

## **MANUAL DE USO UR3**

### **Contenido:**

1. Caracterización general
2. Control externo por medio del lenguaje Python y la librería Socket

### **Caracterización general**

Para lograr la manipulación del brazo por medio de cualquier control externo es necesario comprender el funcionamiento básico del robot UR3. Este funcionamiento se centra en tres aspectos relacionados con los movimientos, grados de libertad y la configuración de las herramientas como se presentan a continuación:

#### **- Movimientos**

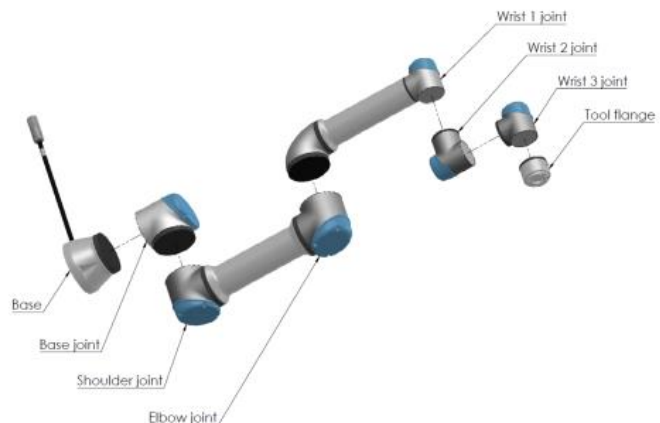
El brazo tiene tres tipos de movimiento:

- **MoveJ**: Movimiento general sin ruta trazada
- **MoveP**: Movimiento con ruta trazada (no lineales)
- **MoveL**: Movimiento lineal del TCP (Tool Center Point)

#### **- Grados de libertad**

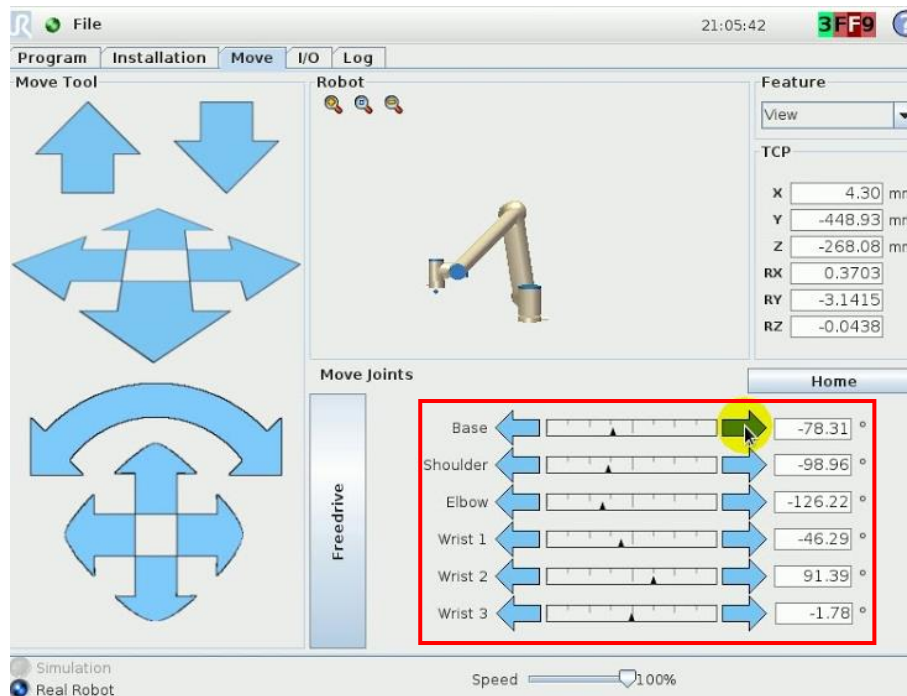
El brazo tiene seis grados de libertad:

- **Base**
- **Shoulder**
- **Elbow**
- **Wrist1**
- **Wrist2**
- **Wrist3**



*Ilustración 1 Juntas, base y brida de herramienta del brazo robótico*

El movimiento del brazo en general se realiza por la configuración de los ángulos de cada grado de libertad (unión del robot), definiendo posiciones específicas a las que se quiere llegar. Esto se observa en la imagen que se encuentra a continuación:



*Ilustración 2 Interfaz gráfica Polyscope*

#### - Herramientas

Las herramientas se configurarán generalmente de forma individual desde **Polyscope (Interfaz gráfica UR)**; sin embargo, en el uso general se podrá usar la salida `Tool_out[0]` para el control ON/OFF desde el puerto del brazo.

## Control externo

### ✓ Python – Socket

Por medio del protocolo Socket desde Python se pueden enviar instrucciones de control directamente al brazo (No requiere usar la interfaz gráfica de Polyscope). Estas instrucciones deberán seguir un formato específico. A continuación, se encuentra el proceso a ejecutar para el uso de esta funcionalidad:

#### 1. Configurar el servidor

Se configura el servidor como se muestra en la imagen inferior.

```
serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serversocket.bind(("localhost", 31001))
serversocket.listen(5)
```

**Nota:** 31001 corresponde a un puerto de entrada del sistema de control

#### 2. Configuración del usuario

Se configura el Host con la IP del robot (La IP se encuentra en la esquina izquierda superior en “About” en la interfaz gráfica del Robot).

```
HOST = "157.253.197.229"
PORT = 30002
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

**Nota:** 30012 corresponde a un puerto de entrada del sistema de control

#### 3. Inicialización de la conexión

Se conecta el usuario por medio del puerto definido previamente.

```
s.connect((HOST, PORT))
```

#### 4. Comandos

Para enviar comandos se utiliza la función *send()* del protocolo sockets . Es importante codificar los mensajes a enviar en *utf8*(Formato de codificación de texto). Un comando está dado por una función y el conjunto de parámetros que dicha función requiera. Ejemplo, la función *movej* requiere como *parámetros* una lista con las posiciones de los *ángulos de las juntas, la aceleración y la velocidad del movimiento*. Estas instrucciones deben enviarse junto con un *tiempo de espera* antes de continuar con la siguiente instrucción, ya que, en caso de no esperar, se enviarán instrucciones sin que se completen las previas.

```
s.send(("movej([-0.5, -0.94, 1.91, -0.90, 1.08, 3.90], a=1.0, v=0.3)" + "\n").encode("utf8"))
time.sleep(20)
```

**Notas:** La función *sleep()* hace referencia al tiempo de espera que se tendrá para la ejecución de la tarea antes de ejecutar la siguiente línea de código, es importante tener en cuenta este tiempo y su calculo aproximado ya que influye directamente en el movimiento, si no se pone este tiempo el robot para de manera abrupta.

Para configurar alguna de las salidas digitales o de herramienta es necesario usar la función *set\_digital\_out()*. Esta función requiere la *salida* a configurar y el *estado* (False -> apagado; True -> encendido). Cabe notar que para la salida de la herramienta es necesario configurar el voltaje de salida en 24V.

```
#digital 8 es la salida de la herramienta
s.send(("set_tool_voltage(24)" + "\n").encode("utf8"))
s.send(("set_digital_out(8,False)" + "\n").encode("utf8"))
```

**Nota:** Todas las funciones a utilizar corresponden a la librería de sockets

## 5. Documentación

- Comandos generales: <https://www.zacobria.com/universal-robots-knowledge-base-tech-support-forum-hints-tips-cb2-cb3/index.php/ur-script-send-commands-from-host-pc-to-robot-via-socket-connection/>
- Para las funciones revisar el archivo: "scriptManual.pdf"