



IEC103

服务器端开发包接口 API 使用说明书

版本：V2.0

深圳天勺电力软件有限公司

深圳 中国

1. 开发包简介

该开发包根据已经正式发行的 IEC 60870-5-103 最新版本开发而成，采用 C 语言编写，可跨平台使用。

为了最大限度地简化使用，完全封装了 IEC 60870-5-103 中的诸多底层细节和各种复杂模型的实现逻辑。用户利用它作开发的时候无需了解底层细节，只需要调用功能接口函数完成相应的功能研发，具有方便、快捷的优点。

本接口说明文档为接口全集，客户可以根据具体业务选择相应的接口进行调用。本开发包既可以用在开发基于 IEC 60870-5-103 标准的所有设备中（包括各个应用领域），也可以集成在网关机和通信管理机中。现场运行稳定可靠，调用简单高效。

2. 接口描述

2.1 服务应用函数

函数声明	int IEC103Int_InitAllCfg (void);
功能	103 服务配置读取和全局资源初始化接口，在 103 服务前调用
参数	无
返回值	0 - 成功; -1 - 失败
示例	IEC103Int_InitAllCfg ();

函数声明	int Server_IEC103_MemFree(void);
功能	103 资源区释放，在程序退出时调用
参数	无
返回值	0 - 成功; -1 - 失败
示例	Server_IEC103_MemFree();

函数声明	void Init_PortCtrl(BYTE byPortNo, BOOLEAN bFirst)
功能	103 端口初始化函数，在 IEC103Int_InitAllCfg 后，103 主服务前调用
参数	byPortNo:是指 Port.cfg 配置文件中的端口编号 PortNo。 bFirst: bFirst=TRUE:第一次初始化
返回值	无
示例	Init_PortCtrl(0 , TRUE);

函数声明	void Tcp_Initial()
功能	103 服务 TCP 连接初始化，在 IEC103Int_InitAllCfg 后，103 主服务前调用
参数	无
返回值	无

示例	Tcp_Initial();
----	----------------

函数声明	void Udp_Initial ()
功能	103 服务 UDP 连接初始化, 在 IEC103Int_InitAllCfg 后, 103 主服务前调用
参数	无
返回值	无
示例	Udp_Initial();

函数声明	void Server_IEC103_TCP_TaskStart(BYTE* arg)
功能	103 服务基于 TCP 协议传输主服务线程包裹函数
参数	BYTE* arg: 携带 portNo 值的 BYTE 类型的指针
返回值	无
示例	<pre> BYTE byPortNo = 1 ; //端口编号 pthread_t threadServer; if(pthread_create(&threadServer,NULL,(void*)Server_IEC103_TCP_TaskStart, &byPortNo)) { perror("pthread_create failed"); return -1; // 创建线程失败 } </pre>

函数声明	void Server_IEC103_TCP_Main(BYTE byPortNo)
功能	网口(TCP)IEC103 规约处理主函数
参数	byPortNo: 端口序号。注意: 这个函数如果要单独调用的话 byPortNo 需从 0 开始计数, 举个例子: Port.cfg 中 PortNo 如果填写的是 1 , 在这里传参就应该是 0 (手动减一)
返回值	无
示例	Server_IEC103_TCP_Main(0);

函数声明	void Server_IEC103_UDP_TaskStart(BYTE* arg)
功能	103 服务基于 UDP 协议传输主服务线程包裹函数
参数	BYTE* arg: 携带 portNo 值的 BYTE 类型的指针
返回值	无
示例	<pre> BYTE byPortNo = 1 ; //端口编号 pthread_t threadServer; if(pthread_create(&threadServer,NULL,(void*)Server_IEC103_UDP_TaskStart, &byPortNo)) { perror("pthread_create failed"); return -1; // 创建线程失败 } </pre>

函数声明	void Server_IEC103_UDP_Main(BYTE byPortNo)
------	---

功能	网口(UDP)IEC103 规约处理主函数
参数	byPortNo: 端口序号。注意: 这个函数如果要单独调用的话 byPortNo 需从 0 开始计数, 举个例子: Port.cfg 中 PortNo 如果填写的是 1, 在这里传参就应该是 0 (手动减一)
返回值	无
示例	Server_IEC103_UDP_Main(0);

函数声明	void Server_IEC103_Init(BYTE byPortNo)
功能	103 服务初始化
参数	byPortNo : 是指 Port.cfg 配置文件中的端口编号 PortNo
返回值	无
示例	Server_IEC103_Init(0);

函数声明	void Server_IEC103_Proc(BYTE byPortNo)
功能	103 数据报文处理函数, 用以处理接收的相关命令报文
参数	byPortNo : 是指 Port.cfg 配置文件中的端口编号 PortNo
返回值	无
示例	Server_IEC103_Proc(0);

函数声明	void Server_IEC103_Send(BYTE byPortNo)
功能	103 数据报文发送函数, 用以发送相关命令报文
参数	byPortNo : 是指 Port.cfg 配置文件中的端口编号 PortNo
返回值	无
示例	Server_IEC103_Send(0);

函数声明	BOOLEAN Server_IEC103_UdpMsgRecv(BYTE byPortNo)
功能	基于 UDP 协议的 Net103 网络数据接收方法
参数	byPortNo : 是指 Port.cfg 配置文件中的端口编号 PortNo
返回值	1 - 成功; 0 - 失败
示例	Server_IEC103_UdpMsgRecv(0)

函数声明	BOOLEAN Time_GetSystemTimeBCD(tagPTimeBCD ptTimeBcd)
功能	获取当前系统时间
参数	tagPTimeBCD ptTimeBcd: 需要赋值的时间参数指针, 具体定义可以查看 TimeDef.h 文件
返回值	1 - 成功; 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd);

函数声明	BOOLEAN Event_GenerateRecordCOS(BYTE typeYX , BYTE byStatusYX , WORD wFun , WORD wInf)
------	---

功能	生成一条 COS 遥信变位记录
参数	BYTE typeYX: 遥信类型 0: 单点 1: 双点 BYTE bySatusYX:遥信变位值 WORD wFun: 对应遥信点 func 号 WORD wInf: 对应的遥信点 inf 号
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	Event_GenerateRecordCOS(0 , 0 , 1 , 1);

函数声明	BOOLEAN Event_GenerateRecordSOE(BYTE typeYX , BYTE byStatusYX , WORD wFun , WORD wInf , tagPTimeBCD ptTimeAct)
功能	生成一条 SOE 遥信变位记录
参数	BYTE typeYX: 遥信类型 0: 单点 1: 双点 BYTE bySatusYX:遥信变位值 WORD wFun: 对应遥信点 func 号 WORD wInf: 对应的遥信点 inf 号 tagPTimeBCD ptTimeAct : 对应变位时间点
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd); Event_GenerateRecordSOE(1 , 3 , 1 , 2 , &timeBcd);

函数声明	BOOLEAN Event_GenerateRecordCHK(WORD wState , WORD wSysFun , WORD wInf , tagPTimeBCD ptTimeBcdChk)
功能	生成一条自检信息
参数	BYTE wState:自检信息状态值 WORD wFun: 对应点 func 号 WORD wInf: 对应的点 inf 号 tagPTimeBCD ptTimeBcdChk: 对应发生时间点
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd); Event_GenerateRecordCHK(0 , 1 , 3 , &timeBcd);

函数声明	BOOLEAN Event_GenerateRecordRUN(WORD wState , WORD wSysFun , WORD wInf , tagPTimeBCD ptTimeBcdRUN)
功能	生成一条运行信息
参数	BYTE wState: 运行信息状态值 WORD wFun: 对应点 func 号 WORD wInf: 对应的点 inf 号 tagPTimeBCD ptTimeBcdRUN: 对应发生时间点
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd); Event_GenerateRecordRUN(0 , 1 , 4 , &timeBcd);

函数声明	BOOLEAN Event_GenerateRecordALM(WORD wState , WORD wSysFunc , WORD wInf , WORD wFaultNo , tagPTimeBCD ptTimeBcdAlm)
功能	生成一条告警信息
参数	BYTE wState: 告警信息状态值 WORD wFun: 对应点 func 号 WORD wInf: 对应的点 inf 号 WORD wFaultNo : 系统故障序号 tagPTimeBCD ptTimeBcdAlm: 对应发生时间点
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd); Event_GenerateRecordALM(0, 1, 5, 1, &timeBcd);

函数声明	BOOLEAN Event_GenerateRecordAct(WORD wState , WORD wSysFunc , WORD wInf , WORD wFaultNo , DWORD dwTripTime , tagPTimeBCD ptTimeBcdAct)
功能	产生一条动作记录
参数	BYTE wState: 动作记录状态值 WORD wFun: 对应点 func 号 WORD wInf: 对应的点 inf 号 WORD wFaultNo : 系统故障序号 DWORD dwTripTime: 相对时间用毫秒表示 tagPTimeBCD ptTimeBcdAct : BCD 时间戳
返回值	TRUE 1- 成功; FALSE 0 - 失败
示例	tagTimeBCD timeBcd; Time_GetSystemTimeBCD(&timeBcd); Event_GenerateRecordAct(stature , 1 , 6 , 2 , 20 , &timeBcd);

2.2 装置接口回调函数

函数原型定义	typedef BYTE (*getBHStateByFun_Inf)(DWORD , DWORD);
函数全局变量	getBHStateByFun_Inf getBHStateByFun_Inf_fun;
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的保护遥信的状态值
注册回调方法	int regist_GetBhValueByFun_Inf(getBHStateByFun_Inf f)
原型参数值	DWORD fun , DWORD inf: Map*.cfg 配置文件中的保护遥信的 (fun , inf)
原型返回值	BYTE: 返回的保护遥信状态值(注意返回的状态值应是 0x01、0x11、0x00、

	0x10)
示例	BYTE demoBH(DWORD func , DWORD inf) {...}; regist_GetBhValueByFun_Inf(demoBH);

函数原型定义	typedef BYTE (*getCHKSatByFun_Inf)(DWORD , DWORD);
函数全局变量	getCHKSatByFun_Inf getCHKStateByFun_Inf_fun;
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的自检信息的状态值
注册回调方法	int regist_GetChkValueByFun_Inf(getCHKSatByFun_Inf f)
原型参数值	DWORD fun , DWORD inf: Map*.cfg 配置文件中的自检信息的 (fun , inf)
原型返回值	BYTE: 返回的自检信息状态值(注意返回的状态值应是 0x01、0x11、0x00、0x10)
示例	BYTE demoCHK(DWORD func , DWORD inf){...}; regist_GetChkValueByFun_Inf(demoCHK);

函数原型定义	typedef BYTE (*getAlmStateByFun_Inf)(DWORD , DWORD);
函数全局变量	getAlmStateByFun_Inf getAlmStateByFun_Inf_fun;
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的告警遥信的状态值
注册回调方法	int regist_GetAlmValueByFun_Inf(getAlmStateByFun_Inf f)
原型参数值	DWORD fun , DWORD inf: Map*.cfg 配置文件中的告警遥信的 (fun , inf)
原型返回值	BYTE: 返回的告警遥信状态值(注意返回的状态值应是 0x01、0x11、0x00、0x10)
示例	BYTE demoALM(DWORD func , DWORD inf){...}; regist_GetAlmValueByFun_Inf(demoALM);

函数原型定义	typedef BYTE (*getInputStateByFun_Inf)(DWORD , DWORD);
函数全局变量	getInputStateByFun_Inf getInputStateByFun_Inf_fun;
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的保护开入遥信的状态值
注册回调方法	int regist_GetInputValueByFun_Inf(getInputStateByFun_Inf f)
原型参数值	DWORD fun , DWORD inf: Map*.cfg 配置文件中的保护开入遥信的 (fun , inf)
原型返回值	BYTE: 返回的保护开入遥信状态值(注意返回的状态值应是 0x01、0x11、0x00、0x10)
示例	BYTE demoINPUT(DWORD func , DWORD inf){...};

	<code>regist_GetInputValueByFun_Inf(demoINPUT);</code>
函数原型定义	<code>typedef BYTE (*getYXStateByFun_Inf)(DWORD , DWORD);</code>
函数全局变量	<code>getYXStateByFun_Inf getYXStateByFun_Inf_fun;</code>
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的普通单点遥信的状态值
注册回调方法	<code>int regist_GetYxValueByFun_Inf(getYXStateByFun_Inf f)</code>
原型参数值	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的普通单点遥信的状态值
原型返回值	BYTE: 返回的普通单点遥信状态值(注意返回的状态值应是非 0 值为合, 0 值为分)
示例	<code>BYTE demoYX(DWORD func , DWORD inf){...}; regist_GetYxValueByFun_Inf(demoYX);</code>
函数原型定义	<code>typedef BYTE (*getDPYXStateByFun_Inf)(DWORD , DWORD);</code>
函数全局变量	<code>getDPYXStateByFun_Inf getDPYXStateByFun_Inf_fun;</code>
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的普通双点遥信的状态值
注册回调方法	<code>int regist_GetDPYxValueByFun_Inf(getDPYXStateByFun_Inf f)</code>
原型参数值	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的普通双点遥信的状态值
原型返回值	BYTE: 返回的普通双点遥信状态值(注意返回的状态值应是 0x01、0x11、0x00、0x10)
示例	<code>BYTE demoDPYX(DWORD func , DWORD inf) {...}; regist_GetDPYxValueByFun_Inf(demoDPYX);</code>
函数原型定义	<code>typedef WORD (*getYCValueByFun_Inf)(DWORD , DWORD);</code>
函数全局变量	<code>getYCValueByFun_Inf getYCValueByFun_Inf_fun;</code>
函数功能	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的遥测值
注册回调方法	<code>int regist_GetYCValue(getYCValueByFun_Inf f)</code>
原型参数值	通过 Map.cfg 配置文件中的(FUN , INF)获取相应的遥测值
原型返回值	WORD: 返回相关遥测值, 注意该值为一个 16 位的整数
示例	<code>WORD demoYC(DWORD func , DWORD inf) {...}; regist_GetYCValue(demoYC);</code>
函数原型定义	<code>typedef BOOLEAN (*setSystemTime)(tagTimeBCD *);</code>
函数全局	<code>setSystemTime setSystemTime_fun;</code>

变量	
函数功能	时间同步的回调函数
注册回调方法	iiint regist_SetSysTime(setSystemTime f)
原型参数值	主站下发的时间同步命令报文中解析出来的相关时间变量结构 tagTimeBCD
原型返回值	BOOLEAN: 相关命令下发是否执行[TRUE, FALSE]
示例	BOOLEAN demoSetSys(tagTimeBCD * timeBCD) {...}; regist_SetSysTime(demoSetSys);

函数原型定义	typedef BOOLEAN (*judgeAllowRemonCtrl)(void);
函数全局变量	judgeAllowRemonCtrl judgeAllowRemonCtrl_fun;
函数功能	是否允许远方进行遥控
注册回调方法	int regist_JudgeAllowRemonCtrl(judgeAllowRemonCtrl f)
原型参数值	无
原型返回值	BOOLEAN: [TRUE:允许, FALSE:不允许]
示例	BOOLEAN demoCheckEnableYkCmd() {...}; regist_JudgeAllowRemonCtrl(demoCheckEnableYkCmd);

函数原型定义	typedef BOOLEAN (*selectYkYtCmd)(BYTE ctlType ,DWORD fun , DWORD inf , BYTE cmdVal, BYTE* retCot);
函数全局变量	selectYkYtCmd selectYkYtCmd_fun;
函数功能	遥控遥调选择回调函数
注册回调方法	int regist_SelectCmdCtrl(selectYkYtCmd f)
原型参数值	BYTE ctlType: 用与区分是遥控还是遥调命令 [0,遥控][1,遥调] DWORD fun: 遥控/遥调点 func 号 DWORD inf: 遥控/遥调点对应的 inf 号 BYTE cmdVal:遥控/遥调命令下发的对应值, {0: 单点遥分,1: 单点遥合, 2: 双点遥分, 3: 双点遥合, 4: 遥降, 5: 遥升, 6: 遥停} BYTE* retCot:如果控制失败, 请填写相关传送错误原因
原型返回值	BOOLEAN: [TRUE1:控制成功, FALSE0:控制失败]
示例	BOOLEAN demoYKYTSelectCmd(BYTE ctlType, DWORD fun, DWORD inf, BYTE cmdVal, BYTE* retCot) {...}; regist_SelectCmdCtrl(demoYKYTSelectCmd);

函数原型定义	typedef BOOLEAN (*cancelYkYtCmd) (BYTE ctlType ,DWORD fun , DWORD inf , BYTE cmdVal, BYTE* retCot);
函数全局	cancelYkYtCmd cancelYkYtCmd_fun;

变量	
函数功能	遥控遥调撤销回调函数
注册回调方法	int regist_CancelCmdCtrl(cancelYkYtCmd f)
原型参数值	BYTE ctlType: 用与区分是遥控还是遥调命令 [0,遥控][1,遥调] DWORD fun: 遥控/遥调点 func 号 DWORD inf: 遥控/遥调点对应的 inf 号 BYTE cmdVal:遥控/遥调命令下发的对应值, {0: 单点遥分,1: 单点遥合 ,2: 双点遥分 ,3: 双点遥合 ,4: 遥降 ,5: 遥升 ,6: 遥停} BYTE* retCot:如果控制失败, 请填写相关传送错误原因
原型返回值	BOOLEAN: [TRUE1:控制成功 , FALSE0:控制失败]
示例	BOOLEAN demoYKYTCancelCmd(BYTE ctlType , DWORD fun , DWORD inf , BYTE cmdVal ,BYTE* retCot){...}; regist_CancelCmdCtrl(demoYKYTCancelCmd);

函数原型定义	typedef BOOLEAN (*executeYkYtCmd) (BYTE ctlType ,DWORD fun , DWORD inf , BYTE cmdVal, BYTE* retCot);
函数全局变量	executeYkYtCmd executeYkYtCmd_fun;
函数功能	遥控遥调执行回调函数
注册回调方法	int regist_ExecuteCmdCtrl(executeYkYtCmd f)
原型参数值	BYTE ctlType: 用与区分是遥控还是遥调命令 [0,遥控][1,遥调] DWORD fun: 遥控/遥调点 func 号 DWORD inf: 遥控/遥调点对应的 inf 号 BYTE cmdVal:遥控/遥调命令下发的对应值, {0: 单点遥分,1: 单点遥合 ,2: 双点遥分 ,3: 双点遥合 ,4: 遥降 ,5: 遥升 ,6: 遥停} BYTE* retCot:如果控制失败, 请填写相关传送错误原因
原型返回值	BOOLEAN: [TRUE1:控制成功 , FALSE0:控制失败]
示例	BOOLEAN demoYKYTExecuteCmd (BYTE ctlType , DWORD fun , DWORD inf , BYTE cmdVal ,BYTE* retCot){...}; regist_ExecuteCmdCtrl (demoYKYTExecuteCmd);