

## This package is for users of V-REP who would like to establish **remoteAPI** communication with **LabVIEW**:

Created: Ing. Peter Mačička

I am not responsible for damage caused on your hardware devices such as unexpected fire or explosions CD / DVD drives, keyboards breaking on the knee, my bad English and the critical timing of communication. ☺

For Labview 2011 and newer (Compatibility of older LabVIEW system please contact the author.)

In V-REP you have simple scene **<ControlTypeExamples>** is necessary open script for **remoteApiControlledBubbleRob** and change all his code with following:

```
simSetThreadSwitchTiming(2)
simDelegateChildScriptExecution()

-- Choose a port that is probably not used (try to always use a similar code):
simSetThreadAutomaticSwitch(false)
local portNb=simGetIntegerParameter(sim_intparam_server_port_next)
local portStart=simGetIntegerParameter(sim_intparam_server_port_start)
local portRange=simGetIntegerParameter(sim_intparam_server_port_range)
local newPortNb=portNb+1
if (newPortNb>=portStart+portRange) then
    newPortNb=portStart
end
simSetIntegerParameter(sim_intparam_server_port_next,newPortNb)
simSetThreadAutomaticSwitch(true)

-- Check what OS we are using:
platf=simGetIntegerParameter(sim_intparam_platform)
if (platf==0) then
    pluginFile='v_repExtRemoteApi.dll'
end
if (platf==1) then
    pluginFile='libv_repExtRemoteApi.dylib'
end
if (platf==2) then
    pluginFile='libv_repExtRemoteApi.so'
end

portNb = 20055
simExtRemoteApiStart(portNb)
```

After the launch of V-REP scene. You can run the VI: **Basescene** where control is created using a regular USB Gamepad (if you do not have it is necessary to remake the keyboard control or other control elements)

All you have to just run. The basic scene is done so that it is possible to create complex control systems.

### **If you have a problem, look below:**

If Labview cannot find the DLL file (and not found, it is not good to use the duplicate file, but rather directly from the installation of V-REP)

## Procedure:

Open one of the VI. (Fig. 1)

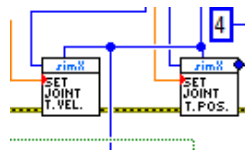


Fig. 1 Custom created Vi

Open Block diagram of the VI. (Fig. 2)

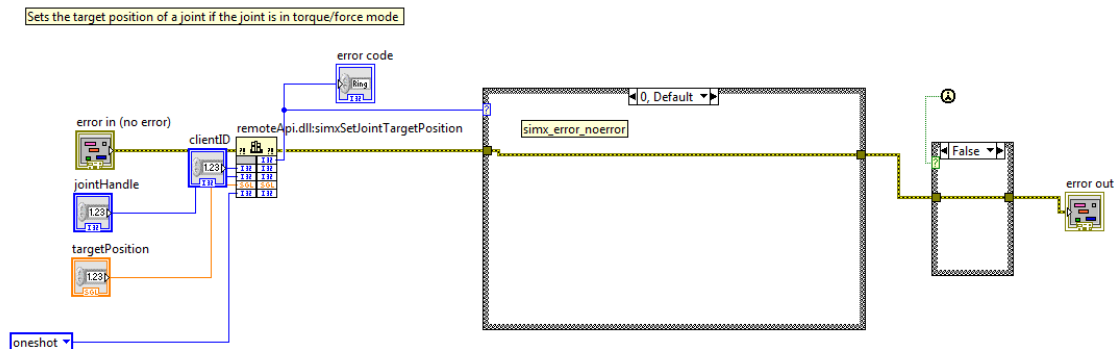


Fig. 2 Block diagram of the custom created Vi

Two times tap on function: **Call library function** (in this picture have name: remoteApi.dll:simxSetJointTargetPosition). This opens you settings dialog. (Fig. 3)

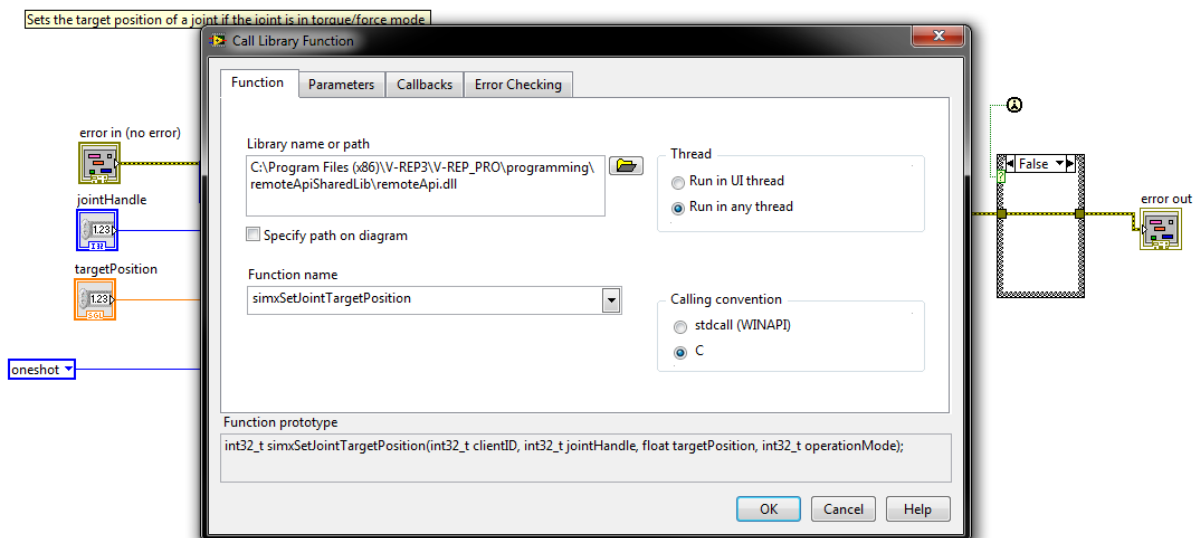


Fig. 3 Settings dialog for Call library function

On Fig. 3 it is Call library function where you must change **Library name or path**, for the location of your file.

When you can create your own function

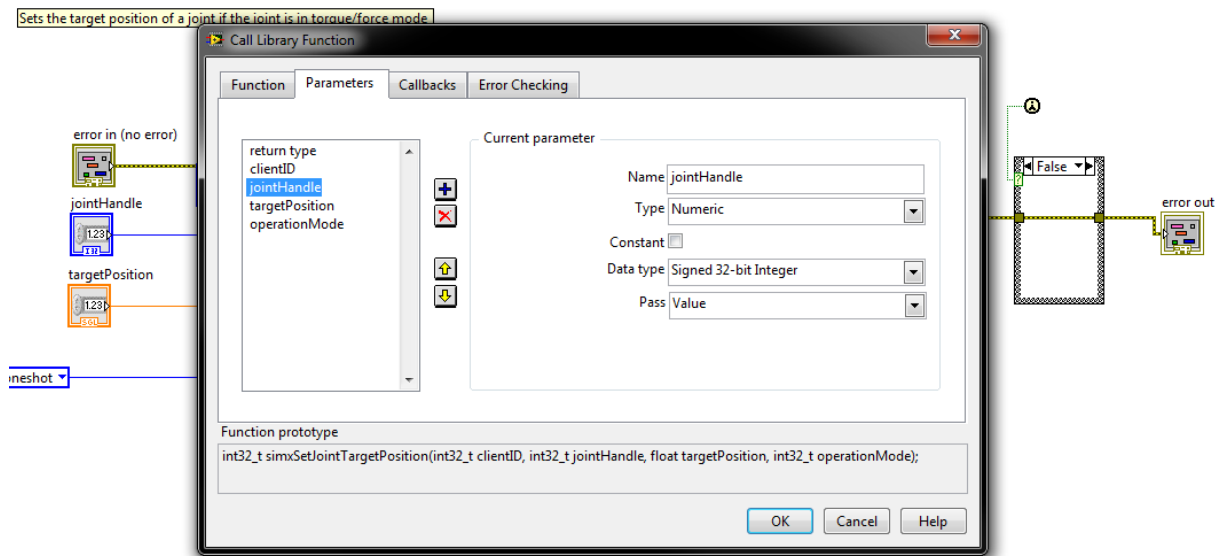


Fig. 4 Editing function prototype

To create another function you must change it in the (Fig. 3) menu: **function name** to select the function. Then it is necessary for her to fix the shape of which are set to: Fig. 4 according to Fig. 5

**simxSetJointTargetPosition**  
regular API equivalent: **simSetJointTargetPosition**  
ROS service equivalent: **simRosSetJointTargetPosition**

Description	Sets the target position of a joint if the joint is in torque/force mode (also make sure that the joint's motor and position control are enabled). See also <a href="#">simxSetJointPosition</a> .
C synopsis	<code>simxInt simxSetJointTargetPosition(simxInt clientID, simxInt jointHandle, simxFloat targetPosition, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>targetPosition</b> : target position of the joint (angular or linear value depending on the joint type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_oneshot</code> or <code>simx_opmode_streaming</code>
C return value	a <a href="#">remote API function error code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Urbi</a>

Fig. 5 V/REP user manual function description on <http://v-rep.eu/helpFiles/index.html>

The attached file contains only a **selected few basic functions** that are sufficient for the establishment of the basic control algorithms. In the event of problems that this guide forgotten or **looking for more** features containing in this package, please contact:

Site forum: <http://www.forum.coppeliarobotics.com/index.php>

I have also Vi for stereoscopic vision (3D glasses) and 3D laser scanners imaging and processing data in LabVIEW. but I would like to address only advanced users.

**Thanks:** Dr. Marc Andreas Freese - for help with some problems.

Contact: **Please write your problems on forum** or if its urgently: [macickap@ztsvvu.eu](mailto:macickap@ztsvvu.eu)