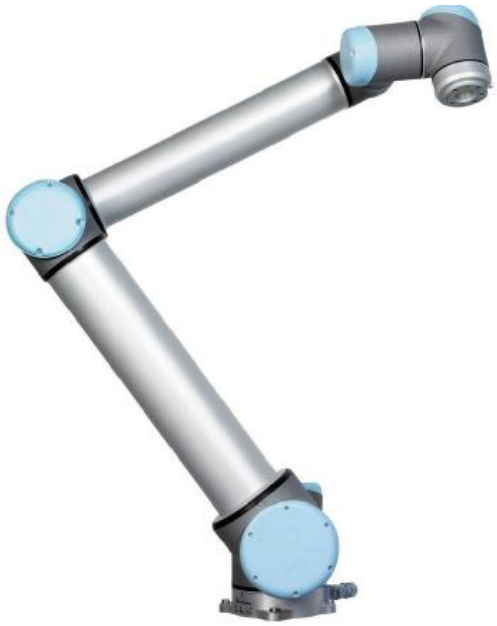
# 优傲机器人技术应用方案
# Universal Robots Application Proposal

方案名称：优傲机器人 **socket** 通信脚本解释



**UR3, UR5 & UR10/CB3**

优傲机器人贸易（上海）有限公司
**Universal Robots(Shanghai) Co.,Ltd.**

**2018 年 8 月 15 日**

**Contains Confidential Information of UR, do not distribute without permission**

| SUMMARY OF DOCUMENT REVISIONS | | | |
|---|---|---|---|
| **Rev. No.** | **Date Revised** | **Section Revised** | **Revision Description** |
| **0** | 2018.8.15 | Draft | Issured by jason xu |

**Contains Confidential Information of UR, do not distribute without permission**

1.1 SOCKET 介绍 ..... 错误!未定义书签。

**2.0** 打开关闭脚本 ..... 4

2.1 打开端口 ..... 4

2.2 关闭端口 ..... 4

**3.0** 接收脚本 ..... 5

3.1 读 ASCII 字符装换为浮点数 ..... 5

3.2 读字符串 ..... 6

3.3 读每 32 位整数保存到列表 ..... 7

3.4 读每字节（8 位）整数保存到列表 ..... 7

**4.0** 发送脚本 ..... 8

4.1 将整数以字节长度发送到服务端 ..... 8

4.2 将整数以 32 位长度发送到服务端 ..... 8

4.3 发送以换行符结尾的字符串到服务端 ..... 9

4.4 发送字符串到服务端 ..... 9

**Contains Confidential Information of UR, do not distribute without permission**

# 1.0 SOCKET 通信介绍

Socket 通讯常用于机器人和相机等设备交互数据，优傲机器人 socket 通信编程只能作为客户端。当打开机器人端口时需要给定服务端 IP 及端口号，机器人可以和多个服务端通讯。当仅连接一个服务端时可以省略 socket_name 参数。当机器人需要和多个服务端通讯时需要给端口指定名字，以便后续区分不同端口。两台机器人如果需要交互信号可以采用 modbus 连接。由于 socket 通信机器人可以作为客户端，说以机器人可以连接其他机器人发送 dashboard 指令，控制其他机器人。

本文档使用 socket 调试助手和机器人虚拟机测试。调试助手可以收发字符串（文本形式）以及字节流（16 进制形式）。

# 2.0 打开关闭脚本

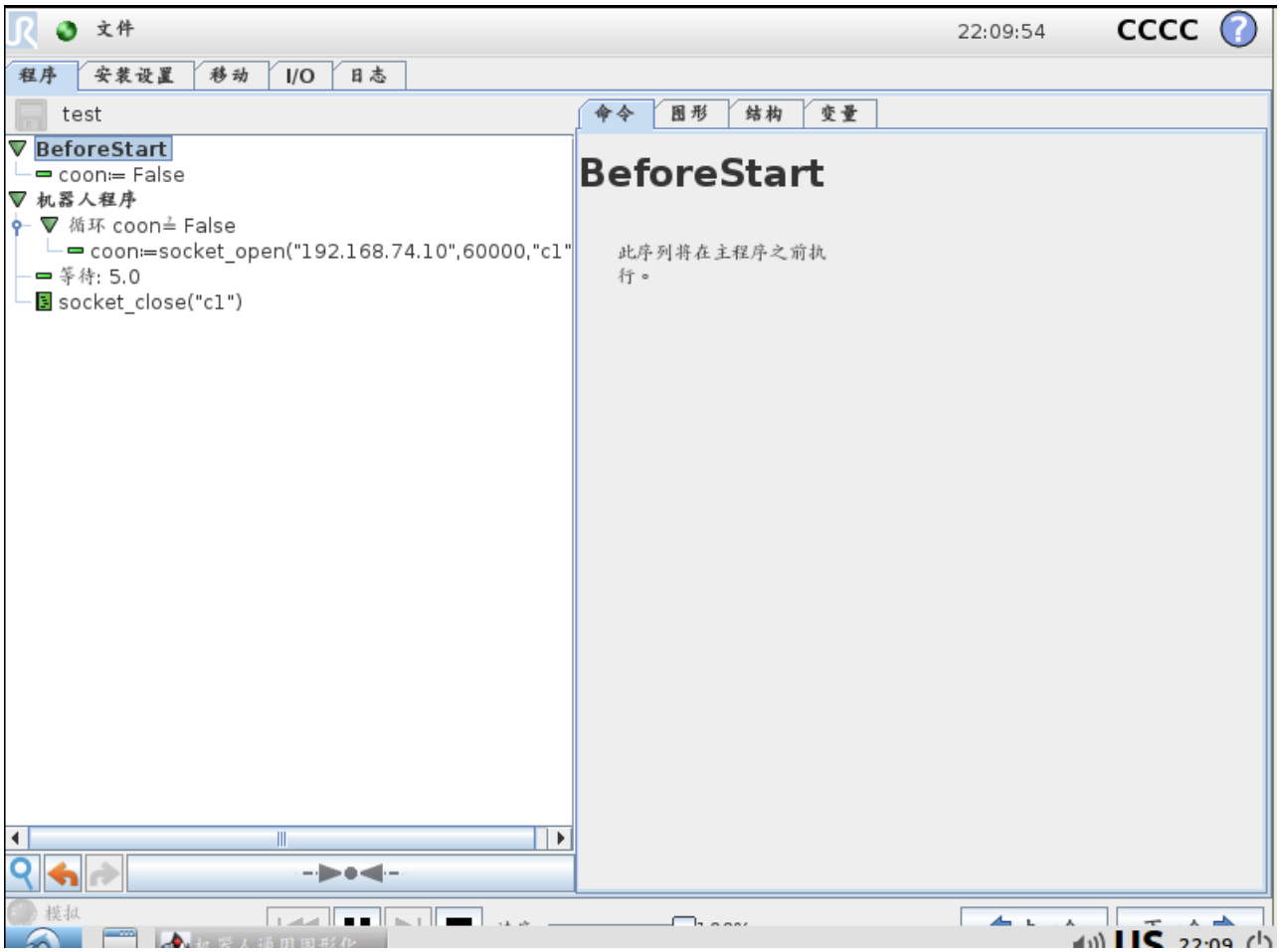
## 2.1 打开端口

**socket_open**(*address, port, socket_name='socket_0'*)
打开以太网通信。
试图打开端口连接，2秒后超时。
参数
address：服务器地址（字符串）
port：端口号（整数）
socket_name：端口的名称（字符串）
返回值
如果失败，则为 False；如果成功建立连接，则为 True

## 2.2 关闭端口

**socket_close**(*socket_name='socket_0'*)
关闭以太网通信。
关闭与服务器的端口连接。
>>> socket_comm_close()
参数
socket_name：端口的名称（字符串）

下图是机器人打开和关闭端口脚本。

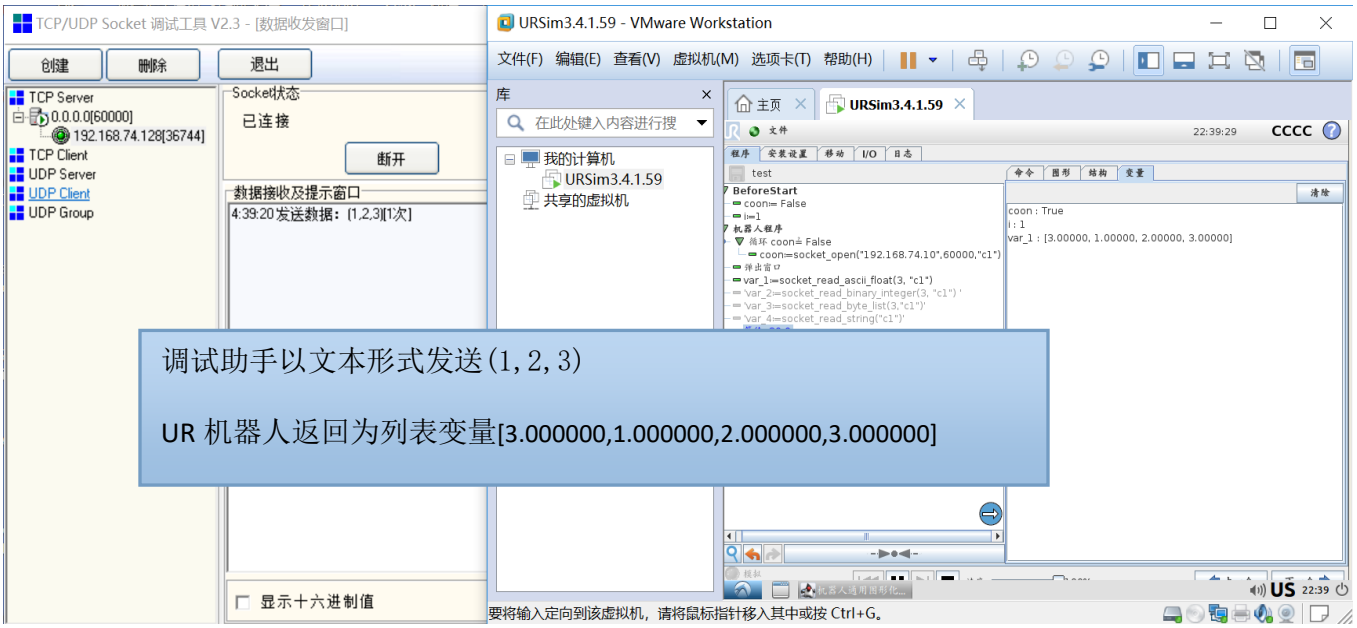



# 3.0 接收脚本

## 3.1 读ASCII字符串转换为浮点数

**socket_read_ascii_float**(*number, socket_name='socket_0'*)
读取来自所连接TCP/IP的多个ASCII格式化标记。一个命令中最多可读取30个数值。
>>> list_of_four_floats = socket_read_ascii_float(4)
编号的格式应当在括号内，并且通过“,”隔开。四个编号的示例列表如下所述：“（1.414, 3.14159, 1.616, 0.0）”。
返回的列表含有读取的所有编号，然后每个编号依次排列。例如，上述示例的read_ascii_float应当返回[4, 1.414, 3.14159, 1.616, 0.0]。
读取失败或2秒后超时将返回如下所述的列表：以0作为第一元素，随后的元素中为“非编号（nan）”（例如，对于三个编号的读取：[0, nan., nan, nan]）。
参数
number：要读取的变量个数（整数）
socket_name：端口的名称（字符串）

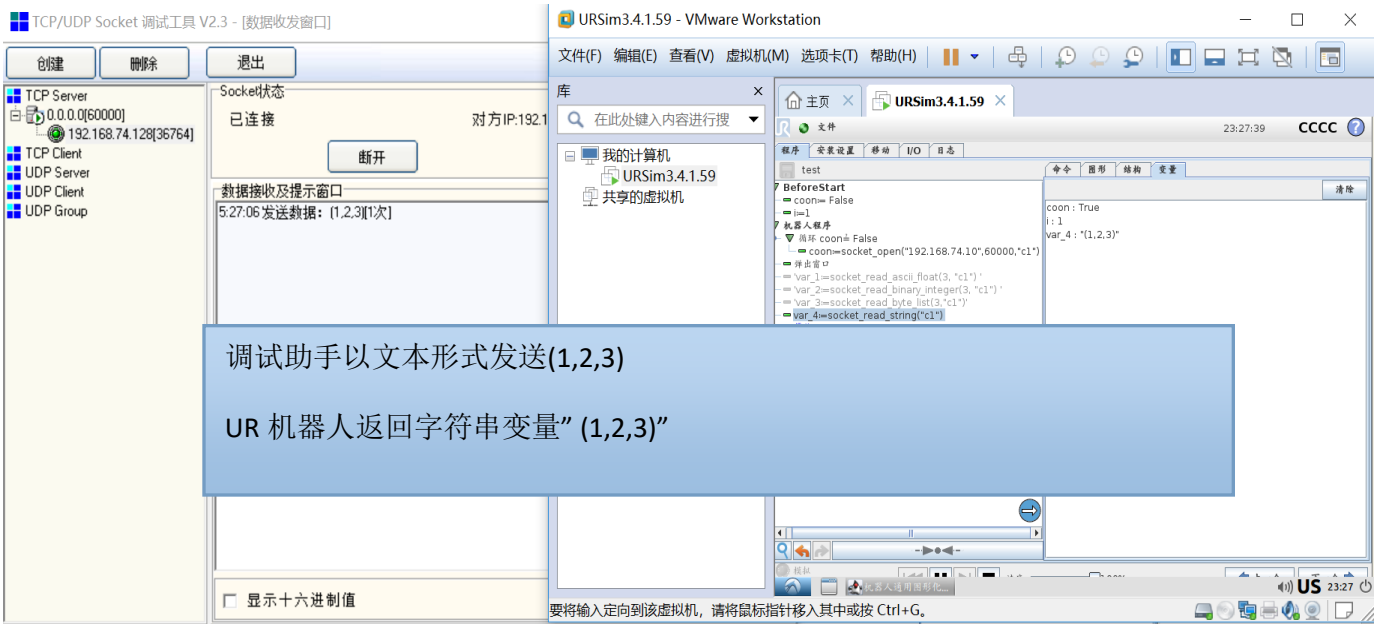
返回值
读取的编号列表（浮点数列表，长度＝个数+1）




## 3.2 读字符串

**socket_read_string**(*socket_name='socket_0'*)

读取来自所连接TCP/IP的字符串。字节为网络字节顺序。
>>> string_from_server = socket_read_string()
返回（例如）“reply string from the server”（返回来自服务器的字符串），如果超时（2秒）或者回复无效，则返回空的字符串（“”）。您可以通过if语句测试字符串是否为空。
>>> if(string from server) :
>>> popup("the string is not empty")
>>> end
参数
socket_name：端口的名称（字符串）
返回值
字符串变量




调试助手以文本形式发送(1,2,3)

UR 机器人返回字符串变量” (1,2,3)”

## 3.3 读每32位整数保存到列表

**socket_read_binary_integer**(*number, socket_name='socket_0'*)
读取来自所连接TCP/IP的多个32位整数。字节为网络字节顺序。一个命令中最多可读取30个数值。
>>> list_of_three_ints = socket_read_binary_integer(3)
返回（例如）[3,100,2000,30000]，如果超时（2秒）或者回复无效，则返回[0,-1,-1,-1]，表示已经读取了0个整数。
参数
number：要读取的变量个数（整数）
socket_name：端口的名称（字符串）
返回值
读取的编号列表（整数列表，长度＝个数+1）

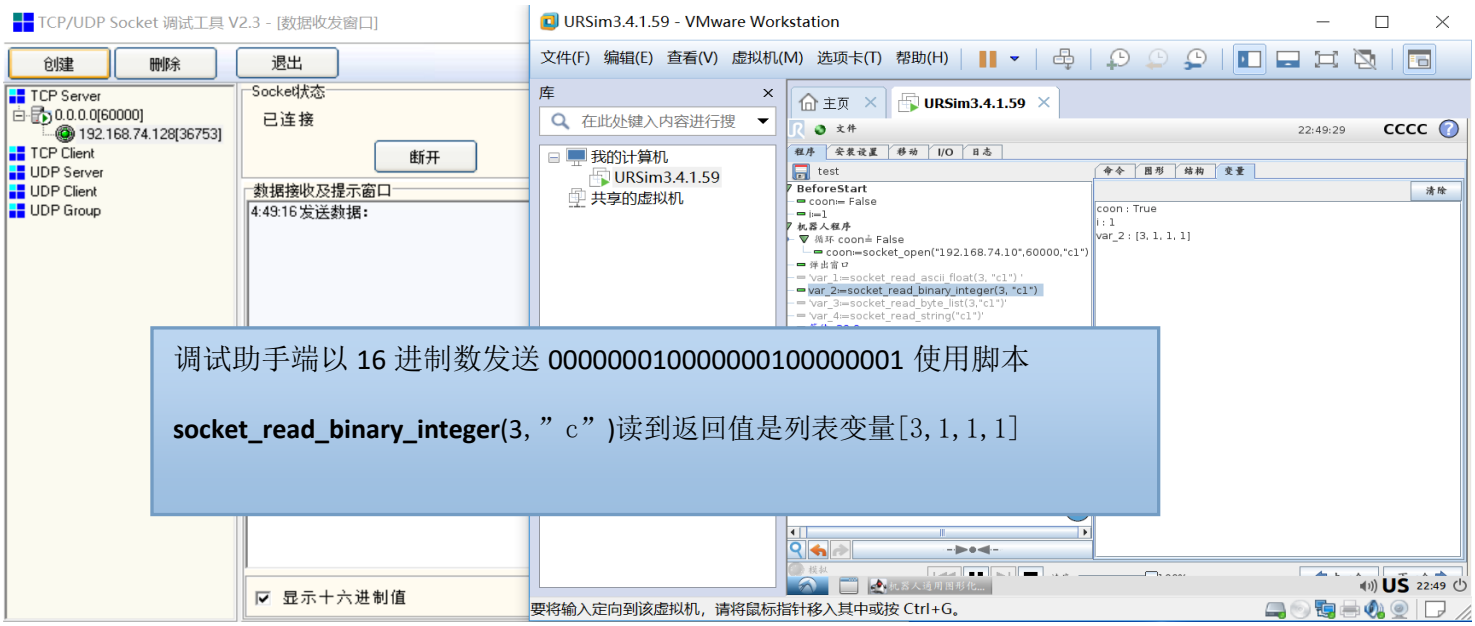
按照 32 位一个整数读取数据并存入列表变量。

## 3.4 读每字节（8位）整数保存到列表

**socket_read_byte_list**(*number, socket_name='socket_0'*)
读取来自所连接TCP/IP的多个字节。字节为网络字节顺序。一个命令中最多可读取30个数值。
>>> list_of_three_ints = socket_read_byte_list(3)
返回（例如）[3,100,200,44]，如果超时（2秒）或者回复无效，则返回[0,-1,-1,-1]，表示已经读取了0个字节。

参数
number：要读取的变量个数（整数）
socket_name：端口的名称（字符串）
返回值
读取的编号列表（整数列表，长度＝个数+1）

按照 8 位一个整数读取数据并存入列表变量。




# 4.0 发送脚本

## 4.1 将整数以字节长度发送到服务端

**socket_send_byte**(*value, socket_name='socket_0'*)
将字节发送给服务器。
通过端口发送字节<value>。预期无响应。可用于发送特殊的ASCII字符；10是换行，2是文本开始，3是文本结束。
参数
value：要发送的编号（字节）
socket_name：端口的名称（字符串）

## 4.2 将整数以32位长度发送到服务端

**socket_send_int**(*value, socket_name='socket_0'*)
将整数（int32_t）发送给服务器。
通过端口发送整数<value>。以网络字节顺序发送。预期无响应。
参数
value：要发送的编号（整数）

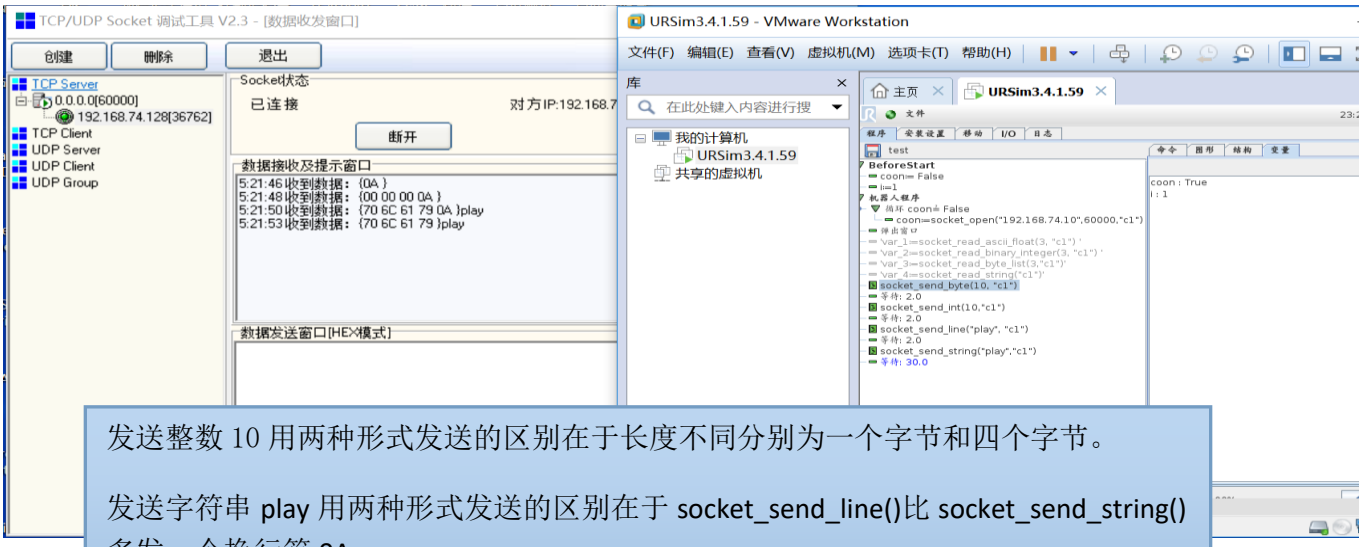
socket_name：端口的名称（字符串）

## 4.3 发送以换行符结尾的字符串到服务端

**socket_send_line**(*str, socket_name='socket_0'*)
将带换行符的字符串发送给服务器– 适用于与UR仪表板服务器的通信。
通过端口以ASCII编码发送字符串<str>。预期无响应。
参数
str：要发送的字符串（ASCII）
socket_name：端口的名称（字符串）

## 4.4 发送字符串到服务端

**socket_send_string**(*str, socket_name='socket_0'*)
将字符串发送给服务器。
通过端口以ASCII编码发送字符串<str>。预期无响应。
参数
str：要发送的字符串（ASCII）
socket_name：端口的名称（字符串）




发送整数 10 用两种形式发送的区别在于长度不同分别为一个字节和四个字节。

发送字符串 play 用两种形式发送的区别在于 socket_send_line()比 socket_send_string()多发一个换行符 0A。

**Contains Confidential Information of UR, do not distribute without permission**