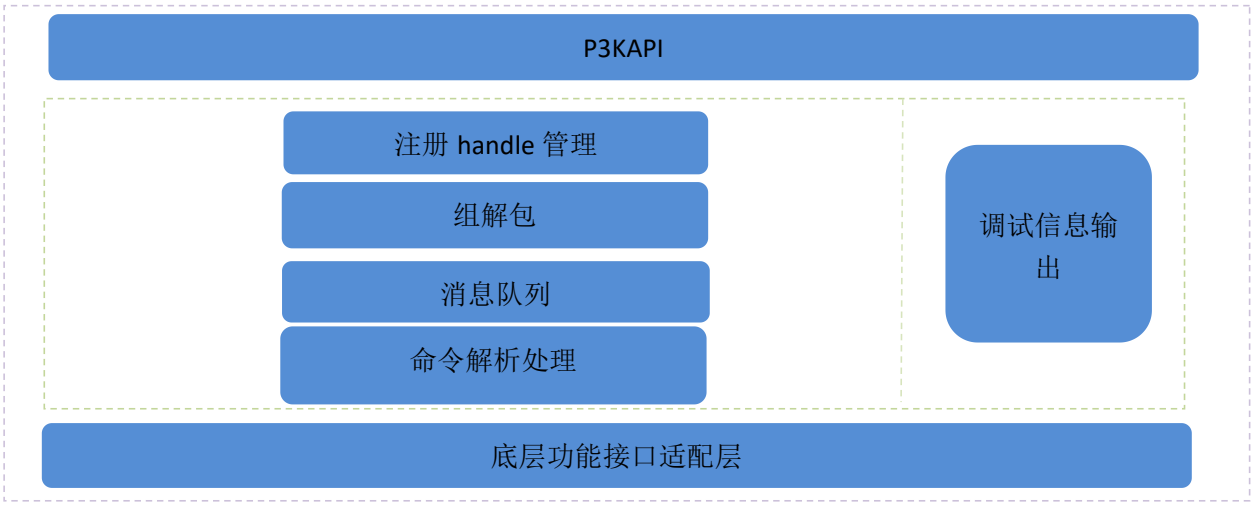
P3K 模块设计

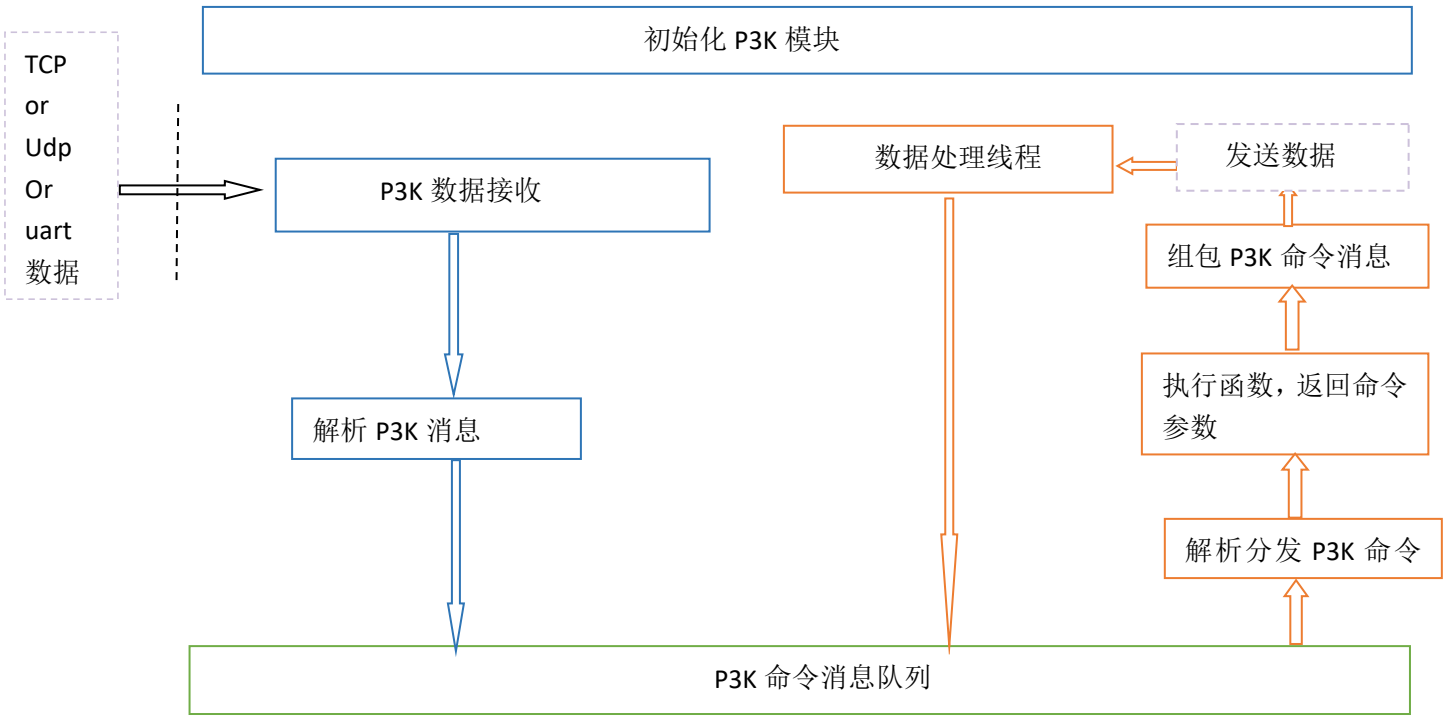
一. 功能需求：

1. P3K 命令消息的接收与解析分发执行;
2. P3K 命令消息的组包与发送;
3. P3K 模块化，标准化，易扩展

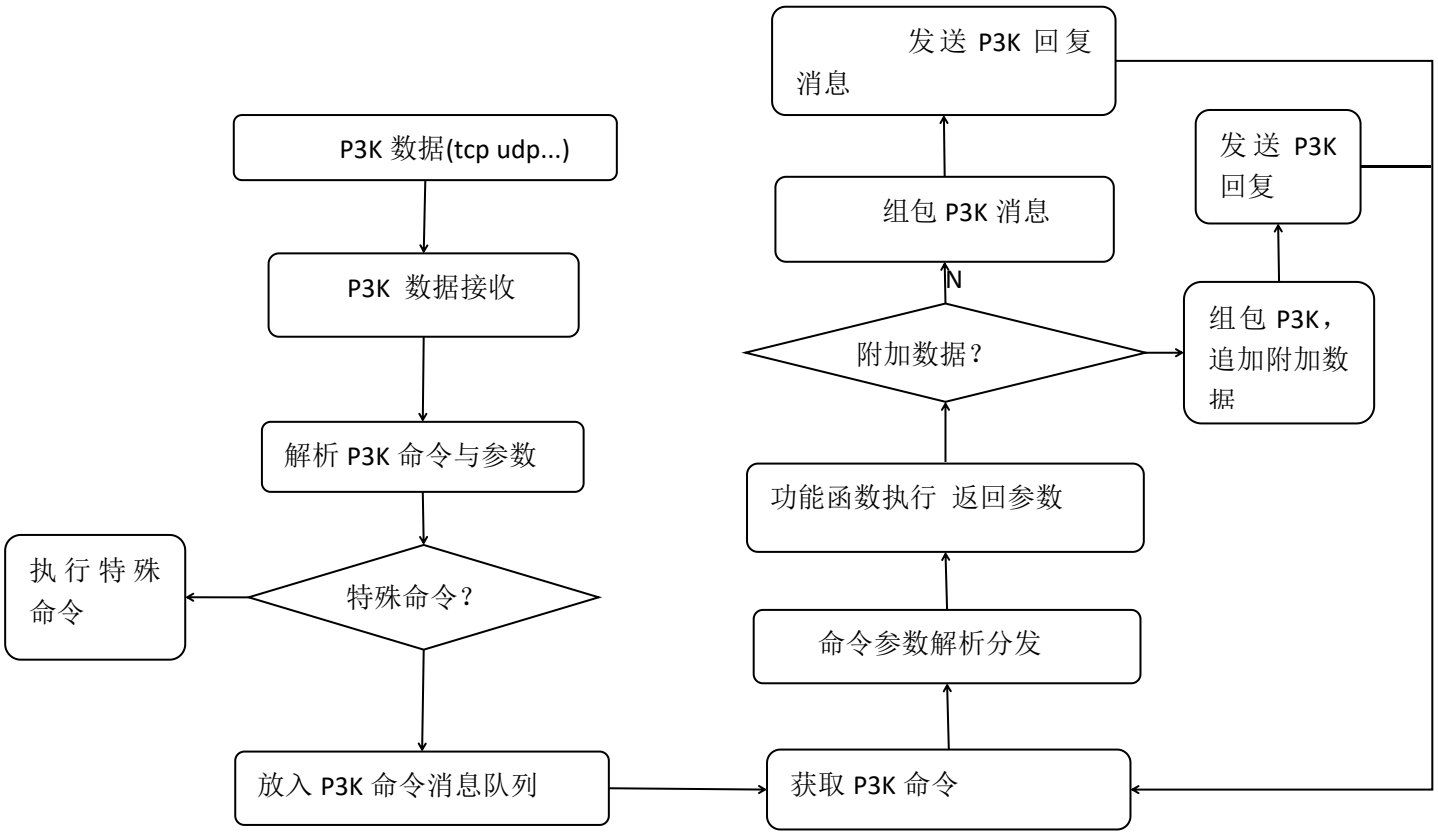
二. 内部模块



三. 模块流程：



四．数据流程图


发送 P3K 回复消息
P3K 数据(tcp udp...)
组包 P3K 消息
发送 P3K 回复
P3K 数据接收
N
附加数据？
组包 P3K，追加附加数据
解析 P3K 命令与参数
功能函数执行 返回参数
执行特殊命令
特殊命令？
命令参数解析分发
放入 P3K 命令消息队列
获取 P3K 命令

五．对外代码功能介绍

>模块初始化接口：

```
/*******************
功 能 :模 块 初 始 化

*******************/
int P3K_ApiInit();
/*******************
```

API 功能:P3K 模块初始化

>模块去初始化接口：

```
/*******************
功 能 :模 块 去 初 始 化

*******************/
int P3K_APIUnInit();
```

API 功能：P3K 模块退出

>数据接收与发送注册数据结构与接口：

```
typedef int (*P3K_MsgSend)(int handleId,char*data, int len); //组 包 好 的 数 据 发 送

typedef struct _P3KApiHandle_S
{
    int handleId; //注 册 获 得 handleId， 不 同 用 户 注 册 返 回 不 同 id最 大 支 持 10个
    int (*P3kMsgRecv)(int handleId,char*data, int len); //接 收 消 息 回 调
    P3K_MsgSend sendMsg;  //发 送 消 息
}P3KApiHandle_S;
```

```
/*******************
功 能 :模 块 注 册 函 数
输 入 参 数 :
handle:注 册 handle
*******************/
int P3K_ApiRegistHandle(P3KApiHandle_S*handle);
/*******************
功 能 :模 块 注 销 函 数
输 入 参 数 :
handle:注 销 handle
```

Api 功能说明：模块初始化后，用户调用注册函数，实现与 P3K 模块 数据消息的互相传递,可支持多用户的注册，最大支持 10 个用户

参数说明：

P3KApiHandle_S:

handleId: 用户 handle，不同用户注册可得到不同 ID

P3KMsgRecv: 对内回调函数，用户可以通过该指针将 P3K 消息传递给 P3K 模块

P3K_MsgSend:回调函数，用户进行回调函数实现，P3K 模块数据通过此回调将回

复消息返回用户

> 接收发送数据注销接口

```
/*********************
功 能 :模 块 注 销 函 数
输 入 参 数 :
handle:注 销 handle
**********************/
int P3K_ApiUnRegistHandle(P3KApiHandle_S*handle);

/*******************
```

API 功能：P3K 模块，用户注销接收与发送

>获取接收命令总数接口

```
/*******************
功 能 :获 取 以 接 收 命 令 消 息 总 数
输 出 参 数 :
sum:命 令 总 数
**********************/

int P3K_ApiGetTotalRecvCmd(int *sum);
```

API 功能：获取 P3K 模块已经接收的消息总数

>获取已处理命令总数接口

```
/*******************
功 能 :获 取 以 处 理 命 令 消 息 总 数
输 出 参 数 :
sum:命 令 总 数
**********************/
int P3K_ApiGetTotalExcuteCmd(int *sum);

#ifdef __cplusplus
```

API 功能：获取 P3K 模块已经处理的消息总数

六. 内部主要功能函数与结构介绍

>解包接口：

```
#define MAX_COMMANDNAME_LEN 32
#define MAX_PARAM_LEN    256
#define MAX_USR_STR_LEN  4*1024
typedef struct _P3K_SimpleCmdInfo_S
{
    char command[MAX_COMMANDNAME_LEN]; //命 令 字 符 串 "KDS-AUD"
    char param[MAX_PARAM_LEN];   //参 数 字 符 串  param1,param2,param3

}P3K_SimpleCmdInfo_S;
```

结构说明： P3K_SimpleCmdInfo_S:

command: p3k 命令

param:参数字符串,及此条命令包含的所有参数

```
/*************************
输 入 参 数
data 接 收 数 据 包 ， 可 能 不 只 一 条 命 令
len 数 据 包 长 度
输 出 值 ：
cmd[] 返 回 命 令 数 组
返 回 值
 int 型 为 解 析 的 命 令 个 数
*************************/
int P3K_SimpleReqPacketUnpack(char*data,int len,P3K_SimpleCmdInfo_S* cmd);
```

Api 功能说明：将接收到的数据机型解包，将字符串变为命令+参数的模式

参数：data,输入的 P3K 字符串

len:输入的字符串长度

cmd：解析出的命令

>组包接口

```
/*************************
输 入 参 数
cmd 返 回 命 令 信 息
输 出 值 ：
dstdata :p3k字 符 串
返 回 值
返 回 字 符 串 长 度
*************************/
int P3K_SimpleRespCmdBurstification(P3K_SimpleCmdInfo_S *cmd,char *dstdata);
```

Api 功能说明：将回复的命令参数组包成为 P3K 消息字符串

参数：

cmd:回复命令参数

dstdata:组包好的 P3K 字符串

>命令解析执行

```
/*************************
输 入 值 ：
cmdreq :命 令 请 求 信 息 回 复 中 param参 数 以 ','号 隔 开
输 出 值 ：
cmdresp:命 令 回 复 信 息
userdata:用 户 附 加 数 据
返 回 值
成 功 返 回 0， 失 败 返 回 -1
*************************/
int P3K_SiImpleReqCmdProcess(P3K_SimpleCmdInfo_S *cmdreq,P3K_SimpleCmdInfo_S *cmdresp,char*userdata);
```

Api 功能说明：对命令进行参数解析与执行，返回命令参数，及附加数据

参数：

cmdreq: 请求命令参数

cmdresp:回复命令参数

userdata:输出附加的数据，像 LOG-TAIL?命令会附带有 log 信息

Api 内部对于命令的分发执行实现：

```
typedef struct _P3K_PhraserToExecute_S
{
    char  *cmd; //P3K指 令
    int (*ParamPhraser)(char*reqparam,char*respParam,char*userdata); //p3k参 数 解 析 处 理
}P3K_PhraserToExecute_S;
```

每条命令对应一个执行函数，执行函数下进行命令参数解析，调用底层功能函数，对返回参数进行参数集组合返回，方便以后命令类型的增加

```
int P3K_SiImpleReqCmdProcess(P3K_SimpleCmdInfo_S *cmdreq,P3K_SimpleCmdInfo_S *cmdresp,char*userdata)
{
    static P3K_PhraserToExecute_S cliFunc[]={
                                        {"KDS-AUD",P3K_SetAudioInputMode},
                                        {"KDS-AUD?",P3K_GetAudioInputMode},
                                        {"X-AUD-LVL",P3K_SetAudLevel},
                                        {"X-AUD-LVL?",P3K_GetAudLevel},
                                        {"X-AUD-DESC?",P3K_GetAudParam},
                                        {"X-AV-SW-MODE",P3K_SetAutoSwitchMode},
                                        {"X-AV-SW-MODE?",P3K_GetAutoSwitchMode},
                                        {"GEDID",P3K_GetEdid},
                                        {"CPEDID",P3K_CopyEdid},
                                        {"EDID-CS",P3K_SetEdidCsMode},
                                        {"EDID-CS?",P3K_GetEdidCsMode},
                                        {"LOCK-EDID",P3K_SetEdidLockMode},
                                        {"LOCK-EDID?",P3K_GetEdidLockMode},
                                        {"HDCP-MOD",P3K_SetHDCPMode},
                                        {"HDCP-MOD?",P3K_GetHDCPMode},
                                        {"HDCP-STAT?",P3K_GetHDCPStatus},
                                        {"VIEW-MOD",P3K_SetVideoWallMode},
                                        {"VIEW-MOD?",P3K_GetVideoWallMode},
                                        {"WND-BEZEL",P3K_SetWndBezel},
                                        {"WND-BEZEL?",P3K_GetWndBezel},
                                        {"VIDEO-WALL-SETUP",P3K_SetVideoWallRotaion},
```