

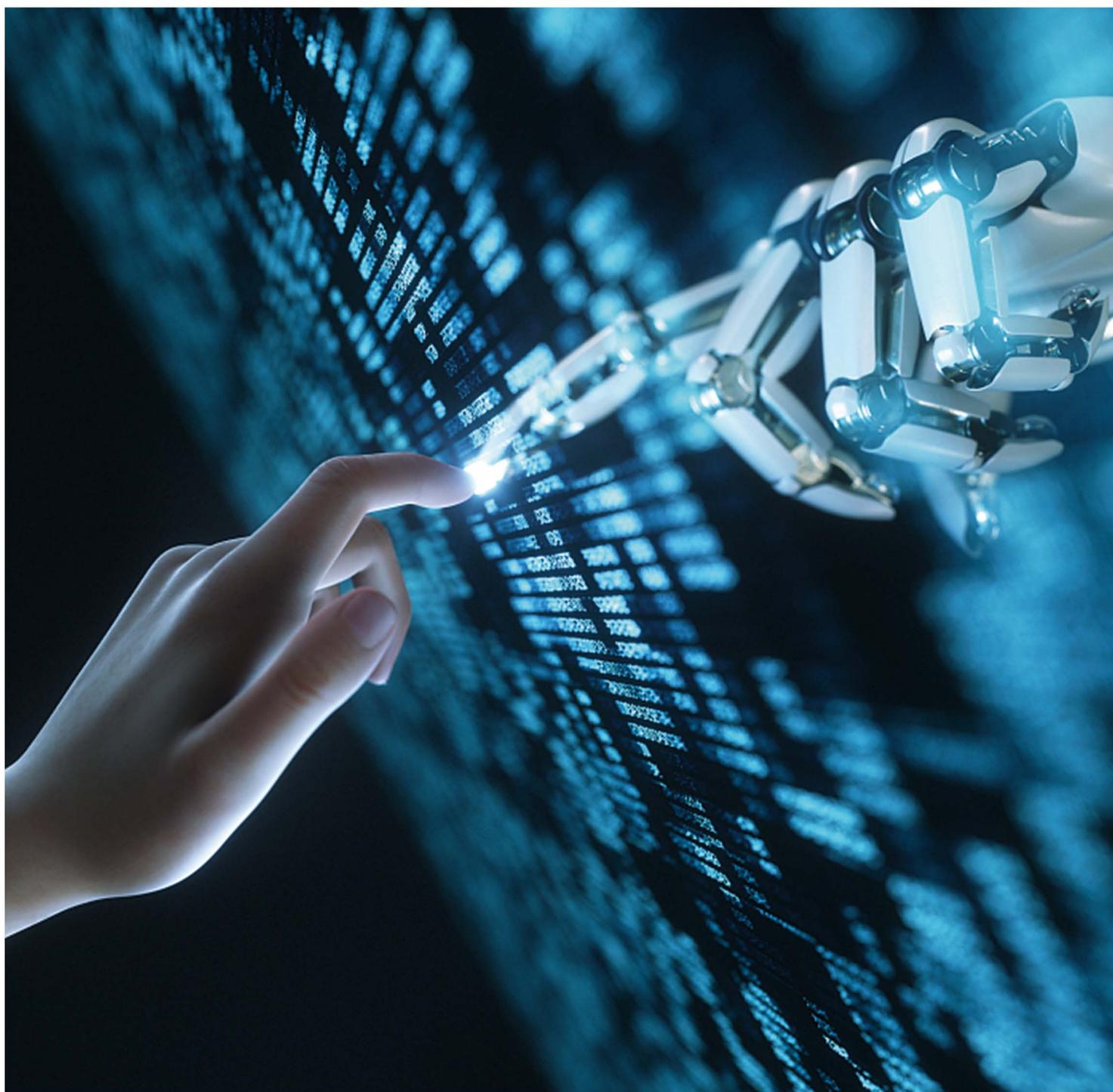
ZLG 致远电子

# 微文摘

ZLG MICRO DIGEST

2024/3 第3期

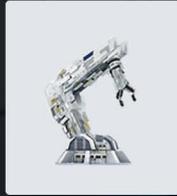
月刊



# ZMC900E

## 国产EtherCAT主站控制器

为满足智能制造设备应用控制而生



工业机器人



协作机器人



Delta机器人



机械手



SCARA机器人



### 规格参数

| 名称         | ZMC900E   |
|------------|---|
| 主控芯片       | 2.0GHz 4+1 个 Arm® Cortex®-A55 64 位微控制器, 3 核 Cortex®-R5F |
| 内存         | LPDDR4 (4GB)  |
| 存储器        | eMMC(8GB)   |
| 铁电存储       | FRAM 32KB   |
| HDMI       | 支持最高分辨率 1920*1080                                       |
| LVDS       | 支持最高分辨率 1920*1080                                       |
| USB 接口     | 2 路 USB3.0 Host   |
| EtherCAT 口 | 主站, ms 周期任务抖动 $\pm 1.5\mu\text{s}$ 以内                   |
| 网口         | 3 路, 支持 TCP/IP、ModbusTCP、OPC UA 协议                      |
| MiniPCle   | 支持 4G/Wi-Fi   |
| M.2 接口     | 支持  |
| CAN-FD     | 1 路, 电气隔离、支持 CANOpen 协议                                 |
| RS485      | 1 路, 电气隔离、通讯参数可配、Modbus RTU 协议 / 自由协议                   |
| 系统 I/O     | 4 路数字量输入, 4 路数字量输出, 最大支持电流 300mA                        |

| 名称        | ZMC900E   |
|-----------|---|
| 用户 I/O 输入 | 16 路数字量输入或 12 路数字量输入 + 1 路正交编码器; NPN 型<br>正交编码器推荐型号 OMRON/E6B2-CWZ6C/2000P/2M;<br>I/O 输入 最高响应频率 4KHz, 导通电流 $\geq 4.2\text{mA}(15\text{V})$<br>典型值 6.9mA |
| 用户 I/O 输出 | 16 路数字量输出, NPN 型, 最大支持电流 300mA, (24v, 吸入)   |
| 编码器       | 1 路正交编码器<br>(推荐型号 OMRON/E6B2-CWZ6C/2000P/2M),<br>计数长度 32 位有符号, 2MHz   |
| ADC       | -   |
| DAC       | -   |
| POE       | 可选 30W POE 配件   |
| 其他接口      | -   |
| RTC 实时时钟  | 年月日时分秒  |
| 看门狗       | 板载独立硬件看门狗   |
| 掉电检测      | 支持掉电检测, 5S 掉电续航   |
| 供电电源      | DC24V $\pm 5\%$ , 电气隔离, 支持掉电检测  |
| 工作温度      | -40° C ~ 70° C  |

# CONTENTS

# 目录

## 技术平台

### EsDA 平台

- 【技术分享】AWTK 开源串口屏开发 (11) - 天气预报····· 04
- 【技术分享】AWTK 开源串口屏开发 (12) - 记事本应用····· 06
- 【技术分享】AWTK 开源串口屏开发 (13) - 计算器应用····· 08
- 【从 0 开始创建 AWTK 应用程序】编译应用到 RTOS 平台····· 11
- 【技术分享】AWTK 开源串口屏开发 (14) - 界面重用····· 13
- 【技术分享】AWTK 开源串口屏开发 (15) - 通过 MODBUS 访问远程数据····· 15

### ZWS 云平台

- 【产品应用】EM 储能网关 & ZWS 智慧储能云应用 (1) — 使用介绍····· 17
- 【产品应用】ZWS 云平台应用 (3) - 设备基于 DTU 接入····· 18

## 边缘计算

### 行业控制器

- 【新品发布】ZMC900E 国产高性能 EtherCAT 总线控制器····· 19
- 【应用案例】ZMC900E 国产 EtherCAT 主站控制器 — 电源口传导骚扰整改····· 21
- 【产品应用】脱胎换骨” — 插针机运动控制····· 23
- 【产品应用】EtherCAT 主站控制器系统实时性测试····· 25

## 互联互通

### CAN-bus 总线

- 【技术分享】CAN 总线波形中为什么 ACK 电平偏高?····· 27

### 无线通讯

- 【新品发布】新一代工业数采边缘网关 GXF222-4G 全新上市····· 29
- 【技术分享】深入了解 LoRa 技术① — LoRa 产品的信道划分····· 32

### 接口与协议转换

- 【新品发布】仅需一连一配，轻松实现各类设备接入 EtherCAT 总线系统····· 33
- 【技术分享】CAN 总线的传输距离与采样点、传输时延之间的关系，你弄明白了吗?····· 35
- 【应用方案】高端制造设备将 DeviceNet 信号转为 EtherCAT····· 37
- 【应用方案】高端制造设备将 Modbus 节点信号接入 EtherCAT 总线系统····· 38
- 【解决方案】汽车测试系统如何实现 CANFD 到 EtherCAT 的协议转换····· 39

## 感知控制

### 电源与隔离

- 【产品应用】SM4500 系列隔离收发芯片应用集锦····· 40

## 【技术分享】

# AWTK 开源串口屏开发(11) - 天气预报

ZLG 致远电子 2024-03-01 11:39:50

AWTK 串口屏内置了 XML/JSON/INI 等各种数据文件的模型，并支持用 HTTP/HTTPS 从网络获取数据。不用编写一行代码，即可实现天气预报、股票行情、航班查询和快递查询等功能。

天气预报是一个很常用的功能，在很多设备上都有这个功能。实现天气预报的功能，不能说很难但是也绝不简单，首先需要从网上获取数据，再解析数据，最后更新到界面上。

在 AWTK 串口屏中，内置了 XML/JSON/INI 等各种数据文件的模型，并支持用 HTTP/HTTPS 从网络获取数据。所以实现天气预报非常简单，不用编写一行代码即可实现天气预报的功能。而且用同样的方式，也可以实现其它功能，比如：股票行情、新闻、公交查询、火车查询、航班查询、快递查询等等。

这里以 <http://t.weather.sojson.com> 提供的接口为例，来实现一个显示天气信息的应用程序。这个是免费的，无需注册的 API，当然也有些限制，在实际工作中，你可以换成自己的接口。

它返回的数据是 JSON 格式的，如下所示：

```
{
  "message": "success 感谢又拍云 (upyun.com) 提供 CDN 赞助 ",
  "status": 200,
  "date": "20240101",
  "time": "2024-01-01 08:13:13",
  "cityInfo": {
    "city": "天津市 ",
    "citykey": "101030100",
    "parent": "天津 ",
    "updateTime": "08:01"
  },
  "data": {
    "shidu": "86%",
    "pm25": 57.0,
    "pm10": 93.0,
    "quality": "良 ",
    "wendu": "-7",
    "ganmao": "极少数敏感人群应减少户外活动 ",
    "forecast": [
      .
    ],
    "yesterday": {
      "date": "31",
      "high": "高温 1°C ",
      "low": "低温 -3°C ",
      "ymd": "2023-12-31",
```

```
"week": "星期日 ",
"sunrise": "07:30",
"sunset": "16:57",
"aqi": 35,
"fx": "北风 ",
"fl": "2级 ",
"type": "晴 ",
"notice": "愿你拥有比阳光明媚的心情 "
}
}
}
```

AWTK 串口屏中的 json 模型，支持用下面的语法从网络获取数据，它会自动解析 JSON 数据，并生成一个模型，通过路径可以引用模型中的数据。

```
json(url=http://t.weather.sojson.com/api/weather/city/101030100)
```

网上有很多天气预报的接口，这里只是举个例子，如果你有自己的接口，可以用自己的接口。

### 1. 功能

不用编写代码，实现天气预报。

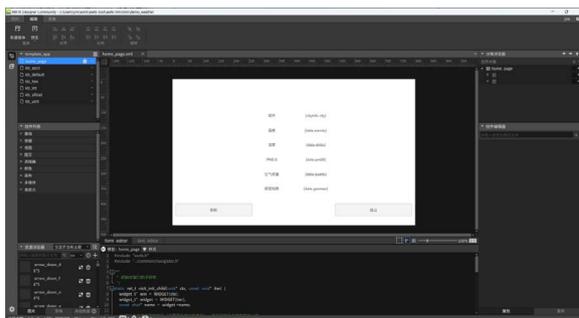
### 2. 创建项目

从模板创建项目，将 hmi/template\_app 拷贝 hmi/weather 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 weather 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

### 4.1 城市

| 绑定属性         | 绑定规则            | 说明                                       |
|--------------|-----------------|--|
| v-data:value | {cityInfo.city} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.2 温度

| 绑定属性         | 绑定规则         | 说明                                       |
|--------------|--------------|--|
| v-data:value | {data.wendu} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.3 湿度

| 绑定属性         | 绑定规则         | 说明                                       |
|--------------|--------------|--|
| v-data:value | {data.shidu} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.4 PM2.5

| 绑定属性         | 绑定规则        | 说明                                       |
|--------------|-------------|--|
| v-data:value | {data.pm25} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.5 空气质量

| 绑定属性         | 绑定规则           | 说明                                       |
|--------------|----------------|--|
| v-data:value | {data.quality} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.6 感冒指数

| 绑定属性         | 绑定规则          | 说明                                       |
|--------------|---------------|--|
| v-data:value | {data.ganmao} | 不同的 JSON API 需要使用不同的路径, 请根据具体的 JSON 数据填写 |

### 4.7 刷新按钮

- 将刷新按钮的点击事件绑定到 reload 命令。添加自定义的属性 v-on:click, 将值设置为 {reload}。

| 绑定属性       | 绑定规则     | 说明   |
|------------|----------|--|
| v-on:click | {reload} | reload 命令是内置的命令, 用于重新加载持久化的配置, 命令要用英文大括号括起来。 |

### 4.8 窗口模型

- 指定窗口的模型为 json, url 为:  
http://t.weather.sojson.com/api/weather/city/101030100

| 绑定属性    | 绑定规则   | 说明                                    |
|---------|--|---------------------------------------|
| v-model | json(url=http://t.weather.sojson.com/api/weather/city/101030100) | 不同的 JSON API 需要使用不同的 URL, 可以换成自己的 URL |

## 5. 初始化数据

无

## 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:



点击刷新按钮, 可以重新加载数据, 但是通常界面没有变化, 因为天气数据没有变化。

## 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\_weather。



# 【技术分享】

# AWTK 开源串口屏开发(12) - 记事本应用

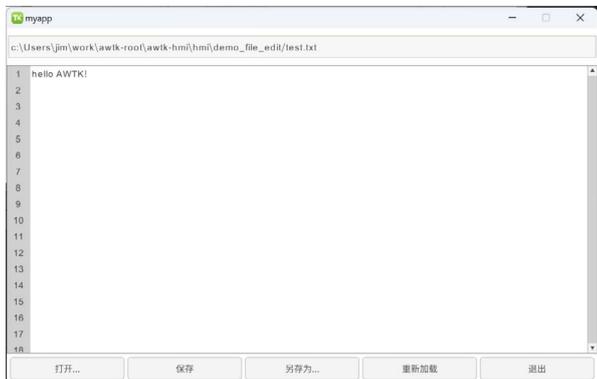
ZLG 致远电子 2024-03-08 11:35:46

记事本是一个常用的应用程序，在 AWTK 串口屏中，内置文件模型和文件选择对话框，无需编写一行传统的代码，即可实现一个简单的记事本应用程序。

记事本是一个很常用的应用，用来编辑和查看文本文件非常方便。在传统的串口屏中，开发一个记事本应用，即使可能，也是非常麻烦的事情。在 AWTK 串口屏中，内置文件模型和文件选择对话框，实现一个简单的记事本，不需要编写代码，设计好界面，添加绑定规则就好了，非常简单。

## 1. 功能

不用编写代码，实现记事本应用。



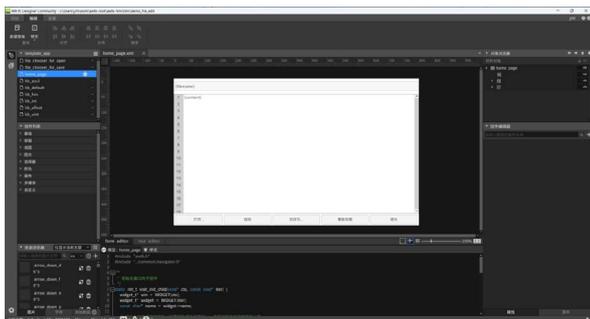
## 2. 创建项目

从模板创建项目，将 hmi/template\_app 拷贝 hmi/notepad 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 notepad 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

### 4.1 文件名

用 edit 控件显示文件名，将文件名的文本属性绑定到 filename 变量。添加自定义的属性 v-data:value，将值设置为 {filename}。

| 绑定属性         | 绑定规则       | 说明                       |
|--------------|------------|--------------------------|
| v-data:value | {filename} | filename 是内置的变量，用于保存文件名。 |

### 4.2 文件内容

用 mledit 控件显示文件内容，将文件内容的文本属性绑定到 content 变量。添加自定义的属性 v-data:value，将值设置为 {content}。

| 绑定属性         | 绑定规则      | 说明                       |
|--------------|-----------|--------------------------|
| v-data:value | {content} | content 是内置的变量，用于保存文件内容。 |

### 4.3 打开文件

- 将打开按钮的点击事件绑定到 browse 命令。添加自定义的属性 v-on:click，将值设置为 {browse, Args=open(filter=' .txt.c.cpp.md' ,title= 'Text File' )}。

| 绑定属性       | 绑定规则   | 说明                                   |
|------------|--|--------------------------------------|
| v-on:click | {browse, Args=open(filter=' .txt.c.cpp.md' , title= 'Text File' )} | browse 命令是内置的命令，参数 open 用于打开文件选择对话框。 |

### 4.4 保存文件

- 将保存按钮的点击事件绑定到 save 命令。添加自定义的属性 v-on:click，将值设置为 {save}。

| 绑定属性       | 绑定规则   | 说明                    |
|------------|--------|-----------------------|
| v-on:click | {save} | save 命令是内置的命令，用于保存文件。 |

### 4.5 另存为文件

- 将另存为按钮的点击事件绑定到 browse 命令。添加自定义的属性 v-on:click，将值设置为 {browse, Args=saveas(filter=' .txt.c.cpp.md' ,title= 'Text File' )}。

| 绑定属性       | 绑定规则   | 说明                                     |
|------------|--|--|
| v-on:click | {browse, Args=saveas(filter=' .txt.c.cpp.md' , title= 'Text File' )} | browse 命令是内置的命令，参数 saveas 用于打开文件选择对话框。 |

### 4.6 重新加载

- 将重新加载按钮的点击事件绑定到 reload 命令。添加自定义的属性 v-on:click，将值设置为 {reload}。

| 绑定属性       | 绑定规则     | 说明   |
|------------|----------|--|
| v-on:click | {reload} | reload 命令是内置的命令，用于重新加载持久化的配置，命令要用英文大括号括起来。 |

#### 4.8 窗口模型

- 指定窗口的模型为 file，路径为 \${app\_dir}/test.txt，自动加载文件。

| 绑定属性    | 绑定规则  | 说明   |
|---------|---|--|
| v-model | file(path=\${app_dir}/test.txt, auto_load=true) | file 是内置的模型，用于保存文件内容，path 是文件路径，auto_load=true 表示自动加载文件。 |

#### 5. 初始化数据

无

#### 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：



#### 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\_notepad。



### ZMP110X系列

👉 点击购买

# 【技术分享】

# AWTK 开源串口屏开发(13) - 计算器应用

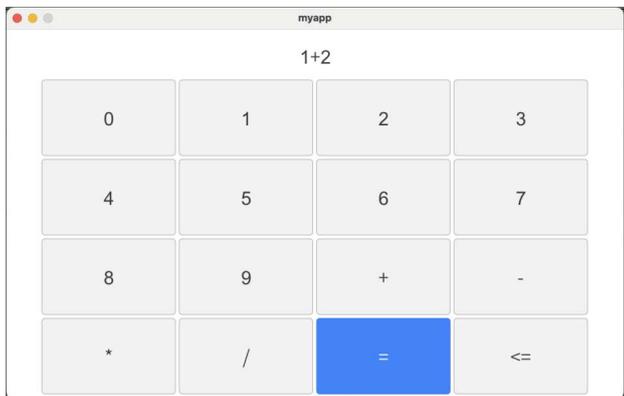
ZLG 致远电子 2024-03-15 11:35:59

计算器是一个常见的应用程序，在 AWTK 串口屏中，利用 fscript 表达式计算函数，无需编写一行传统的代码，即可实现一个简单的计算器应用程序。

## 1. 功能

计算器是一个很常见的应用，比如在电子秤中，可能就需要这样一个应用。在计算器中会用到一些有意思的知识点，比如嵌入键盘，在数字输入或密码输入也会用到。

这里我们实现一个简单的计算器，不需要编写代码，设计好界面，添加绑定规则即可。



在这里例子中，模型（也就是数据）里只有一个 expr:

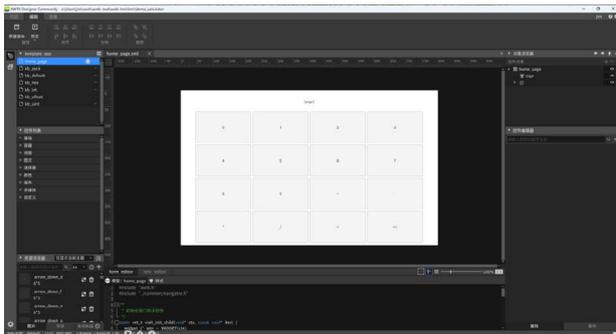
| 变量名  | 数据类型 | 功能说明     |
|------|------|----------|
| expr | 字符串  | 保存表达式或结果 |

## 2. 创建项目

从模板创建项目，将 hmi/template\_app 拷贝 hmi/calculator 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 calculator 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

### 4.1 表达式（结果）

用 edit 控件显示表达式，将表达式的文本属性绑定到 expr 变量。添加自定义的属性 v-data:value，将值设置为 {expr}。

| 绑定属性         | 绑定规则   | 说明                   |
|--------------|--------|----------------------|
| v-data:value | {expr} | expr 是内置的变量，用于保存表达式。 |

### 4.2 输入数字 0

将 0 按钮的点击事件绑定到 fscript 命令。添加自定义的属性 v-on:click，将值设置为 {fscript, Args=set(expr, expr+ '0') }。

| 绑定属性       | 绑定规则                                  | 说明               |
|------------|---------------------------------------|------------------|
| v-on:click | {fscript, Args=set(expr, expr+ '0') } | 在表达式后面追加字符串 '0'。 |

### 4.3 输入数字 1

将 1 按钮的点击事件绑定到 fscript 命令。添加自定义的属性 v-on:click，将值设置为 {fscript, Args=set(expr, expr+ '1') }。

| 绑定属性       | 绑定规则                                  | 说明               |
|------------|---------------------------------------|------------------|
| v-on:click | {fscript, Args=set(expr, expr+ '1') } | 在表达式后面追加字符串 '1'。 |

### 4.4 输入数字 2

将 2 按钮的点击事件绑定到 fscript 命令。添加自定义的属性 v-on:click，将值设置为 {fscript, Args=set(expr, expr+ '2') }。

| 绑定属性       | 绑定规则                                  | 说明               |
|------------|---------------------------------------|------------------|
| v-on:click | {fscript, Args=set(expr, expr+ '2') } | 在表达式后面追加字符串 '2'。 |

#### 4.5 输入数字 3

将 3 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '3')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '3')}</code> | 在表达式后面追加字符串 '3'。 |

#### 4.6 输入数字 4

将 4 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '4')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '4')}</code> | 在表达式后面追加字符串 '4'。 |

#### 4.7 输入数字 5

将 5 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '5')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '5')}</code> | 在表达式后面追加字符串 '5'。 |

#### 4.8 输入数字 6

将 6 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '6')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '6')}</code> | 在表达式后面追加字符串 '6'。 |

#### 4.9 输入数字 7

将 7 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '7')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '7')}</code> | 在表达式后面追加字符串 '7'。 |

#### 4.10 输入数字 8

将 8 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '8')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '8')}</code> | 在表达式后面追加字符串 '8'。 |

#### 4.11 输入数字 9

将 9 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '9')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '9')}</code> | 在表达式后面追加字符串 '9'。 |

#### 4.12 输入运算符 +

将 + 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '+')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '+')}</code> | 在表达式后面追加字符串 '+'。 |

#### 4.13 输入运算符 -

将 - 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '-')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '-')}</code> | 在表达式后面追加字符串 '-'。 |

#### 4.14 输入运算符 \*

将 \* 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '*')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '*')}</code> | 在表达式后面追加字符串 '*'。 |

#### 4.15 输入运算符 /

将 / 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, expr+ '/')}`。

| 绑定属性                    | 绑定规则  | 说明               |
|-------------------------|---|------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, expr+ '/')}</code> | 在表达式后面追加字符串 '/'。 |

#### 4.16 输入 = 计算结果

将 = 按钮的点击事件绑定到 `fscript` 命令。添加自定义的属性 `v-on:click`，将值设置为 `{fscript, Args=set(expr, str(eval(expr)))}`。

| 绑定属性                    | 绑定规则  | 说明                          |
|-------------------------|---|-----------------------------|
| <code>v-on:click</code> | <code>{fscript, Args=set(expr, str(eval(expr)))}</code> | eval 计算表达式结果，str 将结果转换成字符串。 |

#### 4.17 删除键 <=

将 <= 按钮的点击事件绑定到 fscript 命令。添加自定义的属性 v-on:click，将值设置为 {fscript, Args=set(expr, substr(expr,0,-1))}。

| 绑定属性       | 绑定规则  | 说明            |
|------------|---|---------------|
| v-on:click | {fscript, Args=set(expr,substr(expr,0,-1))} | 删除表达式中最后一个字符。 |

#### 4.18 窗口模型

指定窗口的模型为 dummy(expr=' '), dummy 是一个简单的模型，只能存放数据，而且是控件或窗口私有的。expr=' ' 表示将 expr 初始化为空字符串。

### 5. 初始化数据

无

### 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：



# 【从0开始创建AWTK应用程序】 编译应用到RTOS平台

ZLG 致远电子 2024-03-20 11:38:01

AWTK 是基于 C 语言开发的跨平台 GUI 框架。本系列文章介绍如何从 0 开始创建 AWTK 应用程序，包括搭建开发调试环境、使用 AWTK 创建 Hello 工程并在模拟器上运行、将 AWTK 应用程序移植到其它平台。

在 AWStudio 上编写好 AWTK 应用程序后，部署到 RTOS 平台（如 STM32）是很方便的，下面就以 STM32F429 型号为例子来介绍如何编译 AWTK 应用到 RTOS 平台运行。

## 搭建RTOS平台编译环境

### 1. 下载Keil5与配置芯片包

由于将 AWTK 应用移植到 STM32 平台需要用到 Keil5，所以需要做一些准备工作，包括下载 Keil5 以及对应板子型号的芯片包，Keil5 以及芯片包的下载地址分别可以从下面两个网址下载。

Keil5: [www.keil.com/download/product](http://www.keil.com/download/product)

芯片包: [www.keil.arm.com/devices](http://www.keil.arm.com/devices)

### 2. 下载移植工程

在官方 GitHub ([github.com/zlgopen](https://github.com/zlgopen)) 上有实现 STM32 几个型号的移植工程，其中包括 STM32F429 的工程 (awtk-stm32f429igtx-raw)。接下来的移植工作需要在这些工程上操作，因此这里下载并解压。其它型号如 STM32F103、STM32F767 的移植工程均可从官方 GitHub 下载。

| 名称            | 修改日期            | 类型              | 大小   |
|---------------|-----------------|-----------------|------|
| .git          | 2024/2/29 14:14 | 文件夹             |      |
| awtk          | 2024/2/29 14:16 | 文件夹             |      |
| awtk-port     | 2024/2/29 14:14 | 文件夹             |      |
| CORE          | 2024/2/29 14:14 | 文件夹             |      |
| docs          | 2024/2/29 14:14 | 文件夹             |      |
| FATFS         | 2024/2/29 14:14 | 文件夹             |      |
| HALLIB        | 2024/2/29 14:14 | 文件夹             |      |
| HARDWARE      | 2024/2/29 14:14 | 文件夹             |      |
| MALLOC        | 2024/2/29 14:14 | 文件夹             |      |
| SYSTEM        | 2024/2/29 14:14 | 文件夹             |      |
| TEXT          | 2024/2/29 14:14 | 文件夹             |      |
| USER          | 2024/2/29 14:14 | 文件夹             |      |
| .clang-format | 2024/2/29 14:14 | CLANG-FORMAT... | 4 KB |
| .gitignore    | 2024/2/29 14:14 | 文本文档            | 1 KB |
| format.sh     | 2024/2/29 14:14 | Shell Script    | 1 KB |
| gen_files.sh  | 2024/2/29 14:14 | Shell Script    | 1 KB |
| keilkill.bat  | 2024/2/29 14:14 | Windows 批处理...  | 1 KB |
| README.md     | 2024/2/29 14:14 | MD 文件           | 1 KB |

图 1 awtk-stm32f429igtx-raw移植工程目录结构

这些工程已经对接好了硬件设备，用户只需要添加应用到工程中编译即可。如果想要从空白工程从零开始移植，可以参考官方的《AWTK 移植及移植裁剪指南》文档：

[https://awstudio.zlg.cn/docs/awtk\\_docs/AWTK\\_Porting\\_Tailoring/](https://awstudio.zlg.cn/docs/awtk_docs/AWTK_Porting_Tailoring/)

另外，还需要将 AWTK SDK (上图中 awtk 文件夹) 复制到移植工程目录下，最终移植工程目录结构如上图所示：

## 编译移植工程并下载到嵌入式设备

### 1. 编译AWTK-DEMO

使用 Keil 打开 awtk-stm32f429igtx-raw/USER/awtk.uvprojx 文件即可打开移植工程，工程中默认带有一个 AWTK-DEMO 应用，可以直接点击 Keil 菜单栏的“Build”按钮编译工程。

### 2. 编译自己的应用

如果想在移植工程中编译在 PC 上调试好的 AWTK 应用，需要先屏蔽 AWTK-DEMO：

1. 在 Keil 的“Project”中右键 awtk-demo 文件夹 - “Options for Group ‘AWTK-DEMO’”；
2. 取消勾选“Include in Target Build”；
3. 点击“OK”保存。

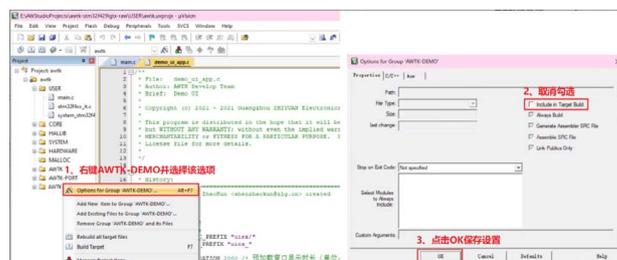


图 2 在Keil中屏蔽AWTK-DEMO

接着再往移植工程中加入自己应用的代码文件，这里拿我们先前编写的 MyHelloWorld-Demo 举例：

1. 点击 Keil 菜单栏的“Manage Project Items”按钮，弹出设置窗口；
2. 在“Groups”点击添加组按钮，并输入“MyHelloWorld-Demo”；
3. 选中新建的组并点击“Add Files”按钮，在文件对话框中选择并添加 MyHelloWorld-Demo 的代码文件；
4. 保存设置后再点击 Keil 菜单栏的“Rebuild”按钮重新编译工程即可。

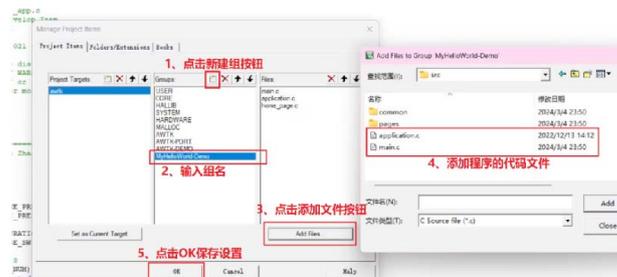
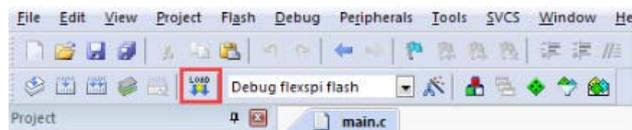


图 3 在Keil中添加MyHelloWorld-Demo代码文件

### 3. 下载应用到嵌入式设备

编译好移植工程后，使用仿真器（如：ST-Link）连接嵌入式板子和电脑，连接成功后在 Keil 上点击菜单栏的“Download”按钮即可将应用下载到板上。



其它型号的 RTOS 平台移植方法也是大同小异，可以从官方的 GitHub 下载对应的移植工程，先屏蔽工程中自带的示例代码文件（一般组名是 AWTK-DEMO），然后往工程中添加用户自己的应用代码及资源文件，最后编译应用并下载到设备即可。

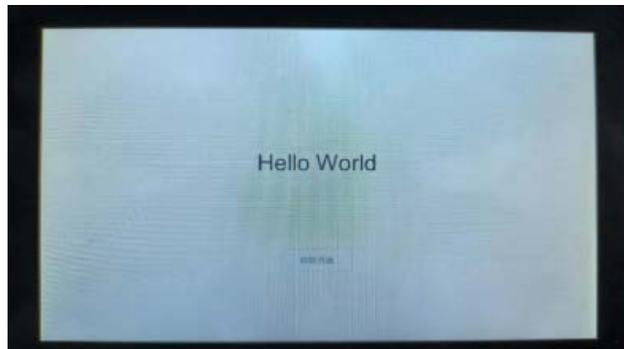


图5 实际板子运行效果

### 移植注意事项

在官方 GitHub 下载的移植工程默认配置好了宏，如果想要自定义开关一些宏可以修改 awtk\_config.h 文件里面的代码，注意使用这个头文件前提是需要定义 HAS\_AWTK\_CONFIG 宏。

还有一些需要注意的事项如下：

- 若要加载 PNG/JPG 格式的图片，StackSize 不可小于 32K，可以在 platform.c 中调整 Heap 大小；
- 在使用 Keil 编译时，要在 C/C++ -> Misc Controls 中加上 --gnu；
- 如果系统有 FPU，初始化时一定要开启 FPU；
- 如果想自定义添加 AWTK 代码文件，可以参考下面文档的第二、三章节内容 [awstudio.zlg.cn/docs/awtk\\_docs/HowTo/porting\\_common.html](http://awstudio.zlg.cn/docs/awtk_docs/HowTo/porting_common.html)。



**M1106/M1107核心板**

[点击图片](#)

# 【技术分享】 AWTK 开源串口屏开发(14) - 界面重用

ZLG 致远电子 2024-03-22 11:33:33

AWTK串口屏支持同一个界面绑定不同的数据，本文以家居控制为例，介绍一下界面重用的方法。

在家居控制应用中，我们演示了空调和咖啡机界面。现在我们来想一想，如果有多个空调怎么办呢？当然最直观的方法，就是把界面再拷贝一份。但是如果界面有多份，维护是一件麻烦的事情。今天我来介绍一下界面重用的问题。

## 1. 功能

用同一个空调界面实现两台空调的设置。

- 有两台空调



- 客厅空调



- 卧室空调



## 2. 创建项目

从模板创建项目，将 hmi/template\_app 拷贝 hmi/ui\_reuse 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

界面重用 demo\_home2 的界面，这里不再细说。

## 4. 添加绑定规则

### 4.1 空调界面

空调界面除了 空调\_位置 之外，其它的变量都是一样的，所以我们只需要添加 空调\_位置 的绑定规则即可。

| 绑定属性         | 绑定规则      | 说明 |
|--------------|-----------|----|
| v-data:value | { 空调_位置 } | 无  |

### 4.2 主窗口

#### 4.2.1 客厅空调

| 绑定属性         | 绑定规则  | 说明            |
|--------------|---|---------------|
| v-data:value | {navigate, Args=air_win?prefix=living_room} | 变量要用英文大括号括起来。 |

和之前的不同的是，我们加了参数 prefix=living\_room，表示获取 living\_room 下面的数据。

#### 4.2.2 卧室空调

| 绑定属性         | 绑定规则                                     | 说明            |
|--------------|--|---------------|
| v-data:value | {navigate, Args=air_win?prefix=bed_room} | 变量要用英文大括号括起来。 |

和之前的不同的是，我们加了参数 prefix=bed\_room，表示获取 living\_room 下面的数据。

## 5. 初始化数据

修改资源文件 design/default/data/default\_model.json，将其内容改为：

```
{
  "coffee": {
    "咖啡_开关": false,
    "咖啡_类型": 1, // 类型 (0: 卡布奇诺; 1: 拿铁; 2: 美式; 3: 意式)
    "咖啡_温度": 60, // 温度 (0-100)
    "咖啡_口味": 1, // 口味 (0: 浓郁; 1: 丝滑; 2: 清淡; 3: 平衡; 4: 温和)
    "咖啡_热奶": 1, // 热奶 (0: 少量; 1: 较少; 2: 较多; 3: 大量)
  }
}
```

```
"咖啡_奶泡":1,//奶泡(0:少量;1:较少;2:较多;3:大量)
"咖啡_水量":150,//水量(50-350ml)
"咖啡_剩余时间":200,//单位秒,格式化为分钟:秒
"咖啡_开始制作":false
},
"bed_room":{
"空调_开关":false,
"空调_位置":"卧室",
"空调_模式":3,//空调模式(0:制冷;1:制热;2:送风;3:除湿;4:
自动
"空调_风速":3,//空调风速(0:自动;1:低速;2:中速;3:高速)
"空调_垂直风向":1,//垂直风向(0:自动;1:上;2:中;3:下)
"空调_水平风向":1,//水平风向(0:自动;1:左;2:中;3:右)
"空调_温度":25,//温度(0-40)
},
"living_room":{
"空调_开关":false,
"空调_位置":"客厅",
"空调_模式":3,//空调模式(0:制冷;1:制热;2:送风;3:除湿;4:
自动
"空调_风速":3,//空调风速(0:自动;1:低速;2:中速;3:高速)
"空调_垂直风向":1,//垂直风向(0:自动;1:上;2:中;3:下)
"空调_水平风向":1,//水平风向(0:自动;1:左;2:中;3:右)
"空调_温度":25,//温度(0-40)
}
}
```

之前我们介绍的数据只有一级，现在我们的数据有两级，所以我们需要加上 prefix 参数，来访问 prefix 参数下的数据。

注意：

- 如果文件内容有中文（非 ASCII 字符），一定要保存为 UTF-8 格式。
- 重新打包资源才能生效。

### 6. 数据持久化

无

### 7. 编译运行



### 8. 注意

- 本项目并没有编写界面相关的代码，AWSudio 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\_home2。



# 【技术分享】AWTK 开源串口屏开发(15) - 通过 MODBUS 访问远程数据

ZLG 致远电子 2024-03-29 11:32:51

AWTK 串口屏内置 MODBUS Client 模型，支持用 MODBUS 协议从远程设备获取数据。不用编写代码即可实现对远程设备数据的显示和修改。

在 AWTK 串口屏中，内置了 MODBUS Client 的模型，支持用 MODBUS 协议从远程设备获取数据。不用编写一行代码即可实现对远程设备数据的显示和修改。

## 1. 功能

不用编写代码，实现对远程 MODBUS 设备数据的访问。

## 2. 创建项目

从模板创建项目，将 hmi/template\_app 拷贝 hmi/modbus\_client\_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\_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

### 4.1 URL

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

### 4.2 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 |

### 4.3 湿度

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

### 4.4 温度

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

### 4.5 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。

## 5. 配置文件

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

```
{
  "url": "tcp://localhost:502",
  "channels": [
    {
      "update_interval": 3000,
      "name": "read_input_registers",
      "access_type": 4,
      "read": {
        "offset": 0,
        "length": 4
      }
    }
  ],
  "variables": {
    "温度": "read_input_registers.word[0]",
    "湿度": "read_input_registers.word[1]"
  }
}
```

- url: 远程设备的 URL
- update\_interval: 更新间隔，单位毫秒
- channels: 通道列表
- name: 通道名称
- access\_type: 访问类型
- read: 读取配置

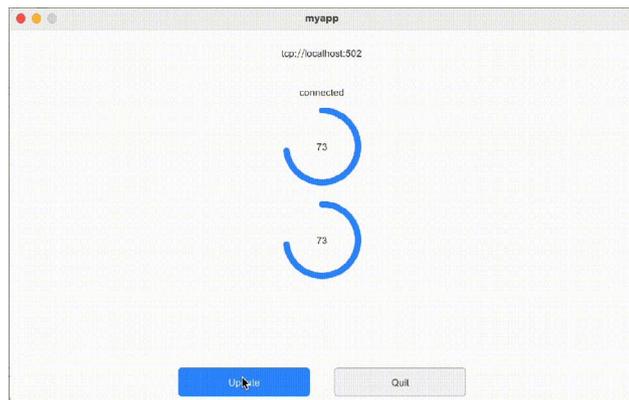
- offset: 偏移
- length: 长度
- variables: 变量列表
- 温度 : 映射到 modbus 通道的 word[0]
- 湿度 : 映射到 modbus 通道的 word[1]

## 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:



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, AWSStudio 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\_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。



# 【产品应用】EM储能网关& ZWS智慧储能云应用(1) 一使用介绍

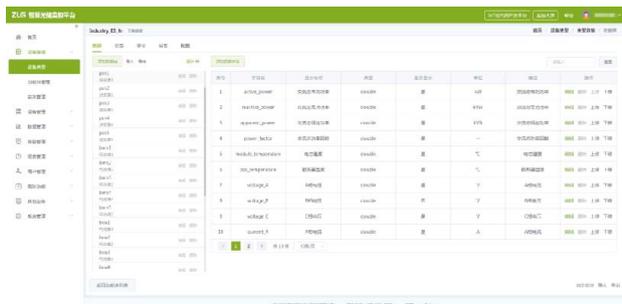
ZLG 致远电子 2024-03-07 11:38:43

在电池降本潮下，工商储行业迎来了发展新机遇，ZWS智慧储能云平台助力优化电力能源使用，降低电费成本。本文将介绍如何使用智慧储能云平台，实现智慧储能管理。

在双碳战略与能源转型的大背景下，工商储能行业迎来了发展新机遇，电池成本降低和储能技术进步，提升了工商储能的经济性。储能系统通过削峰填谷和需量控制，有效平衡供需，提高电力供应的灵活性和可靠性，帮助工商企业优化能源使用，增强电力供应的抗冲击能力，降低电费成本。ZWS智慧储能云平台是一个储能智慧管理平台，提供数字化运维、收益统计与分析、策略管理等功能，实现“开箱即用”的效果。

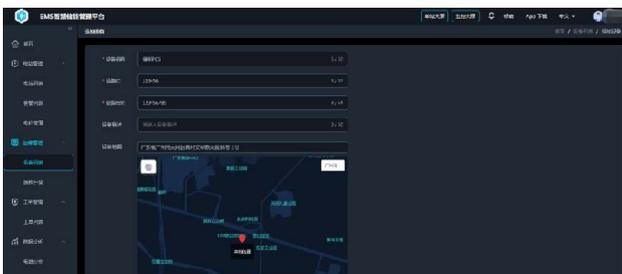
## 使用介绍

在智慧储能云平台中，平台已经建立了一套完整的物模型——Industry\_ES。这套模型抽象并归纳了工商储能设备共有的特性和功能，包括设备的数据点、状态、命令、日志等关键信息。物模型已经包括了工商储能设备标准数据点，客户还能基于自己的需求扩充数据点。



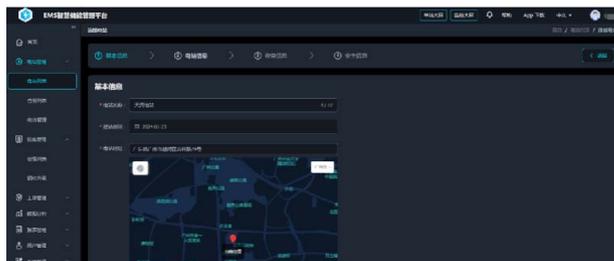
### 1. 创建设备

在云平台中对于储能系统设备进行添加。



### 2. 添加电站，绑定设备

进入智慧储能管理平台，进入“电站管理 > 电站列表”创建电站。电站创建后，将储能设备绑定电站。



### 3. 设备上传数据

设备端通过协议对接储能云平台物模型的数据点，将累计充放电电量等数据上报到云平台。

| 设备ID | 设备名称 | 设备类型 | 设备状态 | 设备容量    | 设备功率  | 设备效率 | 设备温度 | 设备电压 | 设备电流 | 设备电量   | 设备SOC | 设备DOH  |
|------|------|------|------|---------|-------|------|------|------|------|--------|-------|--------|
| 1    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 2    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 3    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 4    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 5    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 6    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 7    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 8    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 9    | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 10   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 11   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 12   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 13   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 14   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 15   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 16   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 17   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 18   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 19   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |
| 20   | 储能设备 | 储能设备 | 正常   | 1000kWh | 100kW | 95%  | 25°C | 500V | 100A | 500kWh | 50%   | 10000h |

### 4. 数据处理与展示

对于设备上传数据，储能云平台会对数据进行处理并展示。例如工商储能云平台能够根据设备上传的发电量数据，基于分时电价，计算出电站的收益，并在大屏上进行展示。



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。

点击申请

# 【产品应用】 ZWS云平台应用(3)-设备基于DTU接入

ZLG 致远电子 2024-03-28 11:34:43

传统设备要接入物联网，选择什么方式比较好？本文将介绍通过 DTU 智能网联终端 CATCOM-100 接入 ZWS 物联网云平台的功能。

对于一些工业制造、机械设备对温度有特殊的要求，我们就需要控制温度。这些设备，硬件电路简单，程序本身也不复杂。我们使用 DTU 智能网联终端 CATCOM-100 和 ZWS 物联网云，实现将一个传统的温控设备，低成本改造成可以远程监测控制的产品，实现硬件智能化。

## 功能目的

这次我们以传统温控器为例，介绍温控器通过 DTU 透传终端接入 ZWS 物联网云平台的过程。

### 1. 创建设备类型

云平台没有事先定义温控器的设备类型，需要参照《设备建模篇》，在自己的账号内创建一个温控器的设备类型。

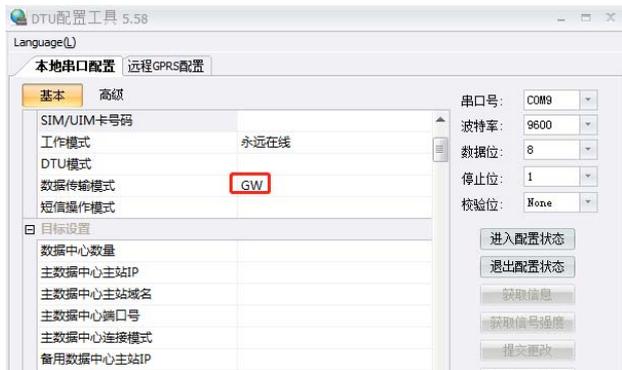
### 2. 创建设备

登录 ZWS 物联网云平台，添加 DTU 设备（CATCOM-100），添加温控器，定义设备类型、设备 ID、设备密钥。

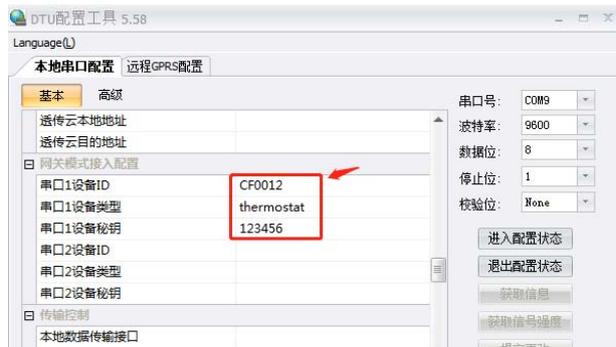


### 3. 配置DTU

打开 DTU 配置工具，将数据传输模式设置为“GW”网关模式。



将温控器的三元组（设备类型、设备 ID、设备密钥），填写到网关模式接入配置中。



### 4. 查看数据

下发获取温控器数据的命令，温控器上传原始数据，再经过云端脚本进行解析后，就能看到温度数据。



## 总结

通过 DTU 智能网联终端 CATCOM-100，用户可以将传统串口设备接入物联网云平台，运用 ZWS 物联网云平台，实现远程监测和分析。





## 边缘计算 ▾

### 多核异构，各有分工

ZMC900E 专为要求独特结合实时处理和通信与应用处理的工业应用而构建。采用 5 个 2.0GHz 高性能的 64 位 Arm® Cortex®-A55 CPU 内核，3 个高实时高可靠 Cortex®-R5F 双核锁步内核，Cortex®-A55 提供 RT-Linux 应用所必需的强大计算元件，Cortex®-R5F 低延迟路径提供出色实时性能。



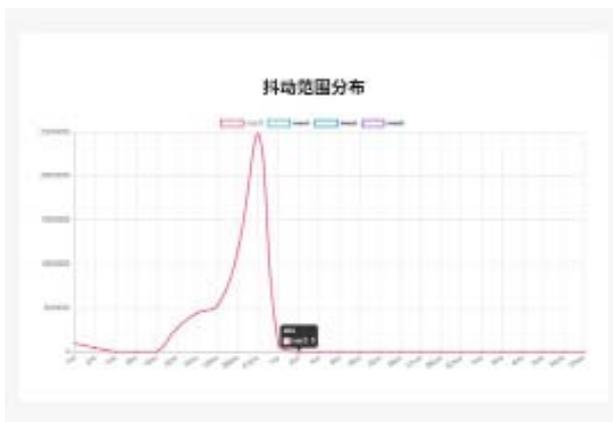
### 商业授权的EtherCAT协议栈稳定性更有保证

ZMC900E 采用商业授权 EtherCAT 协议栈，支持通过 Time-Triggered Send(TTS)TTS 硬件去抖动方式来消除周期报文的抖动，以解决应用层实时性不足的问题，通讯实时性更有保证。



### 高实时性操作系统搭配先进的网络技术，让系统拥有良好的扩展性

搭配的 RT-Linux 高实时性系统，有着实时性、跨平台性、便捷方案维护三个方面的优势。Ether-CAT 通讯可实现 ms 周期任务抖动  $\pm 1.5\mu s$  以内，满足运动控制的高实时性要求。



### 搭配示教器使用，机器人与设备操作更直观

ZMC900E 控制器可与 ZTP800 示教器 (移动 HMI) 搭配使用，ZTP800 优异的人体工学设计与友好的配套软件，可以显著提高使用者操控体验，共同为用户提供完整的机器人控制 / 示教解决方案。



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。

[点击申请](#)

# 【应用案例】ZMC900E 国产EtherCAT主站控制器 — 电源口传导骚扰整改

ZLG 致远电子 2024-03-06 11:40:05

为确保 EtherCAT 主站控制器在工业自动化中的可靠和稳定应用，EMC 测试至关重要。本文以 ZMC900E 电源传导骚扰整改为例，深入解析 EMC 原理，助您精准定位问题，避免无效整改。



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

为了能够在复杂的工业环境中稳定地使用，ZMC900E EtherCAT 主站控制器经过了严格 EMC 的性能测试。测试的 EMC 项目及等级如图 3 所示。控制器传导骚扰测试参考 GB/T 6113.201-2018 CLASS A 的标准进行试验。

| EMC 测试项目        | 接口                 | 限值  | 参考标准               | 等级          |
|-----------------|--------------------|---|--------------------|-------------|
| 谐波电压跌落试验        | 模拟                 | 50 Hz<br>30 A/m                                       | GB/T 17616.6-2006  | A           |
| 谐波电压(带负载)       | 模拟                 | <math>10\text{ kV}</math>                             | GB/T 17616.3-2018  | A           |
| 静电放电(接触放电)      | 模拟                 | $\pm 4\text{ kV}$                                     | GB/T 17626.2-2018  | A           |
| 静电放电(空气放电)接触放电  | 静电放电               | 0.15~80 MHz<br>10 V<br>80 %AM (30ns)                  | GB/T 17626.5-2017  | A           |
| 静电放电(空气放电)非接触放电 | 静电放电               | 0.15~80 MHz<br>10 V<br>80 %AM (30ns)                  | GB/T 17626.6-2017  | A           |
| 电压瞬态脉冲群电压试验     | 电压输入/输出            | $\pm 2\text{ kV}$<br>5, 25, 50, 100, 200, 1000<br>kHz | GB/T 17616.4-2018  | A           |
| 电压瞬态脉冲电压试验      | 电压输入               | $\pm 2\text{ kV}$<br>5, 25, 50, 100, 200, 1000<br>kHz | GB/T 17626.4-2018  | A           |
| 电压瞬态脉冲群电压试验     | CAN, RS485         | $\pm 2\text{ kV}$<br>5, 25, 50, 100, 200, 1000<br>kHz | GB/T 17626.5-2018  | A           |
| 浪涌(10/100)电压试验  | DC-GND             | $\pm 1.5\text{ kV}$ (峰值)<br>1.2/50ns(90% 10/100 us)   | GB/T 17616.5-2019  | A           |
| 浪涌(10/100)电压试验  | PLC-GND<br>CAN-GND | $\pm 1.5\text{ kV}$ (峰值)<br>1.2/50ns(90% 10/100 us)   | GB/T 17626.5-2019  | A           |
| 浪涌(10/100)电压试验  | 静电放电               | $\pm 1.5\text{ kV}$ (峰值)<br>1.2/50ns(90% 10/100 us)   | GB/T 17626.5-2019  | A           |
| 传导骚扰            | 模拟                 | 0.15~100 MHz  | GB/T 6113.201-2018 | 类 A 的限值(限值) |

图3 ZMC900E电磁兼容(EMC)特性参数

## 电源口传导骚扰试验原理

在传导骚扰测试中，LISN（线路阻抗稳定网络）是电源端口测试的关键设备，在接入待测设备后，其等效阻抗接近于 50Ω，由此可以知道，电源端传导骚扰测试的就是测试 50Ω 阻抗两端的电压。当阻抗不变时，也可理解为流经 50Ω 阻抗两端的电流。只要有电流流过 50Ω 阻抗，接收机便可显示出测试值。传导骚扰整改主要的分析方向则是如何减少流过 LISN 和接收机并联的 50Ω 阻抗的电流。电源端口的传导骚扰的电流产生于 IDM 和 ICM。其传导骚扰电流环路示意图如图 4 所示。

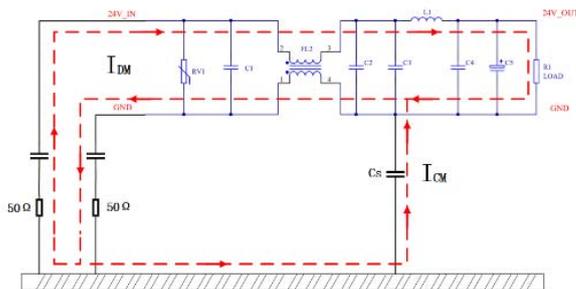


图4 电源端口传导骚扰电流环路

## 电源口传导骚扰试验

下面就和大家分享一下 ZMC900E EtherCAT 主站控制器传导骚扰的整改过程。图 5 为 24V 电源输入的保护电路，RV1 为压敏电阻，电容 C1、C2 容量选 1nF，电容 C3、C4 容量选 4.7uF，电感 L1 选用 3.3uH，共模电感 FT2 选用感值 470uH，电容 C5 容量选用 220uF，耐压值 50V。

## 边缘计算

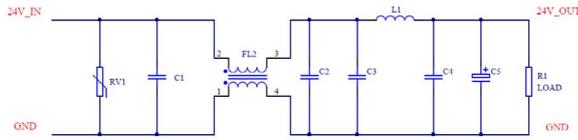


图5 电源输入电路

图6为未整改前的电源口传导骚扰测试结果，存在1个频率点(287.594KHz)超标，此频率点一般是由于开关电源的MOS管导通产生的谐波分量导致的传导性干扰，且还有众多的频率点裕量不足。

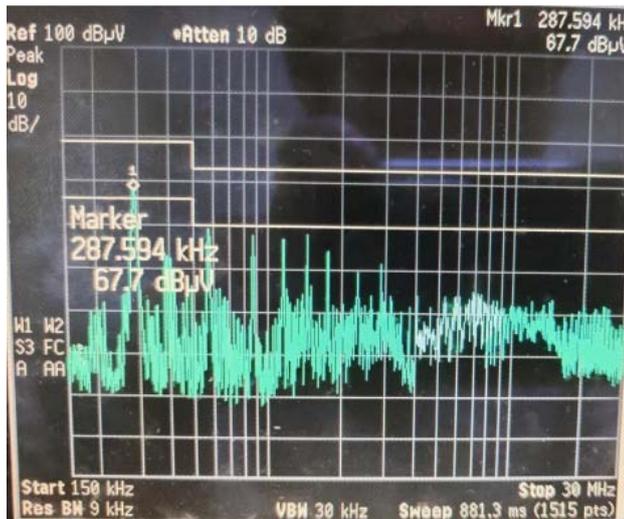


图6 传导骚扰测试结果

## 电源口传导骚扰试验整改措施

电源口传导骚扰试验整改可通过加大输入共模电感进行改善(图4中的FL2)，将共模电感FL2的感值由470UH调整为1mH后，重新进行电源口的传导骚扰试验，其中黄色曲线为峰值曲线，蓝色为实时扫描曲线，试验结果如图7所示。由试验结果可知，加大输入共模电感，可以有效地降低电源口的骚扰噪声，且裕量较充足。

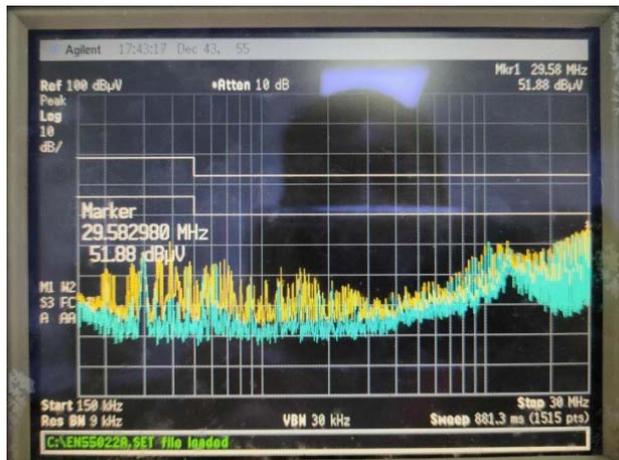


图7 1mH共模电感传导骚扰测试结果

再将共模电感FT2的感值由1mH调整为3.3mH后，重新进行电源口的传导骚扰试验，保留峰值曲线(蓝色)，试验结果如图8所示。

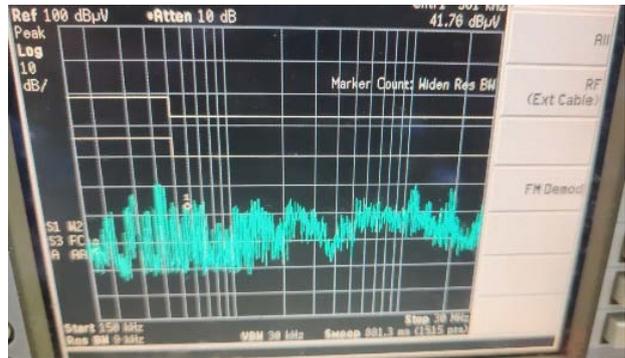


图8 3.3mH共模电感传导骚扰测试结果

## 电源口传导骚扰试验整改总结

1. 对比图7与图8可知，加大共模电感，可有效地降低电源口的骚扰噪声。所以在新方案设计时，在空间允许的情况下，尽量选择感值较大的共模电感。在确定输入电流时，感值越大时，共模电感的体积也会越大，成本也会增加，需综合考虑。
2. 由于不同厂家的电源模块，电源口的传导骚扰噪声也各不相同，所以电源口传导骚扰除了加大共模电感的感值外，也可通过更换电源方案来降低传导骚扰噪声。

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。

点击申请

# 【产品应用】 “脱胎换骨”--插针机运动控制

ZLG 致远电子 2024-03-14 11:37:17

随着 PCB 板及接插件装配行业的快速发展，插针机作为其核心设备的应用范围也越来越广泛。面对行业需求的迅速增长，致远电子推出了插针机运动控制方案。

插针机，也被称为插 PIN 机或入 PIN 机，是一种自动化设备，主要用于将插针插入到电路板上的预孔中，以实现电路板的连接。插针机主要由三大核心部分构成：机械部分、电气部分和控制系统。它们相互协作，共同完成插针的精准插入。机械部分是插针机的物理基础，电气部分则是插针机的“神经网络”，控制系统则是整个设备的核心部分。

## 插针机技术应用

在 PCB 板及接插件的装配过程中，插针机扮演着至关重要的角色。它通过精密的机械和电子控制系统，将金属针准确地插入到板上的孔内。全自动插针机作为当前主流设备，采用了在线式作业模式，实现了伺服运动控制。它具有自动进出板、自动定位、自动切针、自动插针等功能，几乎涵盖了生产的全过程，减轻了工人的劳动强度，提高了生产效率。



图1 插针机

## 面临的挑战

随着科技的进步和市场的不断扩大，插针机行业正面临着前所未有的机遇与挑战。为了满足市场对高精度、高稳定性、高效率和高速度的需求，插针机行业需要不断创新和升级。在竞争激烈的市场环境中，生产效率的提高意味着成本的降低和市场份额的增加。因此，插针机需要具备高效的运动控制能力，快速完成加工任务，提高整体生产效率。此外，插针机需要具备多种加工能力和灵活性，以满足不同产品、不同生产工艺的需求。为了应对这些挑战，插针机行业需要寻求一个稳定的、快速的运动控制解决方案。这个方案不仅需要具备高精度、高稳定性的特点，还要能够实现高效率、高速度、多功能和灵活性的要求。只有这样，插针机行业才能不断提升自身的竞争力，满足市场的不断发展和变化。

## 运动控制方案

运动控制是插针机的重要组成部分，它负责协调和控制机器的各种运动，包括主轴、工作台等各个部分的协同工作。一个成熟的、先进的运动控制解决方案对于插针机的性能和精度有着至关重要的影响，它不仅决定了机器的性能和精度，还影响着生产效率、故障预防和维护以及操作简便性等方面。因此，在插针机的设计和生产过程中，应重视运动控制系统的研发和优化，以提高整机的性能和竞争力。

## 传统的插针机运动控制方案

传统的插针机运动控制方案通常采用 PLC 通过脉冲方式控制脉冲型伺服驱动，伺服驱动在接受外部脉冲指令控制或模拟信号的同时，返回编码器信号。使用脉冲信号控制伺服电机时有如下几个优点：

1. 可靠性较高，可以避免因接线错误或其他问题导致的控制电压异常升高，从而降低伺服电机失控的风险。

2. 数字信号（如脉冲信号）相比模拟信号具有更好的抗干扰性。在复杂的工业环境中，数字信号不容易受到电磁噪声和其他类型的干扰的影响。

但总的来说，使用脉冲控制会导致控制的灵活性下降，其内部的位置环使得系统的 PID 参数调整变得不那么灵活。如果用户需要更高的控制性能，实现起来可能会比较困难，使用反馈控制会使系统存在两个位置环，这增加了控制器设计的复杂性。

其次，脉冲信号控制的驱动速率不高，快速性不足，脉冲控制方式本身可能不是为高速运动而设计的。当需要快速响应或高动态性能时，这种控制方式可能无法满足要求。此外，脉冲信号的传输和处理也可能受到限制，尤其是在高频率下，所以这进一步限制了控制的快速性。

为了提高伺服电机的控制性能，可以考虑使用更先进的控制策略和方法。同时，确保系统的硬件和通信链路具有足够的带宽和稳定性也是非常重要的。

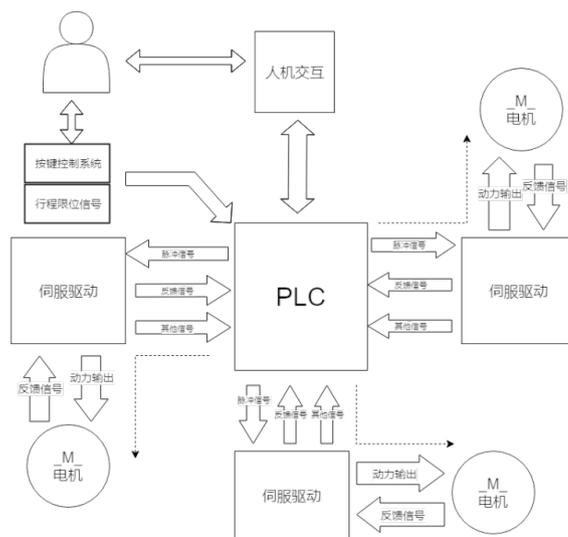


图2 插针机传统运动控制方案框图

## ZTP800/ZMC600E运动控制

ZTP800 / ZMC600E 运动控制 是基于 EtherCAT 总线的运动控制方案，相比于传统的脉冲控制方式，基于 EtherCAT 的总线技术的运动控制方案具有很强的灵活性和很高的性价比。在此基础上，EtherCAT 总线型运动控制器和示教器移动控制单元的搭配也显得尤为重要。这种搭配可以实现高效的人机交互，使得操作人员能够方便地进行设备的控制和监控。同时，这种搭配还可以提高设备的可维护性和可扩展性，使得设备能够更好地适应各种复杂的应用场景。



图3 ZTP800/ZMC600E 运动控制

广州致远电子股份有限公司开发的最新一代智能总线型控制器 ZMC600E 和示教器（移动 HMI）ZTP800 就是这样一套强有力的搭配。这套搭配具有高度的集成化和智能化水平，能够满足各种不同的应用需求，并且易于使用和维护。除此之外，还具有诸多优势，比如：

1. 节约布线成本，减少布线时间，减小出错机率：控制器的一个总线通讯口可以连接多台伺服，伺服之间用简单的 RJ45 口插接即可，缩短施工周期。
2. 信息量更大：全数字信息交互，可以双向传输很多参数、指令和状态等数据，脉冲方式只能单向传送位置或速度信息，无法获取伺服的更多状态或参数。
3. 精度高，数字式通讯方式：无信号漂移问题，指令和反馈数据精度可达 32bit。
4. 可靠性更高，抗干扰能力更强，不会出现丢脉冲现象：脉冲控制方式在高速脉冲时，会不可靠。
5. 降低系统总成本：当超过两台以上伺服时，不用调整控制器配置，而脉冲型伺服需要增加脉冲或轴控模块，伺服台数较多时甚至需要改用更高等级的控制器硬件才能满足要求。
6. 可开发软件功能更强大的设备，而无需额外修改硬件或接线：控制器能够实时通过总线监视伺服电机出现的故障，并在示教器上显示出来。同时控制器还可以监视伺服电机实际位置、实际速度等信息，也可以根据需要由程序自动调整伺服参数。可实现在示教器中设定伺服参数，而不用到伺服面板修改，简洁直观不易出错。
7. 采用标准的运动功能块库，提高编程调试效率：采用总线系统解决方案，避免了传统脉冲控制方式的编程量大、调试复杂等问题，提高了效率，节省了成本和时间。

通过使用 ZMC600E 和示教器（移动 HMI）的搭配，可以实现机器设备的快速配置和调试，提高生产效率和质量。同时，这套搭配还具有高度的可靠性和稳定性，能够保证机器设备的长期稳定运行。

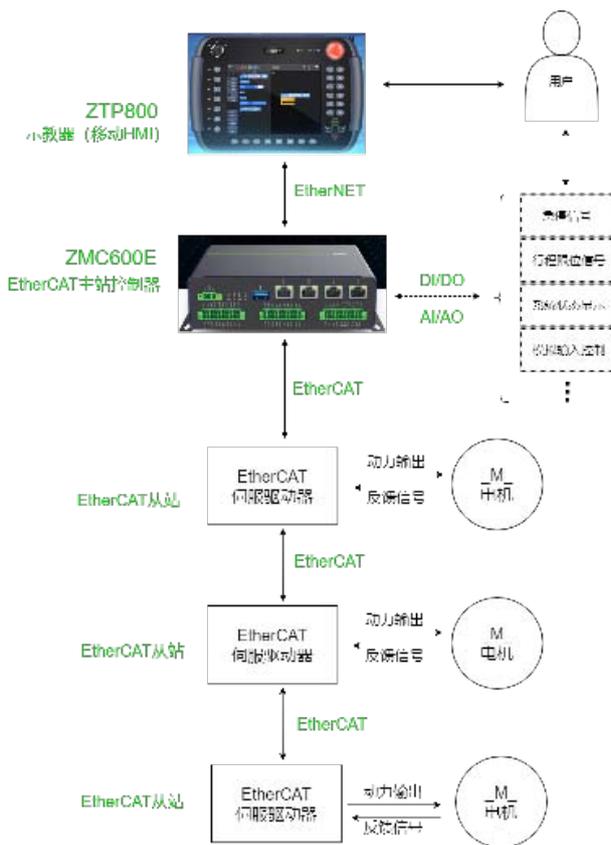


图4 ZTP800/ZMC600E 插针机运动控制框图

## 总结

随着科技的进步和制造业的发展，EtherCAT 技术在 PCB 板及接插件装配领域的应用越来越广泛。这种技术为行业带来了更高的生产效率和更精细的加工质量，满足了市场对高精度、高效率、高品质产品的需求。为 PCB 板及接插件装配行业带来了革命性的变革。随着技术的不断进步和应用领域的拓展，我们有理由相信，基于 EtherCAT 技术的 ZTP800 搭配 ZMC600E EtherCAT 总线型运动控制器的运动控制解决方案将在未来发挥更加重要的作用，推动行业的持续发展和进步。

如需了解更多产品详情，请填写申请表单，  
我们会有专人与您联系。

点击申请

# 【产品应用】 EtherCAT主站控制器系统实时性测试

ZLG 致远电子 2024-03-25 11:32:34

EtherCAT 作为高效的现场总线网络技术，其实时性能需依赖高实时操作系统才能充分展现。本文以 ZMC600E 为例，测试系统实时性并分析操作系统抖动数据。



ZMC600E EtherCAT 主站控制器是广州致远电子股份有限公司开发的一代智能总线型控制器，其采用多核异构应用处理器为核心，其内核包括 2 个 64 位的 Arm®Cortex®-A53 核，主频 1.0GHz；4 个 Cortex®-R5F 内核，主频 800MHz。同时板载 1GB DDR4、4GB eMMC 以及 32KB FRAM。

ZMC600E EtherCAT 主站控制器应用于机器运动时，为了确保精确的控制和协调动作，保证机器的运行稳定性和精度，ZLG 致远电子在开发过程中，根据多年深耕嵌入式领域的积累，对操作系统做了深度优化，提供了稳定可靠的 RT 补丁进行 Linux 系统的适配，并反复测试优化，得到了较好的系统抖动参数。下面使用一个用于测试评估操作系统实时性能的工具“cyclictst”来实测。

## 测试工具介绍

“cyclictst”的测试原理，是基于周期性任务的调度和执行，去创建一个或多个周期性任务，并通过设置任务的优先级和调度策略来模拟实时任务的运行。这些任务会在固定的时间间隔内不断运行，并记录每次运行的开始和结束时间。通过这些时间数据进行分析，便可得到任务的响应时间和延迟情况。

## 测试步骤

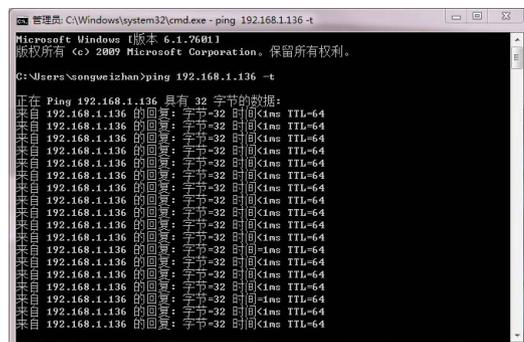
1. 准备好 1 台测试电脑、1 台 ZMC600E 控制器、1 台可输出 24V 的稳压电源、1 条网线。



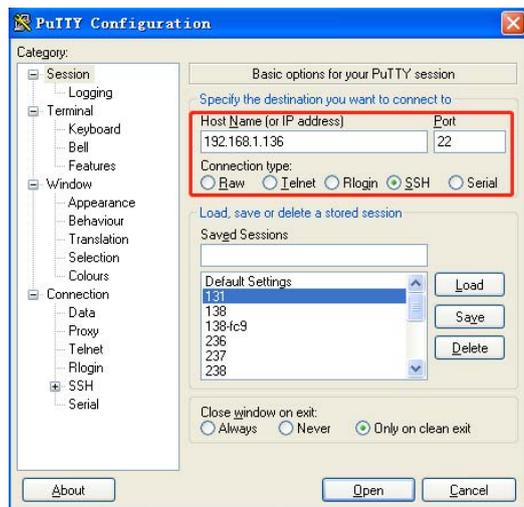
2. 稳压电源设置为 24V/1A，网线连接 ZMC600E 控制器的 NET1 和测试电脑的网口。打开电源，给 ZMC600E 控制器供电。



3. 测试电脑 IP 设置为 192.168.1.168，并通过 ping 包工具确认可以 ping 通 ZMC600E 的 NET1 的 IP 地址：192.168.1.136。



4. 测试电脑打开串口调试工具“Putty”，在“Host Name”栏填入工控板的 IP 地址“192.168.1.136”，Port 栏输入“22”，选中“SSH”协议，如图 4 所示。并点击“Open”连接到 ZMC600E 控制器。



## 边缘计算 ▼

5. 输入用户名“root”和密码“root”，登录后，输入“cd /usr/bin”进入 bin 目录，再输入“cyclictest -p 98 -t 1”命令进行测试。

```
root@ZMC600:~# cd /usr/bin/
root@ZMC600:/usr/bin# cyclictest -p 98 -t 1
WARN: stat /dev/cpu_dma_latency failed: No such file or directory
policy: fifo: loadavg: 0.08 0.07 0.02 1/129 373
T: 0 ( 373) P:98 I:1000 C: 1402 Min: 9 Act: 13 Avg: 12 Max: 23
```

6. 保持测试环境在连续大概 15 个小时的系统抖动性测试中，得到输出结果：最大抖动为 98us，平均抖动为 15us。

```
root@ZMC600:~# cyclictest -p 98 -t 1
WARN: stat /dev/cpu_dma_latency failed: No such file or directory
policy: fifo: loadavg: 0.64 0.66 0.70 1/128 473
T: 0 ( 380) P:98 I:1000 C:60745850 Min: 8 Act: 13 Avg: 15 Max: 98
```

## 测试结果分析

按照同样的方法，分别测试 ZMC300E 和 ZMC900E EtherCAT 主站控制器的系统抖动数据，得到以下表格。

| 型号   | ZMC300E | ZMC600E | ZMC900E |
|------|---------|---------|---------|
| 平均抖动 | 18us    | 15us    | 3us     |
| 最大抖动 | 67us    | 98us    | 11us    |

通过测试步骤及测试结果可知，不同的产品平台，系统的实时性能不一样。其中 ZMC300E 是通过 Xenomai 补丁实现系统的实时性，ZMC600E 和 ZMC900E 是通过 RT 补丁实现系统的实时性。为了确保精确的机器控制和协调动作，保证机器的运行稳定性和精度，需要尽可能减小 EtherCAT 主站操作系统的抖动，才可以充分发挥 EtherCAT 的先进网络性能。

ZMC300E/ZMC600E/ZMC900E 选型表如下。

| 型号      | 内核                  | 主频     | 内存    | 存储    | 以太网              | 无线       | CAN | RS485 | M2 | DI/DO | AI/AO | HDMI |
|---------|---------------------|--------|-------|-------|------------------|----------|-----|-------|----|-------|-------|------|
| ZMC300E | 康威 CortexR-A8       | 800MHz | 256MB | 256MB | 1*EtherCAT+2*NET | --       | 1路  | 1路    | -- | 16/16 | --    | --   |
| ZMC600E | Cortex8-A53*2+R5P*4 | 1GHz   | 1GB   | 4GB   | 1*EtherCAT+3*NET | --       | 2路  | 2路    | -- | 16/16 | --    | --   |
| ZMC900E | Cortex8-A55*4+R5P*3 | 20Hz   | 4GB   | 8GB   | 2*EtherCAT+2*NET | Wi-Fi/BT | 1路  | 1路    | 支持 | 16/16 | --    | 支持   |

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。

点击申请

# 【技术分享】 CAN总线波形中为什么ACK电平偏高?

ZLG 致远电子 2024-03-27 11:33:35

在观察 CAN 通信波形时，我们会发现差分电平在 ACK 段突然增高，这是什么原因导致的呢？本文结合测试实例对 ACK 电平偏高的原因做简单分析。

## ACK简介

**ACK的作用：确认一帧报文是否正常接收。**

以标准数据帧为例，从结构上看分成7段，分别为起始段、仲裁段、控制段、数据段、CRC 校验段、ACK 应答段、帧结束段，如图 1 所示：



图1 标准数据帧结构

ACK 段长度为 2 个位，包含应答间隙 (ACK SLOT) 和应答界定符 (ACK DELIMITER)。在应答场里，发送站发送两个“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。如图 2 (CANScope 测试的 ACK 段波形图) 所示：

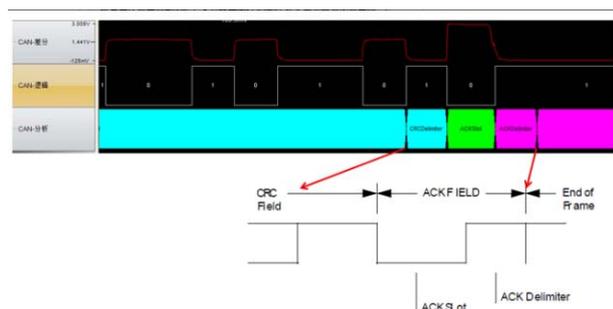


图2 ACK应答位结构

**应答间隙：**所有接收到匹配CRC序列 (CRC SEQUENCE) 的节点会在应答间隙 (ACK SLOT) 期间用一“显性”的位写入发送器的“隐性”位来作出回答，由于CAN总线线的原理，只要总线上有一个节点正确接收到数据，则ACK SLOT 就会被填入显性电平。

**ACK 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 DELIMITER) 和ACK界定符 (ACK DELIMITER)。

当一个接收节点接收的帧起始到CRC段之间的内容没发生错误时，它将在ACK段发送一个显性电平。如图3所示：



图3 ACK应答过程分析

## CAN总线ACK电平偏高原因分析

以使用ZLG致远电子ZPS-CANFD测试某电车CAN网络为例，20多个CAN节点采用手牵手方式组网，线两端各接120欧姆电阻，通讯能够正常，但差分信号波形应答应电平偏高较多，如图4所示，从CANScope的波形图中可以清楚的看到ACK电平突增。

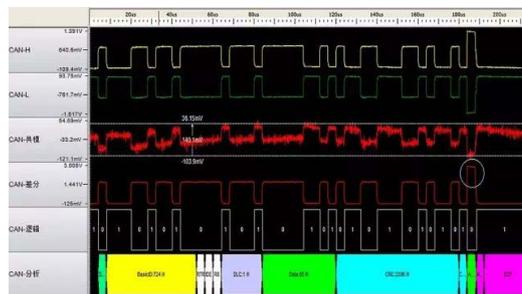


图4 CAN数据帧波形分析

## CAN总线通信存在异常？ACK应答有问题？——NO。

分析如下：

首先，我们看一款CAN收发器的典型内部结构图，CANH、CANL连接总线。

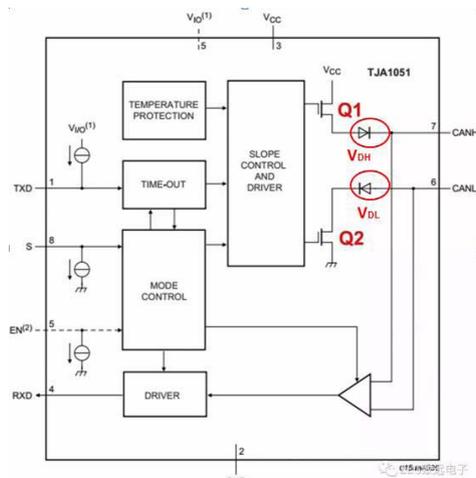


图5 CAN收发器的典型内部结构图

### 互联互通

总线显性时（逻辑为 0），收发器内部 Q1、Q2 导通，CANH、CANL 之间产生压差；隐性时（逻辑为 1），Q1、Q2 截止，CANH、CANL 处于无源状态，压差为 0。

按照标准，在总线输出为显性时，CANH=3.5V，CANL=1.5V，差分电平：CANH-CANL=2V，总线网络电阻为 60 欧，流经终端电阻的电流约为 33mA(2V/60 欧)；

然而，CAN 收发器在输出显性时，CANH 电平并不是标准的 3.5V，而是 5V(VCC) 通过一个二极管降压得到的，二极管的压降由负载电流决定。图 6 所示的 VDH 为 CANH 对应的二极管压降，VDL 为 CANL 对应的二极管压降，流过的电流越大，二极管的压降就越大。可知：

CANH 的电压=5V-VDH

CANL 的电压=0V+VDL

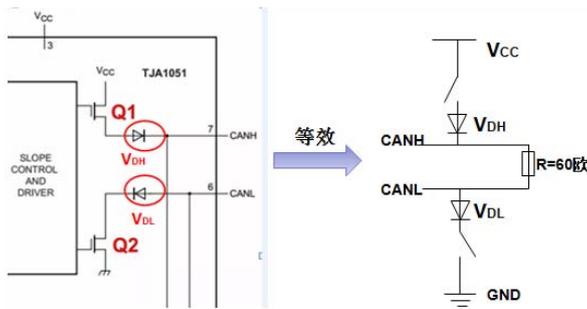


图6 ANI收发器等效结构

在 CAN 总线网络中，当一帧报文被各个节点接收时，在 ACK 阶段，多个 CAN 节点同时响应（都发显性），流过终端电阻的电流被各个 CAN 节点均分，那么平均到每个节点的电流就减小了，如下图所示，VDH、VDL 的压降也减小，相应 CANH-CANL 的差分电压就增大了，即 ACK 应答电平偏高。

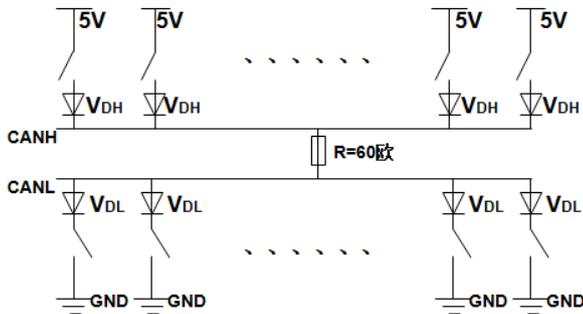


图7 CAN网络等效结构

### CK的意义

当 CAN 总线上只有一个节点可收发数据时，总线上因无接收节点在 ACK SLOT 时间内发送“显性”位填充，而始终保持隐性，发送者会检测到这个隐性位而知道发送失败，此条报文需要重发。因此这个节点会一直重发数据直到发送成功或发送被取消。

当总线上有多个 CAN 节点组网通信时，由于总线电平线性的原理，只要总线上有一个节点正确接收到数据，则 ACK SLOT 就会被填入显性电平（此时认为数据帧发送成功），那么，接收错误的节点如何来告知发送者此次发送不成功呢？这就要用到 CAN 的错误帧，当一个接收节点收到错误数据时，它立即广播发送一个错误帧，其它的节点和发送者也都会收到这个错误帧而丢掉此次报文，发送节点重新发送，这才是 ACK 的意义。

### 总结

作为国内 CAN 总线系统解决方案供应商，同时也是 CIA 协会在中国最主要的 CAN 总线技术传播的窗口，目前广州致远电子的产品覆盖了从分析诊断设备、接口转换设备、到协议组网设备等全面系统的 CAN 总线产品家族，可为用户提供完整的 CAN 总线解决方案。





**车载多通道 CANFD- bus  
数据记录终端**

👉 点击购买

# 【新品发布】 新一代工业数采边缘网关GXF222-4G全新上市

ZLG 致远电子 2024-03-18 11:39:22

## 新一代工业数采边缘网关—GXF222-4G



## 典型应用场景



## 选型表

| 型号          | GXF222-4G               |
|-------------|-------------------------|
| RS-485 数量   | 2 路                     |
| RS-485 波特率  | 2400~230400bps          |
| RS-485 接口形式 | 端子                      |
| CAN 数量      | 2 路 (支持 CAN FD)         |
| CAN 最大速率    | 2500 帧 / 秒 / 每路         |
| CAN 波特率     | 40k~5Mbps               |
| CAN 接口形式    | 端子                      |
| 调试接口        | USB (TypeC)             |
| 4G 数据传输     | Cat.4                   |
| 4G 最大下行速率   | 150Mbps                 |
| 4G 最大上行速率   | 50Mbps                  |
| 数据上云        | 支持                      |
| 定位功能        | GPS/北斗                  |
| 本地数据存储      | 支持                      |
| SD 卡类型      | Class10                 |
| DI          | 4 路                     |
| DO          | 2 路                     |
| ADC         | 4 路                     |
| 120 欧姆终端电阻  | 支持 (CAN FD、RS-485)      |
| 供电电压        | DC 9~36V                |
| 掉电保存        | 支持                      |
| 工作温度        | -40°C ~ +80°C (不含 SD 卡) |
| 产品尺寸        | 158.00×102.40×30.80mm   |
| 安装方式        | 导轨安装 / 挂耳安装             |

## 高性能通讯接口



## 互联互通 ▼

### CAN FD 数据本地存储



### 数据压缩传输，节省客户流量



### 云平台私有化部署

提供平台私有化服务，客户可选用致远 ZWS 云平台部署到自有服务器中，完成自主独立管理与应用。



### 云平台远程维护



### 易用的配置工具



### 工业级品质，稳定可靠



## 行业应用



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。

[点击申请](#)

# 【技术分享】

## 深入了解LoRa技术① — LoRa产品的信道划分

ZLG 致远电子 2024-03-26 11:36:38

得益于 LoRa 技术的通信距离优势，如今世界各地已经在广泛地使用 LoRa 产品。那么 LoRa 产品被划分了哪些频段呢？本期文章将向大家介绍世界各地允许 LoRa 产品使用的信道频率和信道带宽。

### LoRa产品的信道频率

和使用 4G 蜂窝网的产品相比，LoRa 产品使用的是免费的 ISM 频段 (Industrial Scientific Medical Band)。而世界各地对 ISM 频段的划分大不相同，因此在使用 LoRa 产品时需要注意各地具体的法规。ISM 频段，包括 433MHz、470MHz (470-510)、868MHz (862-870)、915MHz (902-928)、2.4GHz 等。其中低于 1GHz 的频段称为 SubG-LoRa。中国大陆使用 470MHz 频段，欧洲各国、非洲各国和印度使用 868MHz 频段，美国和澳大利亚使用 915MHz 频段，亚洲的部分地区使用 433MHz 频段。日本和韩国则使用 920MHz (920-925) 频段。

相比于 2.4GHz，SubG-LoRa 更适合作为长距离和低功耗通信的选择。其优势是频率低，通信距离远，穿墙能力强，功耗低。而且在当前的生态环境中，Wifi、蓝牙和 Zigbee 等都使用 2.4GHz 作为工作频段，所以 SubG-LoRa 频段的频谱干扰相对更少，使用 SubG 频段可以提高网络的整体性能，减少重传次数。主要地区的频段划分如图 1 所示。

\* LoRa operates in the unlicensed ISM (Industrial, Scientific and Medical) radio band that are available worldwide.

| Region                                | Frequency (MHz) | Region    | Frequency (MHz)  |
|---------------------------------------|-----------------|-----------|------------------|
| Asia                                  | 433             | Australia | 915-928          |
| Europe, Russia, India, Africa (parts) | 863-870         | Canada    | 779-787          |
| US                                    | 902-928         | China     | 779-787, 470-510 |

图1 LoRa产品的频段划分

那 2.4GHz 的 LoRa 是否就无用武之地了呢？当然不是。因为 2.4GHz 可用的频率范围比 SubGHz 要大得多，所以使用 2.4GHz 作为工作频率时，可以使用更大的信道带宽，并且可以选择的频率更多。在传输距离不那么远，而且对传输速度有一定要求的场景（如实时物联网和边缘计算）使用 2.4GHz 作为 LoRa 的信道频率是一个比较好的选择。

广州致远电子股份有限公司推出的 ZM68S 系列 LoRa 无线模组是一款高性能低功耗的 LoRa 透传模组。现已在 470MHz 频段的 ZM68S 型号基础上，推出了 868MHz 频段的 ZM68S-B 型号和 915MHz 频段的 ZM68S-C 型号，能够满足各地的法律法规。

### LoRa产品的信道带宽

信道带宽，是限定允许通过该信道的信号下限频率和上限频率，可以理解为一个频率通带。增加带宽可以在同一时间内传输更多数据，即可以缩短传输时间，但是会牺牲部分接收灵敏度作为代价。LoRa 芯片和模组大部分包含 LoRa 和 FSK 两种调制方式，LoRa 带宽是双边带宽（全信道带宽），

而 FSK 带宽是单边带宽。在中国大陆使用的 470MHz 至 510MHz 频段里，LoRa 常用的信道带宽为 125KHz、250KHz 和 500KHz。更详细的带宽选项如表 1 所示。

表1 LoRa产品的带宽选项

| 带宽 (kHz) | 扩频因子 (SF) | 编码率 (CR) | 标称比特率 (bps) |
|----------|-----------|----------|-------------|
| 7.8      | 12        | 4/5      | 18          |
| 10.4     | 12        | 4/5      | 24          |
| 15.6     | 12        | 4/5      | 37          |
| 20.8     | 12        | 4/5      | 49          |
| 31.2     | 12        | 4/5      | 73          |
| 41.7     | 12        | 4/5      | 98          |
| 62.5     | 12        | 4/5      | 146         |
| 125      | 12        | 4/5      | 293         |
| 250      | 12        | 4/5      | 586         |
| 500      | 12        | 4/5      | 1172        |

依据表 1 可以看出，LoRa 带宽越宽，标称比特率就越大，也就是实际传输速率越大。需要注意的是，部分 LoRa 产品不支持太小的信道带宽和过低的标称比特率。



**低成本LoRa射频模组**

[点击购买](#)

# 【新品发布】 仅需一连一配， 轻松实现各类设备接入EtherCAT总线系统

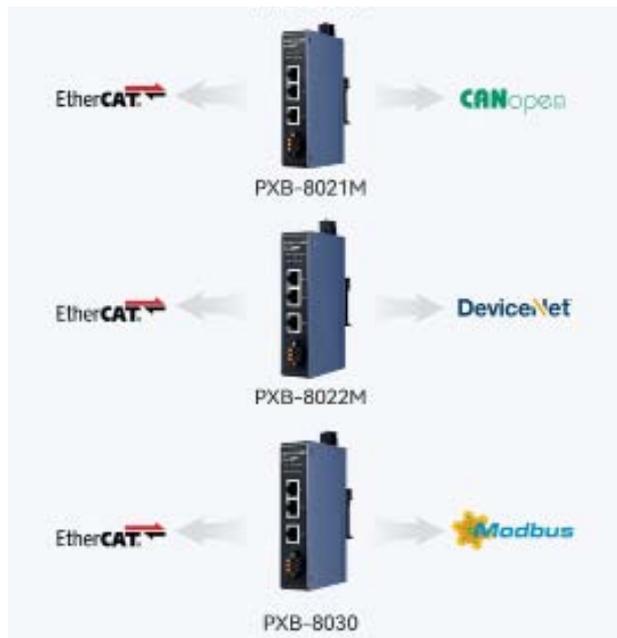
ZLG 致远电子 2024-03-11 11:51:31

## PXB-80系列EtherCAT协议转换

让各类工业总线设备接入 EtherCAT 网络



EtherCAT 是目前全球最快的工业以太网通讯技术之一，由于此特性，在高端制造设备、运动控制等高速需求领域应用广泛。然而，在庞大复杂的自动化系统中，有各类仪器仪表、执行器件、节点设备等需要接入系统，造成不同的总线协议产品需要与 EtherCAT 总线系统互联互通。若自己开发，需要熟悉各类协议代码并伴随大量软件编程工作，会造成大量的精力损失及人力资源浪费，项目周期也无形中被拖长。由此我们推出 PXB-80 系列产品，让您仅需一连一配，经过上位机软件简单配置，就能实现多数工业设备、组件接入 EtherCAT 总线系统。



## 选型表

| 产品型号          | PXB-8030               | PXB-8021M              | PXB-8022M                | PXB-8030                            |
|---------------|------------------------|------------------------|--------------------------|-------------------------------------|
| 协议转换          | EtherCAT to CAN/CAN FD | EtherCAT to CANopen 主站 | EtherCAT to DeviceNet 主站 | EtherCAT to Modbus                  |
| 处理器           | 国产 RISC-V              | 国产 RISC-V              | 国产 RISC-V                | 国产 RISC-V                           |
| 速率            | 10Mbit                 | 10Mbit                 | 10Mbit                   | 10Mbit                              |
| 隔离 CAN 接口     | 1 路                    | 1 路                    | 1 路                      | 1 路                                 |
| 隔离 RS485 接口   | /                      | /                      | /                        | 1 路                                 |
| 普通以太网         | 1 路                    | 1 路                    | 1 路                      | 1 路                                 |
| EtherCAT 从站接口 | RM5* 2 (IN+OUT)        | RM5* 2 (IN+OUT)        | RM5* 2 (IN+OUT)          | RM5* 2 (IN+OUT)                     |
| CAN FD 协议     | 支持 CAN/ CAN FD         | /                      | /                        | /                                   |
| CANopen 协议    | /                      | 支持 CANopen 主站          | /                        | /                                   |
| DeviceNet 协议  | /                      | /                      | 支持 DeviceNet 主站          | /                                   |
| Modbus 协议     | /                      | /                      | /                        | 支持 Modbus 主站/从站；支持 Modbus-TCP 及 RTU |
| 转换时间          | 1ms                    | 1ms                    | 1ms                      | 1ms                                 |

互联互通 ▾

| 产品型号 | PXB-8020   | PXB-8021H   | PXB-8022H  | PXB-8030   |
|------|--|---|--|--|
| 特性   | 标准 EtherCAT 从站；<br>标准 CAN/ CANFD/CANFD 加速；<br>支持用户自定义以及四种 CAN 帧发送模式；<br>支持最大同时处理 128 帧 CAN 发送模式；<br>支持 EtherCat 从站发送和接收过采样数据对象个数可寻，且最大分别为 254； | 标准 EtherCAT 从站；<br>I/O 数据触发方式最大支持 254 个字节的输入输出缓存；标准 CANOpen 主站<br>最大支持 126 个 CANOpen 从站 | 标准 EtherCAT 从站；<br>标准 DeviceNet 主站；<br>最大支持 63 个 DeviceNet 从站；<br>支持四种 I/O 数据触发方式；<br>支持 I/O 数据触发方式最大支持 64 字节的输入输出缓存 | 标准 EtherCAT 从站；<br>标准 Modbus 主/从站；<br>支持 Modbus TCP 及 RTU；<br>Modbus 主站模式最大支持 32 个 Modbus 从站；<br>支持抗刷、高数据量、空时缓存、输入寄存器操作功能等；<br>EtherCAT 转德顿缓冲区最大为 254 字节 |
| 供电电压 | 9-36V  | 9-36V   | 9-36V  | 9-36V  |
| 工作温度 | -40 ~ +85℃   | -40 ~ +85℃  | -40 ~ +85℃   | -40 ~ +85℃   |
| 存储温度 | -40 ~ +85℃   | -40 ~ +85℃  | -40 ~ +85℃   | -40 ~ +85℃   |
| 功耗   | 150mA@12VDC  | 150mA@12VDC   | 150mA@12VDC  | 150mA@12VDC  |
| 尺寸   | 125mm*76mm*28mm  |   |  |  |
| 安装方式 | 标准 35mm DIN 导轨   |   |  |  |



### PXB-80系列

👉 点击购买

# 【技术分享】CAN总线的传输距离与采样点、传输时延之间的关系，你弄明白了吗？

ZLG 致远电子 2024-03-05 11:37:56

CAN 总线的最大传输距离与波特率、采样点、传输时延之间存在什么样的关系呢？如何理论计算出最大传输距离呢？本文将详细解答这两个问题。

## 延时原理分析

影响 CAN 通信距离的主要因素是什么呢？经过我们实际测试长距离通信线缆带来的寄生参数（容抗、阻抗和感抗）影响总体较小，并且可以通过调节匹配电阻削减寄生参数的影响，因此影响通信距离的主因并不是寄生参数而是另有其它。

CAN 总线采用多主从通信模式，标准 CAN 帧报文如图 1 所示可分为 7 段，其中 ACK 域分为 ACK SLOTT（应答间隙）和 ACK DEL（应答界定符）。CAN 发送节点发出的 ACK 域为两个“隐性”位，当接收节点完成正确接收后，将会在 ACK SLOTT 向发送节点发送“显性”位以示应答。发送节点回读到 ACK SLOTT 呈显性状态时，就认为总线上进行了有效的应答。发送节点如果没有在应答间隙内检测到有效的显性位，则会判断总线应答错误，因此通信延时是影响通信距离的主因。整个线路传播延时时必须足够小，以确保 CAN 发送节点在 ACK SLOTT 内接收到有效的应答信号，以下讨论将围绕传输延时展开。



图1 CAN标准帧报文格式

CAN 信号的整个传播涉及到的延时有控制器延时、收发器和隔离延时、线缆延时，信号流向及延时框图如图 2 所示。理论上必须使整个传播延时和  $\Sigma t$  小于采样点位置，否则发送节点将接收不到正确的显性 ( $V_{diff} > 0.9V$ ) 应答信号，导致通信异常，CAN 总线显性应答延时说明如图 3 所示。

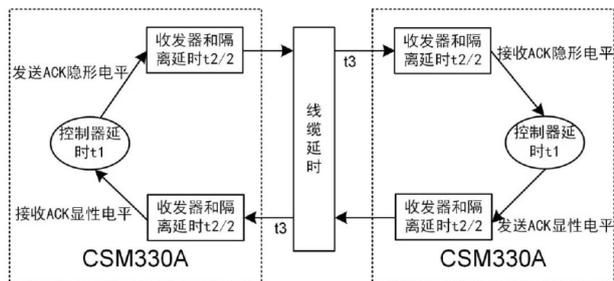


图2 CAN总线信号流向及延时框图

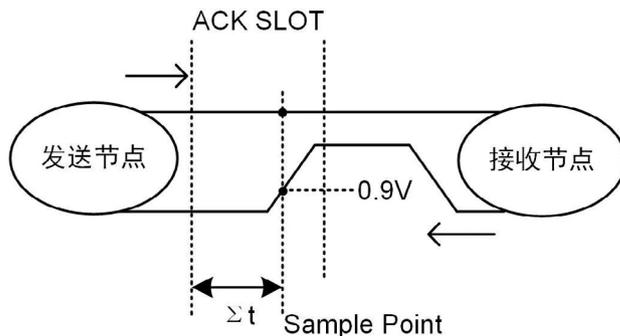


图3 CAN总线显性应答延时说明简易图

## 理论计算

通过上述原理分析，我们得出只有满足关系式  $T_s \geq 2 \cdot (t_1 + t_2 + t_3)$ ， $t_3 = L$  线长 \* (Cable delay)，通信才能成功，将关系式进行转换  $L$  线长  $\leq (T_s - 2 \cdot t_1 - 2 \cdot t_2) / 2 \cdot (\text{Cable delay})$ 。

经过我们实际测试我司的 CSM330A 的收发器和隔离延时  $t_2 = 140\text{ns}$ ，控制器延时  $t_1 = 50\text{ns}$ ，CSM330A 在 1M CAN 波特率采样位置为  $75\% \cdot 1\mu\text{s} = 750\text{ns}$ ，实测 RVSP 通信线缆每米延时 Cable delay = 5.5ns。将数值带入上述关系式中得出 CSM330A 在 1M 波特率下通信线长满足  $L$  线长  $\leq 33.6\text{m}$ 。

## 现场测试

### 如何延长通信距离

视频中有延长通信距离方案，本文做简要概述。如图 4 所示为 1M CAN 协议传输距离简图，接入 1 个用 CSM330A 方案搭建的 CAN\_Repeater，通信距离可延长至 66m 如图 5 所示，接入 n 个 CAN\_Repeater 后通信距离可延长至  $(n+1) \cdot 33\text{m}$ 。

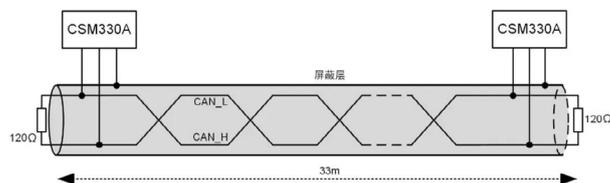


图4 1Mbps的CAN传输距离简图

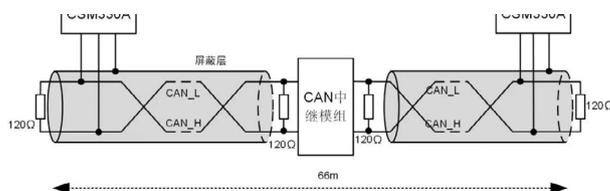


图5 接入1个CAN\_Repeater后传输距离简图

## 互联互通 ▼

### 总结

1. 得知各部分延时和采样位置后，我们可以通过关系式  $L_{\text{线长}} \leq (T_s - 2 \cdot t_1 - 2 \cdot t_2) / 2 \cdot (\text{Cable delay})$  预估出来不同 CAN 波特率下的最大传输距离；

例如：波特率为 10kbps，采样点在 80% 的 CSM330A 传输的理论距离为  $L_{\text{线长}} \leq (80000 - 280 - 100) / 2 \cdot 5.5 \approx 7.2\text{km}$ 。

2. 通过 CSM330A 搭建的 CAN 中继模组有效延长通信距离，该中继模组稳定可靠、成本低、开发周期短，CAN\_Repeater 配套工程文档可在 ZLG 致远电子公众号上回复“CSM330A”获取。



全隔离协议转换芯片

[点击图片](#)

# 【应用方案】 高端制造设备将DeviceNet信号转为EtherCAT

ZLG 致远电子 2024-03-12 11:39:43

EtherCAT 正快速应用于高端制造设备制造行业，然而，复杂的高端制造设备控制系统中存在多协议组件互联互通的需求，面对 EtherCAT 与 DeviceNet 的互通难题，我们该如何解决？

## 延时原理分析

影响 CAN 通信距离的主要因素是什么呢？经过我们实际测试长距离通信线缆带来的寄生参数（容抗、阻抗和感抗）影响总体较小，并且可以通过调节匹配电阻削减寄生参数的影响，因此影响通信距离的主因并不是寄生参数而是另有其它。

CAN 总线采用多主从通信模式，标准 CAN 帧报文如图 1 所示可分为 7 段，其中 ACK 域分为 ACK SLOT（应答间隙）和 ACK DEL（应答界定符）。CAN 发送节点发出的 ACK 域为两个“隐性”位，当接收节点完成正确接收后，将会在 ACK SLOT 向发送节点发送“显性”位以示应答。发送节点回读到 ACK SLOT 呈显性状态时，就认为总线上进行了有效的应答。发送节点如果没有在应答间隙内检测到有效的显性位，则会判断总线应答错误，因此通信延时是影响通信距离的主因。整个线路传播延时时必须足够小，以确保 CAN 发送节点在 ACK SLOT 内接收到有效的应答信号，以下讨论将围绕传播延时展开。



EtherCAT 是目前全球最快的工业以太网通讯技术，凭借该优势，EtherCAT 在高端制造设备制造领域不断被快速引入应用。然而高端制造设备制造是一个庞大复杂的控制系统，其中仍有各类支持不同通讯协议的组件需互联互通。

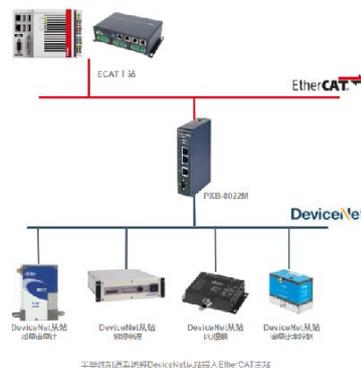
DeviceNet 作为一种低成本的通信连接，在高端制造设备行业应用广泛，高端制造设备中的 IO 模块、MFC、质量流量计、阀组、滑轨、温控器、射频、节流阀等组件，往往都采用 DeviceNet 进行通讯。

当需要将 DeviceNet 协议的从站组件接入 EtherCAT 主控系统时，我们面临一个关键问题：如何高效、便捷地实现两者的连接？

一种常见的解决方案是自主开发，但这需要对相关协议代码有深入的了解，并投入大量的时间和精力进行软件编程开发工作。这种方法不仅可能导致人力资源的浪费，还可能延长项目的整体周期。

然而，一个更为高效且便捷的选择是使用致远电子的 PXB-8022M 网关。该网关能够轻松实现 DeviceNet 与 EtherCAT 网络之间的无缝连接，使 DeviceNet 从站设备能够顺畅地接入 EtherCAT 主站网络。更为重要的是，使用 PXB-8022M 无需进行复杂的编程工作，大大节省了人力和精力的投入。

只需进行简单的配置，即可迅速实现 DeviceNet 设备与 EtherCAT 系统的互联互通。这样，您可以将更多的精力和资源投入到高端制造设备的核心功能开发上，提升整体项目的效率和质量。



PXB-8022M 产品是广州致远电子股份有限公司推出的 EtherCAT 转 DeviceNet 主的工业现场总线协议转换器。

该产品具有 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 DeviceNet 设备接入 EtherCAT 主控系统。

PXB-8022M 协议转换器提供简单好用的上位机配置工具，可以灵活配置相关功能，轻松实现各类工业总线协议数据到 EtherCAT 数据的无缝转换。

**PXB-80系列**

[点击购买](#)

互联互通 ▼

# 【应用方案】 高端制造设备将Modbus节点信号接入EtherCAT总线系统

ZLG 致远电子 2024-03-13 11:39:20

上一期我们讲在复杂的高端制造设备控制系统中，虽然 EtherCAT 得到了行业快速应用，但依然存在多协议组件互联互通的需求，今天来看看 Modbus 设备需要接入 EtherCAT 系统时，我们是如何解决的？



上一期我们讲在复杂的高端制造设备控制系统中，虽然 EtherCAT 得到了行业快速广泛的应用，但依然存在多协议组件互联互通的需求。

Modbus 作为一种低成本的通信连接，在自动化相关各个行业中应用广泛，常规通用级传感器、变送器、小单机设备、执行器等组件，往往都采用 Modbus 进行通讯。而当以上 Modbus 协议的从站组件，需高效快捷连入 EtherCAT 主控系统时，该如何解决呢？

我们知道若企业采用自主开发的方式，需要开发人员花费大量专门的时间和精力进行协议代码学习及协议代码编程工作，造成人力资源浪费的同时延长了项目整体周期。

更为高效而便捷的选择则是采用致远电子 PXB-8030 来实现 Modbus 与 EtherCAT 网络间的转换，PXB-8030 可以让 Modbus 从站设备（包括 RS485 串口设备及 Modbus TCP 工业以太网设备）无缝连入 EtherCAT 主站网络，仅需一前一配，无需编程，不需要额外花费过多人力及精力，就可快速实现 Modbus 设备和 EtherCAT 系统的互联互通，可以让您腾出更多精力专注于高端制造设备的核心功能上。

## PXB-8030产品介绍

PXB-8030 是广州致远电子股份有限公司推出的一款 EtherCAT 与 Modbus 的协议转换器。

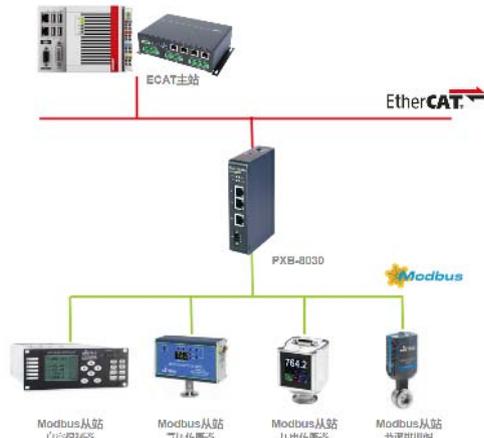
该产品具有 2 路 EtherCAT 从站网口接口、1 路 RS485、1 路标准 10/100M 以太网接口及 1 路 USB Type-C 接口，内置专用 ESC 芯片配合国产高性能 RISC-V 处理器，用以处理 EtherCAT 到各类工业现场总线数据报文的转换。

PXB-8030 协议转换器提供上位机配置工具，可以灵活配置相关功能，轻松实现各类工业总线协议数据到 EtherCAT 数据的无缝转换。



**PXB-80系列**

[点击购买](#)



# 【解决方案】汽车测试系统如何实现CANFD到EtherCAT的协议转换

ZLG 致远电子 2024-03-21 11:34:42

在汽车测试领域，EtherCAT 为可控的汽车底盘测试控制系统，需接入 CANFD 为传输信号的汽车底盘控制器进行采集分析，系统间数据无法直接读取，该如何解决？

我们知道汽车车机通讯系统中 CAN、CANFD 具有广泛的应用，在汽车动力域、底盘域、车身域等控制系统中应用非常普及。

在汽车底盘测试方面，为了尽量保证系统测试的高实时性及低延时性，MTS 等品牌测试系统主控采用基于 EtherCAT 的数字控制器，它需连接各类数据采集系统，完成材料、结构、车辆等力学性能测试过程中的数据采集工作。



当以上测试系统需连入汽车底盘数据时，往往测试系统无法直接通过 EtherCAT 网络读取，因为底盘控制器采用 CANFD 传输信号。

为了顺利的读取底盘控制器中 CANFD 数据，传统的方式需要开发人员熟悉协议代码，伴随繁杂的程序开发工作，经历大量的人力精力投入才能实现系统的互联。



而最快最便捷的方式是采用 PXB-8020 产品，仅需连接后通过 AWPX Tools 软件简单配置，避免过多的精力浪费及协议代码编程工作，就可轻松完成 CANFD 底盘数据和 EtherCAT 主控测试系统信号的互联互通，实现测试系统采集并分析汽车底盘数据，便于后期检测优化各项数据。



除了实现 CANFD 到 EtherCAT 的转换，PXB-8020 产品同时支持将 CAN 信号转换为 EtherCAT，车机 CAN 数据系统同样可以通过转换无缝接入 EtherCAT 主控系统。

## 产品介绍

PXB-8020 产品是广州致远电子股份有限公司推出的 EtherCAT 转 CAN/CAN FD 工业现场总线协议转换器。

该产品具有 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。

PXB-8020 协议转换器提供简单好用的上位机配置工具，仅需一前一配，轻松将 CAN 数据以容易处理的 I/O 数据展示到主站 PLC/ 控制器中，完成 CAN/CANFD 数据接入到 EtherCAT 控制系统。

**PXB-80系列**

[点击购买](#)

## 【产品应用】

# SM4500系列隔离收发芯片应用集锦

ZLG 致远电子 2024-03-19 11:36:13

SM4500 系列芯片是致远电子开发的“三合一”高集成度全隔离收发芯片，很多客户在使用时会遇到一些问题，本文以问答的形式解答 SM4500 系列芯片的疑惑，帮助客户更好的使用 SM4500 系列芯片。

致远电子基于近二十年的总线隔离技术及工艺经验积累，推出 SM 系列集成电源隔离、RS-485 收发电路和信号隔离电路“三合一”的高集成度全隔离 RS-485 收发芯片。

SM 系列全隔离 RS-485 收发芯片相较于传统模块方案，在超小、超薄 的 DFN 封装内部集成完善的 RS-485 总线隔离电路，产品最高支持 10Mbps 波特率，工作温度覆盖 -40°C ~125°C，满足各类复杂恶劣的工业现场 RS-485 总线隔离需求。



### 01

**问：SM4500系列芯片对比通用的RS-485芯片如MAX485有什么区别和优势？**

**答：**SM4500 系列芯片是 DC-DC 电路、信号隔离电路及 RS-485 总线收发电路三合一的芯片，具备高集成度和高可靠性。因其具有高隔离的特点，SM4500 系列芯片对比无隔离的 485 收发芯片更适合使用在工业通信、电力、煤矿等场景，可有效的解决工业现场因信号干扰导致的通信异常问题。

### 02

**问：SM4500系列有什么型号的芯片，怎么选型？**

**答：**具体区别及选型如表 1 所示：

表1 SM4500系列芯片选型表

| 产品型号    | 电源电压(范围)<br>(VDC)    | 传输波特率<br>(bps) | 节点数<br>(pcs) | 隔离耐压<br>(VDC) | 工作温度<br>(°C) | 封装    | 类型 |
|---------|----------------------|----------------|--------------|---------------|--------------|-------|----|
| SM4500  | 3.3/5<br>(3.15-5.25) | 10M            | 256          | 3500          | -40~125      | DFN16 | 高速 |
| SM4500H | 3.3/5<br>(3.15-5.25) | 10M            | 256          | 5000          | -40~125      | DFN16 | 高速 |
| SM4510  | 3.3/5<br>(3.15-5.25) | 10M            | 256          | 3500          | -40~125      | DFN20 | 高速 |
| SM4510H | 3.3/5<br>(3.15-5.25) | 10M            | 256          | 5000          | -40~125      | DFN20 | 高速 |

### 03

**问: SM4500系列芯片的接口电平如何匹配, VISOOOUT引脚是否可以用于其他电路供电?**

**答:** SM4500 系列芯片的 TXD、RXD、DE、RE 脚接口电平需与实际系统电平匹配, 一般若 MCU 为 3.3V 供电, VIO 连接 3.3V, 若 MCU 为 5V 供电, 则 VIO 连接 5V。VISOOOUT 为内部集成的高效隔离电源的输出, 在 5V 供电的条件下最大输出电流为 80mA, 为避免影响芯片正常工作, 建议只用于上拉电阻电路工作。

### 04

**问: SM4500系列芯片使用时需不需要外接保护电路?**

**答:** 建议使用 SM4500 系列芯片搭建电路时外接保护电路。RS-485 接收器差分输入端对“地”的共模电压允许 -7 ~ +12V, 超过此范围的过压瞬变就可能损坏器件, 特别是在工业和户外的使用场景, 大浪涌静电等瞬变干扰的存在会使芯片容易损坏。我们经常接到客户关于模块功能异常的反馈, 大部分是无保护电路造成的浪涌过压等损坏, 所以我们强烈建议客户在芯片的 A/B 端添加保护电路, 我们推荐以下保护电路, 供参考。

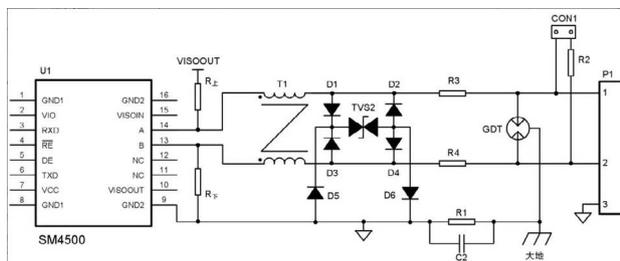


图1 推荐保护电路

表2 推荐参数表

| 标号                              | 型号            | 标号    | 型号             |
|---------------------------------|---------------|-------|----------------|
| C2                              | 102, 2kV      | TVS2  | SMBJ12CA       |
| R1                              | 1MΩ, 1206     | GDT   | 3RL090M-5-S    |
| R2                              | 120Ω, 1206    | T1    | HR600432, 51uH |
| R3, R4                          | SMD1812P014TF | D1~D6 | HFM107         |
| R <sub>上</sub> , R <sub>下</sub> | 选择合适阻值匹配网络    | U1    | SM4500         |
| CON1                            | 短路器           |       |                |

### 05

**问: SM4500系列芯片组网时在高速率下出现RS-485波形变形失真, 通信异常的情况?**

**答:** 在高速率的通信下出现波形变形失真的情况, 大概率是由总线寄生电容引起的, RS-485 总线的寄生电容过大, 信号的充放电时间越长, 波形越失真如出现锯齿波。有些客户在使用 GDT 和 TVS 做保护电路时, 选用的保护器件自身的寄生电容达几百甚至上千 pF, 当组网后容值叠加, 总线上的寄生电容就过大, 推荐使用图 1 的推荐保护电路, 接口电容仅为 20pF。

### 06

**问: SM4500系列芯片组网通信时收到异常电平?**

**答:** 收到异常电平可能的原因之一是接收信号时总线 AB 电平处于芯片的门槛电平, 即 -200mV ~ -10mV 内, 这将导致接收器接收信号不确定, 如图 1 所示, 给 AB 接口加上下拉电阻, 使接收器处于一个确定的电平状态, 如果 AB 端有终端电阻, 上下拉电阻还要调整到合适的阻值, 保证总线空闲时远离门槛电平范围。

### 07

**问:如何提高SM4500的通信距离?**

**答:** 理论上, 通信速率在 100kbps 及以下时, RS485 的最长传输距离可达 1200 米。但是有些客户搭建的总线网络却远远达不到 1200 米, 除了干扰恶劣的布线环境外, 最大的可能是少了终端电阻和选用了劣质的电缆! 在长距离通信系统中, 使用 120Ω 的终端电阻和特性阻抗 120Ω 的 RS-485 专用的双绞屏蔽电缆往往能最大程度的减少回波反射, 延长通信距离。我们的客户大部分是缺少终端电阻引起的回波反射导致通信距离不达标, 请一定要注意终端电阻。同时减小总线寄生电容、提升线缆的屏蔽效果, 都可以延长通信距离。

2024/3 第3期

**微文摘**  
ZLG MICRO DIGEST



ZLG致远电子官方微信