



CDT

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 新部颁 CDT 最新版本开发而成，采用 C 语音编写，可跨平台使用。

为了最大限度地简化使用，完全封装了 新部颁 CDT 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 新部颁 CDT 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	CDT 服务串口连接初始化，在 <code>SZTS_CDT_InitAllCfg()</code> 前调用
参数	无
返回值	无
示例	<code>Initial();</code>

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

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

函数声明	<code>int ClientStart()</code>
功能	CDT 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_CDT_TaskStart ()
功能	CDT 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_CDT_TaskStart();

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

函数声明	void SZTS_CDT_Send(tagSendCtrl *ptSendCtrl);
功能	CDT 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_DataDef.h 文件查看
返回值	无
示例	SZTS_CDT_Send(ptSendCtrl);

函数声明	int SZTS_CDT_MsgRecv(tagPMsgCtrl ptMsgCtrl)
功能	基于串口通信的数据接收方法
参数	tagPMsgCtrl : 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_CDT_Uart_MsgRecv(ptMsgCtrl);

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

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 CDT_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_DataDef.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 CDT_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_DataDef.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_YC_Resp)(tagYCResp);
函数全局变量	extern Svr_YC_Resp CDT_YC_INFO_CallBack;
函数功能	打印遥测信息内容
注册回调方法	void Reg_Svr_YC_Res(Svr_YC_Resp callback);
原型参数值	Svr_YC_Resp: 已经赋值后的遥测信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	void Svr_YC_Resp_CallBack(tagYCResp resp){...}; Reg_Svr_YC_Res(Svr_YC_Resp_CallBack);

函数原型定义	typedef void (*Svr_ALLYC_Resp)(tagALLYCResp);
函数全局变量	extern Svr_ALLYC_Resp CDT_ALLYC_INFO_CallBack;
函数功能	打印总加遥测信息内容
注册回调方法	void Reg_Svr_ALLYC_Res(Svr_ALLYC_Resp callback);
原型参数值	Svr_ALLYC_Resp: 已经赋值后的的总加遥测信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	void Svr_ALLYC_Resp_CallBack(tagALLYCResp resp){...}; Reg_Svr_ALLYC_Res(Svr_ALLYC_Resp_CallBack);

函数原型定义	typedef void (*Svr_YX_Resp)(tagYXResp);
函数全局变量	extern Svr_YX_Resp CDT_YX_INFO_Callback;
函数功能	打印遥信信息内容
注册回调方法	void Reg_Svr_YX_Res(Svr_YX_Resp callback);
原型参数值	Svr_YX_Resp: 已经赋值后的遥信信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	void Svr_YX_Resp_CallBack(tagYXResp resp) {...}; Reg_Svr_YX_Res(Svr_YX_Resp_CallBack);

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 CDT_ELECTRIC_INFO_Callback;</code>
函数功能	打印电能脉冲计数值信息内容
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: 已经赋值后的电能脉冲计数值信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_ELECTRIC_Resp_CallBack(tagElectricResp resp) {...}; Reg_Svr_ELECTRIC_Res(Svr_ELECTRIC_Resp_CallBack);</code>

函数原型定义	<code>typedef void (*Svr_SOE_Resp)(tagSOEResp);</code>
函数全局变量	<code>extern Svr_SOE_Resp CDT_SOE_INFO_Callback;</code>
函数功能	打印事件顺序记录 SOE 内容
注册回调方法	<code>void Reg_Svr_SOE_Res(Svr_SOE_Resp callback);</code>
原型参数值	Svr_SOE_Resp: 已经赋值后的事件顺序记录 SOE 信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_SOE_Resp_CallBack(tagSOEResp resp) {...}; Reg_Svr_SOE_Res(Svr_SOE_Resp_CallBack);</code>

函数原型定义	<code>typedef void (*Svr_STATUS_Resp)(tagStatusResp);</code>
函数全局变量	<code>extern Svr_STATUS_Resp CDT_STATUS_INFO_Callback;</code>
函数功能	打印子站工作状态内容
注册回调方法	<code>void Reg_Svr_STATUS_Res(Svr_STATUS_Resp callback);</code>
原型参数值	Svr_STATUS_Resp: 已经赋值后的子站工作状态信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_STATUS_Resp_CallBack(tagStatusResp resp) {...}; Reg_Svr_STATUS_Res(Svr_STATUS_Resp_CallBack);</code>

函数原型定义	<code>typedef void (*Svr_YKYT_Resp)(tagYKYTResp);</code>
函数全局变量	<code>extern Svr_YKYT_Resp CDT_YKYT_INFO_Callback;</code>
函数功能	打印遥控、升降返校内容
注册回调方法	<code>void Reg_Svr_YKYT_Res(Svr_YKYT_Resp callback);</code>
原型参数值	Svr_YKYT_Resp: 已经赋值后的遥控、升降返校信息结构体, 具体定义可到 szts_DataDef.h 文件查看
原型返回值	无
示例	<code>void Svr_YKYT_Resp_CallBack(tagYKYTResp resp) {...}; Reg_Svr_YKYT_Res(Svr_YKYT_Resp_CallBack);</code>

函数原型定义	<code>typedef void (*Svr_ClockSyn_Resp)(tagClockResp);</code>
函数全局变量	<code>extern Svr_ClockSyn_Resp CDT_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	<code>Svr_ClockSyn_Resp</code> : 已经赋值后的时钟同步响应信息结构体, 具体定义可到 <code>szts_DataDef.h</code> 文件查看
原型返回值	无
示例	<code>void Svr_ClockSyn_Resp_CallBack(tagClockResp resp) {...};</code> <code>Reg_Svr_ClockSyn_Res(Svr_ClockSyn_Resp_CallBack);</code>