

# ABB机器人socket通讯

工业机器人小白菜 于 2021-03-10 16:19:20 发布

## 文章目录

### 前言

#### 一、效果展示

#### 二、知识储备

##### 1.准备工作

##### 2.套接字通信的示意图

##### 3.客户端指令

##### 4.服务器指令

#### 三、源码说明

---

## 前言

**Socket** Messaging的作用是允许RAPID程序员通过TCP/IP网络协议在各台计算机之间传输应用数据。一个**套接字**代表了一条独立于当前所用网络协议的通用通信通道。

“套接字通信”是源于Berkeley所发布软件Unix的一套标准，而除Unix外，Microsoft Windows等平台也支持该项标准。有了Socket Messaging，机器人控制器上的RAPID程序就能与另一台计算机上的C/C++程序等进行通信。

## 一、效果展示

看不清视频请点链接

内容来源: [csdn.net](https://www.csdn.net/)

作者昵称: 工业机器人小白菜

原文链接: <https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页: <https://blog.csdn.net/kukakkkkkkukakkkkk>

ABB机器人socket通讯 (机器人做clint)

去bilibili观看 分享

UP 气球不是圆的

播放: 2316 弹幕: 4

扫一扫 手机看



00:30 / 00:30 360P

进入bilibili,一起发弹幕吐槽! 去吐槽

ABB机器人socket通讯 (机器人做clint)

机器人做clint

ABB机器人socket通讯 (机器人做Server)

去bilibili观看 分享

扫一扫 手机看



00:43 / 00:43 360P

进入bilibili,一起发弹幕吐槽! 去吐槽

ABB机器人socket通讯 (机器人做Server)

内容来源: csdn.net

作者昵称: 工业机器人小白菜

原文链接: <https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页: <https://blog.csdn.net/kukakkkkkkukakkkkk>

## 二、知识储备

### 1.准备工作

robostudio:ABB机器人编程软件

TCP/UDP调试助手：可模拟服务器客户端进行本体测试

选项支持：616-1 PC-inteface

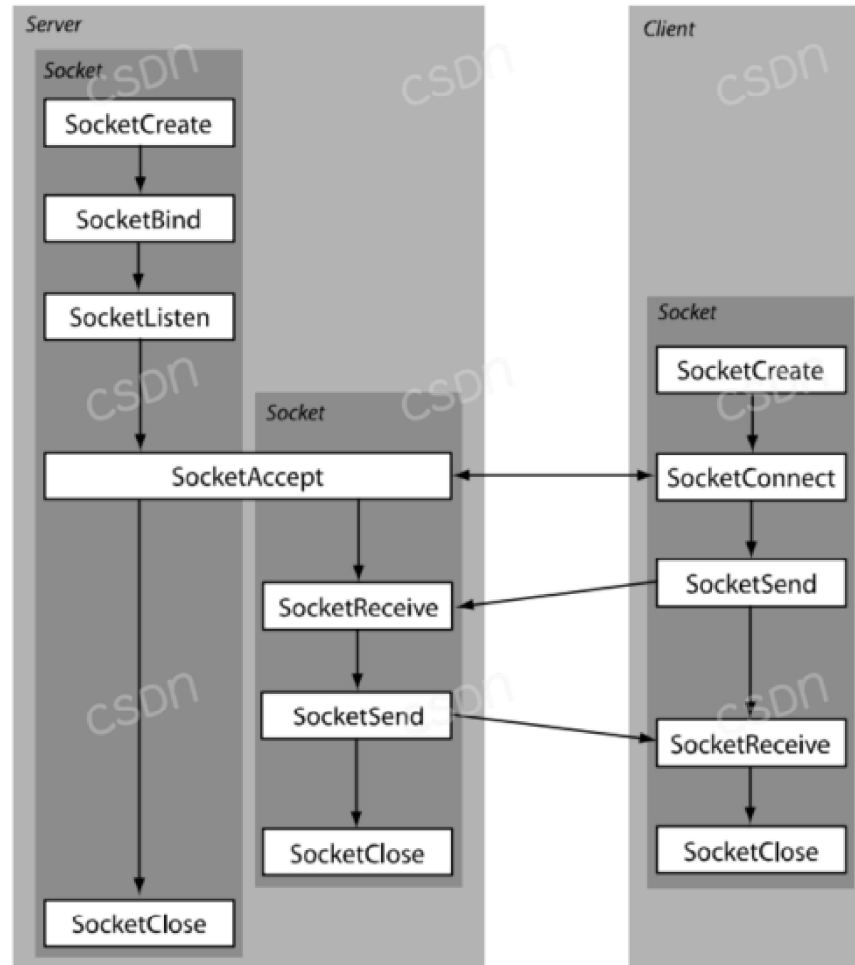
### 2.套接字通信的示意图

内容来源：[csdn.net](https://www.csdn.net)

作者昵称：工业机器人小白菜

原文链接：<https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页：<https://blog.csdn.net/kukakkkkkkukakkkkk>



en0600003224



**提示**

若无必要，则请勿创建和关闭套接字。通信完毕前请一直开启相应的套接字。出于TCP/IP功能之故，在`SocketClose`后，该套接字要过一段时间才会真正关闭。

<https://blog.csdn.net/kukakkkkk>

### 3.客户端指令

内容来源: csdn.net

作者昵称: 工业机器人小白菜

原文链接: <https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页: <https://blog.csdn.net/kukakkkkkkukakkkkk>

指令	描述
套接字创建	创建一个新的套接字，并赋予其一个 <code>socketdev</code> 变量。
套接字连接	向一台远程计算机提出连接请求。客户端将用其连接相应的服务器。
套接字发送	通过套接字连接而向某台远程计算机发送数据。这些数据既可以是 <code>string</code> 或 <code>rawbytes</code> 变量，也可以是 <code>byte</code> 数组。
套接字接收	接收数据，并将其保存在一个 <code>string</code> 或 <code>rawbytes</code> 变量中，或保存在一个 <code>byte</code> 数组中。
套接字关闭	关闭一个套接字，随之释放所有资源。



提示

不要在`SocketSend`之后直接使用`SocketClose`。先等候接受信息，然后再关闭套接字。

<https://blog.csdn.net/kukakkkk>

## 4.服务器指令

除`SocketConnect`外，套接字消息发送服务器与客户端都使用着同一套指令。此外服务器还会使用下列指令：

指令	描述
套接字绑定	将套接字与相关服务器上的一个指定端口号绑定起来。该服务器会用它来定义“用（该服务器上的）哪个端口监听某一连接”。该IP地址定义了一台物理计算机，而该端口则定义了通往该计算机上某一程序的一条逻辑通道。
套接字监听	使该计算机作为一台服务器，并接受外来的连接。其将监听 <code>SocketBind</code> 所指定端口上的某一连接。
套接字接受	接受一项外来连接请求。服务器将用其来接受相关客户端的请求。



注意

必须先启动服务器应用，然后才启动客户端应用，这样才能在任一客户端执行`SocketConnect`前先执行指令`SocketAccept`。

<https://blog.csdn.net/kukakkkk>

## 三、源码<sup>Q</sup>说明

Client代码如下：

内容来源：csdn.net

作者昵称：工业机器人小白菜

原文链接：<https://blog.csdn.net/kukakkkk/article/details/114637639>

作者主页：<https://blog.csdn.net/kukakkkkkukakkkk>

```

1  VAR socketdev temp_socket;
2  VAR socketdev client_socket;
3  VAR string received_string;
4  VAR bool keep_listening:=TRUE;
5  VAR socketdev socket1;
6
7
8  PROC rClient()
9      SocketCreate socket1;    !创建套接字
10     SocketConnect socket1,"192.168.19.193",6000\Time:=10;    !连接到服务器,端口号6000
11     SocketSend socket1\Str:="Hello server";    !发送String数据
12     SocketReceive socket1\Str:=received_string;    !接收String数据
13     TPWrite "Server wrote - "+received_string;
14     received_string:="";
15     SocketSend socket1\Str:="Shutdown connection";    !发送String数据
16     SocketReceive socket1\Str:=received_string;    !接收String数据
17     TPWrite "Server wrote - "+received_string;
18     SocketClose socket1;    !关闭连接
19
20     ERROR
21     IF ERRNO=ERR_SOCKET_TIMEOUT THEN    !连接超时错误处理
22         RETRY;    !重试
23     ELSEIF ERRNO=ERR_SOCKET_CLOSED THEN    !套接字被关闭错误处理
24         SocketClose socket1;
25         WaitTime 10;    !等待10S再次连接,服务器可能第一次启动
26         SocketCreate socket1;
27         SocketConnect socket1,"192.168.19.193",6000;
28         RETRY;
29     ELSE
30         TPWrite "ERRNO = "\Num:=ERRNO;
31         Stop;
32     ENDIF
33 ENDPROC

```

Server代码如下:

```

1  PROC rServer()
2      SocketCreate temp_socket;    !创建服务器套接字
3      SocketBind temp_socket,"192.168.19.193",6000;    !绑定IP地址与端口号

```

内容来源: [csdn.net](https://blog.csdn.net/kukakkkkk/article/details/114637639)

作者昵称: 工业机器人小白菜

原文链接: <https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页: <https://blog.csdn.net/kukakkkkkkukakkkkk>

```

4      SocketListen temp_socket;      !开始监听连接
5      WHILE keep_listening DO
6          SocketAccept temp_socket,client_socket; !接受输入连接请求
7          SocketReceive client_socket\Str:=received_string; !接收String数据
8          TPWrite "Client wrote - "+received_string;
9          received_string:="";
10         SocketSend client_socket\Str:="Message acknowledged"; !发送String数据
11         SocketReceive client_socket\Str:=received_string; !接收String数据
12         TPWrite "Client wrote - "+received_string;
13         SocketSend client_socket\Str:="Shutdown acknowledged"; !发送String数据
14         SocketClose client_socket;      !关闭输入连接请求
15     ENDWHILE
16     SocketClose temp_socket; !关闭服务器套接字
17
18     ERROR
19     IF ERRNO=ERR_SOCKET_TIMEOUT THEN
20         RETRY;
21     ELSEIF ERRNO=ERR_SOCKET_CLOSED THEN
22         SocketClose temp_socket;
23         SocketClose client_socket;
24         SocketCreate temp_socket;
25         SocketBind temp_socket,"192.168.19.193",6000;
26         SocketListen temp_socket;
27         SocketAccept temp_socket,client_socket;
28         RETRY;
29     ELSE
30         TPWrite "ERRNO = "\Num:=ERRNO;
31         Stop;
32     ENDIF
33 ENDPROC
34

```

内容来源: csdn.net

作者昵称: 工业机器人小白菜

原文链接: <https://blog.csdn.net/kukakkkkk/article/details/114637639>

作者主页: <https://blog.csdn.net/kukakkkkkkukakkkkk>