

第四章 通讯协议

4.1 概述

UDP（用户数据报协议）提供一种简单的基于数据报的进程到进程通信机制。UDP 提供区分在单个应用程序上对多个应用程序通信时的端口，使用 UDP 协议通信时，每个 UDP 消息，除了要发送的数据之外，还会包含一个目标端口号和源端口号，使得消息能准确的传到接收方。

UDP 使用底层互联网协议（IP），不使用确认来确保消息到达，也不对传入的消息进行排序，并且也不需要反馈，使信息在机器之间的流动速率自由而不受阻碍。

4.2 通信建立

使用 UDP 通信，需要对设备分配 IP 和端口，默认设置用户电脑端的 IP 为 192.168.0.131，端口为 10000，运动控制器的 IP 为 192.168.0.125，端口为 5000，通信地址不固定，用户可根据需要进行修改。

位置	IP 地址（默认）	端口号
用户电脑端	192.168.0.131	10000
运动控制器	192.168.0.125	5000

表 3 用户电脑端与运动控制器地址

4.3 数据频率

为使运动平台动作连续且平滑，将通信频率设为 100Hz。出于安

全考虑，如果通信间数据延迟超过一定时间，运动控制器将产生超时故障，通信将断开，不再接收运动平台监控软件发送来的数据。

第五章 运动控制器

运动控制器将运动平台监控软件发送来的控制命令数据进行解析，在控制器内部通过一系列运动学控制算法完成对数据的分析和计算，将处理后的运动姿态成功解析为六自由度运动平台可执行数据，使运动平台完成相应动作。为实现控制流程，将运动平台在运行过程中不同的阶段分为多种系统状态。

5.1 系统状态机

根据运动控制需求，将六自由度运动平台的运动行为分为多种系统状态，运动控制器以状态机的形式驱动六自由度运动平台进入不同的状态。运动平台在执行运动命令前，需经过系统初始化、寻底、底位上升到中位的状态过程，正确完成这些状态后，运动平台进入正常运动状态。运动平台处于中位的状态下，运动平台监控软件发送的运动姿态数据方可生效。下表为常见系统状态及描述。

系统状态	描述
寻底	运动平台向运动形态的最底部运行
底位上升到中位	使处在底位的运动平台向运动形态的中位运行，是后续根据姿态数据运动的基础
随动运行	此状态下运动平台可以根据上位机发送来的姿态数据

	进行实时运动
指令运行	在此状态下，运动平台根据运动平台监控软件发来的最终姿态数据，自动运行到指定姿态，或进行连续周期性运动
设置修改运动中心点	运动平台根据命令修改中心点时，进入此状态
脚本运行	此状态表示运动平台跟随着内部存储的脚本数据运行
由中位回落到底位	原本处在运动形态的中位向最底部运行
回中位	由其他运动姿态运行到中位
手动模式	此状态下，运动平台监控软件可对运动平台进行一或多条电动缸运行的控制
紧急停机	此状态下，运动平台立即停止运动且不再对运动平台监控软件发来的命令数据做相应的动作
系统初始化完成	此状态表示运动平台参数初始化，通信连接完成，可正常运行
驱动器报错	此状态表示驱动器出现故障，不再继续运动

表 4 系统状态

5.1.1 运动平台监控软件控制指令

运动平台监控软件通过发送控制指令，运动控制器接收到控制指令后，执行控制指令对应的动作和状态切换。如下表中所示，状态代码为十进制 8 位无符号整型数据。

代码	指令	描述
2	回中位	运动平台收到此命令后，会停止当前自动运行状态，然后运动到运动形态中位
3	紧急停机	运动平台收到此命令后，立即停止当前动作，并对后续命令不进行处理
4	寻底初始化	运动平台进行寻底
5	手动模式	运动平台进入手动模式状态，可以对一或多条电动缸进行单独控制，常用在调试中
6	由底位上升到中位	运动平台运动到运动形态中位
7	由中位回落到底位	运动平台由原先的中位降到底位
9	随动运行	运动平台实时接收和执行上位机发送的姿态值
11	指令运行	发送此指令加姿态数据给运动控制器，使平台可以自动运行到指定姿态或进行连续周期性运动
13	脚本运行	允许运动控制平台从内部存储的数据脚本中获取姿态数据进行运动，脚本文件每行有 7 个数据，用逗号隔开（俯仰、横摇、偏航、横向、纵向、垂向、0（保留））单位° 或 mm。

表 5 运动平台监控软件控制指令

5.1.2 运动控制器反馈状态及代码

运动控制器根据六自由度运动平台当前的系统状态，将当前系统状态对应的状态代码反馈给运动平台监控软件。如下表中所示，状态代码为十进制 8 位无符号整型数据。

代码	系统状态	代码	系统状态
0	寻底初始化中	9	脚本运行完成
1	寻底初始化完成	12	由中位回落到底位过程中
2	由底位上升到中位过程中	13	由中位回落到底位完成
3	由底位上升到中位完成	14	回中位过程中
4	随动运行中	32	手动模式
5	指令运行中	33	紧急停机
8	脚本运行中	55	系统初始化完成

表 6 反馈状态及状态代码

下图为运动控制器系统状态控制流程：

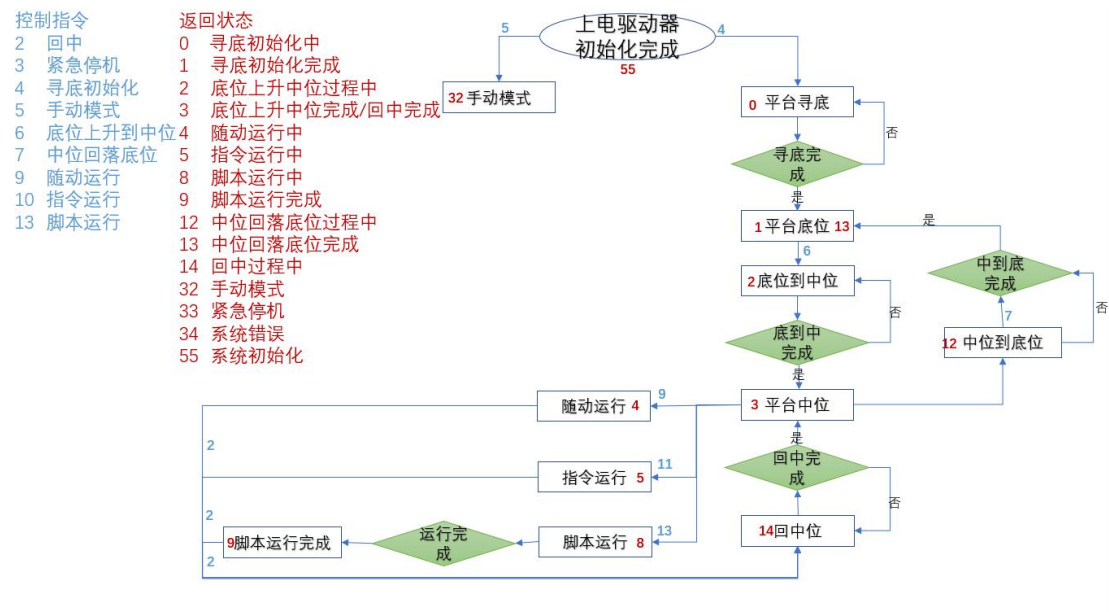


图 4 运动控制器系统状态控制流程

5.2 通信协议（数据交互）

为实现运动平台监控软件与运动控制器之间的数据交换制定此通信协议。通信协议为 UDP 协议，用户数据的通信协议分为两部分，即运动平台监控软件向运动控制器发送控制命令数据和运动控制器向运动平台监控软件返回当前状态信息。

5.2.1 运动平台监控软件向运动控制器发送数据

运动平台监控软件向运动控制器发送的数据中包含标识位、状态指令、文件运行编号、运动缸选择、手动模式的运行速度、运动姿态数据和时间戳等。每次发送的数据包长度不变，不相关的数据置零。

数据名	数据类型	数据大小	数据作用
标识位（ID）	char	1byte	数据包的标识，固定值为55
状态指令（Cmd）	char	1byte	发送状态指令可以改变运动控制器运行状态，指令参照反馈状态及状态代码表
子状态指令（SubCmd）	char	1byte	只有在指令运行时需要子状态指令 1：单步运行 2：正弦波运行 3：缓冲停止

脚本运行文件 编 号 (FileNum)	char	1byte	当用到脚本运行时,用编号调用内部存储中保存的运行数据文件
运 动 缸 选 择 (CyChoose)	char	1byte	只有在手动模式下需要用到,低位到高位 的 6 个 bit 分别对应 6 条电动缸,置 1 选中, 置 0 不选中
数 字 量 输 出 (DO)	char	1byte	当接入外设时,才用到此数据,低位到高位 的 8 个 bit 分别对应 8 个 DO,置 1 高电平, 置 0 低电平
手动模式运行 速 度 (JogSpeed)	short	2byte	在手动模式下,控制电动缸的运行速度
随动模式姿态 (DOFs)	float	24(4*6)byte	随动运行下,6 组数据直接控制平台在六个自由度上的状态量(单位: ° 或 m)
指令运行正弦 幅值 (Amp)	float	24(4*6)byte	在指令运行的正弦运行状态下,6 组数据控制平台在六个自由度上的正弦运动幅值(单位: ° /s

			和 m/s)
指令运行正弦 频率 (Fre)	float	24(4*6)byte	在自动运行的正弦运行 状态下, 6 组数据控制平 台在六个自由度上的正 弦运动频率(单位: Hz)
指令运行正弦 相位 (Pha)	float	24(4*6)byte	在自动运行的正弦运行 状态下, 6 组数据控制平 台在六个自由度上的正 弦运动相位(单位: °)
指令运行单步 目 标 位 置 (Pos)	float	24(4*6)byte	在自动运行的单步运行 状态下, 6 组数据控制平 台在六个自由度上的运 动目标位置(单位: ° 或 m)
指令运行单步 运 行 速 度 (Spd)	float	24(4*6)byte	在自动运行的在单步运 行状态下, 6 组数据控制 平台在六个自由度上的 运动速度(单位: ° /s 或 m/s)
保留位 1	float	12(4*3)byte	保留位
保留位 2	float	12(4*3)byte	保留位
时间戳 (Time)	uint32_t	4byte	数据发送时间

表 7 运动平台监控软件向运动控制器发送数据

注：六个自由度上的姿态数据组成按照俯仰、横摇、偏航、横向、纵向、垂向的顺序，即分别是 X、Y、Z 三轴的旋转量和 X、Y、Z 三轴的位移量。

如需要使用洗出算法，请单独联系厂家。

数据结构体组成如下所示：



typedef struct

{

unsigned char ID; //数据包标识位，固定值为 55

unsigned char Cmd; //状态指令控制字

unsigned char SubCmd; //子状态指令

unsigned char FileNum; //脚本运行文件编号

unsigned char CyChoose; //电动缸选择

unsigned char DO; //数字量输出

short JogSpeed; //手动模式运行速度

float DOFs[6]; //随动模式姿态，0-5 对应 X 轴转动，Y

轴转动，Z 轴转动，X 轴平移，Y 轴平移，Z 轴平移；单位：度或米

float Amp[6]; //指令运行正弦幅值；单位：度或米

float Fre[6]; //指令运行正弦频率；单位：赫兹 Hz

float Pha[6]; //指令运行正弦相位；单位：度

float Pos[6]; //指令运行单步目标位置；单位：度或米

```
float Spd[6];           //指令运行单步运行速度;单位:度或米

float Vxyz[3];          //保留位

float Axyz[3];          //保留位

uint32_t Time;          //数据发送时间

}SCommand_Typedef
```

5.2.2 运动控制器向运动平台监控软件返回当前状态信息

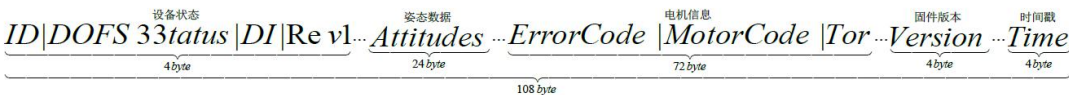
运动控制器向运动平台监控软件返回的数据中包含标识位、当前状态、数字输入、状态信息、固件版本和时间戳等。

数据名	数据类型	数据大小	数据作用
标识位 ID	char	1byte	数据包的标识，固定为 55
运动控制器 状态 DOFStatus	char	1byte	当前系统所处的状态
数字输入 DI	char	1byte	当接入外设时，才用到此数据，低位到高位 的 8 个 bit 分别对应 8 个 DI，有输入置 1，无输入置 0
保留位 Rev1	char	1byte	保留位，目前未使用，默认为 0

实际姿态 Attitudes	float	24(4*6)byte	6 组数据返回运动平台实时六自由度姿态数据
错误代码 ErrorCode	float	24(4*6)byte	当电动缸驱动器报错时，6 组数据记录返回对应电动缸的错误代码
电机码值 MotorCode	float	24(4*6)byte	6 组数据返回对应六条电动缸的电机码值
电机力矩 Tor	float	24(4*6)byte	6 组数据返回对应六条电动缸的电机力矩
运动控制器 固件版本 Version	uint32_t	4byte	运动控制器固件版本号
时间戳 Time	uint32_t	4byte	数据发送时间

表 8 运动控制器向运动平台监控软件返回状态信息

注：姿态 6 组数据分别是俯仰、横摇、偏航、横向、纵向、垂向。错误代码、电机码值、电机力矩的 6 组数据分别是 1~6 条电动缸的数据。数据结构体组成如下所示；



typedef struct

{

unsigned char ID; //标识位

```
unsigned char DOFStatus;    //当前系统所处状态

unsigned char DI;          //数字输入

unsigned char Rev1;        //保留位

float Attitudes[6];        //实际姿态，0-5 对应 X 轴转动，Y
轴转动，Z 轴转动，X 轴平移，Y 轴平移，Z 轴平移;单位;度或米

float ErrorCode[6];        //错误代码

float MotorCode[6];        //电机码值

float Tor[6];              //电机力矩

uint32_t Version;          //运动控制器固件版本

uint32_t Time;             //数据发送时间

}NetSend_Typedef
```

第六章 运动平台监控软件

为方便对运动平台进行运动控制和状态监控，利用运动平台监控软件进行控制指令的发送，根据需要的运动姿态输入相应的参数，运动控制器收到控制数据后运行到指定状态，同时根据运动控制器运动控制软件反馈的信息数据对运动平台当前状态进行监控。

6.1 运动平台监控软件使用前的准备

运动平台监控软件是用户操作平台的入口，它肩负着整个平台的运动控制数据的输出功能。运动平台监控软件对用户电脑要求如下：