远程以太网控制功能用户手册

文档密级：公开

版本说明

本手册对应 API 版本为 3.14 系列版本，对应机器人控制器软件版本为 V4R24 系列版本。

- **历史版本：**

API 版本	控制器版本	更新说明
3.15.0.12	V4R24C2SPC0B21 及之后版本支持	1.修复错误
3.15.0.8	V4R24C2SPC0B3 及之后版本支持	1.新增更新内存中全局位置变量对应的位置参数接口： IMC100_Set_MemRobP 2.新增更新内存中全局关节位置变量对应的位置参数接口： IMC100_Set_MemRobJP
3.15.0.6	V4R24C2SPC0B2 及之后版本支持	1.新增干涉区相关接口 IMC100_Set_InterferZoneActStat IMC100_Get_InterferZoneActStat IMC100_Set_InterferZonePara IMC100_Get_InterferZonePara IMC100_Set_InterferToolActNum IMC100_Get_InterferToolActNum IMC100_Set_InterferToolPara IMC100_Get_InterferToolPara
3.15.0.5	V4R24C2SPC0B3 及之后版本支持	1. 新增线速度相关接口，轨迹恢复阈值可调接口 IMC100_Set_BindTcpSpeedValue IMC100_Get_BindTcpSpeedValue IMC100_Get_TcpSpeedDAOutAndVel IMC100_Close_BindTcpSpeed IMC100_Set_TraceRecoverMode IMC100_Get_TraceRecoverMode IMC100_Set_VarTraceRecoverParams IMC100_Get_VarTraceRecoverParams
3.14.1	V4R24C1SPC0B5 及之后版本支持	1.新增查询机器人本体轴数接口： IMC100_Get_RobotAxisNum
3.14.0.2	S03.24R 及之后版本支持	1.新增独立轴接口： IMC100_IndCMove/IMC100_IndSpeed/ IMC100_IndResetOld/IMC100_IndReset/ IMC100_Get_IndCMoveSts/IMC100_Get_IndResetSts/ IMC100_Get_MotionStsExceptIndAxis 2.新增多点位接口： IMC100_Get_RobPFromFile/IMC100_Set_RobPToFile/ IMC100_Get_CurRobPFileName/IMC100_Set_RobPHereToFile 3.新增限速开关接口 IMC100_Get_SpeedLimitSwitch/IMC100_Set_SpeedLimitSwitch

3.13.0.1	S03.23R 及之后版本支持	1.新增数据流模式下缓存运动数据数量查询接口: IMC100_Get_CurCmdCacheNum 2.新增读取点位标签接口: IMC100_Get_RobP_Label/IMC100_Get_RobJP_Label/ IMC100_Get_RobLP_Label 3.新增碰撞检测参数接口： IMC100_Set_CollModeAndAction/IMC100_Get_CollModeAndAction/ IMC100_Set_AxisCollMode/IMC100_Get_AxisCollMode/ IMC100_Set_AxisCollLevel/IMC100_Get_AxisCollLevel/ IMC100_Set_TeachModeAxisCollMode/IMC100_Get_TeachModeAxisCollMode/ IMC100_Set_TeachModeCollAction/IMC100_Get_TeachModeCollAction/ IMC100_Set_TeachModeAxisCollLevel/IMC100_Get_TeachModeAxisCollLevel/ IMC100_Set_PlayBackModeAxisCollMode/ IMC100_Get_PlayBackModeAxisCollMode/ IMC100_Set_PlayBackModeCollAction/IMC100_Get_PlayBackModeCollAction/ IMC100_Set_PlayBackModeAxisCollLevel/ IMC100_Get_PlayBackModeAxisCollLevel 2、新增 MovJAbs 运动到指定运动点接口： IMC100_MovJAbs_JP
3.12.4	S03.22R 及之后版本支持	1.新增新版点位、工具参数、工件参数相关接口 2.删除旧版点位、工具参数、工件参数接口 3.新增工作原点（适配外部轴）接口: IMC100_Set_HomeJPos/IMC100_Get_HomeJPos 4. 新增协调开关接口： IMC100_Set_TeachCoordinate/IMC100_Get_TeachCoordinate

1 概要说明

通过远程以太网控制功能，用户可以开发出专有的机器人系统应用软件，机器人控制器作为服务器（端口固定为：2222），用户作为客户端，用户可以通过以下 2 种方式交互：

方式 1：通过 API 函数接口和控制器交互，支持 VB、VC、C#、LabView 等程序语言；

方式 2：通过 API 字符串和机器人控制器交互，支持更多系统（例如：linux 系统等）和更多种语言（除现有 API 函数支持的 VB、VC、C#、LabView）。

2 界面配置

用户端和机器人控制器建立通讯后：

（1） 如果用户端仅需监控或查询机器人控制器信息，用户端通过 API 函数接口或字符串可直接查询信息；

（2） 如果用户端需要控制机器人，用户需要切换到“远程以太网”控制权限，并且通过 API 函数接口或字符串获取控制权限。

切换控制权方法：

- 通过 InoRobotLab 软件

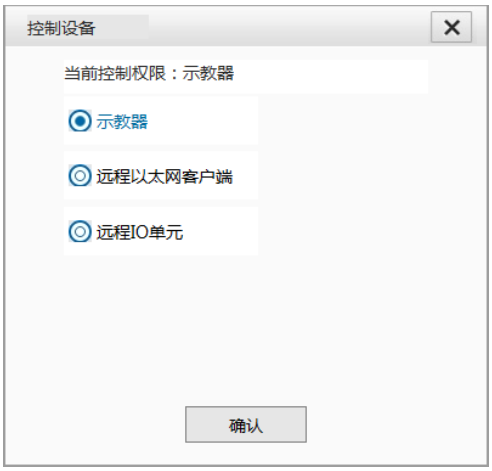
  步骤 1：登陆到管理员及以上权限

  步骤 2：切换到“远程以太网”控制权限

- 通过 InoTeachPad 软件

  步骤 1：登陆到管理员及以上权限

  步骤 2：切换到“远程以太网”控制权限


控制设备
当前控制权限：示教器
示教器
远程以太网客户端
远程IO单元
确认

3 API 函数调用说明

通过 API 函数接口和控制器交互，支持 VB、VC、C#、LabView 等程序语言。

3.1 开发语言选择

1. 基于 VB 开发
   - 以 Visual Basic 6.0 环境为例，新建工程。
   - 将产品提供的 IMC100API.bas、IMC100API.dll 文件拷贝至工程相应目录下。
   - 菜单中选择“工程/添加模块/现存”，找到对应工程目录下的 IMC100API.bas，添加到工程中。
   - 调用具体的函数接口，编写应用程序。
2. 基于 VC 开发
   - 以 Visual Studio 2010 环境为例，新建 C++工程。
   - 将产品提供的 IMC100API.lib、IMC100API.dll、IMC100API.h 文件拷贝至工程相应目录下。
   - 在程序文件中，增加 #include "IMC100API.h"语句。
   - 在程序文件中增加#pragma comment(lib,"IMC100API.lib")语句，或者在“工程属性/链接器/输入/附加依赖库”中添加 IMC100API.lib。
   - 调用具体的函数接口，编写应用程序。
3. 基于 VB.Net 开发
   - 以 Visual Studio 2010 环境为例，新建 Visual Basic .NET 工程。
   - 将产品提供的 IMC100API.vb 文件拷贝至工程目录下，IMC100API.dll 文件拷贝至工程目标输出路径下。
   - 菜单中选择“项目/添加现有项”，找到对应工程目录下的 IMC100API.vb，添加到工程中。
   - 调用具体的函数接口，编写应用程序。
4. 基于 C#开发
   - 以 Visual Studio 2010 环境为例，新建 Visual C# .NET 工程。
   - 将产品提供的 IMC100API.cs 文件拷贝至工程目录下，IMC100API.dll 文件拷贝至工程目标输出路径下。
   - 菜单中选择“项目/添加现有项”，找到对应工程目录下的 IMC100API.cs，添加到工程中。
   - 调用具体的函数接口，编写应用程序。

3.2 基本功能描述

API 函数接口主要分为三类，初始化应用类、监控类以及控制类。

1. 初始化应用类：该类函数是其它函数调用的基础，包括函数IMC100_Init_ETH()、IMC100_Exit_ETH()。
2. 监控类：该类函数在IMC100_Init_ETH()成功调用后均可正常调用，没有控制权、用户级别约束。包括IMC100_Get_RobPosHere()、IMC100_Get_DONum()、IMC100_Get_StrPara()、IMC100_Get_RobP ()等。

3. 控制类：该类函数不仅需要IMC100_Init_ETH()成功调用，还需要控制设备为“远程以太网客户端”，且需要调用IMC100_AcqPermit()获取控制许可，部分函数依据用户等级还需要调用函数IMC100_UserLogin()登陆。包括IMC100_MovJ_P()、IMC100_Set_DO()、IMC100_Get_RobP ()、IMC100_Set_HomeJPos()等。

注：

（1） 详细功能描述参考“附录 1：API 函数说明”；

（2） API 函数和 API 字符串功能一致，API 函数和 API 字符串一一对应。

3.3 API 函数调用说明

下面以 VS 平台中 C/C++应用开发为例，介绍基本功能的实现及调用范例。

调用 IMC100_Init_ETH()函数可以通过网络连接机器人，调用 IMC100_Exit_ETH()可以断开机器人。

在调用其他任何 API 之前，首先应调用一次 IMC100_Init_ETH()，以确保每次打开应用时机器人已经连接。如果该函数返回值非零，请检查机器人控制系统是否正常启动，并对照“附录 3：API 故障说明”排查故障。

调用 IMC100_Exit_ETH()应该在调用其他 API 之后，该 API 返回 0 后机器人断开连接。

调用 IMC100_Init_ETH()代码示例如下：

```
int ret = 0;      DWORD dwIP1 = 0xc0a81719;     // 对应 IP：192.168.23.25
    int ipPort = 2222;   int timeOut = 5;      // 通信超时时间 5s
    int robotNo = 0;     ret = IMC100_Init_ETH(dwIP1, ipPort, timeOut, robotNo);   if(ret < 0)
    {
        // 此处加入异常处理代码
        return;   }
```

调用 IMC100_Exit_ETH()代码示例如下：

```
    ret = IMC100_Exit_ETH(0)
    if(ret < 0)
    {
        // 此处加入异常处理代码
        return;   }
```

b) 监测机器人状态

IMC100 机器人提供数百个 API 函数接口用于监测机器人状态。该类函数不受控制权限、用户级别的约束。调用该类函数前，请确认机器人系统已经正常启动，并且连接机器人成功。该类函数包括 IMC100_Get_RobPosHere()、IMC100_Get_DINum()、IMC100_Get_StrPara()、IMC100_Get_RobP ()等。

调用 IMC100_Get_RobPosHere()代码示例如下：

```
    // 查询机器人工件坐标系下的位置值
int ret = 0;      int robotNo = 0;     int dinum = 0;        int dists = 0; ROB_POS posTemp;
memset(&posTemp, 0, sizeof(posTemp));        ret = IMC100_Get_RobPosHere(&posTemp,
robotNo);        if(ret < 0)
```

```
{
    // 此处加入异常处理代码
}
```

调用 IMC100_Get_DI()代码示例如下：

```
// 查询 DI0 状态，dists 为 1 表示 on
ret = IMC100_Get_DI(dinum, &dists, robotNo);
if(ret < 0)
{
    // 此处加入异常处理代码
}
```

c) 获取控制许可

调用 IMC100_AcqPermit()可以获取到机器人控制权限许可，调用 IMC100_CurPermit()可以查询当前获得许可的客户端。

由于一个机器人系统可以连接多个以太网客户端，所以当其中一个客户端需要控制机器人时，必须获得控制许可。调用前确认机器人系统已经正常启动，并且连接机器人系统成功。并且，示教器界面“系统设置-其他设置-其他-控制设备”选项下请选择“远程以太网客户端”。

调用 IMC100_CurPermit()及 IMC100_AcqPermit()代码示例如下：

```
int ret = 0;        int ower = 0;        DWORD IpAddr = 0;        int ipPort = 0;      int robotNo = 0;
ret = IMC100_CurPermit(&ower, &IpAddr, & ipPort, robotNo);   if(ret < 0)
    {
        // 此处加入异常处理代码
        return;   }
    if(ower != 1)   // 当前客户端设备未获得许可
    {
        ret = IMC100_AcqPermit(1, robotNo);     // 强制获取许可，ower 为 0 时可普通获取
        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }
```

d) 用户登录

调用 IMC100_CurUserType()可以查询当前设备用户等级，调用 IMC100_UserLogin()进行用户登录，调用 IMC100_UserLogout()可以退出当前用户登录状态。

不同的用户等级允许用户对机器人进行不同程度和范围的控制和操作。

调用前确认机器人系统已经正常启动，并且连接机器人系统成功。示教器界面“系统设置-其他设置-其他-控制设备”选项下已选择“远程以太网客户端”。

调用 IMC100_CurUserType()、IMC100_UserLogin()及 IMC100_UserLogout()代码示例如下：

```
int ret = 0;      int type = 0;    char password[8];   int robotNo = 0;        ret =
IMC100_CurUserType(&type, robotNo);      if(ret < 0)
```

```
{
    // 此处加入异常处理代码
}
    type = 2;       memcpy(password, "000000", sizeof(password)); ret =
IMC100_UserLogin(type, password, robotNo);    //登录管理模式，密码同示教器密码
if(ret < 0)
{
    // 此处加入异常处理代码
}
ret = IMC100_UserLogout(robotNo);    if(ret < 0)
{
    // 此处加入异常处理代码
}
```

e) 机器人回原点

调用 IMC100_DsMode()打开数据流模式，调用 IMC100_Home()控制机器人回原点。
调用前确认机器人系统已经正常启动，机器人系统成功连接，并已获得客户端控制许可。

调用 IMC100_DsMode()和 IMC100_Home()代码示例如下：

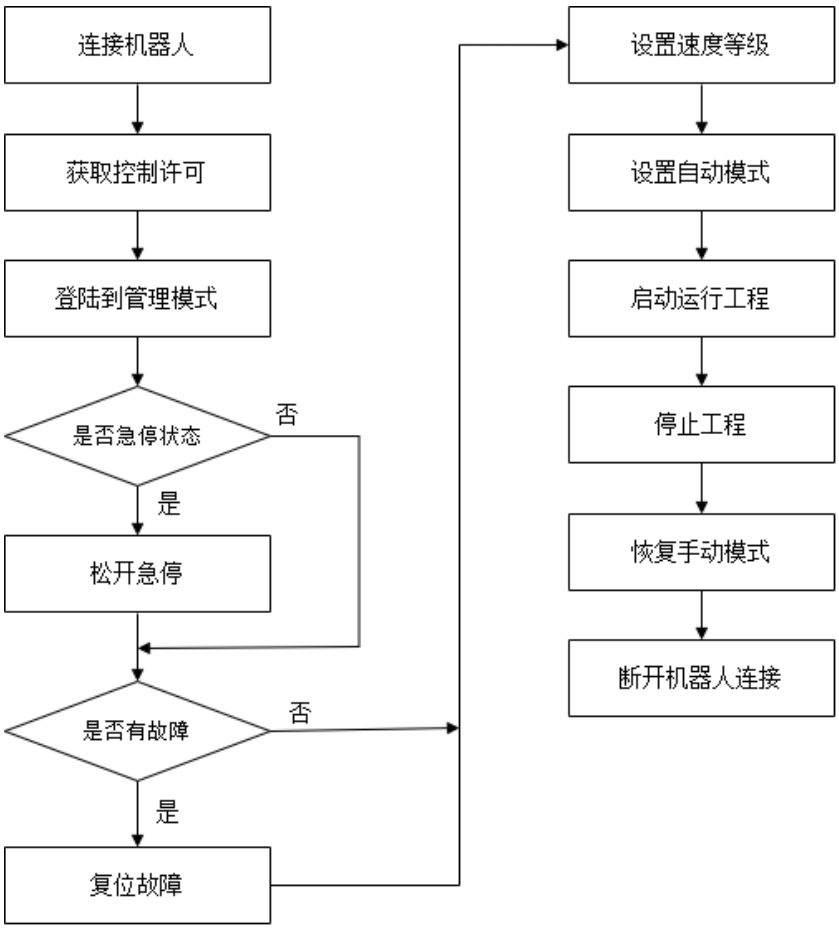
```
int ret = 0;    int sts = 0;    int robotNo = 0;    ret = IMC100_Get_DsMode(&sts, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(sts == 0)  // 数据流模式关闭
    {
        int cmd = 1;        ret = IMC100_DsMode(cmd, robotNo);    // 开启数据流模式
        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }
    int num = 0;   ret = IMC100_Home(num, robotNo);     // 机器人回 0 号原点
    if(ret < 0)
    {
        //此处加入异常处理代码
    }
```

4 API 函数典型应用案例

下面以完整的典型应用案例来进一步说明 API 函数的调用过程。

4.1 上位机打开机器人工程并自动运行

该操作前请确认目标机器人工程可以正常运行。
整体流程：



代码示例如下：

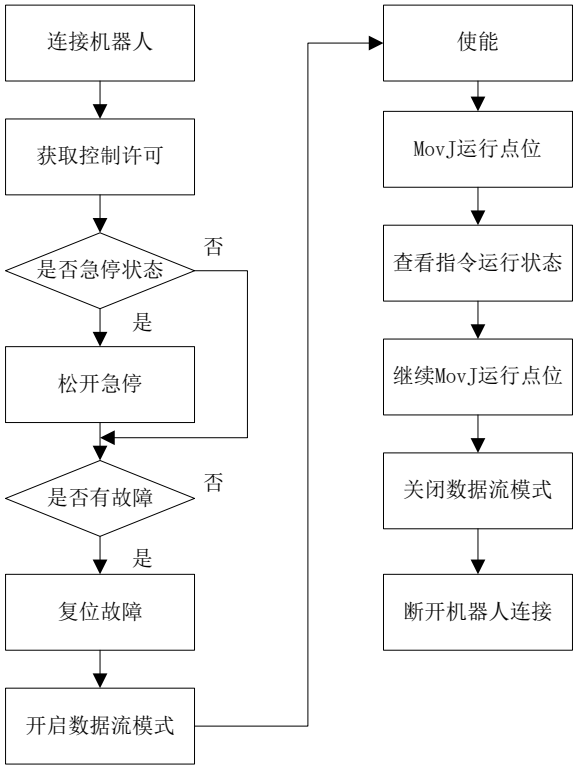
```
int ret = 0;      DWORD dwIP1 = 0xc0a81719;     // 对应 IP：192.168.23.25
      int ipPort = 2222;    int timeOut = 5;       // 通信超时时间 5s
      int robotNo = 0;     /* 连接机器人 */
      ret = IMC100_Init_ETH(dwIP1, ipPort, timeOut, robotNo);   if(ret < 0)
      {
            // 此处加入异常处理代码
            return;    }

      int ower = 0;          DWORD IpAddr = 0;          int ipPort = 0;          /* 获取控制许可 */
      ret = IMC100_CurPermit(&ower, &IpAddr, & ipPort, robotNo);    if(ret < 0)
      {
            // 此处加入异常处理代码
```

```
    }
    if(ower != 1)   // 当前客户端设备未获得许可
    {
        ret = IMC100_AcqPermit(1, robotNo);    // 强制获取许可，ower 为 0 时可普通获取
        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }

int type = 2;   char password[8];   memcpy(password, "000000", sizeof(password));      /* 登录管理模式 */
    ret = IMC100_UserLogin(type, password, robotNo);    // 登录管理模式，密码同示教器密码
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    int sts = 0;      /* 急停状态 */
    ret = IMC100_Get_EStopSts(&sts, robotNo);       if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(sts == 1)
    {
        int cmd = 0;          /* 急停松开 */
        ret = IMC100_EmergStop(cmd, robotNo);          if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }

    int err = 0;      /* 故障查询 */
    ret = IMC100_Get_SysErr(&err, robotNo);    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(err != 0)
    {
        /* 复位故障 */
        ret = IMC100_ResetErr(robotNo);        if(ret < 0)
        {
            // 此处加入异常处理代码
```

```
    }
}

int vel = 50;    /* 设置运行速度 */
ret = IMC100_Set_Vel(vel, robotNo);    if(ret < 0)
{
   // 此处加入异常处理代码
}

int mode = 2;    // 自动模式
/* 设置自动模式 */
ret = IMC100_Set_Mode(mode, robotNo);   if(ret < 0)
{
    // 此处加入异常处理代码
}

cmd = 1; /* 启动程序 */
ret = IMC100_PrgCtrl(cmd, robotNo);   if(ret < 0)
{
    // 此处加入异常处理代码
}

cmd = 0; /* 停止程序 */
ret = IMC100_PrgCtrl(cmd, robotNo);   if(ret < 0)
{
    // 此处加入异常处理代码
}

int mode = 1; /* 设置手动模式 */
ret = IMC100_Set_Mode(mode, robotNo);   if(ret < 0)
{
    // 此处加入异常处理代码
}

/* 断开机器人连接 */
ret = IMC100_Exit_ETH(0)
if(ret < 0)
{
    // 此处加入异常处理代码
}
```

4.2 上位机规划点位控制机器人运动

该操作前请确认上位机规划的点位机器人可安全到达。
整体流程：



代码示例如下：

```
int ret = 0;      DWORD dwIP1 = 0xc0a81719;     // 对应 IP：192.168.23.25
    int ipPort = 2222;    int timeOut = 5;       // 通信超时时间 5s
    int robotNo = 0;     /* 连接机器人 */
    ret = IMC100_Init_ETH(dwIP1, ipPort, timeOut, robotNo);   if(ret < 0)
    {
        // 此处加入异常处理代码
        return;    }

    int ower = 0;         DWORD IpAddr = 0;         int ipPort = 0;        /* 获取控制许可 */
    ret = IMC100_CurPermit(&ower, &IpAddr, & ipPort, robotNo);    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(ower != 1)   // 当前客户端设备未获得许可
    {
        ret = IMC100_AcqPermit(1, robotNo);     // 强制获取许可，ower 为 0 时可普通获取
        if(ret < 0)
        {
            // 此处加入异常处理代码
```

```
    }
}

int sts = 0;    /* 急停状态 */
ret = IMC100_Get_EStopSts(&sts, robotNo);    if(ret < 0)
{
    // 此处加入异常处理代码
}
if(sts == 1)
{
    int cmd = 0;        /* 急停松开 */
    ret = IMC100_EmergStop(cmd, robotNo);        if(ret < 0)
    {
        // 此处加入异常处理代码
    }
}

int err = 0;    /* 故障查询 */
ret = IMC100_Get_SysErr(&err, robotNo);    if(ret < 0)
{
    // 此处加入异常处理代码
}
if(err != 0)
{
    /* 复位故障 */
    ret = IMC100_ResetErr(robotNo);    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
}

cmd = 1; /* 使能 */
ret = IMC100_MotorEnable(cmd, robotNo);    if(ret < 0)
{
    // 此处加入异常处理代码
}

/* 开启数据流模式前保证机器人已经使能 */
int motorEnableSts = 0; while (0 == motorEnableSts)
{
    ret = IMC100_Get_MotorSts(&motorEnableSts, robotNo);    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
```

```
        Sleep(10); }

    cmd = 1;  /* 开启数据流模式 */
    ret = IMC100_DsMode(cmd, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    ROB_POS pos;        memset(&pos, 0, sizeof(pos));     pos. RPosData[0] = 110;        pos.
RPosData[1] = 35; pos. RPosData[2] = 956; pos. RPosData[3] = -108; pos. RPosData[4] = -5; pos.
RPosData[5] = -20; VER_DATA movVel; memset(&movVel, 0, sizeof(movVel)); movVel. velPercent
= 100 ; MOV_IO MovIOInfo ; memset(&MovIOInfo, 0, sizeof(MovIOInfo)) ; MovIOInfo. IONo = 1 ;
MovIOInfo. IOVa = 1 ; MovIOInfo. Value =0.005 ;  /* MovJ 运动 */
    ret = IMC100_MovJ_RobPos(&pos, 100, 0, 1, &MovIOInfo, robotNo)
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    /* 运动完成状态检查 */
    ret = IMC100_Get_CurCmdSts(&sts, robotNo);    if(ret < 0)    //此处加入异常处理代码
    if(sts == 0)
    {
        // 运动未完成，自定义操作，可循坏查询
    }

    memset(&pos, 0, sizeof(pos));      pos. RPosData[0] = 324;        pos. RPosData[1] = -210;
pos. RPosData[2] = 297; pos. RPosData[3] = 25; pos. RPosData[4] = 10; pos. RPosData[5] = 169;
memset(&movVel , 0, sizeof(movVel)); movVel. velPercent = 100 ; memset(&MovIOInfo, 0,
sizeof(MovIOInfo)) ; MovIOInfo. IONo = 1 ; MovIOInfo. IOVa = 0 ; MovIOInfo. Value =0.005 ;   /*
MovJ 运动 */
    ret1 = IMC100_MovJ_RobPos(&pos, 100, 0, 1, &MovIOInfo, robotNo)
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    cmd = 0;  /* 关闭数据流模式 */
    ret = IMC100_DsMode(cmd, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
```

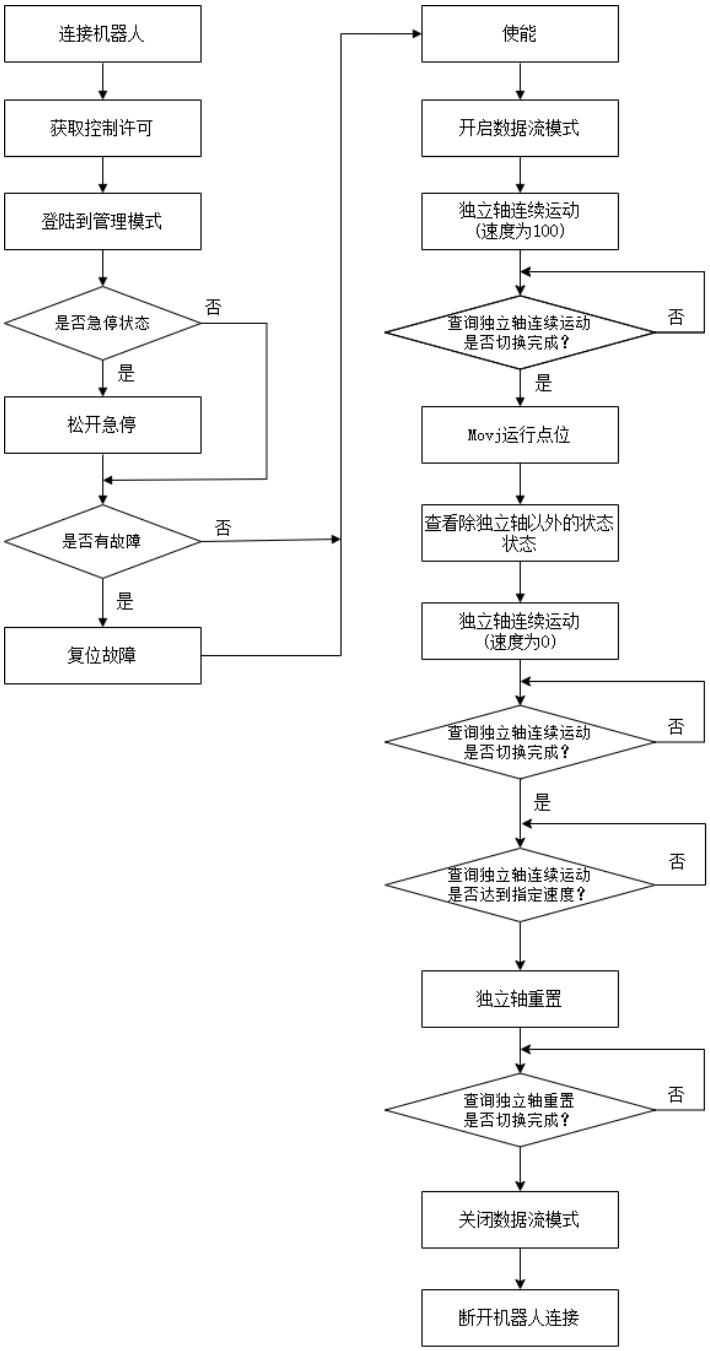
```
/* 断开机器人连接 */
ret = IMC100_Exit_ETH(0)
if(ret < 0)
{
    // 此处加入异常处理代码
}
```

4.3 上位机控制独立轴运动

该操作前请先设置轴支持独立轴，设置路径：

- InoRobotLab 软件：【控制器参数配置】-【外设配置】-【机械单元配置】
- InoTeachPad 软件：【设置】-【外设配置】-【机械单元配置】

整体流程：


连接机器人
获取控制许可
登陆到管理模式
是否急停状态
否
是
松开急停
是否有故障
否
是
复位故障
使能
开启数据流模式
独立轴连续运动
(速度为100)
查询独立轴连续运动
是否切换完成？
否
是
Movj运行点位
查看除独立轴以外的状态
状态
独立轴连续运动
(速度为0)
查询独立轴连续运动
是否切换完成？
否
是
查询独立轴连续运动
是否达到指定速度？
否
独立轴重置
查询独立轴重置
是否切换完成？
否
关闭数据流模式
断开机器人连接

代码示例如下：

```
int ret = 0;      DWORD dwIP1 = 0xc0a81719;     // 对应 IP：192.168.23.25
    int ipPort = 2222;   int timeOut = 5;      // 通信超时时间 5s
    int robotNo = 0;     /* 连接机器人 */
    ret = IMC100_Init_ETH(dwIP1, ipPort, timeOut, robotNo);   if(ret < 0)
    {
        // 此处加入异常处理代码
        return;    }

    int ower = 0;         DWORD IpAddr = 0;         int ipPort = 0;        /* 获取控制许可 */
    ret = IMC100_CurPermit(&ower, &IpAddr, & ipPort, robotNo);    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(ower != 1)   // 当前客户端设备未获得许可
    {
        ret = IMC100_AcqPermit(1, robotNo);     // 强制获取许可，ower 为 0 时可普通获取
        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }

    int sts = 0;      /* 急停状态 */
    ret = IMC100_Get_EStopSts(&sts, robotNo);       if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(sts == 1)
    {
        int cmd = 0;          /* 急停松开 */
        ret = IMC100_EmergStop(cmd, robotNo);          if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }

    int err = 0;      /* 故障查询 */
    ret = IMC100_Get_SysErr(&err, robotNo);     if(ret < 0)
    {
        // 此处加入异常处理代码
    }
    if(err != 0)
    {
```

```
        /* 复位故障 */
        ret = IMC100_ResetErr(robotNo);        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
    }

    cmd = 1; /* 使能 */
    ret = IMC100_MotorEnable(cmd, robotNo);        if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    /* 开启数据流模式前保证机器人已经使能 */
    int motorEnableSts = 0;  while (0 == motorEnableSts)
    {
        ret = IMC100_Get_MotorSts(&motorEnableSts, robotNo);        if(ret < 0)
        {
            // 此处加入异常处理代码
        }
        Sleep(10); }

    cmd = 1; /* 开启数据流模式 */
    ret = IMC100_DsMode(cmd, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    char mecUnit[18] = "Robot";        int indAxis = 6; int indAxisVel = 100; int indAxisAcc = 50;
int indAxisDec = 50;      /* 独立轴连续运动（速度为 100） */
    ret = IMC100_IndCMove(mecUnit, indAxis, indAxisVel, indAxisAcc, indAxisDec, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    int indCMoveSts = 0;     /* 查询独立轴连续运动是否切换完成 */
    while (0 == indCMoveSts)
    {
        ret = IMC100_Get_IndCMoveSts (mecUnit, indAxis, &indCMoveSts, robotNo);
    if(ret < 0)
        {
            // 此处加入异常处理代码
```

```
        }
        Sleep(10); }

    /* 如果需要等待独立轴速度达到设定值，此处增加 IMC100_IndSpeed 查询 */

    ROB_POS pos;        memset(&pos, 0, sizeof(pos));     pos. RPosData[0] = 110;        pos.
RPosData[1] = 35; pos. RPosData[2] = 956; pos. RPosData[3] = -108; pos. RPosData[4] = -5; pos.
RPosData[5] = -20; VER_DATA movVel; memset(&movVel, 0, sizeof(movVel)); movVel. velPercent
= 100 ; MOV_IO MovIOInfo ; memset(&MovIOInfo, 0, sizeof(MovIOInfo)) ; MovIOInfo. IONo = 1 ;
MovIOInfo. IOVa = 1 ; MovIOInfo. Value =0.005 ;  /* MovJ 运动 */
    ret = IMC100_MovJ_RobPos(&pos, 100, 0, 1, &MovIOInfo, robotNo)
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

    /* 运动状态检查：等待运动启动 */
    /* 注意：1、检查除独立轴以外的运行状态；2、运动数据下发到启动，需要一定时间
启动（建议增加超时处理，避免重复点或轨迹非常短的线段未追踪到运动状态） */
    sts = 0;    while (0 == sts)
    {
        ret = IMC100_Get_MotionStsExceptIndAxis (&sts, robotNo);          if(ret < 0)
        {
            // 此处加入异常处理代码
        }
        Sleep(10); }

    /* 运动状态检查：等待运动停止 */
    sts = 1;    while (1 == sts)
    {
        ret = IMC100_Get_MotionStsExceptIndAxis (&sts, robotNo);          if(ret < 0)
        {
            // 此处加入异常处理代码
        }
        Sleep(10); }

indAxisVel = 0;      /* 独立轴连续运动（速度为 0） */
    ret = IMC100_IndCMove(mecUnit, indAxis, indAxisVel, indAxisAcc, indAxisDec, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

indCMoveSts = 0;  /* 查询独立轴连续运动是否切换完成 */
```

```
    while (0 == indCMoveSts)
    {
        ret = IMC100_Get_IndCMoveSts (mecUnit, indAxis, &indCMoveSts, robotNo);
    if(ret < 0)
        {
            // 此处加入异常处理代码
        }
        Sleep(10); }

indCMoveSts = 0;   /* 查询独立轴是否达到设定速度 */
    while (0 == indCMoveSts)
    {
        ret = IMC100_IndSpeed (mecUnit, indAxis, &indCMoveSts, robotNo);            if(ret <
0)
        {
            // 此处加入异常处理代码
        }
        Sleep(10); }

    /* 独立轴重置 */
    ret = IMC100_IndResetOld(mecUnit, indAxis, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }

int indResetSts = 0;       /* 查询独立轴重置是否完成 */
    while (0 == indResetSts)
    {
        ret = IMC100_Get_IndResetSts (mecUnit, indAxis, &indResetSts, robotNo);
    if(ret < 0)
        {
            // 此处加入异常处理代码
        }
        Sleep(100);        //等待 100ms
    }

    cmd = 0;  /* 关闭数据流模式 */
    ret = IMC100_DsMode(cmd, robotNo);
    if(ret < 0)
    {
        // 此处加入异常处理代码
    }
```

```
/* 断开机器人连接 */
ret = IMC100_Exit_ETH(0)
if(ret < 0)
{
    // 此处加入异常处理代码
}
```

5 API 字符串调用说明

通过字符串和机器人控制器交互，支持更多系统（例如：linux 系统等）和更多种语言（除现有 API 函数支持的 VB、VC、C#、LabView）。

使用步骤：
（1）使用外部平台接口通讯接口和机器人控制器建立通讯（机器人控制器为服务器，IP 为控制器 IP，端口号为 2222）；
（2）使用对应的功能接口，和机器人控制器进行字符串交互（如果需要控制机器人，机器人控制器需要切换到“远程以太网”权限。
字符串请求格式：@@+请求内容+ $$
字符串应答格式：##+应答内容+$$

以查询系统当前运行模式为例：向机器人控制器发送字符串”@@Get_Mode$$”，机器人控制器回复字符串”##=1$$”，查看手册，返回系统当前运行模式 1 代表控制器当前处于手动模式

注意事项：
（1）字符串请求内容请参考”附录 2：API 字符串表”的说明，使用时请注意空格和分隔符；
（2）外部每发送一次数据，控制器会主动回复一次数据；没有收到回复指令之前不要发送数据；
（3）如果同时使用 API 函数和字符串，只有一个客户端能获得控制权。

附录 1：API 函数表

附录 1.1 结构体

```
typedef struct{
    double RPosData[6];   // 机器人位置坐标值
    int ArmParm[4];       // 臂参数
    double EPosData[6];   // 机器人外部轴坐标值
}ROB_POS;
typedef struct{
    double JointData[8];  // 机器人关节坐标值(第 7、8 关节值为预留位)
    double EPosData[6];   // 机器人外部轴坐标值
}ROB_JPOS;
typedef struct{
    double Data[6];       // 表示位置与方位
}POSE;
typedef struct{
    double Mass;          // 质量
    double Cog[3];        // 质心位置
    double Orient[3];     // 质心姿态
    double Inertia[3];    // 负载惯量
}LOAD_DATA;
typedef struct{
    int RobHold;          // 是否机器人夹持（1-夹持工具；0-外部工具）
    POSE TFrame;          // 工具坐标系
                          // 针对夹持工具，在腕坐标系（机器人末端法兰盘）中定义
tooldata 中 TCP 的位置和方位
                          // 针对外部工具，在世界坐标系中定义 tooldata 中 TCP 的
位置和方位
    LOAD_DATA TLoad;      // 质心姿态
    double Inertia[3];    // 工具负载
}TOOL_DATA;
typedef struct{
    int RobHold;          // 是否机器人夹持
                          // 1-机器人夹持工件(使用外部固定工具)
                          // 0-机器人未夹持工件
    int UFFix;            // 是否固定的用户坐标系
                          // 1-固定的用户坐标系（固定在世界坐标系或固定在机器人
末端）
                          // 0--可移动的用户坐标系，安装在外部设备上，伴随外部设
备移动)
    char UFMec[18];       // 关联的机械单元
                          // 仅在 UFFix = 0 时有效，使用此参数指明用户坐标系坐落
```

```
于哪个机械单元上
    double Inertia[3];   // 工具负载
    POSE UFrame;         // 用户坐标系
                          // 机器人夹持工具的场景下，在世界坐标系中定义用户坐标系
                          // 外部工具场景下(工件的 Robhold=1),在法兰坐标系(机器人末端法兰盘)中定义用户坐标系
                          // 可移动的用户坐标系(工件的 UFFix=0),用户坐标系由系统持续定义,随着关联的机械单元运动而运动
    POSE OFrame;          // 工件坐标系
                    // 工件的位置。在用户坐标系中定义工件坐标系
}WOBJ_DATA;
typedef struct{
    double vTcp;         // TCP 速度
    double vOri;         // TCP 姿态速度
    double vLeax;        // 线性外部轴速度
    double rReax;        // 旋转外部轴速度
    int bStatic;         // 1-Speed 不受全局百分比速度影响；0-Speed 受全局百分比速度影响
}SPEED;
typedef struct{
    int velType;         // 速度类型（0-百分比速度类型；1-数值型速度类型
    int velPercent;      // 百分比速度，范围 1-100
    SPEED speed;         // 数值型速度
}VER_DATA;
typedef struct{
    int IONo;           // IO 序号(上限：控制器实际配置的 IO 资源)
    int IOVa;           // IO 的输出值(0-OFF，1-ON)
    int Kind;            // 运动中设置 IO 的类型（0-时间类型，1-路径百分比，2-距离）
    double Value;       // 当类型为时间时:
                         // >=0 表示开始运动 Value 秒后输出信号,<0 表示到达运动点之前 Value 秒时输出信号
                         // 当类型为路径百分比时:
                         // >=0 表示从开始运动到达 Value%路径时输出信号，<0 表示到达运动点前 Value%路径时输出信号
                         // 当类型为距离时:
                         // >=0 表示从起点开始运动到 Value 毫米之后时输出信号,<0 表示运动到距终点 Value 毫米之前时输出信号
}MOV_IO;
/* 干涉区 */
typedef struct S_INTERFER_ZONE
{
    int32_t Input;           /* 输入信号 */
```

```
    int32_t Output;          /* 输出信号 */
    uint16_t Scope;          /* 内外侧 0-内侧, 1-外侧 */
    uint16_t IsAlert;        /* 是否报警 0-无, 1-报警 */
    float SafeL;             /* 安全距离 */
    uint16_t WobjNum;        /* 当前工件号 */

    /* 干涉区 */
    uint16_t SetType;        /* 设置方式 0-对角, 1-基准点+边长 */
    float Diagonal[6];       /* 对角点 */
    float PointL[6];         /* 基准点+偏移 */
}S_INTERFER_ZONE;

/* 多 TCP */
typedef struct S_INTERFER_TCP_BOX
{
    uint16_t IsUse[4];          /* 是否配置 */
    uint16_t ToolNum[4];        /* 工具号 */
}S_INTERFER_TCP_BOX;

/* 球型包围盒 */
typedef struct S_INTERFER_BALL_BOX
{
    float ZPos;                        /* 偏移点位 */
    float BallR;                         /* 球形半径 */
}S_INTERFER_BALL_BOX;

/* 方形包围盒 */
typedef struct S_INTERFER_SQUARE_BOX
{
    uint16_t SetType;    /* 设置方式 0-对角，1-基准点+边长，2-取点 */
    float Diagonal[6];   /* 对角点 */
    float PointL[6];     /* 基准点+偏移 */
    float PointH[13];    /* 取点+高度 */
}S_INTERFER_SQUARE_BOX;

/* 监控对象 */
typedef struct S_INTERFER_TOOL
{
    uint16_t Type;          /* 监控对象类型 0-TCP 1-MTCP 2-BALL 3-SQUARE */
    S_INTERFER_TCP_BOX MTcpBox;         /* 多 TCP */
    S_INTERFER_BALL_BOX BallBox;        /* 球形盒 */
    S_INTERFER_SQUARE_BOX SquareBox;    /* 方形盒 */
}S_INTERFER_TOOL;
```

附录 1.2 函数

序号	说明	
1	API 函数	int IMC100_Init_ETH(unsigned int ipAddr,unsigned short ipPort,int timeOut=5, int comId=0)
	说明	建立机器人网络连接
	参数	ipAddr：机器人控制器网络 IP 地址，主机字节序{0000}ipPort：机器人控制器网络端口号，默认 2222{0000}timeOut：通讯超时时间设置，默认 5s{0000}comId：连接编号，标记相同目标 IP 和端口号下不同的连接，默认值 0,最大值 9（下同）；
	应答指令解释	返回 0 表示连接成功，小于 0 表示失败
	备注	1）上位机最多支持 10 个不同的连接；控制器最多支持 4 个不同的连接{0000}2）连接编号作用域：上位机同一个进程{0000}*上位机 IP、端口号和控制器 IP、端口号，任意一个不同即为不同的连接
2	API 函数	int IMC100_Exit_ETH(int comId=0)
	说明	关闭机器人网络连接
	参数	comId：连接编号，标记对应连接{0000}（下面该参数不再赘述）
	应答指令解释	返回 0 表示关闭成功，小于 0 表示失败
	备注	
3	API 函数	int IMC100_EmergStop(int cmd, int comId=0)
	说明	急停开关控制
	参数	cmd：急停命令，1-按下急停，0-松开急停
	应答指令解释	返回 0 表示急停命令完成，小于 0 表示失败
	备注	
4	API 函数	int IMC100_MotorEnable(int cmd, int comId=0)
	说明	电机使能控制
	参数	cmd：电机使能命令，1-使能，0-去使能
	应答指令解释	返回 0 表示电机使能命令完成，小于 0 表示失败
	备注	上使能命令发出后，300ms 后读取使能状态
5	API 函数	int IMC100_ResetErr(int comId=0)
	说明	故障复位
	参数	
	应答指令解释	返回 0 表示故障复位命令完成，小于 0 表示失败
	备注	延时指令，约 50ms
6	API 函数	int IMC100_Set_Mode(int mode, int comId=0)
	说明	设置系统运行模式
	参数	mode：模式，1-手动，2-自动
	应答指令解释	返回 0 表示模式设置成功，小于 0 表示失败
	备注	
7	API 函数	int IMC100_PrgCtrl(int cmd, int comId=0)

	说明	工程运行控制
	参数	cmd：控制命令，0-停止，1-启动/继续
	应答指令解释	返回 0 表示工程运行控制成功，小于 0 表示失败
	备注	
8	API 函数	int IMC100_BackStartLine(int comId=0)
	说明	程序返回启动行
	参数	
	应答指令解释	返回 0 表示返回起始行成功，小于 0 表示失败
	备注	
9	API 函数	int IMC100_Set_Vel(int val, int comId=0)
	说明	设置当前运行速度等级
	参数	val：当前速度等级，范围 1-100
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	
10	API 函数	int IMC100_Set_AccRamp(double startVal, double endVal , int comId=0)
	说明	设置数据流模式运动段的加加速度
	参数	startVal：起始段加加速度百分比，范围 10.0-100.0{0000}endVal:结束段加加速度百分比，范围 10.0-100.0
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	仅在数据流模式下有效
11	API 函数	int IMC100_Set_RapidMove(int movType, int enableFlag, int comId=0);
	说明	设置最优轨迹规划
	参数	movType：运动类型，0-cp 运动,1-ptp 运动{0000} enableFlag：0-关闭最优规划，1-开启最优规划
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	仅在数据流模式下有效
12	API 函数	int IMC100_Set_FlyMode(int cpMode, int flyMode, int comId);
	说明	设置运动指令过渡模式（暂时设置无效）
	参数	cpMode:运动类型，0-cp 运动{0000}flyMode:过渡模式，0-自由过渡，1-固定路径过渡
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	仅在数据流模式下有效
13	API 函数	int IMC100_Set_FlyPress(int flyPressPos, int flyPressOrient, int comId);
	说明	设置固定路径的过渡应力（暂时设置无效）
	参数	flyPressPos:位置过渡应力，范围 50-200{0000}flyPressOrient:姿态过渡应力，范围 50-200
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	仅在数据流模式下有效
14	API 函数	int IMC100_DsMode(int cmd, int comId=0)
	说明	数据流模式控制

	参数	cmd：数据流命令，0-关闭，1-开启，2-暂停，3-继续
	应答指令解释	返回 0 表示数据流模式控制完成，小于 0 表示失败
	备注	
15	API 函数	int IMC100_Set_DO(int num, int status, int comId=0)
	说明	按位设置 DO 的输出状态（对象为 RC 拥有控制权的 DO）
	参数	num：DO 位序号{0000}status：DO 状态，0-off，1-On
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
16	API 函数	int IMC100_Set_SlewMode(int cmd, int comId=0);
	说明	设置 Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式
	参数	cmd：旋转优化方式{0000}0-不采用优化，以位置变量的臂参数为准；{0000}1-采用优化模式 1，保证运动中 J4/J6 中-180°~180°范围内；{0000}2-保证 J4/J6 以最近的方式运动，机器人将计算运动到目标点是否经过 J4/J6 旋转 180°，若角度差≤180°，则完全运动到目标点，若＞180°，则 J4/J6 以相反的方向运动，最终运动到距目标点的 J4/J6 相差一圈（360°）的位置；期间，J4/J6 反向运动若超过机器人极限范围，则停止运动并报警；{0000}3-保证 J4/J6 以最近的方式运动，与模式 2 的差别在于 J4/J6 反向运动若超出极限范围，不会报警，而是不进行反向运动，完全采取原方式正常运动。
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	仅在数据流模式下有效，数据流模式关闭后清零
17	API 函数	int IMC100_Set_DOGroup(int num, int status, int comId=0)
	说明	按组设置 DO 的输出状态
	参数	num：DO 组序号,最大范围 0-15，依据实际配置情况{0000}status：每组中 DO 状态，范围 0-255，其中 bit0-bit7 分别对应每组序号最小至最大的 DO 状态
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
18	API 函数	int IMC100_Set_DA(int num, float val, int comId=0)
	说明	按序号设置 DA 的输出值
	参数	num：DA 序号，最大范围 0-15{0000}val：DA 值，电流型最大范围 0-20mA，电压型最大范围-10V 到 10V，具体依据 DA 通道类型
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
19	API 函数	int IMC100_InchMode(int cmd, int comId=0)
	说明	寸动模式控制
	参数	cmd：寸动模式命令，0-关闭，1-开启
	应答指令解释	返回 0 表示寸动模式控制完成，小于 0 表示失败
	备注	
20	API 函数	int IMC100_Set_InchStep(int val, int comId=0)
	说明	设置寸动运行的步长等级
	参数	val：步长等级值，范围 0-4{0000}0-退出寸动模式{0000}1-寸动等级 G1，步长为 0.05,{0000}2-寸动等级 G2，步长为 0.3{0000}3-寸动等级

		G3，示步长为 2{0000}4-自定义寸动步长，步长为寸动参数设置值{0000}关节坐标寸动时单位为度，其他坐标寸动时单位为 mm
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
21	API 函数	int IMC100_AxisJog(int axis, int cmd, int comId);
	说明	单轴点动命令
	参数	axis：轴序号{0000}当前激活机械单元为 Robot（机器人）时，范围 1-6，关节坐标分别对应 J1-J6 轴，非关节坐标分别对应 X/Y/Z/RZ/RY/RX{0000}当前激活机械单元不为 Robot 时，Axis 范围为 1-外部机械单元总轴数{0000}cmd：单轴点动命令，0-停止，1-正向运动，-1-反向运动
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	手动模式下数据流模式关闭后有效
22	API 函数	int IMC100_AxisInch(int axis, int cmd, int comId);
	说明	寸动运动命令
	参数	axis：轴序号{0000}当前激活机械单元为 Robot（机器人）时，范围 1-6，关节坐标分别对应 J1-J6 轴，非关节坐标分别对应 X/Y/Z/RZ/RY/RX{0000}mecUnit 为外部机械时，Axis 范围为 1-外部机械单元总轴数{0000}cmd：单轴寸动命令，0-停止，1-正向寸动，-1-反向寸动
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	手动模式下数据流模式关闭后有效
23	API 函数	int IMC100_PoseAlign(int coord, int cmd, int comId);
	说明	姿态校准(适用于六关节机器人）
	参数	coord：坐标系，-1-法兰坐标系，1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系（当前仅支持法兰坐标系（当前激活工具为外部工具）、世界坐标系、基坐标系、工件坐标系）{0000}cmd：姿态校准命令，0-停止姿态校准，1-开启姿态校准
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	姿态校准过程中请勿进行其他运动，且进行姿态校准后，数据流模式将自动关闭
24	API 函数	int IMC100_Set_ActiveMechUnit(char *mecUnit, int comId=0)
	说明	设置手动模式运动激活机械单元
	参数	mecUnit：机械单元名称（大小写不敏感）
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
25	API 函数	int IMC100_Get_ActiveMechUnit(char *mecUnit, int comId=0)
	说明	获取手动模式运动的激活机械单元
	参数	mecUnit：机械单元名称
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
26	API 函数	int IMC100_Set_TeachCoordinate(int flag, int comId=0)
	说明	设置手动模式运动协调开关

	参数	flag：协调开关，0-协调开关关闭，1-协调开关开启
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	设置协调开关开启条件：{0000}（1）当前激活机械单元为导轨，当前坐标系为世界或工件坐标系；{0000}（2）当前激活机械单元为变位机，当前坐标系为工件坐标系且工件关联机械单元名称为当前激活的机械单元（即变位机）；
27	API 函数	int IMC100_Get_TeachCoordinate(int *flag, int comId=0)
	说明	获取手动模式运动协调开关
	参数	flag：协调开关，0-协调开关关闭，1-协调开关开启
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
28	API 函数	int IMC100_Home(int num, int comId=0)
	说明	回原点运动命令
	参数	num：原点序号，范围 0-4
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
29	API 函数	int IMC100_Set_DynamicBrake(int flag, int comId=0)
	说明	设置动态制动开关（适用于四关节机器人）
	参数	flag：设置动态制动开关，0-关闭动态制动开关，1-开启动态制动开关
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
30	API 函数	int IMC100_Get_DynamicBrake(int flag, int comId=0)
	说明	读取动态制动开关状态（适用于四关节机器人）
	参数	flag：获取的当前动态制动开关状态
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
31	API 函数	int IMC100_MovJ_P(int posNum, int vel=100, int zone=0, int comId=0)
	说明	关节插补方式运动到指定序号的全局位置点
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100，默认为 100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
32	API 函数	int IMC100_MovL_P(int posNum, int vel=100, int zone=0, int comId=0)
	说明	直线插补方式运动到指定序号的全局位置点
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100，默认为 100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效

33	API 函数	int IMC100_MovC_P(int posMidNum, int posDstNum, int vel=100, int zone=0, int comId=0)
	说明	圆弧插补方式运动到指定序号的全局位置点
	参数	posMidNum：圆弧中间某点的全局位置点序号，范围 0-9999{0000}posDstNum：圆弧终点的全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100，默认为 100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
34	API 函数	int IMC100_MovJ_P_IO(int posNum, int vel, int zone, MOV_IO *movIo, int ioNum, int comId=0)
	说明	关节插补方式运动到指定序号的全局位置点，同时控制对应 IO
	参数	posNum：目标位置全局点序号，范围 0-9999{0000}vel：运动速度，范围 1-100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}movIo：IO 控制结构体，详见定义{0000}ioNum：IO 控制结构体的组数，范围 1-3
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
35	API 函数	int IMC100_MovL_P_IO(int posNum, int vel, int zone, MOV_IO *movIo, int ioNum,int comId=0)
	说明	直线插补方式运动到指定序号的全局位置点，同时控制对应 IO
	参数	posNum：目标位置全局点序号，范围 0-9999{0000}vel：运动速度，范围 1-100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}movIo：IO 控制结构体，详见定义{0000}ioNum：IO 控制结构体的组数，范围 1-3
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
36	API 函数	int IMC100_MovC_P_IO(int posMidNum, int posDstNum, int vel, int zone, MOV_IO *movIo, int ioNum,int comId=0)
	说明	圆弧插补方式运动到指定序号的全局位置点，同时控制对应 IO
	参数	posMidNum：圆弧中间某点的全局位置点序号，范围 0-9999{0000}posDstNum：圆弧终点的全局位置点序号，范围 0-1000{0000}vel：运动速度，范围 1-100{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}movIo：IO 控制结构体，详见定义{0000}ioNum：IO 控制结构体的组数，范围 1-3
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
37	API 函数	int IMC100_MovJ_RobPos(ROB_POS *pos, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	关节插补方式运动到指定值的位置点，同时控制对应 IO
	参数	pos：位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为

		0{0000}ioNum：IO 控制结构体的组数，范围 0-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	1、仅在数据流模式下有效{0000}2、不支持数值型速度类型
38	API 函数	int IMC100_MovL_RobPos(ROB_POS *pos, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	直线插补方式运动到指定值的位置点
	参数	pos：位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 0-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
39	API 函数	int IMC100_MovC_RobPos(ROB_POS *posMid, ROB_POS *posDst, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	圆弧插补方式运动到指定值的位置点
	参数	pos：位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 0-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
40	API 函数	int IMC100_Jump_RobPos(ROB_POS *pos, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	跳跃插补方式运动到指定值的位置点
	参数	pos：位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 0-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
41	API 函数	int IMC100_JumpL_RobPos(ROB_POS *pos, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	直线跳跃插补方式运动到指定值的位置点
	参数	pos：位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 0-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 1 表示失败
	备注	仅在数据流模式下有效
42	API 函数	int IMC100_Jump_P(int posNum, int vel=100, int zone=0, int comId=0)

	说明	跳跃方式运动到指定序号的全局位置点
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100，默认为 100{0000}度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
43	API 函数	int IMC100_JumpL_P(int posNum, int vel=100, int zone=0, int comId=0)
	说明	直线跳跃方式运动到指定序号的全局位置点
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100，默认为 100{0000}度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
44	API 函数	int IMC100_Jump_P_IO(int posNum, int vel, int zone, MOV_IO *movIo, int ioNum, int comId=0)
	说明	跳跃方式运动到指定序号的全局位置点，同时控制对应 IO
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100{0000}度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}movIo：IO 控制结构体，详见定义{0000}ioNum：IO 控制结构体的组数，范围 1-3
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
45	API 函数	int IMC100_JumpL_P_IO(int posNum, int vel, int zone, MOV_IO *movIo, int ioNum,int comId=0)
	说明	直线跳跃方式运动到指定序号的全局位置点，同时控制对应 IO
	参数	posNum：目标全局位置点序号，范围 0-9999{0000}vel：运动速度，范围 1-100{0000}度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}movIo：IO 控制结构体，详见定义{0000}ioNum：IO 控制结构体的组数，范围 1-3
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	仅在数据流模式下有效
46	API 函数	int IMC100_MovJAbs_RobJPos(ROB_JPOS *jpos, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)
	说明	关节插补方式运动到指定值的关节点
	参数	jpos：关节位置参数结构体，详见定义{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 1-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 1 表示失败
	备注	1、仅在数据流模式下有效{0000}2、不支持数值型速度类型
47	API 函数	int IMC100_MovJAbs_JP(int posNum, VER_DATA *vel, int zone=0, int ioNum, MOV_IO *movIo, int comId=0)

	说明	关节插补方式运动到指定序号的全局关节点
	参数	posNu：目标全局关节点序号，范围 0-9999{0000}vel：速度参数结构体，详见定义{0000}zone：插补精度，范围-2 至 200 & 1000 至 1200，默认为 0{0000}ioNum：IO 控制结构体的组数，范围 1-3{0000}movIo:IO 控制结构体，详见定义
	应答指令解释	返回 0 表示命令发送成功，小于 1 表示失败
	备注	1、仅在数据流模式下有效{0000}2、不支持数值型速度类型
48	API 函数	int IMC100_IndCMove(char *mecUnit, int axis, double speed, double acc,double dec, int comId=0)
	说明	独立轴连续运动
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数{0000}speed: 数值型速度，正负代表方向且指定的速度受全局速度百分比的影响，其计算公式为：运行速度 = 指令设置的速度*全局速度百分比，单位：°/s（线性轴：mm/s）{0000}范围：-10000°/s~10000°/s（Speed = 0 代表独立轴连续运动停止）{0000}acc: 加速度，范围：20-100{0000}dec: 减速度，范围：20-100
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	1、仅在数据流模式下有效；{0000}2、请在除独立轴以外所有轴运动停止时调用该接口；{0000}3、当 Speed 大于控制器速度上限时，以控制器速度上限为准。
49	API 函数	int IMC100_IndSpeed(char *mecUnit, int axis, int *sts, int comId=0)
	说明	检测独立轴是否达到设定速度
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数{0000}sts：是否达到设定速度，1-达到指定速度，0-未达到指定速度
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
50	API 函数	int IMC100_IndResetOld(char *mecUnit, int axis, int comId=0)
	说明	重置独立轴，将独立轴切换为正常模式,轴的圈数保留
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	1、仅在数据流模式下有效；{0000}2、请在除独立轴以外所有轴运动停止时调用该接口。
51	API 函数	int IMC100_IndReset(char *mecUnit, int axis, double refNum, int direction, int comId=0)

	说明	重置独立轴，将独立轴切换为正常模式,轴的圈数根据 RefPos 与 Direction 调整
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数{0000}refPos：参考位置，范围：-720°~720° {0000}direction：{0000}-1-调整轴的多圈数，以便参考位置 refNum 位于更改后的当前位置的反向侧。（反向 360 度以内）{0000}0-调整轴的多圈数，以便更改后的当前位置尽可能地接近指定的 refNum 位置。（±180 度以内）{0000}1-调整轴的多圈数，以便参考位置 refNum 位于更改后的当前位置的正向侧。（360 度以内）
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	1、仅在数据流模式下有效；{0000}2、请在除独立轴以外所有轴运动停止时调用该接口。
52	API 函数	int IMC100_Get_IndCMoveSts(char *mecUnit, int axis, int *sts, int comId=0)
	说明	查询独立轴连续运动切换状态
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数{0000}sts：独立轴连续运动是否切换完成，0-独立轴连续运动切换未完成，1-独立轴连续运动切换完成
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
53	API 函数	int IMC100_Get_IndResetSts(char *mecUnit, int axis, int *sts, int comId=0)
	说明	查询独立轴重置状态
	参数	mecUnit：机械单元名。如果是机器人，则是 Robot{0000}axis：轴号{0000}如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴）{0000}非 Robot 机械单元，范围：1-机械单元配置的最大轴数{0000}sts：独立轴重置是否完成，0-独立轴重置未完成，1-独立轴重置完成
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
54	API 函数	int IMC100_Get_MotionStsExceptIndAxis(int *sts, int comId=0)
	说明	查询除独立轴以外的机器人运动状态
	参数	sts：系统运动状态，0-停止/运动完成，1-运动中
	应答指令解释	返回 0 表示命令发送成功，小于 0 表示失败
	备注	
55	API 函数	int IMC100_Get_RobPosHere(ROB_POS *pos, int comId=0)
	说明	获取当前位置{0000}当前工具 TCP 相对当前工件的值。如果用户需得到不同工具或不同工件下的值，需提前使用{0000}IMC100_Set_ToolCNum{0000}IMC100_Set_WobjNum
	参数	pos：位置参数结构体，代表查询的结果

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
56	API 函数	int IMC100_Get_RobJPosHere(ROB_JPOS *pos, int comId=0)
	说明	获取当前关节位置
	参数	pos：关节位置参数结构体，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
57	API 函数	int IMC100_Set_ActiveMechUnitPosFormat(int posFormat, int comId=0)
	说明	设置激活的机械单元的位置显示格式
	参数	posFormat:位置格式，1-关节坐标系格式，2-基坐标系格式，3-法兰坐标系格式，4-工件坐标系格式，5-世界坐标系格式
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
58	API 函数	int IMC100_Get_ActiveMechUnitPos(int *posFormat, double *mechUnitPos, int comId=0)
	说明	获取激活的机械单元的位置
	参数	posFormat:当前激活机械单元的位置格式，1-关节坐标系格式，2-基坐标系格式，3-法兰坐标系格式，4-工件坐标系格式，5-世界坐标系格式 {0000}mechUnitPos:机械单元位置，数组长度为 8
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
59	API 函数	int IMC100_Get_PosHerePulse(double pos[6], int comId=0)
	说明	查询当前位置点的脉冲值
	参数	pos[]：当前脉冲值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
60	API 函数	int IMC100_Get_RobJToRobP(ROB_JPOS *pos, int toolnum, int wobjnum, int loadnum, ROB_POS *posDst, int comId=0)
	说明	根据关节点计算位置点
	参数	pos：待转换的机器人关节点位{0000}toolnum：工具号{0000}wobjnum：工件号{0000}loadnum: 负载号{0000}posDst：转换后的位置点位
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
61	API 函数	int IMC100_Get_RobPToRobJ(ROB_JPOS *pos, int toolnum, int wobjnum, int loadnum, ROB_POS *posDst, int comId=0)
	说明	根据位置点计算关节点
	参数	pos：待转换的机器人点位{0000}toolnum：工具号{0000}wobjnum：工件号{0000}loadnum: 负载号{0000}posDst：转换后的关节点位
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
62	API 函数	int IMC100_Get_FixWobjRobP(ROB_POS *basePos, int wobjnum, ROB_POS *posDst, int comId=0)

	说明	计算世界坐标系下的某一个的位置在指定的非夹持的固定工件(RobHold=0, UFFix=1)坐标系下的位置
	参数	basePos：待转换的机器人点位(世界坐标系下的点位){0000}wobjnum：工件号（固定工件：RobHold=0, UFFix=1）{0000}posDst：转换后的工件坐标系下的点位
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
63	API 函数	int IMC100_Get_RobHoldWobjRobP(ROB_POS *basePos, int wobjnum, ROB_POS *posRef, ROB_POS *posDst, int comId=0)
	说明	计算世界坐标系下的某一个的位置在指定的夹持工件(RobHold=1, UFFix=1)坐标系下的位置
	参数	basePos：待转换的机器人点位(世界坐标系下的点位){0000}wobjnum：工件号（夹持工件：RobHold=1, UFFix=1）{0000}posRef：法兰在世界坐标系下的坐标值{0000}posDst：转换后的工件坐标系下的点位{0000}注意：P[*]的臂参数与 BasePos 相同
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
64	API 函数	int IMC100_Get_OffsetJ_RobJP(ROB_JPOS *pos, double PR[6], ROB_JPOS *posDst, int comId=0)
	说明	查询关节点偏移后的点位
	参数	pos：原始关节点位{0000}PR：平移变量{0000}posDst：偏移后点位结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
65	API 函数	int IMC100_Get_Offset_RobP(ROB_POS *pos, double PR[6], ROB_POS *posDst, int comId=0)
	说明	查询工件坐标系下偏移后的点位
	参数	pos：原始点{0000}PR：平移变量{0000}posDst：偏移后点位结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
66	API 函数	int IMC100_Get_OffsetT_RobP(ROB_POS *pos, double PR[6], ROB_POS *posDst, int comId=0)
	说明	查询工具坐标系下偏移后的点位
	参数	posSrc：原始点{0000}PR：平移变量{0000}posDst：偏移后点位结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
67	API 函数	int IMC100_Get_SysErrSts(int *sts, int comId=0)
	说明	查询系统当前故障状态
	参数	sts：系统故障状态，代表查询的结果，bit0-系统有报警，bit1-系统有警告
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
68	API 函数	int IMC100_Get_SysErr(int *error, int comId=0)
	说明	查询系统当前故障码

	参数	error：系统故障码，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
69	API 函数	int IMC100_Get_TaskPrgPath(int taskId, char prgPath[128], int comId)
	说明	查询当前任务通道程序的路径
	参数	taskId：任务通道，0 为主任务{0000}prgPath：当前程序路径，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
70	API 函数	int IMC100_Get_TaskRunSts(int taskId, int *sts, int comId)
	说明	查询任务通道的运行状态
	参数	taskId：任务通道，0 为主任务{0000}sts：运行状态，代表查询的结果，0-停止，1-启动/继续，10-就绪，100-任务未激活，-1-任务激活但未配置程序
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
71	API 函数	int IMC100_Get_TaskProgramLine(int taskId, int *line, int comId=0)
	说明	查询任务通道程序的当前处理的行号
	参数	line：当前任务当前处理的行号，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
72	API 函数	int IMC100_Get_CurMotionLine(int *line, int comId=0)
	说明	查询当前执行的运动指令行号
	参数	line：当前执行的运动指令行号，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
73	API 函数	int IMC100_Get_InitSts(int *sts, int comId=0)
	说明	查询系统初始化状态
	参数	sts：系统的初始化状态，代表查询的结果，范围为-1 至 11
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
74	API 函数	int IMC100_Get_AccRamp(double *startVal, double *endVal, int comId =0);
	说明	查询数据流模式运动段的加加速度
	参数	startVal：起始段加加速度百分比，范围 10.0-100.0{0000}endVal:结束段加加速度百分比，范围 10.0-100.0
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	
75	API 函数	int IMC100_Get_RapidMove(int movType, int *enableFlag, int comId =0)
	说明	查询当前运动类型最优轨迹开关

	参数	movType：运动类型，0-cp 运动,1-ptp 运动{0000}enableFlag：0-当前运动类型最优规划关闭，1-当前运动最优轨迹开启
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	
76	API 函数	int IMC100_Get_FlyMode(int cpMode, int *flyMode, int comId);
	说明	查询运动指令过渡模式
	参数	cpMode:运动类型，0-cp 运动{0000}flyMode:过渡模式，0-自由过渡，1-固定路径过渡
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	
77	API 函数	int IMC100_Get_FlyPress(int *flyPressPos, int *flyPressOrient, int comId);
	说明	查询固定路径的过渡应力
	参数	flyPressPos:位置过渡应力，范围 50-200{0000}flyPressOrient:姿态过渡应力，范围 50-200
	应答指令解释	返回 0 表示设置速度成功，小于 0 表示失败
	备注	
78	API 函数	int IMC100_Get_CoordType(int *type, int comId=0)
	说明	查询当前坐标系类型
	参数	type：当前坐标系类型，范围 1 至 5，1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
79	API 函数	int IMC100_Get_Vel(int *val, int comId=0)
	说明	查询当前全局速度值
	参数	val：当前全局速度值，代表查询的结果，范围 1-100
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
80	API 函数	int IMC100_Get_Mode(int *mode, int comId=0)
	说明	查询系统当前运行模式
	参数	mode：系统运行模式，代表查询的结果，1-手动，2-自动，3-单步运行，5-连续运行
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
81	API 函数	int IMC100_Get_DsMode(int *val, int comId=0)
	说明	查询数据流模式是否开启
	参数	val：数据流模式开启情况，代表查询的结果，0-关闭，1-开启/继续，2-暂停
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
82	API 函数	int IMC100_Get_InchMode(int *val, int comId=0)
	说明	查询寸动模式
	参数	val：寸动模式，代表查询的结果，0-非寸动，1-寸动

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
83	API 函数	int IMC100_Get_SlewMode(int *val, int comId=0)
	说明	查询 Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式
	参数	val：Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式，代表查询的结果{0000}0-不采用优化，以位置变量的臂参数为准；{0000}1-采用优化模式 1，保证运动中 J4/J6 中-180°~180°范围内；{0000}2-保证 J4/J6 以最近的方式运动，机器人将计算运动到目标点是否经过 J4/J6 旋转 180°，若角度差≤180°，则完全运动到目标点，若＞180°，则 J4/J6 以相反的方向运动，最终运动到距目标点的 J4/J6 相差一圈（360°）的位置；期间，J4/J6 反向运动若超过机器人极限范围，则停止运动并报警；{0000}3-保证 J4/J6 以最近的方式运动，与模式 2 的差别在于 J4/J6 反向运动若超出极限范围，不会报警，而是不进行反向运动，完全采取原方式正常运动。
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
84	API 函数	int IMC100_Get_EStopSts(int *sts, int comId=0)
	说明	查询当前急停开关状态
	参数	sts：急停开关状态，代表查询的结果，0-急停松开，1-急停按下
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
85	API 函数	int IMC100_Get_MotorSts(int *sts, int comId=0)
	说明	查询当前电机使能状态
	参数	sts：电机使能状态，代表查询的结果，0-未使能，1-使能
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
86	API 函数	int IMC100_Get_MotionSts(int *sts, int comId=0)
	说明	查询当前系统运动状态
	参数	sts：系统运动状态，代表查询的结果，0-停止/运动完成，1-运动中，2-运动中断
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
87	API 函数	int IMC100_Get_SysMode(int *mode, int comId=0)
	说明	查询系统当前模式
	参数	mode：系统模式，代表查询的结果，0-正常模式,>0-内部测试模式
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
88	API 函数	int IMC100_Get_PrgRunTime(unsigned int *second, int comId=0)
	说明	查询工程运行时间
	参数	second：时间计数值（秒），代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
89	API 函数	int IMC100_Get_CurCmdNum(unsigned int *num, int comId=0)

	说明	查询当前发送成功的运动指令的编号
	参数	num：指令编号，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	仅在数据流模式下有效
90	API 函数	int IMC100_Get_CurCmdSts(int *sts, int comId=0)
	说明	查询当前发送成功的运动指令实际完成状态（是否到位）
	参数	sts：完成状态，代表查询的结果，0-运动未完成，1-运动完成
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	1、仅在数据流模式下有效{0000}2、该接口判断的是除独立轴外的机器人运行状态，如果最后一段运动指令是独立轴，请使用 IMC100_IndSpeed 判断是否到位{0000}3、请勿发送重复点后用 IMC100_Get_CurCmdSts 判断到位
91	API 函数	int IMC100_Get_CurCmdCacheNum(int *num, int comId);
	说明	查询当前缓存区运动数据的数量
	参数	num：当前缓存区运动数据的数量
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
92	API 函数	int IMC100_Get_CmdSts(int num, int *sts, int comId=0)
	说明	查询指定编号的运动指令实际完成状态
	参数	num：指令编号{0000}sts：完成状态，代表查询的结果，0-运动未完成，1-运动完成
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	仅在数据流模式下有效
93	API 函数	int IMC100_Get_DINum(int *num, int comId=0)
	说明	查询系统 DI 总数
	参数	num：DI 总数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
94	API 函数	int IMC100_Get_DONum(int *num, int comId=0)
	说明	查询系统 DO 总数
	参数	num：DO 总数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
95	API 函数	int IMC100_Get_ADNum(int *num, int comId=0)
	说明	查询系统 AD 总数
	参数	num：AD 总数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
96	API 函数	int IMC100_Get_DANum(int *num, int comId=0)
	说明	查询系统 DA 总数
	参数	num：DA 总数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	

97	API 函数	int IMC100_Get_DI(int num, int *sts, int comId=0)
	说明	按位查询 DI 的输入状态
	参数	num：DI 序号（不超过 DI 总数）{0000}sts：DI 状态，代表查询的结果，0-off，1-On
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
98	API 函数	int IMC100_Get_DIGroup(int num, int *sts, int comId=0)
	说明	按组查询 DI 的输入状态
	参数	num：DI 组序号{0000}sts：每组的 DI 状态，代表查询的结果，范围 0-255，其中 bit0-bit7 分别对应每组序号最小至最大的 DI 状态
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
99	API 函数	int IMC100_Get_AD(int num, float *val, int comId=0)
	说明	按序号查询 AD 的输入值
	参数	num：AD 序号（不超过 AD 总数）{0000}val：AD 值，代表查询的结果，电流型单位为 mA，电压型单位为 V
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
100	API 函数	int IMC100_Get_DOCfg(int num, int *val, int comId=0)
	说明	查询 DO 的配置权
	参数	num：DO 序号（不超过 DO 总数）{0000}val：配置权，代表查询的结果，1 表由 RC 控制，0 表由 PLC 控制
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
101	API 函数	int IMC100_Get_DOGroupCfg(int num, int *val, int comId=0)
	说明	查询每组 DO 的配置权
	参数	num：DO 组号{0000}val：配置权，代表查询的结果，bit0-bit7 分别代表该组每个 DO 的配置权，1 表由 RC 控制，0 表由 PLC 控制
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
102	API 函数	int IMC100_Get_DO(int num, int *sts, int comId=0)
	说明	按位查询 DO 的输出状态
	参数	num：DO 序号（不超过 DO 总数）{0000}sts：DO 状态，代表查询的结果，0-off，1-On
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
103	API 函数	int IMC100_Get_DOGroup(int num, int *sts, int comId=0)
	说明	按组查询 DO 的输出状态
	参数	num：DO 组序号{0000}sts：每组的 DO 状态，代表查询的结果，范围 0-255，其中 bit0-bit7 分别对应每组序号最小至最大的 DO 状态
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
104	API 函数	int IMC100_Get_DACfg(int num, int *val, int comId=0)

	说明	查询 DA 的配置权
	参数	num：DA 序号{0000}val：配置权，代表查询的结果，1 表由 RC 控制，0 表不可由 RC 控制（由 PLC 控制或未连接）
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
105	API 函数	int IMC100_Get_DA(int num, float *val, int comId=0)
	说明	按序号查询 DA 的输出值
	参数	num：DA 序号（不超过 DA 总数）{0000}val：DA 值，代表查询的结果，电流型单位为 mA，电压型单位为 V
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
106	API 函数	int IMC100_Get_DevSts(int sts[6], int comId=0)
	说明	查询系统设备的连接情况
	参数	sts[]：系统设备连接情况，代表查询的结果。其中：{0000}sts[0]：网卡 1 状态，0 表未连接，1 表连接，2 表被禁用{0000}sts[1]：网卡 2 状态，同上{0000}sts[2]：USB 设备状态，0 表未连接，1 表连接挂载成功，2 表挂载失败{0000}sts[3]：SD 卡状态，0 表未连接，1 表连接挂载成功，2 表挂载失败，3 表文件系统格式错误{0000}sts[4]：EtherCAT1 通信状态，0 表通信正常，1 表从站掉线，2 表未连接网线，3 表连接非 ECAT 设备，4 表已禁用{0000}sts[5]：IRLink1 通信状态，0 表通讯正常，1 表从站掉线，2 表设备未挂载，3 表连接了非 IR-Link 设备，4 表被禁用，5 表设备未配置，6 表通讯异常
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
107	API 函数	int IMC100_Get_FwVersion(char ver[32], int comId=0)
	说明	查询系统控制器软件版本
	参数	ver[]：当前系统软件版本，代表查询的结果，例如 S03.20R
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
108	API 函数	int IMC100_Get_SysTime(char time[16], int comId=0)
	说明	查询当前系统时间
	参数	time[]：时间字符串（年月日时分秒），代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
109	API 函数	int IMC100_Get_RobotType(char type[128], int comId=0)
	说明	查询当前系统机型
	参数	type[]：机型字符串，代表查询结果，例如 Scara_A_Ino1
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
110	API 函数	int IMC100_Get_ArmType(double pos[6], int armType[4], int comId=0)
	说明	根据关节坐标值查询臂参数
	参数	pos[]:关节坐标值{0000}armType[]:臂参数，代表查询结果

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
111	API 函数	int IMC100_Get_ServoSts(int sts[8], int comId=0)
	说明	查询系统中所有伺服的故障状态（包括机器人轴和外部轴两部分）
	参数	sts[8]：伺服的故障状态，代表查询的结果。目前最大支持 8 个伺服轴，sts[0]对应第 0 号伺服，依次类推，0-无故障，bit0-有伺服报警，bit1-有伺服警告
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
112	API 函数	int IMC100_Get_ServoErr(int num, int *error, int comId=0)
	说明	查询系统中单个伺服的故障码（机器人轴）
	参数	num：伺服轴序号，从 0 开始{0000}error：伺服故障码，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
113	API 函数	int IMC100_Get_StrPara(float para[6], int comId=0)
	说明	查询机器人结构参数
	参数	para[]：结构参数，代表查询的结果（对于 SCARA 机器人，para[0]-para[3]有效，对于六轴机器人，para[0]-para[5]有效，下同）
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
114	API 函数	int IMC100_Set_StrPara(float para[6], int comId=0)
	说明	设置机器人结构参数
	参数	para[]：结构参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
115	API 函数	int IMC100_Get_StrParaComp(float para[6], int comId=0)
	说明	查询机器人结构补偿参数
	参数	para[]：结构补偿参数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
116	API 函数	int IMC100_Set_StrParaComp(float para[6], int comId=0)
	说明	设置机器人结构补偿参数
	参数	para[]：结构补偿参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
117	API 函数	int IMC100_Get_RdctRatio(float para[6], int comId=0)
	说明	查询关节减速比
	参数	para[]：各关节减速比，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
118	API 函数	int IMC100_Set_RdctRatio(float para[6], int comId=0)
	说明	设置关节减速比
	参数	para[]：各关节减速比

	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
119	API 函数	int IMC100_Get_CpParaM(float para[6], int comId=0)
	说明	查询关节主耦合参数
	参数	para[]：各关节主耦合参数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
120	API 函数	int IMC100_Set_CpParaM(float para[6], int comId=0)
	说明	设置关节主耦合参数
	参数	para[]：各关节主耦合参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
121	API 函数	int IMC100_Get_CpParaS(float para[6], int comId=0)
	说明	查询关节从耦合参数
	参数	para[]：各关节从耦合参数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
122	API 函数	int IMC100_Set_CpParaS(float para[6], int comId=0)
	说明	设置关节从耦合参数
	参数	para[]：各关节从耦合参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
123	API 函数	int IMC100_Get_HomeJPos(int num, ROB_JPOS *pos, int comId=0)
	说明	查询工作原点
	参数	num：工作原点序号，范围 0-4{0000}pos[]：工作原点对应的关节坐标值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
124	API 函数	int IMC100_Set_HomeJPos(int num, ROB_JPOS *pos, int comId=0)
	说明	设置工作原点
	参数	num：工作原点序号，范围 0-4{0000}pos[]：工作原点对应的关节坐标值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
125	API 函数	int IMC100_Get_ZeroPos(int pluse[6], int comId=0)
	说明	查询绝对零点
	参数	pluse[]：绝对零点对应的脉冲值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
126	API 函数	int IMC100_Set_ZeroPos(int pluse[6], int comId=0)
	说明	设置绝对零点
	参数	pluse[]：绝对零点对应的脉冲值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用

127	API 函数	int IMC100_Get_InchStep(int *val, int comId=0)
	说明	查询寸动步长等级
	参数	val：寸动运行的步长等级，代表查询结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
128	API 函数	int IMC100_Get_StepMotionJ(float *para, int comId=0)
	说明	查询寸动的自定义关节步长
	参数	para：自定义关节步长值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
129	API 函数	int IMC100_Set_StepMotionJ(float para, int comId=0)
	说明	设置寸动的自定义关节步长
	参数	para：自定义关节步长值，范围 0.01-10
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
130	API 函数	int IMC100_Get_StepMotionL(float *para, int comId=0)
	说明	查询寸动的自定义线性步长
	参数	para：自定义线性步长值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
131	API 函数	int IMC100_Set_StepMotionL(float para, int comId=0)
	说明	设置寸动的自定义线性步长
	参数	para：自定义线性步长值，范围 0.01-10
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
132	API 函数	int IMC100_Get_StepMotionR(float *para, int comId=0)
	说明	查询寸动的自定义姿态步长
	参数	para：自定义姿态步长值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
133	API 函数	int IMC100_Set_StepMotionR(float para, int comId=0)
	说明	设置寸动的自定义姿态步长
	参数	para：自定义姿态步长值，范围 0.01-10
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
134	API 函数	int IMC100_Get_TeachVelLimJ(float para[6], int comId=0)
	说明	查询手动模式运动时关节速度上限
	参数	para[]：最大允许关节速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
135	API 函数	int IMC100_Set_TeachVelLimJ(float para[6], int comId=0)
	说明	设置手动模式运动时关节速度上限
	参数	para[]：最大允许关节速度

	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
136	API 函数	int IMC100_Get_TeachVelLimL(float para[2], int comId=0)
	说明	查询手动模式运动时位置、姿态速度上限
	参数	para[2]：最大允许位置和姿态速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
137	API 函数	int IMC100_Set_TeachVelLimL(float para[2], int comId=0)
	说明	设置手动模式运动时位置和姿态速度上限
	参数	para[2]：最大允许位置和姿态速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
138	API 函数	int IMC100_Get_TeachAccLimJ(float para[6], int comId=0)
	说明	查询手动模式运动时关节加速度上限
	参数	para[]：最大允许关节加速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
139	API 函数	int IMC100_Set_TeachAccLimJ(float para[6], int comId=0)
	说明	设置手动模式运动时关节加速度上限
	参数	para[]：最大允许关节加速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
140	API 函数	int IMC100_Get_TeachAccLimL(float para[2], int comId=0)
	说明	查询手动模式运动时位置、姿态加速度上限
	参数	para[]：最大允许位置和姿态加速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
141	API 函数	int IMC100_Set_TeachAccLimL(float para[2], int comId=0)
	说明	设置手动模式运动时位置和姿态加速度上限
	参数	para[]：最大允许位置和姿态加速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
142	API 函数	int IMC100_Get_RunVelLimJ(float para[6], int comId=0)
	说明	查询运行时关节速度上限
	参数	para[]：最大允许关节速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
143	API 函数	int IMC100_Set_RunVelLimJ(float para[6], int comId=0)
	说明	设置运行时关节速度上限
	参数	para[]：最大允许关节速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
144	API 函数	int IMC100_Get_RunVelLimL(float para[2], int comId=0)

	说明	查询运行时位置、姿态速度上限
	参数	para[]：最大允许位置和姿态速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
145	API 函数	int IMC100_Set_RunVelLimL(float para[2], int comId=0)
	说明	设置运行时位置、姿态速度上限
	参数	para[]：最大允许位置和姿态速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
146	API 函数	int IMC100_Get_RunAccLimJ(float para[6], int comId=0)
	说明	查询运行时关节加速度上限
	参数	para[]：最大允许关节加速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
147	API 函数	int IMC100_Set_RunAccLimJ(float para[6], int comId=0)
	说明	设置运行时关节加速度上限
	参数	para[]：最大允许关节加速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
148	API 函数	int IMC100_Get_RunAccLimL(float para[2], int comId=0)
	说明	查询运行时位置、姿态加速度上限
	参数	para[]：最大允许位置和姿态加速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
149	API 函数	int IMC100_Set_RunAccLimL(float para[2], int comId=0)
	说明	设置运行时位置、姿态加速度上限
	参数	para[2]：最大允许位置和姿态加速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
150	API 函数	int IMC100_Get_StopDecLimJ(float para[6], int comId=0)
	说明	查询运行时关节减速度上限
	参数	para[]：最大允许关节减速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
151	API 函数	int IMC100_Set_StopDecLimJ(float para[6], int comId=0)
	说明	设置运行时关节减速度上限
	参数	para[]：最大允许关节减速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
152	API 函数	int IMC100_Get_StopDecLimL(float para[2], int comId=0)
	说明	查询运行时位置、姿态减速度上限
	参数	para[]：最大允许位置和姿态减速度，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败

	备注	
153	API 函数	int IMC100_Set_StopDecLimL(float para[2], int comId=0)
	说明	设置运行时位置、姿态减速度上限
	参数	para[]：最大允许位置和姿态减速度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
154	API 函数	int IMC100_Get_ZonePara(float para[2], int comId=0)
	说明	查询过渡精度参数
	参数	para[]：线性和关节过渡精度，代表查询的结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
155	API 函数	int IMC100_Set_ZonePara(float para[2], int comId=0)
	说明	设置过渡精度参数，参数分别为线性和关节过渡精度
	参数	para[]：线性和关节过渡精度
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
156	API 函数	int IMC100_Get_AxisNLim(int axis, float *para, int comId=0)
	说明	查询机器人轴的负向轴极限
	参数	axis：轴序号，与轴数有关，范围 1-6，分别对应 J1-J6 轴{0000}para：负向轴极限值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
157	API 函数	int IMC100_Set_AxisNLim(int axis, float para, int comId=0)
	说明	设置机器人轴的负向轴极限
	参数	axis：轴序号，范围 1-6，分别对应 J1-J6 轴{0000}para：负向轴极限值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
158	API 函数	int IMC100_Get_AxisPLim(int axis, float *para, int comId=0)
	说明	查询机器人轴的正向轴极限
	参数	axis：轴序号，范围 1-6，分别对应 J1-J6 轴{0000}para：正向轴极限值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
159	API 函数	int IMC100_Set_AxisPLim(int axis, float para, int comId=0)
	说明	设置机器人轴的正向轴极限
	参数	axis：轴序号，范围 1-6，分别对应 J1-J6 轴{0000}para：正向轴极限值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
160	API 函数	int IMC100_Get_ToolData(int num, TOOL_DATA *toolData, int comId=0)
	说明	查询工具坐标系参数
	参数	num：工具号，范围 0-15{0000}toolData：工具参数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败

	备注	
161	API 函数	int IMC100_Set_ToolData(int num, TOOL_DATA *toolData, int comId=0)
	说明	设置工具坐标系参数
	参数	num：工具号，范围 1-15{0000}toolData：工具参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
162	API 函数	int IMC100_Get_WobjData(int num, WOBJ_DATA *wobjData, int comId=0)
	说明	查询工件坐标系参数
	参数	num：工件号，范围 0-15{0000}wobjData：工件参数，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
163	API 函数	int IMC100_Set_WobjData(int num, WOBJ_DATA *wobjData, int comId=0)
	说明	设置工件坐标系参数
	参数	num：工件号，范围 1-15{0000}wobjData：：工件参数（关联的机械单元，大小写不敏感）
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
164	API 函数	int IMC100_Get_ToolCNum(int *num, int comId=0)
	说明	查询系统当前激活工具号
	参数	num：工具号，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
165	API 函数	int IMC100_Set_ToolCNum(int num, int comId=0)
	说明	设置系统激活的工具号
	参数	num：工具号，范围 0-15
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
166	API 函数	int IMC100_Get_WobjNum(int *num, int comId=0)
	说明	查询系统当前激活工件号
	参数	num：工件号，代表查询的结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
167	API 函数	int IMC100_Set_WobjNum(int num, int comId=0)
	说明	设置系统激活的工件号
	参数	num：工件号，范围 0-15
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	管理模式及以上使用
168	API 函数	int IMC100_Set_CoordType(int type, int comId=0)
	说明	设置当前坐标系类型

	参数	type：当前坐标系类型，范围 1 至 5，1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
169	API 函数	int IMC100_Get_Interf(int num, double pos[6], int comId=0)
	说明	查询干涉区边界点坐标参数
	参数	num：干涉区序号，范围 0 至 7{0000}pos[]：干涉区边界点坐标，代表查询的结果，pos[0]至 pos[2]分别对应点 1 的 XYZ 坐标，pos[3]至 pos[5]分别对应点 2 的 XYZ 坐标
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
170	API 函数	int IMC100_Set_Interf(int num, double pos[6], int comId=0)
	说明	设置干涉区边界点坐标参数
	参数	num：干涉区序号{0000}pos[]：干涉区边界点坐标
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
171	API 函数	int IMC100_Get_CurInterf(int *num, int comId=0)
	说明	查询当前激活的干涉区编号
	参数	num：干涉区编号，代表查询的结果，范围 0 至 255，其中 bit0 至 bit7 分别对应干涉区 0 至干涉区 7,0-未激活，1-激活
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
172	API 函数	int IMC100_Set_CurInterf(int num, int comId=0)
	说明	设置需激活的干涉区编号
	参数	num：干涉区编号，同上
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
173	API 函数	int IMC100_Get_JumpPara(float *lh, float *mh, float *rh, int comId=0)
	说明	查询跳跃运动(Jump、JumpL)的高度参数
	参数	lh：起始位置提升高度，范围 0 至 2000，代表查询结果{0000}mh：运动最高点相对基坐标系零点的高度，范围-2000 至 2000，代表查询结果{0000}rh：到终止位置的下降高度，范围 0 至 2000，代表查询结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
174	API 函数	int IMC100_Set_JumpPara(float lh, float mh, float rh, int comId=0)
	说明	设置跳跃运动(Jump、JumpL)的高度参数
	参数	lh：起始位置提升高度，范围 0 至 2000{0000}mh：运动最高点相对基坐标系零点的高度，范围-2000 至 2000{0000}rh：到终止位置的下降高度，范围 0 至 2000
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	只对数据流模式有效

175	API 函数	int IMC100_Get_PalletPara(int *rowNum, int *colNum, int *layerNum, double *layerHeight, int comId=0)
	说明	查询托盘参数
	参数	rowNum：行数，范围 0 至 1000{0000}colNum：列数，范围 0 至 1000{0000}layerNum：层数，范围 0 至 1000{0000}layerHeight：层高，单位：mm
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
176	API 函数	int IMC100_Set_PalletPara(int rowNum, int colNum, int layerNum, double layerHeight, int comId=0)
	说明	设置托盘参数
	参数	rowNum：行数，范围 0 至 1000{0000}colNum：列数，范围 0 至 1000{0000}layerNum：层数，范围 0 至 1000{0000}layerHeight：层高，单位：mm
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
177	API 函数	int IMC100_Clear_PalletPara(int comId=0)
	说明	清空托盘参数
	参数	清空托盘参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
178	API 函数	int IMC100_Get_Pallet_RobP(ROB_POS *pos1, ROB_POS *pos2, ROB_POS *pos3, int rowIndex, int colIndex, int layIndex, ROB_POS *posDst, int comId=0)
	说明	查询对应的托盘点位（定义托盘边界点数：3 点）
	参数	pos1~3:定义托盘的 3 个点{0000}rowIndex：待查询点位行号{0000}colIndex：待查询点位列号{0000}layIndex：待查询点位层号{0000}posDst：查询点位结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
179	API 函数	int IMC100_Get_Pallet4_RobP(ROB_POS *pos1, ROB_POS *pos2, ROB_POS *pos3, ROB_POS *pos4, int rowIndex, int colIndex, int layIndex, ROB_POS *posDst, int comId=0)
	说明	查询对应的托盘点位（定义托盘边界点数：4 点）
	参数	pos1~4:定义托盘的 4 个点{0000}rowIndex：待查询点位行号{0000}colIndex：待查询点位列号{0000}layIndex：待查询点位层号{0000}posDst：查询点位结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
180	API 函数	int IMC100_SavePara(int comId=0)
	说明	保存系统参数，掉电可存储
	参数	
	应答指令解释	返回 0 表示操作成功，小于 0 表示失败

	备注	管理模式及以上使用
181	API 函数	int IMC100_RecoverPara(int comId=0)
	说明	恢复系统参数（恢复至上一次保存操作后的参数）
	参数	
	应答指令解释	返回 0 表示操作成功，小于 0 表示失败
	备注	管理模式及以上使用
182	API 函数	int IMC100_Get_RobJP(int pNum, ROB_JPOS *pos, int comId=0)
	说明	查询全局关节位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：关节位置参数结构体，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
183	API 函数	int IMC100_Set_RobJP(int pNum, ROB_JPOS *pos, int comId=0)
	说明	设置全局关节位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：关节位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
184	API 函数	int IMC100_Set_RobJPHere(int pNum, int comId=0)
	说明	用当前点位的参数设置全局关节位置参数
	参数	pNum：全局关节位置变量序号
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
185	API 函数	int IMC100_Get_RobP(int pNum, ROB_POS *pos, int comId=0)
	说明	查询全局位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：位置参数结构体，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
186	API 函数	int IMC100_Set_RobP(int pNum, ROB_POS *pos, int comId=0)
	说明	设置全局位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
187	API 函数	int IMC100_Get_RobPFromFile(char *pFileName, int pNum, ROB_POS *pos, int comId=0)
	说明	从文件中查询全局位置变量对应的位置参数
	参数	pFileName：全局位置变量文件名（API 不区分大小写）{0000}pNum：全局位置变量序号，范围 0-9999{0000}pos：位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
188	API 函数	int IMC100_Set_RobPToFile(char *pFileName, int pNum, ROB_POS *pos, int comId=0)
	说明	设置全局位置变量对应的位置参数到指定文件中

	参数	pFileName：全局位置变量文件名（API 不区分大小写）{0000}pNum：全局位置变量序号，范围 0-9999{0000}pos：位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
189	API 函数	int IMC100_Get_CurRobPFileName(char *pFileName, int comId=0)
	说明	查询当前加载的全局位置变量文件
	参数	pFileName：全局位置变量文件名（API 不区分大小写）
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
190	API 函数	int IMC100_Set_RobPHere(int pNum, int comId=0)
	说明	用当前点位的参数设置全局位置参数
	参数	pNum：全局位置变量序号
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
191	API 函数	int IMC100_Set_RobPHereToFile(char *pFileName,int pNum, int comId=0)
	说明	用当前点位的参数设置指定 P 点文件的点位数据
	参数	pFileName：全局位置变量文件名（API 不区分大小写）{0000}pNum：全局位置变量序号，范围 0-9999
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
192	API 函数	int IMC100_Get_RobP_Label(int pNum, char *plabelName, int comId)
	说明	查询全局位置点标签信息
	参数	pNum：全局位置点序号{0000}plabelName：标签信息
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
193	API 函数	int IMC100_Get_RobJP_Label(int pNum, char *plabelName, int comId)
	说明	查询全局关节点标签信息
	参数	pNum：全局关节点序号{0000}plabelName：标签信息
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
194	API 函数	int IMC100_Get_RobLP_Label(int taskId, char *pProName, int pNum, char *plabelName, int comId)
	说明	查询局部位置点标签信息
	参数	taskId：任务通道{0000}pProName：局部点位所在程序名（大小写不敏感，需带.pro 后缀）{0000}pNum：局部位置点序号{0000}plabelName：标签信息
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
195	API 函数	int IMC100_Get_SpeedLimitSwitch(int *sLimitSwitch, int comId)
	说明	获取限速开关状态
	参数	sLimitSwitch：限速开关状态 0：开启限速　1：关闭限速

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
196	API 函数	int IMC100_Set_SpeedLimitSwitch(int sLimitSwitch, int comId)
	说明	设置机器人是否限速
	参数	sLimitSwitch：限速开关状态 0：开启限速 1：关闭限速
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	编辑模式及以上使用
197	API 函数	int IMC100_Get_PRVar(int prNum, POSE *pos, int comId=0)
	说明	查询全局平移变量对应的参数
	参数	prNum：全局平移变量序号，范围 0 至 255{0000}pos：位置参数结构体，代表查询的结果，其中臂参数无效
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
198	API 函数	int IMC100_Set_PRVar(int prNum, POSE pos, int comId=0)
	说明	设置全局平移变量对应的参数
	参数	prNum：全局平移变量序号，范围 0 至 255{0000}pos：位置参数结构体，其中臂参数无效
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
199	API 函数	int IMC100_Get_B(int num, int *val, int comId=0)
	说明	查询全局 B 变量的值
	参数	num：B 变量序号，范围 0-255{0000}val：B 变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
200	API 函数	int IMC100_Set_B(int num, int val, int comId=0)
	说明	设置全局 B 变量的值
	参数	num：B 变量序号，范围 0-255{0000}val：B 变量值，范围 0-255
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
201	API 函数	int IMC100_Get_R(int num, int *val, int comId=0)
	说明	查询全局 R 变量的值
	参数	num：R 变量序号，范围 0-255{0000}val：R 变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
202	API 函数	int IMC100_Set_R(int num, int val, int comId=0)
	说明	设置全局 R 变量的值
	参数	num：R 变量序号，范围 0-255{0000}val：R 变量值，范围-2147483647-2147483647
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
203	API 函数	int IMC100_Get_D(int num, double *val, int comId=0)
	说明	查询全局 D 变量的值
	参数	num：D 变量序号，范围 0-255{0000}val：D 变量值，代表查询的结果

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
204	API 函数	int IMC100_Set_D(int num, double val, int comId=0)
	说明	设置全局 D 变量的值
	参数	num：D 变量序号，范围 0-255{0000}val：D 变量值，范围-9999999.999 至 9999999.999
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
205	API 函数	int IMC100_Get_ModbusCoil(int address, int sum, int *val, int comId=0)
	说明	查询 Modbus 变量区的线圈值
	参数	address：Modbus 区线圈地址，范围 0-8191{0000}sum：读取的线圈总个数，范围 1-8{0000}val：线圈值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
206	API 函数	int IMC100_Set_ModbusCoil(int address, int sum, int val, int comId=0)
	说明	设置 Modbus 变量区的线圈值
	参数	address：Modbus 区线圈地址，范围 2048-4095,6144-8191{0000}sum：读取的线圈总个数，范围 1-8{0000}val：线圈值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
207	API 函数	int IMC100_Get_ModbusRegUshort(int address, int sum, unsigned short val[], int comId=0)
	说明	查询 Modbus 变量区的寄存器值，数据类型为 unsigned short
	参数	address：modbus 区寄存器地址，范围 0-65535{0000}sum：读取的寄存器总个数,范围 1-8{0000}val：代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
208	API 函数	int IMC100_Set_ModbusRegUshort(int address, int sum, unsigned short val[], int comId=0)
	说明	设置 Modbus 变量区的寄存器值,数据类型为 unsigned short
	参数	address：modbus 区寄存器地址，范围 16384-32767,49152-65535{0000}sum：读取的寄存器总个数,范围 1-8{0000}val：代表查询的结果
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
209	API 函数	int IMC100_Get_ModbusRegFloat(int address, int sum, float val[], int comId=0)
	说明	查询 Modbus 变量区的寄存器值，数据类型为 float
	参数	address：modbus 区寄存器地址，范围 0-65535{0000}sum：读取的寄存器总个数,范围 1-8{0000}val：代表查询的结果(一个 float 数据占用 2 个寄存器)

	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
210	API 函数	int IMC100_Set_ModbusRegFloat(int address, int sum, float val[], int comId=0)
	说明	设置 Modbus 变量区的寄存器值,数据类型为 float
	参数	address：modbus 区寄存器地址，范围 16384-32767,49152-65535{0000}sum：读取的寄存器总个数,范围 1-8{0000}val：变量值
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	编辑模式及以上使用
211	API 函数	int IMC100_Get_PlcVarByte(int num, unsigned char *val, int comId=0)
	说明	查询 PLC Byte 型变量的值
	参数	num：Byte 变量序号，范围 0-255{0000}val：变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
212	API 函数	int IMC100_Get_PlcVarInt(int num, short *val, int comId=0)
	说明	查询 PLC Int 型变量的值
	参数	num：Int 变量序号，范围 0-255{0000}val：变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
213	API 函数	int IMC100_Get_PlcVarDInt(int num, int *val, int comId=0)
	说明	查询 PLC DInt 型变量的值
	参数	num：DInt 变量序号，范围 0-255{0000}val：变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
214	API 函数	int IMC100_Get_PlcVarLReal(int num, double *val, int comId=0)
	说明	查询 PLC LReal 型变量的值
	参数	num：LReal 变量序号，范围 0-255{0000}val：变量值，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
215	API 函数	int IMC100_Get_UserAlarm(int num, char alarm[40], int comId=0)
	说明	查询自定义报警的内容
	参数	num：自定义报警序号，范围 0-15{0000}alarm：报警内容描述，代表查询的结果，字节长不超过 40 bytes
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
216	API 函数	int IMC100_Set_UserAlarm(int num, char alarm[40], int comId=0)
	说明	设置自定义报警的内容
	参数	num：自定义报警序号，范围 0-15{0000}alarm：报警内容描述，字节长不超过 40byte
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	管理模式及以上使用
217	API 函数	int IMC100_Get_Print(char val[120], int comId=0)

	说明	查询控制器打印信息，包括程序 print 指令的打印内容和系统错误提示内容
	参数	val：打印内容，代表查询的结果，字节长不超过 120 bytes
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
218	API 函数	int IMC100_CurCtrlDev(int *dev, int comId=0)
	说明	查询当前控制权所属设备
	参数	dev：当前控制权设备编号，0-InoTeachPad/InoRobotLab，1-InoRobShop 平台，2-远程以太网设备，3-远端 IO，4-远端 modbus
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
219	API 函数	int IMC100_CurPermit(int *owner, unsigned int *ipAddr, unsigned short *ipPort, int comId=0)
	说明	查询当前拥有控制权许可的以太网设备信息
	参数	owner：获得许可的以太网设备身份，代表查询的结果，0-无以太网设备获得许可，1-当前设备获得许可，2-其它以太网设备获得许可 {0000}ipAddr：设备网络 IP 地址，代表查询的结果，当第一个返回值为 0 时，该值无意义{0000}ipPort：设备网络端口号，代表查询的结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
220	API 函数	int IMC100_AcqPermit(int cmd=0, int comId=0)
	说明	当前 API 网络客户端设备请求获取控制权许可
	参数	cmd：请求命令，0 表示一般请求，1 表示强制获取，默认为 0
	应答指令解释	返回 0 表示获取成功，小于 0 表示失败
	备注	
221	API 函数	int IMC100_RemovePermit(int comId=0)
	说明	当前 API 网络客户端设备释放控制权
	参数	
	应答指令解释	返回 0 表示释放成功，小于 0 表示失败
	备注	
222	API 函数	int IMC100_CurUserType(int *type, int comId=0)
	说明	查询当前用户的模式
	参数	type：用户模式，代表查询的结果，0-客户模式，1-编辑模式，2-管理模式，3-厂家模式
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
223	API 函数	int IMC100_UserLogin(int type, char password[8], int comId=0)
	说明	当前 API 网络客户端设备登陆对应的用户模式
	参数	type：用户模式，0-客户模式，1-编辑模式，2-管理模式，3-厂家模式 {0000}password：登陆的密码
	应答指令解释	返回 0 表示登陆成功，小于 0 表示失败
	备注	
224	API 函数	int IMC100_UserLogout(int comId=0)

	说明	当前 API 网络客户端设备退出当前登陆模式，恢复为默认的客户模式
	参数	
	应答指令解释	返回 0 表示退出成功，小于 0 表示失败
	备注	
225	API 函数	int IMC100_Set_SysTime(char time[16], int comId=0)
	说明	设置当前系统时钟
	参数	time：时间字符串（年月日时分秒），长度为 14
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
226	API 函数	int IMC100_LatchEnable(int cmd, int comId=0)
	说明	锁存功能开启控制
	参数	cmd：控制命令，1-开启，0-关闭
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
227	API 函数	int IMC100_Get_LatchSts(int *sts, int comId=0)
	说明	查询锁存功能开启状态
	参数	sts：锁存功能状态，1-开启，0-关闭
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
228	API 函数	int IMC100_Get_LatchSum(int *sum, int comId=0)
	说明	查询锁存位置点的总数
	参数	sum：查询结果
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	管理模式及以上使用
229	API 函数	int IMC100_Get_LatchRobP(int index, int *sts, ROB_POS *pos, int comId=0)
	说明	读取对应锁存点的位置参数(按顺序读取，每个位置只能读到一次)
	参数	index：预留参数{0000}sts：返回状态，表示有无锁存数据（0-无，1-有）{0000}pos：返回锁存值，表示当前工具相对当前工件的值
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
230	API 函数	int IMC100_Clr_LatchPos(int comId=0)
	说明	清除锁存位置
	参数	
	应答指令解释	返回 0 表示命令完成，小于 0 表示失败
	备注	
231	API 函数	int IMC100_Set_CollModeAndAction(int checkflag, int action, int comId=0)
	说明	设置数据流模式下碰撞检测功能开关以及碰撞报警后触发的动作
	参数	checkflag：数据流模式下碰撞检测功能开关：0-关闭碰撞检测，1-打开碰撞检测{0000}action：碰撞报警后触发的动作：{0000}0-不对触发动作进行定义{0000}3-碰撞停机{0000}7-碰撞回退{0000}{0000}当选择"0-不对触发动作进行定义"时，实际触发动作，用以下设置为准：

		{0000}(1)API：IMC100_Set_TeachModeCollAction/ IMC100_Set_PlaybackModeCollAction{0000}(2)InoTeachPad：【设置】-【运动参数】-【碰撞检测设置{0000}(3)InRobotLab：【控制器参数设置】-【运动参数】-【碰撞检测】
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	只对数据流模式有效
232	API 函数	int IMC100_Get_CollModeAndAction(int *checkflag, int *action, int comId=0)
	说明	读取数据流模式或工程指令的碰撞检测功能开关以及碰撞报警后触发的动作
	参数	checkflag：碰撞检测功能开关{0000}action：碰撞报警后触发的动作
	应答指令解释	返回 0 表示命令完成，小于 0 表示失败
	备注	
233	API 函数	int IMC100_Set_AxisCollMode(int axisNo, int checkflag, int comId=0)
	说明	设置数据流模式下单轴的碰撞检测开关
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	只对数据流模式有效
234	API 函数	int IMC100_Get_AxisCollMode(int axisNo, int *checkflag, int comId=0)
	说明	读取数据流模式或工程指令的单轴碰撞检测开关
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
235	API 函数	int IMC100_Set_AxisCollLevel(int axisNo, double level, int comId=0)
	说明	设置数据流模式下单轴的碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度，范围：25~300
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	只对数据流模式有效
236	API 函数	int IMC100_Get_AxisCollLevel(int axisNo, double *level, int comId=0)
	说明	读取数据流模式或工程指令的单轴碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
237	API 函数	int IMC100_Set_TeachModeAxisCollMode(int axisNo, int checkflag, int comId=0)
	说明	设置手动模式的单轴碰撞检测开关

	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
238	API 函数	int IMC100_Get_TeachModeAxisCollMode(int axisNo, int *checkflag, int comId=0)
	说明	读取手动模式的单轴碰撞检测开关
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
239	API 函数	int IMC100_Set_TeachModeCollAction(int action, int comId=0)
	说明	设置手动模式的碰撞报警后触发的动作
	参数	action：碰撞报警后触发的动作：3-碰撞停机，7-碰撞回退
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
240	API 函数	int IMC100_Get_TeachModeCollAction(int *action, int comId=0)
	说明	读取手动模式的碰撞报警后触发的动作
	参数	action：碰撞报警后触发的动作
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
241	API 函数	int IMC100_Set_TeachModeAxisCollLevel(int axisNo, double level, int comId=0)
	说明	设置手动模式的碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度，范围：25~300
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
242	API 函数	int IMC100_Get_TeachModeAxisCollLevel(int axisNo, double *level, int comId=0)
	说明	读取手动模式的碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
243	API 函数	int IMC100_Set_PlayBackModeAxisCollMode(int axisNo, int checkflag, int comId=0)
	说明	设置自动模式的单轴碰撞检测开关
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
244	API 函数	int IMC100_Get_PlayBackModeAxisCollMode(int axisNo, int *checkflag, int comId=0)
	说明	读取自动模式的单轴碰撞检测开关

	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}checkflag：碰撞检测开关
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
245	API 函数	int IMC100_Set_PlayBackModeCollAction(int action, int comId=0)
	说明	设置自动模式的碰撞报警后触发的动作
	参数	action：碰撞报警后触发的动作:3-碰撞停机，7-碰撞回退
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
246	API 函数	int IMC100_Get_PlayBackModeCollAction(int *action, int comId=0)
	说明	读取自动模式的碰撞报警后触发的动作
	参数	action：碰撞报警后触发的动作
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
247	API 函数	int IMC100_Set_PlayBackModeAxisCollLevel(int axisNo, double level, int comId=0)
	说明	设置自动模式的碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度，范围：25~300
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
248	API 函数	int IMC100_Get_PlayBackModeAxisCollLevel(int axisNo, double *level, int comId=0)
	说明	读取自动模式的碰撞检测灵敏度
	参数	axisNo：轴号，与轴数有关，范围：1~6{0000}level：轴碰撞检测灵敏度
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
249	API 函数	int IMC100_Get_RobotAxisNum(int *axisNum, int comId=0)
	说明	查询机器人本体轴数
	参数	axisNum：机器人本体轴数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
250	API 函数	int IMC100_Set_BindTcpSpeedValue (int num, double minValue, double maxValue, double minSpeed, double maxSpeed, int comId = 0)
	说明	将线速度模拟量输出绑定到 DA[num]
	参数	num:绑定的 DA 的 index，范围（0~激活的 DA 数量） minValue:最小输出模拟量，范围（选定的 DA 配置的输出范围） maxValue:最大输出模拟量，范围（选定的 DA 配置的输出范围） minSpeed:最小速度，范围（0~maxSpeed） maxSpeed:最大速度，范围（0~100000）
	应答指令解释	返回 0 表示设置成功,小于 0 表示失败
	备注	
251	API 函数	int IMC100_Get_BindTcpSpeedValue(int num, int *Status , double

		*minValue, double *maxValue, double *minSpeed, double *maxSpeed, int comId = 0)
	说明	得到输入的 DA index 的设置数据
	参数	num:绑定的 DA 的 index，范围（0~激活的 DA 数量） *Status：DA 的绑定状态，范围 0 或 1 *minValue:最小输出模拟量，范围（选定的 DA 配置的输出范围） *maxValue:最大输出模拟量，范围（选定的 DA 配置的输出范围） *minSpeed:最小速度，范围（0~maxSpeed） *maxSpeed:最大速度，范围（0~100000）
	应答指令解释	返回 0 表示设置成功,小于 0 表示失败
	备注	
252	API 函数	int IMC100_Close_BindTcpSpeed(int num, int comId = 0)
	说明	解除线速度模拟量 DA[num]的绑定
	参数	num:解除绑定的 DA 的 num 范围（0~激活的 DA 数量）
	应答指令解释	返回 0 表示设置成功,小于 0 表示失败
	备注	
253	API 函数	int IMC100_Get_TcpSpeedDAOutAndVel(int num, int *Status, double *DAout, double *Velocity, int comId)
	说明	得到所绑定的那个 DA 输出的数据与线速度值
	参数	num:绑定的 DA 的 index，范围（0~激活的 DA 数量） *Status：DA 的绑定状态, 输出范围 0 或 1 *DAout:DA 实时输出值 *Velocity:线速度实时输出值
	应答指令解释	返回 0 表示设置成功,小于 0 表示失败
	备注	
254	API 函数	int IMC100_Set_TraceRecoverMode(int TraceMode, int comId = 0)
	说明	设置轨迹恢复阈值设置模式
	参数	TraceMode:轨迹恢复阈值设置模式，范围 0：关节模式，1：位置模式
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
255	API 函数	int IMC100_Get_TraceRecoverMode(int *TraceMode , int comId = 0)
	说明	获取轨迹恢复阈值设置模式
	参数	TraceMode:轨迹恢复阈值设置模式，范围 0：关节模式，1：位置模式
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
256	API 函数	int IMC100_Set_VarTraceRecoverParams(int Mode, double TCPDis, double TCPRot, double ExternalDis ,double ExternalRot, int comId = 0)
	说明	设置自动或手动下，轨迹恢复阈值参数
	参数	Mode: 轨迹恢复手动或自动，范围 1：手动，2：自动 TCPDis：TCP 距离,范围 0~20000 TCPRot：TCP 旋转，范围 0~360 ExternalDis：外部轴距离，范围 0~20000

		ExternalRot：外部轴旋转，范围 0~360
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
257	API 函数	int IMC100_Get_VarTraceRecoverParams(int Mode, double *TCPDistance, double *TCPRotation, double *ExternalDistance, double *ExternalRotatioon ,int comId = 0)
	说明	获得自动或手动下，轨迹恢复阈值参数
	参数	Mode: 轨迹恢复手动或自动，范围 1：手动，2：自动 TCPDistance：TCP 距离,范围 0~20000 TCPRotation：TCP 旋转，范围 0~360 ExternalDistance：外部轴距离，范围 0~20000 ExternalRotatioon：外部轴旋转，范围 0~360
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
258	API 函数	int IMC100_Set_InterferZoneActStat(int iNum, int iStatus, int comId);
	说明	设置干涉区激活状态
	参数	Index:干涉区序号，范围 0-15 iStatus 干涉区状态
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
259	API 函数	int IMC100_Get_InterferZoneActStat(int iStatus[16], int comId);
	说明	获取干涉区激活状态
	参数	iStatus 干涉区状态
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
260	API 函数	int IMC100_Set_InterferZonePara(int iNum, S_INTERFER_ZONE* pstPara, int comId);
	说明	根据序号，设置干涉区参数
	参数	Index:干涉区序号，范围 0-15 * pstPara: 干涉区参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
261	API 函数	int IMC100_Get_InterferZonePara(int iNum, S_INTERFER_ZONE* pstPara, int comId);
	说明	根据序号，获取某干涉区参数
	参数	Index:干涉区序号，范围 0-15 pstPara: 干涉区参数
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
262	API 函数	int IMC100_Set_InterferToolActNum(int iNum, int comId);
	说明	根据序号，激活监控对象
	参数	iNum:监控对象序号，范围 0-15

	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
263	API 函数	int IMC100_Get_InterferToolActNum(int* iNum, int comId);
	说明	根据序号，获取激活的监控对象
	参数	iNum:监控对象序号，范围 0-15
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
264	API 函数	int IMC100_Set_InterferToolPara(int iNum, S_INTERFER_TOOL* pstPara, int comId);
	说明	根据序号，设置监控对象
	参数	iNum:监控对象序号，范围 0-15 pstPara 监控对象参数
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
265	API 函数	int IMC100_Get_InterferToolPara(int iNum, S_INTERFER_TOOL* pstPara, int comId);
	说明	根据序号，获取监控对象参数
	参数	iNum:监控对象序号，范围 0-15 pstPara 监控对象参数
	应答指令解释	返回 0 表示查询成功，小于 0 表示失败
	备注	
266	API 函数	int IMC100_Set_MemRobP(int pNum, ROB_POS *pos, int comId=0)
	说明	更新内存中全局位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	
267	API 函数	int IMC100_Set_MemRobJP(int pNum, ROB_JPOS *pos, int comId=0)
	说明	更新内存中全局关节位置变量对应的位置参数
	参数	pNum：全局位置变量序号，范围 0-9999{0000}pos：关节位置参数结构体
	应答指令解释	返回 0 表示设置成功，小于 0 表示失败
	备注	

附录 2：API 字符串表

API 字符串应用说明：

1、“应答指令格式”中“e*”对应的错误内容详见相关故障说明

2、 指令请求格式为：@@+指令+$$

3、 请求指令解释栏中，Para*：指设置的参数，以空格区分索引号

注意：单个参数字符串长度请勿超过 128

序号	说明	
1	请求指令格式	调用通用通讯接口和机器人控制器建立通讯，上位机为客户端 API 固定端口号：2222
	请求指令解释	/
	应答指令格式	/
	应答指令解释	/
	备注	1）控制器最多支持 4 个不同的连接 2）连接编号作用域：上位机同一个进程 *上位机 IP、端口号和控制器 IP、端口号，任意一个不同即为不同的连接
2	请求指令格式	调用通用通讯接口和机器人控制器关闭通讯
	请求指令解释	/
	应答指令格式	/
	应答指令解释	/
	备注	
3	请求指令格式	(1)EStop ON (2)EStop OFF
	请求指令解释	(1)按下急停开关 (2)松开急停开关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误，*为具体错误码，下同
	备注	
4	请求指令格式	(1)Motor ON (2)Motor OFF
	请求指令解释	(1)电机使能 (2)电机掉使能
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	上使能命令发出后，300ms 后读取使能状态
5	请求指令格式	ResetErr

	请求指令解释	复位控制器故障
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	延时指令，约 50ms
6	请求指令格式	Set_Mode 2
	请求指令解释	设置系统运行模式 Para1：模式，1-手动，2-自动
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
7	请求指令格式	(1)Prg Start (2)Prg Stop
	请求指令解释	(1)工程启动运行 (2)工程停止运行
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
8	请求指令格式	BackStartLine
	请求指令解释	工程返回启动行
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
9	请求指令格式	Set_Vel 100
	请求指令解释	设置当前运行全局速度 Para1：全局速度，范围 1-100
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
10	请求指令格式	Set_AccRamp 50.0 50.0
	请求指令解释	设置数据流模式运动段的加加速度 Para1：起始段加加速度百分比，范围 10.0-100. Para2：结束段加速度百分比，范围 10.0-100.

	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
11	请求指令格式	Set_RapidMove 0 0
	请求指令解释	设置最优轨迹规划 Para1：运动类型，0-cp 运动，1-ptp 运动 Para2：最优规划开关，0-关闭最优规划，1-开启最优规划
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
12	请求指令格式	Set_Flymode 0 0
	请求指令解释	设置运动指令过渡模式（暂时设置无效） Para1：运动类型，0-cp 运动 Para2：过渡模式，0-自由过渡，1-固定路径过渡
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
13	请求指令格式	Set_Flypress 50 50
	请求指令解释	设置固定路径的过渡应力（暂时设置无效） Para1：位置过渡应力，范围 50-200 Para2：姿态过渡应力，范围 50-200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
14	请求指令格式	(1)Dsmode ON (2)Dsmode OFF (3)Dsmode PAUSE (4)Dsmode CONTINUE
	请求指令解释	(1)数据流模式开启 (2)数据流模式关闭 (3)数据流模式暂停 (3)数据流模式继续
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误

	备注	
15	请求指令格式	Set_DO 0 1
	请求指令解释	设置 DO0 为 on 状态 Para1：RC 可控制 DO 序号 Para2：DO 状态，1 表示 on，0 表示 off
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
16	请求指令格式	Set_Slewmode 1
	请求指令解释	设置 Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式 Para1：旋转优化方式 0-不采用优化，以位置变量的臂参数为准； 1-采用优化模式 1，保证运动中 J4/J6 中-180°~180°范围内； 2-保证 J4/J6 以最近的方式运动，机器人将计算运动到目标点是否经过 J4/J6 旋转 180°，若角度差≤180°，则完全运动到目标点，若＞180°，则 J4/J6 以相反的方向运动，最终运动到距目标点的 J4/J6 相差一圈（360°）的位置；期间，J4/J6 反向运动若超过机器人极限范围，则停止运动并报警； 3-保证 J4/J6 以最近的方式运动，与模式 2 的差别在于 J4/J6 反向运动若超出极限范围，不会报警，而是不进行反向运动，完全采取原方式正常运动。
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效，数据流模式关闭后清零
17	请求指令格式	Set_DOGroup 1 255
	请求指令解释	设置一组 DO 状态，组号为 1，个数为 8 Para1：RC 可控制 DO 组号，最大范围 0-7，依据实际配置情况 Para2：每个 DO 的状态，1 表 on，0 表 off
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
18	请求指令格式	Set_DA 0 0.000
	请求指令解释	按序号设置 DA 输出值 Para1：RC 可控制 DA 序号 Para2：DA 值，电流型最大范围 0-20mA，电压型最大范围-10V 到 10V，具体依据 DA 通道类型

	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
19	请求指令格式	(1)InchMode ON (2)InchMode OFF
	请求指令解释	(1)打开寸动模式 (2)关闭寸动模式
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
20	请求指令格式	Set_InchStep 1
	请求指令解释	设置寸动运行的步长等级 Para1：步长等级值，范围 0-4 0-退出寸动模式 1-寸动等级 G1，步长为 0.05, 2-寸动等级 G2，步长为 0.3 3-寸动等级 G3，示步长为 2 4-自定义寸动步长，步长为寸动参数设置值 关节坐标寸动时单位为度，其他坐标寸动时单位为 mm
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
21	请求指令格式	AxisJog 1 1
	请求指令解释	单轴点动运动命令 Para1：轴序号：当前激活机械单元为 Robot（机器人）时，范围 1-6，关节坐标时分别对应 J1-J6 轴，非关节坐标时分别对应 X/Y/Z/RZ/RY/RX；当前激活单元为外部机械时，Axis 范围为 1-外部机械单元总轴数； Para2：单轴点动运动命令，0-停止，1-正向运动，-1-反向运动
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	手动模式下数据流模式关闭后有效
22	请求指令格式	AxisInch 1 1

	请求指令解释	寸动运动命令 Para1：轴序号 当前激活机械单元为 Robot（机器人）时，范围 1-6，关节坐标时分别对应 J1-J6 轴，非关节坐标时分别对应 X/Y/Z/RZ/RY/RX mecUnit 为外部机械时，Axis 范围为 1-外部机械单元总轴数 Para2：寸动命令，0-停止，1-正向寸动，-1-反向寸动
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	手动模式下数据流模式关闭后有效
23	请求指令格式	PoseAlign 4 1
	请求指令解释	姿态校准(适用于六关节机器人） Para1：坐标系，-1-法兰坐标系，1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系（当前仅支持法兰坐标系（当前激活工具为外部工具）、世界坐标系、基坐标系、工件坐标系） Para2：姿态校准命令，0-停止姿态校准，1-开启姿态校准
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	姿态校准过程中请勿进行其他运动，且进行姿态校准后，数据流模式将自动关闭
24	请求指令格式	Set_ActiveMechUnit Robot
	请求指令解释	设置手动模式运动的激活机械单元 Para1：机械单元名称（大小写不敏感）
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
25	请求指令格式	Get_ActiveMechUnit
	请求指令解释	获取手动模式运动的激活机械单元
	应答指令格式	=Robot e*:
	应答指令解释	返回手动模式运动的激活机械单元名称； 指令返回错误
	备注	
26	请求指令格式	(1)Set_TeachCoordinate ON (2)Set_TeachCoordinate OFF
	请求指令解释	(1)打开协调开关 (2)关闭协调开关

	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	设置协调开关开启条件： （1）当前激活机械单元为导轨，当前坐标系为世界或工件坐标系； （2）当前激活机械单元为变位机，当前坐标系为工件坐标系且工件关联机械单元名称为当前激活的机械单元（即变位机）；
27	请求指令格式	Get_TeachCoordinate
	请求指令解释	获取手动模式运动协调开关
	应答指令格式	=0 e*:
	应答指令解释	返回手动模式运动协调开关：0-协调开关关闭，1-协调开关开启； 指令返回错误
	备注	
28	请求指令格式	Home 0
	请求指令解释	回到工作原点 0 Para1：原点序号，范围 0-4
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
29	请求指令格式	Set_DynamicBrake 1
	请求指令解释	设置动态制动开关 Para1：0-关闭动态制动开关，1-开启动态制动开关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
30	请求指令格式	Get_DynamicBrake
	请求指令解释	获取的当前动态制动开关状态
	应答指令格式	=0 e*:
	应答指令解释	返回当前动态制动开关状态：0-动态制动开关关闭，1-动态制动开关打开； 指令返回错误
	备注	
31	请求指令格式	MovJGP GP0 100 0

	请求指令解释	关节插补方式移动到 GP0 点,速度 100 和插补精度 0 Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
32	请求指令格式	MovLGP GP0 100 0
	请求指令解释	直线插补方式移动到 GP0 点,速度 100 和插补精度 0 和插补精度 0 Para1：目标全局位置点序号，范围 0-9999） Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
33	请求指令格式	MovCGP GP0 GP1 100 0
	请求指令解释	圆弧插补方式运动，GP0 为圆弧中点，GP1 为圆弧终点,速度 100 和插补精度 0 Para1：圆弧中点的全局位置点序号，范围 0-9999 Para2：：圆弧终点的全局位置点序号，范围 0-9999 Para3：运动速度，范围 1-100 Para4：插补精度，范围-2 至 200 & 1000 至 1200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
34	请求指令格式	MovJGP GP0 100 0 [7,1,0,0.05]
	请求指令解释	关节插补方式移动到设定点,其中速度 100 和插补精度 0，且在开始运动后 0.05s 输出 DO7 为 ON Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:

	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
35	请求指令格式	MovLGP GP0 100 0 [7,1,2,5]
	请求指令解释	直线插补方式移动到 GP0 点,其中速度 100 和插补精度 0,且在开始运动后 5mm 时输出 DO7 为 ON Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
36	请求指令格式	MovCGP GP0 GP1 100 0 [7,1,2,5]
	请求指令解释	圆弧插补方式移动到 GP0 点,其中速度 100 和插补精度 0,且在开始运动后 5mm 时输出 DO7 为 ON Para1：圆弧中点全局位置点序号，范围 0-9999 Para2：圆弧终点全局位置点序号，范围 0-9999 Para3：运动速度，范围 1-100 Para4：插补精度，范围-2 至 200 & 1000 至 1200 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
37	请求指令格式	MovJRobP [370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] 0,50,0,0.000,0.000,2.000,1.000 1 1 1,0,1,1.000000;

	请求指令解释	关节插补方式运动到指定值的位置点，同时控制对应 IO Para1：目标位置点信息 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	1、仅在数据流模式下有效 2、不支持数值型速度类型
38	请求指令格式	MovLRobP [370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] 0,50,0,0.000,0.000,2.000,1.000 1 1 1,0,1,1.000000;
	请求指令解释	直线插补方式运动到指定值的位置点，同时控制对应 IO Para1：圆弧中点位置点信息 Para2：圆弧终点位置点信息 Para3：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para4：插补精度，范围-2 至 200 & 1000 至 1200 Para5：运动 IO 组数，范围 0-3 Para6：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效

39	请求指令格式	MovCRobP [370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] [375.263,44.809,769.512,-118.306,-68.009,-47.077; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] 0,50,0,0.000,0.000,2.000,1.000 1 1 1,0,1,1.000000;
	请求指令解释	圆弧插补方式运动到指定值的位置点，同时控制对应 IO Para1：目标位置点信息 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
40	请求指令格式	JumpRobP [370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] 0,50,0,0.000,0.000,2.000,1.000 1 1 1,0,1,1.000000;
	请求指令解释	跳跃插补方式运动到指定值的位置点，同时控制对应 IO Para1：目标位置点信息 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:

	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
41	请求指令格式	JumpLRobP [370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000] 0,50,0,0.000,0.000,2.000,1.000 1 1 1,0,1,1.000000;
	请求指令解释	直线跳跃插补方式运动到指定值的位置点，同时控制对应 IO Para1：目标位置点信息 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
42	请求指令格式	JumpGP GP0 100 0
	请求指令解释	跳跃插补方式移动到 GP0 点,速度 100 和插补精度 0 Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
43	请求指令格式	JumpLGP GP0 100 0
	请求指令解释	直线跳跃插补方式移动到 GP0 点,速度 100 和插补精度 0 Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误

	备注	仅在数据流模式下有效
44	请求指令格式	JumpGP GP0 100 0 [7,1,0,0.05]
	请求指令解释	跳跃插补方式移动到 GP0 点,其中速度 100 和插补精度 0,且在开始运动后 0.05s 输出 D07 为 ON Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
45	请求指令格式	JumpLGP GP0 100 0 [7,1,0,0.05]
	请求指令解释	直线跳跃插补方式移动到 GP0 点,其中速度 100 和插补精度 0,,且在开始运动后 5mm 时输出 D07 为 ON Para1：目标全局位置点序号，范围 0-9999 Para2：运动速度，范围 1-100 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	仅在数据流模式下有效
46	请求指令格式	MovJAbsRobJP [0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000] 0,50,0,0.000,0.000,2.000,1.000 0 1 1,0,1,1.000000

	请求指令解释	关节插补方式运动到指定值的位置点，同时控制对应 IO Para1：目标关节点信息 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	1、仅在数据流模式下有效 2、不支持数值型速度类型
47	请求指令格式	MovJAbsRobIndexJP GJP1 0,50,0,0.000,0.000,2.000,1.000 0 1 1,0,1,1.000000
	请求指令解释	关节插补方式运动到指定序号的全局关节点，同时控制对应 IO Para1：目标全局关节点序号，范围 0-9999 Para2：速度参数，格式[速度类型,百分比速度,数值速度是否受全局百分比速度影响属性，tcp 速度，tcp 姿态速度，线型外部轴速度，旋转外部轴速度] 速度类型：0-百分比速度类型，1-数值型速度类型 百分比速度：范围 1-100 数值速度是否受全局百分比速度影响属性：1-Speed 速度不受全局百分比速度影响；0-Speed 收全局百分比速度影响 Para3：插补精度，范围-2 至 200 & 1000 至 1200 Para4：运动 IO 组数，范围 0-3 Para5：运动 IO 控制信息，支持 1~3 组，格式：[IO 序号 1,IO 输出值 1,IO 类型 1,IO 参数 1; IO 序号 2,IO 输出值 2,IO 类型 2,IO 参数 2; ...]
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	1、仅在数据流模式下有效 2、不支持数值型速度类型
48	请求指令格式	IndCMove 6 20.0 50.0 50.0 Robot

	请求指令解释	独立轴连续运动 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：数值型速度，正负代表方向且指定的速度受全局速度百分比的影响，其计算公式为：运行速度 = 指令设置的速度*全局速度百分比，单位：°/s（线性轴：mm/s） 范围：-10000°/s~10000°/s（Speed = 0 代表独立轴连续运动停止） Para3：加速度，范围：20-100 Para4：减速度，范围：20-100 Para5：机械单元名。如果是机器人，则是 Robot
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	1、仅在数据流模式下有效； 2、请在除独立轴以外所有轴运动停止时调用该接口； 3、当 Speed 大于控制器速度上限时，以控制器速度上限为准。
49	请求指令格式	IndSpeed 6 Robot
	请求指令解释	检测独立轴是否达到设定速度 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：机械单元名。如果是机器人，则是 Robot
	应答指令格式	=1 e*:
	应答指令解释	返回是否达到设定速度，1-达到指定速度，0-未达到指定速度 指令返回错误
	备注	
50	请求指令格式	IndResetOld 6 Robot
	请求指令解释	重置独立轴，将独立轴切换为正常模式,轴的圈数保留 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：机械单元名。如果是机器人，则是 Robot
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误

	备注	1、仅在数据流模式下有效； 2、请在除独立轴以外所有轴运动停止时调用该接口。
51	请求指令格式	IndReset 6 100.0 0 Robot
	请求指令解释	重置独立轴，将独立轴切换为正常模式,轴的圈数根据 RefPos 与 Direction 调整 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：参考位置，范围：-720°~720° Para3： -1-调整轴的多圈数，以便参考位置 refNum 位于更改后的当前位置的反向侧。（反向 360 度以内） 0-调整轴的多圈数，以便更改后的当前位置尽可能地接近指定的 refNum 位置。（±180 度以内） 1-调整轴的多圈数，以便参考位置 refNum 位于更改后的当前位置的正向侧。（360 度以内） Para4：机械单元名。如果是机器人，则是 Robot
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	1、仅在数据流模式下有效； 2、请在除独立轴以外所有轴运动停止时调用该接口。
52	请求指令格式	IMC100_Get_IndCMoveSts 6 Robot
	请求指令解释	查询独立轴连续运动切换状态 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：机械单元名。如果是机器人，则是 Robot
	应答指令格式	=1 e*:
	应答指令解释	返回独立轴连续运动是否切换完成，0-独立轴连续运动切换未完成，1-独立轴连续运动切换完成 指令返回错误
	备注	
53	请求指令格式	Get_IndResetSts 6 Robot
	请求指令解释	查询独立轴重置状态 Para1：轴号 如果是六轴机器人，仅支持第 6 轴（四轴机器人，仅支持非 Robot 机械单元独立轴） 非 Robot 机械单元，范围：1-机械单元配置的最大轴数 Para2：机械单元名。如果是机器人，则是 Robot

	应答指令格式	=1 e*:
	应答指令解释	返回独立轴重置是否完成，0-独立轴重置未完成，1-独立轴重置完成 指令返回错误
	备注	
54	请求指令格式	Get_MotionStsExceptIndAxis
	请求指令解释	查询除独立轴以外的机器人运动状态
	应答指令格式	=1 e*:
	应答指令解释	返回系统运动状态，0-停止/运动完成，1-运动中 指令返回错误
	备注	
55	请求指令格式	Get_RobPHere
	请求指令解释	获取当前位置 Ret:当前工具 TCP 相对当前工件的值。如果用户需得到不同工具或不同工件下的值，需提前使用 Set_ToolCNum Set_WobjNum
	应答指令格式	=425.011,2.741,718.553,-73.956,-88.026,-105.676; 0,0,0,1; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回当前位置值 指令返回错误
	备注	
56	请求指令格式	Get_RobJPHere
	请求指令解释	获取当前关节位置
	应答指令格式	=0.368,0.075,4.656,0.101,-5.205,1.800,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回当前关节位置值 指令返回错误
	备注	
57	请求指令格式	Set_ActiveMechUnitPosFormat 1
	请求指令解释	设置激活的机械单元的位置显示格式 Para1：位置格式，1-关节坐标系格式，2-基坐标系格式，3-法兰坐标系格式，4-工件坐标系格式，5-世界坐标系格式
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
58	请求指令格式	Get_ActiveMechUnitPos

	请求指令解释	获取激活的机械单元的位置 Ret：格式：当前激活机械单元的位置格式，机械单元位置 当前激活机械单元的位置格：1-关节坐标系格式，2-基坐标系格式，3-法兰坐标系格式，4-工件坐标系格式，5-世界坐标系格式 机械单元位置：数组长度 8
	应答指令格式	=1,0.368,0.075,4.656,0.101,-5.205,1.800,0.000,0.000 e*:
	应答指令解释	返回当前位置格式及位置值 格式：当前激活机械单元的位置格式（1-关节坐标系格式，2-基坐标系格式，3-法兰坐标系格式，4-工件坐标系格式，5-世界坐标系格式），机械单元位置（数组长度 8） 指令返回错误
	备注	
59	请求指令格式	Get_PosHerePulse
	请求指令解释	查询当前位置点的脉冲值
	应答指令格式	=87421.0,17473.0,961184.0,17651.0,-758105.0,207982.0 e*:
	应答指令解释	返回当前位置点的脉冲值 指令返回错误
	备注	
60	请求指令格式	Get_RobJToRobP 0.368,0.075,4.656,0.101,-5.205,1.800,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 0,0,0
	请求指令解释	根据关节点计算位置点 Para1：待转换的关节点 Para2：工具号，工件号，负载号
	应答指令格式	=541.960956,3.493287,898.202116,-73.329554,-88.019769,-106.302493; 0,0,0,1; 0.000000,0.000000,0.000000,0.000000,0.000000,0.000000 e*:
	应答指令解释	返回转换后的位置点位 指令返回错误
	备注	
61	请求指令格式	Get_RobPToRobJ 541.960,3.493,898.202,-73.329,-88.019,-106.302; 0,0,0,1; 0.000,0.000,0.000,0.000,0.000,0.000 0,0,0
	请求指令解释	根据位置点计算关节点 Para1：待转换的位置点 Para2：工具号，工件号，负载号
	应答指令格式	=0.368,0.075,4.656,0.115,-5.205,1.787,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 e*:

	应答指令解释	返回转换后的关节点位 指令返回错误
	备注	
62	请求指令格式	Get_FixWobjRobP 329.271,146.581,715.472,-16.876,-69.828,-152.263; 0,-1,0,1; 0.000,0.000,0.000,0.000,0.000,0.000 2
	请求指令解释	计算世界坐标系下的某一个的位置在指定的非夹持的固定工件(RobHold=0, UFFix=1)坐标系下的位置 Para1：待转换的机器人点位(世界坐标系下的点） Para2：工件号
	应答指令格式	=329.271,146.581,715.472,-16.876,-69.828,-152.263; 0,-1,0,1; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回转换后的工件坐标系下的点位 指令返回错误
	备注	
63	请求指令格式	Get_RobHoldWobjRobP 370.763,49.600,765.861,-114.829,-76.872,-56.185; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 1 -765.569,204.516,-117.829,26.024,-68.456,146.217; 0,-1,0,1; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	计算世界坐标系下的某一个的位置在指定的夹持工件(RobHold=1, UFFix=1)坐标系下的位置 Para1：待转换的机器人点位(世界坐标系下的点） Para2：工件号 Para3：法兰在世界坐标系下的坐标值
	应答指令格式	=1171.964,217.554,821.784,25.938,20.712,23.176; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 e*:
	应答指令解释	返回转换后的工件坐标系下的点位 指令返回错误
	备注	
64	请求指令格式	Get_OffsetJ_RobJP 10.000,10.000,10.000,10.000,10.000,10.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 5.000,5.000,5.000,5.000,5.000,5.000
	请求指令解释	查询关节点偏移后的点位 Para1：原始关节点 Para2：平移变量
	应答指令格式	=15.000,15.000,15.000,15.000,15.000,15.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回偏移后点位结果 指令返回错误

	备注	
65	请求指令格式	Get_Offset_RobP 370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 5.000,5.000,5.000,5.000,5.000,5.000
	请求指令解释	查询工件坐标系下偏移后的点位 Para1：原始点 Para2：平移变量
	应答指令格式	=375.000,55.000,770.000,-89.834,-71.398,-76.263; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 e*:
	应答指令解释	返回偏移后点位结果 指令返回错误
	备注	
66	请求指令格式	Get_OffsetT_RobP 370.000,50.000,765.000,-115.000,-75.000,-55.000; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 5.000,5.000,5.000,5.000,5.000,5.000
	请求指令解释	查询工具坐标系下偏移后的点位 Para1：原始点 Para2：平移变量
	应答指令格式	=375.263,44.809,769.512,-118.306,-68.009,-47.077; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 e*:
	应答指令解释	返回偏移后点位结果 指令返回错误
	备注	
67	请求指令格式	Get_SysErrSts
	请求指令解释	查询系统当前故障状态
	应答指令格式	=1 e*:
	应答指令解释	返回故障状态：bit0-系统有报警，bit1-系统有警告； 指令返回错误
	备注	
68	请求指令格式	Get_SysErr
	请求指令解释	查询系统故障码
	应答指令格式	=100dh e*:
	应答指令解释	返回故障值，十六进制； 指令返回错误
	备注	
69	请求指令格式	Get_TaskPrgPath 0

	请求指令解释	查询任务通道程序的路径 Para1：任务通道，0 为主任务
	应答指令格式	=TeachProgram/newprj/main.pro e*:
	应答指令解释	返回指定任务通道路径； 指令返回错误
	备注	
70	请求指令格式	Get_TaskRunSts 0
	请求指令解释	查询任务通道的运行状态 Para1：任务通道，0 为主任务
	应答指令格式	=0 e*:
	应答指令解释	返回任务通道运行状态：0-停止，1-启动/继续，10-就绪， 100-任务未激活，-1-任务激活但未配置程序； 指令返回错误
	备注	
71	请求指令格式	Get_TaskProgramLine 0
	请求指令解释	查询指定任务通道的处理行号 Para1：任务通道，0 为主任务
	应答指令格式	=1 e*:
	应答指令解释	返回值，当前任务通道的处理行号 指令返回错误
	备注	
72	请求指令格式	Get_CurMotionLine
	请求指令解释	查询当前执行的运动指令行号
	应答指令格式	=2 e*:
	应答指令解释	返回值，当前执行的运动指令行号 指令返回错误
	备注	
73	请求指令格式	Get_InitSts
	请求指令解释	查询系统初始化状态
	应答指令格式	=11 e*:
	应答指令解释	返回系统初始化状态，范围-1-11,11 表示系统启动完成 指令返回错误
	备注	
74	请求指令格式	Get_AccRamp
	请求指令解释	查询数据流模式运动段的加加速度
	应答指令格式	=50.000000, 50.000000 e*:

	应答指令解释	返回询数据流模式运动段的加加速度，格式：起始段加加速度百分比（范围 10.0-100.0），结束段加加速度百分比（范围 10.0-100.0） 指令返回错误
	备注	
75	请求指令格式	Get_RapidMove 0
	请求指令解释	查询当前运动类型最优轨迹开关 Para1：运动类型，0-cp 运动,1-ptp 运动
	应答指令格式	=0 e*:
	应答指令解释	返回指定运动类型最优轨迹开关，0-当前运动类型最优规划关闭，1-当前运动最优轨迹开启； 指令返回错误
	备注	
76	请求指令格式	Get_FlyMode 0
	请求指令解释	查询运动指令过渡模式 Para1：运动类型，0-cp 运动
	应答指令格式	=0 e*:
	应答指令解释	返回指定运动类型过渡模式：0-自由过渡，1-固定路径过渡； 指令返回错误
	备注	
77	请求指令格式	Get_FlyPress
	请求指令解释	查询固定路径的过渡应力
	应答指令格式	=50, 50 e*:
	应答指令解释	返回运动指令过渡应力，格式：位置过渡应力（范围 50-200），姿态过渡应力（范围 50-200）； 指令返回错误
	备注	
78	请求指令格式	Get_CoordType
	请求指令解释	查询当前坐标系类型
	应答指令格式	=1 e*:
	应答指令解释	返回当前坐标系类型：1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系； 指令返回错误
	备注	
79	请求指令格式	Get_Vel
	请求指令解释	查询当前速度等级
	应答指令格式	=100 e*:

	应答指令解释	返回当前速度等级，范围 1-100； 指令返回错误
	备注	
80	请求指令格式	Get_Mode
	请求指令解释	查询系统当前运行模式
	应答指令格式	=1 e*:
	应答指令解释	返回系统当前运行模式：1-手动，2-自动，3-单步运行，5-连续运行； 指令返回错误
	备注	
81	请求指令格式	Get_DsMode
	请求指令解释	查询数据流模式是否开启
	应答指令格式	=0 e*:
	应答指令解释	返回数据流模式开启情况：0-OFF,1-ON/CONTINUE,2-PAUSE 指令返回错误
	备注	
82	请求指令格式	Get_InchMode
	请求指令解释	查询寸动模式
	应答指令格式	=0 e*:
	应答指令解释	返回寸动模式：0-非寸动，1-寸动 指令返回错误
	备注	
83	请求指令格式	Get_SlewMode
	请求指令解释	查询 Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式
	应答指令格式	=0 e*:
	应答指令解释	返回 Scara 机器人 J4 或标准 6 关节机器人的 J6 旋转优化方式： 0-不采用优化，以位置变量的臂参数为准； 1-采用优化模式 1，保证运动中 J4/J6 中-180°~180°范围内； 2-保证 J4/J6 以最近的方式运动，机器人将计算运动到目标点是否经过 J4/J6 旋转 180°，若角度差≤180°，则完全运动到目标点，若＞180°，则 J4/J6 以相反的方向运动，最终运动到距目标点的 J4/J6 相差一圈（360°）的位置；期间，J4/J6 反向运动若超过机器人极限范围，则停止运动并报警； 3-保证 J4/J6 以最近的方式运动，与模式 2 的差别在于 J4/J6 反向运动若超出极限范围，不会报警，而是不进行反向运动，完全采取原方式正常运动； 指令返回错误
	备注	

84	请求指令格式	Get_EStopSts
	请求指令解释	查询当前急停状态
	应答指令格式	=0 e*:
	应答指令解释	返回急停开关状态：0-急停松开，1-急停按下； 指令返回错误
	备注	
85	请求指令格式	Get_MotorSts
	请求指令解释	查询当前电机使能状态
	应答指令格式	=0 e*:
	应答指令解释	返回电机使能状态：0-未使能，1-使能； 指令返回错误
	备注	
86	请求指令格式	Get_MotionSts
	请求指令解释	查询当前机器人运动状态
	应答指令格式	=0 e*:
	应答指令解释	返回当前机器人运动状态：0-停止/运动完成，1-运动中，2-运动中断； 指令返回错误
	备注	
87	请求指令格式	Get_SysMode
	请求指令解释	查询系统当前模式
	应答指令格式	=0 e*:
	应答指令解释	返回系统当前模式：0-正常模式，>0-内部测试模式； 指令返回错误
	备注	
88	请求指令格式	Get_PrgRunTime
	请求指令解释	查询工程运行时间
	应答指令格式	=0 e*:
	应答指令解释	返回工程运行时间； 指令返回错误
	备注	
89	请求指令格式	Get_CurCmdNum
	请求指令解释	查询当前发送成功的运动指令的编号
	应答指令格式	=1 e*:
	应答指令解释	返回当前发送成功的运动指令的编号； 指令返回错误
	备注	仅在数据流模式下有效

90	请求指令格式	Get_CurCmdSts
	请求指令解释	查询当前运动指令实际完成状态（是否到位）
	应答指令格式	=0 e*:
	应答指令解释	返回当前发送成功的运动指令的编号：0-运动未完成，1-运动完成； 指令返回错误
	备注	仅在数据流模式下有效
91	请求指令格式	Get_CurCmdCacheNum
	请求指令解释	查询当前缓存区运动数据的数量
	应答指令格式	=0 e*:
	应答指令解释	返回当前缓存区运动数据的数量； 指令返回错误
	备注	
92	请求指令格式	Get_CmdSts 1
	请求指令解释	查询指定编号的运动指令实际完成状态 Para1：待查询的指令编号的运动指令
	应答指令格式	=0 e*:
	应答指令解释	返回指定编号的运动指令实际完成状态：0-运动未完成，1-运动完； 指令返回错误
	备注	仅在数据流模式下有效
93	请求指令格式	Get_DINum
	请求指令解释	查询系统当前 DI 的总数
	应答指令格式	=16 e*:
	应答指令解释	返回系统当前 DI 的总数（依据当前实际配置）； 指令返回错误
	备注	
94	请求指令格式	Get_DONum
	请求指令解释	查询系统当前 DO 的总数
	应答指令格式	=16 e*:
	应答指令解释	返回系统当前 DO 的总数（依据当前实际配置）； 指令返回错误
	备注	
95	请求指令格式	Get_ADNum
	请求指令解释	查询系统当前 AD 模拟输入端口的总数
	应答指令格式	=0 e*:

	应答指令解释	返回系统当前 AD 模拟输入端口的总数（依据当前实际配置）； 指令返回错误
	备注	
96	请求指令格式	Get_DANum
	请求指令解释	查询系统当前 DA 模拟输出端口的总数
	应答指令格式	=0 e*:
	应答指令解释	返回系统当前 DA 模拟输出端口的总数（依据当前实际配置）； 指令返回错误
	备注	
97	请求指令格式	Get_DI 0
	请求指令解释	查询 DIO 的状态 Para1：待查询的 DI 序号，范围：0~实际配置最大值
	应答指令格式	=0 e*:
	应答指令解释	返回 DI 状态：0-OFF，1-ON； 指令返回错误
	备注	
98	请求指令格式	Get_DIGroup 0
	请求指令解释	查询 DIO 组状态， Para1：待查询的 DI 组，每组个数为 8（参数最大范围 0-7，依据实际配置情况）
	应答指令格式	=0 e*:
	应答指令解释	返回 DI 组状态：0-255，对每个 bit 位，0-OFF,1-ON； 指令返回错误
	备注	
99	请求指令格式	Get_AD 0
	请求指令解释	查询 AD0 的输入值 Para1：待查询的 AD 序号，范围：0~实际配置最大值
	应答指令格式	=0 e*:
	应答指令解释	返回 AD 的输入值，电流型单位为 mA，电压型单位为 V； 指令返回错误
	备注	
100	请求指令格式	Get_DOCfg 0
	请求指令解释	查询 DO0 的配置权 Para1：待查询的 DO 序号，范围：0~实际配置最大值
	应答指令格式	=1 e*:

	应答指令解释	返回 DO 配置权：1-由 RC 控制，0-由 PLC 控制； 指令返回错误
	备注	
101	请求指令格式	Get_DOGroupCfg 0
	请求指令解释	查询 DO0 组的配置权 Para1：待查询的 DO 组，范围：0~实际配置最大值
	应答指令格式	=255 e*:
	应答指令解释	返回 DO 组状态：bit0-bit7 分别代表该组每个 DO 的配置权，1 表由 RC 控制，0 表由 PLC 控制； 指令返回错误
	备注	
102	请求指令格式	Get_DO 0
	请求指令解释	查询 DO0 的输出状态 Para1：待查询的 DO，范围：0~实际配置最大值
	应答指令格式	=0 e*:
	应答指令解释	返回 DO 的输出状态：0-OFF,1-ON; 指令返回错误
	备注	
103	请求指令格式	Get_DOGroup 0
	请求指令解释	查询一组 DO0 组状态，组号为 0 Para1：待查询的 DO，范围：0~实际配置最大值
	应答指令格式	=0 e*:
	应答指令解释	返回 DO 组输出状态，0-255，对每个 bit 位：0-OFF,1-ON; 指令返回错误
	备注	
104	请求指令格式	Get_DACfg 0
	请求指令解释	查询 DA 的配置权 Para1：待查询的 DA 序号，范围：0~实际配置最大值
	应答指令格式	=1 e*:
	应答指令解释	返回 DA 配置权：1-RC 控制，0-PLC 控制； 指令返回错误
	备注	
105	请求指令格式	Get_DA 0
	请求指令解释	查询 DA0 的输出值 Para1：待查询的 DA 序号，范围：0~实际配置的最大值
	应答指令格式	=0 e*:

	应答指令解释	返回 DA 输出值，电压型 DA 对应电压值，电流型 DA 对应电流值； 指令返回错误
	备注	
106	请求指令格式	Get_DevSts
	请求指令解释	查询系统设备的连接情况
	应答指令格式	=100105 e*:
	应答指令解释	sts[]：系统设备连接情况，代表查询的结果。其中： sts[0]：网卡 1 状态，0 表未连接，1 表连接，2 表被禁用 sts[1]：网卡 2 状态，同上 sts[2]：USB 设备状态，0 表未连接，1 表连接挂载成功，2 表挂载失败 sts[3]：SD 卡状态，0 表未连接，1 表连接挂载成功，2 表挂载失败，3 表文件系统格式错误 sts[4]：EtherCAT1 通信状态，0 表通信正常，1 表从站掉线，2 表未连接网线，3 表连接非 ECAT 设备，4 表已禁用 sts[5]：IRLink1 通信状态，0 表通讯正常，1 表从站掉线，2 表设备未挂载，3 表连接了非 IR-Link 设备，4 表被禁用，5 表设备未配置，6 表通讯异常
	备注	
107	请求指令格式	Get_FwVersion
	请求指令解释	查询系统控制器软件版本
	应答指令格式	=S03.22R e*:
	应答指令解释	返回系统软件版本号； 指令返回错误
	备注	
108	请求指令格式	Get_SysTime
	请求指令解释	查询系统当前时间
	应答指令格式	=20170101083000 e*:
	应答指令解释	返回系统时间（年月日时分秒）； 指令返回错误
	备注	
109	请求指令格式	Get_RobotType
	请求指令解释	查询系统当前机型
	应答指令格式	=IRS311-3-60TP5-S0_01740352 e*:
	应答指令解释	返回系统机型； 指令返回错误
	备注	
110	请求指令格式	Get_ArmType -30.0,-30.0,-30.0,-30.0,-30.0,-30.0

	请求指令解释	查询关节坐标值对应的臂参数
	应答指令格式	=-1,-1,-1,1 e*:
	应答指令解释	返回关节坐标值对应的臂参数； 指令返回错误
	备注	
111	请求指令格式	Get_ServoSts
	请求指令解释	查询伺服故障状态
	应答指令格式	=1000 e*:
	应答指令解释	返回各个伺服故障状态，从左至右依次为第 0 号伺服和系统最后一个伺服。（伺服总数包括机器人轴和外部轴两部分）； 指令返回错误
	备注	
112	请求指令格式	Get_ServoErr 0
	请求指令解释	查询单个伺服的故障码
	应答指令格式	=0740h e*:
	应答指令解释	返回单个伺服故障码，第一个参数为伺服序号； 指令返回错误
	备注	
113	请求指令格式	Get_StrPara
	请求指令解释	查询结构参数
	应答指令格式	=225.000,175.000,0.000,16.000,0.000,0.000 e*:
	应答指令解释	返回结构参数值(参数有效个数与具体机型相关,对于 SCARA 后两个参数无意义)； 指令返回错误
	备注	
114	请求指令格式	Set_StrPara [225.000,175.000,0.000,16.000,0.000,0.000]
	请求指令解释	设置结构参数(参数个数为 6，有效参数与机型有关，对于 SCARA 后两个参数无意义，可填充 0)
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
115	请求指令格式	Get_StrParaComp
	请求指令解释	查询结构补偿参数
	应答指令格式	=0.000,0.000,0.000,0.000,0.000,0.000 e*:

	应答指令解释	返回结构参数值(参数有效个数与具体机型相关,对于 SCARA 后两个参数无意义)； 指令返回错误
	备注	
116	请求指令格式	Set_StrParaComp [0.000,0.000,0.000,0.000,0.000,0.000]
	请求指令解释	设置结构补偿参数 Para1：结构补偿参数
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
117	请求指令格式	Get_RdctRatio
	请求指令解释	查询减速比
	应答指令格式	=50.000,50.000,2.000,10.000 e*:
	应答指令解释	返回减速比值(参数个数与具体轴数相关)； 指令返回错误
	备注	
118	请求指令格式	Set_RdctRatio [50.000,50.000,2.000,10.000]
	请求指令解释	设置减速比 Para1：减速比，参数有效个数与具体轴数相关，对于 SCARA，参数为 4 个
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
119	请求指令格式	Get_CpParaM
	请求指令解释	查询主耦合参数
	应答指令格式	=0.000,0.000,1.000,0.000 e*:
	应答指令解释	返回主耦合参数(参数有效个数与具体轴数相关)； 指令返回错误
	备注	
120	请求指令格式	Set_CpParaM [0.000,0.000,1.000,0.000]
	请求指令解释	设置主耦合参数 Para1：主耦合参数
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用

121	请求指令格式	Get_CpParaS
	请求指令解释	查询从耦合参数
	应答指令格式	=0.000,0.000,0.000,1.000 e*:
	应答指令解释	返回从耦合参数(参数有效个数与具体轴数相关)； 指令返回错误
	备注	
122	请求指令格式	Set_CpParaS [0.000,0.000,0.000,1.000]
	请求指令解释	设置从耦合参数 Para1：从耦合参数
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
123	请求指令格式	Get_HomeJPos 0
	请求指令解释	查询工作原点 0 Para1：工作原点序号，范围：0~4
	应答指令格式	=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回原点坐标值 指令返回错误
	备注	
124	请求指令格式	Set_HomeJPos 0 0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	设置工作原点 0 Para1：工作原点序号范围：0~4 Para2：关节坐标值（机器人关节（数组长度 8）+外部轴关节（数组长度 6））
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
125	请求指令格式	Get_ZeroPos
	请求指令解释	查询绝对零点值
	应答指令格式	=0,0,0,0 e*:
	应答指令解释	返回零点对应脉冲值（脉冲值个数与具体轴数相关）； 指令返回错误
	备注	

126	请求指令格式	Set_ZeroPos [0,0,0,0]
	请求指令解释	设置绝对零点值 Para1：绝对零点值，参数有效个数与轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
127	请求指令格式	Get_InchStep
	请求指令解释	查询寸动步长的等级
	应答指令格式	=1 e*:
	应答指令解释	返回寸动步长等级（1 表示步长为 0.05，2 表示步长为 0.5，3 表示步长为 2,4 表示步长为寸动参数设置值）； 指令返回错误
	备注	
128	请求指令格式	Get_StepMotionJ
	请求指令解释	查询寸动的自定义关节步长
	应答指令格式	=0 e*:
	应答指令解释	返回寸动的自定义关节步长，单位°； 指令返回错误
	备注	
129	请求指令格式	Set_StepMotionJ 0.100
	请求指令解释	设置寸动的自定义关节步长 Para1：关节步长，范围：0.01~10
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
130	请求指令格式	Get_StepMotionL
	请求指令解释	查询寸动的自定义线性步长
	应答指令格式	=0 e*:
	应答指令解释	返回寸动的自定义线性步长，单位 mm； 指令返回错误
	备注	
131	请求指令格式	Set_StepMotionL 0.1000
	请求指令解释	设置寸动的自定义线性步长 Para1：线性步长，范围：0.01~10
	应答指令格式	ok e*:

	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
132	请求指令格式	Get_StepMotionR
	请求指令解释	查询寸动的自定义姿态步长
	应答指令格式	=0 e*:
	应答指令解释	返回寸动的自定义姿态步长，单位°； 指令返回错误
	备注	
133	请求指令格式	Set_StepMotionR 0.1000
	请求指令解释	设置寸动的自定义姿态步长 Para1：姿态步长，范围：0.01~10
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
134	请求指令格式	Get_TeachVelLimJ
	请求指令解释	查询手动模式运动时关节速度限制
	应答指令格式	=36.000,36.000,900.000,90.000 e*:
	应答指令解释	返回关节速度限制，与轴数相关，单位°/s； 指令返回错误
	备注	
135	请求指令格式	Set_TeachVelLimJ [36.000,36.000,900.000,90.000]
	请求指令解释	设置手动模式运动时关节速度限制 Para1：关节速度限制
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
136	请求指令格式	Get_TeachVelLimL
	请求指令解释	查询手动模式运动时直角坐标下速度限制
	应答指令格式	=150.000,72.000 e*:
	应答指令解释	返回位置和姿态速度； 指令返回错误
	备注	
137	请求指令格式	Set_TeachVelLimL [150.000,72.000]
	请求指令解释	设置手动模式运动时直角坐标下速度限制 Para1：位置速度，姿态速度

	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
138	请求指令格式	Get_TeachAccLimJ
	请求指令解释	查询手动模式运动时关节加速度限制
	应答指令格式	=960.000,960.000,24000.000,2400.000 e*:
	应答指令解释	返回关节加速度限制，与轴数相关； 指令返回错误
	备注	
139	请求指令格式	Set_TeachAccLimJ [960.000,960.000,24000.000,2400.000]
	请求指令解释	设置手动模式运动时关节加速度限制 Para1：关节加速度限制，设置的参数个数和轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
140	请求指令格式	Get_TeachAccLimL
	请求指令解释	查询手动模式运动时直角坐标下加速度限制
	应答指令格式	=800.000,384.000 e*:
	应答指令解释	返回直角坐标下加速度限制，分别为位置和姿态加速度； 指令返回错误
	备注	
141	请求指令格式	Set_TeachAccLimL [800.000,384.000]
	请求指令解释	设置手动模式运动时直角坐标下加速度限制 Para1：加速度限制，设置的参数个数和轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
142	请求指令格式	Get_RunVelLimJ
	请求指令解释	查询运行关节速度限制
	应答指令格式	=360.000,360.000,9000.000,900.000 e*:
	应答指令解释	返回运行关节速度限制，与轴数相关，单位°/s； 指令返回错误
	备注	
143	请求指令格式	Set_RunVelLimJ [360.000,360.000,9000.000,900.000]

	请求指令解释	设置运行关节速度限制 Para1：运行关节速度限制，设置的参数个数和轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
144	请求指令格式	Get_RunVelLimL
	请求指令解释	查询运行直角坐标下速度限制
	应答指令格式	=1500.000,720.000 e*:
	应答指令解释	返回运行直角坐标系下速度限制，分别为位置和姿态速度； 指令返回错误
	备注	
145	请求指令格式	Set_RunVelLimL [1500.000,720.000]
	请求指令解释	设置运行直角坐标下速度限制 Para1：运行直角坐标系下速度限制，位置和姿态速度
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
146	请求指令格式	Get_RunAccLimJ
	请求指令解释	查询运行关节加速度限制
	应答指令格式	=1920.000,1920.000,48000.000,4800.000 e*:
	应答指令解释	返回运行关节加速度限制，与轴数相关； 指令返回错误
	备注	
147	请求指令格式	Set_RunAccLimJ [1920.000,1920.000,48000.000,4800.000]
	请求指令解释	设置运行关节加速度限制 Para1：运行关节加速度限制，设置参数个数与轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
148	请求指令格式	Get_RunAccLimL
	请求指令解释	查询运行直角坐标下加速度限制
	应答指令格式	=8000.000,3840.000 e*:
	应答指令解释	返回运行直角坐标下加速度限制，分别为位置和姿态加速度； 指令返回错误

	备注	
149	请求指令格式	Set_RunAccLimL [8000.000,3840.000]
	请求指令解释	设置运行直角坐标下加速度限制 Para1：位置加速度，姿态加速度
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
150	请求指令格式	Get_StopDecLimJ
	请求指令解释	查询停止关节减速度限制
	应答指令格式	=2400.000,2400.000,60000.000,6000.000 e*:
	应答指令解释	返回停止关节减速度限制，与轴数相关； 指令返回错误
	备注	
151	请求指令格式	Set_StopDecLimJ [2400.000,2400.000,60000.000,6000.000]
	请求指令解释	设置停止关节减速度限制 Para1：停止关节减速度限制，设置参数与轴数相关
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
152	请求指令格式	Get_StopDecLimL
	请求指令解释	查询停止直角坐标下减速度限制
	应答指令格式	=10000.000,4800.000 e*:
	应答指令解释	返回停止直角坐标下减速度限制，分别为位置和姿态减速度； 指令返回错误
	备注	
153	请求指令格式	Set_StopDecLimL [10000.000,4800.000]
	请求指令解释	设置停止直角坐标下减速度限制 Para1：位置和姿态减速度
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
154	请求指令格式	Get_ZonePara
	请求指令解释	查询过渡精度参数
	应答指令格式	=10.000,3.000 e*:

	应答指令解释	返回线性和关节过渡精度； 指令返回错误
	备注	
155	请求指令格式	Set_ZonePara [10.000,3.000]
	请求指令解释	设置过渡精度参数 Para1：线性和关节过渡精度
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
156	请求指令格式	Get_AxisNLim J1
	请求指令解释	查询 J1 轴负限位 Para1：轴序号，与轴数有关，范围 J1-J6
	应答指令格式	=-130.000 e*:
	应答指令解释	返回对应轴的负向轴极限值； 指令返回错误
	备注	
157	请求指令格式	Set_AxisNLim J1 -130.000
	请求指令解释	设置 J1 轴负限位 Para1：轴序号，与轴数有关，范围 J1-J6 Para2：负向极限参数值
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
158	请求指令格式	Get_AxisPLim J1
	请求指令解释	查询 J1 轴正限位 Para1：轴序号，与轴数有关，范围 J1-J6
	应答指令格式	=130.000 e*:
	应答指令解释	返回对应轴的负向轴极限值； 指令返回错误
	备注	
159	请求指令格式	Set_AxisPLim J1 130.000
	请求指令解释	设置 J1 轴正限位 Para1：轴序号，与轴数有关，范围 J1-J6 Para2：正向极限参数值
	应答指令格式	ok e*:

	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
160	请求指令格式	Get_ToolData 1
	请求指令解释	查询工具坐标系 1 参数 Para1：工具号，范围 0-15
	应答指令格式	=1; 0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000, 0.000 e*:
	应答指令解释	返回工具参数 指令返回错误
	备注	
161	请求指令格式	Set_ToolData 1 1; 0.000,0.000,0.000,0.000,0.000,0.000 0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000, 0.000
	请求指令解释	设置工具坐标系 1 参数 Para1：工具号，范围:1-15 Para2：工具参数，格式：是否机器人夹持；工具坐标系（数组长度 6） Para3：工具负载参数（质量+质心位置（数组长度 3）+质心姿态（数组长度 3）+负载惯量（数组长度 3））
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
162	请求指令格式	Get_WobjData 1
	请求指令解释	查询工件坐标系 1 参数 Para1：工件号，范围 0-15
	应答指令格式	0,1; 0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000; e*:
	应答指令解释	返回工件参数（"; "后为关联的机械单元，无参数代表未关联） 指令返回错误
	备注	
163	请求指令格式	Set_WobjData 1 0,1; 0.000,0.000,0.000,0.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000;

	请求指令解释	设置工件坐标系 1 参数 Para1：工件号，范围:1-15 Para2：工件参数，格式：是否机器人夹持，是否固定的用户坐标系；用户坐标系（数组长度 6）；工件坐标系（数组长度 6）；关联的机械单元名称（可选）
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
164	请求指令格式	Get_ToolCNum
	请求指令解释	查询当前工具坐标系号
	应答指令格式	=0 e*:
	应答指令解释	返回当前激活的工具号，范围：0~15 指令返回错误
	备注	
165	请求指令格式	Set_ToolCNum 0
	请求指令解释	设置当前工具坐标系号 Para1：工具号，范围:0-15
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
166	请求指令格式	Get_WobjNum
	请求指令解释	查询系统当前激活工件号
	应答指令格式	=0 e*:
	应答指令解释	返回当前激活的工件号，范围：0~15； 指令返回错误
	备注	
167	请求指令格式	Set_WobjNum 0
	请求指令解释	设置系统激活的工件号 Para1：工件号，范围:0-15
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
168	请求指令格式	Set_CoordType 1

	请求指令解释	设置当前坐标系类型 Para1：当前坐标系类型，范围 1 至 5，1-关节坐标系，2-基坐标系，3-工具坐标系，4-工件坐标系，5-世界坐标系
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
169	请求指令格式	Get_Interf 0
	请求指令解释	查询干涉区 0 边界坐标 Para1：干涉区序号，范围：0-7
	应答指令格式	=400.000,0.000,0.000; 372.000,130.000,-5.300 e*:
	应答指令解释	返回干涉区 0 的两个边界点的 xyz 坐标值； 指令返回错误
	备注	
170	请求指令格式	Set_Interf 0 [400.000000,0.000000,0.000000; 372.000000,130.000000,-5.300000]
	请求指令解释	设置干涉区 0 边界坐标 Para1：干涉区序号，范围：0-7 Para2：干涉区两个边界点的 xyz 坐标值
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用
171	请求指令格式	Get_CurInterf
	请求指令解释	查询当前激活的干涉区编号
	应答指令格式	=10000000 e*:
	应答指令解释	返回当前激活的干涉区编号：0 表示未激活，1 表示激活，从左至右分别代表干涉区 0-7； 指令返回错误
	备注	
172	请求指令格式	Set_CurInterf 10000000
	请求指令解释	设置当前激活的干涉区编号 Para1：干涉区编号，0 表示未激活，1 表示激活，从左至右分别代表干涉区 0-7
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	管理模式及以上使用

173	请求指令格式	Get_JumpPara
	请求指令解释	查询当前跳跃运动的高度参数
	应答指令格式	=0,0,0 e*:
	应答指令解释	返回跳跃运动的高度参数，依次对应 LH、MH、RH
	备注	
174	请求指令格式	Set_JumpPara [0,0,0]
	请求指令解释	设置当期跳跃运动的高度参数 Para1：跳跃运动高度参数，依次为 LH、MH、RH）
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
175	请求指令格式	Get_PalletPara
	请求指令解释	查询托盘参数
	应答指令格式	= 2, 3, 1, 15.000000 e*:
	应答指令解释	返回托盘参数，依次为行数，列数，层数，层高 指令返回错误
	备注	
176	请求指令格式	Set_PalletPara 2,3,1,15
	请求指令解释	设置托盘参数 Para1：托盘参数：行数（范围 0-1000），列数（范围 0-1000），层数（范围 0-1000），层高（单位 mm）
	应答指令格式	ok e*:
	应答指令解释	设置成功 设置返回错误
	备注	
177	请求指令格式	Clear_PalletPara
	请求指令解释	清空托盘参数
	应答指令格式	ok e*:
	应答指令解释	清空成功 清空返回错误
	备注	
178	请求指令格式	Get_PalletRobP 378.910,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 502.715,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 502.715,172.988,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 1,0,0

	请求指令解释	查询对应的托盘点位 Para1：定义托盘的第一个点 Para2：定义托盘的第二个点 Para3：定义托盘的第三个点 Para4：待查询点位行号，待查询点位列号，待查询点位层号
	应答指令格式	=502.715,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回值待查询的托盘点 返回错误
	备注	
179	请求指令格式	Get_Pallet4RobP 378.910,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 502.715,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 502.715,172.988,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 368.910,162.988,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 1,0,0
	请求指令解释	查询对应的托盘点位 Para1：定义托盘的第一个点 Para2：定义托盘的第二个点 Para3：定义托盘的第三个点 Para4：定义托盘的第四个点 Para5：待查询点位行号，待查询点位列号，待查询点位层号
	应答指令格式	=502.715,-40.179,435.563,20.388,0.205,-179.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回值待查询的托盘点 返回错误
	备注	
180	请求指令格式	SavePara
	请求指令解释	保存系统参数，掉电可存储
	应答指令格式	ok e*:
	应答指令解释	保存完成； 指令返回错误
	备注	管理模式及以上使用
181	请求指令格式	RecoverPara
	请求指令解释	还原系统参数（上一次保存操作后的参数）
	应答指令格式	ok e*:
	应答指令解释	还原完成； 保存返回错误

	备注	管理模式及以上使用
182	请求指令格式	Get_RobJP 0
	请求指令解释	查询 JP0 关节位置点的关节位置参数 Para1：位置变量 JP 的序号，范围 JP0-JP9999 之间已存在点
	应答指令格式	=30.000,30.000,30.000,30.000,30.000,30.000,0.000,0.000 ; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回指定点位序号的关节位置信息； 指令返回错误
	备注	
183	请求指令格式	Set_RobJP 0 15.000,15.000,15.000,15.000,-15.000,- 15.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	设置 JP0 位置点的位置参数 Para1：位置变量 JP 的序号，范围 JP0-JP9999 之间已存在点 Para2：关节位置值
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	编辑模式及以上使用
184	请求指令格式	Set_RobJPHere 0
	请求指令解释	用当前点位的参数设置全局关节位置参数 Para1：位置变量 JP 的序号，范围 JP0-JP9999 之间已存在点
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	编辑模式及以上使用
185	请求指令格式	Get_RobP 0
	请求指令解释	查询 P0 位置点的位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点
	应答指令格式	=370.763,49.600,765.861,-114.829,-76.872,-56.185; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 e*:
	应答指令解释	返回指定点位序号的位置信息； 指令返回错误
	备注	
186	请求指令格式	Set_RobP 0 370.000,50.000,765.000,-115.000,-77.000,- 56.185; 0,0,0,1; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	设置 P0 位置点的位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点 Para2：位置值

	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	编辑模式及以上使用
187	请求指令格式	Get_RobPFromFile 0 P.pts
	请求指令解释	查询 P.pts 文件的 P0 位置点的位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点 Para2；全局位置点文件名称（注意：文件名大小匹配不敏感）
	应答指令格式	=370.763,49.600,765.861,-114.829,-76.872,-56.185; 0,0,0,1; 1.000,2.000,3.000,4.000,5.000,6.000 e*:
	应答指令解释	返回指定全局位置点文件的点位序号的位置信息； 指令返回错误
	备注	
188	请求指令格式	Set_RobPToFile 0 370.000,50.000,765.000,-115.000,-77.000,-56.185; 0,0,0,1; 0.000,0.000,0.000,0.000,0.000,0.000 P.pts
	请求指令解释	设置 P.pts 文件的 P0 位置点的位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点 Para2：位置值 Para3：全局位置点文件名称（注意：文件名大小匹配不敏感）
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	编辑模式及以上使用
189	请求指令格式	Get_CurRobPFileName
	请求指令解释	查询控制器当前加载的全局点文件名称
	应答指令格式	=P.pts e*:
	应答指令解释	返回控制器当前加载的位置点文件名称； 指令返回错误
	备注	
190	请求指令格式	Set_RobPHere 0
	请求指令解释	用当前点位的参数设置全局位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误

	备注	
191	请求指令格式	Set_RobPHereToFile 0 P.pts
	请求指令解释	用当前点位的参数设置指定文件的全局位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点 para2：全局位置点文件名称（注意：文件名大小匹配不敏感）
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
192	请求指令格式	Get_RobP_Label 0
	请求指令解释	查询全局位置点标签信息 Para1：位置变量 P 的序号，范围 0-9999
	应答指令格式	=RobP0 e*:
	应答指令解释	返回指定点位序号的标签信息 指令返回错误
	备注	
193	请求指令格式	Get_RobJP_Label 0
	请求指令解释	查询全局关节点标签信息 Para1：位置变量 JP 的序号，范围 0-9999
	应答指令格式	=RobJP0 e*:
	应答指令解释	返回指定点位序号的标签信息 指令返回错误
	备注	
194	请求指令格式	Get_RobLP_Label 1 0 test.pro
	请求指令解释	查询局部位置点标签信息 Para1：任务通道，范围 0-3 Para2：位置变量 LP 的序号，范围 0-9999 Para3：局部点位所在程序名（大小写不敏感，需带.pro 后缀）
	应答指令格式	=Task1LP0 e*:
	应答指令解释	返回指定点位序号的标签信息 指令返回错误
	备注	
195	请求指令格式	Get_SpeedLimitSwitch
	请求指令解释	查询控制器限速开关状态
	应答指令格式	=0 e*;

	应答指令解释	返回限速开关状态 指令返回错误
	备注	
196	请求指令格式	Set_SpeedLimitSwitch 0
	请求指令解释	设置控制器限速开关状态 Para1: 0 ：开启限速　1 ：关闭限速
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
197	请求指令格式	Get_PR 0
	请求指令解释	查询全局偏移变量 PR0 Para1：PR 的序号，范围 0-255
	应答指令格式	=0.000000,0.000000,0.000000,0.000000, 0.000000,0.000000,0.000000 e*:
	应答指令解释	返回指定 PR 的值； 指令返回错误
	备注	
198	请求指令格式	Set_PR 0 [0.0,0.0,0.0,0.0,0.0,0.0]
	请求指令解释	设置全局偏移变量 PR0 Para1：PR 的序号，范围 0-255 Para2：PR 值
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
199	请求指令格式	Get_B 0
	请求指令解释	查询全局数值变量 B0 的值 Para1：B 的序号，范围 0-255
	应答指令格式	ok e*:
	应答指令解释	返回指定 B 的值 设置返回错误
	备注	
200	请求指令格式	Set_B 0 1
	请求指令解释	设置全局数值变量 B0 的值为 1 Para1：B 的序号，范围 0-255 Para2：变量值，范围 0-255
	应答指令格式	ok e*:

	应答指令解释	设置成功； 设置返回错误
	备注	
201	请求指令格式	Get_R 0
	请求指令解释	查询全局数值变量 R0 的值 Para1：R 的序号，范围 0-255
	应答指令格式	=1 e*:
	应答指令解释	返回指定 R 变量值； 指令返回错误
	备注	
202	请求指令格式	Set_R 0 1
	请求指令解释	设置全局数值变量 R0 的值为 1 Para1：R 的序号，范围 0-255 Para2：变量值，范围-2147483647 至 2147483647
	应答指令格式	ok e*:
	应答指令解释	设置成功； 设置返回错误
	备注	编辑模式及以上使用
203	请求指令格式	Get_D 0
	请求指令解释	查询全局数值变量 D0 的值 Para1：D 的序号，范围 0-255
	应答指令格式	=1.000 e*:
	应答指令解释	返回指定 D 变量值； 指令返回错误
	备注	
204	请求指令格式	Set_D 0 1
	请求指令解释	设置全局数值变量 D0 的值为 1 Para1：D 的序号，范围 0-255 Para2：变量值，范围-9999999.999 至 9999999.999
	应答指令格式	ok e*:
	应答指令解释	设置成功； 设置返回错误
	备注	编辑模式及以上使用
205	请求指令格式	Get_ModbusCoil 2048 1
	请求指令解释	查询 Modbus 变量区地址为 0 的 1 个线圈的值 Para1：:线圈地址，范围 0-8191 Para2：线圈个数，范围 1-8
	应答指令格式	=1 e*:

	应答指令解释	返回对应值； 指令返回错误
	备注	
206	请求指令格式	Set_ModbusCoil 2048 1 1
	请求指令解释	设置 Modbus 变量区地址为 2048 的 1 个线圈的值为 0 Para1：线圈地址，范围 2048-4095,6144-8191 Para2：:线圈个数，范围 1-8 Para3：需要设置的线圈值,范围 0-255
	应答指令格式	ok e*:
	应答指令解释	设置成功； 设置返回错误
	备注	编辑模式及以上使用
207	请求指令格式	Get_ModbusReg 16384 2 u16
	请求指令解释	查询 Modbus 变量区地址为 0 的 1 个寄存器的值 Para1：寄存器地址，范围 0-65535 Para2：读取的寄存器个数，范围 1-8 Para3：读取的寄存器类型，u16 为无符号 16 位整型，f32 为单精度浮点型
	应答指令格式	=1,1 e*:
	应答指令解释	返回对应值； 指令返回错误
	备注	
208	请求指令格式	Set_ModbusReg 16384 2 u16 [1,1]
	请求指令解释	设置 Modbus 变量区地址为 16384 的 1 个寄存器的值为 0 Para1：:寄存器地址，范围 16384-32767,49152-65535 Para2：:写入的寄存器个数 Para3：:寄存器类型，u16 为无符号 16 位整型，f32 为单精度浮点型 Para4：:需要设置的寄存器值
	应答指令格式	ok e*:
	应答指令解释	设置成功； 设置返回错误
	备注	编辑模式及以上使用
209	请求指令格式	Get_ModbusReg 60000 8 f32
	请求指令解释	查询 Modbus 变量区的寄存器值，数据类型为 float Para1：modbus 区寄存器地址，范围 0-65535 Para2：读取的寄存器总个数,范围 1-8 Para3：float 数据类型
	应答指令格式	=0.1,0.2,0.3,0.4 e*:

	应答指令解释	返回查询 Modbus 变量区的寄存器值; 指令返回错误
	备注	
210	请求指令格式	Set_ModbusReg 60000 2 f32 1.5
	请求指令解释	设置 Modbus 变量区的寄存器值,数据类型为 float Para1：modbus 区寄存器地址，范围 16384-32767,49152-65535 Para2：读取的寄存器总个数,范围 1-8 Para3：float 数据类型 Para4：设置的参数，如果需要设置多个参数，请用逗号分隔
	应答指令格式	ok e*:
	应答指令解释	设置成功; 设置返回错误
	备注	编辑模式及以上使用
211	请求指令格式	Get_PlcVar Byte 0
	请求指令解释	查询 PLC Byte 型变量的值 Para1：Byte 类型 Para2：Byte 变量序号，范围 0-255
	应答指令格式	=5 e*:
	应答指令解释	返回 PLC Byte 型变量的值 查询返回错误
	备注	
212	请求指令格式	Get_PlcVar Int 0
	请求指令解释	查询 PLC Int 型变量的值 Para1：Int 类型 Para2：Int 变量序号，范围 0-255
	应答指令格式	=15 e*:
	应答指令解释	返回 PLC Int 型变量的值; 查询返回错误
	备注	
213	请求指令格式	Get_PlcVar DInt 0
	请求指令解释	查询 PLC DInt 型变量的值 Para1：DInt 类型 Para2：DInt 变量序号，范围 0-255
	应答指令格式	=0 e*:
	应答指令解释	查询 PLC DInt 型变量的值 查询返回错误
	备注	
214	请求指令格式	Get_PlcVar LReal 0

	请求指令解释	查询 PLC LReal 型变量的值 Para1：LReal 类型 Para2： LReal 变量序号，范围 0-255
	应答指令格式	=0.000000 e*:
	应答指令解释	查询 PLC LReal 型变量的值 查询返回错误
	备注	
215	请求指令格式	Get_UserAlarm 0
	请求指令解释	查询 0 号自定义报警的内容 Para1：自定义报警序号，范围 0-15
	应答指令格式	=UserError1 = e*:
	应答指令解释	返回对应的报警内容描述 返回空； 指令返回错误
	备注	
216	请求指令格式	Set_UserAlarm 0 [UserError1]
	请求指令解释	设置 0 号自定义报警 Para1：自定义报警序号，范围 0-15 Para2：要设置的报警内容,字节长不超过 40byte 注：当该指令只有第一个参数时，表示将报警内容置清空
	应答指令格式	ok e*:
	应答指令解释	设置成功； 设置返回错误
	备注	管理模式及以上使用
217	请求指令格式	Get_Print
	请求指令解释	查询控制器打印信息，包括程序 print 指令的打印内容和系统错误提示内容
	应答指令格式	=PRINT:test = e*:
	应答指令解释	返回对应值； 返回空； 指令返回错误
	备注	
218	请求指令格式	CurCtrlDev
	请求指令解释	查询当前控制权所属设备
	应答指令格式	=0 e*:

	应答指令解释	返回设备编号，0-InoTeachPad/InoRobLab，1-InoRobShop 平台，2-API 网络设备，3-远端 IO，4-远端 modbus； 指令返回错误
	备注	
219	请求指令格式	CurPermit
	请求指令解释	查询当前拥有控制权许可的 API 网络设备
	应答指令格式	=0 null =1 Ip:10.44.52.35 Port:5000 e*:
	应答指令解释	对应返回值，第一个返回值 0 表示无 API 设备获得许可，1 表示当前 API 设备获得许可，后面返回值分别为当前设备的 Ip 地址和端口，2 表示其它 API 设备，后面返回值分别为其 Ip 地址和端口； 指令返回错误
	备注	
220	请求指令格式	(1)AcqPermit (2)AcqPermit forcibly
	请求指令解释	(1)当前 API 网络客户端设备请求获取控制权许可 (2)当前 API 网络客户端设备强制获取控制权许可
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
221	请求指令格式	RemovePermit
	请求指令解释	当前 API 网络客户端设备释放控制权
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
222	请求指令格式	CurUserType
	请求指令解释	查询当前用户的模式
	应答指令格式	=1 e*:
	应答指令解释	返回当前用户的模式，0-客户模式，1-编辑模式，2-管理模式，3-厂家模式； 指令返回错误
	备注	
223	请求指令格式	UserLogin 1 000000

	请求指令解释	登陆对应的用户模式 Para1：用户模式：0-客户模式，1-编辑模式，2-管理模式，3-厂家模式 Para2：密码
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
224	请求指令格式	UserLogout
	请求指令解释	退出当前登陆模式，默认模式为客户模式
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
225	请求指令格式	Set_SysTime 20170101083000
	请求指令解释	设置系统时间 Para1：时间的年月日时分秒形式
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
226	请求指令格式	(1)Latch ON (2)Latch OFF
	请求指令解释	(1)锁存功能开启 (2)锁存功能关闭
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
227	请求指令格式	Get_LatchSts
	请求指令解释	查询锁存功能开启状态
	应答指令格式	=1 e*:
	应答指令解释	返回锁存功能开启状态：0-关闭，1-开启； 指令返回错误
	备注	
228	请求指令格式	Get_LatchSum
	请求指令解释	查询锁存位置点的总数

	应答指令格式	=1 e*:
	应答指令解释	返回锁存位置点的总数 指令返回错误
	备注	管理模式及以上使用
229	请求指令格式	Get_LatchRobP 0
	请求指令解释	读取对应锁存点的位置参数 Para1：预留参数 注：按顺序读取，每个位置只能读到一次
	应答指令格式	=0; 0.000,0.000,0.000,0.000,0.000,0.000; 0,0,0,0; 0.000,0.000,0.000,0.000,0.000,0.000 e*:
	应答指令解释	返回值，第一个返回数表示有无锁存数据（0-无，1-有），后面表示锁存值，表示当前工具相对当前工件的值； 指令返回错误
	备注	
230	请求指令格式	Clr_LatchPos
	请求指令解释	清除锁存位置
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
231	请求指令格式	Set_CollModeAndAction 0 0
	请求指令解释	设置数据流模式下碰撞检测功能开关以及碰撞报警后触发的动作 Para1：数据流模式下碰撞检测功能开关：0-关闭碰撞检测，1-打开碰撞检测 Para2：碰撞报警后触发的动作: 0-不对触发动作进行定义 3-碰撞停机 7-碰撞回退 当选择"0-不对触发动作进行定义"时，实际触发动作，用以下设置为准： (1)API：IMC100_Set_TeachModeCollAction/ IMC100_Set_PlaybackModeCollAction (2)InoTeachPad：【设置】-【运动参数】-【碰撞检测设置 (3)InRobotLab：【控制器参数设置】-【运动参数】-【碰撞检测】
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误

	备注	只对数据流模式有效
232	请求指令格式	Get_CollModeAndAction
	请求指令解释	读取数据流模式或工程指令的碰撞检测功能开关以及碰撞报警后触发的动作
	应答指令格式	=0,0 e*:
	应答指令解释	返回值，第一个返回碰撞检测功能开关，0-关闭碰撞检测，1-打开碰撞检测，第二个返回碰撞报警后触发的动作，0-不对触发动作进行定义，3-碰撞停机，7-碰撞回退； 指令返回错误
	备注	
233	请求指令格式	Set_AxisCollMode 1 0
	请求指令解释	设置数据流模式下单轴的碰撞检测开关 Para1：轴号，与轴数有关，范围：1~6 Para2：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
234	请求指令格式	Get_AxisCollMode 1
	请求指令解释	读取数据流模式或工程指令的单轴碰撞检测开关
	应答指令格式	=0 e*:
	应答指令解释	返回单轴碰撞检测开关； 指令返回错误
	备注	
235	请求指令格式	Set_AxisCollLevel 1 100
	请求指令解释	设置数据流模式下单轴的碰撞检测灵敏度 Para1：轴号，与轴数有关，范围：1~6 Para2：轴碰撞检测灵敏度，范围：25~300
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
236	请求指令格式	Get_AxisCollLevel 1
	请求指令解释	读取数据流模式或工程指令的单轴碰撞检测灵敏度
	应答指令格式	=100 e*:
	应答指令解释	返回单轴碰撞检测灵敏度； 指令返回错误

	备注	
237	请求指令格式	Set_TeachModeAxisCollMode 1 0
	请求指令解释	设置手动模式的单轴碰撞检测开关 Para1：轴号，与轴数有关，范围：1~6 Para2：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
238	请求指令格式	Get_TeachModeAxisCollMode 1
	请求指令解释	读取手动模式的单轴碰撞检测开关 Para1：轴号，与轴数有关，范围：1~6
	应答指令格式	=0 e*:
	应答指令解释	返回手动模式的单轴碰撞检测开关； 指令返回错误
	备注	
239	请求指令格式	Set_TeachModeCollAction 3
	请求指令解释	设置手动模式的碰撞报警后触发的动作 Para1：碰撞报警后触发的动作:3-碰撞停机，7-碰撞回退
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
240	请求指令格式	Get_TeachModeCollAction
	请求指令解释	读取手动模式的碰撞报警后触发的动作
	应答指令格式	=3 e*:
	应答指令解释	返回碰撞报警后触发的动作； 指令返回错误
	备注	
241	请求指令格式	Set_TeachModeAxisCollLevel 1 100
	请求指令解释	设置手动模式的碰撞检测灵敏度 Para1：轴号，与轴数有关，范围：1~6 Para2：轴碰撞检测灵敏度，范围：25~300
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	

242	请求指令格式	Get_TeachModeAxisCollLevel 1
	请求指令解释	读取手动模式的碰撞检测灵敏度 Para1：轴号，与轴数有关，范围：1~6
	应答指令格式	=100 e*:
	应答指令解释	返回的碰撞检测灵敏度； 指令返回错误
	备注	
243	请求指令格式	Set_PlayBackModeAxisCollMode 1 0
	请求指令解释	设置自动模式的单轴碰撞检测开关 Para1：轴号，与轴数有关，范围：1~6 Para2：碰撞检测开关：0-关闭某轴碰撞检测，1-打开某轴碰撞检测
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
244	请求指令格式	Get_PlayBackModeAxisCollMode 1
	请求指令解释	读取自动模式的单轴碰撞检测开关 Para1：轴号，与轴数有关，范围：1~6
	应答指令格式	=0 e*:
	应答指令解释	返回单轴碰撞检测开关； 指令返回错误
	备注	
245	请求指令格式	Set_PlayBackModeCollAction 0
	请求指令解释	设置自动模式的碰撞报警后触发的动作 Para1：碰撞报警后触发的动作:3-碰撞停机，7-碰撞回退
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
246	请求指令格式	Get_PlayBackModeCollAction
	请求指令解释	读取自动模式的碰撞报警后触发的动作
	应答指令格式	=0 e*:
	应答指令解释	返回碰撞报警后触发的动作； 指令返回错误
	备注	
247	请求指令格式	Set_PlayBackModeAxisCollLevel 1 100

	请求指令解释	设置自动模式的碰撞检测灵敏度 Para1：轴号，与轴数有关，范围：1~6 Para2：轴碰撞检测灵敏度，范围：25~300
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
248	请求指令格式	Get_PlayBackModeAxisCollLevel 1
	请求指令解释	读取自动模式的碰撞检测灵敏度 Para1：轴号，与轴数有关，范围：1~6
	应答指令格式	=100 e*:
	应答指令解释	返回的碰撞检测灵敏度； 指令返回错误
	备注	
249	请求指令格式	Get_RobotAxisNum
	请求指令解释	读取机器人当前本体轴数
	应答指令格式	=4 e*:
	应答指令解释	返回机器人当前本体轴数； 指令返回错误
	备注	
250	请求指令格式	Set_BindTcpSpeedValue 0 0 5 0 500
	请求指令解释	将线速度模拟量输出绑定到 DA[num] Para1:绑定的 DA 的 index，范围（0~激活的 DA 数量） Para2:最小输出模拟量，范围（选定的 DA 配置的输出范围） Para3:最大输出模拟量，范围（选定的 DA 配置的输出范围） Para4:最小速度，范围（0~maxSpeed） Para5:最大速度，范围（0~100000）
	应答指令格式	=0 e*
	应答指令解释	指令完成； 指令返回错误
	备注	
251	请求指令格式	Get_BindTcpSpeedValue 0
	请求指令解释	得到输入的 DA index 的设置数据 Para1:绑定的 DA 的 index，范围（0~激活的 DA 数量）
	应答指令格式	=0 e*
	应答指令解释	指令完成； 指令返回错误
	备注	

252	请求指令格式	Close_BindTcpSpeed 0
	请求指令解释	解除线速度模拟量 DA[num]的绑定 Para1:绑定的 DA 的 index，范围（0~激活的 DA 数量）
	应答指令格式	=0 e*
	应答指令解释	指令完成； 指令返回错误
	备注	
253	请求指令格式	Get_TcpSpeedDAOutAndVel 0
	请求指令解释	DA 输出的数据与线速度值 Para1:绑定的 DA 的 index，范围（0~激活的 DA 数量）
	应答指令格式	=0 e*
	应答指令解释	指令完成； 指令返回错误
	备注	
254	请求指令格式	Set_TraceRecoverMode 1
	请求指令解释	设置轨迹恢复阈值设置模式 Para1: 轨迹恢复阈值设置模式，范围 0：关节模式，1：位置模式
	应答指令格式	=0 e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
255	请求指令格式	Get_TraceRecoverMode
	请求指令解释	获取轨迹恢复阈值设置模式
	应答指令格式	=0 e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
256	请求指令格式	Set_VarTraceRecoverParams 1 50 90 50 90
	请求指令解释	设置自动或手动下，轨迹恢复阈值参数 Para1:轨迹恢复手动或自动，范围 1：手动，2：自动 Para2: TCP 距离,范围 0~20000 Para3: TCP 旋转，范围 0~360 Para4: 外部轴距离，范围 0~20000 Para5: 外部轴旋转，范围 0~360
	应答指令格式	=0 e*:
	应答指令解释	指令完成； 指令返回错误

	备注	
257	请求指令格式	Get_VarTraceRecoverParams 1
	请求指令解释	获得自动或手动下，轨迹恢复阈值参数 Para1：轨迹恢复阈值位置模式下的自动或手动选择，范围 1：手动模式，2：自动模式
	应答指令格式	=0 e*:
	应答指令解释	指令完成； 指令返回错误
	备注	
258	请求指令格式	Set_InterferZoneActStat 5 1
	请求指令解释	设置激活的干涉区 Para1：干涉区序号，范围：0~15 Para2：激活的状态：0-关闭某个干涉区，1-打开个干涉区
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
259	请求指令格式	Get_InterferZoneActStat
	请求指令解释	获取所有干涉区激活状态
	应答指令格式	=0,1,1,0,1,1,1,1,1,1,1,1,1,1,0 e*:
	应答指令解释	按顺序输出所有干涉区状态，0-关闭某个干涉区，1-打开个干涉区； 指令返回错误
	备注	只对数据流模式有效
260	请求指令格式	Set_InterferZonePara 5 22,25,0,1,10,13,1,0.0,0.0,0.0,10.1,11.1,12.1
	请求指令解释	设置干涉区参数 Para1：干涉区序号，范围：0~15 Para2： 输入信号，输出信号，内外侧(0-内侧/1-外侧)，是否报警(0-无/1-报警)，安全距离，当前工件号,设置方(0-对角/1-基准点+边长)，对角点 1x/基准点 X，对角点 1y/基准点 Y，对角点 1z/基准点 Z，对角点 2x/偏移 dx，对角点 2y/偏移 dy，对角点 2z/偏移 dz
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效 备注中的中文字符支持 15 个，英文字符支持 30 个字符
261	请求指令格式	Get_InterferZonePara 5

	请求指令解释	获取指定干涉区的参数
	应答指令格式	= 22,25,0,1,10,13,1,0.0,0.0,0.0,10.1,11.1,12.1
	应答指令解释	获取指定干涉区配置参数 输入信号，输出信号，内外侧(0-内侧/1-外侧)，是否报警(0-无/1-报警)，安全距离，当前工件号,设置方(0-对角/1-基准点+边长)，对角点 1x/基准点 X，对角点 1y/基准点 Y，对角点 1z/基准点 Z，对角点 2x/偏移 dx，对角点 2y/偏移 dy，对角点 2z/偏移 dz
	备注	
262	请求指令格式	Set_InterferToolActNum 14
	请求指令解释	激活监控对象 Para1：监控对象序号，范围：0~15
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
263	请求指令格式	Get_InterferToolActNum
	请求指令解释	获取所有干涉区激活状态
	应答指令格式	=14 e*:
	应答指令解释	激活的监控对象序号； 指令返回错误
	备注	只对数据流模式有效
264	请求指令格式	1. Set_InterferToolPara 10 0 2. Set_InterferToolPara 10 1,1,1,1,1,5,6,7,5 3. Set_InterferToolPara 10 2,10,20 4. Set_InterferToolPara 10 3,0,10.0,11.0,12.0,13.0,14.0,15.0 5. Set_InterferToolPara 10 3,1,10.0,11.0,12.0,13.0,14.0,15.0 6. Set_InterferToolPara 10 3,2,10.0,11.0,12.0,13.0,14.0,15.0,10.0,11.0,12.0,13.0,14.0,15.0,18.1

	请求指令解释	设置监控对象参数 Para1：监控对象序号，范围：0~15 Para2: 1. 监控对象类型-TCP 2. 监控对象类型-多 TCP ,TCP1 激活状态,TCP2 激活状态,TCP3 激活状态,TCP4 激活状态,TCP1 序号,TCP2 序号,TCP3 序号,TCP4 序号 3. 监控对象类型-球, 球心 Z, 球半径 R 4. 监控对象类型-方形盒 ,设置方式-0 对角点,点 1X,点 1Y,点 1Z,点 2X,点 2Y,点 3Z 5. 监控对象类型-方形盒 ,设置方式-1 基准点+偏移, 基准点 X, 基准点 Y, 基准点 Z,偏移 dx, 偏移 dy, 偏移 dz 监控对象类型-方形盒 ,设置方式-2 取点 4 点+高,点 1X,点 1Y,点 1Z,点 2X,点 2Y,点 2Z,点 3X,点 3Y,点 3Z,点 4X,点 4Y,点 4Z,高 H
	应答指令格式	ok e*:
	应答指令解释	指令完成； 指令返回错误
	备注	只对数据流模式有效
265	请求指令格式	Get_InterferToolPara 5
	请求指令解释	获取指定监控对象的参数
	应答指令格式	1. =10 0 2. =10 1,1,1,1,1,5,6,7,5 3. =10 2,10,20 4. =10 3,0,10.0,11.0,12.0,13.0,14.0,15.0 5. =10 3,1,10.0,11.0,12.0,13.0,14.0,15.0 =10 3,2,10.0,11.0,12.0,13.0,14.0,15.0,10.0,11.0,12.0,13.0,14.0,15.0,18.1
	应答指令解释	获取指定监控对象配置参数 1. 监控对象类型-TCP 2. 监控对象类型-多 TCP ,TCP1 激活状态,TCP2 激活状态,TCP3 激活状态,TCP4 激活状态,TCP1 序号,TCP2 序号,TCP3 序号,TCP4 序号 3. 监控对象类型-球, 球心 Z, 球半径 R 4. 监控对象类型-方形盒 ,设置方式-0 对角点,点 1X,点 1Y,点 1Z,点 2X,点 2Y,点 3Z 5. 监控对象类型-方形盒 ,设置方式-1 基准点+偏移, 基准点 X, 基准点 Y, 基准点 Z,偏移 dx, 偏移 dy, 偏移 dz 6. 监控对象类型-方形盒 ,设置方式-2 取点 4 点+高,点 1X,点 1Y,点 1Z,点 2X,点 2Y,点 2Z,点 3X,点 3Y,点 3Z,点 4X,点 4Y,点 4Z,高 H

	备注	
266	请求指令格式	SetMemRobP 0 370.000,50.000,765.000,-115.000,-77.000,-56.185; 0,0,0,1; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	设置 P0 位置点的位置参数 Para1：位置变量 P 的序号，范围 P0-P9999 之间已存在点 Para2：位置值
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	
267	请求指令格式	SetMemRobJP 0 15.000,15.000,15.000,15.000,-15.000,-15.000,0.000,0.000; 0.000,0.000,0.000,0.000,0.000,0.000
	请求指令解释	设置 JP0 位置点的位置参数 Para1：位置变量 JP 的序号，范围 JP0-JP9999 之间已存在点 Para2：关节位置值
	应答指令格式	ok e*:
	应答指令解释	设置完成； 设置返回错误
	备注	

附录 3：API 故障说明

说明	
函数返回值	0
故障打印信息	/
说明	(指令正常)
故障处理方法	/
函数返回值	-1
故障打印信息	e1:syntax error
说明	指令语法错误
故障处理方法	动态链接库版本问题，寻求技术支持。(或点位指令的参数未初始化。)
函数返回值	-2
故障打印信息	e2:number of parameter unmatched
说明	指令参数个数不匹配
故障处理方法	动态链接库版本问题，寻求技术支持。
函数返回值	-3
故障打印信息	e3:parameter value illegal
说明	指令参数值不合理
故障处理方法	依据函数说明，重新给定函数参数。
函数返回值	-4
故障打印信息	e4:not allowed in current mode
说明	当前模式下不允许该指令
故障处理方法	根据实际情况切换手动或自动或数据流模式后重新调用。
函数返回值	-5
故障打印信息	e5:not allowed when robot is running
说明	机器人运行时不允许该指令
故障处理方法	机器人停止后重新调用。
函数返回值	-6
故障打印信息	e6:system in emergency
说明	系统处于急停状态
故障处理方法	系统处于急停松开后重新调用。
函数返回值	-7
故障打印信息	e7:system fault
说明	系统有故障
故障处理方法	清除系统故障后重新调用。
函数返回值	-8
故障打印信息	e8:motion mode closed
说明	数据流模式关闭
故障处理方法	打开数据流模式后重新调用
函数返回值	-9
故障打印信息	e9:motor off

说明	电机未使能
故障处理方法	电机使能后重新调用。
函数返回值	-10
故障打印信息	e10:rsv
说明	保留
故障处理方法	保留
函数返回值	-11
故障打印信息	e11:instruction unfinished
说明	运动指令缓冲区仍有未完成的指令
故障处理方法	待运动指令缓冲区空闲时重新调用。
函数返回值	-12
故障打印信息	e12:ouput unavaiable
说明	用户无权限控制输出端口
故障处理方法	获取该端口权限后重新调用，或者控制其他可用端口。
函数返回值	-13
故障打印信息	e13:read AD failed
说明	读取 AD 通道失败
故障处理方法	确认模块正确配置后重新调用。
函数返回值	-14
故障打印信息	e14:write DA failed
说明	写入 AD 通道失败
故障处理方法	确认模块正确配置后重新调用。
函数返回值	-15
故障打印信息	e15:write DO failed
说明	写入 DO 通道失败
故障处理方法	确认模块正确配置后重新调用。
函数返回值	-16
故障打印信息	e16:command invalid
说明	暂停指令无效
故障处理方法	程序停止状态下不调用暂停函数。
函数返回值	-17
故障打印信息	e17:not allowed in current coordinate
说明	当前坐标系下不允许该指令
故障处理方法	切换坐标系后重新调用。
函数返回值	-18
故障打印信息	e18:mode conflict
说明	运动模式冲突
故障处理方法	关闭数据流模式后重新调用函数。
函数返回值	-19
故障打印信息	e19:indAxis mode err
说明	独立轴模式错误
故障处理方法	请切换成独立轴后再调用
函数返回值	-20

故障打印信息	e20:program non-existent
说明	设置的程序路径不存在
故障处理方法	重新输入路径。
函数返回值	-21
故障打印信息	e21:point non-existent
说明	位置点 P 或偏移 LPR 不存在
故障处理方法	确认点位无误并重新打开工程后调用。
函数返回值	-22
故障打印信息	e22:calc error
说明	内部计算错误
故障处理方法	确认传入参数准确后重新调用。
函数返回值	-23
故障打印信息	e23:rsv
说明	保留
故障处理方法	保留
函数返回值	-24
故障打印信息	e24:without permit
说明	当前以太网设备未获得控制许可
故障处理方法	切换控制权并获取控制许可后重新调用。
函数返回值	-25
故障打印信息	e25:ETH without authorization
说明	当前控制权不是以太网设备
故障处理方法	切换控制权后重新调用。
函数返回值	-26
故障打印信息	e26:rsv
说明	保留
故障处理方法	保留
函数返回值	-27
故障打印信息	e27:low user grade
说明	当前用户权限等级不够
故障处理方法	登陆更高用户等级后重新调用
函数返回值	-28
故障打印信息	e28:permit occupied
说明	控制许可已被其它以太网设备申请
故障处理方法	强制获取控制许可后重新调用。
函数返回值	-29
故障打印信息	e29:internal fault
说明	内部调用错误
故障处理方法	根据规格调用 api 指令。
函数返回值	-30
故障打印信息	e30:modbus unavailable
说明	未配置 modbus 从站
故障处理方法	配置 modbus 从站后重新调用。

函数返回值	-31
故障打印信息	e31:condition unfulfilled to write
说明	当前不可设置系统参数
故障处理方法	机器人停止或配置文件保存完成后重新调用。
函数返回值	-32
故障打印信息	e32:can't write para when no project
说明	未选择工程不允许写参数
故障处理方法	选择工程后再进行写工程参数操作。
函数返回值	-33
故障打印信息	e33:pallet para err
说明	托盘参数错误
故障处理方法	托盘未建立或托盘参数错误
函数返回值	-253
故障打印信息	/
说明	内部计算异常
故障处理方法	动态链接库版本问题，寻求技术支持。
函数返回值	-254
故障打印信息	/
说明	函数输入参数错误
故障处理方法	依据函数说明，重新给定函数参数。
函数返回值	-255
故障打印信息	/
说明	网络通信异常
故障处理方法	检查网络连接确认无误后重新调用。

附录 4：常见问题及处理方法

问题 1：远程以太网控制权限切换到 InoTeachPad 或者 InoRobotLab 控制权限，机器人控制器报错 0x0084？

0x0084 报错原因：（1）控制器向 API 客户端发送数据错误；（2）检测 API 通讯断开时，对应的 API 控制权未释放，且系统处于使能状态。

为确保安全，从远程以太网权限切换到 InoTeachPad/InoRobotLab 控制权限之前，API 客户端先向控制器发送下使能命令【IMC100_MotorEnable(0)】，再释放 API 客户端控制权【IMC100_RemovePermit】，退出 API 通讯连接【IMC100_Exit_ETH】，然后再切换到示教器/PC 编程平台控制权限。

问题 2：如何查询 api 版本号？

dll 右键->属性->详细信息