KL 硬件产品MCU与Win Host通信协议

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 完成日期 | 备注 |
| V1.1 | Lbs | 2018-07-10 | 适用于FiuNv |

# 软硬件系统

FIU故障注入箱 : f429 + ucosii

# 通信方式

UDP通信

默认 ip 192.168.1.100

端口 50099

2/

逻辑说明：

1、初始状态：对应管脚电平为 低

2、IP等可更改。

# 通信协议

(hex)，Nv1.0.0版mcu侧 协议长度支持最大 128

控制命令

be be be be be be be be cmd len (内容) crc ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

总长度 = 8 + 1 + 1 + data + 1 + 8 + 8 = 27+len

len = 内容的字节数

crc = 0 (fiu忽略 crc 的校验)

ff 保留字

mcu将 各内容数据加 1 返回

配置命令，修改网络信息等（适用于 KL MaC）

be be be be be be be be cmd len (内容) crc (12 byte’s MCU UID) ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

总长度 = 8 + 1 + 1 + data + 1 + 12 + 8 + 8 = 39 + len

len = 内容的字节数

crc = 内容的和校验

ff 保留字

## 控制命令

ff ： mcu保留，不可用

00 ： 通信连接测试

-> be be be be be be be be 00 08 01 02 03 04 05 06 07 08 24 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

<- be be be be be be be be 00 08 02 03 04 05 06 07 08 09 2c ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

06 用于控制故障注入箱继电器

假如一包数据

-> be be be be be be be be 06 08 01 01 01 01 01 01 01 01 08 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

<- be be be be be be be be 06 08 (8bytes依命令而定) 08 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

## 配置命令

c8 : IP\_PORT 更改操作（第 10 11 12 13 个数据依次代表IP地址,第 14 15 个数据依次代表 端口低8位 高8位,掩码，网关），UDP组播向 235.236.237.238 45654 处发送

-> be be be be be be be be c8 0e c0 a8 01 c8 20 4e Ff Ff Ff 0 c0 a8 01 01 06 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

<- 暂无

c9 ：可查询 MCU 的 96位UID，UDP单播

-> be be be be be be be be c9 0c 00 00 00 00 00 00 00 00 00 00 00 00 00 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

<- be be be be be be be be c9 0c xx xx xx xx xx xx xx xx xx xx xx xx xx ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

ca ：对设备 MCU 进行软复位，UDP单播

-> be be be be be be be be ca 00 00 ff ff ff ff ff ff ff ff ed ed ed ed ed ed ed ed

<- 暂无

4、

MCU每隔大概1s会向组播地址 225.226.227.228 端口54345 处发送设备信息，

结构体格式如下

#pragma pack(1)

typedef struct mcuinfo {

u16 devType;//设置类型标识，0x0100电源类 0x0200故障类 0x0300高压类

u16 port;//端口

s8 devName[32]; //设备名称

s8 devSN[16]; //设备SN号，暂无，待公司生产正规化后加入

s8 mcuType[16]; //芯片类型

u32 mcuUID[3]; //芯片UID 标识

u8 mcuSV[3]; //软件版本号，[0]:主版本 [1]:次版本 [2]:debug版本

u8 mcuIP[4]; //ip

u8 mcuNetMask[4]; //掩码

u8 mcuGW[4]; //网关

u8 mcuMAC[6]; //网络初始化时使用的mac地址

u8 mcuHV[2]; //母板硬件版本 [0]主版本 [1]次版本

u8 reserve[1]; //保留后续扩展

}McuInfo\_t;

#pragma pack()

be be be be be be be be aa aa aa aa aa aa aa aa len packettype struct\_mcuinfo crc 55 55 55 55 55 55 55 55 ed ed ed ed ed ed ed ed

8 + 8 + 2 + 1 + len + 1 + 8 + 8

具体数据包格式

for (i=0;i<8;i++) {

tmp[i] = 0xbe;

}

for (i=8;i<16;i++) {

tmp[i] = 0xaa;

}

len = sizeof(struct mcuinfo);

tmp[16] = len&0xff;

tmp[17] = len>>8;

tmp[18] = MCUPACKETTYPE;//packettype:0x01,设备发送mcu信息

g\_MCUinfo.devType = DEVTYPECODE;

g\_MCUinfo.port = g\_LocalUdpPort;

memset(g\_MCUinfo.devName,'\0',32);

memset(g\_MCUinfo.devSN,'\0',16);

memset(g\_MCUinfo.mcuType,'\0',16);

strcpy(g\_MCUinfo.devName ,DEVNAME);

strcpy(g\_MCUinfo.devSN,DEVSN);

strcpy(g\_MCUinfo.mcuType,MCUTYPE);

g\_MCUinfo.mcuSV[0] = MCUSV\_MAJOR;

g\_MCUinfo.mcuSV[1] = MCUSV\_MINOR;

g\_MCUinfo.mcuSV[2] = MCUSV\_DEBUG;

g\_MCUinfo.mcuHV[0] = MCUHV\_MAJOR;

g\_MCUinfo.mcuHV[1] = MCUHV\_MINOR;

memcpy(&tmp[19],&g\_MCUinfo,len);

for (i=19;i<(19+len);i++) {

crc += tmp[i];

}

tmp[i++] = crc;

for (j=i;j<(i+8);j++){

tmp[j] = 0x55;

}

for (j=i+8;j<(i+8+8);j++){

tmp[j] = 0xed;

}

---------------------- end at here 201800710 -------------------------------------------------------------