## 上肢关节康复器XY-AI-UPLIMB-II通信协议

---母海亮 2021.6.2

说明：该通信协议适用于上下位机间的通信，该协议主要借鉴开源协议mavlink中的主要思想。

|  |  |
| --- | --- |
| **Index(Hex)** | **Object** |
| 0000 - 00FF | Error Type(紧急ERROR) |
| 0100 - 01FF | 通用控制指令 |
| 1000 - 1FFF | 关节活动度设定 |
| 2000 - 20FF | 助力训练相关 |
| 2100 - 21FF | 主动训练相关 |
| 2200 - 22FF | 被动训练相关 |
| 2300 - 23FF | 记忆训练相关 |
| 3000 - 30FF | 力量评估相关 |
| 3100 - 31FF | 活动度评估相关 |
| F000 | Heartbeat(从心跳包,NoACK) |

* 1. 协议格式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STX | LEN | SEQ | SYSID | MSGID | PAYLOAD | CHECKA | CHECKB |

7~262Byte

1.2协议说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 字节索引 | 内容 | 值 | 说明 |
| STX | 0 | 包的起始位 | 0xFE | 协议帧头 |
| LEN | 1 | 有效数据长度 | 0~255 | 有效数据的字节数（N） |
| SEQ | 2 | 包的序号 | 0~255 | 该包的序号只有待发送数据>255字节时，需要分包发送的情况下才起作用，比如传输文件。正常情况都为0。分包序号1~n |
| SYSID | 3 | 系统ID编号 | 1~255 | 该ID用于区分设备，默认情况下上位机ID为1，下位机ID从2开始递增，游戏ID为10 |
| MSGID | 4 | 协议ID编号 | 0x0000~  0xFFFF | 协议ID，用于判别该包发送的内容，该ID根据具体需求提前约定好 |
| PAYLOAD | 6~N+6 | 有效数据 | 0~255 Byte | 实际数据内容，该内容跟MSGID一一对应，最小数据为0字节，最大数据长度为255字节 |
| CHECKA | N+6 | 校验和低字节 |  | 16位CRC校验（上下位机使用同一套校验函数） |
| CHECKB | N+7 | 校验和高字节 |  |

1.3消息ID定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议命令 | 值 | 流方向 | 数据类型 | 备注 |
| 心跳 | 0x0000 | 上位机<-->下位机 | 无 | 心跳中无实际内容 |
| 自检结果 | 0x0001 | 下位机-->上 位机 | Int8\_t | 0-自检失败 1-自检成功 |
| 设备报警 | 0x0002 | 下位机-->上位机 | Int8\_t | 报警内容待定 |
| 痉挛报警 | 0x0003 | 下位机-->上位机 | Uint8\_t |  |
| 痉挛复位 | 0x0004 | 上位机-->下位机 | 无 |  |
| 模式设定 | 0x0100 | 上位机-->下位机 | Uint8\_t | E\_TRAINMODE，由于枚举长度不定，所以使用uint8传输 |
| 机械零点校准 | 0x1000 | 下位机<-->上位机 | Int8\_t | 1-校准中 2-校准成功 3-校准失败 |
| 选择左右手以及手指 | 0x1001 | 上位机-->下位机 | ST\_HandTypeFingers |  |
| 获取手指位置开关 | 0x1002 | 上位机-->下位机 | Int8\_t 1-on  0-off | 回原点 |
| 回复手指位置信息 | 0x1003 | 下位机-->上位机 | ST\_RealTimeFingersPosition |  |
| 获取手指力量开关 | 0x1004 | 上位机-->下位机 | Int8\_t 1-on  0-off |  |
| 回复手指力量信息 | 0x1005 | 下位机-->上位机 | ST\_RealTimeStrength |  |
| 控制单个手指上下运动 | 0x1006 | 上位机-->下位机 | ST\_FingerUpDownControl |  |
| 验证当前范围 | 0x1007 | 上位机-->下位机 | ST\_DestPosition |  |
| 关节活动设定-锁定按钮 | 0x1008 | 上位机-->下位机 | Int8\_t lock; //  1-锁定 0-解锁 |  |
| 手指运动范围 | 0x1009 | 上位机-->下位机 | ST\_FingersRange |  |
| 发送助力的活动类型 | 0x2000 | 上位机-->下位机 | Int8\_t 1-四指  0-拇指 |  |
| 发送助力训练难易等级 | 0x2001 | 上位机-->下位机 | Uint8\_t 1,2,3代表难易等级 |  |
| 开始助力训练 | 0x2002 | 上位机<-->下位机 | 无 |  |
| 结束助力训练 | 0x2003 | 上位机<-->下位机 | 无 |  |
| 助力模式告知运动到目标位置 | 0x2003 | 上位机-->下位机 | Int8\_t 1-上限 -1-下限 |  |
| 助力模式 下位机回复到达目标位置 | 0x2004 | 下位机--->上位机 | NoData |  |
| 开始主动训练 | 0x2101 | 上位机<-->下位机 | 无 |  |
| 结束主动训练 | 0x2102 | 上位机<-->下位机 | 无 |  |
| 主动训练控制模式 | 0x2103 | 上位机<-->下位机 | Uint8\_t 1-力量模式 2-位置模式 |  |
| 运动手指 | 0x2104 | 上位机<-->下位机 | ST\_SetFingerPosiotion手指下标以及方向 | 上位机告知下位机要运动的手指，下位机回复当前运动的手指 |
| 主动训练难度易等级 | 0x2015 | 上位机<-->下位机 | Uint8\_t 1,2,3代表难易等级 |  |
| 告知下位机给出运动提示 | 0x2016 | 上位机--->下位机 | NoData |  |
| 下位机告知上位机完成运动提示 | 0x2017 | 下位机-->上位机 | NoData |  |
| 设置被动训练的手指 | 0x2200 | 上位机<--->下位机 | ST\_HandTypeFingers |  |
| 被动训练参数 | 0x2201 | 上位机<--->下位机 | ST\_PassiveParam |  |
| 开始被动训练 | 0x2202 | 上位机<-->下位机 |  |  |
| 结束被动训练 | 0x2203 | 上位机<-->下位机 |  |  |
| 开始记忆训练 | 0x2301 | 上位机<-->下位机 | 无 |  |
| 结束记忆训练 | 0x2302 | 上位机<-->下位机 |  |  |
| 控制设置 | 0x2303 | 上位机<-->下位机 | Uint8\_t 1位置模式、 2力的模式 |  |
| 发送训练难易程度 | 0x2304 | 上位机-<->下位机 | Uint8\_t 1,2,3代表难易等级 |  |
| 发送选择屈曲力还是伸展力 | 0x2305 | 上位机<-->下位机 | Uint8\_t 1伸展力2屈曲力 |  |
| 力量超限是否继续训练 | 0x2306 | 上位机<-->下位机 | Uint\_t 1 1继续训练 2退出后重新开始 |  |
| 力量评定开始 | 0x3001 | 上位机<-->下位机 | 无 |  |
| 力量评定结束 | 0x3002 | 上位机<-->下位机 | 无 |  |
| 活动度评定开始 | 0x3101 | 上位机<-->下位机 | 无 |  |
| 活动度评定结束 | 0x3102 | 上位机<-->下位机 | 无 |  |
| 手指的类型 | 0x3103 | 上位机-->下位机 | Uint8\_t 1四指 ，2拇指 |  |
| 伸展力的数值设定 | 0x3104 | 上位机-->下位机 | Int16\_t ExtendForce |  |
| 屈曲力的数值设定 | 0x3105 | 上位机-->下位机 | Int16\_t FlextionForce |  |
| 手指的选择 | 0x3106 | 上位机-->下位机 | Uint8\_t 5个手指， 选中为1，未选为0 |  |
| 力传感器数值标定（开始标定） | 0x4000 | 上位机-->下位机 | 无 |  |
| 力传感器数值标定（停止标定） | 0x4001 | 上位机-->下位机 | 无 |  |
| 数值设定 | 0x4002 | 上位机-->下位机 | ST\_SetForceStander |  |

1.4数据类型定义

//训练模式

Typedef enum

{

E\_TRAINMODE\_PASSIVE = 0, //被动

E\_TRAINMODE\_ACTIVE, //主动

E\_TRAINMODE\_BOOST, //助力

E\_TRAINMODE\_MEMORY //记忆

}E\_TRAINMODE;

//消息ID

Typedef enum

{

ERROR\_EMERGENCY = 0X0000, //急停报警

ERROR\_CHECKRESULT = 0X0001, //自检结果

ERROR\_DEVICEERROR = 0X0002, //设备报警

ERROR\_SPASMWARNING = 0X0003, //痉挛报警

SPASMERRORRESET = 0X0004, //痉挛复位

CONTROLPARAMSET = 0x0100, //模式设定

MACHINEZEROCHECK= 0x1000, //机械零点校准

HANDFINGERSELECTED = 0X1001, //左右手以及手指选择

GETFINGERPOSITION = 0X1002, //获取手指位置 int8\_t 1-开始获取 0-停止获取

RESPONDPOSITION = 0X1003, //回复手指位置信息

GETFINGERFORCE = 0X1004, //获取手指力量大小

RESPONDFORCE = 0X1005, //回复手指力量大小

CONTROLSIGLEFINGER = 0X1006, //控制单个手指上下运动 int8\_t -1向下 1向上

TESTFINGERSPOSITION = 0X1007, //验证当前范围，(1五指朝外 2四指朝内 3拇指朝内)

LOCKFINGERS = 0X1006, //锁定手指 0-解锁 1-锁定

}E\_CMDID;

//关节活动度测试 下位机<--->上位机

//左右手以及手指是否被选中

Typedef struct

{

Int8\_t handType; //0-左手 1-右手

Int8\_t thumb; //0-未被选中 1-被选中

Int8\_t forfinger;

Int8\_t middlefinger;

Int8\_t ringfinger;

Int8\_t littlefinger;

}ST\_HandTypeFingers;

Typedef struct

{

Int8\_t fingerIndex; //手指下标 0-4 拇指到小指

Int8\_t direction; //控制方向 -1向下 1向上

}ST\_FingerUpDownControl;

//实时位置采集

Typedef struct

{

Uint8\_t thumb;

Uint8\_t forefinger;

Uint8\_t middlefinger;

Uint8\_t ringfinger;

Uint8\_t littlefinger;

}ST\_RealTimeFingersPosition;

//实时力量采集

Typedef struct

{

Int16\_t thumb;

Int16\_t forefinger;

Int16\_t middlefinger;

Int16\_t ringfinger;

Int16\_t littlefinger;

}ST\_RealTimeStrength;

//运动目标位置

Typedef struct

{

Uint8\_t thumb;

Uint8\_t forefinger;

Uint8\_t middlefinger;

Uint8\_t ringfinger;

Uint8\_t littlefinger;

}ST\_DestPosition;

//单个手指关节位置调整 上位机---->下位机

Typedef struct

{

Uint8\_t fingerIndex; //0~5 依次为左拇指、右拇指、食指、中指、无名指、小指

Int8\_t flag; //-1向下 1向上

}ST\_SetFingerPosiotion;

//被动训练速度力量设定

Typedef struct

{

Int8\_t speed; //0-5设置速度档位

Int8\_t strength; //0-5 设置力量档位

}ST\_StrengthAndSpeed;

//主动训练和记忆训练中力量设定

Typedef struct

{

Int8\_t buckling;//屈曲力等级 1~3

Int8\_t stretch;//伸展力等级 1~3

Int8\_t controlType;//控制方式 1~趋势 2~位移

}ST\_ControlSet;

//被动训练

//被动训练参数

Typedef struct

{

Int8\_t speed;//速度

Uint8\_t jointScorp;//关节活动度

Uint8\_t strength;//力量极限

}ST\_PassiveParam;

//电机位置校准

Typedef struct

{

Uint16\_t Lthumb;

Uint16\_t Rthumb;

Uint16\_t forefinger;

Uint16\_t middlefinger;

Uint16\_t ringfinger;

Uint16\_t littlefinger;

}ST\_FingerMotionPosition;

//称重传感器校准

Typedef struct

{

Uint16\_t Lthumb;

Uint16\_t Rthumb;

Uint16\_t forefinger;

Uint16\_t middlefinger;

Uint16\_t ringfinger;

Uint16\_t littlefinger;

}ST\_FingerWeightSensor;

//助力模式最大静摩擦力标定

Typedef struct

{

Uint16\_t Lthumb;

Uint16\_t Rthumb;

Uint16\_t forefinger;

Uint16\_t middlefinger;

Uint16\_t ringfinger;

Uint16\_t littlefinger;

}ST\_ForceOfFriction;

//各手指的运动范围

typedef struct

{

int8\_t fingersRange[10];//[0]拇指min [1]拇指max [2]食指min [3]食指max ...

}ST\_FingersRange;

//手指力量标定 上位机---->下位机

Typedef struct

{

Uint8\_t fingerIndex; //0~5 依次为左拇指、食指、中指、无名指、小指、右拇指

uint8\_t force; //力量值N

}ST\_SetForceStander;

补充：CRC 校验说明

buf从针头开始，除了校验和之外的所有数据都参与校验

void **Pressure\_CheckCRC**(uint8\_t\*buf,int len,uint8\_t\* CRC\_H,uint8\_t\* CRC\_L)

{

uint16\_t i,j,tmp,CRC16;

CRC16=0xffff;

for (i=0;i<len;i++)

{

CRC16=\*buf^CRC16;

for (j=0;j< 8;j++)

{

tmp=CRC16 & 0x0001;

CRC16 =CRC16 >>1;

if (tmp)

CRC16=CRC16 ^ 0xA001;

}

\*buf++;

}

CRC\_H[0]=CRC16>>8;

CRC\_L[0]=CRC16&0xff;

}

/\*上位机发送消息\*/

typedef enum

{

e\_HEARTBEAT = 0X0000, //心跳

e\_CHECKRESULT = 0X0001, //自检结果

e\_ERROR\_DEVICEERROR = 0X0002, //设备报警

e\_ERROR\_SPASMWARNING = 0X0003, //痉挛报警

e\_SPASMERRORRESET = 0X0004, //痉挛复位

e\_SETTRAINMODE = 0X0100, //设置模式

e\_CALIBRATION\_ZERO = 0x1000, //机械零点校准

e\_SELECT\_HAND\_FINGER = 0X1001, //选择左右手以及手指

e\_GET\_POSITION\_CMD = 0X1002, //获取手指位置开关

e\_SEND\_POSITION\_CMD = 0X1003, //回复手指位置信息

e\_GET\_STRENGTH\_CMD = 0X1004, //获取手指力量开关

e\_SEND\_STRENGTH = 0X1005, //回复手指力量信息

e\_CTRL\_SINGLE\_FEINGER = 0X1006, //控制单个手指上下运动

e\_Motion\_RANGE\_TEST = 0X1007, //验证当前范围

e\_MOTION\_RANGE\_SETTING\_LOCK = 0X1008,//关节活动设定-锁定按钮

e\_ASSISTANCE\_MODE\_FINGER\_SELECT = 0x2000,//是四指还是拇指

e\_ASSISTANCE\_MODE\_HELP\_TYPE = 0X2001,//助力的等级 1简单、2中等、3困难

e\_START\_ASSISTANCE\_MODE = 0X2002,//开始助力训练

e\_CLOSE\_ASSISTANCE\_MODE = 0X2003,//结束助力训练

e\_START\_ACTIVE\_TRANNING\_MODE = 0X2101, //开始主动训练模式

e\_CLOSE\_ACTIVE\_TRANNING\_MODE = 0X2102, //结束主动训练模式

e\_ACTIVE\_TRANNING\_MODE\_CTRL\_TYPE = 0X2103, //1位置模式，2力量模式、

e\_MOVE\_FINGER\_Index = 0x2104, //动哪个手指选择哪个，并返回。

e\_ACTIVE\_TRANNING\_MODE\_HELP\_TYPE = 0X2105, //主动训练模式的难易等级 1简单2中等3困难

e\_SELECT\_PASSIVE\_TRANNING\_FINGER = 0X2200, //选择

e\_PASSIVE\_TRANNING\_MODE\_PARAMETER = 0X2201, //被动训练参数

e\_START\_PASSIVE\_TRANNING\_MODE = 0X2202, //开始被动训练

e\_CLOSE\_PASSIVE\_TRANNING\_MODE = 0X2203, //结束被动训练

e\_START\_MEMORY\_TRANNING\_MODE = 0X2301, //开始记忆训练

e\_CLOSE\_MEMORY\_TRANNING\_MODE = 0X2302, //结束记忆训练

e\_MEMORY\_MODE\_CTRL\_MODE = 0X2303, //1位置模式，2力的模式

e\_MEMORY\_MODE\_HELP\_TYPE = 0X2304, //训练的难易程度

e\_MEMORY\_MODE\_FORCE\_TYPE = 0X2305, //记忆训练力的类型，1伸展力，2屈曲力

e\_MEMORY\_MODE\_TRANNING\_CYCLE = 0X2306, //当力超限时，是否继续运行。

e\_START\_FORCE\_ASSESS\_MODE = 0X3001, //力量评估模式开始

e\_CLOSE\_FORCE\_ASSESS\_MODE = 0X3002, //力量评估模式结束

e\_START\_MOTION\_RANG\_JUDGE = 0X3101, //开始关节活动度评定

e\_CLOSE\_MOTION\_RANG\_JUDGE = 0X3102, //结束关节活动度评定。

e\_MOTION\_RANG\_JUDGE\_FINGER\_TYPE = 0x3103, //手指类型，四指或者拇指

e\_MOTION\_RANG\_JUDGE\_EXTEND\_FORCE = 0x3104, //伸展力数值

e\_MOTION\_RANG\_JUDGE\_FLEXION\_FORCE =0x3105//屈曲力数值

}

E\_COMM\_TO\_PC ;