botón para seleccionar un programa

Para iniciar un programa se dispone de las teclas de inicio Adelante y Atrás .

**i** Solo es posible ejecutar una programa hacia atrás con la tecla , si se ha ejecutado la líneaINI de un programa y el robot se ha desplazado hacia delante por la trayectoria inmediatamente antes. Sin embargo, si se desplaza manualmente el robot, p. ej. tras una parada del programa, el desplazamiento hacia atrás ya no es posible.



**Fig. 4-4: Direcciones de ejecución de programa: Adelante/atrás**

Pos.	Descripción
1	Tecla de arranque <b>Avance</b>
2	Tecla de arranque <b>Retroceso</b>

Para ejecutar en un programa existen varios **modos de ejecución** para el movimiento programado del robot:

	<b>IR</b> <ul style="list-style-type: none"> <li>El programa se ejecuta de forma continuada hasta finalizar.</li> <li>En el modo de test se debe mantener pulsada la tecla de arranque.</li> </ul>
	<b>Movimiento</b> <ul style="list-style-type: none"> <li>En el modo de ejecución Motion Step, cada instrucción de movimiento se ejecuta individualmente.</li> <li>Al finalizar un movimiento se debe pulsar otra vez "Inicio".</li> </ul>
	<b>Paso a paso</b>   Disponible únicamente en el grupo de usuario "Experto" <ul style="list-style-type: none"> <li>En el modo Incremental Step la ejecución se realiza línea a línea (independientemente del contenido de la línea).</li> <li>Después de cada línea se debe volver a accionar la tecla de arranque.</li> </ul>

¿Qué aspecto tiene un programa de robot?

```

1 DEF kuka_rocks( )
2INI
3 PTP HOME  Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[1] Base[0]
5 PTP P2 Vel=100 % PDAT2 Tool[1] Base[0]
6 PTP P3 Vel=100 % PDAT3 Tool[1] Base[0]
7 OUT 1'' State=TRUE CONT
8 LIN P4 Vel=2 m/s CPDAT1 Tool[1] Base[0]
9 PTP HOME  Vel= 100 % DEFAULT
10END

```

Fig. 4-5: Estructura de un programa de robot

	<b>Visible únicamente para el grupo de usuario "Experto":</b>
1	<ul style="list-style-type: none"> <li>■ "DEF nombre del programa ()" aparece siempre al comenzar un programa</li> <li>■ "END" describe el final de un programa</li> </ul>
2	<ul style="list-style-type: none"> <li>■ La línea "INI" contiene la activación de los parámetros estándar necesarios para la correcta ejecución del programa.</li> <li>■ LA línea "INI" se debe ejecutar siempre en primer lugar.</li> </ul>
3	<ul style="list-style-type: none"> <li>■ Texto propio del programa con comandos de movimiento, instrucciones de espera/lógicas, etc.</li> <li>■ El comando de desplazamiento "PTP Home" se utiliza frecuentemente al iniciar y al finalizar un programa, ya que es una posición clara y conocida.</li> </ul>

Estado del programa

Estado del programa	Símbolo	Color	Descripción
		gris	No se encuentra seleccionado ningún programa.
		amarillo	El puntero de paso está sobre la primera línea del programa seleccionado.
		verde	El programa ha sido seleccionado y se encuentra en ejecución.

Símbolo	Color	Descripción
	rojo	El programa seleccionado y arrancado ha sido detenido.
	negro	El puntero de paso está en el final del programa seleccionado.

## Iniciar el programa

### Procedimiento para iniciar programas de robots:

1. Seleccionar un programa

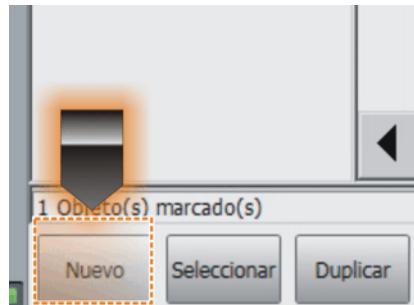


Fig. 4-6: Selección de programa

2. Ajustar la velocidad del programa (override del programa, POV)



Fig. 4-7: Ajuste POV

3. Accionar el pulsador de validación

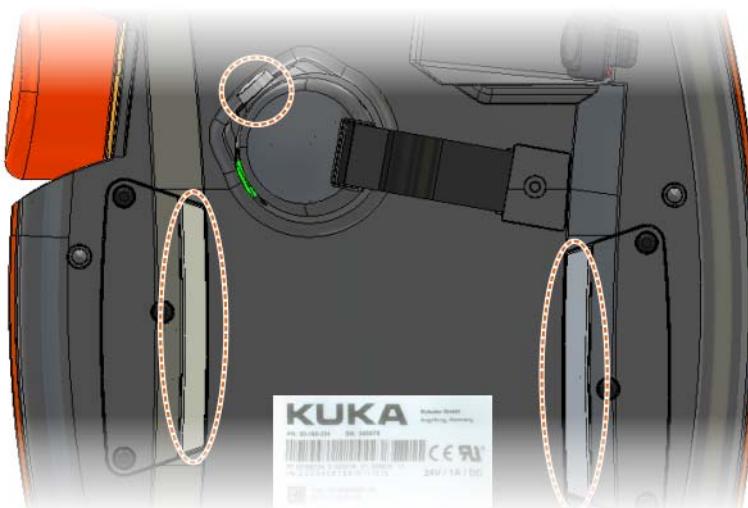


Fig. 4-8: Interruptor de confirmación

4. Mantener pulsada la tecla de inicio (+):
  - Se procesa la línea "INI".

- El robot ejecuta el desplazamiento COI.



**Fig. 4-9: Direcciones de ejecución de programa: Adelante/atrás**

Pos.	Descripción
1	Teclas de arranque <b>Avance</b>
2	Tecla de arranque <b>Retroceso</b>

**ADVERTENCIA**

Cuando el paso de movimiento seleccionado contiene el comando de desplazamiento PTP, se produce un desplazamiento COI a modo de movimiento PTP desde la posición real hasta la posición de destino. Si el paso de movimiento seleccionado contiene LIN o CIRC, el desplazamiento COI se ejecuta como movimiento LIN. Es necesario observar el movimiento para evitar colisiones. En el desplazamiento COI la velocidad se reduce automáticamente.

5. Al alcanzar la posición de destino se detiene el movimiento.

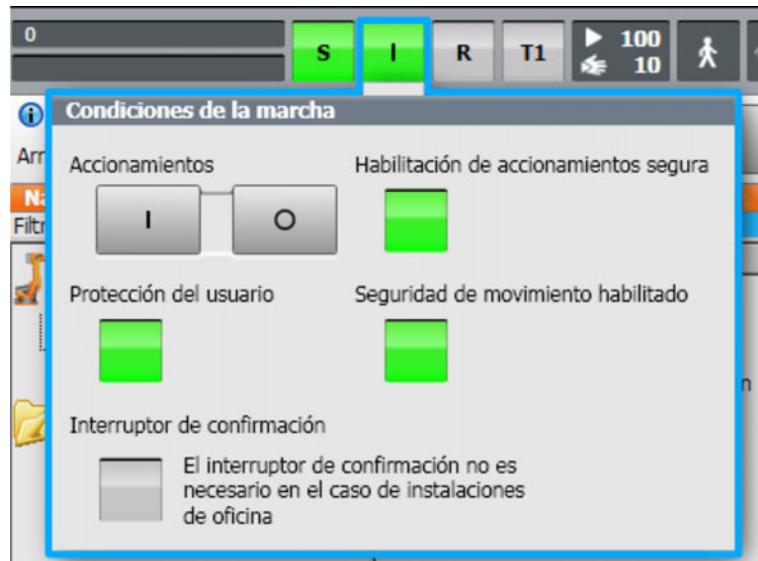


to.

Aparece el mensaje de observación "COI alcanzada".

6. Otras ejecuciones (en función del modo de servicio ajustado):
  - **T1 y T2:** Reanudar el programa pulsando la tecla de inicio.

- **AUT:** Activar accionamientos.



A continuación, iniciar el programa con impulso en *Start*.

- En el programa Cell, cambiar al modo de servicio **EXT** y transferir la instrucción de desplazamiento del PLC.

#### 4.4 Ejercicio: Ejecutar programas del robot

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"><li>■ Seleccionar y deseleccionar programas</li><li>■ Ejecutar, detener y resetear programas en los modos de servicio requeridos (comprobar el flujo de programa)</li><li>■ Efectuar y entender la selección de paso</li><li>■ Efectuar el desplazamiento de coincidencia (COI)</li></ul>
<b>Requisitos</b>	Los siguientes requisitos son necesarios para completar este ejercicio correctamente: <ul style="list-style-type: none"><li>■ Conocimientos teóricos del manejo del navegador</li><li>■ Conocimientos sobre la selección y desección de programas</li></ul>
<b>Formulación de tarea</b>	<ol style="list-style-type: none"><li>1. Seleccionar el módulo Air</li><li>2. Comprobar de la siguiente manera el programa en los diferentes modos de servicio:<ul style="list-style-type: none"><li>■ T1 con 100%</li><li>■ T2 con 10%, 30%, 50%, 75%, 100%</li><li>■ Automático con 100%</li></ul></li><li>3. Comprobar el programa con los tipos ejecución del programa Go y Movimiento.</li></ol>

**¡Peligro!**

Deben respetarse obligatoriamente las prescripciones de seguridad de la instrucción.



## 5 Uso de archivos de programas

### 5.1 Vista general

Esta unidad incluye los siguientes contenidos:

- Crear y editar módulos de programa
- Archivar y restaurar programas de robot
- Manejo del listado LOG

### 5.2 Crear módulos de programa

#### Módulos de programa en el navegador

Los módulos de programa deberían colocarse siempre en la carpeta "Programas". También existe la posibilidad de crear nuevas carpetas en las que guardar los módulos de programa. Los módulos se identifican con el símbolo "M". Se puede incluir un comentario los módulos. Este tipo de comentario puede incluir, p. ej., una breve descripción del funcionamiento del programa.

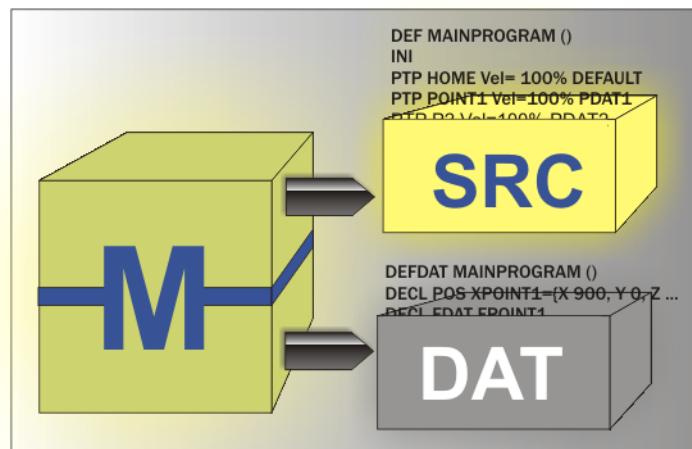


Fig. 5-1: Módulos en el navegador

- 1 Carpeta principal para programas: "Programa"
- 2 Subcarpeta para otros programas
- 3 Módulo de programa/módulo
- 4 Comentario de un módulo de programa

#### Propiedades de los módulos de programación

Un módulo consta siempre de dos partes:



**Fig. 5-2: Estructura del módulo**

- **Código fuente:** El archivo SRC contiene el código del programa.

```

DEF MAINPROGRAM()
INI
PTP HOME Vel= 100% DEFAULT
PTP P1 Vel=100% PDAT1 TOOL[1] BASE[2]
PTP P2 Vel=100% PDAT2 TOOL[1] BASE[2]
...
END
  
```

- **Lista de datos:** El archivo DAT contiene datos y coordenadas de puntos permanentes.

```

DEFDAT MAINPROGRAM()
DECL E6POS XP1={X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27, E1 0,
E2 0, E3 0, E4 0, E5 0, E6 0}
DECL FDAT FPOINT1 ...
...
ENDDAT
  
```

**Procedimiento para crear módulos de programa propios**

1. En la estructura de directorios, marcar la carpeta en que debe crearse el programa, p. ej. la carpeta **Programa**.
2. Pulsar la tecla de función programable **Nuevo**.
3. Introducir un nombre para programa y un comentario en caso necesario y confirmar con **OK**.

### 5.3 Editar módulos de programa

**Opciones de edición**

Como en los sistemas de archivos convencionales, en el navegador del KUKA smartPAD también se pueden editar módulos de programa.

Editar incluye:

- Duplicar/copiar
- Borrar
- Renombrar

**Procedimiento para duplicar un programa**

1. En la estructura de directorios marcar la carpeta en la que se encuentra el archivo.
2. Marcar el archivo en la lista de archivos.
3. Pulsar la tecla de función **Duplicar**.
4. Asignar un nombre de archivo nuevo al nuevo módulo y confirmar pulsando **OK**.



En el grupo de usuario "Experto" y el ajuste de filtro "Detalle" figuran dos archivos por módulo en el navegador (archivo SRC y DAT). Si éste es el caso, se deben duplicar ambos archivos.

#### Procedimiento para borrar un programa

1. En la estructura de directorios marcar la carpeta en la que se encuentra el archivo.
2. Marcar el archivo en la lista de archivos.
3. Pulsar la tecla de función **Borrar >**.
4. Responder **Sí** a la pregunta de seguridad. Se borra el módulo.



En el grupo de usuario "Experto" y el ajuste de filtro "Detalle" figuran dos archivos por módulo en el navegador (archivo SRC y DAT). Si éste es el caso, se deben borrar ambos archivos. Los ficheros borrados no se pueden restaurar.

#### Procedimiento para renombrar un programa

1. En la estructura de directorios marcar la carpeta en la que se encuentra el archivo.
2. Marcar el archivo en la lista de archivos.
3. Seleccionar la tecla de función **Editar > Renombrar**.
4. Sobrescribir el nombre del archivo con el nombre nuevo y confirmar con **OK**.



En el grupo de usuario "Experto" y el ajuste de filtro "Detalle" figuran dos archivos por módulo en el navegador (archivo SRC y DAT). ¡Si éste es el caso, se deben renombrar ambos archivos!

### 5.4 Archivar y restaurar programas de robot

#### Opciones de archivo

Cada proceso de archivo genera un archivo ZIP en el medio de destino correspondiente con el nombre del robot. En el apartado **Datos del robot** se puede cambiar el nombre del archivo.

**Lugar de almacenamiento:** Se dispone de tres lugares de almacenamiento distintos:

- **USB (KCP)** | Memoria USB en el KCP (smartPAD)
- **USB (armario)** | Lápiz USB en el armario de control del robot
- **Red** | Los datos se guardan en una ruta de red  
La ruta de red deseada se debe configurar en **PickControl**.



Cada vez que se archiva, además del archivo ZIP que se genera en el soporte de almacenamiento, paralelamente se crea en la unidad D:\ otro archivo (INTERN.ZIP).

**Datos:** Para el proceso de almacenamiento se pueden seleccionar los datos siguientes:

- **Todo:**  
Se archivan los datos necesarios para restituir un sistema existente.
- **Aplicaciones:**  
Se archivan todos los módulos KRL (programas) definidos por el usuario y los archivos correspondientes del sistema.
- **Datos de sistema:**  
Se archivan los datos de la máquina.
- **Datos del log:**  
Se archivan los ficheros Log.

**■ KrcDiag:**

Se archivan los datos para entregarlos a KUKA Roboter GmbH para que los analicen en busca de errores. Se crea una carpeta (nombre **KRCDiag**) en la que se pueden registrar hasta diez archivos ZIP. Además, se guardará en la unidad de control en C:\KUKA\KRCDiag.

**Restaurar datos**

**ADVERTENCIA** Por regla general sólo se pueden cargar archivos con la versión de software compatible. Si se cargan otros archivos, pueden producirse las situaciones siguientes:

- Mensajes de error
- La unidad de control del robot no puede operar.
- Lesiones y daños materiales

Al restaurar pueden seleccionarse entre las siguientes opciones de menú:

- **Todo**
- **Aplicaciones**
- **Datos de sistema**



En los siguientes casos el sistema emite un mensaje de error:

- Si los datos archivados están guardados en otra versión diferente a la que se encuentra en el sistema.
- Si la versión de los paquetes de tecnología no coincide con la versión instalada.

**Procedimiento para archivar**

**AVISO** Sólo debe utilizarse la memoria KUKA.USBDATA. Si se utiliza otra clave USB, pueden perderse o cambiarse los datos.

1. Seleccionar la secuencia **Archivo > Archivar > USB (KCP) o USB (armario)** y la opción secundaria deseada.
2. Responder a la pregunta de seguridad con **Sí**.  
La ventana de mensajes indica cuando se ha finalizado con el archivado.
3. Se puede extraer la memoria USB cuando su LED se apague.

**Restablecer procedimiento**

1. Seleccionar la secuencia de menús **Archivo > Restaurar >** y luego las opciones secundarias.
2. Responder **Sí** a la pregunta de seguridad. Los ficheros archivados se restauran en la unidad de control del robot. Se indica con un mensaje que la restauración ha finalizado.
3. Si se restauró desde una memoria USB: retirar el dispositivo USB.



**AVISO** Al restaurar desde el dispositivo USB: en cuanto se apague el LED del dispositivo USB, este podrá retirarse. De lo contrario, el dispositivo puede sufrir daños.

4. Reiniciar la unidad de control del robot. Para ello es necesario un arranque en frío.
  - Perfil de usuario experto  
**Ruta:** botón KUKA > Configuración > Grupo de usuarios > Experto
  - Iniciar el arranque en frío con:  
**Ruta:** botón KUKA > Apagar > Arranque en frío

## 5.5 Registrar los cambios de programa y de estado con el listado LOG

### Opciones de protocolización

Las operaciones que efectúa el usuario en el smartPAD se guardan automáticamente en un protocolo. La instrucción **Protocolo** muestra el protocolo.

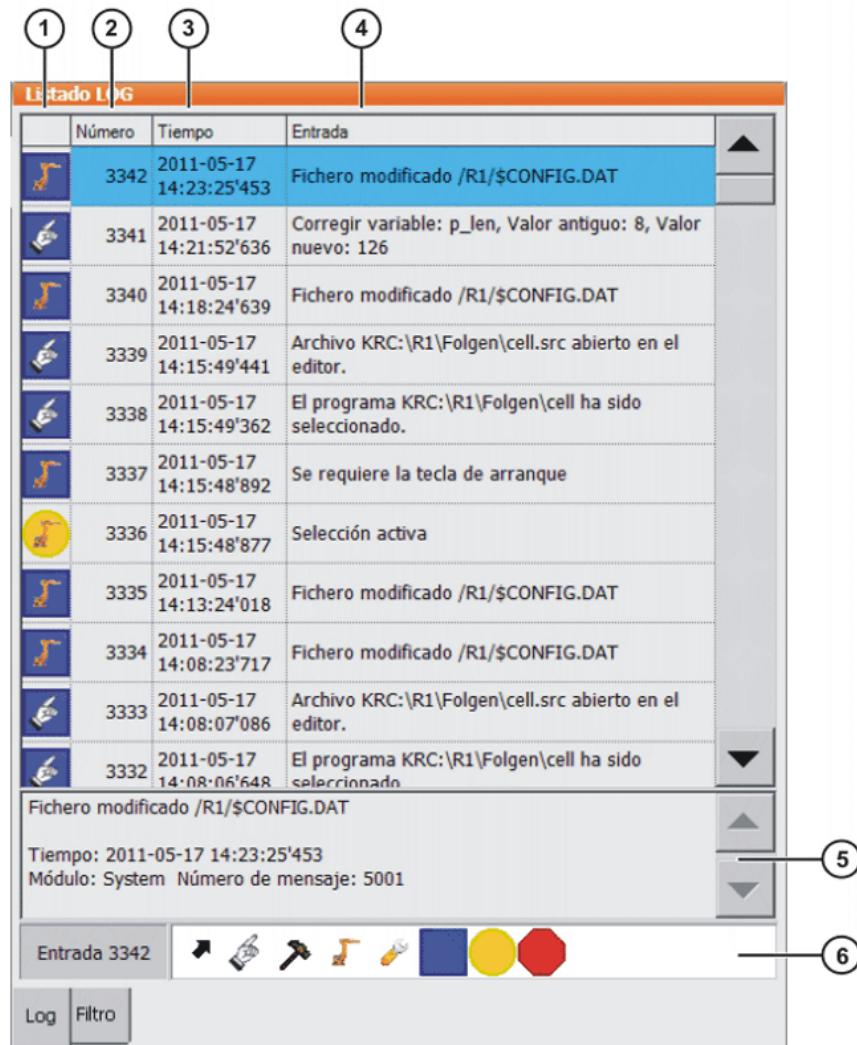


Fig. 5-3: Protocolo, tarjeta de registro Log

Pos.	Descripción
1	Tipo de incidencia Log Los distintos tipos de filtro están en un listado de la tarjeta de registro <b>Filtro</b> .
2	Número de incidencia Log
3	Fecha y hora del incidencia Log
4	Breve descripción del incidencia Log
5	Descripción detallada del incidencia Log marcada
6	Visualización del filtro activo

## Filtrado de incidencias Log

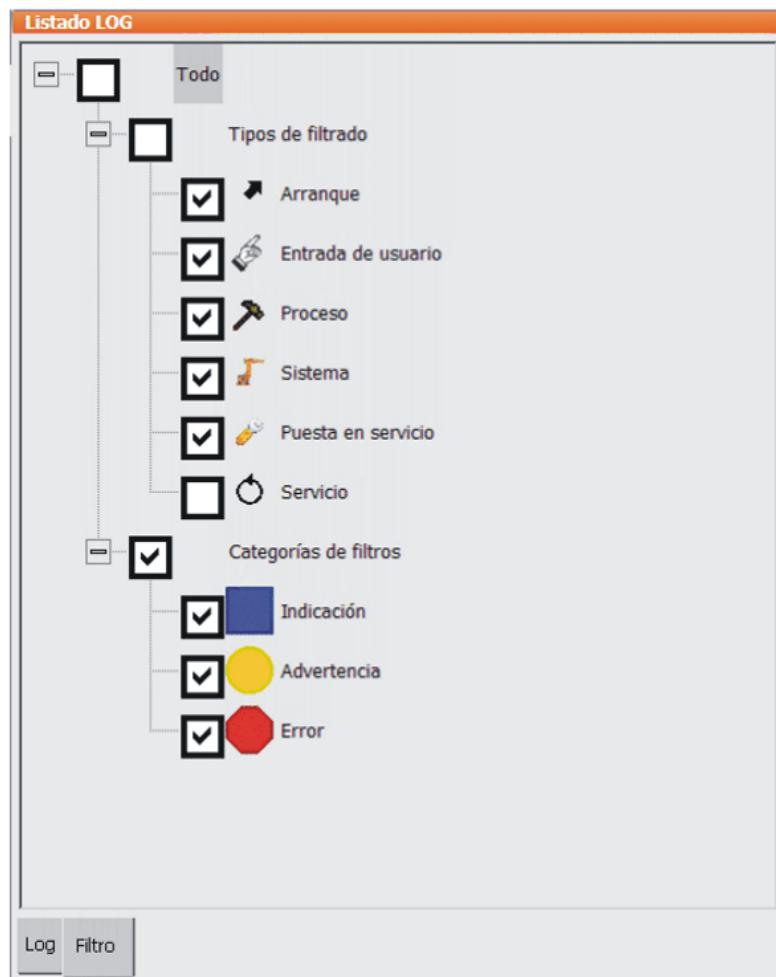


Fig. 5-4: Protocolo, tarjeta de registro Filtro

### Uso de la función listado LOG

Todos los grupos de usuarios pueden ver y configurar esta función.

#### Mostrar listado LOG:

- Seleccionar en el menú principal la secuencia **Diagnóstico > Listado LOG > Visualizar**.

#### Configurar listado LOG:

1. Seleccionar en el menú principal **Diagnóstico > Listado LOG > Configuración**.
2. Pulsar **OK** para guardar la configuración y cerrar la ventana.

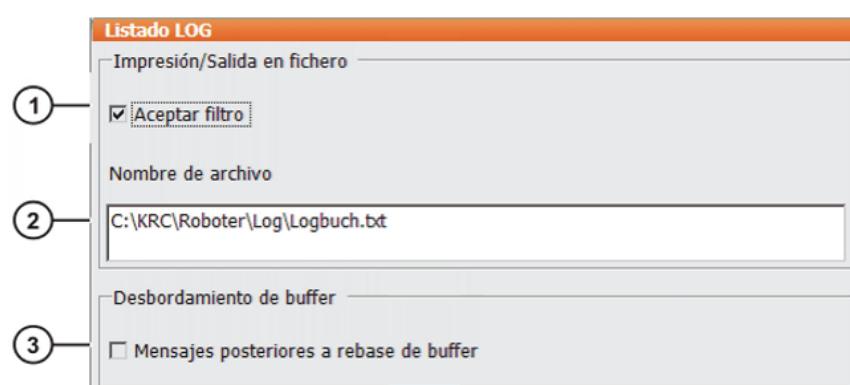


Fig. 5-5: Ventana Configuración del listado LOG

1	Realizar los ajuste del filtro para la edición. Si no hay ninguna marca de verificación, la edición no está filtrada.
2	Ruta del archivo de texto.
3	Los datos del log que se han borrado por el desbordamiento de la memoria intermedia aparecen en gris en el archivo de texto.



## 6 Crear y modificar movimientos programados

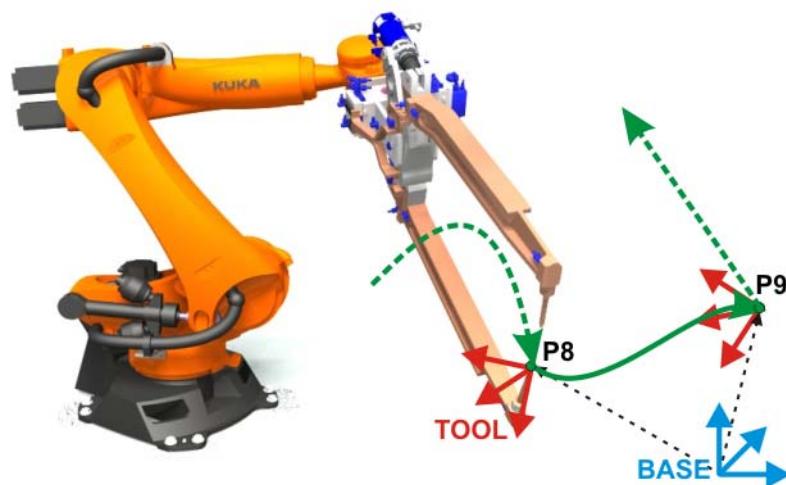
### 6.1 Resumen

**Esta unidad incluye los siguientes contenidos:**

- Creación de nuevas instrucciones de movimiento
- Crear movimientos con optimización del tiempo de ciclo
- Crear movimientos de trayectoria
- Modificar instrucciones de movimiento
- Programar la detección de colisión

### 6.2 Creación de nuevas instrucciones de movimiento

**Programar  
movimientos de  
robot**



**Fig. 6-1: Movimiento de robot**

Cuando se deban programar movimientos del robot, se plantean una serie de preguntas:

Pregunta	Solución	Palabra clave
¿Cómo detecta el robot sus posiciones?	La posición de la herramienta en el espacio se guarda (posición del robot según el sistema Tool y Base ajustados).	POS E6POS
¿Cómo sabe el robot que se debe mover?	Por la indicación del modo de movimiento: punto a punto, lineal o circular.	PTP LIN CIRC
¿Con qué rapidez se mueve el robot?	La velocidad entre dos puntos y la aceleración se indican en la programación.	Vel. Acc.
¿El robot se debe parar en cada punto?	Para poder acortar la duración del ciclo también se pueden aproximar puntos. En este caso, no tiene lugar una parada exacta.	CONT

Pregunta	Solución	Palabra clave
¿Qué orientación adopta la herramienta cuando se alcanza un punto?	Para cada movimiento se puede ajustar individualmente el control de la orientación. Este ajuste solo es válido para movimientos de trayectoria. ( <a href="#">&gt;&gt;&gt; "Tipos de movimiento"</a> Página 118)	ORI_TYPE
¿El robot detecta un obstáculo?	No, el robot sigue la trayectoria programada sin desviarse. El programador será el responsable de garantizar que no se producen colisiones.  No obstante, existe la posibilidad de implementar un "control contra colisiones" para proteger la máquina.	Control contra colisiones

Al programar movimientos de robot en el proceso de aprendizaje se debe transmitir esta información. Para ello se usan formularios inline en los que se puede introducir cómodamente esta información.

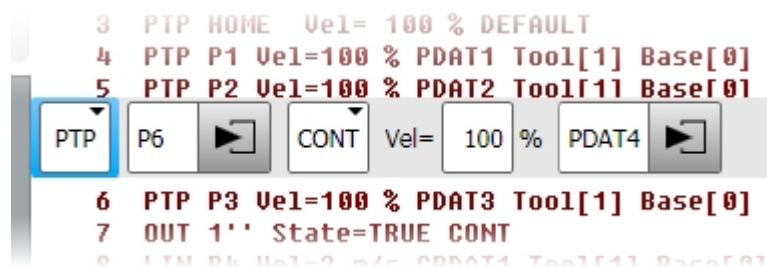


Fig. 6-2: Formulario online para programar movimientos

### Tipos de movimiento

Para programas las instrucciones de movimiento se dispone de varios tipos de movimiento. Los movimientos se pueden programar en función de los requisitos del proceso de trabajo del robot.

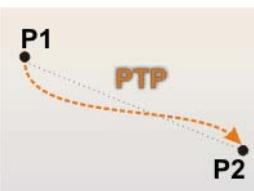
- Movimiento específico del eje (PTP: Point to Point)
- Movimientos de trayectoria: LIN (lineal) y CIRC (circular)
- SPLINE: SPLINE es un tipo de movimiento especialmente apropiado para trayectorias curvas complejas. En principio, este tipo de trayectorias también se pueden crear con movimientos LIN y CIRC, pero el SPLINE presenta una serie de ventajas.



Los movimientos SPLINE no se incluyen en esta documentación para la formación. Encontrará información más detallada al respecto en las *Manual de servicio y programación KUKA System Software 8.x*.

## 6.3 Crear movimientos con optimización del tiempo de ciclo (movimiento del eje)

### Tipo de movimiento PTP

Tipo de movimiento	Significado	Ejemplo de aplicación
	<p><i>Point to Point:</i> punto a punto</p> <ul style="list-style-type: none"> <li>■ Movimiento específico del eje: El robot desplaza el TCP al punto de destino con el movimiento <b>más rápido</b>. La trayectoria más rápida no es, por lo general, la trayectoria más corta y, por tanto, <b>no</b> es una recta. Dado que los ejes del robot se mueven de forma rotacional, trayectorias curvas pueden ser ejecutadas de forma más rápida que las rectas.</li> <li>■ No puede predecirse la trayectoria exacta.</li> <li>■ El eje directriz es el eje que más tarda en alcanzar el punto de destino.</li> <li>■ SYNCHRO PTP: todos los ejes arrancan al mismo tiempo y se detienen sincronizados.</li> <li>■ El primer movimiento en el programa debe ser un movimiento PTP ya que, en este caso, únicamente se van a evaluar Status y Turn.</li> </ul>	<p>Aplicaciones de trayectoria, p. ej.:</p> <ul style="list-style-type: none"> <li>■ Soldadura por puntos</li> <li>■ Transportar</li> <li>■ Medir, controlar</li> </ul> <p>Posiciones auxiliares:</p> <ul style="list-style-type: none"> <li>■ puntos intermedios</li> <li>■ puntos libres en el espacio</li> </ul>

### Synchro-PTP

El eje directriz es el eje que más tarda en alcanzar el punto de destino. Se tiene en cuenta la indicación de la velocidad en el formulario inline.

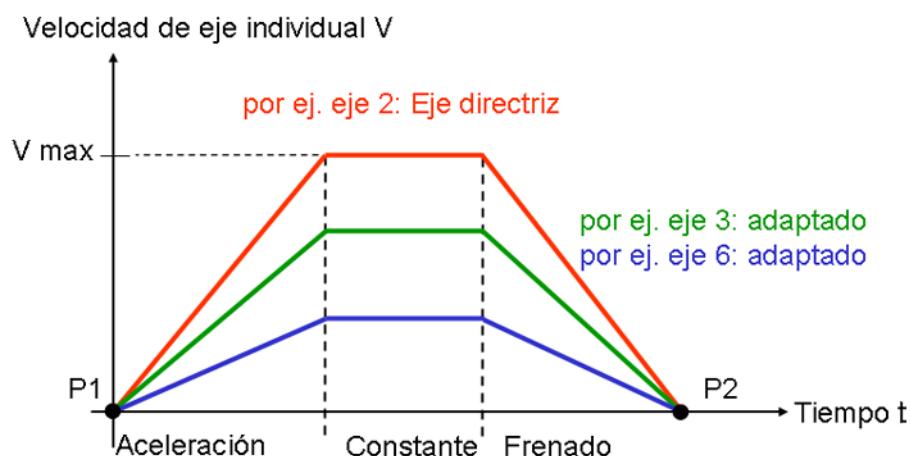
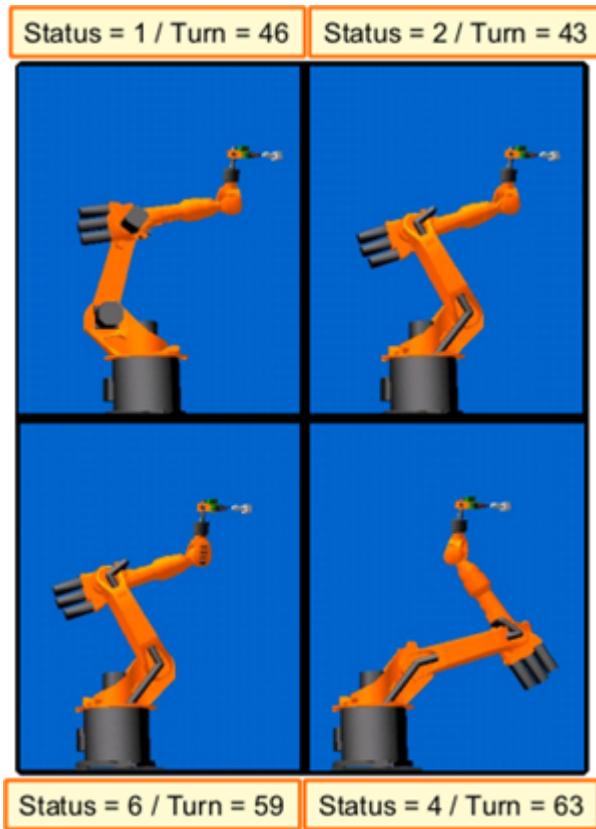


Fig. 6-3: Synchro-PTP

### Status y Turn

Status y Turn sirven para determinar una posición de eje única a partir de varias posiciones de eje posibles para una misma posición del TCP.

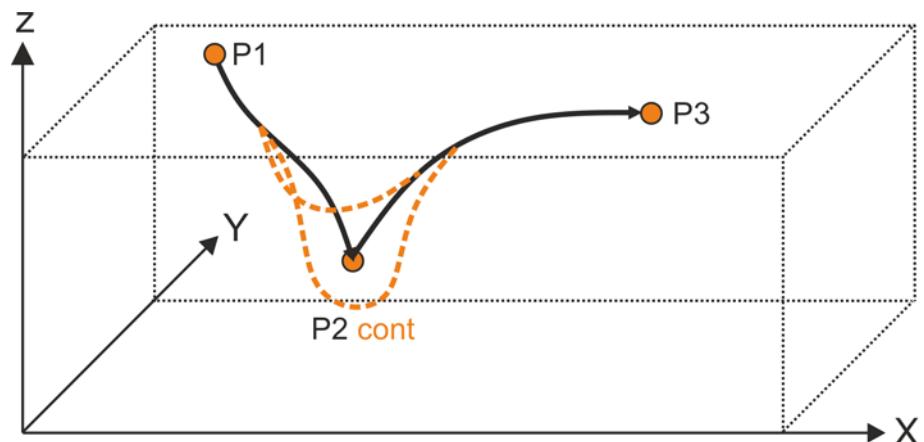


**Fig. 6-4: Diferentes posiciones de eje debidos a diferentes valores de Status y Turn**

La unidad de control del robot considera los valores de Status y Turn programados sólo en caso de movimientos PTP. En movimientos de trayectoria (CP) se ignoran. Por ello, la primera instrucción de movimiento en un programa KRL debe ser una instrucción PTP completa del tipo POS o E6POS para definir una posición de salida única. (o bien una instrucción PTP completa del tipo AXIS o E6AXIS.)

```
DEFDAT MAINPROGRAM ()
DECL POS XPOINT1={X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27}
DECL FDAT FPOINT1 ...
...
ENDDDAT
```

### Aproximación

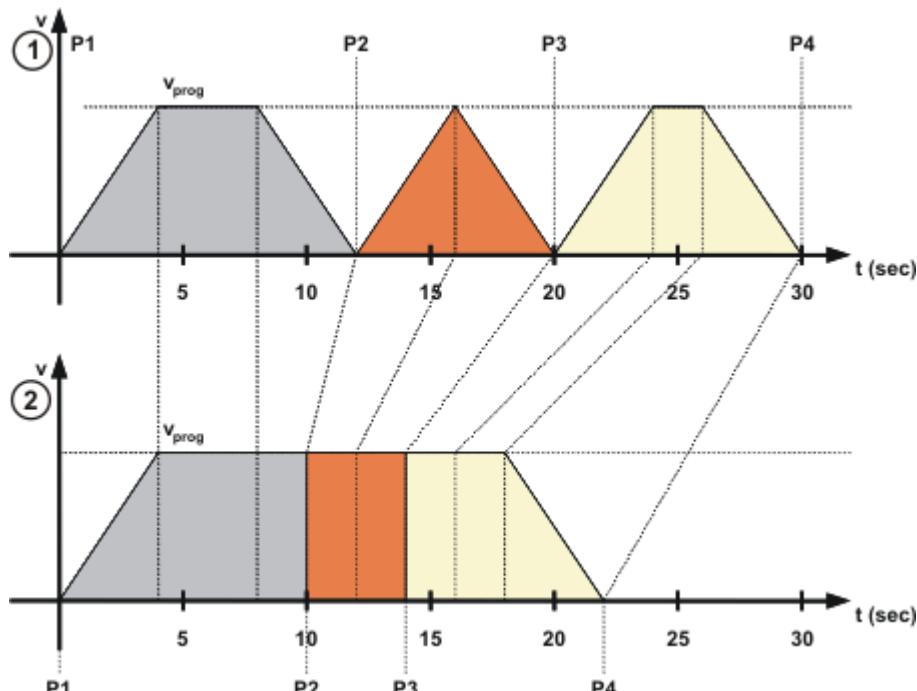


**Fig. 6-5: Aproximación a un punto PTP**

La unidad de control es capaz de aproximar las instrucciones de movimiento marcadas con CONT para acelerar el movimiento. Aproximar significa que no se desplaza exactamente a las coordenadas de punto. Previamente se aban-

dona la trayectoria del contorno de paradas exactas. El TCP se conduce a lo largo de los contornos de aproximación que desemboca en el contorno de paradas exactas de la siguiente instrucción de movimiento.

### Ventajas de la aproximación

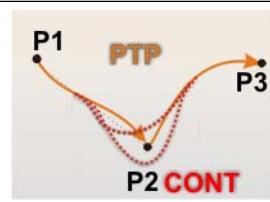


**Fig. 6-6: Parada exacta - Posicionamiento aproximado en comparación**

- La cinemática se desgasta menos, puesto que ya no es necesario frenar y acelerar entre los puntos (ver Punto 1).
- De esta forma se optimiza el tiempo de ciclo y el programa se ejecuta más rápido (ver Punto 2).

Para poder ejecutar el movimiento de aproximación la unidad de control debe poder leer los siguientes pasos de movimiento. Ello se realiza con procesamiento en avance.

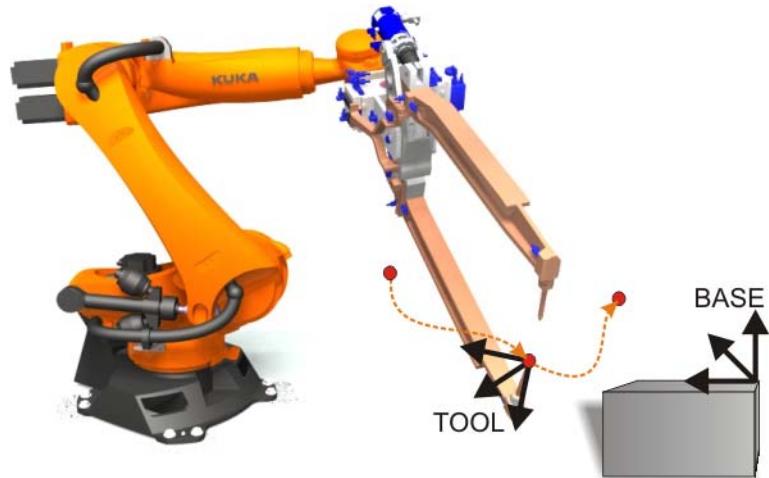
#### Aproximación en el tipo de desplazamiento PTP

Tipo de movimiento	Característica	Distancia de aproximación
	<ul style="list-style-type: none"> <li>■ El contorno de aproximación no es predecible.</li> </ul>	% o indicación en mm

#### Procedimiento para crear movimientos PTP

#### Requisitos

- Se ha activado el modo de servicio T1.
  - Está seleccionado un programa de robot.
1. Mover el TCP a la posición que se programará por aprendizaje como punto de destino.

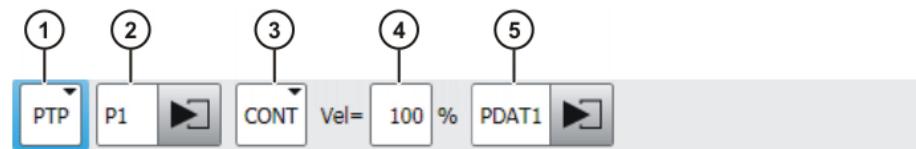


**Fig. 6-7: Instrucción de movimiento**

2. Colocar el cursor en la línea detrás de la cual se insertará la instrucción de movimiento.
3. Seleccionar la secuencia de menú **Instrucciones > Movimiento > PTP**. Como alternativa también se puede accionar la tecla de función **Movimiento** localizada en la línea pertinente.

Aparece un formulario inline:

- **Formulario online PTP**



**Fig. 6-8: Formulario inline Movimiento PTP**

4. Introducir parámetros en el formulario inline.

Pos.	Descripción
1	Tipo de movimiento <b>PTP</b>
2	El nombre del punto de destino se determina automáticamente, no obstante, se puede sobrescribir individualmente. Para editar los puntos de datos, debe tocarse la flecha. Se abre la ventana de opciones <b>Vectores</b> .
3	<ul style="list-style-type: none"> <li>■ <b>CONT</b>: El punto de destino es de posicionamiento aproximado.</li> <li>■ <b>[vacío]</b>: El punto de destino se alcanza con exactitud.</li> </ul>
4	Velocidad <ul style="list-style-type: none"> <li>■ <b>Movimientos PTP: 1 ... 100 %</b></li> </ul>
5	Juego de datos de movimiento: <ul style="list-style-type: none"> <li>■ <b>Aceleración</b></li> <li>■ <b>Distancia de aproximación</b> (cuando en el campo [3] se haya introducido <b>CONT</b>).</li> </ul>

5. En la ventana de opciones Vectores, introducir los datos correctos para el sistema de coordenadas de base y herramienta, además de las indicaciones sobre el modo de interpolación (TCP externo: CON./DSECON.) y el control contra colisiones.

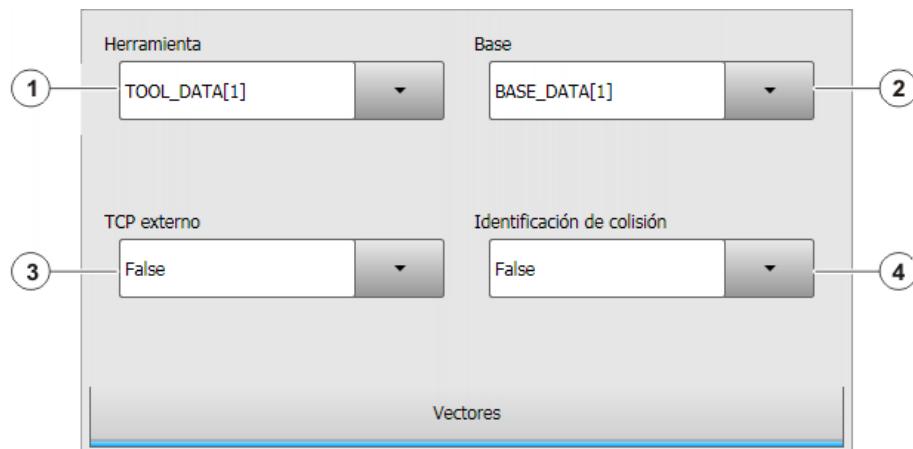


Fig. 6-9: Ventana de opciones Frames

Pos.	Descripción
1	Seleccionar herramienta. Si se tiene <b>True</b> en el campo <b>TCP externo</b> : Seleccionar la pieza. Rango de valores: [1] ... [16]
2	Seleccionar base. Si se tiene <b>True</b> en el campo <b>TCP externo</b> : Seleccionar herramienta fija. Rango de valores: [1] ... [32]
3	Modo de interpolación TCP externo: ■ <b>FALSE</b> : la herramienta está montada en la brida de acople. ■ <b>TRUE</b> : la herramienta es una herramienta fija.
4	Detección de colisión ■ <b>TRUE</b> : para este movimiento, la unidad de control del robot determina los momentos axiales. Éstos son necesarios para la detección de colisión. ■ <b>FALSE</b> : para este movimiento, la unidad de control del robot no determina momentos axiales. Por lo tanto, no es posible una detección de colisión para este movimiento.

6. En la ventana de opciones Parámetros de movimiento, puede reducirse la aceleración del valor máximo. En caso de que se active la aproximación, también se puede modificar la distancia de aproximación. Dependiendo de la configuración, la distancia se ajusta en **mm** o **%**.

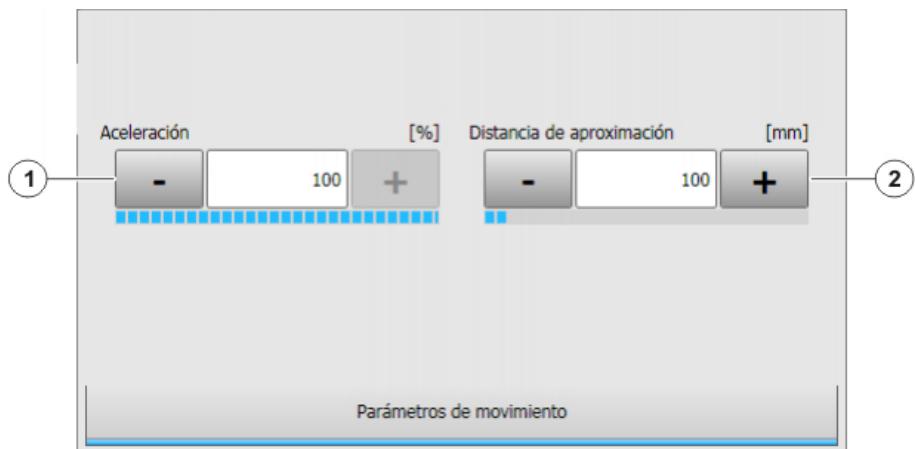


Fig. 6-10: Ventana de opciones Parámetros de movimiento (PTP)

Pos.	Descripción
1	<b>Aceleración</b> Se refiere al valor máximo indicado en los datos de la máquina. El valor máximo depende del tipo de robot y del modo de servicio ajustado. La aceleración es válida para los ejes directrices de este conjunto de movimientos. <input type="checkbox"/> 1 ... 100 %
2	<b>Distancia de aproximación</b> Este campo solo se visualiza cuando se haya seleccionado <b>CONT</b> en el formulario inline. Distancia antes del punto de destino en donde comienza, como muy pronto, la aproximación. Distancia máxima: la mitad de la distancia entre el punto de inicio y el de destino, referido al contorno del movimiento PTP sin aproximación. <input type="checkbox"/> 1 ... 100 % <input type="checkbox"/> 1 ... 1000 mm

7. Guardar la instrucción, pulsando **Instrucción OK**. La posición opuesta del TCP se programa como punto de destino.

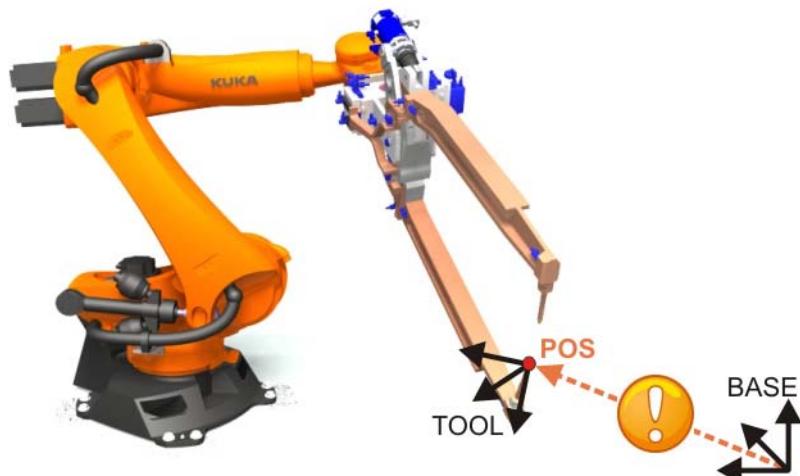


Fig. 6-11: Memorización de las coordenadas de puntos en "Instrucción OK" y "TouchUp".

## 6.4 Ejercicio: Programa en el aire - Tratamiento de programa y movimientos PTP

### Objetivo del ejercicio

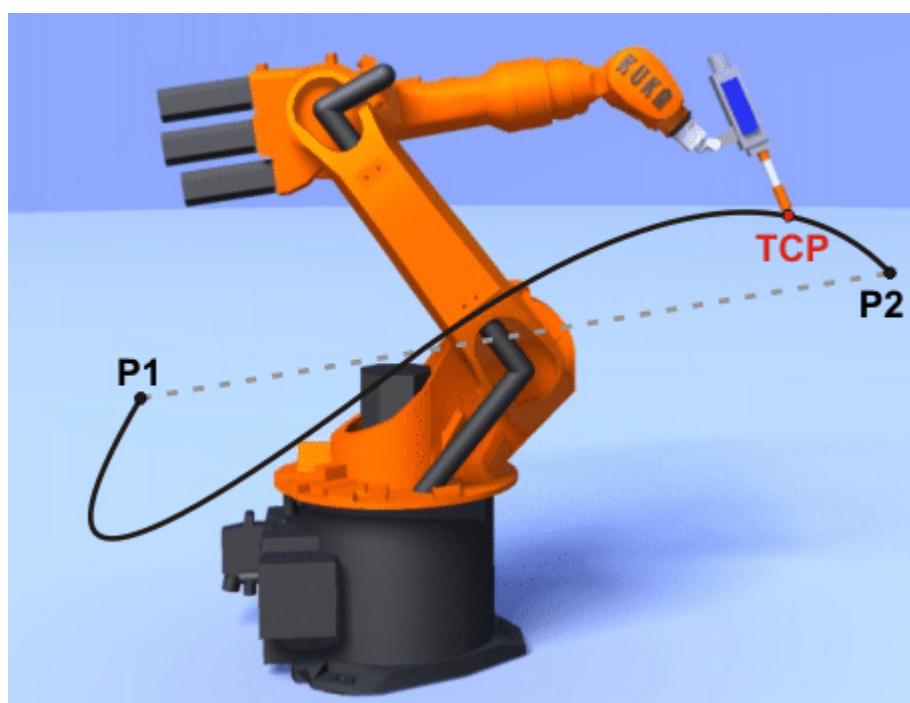
Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Seleccionar y deseleccionar programas
- Ejecutar, detener y resetear programas en los modos de servicio requeridos (comprobar el flujo de programa)
- Corregir puntos del programa existentes
- Borrar los conjuntos de movimientos e insertar nuevos movimientos PTP
- Cambiar modo de desarrollo de programa y desplazar a puntos programados paso a paso
- Efectuar y entender la selección de paso
- Efectuar el desplazamiento de coincidencia (COI)

### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos del manejo del navegador
- Conocimientos teóricos del tipo de movimiento PTP



### Tarea sección A

Ejecutar las siguientes tareas: Crear y someter a prueba programas

1. Crear un nuevo módulo con el nombre **Air\_PROG**



#### Peligro

Deben respetarse obligatoriamente las prescripciones de seguridad de la instrucción.

2. Crear una secuencia de unos cinco conjuntos de movimientos PTP.
3. En caso de que no se disponga de un desplazamiento sin colisiones, deben eliminarse los puntos afectados y establecerse un punto nuevo.
4. Comprobar el programa en el modo de servicio T1, con diferentes velocidades de programa (POV).
5. Comprobar el programa en el modo de servicio T2, con diferentes velocidades de programa (POV).
6. Comprobar el programa en el modo de servicio "Automático".

**Tarea sección B**

Ejecutar las siguientes tareas: Corrección del programa

1. Utilizar distintas velocidades para sus puntos en el espacio.
2. Seleccionar repetidas veces el mismo punto en el programa.
3. Borrar los conjuntos de movimientos e insertar otros nuevos en otro lugar en el programa.
4. Efectuar una selección de paso.
5. Parar el programa en el modo de test y utilizar la función **Arranque del programa hacia atrás**.
6. Comprobar el programa en los modos de servicio T1, T2 y Automático.

**Lo que se debe saber tras el ejercicio:**

1. ¿Cuál es la diferencia entre seleccionar y abrir un programa?

.....  
.....  
.....

2. ¿Qué modos de ejecución del programa existen y por qué son necesarios?

.....  
.....  
.....

3. ¿Qué es un desplazamiento COI?

.....  
.....  
.....

4. ¿Cómo se puede influenciar la velocidad de programa?

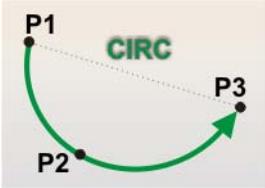
.....  
.....  
.....

5. ¿Qué es lo característico de un movimiento PTP?

.....  
.....  
.....

## 6.5 Crear movimientos de trayectoria

### Tipos de movimiento LIN y CIRC

Tipo de movimiento	Significado	Ejemplo de aplicación
	<p><i>Linear:</i> Lineal</p> <ul style="list-style-type: none"> <li>■ Movimiento de trayectoria rectilíneo.</li> <li>■ El TCP de la herramienta se desplaza desde el inicio hasta el punto de destino con una velocidad constante y una orientación definida.</li> <li>■ La velocidad y la orientación hacen referencia al TCP.</li> </ul>	Aplicaciones de trayectoria, p. ej.: <ul style="list-style-type: none"> <li>■ Soldadura de trayectoria</li> <li>■ Aplicación de pegamento</li> <li>■ Soldadura/corte láser</li> </ul>
	<p><i>Circular:</i> Circular</p> <ul style="list-style-type: none"> <li>■ El movimiento de trayectoria circular se define a través del punto de inicio, el punto auxiliar y el punto de destino.</li> <li>■ El TCP de la herramienta se desplaza desde el inicio hasta el punto de destino con una velocidad constante y una orientación definida.</li> <li>■ La velocidad y la orientación hacen referencia al TCP de la herramienta (sistema de coordenadas de la herramienta).</li> </ul>	Aplicaciones de trayectoria como en el LIN: <ul style="list-style-type: none"> <li>■ Círculos, radios, curvaturas</li> </ul>

### Posiciones singulares

Los robots KUKA con 6 grados de libertad tienen 3 posiciones singulares distintas.

Una posición singular se caracteriza por no permitir una transformación de retroceso (conversión de las coordenadas cartesianas en valores específicos de los ejes) única aunque se hayan preestablecido los datos Status y Turn. En este caso o cuando las más pequeñas modificaciones cartesianas provocan grandes cambios en el ángulo de los ejes, se habla de posiciones singulares. Esta no es una característica mecánica, sino matemática, y por este motivo solo existe en el área de la trayectoria pero no en los movimientos de los ejes.

### Singularidad por encima de la cabeza α1

En la singularidad por encima de la cabeza, el punto de la raíz de la muñeca (= centro del eje A5) se halla vertical al eje A1 del robot.

La posición del eje A1 no se puede establecer únicamente mediante la transformación de retroceso y puede por tanto aceptar cualquier valor.

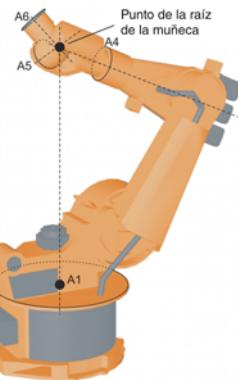


Fig. 6-12: Singularidad por encima de la cabeza (posición  $\alpha 1$ )

### Singularidad de las posiciones extendidas $\alpha 2$

En la singularidad de las posiciones extendidas, el punto de la raíz de la muñeca (= centro del eje A5) se halla en prolongación de los ejes A1 y A3 del robot.

El robot se encuentra en el límite de su área de trabajo.

La transformación de retroceso proporciona un ángulo de eje unívoco, pero las pequeñas velocidades cartesianas dan lugar a grandes velocidades axiales en los ejes A2 y A3.

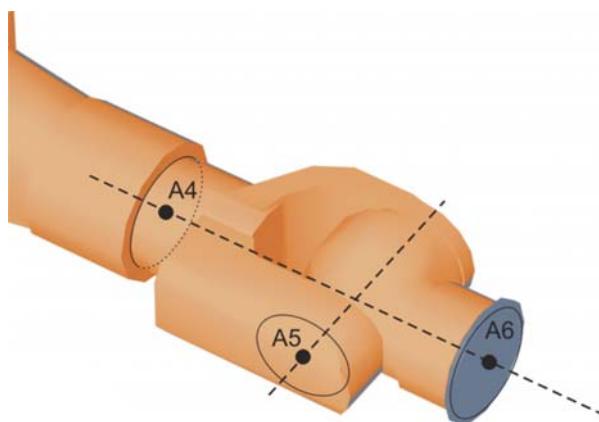


Fig. 6-13: Posición extendida (posición  $\alpha 2$ )

### Singularidades de los ejes de la muñeca $\alpha 5$

En una singularidad de los ejes de la muñeca los ejes A4 y A6 se hallan paralelos uno con el otro y el eje A5 dentro del área de  $\pm 0,01812^\circ$ .

La posición de ambos ejes no se puede determinar inequívocamente por medio de una transformación de retroceso. Pero existen muchas posiciones axiales para los ejes A4 y A6 en las que las sumas de los ángulos de eje son idénticas.



**Fig. 6-14: Singularidad por encima de la cabeza (posición a1)**

## Control de la orientación en movimientos de trayectoria

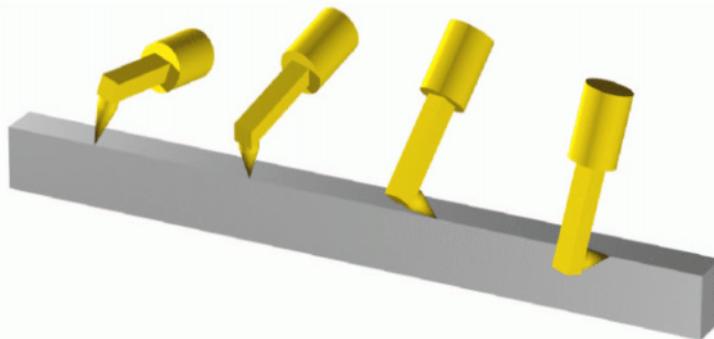
Para los movimientos de trayectoria existe la posibilidad de definir con exactitud el control de la orientación. La herramienta puede tener en los puntos de partida y de destino, un movimiento con distinta orientación.

Controles de la orientación en el tipo de movimiento **LIN**

- **Estándar o PTP manual**

La orientación de la herramienta se modifica de forma continua durante el movimiento.

Utilizar el PTP manual cuando el robot en estándar entra en una singularidad de los ejes de la muñeca, ya que la orientación se logra mediante el desplazamiento lineal (desplazamiento específico del eje) del ángulo de los ejes de la muñeca.



**Fig. 6-15: Estándar**

- **Constante**

La orientación de la herramienta se mantiene de forma constante durante el movimiento, es decir, como cuando se programó en el punto de inicio. La orientación programada en el punto de destino se ignora.

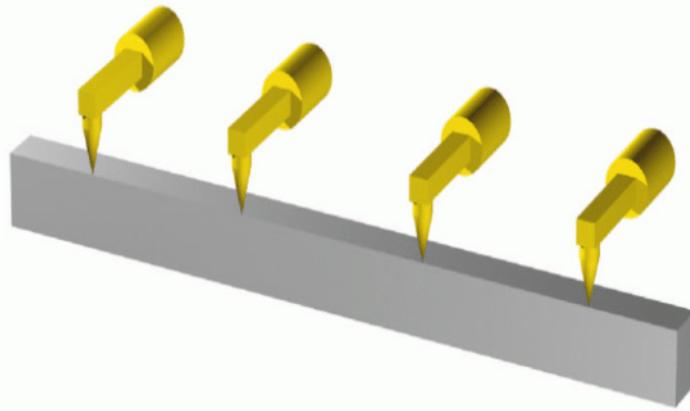


Fig. 6-16: Orientación constante

Controles de la orientación en el tipo de movimiento **CIRC**

■ **Estándar o PTP manual**

La orientación de la herramienta se modifica de forma continua durante el movimiento.

Utilizar el PTP manual cuando el robot en estándar entra en una singularidad de los ejes de la muñeca, ya que la orientación se logra mediante el desplazamiento lineal (desplazamiento específico del eje) del ángulo de los ejes de la muñeca.

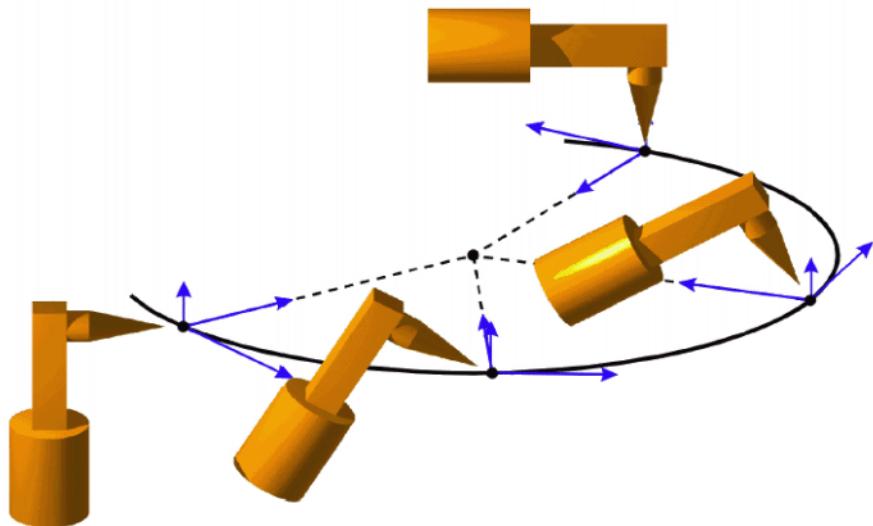
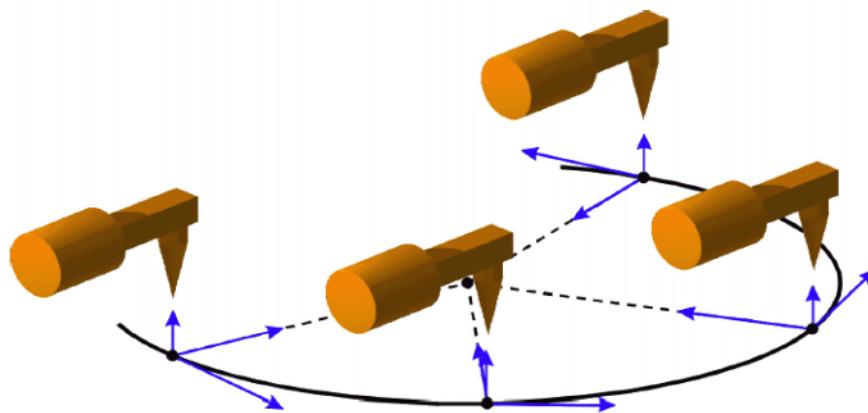


Fig. 6-17: Estándar + referida a la base

■ **Constante**

La orientación de la herramienta se mantiene de forma constante durante el movimiento, es decir, como cuando se programó en el punto de inicio. La orientación programada en el punto de destino se ignora.

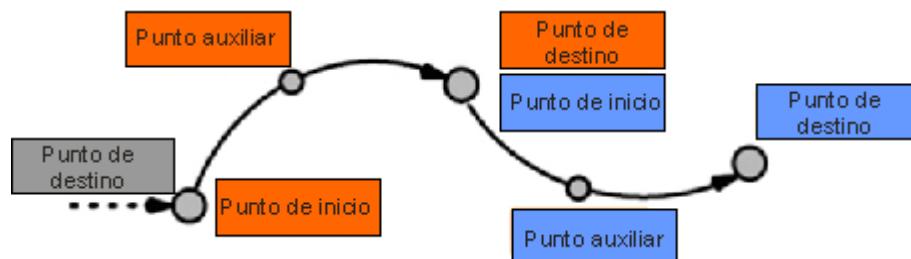


**Fig. 6-18: Control de orientación constante + referida a la base**

#### Ejecución del movimiento con CIRC:

Aquí se mueve el punto de referencia de la herramienta o de la pieza sobre un arco circular al punto de destino. El recorrido se describe mediante puntos de inicio, puntos auxiliares y puntos de destino. Como punto de inicio vale en este caso el punto de destino de la instrucción del movimiento anterior.

La orientación de la herramienta en el punto auxiliar no es relevante.



**Fig. 6-19: Dos segmentos circulares con CIRC**

#### Aproximación de los movimientos de trayectoria



La función de aproximación no es apropiada para crear movimientos circulares. Se trata tan solo de una función destinada a evitar una detención exacta en el punto.

Aproximación a los tipos de movimiento PTP, LIN y CIRC

Tipo de movimiento	Característica	Distancia de aproximación
	<ul style="list-style-type: none"> <li>El curso de la trayectoria corresponde a dos paráolas.</li> </ul>	Indicación en mm
	<ul style="list-style-type: none"> <li>El curso de la trayectoria corresponde a dos paráolas.</li> <li>Se produce una aproximación del punto de destino.</li> </ul>	Indicación en mm

**Procedimiento  
para crear  
movimientos LIN  
y CIRC**

**Requisitos**

- Se ha activado el modo de servicio T1.
  - Está seleccionado un programa de robot.
1. Mover el TCP a la posición que se programará por aprendizaje como punto de destino.

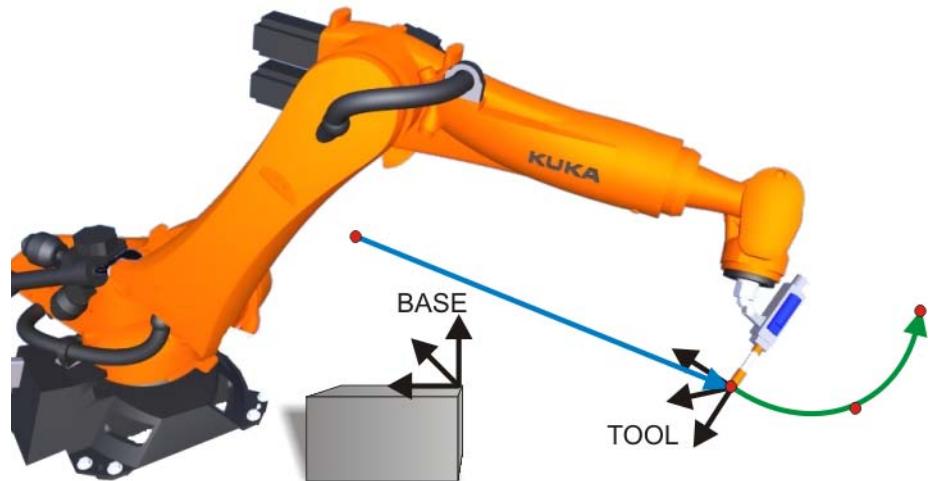


Fig. 6-20: Instrucción de movimiento con LIN y CIR

2. Colocar el cursor en la línea detrás de la cual se insertará la instrucción de movimiento.
3. Seleccionar la secuencia de menú **Instrucciones > Movimiento > LIN o CIRC**.

Como alternativa también se puede accionar la tecla de función **Movimiento** localizada en la línea pertinente.

Aparece un formulario inline:

■ **Formulario inline LIN**

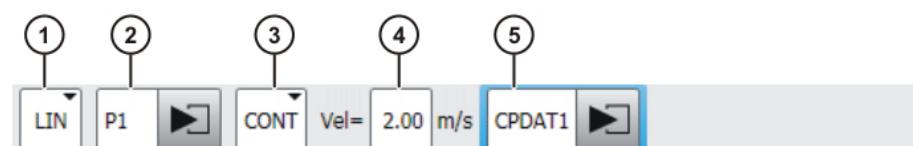


Fig. 6-21: Formulario inline para movimiento LIN

■ **Formulario inline CIRC**

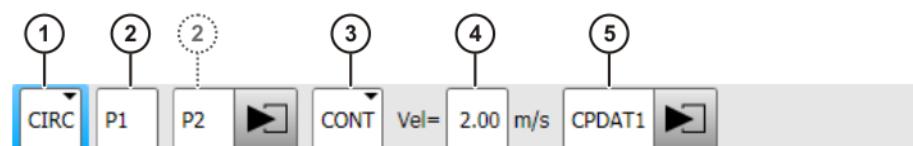
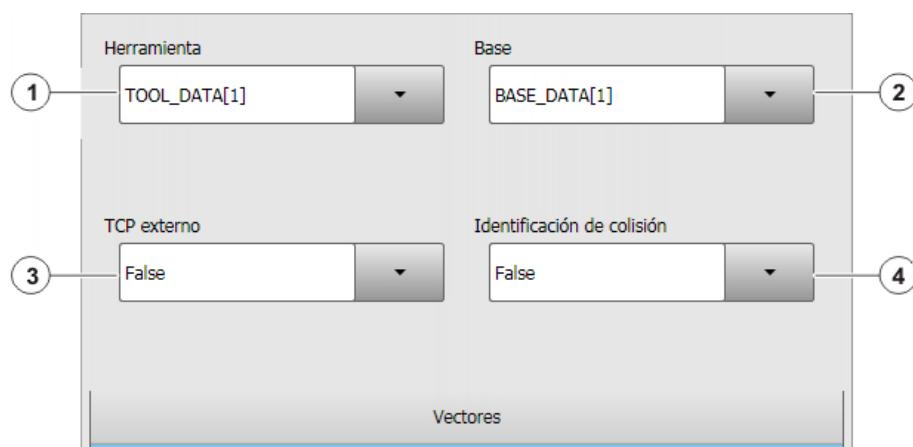


Fig. 6-22: Formulario inline para movimientos CIRC

4. Introducir parámetros en el formulario inline.

Po. s.	Descripción
1	Tipo de movimiento <b>PTP,LIN o CIRC</b>
2	El nombre del punto de destino se determina automáticamente, no obstante, se puede sobreescribir individualmente.  Para editar los puntos de datos, debe tocarse la flecha. Se abre la ventana de opciones <b>Vectores</b> .  En <b>CIRC</b> se debe programar un punto auxiliar adicional para el punto de destino: Aproximar la posición del punto auxiliar y accionar <b>TouchUp PI</b> . La orientación de la herramienta en el punto auxiliar no es relevante.
3	<ul style="list-style-type: none"> <li>■ <b>CONT</b>: El punto de destino es de posicionamiento aproximado.</li> <li>■ <b>[vacío]</b>: El punto de destino se alcanza con exactitud.</li> </ul>
4	<p>Velocidad</p> <ul style="list-style-type: none"> <li>■ Movimientos PTP: <b>1 ... 100 %</b></li> <li>■ Movimientos de trayectoria: <b>0.001 ... 2 m/s</b></li> </ul>
5	<p>Juego de datos de movimiento:</p> <ul style="list-style-type: none"> <li>■ Aceleración</li> <li>■ Distancia de aproximación (cuando en el campo [3] se haya introducido <b>CONT</b>).</li> <li>■ Control de la orientación (únicamente en movimientos de trayectoria)</li> </ul>

5. En la ventana de opciones Vectores, introducir los datos correctos para el sistema de coordenadas de base y herramienta, además de las indicaciones sobre el modo de interpolación (TCP externo: CON./DSECON.) y el control contra colisiones.



**Fig. 6-23: Ventana de opciones Frames**

Pos.	Descripción
1	Seleccionar herramienta.  Si se tiene <b>True</b> en el campo <b>TCP externo</b> : Seleccionar la pieza. Rango de valores: [1] ... [16]
2	Seleccionar base.  Si se tiene <b>True</b> en el campo <b>TCP externo</b> : Seleccionar herramienta fija. Rango de valores: [1] ... [32]

Pos.	Descripción
3	<p>Modo de interpolación</p> <ul style="list-style-type: none"> <li>■ <b>FALSE</b>: la herramienta está montada en la brida de acople.</li> <li>■ <b>TRUE</b>: la herramienta es una herramienta fija.</li> </ul>
4	<p>Detección de colisiones</p> <ul style="list-style-type: none"> <li>■ <b>TRUE</b>: para este movimiento, la unidad de control del robot determina los momentos axiales. Éstos son necesarios para la detección de colisión.</li> <li>■ <b>FALSE</b>: para este movimiento, la unidad de control del robot no determina momentos axiales. Por lo tanto, no es posible una detección de colisión para este movimiento.</li> </ul>

6. En la ventana de opciones Parámetros de movimiento, puede reducirse la aceleración del valor máximo. En caso de que se active la aproximación, también se puede modificar la distancia de aproximación. Por lo demás, también puede modificarse el control de orientación.

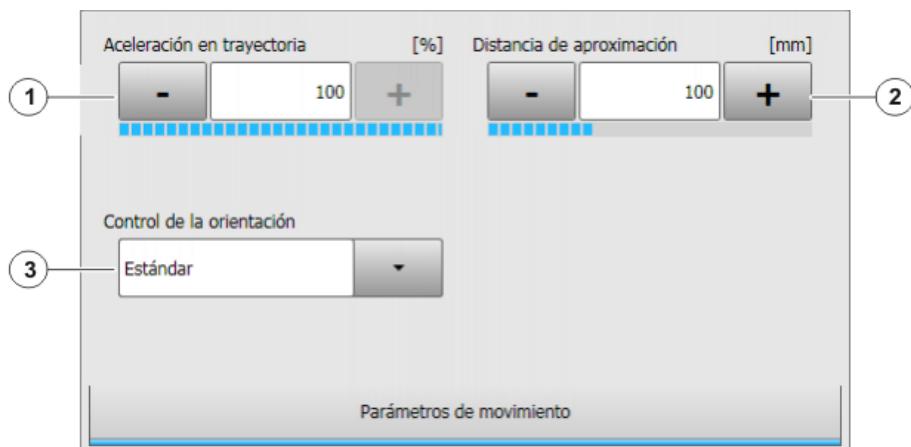


Fig. 6-24: Ventana de opciones Parámetros de movimiento (LIN, CIRC)

Pos.	Descripción
1	Aceleración  Se refiere al valor máximo indicado en los datos de la máquina. El valor máximo depende del tipo de robot y del modo de servicio ajustado.
2	Distancia de aproximación  Distancia antes del punto de destino en donde comienza, como muy pronto, la aproximación.  La distancia puede comprender, como máximo, la mitad de la distancia entre el punto de inicio y el de destino. Si se introduce un valor mayor, este se ignora y se utiliza el valor máximo.  Este campo solo se visualiza cuando se haya seleccionado <b>CONT</b> en el formulario inline.
3	Seleccionar el control de la orientación <ul style="list-style-type: none"> <li>■ <b>Estándar</b></li> <li>■ <b>PTP manual</b></li> <li>■ <b>Control orientación constante</b></li> </ul> <p>(&gt;&gt;&gt; "Control de la orientación en movimientos de trayectoria" Página 129)</p>

7. Guardar la instrucción, pulsando **Instrucción OK**. La posición opuesta del TCP se programa como punto de destino.

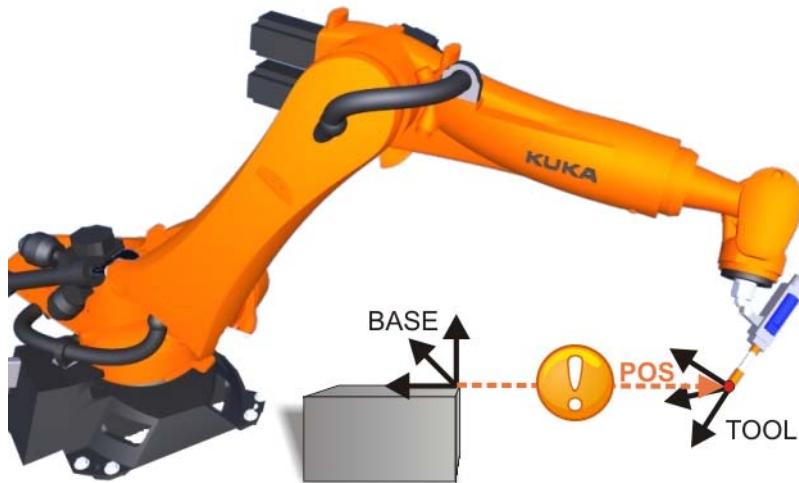


Fig. 6-25: Memorización de las coordenadas de puntos en "Instrucción OK" y "TouchUp".

## 6.6 Modificación de las instrucciones de movimiento

### Modificar instrucciones de movimiento

Existen los motivos más diversos para modificar las instrucciones del movimiento:

Ejemplos de motivos	Modificación a llevar a cabo
La posición de la pieza que se debe agarrar cambia.	Modificación de datos de posición
La posición de uno de los cinco orificios en la ejecución de los trabajos cambia.	
Una costura soldada se debe acortar.	
La posición del palet cambia.	Modificación de los datos vectoriales: Base
Por error se ha realizado el aprendizaje de una posición con la base o con el TOOL incorrectos.	Modificación de los datos vectoriales: base y/o TOOL con actualización de la posición
Los trabajos se ejecutan demasiado lentamente: se debe mejorar el tiempo de ciclo.	Modificación de los datos de movimientos: Velocidad, aceleración Modificación del tipo de movimiento

### Efectos al modificar las instrucciones de movimiento

#### Modificar los datos de posición

- Solo cambian los datos de ese punto: el punto recibe coordenadas nuevas, ya que los valores se actualizaron con "TouchUp".
- Las coordenadas anteriores se sobrescriben y, por consiguiente, dejan de estar disponible.

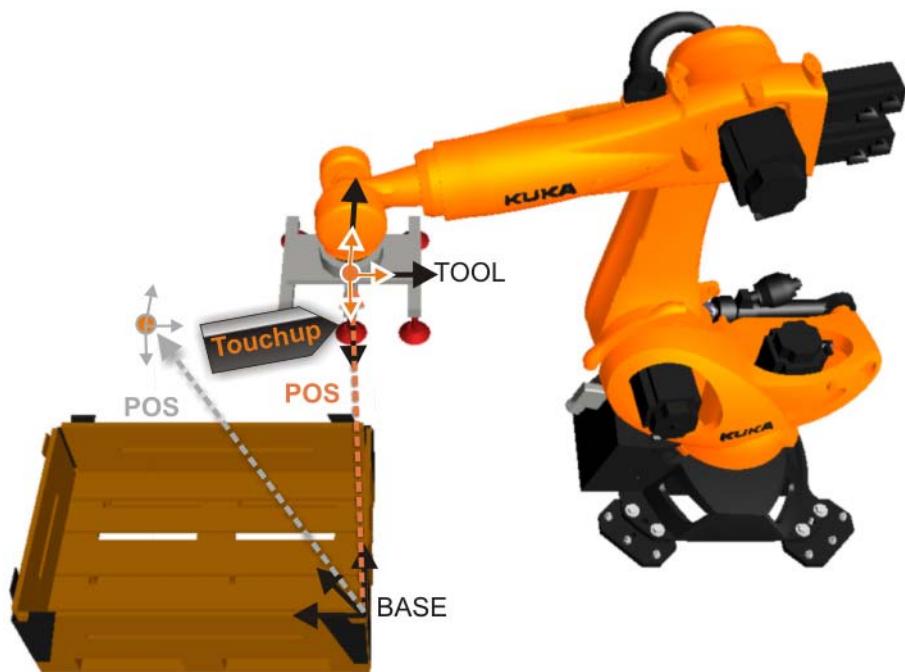


Fig. 6-26: Modificación de la posición del robot con "TouchUp"

#### Modificación de datos vectoriales

- Al cambiar los datos vectoriales (por ejemplo, Tool, Base) se produce un desplazamiento de la posición (comp. "Datos de desplazamiento")
- De ese modo cambia la posición del robot. (se corresponde con la posición angular del eje del robot)
 

Las coordenadas antiguas del punto siguen guardadas y válidas. Solo cambia la referencia (por ejemplo, la base).
- Puede sobrepasarse el campo de trabajo. Por ello el robot no puede alcanzar determinadas posiciones.
- Si el robot debe quedar en la misma posición pero, no obstante, deben cambiar los parámetros vectoriales, después de modificar los parámetros (p. ej., base) en la posición deseada, se deben actualizar las coordenadas con "TouchUp".

**i** Además, aparece un aviso: "¡Atención, al cambiar los parámetros vectoriales referentes a los puntos, existe peligro de colisión!".

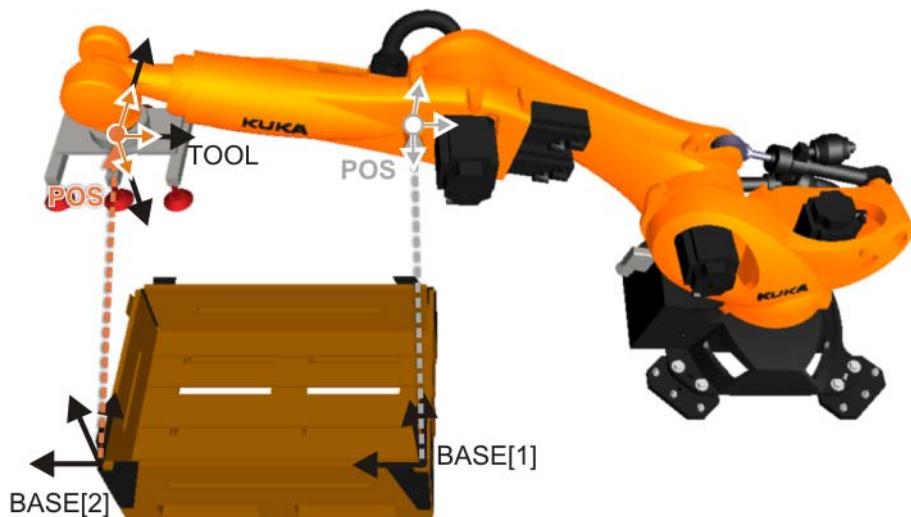


Fig. 6-27: Modificación de los datos vectoriales (ejemplo base)

## Modificar los datos de movimiento

- Al cambiar la velocidad o la aceleración cambiar el perfil de marcha. Ello puede repercutir en el proceso de fabricación, sobre todo en las aplicaciones de trayectoria:
  - Espesor de un cordón de pegado.
  - Calidad de una costura soldada.

## Cambiar el tipo de movimiento

- Al cambiar el tipo de movimiento siempre cambia el cálculo de trayectoria. En determinados casos se podrían producir colisiones porque la trayectoria podría cambiar de forma imprevisible.

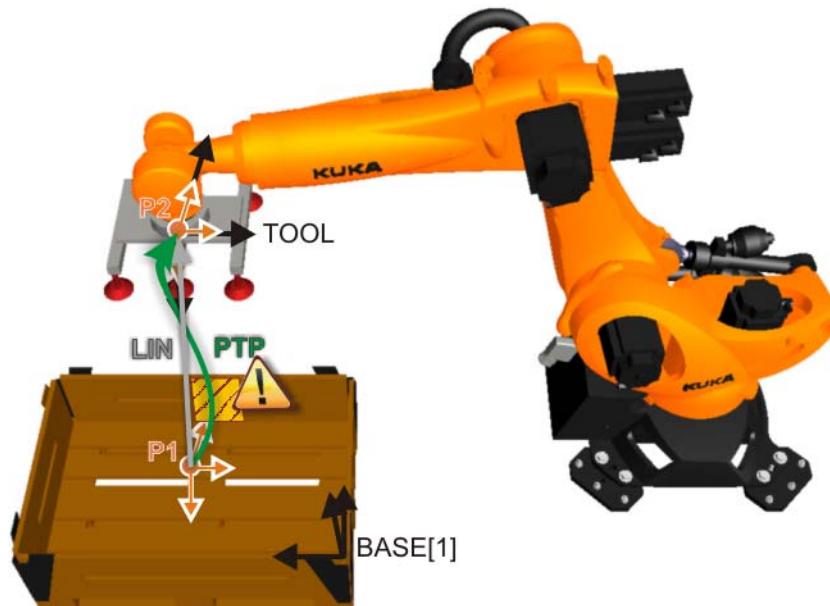


Fig. 6-28: Modificar el tipo de movimiento

### Advertencias de seguridad al modificar instrucciones de movimiento

**AVISO** En función del cambio que se debe efectuar en una instrucción del movimiento, el programa de robot se debe comprobar a velocidad reducida (modo de servicio T1). Si se inicia el programa inmediatamente a velocidad alta, el sistema y la instalación completa pueden sufrir daños porque los movimientos son imprevisibles. Si una persona se encuentra en la zona de peligro, pueden causarse lesiones con peligro de muerte.

### Modificar los parámetros de movimiento - Vectores

1. Colocar el cursor en la línea de la instrucción que se debe modificar.
2. Pulsar **Modificar**. Se abre el formulario inline para la instrucción.
3. Abrir la ventana de opciones Vectores.
4. Ajustar la nueva herramienta o base o TCP externo.
5. Confirmar el cuadro de diálogo "Atención, al cambiar los parámetros vectoriales referentes a los puntos, existe peligro de colisión" pulsando **OK**.
6. Si se desea **conservar la posición actual del robot** con los ajustes modificados de herramienta y/o base, es necesario pulsar la tecla **Touch Up** para volver a calcular y guardar la posición actual. Se deberá haber desplazado anteriormente a la posición deseada.
7. Guardar la modificación pulsando **Instrucción OK**.



Si se modifican los parámetros vectoriales, se deben volver a comprobar los programas en lo relativo a la ausencia de colisiones.

<b>Modificar posición</b>	<p>Procedimiento para modificar la posición del robot:</p> <ol style="list-style-type: none"><li>1. Ajustar el modo de servicio T1 y colocar el cursor en la línea con la instrucción que se debe modificar.</li><li>2. Colocar el robot en la posición deseada.</li><li>3. Pulsar <b>Modificar</b>. Se abre el formulario inline para la instrucción.</li><li>4. Para movimientos PTP y LIN:<ul style="list-style-type: none"><li>■ Pulsar <b>TouchUp</b> para aceptar la posición actual del TCP como nuevo punto de destino.</li></ul></li><li>Para movimientos CIRC:<ul style="list-style-type: none"><li>■ Pulsar <b>TouchUp PI</b> para aceptar la posición actual del TCP como nuevo punto auxiliar.</li><li>■ Pulsar <b>TouchUp PF</b> para aceptar la posición actual del TCP como nuevo punto de destino.</li></ul></li><li>5. Responder <b>Sí</b> a la pregunta de seguridad.</li><li>6. Guardar los cambios con <b>Instrucción OK</b>.</li></ol>
<b>Modificar los parámetros de movimiento</b>	<p>Este procedimiento sólo se puede usar para realizar las modificaciones siguientes:</p> <ul style="list-style-type: none"><li>■ Tipo de movimiento</li><li>■ Velocidad</li><li>■ Aceleración</li><li>■ Posicionamiento aproximado</li><li>■ Distancia de aproximación</li></ul> <ol style="list-style-type: none"><li>1. Colocar el cursor en la línea de la instrucción que se debe modificar.</li><li>2. Pulsar <b>Modificar</b>. Se abre el formulario inline para la instrucción.</li><li>3. Modificar parámetro.</li><li>4. Guardar la modificación pulsando <b>Instrucción OK</b>.</li></ol>
<b>Particularidades de la programación con formularios inline</b>	<p> Si se modifican parámetros de movimiento, los programas se deben volver a comprobar en lo relativo a la ausencia de colisiones y a la seguridad de proceso.</p> <p>■ <b>Renombrado de puntos</b></p> <ol style="list-style-type: none"><li>a. Marcar con el cursor el punto a renombrar.</li><li>b. Abrir el formulario inline con el botón "Modificar".</li><li>c. Modificar el nombre del punto en el formulario inline (campo Nombre del punto).</li><li>d. Aceptar las modificaciones con el botón "Instrucción OK".</li><li>e. Aceptar las modificaciones con el botón "Instrucción OK".</li><li>f. Aparece el siguiente diálogo: <i>¿Mantener las coordenadas anteriores para el punto "Nombre del punto"? (TOOL_DATA[n], BASE_DATA[n], #BASE)</i></li><li>g. Con el botón "Sí" se copian las coordenadas anteriores del punto original.</li></ol> <p>■ <b>Uso múltiple de puntos</b></p> <ol style="list-style-type: none"><li>a. Insertar un nuevo formulario inline en la posición de programa deseada.</li><li>b. Introducir el nombre del punto a reutilizar en el formulario inline (campo Nombre del punto). Tener en cuenta la correcta escritura del nombre original del punto.</li><li>c. Realizar ajustes adicionales y aceptar con el botón "Instrucción OK".</li><li>d. Aparece el siguiente diálogo: <i>El punto "Nombre del punto" ya existe: ¿sobrescribir? (TOOL_DATA[n], BASE_DATA[n], #BASE)</i></li></ol>

- e. Para no sobrescribir los parámetros existentes del punto original, pulse el botón "No".

- **Insertar líneas de programa**

- a. Marcar con el cursor una línea de programa existente.
- b. Abrir un nuevo formulario inline con el botón "Movimiento".



El nuevo formulario inline siempre se inserta después de la línea de programa marcada. Las líneas de programa siguientes se desplazan una línea hacia abajo.

- **Borrar líneas de programa**

- a. Marcar con el cursor la línea de programa a borrar.
- b. Ruta: botón **Editar > Borrar**



Los valores relativos al punto permanecen en el archivo DAT correspondiente. Ver también Uso múltiple de puntos.

## 6.7 Ejercicio: Desplazamiento de trayectoria y posicionamiento aproximado

### Objetivo del ejercicio

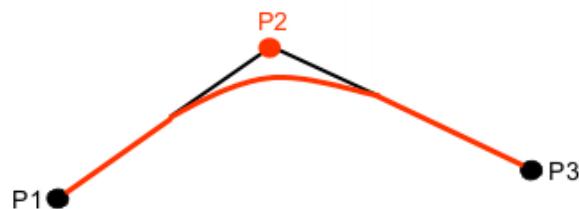
Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Creación de programas de desplazamiento sencillos con los tipos de desplazamiento PTP, LIN y CIRC
- Creación de programas de desplazamiento con puntos de parada exacta y movimientos de aproximación
- Utilización de programas en el navegador (copiar, duplicar, renombrar, borrar)

### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Principios de la programación de desplazamiento con los tipos de desplazamiento PTP, LIN, CIRC
- Conocimientos teóricos de la aproximación de movimientos

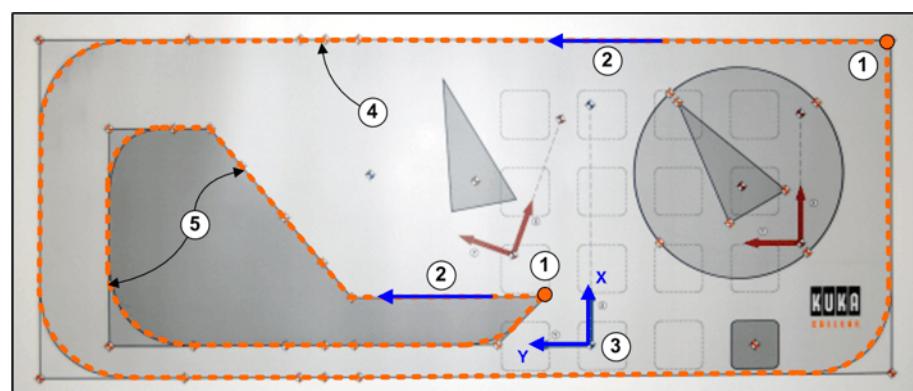


- Conocimientos teóricos de la posición inicial

### Tarea sección A

Ejecutar las siguientes tareas: Creación de programa contorno de pieza 1

1. Cree un nuevo programa con la denominación **Contorno de pieza 1**
2. Programar por aprendizaje el contorno de pieza 1, sobre la mesa de trabajo, usando la base azul y la clavija 1 como herramienta
  - La velocidad de desplazamiento sobre la mesa de trabajo es de 0,3 m/s.
  - Tener en cuenta que el eje longitudinal de la herramienta siempre se debe encontrar en posición vertical respecto al contorno de la trayectoria (control de la orientación: *estándar*)
3. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.



**Fig. 6-29: Desplazamiento de trayectoria y posicionamiento aproximado: Contorno de pieza 1 y 2**

1 Puntos de inicio

2 Dirección del movimiento

- 3 Base de referencia
- 5 Contorno de la pieza 2

- 4 Contorno de la pieza 1

**Tarea sección B**

Ejecutar las siguientes tareas: Copiar programa y aproximar

1. Cree un duplicado del programa Contorno de pieza 1 con la denominación **Pieza1\_CONT**
2. Agregar la instrucción de aproximación en las instrucciones de desplazamiento del programa nuevo de tal modo que se desplace al contorno de modo continuo.
3. Las esquinas del contorno deben desplazarse con parámetros de aproximado diversos.
4. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.

**Tarea adicional**

Ejecutar las siguientes tareas: Creación de programa contorno de pieza 2

1. Cree un segundo programa con la denominación **Contorno de pieza 2**  
Utilizar la misma base y la misma herramienta.
  - La velocidad de desplazamiento sobre la mesa de trabajo es de 0,3 m/s.
  - Tener en cuenta que el eje longitudinal de la herramienta siempre se debe encontrar en posición vertical respecto al contorno de la trayectoria (control de la orientación)
2. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.
3. Cree un duplicado del programa Contorno de pieza 2 con la denominación **Pieza2\_CONT**
4. Agregar la instrucción de aproximación en las instrucciones de desplazamiento del programa nuevo de tal modo que se desplace al contorno de modo continuo.
5. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.

**Lo que se debe saber tras el ejercicio:**

1. Cuales son las características de los movimientos LIN y CIRC?

.....  
.....  
.....

2. Como se indica la velocidad de desplazamiento de movimientos PTP, LIN y CIRC y cual es el punto de referencia de esta velocidad?

.....  
.....  
.....

3. ¿Cómo se indica la distancia de aproximación en los movimientos PTP, LIN y CIRC y a qué hace referencia esta distancia?

.....  
.....  
.....

4. ¿Qué se debe tener en cuenta al insertar instrucciones **CONT** en instrucciones de movimiento existentes?

.....  
.....  
.....

5. ¿Qué debe tenerse en cuenta al cambiar la posición inicial?

.....

6. ¿Qué se debe tener en cuenta a la hora de corregir, modificar o borrar puntos programados?

.....

.....

## 6.8 Programar movimientos con detección de colisión

### Descripción



**Fig. 6-30: Colisión**

En robótica, el control de momentos axiales se utiliza para detectar si el robot ha colisionado con un objeto. En la mayoría de los casos, esta colisión no es deseada y puede destruir el robot, la herramienta o componentes.

### Control contra colisiones

- Si un robot colisiona con un objeto, la unidad de control del robot incrementa los momentos axiales para vencer la resistencia. Esto puede dañar el robot, la herramienta u otras piezas.
- La detección de colisión reduce el riesgo y la gravedad de este tipo de daños. Controla los momentos axiales.
- El usuario puede determinar cómo se debe proceder tras una colisión después de que el algoritmo haya detectado una colisión y haya detenido el robot.
  - El robot se detiene con STOP 1.
  - La unidad de control del robot activa el programa tm\_useraction. Se encuentra en la carpeta Programa y contiene la instrucción PARADA. Como alternativa, el usuario puede programar otras reacciones en el programa tm\_useraction.
- La unidad de control del robot calcula el rango de tolerancia automáticamente.
- Por norma general, un programa se debe ejecutar entre 2 y 3 veces hasta que la unidad de control del robot calcula un rango de tolerancia adecuado para la práctica.
- El usuario puede definir un offset mediante la interfaz de usuario para el rango de tolerancia calculado por la unidad de control del robot.
- Si el robot no se encuentra en servicio durante un período prolongado (p. ej., fin de semana), los motores, los engranajes, etc. se enfrián. Durante los primeros desplazamientos tras una pausa de este tipo, son necesarios momentos axiales distintos a los de un robot a temperatura de servicio. La unidad de control del robot adapta la detección de colisión automáticamente a la temperatura modificada.

### Restricciones

- En el **modo de servicio T1** no es posible detectar una colisión.
- En el caso de las posiciones iniciales y otras posiciones globales tampoco es posible detectar una colisión.

- En el caso de los ejes adicionales tampoco se puede detectar una colisión.
- En caso de retroceso tampoco se puede detectar una colisión.
- Si el robot está parado, durante el arranque se dan momentos axiales muy altos. Por eso, en la fase de arranque (aprox. 700 ms) no se controlan los momentos axiales.
- Tras una modificación del override del programa, la detección de colisión reacciona con una insensibilidad considerable durante 2 o 3 ejecuciones del programa. Después de esto la unidad de control del robot habrá adaptado ya el rango de tolerancia al nuevo override del programa.

### Principio de la detección de colisiones

### Programar por aprendizaje un programa con detección de colisión

- La adaptación de la velocidad debe estar conectada con la variable de sistema \$ADAP\_ACC.
  - La variable del sistema se halla en el fichero C:\KRC\Roboter\KRC\R1\MaDa\\$ROBCOR.DAT.
  - \$ADAP\_ACC = #NONE: la modificación de la aceleración no está activada.
  - \$ADAP\_ACC = #STEP1: modelo dinámico sin energía cinética.
  - \$ADAP\_ACC = #STEP2: modelo dinámico con energía cinética.
- Para conectar la detección de colisión para un movimiento, el parámetro **Detección de colisión** se debe definir como TRUE durante la programación. En el código del programa, esto se puede saber por el complemento CD:

```
PTP P2 Vel= 100 % PDAT1 Tool[1] Base[1] CD
```



El parámetro **Detección de colisión** solo está disponible si el movimiento se programa mediante un formulario inline.

### Ajuste de los valores de offset

- Para el rango de tolerancia se puede definir un offset para el momento de fuerza y para el momento de impacto.
- **Momento de fuerza:** el momento de fuerza tiene efecto cuando el robot se ve sometido a una resistencia prolongada. Ejemplos:
  - El robot colisiona con una pared y hace presión contra ella.
  - El robot colisiona con un contenedor. El robot hace presión contra el contenedor y lo mueve.
- **Momento de impacto:** el momento de impacto tiene efecto cuando el robot se ve sometido a una resistencia breve. Ejemplo:
  - El robot colisiona con una placa que sale despedida por el impacto.
- Cuanto menor sea el offset, más sensible será la reacción de la detección de colisión.
- Cuanto mayor sea el offset, menos sensible será la reacción de la detección de colisión.



Si la detección de colisión reacciona con demasiada sensibilidad, no incrementar el offset de forma inmediata. En lugar de ello, primero volver a calcular el rango de tolerancia y comprobar si la detección de colisión reacciona ahora como se desea.

- Ventana de opciones: Ventana de colisión

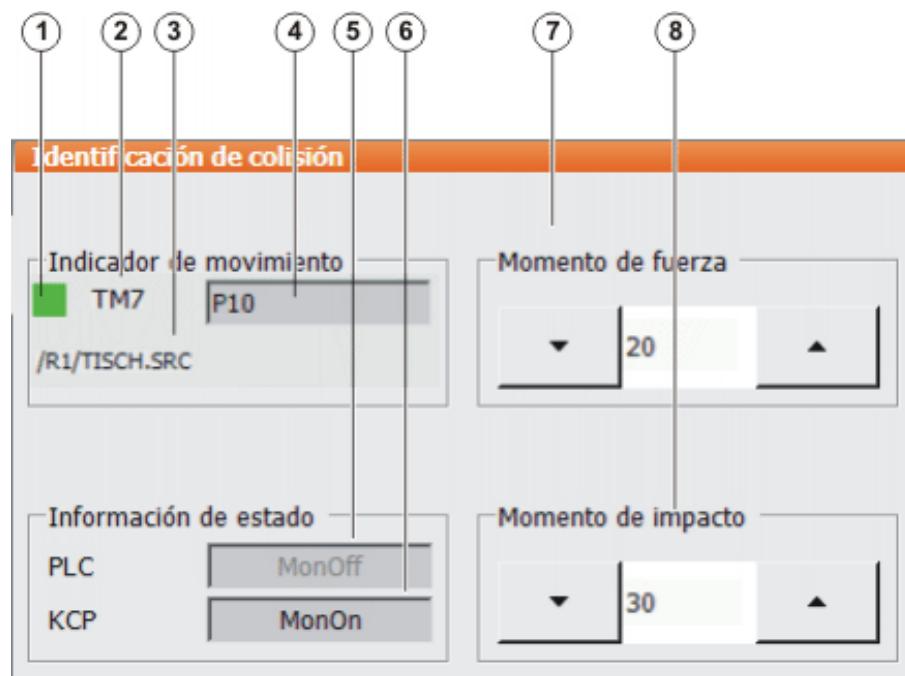


Fig. 6-31: Ventana de opciones Detección de colisión



Los datos en la ventana de opción **Detección de colisión** siempre hacen referencia al movimiento actual. Se pueden dar desviaciones, en especial en el caso de las distancias de punto cortas y de los movimientos aproximados.

Pos.	Descripción
1	<p>El botón indica el estado de un movimiento.</p> <ul style="list-style-type: none"> <li>■ Rojo: el movimiento actual no se controla.</li> <li>■ Verde: el movimiento actual sí se controla.</li> <li>■ Naranja: se ha pulsado una tecla de ajuste de valores a izquierda y derecha, junto al valor numérico del momento de fuerza o del momento de impacto. La ventana permanece en el movimiento, y se puede modificar el offset. La modificación se puede aceptar con <b>Guardar</b>.</li> <li>■ Enclavado: Por norma general, un programa se debe ejecutar entre 2 y 3 veces hasta que la unidad de control del robot calcula un rango de tolerancia adecuado para la práctica. Mientras la unidad de control del robot se encuentre en esta fase de aprendizaje, el botón se mostrará enclavado.</li> </ul>
2	<p>Número de las variables TMx</p> <p>Por cada conjunto de movimientos para los que el parámetro <b>Detección de colisión</b> sea TRUE, la unidad de control del robot crea una variable TMx. TMx contiene todos los valores para el rango de tolerancia de este conjunto de movimientos. Si 2 conjunto de movimientos hacen referencia al mismo punto Px, la unidad de control del robot creará 2 variables TMx.</p>
3	Ruta y nombre del programa seleccionado.
4	Nombre del punto

Pos.	Descripción
5	<p>Este campo sólo está activado en el modo de servicio "Automático externo". En los demás casos está desactivado (de color gris).</p> <p><b>MonOn:</b> el PLC ha activado la detección de colisión.</p> <p>Cuando se activa la detección de colisión mediante el PLC, éste envía la señal de entrada <b>sTQM_SPSACTIVE</b> a la unidad de control del robot. La unidad de control del robot responde con la señal de salida <b>sTQM_SPSSTATUS</b>. Las señales están definidas en el fichero \$config.dat.</p> <p><b>Indicación:</b> En el modo Automático externo, la detección de colisión solo está activa si, tanto en el campo <b>PLC</b> como en el campo <b>KCP</b>, se muestra la entrada <b>MonOn</b>.</p>
6	<p><b>MonOn:</b> el KCP ha activado la detección de colisión.</p> <p><b>Indicación:</b> En el modo Automático externo, la detección de colisión solo está activa si, tanto en el campo <b>PLC</b> como en el campo <b>KCP</b>, se muestra la entrada <b>MonOn</b>.</p>
7	<p>Offset para el momento de fuerza. Cuanto menor sea el offset, más sensible será la reacción de la detección de colisión. Valor por defecto: 20.</p> <p>La ventana permanece en el movimiento, y se puede modificar el offset. Ver también: (<a href="#">&gt;&gt;&gt;</a> "Adaptar el offset para movimientos" Página 147). La modificación se puede aceptar con <b>Guardar</b>.</p> <p><b>Nota:</b> Para este movimiento, la opción <b>Detección de colisión</b> en el formulario inline es FALSE.</p>
8	<p>Offset para el momento de fuerza. Cuanto menor sea el offset, más sensible será la reacción de la detección de colisión. Valor por defecto: 30.</p> <p>La ventana permanece en el movimiento, y se puede modificar el offset. Ver también: (<a href="#">&gt;&gt;&gt;</a> "Adaptar el offset para movimientos" Página 147). La modificación se puede aceptar con <b>Guardar</b>.</p> <p><b>Nota:</b> Para este movimiento, la opción <b>Detección de colisión</b> en el formulario inline es FALSE.</p>

Botón	Descripción
<b>Activar</b>	Activa la detección de colisión. Este botón no se muestra si se ha modificado el momento de fuerza o el momento de impacto pero las modificaciones no se han guardado todavía.
<b>Desactivar</b>	Desactiva la detección de colisión. Este botón no se muestra si se ha modificado el momento de fuerza o el momento de impacto pero las modificaciones no se han guardado todavía.
<b>Guardar</b>	Acepta las modificaciones del momento de fuerza y/o el momento de impacto.
<b>Cancelar</b>	Desecha las modificaciones del momento de fuerza y/o el momento de impacto.

### Activar la detección de colisión en el formulario inline

1. Crear movimiento con el formulario en línea.
2. Abrir la ventana de opciones Frames y activar la detección de colisión.

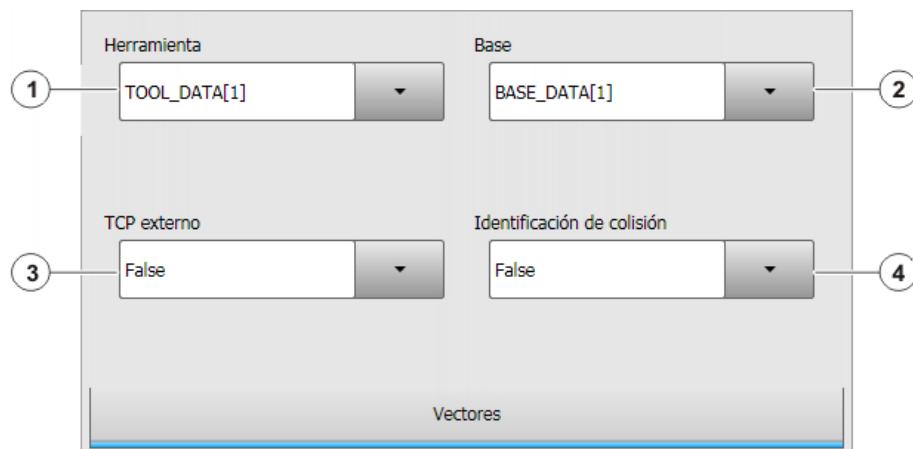


Fig. 6-32: Ventana de opciones Frames

Pos.	Descripción
1	Seleccionar herramienta. Si consta <b>True</b> en el campo <b>TCP externo</b> : Seleccionar la pieza. Rango de valores: [1] ... [16]
2	Seleccionar base. Si consta <b>True</b> en el campo <b>TCP externo</b> : Seleccionar herramienta fija. Rango de valores: [1] ... [32]
3	Modo de interpolación <ul style="list-style-type: none"> <li>■ <b>FALSE</b>: la herramienta está montada en la brida.</li> <li>■ <b>TRUE</b>: la herramienta es una herramienta fija.</li> </ul>
4	Detección de colisión <ul style="list-style-type: none"> <li>■ <b>TRUE</b>: para este movimiento, la unidad de control del robot determina los momentos axiales. Éstos son necesarios para la detección de colisión.</li> <li>■ <b>FALSE</b>: para este movimiento, la unidad de control del robot no determina momentos axiales. Por lo tanto, no es posible una detección de colisión para este movimiento.</li> </ul>

3. Finalizar el movimiento.

### Determinación del rango de tolerancia y activación de la detección de colisión

1. Seleccionar en el menú principal la secuencia **Configuración > Extras > Detección de colisión**.  
(>>> Fig. 6-31 )
2. En el campo **KCP** debe figurar la entrada **MonOff**. Si no fuera así, pulsar **Desactivar**.
3. Iniciar el programa y ejecutarlo varias veces. Al cabo de 2 o 3 ejecuciones del programa, la unidad de control del robot calcula un rango de tolerancia adecuado para la práctica.
4. Pulsar **Activar**. Ahora en la ventana **Detección de colisión**, en el campo **KCP** figura la entrada **MonOn**.  
Guardar la configuración con **Cerrar**.

### Adaptar el offset para movimientos

1. Seleccionar un programa.
2. Seleccionar en el menú principal la secuencia **Configuración > Extras > Detección de colisión**.

3. El offset para un movimiento se puede modificar mientras se ejecuta un programa: cuando el movimiento deseado se muestre en la ventana **Detección de colisión**, pulsar las teclas de flecha situadas debajo de los campos **momento de fuerza o momento de impacto**. La ventana se queda en este movimiento. Modificar el offset mediante estas teclas.

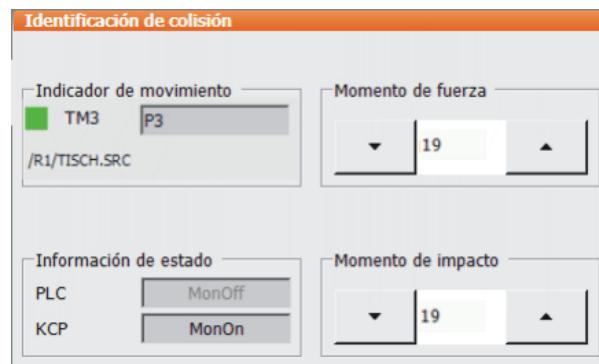


Fig. 6-35: Valores modificados de detección de colisiones

Como alternativa se puede seleccionar un paso en el movimiento deseado.

4. Aceptar el cambio con **Guardar**.
5. Guardar la configuración con **Cerrar**.
6. Ajustar el modo de servicio original y el modo de ejecución del programa.

## 7 Usar funciones lógicas en el programa del robot

### 7.1 Vista general

Esta unidad incluye los siguientes contenidos:

- Programar funciones de espera
- Programar funciones sencillas de conmutación
- Programar funciones de conmutación de trayectoria

### 7.2 Introducción a la programación lógica

**Utilización de entradas y salidas en la programación lógica**

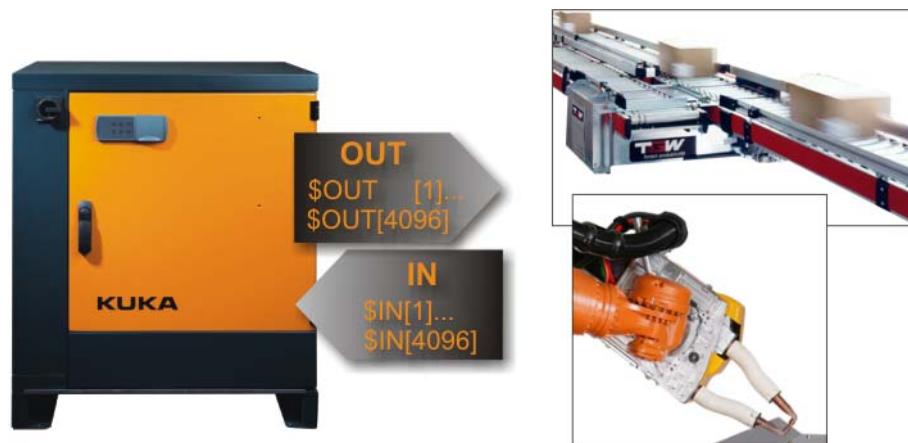


Fig. 7-1: Entradas y salidas digitales

Para establecer la **comunicación** con los **periféricos** de la unidad de control de robot, se pueden utilizar **entradas y salidas digitales y analógicas**.

Definición

Término	Explicación	Ejemplo
<b>Comunicación</b>	Intercambio de señales a través de una interfaz	Consulta de un estado (garra abierta/cerrada)
<b>Periferia</b>	"Entorno"	Herramienta (p. ej., garra, pinzas de soldadura, etc.), sensores, sistemas de transporte del material, etc.
<b>digital</b>	Tecnología digital: señales de tiempo discreto y de valor discreto	Señal de sensor: la pieza está: valor 1 (TRUE/VERDADERO), la pieza no está: valor 0 (FALSE/FALSO)
<b>analógico</b>	Representación de una dimensión física	Medición de temperatura

Término	Explicación	Ejemplo
<b>Entradas</b>	Las señales <i>procedentes</i> de la interfaz de bus de campo para el control	Señal de sensor: Garra abierta/garra cerrada
<b>Salidas</b>	Las señales <i>enviadas</i> a través de la interfaz de bus de campo para el control de los periféricos	Instrucción para la conmutación de una válvula, lo que lleva a que cierre una mordaza.

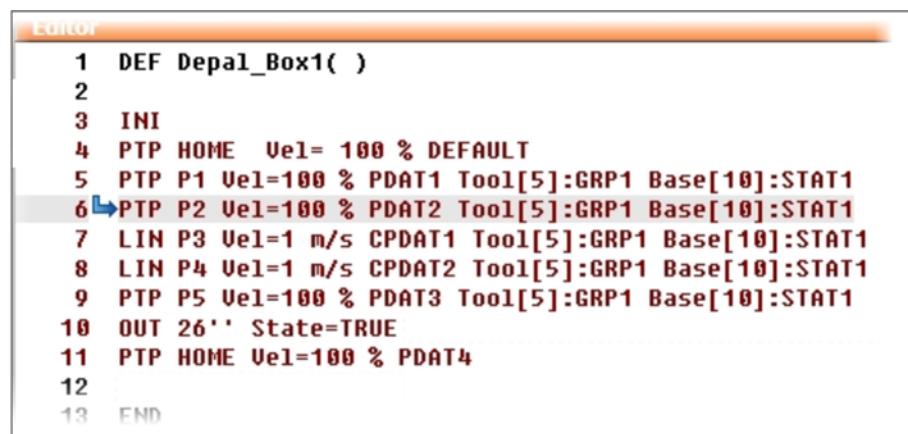
En la programación de robots de KUKA se utilizan señales de entrada y salida para instrucciones lógicas:

- **OUT** | Conexión de una salida en un lugar determinado del programa
- **WAIT FOR** | Función de espera dependiente de una señal: la unidad de mando espera una señal:
  - Entrada **IN**
  - Salida **OUT**
  - Señal horaria **TIMER**
  - Dirección de memoria interna (indicador/memoria de 1 bit) **FLAG** o **CYCFLAG** (cuando se evalúe cíclicamente de manera constante)
- **WAIT** | Función de espera dependiente del tiempo: la unidad de mando espera en esta posición del programa durante un periodo de tiempo registrado

### 7.3 Programación de funciones de espera

#### Procesamiento en avance

La computarización en movimiento de avance lee (de manera invisible para el usuario) los pasos de movimiento en avance para poder permitir el control de la planificación de la trayectoria en instrucciones de aproximación. No sólo se ejecutan datos de movimiento con el avance sino también instrucciones aritméticas y de control de los periféricos.



```

1 DEF Depal_Box1( )
2
3INI
4 PTP HOME  Vel= 100 % DEFAULT
5 PTP P1 Vel=100 % PDAT1 Tool[5]:GRP1 Base[10]:STAT1
6 PTP P2 Vel=100 % PDAT2 Tool[5]:GRP1 Base[10]:STAT1
7 LIN P3 Vel=1 m/s CPDAT1 Tool[5]:GRP1 Base[10]:STAT1
8 LIN P4 Vel=1 m/s CPDAT2 Tool[5]:GRP1 Base[10]:STAT1
9 PTP P5 Vel=100 % PDAT3 Tool[5]:GRP1 Base[10]:STAT1
10 OUT 26'' State=TRUE
11 PTP HOME  Vel=100 % PDAT4
12
13 END

```

Fig. 7-2: Procesamiento en avance

Línea	
6	Posición del puntero de ejecución principal (barra gris)
9	Possible posición del puntero de movimiento de avance (no visible)
10	Juego de instrucciones que activan una parada del movimiento de avance

Algunas instrucciones crean una parada de la ejecución en avance. Entre ellas se encuentran, entre otras, instrucciones que tengan influencia sobre la periferia, por ejemplo, instrucciones OUT (Cerrar garra, Abrir pinza de soldadura). Si se para el puntero de movimiento de avance, no es posible un posicionamiento aproximado.

### Funciones de espera

Las funciones de espera de un programa de movimiento son fácilmente programables a través de los formularios inline. Así, se distingue entre función de espera dependiente del tiempo y función de espera dependiente de señales.

Con **WAIT** se detiene el movimiento del robot durante un periodo de tiempo programado. WAIT genera una parada del procesamiento en avance.

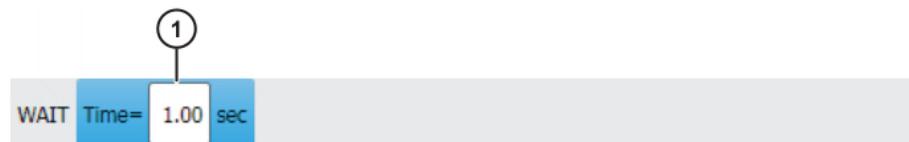


Fig. 7-3: Formulario inline WAIT

Pos.	Descripción
1	Tiempo de espera ■ $\geq 0$ s

#### Programa de ejemplo:

```
PTP P1 Vel=100% PDAT1 Tool[1] Base[1]
PTP P2 Vel=100% PDAT2 Tool[1] Base[1]
WAIT Time=2 sec
PTP P3 Vel=100% PDAT3 Tool[1] Base[1]
```

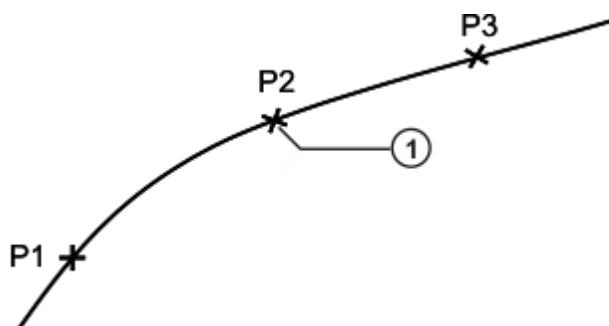


Fig. 7-4: Movimiento exemplar para lógica

Pos.	Observación
1	El movimiento se interrumpe para 2 segundos en el punto P2.

**WAIT FOR** determina una función de espera dependiente de señales.

En caso necesario, pueden combinarse de forma lógica varias señales (máximo 12). Si se agrega una combinación lógica, en el formulario inline aparecen campos para las señales adicionales y para más combinaciones.

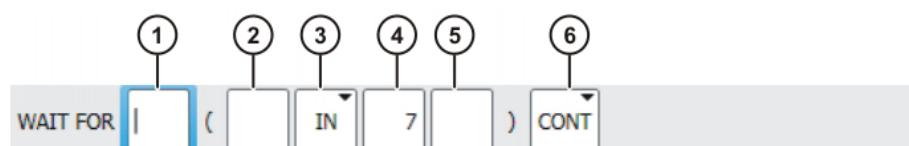


Fig. 7-5: Formulario inline WAITFOR

Pos.	Descripción
1	<p>Agregar la combinación lógica externa. El operador se ubica entre las expresiones colocadas entre paréntesis.</p> <ul style="list-style-type: none"> <li>■ <b>AND</b></li> <li>■ <b>OR</b></li> <li>■ <b>EXOR</b></li> </ul> <p>Agregar NOT.</p> <ul style="list-style-type: none"> <li>■ <b>NOT</b></li> <li>■ <b>[vacío]</b></li> </ul> <p>Agregar el operador deseado utilizando el correspondiente botón.</p>
2	<p>Agregar la combinación lógica interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <ul style="list-style-type: none"> <li>■ <b>AND</b></li> <li>■ <b>OR</b></li> <li>■ <b>EXOR</b></li> </ul> <p>Agregar NOT.</p> <ul style="list-style-type: none"> <li>■ <b>NOT</b></li> <li>■ <b>[vacío]</b></li> </ul> <p>Agregar el operador deseado utilizando el correspondiente botón.</p>
3	<p>Señal que se está esperando</p> <ul style="list-style-type: none"> <li>■ <b>IN</b></li> <li>■ <b>OUT</b></li> <li>■ <b>CYCFLAG</b></li> <li>■ <b>TIMER</b></li> <li>■ <b>FLAG</b></li> </ul>
4	<p>Número de la señal</p> <ul style="list-style-type: none"> <li>■ <b>1 ... 4096</b></li> </ul>
5	<p>Si la señal ya tiene nombre, éste se muestra.</p> <p>Sólo para el grupo de expertos:</p> <p>Pulsando en <b>Texto largo</b> puede introducirse un nombre. Se puede elegir cualquier nombre.</p>
6	<ul style="list-style-type: none"> <li>■ <b>CONT</b>: procesamiento en el movimiento de avance</li> <li>■ <b>[vacío]</b>: procesamiento con parada del movimiento de avance</li> </ul>



Durante la utilización de la entrada CONT debe asegurarse de que se va a consultar la señal en el movimiento de avance. Las modificaciones en la señal no se reconocen una vez transcurrido el periodo de avance.

## Combinaciones lógicas

En la utilización de funciones de espera dependientes de señales se pueden utilizar también combinaciones lógicas. Gracias a los operadores lógicos se pueden combinar las consultas de diferentes señales o estados: p. ej., pueden establecerse dependencias o excluirse determinados estados.

El resultado de una función con un operador lógico proporciona siempre un valor de verdad, es decir, el resultado final siempre es "verdadero" (valor 1) o "falso" (valor 0).



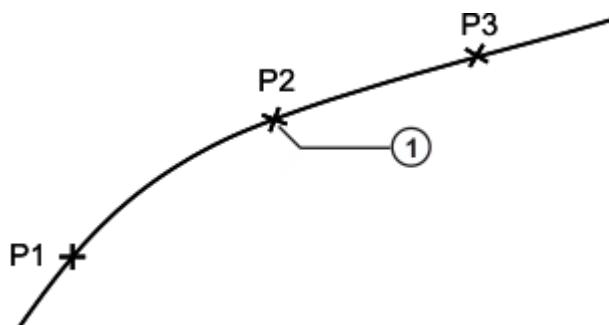
**Fig. 7-6: Ejemplo y principio de una combinación lógica**

Los **operadores** para las combinaciones lógicas son:

- **NOT** | Este operador se utiliza para la negación, es decir, el valor se invierte (de "verdadero" pasa a "falso").
- **AND** | el resultado de la expresión es verdadero si ambas expresiones combinadas son verdaderas.
- **OR** | el resultado de la expresión es verdadero si al menos una de las expresiones combinadas es verdadera.
- **EXOR** | el resultado de la expresión es verdadero cuando ambas afirmaciones combinadas por este operador presentan diferentes valores de verdad.

### Procesamiento con y sin movimiento de avance (CONT)

Las funciones de espera dependientes de señales pueden programarse con y sin su procesamiento en movimiento de avance. **Sin movimiento de avance** significa que, en cada caso, se detiene el movimiento en el punto y se controla la señal: (1) (=> Fig. 7-7). El punto tampoco puede repasararse.



**Fig. 7-7: Movimiento ejemplar para lógica**

```

PTP P1 Vel=100% PDAT1 Tool[1] Base[1]
PTP P2 CONT Vel=100% PDAT2 Tool[1] Base[1]
WAIT FOR IN 10 'door_signal'
PTP P3 Vel=100% PDAT3 Tool[1] Base[1]

```



Al procesar una línea **WAIT FOR** sin **CONT**, aparece un mensaje de aviso "Aproximación no posible".

**Con movimiento de avance**, las funciones de espera dependientes de señales programadas permiten que se pueda repasar antes del punto creado antes de la línea de instrucción. No obstante, la anterior posición del puntero de movimiento de avance no es clara (valor estándar: tres conjuntos de movimientos), por lo que el momento exacto para el control de la señal es indefinido (1) (=> Fig. 7-8). Además, no se reconocerán las modificaciones en la señal tras el control de la señal.

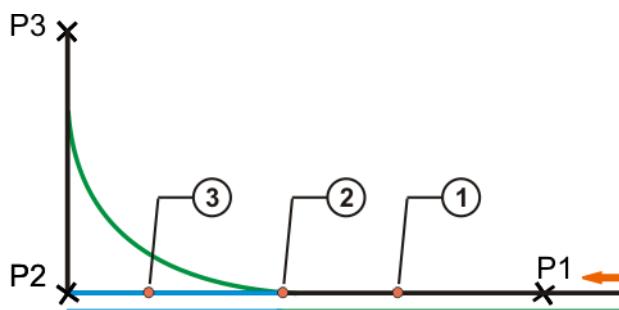


Fig. 7-8: Movimiento ejemplar para lógica con avance

	Posición	Rango de comutación
1	Posición del robot en la que se cumple la condición para que el robot realice un posicionamiento aproximado	Rango de comutación verde para la Activación del contorno de aproximación. Solo se ajusta y no se puede volver a desactivar.
2	Inicio del movimiento de aproximación	Consulta de la Activación <ul style="list-style-type: none"> <li>■ TRUE: Posicionamiento aproximado</li> <li>■ FALSE: Acercamiento al punto de destino</li> </ul>
3	Posición del robot en la que se cumple la condición para que el robot no realice un posicionamiento aproximado	Rango de comutación azul para el acercamiento y la parada en el punto P2

```

PTP P1 Vel=100% PDAT1 Tool[1] Base[1]
PTP P2 CONT Vel=100% PDAT2 Tool[1] Base[1]
WAIT FOR IN 10 'door_signal' CONT
PTP P3 Vel=100% PDAT3 Tool[1] Base[1]
  
```

#### Procedimiento

1. Colocar el cursor en la línea detrás de la cual se insertará la instrucción lógica.
2. Seleccionar la secuencia de menú **Instrucciones > Lógica > WAIT FOR o WAIT**.
3. Declarar los parámetros en el formulario inline.
4. Guardar la instrucción pulsando **Instruc.OK**.

## 7.4 Programación de funciones de conmutación simples

### Función de conmutación simple

Mediante una acción de conmutación puede enviarse una señal digital a los periféricos. Para ello, se utiliza el número de salida que se definió previamente para la interfaz.

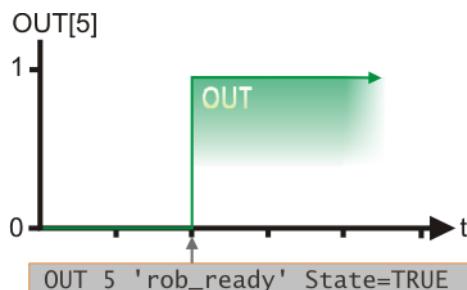


Fig. 7-9: Conectar estática

La señal será estática, es decir, se mantiene en tanto que la salida se ocupe con otro valor. La función de conmutación se ejecuta en el programa mediante un formulario inline:

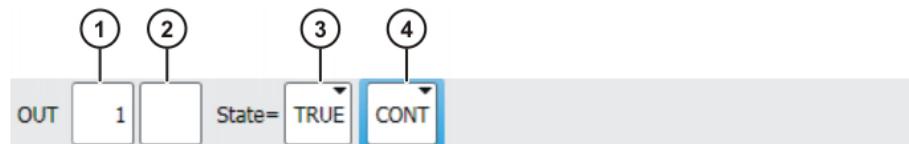


Fig. 7-10: Formulario inline OUT

Pos.	Descripción
1	Número de la salida ■ 1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra. Sólo para el grupo de expertos: Pulsando en <b>Texto largo</b> puede introducirse un nombre. Se puede elegir cualquier nombre.
3	Estado en el que la salida se conecta ■ TRUE ■ FALSE
4	■ CONT: procesamiento en el movimiento de avance ■ [vacío]: procesamiento con parada del movimiento de avance



Durante la utilización de la entrada CONT debe asegurarse de que se va a establecer la señal en el movimiento de avance.

### Funciones pulsadas de conmutación

Tal y como se produce en la función de conmutación simple, aquí también se modifica el valor para una salida. No obstante, con las pulsaciones se puede anular la señal transcurrido un periodo de tiempo definido.

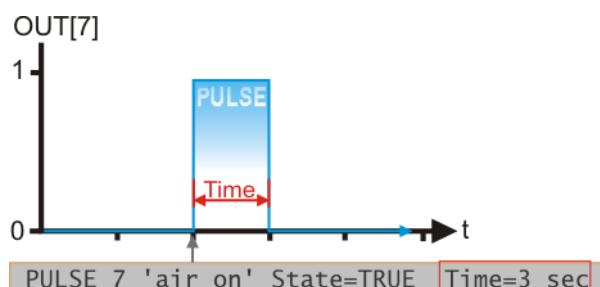


Fig. 7-11: Nivel pulsado

La programación se realiza también con un formulario inline en el que se determina un impulso con una duración determinada.

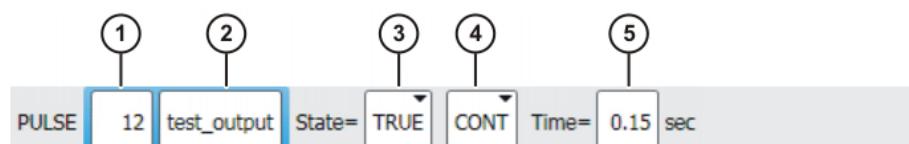


Fig. 7-12: Formulario inline PULSE

Pos.	Descripción
1	Número de la salida ■ 1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra. Sólo para el grupo de expertos: Pulsando en <b>Texto largo</b> puede introducirse un nombre. Se puede elegir cualquier nombre.
3	Estado en el que la salida se conecta ■ <b>TRUE</b> : Nivel "High" ■ <b>FALSE</b> : Nivel "Low"
4	■ <b>CONT</b> : procesamiento en el movimiento de avance ■ <b>[vacío]</b> : procesamiento con parada del movimiento de avance
5	Duración del pulso ■ <b>0.10 ... 3.00 s</b>

### Efectos de CONT en las funciones de conmutación

#### sin CONT:

Si se suprime la entrada CONT en el formulario inline OUT, se fuerza una **parada del movimiento de avance** en el proceso de conmutación y se produce una parada exacta en el punto antes de la instrucción de conmutación. Tras establecer la salida se continúa con el movimiento.

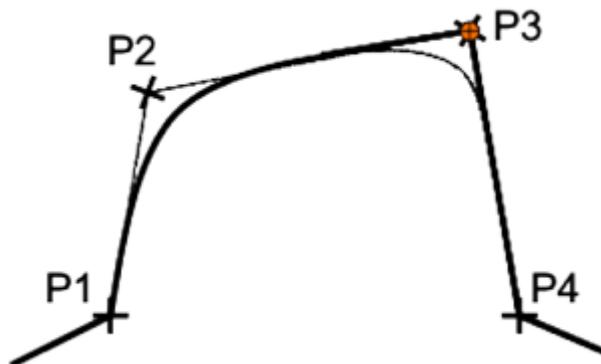


Fig. 7-13: Movimiento ejemplar con conmutación con parada del procesamiento en avance

```

LIN P1 Vel=0.2 m/s CPDAT1 Tool[1] Base[1]
LIN P2 CONT Vel=0.2 m/s CPDAT2 Tool[1] Base[1]
LIN P3 CONT Vel=0.2 m/s CPDAT3 Tool[1] Base[1]
OUT 5 'rob_ready' State=TRUE
LIN P4 Vel=0.2 m/s CPDAT4 Tool[1] Base[1]

```

#### con CONT:

El efecto producido por el establecimiento de la entrada CONT es que el puntero de movimiento de avance no se va a mantener (no se provoca ninguna parada del movimiento de avance). De este modo se puede aproximar un movimiento antes de la instrucción de conmutación. El establecimiento de la señal se produce en el **movimiento de avance**.

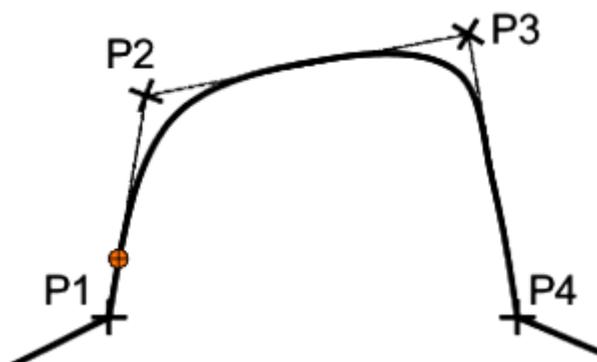


Fig. 7-14: Movimiento ejemplar con conmutación en el avance

```
LIN P1 Vel=0.2 m/s CPDAT1 Tool[1] Base[1]
LIN P2 CONT Vel=0.2 m/s CPDAT2 Tool[1] Base[1]
LIN P3 CONT Vel=0.2 m/s CPDAT3 Tool[1] Base[1]
OUT 5 'rob_ready' STATE=TRUE CONT
LIN P4 Vel=0.2 m/s CPDAT4 Tool[1] Base[1]
```



El valor estándar para el puntero de movimiento de avance es de tres conjuntos de movimientos. No obstante, el movimiento de avance puede variar, es decir, debe tenerse en cuenta que el momento de conmutación no siempre es el mismo.

#### Procedimiento

1. Colocar el cursor en la línea detrás de la cual se insertará la instrucción lógica.
2. Seleccionar la secuencia de menú **Instrucciones > Lógica > OUT > OUT o PULSE**.
3. Declarar los parámetros en el formulario inline.
4. Guardar la instrucción pulsando **Instruc.OK**.

## 7.5 Programación de funciones de conmutación de trayectoria

#### Generalidades

Es posible usar una función de conmutación de trayectoria para fijar una salida en un punto concreto de la trayectoria sin tener que interrumpir el movimiento del robot. En este caso, se distingue entre la conmutación "estática" (SYN OUT) y la "dinámica" (SYN PULSE). Así, para la conmutación de un SYN OUT 5 se utiliza la misma señal que para un SYN PULSE 5, pero el modo en que se va a conectar es diferente. Las opciones de selección PATH, START y END se explican a continuación.

En la práctica se usa principalmente la opción PATH, por lo que es la que se estudia con detenimiento.

Las opciones START y END se incluyen en la documentación para completar.

#### Opción PATH

Con la opción PATH es posible activar una función de conmutación con referencia al punto de destino de un paso de movimiento. La acción de conmutación puede ser desplazada en el espacio y/o en el tiempo. El paso de movimiento de referencia puede ser un movimiento LIN o CIRC. **No** puede ser un movimiento PTP.

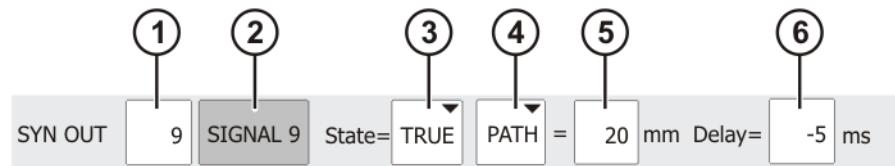


Fig. 7-15: Formulario inline SYN OUT, opción Path

Pos.	Descripción	Rango de valores
1	Número de la salida	1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra. Sólo para el grupo del expertos: Pulsando la tecla de función <b>Texto largo</b> puede introducirse un nombre.	A libre elección
3	Estado, al cual se ha de activar o colocar la salida.	TRUE, FALSE
4	Punto en el que se conecta <b>PATH</b> : se comunica referido al punto de destino del paso de movimiento.	Opción PATH START, END
5	Desplazamiento en el espacio de la acción de conmutación <b>Indicación</b> : la indicación del lugar hace clara referencia al punto de destino del paso de movimiento. Así, la posición del punto de conmutación no cambia modificando la velocidad del robot. De esta forma se compensa el tiempo de reacción del dispositivo activado (p. ej. pistola de pegamento).	-2000 ... +2000 mm
6	Desplazamiento en el tiempo de la acción de conmutación <b>Indicación</b> : el desplazamiento en el tiempo está relacionado con el desplazamiento en el espacio. La indicación de tiempo es absoluta, es decir, el punto de conmutación cambia en función de la velocidad del robot.	-1000 ... +1000 ms

Con SYN PULSE se puede activar **un impulso** en el punto de inicio o en el punto de destino del movimiento. El impulso puede desplazarse en el espacio y/o en el tiempo: es decir, no debe activarse exactamente en el punto, sino se puede activar antes o después.



Fig. 7-16: Formulario inline SYN PULSE

Pos.	Descripción	Rango de valores
1	Número de la salida	1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra.  Sólo para el grupo de expertos:  Pulsando en <b>Texto largo</b> puede introducirse un nombre. Se puede elegir cualquier nombre.	A libre elección
3	Estado en el que la salida se conecta	TRUE, FALSE
4	Duración del impulso	0.1 ... 3s
5	Puntos a los que hace referencia  <b>SYN PULSE:</b>  ■ <b>PATH:</b> SYN PULSE hace referencia al punto de destino. Adicionalmente, se puede realizar un desplazamiento en el espacio.	Opción PATH  START, END
6	Desplazamiento en el espacio de la acción de conmutación  <b>Indicación:</b> la indicación del lugar hace clara referencia al punto de destino del paso de movimiento. Así, la posición del punto de conmutación no cambia modificando la velocidad del robot.	-2000 ... +2000 mm
7	Desplazamiento en el tiempo de la acción de conmutación  <b>Indicación:</b> el desplazamiento en el tiempo está relacionado con el desplazamiento en el espacio.  El tiempo se indica en valores absolutos. El punto de conmutación cambia en función de la velocidad del robot.	1000 ... +1000 ms

### Efecto de la opción de comunicación Path

Programa de ejemplo:

Debe conectarse una fresadora en la trayectoria. El procesamiento del componente debe iniciarse libremente 20 mm tras P2. Para que la fresadora alcance su velocidad máxima a 20 mm de P2 (path = 20), debe haberse conectado 5 ms antes (retraso = -5 ms).



La opción de comunicación **Path** se refiere siempre a **Punto de destino**.

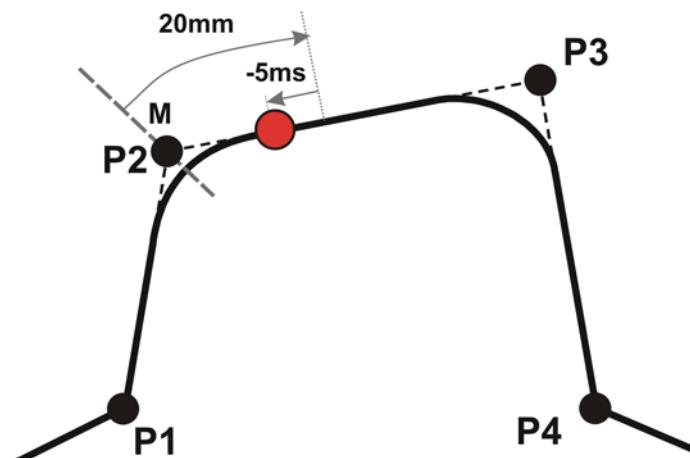


Fig. 7-17: Punto de conmutación SYN OUT Path

```
LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
;función de conmutación referenciada a P2
SYN OUT 9 'SIGNAL 9' Status= True Path=20 Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
LIN P3 CONT VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]
```

#### Límites de conmutación

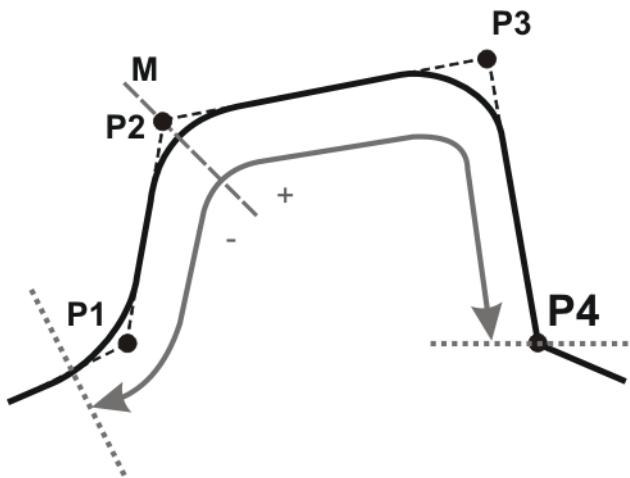


Fig. 7-18: Límites de conmutación SYN OUT PATH

#### Procedimiento

1. Colocar el cursor en la línea detrás de la cual se insertará la instrucción lógica.
2. Seleccionar la secuencia de menú **Instrucciones > Lógica > OUT > SYN OUT o SYN PULSE**.
3. Declarar los parámetros en el formulario inline.
4. Guardar la instrucción pulsando **Instruc.OK**.

#### Opción Start/End

Una función de conmutación puede ser disparada con referencia al punto de arranque o al de destino dentro de un paso de movimiento. La función de conmutación puede **aplazarse en el tiempo**. El paso de movimiento de referencia puede ser un movimiento LIN, CIRC o PTP.

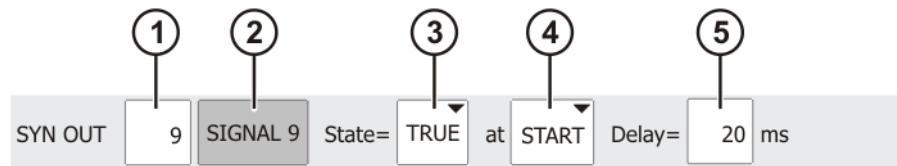


Fig. 7-19: Formulario inline SYN OUT, opción START

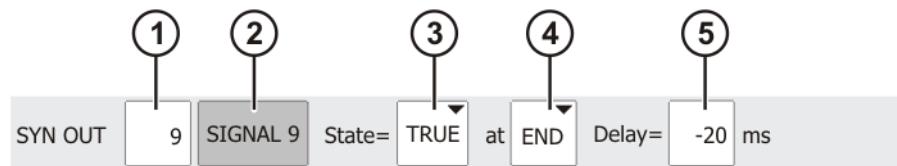


Fig. 7-20: Formulario inline SYN OUT, opción END

Pos.	Descripción	Rango de valores
1	Número de la salida	1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra.  Sólo para el grupo del expertos: Pulsando la tecla de función <b>Texto largo</b> puede introducirse un nombre.	A libre elección
3	Estado, al cual se ha de activar o colocar la salida.	TRUE, FALSE
4	Punto en el que se conecta <ul style="list-style-type: none"> <li>■ <b>START</b>: se conmuta referido al punto de inicio del paso de movimiento.</li> <li>■ <b>END</b>: se conmuta referido al punto de destino del paso de movimiento.</li> </ul>	START, END PATH ( <a href="#">&gt;&gt;&gt; Fig. 7-15</a> )
5	Desplazamiento en el tiempo de la acción de conmutación  <b>Indicación</b> : el tiempo se indica en valores absolutos. Por tanto, la posición del punto de conmutación se modifica en función de la velocidad del robot.	-1000 ... +1000 ms

Efecto de la opciones de conmutación Start/End

- **Programa de ejemplo 1:**  
*opción Start*

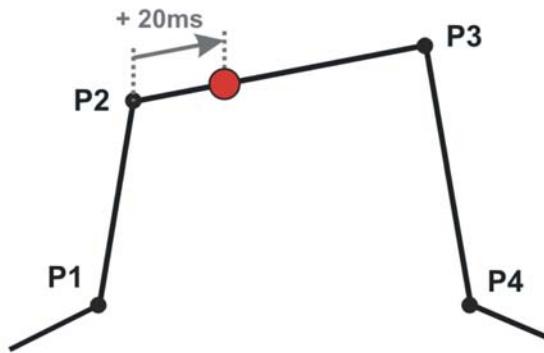


Fig. 7-21: Inicio SYN OUT con retardo positivo

```
LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
LIN P2 VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
;función de conmutación referenciada a P2
SYN OUT 8 'SIGNAL 8' State= TRUE at Start Delay=20ms
LIN P3 VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]
```

#### ■ Programa de ejemplo 2:

Opción Start con CONT y retraso positivo

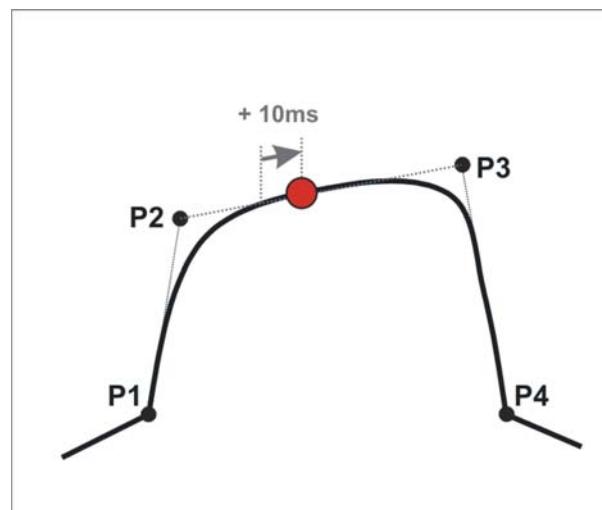
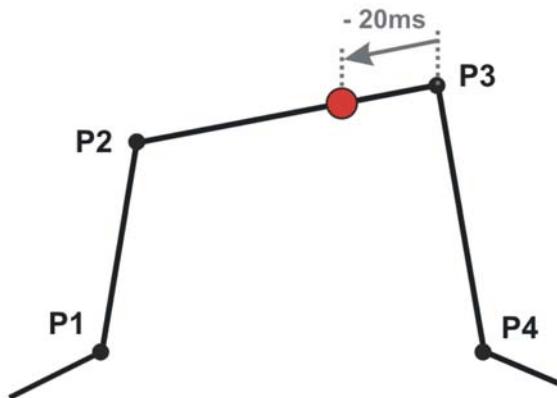


Fig. 7-22: Arranque SYN OUT con CONT y retraso positivo

```
LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
LIN P2 CONT VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
;función de conmutación referenciada a P2
SYN OUT 8 'SIGNAL 8' State= TRUE at Start Delay=10ms
LIN P3 CONT VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]
```

#### ■ Programa de ejemplo 3:

Opción End con retraso negativo



**Fig. 7-23: SYN OUT END con retardo negativo**

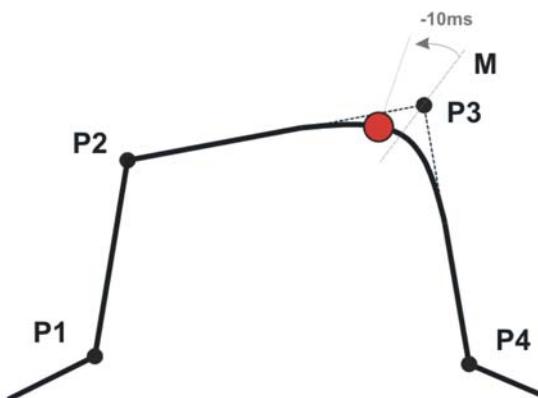
```

LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
LIN P2 VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
; función de conmutación referenciada a P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=-20ms
LIN P3 VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]

```

■ **Programa de ejemplo 4:**

Opción *End* con *CONT* y retraso negativo



**Fig. 7-24: SYN OUT con opción END con retraso negativo**

```

LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
LIN P2 VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
; función de conmutación referenciada a P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=-10ms
LIN P3 CONT VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]

```

■ **Programa de ejemplo 5:**

Opción *End* con *CONT* y retraso positivo

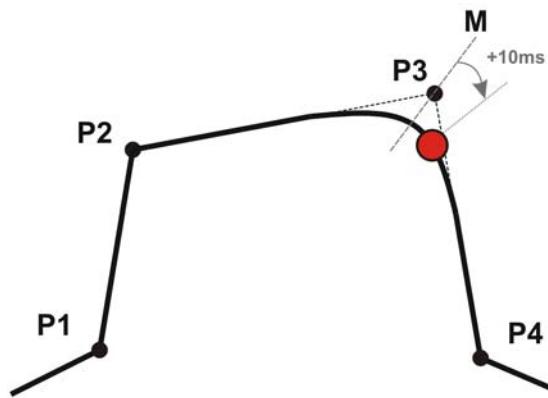


Fig. 7-25: SYN OUT con opción END con retraso positivo

```

LIN P1 VEL=0.3m/s CPDAT1 TOOL[1] BASE[1]
LIN P2 VEL=0.3m/s CPDAT2 TOOL[1] BASE[1]
;función de comutación referenciada a P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=10ms
LIN P3 CONT VEL=0.3m/s CPDAT3 TOOL[1] BASE[1]
LIN P4 VEL=0.3m/s CPDAT4 TOOL[1] BASE[1]

```

**Límites de  
comutación en  
funciones de  
comutación de  
trayectoria**

■ **Límites de comutación  
sin CONT**

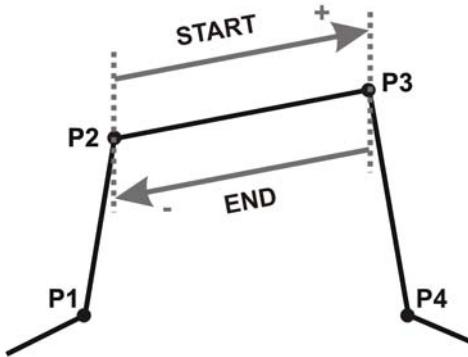


Fig. 7-26: Límites de comutación, opción Start/End sin CONT

■ **Límites de comutación con CONT:  
con CONT**

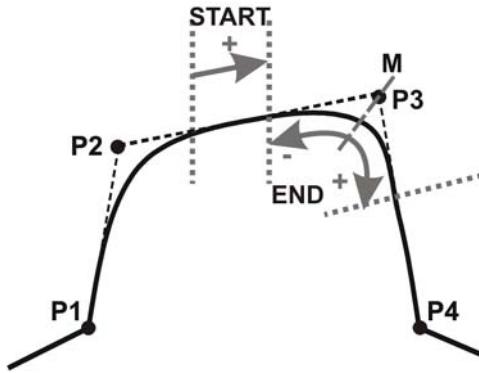


Fig. 7-27: Límites de comutación, opción Start/End con CONT

## 7.6 Ejercicio: Instrucciones lógicas y funciones de conmutación

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"> <li>■ Programar instrucciones lógicas sencillas</li> <li>■ Efectuar funciones de conmutación sencillas</li> <li>■ Efectuar funciones de conmutación referidas a la trayectoria</li> <li>■ Programar funciones de espera dependientes del estado de señales</li> </ul>
<b>Requisitos</b>	Los siguientes requisitos son necesarios para completar este ejercicio correctamente: <ul style="list-style-type: none"> <li>■ Conocimientos teóricos de la programación de instrucciones lógicas sencillas           <ul style="list-style-type: none"> <li>■ Conocimientos de funciones de conmutación sencillas</li> <li>■ Conocimientos de funciones de impulso sencillas</li> <li>■ Conocimientos de funciones de conmutación en función de la trayectoria</li> <li>■ Conocimientos de funciones de impulso en función de la trayectoria</li> <li>■ Conocimientos de funciones de espera</li> </ul> </li> </ul>
<b>Formulación de tarea</b>	<p>Ejecutar las siguientes tareas: Programación lógica contorno de pieza 1 con aplicación de adhesivo</p> <ol style="list-style-type: none"> <li>1. Crear un duplicado del programa Componente1_CONT con la denominación <b>Contorno de pegado</b>.</li> <li>2. Amplíe el programa por la funcionalidad lógica siguiente:           <ul style="list-style-type: none"> <li>■ La posición inicial no se puede abandonar hasta que no se disponga de la señal de liberación del PLC (entrada 11).</li> <li>■ 0,5 segundos antes de alcanzar la pieza, la boquilla de pegamento debe activarse (salida 13).</li> <li>■ En el punto de transición del plano a la curvatura de la pieza debe conmutarse una lámpara de señal que debe apagarse de nuevo en el punto de transición de la curvatura al plano (salida 12).</li> <li>■ 0,75 segundos antes de abandonar la pieza, la boquilla de pegamento debe desactivarse de nuevo (salida 13).</li> <li>■ 50 mm antes de terminar el tratamiento de la pieza, el PLC debe recibir un aviso de tarea finalizada. La señal (salida 11) para el PLC debe estar presente durante 2 segundos.</li> </ul> </li> <li>3. Compruebe su programa según prescripción.</li> </ol>

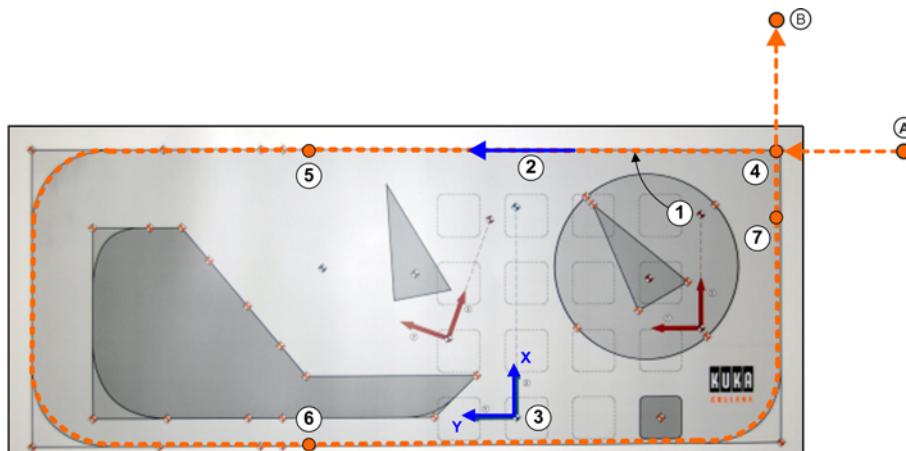


Fig. 7-28: Entradas y salidas: Aplicación de adhesivo

A	Posición inicial	B	Fin de ejecución
1	Contorno de la pieza 1	2	Dirección del movimiento
3	Base de referencia	4	Punto de inicio y punto final del componente
5	Transición plano - curvatura	6	Transición curvatura - plano
7	Punto antes del extremo de la pieza		

### Preguntas sobre el ejercicio

1. ¿Cuál es la diferencia entre las instrucciones OUT y OUT CONT? ¿Qué debe tenerse en cuenta?

.....

2. ¿En qué se diferencian las instrucciones PULSE y OUT?

.....

3. ¿Cuando se utilizan las indicaciones SYN OUT?

.....

4. ¿Existen restricciones para las instrucciones SYN OUT Path en relación con la programación de movimiento?

.....

5. ¿Para qué sirve el tiempo de retraso en las instrucciones SYN OUT con PATH?

6. ¿Cuáles son los peligros que se presentan al utilizar la instrucción WAIT FOR con una instrucción CONT?

.....

## 8 Utilizar paquetes tecnológicos

### 8.1 Vista general

**Esta unidad incluye los siguientes contenidos:**

- Manejo de KUKA.GripperTech
- Programación con KUKA.GripperTech
- Configuración de KUKA.GripperTech

### 8.2 Manejo de la garra con KUKA.GripperTech

**Paquete de tecnología** KUKA.Gripper&SpotTech es un paquete software de tecnología para instalación posterior. Simplifica la utilización de una garra con respecto a:

- KUKA.GripperTech**
- **Manejo**  
Los estados de la garra se verifican manualmente con las teclas de estado.
  - **Programación**  
Programación sencilla de instrucciones de la garra mediante formularios inline predefinidos.
  - **Configuración**  
Ajuste de una garra mediante cinco tipos de garra predefinidos o un tipo de garra definido por el usuario.

Para el servicio de la garra son necesarias las siguientes teclas de estado:

Tecla de estado	Descripción
 1	Seleccionar la garra.  Se muestra el número de la garra. <ul style="list-style-type: none"> <li>■ Al accionar la tecla superior se cuentan empezando desde arriba.</li> <li>■ Al accionar la tecla inferior se cuentan empezando desde abajo.</li> </ul>
	Cambiar entre los diferentes estados de la garra (p. ej., abrir o cerrar).  No se muestra el estado actual. Los estados disponibles dependen del tipo de garra configurado. En el caso de pinzas de soldadura: los estados disponibles dependen de la configuración del control manual de la pinza.

**Procedimiento para el manejo de las garras**

**AVISO** Antes de poder manejar una garra con las teclas de estado, estas deberán activarse.  
Seleccionar en el menú principal **Configuración > Teclas de estado > GripperTech**.

**ADVERTENCIA** **Advertencia** Al utilizar el sistema de garras existe riesgo de aplastamiento y corte. El personal que maneje las garras debe asegurarse de que estas no puedan aplastar ninguna parte del cuerpo.

1. Seleccionar la garra con la tecla de estado.



2. Activar el modo de servicio T1 o T2.
3. Accionar el pulsador de validación
4. Controlar la garra con la tecla de estado



### 8.3 Programación de garra con KUKA.GripperTech

#### Programación de instrucciones de la garra

Con el paquete de tecnología KUKA.GripperTech es posible programar, directamente en el programa seleccionado, las instrucciones de la garra utilizando formularios inline predefinidos. Existen dos instrucciones disponibles:

- **SET Gripper** | Instrucción para abrir/cerrar la garra en el programa
- **CHECK Gripper** | Instrucción para comprobar la posición de la garra

#### Funciones de la programación de garra

##### Instrucción de garra durante el movimiento

- Básicamente, es posible programar la instrucción de garra de tal manera que se ejecute respecto al punto de inicio o de destino.
- Por tanto, en el formulario inline únicamente necesario activar la entrada **CONT** e introducir la duración del retraso en ms (**Retraso**)
- 

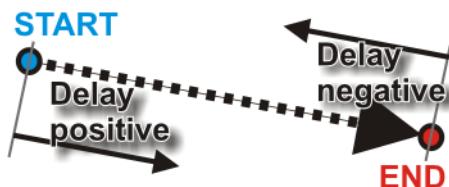


Fig. 8-1: Esquema de retardo

#### **ADVERTENCIA**

Una instrucción de garra con procesamiento durante el movimiento debe ser seleccionada cuidadosamente, ya que el uso imprudente puede provocar daños personales o materiales por piezas volantes o colisiones.

#### Ajustes de garra a aplicar durante la parada exacta

-

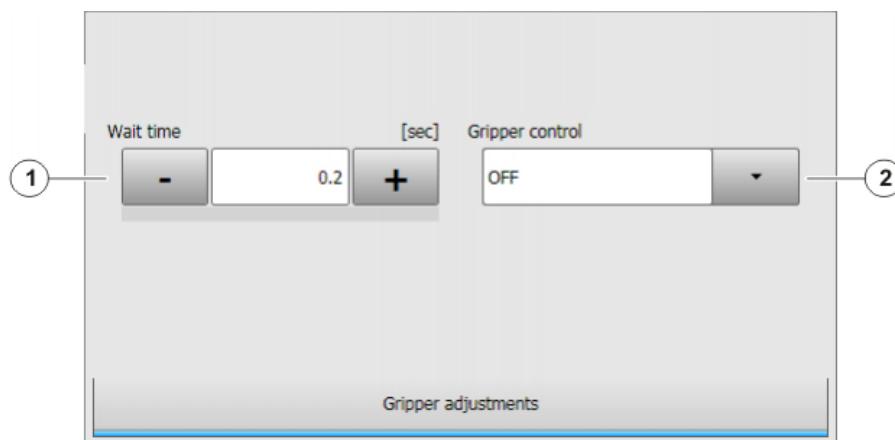


Fig. 8-2: Ajustes de garra

- Utilizar el control de las garras:
  - En caso de que el control de las garras se active con **ON**, se consultará la técnica de sensores parametrizada.
  - En caso de no obtener respuesta de los sensores, se produce un error de Timeout con la posibilidad de realizar una simulación del sensor en modo de prueba.
  - En caso de que no se active el control de las garras (**OFF**), se esperará durante el tiempo de espera parametrizado antes de continuar con el programa.

### Procedimiento para la programación de garra

#### Procedimiento

1. Seleccionar la secuencia de menú **Instrucciones > GripperTech > Gripper**.
2. Ajustar los parámetros en el formulario inline.
3. Guardar con **Instrucción OK**.

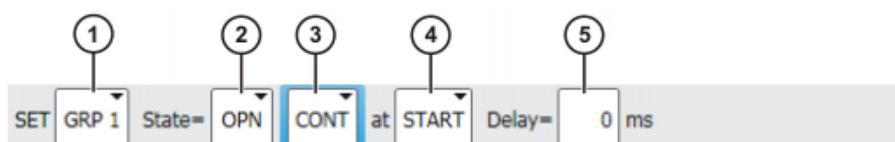


Fig. 8-3: Formulario inline Gripper con aproximación

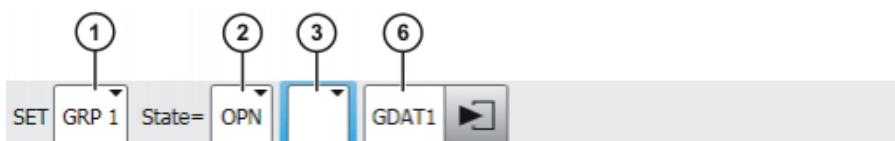


Fig. 8-4: Formulario inline Gripper sin posicionamiento aproximado

Pos.	Descripción
1	Seleccionar la garra. <ul style="list-style-type: none"> <li>■ Todas las garras configuradas se encuentran disponibles.</li> </ul>
2	Seleccionar el estado de comutación de la garra. <ul style="list-style-type: none"> <li>■ La cantidad depende del tipo de garra.</li> <li>■ La denominación depende de la configuración.</li> </ul>
3	Procesamiento en movimiento de avance. <ul style="list-style-type: none"> <li>■ <b>CONT</b>: Procesamiento durante el movimiento</li> <li>■ <b>[vacío]</b>: Ejecución con parada del procesamiento.</li> </ul>

Pos.	Descripción
4	El campo únicamente está disponible al seleccionar <b>CONT</b> . <ul style="list-style-type: none"> <li>■ <b>START</b>: La acción de garra se realiza en el punto de inicio del movimiento.</li> <li>■ <b>END</b>: La acción de garra se realiza en el punto de destino del movimiento.</li> </ul>
5	El campo únicamente está disponible al seleccionar <b>CONT</b> . <p>Determinar el tiempo de espera en el cual se ejecuta la función de garra respecto al punto de inicio o de destino del movimiento.</p> <ul style="list-style-type: none"> <li>■ -200 ... 200 ms</li> </ul>
6	Juego de datos con parámetros de garra

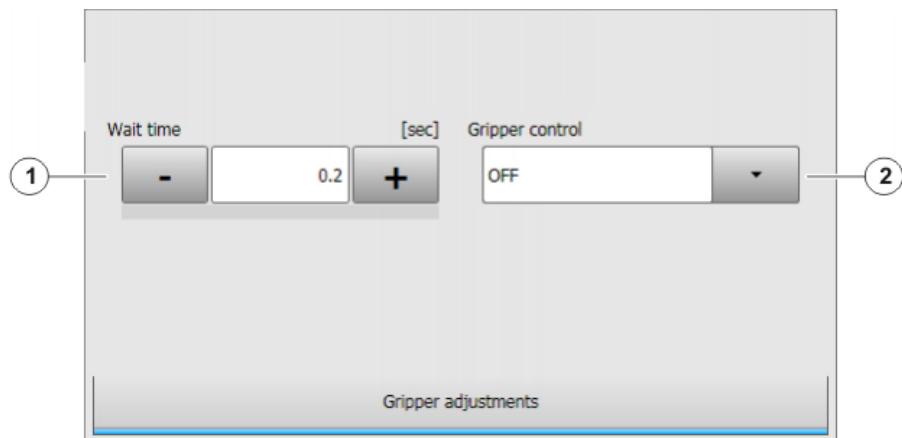


Fig. 8-5: Ajustes de garra

Pos.	Descripción
1	Tiempo de espera tras el que se va a continuar el movimiento programado <ul style="list-style-type: none"> <li>■ 0 ... 10 s</li> </ul>
2	Control de las garras <ul style="list-style-type: none"> <li>■ OFF (por defecto): Esperar el tiempo de espera ajustado arriba</li> <li>■ ON: Esperar los sensores</li> </ul>

#### Procedimiento para programar el control de garra

#### Procedimiento

1. Seleccionar la secuencia de menú **Instrucciones > GripperTech> Check Gripper**.
2. Ajustar los parámetros en el formulario inline.
3. Guardar con **Instrucción OK**.

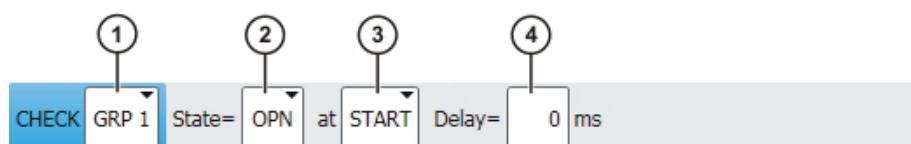


Fig. 8-6: Formulario inline Check Gripper

Pos.	Descripción
1	Seleccionar la garra. ■ Todas las garras configuradas se encuentran disponibles.
2	Seleccionar el estado de conmutación de la garra. ■ La cantidad depende del tipo de garra. ■ La denominación depende de la configuración.
3	Seleccionar el momento de la consulta del sensor. ■ <b>START</b> : La consulta de sensor se realiza en el punto de inicio del movimiento. ■ <b>END</b> : La consulta de sensor se realiza en el punto de destino del movimiento.
4	Determinar el tiempo de espera en el cual se ejecuta la consulta de sensor respecto al punto de inicio o de destino del movimiento.

#### 8.4 Configuración de KUKA.GripperTech

##### Possibilidades de configuración y tipos de garra

KUKA.GripperTech ofrece la posibilidad de configurar las garras de forma sencilla. Están disponibles cinco tipos predefinidos de garra. De manera adicional, pueden configurarse garras definidas por el usuario.

**AVISO**

Se pueden configurar hasta 16 tipos diferentes de garras en la unidad de control.

##### Tipos de garra

Tipo	OUT	IN	Estados	Ejemplo
Tipo 1	2	4	2	Garra simple con función DESCON. y CON.
Tipo 2	2	2	3	Carros con posición intermedia
Tipo 3	2	2	3	Garra de vacío con las funciones ASPIRACIÓN, SOPLADO, DESCON.
Tipo 4	3	2	3	Como el tipo 3, pero con tres salidas de control
Tipo 5	2	4	2	Como el tipo 1, pero con impulso en lugar de señal continua
Libre	Configurable			

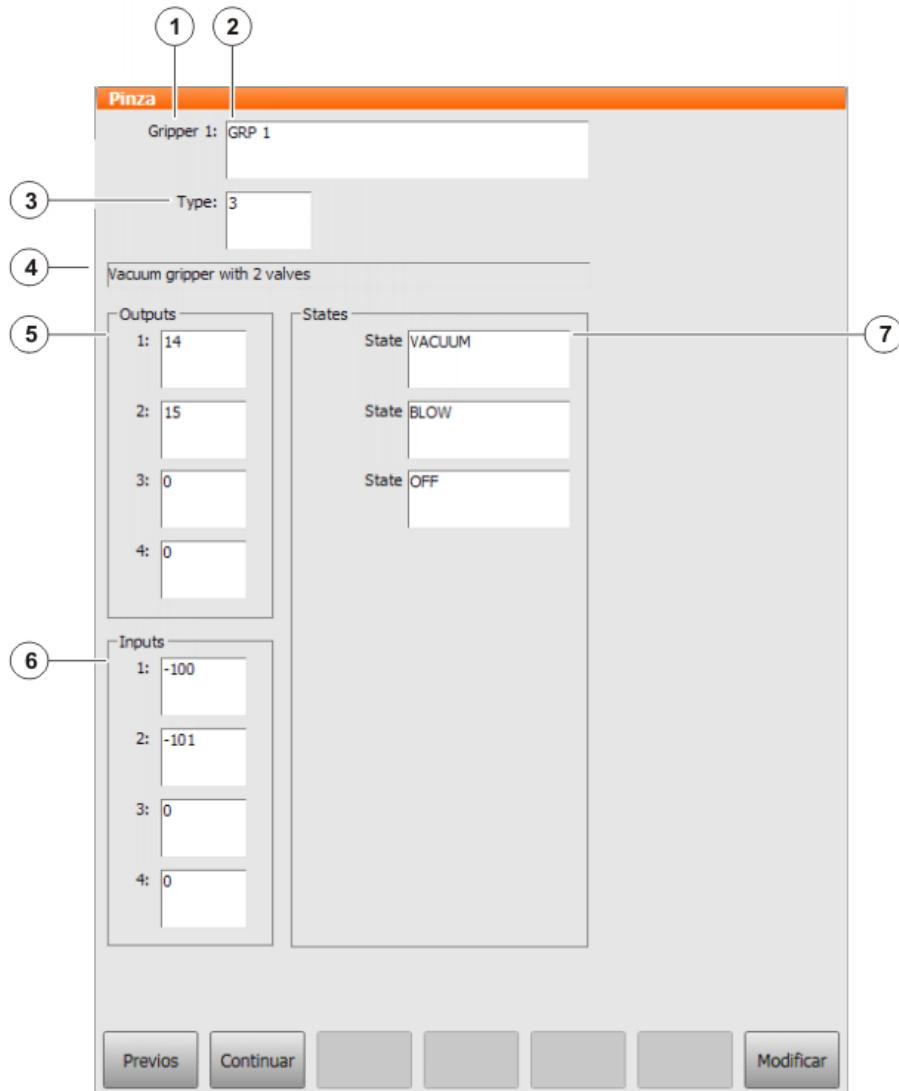


Fig. 8-7: Ejemplo: garra predefinida

Pos.	Descripción
1	Número de la garra ■ 1 ... 16
2	Nombre de la garra El nombre se muestra en el formulario inline. El nombre asignado por defecto se puede modificar: ■ 1 ... 24 caracteres
3	Tipo ■ Para garras predefinidas: 1 ... 5 (ver la tabla de los tipos de garras)
4	Denominación del tipo de garra (se actualiza después de guardar) La denominación no se puede modificar.
5	Asignación de los números de salida Para las salidas que no sean necesarias se puede asignar "0". De este modo se ve inmediatamente que no se están utilizando. Si, no obstante, se les asigna un número, este no tendrá efecto alguno.

Pos.	Descripción
6	Asignación de los números de entrada A las entradas que no sean necesarias se les puede asignar "0". De este modo se ve inmediatamente que no se están utilizando. Si, no obstante, se les asigna un número, este no tendrá efecto alguno.
7	Estados de conmutación Los nombres asignados por defecto se pueden modificar. Los nombres se muestran en el formulario inline cuando se selecciona en el mismo la garra correspondiente.

**Tipos libres de garra**

Para poder satisfacer todas las necesidades del usuario, se ha integrado un tipo de garra programable. Mediante entradas en los archivos \$CONFIG.DAT, USERGRP.DAT y USER\_GRP.SRC pueden configurarse todas las garras definibles que se deseen.

**AVISO**

Para más información sobre la configuración de garras, consultar las instrucciones de servicio KUKA.Gripper&SpotTech de KUKA System Technology.

**Procedimiento para la configuración de garras**

Configuración con el tipo de garra predefinido

1. Seleccionar en el menú principal **Configuración > Entradas/Salidas > Garra**. Se abre una ventana.
2. Seleccionar el número de garra deseado con **Continuar** o **Previo**.
3. Si se desea, modificar el nombre por defecto de la garra.
4. Asignar a la garra un tipo entre 1 y 5.
5. Asignar las entradas y las salidas.
6. Si se desea, modificar los nombres de los estados otorgados por defecto.
7. Guardar la configuración con **Modificar**.

## 8.5 Ejercicio: Programación de la garra, placa

### Objetivo del ejercicio

Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Programación de instrucciones para mando y control de garras y pinzas (KUKA.Gripper & SpotTech)
- Activar y trabajar con teclas de estado específicas de la tecnología

### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos del paquete de tecnología KUKA.Gripper & SpotTech

### Formulación de tarea

Ejecutar las siguientes tareas: Tomar y descargar placa

1. Cree un nuevo programa con la denominación **Tomar\_placa**, utilice la herramienta garra y la base azul.
2. Programar por aprendizaje el proceso *Tomar placa*, de modo que se obtenga la posición de descarga y de toma representada en la figura (*>>> Fig. 8-8*). Para ello, reducir la velocidad a 0,3 m/s durante la retirada y colocación de la placa.
3. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.
4. Crear un segundo programa con la denominación **Descargar\_placa**. Para ello, utilizar una base adecuada y la herramienta correspondiente.
5. Programe por aprendizaje el proceso Descargar placa.
6. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.
7. Archive sus programas.



Fig. 8-8: Placa con posición de descarga

1 Placa

2 Posición de descarga

Lo que se debe saber tras el ejercicio:

1. ¿De qué formularios inline se dispone con el paquete de tecnología **KUKA.GripperTech**? Describir su función.

.....  
.....  
.....  
.....

2. ¿Qué efecto tiene el tiempo de espera en los *ajustes de la garra*?

.....  
.....  
.....

## 8.6 Ejercicio: Programación de la garra, clavija

### Objetivo del ejercicio

Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Programación de instrucciones para mando y control de garras y pinzas (KUKA.Gripper & SpotTech)
- Activar y trabajar con teclas de estado específicas de la tecnología

### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos del paquete de tecnología KUKA.Gripper & SpotTech

### Formulación de tarea

Ejecutar las siguientes tareas: Tomar y descargar clavija 1

1. Cree dos nuevos programas con la denominación **Tomar\_clavija 1** y **Descargar\_clavija1**
2. Al programar, disfrute las ventajas de la dirección de impacto de la herramienta en el desplazamiento manual.
3. Tenga en cuenta que al tomar y descargar en el depósito de clavijas, la velocidad de desplazamiento no sea mayor que 0,3 m/s.
4. Antes de tomar la clavija, efectúe una pregunta de seguridad respecto a la posición de la garra.

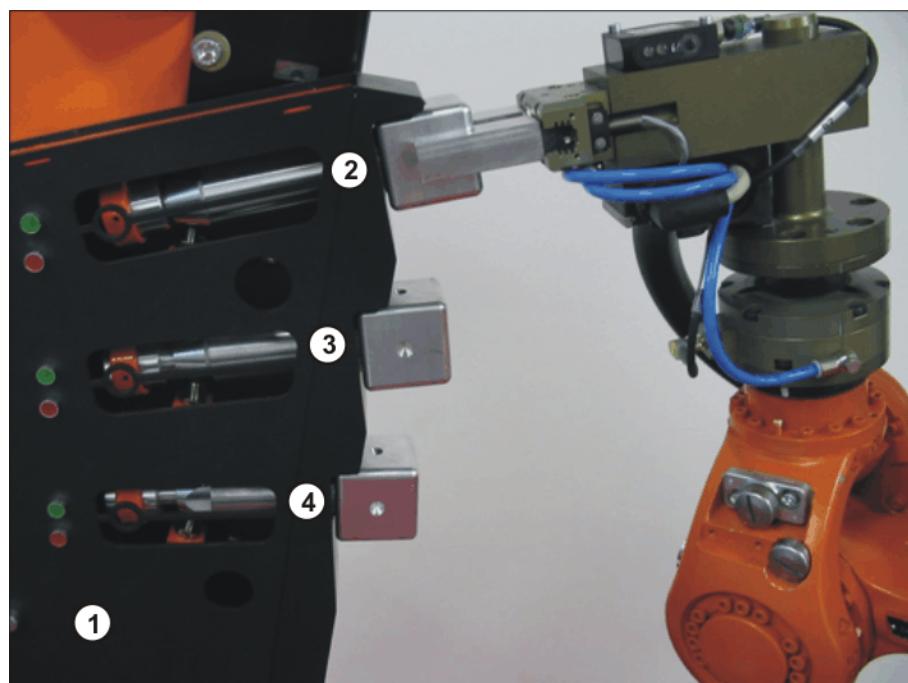


Fig. 8-9: Depósito de clavijas

- |                        |             |
|------------------------|-------------|
| 1 Depósito de clavijas | 2 Clavija 1 |
| 3 Clavija 2            | 4 Clavija 3 |

### Lo que se debe saber tras el ejercicio:

1. ¿Cuál es la diferencia entre un tiempo de espera y el control de garra ON/OFF?

.....

.....

2. Aparece el aviso *Posicionamiento aproximado no posible* como texto de mensaje. Cuáles son las causas posibles para ello?

.....  
.....  
.....  
.....  
3. ¿Cuántos tipos de garra estándar KUKA existen?  
.....  
.....



## 9 Configuración y programación de herramientas externas

### 9.1 Resumen

Esta unidad incluye los siguientes contenidos:

- Desplazamiento manual con una herramienta fija
- Puesta en servicio del robot
  - Medición de una herramienta fija
  - Medición de una pieza guiada por robot
- Programación de movimientos con un TCP externo

### 9.2 Mover el robot

#### 9.2.1 Procesos manuales con una herramienta fija

##### Ventajas y campos de aplicación

Algunos procesos de producción y mecanizado requieren que el robot maneje la **pieza** y no la **herramienta**. La ventaja es que el componente no se debe depositar sólo para el mecanizado con lo que se pueden ahorrar dispositivos de seguridad. Este es el caso, por ejemplo, en las situaciones siguientes:

- Aplicaciones de pegamentos
- Aplicaciones soldadas
- etc.

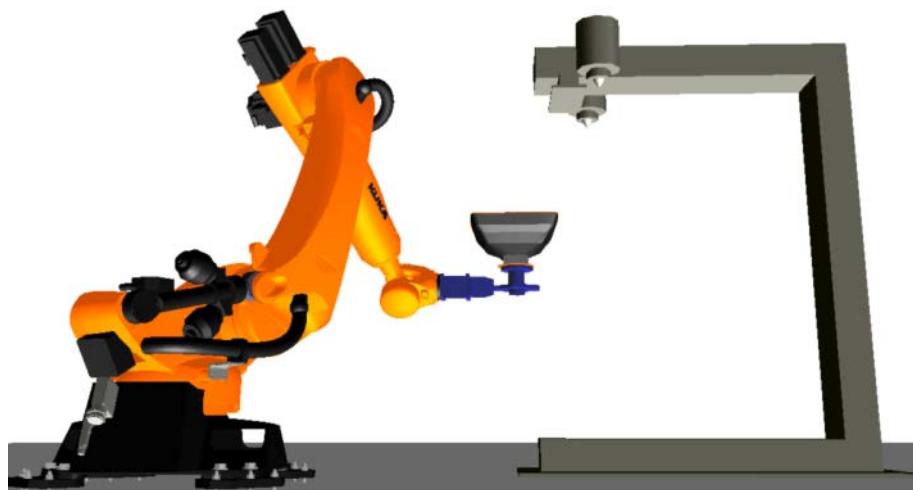


Fig. 9-1: Ejemplo de herramienta fija

#### AVISO

Para poder programar correctamente una aplicación de este tipo se debe medir tanto el TCP externo de la herramienta fija como la pieza.

##### Ejecución modificada del ciclo de movimiento con herramienta fija

A pesar de que la herramienta es un objeto fijo (no se mueve), también tiene un punto de referencia con el sistema de coordenadas. Este punto de referencia recibe el nombre de **TCP externo**. Como se trata de un sistema de coordenadas no móvil, los datos se administran como en un sistema de coordenadas base y, por tanto, se guardan como **base**.

La pieza (móvil) **Pieza** se vuelve a guardar como **Herramienta**. Por consiguiente, se puede realizar un desplazamiento a lo largo de los cantos de la pieza con relación al TCP.

**AVISO**

Debe tenerse en cuenta que en los procesos manuales con herramienta fija, los movimientos con relación al TCP externo.

**Procedimiento de los procesos manuales con herramienta fija**

Fig. 9-2: Selección ext. TCP en el menú de opciones

1. Seleccionar la pieza guiada por el robot en la ventana Seleccionar herramienta.
2. Seleccionar la herramienta fija en la ventana Selección de base.
3. Ajustar la selección del modo IpolpoMode a **herramienta externa**.
4. Como alternativa a las teclas de desplazamiento/space-mouse, ajustar:
  - Ajustar herramienta para desplazar la pieza en el sistema de coordenadas.
  - Ajustar la base para desplazar la herramienta ext. en el sistema de coordenadas.
5. Ajustar el override manual
6. Pulsar y mantener pulsado el pulsador de validación en la posición intermedia.
7. Con las teclas de desplazamiento o el Space Mouse, realizar el desplazamiento en la dirección deseada.

Al seleccionar **Herramienta ext.** en la ventana opcional **Opciones de desplazamiento manual**, la unidad de control cambia: ahora todos los movimientos se realizan con relación al TCP externo y no a la herramienta guiada por el robot.

## 9.2.2 Ejercicio: Procesos manuales con herramienta fija

<b>Objetivo del ejercicio</b>	Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:
	<ul style="list-style-type: none"><li>■ Desplazamiento manual de una pieza guiada por el robot respecto a una herramienta fija</li></ul>
<b>Requisitos</b>	Los siguientes requisitos son necesarios para completar este ejercicio correctamente: <ul style="list-style-type: none"><li>■ Conocimientos teóricos acerca del servicio general de un sistema de robot industrial KUKA</li><li>■ Conocimientos teóricos sobre el desplazamiento con la herramienta externa</li></ul>
<b>Formulación de tarea</b>	<ol style="list-style-type: none"><li>1. Ajustar el sistema de coordenadas de la herramienta <i>D_Panel</i>.</li><li>2. Ajustar el sistema de coordenadas de base <i>D_Glue Nozzle</i>.</li><li>3. Ajustar, en el menú de opciones de procesos manuales, la opción <i>Herram. ext.</i> (<a href="#">&gt;&gt;&gt; Fig. 9-2</a> )</li><li>4. Mover la <i>placa hacia la clavija externa</i>.</li><li>5. Mover y orientar la <i>placa en la clavija externa</i>. Probar las diferencias entre desplazamiento en el sistema de coordinadas de la herramienta y el de la base.</li><li>6. Ajustar, en el menú de opciones de procesos manuales, la opción <i>Brida</i>.</li><li>7. Mover y orientar la <i>placa en la clavija externa</i>. (<a href="#">&gt;&gt;&gt; Fig. 9-2</a> )</li></ol>

## 9.3 Puesta en servicio del robot

### 9.3.1 Medición de una herramienta fija

#### Resumen

La medición de la herramienta fija consta de dos pasos:

1. Determinación de la posición del TCP externo en relación al origen del sistema de coordenadas universal.
2. Orientación del sistema de coordenadas en el TCP externo.

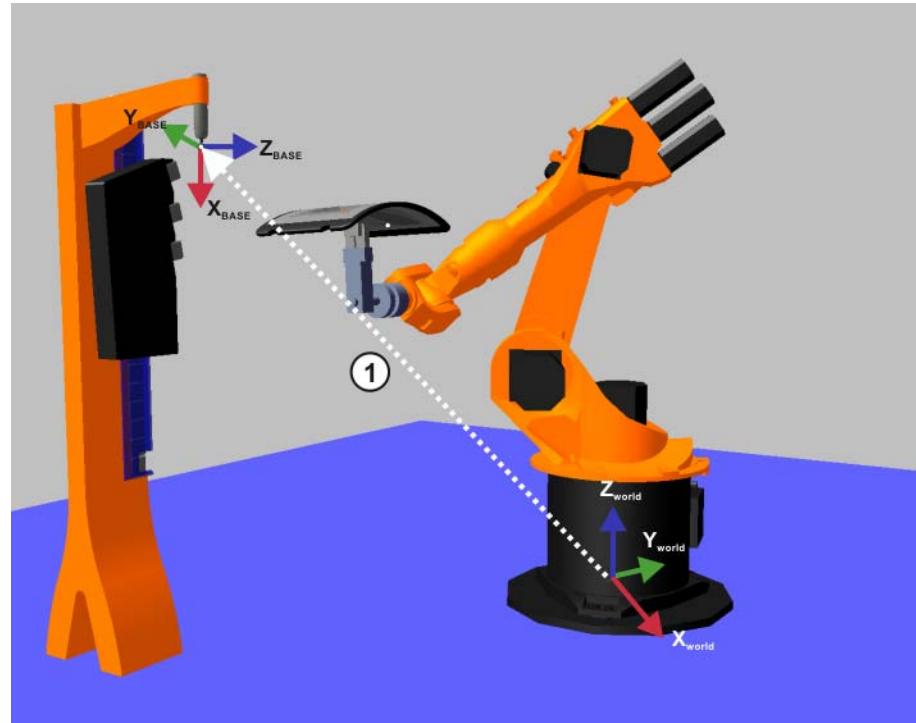


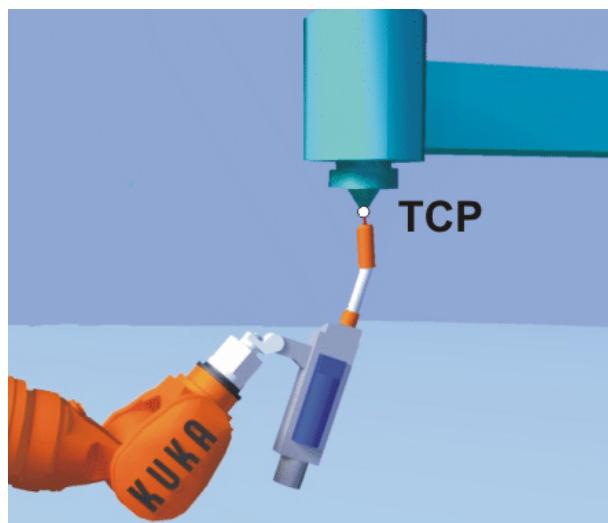
Fig. 9-3: Medición de la herramienta fija

Como muestra la figura (1) ([>>> Fig. 9-3](#)), el TCP externo se administra referido a \$WORLD (o \$ROBROOT), es decir, como un sistema de coordenadas de base.

#### Descripción de la medición

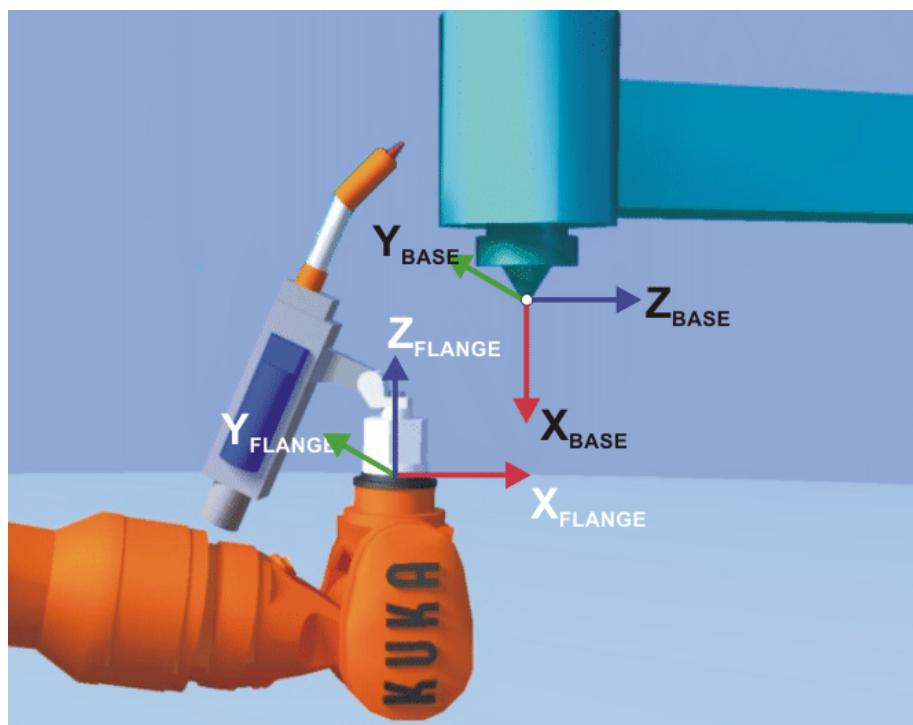
##### 1. Determinación del TCP externo

Para la medición del TCP externo se necesita una herramienta guiada por robot que ya haya sido medida. Con la punta de la misma se realiza la aproximación al TCP externo.



**Fig. 9-4: Desplazamiento al TCP externo**

## 2. Determinación de la orientación



**Fig. 9-5: Alinear de forma paralela los sistemas de coordenadas**

Para determinar la orientación el sistema de coordenadas de brida se alinea paralelamente al nuevo sistema de coordenadas.

Existen 2 variantes:

- **5D:** A la unidad de control sólo se le indica la dirección de trabajo de la herramienta fija. Por defecto, la dirección de avance es el eje X. La orientación de los demás ejes la determina el sistema, y en general no puede ser reconocida fácilmente por el usuario.
- **6D:** A la unidad de control se le indican las orientaciones de todos los 3 ejes.

### Procedimiento

1. En el menú principal, seleccionar **Puesta en servicio > Medir > Herramienta fija > Herramienta**.
2. Introducir un número y un nombre para la herramienta fija. Confirmar pulsando **Continuar**.

3. Introducir el número de la herramienta de referencia utilizada. Confirmar pulsando **Continuar**.
4. En el campo **5D/6D** seleccionar una variante. Confirmar pulsando **Continuar**.
5. Con el TCP de la herramienta ya medida, desplazarse al TCP de la herramienta fija. Pulsar **Medición**. Confirmar la posición con **Sí**.
6. Si se ha seleccionado **5D**:  
Alinear  $+X_{BASE}$  de forma paralela a  $-Z_{FLANGE}$ .  
(es decir, alinear la brida de acople perpendicular a la dirección de avance de la herramienta fija).  
Si se ha seleccionado **6D**:  
Alinear la brida de acople de modo tal que sus ejes se encuentren paralelos a los ejes de la herramienta fija:
  - Alinear  $+X_{BASE}$  de forma paralela a  $-Z_{FLANGE}$   
(es decir, alinear la brida de acople perpendicular a la dirección de avance de la herramienta).
  - $+Y_{BASE}$  de forma paralela a  $+Y_{FLANGE}$
  - $+Z_{BASE}$  de forma paralela a  $+X_{FLANGE}$
7. Pulsar **Medición**. Confirmar la posición con **Sí**.
8. Pulsar **Guardar**.

### 9.3.2 Medición de una pieza guiada por robot

**Sinopsis:**

**Medición directa**

#### AVISO

A continuación sólo se describe el método de medición directo. La medición indirecta es muy poco frecuente y se describe detalladamente en la documentación *Instrucciones de uso y programación del KUKA System Software 8.2*.

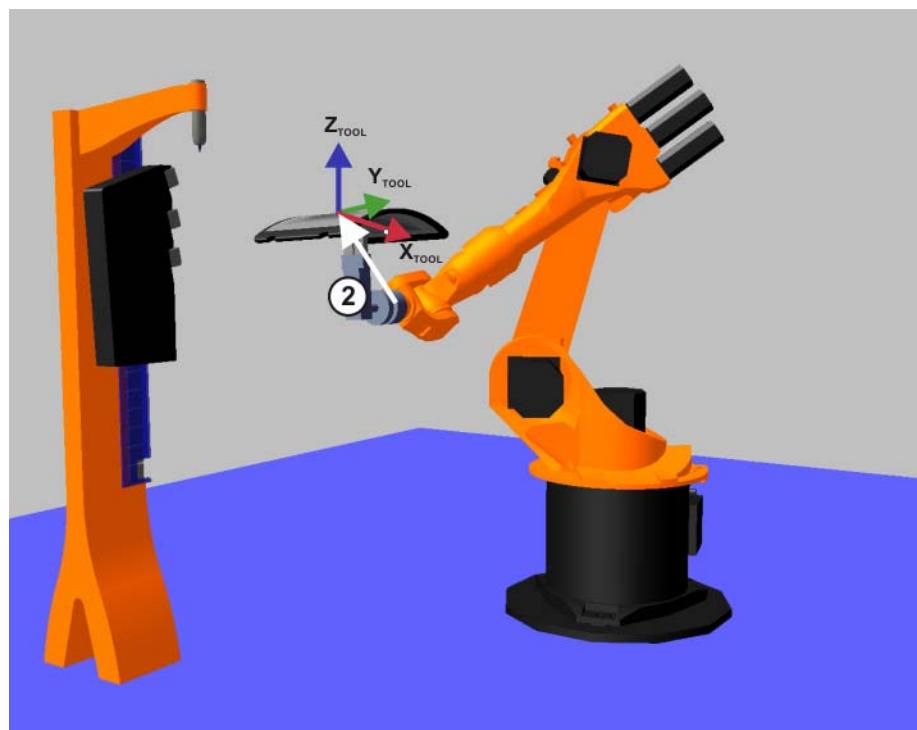
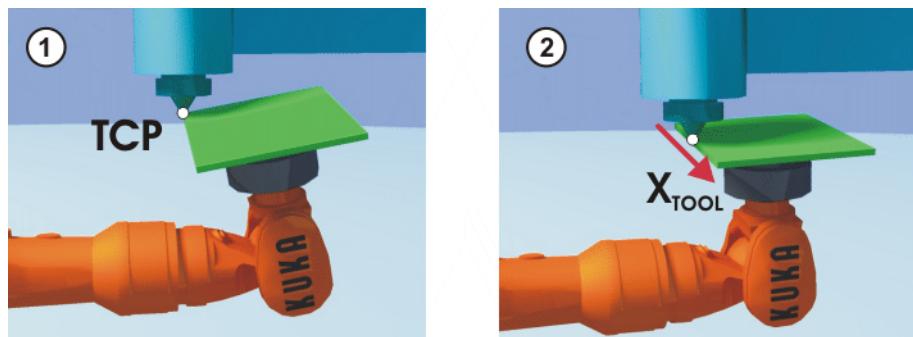


Fig. 9-6: Medición de la pieza por medición directa

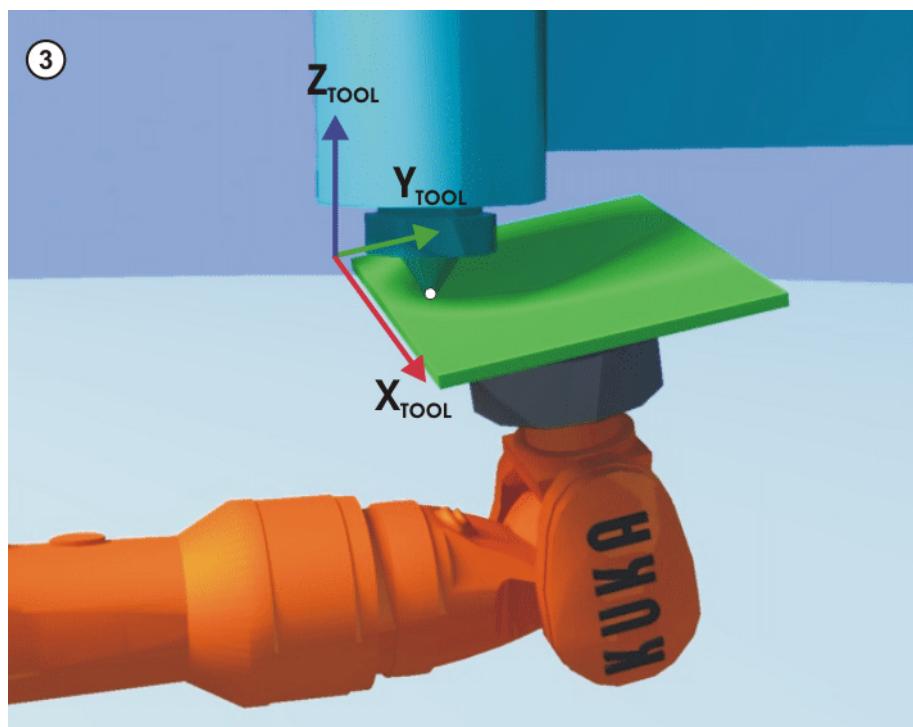
Pieza	Medición
2	Medición la pieza

**Descripción**

A la unidad de control del robot se le comunican el origen y 2 puntos más de la pieza. Estos 3 puntos definen la pieza de forma única.



**Fig. 9-7**



**Fig. 9-8: Medir la pieza Método directo**

**Procedimiento**

1. En el menú principal, seleccionar la secuencia **Puesta en servicio > Medir > Herramienta fija > Pieza de trabajo > Medición directa**.
2. Indicar un número y un nombre para la pieza. Confirmar pulsando **Continuar**.
3. Introducir el número de herramienta fija. Confirmar pulsando **Continuar**.
4. Desplazar el origen del sistema de coordenadas de la pieza al TCP de la herramienta fija.  
Pulsar **Medir** y confirmar la posición pulsando **Sí**.
5. Desplazar a un punto sobre el eje X positivo del sistema de coordenadas de la pieza en el TCP de la herramienta fija.  
Pulsar **Medir** y confirmar la posición pulsando **Sí**.
6. Desplazar a un punto que, en el plano XY del sistema de coordenadas de la pieza tiene un valor Y positivo, en el TCP de la herramienta fija.  
Pulsar **Medir** y confirmar la posición pulsando **Sí**.
7. Introducir los datos de carga de la pieza y confirmar con **Continuar**.
8. Pulsar **Guardar**.

### 9.3.3 Ejercicio: Medir herramienta externa y pieza guiada por robot

#### Objetivo del ejercicio

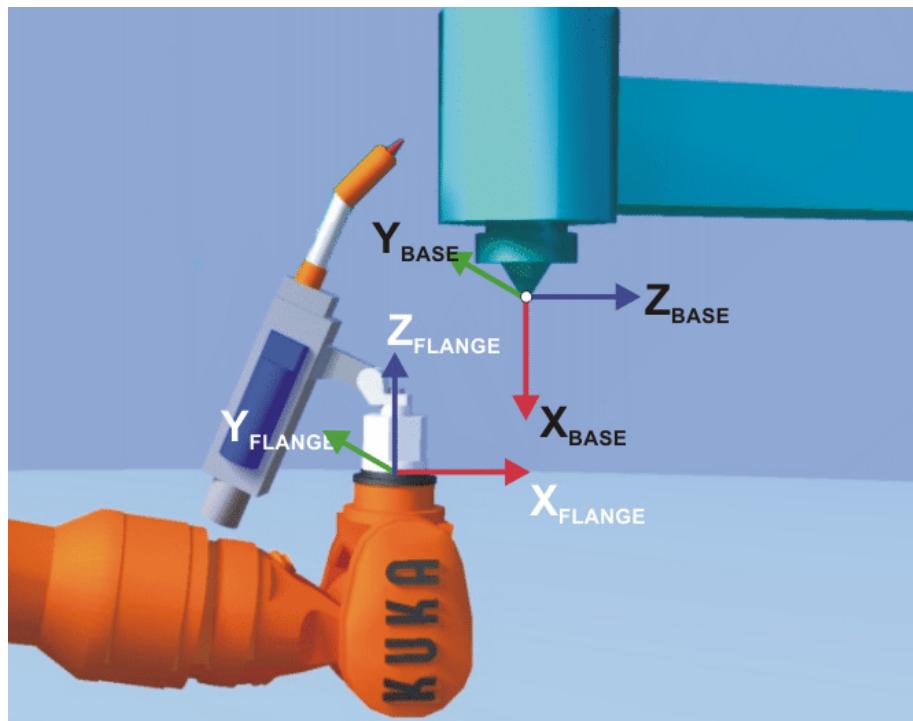
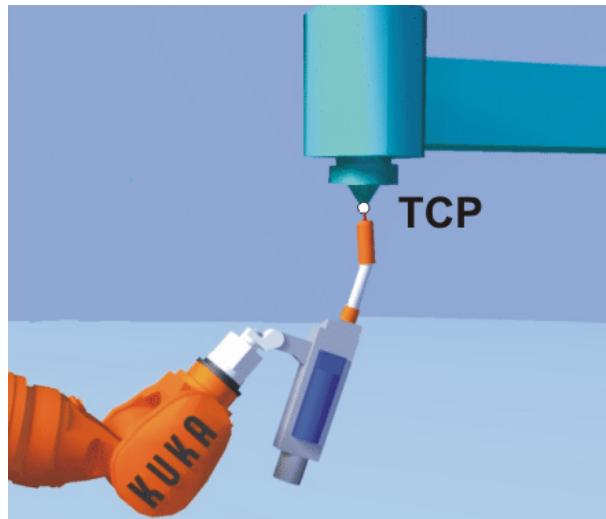
Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Medir herramientas fijas
- Medir piezas móviles guiadas por robot
- Desplazamiento manual con una herramienta externa

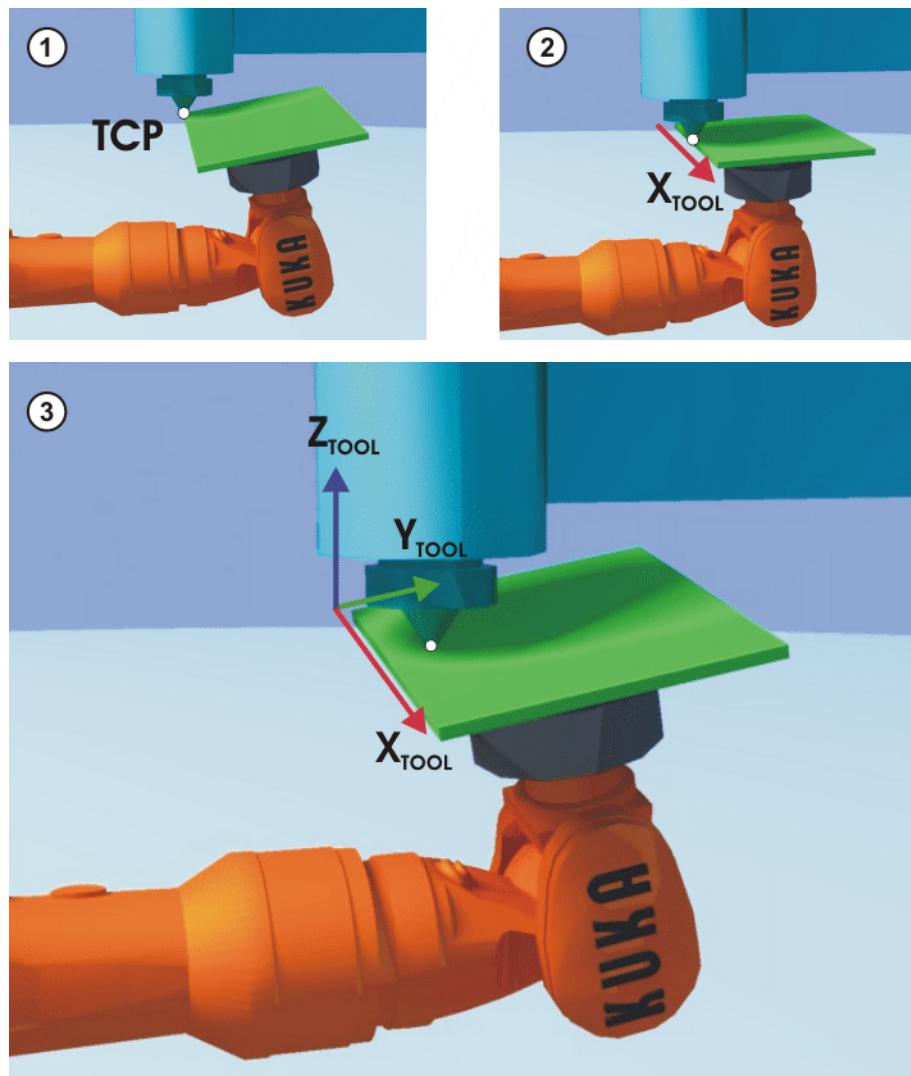
#### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos de los métodos de medición de herramientas fijas



- Conocimientos teóricos de la medición de pieza con herramientas fijas, especialmente el método directo



#### Formulación de tarea

Ejecutar las siguientes tareas: Medición boquilla y placa

- Para medir la herramienta fija debe utilizarse la clavija1 ya medida (número de herramienta 2) como herramienta de referencia. Asignar para la herramienta fija el **número de herramienta 10** y la denominación **Boquilla**.
  - Tener en cuenta que en cada medición se deben guardar los datos.
- Efectuar la medición de la pieza guiada por el robot. Asignar el **número de herramienta 12** y el nombre **Placa**.

Introducir los datos de carga.

Usar para ello la garra con placa correcta (ver más abajo).

- Al terminar la medición, activar la herramienta externa para el desplazamiento manual. Utilizar el sistema de coordenadas de base y de herramienta de forma sensata y desplazar el robot.
- Desplazar el TCP al origen de coordenadas base de la pieza medida y visualizar la posición cartesiana real.

Posición actual:

X	Y	Z	A	B	C
.....	.....	.....	.....	.....	.....

.....

**Datos de carga de la herramienta  
garra con placa**

- Garra de formación KR 16

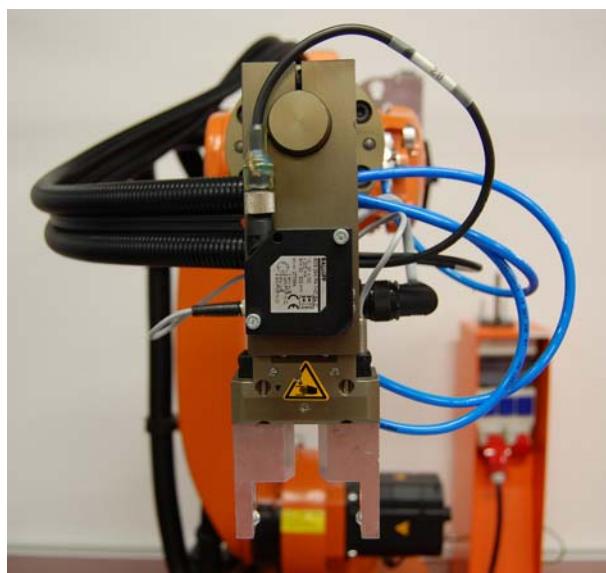


Fig. 9-9: Garra de formación KR 16

<b>Masa:</b>		
M = 6 kg		
<b>Centro de masa:</b>		
X = 69 mm	Y = 67 mm	Z = 84 mm
<b>Orientación:</b>		
A = 0°	B = 0°	C = 0°
<b>Momentos de inercia:</b>		
J <sub>X</sub> = 0,02 kgm <sup>2</sup>	J <sub>Y</sub> = 0,06 kgm <sup>2</sup>	J <sub>Z</sub> = 0,18 kgm <sup>2</sup>

- Garra de formación celda modular

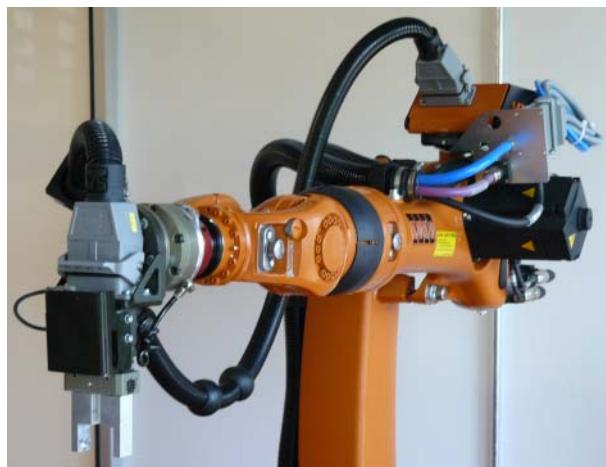


Fig. 9-10: Garra de formación celda modular

<b>Masa:</b>		
M = 6,5 kg		
<b>Centro de masa:</b>		
X = 42 mm	Y = -54 mm	Z = 93 mm
<b>Orientación:</b>		
A = 0°	B = 0°	C = 0°

<b>Momentos de inercia:</b>		
$J_X = 0,02 \text{ kgm}^2$	$J_Y = 0,11 \text{ kgm}^2$	$J_Z = 0,18 \text{ kgm}^2$

- Garra de formación celda móvil



**Fig. 9-11: Garra de formación celda móvil**

<b>Masa:</b>		
$M = 2,18 \text{ kg}$		
<b>Centro de masa:</b>		
$X = 34 \text{ mm}$	$Y = 0 \text{ mm}$	$Z = 68 \text{ mm}$
<b>Orientación:</b>		
$A = 0^\circ$	$B = 0^\circ$	$C = 0^\circ$
<b>Momentos de inercia:</b>		
$J_X = 0,002 \text{ kgm}^2$	$J_Y = 0,005 \text{ kgm}^2$	$J_Z = 0,004 \text{ kgm}^2$

#### **Lo que se debe saber tras el ejercicio:**

1. ¿Cómo se mide un sistema de coordenadas en una pieza montada en una brida de robot?

.....  
.....  
.....

2. Cómo se calcula el TCP de una herramienta externa?

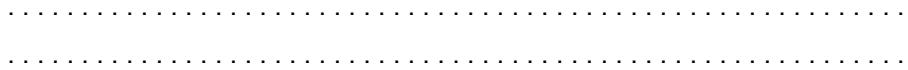
.....  
.....  
.....

3. ¿Qué ventajas tiene usar un TCP externo?

.....  
.....  
.....

4. Cuáles son los ajustes necesarios para desplazar con un TCP externo en dirección de avance de la herramienta?

.....  
.....



## 9.4 Crear y modificar movimiento programado

### 9.4.1 Programación de movimiento con TCP externo

#### Programar movimientos con el TCP externo

Al programar movimientos con una herramienta fija, comparado con el movimiento estándar, en este caso el movimiento presenta las diferencias siguientes:

- Identificación en el formulario inline: En la ventana de opciones **Vectores**, la entrada **TCP externo** debe estar en TRUE.

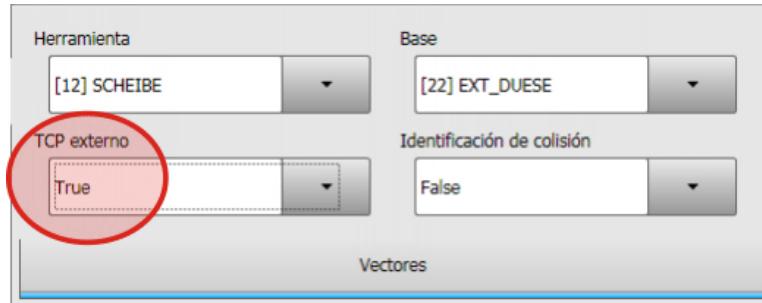


Fig. 9-12: Ventana de opciones "Vectores": TCP ext.

- La **velocidad de movimiento** está referenciada al TCP externo.
- La **orientación** a lo largo de la trayectoria está referenciada también a la herramienta externa.
- Se debe indicar tanto el sistema de coordenadas de base (herramienta fija / TCP externo) como el sistema de coordenadas de herramienta (pieza móvil).

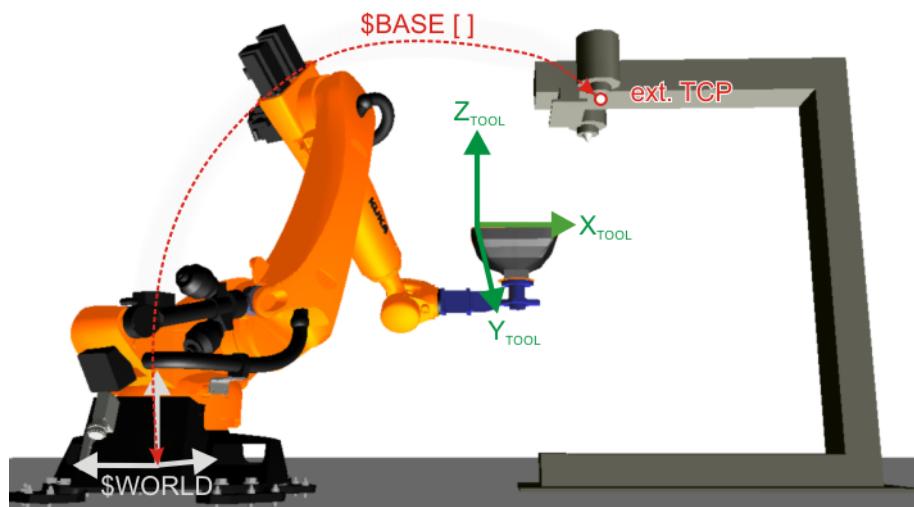


Fig. 9-13: Sistemas de coordenadas con herramienta fija

#### 9.4.2 Ejercicio: Programación de movimientos con TCP externo

**Objetivo del ejercicio** Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Programación de movimientos de una pieza desplazada por el robot respecto a una herramienta fija

**Requisitos** Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos de la activación de la herramienta externa al programar movimientos.

**Formulación de tarea** Ejecutar las siguientes tareas: Programación del contorno para la aplicación de adhesivo:

1. Sujetar manualmente la placa en la garra.
2. Programe por aprendizaje el contorno indicado sobre la placa con la denominación de programa **Pegar\_placa**.
  - Utilizar para ello la herramienta externa medida **Boquilla** y la pieza **Placa**.
  - Tener en cuenta que el eje longitudinal de la herramienta fija siempre debe encontrarse en una posición vertical respecto al contorno de pegado.
  - La velocidad de desplazamiento sobre la placa es de 0,2 m/s
3. Compruebe su programa según prescripción.
4. Archivar el programa.

**Lo que se debe saber tras el ejercicio:**

1. ¿Cuál es el punto de referencia de la velocidad de pegado programada?
- .....
- .....

2. Cómo se activa la herramienta externa en su programa?
- .....
- .....

## 10 Introducción al nivel del experto

### 10.1 Resumen

**Esta unidad incluye los siguientes contenidos:**

- Utilizar el nivel experto
- Estructurar programas de robot
- Concatenación de programas del robot

### 10.2 Utilizar el nivel experto

<b>Descripción</b>	<p>La unidad de control del robot ofrece diferentes <b>grupos de usuarios</b> con diferentes funciones. Se pueden seleccionar los siguientes <b>grupos de usuarios</b>:</p> <ul style="list-style-type: none"> <li>■ <b>Operario</b> Grupo para el operario. Esto es el grupo de usuario por defecto.</li> <li>■ <b>Usuario</b> Grupo para el operario. (Los grupos de usuarios del operario y del usuario están declarados por defecto para el mismo grupo destinatario.)</li> <li>■ <b>Experto</b> Grupo de usuario para el programador. Este grupo de usuario está protegido por una contraseña.</li> <li>■ <b>Administrador</b> Mismas funciones a las del grupo de usuarios "Experto". A modo adicional es posible integrar plug-ins en la unidad de control del robot. Este grupo de usuario está protegido por una contraseña. La contraseña por defecto KUKA se debería cambiar.</li> <li>■ <b>Técnico de mantenimiento de seguridad</b> Este usuario puede activar una configuración de seguridad existente del robot mediante un código de activación. Si no se utiliza ninguna opción Safe, p. ej. KUKA.SafeOperation o KUKA.SafeRangeMonitoring, el técnico de mantenimiento de seguridad dispone de derechos complementarios. En este caso, por ejemplo, está autorizado para configurar las funciones de seguridad estándar. El grupo de usuario está protegido por una contraseña. La contraseña por defecto KUKA se debería cambiar.</li> <li>■ <b>Personal responsable de la puesta en servicio con seguridad</b> Este grupo de usuario solo es relevante cuando se utiliza KUKA.SafeOperation o KUKA.SafeRangeMonitoring. El grupo de usuario está protegido por una contraseña. La contraseña por defecto KUKA se debería cambiar.</li> </ul>
--------------------	--

#### Funciones avanzadas del grupo de usuario Experto:

- Protegido por contraseña (predeterminada: kuka)
- La programación en el editor mediante KRL es posible
- El grupo de usuario Experto se vuelve a abandonar automáticamente
  - cuando se cambia al modo de servicio AUT o AUT EXT
  - cuando durante un tiempo determinado no se realiza ninguna operación en la interfaz de usuario (300 s)



Como experto, en el menú **Procesar** se encuentran disponibles todas las funciones.

**Funciones****Crear programas mediante plantillas**

- **Cell:** El programa Cell disponible solo se puede sustituir o crearse de nuevo en caso de que se borre Cell.
- **Expert:** Módulo compuesto por ficheros SRC y DAT en los que solo está disponible el encabezamiento y el final del programa.
- **Expert Submit:** Archivo Submit (SUB) adicional compuesto de encabezamiento y final del programa.
- **Function:** Creación de función de fichero SRC, en el que solo se crea el encabezamiento de la función con una variable BOOL. El final de la función está disponible, pero el retorno debe ser programado previamente.
- **Modul:** Módulo compuesto de ficheros SRC y DAT, en los que está disponible el encabezamiento y el final del programa y la estructura base (INI y 2x PTP HOME).
- **Submit:** Fichero Submit (SUB) adicional compuesto de encabezamiento y final del programa y estructura base (DECLARATION,INI,LOOP/ENDLOOP).

El **filtro** determina cómo se deben mostrar los programas en la lista de ficheros. Se puede elegir entre los siguientes **filtros**:

- Botón **Editar > Filtro** y, después, seleccionar **Detalle** o **Módulo**
- **Detalle**  
Los programas se visualizan como ficheros SRC y DAT (ajuste por defecto).
- **Módulos**  
Los programas se muestran como módulos.



**Fig. 10-1: Menú "Editar"**

#### Abrir/cerrar FOLD

- Los **FOLD** siempre están cerrados para el usuario y se pueden abrir como experto.
- El experto también puede programar **FOLDs** propios.

- La **sintaxis** para un FOLD es la siguiente:

```
;FOLD Nombre
Instrucciones
;ENDFOLD <Nombre>
```

Las filas ENDFOLD pueden asignarse más fácilmente si se introduce aquí el nombre del FOLD. Pueden encajarse los FOLDS.

#### Mostrar/ocultar la **Línea DEF**

- Por defecto, la **Línea DEF** está oculta. Solo se pueden efectuar declaraciones en un programa una vez que la **Línea DEF** sea visible.
- La **Línea DEF** se muestra y oculta independientemente para cada programa abierto y seleccionado. Si está activada una vista en detalle, la **Línea DEF** está visible y no es necesario mostrarla expresamente.

### Procedimiento para activar el nivel experto y subsanar errores

#### Activar el nivel experto

1. Seleccionar en el menú principal la secuencia **Configuración > Grupo de usuario**.
2. Registrarse como **experto**: Pulsar **Iniciar sesión**. Marcar el grupo de usuario **Experto** y confirmar con **Iniciar sesión**.
3. Introducir la contraseña (predeterminada: kuka) y confirmar con **Iniciar sesión**.

#### Subsanar el error en el programa

1. Seleccionar un módulo defectuoso en el navegador



Fig. 10-2: Programa incorrecto

2. Seleccionar el menú **Lista de errores**
3. Se abre la indicación de errores (*nombre del programa.ERR*)
4. Seleccionar el error, se representa una descripción detallada abajo en la indicación de errores
5. En la ventana de indicación de errores, pulsar la tecla **Visualizar** y saltar al programa defectuoso
6. Subsanar el error
7. Abandonar el editor y guardar

### 10.3 Estructurar programas de robot

#### Posibilidades de estructuración de un programa de robot

La estructura de un programa de robot es un factor relevante para su valor de utilidad. Cuanto más estructurado esté un programa, más comprensible, efectivo, legible y rentable será. Para estructurar un programa pueden aplicarse las siguientes técnicas:

- **Comentar** | Comentarios y marcas
- **Acoplar** | Espacios en blanco
- **Ocultar** | Folds ("Plegar")

- **Tecnología de módulos | Subprogramas**

### Comentarios y marcas

La opción para añadir un comentario permite colocar un texto en el programa de robot dirigido a los lectores del programa. Por tanto, el intérprete de robot va a leer este texto. Se incluye únicamente para mejorar la legibilidad de un programa.

Los comentarios pueden introducirse en el programa de robot con diferentes fines:

- **Informaciones** sobre el texto de programa: autor, versión, fecha de creación

#### Editor

```

1 DEF welding1( )
2 ; Programmed by JACK SPARROW
3 ; Version 1.5 (10/10/2010)
4INI
5
6 PTP HOME  Vel= 100 % DEFAULT
7

```

**Fig. 10-3: Ejemplo de comentario: informaciones**

- **Estructura** del texto del programa: especialmente mediante el uso de elementos gráficos (caracteres especiales como #, \*, ~, )

```

0
7 ;*****Initialisation*****
8 INIT
9 BASISTECHINI
10 CHECK HOME

```

**Fig. 10-4: Ejemplo de comentario: Esquematización**

- **Definir comentario** (nivel experto): al colocar un punto y coma al principio de una línea de programa, esta se define como comentario y se salta en la ejecución del programa.

```

10 CHECK HOME
11 PTP HOME  Vel= 100 % DEFAULT
12 ;$OUT[33]=TRUE
13 AUTOEXTINI
14 LOOP

```

**Fig. 10-5: Ejemplo de comentario: Insertar punto y coma**



Los formularios inline **no** se pueden definir como comentarios (mediante ;).

- **Aclaraciones** sobre líneas individuales, así como **indicaciones sobre trabajos pendientes**: marcado de partes no definidas del programa

```

17
18 CASE 1
19 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0) ; Reset
  ↳ Progr.No.-Request
20 Main_Prog( ) ; Call User-Program

```

**Fig. 10-6: Ejemplo de comentario: Explicaciones**

**AVISO** El uso de comentarios tienen sentido únicamente si se mantienen actualizados. Es obligatorio actualizar los comentarios siempre que se realicen modificaciones en las instrucciones.

**Pueden introducirse diferentes tipos de comentarios:**

- Punto y coma (nivel experto): al introducir un punto y coma (" ; "), la parte de la línea que sigue al mismo se convierte en un comentario.
- Introducción del formulario inline "Comentario"

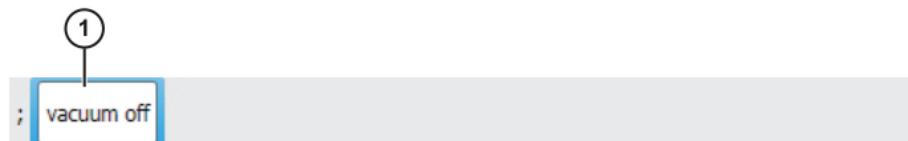


Fig. 10-7: Formulario inline Comentario

Pos.	Descripción
1	Texto cualquiera

- Insertar el formulario inline "Sello": aquí se inserta una marca de tiempo adicional. Asimismo, existe la posibilidad de introducir el nombre del editor de la marca.



Fig. 10-8: Formulario inline Sello

Pos.	Descripción
1	Fecha del sistema (no editable)
2	Hora del sistema (no editable)
3	Nombre o identificación del usuario
4	Texto cualquiera

**Insertar marcas y comentarios relativos al procedimiento**

1. Marcar la línea detrás de la cuales debe insertarse el comentario o la marca.
2. Seleccionar la secuencia de menú **Instrucciones > Comentario > Normal o Marca**.
3. Introducir los datos deseados. Si ya se introdujo anteriormente un comentario o una marca, el formulario inline contendrá las mismas indicaciones.
  - En el caso de los comentarios, con **Texto NUEVO** es posible vaciar el campo para poder introducir texto nuevo.
  - En el caso de las marcas, con **Hora NUEVA** es posible actualizar el sistema y con **Nombre NUEVO**, es posible vaciar el campo **NOMBRE**.
4. Guardar con **Instrucción OK**.

**Acoplar líneas de programa**

Un método efectivo para aumentar la legibilidad de un programa de robot es el sangrado de líneas de programa. De este modo se consigue aclarar la relación entre módulos de programa.

```

13 AUTOEXTINI
14 LOOP
15 P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
16 SWITCH PGNO ; Select with Programnumber
17 CASE 1
18 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
19 Main_Prog( ) ; Call User-Program
20 CASE 2
21 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
22 Sub_Prog1( ) ; Call User-Program
23 DEFAULT
24 P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
25 END SWITCH
26 ENDLOOP

```

Fig. 10-9: Acoplar líneas de programa

**AVISO**

El efecto del sangrado es meramente visual. Durante la ejecución del programa las partes con sangrado se procesan igual que las que no tienen sangrado.

**Ocultar las líneas del programa a través de FOLD**

- El lenguaje del robot de KUKA ofrece la posibilidad de agrupar y ocultar en *FOLD* las líneas del programa.
- Así, ciertas partes del programa no serán visibles para el usuario, lo que simplifica la lectura del programa.
- En el grupo de usuario Experto es posible abrir y editar los FOLD.

```

13
14
15 CHECK HOME
16

```

Fig. 10-10: Fold cerrado

```

14
15 CHECK HOME
16 $H_POS=XHOME
17 IF CHECK_HOME==TRUE THEN
18 P00 (#CHK_HOME,#PGNO_GET,DMY[],0 ) ;Test HPos
19 ENDIF
20

```

Fig. 10-11: Fold abierto

**Marcas de color en las folds:**

Color	Descripción
Rojo oscuro	Fold cerrado
Rojo claro	Fold abierto
Azul oscuro	Subfold cerrado
Azul claro	Subfold abierto
Verde	Contenido del fold

**10.4 Concatenar programas de robot****Técnica de subprogramas**

La técnica de subprogramas permite estructurar modularmente los programas de robot y, así, lograr una composición estructuralmente eficiente. El objetivo

no es registrar todas las instrucciones en un programa, si no almacenar determinadas secuencias, cálculos o procesos en programas independientes.

#### Ventajas de la utilización de subprogramas:

- El programa principal obtiene una estructura clara y es más fácil de leer, puesto que se reduce la longitud del programa.
- Los subprogramas se pueden desarrollar por separado: el esfuerzo de programación puede dividirse y se minimizan las causas de fallos.
- Los subprogramas pueden reutilizarse varias veces.

#### Básicamente, se pueden distinguir dos tipos de subprogramas:

- Subprogramas globales

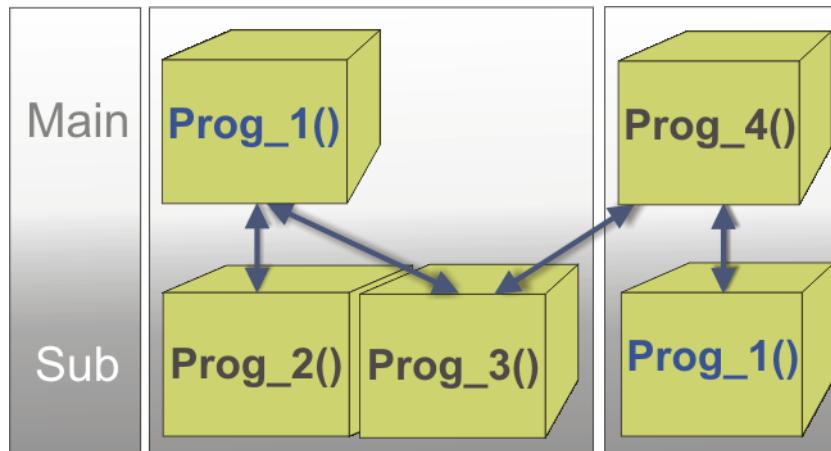


Fig. 10-12: Ejemplo de esquema para subprogramas globales

Un subprograma global es un programa de robot independiente que se activa con otro programa de robot. La ramificación de los programas puede realizarse específicamente según la demanda, es decir, un programa puede ser una vez un programa principal, pero la siguiente vez, puede operar como un subprograma.

- Subprogramas locales

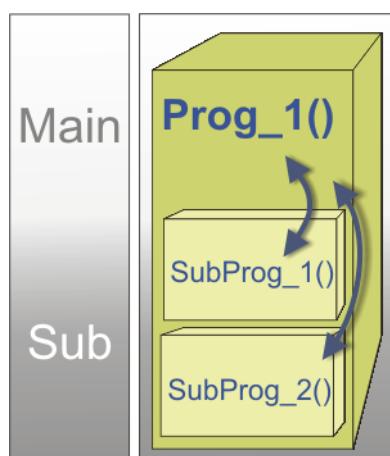


Fig. 10-13: Esquema: subprogramas locales

Los subprogramas locales son programas integrados en un programa principal, es decir, las instrucciones están incluidas en el mismo archivo SRC. Así, las coordenadas de punto del subprograma se guardan en el mismo archivo DAT.

### Procedimiento de apertura de un subprograma

Cada programa empieza con una línea DEF y acaba con una END. En caso de que se active un subprograma en el programa principal se procesará, por regla general, ese subprograma de DEF a END. Al llegar a la línea END, el puntero de ejecución del programa saltará otra vez al programa desde el que se abrió (programa principal).

```

1 DEF main( )
2INI
3 PTP HOME Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[2] Base[2]
5 PTP P2 Vel=100 % PDAT2 Tool[2] Base[2]
6 PTP P3 Vel=100 %
7
8 sub_prog()
9
10 PTP P4 Vel=100 %
11 PTP P5 Vel=100 %
12 PTP P6 Vel=100 % PDAT6 Tool[2] Base[2]
13 PTP HOME Vel= 100 % DEFAULT
14 END
15
  
```

```

1 DEF Sub_Prog( )
2INI
3 PTP P1 Vel=100 % PDAT1 Tool[2] Base[2]
4 OUT 25' State=TRUE
5 PTP P4 Vel=100 % PDAT4 Tool[2] Base[2]
6 END
  
```

Fig. 10-14: Procedimiento de apertura de un subprograma

#### AVISO

Para poder abandonar un subprograma antes de tiempo (es decir, antes de las líneas DEF y END), se puede programar la instrucción RETRUN. La lectura de estas líneas de programa provoca la cancelación anticipada del subprograma.

### Procedimiento para abrir un subprograma

Para poder programar la apertura de un subprograma, debe estar seleccionando el grupo de usuario Experto. La **sintaxis** para la apertura de un subprograma es:

Nombre( )

1. Seleccionar en el menú principal la secuencia **Configuración > Grupo de usuario**. Se muestra el grupo de usuario actual.
2. Para cambiar a otro grupo de usuario: Pulsar en **Iniciar sesión....** Marcar el grupo de usuario **Experto**.
3. Introducir la contraseña **kuka** y confirmar con **Iniciar sesión**.
4. Cargar el programa principal deseado en el editor pulsando en **Abrir**.

```
INI
```

```
PTP HOME Vel= 100% DEFAULT
```

```
PTP HOME Vel= 100% DEFAULT
```

5. Posicionar el cursor en la línea deseada.
6. Introducir el nombre del subprograma entre paréntesis, p. ej., **myprog()**.

```
INI
```

```
PTP HOME Vel= 100% DEFAULT
```

```
myprog( )
```

```
PTP HOME Vel= 100% DEFAULT
```

7. Cerrar el editor pulsando sobre el símbolo de cierre y guardar las modificaciones.

## 10.5 Ejercicio: Programar una llamada a subprograma

**Objetivo del ejercicio** Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Programación de llamadas de subprogramas

**Requisitos** Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos en la utilización del navegador en el nivel del experto
- Conocimientos principales de la programación a nivel de experto (KRL)

**Formulación de tarea** Ejecutar las siguientes tareas:

1. Crear un nuevo módulo con la denominación **Procedimiento** en el nivel experto.
2. A partir de este programa central se llaman todos los demás programas como subprogramas.

El desarrollo exacto del programa se encuentra en el plano de desarrollo del programa. ([>>> Fig. 10-15](#))

3. Después de que el robot haya ejecutado todos los subprogramas globales, en el programa **Procedimiento** se debe volver al principio del mismo y se deben volver a llamar a llamar a los subprogramas. Para esto es necesario el bucle infinito **LOOP - ENDLOOP**, el cual se describe en detalle en uno de los siguientes capítulos. ([>>> 12.2.1 "Programar un bucle sínfin" Página 217](#)) Para el ejercicio, ampliar el programa como sigue:

```
....  
LOOP  
    subprogram_1 ()  
    subprogram_2 ()  
    subprogram_3 ()  
    subprogram_4 ()  
    subprogram_5 ()  
    subprogram_6 ()  
ENDLOOP  
....
```

4. Compruebe su programa nuevo **Procedimiento** en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.

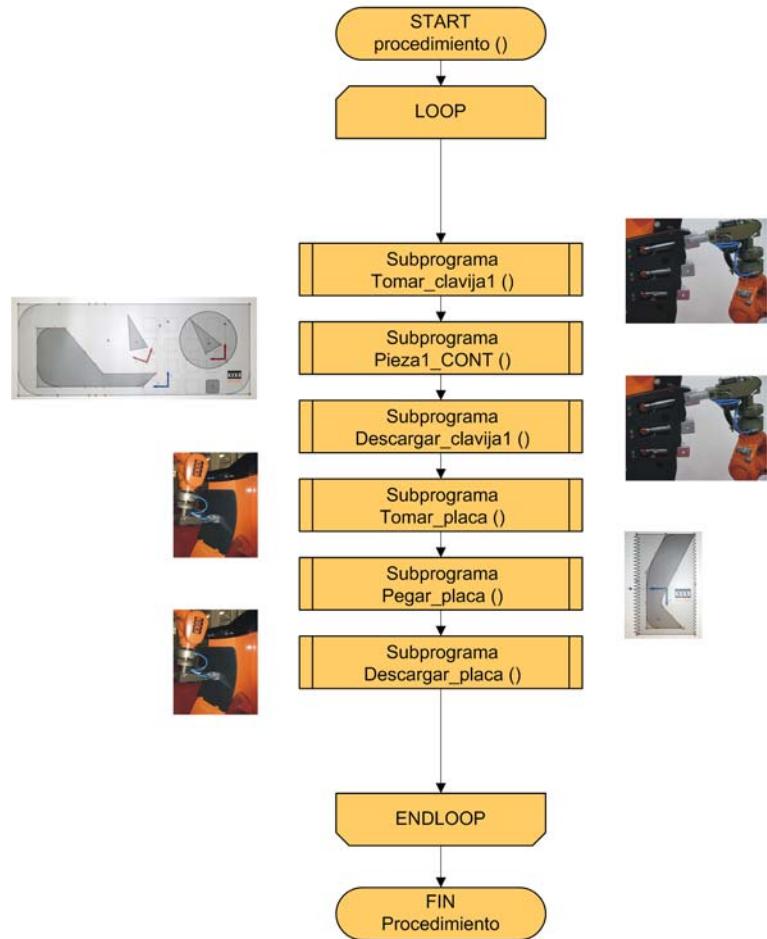


Fig. 10-15: PEP - Llamada del subprograma

Lo que se debe saber tras el ejercicio:

1. ¿Qué significan las extensiones SRC y DAT de los ficheros KUKA?

.....

2. ¿Cómo se realiza la llamada a un subprograma y qué es lo importante?

.....

3. ¿Cuál es la diferencia entre subprogramas locales y globales?

.....

# 11 Variables y declaraciones

## 11.1 Resumen

Esta unidad incluye los siguientes contenidos:

- Gestión de datos en KRL
- Trabajar con tipos de datos simples
- Indicación de variables

## 11.2 Gestión de datos en KRL

**Generalidades sobre variables**

Variable	
Luego almac.	local / global
Tipo	Nro. entero/nro. decimal, verd./falso, caracter
Nombre	Nombre
Valor	Contenido / valor

Fig. 11-1: Identificación de variables

- En la programación de robots con KRL, una variable es en, sentido amplio de la palabra, simplemente un recipiente para operandos ("valores") que aparecen en el desarrollo de un proceso del robot.
- Una variable dispone de una dirección determinada asignada en la memoria de un ordenador.
- Una variable se denomina mediante un nombre que no es una palabra clave de KUKA.
- Cada variable está unida a un tipo determinado de archivos.
- La declaración del tipo de datos es necesaria antes de la utilización.
- En KRL se distingue entre variables locales y globales.

**Convenciones sobre los nombres**

En la elección del nombre de la variable se deben tener en cuenta las siguientes reglas:

- Los nombres en KRL pueden tener una longitud máxima de 24 caracteres.
- Los nombres en KRL pueden contener letras (A-Z), cifras (0-9) y los caracteres especiales "\_" y "\$".
- Los nombres en KRL no deben comenzar con cifras.
- Los nombres en KRL no deben ser palabras clave.
- La utilización de mayúsculas y minúsculas es indiferente.

Consejos

- Utilizar nombres de variables adecuados y autoexplicativos.
- No utilizar nombres o abreviaturas crípticos.
- Utilizar una longitud de nombre adecuada, no gastar cada vez 24 caracteres.

**Doble declaración de variables**

- Una doble declaración ocurre siempre que se usan los mismos nombres de variables (cadenas).
- **No** se trata de una doble declaración cuando se utiliza el mismo nombre en archivos SRC o DAT distintos.
- Las dobles declaraciones en el mismo archivo SRCy DAT no están permitidas y generan un mensaje de error.
- Las dobles declaraciones en el archivo SRC o DAT y en el archivo \$CONFIG.DAT están permitidas:

- Durante la ejecución de la rutina de programación en la que ha sido declarada la variable, solo se modifica el valor local y no el valor en el archivo \$CONFIG.DAT.
- Durante la ejecución de la rutina de programación "ajena" solo se recurre al valor procedente del archivo \$CONFIG.DAT y se modifica.

### Tipos de datos en KRL

#### ■ Tipos de datos estándar predefinidos

- **BOOL:** resultados del tipo "SÍ"/"NO"
- **REAL:** número de coma flotante, resultados de operaciones, para poder evitar errores de redondeo
- **INT:** número entero, típica variable de recuento para bucles de recuento o contadores de piezas
- **CHAR:** solo un carácter

La cadena o el texto solo se puede realizar como campo CHAR

Tipos de datos simples	Número entero	Número con coma flotante	Valores lógicos	Carácter individual
Palabra clave	INT	REAL	BOOL	CHAR
Rango de valores	$-2^{31} \dots (2^{31}-1)$	$\pm 1.1 \cdot 10^{-38} \dots \pm 3.4 \cdot 10^{38}$	TRUE / FALSE	Juego de caracteres ASCII
Ejemplos	-199 o 56	-0,0000123 o 3,1415	TRUE o FALSE	"A" o "q" o "7"

#### ■ Campos / Grupo

```
Voltage[10] = 12.75
Voltage[11] = 15.59
```

- Guardar varias variables del mismo tipo de datos mediante el índice
- Durante la inicialización o en caso de cambio del valor se debe indicar el índice
- El tamaño máximo del campo depende de las necesidades de almacenamiento del tipo de datos

#### ■ Tipo de datos de enumeración

```
color = #red
```

- Todos los valores del tipo de enumeración se definen en la creación con nombre (en forma de texto)
- El sistema también determina un orden
- El número máximo de elementos depende del espacio de almacenamiento

#### ■ Tipo de datos combinado / estructura

```
Date = {day 14, month 12, year 1996}
```

- Tipo de datos combinado de componentes de diferentes tipos de datos
- Los componentes pueden estar compuestos por tipos de datos simples, pero también por estructuras
- Es posible el acceso a componentes individuales

### Creación de variables

#### Declaración de variables

- La declaración se debe realizar siempre antes de su uso.
- A cada variable se le debe asignar un tipo de datos.
- Se deben respetar las convenciones sobre nombres en la asignación de nombres.
- La palabra clave para la declaración es **DECL**.

- Se puede suprimir la palabra clave DECL en los cuatro tipos de datos simples.
- Las asignaciones de valores se realizan en avance.
- La declaración de variables se puede realizar en diferentes puntos. Esto afecta a la vida útil y la validez de las correspondientes variables.

## **Vida útil y validez de las variables**

### **Vida útil de las variables en KRL**

- Por vida útil se entiende el periodo de tiempo en el que se reserva memoria para la variable.
- Las variables de duración temporal vuelven a liberar su espacio de almacenamiento al abandonar el programa o la función.
- Las variables de una lista de datos mantienen su valor actual (el más reciente) de forma permanente en la memoria.

### **Validez de las variables en KRL**

- Las variables declaradas como locales solo están disponibles y visibles en el programa en el cual han sido declaradas.
- Las variables globales están registradas en una lista de datos central (global).
- Las variables globales se pueden registrar también en un lista de datos local y aplicarles la palabra clave **global** al declararlas.

## **Declaración de variables en función del lugar de almacenamiento**

### **Variable en el fichero SRC**

- Una variable creada en el *archivo SRC* se denomina variable de tiempo de ejecución.
- Esta no siempre se puede visualizar.
- Solo se encuentra disponible en la rutina de programación en la que ha sido declarada. Por tanto, la variable estará disponible durante la ejecución del programa. (Programa principal o subprograma local).
- Deja libre de nuevo su espacio de memoria al alcanzar la última línea del programa (línea END).

### **Variable en el archivo DAT local**

- Se puede visualizar siempre durante la ejecución del programa del *archivo SRC* correspondiente.
- El valor de la variable se mantiene tras la finalización del programa.
- Está disponible en todo el *archivo SRC*, es decir, también en los subprogramas locales.
- También se puede crear como variable global.
- Recibe el valor actual en el *archivo DAT* y comienza con el valor guardado en una nueva llamada.
- Si la variable se declara como global, también está disponible a nivel global. Permite un acceso de lectura y de escritura en todas las rutinas de programación, si en el archivo DAT se define la palabra clave PUBLIC y en la declaración se usa adicionalmente la palabra clave GLOBAL.

### **Variable en el archivo de sistema \$CONFIG.DAT**

- Está disponible en todos los programas (global).
- Se puede visualizar siempre, incluso si ningún programa está activo.
- Está disponible de forma global, es decir, en todas las rutinas de programación es posible un acceso de lectura y de escritura.
- Guarda el valor actual en el archivo \$CONFIG.DAT.
- Los datos de sistema KUKA están presentes en todos los tipos de datos, p. ej., como
  - Tipo de datos de enumeración, p. ej.: modo de servicio

## **Datos de sistema de KUKA**

- Estructura, p. ej.: fecha/hora
- Las informaciones de sistema se obtienen a través de variables de sistema de KUKA. Estas
  - leen la información actual del sistema
  - modifican las configuraciones actuales del sistema
  - están predefinidas y comienzan con el carácter "\$"
    - \$DATE (fecha y hora actuales)
    - \$POS\_ACT (posición actual del robot)
    - \$MODE\_OP(modo de servicio actual)
  - ...

## 11.3 Trabajar con tipos de datos simples

A continuación se explica la creación, la inicialización y la modificación de variables. En este caso solo se utilizan los tipos de datos simples.

### Tipos de datos simples con KRL

- Números enteros (INT)
- Números con coma flotante (REAL)
- Valores lógicos (BOOL)
- Carácter individual (CHAR)

#### 11.3.1 Declaración de variables

##### Principio de la declaración de variables

##### Ejemplo de estructura del programa en el fichero SRC

- En la sección de declaración se deben declarar variables
- La sección de inicialización comienza con la primera asignación de valor, aunque a menudo con la línea "INI"
- En la sección de instrucciones se asignan o se modifican valores

```
DEF main( )
: Sección de declaraciones
...
; Sección de inicialización
INI
...
; Sección de instrucciones
PTP HOME Vel=100% DEFAULT
...
END
```

##### Modificar la vista estándar

- La visualización de la línea DEF solo es posible como experto
- El proceso es necesario para acceder a la sección de declaración en los módulos delante de la línea "INI".
- Para poder ver la línea DEF y END, también es importante para la transferencia de variables a subprogramas

##### Planificar la declaración de variables

- Establecer la vida útil
- Determinar la validez/disponibilidad
- Determinar el tipo de datos
- Asignación de nombre y declaración

**Procedimiento en la declaración de variables con un tipo de datos simple**

**Crear la variable en el fichero SRC**

1. Grupo de usuario Experto
2. Dejar mostrar la línea DEF
3. Abrir el fichero SRC en el editor
4. Realizar la declaración de las variables

```
DEF MY_PROG ( )
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
INI
...
END
```

5. Cerrar el programa y guardar

**Crear la variable en el fichero DAT**

1. Grupo de usuario Experto
2. Abrir el fichero DAT en el editor
3. Realizar la declaración de las variables

```
DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
...
ENDDAT
```

4. Cerrar la lista de datos y guardar

**Crear variables en el fichero \$CONFIG.DAT**

1. Grupo de usuario Experto
2. Abrir en la carpeta SYSTEM el fichero \$CONFIG.DAT en el editor

```
DEFDAT $CONFIG
BASISTECH GLOBALS
AUTOEXT GLOBALS
USER GLOBALS
ENDDAT
```

3. Seleccionar el Fold "USER GLOBALS" y abrir con la tecla de función programable "Abrir/Cerrar Fold"
4. Realizar la declaración de las variables

```
DEFDAT $CONFIG
...
;=====
; Userdefined Types
;=====
; Userdefined Externals
;=====
; Userdefined Variables
;=====
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
...
ENDDAT
```

5. Cerrar la lista de datos y guardar

### Crear la variable global en el fichero DAT

1. Grupo de usuario Experto
2. Abrir el fichero DAT en el editor
3. Ampliar lista de datos en el encabezamiento del programa con la palabra clave PUBLIC

```
DEFDAT MY_PROG PUBLIC
```

4. Realizar la declaración de las variables

```
DEFDAT MY_PROG PUBLIC
EXTERNAL DECLARATIONS
DECL GLOBAL INT counter
DECL GLOBAL REAL price
DECL GLOBAL BOOL error
DECL GLOBAL CHAR symbol
...
ENDDAT
```

5. Cerrar la lista de datos y guardar

### 11.3.2 Inicialización de variables con tipos de datos simples

#### Descripción de la inicialización con KRL

- Despues de la declaración, la variable solo tiene reservado un espacio de memoria. Su valor siempre es un valor no válido.
  - En el archivo SRC se realiza la declaración y la inicialización siempre en dos líneas separadas.
  - En el archivo DAT se realiza la declaración y la inicialización siempre en una línea.
- Un constante solo puede estar declarada en una lista de datos y debe ser inicializada inmediatamente.
- La sección de inicialización comienza con la primera asignación de valor.

#### Principio de la inicialización

##### Inicialización de números enteros

- Inicialización como número decimal

```
value = 58
```

- Inicialización como número binario

```
value = 'B111010'
```

Cálculo:  $1*32+1*16+1*8+0*4+1*2+0*1 = 58$

Binario	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Dec	32	16	8	4	2	1

- Inicialización hexadecimal

```
value = 'H3A'
```

Cálculo:  $3*16 + 10 = 58$

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

#### Procedimiento en la inicialización con KRL

##### Declaración e inicialización en el fichero SRC

1. Abrir el fichero SRC en el editor
2. La declaración se ha realizado
3. Realizar la inicialización

```

DEF MY_PROG ( )
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
INI
counter = 10
price = 0.0
error = FALSE
symbol = "X"
...
END

```

#### 4. Cerrar el programa y guardar

#### **Declaración e inicialización en el fichero DAT**

1. Abrir el fichero DAT en el editor
2. La declaración se ha realizado
3. Realizar la inicialización

```

DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL INT counter = 10
DECL REAL price = 0.0
DECL BOOL error = FALSE
DECL CHAR symbol = "X"
...
ENDDAT

```

#### 4. Cerrar la lista de datos y guardar

#### **Declaración en el fichero DAT e inicialización en el fichero SRC**

1. Abrir el fichero DAT en el editor
2. Realizar la declaración

```

DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL INT counter
DECL REAL price
DECL BOOL error
DECL CHAR symbol
...
ENDDAT

```

#### 3. Cerrar la lista de datos y guardar

4. Abrir el fichero SRC en el editor
5. Realizar la inicialización

```

DEF MY_PROG ( )
...
INI
counter = 10
price = 0.0
error = FALSE
symbol = "X"
...
END

```

#### 6. Cerrar el programa y guardar

#### **Declaración e inicialización de una constante**

##### Descripción

- Las constantes se crean con la palabra clave **CONST**.
- Las constantes solo se pueden crear en listas de datos.

##### creación de constantes

1. Abrir el fichero DAT en el editor.
2. Realizar la declaración y la inicialización.

```

DEFDAT MY_PROG
EXTERNAL DECLARATIONS
DECL CONST INT max_size = 99
DECL CONST REAL PI = 3.1415
...
ENDDAT

```

3. Cerrar la lista de datos y guardar.

### 11.3.3 Manipulación de valores de variables de tipos de datos simples con KRL

#### Listado de las posibilidades para la modificación de valores de variables con KRL

Los valores de las variables se modifican de diferentes maneras en las rutinas de programación (fichero SRC) en función de la tarea. A continuación se comentarán los métodos más comunes. La manipulación mediante operaciones de bit y funciones estándar es posible, pero aquí no se profundizará en ello.

#### Manipulación de datos por medio de

- **Tipos de cálculo básicos**
  - (+) Suma
  - (-) Resta
  - (\*) Multiplicación
  - (/) División
- Operaciones de comparación
  - (==) idéntico / igualdad
  - (<>) desigual
  - (>) mayor que
  - (<) menor que
  - (>=) mayor o igual que
  - (<=) menor o igual que
- Operaciones lógicas
  - (**NOT**) Inversión
  - (**AND**) Y lógico
  - (**OR**) O lógico
  - (**EXOR**) O excluyente
- Operaciones de bit
  - (**B\_NOT**) Inversión por bits
  - (**B\_AND**) Combinación de Y por bits
  - (**B\_OR**) Combinación de O por bits
  - (**B\_EXOR**) Combinación de O excluyente por bits

#### Funciones estándar

- Función absoluta
- Función raíz
- Función de seno y coseno
- Función de tangente
- Función de arcocoseno
- Función de arcotangente
- Varias funciones para la manipulación de cadenas

#### Relaciones en la manipulación de datos

Modificación de valores con la utilización del tipo de datos REAL y INT

- Redondear hacia arriba/hacia abajo

```

; Declaración
DECL INT A,B,C
DECL REAL R,S,T
; Inicialización
A = 3           ; A=3
B = 5.5         ; B=6 (a partir de x.5 se redondea a la alza)
C = 2.25        ; C=2 (redondear a la baja)
R = 4           ; R=4.0
S = 6.5         ; S=6.5
T = C           ; T=2.0 (se utiliza el valor redondeado a la baja)

```

#### ■ Resultados de operaciones aritméticas (+;-;\*)

Operandos	INT	REAL
INT	INT	REAL
REAL	REAL	REAL

```

; Declaración
DECL INT D,E
DECL REAL U,V
; Inicialización
D = 2
E = 5
U = 0.5
V = 10.6
; Sección de instrucciones (manipulación de datos)
D = D*E ; D = 2 * 5 = 10
E = E+V ; E= 5 + 10.6 = 15.6 -> Redondear a la alza E=16
U = U*V ; U= 0.5 * 10.6 = 5.3
V = E+V ; V= 16 + 10.6 = 26.6

```

#### ■ Resultados de operaciones aritméticas (/)

Particularidades en las operaciones aritméticas con valores enteros:

- En caso de resultados intermedios de operaciones enteras puras, se eliminarán todas las posiciones decimales.
- En la asignación de valor a una variable entera, el resultado se redondeará de acuerdo con las reglas normales de cálculo.

```

; Declaración
DECL INT F
DECL REAL W
; Inicialización
F = 10
W = 10.0
; Sección de instrucciones (manipulación de datos)
; INT / INT -> INT
F = F/2 ; F=5
F = 10/4 ; F=2 (10/4 = 2.5 -> Cortar posición decimal)
; REAL / INT -> REAL
F = W/4 ; F=3 (10.0/4=2.5 -> Redondear a la alza)
W = W/4 ; W=2.5

```

### Operaciones de comparación

Con las operaciones de comparación se pueden formar expresiones lógicas. El resultado de una comparación siempre es del tipo de datos BOOL.

Operador / KRL	Descripción	Tipos de datos admisibles
==	idéntico/igual-dad	INT, REAL, CHAR, BOOL
<>	desigual	INT, REAL, CHAR, BOOL
>	mayor que	INT, REAL, CHAR
<	menor que	INT, REAL, CHAR

Operador / KRL	Descripción	Tipos de datos admisibles
<b>&gt;=</b>	mayor o igual que	INT, REAL, CHAR
<b>&lt;=</b>	menor o igual que	INT, REAL, CHAR

```
; Declaración
DECL BOOL G,H
; Inicialización/sección de instrucciones
G = 10>10.1 ; G=FALSE
H = 10/3 == 3 ; H=TRUE
G = G<>H ; G=TRUE
```

### Operaciones lógicas

Con las operaciones lógicas se pueden formar expresiones lógicas. El resultado de una operación de este tipo siempre es del tipo de datos BOOL.

Operaciones		NOT A	A AND B	A OR B	A EXOR B
A=FALSE	B=FALSE	TRUE	FALSE	FALSE	FALSE
A=FALSE	B=TRUE	TRUE	FALSE	TRUE	TRUE
A=TRUE	B=FALSE	FALSE	FALSE	TRUE	TRUE
A=TRUE	B=TRUE	FALSE	TRUE	TRUE	FALSE

```
; Declaración
DECL BOOL K,L,M
; Inicialización/sección de instrucciones
K = TRUE
L = NOT K ; L=FALSE
M = (K AND L) OR (K EXOR L) ; M=TRUE
L = NOT (NOT K) ; L=TRUE
```

Los operadores se ejecutan en el orden de su prioridad

Prioridad	Operador
1	NOT (B_NOT)
2	Multiplicación (*); división (/)
3	Suma (+), resta (-)
4	AND (B_AND)
5	EXOR (B_EXOR)
6	OR (B_OR)
7	cualquier comparación (==; <>; ...)

```
; Declaración
DECL BOOL X, Y
DECL INT Z
; Inicialización/sección de instrucciones
X = TRUE
Z = 4
Y = (4*Z+16 <> 32) AND X ; Y=FALSE
```

- Procedimiento para la manipulación de datos**
1. Especificar el tipo de datos para la variable o variables
  2. Determinar la validez y la vida útil de las variables
  3. Realizar la declaración de variables
  4. Inicializar la variable
  5. Manipular la variable en las rutinas de programación, es decir, siempre en el fichero \*.SRC
  6. Guardar y cerrar el fichero \*.SRC

### Instrucción HALT

HALT se usa, sobre todo, para realizar pruebas en la fase de programación, por ejemplo, para visualizar el contenido de una variable en tiempo de ejecución.

- La instrucción HALT detiene el programa. La última instrucción de movimiento lanzada se ejecuta por completo.
- El programa puede continuarse exclusivamente con la tecla de arranque. A continuación se ejecutará la siguiente instrucción después de HALT.



En un programa de interrupción el programa no se detiene hasta que se haya ejecutado por completo el procesamiento.

#### Ejemplo:

```
DEF program()
DECL BOOL a,b
INI
...
PTP XP1
a=$IN[1]
b=$IN[2]
HALT
IF ((a == TRUE) AND (b == FALSE)) THEN
..
ENDIF
...
```

## 11.4 Visualizar variables

### Mostrar y modificar el valor de una variable

1. Seleccionar en el menú principal **Visualización > Variable > Individual**. Se abre la ventana **Visualización de variable individual**.
  2. Introducir el nombre de la variable en el campo **Nombre**.
  3. Cuando un programa está seleccionado, el programa se incluye automáticamente en el campo **Módulo**. Si debe mostrarse una variable de otro programa, introducir el programa como sigue:  
*/R1/Nombre de programa*
  4. Pulsar la tecla Enter.
- En el campo **Valor actual** se visualiza el valor actual de la variable. Si no se indica nada, entonces significa que todavía no se ha asignado ningún valor a la variable.
5. Introducir el valor que se desee en el campo **Valor nuevo**.
  6. Pulsar la tecla Enter.
- En el campo **Valor actual** aparece el nuevo valor.



Fig. 11-2: Ventana Visualización de variable individual

Pos.	Descripción
1	Nombre de la variable que debe modificarse.
2	Nuevo valor que se debe asignar a la variable.
3	Programa en el cual se busca la variable. En las variables del sistema el campo <b>Módulo</b> no es relevante.
4	<p>Este campo tiene dos estados:</p> <ul style="list-style-type: none"> <li>■  : el valor mostrado no se actualiza de forma automática.</li> <li>■  : el valor mostrado se actualiza de forma automática.</li> </ul> <p>Para alternar entre los diferentes estados:</p> <ul style="list-style-type: none"> <li>■ Pulsar <b>Actualizar</b>.</li> <li>■ Alternativa: <b>Shift + Enter</b></li> </ul>

### Visualización de informaciones del sistema

Procedimiento para visualizar marcas, contadores y temporizadores:

- Seleccionar en el menú principal la secuencia **Visualización > Variable**.

Se pueden seleccionar diferentes variables del sistema:

- **Marcas cíclicas**
- **Marcas**
- **Contador**
- **Temporizador**

Procedimiento para visualizar entradas y salidas:

- Seleccionar en el menú principal la secuencia **Visualización > Entradas/salidas > Entradas digitales o Salidas digitales**.

## 11.5 Ejercicio: Tipos de datos simples

**Objetivo del ejercicio** Despu s de completar correctamente este ejercicio, se dispondr  de la competencia necesaria para efectuar las siguientes tareas:

- Utilizaci n de tipos de datos simples
- Declaraci n, inicializaci n y utilizaci n de variables
- Utilizaci n correcta de variables en relaci n con su vida  til

**Requisitos** Los siguientes requisitos son imprescindibles para superar el ejercicio con  xito:

- Conocimientos te ricos sobre tipos de datos simples y su manipulaci n

**Formulaci n de tarea** Crear un nuevo programa denominado Procedimiento2:

- Copiar el programa Procedimiento y borrar las partes del programa relativas a la clavija y al contorno de pieza, de forma que solo se recoja la placa, se pegue y se vuelva a depositar.
- Una variable cuenta las veces que se ha ejecutado el programa *Pegar\_placa* desde la  ltima selecci n de programa.
- Una variable cuenta las veces que se ha ejecutado el programa *Pegar\_placa* en total.
- Una variable debe sumar la longitud total del pegamento aplicado (en metros). La longitud de una trayectoria, para el programa *Pegar\_placa*, es de 0,91 m. La longitud de la trayectoria se debe declarar como constante.
- Una variable que durante el proceso de recogida de la placa est  en *TRUE*, de lo contrario en *FALSE*.
- Una variable que contenga la letra "O" con la garra abierta y con la garra cerrada contenga la letra "G". Durante la inicializaci n, la variable recibe el valor "X".

Utilizar nombres de variables y tipos de datos adecuados. Es importante tambi n declarar la variable en el lugar correcto y de forma conveniente.

Nombre de variable	Tipo de datos	Lugar de la declaraci�n

- Determinar d nde se declaran las variables.
- Ampliar el plan de ejecuci n del programa existente con esta nueva variable.
- Tener en cuenta las diferentes inicializaciones de las variables.
- Comprobar el programa en los modos de servicio T1, T2 y Autom tico. Se deben tener en cuenta las prescripciones de seguridad ense adas.

### Lo que se debe saber tras el ejercicio:

1. ¿Cu l es la longitud m xima para un nombre de variable?

.....

2. ¿Qu  tipos de datos simples existen?

.....

.....  
3. ¿Dónde se declaran las variables en el fichero SRC?

.....  
4. ¿Qué vida útil tiene una variable declarada en el fichero \$CONFIG.DAT?

.....  
5. Declarar un número con coma flotante con el nombre "Value" en el fichero DAT con el valor 138,74.

## 12 Utilización de controles de ejecución de programa

### 12.1 Resumen

Esta unidad incluye los siguientes contenidos:

- Bucles
- Ramificaciones
- Distribuidor
- Instrucción de salto
- Funciones de espera en KRL

### 12.2 Programar bucles

Además de las instrucciones de movimiento y comunicación (funciones de conmutación y espera) en sí, los programas de robot albergan una gran cantidad de rutinas que sirven para el control de la ejecución del programa.

#### Generalidades sobre bucles

- Los bucles son estructuras de control.
- Repiten bloques de instrucciones hasta que se produce una condición de interrupción.
- Un salto desde fuera a un cuerpo de bucle no está permitido
- Los bucles se pueden intercalar entre sí
- Existen diferentes tipos de bucles
  - Bucle sinfín
  - Bucle de conteo
  - Bucle condicionados
    - bucle finito
    - bucle infinito

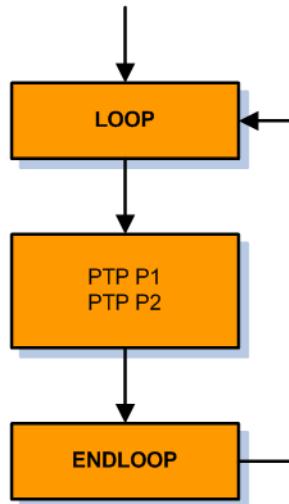
#### 12.2.1 Programar un bucle sinfín

##### Descripción de un bucle sinfín

- El bucle infinito es un bucle que se vuelven a ejecutar después de cada ejecución.
- La ejecución se puede interrumpir por influencias externas.
- Sintaxis

```
LOOP  
;   Instrucción  
...  
;   Instrucción  
ENDLOOP
```

## Principio de un bucle sinfín



**Fig. 12-1: Plano de desarrollo del programa: Bucle sifín**

- El bucle sifín se puede abandonar mediante EXIT
- Al abandonar un bucle sifín con EXIT se deberá asegurar la ausencia de colisión
- En caso de que haya dos bucles infinitos encajados, también serán necesarias dos instrucciones EXIT para abandonar ambos bucles

### Ejemplo para la programación de un bucle sifín

#### Bucle sifín sin interrupción

```

DEF MY_PROG( )
INI
PTP HOME Vel=100% DEFAULT

LOOP
  PTP XP1
  PTP XP2
  PTP XP3
  PTP XP4
ENDLOOP

PTP P5 Vel=30% PDAT5 Tool[1] Base[1]
PTP HOME Vel=100% DEFAULT
END
  
```



El punto P5 nunca se alcanza desde el punto de vista técnico del programa.

#### Bucle sifín con interrupción

```

DEF MY_PROG( )
INI
PTP HOME Vel=100% DEFAULT

LOOP
  PTP XP1
  PTP XP2
  IF $IN[3]==TRUE THEN ; Condición para interrupción
    EXIT
  ENDIF
  PTP XP3
  PTP XP4
ENDLOOP

PTP P5 Vel=30% PDAT5 Tool[1] Base[1]
PTP HOME Vel=100% DEFAULT
END
  
```

**i** El punto P5 se alcanza en cuanto está activa la entrada 3.  
Importante: Se deberá comprobar la ausencia de colisión en el desplazamiento entre P2 y P5.

### 12.2.2 Programar un bucle de conteo

#### Descripción de un bucle de recuento

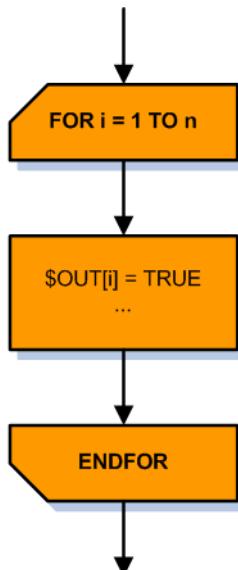


Fig. 12-2: Plano de desarrollo del programa: Bucle de conteo

- El bucle FOR es una estructura de control con la que se puede llevar a cabo una o varias indicaciones con un número fijo de repeticiones.
  - Para un bucle de recuento es necesario declarar con anterioridad un contador de bucle del tipo de datos entero (Integer).
  - El bucle de recuento comienza en el valor start y finaliza, como muy tarde, en el valor last
- Sintaxis con anchura de paso +1

```

FOR counter = start TO last
; Instrucción
ENDFOR
  
```

- La anchura de paso (increment) también se puede indicar como número entero con la palabra clave STEP.

```

FOR counter = start TO last STEP increment
; Instrucción
ENDFOR
  
```

- Del bucle de recuento se puede salir de forma inmediata con EXIT

#### Cuenta ascendente con un bucle de recuento

```

DECL INT counter

FOR counter = 1 TO 3 Step 1
; Instrucción
ENDFOR
  
```

1. El contador de bucles se inicializa con el valor de inicio: counter = 1
2. El contador de bucles se incremente en ENDFOR con la anchura de paso STEP

3. El bucle vuelve a comenzar en la línea FOR
4. Control de la condición de entrada: El contador de bucles debe ser menor o igual que el valor final indicado, en caso contrario se finalizará el bucle
5. Dependiendo de la comprobación, se incrementarán de nuevo el contador de bucles o el bucle se finaliza y el programa continúa después de la línea ENDFOR

#### Ejemplos:

- Bucle de conteo simple sin indicación de la anchura de paso

```
DECL INT counter

FOR counter = 1 TO 50
$OUT[counter] = FALSE
ENDFOR
```



Si no se indica la anchura de paso mediante STEP se utiliza automáticamente la anchura de paso +1.

- Bucle de conteo simple con indicación de la anchura de paso

```
DECL INT counter

FOR counter = 1 TO 4 STEP 2
$OUT[counter] = TRUE
ENDFOR
```



Este bucle solo se recorre dos veces. Una vez con el valor de inicio counter=1 y la segunda vez con counter=3. Con el valor de contador 5, el bucle se interrumpe inmediatamente.

#### Conteo descendente con un bucle de conteo

```
DECL INT counter

FOR counter = 15 TO 1 Step -1
; Instrucción
ENDFOR
```



El valor inicial o valor de inicio del bucle debe ser mayor que el valor final para que el bucle pueda recorrerse varias veces.

#### Ejemplo:

- Bucle de recuento con indicación de anchura de paso negativa

```
DECL INT counter

FOR counter = 10 TO 1 STEP -1
; Instrucción
ENDFOR
```

- Bucle de recuento intercalado con indicación de la anchura de paso

```
DECL INT counter1, counter2

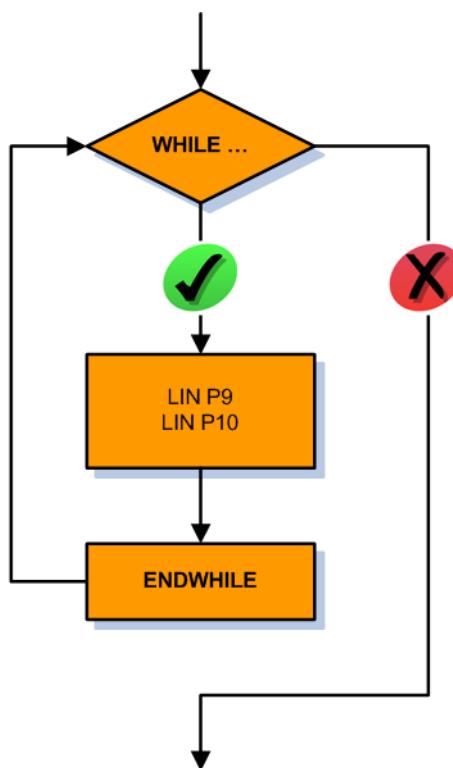
FOR counter1 = 1 TO 21 STEP 2
FOR counter2 = 20 TO 2 STEP -2
...
ENDFOR
ENDFOR
```



Siempre se ejecuta primero el bucle interior, aquí counter2 y, después, el exterior (counter1).

### 12.2.3 Programar un bucle finito

#### Descripción de un bucle finito



**Fig. 12-3: Plano de desarrollo del programa: Bucle finito**

- Un bucle WHILE se denomina también *bucle controlado desde arriba*.
- Un bucle WHILE es un bucle *finito* o *preliminar* al que se le va a comprobar la condición de interrupción antes de que se ejecute la sección de instrucción del bucle.
- Un bucle de este tipo repite procesos mientras que se cumpla una determinada condición de ejecución (*condition*).
- El incumplimiento de la condición de ejecución tiene como consecuencia que el bucle finaliza inmediatamente y se ejecutan las instrucciones que siguen a ENDWHILE
- Sintaxis

```

WHILE condition
    ; Instrucción
ENDWHILE

```

- Del bucle finito se puede salir de forma inmediata con EXIT

#### Bucle finito con condición de ejecución sencilla

```

...
WHILE IN $41]==TRUE ; Pieza preparada en el depósito
    PICK_PART( )
ENDWHILE
...

```



La expresión WHILE \$IN[41]==TRUE también se puede reducir a WHILE \$IN[41]. Una omisión significa siempre la comparación con TRUE.

#### Programación con un bucle finito

■ **Bucle finito con condición de ejecución negada sencilla**

```
...
WHILE NOT $IN[42]==TRUE ; Entrada 42: Depósito está vacío
    PICK_PART( )
ENDWILE...
```

O

```
...
WHILE $IN[42]==FALSE ; Entrada 42: Depósito está vacío
    PICK_PART( )
ENDWILE...
```

■ **Bucle finito con condición de ejecución compleja**

```
...
WHILE (( $IN[40]==TRUE) AND ($IN[41]==FALSE) OR (counter>20))
    PALLET( )
ENDWILE
...
```

#### 12.2.4 Programar un bucle infinito

**Descripción de un bucle infinito**

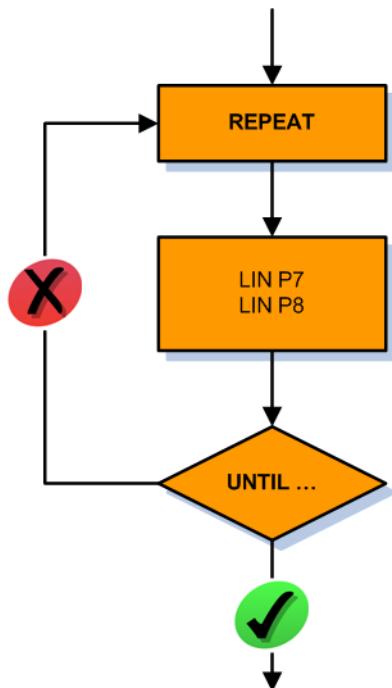


Fig. 12-4: Plano de desarrollo del programa: Bucle infinito

- El bucle infinito también se denomina *bucle controlado desde abajo*.
- Un bucle REPEAT es un bucle *infinito* o *verificable* al que se le va a comprobar la condición de interrupción después que se ejecute por primera vez la sección de instrucción del bucle.
- Una vez ejecutada la sección de instrucciones, se comprueba si se cumple una condición (*condition*) para poder salir del bucle.
  - Si el resultado de la condición es positivo, se sale del bucle y se ejecutan las instrucciones que siguen a UNTIL.
  - Con un resultado negativo de la condición, el bucle se inicia de nuevo con REPEAT.
- El bucle infinito se puede abandonar de forma inmediata mediante EXIT.

- Sintaxis

```
REPEAT
    ; Instrucción
UNTIL condition
```

## Programación de un bucle infinito

- Bucle infinito con condición de ejecución sencilla

```
...
REPEAT
PICK_PART( )
UNTIL $IN[42]==TRUE ; Entrada 42: Depósito está vacío
...
```



La expresión UNTIL \$IN[42]==TRUE también se puede reducir a UNTIL \$IN[42]. Una omisión significa siempre la comparación con TRUE.

- Bucle infinito con condición de ejecución compleja

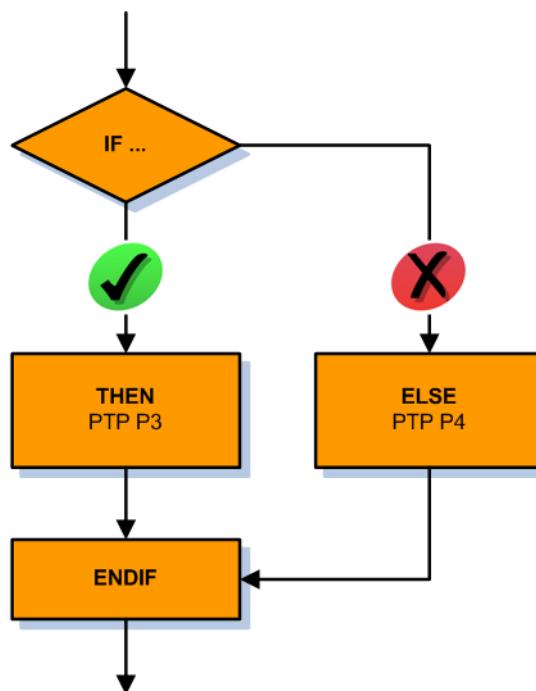
```
...
REPEAT
PALLET( )
UNTIL (($IN[40]==TRUE) AND ($IN[41]==FALSE) OR (counter>20))
...
```



El bucle se finaliza en caso de un resultado positivo.

## 12.3 Programar consultas o ramificaciones

### Descripción de ramificaciones



**Fig. 12-5: Plano de desarrollo del programa: Ramificación IF**

- El uso de ramificaciones permite que secciones del programa se ejecuten únicamente si se da una condición determinada.
- Las ramificaciones se utilizan para dividir un programa en varias rutas.

- Una *ramificación condicionada* (consulta IF) está compuesta por una condición y dos secciones de instrucción.
- La instrucción IF comprueba si esta condición es verdadera (TRUE) o falsa (FALSE).
  - En cuanto se cumple la condición, puede ejecutarse la primera instrucción.
  - Si no se cumple la condición, se ejecuta la segunda instrucción.
- Variaciones de la consulta IF:
  - La segunda sección de instrucción puede omitirse: consulta IF sin ELSE. Por tanto, en caso de que no se cumpla la condición, el programa continuará directamente tras la ramificación.
  - Es posible intercalar varias consultas IF (*ramificación múltiple*): las consultas se ejecutan por orden y se comprueba si se cumple una condición.

## Programación de una ramificación

### Sintaxis

#### ■ con ramificación alternativa

```
IF condition THEN
  Instrucción
ELSE
  Instrucción
ENDIF
```

#### ■ sin ramificación alternativa (consulta)

```
IF condition THEN
  Instrucción
ENDIF
```

## Ejemplos de ramificaciones

### Ramificación con ramificación alternativa

```
DEF MY_PROG( )
DECL INT error_nr
...
INI
error_nr = 4
...
; solo con error_nr 5 se desplaza a P21, de lo contrario a P22
IF error_nr == 5 THEN
  PTP XP21
ELSE
  PTP XP22
ENDIF
...
END
```

### Ramificación sin ramificación alternativa

```
DEF MY_PROG( )
DECL INT error_nr
...
INI
error_nr = 4
...
; solo con error_nr 5 se desplaza a P21
IF error_nr == 5 THEN
  PTP XP21
ENDIF
...
END
```

### Ramificación con condiciones de ejecución complejas

```

DEF MY_PROG( )
DECL INT error_nr
...
INI
error_nr = 4
...
; solo con error_nr 1 o 10 o mayor que 99, se desplaza a P21
IF ((error_nr == 1) OR (error_nr == 10) OR (error_nr > 99)) THEN
    PTP XP21
ENDIF
...
END

```

### Ramificación con expresiones booleanas

```

DEF MY_PROG( )
DECL BOOL no_error
...
INI
no_error = TRUE
...
; solo en caso de ausencia de error (no_error) se desplaza a P21
IF no_error == TRUE THEN
    PTP XP21
ENDIF
...
END

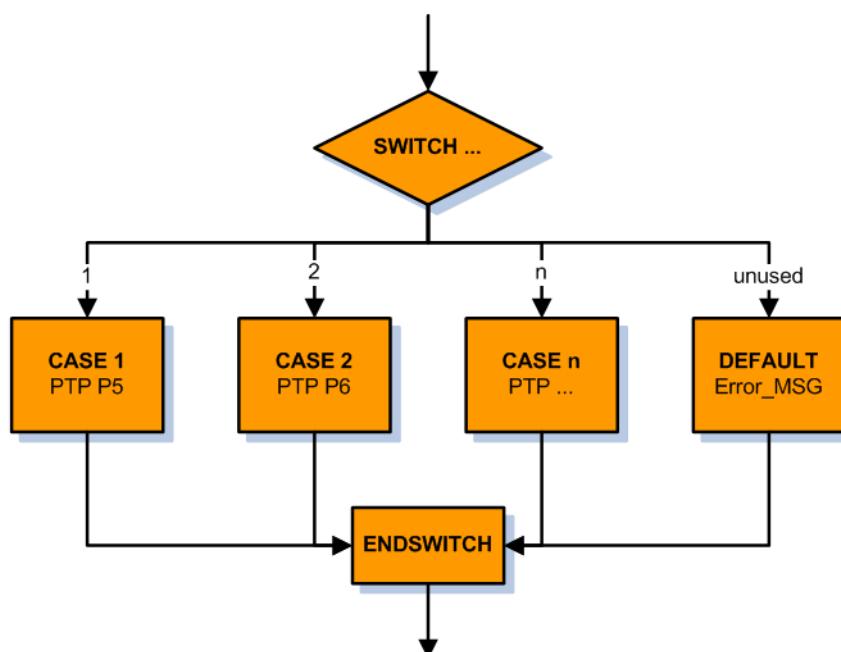
```



La expresión IF no\_error==TRUE THEN también se puede reducir a IF no\_error THEN. Una omisión significa siempre la comparación con TRUE.

## 12.4 Programar distribuidores (SWITCH- CASE)

### Descripción del distribuidor (SWITCH- CASE)



**Fig. 12-6: Plano de desarrollo del programa: Distribuidor SWITCH - CASE**

- Si se desea diferenciar entre varios casos y ejecutar diferentes acciones para cada uno, se puede lograr con una instrucción SWITCH - CASE.

- Una rama SWITCH - CASE es un *distribuidor* o, mejor dicho, un *distribuidor múltiple* y sirve para la diferenciación de casos.
- Una variable transmitida a la instrucción SWITCH se utiliza como comutador y salta, en el bloque de instrucciones, a la instrucción CASE predefinida.
- Si la instrucción SWITCH no encuentra ninguna instrucción CASE predefinida, se ejecutará la sección DEFAULT, en caso de que esta haya sido definida con anterioridad.

### Sintaxis

```

SWITCH Criterio de selección
CASE Valor
Instrucción
CASE Valor
Instrucción
CASE Valor
Instrucción
...
DEFAULT
Instrucción
ENDSWITCH

```

### Transferencia de valores mediante tipos de datos simples

- **SWITCH-CASE** puede ser usada en combinación con los siguientes tipos de datos:

- **INT (número entero)**

Ejemplo:

```

DEF MY_PROG( )
DECL INT my_number
...
INI
my_number = 2
...
SWITCH my_number
CASE 1
    PTP XP21
CASE 2
    PTP XP22
CASE 3
    PTP XP23
ENDSWITCH
...

```

- **CHAR (un carácter)**

Ejemplo:

```

DEF MY_PROG( )
DECL CHAR my_sign
...
INI
my_sign = "a"
...
SWITCH my_sign
CASE "a"
    PTP XP21
CASE "b"
    PTP XP22
CASE "c"
    PTP XP23
ENDSWITCH
...

```

### Variantes SWITCH-CASE

Un distribuidor SWITCH-CASE se puede programar:

- **Solo con distribuidores definidos y sin ruta alternativa**

```

DEF MY_PROG( )
DECL INT error_nr
...
INI
error_nr = 4
...
; sólo se puede desplazar en caso de un caso declarado
SWITCH error_nr
    CASE 1
        PTP XP21
    CASE 2
        PTP XP22
    CASE 3
        PTP XP23
ENDSWITCH
...

```



En caso de *error\_nr* distinto de 1, 2 o 3 se salta directamente a END-SWITCH, sin ejecutar ninguna instrucción.

#### ■ Solo con distribuidores definidos y un caso alternativo

Ejemplo:

```

DEF MY_PROG( )
DECL INT error_nr
...
INI
error_nr = 99
...
; en caso de un caso no definido se desplaza a HOME
SWITCH error_nr
    CASE 1
        PTP XP21
    CASE 2
        PTP XP22
    CASE 3
        PTP XP23
    DEFAULT
        PTP XHOME
ENDSWITCH
...

```



En caso de *error\_nr* distinto de 1, 2 o 3 se salta al "caso DEFAULT" para ejecutar las correspondientes instrucciones.

#### ■ Con varias soluciones en un distribuidor y una ruta alternativa

```

SWITCH number
CASE 1,2
...
CASE 3,4,5
...
CASE 6
...
DEFAULT
...
ENDSWITCH

```



En *number* 3, 4 o 5 se salta al segundo "CASE" para ejecutar las correspondientes instrucciones.

## 12.5 Programar instrucción de salto

### Descripción

- Salto obligatorio al lugar específico del programa. El programa se reanuda en este punto.

- El destino del salto se debe encontrar en la misma parte del programa o en la misma función que la instrucción GOTO.
- Los saltos siguientes **no** son posibles:
  - Saltar del exterior a una instrucción IF.
  - Saltar del exterior a un bucle.
  - Saltar de una instrucción CASE a otra instrucción CASE.



En un salto incondicional no se comprueba, por parte del programa, si se cumple o no una determinada condición.

El salto se ejecuta siempre. Un posible salto no deseado debe ser interceptado por parte de la programación. Ver también el segundo ejemplo de programa.

La programación con GOTO puede generar programas poco claros.  
Mejor: en lugar de ello, trabajar con IF, SWITCH o un bucle.

## Sintaxis

```
...  
GOTO Marke  
...  
Marke:  
...
```

Elemento	Descripción
Marca	Lugar al cual se saltará. En este punto de destino <i>Marca</i> debe tener dos puntos al final.

## Ejemplos

- Salto obligatorio al punto del programa GLUESTOP.

```
GOTO GLUE_STOP  
...  
GLUE_STOP:
```

- Conversión de un salto incondicional a un salto condicionado mediante una ampliación con una instrucción IF. El salto lleva a la posición de programa GLUE\_END.

```
IF X>100 THEN  
    GOTO GLUE_END  
ELSE  
    X=X+1  
ENDIF  
A=A*X  
...  
GLUE_END:  
END
```

## 12.6 Programar funciones de espera en KRL

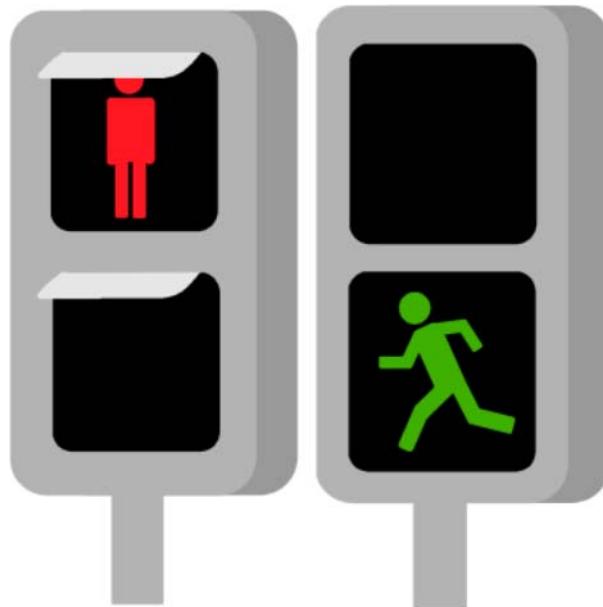


Fig. 12-7

Programación en KRL de

- función de espera dependiente del tiempo
- función de espera dependiente de una señal

### 12.6.1 Función de espera dependiente del tiempo

#### Descripción de una función de espera dependiente del tiempo con KRL

- La función de espera dependiente del tiempo, espera el tiempo indicado (*time*) antes de que se pueda continuar con el proceso
- Sintaxis

```
WAIT SEC time
```

#### Principio de la función de espera dependiente del tiempo

- La función de espera dependiente del tiempo tiene la base de tiempo de segundos (s)
- El tiempo máximo es de 2147484 segundos, que son más de 24 días



El formulario en línea para la función de espera dependiente del tiempo puede esperar como máximo 30 segundos

- El valor del tiempo también se puede transmitir con una variable adecuada
- La unidad de tiempo más pequeña adecuada es de 0,012 segundos (ciclo de interpolación)
- Si el tiempo indicado es negativo, no se esperará
- La función de espera dependiente del tiempo activa una parada del procesamiento en avance, por lo que el posicionamiento aproximado no es posible
- Para generar de forma precisa solo una parada del procesamiento en avance, se utilizar la instrucción WAIT SEC 0

### Programación de una función de espera dependiente del tiempo

- Función de espera dependiente del tiempo con un tiempo fijo

```
PTP P1 Vel=100% PDAT1
PTP P2 Vel=100% PDAT2
WAIT SEC 5.25
PTP P3 Vel=100% PDAT3
```

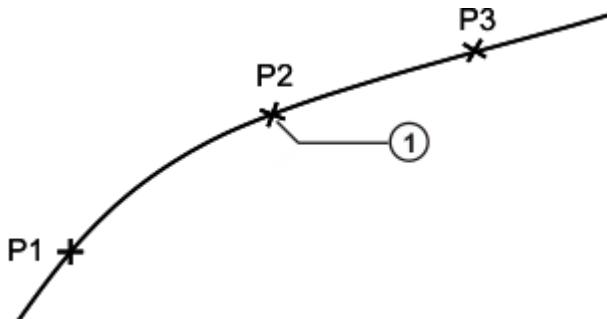


Fig. 12-8: Movimiento ejemplar para lógica

Pos.	Observación
1	El movimiento se interrumpe durante 5.25 segundos en el punto P2

- Función de espera dependiente del tiempo con un tiempo calculado

```
WAIT SEC 3*0.25
```

- Función de espera dependiente del tiempo con una variable

```
DECL REAL time
time = 12.75
WAIT SEC time
```

### 12.6.2 Función de espera dependiente de una señal

#### Descripción de una función de espera dependiente de una señal

- La función de espera dependiente de una señal continúa comutando si se cumple la condición (*condition*) y el proceso continúa
- Sintaxis

```
WAIT FOR condition
```

#### Principio de la función de espera dependiente de una señal

- La función de espera dependiente de una señal activa una parada del procesamiento en avance, por lo que el posicionamiento aproximado no es posible
- A pesar de que se ha cumplido la condición, se genera una parada del procesamiento en avance
- Si en la línea inmediatamente anterior a la instrucción de espera se programa la instrucción *CONTINUE*, se puede evitar una parada de procesamiento en avance en caso de que se cumpla a tiempo la condición.

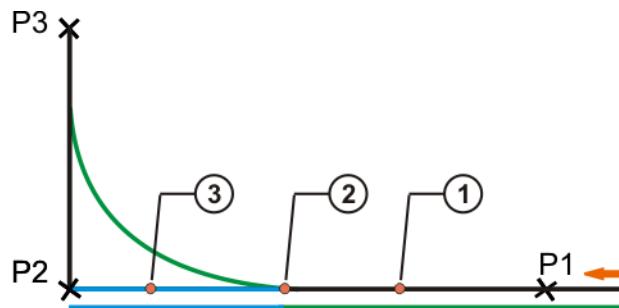


Fig. 12-9: Movimiento ejemplar para lógica con avance

	Posición	Rango de comutación
1	El robot realiza un posicionamiento aproximado en caso de cumplirse la condición de espera antes de entrar en el contorno de aproximación.	Rango de comutación verde para la Activación del contorno de aproximación. Esta solo se ajusta y no se puede volver a desactivar.
2	Inicio del movimiento de aproximación (entrada en el contorno de aproximación)	Consulta de la Activación <ul style="list-style-type: none"> <li>■ TRUE: Posicionamiento aproximado</li> <li>■ FALSE: Acercamiento al punto de destino</li> </ul>
3	El robot no realiza un posicionamiento aproximado, si la condición de espera se cumple después de haber entrado en el contorno de aproximación.	Rango de comutación azul para el acercamiento y la parada en el punto P2

### Programación de una función de espera dependiente de una señal

- WAIT FOR con parada del procesamiento en avance

```

PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel=100% PDAT2
WAIT FOR $IN[20]==TRUE
PTP P3 Vel=100% PDAT3

```

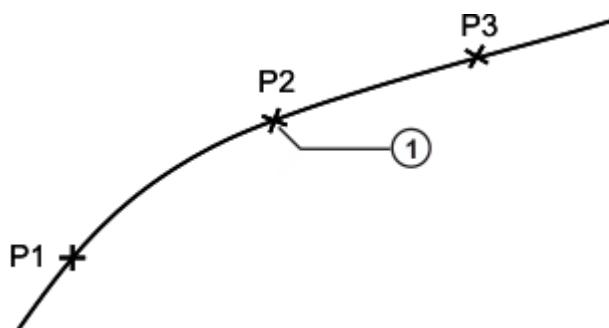


Fig. 12-10: Movimiento ejemplar para lógica

	Observación
1	El movimiento se interrumpe en el punto P2. Tras la parada exacta se verifica la entrada 20. Si el estado de la entrada es correcto se puede continuar directamente; de lo contrario, se espera hasta alcanzar el estado.

- WAIT FOR con ejecución en avance (aplicación de CONTINUE)

```

PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel=100% PDAT2
CONTINUE
WAIT FOR ($IN[10] OR $IN[20])
PTP P3 Vel=100% PDAT3

```

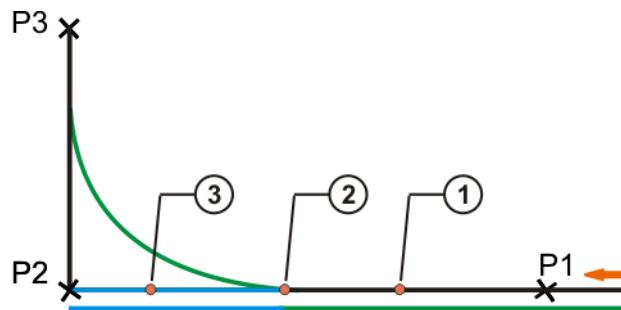


Fig. 12-11: Movimiento ejemplar para lógica con avance

	Acción
1	Las entradas 10 o 20 se encuentran o se encontraban ya en TRUE en el avance. Por tanto, se realiza una aproximación.
2	Condición cumplida poco antes, se realiza un movimiento de aproximación
3	Condición cumplida demasiado tarde. No se puede realizar un movimiento de aproximación, por lo que se debe desplazar al punto P2. En el punto P2, sin embargo, se puede continuar inmediatamente el desplazamiento si en ese momento se ha cumplido la condición. Si la condición no se ha cumplido, el robot espera en el punto P2 hasta que se cumpla. La indicación en la ventana de mensajes es "Espera a" (entrada 10 o entrada 20). En los modos de prueba (T1 o T2) se dispone del botón "Simular".

## 12.7 Ejercicio: Técnicas de bucles

**Objetivo del ejercicio** Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Programación de bucles en KRL

**Requisitos** Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos teóricos sobre la funcionalidad de las distintas técnicas de bucles para la programación

- Formulación de tarea**
1. Duplicar el programa **Procedimiento** con el nombre **Procedimiento\_bucle**.
  2. Sustituir el bucle *LOOP - ENDLOOP* por un bucle *FOR - WHILE*
  3. Para ello, declarar una variable de número entero como contador de bucle.
  4. Repetir el bucle 5 veces.
  5. Debe ser posible también salir con antelación de la ejecución. Usar para ello *\$IN[11]* como criterio de cancelación.

**Lo que se debe saber tras el ejercicio:**

1. En el distribuidor SWITCH/CASE existe la instrucción "DEFAULT". ¿Qué función tiene esta instrucción "DEFAULT"?
- .....

2. ¿Con qué instrucción se puede ajustar la anchura de paso en el bucle FOR?
- .....

3. ¿Qué bucles se pueden abandonar con la instrucción "EXIT"?
- .....

4. ¿Qué parte se puede omitir en una ramificación ? a. IF b. THEN c. ELSE d. ENDIF
- .....

5. ¿Qué es incorrecto en este ejemplo de programa?
- .....

```

IF $IN[14]==FALSE THEN
    $OUT[12]=TRUE
    GOTO MARKE1
ELSE
    $OUT[12]=FALSE
    GOTO MARKE2
ENDIF
WHILE $IN[17]==TRUE
    PTP P1
    MARKE1: ...
ENDWHILE
MARKE2: ...
PTP HOME

```



## 13 Trabajar con un control superior

### 13.1 Vista general

Esta unidad incluye los siguientes contenidos:

- Preparar el inicio del programa desde el PLC
- Adaptar el enlace PLC

### 13.2 Preparación para el inicio de programa del PLC

#### Robots en red

Si se van a controlar procesos de robots desde un puesto central (desde un ordenador principal o un PLC), se usa para ello la interfaz *Automático externo*.

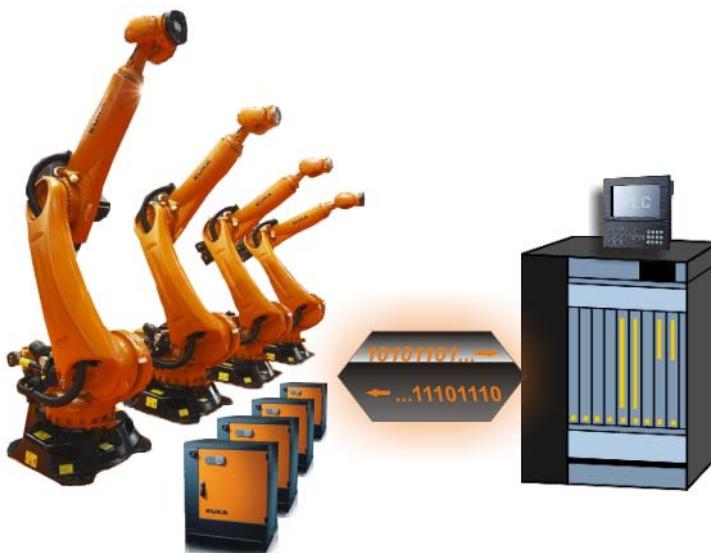


Fig. 13-1: Enlace PLC

#### Principio de la estructura del sistema

Mediante la interfaz *Automático externo* se controlan los procesos del robot a través de una unidad de control superior (p. ej., un PLC).

**Para la comunicación entre PLC y robot se necesita lo siguiente:**

- Un bus de campo físico existente y configurado entre robot y PLC, p. ej., PROFINET.
- Las señales para los procesos de robot se deben transmitir a través del bus de campo. Esto se consigue mediante entradas y salidas digitales configurables en la interfaz *Automático externo*.

■ **Señales de control al robot (entradas):**

A través de la interfaz Automático externo, la unidad de control superior transmite a la unidad de control del robot las señales para los procesos del mismo (p. ej. movimiento habilitado, confirmación de errores, inicio de programa, etc.).

■ **Estado del robot (salidas):**

La unidad de control del robot transmite a la unidad de control superior información sobre los estados de funcionamiento y averías.

- Un programa de organización **CELL.SRC** adaptado para seleccionar los programas del robot desde fuera.
- Selección del modo de servicio **Automático externo** en el que un ordenador central o un PLC asume el control del sistema de robot.

### Advertencias de seguridad en el inicio de programa externo

Después de seleccionar el programa CELL, se debe realizar un desplazamiento COI en modo de servicio T1 o T2.

#### **ADVERTENCIA**

Cuando el paso de movimiento seleccionado contiene la instrucción de avance PTP, se produce un desplazamiento COI a modo de movimiento PTP desde la posición real hasta la posición de destino. Si el paso de movimiento seleccionado contiene LIN o CIR, el desplazamiento COI se ejecuta como movimiento LIN. Observar el movimiento para evitar colisiones. En el desplazamiento COI, la velocidad se reduce automáticamente.

Cuando se ha ejecutado el desplazamiento COI una vez, en el inicio externo ya no ejecutará ningún desplazamiento COI más.

#### **ADVERTENCIA**

En el modo de servicio Automático externo no se efectúa ningún desplazamiento COI. Esto significa que, una vez arrancado, el robot se desplaza a la primera posición programada a la velocidad programada (no reducida). El robot no para allí.

### Procedimiento para el inicio de programa externo

#### Requisitos

- Modo de servicio T1 o T2
  - Las entradas/salidas para Automático externo y el programa CELL.SRC están configuradas
1. En el navegador, seleccionar el programa CELL.SRC. El programa CELL se encuentra siempre en el directorio KRC:\R1.
  2. Ajustar el override del programa al 100% (es el ajuste recomendado; en caso necesario, se puede ajustar otro valor).



Fig. 13-2: Selección de Cell y ajuste del override del programa

- 1 Ajuste POV
- 2 Selección de CELL.SRC
3. Realizar un desplazamiento COI:  
Pulsar y mantener el pulsador de validación. Mantener pulsada la tecla de arranque hasta que en la ventana de mensajes aparezca la leyenda "COI alcanzado".
4. Seleccionar el modo de servicio "Automático externo".
5. Iniciar el programa desde una unidad de control superior (PLC).

### 13.3 Adaptar el enlace PLC (Cell.src)

#### Programa de organización Cell.src

Para administrar los números de programas transmitidos por el PLC, se utiliza el programa organizativo Cell.src. Se encuentra siempre en la carpeta "R1". Como cualquier otro programa, el programa Cell puede adaptarse individualmente mientras que la estructura de base del programa debe mantenerse igual.

#### Estructura y funcionalidad del programa Cell

```

1 DEF CELL ( )
6 INIT
7 BASISTECHINI
8 CHECK HOME
9 PTP HOME Vel= 100 % DEFAULT
10 AUTOEXTINI
11 LOOP
12 P00 (#EXT PGNO,#PGNO_GET_DMY[],0 )
13 SWITCH PGNO ; Select with Programnumber
14
15 CASE 1
16 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
17 ;EXAMPLE1( ) ; Call User-Program
18
19 CASE 2
20 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
21 ;EXAMPLE2( ) ; Call User-Program
22
23 CASE 3
24 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
25 ;EXAMPLE3( ) ; Call User-Program
26
27 DEFAULT
28 P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
29 ENDSWITCH
30 ENDOOP
31 END

```

Fig. 13-3: Programa CELL

1	Inicialización y posición inicial <ul style="list-style-type: none"> <li>■ Inicialización de los parámetros básicos</li> <li>■ Comprobación de la posición del robot respecto a la posición inicial</li> <li>■ Inicialización de la interfaz "Automático externo"</li> </ul>
2	Bucle infinito: <ul style="list-style-type: none"> <li>■ Consulta del número de programa a través del módulo "P00"</li> <li>■ Acceso al bucle de selección con el número de programa facilitado.</li> </ul>
3	Bucle de selección con número del programa <ul style="list-style-type: none"> <li>■ El salto en la ramificación correspondiente ("CASE") se realiza conforme al número de programa (determinado en la variable "PGNO").</li> <li>■ Se ejecuta el programa de robot especificado en la ramificación.</li> <li>■ Los números de programa no válidos provocan que se salte a la ramificación "Por defecto".</li> <li>■ Una vez finalizada la ejecución, se repite el bucle.</li> </ul>

**Procedimiento**

1. Cambiar al grupo de usuario "Experto".
2. Abrir CELL.SRC
3. Reemplazar en la sección "CASE" la denominación "EXAMPLE" por el nombre del programa que se debe activar con el correspondiente número de programa. Eliminar el punto y coma de delante del nombre.

```
LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber

CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
main()

CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
body_38()
body_515()

DEFAULT
P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP
```

Fig. 13-4: Ejemplo de un programa Cell adaptado

4. Cerrar el programa y guardar las modificaciones.

## 13.4 Preguntas: Trabajos en una unidad de control superior

### Lo que se debe saber tras el ejercicio:

1. ¿Cuál es el requisito previo para la comunicación con un PLC?

.....  
.....  
.....

2. ¿Mediante qué subprograma global se transmite el número de programa desde el PLC?

.....  
.....  
.....

3. ¿Qué es lo que se puede programar en los "CASE" del bucle SWITCH CASE del CELL.SRC? ¿Qué es lo que no se debe/puede programar?

.....  
.....



## 14 Anexo

### 14.1 Configuración y aplicación del modo automático externo

#### Trabajar con un control superior

- Véase: ([>>> 13.2 "Preparación para el inicio de programa del PLC"](#) Página 235)

**Utilización de entradas/salidas de la interfaz en modo Automático externo**

Cuadro resumen de las señales más importantes de la interfaz

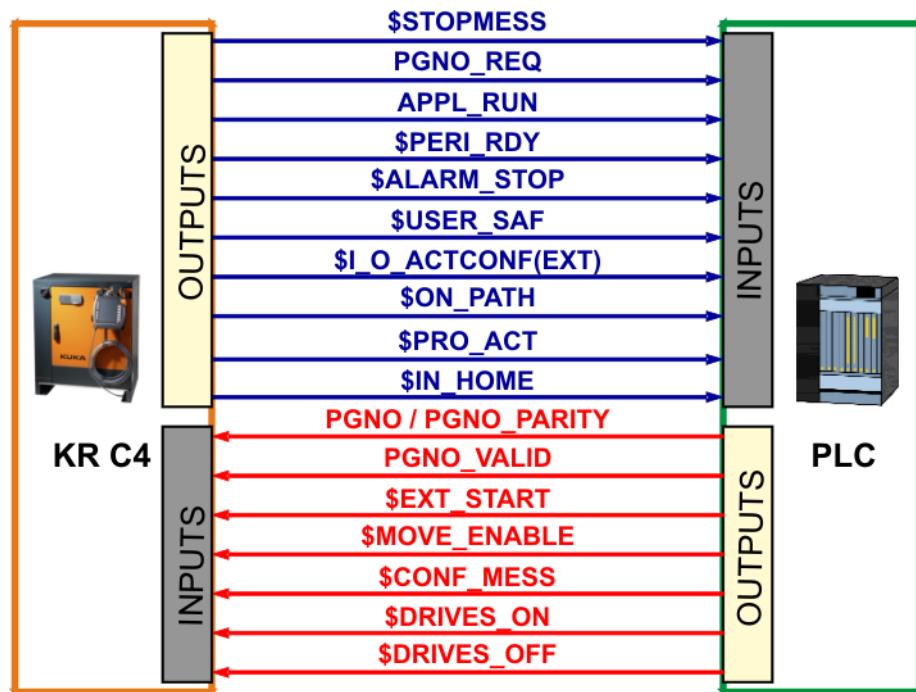


Fig. 14-1: Resumen de las señales más importantes en modo automático externo

Entradas (desde la perspectiva de la unidad de control del robot)

- **PGNO\_TYPE:** tipo de números del programa

Esta variable define en qué formato se leerán los números de programa transmitidos por la unidad de control superior.

Valor	Descripción	Ejemplo
1	Ler como número binario. El número de programa se transmite por la unidad de control superior en forma de valor entero con codificación binaria.	0 0 1 0 0 1 1 1 => PGNO = 39
2	Ler como valor BCD. El número de programa se transmite por la unidad de control superior en forma de valor decimal con codificación binaria.	0 0 1 0 0 1 1 1 => PGNO = 27
3	Ler como "1 de N". El número de programa es transmitido por la unidad de control superior o por los dispositivos periféricos en forma de valor codificado "1 de n".	0 0 0 0 0 0 0 1 => PGNO = 1 0 0 0 1 0 0 0 => PGNO = 4

\* En este formato de transferencia, los valores de PGNO\_REQ, PGNO\_PARITY y PGNO\_VALID no son evaluados, y carecen por tanto de importancia.

■ **PGNO\_LENGTH:** longitud de números del programa

Esta variable establece el ancho de bit del número de programa transmitido por la unidad de control superior. Rango de valores: 1 ... 16.

Si PGNO\_TYPE posee el valor 2, sólo se permiten anchos de 4, 8, 12 y 16 bits.

■ **PGNO\_PARITY:** bit de paridad de número del programa

Entrada por la que la unidad de control superior transmite el bit de paridad.

Entrada	Función
Valor negativo	Paridad impar
0	Sin evaluación!
Valor positivo	Paridad par

Si PGNO\_TYPE tiene valor 3, PGNO\_PARITY no se evaluará..

■ **PGNO\_VALID:** número de programa válido

Entrada por la que la unidad de control superior transmite el comando de lectura del número de programa.

Entrada	Función
Valor negativo	Se acepta el número con el flanco decreciente de la señal.
0	Se acepta el número con el flanco creciente de la señal en el conductor EXT_START.
Valor positivo	Se acepta el número con el flanco creciente de la señal.

■ **\$EXT\_START:** arranque externo

Al establecer esta entrada, se puede iniciar o continuar un programa (normalmente CELL.SRC) si la interfaz de E/S se encuentra activa.



Sólo se evalúa el flanco creciente de la señal.



**ADVERTENCIA** En el modo de servicio Automático externo no se efectúa ningún desplazamiento COI. Esto significa que, una vez arrancado, el robot se desplaza a la primera posición programada a la velocidad programada (no reducida). El robot no para allí.

■ **\$MOVE\_ENABLE:** marcha habilitada

Esta entrada se utiliza para controlar los accionamientos del robot a través de la unidad de control superior.

Señal	Función
TRUE	Desplazamiento manual posible en los modos de servicio T1 y T2. Ejecución de programa posible.
FALSE	Detención de todos los accionamientos y bloqueo de todos los comandos activos



Si los accionamientos han sido detenidos por la unidad de control superior, aparecerá el mensaje "LIBERACIÓN DE MOVIMIENTO GENERAL". El robot sólo se podrá mover una vez que se haya borrado este mensaje y se haya recibido una nueva señal de arranque externa.



Durante la puesta en servicio, a la variable "\$MOVE\_ENABLE" se le asigna con frecuencia el valor "\$IN[1025]". Si después se olvida configurar otra entrada, no será posible efectuar un arranque externo.

■ **\$CONF\_MESS:** confirmación de mensaje

Al activar esta entrada, la unidad de control superior confirmará los mensajes de error por sí misma en cuanto se haya subsanado la causa del fallo.



Sólo se evalúa el flanco creciente de la señal.

■ **\$DRIVES\_ON:** accionamientos conectados

Si en esta entrada se crea un impulso de nivel high de 20 ms de duración como mínimo, la unidad de control principal activa los accionamientos del robot.

■ **\$DRIVES\_OFF:** accionamientos desconectados

Si en esta entrada se crea un impulso de nivel low de 20 ms de duración como mínimo, la unidad de control principal desactiva los accionamientos del robot.

Salidas (desde la perspectiva de la unidad de control del robot)

■ **\$ALARM\_STOP:** parada de emergencia

Esta salida se reinicia en las siguientes situaciones de PARADA DE EMERGENCIA:

- Se va a accionar el pulsador de PARADA DE EMERGENCIA del KCP (parada de emergencia int.)
- PARADA DE EMERGENCIA externa



En caso de una PARADA DE EMERGENCIA, en los estados de las salidas **\$ALARM\_STOP** e **parada de emergencia int.** se detecta de qué PARADA DE EMERGENCIA se trata:

- Ambas salidas son FALSE: la PARADA DE EMERGENCIA se ha desencadenado en el KCP.
- **\$ALARM\_STOP** es FALSE y **parada de emergencia int.** es TRUE: PARADA DE EMERGENCIA externa

■ **\$USER\_SAF:** protección del operario/puertas de protección

Esta salida se reinicializa al abrir un conmutador de muestreo de la valla protectora (en el modo de servicio AUT) o bien al soltar un pulsador de validación (en el modo de servicio T1 o T2).

- **\$PERI\_RDY:** los accionamientos están listos  
Al establecer esta salida, la unidad de control del robot comunica a la unidad de control superior que los accionamientos del robot están activados.
- **\$STOPMESS:** mensajes de parada  
Esta salida es establecida por la unidad de control del robot para indicar a la unidad de control superior que se ha producido un mensaje que requiere la parada del robot. (Ejemplos: PARADA-EMERGENCIA, liberación de la marcha o protección del operador).
- **\$I\_O\_ACTCONF:** el modo Automático externo está activado  
Esta salida es TRUE si se ha seleccionado el modo *Automático externo* y la entrada \$I\_O\_ACT es TRUE (normalmente siempre en \$IN[1025]).
- **\$PRO\_ACT:** el programa está activo/en ejecución  
Esta salida estará fijada cuando esté activo un proceso en el nivel de robot. El proceso continúa activo mientras se esté procesando el programa o una interrupción. Al final del programa, la ejecución del mismo pasará a estado inactivo justo cuando se hayan procesado todas las salidas de impulsos y de activación (Trigger).
- **PGNO\_REQ:** solicitud de números de programa  
Con un cambio de señal en esta salida se requiere a la unidad de control superior que transmita un número de programa.  
Si PGNO\_TYPE tiene valor 3, PGNO\_REQ no se evaluará.
- **APPL\_RUN:** el programa de la aplicación está en ejecución  
Al establecer esta salida, la unidad de control del robot comunica a la unidad de control superior que se está procesando un programa en ese preciso instante.
- **\$IN\_HOME:** el robot se encuentra en la posición inicial  
Esta salida comunica a la unidad de control superior si el robot se encuentra en su posición inicial.
- **\$ON\_PATH:** el robot se encuentra sobre la trayectoria  
Esta salida estará activa mientras el robot se encuentra dentro de su trayectoria programada. Después de un desplazamiento COI, se activa la salida ON\_PATH. Esta salida permanecerá activa hasta que el robot se salga de su trayectoria, se resetee el programa o se ejecute una selección de paso. La señal ON\_PATH no dispone de una ventana de tolerancia; en el momento en que el robot abandona la trayectoria, se reinicia la señal.

**Principio de la comunicación en el modo Automático externo**

Resumen de toda la secuencia

Signalname	Signalrichtung	Automatischer Anlagenanlauf mit Normalbetrieb	Voraussetzungen:	SPROG_DEVRKS8PSU Programm-Nr. gültig → PGNO_VALID
APPL_RUN	KRC → SPS	.	.	.
IR1/EXAMPLE.SRC läuft	.	.	.	.
IR1CELLSRC läuft	.	.	.	.
PGNO_REQ	KRC → SPS	.	.	.
PGNO/PGNO_PARITY	SPS → KRC	.	.	.
PGNO_VALID	SPS → KRC	.	.	.
SEXT_START	SPS → KRC	.	.	.
SPROG_ACT	KRC → SPS	.	.	.
\$STOPMESS	KRC → SPS	.	.	.
\$CONF_MESS	SPS → KRC	.	.	.
\$LDO_ACTCONF(EXT)	KRC → SPS	.	.	.
\$PERI_RDY	KRC → SPS	.	.	.
\$DRIVES_ON	SPS → KRC	.	.	.
\$DRIVES_OFF	SPS → KRC	.	.	.
\$ALARM_STOP	KRC → SPS	.	.	.
\$MOVE_ENABLE	SPS → KRC	.	.	.
\$USER_SAF	KRC → SPS	.	.	.
\$ON_PATH	KRC → SPS	.	.	.
\$IN_HOME	KRC → SPS	.	.	.

**Fig. 14-2: Arranque automático de la instalación y servicio normal con confirmación del número de programa por medio de PGNO\_VALID**

División en subcampos

1. Conectar los accionamientos
2. Confirmar mensajes
3. Iniciar el programa Cell
4. Transmitir el número de programa y ejecutar la aplicación

Cada uno de estos campos está ejecutado como protocolo de enlace. Hay condiciones que se deben cumplir; el PLC envía señales y la unidad de control del robot comunica los correspondientes estados del robot al PLC.



**Fig. 14-3: Protocolo de enlace**

Es conveniente utilizar este protocolo de enlace predefinido.

**Conectar los accionamientos**

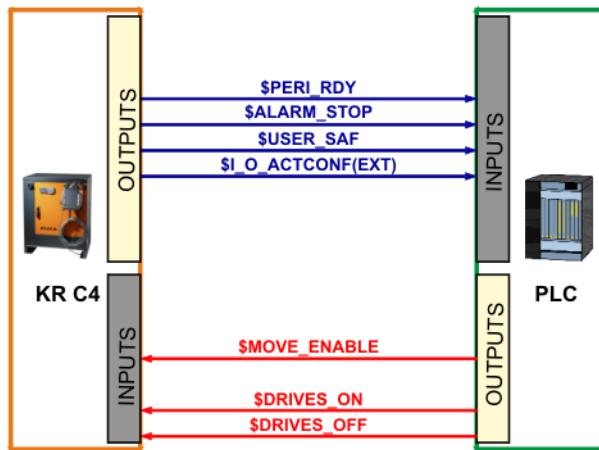


Fig. 14-6

- Requisitos
  - \$USER\_SAF: puertas de protección cerradas.
  - \$ALARM\_STOP: no hay activa ninguna parada de emergencia.
  - \$I\_O\_ACTCONF: está activo el modo Automático externo
  - \$MOVE\_ENABLE: marcha habilitada disponible
  - \$DRIVES\_OFF: no está activada la desconexión de los accionamientos
- Conectar los accionamientos  
\$DRIVES\_ON: Conectar la señal de los accionamientos, como mínimo, durante 20 ms.
- Accionamientos preparados  
\$PERI\_RDY: en cuanto se reciba la confirmación para los accionamientos, se anula la señal \$DRIVES\_ON.

#### Confirmar mensajes

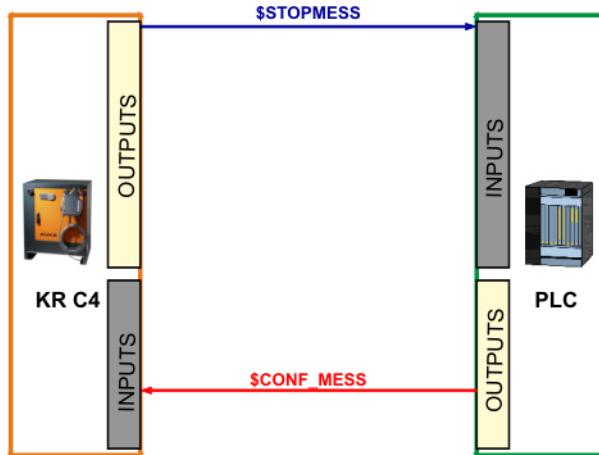


Fig. 14-10

- Requisitos  
\$STOPMESS: hay activo un mensaje de Stop.
- Confirmar el mensaje  
\$CONF\_MESS: confirmar el mensaje.
- Se han borrado los mensajes que se pueden confirmar.  
\$STOPMESS: el mensaje de Stop ya no está en \$CONF\_MESS, ahora se puede anular.

**Iniciar el programa (CELL.SRC) externamente.**

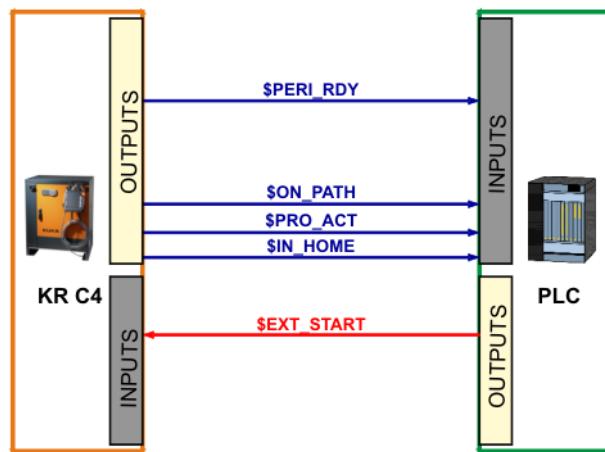


Fig. 14-15

- Requisitos
  - \$PERI\_RDY: los accionamientos están listos.
  - \$IN\_HOME: el robot se encuentra en la posición HOME.
  - Sin \$STOPMESS: no hay activo ningún mensaje de Stop.
- Arranque externo
  - \$EXT\_START: activar el arranque externo (flanco positivo).
- El programa CELL está en ejecución.
  - \$PRO\_ACT: indica que el programa CELL está en ejecución.
  - \$ON\_PATH: en cuanto el robot esté en su trayectoria programada, se anula la señal \$EXT\_START.

**Ejecutar la transferencia del número de programa y el programa de aplicación**

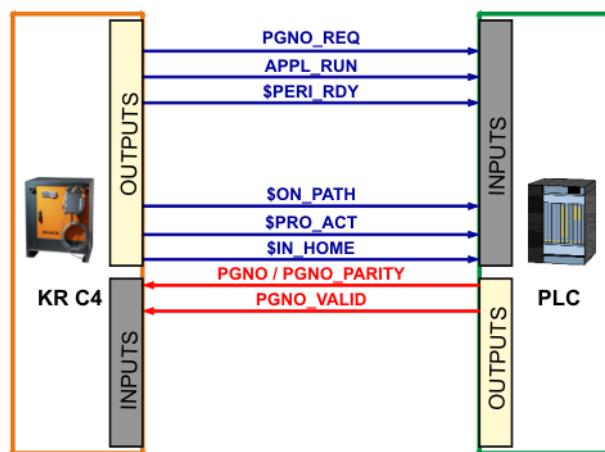


Fig. 14-22

- Requisitos
  - \$PERI\_RDY: los accionamientos están listos.
  - \$PRO\_ACT: el programa CELL está en ejecución.
  - \$ON\_PATH: el robot se encuentra sobre la trayectoria.
  - \$IN\_HOME: el robot se encuentra en la posición inicial; esto no es necesario en caso de reanudación.
  - PGNO\_REQ: la solicitud de números de programa está activada.
- Transmisión de números de programa y confirmación
  - Transmisión del número de programa

(se han ajustado el tipo de datos correcto (PGNO\_TYPE), la longitud de número de programa (PGNO\_LENGTH) y el primer bit del número de programa (PGNO\_FBIT)).

- PGNO\_VALID: activar el número de programa válido (confirmación, flanco positivo)
- El programa de aplicación está en ejecución.
  - APPL\_RUN: indica que el programa de aplicación está en ejecución.
  - El robot abandona la posición HOME. Una vez finalizado el programa de aplicación, el robot regresa a la posición HOME.

### Procedimiento

1. En el menú principal seleccionar **Configuración > Entradas/Salidas > Automático externo**.
2. En la columna **Valor**, marcar la celda que se debe editar y pulsar **Editar**.
3. Introducir el valor que se desee y guardarlo con **OK**.
4. Repetir los pasos 2 y 3 para todos los valores que se desean editar.
5. Cerrar la ventana. Las modificaciones son aceptadas.

	Denominación	Tipo	Nombre	Valor
1	Tipo de nº de programa	Verde	PGNO_TYPE	1
2	Número de programa reflejado	Verde	REFLECT_PROG_I_0	
3	Ancho de bit del nº de programa	Verde	PGNO_LENGTH	8
4	Primer bit de nº de programa	Verde	PGNO_FBIT	33
5	Bit de paridad	Verde	PGNO_PARITY	41
6	Nº de programa válido.	Verde	PGNO_VALID	42
7	Inicio de programa	Verde	\$EXT_START	1026
8	Movimiento habilitado	Verde	\$MOVE_ENABLE	1025
9	Confirmación de error	Verde	\$CONF_MESS	1026
10	Accionamientos desconectados (invertido)	Verde	\$DRIVES_OFF	1025
11	Accionamientos ON	Verde	\$DRIVES_ON	140
12	Activar interfaz	Verde	\$I_O_ACT	1025

Fig. 14-25: Configuración de entradas en el modo Automático Externo

Pos.	Descripción
1	Número
2	Nombre largo de la entrada/salida
3	Tipo <ul style="list-style-type: none"> <li>■ <b>Verde:</b> Entrada/Salida</li> <li>■ <b>Amarillo:</b> Variable o variable del sistema (\$...)</li> </ul>
4	Nombre de la señal o de la variable
5	Número de la entrada/salida o número del canal
6	Las salidas se encuentran clasificadas en pestañas por temas.

Configuración de automático externo: Salidas				
	Denominación	Tipo	Nombre	Valor
1	Control preparado	HO	\$RC_RDY1	137
2	Círculo de parada de emergencia cerrado	HO	\$ALARM_STOP	1013
3	Protección del operario cerrada	HO	\$USER_SAF	1011
4	Accionamientos preparados	HO	\$PERI_RDY	1012
5	Robot ajustado	HO	\$ROB_CAL	1001
6	Interfaz activa	HO	\$I_O_ACTCONF	140
7	Avería general	HO	\$STOPMESS	1010
8	Primer bit para la reflexión imagen del programa	HO	PGNO_FBIT_REFI	999
9	Parada de emergencia interna	HO	Parada de emergencia interna	1002

Conditions of start    Program state    Position of the robot    Mode of operation

Fig. 14-26: Configuración de salidas en el modo Automático Externo

## 14.2 Ejercicio: El modo Automático externo

### Objetivo del ejercicio

Después de completar correctamente este ejercicio, se dispondrá de la competencia necesaria para efectuar las siguientes tareas:

- Integración selectiva de un programa de robot en el modo de servicio Automático externo.
- Adaptación del programa "Cell".
- Configuración de la interfaz para el modo Automático externo.
- Conocer la secuencia del servicio en modo Automático externo.

### Requisitos

Los siguientes requisitos son necesarios para completar este ejercicio correctamente:

- Conocimientos sobre el procesamiento del programa "Cell".
- Conocimientos sobre la configuración de la interfaz para el modo Automático externo.
- Conocimientos teóricos sobre la secuencia técnica de señales para el modo Automático externo.

### Formulación de tarea

1. Configurar la interfaz para el modo Automático externo según las especificaciones del panel de mando.
2. Ampliar el programa Cell con los 3 módulos que se deseen después de haber comprobado su función.
3. Comprobar el programa en los modos de servicio T1, T2 y Automático. Se deben tener en cuenta las prescripciones de seguridad enseñadas.
4. Simular la funcionalidad de la unidad de control del PLC mediante el pulsador.

### Lo que se debe saber tras el ejercicio:

1. ¿Qué requisito se necesita para que no se evalúe PGNO\_REQ?

.....  
.....

2. ¿Con qué señal se conectan los accionamientos y a qué se debe prestar atención en este caso?

.....  
.....

3. ¿Qué variable de la interfaz para el modo Automático externo repercute también en el desplazamiento manual?

.....  
.....

4. ¿Qué pliegue comprueba la posición inicial en el programa CELL?

.....  
.....

5. ¿Qué requisitos se necesitan para el modo Automático externo?

.....  
.....

## 14.3 Abreviaturas

Para facilitar el trabajo con la unidad de control del KR C4, a continuación se listan las abreviaturas más importantes acompañadas por una breve explicación.

Término	Descripción
CCU	<b>Cabinet Control Unit</b>
CCUsr	<b>Cabinet Control Unit</b>  small robot
CIB	<b>Cabinet Interface Board</b>
CIBsr	<b>Cabinet Interface Board</b>  small robot
CK	<b>Customer Built Kinematics</b>
CSP	<b>Controller System Panel</b>
Tarjeta Dual NIC	Tarjeta de red doble
EDS	<b>Electronic Data Storage</b> (tarjeta de memoria)
EMD	<b>Electronic Mastering Device</b> (antes EMT) para el ajuste de robot
CEM	Compatibilidad electromagnética.
GBE	<b>Giga Bit EtherNet</b>
Catálogo	Puede contener diferentes elementos, p. ej.: modelos, componentes, cinemáticas.
KCB	<b>KUKA Controller Bus</b>
KEB	<b>KUKA Extension Bus</b>
KCP	<b>KUKA Control Panel</b> (unidad manual de programación), nuevo término: smartPAD
KLI	<b>KUKA Line Interface</b>
KOI	<b>KUKA Operator Panel Interface</b>
KPC	<b>PC</b> de control de KUKA
KPP	<b>KUKA Power Pack</b>
KPPsr	<b>KUKA Power Pack</b>  small robot
KRL	<b>KUKA Robot Language</b> (Lenguaje de programación de KUKA Roboter)
KSB	<b>KUKA System Bus</b>
KSP	<b>KUKA Power Pack</b>
KSPsr	<b>KUKA Power Pack</b>  small robot
KSI	<b>KUKA Service Interface</b>
KUKA.HMI	<b>Human Machine Interface</b>  KUKA.HMI es la interfaz de usuario de KUKA.
LWL	Conductor de fibra óptica ( <b>Licht-Wellen-Leiter</b> )
MCFB	<b>Motion Control Function Block</b>  Módulo para la programación de tareas de movimiento. Estos módulos son compatibles con PLC y específicos de KUKA.
MGU	<b>Motor Gear Unit</b>  Combinación motor-reductor KUKA para cinemáticas

Término	Descripción
OPI	<b>Operator Panel Interface</b> (conexión smartPAD)
PMB	<b>Power Management Board</b>
RCD	<b>Residual Current Device</b> ; interruptor diferencial (residual)
RDC	<b>Resolver Digital Converter</b>
SATA	<b>Serial Advanced Technology Attachment</b> (bus de datos entre el procesador y el disco duro)
SIB	<b>Safety Interface Board</b>
SBC	<b>Single Brake Control</b>
STO	<b>Safe Torque Off</b>
SION	<b>Safety Input Output Node</b>
USB	<b>Universal Serial Bus</b> (sistema bus para conectar un ordenador con dispositivos adicionales)
USV	Sistema de alimentación ininterrumpida ( <b>Unterbrechungsfreie Stromversorgung</b> )
Editor de catálogo WorkVisual	Software para la creación de elementos de catálogo para WorkVisual

# Índice

## Símbolos

\$LDC\_LOADED 59

## A

Abreviaturas 251

Acoplar 197

Activar la detección de colisión en el formulario inline 147

Adaptar el offset para movimientos 147

Administrador 193

Anexo 241

Aproximación 124, 134

Aproximación CIRC 131

Aproximación LIN 131

Aproximación PTP 120

Archivar 111

Automático externo 235, 241

Avance 205

## B

Bucle de conteo 219

Bucle sinfín 217

Bucle, finito 221

Bucle, infinito 222

Bucles 217

## C

Cargas sobre el robot 58

Categoría de parada 0 26

Categoría de parada 1 26

Categoría de parada 2 26

Catálogo 251

CCU 251

CCUsr 251

CEM 251

CIB 251

CIBsr 251

CK 251

COI 99

Comentarios 196

Configuración de garras 171

Configurar Automático externo, ejercicio 250

Consultas 223

Control de ejecución de programa 217

Control de la orientación 129, 134

CSP 251

## D

Datos de carga adicional (opción de menú) 62

Datos de carga de la herramienta 58

Datos de carga de la herramienta (opción de menú) 59

DECL 204, 206

Declaraciones 203

Declaración 204, 207

Detección de colisión 134, 143, 147

Detección de colisión (opción de menú) 147

Detección de colisión, Automático externo 146

Detección de colisión, variable 145

Dispositivo de liberación 33

Distribuidores (SWITCH- CASE) 225

## E

Editor de catálogo WorkVisual 252

EDS 251

Ejercicio, ajuste de carga con Offset 56

Ejercicio, Ajuste de robot 56

Ejercicio, desplazamiento de trayectoria 140

Ejercicio, desplazamiento manual herramienta 66

Ejercicio, ejecutar programas del robot 107

Ejercicio, funciones de conmutación 165

Ejercicio, lógica 165

Ejercicio, Medición de la base mesa 94

Ejercicio, Medición de la herramienta clavija 77

Ejercicio, Medición de la herramienta garra 80

Ejercicio, medición numérica de la herramienta 80

Ejercicio, medir herramienta externa 186

Ejercicio, operación y desplazamiento manual en el sistema de coordenadas universales 43

Ejercicio, Operación y desplazamiento manual específico del eje 36

Ejercicio, posicionamiento aproximado 140

Ejercicio, procesos manuales con herramienta fija 181

Ejercicio, Programa en el aire 125

Ejercicio, programación de la garra, clavija 176

Ejercicio, programación de la garra, placa 174

Ejercicio, programación de movimientos con TCP externo 192

EMD 251

## F

FOLD 198

Formularios inline 118

Funciones estándar 210

Función de conmutación, simple 154

Función de conmutación, trayectoria 157

Función de espera 151

Función de espera, dependiente de una señal 230

Función de espera, dependiente del tiempo 229

## G

GBE 251

Gestor de conexiones 14

global 205

GOTO 227

Grupo de usuario, por defecto 193

## H

Herramienta fija, procesos manuales 179

## I

Incremento 33

- Individual (opción de menú) 213  
Inicialización 99, 208  
Inicio de programa 100  
Inicio de programa del PLC 235  
Instrucción de salto 227  
Introducción al nivel del experto 193
- K**  
KCB 251  
KCP 251  
KEB 251  
KLI 251  
KOI 251  
KPC 251  
KPP 251  
KRL 251  
KSB 251  
KSI 251  
KSP 251  
KUKA.GripperTech 167  
KUKA.HMI 251
- L**  
Listado LOG 113  
local 205  
LWL 251  
Lógica, general 149
- M**  
Manejo de la garra 167  
Manipulación 210  
Marcas 196  
MCFB 251  
Medición de base 89  
Medición, herramienta fija 182  
Medición, pieza guiada por robot 184  
Medir herramienta 67  
Mensaje de acuse de recibo 21  
Mensaje de diálogo 22  
Mensaje de espera 22  
Mensaje de estado 21  
Mensaje de observación 22  
Mensajes 21  
MGU 251  
Modificación, instrucciones de movimiento 135  
Modificar variable 213  
Modo de interpolación 123, 134, 147  
Modo de puesta en servicio 45  
Modo de servicio 23  
momento de fuerza 144  
momento de impacto 144  
Mover ejes del robot individualmente 30  
Movimiento CIRC 127  
Movimiento LIN 127  
Movimiento PTP 119
- O**  
Operaciones de bit 210  
Operaciones de comparación 210  
Operaciones lógicas 210  
Operario 193
- OPI 252
- P**  
Palabra clave 205  
PARADA DE EMERGENCIA 14  
Parada de seguridad STOP 0 25  
Parada de seguridad STOP 1 25  
Parada de seguridad STOP 2 25  
Parada de seguridad 0 25  
Parada de seguridad 1 25  
Parada de seguridad 2 25  
Parada de seguridad, externa 18, 19  
Parada de servicio segura 25  
PMB 252  
Posición de pánico 31  
Prioridad 212  
Procesos manuales específicos del eje 30  
Procesos manuales, base 84  
Procesos manuales, herramienta 62  
Procesos manuales, herramienta fija 179  
Procesos manuales, universales 38  
Programa Borrar 111  
Programa duplicar 110  
Programa renombrar 111  
Programación de garra 168  
Programación del movimiento 117  
Programación, TCP externo 191  
Programador 193  
Programar una llamada a subprograma 201  
Protección del operario 18  
Puerta de protección 18  
Pulsador de PARADA DE EMERGENCIA 17  
Punto de la raíz de la muñeca 127, 128
- R**  
Ramificación 223  
RCD 252  
RDW 252  
Restaurar 111  
Robroot 37
- S**  
Salto 227  
SATA 252  
SBC 252  
Seguridad del robot 16  
Selección de programa 100  
SIB 252  
Singularidad 127  
SION 252  
Sistema de coordenadas 37  
Sistema de coordenadas Base 37  
Sistema de coordenadas Flange 37  
Sistema de coordenadas Tool 37  
Sistema de coordenadas World 37  
Space Mouse 14  
STO 252  
STOP 0 26  
STOP 1 26  
STOP 2 26  
Subprograma global 199

Subprograma local 199  
Subprogramas 198

**T**

Tarjeta Dual NIC 251  
Tecla de arranque 15  
Tecla de arranque hacia atrás 15  
Tecla del teclado 15  
Tecla STOP 15  
Teclado 15  
Teclas de desplazamiento 14  
Teclas de estado 15  
Tipos de cálculo básicos 210  
tm\_useraction 143  
TMx 145

**U**

USB 252  
USV 252

**V**

Variable, mostrar individual 213  
Variables 203, 206  
Vida útil 205

**W**

WAIT 151  
WAIT FOR 151  
WHILE 221



