Air724 数据模板和任务支持的 API 文档

深圳市银尔达电子有限公司

版本：Air724 数据模板和任务支持的 API 文档 V0.1

发布时间：2022 年 06 月 16 日星期五

■ 版权声明

■ 说明

本文档用于记录、指导研发流程和人员基本文档。

公司网站：http://www.yinerda.com

联系电话： 0755-23732189

联系地址： 深圳市龙华区大浪街道华宁路 117 号中安科技园 A 栋 2003-2005

版本记录：

版本	时间	备注	描述
V0.1	20220616		初版

目录

本文档记录银尔达 DTU 透传固件内部 API。本文档适合 air724、air820 系列设备，固件版本大于 YED_DTU_1.2.1。

这些 API 可以用于任务和数据模板。利用 API 可以实现利用银尔达 DTU 配置平台配置参数，同时可以自己灵活的利用 DTU 资源，做自己特殊的逻辑。可以完成的工作:

1、接收自定义的服务器数据，控制输出，查询输入；
2、接收服务器自定义的数据格式（比如 json）转换成串口的自定义数据格式(比如 modbus)；
3、接收串口的数据，定时打包，比如定时 5 分钟上报一次数据，分钟之前的数据都丢弃；
4、接收串口数据，对比串口数据是否有变化，变化后上传，没变化 5 分钟定时上传
5、接收串口的数据，转换成服务器需要的格式；
6、自定义轮询命令，然后把轮询结果转成 json 格式发给服务器等；
7、自定义 ADC 采集模式；比如修改电阻后，采集原始电压，做自己的逻辑上报
8、自定义输入采集模式；比如检查到输入后，控制开关，并且上报数据到服务器；
9、自定义继电器等控制模式逻辑；比如收到服务器命令后，控制继电器开多久，关多久；
10、Air820 可以完全接管 GPS 模块，实现 GPS 数据采集和低功耗采集；
11、可以实现 TTS 播报，比如支付功能开启继电器后，播放一段语音提醒；
12、实现低功耗处理，只使用串口 1 和不用串口的数据，可以做保持服务器连接低功耗(不做低功耗 5V 30ma ，做低功耗 5V 10ma)。
13、实现上传数据，拼接特殊 ID，比如拼接 IMEI，信号质量 CSQ 等；
14、自定义逻辑，完全截取网络和串口数据，自己处理全部的数据(边缘计算)。

一、原生常用 API

DTU 任务和数据模板能用到的大概是如下函数，其他函数和具体使用方法也可以去查询合宙官方 API。

作用	函数	参数	说明
重启模块	sys.restart		
推送消息	sys.publish		
等待消息	sys.waitUntil(id, ms)		
延迟函数	sys.wait(ms)		
获取 IMEI	misc.getImei()		
获取卡号	sim.getIccid()	返回 nil、ICCID 号	
获取信号强度	net.getRssi()		
网络同步时间	ntp.timeSync(ts)	ts 自动同步周期，1~24	
获取时间	misc.getClock()		
string 转换成 hex	string.fromHex(str)	输入”010203” 返回 0x01 0x02 0x03	
hex 转换成 string	string.toHex(str)	输入 0x01 0x02 0x03 返回”010203”	
Modbus CRC16 校验	crypto.crc16		
小数据储存	nvm.set	适合少量数据储存	
	nvm.get		
数据位操作	bit 库	左移右移，或 与 非等操作	

二、银尔达 DTU 透传固件特殊 API

2.1、硬件相关 API

作用	函数	参数
控制数字输出	per.SetDoOut(id,onoff)	Id：输出 ID，1~n onoff：GPIO 0 输出关/低，1 输出开/高 返回:nil、1
获取数字输出状态	per.GetDoOut(id)	Id：输出 ID，1~n 返回:nil、0、1
获取数字量输入状态	per.GetDiInputById(id)	Id：输入 ID，1~n 返回:nil 、0、1
控制可控电源输出	per.CtlPowerOut(on,t)	on：0 关闭，1 输出 t：延迟时间，如果设置了，当 on=1，先关闭，在延迟，然后在开启输出 返回:nil 或者 1
读取模拟量 ADC 原始电压	per.GetAdcSrcValByAdcId(adcid)	adcid：ADC 通道 ID，1~n 返回:nil 、原始电压 mv
	与外部电阻无关，读取的是 ADC 上面原始的电压 ADC 是模块 ADC 通道 ID，不是硬件上面的输入编号，比如 YED_RN1222 的电流通道 1 对应的 adcid=2,电流通道 2 对应的 adcid=3	
读取模拟量计算数据	per.GetAdcGatherValByAdcId(adcid)	adcid：ADC 通道 ID，1~n 返回:nil 、原始电压 mv
	与外部电阻有关，ADC 通过硬件电阻转换后的数据，可能是电流、电压或者电阻等	
重新初始化 ADC 参数	per.AdcReInit(adcid,scale,ty,r1,r2,mul,dma,dmi)	参数: adcid:ADC 通道 ID，1~n scale:ADC 管脚最大电压，取值为 adc.SCALE_1V250,adc.SCALE_2V444,adc.SCALE_3V233,adc.SCALE_5V000 ty:采集类型，per.ADC_A ,per.ADC_V r1，采样电阻 r2，总电阻,如果是电流采集设置为 0 mul：有效数据扩大倍数 dma:有效数据最大值 dmi:有效数据最小值
	1、硬件上面有默认采样电阻 2、如果直接更换了电阻，想采集其他数据值，可以重新初始化，然后再调用 API 获取 3、电流模式，模拟量 ADC 原始电压=I*r1；电压模式，模拟量 ADC 原始电压=V*r1/r2 4、Mul，dma，dmi 控制读取模拟量计算数据的表现形式， 比如转换实际数据=14.1234ma，获得的计算数据=14.1234*mul，如果计算数据>dma,等于 dma,如果计算数据<dmi,等于 dmi	

2.2、定位相关 API

作用	函数	参数
获取基站定位	userlbs.GetLbs()	返回 2 个参数，基站经度、基站纬度
	字符串，失败全是 0	
获取 GPS 定位	usergps.GetGps()	返回 4 个参数，GPS 经度方向，经度，纬度方向，纬度
	字符串，失败全是 0	
获取 GPS 结构	usergps.GetGpsFun()	返回""或者 nil 或者 gps 对象 只有 gps 对象有效
	air820 系列支持，gps 结构，可以调用函数如 gps.getAltitude() gps.getSpeed() gps.setRunMode()	
完全接管 GPS 逻辑	autoGPS.GpsKindInit()	自动识别 GPS 模块类型
	autoGPS.GpsGetInfo()	返回值:kind,baudrate,gps,agps 分别是:类型，波特率，GPS 对象，辅助 GPS 对象
	DTU 在不开启定位功能的时候，可以用这 2 个函数获取到 gps 对象，完全接管 GPS 功能包括实现低功耗休眠等，通过 gps 对象访问 GPS 资源	

2.3、数据交互相关 API

作用	函数	参数
调用串口对外发送数据	prouart.ProuartSetSendChace(uid,s)	参数:uid 是串口编号范围 1,2,3，对应的硬件类型，可以查对应硬件规格书 s 是需要发送的字符串，类型为 string
	msg="UART"..uid.."_NEED_SEND" sys.publish(msg)	msg 是固定格式，uid 是需要发送的串口编号
	如果不发送 msg，串口发送数据会延迟。发送的数据不经过数据模板	
停止 DTU 内部处理串口接收数据	prouart.ProuartStopProReciveCache(on)	参数:1 表示停止，0 表示启用
	停止内部处理接收的串口数据，可以用任务读取缓存，截取接收到的数据自己处理逻辑	
获取串口缓存数据	prouart.ProuartGetReciveChaceAndDel(uid)	参数:uid 是串口编号范围 1,2,3，对应的硬件类型，可以查对应硬件规格书 返回:nil 、有效数据
	内部使用 talbe 缓存的 需要先停止,获取后会自动删除缓存被获取的数据，如果要获取全部，循环获取，直到返回 nil	
给网络通道发送数据	pronet.PronetGetNetState(netid)	判断网络是否连接成功 参数 netid：网络通道，1~n 返回:nil ,0,1 1 表示连接成功
	pronet.PronetInsertSendChache(netid,senddata)	参数: netid：网络通道，1~n senddata:需要发送的数据，string 格式
	sendmsg="UART_DATA_TO_NET"..netid	msg 是固定格式，netid 是需要发送的网络通

	sys.publish(sendmsg)	道编号
	先判断网络是否连接，然后压入发生数据到缓存，然后发生 msg 如果不发送 msg，网络发送数据会延迟。发送的数据不经过数据模板 推送的消息会经过发送数据模板	
停止 DTU 内部处理网络接收数据	pronet.PronetStopProNetRecive(on)	参数:1 表示停止，0 表示启用
	停止内部处理接收的网络数据，可以用任务读取缓存，截取接收到的数据自己处理逻辑	
读取 DTU 内部网络接收数据缓存	pronet.PronetGetReciveChacheAdnDel(id)	参数:id 是网络通道 ID，1~n 返回:nil、有效数据
	内部使用 talbe 缓存的 需要先停止,获取后会自动删除缓存被获取的数据，如果要获取全部，循环获取，直到返回 nil	

2.4、工具类 API

作用	函数	参数
Hex 字符串转换成 table	usertool.ToolHexStrToTable(str)	参数:str,hex 的字符串，如 0x00 0x01 0xaa 返回:nil 或者 table
	本函数专门处理 16 进制的 hex 数据，转换成 table，然后可以用下标的方式引用数据，下标从 1 开始。 比如输入数据是 0x00 0x01 0xaa。返回的数据是 t,那么 t[1]=0x00,t[2]=0x01	
把 table 转换成 hex sting	usertool.ToolTableToHexStr(t)	参数:t 需要转换的 table 返回:16 进制的 string
	比如输入数据是 t={0x00 0x01 0xaa}。返回的数据是 str=0x00 0x01 0xaa	
获取 Modbus CRC16	usertool.ToolGetModbusCRC16(ptr, dlen)	参数 ptr：16 进制的 table dlen：字符串的长度 返回:crc16 小端格式
	获取的 modbus 数据的数据先用 usertool.ToolHexStrToTable(str)转换	
校验 Modbus CRC16	usertool.ToolCheckModbusCRC16(ptr, dlen,mode)	参数: ptr：16 进制的 table dlen：字符串的长度 Mode：0：CRC16 低字节在前 1:CC16 高字节在前 返回:0 校验失败 1 校验成功
	获取的 modbus 数据的数据先用 usertool.ToolHexStrToTable(str)转换	
和校验	usertool.ToolChecksum(data, dlen)	参数:data :16 进制的 table 返回:和校验
	数据先用 usertool.ToolHexStrToTable(str)转换	
打印 hex 数据	usertool.ToolDebugPrintfHexTable(ptr, dlen)	参数：ptr:16 进制的 table
	数据先用 usertool.ToolHexStrToTable(str)转换 只能打印 16 进制的 table	

控制设备是否休眠	usertool.ToolUserCtrWake(sta)	参数:sta 1 唤醒，0 休眠
	只有使用 uart1 的设备才能调用休眠，进入休眠模式，否则其他串口的设备会丢失数据 进入休眠后，模块会保持网络连接，有数据和接收网络数据会唤醒。 以 YED-D724L1 为例，可以做到 5V 8ma ，12V 3ma 左右的平均功耗	
控制 DTU 不检查内部错误	Process.ProcessStopProError(on)	参数:on:1 停止检查
	DTU 内部停止检查错误，主要是网络连接不上后自动重启设备。 通过任务实现，需要实现超低功耗数据采集的时候设置，关闭内部自动重启功能，在任务里面通过数据飞行模式的的方式进入低功耗，可以做到 5V 1.3ma 左右平均功耗	