



使用手册

优爱宝机器人
SDK 介绍

优爱宝机器人 SDK

[知识产权保护声明]

使用UIROBOT产品前请注意以下三点:

- UIROBOT的产品均达到UIROBOT使用手册中所述的技术功能要求。
- UIROBOT愿与那些注重知识产权保护的客户合作。
- 任何试图破坏UIROBOT器件代码保护功能的行为均可视为违反了知识产权保护法案和条例。如果这种行为导致在未经UIROBOT授权的情况下, 获取软件或其他受知识产权保护的成果, UIROBOT有权依据该法案提起诉讼制止这种行为。

[免责声明]

本使用手册中所述的器件使用信息及其他内容仅为为您提供便利, 它们可能在未来版本中被更新。确保应用符合技术规范, 是您自身应负的责任。UIROBOT对这些信息不作任何形式的声明或担保, 包括但不限于使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。UIROBOT对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将UIROBOT器件用于生命维持和/或生命安全应用, 一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时, 会维护和保障UIROBOT免于承担法律责任和赔偿。未经UIROBOT同意, 不得以任何方式转让任何许可证。

[商标和外观设计声明]

UIROBOT 的名称和徽标组合为 UIROBOT Ltd.在中国和其他国家或地区的注册商标。
UIROBOT的SCARA机器人设计已申请专利保护。

[联系方式]

上海优爱宝智能机器人科技股份有限公司
地址: 上海浦东新区亮秀路 112 号 Y2 座 202-203 室
电话: 021 - 61182435 (销售/市场); 61182432(总机)
传真: 021 - 61182431
邮箱: info@uirobot.com
网址: www.uirobot.com



请扫二维码，关注优爱宝微信平台

目 录

1.0	引言	6
1.1	编写目的	6
1.2	参考资料	6
2.0	运行环境	7
2.1	硬件环境	7
2.2	软件环境	7
3.0	编程模型	8
4.0	安装方法	10
4.1	机器人软件安装	10
4.2	机器人软件卸载	13
4.3	动态调用	14
4.3.1	VC	14
4.3.2	VB	14
4.3.3	C#	14
4.4	静态调用	15
4.4.1	VC	15
5.0	数据结构	16
5.1	UI_SETPASSWORD_OBJ	16
5.2	UI_NETWORKCONFIG_OBJ	17
5.3	UI_DATETIME_OBJ	18
5.4	UI_RUNNINGPROGRAM_OBJ	19
5.5	UI_COMMANDBUFFER_OBJ	20
5.6	UI_MOTORTEMPERATURE_OBJ	21
5.7	UI_ROBOTTCP_OBJ	22
5.8	UI_INITIALIZETCP_OBJ	23
5.9	UI_JOINTANGLE_OBJ	24
5.10	DIGITAL_VALUE_OBJ	25
5.11	UI_FILECONTENT_OBJ	27
5.12	UI_SOCKETMSG_OBJ	28
5.13	TCP	29
5.14	NodePVT	30
5.15	RobotPVTLists	31
6.0	函数接口	32
6.1	机器人连接及关闭	32
6.1.1	回调函数注册函数	32
6.1.2	根据连接值连接机器人	32
6.1.3	根据连接字符串连接机器人	33
6.1.4	断开机器人连接	33
6.1.5	关闭机器人	33
6.2	机器人登陆及设置	34
6.2.1	登陆机器人	34
6.2.2	设置机器人密码	34
6.2.3	设置机器人网络配置	34
6.2.4	设置机器人时间	35
6.2.5	设置机器人默认用户程序	35
6.2.6	设置机器人类型	35
6.3	用户获取机器人信息	37
6.3.1	获取机器人网络配置	37
6.3.2	获取机器人时间	37

6.3.3	获取机器人软件版本号	37
6.3.4	获取机器人工作状态	38
6.3.5	获取机器人当前程序信息	38
6.3.6	获取用户程序列表	38
6.3.7	删除机器人指定用户程序	39
6.3.8	获取机器人默认用户程序	39
6.3.9	获取机器人各个臂的温度	39
6.3.10	获取机器人类型	40
6.4	机器人程序	41
6.4.1	运行机器人用户程序	41
6.4.2	模拟运行机器人用户程序	41
6.4.3	机器人紧急停止	41
6.4.4	复位机器人	42
6.4.5	设置机器人夹具中心点	42
6.4.6	获取机器人夹具中心点	42
6.4.7	设置机器人初始工具位置	43
6.4.8	获取机器人初始工具位置	43
6.4.9	设置机器人工具坐标	43
6.4.10	获取机器人工具坐标	44
6.4.11	松开所有关节	44
6.4.12	锁住所有关节	44
6.4.13	设置机器人各关节角度	45
6.4.14	获取机器人各关节角度	45
6.4.15	持续移动机器人关节	45
6.4.16	设置某输出IO端口数字输出值	46
6.4.17	获取某输出IO端口数字输出值	46
6.4.18	获取某输入IO端口输入模拟量	46
6.4.19	设置工具末端位置	47
6.4.20	获取工具末端位置	47
6.4.21	计算机器人曲线运行时的PVT列表	47
6.4.22	计算机器人直线运行时的PVT列表	48
6.4.23	设置PVT点之间的时间间隔	48
6.4.24	获取PVT点之间的时间间隔	49
6.4.25	释放PVT列表	49
6.4.26	根据机器人工具末端坐标计算各轴角度	49
6.4.27	根据机器人各轴角度计算工具末端坐标	50
6.4.28	应用层向控制层传输文件	51
6.4.29	控制层向应用层传输文件	51

1.0 引言

1.1 编写目的

此用户手册是为了使用户了解优爱宝机器人 SDK 的安装、使用、维护方法以及用户在使用时应注意的地方。其阅读对象为软件开发人员。

上海优爱宝智能机器人科技股份有限公司提供使用 SDK 的演示程序及相关源代码，但不包含在本 SDK 的安装包内。有需要的用户可以联系上海优爱宝智能机器人科技股份有限公司索取。

1.2 参考资料

以下资料可从优爱宝官网下载，或者联系上海优爱宝智能机器人科技股份有限公司索取：

- UIM SimpleCAN3.0B 说明书
- UIM2503 使用手册
- UIM62HXX 使用手册
- SCARA 机器人使用手册

2.0 运行环境

2.1 硬件环境

硬盘：100M 以上剩余空间；

内存：1GB 以上（推荐使用 2GB）；

CPU：1.4GHz 以上。

2.2 软件环境

Windows XP/Windows 7/Windows 8 32/64 位，Linux 系统。

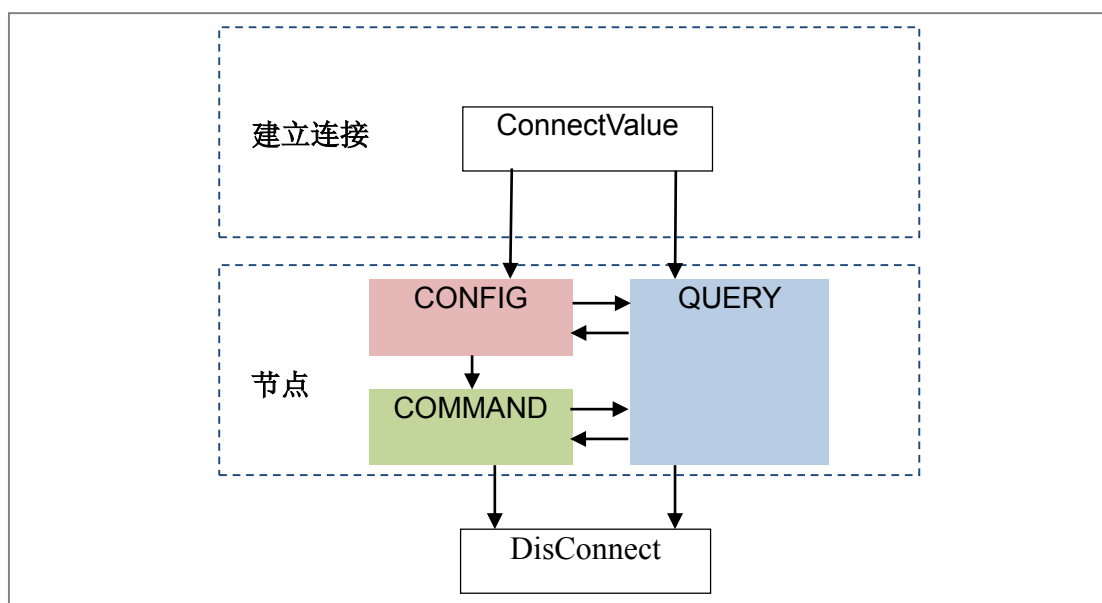
3.0 编程模型

建议在使用 SDK 的过程中，先进行连接，再发送命令，在连接的过程中不要发送命令，如设置密码 SetPassword、设置时间 SetDateTime、获取机器人状态 GetRobotStatus 等等。

SDK 内部对每个函数采取了 CriticalSection 进行多线程保护，如果在调用某个 SDK 函数的过程中，调用线程被结束，会导致 SDK 内部的 CriticalSection 没有释放，从而使得其他 SDK 函数的调用出现异常。在结束通讯前，应该先调用断开连接函数 Disconnect。

函数调用流程示意图

函数调用流程示意图如下：



此外，在上图所示的节点框内，CONFIG、QUERY 和 COMMAND 栏为示意用，分别包含了若干函数。

QUERY 包括的函数有：

- GetNetworkconfig
- GetDateTime
- GetRobotVersion
- GetRobotStatus
- GetRunningProgram
- GetUserProgramList
- GetDefaultUserProgram
- GetMotorTemperature
- GetRobotTCP
- GetInitializeTCP
- 等等

CONFIG 包括的函数有:

- SetPassword
- SetNetworkConfig
- SetDateTime
- SetInitializeTCP
- 等等

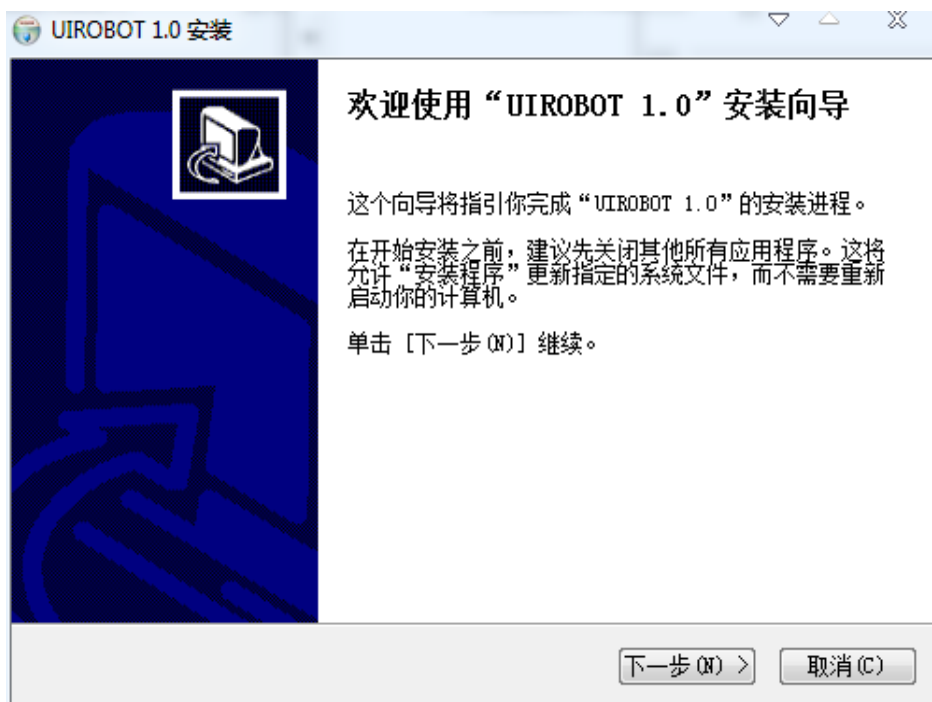
COMMAND 包括的函数有:

- MoveArmJoint
- EmergencyStop
- ResetRobot
- SetRobotTCP
- LooseAllJoint
- LockAllJoint
- 等等

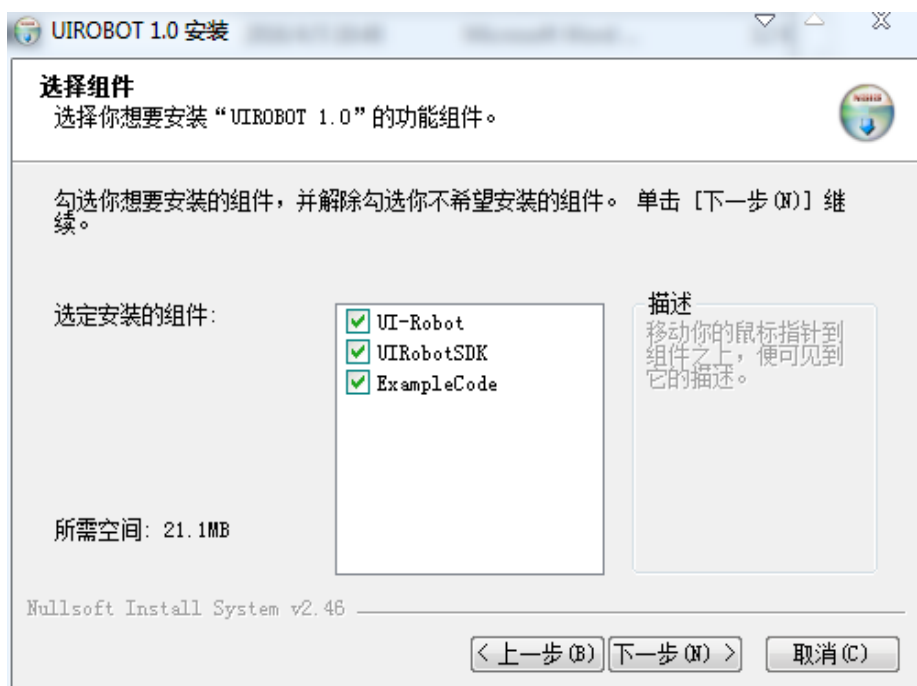
4.0 安装方法

4.1 机器人软件安装

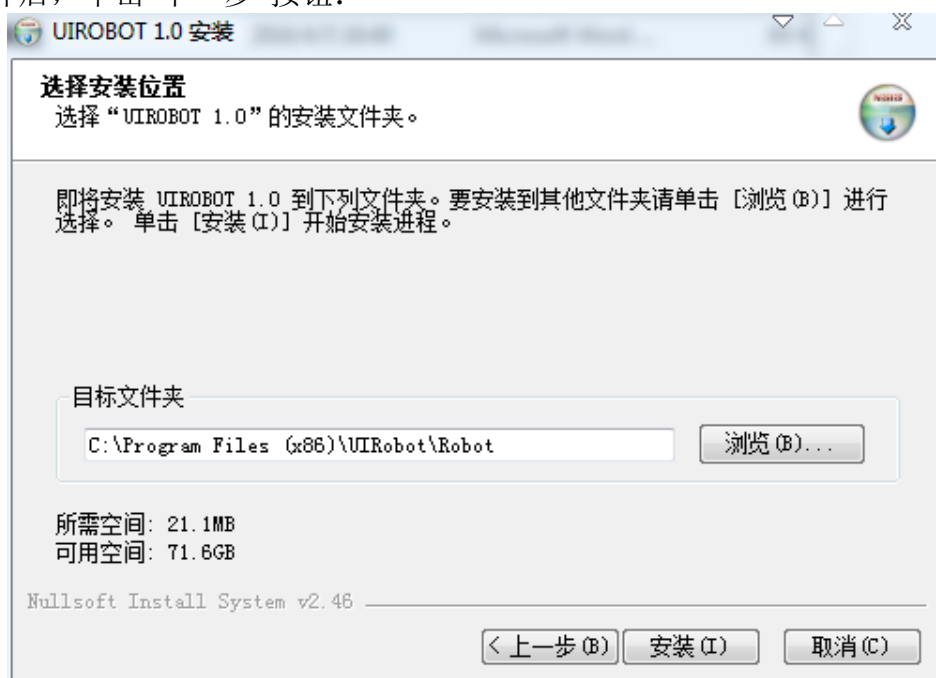
将安装文件“UIROBOTSetup.exe”拷贝到电脑硬盘上，双击安装文件进行安装。出现“欢迎使用‘UIROBOT 1.0’安装向导”界面。



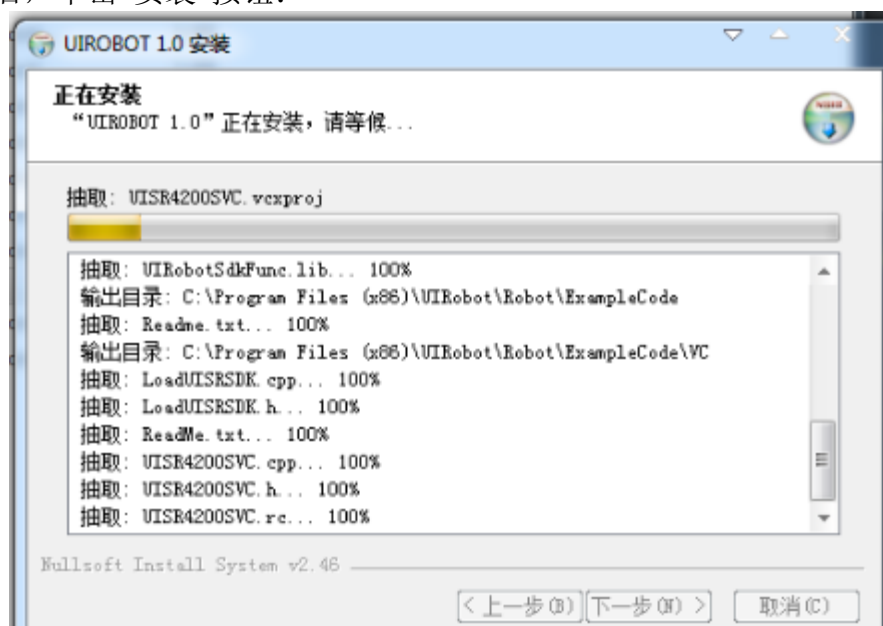
单击“下一步”按钮：



选择安装组件后，单击“下一步”按钮：

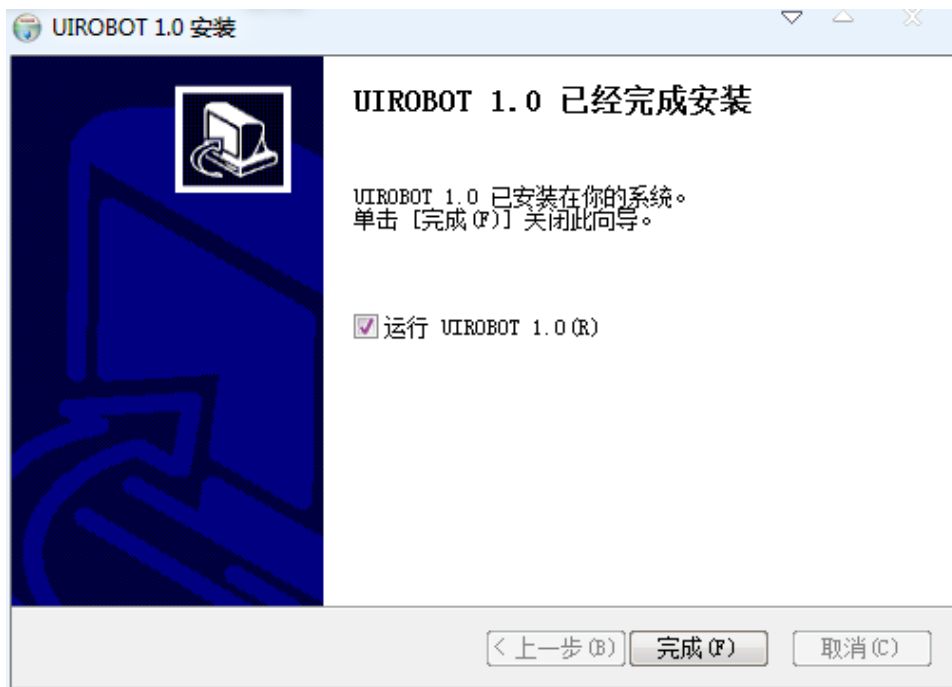


选择安装目录后，单击“安装”按钮：



软件开始安装，等待安装完成。

软件安装完成后，会出现“UIROBOT 1.0 已经完成安装”界面，点击“完成”。



按默认安装目录完成安装后，安装目录下会出现一个 Robot 文件夹：

DLL 文件夹内包括 SDK 调用动态库；

Include 文件夹内包括调用 SDK 动态库头文件；

LIB 文件夹内包括 SDK 静态调用所需文件；

ExampleCode 文件夹内包括调用 SDK 动态库范例源程序代码；

UIRobotWebsite UIROBOT 公司链接，如需帮助欢迎咨询；

UIROBOT.exe 机器人软件启动程序；

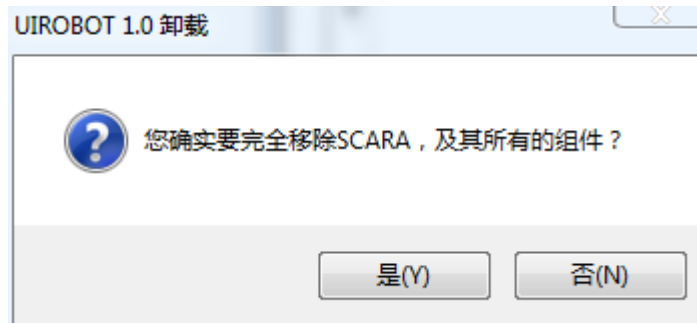
Uninst.exe 机器人软件卸载程序；

libUIScara.dll, libDelta.dll, UIRobotSdkFunc.dll 运行程序所需文件；

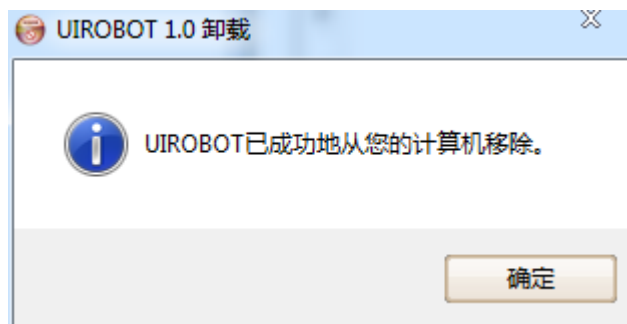


4.2 机器人软件卸载

双击安装目录下“uninst.exe”文件，进行卸载，如下图所示：



单击“是”按钮，将清空安装目录下所有文件，卸载完成。



优爱宝机器人 SDK

4.3 动态调用

安装好 SDK 后，用户可以选择将所需的 DLL 文件和头文件拷贝到工作目录下使用，也可以通过路径设置找到安装目录进行使用；

4.3.1 VC

1. 在 CPP 文件中包含"UIRobotSDK.h"文件，如：

```
#include "UIRobotSDK.h"
```

2. 加载 UIRobotSDK 的动态链接库，如：

```
m_hInstDevFunc = LoadLibrary(_T("UIRobotSdkFunc.dll"));
```

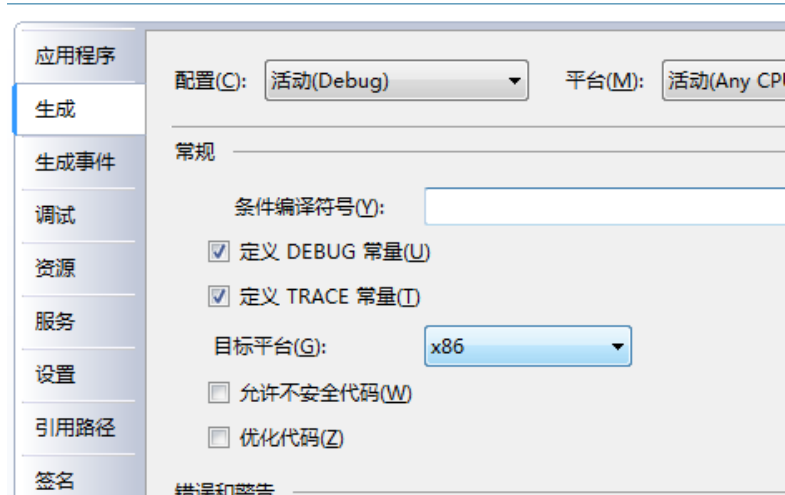
3. 加载库函数，如：

```
ConnectValue = (CONNECTVALUE)GetProcAddress(m_hInstDevFunc, "ConnectValue ");
```

4.3.2 VB

```
Public Declare Function ConnectValue Lib "UIRobotSdkFunc.dll" Alias "ConnectValue "  
(ByVal dwConnectType As Integer, ByVal iIPValue As Integer) As Integer
```

需要注意的地方，VB 目标平台选 x86,才可正常运行：

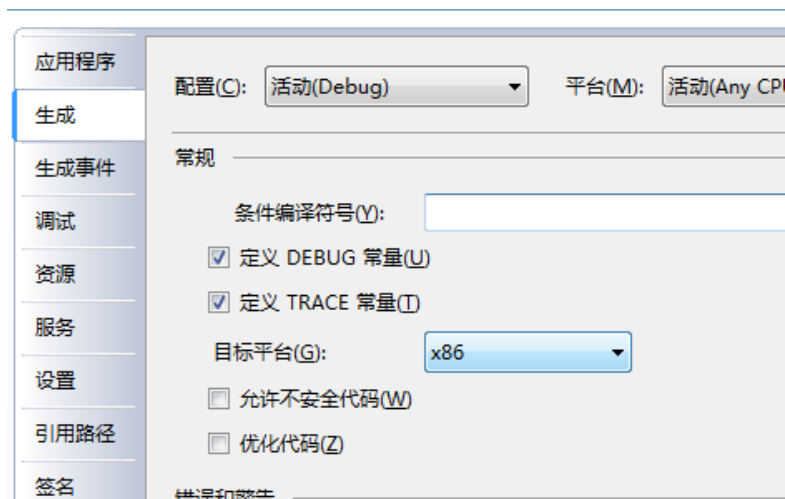


4.3.3 C#

函数导入，如：

```
[DllImport(" UIRobotSdkFunc.dll ",  
    EntryPoint = " ConnectValue ",  
    CharSet = CharSet.Ansi)]  
private static extern int ConnectValue (int dwConnectType, int iIPValue);
```

需要注意的地方，c#目标平台选 x86,才可正常运行：



4.4 静态调用

静态调用需要使用 LIB 目录下的两个文件：UIRobotSdkFunc.h 和 UIRobotSdkFunc.lib。

4.4.1 VC

1. 在扩展名为.CPP 的文件中包含 UIRobotSdkFunc.h 头文件。如：

```
#include "UIRobotSdkFunc.h";
```

2. 在工程的连接器设置中连接到 UIRobotSdkFunc.lib 文件。如：

在 VC7 环境下，在项目属性页里的配置属性 → 链接器 → 输入 → 附加依赖项中添加 UIRobotSdkFunc.lib。

5.0 数据结构

本章介绍了优爱宝机器人中涉及的各数据结构体的定义。

5.1 UI_SETPASSWORD_OBJ

功能：密码设置

VC 结构体定义：

```
typedef struct _UI_SETPASSWORD_OBJ
{
    char        OldPassword[UISDK_MAXPASSWORDLEN];
    int         iOldPasswordLen;
    char        NewPassword[UISDK_MAXPASSWORDLEN];
    int         iNewPasswordLen;
} UI_SETPASSWORD_OBJ, *P_UI_SETPASSWORD_OBJ;
```

OldPassword[UISDK_MAXPASSWORDLEN] 机器人当前密码

iOldPassworLen 机器人当前密码长度

NewPassword[UISDK_MAXPASSWORDLEN] 机器人新密码

iNewPasswordLen 机器人新密码长度

5.2 UI_NETWORKCONFIG_OBJ

功能：网络信息

VC 结构体定义：

```
typedef struct _UI_NETWORKCONFIG_OBJ
{
    char        IPAddress[UISDK_MAXPASSWORDLEN];
    int         iIPAddressLen;
    char        MaskAddress[UISDK_MAXPASSWORDLEN];
    int         iMaskAddressLen;
    char        GatewayAddress[UISDK_MAXPASSWORDLEN];
    int         iGatewayAddressLen;
} UI_NETWORKCONFIG_OBJ, *P_UI_NETWORKCONFIG_OBJ;
```

IPAddress[UISDK_MAXPASSWORDLEN] 机器人 IP 地址

iIPAddressLen 机器人 IP 地址长度

MaskAddress[UISDK_MAXPASSWORDLEN] 机器人 IP 地址子网掩码

iMaskAddressLen 机器人 IP 地址子网掩码长度

GatewayAddress[UISDK_MAXPASSWORDLEN] 机器人 IP 地址网关地址

iGatewayAddressLen 机器人 IP 地址网关长度

5.3 UI_DATETIME_OBJ

功能：日期时间

VC 结构体定义：

```
typedef struct _UI_DATETIME_OBJ
{
    int     dt_sec;
    int     dt_min;
    int     dt_hour;
    int     dt_mday;
    int     dt_mon;
    int     dt_year;
} UI_DATETIME_OBJ, *P_UI_DATETIME_OBJ;
```

dt_sec 秒

注释：范围为 0 ~ 59

dt_min 分钟

注释：范围为 0 ~ 59

dt_hour 小时

注释：范围为 0 ~ 23

dt_mday 日

注释：范围为 1 ~ 31

dt_mon 月

注释：范围为 0 ~ 11

dt_year 年

5.4 UI_RUNNINGPROGRAM_OBJ

功能：机器人运行程序信息

VC 结构体定义：

```
typedef struct _UI_RUNNINGPROGRAM_OBJ
{
    int    iResult;
    int    dt_mday;
    int    dt_hour;
    int    dt_min;
    int    dt_sec;
    char   name[UISDK_MAX_DATA_LEN];
} UI_RUNNINGPROGRAM_OBJ, *P_UI_RUNNINGPROGRAM_OBJ;
```

iResult 程序运行结果

注释：0：错误；1：成功。

dt_mday 日

注释：范围为 1～31

dt_hour 小时

注释：范围为 0～23

dt_min 分钟

注释：范围为 0～59

dt_sec 秒

注释：范围为 0～59

name[UISDK_MAX_DATA_LEN] 机器人程序名称

5.5 UI_COMMANDBUFFER_OBJ

功能：机器人数据通讯缓冲器数据结构

VC 结构体定义：

```
typedef struct _UI_COMMANDBUFFER_OBJ
{
    int      iCount;
    char     chDataBuffer [1024];
} UI_COMMANDBUFFER_OBJ, *P_UI_COMMANDBUFFER_OBJ;
```

iCount 用户数据长度，取范围为 0 ~ 1023

chDataBuffer[1024] 用户数据内容

5.6 UI_MOTORTEMPERATURE_OBJ

功能：机器人机械臂温度

VC 结构体定义：

```
typedef struct _UI_MOTORTEMPERATURE_OBJ
{
    int    iBaseTemperature;
    int    iMainArmTemperature;
    int    iAssistantArmTemperature;
    int    iBarArmTemperature;
} UI_MOTORTEMPERATURE_OBJ, *P_UI_MOTORTEMPERATURE_OBJ;
```

iBaseTemperature	基座温度 注释：单位 0.1 度
iMainArmTemperature	主臂温度 注释：单位 0.1 度
iAssistantArmTemperature	副臂温度 注释：单位 0.1 度
iBarArmTemperature	丝杆温度 注释：单位 0.1 度

5.7 UI_ROBOTTCP_OBJ

功能：机器人工具中心点坐标

VC 结构体定义：

```
typedef struct _UI_ROBOTTCP_OBJ
{
    double      fTCPx;
    double      fTCPy;
    double      fTCPz;
} UI_ROBOTTCP_OBJ, *P_UI_ROBOTTCP_OBJ;
```

fTCPx 工具中心点相对 X 坐标值

fTCPy 工具中心点 Y 坐标值

fTCPz 工具中心点 Z 坐标值

5.8 UI_INITIALIZETCP_OBJ

功能：机器人初始工具位置坐标

VC 结构体定义：

```
typedef struct _UI_INITIALIZETCP_OBJ
{
    double      fTCPx;
    double      fTCPy;
    double      fTCPz;
    double      fThetaX;
    double      fThetaY;
    double      fThetaZ;
} UI_INITIALIZETCP_OBJ, *P_UI_INITIALIZETCP_OBJ;
```

fTCPx 工具中心点相对 X 坐标值

fTCPy 工具中心点 Y 坐标值

fTCPz 工具中心点 Z 坐标值

fThetaX 工具中心点相对 X 方向角度值

fThetaY 工具中心点 Y 方向角度值

fThetaZ 工具中心点 Z 方向角度值

5.9 UI_JOINTANAGLE_OBJ

功能：机器人各个关节角度

VC 结构体定义：

```
typedef struct _UI_JOINTANAGLE_OBJ
{
    double      fBaseAngle;
    double      fMainArmAngle;
    double      fAssistantArmAngle;
    double      fBarArmAngle;
} UI_JOINTANAGLE_OBJ, *P_UI_JOINTANAGLE_OBJ;#include "..\
UI620SDK_RT.h.h";#include "UIRobotSDK.h"
```

fBaseAngle 基座角度

fMainArmAngle 主臂角度

fAssistantArmAngle 副臂角度

fBarArmAngle 丝杆角度

5.10 DIGITAL_VALUE_OBJ

功能：机器人各个数字输出端口

VC 结构体定义：

```
typedef union _DIGITAL_VALUE_OBJ
{
    struct
    {
        unsigned    bP1DVA :1;
        unsigned    bP2DVA :1;
        unsigned    bP3DVA :1;
        unsigned    bP4DVA :1;
        unsigned    bP5DVA :1;
        unsigned    bP6DVA :1;
        unsigned    bP7DVA :1;
        unsigned    bP8DVA :1;
        unsigned    bP9DVA :1;
        unsigned    bP10DVA :1;
        unsigned    bP11DVA :1;
        unsigned    bP12DVA :1;
        unsigned    bP13DVA :1;
        unsigned    bP14DVA :1;
        unsigned    bP15DVA :1;
        unsigned    bP16DVA :1;
    }
    unsigned int    uiDigitalValue;
} DIGITAL_VALUE_OBJ, *P_DIGITAL_VALUE_OBJ;
```

- bP1DVA :1;** 端口 1 的电平值
 注释： 0: 低电平 0V
 1: 高电平 3.3V
- bP2DVA :1;** 端口 2 的电平值
 注释： 0: 低电平 0V
 1: 高电平 3.3V
- bP3DVA :1;** 端口 3 的电平值
 注释： 0: 低电平 0V
 1: 高电平 3.3V
- bP4DVA :1;** 端口 4 的电平值
 注释： 0: 低电平 0V
 1: 高电平 3.3V
- bP5DVA :1;** 端口 5 的电平值
 注释： 0: 低电平 0V
 1: 高电平 3.3V
- bP6DVA :1;** 端口 6 的电平值

	注释: 0: 低电平 0V 1: 高电平 3.3V
bP7DVA :1;	端口 7 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP8DVA :1;	端口 8 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP9DVA :1;	端口 9 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP10DVA :1;	端口 10 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP11DVA :1;	端口 11 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP12DVA :1;	端口 12 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP13DVA :1;	端口 13 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP14DVA :1;	端口 14 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP15DVA :1;	端口 15 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V
bP16DVA :1;	端口 16 的电平值 注释: 0: 低电平 0V 1: 高电平 3.3V

5.11 UI_FILECONTENT_OBJ

功能：机器人文件传输数据块数据结构

VC 结构体定义：

```
typedef struct _UI_FILECONTENT_OBJ
{
    unsigned int    fileSize;
    unsigned int    position;
    unsigned int    length;
    char    data[UISDK_MAX_FILECONTENT_LEN];

} UI_ROBOTFILE_OBJ, *P_UI_ROBOTFILE_OBJ;
```

fileSize 文件大小，指程序文件总字节数

position 该段内容在文件中的位置，传输的文件内容块在程序文件中起始位置

length 缓冲器中的有效数据长度，传输的文件内容块数据长度

data[UISDK_MAX_FILECONTENT_LEN] 缓冲区，传输的文件内容块数据

5.12 UI_SOCKETMSG_OBJ

功能：机器人报文结构

VC 结构体定义：

```
typedef struct _UI_SOCKETMSG_OBJ
{
    DWORD      ID;
    BYTE       can_dlc;
    BYTE       Data[8];
} UI_SOCKETMSG_OBJ, *P_UI_SOCKETMSG_OBJ;
```

ID 报文的 ID

can_dlc 数据长度，长度不能超过 8

Data 数据

5.13 TCP

功能：机器人 TCP 坐标

VC 结构体定义：

```
typedef struct _TCP
{
    double      x;
    double      y;
    double      z;
    double      theta;
} TCP;
```

x TCP 坐标中 X 坐标值

y TCP 坐标中 Y 坐标值

z TCP 坐标中 Z 坐标值

theta 保留

5.14 NodePVT

功能：机器人 PVT 列表中坐标数据

VC 结构体定义：

```
typedef struct _NodePVT
{
    int          *P;
    int          *V;
    int          *T;
    int          *A;
} NodePVT;
```

***P** PVT 列表中的 QP

***V** PVT 列表中的 QV

***T** PVT 列表中的 QT

***A** 保留

5.15 RobotPVTLists

功能：机器人各关节的 PVT 列表结点

VC 结构体定义：

```
typedef struct _RobotPVTLists
{
    NodePVT      firstArmPVT;
    NodePVT      secondArmPVT;
    NodePVT      thirdArmPVT;
    NodePVT      fourArmPVT;
    int          pvtCount;
} RobotPVTLists;
```

firstArmPVT 第一关节的 PVT 列表结点

secondArmPVT 第二关节的 PVT 列表结点

thirdArmPVT 第三关节的 PVT 列表结点

fourArmPVT 第四关节的 PVT 列表结点

pvtCount PVT 列表中数据个数

6.0 函数接口

本章按照不同的功能，分类介绍优爱宝机器人 SDK 中的各函数接口。

6.1 机器人连接及关闭

连接机器人包括回调函数注册函数，连接机器人，关闭机器人等。

6.1.1. 回调函数注册函数

```
BOOL UIMRegRtcnCallBack(  
    PF_UIROBOTSDK_CALLBACK pCallBackFunc);
```

PF_UIROBOTSDK_CALLBACK pCallBackFunc 回调函数指针

返回值 非零值 = 操作失败
 0 = 操作成功

Note:

用于网关回调函数的注册

6.1.2. 根据连接值连接机器人

```
int ConnectValue(  
    unsigned int dwConnectType,  
    int iIPValue);
```

dwConnectType 连接类型
 1 为 TCP/IP 方式连接，机器人为服务器端，端口号为 8877
 2 为蓝牙方式连接

iIPValue 连接的参数值
 如果为 TCP/IP 方式连接，该值为服务端地址对应的 32 位数值
 如果为蓝牙方式连接，该值为串口号对应的序号

返回值 非零值 = 操作成功，连接通讯编号
 0 =操作失败

6.1.3. 根据连接字符串连接机器人

```
int ConnectName(  
    unsigned int dwConnectType,  
    char *pConnectParam);
```

dwConnectType 连接类型

1 为 TCP/IP 方式连接，机器人为服务器端，端口号为 8877

2 为蓝牙方式连接

***pConnectParam** 连接的参数字符串

若为 TCP/IP 方式连接，该值为服务端地址字符串，如字符串 “192.168.1.6”

若为蓝牙方式连接，该值为串口号字符，如字符串 “COM2”

返回值 非零值 = 操作成功，连接通讯编号

0 = 操作失败

6.1.4 断开机器人连接

```
int Disconnect(  
    unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 非零值 = 操作失败

0 = 操作成功

6.1.5 关闭机器人

```
int Shutdown(  
    unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 非零值 = 操作失败

0 = 操作成功

优爱宝机器人 SDK

6.2 机器人登陆及设置

机器人登陆及设置包括机器人登陆，机器人设置密码，机器人设置网络配置，机器人设置时间等。

6.2.1 登陆机器人

```
int Login(  
    unsigned int dwConnectCode,  
    char *pPassword,  
    int iPasswordLen);
```

dwConnectCode 连接通讯编号

***pPassword** 密码

iPasswordLen 密码长度

返回值 成功则返回 0
 失败则返回失败原因

6.2.2 设置机器人密码

```
int SetPassword(  
    unsigned int dwConnectCode,  
    P_UI_SETPASSWORD_OBJ setPasswordObj);
```

dwConnectCode 连接类型

P_UI_SETPASSWORD_OBJ 设置密码结构体

返回值 成功则返回 0
 失败则返回失败原因

6.2.3 设置机器人网络配置

```
int SetNetworkConfig(  
    unsigned int dwConnectCode,  
    P_UI_NETWORKCONFIG_OBJ setNetworkInObj);
```

dwConnectCode 连接通讯编号
P_UI_SETPASSWORD_OBJ 设置网络配置结构体

返回值 成功则返回 0
 失败则返回失败原因

6.2.4 设置机器人时间

```
int SetDateTime(  
                unsigned int dwConnectCode,  
                P_UI_DATETIME_OBJ setDateTimeObj);
```

dwConnectCode 连接通讯编号
P_UI_DATETIME_OBJ 设置时间结构体

返回值 成功则返回 0
 失败则返回失败原因

6.2.5 设置机器人默认用户程序

```
int SetDefaultUserProgram(  
                           unsigned int dwConnectCode,  
                           char *pProgramName);
```

dwConnectCode 连接通讯编号
***pProgramName** 机器人用户程序名称

返回值 成功则返回 0
 失败则返回失败原因

6.2.6 设置机器人类型

```
int SetRobotType(  
                 int iType);
```

iType	机器人类型
机器人类型定义	<pre>#define UIROBOTTYPE_400SCARA (1) #define UIROBOTTYPE_DELTA (2) #define UIROBOTTYPE_500SCARA (3) #define UIROBOTTYPE_600SCARA (4) #define UIROBOTTYPE_FLIPSCARA (5)</pre>
返回值	成功则返回 0 失败则返回失败原因
注意	脱机模式下使用

6.3 用户获取机器人信息

用户获取机器人信息包括获取机器人时间，获取机器人软件版本号，获取机器人工作状态，获取用户程序列表，获取机器人默认用户程序，获取机器人各个臂的温度等。

6.3.1 获取机器人网络配置

```
int GetNetworkConfig(  
    unsigned int dwConnectCode,  
    P_UI_NETWORKCONFIG_OBJ getNetworkObj);
```

dwConnectCode 连接通讯编号
P_UI_NETWORKCONFIG_OBJ 设置网络配置结构体

返回值 成功则返回 0
 失败则返回失败原因

6.3.2 获取机器人时间

```
int GetDateTime(  
    unsigned int dwConnectCode,  
    P_UI_DATETIME_OBJ getDateTimeObj);
```

dwConnectCode 连接通讯编号
P_UI_DATETIME_OBJ 获取时间结构体

返回值 成功则返回 0
 失败则返回失败原因

6.3.3 获取机器人软件版本号

```
int GetRobotVersion(  
    unsigned int dwConnectCode,  
    char *pVersion);
```

dwConnectCode 连接通讯编号

优爱宝机器人 SDK

***pVersion** 版本号字符串缓冲区

返回值 成功则返回 0
 失败则返回失败原因

6.3.4 获取机器人工作状态

```
int GetRobotStatus(  
                    unsigned int dwConnectCode,  
                    int *pStatus);
```

dwConnectCode 连接通讯编号

***pStatus** 机器人工作状态变量地址

返回值 成功则返回 0
 失败则返回失败原因

6.3.5 获取机器人当前程序信息

```
int GetRunningPogram(  
                      unsigned int dwConnectCode,  
                      P_UI_RUNNINGPROGRAM_OBJ runningProgramObj);
```

dwConnectCode 连接通讯编号

P_UI_RUNNINGPROGRAM_OBJ 机器人当前程序信息

返回值 成功则返回 0
 失败则返回失败原因

6.3.6 获取用户程序列表

```
int GetUserProgramList(  
                      unsigned int dwConnectCode,  
                      P_UI_PROGRAMLIST_OBJ programListObj);
```

dwConnectCode 连接通讯编号
P_UI_PROGRAMLIST_OBJ 机器人用户程序列表

返回值 成功则返回 0
 失败则返回失败原因

6.3.7 删除机器人指定用户程序

```
int DeleteUserProgram(  
                                        unsigned int dwConnectCode,  
                                        char *pProgramName);
```

dwConnectCode 连接通讯编号
***pProgramName** 机器人用户程序名称

返回值 成功则返回 0
 失败则返回失败原因

6.3.8 获取机器人默认用户程序

```
int GetDefaultUserProgram(  
                                        unsigned int dwConnectCode,  
                                        char *pProgramName);
```

dwConnectCode 连接通讯编号
***pProgramName** 机器人用户程序名称

返回值 成功则返回 0
 失败则返回失败原因

6.3.9 获取机器人各个臂的温度

```
int GetMotorTemperature(  
                                        unsigned int dwConnectCode,  
                                        P_UI_MOTORTEMPERATURE_OBJ motorTemperature);
```


6.4 机器人程序

运行机器人程序包括实际运行机器人用户程序，模拟运行机器人用户程序，机器人紧急停止，复位机器人，获取或设置机器人夹具中心点，获取或设置机器人初始工具位置，获取或设置机器人工具坐标，获取或设置机器人各关节角度，获取或设置某输出 IO 端口数字输出值，获取某输入 IO 端口数字输入值等。

6.4.1 运行机器人用户程序

```
int RunUserProgram(  
                    unsigned int dwConnectCode,  
                    char *pProgramName);
```

dwConnectCode 连接通讯编号
***pProgramName** 机器人用户程序名称

返回值 成功则返回 0
 失败则返回失败原因

6.4.2 模拟运行机器人用户程序

```
int SimulatorRunUserProgram(  
                             unsigned int dwConnectCode,  
                             char *pProgramName);
```

dwConnectCode 连接通讯编号
***pProgramName** 机器人用户程序名称

返回值 成功则返回 0
 失败则返回失败原因

6.4.3 机器人紧急停止

```
int EmergencyStop(  
                  unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 成功则返回 0
 失败则返回失败原因

6.4.4 复位机器人

```
int ResetRobot(  
                unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 成功则返回 0
 失败则返回失败原因

6.4.5 设置机器人夹具中心点

```
int SetRobotTCP(  
                unsigned int dwConnectCode,  
                P_UI_ROBOTTCP_OBJ robotTCP);
```

dwConnectCode 连接通讯编号
P_UI_ROBOTTCP_OBJ 机器人夹具中心点数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.6 获取机器人夹具中心点

```
int GetRobotTCP(  
                unsigned int dwConnectCode,  
                P_UI_ROBOTTCP_OBJ robotTCP);
```

dwConnectCode 连接通讯编号
P_UI_ROBOTTCP_OBJ 机器人夹具中心点数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.7 设置机器人初始工具位置

```
int SetInitializeTCP(  
                                 unsigned int dwConnectCode,  
                                 P_UI_INITIALIZETCP_OBJ initializeTCP);
```

dwConnectCode 连接通讯编号
P_UI_INITIALIZETCP_OBJ 机器人初始工具位置数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.8 获取机器人初始工具位置

```
int GetInitializeTCP(  
                                 unsigned int dwConnectCode,  
                                 P_UI_INITIALIZETCP_OBJ initializeTCP);
```

dwConnectCode 连接通讯编号
P_UI_INITIALIZETCP_OBJ 机器人初始工具位置数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.9 设置机器人工具坐标

```
int SetTargetTCP(  
                                 unsigned int dwConnectCode,  
                                 P_UI_INITIALIZETCP_OBJ initializeTCP);
```

dwConnectCode 连接通讯编号
P_UI_INITIALIZETCP_OBJ 机器人初始工具位置数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.10 获取机器人工具坐标

```
int GetTargetTCP(  
                  unsigned int dwConnectCode,  
                  P_UI_INITIALIZETCP_OBJ initializeTCP);
```

dwConnectCode 连接通讯编号
P_UI_INITIALIZETCP_OBJ 机器人初始工具位置数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.11 松开所有关节

```
int LooseAllJoint(  
                  unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 成功则返回 0
 失败则返回失败原因

6.4.12 锁住所有关节

```
int LockAllJoint(  
                  unsigned int dwConnectCode);
```

dwConnectCode 连接通讯编号

返回值 成功则返回 0
 失败则返回失败原因

6.4.13 设置机器人各关节角度

```
int SetJointAngle(  
    unsigned int dwConnectCode,  
    P_UI_JOINTANAGLE_OBJ jointAngle);
```

dwConnectCode 连接通讯编号
P_UI_JOINTANAGLE_OBJ 机器人各关节角度数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.14 获取机器人各关节角度

```
int GetJointAngle(  
    unsigned int dwConnectCode,  
    P_UI_JOINTANAGLE_OBJ jointAngle);
```

dwConnectCode 连接通讯编号
P_UI_JOINTANAGLE_OBJ 机器人各关节角度数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.15 持续移动机器人关节

```
int MoveArmJoint(  
    unsigned int dwConnectCode,  
    int iDirectoryIndex);
```

dwConnectCode 连接通讯编号
iDirectoryIndex 持续移动机器人方向索引值
 1: 向 Y 轴正方向持续移动;
 2: 向 Y 轴负方向持续移动;

3: 向 X 轴正方向持续移动;

4: 向 X 轴负方向持续移动;

返回值 成功则返回 0
 失败则返回失败原因

6.4.16 设置某输出IO端口数字输出值

```
int SetOutPortDigitalValue(  
                                 unsigned int dwConnectCode,  
                                 int iPortIndex,  
                                 bool bHighLevel);
```

dwConnectCode 连接通讯编号
iPortIndex 输出端口索引号:1~16
bHighLevel 输出电平高或低:TRUE 表示高电平; FALSE 表示低电平

返回值 成功则返回 0
 失败则返回失败原因

6.4.17 获取某输出IO端口数字输出值

```
int GetOutPortDigitalValue(  
                                 unsigned int dwConnectCode,  
                                 P_DIGITAL_VALUE_OBJ digitalValueObj);
```

dwConnectCode 连接通讯编号
P_DIGITAL_VALUE_OBJ 输入端口电平值

返回值 成功则返回 0
 失败则返回失败原因

6.4.18 获取某输入IO端口输入模拟量

```
int GetAnalogValue(  
                                 unsigned int dwConnectCode,  
                                 int iPortIndex);
```

```
unsigned int dwConnectCode,  
int iPortIndex,  
int * pAnalogValue);
```

dwConnectCode	连接通讯编号
iPortIndex	输出端口索引号:1~ROBOTINPUTPORTNUM
* pAnalogValue	某输入 IO 端口的输入模拟量

返回值 成功则返回 0
 失败则返回失败原因

6.4.19 设置工具末端位置

```
int SetToolTCP(  
                    TCP toolTcp);
```

toolTcp	工具末端相对 Z 轴中心点坐标
----------------	-----------------

返回值 成功则返回 0
 失败则返回失败原因

6.4.20 获取工具末端位置

```
int GetToolTCP(  
                    TCP toolTcp);
```

toolTcp	工具末端相对 Z 轴中心点坐标
----------------	-----------------

返回值 成功则返回 0
 失败则返回失败原因

6.4.21 计算机器人曲线运行时的PVT列表

```
int GetCurvePVTLists(  
                    int NodeCount,  
                    double position[][4],
```

```
int speed,  
RobotPVTLists *task);
```

NodeCount	曲线上坐标的个数
position	曲线上坐标点坐标，即 x,y,z,rz 值
speed	速度的百分比
task	PVT 列表（空间需要释放）

返回值	成功则返回 0
	失败则返回失败原因

6.4.22 计算机器人直线运行时的PVT列表

```
int GetLinePVTLists(  
    double positionA[],  
    double positionB[],  
    int speed,  
    RobotPVTLists *task);
```

positionA[]	直线起点坐标，即 x,y,z,rz 值
positionB[]	直线终点坐标，即 x,y,z,rz 值
speed	速度的百分比
task	PVT 列表（空间需要释放）

返回值	成功则返回 0
	失败则返回失败原因

6.4.23 设置PVT点之间的时间间隔

```
int SetPVTInterval(  
    int iTimeInterval);
```

TimeInterval	PVT 点之间的时间间隔（单位为毫秒，最小为 20 毫秒）
---------------------	-------------------------------

返回值 成功则返回 0
 失败则返回失败原因

6.4.24 获取PVT点之间的时间间隔

```
int GetPVTInterval();
```

返回值 成功则返回 0
 失败则返回失败原因

6.4.25 释放PVT列表

```
int ReleasePVTLists (  
                         RobotPVTLists *task);
```

task PVT 列表（空间需要释放）

返回值 成功则返回 0
 失败则返回失败原因

6.4.26 根据机器人工具末端坐标计算各轴角度

```
int GetAngleFromPosition(  
                         double x,  
                         double y,  
                         double z,  
                         double rz,  
                         double *angle_1,  
                         double *angle_2,  
                         double *angle_3,  
                         double *Toolz);
```

x 工具末端 x 坐标值（单位：mm）

y 工具末端 y 坐标值（单位：mm）

z 工具末端 z 坐标值（单位：mm）

rz 工具相对 X 轴正方向角度（单位：度）

*angle_1	第一机械臂角度值（单位：度）
*angle_2	第二机械臂角度值（单位：度）
*angle_3	第三机械臂角度值（单位：度）
*Toolz	第四机械臂长度值（单位：mm）
返回值	成功则返回 0 失败则返回失败原因

6.4.27根据机器人各轴角度计算工具末端坐标

```
int GetPositionFromAngle (  
    double theta_1,  
    double theta_2,  
    double theta_3,  
    double Toolz  
    double *x,  
    double *y,  
    double *z,  
    double *rz);
```

theta_1	第一机械臂角度值（单位：度）
theta_2	第二机械臂角度值（单位：度）
theta_3	第三机械臂角度值（单位：度）
Toolz	第四机械臂长度值（单位：mm）
*x	工具末端 x 坐标值（单位：mm）
*y	工具末端 y 坐标值（单位：mm）
*z	工具末端 z 坐标值（单位：mm）
*rz	工具相对 X 轴正方向角度（单位：度）
返回值	成功则返回 0 失败则返回失败原因

6.4.28应用层向控制层传输文件

```
int DownloadProgramFileContent(  
    unsigned int dwConnectCode,  
    P_UI_COMMANDBUFFER_OBJ robotFileObj);
```

dwConnectCode 连接通讯编号
P_UI_COMMANDBUFFER_OBJ 机器人程序文件数据结构

返回值 成功则返回 0
 失败则返回失败原因

6.4.29控制层向应用层传输文件

```
int UpProgramFile(  
    unsigned int dwConnectCode,  
    P_UI_COMMANDBUFFER_OBJ robotFileObj);
```

dwConnectCode 连接通讯编号
P_UI_COMMANDBUFFER_OBJ 机器人程序文件数据结构

返回值 成功则返回 0
 失败则返回失败原因