



PRÁCTICA #3

TEMA: Programación offline en RoboDk

1. Objetivos

- Familiarización RoboDk.
- Crear un programa usando instrucciones básicas de RoboDK, para realizar una tarea de Pick and Place.

2. Marco teórico

RoboDK es un software para simulación y programación fuera de línea. La programación fuera de línea significa que los programas de robot se pueden crear, simular y generar fuera de línea para un brazo de robot y un controlador de robot específicos. RoboDK contiene un menú principal, una barra de herramientas, una barra de estado y la pantalla principal. El árbol de la estación en la pantalla principal contiene todos los elementos disponibles en la estación, como robots, marcos de referencia, herramientas, programas, etc. El robot que se usará en la práctica será Kawasaki RS03N, para realizar varias pruebas con el software RoboDK

La programación fuera de línea no tiene límites con RoboDK. Este software proporciona una interfaz gráfica de usuario fácil de usar para simular y programar robots industriales. RoboDK te ayudará a evitar singularidades y límites de ejes. No se requiere experiencia en programación. Más información disponible en la sección de Programación fuera de línea de la documentación (<https://robodk.com/doc/en/Getting-Started.html>).

Con la API de RoboDK también puede programar y simular robots usando Python. Python es un lenguaje de programación que le permite trabajar más rápido e integrar sus sistemas de manera más efectiva. Python permite expresar conceptos en menos líneas de código en comparación con otros idiomas, por lo que es fácil y fácil de aprender.

Más información disponible en la sección de RoboDK API de la documentación. La API de RoboDK también está disponible para C

y Matlab .

3. Descripción.

En esta práctica se tendrá como objetivo principal realizar el movimiento del robot con un ejemplo básico de Pick and place mediante programación punto a punto de RoboDK. Los comandos que se les dará son sencillos y básicos que les será de utilidad para completar la práctica.

3.1 Seleccionar un Robot

Se pueden agregar nuevos robots desde un disco local o desde la biblioteca en línea:

Desde un disco local:



1. Seleccione **Archivo** → **Abrir**
2. Escoger uno de los Robot disponibles en el disco local.

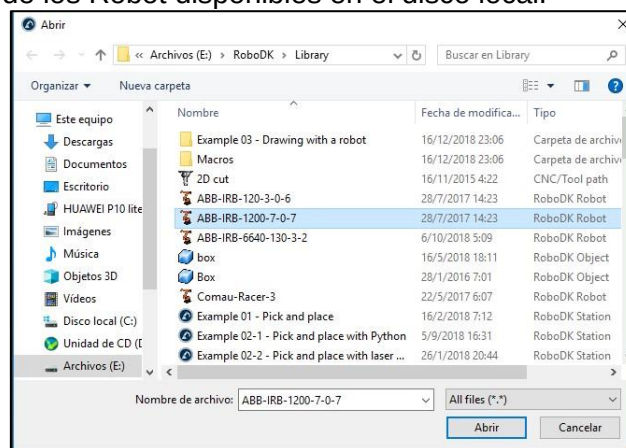


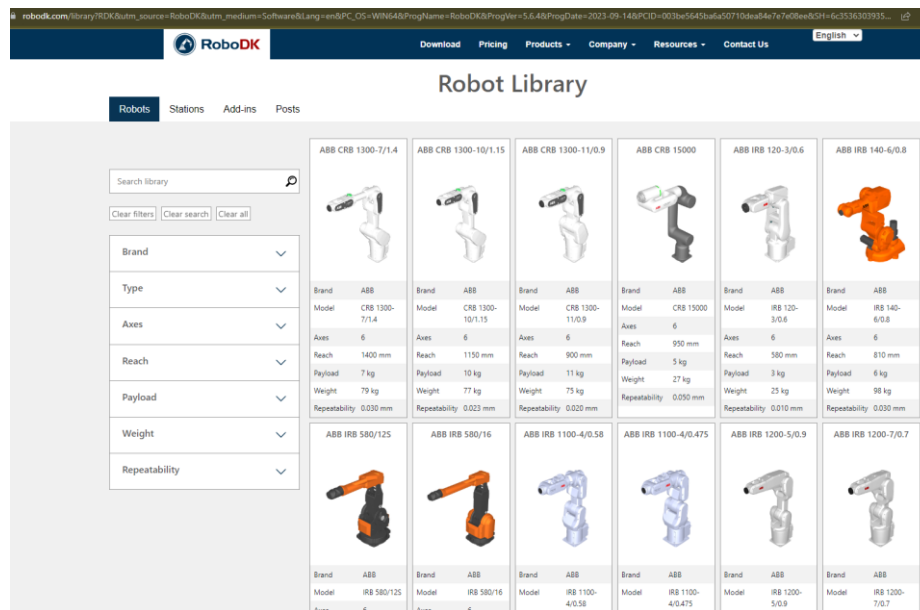
Figura 1. Disco Local

3. **Abrir.** El robot debería aparecer automáticamente en la estación en unos segundos.

Desde la biblioteca en línea:

1. Seleccione **Archivo** → **Abrir biblioteca en línea** (Ctrl + Shift + O). Aparecerá una nueva ventana en su navegador predeterminado que muestra la biblioteca en línea. También es posible seleccionar el botón correspondiente en la barra de herramientas.

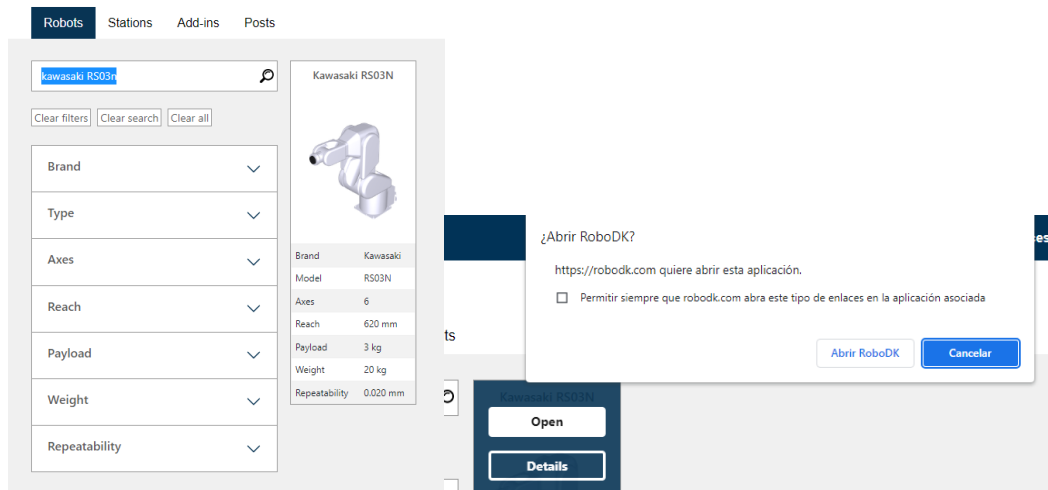




2. Use los filtros para encontrar su robot por marca, carga útil, etc.

En este ejemplo, usaremos un robot Kawasaki RS03N.

3. Seleccione **Descargar**. **Seleccionar abrir Robodk** el robot debería aparecer automáticamente en la estación en unos segundos.



4. La biblioteca en línea se puede cerrar una vez que se carga el robot o de una vez seleccionar todos los elementos a usarse

Una vez abiertos todos los elementos en robodk, estos ya se encontrarán disponibles en la librería local.

3.2 Interfaz

- El **menú principal** se encuentra en la parte superior. Todas las acciones y opciones disponibles están disponibles en este menú.
- La barra de herramientas contiene iconos gráficos que permiten un acceso rápido a las acciones utilizadas con frecuencia en el menú. Más información disponible en la sección de **la barra de herramientas**.



- El árbol de la estación enumera todos los componentes (elementos) que están presentes en la estación. Estos elementos pueden ser robots, herramientas, objetos, objetivos o configuraciones específicas para fines de fabricación o calibración. El árbol permite comprender y modificar la dependencia que existe en el entorno real. Por ejemplo, un objetivo se puede adjuntar a un marco de referencia específico, este marco de referencia se puede adjuntar al marco base del robot, las herramientas del robot generalmente se adjuntan al robot, etc.
- La barra de estado se encuentra en la parte inferior y puede mostrar consejos útiles para ciertas operaciones.
- La vista 3D (Pantalla principal) muestra la vista en un entorno virtual 3D y reproduce el Árbol de estaciones con su jerarquía.
- Al hacer doble clic en un elemento (en el árbol o en la vista 3D) se mostrará una nueva ventana con las propiedades del elemento. Por ejemplo, al hacer doble clic en un robot se mostrará el **Panel de robots**. Estas ventanas se pueden cerrar seleccionando la cruz en la parte superior derecha de la ventana secundaria.

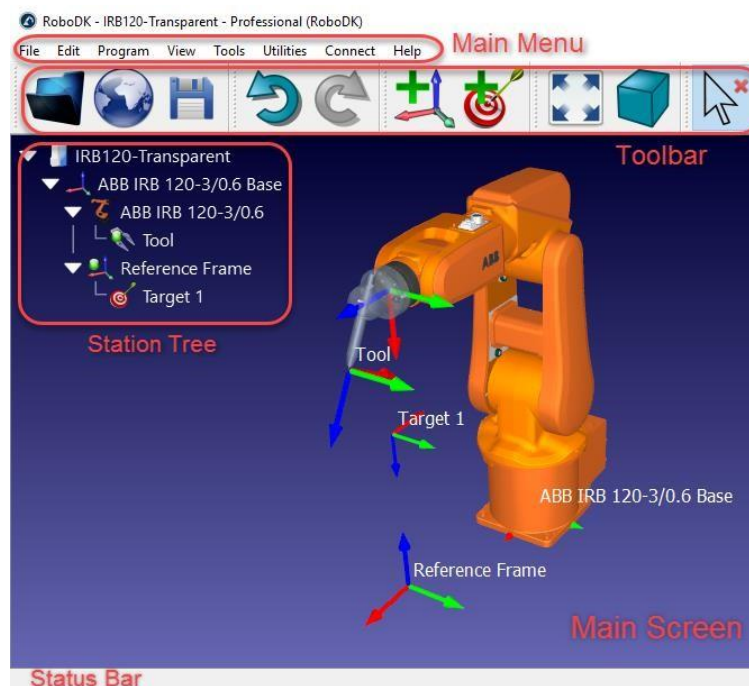


Figura 2. Software RoboDK

3.3 Programación básica de RoboDK

Para crear un nuevo programa usando RoboDK es necesario tomar en cuenta lo siguiente:

- Para crear un programa se debe presionar sobre





- Para añadir una instrucción de movimiento se presiona click derecho sobre **Prog1 ->Agregar instrucción**.
- Escoger el tipo de movimiento que se desea realizar en el robot:

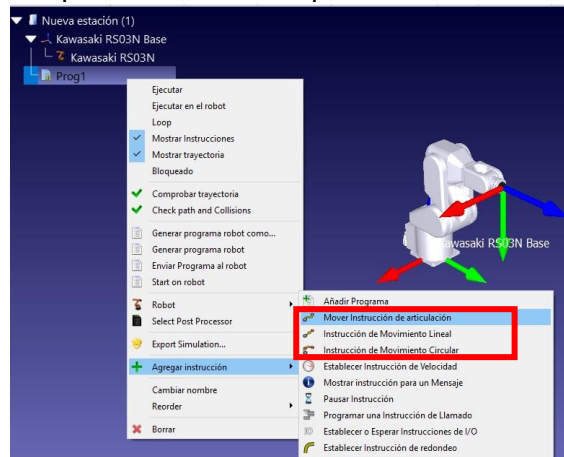


Figura 3. Agregar una instrucción

Otra forma de agregar una instrucción de movimiento es desde la barra de herramientas, presionando sobre cualquiera de estos iconos:



Al agregar una instrucción de movimiento se crea el Target 1 en la estación de trabajo, sin embargo, su presencia se puede ocultar presionando click derecho sobre **Target 1 -> Visible**, Figura 4. De esta forma se pueden ocultar todos los Targets creados.

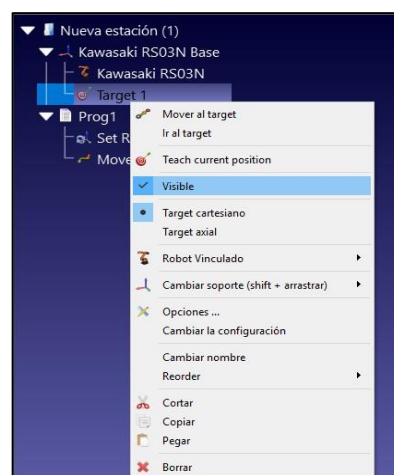


Figura 4. Quitar la visibilidad a un Target



Para manipular el robot de un punto a otro se debe seleccionar , de la barra de herramientas. Luego se observa en la estación de trabajo que aparecen varios ejes y planos sobre el robot, se puede hacer clic sobre cualquiera de estos ejes para empezar a mover el robot.

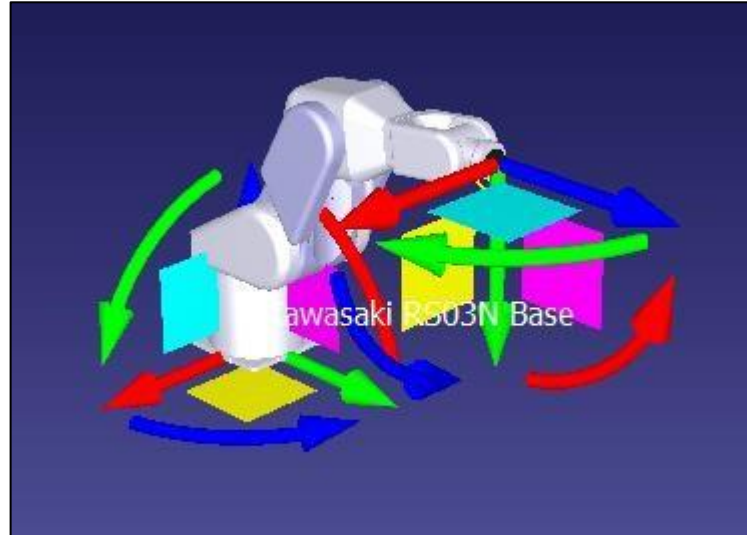


Figura 5. Kawasaki RS03N con todos sus ejes y planos.

Mover instrucción de articulación:

El robot escoge la trayectoria de mayor conveniencia para llegar hasta este punto, el objetivo será exitoso sin mayor problema siempre y cuando la posición del robot sea la indicada para poder alcanzar el target propuesto.

Instrucción de movimiento lineal:

El robot sigue una trayectoria lineal para llegar a este punto. Contrariamente a los movimientos conjuntos o articulares, los movimientos lineales son sensibles a las singularidades de los robots y los límites de los ejes. La velocidad del robot es ligeramente mayor cuando realiza un movimiento lineal en comparación con un movimiento conjunto.

Instrucción de movimiento circular:

A menos que se seleccionen dos objetivos antes de agregar la instrucción, la instrucción de movimiento no creará nuevos objetivos. Es necesario agregar dos objetivos más por separado y vincularlos desde la instrucción de movimiento circular, como se muestra en la siguiente imagen.

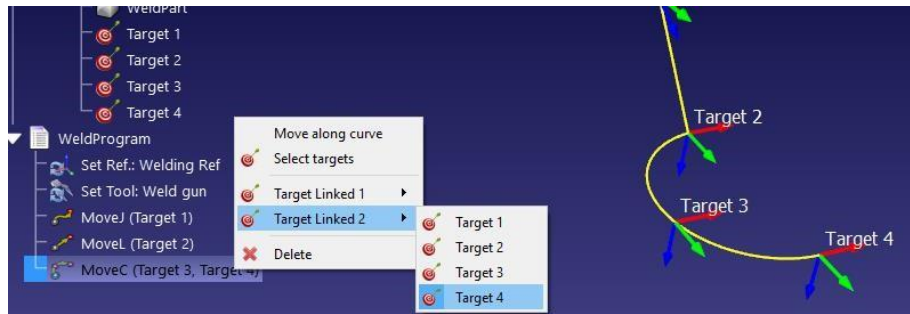


Figura 6. Movimiento Circular.

La ruta circular es un arco creado desde el punto donde se ubica el robot, pasando por el primer punto circular (Target Linked 1) y finalizando en el punto final (Target Linked 2).

Importante: No es posible realizar un círculo completo con solo una instrucción circular. Un círculo completo debe dividirse en dos movimientos circulares separados.

Velocidad fijada:

Seleccione click derecho **Prog1** → **Agregar Instrucción** → **Configurar**

instrucción de velocidad  para agregar una nueva instrucción que cambie la velocidad y / o la aceleración. Es posible especificar la velocidad y las aceleraciones en el espacio conjunto y en el espacio cartesiano.

Active los casos correspondientes para imponer una velocidad y / o aceleración específica en el programa. La velocidad del robot se aplica desde el momento en que se ejecuta esta instrucción.

La velocidad del robot también se puede cambiar en el menú de parámetros del robot: haga doble clic en el robot y luego seleccione los parámetros.

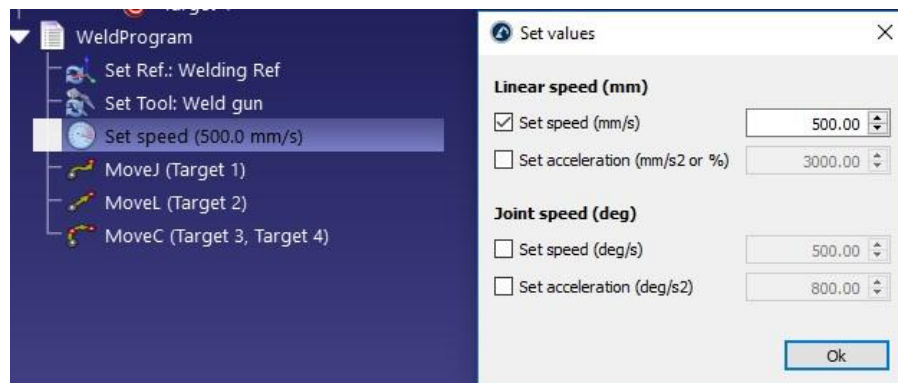


Figura 7. Instrucción de velocidad



Llamada a programa:

Seleccione click derecho **Prog1** → **Agregar Instrucción** → **Instrucción de llamada de programa**  para agregar una llamada a un subprograma del programa actual.

Por defecto, esta es una llamada de bloqueo a un programa específico. Sin embargo, es posible cambiar a **Insertar código** para ingresar un código específico en la ubicación de esta instrucción. Esto podría ser útil para una aplicación específica y un controlador específico.

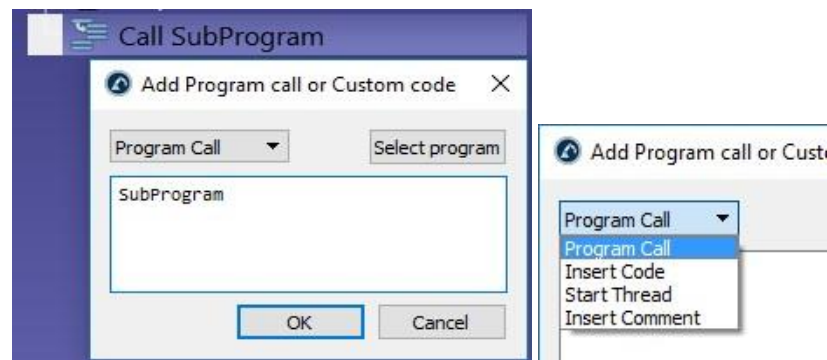


Figura 8. Programar una instrucción de llamado.

Establecer / Esperar IO:

Seleccione **Prog1** → **Agregar Instrucción** → **Configurar o esperar instrucciones de E/S** para cambiar el estado de las salidas digitales (DO). De forma predeterminada, esta instrucción se establece en **Establecer salida digital**. Esta instrucción también permite esperar a que una entrada digital (DI) específica cambie a un estado específico.

El nombre IO puede ser un número o un valor de texto si es una variable nombrada. El valor de IO puede ser un número (0 para falso y 1 para verdadero) o un valor de texto si es un estado con nombre.

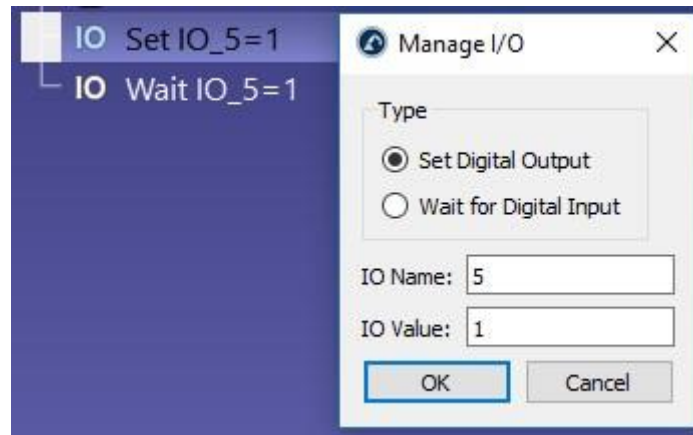


Figura 9. Establecer o esperar una instrucción de IO.

Adjuntar-Dejar objeto:

Adjuntar objeto a terminal de robot

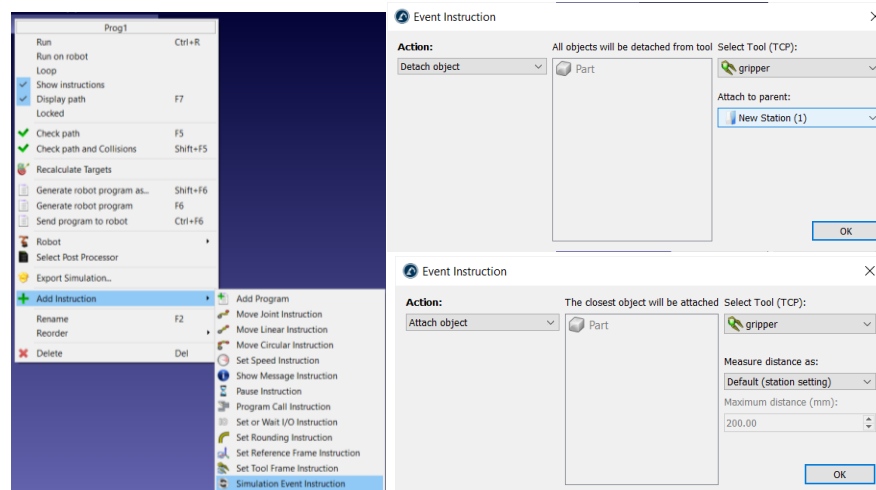


Figura 10. Instrucción de simulación de evento

Ejecutar el programa:

Dar doble clic sobre Prog1, entonces se ejecutarán todos los movimientos e instrucciones creadas en el programa 1.

4. Procedimiento

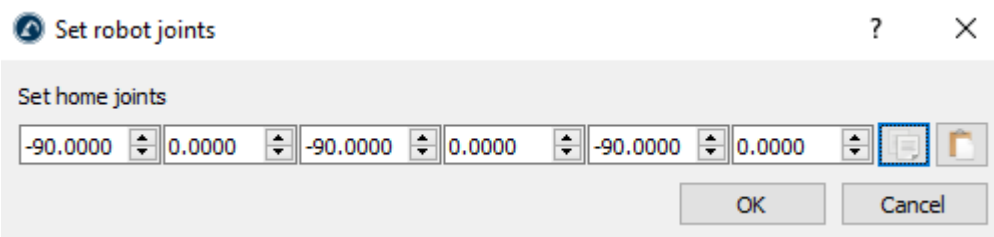
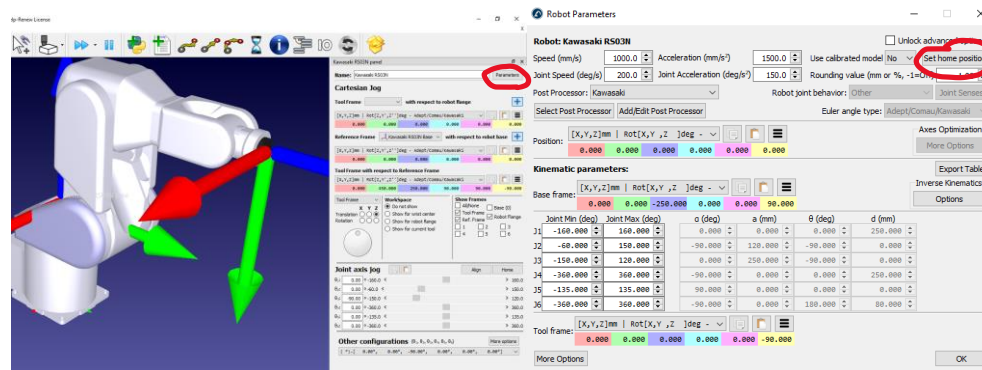
4.1 Programación básica mediante RoboDK.

- Agregar robot Kawasaki RS03N en el espacio de trabajo.



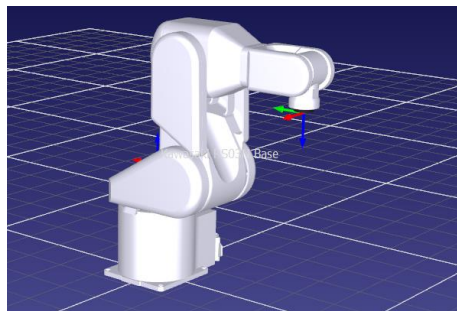
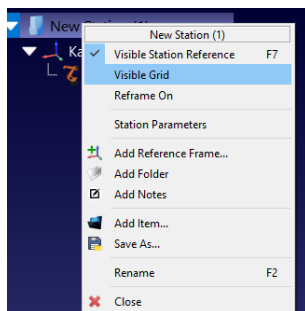
- b) Revisar la posición home $[-90,0,-90,0,-90,0]$ del robot en el software RoboDk.

Para esto hacer doble clic sobre el robot y se abrirá la siguiente ventana:



Seleccionar Parameters, set Home Position, dar ok, Ok al menú anterior y en la ventana del robot hacer clic en home, se observará que el robot se mueve a la posición indicada.

- c) Guarda la estación para evitar perdidas de avances, cada 5 minutos máximo.
d) A continuación, dar clic derecho al nombre de su archivo en el árbol de trabajo y seleccionar rejilla o visible grid, esto activará una referencia del suelo.



- e) Añadir un nuevo terminal al robot, en la librería online buscar:



RobotiQ EPick Vacuum - Gripper (1 Cup)

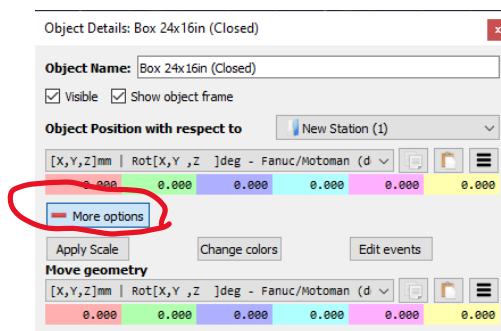


f) Añadir una caja en la librería online buscar:

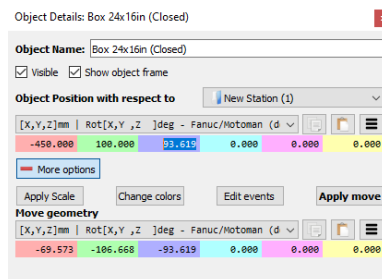


Box 24x16in (Closed)

g) Hacer doble clic en la caja y aplicar escala de 0.35 en more options :

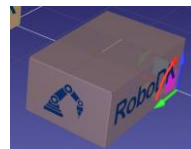


h) Después dejar los valores de la caja como se muestra a continuación:



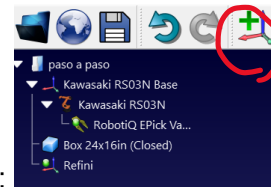
Con esto se establece que los coordenadas de la caja se encuentre exactamente en el centro de la cara superior de la caja, a continuación, la comparación del antes y después:

Antes



Después

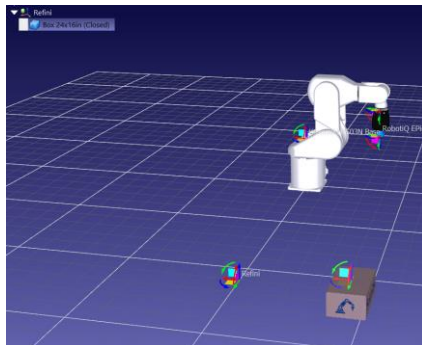




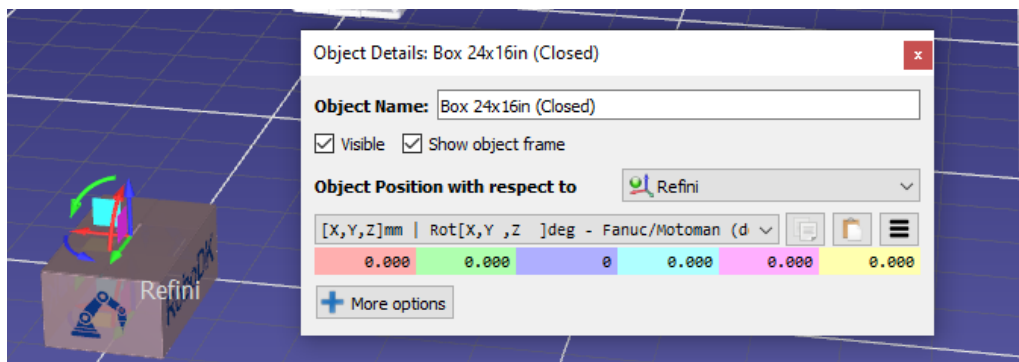
i) Agregar un nuevo sistema de referencia:

Colocarle el nombre de Refini y arrastras la caja al sistema de referencia creado

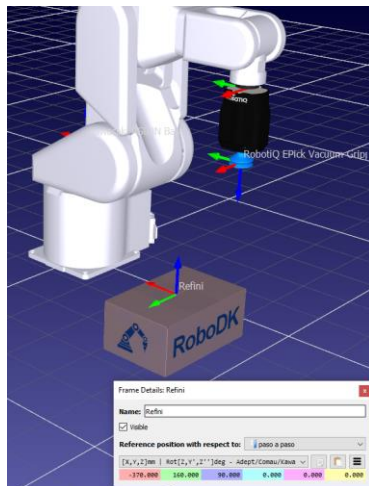
Probablemente les quede algo así:



Si es el caso, hacer doble clic sobre la caja y seleccionar sistema de referencia Refini y colocar todos los valores en 0, les debe quedar así:

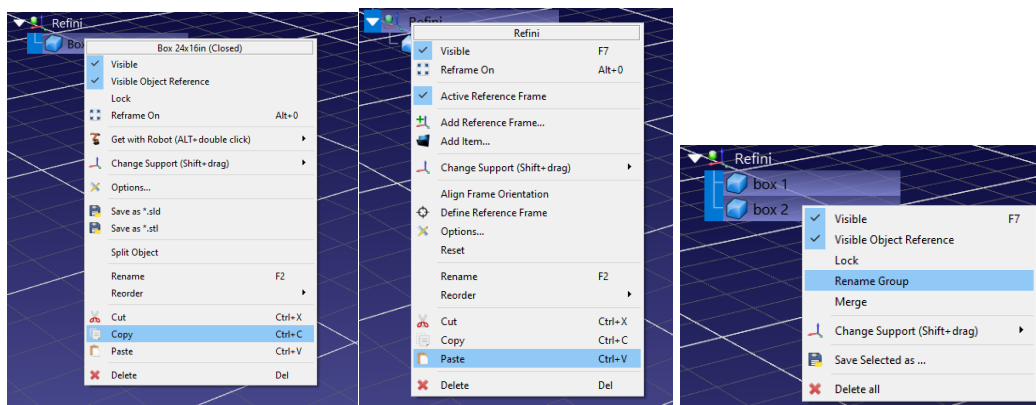


j) Ahora cerrar esa ventana y hacer doble clic sobre el sistema de referencia creado en el arbol de trabajo:



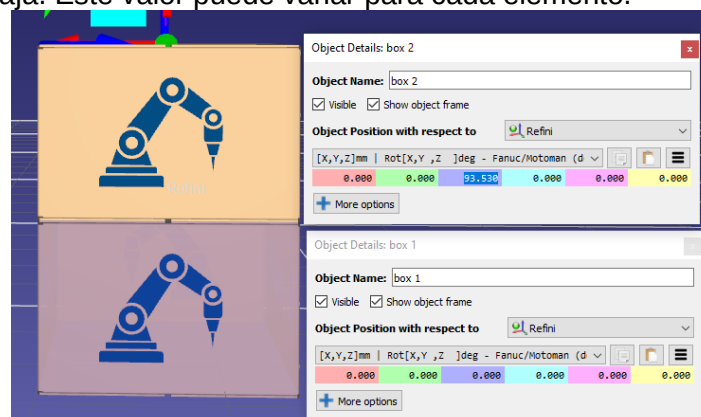
Con estos valores nos aseguramos que el robot pueda alcanzar la caja, para ello mantener presionada la tecla alt y hacer doble clic en el robot, esto hará que el terminal del robot se coloque sobre la caja justo en el sistema de referencia de la misma.

- k) Ahora agregar una segunda caja, para ellos podemos repetir los pasos anteriores y modificar la caja desde el inicio, o más sencillo copiar y pegar la caja ya modificada. Clic derecho sobre la caja copiar, y a continuación, clic derecho sobre el sistema de referencia en el árbol de trabajo y pegar.

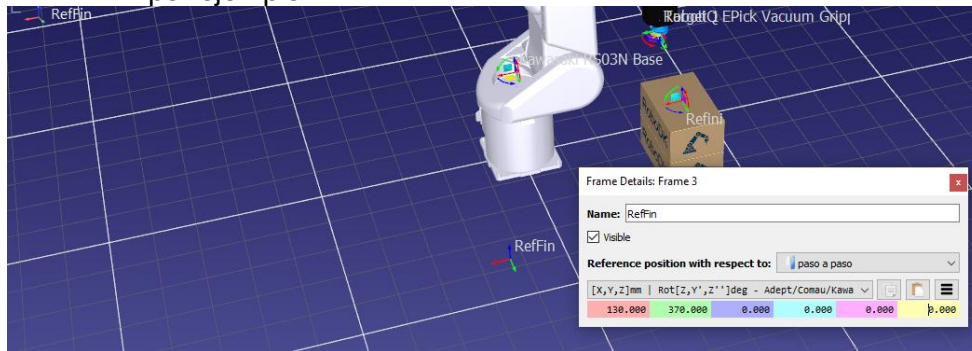


Finalmente seleccionar ambas cajas y renombrar grupo como box

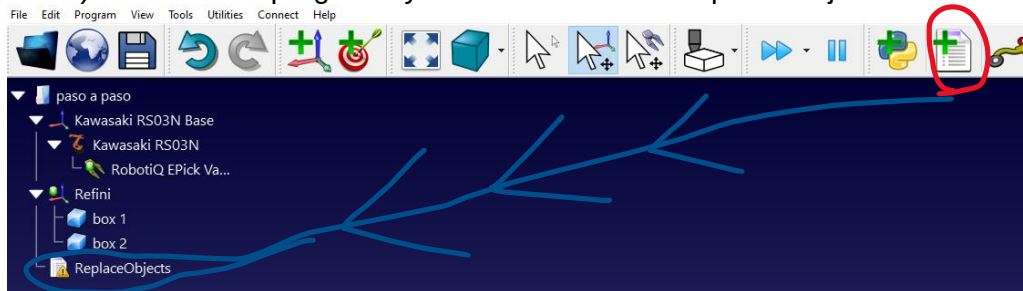
- l) Ahora hacer doble clic a la caja 2 y vamos a aumentar la posición de Z a 93.530, para que se coloque exactamente arriba de la primera caja. Este valor puede variar para cada elemento.



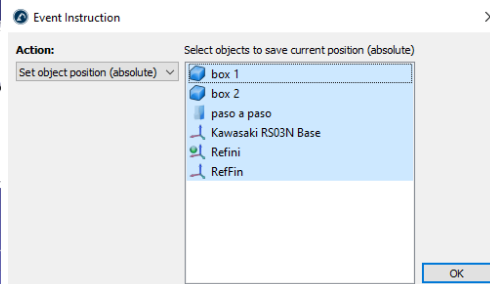
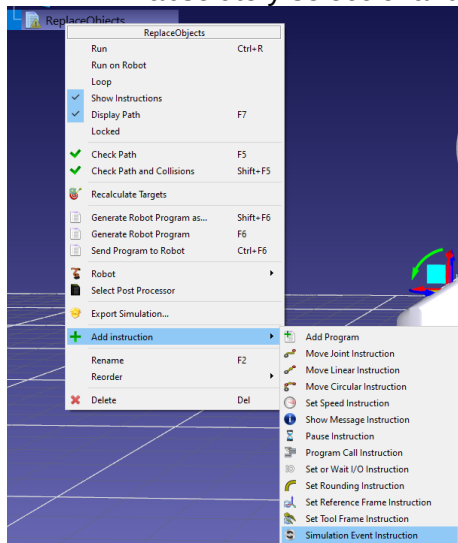
- m) Agregar una nueva referencia que se coloque cerca del robot, como, por ejemplo:



- n) Crear un programa y renombrarlo como Replace Objects



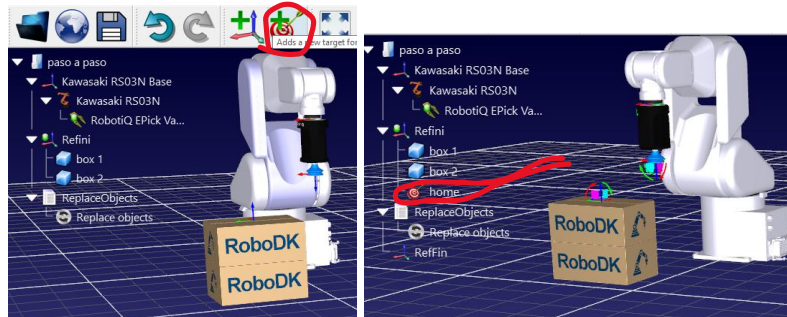
- o) Hacer clic derecho en ese programa y seleccionar Añadir instrucción- Instrucción de simulación de evento, seleccionar set object position absolute y seleccionar todos los elementos colocados alado y ok.



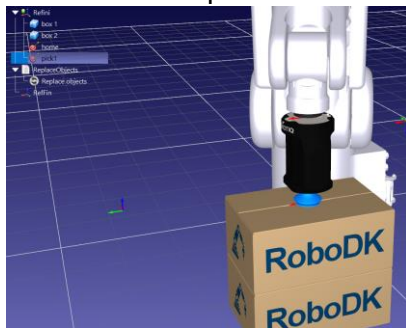
Con esto nos aseguramos que si los objetos son movidos erróneamente a otra posición, al ejecutar este programa los objetos volverán a esta posición base, en cada de agregar más objetos después de haber añadido esta instrucción, debe hacer clic derecho en el evento, y seleccionar todos los elementos de nuevo y dar ok, con esto se añadirán a la instrucción deseada, pero **tener cuidado primero debe asegurarse que todos los objetos se encuentren en la posición deseada inicial, caso contrario se guardará la nueva posición en la que se encuentren.**



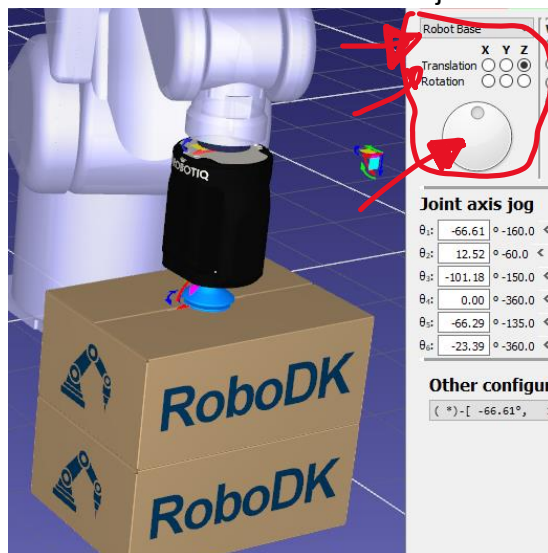
- p) Realizar un programa que permita hacer una tarea de “pick&place”
- i En la posición home del robot, añadir ese target al arbol de trabajo, para ellos hacer clic en new target Y renombrarlo como home:



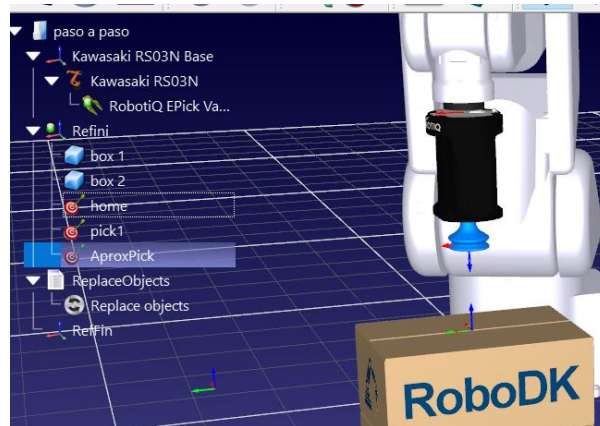
- ii A continuación, mantener presional alt y doble clic en la caja 2(la caja que está más arriba), y guardaremos esa posición como pick1.



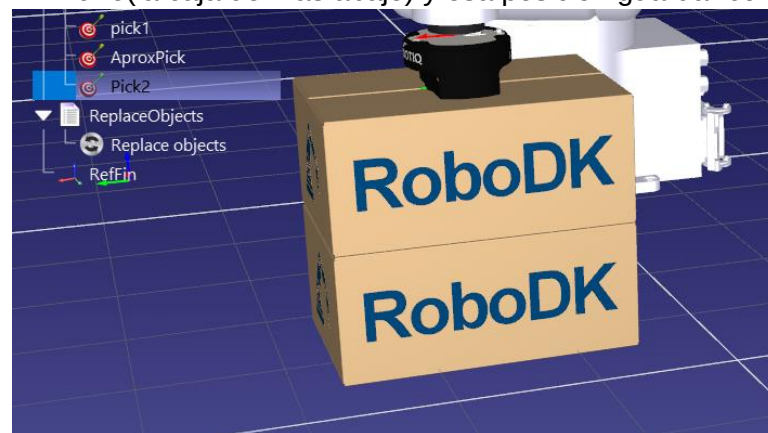
- iii Ahora al hacer doble clic en el robot, en la ventana que aparece del lado derecho, mover el valor de Z en traslación para que el robot suba en el eje Z.



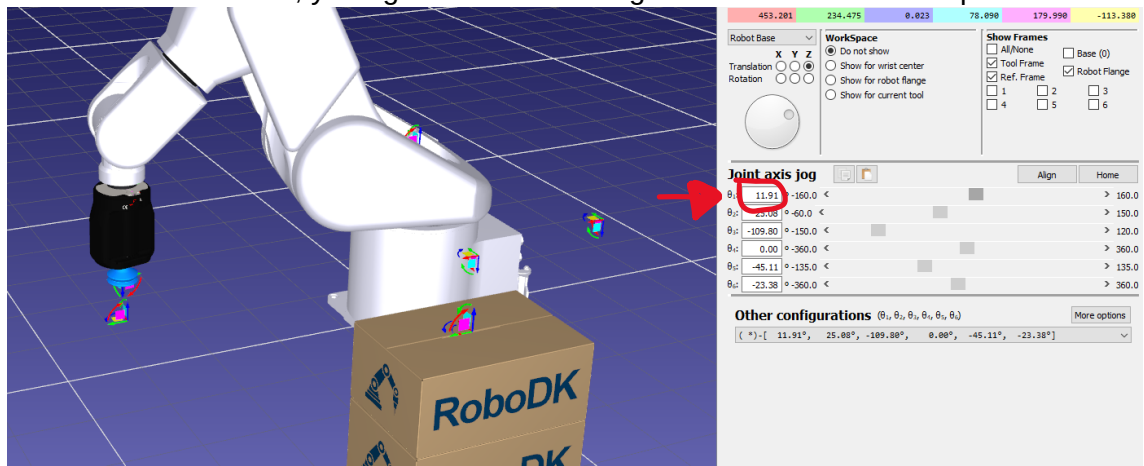
- iv Girar la perilla hasta que el robot suba un poco, y ese punto lo guardaremos como aproxPick



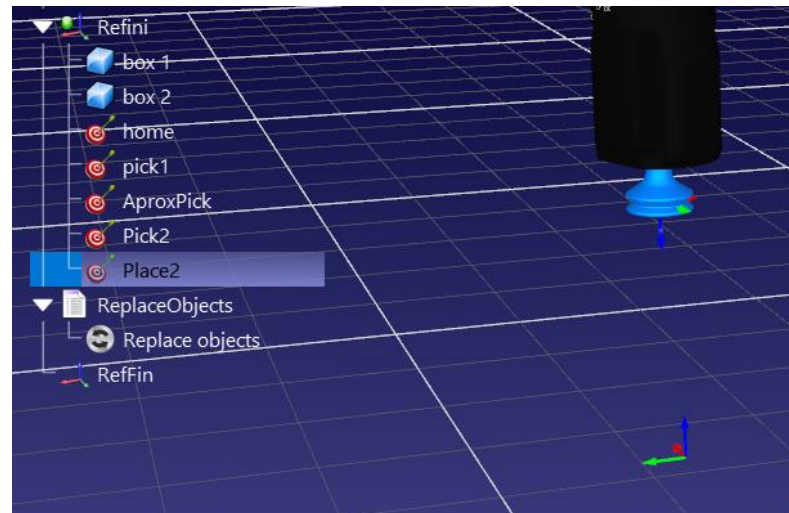
- v A continuación, mantener presionado alt, y doble clic en la caja uno (la caja de más abajo) y esa posición guardar como pick2



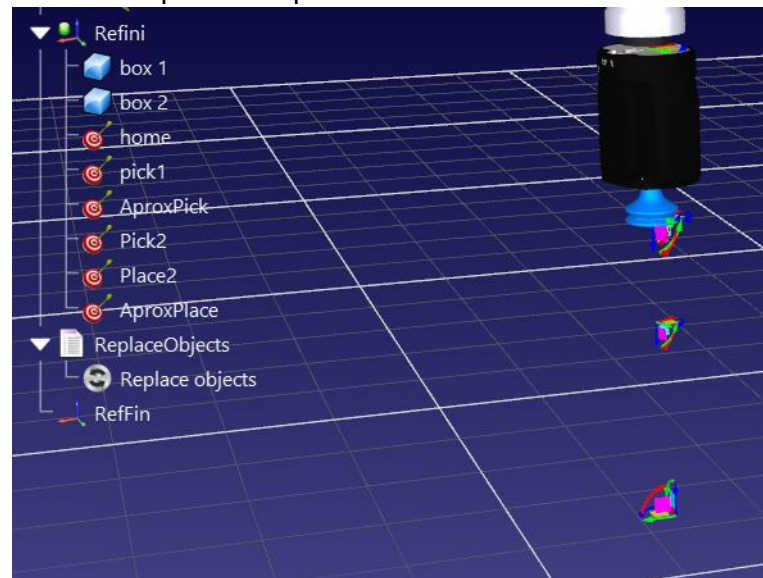
- vi Ahora, en el menú del robot mover el slider de la primera articulación de tal manera que se mueva hacia la referencia RefFin, y ahí guardaremos el target nuevo con el nombre place1



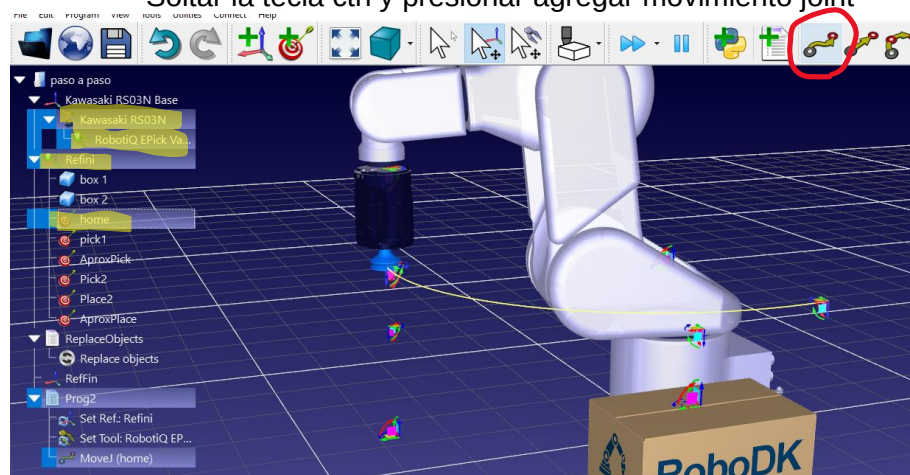
- vii Anotar ese valor del angulo de j1, ya que se usará para dejar a la otra caja exactamente sobre la primera. En mi caso es 11.91
- viii Ahora volver a hacer alt+doble clic en la caja de arriba y en la articulación 1 colocar el valor que a ustedes les quedó en mi caso 11.91, y guardar ese target con place2.



- ix Ahora subir de nuevo solo con el eje z al robot para guardar la última posición aproxPlace.

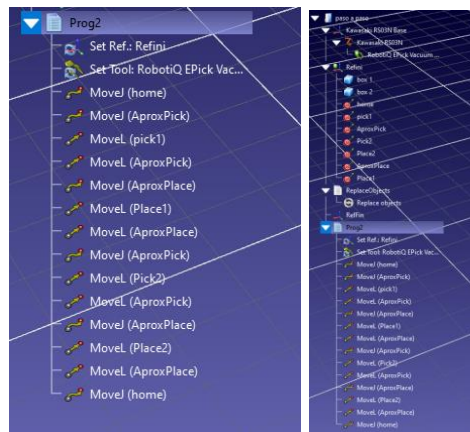


- x Con esto procederemos a crear un programa con cada uno de los targets creados.
- xi Para ello seleccionamos la referencia **Refini**, el **robot**, el **terminal** y el **target home** manteniendo presionada la tecla ctrl. Soltar la tecla ctrl y presionar agregar movimiento joint

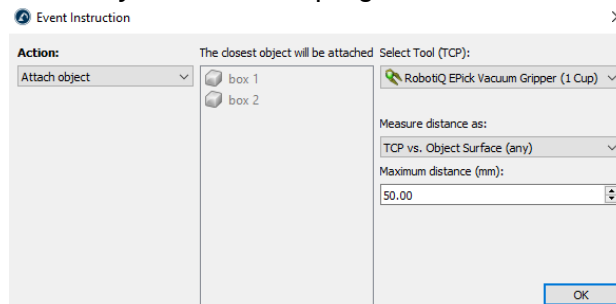




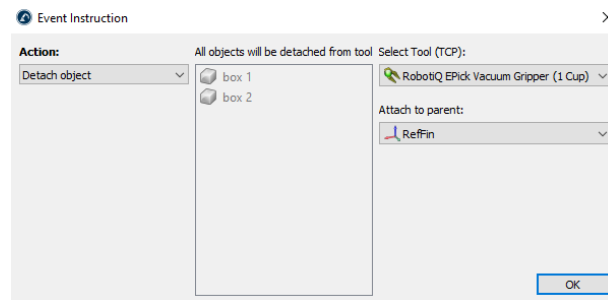
- xii Automáticamente se agrega un prog2 con la referencia seleccionada, el terminal del robot y el primer movimiento a home
- xiii A continuación, seleccionar manteniendo preionada la tecla ctrl el **Prog2** el joint **AproxPick** y seleccionar movimiento joint
- xiv A continuación, agregar movimiento lineal con el target **Pick1 y AproxPick**
- xv Ahora un movimiento joint con **AproxPlace** y movimiento lineal para Place1
- xvi Repetir con los respectivos tipos de movimientos y targets para hacer que la segunda caja quede sobre la primera caja y el robot finalmente regrese a la posición home, como se muestra a continuación.



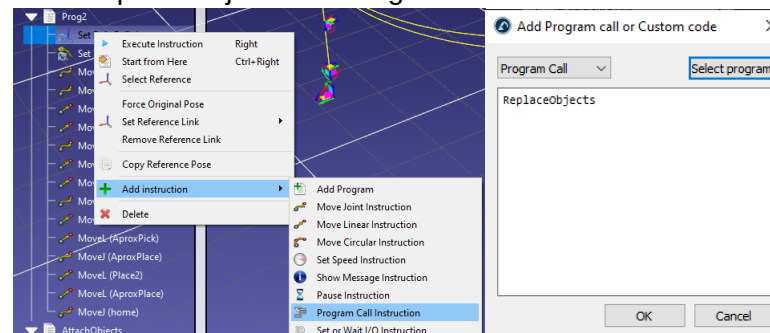
- xvii Ahora se va a crear un nuevo programa en el cual se va a crear una instrucción de simulación de evento de attachObject y se realizará la siguiente configuración, misma que permitirá agarrar el objeto mas cercano al terminal con 50 mm de distancia máximo, y renombrar el programa como attachObjects



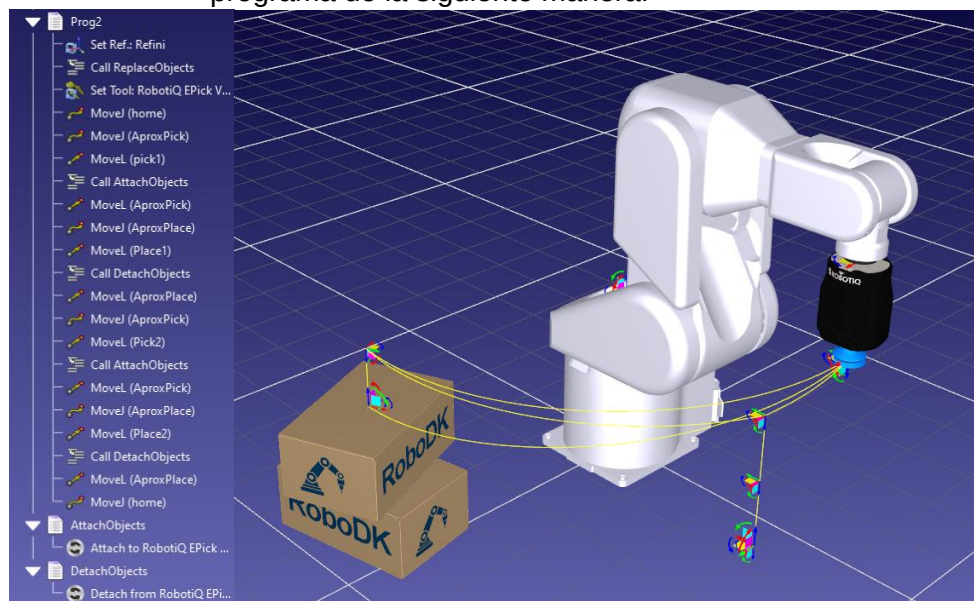
- xviii Ahora se va a crear un nuevo programa en el cual se va a crear una instrucción de simulación de evento de dettachObject y se realizará la siguiente configuración, misma que permitirá dejar el objeto dentro de un sistema de referencia en este caso seleccionaremos el RefFin, y renombrar el programa como dettachObjects



- xix En la primera línea del Prog2, se hará clic derecho, añadir nueva instrucción-instrucción de llamada a programa y se llamará a ReplaceObjects de la siguiente manera.



- xx De la misma forma llamaremos a attachObjects cuando se requiera y lo mismo para detachObjects, quedando finalmente el programa de la siguiente manera.



Finalmente observarán que las cajas quedan como muestra la imagen anterior.
Nota: corregir para que finalmente esa caja se coloque correctamente orientada igual que la caja de abajo.(corregir place2)