



2024/4 第4期 月刊



ZM82 系列国产 ZigBee 模组

低功耗、高性能,工业级品质全新上市



ZM82 系列模组是广州致远电子股份有限公司开发的一款低功耗、高性能国产 ZigBee 模组。产品采用星型组网模式,支持串口透传、AT 指令、空中升级、 定时休眠唤醒等功能。工业级性能设计,可广泛应用于工业控制、灯控照明、智能门锁、智慧楼宇等场景。

选型表

项目	ZM82P0S22E	ZM82P0S22P	ZM82P2S22E	ZM82P2S22P	ZM82P2S22S
组网协议	FastZigBee	FastZigBee	FastZigBee	FastZigBee	FastZigBee
工作频段	2402~2480MHz	2402~2480MHz	2402~2480MHz	2402~2480MHz	2402~2480MHz
发射功率	12dBm	12dBm	21dBm	21dBm	21dBm
	-98dBm	-98dBm	-102dBm	-102dBm	-102dBm
天线形式	IPEX 接口	PCB 天线	IPEX 接口	PCB 天线	邮票孔接口
通信接口	UART	UART	UART	UART	UART
工作温度	-40~+85°C	-40~+85°C	-40~+85°C	-40~+85°C	-40~+85℃
产品尺寸	17.0×12.0×2.1	17.0×12.0×2.1	19.0×13.5×2.3	19.0×13.5×2.3	19.0×13.5×2.3
包装类型	卷带	卷带	卷带	卷带	卷带
最小包装数量	1800	1800	850	850	850





致远电子官方微信



技术平台

EsDA 平台

【技术分享】AWTK 开源串口屏开发 (16)——提供 MODBUS 服务 ・・・・・・・・・・・・・・・・・・・・	04
【技术分享】AWTK 开源串口屏开发 (17)—— 通过 MODBUS 访问数组数据 ······	06
【AWTK 使用经验】如何设计立体电池进度条? · · · · · · · · · · · · · · · · · · ·	08
【AWTK 使用经验】加载和释放外部图片 ······	09
ZWS 云平台	
【产品应用】ZWS 云平台应用 (4)- 设备数据上报 · · · · · · · · · · · · · · · · · · ·	11
【产品应用】EM 储能网关 &ZWS 智慧储能云应用 (2) — 建模介绍 · · · · · · · · · · · · · · · · · · ·	12

边缘计算

工控板 / 工控机

【技术分享】探秘 TSN:如何让实时网络通信变得既快速又可靠? · · · · · · · · · · · · · · · · · · ·	13
【技术分享】EPCM3568A-LI 如何实现 WiFi 冲浪和热点共享 · · · · · · · · · · · · · · · · · · ·	15
行业控制器	
【技术分享】EtherCAT 数据帧格式和寻址方式简介 · · · · · · · · · · · · · · · · · · ·	18
【插针机 HMI 开发】用 AWTK 开发人机界面 ······	20
【插针机 PLC 开发】用 AWBlock 开发运动逻辑 · · · · · · · · · · · · · · · · · · ·	22
【产品应用】ZMC900E 控制器之 ROS2 环境安装指南 · · · · · · · · · · · · · · · · · · ·	25

互联互通

CAN-bus 总线

【技术分享】	CAN 会收到错误的数据吗?	28
【技术分享】	CAN 报文为什么会发送失败? · · · · · · · · · · · · · · · · · · ·	30
【技术分享】	一文读懂 CAN 控制器错误处理的原理·····	32
【产品应用】	飞行汽车专用的 CAN(FD) 数据记录终端 · · · · · · · · · · · · · · · · · · ·	34
无线通讯		
【新品发布】	新一代高性能小巧型 ZigBee 终端采集器全新上市 · · · · · · · · · · · · · · · · · · ·	35
【新品发布】	低功耗、高性能国产 ZigBee 模组 ZM82 系列全新上市 ·····	38
【产品详解】	新一代工业智能利器 ——GXF222-4G 工业数采边缘网关 ·····	41
接口与协议	议转换	
【解决方案】	电机控制领域如何实现 CANopen 设备接入 EtherCAT 主站控制器 · · · · · · · ·	43

致远微电子芯片

【新品发布】4合1,	300mA 带 Watchdog 低压差线性稳压器·····	44
ENTREPS OF A THE PARTY		

技术平台▼

【技术分享】AWTK 开源串口屏开发(16) 提供 MODBUS 服务

ZLG 致远电子 2024-04-12 11:36:51

AWTK 串口屏内置 MODBUS 从站模型,不用编写一行代码即可让串口 屏提供 MODBUS 服务, 让远程设备通过 MODBUS 协议访问串口屏上 的数据。

1. 功能

让串口屏提供 Modbus 服务:

界面可以修改 input_register 中的数据, Modbus 客户端可以读取 input_register 的数据。

界面可以显示 registers 中的数据,Modbus 客户端可以修改 registers 的数据。



2. 创建项目

从模板创建项目,将 hmi/template_app 拷贝 hmi/modbus_server_ registers 即可。(第一个项目最好不要放到其它目录,因为放到其它目录需要修改配 置文件中的路径,等熟悉之后再考虑放到其它目录。路径中也不要中文和空格,避免不 必要的麻烦。)

3. 制作界面

用 AWStudio 打开上面 modbus_server_registers 目录下的 project. json 文件。里面有一个空的窗口,做出类似下面的界面。



4. 添加绑定规则

1 URL	
-------	--

4

绑定属性	绑定规则	说明
v-data:value	{url}	url 表示远程设备的 URL

4.2 unit id

绑定属性	绑定规则	说明
v-data:value	{unit_id}	设备 ID

4.3 当前湿度

绑定属性	绑定规则	说明	
v-data:value	{ 当前湿度 }	当前湿度 在配置文件中定义,	参考后面的配置

4.4 当前温度

绑定属性	绑定规则	说明	
v-data:value	{当前温度}	当前温度 在配置文件中定义,	参考后面的配置

4.5 预设湿度

绑定属性	绑定规则	说明	
v-data:value	{ 预设湿度 }	预设湿度 在配置文件中定义,	参考后面的配置

4.6 预设温度

绑定属性	绑定规则	说明	
v-data:value	{ 预设温度 }	预设温度 在配置文件中定义,参考后面的配置	

4.7 窗口模型

指定窗口的模型为:

绑定属性	绑定规则	说明	
v-model	modbus_server	配置文件名为 modbus.json	

5. 配置文件

{

配置文件路径 design/default/data/modbus.json

"url": "tcp://localhost:502",
"channels": [
{
"name": "bits",
"writable": true,
"start": O,
"length": 100
},
{
"name": "input_bits",
"start": O,
"length": 200

},
{
"name": "registers",
"writable": true,
"start": 0,
"length": 300
},
{
"name": "input_registers",
"start": 0,
"length": 400
}
],
"variables": {
" 当前温度 ": "input_registers.word[0]",
" 当前湿度 ": "input_registers.word[1]",
" 预设温度 ": "registers.word[0]",
" 预设湿度 ": "registers.word[1]"
}
}



配置文件格式请参考:

https://gitee.com/zlgopen/awtk-hmi/blob/master/docs/modbus_ server_model.md

6. 描述需要持久化的数据

7. 编译运行

运行 bin 目录下的 demo 程序:

• • •		myapp		
	URL		tep://localhost:502	
	UNIT ID		255	
1	当前温度	22		
	当前湿度	33		
預设溫度(名	9户需发送请求修改)		56	
预设湿度(答	5户辈发送请求修改)		8	
		Quit		

8.注意

本项目并没有编写界面相关的代码,AWStudio 在 src/pages 目录下生成了一些代码框架,这些代码并没有用到,可以删除也可以不用管它,但 是不能加入编译。

完整示例请参考:

https://gitee.com/zlgopen/awtk-hmi/tree/master/hmi/demo_ modbus_server_registers 技术平台 ▼

【技术分享】AWTK 开源串口屏开发(17) —— 通过 MODBUS 访问数组数据

ZLG 致远电子 2024-04-26 11:41:12

在 AWTK 串口屏中,内置 MODBUS 客户端通道模型,不用编写代码 即可实现在 ListView 中显示远程设备上的数组数据。

4.2 序数

绑定属性	绑定规则	说明
v-data:value	{index}	index 特指序数。

4.3 告警级别

前面提到告警级别是正数,可以通过 item.level 来获取它。它的意义对 应为:调试 (0);信息 (1);警告 (2);错误 (3),我们需要用 one_of 函数将它转 换为对应的字符串。

绑定属性	绑定规则	说明
v-data:value	{one_of('调试; 信息; 警告; 错误' , item.level)}	这里的 one_of 的功能是从指定的字符串数 组中取出对应的子串。

4.4 时间

时间是整数(秒数),可以通过 item.time 来获取。

绑定属性	绑定规则	说明
v-data:value	{date_time_ format(item.time, 'Y-M-D h:m:s')}	需要用 date_time_format 将 epoch 时间
		转换成人类可读的时间。

4.5 告警信息

告警信息是一个字符串,可以通过 item.message 来获取。

绑定属性	绑定规则	说明
v-data:value	{item.message}	

4.6 URL

绑定属性	绑定规则	说明
v-data:value	{url}	url 表示远程设备的 URL

4.7 连接状态

绑定属性	绑定规则	说明
v-data:value	{connected ? 'connected' : 'no connection' }	connected 表示当前是否连接

4.8 窗口模型

指定窗口的模型为:

绑定属性	绑定规则	说明
v-model	modbus_ client(name=modbus_ demo)	配置文件名对应于 modbus_demo.json
modbus client 是内置的模型 name 是配置文件的名称 不需要加后缀		

1. 功能

不用编写代码,实现对远程设备上数组数据的显示。

2. 创建项目

从模板创建项目,将hmi/template_app 拷贝hmi/modbus_client_ channel_input_registers 即可。(第一个项目最好不要放到其它目录,因为放到 其它目录需要修改配置文件中的路径,等熟悉之后再考虑放到其它目录。路径中也不要 中文和空格,避免不必要的麻烦。)

3. 制作界面

用 AWStudio 打开上面 modbus_client_channel_input_registers 目录 下的 project.json 文件。里面有一个空的窗口,做出类似下面的界面。



4. 添加绑定规则

第一次用到列表视图,有几点需要特别说明一下: 列表视图中的滚动视图需要指定 v-for-items 属性。

绑定属性	绑定规则	说明
v-for-items	true	它保证其下的列表项,会根据数据自动生成

4.1 几个特殊的变量

- index 特指序数。
- item 特指当前的数据。比如在这里 'item.level' 表示告警级 别, 'item.time' 表示时间, 'item.device' 表示设备, 'item. message' 表示告警信息。
- selected_index 表示当前选中的序数(可在列表视图之外绑定)。
- items 表示当前列表视图中的数据个数(可在列表视图之外绑定)。

4.9 视图模型

为了使用 channel 的模型,还需要在窗口的下一级控件指定 modbus_ client_channel 模型。

绑定属性	绑定规则	说明
v-model	modbus_client_ channel(name=modbus	配置文件名对应于 modbus_demo_
Viniodet	demo_input_register)	input_register.json

modbus_client_channel 是内置的模型,name 是配置文件的名称,不需要加后缀。

5. 配置文件

配置文件路径 design/default/data/modbus_demo.json

```
{
    "url":"tcp://localhost:502",
    "channels" : [
    {
        "update_interval" : 5000,
        "name" : "read_input_registers",
        "access_type" : 4,
        "read" : {
            "offset" : 0,
            "length" : 3200
    }
    }
}
```

在上面的配置中,定义了通道 read_input_registers,长度为 3200 个 registers,共 6400 字节。

配置文件路径 design/default/data/modbus_demo_input_register. json

{
"channel":"read_input_registers",
"items": 100,
"variables": {
"level": "byte[0].uint32",
"time": "byte[4].uint32",
"message": "byte[8].str[56]"
}
}

在上面的配置中,将通道 read_input_registers 中的数据,分成 100 等份,也就是看作 100 元素的数组,每个元素 64 字节。这 64 字节的数据为 三个成员:

- level:4字节,无符号整数;
- time:4字节,无符号整数;
- message: 56 字节,字符串。

6. 描述需要持久化的数据

无

7.编译运行

运行 bin 目录下的 demo 程序:

		myapp		•	
0 / 100	tcp://localhost:502 connected		1 时间 滿息	告營銀别	序数
		message0	1970-1-12 19:20:0	调试	0
		message1	1970-1-12 19:36:40	信息	1
		message2	1970-1-12 19:53:20	警告	2
		message3	1970-1-12 20:10:0	调试	3
		message4	1970-1-12 20:26:40	信息	4
		message5	1970-1-12 20:43:20	警告	5
		message6	1970-1-12 21:0:0	调试	6
		message7	1970-1-12 21:16:40	信息	7
		message8	1970-1-12:21:33:20	警告	8
		message9	1970-1-12 21:50:0	调试	9
	退出 111		更新 🔖		
	:EH	mossage®	1970 1 12 21:33:20 1970-1-12 21:50:0 更新 》	響告	9

点击 Update 按钮,可以手工刷新数据

8.注意

- 本项目并没有编写界面相关的代码,AWStudio 在 src/pages 目录下 生成了一些代码框架,这些代码并没有用到,可以删除也可以不用 管它,但是不能加入编译。
- 完整示例请参考:demo_modbus_client_channel_input_ registers。
- 使用 TCP 协议时,如果使用 modbus slave 工具 配套测试,需要将 unit id 设置为 255。



【AWTK使用经验】 如何设计立体电池进度条?

ZLG 致远电子 2024-04-17 11:40:36

AWTK 是基于 C 语言开发的跨平台 GUI 框架。《AWTK 使用经验》系 列文章将介绍开发 AWTK 过程中一些常见问题与解决方案,例如:如 何加载外部资源?如何设计自定义进度条?这些都会在系列文章进行 解答。

如何设计立体电池进度条

在 AWTK 提供的进度条控件默认样式是比较简单的平面进度条,而在 实际开发过程中可能用到需要特殊样式的进度条,比如不规则进度条、分 段式进度条以及立体进度条等。本章节将以立体电池进度条为例子介绍如 何开发其它样式的进度条。



图1 电池进度条效果图

对于上面提到的特殊样式进度条,大多可以直接在 progress_bar 控件 中使用前景图片和后景图片结合的方式来实现。

首先要准备一下图片素材,一般准备两张图片素材,一张是进度条值为0的图片,另一张是进度条值为100的图片。注意两张图片尺寸需要一样, 图片尺寸会直接影响进度条控件的大小。



图2 进度条值为0和100的图片素材

在 AWTK Designer 中拖拽出一个进度条 progress_bar 控件,将它的背 景颜色 bg_color、前景 fg_color 设置为透明,并且将进度条控件的宽高设 置为图片的宽高。

•	控件编辑器		
请	输入搜索的属性名称		Q +
	text_align_v	middle	✓ Ĵ ^X _{All}
	bg_color	🖊 #00000000	fx
	bg_image	<无>	~ <i>f</i> X
	bg_image_draw_type		~ <i>f</i> X
	fg_color	📈 #00000000	fx
	fg_image	<无>	~ <i>f</i> X

图3 设置progress_bar控件前背景颜色

接着是设置 progress_bar 控件的前背景图片:将进度条值为 0 的图片 设置成背景图片 bg_image;将进度条值为 100 的图片设置成前景图片 fg_ image。

同时,设置背景图片显示方式 bg_image_draw_type 与前景图片显示 方式 fg_image_draw_type 都为 default 显示方式。最终得出的控件效果图 如下:

			- hiefiere_nen		
			▼ 控件编辑器		
1			请输入搜索的属性名称		a +
/			text_align_h	center	~ <i>J</i> .â.,
		and the second se	text_align_v	middle	~ <i>f</i> X
			bg_color	Z #00000000	f×
			bg_image	bat_bg	→ fx
	-		bg_image_draw_type	default	✓ fx
·	_	V I	fg_color	Z #00000000	fx
d	u	ь	fg_image	bat_fg	✓ fx
			fg_image_draw_type	default	→ fx

图4 progress_bar效果图

测试进度条效果

在完成 progress_bar 控件的设置之后,接下来可以测试一下它的效果。 选中 progress_bar 控件,并为其添加循环播放的值动画,即可看到进度条 实际播放的效果。



图5 progress_bar动画运行效果图

其它样式的进度条如分段式进度条也是可以用类似的办法实现,准备 两张进度条值为0与值为100的图片,然后设置到progress_bar控件样式中, 最后模拟运行查看效果即可。



【AWTK使用经验】 加载和释放外部图片

ZLG 致远电子 2024-04-24 11:37:45

AWTK 是基于 C 语言开发的跨平台 GUI 框架。《AWTK 使用经验》系 列文章将介绍开发 AWTK 过程中一些常见问题与解决方案,例如:如 何加载外部资源?如何设计自定义进度条?这些都会在系列文章进行 解答。

加载指定路径的图片资源

AWTK 默认加载资源方式有两种,一种是将资源打包到程序中放入 Flash,另一种是放在 res 资源目录中。假设现在需要实现一个图片浏览器 并移植到 ZTP800 示教器 设备上,该图片浏览器需要加载 U 盘图片,此时 就要用到 AWTK 加载外部资源的功能来实现该需求。下面将分为有文件系 统和无文件系统两种情况来介绍此功能。



图1 ZTP800示教器加载U盘图片效果

1. 有文件系统

在有文件系统时若想加载非AWTK默认目录结构的图片资源,可以使用: file:// + 图片所在路径的形式来加载图片,并且可以使用 C 代码或者 xml 形 式来加载。

• 1.1 C 代码方式:

widget_t* image = widget_lookup(win, "image", TRUE); image_set_image(image, "file:///media/sda1/AWTK.png");

• 1.2 XML 方式:

<window name="home_page ">

<image name="image" x="0" y="0" w="50" h="50" draw_ type="default" image="file:///media/sda1/AWTK.png"/>

</window>

2. 无文件系统

若是在没有文件系统的设备中加载外部图片,可以先读取图片文件数据,再将数据作为参数传给 assets_manager_add_data()函数。该函数会将数据添加到 AWTK资源管理器中,添加后就可以通过文件名的形式来显示图片了。

下面是一段 STM32 平台加载 SD 卡图片的示例代码:

uint8_t read_buffer[512];

int sdcard_status = HAL_SD_ReadBlocks(&sd_handle, (uint8_ t*) read_buffer, 0, 1, 0xffff);

if (sdcard_status == HAL_OK) {

assets_manager_add_data(assets_ manager(),"AWTK",ASSET_TYPE_IMAGE,ASSET_TYPE_IMAGE_ PNG, read_buffer, size);

image_set_image(image, "AWTK");

}

释放图片资源并重新加载

假设目前实现了图片浏览器并移植到了 ZTP800 示教器上,但是需要图 片浏览器加载的图片名称固定不变,而本地图片数据会经常发生改变,这 时候就需要在 AWTK 重新加载并更新图片到画面中。

AWTK 在加载一张图片时,会先将图片缓存到 assets_manager 资源管 理器当中,接着再解码放到 image_manager 图片管理器,最后显示的图片 来自图片管理器解码好的图片,因此对于上面的需求就要手动卸载与重新 加载图片缓存。下图为 ZTP800 示教器上的示例程序重新加载图片前后效果 图:



图2示例程序的初始图





图2 示例程序的初始图

1. 卸载图片缓存

假设在示例程序中已经加载并显示了一张图片,并且此时本地图片文 件数据发生了改变,可以按照以下步骤卸载图片管理器和资源管理器上的 图片缓存。

#define IMAGE_NAME "/media/sda1/AWTK.png" // 图片名默认 为 assets_manager_load_file 加载的路径

static ret_t on_unload_button_click(void* ctx, event_ t* e){//点击卸载图片按钮卸载图片缓存

bitmap_t bitmap = {0}; widget_t* win = WIDGET(ctx); widget_t* image = widget_lookup(win, "image", TRUE);

// 卸载图片管理器缓存

image_manager_get_bitmap(image_manager(), IMAGE_ NAME, &bitmap);

image_manager_unload_bitmap(image_ manager(), &bitmap);

// 卸载资源管理器缓存

assets_manager_clear_cache_ex(assets_manager(), ASSET_ TYPE_IMAGE, IMAGE_NAME);

widget_invalidate(image, NULL);

```
return RET_OK;
```

}

上面代码中 image_manager_unload_bitmap 是用于卸载图片在图片 管理器的缓存卸载缓存成功后会输出 "unload image xxx"的 debug 信息; assets_manager_clear_cache_ex 是用于卸载资源管理器的该图片缓存。

2. 重新加载图片缓存

卸载图片缓存后,可以通过以下步骤再次加载图片到资源管理器中:

static ret_t on_load_button_click(void* ctx, event_t* e) { // 点 击加载图片按钮重新加载图片缓存 widget_t* win = WIDGET(ctx); widget_t* image = widget_lookup(win, "image", TRUE);

// 将新的图片数据添加到资源管理器缓存中 asset_info_t* img = assets_manager_load_file(assets_

manager(), ASSET_TYPE_IMAGE, IMAGE_NAME); assets_manager_add(assets_manager(), img);

image_set_image(image, IMAGE_NAME);

widget_invalidate(image, NULL);

return RET_OK;

}

上面代码调用 assets_manager_load_file 与 assets_manager_add 重 新加载了一次图片数据到 AWTK 资源管理器的缓存,此时加载的数据是新 的图片数据。

后续在调用 image_set_image 以及 widget_invalidate 刷新图片控件 时会自动将位于资源管理器的缓存解码并放到图片管理器当中,最后显示 新的图片数据。



【产品应用】 ZWS云平台应用(4)-设备数据上报

ZLG 致远电子 2024-04-10 11:43:08

智能设备连接到物联网,会产生大量的数据,可以说:物联网最大的 价值,其实是数据,那么设备数据是如何上报到云平台的呢?

随着物联网的不断发展,在生产生活中,可以看到很多智能设备,比 如智能传感器、智能工业设备、共享单车、智能手表等。这些设备在工作 运行时,会产生大量的数据,并传输给物联网云平台,进行智能化管理分析。 那么,设备数据是如何上报到云平台呢?

发布/订阅

物联网通信常用的是 MQTT 协议,它是基于发布 - 订阅模式的消息传 输协议,能够实现传感器、控制器和其他设备之间的高效通信。

MQTT发布/订阅过程,一个设备向"test"主题发送数据,每个订阅"test" 主题的设备或平台,就能接收到来自"test"主题的信息,像我们订阅公众 号一样。



ZWS 物联网云平台预定义了 data 主题和 raw 主题,用于上报设备数据。

ZWS 预置 Topic	说明	发布者	订阅者
data 主题	上报 data 数据,键值对数据	设备	云平台
raw 主题	上报 raw 数据,透传数据	设备	云平台

上报data数据

智能设备可以向"data"主题发送设备数据,data 主题支持 key/value 键值对的方式上报,ZWS 云平台接收到数据后,无需解析可直接显示。 发布 data 主题如下:

/d2s/\${owner}/\${devtype}/\${devid}/data

data 主题的消息内容格式如下:

\0key\0value\key\0\value\0key\0value\0

上报raw数据

智能设备可以向"raw"主题发送设备数据,raw 主题支持纯透传的方 式上报,可以上报任意格式的二进制数据。ZWS 云平台接收到数据后,需 根据配置的解析脚本,将透传数据解析后才能看到更易读的 JSON 数据。 发布 raw 主题如下:

/d2s/\${owner}/\${devtype}/\${devid}/raw

raw主题是透传设备数据,消息内容格式可以是任意格式的二进制数据。



ZLG微文摘

技术平台 ▼

【产品应用】EM储能网关&ZWS智慧储能云 应用(2) — 建模介绍

ZLG 致远电子 2024-04-29 11:37:42

ZWS 智慧储能云平台,能让储能系统快速完成数据对接,自定义配置 电站差异化,快速实现上云服务,智能化运维管理。

工商储能的设备厂家,其中储能系统,因不同的项目需求,场景需求, 有多形式储能产品,需在统一的储能云平台上进行管理,其中如何快速接 入云平台与在云平台展示差异性,比较难以处理。

ZWS 智慧储能云平台是为储能系统提供云端数字化运维与管控的行业 系统平台,配合致远电子的 EM 系列储能边缘网关,让储能系统以建模的形 式快速接入,通过自定义拓展物模型、配置等形式,达到储能设备快速接 入云平台并差异化展示,能快速拥有相应储能系统适配的储能云平台。

物模型接入

 1、针对储能系统设备,云端拥有标准的物模型可供接入,其中物模型 分为了"数据、状态、命令、日志"等。设备端通过开发云端的 SDK 包, 将相应的数据进行对接。其中针对核心数据、命令等已内置,用于终端用 户的展示。管理员可选定部分核心数据是否进行展示。

28 AB 2803	27.0	8338	712						ANA A
atistics data		19-13	986	8769	M2	2007	1012	9012	18/19
atistics_bees		1	Edischg_total	東計放現量	double		kWh	原计按电量	85 109 1.0
51 281	84 10	2	Echg_total	累计元电量	double		kWh	家计方观堂	RH 101 1.0
152 (2)82	P6 101	3	active_power	磺胺胺物物率	double		kW	编程和功功率	44 800 FB
rs3 1080	- 446 227	4	reactive_power	100003388	double		KVIør	9600398	84 101 1.6
284	44 10	5	grid_power	电局有2525章	double		kW	电局有2523重	418 101 1.18
55 085	44 10	6	grid_Rpower	电风花动动车	double		kMar	电用花动动率	MR 101 1.0
196 2084	84.00	7	total power	第 计位数用电量	double	8	kWb	第 计公司用电量	SH 101 18
157 1287	94 221		load_Rpower	电影影动动革	double		kldar	负载无动功率	84 IN 18
198 1288	64 10	2	dump_energy	2000CE	double		KWD	SNOR	88 10 18
1951	44.00	10	grid_energy	東北市民政主要	double		kWb.	東井市河市市里	81 10 18
ms2 (5/82	HE 201	<	1 2 2 >	月25条 10条件 -					
Lan Lan	84 22	-							
ma4									

2、标准物模型,若无法全量满足上云需求。云端支持对于物模型进行 自定义拓展,通过添加数据字段、命令,拓展运维所需的全量 PCS、BMS 数据点与命令,实现对于设备进行远程运维与监控。

3、以数据点为例,可以在已内置的标准 PCS、BMS 等数据组中添加自 定义拓展的数据点。

	Industry_E5_fm 10080			R / NARE / REH / OKR
E 2322	10 10 0+ D0 KM	1		
24.45	Statig Br. Bar 22.4	添加数据学校	×	BELFER BR
24928	statistics_data and ano 1.00 Tat		81 84	84
1022	statistications and ann 1-15 T-85	- 9808 0/100	226242328 KW	854 800 2.0 FØ
H Garn -	ICAL MOR DIA 1.0 TO	A2 H · ·	23035034 KW	848 877 2.0 7.0
ia asses	PCS2 #848 000 2.08 Tak	MI 0/20	2305802# KM	944 811 2.0 7.0
0 neze	pcil #88 899 128 748		2377488 -	88 80 20 10
() NATE -	0014 0018 000 1.00 710			
A 10-21 -	pens mail and 2.88 Yes	B/2 0750	the second se	
() ANOM -	2106 NR 070 128 78	BRADE V 20 REAR	· · · · · · · · · · · · · · · · · · ·	
C MODE -	DEAT AND AN TH		. V EBRA	854 DI1 25 75
© NARE -	PCR RER DIN 145 TH	ka ka	PD63 Y	WH 107 28 78
	· • 2 - 0 · m+0.00		CHINES V	404 HTT 2.0 TO
		10 current,A double	A ROOM A	864 871 £8 748
		1 2 2 HINR 108.00		
	adioacala			STREET BA. BE
		CAUSE / HERDER / BOOMES P. BURGE - HIS BOO	ans and completen	

4、支持拓展额外数据组,对于想同时观察、对比的数据点。可以自定 义拓展为同一数据组中,在数据展示页面,可同时进行查看与曲线可视化 对比。



关联产品



EM 系列储能边缘智能网关是 ZLG 致远电子专为新能源储能系统设计的 一款高性能、多接口通讯管理设备,可在储能系统应用中作为边缘 EMS(能 源管理系统)总控、通讯管理机、规约转换器或 BAU(电池管理总控)使用。 该系列产品集成丰富的外设接口,支持各类 BMS、PCS、空调、电表、屏 显等设备的通讯传输,且软件上支持 RT-Linux、Ubuntu 等操作系统,支持 IEC-61850/IEC-104/EtherCAT 等专用协议,可广泛满足各类储能系统的本 地能源管理应用需求。



【技术分享】探秘TSN:如何让实时网络通信 变得既快速又可靠?

ZLG 致远电子 2024-04-09 11:43:40

TSN,全称 Time-Sensitive Networking,中文名称为时间敏感型网络, 它是目前国际产业界正在积极推动的全新工业通信技术,也是下一代 网络技术的核心。

TSN技术简介

时间敏感网络(Time-Sensitive Networking, TSN)是在 IEEE 802.1 标准以太网框架下制定的新一代标准以太网技术,运行在 OSI 模型中的数据 链路层,如图 1 所示, TSN 技术源于标准以太网的队列传输机制,网络帧 格式也是采用包含 VLAN 标签的标准以太网帧格式。

7 应用层		
6 表示层		
5 会话层		
4 传输层		
3 网络层		
 2 数据链路层	TSN	IEEE802.1
 1物理层		

图 1 TSN 的 IEEE 802.1 框架

在标准层面,IEEE 802.1 时间敏感网络工作组发布了一套用于工业和汽车网络的开放实时以太网标准 TSN,包括 IEEE 802.1Qbv、IEEE 802.1Qcc、 IEEE 802.1AS 等标准,如图 2 所示,旨在降低布线成本,打破信息孤岛,同时在通信过程中传输不同时间敏感性的数据。



借助 TSN 技术的优势,可将 TSN 网络交换机(即支持 TSN 技术的交换 机)部署到工业互联网网络环境,实现智能制造、5G 融合、无人驾驶等应用, 如图 3 所示。



TSN时间同步技术应用实践

当前时间敏感网络技术已经成为包括芯片厂商、通信设备厂商、自动 化厂商、相关行业组织以及各类研究机构在内的产业链各个组成环节关注的 热点,我司紧跟科技创新发展的新趋势,推出了搭载 TSN 技术的 M62xx-T 系列核心板。



图 4 M62xx-T 系列核心板

M62xx-T 系列核心板板载处理器为德州仪器(TI) 推出的支持多协议 千兆位 (Gb) 时间敏感网络 (TSN) 处理器系列。这款高度集成的新型 Sitara AM6x 处理器可提供工业级可靠性,有四核和双核 Arm Cortex-A53 两种处 理器型号可选: M6234 和 M6232,可满足工厂自动化、电机驱动和电网基 础设施等应用中快速增长的工业 4.0 需求。

Sitara AM6x 处理器通过子系统支持 TSN 标准和其它工业协议下的千兆 位传输速率,可在单一网络上融合以太网和实时数据传输。这一特性对于 工业 4.0 应用中的实时通信至关重要,可在工厂中实现可重新配置软件的网 络物理系统。

在通信网络中,许多业务的正常运行都要求网络时间同步。时间同步 包括频率和相位两个方面的同步,通过时间同步可以使得整个网络各设备 之间的频率和相位差保持在合理的误差范围内。

本文基于 M6234-T 核心板,对其 TSN 时间同步功能进行了全方位的测 试,测试结果可为用户进行 TSN 技术选型提供参考。

ZLGi微文摘 边缘计算▼

1. 测试环境

内核版本,如图5所示:

root@am62xx:x# uname -a Linux am62xx 5.10.168-rt83-gc1a1201911 #1 SMP PREEMPT_RT Wed Mar 27 05:57:17 UTC 2024 aarch64 GNU/Line 图5 内核版本

ptp4l版本,如图6所示:



图6 ptp4l版本

2. ptp4l命令,如图7所示



图7 ptp4l命令

3. 创建ptp配置文件,文件内容如图8所示



图8 ptp4l配置文件

4. Master侧配置,如图9所示



图9 Master侧配置方法

5. Slave侧配置

root@am62xx:~# p	otp4l-E		-l eth1 -s		6 - m	-q-p/	dev/p1	tp0 -† ptp	.ctg	
ptp4l[98.052]: s	selected	d /dev/p	tp0 as PTP	clo	ock					
ptp4l[98.088]: p	port 1 ((eth1):	INITIALIZIN	IG 1	to LIS	STENING	on INI	T_COMPLET	E	
ptp41[98.088]: p	port 0 (/var/ru	n/ptp4l): 1	INI	TIALIZ	ZING to	LISTER	ING on IN	IT_COMPL	ETE.
ptp4l[98.088]: p	port 0 ((/var/ru	n/ptp4lro):	: 11	NITIAL	IZING t	0 LIS	TENING on	INIT_COM	IPLETE
ptp41[102.387]:	port 1	(eth1):	new foreig	jn r	master	641c10	.tffe	.28d5c5-1		
ptp4l[105.812]:	selecte	ed local	clock 3efo	:4d	.tffe.	eaebbe	as bes	st master		
ptp4l[106.387]:	selecte	ed best	master cloo	∶k (641c10).fffe.2	8d5c5			
ptp4l[106.387]:	port 1	(eth1):	LISTENING	to	UNCAL	IBRATED	on RS	S_SLAVE		
ptp4l[108.387]:	master	offset	-2148308082	2679	97 s0	freq		path dela		426
ptp4l[109.387]:	master	offset	-2148308082	2654	41 s1	treq	+256	path dela		426
ptp4l[110.387]:	master	offset	-3860	s 2	freq	-3604	path	delay	426	
ptp4l[110.388]:	port 1	(eth1):	UNCALIBRAT	TED	to SL	AVE on I	MASTER	R_CLOCK_SE	LECTED	
ptp4l[111.388]:	master	offset		s 2	freq	-911	path	delay	426	
ptp4l[112.388]:	master	offset	1184	s2	treq	+279	path	delay	407	
ptp4l[113.388]:	master	offset	1144	s 2	freq	+594	path	delay	426	
ptp4l[114.388]:	master	offset	813	s2	freq	+607	path	delay	426	
ptp4l[115.388]:	master	offset	448		freq	+486	path	delay	446	
ptp4l[116.389]:	master	offset	232	s2	freq	+404	path	delay	446	
ptp4l[117.389]:	master	offset	56		freq	+298	path	delay	484	
ptp4l[118.389]:	master	offset		s2	freq	+242	path	delay	521	
ptp4l[119.389]:	master	offset	10		freq	+264	path	delay	520	
ptp4l[120.391]:	master	offset	24		freq	+281	path	delay	520	
ptp4l[121.391]:	master	offset			freq	+287	path	delay	520	
ptp4l[122.391]:	master	offset			freq	+271	path	delay	520	
ptp4l[123.391]:	master	offset			freq	+262	path	delay	520	
ptp4l[124.391]:	master	offset			freq		path	delay	520	
ptp4l[125.391]:	master	offset			freq	+268	path	delay	519	
ptp4l[126.391]:	master	offset			freq	+271	path	delay	519	
ptp4l[127.392]:	master	offset			freq	+274	path	delay	518	
ptp4l[128.392]:	master	offset			freq	+275	path	delay	518	
ptp4l[129.392]:	master	offset			freq	+282	path	delay	517	
ptp4l[130.393]:	master	offset			freq	+272	path	delay	517	
ptp4l[131.393]:	master	offset		s2	freq	+256	path	delay	517	
ptp4l[132.393]:	master	offset			freq	+257	path	delay	516	

图10 ptp测试结果

图 10 的测试结果说明如下:

- Master offset,表示 ptp 协议中定义的主从端时间差(单位: ns);
- s0、s1、s2,指示器显示时钟伺服的不同状态:s0表示已解锁,s1 表示时钟步进,s2表示已锁定。如果伺服处于已锁定状态(s2),并 且 pi_offset_const 选项在配置文件中设置为负值,则时钟不会步进, 而只会缓慢调整;
- freq,表示时钟的频率调整(以十亿分率(ppb)为单位);
- path delay,表示从主时钟发送的同步消息的预计延迟(以纳秒为单位)。

由上可知 M6234-T 核心板的 TSN 时间同步性能参数中,单跳时延小于 525ns,抖动小于 20ns,时间同步精度 20ns。



【技术分享】EPCM3568A-LI如何实现WiFi 冲浪和热点共享

ZLG 致远电子 2024-04-16 11:44:41

在物联网时代,无线连接对智能设备至关重要。EPCM3568A-LI工控 机作为边缘计算网关,简化了设备联网流程,支持远程数据传输与智 能操作。本文将指导如何通过该工控机轻松实现WiFi连接和热点共享。

EPCM3568A-LI产品简介

EPCM3568A-LI 采用 RK3568 四核处理器, 主频高达 2.0GHz, 内置 1TOPs NPU, 作为边缘计算网关, EPCM3568A-LI 以其强大的计算能力, 轻 松高效处理数据,为用户带来更加便捷、智能的能源管理体验。



前期准备

1. 软件方面

- 安装好虚拟机的电脑(VMware Workstation 12 及以上最佳,电脑配置 越高越好);
- EPCM3568A-LI开发环境,下载地址:<u>manual.zlg.cn/web/#/269/10177</u>
- MobaXterm 虚拟终端。
- 2. 硬件方面
- EPCM3568A-LI 小型边缘计算网关;
- 网线1根/交换机1个;
- HDMI 显示屏一个。

WiFi无线网络配置

EPCM3568A-LI 接通电源,用网线和笔记本一起接入同一个交换机。待系统 启动后,使用 MobaXterm 的 SSH 登录 IP: 192.168.1.136,用户名: zlgmcu,密码: zlgmcu。



图1 SSH登录EPCM3568A-LI

然后,进入系统后,使用以下指令查看是否生成网络接口:

zlgmcu@	EPCM3568:~\$ sudo ifconfig
eth0: f	lags=4163 <up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast>
	inet 192.168.137.30 netmask 255.255.255.0 broadcast 192.168.137.255
	ether 6a:23:de:8c:59:35 txqueuelen 1000 (Ethernet)
	RX packets 100 bytes 10650 (10.4 K\B)
	RX errors 0 dropped 0 overruns 0 frame 0
	TX packets 8 bytes 894 (894.0 B)
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
	device interrupt 42
eth1: f	ags=4163 <up.broadcast.running.nulticast> mtu 1500</up.broadcast.running.nulticast>
	inet 192,168,1,136, netmask 255,255,255,0, broadcast 192,168,1,255
	ether 6e:23:de:8c:59:35_txqueuelen 1000 (Ethernet)
	RX packets 1809 bytes 692349 (676.1 KiB)
	RX errors 0 dropped 0 overruns 0 frame 0
	TX packets 829 bytes 97453 (95.1 KiB)
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
	device interrupt 44
101 610	-72 JUD LOODDACK DUNNITACA
	g^{3-3-5} (1) and g^{3-3}
	ingt
	loop txqueuelen 1000 (local Loopback)
	RX packets 8 bytes 480 (480.0 B)
	BX errors 0 dropped 0 overruns 0 frame 0
	TX packets 8 bytes 480 (480.0 B)
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Jan0.	flage=4009cHP_RECADCAST_MULTICASTmtu_1500
reario.	athar 34.6f.24.5f.f4.35 traveuelan 1000 (Ethernat)
	RY nackate 0 hytee 0 (0 0 R)
	BY errors A dropped 197 overrups A frame A
	TX nackets A hytes A (A A B)
	TX errors A dropped A overrups A carrier A collisions A
L	

图2网络配置信息

EPCM3568A-LI作为无线客户端

1. 扫描WiFi热点

首先,我们继续在 MobaXterm 终端执行如下命令,可查看开发板周围 可见的 WiFi 热点,如图 3:

sudo iw dev wlan0 scan | grep SSID

zlgmcu@EPCM35	68:~\$ sudo iw dev wlan0 scan grep SSID
SSID:	TP-LINK C205
SSID:	HUAWEI B311 DD72
SSID:	OpenWrt
	* UTF-8 SSID
SSID:	Zlg-Office
SSID:	орро
	SID List
SSID:	Zlamcu
SSID:	Zlg-Office
SSTD	71g-Market
SSTD:	SURFboard-2898

图3 扫描出来的热点ID

2. 连接WiFi热点

然后,我们继续在 MobaXterm 终端编辑 WiFi 配置文件 /etc/wpa_supplicant.conf,输入以下命令:

sudo vim /etc/wpa_supplicant.conf

增加需要连接的热点信息。例如本例的连接热点为"oppo",密码 123456780,/etc/wpa_supplicant.conf文件内容如下所示:

ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
ssid="oppo"
psk="123456780"
}



ZLGi微文摘 边缘计算▼

然后,执行如下命令,连接 /etc/wpa_supplicant.conf 配置文件中描	若成功连接上"oppo"这个热点,则显示信息如图 7 所示。
还的 WFI 热思:	zigmecu@ervm3563:~\$ sudo wpa_cii -twian0 status bssid=3e:d9:es:65:89:6e freq_2437 csid=camp
<pre>sudo wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_</pre>	sta-oppo ld=0 mode=station nairvise_cinber_NONE
supplicant.conf -B	group_cipher=MONL group_cipher=MONL key_mgmt=MONE wna_state=CMMPIFTED
	p2_device_address=36:6f:24:5f:f4:a5 address=34:6f:24:5f:f4:a5 uuid=9fcdpfrcf.755.7549:8514_48:f667b08ea0
首次执行该命令可能会遇到如图 4 情况: Zigncouge/Lissosics; sudo wps_supplicant -UnitW211 -twintw -c/etc/wps_supplicant.cont -B	图7查看已连接的WiFi信息
sactorside: United team and support can citicate exists and seems to be in use - cannot override it is not sact any and the same of the same of the same of the same of the same failed to initialize control interface /var/num/ag supplicant; To may have another was usualized to recess already running on the file was the same of the same	
left bý an uncleam ternümltion of vog supplicant in which čase you will need to manually remove this file before starting wpg_supplicant again. n100211: deinit ifmamewikand disakled 11b rates=0	连接 WiFi 热点后,就可以配置 IP 地址了,使用动态获取 IP 地址,命 今如下:
图4 wpa_supplicant连接	- • • • • •
	sudo udhcpc -i wlan0
检查是否有其他 wpa_supplicant 进程正在运行: 运行以下命令来查	如果第一次执行此命今遇到如图 8 失败情况,按 Ctrl+c 失结束,然后
自たロ市共同 wpa_supplicant 如柱正立及门.	再次执行此命令,获取 IP 成功,如图 9:
ps aux grep wpa_supplicant	:lgmcu@EPCM3568:∼\$ sudo udhcpc -i wlan0 ıdhcpc: started, v1.30.1
	ıdhcpc: sending discover ıdhcpc: sending discover
typecupervs308:+9 ps and pype pp_30ptcumit 55 17:48 0:00 /sbin/wpa_supplicant -u -s -0 /run/wpa_supplicant 1 gmcu 223 0:0 0:1104 735 pts/0 5+ 17:50 0:00 grepcolor=auto wpa_supplicant 1 gmcu02F0458-i sudo kill 43	Idhcpc: sending discover
图5 查看使用wpa_supplicant的进程	图8 使用udhcpc自动获取IP地址失败
	zlgmcu@EPCM3568:∼\$ sudo udhcpc -i wlan0 µdhcpc: started. v1.30.1
	udhcpc: sending discover udhcpc: sending discover
查看到确实有其他 wpa_supplicant 进程,执行如下命令杀死进程:	udhcpc: sending select for 192.168.208.191 udhcpc: sending select for 192.168.208.191 udhcpc: lease of 192.168.208.191 obtained, lease time 3599
sudo kill 433	ip: RTNETLINK answers: File exists
	图9 使用udhcpc自动获取IP地址成功
再次执行:	接下来,输入以下命令,查看是否获取 IP 成功。
sudo wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_	sudo ifconfig
supplicant.conf -B	
	wland: flags=4163-4UP.BROADCAST, RUNNING, NULTICAST> mtu 1500
可以看到连接成功,如图 6。	inet 192.168.208.192 netmask 255.255.05 / broadcast 192.168.208.255 ether 34/672/457/4754 / sugueuelen 1000 (Cherrent) RX packets 4 bytes 832 (832.0 B) RX errors 0 droped 252 overruns 0 frame 0
z]amru8EPFN3568-≴ sudo wa supolicant -D180211 .iv]an0 -c/stc/wpa supolicant conf -R	TX packets 15 bytes 2490 (2.4 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Successfully initialized was supplicant	图10查看WiFi的IP地址
图6 wpa_supplicant连接WiFi成功	图 10 的红框即 WiFi 新获取的 ip 地址。
	连接上 WiFi 热点后便可测试外围通信了,如图 11 所示。
查看是否成功连接"oppo"这个 WiFi 热点,可使用 wpa_cli 工具,操 作命今如下·	sudo ping www.baidu.com
1 L M X YM I Y	
sudo wpa_cli -iwlan0 status	zlgmcu@EPCM3568:~\$ sudo ping www.baidu.com PING www.a.shifen.com (157.148.69.74) 56(84) bytes of data.
	64 bytes from 157.148.69.74 (157.148.69.74): icmp_seqr4_ttl=51 time=124 ms 64 bytes from 157.148.69.74 (157.148.69.74): icmp_seqr5_ttl=51 time=66.7 ms 64 bytes from 157.148.69.74 (157.148.69.74): icmp_seqr5_ttl=51 time=113 ms
岩成切连接上"oppo"这个热点,则显示信息如图7所示。	154 bytes from 157,148,697,44 (157,148,697,44); torm_seq=7 ttl=51 time=632, ms 64 bytes from 157,148,697,44 (157,148,697,44); torm_seq=8 ttl=51 time=141 ms 64 bytes from 157,148,697,44 (157,148,697,44); torm_seq=9 ttl=51 time=138, ms 64 bytes from 157,148,697,44 (157,148,697,44); torm_seq=10 ttl=51 time=108 ms
sudo wpa_cli -iwlan0 status	图11 机子ping百度地址

然后,我们用 hdmi 线将 EPCM3568A-LI 与显示屏对接,打开谷歌浏览器,输入 www.baidu.com,搜索"腾讯视频",就能看到如图 12,为 EPCM3568A-LI 浏览网页图片。



图12 EPCM3568A-LI浏览网页

模块作为热点

1. 搭建DHCP服务器

接下来,我们将 EPCM3568A-LI 作为 WiFi 热点,需要为每一个接入该 热点的终端(例如手机)分配 IP,路由等网络参数。而这些工作可以 通过搭建一个 DHCP 服务器来完成,默认只需要创建 udhcpd.conf 配 置文件,我们继续在 MobaXterm 终端执行如下命令(本例中配置文件 路径为 /opt/udhcpd.conf):

sudo udhcpd /opt/udhcpd.conf &

dhcpd.conf 配置文件的内容如下所示(注意:其中的网络参数可以根 据实际情况更改):

#the start and end of the IP lease block
start 192.168.5.10
end 192.168.5.254
#the interface that udhcpd will use
interface wlan0
option subnet 255.255.255.0
opt router 192.168.5.1
option domain local
option lease 864000 #10 days of seconds

2. 开启WiFi工作在AP模式(即作为热点)

然后我们继续在 MobaXterm 终端执行如下命令:

sudo ifconfig wlan0 up 192.168.5.1 sudo hostapd -B /opt/hostapd.conf

执行成功后如图 13 所示

lgmcu@EPCN3568:∽5 sudo hostapd =0 /opt/hostapd.conf onfiguration file: /opt/hostapd.conf sing interface wlan0 with hwaddr 34:6f:24:5f:f4:a5 and ssid "EPCM3568A-LI" Lan0: interface state UNINITIALIZED→ENABLED Lan0: AP-ENABLED

图13 使用hostapd配置热点信息

其中,/opt/hostapd.conf 配置文件内容如下所示,ssid 设置 WiFi 热 点的名称,使用 wpa2 加密方式,wpa_passphrase 设置密码。更多参 数可到 hostapd 主页查看。

interface=wlan0 driver=nl80211 ctrl_interface=/var/run/hostapd hw_mode=g channel=6 ssid=EPCM3568A-LI wpa=2 ieee80211n=1 wpa_passphrase=12345678 ignore_broadcast_ssid=0 wpa_key_mgmt=WPA-PSK rsn_pairwise=CCMP

使用手机搜索 WiFi 热点,可以看到"EPCM3568A-LI"热点。连接手机 到 EPCM3568A-LI 热点,密码为 12345678(注:密码定义在 hostapd. conf)。手机可以下载 ping 的软件,测试是否 ping 通开发板的 wlan0 (本 例 IP 地址为: 192.168.5.1)。

归纳总结

本节使用 EPCM3568A-LI 的板载无线网卡实现 WiFi 冲浪和热点共享。 首先要在 EPCM3568A-LI 上设置相应的网络参数,查找可连接的 WiFi 信息,再通过相应的配置文件修改,从而连入 WiFi。同时,板载的无 线网卡也能作为热点共享给其他手机、平板电脑、笔记本电脑等智能 产品。 边缘计算▼

【技术分享】 EtherCAT 数据帧格式和寻址方式简介

ZLG 致远电子 2024-04-08 11:50:42

EtherCAT 是一个高实时性,高速和高效率的工业以太网技术,数据根据自身独有的数据帧格式进行可靠传输,本文主要针对 EtherCAT 的数据帧格式和寻址方式进行简单描述。

EtherCAT 的数据帧格式

EtherCAT (Ethernet Control Automation Technology) 是由德国倍福 公司在 2003 年提出的工业以太网技术,具有很高的实时性。传统的以太 网通信中每一个节点都是按照接收,处理和转发这个三个流程进行,而 EtherCAT 则是同时传输和处理 EtherCAT 数据。

由于 EtherCAT 始终依托以太网进行数据的传输,因此和传统的以太网 数据帧格式大致相同。EtherCAT 在传统以太网数据帧格式作出了一些修改, 譬如在 EtherCAT 中,EtherType(帧类型)为 0x88A4, EtherType 是用于 指明应用于帧数据字段的协议,如 0x0806 为地址解析协议 ARP。

传统的以太网帧格式包含了以太网帧头,以太网数据和 FCS 帧校验, 而 EtherCAT 的数据帧格式跟传统以太网帧格式一样,只是对帧数据段进行 了细化,将数据字段分为帧头和 EtherCAT 数据,其中 EtherCAT 帧头包含 了数据帧长度,保留字。将 EtherCAT 数据认为若干个子报文,每一个子报 文又包含了子报文头,子报文的数据,WKC(工作计数器),具体以下图所示。



图1 EtherCAT数据帧格式

表1 EtherCAT数据帧格式说明

目的地址	接收方的 MAC 地址
目的地址	接收方的 MAC 地址
源地址	发送方的 MAC 地址
帧类型	0x88A4
FCS	帧校验序列
EtherCAT 头(数据长度)	EtherCAT 数据区长度
EtherCAT 头(类型)	1表示从站通信,其余保留

EtherCAT 子报文结构定义:

表2 EtherCAT子报文定义

命令	寻址方式
索引	帧编码
地址	从站地址
长度	报文数据区长度
R	保留位
М	后续报文标志
状态位	中断到来的标志
数据位	子报文数据结构,用户定义
WKC	工作计数器

其中子报文头的作用是确定此子报文由哪一个从站使用,什么操作命 令,处理多长的数据等,而工作计数器主要记录了报文被对应的从站操作 的类型以及次数,一般而言,主站在每次发送 EtherCAT 报文前都要预设一 个 WKC 值,将数据帧发送给从站。

EtherCAT的寻址方式

1. 网段寻址

直连模式: 主站设备通过于 EtherCAT 网段直接端口连接来对 MAC 地 址进行广播;

开放模式: 主站设备通过交换机与 EtherCAT 网段进行连接,网段可以 使用 EtherCAT 数据帧中的目的地址来做 MAC 地址。

2. 设备寻址

EtherCAT 数据帧中子报文头有 32 位地址,其中高 16 位是从站的设备地址,低 16 位是设备内部物理存储的地址,16 位从站设备地址可以寻 65535 个从站设备,而每一个设备地址最多可以有 64KB 的本地存储空间。

2.1 顺序寻址

从站的地址由物理连接的顺序决定,从站地址自动加1,一般在主站启 动阶段使用,主站为从站分配地址,无需从站设置。



2.2 配置寻址

该地址与从站物理连接顺序无关,通常有两种配置方式: 数据链路层启动阶段由主站配置给从站;

在上电初始化的时候,从站从自身的 EEPROM 的配置文件中读取出来。 主站利用顺序寻址的方式来读取从站设置的地址,掉电之后会丢失固定的 地址。



图3 配置寻址示意图

2.3 逻辑寻址

在逻辑寻址模式下,报文内的 32 位地址空间用来寻址,整个网段有 4GB 地址空间可以被寻址。逻辑寻址方式由 FMMU 来实现,ESC 芯片中的 FMMU 单元将从站本地的物理地址映射到网段内的逻辑地址。当从站收到 主站的报文时,会检查报文中的地址是否和 FMMU 中的地址是否相符,如 果相符,就会根据类型进行读写操作。



图4 逻辑寻址示意图

广州致远电子以 EtherCAT 工业以太网协议为向导,开发了一系列 EtherCAT 主站控制器和通讯卡,其中包括 EtherCAT FOE (File Over EtherCAT)功能。这些 EtherCAT 主站控制器和 PCIe EtherCAT 通讯卡可以快速、有效、便捷地构建数控智能化设备。通过支持固件更新、配置文件下载、数据记录与监控、远程控制与诊断、分布式文件系统等功能,更能够适应工厂智能化、信息化产业的需求。

表3 EtherCAT主站控制器

코号	内核	主祭	内存	存储	以太网	无线	CAN	RS485	M.2	DI/DO	AI/AO	HDMI
ZMC300E	単位Cortex®-A8	800MHz	256MB	256MB	1*EtherCAT+2*NET		1 18	1 \$25		16/16		
ZMC600E	Cortex®-A53*2+R5F*4	1GHz	1GB	4GB	1*EtherCAT+3*NET		2 路	2 路		16/16		
ZMC601E	Cortex®-A53*2+R5F*4	1GHz	168	4G8	1*EtherCAT+3*NET		2語	2語		16/16	2/2	
ZIMC602E	Cortex®-A53*2+R5F*4	1GHz	1GB	4GB	1*EtherCAT+5*NET		3 歸	2路		16/16		
ZMC60E	Cortex®-A53*2+R5F*4	1GHz	1GB	4G8	1*EtherCAT+2*NET	Wi-Fi/4G	2 #8	2 18		8/8		
ZMC61E	Cortex®-A53*2+R5F*4	1GHz	1GB	4G8	1*EtherCAT+2*NET	Wi-Fi/4G	1 28	2 路		8/8	6/2	
ZIMC900E	Cortex®-A55*4+R5F*3	2GHz	4G8	8G8	2*EtherCAT+2*NET	Wi-Fi/4G	1 25	1 25	支持	16/16		支持



图5 EtherCAT主站控制器系统框图

表4 PCIe EtherCAT主站通讯卡

型号	结构	EtherCAT	线路冗余	热插拔	CAN/CAN FD	DI/DO	编码器	PWM
HPCIe-2E	PCIe 半高	1路主站, 2网口	支持	支持				
PCIe-4E	PCle 全高	2 路主站, 4 网口	支持	支持				
PCIe-2EC04	PCIe 全高	1路主站, 2网口	支持	支持	1路	4DI/4DO		
PCIe-1E16P	PCle 全高	1路主站, 1网口		支持		16DI/16DO	1路	4路
MiniPCIe-2E	MiniPCle	1 路主站, 2 网口	支持					



图6 PCIe EtherCAT通讯卡



【插针机HMI开发】 用AWTK开发人机界面

ZLG 致远电子 2024-04-18 11:43:33

插针机中有两个重要的部件 HMI 和 PLC,HMI 提供用户操作和界面显示,而 PLC 则控制电机的运动逻辑。致远电子的 ZTP800 和 AWTK-HMI 解决方案可以快速开发插针机或其他设备的 HMI 界面。

AWTK介绍

致远电子的 AWTK 是开源 GUI 引擎,提供了 AWTK-C、AWTK-MVVM 和 AWTK-HMI 等多种解决方案来实现界面编程。其中 AWTK-C 解决方案 使用 C 语言来开发 GUI,灵活度最高,但是学习使用门槛也较高。AWTK-MVVM 解决方案则进一步简化 C 的编程,支持控件与变量绑定联动,让界 面和业务开发分离。而 AWTK-HMI 解决方案使用最简单,内置 Modbus 通 信模块,不用写 C 代码,只需使用 AWStudio 软件拖拽控件和配置属性即 可实现 HMI 界面,因此本次也是使用 AWTK-HMI 方案来实现插针机的上位 机。



图1 AWTK开源GUI引擎

插针机系统介绍

本次使用 ZTP800 作为插针机的 HMI 设备,中间使用 Modbus 协议与 PLC 通信。ZTP800 中内置的 AWTK-HMI 运行时引擎,让用户无需额外编写 C 代码即可实现插针机的人机交互界面。



图2 插针机系统介绍

注: AWTK-HMI 是开源的 HMI 运行时引擎, 如想了解编译方法请参考 AWTK-HMI 仓库的说明文档,在 ZTP800 的部署 AWTK 可以参考 ZTP800 在线文档的 AWTK移植章节。

插针机HMI开发流程

- 只需要3步:
- 1. 在 PC 配置好 Modbus 的通信点表文件;
- 2. 使用 AWStudio 软件通过简单拖拽做界面设计;
- 3. 最后把工程下载到 ZTP800 运行。

1. 定义Modbus寄存器点表

先定义一个 Modbus 寄存器点表来约定 HMI 和 PLC 的通信规范,HMI 作为 Modbus 主机,PLC 作为从机。该表格需要根据插针机的需求来定,例如: 手动移动托板,手动移动主轴,调整主轴/托板速度等。





图4 Modbus 寄存器写入表格图2

2. 创建AWTK-HMI的Modbus配置文件

AWTK-HMI 要使用 Modbus 通信功能,需要用户创建一个配置文件 modbus_model.json。



图5 modbus_model.json配置文件

modbus_model.json 主要用于配置 Modbus 通信的寄存器和 AWTK 变 量对象的映射关系,每个 channel(通道)用于表示一片读写 Modbus 寄 存器区域,而 variables 定义了 AWTK 变量对象和 channel 的关系。

3. AWTK-HMI界面开发

在 AWStudio 创建一个新 AWTK 工程,如下图:



图6 插针机首页开发界面

把需要显示的控件从工具栏拖放到页面上:

-													
Ξ.				Week and show do -		10	(As 60 TR		_				
				四时40步台:	(g_name)	埰	TF3FUL		NR 14	20.45			
			20 I										
	C leading												
	2 operation	_	18						1				
	D program, manager								s_model ? "IS IS	232			
	C program, writing	-											
	0.000,000	_						相工作原点	昆机械零点			To Manual T	
	- 0.7779 M											• 1014888	
			199										
				Z -	Χ-	Z+		种针细	种物:				
	1.000												
				¥	X+	¥+		N-112630	料11港台				
			10	1514	1407								
					1.000								
			40										
									The second se				
									派回				

图7 插针机操作界面

首先,需要给窗口配置 v-model 属性,启用 Modbus 功能。 在窗口增加一个自定义属性 v-model,属性值设置为【modbus_ client(name=modbus_model,share=true)】,代表该页面使用 modbus_ model.json 的通信配置。如下图:

属性名称: v-model 属性值: modbus_client(name=modbus_model,sha e=true)) 現定 取消	添加自定义属性		×
屬性值: modbus_client(name=modbus_model,sha e=true) 建築定 取消	属性名称:	v-model	
	属性值:	modbus_client(name=modbus_model,sha• e=true) •	
		施定 取消	כ

图8 变量绑定规则界面

下一步,把上面 modbus_model.json 映射好的变量绑定到控件上面, 让 Label 控件根据变量当前值自动改变显示内容,或自动把 Edit 控件设置 的参数下发给下位机。

例如要让 Label 控件显示补针状态,可以给这个 Label 控件添加自定义 属性【v-data:text】,属性值设置为【g_is_needle_model?"补针状态":" 非补针状态"】,代表控件的 text 属性会跟随 g_is_needle_model 变量变化。 如下图:



图9 添加自定义属性界面

例如要把设置托板速度的 Edit 控件的输入数值下发到下位机,可以给 这个 Edit 控件添加自定义属性【v-data:text】,属性值设置为【g_plane_ speed】,代表控件的 text 属性改变后会立即更新 g_plane_speed 变量。 如下图:

添加自定义属性		×	
属性名称: 属性值:	v-data:text g_plane_speed	İ	
	确定	取消	

图10 添加自定义属性界面2

Modbus模块会自动同步g_is_needle_model、g_plane_speed等变量,并周期读写从机,因此上面控件显示就会周期性的更新,或周期性的把输入值写入从机。

用户甚至可以把变量值立即写入从机,在控件的事件响应脚本中增加 代码【exec('update', 'write_registers')】即可,其中 write_registers 就是 modbus_model.json 定义的通道名。

4. AWTK-HMI工程下载运行

上面的工程开发完成后,点击 AWStudio 上的打包按钮,在工程目 录下会生成出一个 res 文件夹,如下图所示。把 res 文件夹的内容下载到 ZTP800 的 AWTK-HMI 运行时程序的对应位置,重启设备就可以顺利启动我 们新开发的 HMI 界面。

名称	修改日期	类型	大小
👩 3rd	2024/2/2 17:42	文件夹	
🤣 bin	2024/2/2 17:42	文件夹	
💋 design	2024/2/2 17:42	文件夹	
🔗 res	2024/3/13 14:11	文件夹	
scripts	2024/2/2 17:42	文件夹	
🤣 src	2024/3/13 14:11	文件夹	
🧟 .gitignore	2024/2/2 17:42	文本文档	
🔊 crc.json	2024/3/13 14:11	JSON File	
🔊 manifest.json	2024/2/2 17:42	JSON File	
🔊 project.json	2024/2/2 17:42	JSON File	
README.md	2024/2/2 17:42	Markdown File	
SConstruct	2024/2/2 17:42	文件	

图11 res文件夹





点击申请

边缘计算 ▼

【插针机PLC开发】 用AWBlock开发运动逻辑

ZLG 致远电子 2024-04-19 11:41:52

上期我们介绍了 HMI 开发过程,这期将介绍 PLC 运动逻辑开发。致远 电子的 ZMC600E 和 AWStudio 提供了 ST 语言和 AWBlock 积木式中 文编程环境,可快速开发插针机或其他设备的 PLC 逻辑。

AWStudio介绍

AWStudio 是工业自动化 All In One 可视化集成开发环境,可以用于开 发 HMI 和 PLC 程序。针对 PLC 开发目前提供了多种解决方案:一种是符合 IEC61131-3 标准的 ST 开发环境,另一种是 AWBlock 积木式中文编程环境, 未来还将提供梯形图和功能块图编程。用 AWBlock 写 PLC 逻辑像搭积木一 样直观容易上手,不需要敲代码也避免了出错,AWBlock提供了运动控制块, 特别适合做机器人和运动控制等应用,所以本次 PLC 使用 AWBlock 编程方 案。

插针机系统介绍

本次使用 ZMC600E 作为插针机的 PLC 设备,中间使用 Modbus 协议 与 HMI 通信,使用 EtherCAT 协议与电机伺服器通信。ZMC600E 中内置的 AWPLC(含 AWBlock)运行时引擎,可以运行用户编写的运动逻辑,实现 插针机的多轴协同运动控制。



图1 插针机系统介绍



图 2 创建解决方案

2. EtherCAT总线配置

插针机系统中,PLC 作为 EtherCAT 主站,电机伺服器作为 EtherCAT 从站,所以要在 PLC 总线协议中添加 EtherCAT 主站功能。在总线协议节点 上面添加总线,选择 EtherCAT 主站,如下图:



图 3 添加 EtherCAT 主站协议

在 EtherCAT 主站节点上添加所有轴的电机伺服器,分别是托盘 X 轴、 托盘 Y 轴、主轴 Z 轴。再配置 EtherCAT 的分布时钟,如下图:

	文件 总线协议 构i	
0	 新決方面 	Master1 * ×
ø	Ambiocontrancontrollero A >2 LogicConfiguration	主站 过程映像 周期命令 分布时钟 从站间通信
۲	- 🖿 任务管理	
	■ MainTask.json ▲ ■ 相序知理 ■ main.json	 → 土地力参考时钟 ● 从地力参考时钟
	 → EverceConfiguration → 硬作资源 → Di → → →	
	 DO Ethernet 	
	• <u>Bitthin</u> 双击工站节点 • ■ Master1 (EtherCAT Master)	 >>訪問参規式
	Slave1 (MR-JET-G-N1) (EtherCAT Slave)	
	Slave3 (MR-JET-G-NL) (EtherCAT Slave)	
		□ 19180099140 81991089714000 100 0

插针机PLC开发流程

只需要四步:

- 一、配置 EtherCAT 总线;
- 二、配置 Modbus 从机;
- 三、编写 AWBlock 控制电机运动逻辑;
- 四、最后把工程下载到 ZMC600E 运行。

1. 创建AWBlock解决方案

在 AWStudio 创建一个新 AWBlock 编程解决方案工程,如下图:

配置参考时钟有利于运动控制的时候所有电机的同步移动,AWStudio 也提供了完整的 EtherCAT 的配置参数,用户可以根据实际情况来配置自己 的 EtherCAT 通信。

3. Modbus从机配置

插针机系统中,PLC 作为 Modbus 从站,HMI 作为 Modbus 主站,所 以要在 PLC 总线协议中添加 Modbus 从站功能。然后根据主从机约定好的 Modbus 寄存器点表来配置寄存器地址和个数,PLC 运行过程中实时更新这 些寄存器供 HMI 访问,如下图:



图5 配置Modbus从机寄存器

4. AWBlock运动逻辑开发

插针机最基础的功能是控制托盘和主轴(Z轴)移动,所以要创建 move_plane 和 move_main_axis 两个函数。编辑 move_main_axis 函数 文件,分别拖入 MC_Power 和 MC_MoveAbsolute 功能块来控制电机运动, 如下图:

	从 站车管理 科建 导出 总线协议 的建				
Harding Advance Advanced					
 AWBlock ▼ RE 	MC, ReadStatussAnise M Er	uble A · Ya			
11 RF 26 21	HC. Down	Axis• 10 Enable• 11 v bReg	ulatorOne 🕅 - bDriveStarte 🕅 -	Statuer 🕅 👻 bRegulatorRealStat	er 🕅 🔸 börvestanskalst
C 604					
S 1201					

图6 拖入运动控制块

上面已经在 EtherCAT 主站下添加了电机了,这里会自动识别出电机并 且会自动给每一个电机生成一个轴对象,因此轴参数便可以选择需要控制 的电机,如下图:



图7 选择控制轴

	+	u ⊎ - +			
	77-5				
	0	VAR_INPUT	Speed		LREAL
100	4				
		MC_Power:Axis= POUDE	/_VARAXIS2 Enable= ┦ → bRegulatorOn= ┦	× bDriveStart= 報 × :	Status= <mark>假 、</mark> bRegulator
		MC_MoveAbsolute:Axis= PO	UDEV_VARAXIS2 Execute= Execute Position= I	osition Velocity= Speed	Acce Accel Dece Decel

.

接着基于上面写好的基础运动功能块,编写控制托盘和主轴运动顺序 和逻辑(主轴归 0-- 移动托盘 -- 主轴插针 -- 主轴归 0),比如控制自动插针 的 auto_move 函数,如下图:



图9 用AWBlock写运动逻辑

最后把 PLC 的运行参数和状态对接到 Modbus 从站寄存器,例如: MainAxisSpeed 是主轴速度变量,对应到 PLC 的地址是 %IW52,%IW52 映射到 Modbus 从站地址为 40005,如下图。这样,就可以让 HMI 通过 Modbus 控制和获取 PLC 运行状态。



图10 关系映射表



图11 变量绑定地址



5. AWBlock工程下载运行

上面的工程开发完成后,在设备列表中选择目标设备 ZMC600E,点击 启动(调试)按钮,就会进入调试模式,可以进行单步调试程序;点击下 载即可发布固件,ZMC600E 重启后就会自动运行该工程。如下图:

上 ▶ 下载 启动	了 。 启动(调试)	备:	Unknown (00 1 💉	
· ·	运行		模拟器	
			AWPLC Simulator	
			以太网设备	C
			Unknown (00 14 97 0f 01 90) Host: 172.168.18.203:8889, Version: 0.0.1	
		INT		

图12 设备选择界面

ZMC600E产品介绍



图13 ZMC600E EtherCAT主站控制器

ZMC600E 是 ZLG 致远电子开发的最新一代智能总线型运动控制器,是 面向工厂智能化时代的机器控制器。其采用工业领域内先进的嵌入式 ARM 方案,集实时操作系统、智能算法于一身,配套工业图形化编程软件开发 环境。

ZMC600E 采用 TI 的双核 64 位 Arm-Corte-A53,四核 Cortex-R5F 的 AM6442 应用处理器为核心,主频 1GHz,内置 1GB DDR4、4GB eMMC 以 及 32KB FRAM,预留有多路以太网、CAN、IO、USB 等硬件接口。同时 ZMC600E 支持点位运动、连续轨迹、直线圆弧插补、连续插补、螺旋线等 运动功能,可以自由设定运行速度、停止速度、加、减速时间可以独立设置, S 型曲线平滑等参数,支持在线改变速度和在线改变位置,用户可以轻松构 智能化控制系统,快速实现和部署现场各种工艺的应用。



【产品应用】 ZMC900E控制器之ROS2环境安装指南

ZLG 致远电子 2024-04-28 11:38:17

在物联网时代,无线连接对智能设备至关重要。EPCM3568A-LI工控 机作为边缘计算网关,简化了设备联网流程,支持远程数据传输与智 能操作。本文将指导如何通过该工控机轻松实现 WiFi连接和热点共享。

ZMC900E 是致远电子研发的最新一代高性能 EtherCAT 控制器,支持 ROS2 系统,兼备强大功能和实时性,为工业应用带来新的可能。本文将详述 ROS2 特点和安装,助力用户在机器人上的二次开发和应用。



图1 ZMC900E高性能EtherCAT主站控制器

ZMC900E 是 ZLG 致远电子开发的最新一代 EtherCAT 主站控制器,其核心采用多核异构的应用处理器,内核包括 4+1 个 64 位的Arm®Cortex®-A55 核,主频 2GHz; 3 个 Cortex-R5F 内核,主频 800MHz。同时板载 4GB LPDDR4、8GB eMMC 以及 32KB FRAM。

ZMC900E EtherCAT 主站控制器为了满足不同的自动化应用需求,集成 1路专用 EtherCAT 口、3路通用以太网、1路 CANFD、1路 RS485、1路 TF 卡、 1路 USB3.0 Host、1路 HDMI 接口、16路 DI 数字输入、16路 DO 数字输 出等接口,灵活满足自动化设备应用需求。



图2 ZMC900E接口

ROS 因其强大的功能而备受机器人应用开发者的欢迎,而新一代的 ROS2 更具有产品化特性,结合新一代高性能的 ZMC900E EtherCat 控制 器将给工业应用带来更多想象力。下面就详细介绍 ROS 特性以及 ROS2 在 ZMC900E 上的安装步骤。

ROS是什么?

ROS(Robot Operating System),这一为机器人软件程序编写的软件架构,展现了其独特的灵活性。其起源可追溯到斯坦福大学的 STanford Artificial Intelligence Robot(STAIR)与 Personal Robotics(PR)项目,象征着机器人领域的创新与探索。

在机器人行业,工业界与学术界在软件工具的使用上呈现出截然不同 的态度。工业界倾向于构建封闭的生态系统和设置高技术壁垒,以保护其 核心优势。而学术界则更偏向于拥抱开源社区,选择现有的工具进行研发, 其中,ROS 便是学术界广受欢迎的开源工具。

自 2010 年首个 ROS 版本发布以来,它基于 PR2 机器人构建了一系列 基础软件包,并不断进行迭代更新。随着 Ubuntu 系统长期支持(LTS)版 本的演进, ROS 也逐步升级,目前最新的版本已经与 Ubuntu 20.04 LTS 完 美兼容。尽管 ROS 的 CI 测试主要在 Ubuntu 上进行,但社区成员的努力使 得 ROS 在其他 Linux 版本、Mac OS X、Android、Windows 等平台上也具 备一定的兼容性,尽管功能支持可能受到一些限制。



ROS的的优缺点分析

ROS 作为一种广泛使用的机器人软件架构,为开发者提供了许多便利, 但同时也存在一些明显的问题和挑战。以下是对 ROS 的优缺点进行的分析:

- 1. 优点
- 松散耦合的机制提供框架: ROS采用了一种松散耦合的设计,使得 各个组件之间的通信更加灵活和方便。这种设计使得开发者可以根 据需要轻松地添加、删除或修改功能,而无需对整个系统进行大规 模的修改。
- 丰富的机器人功能库: ROS 提供了大量的机器人功能库,涵盖了感知、导航、控制、定位、建图等多个方面。这些库为开发者提供了丰富的资源,可以加速机器人应用的开发进程。

- 庞大的用户群体: ROS 在机器人领域拥有庞大的用户群体,这意味着开发者可以更容易地找到帮助、解决问题和分享经验。同时,这也促进了 ROS 社区的发展,使得 ROS 不断完善和进步。
- 免费开源: ROS 是一个开源项目,这意味着任何人都可以免费地使 用其中的代码、库和工具。这为开发者提供了更多的选择和灵活性, 同时也促进了机器人技术的普及和发展。
- 便利的数据记录、分析、仿真工具: ROS 提供了一系列的数据记录、 分析和仿真工具,使得开发者可以更方便地进行机器人系统的测试、 调试和优化。

2. 缺点:

- 通信实时性能有限:由于 ROS 的通信机制是基于消息的发布 / 订阅 模式,因此在实际应用中可能会遇到实时性能有限的问题。特别是在 高负载或复杂场景下,通信延迟和丢包等问题可能会影响系统的性 能。
- 系统稳定性尚不满足工业级要求:虽然 ROS 在学术界和科研领域得 到了广泛应用,但在工业级应用中,其系统稳定性仍然存在一定的不 足。这主要是因为 ROS 在设计时更多地考虑了灵活性和可扩展性, 而在稳定性方面的考虑相对较少。
- 安全性上没有防护措施: ROS 的安全性是另一个值得关注的问题。
 由于 ROS 是一个开源项目,缺乏统一的安全标准和防护措施,因此
 在实际应用中可能会面临一定的安全风险。例如,恶意攻击者可能会利用 ROS 的漏洞进行攻击或破坏。
- 仅支持 Linux(Ubuntu): 虽然 ROS 在 Linux 系统上得到了广泛的支持 和应用,但这也限制了其在其他操作系统上的使用。这可能会使得一 些开发者在选择 ROS 时面临一些挑战和限制。
- 核心机制的性能没有优化占用资源:ROS的核心机制在某些情况下可能存在性能瓶颈和资源占用过高的问题。这可能会影响系统的整体性能和效率,尤其是在资源有限的嵌入式系统或移动机器人上。

ROS2: 推动ROS进入产业化与产品化时代

ROS 在学术界和科研领域取得了显著的成就。然而,对于工业界和产业 应用而言,原始的 ROS 版本(即 ROS1)存在一些明显的不足,阻碍了其真 正的产业化与产品化进程。为了解决这些问题,ROS 社区推出了 ROS2,一 个更为成熟、稳定和适应产业化需求的版本。

1. 实时化与分布式通信

ROS2 的核心改进之一是实现了实时化和分布式通信。在 ROS1 中,所 有的节点通信都依赖于一个中心化的 Master 节点。这意味着如果 Master 节 点发生故障,整个系统的通信将中断,导致所有节点功能失效。这种设计限 制了 ROS1 在需要高可靠性和稳定性的工业环境中的应用。



图4 ROS1和ROS2比较

为了克服这一挑战,ROS2 引入了 Data Distribution Service (DDS) 作 为其通信基础。DDS 是一种广泛应用于国防、民航、工业控制等领域的实时 系统数据发布 / 订阅标准解决方案。通过 DDS,ROS2 实现了去中心化的通 信方式,允许节点之间进行直接通信,无需依赖于 Master 节点。这种分布 式通信方式不仅提高了系统的可靠性和稳定性,还有助于解决多机器人系统 中的通信问题。

2. 适应全平台与低性能硬件

除了实时化和分布式通信外,ROS2 还致力于提高其对不同平台和硬件 的适应性。在 ROS1 中,由于其对特定操作系统(如 Ubuntu)的依赖,使 得其在其他平台上的应用受到限制。而 ROS2 则致力于实现跨平台兼容性, 支持更多的操作系统和硬件平台。

此外,ROS2 还针对低性能硬件(如 MCU+RTOS)进行了优化。这意味 着在资源有限的嵌入式系统或移动机器人上,ROS2 仍然能够保持良好的性 能和效率。

3. 数据加密与现代编程语言支持

在数据安全方面,ROS2引入了数据加密机制,确保通信过程中的数据 安全性。这一特性对于需要高度安全保护的工业应用至关重要。

同时, ROS2 还支持现代编程语言,如 C++ 和 Python。这使得开发者 能够使用更加高效和灵活的编程工具来开发机器人应用。

通过引入实时化、分布式通信、跨平台兼容性、低性能硬件适应性、数 据加密以及现代编程语言支持等特性,ROS2为机器人技术的产业化与产品 化奠定了坚实的基础。随着ROS2的不断发展和完善,我们有理由相信它将 在未来的机器人领域中发挥更加重要的作用。

ROS2环境安装

ZMC900E 支持 ROS2 的开发,本地安装的系统是 Ubuntu 18.04,可以 安装对应的 eloquent 版本的 ROS2。依次按以下步骤操作。

1. 更换apt源

sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak sudo vi /etc/apt/sources.list

2. 替换成以下内容

deb http://mirrors.huaweicloud.com/ubuntu-ports/ bionic main multiverse restricted universe

deb http://mirrors.huaweicloud.com/ubuntu-ports/ bionicbackports main multiverse restricted universe

deb http://mirrors.huaweicloud.com/ubuntu-ports/ bionicproposed main multiverse restricted universe

deb http://mirrors.huaweicloud.com/ubuntu-ports/ bionicsecurity main multiverse restricted universe

deb http://mirrors.huaweicloud.com/ubuntu-ports/ bionicupdates main multiverse restricted universe

deb-src http://mirrors.huaweicloud.com/ubuntu-ports/ bionic main multiverse restricted universe

deb-src http://mirrors.huaweicloud.com/ubuntu-ports/

bionic-backports main multiverse restricted universe deb-src http://mirrors.huaweicloud.com/ubuntu-ports/ bionic-proposed main multiverse restricted universe deb-src http://mirrors.huaweicloud.com/ubuntu-ports/ bionic-security main multiverse restricted universe deb-src http://mirrors.huaweicloud.com/ubuntu-ports/ bionic-updates main multiverse restricted universe

3. 安装aptitude包

sudo apt update && sudo apt install aptitude

4. 设置语言环境

sudo apt update && sudo apt install locales sudo locale-gen en_US en_US.UTF-8 sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_ US.UTF-8 export LANG=en_US.UTF-8

5. 需要将 ROS2 apt 存储库添加到系统中,为此,首先使用 apt 授权 GPG 密钥

sudo apt update && sudo apt install curl gnupg2 lsb-release curl -s https://raw.githubusercontent.com/ros/rosdistro/ master/ros.asc | sudo apt-key add -

6. 然后将存储库添加到源列表

sudo sh -c 'echo "deb [arch=\$(dpkg --print-architecture)]
http://packages.ros.org/ros2/ubuntu \$(lsb_release -cs) main"
> /etc/apt/sources.list.d/ros2-latest.list'

7. 安装ROS2软件包,设置存储库后,更新apt存储库缓存,安装桌面版

sudo apt update && sudo aptitude install ros-eloquentdesktop

注: 安装过程可能出现问题。 问题1: GPG error: http://packages.ros.org/ros2/ubuntu bionic InRelease: The followi ng signatures couldn't be verified because the public key is not available: NO_ PUBKEY F42ED6FBAB17C654 方案1: sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys F42ED6FBAB17C654 经过上述操作, eloquent版ROS2安装完成。

ROS2测试

在终端中,更新配置文件,运行 talker:

source /opt/ros/eloquent/setup.bash
ros2 run demo_nodes_cpp talker

在另一个终端中,更新配置文件,运行 listener:

source /opt/ros/eloquent/setup.bash ros2 run demo_nodes_py listener

可以看到 talker 发布 Publishing 消息,同时 listener 接收并告知 l heard 消息,如下图所示,说明安装成功。



如需了解更多产品详情,可填写申请表单, 我们会有专人与您联系。



ZLGi微文摘

互联互通▼

【技术分享】 CAN 会收到错误的数据吗?

ZLG 致远电子 2024-04-03 11:43:07

CAN-bus 总线协议以高稳定性,高容错率而著称于世,然而仍有很多 用户在使用的时候担心 CAN 会接受到错误的信息,在数据里增加了 CRC 校验的部分,这种做法是否有必要? CAN 会收到错误的数据吗?

信息的传递,古往今来都是人类无比关注的一个问题。从最原始的肢体语言到高端的电子信号,信息传达的方法五花八门。而对于信息安全的 追求也是从古至今未有变过,我国西周时期的《太公兵法》就有过"阴符""阴 书"的设计来保证信息的安全。



图1让人心忧的信息安全

而在我们的工业生产中,为了保证信号的正确传递的方法更是五花八 门。而在信息传递过程中采取 CAN 协议是一种常见的减少出错率的方案。 那么,问题来了,CAN 协议何德何能能让传输的信号不出错呢? 今天就让 我们来深入分析一番。利用 CANScope 总线综合分析仪来抓取一帧 CAN 的 报文如下:



图2 CANScope总线综合分析仪抓取的报文

从图中可以看出 CAN 协议采取了差分信号传输的方式,可以有效杜绝 来自外部的屏蔽干扰。而在最后一行的协议解析部分,经过观察我们发现一 帧信号被分割成了不同颜色的一段段,每一段究竟是何含义?保证信号正 确传输的秘密就隐藏在这些段落里。让我们来庖丁解牛分别为大家分析下。 • 帧起启:在数据的开始,是一个1位的数据头,表示数据帧都开始



仲裁段:标志了本帧数据的优先级,其中包含了一个ID码,仲裁段中的ID码值越小,帧数据的优先级就越高,CAN控制器在发送数据的同时会监听电缆上的电平状态,如果发现仲裁位的电平与本节点发出的电平不一致,则退出发送放弃总线使用权。这样的设计可以提高总线的利用率,并且能让重要的信息优先发送。

CAN-分析		BasicID:256 H	RTRIDE R0
	图4	仲裁段	

• 控制段:共六位,用于表示数据长度。在数据的控制段存有保留位 以供未来协议规则扩展。



数据段:经过前面的铺垫,数据段所编码的即是本帧数据所需要传达的信息。一帧信号可以传送 0~8 位数据,每字节 8 位。短小精悍保证信息的实时性。



• CRC 段: CRC 段即是保证数据准确的一个关键所在(敲黑板)。为防止信号由于某种原因被更改,CAN 的数据链路层上加入了 CRC 校验。发送节点会根据发送内容计算得到一个 CRC 值填入 CRC 段进行发送,而相应的接收节点也会对接收到的数据进行计算,并将计算出的 CRC 值和接收到的进行比对。能够对得上暗号的才是自己人,如果对照有误那么就说明传输的信号出现了问题,需要反馈错误消息。这样的机制保证了 CAN 不会收到错误的信息,其安全性毋庸置疑。



ACK 段:用于表征信号是否被正确接收,接受正常的节点在ACK的第一位会发出一个显性位。根据ACK的状态,发送节点就可以了解到数据是否被传输成功。若发送失败,发送节点会根据自身状态来决定是否重传。



• 帧结束:由7个隐性位组成,表示该帧结束。





- 经过这样一番抽丝剥茧的分析, CAN 的报文结构就这样清晰的展现 在我们面前。由于 CRC 段的存在, CAN 出错的概率十分之小。
- CRC 校验所使用的 CRC 多项式最多可以检测出 5 个离散错误,或发现长度在 15 位以下偶然出现的突发错误。CRC 校验对 SOF 位、仲裁段、控制段和数据段的位序列进行计算,但不考虑填充位。CAN协议中规定的 15 位校验序列源自于 BCH 代码,它是一种特别适用于 127 位以下消息长度的循环代码。CAN 协议中所应用的 15 位多项式如下:

$$X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

- 在发送或接收收到数据场的最后一位后,CRC寄存器就会包含待传 输或者待接收的CRC序列。将计算出的CRC序列与接收到的CRC 序列相除,接收器就可以识别出可能存在的CRC错误。
- 有些工程师担心 CAN 收到错误的信息,在数据中又做了 CRC 校验 的工作,岂不知在数据链路层 CAN 已经自备了 CRC 校验的工作, 在数据中再加入 CRC 校验实际上是没有什么必要的。



CAN 总线不但规定了物理层的差分传输规范,还规定了数据链路层的分包校验规则,而这两个都是由硬件自动完成,接收时,无需考虑是否有错误,只要从缓冲区取出数据即可,CAN 的CRC 校验可以保证错误率在10的-9次方以下。毫无疑问是一种非常安全可靠的传输协议。



 CAN 总线在信号的实时传输方面具有非常好的优越性,通过 ZPS-CANFD 总线分析仪 可以很好的完成 CAN 总线的故障排查与检测标 定。致远电子凭借自身掌握的核心技术可为用户解决工业现场的各 种疑难问题,期待与您一同成长。



ZLG微文摘

互联互诵▼

【技术分享】 CAN报文为什么会发送失败?

ZLG 致远电子 2024-04-11 11:41:12

CAN 总线调试过程中出现报文发送失败。很多工程师都对此只知其一 不知其二,今天我们就以 CAN 报文发送失败的问题来做一次探讨。

在了解 CAN 报文为什么会发送失败之前我们先看看一帧标准的 CAN 报 文到底应该是怎么样的。表1是一帧正常标准数据帧的报文组成。



表1标准数据帧报文格式组成

图1标准数据帧格式

CAN 总线是一种基于广播的通讯方式,为了保证总线上的每一个正常 节点都能正确的接收到报文,报文的发送者要求至少一个接收节点在报文 发送结束前要作出应答,这也是报文里 ACK 存在的原因。

一帧 CAN 报文中 ACK 段长度为 2 个位,包含应答间隙(ACK Slot)和 应答界定符(ACK Delimter)。在应答场里,发送器发送两个隐性位。当接 收器正确地接收到有效的报文,接收器就会在应答间隙(ACK Slot)期间(发 送 ACK 信号) 向发送器发送一"显性"的位以示应答。

- 应答间隙:所有接收到匹配 CRC 序列(CRC SEQUENCE)的站会在 应答间隙(ACK Slot)期间用一显性的位写入发送器的隐性位来作出 回答。
- 应答界定符: ACK 界定符是 ACK 场的第二个位,并且是一个必 须为隐性的位。因此,应答间隙(ACK Slot)被两个隐性的位所 包围,也就是CRC 界定符(CRC Delimter)和ACK 界定符(ACK Delimter)



图2正常ACK段报文

而如果总线上没有 ACK 应答(即应答间隙为隐性),发送器就会发送 一个错误标志,并且发送错误计数器值加8,节点就会对报文进行自动重发, 若自动重发依然收不到 ACK,则在发送错误计数器计数满 128 后(即出现 16 帧错误帧),由错误主动转为错误被动状态,如图3所示。

那导致 ACK 段出错的原因有哪些呢?下面小编总结了一些。

- 总线上只有一个有效节点:发送报文的节点在发送出一帧报文后会 检测总线上应答间隙的状态,如果检测到应答间隙为隐性位,则表 示该帧报文没有得到 ACK,发送失败,需要重发,而由于发送错误 计数器会在发送失败后累加,直到该节点关闭。所以,当总线上只 有一个有效节点时,这个节点是发不出去数据的,因为它所发出的 数据帧中的 ACK Slot 没有另外一个节点来填充,将永远是隐性位, 这个节点会一直重发数据直到发送成功或发送被取消。
- · 波特率不匹配或者节点没有初始化,导致没有 ACK;
- · 总线线缆短路,断路,接反;
- 高速 CAN 总线上接的节点不是高速 CAN,而是容错低速 CAN,导 致不匹配。

- CAN报文 × 🕢 网络共享 🔄 CAN波形 🗰 CAN眼图 🔍 CAN示波器											
4) 🔥 🎋 🎋 🚔 🗮 自动量程 😂 自动液屏 🐻 清除过滤 🚽 清除列表 🐠											
序号		时间	状态	方向	帧类型		数	帧ID		帧数据	事件标记
在此处料	輸入、	7 在此处输入文字	☞ 在此处输入	7 在此… '	7 在此处输	7	17	在	7	在此	▼ 在此处输入:
·	1	00:00:18.337 914	应答定界符	接收							幀错误
· ·	2	00:00:18.338 034	定界符格式	接收							帧错误
	3	00:00:18.338 154	应答定界符	接收							帧错误
NI 🚩	4	00:00:18.338 274	定界符格式	接收							响错误
- 101	5	00:00:18.338 394	应答定界符	接收							帧错误
·	6	00:00:18.338 514	定界符格式	接收							幀错误
· ·	7	00:00:18.338 634	应答定界符	接收							响错误
TUI 🚩	8	00:00:18.338 754	定界符格式	接收							帧错误
(UI) 🚩	9	00:00:18.338 874	应答定界符	接收							幀错误
TU 1	10	00:00:18.338 994	定界符格式	接收							幀错误
·	11	00:00:18.339 114	应答定界符	接收							幀错误
· •	12	00:00:18.339 234	定界符格式	接收							响错误
TUS 🚩	13	00:00:18.339 354	应答定界符	接收							响错误
·	14	00:00:18.339 474	定界符格式	接收							帧错误
>	15	00:00:18.339 594	应答定界符	接收							幀错误
·	16	00:00:18.339 714	定界符格式	接收							幀错误
	17	00:00:18.339 833	成功	接收	标准数据响	8	3	001 H		11 2	
	18	00:00:18.339 961	成功	接收	标准数据响	8	3	001 H		11 2	
	19	00:00:18.340 089	成功	接收	标准数据帧	8	3	001 H		11 2	

图3 应答界定符错误帧



图4 没有ACK的报文

当你在调试 CAN 总线时出现节点发送报文失败的情况时,一定要检 查是不是以上几点疏漏导致你的总线上 ACK 异常。而借助恰当的仪器,可 以在查找 CAN 总线错误时事半功倍。图 4 即采用致远电子的 CANScope 来对错误帧进行标记,同时找到错误帧对应的波形来查找出错误情况。 CANScope 还可以对 CAN 总线物理层、数据链路层、应用层做一系列的测试, 为 CAN 工程师解决测试难题。

项目		结果	备注	设备要求
4	全部测试	1.4440	1	[wares
H	口 1. 电压测试			Std Pro
-	口 2. 边沿测试			Std Pro
H	□ 3. 总线延时测试			Std Pro
H	口 4. 总线利用率测试			Std Pro
-	5. 总线错误率测试			Std Pro
+	🔲 6. 对称性测试			Std Pro
+	口 7.采样点测试			Std Pro
H	□ 8. 位宽容忍度测试			Std Pro
+	口 9. 报文压力测试			Std Pro
-	口 10. 错误波特率压力测试			Std Pro
-	口 11. 抗干扰能力测试			Pro
H	口 12. 总线短路测试			Pro+Stre
+	🗖 13. 总线断路测试			Pro+Stre
F	口 14. 容抗增加压力测试			Pro+Stre.
-	口 15. 对正负电源或地短路测			Pro+Stre.
H	口 16. 终端电阻变化压力测试			Pro+Stre.
L	口17. 总线网络阻抗测试			Pro+Stre.

图5 CANScope测试项目



互联互通▼

【技术分享】 一文读懂CAN控制器错误处理的原理

ZLG 致远电子 2024-04-25 11:37:33

CAN 通讯的错误帧到底是如何被界定的?本文带你轻松了解!

错误标定

检测到错误条件的站通过发送错误标志指示错误。对于"错误主动" 的节点,错误信息为"主动错误标志",对于"错误被动"的节点,错误 信息为"被动错误标志"。站检测到无论是位错误、填充错误、形式错误, 还是应答错误,这个站会在下一位时发出错误标志信息。

只要检测到的错误的条件是 CRC 错误,错误标志的发送开始于 ACK 界 定符之后的位(其他的错误条件除外)。如图 1 所示:



图1 CRC错误帧

故障界定

至于故障界定,单元的状态可能为以下三种之一: "错误主动"、"错误被动"、"总线关闭"。

1. "错误主动"的单元可以正常地参与总线通讯并在错误被检测到时 发出主动错误标志。

 "错误被动"的单元不允许发送主动错误标志。"错误被动"的单元参与总线通讯,在错误被检测到时只发出被动错误标志。而且,发送以后, "错误被动"单元将在初始化下一个发送之前处于等待状态(见"挂起发送")。

3. **"总线关闭"**的单元不允许在总线上有任何的影响(比如,关闭输 出驱动器)。

三个状态的转换方式如图 2 所示:



在每一总线单元里使用两种计数以便故障界定: 发送错误计数、 接收 错误计数。

这些计数按以下规则改变(注意,在给定的报文发送期间,可能要用 到的规则不只一个):

 当接收器检测到一个错误,接收错误计数就加 1。在发送主动错误标 志或过载标志期间所检测到的错误为位错误时,接收错误计数器值不加 1。

2. 当错误标志发送以后,接收器检测到的第一个位为"显性"时,接 收错误计数值加8。

3. 当发送器发送一错误标志时,发送错误计数器值加 8。

- 例外情况1:发送器为"错误被动",并检测到一应答错误(注: 此应答错误由检测不到一"显性"ACK以及当发送被动错误标志时 检测不到一"显性"位而引起)。
- 例外情况2:发送器因为填充错误而发送错误标志(注:此填充错误发生于仲裁期间。引起填充错误是由于:填充位〈填充位〉位于RTR位之前,并已作为"隐性"发送,但是却被监视为"显性")。
 例外情况1和例外情况2时,发送错误计数器值不改变。

4. 发送主动错误标志或过载标志时,如果发送器检测到位错误,则发送错误计数器值加 8。

 5. 当发送主动错误标志或过载标志时,如果接受器检测到位错误(位 错误),则接收错误计数器值加8。

6. 在发送主动错误标志、被动错误标志或过载标志以后,任何节点最 多容许7个连续的"显性"位。以下的情况,每一发送器将它们的发送错 误计数值加8,及每一接收器的接收错误计数值加8:

- 当检测到第14个连续的"显性"位后;
- 在检测到第8个跟随着被动错误标志的连续的"显性" 位以后;
- 在每一附加的8个连续"显性"位顺序之后。



ZLGi微文摘

7. 报文成功传送后(得到 ACK 及直到帧末尾结束没有错误),发送错误 计数器值减 1,除非已经是 0。

8. 如果接收错误计数值介于1和127之间,在成功地接收到报文后(直到应答间隙接收没有错误,并成功地发送了ACK位),接收错误计数器值减
1。如果接收错误计数器值是0,则它保持0,如果大于127,则它会设置一个介于119到127之间值。

9. 当发送错误计数器值等于或超过 128 时,或当接收错误计数器值等于 或超过 128 时,节点为"错误被动"。让节点成为"错误被动"的错误条件 致使节点发出主动错误标志。

10. 当发送错误计数器值大于或等于 256 时,节点为"总线关闭"。

11. 当发送错误计数器值和接收错误计数器值都小于或等于 127 时,"错 误被动"的节点重新变为"错误主动"。

12. 在总线监视到 128 次出现 11 个连续"隐性"位之后,"总线关闭"的节点可以变成"错误主动"(不再是"总线关闭"),它的错误计数值也 被设置为 0。

备注1:

一个大约大于96的错误计数值显示总线被严重干扰。最好能够预先采 取措施测试这个条件。

备注 2:

起动/睡眠:如果起动期间内只有1个节点在线,以及如果这个节点发送一些报文,则将不会有应答,并检测到错误和重复报文。由此,节点会变为"错误被动",而不是"总线关闭"。



互联互通▼

【产品应用】 飞行汽车专用的CAN(FD)数据记录终端

ZLG 致远电子 2024-04-30 11:40:40

飞行汽车的飞行安全永远是第一要素,搭载致远电子专用的多通道 CAN(FD)数据记录终端CANFDDTU-300ER,时刻记录存储并且分析各 类飞行数据,为飞行汽车的硬件研发、实体飞行测试提供数据支撑。

飞行汽车:未来出行的新纪元

随着科技的飞速发展,人类对于出行方式的探索从未停止。从马车到 汽车,从火车到飞机,每一次革新都极大地改变了我们的生活方式。现在, 一个新的概念——飞行汽车,正逐渐从科幻小说走向现实,预示着未来出 行的新纪元。

在国家低空经济的政策支持下,国内飞行汽车厂商大力发展飞行汽车 技术,开启群雄争霸的时代,各个厂商相继拿到飞行牌照,相信很快"空 中出租车"就可以正式投入运营。



飞行汽车: 我CAN行

在飞行汽车中,CAN(FD)总线通讯主要用于连接各种电子控制单元 (ECU),如飞行控制器、电子速度控制器(ESC)、传感器、执行器和其他外 围设备。飞行模式下,飞行控制器实时接收速度、高度、姿态等各种传感器 数据;地面行驶模式下,各类传感器还需要支持飞行汽车在地面行驶时的 制动、转向和动力分配等操作。CAN(FD)总线通讯系统在确保飞行汽车安全、 高效运行中扮演着至关重要的角色。



CANFDDTU-300ER,飞行汽车专用的多通道CAN(FD)数 据记录终端

针对飞行汽车数据的记录、存储和分析,致远电子最新推出飞行汽车 专用多通道 CAN(FD)-bus 数据记录终端 CANFDDTU-300ER,默认搭载较强 抗震性能的 64GB EMMC 存储芯片,且支持 WiFi 数据导出功能,用户无需 拆卸即可读取存储的数据,运行中还可通过无线 4G 数据传输数据上云实时 监控,自带北斗 /GPS 定位功能,轻松实现远程定位。



飞行汽车正在"飞"进现实,致远电子一直致力于飞行汽车相关领域 的创新与研发,旨在通过尖端技术,助力国家低空经济发展!



【新品发布】 新一代高性能小巧型ZigBee终端采集器全新上市

ZLG 致远电子 2024-04-01 11:39:49



选型表

项目		描述
	协议	ZLG Mesh
ZigBee	频段	2400~2483.5MHz
	发射功率	-30 ~ 20dBm
	天线	吸盘天线 / 棒状天线 (2.4GHz)
RS-485	波特率	2400~115200bps
DI/DO	DI 输入电平	低电平 0~2.0V, 高电平 2.5~24V
DI/DO	DO 负载 (Max)	最大负载电压 30V,最大负载电流 5A
ADC	采集精度	12 位 ADC,最大可识别电压 36V
按键	DEF	恢复出厂设置
	JOIN	无线组网功能
指示灯	状态指示灯	电源指示灯,RS-485 通信功能灯, ZigBee 通信功能灯
	工作温度	-40°C ~85℃
温度	存储温度	-40°C ~85℃
海府	工作湿度	10%~95%RH(无凝霜)
迎友	存储湿度	10%~95%RH(无凝霜)
(#*中	供电电压	DC 9~36V
洪电	工作电流 (Max)	9V/200mA
体积	尺寸大小	84.0×63.0×30.0(mm)







ZLGi微文摘

互联互通▼



多种通信方式动态可切换



数据加密,保障传输安全



工业级品质支撑参数



强大的Mesh组网能力



远距离通信



深入调研应用场景,切实解决客户痛点

主机支持查询已入网节点 mac 地址,帮助客户快速定位异常节点



如需了解更多产品详情,可填写申请表单, 我们会有专人与您联系。



深入调研应用场景,切实解决客户痛点



ZLGi微文摘

互联互通▼

【新品发布】低功耗、高性能国产ZigBee模组 ZM82系列全新上市

2024-04-22 11:40:39



选型表

型号	ZM82P0S22E	ZM82P0S22P	ZM82P2S22E	ZM82P2S22P	ZM82P2S22S	
组网协议	FastZigBee					
工作频段	2402~2480MHz					
发射功率	12dBm	12dBm	21dBm	21dBm	21dBm	
接收 灵敏度	-98dBm	-98dBm	-102dBm	-102dBm	-102dBm	
天线形式	IPEX 接口	PCB 天线	IPEX 接口	PCB 天线	邮票孔接口	
通信接口	UART	UART	UART	UART	UART	
工作温度	-40~+85°C	-40~+85°C	-40~+85°C	-40~+85°C	-40~+85°C	
产品尺寸	17.0×12.0×2.1	17.0×12.0×2.1	19.0×13.5×2.3	19.0×13.5×2.3	19.0×13.5×2.3	
包装类型	卷带	卷带	卷带	卷带	卷带	
最小包装数量	1800	1800	850	850	850	

产品应用方案



FastZigBee快速自组网

ZM82 模组采用致远自研的 FastZigBee 组网协议,可构建多种网络拓扑结构,支持快速自组网,使用简便,布局灵活,传输效率高、网络性能稳定。



串口透明传输

FastZigbee 采用全透明传输方式,发送方发送的数据和接收方接收的 数据内容和长度完全一致,相当于一段无形的传输线。用户可以在此基础上, 创建自己需要的协议格式,快速灵活。





定时休眠唤醒

ZM82 模组支持节点定时休眠唤醒,客户可灵活设置唤醒间隔,实现按 需远程唤醒主设备。





强大的无线配置工具

设备配置、距离测试、空中升级,指令生成 / 解析,你所需的功能应有 尽有。





ZLGi微文摘

互联互通▼

天线一体化设计

专业射频团队精心打造,高性能 PCB 天线与 RF 连接器一体式设计,小体积也可以高性能。



细节方能"致远"

ZM82 模组经过致远专业射频团队精心打磨,具备优秀的射频性能,空 旷环境下,通信距离可达 3KM 以上。



300nA超低功耗

国产 ZigBee 芯片方案,两种低功耗模式,最低功耗 300nA,低功耗需 求轻松应对。



工业级品质



行业应用





【产品详解】新一代工业智能利器 ——GXF222-4G工业数采边缘网关

ZLG 致远电子 2024-04-15 11:36:54

在数字化浪潮中,广州致远电子股份有限公司倾力打造的 GXF222-4G 工业级边缘网关,以其强大的 4G 通信、GPS 定位、云平台以及数据 存储功能,成为工业领域数据采集、远程管理与智能化升级的全新利器。

全面接口集成,满足多样需求

GXF222-4G 工业级边缘网关 凭借其多样化的接口设计,如 CAN、RS-485、DI、DO、ADC 等,轻松满足工业现场复杂设备的接入需求,确保数 据采集与传输的无缝对接。

- 2 路 CAN\CANFD 通信接口,支持波特率: 40k~5Mbps。
- 2路 RS-485 串口通信,支持波特率: 2400~230400bps。
- 4 路 DI 接口,可作为数字输入接口,低电平电压范围 0~1V,高电 平电压范围 2.0~24V。
- 2 路 DO 接口,可作为数字输出接口,最大负载电流为 5A,最大负载直流电压为 30VDC、最大负载交流电压为 250VAC。
- 4 通道 ADC 数据采集,用于采集模拟信号,ADC 最大可采样电压为 22V,分辨率为 12 位、5M 采样率。



4G通信加持,实时数据传输

GXF222-4G 网关凭借其强大的 4G 通信能力,无论您身处何地,都能确 保数据的实时、稳定传输,为您带来前所未有的便捷与高效体验。无论您 需要远程监控、数据传输还是设备控制,GXF222-4G 网关都能轻松满足, 真正打破了地域限制,让工业管理变得更加简单高效。

网关采用了 Cat 4 高速传输技术,具备卓越的无线通信性能:

- 最大下行速率 150 Mbps,最大上行速率 50 Mbps。
- 支持 LTE-FDD: B1/B3/B5/B8。
- 支持 LTE-TDD: B34/B38/B39/B40/B41。
- 支持 WCDMA: B1/B5/B8。
- 支持 GSM: EGSM900/DCS1800。
- 最大支持 3GPP Rel-9 non-CA Cat 4 FDD 和 TDD。
- 支持 1.4/3/5/10/15/20 MHz 射频带宽。



GPS精准定位,掌握设备动态

GXF222-4G 网关集成了先进的 GPS 定位功能,能够实时、精确地定位 设备的具体位置,为设备管理与调度提供强大的支持。无论是进行设备巡检、 故障排查还是资产管理,都能轻松掌握设备的动态信息,进而显著提升管 理效率。

- 支持星系: GPS: L1 C/A、L1 C/A、BeiDou: B11。
- 水平位置精度: 2.0 m。
- 冷启动: 30 s 、热启动: 2 s。
- 更新速率:默认:1Hz,最高可达5Hz。
- 1PPS 精度:典型精度:100 ns、脉冲宽度:100 ms。
- 速度精度: 0.1 m/s。
- 加速度精度: 0.1 m/s2。
- 动态特性:最高海拔: 18000 m 、最大速度: 515 m/s 、加速度: 4g。

ZLGi微文摘

互联互通 ▼

云平台功能,实现智能化管理

GXF222-4G 网关完美支持云平台接入,用户仅需通过云平台,即可轻松 实现远程监控、数据分析、设备控制等一系列操作,从而迈向真正的智能化 管理新纪元。云平台不仅提供了强大的功能支持,更通过丰富的数据可视化 功能,为用户呈现设备运行状况的直观图像,助力用户做出更为明智的决策。

- 云网关在线监督。
- 批量配置和升级。
- 自定义设备名称。
- 日志查看。
- 自动注册设备。
- 组态大屏。
- 平台私有化部署,客户可选用致远 ZWS 云平台部署到自有服务器中, 自主独立管理。

数据存储功能,确保数据安全与追溯

GXF222-4G 网关内置了超大的存储空间,具备实时存储各类采集数据的 强大能力。这一设计不仅确保了数据的安全稳固,有效防止数据丢失或被非 法篡改,而且支持数据的回溯与深度分析,从而为后续的故障排查、系统性 能优化等工作提供了坚实的数据支持。

- 支持双路 CAN 同时存储,实现数据的并行处理。
- 在高达 2500 帧 / 秒的传输速度下,依然能够确保数据的完整性和准确性,实现不丢数据。
- 支持 32G 超大存储空间,满足长期、大量数据的存储需求。



电气隔离保护与宽温、宽压工作,确保稳定运行

GXF222-4G 网关的 CAN 接口、RS-485、DI、DO 接口均集成了电气隔离 保护模块,这一设计能够有效抵御瞬间过流 / 过压对设备造成的潜在损害。 同时,其宽温、宽压的工作范围使得该网关能够在各种极端环境条件下稳定 运行,确保系统的持续可靠运行。

- 工业级系统:采用高稳定嵌入式实时操作系统,内置看门狗设计。
- 工业级电源: 9~36V 宽压输入,支持防反接、过流保护。
- 工业级抗干扰:
 防静电:接触± 8kV,空气±15KV;
 群脉冲:电源±2kV,通讯线±1kv;
 浪涌:共模±2kV、RF传导骚扰抗扰度:3V。
 工业级温湿度:40~+80C 宽温工作、10%~95% 湿度范围。



总结

GXF222-4G 工业级边缘网关凭借 4G 高速通信、GPS 精确定位、云平台 管理、本地数据存储以及接口集成、电气隔离保护和宽温宽压工作等优势, 为工业领域的数据采集、远程管理与设备定位提供了前所未有的支持。如果 您正在寻找一款高性能、高可靠性的网关产品,GXF222-4G 工业级边缘网关 将是您实现智能化管理、提升工作效率的明智之选!



【解决方案】电机控制领域如何实现CANopen设备 接入EtherCAT主站控制器

ZLG 致远电子 2024-04-02 11:46:08

EtherCAT 是目前全球最快的工业以太网通讯技术,在电机控制领域具 有广泛的应用。当 CANopen 从站产品,需要接入 EtherCAT 主控系统, 如何解决?

EtherCAT 是目前全球最快的工业以太网通讯技术,在电机控制领域 具有广泛的应用,典型的主站产品中 PLC、运动控制器等产品都有大量 EtherCAT 通讯产品.

此外 CANopen 是一种传统常见的工业自动化通讯协议,典型的从站产 品如电机驱动器、分布式 I/O、执行器、传感变送器等采用该协议。



当控制系统主站采用EtherCAT通讯,缺少CAN接口,同时 需要接入CANopen的电机驱动、扩展I/O等从站设备,遇 到这种情况需要如何解决呢?

我们都知道 CANopen 信号是无法直接接入 EtherCAT 通讯系统的,如 果采用自己开发解决通讯问题,需要花费大量的时间及人工成本去熟悉协 议代码及转换编程工作。

为了快捷解决以上问题,我们采用 PXB-8021M 协议转换器,将 CANopen 从机产品信号转换为 EtherCAT 信号,轻松实现 EtherCAT 通讯系 统与各 CANopen 从站节点的连接,信号转换传输稳定可靠。



产品介绍



PXB-8021M 产品是广州致远电子股份有限公司推出的 EtherCAT 转 CANopen 主的工业现场总线协议转换器。

本产品具有 2 路 EtherCAT 从站接口、1 路 CAN FD、1 路标准 10/100M 以太网接口及 1 路 USB Type-C 接口,可实现 CANopen 设备接入 EtherCAT PCL、控制器等主控系统。PXB-8021M 协议转换器提供简单好用的上位机 配置工具,仅需一连一配,轻松实现 CANopen 网络到 EtherCAT 网络间的 无缝转换。

致远微电子芯片 ▼

【新品发布】 4合1,300mA带Watchdog低压差线性稳压器

ZLG 致远电子 2024-04-23 12:07:03

MCU 的稳定运行需 LDO 供电,可靠运行则依靠 Watchdog IC。那么 这两颗芯片的功能是否能用一颗芯片来实现? ZL6303 正是广州致远 微电子为解决这一问题而设计。

功能描述

ZL6303 是广州致远微电子有限公司设计的一款带有 Watchdog 功能的最 大输出 300mA 低压差线性稳压器。特别适用于 MCU 需要 LDO 供电同时需要外 置 Watchdog 的应用场景。一般为了保证 MCU 工作的可靠外部需要 2 颗 IC, 即 LDO+Watchdog。ZL6303 采用 "4 合 1" 集成技术,整合了 LDO、Watchdog、MR 和电源电压监控功能于单一 IC 之中,确保 MCU 的高效稳定运行。





图1 内部电路示意图

- **手动复位功能:** 内部有 52K 的上拉电阻。当脚被拉低到地时,手动复位功能开启;
- **禁能功能:** 内部有 52K 的上拉电阻。当脚被拉低到地时,Watchdog 的 禁能功能开启,隔绝 MCU 的喂狗信号输入。这一功能特别适用于大批量 生产时 MCU 烧写程序时的控制;
- **内置掉电快速放电电路:**利用输出电容残存电荷快速放电,可以极大提 高电子系统的可靠性;
- 完善的保护功能: 欠压保护、过流保护、过温保护和短路保护。
- 1. 电源输入电压小于 2.15V (典型值) , IC 内部欠压锁定电路将禁止输出;
- 2. 当输出短路或者输出电流超过过流保护阈值, IC 将进入过流保护状态, 限制电流输出;
- 3. 当芯片温度过高时,芯片将过温关断,温度下降到一定值时,芯片将 重新启动。
- 低静态电流、良好的负载调整率和线性调整率: 能在输入、输出电压差 200mV 的情况下提供 300mA 的输出电流;
- **封装:** ZL6303 封装不含铅、卤素和 BFR, 符合 RoHS 标准。ESD 等级(人 体模型)达 4KV,确保高可靠性。封装形式可供选择: SOP8、DFN2X3-08-C。



图2 管脚信息

ZL6303 各管脚的详细功能描述。

表1 管脚描述

管脚名称	产品封装	描述	
	SOP8 DFN2X3-08-C		
VSS	2	2	芯片接地端
VIN	1	1	电压输入端
VOUT	8	8	电压输出端
WEN	4	4	看门狗禁能
MR	3	3	手动复位
WDI	5	5	喂狗输入
WDO	7	7	看门狗输出
NC	6	6	NC

设计实例

如图 3 是 ZL6303 的典型应用电路图。



ZL6303 内置 Watchdog 手动复位功能,可通过轻触开关 SW1 与下拉电阻 R1 来实现外部手动复位,R1 建议选择小于 20KΩ 电阻;可通过轻触开关 SW2 与下拉 电阻 R2 来实现隔绝 MCU 对 Watchdog 喂狗动作,R2 建议选择小于 20KΩ 电阻。







ZLG致远电子官方微信

广州致远电子股份有限公司

 更多详情请访问
 欢迎拨打全国服务热线

 www.zlg.cn
 400-888-4005