API 使用说明

南京艾科朗克（AceleCom）信息科技有限公司自主研发行情加速系统 Xele-MD 提供微秒级最优最快多档行情。本说明书适用于 Xele-MD 网络版和本地版。

AcceleCom

南京艾科朗克信息科技有限公司

XELE-MD

API 使用说明

声明

商标声明

AcceleCom和其他艾科朗克商标均为南京艾科朗克信息科技有限公司（以下简称艾科朗克）的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受艾科朗克商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，艾科朗克对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

南京艾科朗克信息科技有限公司

地址：南京市秦淮区永丰大道 8 号白下高新园区 3 号楼 B 栋 101　　邮编：210014

电话：025-88017287

传真：025-85766287

邮件：support@accelecom.com

网址：www.accelecom.com

版本记录

日期	版本号	API 版本号	备注
22/01/2015	Lichun 1.0	(飞马行情 API)	文档初始化完成
02/02/2015	Lichun 1.1	(飞马行情 API)	更新了环境变量配置信息
03/02/2015	Lichun 1.1.1	(飞马行情 API)	更新了环境变量配置信息 typo
29/03/2015	Lichun 2.0	(飞马行情 API)	合并网络版和本地版 API 说明书
13/04/2015	Lichun 2.1	r315(Accelecom 自有 API)	推出 Accelecom API，不再支持飞马行情 API。
20/04/2015	Lichun 2.2	r342(Accelecom 自有 API)	Bug 修复
27/04/2015	Lichun 2.2	r364(Or above)	API 接口精简，优化
30/04/2015	Lichun 2.2	r389	多线程策略调用 api，增加接口
01/06/2015	Lichun 2.2	r582	Bug 修复
02/09/2015	Lichun 2.1	本地 r839，网络 r800	本地版 API 说明更新
03/09/2015	Lichun 2.1	本地 r839, 网络 r800	网络版 API 说明更新
28/10/2015	Lichun 2.1	网络 r867	网络版 API 添加上期所行情支持
18/11/2015	Lichun 2.1	网络版 r972	一档行情更名为一档高频行情，增加一档低频行情， 网络版暂不支持无限深度行情
23/11/2015	Jingzhe 1.0	本地版 r979	一档行情更名为一档高频行情，增加一档低频行情，本地版支持无限深度行情
23/11/2015	Jingzhe 1.0	本地版 r1012	修改一档低频行情不正确接收的 bug，增加行情快照
12/07/2016		上期网络版 r1248	静态库增加-fPIC 编译选项，供外部编译为动态库使用
22/03/2017		大商网络版 r2245	新增大商网络版一档行情 API，暂不支持多档行情及本地版。

目录

1 前言

该文档是行情加速系统 Xele-MD 的投资者使用手册，它提供 API 说明，及使用说明。本说明书适用于 Xele-MD 本地版和网络版。

2 Xele-MD 简介

Xele-MD 是业内领先的基于 FPGA、亚微秒级行情优化通报系统，独创完整流程的 FPGA 数据传输系统，拥有纳秒级响应速度，提供最快速准确的资讯通道；是为证券、期货高端人士及机构、基金类专业投资机构及产业巨头量身打造的高性能交易系统的重要一环。

	网络版	混合模式	本地版
速度	快	更快	最快
客户数	多用户	多用户	独占式

Xele-MD 分为网络版、本地版。可分别单独使用，也可串接使用。

各自特点对比详情请见 http://www.accelecom.com/product/xele-md

Xele-MD 主要具有以下技术特点：

1. 汇集所有行情，配置简单，对交换机影响小；
2. 两种部署方式，灵活方便，既可以用户独享，也可以通过万兆光口批量转发；
3. 采用硬件行情加速卡，集成 TCP 解压引擎，无需内核处理；
4. 为低延迟定制的 DMA 通道，无需内核干预，数据搬运速度达到极致；
5. 硬件协助处理行情信息，进一步提高行情使用速度；
6. 简单易用的 API 和用例，可直接上手使用。

3 Xele-MD 网络版使用说明

3.1 API

3.1.1API 介绍

用户**不能再使用**飞马行情 API 接入。行情 API 请参考如下两个文档：

1. API 发布包中的有关 user_demo 的 README 说明。
2. “Xele-MD_API_Guide”网络版 API 详细使用手册(以手册为准)

注意：本地版 Xele-MD 需要使用不同的艾科朗克 API，它支持 PCIe DMA 高速行情接入。

Xele-MD 网络版仅支持如下必须的接口：

3.1.2CXeleMdSpi 接口

1. OnFrontUserLoginSuccess
2. OnFrontDisconnected
3. ~~OnRtnDepthMarketData~~

3.1.3CXeleMdApi 接口

1. CreateMdApi
2. Release
3. LoginInit
4. GetVersion
5. GetHandle
6. RecvMarketDataTick

3.2 API 接口变化

3.2.1r800 开始支持中金所多播行情

由于中金所多播行情接口变化较大，API 也随之进行了较大变动，不再采用之前的 index 索引形式获取行情，采用轮询的方法获取最新行情。

3.2.2r867 开始增加支持上期所行情

RecvMarketDataTick 参数增加了 handle，脱离 API 类成为全局函数，并添加了 RecvShfeMarketDataTick 用于接收上期所行情。

3.2.3r2229 开始增加支持大商所行情

新增全局函数 RecvDceMarketDataTick 用于接收大商所行情。

3.3 CXeleMdSpi 接口说明

3.3.1OnFrontUserLoginSuccess

原型：void　OnFrontUserLoginSuccess()

参数：无

返回值：无

从上一版本的 OnFrontConnected 变化而来，当 API 成功连接并成功登陆时回调 SPI 此方法进行通知。

3.3.2OnFrontDisconnected

原型:voidOnFrontDisconnected(int*nReason)*

参数： API 通知的错误代码

返回值：无

与上一版本基本相同，当 API 成功连接时意外情况断开时的回调方法，*nReason* 为内部错误码。

3.4 CXeleMdApi 接口说明

3.4.1CreateMdApi

原型： CXeleMdApi *CreateMdApi(CXeleMdSpi* *spi)*

参数： MdSpi 的子类实体指针

返回值：NULL（表示创建失败）或者一个合法的 api 对象指针

与上一版本除函数名称外，参数直接使用了 spi 的指针，相当于整合了老版本 RegisterSpi 接口。

3.4.2GetVersion

原型： const char *GetVersion()

参数：无

返回值：该版本 API 库的信息

3.4.3LoginInit

原型： int LoginInit(constchar **frontAddress*,

const char *multicastAddress,

const char *nic

CXeleFtdcReqUserLoginField **pReqUserLogin)*

参数： frontAddress：前置机地址,格式为 tcp://<ip>:<port>

multicastAddress:多播地址,格式为 udp://<ip>:<port>

nic:侦听多播的网卡接口名，如”eth0”,可填写空字符串（非 NULL）自动获取

pReqUserLogin：登陆域信息结构体

返回值：执行结果代码

该方法是上个版本 Init 的改良，整合了 RegisterFront,不再需要 ReqUserLogin 方法，也就是 Init 和 userlogin 一起完成，同时返回执行结果的枚举值。

3.4.4Release

原型:void Release()

参数：无

返回值：无

自动登出，并销毁 API 对象资源。

3.4.5GetHandle

返回值：API 创建成功后，获取行情接收的句柄，该句柄被用作接收函数的参数。

3.5 RecvMarketDataTick 中金所行情收取（全局函数）

原型：bool RecvMarketDataTick(int handle, MarketDataTick* mdtick)

参数：handle 句柄，mdtick 指针

返回：是否成功接收 tick 的 bool 值

收取中金所行情的接口。用户提供 MarketDataTick 的变量指针，利用 while 循环根据返回值判断该内存是否写入了最新的合法 tick 数据，实际的数据域为 mdtick->data。可以参考“userdemo.cpp”示例代码。

3.6 RecvShfeMarketDataTick 上期所行情收取(全局函数)

原型：bool RecvShfeMarketDataTick(int handle, CXeleShfeMarketDataUnion* tick)

参数：handle 句柄，联合体指针 tick

返回：是否成功接收 tick 的 bool 值

收取上期所行情的接口。用户提供 CXeleShfeMarketDataUnion 类型结构的变量的指针，利用 while 循环根据返回值判断内存是否写入了最新的 tick 数据。有以下两种数据类型：

1. tick->md_type[0] == ‘M’,代表一档高频行情，提取数据的方法为 tick->type_high 变量。
2. tick->md_type[0] == ‘S’,代表一档低频行情，提取数据的方法为 tick->type_low 变量。
3. tick->md_type[0] == ‘Q’,代表无限深度行情，提取数据的方法为 tick->type_depth 变量。
4. 网络版暂不支持无限深度行情

对应的变量类型即为相应行情的数据类型。可以参考“userdemo_shfe.cpp”示例代码。

3.7 RecvDceMarketDataTick 大商所行情收取(全局函数)

原型：bool RecvDceMarketDataTick(int handle, CXeleDceMarketDataUnion* tick)

参数：handle 句柄，联合体指针 tick

返回：是否成功接收 tick 的 bool 值

收取上期所行情的接口。用户提供 CXeleDceMarketDataUnion 类型结构的变量的指针，利用 while 循环根据返回值判断内存是否写入了最新的 tick 数据。有以下两种数据类型：

1. tick->md_type[0] == ‘H’,代表一档高频行情，提取数据的方法为 tick->type_high 变量。

2. tick->md_type[0] == ‘L’,代表一档低频行情，提取数据的方法为 tick->type_low 变量。
3. 目前只支持一档行情。

对应的变量类型即为相应行情的数据类型。可以参考“userdemo_dce.cpp”示例代码。

3.8 行情 API 使用方法

在创建 API 并登录成功后，调用 GetHandle 接口获取行情收取句柄，然后根据不同的交易所行情调用相应的行情收取接口

中金所行情 API 数据类型	
名称	说明
MarketDataTick	行情获取的原始结构
CXeleMdFtdcDepthMarketDataField	深度行情数据域

上期所行情 API 数据类型	
名称	说明
CXeleShfeMarketDataUnion	行情获取的原始结构
CXeleShfeHighLevelOneMarketData	一档高频行情数据域
CXeleShfeLowLevelOneMarketData	一档低频行情数据域
CXeleShfeDepthMarketData	无限深度行情数据域

大商所行情 API 数据类型	
名称	说明
CXeleDceMarketDataUnion	行情获取的原始结构
CXeleDceHighMarketData	一档高频行情数据域
CXeleDceLowMarketData	一档低频行情数据域

3.9 LoginInit 返回值说明

3.9.1XELEAPI_SUCCESS

成功登陆。

3.9.2XELEAPI_TCP_CLOSED

Tcp 处于关闭状态。

3.9.3XELEAPI_SOCKET_ERROR

建立 socket 失败。

3.9.4XELEAPI_USER_LOGIN_RESPONSE_TIMEOUT

成功建立连接并发出 login 请求，但是应答超时。

3.9.5XELEAPI_BAD_USERID_OR_PASSWORD
错误的用户名或者密码。

3.9.6XELEAPI_BAD_FRONTADDRESS
错误的前置机地址格式。

3.10 登录信息
登录 Xele-MD 的用户信息与投资者现有的飞马行情登录信息**不一致**。请联系期货公司运维人员获取。user_demo 配置文件为其所在目录的 refer.ini,配置如下：

```
################################################################################

##用户名（由期货公司提供）
USERID=accelecom
##密码（由期货公司提供）
PASSWD=654321
##订阅行情前置的地址和端口号（由期货公司提供）
FRONTADDRESS=tcp://10.128.123.74:32002
##Xele-MD多播行情地址
MCASTADDRESS=tcp://233.0.0.1:28929
##Xele-MD多播侦听网卡接口名
NIC=eth0
################################################################################
```

3.11 运行

投资者登录 Xele-MD 后，Xele-MD 会主动向您推送高速行情。如果运行 demo 中的例子，则需要在目录下运行：

注意：如果提示找不到库:
error while loading shared libraries:
libXeleFemaMduserAPI64.so: cannot open shared object file: No such file or directory
则可以添加 LD_LIBRARY_PATH 环境变量到~/.bashrc 最后面，如:
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
注意最后的“.”，表示当前目录。
保存后，执行:
source ~/.bashrc

```
./user_demo
```

4 Xele-MD 中金所本地版使用说明

为了追求极致，我们特别提供了 Xele-MD 本地版。相比 Xele-MD 网络版，它更加极速。

4.1 API 接口变化

精简 API 接口，减少非必要的冗余操作；去除回调接口，消除因回调产生的 CPU 100%占用率的副作用；去除多策略时可能因为一个策略慢而阻塞整个行情的接收的问题；可随时获取一整日的行情，直到 Xele-MD 或者机器重启；进一步提升行情的接收速度。

4.1.1删除 AcXeleMDSpi 接口

不再支持 spi 回调，新增读取当前行情池的个数方法，通过行情首地址计算出最新行情的位置。

4.1.2AcXeleMD 接口精简

AcXeleMD 接口减少至 4 个，其中必要接口只有 3 个，更加易用、灵活。

4.1.3行情数据结构修改

不再使用飞马数据结构，新的结构中存在部分的 reserved 字段，使用时请注意剔除。

4.2 API

4.2.1API 介绍

注意：本 API 仅适用于 Xele-MD 本地版。

Xele-MD API 包含以下个文件：

文件名	版本	文件描述
ac_xele_md.h	V2.0	行情接口头文件，原 ac_xele_md_api.h
~~ac_xele_md_spi.h~~	~~V1.0~~	~~该文件已删除~~
ac_xele_md_struct.h	V2.0	行情数据结构体头文件
ac_xele_md_type.h	V2.0	行情数据类型头文件
libac_xele_md.a	V2.0	静态链接库
libac_xele_md.so	V2.0	动态链接库

Xele-MD API 支持多个不同用户进程、程序同时调用来处理不同的用户策略，不同策略间无影响。同时根据需要，可通过 API 获取一整天的行情（若 Xele-MD 重启，则数据会丢失）。

4.2.2运行模式

Xele-MD API 线程安全，可支持同一个应用程序多个线程同时接收行情，亦支持多个不同的应用程序同时接收行情。API 的行情获取接口需要用户根据实际情况主动调用。各进程或线程间独立调用 Xele-MD API，无交叉影响。

4.2.3AcXeleMD 接口

AcXeleMD 接口实现 Xele-MD API 的初始化、版本信息以及行情信息的接收。

4.2.3.1AcXeleMDInit

初始化接口

函数原型：boolAcXeleMDInit();

参数：无

返回值：true 成功，false 失败

4.2.3.2AcXeleMDCount

获取当前行情数量，当返回数值变化是，表示有新的行情到来

函数原型：intAcXeleMDCount ();

参数：无

返回值：行情数据个数

4.2.3.3AcXeleMDData

获取行情数据首地址 market_data，根据 AcXeleMDCount 得到的数据个数 N 来计算最新的行情数据指针 market_data + N。

函数原型：

```
const DepthMarketDataField *AcXeleMDData();
```

参数：无

返回值：const DepthMarketDataField *market_data 行情数据首地址

深度行情信息结构：

```
///深度行情
structDepthMarketDataField
{
	///合约代码
	InstrumentIDTypeInstrumentID;
	///最后修改时间
	TimeTypeUpdateTime;
	///最后修改毫秒
	MillisecTypeUpdateMillisec;
	///预留
	Reserved4Type	__reserved4_1;
	///今开盘
	PriceTypeOpenPrice;
	///最高价
	PriceTypeHighestPrice;
	///最低价
	PriceTypeLowestPrice;
	///今收盘
	PriceTypeClosePrice;
	///涨停板价
	PriceTypeUpperLimitPrice;
	///跌停板价
	PriceTypeLowerLimitPrice;
	///今结算
	PriceTypeSettlementPrice;
	///今虚实度
	RatioTypeCurrDelta;
	///预留
	Reserved4Type	__reserved4_2;
	///最新价
	PriceTypeLastPrice;
	///数量
	VolumeType	Volume;
	///成交金额
	MoneyType	Turnover;
	///持仓量
```

```
    LargeVolumeTypeOpenInterest;
    ///预留
    Reserved4Type     __reserved4_3;
    ///申买价一
    PriceType         BidPrice1;
    ///申买量一
    VolumeType        BidVolume1;
    ///申卖价一
    PriceType         AskPrice1;
    ///申卖量一
    VolumeType        AskVolume1;
    ///预留
    Reserved4Type     __reserved4_4;
    ///申买价二
    PriceType         BidPrice2;
    ///申买量二
    VolumeType        BidVolume2;
    ///申买价三
    PriceType         BidPrice3;
    ///申买量三
    VolumeType        BidVolume3;
    ///预留
    Reserved4Type     __reserved4_5;
    ///申卖价二
    PriceType         AskPrice2;
    ///申卖量二
    VolumeType        AskVolume2;
    ///申卖价三
    PriceType         AskPrice3;
    ///申卖量三
    VolumeType        AskVolume3;
    ///预留
    Reserved4Type     __reserved4_6;
    ///申买价四
    PriceType         BidPrice4;
    ///申买量四
    VolumeType        BidVolume4;
    ///申买价五
    PriceType         BidPrice5;
    ///申买量五
    VolumeType        BidVolume5;
    ///预留
    Reserved4Type     __reserved4_7;
    ///申卖价四
    PriceType         AskPrice4;
    ///申卖量四
    VolumeType        AskVolume4;
    ///申卖价五
    PriceType         AskPrice5;
    ///申卖量五
    VolumeType        AskVolume5;
} __attribute__((packed));
```

4.2.3.4AcXeleMDVersion

获取当前 API 版本号。

函数原型：const char *AcXeleMDVersion();

参数：无

返回值：当前 API 版本号

4.3 登录信息

Xele-MD 本地版登录信息通过《Xele-MD 安装运维手册》中 8.1《ftd_md 配置文件》所述步骤设置。并且不再需要 mduserdemo\target\lnx64\mduserdemo.ini。

4.4 运行

如果运行 demo 中的例子，则./SReference，Xele-MD 会主动向您推送高速行情。

4.5 备注

如果希望尽早处理接收到的行情，请高频率的调用 AcXeleMDCount 函数来判断是否已有新的行情到达。

5 Xele-MD 上期所本地版本使用说明

5.1 API

5.1.1API 介绍

上期 Xele-MD API 包含以下个文件：

文件名	版本	文件描述
ac_xele_md.h	V1.0	行情接口头文件，原 ac_xele_md_api.h
ac_xele_md_struct.h	V1.0	行情数据结构体头文件
ac_xele_md_type.h	V1.0	行情数据类型头文件
libac_xele_md.a	V1.0	静态链接库
libac_xele_md.so	V1.0	动态链接库

上期 Xele-MD API 支持多个不同用户进程、程序同时调用来处理不同的用户策略，不同策略间无影响。同时根据需要，可通过 API 获取一整天的行情（若 Xele-MD 重启，则数据会丢失）。

5.1.2运行模式

Xele-MD API 线程安全，可支持同一个应用程序多个线程同时接收行情，亦支持多个不同的应用程序同时接收行情。API 的行情获取接口需要用户根据实际情况主动调用。各进程或线程间独立调用 Xele-MD API，无交叉影响。

5.1.3AcXeleMD 接口

AcXeleMD 接口实现 Xele-MD API 的初始化、版本信息以及行情信息的接收。

5.1.3.1AcXeleMDInit

初始化接口

函数原型：boolAcXeleMDInit();

参数：无

返回值：true 成功，false 失败

5.1.3.2AcXeleMDCount

获取当前行情数量，当返回数值变化是，表示有新的行情到来。

函数原型：intAcXeleMDCount ();

参数：无

返回值：行情数据个数

5.1.3.3AcXeleMDData

获取行情数据首地址 market_data，根据 AcXeleMDCount 得到的数据个数 N 来计算最新的行情数据指针 market_data + N。

函数原型：

constMarketData *AcXeleMDData();

参数：无

返回值：constMarketData *market_data 行情数据首地址

行情信息结构：

```
struct MarketData
{
	//行情类型
	//一档高频 “MD” 一档低频 “SM” 无限深度 “QM” 快照 “K” 快照结束 “E”
	MdType Md;
	//行情数据域
	char dataItem[86];
};
```

行情数据域根据行情类型分为三种数据结构：

5.1.3.3.1一档高频行情

```
struct LevelOneMarketDataField
{
//预留字段
    Reserved4Type    __reserved_1;
```

```
//合约代码
InstrumentIDType InstrumentID;
//最后修改时间
TimeType         UpdateTime;
//最后修改毫秒
MillisecType     UpdateMillisec;
//数量
VolumeType       Volume;
//最新价
PriceType        LastPrice;
//成交金额
MoneyType        Turnover;
//持仓量
LargeVolumeType  OpenInterest;
//申买价
PriceType        BidPrice;
//申卖价
PriceType        AskPrice;
//申买量
VolumeType       BidVolume;
//申卖价
VolumeType       AskVolume;
}__attribute__((packed));
```

5.1.3.3.2一档低频行情

```
struct   LowLevelOneMarketDataField
{
///预留字段
    Reserved4Type    __reserved_1;
///合约代码
InstrumentIDType1    InstrumentID;
///最后修改时间
TimeType             UpdateTime;
//今开盘
PriceType            OpenPrice;
//最高价
PriceType            HighestPrice;
//最低价
PriceType            LowestPrice;
//今收盘
PriceType            ClosePrice;
//涨停板价
PriceType            UpperLimitPrice;
//跌停板价
PriceType            LowerLimitPrice;
//今结
```

```
PriceType            SettlementPrice;
//今需实度
RatioType            CurrDelta;
}__attribute__((packed));
```

5.1.3.3.3快照行情

```
struct   QuickStartMarketDataField
{
///预留字段
    Reserved4Type    __reserved_1;
///合约代码
InstrumentIDType1    InstrumentID;
///最后修改时间
TimeType             UpdateTime;
//今开盘
PriceType            OpenPrice;
//最高价
PriceType            HighestPrice;
//最低价
PriceType            LowestPrice;
//今收盘
PriceType            ClosePrice;
//涨停板价
PriceType            UpperLimitPrice;
//跌停板价
PriceType            LowerLimitPrice;
//今结
PriceType            SettlementPrice;
//今需实度
RatioType            CurrDelta;
}__attribute__((packed));
```

5.1.3.3.4无限深度行情

```
struct        QueryMarketDataField
{
//预留字段
    Reserved4Type       __reserved_1;
//合约代码
ExInstrumentIDType  InstrumentID;
//买卖方向
DirectionType       Direction;
//最后修改时间
TimeType            UpdateTime;
//最后修改毫秒
MillisecType        UpdateMillisec;
//数量 1
```

```
VolumeType          Volume1;
//价格 1
PriceType           Price1;
//数量 2
VolumeType          Volume2;
//价格 2
PriceType           Price2;
//数量 3
VolumeType          Volume3;
//价格 3
PriceType           Price3;
//数量 4
VolumeType          Volume4;
//价格 4
PriceType           Price4;
//数量 5
VolumeType          Volume5;
//价格 5
PriceType           Price5;
}__attribute__((packed));
```

5.1.3.4AcXeleMDVersion

获取当前 API 版本号。

函数原型：const char *AcXeleMDVersion();

参数：无

返回值：当前 API 版本号

5.2 登录信息

Xele-MD 本地版登录信息通过《Xele-MD 安装运维手册》中 8.1《ftd_md 配置文件》所述步骤设置。并且不再需要 mduserdemo\target\lnx64\mduserdemo.ini。

5.3 运行

如果运行 demo 中的例子，则./SReference，Xele-MD 会主动向您推送高速行情。

5.4 备注

如果希望尽早处理接收到的行情，请高频率的调用 AcXeleMDCount 函数来判断是否已有新的行情到达。