



IEC101

客户端开发包接口 API 使用说明书

版本：V1.0

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 60870-5-101 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 60870-5-101 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 60870-5-101 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	<code>void Initial();</code>
功能	101 服务串口连接初始化，在 <code>SZTS_IEC101_InitAllCfg()</code> 前调用
参数	无
返回值	无
示例	<code>Initial();</code>

函数声明	<code>void Init_IEC101Cfg();</code>
功能	101 服务配置初始化，在 101 服务通信前调用
参数	无
返回值	无
示例	<code>Init_IEC101Cfg();</code>

函数声明	<code>int SZTS_IEC101_InitAllCfg();</code>
功能	101 配置读取接口，在 101 服务通信前调用
参数	无
返回值	0 - 成功; -1 - 失败
示例	<code>SZTS_IEC101_InitAllCfg();</code>

函数声明	<code>int ClientStart()</code>
功能	101 服务基于串口协议传输主服务线程包裹函数
参数	无
返回值	1 - 成功; -1 - 失败

示例	ClientStart();
----	-----------------

函数声明	int SZTS_IEC101_TaskStart ()
功能	101 服务基于 TCP、UDP、串口协议传输服务线程包裹函数
参数	无
返回值	1 - 成功; 0 - 失败
示例	SZTS_IEC101_TaskStart();

函数声明	int SZTS_IEC101_Main()
功能	IEC101 规约报文处理主函数
参数	无
返回值	无
示例	SZTS_IEC101_Main();

函数声明	void SZTS_IEC101_Send(tagSendCtrl *ptSendCtrl);
功能	101 数据报文串口发送函数，用以发送相关命令报文
参数	ptSendCtrl : 已经赋值后的报文发送指针，具体定义可到 szts_101DataDef.h 文件查看
返回值	无
示例	SZTS_IEC101_Send(ptSendCtrl);

函数声明	int SZTS_IEC101_MsgRecv(BYTE *pbyMsgBuf)
功能	基于串口通信的数据接收方法
参数	pbyMsgBuf : 需要赋值的报文接收指针，用于接收报文内容
返回值	0 - 成功; -1 - 失败
示例	SZTS_IEC101_Uart_MsgRecv(pbyMsgBuf);

函数声明	void SZTS_IEC101_DataParse(BYTE *pbyMsg101);
功能	报文解析主函数，用于解析接收到的报文
参数	pbyMsg101 : 已经赋值后的报文接收指针
返回值	0 - 成功; -1 - 失败
示例	SZTS_IEC101_DataParse(pbyMsg101);

2.2 接口回调函数

函数原型定义	typedef void (*Svr_GetSend_Msg)(tagPSendCtrl);
函数全局变量	extern Svr_GetSend_Msg IEC101_GetSend_Msg_CallBack;
函数功能	打印串口发送的相关报文
注册回调方法	void Reg_Svr_GetSend_Msg(Svr_GetSend_Msg callBack);
原型参数值	Svr_GetSend_Msg: 已经赋值后的报文发送指针，具体定义可到 szts_101DataDef.h 文件查看

原型返回值	无
示例	void Svr_GetSend_Msg_Callback(tagPSendCtrl msg) {...}; Reg_Svr_GetSend_Msg(Svr_GetSend_Msg_Callback);

函数原型定义	typedef void (*Svr_GetRecv_Msg)(tagPMsgCtrl);
函数全局变量	extern Svr_GetRecv_Msg IEC101_GetRecv_Msg_Callback;
函数功能	打印串口接收到的相关报文
注册回调方法	void Reg_Svr_GetRecv_Msg(Svr_GetRecv_Msg callback);
原型参数值	Svr_GetRecv_Msg: 已经赋值后的报文接收指针, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	void Svr_GetRecv_Msg_Callback(tagPMsgCtrl msg) {...}; Reg_Svr_GetRecv_Msg(Svr_GetRecv_Msg_Callback);

函数原型定义	typedef void (*Svr_INIT_Resp)(tagINITResp);
函数全局变量	extern Svr_INIT_Resp IEC101_INIT_INFO_Callback;
函数功能	打印初始化结束响应内容
注册回调方法	void Reg_Svr_INIT_Res(Svr_INIT_Resp callback);
原型参数值	Svr_INIT_Resp: 已经赋值后的初始化响应信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	void Svr_INIT_Resp_Callback(tagINITResp resp){...}; Reg_Svr_INIT_Res(Svr_INIT_Resp_Callback);

函数原型定义	typedef void (*Svr_GI_Resp)(tagGIResp);
函数全局变量	extern Svr_GI_Resp IEC101_GI_INFO_Callback;
函数功能	打印总召响应内容
注册回调方法	void Reg_Svr_GI_Res(Svr_GI_Resp callback);
原型参数值	Svr_GI_Resp: 已经赋值后的的总召响应信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	void Svr_GI_Resp_Callback(tagGIResp resp){...}; Reg_Svr_GI_Res(Svr_GI_Resp_Callback);

函数原型定义	typedef void (*Svr_SPI_Resp)(tagSPIResp);
函数全局变量	extern Svr_SPI_Resp IEC101_SPI_INFO_Callback;
函数功能	打印通过总召上送的单点信息内容
注册回调方法	void Reg_Svr_SPI_Res(Svr_SPI_Resp callback);
原型参数值	Svr_SPI_Resp: 已经赋值后的单点信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	void Svr_SPI_Resp_Callback(tagSPIResp resp) {...}; Reg_Svr_SPI_Res(Svr_SPI_Resp_Callback);

函数原型定义	<code>typedef void (*Svr_DPI_Resp)(tagDPIResp);</code>
函数全局变量	<code>extern Svr_DPI_Resp IEC101_DPI_INFO_Callback;</code>
函数功能	打印通过总召上送的双点信息内容
注册回调方法	<code>void Reg_Svr_DPI_Res(Svr_DPI_Resp callback);</code>
原型参数值	Svr_DPI_Resp: 已经赋值后的双点信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_DPI_Resp_Callback(tagDPIResp resp) {...}; Reg_Svr_DPI_Res(Svr_DPI_Resp_Callback);</code>

函数原型定义	<code>typedef void (*Svr_MEASURE_Resp)(tagMeasureResp);</code>
函数全局变量	<code>extern Svr_MEASURE_Resp IEC101_MEASURE_INFO_Callback;</code>
函数功能	打印通过总召上送的测量值信息内容
注册回调方法	<code>void Reg_Svr_MEASURE_Res(Svr_MEASURE_Resp callback);</code>
原型参数值	Svr_MEASURE_Resp: 已经赋值后的测量信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_MEASURE_Resp_Callback(tagMeasureResp resp) {...}; Reg_Svr_MEASURE_Res(Svr_MEASURE_Resp_Callback);</code>

函数原型定义	<code>typedef void (*Svr_ELECTRIC_Resp)(tagElectricalEnergyResp);</code>
函数全局变量	<code>extern Svr_ELECTRIC_Resp IEC101_ELECTRIC_INFO_Callback;</code>
函数功能	打印通过电能量召唤上送的累计量值信息内容
注册回调方法	<code>void Reg_Svr_ELECTRIC_Res(Svr_ELECTRIC_Resp callback);</code>
原型参数值	Svr_ELECTRIC_Resp: 已经赋值后的累计量信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_ELECTRIC_Resp_Callback(tagElectricalEnergyResp resp) {...}; Reg_Svr_ELECTRIC_Res(Svr_ELECTRIC_Resp_Callback);</code>

函数原型定义	<code>typedef void (*Svr_ClockSyn_Resp)(tagClockSynResp);</code>
函数全局变量	<code>extern Svr_ClockSyn_Resp IEC101_ClockSyn_INFO_Callback;</code>
函数功能	打印时钟同步响应内容
注册回调方法	<code>void Reg_Svr_ClockSyn_Res(Svr_ClockSyn_Resp callback);</code>
原型参数值	Svr_ClockSyn_Resp: 已经赋值后的时钟同步响应信息结构体, 具体定义可到 szts_101DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_ClockSyn_Resp_Callback(tagClockSynResp resp) {...}; Reg_Svr_ClockSyn_Res(Svr_ClockSyn_Resp_Callback);</code>

函数原型定义	<code>typedef void (*Svr_YK_Resp)(tagYKResp);</code>
--------	--

函数全局变量	<code>extern Svr_YK_Resp IEC101_YK_INFO_CallBack;</code>
函数功能	打印遥控响应内容
注册回调方法	<code>void Reg_Svr_YK_Res(Svr_YK_Resp callback);</code>
原型参数值	Svr_YK_Resp : 已经赋值后的遥控命令响应信息结构体，具体定义可到 <code>szts_101DataDef.h</code> 文件查看
原型返回值	无
示例	<code>void Svr_YK_Resp_CallBack(tagYKResp resp) {...};</code> <code>Reg_Svr_YK_Res(Svr_YK_Resp_CallBack);</code>

函数原型定义	<code>typedef void (*Svr_YT_Resp)(tagYTResp);</code>
函数全局变量	<code>extern Svr_YT_Resp IEC101_YT_INFO_CallBack;</code>
函数功能	打印设点命令响应内容
注册回调方法	<code>void Reg_Svr_YT_Res(Svr_YT_Resp callback);</code>
原型参数值	Svr_YT_Resp : 已经赋值后的设点命令响应信息结构体，具体定义可到 <code>szts_101DataDef.h</code> 文件查看
原型返回值	无
示例	<code>void Svr_YT_Resp_CallBack(tagYTResp resp) {...};</code> <code>Reg_Svr_YT_Res(Svr_YT_Resp_CallBack);</code>