

# kyLink

V0.1.0

kyChu@2019/03/05

kylink 是为机器人各嵌入式模块之间及上、下位机之间通讯设计的通讯协议，为机器人内部通讯提供一种精简高效的通讯协议。

kylink 消息帧结构

kylink 以数据帧为传输单元，数据长度为 9-256 字节，结构如下：

STX1 (1B)	STX2 (1B)	DevID (1B)	MsgID (1B)	Length (2B)	PayLoad (1..248B)	CRC16 (2B)
--------------	--------------	---------------	---------------	----------------	----------------------	---------------

名字	索引	描述	值	大小(Byte)
STX1	0	帧识别头 1	0x55	1
STX2	1	帧识别头 2	0xAA	1
DevID	2	设备 ID	由模块定义	1
MsgID	3	消息 ID	数据消息号	1
Length	4	缓存区数据字节数	n(n>0)	2
Payload	6	缓存区	Data[n]	n
CRC16	6+n	范围：DevID+MsgID+Length+Payload	CRC16	2

STX1/STX2:

帧头，用于通讯同步，定义为 0x55 0xAA

DevID(根据硬件扩充):

0x00	未指定
0x01	上位机
0x02	底盘驱动板
0x03	GPS+IMU 导航模块
0x04	IMU 惯导模块
0x05	传感器模块 CAN 转接板（串口转 CAN）
0x06	超声波测距模块
...	...

MsgID(不同设备对应不同消息列表):

0x00	通讯测试
0x01	心跳消息
0x02	kylink 协议版本号请求
0x03	kylink 协议版本号应答
0x04	kylink 协议名称应答
...	... 根据设备定义

CRC16 校验算法(C) :

```
uint16_t do_crc_table(uint8_t *ptr, uint32_t len) {
    unsigned short int crc = 0;
    uint8_t H8;
    while(len -- != 0) {
        H8 = (unsigned short)crc >> 8; crc <=< 8;
        crc ^= crcTab16[H8 ^ *ptr ++];
    }
    return(crc);
}
```

CRC16 数据表:

0x0000,	0x1021,	0x2042,	0x3063,	0x4084,	0x50a5,	0x60c6,	0x70e7,
0x8108,	0x9129,	0xa14a,	0xb16b,	0xc18c,	0xd1ad,	0xe1ce,	0xf1ef,
0x1231,	0x0210,	0x3273,	0x2252,	0x52b5,	0x4294,	0x72f7,	0x62d6,
0x9339,	0x8318,	0xb37b,	0xa35a,	0xd3bd,	0xc39c,	0xf3ff,	0xe3de,
0x2462,	0x3443,	0x0420,	0x1401,	0x64e6,	0x74c7,	0x44a4,	0x5485,
0xa56a,	0xb54b,	0x8528,	0x9509,	0xe5ee,	0xf5cf,	0xc5ac,	0xd58d,
0x3653,	0x2672,	0x1611,	0x0630,	0x76d7,	0x66f6,	0x5695,	0x46b4,
0xb75b,	0xa77a,	0x9719,	0x8738,	0xf7df,	0xe7fe,	0xd79d,	0xc7bc,
0x48c4,	0x58e5,	0x6886,	0x78a7,	0x0840,	0x1861,	0x2802,	0x3823,
0xc9cc,	0xd9ed,	0xe98e,	0xf9af,	0x8948,	0x9969,	0xa90a,	0xb92b,
0x5af5,	0x4ad4,	0x7ab7,	0x6a96,	0x1a71,	0x0a50,	0x3a33,	0x2a12,
0xdbfd,	0xcdbc,	0xfbbf,	0xeb9e,	0x9b79,	0x8b58,	0xbb3b,	0xab1a,
0x6ca6,	0x7c87,	0x4ce4,	0x5cc5,	0x2c22,	0x3c03,	0x0c60,	0x1c41,
0xedae,	0xfd8f,	0xcdec,	0xddcd,	0xad2a,	0xbd0b,	0x8d68,	0x9d49,
0x7e97,	0x6eb6,	0x5ed5,	0x4ef4,	0x3e13,	0x2e32,	0x1e51,	0x0e70,
0xff9f,	0xefbe,	0xdfdd,	0xcffc,	0xbf1b,	0xaf3a,	0x9f59,	0x8f78,
0x9188,	0x81a9,	0xb1ca,	0xa1eb,	0xd10c,	0xc12d,	0xf14e,	0xe16f,
0x1080,	0x00a1,	0x30c2,	0x20e3,	0x5004,	0x4025,	0x7046,	0x6067,
0x83b9,	0x9398,	0xa3fb,	0xb3da,	0xc33d,	0xd31c,	0xe37f,	0xf35e,
0x02b1,	0x1290,	0x22f3,	0x32d2,	0x4235,	0x5214,	0x6277,	0x7256,
0xb5ea,	0xa5cb,	0x95a8,	0x8589,	0xf56e,	0xe54f,	0xd52c,	0xc50d,
0x34e2,	0x24c3,	0x14a0,	0x0481,	0x7466,	0x6447,	0x5424,	0x4405,
0xa7db,	0xb7fa,	0x8799,	0x97b8,	0xe75f,	0xf77e,	0xc71d,	0xd73c,
0x26d3,	0x36f2,	0x0691,	0x16b0,	0x6657,	0x7676,	0x4615,	0x5634,
0xd94c,	0xc96d,	0xf90e,	0xe92f,	0x99c8,	0x89e9,	0xb98a,	0xa9ab,
0x5844,	0x4865,	0x7806,	0x6827,	0x18c0,	0x08e1,	0x3882,	0x28a3,
0xcb7d,	0xdb5c,	0xeb3f,	0xfb1e,	0x8bf9,	0x9bd8,	0xabbb,	0xbb9a,
0x4a75,	0x5a54,	0x6a37,	0x7a16,	0x0af1,	0x1ad0,	0x2ab3,	0x3a92,
0xfd2e,	0xed0f,	0xdd6c,	0xcd4d,	0xbdaa,	0xad8b,	0x9de8,	0x8dc9,
0x7c26,	0x6c07,	0x5c64,	0x4c45,	0x3ca2,	0x2c83,	0x1ce0,	0x0cc1,
0xef1f,	0xff3e,	0xcf5d,	0xdf7c,	0xaf9b,	0xbfbfa,	0x8fd9,	0x9ff8,
0x6e17,	0x7e36,	0x4e55,	0x5e74,	0x2e93,	0x3eb2,	0x0ed1,	0x1ef0,

固件升级协议：

(1) 流程：

- S1：主机发送固件升级请求数据包，等待设备应答；
- S2：设备接收到升级请求后进入固件升级模式，检查硬件及固件状态，判断是否可以升级并反馈给主机；
- S3：主机根据反馈进行判断，若不可升级则退出，若可升级则继续；
- S4：设备进入升级状态后开始执行擦除流程，并返回擦除进度；
- S5：设备擦除完毕后定时向主机请求数据包；
- S6：主机根据请求的数据包 ID 准备数据并定时向设备发送；
- S7：循环执行至固件数据全部发送完毕；
- S8：升级完成后，主机退出，设备跳转至应用程序执行。

(2) 消息：

0x80	固件升级请求	主机请求
0x81	固件升级数据包	主机返回
0x82	升级状态包	模块返回

(3) Payload 数据包：

1, 固件信息 (主机请求包)

```
struct {
    FW_TYPE FW_Type;
    FW_ENC_TYPE Enc_Type;
    uint32_t PacketNum;
    uint32_t FileSize;
    uint16_t FW_Version;
    uint32_t FileCRC;
} FWInfoDef;
```

2, 固件数据 (主机应答包)

```
struct {
    uint32_t PacketID;
    uint8_t PacketLen;
    uint8_t PacketData[FILE_DATA_CACHE]; // FILE_DATA_CACHE = 80
} UpgradeDataDef;
```

3, 设备状态 (设备应答包)

```
struct {
    DevpgradeState Dev_State;
    uint8_t reserve[5];
} DevResponseDef;
```

4, 加密类型

0	明文
1	AES/ECB
...	...

5,设备类型 ID:(FW\_TYPE)

0	未知固件
1	驱动板固件
2	IMU 模块固件
3	IMU+GPS 模块固件
4	超声波测距模块固件

6,设备升级状态:

0	就绪状态
1	擦除中
2	升级中(请求数据包)
3	拒绝升级请求
4	跳转应用失败
...	...

IMU 惯导模块通讯协议:

设备 ID(DevID): 0x04

消息列表(MsgID):

0x08	IMU 模块输出请求	主机请求
0x11	IMU 原始数据(3 轴加速度+3 轴陀螺仪数据)	模块返回
0x12	模块姿态四元数(qw, qx, qy, qz)	模块返回
0x20	模块设置	主机请求
0xC0-0xEF	工程应用,不开放	

Payload:

1,MsgID = 0x08:

主机向模块发送该请求后,模块开始往外发送 IMU 数据,否则无数据输出,请求数据固定为: 0x6B, 0x79, 0x43, 0x68, 0x75, 0x40, 0x49, 0x4D, 0x55

2,MsgID = 0x11:

IMU 6 轴数据,经过校准.(注: float:4B)

```
union {
    struct {
        float accX, accY, accZ, gyrX, gyrY, gyrZ;
    } Raw;
    uint8_t data[24];
} IMU_RAW;
```

3,MsgID = 0x12:

模块姿态数据,四元数格式:

```
union {
    struct {
        float qw, qx, qy, qz;
    } Raw;
    uint8_t data[16];
} IMU_Q;
```

4,MsgID = 0x80, 0x81, 0x81, 支持固件升级协议。

# 其他设备...