Connect your robot with RoboDK (Kuka)

A connection between RoboDK and the robot can be accomplished to move the robot automatically from RoboDK. The connection can be established through Ethernet (TCP/IP).

1. Prepare the Kuka robot

We must follow these steps to prepare the communication between a Kuka robot and RoboDK. This method works with a KRC4 KUKA controller. The KRC4 robot controller runs the Microsoft Windows 7 operating system. The teach pendant shows an "HMI" which is a program that KUKA developed to run on Windows and it is the interface that the robot user has to manipulate the robot. We can use a USB flash drive to exchange the robot files (it is much faster to plug the USB drive on the robot controller than the teach pendant).

1. Connect a screen (DVI), a mouse and a keyboard to the controller.
2. Reboot the controller.
3. The screen will show a KukaUser windows login. Insert the credentials:

   user: kukauser   (or KukaUser)
   password: 68kuka1secpw59
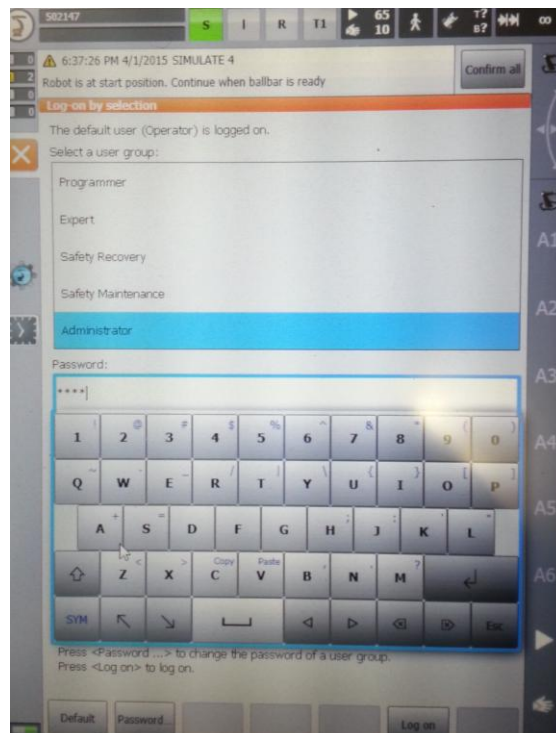
4. Alternatively, we can try a remote desktop connection or using the teach pendant's touch screen and the virtual keyboard.

5. If we see the KUKA HMI on the screen:
    a. KUKA→Configuration→User group→ choose Administrator (password: kuka)
    b. KUKA→Start-up→Service→Minimize HMI (we will see the windows screen)

6. Copy the KUKAVARPROXY folder in the Desktop (or somewhere in the PC).

7. Unlock the port 7000 for the server (HMI):
    a. Select the HMI.
    b. KUKA→Start-up→Network configuration→Advanced
    c. NAT→Add port→Port number 7000 .
    d. Set permitted protocols: tcp/udp

8. Start the KUKAVARPROXY.EXE program on the robot controller (running on Windows).

9. Follow these steps if we want this server to always start automatically on reboot (recommended):
    a. Create a shortcut of the KUKAVARPROXY.EXE file
    b. Select Windows START→All programs→Right click startup→Open
    c. Paste the shortcut in the startup folder

The KUKAVARPROXY server is now ready. It is not unsafe to leave this program running all the time. This server allows exchanging global variables from the KUKA controller to the remote PC.

We must now setup the main program that will handle the robot movements:

1. Add the declaration of these two global variables:
   To do so, we must modify the file "KRC\R1\STEU\$config.dat". If there are no variables set we can copy the $config.dat file. The folder "KC\R1\" can also be accessed from the C:\ drive at the following path: "C:\KRC\ROBOTER\KRC\"
   a. INT COM_ACTION=0
    b. DECL E6AXIS COM_E6AXIS

2. Copy the program MainCom.src in the folder "KRC\R1".

We can now manually start the MainCom.src program to move the robot with RoboDK (or from command line, see section 2). If this program is not running RoboDK will still be able to read the robot joints anytime as long as the KUKAVARPROXY program is running in the robot controller.

2. Test the robot communication in RoboDK:

We can right click our robot in RoboDK and select "Connect to robot...". A new nested window will open, as shown in the next image. We can enter the IP address or the COM port. The robot port is not

needed for Motoman communication. We can then select "Connect", we should read "Ready" in green if the communication works. Then we can test getting the current robot joints by pressing "Get current joints" or move the robot to a specific position the we preselected in the simulator by pressing "Move to current joints". In the RoboDK simulator we should see the exact same position as the real robot every time we press one of these two buttons.
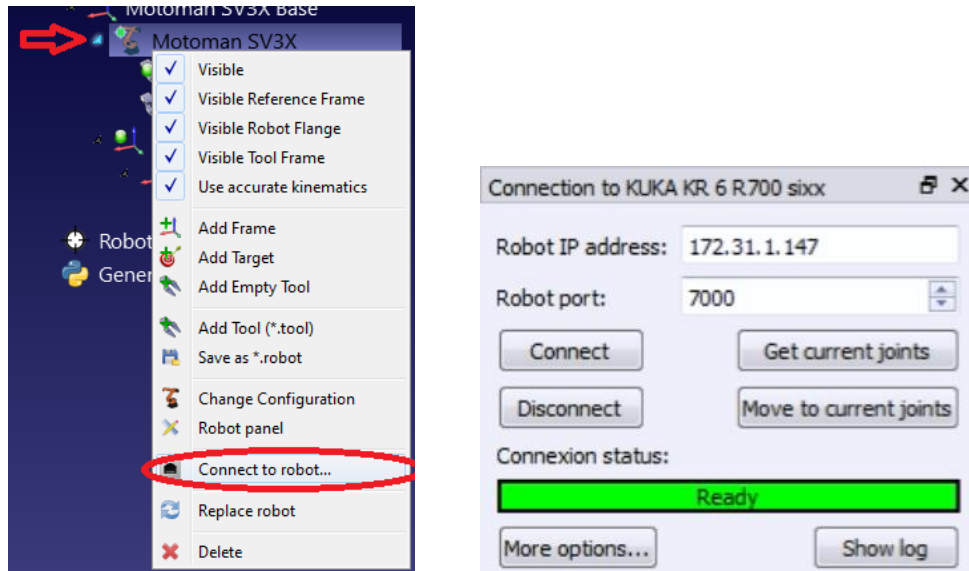
Continue reading and follow next steps if any problems happened during the connection.

The connection for each robot is managed by an executable that works as a driver. This driver can be tested in console mode. The drivers of each robot can be found in the folder "C:/RoboDK/api/robot/".

The IP address of the robot must be known (or the COM port for serial connections). Follow these steps to verify the communication between a robot and a computer (Ethernet connection only).

The goal is to be able to easily move the robot and read the current position of the robot inside RoboDK (step 3). If step 3 does not work we should go back to steps 1 and 2 to troubleshoot the communication problems.

1. Ping test: validates that the network parameters are correct.
    a. Open console: select [⊞] + [R] (windows key + R)→type "cmd"→select Ok.
    b. Type "ping IP" (where the IP should be the IP of your robot). Example:
       ```
       ping 192.168.125.2
       ```

c. We should see a response similar to the image with a 0% signal loss.

 change

If there is no communication we should verify that the IP of the robot and the PC are in the same local area network (LAN). We may have to deactivate the Windows Firewall or add a rule to allow the communication.

To make sure the laser tracker and the PC are in the same local area network we must set the IP of the computer to be the same as the robot except for the last number (such as 192.168.125.123). We can leave the subnet mask at 255.255.255.0. To change the IP of the computer we must follow these steps:

- Start→Control Panel→Network and Internet→Network Connections.
- Double click the local area connection driver.
- Select Properties.
- Select "Internet Protocol Version 4 (TCP/IPv4)"→Properties.
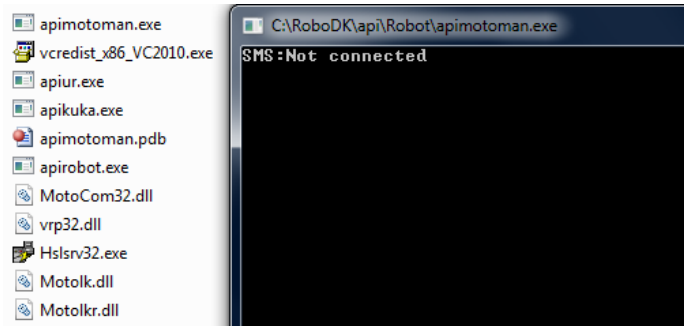- Enter the IP and subnet mask.
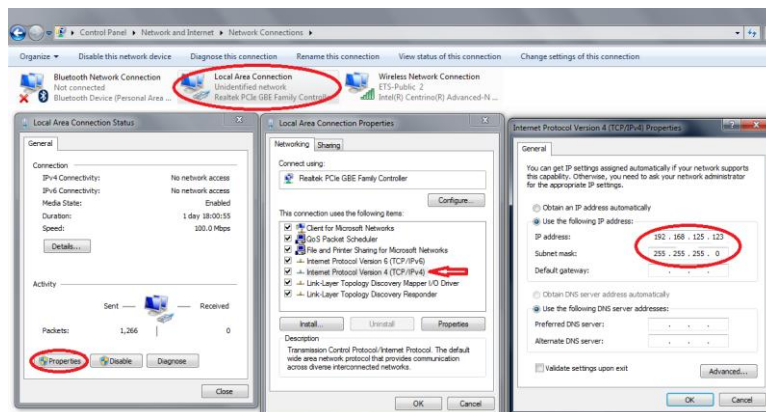
2. Test the robot driver:
   a. Open the folder that contains the robot driver binaries (C:/RoboDK/api/robot)
   b. Double click the "api.exe" file. The following window should appear:
      If an error message appears we must install the following files in order:
      i. "vcredist_x86_VS2008.exe".
      ii. "vcredist_x86_VS2013.exe"`

c. Enter "CONNECT IP" using the IP of your laser tracker or "CONNECT COM12" for serial connections.

Example:

```
CONNECT 192.168.125.2                    (Ethernet)
CONNECT COM12;Baud=9600;DataBits=8    (Serial)
```

Once the connection succeeded we should see the message "Ready".

Note: The following parameters are used by default for a serial connection: 19200 bps, 8 data bits, 1 stop bit and even parity. These parameters can be changed as shown in the serial example by using:

```
CONNECT COM12;Baud=4800;Parity=1;StopBits=0;DataBits=7
```

IMPORTANT: We may have to deactivate the Windows Firewall or add a rule to allow the communication; otherwise the communication may not work.
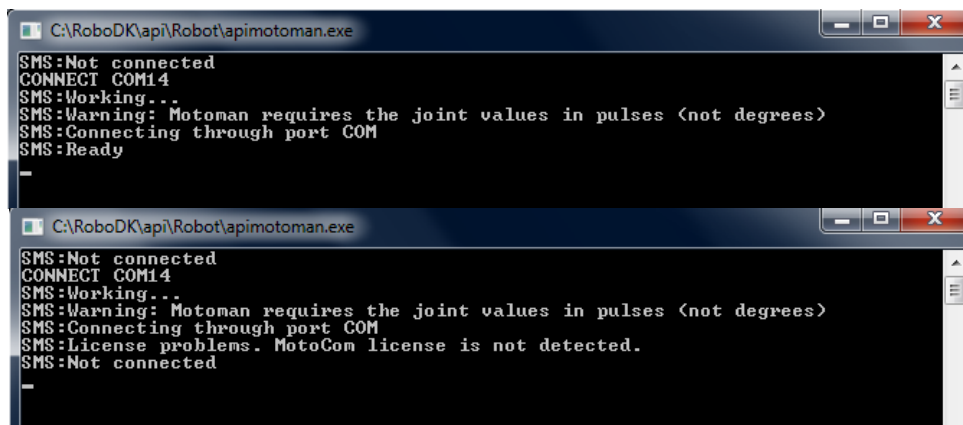
d. Test reading robot joints:
Once the laser tracker is ready, enter the command:
```
CJNT
```

The robot should provide the joints in the following format (Motoman uses pulses instead of degrees):
```
JNTS J1 J2 J3 J4 J5 J6
```

e. Test moving the robot:

Enter the following command to move the laser tracker towards a position close to the current position (movement in joint space).

```
MOVJ 10 20 -30 40 50 -60
```

The message "SMS:Ready" should appear once the operation is completed.