ZLGi 致远电子



2023/5 ^{第5期} 月刊



ZAM系列高精度测温模块

原来温度采集设计可以如此简单!

- ☑ 即插即用,可直接通过Ⅰ²C输出温度信号,跳过模拟电路设计难点,节省人力和项目周期
- ☑ 出厂自带校准,精度高、温漂小、一致性好
- ☑ 供应链管理简单,免去分立电路管理物料管理麻烦的烦恼
- ☑ 系列拥有丰富的产品供您选择,满足您热电阻测量、热电偶测量、抗干扰、多通道等多种应用需求



ZLG致远电子测温系列模块是针对行业测温电路设计痛点开发的一系列温度信号采集产品,拥有热电阻信号采集、热电偶信号采 集功能。使用该系列产品,用户可免于繁杂的测温模拟电路设计,即插即用,快速高精度实现温度采集电路功能。

产品型号	传感器类型	通道数	精度	温漂系数	采样率	隔离	供电	封装	通讯接口
ZAM6222	热电阻PT100	2	0.02%±0.1°C	10ppm	10SPS	非隔离	3.3V	DIP16	IIC
ZAM6222B	热电阻PT1000	2	0.02%±0.1°C	10ppm	20SPS	非隔离	3.3V	DIP16	IIC
ZAM6218A	热电偶T型、K型	8	0.02%±0.1°C	15ppm	20SPS	非隔离	3.3V	DIP24	IIC
ZAM6228	热电阻PT100	8	0.02%±0.1°C	15ppm	20SPS	非隔离	3.3V	DIP24	IIC
SM6228	热电阻PT100	8	0.02%±0.1°C	15ppm	20SPS	非隔离	3.3V	SiP	IIC
TPS02RAH	热电阻PT100	2	0.02%±0.1°C	10ppm	5SPS	4000Vrms	3.3V	DIP16	IIC

应用行业



SM6228/ZAM6228

ZAM6218

ZAM6222

锂电生产测试设备 环境温箱



充电桩温度监测 7AM6222B ZAM6222





测温二次仪表 TPS02RAH ZAM6222 ZAM6222B





7AM6222 ZAM6222B

工业自动化测温 TPS02RAH ZAM6222 ZAM6222B

致远电子官方网站

致远电子官方微信

CONTENTS



技术平台

EsDA 平台

【EsDA 应用】 蓝牙客户端节点详解	04
【EsDA协议转换】5分钟快速实现图像人脸检测	09
【EsDA应用】蓝牙服务端节点详解	12
ZLG 云平台	
【产品应用】ZWS 物联网云平台如何下发命令给设备?	17
【产品应用】AWorksLP例程介绍(MR6450)-SD卡	19
【产品应用】基于 CAN 智慧云实现远程 ECU 升级	22
【产品应用】CATCOM-100搭配ZWS物联网云平台,实现Modbus设备上云	24
【产品应用】如何利用 IoT 云平台统计设备数据?—基础篇	25
【产品应用】如何利用 IoT 云平台统计设备数据?—进阶篇	26
【产品应用】如何利用 IoT 云平台统计设备数据?— 下载篇	28
【产品应用】基于CAN 智慧云实时查看汽车 GPS 轨迹	29

边缘计算

核心板

互联互通

无线通讯	
【技术分享】蓝牙技术的前世今生	
【技术分享】物联网ZigBee技术详解②—组网类型	34
【技术分享】一文读懂蓝牙模块的工作模式	35
接口与协议转换	
【产品应用】CANopen主站如何获取Modbus设备的数据	

感知控制

电源与隔离

【技术分享】自动收发 RS-485 偏置电阻与终端电阻的选用,你知多少?	38
【技术分享】双电源并用问题与解决方案	40
【产品应用】隔离收发器全"芯"升级,你知道哪些?	42
数据采集	
【技术分享】如何提高多路热电偶测量的通道一致性	43

ZLGi微文摘

技术平台 ▼

【EsDA应用】 蓝牙客户端节点详解

ZLG 致远电子 2023-05-10 11:32:31

蓝牙作为一种近距离无线通信技术,被广泛使用在汽车,智能家居和 医疗等领域中。本文主要介绍如何通过 EsDA 开发套件,快速实现蓝 牙客户端的订阅以及读写等操作。

简介

蓝牙是一种近距离无线通信技术,可以免费运行在 2.4G 频段,所以通 常广泛使用在物联网,移动终端和智能家居等行业中。我们 EsDA 大家族里 面自然也有对蓝牙功能的实现,且只需要很短的时间就可实现蓝牙通信, 功能好用简单又便捷。

本文主要基于 MPC-ZC1 的开发板,以及可将节点进行拖拽连接的图形 化设计工具 AWFlow Designer,着重对 UIS8910DM 平台的板载蓝牙的配套 节点以及示例进行讲解。蓝牙节点一共分为配置节点类型 ble_server, ble_ client2 个,客户端节点类型 ble_client_in, ble_client_out, ble_client_ subscribe3 个以及服务端节点类型 ble_server_adv, ble_server_service2 个;因本文篇幅的原因所以主要围绕着客户端节点的使用,以下进行详细 介绍。

节点简介

1. 配置节点

1.1 ble_client 节点

1.1.1 属性



• 扫描类型: 启动扫描时需要的扫描类型配置:

值为0表示被动扫描(仅仅接收广播,不会发送扫描请求);

值为1时表示主动扫描(发送扫描请求数据给广播的服务端,服务端 收到后会返回扫描响应数据)。

- 扫描窗口的时间:扫描窗口时间的配置;
- 扫描间隔的时间:扫描间隔时间的配置;

注意: 扫描窗口时间必须小于扫描间隔时间。

• 扫描过滤策略:扫描过滤策略的配置:

1表示只处理地址在白名单中的数据包,及发给自己的可连接定向广播;
 2表示处理所有非定向广播和可解析私有地址以及发给自己的可连接定向广播;

3 表示处理地址在白名单中和可解析私有地址以及发给自己的可连接定向广播。

目标服务端设备名称:目标服务端设备本地名称; • SMP 配对模式与参数:

SMP 配对模式	参数	安全等级排名
SECURE_CONNECTION_PASSKEY	KEYBOARD_DISP	最高
SECURE_CONNECTION_JUSTWORK	NOIN_NOOUT	次之
LEGACY_PASSKEY	KEYBOARD_DISP	第三
LEGACY_JUSTWORK	NOIN_NOOUT	最低

 扫描超时时间:表示持续扫描,直到与指定服务端建立连接。如果 值不为 0,则开启扫描后使能扫描定时器,超时后如果仍未与指定 服务端建立连接,则扫描停止。单位 ms
 配置节点无输入输出

2. 客户端节点

2.1 ble_client_in 节点 2.1.1 属性

编辑 ble_client_in 节点						
删除			取消		完成	Ż
						٠
●名称						
显示名称						
BLE客户端配置	ble_client			~	1	
服务UUID	Oxff11					
特征值UUID	0xff01					

- BLE 客户端配置: 绑定一个 ble_client 蓝牙客户端的配置节点;
- 服务 UUID:需要读的服务 UUID。若服务 UUID 为自定义的 16 字节 如 00 00 ff 11 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、4 两

个字节 0xff11;

 特征值 UUID:需要读的特征值 UUID。若特征值 UUID 为自定义的 16 字节如 00 00 ff 01 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、 4 两个字节 0xff01。

2.1.2 输入

- payload:可以使用"timer"节点作为前级节点,定时读取数据。
- 2.1.3 输出
- characteristic_uuid:将读取到的指定特征值的 UUID 输出;
- payload:将读取到的指定特征值的数组指针输出。类型为 uint8_t 类型的指针;
- payloadLength:将读取到的指定特征值的数据长度输出;
- desc_num: 将读取到的指定特征值的描述符个数输出。如果存在多个描述符,则存在多个名称为"desc_uuid_%d"、"desc_ value_%d"和"desc_value_length_%d"的属性,其中%d指代当前是第几个描述符;
- 举例:该特征存在2个描述符,则存在以下名称的属性:desc_ uuid_0,desc_value_0,desc_value_length_0和 desc_uuid_1, desc_value_1,desc_value_length_1,desc_uuid_0;
- desc_uuid_0:第一个描述符的 UUID,若为第二个描述符的 UUID 也可表示为 desc_uuid_1。以此类推;
- desc_value_0: 第一个描述符的值指针,若为第二个描述符的值也 可表示为 desc_value_1。以此类推;
- desc_value_length_0:第一个描述符的值的数据长度,若为第二 个描述符的值的数据长度也可表示为 desc_value_length_1。以此类 推。

2.2 ble_client_out 节点

2.2.1 属性

编辑 ble_client_out	节点
删除	取消完成
◎ 居性	•
▶名称	ble client out
显示名称	ble_client_out
BLE客户端配置	ble_client 🗸 🖌
服务UUID	Oxff11
特征值UUID	0xff01

- BLE 客户端配置:绑定一个 ble_client 蓝牙客户端的配置节点;
- 服务 UUID:需要读的服务 UUID。若服务 UUID 为自定义的 16字 节如 00 00 ff 11 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、4 两个字节 0xff11;
- 特征值 UUID: 需要读的特征值 UUID。若特征值 UUID 为自定义的 16 字节如 00 00 ff 01 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、 4 两个字节 0xff01。

2.2.2 输入

- payload:需要写入的数据,字符串类型。可以使用 fscript 节点对 payload 进行设置
- payloadLength:需要写入的数据长度。
- sink 类型节点没有输出功能
- 2.3 ble_client_subscribe 节点

2.3.1 属性

编辑 ble_client_subscribe 节点					
删除		取消	完成		
◆ 属性			٠		
▶名称	ble_client_subscribe				
显示名称	ble_client_subscribe				
BLE客户端配置	ble_client		ø		
订阅服务UUID	Oxff11				
订阅特征值UUID	Oxff01				
订阅模式	Enable Notification				

- BLE 客户端配置:绑定一个 ble_client 蓝牙客户端的配置节点;
- 服务 UUID:需要读的服务 UUID。若服务 UUID 为自定义的 16 字 节如 00 00 ff 11 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、4 两个字节 0xff11;
- 特征值 UUID: 需要读的特征值 UUID。若特征值 UUID 为自定义的 16 字节如 00 00 ff 01 00 00 10 00 80 00 00 80 5f 9b 34 fb,则传入第 3、 4 两个字节 0xff01;
- 订阅模式:值为0表示禁能通知和指示,值为1表示使能通知,值 为2表示使能指示。

pump 类型节点无输入数据

2.3.2 输出

- payload:订阅的特征值的数据内容;
- payloadLength: 订阅的特征值的数据长度;
- characteristic_uuid: 订阅的特征值的 UUID。

示例讲解

1. 硬件准备

如下图所示,准备一块 MPC-ZC1 开发板,并按照下图所示连接好硬件。 用于下载流图的 USB Type-C 口另外一端接到 PC 机上。





2. 软件准备

目前 MPC-ZC1 开发板作为客户端,对 BLE 调试宝 (手机 app) 服务端 (外 围模式)提供的服务进行订阅、读写等操作,依下图所示可根据用户需求随 意设置服务 UUID、特征值 UUID、特征数据以及通知数据,最后点击开启 广播,服务端 BLE 调试宝的配置就设置完成了。





随之修改手机本地蓝牙(服务端设备)名称,后期此名称用来配置 ble_ client 节点的属性参数目标服务端设备名称。

← 蓝牙	\bigcirc
蓝牙	
当前可被附近的蓝	于设备发现
设备名称	BLE-ZWT >
接收的文件	本地蓝牙名称修改 >

3. EsDA业务开发

本示例主要是将 MPC-ZC1 开发板作为蓝牙客户端,对 BLE 调试宝(蓝 牙服务端)指定服务下指定的特征值数据进行读写以及订阅的操作,具体实 现如下图。本例程只对服务端提供的一个服务的特征值进行了订阅和读写 操作,根据实际需求,可支持对多个指定服务下的指定特征值进行读写和 订阅的操作。



3.1 添加节点并绘制流图

添加 timer, ble_client_in, ble_client_out, ble_client_subscribe, fscript 节点如下图所示。

timer	ble_client_in		fscript
输出 〇—	- ●· 输入	##出●	● 輸入 輸出 ●
timer	fscript		ble_client_out
續出 ●	- ●• 뉇入	☆出 ●	● 输入
ble_client_subscribe	fscript		
输出 ●	-●• 输入	輸出 ●•	

3.2 配置节点

周期性的去读取蓝牙服务端指定服务下指定的特征值的特征数据。

双击 ble_client_in 节点的前驱 timer 节点,打开属性面板设置定时周期 (ms),MPC-ZC1 开发板周期性的去读取 BLE 调试宝指定服务下指定的特征值数据。修改完后点击右上角的完成即可保存属性配置。

	取消	完成
		0
0		
	r 700	取消 r c 00

双击 ble_client_in 节点中的 BLE 客户端配置,本属性配置在一份流图上只需要配置一次即可,后面 ble_client_out 以及 ble_client_subscribe

节点的属性 BLE 客户端配置都会链接到同一个 ble_server,不需再次重复 配置所以后面不再赘述。这里可以只修改属性值:目标服务端设备名称, 需注意的是此处的参数与服务端提供的信息需一致。修改完后点击右上角 的更新即可保存属性配置。

删除	Ę	刘消	更新	
✿ 属性				٠
◆名称	ble_client			
显示名称	ble_client			
扫描类型	active		~	
扫描窗口的时间	0x06			
扫描间隔的时间	0x30			
扫描过滤策略	0x00			
SMP配对模式	SECURE_CONNECTION_PASS	KEY	~	
目标服务端设备名 称	BLE-ZWT			
SMP配对模式的参数	KEYBOARD_DISP		~	
扫描超时时间	0			

双击 ble_client_in 节点,修改需要读取指定的服务 UUID 和特征值 UUID。修改完后点击右上角的完成即可保存属性配置。

编辑 ble_client_out 节点							
删除				取消		完成	
♀ 属性							٠
◆名称	ble_clien	t_out					
显示名称	ble_clien	t_out					
BLE客户端配置	ble_clie	nt			•	1	
服务UUID	Oxff11						
特征值UUID	0xff01						

双击 ble_client_in 节点的后继 fscript 节点,主要内容是对读取的特征 值 UUID 进行输出以及将服务端指定的服务 UUID 及特征值 UUID 下的特征 内容进行输出。

print("payloadLength is"+msg.payloadLength)

print("characteristic_uuid is" + msg.characteristic_uuid);

rb = rbuffer_create(msg.payload, msg.payloadLength)
print("length: "+ msg.payloadLength)

f = "recv ble server data is" for(var i=0; i<msg.payloadLength; i=i+1) {

d = rbuffer_read_uint8(rb);

f = join(",",f,d);

}

msg.payload = f print("data "+msg.payload)

周期性的去写入蓝牙服务端指定服务下指定的特征值的特征数据。

双击 ble_client_out 节点的前驱 *2 的 timer 节点,打开属性面板设置 定时周期 (ms),MPC-ZC1 开发板周期性的去写入 BLE 调试宝指定服务下指 定的特征值数据。修改完后点击右上角的完成即可保存属性配置。

编辑 timer 节点			
删除		取消	完成
♀ 属性			٠
●名称	timer		
显示名称	timer		
定时周期	20000		

因 ble_client_out 节点作为 sink 类型节点,所以 fscript 节点即为他的 前驱节点。双击 fscript,主要功能实现是设置 ble_client_out 节点的输入 数据: payload (要写入的数据),payloadLength(需要写入的数据长度)。

msg.payload = "123456789" set(msg.payloadLength, u8("9"))

双击 ble_client_out 节点,配置需要写入数据的服务 UUID 和特征值 UUID。修改完后点击右上角的完成即可保存属性配置。

编辑 ble_client_in 节	点		
删除		取消	完成
◎ 属性			٠
♥名称			
显示名称			
BLE客户端配置	ble_client	~	1
服务UUID	Oxff11		
特征值UUID	0xff01		

ZLGi微文摘

对蓝牙服务端指定服务下指定的特征值的订阅数据进行订阅操作。 双击 ble_client_subscribe 节点,配置需要订阅的服务 UUID,特征值 UUID 以及订阅模式。修改完后点击右上角的完成即可保存属性配置。

编辑 ble_client_sub	scribe 节点		
删除		取消	完成
♀ 雇性			٥
◆名称	ble_client_subscribe		
显示名称	ble_client_subscribe		
BLE客户端配置	ble_client	~	
订阅服务UUID	Oxff11		
订阅特征值UUID	0xff01		
订阅模式	Enable Indication		~

双击 ble_client_subscribe 的后继节点 fscript,主要功能是将接收到 的订阅数据进行输出打印。

rb = rbuffer_create(msg.payload, msg.payloadLength) print("length: "+ msg.payloadLength)

```
f = "recv subscribe data is"
for(var i=0; i<msg.payloadLength; i=i+1) {
    d = rbuffer_read_uint8(rb);
    f = join(",",f,d);
}</pre>
```

msg.payload = f print("data "+msg.payload) 3.3 下载验证 MPC-ZC1 开发板调试口打印信息如下:

[16:00:13.509]₩ ← ◆ characteristic uuid: 0xff01 characteristic value: 0x01 0x02 0x03 0x04 0x05 0x06 the 1th description value is null. payloadLength is6 characteristic_uuid is65281 length: 6 data recv ble server data is, 1, 2, 3, 4, 5, 6 subscribe characteristdata recv subscribe data is, 6, 5, 4, 3, 2, 1

BLE 调试宝操作日志显示如下:

÷	夕	围模式	设置	:
15:59:58.9	948>	[C0:FB:F9:78:5F:B4] 已连接		
16:00:00.0	067>	[C0:FB:F9:78:5F:B4] 读取特征值		
16:00:00.3	305>	[C0:FB:F9:78:5F:B4] Notification开	启	
16:00:04.1	148>	[C0:FB:F9:78:5F:B4] 读取特征值		
16:00:14.1	189>	接收[C0:FB:F9:78:5F:B4]: 31 32 33 38 39	34 35 36	5 37
16:00:14.1	193>	通知[C0:FB:F9:78:5F:B4]: 06 05 04	03 02 01	Ğ.
16:00:14.2	267>	[C0:FB:F9:78:5F:B4] 读取特征值		
16:00:24.1	147>	[C0:FB:F9:78:5F:B4] 读取特征值		

由操作日志可以看到,MPC-ZC1和手机蓝牙建立连接成功、MPC-ZC1 读取特征值,开启通知、接收 MPC-ZC1 写入新的特征值数据、接收通知。 至此示例实验已完成。

【EsDA协议转换】 5分钟快速实现图像人脸检测

原创 研发部 ZLG 致远电子 2023-05-17 11:36:30

人脸识别是一种高安全性的生物识别技术,广泛应用在安防、身份验 证等领域。而人脸检测是进行人脸识别的重要前提,本文主要介绍如 何基于 EsDA 设计的人脸检测节点,快速实现人脸检测功能。

简介

人脸识别是一种高安全性的生物识别技术,在安防、身份验证等领域 被广泛应用。

本文在【EsDA 应用】5 分钟快速实现摄像头图像采集 的基础上,使用 M6G2C 核心板,简单介绍如何基于 EsDA 设计的 face_detection 人脸检测 功能节点,快速实现人脸检测识别。

前期准备

若是刚开始接触 EsDA,可先阅读 EsDA 其他系列文章,从零开始搭建 环境和掌握基本开发流程,已有基础的可以跳过:

- EsDA MPC-ZC1 入门(一) —— 软件安装
- EsDA MPC-ZC1 入门(二)—— LED 控制
- 【EsDA 应用】5 分钟实现一个串口通信业务
- 【EsDA 应用】5 分钟快速实现摄像头图像采集

1. 硬件准备

1.1 准备一个摄像头,这里我们以OV7725 摄像头为例,搭配转接板使用。



1.2 准备一块 LCD 显示屏。

1.3 准备一张 SD 卡,用于存放人脸模型。

1.4 准备一块 M6G2C 核心板和 EPC-6G2C-L 评估板,并按照下图所示 连接好硬件。其中,eth0 网口用于下载流图,并连接 LCD 显示屏接口、电源, 以及将 OV7725 摄像头通过转接板连接到开发板背面的 CSI 接口上。





<u>立即购买</u>

2. 其它准备

将人脸检测相关的模型文件存放到 SD 卡中。

名称 修改日期 类型 大	小
☐ det1-int8.bin 2023/3/13 15:11 BIN 文件	8 KB
det1-int8.param 2023/3/13 15:11 PARAM 文件	2 KB
☐ det2-int8.bin 2023/3/13 15:11 BIN 文件	101 KB
🗋 det2-int8.param 2023/3/13 15:11 PARAM 文件	2 KB
接: det3-int8.bin 2023/3/13 15:11 BIN 文件	386 KB
[] det3-int8.param 2023/3/13 15:11 PARAM 文件	2 KB

节点介绍

这里我们需要使用到摄像头、图像格式转换以及人脸检测相关的 AWFlow 节点,包括在【EsDA 应用】5 分钟快速实现摄像头图像采集 中介 绍的 camera_video 节点、camera 配置节点、lcd_display 节点,以及本文 介绍的用于图像数据格式转换的 image_format_convert 节点和人脸检测节 点 face_detection。

1. image_format_convert节点

图像格式转换节点,用于将摄像头输出的图像转换为人脸检测节点支持的 RGB888 格式。

1.1 属性

• 输入图像格式:待转换图像的格式,一般是摄像头输出的原始数据

编辑 image_format_	convert 节点			
删除		取消	完成	
✿ 属性			0	
●名称	image_format_convert			
显示名称	image_format_convert			
输入图像格式	RGB565			
輸入图像宽度	640			
输入图像高度	480			
输出图像格式	RGB888			
输出图像宽度	640			
輸出图像高度	480			

2023年第五期

ZLGi微文摘

格式;

- 输入图像宽度:待转换图像的宽度;
- 输入图像高度:待转换图像的高度;
- 输出图像格式:转换后的图像的格式,必须指定为 RGB888;
- 输出图像宽度:转换后的图像的宽度;
- 输出图像高度:转换后的图像的高度。

1.2 输入

 image:需要进行转换的原始图像的地址,一般由 camera_video 节点提供输入;

1.3 输出

- image:指向转换完成后的目标格式图像数据的指针,一般由后级 节点接收并处理;
- format: 转换后的图像的格式;
- width:转换后的图像的宽度;
- height: 转换后的图像的高度。

2. face_detection节点

face_detection 是人脸检测节点,该节点对输入的图像数据进行人脸 检测处理,当检测到图像中存在人脸时,截取图像中的人脸数据并输出。 后级节点可以使用这个图像数据进行特征值提取、比对等操作。

2.1 属性

• 活体检测:是否进行活体检测;

编辑 face_detection	市点		
删除		取消	完成
◎ 属性			٠
●名称	face_detection		
显示名称	face_detection		
活体检测	disable		
模糊检测	disable		
人脸矫正	disable		
模型和参数路径	/flow/face_model/models-int8		
人脸图像主题	face		

- 模糊检测:是否进行模糊检测;
- 人脸矫正:是否进行人脸整理矫正;
- 模型和参数路径: 检测模型和参数文件路径;
- 人脸图像主题:需要进行人脸检测的图像的主题,此节点只对与此 属性匹配的输入图像进行检测。

2.2 输入

- image: 输入图像数据的缓冲区指针,必须是 RGB888 格式,可以 由 image_format_convert 转换后提供输入;
- width: 输入图像的宽度;
- height: 输入图像的高度;
- topic: 输入图像的主题。

2.3 输出

- image: 从输入图像中截取的人脸图像缓冲区;
- coord_x1:人脸的左上角横坐标,与后面三个坐标共同决定坐标系 上的两个点,这两个点可以确定图像中的人脸位置;

- coord_y1: 左上角纵坐标;
- coord_x2: 右下角横坐标;
- coord_y2: 右下角纵坐标。

业务处理

本应用主要实现 M6G2C 开发板采集摄像头图像,并在 LCD 显示屏上 进行显示。同时,将摄像头输出的原始数据进行格式转换后,输入到 face_ detection 节点中进行人脸检测。具体实现如下:

1. 添加节点并连线

首先我们需要添加一个 camera_video 节点用于采集图像,并输出到 lcd_display 节点中显示。然后,将 camera_video 节点输出的 RGB565 原 始图像在 image_format_convert 节点中转换为 RGB888 格式,接下来将转 换后的图像输入到 face_detection 中进行人脸检测,并将检测的结果输入 到 lcd_display 节点中。另外,增加两个 fscript 节点,用于指定图像主题 和打印坐标点信息。

2. 配置节点



2.1 配置 camera_video 节点

双击 camera_video 节点,并在配置项中选择添加一个新的 camera 节点。 在 camera 配置节点中,根据所使用的摄像头,分别设置设备名、输出

编辑 camera_video 节点						
删除		取消	完成			
♀ 属性			٠			
◆ 名称	camera_video					
显示名称	camera_video					
配置项	添加新的 camera 节点	~	2			
主题	主题					

图像的数据格式和图像宽度、高度。

然后更新配置,在 camera_video 节点的配置项中,选择上一步新增



的 camera 配置节点即可。

2.2 配置 image_format_convert

编辑 camera_video 节点						
删除		取消	完成			
✿ 属性			٥			
◆名称	camera_video					
显示名称	camera_video					
配置项	camera	~	ø			

使用 image_format_convert 节点目的是将图像格式转换为 RGB888, 因此,根据图像的信息进行设置即可。如果摄像头输出的原始数据即为 RGB888,那么这一步可以省略。

2.3 face_detec	tion 节点		
编辑 image_forma	t_convert 节点		
删除		取消	完成
✿ 属性			•
●名称	image_format_convert		
显示名称	image_format_convert		
输入图像格式	RGB565		~
输入图像宽度	640		
输入图像高度	480		
输出图像格式	RGB888		~

使用人脸检测节点需要将相关的模型文件存放到文件系统中,并在 face_detection 节点中指定模型存放的路径,这里我们存放到 SD 卡中,并 指定路径。人脸图像主题属性用于对输入的图像进行过滤,这里我们设置 为摄像头输出的图像的主题即可。



流图中标识为"1"的 fscript 节点主要是为图像指定一个主题,我们将 其设置为 face, 和 face_detection 节点中的主题一致。

msg.topic = "face"

另一个节点用于打印人脸检测结果的坐标点信息,方便我们对结果进 行验证。其内容如下:

print("(x1,y1)#: (" + msg.coord_x1 + ", " + msg.coord_y1 + ")") print("(x2,y2)#: (" + msg.coord_x2 + ", " + msg.coord_y2 + ")") 2.5 lcd_display 节点

lcd_display 节点用于显示图像,不需要我们进行配置。它会实时显示 camera_video 节点输出的图像,并且在检测到图像中存在人脸后,根据 face_detection 节点输入的坐标,在图像中将人脸框选显示出来。

3. 下载验证

因为 M6G2C 开发板是通过网口去下载流图,所以打开 AWFlow Designer 后需要配置目标板 IP 和端口号。然后点击下载运行即可。



流图下载完成后,通过LCD可以看到,我们已经成功从摄像头获取图像,并且成功在显示屏上显示。



当摄像头采集的图像中含有人脸时,face_detection 节点可以正常检 测到人脸并且输出准确的坐标点。



技术平台 ▼

【EsDA应用】 蓝牙服务端节点详解

ZLG 致远电子 2023-05-24 11:39:46

蓝牙作为一种近距离无线通信技术,被广泛使用在汽车,智能家居和 医疗等领域中。本文主要介绍如何通过 EsDA 开发套件,快速实现蓝 牙服务器的相关应用业务。

简介

前面文章讲解了 ble_client 的相关节点和业务处理,有兴趣的可以跳转链接【EsDA 应用】蓝牙客户端节点详解去查看。

本文主要基于 MPC_ZC1 开发板以及可将节点进行拖拽连接的图形化设 计工具 AWFlow Designer,着重对 UIS8910DM 平台的板载蓝牙的配套节点 以及示例进行讲解。本文除了对 ble_server 设备端节点进行详细阐述还有 以 MPC_ZC1 开发板做蓝牙服务器的相关应用业务介绍如:定义支持的服务, 发布服务器以便远程客户端可连接,响应读取和写入请求等



节点介绍

因蓝牙服务端节点中的属性数量过多,所以本节只介绍几个重点属性, 其他的属性内容详见 AWFlow Designer 上相关节点的帮助文档。

1. 配置节点

1.1 ble_server

创建 BLE 服务端设备,保存用户传入的广播配置。

1.1.1 属性

- 发现模式:值为0表示不可发现模式;值为1表示有限可发现模式;
 值为2表示普通可发现模式。
- 广播类型:值为0表示可连接的非定向广播;值为1表示可连接的 定向广播;值为2表示不可连接的非定向广播;值为3表示可扫描 的非定向广播。
- 广播信道:广播通道映射在 37、38、39 通道通信。值为 0 时表示通道 37 被使用,值为 1 时表示通道 38 被使用,值为 2 时表示通道 39 被使用。
- 广播的过滤策略:

值为0时表示处理所有设备的扫描和连接请求;

值为1时表示处理所有设备的连接请求和只有白名单上的设备的扫描 请求;

值为2时表示处理所有设备的扫描请求和只有白名单上的设备的连接 请求;

- 值为3时表示只处理在白名单上的设备的扫描和连接请求。
- SMP 配对模式的参数:

smp_pair_mode	io_cap	安全等级
SECURE_CONNECTION_PASSKEY	KEYBOARD_DISP	最高
SECURE_CONNECTION_JUSTWORK	NOIN_NOOUT	次之
LEGACY_PASSKEY	KEYBOARD_DISP	第三
LEGACY_JUSTWORK	NOIN_NOOUT	最低

- 支持的服务个数:支持提供的服务个数,需要与流图中添加的服务 数量一致。
- *配置节点无输入输出。
- 2. 服务端节点
- 2.1 ble_server_service
- 2.1.1 属性
- BLE 服务端配置节点:关联到的 BLE 服务端配置节点。
- 服务 UUID: 支持提供服务的 UUID。支持 2 个字节或 16 个字节, 按照小端模式依次用逗号隔开。
- 支持服务的特征相关属性:

特征值的格式	特征声明的性质	特征值的权限			
短 UUID	广播属性	可读			
修补长度	读属性	可写			
写通知	不带回复的写属性	读取认证			
组	带回复的写属性	读取授权			
冲突通知	通知属性	读取加密			
追加写	指示属性	读取认证文件			
	签名认证的写属性	写认证			
	扩展属性	写授权			
		写加密			
		写认证文件			
		只允许经典蓝牙访问			

2.1.2 输入

- payload: 输入数据。传入特征值数据,可以使用"fscript"节点作 为前级节点,设置需要写入的数据。
- payloadLength: 输入数据。传入特征值数据的长度,可以使用 "fscript"节点作为前级节点,设置需要写入的数据长度,最大不超 过 255。

2.1.3 输出

本节点无实质性数据输出,其消费者节点一般是 ble_server_adv 节点。 2.2 ble_server_adv

2.2.1 属性

- BLE 服务端配置节点:关联到的 BLE 服务端配置节点。
- 蓝牙本地设备名称:蓝牙服务端本地设备的名称,代码里自动将此参数加入到广播数据包的类型为 09 的小数据包中。

- 广播数据包最长31个字节,而这31个字节又被拆分成很多小数据包, 每个小数据包代表一类广播数据。
- 每个小数据包是由: length(数据长度)+type(类型)+data(数据内容) 构成的。
- 需要广播的每个小数据包的数据长度:表示需要广播的每个小数据
 包的数据长度(数据内容的字节长度 +1)。因为格式是 csv,所以多个数据包的数据长度按顺序用逗号来隔开。
- 需要广播的每个小数据包的数据类型:表示需要广播的每个小数据
 包的数据长度(数据内容的字节长度+1)。多个数据包的数据类型按顺序用逗号来隔开。
- 扫描响应的每个小数据包的数据内容:表示扫描响应的每个小数据 包的数据内容。多个数据包的数据内容按顺序用逗号来隔开。
- 扫描响应的每个小数据包的数据类型:表示扫描响应的每个小数据
 包的数据类型,多个数据包的数据类型按顺序用逗号来隔开。具体
 的广播数据包类型详细信息可在网上进行查阅。
- 数据类型为 09 表示本地设备名称即属性蓝牙本地设备名称,现已 被后续代码自动处理加入到了广播数据包中。计算长度时需注意, 如若广播数据包超过最长 31 个字节的内容,则截断。
- 扫描响应的每个小数据包的数据内容:表示扫描响应的每个小数据 包的数据内容。多个数据包的数据内容按顺序用逗号来隔开。

如:需要广播的每个小数据包的数据长度为 0x02;需要广播的每个 小数据包的数据类型为: 0x01;需要广播的每个小数据包的数据内容为: 0x06。那么算上蓝牙本地设备名称:(BLE-ZC1)0x42,0x4c,0x45,0x2d,0x5a,0x4 3,0x31,

- 那么最终的广播数据包为 0x08(数据长度),0x09(数据类型),
- 0x42,0x4c,0x45,0x2d,0x5a,0x43,0x31(数据内容 BLE-ZC1),
- 0x02(数据长度),0x01(数据类型),
- 0x06(数据内容),

2.2.2 输入

- ble_server_adv 节点的输入并无实际意义的输入,由 ble_server_ service 节点作为前级节点,进入当前节点的 on_data 虚函数中开启 广播。
- * sink 类型节点无输出功能

业务处理

定义支持的服务:由 ble_server_service 节点添加服务 UUID 和特征 值 UUID 以及特征值数据;发布服务器,以便远程客户端可连接:由 ble_ server_adv 节点开启广播(在一个运行程序中不能多次开启广播)后服务端 蓝牙准备完成,开始等待客户端蓝牙设备连接;以及后续的响应读取和写 入请求。实现框图如下所示。



1. 资源准备

硬件上依旧如【EsDA 应用】蓝牙客户端节点详解使用 MPC_ZC1 开 发板作为服务器去广播数据,软件上使用可拖拽连线的图形化设计工具 AWFlow Designer 去设计流图以及手机 app(BLE 调试宝)作为客户端进 行蓝牙通信。



2. 添加节点并绘制流图

添加 timer, gpio, trigger, fscript, ble_server_service 和 ble_ server_adv 节点到画布中,如下图所示。



3. 配置节点

添加多个蓝牙服务端的服务和特征值数据,并开启蓝牙广播。需注意 的是本蓝牙模块在程序运行时不能重复多次开启蓝牙广播,否则会打开蓝 牙失败。

双击 gpio 节点的前驱 timer 节点,打开属性面板设置定时周期 (ms), 定时去读取 gpio 的引脚电平状态。修改完后点击右上角的完成即可保存属 性配置。



ZLGi微文摘

双击 gpio 节点,配置属性如下图所示,根据需求选择所需的 gpio 引脚,本章节使用的是 P4.5,并设置为浮空输入,用户可根据自身需求设置 该 gpio 引脚号。修改完后点击右上角的完成即可保存属性配置。

编辑 gpio 节点				
删除			取消	完成
◎ 屋性				٠
●名称	gpio			
显示名称	anio			
200100	900			
	P4.5 O	OP4.6		
	P4.70	OP4.8	l.	
	P4.90	OP5.0	Ì	
	P5.1 ()	OP5.4		
	P5.5 〇	OP5.6		
	P5.7 ()	OP5.8		
	P5.9O	OP6.0		
	P6.10	OPIN_GPIO_7		
	PIN_AP_JTAG_TCKO	OPIN_AP_JTAG	TRST	
	PIN_AP_JTAG_TMSO	OPIN_AP_JTAG	_TDI	
	PIN_AP_JTAG_TDOO	OPIN_GPIO_13		
	PIN_GPIO_14O	OPIN_GPIO_15		
	PIN_SDMMC1_CMDO	OPIN_SDMMC1	_DATA_0	
	PIN_SDMMC1_DATA_1O	OPIN_SDMMC1	_DATA_2	
	PIN_SDMMC1_DATA_3O	OPIN_SIM2_CLI	K	
L.	PIN_SIM2_DIOO	OPIN_SIM2_RS	1	
GPIO序号	27			
模式	float			

双击 gpio 的后继节点 fscript,功能代码内容的主要目的是让后续的 trigger 节点只能被触发一次,因 P4.5 的引脚默认是低电平,且 trigger 节 点被触发只能收到 msg.trigger_stop 属性消息,所以具体代码如下。

if (msg.payload == 1) {
 msg.trigger_stop = "reset";

}

双击 trigger 节点,本节点主要目的是只一次触发多个后继节点,所以在 此选择等待被重置模式(Wait_to_be_reset_mode)在 trigger 被触发前,不 传递任何的消息给后继节点。修改完后点击右上角的完成即可保存属性配置。

编辑 trigger 节点			
删除		取消	完成
◎ 雇性			٠
◆名称	trigger		
显示名称	trigger		
触发模式	Wait_to_be_reset_mode		~
等待时间(ms)	1000		
第一条消息的来源	From_Payload		~
第二条消息的来源	From_Payload		~
是否延时刷新	False		~

双击第一栏的 fscript 节点,本节点作为 ble_server_service 节点的前 驱节点,所以具体内容即 ble_server_service 的输入组: payload(传入的 特征值数据), payloadLength(传入的特征值数据的长度)。修改完后点击 右上角的完成即可保存属性配置。

msg.payload = "hello";

set(msg.payloadLength, u16("5"));

双击第一栏的 ble_server_service 节点中的属性: BLE 服务端配置节 点旁的铅笔图标,属性: BLE 服务端配置节点在本流图中只需要配置一遍 即可,后续不再阐述。需注意的是支持的服务个数的属性值必须与本流图 中所添加的服务个数数量一致,其他属性参数可用 demo 流图默认值。修 改完后点击右上角的完成即可保存属性配置。

REF.	50H 95
0 ME112	
◆名称	ble_server
显示名称	
广播的最小时间	128
广播的最大时间	160
发现模式	2
广播类型	ADV_IND ~
不支持BR/EDR	1
使能广播	1
广播信道	0x07
广播的过速策略	0
Peer地址类型	0
Peerttat	
0,0,0,0,0,0	
SMP配对模式的参数	SECURE_CONNECTION_PASSKEY
CMDE0:08-700-8	
数	KEYBOARD_DISP
自己的地址类型	0
支持的服务个数	2

双击第一栏的 ble_server_service 节点,只需要确定需要添加的服务 UUID 和特征值 UUID,其他参数视具体情况而定,也可使用默认的参数值。 修改完后点击右上角的完成即可保存属性配置。

899		16291	18:11
• Ett			۰
● 5₩			
显示古作			
BLEND AND TO	ble_server	÷	
0x0F,0x18			
/* MIR11	False		~
保服性	True		~
不带回复的写题性	False		~
HEATH CET	True		~
網知識性	True		~
指示属性	True		
当台以证的写题性	False		~
17层展性	False		~
19EEMUUID			
0x19.0x2A			
गढ	True		~
न म	Тлие		~
读取认证。	False		~
读取授权。	Fatse		~
读取出的	False		~
该取认证文件	False		~
写认证	False		~
瑞織权	False		
Sum	False		~
写认证文件	False		~

双击第二栏的 fscript 节点,其内容与第一栏的 fscript 节点大体 一致,都是作为 ble_server_service 节点的前驱节点,所以具体内 容即 ble_server_service 的输入组: payload(传入的特征值数据), payloadLength(传入的特征值数据的长度)。修改完后点击右上角的完成即 可保存属性配置。 msg.payload = "server";

set(msg.payloadLength, u16("6"));

双击第二栏的 ble_server_service 节点,其属性与第一栏的 ble_ server_service 节点大体一致,需要添加的服务 UUID 和特征值 UUID 不同, 其他参数视具体情况而定,也可使用默认的参数值。

服务 UUID:

特征值 UUID:

最后双击 ble_server_adv 节点,自行设置蓝牙本地设备名称,以及需 要广播的每个小数据包的数据长度,类型,内容和扫描的每个小数据包的 数据长度,类型,内容。修改完后点击右上角的完成即可保存属性配置。

编辑 ble_server_adv	节点					
副林			取用		完成	
● 履性						•
◆召称						
显示名称						
BLE服务端配置节 点	ble_server			. ,		
蓝牙本地设备名称	BLE-ZWT					
需要广播的每个小						
数据包的数据长度						
0x02,0x03,						
需要广播的每个小 数据包的数据类型						
0x01,0x02,						
需要广播的每个小 数据包的数据内容						
0x06,0x0f,0x18,						
扫描响应的每个小 数据包的数据长度	t.					
0x02,						
日描桌应的每个小数据包的数据关型						
0x01,						
扫描响应的每个小数据包的数据内容						
0x06,						

ZLGi微文摘

4. 下载验证

下载流图后可在 BLE 调试宝点击蓝牙服务选项对流图提供的 service 进 行访问,点击每个 service 的左侧箭头,展开对各个 service 提供的特征进 行查看,还可对支持操作的特性进行读写。



每次操作都会有相应的日志信息,注意各服务 UUID 的联系,实时日志 如下所示。

09:33:31.537> 连接中
09:33:32.725> 连接成功, 等待发现服务
09:33:33.229> 连接成功,正在发现服务
09:33:33.783> 连接成功,并成功发现服务
09:33:33.841> 00002a05 Indication开启
09:33:33.841> [1111fffe] Notification开启
09:33:33.926> [1111fffe] Notification开启
09:33:35.683> [1111fffe] 读取成功: '73 65 72 76 65 72 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
09:33:36.511> [1111fffe] 读取成功: "73 65 72 76 65 72 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00"
09:33:36.799> characteristic读取失败
09:33:38.494> characteristic读取失败
09:33:43.483> [1111fffe] Notify: "12 34 56 00 65 72 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00"
09:33:43.542> [1111fffe] 成功写入: "12 34 56"
09:33:45.822> [00002a00] 读取成功: "75 6E 69 73 6F 63 5F 58
09:33:46.649> [00002a01] 速取成功: "03.01"
09:33:47 333> [00002a04] 連取成功: "00 00 00 00 00 00 00 00
09:33:48 091> [00002aa6] 連取成功: "00"
09:33:48.892> [00002ac9] 读取成功: "00"

BLE 调试宝向特征值 UUID 为 0x1111fffe 写入 12 34 56,作为服务端板 子的串口打印可显示收到的相应数据如下。

[09:	33:59	114]	枚←◆	•																				
char	acter	istic	data	write	e cb																			
noti	fy da	ta:																						
0×12	0x34	0x58	$0 \ge 00$	0x65	0x72	0x00	0x00	0x00	0x00	00x0	0x00	0x00	0x00	0x00	0x00	00x0	0x00	0x00	0x00	0x00	$0 \ge 00$	0x00	0x00	0x00
0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	$0 \ge 00$	0x00	0x00						
0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	00x0	00x0	00x0	0×00	0x00	0×0	:00 0:	:00 0:	:00 0;	x00 0.	:00 0:	:00 0;	:00
0x00	0x00	0x00	0×00	0×00	0x00	0x00	0x00	0×00	0x00	00x0	0x00	00x0	00x0	0x00	0×00	0x00	0×00	0x00	0x00	0×00	0×00	0×00	0x00	0x00
0x00	0x00	0x00	0×00	0×00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0×00	0x00	0x00
0x00	0x00	0x00	0×00	0x00	0x00	0x000	0x00	0x00	0x00	0x00 (0 0x00) 0x00	0x00	0x00	0 0x00	0x00 (0x0	0×0	0x0	0x00	0x0	0x0) 0x00	00x00
0x00	0x00	0x00	0×00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0x00	0×00	0x00	0x00	0x00	0x00	0×00	0×00	0x00	0x00
0x00	0x00	0x00	0×00																					
F09:	34:01	2111	₩ ← ●	0x0	0 0x00	0.0x00) 0x00	0x00	0::00	0x00	0 0x00	0x00	0x00	0x00	0x00	0x0								
1.0			n																					

【产品应用】 ZWS物联网云平台如何下发命令给设备?

原创 研发部 ZLG 致远电子 2023-05-04 11:35:24

对于一些位于偏远环境的设备,可以远程配置或控制设备显得尤为重要。ZWS 云从设计之初就充分考虑了这一需求,支持云端远程下发命 令给设备。

简介

ZWS 云平台是致远电子研发的通用物联网云平台,提供多种不同物联 网云服务,为设备厂商提供上云服务,支持对厂商不同种类设备的数据采 集和分析。

物联网的设备有千万种,不同的设备有着不同的控制命令,比如"开 关灯","时间同步","文件召回"等。ZWS 云针对这些不同命令做了 归并分类,可以分为三种:

1. 通用命令:所有接入 ZWS 云都需要支持的命令,例如文件召回,固件升级等。

2. 透传命令:命令下发过程不经平台校验,只做透传的命令。

3. 差异化命令:设备类型自己跟其他设备类型不同的命令。

对于这三种命令,虽然他们分类不一样,但他们下发的流程是差不多的, 都是先在云端定义好命令的格式,然后再下发给设备,设备收到命令后进 行命令的响应操作。具体流程如下图所示:



定义命令

定义命令是设备建模的一部分,可在功能块中定义命令字段。对于通 用命令和透传命令,ZWS 云平台已经在 common_fn 功能块进行了定义, 用户可以直接使用,不需要重复定义。这两种命令定义如下所示:



对于第三种差异化命令,这个是由用户自定义的,用户可以在自定义 设备类型的功能块下添加自己差异化的命令,如下图:

	新生物				NIEMENTE (1218 40)
÷ =1	î	2728		• antrane+	NT I BARR I DAIN
C inter	a - 1	DARK DROO	ANDS ANDS DADS	29688 MINOR	
REAL		0.0			
20185		1258/0282			
2 040	6 - A	Gevitz -			
CRO		erely -			
Sec.10	•	· netcecitiest			
th adde	8	OPENDE.			
712.81		194	23.135336		
NUR-L	a	- 68	113,271401		
1000	e .	- 04030188			
0 8923	e - 2				
A. 10-21	e				
	e -				
	· · · ·			10007100078048375 Room	. AN AND AND

说明: 自定义命令如果需要设备响应必须配置以下两个字段: cmdid: 命令 id,用于关联响应结果;

mtype: 消息类型,命令类型的消息此字段值固定是 2。

差异化命令,用户除了可以在功能块里面添加自己的命令实现命令下 发的需求,还可以复用通用功能块的 set_config 命令配置。用户只需要把 自己的命令参数放到 set_config 命令的 cfg_info 字段里进行下发,设备收 到该命令后进行响应即可。

set_config 命令不仅可以给用户自定义命令复用,而且,可以通过 set_config 命令配置设备的状态值。

A NT	GREE GROO GREE INTER TREES 27412 SECON	AR SAME CO
NAME .		
1000	* 1.9.8/17/8/P	
- 2412	dars webster	
28.94	M'MP . Enclored	
2194	- 18028GP540	
231220	GPSRJad	
1288	PMR 23.135306	
8357.8	42章 113-271-03	
ROBELT	- 10/H420H468	
19561		
/0/1818		
MILLION .		

设备接收命令

设备固件开发需要集成"ZWS_SDK",SDK 里面提供了设备连接云服 务关于网络层的封装同时也提供了一些 demo,其中 device_example 这个 demo 演示设备怎么连接云,然后设备接收处理云端命令。用户可以直接通 过 make 指令编译 SDK 里的 device_expl.c,编译成功后,把设备的三元组 传给运行程序,把 demo 跑起来。

linux 系统,在 SDK 所在目录执行以下命令:

./bin/device_example demo_dev_type d123 123123123 代码说明:

1. demo 中关于命令的回调是在 device_expl.c 文件中的 on_message 函数。

2. demo 中只实现部分通用命令的响应处理,自定义命令需要用户自己 实现。

ZLGi微文摘

下发命令

在设备管理的高阶功能界面,通过 set_config 命令下发给设备,设备 接收命令,响应成功后返回给 ZWS 云平台,如下:



设备收到命令的处理日志:





智能网联终端 CATCOM Cat.1

⊕ 点击购买

【产品应用】 AWorksLP例程介绍(MR6450)-SD卡

原创 研发部 ZLG 致远电子 2023-05-05 11:31:04

AWorksLP 对存储类设备进行了高度抽象化,为存储类设备提供了通用的文件操作接口,应用程序可以轻松跨平台。本文以 MR6450 平台为例,介绍 AWorksLP 基于 FatFs 的 SD 卡的基本用法。

简介

SD 卡(Secure Digital Card)即安全数字卡。是一种基于半导体快闪 记忆器的新一代记忆设备,从 MMC 的基础上发展而来。由于它的体积小、 数据传输速度快、可热拔插等优点, 被广泛的运用于便携式和嵌入式设备上。

FatFs 是一种面向小型嵌入式系统的通用的 FAT 文件系统。它完全是由 ANSIC 语言编写并且完全独立于底层 I/O。因此它可以很容易的移植到不同 且资源有限的微控制器中。

由此可见在嵌入式开发中对 SD 卡中的文件进行管理时,FatFs 是使用的最多的文件系统。

在 AWorksLP 中已经集成了 FatFs 文件系统,并支持使用该文件系统对 SD 卡的文件进行管理。

相关API

在 SD 卡例程中,是通过文件接口实现对 SD 卡中文件进行操作的。在本文中仅介绍用到的文件接口,其他文件接口,请参考 SDK 中《AWorksLP OS 标准 API 参考手册 (html)》。

AWorksLP 函数列表:

函数原型	简要描述	备注
Int aw_open (const char *path, aw_oflag_t oflag, aw_mode_t	aw_open 函数可用于打开	也可使用 posix 标准
mode);	SD卡设备和文件	接口 open 函数
Int aw_close (int fd);	aw_close 函数可用于关闭	也可使用 posix 标准
	SD卡设备和文件	接口 close 函数
<pre>aw_err_t aw_write (int fd, const void *buf, size_t nbyte);</pre>	aw_write 可向打开后的文	也可使用 posix 标准
	件中写入数据	接口 write 函数
<pre>aw_err_t aw_read (int fd, void *buf, size_t nbyte);</pre>	aw_read 可从打开后的文	也可使用 posix 标准
	件中读出数据	接口 read 函数
<pre>aw_err_t aw_mount (const char *p_mnt, const char *p_dp_name,</pre>	用于挂载 sd 卡到指定的文	也可使用 posix 标准
<pre>const char *p_fs_name, uint32_t flags, void *p_mnt_par);</pre>	件夹中	接口 mount 函数
<pre>aw_err_t aw_make_fs (const char *p_dp_name, const char</pre>	用于将存储设备格式化为	
<pre>*p_fs_name, const struct aw_fs_format_arg *p_fmt_arg);</pre>	指定的文件系统格式	
<pre>int aw_mkdir (const char *path, aw_mode_t mode);</pre>	aw_mkdir 可用于创建文件	也可使用 posix 标准
	夹	接口 mkdir 函数

部分接口参数属性表:

参数属性	定义	描述
	AW_O_RDONLY	只读
	AW_O_WRONLY	只写
	AW_O_RDWR	可读写
	AW_O_APPEND	追加打开
文件标志 (any offag toffag)	AW_O_CREAT	如果文件不存在,则创建,如果存在,则继续
文田小志 (aw_onag_tonag/		配合 AW_O_CREAT 使用,单独使用没效果,如果文
	AW_O_EXCL	件存在,则失败
	AW_O_TRUNC	截断
	AW_O_DIRECTORY	目标应该是个目录,如果不是则返回失败
	AW_O_NONBLOCK	非阻塞模式打开,某些外设也许会忽略这个设定
	AW_S_IRWXU	文件所有者具有读、写、可执行权限
	AW_S_IRUSR	文件所有者具有读权限
	AW_S_IWUSR	文件所有者具有写权限
	AW_S_IXUSR	文件所有者具有可执行权限
	AW_S_IRWXG	用户组具有读、写、可执行权限
文件模式 (au mada tmada)	AW_S_IRGRP	用户组具有读权限
CTF1REL (aw_mode_tmode)	AW_S_IWGRP	用户组具有写权限
	AW_S_IXGRP	用户组具有可执行权限
	AW_S_IRWXO	其他用户具有读、写、可执行权限
	AW_S_IROTH	其他用户具有读权限
	AW_S_IWOTH	其他用户具有写权限
	AW_S_IXOTH	其他用户具有可执行权限

工程编译

环境的搭建、Eclipse 工程的编译与配置、开发板的仿真与调试、请参考《AWorksLP SDK 快速入门(MR6450)——开箱体验》。

例程介绍

本文介绍例程在 {SDK}\demos\peripheral\sdcard 目录下。

在 src 目录下的 demo_sdcard_fs.c 例程代码中的第 149 行到第 157 行 使用了 while 循环如代码 1 示,在循环中每隔 500ms 打开一次 SD 卡设备, 通过检查返回值来检测是否有 SD 卡插入。如检测到返回值大于等于 0 则表 明有 SD 卡插入。需要注意的是打开 SD 卡的设备名("__BLK_NAME"在 demo_sdcard_fs.c 的第 67 行中默认被宏定义为"/dev/sdcardB0")需要 与实际检测到的 SD 卡设备名一致。在 AWorksLP 中 SD 卡默认是动态监测, 在 SD 卡插入时会在串口打印设备信息如图 1 示,串口显示 SD 卡设备名为 "sdcardA0",所以我们需要修改"__BLK_NAME"的宏定义为"/dev/ sdcardA0"。

64 /* sd 卡设备分区名,是由 sd 卡设备分区注册为字符型设备 65 * sd 卡设备分区名一般是由 SD 卡设备名加上 part number 66 */ 67 #define __BLK_NAME "/dev/sdcardB0" 149 do{ 150 fd = aw_open(__BLK_NAME,AW_O_RDWR,0); 151 if (fd < 0) { 152 aw_kprintf("open device failed\r\n"); 153 aw_mdelay(500); 154 } 155 }while(fd < 0); 156

150

COM3 - Tera Term VI

157 aw_close(fd); 代码1 SD卡设备检测流程

文件(F) 编辑(E) 设置(S) 控制(O) 窗□(W) 帮助(H)	
open device failed	^
open device failed	
open device tailed	
open device failed	
Insert an sdio device,start to identify	
Found a new memory card:	
Product name: SD128	
Capacity: 118848 MB	
Manufacturing date: 2022-08	
dev_name:sdcardA, valid part_no:0	
detect dynamic insert success !	~

图1串口打印SD卡设备名

代码 2 中第 159 行到 172 行的功能为使用 aw_make_fs 函数接口将 SD

ZLGi微文摘

```
卡格式化为 FatFs 文件系统。第159 行的宏定义默认为 #if 0,本次例程为
了演示该步骤将宏改为 #if 1。需要注意的是格式化会把卡内的数据清除,
在执行格式化之前需要确保卡内没有重要数据。
```

159 #if 1 /* 格式化一次即可 */

```
160
```

- 161 /* 卷名为 "awdisk",卷大小为 4k */
- 162 struct aw_fs_format_arg fmt = {"awdisk", 1024 * 4, 0};
- 163
- 164 /* 制作文件系统 ,将存储器名为 "/dev/sd0" 制作为 "vfat" 类型

的文件系统 */

- 165 ret = aw_make_fs(__BLK_NAME, "vfat", &fmt);
- 166 if (ret != AW_OK) {
- 167 AW_ERRF(("failed: %d\n", ret));
- 168 return;
- 169 }
- 170 AW_INFOF(("make fs OK\n"));
- 171

```
172 #endif
```

代码2 SD卡格式化

格式化完成后,代码 3 中第 174 行通过 aw_mkdir 函数创建一个名为 "/ sd"的目录,以该目录为 SD 卡的挂载点。创建目录的文件模式为文件所有者、 用户组和其他用户都可以对该目录进行读写。第 181 行通过 aw_mount 函 数将 SD 卡挂载在"/sd"目录中。需要注意的挂载前需要确保目录存在, 如果将 SD 卡挂载在不存在的目录上,挂载会失败。挂载的文件系统也需要 与 SD 卡格式化的文件系统格式一致,否则挂载同样会失败。

173 /* 创建挂载节点 */

174 ret = aw_mkdir("/sd", AW_S_IRWXU | AW_S_IRWXG | AW_S_ IRWXO);

- 175 if (ret != AW_OK) {
- 176 AW_ERRF(("/sd create error: %d!\n", ret));177 return:
- 178}
- 179

180 /* 文件系统挂载到 "/sd" 节点 */

181 ret = aw_mount("/sd", __BLK_NAME, "vfat", 0, NULL);

182 if (ret != AW_OK) {

183 AW_ERRF(("/sd mount FATFS error: %d!\n", ret));

- 184 return;
- 185}

186 AW_INFOF(("mount OK\n"));

```
187
```

188 /* SD 卡读写测试 */

```
189 __fs_file_rw();
```

代码3 挂载SD卡

最终例程会在第 189 行调用 __fs_file_rw 函数,函数体如代码 4 所示。 在该函数中会在"/sd"目录下创建一个名为"aworks_sd_test.txt"的测试 文件,通过对该文件进行读写测试,读写完成后对数据进行效验,效验通 过则表明 SD 卡读写测试成功。至此 SD 卡例程执行完毕,整个过程会有串 口信息打印如图 2 所示。

```
69 aw_local void __fs_file_rw (void)
70 {
```

71 int i = 0: 72 int handle; 73 char *p_file_name = "/sd/aworks_sd_test.txt"; 74 uint8_t str_buf[256] = {0}; 75 int len: 76 77 /* 78 * 写文件测试(包括创建,文件写操作,关闭操作) 79 */ 80 /* 创建新文件 */ 81 handle = aw_open(p_file_name, AW_O_RDWR | AW_O_CREAT, 0777); 82 if (handle < 0) { AW_ERRF(("creat file error: %d\n", handle)); 83 84 return[.] 85 } 86 AW_INFOF(("creat file %s ok\n", p_file_name)); 87 88 len = sizeof(str_buf); 89 for (i = 0; i < len; i++) { 90 str_buf[i] = (uint8_t)i; 91 } 92 93 /* 写文件 */ 94 if (aw_write(handle, str_buf, sizeof(str_buf)) != sizeof(str_buf)) { 95 aw_close(handle); 96 AW_ERRF(("write file errorn")); 97 return: 98 } 99 AW_INFOF(("write file %s ok\n", p_file_name)); 100 101 /* 关闭文件 */ 102 aw_close(handle); 103 AW_INFOF(("close file %s ok\n", p_file_name)); 104 105 /* 106 *读文件测试(包括打开,文件读操作,关闭操作) 107 */ 108 /* 打开文件 */ 109 handle = aw_open(p_file_name, AW_O_RDONLY, 0777); 110 if (handle < 0) { AW_ERRF(("open file error: %d\n", handle)); 111 112 return: 113 } 114 AW_INFOF(("open file %s ok\n", p_file_name)); 115 116 memset(str_buf, 0, sizeof(str_buf)); 117 118 /* 读取文件 */ 119 len = sizeof(str_buf); 120 if (aw_read(handle, str_buf, sizeof(str_buf)) != sizeof(str_buf)) { 121 aw_close(handle);

- 122 AW_ERRF(("read file error!\n"));
- 123 return;

```
ZLG微文摘
```

```
124 }
125 AW_INFOF(("read file %s ok\n", p_file_name));
126
127 /* 检验数据是否正确*/
128 for (i = 0; i < len; i++) {
129
     if ((uint8 t)i != str buf[i]) {
        AW_ERRF(("file data error!\n"));
130
131
        aw_close(handle);
132
         return;
133
      }
134 }
135
136 AW_INFOF(("file %s data check ok\n", p_file_name));
137
               代码4 __fs_file_rw读写测试函数
```



图2 SD卡例程执行成功的串口信息

扩展介绍

在上节中有提到 AWorksLP SD 卡默认是动态检测设备,即支持热拔插, 这是通过一个检测引脚实现的。在实际应用中,可能出于节约 I/O 资源的考 虑,需要将检测引脚复用为其他功能。在这种情况下可以将 SD 卡定义为静 态设备,这样检测引脚就能复用作其他功能了。

在 AWorksLP 中 SD 卡的动态设备在图形化配置界面中没有 SD 卡设备 可供使用,只需将对应的 SDIO 控制器选上即可。上文例程中 SD 卡是通过 sdio1 设备来控制的,所以仅需保证 sdio1 设备使能即可,如图 3 所示。



图3 动态设备下确认控制器被选上 将 SD 卡设备以静态方式注册时,则需要进行如下操作: 1. 将 pins.dts({board}\dts\pins.dts) 文件中 CD 引脚配置注释,修改完 成后如代码 5 所示;

2. 在 board.dts 文件中添加 SD 卡设备,如代码 5 所示。

/* cd-pins = <&pin1 PIN_PD28

(IOC_PD28_FUNC_CTL_SDC1_CDN|HPM_PIN_DS(6)|HPM_PIN_PE(_ HPM_PIN_PE_ON)|AW_PIN_CFG_PULL_UP)

(IOC_PD28_FUNC_CTL_SDC1_CDN|HPM_PIN_DS(6)|HPM_PIN_PE(_ HPM_PIN_PE_ON)|AW_PIN_CFG_PULL_UP)

(IOC_PD28_FUNC_CTL_SDC1_CDN|HPM_PIN_DS(6)|HPM_PIN_PE(_

HPM_PIN_PE_ON)|AW_PIN_CFG_PULL_UP) >; */

代码5 注释引脚

&sdio1{

sdio_mem_card0:sdio_mem_card0 {
 compatible = "general,sdio_mem_card";
 label = "sdcardA";
 status = "disabled";
 };
};
代码6 添加SD卡设备

完成修改后,重新打开例程编译图形化配置界面,即可在Board EPC6450-AWI/Devices/External Memories 选项下看到板卡下新增 sdcardA 设备如图 4 所示,使能该项并保存退出,再次编译工程并执行固件,例程 现象与上节中所描述完全一致。

Aworks System Configuration	
Save Search	
(top) -> social trx-city-ann -> uncersis -> External Memorys (z) #	<pre>Symbol: DEV_SDCARDA Filename: "/dev/sdcardA" dts_dev_prop: name(unit): "sdio_mem_card" (0) compatible: ['general_sdio_mem_card'] dts_path /soc/sdio@st2034000/sdio_mem_cardd hinding: D:\aworks-team\temp\aworks_lp_hpm645</pre>
HPM Platform	v C

图4 选择SD卡设备

注意: 例程在静态 SD 卡设备下执行需要在固件运行前将 SD 卡插入。

【产品应用】 基于CAN智慧云实现远程ECU升级

原创 研发部 ZLG 致远电子 2023-05-09 11:34:01

在日常研发中,调试已经成为日常工作中核心的一部分。汽车电子行业中,测试时,常需对 ECU 固件进行刷写。CAN 智慧云基于 CANDTU 系列产品实现远程刷写 ECU 固件,减少人力与时间成本。

简介

在汽车路试的时候,因路试的特殊性,汽车需要长期在外进行路试。 测试人员不仅同时兼顾几个路试车辆,而且测试场所距离远与位置不固定, 测试人员需到测试现场对 ECU 进行 ECU 固件刷写,从而导致人力和时间 成本增加,测试效率也不高。为解决该应用痛点,CAN 智慧云推出了远程 ECU 升级。

实际操作

ECU 升级对 CANDTU 系列设备有一定的要求,需要 SD 卡的支持,如 "CANDTU-200UWGR、CANFDDTU-400EWGR"等系列设备。

云端 ECU 升级总体流程如下:



1. 使用ZCANPRO编写刷写包

1.1 使用致远电子的 ZCANPRO 的软件,在 "ECU 刷新"模块,自行编 写刷新的流程与参数。ECU 刷新需要经过多个流程,一个服务为一个流程, 不同 ECU 刷新流程有所不同。

编辑器 - sample.zfash* 获得器	
(#142) (F777) (#1777) (#2171) (E2171)	CANERE: CAN • 1512: 700 748 701 8554938 85549488
Diage#264294 Diage#26429 Diage#264294 Diage#26429 Diage#26429 Diage#26429 Diage#26429 Dia	L99 Trgs Bible Ref Bible 1 R.J.Y.WERK 10.0 2 2 USERANCEALSH 10.0 2 3 R.J.Y.WERK 10.0 2 4 MARTING 10.0 2 5 S.O. 10.1 5 3 4 R.J.WERKEG 10.02 4 5 Z.Y.WERL 2 COCOLE 8 Z.Y.COLE 10.01 7 7 DOLE 10.01 2 8 VCOCRE 10.01 2

1.2 对于已完成编写的流程,可点击【导出】,下载至本地保存。或点击【应 用到执行器】,导出云端刷写包。

编编器 - sample.zfash* 终行器			
848 FTH 6478 SH CHHICH8	CANB	IZI : CAN 💌 ISI	2: 700 7df 701 (R.F.G.B. M.S.400)
000499403046 00050028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 01430028 0140028 0140028 0140028 01400028 0140028 0140028 0140028 0140028 0140028 0140028 0140028 0140028 0140000 014000000000000000000000000000	第日 (CURE) 約日 第月10日。 第月10日。 第月10日の 11 7月1日、 (2世紀二〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇	上日 7.6 影響 第.5 第.6 1 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5	2 (moder) 1990 2007 2017 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 2

2. 导出远程刷写包

点击右侧的【执行器】,对于云端的刷写包,需经协议特殊处理,点击【导 出远程刷写包】,即可下载.zrflash 格式刷写包。

for faces and a self-code				
for farmers to self-rate				
sg/sampie.znasn	打开文件 想让:	700 7d	701	RF (1)
用写次数: 1 🕴 间隔时间: Os	并始用写			/
时间 尚型 NID 方向 长度	数据/描述			
	Reveal 1 0 000010 0 Rev 22 1 0 000010 0 Rev 24 1 0 0000 0 0 Rev 24 1 0 000 0 0	RP30.01 D19750 R0 D19750 D19750 <td>「使いかい時からない」であった。 「使いい時からからない」であった。 「使いい時からからた」 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」 「しいい」であった。 「しいい」 「」であった。 「しいい」 「しいい」 「しいい」 「」であった。 「しいい」 「しいい」 「しいい」 「しいい」 「」であった。 「しいい」 「しいい」 「」でのか。 「」であった。 「」でのか。 「」でのか。 「」でののでのでの。 「」でののでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでので</td> <td>goyampid-table goyampid-table goya</td>	「使いかい時からない」であった。 「使いい時からからない」であった。 「使いい時からからた」 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「使いい時からない」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「たいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「」であった。 「しいい」であった。 「しいい」であった。 「しいい」であった。 「しいい」 「しいい」であった。 「しいい」 「」であった。 「しいい」 「しいい」 「しいい」 「」であった。 「しいい」 「しいい」 「しいい」 「しいい」 「」であった。 「しいい」 「しいい」 「」でのか。 「」であった。 「」でのか。 「」でのか。 「」でののでのでの。 「」でののでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでので	goyampid-table goya

3. 上传刷写包至云端

3.1 点击顶部栏【ECU 管理】,点击【+ 添加 ECU】,对 ECU 进行添加。 对于部分 ECU 升级需安全验证,可自行添加解锁文件。

	MARCHAEL (Do'	Contract of	Therease a second	and the second s	1917
ECU_01	700		添加ECU	×	840
dvedvo	123412913256-	1 COLUMN	Pour as		414 224
1 2 AZR	1095页 -	- ECOMM	ECO TA	100	
		物理地址(0×)	731	350	
		#6位15社(0X)	701	350	
		2582153让(0x)	7df	348	
		42M2	15/84t	9	
		故障码文件	003/07/07/02/219	Sia	
		ECURN+10	Inflash	376	
		解输文件	5k.50	2016	

3.2 对于解锁文件,由于远程解锁操作的动态库在 Linux 64 位运行,故 需将原解锁文件 (dll 文件)编译为 linux 下的动态库(so 文件)。仅需将 Windows 下编译 dll,在 Linux 下编译为 so 文件即可。Linux编译命令为: `g++ ecu_key.cc -fPIC -shared -o libecu_unlock.so` 其中 ecu_key.cc 文件为解锁函数的实现,生成解锁文件动态库为 libecu_unlock.so。

4. ECU升级

4.1 进入到相应的设备,选定 ECU 升级,选择相应的通道与 ECU。选 定了相应 ECU,会默认配置地址、帧、刷写包等,同时也支持自行更改。

CANGER CAND		Manteso	38.51	252L(0x)	1002010-0-	0.0101	Transal	0.844	
Ht文作		ECU_02							
FIMECL084	10105	ECU_01							
RICH-R		dwedwe							
2014 1800									

4.2 确定参数信息,点击【升级 ECU 固件】,即可查看相应的调试信息, 查看升级反馈。

CANORIE	CAND V	ECU ECU 02	16.91	1522-000	731	701	701	1° EM
					10781014	49157512	15824844	
拼文件								
	Printecultur	Berrische Official Berlische			- 10			
muccam.								
3250	1817 1							

4.3 可查看调试信息中的 ECU 的升级日志信息,日志反馈"ECU 刷写完成",即代表 ECU 升级成功。

NUMBER 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 <th>200 2010 0 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 1 82A 2010 0 1 2 52 2010 0 1 2 52 2010 0 1 5 55 2010 0 1 5 55</th> <th>2712-0.0888 - 31編載記 - 5000ms - 84回 - 154回 - 154 - 1</th> <th>物語: 0x30000 f 利品度: 0x1500 和品での1500 和品での1500 和品での1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1</th> <th>6貫: 4788 55 58: 3136 58: 241 68: 241 405 405</th> <th></th>	200 2010 0 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 2 1 82A 2010 0 1 82A 2010 0 1 2 52 2010 0 1 2 52 2010 0 1 5 55 2010 0 1 5 55	2712-0.0888 - 31編載記 - 5000ms - 84回 - 154回 - 154 - 1	物語: 0x30000 f 利品度: 0x1500 和品での1500 和品での1500 和品での1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1500 年間の1	6貫: 4788 55 58: 3136 58: 241 68: 241 405 405														
				ž	主教	3 F	通i り,	道C 点:	CAN 击贝	NFI 勾핏	D-	bu	s数	塘	记	录	لا	

技术平台▼

【产品应用】CATCOM-100搭配 ZWS物联网云平台,实现Modbus设备上云

原创 研发部 ZLG 致远电子 2023-05-12 11:50:05

Modbus 协议是应用于控制器上的一种通用标准,常被用于本地集中 监控。在平台数字化需求推动下,如何能快速实现设备上云?本文将 介绍通过 CATCOM-100 和 ZWS 物联网云平台实现 Modbus 设备上云。

应用场景

建筑楼宇、工业控制、环保等领域,广泛的应用 Modbus 设备进行本 地控制和管理。但随着企业的发展,出现了跨地区、跨领域甚至跨国的业 务项目,需要将不同地区的设备数据采集上云,并进行远程监控、远程维 护等,同时还需要设备信息展示、监控大屏展示等功能。

这些 Modbus 设备要怎样才能快速接入物联网云平台?

CATCOM-100 + ZWS云平台应用方案

ZWS物联网云平台,提供设备上云服务,满足设备远程控制、实时监测、 故障告警等应用需求,助力企业完成数字化平台升级。

Modbus 设备上云,可以将 CATCOM-100 和设备进行串口连接, CATCOM-100 将设备数据采集到 ZWS 云端,在云端进行设备数据监控,远 程维护等工作。



通过 CATCOM-100 将 Modbus 设备接入云端,可以将 DTU 配置为网关 模式,此时 CATCOM-100 作为网关,Modbus 设备作为子设备,网关模式 下的整体通信流程如下:





云端接入设备和查看数据

1. 在云端添加设备

首先,在物联网平台中,添加 CATCOM-100 类型的 DTU 设备。

	-	ANER	
8	ONER	- 104-102	CATCOM HID
	0842	10809	HECTURE IN
	DECENT.	1080	7520050001 2010
		0.0001	and contained to be
		0.8162	and ARTINE MIL
	0898		A David VI Trees
	8048		and the second
18	1011		Catin Our
۲	00017		
1,	8-23		Direct O Tall Bans

然后,创建 Modbus 设备类型,并添加设备。

						2. millimodeus	25	
-	A	1. 198228	an and an a		4 AT	-		
101 0	***	-	siac		B seen -	10eets	WZ_status	
		- 100.00	12,000	-	0442	12850	ectualda	3(12
1 10	Care.		WE realist		LADATE	1.080	#1271621	1/17
		canti			E	. 59444	teettime	1094
		80		1.00	ORME	STATE.	and-lower	
			1.1	_	0808	intel	and dimension	

最后,关联 DTU 网关和 Modbus 子设备。

4.45	1907056													65.	设备列表	0.0115
C dates	128402	12862	DTUNET	0804	0800	67/8/8	7195	1482	(P08210)							
0992	+ 16.00	F@8														
2409808	84		1865			1455			0.00	2.001	MB	1.664(18)	A10114		4.7	
生活管理			Blivebon			w2_rodba			cd12x852	bru225666c	0				62.10	a
	181	1 (k) (k)	16 1060													
GRAM																

云端成功添加设备后,需要配置 DTU 的数据传输模式为网关模式。同时, 将子设备的设备 ID、设备类型、设备密钥配置到 DTU 的串口信息中。

2. 设备数据解析配置

配置数据解析脚本。

	-	功能块	新新创造 属性起居 阿本雷语
8	设备建模		
	18 89 92	1219 10	6
	功総決管律	解析方式 〇 即本	医念规则
	实体管理	通过算事对ra	www.据进行解析
	设备管理	脚本配置	
	设备列表	*脚本名称	设备解告脚本
	设备分组	* 與聖	Growy
	(8)件管理		
Lb.	政府管理	*内容	import com zigcloud.ictplatform iotmapping.commons.GroovyUtils Import java.nio.ByteBuffer;
Ξ	告發管理		788
1,	用户管理		* 将二进制限之始新自然的和型的maploga
Ð	离阶功能		* @param rawData 二进制税文 * @return 物理型题题
6	其他应务		* woeschpeion /ssatesoistsfrawivataio/son , ###xxxiedaloyre[]#fga
~			static Map < String, Object > rawDataTolson(byte]] rawData) (

3. 设备数据查看

设备端上报数据,在云端就能查看设备数据了。



【产品应用】 如何利用IoT云平台统计设备数据? — 基础篇

原创 研发部 ZLG 致远电子 2023-05-18 11:35:21

通过查询设备历史数据,可以看到数据一段时间的变化,如果想将设备数据进行进一步处理和聚合,数据统计必不可少。本文将介绍如何使用 ZWS 云平台的数据统计功能,以及内置算法的实例操作。

应用场景

设备上报到 ZWS 云平台的数据经过解析后可以直观地看到某一时刻设 备的数据状态,而想观察设备数据整体的特征或者分析某一个时间段数据, 就得利用统计功能对数据进行处理:



统计数据准备

首先,需要在 ZWS 云平台中创建设备并上报设备数据:

ine an	i.	· @ 2023-01-	16 18:01:34 🕱 2023-02-	5 18 01 34 388	THE				
98	COLL	REED	RANDE	ProMetri	805465	M/199732-0540	正式登録が天ちら早	122304/3	101
1	12.86	52.99	98.19	5.09	37.55	35.52	94.97	75.0	2023-02-15 18:01:1
2	92	35.72	6.94	66.86	50.24	29.65	10.70	61.7	2023-02-15 18:00:1
3	67.94	97,01	22.44	85.89	49.5	18.1	60.99	88.3	2023-02-15 17:59:1
4	40.94	34.69	28.66	76.39	58.07	33.04	56.18	23.5	2023-02-15 17:58:3
5	56.62	31.51	55.52	10.63	2.87	42.97	69.1	92.0	2023 02 15 17:57:1
6	82.25	31.42	81.72	47.75	22.08	\$2.63	33.3	27.4	2023-02-15 17:56:1
7	68.4	32.5	87.51	75.5	39.15	10.89	10.9	4.7	2023-02-15 17:55.1
8	47.93	92.5	1.04	40.72	35.89	1.91	89.23	28.5	2023-02-15 17:54-1
9	3.19	22.68	59.02	47.35	64.37	38.56	96.66	42.	2023-02-15 17:53:1
10	55.08	91.28	33.45	73.87	\$4.81	11.94	13.12	14.5	2023-02-15 17:52:1

统计算法

ZWS 云平台的统计算法有两种,分别是内置统计算法和自定义统计算法: 1. 内置算法是 ZWS 云平台定义好的算法案例,可以直接关联设备进行

统计,无需关心统计算法内容和过程;

 自定义算法可以自己拟定数据统计的方式(例如取某段时间的最大最 小值,总和等等)来统计设备数据。

使用内置统计算法

ZWS 云平台内置了三个类型的算法案例:

- 统计设备数据的某个数值在一段时间内的占比;
- 统计分组设备在一段时间的总和;
- 设备的数据累加。

RE SERVICE				NG DOBINIT
+1	AL PEA			
1945	Watt	10.4	20	10.9
1	设备和质量的 点比	國民主導動整個的第一個運動一級同共同的政治的	n#	28 m
2	力地设施的奴隶家公	他们立地获得他一把拉利的方面和	n#	11R 200
1	0-entremail	63467301921921	10	nn
	1 21.8 108/01 -			

在一个算法中,可以创建两种类型的实例:

- 立即统计实例:可以立即统计某个历史时间段的数据,比如:立即 统计前一天的电量总和;
- 定时统计实例:可以按照一定时间间隔对设备数据执行统计,也可 以按某个时刻来定时执行统计。

1. 创建实时统计实例

我们以数据累加算法为例,点击添加实例按钮,这里我们创建一个立 即统计实例,选择设备、选择统计的字段以及统计的时间范围:

NT/RB				
area.				
11.00 242,6085	p.	WANTE GROWWICH		
8796 88		allet en		
1001R				
	建加采用			
10 Million III	8.995.92	6.0040	8/78/0	80
1285951+		wing.		

可以在统计结果页面查看数据累加统计后的结果。

149			ROL BORDET WEATH
18.7012-01			
RH64F DERITEN	6/50		
请求学校			
#85	10.6	95/2	
info_model_id	10 MERIO	mon	
third_things_id	三方印象の	839:28 -	
property	職性の作	ana@	
统计设置			
HE1181RER 0 2023-02-14 00 0	0.00 至 2023-02-16.00.00.00		
_			
2011 462			

2. 创建定时统计实例

创建一个定时统计实例,选择设备、选择统计的字段以及定时设置, 统计周期以10分钟统计一次为例:

NUM NUM NUM EN Statute Name NUM IN NUM NUM	10.2.7%			NG · RRANT · PULCE
DR. ERG/F28 300145-800800 El 3950-6480900 RE 46 ALCER ALCER ALCER 1 Charl-GRAB0000077 BL/ME 2224-6110233 1 Charl-GRAB0000077 BL/ME 2224-6110233 1 Charl-GRAB0000077 BL/ME 2224-6110233	中古在市 的计结束			
All Ref (R) Ref (R) 1 Charled 00000000000 Ref (R) 202024 (S) 2020	CH DENHERM	······································		
1 (http://02186006999999) #U (221-02-1519275)	40	9611+c2.85	9414-90-0	16/19/94
2 Photol ATTA & SWYCOALSUT	1	("total".4509.860006093979)	统计规定	2023-02-15 19:27:53
	2	("bolal":4474.450006246567)	0411 7638	2023-02-15 19-26-49

在实例列表中点击启用,则可开启定时统计。

BACE Introduction	an ever be
Image:	
ANI ANI Para Marching State State	
EEE EEE 68 0 69 0 60 0 60 0 60 0 60 0 60 0 60 0 61 0 62 0 63 0 64 0 65 0 66 0 67 0	
Pr () 2010 - 424 () RE () 2010 - 414 ()	
MAX - 2000-100 MAX - 2000-100 MAX - 2000-100	
Extend of 10 mm	
1987	

可以在统计结果页面查看定时统计后的结果。

20768				
129709(12	- BLOOM			
80	±910.0	211922	2.010	80
1	104105114012	("into_model_ial".91,"thind_things_ial"."1234567899","propert	日純用	1170 1278 NOR 207 15.00
100 000 000	11 1 M 11 M 11			

除了内置的算法案例外,ZWS 云平台还支持自定义统计算法,开发者 可以自行编写算法脚本来实现数据统计。后续篇章会在上述内容的基础上, 讲述自定义统计算法的使用方法。

IN:LISESS			MALE - MARINE - 300
113.02.0 Ma+0.2			
CH EPHILING	- 0 2023-02-14 00:00:00 X 2023-02-44 00:00:00		
-	NUMBER	Mittel	WORK
1	["total":4162.580010175705]	病计用度	2021-02-15 19:20:00
1 2	[*tetal*/4142.580010175705] (*total*:1664.2900054454603)	编计和增 统计和增	2023-02-15 19:20:00

【产品应用】 如何利用IoT云平台统计设备数据? — 进阶篇

原创 研发部 ZLG 致远电子 2023-05-23 11:37:16

设备数据上云,解析后的设备数据一般是时序存储,但纯粹的设备时 序数据无法给用户带来更大的业务价值,需要根据业务需求进行额外 的数据统计分析。本文将介绍如何利用 ZWS 云平台的自定义统计算法 对数据进行统计。

应用场景

上一篇《如何利用 IoT 云平台统计设备数据? — 基础篇》介绍了如何 运用 ZWS 云平台中的内置统计算法进行数据统计,这次将介绍如何创建自 定义统计算法。

为什么需要自定义统计算法?由于业务需求各式各样,有的需要求平均值,有的需要求累加值,不同维度需要不同统计算法,开发者可以通过 自定义统计算法来满足对应的项目需求。

统计数据准备

首先,需要在 ZWS 云平台中创建设备并上报设备数据:

THE OF		~ 0 2023-01-	15 18 01 34 X 2023-02-	15 18.01.34	7.82				
· 101	enet	RECTO	ROBER	1108015	8058/5	MINR2528	ROBANION	acentri	100
1	12.86	52.99	56.79	5.09	37.35	36.52	54.97	155	2023-02-15 18:01:1
z	92	35.72	6.94	66.85	50.24	29.65	10.78	61.7	2023-02-15 10.00
3	67.94	97.01	22.44	85.89	49.5	18.1	60.99	68.3	2023-02-15 17:59
4	40.94	34.09	20.01	76.39	58.07	33.04	56.18	23.0	2023-02-15 17:58
5	56.62	31.51	55.52	10.63	2.87	42.97	69.1	3.50	2023-02-15 17:57
6	82.25	31.42	81.72	47.75	22.08	52.63	33.3	27.4	2023-02-15 17:56
7	68.4	32.5	87.31	73.5	39.15	10.89	10.9	4.7	2023-02-15 17:55
8	47,93	92.5	1.04	40.72	30.89	1.93	89.23	28.5	2023-02-15 17:54
9	3.19	22.68	59.02	47.35	64.57	38.56	96.95	42.1	2023-02-15 17:53
10	55.08	91.28	33.45	73.87	84.81	11.94	13.32	143	2023-02-15 17:52

统计算法

ZWS 云平台的统计算法有两种,分别是内置统计算法和自定义统计算法。

1. 内置算法为云平台定义好的算法,可以直接关联设备进行统计,无 需关心统计算法内容和过程。

 2. 自定义算法可以自己拟定数据统计的方式(例如取某段时间的最大最 小值、总和等等)来统计设备数据。

创建自定义统计算法

不同于内置算法,自定义算法从请求参数、统计过程、统计结果均由 用户自定义,需要开发者编写脚本。目前自定义统计中支持了 Groovy 和 JavaScript 两种脚本语言解析。



1. 请求参数

- 类别:云平台提供给自定义算法的参数;
- 参数名:参数的别名。
- 在云平台中,提供了6种参数给自定义算法的开发者使用:
- devType: 设备类型;
- devld: 创建设备定义的 ID, 可以用于选取一个设备进行统计;
- dataSource: 设备类型功能块中的某一个字段;
- dataCompareOperation:字段的比较条件,有大于、小于和等于;
- dataValue: 需要比较的数据值;
- group:软件分组,可以对一整个软件分组进行统计。

2. 算法代码

统计数据的过程代码,可以使用 Groovy 或者 JavaScript 语言编写。

3. 返回参数

定义统计后结果显示的字段。

这里以统计一小时内设备的某个数据的总和为例,编写一个 JavaScript 脚本:

// 载入需要实现的接口

var DataStatisticScriptMethod = Java.type("com.zlgcloud.iotplatform.

iot data. service. data. statistics. Data Statistic Script Method");

// 创建继承实现接口的类

var JsDataStatisticScriptMethod = Java.extend(DataStatisticScriptMe
thod);

// 创建 func 对象

var func = new JsDataStatisticScriptMethod() {
 execute: function(param) {

// 获取创建算法页面的请求参数,参数值会由云平台根据算法实例

的选择项,进行自动填充

var thirdThingsId = param.get("devId");

var infoModelId = param.get("devType");

var property = param.get("property");

// 可以自定义统计时间范围,如果查询条件没有 range,则会根据 算法实例的数据开始时间和数据结束时间范围进行统计

var currentTime = new Date().valueOf();

var anHourAgo = currentTime - 3600000;

// 拼装 elasticseach 查询的 dsl

// 这里根据业务需要设置数据范围.查询时会限制租户和用户管理的设备.

var esParam = {

```
"aggregations": {
    "result_in_hour": {
      "sum": {
        "field": "data." + property
     }
    }
  },
  "query": {
    "bool":{
      "must": [
       {
          "term": {
              "third_things_id.keyword": thirdThingsId
          }
        },
        {
         "range": {
          "timestamp": {
           "gte": anHourAgo,
           "lt": currentTime
          }
         }
        }
      ]
    }
  }
};
// 获取统计结果
var result = Java.super(func).queryEs(infoModelId, esParam);
```

var rtn = {"result_in_hour":result['aggregations']['result_in_hour']

['value']};

```
return rtn;
```

```
}
```

}

将脚本复制到自定义算法中。



同时创建实例,给设备的数据进行统计。

添加实例				
实例名称 定时统计		4/50		
请求参数				
参数名		描述	参数值	
devType		设备类型	invert	
devld		设备id	反转设备	
property		数据项	总发电量	
统计设置				
统计周期 自定义				
重复方式 💿 周期性执行	○ 指途时间周期执行			
执行周期 每隔 1	0	分钟 ~		
数据开始 ③ 2023-02-1	3 00:00:00			
数据结束 ③ 2023-02-1	18 00:00:00			
生效时间 ③ 2023-02-1	13 11:04:40			
朱效时间 ③ 2023-02-1	18 11:04:40			
Rojiji daže				

执行算法过后,可以看到自定义统计结果。

算品評問			BR BERG STATE
Rane	1511-04B		
201	abt1平W U 2523-63-14 00 00:00 至 2523-63-17 00:00:00 開加		
1942	807568	WUTHD	847638
	("www/jin_hour"3003.2099966426420	(6)十年	2023-02-16 11 00 00
2	("result_in_hour")2853.519999921322	801-11	2023 02 16 10,50:00
3	("www.jin_hour"2290.0999943614006)	957+	2023-02-16 10:40:00
4	Chew/t_Jrt_hour12915.3199992173871	627.0	2023-02-16 10:30:00
5	["wwwft_in_hour"\\$126.2800097465515]	162+1	2023 02 16 10 20:00
6	("result_in_hour"3200.1000117452076)	811+	2023-02-16 10.10:00
7	("result in hour13262,929015335083)	827.0	2023-02-16 10:00:00



【产品应用】 如何利用IoT云平台统计设备数据?— 下载篇

原创 研发部 ZLG 致远电子 2023-05-30 11:37:01

在云平台统计完设备数据后,有时需要下载统计数据。以往是把统计数据全部打包下载,这不仅数据量大,而且格式单一,不便于二次分析。 为此,ZWS 云平台支持通过编写脚本来提取统计数据、转换成不同的 文件格式。

应用场景

上一篇的《如何利用 IoT 云平台统计设备数据? — 进阶篇》介绍了如 何在 ZWS 云平台中创建自定义统计算法来实现项目的业务数据统计。数据 统计之后会产生统计结果,用户可以在云端在线查看结果,也可以下载统 计结果到本地进行二次分析。

通常情况下,用户在云平台下载设备统计数据,会遇到以下问题:

 有些业务统计数据下载,需要指定数据项的先后顺序或者只提取部 分统计数据项。

2. 某些特殊业务,需要用特定专业软件工具进行分析处理,而工具只 支持打开指定的文件格式,比如: ASC、BLF 格式等等。

为了满足用户不同的下载需求,ZWS 云平台支持自定义下载脚本,开 发者可以通过编写下载脚本,来决定提取哪些统计数据项以及用什么格式 进行下载。

使用下载脚本步骤

在 ZWS 物联网云平台数据统计模块,每个统计算法,都可以编写独立 的下载脚本,步骤如下:

	1.	泉	山	"	,点击【卜载脚本】	0	
	#/I		8.965				BR > RHUT
8	0884		. • •	8-x1912			
	设施管理		19.4	8259	Page	.82	10.0
-	1027		1	核计论各类型所有的设备数据	统计一个信誉减至了的所有信息数据	nz	T2225 ## 105
	APTAIR.		2	核计数量分级的设备数据	统计每个分级设备数据约束加	nz	TENA WE IN
	224,808		- 1	(最高数38620-515	统计设备数据的某个数语是一段的间内的态能	ne .	TENA NO INI
	ecima.		4	计信息条约数据意志	统计分组设备在一届时间的意料	7.8	TENS WE IN
_	10月1日	_	5	1400m3823	业务的教育条则	7.8	Yalkin mit inti
	matrix:			1 N R S R 128 R			
0	1064						
1.	用小常用						

 2. 选择脚本语言,编写下载脚本内容(处理数据项),编写下载模板(定 义文件内容格式)。



3. 在统计结果页面,点击【下载】。

CAN IN					NUM - NUMPLIF - HOUSE
er as	442+42.00				
RR.	search - alleast 04 - 08 and		0 2023-04-11 16:55:53 至 2023-05-11 16:55	25 R.R. Fit. +	
19.1	851×65.86	1011058	INTERN	10.00.07 (0.01.04)	和油店用+14
1	["gw_zigbee_lamp_fn-data.hour", 03, "gw_zigbee_lamp_fn-data.p	核计完成	2023-05-10 11:00:00	2023-05-10 10.50.00	2023-05-10 11:00:00
2	("gw.zigbee.lamo.ln.data.hour": 47, "gw.zigbee.lamo.ln.data.p	统计完成	2023-05-10 11:00:00	2023-05-10 10:59:00	2023-05-10 11:00:00
3	("gw_zigbee_lamp_fn-data.hour". 87, "gw_zigbee_lamp_fn-data.p-	統计完成	2023-05-10 11:01:00	2023-05-10 11:00:00	2023-05-10 11:01:00
4	I'pw zigbee lamp in data.hour": 80, "pw zigbee lamp in-data.p -	统计完成	2023-05-10 11:01:00	2023-05-1011:00:00	2023-05-10 11:01:00
5	["pr_zighee_lamp_fn-data.hour", 75, "pr_zighee_lamp_fn-data.p	统计完成	2023-05-10 11:02:00	2023-05-1011.01.00	2023-05-10 11:02:00
6	("gw_zighee_lamp_fn_data.hour":32, "gw_zighee_lamp_fn_data.p	统计完成	2023-05-10 11 02:00	2023-05-1011-01-00	2023-05-10 11:02:00
7	("prc_tighter_lamp_fr-data.hour", 94, "prc_tighter_lamp_fr-data.p	统计传域	2023-05-10 11.03.00	2023-05-10 11.02.00	2023-05-10 11.03.00
8	("ps_righee_lamp_fn_data.hour") 75, "ps_righee_lamp_fn_data.p-	0011010	3023-05-10 11:03:00	2023-05-1011-02-00	2023-05-10 11:03:00
.9	("prc_zigbee_lamp_fridata.hour".10, "prc_zigbee_lamp_fridata.pri	统计完成	2023-05-10 11:04:00	2023-05-10 11:03:00	2023-05-10 11.04.00
20	("gw_zigbee_lamp_fn_data.hour": 31, "gw_zigbee_lamp_fn_data.p	统计常成	2022-05-10 11:04:00	2023-05-10 11:03:00	2023-05-10 12:04:00

4.	选择一个	\uparrow	卜载脚本,	进行	卜载即可。

101000 - Incore 128 - 10			·	o 100 TH	
	DERIC	ASC程式			
RUMA .			PSN .	MUMOR REVENUE	RENARCH
("pw.righee.lamp_hedata.tear") 93, "pw.righee.lamp.feda		45780.0	0 11 00.00	2023 05-10 10:55:00	2023-05-10 11.00.00
("per_sighter_tamp_in_data.http://st. *pe_sighter_lamp_in_da-	_	hestroy	10 11 99 00	2023-05-10.10:59:00	2023-05-10 11:00:00
$\label{eq:product} ["gen_signer_lamp_fredstates"" B1, "ge_stgter_lamp_fredstates" - \mbox{"gen_signer_lamp_fredstates - \mbox{"gen$		and party	13-05-10 11 01:00	2023-05-10 11:00:00	2023-05-10 11:01:00
Tipe, righer, lamp, fri data hour? \$5, "ga, righer, lamp, fri data p		ALL FROM	3923-05-10 11.01.00	2013-05-10 11:00:00	3023-05-10 11:01:00
("pe_agnee_lamp_te-detailour") 75, "pe_agnee_lamp_te-detaip-		机计元库	2023-05-10 11 02 00	2023-05-10 11:01:00	2023-05-10 1132-08
("projugates, large frontationes", \$2, "projugates, large for data p-		相计元成	3023-05-10.11/02/00	2023-05-10 11:01:00	2022-05-10.11/02:00
("gw zigbee land fwdata.hou") 94, "gw zigbee land, fwdata.p-		放け完成	2023-05-10 11 02.00	2023-05-10 11:02:00	2023-05-10 11:08:00
Pgw_sighee_lang_hvdsta.hour*/75, "gw_sighee_lang_tvdstap		抓以売成	2023-05-10 11:03:00	2023-05-10 11:02:00	2023-05-10 11:03:00
I'pw sighee lante Andata koor" 98, "pw sighee lante, in data p-		就计完成	2023-05-10 11:54:00	2023-05-10 11 03:00	2023-05-10 11:04:00

编写下载脚本,需要对数据结构有所了解,这里仅介绍添加下载脚本的操作过程。如果需要了解更多关于脚本的内容,请联系致远电子技术支持。



【产品应用】 基于CAN智慧云实时查看汽车GPS轨迹

原创 研发部 ZLG 致远电子 2023-05-25 11:40:56

汽车路试场景下,测试车辆还未完善,缺少完整的远程监控功能,难 以查看车辆的路程信息,适配 ZLG 致远电子提供的 CANDTU 系列设 备与 CAN 智慧云,可以云端实时查看路试车辆行驶轨迹与速度。

应用场景

在汽车路试中,众多厂商只能获取部分的测试数据。然而实际上,测 试工程师不但需同时查看多台测试车辆当前的位置,还需查看路试的行驶 轨迹或车辆的行驶速度,配合测试的数据进行相互校验。ZLG 致远电子的 CANDTU 系列设备自身配带 GPS 模块,可将行驶数据上传至云端,实现实 时查看路试车辆行驶轨迹与速度,解决该痛点。

实际操作

1. GPS天线连接

在 CANDTU 系列的产品上,查看 GPS 连接口,将 GPS 天线插入。



2. CANDTU工具配置GPS使能

在通讯 >GPS 配置中,对 GPS 配置进行选定,自行选定记录周期与定 位系统,而在云服务器配置中,对 GPS 数据上传使其能进行选定。





3. 云端添加CANDTU设备

登录 CAN 智慧云,添加 CANDTU 设备。

https://cans.zlgcloud.com



4. 云端查看轨迹信息

在设备列表中,点击【查看】设备,跳转到设备详情页。在 GPS 模块中, 点击【开始】,即可查看车载设备的实时轨迹信息、总里程、平均速度。



在历史轨迹模块中,可以根据时间区间,搜索车载设备的历史轨迹信息。



【技术分享】 M6Y2C+ePort-M辐射整改

原创 研发部 ZLG 致远电子 2023-05-15 11:36:02

本文以 M6Y2C 核心板平台搭配 ePort-M 的工控整机为案例,通过测试 结果直观感受时钟驱动强度与 Slew Rate 对辐射测试的影响。

ePort-M 模块与 MAC 端通过 RMII 接口进行通讯,其 Pin6 脚为 RMII 的 参考时钟 REFCLK,频率为 50MHz,该时钟由 M6Y2C 核心板提供,ePort-M 作为接收端。下图 1 和图 2 所示为某工控整机的辐射测试结果,该工控整 机使用 ZLG 致远电子 M6Y2C 系列核心板搭配 ePort-M 模块,整机需要通过 辐射测试 Class B 等级,从图中可以看出该整机在水平方向与垂直方向上的 50MHz 及其倍频点上辐射值均较大,辐射超标频点也集中在此。



图1水平方向



图2 垂直方向

当产品已经成型,而辐射却超标时,除了改动硬件,有没有更简单便 捷的方式来尝试降低辐射值呢?答案是有的,那就是通过软件调节相应时 钟引脚的驱动强度以及 Slew Rate。从上图的超标频点以及该工控整机的 板上外设可以基本确定,超标频点来源于 M6Y2C 核心板输出的 RMII 接口 REFCLK 50MHz 时钟信号。 通过查阅 M6Y2C 核心板处理器 NXP Cortex-A7 i.MX6ULL 手册得知, ENET1 REFCLK 时钟信号引脚的控制寄存器为 "IOMUXC_SW_PAD_CTL_ PAD_ENET1_TX_CLK",其驱动强度 Drive Strength 与 Slew Rate 的说明 如下图 3 所示。

5-3 DSE	Drive Strength Field
002	Select one out of next values for pad: ENET1_TX_CLK
	000 DSE_0_output_driver_disabled_ — output driver disabled;
	001 DSE_1_R0_260_Ohm3_3V150_Ohm_1_8V240_Ohm_for_DDR_ — R0(260 Ohm @ 3.3V, 150 Ohm@1.8V, 240 Ohm for DDR)
	010 DSE 2 R0 2-R0/2
	011 DSE_3_R0_3 - R0/3
	100 DSE_4_R0_4 R0/4
	101 DSE_5_R0_5 - R0/5
	110 DSE_6_R0_6 R0/6
	111 DSE_7_R0_7 R0/7
2-1	This field is reserved.
-	Reserved
0 SRE	Slew Rate Field
	Select one out of next values for pad: ENET1_TX_CLK
	0 SRE_0_Slow_Slew_Rate — Slow Slew Rate
	1 SRE_1_Fast_Slew_Rate — Fast Slew Rate
	图3
⊦ờ	图 1 和图 2 测试结果对应的驱动强度为 "110",Slew Rate 为 "Fast

上文图 1和图 2 测试结果对应的驱动强度为 "110", Slew Rate为 "Fast Slew Rate",现在我们将该时钟引脚的驱动强度调低,调至 "001" 最低档(以 太网通讯正常),此时辐射测试结果如下图 4 和图 5 所示,可见其辐射值 大幅下降,超标频点也减少许多。







图5 垂直方向

保持驱动强度最低,同时将 Slew Rate 修改为"Slow Slew Rate"(以 太网通讯正常),此时辐射测试结果如下图 6 和图 7 所示,可见该工控整 机此时已成功通过辐射测试 Class B 等级。 注:针对时钟驱动等的调节本文未对具体操作方法进行说明,若有需要,可联系致远电子技术支持或加入工控板 / 核心板技术交流群。

Cortex-A7工业控制核心板





图7 垂直方向



ZLGi微文摘

互联互通▼

【技术分享】 蓝牙技术的前世今生

原创 研发部 ZLG 致远电子 2023-05-08 11:37:30

蓝牙是一种支持设备短距离通信的低功耗、低成本无线电技术。它利 用短程无线链路取代专用电缆,便于人们在室内或户外流动操作。那 么这种技术为什么叫蓝牙?又历经了怎样的发展?本文将带你了解蓝 牙技术的前世今生。

蓝牙的由来

"蓝牙"(Bluetooth)原是一位在10世纪统一丹麦的国王哈拉尔(HaralBluetooth),他将当时的瑞典、芬兰与丹麦统一起来。而将"蓝 牙"与后来的无线通讯技术标准关联在一起的,是一位来自英特尔的工程师 JimKardach。他在一次无线通讯行业会议上,提议将"Bluetooth"作为无 线通讯技术标准的名称。用"Bluetooth"名字来命名这种新的技术标准,含 有将四分五裂的局面统一起来的意思。

1998 年,东芝、爱立信、IBM、Intel 和诺基亚共同提出近距离无线数字 通信的技术标准。蓝牙标准正式形成。



蓝牙技术的变迁史

1. 第一代蓝牙

蓝牙1.0标准 1999年发布,传输速率约在723.1Kbit/s,通信距离约10米, 蓝牙1.0存在很多问题,并未得到广泛的应用。

蓝牙 1.1 标准:2001 年发布,正式列入 IEEE 802.15.1 标准,该标准定 义了物理层 (PHY) 和媒体访问控制 (MAC)规范,用于设备间的无线连接, 易受到同频率之间产品干扰,影响通信质量。

蓝牙 1.2 标准:2003 年发布,传输速率 748~810Kb/s,增加了抗干扰跳 频功能。

2. 第二代蓝牙

蓝牙 2.0 标准:2004 年发布,新增的 EDR(Enhanced Data Rate)技术 通过提高多任务处理和多种蓝牙设备同时运行的能力,同时开始支持双工模 式。

蓝牙 2.1 标准:2007 年发布,蓝牙技术联盟(Bluetooth SIG)正式批准 了蓝牙 2.1 核心规范,即"蓝牙 2.1+EDR"新增了 Sniff Subrating 省电功能, 让蓝牙芯片的工作负载大幅降低。

3. 第三代蓝牙

蓝牙 3.0 标准:2009 年发布,蓝牙技术联盟(Bluetooth SIG)正式颁布 了新一代标准规范蓝牙 3.0,数据传输率提高到了大约 24Mbps。

4. 第四代蓝牙

蓝牙 4.0 标准: 蓝牙 4.0 是迄今为止第一个蓝牙综合协议规范,将三种 规格集成在一起。其中最重要的变化就是 BLE(Bluetooth Low Energy)低

功耗功能。从蓝牙 4.0 协议开始,后面的版本都包含了经典蓝牙和低功耗蓝 牙两种模式。

蓝牙 4.1 标准:2013 年发布,提升了连接速度并且更加智能化,同时也 提升了传输效率。

蓝牙 4.2 标准:2014 年发布,增强了安全性,改善了数据传输速度,比 上一代提高了 2.5 倍。

5. 第五代蓝牙

蓝牙 5.0 标准:在低功耗模式下具备更快更远的传输能力,传输速率是 蓝牙 4.2 的两倍(速度上限为 2Mbps),有效传输距离是蓝牙 4.2 的 4 倍(理 论上达 300 米),数据包容量是蓝牙 4.2 的 8 倍。

蓝牙 5.1 标准:2019 年 1 月发布,新增寻向功能(AOA/AOD),将蓝牙 定位的精准度提升到厘米级。

蓝牙 5.2 标准:2019 年 12 月发布,新增三项主要特性:增强版 ATT 协议、 LE 功率控制、LE 同步信道,为蓝牙低功耗音频方案提供基础。

广州致远电子股份有限公司推出的 ZLG52810 系列是一款全新的高吞吐 量的蓝牙模块,具有低功耗、小尺寸、低成本等特点。



蓝牙技术分类

从蓝牙 4.0 协议开始,后续的版本都包含经典蓝牙和低功耗蓝牙两种版本。经典蓝牙和低功耗蓝牙是两种完全不同的技术,两者不能互相通信。

经典蓝牙(Bluetooth Classic):即基本速率 / 增强数据速率蓝牙(简称: BR/EDR),采用自适应跳频方法,在 79 个通道上传输数据。

低功耗蓝牙(Bluetooth Low Energy):低功耗蓝牙在蓝牙 4.0 协议中 第一次提出,是为了实现极低功率运行而设计的。该技术采用跳频扩频方法, 支持在 40 个信道上传输数据。

一般来说,经典蓝牙目前主要用于音频,例如无线电话连接、无线耳机 和无线扬声器,用于数据量比较大的传输;低功耗蓝牙更常见于可穿戴设备、 智能物联网设备、健身监测设备和电池供电的配件(如键盘)。

我们常常听到单模蓝牙和双模蓝牙的说法。其实,单模蓝牙指的就是低 功耗蓝牙。而双模蓝牙则是同时支持低功耗蓝牙和经典蓝牙,最常见的就是 手机或者笔记本电脑,这些产品既能连接经典蓝牙设备,也能连接低功耗蓝 牙设备。

	Bluetooth Low Energy (LE)	Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR)				
Frequency Band	2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)	2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)				
Channels	40 channels with 2 MHz spacing (3 advertising channels/37 data channels)					
Channel Usage	Frequency-Hopping Spread Spectrum (FHSS)	Frequency-Hopping Spread Spectrum (FHSS)				
Modulation		GFSK, π/4 DQPSK, 8DPSK				
Power Consumption	~0.01x to 0.5x of reference (depending on use case)	1 (reference value)				
Data Rate	LE 2M PHY: 2 Mb/s LE 1M PHY: 1 Mb/s LE Coded PHY (S=2): 500 Kb/s LE Coded PHY (S=8): 125 Kb/s	EDR PHY (8DPSK): 3 Mb/s EDR PHY (π/4 DQPSK): 2 Mb/s BR PHY (GFSK): 1 Mb/s				
Network Topologies	Point-to-Point (including piconet) Broadcast					

图3低功耗蓝牙VS经典蓝牙

互联互通▼

【技术分享】 物联网ZigBee技术详解② — 组网类型

原创 研发部 ZLG 致远电子 2023-05-22 11:45:26

ZigBee 是无线通信中最重要的无线协议之一,在各种物联网通信领域 中得到了广泛应用。ZigBee 技术在物联网应用中的成功,与其强大的 组网功能密不可分。本期文章和你一起了解 ZigBee 的组网类型。

ZigBee网络设备类型

ZigBee 为了充分利用硬件资源避免硬件资源的浪费,同时兼顾低功耗、 低成本的硬件考虑,定义了两种设备类型,它们分别是全功能设备和精简功 能设备。

- 全功能设备 (Full-Function-Device): 具有数据发送、接收和转发的 能力。能建立新的网络,也能进行路由发现和维护等功能,提供全 部的 IEEE 802.15.04 MAC 服务。
- 精简功能设备 (Reduced-Function-Device):占用资源少,存储容量小,成本低。只具有数据发送、接收的能力,提供部分的 IEEE 802.15.04 MAC 服务。

ZigBee网络设备角色

定义了设备类型,接着就要让设备在一个网络中担任不同的角色。在 ZigBee 网络中,有三种设备角色:协调器,路由器、终端。三种设备角色有 以下特点:

- 协调器 (Co-Ordinator): ZigBee 协调器是整个网络的信息集合点和核 心节点。它负责网络的构建、维护和管理。协调器通常为 ZigBee 的 网关,负责与 Wi-Fi 等其他协议的转换,同时具有路由器的所有功能。
- 路由器 (Router):路由器能进行数据的收发,并负责数据的路径搜索和维持,使路由器或终端设备能加入到网络中,通常作为协调器 与终端设备的中继节点使用。
- 终端设备 (End Device):终端设备能进行数据的收发,但不能进行数据的路由。终端设备只能挂载到协调器或者路由器节点。通常为低功耗设备,如挂载各类传感器、继电器、开关等。

星型组网

星型网络是最常见的网络结构。星型网络中一般包括协调器和终端两种 设备,协调器作为中央节点,所有终端节点的信息都直接发送给中央节点, 形成星型结构。这种组网结构最大的特点是简单易用,由于所有节点均要连 接在中央节点下,支持的节点数量较少,网络规模较小。



ZLG 致远电子推出的 FastZigBee 协议使用了星型组网结构,与传统星型 网络不同的是,FastZigBee 没有协调器(中央节点由某个终端设备担任), 所有设备均为终端节点,采用点对点的通信方式,节点与节点间可以互相通信。



Mesh组网

Mesh 网络一般由协调器、路由和终端组成。其中协调器是必须的,它 用于网络的组建和维护。而路由可以进行数据转发,规划信息路径,让通信 路径更加灵活的同时,增大网络规模。在 Mesh 网络中,终端信息可以经过 多级路由跳转,最终到达目标节点。信息传递的路径也不是唯一的,各级路 由会根据网络环境,选择最合适的传递路径,网络可靠性和灵活性更高。相 较于星型网络, Mesh 网络的网络规模更大,支持的网络节点数量更多,且由 于多级路由的使用,组网距离也大大增加。



图2 Mesh组网

ZLG 致远电子推出的 ZLG Mesh 协议使用了 Mesh 组网结构,该协议具 有以下功能特点:

快速添加路由,智能路由算法,快速恢复通信; 实现自动维护路径,不需要手动配置路由器;

支持多级跳转,最高15级路由。

需要注意的是,FastZigBee 与 ZLG Mesh 属于两种不同的通信协议,两 者不能互相通信。



ZigBee高性能透传模组 ZM21P2S24E / ZM21P2S24S

🖑 点击购买

【技术分享】 一文读懂蓝牙模块的工作模式

原创 研发部 ZLG 致远电子 2023-05-29 11:32:06

蓝牙是无线通信中重要的通信协议之一,广泛应用于各种物联网通信 领域。蓝牙设备在各种应用场景扮演着不同的角色,而蓝牙应用场景 与其工作模式息息相关。本期文章将带你了解蓝牙的工作模式。

主机模式/从机模式

主机模式: 蓝牙模块能够扫描从机的广播信号并主动建立连接,可以和 一个或多个从设备进行连接通信,主机定期扫描周围处于广播状态的设备发 送的广播信息,连接后主从双方可以发送接收数据。如智能手机,数据传输 中做主机的蓝牙模块。

从机模式: 蓝牙模块首先进入广播状态,等待被主机扫描,当主机扫描 到从设备建立连接后,可以和主机设备进行数据收发,从机不能主动建立连接, 只能等待主机扫描连接。与广播者模式的区别在于从设备模式的蓝牙模块是 可以被连接,定期和主机进行连接和数据传输,在数据传输过程中作从机, 而处于广播者模式的设备是不可被连接,如蓝牙手表手环,蓝牙鼠标等。

广州致远电子股份有限公司推出的 ZLG52810 系列是一款全新的高吞吐 量的蓝牙从机模块,具有低功耗、小尺寸、低成本等特点。



主从一体工作模式

主从一体工作模式是指蓝牙模块可以同时作为主设备和从设备。其可以 在两个角色间切换,工作在从模式时,等待其它主设备来连接,需要时转换 为主模式,向其它设备发起连接。主从一体额外增加了蓝牙模块的功能,促 使成本优化和增强其易用性,优化系统架构。

ZM52820P 是一款 BLE 主从一体数传模块,最高支持 BLE5.0 蓝牙协议, 具有低成本、小尺寸但速率更高、功耗更低等优点。该模块使用简单方便, 能迅速桥接电子产品和智能移动设备,广泛应用于各种电子设备,如仪器仪表、 健康医疗、智能家居、可穿戴设备、汽车电子和数码产品等。该型号是我司 即将发布的新产品,更多信息欢迎咨询我司销售人员。



图1主从一体模式

广播者/观察者模式

广播模式: 蓝牙模块定期持续的向周围发送一定长度的广播数据包,该 数据可以被扫描者搜索到。模块可以在低功耗的模式下持续的进行广播,应 用于极低功耗,小数据量,单向传输的应用场合。蓝牙广播通道的重要功能 就是用于发现设备,发起连接和发放数据。

观察者模式 该模式下模块为非连接,相对广播者模式的一对多发送广播, 观察者可以一对多接收数据。在该模式中,设备可以仅监听和读取空中的广 播数据,不能发起连接,只能持续扫描从机。观察者工作模式可应用于数据 采集集中器的应用场合,如传感器集中器采集等功能。



iBeacon模式

iBeacon 是苹果公司于 2013 年 9 月发布的一种基于低功耗蓝牙的通信协议, 通信协议名称: iBeacon, 工作方式是: iBeacon 是一个低功耗的蓝牙信标, 基于 BLE 技术, 工作在广播模式, 利用 BLE 中名为"通告帧" (Advertising) 的广播帧, 定期发送通告帧, 支持 BLE 的设备可以接收到。广播的发射功率 可以调整, 不停广播蓝牙设备的 mac 地址、UUID 等固定字节的字符串信息。

常见应用于精确营销,如博物馆、展厅的信息推送或者购物中心服务商 向顾客发送折扣卷及进店积分等。如果把它放在室内固定的位置,可以作为 定位器,手机打开蓝牙连接之后通过 APP 就可以获取其位置,同时会推送设 置好的信息到手机上。蓝牙 iBeacon 是一种低功耗协议,也是一种低功耗蓝 牙设备,在智能领域有非常广泛的前景。



iBeacon 图3 iBeacon模式

互联互通▼

【产品应用】 CANopen主站如何获取Modbus设备的数据

原创 研发部 ZLG 致远电子 2023-05-11 11:31:35

CANopen 通信硬件接口为 CAN 总线,然而 Modbus 通信硬件接口为 串口 / 以太网口;那么问题来了:CANopen 主站如何获取 Modbus 型 传感器的数据呢?让我们一起来实现。

PXB-6021 是 ZLG 致远电子推出的一款 Modbus to CANopen 协议转换器, 具有 1 路 CAN, 1 路百兆以太网和 1 路 RS485。该产品支持 Modbus RTU 及 Modbus TCP,可以实现 Modbus 主站到 CANopen 从站的数据转换。PXB-6021 最大支持 12 组 TPDO 和 12 组 RPDO,同时支持 EDS 下发和自定义 PDO 和 Modbus 的数据映射;具有转换快、设置简单、使用灵活等特点。

接下来我们将使用 USBCAN-E-P CANopen 主站卡和 PXB-6021 协议转换 器来实现 CANopen 主站获取 Modbus 设备的数据



<u>立即购买</u>

准备条件

- USBCAN-E-P CANopen 主站卡和上位机软件 CANManager for CANopen;
- PXB-6021 转换器和上位机配置软件 AWPX Tools;
- RS485 温湿度变送器;
- 24V 电源适配器;
- 杜邦线6根,USB线1根,网线1根,终端电阻120欧1个。

硬件接线

设备接线如图1所示:



图1 设备连接图

- PC 通过网线连接 PXB-6021 和 USB 线连接 USBCAN-E-P CANopen 主站卡;
- PXB-6021的 CAN 总线的 CAN_H 和 CAN_L 分别连接到 USBCAN-E-P CANopen 主站卡的 CAN_H 和 CAN_L;
- PXB-6021 的 RS485 的 A、B 分别连接到 Modbus 温湿度变送器的 A、 B 端子。

PXB-6021的参数配置

打开 AWPX Tools 工具,点击搜索设备,然后选中 PXB-6021 设备。 根据温湿度传感器的说明书,温湿度变送器的波特率为 9600,数据长度 为 8,校验位为 0,停止位为 1,配置 PXB-6021 的 Modbus 参数,如图 2 所示:

		at scentar	10 MRH *	
目标板				
[00:14:97:0f:00:0	c]-192.168.1.13	16 *	工作模式 Modbus RTU主站	*
10 汉韩操类型 PXB-6021		~	波特軍 9600	
设备配置			(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	Ţ
Mod	lbusi@W			
CAN	npen恭敬 DO参数		1 1	•
TP	DO参数		R28102 None	

图2 设置Modbus参数 然后再配置 PXB-6021 的 CANopen 参数,如图 3 所示。

图3 设置CANopen参数

根据温湿度变送器的介绍,温度数据在保持寄存器 0 地址,湿度数据 在 1 地址,数据均扩大了 10 倍。我们还要对 PXB-6021 的 TPDO 参数以及 Modbus 和 TPDO 的映射关系进行配置,如图 4 所示。

MARK Tools							1											- 0
Q、扫描设备 ∓ GAPACE 📑	F SERIES	an 2	0 75.0	iQN -														
目标板																		iata -
{00:14:97:0f:00:0c]-192.168.1.136 *		PDOT	31	10/7-1	●元	91	编标量		位编标量		Modbus	宇节序	Modbus/ki81D	Modbus実:	¥		地址	18/5
协议转换类型		1	*	WORD	×	0	*	0		e.	大端	*	1	保持寄存器	v	0		
PXB-6021 ~		2		WORD	+	0	*	0			大調		1	保持责存器		1		
设备配置																		
Modbusi9 爾																		
CANopen参数																		
RPDO参数																		
TPDO参数																		
						攵	ит	D		π.	罟							
						ŝ	4 1	Г	00	Ħl	追							

CANopen主站设置

USBCAN-E-P CANopen 主站卡加载从站 EDS,如图 5 所示。

業性	Damen 1	^	屬性紅
	UNICUS XIV		6181
	15521# F15001.105 ×		日 炎条供用
	与入加5文件 整型数据按理认为式目示 ▼		(2 景美型(100)
			设委会称(100)
			2.94世中版本(
	BA 1(TPD0 1)		10日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日
			D SCB1990(1010)
	1000 (#c): 58/2		08/12
			000
			序列号
	編入 2(TPDO 2) ×		E 1837/2493/32
			输送控制协议
	1000 (0.1100) (0.1100)		用[周](ma)
			間期系数
	L	¥	E PDO
	日本愛口	×	g RPDO 1
	岸马 助司 信息		B RPDO 2
			(B RPDO 3
			8 10.004
			El INPOUS
			and the second s

图5 导入EDS

然后点击 CANManager for CANopen 软件的搜索按钮搜索从站,如图 6 所示。



找到从站,如图7所示。

 No.
 No.</th

图7 找到从站

Modbus数据获取

单击启动按钮采集从站数据,如图8所示。

									不 運性性	
 PX86021.ED5 	WA I(IPDO I)								8191	
									日 没参信:	2
						後の東朝ims	3 1000	特定接収	29.93	E(100C 0x000F0191
									2821	R(100E CANopen SI
	HEA 2(TPDO Z)							*	080	HERE VILLA
	The second second	1 10 100	1	1 1 1 1 1 1	1 5 44				0.28	adi/101m
	0x01 0x18	0.00	0000	0400	0x00	0400 040	0		(FR)	D 0x0000003F
						後の周期(ma	0 1000	停止接农	0.94	CE 0.00000000
									1913	\$ 0x00050026
	編入 3(TPDO 3)							A.	19(3)	0.00000001
	inco iner	1 0.00	200	1000	0.00	0.00	0		E 10/2021	65512
	Save Saba	0.000	0.000	1 10400	DADV	0400 040	lana l		10,7321	4990 100 1000
						獲砍問題(ma	1000	停止接收	10,000,000	0 500
									F1 PD0	a
	输入 4(TPDO 4)								IN RPD	0.1
									(8) RPD	D Z
							. (internal)	-	E RPD	D 3
						SECCEPEDING	0 1000	9290	B RPD	D-4
									E RPD	D 5
	输入 5(TPDO 5)									

图8采集数据

从图上可知:

- 温度为 0x118,转换十进制为 280,由于参数扩大了 10 倍,所以当前 温度为 28 度;
- 湿度为 0x294,转换十进制为 660,由于参数扩大了 10 倍,所以当前 湿度为 66。

感知控制 ▼

【技术分享】自动收发RS-485偏置电阻与终端 电阻的选用,你知多少?

原创 研发部 ZLG 致远电子 2023-05-06 11:49:48

RS-485 自动收发电路比带控制脚电路在应用上少一个 I/O 脚,在主控 资源紧张时会更受欢迎。那么自动收发电路是怎么实现自动收发功能 以及在选用偏置电阻与终端电阻时需考虑什么因素呢?

RS-485自动收发原理

常见 RS-485 自动收发电路简图如图 1 所示,将收发器接收使能脚 RE 和发送使能脚 DE 短接在三极管 Q1 的集电极,发送端 DI 和三极管 Q1 发射 集接地,MCU 的 TXD 接在三极管 Q1 的基极。以下是自动收发电路工作的 过程。

当 MCU 通过 TXD 发送 0 时,三极管 Q1 截止,DE 使能,因为 DI 接地, 此时发送低电平;

当 MCU 通过 TXD 发送 1 时,三极管 Q1 导通,RE 使能,此时收发器 的 DI 引脚与 AB 端之间为高阻状态,因为有 AB 的上下拉电阻的存在,此时 总线逻辑状态为 1,该节点进入接收模式和发送高电平模式。



图1 RS-485自动收发电路简图

偏置电阻与终端电阻的作用

RS-485 总线的偏置电阻主要是给 A、B 确定的逻辑状态,我司 RS-485 自动收发产品内部都带有上下拉电阻,用户可以根据实际应用环境如逻辑 1 电平幅值低等原因,选择外加一个较小阻值的上下拉电阻来提升驱动能力, 该电阻与模块内部 A/B 线自带上下拉电阻是并联的。

RS-485 总线的终端电阻主要是用于信号线的阻抗匹配、提供通信线 缆寄生电容能量的泄放路径、提高信号质量。我们常用的 RS-485 的屏蔽 双绞线的差分特性阻抗为 100Ω~150Ω,由于 RS-485 收发器输入阻抗较高 (RSM485PHT 的输入阻抗最小为 1/4 单位负载即 48kΩ),在信号传输到 总线末端时由于接收的瞬时阻抗发送突变,导致信号发送反射,同时若通 信距离远线缆寄生电容较大,能量泄放缓慢。这时我们需要选用终端电阻 来消除或降低此情况对通信信号的影响。

因为 RS-485 自收发电路总线端的逻辑 1 是通过 AB 的偏置电阻来提供, 其驱动能力要弱于推挽方式,因此自收发电路选用的终端电阻值一般偏大 并且一般会再加小的偏置电阻来调节总线的电压。

理想的RS-485总线电平

通常情况下,发送驱动器 A、B 之间的逻辑电平1位于+2~+6V,逻 辑电平0位于-2~-6V。接收器也作与发送端相对的规定,通常情况下当 接收端 AB 之间电平大于+200mV时,输出逻辑电平1,小于-200mV时, 输出逻辑电平0,见下图2。空闲时刻A、B 差分电平应处于逻辑1。



图2 RS-485逻辑电平图

考虑到线阻及信号的抗扰能力,通信时我们一般会使总线端的逻辑 1 电平尽可能远离 +200mV,逻辑 0 电平尽可能远离 -200mV。数据波形的上 升沿和下降沿尽可能地陡峭,同时波形没有过冲或振铃等。如下图 3 所示 为较为理想的 RS-485 通信波形图。



图3 较理想的RS-485通信波形图

偏置电阻的选用

本文以我司自收发产品 RSM485PHT 为例进行说明,该产品内置完整 的 DC-DC 电路、信号隔离电路,具有较强的抗扰性和高可靠性,具备自动 收发功能。该产品的 A、B 线内置 47kΩ 的上、下拉电阻,收发器输入阻抗 最小值为 48kΩ。

本次测试的硬件框图如图 4 所示,在通信速率为 500kbps、总线上挂 6 个通信节点、双绞线总长度约为 3m 的情况下,抓取总线波形如图 5 所示, 逻辑 1 的差分电压约为 1.60V。



图4 RSM485PHT组网通信框图



图53m双绞线、500kbps时VAB波形

下图为本次 RSM485PHT 测试的电阻分压等效图,当总线上有6个节点进行通信时,相当于6个R上、6个R下、6个R内并联,此时VAB高电平电压计算值为VAB=(R内/6)/(R上/6+R内/6+R下/6)*VCC,取VCC=5.1V,VAB=1.72V。考虑到线阻的分压,此计算值1.72V与实测波形幅值1.60V基本吻合。



图6 RSM485PHT电阻分压等效图

因为总线端逻辑 1 电平幅值仅有 1.6V 左右, 该幅值的抗扰能力相对较 弱且影响了通信距离的进一步拉长, 现考虑通过外加偏置电阻将总线幅值 抬高至 3.5V 左右。通过公式 VAB=(R内/6)/(R上等效 +R内/6+R下等效)*VCC, 可计算出 R上等效 =R 上等效 ≈ 2.75 k Ω , 外加上下拉电阻值约为 4.1k Ω 。 如图 7 所示为外接 3.5k Ω 上下拉电阻时总线的通信波形图(外加偏置电阻 增加功耗约为 5.1V/3.5k \approx 1.4mA, 在可接受的范围内), 因为实焊偏置电 阻值 3.5k Ω 小于 4.1k Ω , 实际总线逻辑 1 幅值为 3.92V 高于预设值 3.5V。



图7外加3.5kΩ偏置电阻时差分波形图

接入终端电阻120Ω*2

在上述接入 3.5kΩ上下拉电阻的环境下再接入 120Ω 的终端电阻,此时电阻分压等效图 6 中的 R 内等效≈ 60Ω,将各数值代入 VAB=(R 内等效)/(R 上等效 +R 内等效 +R 下等效)*VCC,计算得出电压约为 60mV,测试波形如 图 8 所示。此时高电平处于门限 -200mV~+200mV 门限内,收发器无法识 别逻辑 1,造成通信错误。



图8接入120Ω终端时差分波形图

总结

在使用我司自动收发模块 RSM485PHT 或 RSM485M 时,如果总线逻辑 1 电平较低可通过外接偏置电阻来调节总线电平,偏置电阻值过小将增加 额外的功耗,电阻值太大调节效果将不明显。偏置电阻值可以根据实际节 点数量算出等效电阻值,再代入阻抗分压公式(VCC*R内等效)/(R上等 效+R内等效+R下等效)=VAB 计算出,其中 VCC 可取 5.1V,VAB 一般取 2.5V~4.0V。

具备自动收发功能的模块 RSM485PHT 或 RSM485M 的总线逻辑 1 电平 是通过 AB 线的偏置电阻来驱动,其能力弱于推挽驱动,因此一般情况下我 们不建议用户加终端电阻。如果通信速率高、通信距离长,总线信号质量 很差,需要加终端电阻来减弱反射信号或提供泄放寄生电容能量的路径, 可以选择稍大阻值的电阻,并且可以考虑通过在 AB 线上加小阻值的偏置电 阻两者配合共同来调节总线的电平。

总的来说在使用自动收发 RS-485 进行通信时一定要确保 A/B 线差分电 压不会处于 -200mV ~+200mV 范围内;若逻辑 1 或逻辑 0 的差分电平幅值 较低,可以通过外加小的偏置电阻来进行调整;一般情况下不建议用户接 终端电阻,如果要接尽量选择较大的阻值同时与外接偏置电阻搭配使用。



感知控制 ▼

【技术分享】 双电源并用问题与解决方案

原创 研发部 ZLG 致远电子 2023-05-26 12:11:13

R 在集成度越来越高的电子产品上,往往同一块电路板上会设计多路、 多种电源以供不同的需求使用。组合使用不同电源的话,电源之间难 免会出现相互影响的情况。本文选择一种情况进行分析并提供参考解 决方案。

双电源并用问题

曾经有客户在电源模块应用过程中出现过这样的应用场景,如下图 1 所示。客户使用两路电源给后端电路进行供电,要求在不断电的情况下切换 输入电源,此过程中发现后端电路会出现损坏。对各个节点波形进行分析 后发现,在给 DC-DC 模块进行上电的时候,模块输出端会产生一个 13.12V 的尖峰电压,当尖峰电压超过了后端电路的承受电压 13V,就会导致后端电 路损坏。



图1 客户的应用电路图

双电源并用问题分析

对客户板子使用双电源切换的情况进行了了解,客户在使用过程中是 先通过 TYPEC 接口进行供电,然后再给 DC-DC 模块进行上电建立稳定的 12V 电源后,切换为由 DC-DC 模块进行供电。对客户电路进行分析后发现, 在通过适配器进行供电时,适配器的 12V 电压会同时连接到 DC-DC 模块的 输出端上。出现此情况原因可能是因为适配器的 12V 电压加到 DC-DC 模块 输出端上,破坏了 DC-DC 模块的反馈环路信号,导致 DC-DC 模块输出电 压出现尖峰,从而损坏了后端电路。通过对比验证在有无适配器供电时的 DC-DC 模块起机波形,验证我们的分析是正确的。对比波形如下图 2、图



图2先接入适配器供电后起机输出电压波形



图3直接起机输出电压波形

双电源并用问题解决方案

这个问题可以从源头或传播途径上去解决:从源头解决需要隔绝适配器电压对 DC-DC 模块的影响;从传播途径解决则需要将 DC-DC 模块产生的 尖峰电压消除掉。

1. 在 DC-DC 模块输出端增加一个二极管,以隔绝适配器电压的影响, 如图 4;

 2. 在后端负载电路前增加一个稳压二极管,将超出承受范围的电压尖 峰消除掉,如图 5。



图4 增加二极管电路



图5 增加稳压管电路

双电源并用问题解决方案验证结果

以上两种方案验证结果如图 6、图 7 所示。在 DC-DC 模块输出端增加 一个二极管能够完全隔绝适配器电压对模块环路的影响,DC-DC 模块在上 电的时候完全没有产生尖峰电压。但是由于二极管存在的压降问题,使得 DC-DC 模块对后端电路的电压下降。在后端电路之前增加一个稳压二极管 以消除掉产生的尖峰电压,能够看出来是有效果的,尖峰电压备压制到了 12.4V ,已经小于后端最大承受电压,对此电路是有明显效果的。如果后端 电路对于电压范围要求不是特别严格,选择在 DC-DC 模块输出端增加一个 二极管能后更快捷的解决问题;如果电压要求严格的话推荐在后端电路之 前接入稳压管更适合。



图6增加二极管起机输出电压波形



图7 增加稳压管起机输出电压波形



感知控制 ▼

【产品应用】 隔离收发器全"芯"升级,你知道哪些?

原创 研发部 ZLG 致远电子 2023-05-19 11:39:07

传统灌封类隔离收发器, 虽能满足一般的工业环境,但对于环境要求 更高场合并不适用,加上体积较大,用于高密度 PCB 器件布局更是捉 襟见肘。ZLG 致远电子全"芯"升级的隔离收发器恰恰解决了这些痛点。

新升级"芯"优势

ZLG 致远电子目前推出的 SM 系列全隔离收发芯片使用了先进的 SiP 封 装技术,相较于传统的灌封类技术, SM 系列全隔离收发芯片具有高集成度、 物理尺寸小、电性能高、稳定性好等特点。

1. "芯"体积

传统 CTM/RSM 系列灌封模块尺寸为 19.90*16.90*7.30mm, 而 SM 系 列全隔离收发芯片尺寸为 12.45*9.85*3.00mm,体积缩小约 85%,具体对 比如下图 1。



图1 RSM485ECHT与SM4500芯片体积对比

2. 温度覆盖范围更广

众所周知,不同的器件对工作温度要求不同。传统 CTM/RSM 系列灌封 模块工作温度处于 -40℃~+85℃的工业级别,而 SM 系列隔离收发芯片最 高工作温度已达 +125℃,具有更强劲的工况适应能力,满足绝大多数工业 现场应用需求。

3. 兼容3.3V和5V电路系统

不管是 CTM 系列还是 RSM 系列的收发器,都不能同时兼容 3.3V 和 5V 电路系统。如图 2, CTM/RSM 系列收发器的 IO 电平必须跟随电源,用于 3.3V 系统时需使用 3.3V 供电的模块,用于 5V 系统时需使用 5V 供电的模块。 而 SM 系列全隔离收发芯片具有独立的 VIO 逻辑供电引脚,用户只需要保证 VIO 电压与 MCU 接口电压一致,芯片就能正常工作。



图2传统收发器与"芯"收发器典型应用对比

4. 隔离耐压与ESD能力提升

SM 系列全隔离收发芯片的隔离耐压高达 3500VDC, 较一般的 CTM/ RSM 系列收发器提升了 1kV, ESD 能力也提升到了 IEC/EN 61000-4-2 Contact ±6KV 标准。在同等裸机条件下, SM 系列全隔离收发芯片抗干扰 能力更强。

5. "芯"优势总结

"芯"升级后亮点除了以上几点外,为了用户生产效率的提升,SM 系列全隔离收发芯片全部采用 DFN 贴片封装;在通讯速率方面,SM 系列 CAN 收发芯片,支持 CAN 及 CAN FD 协议,波特率覆盖 5kbps~5Mbps, SM 系列 RS-485 收发芯片波特率高达 10Mbps。"芯"优势总结如表 1。

尺寸	19.90*16.90*7.30mm	12.45*9.85*3.00mm	集成度更高,体积缩小 85%		
工作温度	-40°C~+85°C	-40°C~+125°C	温度覆盖范围更广		
系统兼容性	IO 逻辑跟随模块电源	具有 VIO 独立逻辑供电引脚	同时满足 3.3V 和 5V 电路系统		
隔离耐压	隔离耐压: 2500VDC	隔离耐压: 3500VDC	耐压能力提升		
ESD 能力	IEC/EN 61000-4-2 Contact ±4KV	IEC/EN 61000-4-2 Contact ±6KV	等级提升		
通讯能力	不高于 1Mbps	CAN 收发芯片最高 5Mbps RS-485 收发芯片高达 10Mbps	大幅提升		
封装	DIP 封装	DFN 封装	升级为贴片封装,生产效率提升		

表1芯"优势总结

"芯"产品应用

SM 系列全隔离收发芯片适用于储能、光伏电力、工业控制、智能楼宇、 仪器仪表、新能源充电桩等领域。如下图 3,为某用户直流充电桩控制系统 方案,该系统主要由主控 ARM、显示液晶、计量表、读卡器、总线通信模 组等构成。其中主控与充电机、BMS 系统的 CAN 通信,使用了 SM1500 全 隔离 CAN 收发芯片;计量表及远程监控的数据交互则使用 SM4500 全隔离 RS-485 收发芯片。







【技术分享】 如何提高多路热电偶测量的通道一致性

原创 研发部 ZLG 致远电子 2023-05-16 11:40:03

多路热电偶测温容易遇到通道精度的一致性问题,主要原因是冷端的 温度不一致性。本文分析该问题,并推荐相应的电路解决方案。

热电偶冷端的实际位置

热电偶是一种温差元件,如下图 1,它的输出信号是两个线端之间的电 压差: a(TA1-TA2)-b(TB1-TB2)。由于热源处是一个点,TA1 和 TB1 相等, 可记为 TH。若室温处的两个线端 A2 和 B2 的温度 TA2 和 TB2 也相等,记 为 TC。则热电偶两个线端之间的电压差,简化为 (a-b)*(TH-TC),对于特定 的两种金属,可以进一步简化为 VTC= α (TH-TC)。



如果热电偶两个线端温度不相等,则以上简化不成立,将出现额外的 偏差。因此,热电偶的两个线端需要保持在同一温度,保持等温。在等温环境, 两个线端与第三种金属(例如:铜质接线端子的螺丝)相连接位置才是实 际冷端的所在位置,而不是接线端子在 PCB 上的焊盘位置。

多通道时冷端温度的一致性问题

接线端子的螺丝或簧片通常处在 PCB 表面之外的空气中,当多个通道时,由于空气流动和附近热源的影响,就会难以保证各个接线端子的螺丝或簧片处于等温状态,使得各通道的冷端环境温度存在差异。常规的热电偶测量电路,冷端测温元件放在单个通道接线端子附近(如下图2)会顾及不到各通道的冷端温度差异,导致各通道的测量精度出现偏差。



冷端温度的一致性是可以通过结构设计改善,例如:使用导热良好的 金属块与各通道冷端位置靠近,使得各通道的冷端温度接近于相等;设计 密封的接线盒,隔开热源的热传导,使得内部温度保持均匀相等。但是这 些方式不但增加了结构件设计及装配,还增加设计复杂度和成本。

推荐的测量方案

在不增加结构件的情况下,我们可以通过增加冷端测温元件的数量来 提高各通道测量精度的一致性。推荐 ZLG 致远电子 ZAM6218A 八通道热电 偶测温模块,内部集成 24 位 ADC、信号调理电路、以及数据处理代码, 直接输出以 [™]C[™]为单位的温度数据。其冷端测量采用测温芯片,提供两路 I2C 和冷端芯片通讯,提供两种冷端芯片可选,最多能读取八个冷端芯片的 数据,如下图 3。ZAM6218A 的冷端测温芯片可根据实际应用需求与热电偶 测温通道组合配置,例如每个热电偶测温通道可配置一颗冷端测温芯片, 也可两个、四个、六个、八个热电偶测温通道配置一颗冷端测温芯片。



ZAM6218A 电路部分测量精度 0.02%±0.1℃,搭配可选的冷端芯片, 可实现 0.2℃以内的测量通道精度一致性。ZAM6218A 还提供了评估套件, 如下图 4。通过该评估套件,基于 ZAM6218A 模块可以快速搭建出高通道 精度一致性的热电偶测量电路。







ZLG致远电子官方微信



更多详情请访问 欢迎拨打全国服务热线 www.zlg.cn

400-888-4005