**线端检测器通信协议**

**（V0.2）**

1. **通信模式**

通信中线端检测器作为服务器，PC端作为客户端。

检测器服务的信息如下：TCP port = 2016，该端口号不可改变。

检测器同时仅仅允许一台PC进行远程操作。

检测器的网络地址有2种方式进行设置：1）DHCP 2）静态IP4地址

通信过程中采用应答方式。PC给检测器发送操作请求，检测器操作完成后检测器返回结果。通信过程中不予许同时执行多条命令。

1. **通信协议**
   1. **通用命令格式**

PC发往检测器的通用命令格式如下：

typedefstruct \_tag\_command\_request{

uint8\_t u8\_cmd\_id; //命令ID，见上文CMD\_ID\_T中的定义

uint8\_t u8\_para1; //下面的4个成员变量用于传递参数，具体

uint16\_t u16\_para2; //的用法详见每个命令

uint32\_t u32\_para3;

char s\_buffer[20];

}command\_request\_t;

PC发送的数据采用该结构体的二进制传输，固定长度。CPU字节序为小端。

若0x04030201 的32位整数在网络上的传送顺序为：01 02 03 04

若0x02的16位整数在网络上的传送顺序为：02 00

检测器返回的数据如下：

typedef struct \_tag\_command\_result\_t{

uint8\_t u8\_is\_ok; //1 - OK, 0 - FAIL

uint8\_t u8\_result1;

uint16\_t u16\_result2;

uint32\_t u32\_append\_data\_size;

}command\_result\_t;

其中，u32\_append\_data\_size表示该数值后是否还有详细结果的信息。

若为0，表示没有；其他值，返回检测器详细信息，具体参见各命令。

* 1. **CMD\_IS\_NET\_ACTIVE**

u8\_cmd\_id = 255（0xFF）

返回：

若检测器正常，返回如下的数值：

p\_info->result.u8\_is\_ok = 1;

p\_info->result.u8\_result1 = 255;

p\_info->result.u16\_result2 = 2;

p\_info->result.u32\_append\_data\_size = 0x04030201;

此时u32\_append\_data\_size值无效，该结构后面没有其他任何数据，该数据可用于判断与检测器相连的PC是否小端传输

* 1. **CMD\_SELF\_TEST**

u8\_cmd\_id = 1（0x01）

返回：

若自检通过，返回如下的数值：

p\_info->result.u8\_is\_ok = 1;

p\_info->result.u8\_result1 = 0;

p\_info->result.u16\_result2 = 0;

p\_info->result.u32\_append\_data\_size = 0;

自检失败，返回：

p\_info->result.u8\_is\_ok = 0;

p\_info->result.u8\_result1 = 0;

p\_info->result.u16\_result2 = 0;

p\_info->result.u32\_append\_data\_size = \*\*\*\*;

后续传输的数据结构如下：

#define CPLD\_BOARD\_COUNT 32

#define CPLD\_SINGLE\_LINE\_COUNT 128

typedef struct \_\_tag\_short\_circuit\_info\_t{

u8 u8\_cpld\_count; //1

u8 u8\_reserved[3]; //3

u16 u16\_short\_circuit\_group\_count; //2

u16 u16\_short\_lines\_count; //2

board\_lines\_state\_t ary\_cpld\_state[CPLD\_BOARD\_COUNT]; //bcount 32\*

}short\_circuit\_info\_t;

typedef struct \_\_tag\_line\_addr\_t{

u8 u8\_board\_addr;

u8 u8\_index\_on\_board;

}line\_addr\_t;

typedef struct \_\_tag\_board\_lines\_state\_t{

u8 u8\_board\_addr; //board and point

u8 u8\_reserved[3]; //4

u8 ary\_u8\_checked\_flag[CPLD\_SINGLE\_LINE\_COUNT/8]; //16

line\_addr\_t ary\_short\_circuit\_info[CPLD\_SINGLE\_LINE\_COUNT]; //256

}board\_lines\_state\_t;

* 1. CMD\_SAMPLE\_NORMAL

功能：实现检测器正常采样。

参数：无，全部传空

返回：1 –成功；0 –失败

该命令返回成功后：

1. 使用CMD\_SAMPLE\_NORMAL\_SAVE命令保存该采样数据到机器的内置的SD卡中。
2. 使用CMD\_GET\_DATA\_SAMPLE\_NORMAL命令远程获取该采样数据。
   1. CMD\_SAMPLE\_NORMAL\_SAVE

功能：将当前普通采样的数据保存至内置的SD卡中。

参数：u16\_para2 –存储位置数据范围 1 – 999

s\_buffer –样品的名称，必须为英文、数字和可见字符的组合，长度最大12

u8\_para1 –样品名称的长度

返回：1 –成功；0 –失败

该命令必须在CMD\_SAMPLE\_NORMAL执行后方可调用，否则出错。

* 1. CMD\_GET\_DATA\_SAMPLE\_NORMAL

功能：获取SD卡中普通采样的详细数据。

参数：u16\_para2 –存储位置数据范围 1 – 999

返回：样品数据。

返回的数据如下结构体：

#define CPLD\_BOARD\_COUNT 32

#define CPLD\_SINGLE\_LINE\_COUNT 128

#define LINES\_SAMPLE\_NAME\_LEN 12

typedefstruct \_\_tag\_short\_circuit\_info\_t{

u8 u8\_cpld\_count;

u8 u8\_reserved[3];

u16 u16\_short\_circuit\_group\_count;

u16 u16\_short\_lines\_count;

board\_lines\_state\_tary\_cpld\_state[CPLD\_BOARD\_COUNT];

}short\_circuit\_info\_t;

typedefstruct \_\_tag\_point\_sample\_info\_t{

u8 u8\_is\_order\_sample;

u8 u8\_reserved;

u16 u16\_point\_count;

line\_addr\_tary\_lines[CPLD\_BOARD\_COUNT \* CPLD\_SINGLE\_LINE\_COUNT];

u8 ary\_flags\_passed[CPLD\_BOARD\_COUNT \* CPLD\_SINGLE\_LINE\_COUNT/8];

}point\_sample\_t;

#define CPLD\_SAMPLE\_TYPE\_NORMAL 1 //普通采样

#define CPLD\_SAMPLE\_TYPE\_POINT 2 //点采样

#define CPLD\_SAMPLE\_TYPE\_GROUP 3 //分组采样

typedefstruct \_\_tag\_lines\_sample\_info\_t{

char s\_sample\_name[LINES\_SAMPLE\_NAME\_LEN+2]; //样品名称

u8 u8\_sample\_type; //样品的类型，SAMPLE\_TYPE定义

u8 u8\_board\_count; //CPLD检测板的数量

union

{

short\_circuit\_info\_tinfo\_short\_circuit; //普通采样的数据

point\_sample\_tinfo\_point; //点采样的数据

}\_sample\_data;

}lines\_sample\_info\_t;

* 1. CMD\_DETECT\_NORMAL

功能：执行普通检测

参数：u16\_para2 –检测所使用的样本编号

返回：1 –成功；0 –失败

该命令将使得检测器执行检测功能，根据当前配置的不同，进行手动、等待OK、延时检测等待功能。该命令的返回成功仅仅表示加载检测样本的成功，并不代表检测成功。

PC端在进行检测前应先获取当前检测器的配置，确认当前处于哪一种检测状态，然后再根据返回的数据进行相应的处理。

**手动检测：**

PC ->CMD\_DETECT\_NORMAL 检测器

PC <- 成功 检测器

PC <- 检测结果 检测器 //检测结果见后续定义

若成功，检测器等待线端拔出后，通知PC，通知的格式为32位整数数值为2

本次检测完成后需要PC再次发送CMD\_DETECT\_NORMAL命令启动下一次检测。检测器此时不记录检测成功与否的次数。PC应根据检测器的返回进行统计。

检测结果的返回：若当前检测结果与样本一致，检测器返回32位整数 1，否则返回32位整数的最大值（0XFFFFFFFF）。PC收到的返回结果若为1，此时应该继续等待线端拔出的通知，该数值为2，收到后表示这一次检测的完成。若收到的返回结果为0xFFFFFFFF，则应该继续接收此时线端检测状态信息，该信息由如下结构体表示：

typedefstruct \_\_tag\_board\_lines\_state\_t{

u8 u8\_board\_addr;

u8 u8\_reserved[3];

u8 ary\_u8\_checked\_flag[CPLD\_SINGLE\_LINE\_COUNT/8];

line\_addr\_tary\_short\_circuit\_info[CPLD\_SINGLE\_LINE\_COUNT];

}board\_lines\_state\_t;

typedefstruct \_\_tag\_short\_circuit\_info\_t{

u8 u8\_cpld\_count;

u8 u8\_reserved[3];

u16 u16\_short\_circuit\_group\_count;

u16 u16\_short\_lines\_count;

board\_lines\_state\_tary\_cpld\_state[CPLD\_BOARD\_COUNT];

}short\_circuit\_info\_t;

**OK等待：**

PC ->CMD\_DETECT\_NORMAL 检测器

PC <- 成功 检测器

PC <- 0xFFFFFFFF, 检测的当前结果 检测器

PC <- 1（样本与结果一致） 检测器

PC <- 2（所有线端被拔出） 检测器

**延时检测：**流程与手动检测相同，只是在收到第一次成功后等待n秒后开始检测。

PC ->CMD\_DETECT\_NORMAL 检测器

PC <- 成功 检测器

PC 等待n秒

PC <- 检测结果 检测器 //检测结果见后续定义

若成功，检测器等待线端拔出后，通知PC，通知的格式为32位整数数值为2

* 1. CMD\_SAMPLE\_POINT

功能：实现检测器点检测采样。

参数：无，全部传空

返回：1 –成功；0 –失败

1）该命令返回成功后，PC端应该进入等待状态，接收来自点取样过程中的点的信息：

检测器会将每次发现的线端信息发送给PC，其数据格式如下：

typedefstruct \_\_tag\_line\_addr\_t{

u8 u8\_board\_addr;

u8 u8\_index\_on\_board;

}line\_addr\_t;

检测器返回给PC的点信息有可能之前已经取样了，PC端应该处理这个异常。

当检测器在取样过程中发现有2个或2个以上的点被检测到，此时返回给PC的数据仍为该结构体，但其中的数据为{0xFF, 0xFF}，是非法值。此时PC端应该进行报警，表示点取样错误。

2）PC端收到检测器返回的数据后应发送数据通知检测器继续点采样还是结束采样，当发送32位的0x00000000表示继续采样，发送0xFFFFFFFF表示停止采样。

3）PC端继续发送CMD\_SAMPLE\_POINT\_SAVE表示将当前采样进行存储。

* 1. CMD\_SAMPLE\_POINT\_SAVE

功能：将当前点检测采样的数据保存至内置的SD卡中。

参数：u16\_para2 –存储位置数据范围 1 – 999

s\_buffer –样品的名称，必须为英文、数字和可见字符的组合，长度最大12

u8\_para1 –样品名称的长度

返回：1 –成功；0 –失败

该命令必须在2.6 CMD\_SAMPLE\_POINT执行后方可调用，否则出错。

* 1. CMD\_GET\_DATA\_SAMPLE\_POINT

功能：获取SD卡中点检测采样的详细数据。

参数：u16\_para2 –存储位置数据范围 1 – 999

返回：样品数据。

参见CMD\_GET\_DATA\_SAMPLE\_NORMAL命令。

* 1. CMD\_DETECT\_POINT

功能：执行点检测

参数：u16\_para2 –检测所使用的样本编号

u8\_para1 –是否按顺序进行点检测。1 表示顺序，2表示不按顺序

返回：1 –成功；0 –失败

流程同CMD\_SAMPLE\_POINT

* 1. CMD\_GET\_DATA\_CONFIG

功能：获取当前检测器所有配置

参数：无

返回：配置信息的二进制数组。详见定义。

#define START\_STYLE\_ADDRESS 0x00 //开始方式地址

#define OK\_SOUND\_ADDRESS 0x01 //OK音样式地址

#define NG\_SOUND\_ADDRESS 0x02 //NG音样式地址

#define IP\_ADDRESS\_0 0x03 //IP0段储存地址

#define IP\_ADDRESS\_1 0x04 //IP1段储存地址

#define IP\_ADDRESS\_2 0x05 //IP2段储存地址

#define SERVER\_IP\_ADDRESS\_3 0x06 //服务器端IP3段储存地址

#define LOCAL\_IP\_ADDRESS\_3 0x07 //客户端端IP3段储存地址

#define DHCP\_SWITCH\_ADDRESS 0x08 //DHCP功能开关储存地址

#define PRINTER\_SWITCH\_ADDRESS 0x09 //打印机开关状态地址

#define PRINTER\_LOTNUM\_ADDRESS 0x0A //打印批号地址(需要3字节)

#define PRINTER\_SERIALNUM\_ADDRESS 0x0D //打印流水号地址(需要3字节)

#define PRINTER\_CONTRACTNUM\_ADDRESS 0x10 //打印合同号地址(需要3字节)