

KUKA



Training

Programación de robots 1

KUKA System Software 8.6, 8.7

Cuaderno de ejercicios y de trabajo



Documentación para la formación

Edición: 26.07.2024

PROG P1 KSS 8 V4.1.1

KUKA Deutschland GmbH

© Copyright 2024

KUKA Deutschland 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 Deutschland 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.

KIM-PS5-DOC

Traducción de la documentación original

Publicación: Pub COLLEGE - PROG P1 KSS 8 (Presentation) (V4.1.1) (PDF-COL) es
PB20173

Estructura de libro: PROG P1 KSS 8 - Presentation (R4) V1.53
BS17270

Versión: PROG P1 KSS 8 V4.1.1

Índice

1	Introducción.....	11
1.1	Bienvenido.....	11
1.2	Introducción a la robótica.....	12
2	Seguridad en la célula de robot.....	13
2.1	Navegador del curso.....	13
2.2	Descripción de la situación.....	13
2.3	Seguridad del robot en el ejemplo de la celda de formación College.....	14
2.4	Ejercicio: Seguridad en una célula de robot.....	16
2.5	Preguntas: Seguridad en una célula de robot.....	18
2.6	Indicaciones generales de seguridad.....	20
3	Documentación del sistema de robot.....	25
3.1	Navegador del curso.....	25
3.2	KUKA Xpert.....	25
3.3	Registro.....	27
3.4	Utilizar KUKA Xpert Basic.....	29
3.5	Ejercicio: Buscar documentación sobre Xpert-Basic.....	31
4	Manejo de programas: Probar programas con velocidad reducida manualmente.....	33
4.1	Navegador del curso.....	33
4.2	Descripción de la situación.....	34
4.3	Seleccionar programas de robot.....	35
4.3.1	Seleccionar programas de robot.....	35
4.4	Mensajes en el smartPAD	36
4.5	Conocer los modos de flujo de programa.....	38
4.6	Iniciar el programa.....	39
4.7	Conocer el desplazamiento de inicialización.....	39
4.8	¿Cómo se prueban los programas?.....	41
4.9	Ejercicio: Ejecutar programas del robot.....	42
5	Manejo de programas: Probar programas con velocidad de proceso.....	43
5.1	Navegador del curso.....	43
5.2	Descripción de la situación.....	43
5.3	Probar un programa con velocidad alta.....	44
5.4	Conocer los modos de servicio de la unidad de control del robot.....	46
5.5	Cambiar el modo de servicio.....	47
5.6	Ejercicio: Ejecutar programas del robot.....	48
5.7	Preguntas: Ejecutar los programas del robot.....	50
6	Eliminar un fallo en la instalación: Reaccionar ante una estación de descarga vacía.....	53
6.1	Navegador del curso.....	53
6.2	Descripción de la situación.....	54
6.3	¿Qué pasa si un cargador se vacía?.....	55
6.4	Manejo de la garra con KUKA.GripperTech.....	57

6.5	Conocer las teclas de ejecución del programa "Inicio hacia delante" e "Inicio hacia atrás".....	59
6.6	¿Cómo se puede reaccionar ante un cargador vacío?.....	59
6.7	Otra posibilidad para responder ante un cargador vacío.....	61
6.8	Conocer las teclas de ejecución del programa "Inicio hacia delante" e "Inicio hacia atrás".....	64
6.9	Ejercicio: Reaccionar ante el fallo de un cargador vacío.....	66
6.10	Preguntas: Reaccionar ante el fallo de un cargador vacío.....	67
7	Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente.....	69
7.1	Navegador del curso.....	69
7.2	Reaccionar ante un encargo de proceso deficiente.....	69
7.3	¿Qué sucede si el robot se detiene durante una aplicación de trayectoria?....	70
7.4	¿Cómo se puede reaccionar ante un encargo de proceso erróneo?.....	71
7.5	Ejercicio: Reaccionar ante un encargo de proceso deficiente.....	75
7.6	Movimiento del robot en el sistema de coordenadas universales.....	77
7.7	Ejercicio: Liberar el robot con el ratón 6D.....	79
8	Guardar y restaurar archivos del robot.....	81
8.1	Navegador del curso.....	81
8.2	Descripción de la situación.....	81
8.3	¿Por qué es necesario guardar los datos?.....	82
8.4	¿Cómo se puede crear una imagen y archivar datos?.....	82
8.5	Ejercicio: Crear una copia de seguridad de datos mediante archivo del robot..	85
9	Corregir las posiciones del programa en el sistema de coordenadas base.....	87
9.1	Navegador del curso.....	87
9.2	Descripción de la situación.....	87
9.3	¿Qué es necesario para una trayectoria de pegamento perfecta?.....	89
9.4	Conocer los modos de flujo de programa.....	89
9.5	SPTP – Desarrollos del movimiento con optimización del tiempo de ciclo (movimiento de eje).....	90
9.6	Crear movimientos de trayectoria.....	92
9.7	El sistema de coordenadas Base.....	93
9.8	Conocer el desplazamiento manual incremental.....	94
9.8.1	Desplazamiento incremental del robot.....	95
9.9	Mover el robot mediante el tipo de desplazamiento "Alinear" (referido a la base).....	97
9.10	Corregir las coordenadas de puntos de movimiento.....	99
9.11	Ejercicio: Reaccionar ante un encargo de proceso deficiente.....	104
9.12	Preguntas: Reaccionar ante un encargo de proceso deficiente.....	105
10	Corregir las posiciones de la garra en el sistema de coordenadas Tool.....	107
10.1	Navegador del curso.....	107
10.2	Descripción de la situación.....	107
10.3	Agarre de la herramienta de clavija central.....	108
10.4	¿Cómo puede modificarse la posición de la garra?.....	109
10.5	Mover el robot en el sistema de coordenadas TOOL.....	111

10.5.1	Mover el robot en el sistema de coordenadas de TOOL.....	112
10.6	Ejercicio: Modificar la posición de la garra en el cargador de clavijas.....	115
11	Configuración de una nueva estación de robot: Ajuste del robot...	117
11.1	Navegador del curso.....	117
11.2	Conocer el desplazamiento manual específico del eje.....	117
11.2.1	Desplazamiento del robot por medio de ejes específicos.....	119
11.3	Ajuste del robot.....	119
11.4	Principio de ajuste.....	120
11.4.1	Medio de ajuste.....	125
11.4.2	Determinación de la posición mecánica cero.....	127
11.4.3	Ajuste del robot.....	131
11.4.3.1	Conocer el ajuste inicial.....	132
11.4.3.2	Ajuste con corrección de peso.....	136
11.4.3.3	Conocer el ajuste estándar.....	143
11.5	Ejercicio: Búsqueda de errores después de la prueba de programa.....	145
11.6	Ejercicio: Búsqueda de errores después de la prueba de programa.....	147
12	Configuración de una nueva estación de robot: Medir la herramienta del robot.....	149
12.1	Navegador del curso.....	149
12.2	Descripción de la situación.....	149
12.3	Ejercicio: Comprobar el TCP y la orientación de la herramienta.....	151
12.4	¿Cómo puedo comprobar el TCP activo en la herramienta?.....	152
12.5	¿Qué pasos son necesarios para medir una nueva herramienta?.....	153
12.6	Preguntas: Comprobar el TCP y la orientación de la herramienta.....	155
12.7	Herramientas guiadas por robot.....	157
12.7.1	Conocer las ventajas de las herramientas medidas.....	157
12.8	Medición de una herramienta.....	159
12.8.1	Añadir herramienta/pieza nueva.....	161
12.8.2	Conocer la medición del TCP, método de 4 puntos.....	163
12.8.2.1	Medir el TCP con el método de 4 puntos.....	164
12.8.2.2	Guardar datos de carga de herramienta/pieza de trabajo.....	165
12.8.3	Conocer la orientación de la herramienta con el método ABC World.....	166
12.8.3.1	Orientación de la herramienta, establecer ABC-World.....	167
12.8.4	Conocer la orientación de la herramienta con el método ABC de 2 puntos.....	168
12.8.4.1	Orientación de la herramienta, establecer ABC-World.....	169
12.9	Ejercicio: Medir una herramienta nueva.....	171
13	Adaptación del programa en caso de cambio de herramienta.....	173
13.1	Navegador del curso.....	173
13.2	Descripción de la situación.....	173
13.3	Ejercicio: Prueba de programa después del cambio de herramienta.....	175
13.4	¿Qué sucede cuando se ha cambiado la herramienta?.....	176
13.5	Modificar instrucciones de movimiento.....	178
13.5.1	Modificación de los datos de movimiento.....	179
13.5.2	Modificación del tipo de movimiento.....	179
13.5.3	Manejo de puntos de movimiento existentes.....	181
13.6	Ejercicio: Adaptación del programa en caso de cambio de herramienta.....	182
13.7	Modificación de datos de posición.....	187

13.7.1	Modificación de datos de posición.....	187
14	Añadirá puntos de movimiento en un programa.....	189
14.1	Navegador del curso.....	189
14.2	Descripción de la situación.....	189
14.3	¿Cómo se añaden nuevos comandos de movimiento?.....	191
14.4	¿Qué información es necesaria para un movimiento de robot?.....	191
14.5	Ejercicio: Adaptar el contorno de trayectoria.....	197
14.6	Ejercicio: Adaptar el contorno de trayectoria.....	198
15	Configuración de una nueva estación de robot: Unidad de calibración para BASE.....	199
15.1	Navegador del curso.....	199
15.2	Descripción de la situación.....	199
15.3	Ventajas de una base medida.....	200
15.4	Opciones de la medición de base.....	202
15.5	Calibración de BASE con el método de 3 puntos.....	203
15.6	Modificación de los datos del Frame.....	204
15.6.1	Ejercicio (variante 2): Programar movimientos con TCP externo.....	206
15.7	Ejercicio (variante 1): Medición BASE "Contorno de trayectoria".....	209
15.8	Ejercicio (variante 2): Medición base "Cuadrícula de cubos".....	210
15.9	Ejercicio (variante 3): Medición base "Componente de soldadura".....	211
15.10	Preguntas: Medir BASE.....	212
16	Configuración de una nueva estación de robot: Creación de programas.....	213
16.1	Navegador del curso.....	213
16.1.1	Status y Turn.....	213
16.1.2	SPTP – Aproximación.....	214
16.2	Programar un movimiento de trayectoria SLIN.....	217
16.2.1	Singularidades.....	218
16.2.2	Conocer los controles de orientación SLIN.....	221
16.3	Programar un movimiento de trayectoria SCIRC.....	222
16.3.1	Programar SCIRC.....	223
16.3.2	Conocer el control de orientación SCIRC.....	225
16.4	Aproximación en los movimientos de trayectoria.....	227
16.5	Ejercicio (variante 1): Creación de programas con movimientos individuales....	229
16.6	Ejercicio (variante 2): Desplazamiento de trayectoria y aproximación.....	230
17	Configuración de una nueva estación de robot: Programación de un movimiento SPLINE.....	233
17.1	Navegador del curso.....	233
17.2	Descripción de la situación.....	233
17.3	¿Cómo se recorren los puntos programados por el robot?.....	234
17.3.1	¿Qué es SPLINE?.....	235
17.3.2	SPLINE – Conocer los segundos planos matemáticos.....	237
17.4	Conocer un bloque SPLINE.....	238
17.4.1	Conocer la trayectoria SPLINE.....	239
17.4.2	Estructura de un bloque SPLINE.....	240
17.5	Indicaciones para la programación.....	242
17.5.1	Modificar puntos en el bloque SPLINE.....	243

17.5.1.1	Modificar puntos en movimientos SPLINE combinados.....	243
17.5.1.2	Modificar puntos en movimientos SPLINE con segmentos SPL puros.....	245
17.5.2	Evitar la reducción de velocidad no deseada en los movimientos Spline.....	245
17.5.2.1	Reducción de velocidad mediante segmentos de movimiento consecutivos sin SPL.....	245
17.5.2.2	Reducción de la velocidad debida a una programación por aprendizaje no uniforme.....	247
17.6	Ejercicio (variante 1): Programar el contorno de la trayectoria con bloque Spline.....	250
17.7	Ejercicio (variante 2): Programar el contorno de soldadura con ejes adicionales.....	253
17.8	Preguntas: Contorno de la trayectoria con bloque SPLINE.....	255
18	Configuración de una nueva estación de robot: Agarrar piezas de trabajo.....	257
18.1	Navegador del curso.....	257
18.2	Descripción de la situación.....	257
18.3	Conocer los formularios inline para la garra.....	258
18.4	Añadir comandos de la garra en un programa.....	259
18.5	Comprobar los estados de la garra con formularios inline.....	262
18.6	Procedimiento.....	264
18.7	Ejercicios: Introducción.....	266
18.8	Ejercicio (variante 1): Tomar y descargar el cubo.....	267
18.9	Ejercicio (variante 2): Tomar y descargar placa.....	269
18.10	Preparación: Fijación de la placa de aplicación 3D.....	271
18.11	Ejercicio (variante 3): Recoger y depositar el componente 3D.....	273
18.12	Preguntas: Agarrar piezas de trabajo.....	275
19	Configuración de una nueva estación de robot: Liberación de la estación.....	277
19.1	Navegador del curso.....	277
19.2	Descripción de la situación.....	277
19.3	¿Qué entradas y salidas se pueden indicar?.....	278
19.4	¿Qué variables se pueden visualizar?.....	279
19.5	Ejercicio: Ampliar programas con una consulta de la estación.....	281
20	Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot.....	283
20.1	Navegador del curso.....	283
20.2	Descripción de la situación.....	283
20.3	Programación lógica simple.....	285
20.3.1	Programación de funciones de conmutación sencillas.....	285
20.3.2	Conocer el puntero de movimiento de avance.....	286
20.3.3	Comportamiento de CONT en las funciones de conmutación.....	287
20.3.4	Conocer las funciones de conmutación pulsadas "PULSE".....	288
20.3.5	Conocer la función de espera "WAIT".....	290
20.3.6	Comportamiento de conmutación en el entorno de aproximación.....	291
20.3.7	Programar una función de dependientes dependiente de la señal "WAIT FOR" con formularios inline.....	292
20.3.8	Ejercicio: Programar "WAIT-FOR".....	295
20.3.9	Preguntas: Equipar un movimiento de trayectoria con lógica.....	296

20.4	Programación de lógica con SPLINE y pasos individuales SPLINE.....	297
20.4.1	Programar el activador Spline.....	297
20.4.1.1	Programar el Trigger SPLINE en el bloque SPLINE.....	298
20.4.1.2	Programar el Trigger SPLINE en el formulario inline.....	301
20.4.1.3	Ejercicio (variante 1): Equipar el movimiento Spline con lógica, contorno 1.....	304
20.4.1.4	Ejercicio (variante 2): Equipar el movimiento SPLINE con lógica, contorno 2...	305
20.4.1.5	Ejercicio (variante 3): Equipar el desarrollo del movimiento Spline con lógica, componente de soldadura.....	306
20.4.1.6	Ejercicio (variante 4): Equipar movimiento de trayectoria con lógica, contorno de pegado.....	307
20.4.1.7	Preguntas: Equipar el movimiento SPLINE con lógica.....	308
20.4.2	Programar parada condicionada.....	309
20.4.2.1	Programar una parada condicionada en el bloque SPLINE.....	310
20.4.2.2	Programar una parada condicionada mediante el formulario inline.....	312
20.4.3	Programación de zona de desplazamiento constante.....	314
20.4.3.1	Programación de zona de desplazamiento constante.....	315
20.4.3.2	Ejercicio (variante 1): Zona de desplazamiento constante y parada condicionada, contorno 1.....	318
20.4.3.3	Ejercicio (variante 2): Zona de desplazamiento constante y parada condicionada, contorno 2.....	319
21	Configuración de una nueva estación de robot: Herramienta ext. y pieza de trabajo (proyecto de programación).....	321
21.1	Navegador del curso.....	321
21.2	Descripción de la situación.....	321
21.3	¿Cuál es el objetivo del proyecto de ejercicio?.....	322
21.3.1	Interpolación del movimiento con la pieza de trabajo guiada por el robot.....	323
21.3.2	Ejercicio: Desplazamientos manuales con herramienta fija.....	326
21.4	Conocer la medición de una herramienta fija.....	328
21.4.1	¿Cómo se mide una herramienta fija?.....	328
21.4.2	Preguntas: Medir una herramienta externa y una pieza guiada por robot.....	330
21.5	Medición de una pieza de trabajo guiada por robot.....	332
21.5.1	Medir una pieza guiada por robot.....	333
21.6	Ejercicio: Unidad de calibración para BASE.....	336
21.7	Ejercicio: Medición TOOL con el método de 2 puntos.....	337
21.8	Ejercicio (variante 1): Medir una herramienta externa y una pieza guiada por robot.....	339
21.9	Ejercicio (variante 2): Medir una herramienta de rodillos externa y una pieza de trabajo guiada por robot.....	342
21.10	Ejercicio (variante 1): Programar movimientos con TCP externo.....	345
21.11	Ejercicio (variante 3): Paletizado y despaletizado de cubos.....	349
21.12	Ejercicio: Programar "WAIT-FOR".....	354
22	Repetición condicionada de programas.....	355
22.1	Navegador del curso.....	355
22.2	Descripción de la situación.....	355
22.3	Programar un bucle sinfín.....	356
22.3.1	Programar un bucle sinfín con formularios inline	357
22.3.2	Ejercicio: Programar "LOOP-ENDLOOP".....	363
22.4	Programar un bucle de conteo.....	364
22.4.1	Programar un bucle de conteo con formularios inline	364

22.5	Conocer un bucle finito.....	371
22.5.1	Programar un bucle finito con formularios inline.....	372
22.5.2	Ejercicio: Programar "WHILE-ENDWHILE".....	380
22.6	Programar un bucle infinito.....	381
22.6.1	Programar un bucle infinito con formularios inline.....	381
22.7	Conocer consultas o ramificaciones (IF-THEN-ELSE).....	388
22.7.1	Programar consultas o ramificaciones (IF-THEN-ELSE) con formularios inline.....	389
22.8	Conocer el distribuidor (SWITCH-CASE).....	398
22.8.1	Programar distribuidores (SWITCH - CASE) con formularios inline.....	398
22.9	Preguntas: Técnicas de bucles.....	408
23	Operar la estación con un PLC de orden superior.....	409
23.1	Navegador del curso.....	409
23.2	Conocer el inicio del programa a través de un PLC de orden superior.....	409
23.3	Conocer el programa de organización Cell.src.....	410
23.4	Ejercicio: Configurar la interfaz externa de automática.....	413
23.5	Comunicación entre la unidad de control y el PLC en el ejemplo de la celda de formación.....	414
23.6	Preguntas: Trabajos en una unidad de control superior.....	418
24	Se presenta el entorno de desarrollo de WorkVisual.....	419
24.1	Navegador del curso.....	419
24.2	Demostración en directo del entorno de desarrollo de WorkVisual.....	419
25	Anexo.....	421
25.1	Vista general de los sistemas de herramientas y base.....	421
25.2	KUKA College: Datos de carga/valores de offset para herramientas/Herramientas.....	421
25.2.1	Garra en KR 16, ancho de agarre 50 mm.....	422
25.2.2	Garra en Agilus 2, ancho de agarre 25 mm.....	426
25.2.3	Soplete (estándar).....	430
25.2.4	Garra en Agilus 1, ancho de agarre 25 mm.....	431
	Índice	437

1 Introducción

1.1 Bienvenido

Nuestro lema

Mediante ejercicios prácticos adquirirá experiencia en el manejo del robot. Nuestro lema: "Lo que haga usted mismo podrá entenderlo más fácilmente y ya no lo olvidará tan rápido".



Fig. 1-1

Durante el curso



Fig. 1-2

No dude en preguntar en cualquier momento durante el curso si no ha entendido algo.

¡No existen las preguntas tontas!

Después del curso

Nuestra línea de atención al cliente de KUKA.Hotline le ayudará en caso de problemas o preguntas difíciles.



Fig. 1-3: Línea de atención al cliente

Si después del curso aún hay preguntas pendientes o surgen dudas, encontrará ayuda en nuestra base de datos de conocimientos gratuita KUKA.Xpert.



Fig. 1-4

1.2 Introducción a la robótica

Si quieres saber...

- cómo está **construido un robot**,
- cómo **se controla un robot** y
- qué **componentes** contiene una **célula de robot** ...

hemos habilitado la siguiente **formación en línea** para usted. **Simplemente eche un vistazo.**



Formación en línea Introducción a la robótica

2 Seguridad en la célula de robot

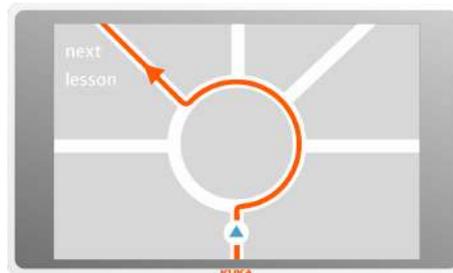
2.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está al principio de un seminario y tienes curiosidad por lo que te espera esta semana.
- Para la preparación del curso ha finalizado con éxito la formación en línea "Introducción a la robótica".

¿Qué le espera en esta unidad de aprendizaje?



- ¿Cuándo debo pulsar una PARADA DE EMERGENCIA?
- ¿Qué sucede al abrir la puerta de protección?
- ¿Qué componentes de seguridad contiene una celda de formación?
- ¿Cómo me comporto durante los ejercicios prácticos?

2.2 Descripción de la situación

Situación

La nueva instalación de producción se ha puesto en funcionamiento de acuerdo con el estado actual de la técnica. Se encuentra en perfecto estado técnico. En el futuro atenderá a la nueva instalación de producción.



Fig. 2-1: Célula de robot

Recuerde

El robot solo se mueve si usted lo desea.

En caso de que sucediera algo, existen dispositivos de seguridad adecuados.



Fig. 2-2: PARADA DE EMERGENCIA

2.3 Seguridad del robot en el ejemplo de la celda de formación College

Celda de formación en el College

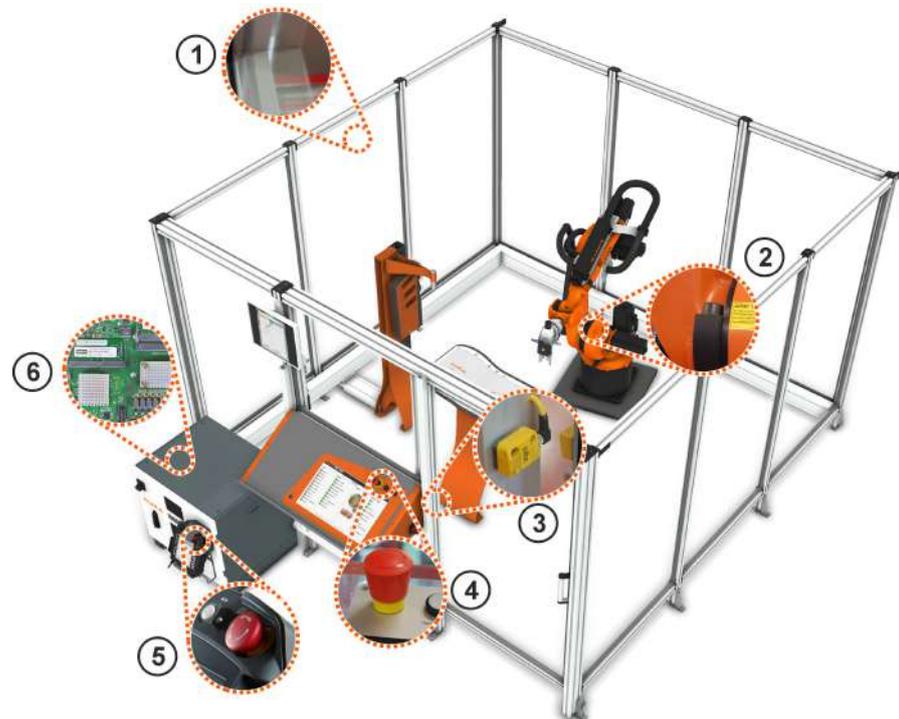


Fig. 2-3: Celda de formación

- 1 Valla de protección
- 2 Topes finales mecánicos o limitación del campo del eje para los ejes 1, 2 y 3
- 3 Puerta de protección con contacto para el control del dispositivo de seguridad móvil

- 4 Dispositivos de PARADA DE EMERGENCIA (externos)
- 5 Dispositivo de PARADA DE EMERGENCIA, pulsador de validación, interruptor de llave para abrir el gestor de conexiones
- 6 Control de seguridad integrado

¿Por qué se necesitan los componentes de seguridad individuales?

1. Las personas no pueden entrar. Los componentes suspendidos y, en su caso, otras emisiones se retienen.
2. El robot no puede abandonar la celda.
3. El personal no puede entrar en la zona de movimiento del robot durante la producción.
4. La instalación se puede detener sin smartPAD.
5. El operario puede detener la instalación.
6. Se garantiza una reacción segura.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Estructura y función de un sistema de robot > Seguridad del robot en el ejemplo de la celda de formación College

2.4 Ejercicio: Seguridad en una célula de robot

Enunciado 1

Accione la PARADA DE EMERGENCIA local durante el modo automático.



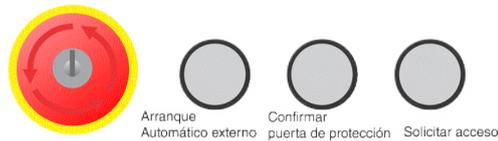
Enunciado 2

Se le mostrará cómo manejar la garra durante la formación.



Enunciado 3

Se le mostrará la pulsación de la PARADA DE EMERGENCIA externa durante el modo automático.



Enunciado 4

Se le mostrará cómo cambiar los modos de servicio en el smartPAD.



Enunciado 5

Se le mostrará cómo puede mover los ejes individuales del robot en dirección positiva y negativa.



2.5 Preguntas: Seguridad en una célula de robot

Lo que se debe saber tras el ejercicio:

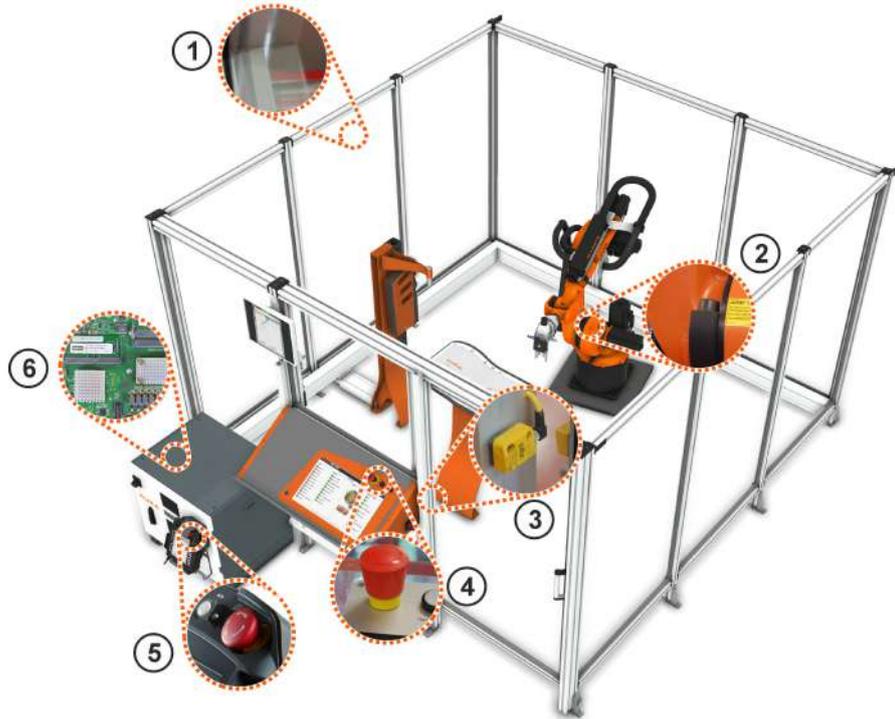


Fig. 2-4: Celda de formación

¿Cómo se puede evitar que durante la producción haya personas dentro de la zona de peligro?



Fig. 2-5: Campo de respuesta

¿Cómo se puede detener el robot de forma segura durante la producción?



Fig. 2-6: Campo de respuesta

¿Por qué debe haber una PARADA DE EMERGENCIA externa en una instalación?



Fig. 2-7: Campo de respuesta

¿Cómo se garantiza que el robot no abandone la valla de seguridad?



Fig. 2-8: Campo de respuesta

¿En qué modos de servicio se puede mover el robot con la puerta de protección abierta?



Fig. 2-9: Campo de respuesta

2.6 Indicaciones generales de seguridad

Instrucción

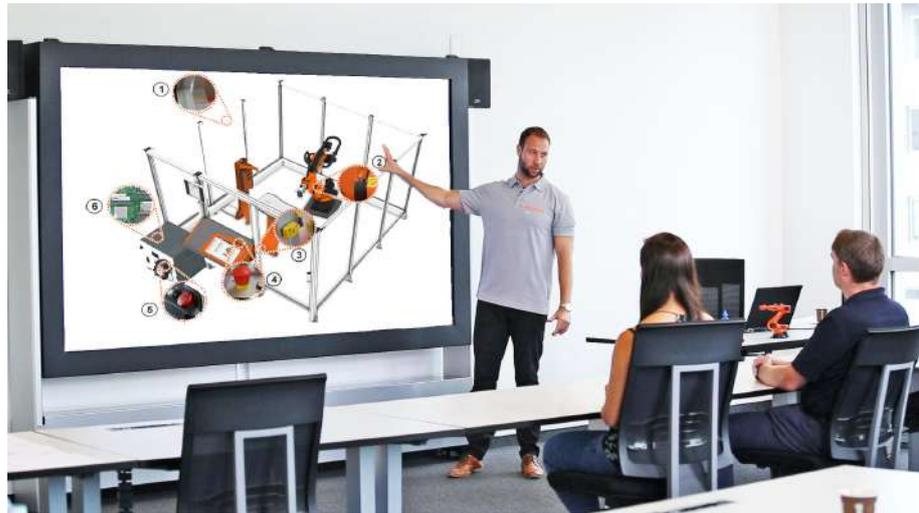


Fig. 2-10: Instrucciones

- Antes de comenzar a trabajar se deberá informar al personal implicado sobre la naturaleza y el alcance de los trabajos que se realizarán, así como sobre los posibles peligros.
- Debe llevarse un registro de los motivos y el alcance de las instrucciones.

Comportamiento en caso de incendio



Fig. 2-11: Fuego

- Se deben tener en cuenta las **normas sobre seguridad contra incendios** específicas del país.
- Tener en cuenta los **planos de evacuación**.

Indicaciones generales de seguridad



Fig. 2-12: "Músculos" del robot

- El sistema de robot está fabricado conforme al estado actual de la técnica y las normas técnicas de seguridad vigentes. No obstante, una utilización inadecuada o no conforme a los fines previstos puede

provocar **peligro de sufrir lesiones o accidentes mortales**, así como **daños materiales**.



PELIGRO

El robot siempre será más resistente que una persona. Incluso los robots con cargas pequeñas producen el suficiente par de apriete como para provocar aplastamientos o lesiones graves y mortales. En estos casos, la velocidad de desplazamiento **es irrelevante**.

- El sistema de robot debe utilizarse únicamente en **perfecto estado técnico** y conforme a los fines previstos, respetando las medidas de seguridad y teniendo en cuenta los peligros que puedan generarse. Cualquier avería que pueda afectar a la seguridad deberá subsanarse de inmediato.
- Antes de iniciar el trabajo, de la puesta en servicio o de la reanudación del servicio, realizar una comprobación visual para descartar daños en el sistema de robot.



ADVERTENCIA

Peligro de muerte por colisión con el obstáculo

Si el manipulador o un eje adicional chocan contra un obstáculo o un tope final mecánico o bien contra la limitación mecánica de eje, el manipulador ya no podrá funcionar con seguridad. Pueden producirse la muerte, lesiones graves o daños materiales.

- Poner fuera de servicio el manipulador.
- Poner fuera de servicio el eje adicional.
- Antes de la reanudación del servicio, debe consultarse con la empresa KUKA.

Comportamiento para el acceso a la celda de formación

1. Deben seguirse todas las indicaciones del formador.



Fig. 2-13: Demostración del formador

2. Únicamente podrá accederse a la celda de formación de programación utilizando **calzado resistente y cerrado**. Solo se puede acceder a las zonas de montaje del área mecánica con **calzado de seguridad**.



Fig. 2-14: Calzado de seguridad

3. En el área de práctica silenciar los teléfonos móviles.



Fig. 2-15: evitar el teléfono móvil

4. Está prohibido comer y beber en el área de práctica.



Fig. 2-16: no se permite comida ni bebida

5. Por lo general, está permitido permanecer en la zona de movimiento del robot únicamente en el modo de servicio "**Manual velocidad reducida (T1)**" con el smartPAD.



Fig. 2-17: Modos de servicio

6. Se debe avisar a todas las personas que se encuentren en los alrededores del robot sobre el próximo movimiento del mismo.



Fig. 2-18: Advertencia

7. Ni las herramientas ni el robot deben tocar nunca la valla de seguridad o sobresalir de la misma.



Fig. 2-19: Zona de celda

Comportamiento al abandonar la celda de formación



Fig. 2-20: Abandonar la celda de formación

1. Ajustar el **modo de servicio "Manual velocidad reducida" (T1)**.
2. Fijar el smartPAD en el **soporte** previsto para ello.
3. Enrollar el **cable del smartPAD** para que no se produzcan puntos de tropiezo.

Sustancias estupefacientes

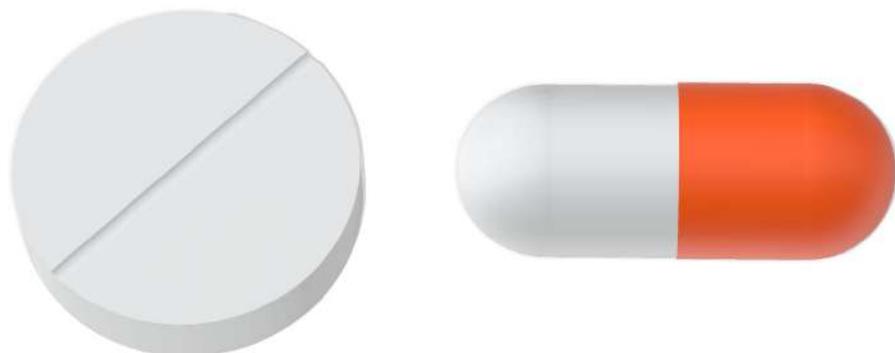


Fig. 2-21: Tabletas

Los participantes de los cursos no deben encontrarse bajo la influencia del alcohol, las drogas, los medicamentos u otras sustancias estupefacientes que pudieran poner en peligro la seguridad de los mismos o de terceras personas.

3 Documentación del sistema de robot

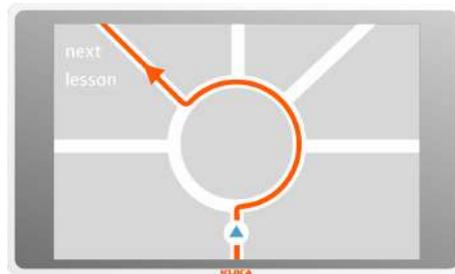
3.1 Navegador del curso

¿Qué requisitos aporta usted?



- No se requieren conocimientos previos especiales.
- Está buscando documentación, soluciones o instrucciones de trabajo en relación con la gama de productos de KUKA.

¿Qué le espera en esta unidad de aprendizaje?



Conocer y aplicar KUKA Xpert

- Conocerá la base de datos de conocimientos digital KUKA Xpert.
- Conocerá el valor añadido de las variantes KUKA Xpert de pago.
- Sabrá cómo iniciar sesión y encontrar información en la base de datos de conocimientos.

3.2 KUKA Xpert

Descripción



Fig. 3-1

Una base de datos de conocimientos digital para todos los productos KUKA, accesible en cualquier momento y desde cualquier lugar: KUKA Xpert ofrece información técnica detallada para técnicos de servicio, planificadores, programadores, operarios y técnicos de puesta en servicio.

Beneficios del cliente de KUKA Xpert

- **Reducir los tiempos de parada**
Comprenderá mejor los mensajes del sistema con ayuda de las causas, los síntomas y las soluciones. Solucionará los problemas con instrucciones de trabajo de alta calidad.

- **Ser más eficiente**
Encontrará la información que necesita ad hoc utilizando la potente búsqueda de texto libre y filtros por productos y tipos de información.
- **Información de calidad**
Los contenidos los elaboran redactores técnicos y son revisados por expertos en los productos. El servicio de atención al cliente de KUKA también utiliza Xpert.
- **Conocimientos 24/7**
Obtendrá la información adecuada cuando más la necesite sin malgastar su tiempo valioso.
- **No requiere ninguna instalación**
KUKA Xpert se basa en la web y es accesible desde una gran variedad de dispositivos. Se necesita una conexión a internet y un navegador web actual.
- **Se beneficiará de información enlazada e interactiva.**
Los contenidos interactivos, como vídeos con datos vinculados semánticamente, le ayudan a obtener toda la información relevante del producto seleccionado.

Comparación de contenido de las licencias

Tipo de información	KUKA Xpert & KUKA Xpert Trial	KUKA Xpert Basic Monitorización
Información acerca de los productos		
Piezas de repuesto		
Especificaciones		
Seguridad		
Instrucciones de montaje		
Instrucciones de servicio		
Instrucciones de servicio y de programación		
Datos de simulación y de CAD		
Base de datos de casos con síntoma, causa y solución		
Instrucciones de trabajo detalladas		
Variables del sistema KSS		
Sinopsis de compatibilidades		
Directivas		

Tipo de información	KUKA Xpert & KUKA Xpert Trial	KUKA Xpert Basic Monitorización
Descarga del código del programa y configuraciones		
Descripción de funcionamiento		

Contenidos para diferentes grupos destinatarios

Tipo de información	Planificador Constructor	Técnicos de servicio Técnico de mantenimiento	Técnico de puesta en servicio Programador
Información acerca de los productos			
Documentación			
Especificaciones/Datos técnicos			
Montaje			
Transformación/Retrofit			
Instrucciones de trabajo			
Piezas de repuesto			
Base de datos de casos			
Configuraciones			
Reparación y mantenimiento			
Reanudación del servicio			
Programación			
Adaptación			

3.3 Registro

Procedimiento

1. Introducir el enlace de la hoja adjunta o escanear el código QR. Alternativamente, hacer clic aquí: <https://xpert.kuka.com>.
2. Hacer clic en **¿Aún no tiene una cuenta en my.KUKA?** para crear una nueva cuenta.

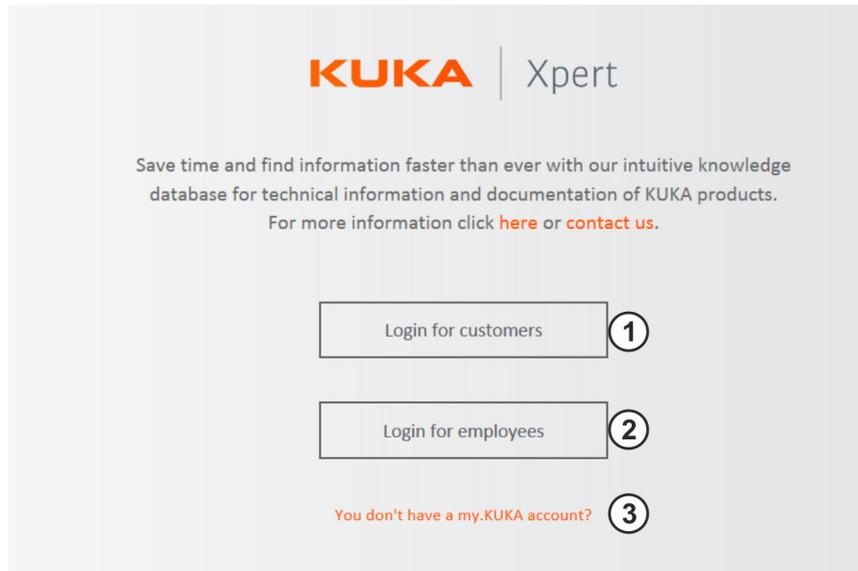


Fig. 3-2

Pos.	Explicación
1	Inicio de sesión para clientes
2	Inicio de sesión para empleados de KUKA
3	Reenvío al formulario de registro

3. Rellenar el formulario de registro y aceptar la política de privacidad, así como las condiciones de uso. Se enviará un correo electrónico para el registro a la dirección indicada.

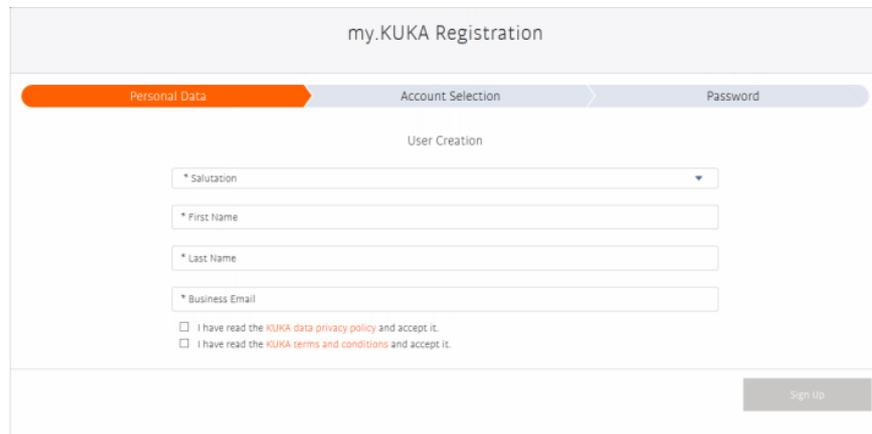


Fig. 3-3

4. En el correo electrónico recibido, hacer clic en el botón **Verify your Account**. Se abrirá la página web.
5. Establecer la contraseña y completar la información.

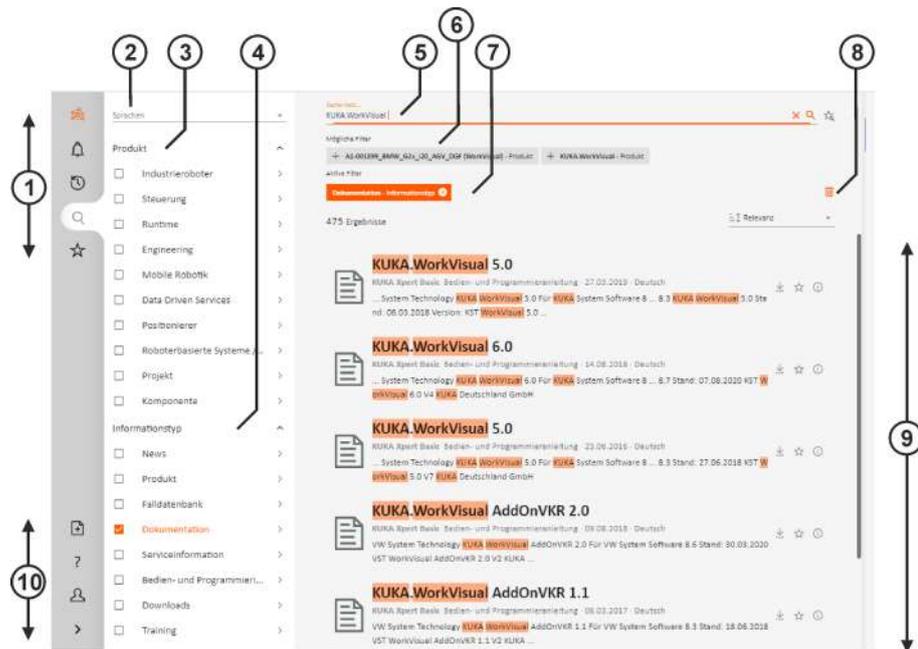
El registro se realiza en el portal my.KUKA. El portal my.KUKA es un portal global para los clientes de KUKA y el punto de acceso digital para la empresa. En el portal los clientes pueden encontrar información adicional específica, funciones digitales y servicios. Si todavía no se dispone de acceso a my.KUKA, debe crearse una cuenta nueva.



Solo es posible registrarse con una dirección de correo electrónico de su empresa. Queda excluido el registro con una dirección de correo electrónico privada.

3.4 Utilizar KUKA Xpert Basic

Vista general

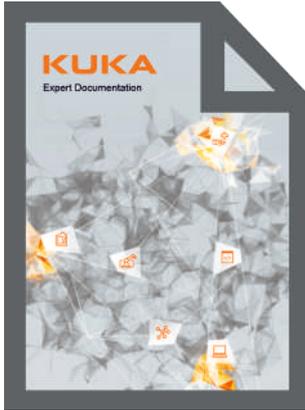


Documentación del sistema de robot

Fig. 3-4: Vista general KUKA Xpert

- 1 Novedades: visualización de las últimas publicaciones
- Historial de búsqueda: vista general del historial de búsqueda personal
- Búsqueda: acceso a la página de búsqueda
- Favoritos: lista personal de favoritos
- 2 Ajuste del filtro de voz para la búsqueda
- 3 Filtro de búsqueda para el producto
- 4 Definir el tipo de información para la búsqueda
- 5 Campo de entrada para la búsqueda por palabra clave
- 6 Subfiltros propuestos para limitar los resultados de la búsqueda
- 7 Filtros activos
- 8 Eliminación de todos los filtros y términos de búsqueda
- 9 Resultados de la búsqueda
- 10 Instrucciones breves y ajustes personales

Podrá encontrar más información en la documentación del cliente.



Libro

KUKA Xpert Basic

Instrucciones de servicio y programación

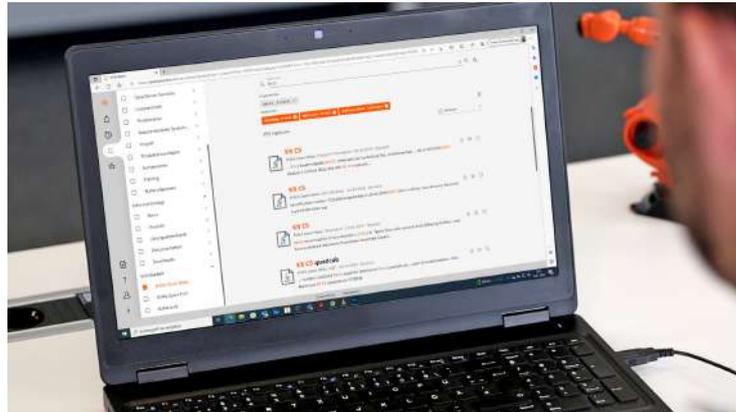
Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

3.5 Ejercicio: Buscar documentación sobre Xpert-Basic

¡Ahora es su turno!



1. Abra KUKA Xpert Basic a través de su tablet e inicie sesión.
2. Busque la documentación indicada a continuación para obtener más detalles sobre KUKA Xpert Basic.



Libro	KUKA Xpert Basic Instrucciones de servicio y programación
Filtro de búsqueda	Documentación > Instrucciones de manejo y programación
Capítulo/apartado	---

4 Manejo de programas: Probar programas con velocidad reducida manualmente

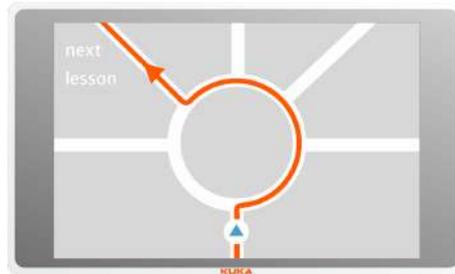
4.1 Navegador del curso

¿Qué requisitos aporta usted?



- Conoce los diferentes dispositivos de seguridad de una célula de robot.
- Sabe cómo debe comportarse correctamente en y dentro de la celda.

¿Qué le espera en esta unidad de aprendizaje?



Manejo de programas

- Conocerá el uso seguro del pulsador de validación en el smartPAD.
- Podrá cambiar al modo de servicio manual "T1".
- Sabrá cómo se adapta la velocidad de desplazamiento "Override manual".
- Conocerá la función de las "teclas de inicio" en el smartPAD.
- Sabrá cómo seleccionar/deseleccionar o restablecer programas.
- Aprenderá a interpretar los mensajes en el smartPAD.

4.2 Descripción de la situación

Situación



Fig. 4-1: Visualizar programas en el smartPAD

- En la producción se ha configurado una nueva instalación. Aún no conoce los programas. Debe revisar la instalación en el primer paso.
- Después de revisar todas las funciones de seguridad (módulo de seguridad), obtenga una visión general de los módulos de programa existentes y pruébelos.

Pregunta



Fig. 4-2

¿Conoce ejemplos de la producción en los que el robot recoge un componente para el procesamiento posterior?

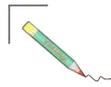


Fig. 4-3: Campo de respuesta

4.3 Seleccionar programas de robot

Para poder ejecutar un programa de robot, debe seleccionarse. Los programas del robot se encuentran disponibles en el navegador de la interfaz de usuario.



4.3.1 Seleccionar programas de robot

Conocer el navegador smartPAD

En el navegador se organizan programas de desplazamiento en estructuras de carpetas. Estas se encuentran generalmente en la carpeta principal "R1".



Fig. 4-4: Seleccionar un módulo

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Ejecutar el inicio del programa

4.4 Mensajes en el smartPAD

Descripción

Si hay mensajes o fallos por parte de la unidad de control, estos se visualizan en la ventana de mensajes en el smartPAD.



Fig. 4-5: Conocer la ventana de mensajes

Resumen de mensajes



Fig. 4-6: Contador de mensajes

Mensaje de confirmación	Mensaje de estado	Mensaje de observación	Mensaje de espera

Consejos para el manejo de los mensajes

- Leer el mensaje detenidamente.
- Primero leer los mensajes más antiguos. Un mensaje nuevo podría ser una consecuencia de un mensaje anterior.
- No pulsar simplemente el botón **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.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Reaccionar a los mensajes del smartPAD > Leer e interpretar mensajes

4.5 Conocer los modos de flujo de programa

Descripción

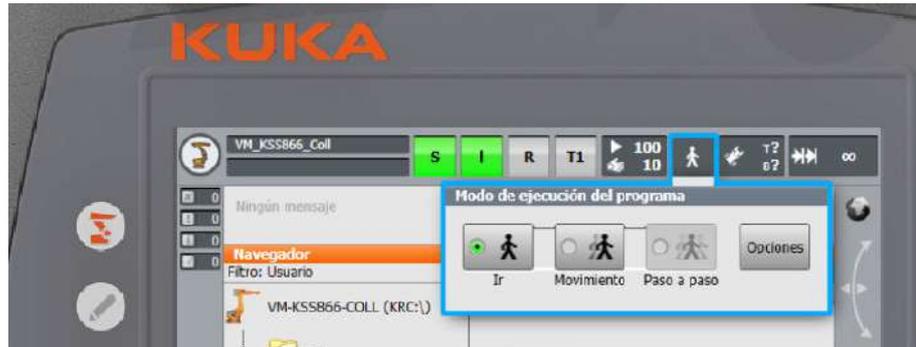


Fig. 4-7: Tipos de flujo de programa, smartPAD



GO



Movimiento



Paso a paso | Disponible únicamente en el grupo de usuarios "Experto".

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Conocer los modos de flujo de programa

4.6 Iniciar el programa

Procedimiento



Fig. 4-8: Seleccionar un módulo

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Ejecutar el inicio del programa

4.7 Conocer el desplazamiento de inicialización

Descripción

El desplazamiento de inicialización compara la posición del robot actual con las coordenadas del punto programado actual.

Si coinciden, se puede realizar una planificación de la trayectoria.

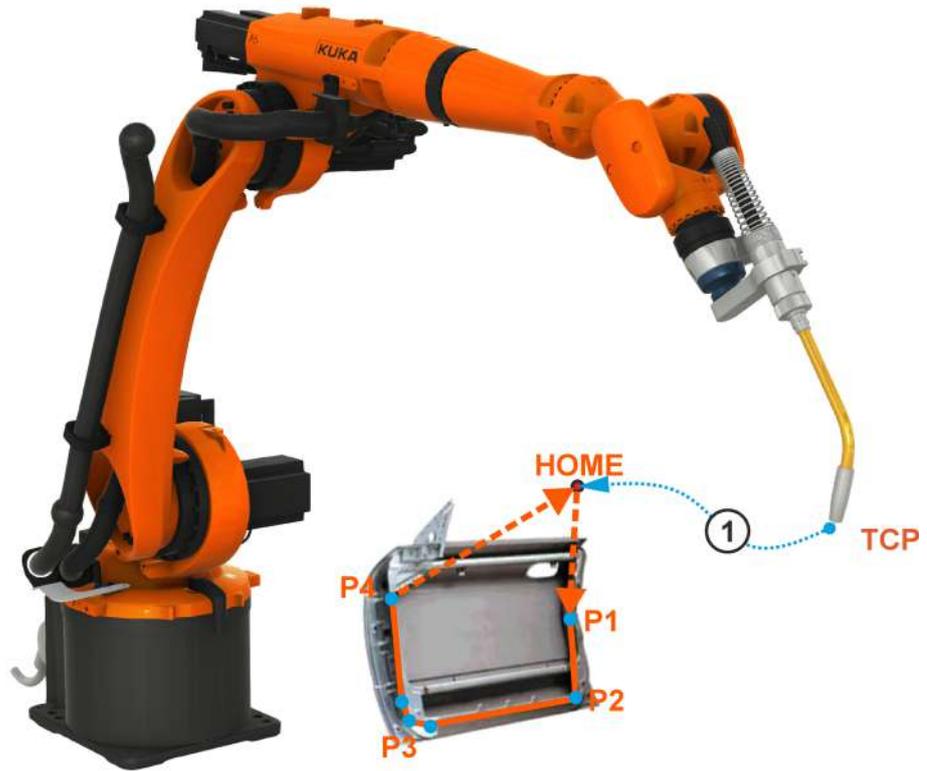


Fig. 4-15: Ejemplo de un desplazamiento COI en selección de programa



Desplazamiento de inicialización = desplazamiento COI

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Conocer el desplazamiento de inicialización

4.8 ¿Cómo se prueban los programas?

Procedimiento a seguir



1. Visualizar la estructura de la célula completa para detectar y comprender su funcionamiento y las fuentes de peligro.
2. Visualizar los programas preparados en el smartPAD.
3. Antes de probar un programa, seleccionar obligatoriamente el modo de servicio T1 y reducir el POV.

De este modo, el programa se ejecuta a velocidad reducida y segura.

4. A continuación y con el modo de servicio T1 seleccionado, se puede acceder a la celda con el smartPAD para efectuar la prueba.

A menudo se pueden reconocer mejor los procesos de producción in situ.



ATENCIÓN

La puerta de protección permanece abierta en este caso.
Prestar atención a las fuentes de peligro, como las piezas móviles > Peligro de aplastamiento y lesiones.

5. Si se comprueba un módulo de programa fuera de un desarrollo del proceso, debe asegurarse de que este arranca siempre desde una posición de salida definida, la llamada posición HOME.

4.9 Ejercicio: Ejecutar programas del robot

Enunciado 1



1. Lea e interprete los mensajes pendientes en el robot.
2. Si es necesario, desbloquee la PARADA DE EMERGENCIA y confírmela.
3. Asegúrese de que esté seleccionado el modo de servicio "T1".
4. Seleccione el programa _____.
5. Ejecute el programa con un POV inicial de _____.

AVISO

Al realizar el desplazamiento COI, preste atención a posibles obstáculos en la trayectoria.



ATENCIÓN

La puerta de protección permanece abierta en este caso. Prestar atención a las fuentes de peligro, como las piezas móviles > Peligro de aplastamiento y lesiones.

6. Después del final del programa, restablezca el programa y vuelva a iniciarlo con un POV superior. Repita el proceso hasta alcanzar el POV completo del 100 %.

5 Manejo de programas: Probar programas con velocidad de proceso

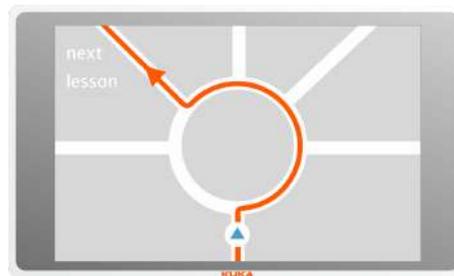
5.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para seleccionar/deseleccionar y resetear programas.
- Está capacitado para probar programas en el modo de servicio manual "T1".
- Está capacitado para interpretar los mensajes en el smartPAD.

¿Qué le espera en esta unidad de aprendizaje?



Probar programas con velocidad alta

- Aprenderá a probar los programas de robot de forma segura con una **velocidad alta manual**.
- Aprenderá cómo cambiar a los modos de servicio "T2" o "Automático".
- Evaluará correctamente el potencial de peligro de un robot a alta velocidad y aprenderá a comportarse de un modo consciente de la seguridad de la celda.

5.2 Descripción de la situación

Situación

Al configurar la celda, el nuevo programa ya se ha probado con éxito con velocidad reducida manualmente.

En las aplicaciones, como p. ej. el pegado, la calidad de aplicación es decisiva. Por lo tanto, es necesario que el programa también se pruebe desde una distancia cercana y segura con velocidad de proceso completa.



Fig. 5-1: Probar robot con velocidad alta manual

5.3 Probar un programa con velocidad alta

Descripción

El programa se prueba con alta velocidad para poder validar el proceso (ejemplo en una aplicación de pegamento).

El programa se prueba con alta velocidad aumentando gradualmente el override del programa después de cada ejecución del programa. De este modo se puede validar un comportamiento diferente de las piezas de montaje o las piezas de trabajo en las garras.

Aparece un mensaje de estado.



Fig. 5-2: Mensaje

¿Qué hay que tener en cuenta al probar programas con alta velocidad?

Si la aplicación lo **requiere obligatoriamente**, los programas también pueden probarse dentro del dispositivo de seguridad con la puerta de protección abierta.

El programador puede realizar trabajos de ajuste con alta velocidad dentro del dispositivo de seguridad sin puentear ni desactivar las medidas de protección.



Fig. 5-3: Modo de servicio T2 seleccionado



PELIGRO

En esta situación, la seguridad solo se garantiza manteniendo presionado el pulsador de validación.

El comportamiento es impredecible, sobre todo en el caso de movimientos imprevisibles del robot (SPTP, PTP) combinados con velocidades de desplazamiento elevadas.

Por lo tanto, por lo general no se recomienda permanecer en la celda durante la realización de pruebas con el modo de servicio T2.

Con la desactivación consciente o inconsciente de los dispositivos de seguridad existe peligro de muerte.



PELIGRO

Durante el modo automático no debe haber ninguna persona dentro del dispositivo de seguridad.

¿Por qué aparece el mensaje de estado?

Es obligatoria una limitación de la velocidad.

Si se modifica el modo de servicio a T2, posteriormente el override de programa (POV) será del 10 % como máximo.

Las limitaciones se aplicarán hasta que el POV se modifique manualmente. Posteriormente, se aplicará el POV ajustado manualmente o la siguiente asignación en el programa.

Si el pulsador de validación no se presiona en T2 durante 5 min (o se presiona a fondo durante 5 min), también se activarán las limitaciones de velocidad



Fig. 5-4: Mensaje

5.4 Conocer los modos de servicio de la unidad de control del robot



Fig. 5-5: Modos de servicio



En función del grupo de usuarios seleccionado, no todos los modos de servicio están disponibles.
La asignación del modo de servicio que puede aplicar un grupo de usuarios puede asignarse a través del grupo de usuarios "Administrador".

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
- La velocidad en el servicio con programa corresponde a la velocidad programada.
 - > El usuario puede reducir la velocidad con el POV (*override del programa*)
- El robot no puede moverse con el ratón 6D ni con las teclas de desplazamiento

AUT (Automático)

- Para robots industriales **sin** unidad de control superior (PLC)
- ¡Velocidad en el modo de programación según la velocidad programada!
 - > El usuario puede reducir la velocidad con el POV (*override del programa*)
- El robot no puede moverse con el ratón 6D ni con las teclas de desplazamiento

5.5 Cambiar el modo de servicio

Procedimiento



Fig. 5-6: Abrir el administrador de conexiones

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer y cambiar los modos de servicio de la unidad de control del robot

5.6 Ejercicio: Ejecutar programas del robot

Enunciado 1

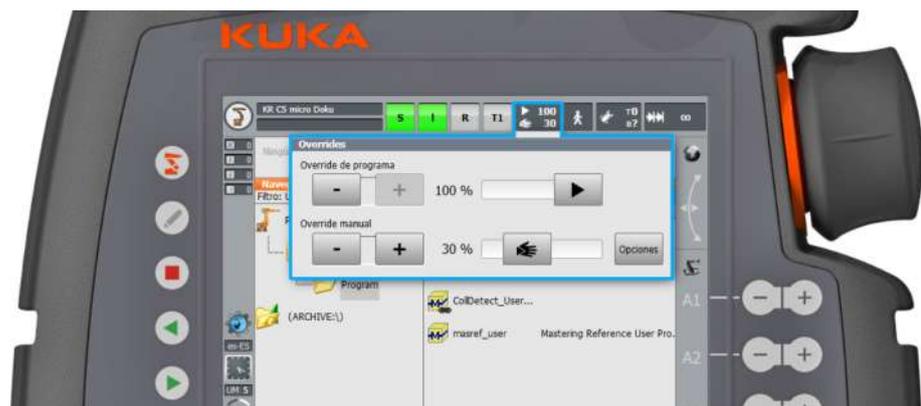
1. Cambie a un modo de servicio adecuado.
2. Seleccione el **programa:**_____ .
3. Tenga en cuenta los posibles mensajes que puedan aparecer y actúe en consecuencia.
4. Pruebe el programa con alta velocidad manual aumentando gradualmente el POV del 30% al 50%, 75% y 100% después de cada ejecución. Observe el comportamiento del robot y de todas las piezas de montaje.

Enunciado 2

1. Cambie al modo de servicio Automático.



2. Seleccione un override de programa adecuado. Tenga en cuenta los posibles mensajes que puedan aparecer y actúe en consecuencia.



3. Conecte los accionamientos del robot e inicie la producción.



Manejo de programas: Probar programas con velocidad de proceso

5.7 Preguntas: Ejecutar los programas del robot

Lo que se debe saber tras el ejercicio:



¿Qué indica el siguiente mensaje en el smartPAD?



Fig. 5-7: Mensaje

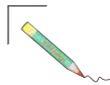


Fig. 5-8: Campo de respuesta

¿Qué pasos deben cumplirse para poder probar un programa en el modo de servicio T1?

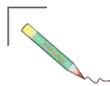


Fig. 5-9: Campo de respuesta

¿Ha notado algo al realizar la prueba de su programa? ¿A qué debe prestarse atención adicionalmente al ejecutar el programa de robot?



Fig. 5-10: Campo de respuesta

Manejo de programas: Probar programas con velocidad de proceso

6 Eliminar un fallo en la instalación: Reaccionar ante una estación de descarga vacía

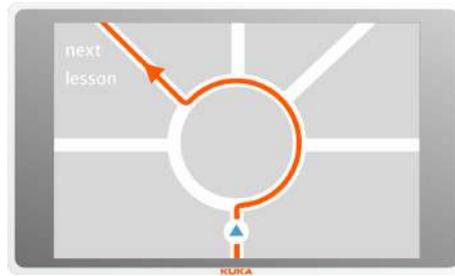
6.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar programas de robot.
- Está capacitado para probar programas de robot en modos de servicio manuales o en modo automático.

¿Qué le espera en esta unidad de aprendizaje?



Reaccionar ante las averías del programa

- Aprenderá cómo reaccionar ante averías/mensajes en el robot.
- Aprenderá cómo liberar el robot de la situación de avería.
- Aprenderá a utilizar el tipo de desplazamiento "Guía".
- Puede iniciar el programa del robot de forma específica mediante la selección de paso en cualquier punto.
- Para ello, utilizará las teclas de desplazamiento, las teclas de programa y el ratón 6D en función de la situación.
- Manejará la garra a través del paquete de opciones KUKA.Gripper-Tech.

6.2 Descripción de la situación

Situación



En su empresa hay una célula de robot para el procesamiento de componentes. Los componentes se extraen de un cargador o una caja de almacenamiento. A continuación, los componentes se procesan y se depositan. La reserva de piezas debe rellenarse manualmente.

En esta aplicación, una cámara detecta la posición y la posición de los componentes antes de agarrar (esta técnica no forma parte de la formación).

Para evitar que un recipiente se vacíe, existen varias estaciones de descarga o cargadores. Si un recipiente se vacía, el robot recoge automáticamente las piezas del siguiente recipiente.

En muchas instalaciones hay indicadores luminosos para mostrar un atasco o de un almacén vacío.



Fig. 6-1: Cargar piezas adicionales

6.3 ¿Qué pasa si un cargador se vacía?

Descripción



Fig. 6-2: Garra, cargador 4

1. El robot se desplaza a la estación de descarga.
2. La garra se cierra a pesar de que no hay ningún componente en la estación de descarga.
3. Aparece un mensaje de diálogo en el smartPAD.
4. El robot detiene la ejecución del programa.

Información adicional

En muchas instalaciones, las estaciones de descarga están equipadas con un iniciador de componentes. Si no hay ningún componente en la es-

tación de descarga, el robot se desplaza a otro recipiente o espera el comando para continuar trabajando en una posición no crítica.



Fig. 6-3: Iniciador, óptico

¿Por qué el robot no sigue trabajando?

La garra se encuentra en una posición incorrecta / se ha producido el estado de la garra incorrecto. A través de los iniciadores montados se transmite este estado al robot, lo que permite detectar el error y detiene el robot. Como reacción se emite un mensaje de diálogo con el estado incorrecto de la garra.



Fig. 6-4

6.4 Manejo de la garra con KUKA.GripperTech

Descripción

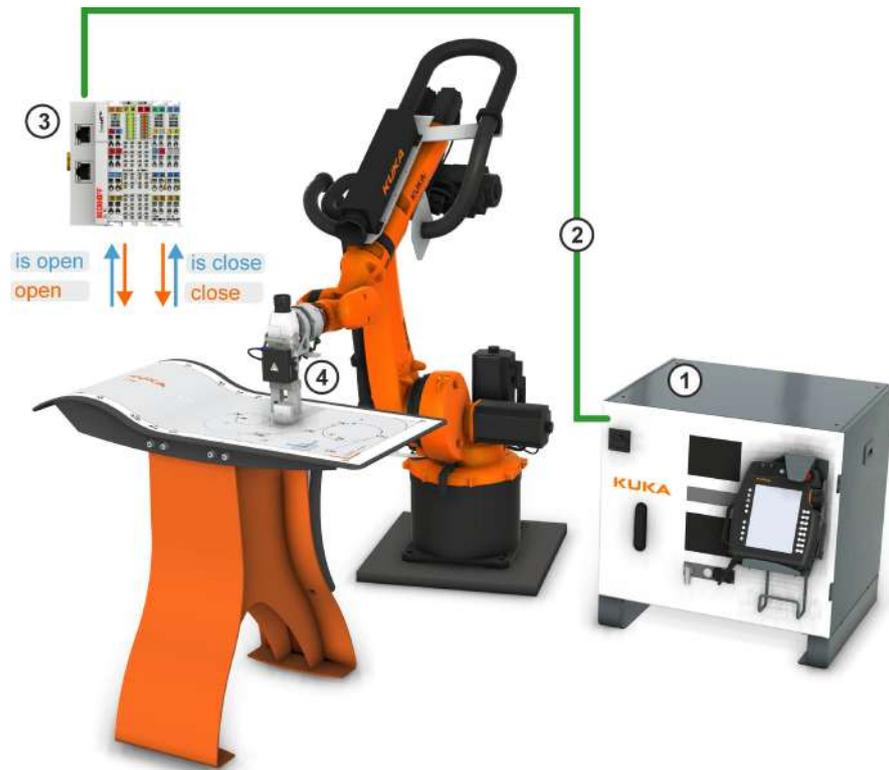


Fig. 6-5: Control de la garra

- | | | | |
|---|-------------------|---|--|
| 1 | Unidad de control | 3 | Módulo de bus de campo con módulos de entrada y salida |
| 2 | Bus de campo | 4 | Garras con entradas y salidas digitales |

Las herramientas, como las garras de la célula de formación (4), deben abrirse o cerrarse durante el tiempo de ejecución del programa. Por regla general, el mando de la garra se realiza a través de un sistema de bus de campo configurado mediante los módulos de entrada y salida digitales (3).

Manejo de la garra

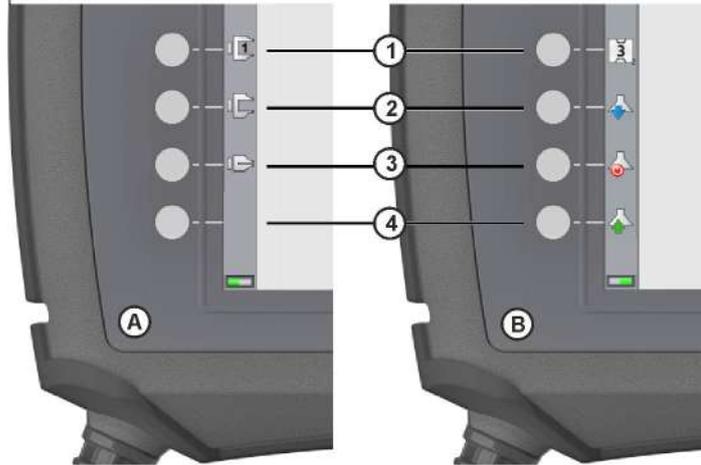


Fig. 6-6: Teclas de garra

- Pulsando varias veces la tecla de estado (1) puede pasarse de una garra configurada a otra.
- **Imagen A: Garra estándar**
Garra con teclas de estado para las funciones Abrir (2) y Cerrar (3).
> Pulsando el pulsador de validación se "activan" los símbolos.
- **Imagen B: Garra de aspiración**
Garra con teclas de estado para las funciones Soplado (2), OFF (3) y Aspiración (4).
> Pulsando el pulsador de validación se "activan" los símbolos.



ADVERTENCIA

Al utilizar el sistema de garras existe riesgo de aplastamiento y corte. Las personas que manejan la garra deben asegurarse de que la garra no pueda aplastar ninguna parte del cuerpo.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer el paquete de opciones KUKA.Gripper y SpotTech

Manejar y programar las garras a través del paquete de opciones KUKA.Gripper y SpotTech

6.5 Conocer las teclas de ejecución del programa "Inicio hacia delante" e "Inicio hacia atrás"

Descripción



Fig. 6-7: Dirección de avance del programa: Adelante/atrás

- Para iniciar un programa se dispone de las teclas de inicio "Hacia delante"  y de inicio "Hacia atrás" .
- Solo es posible ejecutar un programa hacia atrás con la tecla , si se ha ejecutado la línea INI de un programa y el robot se ha desplazado inmediatamente antes hacia delante por la trayectoria que debe recorrerse hacia atrás.
- Sin embargo, si el robot se desplaza manualmente, p. ej., después de una parada del programa, ya no se podrá realizar el desplazamiento hacia atrás.
- A partir de las versiones de software KSS 8.5 y superiores está disponible el tipo de desplazamiento manual separado "Guía". En este caso, el desplazamiento se realiza por la trayectoria de manera óptica/física, pero se ha abandonado la trayectoria programada.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro	Programación de robots 1 KUKA System Software 8.6 Referencia de formación
Capítulo/apartado	Ejecutar programas de robot > Conocer las teclas de ejecución del programa "Inicio hacia delante" e "Inicio hacia atrás"

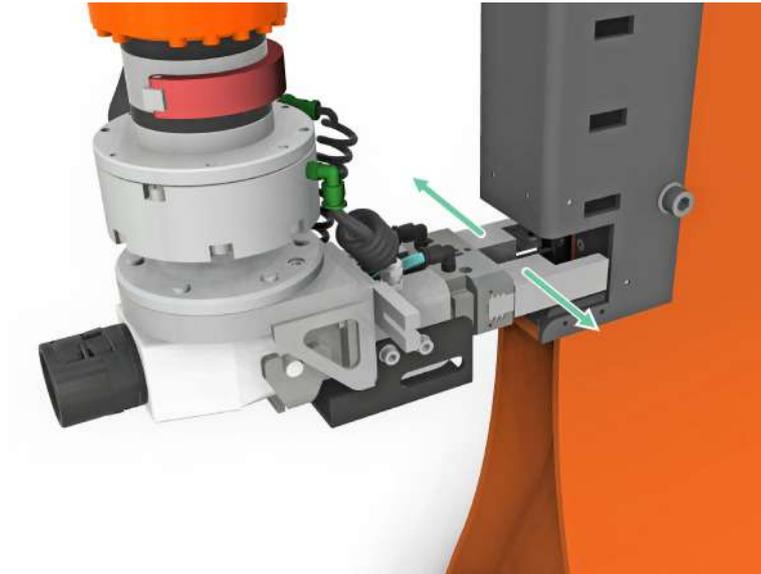
6.6 ¿Cómo se puede reaccionar ante una cargador vacío?

Variante 1

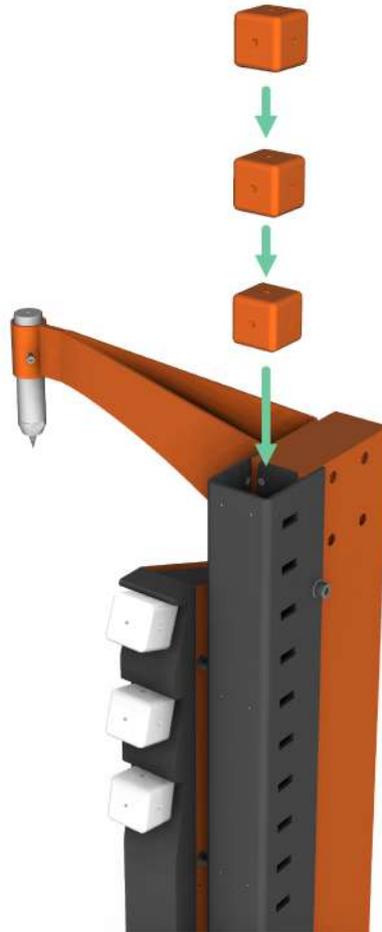
1. Se comprueba la posición de la garra.
> La garra se detiene con un mensaje de diálogo.



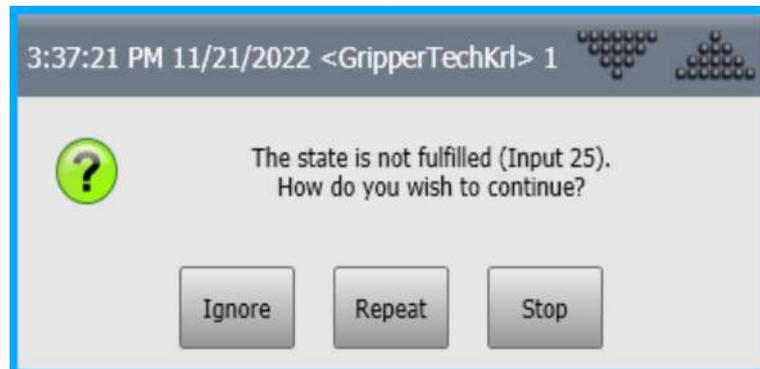
2. Abrir la garra a través del smartPAD.



3. Rellenar el cargador.



4. Ejecutar el proceso de agarre con el botón "Repetir" del mismo nombre.



Esta pregunta de diálogo solo se realiza en los modos de servicio de prueba T1 y T2.

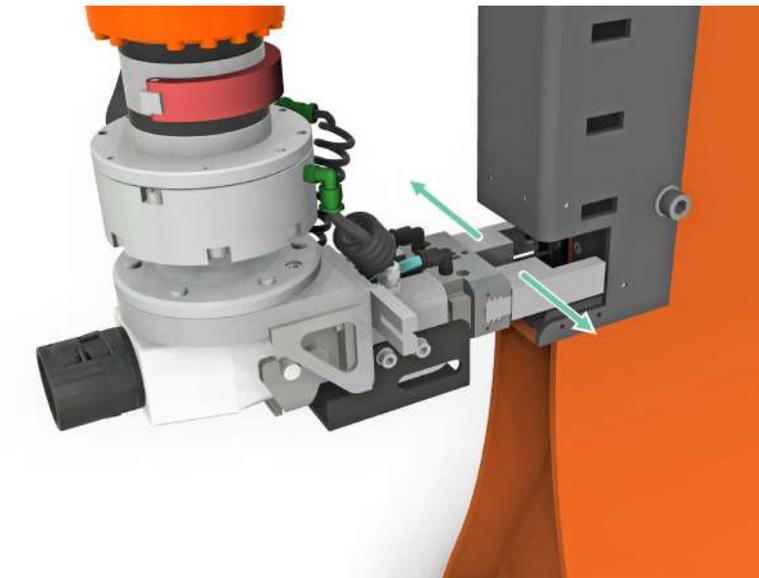


Esta pregunta de diálogo solo se realiza en los modos de servicio de prueba T1 y T2.

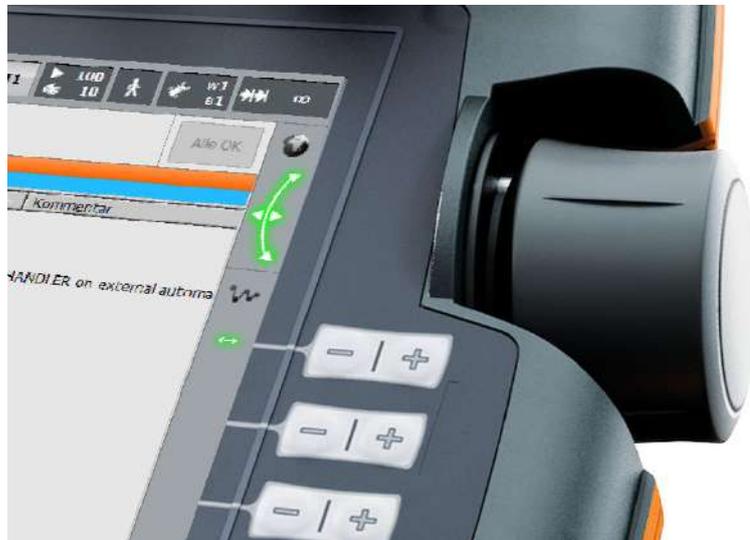
6.7 Otra posibilidad para responder ante un cargador vacío

Variante 2

1. Abrir la garra a través del smartPAD.



2. Desplazar el robot con la guía desde la posición de la garra.



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

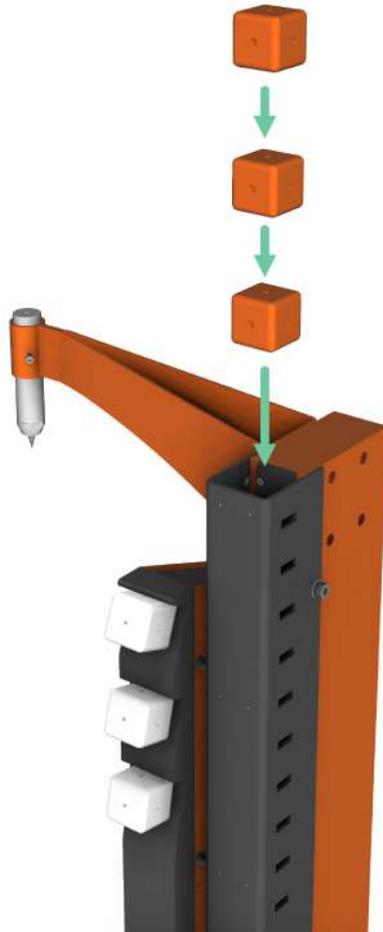
KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Mover el robot con el tipo de desplazamiento Guía

3. Rellenar el cargador



- 4. La posición previa se desplaza con una selección de paso.
> Se realiza un desplazamiento COI



Eliminar un fallo en la instalación: Reaccionar ante una estación de descarga vacía

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Conocer el desplazamiento de inicialización

5. El programa se puede continuar en el modo automático.



6.8 Conocer las teclas de ejecución del programa "Inicio hacia delante" e "Inicio hacia atrás"

Descripción

Para iniciar un programa se dispone de las teclas de Inicio hacia delante

 y también Inicio hacia atrás .



Fig. 6-8: Dirección de avance del programa: Adelante/atrás

Solo es posible ejecutar un programa hacia atrás con la tecla , si se ha ejecutado la línea INI de un programa y el robot se ha desplazado inmediatamente antes hacia delante por la trayectoria que debe recorrerse hacia atrás.

Sin embargo, si se desplaza manualmente el robot, p. ej., tras una parada del programa, el desplazamiento hacia atrás ya no es posible.

A partir de las versiones de software KSS 8.5 y superiores está disponible el "tipo de desplazamiento manual" separado Guía. En este caso, el desplazamiento se realiza por la trayectoria de manera óptica/física, pero se ha abandonado la trayectoria programada.



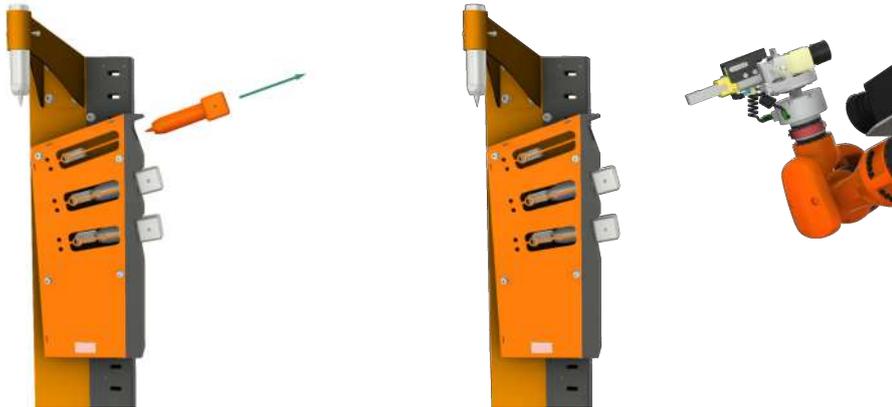
A partir de las versiones de software KSS 8.5 y superiores está disponible el "tipo de desplazamiento manual" separado Guía.

6.9 Ejercicio: Reaccionar ante el fallo de un cargador vacío

¡Ahora es su turno!



Enunciado



1. Retire la clavija superior del depósito de clavijas de la celda de formación.
2. Inicie el **programa:** _____ en modo automático.
3. El robot se detiene en la posición de agarre.
4. Cambie al modo de servicio "T1".
5. Realice el desplazamiento hasta la posición anterior al punto de agarre con la tecla de ejecución del programa "Arranque hacia atrás".
6. Introduzca de nuevo la clavija en el cargador de clavijas.
7. Cambie de nuevo al modo de servicio "AUT" y deje que el programa siga ejecutándose.

6.10 Preguntas: Reaccionar ante el fallo de un cargador vacío

Lo que se debe saber tras el ejercicio:

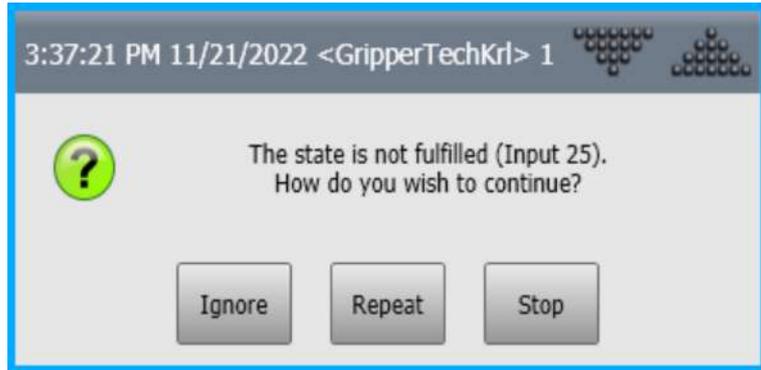


Fig. 6-9: Mensaje de diálogo

¿Qué reacción espera respectivamente al pulsar los tres botones Ignorar, Repetir y Parar?

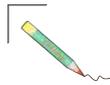


Fig. 6-10: Campo de respuesta

¿A qué posición se desplaza el robot en un desplazamiento COI?

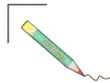


Fig. 6-11: Campo de respuesta

Eliminar un fallo en la instalación: Reaccionar ante una estación de descarga vacía

¿Con qué trayectoria del movimiento se desplaza el robot al conjunto de movimientos seleccionado?

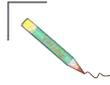


Fig. 6-12: Campo de respuesta

7 Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

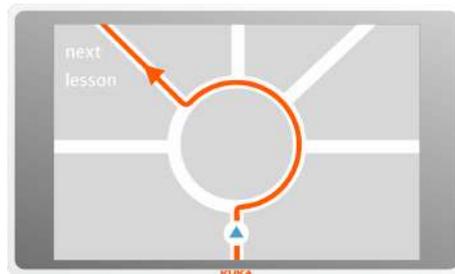
7.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar el robot teniendo en cuenta las prescripciones de seguridad.
- Está capacitado para seleccionar programas y probarlos en los modos de servicio "T1" y "T2"
- Está capacitado para iniciar los programas en el modo de servicio "Automático".
- Sabe cómo liberar el robot de una situación de avería.

¿Qué le espera en esta unidad de aprendizaje?



Eliminar un fallo en la instalación

- Aprenderá a liberar el robot de una situación de avería en el sistema de coordenadas universales.
- Aprenderá a resetear programas.

7.2 Reaccionar ante un encargo de proceso deficiente

Situación

Su empresa posee un robot que aplica material de obturación. El proceso se ha interrumpido por un motivo imprevisible. El proceso solo se puede reiniciar después de la extracción de la pieza de trabajo. Además, el robot debe liberarse de la estación de procesamiento y el programa debe reiniciarse.

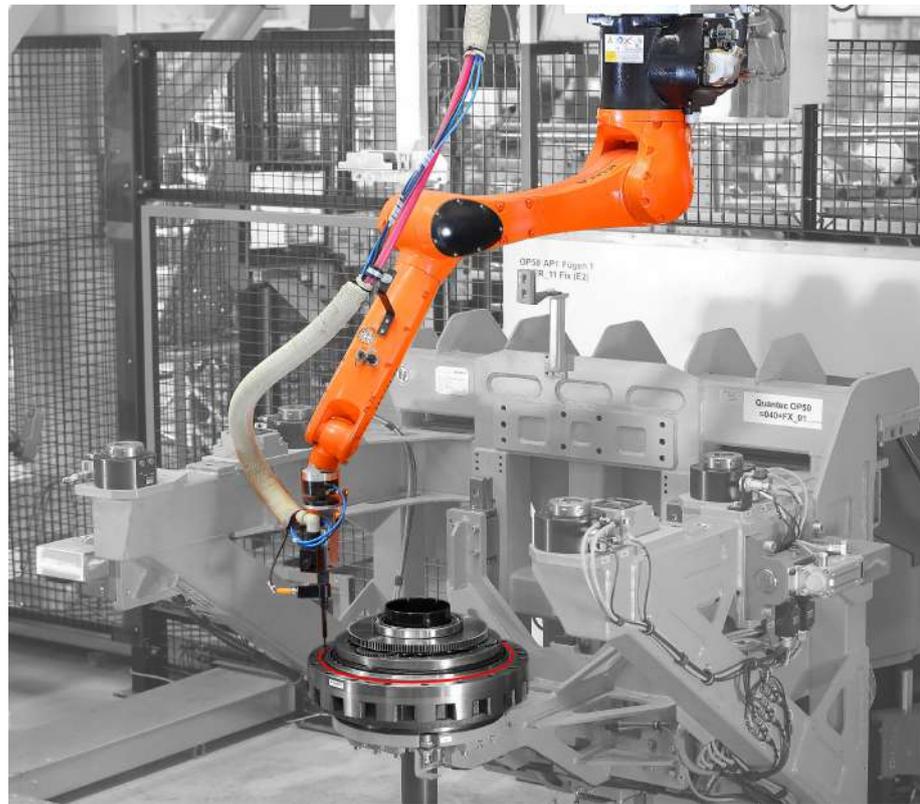


Fig. 7-1: Aplicación de pegamento

7.3 ¿Qué sucede si el robot se detiene durante una aplicación de trayectoria?

Descripción

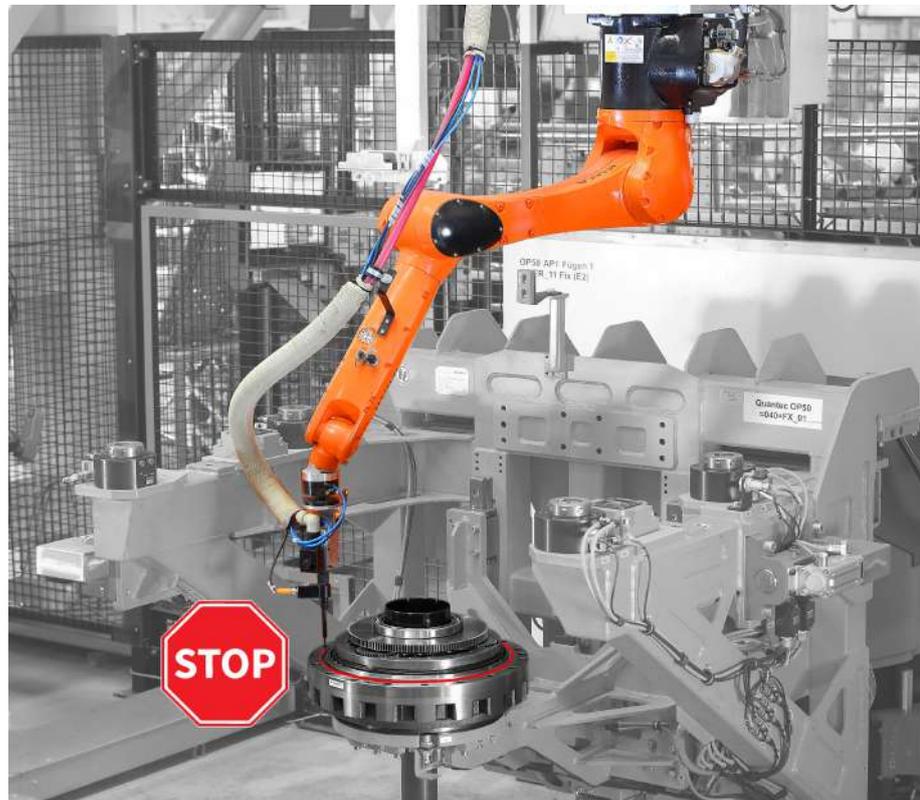


Fig. 7-2: Parada repentina

El proceso se detiene repentinamente, por lo que deja de tener lugar la aplicación del material. El verdadero problema debe resolverse. A continuación, el robot puede volver a liberarse y el programa puede restablecerse. En algunos casos, no es necesario restablecer el programa.

¿Por qué el robot no sigue trabajando?

El control de los materiales del proceso provoca la parada del robot.

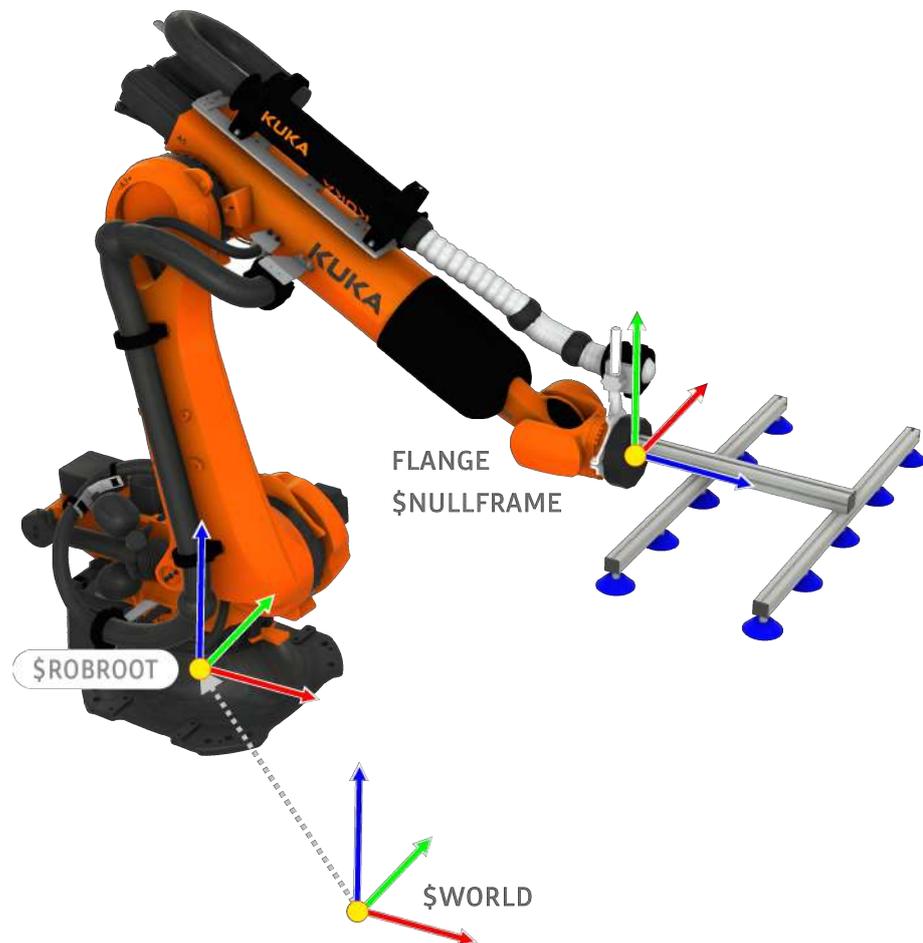
El encargo de proceso debe detenerse inmediatamente en caso de parada.

El procesamiento de componentes no se puede finalizar correctamente.

7.4 ¿Cómo se puede reaccionar ante un encargo de proceso erróneo?

Procedimiento

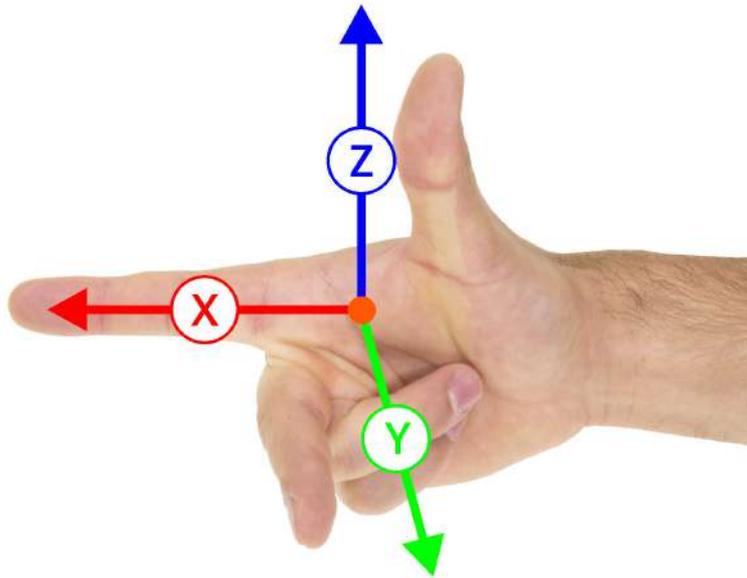
1. Liberar el robot en el sistema de coordenadas universales



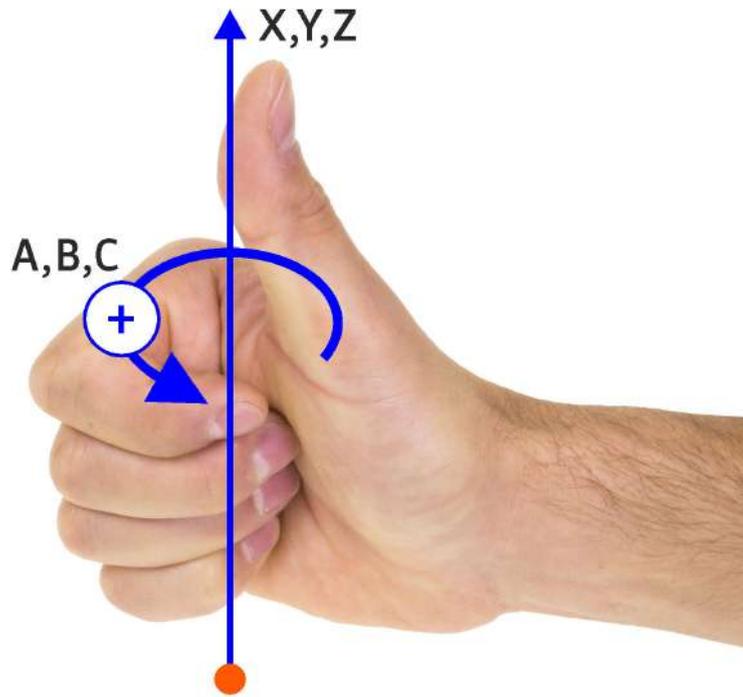
Para ello, activar el desplazamiento manual en el sistema de coordenadas universales en el smartPAD.



Como ayuda para la **orientación** utilizar la mano derecha.



Como ayuda para el **giro** utilizar la mano derecha.



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Mover el robot en el sistema de coordenadas universales con el smartPAD

Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

2. Resetear el programa



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Ejecutar programas de robot >
Resetear y deseleccionar el programa

3. Resetear y reiniciar el programa



7.5 Ejercicio: Reaccionar ante un encargo de proceso deficiente

¡Ahora es su turno!



Enunciado

- Seleccione el **programa**:_____.
- Inicie el programa en modo automático.
- Detenga el programa en diferentes puntos con la tecla de parada.
- Después de cada parada, libere el robot con los métodos aprendidos.
- Continúe con el programa.

Lo que se debe saber tras el ejercicio:



¿A dónde se desplaza el robot si después del desplazamiento manual se sigue ejecutando un programa mediante la tecla de inicio hacia delante?

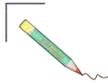


Fig. 7-3: Campo de respuesta

¿A dónde se desplaza el robot después de que se haya restablecido un programa?

Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

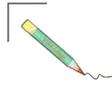


Fig. 7-4: Campo de respuesta

7.6 Movimiento del robot en el sistema de coordenadas universales

Uso del ratón 6D

Todos los tipos de movimiento se pueden realizar con el ratón 6D:

- **Translatorio:** pulsando y arrastrando el ratón 6D



Fig. 7-5: Ejemplo: Movimiento a la izquierda

- **Rotatorio:** girando y basculando el ratón 6D

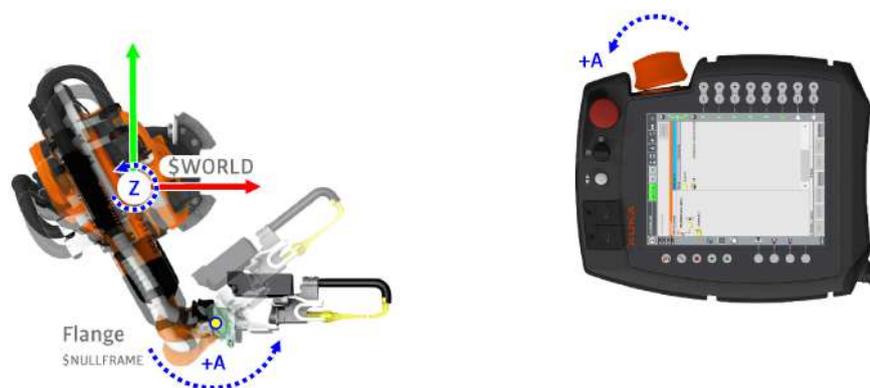


Fig. 7-6: Ejemplo: Movimiento rotacional en Z: Ángulo A

Posición del operario

La posición del ratón 6D puede adaptarse en función de la posición del operario con respecto al robot.



Fig. 7-7: Ratón 6D, posiciones

7.7 Ejercicio: Liberar el robot con el ratón 6D

¡Ahora es su turno!



Enunciado

- Seleccione el **programa**:_____.
- Inicie el programa en modo automático.
- Detenga el programa en diferentes puntos con la tecla de parada.
- Después de cada parada, libere el robot con el ratón 6D.
- Continúe con el programa.

Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

Eliminar un fallo en la instalación: Reaccionar ante un encargo de proceso deficiente

8 Guardar y restaurar archivos del robot

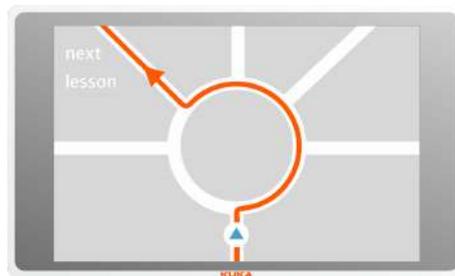
8.1 Navegador del curso

¿Qué requisitos aporta usted?



- Tiene conocimientos básicos sobre el manejo de un robot KUKA.
- Ha adquirido los primeros conocimientos con el smartPAD como unidad de operación.

¿Qué le espera en esta unidad de aprendizaje?



Guardar la imagen del disco duro y los datos

- Puede crear y restaurar copias de seguridad a través de KUKA.RecoveryUSB.
- Aprenderá a crear y restaurar archivos.
- Aprenderá a renombrar, duplicar y borrar programas.

8.2 Descripción de la situación

Situación

Recibe el encargo de optimizar una aplicación de robot.

¿Qué cree que se debe hacer haga antes de comenzar el trabajo?

Exacto, el estado del sistema actual debe guardarse.



8.3 ¿Por qué es necesario guardar los datos?

Una SSD corrupta se puede restaurar por completo mediante una imagen y un archivo existentes.

Durante las tareas de programación, el progreso de programación debe archivarse en intervalos regulares.

Una copia de seguridad guarda los paquetes de opciones de forma adicional al archivo. En caso de error, nuestro servicio de atención al cliente puede ayudarle lo antes posible.



Fig. 8-1: SSD M.2

8.4 ¿Cómo se puede crear una imagen y archivar datos?



Las imágenes del software pueden instalarse o guardarse con la memoria KUKA.RecoveryUSB en unidades de control de KUKA.

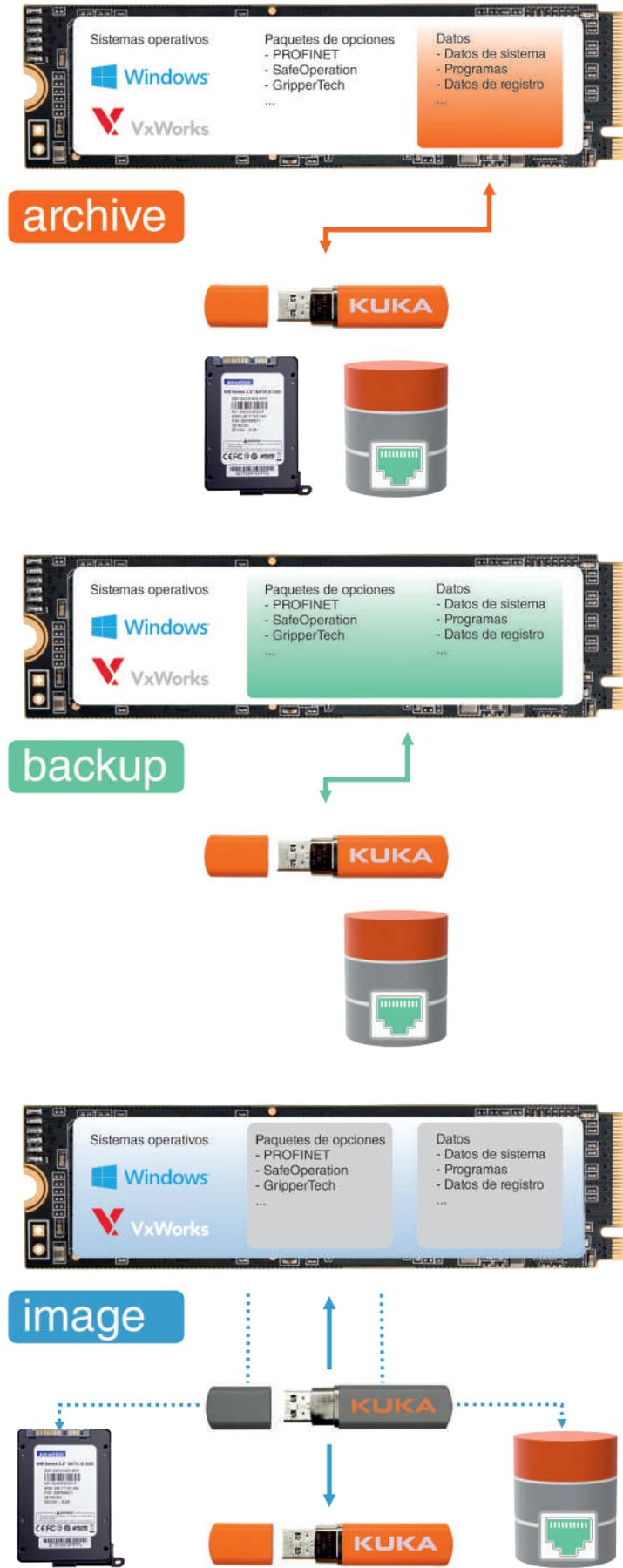
KUKA.Recovery es un software de archivado basado en WinPE.

Pueden preajustarse diferentes modos, como p. ej. la copia de seguridad automática de una imagen tras el arranque de la unidad de control con la memoria KUKA.RecoveryUSB conectada.

Un archivo puede guardarse en una memoria USB convencional o a través de la red.

¿En qué se diferencian el archivo, la copia de seguridad y la imagen entre sí?

Guardar y restaurar archivos del robot



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro	Programación de robots 1 KUKA System Software 8.6 Referencia de formación
Capítulo/apartado	Guardar la imagen del disco duro y los datos

Podrá encontrar más información en la documentación del cliente.



Libro	KUKA System Technology KUKA.Recovery 4.0 Para KR C4, KR C5 y KUKA Sunrise Cabinet Instrucciones de servicio y programación para los integradores de sistemas
Filtro de búsqueda	Documentación > Instrucciones de servicio
Capítulo/apartado	---

8.5 Ejercicio: Crear una copia de seguridad de datos mediante archivo del robot

Enunciado



Utilice exclusivamente la memoria USB asignada por el formador.

- Cree un archivo del robot con ayuda de una memoria USB.
- Después de crear el archivo, borre el **programa "_____"** de la unidad de control.
- Restablezca el archivo del robot con ayuda de la memoria USB.
- Compruebe si el **programa "_____"** está de nuevo disponible en la unidad de control.

Lo que se debe saber tras el ejercicio:



¿Cómo se pueden guardar los paquetes de opciones?

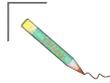


Fig. 8-2: Campo de respuesta

9 Corregir las posiciones del programa en el sistema de coordenadas base

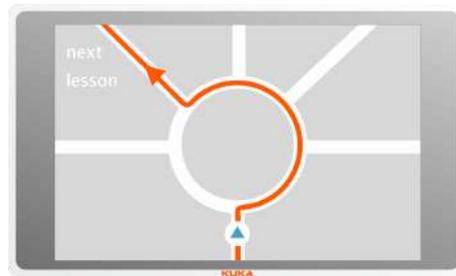
9.1 Navegador del curso

¿Qué requisitos aporta usted?



- Ha aprendido a manejar el robot con el smartPAD.
- Sabe cómo seleccionar programas y cómo probarlos en los modos de servicio T1, T2 y Automático.

¿Qué le espera en esta unidad de aprendizaje?



Corregir las posiciones del programa en el sistema de coordenadas "Base"

- Conocerá los tipos de movimiento SPTP, SLIN y SCIRC.
- Moverá el robot en el sistema de coordenadas "Base".
- Orientará su herramienta mediante el tipo de desplazamiento "Alinear".
- Puede corregir los puntos en los programas existentes con la función "Touch-Up".
- El "desplazamiento manual incremental" le ayudará en la corrección de puntos.

9.2 Descripción de la situación

Situación

Su empresa utiliza un robot para el pegado de componentes. La aplicación de pegamento se realiza a través de una boquilla de aplicación de pegamento. El programa es ejecutable, pero el contorno de pegado no se alcanza exactamente en todos los puntos. El contorno de pegado debe corregirse ahora de forma puntual.

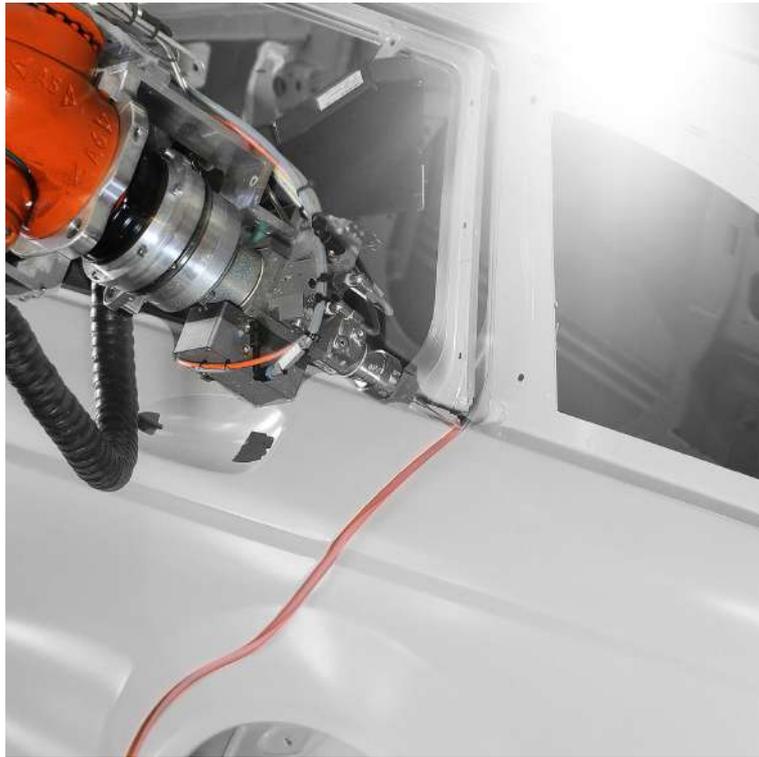


Fig. 9-1: Pegar y sellar

Aplicación de pegamento con errores

En la célula de formación College, la clavija simula la boquilla de aplicación de pegamento. La clavija está sujeta en la garra. La trayectoria de aplicación de pegamento corresponde al contorno exterior de la lámina de ejercicio.

La trayectoria debe corregirse.



Fig. 9-2: Aplicación de pegamento con errores

9.3 ¿Qué es necesario para una trayectoria de pegamento perfecta?

Procedimiento



1. Seleccione el programa de aplicación de pegamento.
2. Al ejecutar el programa, compruebe si el curso de trayectoria se encuentra dentro de las tolerancias.
3. Delimite los puntos de programa o los movimientos que difieran.
4. Corrija los puntos y compruebe de nuevo el curso de trayectoria.

9.4 Conocer los modos de flujo de programa

Descripción

Para ejecutar en un programa existen varios **modos de flujo de programa** para el desarrollo del movimiento controlado por programa del robot.



Fig. 9-3: Tipos de flujo de programa, smartPAD

	<p>GO</p> <ul style="list-style-type: none"> • El programa se ejecuta de forma continuada hasta finalizar. • En el modo de prueba se debe mantener pulsada la tecla de arranque.
	<p>Movimiento</p> <ul style="list-style-type: none"> • Cada instrucción de movimiento se ejecuta de forma individual. • Después de finalizar un movimiento se debe pulsar otra vez "Inicio" respectivamente.

	<p>Paso a paso Disponible únicamente en el grupo de usuario "Experto"</p> <ul style="list-style-type: none"> • Se ejecuta línea a línea (independientemente del contenido de la línea). • Después de cada línea se debe volver a accionar la tecla de arranque.
---	--

9.5 SPTP – Desarrollos del movimiento con optimización del tiempo de ciclo (movimiento de eje)

Descripción

SPTP – *Spline Point To Point* – es un desarrollo del movimiento específico del eje. El robot desplaza el TCP de la herramienta al punto de destino con el movimiento **más rápido**. La trayectoria más rápida no es, por lo general, la trayectoria más corta y, por tanto, **no** 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. El **primer desarrollo del movimiento en el programa** debe ser un **desarrollo del movimiento SPTP**, ya que aquí únicamente se van a evaluar Status y Turn (>>> [16.1.1 "Status y Turn" Página 213](#)). No puede predecirse la trayectoria exacta después de la programación por aprendizaje de los puntos. Sin embargo, una vez que el robot se ha movido, repetirá el movimiento de forma idéntica una y otra vez.

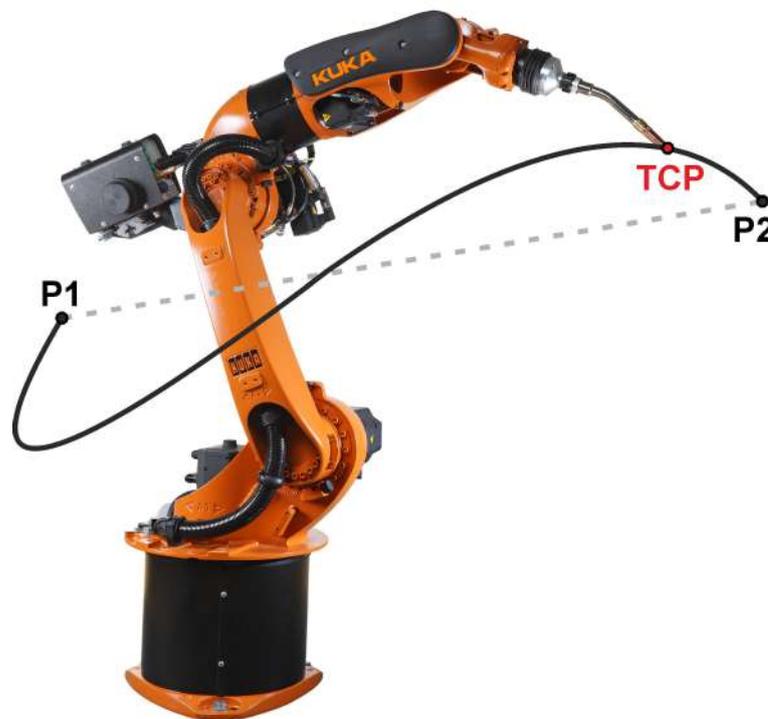


Fig. 9-4: Movimiento SPTP

Synchro PTP

Todos los ejes arrancan juntos con Synchro-PTP y se detienen al mismo tiempo. Para ello se tienen en cuenta la indicación de la velocidad en el formulario inline. El eje directriz es el eje que **más** tarda en alcanzar el punto de destino.

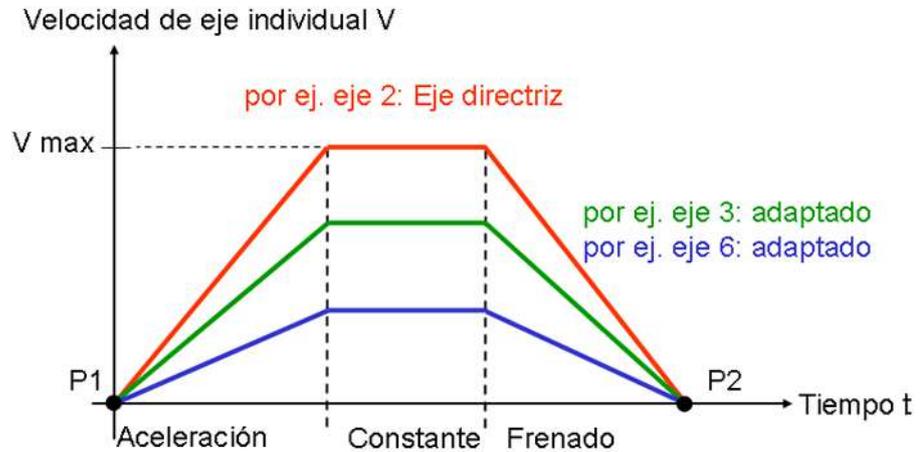


Fig. 9-5: Synchro-PTP

Uso

El desarrollo del movimiento SPTP se utiliza cuando el robot debe aproximarse a los puntos lo más rápidamente posible. La trayectoria desempeña un papel secundario. Para garantizar un desarrollo del movimiento sin colisiones, el robot puede ser "guiado" hacia el objetivo mediante puntos intermedios aproximados. También el desplazamiento del robot hacia o entre un movimiento de trayectoria puede optimizarse en tiempo de ciclo por medio de movimientos SPTP aproximados.



Fig. 9-6: Soldadura por puntos

Ejemplos

- Soldadura por puntos y remaches
- Transportar
- Medición/comprobación

9.6 Crear movimientos de trayectoria

Trayectorias de los movimientos SLIN y SCIRC

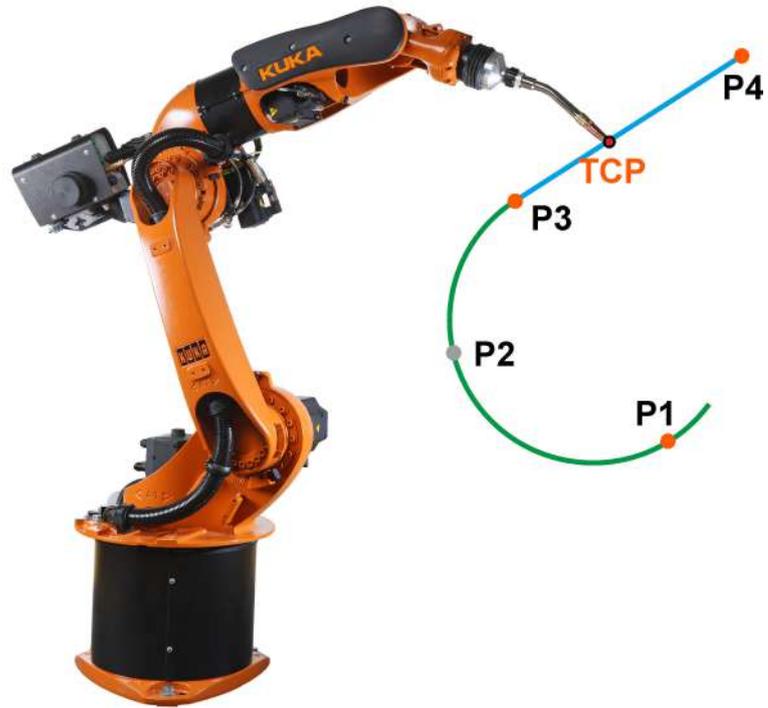
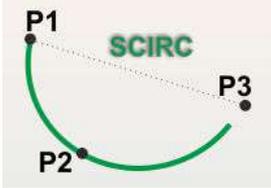


Fig. 9-7: Movimientos de trayectoria

Trayectoria del movimiento	Significado	Ejemplo de aplicación
	<p>SLIN: Lineal</p> <ul style="list-style-type: none"> • Movimiento de trayectoria rectilíneo • El TCP de la herramienta se desplaza desde el inicio hasta el punto de destino con una velocidad constante y una orientación definida. • La velocidad y la orientación hacen referencia al TCP. 	<p>Aplicaciones de trayectoria, p. ej.:</p> <ul style="list-style-type: none"> • Soldadura de trayectoria • Pegado • Soldadura y corte láser
	<p>SCIRC: Circular</p> <ul style="list-style-type: none"> • El movimiento de trayectoria circular se define a través del punto de inicio, el punto auxiliar y el punto de destino. • El TCP de la herramienta se desplaza desde el inicio hasta el punto de destino con una velocidad constante y una orientación definida. • La velocidad y la orientación hacen referencia al TCP de la herramienta. 	<p>Aplicaciones de trayectoria como en el SLIN:</p> <ul style="list-style-type: none"> • Círculos, radios, curvaturas

Corregir las posiciones del programa en el sistema de coordenadas base

9.7 El sistema de coordenadas Base

Descripción

Medir una base significa la definición de un sistema de coordenadas en el campo de trabajo del robot en relación con el sistema de coordenadas universal. El objetivo es que las posiciones programadas del robot se refieran a este sistema de coordenadas y que se facilite el desplazamiento manual. Para ello se utilizan, p. ej. los bordes definidos de alojamientos de piezas, compartimentos, palets o máquinas, como puntos de referencia muy útiles para un sistema de coordenadas base.



Fig. 9-8: Base, uso

9.8 Conocer el desplazamiento manual incremental

Movimiento manual incremental

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

La tecla de desplazamiento correspondiente debe permanecer pulsada hasta que se alcance la distancia preestablecida. Una vez alcanzada la distancia predefinida, el robot se detiene por sí mismo.

En el desplazamiento con el ratón Space, no es posible aplicar el desplazamiento manual incremental.

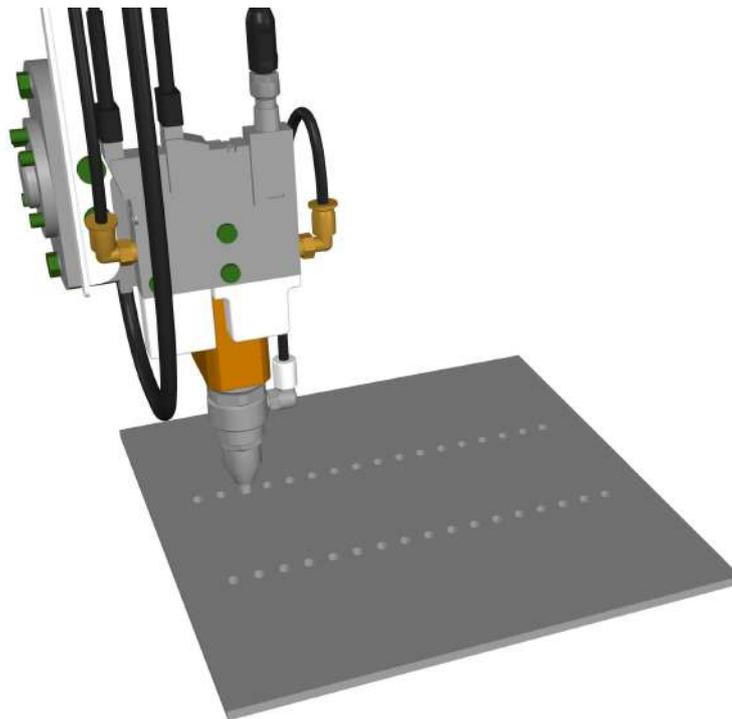


Fig. 9-9: Desplazamiento manual incremental, ejemplo

Ámbitos de aplicación

- Posicionamiento de puntos a distancias iguales.
- Corrección de punto exacta cuando la distancia se especifica en milímetros o en grados, p. ej., datos de un dispositivo de medición
- Movimiento de salida desde una posición con una distancia definida, p. ej., en caso de error

9.8.1 Desplazamiento incremental del robot

Procedimiento

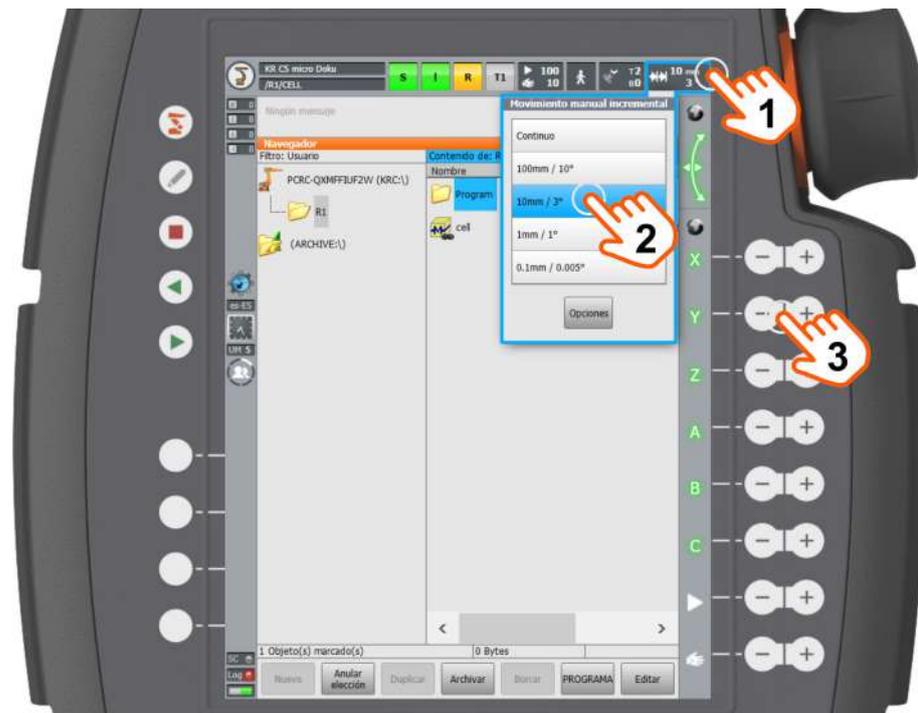


Fig. 9-10: Movimiento manual incremental

1. En la barra de estado, pulsar el campo "Desplazamiento manual incremental" (1).
2. Seleccionar la distancia de desplazamiento incremental deseada (2).
3. Pulsar y mantener pulsado uno de los cuatro tastos de validación.
4. El robot se puede desplazar con las teclas de desplazamiento de forma cartesiana o de forma específica por ejes de acuerdo con la distancia de desplazamiento ajustada.

En el ejemplo, esta se refiere al TCP de la herramienta activa

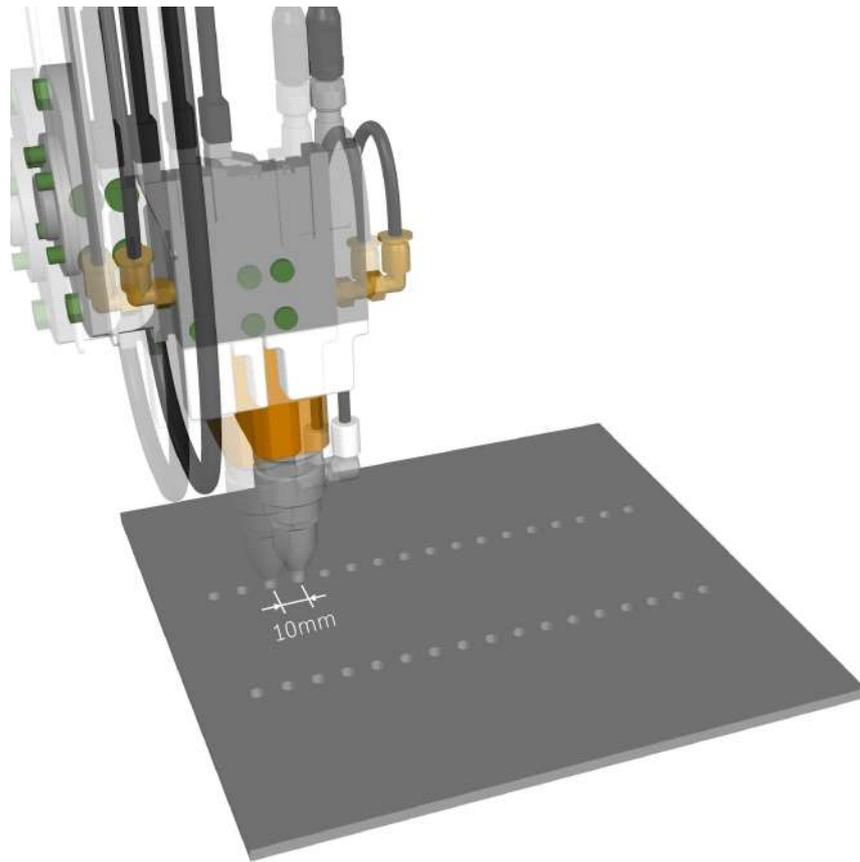


Fig. 9-11: Ejemplo, chapa perforada



Cuando se interrumpe el movimiento del robot como, por ej., al soltar el pulsador de validación, en el próximo movimiento no se continúa con la distancia de desplazamiento interrumpida, sino que se comienza de nuevo.

Ajustes/valor del incremento:

Ajuste	Descripción
Continuo	El desplazamiento manual incremental está desconectado.
100 mm / 10°	1 incremento = 100 mm o 10°
10mm / 3°	1 incremento = 10 mm o 3°
1mm / 1°	1 incremento = 1 mm o 1°
0,1mm / 0,005°	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 rotación A, B o C.
- Válido para movimientos específicos del eje.

9.9 Mover el robot mediante el tipo de desplazamiento "Alinear" (referido a la base)

Descripción

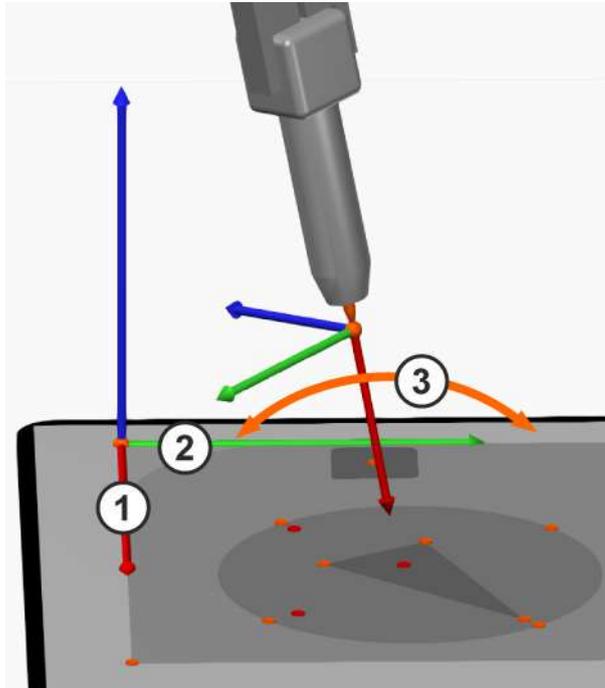
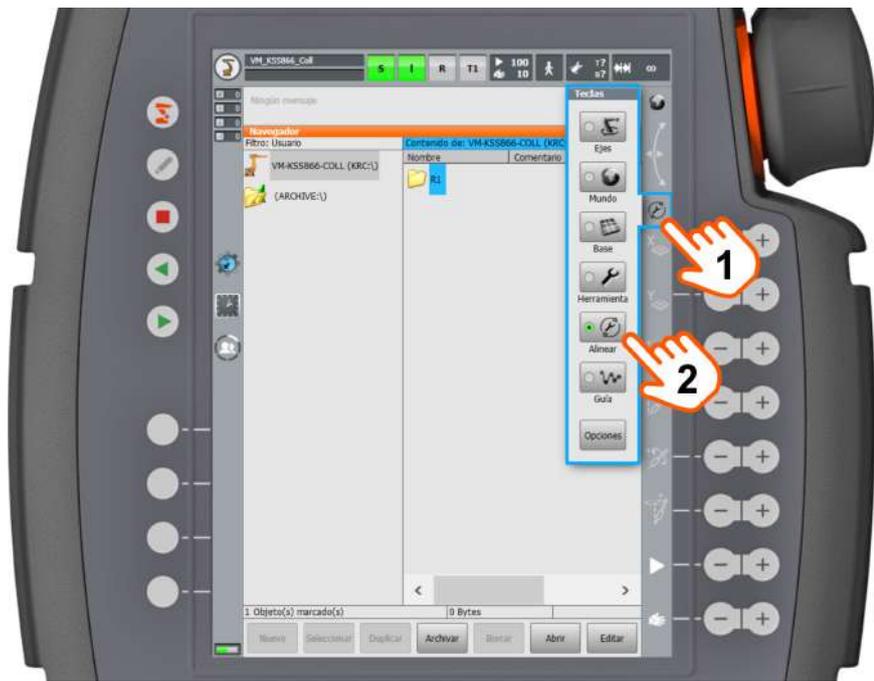


Fig. 9-12: Tipo de desplazamiento "Alinear", referido a la base

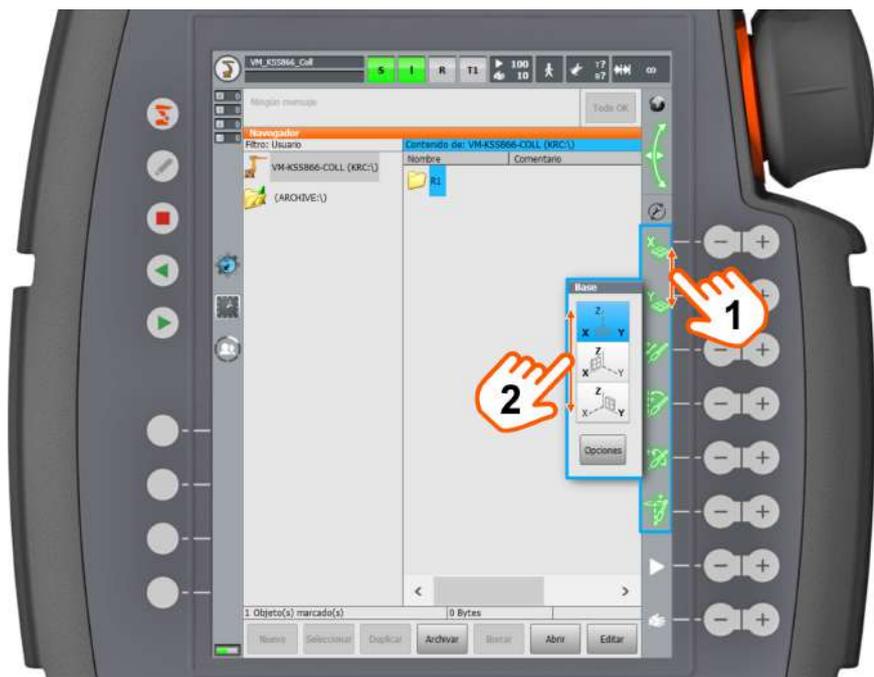
1,2	Desplazamiento de la herramienta en paralelo al plano de base activo, que se encuentra perpendicular a la alineación. En el ejemplo X (1) o Y (2).
3	Alinear la herramienta en perpendicular a la base activa.

Procedimiento

1. A la izquierda de las teclas de desplazamiento, pulsar el símbolo "Tipo de desplazamiento activo" (1).
En el menú contextual "Teclas", utilice el símbolo "Alinear" (2) para activar la alineación específica de la herramienta.



2. Los símbolos superiores X e Y (1) permiten reorientar la herramienta mediante la base activa. El nivel al que se debe hacer referencia puede activarse en otro menú contextual (2).



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro	Programación de robots 1 KUKA System Software 8.6 Referencia de formación
Capítulo/apartado	Alinear la herramienta del robot mediante el sistema de coordenadas TOOL y BASE

9.10 Corregir las coordenadas de puntos de movimiento

Procedimiento

1. El programa está seleccionado y se ha ejecutado la línea INI o se ha llevado a cabo el desplazamiento COI (1).

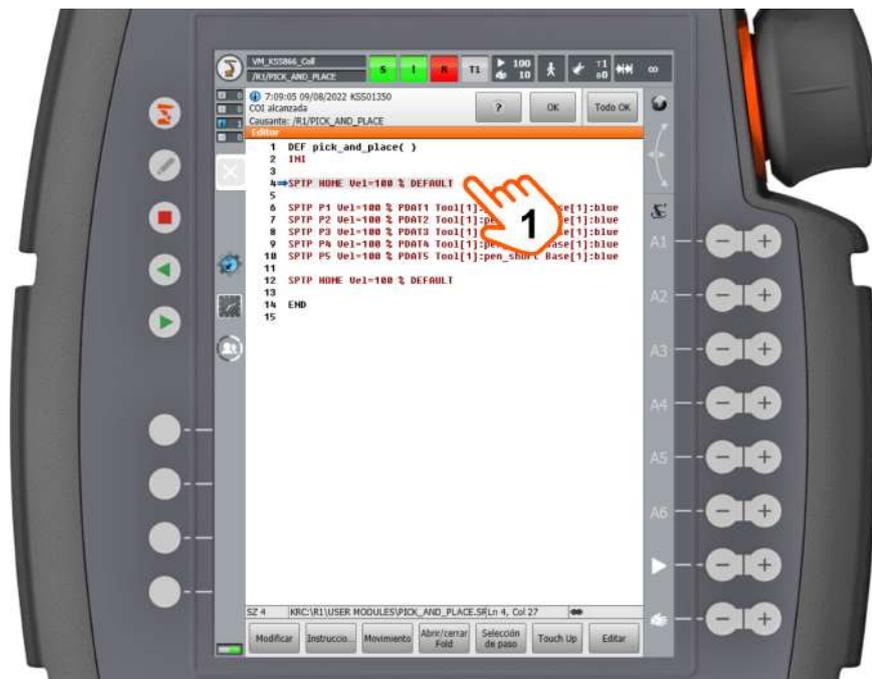


Fig. 9-13: Corregir puntos, paso 1

2. Pulsar la línea del programa que debe modificarse (2). Se muestra un cursor intermitente.

Corregir las posiciones del programa en el sistema de coordenadas base



Fig. 9-14: Corregir puntos, paso 2



Dependiendo del lugar en el que se haya pulsado en la línea, también puede aparecer el cursor dentro de la línea del programa.

- Desplazar el robot a la posición deseada con las teclas de desplazamiento o con el ratón 6-D.

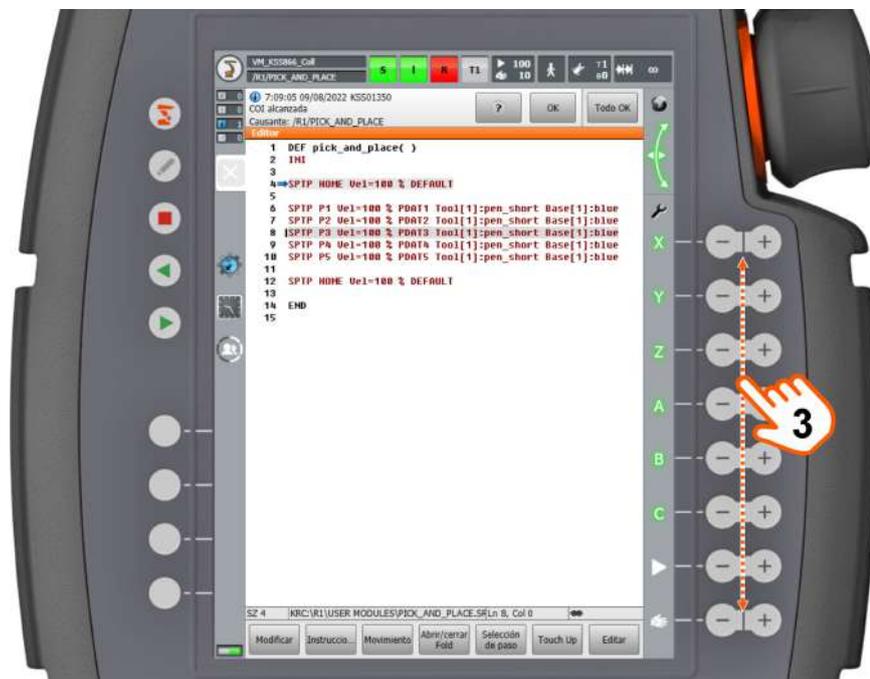


Fig. 9-15: Corregir puntos, paso 3



Consejo: Realizar la selección de paso en la línea de programa que debe modificarse mediante el botón del mismo nombre. El puntero de ejecución del programa azul salta a la posición de la línea. Realizar el desplazamiento hasta el punto y, a continuación, corregir los puntos.

4. Aceptar las nuevas coordenadas del punto en el punto marcado con el botón "Touch-Up" (4). Un diálogo correspondiente indica que el punto con este título ya existe. Confirmarlo con el botón "Sí" (5).

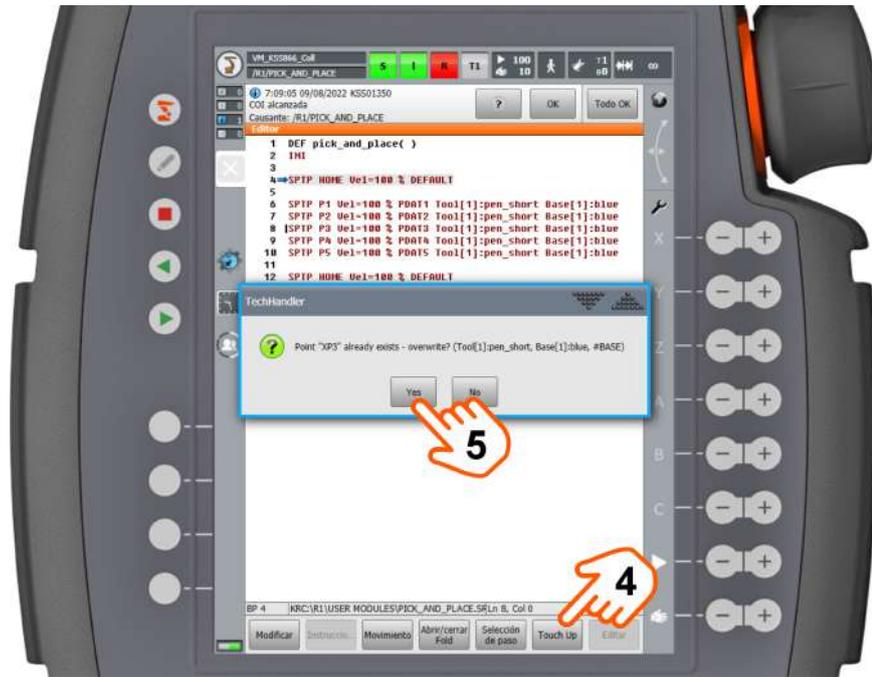


Fig. 9-16: Corregir puntos, paso 4,5

AVISO

Las coordenadas del punto se guardan siempre en la línea del programa en la que se encuentra el cursor (ver también el paso 2). La posición del puntero de paso (flecha azul) no tiene ninguna importancia aquí.

5. El puntero de paso salta a la posición de línea marcada. Las nuevas coordenadas del punto se han guardado.

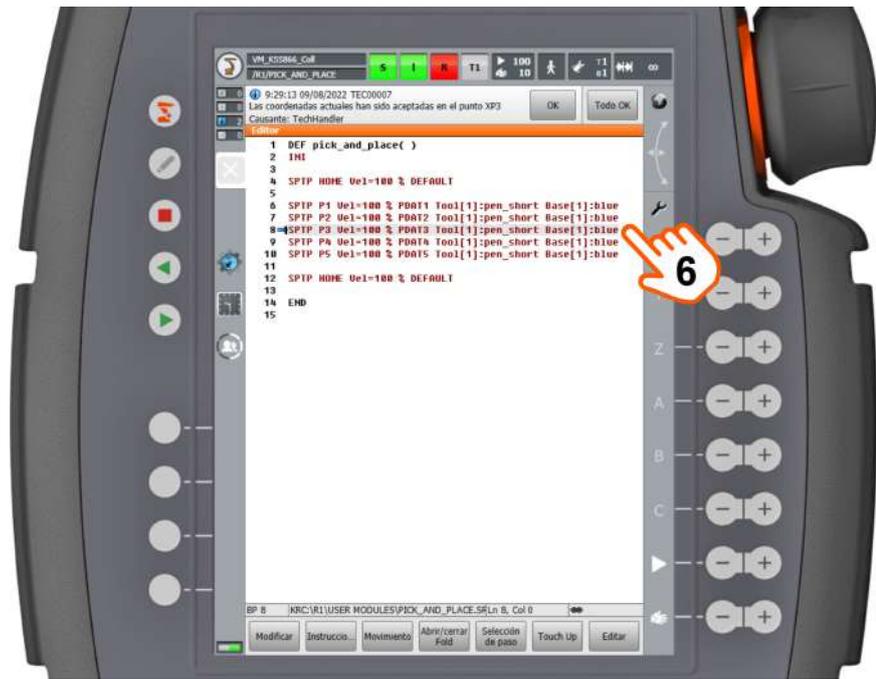


Fig. 9-17: Corregir puntos, paso 6

Efectos al modificar los datos de posición

El punto recibe coordenadas nuevas, porque con "Touch-Up" se han actualizado los valores. De este modo, solo se ha modificado el juego de datos del punto y el resto del programa permanece igual.

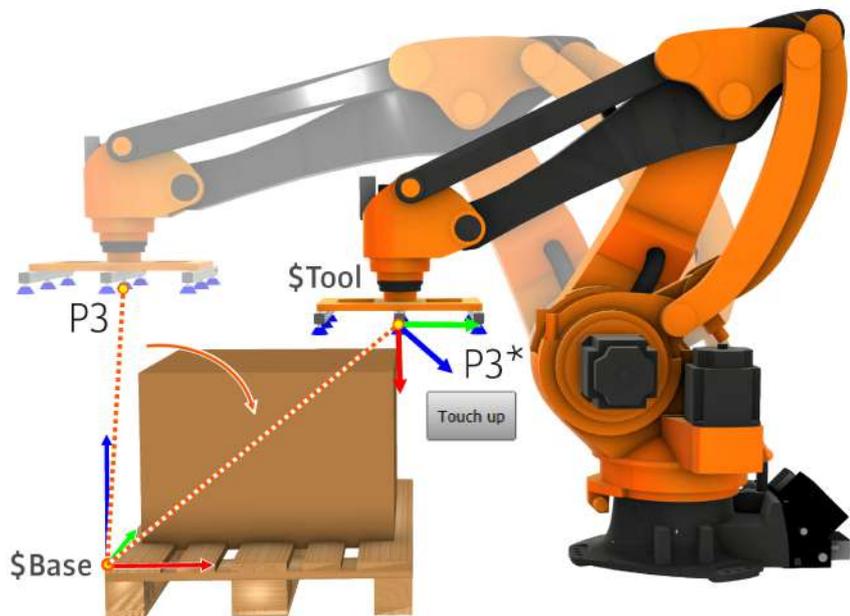


Fig. 9-18: Modificación de la posición del robot con "Touch-Up"

AVISO

Las coordenadas anteriores se sobrescriben y, por consiguiente, dejan de estar disponibles.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Modificar puntos de movimiento programados

9.11 Ejercicio: Reaccionar ante un encargo de proceso deficiente

¡Ahora es su turno!



Enunciado

1. Pruebe el **programa**: _____ en el modo T1.
2. Identifique los conjuntos de movimientos cuyas coordenadas deben corregirse.
3. Desplace estos conjuntos con el modo de flujo de programa "Desarrollo del movimiento" y corrija los puntos programados. La herramienta debe estar en posición vertical sobre la trayectoria programada. Preste especial atención al posicionamiento correcto del cursor negro en el smartPAD. Compruebe en el mensaje de diálogo si ha hecho clic en el conjunto de movimientos correcto.
4. Pruebe el programa de nuevo en T1 y AUT.

9.12 Preguntas: Reaccionar ante un encargo de proceso deficiente

Lo que se debe saber tras el ejercicio:



¿Qué sucede si pulsa el botón "Touch-Up"?

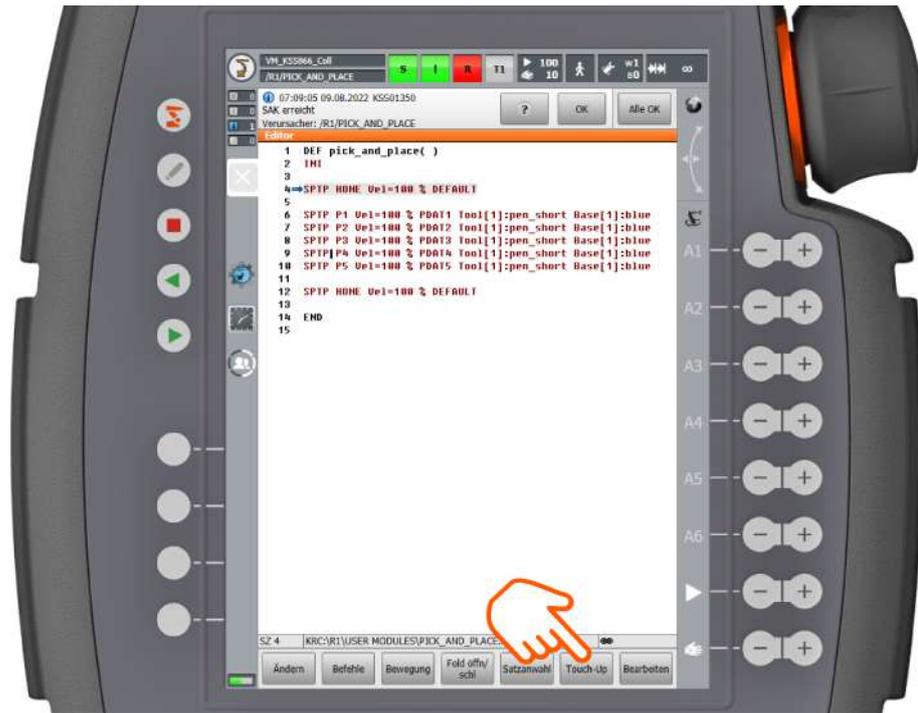


Fig. 9-19: Registrar coordenadas de puntos



Fig. 9-20: Campo de respuesta

¿Cómo se puede aproximar un punto para poder corregirlo a continuación?

Corregir las posiciones del programa en el sistema de coordenadas base



Fig. 9-21: Campo de respuesta

10 Corregir las posiciones de la garra en el sistema de coordenadas Tool

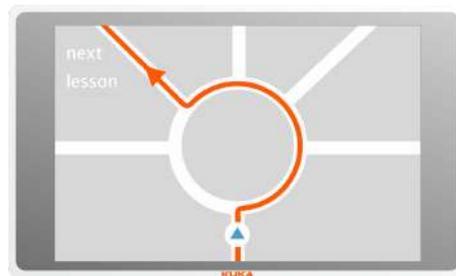
10.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar el robot con el smartPAD.
- Ha aprendido a desplazar el robot en el sistema de coordenadas Base.
- Sabe cómo corregir y modificar los movimientos del robot existentes.

¿Qué le espera en esta unidad de aprendizaje?



Corregirá las posiciones de la garra en el sistema de coordenadas "Tool"

- Puede manejar la garra mediante el paquete de opciones "KUKA.GripperSpotTech".
- Moverá el robot en el sistema de coordenadas Tool.
- Corregirá las posiciones de la garra en los formularios inline existentes.

10.2 Descripción de la situación

Situación

Usted dispone en su empresa de una célula de robot con cambiador de herramientas. Una pieza debe ser mecanizada con dos herramientas diferentes. Para la segunda herramienta ya tiene asignado un nuevo lugar de depósito en el cambiador de herramientas. Sin embargo, todavía tiene que crear la recogida y el depósito de la segunda herramienta. Esto se ejecutarse en dos subprogramas.



Programar una nueva posición de la herramienta

En el cargador de clavijas de KUKA-College hay disponibles tres posiciones de descarga diferentes para clavijas (herramientas). Las clavijas se diferencian en cuanto a la longitud y pueden ser recogidas y depositadas por el robot.



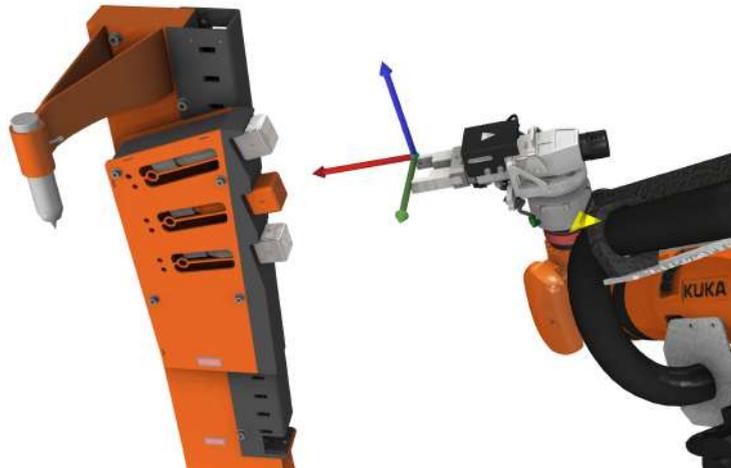
10.3 Agarre de la herramienta de clavija central

Procedimiento

1. Duplicar el programa "Tomar_clavija_L()". Asignar un nuevo nombre del programa: "Tomar_clavija_M()".



2. La posición de agarre, así como la posición anterior y posterior, deben adaptarse a la nueva posición de agarre y programarse posteriormente.

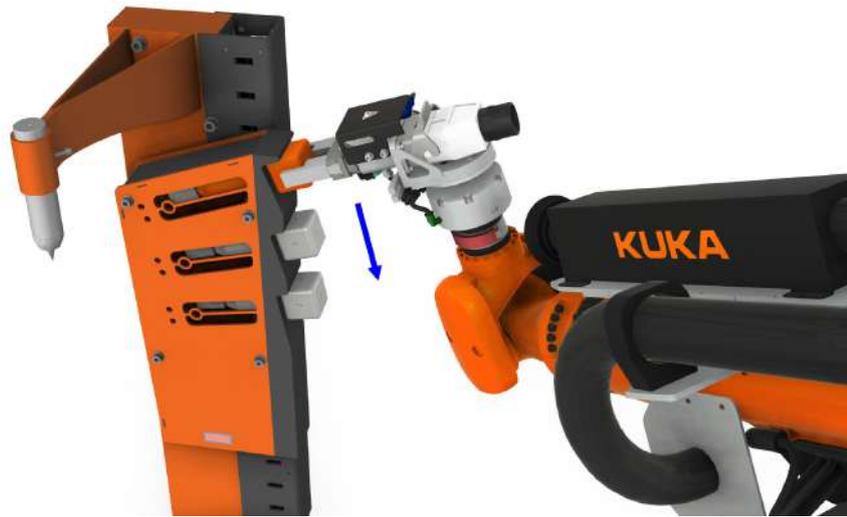


3. Como herramienta debe utilizarse la garra en el formulario inline.
4. El nuevo programa debe probarse.

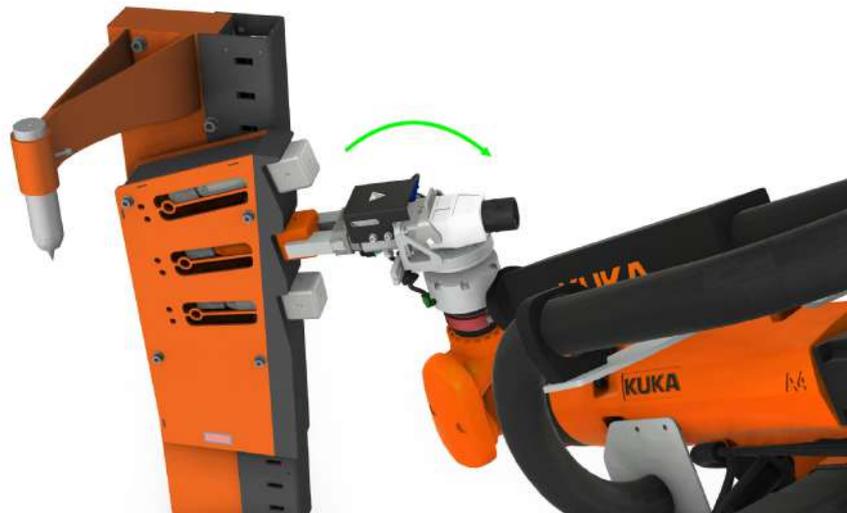
10.4 ¿Cómo puede modificarse la posición de la garra?

Procedimiento

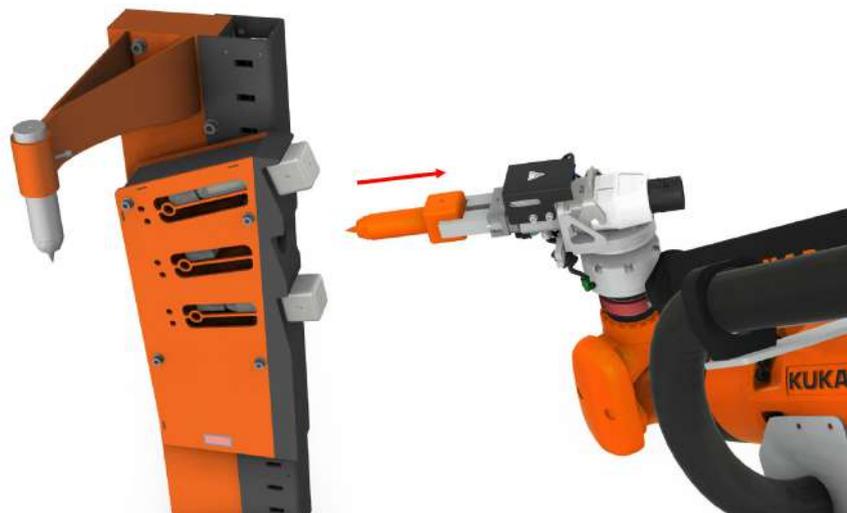
1. Ejecute el programa en el modo de flujo de programa "Desarrollo del movimiento" hasta que el robot haya alcanzado la "Posición previa".
2. Desplace el robot en el sistema de coordenadas de la herramienta con la garra abierta hasta la nueva posición de recogida.



3. Adapte la orientación de la herramienta con ayuda del sistema de coordenadas de la herramienta.
Acepte la nueva coordenada del punto en el formulario inline del punto de agarre.



4. Extraiga la clavija del cargador (sistema de coordenadas de la herramienta > dirección de impacto negativa).
Acepte la nueva coordenada del punto en el formulario inline del punto de agarre.



- Después de realizar la corrección, resetee el programa y compruebe en el modo de servicio T1.

10.5 Mover el robot en el sistema de coordenadas TOOL

Sistema de coordenadas Tool

Con el desplazamiento manual en el sistema de coordenadas TOOL o de herramientas, es posible mover el robot linealmente a lo largo de los ejes del sistema de coordenadas de una herramienta medida. El sistema de coordenadas no es fijo, como un sistema de coordenadas World o BASE, sino que es guiado por el robot. **Todos** los ejes del robot se mueven. El origen del sistema de coordenadas TOOL 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.

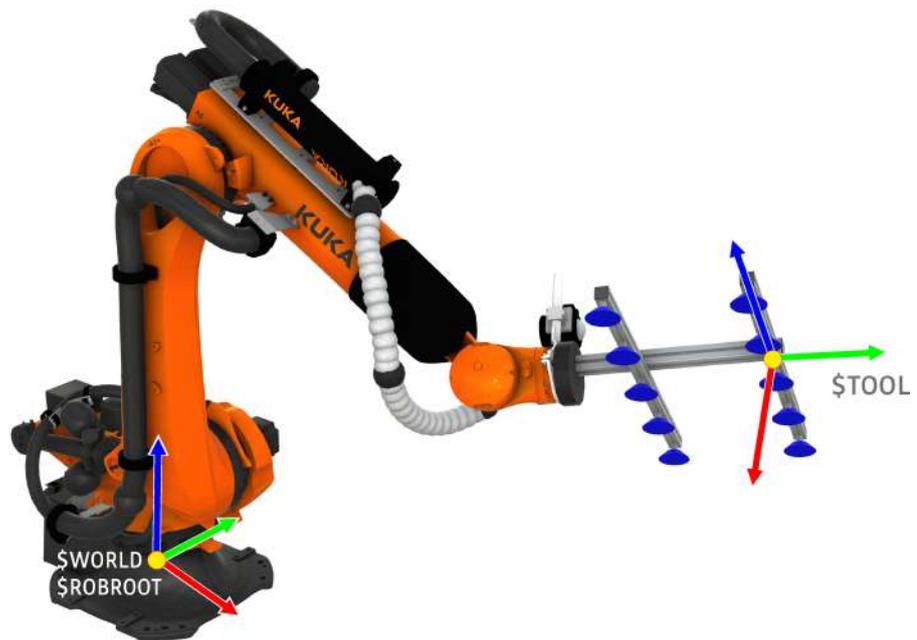


Fig. 10-1: Sistema de coordenadas de la herramienta del robot

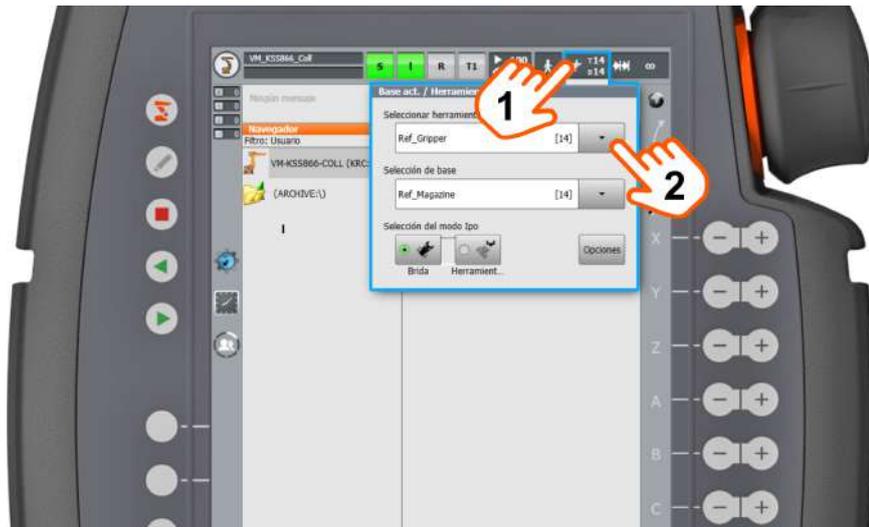
10.5.1 Mover el robot en el sistema de coordenadas de TOOL

Procedimiento

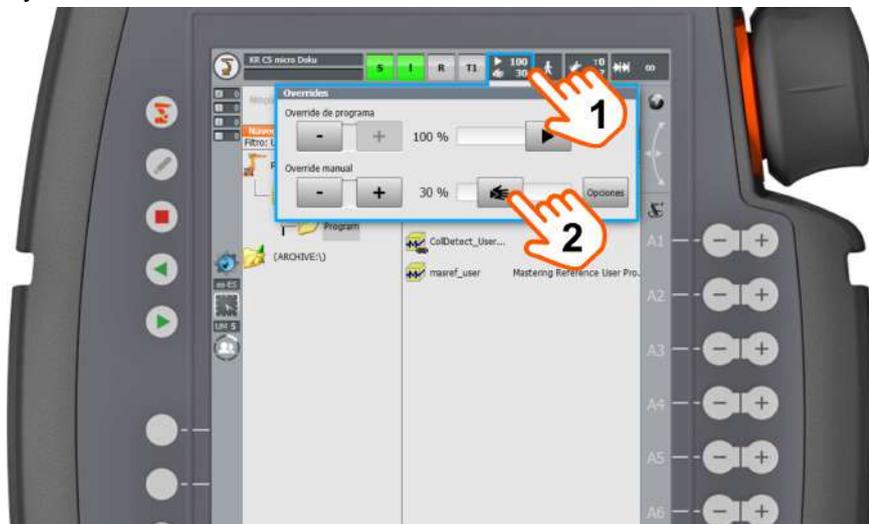
1. Seleccionar **Herramienta** como el sistema de coordenadas que debe utilizarse



2. Seleccionar el número de herramienta



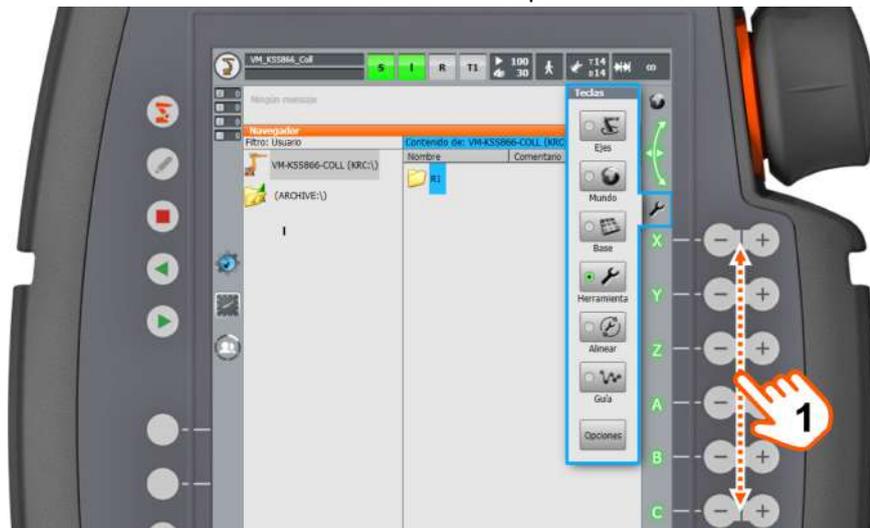
3. Ajustar el override manual



4. Pulsar y mantener pulsado uno de los cuatro pulsadores de validación en la posición central

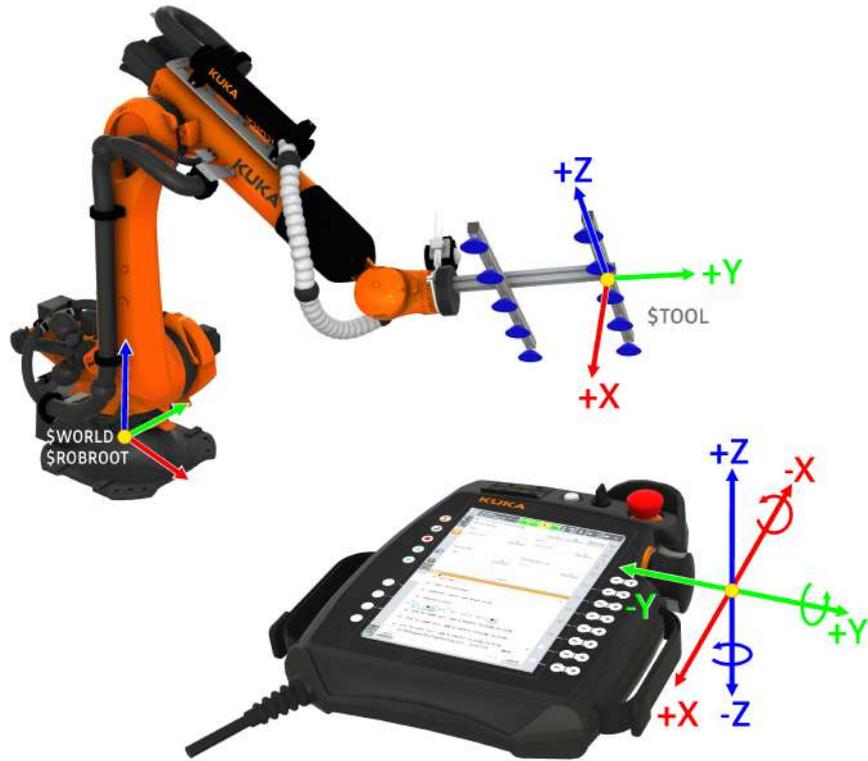


5. Mover el robot mediante las teclas de desplazamiento



Corregir las posiciones de la garra en el sistema de coordenadas Tool

6. Alternativa: Desplazamiento con el ratón 6D



10.6 Ejercicio: Modificar la posición de la garra en el cargador de clavijas

¡Ahora es su turno!



1. Duplique el **programa:**_____ para recoger la clavija larga.
Nuevo **nombre del programa:**_____
2. El programa debe extraer la clavija central del cargador de clavijas en lugar de la clavija larga y superior.
3. Guarde las nuevas posiciones en los formularios inline existentes para la posición de agarre y de avance.
4. A continuación, pruebe el nuevo programa.
5. Cree del mismo modo un **programa:**_____ para depositar la clavija.

Lo que se debe saber tras el ejercicio:



¿Qué dirección de las coordenadas en el sistema de coordenadas de la herramienta se denomina en el robot KUKA como dirección de impacto de la herramienta?



Fig. 10-2: Campo de respuesta

¿Cómo se comporta la punta de la herramienta durante el giro de la herramienta alrededor de los ejes de giro A, B y C?

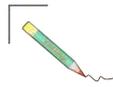


Fig. 10-3: Campo de respuesta

¿Cómo se puede desplazar el robot libremente fuera de la instalación en caso de un fallo?

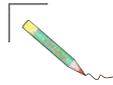


Fig. 10-4: Campo de respuesta

11 Configuración de una nueva estación de robot: Ajuste del robot

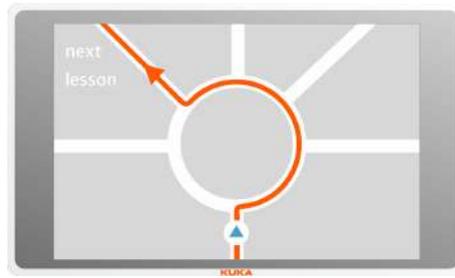
11.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar el robot con el smartPAD.
- Sabe cómo desplazar el robot manualmente de forma cartesiana.

¿Qué le espera en esta unidad de aprendizaje?



Ajuste del robot

- Entenderá por qué el robot debe ajustarse.
- Conocerá los diferentes medios de ajuste.
- Conocerá los tipos de ajuste "Ajuste estándar" y "Ajuste con corrección de peso".
- Aprenderá a desplazar el robot de forma específica del eje a la posición de preajuste.
- Ajustará el robot de acuerdo con el método especificado.

11.2 Conocer el desplazamiento manual específico del eje

Desplazamiento de los ejes del robot por medio de ejes específicos

Cada eje de un robot puede desplazarse individualmente en dirección positiva y negativa. Para ello se utilizan las teclas de desplazamiento del KUKA smartPAD. La velocidad de desplazamiento se reduce mediante el HOV (*Override manual*). El desplazamiento manual solo se puede efectuar en el modo de servicio T1. Para ello, uno de los pulsadores de validación debe mantenerse en la posición central.

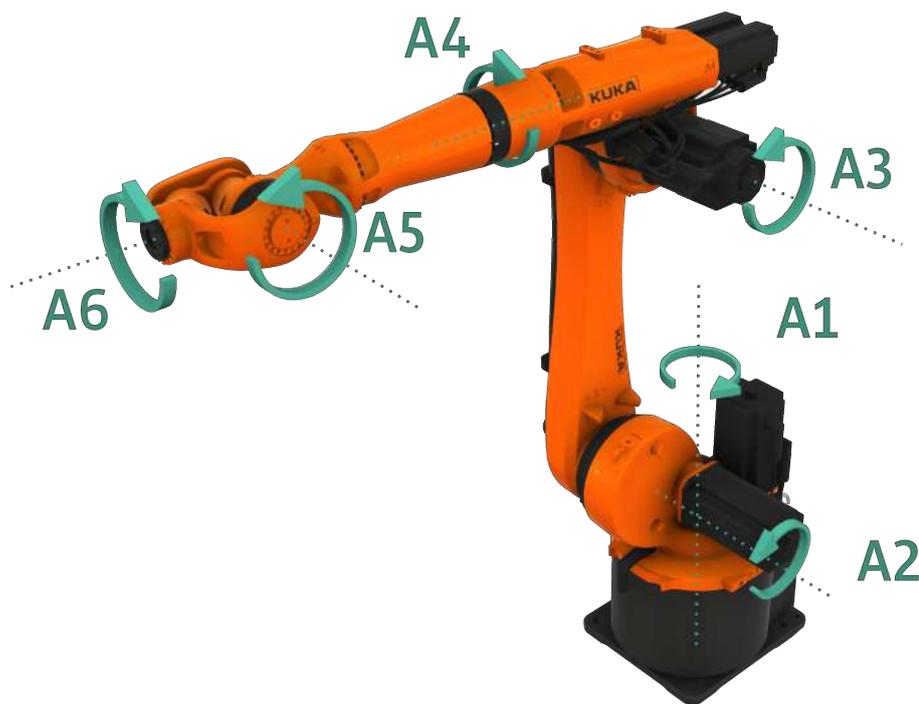


Fig. 11-1: Ejes de un robot KUKA



Las flechas de la ilustración apuntan en cada caso en el sentido positivo de la dirección de desplazamiento (+).

Principio

- Al pulsar uno de los pulsadores 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 ratón 6D, se inicia la regulación los ejes del robot y se ejecuta el desarrollo del 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).

Mensajes que influyen en el funcionamiento 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 confirmar los mensajes de la ventana de mensajes. Al pulsar un pulsador de validación se activan 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.

Consejo práctico

1. Seleccionar el programa deseado.
2. Accionar uno de los pulsadores de validación.

3. Esperar hasta que los accionamientos estén preparados (teclas de desplazamiento iluminadas en verde o indicación del accionamiento I en verde).
4. En ese momento, iniciar la acción deseada » Tecla de inicio, teclas de desplazamiento o ratón 6D.



Si no se respeta este orden, se visualizará el mensaje "Comandos activos bloqueados".

11.2.1 Desplazamiento del robot por medio de ejes específicos

Procedimiento



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Desplazamiento del robot por medio de ejes específicos

11.3 Ajuste del robot

Ha duplicado un nuevo programa en el módulo anterior para recoger la clavija central. Todas las nuevas coordenadas de posición se han guardado correctamente.

Sin embargo, durante la prueba del programa se produce una colisión cuando el robot debe recoger la clavija del cargador de clavijas.

¿Cuál cree que podría ser la causa de ello?



Fig. 11-2: Colisión en el cargador de clavijas

11.4 Principio de ajuste

¿Por qué se realiza el ajuste?



Fig. 11-3

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 dónde se encuentra cada eje.

Sólo un robot industrial perfecta y completamente ajustado puede funcionar de manera óptima. Solo entonces presenta una precisión de trayectoria y de punto completa y los programas pueden ejecutarse correctamente. Solo un robot no ajustado puede desplazarse en relación al eje.

El proceso completo de ajuste incluye el ajuste de cada uno de los ejes. Con un elemento auxiliar técnico (**EMD** – *Electronic Mastering Device*) se asigna un valor de referencia específico del robot a cada eje en su **posición mecánica cero** (por ejemplo, 0 °, -90 °, 110 °...).

Como con este proceso se hace coincidir la posición mecánica y eléctrica del eje, cada eje recibe un valor angular unívoco. Para todos los robots la posición de ajuste es similar, pero no idéntica. Las posiciones exactas pueden diferir también entre cada uno de los robots de un tipo de robot.

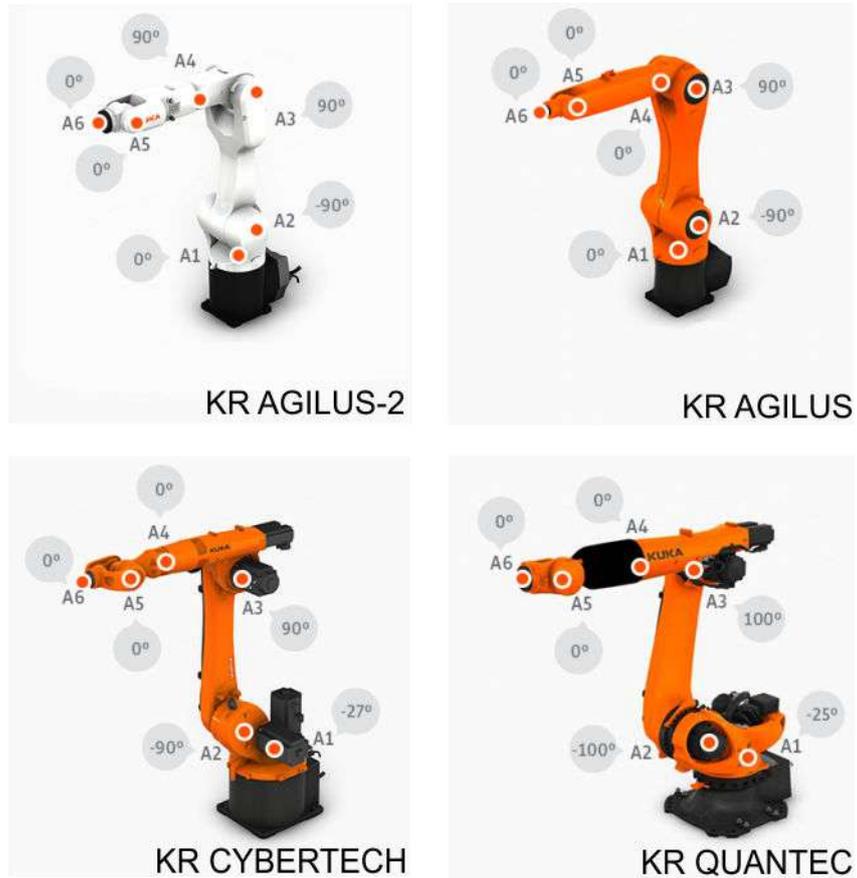


Fig. 11-4: Posiciones de ajuste

Valores angulares a modo de ejemplo para unas posiciones de referencia mecánicas:

Eje	CYBERTECH KR16 R1610	QUANTEC KR120 R2700-2
A1	-27 °	-25 °
A2	-90 °	-100 °
A3	90 °	100 °
A4	0 °	0 °
A5	0 °	0 °
A6	0 °	0 °

¿Cuándo se realiza un ajuste?

En principio, un robot siempre debe estar ajustado. En los siguientes casos es necesario un ajuste:

- Durante la puesta en servicio del robot.



Fig. 11-5: Puesta en servicio del robot

- 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).

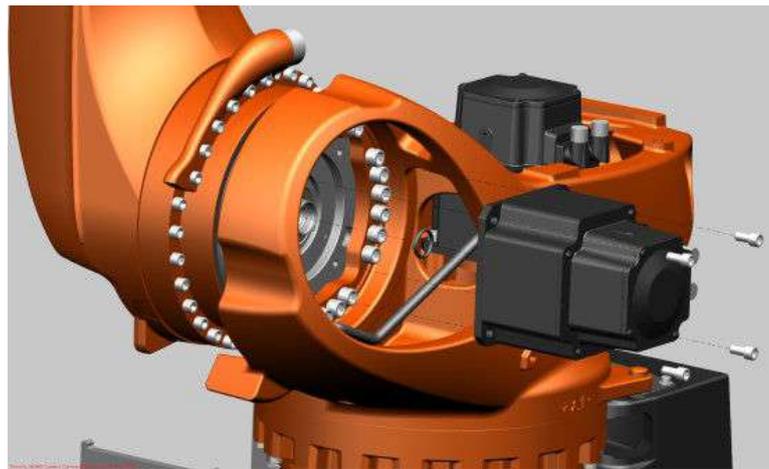


Fig. 11-6: Cambio de motor

- Cuando se hayan movido los ejes del robot sin unidad de control, por ejemplo, mediante un dispositivo de liberación.



Fig. 11-7: Dispositivo de liberación

- Tras problemas/reparaciones mecánicas, hay que realizar un ajuste o un ajuste de carga en función de la especificación, y confirmar las desviaciones calculadas.
 - Después de haber cambiado un engranaje.

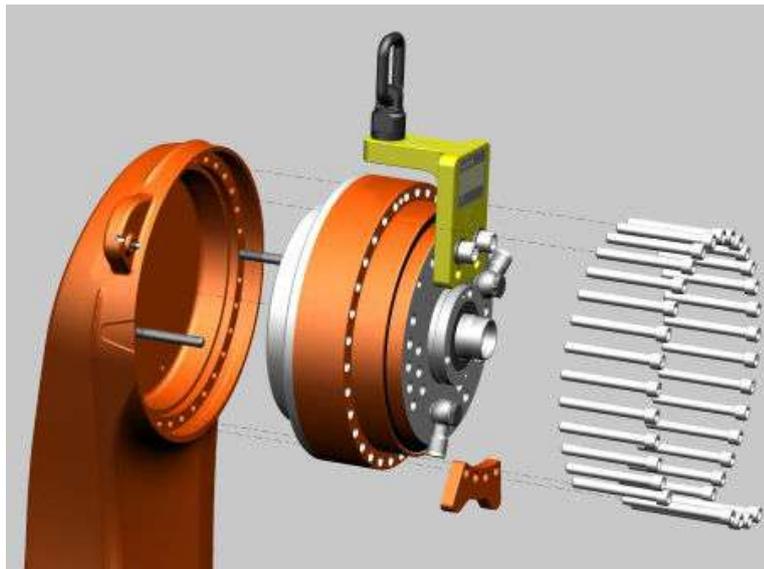


Fig. 11-8: Sustitución, engranaje

- Después de una colisión contra un tope final con una velocidad mayor de 250 mm/s



Fig. 11-9: Desplazarse hasta el tope final

- Tras una colisión.



Fig. 11-10: 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**
Esto permite desplazarse contra los topes finales mecánicos.

11.4.1 Medio de ajuste

Descripción

Hay dos medios de ajuste disponibles para el ajuste de los robots:

- Reloj comparador
- Juego de ajuste

Reloj comparador



Fig. 11-11: Reloj comparador

- El robot puede ajustarse de forma estándar con la ayuda del reloj comparador.
- La precisión depende mucho de lo que se ajuste (por ejemplo, error de paralaje).
- Se recomienda recurrir al juego de ajuste SEMD/MEMD.

AVISO

No es posible un ajuste de carga con el reloj comparador.

Juego de ajuste SEMD/MEMD



Fig. 11-12: Juego de ajuste SEMD/MEMD

- 1 Cables
- 2 Caja de ajuste universal
- 3 Sensor MEMD para cartuchos de ajuste pequeños
- 4 Sensor SEMD para cartuchos de ajuste grandes

Con el juego de ajuste SEMD/MEMD se pueden ajustar todas las variantes de robot con cartucho de ajuste **grande** y **pequeño**.

- **MEMD** – *Micro Electronic Mastering Device* para cartuchos de ajuste pequeños
- **SEMD** – *Standard Electronic Mastering Device* para cartuchos de ajuste grandes

El juego de ajuste también está disponible por separado como juego de ajuste SEMD o MEMD.

Juegos de ajuste disponibles

Juego de ajuste	Número de artículo	Descripción
SEMD/MEMD	00-228-936	Dispositivo de ajuste para todos los cartuchos de medición (rosca fina M8/M20) aplicable a todos los tipos de robot
SEMD	00-228-934	Dispositivo de ajuste para cartuchos de medición con rosca fina M20 aplicable, p. ej. para la serie QUANTEC
MEMD	00-208-642	Dispositivo de ajuste para cartuchos de medición con rosca fina M8 aplicable, p. ej., para AGILUS

Juego de ajuste	Número de artículo	Descripción
Opción KR C2	<u>00-228-327</u>	Cable adaptador para la utilización del SEMD como dispositivo de ajuste para sistemas KR C2 (KTL).

MEMD/SEMD en uso



Fig. 11-13: Ejemplo de conexión SEMD

Juegos de ajuste antiguos ya no disponibles

- **UEA** – *Palpador electrónico* para unidades de control basadas en KR C1/C2.
Para ello es necesaria la **opción KR C2** (adaptador).
- **EMD** – *Electronic Mastering Device* para unidades de control basadas en KR C4/C5.

11.4.2 Determinación de la posición mecánica cero

¿Qué aspecto tiene un punto de ajuste?

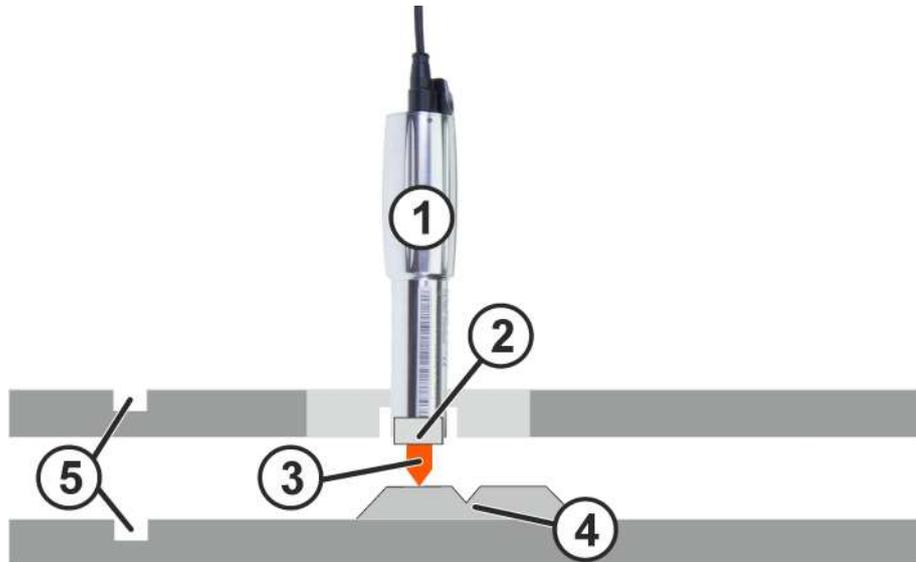


Fig. 11-14: Estructura de un punto de ajuste en el robot

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

Secuencia de un ajuste

1. La posición de preajuste se aproxima de forma específica del eje y con HOV lento a través de dos entalladuras opuestas del eje. Estas deben estar superpuestas ópticamente.

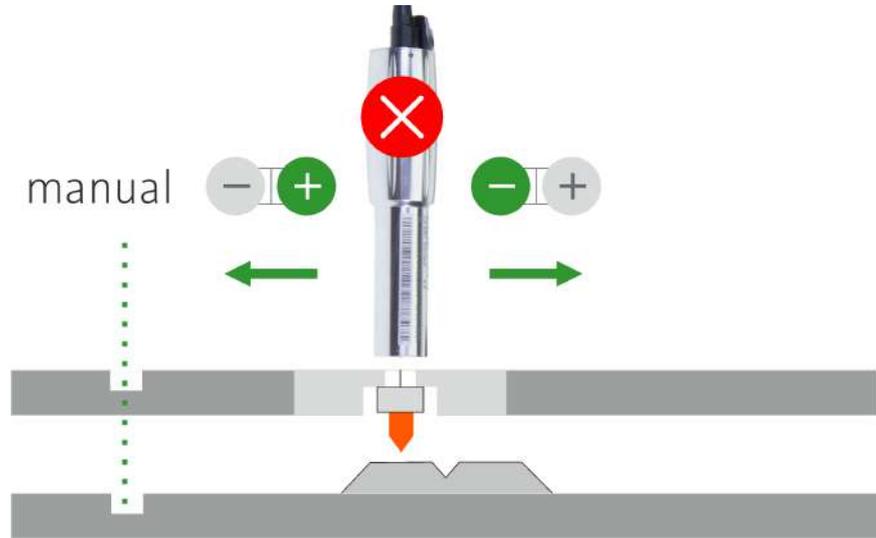


Fig. 11-15: Posición de preajuste

2. A través del menú de ajuste se inicia el desplazamiento de ajuste totalmente automático pulsando y manteniendo pulsada la tecla de inicio del smartPAD.

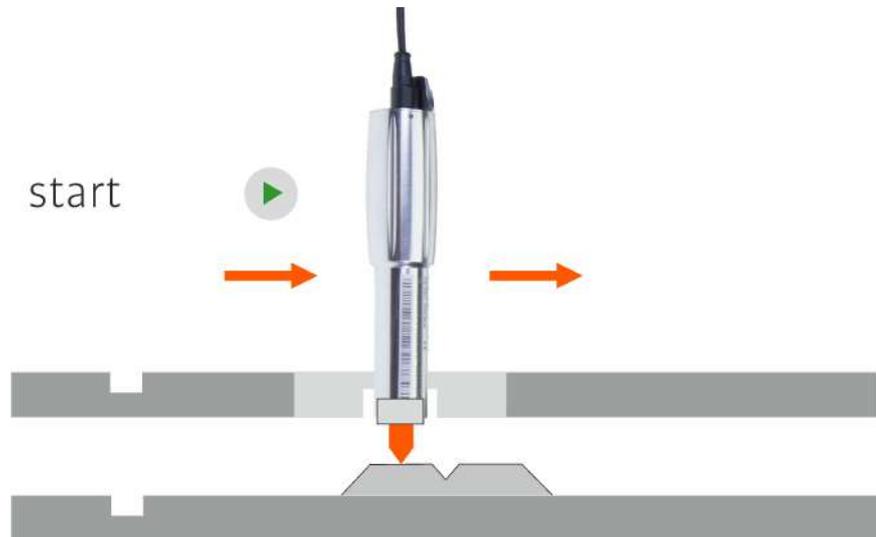


Fig. 11-16: Secuencia del ajuste

3. La entalladura de ajuste debe alcanzarse en un intervalo de tiempo definido. Si esto no ocurre, un mensaje de error indicará que el trayecto de ajuste se ha superado.

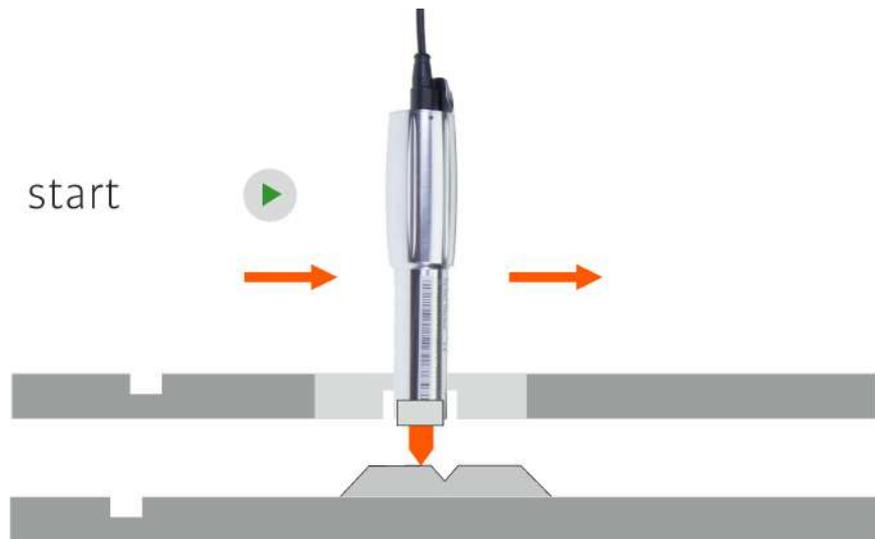


Fig. 11-17: Secuencia del ajuste

4. Si el punto más bajo de la entalladura de ajuste se alcanza ópticamente, el robot se desplazará más allá de este punto. Esto será importante posteriormente para el cálculo de la entalladura de ajuste. (>>> *"¿Cómo se calcula el punto cero mecánico?" Página 130*)

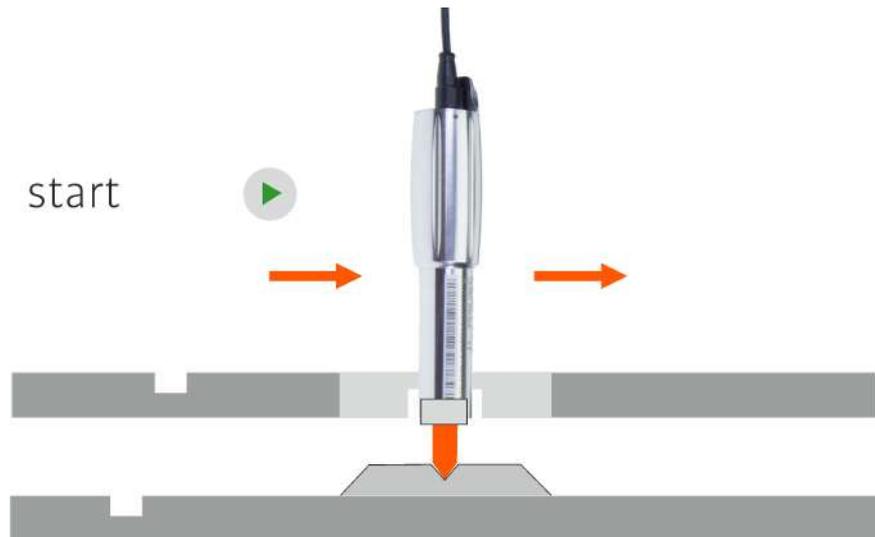


Fig. 11-18: Secuencia del ajuste

5. El robot detiene automáticamente el desplazamiento de medición al final de la grabación. Después se puede soltar la tecla de inicio.

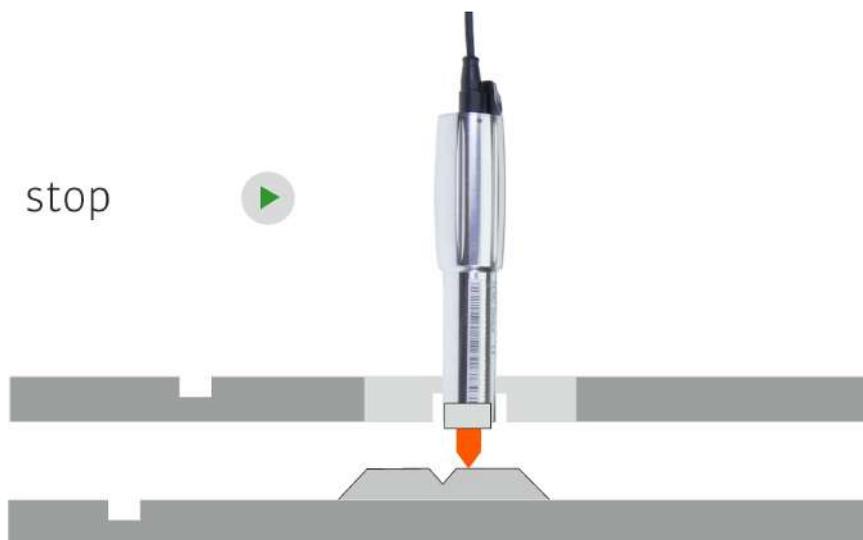
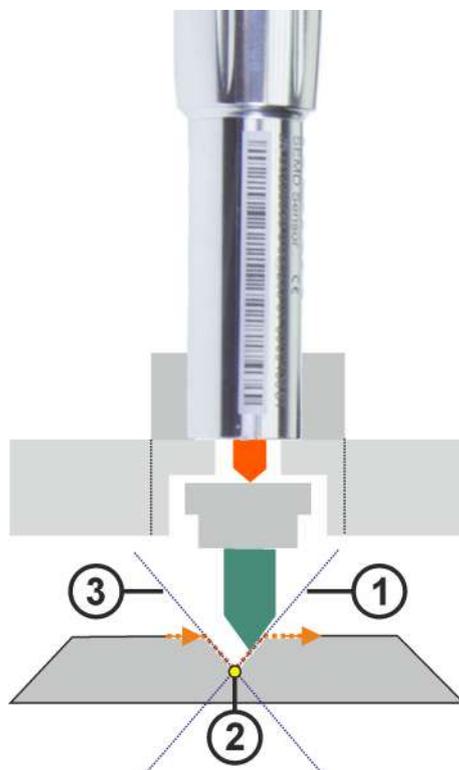


Fig. 11-19: Secuencia del ajuste

¿Cómo se calcula el punto cero mecánico?

En el ajuste se determina el punto cero mecánico (2) del eje. Para ello, el eje se mueve hasta que el palpador de medición haya realizado una vez la entalladura de ajuste. A continuación, se determina la posición exacta de la entalladura. El punto de intersección del flanco descendente (3) y ascendente (1) se calcula y corresponde al punto más profundo (2). El punto más profundo (2) corresponde a su vez a la posición cero mecánica del eje. El eje se ajusta al valor de ajuste (indicación en grados) depositado en los datos de la máquina. Por ello, cada eje está equipado con un cartucho de medición, una entalladura de medición y una marca de preajuste.



En caso de que haya diferentes tipos de robot, p. ej., Agilus 1 y KR CYBERTECH nano, el eje 6 no tiene ningún cartucho de medición. Para estos tipos de robots, la posición cero mecánica se realiza mediante una marca en el eje correspondiente.

11.4.3 Ajuste del robot

Descripción

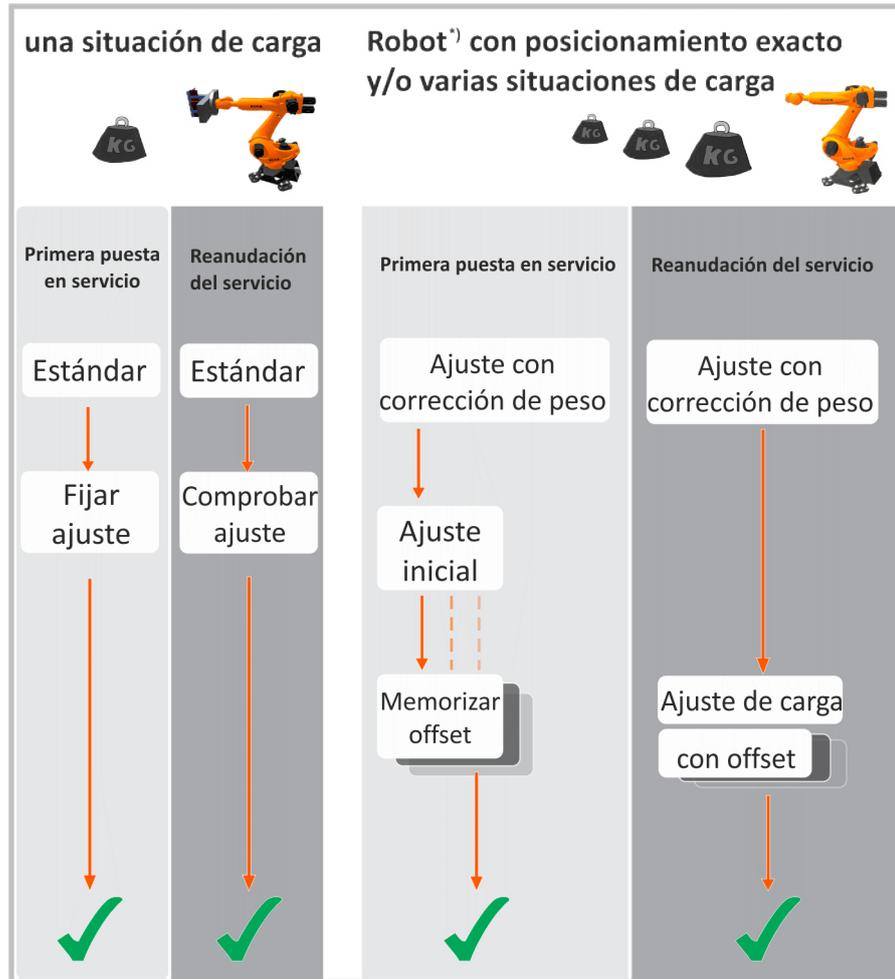


Fig. 11-20: Opciones de ajuste

Ajuste estándar

Columna de la figura: *una situación de carga*

Este tipo de ajuste se utiliza cuando el robot guía una herramienta instalada fija con un peso constante, p. ej., un soplete, una pinza de soldadura por puntos o una garra vacía.

Efectos sobre la reanudación del servicio posible

- Si **tras** la creación del programa el robot se ajusta con una herramienta modificada o una carga modificada, esto tiene una influencia directa sobre la exactitud de los puntos programados por aprendizaje.
- La modificación de la carga deriva en una "torsión" mayor/menor del robot y, por lo tanto, también en los valores de resolver modificados que guardados.
- Los propios valores de resolver son bases de cálculo ("posiciones de inicio") para la determinación de la posición del robot.

Ajuste con corrección de peso

Columna de la figura: *Robot con posicionamiento exacto y/o varias situaciones de carga*

Este tipo de ajuste se usa, cuando:

- sin desmontar la herramienta y la carga adicional debe restaurarse un ajuste inicial (sin carga). *)
- el robot trabaja en su ámbito de aplicación con un sistema de cambio de herramienta, por ejemplo, garra y una pinza para soldar por puntos.



*)El modelo de robot con posicionamiento exacto se basa en un ajuste inicial sin carga. Por ello es muy importante que con cada proceso de ajuste se vuelva a restaurar este estado. Para simplificarlo y evitar el desmontaje de herramientas y cargas adicionales, está disponible el camino a través del ajuste con corrección de carga.

Efectos sobre la reanudación del servicio posible



Load mastering with offset ⊖



Offset tool



⊖ First mastering

Durante la puesta en servicio, el robot siempre se ajusta de forma inicial (sin herramienta/carga) y a continuación se programan los offsets para cada herramienta.

El offset calculado en ello para el ajuste inicial se guarda de forma concreta para cada herramienta.

Tras una posible avería, el robot se puede ajustar con la herramienta/carga montada en ese momento.

Mediante el offset específico de la herramienta, el robot está capacitado para volver a calcular el ajuste inicial.

11.4.3.1 Conocer el ajuste inicial

Descripción

Un ajuste inicial solo se puede realizar cuando el robot está sin carga. No se debe montar ninguna herramienta ni carga adicional.



Fig. 11-21: Ajuste inicial

Realizar el ajuste inicial

Procedimiento

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



Fig. 11-22: Ejemplos de posición de preajuste

- En caso necesario, ajustar un override lento.
- Desplazar lentamente el eje que debe ajustarse hasta la marca de ranura y punto de mira con las teclas \pm .

2. Atornillar y conectar MEMD/SEMD al cartucho de medición.

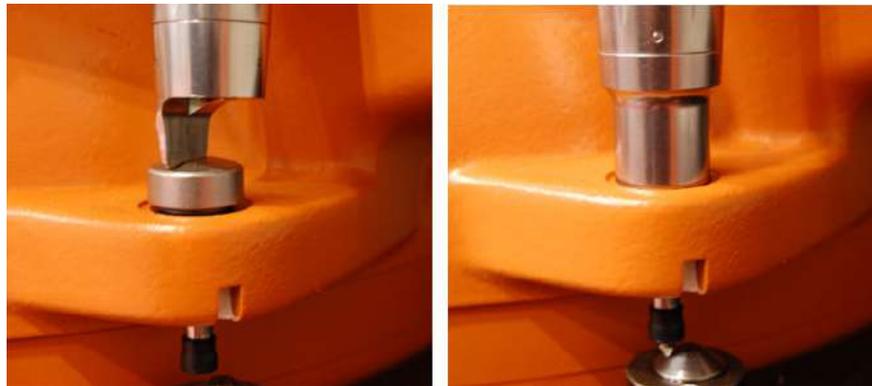


Fig. 11-23: SEMD, enroscado sobre el cartucho de medición

- Retirar del eje físico marcado en la ventana la tapa protectora del cartucho de medición.
Para ello se puede utilizar el destornillador integrado en MEMD/SEMD.

3. Conectar el cable MEMD/SEMD



Fig. 11-24: Cable EMD, conectado

- Atornillar el MEMD/SEMD al cartucho de medición.
- Conectar el cable de medición al EMD y a la conexión X32 de la caja de conexiones de la base.



ATENCIÓN

Enroskar 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.

4. Activar el ajuste inicial

Ruta de menú: Tecla de robot > Puesta en servicio > Ajustar > EMD > Con corrección de carga > Ajuste inicial

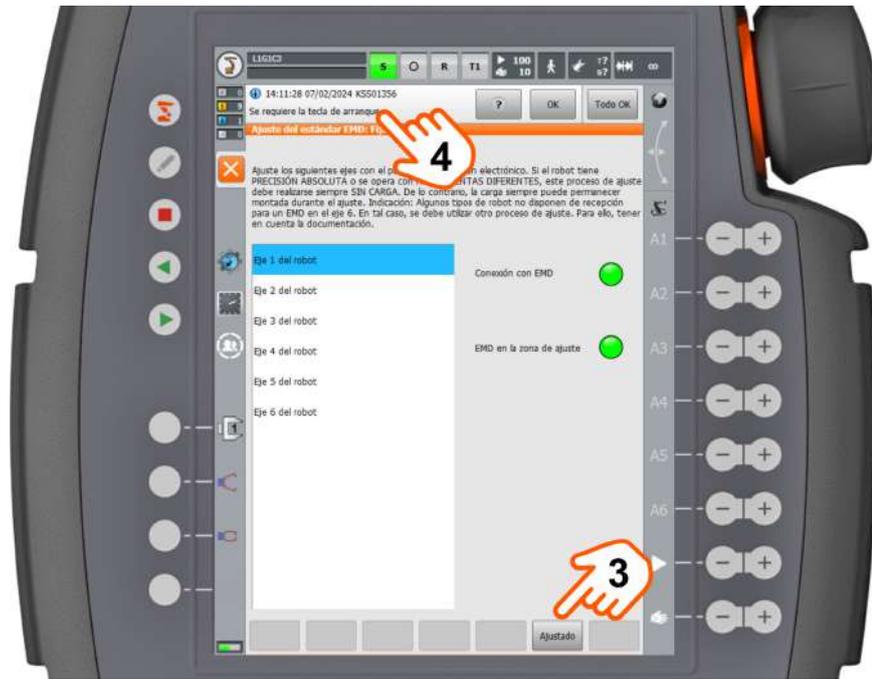


5. Ventana Ajuste inicial



- Pulsar el eje que hay que ajustar en el menú (2).
El eje activo se marca en azul.

6. Realizar el desplazamiento de ajuste, programar un offset



- Iniciar el recorrido de ajuste con el botón Ajuste (3).
- En la ventana de mensajes se mostrará el mensaje *Tecla de inicio necesaria* (4).

7. Pulsar el pulsador de validación y la tecla de inicio



Fig. 11-25: Posición de las teclas de arranque en el smartPAD

- El robot sale de la marca de ajuste y calcula la posición de ajuste.

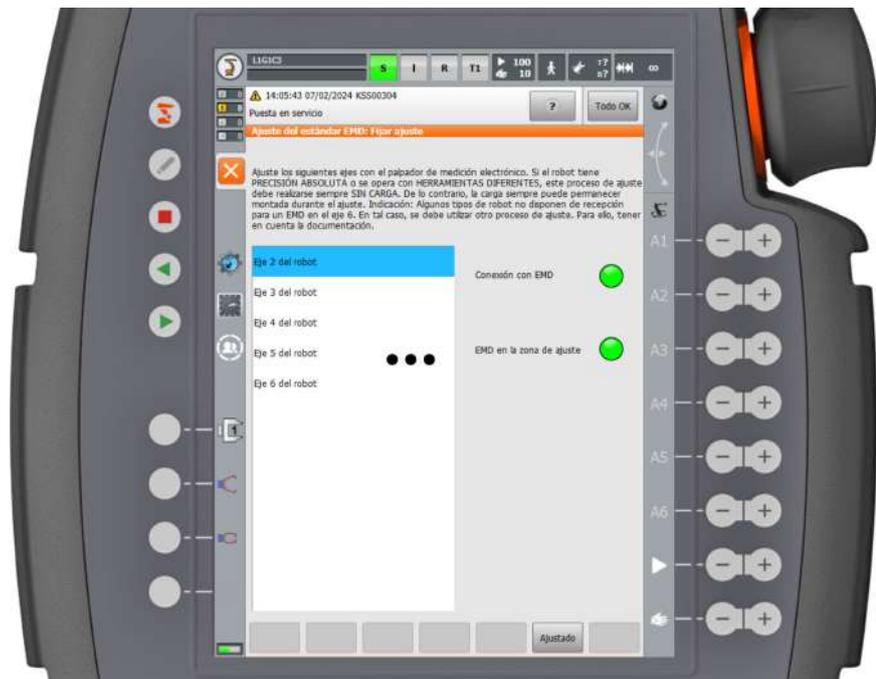


Hay que mantener pulsadas las teclas durante todo el recorrido de medición.

- El robot se detiene automáticamente.

8. Ocultar los ejes ya ajustados

El eje que se acaba de ajustar se oculta en la ventana de selección.



9. Retirar el cable de medición del EMD. A continuación, retirar el EMD del cartucho de medición y volver a colocar la tapa protectora.
10. Repetir los pasos para todos los ejes que haya que ajustar.
11. Cerrar la ventana.
12. Retirar de la conexión X32 el cable de medición.

11.4.3.2 Ajuste con corrección de peso

Situación cotidiana de un pescador

Un pescador quiere atrapar un pez. Para ello, sostiene su caña por encima del agua e intenta alcanzar un punto fijo (1) con una distancia definida por encima de la superficie del agua. Para ello, "gira" la mano hasta alcanzar esta posición. La caña de pescar no está sometida a carga en este momento (2a) y aún no ha picado ningún pez. Si un pez pica (2b), la caña de pescar se somete a carga. Esto da lugar a una torsión/flexión, ya que la caña de pescar es elástica. Para mantener el punto de partida (1) se debe "volver a girar" la mano (3b).

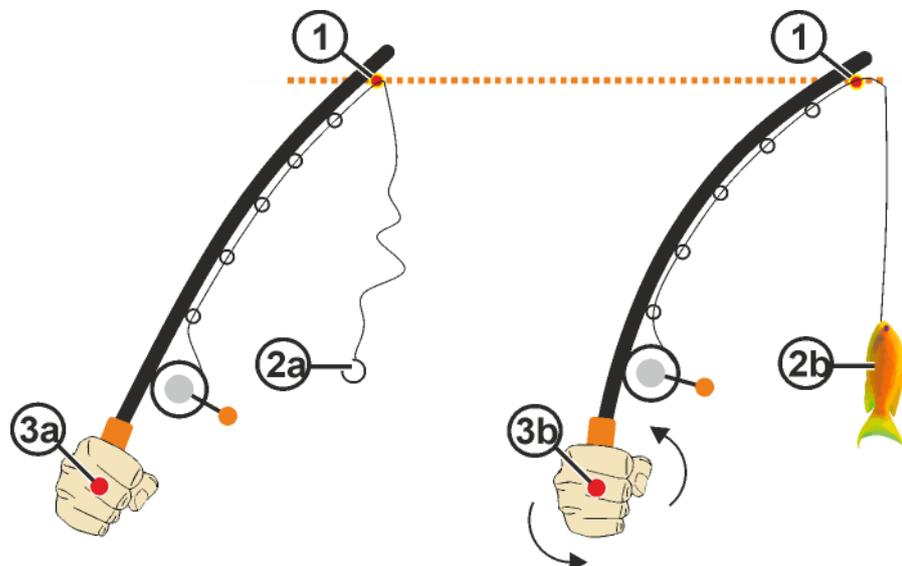


Fig. 11-26: Efecto de la carga en punto de giro

¿Para qué sirve memorizar el offset?



Fig. 11-27: Juego de engranaje y elasticidad

Debido al peso de la herramienta que está sujeta a la brida, el robot está sometido a una carga estática debido a las grandes influencias de la palanca. A consecuencia de la elasticidad de los componentes y accionamientos, puede haber diferencias en las posiciones de un robot cargado y otro sin carga. Estas diferencias de escasos incrementos (grado de ángulo de motor) repercuten en la exactitud del robot.



Los incrementos son los valores digitales más pequeños posibles para registrar el giro en un motor del robot. Los incrementos pueden compararse con una graduación.



Fig. 11-28: Memorizar offset

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

- Cuando el robot trabaja con diferentes cargas, debe ejecutarse la función "Programar offset" para cada una de las cargas para poder restablecer el ajuste más fácilmente.
- En el caso de garras que sujetan piezas pesadas y en las que se desea una restauración simplificada del ajuste inicial, se debe ejecutar "Programar offset" respectivamente para la garra y para la garra con el componente.



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.

Realizar el ajuste con corrección de peso

Procedimiento para memorizar Offset



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

1. El ajuste inicial se hizo en el robot sin herramienta y sin carga.



Fig. 11-29: Ajuste inicial

2. Ruta de menú: *Tecla de robot > Puesta en servicio > Ajustar > EMD > Con corrección de carga > Ajuste inicial*

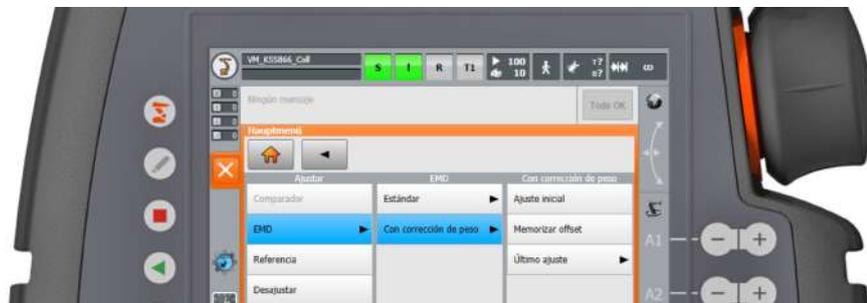


Fig. 11-30: Ruta de menú

3. Atornillar la herramienta necesaria para el ajuste en la brida del robot.



Fig. 11-31: Ajuste con herramienta

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

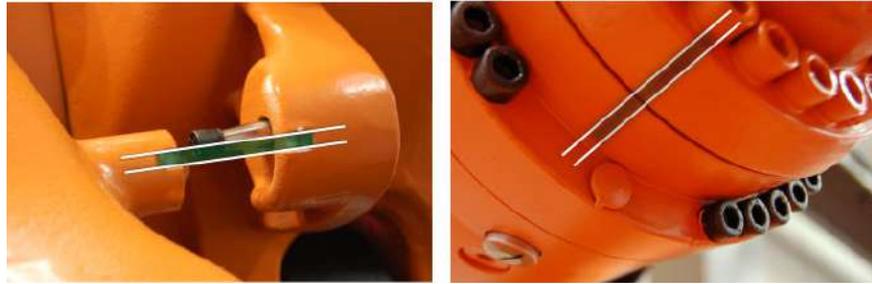


Fig. 11-32: Ejemplos de posición de preajuste

- En caso necesario, ajustar un override lento.
 - Desplazar lentamente todos los ejes que hay que ajustar hasta la marca de ranura y punto de mira con las teclas ±.
5. **Ruta de menú:** *Tecla de robot > Puesta en servicio > Ajustar > EMD > Con corrección de carga > Memorizar offset*

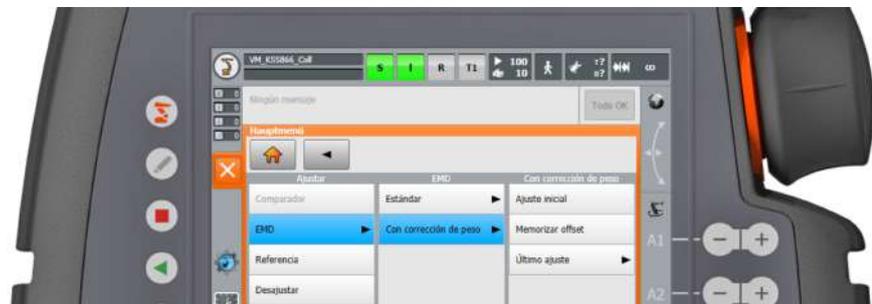


Fig. 11-33: Ruta de menú

6. **Introducir el número de herramienta.**

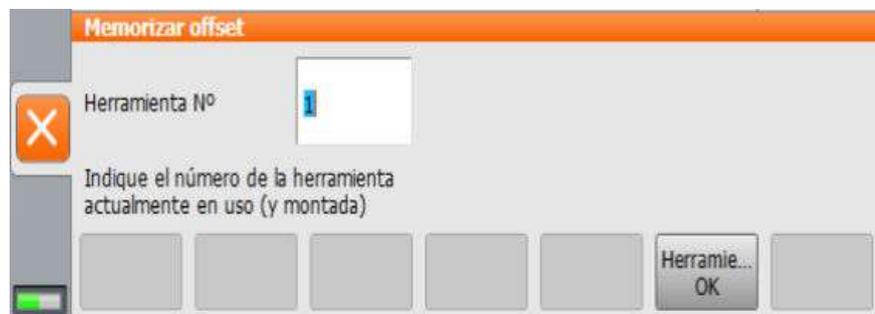


Fig. 11-34: Selección de herramientas, ajuste de carga con offset

- Introducir numéricamente el número de herramienta en el campo de entrada.
 - Confirmar con el botón **Herramienta. OK**.
7. Ventana de ajuste **Memorizar offset**

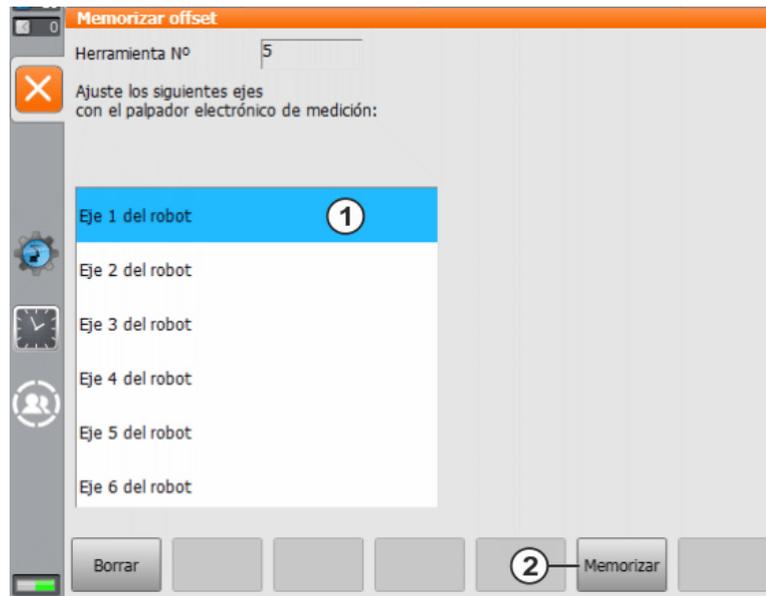


Fig. 11-35: Memorizar offset

- Pulsar el eje que hay que ajustar en el menú (1).
El eje activo se marca en azul.

8. Atornillar y conectar MEMD/SEMD al cartucho de medición.



Fig. 11-36: SEMD, enroscado sobre el cartucho de medición

- Retirar del eje físico marcado en la ventana la tapa protectora del cartucho de medición.
Para ello se puede utilizar el destornillador integrado en MEMD/SEMD.

9. Conectar cable MEMD/SEMD



Fig. 11-37: Cable EMD, conectado

- Atornillar el MEMD/SEMD al cartucho de medición.

- Conectar el cable de medición al EMD y a la conexión X32 de la caja de conexiones de la base.



ATENCIÓN

Enroskar 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.

10. Realizar el recorrido de ajuste.

- Iniciar el recorrido de ajuste con el botón Memorizar (2).
- En la ventana de mensajes se mostrará el mensaje *Tecla de arranque necesaria*.

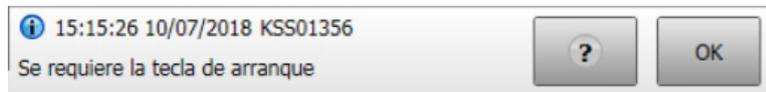


Fig. 11-38: Ventana de mensajes.

11. Pulsar el pulsador de validación y la tecla de arranque.



Fig. 11-39: Posición de las teclas de arranque en el smartPAD

- El robot sale solo de la marca de ajuste y calcula la posición de ajuste.
Hay que mantener pulsadas las teclas durante todo el recorrido de medición.
- El robot se detiene automáticamente.

12. Pulsar el pulsador de validación y la tecla de arranque.

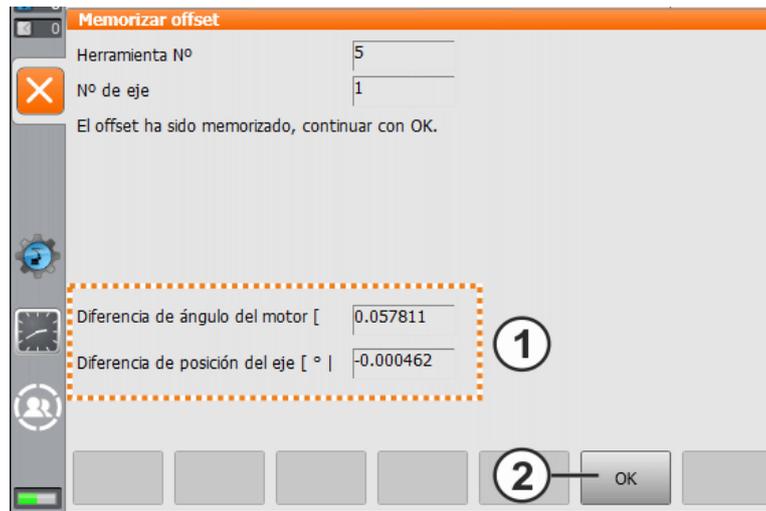


Fig. 11-40: Diferencia ángulo del motor/posición del eje

- Se abre una ventana.
La desviación de este eje respecto al ajuste inicial se muestra en grados de ángulo del motor y ángulo de ejes (1).
- Con el botón OK se guardan los valores y se puede ajustar el siguiente eje.
- El eje que se acaba de ajustar se oculta en la ventana de selección.
(>>> Fig. 11-35)

13. Retirar el cable de medición del EMD. A continuación, retirar el EMD del cartucho de medición y volver a colocar la tapa protectora.
14. Repetir los pasos para todos los ejes que haya que ajustar.
15. Retirar de la conexión X32 el cable de medición.
16. Abandonar la ventana pulsando en **Cerrar**.

Realizar el ajuste con corrección de peso después de una avería

Ajuste de carga tras un accidente

- El robot deberá volver a ajustarse si, debido a una avería/colisión, tiene que cambiarse un motor o un engranaje.
- Básicamente debe comprobarse el estado en el que está el robot.
 - No hay **ninguna** herramienta en el robot.



Ruta de menú: Tecla de robot > Ajuste > EMD > Con corrección de carga > Ajuste de carga > Primer ajuste

El proceso de medición se corresponde con la primera puesta en servicio "Primer ajuste"

- Hay una herramienta en el robot.



Ruta de menú: Tecla de robot > Ajuste > EMD > Con corrección de carga > Ajuste de carga > Con offset

El proceso de medición se corresponde con la primera puesta en servicio "Ajuste con corrección de carga"

AVISO
<p>El ajuste de carga sin offset solo debe utilizarse si el robot no ha sufrido ninguna modificación mecánica. No es admisible en caso de colisión, cambio de motor, cambio de engranaje, etc.</p>

11.4.3.3 Conocer el ajuste estándar

Descripción

El ajuste estándar se mide con la herramienta montada.
 El ajuste estándar sobrescribe un posible ajuste inicial existente.



Fig. 11-41: Robot en el ajuste estándar



ATENCIÓN
<p>Si un robot se ha ajustado por primera vez y se han programado offsets, no se debe realizar ningún ajuste estándar con la herramienta montada. El primer ajuste y el ajuste estándar se guardan en el mismo sitio dentro de la unidad de control. Los puntos programados ya no pueden aproximarse con precisión.</p>

Realizar el ajuste estándar

Procedimiento

- **Ruta de menú:** Tecla de robot > Puesta en servicio > Ajuste > EMD > Estándar > Fijar ajuste

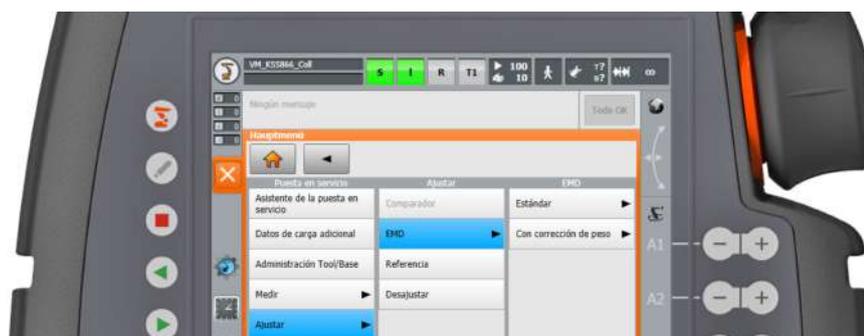


Fig. 11-42: Ruta de menú, ajuste estándar

- El resto de pasos son idénticos a los del primer ajuste.



11.5 Ejercicio: Búsqueda de errores después de la prueba de programa

¡Ahora es su turno!



- Ha probado su nuevo programa creado. Durante la ejecución en el modo de servicio T1 se produjo una colisión del robot.
- Antes de programar posteriormente o de crear de nuevo puntos en un programa de robot, deberá comprobar el ajuste.
- Por la documentación de la instalación le consta que se ha programado un offset de ajuste para la garra vacía. El número de herramienta para la garra vacía es n.º
- Compruebe si el ajuste ha sufrido daños debido a la colisión y guarde en este caso los nuevos valores de ajuste.
- Para ello, realice un ajuste de carga con offset.

Lo que se debe saber tras el ejercicio:



¿Qué debe hacer si ha detectado desviaciones mayores de ajuste?

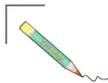


Fig. 11-43: Campo de respuesta

¿Por qué cada eje debería estar ajustado correctamente?

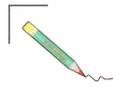


Fig. 11-44: Campo de respuesta

¿Qué efecto tiene el reajuste de un eje en los programas existentes?



Fig. 11-45: Campo de respuesta

11.6 Ejercicio: Búsqueda de errores después de la prueba de programa

¡Ahora es su turno!



- Ha probado su nuevo programa creado recientemente. Durante la ejecución en el modo de servicio T1 se produjo una colisión del robot.
- Compruebe si el ajuste ha sufrido daños debido a la colisión.
- Está informado de que ayer se sustituyó la garra en el robot. Esta debería haberse vuelto a programar con un offset de ajuste.
- Antes de programar posteriormente o de crear de nuevo puntos en un programa de robot, compruebe el ajuste.
- Para la garra utilizada debe realizarse un ajuste con offset según las especificaciones de la empresa. El número de herramienta para la garra vacía es n.º _____
- Compruebe si el ajuste de carga con offset se ha realizado correctamente.

Lo que se debe saber tras el ejercicio:



¿Qué debe hacer si ha detectado desviaciones mayores de ajuste?

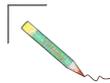


Fig. 11-46: Campo de respuesta

¿Por qué cada eje debería estar ajustado correctamente?

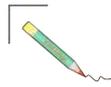


Fig. 11-47: Campo de respuesta

¿Qué efecto tiene el reajuste de un eje en los programas existentes?

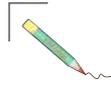


Fig. 11-48: Campo de respuesta

12 Configuración de una nueva estación de robot: Medir la herramienta del robot

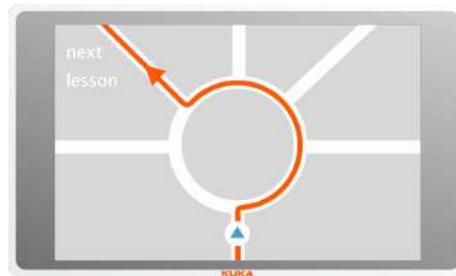
12.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots.
- Está capacitado para desplazar el robot mediante el sistema de coordenadas WORLD, BRIDA, BASE y TOOL.

¿Qué le espera en esta unidad de aprendizaje?



Medir la herramienta del robot

- Aprenderá cómo se mide una herramienta de robot.
- Aprenderá los pasos necesarios para medir una herramienta.
- Podrá asignar correctamente términos como "TCP" y "dirección de impacto de la herramienta".

12.2 Descripción de la situación

Los robots de KUKA también se utilizan en los conocidos estudios de cine de Babelsberg. En este ejemplo, un robot con acoplamientos diferentes fresa objetos detallados de poliestireno (aquí un reloj gigante).



Fig. 12-1: Robot milling

Imagine qué pasaría si el robot sacara la herramienta incorrecta de la estación al inicio del programa y comenzara a fresar.

¿Cuáles son los efectos en relación con el material a procesar?

Correcto, después de que la trayectoria se refiera al TCP, este no coincide con el TCP real de la herramienta. En ese caso, es posible que el material ya no se pueda utilizar de nuevo.

12.3 Ejercicio: Comprobar el TCP y la orientación de la herramienta

¡Ahora es su turno!

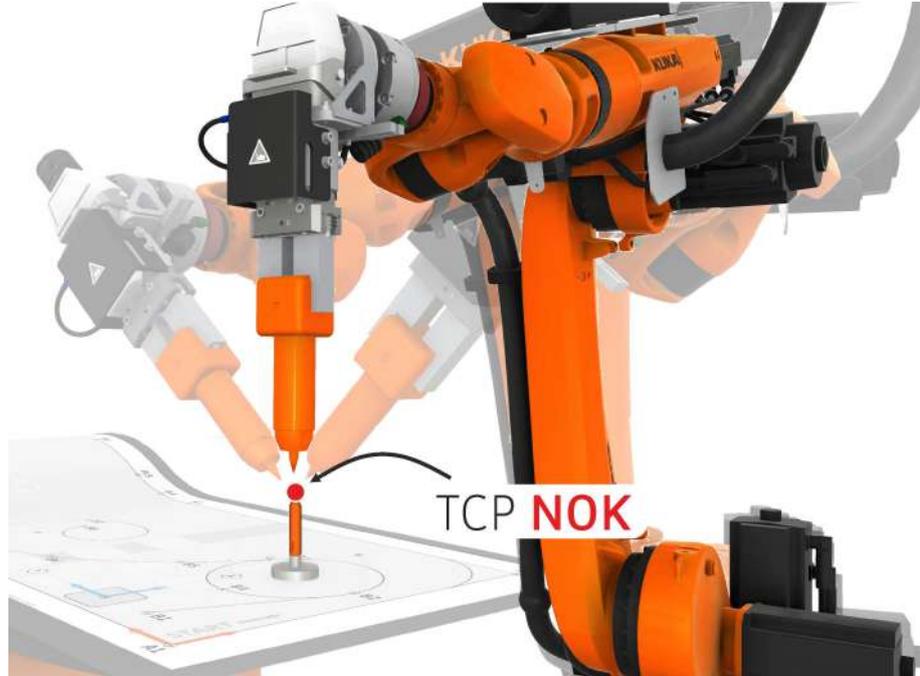
- Sujete la clavija central en la garra y realice el desplazamiento hasta un punto de referencia, p. ej. la punta de medición.
- Active la herramienta existente "Clavija_Larga" y modifique la orientación con las teclas de desplazamiento A, B, C.
- **¿Cómo se comporta la punta de la clavija?**
- Modifique el sistema de coordenadas seleccionado para el desplazamiento manual (World, Tool, Base) y vuelva a pulsar las teclas de desplazamiento A, B, C.
- **¿Qué le llama la atención?**

12.4 ¿Cómo puedo comprobar el TCP activo en la herramienta?

El TCP activo no corresponde a la herramienta física

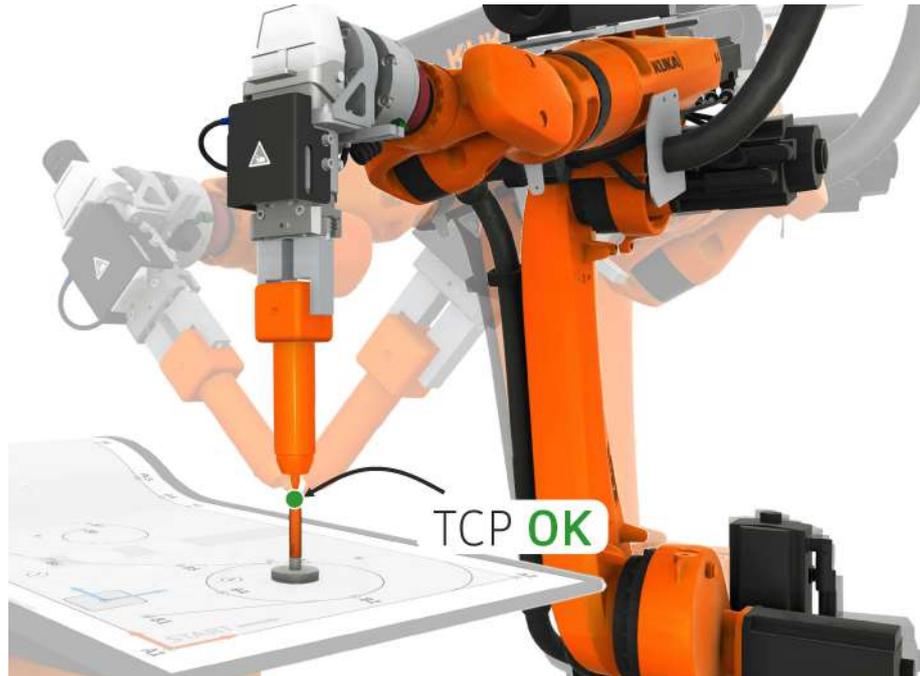
Comprobar el TCP y la orientación de la herramienta

En el módulo de aprendizaje anterior ha duplicado el programa para la recogida y el depósito de la "Clavija_L" y lo ha adaptado para la "Clavija_M". Usted ejecuta el programa y quieren ampliarlo para incluir puntos en la mesa de ejercicios. Al reorientar, se da cuenta de que el TCP de la punta de la herramienta no es correcto.



El TCP activo corresponde a la herramienta física

La comprobación cruzada con la "Clavija_L" da como resultado que esta herramienta coincide con el TCP activo.



12.5 ¿Qué pasos son necesarios para medir una nueva herramienta?

1. Medición del TCP – *Tool Center Points*



2. Medición de la orientación de la herramienta

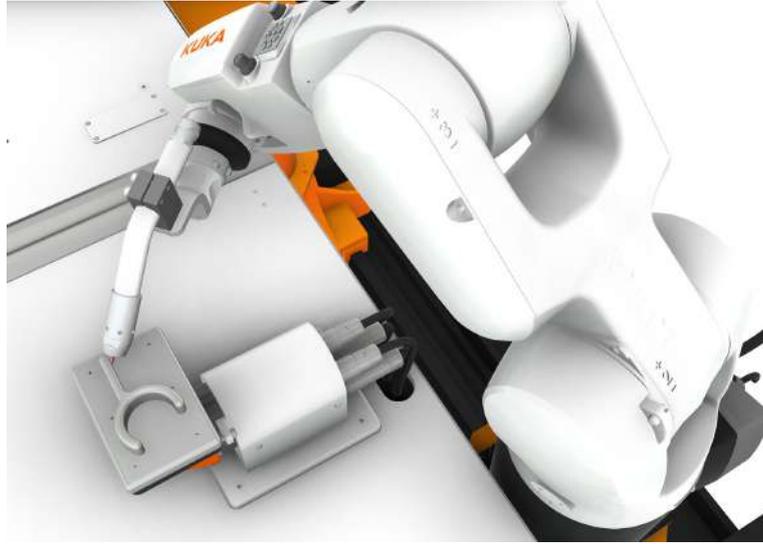


3. Determinar y guardar los datos de carga



4. Comprobar la herramienta nueva

Configuración de una nueva estación de robot: Medir la herramienta del robot



12.6 Preguntas: Comprobar el TCP y la orientación de la herramienta

Lo que se debe saber tras el ejercicio:



¿Cuándo debe crearse una nueva herramienta?



Fig. 12-2: Campo de respuesta

¿Cómo se puede determinar la posición del TCP de una herramienta?

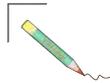


Fig. 12-3: Campo de respuesta

¿Cómo se puede averiguar si una herramienta activada es adecuada para la herramienta montada actualmente en el robot?

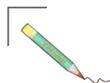


Fig. 12-4: Campo de respuesta

¿Qué son los datos de carga de la herramienta?

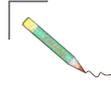


Fig. 12-5: Campo de respuesta

¿Por qué se necesitan datos de carga de la herramientas?

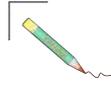


Fig. 12-6: Campo de respuesta

12.7 Herramientas guiadas por robot

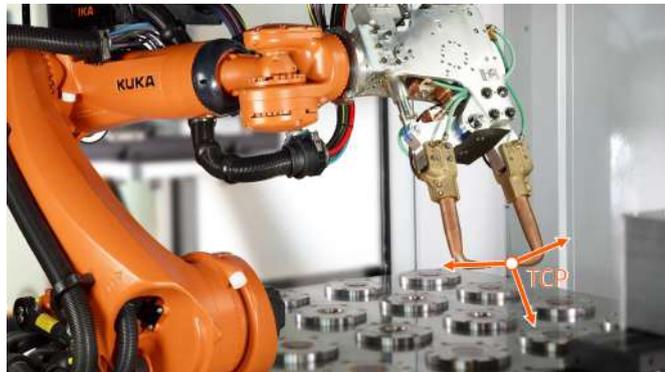
Descripción

Medir una herramienta significa que se genera un sistema de coordenadas que tiene su origen en el punto de referencia de una herramienta. Este punto de referencia recibe el nombre de **TCP** (Tool Center Point), mientras que el sistema de coordenadas es el sistema de coordenadas de la **herramienta**.

Soplete



Pinza de soldadura por puntos



Garra



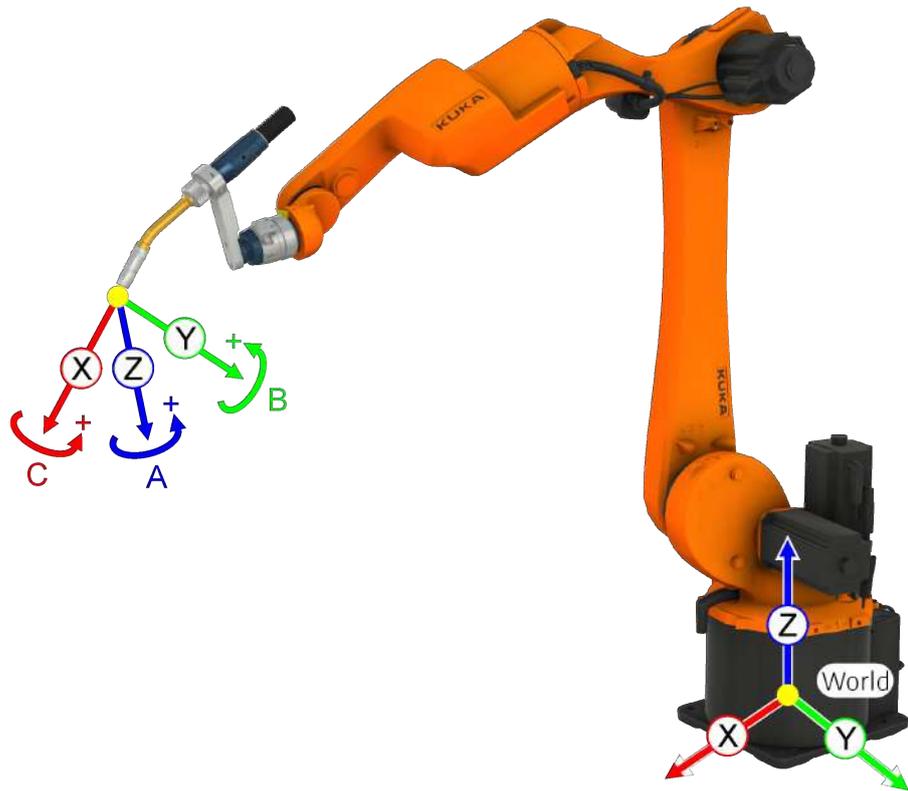
12.7.1 Conocer las ventajas de las herramientas medidas

Ventajas

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

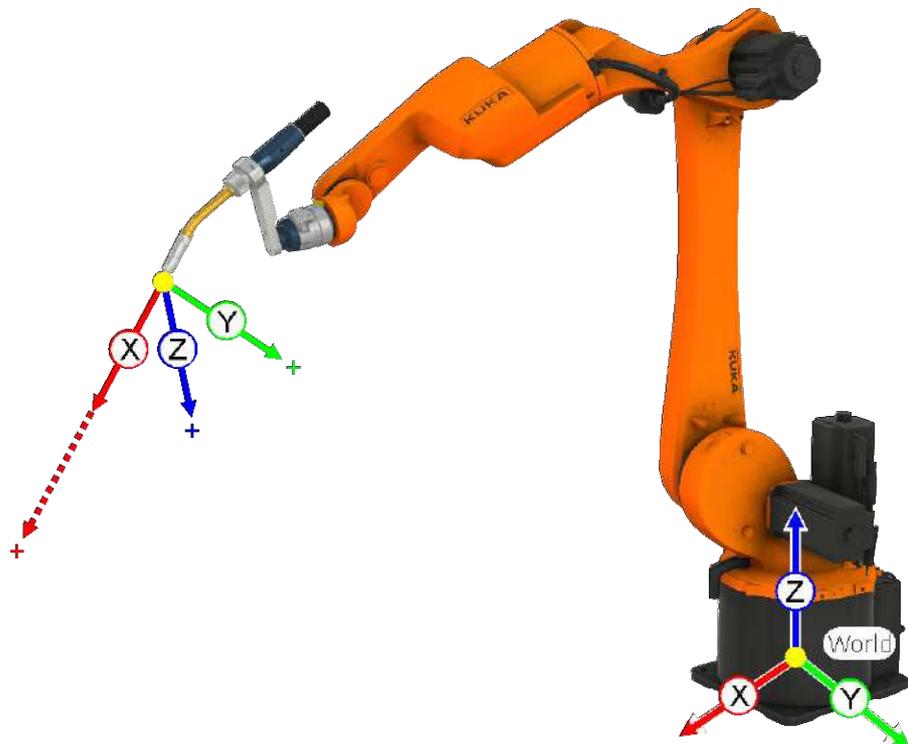
Reorientación alrededor del TCP

La herramienta se puede reorientar/alinear alrededor del TCP (p. ej., la punta de la herramienta).



Desplazamiento en dirección de impacto de la herramienta

El robot puede desplazarse a lo largo de la dirección de avance de la herramienta.



Programación de movimientos de trayectoria

La velocidad programada se mantiene en el TCP a lo largo de toda la trayectoria.

Un control de la orientación definido a lo largo de la trayectoria es posible con un TCP medido de la herramienta.

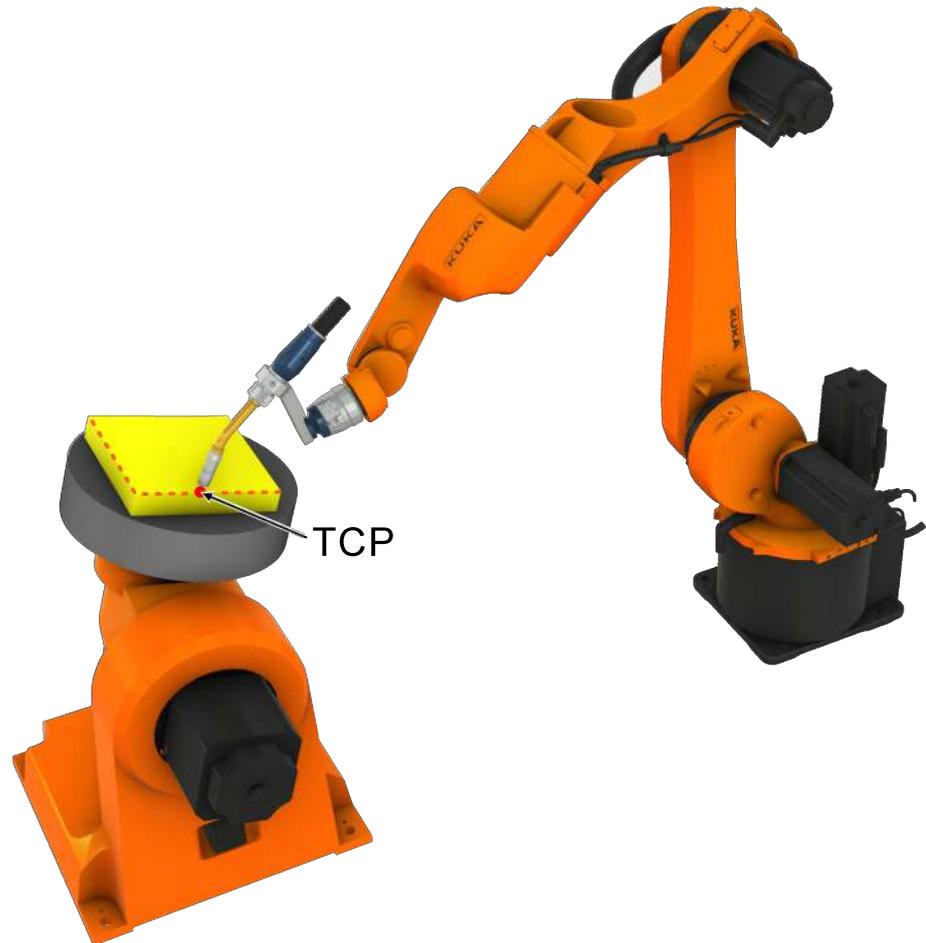


Fig. 12-7: Servicio de programa con TCP

12.8 Medición de una herramienta

Pasos

Durante la medición se guarda la distancia entre el sistema de coordenadas Tool (en X, Y y Z) y el sistema de coordenadas de la brida, así como el giro entre sí (ángulo A, B y C).

Configuración de una nueva estación de robot: Medir la herramienta del robot

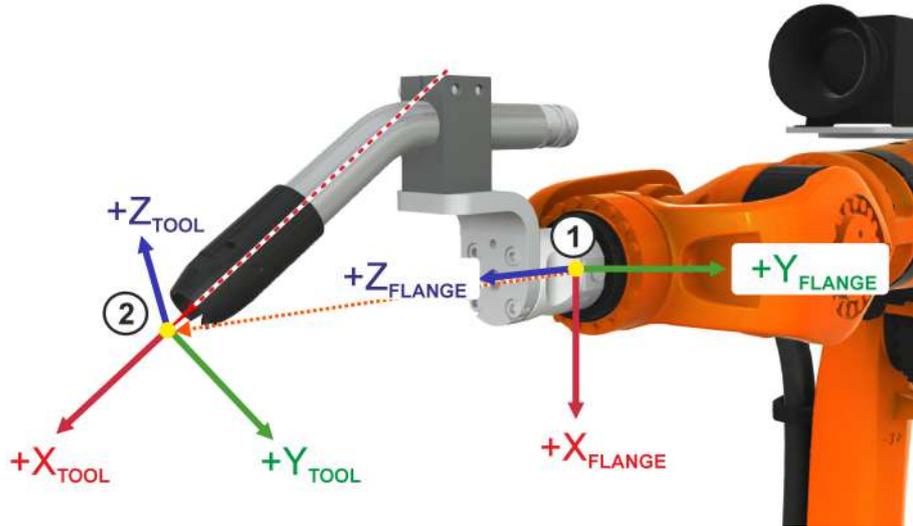


Fig. 12-8: Medición de una herramienta

- 1 Sistema de coordenadas FLANGE
- 2 Sistema de coordenadas TOOL

AVISO

En la configuración estándar se pueden guardar como máximo 16 sistemas de coordenadas de herramienta (variable: TOOL_DATA[1...16]).

Opciones de medición de la herramienta

La medición de la herramienta consta de tres pasos:

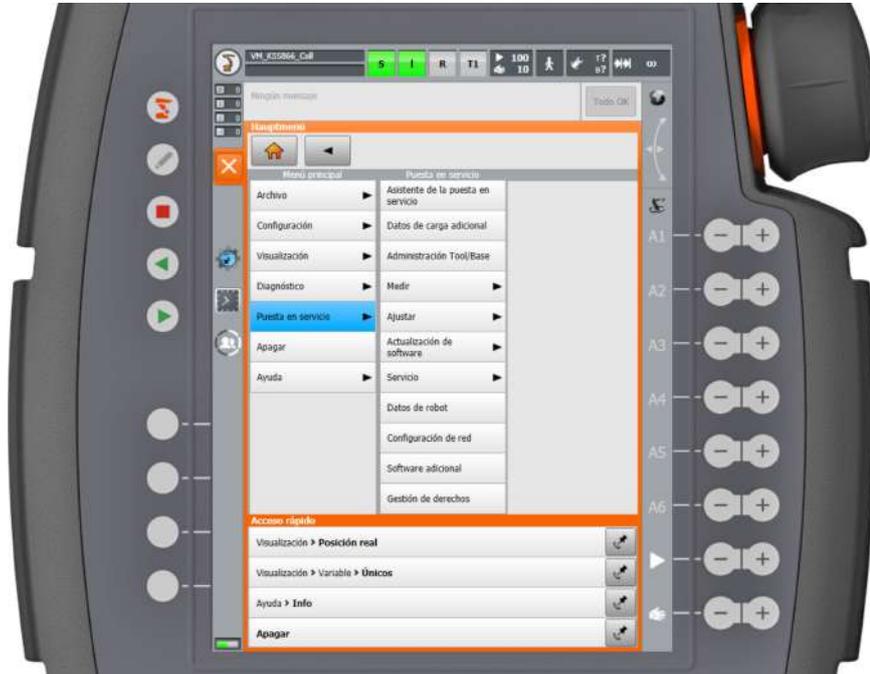
Paso	Descripción
1	<p>Definir el origen del sistema de coordenadas TOOL</p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> • XYZ de 4 puntos • Referencia XYZ
2	<p>Definir la orientación del sistema de coordenadas TOOL</p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> • ABC World • ABC de 2 puntos
Alternativa:	<p>Entrada directa de los valores para la distancia hasta el punto central de la brida (X, Y, Z) y el giro (A, B, C)</p> <ul style="list-style-type: none"> • Entrada numérica
3	<p>Entrada de los datos de carga de la herramienta</p> <ul style="list-style-type: none"> • Carga de la herramienta + pieza de trabajo en [kg] • Distancia del centro de gravedad respecto a la brida del robot • Momentos de inercia

12.8.1 Añadir herramienta/pieza nueva

Procedimiento

Añadir una herramienta/pieza

1. Abrir la administración Tool/Base en la unidad de control (1).
Ruta del menú: Tecla de robot > Puesta en servicio > Administración Tool/Base



2. Visualizar las herramientas/piezas de trabajo existentes en la vista general (2).
 En la pestaña **Herramienta/pieza** (3) pulsar el botón **Añadir** (4).



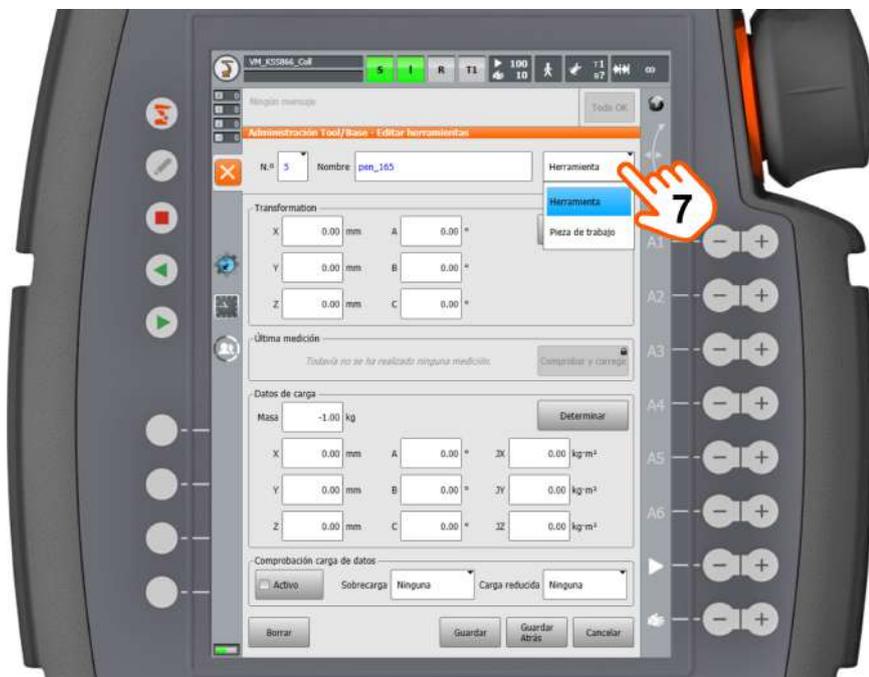
Fig. 12-9: Paso 2

3. Crear de nuevo la herramienta



Fig. 12-10: Paso 3

- Seleccionar un número de herramienta "libre" en la ventana desplegable (5).
 - Introducir el nombre para la nueva herramienta en el campo de introducción (6).
4. Establecer la pieza de trabajo o la herramienta mediante la ventana desplegable (7).



5. Establecer el método de medición deseado a través de la ventana desplegable (8).



Configuración de una nueva estación de robot: Medir la herramienta del robot

Determinación del TCP

- **XYZ de 4 puntos**
- **Referencia XYZ**

Determinación de la orientación

- **ABC universal**
- **ABC de 2 puntos**

12.8.2 Conocer la medición del TCP, método de 4 puntos

Calibración del TCP con el método XYZ de 4 puntos

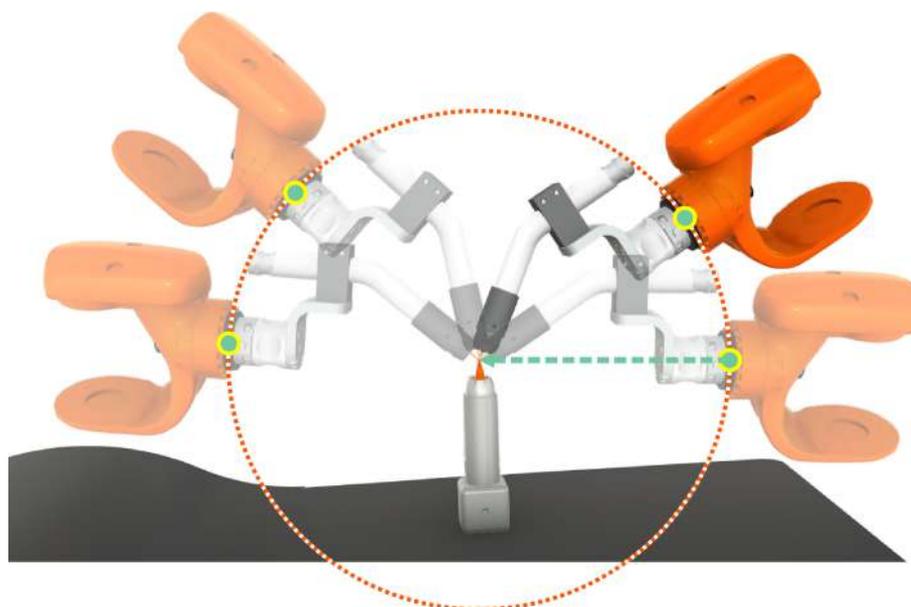
Con el TCP de la herramienta que se va a medir se aproxima un punto de referencia desde 4 direcciones diferentes.

El punto de referencia puede ser cualquiera.

La unidad de control del robot calcula el TCP a partir de las cuatro posiciones de la brida medidas y guardadas.



Las cuatro posiciones de la brida con las que el robot se aproxima al punto de referencia deben estar suficientemente separadas y no encontrarse en un mismo plano.



12.8.2.1 Medir el TCP con el método de 4 puntos

Procedimiento

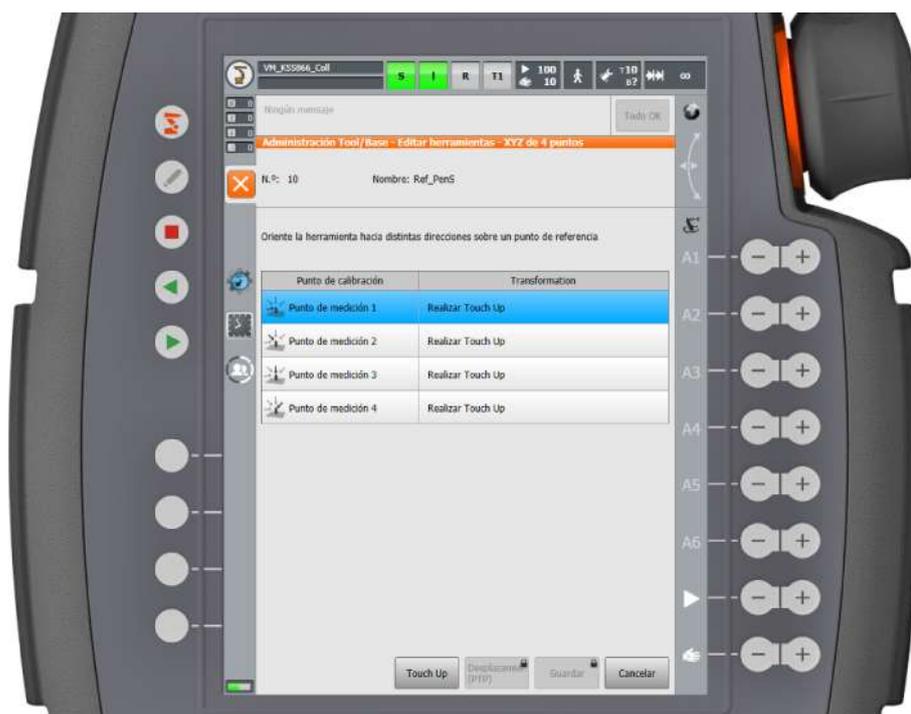


Fig. 12-11: Método de 4 puntos

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

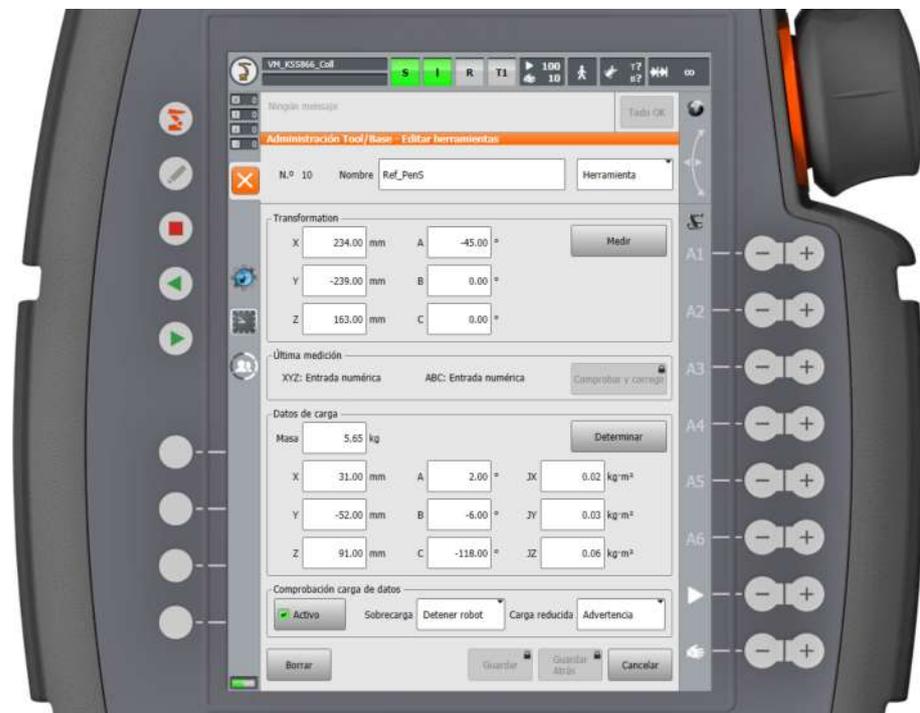
Referencia de formación

Capítulo/apartado

Conocer y medir TOOL > Medir el TCP, conocer el método de 4 puntos

12.8.2.2 Guardar datos de carga de herramienta/pieza de trabajo

Procedimiento



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer y medir TOOL > Guardar datos de herramienta/pieza de trabajo

Configuración de una nueva estación de robot: Medir la herramienta del robot

Encontrará más información sobre este apartado de aprendizaje en la formación en línea.



Formación en línea Cargas en el robot

12.8.3 Conocer la orientación de la herramienta con el método ABC World

Descripción

Los ejes del sistema de coordenadas TOOL se alinean de forma paralela a los ejes del sistema de coordenadas WORLD.

De este modo, la unidad de control del robot conoce la orientación del sistema de coordenadas TOOL.

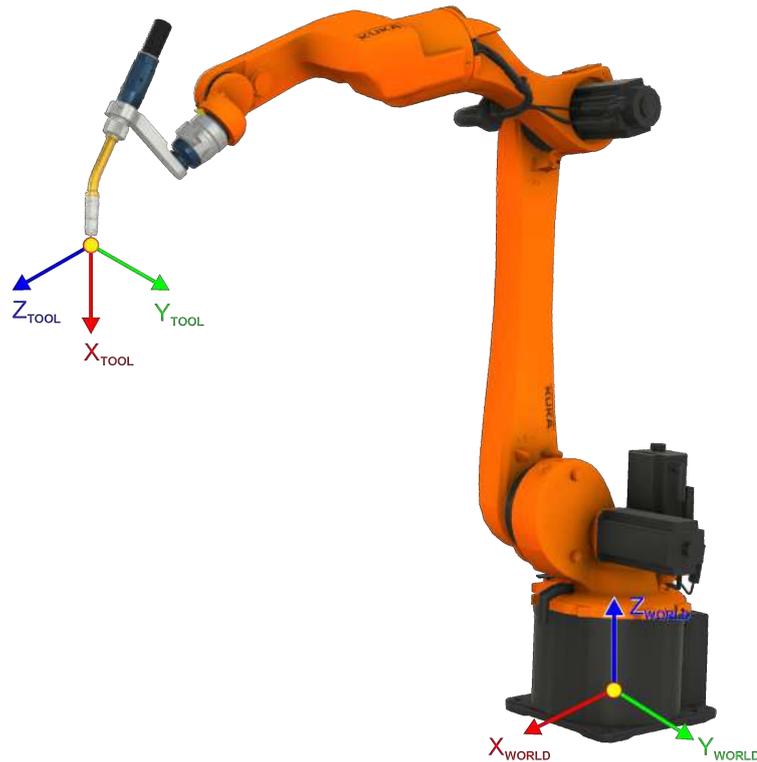


Fig. 12-12: Método ABC World

Variantes

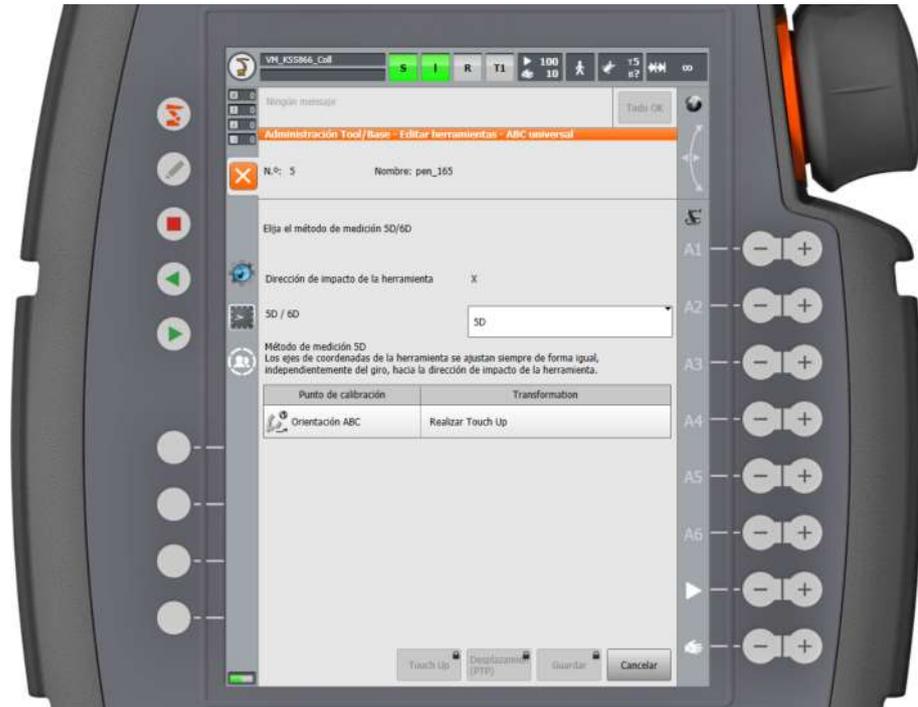
Este método tiene dos variantes:

- **5D**
 - La variante 5D se utiliza para definir la dirección de impacto de la herramienta.
 - En el ajuste estándar, 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.
- Ámbito de aplicación: p. ej. soldadura MIG/MAG, láser o corte por chorro de agua
- **6D**
 - Con la variante 6D es posible establecer todas las direcciones de los ejes, y no solo la dirección de avance
 - Ámbito de aplicación: p. ej. pinzas de soldadura, garras o boquillas de aplicación de pegamentos

12.8.3.1 Orientación de la herramienta, establecer ABC-World

Procedimiento



Configuración de una nueva estación de robot: Medir la herramienta del robot

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer y medir TOOL

> Conocer la orientación de la herramienta con el método ABC World

> Establecer la orientación de la herramienta con el método ABC-World

12.8.4 Conocer la orientación de la herramienta con el método ABC de 2 puntos

A la unidad de control del robot se le comunican los ejes del sistema de coordenadas de herramienta desplazando el robot con el TCP a un punto del eje X negativo y un punto en el nivel XY. Este método se utiliza cuando no solo hay que definir la dirección de avance, sino también todas las direcciones de los ejes.

Puntos de calibración

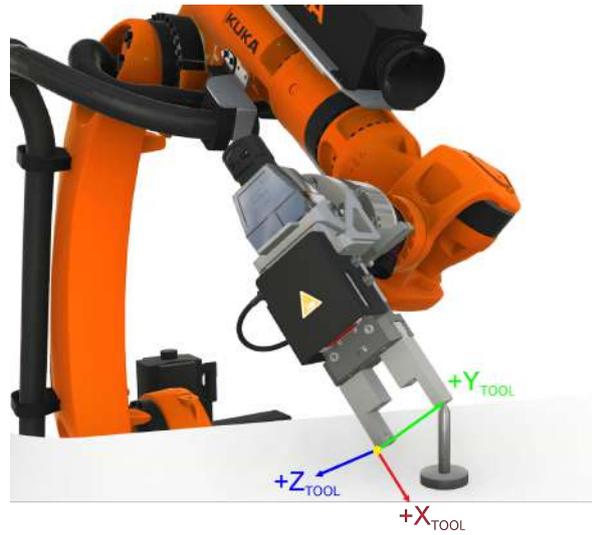
Paso 1:
TCP



Paso 2:
eje X

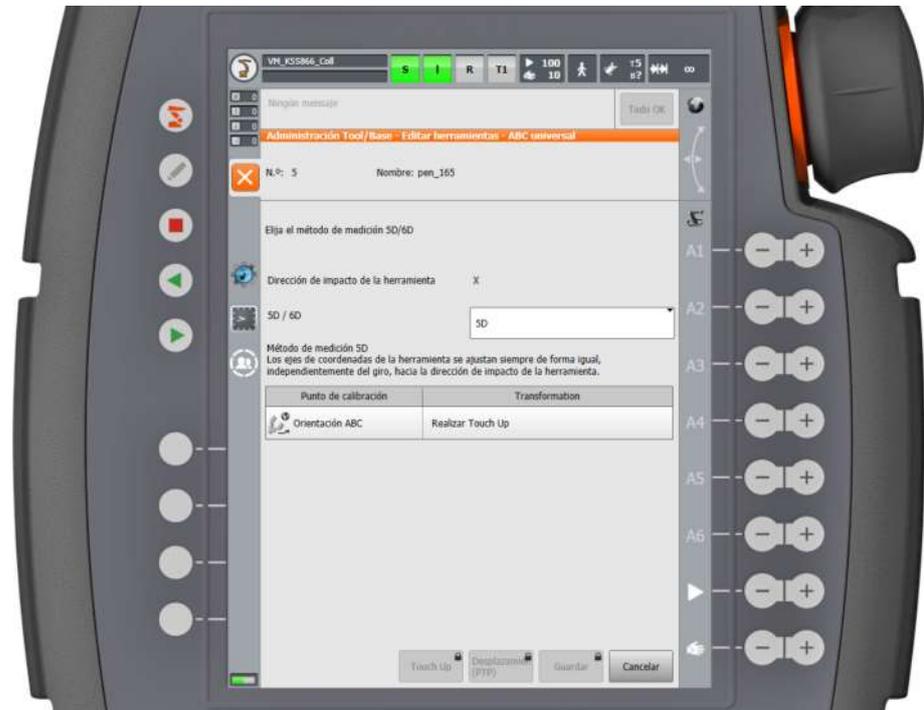


**Paso 3:
nivel XY**



12.8.4.1 Orientación de la herramienta, establecer ABC-World

Procedimiento



Configuración de una nueva estación de robot: Medir la herramienta del robot

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer y medir TOOL

> Conocer la orientación de la herramienta con el método de 2 puntos

> Establecer la orientación de la herramienta con el método de 2 puntos

12.9 Ejercicio: Medir una herramienta nueva

¡Ahora es su turno!



Enunciado

1. Cree una nueva herramienta con el nombre "Clavija_centro".
2. Utilice el número de herramienta _____ .
3. Mida el TCP de la nueva herramienta con el "Método de 4 puntos XYZ".
4. Mida la orientación de la herramienta con el "Método 5D universal ABC".
5. Introduzca los datos de carga de la herramienta indicados aquí u oscile la herramienta con la opción "KUKA.LoadDataDetermination".
6. Compruebe si sus mediciones han ofrecido el resultado deseado. Para ello, gire/oriente su herramienta alrededor de la punta de la herramienta y compruebe la dirección de impacto.

Tarea extra

1. Aprenda un offset para la nueva herramienta "Clavija_centro".
2. Guarde la herramienta en el mismo número de offset que en la administración Tool/Base.

13 Adaptación del programa en caso de cambio de herramienta

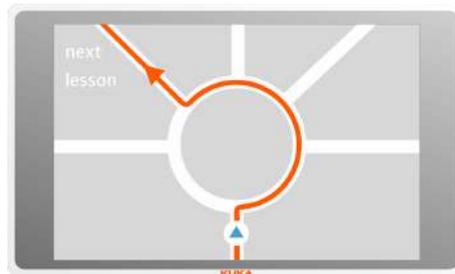
13.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots.
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para corregir y modificar los movimientos del robot existentes.
- Está capacitado para desplazar el robot mediante el sistema de coordenadas WORLD, BRIDA, BASE y TOOL.

¿Qué le espera en esta unidad de aprendizaje?



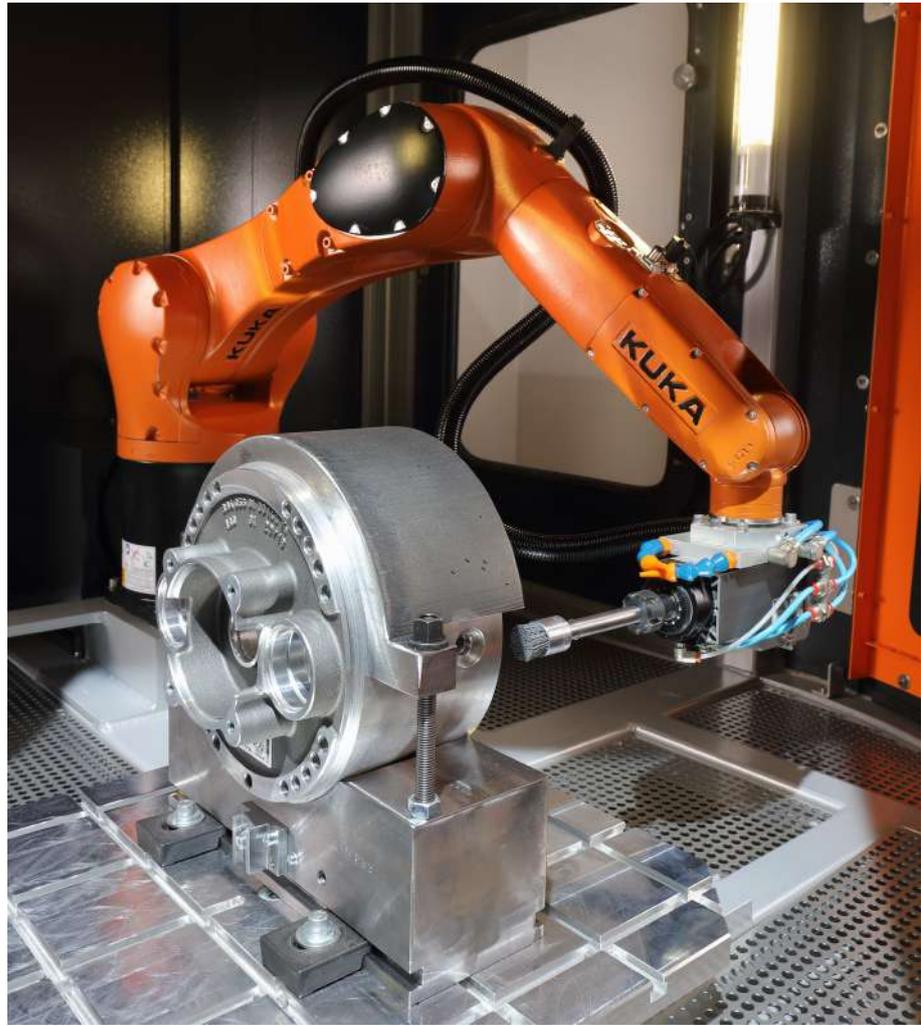
Modificar instrucciones de movimiento

Puede adaptar los movimientos existentes a los cambios:

- Datos de posición
- Velocidad y aceleración
- Tipo de movimiento

13.2 Descripción de la situación

En esta aplicación, el robot pule piezas de carcasa. El pulido se realiza en varias pasadas con diferentes "cabezales de cepillo". Su tarea es adaptar el programa existente a los nuevos cabezales de cepillo. Estos se diferencian principalmente en cuanto a la longitud (TCP).



13.3 Ejercicio: Prueba de programa después del cambio de herramienta

¡Ahora es su turno!

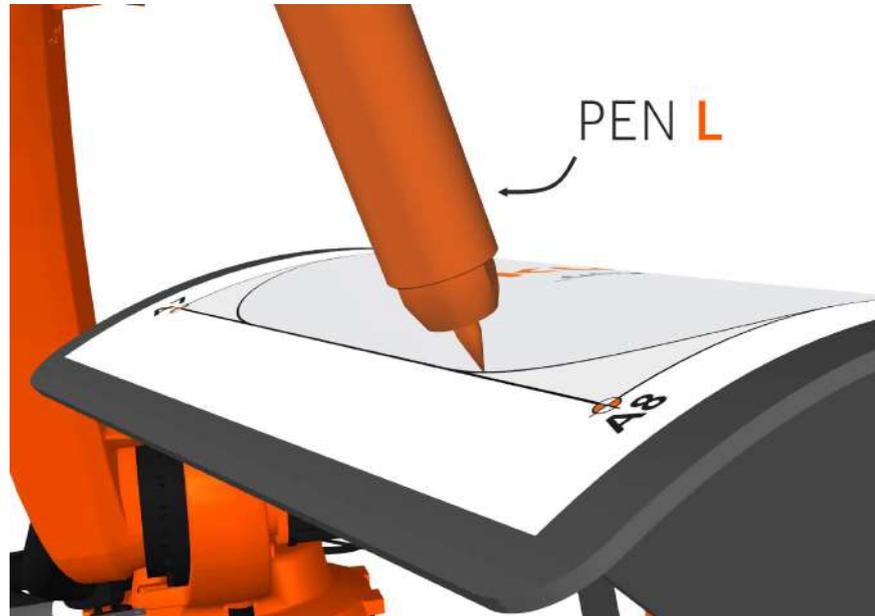
- Sujete la herramienta n.º ____ medida por usted en la garra de formación (Clavija_M).
- Seleccione el programa "Contorno de trayectoria" y pruébelo en el modo de servicio T1.

¿Qué le llama la atención?

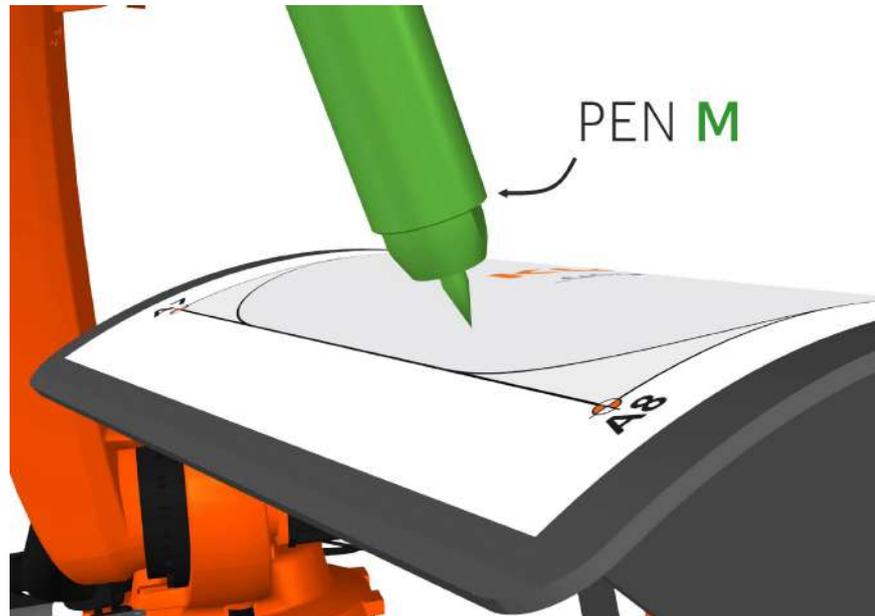
13.4 ¿Qué sucede cuando se ha cambiado la herramienta?

Resultado del ejercicio

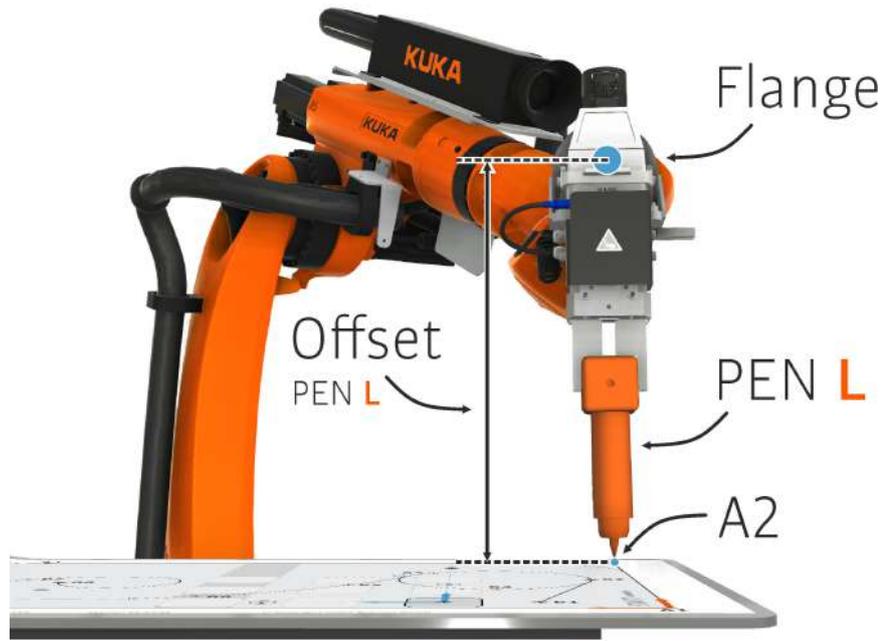
1. El programa "Desplazamiento de trayectoria" se ha probado satisfactoriamente con la herramienta "Clavija_L" utilizada en el programa.



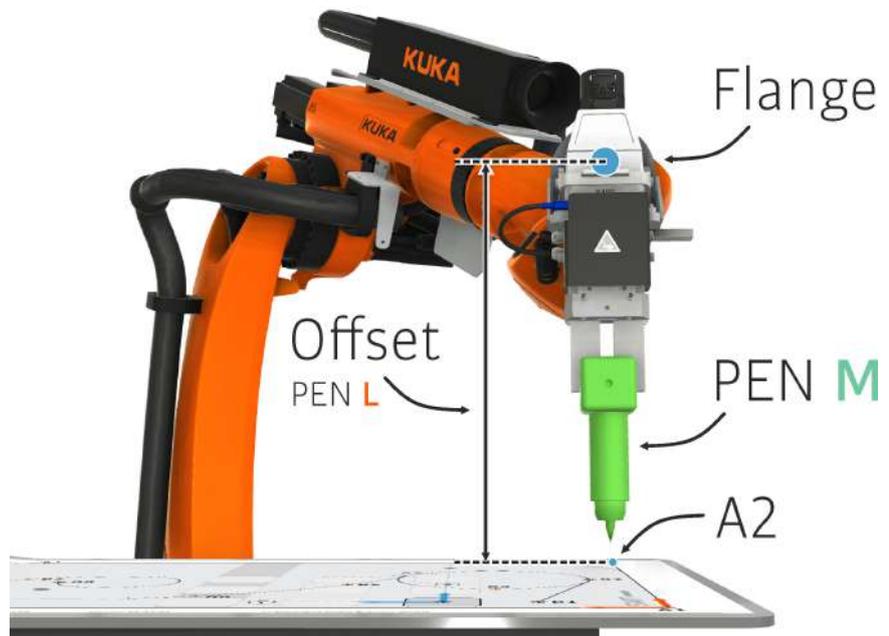
2. La ejecución del programa con la "Clavija_M" da lugar a un offset entre el TCP y la trayectoria.



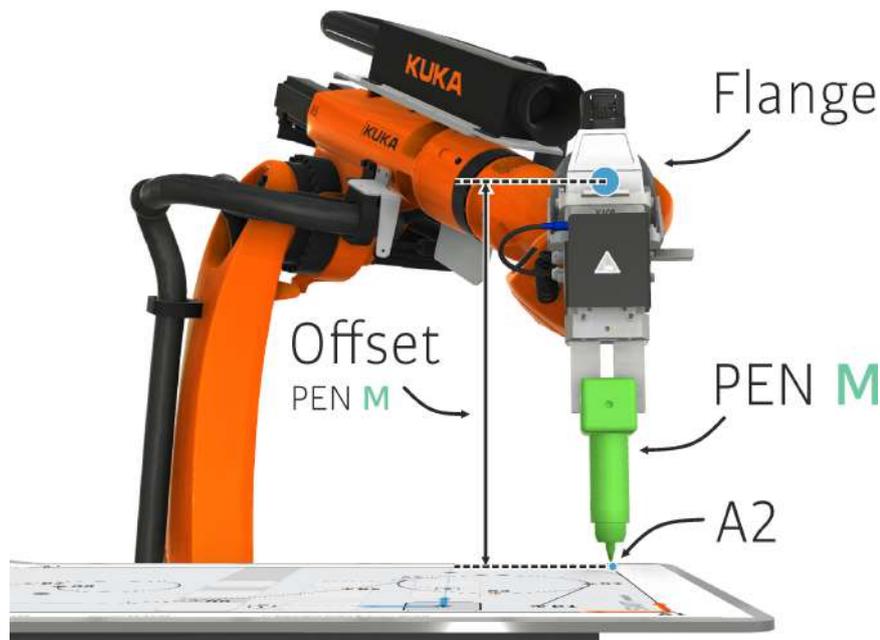
1. El programa con la herramienta "Clavija_L" se duplica.



- Después de un cambio de herramienta, la "Clavija_M" deseada se encuentra en la garra, pero en el conjunto de movimientos todavía está activa la herramienta "Clavija_L". La distancia con respecto a la trayectoria es demasiado grande.



- El robot "conoce" el offset de todas las herramientas medidas entre el TCP y la brida de robot y adapta el movimiento de robot. En nuestro ejemplo de formación, se acerca más a la mesa en la trayectoria después de que la herramienta activa se haya cambiado en el formulario inline a "Clavija_M".



No se requiere **ninguna** programación posterior.

- El programa duplicado se comprueba finalmente con la nueva herramienta.

13.5 Modificar instrucciones de movimiento

Existen los motivos más diversos para modificar los comandos de movimiento:

Ejemplos de motivos	Modificación a llevar a cabo
<ul style="list-style-type: none"> La posición de la pieza que se debe agarrar cambia. La posición de uno de los cinco orificios en la ejecución de los trabajos cambia. Una costura soldada se debe acortar. 	<ul style="list-style-type: none"> Modificación de datos de posición
<ul style="list-style-type: none"> La posición del palet cambia. 	<ul style="list-style-type: none"> Modificación de los datos vectoriales: Base
<ul style="list-style-type: none"> Por error se ha realizado el aprendizaje de una posición con la base o con el TOOL incorrectos. 	<ul style="list-style-type: none"> Modificación de los datos vectoriales: Base y/o Tool con actualización de la posición
<ul style="list-style-type: none"> El procesamiento es demasiado lento: se debe mejorar el tiempo de ciclo. 	<ul style="list-style-type: none"> Modificación de los datos de movimientos: Velocidad, aceleración Modificación del tipo de movimiento



Fig. 13-1: Campo de notas

13.5.1 Modificación de los datos de movimiento

Efectos de modificar los datos de movimiento

Al modificar la velocidad o la aceleración, cambia el perfil de traslación. Ello puede repercutir en el proceso de fabricación, sobre todo en las aplicaciones de trayectoria:

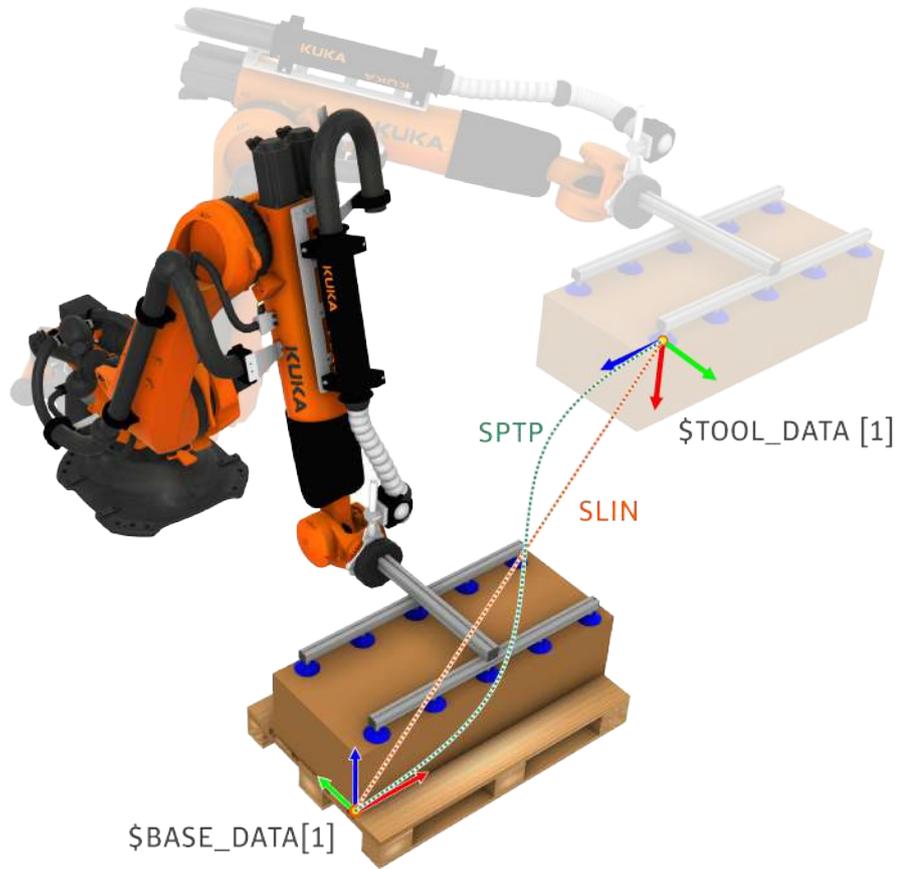
- Espesor de un cordón de pegado.
- Calidad de una costura soldada.



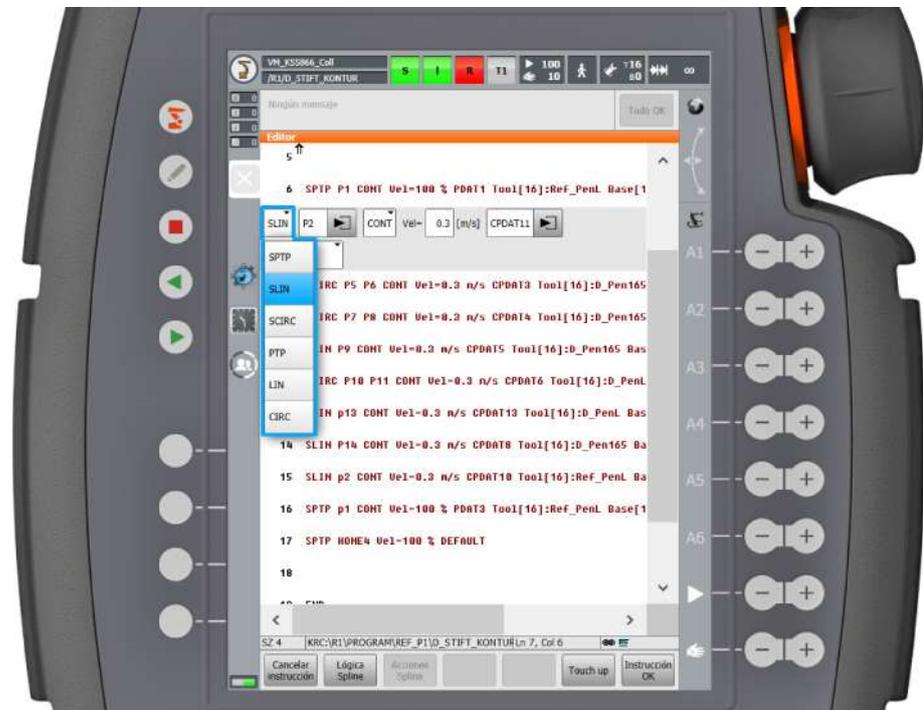
13.5.2 Modificación del tipo de movimiento

Efectos de modificar el tipo de movimiento

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



Procedimiento



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

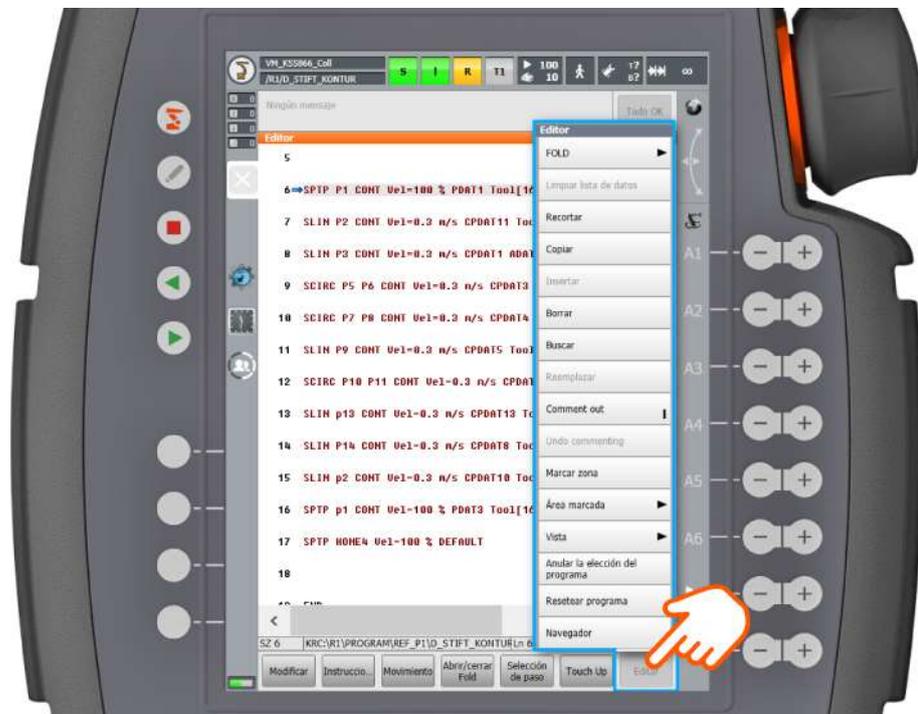
Referencia de formación

Capítulo/apartado

Modificar los puntos de movimiento programados > Modificación de la trayectoria del movimiento

13.5.3 Manejo de puntos de movimiento existentes

Procedimiento



Podrá encontrar más información en la documentación del cliente.



Libro

KUKA System Software 8.6

Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

Programación con formularios inline > **Modificar parámetros de movimiento**

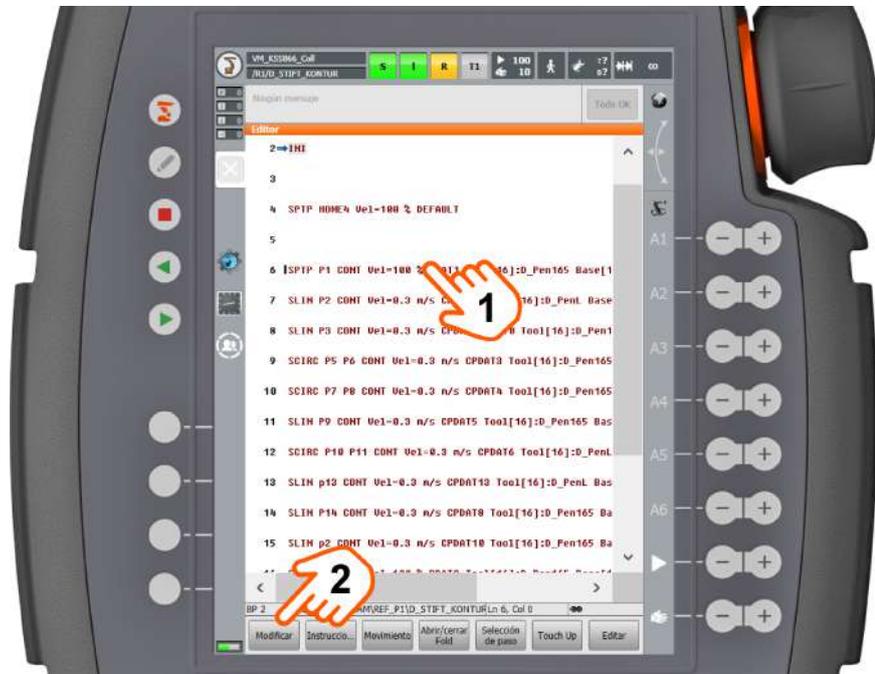
13.6 Ejercicio: Adaptación del programa en caso de cambio de herramienta

¡Ahora es su turno!



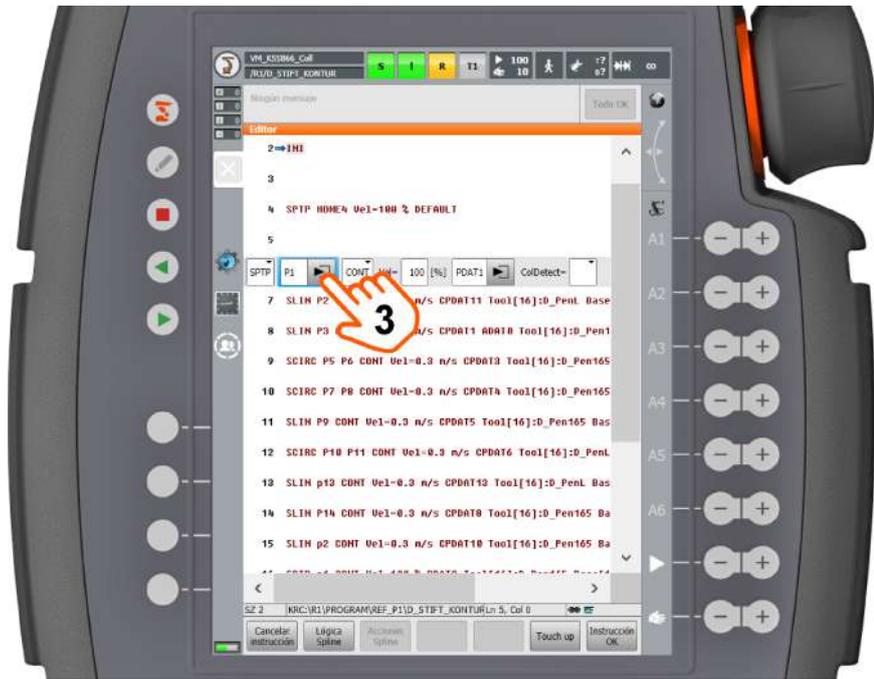
Enunciado

1. Duplique el programa "Contorno de trayectoria" y guárdelo con un nombre nuevo.
2. Seleccione el programa duplicado.
3. Pulsar en el primer punto a modificar (1). (Aviso: cursor intermitente)

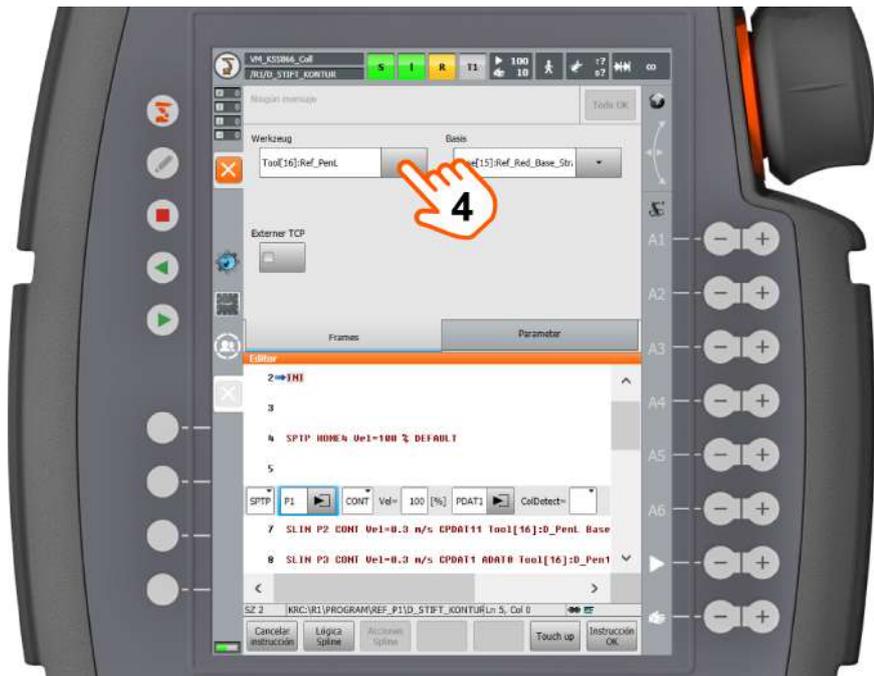


Abrir el formulario inline con el botón "Modificar" (2).

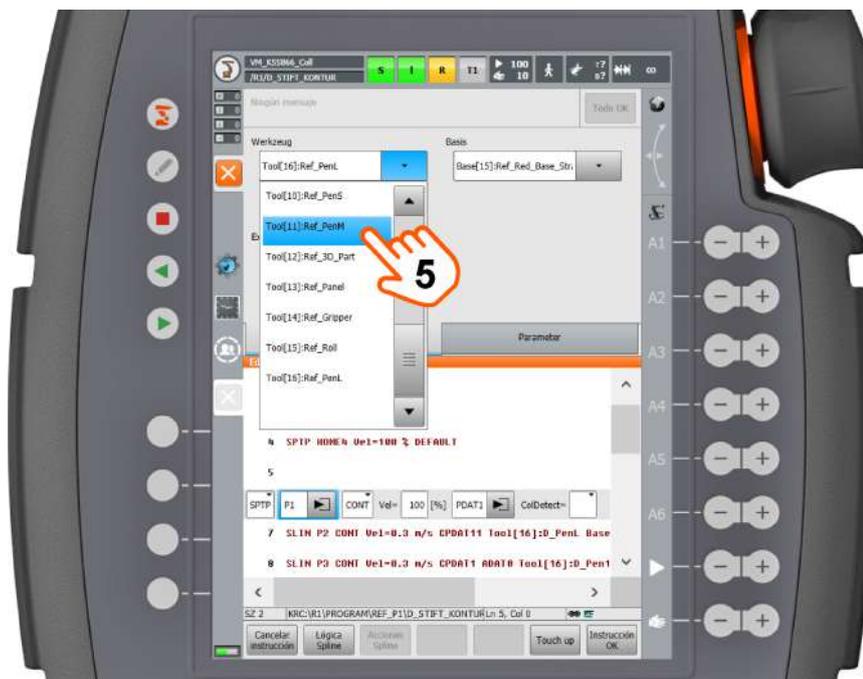
4. Se abre el formulario inline. Pulsar el símbolo de flecha (3) junto al nombre del punto.



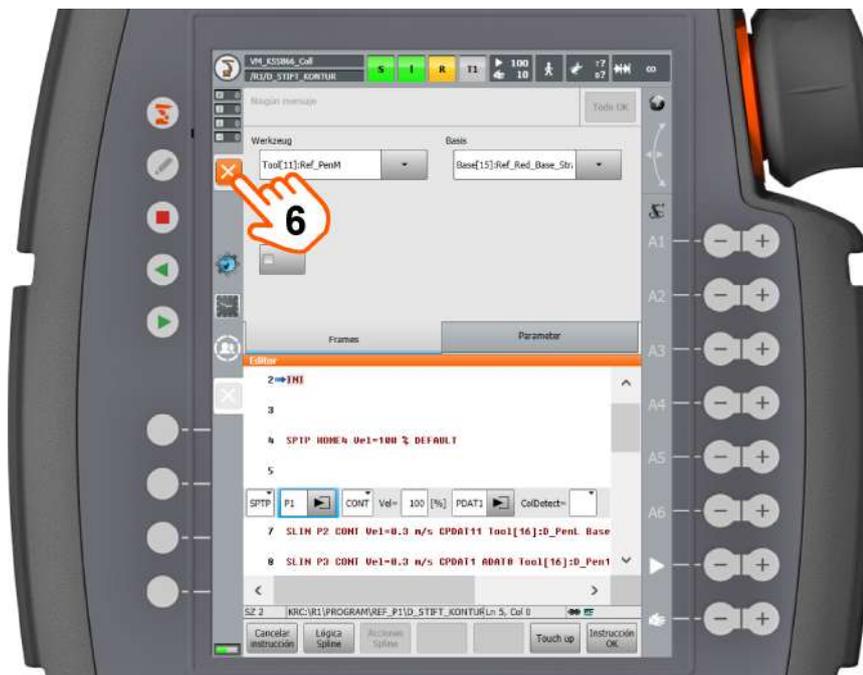
5. En la "ventana Frame" abierta, abrir la ventana desplegable "Herramientas" (4).



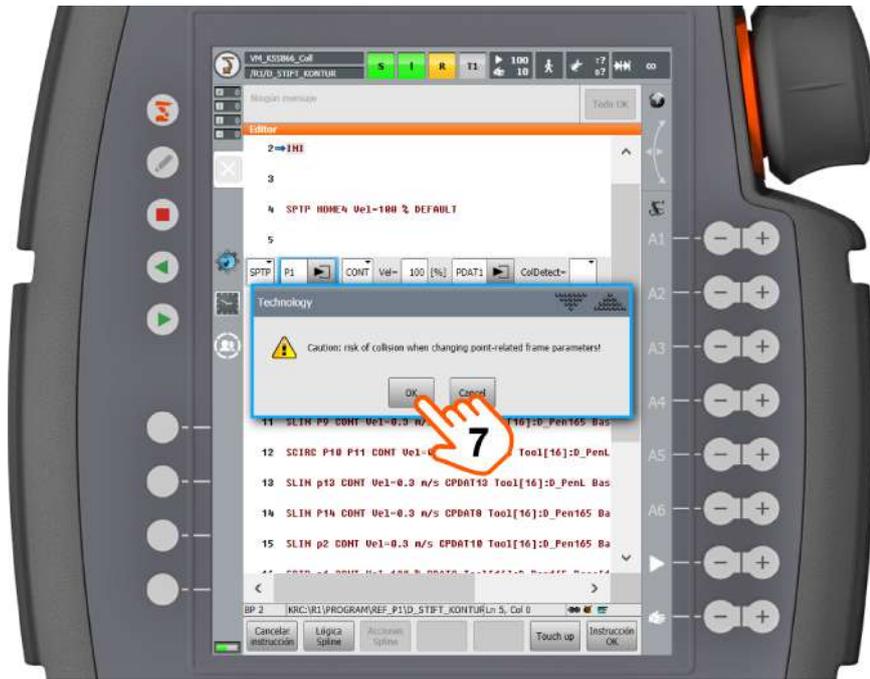
6. Seleccionar la herramienta nueva y medida (5).



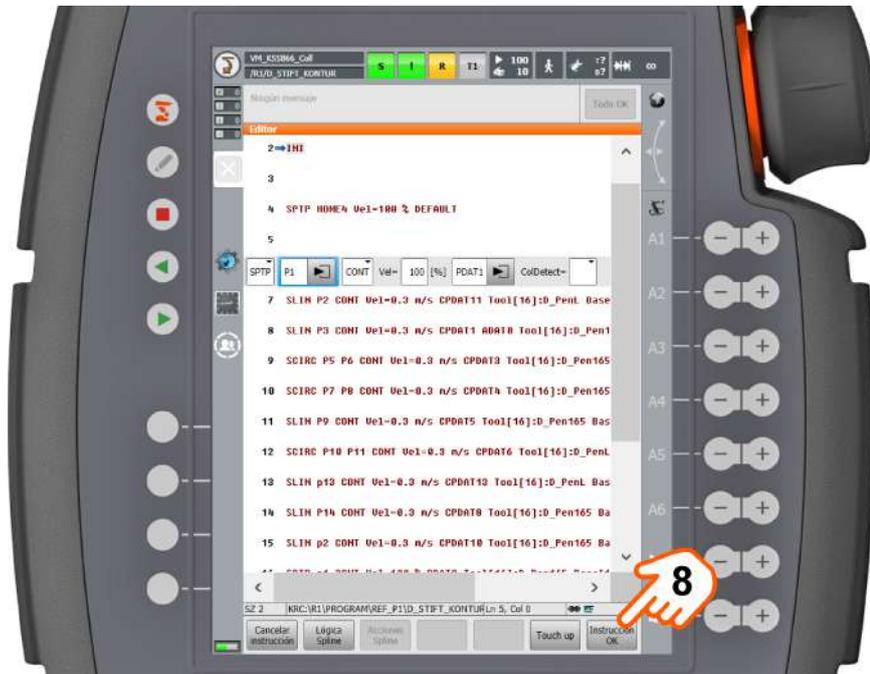
7. Cerrar la "ventana Frame" (6).



8. Confirmar el mensaje "Atención al modificar los parámetros Frame referidos al punto, existe peligro de colisión" con el botón "OK".



9. Cerrar el formulario inline con el botón "Comando OK".



10. Repetir los pasos para todos los conjuntos de movimiento.
11. Deje que el programa se ejecute en el modo de servicio T1 con override de programa bajo y compruebe si el TCP sigue el contorno de trayectoria.

Lo que se debe saber tras el ejercicio:



¿Por qué deben adaptarse los parámetros referidos al frame en el programa correspondiente después de un cambio de herramienta?



Fig. 13-2: Campo de respuesta

Al crear un programa, ha olvidado introducir los datos de la herramienta y de la base correctos en los parámetros referidos al frame de los conjuntos de movimientos. Desea corregirlo posteriormente. ¿Qué debe tenerse en cuenta?

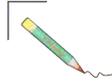


Fig. 13-3: Campo de respuesta

¿Qué debe tenerse en cuenta al modificar la posición Home?



Fig. 13-4: Campo de respuesta

13.7 Modificación de datos de posición

Efectos al modificar los datos de posición

El punto recibe coordenadas nuevas, porque con "Touch-Up" se han actualizado los valores. De este modo, solo se ha modificado el juego de datos del punto y el resto del programa permanece igual.

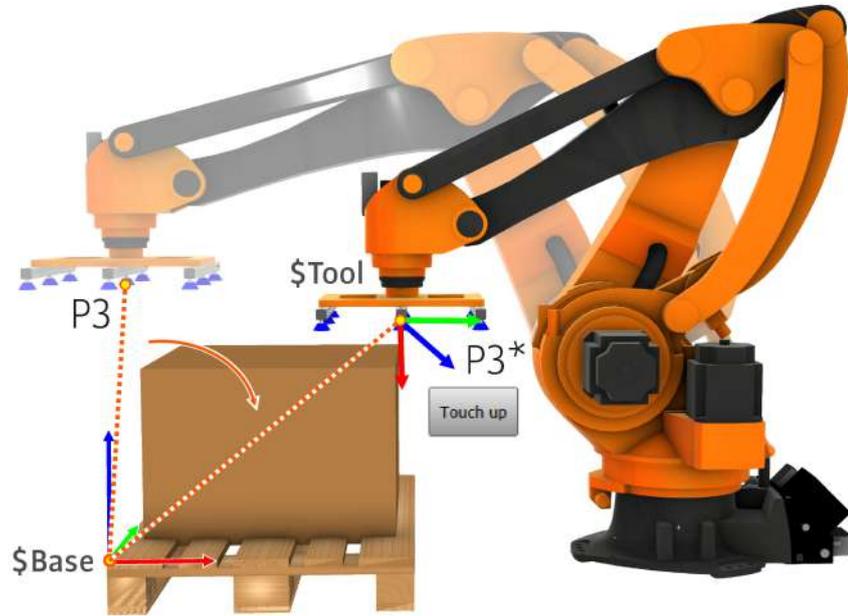
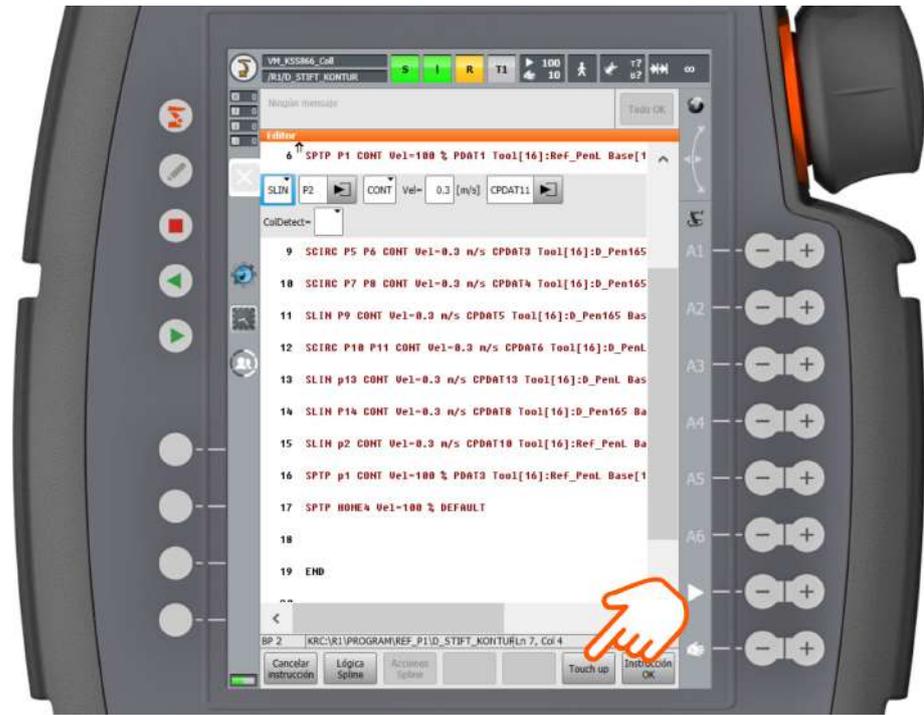


Fig. 13-5: Modificación de la posición del robot con "Touch-Up"

AVISO
Las coordenadas anteriores se sobrescriben y, por consiguiente, dejan de estar disponibles.

13.7.1 Modificación de datos de posición

Procedimiento



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Modificar movimientos programados > Modificación de datos de posición

14 Añadirá puntos de movimiento en un programa

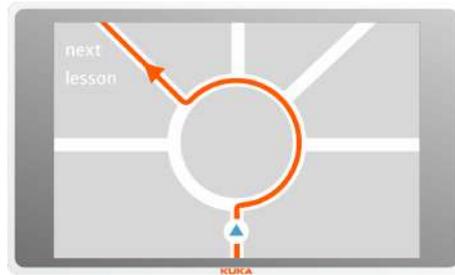
14.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots.
- Está capacitado para desplazar el robot mediante el sistema de coordenadas WORLD, BRIDA, BASE y TOOL.
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para corregir y modificar los movimientos del robot existentes.

¿Qué le espera en esta unidad de aprendizaje?



Añadirá puntos de movimiento en un programa

- Podrá añadir puntos a los programas existentes mediante formularios inline.
- Programará los movimientos del robot SPTP y SLIN.

14.2 Descripción de la situación

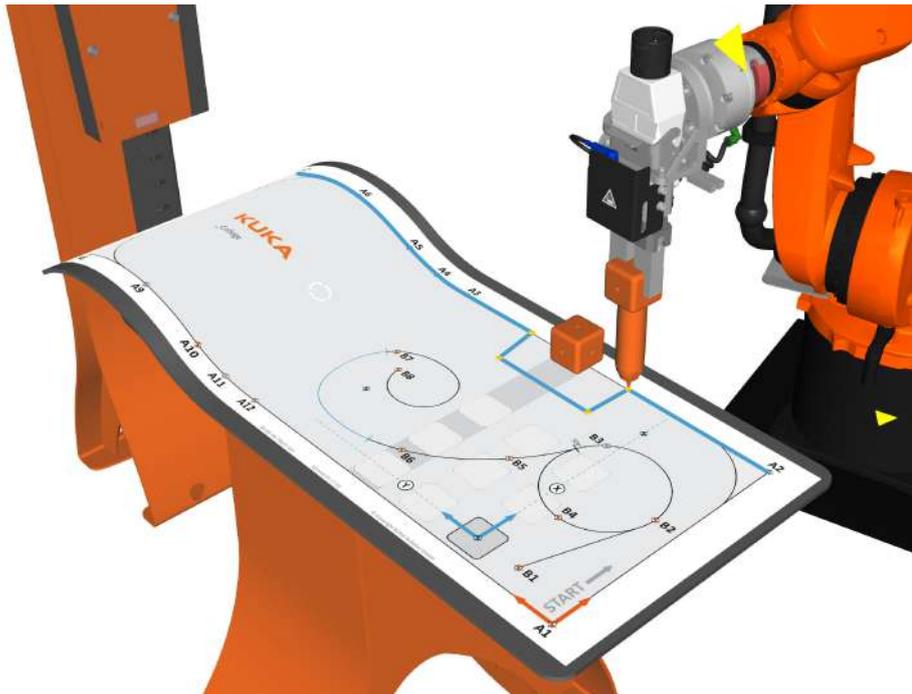
En su estación de robot se sueldan perfiles de bastidor de un tamaño de forma automatizada. De forma complementaria para ello, ahora deben ser posibles otros tamaños y variantes. Para ello, cree variantes de programas ampliando puntos de programa adicionales sobre la base del programa existente.



En la situación representada se utiliza la opción de pago **KU-KA.ready2pilot**.

Desplazamiento de trayectoria con obstáculo

En el programa existente "Desplazamiento de trayectoria", un cubo colocado entre los puntos A2 y A3 debe constituir una variante de fabricación. El programa debe adaptarse de forma correspondiente.



14.3 ¿Cómo se añaden nuevos comandos de movimiento?

Instrucciones

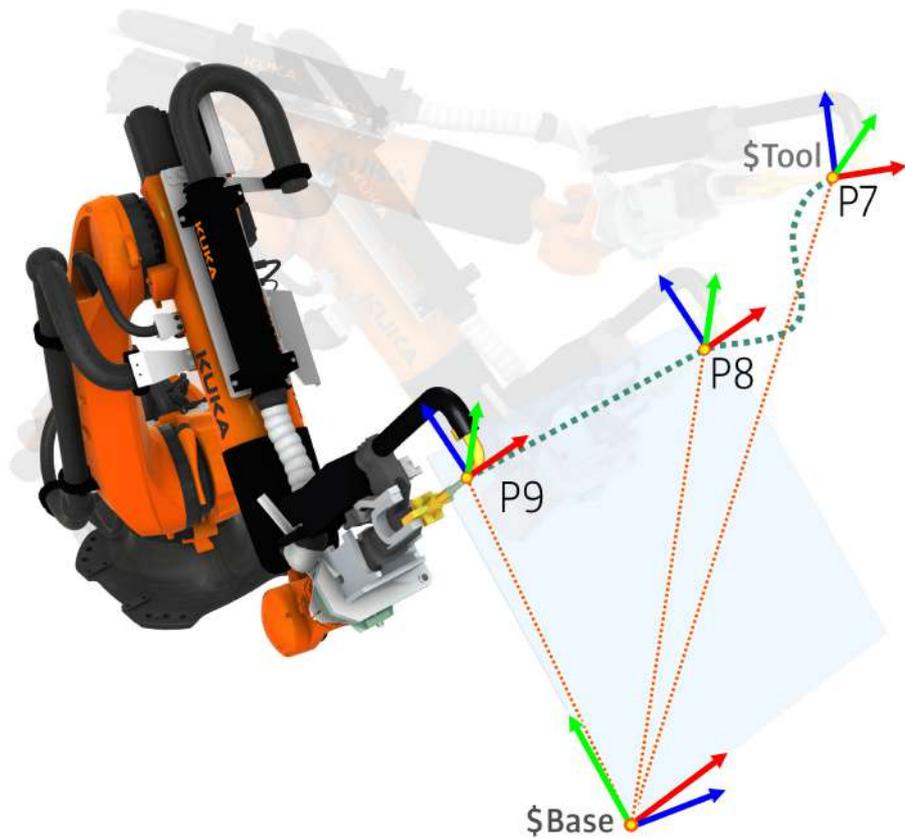


1. **Duplicar el programa**
Duplicar simplifica el proceso de creación de programas para el nuevo componente.
2. **Añadir nuevos conjuntos de movimientos**
El programa duplicado se amplía, mientras que el programa existente permanece intacto.
Al añadir nuevos comandos de movimiento en el lugar correcto, el trabajo se reduce al mínimo.
3. **Someter a test el programa**

14.4 ¿Qué información es necesaria para un movimiento de robot?

Programación de un movimiento de robot

Quando se deben programar los movimientos del robot, surgen numerosas preguntas:



Pregunta	Solución	Palabra clave
¿Cómo detecta el robot sus posiciones?	Se memoriza la correspondiente posición de la herramienta en el espacio (posición del robot según el sistema Tool y Base ajustado).	POS E6POS
¿Cómo sabe el robot de qué manera debe moverse?	Por la indicación del modo de movimiento: punto a punto, lineal o circular.	SPTP SLIN SCIRC
¿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 ahorrar tiempo de ciclo, también se pueden aproximar puntos. En este caso no se realiza ninguna parada exacta.	CONT
¿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.	ORI_TYPE
¿El robot detecta obstáculos?	No, el robot sigue la trayectoria programada "sin desviarse". El programador será el responsable de garantizar que no se produzcan colisiones. No obstante, existe la posibilidad de implementar un control contra colisiones para la protección de máquina.	Control de colisiones

Procedimiento

1. Duplicar el programa
2. Seleccionar programa
(>>> 4.3.1 "Seleccionar programas de robot" Página 35)
3. Efectuar el desplazamiento SAK
(>>> 4.7 "Conocer el desplazamiento de inicialización" Página 39)
4. Pulsar el último conjunto de movimientos después del que se debe insertar el nuevo conjunto de movimientos (cursor) (1).
Mediante el botón "Desarrollo del movimiento" (2) se añade un nuevo conjunto de movimientos después del conjunto de movimientos marcado.

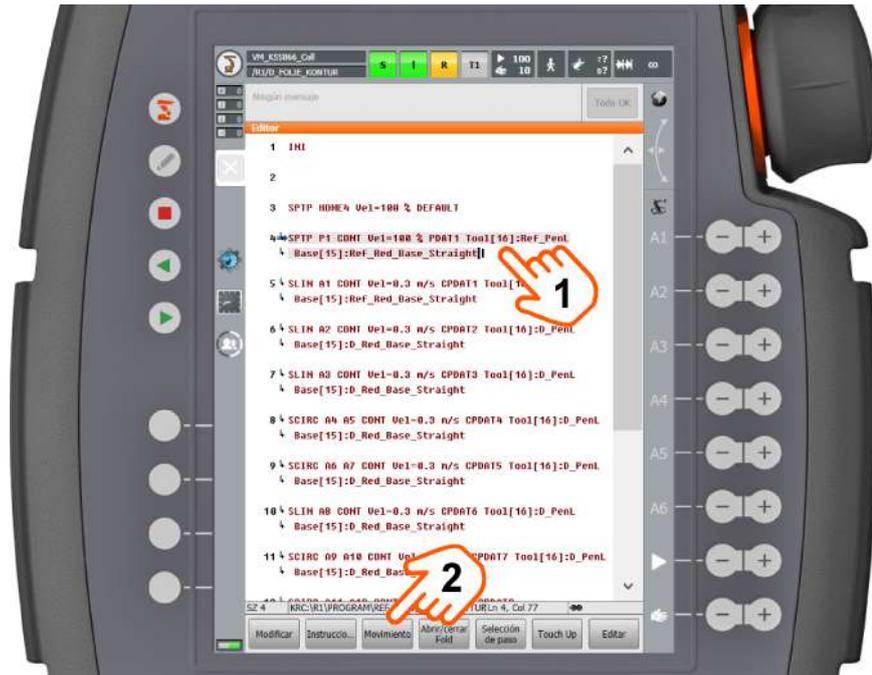


Fig. 14-1

5. Seleccionar la trayectoria del movimiento en la primera ventana desplegable (3) del formulario inline.

Añadirá puntos de movimiento en un programa

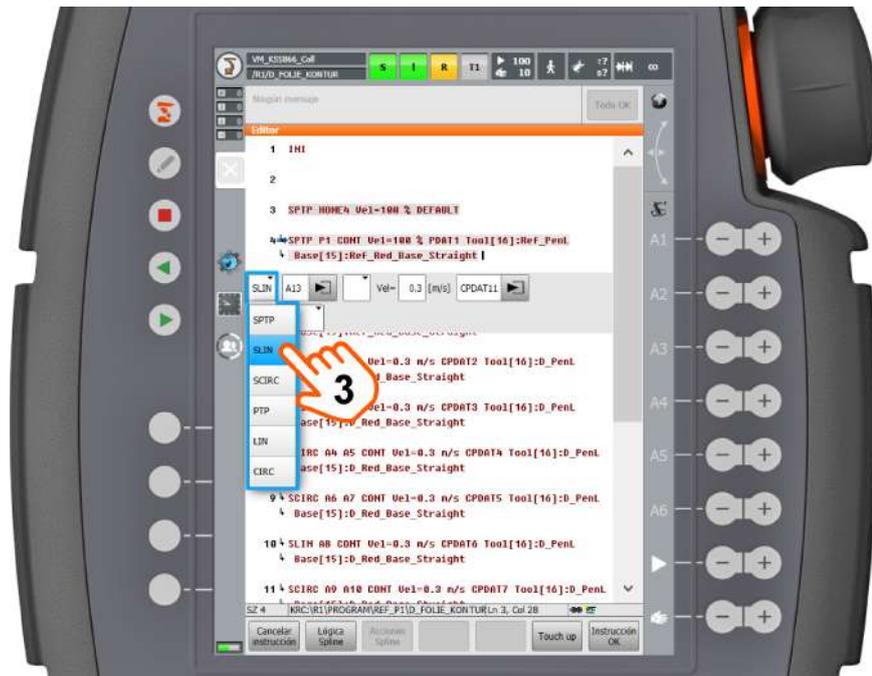


Fig. 14-2

- Introducir los nombres de los puntos en el siguiente campo (4).

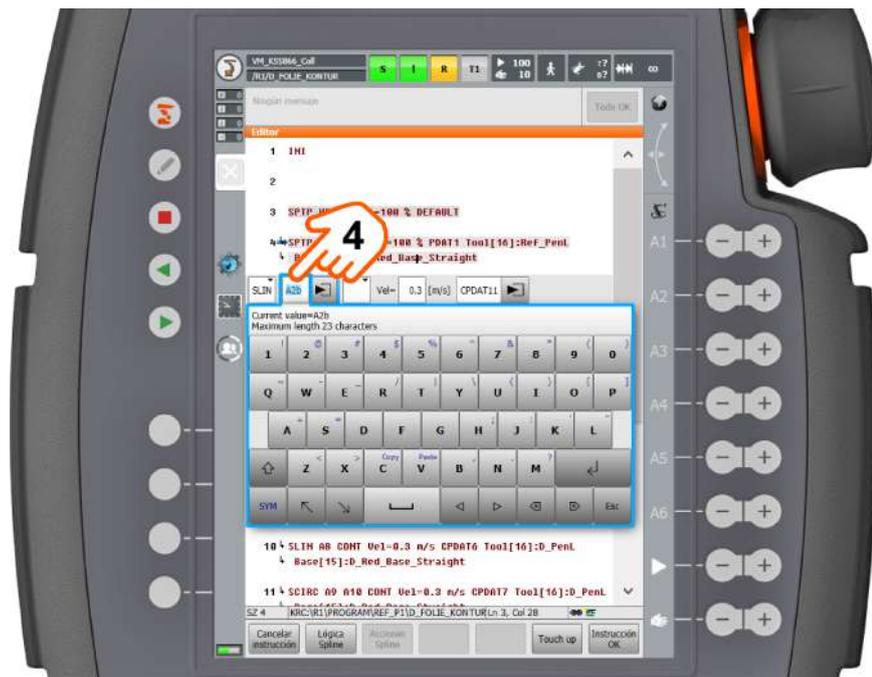


Fig. 14-3

- A través del símbolo de flecha junto al nombre del punto (5) abrir las ventanas Datos del Frame y establecer la herramienta (6) y la base (7).

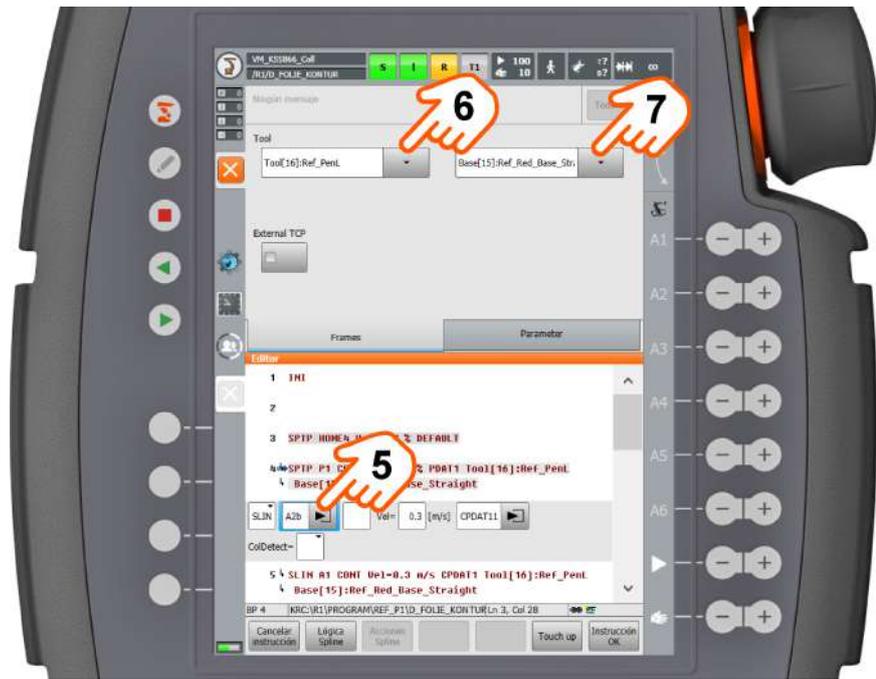


Fig. 14-4

8. Introducir la velocidad de TCP deseada en el campo "VEL" (8).



Fig. 14-5

9. Desplazar el robot hasta la posición deseada y aceptar el desarrollo del movimiento con el botón "Comando OK".

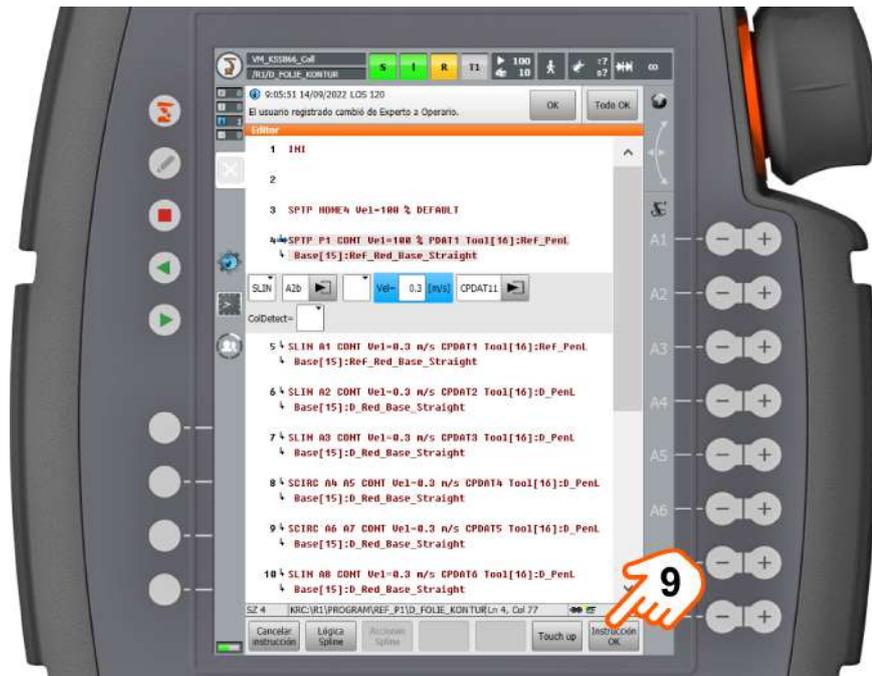


Fig. 14-6

El formulario inline se cierra y aparece en el listado de programas.
10. Repetir los pasos para otros movimientos.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Creación de movimientos con optimización del tiempo de ciclo (SPTP)

Crear movimientos de trayectoria (SLIN, SCIRC)

14.5 Ejercicio: Adaptar el contorno de trayectoria

¡Ahora es su turno!



Enunciado

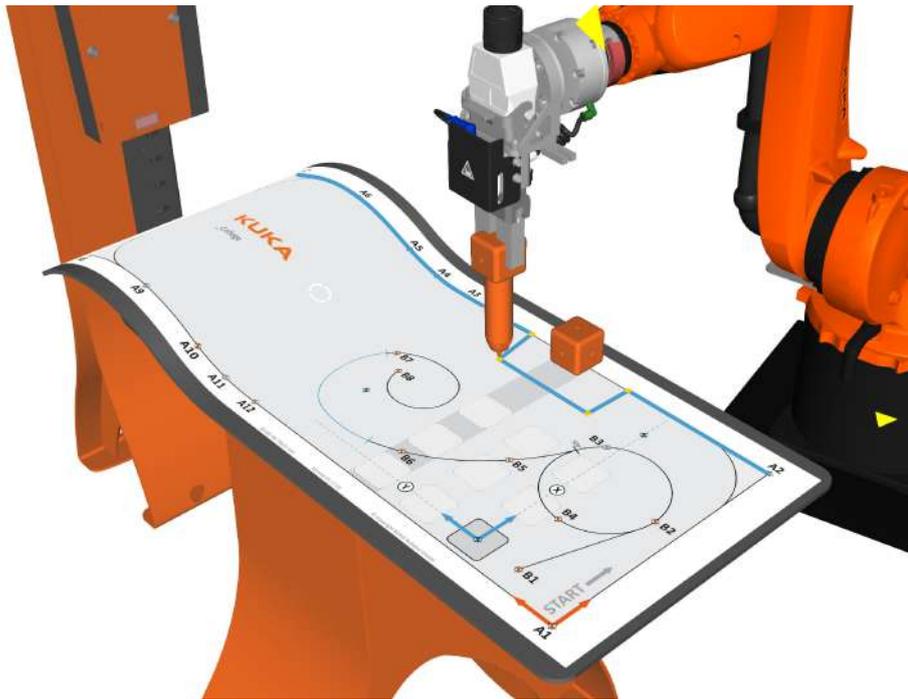


Fig. 14-7

1. Duplique el **programa:** _____ y asigne un **nuevonombre del programa:** _____
2. Coloque un cubo del depósito de cubos sobre el contorno de trayectoria según las especificaciones del formador.
3. Inserte comandos de movimiento adicionales en el programa para que el cubo sea sorteado en la trayectoria. Para ello, preste atención a la posición del cursor negro; un nuevo conjunto de movimientos se inserta siempre como una línea nueva debajo del conjunto de movimientos en el que se encuentra el cursor negro.

14.6 Ejercicio: Adaptar el contorno de trayectoria

Lo que se debe saber tras el ejercicio:



¿Cómo se puede guardar la coordenada después de crear un conjunto de movimiento?

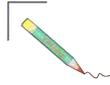


Fig. 14-8: Campo de respuesta

¿Cómo se pueden crear copias de seguridad de un módulo?



Fig. 14-9: Campo de respuesta

15 Configuración de una nueva estación de robot: Unidad de calibración para BASE

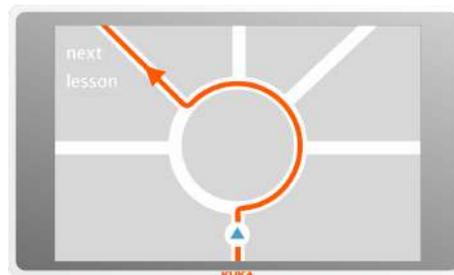
15.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para desplazar el robot mediante el sistema de coordenadas WORLD, BRIDA, BASE y TOOL.

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una estación de robot: Unidad de calibración para BASE

- Conocerá las ventajas de una base medida.
- Medirá los sistemas de coordenadas de referencia de base con el método de 3 puntos.

15.2 Descripción de la situación

Para las lavadoras, se realiza la fabricación final de motores en una estación de robot. Para ello, los componentes individuales, como aquí el arrollamiento del estator, se suministran al robot a través de cajas. El robot los agarra con precisión segura a través de la cuadrícula predefinida dentro de estas cajas. Un sistema de coordenadas BASE predefinido facilita la creación y modificación de las posiciones de agarre.



15.3 Ventajas de una base medida

Situación inicial

Las memorias USB deben ser recogidas por el robot desde una rejilla de almacenamiento para su posterior procesamiento. A la hora de aplicar el programa, hay que establecer una herramienta definida y una base de referencia. Esto ofrece varias ventajas.

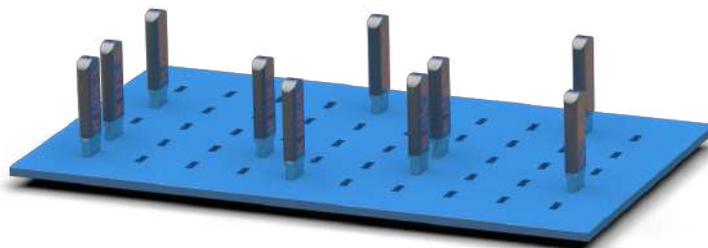


Fig. 15-1: Ejemplo de salida

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.

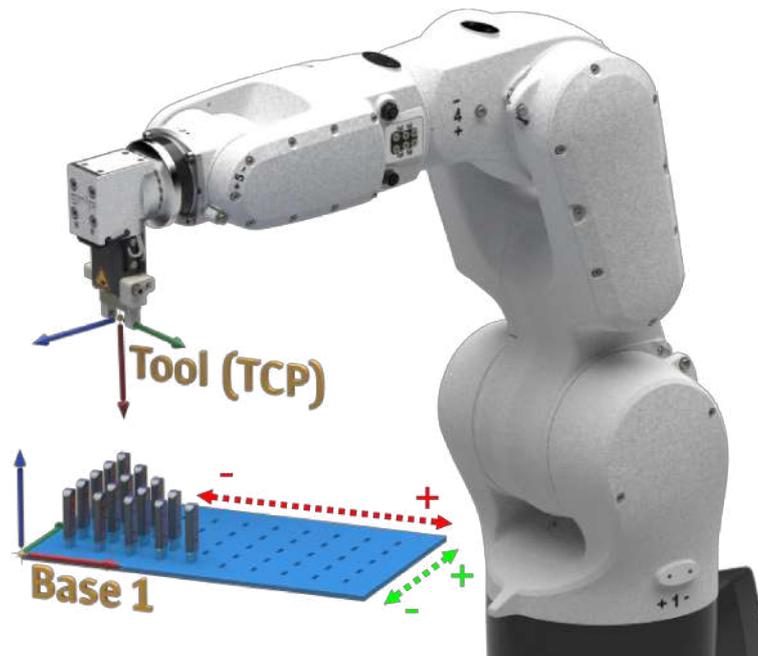


Fig. 15-2: Ventajas de la medición BASE: Dirección de desplazamiento

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

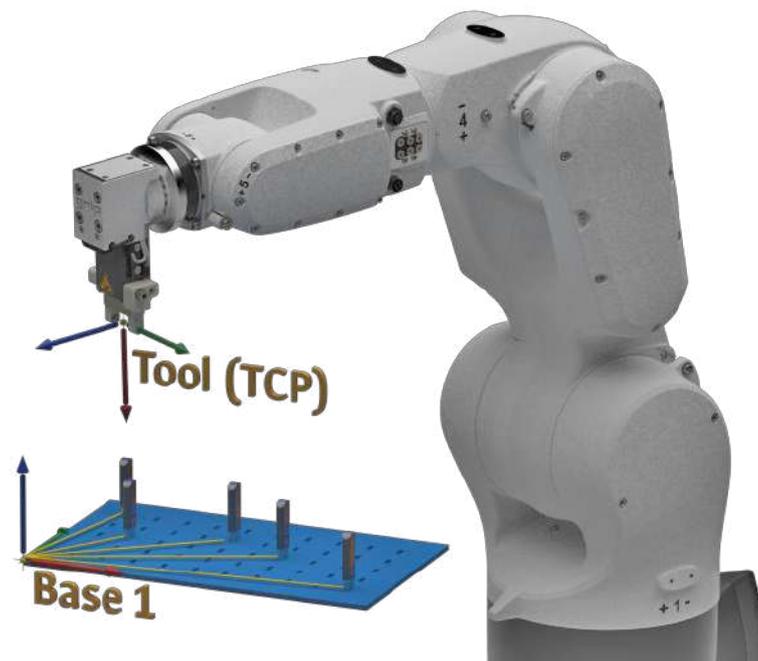


Fig. 15-3: Ventajas de la medición BASE: Referencia al sistema de coordenadas deseado

- Corrección/desplazamiento sencillo del sistema de coordenadas:
Puntos pueden ser programados por aprendizaje en relación a la base. Si es necesario desplazar la base, p. ej. porque la superficie de trabajo se ha desplazado, se desplazan también los puntos y no es necesario programarlos de nuevo.

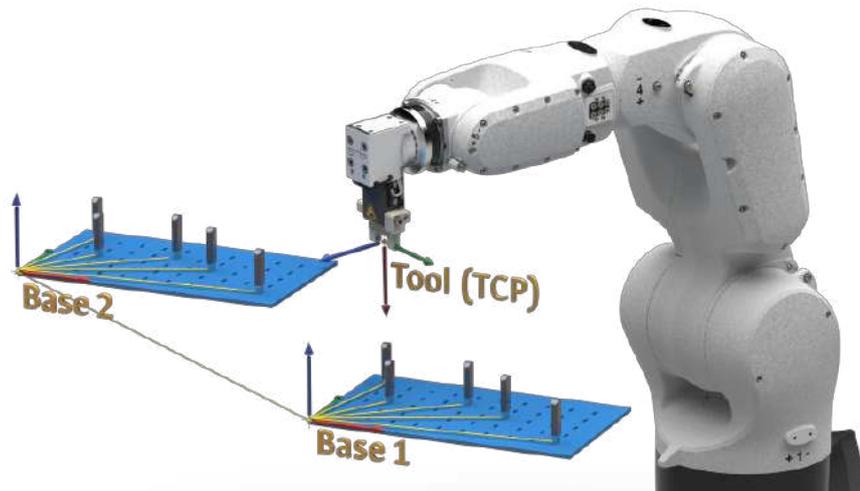


Fig. 15-4: Ventajas de la medición BASE: Corrimiento del sistema de coordenadas de base

- 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.

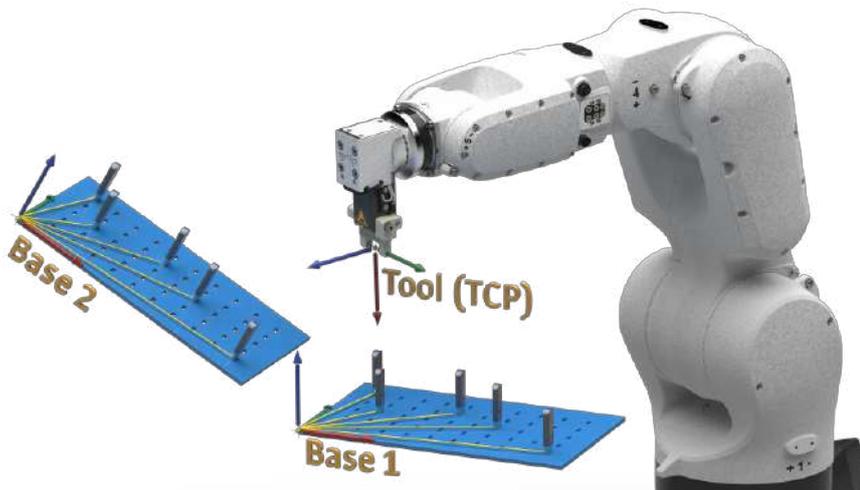


Fig. 15-5: Ventajas de la medición BASE: Utilización de varios sistemas de coordenadas de base

15.4 Opciones de la medición de base

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

Métodos	Descripción
Método de 3 puntos	<ol style="list-style-type: none"> Definición del origen Definición de la dirección X positiva Definición de la dirección Y positiva (plano XY)
Método indirecto//	<ul style="list-style-type: none"> 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.
Entrada numérica	<ul style="list-style-type: none"> Entrada directa de valores para la distancia al sistema de coordenadas universales (X, Y, Z) y del giro (A, B, C).

15.5 Calibración de BASE con el método de 3 puntos

Procedimiento

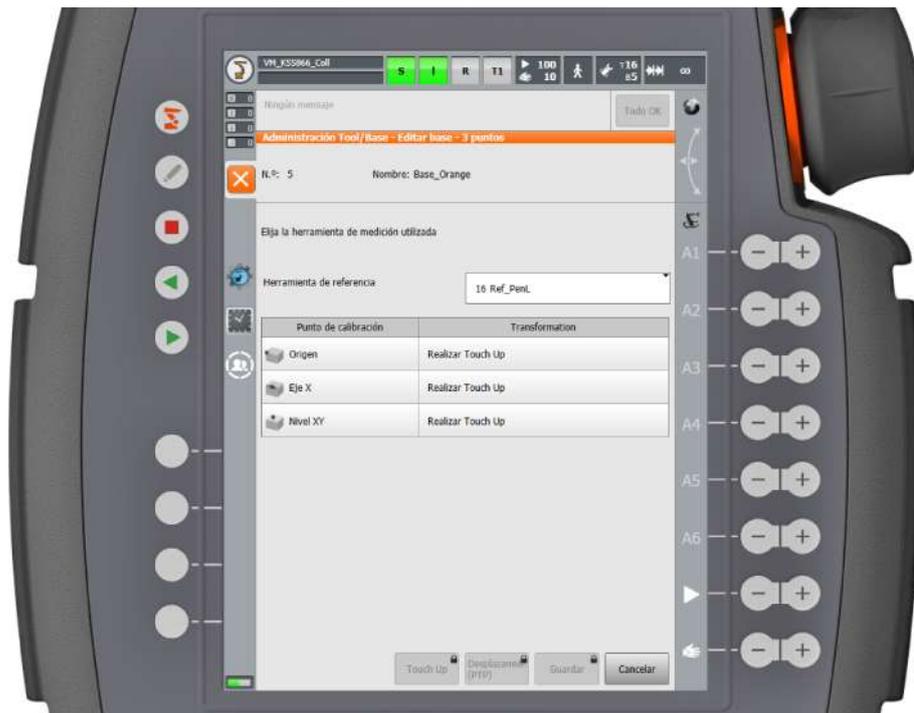


Fig. 15-6

Configuración de una nueva estación de robot: Unidad de calibración para BASE

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

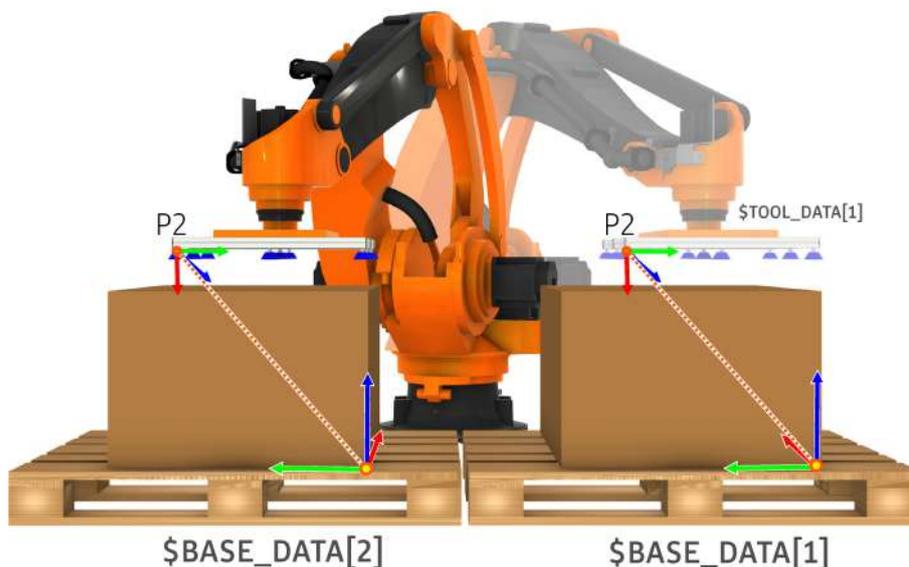
Capítulo/apartado

Conocer y medir BASE > Añadir base/herramienta fija nueva > Método de 3 puntos

15.6 Modificación de los datos del Frame

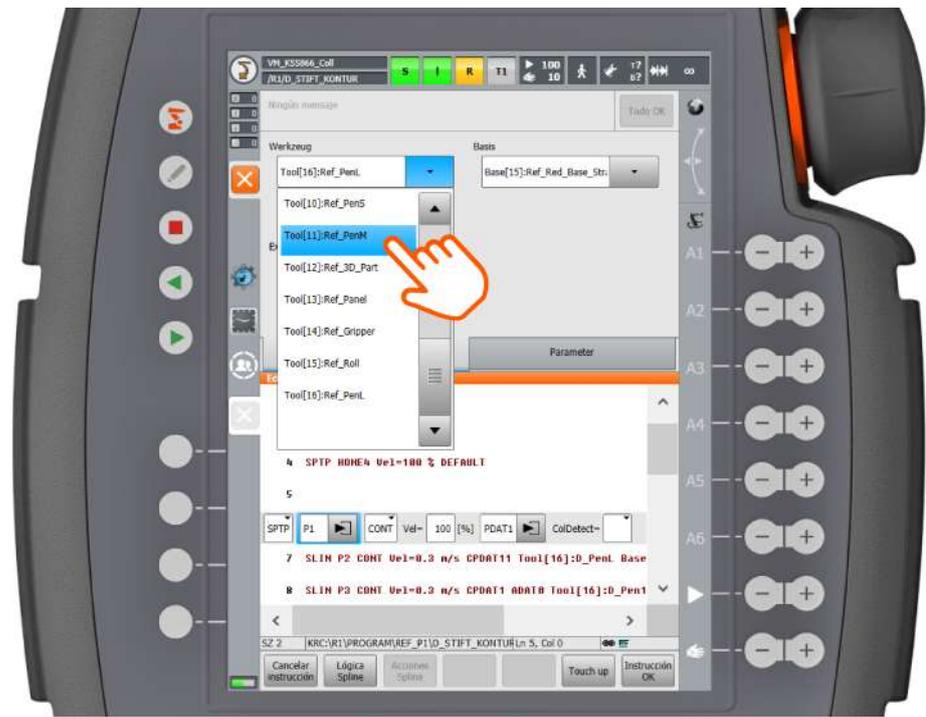
Efectos al modificar los datos del Frame

Si los datos del Frame (p. ej. Tool, Base) se modifican, se produce un desplazamiento de la posición (ver "Datos de desplazamiento"). Cambia la posición del robot (corresponde a la posición angular del eje del robot). Las coordenadas antiguas del punto siguen guardadas y válidas. Solo cambia la referencia (p. ej., 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 del Frame, se deben actualizar las coordenadas con "TouchUp" después de modificar los parámetros (p. ej., base) en la posición deseada.



Además, aparece un aviso: "Atención, al cambiar los parámetros del Frame referentes a los puntos, existe peligro de colisión."

Procedimiento



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Modificar movimientos programados > Modificación de los datos del Frame

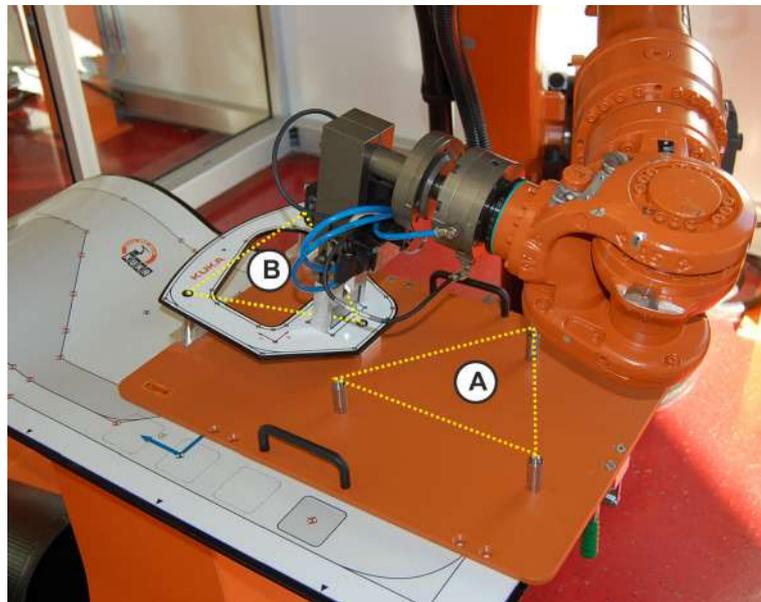
15.6.1 Ejercicio (variante 2): Programar movimientos con TCP externo

¡Ahora es su turno!



Enunciado 1

Los dos módulos de programa para recoger y depositar el componente 3D se han creado y se han probado.



Enunciado 2

1. Compruebe el ajuste del robot.
2. Programe un offset para la pieza de trabajo del componente 3D con garra y guarde el offset de ajuste.
3. Mida la boquilla de aplicación de pegamento externa como base n.º ____ y la placa como herramienta n.º ____ .

Enunciado 3

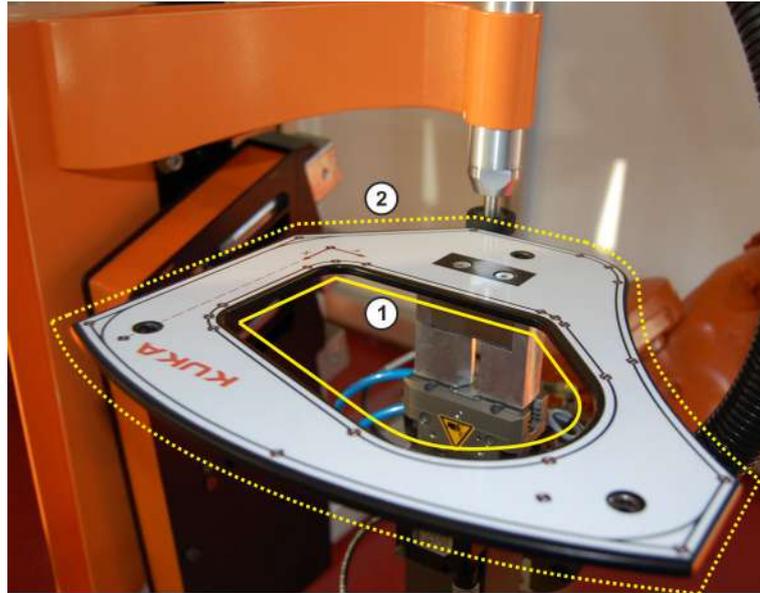
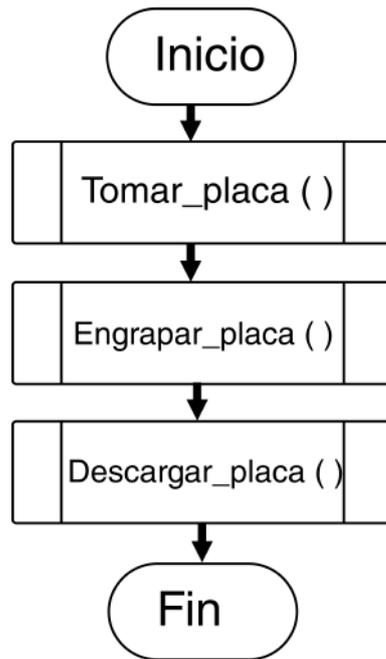


Fig. 15-7: Contornos de engrapado en el componente 3D

1. Sujete manualmente el componente 3D en la garra.
2. Programe un contorno indicado sobre la placa con el **programa:** _____ .
 - Utilizar para ello la herramienta externa medida y la placa.
 - **ext. Herramienta**
N.º _____ nombre: _____
 - **(Brida) básica**
N.º _____ nombre: _____
 - 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*.
 - Utilice un bloque spline para la programación de movimiento.
3. Pruebe su programa en los modos de servicio T1, T2 y AUT.

Enunciado 4

1. Cree un nuevo módulo de programa con el nombre **main()** .
2. Abra el programa main() y vincule correctamente los programas creados.

```
DEF main()  
INI  
  
schild_holen()  
schild_falzen()  
schild_ablegen()  
  
end
```

3. Cierre su programa main() y selecciónelo para probarlo.
4. Pruebe el programa main() en los diferentes modos de servicio.

15.7 Ejercicio (variante 1): Medición BASE "Contorno de trayectoria"

¡Ahora es su turno!



Enunciado

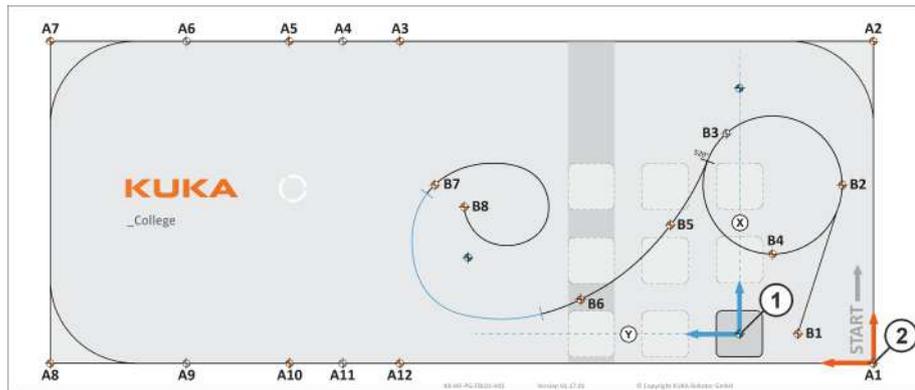


Fig. 15-8: Medición de BASE sobre la mesa

1. Medir la base roja n.º _____ nombre: _____ sobre la mesa con el método de 3 puntos.
 - Emplear la herramienta ya medida n.º _____ nombre: _____ como herramienta de medición.
2. Guardar los datos de la base medida.
3. Activar la herramienta n.º _____ nombre: _____ y la base n.º _____ nombre: _____.
 - Desplazar la herramienta al origen de la base y mostrar la posición cartesiana real.

X Y Z A B C

Configuración de una nueva estación de robot: Unidad de calibración para BASE

15.8 Ejercicio (variante 2): Medición base "Cuadrícula de cubos"

Enunciado



Fig. 15-9: Cuadrícula de cubos, medición de base

1. Colocar la cuadrícula de cubos libre sobre su mesa de formación. Tener en cuenta que el lugar de almacenamiento puede restaurarse en cualquier momento.
2. Medir la base roja *n.º* _____ *nombre:* _____ sobre la cuadrícula de cubos con el método de 3 puntos.

Utilizar reducciones a lo largo de los bordes de la cuadrícula del cubo para la medición de la base.

- Emplear la herramienta ya medida *n.º* _____ *nombre:* _____ como herramienta de medición.
3. Guardar los datos de la base medida.
 4. Activar la herramienta *n.º* _____ *nombre:* _____ y la base *n.º* _____ *nombre:* _____.
- Desplazar la herramienta al origen de la base y mostrar la posición cartesiana real.

X	Y	Z	A	B	C
.....

15.9 Ejercicio (variante 3): Medición base "Componente de soldadura"

Enunciado

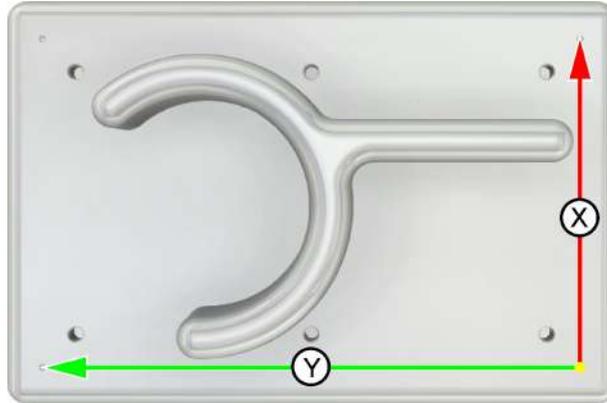


Fig. 15-10: Medición de base

Medición de base en el componente de soldadura

- Colocar el componente de soldadura libre sobre su mesa de formación. Tener en cuenta que el lugar de almacenamiento puede restaurarse en cualquier momento.



Retirar previamente la lámina magnética de la mesa.

- Medir la base roja *n.º* _____ *nombre:* _____ sobre el componente de soldadura con el método de 3 puntos.
 - Emplear la herramienta ya medida *n.º* _____ *nombre:* _____ como herramienta de medición.
- Guardar los datos de la base medida.
- Activar la herramienta *n.º* _____ *nombre:* _____ y la base *n.º* _____ *nombre:* _____.
 - Desplazar la herramienta al origen de la base y mostrar la posición cartesiana real.

X Y Z A B C

.....

15.10 Preguntas: Medir BASE

Lo que se debe saber tras el ejercicio:



¿Qué ventajas ofrece la mediación de una BASE?



Fig. 15-11: Campo de respuesta

¿Cómo se puede activar una BASE para el desplazamiento manual?

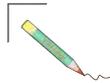


Fig. 15-12: Campo de respuesta

¿Cómo se puede establecer la BASE de referencia para un punto de movimiento programado?

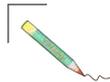


Fig. 15-13: Campo de respuesta

16 Configuración de una nueva estación de robot: Creación de programas

16.1 Navegador del curso

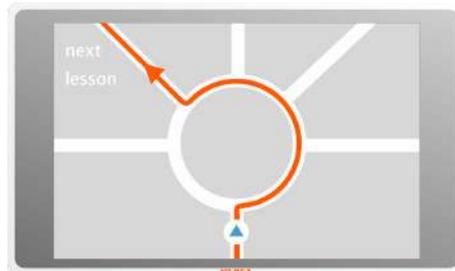
¿Qué requisitos aporta usted?



¡Ya ha conseguido acumular mucha práctica!

- Está capacitado para manejar y desplazar robots.
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- SLIN, SCIRC y SPTP ya no son desconocidos.
- Está capacitado para corregir y modificar los movimientos del robot existentes.

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Creación de programas

- Para crear su primer programa completamente creado, aún conocerá los últimos detalles en este apartado de aprendizaje.
- ¿Por qué hay que tener en cuenta Status y Turn en un desarrollo del movimiento SPTP?
- ¿Qué son las singularidades y cómo afectan a SLIN y SCIRC?
- ¿Cómo se aproximan los movimientos?

16.1.1 Status y Turn

Efecto Status y Turn

Status y Turn sirven para determinar una posición de eje unívoca a partir de varias posiciones de eje posibles para una misma posición del TCP. Diferentes posiciones de eje debidos a diferentes valores de Status y Turn:



Evaluación de Status y Turn

La unidad de control del robot considera los valores de Status y Turn programados solo en caso de movimientos SPTP. Se ignoran en los movimientos de trayectoria (SLIN, SCIRC) y en los bloques SPLINE (SPL). La primera instrucción de movimiento de un programa es SPTP (HOME).

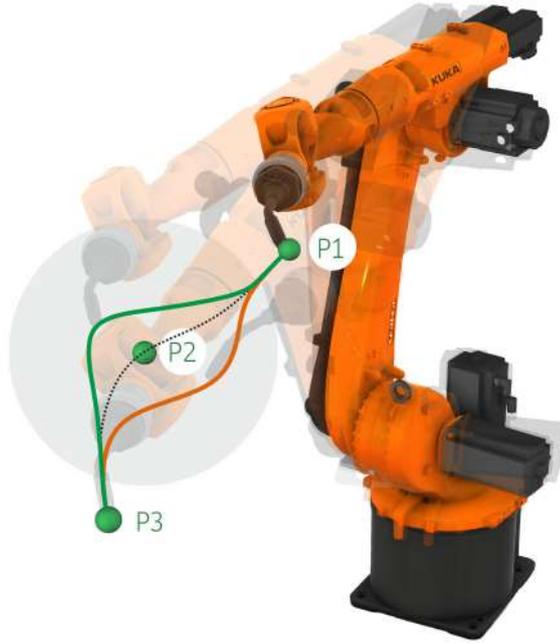
- Las posiciones HOME son del tipo E6AXIS (config.dat) y definen una posición de salida única.
- Otros tipos posibles para SPTP son AXIS, E6POS y POS

```
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
```

16.1.2 SPTP – Aproximación

Descripción

La unidad de control es capaz de aproximar las instrucciones de un desarrollo del movimiento marcadas con CONT para la aceleración del movimiento. Aproximar significa que no se desplaza exactamente a las coordenadas de punto. La trayectoria se abandona antes. El TCP se guía a lo largo de un contorno de posicionamiento aproximado que desemboca en la trayectoria del siguiente comando de movimiento.

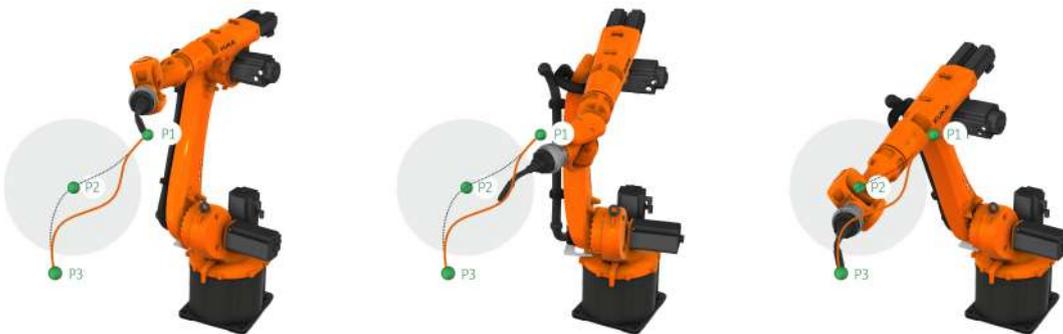


Los cursos de trayectoria de ejemplo están muy sobredimensionados para una mejor representabilidad y no reflejan necesariamente el curso de trayectoria.

Posible trayectoria de aproximación A



Posible trayectoria de aproximación B



Ventajas del posicionamiento aproximado

El robot ya no necesita desacelerar y acelerar entre los puntos (1). Como resultado, el programa se procesa más rápidamente (2), lo que da lugar a un tiempo de ciclo optimizado y a una alta eficiencia energética.

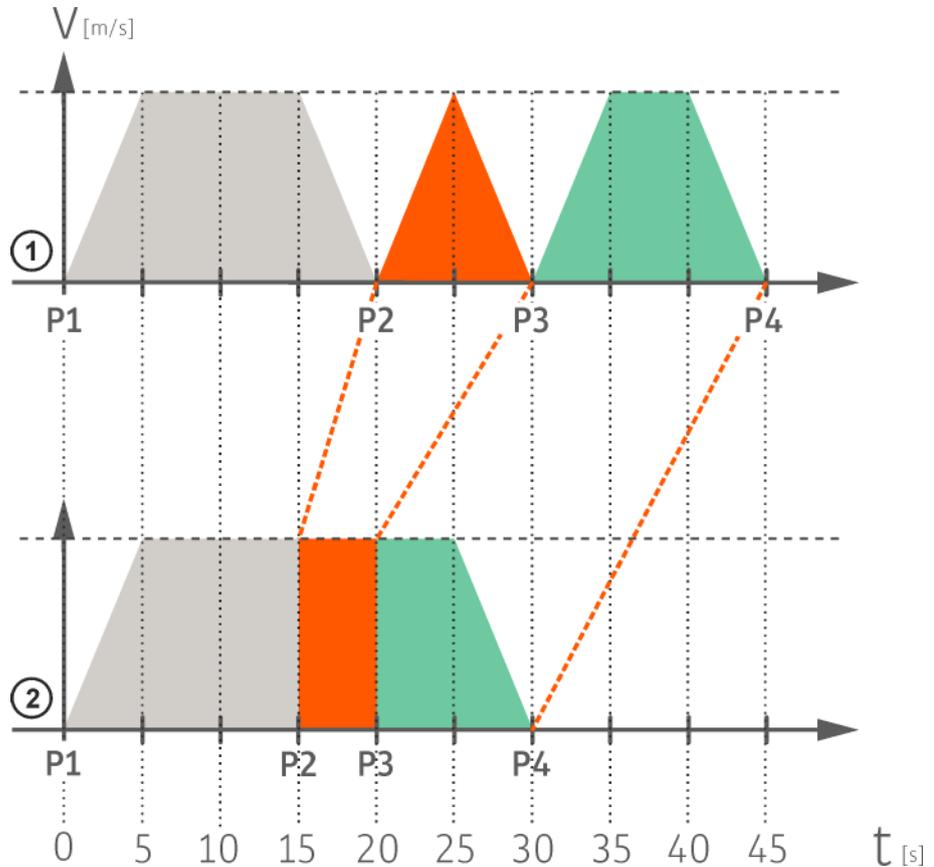


Fig. 16-1: Parada exacta – Aproximación en comparación

SPTP – Comportamiento de trayectoria para la aproximación

Para poder ejecutar el movimiento de aproximación la unidad de control debe poder leer los siguientes conjuntos de movimientos. Ello se realiza con el procesamiento en avance. La información sobre la aproximación se indica en mm.

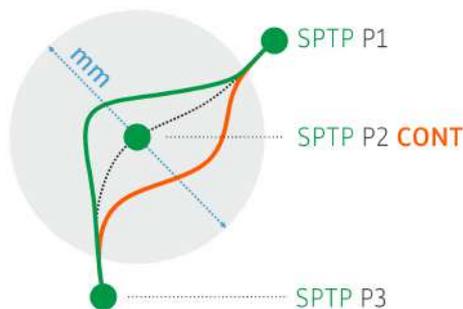


Fig. 16-2: SPTP – Comportamiento de trayectoria para la aproximación



El contorno de aproximación no es predecible.

16.2 Programar un movimiento de trayectoria SLIN

1. Mover el TCP a la posición que se programará por aprendizaje como punto de destino.

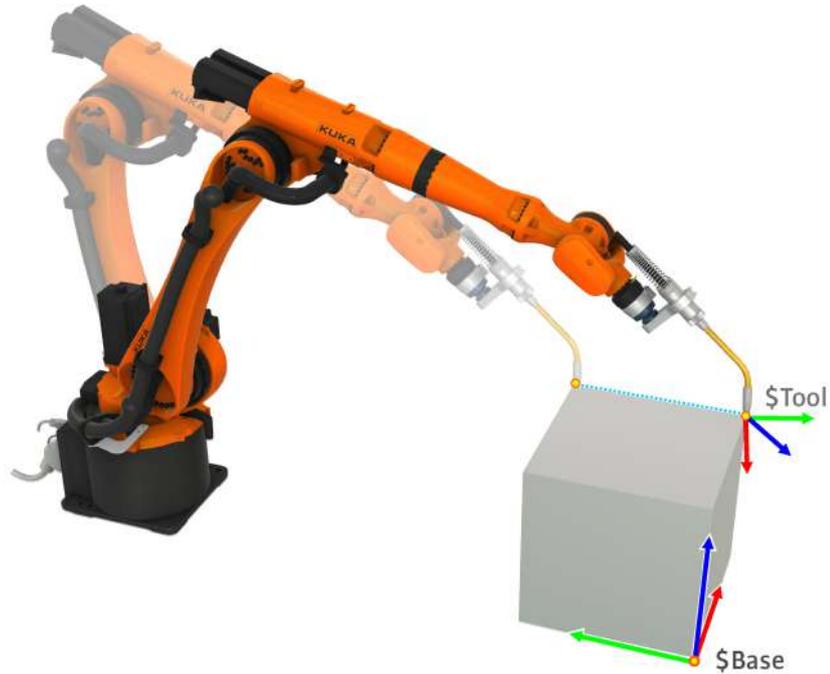


Fig. 16-3: Instrucción de movimiento con SLIN

2. Insertar en el formulario inline SLIN.

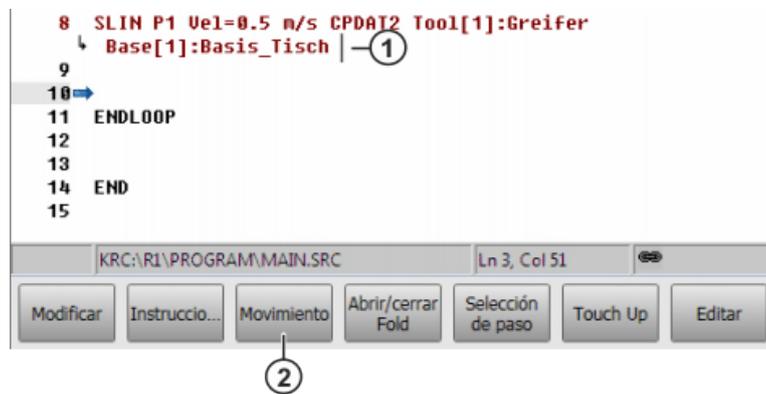


Fig. 16-4: Insertar movimiento

- Colocar el cursor en la línea (1) detrás de la cual se debe insertar la instrucción de movimiento.
 - Insertar un **movimiento** (2) usando el botón con el mismo nombre.
3. Se inserta un formulario inline previamente completado tras la línea de programa marcada.

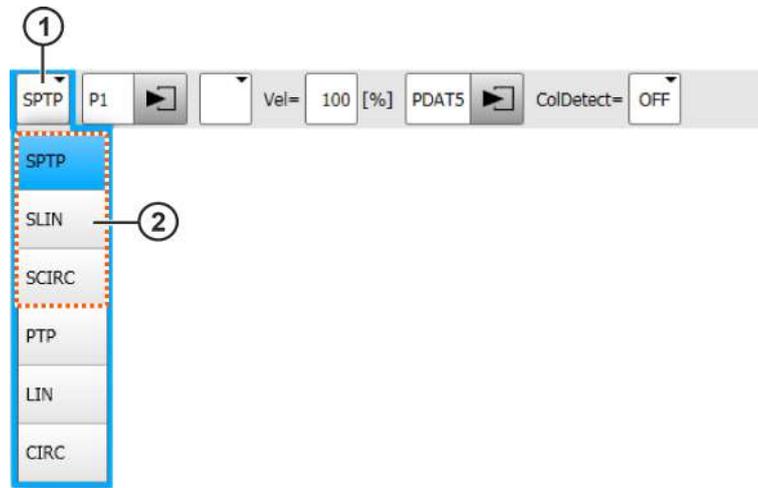


Fig. 16-5: Selección del tipo de movimiento

- En el formulario inline, pulsar el campo con la trayectoria del movimiento (1).
 - Se abre una selección; seleccionar aquí la trayectoria del movimiento SLIN (2).
4. Configurar los diferentes parámetros de movimiento.
 - **SLIN**
 5. Guardar la instrucción pulsando **Instrucción OK**. La posición opuesta del TCP se programa como punto de destino.

16.2.1 Singularidades

Descripción

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) unívoca 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 la zona de los movimientos de trayectoria, pero no en los movimientos de eje.

Singularidad por encima de la cabeza

- 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 unívocamente mediante la transformación de retroceso y puede por tanto aceptar cualquier valor.

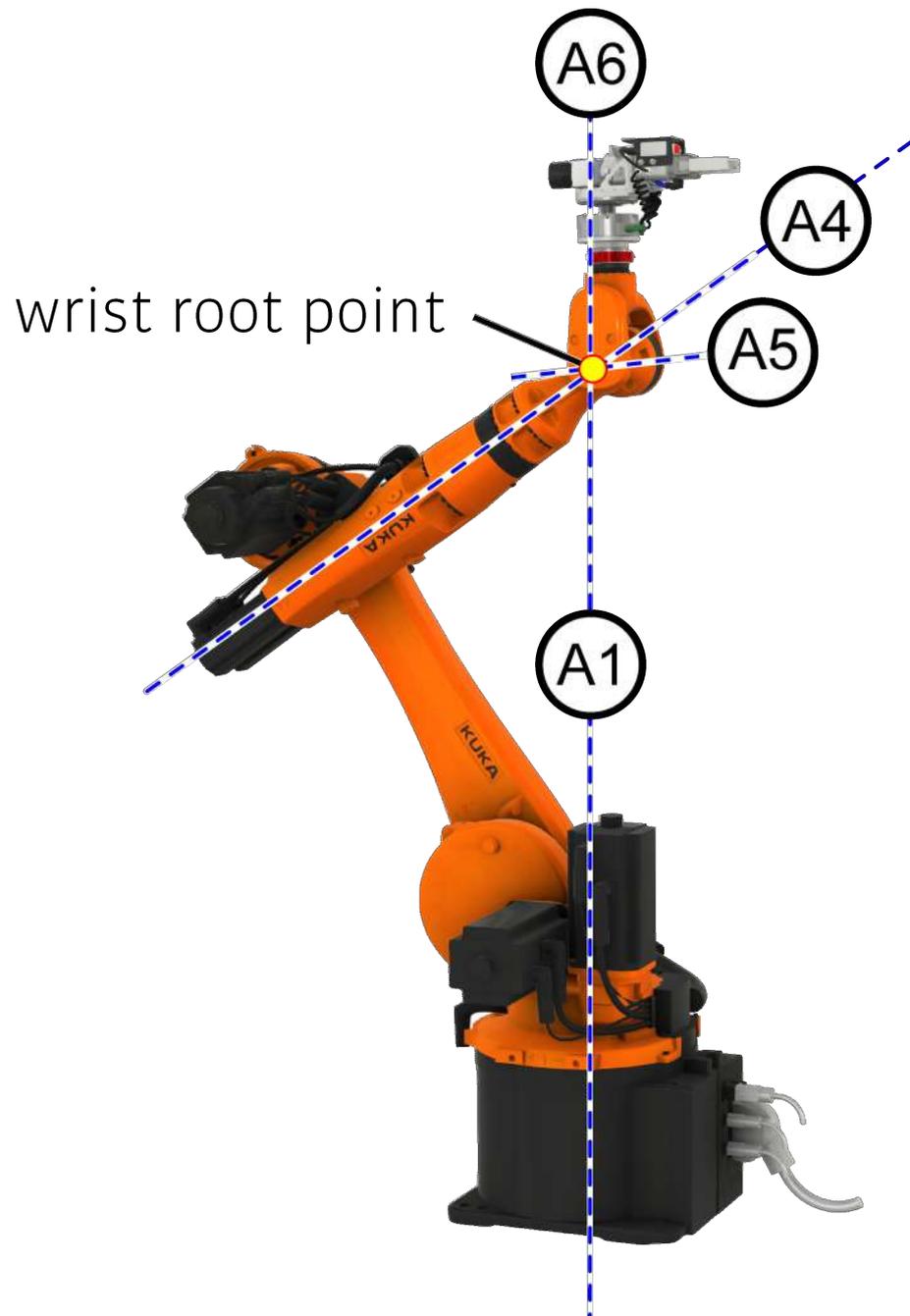


Fig. 16-6: Singularidad por encima de la cabeza

Singularidad de las posiciones extendidas

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 A2 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.

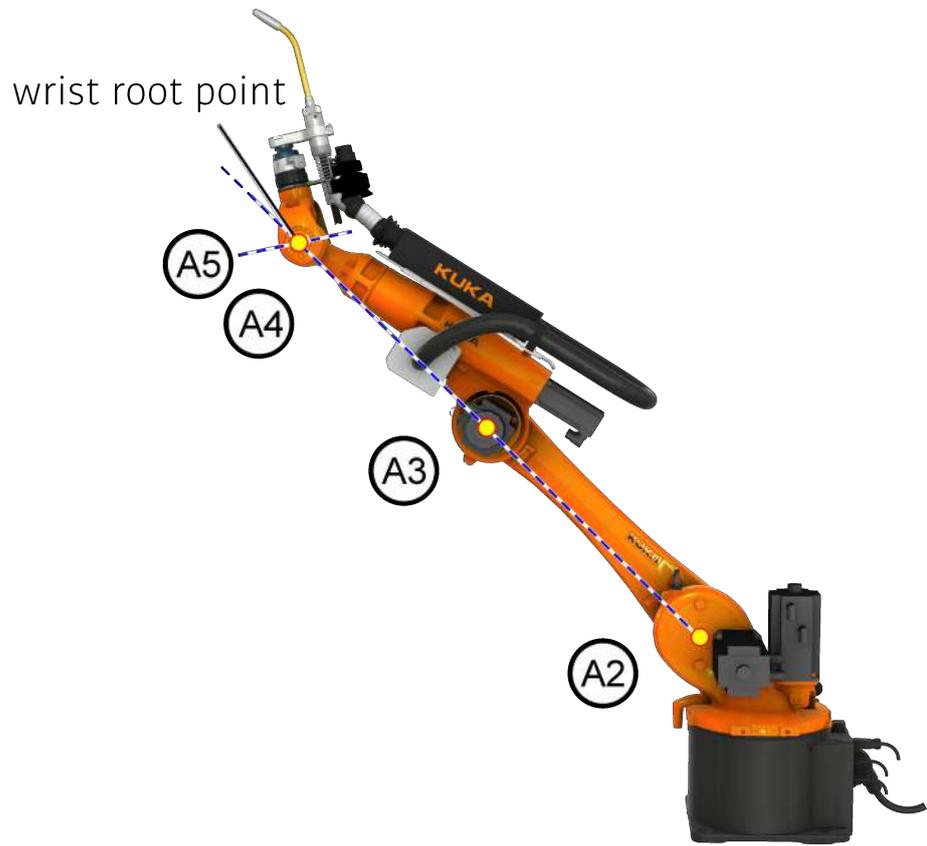


Fig. 16-7: Posición extendida

Singularidades de los ejes de la muñeca

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.

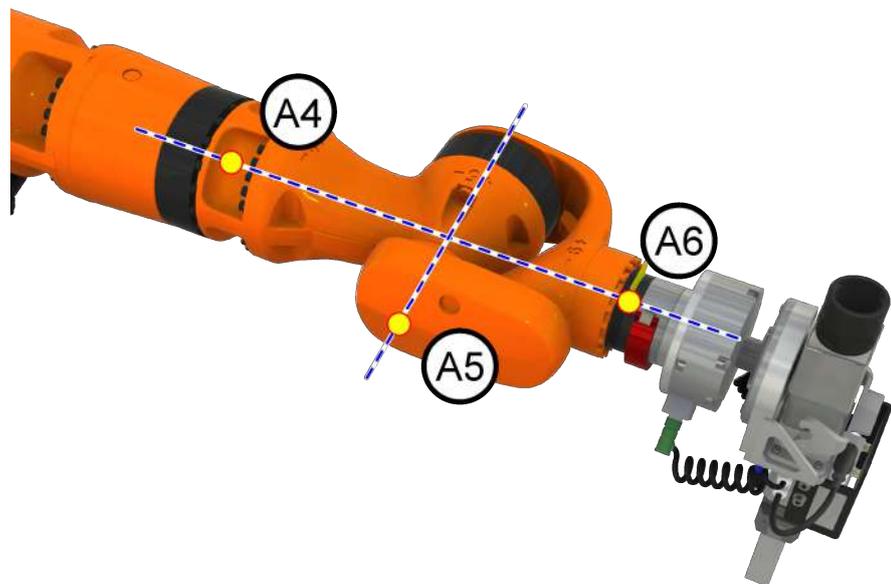


Fig. 16-8: Singularidades de los ejes de la muñeca

16.2.2 Conocer los controles de orientación SLIN

PTP estándar o manual

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

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

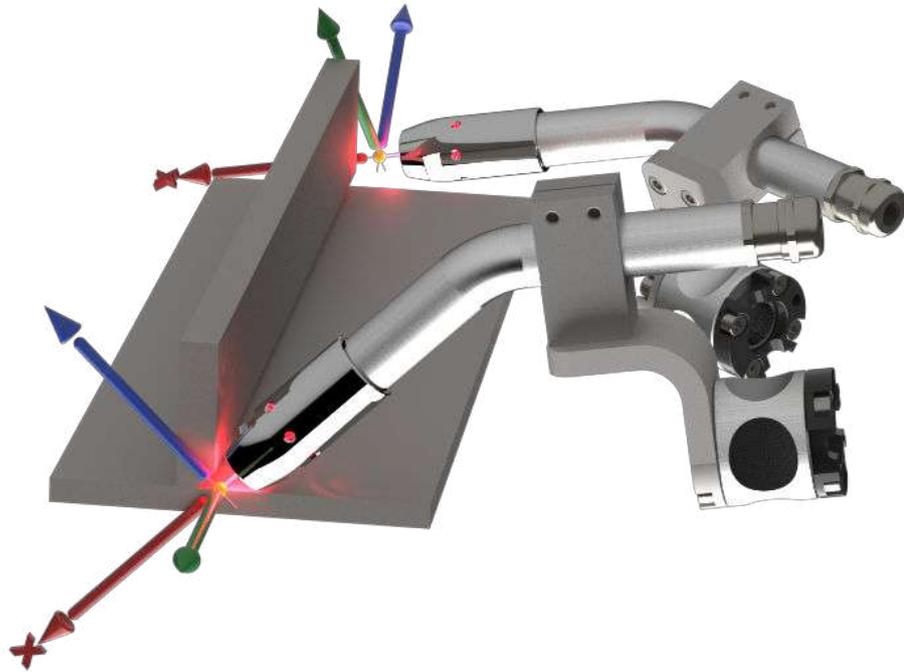


Fig. 16-9: Control de la orientación: Estándar



ATENCIÓN

Las singularidades pueden provocar una reorientación rápida y pronunciada. Especialmente en el caso de las herramientas de gran tamaño, las grandes desviaciones pueden provocar daños materiales o incluso personales.

Constante

La orientación programada por aprendizaje en el punto de inicio de la herramienta se mantiene de forma constante durante el movimiento. La orientación programada en el punto de destino se ignora (orientación de la herramienta naranja en la imagen).

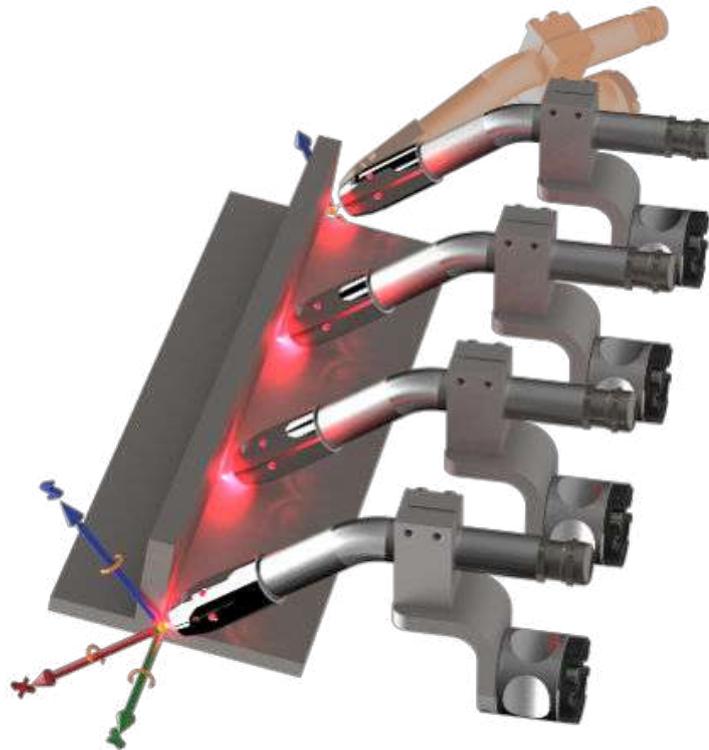


Fig. 16-10: Control de la orientación: Constante

16.3 Programar un movimiento de trayectoria SCIRC

Aquí se mueve el punto de referencia de la herramienta 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.

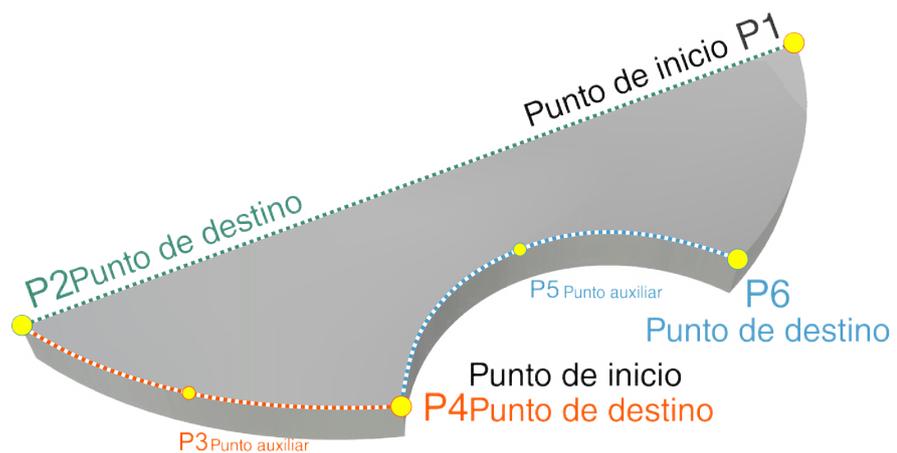


Fig. 16-11: Dos segmentos circulares con SCIRC

16.3.1 Programar SCIRC

Procedimiento

1. Desplazar el TCP a la posición que se debe procesar como **HP** (punto auxiliar) y **ZP** (punto de destino).

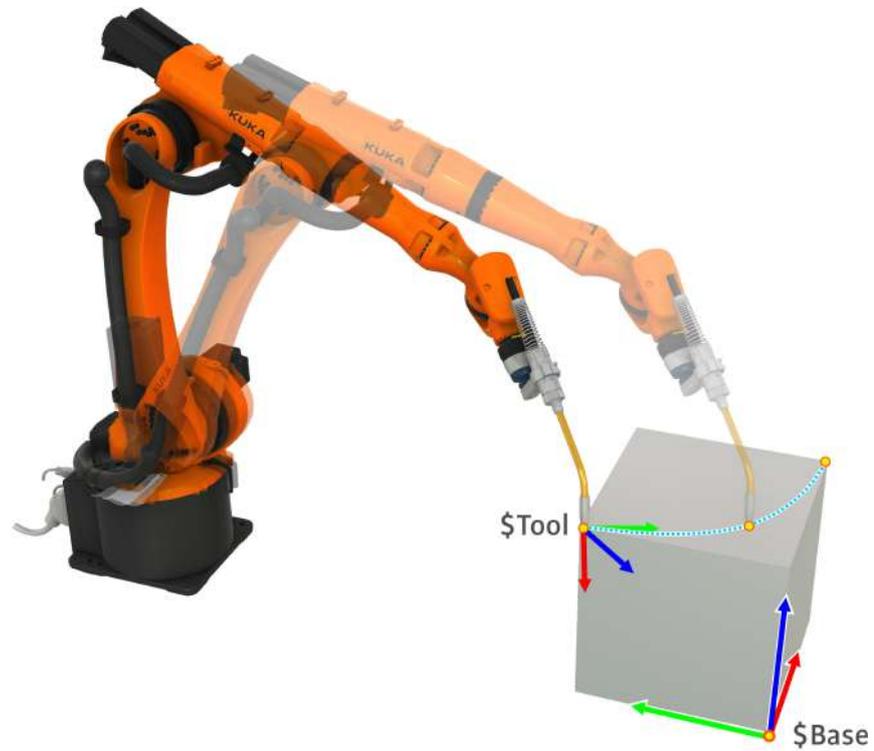


Fig. 16-12: Instrucción de movimiento con SCIRC

2. Insertar en el formulario inline **SCIRC**.

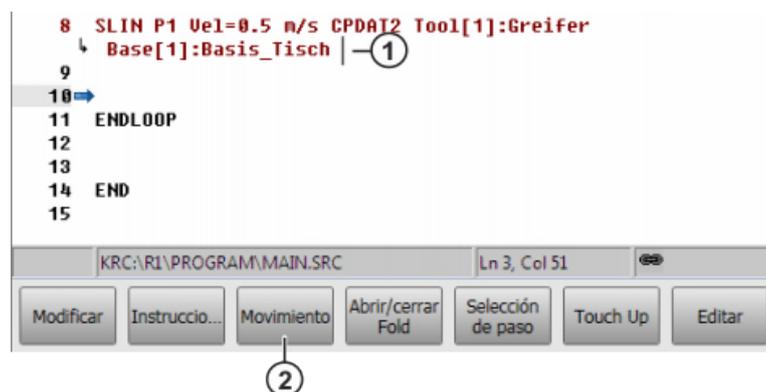


Fig. 16-13: Insertar movimiento

- Colocar el cursor en la línea (1) detrás de la cual se debe insertar la instrucción de movimiento.
 - Insertar un **movimiento** (2) usando el botón con el mismo nombre.
3. Se inserta un formulario inline previamente completado tras la línea de programa marcada.

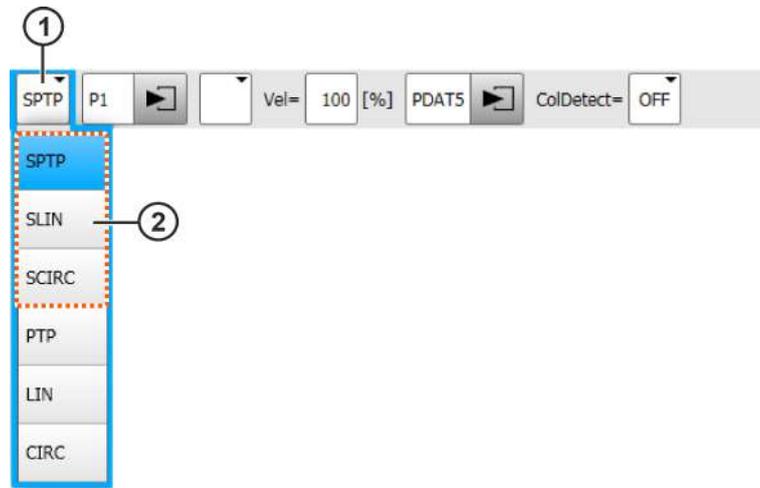


Fig. 16-14: Selección del tipo de movimiento

- En el formulario inline, pulsar el campo Tipo de movimiento (1).
 - Se abre una selección; seleccionar el tipo de movimiento SCIRC (2).
4. Configurar los diferentes parámetros de movimiento.
 - **SCIRC**
 5. Programar el punto auxiliar y de destino.

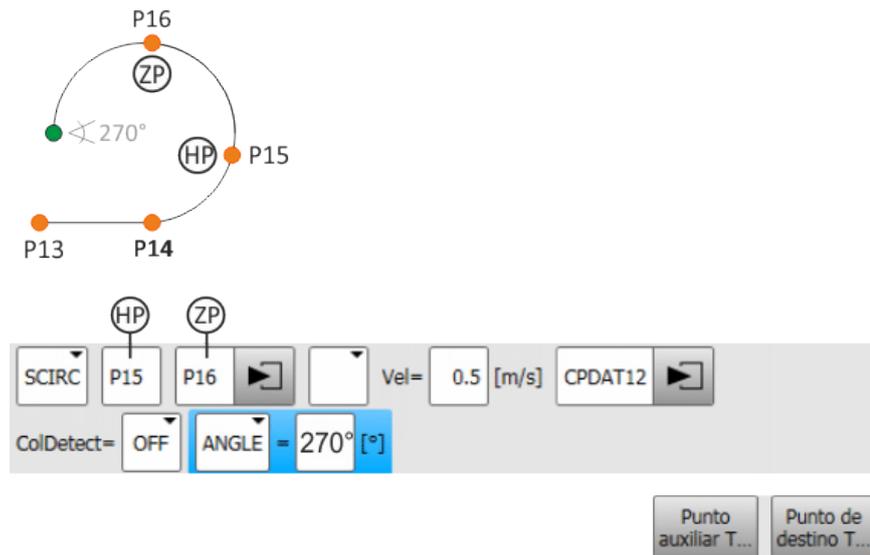


Fig. 16-15: Programar por aprendizaje los puntos SCIRC

- Aproximar el punto auxiliar en la trayectoria circular (HP) y aceptarlo usando el botón **Punto auxiliar Touch-Up**.
- Aproximar el/un punto de destino en la trayectoria circular (ZP) y aceptarlo usando el botón **Punto de destino Touch-Up**.



De manera opcional, mediante el parámetro adicional ANGLE (campo azul) se puede aumentar o reducir la trayectoria circular usando el punto de destino. Para ello es necesario introducir el ángulo.

AVISO

Al modificar el ángulo circular se debe comprobar el siguiente curso de trayectoria » Peligro de colisión.

6. Guardar la instrucción pulsando **Instrucción OK**.

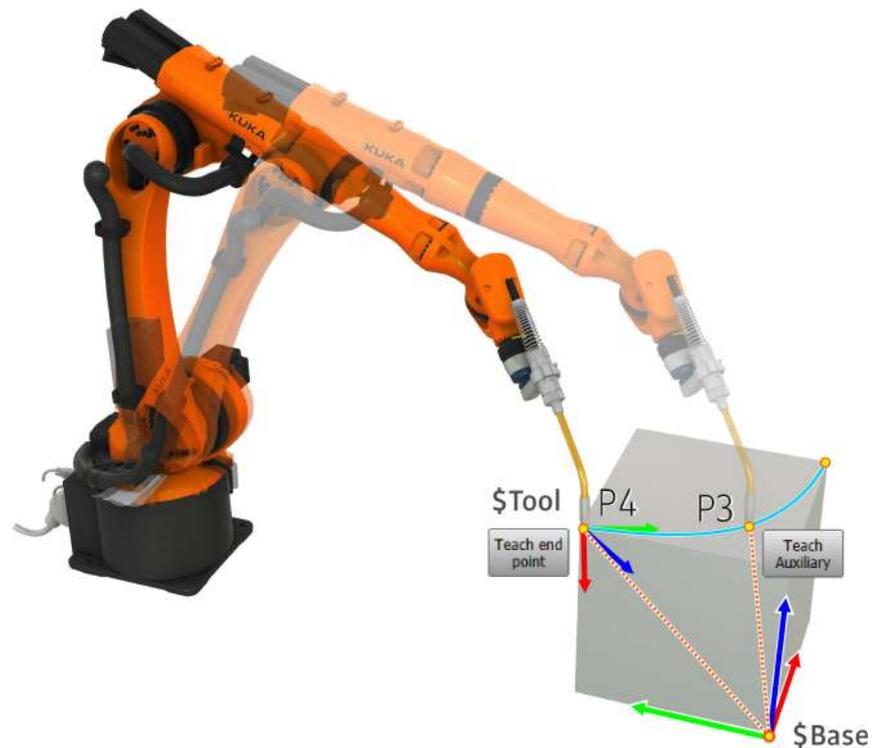


Fig. 16-16: Almacenamiento de las coordenadas del punto en "Touch Up HP" y "Touch Up ZP"

AVISO

Cerrar y guardar el formulario inline con el botón "Comando OK" después de que se hayan registrado los puntos auxiliares y de destino.

16.3.2 Conocer el control de orientación SCIRC

Control de la orientación: Estándar o PTP manual

La orientación de la herramienta se modifica de forma continua durante el movimiento. PTP manual se utiliza cuando exista la posibilidad de que el robot con Estándar pueda entrar en una singularidad de los ejes de la muñeca. Ya que la orientación se logra mediante la transferencia lineal (desplazamiento específico del eje) del ángulo del eje de la muñeca .

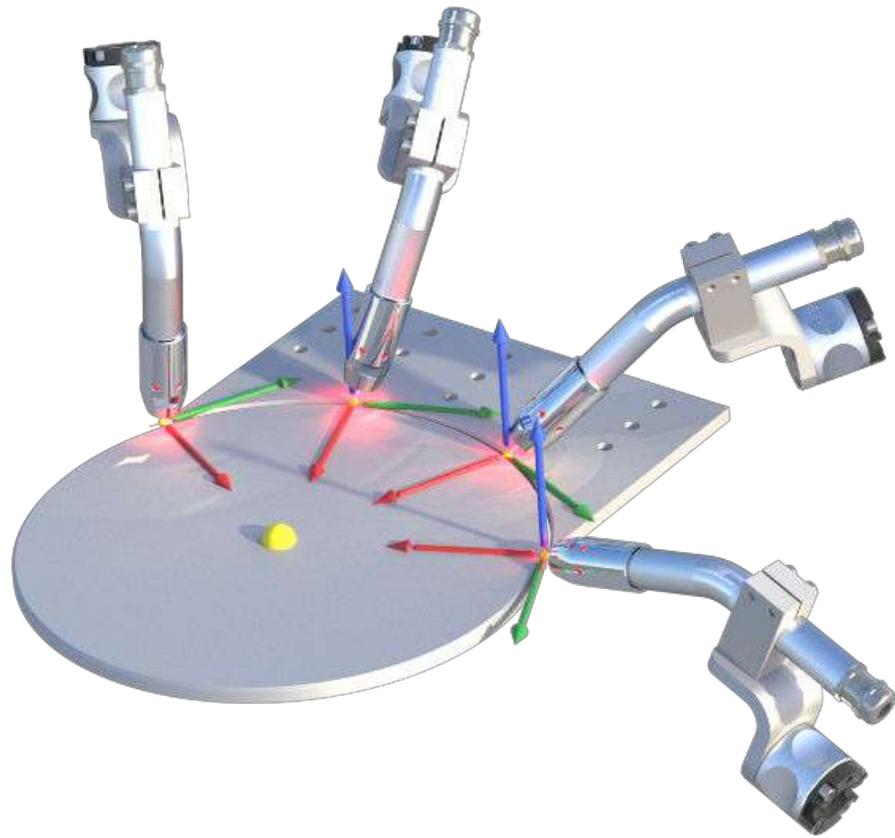


Fig. 16-17: Orientación: Relación base y estándar

Control de la orientación: Relación base y constante

La orientación de la herramienta se mantiene de forma constante durante el desarrollo del movimiento, tal y como ha sido programada en el punto de inicio. La orientación programada en el punto de destino se ignora.

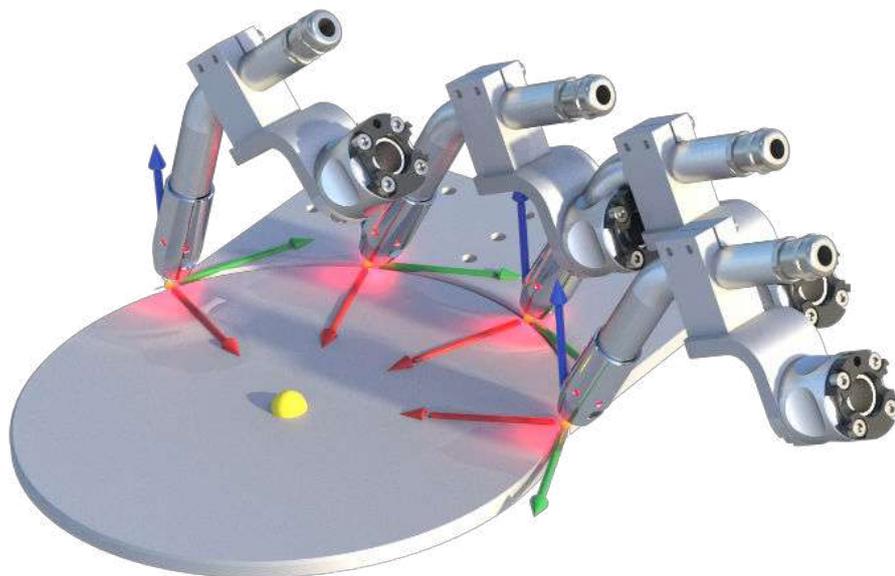


Fig. 16-18: Orientación: Relación base y constante

Control de la orientación: Relación trayectoria y constante

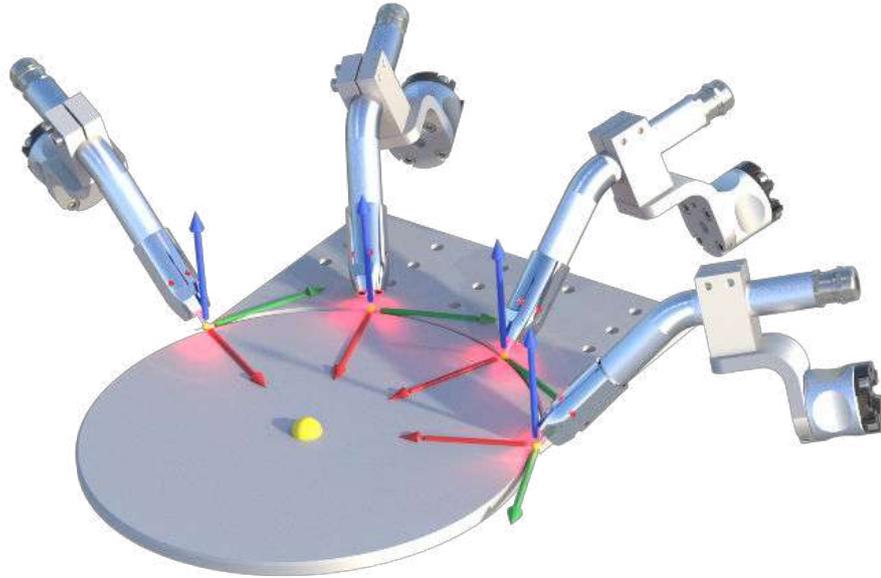


Fig. 16-19: Orientación: Relación trayectoria y constante

Podrá encontrar más información en la documentación del cliente.



Libro	KUKA System Software 8.6 Instrucciones de servicio y programación para los integradores de sistemas
Filtro de búsqueda	Documentación > Instrucciones de manejo y programación
Capítulo/apartado	Principios de la programación de movimiento > Control de orientación CP-Spline

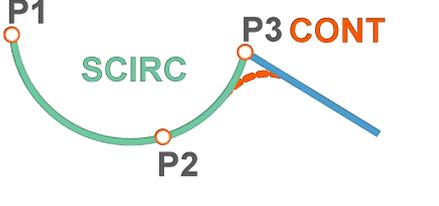
16.4 Aproximación en los movimientos de trayectoria

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 parada exacta en el punto.

Aproximación en los tipos de desplazamiento SLIN y SCIRC

Tipo de movimiento	Característica
 <p>P1 SLIN P2 CONT</p>	<ul style="list-style-type: none"> • El curso de la trayectoria corresponde a dos parábolas. • Indicación de la distancia de aproximación en mm
 <p>P1 SCIRC P2 P3 CONT</p>	<ul style="list-style-type: none"> • El curso de la trayectoria corresponde a dos parábolas. • El punto de destino es de posicionamiento aproximado • Indicación de la distancia de aproximación en mm

16.5 Ejercicio (variante 1): Creación de programas con movimientos individuales

Enunciado

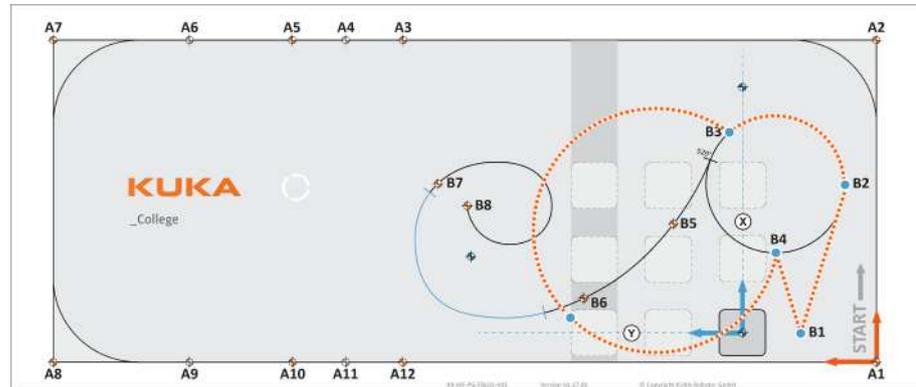


Fig. 16-20: Movimiento de trayectoria

1. Recorra con la herramienta N.º _____ nombre: _____ y la base N.º _____ Nombre: _____ el contorno marcado en naranja.
2. Realice la aproximación al punto de inicio B1 con una posición intermedia.
3. El contorno termina de nuevo en el punto B1.
4. Utilice las trayectorias del movimiento SPTP, SLIN, SCIRC.
5. Pruebe su programa en diferentes modos de servicio.

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.

Lo que se debe saber tras el ejercicio:



¿Cuáles son las características de los movimientos SLIN y SCIRC?



Fig. 16-23: Campo de respuesta

¿Cómo se indica la velocidad de desplazamiento en los movimientos SPTP, SLIN y SCIRC y a qué hace referencia esta velocidad?

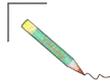


Fig. 16-24: Campo de respuesta

¿Cómo se indica la distancia de aproximación en los movimientos SPTP-, SLIN y SCIRC y a qué hace referencia esta velocidad?

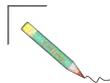


Fig. 16-25: Campo de respuesta

¿Qué se debe tener en cuenta al añadir instrucciones CONT en comandos de movimiento existentes?

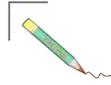


Fig. 16-26: Campo de respuesta

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



Fig. 16-27: Campo de respuesta

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



Fig. 16-28: Campo de respuesta

17 Configuración de una nueva estación de robot: Programación de un movimiento SPLINE

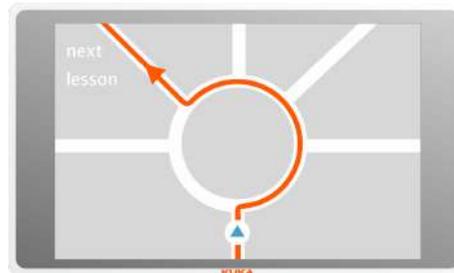
17.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para crear programas propios con formularios inline y las trayectorias de los movimientos SPTP, SLIN y SCIRC.

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Programación de un movimiento SPLINE

- Además de los movimientos de la trayectoria SLIN y SCIRC conocidos, conocerá SPLINE. Conocerá estos contornos posiblemente de programas de caracteres con el nombre "Línea de manos libres".
- Agrupará varios movimientos con SLIN, SCIRC y SPL (SPLINE) con un bloque SPLINE y aprovechará sus ventajas.

17.2 Descripción de la situación

En las carcasas para el sistema electrónico de control se aplica una costura estanca antes de cerrarlas.

El contorno de la junta no se compone solo de líneas y curvas, sino también de una "curva de forma libre".

Una "curva de forma libre" puede realizarse en el robot con la trayectoria del movimiento "SPLINE".



17.3 ¿Cómo se recorren los puntos programados por el robot?

Programación mediante movimientos individuales

- Un programa de robot está compuesto, en su primer paso, por puntos individuales programados por aprendizaje o calculados.

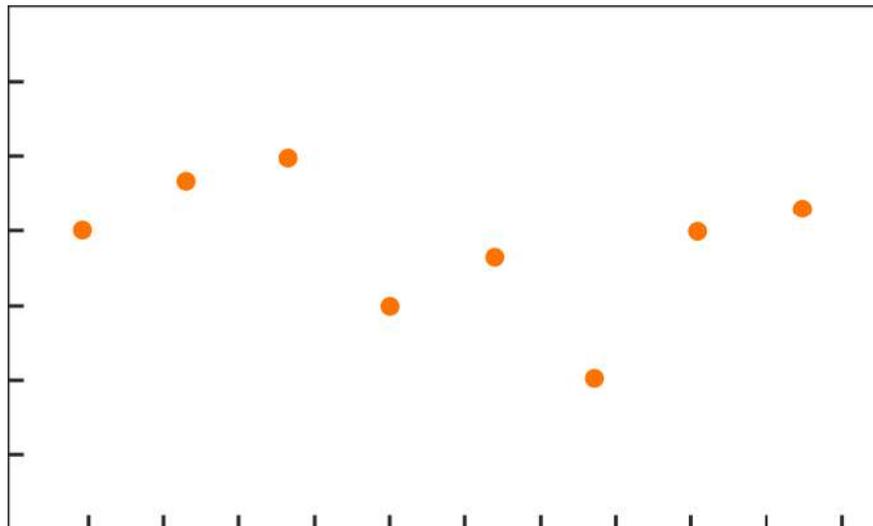


Fig. 17-1: Posiciones de situación de partida

- A través de un formulario inline o la programación KRL se comunica adicionalmente al robot cómo deben interconectarse estos puntos.
- **Ejemplo:** En este caso, el posible conjunto de movimientos es (S)LIN. » Los puntos se interconectan con un trayecto.

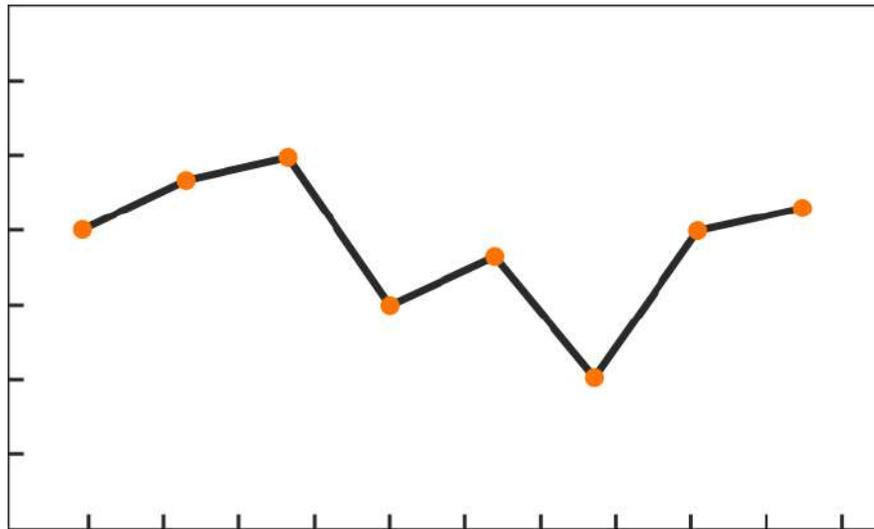


Fig. 17-2: Interpolación lineal



Inconveniente: El robot se para en cada punto individual y vuelve a acelerar.

- Para evitar la parada exacta, los puntos pueden aproximarse con el parámetro CONT.

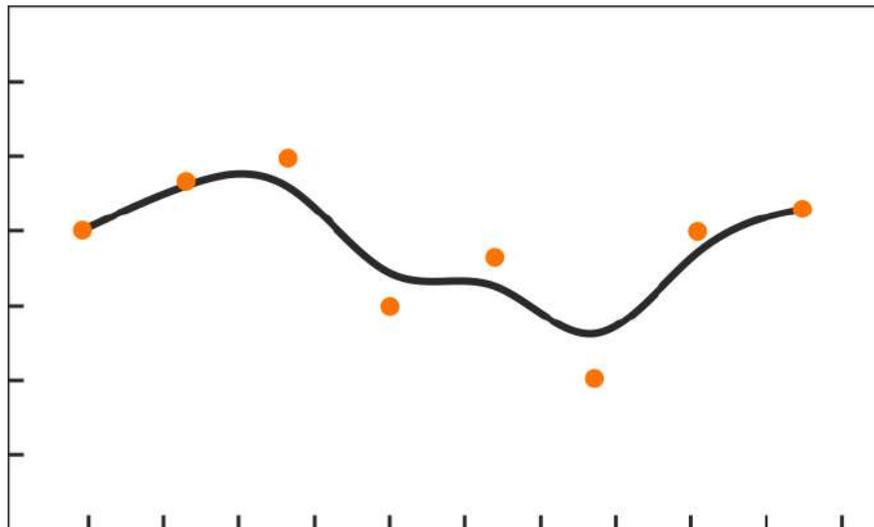


Fig. 17-3: Interpolación B-SPLINE



Inconveniente: El robot aproxima los puntos programados por aprendizaje/calculados. Solo los puntos de inicio y final están en la trayectoria.

Ventaja: El robot recorre toda la trayectoria con un movimiento fluido.



Para ser exactos, se trata ya de una forma especial de SPLINE.

17.3.1 ¿Qué es SPLINE?

Descripción

- Flexicurva

- La denominación "Spline" se refiere a la flexión de tabloncillos de madera de la "hilada de planchas" con la que anteriormente se determinaban los contornos lisos del armazón en la ingeniería naval y aeronáutica.

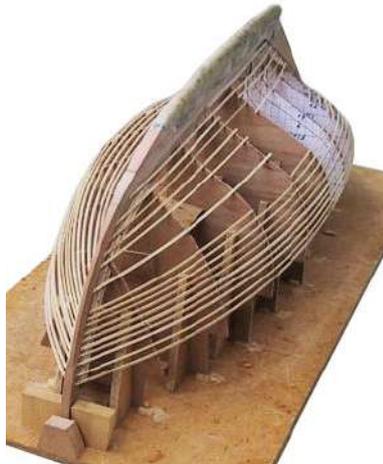


Fig. 17-4: Casco, construcción de tabloncillos

- Con este método, se obliga a que la flexicurva (curva elástica; inglés: spline) cumpla las condiciones de interpolación (contorno) mediante el uso de pesas de hilada colocadas en los puntos de interpolación (en la imagen, clavos/machos).

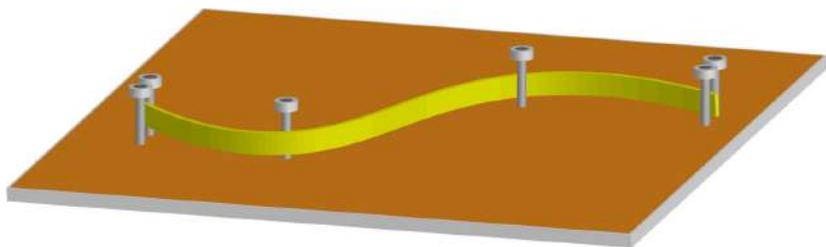


Fig. 17-5: Flexicurva

- Debido a la elasticidad del material se genera una línea elástica de curvatura total mínima.

- **Punto de apoyo desplazado**

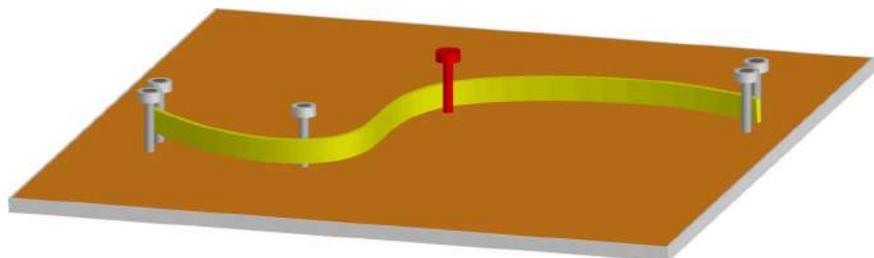


Fig. 17-6: Desplazamiento de un punto de apoyo

- El desplazamiento de un punto de apoyo (clavo/macho rojo) afecta a la trayectoria global de la flexicurva.
- La trayectoria puede precisarse con varios puntos de apoyo.

17.3.2 SPLINE – Conocer los segundos planos matemáticos

Descripción

- Interpolación polinómica

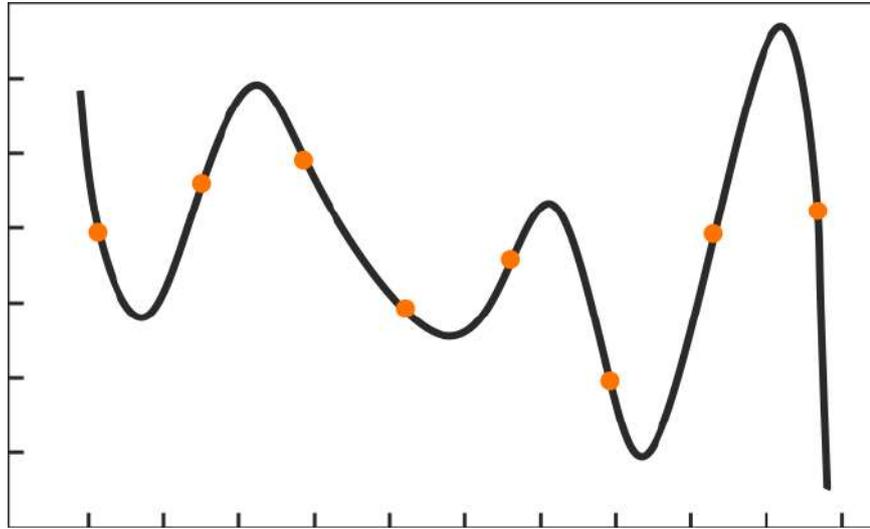


Fig. 17-7: Interpolación polinómica

- Interpolación polinómica: Función de grado n por los puntos
- Si se utiliza, por ejemplo, una función polinómica de grado superior a la función de interpolación (ver imagen), pueden producirse **oscilaciones muy extremas** de la curva.
- De 8 posiciones indicadas se obtiene un polinomio de grado 7 ($ax^7+bx^6+\dots$)



Sin embargo, en aplicaciones de construcción, como la ingeniería naval o la industria aeronáutica, este tipo de oscilaciones tan fuertes son muy indeseables. Aquí solo se necesitan curvas ligeramente arqueadas y adaptables entre los distintos puntos de apoyo (imagine un casco en formato de diente de sierra...).

- Interpolación SPLINE

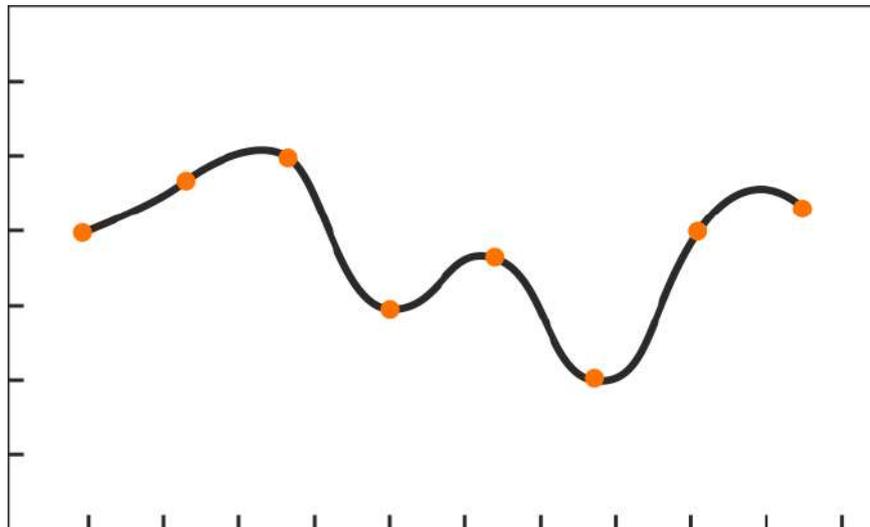


Fig. 17-8: Interpolación SPLINE

- Los Splines no intentan interpolar una función en todos los puntos de apoyo, si no que tratan de hacerlo localmente a través de dos puntos de apoyo respectivamente.
- Estos Splines tienen un efecto alisador y solamente tienen una ondulación muy reducida. De este modo se pueden representar contornos de manera fiel a la realidad.
- En este caso se usan con mayor frecuencia Splines lineales o cúbicos interpolados mediante una función lineal o bien un polinomio de 3.º o 4.º grado (**KUKA**).



Ver también (>>> [Fig. 17-3](#))

17.4 Conocer un bloque SPLINE

Descripción

Además de los pasos individuales con SPTP, SLIN, SCIRC, se dispone de un "bloque Spline". El bloque Spline es un tipo de movimiento especialmente apropiado para trayectorias curvas complejas. El bloque Spline se visualiza y se planifica como movimiento individual con una "trayectoria compleja".



Fig. 17-9

Existen dos clases de bloques Spline:

- **Bloque Spline**
Spline con movimientos de trayectoria (SPL, SLIN, SCIRC)
- **Bloque SPLINE PTP**
Spline con movimientos exclusivamente en la zona del eje (solo SPTP)

La trayectoria se planifica con todos los puntos, pasando por todos ellos. La trayectoria del bloque Spline se calcula previamente en su totalidad.

En las trayectorias Spline que tienen contornos muy estrechos, la velocidad siempre se reduce, ya que los ejes de los robots son los elementos que actúan como limitación.

17.4.1 Conocer la trayectoria SPLINE

Descripción

Trayectoria Spline

La trayectoria se define con puntos de apoyo, mediante los cuales se coloca la propia trayectoria. El robot calcula la trayectoria a partir de los puntos de apoyo. En los bloques Spline, además se pueden definir zonas especiales de desplazamiento constante. El curso de la trayectoria siempre es el mismo, independientemente del override, de la velocidad o de la aceleración. Los círculos y los radios estrechos se recorren con alta precisión.

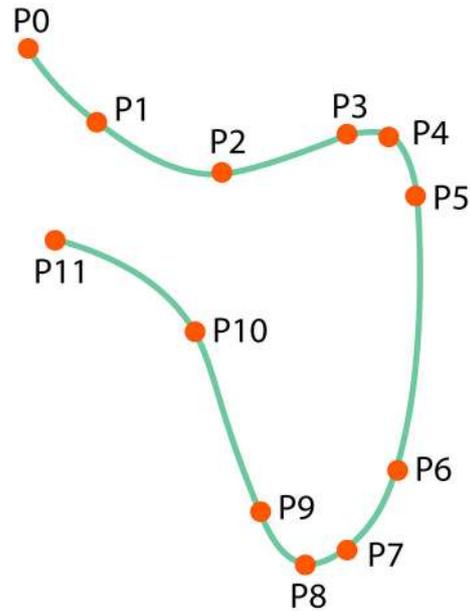


Fig. 17-10: Trayectoria curva, bloque SPLINE

17.4.2 Estructura de un bloque SPLINE

Descripción

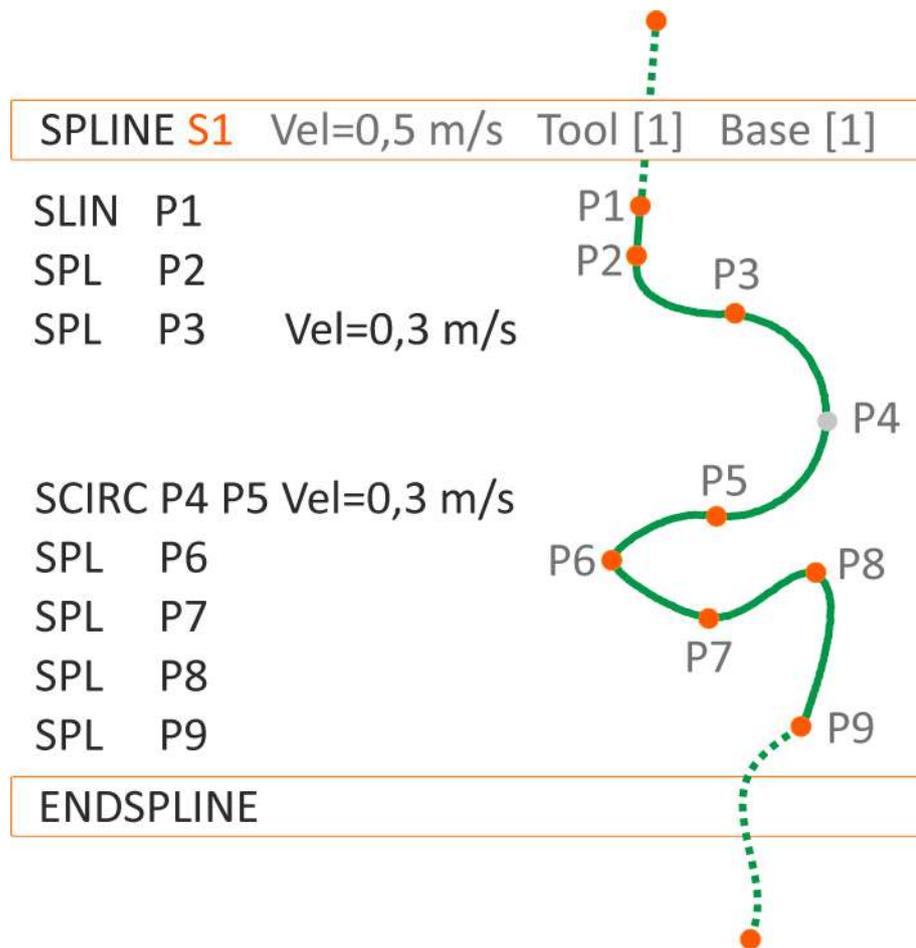


Fig. 17-11: Bloque SPLINE

- Un bloque Spline es un Fold (pliegue) interpolado por la unidad de control como un movimiento.
- En este pliegue se programan por aprendizaje puntos de apoyo que se interpolan como movimiento Spline.
- Además, el robot se puede forzar por una trayectoria lineal o circular en los diferentes segmentos Spline. Esto se realiza seleccionando el tipo de movimiento en el segmento respecto del punto de apoyo.
- De este modo se obtienen los siguientes tipos de movimiento para los diferentes segmentos Spline:
 - SLIN
 - SCIRC
 - SPL
- Para todo el bloque Spline, en el «encabezado» se establecen el nombre del movimiento y los parámetros globales.
 - Herramienta, base
 - Velocidad
 - Aproximación
 - Parámetros de movimiento
 - Detección de colisión con./desc.
- Cada uno de los diferentes movimientos programados por aprendizaje puede diferir.



Excepción: La herramienta y la base se establecen en el «encabezado».

- Además de los segmentos de movimiento, un bloque Spline puede incluir los siguientes elementos:
 - Instrucciones inline de paquetes de tecnología que cuentan con función Spline
 - Comentarios y líneas vacías
- La lógica Spline puede incluirse dentro del formulario inline (trigger, parada condicionada, velocidad constante)
Un bloque Spline no debe contener otras instrucciones.



Los puntos de delante y detrás del bloque Spline establecen el contorno con el que se va a entrar y salir del bloque Spline.



El punto de inicio de un bloque Spline es el último punto antes del bloque Spline.
El punto de destino de un bloque Spline es el último punto del bloque Spline.
Un bloque Spline no genera ninguna parada del movimiento de avance.

Procedimiento



Fig. 17-12

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Programación de movimientos Spline > Programar el bloque Spline con desarrollo del movimiento

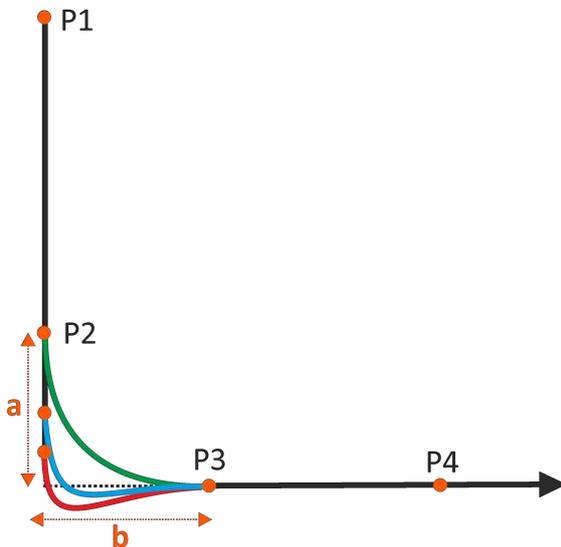
Podrá encontrar más información en la documentación del cliente.



Libro	KUKA System Software 8.6 Instrucciones de servicio y programación para los integradores de sistemas
Filtro de búsqueda	Documentación > Instrucciones de manejo y programación
Capítulo/apartado	Programación con formularios inline > Programar movimientos Spline

17.5 Indicaciones para la programación

Transición SLIN-SPL-SLIN



Para evitar una colisión durante la programación de una transición SLIN > SPL > SLIN, debe tenerse en cuenta lo siguiente:

- El punto de intersección de los dos segmentos SLIN se encuentra en sus prolongaciones.
- $2/3 \leq a/b \leq 3/2$
 a = Distancia del punto de inicio del segmento SPL al punto de intersección de los segmentos SLIN
 b = distancia del punto de intersección de los segmentos SLIN al punto de destino del segmento SPL

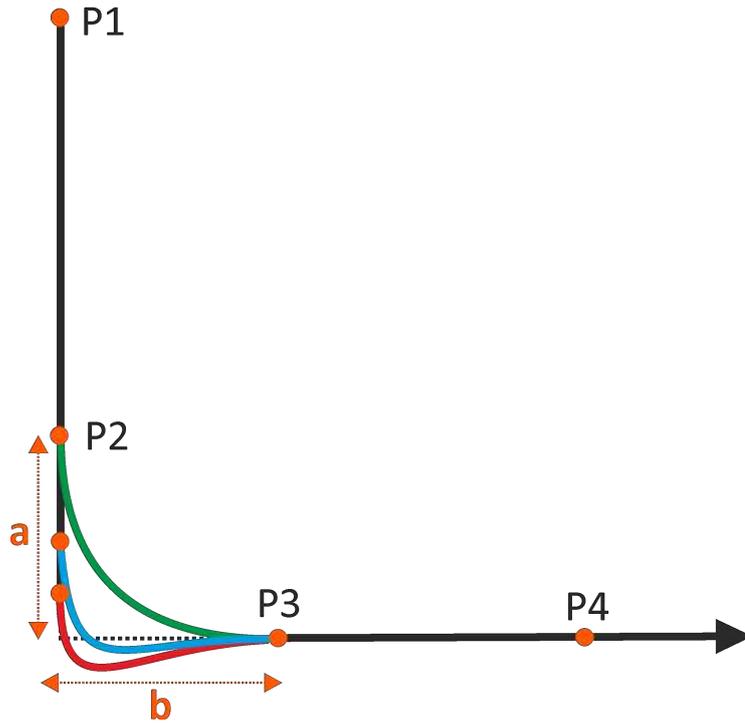


Fig. 17-13: Transición SLIN > SPL > SLIN

Para evitar una colisión durante la programación de una transición SLIN > SPL > SLIN, debe tenerse en cuenta lo siguiente:

- El punto de intersección de los dos segmentos SLIN se encuentra en sus prolongaciones.
- $2/3 \leq a/b \leq 3/2$
 a = Distancia del punto de inicio del segmento SPL al punto de intersección de los segmentos SLIN
 b = distancia del punto de intersección de los segmentos SLIN al punto de destino del segmento SPL

17.5.1 Modificar puntos en el bloque SPLINE

17.5.1.1 Modificar puntos en movimientos SPLINE combinados

Ejemplo 1

- **Trayectoria original:**

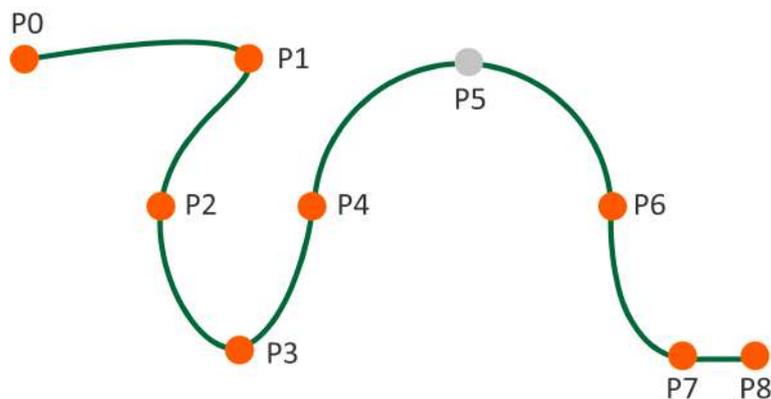


Fig. 17-14: SPLINE, trayectoria original

```
SPTP P0
SPLINE S1
  SPL P1
  SPL P2
  SPL P3
  SPL P4
  SCIRC P5, P6
  SPL P7
  SLIN P8
ENDSPLINE
```

- **Respecto a la trayectoria original, se desplaza el punto 3:**

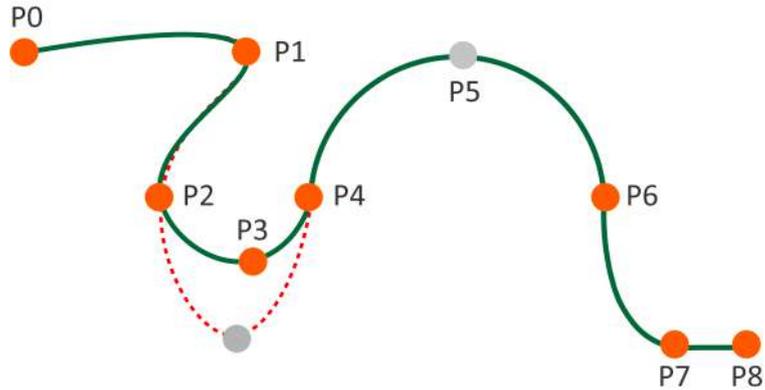


Fig. 17-15: Punto desplazado

- P3 se desplaza.
- Así, se modifica la trayectoria en los segmentos P1 - P2, P2 - P3 y P3 - P4.
- El segmento P4 - P5 no se modifica en este caso porque pertenece a un SCIRC, definiéndose así una trayectoria circular.
- **El tipo de trayectoria se modifica respecto de la trayectoria original.**

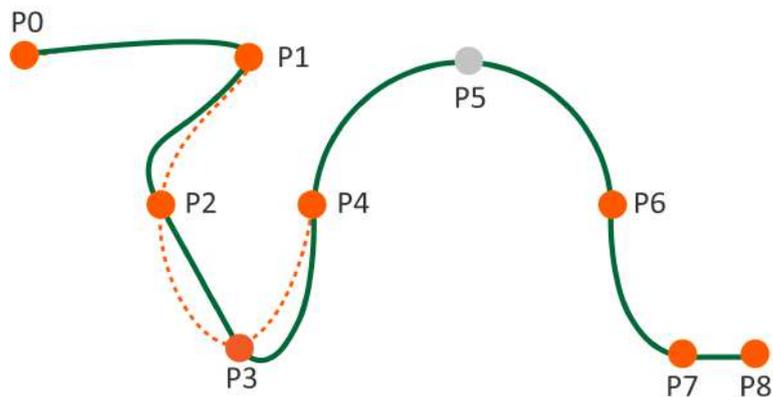


Fig. 17-16: Tipo de trayectoria modificada

```
SPTP P0
SPLINE S1
  SPL P1
  SPL P2
  SLIN P3
  SPL P4
  SCIRC P5, P6
  SPL P7
  SLIN P8
```

ENDSPLINE

- En el caso de la trayectoria original el tipo de segmento cambia de P2 - P3 de SPL a SLIN.
- La trayectoria cambia en los segmentos P1 - P2, P2 - P3 y P3 - P4.

17.5.1.2 Modificar puntos en movimientos SPLINE con segmentos SPL puros

Ejemplo 2

En este ejemplo, se programó un trayecto exclusivamente con segmentos SPL.

- **Trayectoria original:**

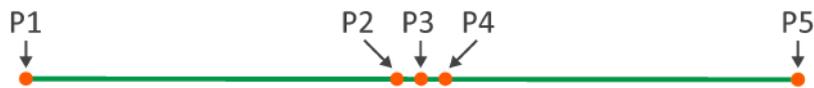


Fig. 17-17: SPL, trayectoria original

- **Respecto a la trayectoria original, se desplaza un punto:**

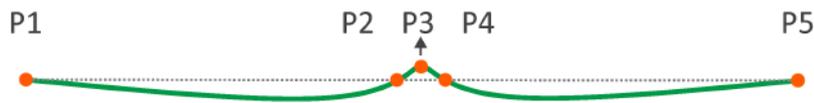


Fig. 17-18: SPL, desplazamiento de puntos

- P3 se desplaza.
- Por ello la trayectoria cambia en todos los segmentos representados.
- Dado que P2 - P3 y P3 - P4 son segmentos muy cortos y P1 - P2 y P4 - P5 segmentos largos, el desplazamiento pequeño provoca un cambio importante de la trayectoria.



- Distribuir las distancias de puntos de forma más constante
- Programar las rectas (excepto las rectas muy cortas) como segmentos SLIN

17.5.2 Evitar la reducción de velocidad no deseada en los movimientos Spline

17.5.2.1 Reducción de velocidad mediante segmentos de movimiento consecutivos sin SPL

Descripción

Reducción de la velocidad a 0

Este caso se presenta en las situaciones siguientes:

- Puntos consecutivos con coordenadas idénticas.

- Segmentos **SLIN** y/o **SCIRC** consecutivos dentro del bloque Spline.
Causa: Transcurso discontinuo de la dirección de la velocidad.

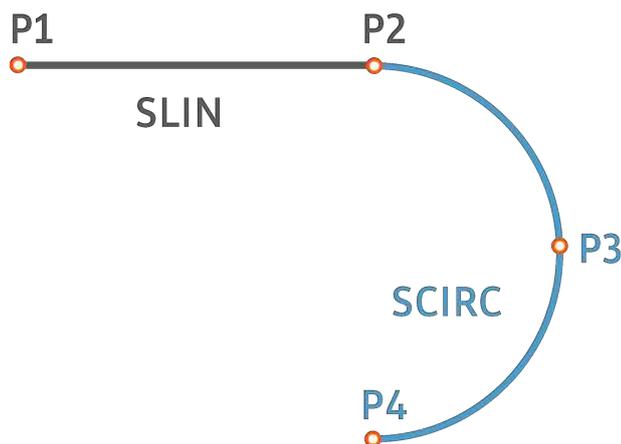


Fig. 17-19: Parada exacta en P2

En el caso de **transiciones SLIN-SCIRC** la velocidad también será 0 cuando la recta pasa tangencialmente por el círculo porque, a diferencia de la recta, el círculo es una curva.

- Segmentos consecutivos de **SLIN** a **SLIN** dentro del bloque Spline.

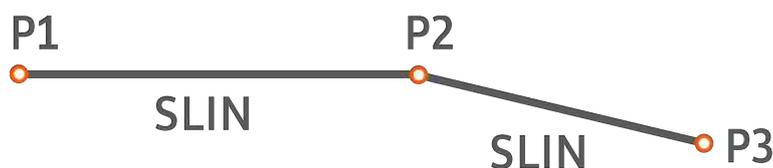


Fig. 17-20: Parada exacta en P2

Solución

Al insertar un segmento **SPL** entre **SLIN** y **SCIRC**, se evita una parada exacta y el robot puede ejecutar el movimiento a la velocidad programada en una sola pieza.

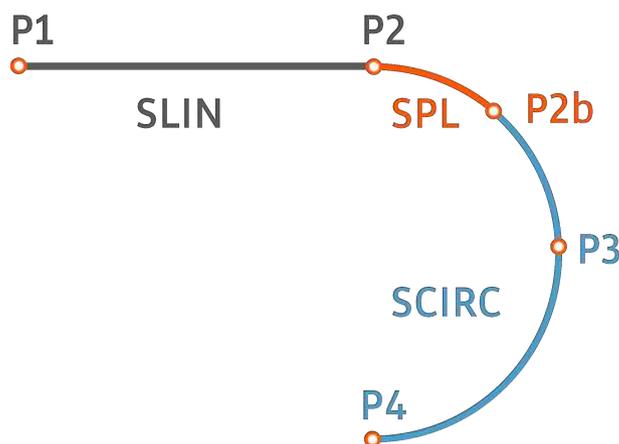


Fig. 17-21: Transición SLIN-SCIRC con SPL

Otro ejemplo



Excepciones

- Si se suceden una serie de segmentos SLIN que crean una recta y que cambian la orientación uniformemente, la velocidad no se reduce.



Fig. 17-22: P2 es pasado sin parada exacta

- En una transición SCIRC-SCIRC la velocidad no se reduce cuando ambos círculos o segmentos circulares tienen el mismo punto central y el mismo radio, ni cuando las orientaciones cambian uniformemente. (Aprendizaje difícil, por eso hay que calcular y programar los puntos.)



Los círculos con el mismo centro y el mismo radio a veces se programan para obtener círculos de $\geq 360^\circ$. Otra posibilidad más sencilla consiste en programar un ángulo circular.

17.5.2.2 Reducción de la velocidad debida a una programación por aprendizaje no uniforme

Descripción

Reducción de la velocidad debida a una programación por aprendizaje no uniforme

- A menudo, las modificaciones en la orientación, incluso en relación con los ejes adicionales, de la longitud cartesiana del arco distribuidas de una forma irregular provocan caídas de velocidad indeseadas.
- Una distribución irregular de las trayectorias de orientación en el trayecto cartesiano (longitud del arco) debe acelerar o retardar la orientación muy a menudo, hecho que también tiende a producirse con los grandes tirones de orientación.
- En consecuencia, una distribución irregular provoca muchos más descensos en la velocidad que una distribución constante de los trayectos de orientación.

Solución



Mantener la mayor uniformidad posible en la orientación y en los ejes externos

Ejemplo de una distribución regular:

```

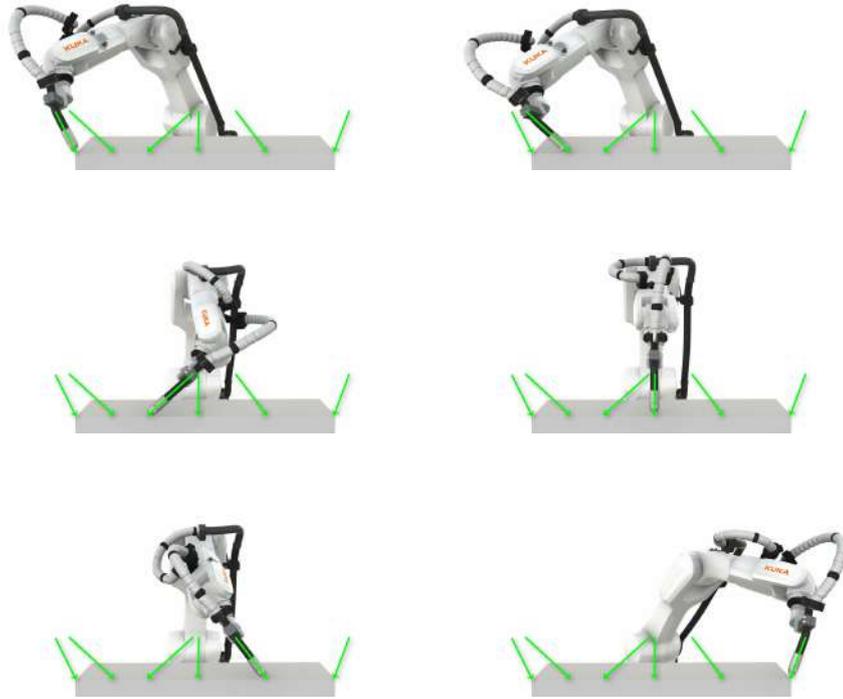
; punto de inicio del Spline
SPTP {x 0, y 0, z 0, A 0, B 0, C 0}
SPLINE S1
    
```

Configuración de una nueva estación de robot: Programación de un movimiento SPLINE

```

; 0,1° Cambio de orientación por mm de trayecto Carrera
SPL {x 0, y 100, z 0, A 10, B 0, C 0}
; 0,1° Cambio de orientación por mm de trayecto Carrera
SPL {x 0, y 110, z 0, A 11, B 0, C 0}
; 0,1° Cambio de orientación por mm de trayecto Carrera
SPL {x 0, y 310, z 0, A 31, B 0, C 0}
ENDSPLINE
    
```

Con orientación programada



Flecha verde

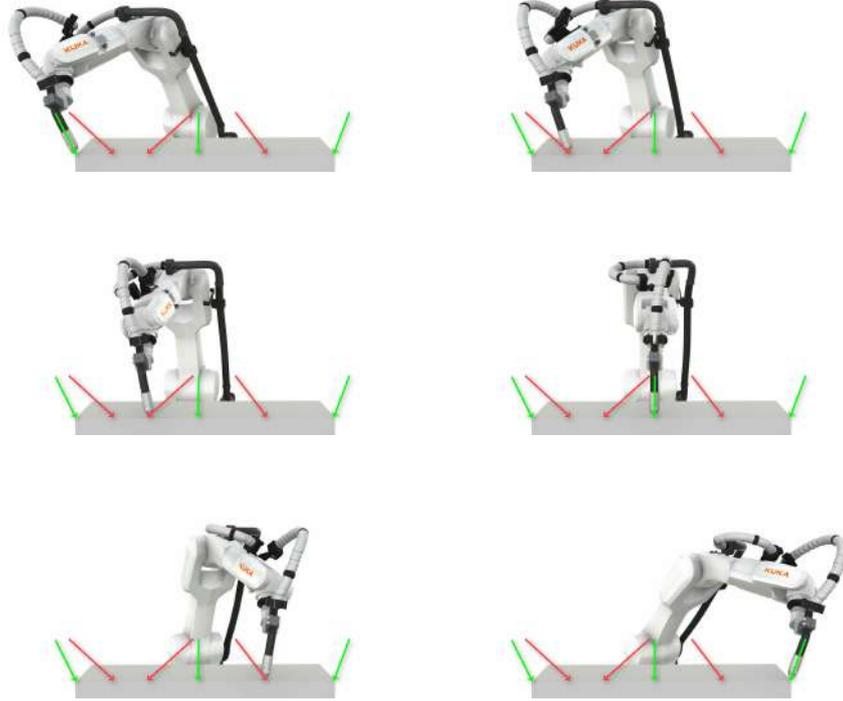
posición programada por aprendizaje con la orientación correspondiente

Conclusión

AVISO

- Muchas veces se programan muchos puntos guardando una distancia relativamente pequeña.
- Lo importante es, sobre todo, la trayectoria cartesiana (x, y, z).
- Pero el spline también interpola la orientación programada, lo cual puede producir reducciones de velocidad.
- Por esta razón, en un caso así, es preferible seleccionar **IGNORE** en el formulario inline.

Sin orientación programada



Flecha verde

posición programada por aprendizaje con la orientación correspondiente

Flecha roja

posición programada por aprendizaje con la orientación correspondiente, su orientación no se acepta.

17.6 Ejercicio (variante 1): Programar el contorno de la trayectoria con bloque Spline

¡Ahora es su turno!



Enunciado 1

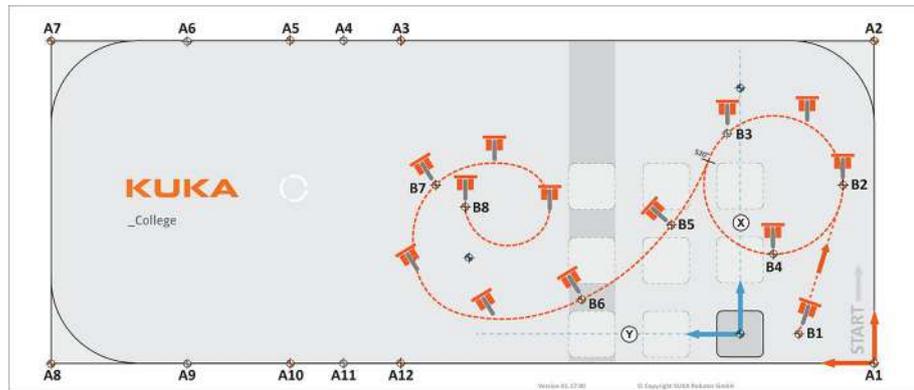


Fig. 17-23: Contorno de la trayectoria con bloque SPLINE

1. Cree un nuevo programa: _____.
2. Utilizar como herramienta n.º _____ nombre: _____ y base n.º _____ nombre: _____.
3. La velocidad en la trayectoria debe ser de 0,25 m/s.
4. Programe el contorno SPLINE en la lámina. Para ello, utilice un bloque SPLINE, SPL, SLIN y SCIRC.
5. El primer paso es programar por aprendizaje los puntos B1 a B8 indicados.
 - Tener en cuenta la orientación de la herramienta.
 - El ángulo circular asciende a 520 °.
6. Comprobar el programa en el modo de servicio T1.



La trayectoria no se corresponde con el contorno deseado.

AVISO

- Desplace el robot hasta el punto de inicio del contorno SPLINE y compruebe la posición del robot.
- Desplace el robot con las teclas de desplazamiento en relación con el sistema de coordenadas de base en dirección X e Y. Orientar el robot solo en torno al eje de giro A.
- Un cambio pronunciado de orientación conlleva grandes caídas de velocidad en la trayectoria de spline.

Enunciado 2

1. Amplíe el contorno Spline mediante varios puntos.

Nómbrelos de forma práctica y delimítelos nominalmente de los puntos de apoyo ya programados.

2. Comprobar el programa en el modo de servicio T1.
 3. Amplíe el contorno Spline mediante varios puntos hasta que la trayectoria se corresponda con el resultado deseado.
1. Duplique el **programa:** _____ y denomínelo **programa:** _____ .
 2. Modifique el control de orientación de los puntos programados a una reorientación constante.

Opcional: Enunciado

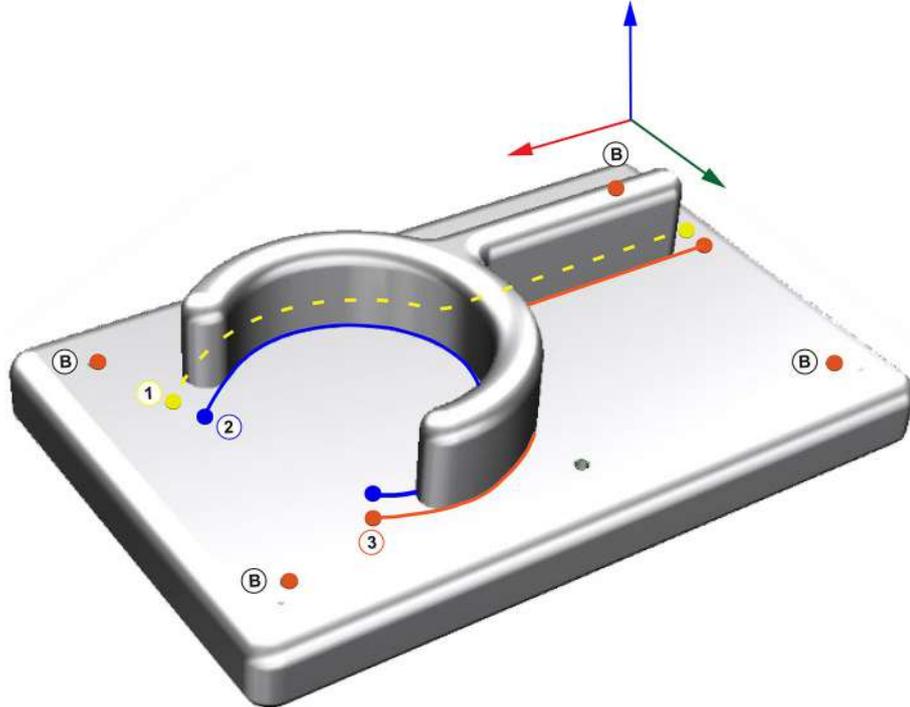


Fig. 17-24: Componente de soldadura

1. Colocar la pieza soldada sobre el nivel plano de la mesa conforme a las indicaciones del formador.

AVISO
Para ello, retirar la lámina magnética de la mesa.

2. Medir la base *n.º* _____ *nombre:* _____ sobre la pieza soldada mediante los orificios de las esquinas (B) conforme a las indicaciones del formador.
3. Utilizar para ello la herramienta *n.º* _____ *nombre:* _____.
4. Cree un nuevo **programa:** _____
5. Programe las trayectorias indicadas en el plano (1), (2) y (3) mediante los bloques SPLINE individuales.
6. La velocidad en la trayectoria debe ser de 0,25 m/s.

Opcional: Enunciado

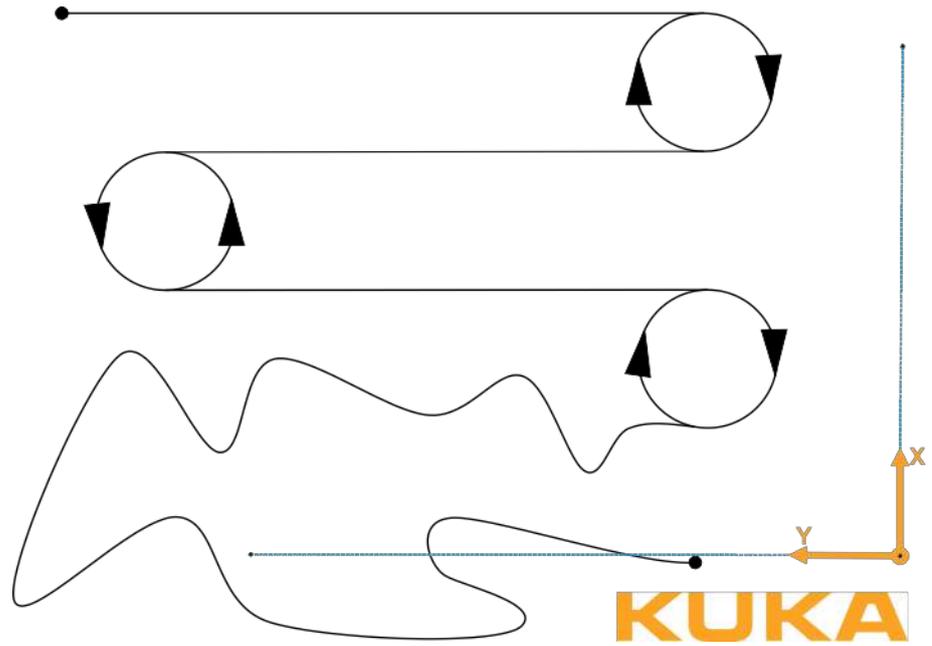


Fig. 17-25: Contorno de trayectoria del bloque SPLINE

1. Posicione la lámina Spline sobre el nivel plano de la mesa.
2. Mida la base especificada *n.º* _____ *nombre:* _____ sobre la lámina SPLINE.
3. Cree un nuevo programa: _____ .
4. Utilizar una herramienta adecuada.
5. Programe la trayectoria marcadas mediante el bloque SPLINE.
6. La velocidad en la trayectoria debe ser de 0,25 m/s.

17.7 Ejercicio (variante 2): Programar el contorno de soldadura con ejes adicionales

¡Ahora es su turno!



Enunciado

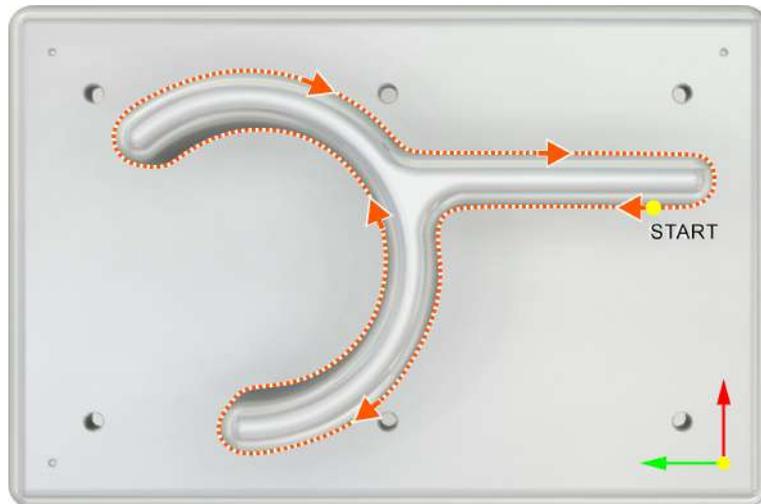


Fig. 17-26: Trayectoria de soldadura

1. Coloque el componente de soldadura en el miniDKP.
2. Cree un nuevo programa con el **nombre:** _____
3. Programe el contorno de soldadura (soldadura en ángulo) sobre el componente. Utilice para ello la herramienta medida y la base.



Fig. 17-27: Orientación del soplete

- La velocidad de desplazamiento sobre la mesa de trabajo es de 0,3 m/s.

- Alinee la herramienta "Soplete" en el ángulo correcto con respecto al componente.
 - Durante la creación de programas utilice los ejes externos existentes de tal manera que el miniDKP College gire siempre en dirección contraria a la herramienta.
4. Comprobar el programa en los modos de servicio T1, T2 y automático. Para ello se deben tener en cuenta las prescripciones de seguridad enseñadas.

17.8 Preguntas: Contorno de la trayectoria con bloque SPLINE

Lo que se debe saber tras el ejercicio:



¿Cuáles son los comandos de desplazamiento que pueden utilizarse en un bloque SPLINE PTP?

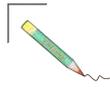


Fig. 17-28: Campo de respuesta

¿Dónde se encuentra el punto de inicio de un bloque SPLINE?

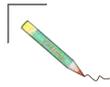


Fig. 17-29: Campo de respuesta

¿En qué condiciones no se alcanza la velocidad programada de los movimientos SPLINE?

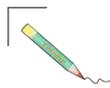


Fig. 17-30: Campo de respuesta

¿Qué se debe tener en cuenta al modificar movimientos SPLINE?

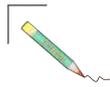


Fig. 17-31: Campo de respuesta

18 Configuración de una nueva estación de robot: Agarrar piezas de trabajo

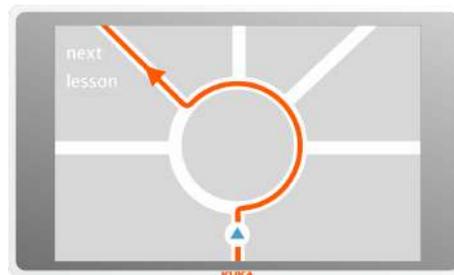
18.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para crear programas propios con formularios inline y las trayectorias de los movimientos SPTP, SLIN y SCIRC.

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Agarrar piezas de trabajo

- Aprenderá cómo programar las garras mediante el formulario inline.
- Aprenderá a comprobar el estado de la garra actual a través del programa.

18.2 Descripción de la situación

Descripción

Las herramientas, como las garras de la célula de formación (3), deben abrirse o cerrarse durante el tiempo de ejecución del programa. Se logra una modificación del estado mediante las salidas configuradas (\$OUT). La herramienta indica el estado actual en la unidad de control mediante las entradas (\$IN). Por norma general, la transmisión a la garra se realiza a través de un sistema configurado del bus de campo mediante los módulos digitales de entrada y salida (2).

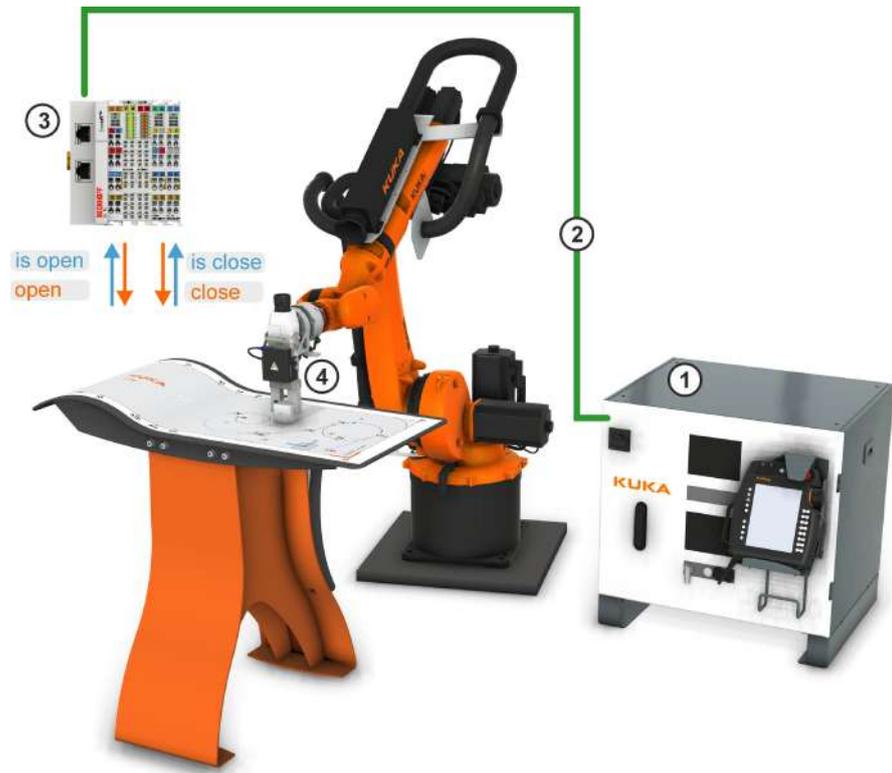


Fig. 18-1: Control de la garra

- | | | | |
|---|-------------------|---|--|
| 1 | Unidad de control | 3 | Módulo de bus de campo con módulos de entrada y salida |
| 2 | Bus de campo | 4 | Garras con entradas y salidas digitales |

Asignación de la garra en el ejemplo de las células de formación College

Abrir garra	\$OUT[25]	Garra abierta	\$IN[25]
Cerrar garra	\$OUT[26]	Garra cerrada	\$IN[26]

18.3 Conocer los formularios inline para la garra

Descripción

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 para controlar la garra.

- **Gripper SET**
Instrucción para abrir y cerrar la garra College en el programa.
- **Gripper SYN SET**
Instrucción para abrir y cerrar la garra College en el programa. Además, el punto de conmutación temporalmente y en relación a la trayectoria.

Funciones de la programación de garra

Básicamente, es posible programar la instrucción de garra de tal manera que se ejecute respecto al punto de inicio o de destino.

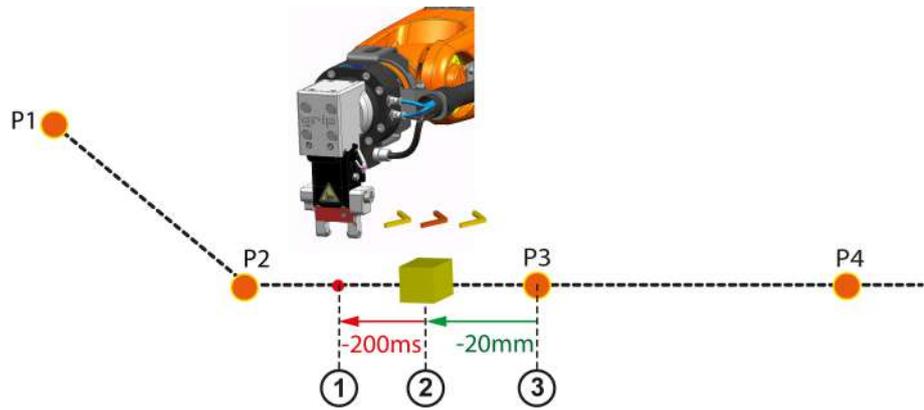


Fig. 18-2: Ejemplo de funciones de agarre con desplazamiento local y temporal

Pos.	Descripción
1	Abrir garra
2	Posición de la garra
3	Punto programado

- En el ejemplo, un cubo (2) que se debe recoger está situado 20 mm por delante del punto programado P3 (3).
- 200 ms antes de alcanzar el cubo (1) debe abrirse la garra del robot.

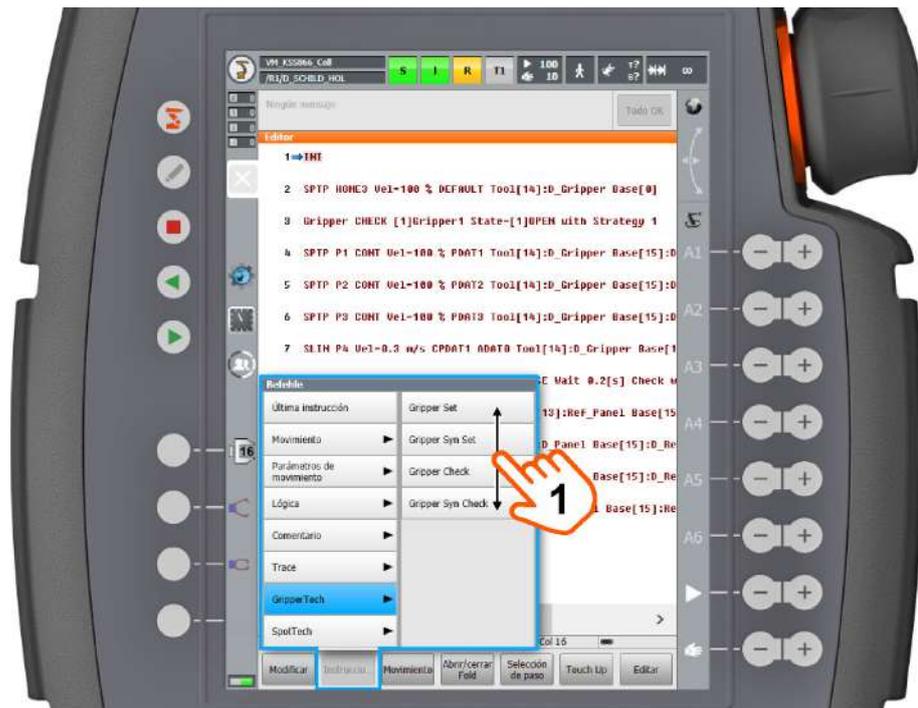


ADVERTENCIA

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.

18.4 Añadir comandos de la garra en un programa

Ruta de menú: *Comandos > GripperTech>*



Se pueden añadir los siguientes formularios inline

- Gripper Set
- Gripper Syn Set
- Gripper Check
- Gripper Syn Check

Gripper SET

El estado de conmutación de la garra se establece con el formulario inline.

El estado de conmutación puede conectarse en la ejecución principal o en la alimentación.

En caso de utilizarse en la marcha principal, tras la acción de las garras debe esperarse un tiempo determinado hasta que se haya arrancado el movimiento.

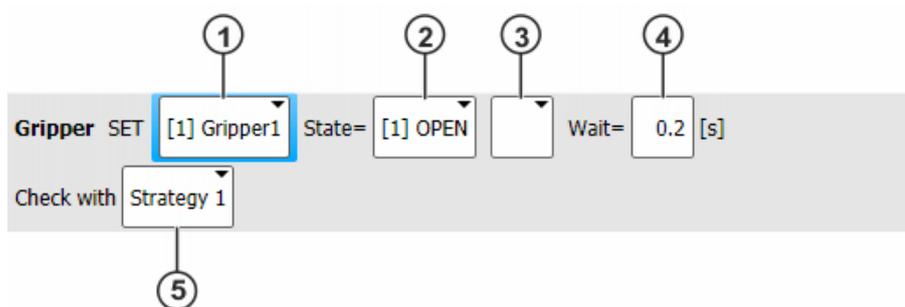


Fig. 18-3: Formulario inline Gripper SET

Pos.	Descripción
1	Garra activa Solo se muestran las garras activas.
2	Estado de conmutación establecido de la garra. El número de estados de conmutación seleccionables y su denominación depende de la configuración.
3	<ul style="list-style-type: none"> • CONT: La alimentación establece el estado de conmutación. • [vacío]: La alimentación se establece con la marcha principal.
4	Tiempo de espera El programa se reanuda una vez transcurrido el tiempo de espera indicado. <ul style="list-style-type: none"> • 0,0 ... 10,0 s Por defecto: 0,2 s Este campo se muestra solo si se conecta con la marcha principal.
5	Estrategia de error Si hay un estado que no existe, puede configurarse una estrategia de error con esta finalidad. <ul style="list-style-type: none"> • No Check: El programa se reanuda sin comprobar si existe el estado de conmutación. • Strategy 1...3: Estrategias de error configuradas

Gripper SYN SET

Con el formulario inline se puede establecer un estado de conmutación en el punto de inicio o en el punto de destino del movimiento.

El estado de conmutación puede desplazarse en el espacio y/o en el tiempo.

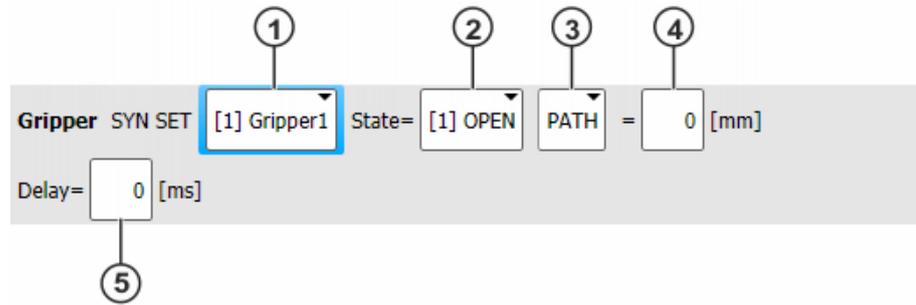


Fig. 18-4: Formulario inline Gripper SYN SET

Pos.	Descripción
1	Garra activa Solo se muestran las garras activas.
2	El estado de conmutación de la garra que se establece El número de estados de conmutación seleccionables y su denominación depende de la configuración.
3	Puntos a los que hace referencia Gripper SYN SET <ul style="list-style-type: none"> • START: Punto de referencia inicial del desplazamiento temporal • END: Punto de referencia final del desplazamiento temporal • PATH: La selección del estado de conmutación se fundamenta en el punto de destino del movimiento. Además del desplazamiento en el espacio, es posible un desplazamiento en el tiempo.
4	Este campo únicamente se visualiza cuando está seleccionado PATH . Distancia del punto de conmutación desde el punto de destino <ul style="list-style-type: none"> • -2000 ... +2000 mm
5	Desplazamiento en el tiempo de la acción de conmutación <ul style="list-style-type: none"> • -1000 ... 1000 ms El tiempo se indica en valores absolutos. El punto de conmutación cambia en función de la velocidad del robot.

Ejemplo SYN SET

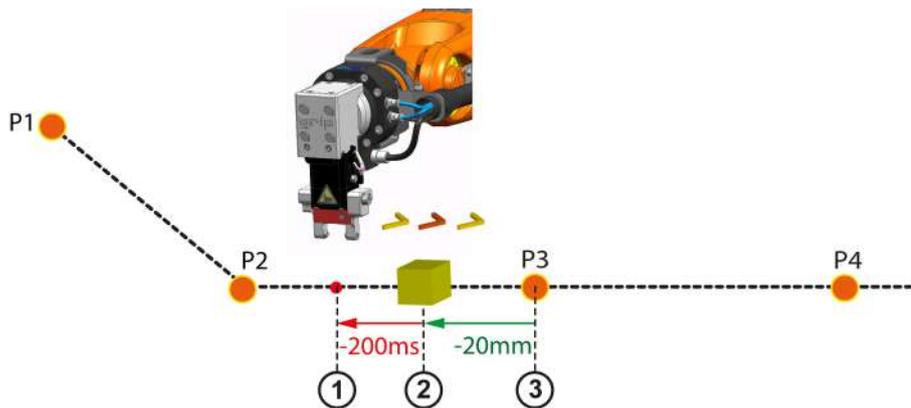


Fig. 18-5: Ejemplo de funciones de agarre con desplazamiento local y temporal

```

1 INI
2
3 SPTP HOME Vel=100% % DEFAULT
4 SLIN P1 Vel=2m/s CPDAT1 Tool[1] Base[1]
5 SLIN P2 Vel=0.2m/s CPDAT2 Tool[1] Base[1]
6 ; *** Abrir garra ***
7 Gripper Syn SET [1]Gripper State=[1]OPEN Path=-20mm
  Delay=-200ms
8 ; *** Cerrar garra ***
9 Gripper Syn SET [1]Gripper State=[1]CLOSE Path=-20mm
  Delay=0ms
10 SLIN P3 Vel=0.2m/s CPDAT3 Tool[1] Base[1]
16 ...
    
```

18.5 Comprobar los estados de la garra con formularios inline

Descripción

Con el paquete de tecnología KUKA.GripperTech es posible comprobar, directamente en el programa seleccionado, los estados de la garra utilizando formularios inline predefinidos.

Existen dos instrucciones disponibles

- **Gripper CHECK**
 La comprobación del momento de conmutación con respecto a la marcha principal o en avance.
 (>>> [18.5 "Comprobar los estados de la garra con formularios inline" Página 262](#))
- **Gripper SYN CHECK**
 La comprobación del momento de conmutación puede desplazarse en el tiempo y en el espacio.
 (>>> [18.5 "Comprobar los estados de la garra con formularios inline" Página 262](#))

Gripper CHECK

El estado de conmutación de la garra se comprueba mediante el formulario inline.

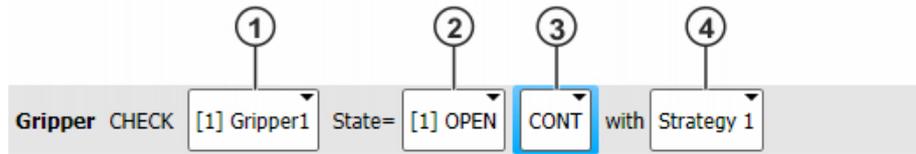


Fig. 18-6: Formulario inline Gripper CHECK

Pos.	Descripción
1	Garra activa Solo se muestran las garras activas.
2	Estado de conmutación de la garra que se comprueba El número de estados de conmutación seleccionables y su denominación depende de la configuración.
3	<ul style="list-style-type: none"> CONT: El estado de conmutación se comprueba durante el avance. [vacío]: La señal se comprueba con el puntero de movimiento de avance.
4	Estrategia de error Si hay un estado que no existe, puede configurarse una estrategia de error con esta finalidad. <ul style="list-style-type: none"> No Check: El programa se reanuda sin comprobar si existe el estado de conmutación. Strategy 1...3: Estrategias de error configuradas

Gripper SYN CHECK

Con el formulario inline se puede supervisar un estado de conmutación en el punto de inicio o en el punto de destino del movimiento.

La comprobación del estado de conmutación puede desplazarse en el espacio y/o en el tiempo.

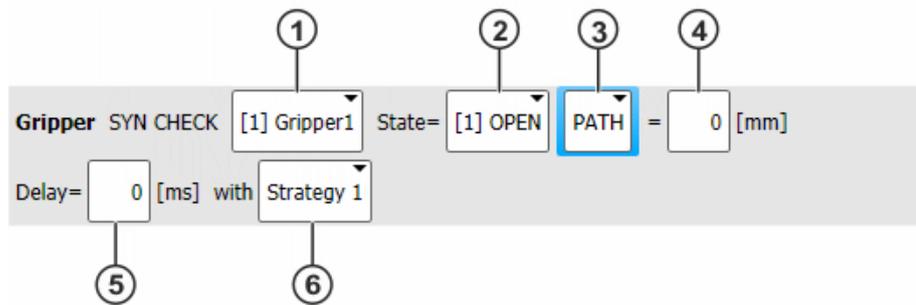


Fig. 18-7: Formulario inline Gripper SYN CHECK

Pos.	Descripción
1	Garra activa Solo se muestran las garras activas.
2	Estado de conmutación de la garra que se comprueba El número de estados de conmutación seleccionables y su denominación depende de la configuración.

Pos.	Descripción
3	<p>Puntos a los que hace referencia Gripper SYN CHECK</p> <ul style="list-style-type: none"> • START: Desplazamiento en el tiempo con respecto al punto de inicio • END: Desplazamiento en el tiempo con respecto al punto de destino • PATH: La comprobación se basa en el punto de destino del movimiento. Además de en el tiempo, se puede realizar un desplazamiento en el espacio.
4	<p>Este campo únicamente se visualiza cuando está seleccionado PATH.</p> <p>Distancia del punto de conmutación desde el punto de destino</p> <ul style="list-style-type: none"> • -2000 ... +2000 mm
5	<p>Desplazamiento en el tiempo de la comprobación</p> <ul style="list-style-type: none"> • -1000 ... 1000 ms <p>El tiempo se indica en valores absolutos. El punto de conexión de la comprobación cambia en función de la velocidad del robot.</p>
6	<p>Estrategia de error</p> <p>Si hay un estado que no existe, puede configurarse una estrategia de error con esta finalidad.</p> <ul style="list-style-type: none"> • No Check: El programa se reanuda sin comprobar si existe el estado de conmutación. • Strategy 1...3: Estrategias de error configuradas

18.6 Procedimiento

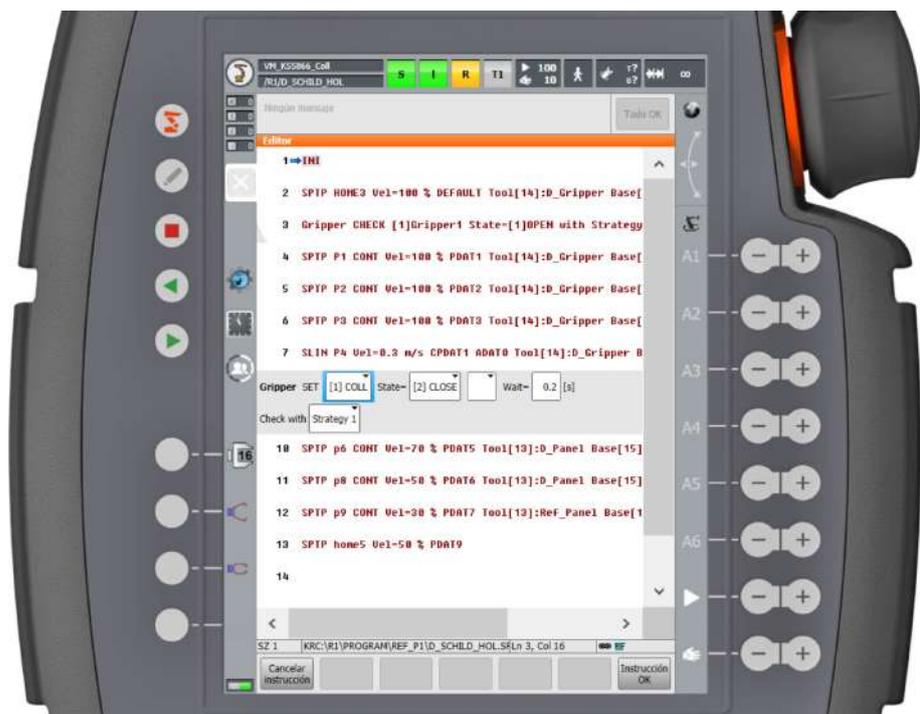


Fig. 18-8: Formulario inline "Gripper SET"

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer el paquete de opciones KUKA.Gripper y SpotTech

Manejar y programar las garras a través del paquete de opciones KUKA.Gripper y SpotTech

Podrá encontrar más información en la documentación del cliente.



Libro

KUKA.GripperSpotTech 5.0

Para KUKA System Software 8.5, 8.6 y 8.7

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

18.7 Ejercicios: Introducción



Estos ejercicios solo deben realizarse si no ve ninguna necesidad de realizar ejercicios en relación con el tema "TCP externo" durante el transcurso posterior de la formación. Sin embargo, desea perfeccionar el depósito y agarre de objetos y convertirse en el especialista en tareas de manipulación en su empresa.

Su formador estará encantado de asesorarle sobre los ejercicios que aún están preparados para usted y los estructurará de forma consecutiva.

18.8 Ejercicio (variante 1): Tomar y descargar el cubo

¡Ahora es su turno!



Enunciado



1. Cree un **programa**: _____, utilice para ello la herramienta *n.º*: _____ *nombre*: _____ y la base *n.º*: _____ *nombre*: _____.
2. El robot debe agarrar un cubo del cargador y desplazarse con él hasta a la posición Home.
3. Comprobar el programa en los modos de servicio T1 y T2. Para ello se deben tener en cuenta las prescripciones de seguridad enseñadas.



1. Cree un segundo **programa:** _____ y utilice para ello una base necesaria y la herramienta correspondiente.
2. El robot debe volver a introducir el cubo en el depósito de cubos.
3. Comprobar el programa en los modos de servicio T1 y T2. Para ello se deben tener en cuenta las prescripciones de seguridad enseñadas.

18.9 Ejercicio (variante 2): Tomar y descargar placa

¡Ahora es su turno!



Enunciado



1 Cartel

2 Posición de descarga

1. Crear un nuevo programa: _____ que recoge la placa.

- **Herramienta**

N.º _____ nombre: _____

- **Base**

N.º _____ nombre: _____

AVISO

Al agarrar o descargar la placa, tener en cuenta los datos de carga modificados. Para ello, debe cambiar la herramienta en el formulario inline de la garra a la placa y viceversa.

2. Programar el proceso para **Tomar placa**, de modo que se obtenga la posición de depósito y de recogida representada en la figura . Para ello, reducir la velocidad a $0,3 \text{ m/s}$ durante la retirada y colocación de la placa.

3. Comprobar el programa en los modos de servicio T1, T2 y automático. Para ello se deben tener en cuenta las prescripciones de seguridad enseñadas.
4. Crear un segundo **programa:**_____ para depositar la placa. Para ello, utilizar una base adecuada y la herramienta correspondiente.
5. Programar por aprendizaje el proceso **Descargar placa**.
6. Comprobar el programa en los modos de servicio T1, T2 y automático.

18.10 Preparación: Fijación de la placa de aplicación 3D

1. La placa de aplicación 3D (A) se coloca sobre la mesa (B) en el lado derecho orientado hacia el robot. Para ello debe tenerse en cuenta que los topes 1, 2 y 3 estén situados en los bordes de la mesa. Ver también (>>> Fig. 18-10) el punto 1.

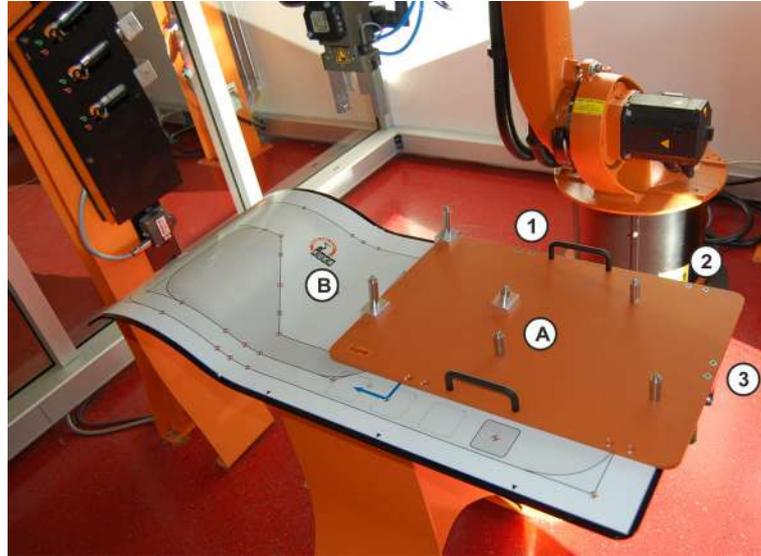


Fig. 18-9: Fijar la placa de aplicación 3D 1

2. Tras el control de los topes (1), la placa se fija por los dos lados con los tres tensores. Para ello, mover el agarre del tensor correspondiente (2) hacia el centro de la mesa. Comprobar si los amortiguadores de goma (3) están fijos.

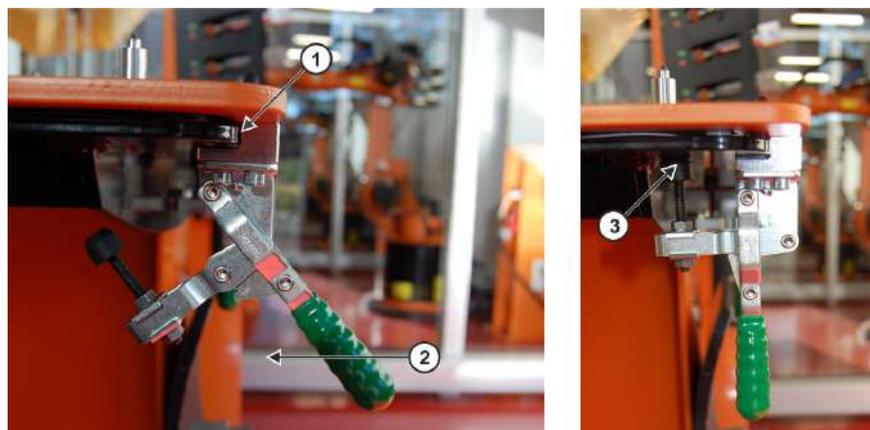


Fig. 18-10: Fijar la placa de aplicación 3D 2

3. Sobre la propia placa de aplicación 3D se puede depositar el componente 3D en los puntos de depósito A o B.

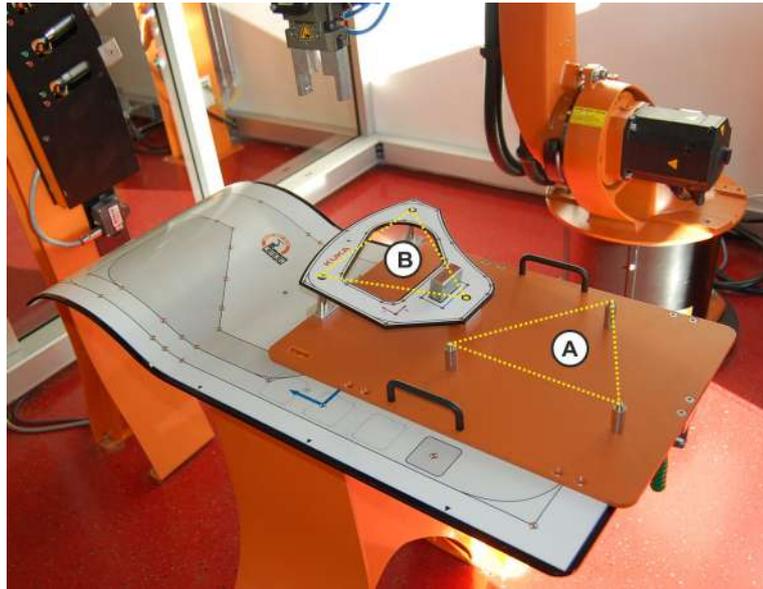


Fig. 18-11: Puntos de depósito para el componente 3D

18.11 Ejercicio (variante 3): Recoger y depositar el componente 3D

¡Ahora es su turno!



Enunciado

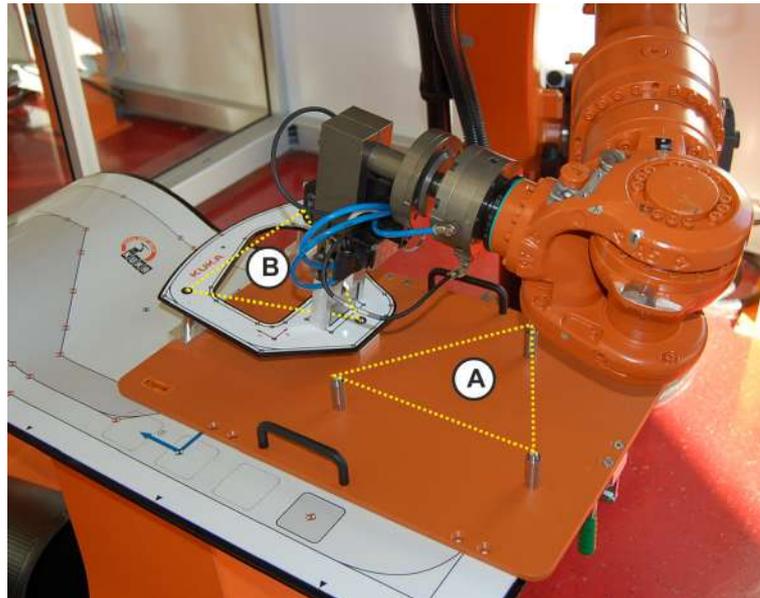


Fig. 18-12: Cambiar la posición del componente 3D

A Posición del componente A B Posición del componente B

1. Cree un nuevo **programa**:_____ para la recogida de la placa. Utilice opcionalmente la posición A o B.

- **Herramienta**

N.º _____ nombre: _____

- **Base**

N.º _____ nombre: _____

AVISO

Al agarrar o descargar la placa, tener en cuenta los datos de carga modificados. Para ello, debe cambiar la herramienta en el formulario inline de la garra a la placa y viceversa.

2. Programe el proceso **Tomar_placa**, de modo que se obtenga la posición de depósito y de recogida representada en la figura. Para ello, reduzca la velocidad a $0,3 \text{ m/s}$ durante la retirada y colocación de la placa.
3. Comprobar el programa en los modos de servicio T1, T2 y automático. Para ello se deben tener en cuenta las prescripciones de seguridad enseñadas.

4. Cree un segundo **programa:**_____ . Utilice para ello una base adecuada y la herramienta correspondiente.
5. Programar por aprendizaje el proceso **Descargar placa**.
6. Comprobar el programa en los modos de servicio T1, T2 y automático.

18.12 Preguntas: Agarrar piezas de trabajo

Lo que se debe saber tras el ejercicio:



¿Qué formularios inline están disponibles con el paquete de tecnología **KUKA.GripperTech**? Describir su función.

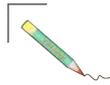


Fig. 18-13: Campo de respuesta

¿Qué causa el tiempo de espera en el comando *Gripper-Set*?

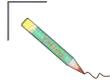


Fig. 18-14: Campo de respuesta

En el ejercicio anterior se debía modificar el número de herramienta con la recepción de la placa. ¿Qué efecto tuvo esto para el robot?



Fig. 18-15: Campo de respuesta

19 Configuración de una nueva estación de robot: Liberación de la estación

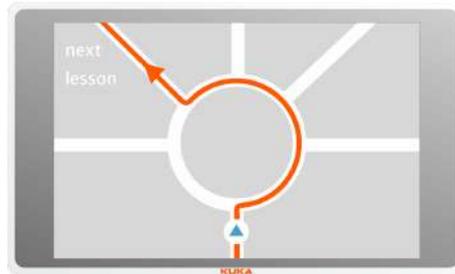
19.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Liberación de la estación

- Obtendrá una primera idea de cómo se pueden utilizar entradas y salidas digitales en una instalación.
- Sabrá cómo comprobar el estado actual de las entradas y salidas en el smartPAD.
- Podrá comprobar estados de flags, contadores y temporizadores.

19.2 Descripción de la situación

En esta célula de robot se prepara un bastidor para la soldadura. Un robot coloca el nuevo bastidor en el amarre. Los dos robots de soldadura a la izquierda y a la derecha no deben moverse en ningún caso y esperan hasta que el proceso haya finalizado.

Solo cuando la zona de soldadura esté libre se puede iniciar el proceso de soldadura después de una "señal de habilitación".



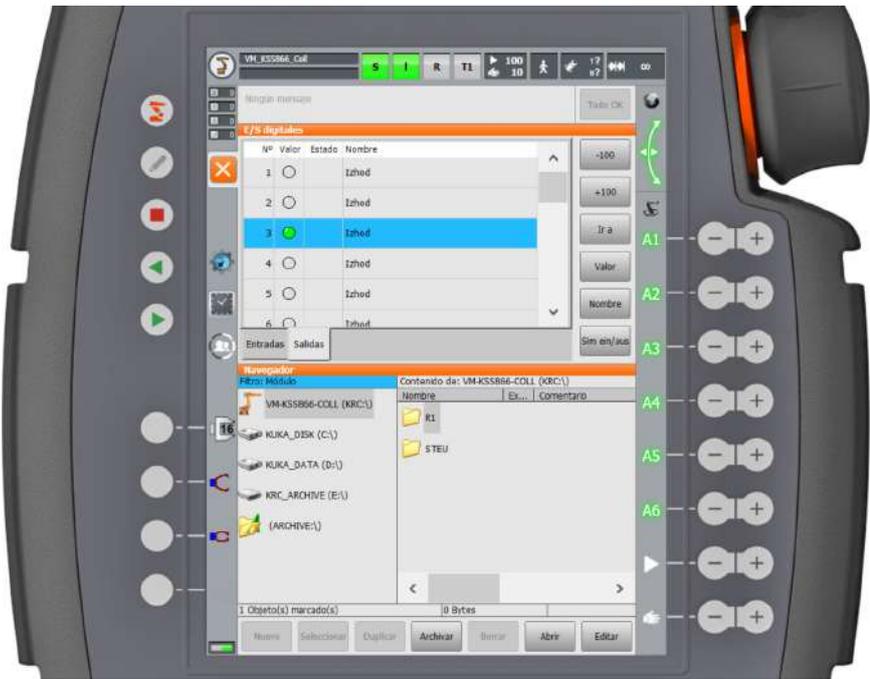
19.3 ¿Qué entradas y salidas se pueden indicar?

1. A través del **Ruta del menú: Tecla de robot > Indicación > Entradas/salidas** se pueden abrir las siguientes vistas:



Entradas y salidas

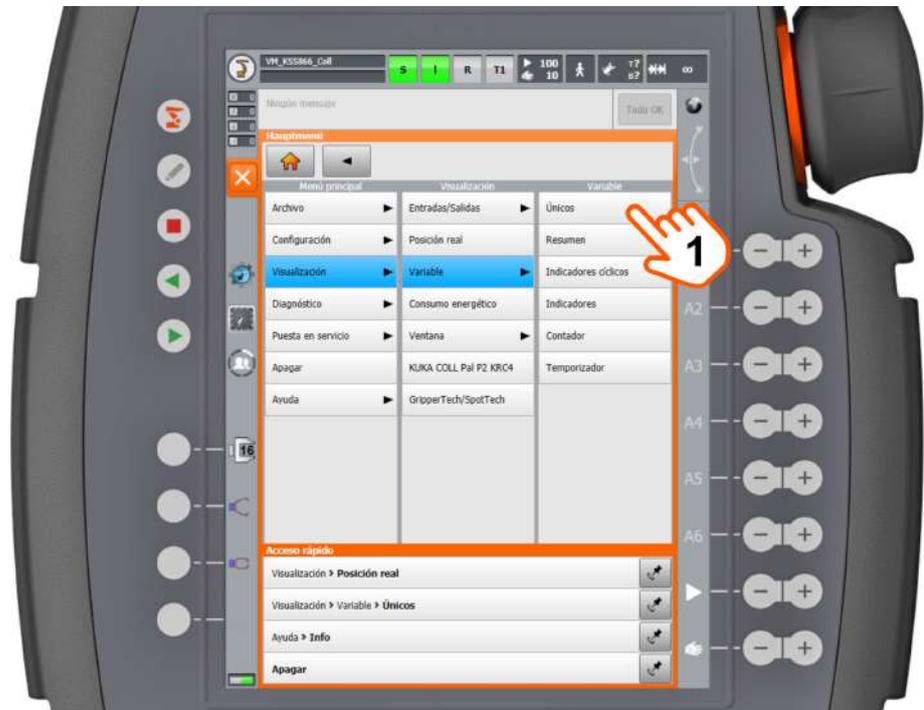
- **Entradas y salidas digitales**
 - **Entradas y salidas analógicas**
2. Mediante LEDs se indica el estado de conmutación actual de la entrada o salida.
Mediante los botones se pueden ajustar salidas o modificar nombres.



19.4 ¿Qué variables se pueden visualizar?

Descripción

Los valores de las variables pueden mostrarse y modificarse a través del smartPAD.



Variables del sistema

- **Marcas cíclicas**
- **Marcas (flags)**
- **Contador**
- **Temporizador**

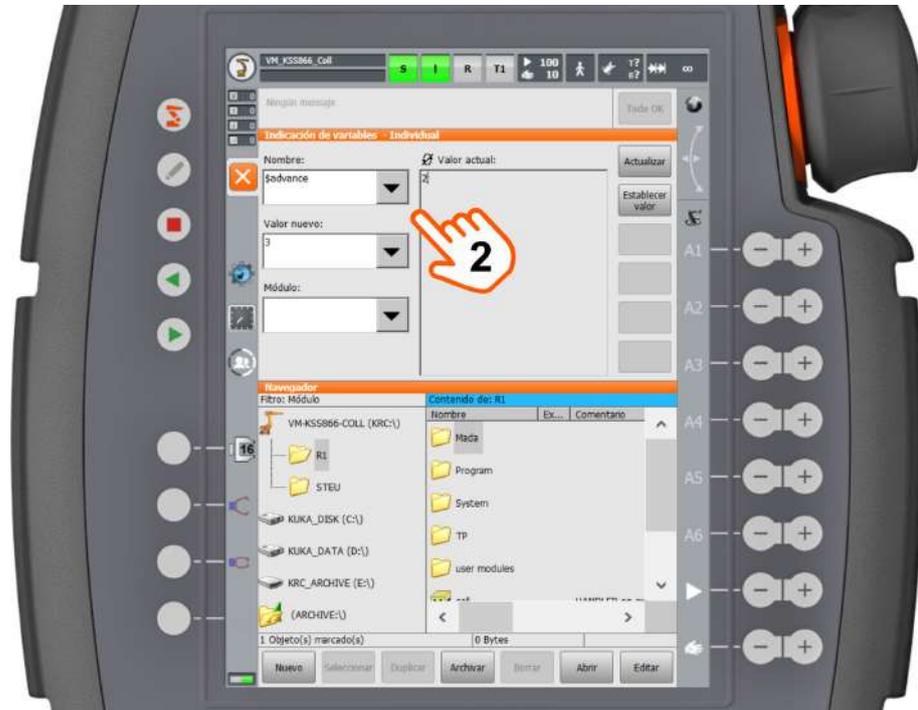
Mostrar y modificar variables individuales

Las variables del sistema pueden mostrarse y modificarse en cualquier momento (si no están protegidas contra escritura).

Variables de tiempo de ejecución solo durante un programa activo

AVISO

Se requieren derechos de usuario superiores.



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Mostrar variables y estados de conmutación en el smartPAD

19.5 Ejercicio: Ampliar programas con una consulta de la estación

¡Ahora es su turno!

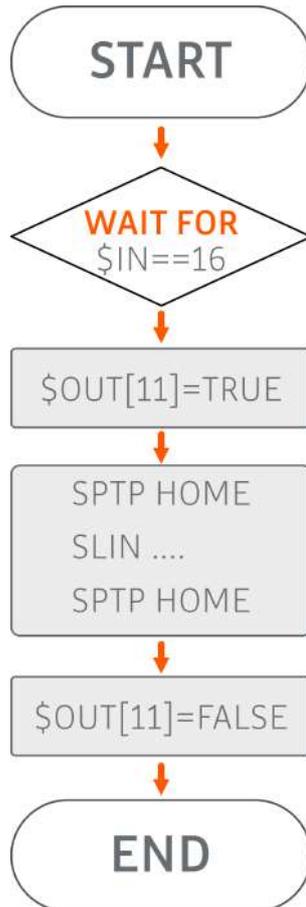


Fig. 19-1: PAP para la liberación de la estación

1. Su programa: _____ debe iniciarse después de que se haya realizado la liberación de movimiento a través de la entrada digital [16].
2. Antes de entrar en la zona de trabajo, la salida digital [11] debe ajustarse en TRUE.
3. Después de salir de la zona de trabajo, la salida digital [11] debe volver a ajustarse en FALSE.

Formularios inline utilizados



Fig. 19-2: Selección Lógica

1. **WAIT FOR**

WAIT FOR (IN 16)

2. **OUT**

OUT in progress State TRUE

20 Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

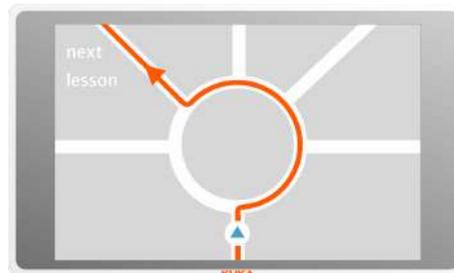
20.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y Automático.
- Está capacitado para crear programas propios con formularios inline y las trayectorias de los movimientos SPTP, SLIN, SCIRC, SPLINE y bloque SPLINE

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

- Aprenderá a aplicar funciones de conmutación OUT y PULSE sencillas.
- Aprenderá a aplicar instrucciones de espera sencillas.
- Aprenderá a conmutar a una trayectoria Spline.
- Programará una "parada condicionada" en un bloque Spline o un formulario inline.
- Programará una zona de desplazamiento constante en un bloque Spline.

20.2 Descripción de la situación

La unidad de control dispone de interfaces de bus de campo para activar, p. ej. una herramienta en una posición de programa definida o bien para esperar a una señal.

Mediante esta interfaz tiene lugar la **comunicación** con los **periféricos** a través de los que se utilizan **entradas y salidas digitales y analógicas**.



Fig. 20-1: Entradas y salidas

Descripción de conceptos

Término	Explicación	Ejemplo
Comunicación	Intercambio de señales a través de una interfaz	Consulta de un estado (garra abierta/cerrada)
Periféricos	"Entorno"	Herramienta (p. ej., garra, pinzas de soldadura, etc.), sensores, sistemas de transporte del material, etc.
Digital	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)
Análogica	Representación de una dimensión física	Medición de temperatura
Entradas	Las señales <i>procedentes</i> de la interfaz de bus de campo para el control	Señal de sensor: Garra abierta/garra cerrada

Término	Explicación	Ejemplo
Salidas	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 conlleva a que cierre una mordaza.

20.3 Programación lógica simple

Descripción

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

- **OUT**
Conectar una salida a un punto concreto en un programa
- **WAIT FOR**
Función de espera dependiente de la señal. Aquí la unidad de control espera una señal:
 - Entrada **IN**
 - Salida **OUT**
 - Señal horaria **TIMER**
 - Dirección de almacenamiento interno de la unidad de control (marcador/memoria de 1 bit) **FLAG** o **CYCFLAG** (cuando se valora de forma cíclica continua)
- **WAIT**
Función de espera por tiempo. En este lugar del programa la unidad de control espera un tiempo establecido.



Los comandos lógicos descritos aquí son parte esencial también en versiones de software más antiguas. El uso es siempre un compromiso entre un punto de conexión exacto y una parada del avance. En los proyectos nuevos se recomienda comprobar el uso de comandos de Trigger (SYNOUT, conmutación en la trayectoria) en combinación con bloques SPTP, SLIN, SCIRC y SPLINE. Aquí se puede garantizar un punto de conexión exacto independiente de una parada del avance.

20.3.1 Programación de funciones de conmutación sencillas

Descripción

Mediante una acción de conmutación puede enviarse una señal digital a los periféricos.

Para ello se utiliza un número de salida que se haya definido antes según la interfaz.

La señal se configura estática, es decir, se mantiene hasta que la salida se ocupe con otro valor.

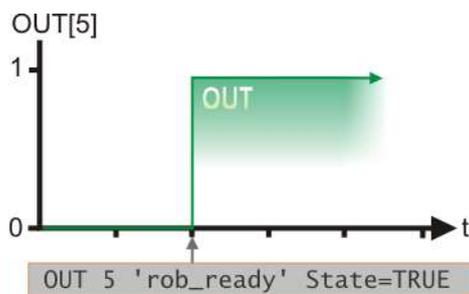


Fig. 20-2: Conectar estática

Formulario inline OUT

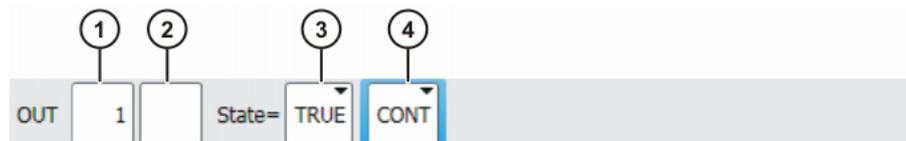


Fig. 20-3: Formulario inline OUT

Pos.	Descripción
1	Número de la salida <ul style="list-style-type: none"> • 1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra. Solo para el grupo de usuario Experto: Pulsando en Texto largo puede introducirse un nombre. Se puede elegir cualquier nombre.
3	Estado en el que la salida se conecta. <ul style="list-style-type: none"> • TRUE • FALSE
4	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance



La instrucción de conexión simple OUT se utiliza para conectar al comienzo de un programa salidas de bus de campo en un estado de salida definido.

20.3.2 Conocer el puntero de movimiento de avance

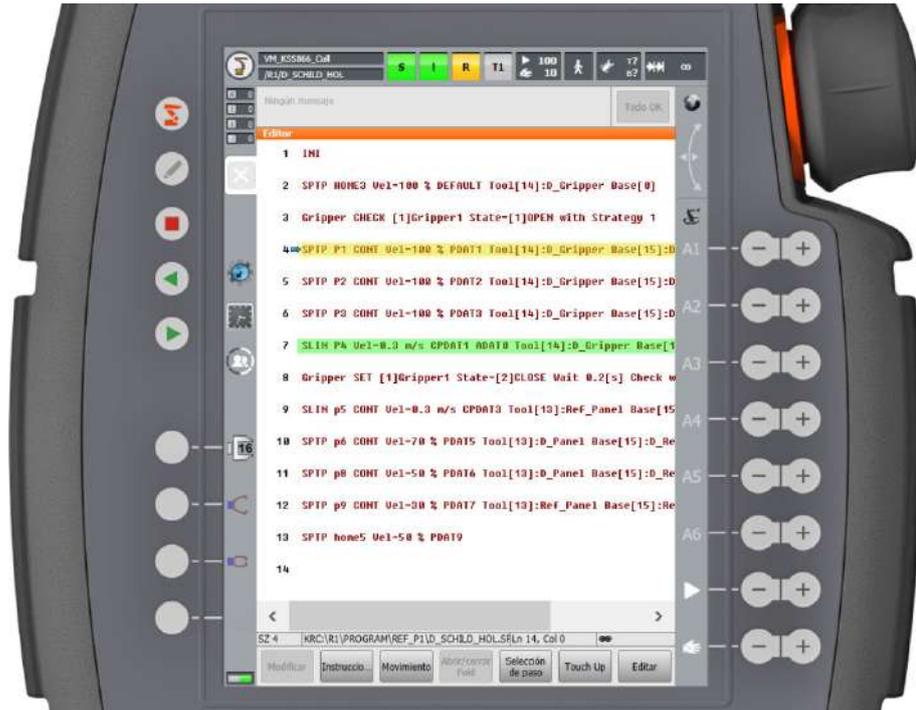
¿Qué es el procesamiento en avance?

La computarización en movimiento de avance lee (de manera invisible para el usuario) los conjuntos de movimientos en avance para poder permitir el control de la planificación de la trayectoria en instrucciones de aproximación.

Con el avance no solo se ejecutan datos de movimiento, sino también instrucciones aritméticas y de control de los periféricos.

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.



amarillo Puntero de ejecución principal
verde Puntero de movimiento de avance

20.3.3 Comportamiento 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 procesamiento en avance** y se produce una parada exacta en el punto antes del comando de conmutación. Tras establecer la salida se continúa con el movimiento.

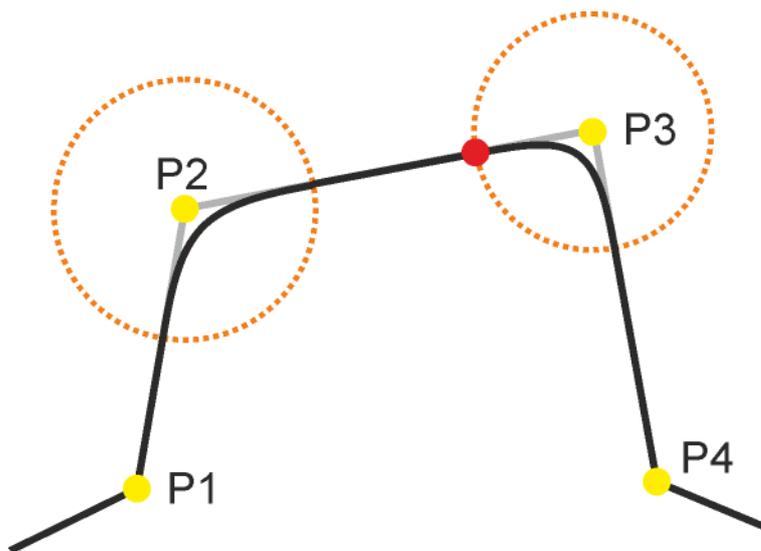


Fig. 20-4: Movimiento ejemplar con conmutación con parada del procesamiento en avance

```
SLIN P1 Vel=0.2 m/s CPDAT1 Tool[1] Base[1]
SLIN P2 CONT Vel=0.2 m/s CPDAT2 Tool[1] Base[1]
```

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

```
SLIN P3 CONT Vel=0.2 m/s CPDAT3 Tool[1] Base[1]
OUT 5 'rob_ready' State=TRUE
SLIN P4 Vel=0.2 m/s CPDAT4 Tool[1] Base[1]
```

con CONT

El ajuste de la entrada CONT provoca que el puntero de movimiento de avance no se detenga (no se provoca ninguna parada del procesamiento en 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**.

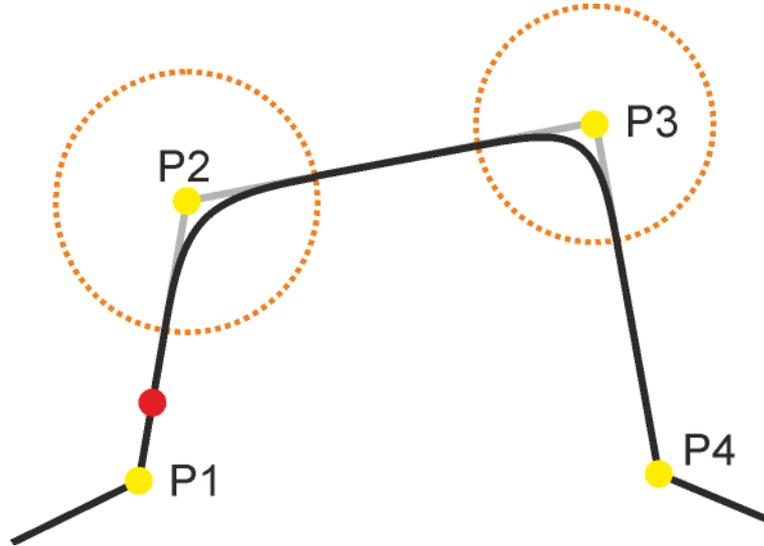


Fig. 20-5: Movimiento ejemplar con conmutación en el avance

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

20.3.4 Conocer las funciones de conmutación pulsadas "PULSE"

Descripción

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.

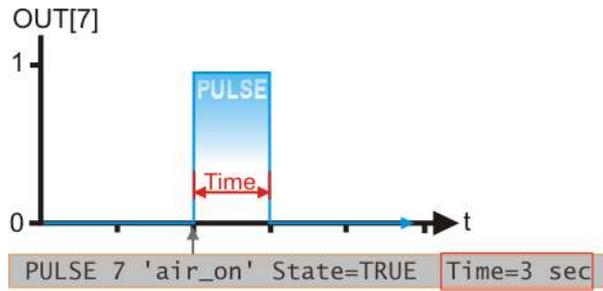


Fig. 20-6: Nivel pulsado

Formulario inline "PULSE"

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

Fig. 20-7: Formulario inline PULSE

Pos.	Descripción
1	Número de la salida <ul style="list-style-type: none"> • 1 ... 4096
2	Si para la salida existe ya un nombre, éste se muestra. Solo para el grupo de usuario Experto: Pulsando en Texto largo puede introducirse un nombre. Se puede elegir cualquier nombre.
3	Estado en el que la salida se conecta <ul style="list-style-type: none"> • TRUE: Nivel "High" • FALSE: Nivel "Low"
4	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance
5	Duración del pulso <ul style="list-style-type: none"> • 0,10 ... 3,00 s

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Programar funciones de conmutación clásicas mediante formularios inline

Conocer la relación entre el puntero de movimiento de avance y el puntero de ejecución principal

20.3.5 Conocer la función de espera "WAIT"

Las funciones de espera dependientes del tiempo de un programa de movimiento se programan de forma muy sencilla mediante formularios inline. Con el formulario inline **WAIT** se detiene el movimiento de robot para el tiempo programado. WAIT provoca siempre una parada del procesamiento en avance.

Con **WAIT** se detiene el movimiento de robot durante el tiempo programado. WAIT provoca siempre una parada del procesamiento en avance.

WAIT Time= 2.5 [sec]

Pos.	Descripción
1	Tiempo de espera • ≥ 0 s

El propio formulario inline insertado es un pliegue y puede abrirse.

En verde se visualiza la correspondiente instrucción KRL.

```
7 → WAIT Time= 2.5 sec
8 WAIT SEC 2.5
```

Función de espera dependiente del tiempo con un tiempo fijo

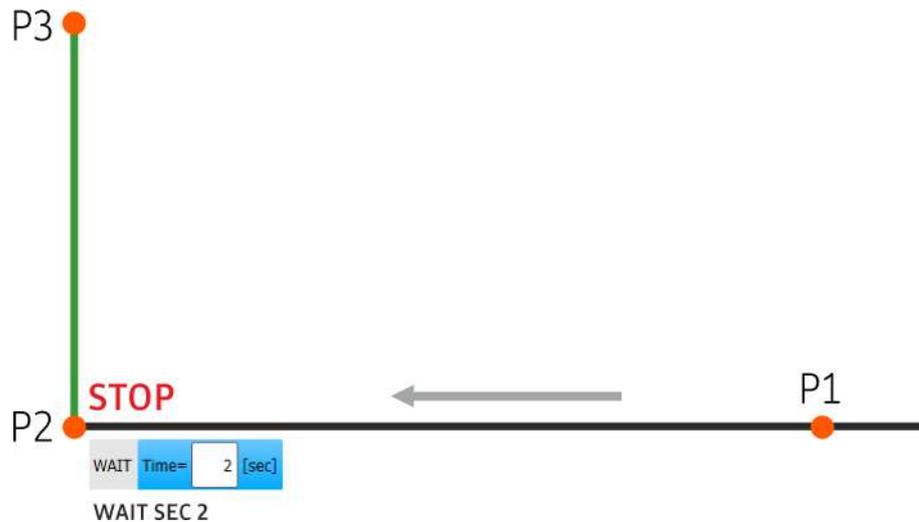
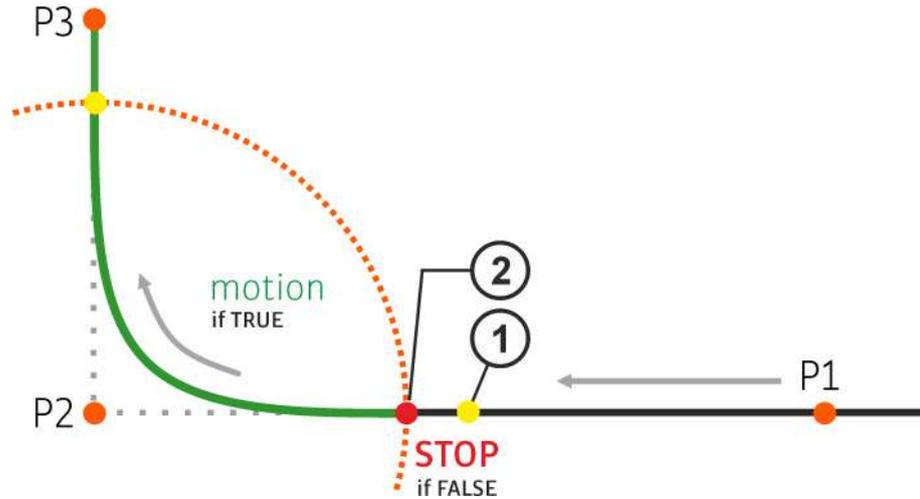


Fig. 20-8: Ejemplo, función de espera dependiente del tiempo

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

```
SPTP P1 Vel=100 % PDAT1
SPTP P2 Vel=100 % PDAT2
WAIT TIME= 2.0 [sec]
SPTP P3 Vel=100 % PDAT3
```

20.3.6 Comportamiento de conmutación en el entorno de aproximación



• **WAIT FOR sin CONT**

WAIT FOR (IN 16)

```
SLIN P1 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
WAIT FOR ( IN 16 '' )
SLIN P2 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
SLIN P3 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
```

- Si se programa "WAIT FOR" **sin** CONTINUE, el robot siempre se detiene con una parada exacta al comienzo del entorno de aproximación (2) del punto P2.
- Si el estado del \$IN[16] (1) antes de alcanzar el entorno de aproximación (2) es **TRUE**, el robot reanudará su desarrollo del movimiento inmediatamente después de la parada exacta.
- Si el estado del \$IN[16] (1) antes de alcanzar el entorno de aproximación (2) es **FALSE**, el robot esperará en el límite de aproximación hasta que el estado cambie a TRUE y, a continuación, reanudará su desarrollo del movimiento.

• **WAIT FOR con CONT**

WAIT FOR (IN 16) CONT

CONT

```
SLIN P1 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
WAIT FOR ( IN 16 '' ) CONT
SLIN P2 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
SLIN P3 CONT Vel=2 m/s CPDAT1 Tool[1] Base[1]
```

- Si el estado del \$IN[16] (1) antes de alcanzar el entorno de aproximación (2) es **TRUE**, el robot se desplazará al entorno de aproximación sin parada.
- Si el estado del \$IN[16] (1) antes de alcanzar el entorno de aproximación (2) es **FALSE**, el robot esperará en el límite de aproxi-

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

mación hasta que el estado cambie a TRUE y, a continuación, reanuda su desarrollo del movimiento.

20.3.7 Programar una función de dependientes dependiente de la señal "WAIT FOR" con formularios inline

Función de espera "WAIT FOR"

En caso necesario, pueden combinarse de forma lógica varias señales (máximo 12).

Si se agrega una combinación, en el formulario inline aparecen campos para las señales adicionales y para otras combinaciones.

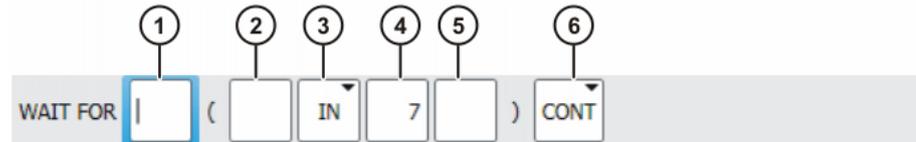


Fig. 20-9: Formulario inline WAIT FOR

Pos.	Descripción
1	<p>Agregar la combinación externa. El operador se ubica entre las expresiones colocadas entre paréntesis.</p> <ul style="list-style-type: none"> • AND • OR • EXOR <p>Agregar NOT.</p> <ul style="list-style-type: none"> • NOT • [vacío] <p>Agregar el operador deseado utilizando el correspondiente botón.</p>
2	<p>Agregar la combinación interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <ul style="list-style-type: none"> • AND • OR • EXOR <p>Agregar NOT.</p> <ul style="list-style-type: none"> • NOT • [vacío] <p>Agregar el operador deseado utilizando el correspondiente botón.</p>
3	<p>Señal que se está esperando</p> <ul style="list-style-type: none"> • IN • OUT • CYCFLAG • TIMER • FLAG
4	<p>Número de la señal</p> <ul style="list-style-type: none"> • 1 ... 4096

Pos.	Descripción
5	Si la señal ya tiene nombre, este se muestra. Solo para el grupo de usuario Experto: Pulsando en Texto largo puede introducirse un nombre. Se puede elegir cualquier nombre.
6	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance

Combinaciones lógicas

En la utilización de funciones de espera dependientes de señales se pueden utilizar también combinaciones lógicas.

Las combinaciones lógicas permiten combinar las consultas de diversas señales o estados.

- Se puede crear dependencias.
- Se pueden excluir 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. 20-10: 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.



Las funciones de espera dependientes de señales pueden programarse con y sin su procesamiento en movimiento de avance.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

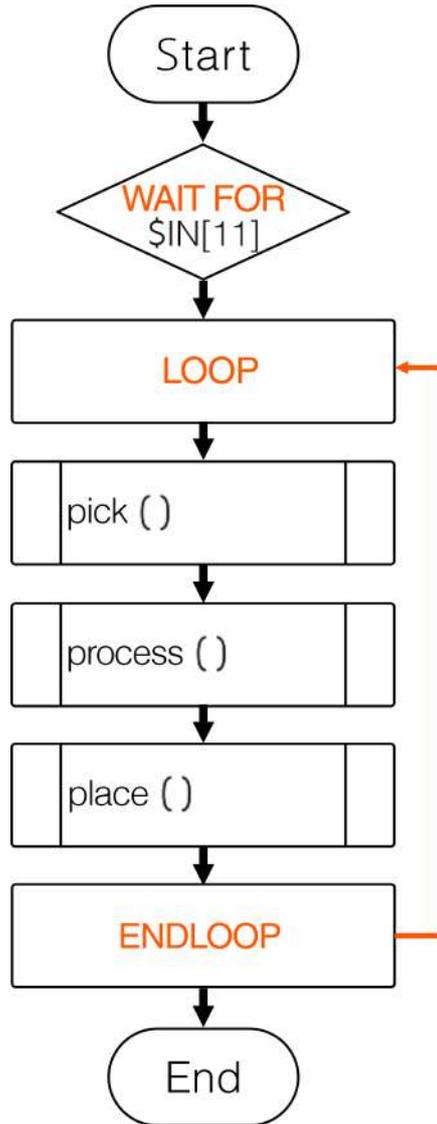
Programar funciones de espera clásicas a través de formularios inline

20.3.8 Ejercicio: Programar "WAIT-FOR"

Enunciado

Antes de ejecutar el bucle sinfin se debe consultar si la estación del robot está libre.

1. Su programa principal no debe ejecutarse hasta que la entrada digital [11] sea TRUE. Esta comprobación debe realizarse una vez antes del flujo de programa.
2. Pruebe el programa en los distintos modos de servicio T1, T2 y Automático.



20.3.9 Preguntas: Equipar un movimiento de trayectoria con lógica

Lo que se debe saber tras el ejercicio:



¿Cuál es la diferencia entre las instrucciones `OUT` y `OUT CONT`? ¿Qué debe tenerse en cuenta?



Fig. 20-11: Campo de respuesta

¿En qué se diferencian las instrucciones `PULSE` y `OUT`?

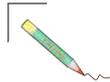


Fig. 20-12: Campo de respuesta

¿Cuándo se utilizan las instrucciones `SYN OUT`?

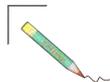


Fig. 20-13: Campo de respuesta

¿Cuáles son los peligros que se presentan al utilizar la instrucción `WAIT FOR` con una instrucción `CONT`?

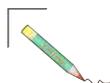


Fig. 20-14: Campo de respuesta

20.4 Programación de lógica con SPLINE y pasos individuales SPLINE

Descripción

Al programar pasos individuales Spline (SLIN, SCIRC, SPTP) y bloques Spline, se puede integrar la lógica además del movimiento. La lógica se puede programar a través de KRL o cómodamente a través de formularios inline.



Fig. 20-15: Lógica SPLINE en el formulario inline

Siempre disponible (ADAT):

- Trigger (instrucción de conmutación en la trayectoria)
- Parada condicionada

Solo en el bloque SPLINE

- Zona de desplazamiento constante
- Activador como formulario inline

Solo en el bloque SPLINE y solo como comando KRL

- Bloque de tiempo

Solo fuera del bloque SPLINE

- Parada condicionada como formulario inline (Spline Stop Condition)

20.4.1 Programar el activador Spline

Descripción

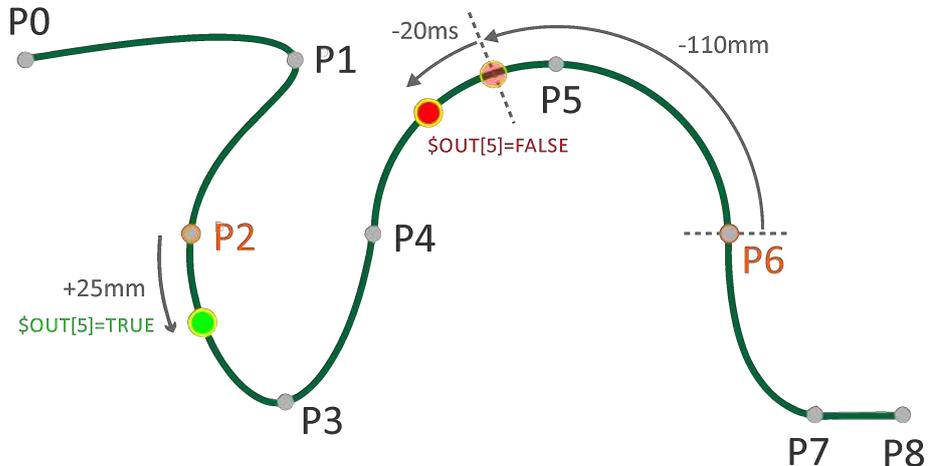


Fig. 20-16: Activador con SPLINE

El ejemplo muestra una trayectoria que se ha programado mediante el **bloque Spline**. La **función del activador** también se puede transferir a un movimiento de trayectoria, el cual se compone de **pasos individuales**.

El activador lanza una instrucción definida por el usuario, en el ejemplo la salida digital 5. La unidad de control del robot ejecuta la instrucción de forma paralela al movimiento del robot. El activador puede referirse opcionalmente al punto de inicio o de destino del movimiento. La instrucción se activa directamente en el punto de referencia o se desplaza en el espacio y/o temporalmente.



Si el activador se utiliza en un bloque spline, no podrá encontrarse entre el último segmento y ENDSPLINE.

Posibilidades de la programación

- En el formulario inline, ventana de opciones lógica de trigger
- Formulario inline SPLINE-Trigger
- Programación mediante comando KRL `TRIGGER WHEN PATH`

20.4.1.1 Programar el Trigger SPLINE en el bloque SPLINE

Variante 1: Activar el trigger en el punto

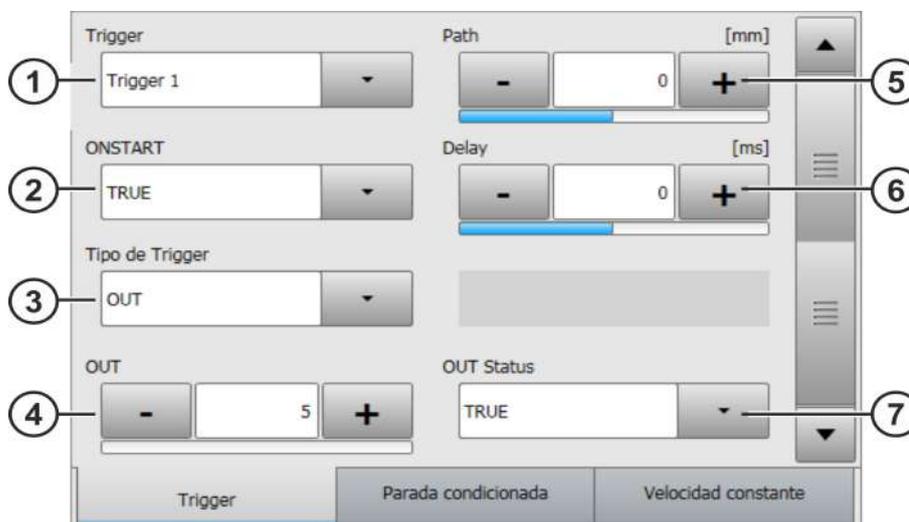


Fig. 20-17: Definir el activador Spline

- **Activador (1)**
El sistema crea un nuevo activador con el nombre Activador 1. Si ya hay otros activadores definidos, esta ventana permite acceder a ellos y ajustarlos.
- **ONSTART (2)**
Si el parámetro ONSTART se establece en **TRUE**, el punto de referencia para la activación del activador se refiere al punto de inicio. Este punto puede desplazarse de forma adicional mediante los parámetros Path y Delay. En nuestro ejemplo la salida digital 5.
- **Tipo de activador (3)**

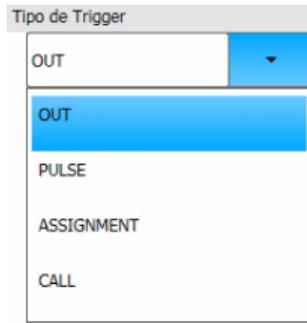


Fig. 20-18: Tipo de activador

Define qué tipo de activador se debe utilizar. Aquí es necesario parte de experiencia adicional en KRL.

En nuestro ejemplo, seleccionar OUT para conectar una salida digital.



En función del tipo de activador seleccionado PULSE, ASSIGNMENT o CALL, cambian las ventanas de entrada y selección en los campos 3, 4 y 7.
En la documentación del software se puede encontrar información más detallada.

- **OUT (4)**
Selección de la salida digital que conecta el activador.
- **PATH (5)**
Dado que el activador se conecta directamente en el punto y este está en TRUE, el desplazamiento en el espacio está inactivo.
- **Delay (6)**
En referencia al punto, el punto puede desplazarse en el tiempo dependiendo de la velocidad a la que se circule. Resulta útil, por ejemplo, en aplicaciones de adhesivo
- **Estado OUT (7)**
Aquí se puede definir el estado de conexión deseado de la salida digital. En este ejemplo, este se conecta mediante la expresión booleana TRUE.

Variante 2: Hacer que el sistema registre el valor de Trigger Path (distancia)

En el ejemplo: Desplazamiento en el espacio y el tiempo con respecto al punto 5.



Fig. 20-19: Desplazamiento local y temporal

- Con ayuda de las teclas del programa, mover el robot hacia delante o hacia atrás sobre la trayectoria Spline hasta que se alcance el "punto Trigger".
- Con el botón *Acciones Spline* > *Registrar ruta Activador Path* aceptar la posición actual en el campo Path.

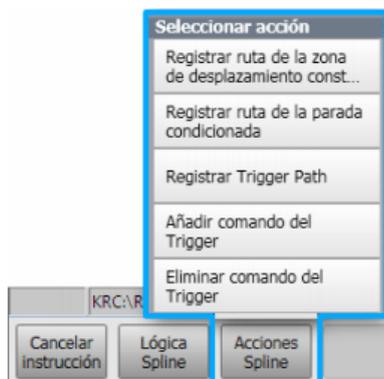


Fig. 20-20: Acciones Spline



Si se abandona la trayectoria desplazando manualmente el robot no se puede registrar con Path. Aquí es necesario volver a seleccionar un conjunto.



Con la programación por aprendizaje, ONSTART se configura siempre en FALSE y se introduce el desplazamiento en el espacio con respecto al punto de destino (negativo).



Con ONSTART en TRUE, en principio, es factible un desplazamiento con respecto al punto de inicio. No obstante, el offset (positivo) solo se puede hacer con una entrada numérica.

- En caso necesario introducir el desplazamiento en el tiempo numéricamente en el campo Delay [ms].

20.4.1.2 Programar el Trigger SPLINE en el formulario inline

Formulario inline

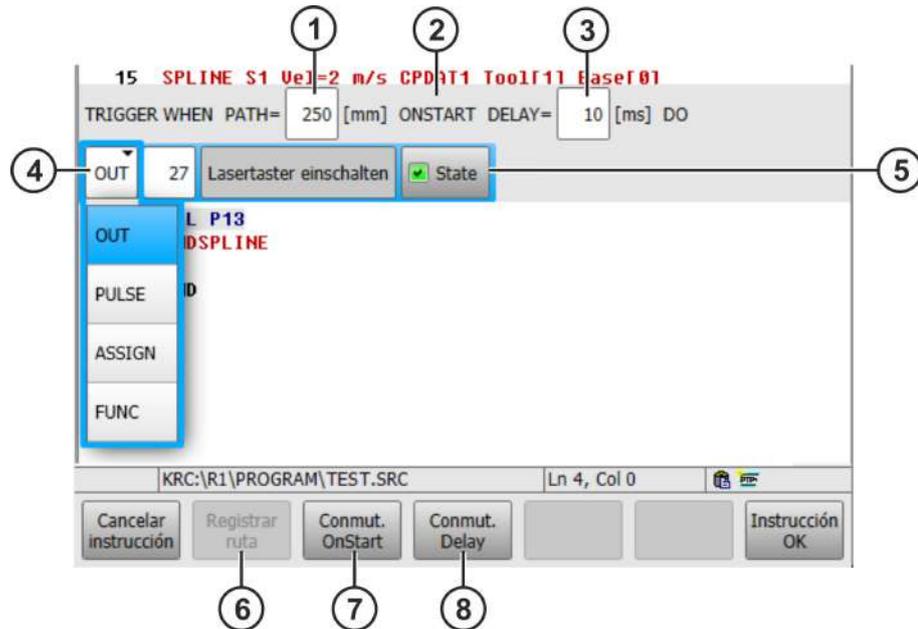


Fig. 20-21: Formulario inline, Spline Trigger

Pos.	Denominación	Descripción
1	Registrar ruta	El desplazamiento en el espacio (Path) del punto de conexión sobre la trayectoria se puede "programar por aprendizaje" con Registrar ruta.
2	ONSTART	Punto al que se refiere el activador SPLINE <ul style="list-style-type: none"> • Con ONSTART: último punto antes del Trigger SPLINE • Sin ONSTART: Punto después del Trigger SPLINE ONSTART puede ajustarse o eliminarse con el botón Conmut. OnStart .
3	DELAY	Desplazamiento temporal del Trigger SPLINE
4	OUT	Reacción de activación del Trigger <ul style="list-style-type: none"> • OUT (establece la salida) • PULSE (establece la salida de pulsos) • ASSIGN (asignación del Trigger) • FUNC (activación de la función del Trigger)

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

Pos.	Denominación	Descripción
5	State	Indicación del estado que debe adoptar la salida TRUE =casilla verde en el botón FALSE = botón gris
6	Registrar ruta	Cálculo automático del valor PATH al recorrer el punto del activador sobre la trayectoria
7	Conmut. OnStart	Ver pos. 2
8	Conm. Delay	Ver pos. 3

Ejemplo: Acceso a un subprograma.



Fig. 20-22: Ejemplo de Interrupt

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro **Programación de robots 1**
KUKA System Software 8.6
Referencia de formación

Capítulo/apartado Programación de la lógica en el bloque SPLINE y en pasos individuales SPLINE

Podrá encontrar más información en la documentación del cliente.



Libro **KUKA System Software 8.6**
Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda Documentación > Instrucciones de manejo y programación

Capítulo/apartado Programación con formularios inline > Programar movimientos Spline > Programar Trigger para SPLINE

Podrá encontrar más información en la documentación del cliente.



Libro

KUKA System Software 8.6

Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

Programación KRL > Acciones de conmutación referentes a la trayectoria (=Trigger)

20.4.1.3 Ejercicio (variante 1): Equipar el movimiento Spline con lógica, contorno 1

¡Ahora es su turno!



Enunciado

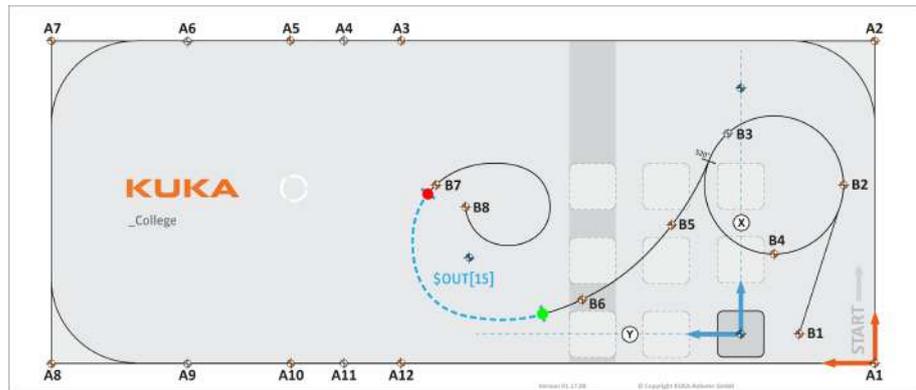


Fig. 20-23: Bloque SPLINE, conmutar

1. Duplique su programa: _____ y denomínelo **progr-**ma: _____.
2. Utilizar el formulario inline TRIGGER WHEN PATH o las funciones de disparo dentro del bloque Spline.
3. Activar la salida 15 en la zona azul.
4. Comprobar el programa en los modos de servicio T1, T2 y automático.

20.4.1.4 Ejercicio (variante 2): Equipar el movimiento SPLINE con lógica, contorno 2

Enunciado

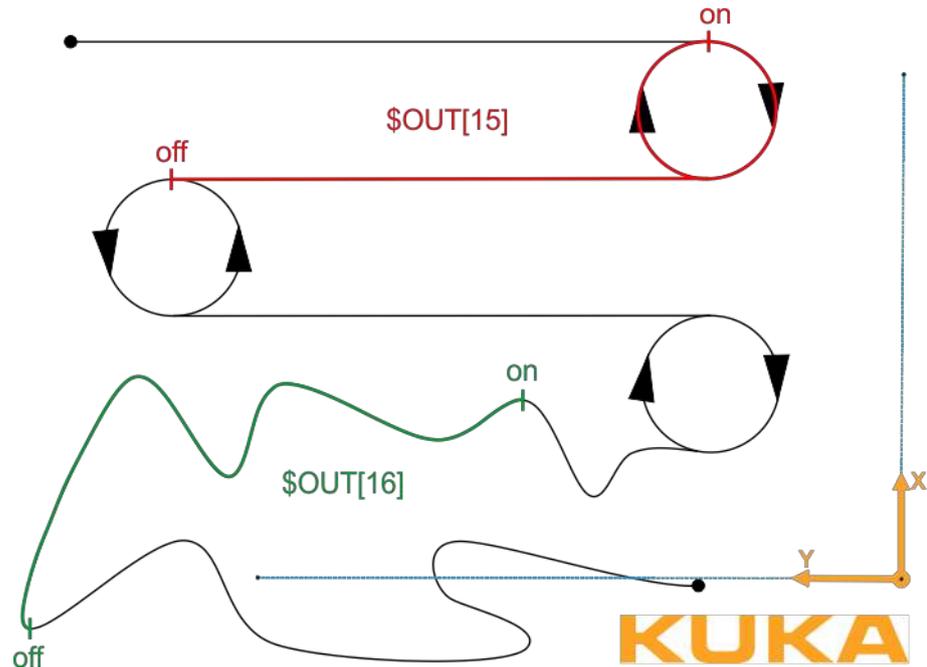


Fig. 20-24: Lógica con Spline

1. Duplique su programa: _____ y denomínelo programa: _____.
2. Utilice aquí el formulario inline TRIGGER WHEN PATH o las funciones de disparo dentro del bloque Spline.
3. Activar la salida 15 en la zona roja.
4. Activar la salida 16 en la zona verde.
5. Comprobar el programa en los modos de servicio T1, T2 y automático.

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

20.4.1.5 Ejercicio (variante 3): Equipar el desarrollo del movimiento Spline con lógica, componente de soldadura

Enunciado

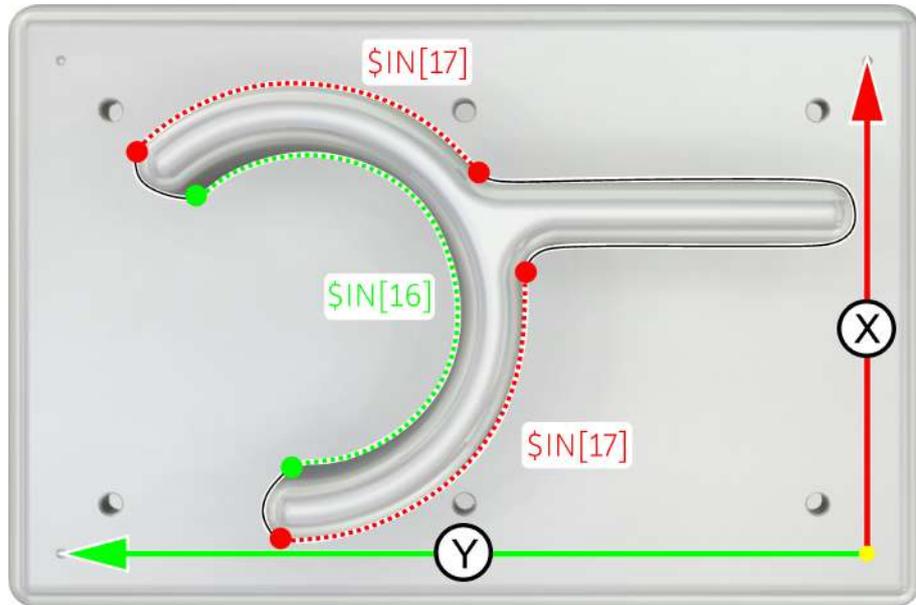


Fig. 20-25: Componente de soldadura, conmutación de trayectoria

1. Duplique su programa para el componente de soldadura y renómbrelo.
2. Utilice aquí el formulario inline TRIGGER WHEN PATH o las funciones de disparo dentro del bloque Spline.
3. Active la salida 16 en la zona verde.
4. Activar la salida 17 en la zona roja.
5. Compruebe el programa en los modos de servicio T1, T2 y automático.

20.4.1.6 Ejercicio (variante 4): Equipar movimiento de trayectoria con lógica, contorno de pegado

¡Ahora es su turno!



Enunciado

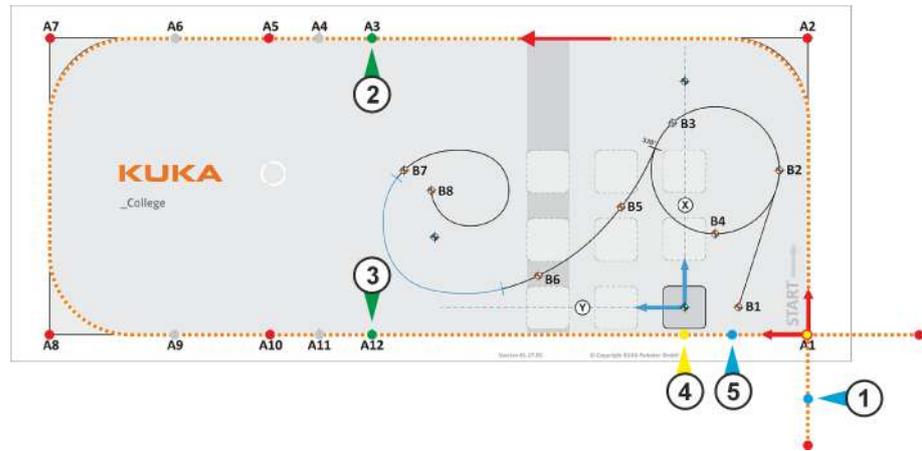


Fig. 20-26: Acciones de conmutación, desplazamiento de trayectoria

1. Cree de su programa: _____ un duplicado con el nombre programa: _____
2. Amplíe el programa por la funcionalidad lógica siguiente:
 - Antes de abandonar la posición HOME, el PLC debe producir una señal de habilitación (entrada 11).
 - 0,2 segundos antes de que la boquilla de aplicación de pegamento alcance la mesa, debe activarse la boquilla (1) (salida 13).
 - En el punto de transición del plano a la curvatura del componente debe conmutarse un indicador luminoso (2) que debe apagarse de nuevo en el punto de transición de la curvatura al plano (3) (salida 12).
 - 0,2 segundos antes de abandonar la pieza (5), la boquilla de aplicación de pegamento debe desactivarse de nuevo (salida 13).
 - 50 mm antes de terminar el tratamiento de la mesa (4), el PLC debe recibir un aviso de tarea finalizada. La señal (salida 11) para el PLC debe estar presente durante 2 segundos.
3. Comprobar el programa como indique el formador.

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

20.4.1.7 Preguntas: Equipar el movimiento SPLINE con lógica

Lo que se debe saber tras el ejercicio:



¿Qué hay que tener en cuenta al utilizar la función WAIT?



Fig. 20-27: Campo de respuesta

¿Qué se consigue al seleccionar CONT en el formulario inline WAITFOR?



Fig. 20-28: Campo de respuesta

¿Cuál es la diferencia entre OUT y PULSE?



Fig. 20-29: Campo de respuesta

¿Qué entendemos por Trigger?

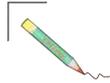


Fig. 20-30: Campo de respuesta

20.4.2 Programar parada condicionada

Descripción

La "parada condicionada" permite al usuario definir un lugar en la trayectoria donde se para el robot si se cumple una condición determinada. El lugar se denomina "punto de parada". En cuanto deja de cumplirse la condición, el robot reanuda la marcha. Durante el tiempo de ejecución, la unidad de control del robot calcula el punto donde, posteriormente, se debe frenar para que se pueda producir la detención en el punto de parada. A partir de ese punto (= "punto de frenado"), la unidad de control del robot evalúa si se ha cumplido o no la condición. En cuanto se cumple la condición en el punto de frenado, el robot frena para detenerse en el punto de parada. En caso de que deje de cumplirse la condición antes de que se alcance el punto de parada, el robot vuelve a acelerar y no se para. Si la condición para la parada solo se cumple cuando el robot ya ha pasado por el punto de frenado, es demasiado tarde para detenerse con una rampa de frenado normal en un punto de parada.

- En este caso, el robot se para con una PARADA DE EMERGENCIA sobre la trayectoria y se detiene en un punto no previsto.
- Si el robot se detiene tras el punto de parada con una PARADA DE EMERGENCIA, el programa solo puede reanudarse cuando ya no se dé la condición.

Si el robot se detiene antes del punto de parada con la PARADA DE EMERGENCIA sobre la trayectoria, pasa lo siguiente cuando se reanuda el programa:

- Cuando **ya no se cumple** la condición de parada:
El robot se sigue desplazando.
- Cuando **aún se cumple** la condición de parada:
el robot continúa hasta el punto de parada y se queda parado allí.

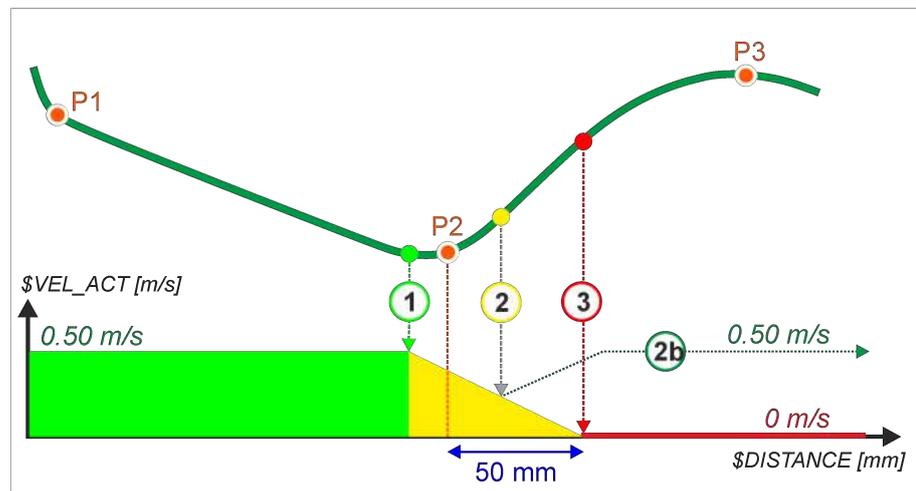


Fig. 20-31: Spline con parada condicionada

Pos.	Explicación
1	Punto de frenado calculado en función de la carrera y la velocidad
2	Durante el proceso de frenado puede producirse un cambio de estado (la condición ya no se cumple). El robot vuelve a acelerar a su velocidad programada.

Pos.	Explicación
2 b	Se ha alcanzado la velocidad programada tras la reacceleración.
3	Punto de parada predefinido (mediante formulario inline)



En principio se pueden programar tantas paradas condicionadas como se desee. No obstante, no deben solaparse más de 10 trayectos "Punto de frenado > punto de parada".
 Durante un proceso de frenado, la unidad de control del robot muestra el siguiente mensaje en T1/T2: *Parada condicionada activa (línea {Número de línea})*.

Posibilidades de la programación

- **En** el bloque Spline o en el paso individual Spline:
 En el formulario inline, ventana de opciones lógica de parada condicionada
- **Antes** de un bloque Spline:
 En el formulario inline Spline Conditional Stop

20.4.2.1 Programar una parada condicionada en el bloque SPLINE

Conmutación con entrada digital

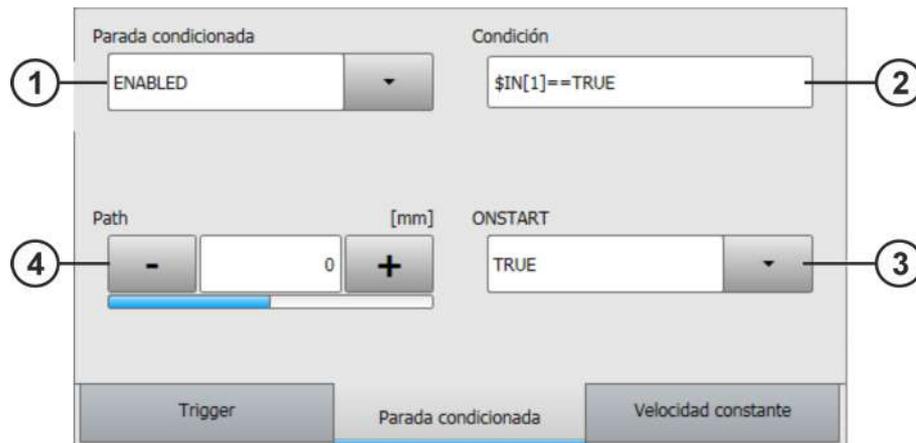


Fig. 20-32: Programar una parada condicionada

Pos.	Parámetro	Valor	Descripción
1	Parada condicionada	ENABLED	La parada condicionada está activada con respecto a este punto
2	Condición	KRL	Condición que se consulta para activar la parada condicionada al alcanzar este punto de movimiento o para seguir desplazándose sin parada. El manejo se programa en KRL.

Pos.	Parámetro	Valor	Descripción
3	ONSTART	TRUE	Start hace referencia al punto de inicio Si el punto de inicio es de posicionamiento aproximado, el punto de referencia se genera de la misma manera que durante la aproximación homogénea del activador PATH.
4	Path	0 mm	ningún desplazamiento en el espacio del inicio para la zona de recorrido constante

Variante con desplazamiento local

Desplazamiento en el espacio del inicio para la parada condicionada con respecto al punto 4.

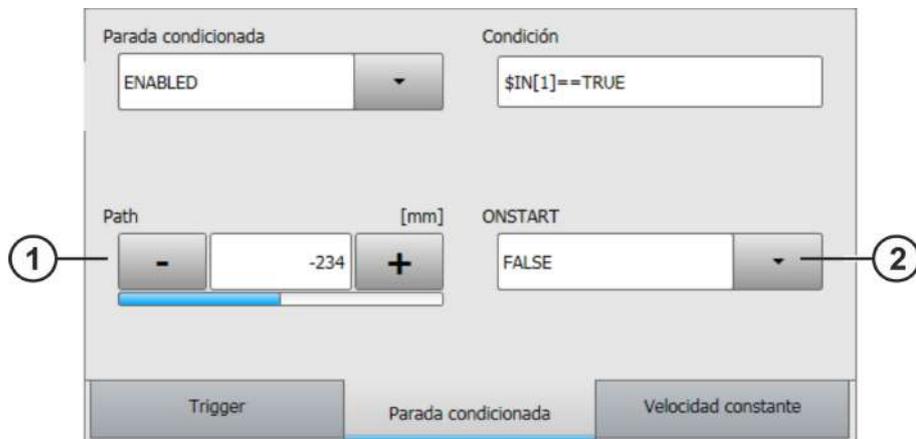


Fig. 20-33: Punto de referencia desplazado

Pos.	Parámetros	Valor	Descripción
1	Path	-234 mm	El punto de referencia para la parada condicionada se determina 234 mm antes de alcanzar el punto de destino.
2	ONSTART	FALSE	Start hace referencia al punto de destino Si el punto de destino es de posicionamiento aproximado, Start o End hacen referencia al inicio del arco de aproximación.

Recepción de los puntos Trigger con acciones Spline

- Utilizar las teclas del programa para mover el robot hacia delante o hacia atrás sobre la trayectoria SPLINE hasta que se alcance el punto de referencia para la parada condicionada.
- Con el botón *Acciones Spline > Registrar ruta parada condicionada* aceptar la posición actual en el campo Path.

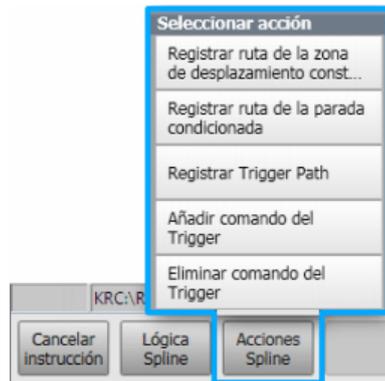


Fig. 20-34: Acciones Spline



Si se abandona la trayectoria desplazando manualmente el robot no se puede registrar con Path. Aquí es necesario volver a seleccionar un conjunto.



Con la programación por aprendizaje, ONSTART se configura siempre en FALSE y se introduce el desplazamiento en el espacio con respecto al punto de destino (negativo).



Si ONSTART está en TRUE, en principio, es factible un desplazamiento con respecto al punto de inicio. No obstante, el offset (positivo) solo se puede hacer con una entrada numérica.

20.4.2.2 Programar una parada condicionada mediante el formulario inline

Formulario inline



Fig. 20-35: Formulario inline Condición de parada para Spline

Pos.	Descripción
1	<p>Punto en el que se basa la parada condicionada</p> <ul style="list-style-type: none"> • con ONSTART: último punto antes del bloque spline • sin ONSTART: último punto del bloque spline <p>En caso de que se efectúe un posicionamiento aproximado del spline, tendrán validez las mismas normas aplicadas para Trigger PATH.</p> <p>Indicación: para más información sobre la aproximación con Trigger PATH, consultar las instrucciones de servicio/programación para integradores de sistemas.</p> <p>ONSTART puede ajustarse o eliminarse con el botón Conmut. OnStart.</p>
2	<p>El punto de parada puede desplazarse en el espacio. Para ello, aquí se debe indicar la distancia deseada con respecto al punto de referencia. Si no se desea ningún desplazamiento en el espacio, introducir "0".</p> <ul style="list-style-type: none"> • Valor positivo: desplazamiento en dirección al final del movimiento • Valor negativo: desplazamiento en dirección al inicio del movimiento <p>El punto de parada no puede desplazarse libremente en el espacio. Tienen validez los mismos límites aplicados para Trigger PATH.</p> <p>El desplazamiento en el espacio también se puede programar por aprendizaje.</p>
3	<p>Condición de parada</p> <p>Son admisibles:</p> <ul style="list-style-type: none"> • una variable global booleana • un nombre de señal • una comparación • un combinación lógica simple: NOT, OR, AND o EXOR
Provoca un mensaje de error:	
$(\$IN[29]==FALSE) \text{ OR } (\$IN[22]==TRUE)$	
Correcto:	
$NOT \$IN[29] \text{ OR } \$IN[22]$	

Botones

En el estado abierto del formulario inline se muestran botones adicionales.

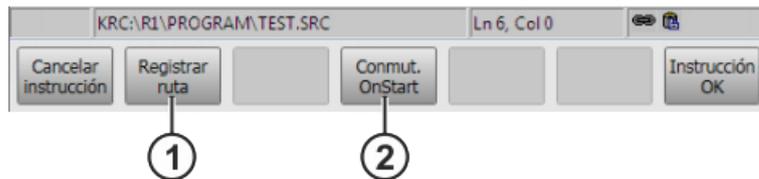


Fig. 20-36: Botones

Pos.	Denominación	Descripción
1	Registrar ruta	El desplazamiento local (path) del punto de conmutación por la trayectoria puede programarse por aprendizaje con "Aceptar path".
2	Conmut. OnStart	Determinación del punto de conmutación en el punto sin desplazamiento local.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro **Programación de robots 1**
 KUKA System Software 8.6
 Referencia de formación

Capítulo/apartado Detener el robot en el bloque SPLINE mediante una condición

Podrá encontrar más información en la documentación del cliente.



Libro **KUKA System Software 8.6**
 Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda Documentación > Instrucciones de manejo y programación

Capítulo/apartado Programación con formularios inline > Programar movimientos Spline > Programar una parada condicionada para Spline

20.4.3 Programación de zona de desplazamiento constante

Descripción

En un bloque spline CP, se puede definir una zona en la que el robot mantiene constante la velocidad programada, siempre y cuando sea posible.

El margen se denomina "Zona de desplazamiento constante".

- Para cada bloque Spline se puede definir 1 zona de desplazamiento constante.
- La zona de desplazamiento constante se define mediante una instrucción de inicio y una instrucción de final.
- La zona no puede superar el bloque spline.
- La zona puede ser tan pequeña como se desee.

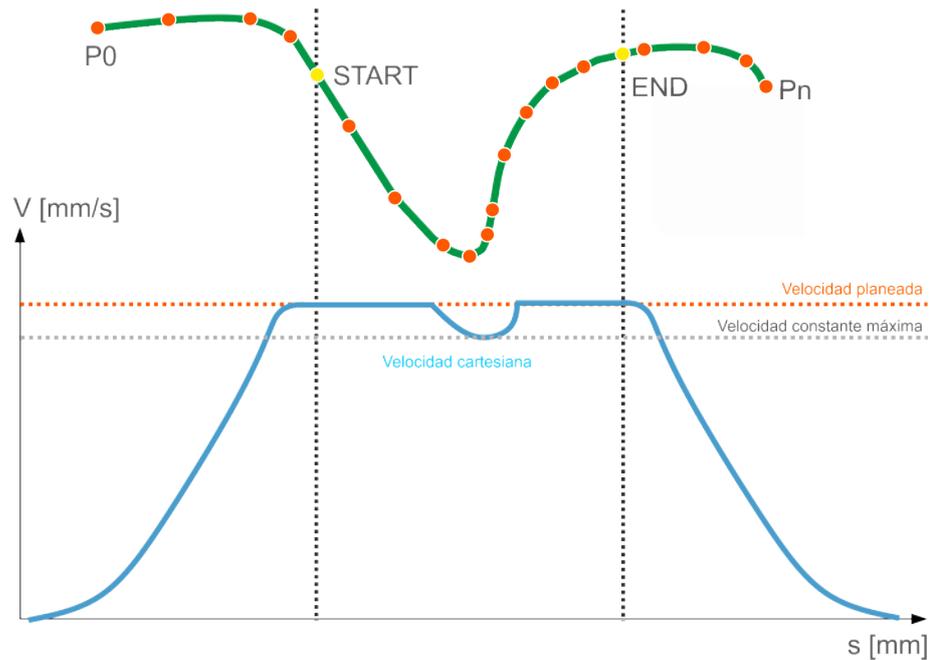


Fig. 20-37: Velocidad constante, principio



En caso de que no sea posible mantener constante la velocidad programada, la unidad de control del robot lo notifica durante la ejecución del programa dentro de los tipos de modos de servicio de test mediante un mensaje.

Zona de desplazamiento constante a través de varios segmentos:

- La zona de desplazamiento constante se puede extender por varios segmentos con diferentes velocidades programadas.
En este caso, las velocidades más reducidas programadas serán válidas para toda la zona.
- Incluso en los segmentos con una velocidad programada más elevada, en este caso se procederá con el desplazamiento aplicando la velocidad más reducida.
Aquí no se emite ningún mensaje notificando que no se ha alcanzado la velocidad.
Esto sucede únicamente si no se puede mantener la velocidad más reducida programada.

20.4.3.1 Programación de zona de desplazamiento constante

Descripción de la situación

La velocidad debe mantenerse constante entre **P5** y **P12**.

```

4  SPLINE S1 Vel=0.5 m/s CPDAT 2 Tool[1]:pen Base[5] blue
5  SPL P4 CPDAT4
6  SLIN P5 ADAT
7  SPL P6
8  SLIN P7
9  SPL P8
10 SPL P9
11 SPL P10
12 SPL P11
13 SLIN P12 ADAT
14 SPL P13
    
```

15 ENDSPLINE

Definir un punto de inicio

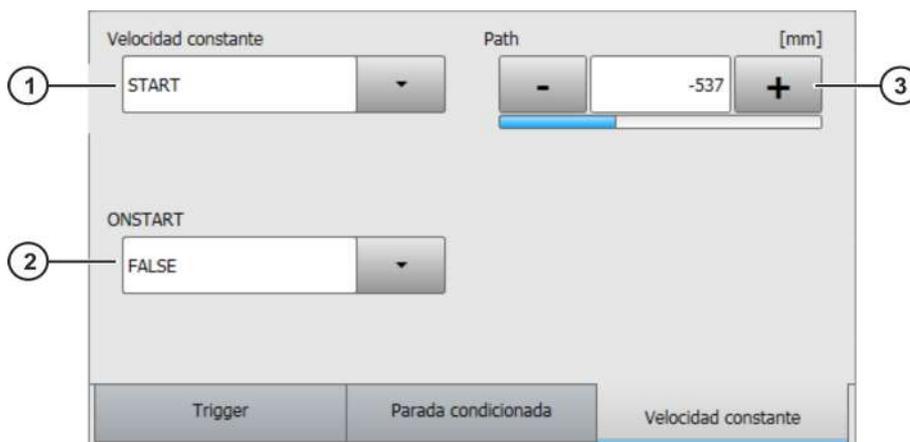


Fig. 20-38: Punto de inicio desplazado

Parámetro	Valor	Descripción
Velocidad constante	Inicio	Inicio de la zona de recorrido constante
ONSTART	FALSE	Start hace referencia al punto de destino Si el punto de destino es de posicionamiento aproximado, Start o End hacen referencia al inicio del arco de aproximación.
Path	-537 mm	El desplazamiento constante comienza 537 mm antes de alcanzar el punto de destino.

Definir el punto final

El procedimiento se corresponde con el registro del punto de inicio. Aquí hay que seleccionar END en el campo de Velocidad constante (1). También aquí se puede desplazar y registrar el punto en el espacio mediante *ADAT > Velocidad constante > END > Registrar ruta*.

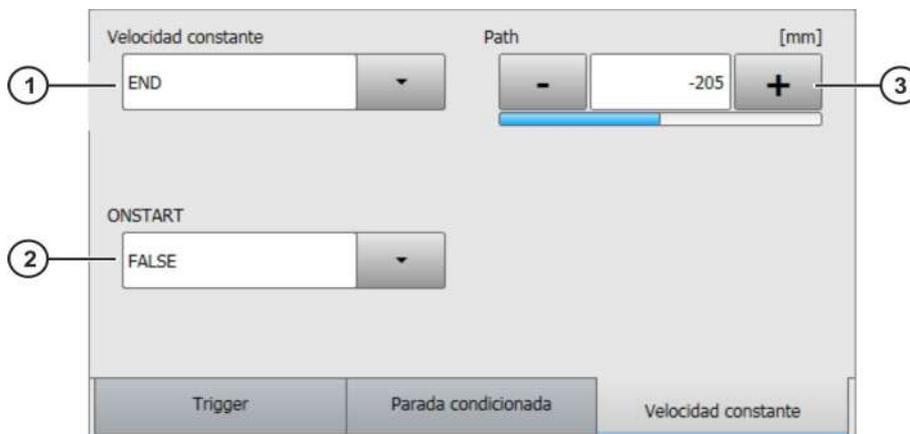


Fig. 20-39: Finalizar velocidad constante

Comprobar el desplazamiento constante

Probar el desplazamiento constante programado en el modo de servicio T1.

Aunque el robot recorre el contorno SPLINE a velocidad reducida, mediante la emisión del correspondiente mensaje, indica qué velocidad máxima que difiera de la velocidad programada puede alcanzar durante un desplazamiento constante.

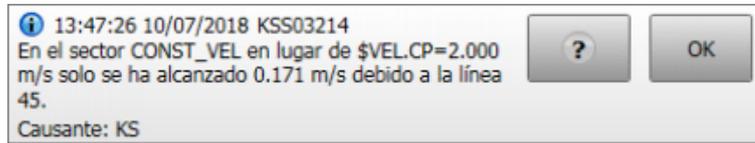


Fig. 20-40: Reducción de velocidad

Se visualiza el número de línea cuyo punto es la magnitud limitante en este caso.



Para ello, todos los Folds deben estar abiertos y, en la visualización, deben estar activadas también las líneas DEF, vista en detalle (ASCII) para poder asignar los números de línea.



Mediante la optimización selectiva del punto (posición u orientación) puede maximizarse la velocidad que se debe alcanzar.



Debe evitarse la prueba inicial directamente en T2 o Automático, ya que se emite un mensaje por cada punto de apoyo. Esto no es adecuado para la optimización de puntos de movimiento individuales.

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro	Programación de robots 1 KUKA System Software 8.6 Referencia de formación
Capítulo/apartado	Recorrer el contorno SPLINE con velocidad constante

Podrá encontrar más información en la documentación del cliente.



Libro	KUKA System Software 8.6 Instrucciones de servicio y programación para los integradores de sistemas
Filtro de búsqueda	Documentación > Instrucciones de manejo y programación
Capítulo/apartado	Programación con formularios inline > Programar movimientos Spline > Programar la zona de desplazamiento constante para el bloque Spline CP

20.4.3.2 Ejercicio (variante 1): Zona de desplazamiento constante y parada condicionada, contorno 1

¡Ahora es su turno!



Enunciado

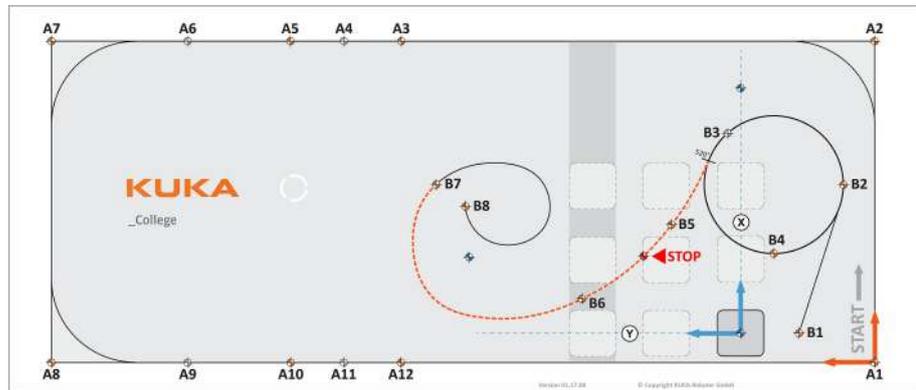


Fig. 20-41: Margen de desplazamiento constante, parada condicionada

1. Duplique su **programa:** _____ y guárdelo en el **programa:** _____.
2. Definir la zona naranja como zona de desplazamiento constante.
3. Ejecute una parada condicionada en el punto de la trayectoria identificado en rojo.
4. Comprobar el programa en los modos de servicio T1, T2 y automático.

20.4.3.3 Ejercicio (variante 2): Zona de desplazamiento constante y parada condicionada, contorno 2

Enunciado

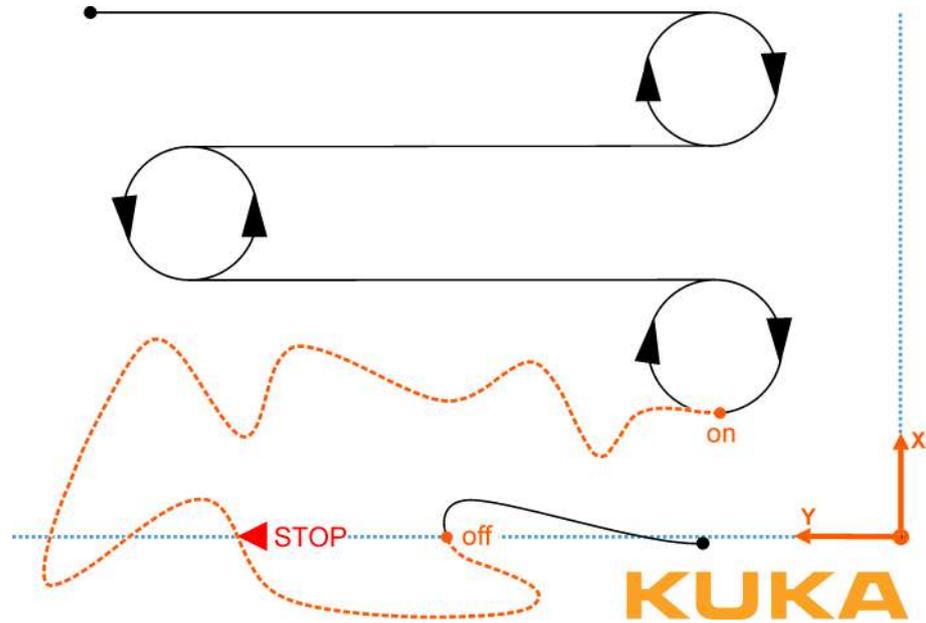


Fig. 20-42: Lógica ampliada con spline

1. Duplique su programa: _____ en el programa: _____.
2. Definir la zona verde como zona de desplazamiento constante.
3. Ejecute una parada condicionada en el punto de la trayectoria identificado en rojo.
4. Comprobar el programa en los modos de servicio T1, T2 y automático.

Configuración de una nueva estación de robot: Conmutar a la trayectoria del robot

21 Configuración de una nueva estación de robot: Herramienta ext. y pieza de trabajo (proyecto de programación)

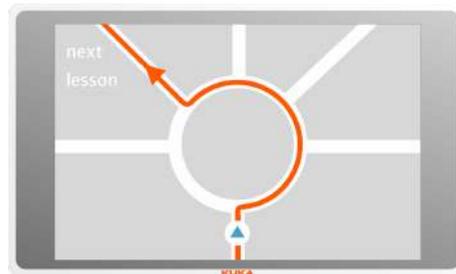
21.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y AUT.
- Está capacitado para medir los sistemas de coordenadas de herramienta y base.
- Está capacitado para crear programas propios con formularios inline y las trayectorias de los movimientos SPTP, SLIN y SCIRC

¿Qué le espera en esta unidad de aprendizaje?



Configuración de una nueva estación de robot: Herramienta ext. y pieza de trabajo (proyecto de programación)

- ¿Qué es una herramienta fija externa y cómo se mide?
- ¿Qué es una pieza de trabajo guiada por robot y cómo se mide?
- Aprenderá a desplazar y programar la pieza de trabajo guiada por robot en relación con la herramienta fija.
- En el marco de un proyecto de ejercicio, aplique lo que ha aprendido hasta ahora y optimice sus programas mediante la activación de programas y bucles.

21.2 Descripción de la situación

En este ejemplo, un robot guía las planchas laterales y el adhesivo se aplica en una herramienta fija.



Experimento mental:

Todo el mundo conoce el procedimiento matinal de cepillarse los dientes. ¿Cómo llega la pasta de dientes al cepillo de dientes? El cepillo es guiado exactamente a lo largo del tubo (TCP externo). Incluso antes de que el cepillo esté debajo del tubo se presiona sobre el tubo y la aplicación se realiza con precisión. También ha tenido en cuenta la consistencia de la pasta. Asimismo durante la parada. Antes de que el cepillo alcance el extremo, se deja de presionar el tubo. La pasta se "recoge" y de este modo ha evitado que se derrame.

Aviso: Sujete el tubo y mueva el cepillo de dientes debajo de este.



Fig. 21-1: Ejemplo pasta de dientes, TCP externo

21.3 ¿Cuál es el objetivo del proyecto de ejercicio?

Dentro del marco del proyecto de ejercicio, en la "placa" de la celda de formación College se debe aplicar pegamento a través de la "boquilla de

aplicación de pegamento externa". La trayectoria correspondiente está especificada en la lámina.

En este apartado de ejercicios aprenderá todos los pasos necesarios para realizar las tareas

- Medición de todos los sistemas de herramientas y sistemas básicos
- Programación de movimientos en un TCP externo



21.3.1 Interpolación del movimiento con la pieza de trabajo guiada por el robot

Herramienta guiada por robot

En la mayoría de los casos, el robot guía la herramienta y procesa o transporta las piezas. Para cada conjunto de movimientos programado (programar) se guarda la distancia desde el sistema de coordenadas Base activo hasta el sistema de coordenadas Tool activo (TCP). El movimiento depositado (interpolación) se refiere siempre al TCP de la herramienta guiada por el robot.

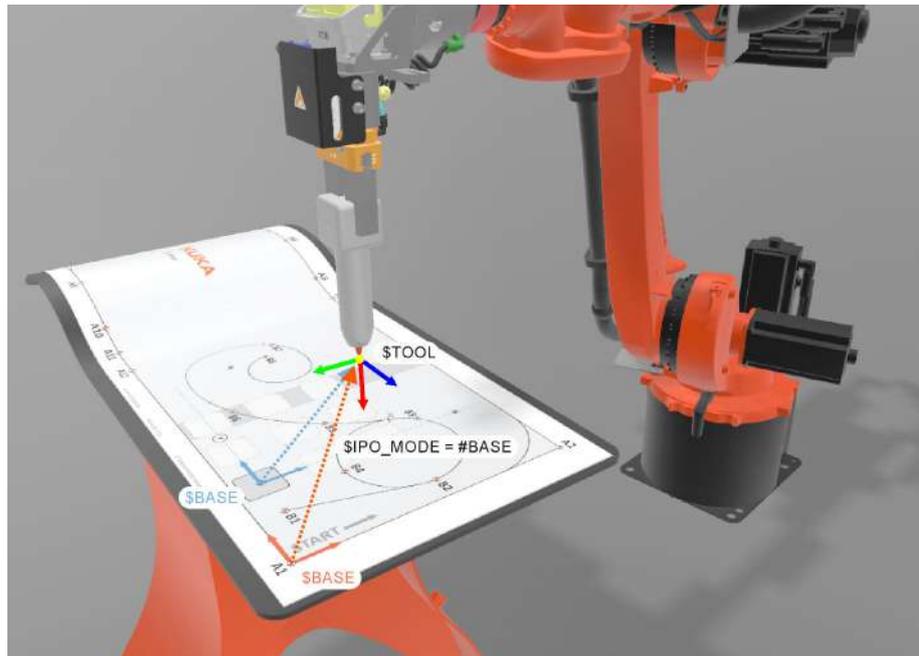


Fig. 21-2: \$IPO_MODE = #BASE

Herramienta guiada por robot/herramienta fija

En una aplicación con una herramienta fija y sin movimiento, la pieza de trabajo es guiada por el robot. La herramienta es inalterable en su posición y se almacena dentro de la unidad de control como base. La pieza es guiada por el robot y se gestiona como una herramienta. Sin embargo, la herramienta y la base tienen papeles invertidos. La unidad de control debe ser informada mediante la variable $\$IPO_MODE = \#TCP$ en el conjunto de movimientos, así como en el modo de desplazamiento manual, de que el movimiento (p. ej., SLIN) se refiere al conocido **Tool Center Point externo**. De este modo, es posible recorrer sin problemas los contornos de la pieza guiada por el robot.

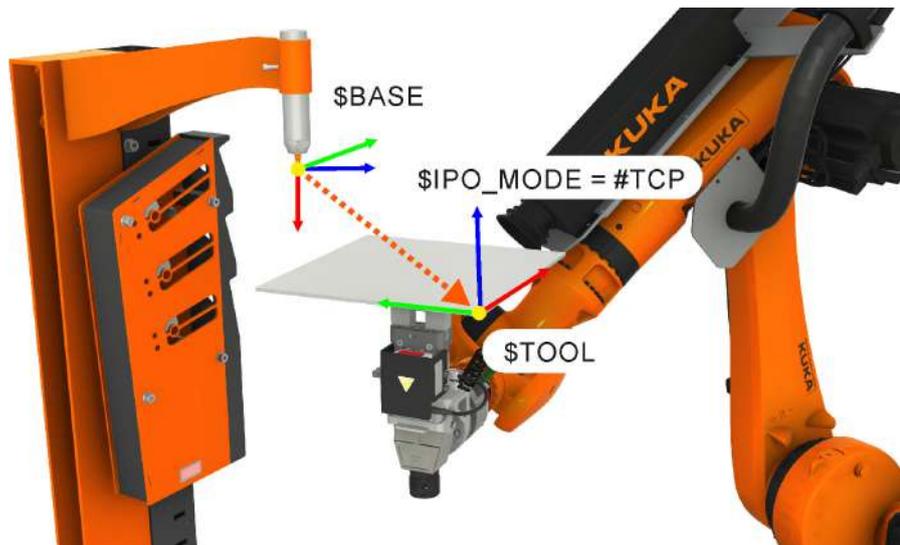


Fig. 21-3: \$IPO_MODE = #TCP

Procedimiento

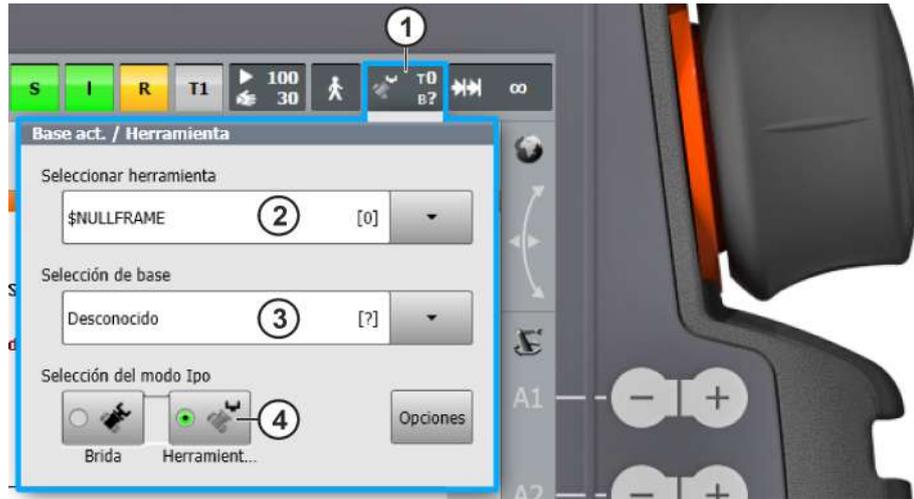


Fig. 21-4: Selección ext. TCP en el menú de opciones

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Conocer la pieza de trabajo guiada por robot y la herramienta fija

Desplazamiento manual de la pieza de trabajo guiada por robot y la herramienta fija

Programar la pieza de trabajo guiada por robot y la herramienta fija

21.3.2 Ejercicio: Desplazamientos manuales con herramienta fija

¡Ahora es su turno!



Enunciado 1



Fig. 21-5: Ejercicio: Desplazamiento manual con TCP externo

1. Activar la siguiente herramienta y base.
 - **ext. Herramienta**
N.º _____ nombre: _____
 - **(Brida) básica**
N.º _____ nombre: _____
2. Ajustar, en el menú de opciones de procesos manuales, la opción *Herram. ext.* (>>> Fig. 21-4)
3. Mover la *placa* hacia la *clavija externa*.
4. Mover y orientar la *placa* hacia la *clavija externa*. Para ello, probar las diferencias entre el desplazamiento en el sistema de coordenadas Tool y Base.
5. Ajustar, en el menú de opciones de procesos manuales, la opción *Brida*.
6. Mover y orientar la *placa* en la *clavija externa*.

Enunciado 2

1. Desplazar la placa también con el ratón 6D.
2. Configurar intencionadamente el IpoMode "incorrecto" y desplazar el robot.
¿Qué le llama la atención?
3. Comprobar el desplazamiento manual en el contexto de otros sistemas de coordenadas.

21.4 Conocer la medición de una herramienta fija

Descripción

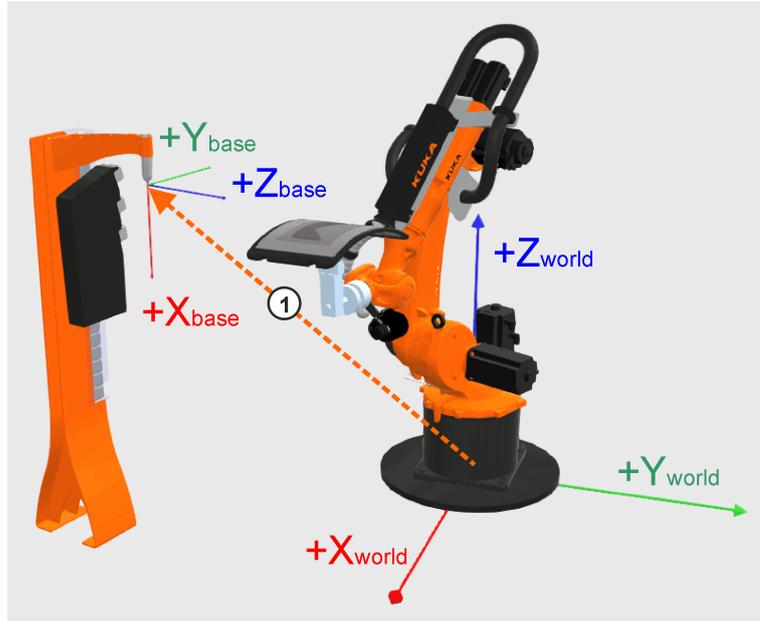


Fig. 21-6: Fundamentos de medición de la herramienta fija

- 1 Distancia (XYZ) y alineación (ABC) del TCP externo con respecto al sistema de coordenadas universales



El TCP externo (1) relacionado con el sistema de coordenadas universal (\$WORLD) se gestiona como sistema de coordenadas base.

La medición de la herramienta fija (externa) consta de dos pasos:

1. Determinación de la posición del TCP externo en relación con el origen del sistema de coordenadas universal
2. Orientación del sistema de coordenadas en el TCP externo

21.4.1 ¿Cómo se mide una herramienta fija?

Paso 1: Determinación del TCP externo

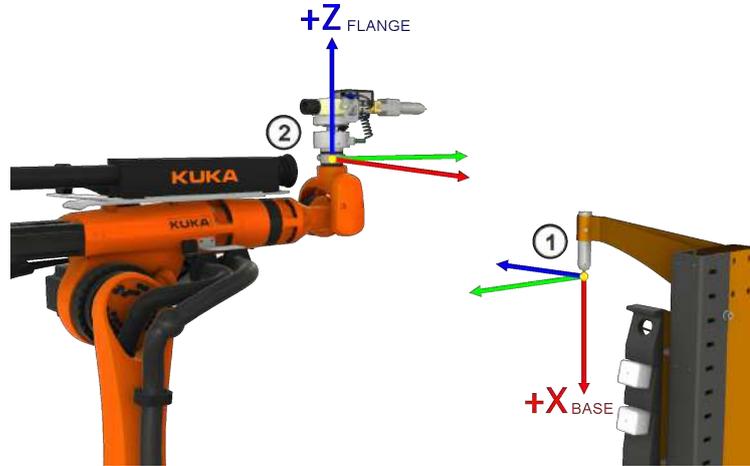
Con un TCP de herramienta conocido se programa el TCP de la herramienta externa.



Paso 2: Determinar la dirección de avance

La dirección de impacto "virtual" de la herramienta externa (1) se determina con el sistema de coordenadas de la brida.

En este caso, la orientación Z negativa corresponde a la orientación X positiva con el método 5D.



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Medir la pieza de trabajo guiada por robot y la herramienta fija

21.4.2 Preguntas: Medir una herramienta externa y una pieza guiada por robot

Lo que se debe saber tras el ejercicio:



¿Cómo se mide un sistema de coordenadas en una pieza montada en una brida de robot?

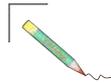


Fig. 21-7: Campo de respuesta

¿Cómo se calcula el TCP de una herramienta externa?



Fig. 21-8: Campo de respuesta

¿Qué ventajas se obtienen mediante el uso de un TCP externo?

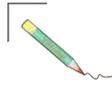


Fig. 21-9: Campo de respuesta

¿Cuáles son los ajustes necesarios para desplazar con un TCP externo en dirección de impacto de la herramienta?



Fig. 21-10: Campo de respuesta

21.5 Medición de una pieza de trabajo guiada por robot

Descripción

En la pieza de trabajo guiada por robot se mide un sistema de coordenadas definido por el cliente.

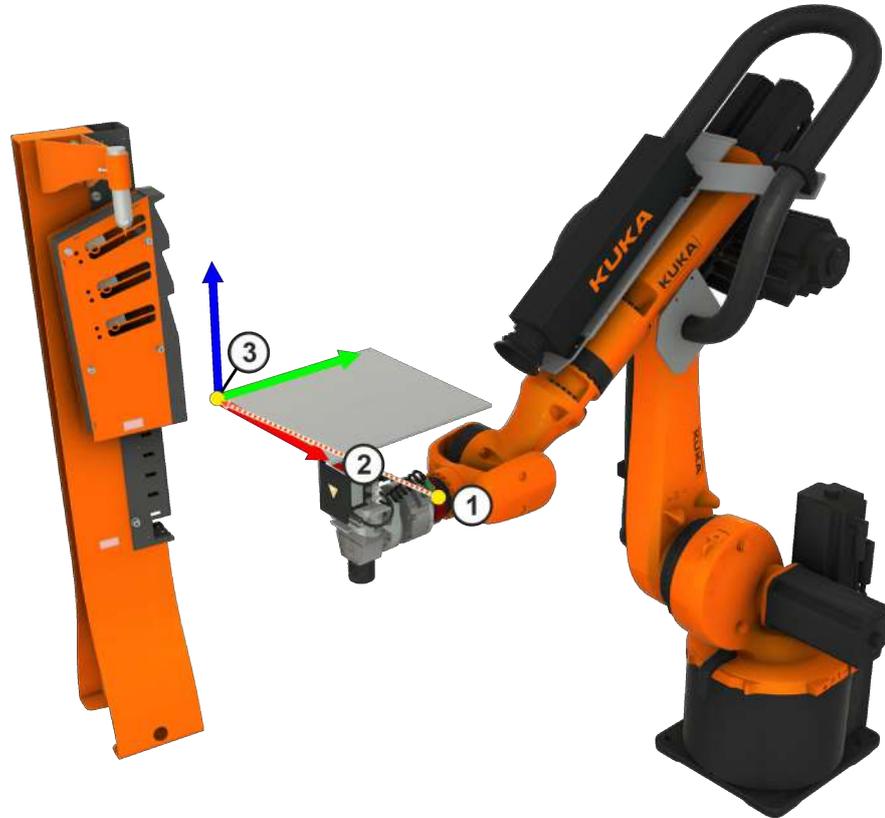


Fig. 21-11: Medición de la pieza por medición directa

- 1 Punto central de la brida
- 2 Distancia entre 1 y 3, así como giro
- 3 Origen del sistema de coordenadas de la pieza medida

Podrá encontrar más información en la documentación del cliente.

La medición indirecta es muy poco frecuente y se describe en la documentación *KUKA System Software 8.6*. No es un componente de este módulo de aprendizaje.



Libro

KUKA System Software 8.6

Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

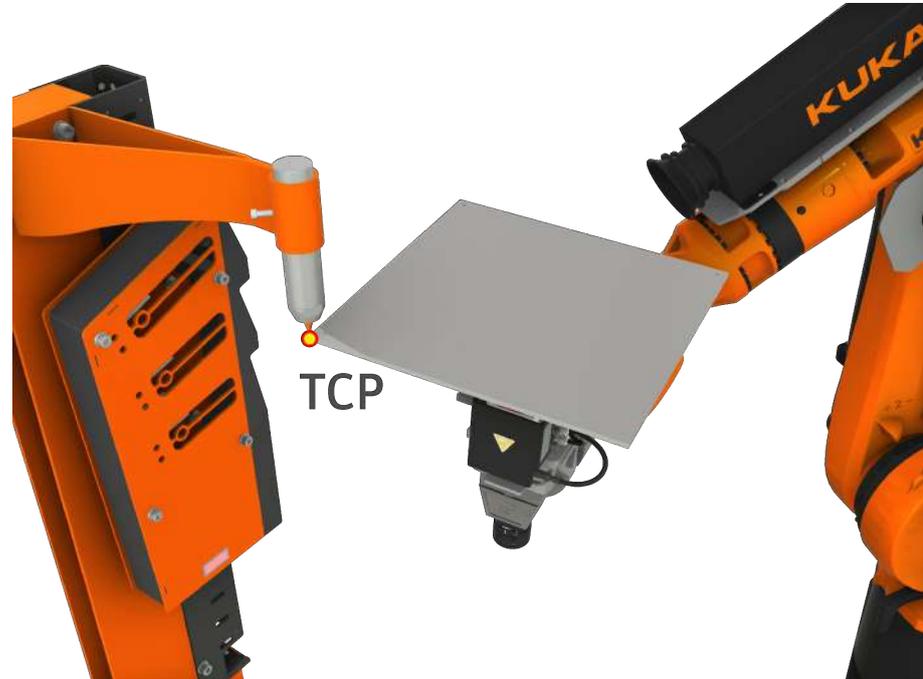
Medición > Vista general de los métodos de medición > Método indirecto

21.5.1 Medir una pieza guiada por robot

Paso 1: Desplazar al origen de coordenadas

Desplazar con la herramienta al TCP de la herramienta fija.

En este proceso hay que hacer coincidir el origen del sistema de coordenadas definido con el TCP de la herramienta fija.



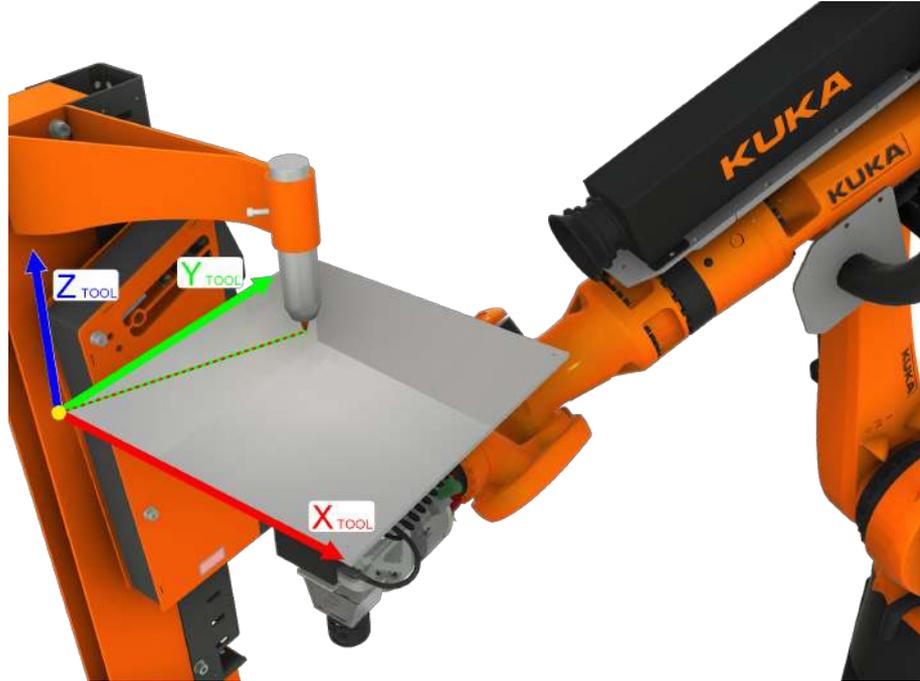
Paso 2: Desplazar el punto sobre la orientación X

Con el TCP de la herramienta fija desplazarse a un punto sobre el eje X positivo del sistema de coordenada de la pieza de trabajo que hay que definir.



Paso 3: Desplazar el punto sobre el plano XY

Con el TCP de la herramienta fija desplazarse a un punto sobre el plano XY del sistema de coordenada de la pieza de trabajo que hay que definir.



Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Medir la pieza de trabajo guiada por robot y la herramienta fija

Podrá encontrar más información en la documentación del cliente.



Libro	KUKA System Software 8.7 Instrucciones de servicio y programación para los integradores de sistemas
Filtro de búsqueda	Documentación > Instrucciones de manejo y programación
Capítulo/apartado	Puesta en servicio y reanudación del servicio > Medición Medir TOOL o indicarla de forma numérica (herramienta/pieza en brida) Medir BASE o indicarla de forma numérica (base/herramienta fija)

Configuración de una nueva estación de robot: Herramienta ext. y pieza de trabajo (proyecto de programación)

21.6 Ejercicio: Unidad de calibración para BASE

¡Ahora es su turno!



Enunciado

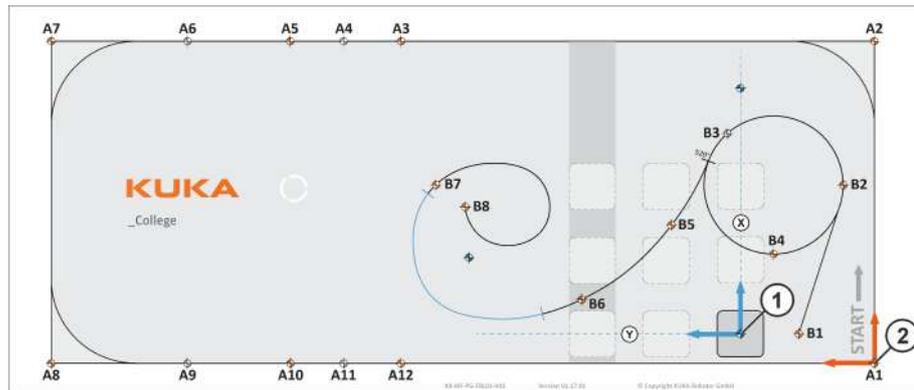


Fig. 21-12: Medición de BASE sobre la mesa

1. Medir la base roja n.º _____ nombre: _____ sobre la mesa con el método de 3 puntos.
 - Emplear la herramienta ya medida n.º _____ nombre: _____ como herramienta de medición.
2. Guardar los datos de la base medida.
3. Activar la herramienta n.º _____ nombre: _____ y la base n.º _____ nombre: _____.
 - Desplazar la herramienta al origen de la base y mostrar la posición cartesiana real.

X Y Z A B C

.....



- Utilizar las herramientas y los sistemas base indicados por el instructor.
- Una vista general de la ocupación prevista del sistema en:
- (>>> 25.1 "Vista general de los sistemas de herramientas y base" Página 421)

21.7 Ejercicio: Medición TOOL con el método de 2 puntos

¡Ahora es su turno!



Enunciado

1. Utilizar para la medición de la herramienta de la garra n.º _____ nombre: _____.
2. Mida el TCP de la garra con la ayuda del método de 4 puntos XYZ, tal y como se puede ver en la figura:

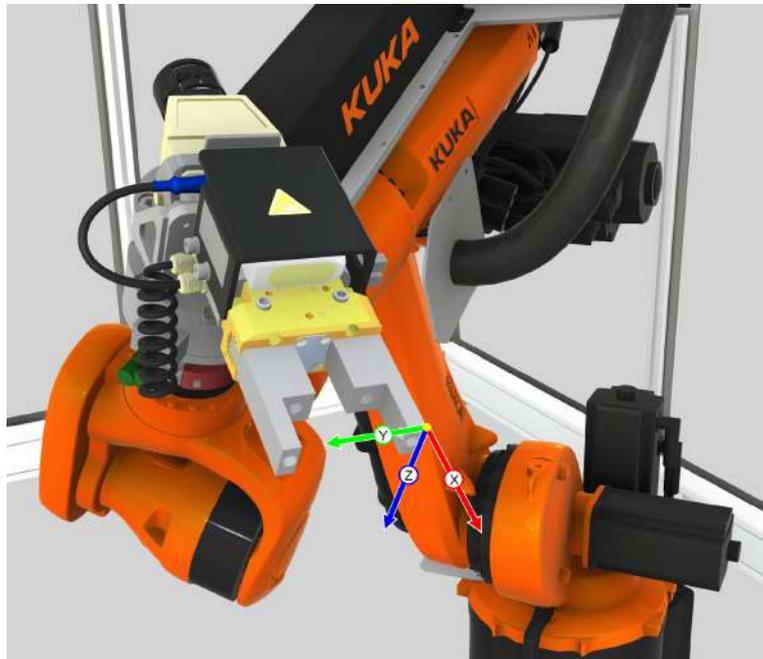


Fig. 21-13: College garra: Posición del TCP

3. Mida la orientación del sistema de coordenadas de la garra con la ayuda del método de 2 puntos ABC.
4. Introducir los datos de carga para la garra n.º _____ nombre: _____.
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.



- Utilizar las herramientas y los sistemas base indicados por el instructor.
- Una vista general de la ocupación prevista del sistema en:
- (>>> 25.1 "Vista general de los sistemas de herramientas y base" Página 421)



De acuerdo con las indicaciones del formador, utilizar dichas indicaciones para la medición/verificación del siguiente apartado.

21.8 Ejercicio (variante 1): Medir una herramienta externa y una pieza guiada por robot

Enunciado 1

Medición "Boquilla de aplicación de pegamento"

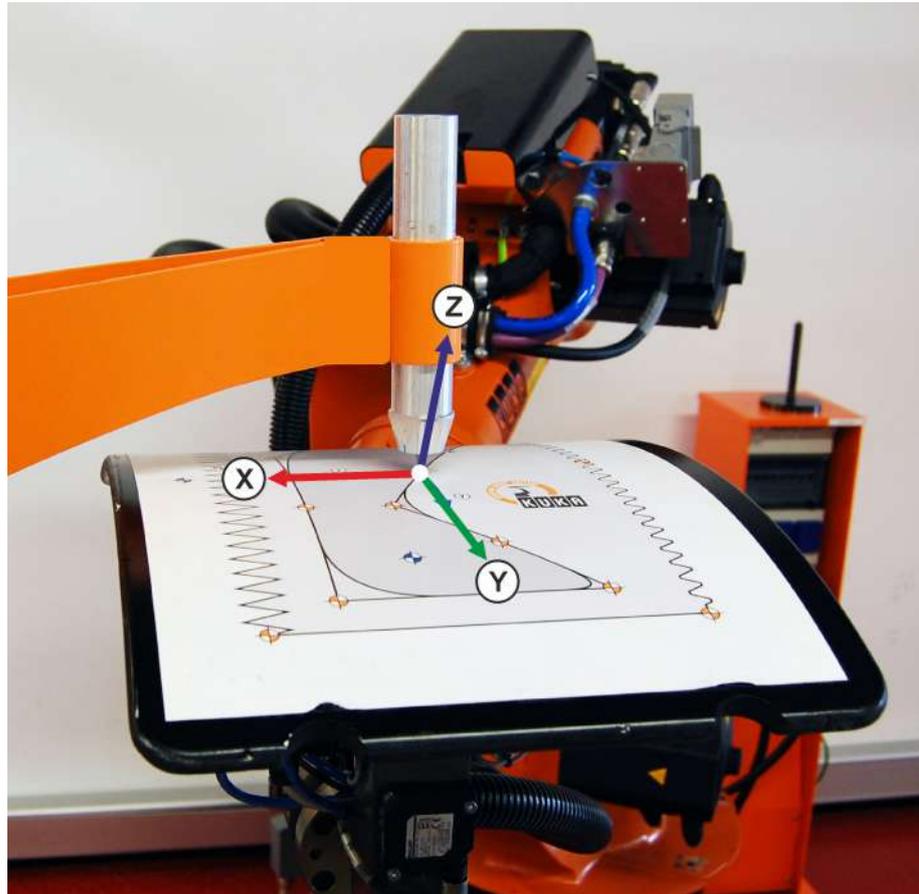


Para medir la herramienta fija, la herramienta ya medida *n.º*: _____ *nombre*: _____ debe emplearse como herramienta de referencia.

- Asignar a la herramienta fija el *n.º*: _____ *nombre*: _____.
- Tener en cuenta que en cada medición se deben guardar los datos.

Enunciado 2

Medición de la "Placa"



Efectuar la medición de la pieza guiada por el robot.

- Asignar, al hacerlo, el n.º: _____ nombre: _____.
- Introducir los datos de carga.
- Usar para ello la garra correcta con placa. (>>> 25.2 "KUKA College: Datos de carga/valores de offset para herramientas/Herramientas" Página 421)

Enunciado 3

Comprobar la "herramienta externa" medida y la "pieza de trabajo guiada por robot"

1. Al terminar la medición, activar la herramienta externa y la pieza de trabajo para el desplazamiento manual.
 - Utilizar correctamente y de forma lógica el sistema de coordenadas de base y de herramienta y desplazar el robot.
2. Desplazarse con el TCP al origen de coordenadas base de la pieza medida y visualizar la posición actual de forma cartesiana.

Posición actual:

X	Y	Z	A	B	C
.....



- Utilizar las herramientas y los sistemas base indicados por el instructor.
- Una vista general de la ocupación prevista del sistema en:
- (>>> 25.1 "Vista general de los sistemas de herramientas y base" Página 421)

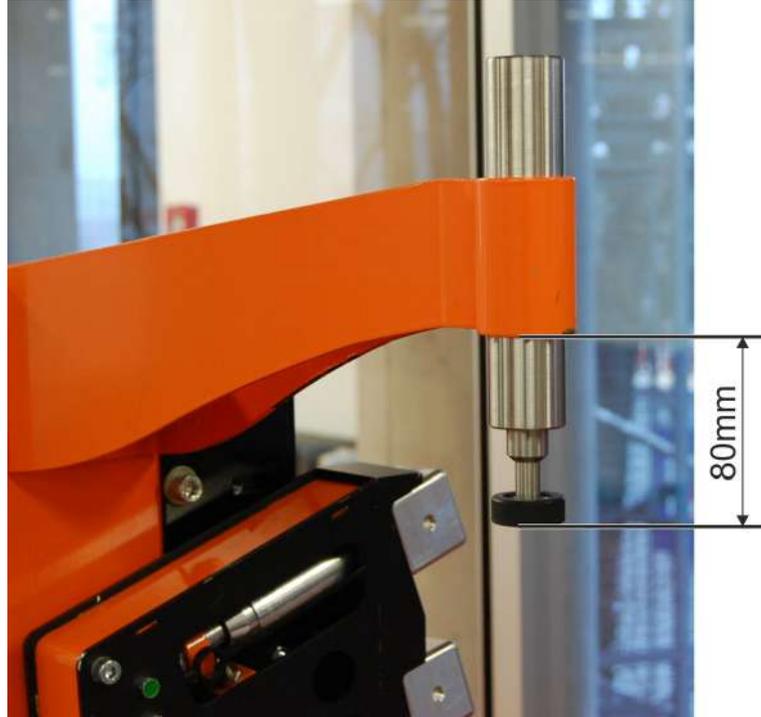


De acuerdo con las indicaciones del formador, utilizar dichas indicaciones para la medición/verificación del siguiente apartado.

21.9 Ejercicio (variante 2): Medir una herramienta de rodillos externa y una pieza de trabajo guiada por robot

Enunciado 1

Medición "Herramienta de rodillo"



Para medir la herramienta fija, la herramienta ya medida *n.º*:_____ *nombre*:_____ debe emplearse como herramienta de referencia.

- Asignar a la herramienta fija el *n.º*:_____ *nombre*:_____.
- Tener en cuenta que en cada medición se deben guardar los datos.

Enunciado 2

Medición de un "Componente 3D"

Efectuar la medición de la pieza guiada por el robot.

- Asignar, al hacerlo, el *n.º*:_____ *nombre*:_____.
- Introducir los datos de carga.
- Usar para ello la garra correcta con placa.

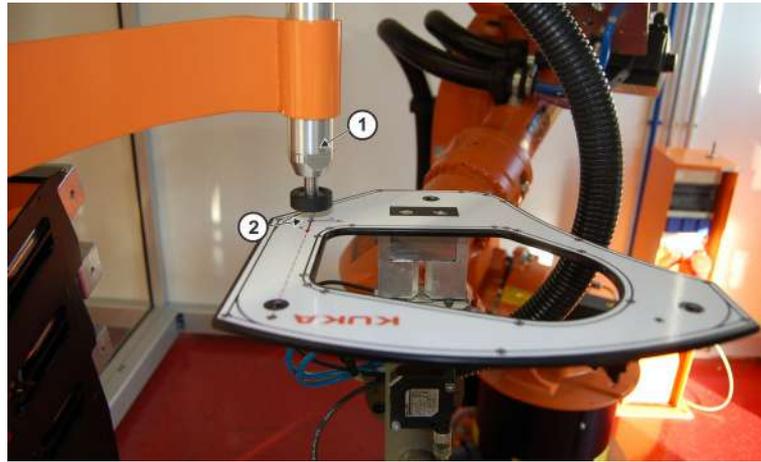


Fig. 21-14: Componente 3D en la herramienta de engrapado externa

- | | |
|-------------------------------|---|
| 1 Herramienta de engrapado | 2 Sistema de coordenadas de la pieza |
|-------------------------------|---|

Variante: Medición de un "Componente 3D"

También es posible la medición del componente 3D con la variante "Clavija" como herramienta externa.

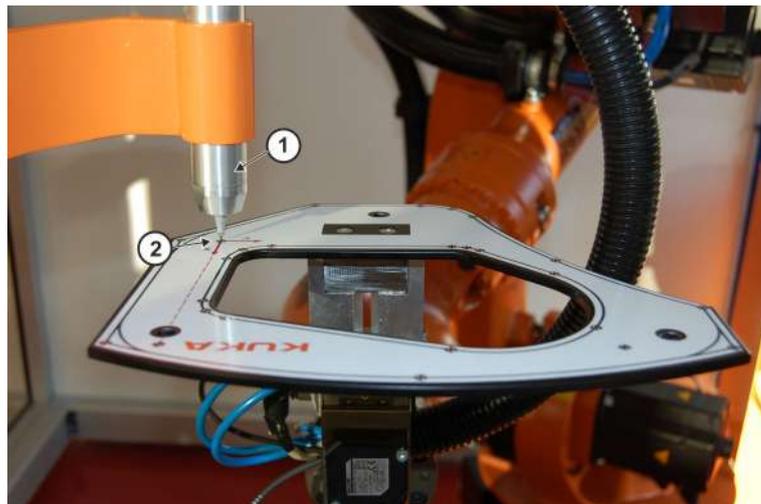


Fig. 21-15: Componente 3D en el TCP externo

Enunciado 3

Comprobar la "herramienta externa" medida y la "pieza de trabajo guiada por robot"

1. Al terminar la medición, activar la herramienta externa y la pieza de trabajo para el desplazamiento manual.
 - Utilizar correctamente y de forma lógica el sistema de coordenadas de base y de herramienta y desplazar el robot.
2. Desplazarse con el TCP al origen de coordenadas base de la pieza medida y visualizar la posición actual de forma cartesiana.

Posición actual:

X	Y	Z	A	B	C
.....

Configuración de una nueva estación de robot: Herramienta ext. y pieza de trabajo (proyecto de programación)



- Utilizar las herramientas y los sistemas base indicados por el instructor.
- Una vista general de la ocupación prevista del sistema en:
- (>>> 25.1 "Vista general de los sistemas de herramientas y base" Página 421)



De acuerdo con las indicaciones del formador, utilizar dichas indicaciones para la medición/verificación del siguiente apartado.

21.10 Ejercicio (variante 1): Programar movimientos con TCP externo

Tiene el encargo de configurar la instalación de pegado de cristales "llave en mano".



Enunciado 1

Los dos módulos de programa para recoger y depositar la placa se han creado y se han probado.



Enunciado 2

1. Compruebe el ajuste del robot.
2. Programe un offset para la "Pieza de trabajo placa con garra" y guarde el offset de ajuste.
3. Mida la boquilla de aplicación de pegamento externa como base n.º ____ y la placa como herramienta n.º ____ .

Enunciado 3

Programación de un contorno para la aplicación de adhesivo:

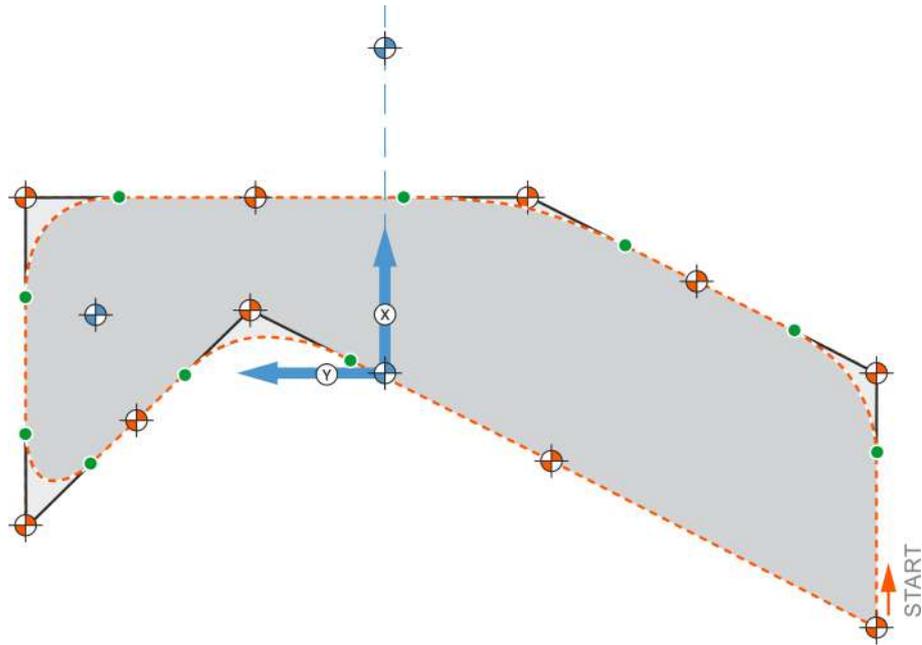
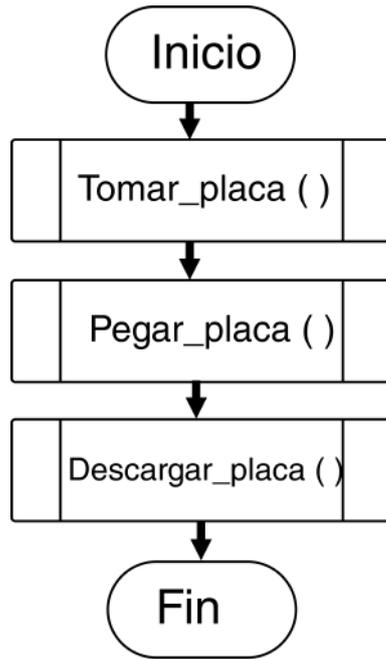


Fig. 21-16: Pegar la placa

1. Sujetar manualmente la placa en la garra.
2. Programar el contorno indicado en la placa en un nuevo **programa**: _____ .
 - Utilizar para ello la herramienta externa medida y la placa.
 - **ext. Herramienta**
N.º _____ nombre: _____
 - **(Brida) básica**
N.º _____ nombre: _____
 - 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*.
 - Utilice un bloque SPLINE para la programación de movimiento.
3. Pruebe su programa en los modos de servicio T1, T2 y AUT.

Enunciado 4



1. Cree un nuevo módulo de programa con el nombre **main()** .
2. Abra el programa main() y vincule correctamente los programas creados.

```

DEF main()
INI

schild_holen()
schild_kleben()
schild_ablegen()

end
    
```

3. Cierre su programa main() y selecciónelo para probarlo.
4. Pruebe el programa main() en los diferentes modos de servicio.



Ver también: (>>> [22.3 "Programar un bucle sinfin" Página 356](#))

Lo que se debe saber tras el ejercicio:



¿Cuál es el punto de referencia de la velocidad de pegado que se ha programado?

Fig. 21-17: Campo de respuesta

¿Cómo se activa la herramienta externa en el programa?



Fig. 21-18: Campo de respuesta

21.11 Ejercicio (variante 3): Paletizado y despaletizado de cubos

¡Ahora es su turno!



Enunciado

Tiene el encargo de recoger cubos del cargador y depositarlos en una cuadrícula predefinida de 3x3.



Fig. 21-19: 9 cubos

- Deben medirse los sistemas de coordenadas base y de herramienta necesarios.
- Los programas están diseñados de forma modular para garantizar una "activación múltiple".

Enunciado 1

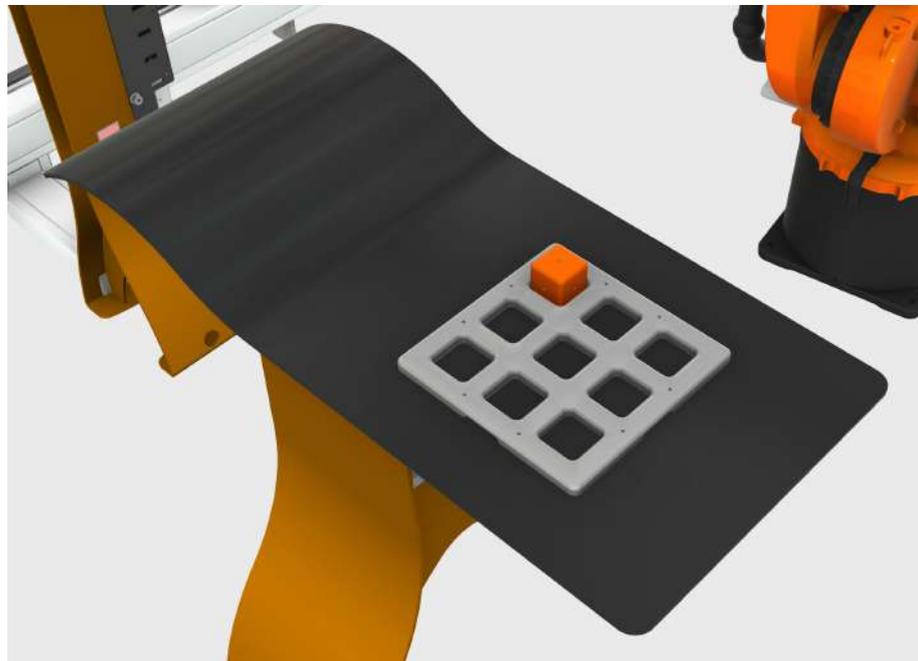


Los dos módulos de programa para recoger/depositar cubos se han creado y se han probado.



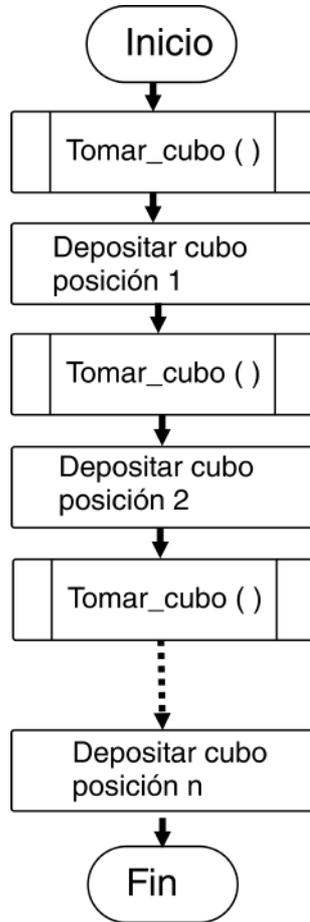
Ver también: (>>> *"Ejercicio (variante 1): Tomar y descargar el cubo"*
Página 267)

Enunciado 2



1. Cree un nuevo **módulo de programa**: _____
2. Utilizar para ello la siguiente herramienta y base.
 - **Herramienta**
N.º _____ nombre: _____
 - **Base**
N.º _____ nombre: _____

3. Posicione la cuadrícula de cubos sobre su mesa de formación. Compruebe la base medida con respecto a la cuadrícula de cubos.
4. Saque un cubo del depósito de cubos con la garra. Para ello, utilice los posibles módulos de programa ya existentes.
5. Saque más cubos del depósito de cubos y rellene los campos que aún estén libres según las especificaciones del formador.

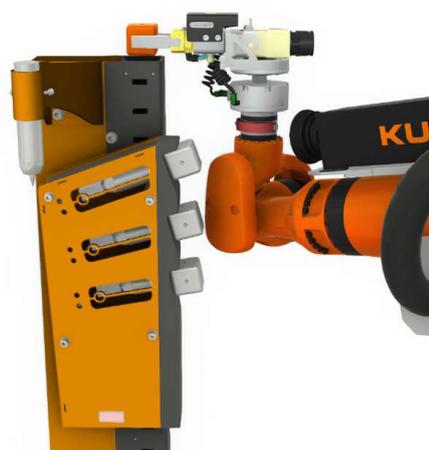


Utilice módulos de programa ya existentes para la recogida/recuperación de los cubos.

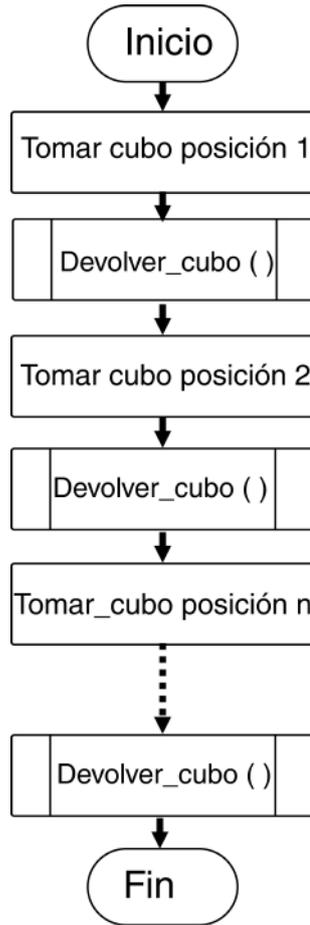


Ver también: (>>> [22.3 "Programar un bucle sinfin" Página 356](#))

Enunciado 3



1. Cree un nuevo **programa**:_____ para la despaletización de los cubos.
2. Utilizar para ello la siguiente herramienta y base.
 - **Herramienta**
N.º _____ nombre: _____
 - **Base**
N.º _____ nombre: _____
3. Agarre un cubo en la cuadrícula de cubos y vuelva a llevarlo al depósito de cubos. Para ello, utilice los posibles módulos de programa ya existentes.
4. Repita el proceso hasta que la cuadrícula de cubos esté vacía.



Utilice módulos de programa ya existentes para la recogida/devolución del cubo.



Ver también: (>>> [22.3 "Programar un bucle sinfin" Página 356](#))

Enunciado 4

1. Crear un nuevo módulo de programa con el nombre **Procedimiento()**.
2. Abra el programa **Procedimiento()** y vincule los módulos de programa creados para el paletizado y el despaletizado.

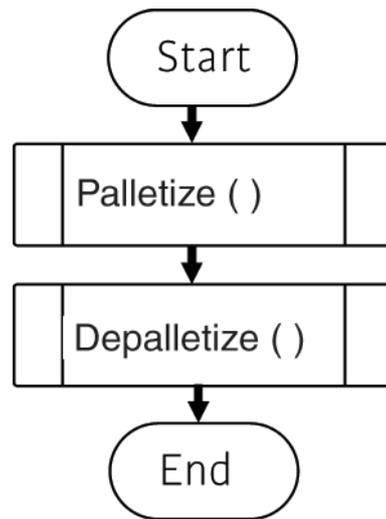


Fig. 21-20: PAP

3. Cierre su programa principal **Procedimiento()** y selecciónelo para realizar la prueba.
4. Pruebe el programa principal **Procedimiento()** en los diferentes modos de servicio.



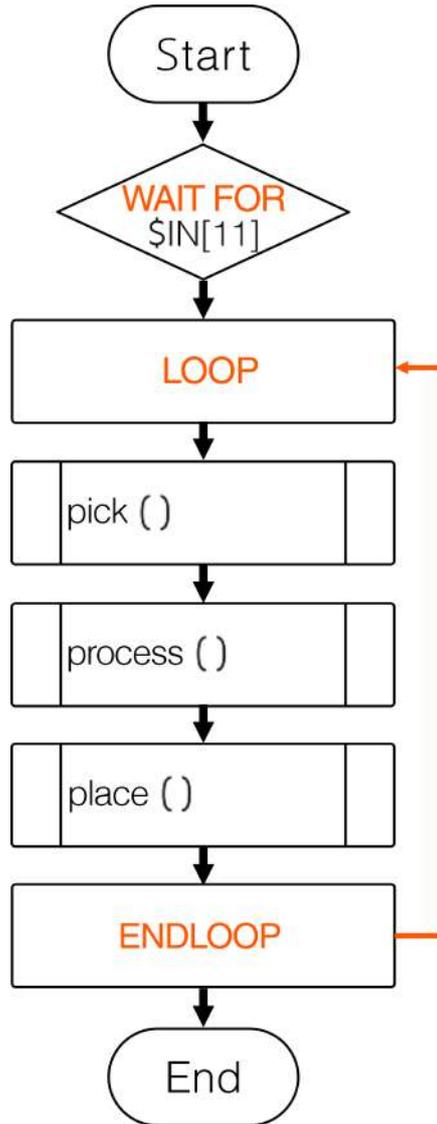
Ver también: (>>> [22.3 "Programar un bucle sinfín" Página 356](#))

21.12 Ejercicio: Programar "WAIT-FOR"

Enunciado

Antes de ejecutar el bucle sinfin se debe consultar si la estación del robot está libre.

1. Su programa principal no debe ejecutarse hasta que la entrada digital [11] sea TRUE. Esta comprobación debe realizarse una vez antes del flujo de programa.
2. Pruebe el programa en los distintos modos de servicio T1, T2 y Automático.



22 Repetición condicionada de programas

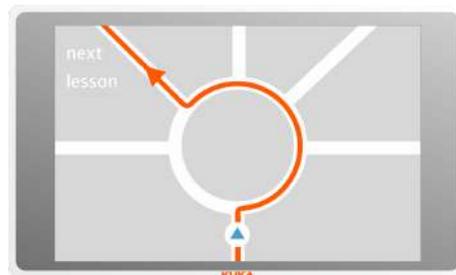
22.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y AUT.
- Está capacitado para abrir programas y realizar modificaciones, también en KRL.

¿Qué le espera en esta unidad de aprendizaje?



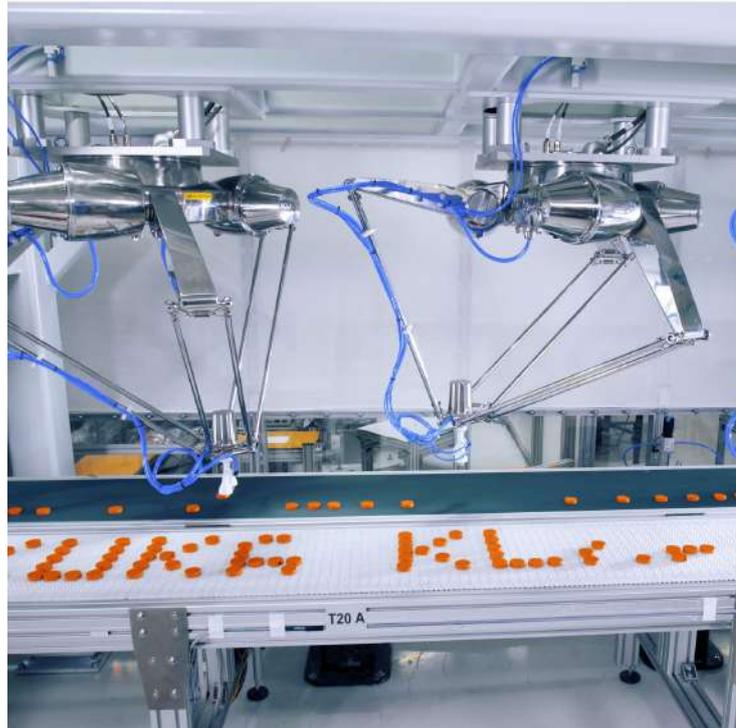
Repetición condicionada de programas principales

- Ampliará y optimizará sus programas mediante la repetición y la ramificación específicas.
- Este módulo de aprendizaje complementa de forma puntual o completa el proyecto de programación del seminario.
- En este documento se explica la programación de estructuras de control mediante el formulario inline:
 - Bucle sinfin (LOOP-ENDLOOP)
 - Bucle de conteo (FOR-ENDFOR)
 - Bucle finito (WHILE-ENDWHILE)
 - Bucle infinito (REPEAT UNTIL)
 - Consulta o ramificaciones (IF-THEN-ELSE)
 - Distribuidor (SWITCH-CASE)

22.2 Descripción de la situación

En esta aplicación con un robot KUKA Delta todo va muy rápido. Desde una cinta transportadora se agarran pequeños objetos redondos en décimas de segundo y se depositan en una posición predeterminada. Una vez terminado, el procedimiento comienza desde el principio. Los robots se encuentran en un bucle sinfin. Una cámara capta dónde y cómo se agarran y se depositan las piezas. Con la información suministrada por la cámara se pueden tomar decisiones, p. ej., dónde se deposita una pieza.

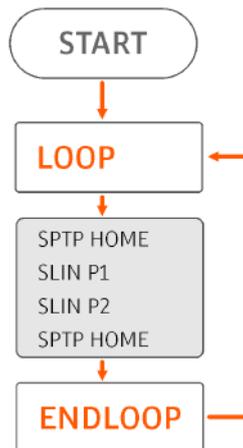
El depósito en sí podría ejecutarse mediante estructuras clásicas "IF THEN" o "SWITCH CASE".



22.3 Programar un bucle sinfin

Descripción

El bucle sinfin es un bucle que se vuelven a ejecutar después de cada ejecución. La ejecución se puede interrumpir por influencias externas.



- El bucle sinfin se puede abandonar mediante `EXIT`.
- Al abandonar un bucle sinfin con `EXIT` se deberá asegurar la ausencia de colisiones.

- En caso de que haya dos bucles sinfin intercalados, también serán necesarios dos comandos EXIT para abandonar ambos bucles.

22.3.1 Programar un bucle sinfin con formularios inline

Descripción de la situación

Ya ha preparado una serie de programas que representan un ciclo de producción completo. Estos programas deben ejecutarse ahora de forma consecutiva y el ciclo debe repetirse.

En nuestro ejemplo, estos son los programas:

1	pick_workpiece	Recoger_pieza de trabajo
2	process_workpiece	Procesar_pieza de trabajo
3	place_workpiece	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-1: Programar LOOP-ENDLOOP, paso 1



Solo en el modo "Abrir" es posible añadir formularios inline de lógica.

2. Marque la célula de programa en la que se debe añadir el bucle LOOP-ENDLOOP.
Pulsar el botón "Comandos" y seleccionar "Lógica" en el menú contextual.



Fig. 22-2: Programar LOOP-ENDLOOP, paso 2

3. En el menú contextual "Lógica" seleccionar el bucle "LOOP..ENDLOOP".



Fig. 22-3: Programar LOOP-ENDLOOP, paso 3

4. El bucle "LOOP-ENDLOOP" se añade automáticamente en el área del programa marcada.
Después de la comprobación, aceptar el formulario inline con el botón "Comando OK".



Fig. 22-4: Programar LOOP-ENDLOOP, paso 4



Con el botón "Añadir cond. canc." se puede ampliar el bucle LOOP-ENDLOOP con un bucle FOR como condición de cancelación.

5. Marque la línea con el texto guardado en gris como comentario *";insert your code here"* (1) y muestre el teclado de pantalla con la tecla "Clavija" (2).



Fig. 22-5: Programar LOOP-ENDLOOP, paso 5

6. Introducir sucesivamente los módulos de programa que se deben activar.

La activación del módulo se identifica con () de acuerdo con el nombre del programa.

```
pick_workpiece()
process_workpiece()
place_workpiece()
```



Fig. 22-6: Programar LOOP-ENDLOOP, paso 6



Mediante el botón "Desarrollo del movimiento" también se pueden registrar y programar movimientos del robot.

Después de comprobar las entradas (2), cerrar el teclado de pantalla con el símbolo "X" naranja (3).

7. Cierre el programa abierto (1). Se ha modificado el diálogo KRC:\..../Nombre de módulo.src. Confirmar la pregunta ¿De-sea guardar las modificaciones? con el botón "S" (2).



Fig. 22-7: Programar LOOP-ENDLOOP, paso 7

8. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-8: Programar LOOP-ENDLOOP, paso 8

Bucle sinfín a través del formulario inline

LOOP

Fig. 22-9: Formulario inline LOOP ... ENDLOOP sin campos

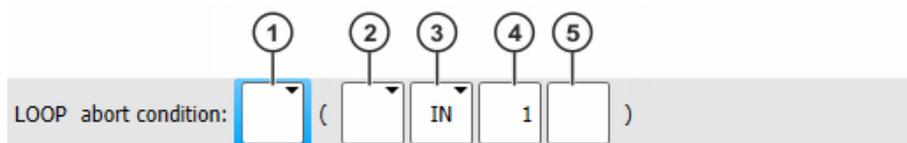


Fig. 22-10: Formulario inline LOOP ... ENDLOOP con campos

Para mostrar los campos, tocar el botón **Añadir cond. canc..**

Pos.	Descripción
1	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío]
	<p>Agregar la combinación externa. El término se encuentra entre las expresiones entre paréntesis.</p> <p>Añadir el término deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
2	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío]
	<p>Agregar la combinación interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <p>Añadir el operador deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
3	<p>Señal que se evalúa. Selección por defecto:</p> <ul style="list-style-type: none"> • IN, OUT, CYCFLAG, TIMER o FLAG <p>También es posible programar otras señales: Mediante el botón Texto libre es posible mostrar un campo en vez de la selección por defecto en el que se puede introducir el KRL.</p> <p>Lista de sistema vuelve a mostrar la selección por defecto. Al tocar los botones puede cambiarse entre ellos.</p>
4	<p>Este campo solo está disponible cuando se ha seleccionado una señal de la selección por defecto.</p> <p>Introducir el número de la señal.</p>
5	<p>Este campo solo está disponible cuando se ha seleccionado una señal de la selección por defecto. Si la señal ya tiene nombre, este se muestra.</p> <p>A partir del grupo de usuario experto: Puede introducirse un nombre o puede modificarse el nombre existente. Mientras no se haya guardado el formulario inline, pueden volver a restablecerse estas modificaciones pulsando Texto largo.</p>

Restricciones en "Modificar"

No es posible modificar el formulario inline mediante el botón **Modificar**.

22.3.2 Ejercicio: Programar "LOOP-ENDLOOP"

¡Ahora es su turno!



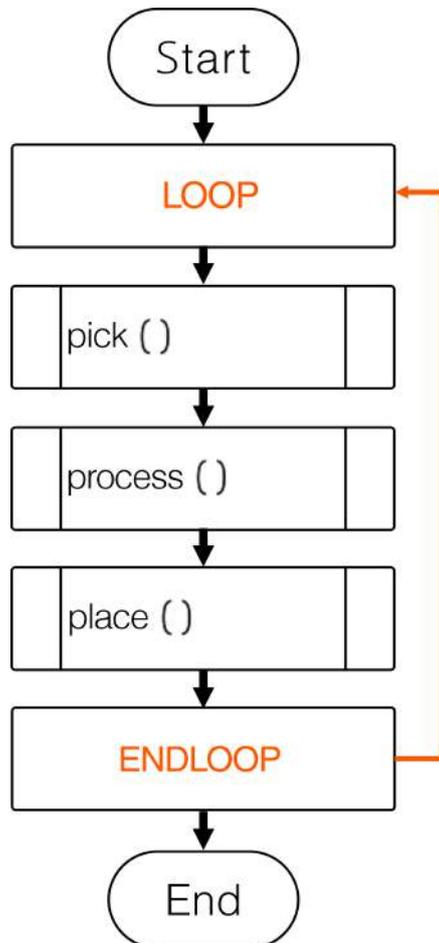
Enunciado

Esto ya está funcionando bastante bien.

¿Qué más podría optimizarse?

Correcto, actualmente debe resetear y reiniciar el programa después de cada ejecución.

Esta repetición cíclica se puede solucionar con un bucle sinfin.



Enunciado

1. Abra su programa principal y amplíelo con un bucle sinfin.
2. Pruebe el programa en los distintos modos de servicio T1, T2 y Automático.

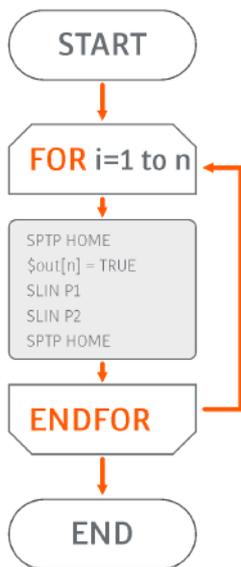


Ver: (>>> 22.3 "Programar un bucle sinfin" Página 356)

22.4 Programar un bucle de conteo

Descripción

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 valor para *increment* debe indicarse como número entero y no puede ser ninguna variable.

22.4.1 Programar un bucle de conteo con formularios inline

Descripción de la situación

Sus programas preparados que representan un ciclo de producción completo, deben ejecutarse de forma consecutiva. Después de cinco ciclos, el robot debe detenerse automáticamente y retroceder la posición Home.

En nuestro ejemplo, estos son los programas:

1	pick_workpiece	Recoger_pieza de trabajo
2	process_workpiece	Procesar_pieza de trabajo
3	place_workpiece	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-11: Programar FOR-ENDFOR, paso 1



Solo en el modo "Abrir" es posible introducir formularios inline lógicos.

2. Marque la célula de programa (1) en la que se debe añadir el bucle FOR-ENDFOR.
Pulsar el botón "Comandos" (2).



Fig. 22-12: Programar FOR-ENDFOR, paso 2

3. En el menú contextual seleccione "Lógica" (1) y "FOR-ENDFOR" (2).



Fig. 22-13: Programar FOR-ENDFOR, paso 3

4. En el formulario inline FOR ENDFOR añadido, pulsar el primer campo de entrada (1).



Fig. 22-14: Programar FOR-ENDFOR, paso 4

5. El bucle FOR-ENDFOR debe "memorizar" el número de ejecuciones después de cada ejecución del bucle. Para ello se necesita una "variable auxiliar" que se genera a través de este campo (1). El nombre puede seleccionarse libremente y se introduce a través del teclado de la pantalla visualizado (2).



Fig. 22-15: Programar FOR-ENDFOR, paso 5

6. El número de ejecuciones del bucle deseado se determina en el tercer campo de entrada (1,2). Se pueden definir diferentes modos de conteo de forma diferente a través de otros campos.



Fig. 22-16: Programar FOR-ENDFOR, paso 6

Después de la entrada, aceptar el bucle FOR-ENDFOR con el botón "Comando OK" (3).

7. El bucle definido en el paso 5 y la variable auxiliar se introducen ahora automáticamente por el sistema como sintaxis KRL en el programa. Para ello, el diálogo debe contestarse con "Sí" (1).



Fig. 22-17: Programar FOR-ENDFOR, paso 7

8. Marque la línea con el texto guardado en gris como comentario ";insert your code here" (1) y muestre el teclado de pantalla con la tecla "Clavija" (2).



Fig. 22-18: Programar FOR-ENDFOR, paso 8

9. Introduzca sucesivamente los módulos de programa que se van a activar a través del teclado de pantalla (1, 2).
La activación del módulo se identifica con () de acuerdo con el nombre del programa.

```
pick_workpiece()
process_workpiece()
```

```
place_workpiece()
```



Fig. 22-19: Programar FOR-ENDFOR, paso 9



Mediante el botón "Desarrollo del movimiento" también se pueden registrar y programar movimientos del robot.

Después de comprobar las entradas (2), cerrar el teclado de pantalla con el símbolo X naranja (3).

10. Cierre el programa abierto (1). Confirmar el diálogo La lista de datos se modificará irrevocablemente. Al cerrar el archivo SRC se ejecutará una copia de seguridad automática sin realizar la consulta correspondiente. Confirmar la pregunta ¿Desea continuar? con el botón "Sí" (2).



Fig. 22-20: Programar FOR-ENDFOR, paso 10

11. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-21: Programar FOR-ENDFOR, paso 11

Programar el bucle de conteo con el formulario inline

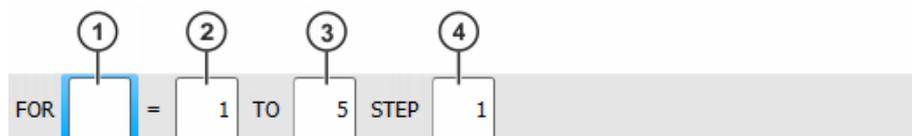


Fig. 22-22: Formulario inline FOR ... ENDFOR

Pos.	Descripción
1	Variable INT que cuenta las ejecuciones, la "Variable de recuento". Es posible la declaración previa de la variable, pero no es obligatoria. El valor de las variables puede emplearse en instrucciones dentro y fuera del bucle. Después de salir del bucle, la variable adopta el último valor tomado.
2	Valor inicial La variable de recuento avanza automáticamente con el valor de inicio.
3	Valor final Cuando se alcanza o se sobrepasa el valor final, el bucle ha finalizado.
4	Anchura de paso Después de cada ejecución del bucle la variable de recuento cambia automáticamente en la amplitud del paso. El valor puede ser negativo. Por defecto: 1. <ul style="list-style-type: none"> • Valor positivo: El bucle ha finalizado cuando la variable de recuento es mayor que el valor final. • Valor negativo: El bucle ha finalizado cuando la variable de recuento es inferior al valor final. El valor no puede ser cero ni una variable.

Podrá encontrar más información en la documentación del cliente.



Libro

KUKA System Software 8.6

Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

Programación KRL > Control de ejecución del programa

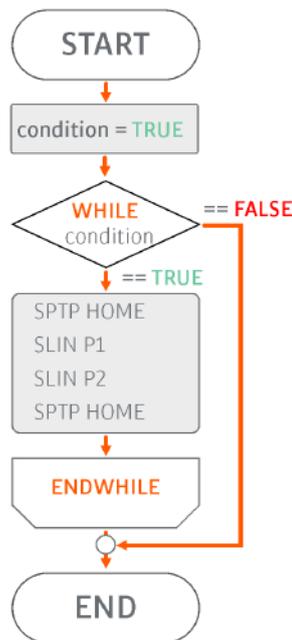


Consejo de búsqueda: Los bucles también pueden interrumpirse antes de tiempo. Simplemente eche un vistazo. Palabra clave "EXIT".

22.5 Conocer un bucle finito

Descripción

Un bucle WHILE-ENDWHILE se denomina también *bucle controlado desde arriba*. Es un bucle *finito* o *preliminar* al que se le va a comprobar la condición de ejecución antes de que se ejecute la sección de instrucción del bucle.



22.5.1 Programar un bucle finito con formularios inline

Descripción de la situación

Sus programas preparados que representan un ciclo de producción completo, deben ejecutarse de forma consecutiva. Al inicio de la ejecución del bucle, se comprueba si la entrada digital 20 es TRUE. En caso contrario, se omite la ejecución del bucle. Si se ha efectuado una ejecución del bucle, el bucle vuelve a saltar al principio y vuelve a comprobar la entrada digital 20.

En nuestro ejemplo, estos son los programas:

1	pick_workpiece	Recoger_pieza de trabajo
2	process_workpiece	Procesar_pieza de trabajo
3	place_workpiece	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-23: Programar WHILE-ENDWHILE, paso 1



Solo en el modo "Abrir" es posible añadir formularios inline de lógica.

2. Marque la célula de programa (1) en la que se debe añadir el bucle FOR-ENDFOR.
Pulse el botón "Comandos" (2).



Fig. 22-24: Programar WHILE-ENDWHILE, paso 2

3. En el menú contextual seleccione "Lógica" (1) y "REPEAT-UNTIL" (2).



Fig. 22-25: Programar WHILE-ENDWHILE, paso 3

4. En el formulario inline WHILE-ENDWHILE añadido, pulsar el tercer campo de entrada (1). En el menú desplegable seleccione el grupo ante el que debe reaccionarse. En el ejemplo es "IN" para entradas digitales.



Fig. 22-26: Programar WHILE-ENDWHILE, paso 4

5. Pulse el cuarto campo de entrada (1). Introduzca el número de canal deseado del grupo seleccionado mediante el teclado visualizado (2). Acepte las introducciones con el botón "Comando OK" (3).



Fig. 22-27: Programar WHILE-ENDWHILE, paso 5

6. Confirmar el diálogo La lista de datos se modificará irrevocablemente. Al cerrar el archivo SRC se ejecutará una copia de seguridad automática sin realizar la consulta correspondiente. Confirmar la pregunta ¿Desea continuar? con el botón "Sí" (1).



Fig. 22-28: Programar WHILE-ENDWHILE, paso 6

7. Marque la línea con el texto guardado en gris como comentario *"insert your code here"* (1) y muestre el teclado de pantalla con la tecla "Clavija" (2).



Fig. 22-29: Programar WHILE-ENDWHILE, paso 7

8. Introduzca sucesivamente los módulos de programa que se van a activar a través del teclado de pantalla (1,2).
La activación del módulo se identifica con () de acuerdo con el nombre del programa.

```
pick_workpiece()
process_workpiece()
place_workpiece()
```



Fig. 22-30: Programar WHILE-ENDWHILE, paso 8



Mediante el botón "Desarrollo del movimiento" también se pueden registrar y programar movimientos del robot.

Después de comprobar las entradas (2), cerrar el teclado de pantalla con el símbolo "X" naranja (3).

9. Cierre el programa abierto (1).



Fig. 22-31: Programar WHILE-ENDWHILE, paso 9

10. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-32: Programar WHILE-ENDWHILE, paso 10

Programar el bucle finito mediante formularios inline

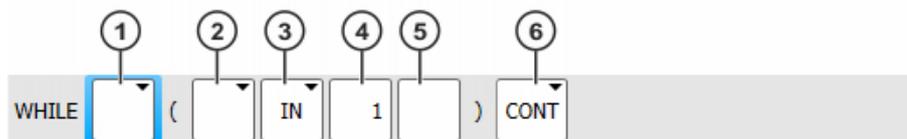


Fig. 22-33: Formulario inline WHILE ... ENDWHILE

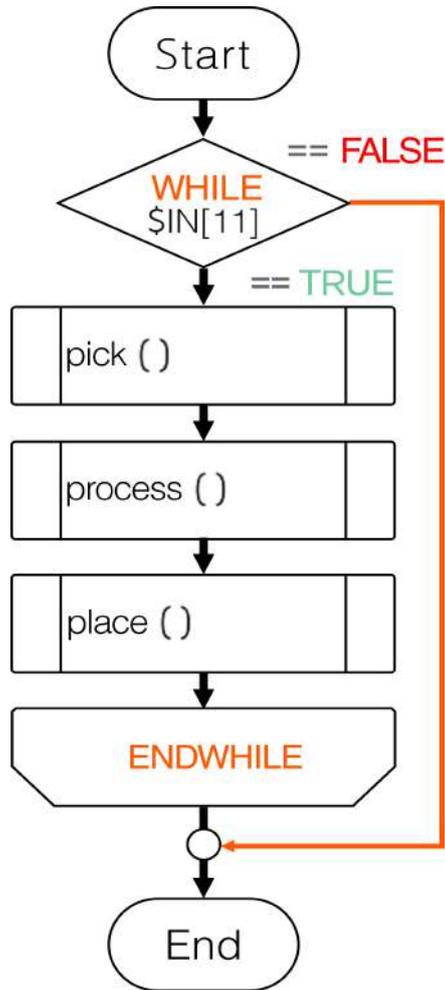
Pos.	Descripción
1	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío]
	<p>Agregar la combinación externa. El término se encuentra entre las expresiones entre paréntesis.</p> <p>Añadir el término deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
2	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío]
	<p>Agregar la combinación interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <p>Añadir el operador deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
3	<p>Señal que se evalúa. Selección por defecto:</p> <ul style="list-style-type: none"> • IN, OUT, CYCFLAG, TIMER o FLAG <p>También es posible programar otras señales. Para ello están disponibles los siguientes botones. Al tocar el botón se muestra el siguiente.</p> <ul style="list-style-type: none"> • Texto libre: Muestra un campo en el que se puede introducir el KRL. • Lista de usuario: Muestra una lista con variables definidas por el usuario. Requisito: Se ha configurado la lista. • Lista de sistema: Vuelve a mostrar la selección por defecto.
4	<ul style="list-style-type: none"> • Si se ha seleccionado una señal de la selección por defecto: Introducir el número. • Si se ha seleccionado una variable de la lista de usuario: Introducir el valor.
5	<p>Este campo solo está disponible cuando se ha seleccionado una señal de la selección por defecto. Si la señal ya tiene nombre, este se muestra.</p> <p>A partir del grupo de usuario experto: Puede introducirse un nombre o puede modificarse el nombre existente. Mientras no se haya guardado el formulario inline, pueden volver a restablecerse estas modificaciones pulsando Texto largo.</p>
6	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance

Restricciones en "Modificar"

Mediante el botón **Modificar** solo se puede modificar la condición. No es posible añadir ni eliminar CONT.

22.5.2 Ejercicio: Programar "WHILE-ENDWHILE"

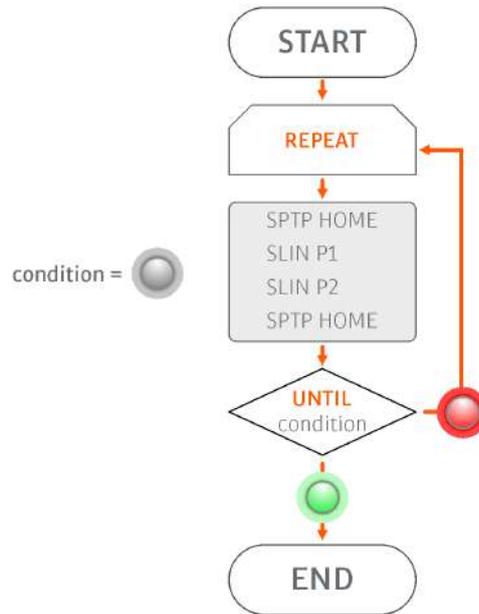
Enunciado



1. A diferencia del ejercicio anterior, la "liberación" a través de la entrada digital [11] no solo se debe comprobar una vez, sino también con cada nuevo paso de bucle.
Sustituya su bucle LOOP-ENDLOOP y la consulta WAIT-FOR con un bucle WHILE-ENDWHILE.
2. Pruebe el programa en los distintos modos de servicio T1, T2 y Automático.

22.6 Programar un bucle infinito

Descripción



- 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.
- Sintaxis

```
REPEAT
; Instrucción
UNTIL condition
```

- El bucle infinito se puede abandonar de forma inmediata mediante EXIT.

22.6.1 Programar un bucle infinito con formularios inline

Descripción de la situación

Sus programas preparados que representan un ciclo de producción completo, deben ejecutarse de forma consecutiva. El bucle debe ser ejecutado al menos una vez. Tras la finalización de la ejecución del bucle, se comprueba si la entrada digital 20 es TRUE. Si no es así, ya no se repetirá la ejecución del bucle.

En nuestro ejemplo, estos son los programas:

1	pick_workpiece	Recoger_pieza de trabajo
2	process_workpiece	Procesar_pieza de trabajo
3	place_workpiece	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-34: Programar REPEAT-UNTIL, paso 1



Solo en el modo "Abrir" es posible introducir formularios inline lógicos.

2. Marque la célula de programa (1) en la que se debe añadir el bucle FOR-ENDFOR.
Pulsar el botón "Comandos" (2).



Fig. 22-35: Programar REPEAT-UNTIL, paso 2

3. En el menú contextual seleccione "Lógica" (1) y "REPEAT-UNTIL" (2).



Fig. 22-36: Programar REPEAT-UNTIL, paso 3

4. En el formulario inline REPEAT UNTIL añadido, pulse el tercer campo de entrada (1). En el menú desplegable seleccione el grupo ante el que debe reaccionarse. En el ejemplo es "IN" para entradas digitales.



Fig. 22-37: Programar REPEAT-UNTIL, paso 4

5. Pulsar el cuarto campo de entrada (1). Introduzca el número de canal deseado del grupo seleccionado mediante el teclado visualizado (2). Acepte las introducciones con el botón "Comando OK" (3).



Fig. 22-38: Programar REPEAT-UNTIL, paso 5

6. Confirmar el diálogo La lista de datos se modificará irrevocablemente. Al cerrar el archivo SRC se ejecutará una copia de seguridad automática sin realizar la consulta correspondiente. Confirmar la pregunta ¿Desea continuar? con el botón "Sí" (2).



Fig. 22-39: Programar REPEAT-UNTIL, paso 6

7. Marque la línea con el texto guardado en gris como comentario ";insert your code here" (1) y muestre el teclado de pantalla con la tecla "Clavija" (2).



Fig. 22-40: Programar REPEAT-UNTIL, paso 7

8. Introduzca sucesivamente los módulos de programa que se van a activar a través del teclado de pantalla (1,2).
La activación del módulo se identifica con () de acuerdo con el nombre del programa.

```
pick_workpiece()  
process_workpiece()
```

```
place_workpiece()
```



Fig. 22-41: Programar REPEAT-UNTIL, paso 8

9. Compruebe las entradas (1).
- Cierre el programa abierto (2).



Fig. 22-42: Programar REPEAT-UNTIL, paso 9

Los errores tipográficos dan lugar a un programa no ejecutable.

10. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-43: Programar REPEAT-UNTIL, paso 10

Programar el bucle infinito mediante el formulario inline

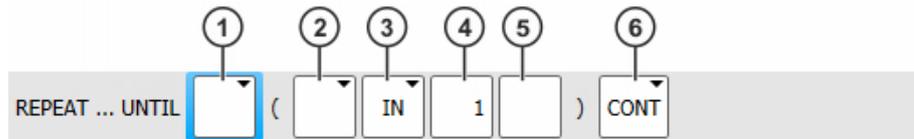


Fig. 22-44: Formulario inline REPEAT ... UNTIL

Pos.	Descripción
1	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío] <p>Agregar la combinación externa. El término se encuentra entre las expresiones entre paréntesis.</p> <p>Añadir el término deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
2	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío] <p>Agregar la combinación interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <p>Añadir el operador deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR

Pos.	Descripción
3	<p>Señal que se evalúa. Selección por defecto:</p> <ul style="list-style-type: none"> • IN, OUT, CYCFLAG, TIMER o FLAG <p>También es posible programar otras señales. Para ello están disponibles los siguientes botones. Al tocar el botón se muestra el siguiente.</p> <ul style="list-style-type: none"> • Texto libre: Muestra un campo en el que se puede introducir el KRL. • Lista de usuario: Muestra una lista con variables definidas por el usuario. Requisito: Se ha configurado la lista. • Lista de sistema: Vuelve a mostrar la selección por defecto.
4	<ul style="list-style-type: none"> • Si se ha seleccionado una señal de la selección por defecto: Introducir el número. • Si se ha seleccionado una variable de la lista de usuario: Introducir el valor.
5	<p>Este campo solo está disponible cuando se ha seleccionado una señal de la selección por defecto. Si la señal ya tiene nombre, este se muestra.</p> <p>A partir del grupo de usuario experto: Puede introducirse un nombre o puede modificarse el nombre existente. Mientras no se haya guardado el formulario inline, pueden volver a restablecerse estas modificaciones pulsando Texto largo.</p>
6	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance

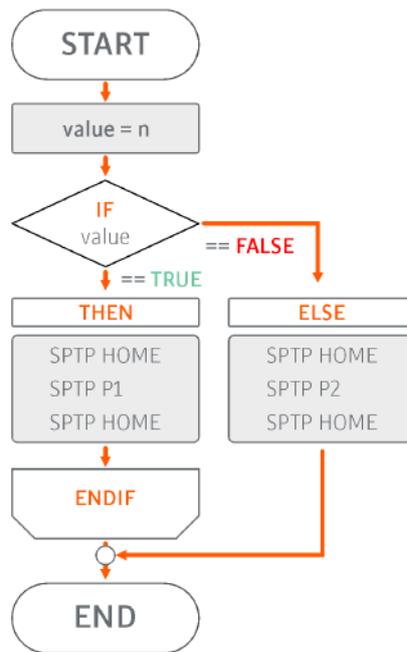
Restricciones en "Modificar"

Mediante el botón **Modificar** solo se puede modificar la condición. No es posible añadir ni eliminar CONT.

22.7 Conocer consultas o ramificaciones (IF-THEN-ELSE)

Descripción

El uso de ramificaciones permite que secciones del programa se ejecuten únicamente si se da una condición determinada. 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.

Variantes 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 (`ENDIF`).
- Es posible intercalar varias consultas `IF` (*ramificación múltiple*): las consultas se ejecutan por orden y se comprueba si se cumple una condición.

22.7.1 Programar consultas o ramificaciones (IF-THEN-ELSE) con formularios inline

Descripción de la situación

Ha preparado programas que procesan dos piezas similares, pero diferentes. El propio proceso de mecanizado es idéntico.

Antes de cada ciclo se debe consultar qué pieza debe ser mecanizada a continuación.

Una consulta `IF` decide a través de una entrada digital, qué programa se inicia para la recogida del nuevo componente.

En nuestro ejemplo, estos son los programas:

1	<code>pick_workpiece_01</code>	<code>Werkstück_holen_01</code>
2	<code>pick_workpiece_02</code>	<code>Werkstück_holen_02</code>
3	<code>process_workpiece</code>	Procesar_pieza de trabajo
4	<code>place_workpiece</code>	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-45: Programar IF-THEN-ELSE, paso 1



Solo en el modo "Abrir" es posible introducir formularios inline lógicos.

2. Marque la célula de programa (1) en la que se debe añadir el bucle FOR-ENDFOR.
Pulsar el botón "Comandos" (2).



Fig. 22-46: Programar IF-THEN-ELSE, paso 2

3. En el menú contextual seleccione "Lógica" (1) y "IF-THEN" (2).



Fig. 22-47: Programar IF-THEN-ELSE, paso 3

4. En el formulario inline IF THEN añadido, pulse el primer campo de entrada (1).

Si se desea una ramificación "ELSE", en este punto debe ampliarse el formulario inline con la el botón "Añadir término ELSE" (2).



Fig. 22-48: Programar IF-THEN-ELSE, paso 4

AVISO

La ampliación posterior del término ELSE ya no será posible en el curso posterior a través del formulario inline.



Desde el menú desplegable se puede negar la siguiente impresión entre paréntesis con la consulta IF.

- En el formulario inline IF THEN añadido, pulse el tercer campo de entrada (1). En el menú desplegable seleccione el grupo ante el que debe reaccionarse.

En el ejemplo es "IN" para entradas digitales.



Fig. 22-49: Programar IF-THEN-ELSE, paso 5

- Pulse el cuarto campo de entrada (1). Introduzca el número de canal deseado del grupo seleccionado mediante el teclado visualizado (2).



Fig. 22-50: Programar IF-THEN-ELSE, paso 6

7. Acepte las introducciones con el botón "Comando OK" (1).
 Confirmar el diálogo La lista de datos se modificará irrevocablemente. Al cerrar el archivo SRC se ejecutará una copia de seguridad automática sin realizar la consulta correspondiente. Confirmar la pregunta ¿Desea continuar? con el botón "Sí" (2).



Fig. 22-51: Programar IF-THEN-ELSE, paso 7

8. En IF-TERM, marque la línea con el texto guardado en gris como comentario ";insert your code here" (1) y muestre el teclado de pantalla con la tecla "Clavija" (2).



Fig. 22-52: Programar IF-THEN-ELSE, paso 8

- Introduzca sucesivamente los módulos de programa que se van a activar a través del teclado de pantalla (1,2).

La activación del módulo se identifica con () de acuerdo con el nombre del programa.

```
process_workpiece_01()
```



Fig. 22-53: Programar IF-THEN-ELSE, paso 9



Los movimientos se pueden insertar a través del botón del mismo nombre.

- En ELSE-TERM, marque la línea con el texto guardado en gris como comentario "*;insert your code here*" (1).

Introduzca sucesivamente los módulos de programa que se van a activar a través del teclado de pantalla (1,2).

Cerrar el teclado en la pantalla con el botón X naranja (3).



Fig. 22-54: Programar IF-THEN-ELSE, paso 10

11. Cerrar el teclado en la pantalla con el botón X naranja (1).



Fig. 22-55: Programar IF-THEN-ELSE, paso 11

12. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-56: Programar IF-THEN-ELSE, paso 12

Programar consultas o ramificaciones con un formulario inline

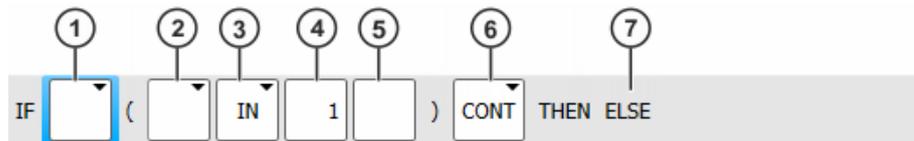


Fig. 22-57: Formulario inline IF ... THEN

Pos.	Descripción
1	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío] <p>Agregar la combinación externa. El término se encuentra entre las expresiones entre paréntesis.</p> <p>Añadir el término deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR
2	<ul style="list-style-type: none"> • NOT: Agregar NOT. • [vacío] <p>Agregar la combinación interna. El operador se ubica dentro de una expresión colocada entre paréntesis.</p> <p>Añadir el operador deseado mediante el botón correspondiente:</p> <ul style="list-style-type: none"> • AND, OR o EXOR

Pos.	Descripción
3	<p>Señal que se evalúa. Selección por defecto:</p> <ul style="list-style-type: none"> • IN, OUT, CYCFLAG, TIMER o FLAG <p>También es posible programar otras señales. Para ello están disponibles los siguientes botones. Al tocar el botón se muestra el siguiente.</p> <ul style="list-style-type: none"> • Texto libre: Muestra un campo en el que se puede introducir el KRL. • Lista de usuario: Muestra una lista con variables definidas por el usuario. Requisito: Se ha configurado la lista. • Lista de sistema: Vuelve a mostrar la selección por defecto.
4	<ul style="list-style-type: none"> • Si se ha seleccionado una señal de la selección por defecto: Introducir el número. • Si se ha seleccionado una variable de la lista de usuario: Introducir el valor.
5	<p>Este campo solo está disponible cuando se ha seleccionado una señal de la selección por defecto. Si la señal ya tiene nombre, este se muestra.</p> <p>A partir del grupo de usuario experto: Puede introducirse un nombre o puede modificarse el nombre existente. Mientras no se haya guardado el formulario inline, pueden volver a restablecerse estas modificaciones pulsando Texto largo.</p>
6	<ul style="list-style-type: none"> • CONT: procesamiento en el movimiento de avance • [vacío]: procesamiento con parada del movimiento de avance
7	<p>Añadir ELSE mediante el botón correspondiente.</p> <p>El foco debe estar en el campo 1 o 2.</p>

Restricciones en "Modificar"

Mediante el botón **Modificar** solo se puede modificar la condición. No es posible añadir ni eliminar ELSE o CONT.

22.8 Conocer el distribuidor (SWITCH-CASE)

Descripción

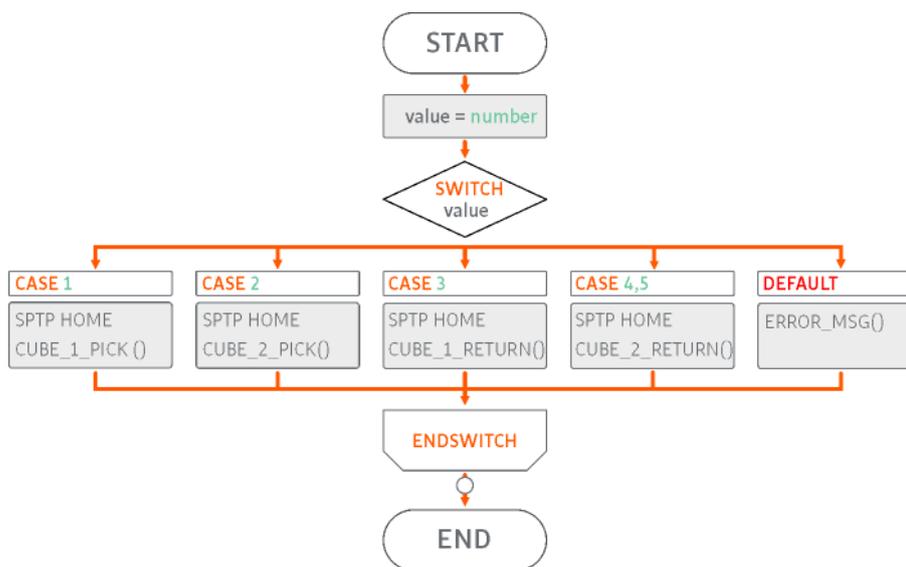


Fig. 22-58: Plano de desarrollo del programa: Distribuidor SWITCH - CASE

- Si se desea diferenciar entre varios casos y ejecutar diferentes acciones para cada caso, se puede lograr con una instrucción SWITCH-CASE.
- Una ramificación SWITCH-CASE es un *distribuidor* o una *ramificación múltiple* y sirve para la diferenciación de casos.
- Una variable transmitida a la instrucción SWITCH se utiliza como conmutador 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.

22.8.1 Programar distribuidores (SWITCH - CASE) con formularios inline

Descripción de la situación

Ha preparado programas que procesan dos piezas similares, pero diferentes. El propio proceso de mecanizado es idéntico.

Antes de cada ciclo se debe consultar qué pieza debe ser mecanizada a continuación.

Un distribuidor decide por medio de un "número de programa" (variable INT) qué programa se inicia para la recogida del nuevo componente.

En nuestro ejemplo, estos son los programas:

1	pick_workpiece_01	Werkstück_holen_01
2	pick_workpiece_02	Werkstück_holen_02
3	process_workpiece	Procesar_pieza de trabajo
4	place_workpiece	Depositar_pieza de trabajo

Procedimiento

1. Marque el módulo de programa que debe ampliarse (1) y ábralo con el botón con el mismo nombre "Abrir" (2).



Fig. 22-59: Programar SWITCH-CASE, paso 1



Solo en el modo "Abrir" es posible introducir formularios inline lógicos.

2. Marque la célula de programa (1) en la que se debe añadir el distribuidor SWITCH-CASE.
Pulsar el botón "Comandos" (2).



Fig. 22-60: Programar SWITCH-CASE, paso 2

3. En el menú contextual seleccione "Lógica" (1) y "SWITCH-CASE" (2).



Fig. 22-61: Programar SWITCH-CASE, paso 3

4. En el formulario inline SWITCH-CASE añadido, pulsar cualquier campo de entrada. Añadir (2) o eliminar (2) un "término CASE" a través del botón del mismo nombre.

El "término Default" se genera automáticamente.



Fig. 22-62: Programar SWITCH-CASE, paso 4

5. Pulsar el primer campo de entrada (1). Introducir la variable que debe evaluarse a través del teclado de pantalla visualizado (2).



Fig. 22-63: Programar SWITCH-CASE, paso 5



Si la variable aún no está disponible, se declarará a continuación en el programa abierto.

6. Para cada término CASE (1,2,3), introducir el número de programa correspondiente a través del teclado de la pantalla.
Aceptar las introducciones con el botón "Comandos OK" (4).



Fig. 22-64: Programar SWITCH-CASE, paso 6

7. Confirmar el diálogo La lista de datos se modificará irrevocablemente. Al cerrar el archivo SRC se ejecutará una copia de seguridad automática sin realizar

la consulta correspondiente. Confirmar la pregunta ¿Desea continuar? con el botón Si" (2).



Fig. 22-65: Programar SWITCH-CASE, paso 7

8. En el diálogo ¿Desea declarar una variable con el nombre "Value"? seleccionar el tipo de datos deseado. En nuestro ejemplo, el tipo de datos completo INT (entero) (1).



Fig. 22-66: Programar SWITCH-CASE, paso 8

9. El distribuidor SWITCH-CASE se ha añadido a la parte del programa marcada. Mostrar el teclado de la pantalla con la tecla "Clavija" (2).



Fig. 22-67: Programar SWITCH-CASE, paso 9

10. En cada término CASE individual, marcar el texto *"insert your code here"* (1,2,3) guardado con el comentario e introducir con el teclado de la pantalla los programas, movimientos o comandos que se deben activar.



Fig. 22-68: Programar SWITCH-CASE, paso 10

11. Si nuestro "número de programa" no corresponde a un término CASE predefinido, se procesará el término DEFAULT.
 Marcar también aquí el texto *"insert your code here"* guardado con el comentario.

En nuestro ejemplo, el robot debe retroceder a la posición HOME. El formulario inline correspondiente se inserta como de costumbre mediante el botón "Desarrollo del movimiento" (2).



Fig. 22-69: Programar SWITCH-CASE, paso 11

12. Adaptar el formulario inline visualizado (1) y aceptar con el botón "Comando OK" (2).



Fig. 22-70: Programar SWITCH-CASE, paso 12

13. En el formulario inline comprobar el distribuidor SWITCH-CASE (1) y cerrar el teclado en pantalla con el botón X naranja (2).



Fig. 22-71: Programar SWITCH-CASE, paso 13

14. Comprobar la lógica y la presencia de errores tipográficos en el distribuidor SWITCH-CASE (1) y cerrar el programa a través del botón X naranja (2).



Fig. 22-72: Programar SWITCH-CASE, paso 14

15. Para realizar pruebas, marque el programa modificado en el navegador y selecciónelo con el botón del mismo nombre "Seleccionar".



Fig. 22-73: Programar SWITCH-CASE, paso 15

Programar los distribuidores con el formulario inline (SWITCH-CASE)

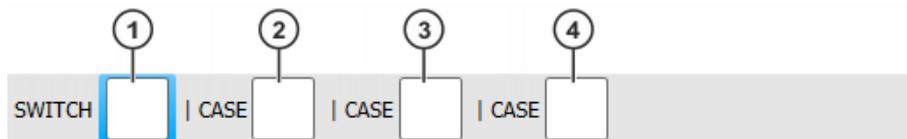


Fig. 22-74: Formulario inline SWITCH ... CASE

Pos.	Descripción
1	Variable para el criterio de selección <ul style="list-style-type: none"> Es posible declarar previamente una variable INT o CHAR, aunque no es obligatorio. Debe declararse manualmente de forma previa o posteriormente una variable ENUM.
2	Identificación para el bloque de instrucciones El tipo de datos de la identificación debe coincidir con el tipo de datos de las variables (pos. 1). Un bloque de instrucciones puede tener todas las identificaciones que se desee. Deben separarse mediante comas las diferentes identificaciones en un campo CASE.
3	Como pos. 2
4	Los campos CASE disponibles deben rellenarse con identificaciones o eliminarse. Mediante los botones correspondientes es posible eliminar o añadir campos CASE.

Restricciones en "Modificar"

A través del botón **Modificar** no es posible añadir ni eliminar ningún CASE.

Posibles modificaciones:

- Cuando el cursor se encuentra en la línea con SWITCH, es posible modificar la variable.
- Si el cursor se encuentra en una línea con CASE, es posible modificar el valor.

22.9 Preguntas: Técnicas de bucles

¡Ahora es su turno!



En el distribuidor SWITCH/CASE existe la instrucción "DEFAULT". ¿Qué función tiene esta instrucción "DEFAULT"?

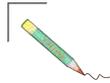


Fig. 22-75: Campo de respuesta

¿Con qué instrucción se puede ajustar la anchura de paso en el bucle FOR?

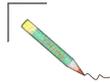


Fig. 22-76: Campo de respuesta

¿Qué bucles se pueden abandonar con la instrucción "EXIT"?

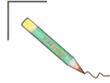


Fig. 22-77: Campo de respuesta

¿Qué parte se puede omitir en una ramificación ? a. IF b. THEN c. ELSE d. ENDIF



Fig. 22-78: Campo de respuesta

23 Operar la estación con un PLC de orden superior

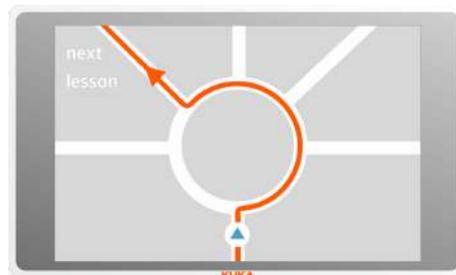
23.1 Navegador del curso

¿Qué requisitos aporta usted?



- Está capacitado para manejar y desplazar robots
- Está capacitado para seleccionar programas y probarlos en los modos de servicio T1, T2 y AUT.
- Está capacitado para abrir programas y realizar modificaciones, también en KRL.

¿Qué le espera en esta unidad de aprendizaje?



Operar la estación con un PLC de orden superior

- Conocerá la interfaz de comunicación entre el PLC y el robot.
- Podrá adaptar el programa de organización para la activación de programas.

23.2 Conocer el inicio del programa a través de un PLC de orden superior

Robots en red

En una línea de producción, muchos robots trabajan "mano a mano". Para que esto funcione sin problemas, un ordenador principal de orden superior o un PLC coordina la secuencia. Para la comunicación entre el nivel de dirección y el robot están disponibles el modo de servicio "Automático externo" y el programa "SPS.SUB".

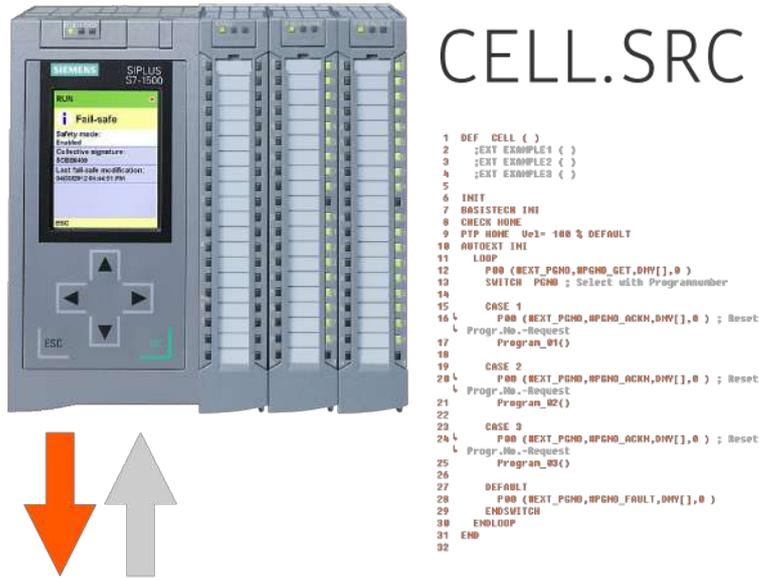


Fig. 23-1: Conexión de PLC

23.3 Conocer el programa de organización Cell.src

Descripción

Para administrar los números de programas transmitidos por el PLC, se utiliza el programa organizativo Cell.src.

Se encuentra siempre en la carpeta "KRC:\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

Operar la estación con un PLC de orden superior

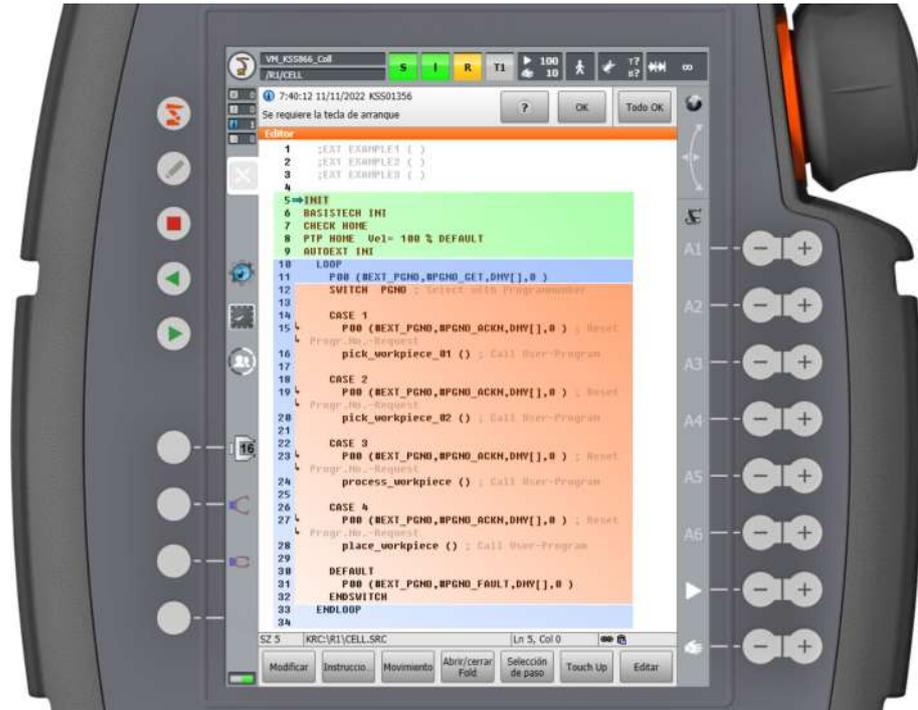


Fig. 23-2: Programa CELL

verde	<p>Inicialización y posición inicial</p> <ul style="list-style-type: none"> • Inicialización de los parámetros básicos • Comprobación de la posición del robot respecto a la posición inicial • Inicialización de la interfaz "Automático externo" • Desplazamiento hacia la "posición inicial"
azul	<p>Bucle infinito:</p> <ul style="list-style-type: none"> • Consulta del número de programa a través del módulo "P00" • Acceso al bucle de selección con el número de programa facilitado.
naranja	<p>Bucle de selección con número del programa</p> <ul style="list-style-type: none"> • El salto en la ramificación correspondiente ("CASE") se realiza conforme al número de programa (determinado en la variable "PGNO"). • Se ejecuta el programa de robot especificado en la ramificación. • Los números de programa no válidos provocan que se salte a la ramificación "Por defecto". • Una vez finalizada la ejecución, se repite el bucle.



Las rutas CASE pueden ampliarse copiando las líneas de programa e introduciéndolas en el SWITCH-CSE existente. El número de programa y los programas que se deben activarse pueden adaptarse individualmente.

Procedimiento

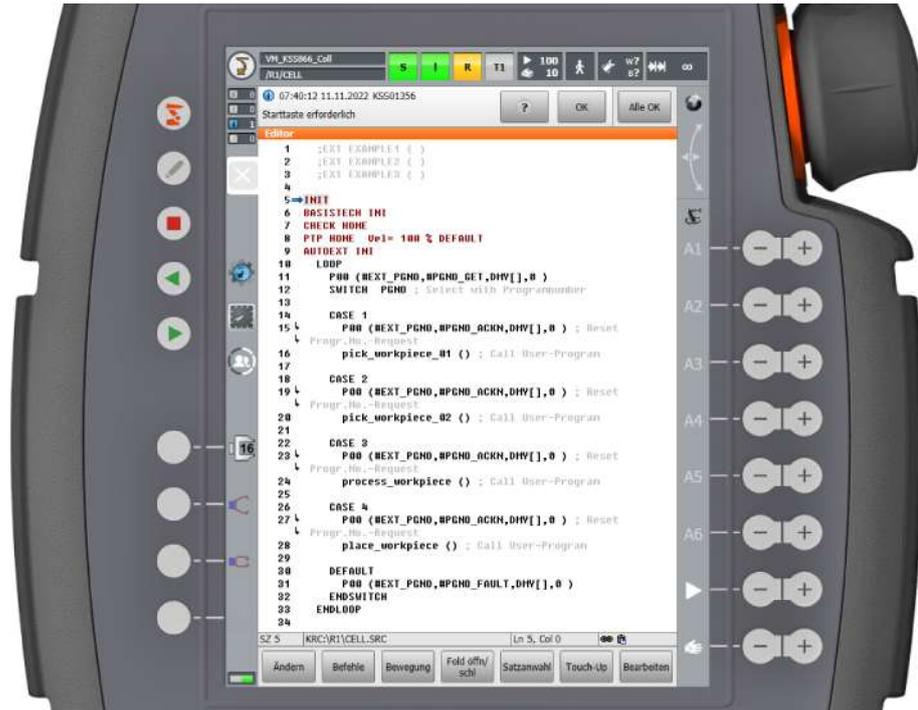


Fig. 23-3: Cell.src

Encontrará más información sobre este apartado de aprendizaje en su manual.



Libro

Programación de robots 1

KUKA System Software 8.6

Referencia de formación

Capítulo/apartado

Activación del programa a través de un PLC de orden superior

Podrá encontrar más información en la documentación del cliente.



Libro

KUKA System Software 8.6

Instrucciones de servicio y programación para los integradores de sistemas

Filtro de búsqueda

Documentación > Instrucciones de manejo y programación

Capítulo/apartado

Configuración > Configurar Automático Externo

23.4 Ejercicio: Configurar la interfaz externa de automática

¡Ahora es su turno!



Enunciado



Fig. 23-4: Pupitre College 4.0

1. Abra y amplíe su programa Cell con módulos de programa probados que haya creado en el transcurso anterior del curso.
2. Simular la funcionalidad de la unidad de control del PLC mediante el pulsador.
 (>>> "PAP para la probar la interfaz externa automático en la celda de formación" Página 414)

Operar la estación con un PLC de orden superior

23.5 Comunicación entre la unidad de control y el PLC en el ejemplo de la celda de formación

PAP para la probar la interfaz externa automático en la celda de formación

La secuencia de la comunicación entre el robot y el PLC se establece de forma fija a través de la interfaz externa automática. El programa correspondiente se representa e se inicia en el PLC.

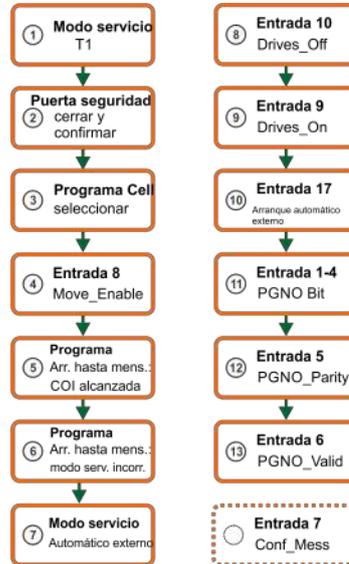


Fig. 23-5: PAP - Automático externo, celda College

Panel de mando College

Para entender qué entradas y pasos de programa son necesarios, puede aceptar el rol del PLC a través del panel de mando College. Mediante LEDs se muestra el estado actual del robot y a través de los interruptores "digitales" puede enviar entradas y comandos al robot.



Fig. 23-6: Pupitre College 4.0

Panel de mando basado en interruptores

En parte, encontrará el panel de mando basado en interruptores en las celdas de formación College.

Sin embargo, la comunicación desde y hacia el PLC de orden superior se realiza a través de las mismas entradas y salidas.

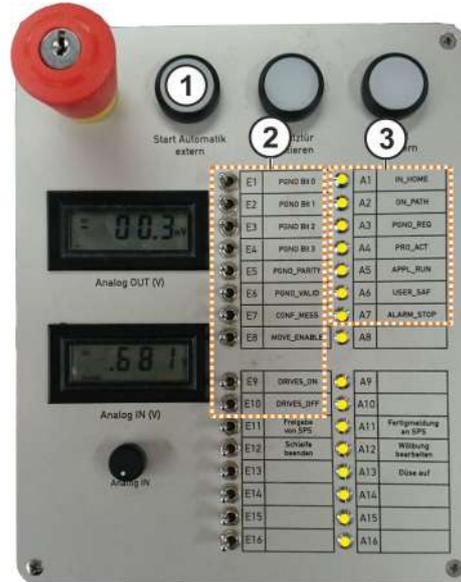


Fig. 23-7: Panel de mando College

Pos.	Señal	Descripción
1	E17	Arranque automático externo
Entradas "PLC"	E1	PGNO Bit 0
	E2	PGNO Bit 1
	E3	PGNO Bit 2
	E4	PGNO Bit 3
	E5	PGNO_PARITY
	E6	PGNO_VALID
	E7	CONF_MESS
	E8	MOVE_ENABLE
	E9	DRIVES_ON
	E10	DRIVES_OFF
Salidas "PLC"	A1	IN_HOME
	A2	ON_PATH
	A3	PGNO_REQ
	A4	PRO_ACT
	A5	APPL_RUN
	A6	USER_SAF
	A7	ALARM_STOP

Entradas

Para poder simular PLC con el panel de mando, deben realizarse las siguientes modificaciones (en negrita) en la configuración automática externa.

Operar la estación con un PLC de orden superior

Denominación	Nombre	Valor
Tipo de nº de programa	PGNO_TYPE	1
Número de programa reflejado	REFLECT_PROG_NR	0
Ancho de bit del nº de programa	PGNO_LENGTH	4
Primer bit del n.º de programa	PGNO_FBIT	1
Bit de paridad	PGNO_PARITY	5
N.º programa válido	PGNO_VALID	6
Inicio del programa	\$EXT_START	17
Liberación de movimiento	\$MOVE_ENABLE	8
Confirmación de error	\$CONF_MESS	7
Accionamientos desconectados (invertido)	\$DRIVES_OFF	10
Accionamientos ON	\$DRIVES_ON	9
Activar interfaz	\$I_O_ACT	1025

Salidas

Para poder simular PLC con el panel de mando, deben realizarse las siguientes modificaciones (en negrita) en la configuración automática externa.

Denominación	Nombre	Valor
Unidad de control preparada	\$RC_RDY1	137
Circuito de parada de emergencia cerrado	\$ALARM_STOP	7
Protección del operario cerrada	\$USER_SAF	6
Accionamientos preparados	\$PERI_RDY	1012
Robot ajustado	\$ROB_CAL	1001
Interfaz activa	\$I_O_ACTCONF	140
Fallo colectivo	\$STOPMESS	1010
Primer bit para la reflexión imagen del programa	\$PGNO_FBIT_REFL	999
PARADA DE EMERGENCIA interna	\$ALARM_STOP	1002

Denominación	Nombre	Valor
Programa activo	\$PRO_ACT	4
Requisito n.º programa	PGNO_REQ	3
Aplicación en marcha	APPL_RUN	5
Movimiento de programa activo	\$PRO_MOVE	1022

Denominación	Nombre	Valor
En posición inicial	\$IN_HOME	1
1.ª Posición Home	\$IN_HOME1	FALSE
2.ª Posición Home	\$IN_HOME2	FALSE
3.ª Posición Home	\$IN_HOME3	FALSE
4.ª Posición Home	\$IN_HOME4	FALSE
5.ª Posición Home	\$IN_HOME5	FALSE

Denominación	Nombre	Valor
Robot en la trayectoria	\$ON_PATH	2
Robot cerca de la trayectoria	\$NEAR_POSRET	147
Robot no está en movimiento	\$ROB_STOPPED	1023

Denominación	Nombre	Valor
Funcionamiento prueba 1	\$T1	21
Funcionamiento prueba 2	\$T2	22
Modo de servicio automático	\$AUT	23
Servicio automático externo	\$EXT	24

23.6 Preguntas: Trabajos en una unidad de control superior

Lo que se debe saber tras el ejercicio:



¿Cuál es el requisito previo para la comunicación con un PLC?



Fig. 23-8: Campo de respuesta

¿Mediante qué subprograma global se transmite el número de programa desde el PLC?

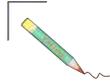


Fig. 23-9: Campo de respuesta

¿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?

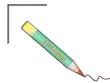


Fig. 23-10: Campo de respuesta

24 Se presenta el entorno de desarrollo de WorkVisual

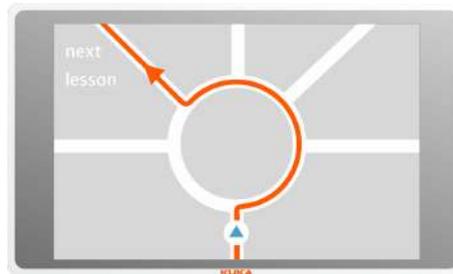
24.1 Navegador del curso

¿Qué requisitos aporta usted?



- Sabe cómo es la estructura técnica de un sistema de robot.
- Dispone de los primeros conocimientos básicos en el manejo y la programación.

¿Qué le espera en esta unidad de aprendizaje?

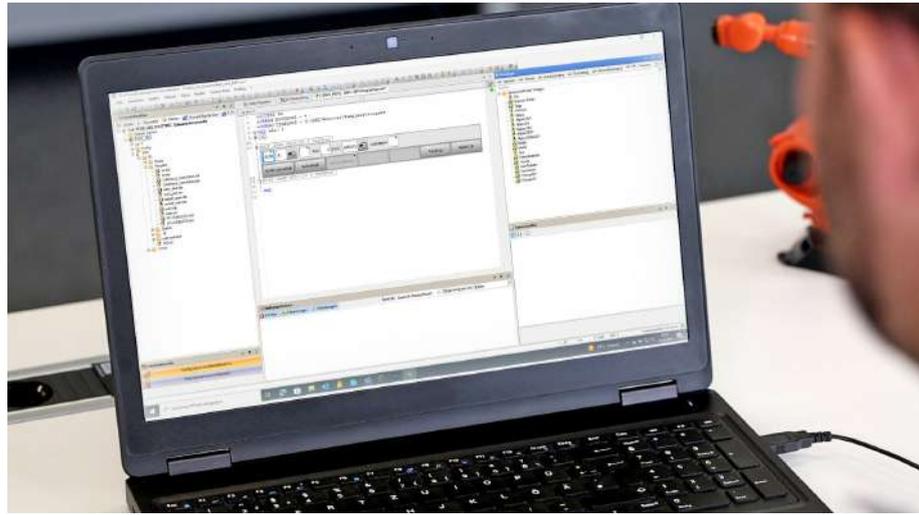


- Se le mostrará el entorno de desarrollo de WorkVisual.
- El formador les mostrará en una breve demostración en directo lo sencillo que pueden programarse, configurarse o incluso diagnosticarse los robots.
- Disfrute de su seminario de continuación en el KUKA-College y aplique allí WorkVisual con éxito.

24.2 Demostración en directo del entorno de desarrollo de WorkVisual

Programar con WorkVisual y más ...

Se le presentará el entorno de desarrollo de WorkVisual y las posibilidades de aplicación relacionadas con ello.



Podrá encontrar más información en la documentación del cliente.



Libro

KUKA.WorkVisual

Filtro de búsqueda

Tipo de información: Documentación > Instrucciones de manejo y programación

Capítulo/apartado

25 Anexo

25.1 Vista general de los sistemas de herramientas y base

Descripción

- El siguiente listado de herramientas y bases se utiliza para la celda de formación modular y móvil de KUKA College.
- Pueden variar en función del equipamiento y las instalaciones de formación.

N.º	Tool	N.º	Base
0	\$NULLFRAME	0	\$NULLFRAME
1		1	
2	Garra_nueva	2	Palet_azul
3	Clavija_grande	3	Trayectoria_naranja
4	Placa_Wst	4	Tobera_externa
5		5	Palet_indirecto
6		6	Pieza de soldadura_indirecta
7		7	Bloque spline_largo
8		8	
9		9	
10		10	
11		11	
12	D_Crimping_Roll	12	
13	D_Panel_Ext_3D	13	D_Blue_Base_Pallet
14	D_Gripper	14	D_Orange_Base_Spline
15	D_Panel	15	Red_Base_Straight
16	D_Pen165	16	D_Glue_Nozzle
		17	D_Pos_A_3D
		18	D_Pos_B_3D
		19	D_Crimping_Roll
		20	D_Spline
		...	

25.2 KUKA College: Datos de carga/valores de offset para herramientas/Herramientas

Descripción

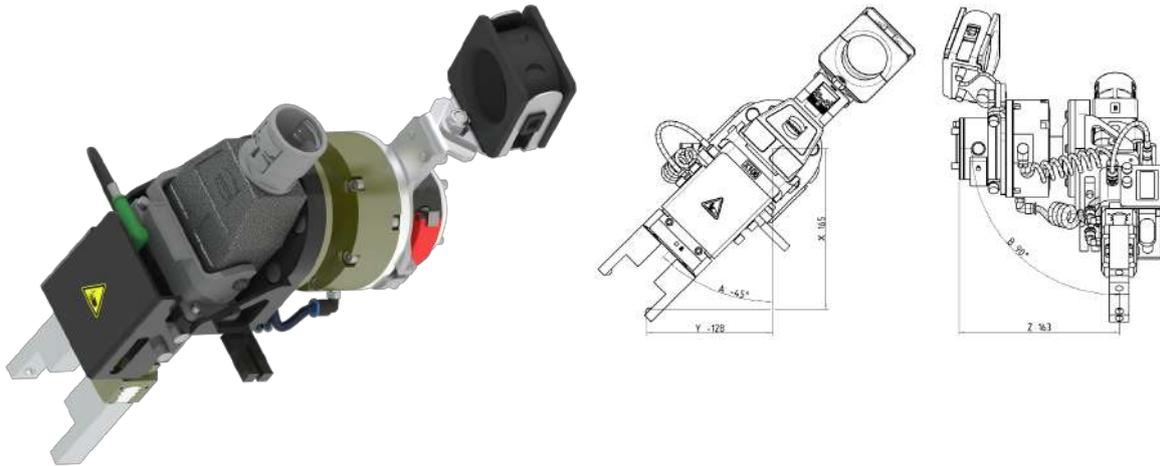
- En función del lugar del curso de formación, las células de robot utilizadas pueden ser diferentes en tamaño a los componentes utilizados.
- En especial, al medir las herramientas y piezas es importante utilizar los datos de carga correctos.
- En los siguientes apartados se enumeran todas las variantes utilizadas de los KUKA Colleges.

- En el cierre de redacción no todos los datos estaban disponibles.
- Introduzca los datos de carga necesarios para la formación en la tabla prevista y utilice estos datos para sus ejercicios. Los recibirá de su formador/a.

25.2.1 Garra en KR 16, ancho de agarre 50 mm

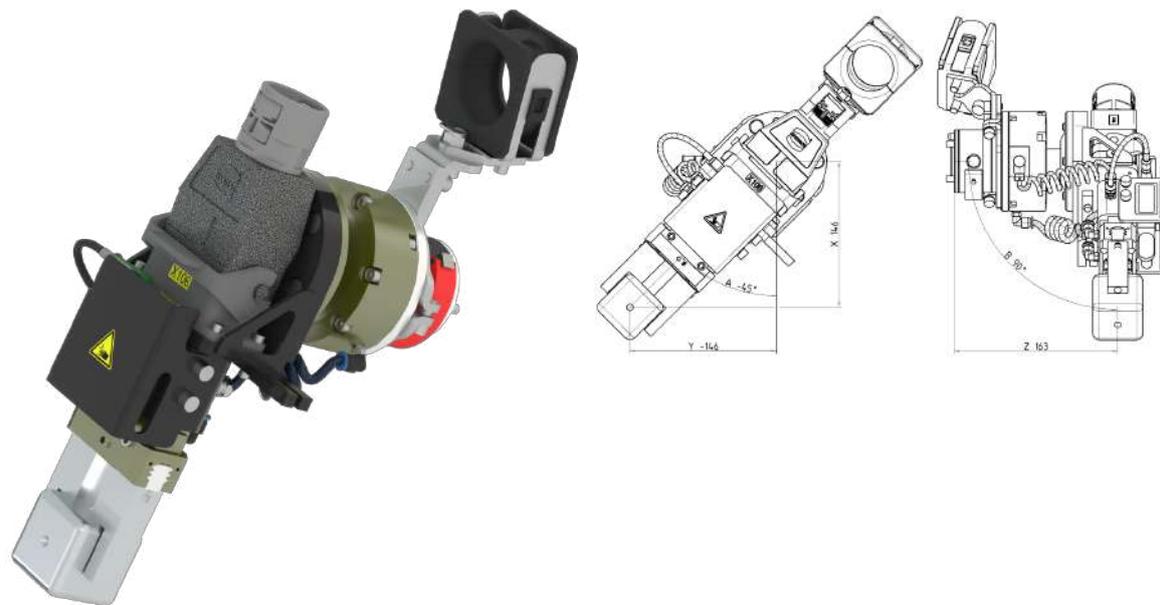
Descripción

Garra vacía



Masa [kg]	6,78				
X [mm]	13,62	A		I_x [kgm ²]	0,04
Y [mm]	-13,10	B		I_y [kgm ²]	0,04
Z [mm]	108,13	C		I_z [kgm ²]	0,04

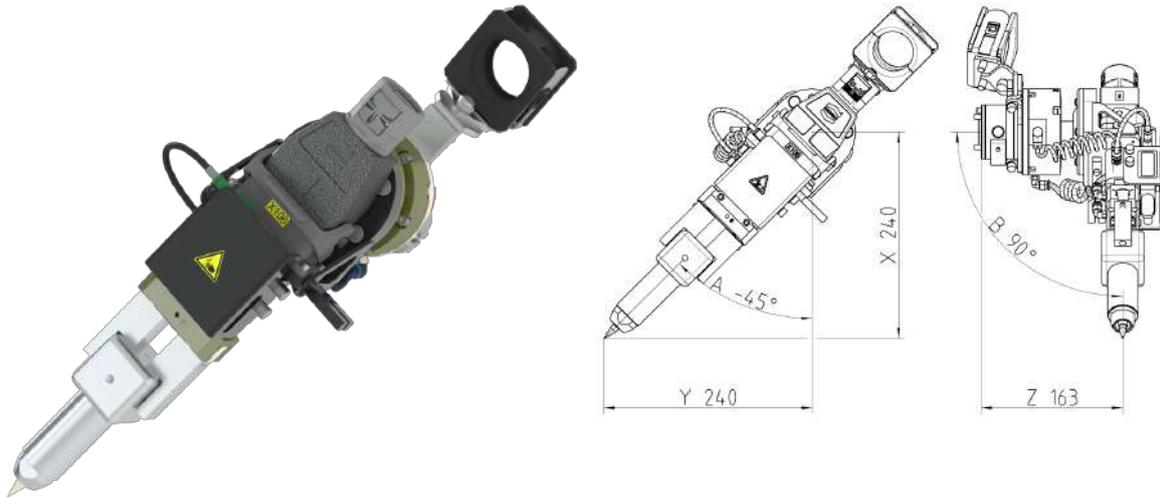
Garra con cubo



Masa [kg]	7.10				
X [mm]	19,69	A		I_x [kgm ²]	0,05
Y [mm]	-19,20	B		I_y [kgm ²]	0,05

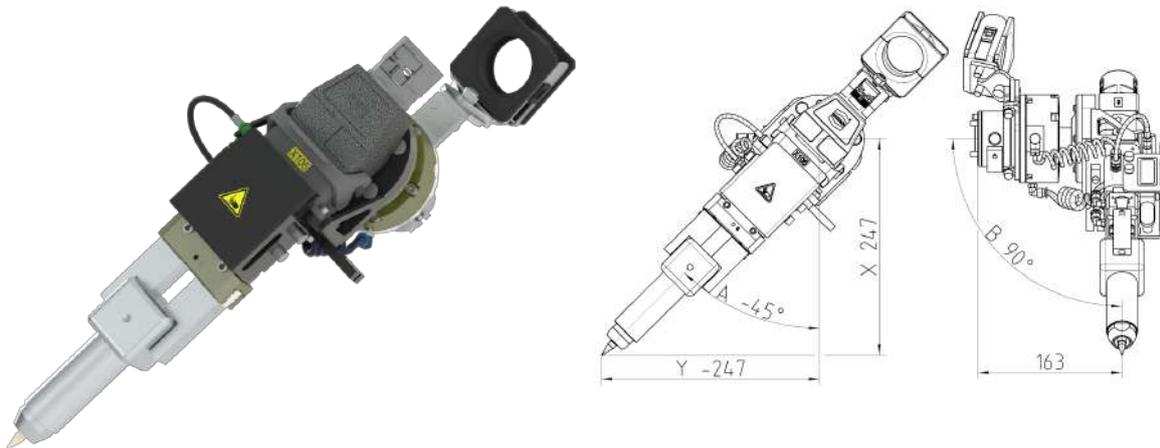
Z [mm]	110,64	C		lz [kgm ²]	0,05
--------	--------	---	--	------------------------	------

Garra con clavija corta



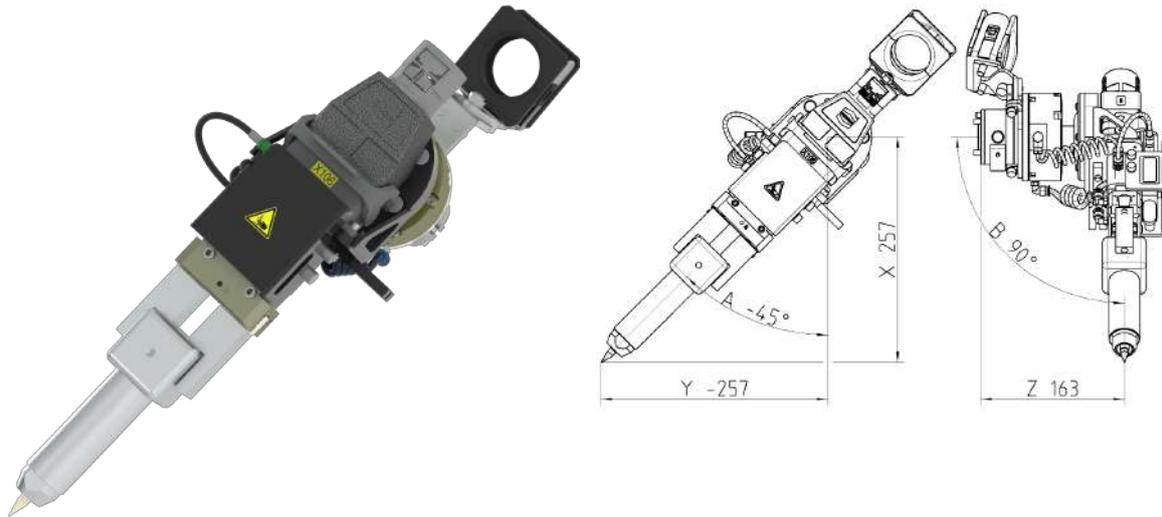
Masa [kg]	7.26				
X [mm]	23.49	A		lx [kgm ²]	0.06
Y [mm]	-23.01	B		ly [kgm ²]	0.06
Z [mm]	111.78	C		lz [kgm ²]	0.06

Garra con clavija medio larga



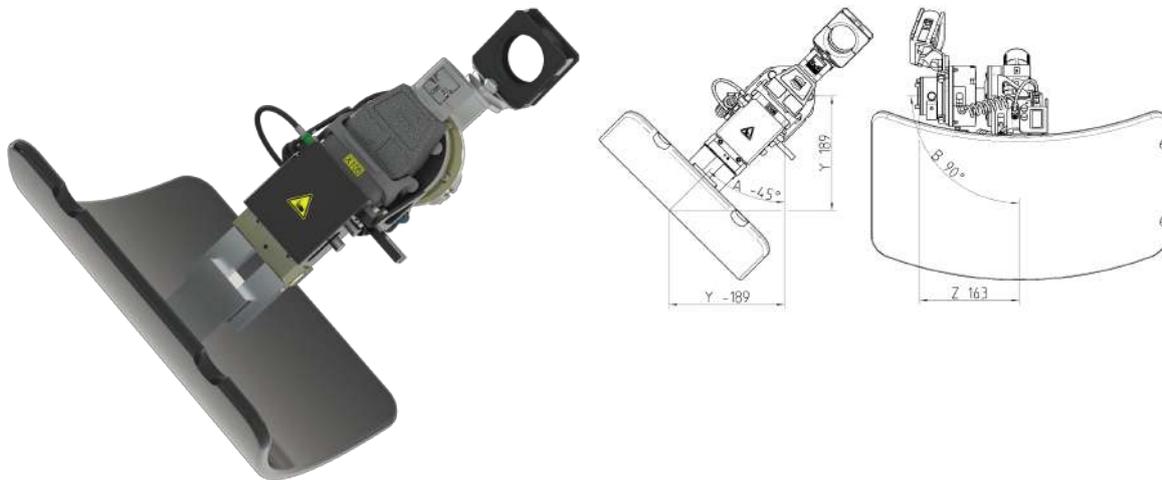
Masa [kg]	7.28				
X [mm]	24.07	A		lx [kgm ²]	0.06
Y [mm]	-23.59	B		ly [kgm ²]	0.06
Z [mm]	111.93	C		lz [kgm ²]	0.05

Garra con clavija larga



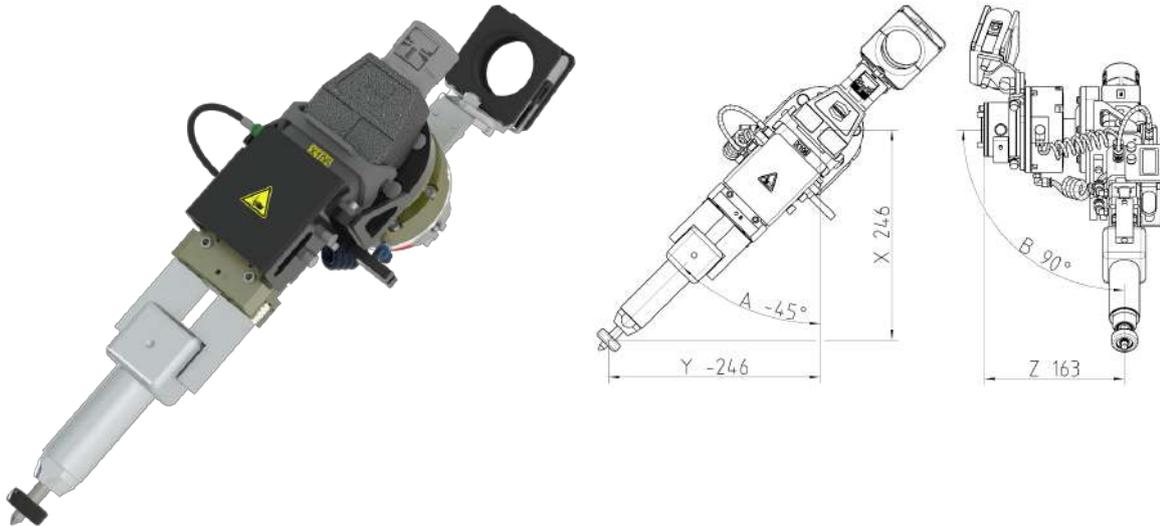
Masa [kg]	7.32				
X [mm]	25,10	A		I_x [kgm ²]	0,06
Y [mm]	-24,62	B		I_y [kgm ²]	0,06
Z [mm]	112,20	C		I_z [kgm ²]	0,07

Garra con placa



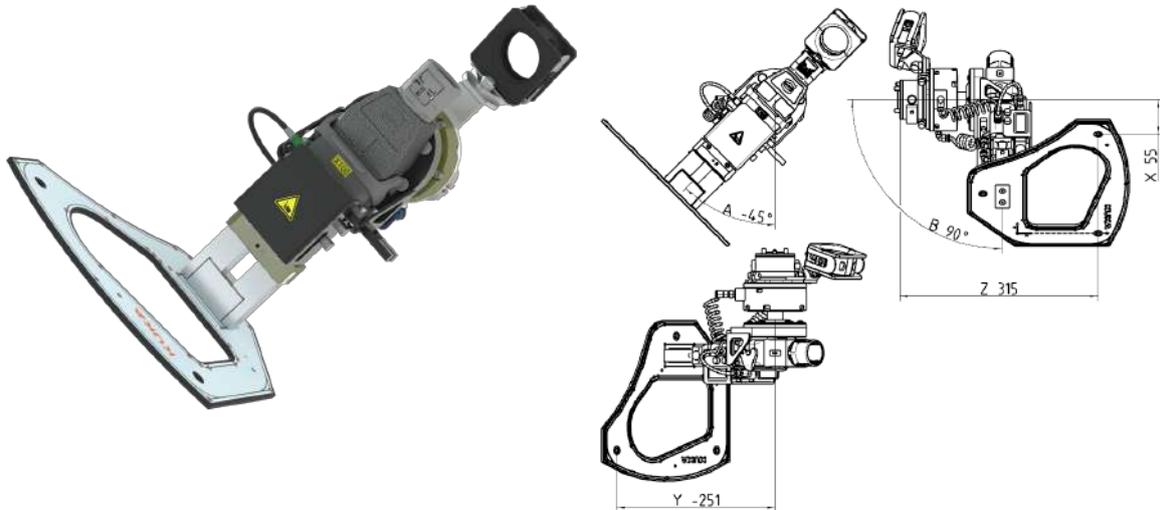
Masa [kg]	9,80				
X [mm]	60,52	A		I_x [kgm ²]	0,16
Y [mm]	-60,16	B		I_y [kgm ²]	0,16
Z [mm]	124,81	C		I_z [kgm ²]	0,16

Garra con rodillo de engrapado



Masa [kg]	7.47				
X [mm]	28.63	A		I_x [kgm ²]	0.06
Y [mm]	-28.16	B		I_y [kgm ²]	0.06
Z [mm]	113.21	C		I_z [kgm ²]	0.08

Garra con placa "Operario PRO"



Masa [kg]	8.38				
X [mm]	38,85	A		I_x [kgm ²]	0,11
Y [mm]	-41,20	B		I_y [kgm ²]	0,10
Z [mm]	133,20	C		I_z [kgm ²]	0,10

Terminal de válvulas

Carga adicional en el eje 3

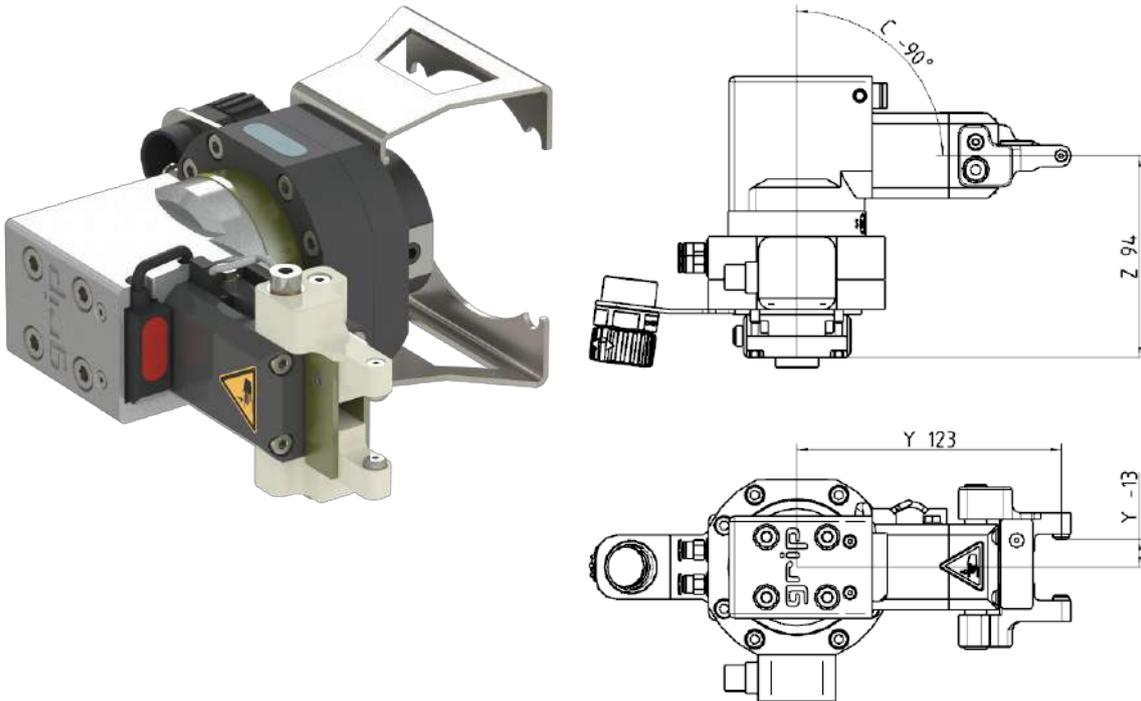


Masa [kg]	9.34				
X [mm]	-249.33	A		I_x [kgm ²]	0.97
Y [mm]	17.37	B		I_y [kgm ²]	1.55
Z [mm]	-619.02	C		I_z [kgm ²]	0.63

25.2.2 Garra en Agilus 2, ancho de agarre 25 mm

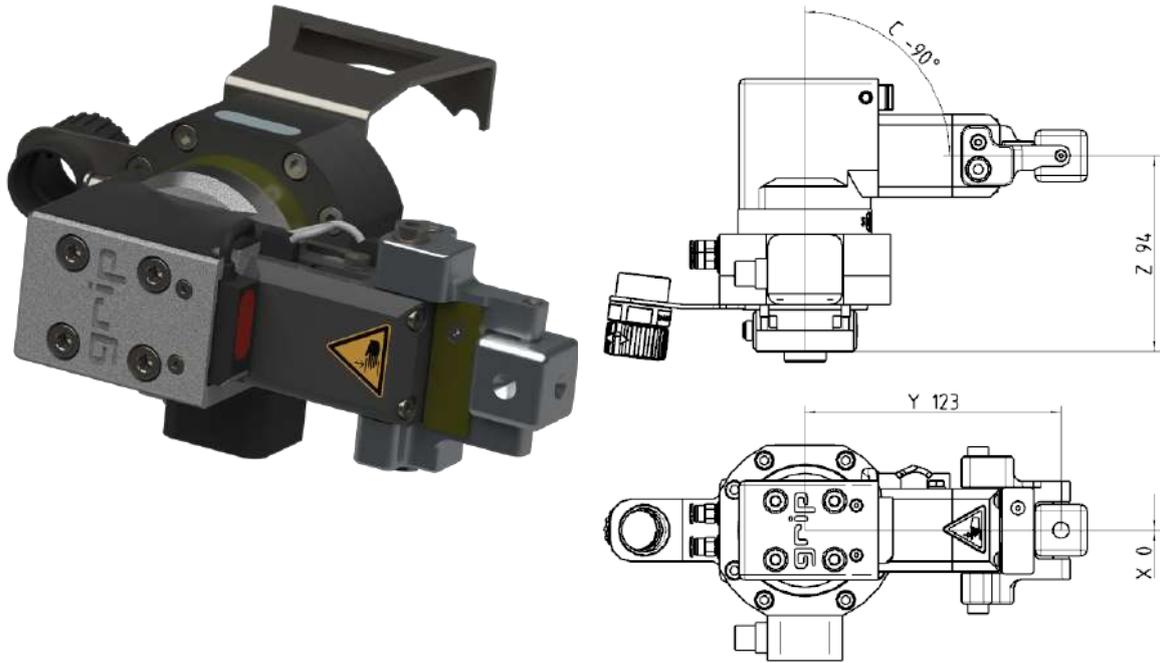
Descripción

Garra, vacía



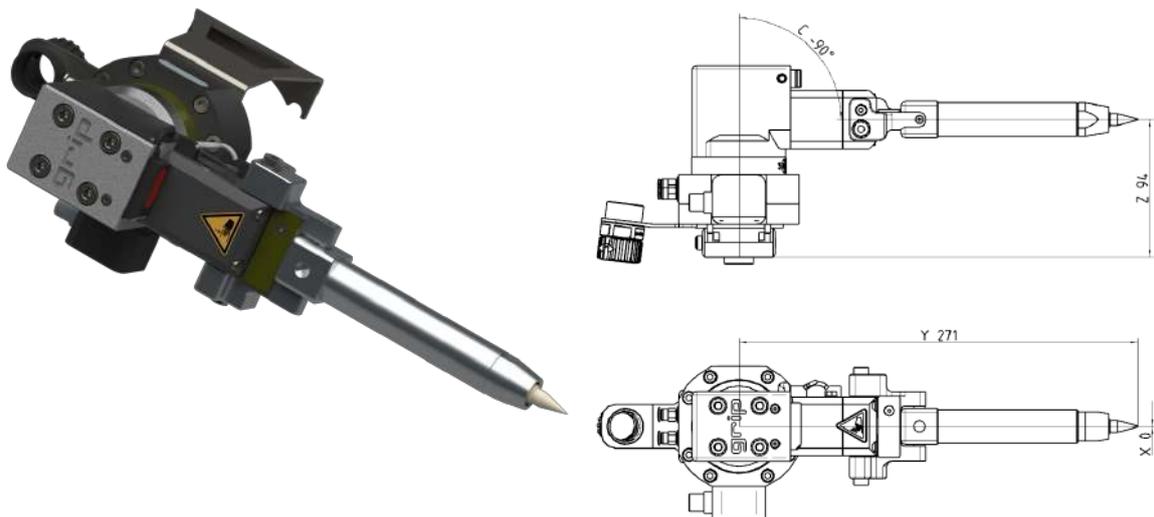
Masa [kg]	2.00				
X [mm]	1,82	A		I_x [kgm ²]	0,01
Y [mm]	13.08	B		I_y [kgm ²]	0,00
Z [mm]	54,65	C		I_z [kgm ²]	0,00

Garra con cubo



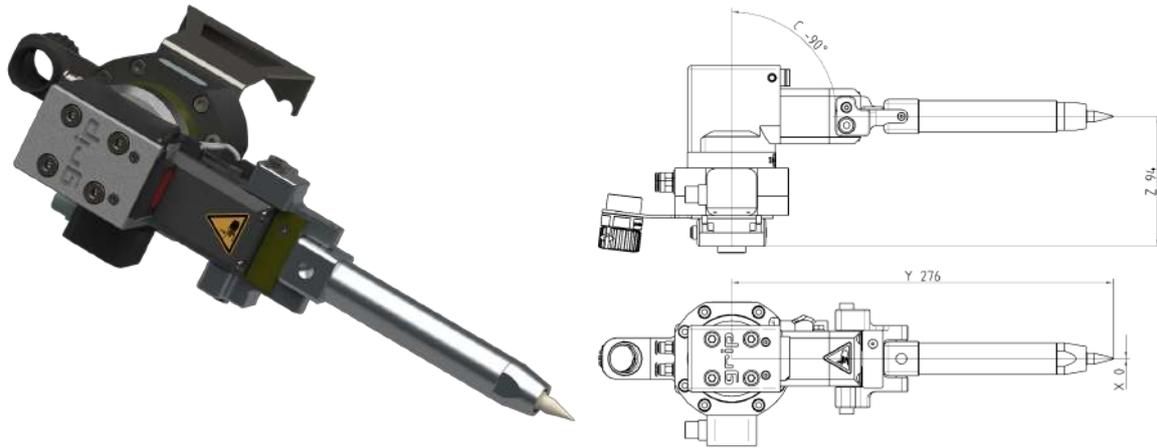
Masa [kg]	2.04				
X [mm]	1,79	A		I_x [kgm ²]	0,01
Y [mm]	15,21	B		I_y [kgm ²]	0,00
Z [mm]	55,43	C		I_z [kgm ²]	0,00

Garra con clavija "corta"



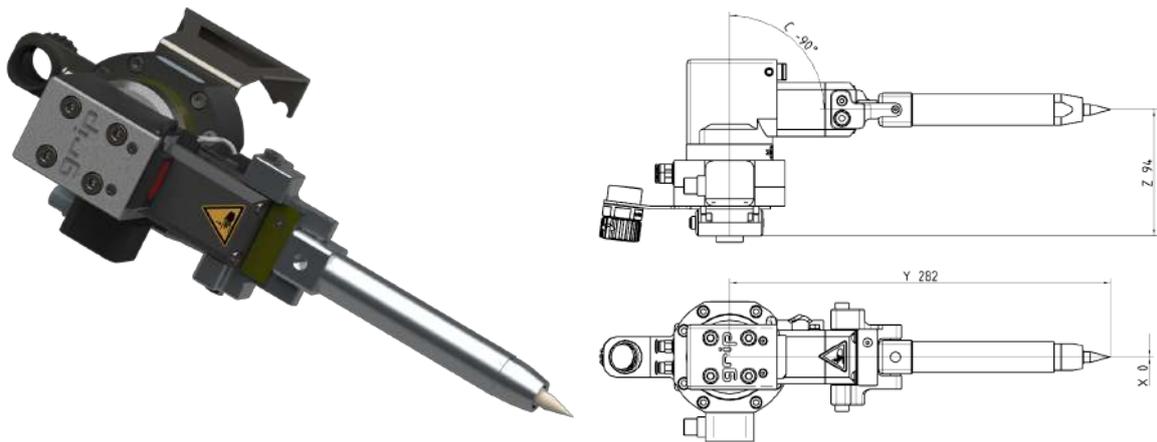
Masa [kg]	2.09				
X [mm]	1,74	A		I_x [kgm ²]	0,01
Y [mm]	20,49	B		I_y [kgm ²]	0,00
Z [mm]	56,51	C		I_z [kgm ²]	0,01

Garra con clavija "mediana"



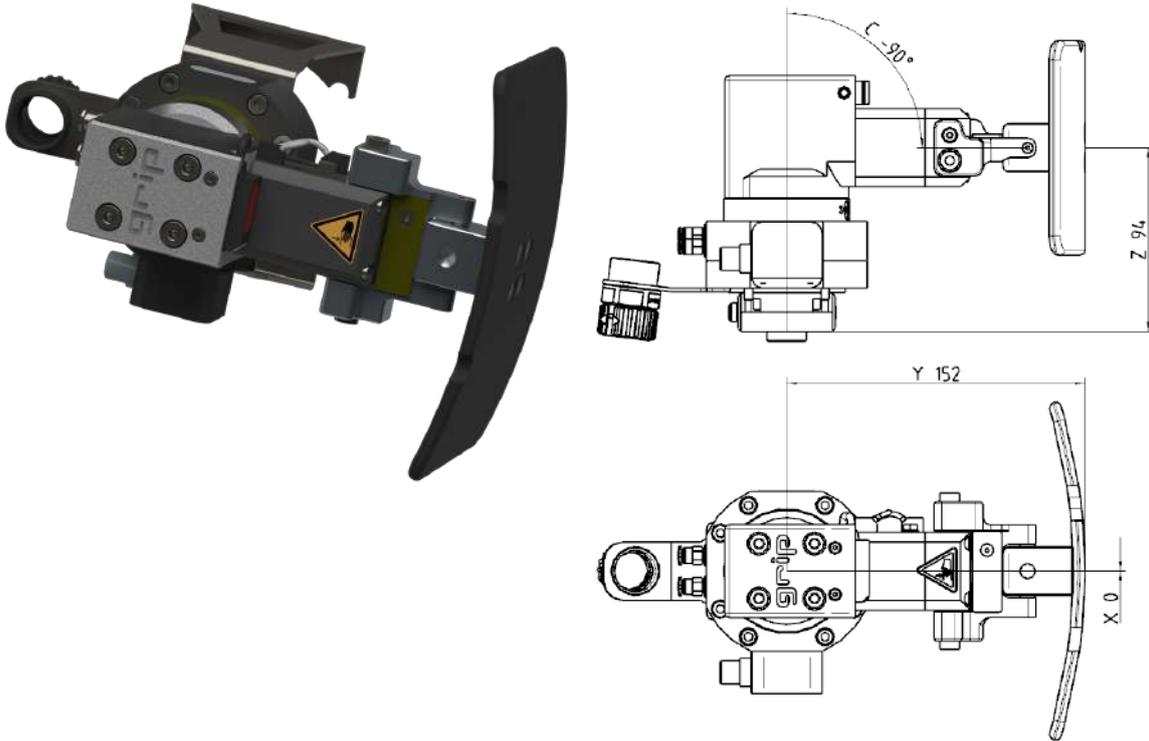
Masa [kg]	2,10				
X [mm]	1,73	A		I_x [kgm ²]	0,01
Y [mm]	20,94	B		I_y [kgm ²]	0,00
Z [mm]	56,61	C		I_z [kgm ²]	0,01

Garra con clavija "larga"



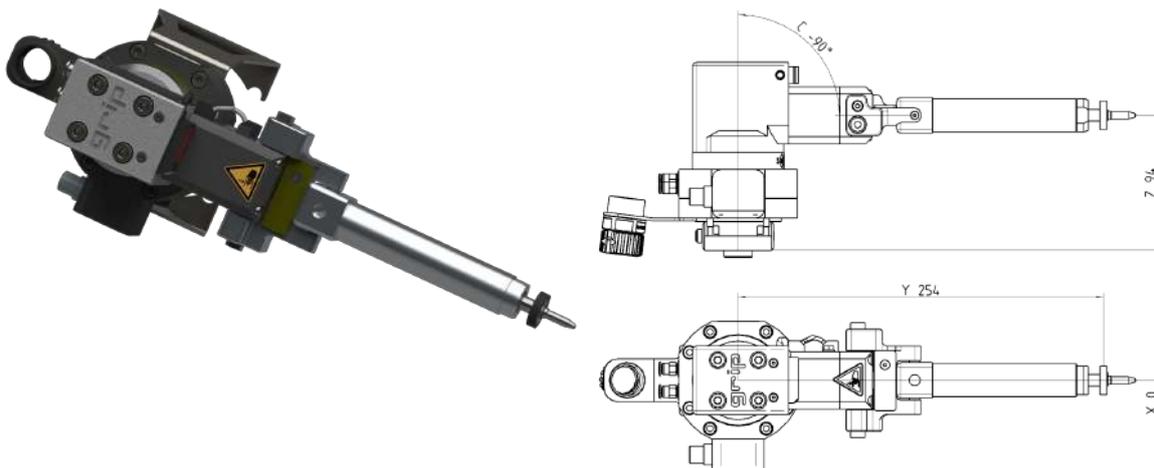
Masa [kg]	2,10				
X [mm]	1,73	A		I_x [kgm ²]	0,01
Y [mm]	21,40	B		I_y [kgm ²]	0,00
Z [mm]	56,71	C		I_z [kgm ²]	0,01

Garra con placa



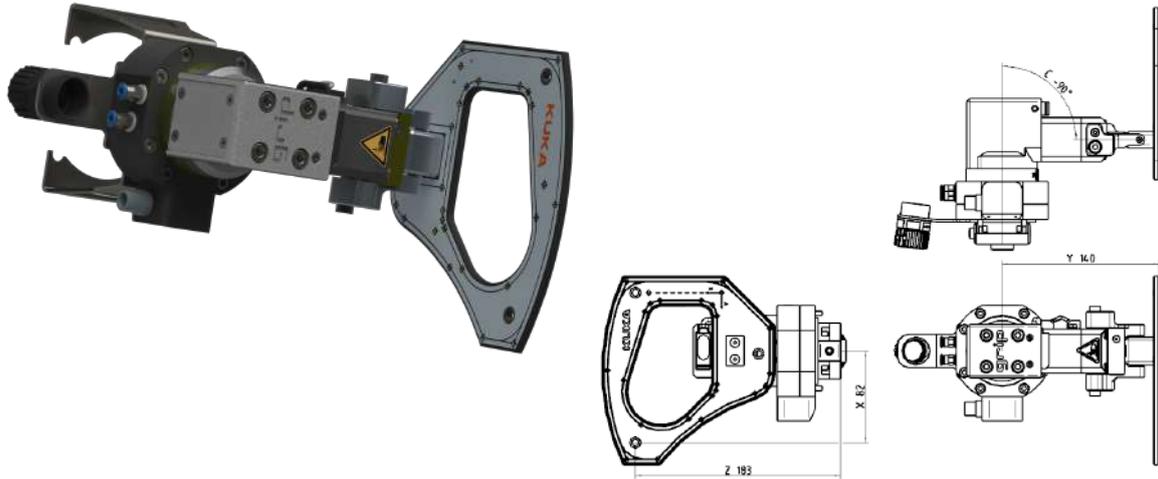
Masa [kg]	2.20				
X [mm]	1,67	A		I_x [kgm ²]	0,01
Y [mm]	24,75	B		I_y [kgm ²]	0,00
Z [mm]	58,28	C		I_z [kgm ²]	0,01

Garra con herramienta de rodillo



Masa [kg]	2.18				
X [mm]	1,67	A		I_x [kgm ²]	0,00
Y [mm]	27,45	B		I_y [kgm ²]	0,00
Z [mm]	58,02	C		I_z [kgm ²]	0,01

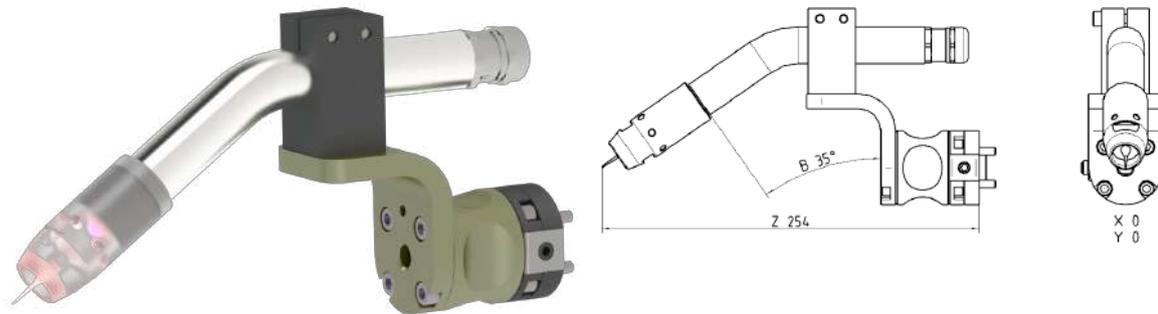
Garra con placa "Operario Pro"



Masa [kg]	2,55				
X [mm]	2,84	A		I_x [kgm ²]	0,00
Y [mm]	40,18	B		I_y [kgm ²]	0,00
Z [mm]	72,93	C		I_z [kgm ²]	0,00

25.2.3 Soplete (estándar)

Descripción

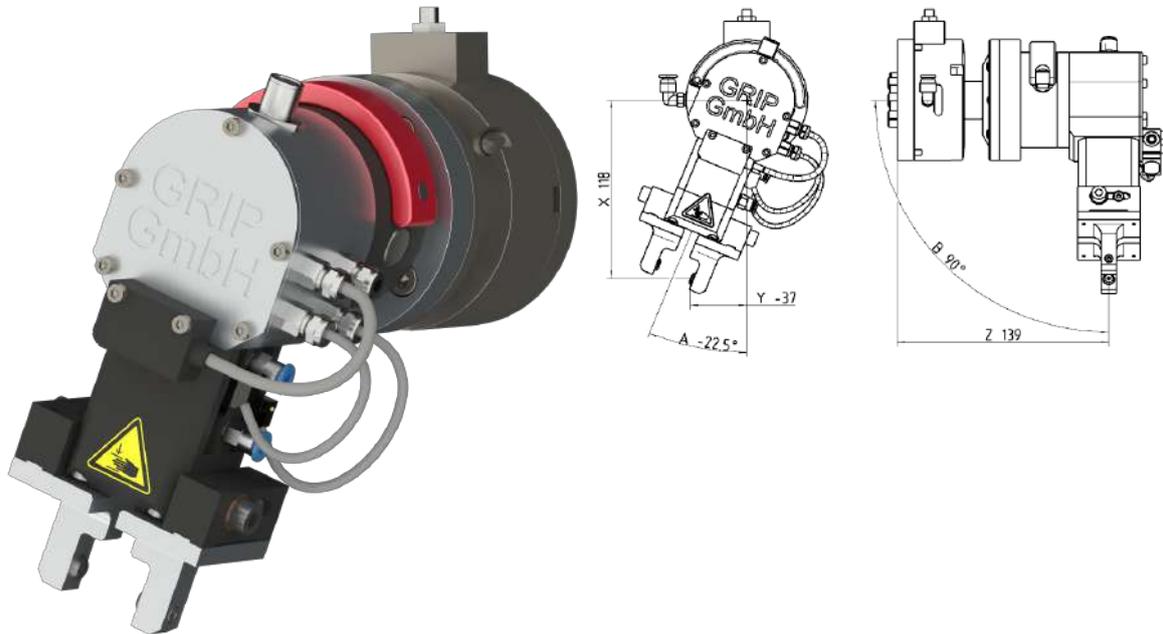


Masa [kg]	0,85				
X [mm]	-38,60	A		I_x [kgm ²]	0,00
Y [mm]	-0,17	B		I_y [kgm ²]	0,00
Z [mm]	71,90	C		I_z [kgm ²]	0,00

25.2.4 Garra en Agilus 1, ancho de agarre 25 mm

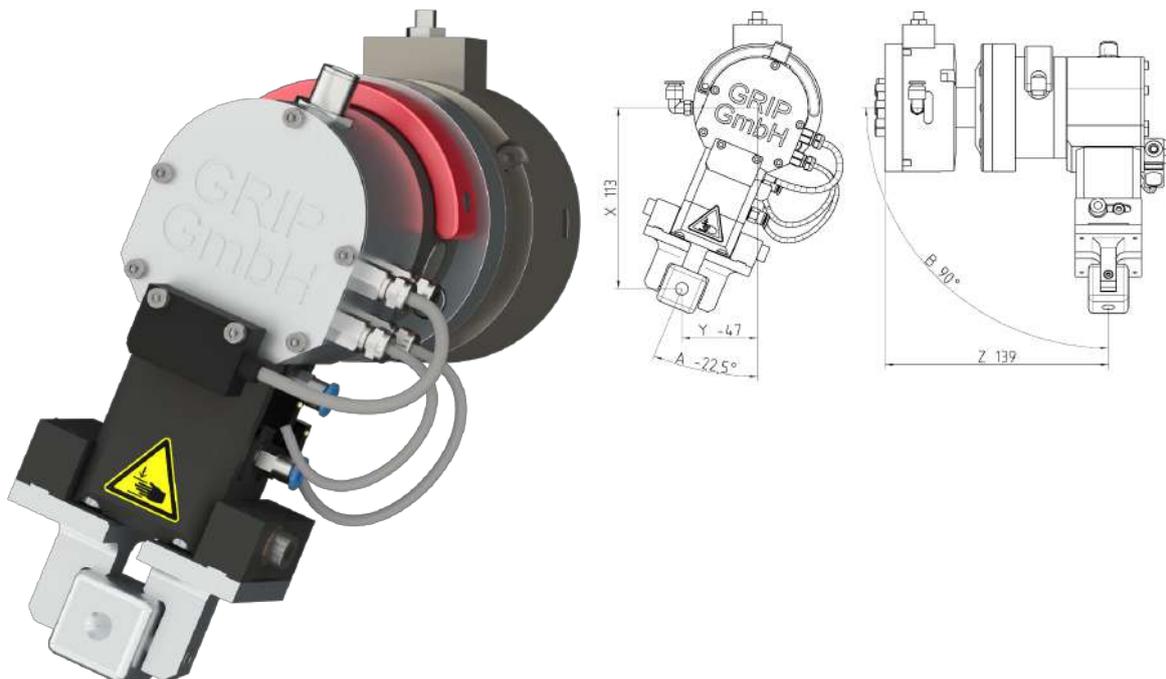
Descripción

Garra, vacía



Masa [kg]	1,80				
X [mm]	14.44	A		Ix [kgm ²]	0,01
Y [mm]	-5,63	B		Iy [kgm ²]	0,01
Z [mm]	84,85	C		Iz [kgm ²]	0,00

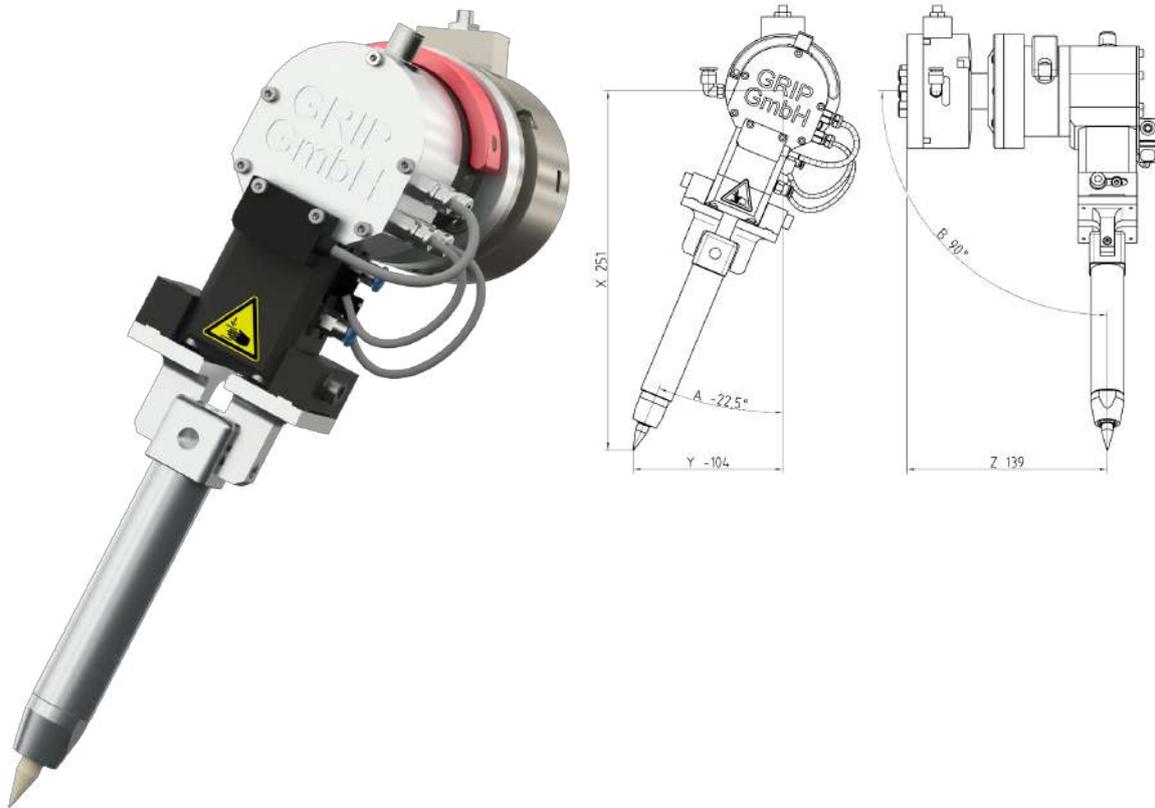
Garra con cubo



Masa [kg]	1,84				
X [mm]	16.57	A		Ix [kgm ²]	0,01

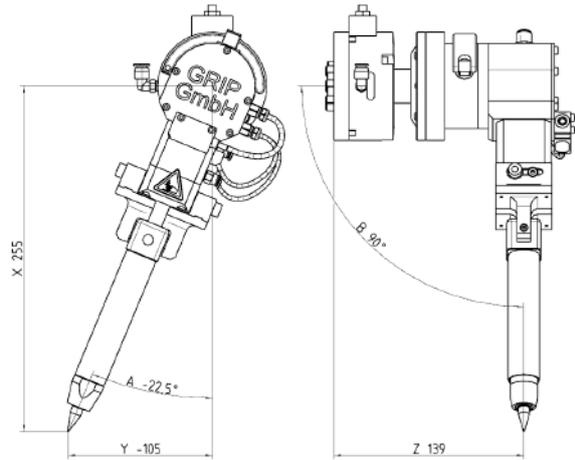
Y [mm]	-6,52	B		ly [kgm ²]	0,01
Z [mm]	86,02	C		lz [kgm ²]	0,00

Garra con clavija corta



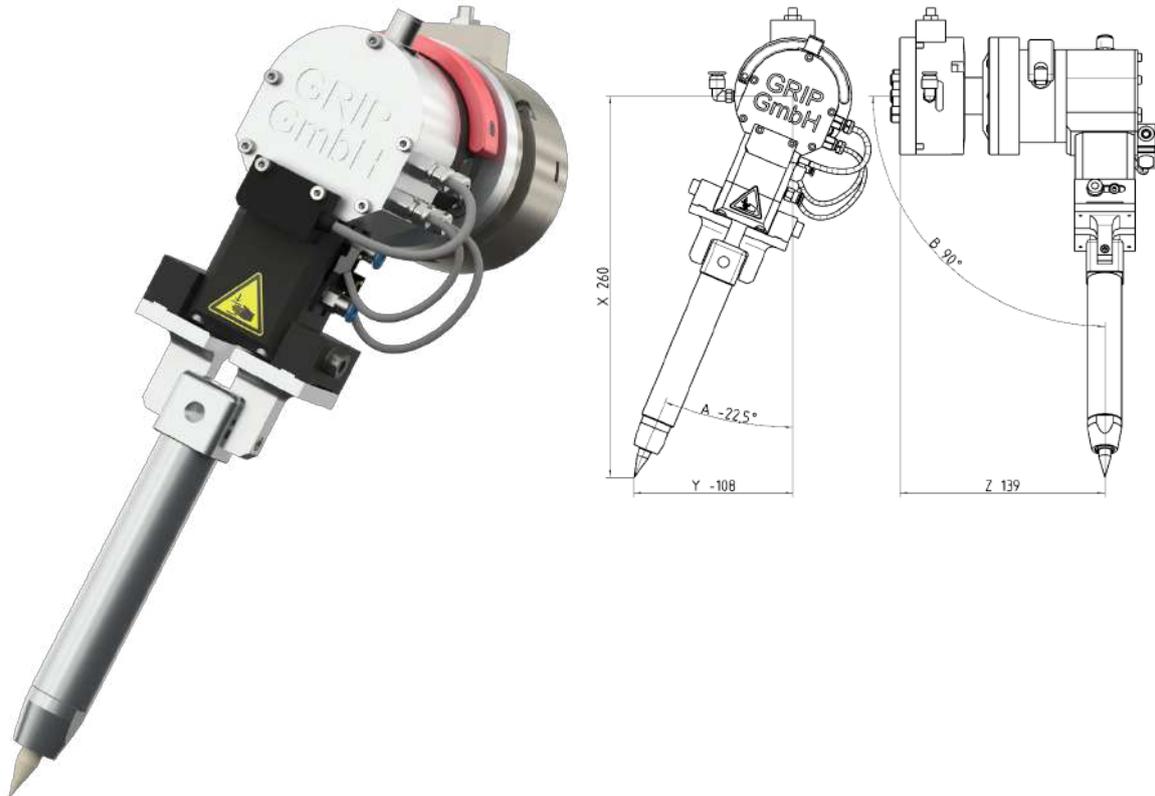
Masa [kg]	1,90				
X [mm]	21,87	A		lx [kgm ²]	0,01
Y [mm]	-8,73	B		ly [kgm ²]	0,01
Z [mm]	87,64	C		lz [kgm ²]	0,01

Garra con clavija medio larga



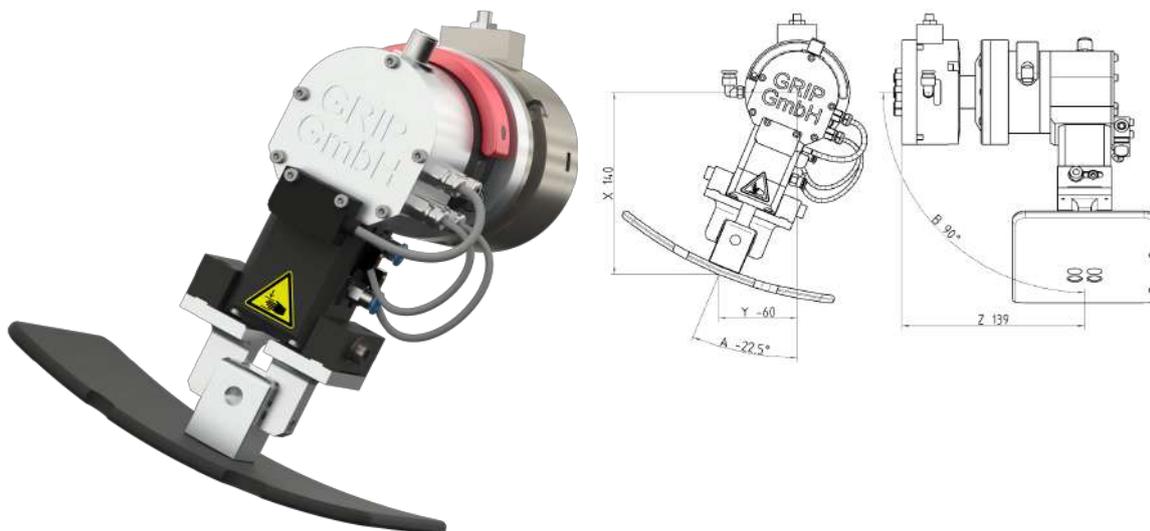
Masa [kg]	2.14				
X [mm]	38,11	A		I_x [kgm ²]	0,01
Y [mm]	-15,49	B		I_y [kgm ²]	0,01
Z [mm]	93,48	C		I_z [kgm ²]	0,01

Garra con clavija larga



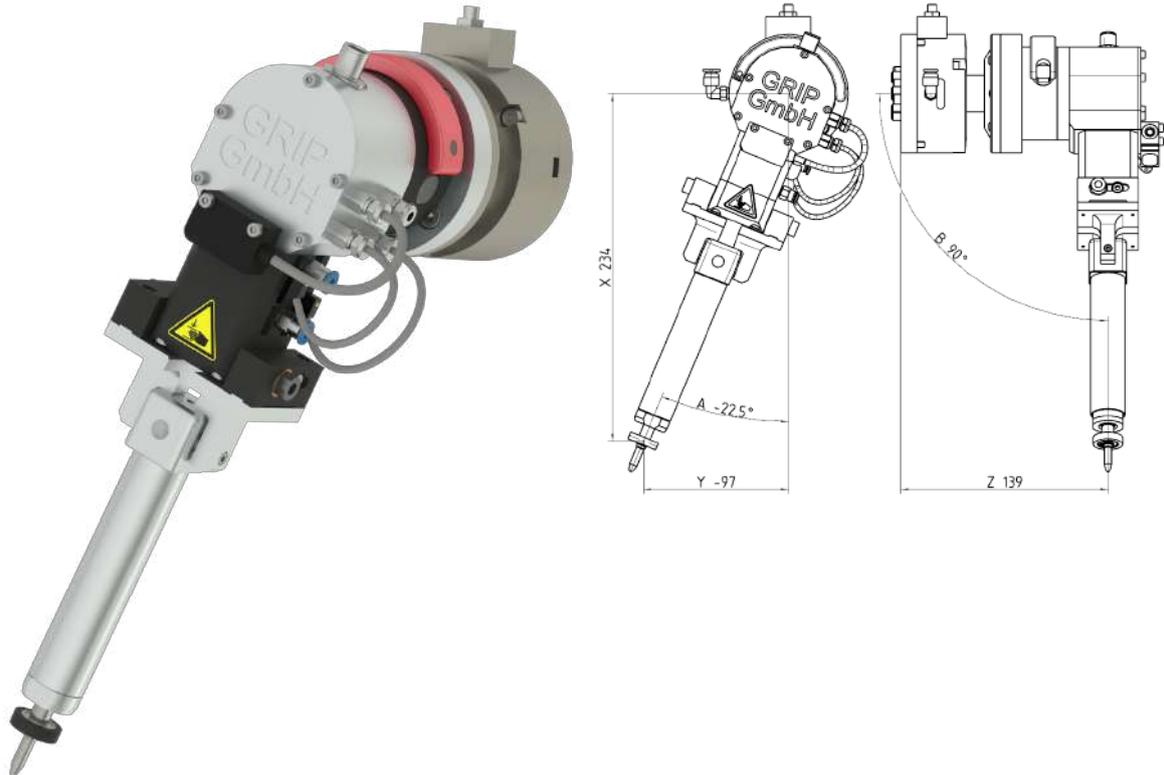
Masa [kg]	2.16				
X [mm]	39.27	A		Ix [kgm ²]	0.01
Y [mm]	-15.97	B		Iy [kgm ²]	0.01
Z [mm]	93.83	C		Iz [kgm ²]	0.01

Garra con placa



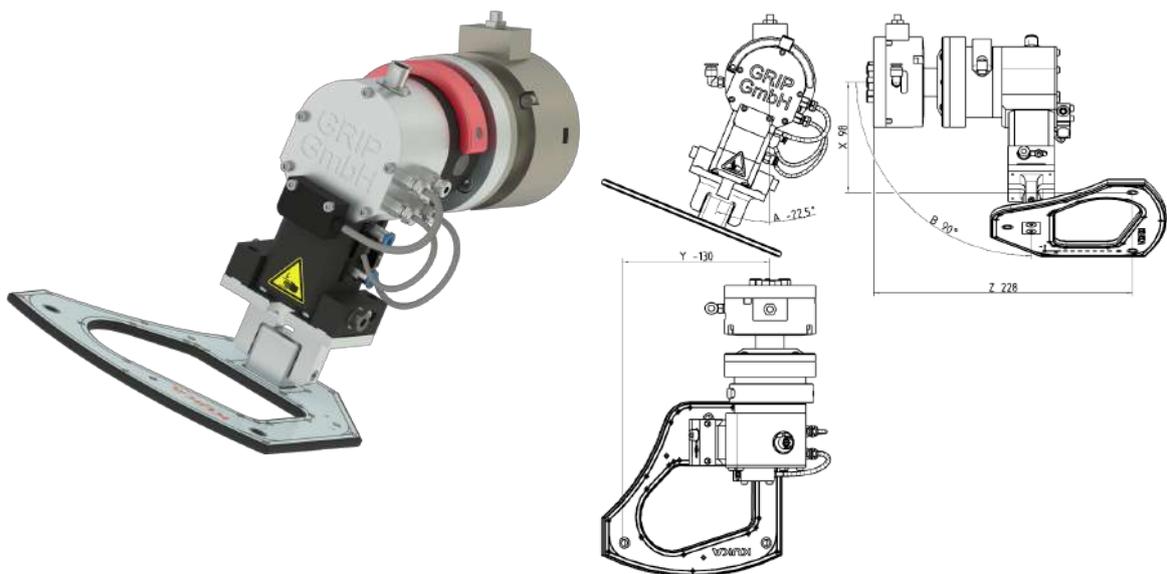
Masa [kg]	2,86				
X [mm]	57,59	A		Ix [kgm ²]	0,01
Y [mm]	-23,66	B		Iy [kgm ²]	0,02
Z [mm]	104,70	C		Iz [kgm ²]	0,02

Garra con herramienta de rodillo



Masa [kg]	1.99				
X [mm]	28.81	A		Ix [kgm ²]	0.01
Y [mm]	-11.62	B		Iy [kgm ²]	0.01
Z [mm]	29.89	C		Iz [kgm ²]	0.01

Garra con placa "Operario PRO"



Masa [kg]	2.36				
X [mm]	40,42	A		Ix [kgm ²]	0,01
Y [mm]	-18,13	B		Iy [kgm ²]	0,02
Z [mm]	108,03	C		Iz [kgm ²]	0,01

Índice

Pieza de trabajo guiada por robot.....332

A

Abandonar la celda de formación..... 24
 Acceso a la celda de formación.....21
 Ajuste..... 120
 Anexo.....421
 Aproximación: SCIRC.....227
 Aproximación: SLIN.....227

B

Bucle de conteo..... 364
 Bucle sinfin..... 357
 Bucle sinfin..... 356
 Bucle, finito.....371, 372
 Bucle, infinito..... 381

C

Cambiar modo de servicio.....47
 Caso de incendio..... 20
 Comportamiento..... 24
 Comprobar la garra
 Formulario inline.....262
 Conocer el desplazamiento manual
 incremental..... 94
 Conocer el sistema de coordenadas Base...93
 Conocer las ventajas de las herramientas
 medidas..... 157
 Consultas.....388, 389

D

Desplazamiento incremental del robot..... 95
 Desplazamiento manual, Tool..... 112
 Distribuidor SWITCH-CASE..... 398
 Distribuidores (SWITCH - CASE)..... 398
 Documentación del sistema de robot.....25

E

Ejercicio
 Ejecutar programas del robot.....42, 48
 Ejercicio, bloque Spline.....250
 Ejercicio, desplazamientos manuales con
 herramienta fija.....326
 Ejercicio, Spline.....304, 305
 Ejercicio:
 Medición TOOL con el método de 2
 puntos..... 337

H

Herramienta fija..... 328
 Herramientas guiadas por robot..... 157

I

Incremento..... 96
 Indicaciones de seguridad..... 20
 Indicaciones generales de seguridad..... 20
 Inicio del programa.....39

K

KUKA Xpert..... 25
 Registro.....27
 utilizar.....29

L

Lógica..... 296

M

Medición..... 328, 332
 Medir una herramienta..... 159
 Mensajes en el smartPAD..... 36
 Modificar, instrucciones de movimiento..... 178
 Modos de servicio
 Conocer..... 46
 Movimiento CIRC..... 92
 Movimiento LIN.....92

P

Parada condicionada.....309
 Principio..... 120
 Programa de robot..... 35
 seleccionar.....35
 Programación lógica
 lógica simple.....285
 Punto de la raíz de la muñeca..... 218, 219

R

Ramificación..... 388, 389

S

Seguridad del robot..... 14
 Sistema de coordenadas TOOL.....111
 SPTP – Desarrollo del movimiento..... 90
 Sustancias estupefacientes.....24

W

WHILE..... 371, 372