FC数据采集卡

API手册

# 错误码说明

|  |  |  |
| --- | --- | --- |
| **错误码定义** | **取值** | **含义** |
| HWA\_FC\_OP\_OK | 0 | 操作成功 |
| HWA\_FC\_ERR\_PARA | -1 | 传入参数错误 |
| HWA\_FC\_ERR\_INIT | -2 | 驱动初始化错误 |
| HWA\_FC\_ERR\_UNINIT | -3 | 驱动注销错误 |
| HWA\_FC\_ERR\_OPEN | -4 | 打开板卡错误 |
| HWA\_FC\_ERR\_CLOSE | -5 | 关闭板卡错误 |
| HWA\_FC\_ERR\_SEND | -6 | 发送数据错误 |
| HWA\_FC\_ERR\_RECV | -7 | 接收数据错误 |
| HWA\_FC\_ERR\_SEND\_OVERFLOW | -8 | 发送数据缓存溢出 |
| HWA\_FC\_ERR\_XML | -9 | 配置XML文件错误 |
| HWA\_FC\_ERR\_CFG | -10 | 加载板卡错误 |
| HWA\_FC\_ERR\_MSG | -11 | 接收数据消息错误 |
| HWA\_FC\_ERR\_FIFO | -12 | 操作数据存储FIFO错误 |
| HWA\_FC\_ERR\_PORT | -13 | 端口操作错误 |
| HWA\_FC\_ERR\_LOGIN | -14 | 登陆错误 |
| 说明：  上述错误码中红色字体标识的错误码为在正常模式中使用的错误码，在采集模式下不使用。针对每一种错误可以通过调用函数***HwaFcGetLastError***函数获取最后一次发生错误的描述信息。 | | |

# 常用宏定义

|  |  |  |
| --- | --- | --- |
| **错误码定义** | **取值** | **含义** |
| SERIAL\_LEN | 16 | FC卡 Serial Number 长度 |
| HWA\_FC\_FILTER\_FILED\_CNT | 6 | FC采集卡过滤条件参数类型个数 |

# 结构体说明

## 板卡信息结构体

typedef struct{

unsigned int nVendorID;

unsigned int nDeviceID;

unsigned int nBusNum;

unsigned int nSlotNum;

unsigned int nFuncNum;

unsigned int nHwVer;

char sFcSerial[SERIAL\_LEN];

}ThwaFcDevInfo;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| nVendorID | 板卡PCI-E Vendor ID |  |
| nDeviceID | 板卡PCI-E DeviceID |  |
| nBusNum | 板卡PCI-E 总线号 |  |
| nSlotNum | 板卡PCI-E 槽位号 |  |
| nFuncNum | 板卡PCI-E 功能号 |  |
| nHwVer | 板卡固件硬件版本 |  |
| sFcSerial | 板卡序列号 |  |

## 端口状态结构体

typedef struct{

unsigned int nLinkStatus;

unsigned int nFcSpeed;

unsigned int nFiberStatus;

unsigned char nRes[8];

}THwaFcPortStatus;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| nLinkStatus | 端口link状态 | Bit0：端口A状态；Bit1端口B状态 |
| nFcSpeed | 端口速度 |  |
| nFiberStatus | 端口光模块状态 | Bit0：端口A模块状态；Bit1：端口B模块状态 |
| nRes |  |  |

## 接收端口MIB统计结构体

typedef struct{

unsigned int nRXAllFrm;

unsigned int nRXAllByte0;

unsigned int nRxAllByte1;

unsigned int nRX0To127;

unsigned int nRX128To255;

unsigned int nRX256To511;

unsigned int nRX512To1023;

unsigned int nRX1024To2112;

unsigned int nRXShortErr;

unsigned int nRXLongErr;

unsigned int nRxCrcError;

unsigned int nRXByteFlux;

unsigned int nRXFrmFlux;

unsigned int nRxPortIdError;

unsigned int nRxPortEnableError;

unsigned int nRxOffsetError;

unsigned int nRxSeqIDError;

}THwaFcRXMib;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| nRXAllFrm | 端口所有接收的数据帧 |  |
| nRXAllByte0 | 端口接收字节数低32Bit |  |
| nRxAllByte1 | 端口接收字节数高32Bit |  |
| nRX0To127 | 字节数位于0至127数据帧数 |  |
| nRX128To255 | 字节数位于128至255数据帧数 |  |
| nRX256To511 | 字节数位于256至511数据帧数 |  |
| nRX512To1023 | 字节数位于512至1023数据帧数 |  |
| nRX1024To2112 | 字节数位于1024至2112数据帧数 |  |
| nRXShortErr | 长度小于最小长度的错误帧 |  |
| nRXLongErr | 长度大于最大长度的错误帧 |  |
| nRxCrcError | CRC错误数据帧数目 |  |
| nRXByteFlux | 接收字节流量 | 单位(Byte/s) |
| nRXFrmFlux | 接收帧流量 |  |
| nRxPortIdError | ID错误数据帧数目 |  |
| nRxPortEnableError | 端口未使能而丢弃的数据帧 |  |
| nRxOffsetError | Offset 错误的数据帧数目 |  |
| nRxSeqIDError | Sequence 错误数据帧数目 |  |

## 发送端口MIB统计结构体

typedef struct{

unsigned int nTXAllFrm;

unsigned int nTXAllByte0;

unsigned int nTxAllByte1;

unsigned int nTx0To127;

unsigned int nTx128To255;

unsigned int nTx256To511;

unsigned int nTx512To1023;

unsigned int nTx1024To2112;

unsigned int nTxByteFlux;

unsigned int nTxFrmFlux;

}THwaFcTxMib;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| nTXAllFrm | 总共发送的数据帧数据 |  |
| nTXAllByte0 | 发送的字节数低32bit |  |
| nTxAllByte1 | 发送的字节数高32bit |  |
| nTx0To127 | 发送长度为0至127数据帧数 |  |
| nTx128To255 | 发送长度为127至255数据帧数 |  |
| nTx256To511 | 发送长度为256至511数据帧数 |  |
| nTx512To1023 | 发送长度为512至1023数据帧数 |  |
| nTx1024To2112 | 发送长度为1024至2112数据帧数 |  |
| nTxByteFlux | 发送字节流量 | 单位：byte/s |
| nTxFrmFlux | 发送数据帧流量 |  |

## MIB统计类型

typedef enum{

eMibRx,

eMibTx,

eMibSys,

eMibNr

}THwaFcMib;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| eMibRx | 接收MIB统计信息 |  |
| eMibTx | 发送MIB统计信息 |  |
| eMibSys | 系统MIB统计信息 | 保留 |
| eMibNr |  |  |

## 端口速度定义

typedef enum{

eSpeed1G,

eSpeed2G,

eSpeedNr

}THwaFcDevSpeed;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| eSpeed1G | 工作速度1G |  |
| eSpeed2G | 工作速度2G |  |
| eSpeedNr |  |  |

## 端口选择定义

typedef enum{

ePortA = 1,

ePortB,

ePortAB,

}THwaFcPortSel;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| ePortA | 端口A |  |
| ePortB | 端口B |  |
| ePortAB | 端口A和端口B |  |

## 过滤条件定义

typedef struct{

unsigned int nSid;

unsigned int nDid;

unsigned int nType;

unsigned int nMsgId;

unsigned int nOxId;

unsigned int nSeqId;

unsigned int nPortSel;

unsigned char nFilterFiledSel[HWA\_FC\_FILTER\_FILED\_CNT];

}THwaFcMonFilter;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| nSid | 想要过滤的SID数值 |  |
| nDid | 想要过滤的DID数值 |  |
| nType | 想要过滤的type数值 |  |
| nMsgId | 想要过滤的消息ID数值 |  |
| nOxId | 想要过滤的OXID数值 | 暂保留 |
| nSeqId | 想要过滤的sequence ID数值 | 暂保留 |
| nPortSel | 想要过滤的端口 | 暂保留 |
| nFilterFiledSel | 板卡支持过滤条件的参数单独使能 | 该数据的前四个参数有效，分别代表SID、DID、TYPE、MSGID过滤使能  1 代表使能  0 代表不使能  例如nFilterFiledSel[0]=1,代表使能当前项的SID过滤条件 |

## 采集操作选择

typedef enum{

eStopRecv,

eStartRecv,

ePauseRecv,

eMonOpNr,

}THwaFcMonOperation;

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| eStopRecv | 停止采集 |  |
| eStartRecv | 开始采集 |  |
| ePauseRecv | 暂停采集 |  |
| eMonOpNr |  |  |

# 函数说明

## 驱动初始化

函数原型：int HwaFcInit(void);

函数功能：初始化FC采集卡驱动

参数说明：无

返回值：成功返回当前卡的数据，正确返回2，错误返回1

## 驱动注销

函数原型：int HwaFcDeInit(void);

函数功能：注销FC采集卡驱动

参数说明：无

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 打开卡

函数原型：int HwaFcOpenCard(int nCardNum);

函数功能：打开FC采集卡

参数说明：

nCardNum 输入参数，采集卡ID

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 关闭卡

函数原型： int HwaFcCloseCard(int nCardNum);

函数功能：关闭FC采集卡

参数说明：

nCardNum 输入参数，采集卡ID

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 设置板卡参数

函数原型：int HwaFcSetCardCfg(int nCardNum,THwaFcDevSpeed eSpeed,int nTimedOut);

函数功能：设置采集卡的工作速率

参数说明：

nCardNum:输入参数，采集卡ID

eSpeed:输入参数，采集卡工作速度

nTimedOut:输入参数，采集卡中断超时时间

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 设置采集状态

函数原型： int HwaFcSetMonOperation(int nCardNum,THwaFcMonOperation eOperation);

函数功能：设置采集卡的工作状态

参数说明：

nCardNum:输入参数，采集卡ID

eOperation:输入参数，采集的工作状态

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 接收采集数据

函数原型： int HwaFcMonRecvData(int nCardNum,char \*pBuf, int nBufSize,int nWaitTime,int \*pDataType);

函数功能：接收采集卡采集的数据

参数说明：

nCardNum: 输入参数，采集卡ID

pBuf:输出参数，放置接收的数据

nBufSize:输入参数，接收缓冲区的大小

nWaitTime:输入参数，接收等待时间

pDataType:输出参数，当前数据类型；1-满足过滤条件的数据，0-不满足过滤条件的数据

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 添加采集过滤条件

函数原型：int HwaFcAddMonFilter(int nCardNum,int nFilterCnt,THwaFcMonFilter \*pMonFilter);

函数功能：添加多条过滤条件到采集卡中

参数说明：

nCardNum:输入参数，采集卡ID

nFilterCnt:输入参数，采集卡过滤条件的数目

pMonFilter:输入参数，采集卡过滤条件首地址指针

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 删除采集过滤条件

函数原型： int HwaFcDelMonFilter(int nCardNum,int nIndex);

函数功能：删除一条过滤条件

参数说明：

nCardNum:输入参数，采集卡ID

nIndex:输入参数，过滤条件ID

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 获取板卡数目

函数原型： int HwaFcGetCardNum(void);

函数功能：获取电脑中存在的FC卡的数目

参数说明：无

返回值：成功返回板卡数目，否则返回错误码

## 获取板卡信息

函数原型： int HwaFcGetCardInfo(int nCardNum,ThwaFcDevInfo \*pDevInfo);

函数功能：获取板卡的信息

参数说明：

nCardNum:输入参数，采集卡ID

pDevInfo:输出参数，采集信息

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 获取端口状态

函数原型： int HwaFcGetPortStatus(int nCardNum,THwaFcPortStatus \* pPortStatus);

函数功能：获取采集端口状态

参数说明：

nCardNum:输入参数，采集卡ID

pPortStatus:输出参数，采集卡端口状态

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 打印板卡信息

函数原型： void HwaFcDumpCardInfo(int nCardNum,ThwaFcDevInfo nDevInfo);

函数功能：打印采集卡信息到控制台

参数说明：

nCardNum:输入参数，采集卡ID

nDevInInfo:输入参数，数据卡信息

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 获取最后一个错误描述

函数原型：const char \*HwaFcGetLastErr(void);

函数功能：获取驱动中的最后一个错误的错误描述信息

参数说明：无

返回值：返回指向错误描述信息的首地址

## 获取发送MIB信息

函数原型： int HwaFcGetTxMibs(int nCardNum,THwaFcTxMib \*pMibs,THwaFcPortSel ePort);

函数功能：获取发端口的MIB统计信息

参数说明：

nCardNum:输入参数，采集卡ID

pMibs:输出参数，输出端口的MIB信息

ePort:输入参数，选择想要获取MIB信息的端口

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 获取接收MIB信息

函数原型： int HwaFcGetRxMibs(int nCardNum,THwaFcRXMib \*pMibs,THwaFcPortSel ePort);

函数功能：获取发端口的MIB统计信息

参数说明：

nCardNum:输入参数，采集卡ID

pMibs:输出参数，输入端口的MIB信息

ePort:输入参数，选择想要获取MIB信息的端口

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

## 清除MIB统计信息

函数原型： int HwaFcClearMibs(int nCardNum);

函数功能：清除板卡的MIB统计信息

参数说明：

nCardNum:输入参数，采集卡ID

返回值：成功返回HWA\_FC\_OP\_OK，否则返回错误码

# 5 接收数据块格式说明

通过驱动接收采集数据函数得到的pBuf指针地址，是一个连续的数据包块。该数据块内包含了连续的各种数据包，每个数据包前均有一个自定义的数据包头。该自定义数据包头的格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 顺序 | 位宽 | 名称 | 描述 |
| 1 | 1 | 保留 | 保留 |
| 2 | 2 | 接收通道标志 | 01：A网  10：B网 |
| 3 | 1 | 实时数据标志 | 1：实时过滤条件数据  0：非过滤条件数据 |
| 4 | 1 | 保留 | 保留 |
| 5 | 3 | 数据包错误码标记 | 000：正确数据包  001：CRC错误  010：帧过小错误  100：帧过大错误  011：非法字符错误  101：缺少eof错误  110：在非活动状态下接收数据包的错误 |
| 6 | 2 | 保留 | 保留 |
| 7 | 9 | Port ID | 当为实时过滤数据包时有效，标示满足过滤条件的索引值 |
| 8 | 1 | 保留 | 保留 |
| 9 | 12 | 数据包长度 | 当前数据帧的整体包长字节数 |
| 10 | 96 | 保留 | 保留 |
| 11 | 18 | 保留 | 保留 |
| 12 | 14 | 数据帧接收时间定义2 | Bit13-bit5 ：天  Bit4 - bit0：小时 |
| 13 | 32 | 数据帧接收时间定义1 | Bit31-bit26：分钟  Bit25-bit20：秒钟  Bit19-bit10：毫秒  Bit9 - bit0 ：微妙 |
| 14 | 64 | 保留 | 保留整个自定义包头共占32字节。 |
| 15 | 可变 | FC数据帧 | FC数据帧的内容，注意为8字节对齐，当数据包字节数不是8的整数倍时，则每一个数据包尾部会补0. |
| 。。。 | 。。。 | 。。。 | 。。。（下一个数据包的自定义头） |
| 。。。 | 。。。 | 。。。 | 。。。 |