ZLG 致远电子

微文摘

ZLG MICRO DIGEST

2024/9 第9期

月刊

ZLG 致远电子

E系列 EtherCAT高速I/O模块

μs级响应　纤薄外观　极易拆装　清晰指示　通讯稳定　品质保障

选型表

模块功能	型号	规格
耦合器	ZPT-8080	高性能版，EtherCAT 总线耦合器套件，μs 级模块刷新周期（含电源，尾板）
数字量输入模块	ZDM-E1600P	高性能版，16 通道数字量输入，PNP 型，直插端子
	ZDM-E1600	高性能版，16 通道数字量输入，PNP/NPN 兼容，直插端子
	ZDM-E1600N	高性能版，16 通道数字量输入，NPN 型，直插端子
数字量输出模块	ZDM-E0016P	高性能版，16 通道数字量输出，PNP 型，0.5A，直插端子
	ZDM-E0016N	高性能版，16 通道数字量输出，NPN 型，0.5A，直插端子
	ZDM-E0008M	高性能版，8 通道机械继电器输出，2A，继电器隔离，直插端子
	ZDM-E0008S	高性能版，8 通道固态继电器输出，1A，继电器隔离，直插端子
模拟量输入模块	ZDM-E0800V	高性能版，8 通道模拟量电压输入，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZDM-E0800I	高性能版，8 通道模拟量电流输入，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
模拟量输出模块	ZMD-E0008V	高性能版，8 通道模拟量电压输出，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZMD-E0008I	高性能版，8 通道模拟量电流输出，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
温度采集	ZDM-E0400P3	高性能版，4 通道 RTD 温度信号采集，3 线制，PT100/PT1000，±0.5℃精度
	ZDM-E0800TC	高性能版，8 通道 16 位热电偶温度信号采集，K 型 /T 型，±0.5℃精度

致远电子官方网站　致远电子官方微信

CONTENTS
目录

【AWTK使用经验】如何在AWTK显示阿拉伯文本

ZLG 致远电子 2024-09-11 11:38:13

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：如何播放视频或摄像头画面？如何播放序列帧动画？这些都会在系列文章进行解答。

本篇文章将简单介绍阿拉伯文本相关整形与排序规则，接着介绍在 AWStudio 设置阿拉伯语言翻译的步骤。

阿拉伯文本整形规则

一般 GUI 显示英文或者中文时，内存中存储的字符串和我们人眼看到的字符串是一样的；但是阿拉伯文本比较特殊，内存中的字符串需要经过几个步骤处理（主要是整形与文本方向排序），最终才形成成我们人眼看到的字符串。

关于阿拉伯文本整形可以分为 5 种，下面将简单介绍这几种类型的整形。

1、字母位置不同，如在开头或结尾可能会导致不一样的变形：

Figure 29: Letters join by taking their relevant form.

图 1 字母位置变形（Shaper）

2、字母会受到前后字母影响而变形，如下图两个字符会结合形成第三个字符：

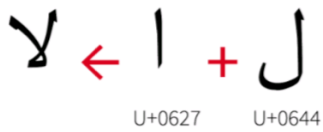


图 2 连写变形（Ligature）

3、元音符号和字母会有组合的特性，如下图，一个字母与一个元音符号会有位移结合的变形：

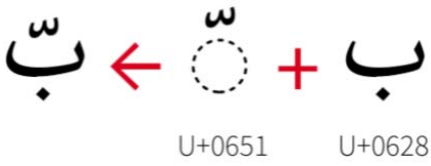


图 3 元音符号变形（Diacritics）

4、有时候为了美观的视觉效果会将字母延长凑成一整行的长度，这种是字母延长变形：

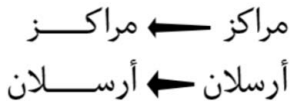


Figure 43: Two words extended with kashida.

图 4 字母延长变形（Kashida）

5、字母之间插入连接符号会影响变形：

Figure 47: Tatweel.

图 5 字母插入连接符号变形（Tatwell）

阿拉伯文本双向排序

Unicode不同类型字符的方向属性

Unicode 字符可以分为强字符、弱字符和中性字符，不同类型的字符具有不同的方向性特征：

1、强字符：英文、汉字、阿拉伯字母等，它们方向性都是确定的，比如英文属于从左到右的 LTR 字符，阿拉伯属于从右到左的 RTL 字符，它们可能影响前后字符的方向性；

2、弱字符：数字和数字相关的符号，它们方向性是确定的，但是对前后字符不会产生影响；

3、中性字符：大部分标点符号和空格，它们方向是不确定的，由上下文 bidi 属性决定其方向。

AWTK支持阿拉伯文本的哪些规则

由于阿拉伯文本整形规则和排序规则比较复杂，AWTK 内部调用了 bidi 算法对阿拉伯文本进行排序并做简单的变形，bidi 算法支持文本双向排序算法，在整形规则上暂时只支持字母位置变形和连写变形，不支持元音符号变形。

如何在AWTK使用阿拉伯语言

如果想在 AWTK 正确使用阿拉伯语言，可以参考以下步骤：

下载阿拉伯语言字库

AWStudio 新建工程的默认字库是不支持显示阿拉伯文本的，因此需要先找到支持阿拉伯文本字库文件。字库文件可以在相关字体下载网站搜索并下载，或者从 awtk 资源目录拷贝自带的阿拉伯语言字库文件：

awtk/design/default/fonts/trado.ttf

导入阿拉伯语言字库到工程

准备好字库文件后，可以使用 AWStudio 将字库文件导入到自己的项目工程。在 AWStudio 资源浏览器选择“字体”，点击右上角加号选择字库文件并导入。

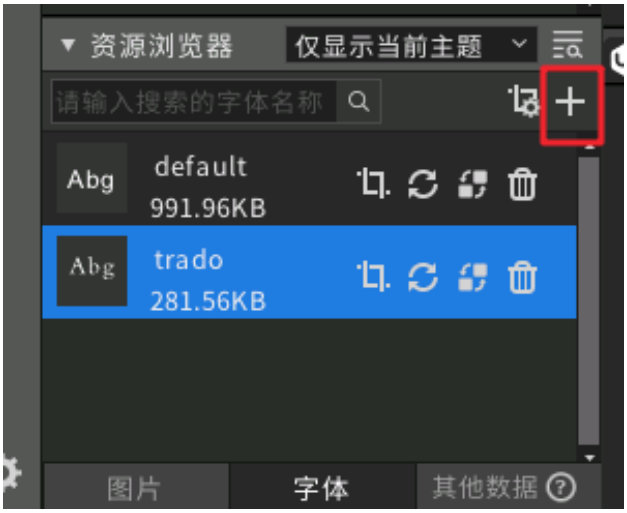


图 6 使用AWStudio导入字库文件

在AWStudio设置阿拉伯语言翻译文本

AWStudio 支持给工程设置多国语言翻译并实时切换的功能，点击 AWStudio 上方菜单栏的“翻译”按钮进入多语言翻译设置页面，页面操作步骤如下：

1、点击界面右下角添加语言按钮；

2、在对话框选择要添加的语言、国家或地区，比如阿拉伯语言选择“Arabic”；

3、点击对话框确定按钮；

4、在对应语言框输入翻译后的文本内容。（**如果显示异常可能是字体缺失，可以在添加语言对话框点击“导入字体”添加字库文件**）

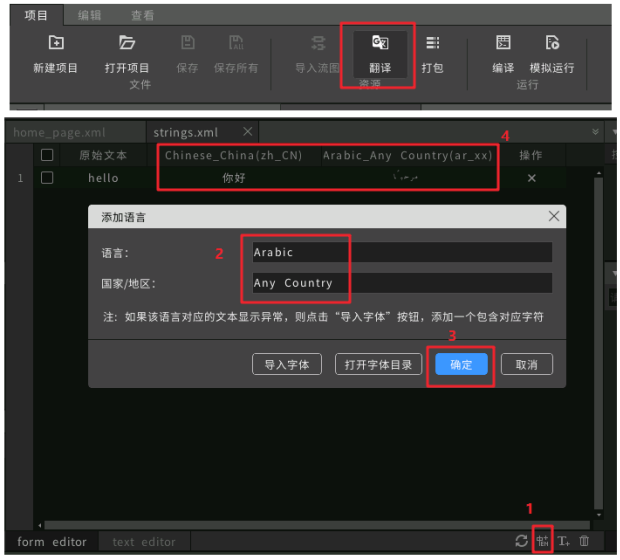


图 7 使用AWStudio添加阿拉伯语言翻译

最后在 AWStudio 给控件 text 属性设置原始文本，并且开启翻译选项，之后该控件就会根据程序当前语言翻译进行翻译。

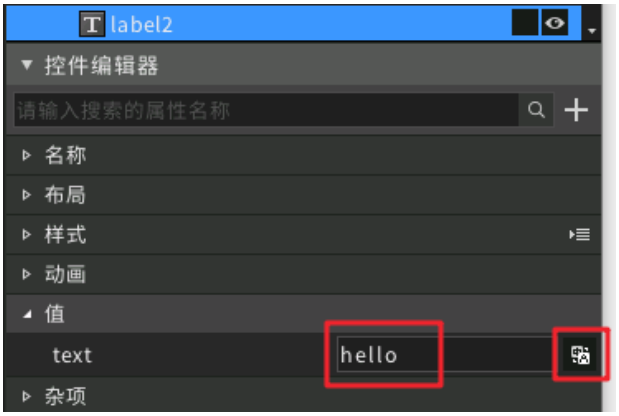


图 8 开启控件翻译选项

关于在程序实时翻译的详细步骤可以参考官方文档《多国语言互译》里面的“实时切换语言”章节：

https://awstudio.zlg.cn/docs/awtk_docs/AWTK_Guide/8.Locale.html

阿拉伯语言方向性问题

由于阿拉伯书写习惯是从右到左，在切换语言后同样需要遵循这个习惯。AWTK 内置的 bidi 算法可以自动处理阿拉伯文本排序，也可以手动定义 bidi 属性（默认为 auto），如强制设置 RTL 排序。关于 bidi 双向排序算法详细的使用说明可以阅读官方在线文档《如何启用文本双向排版》：

https://awstudio.zlg.cn/docs/awtk_docs/HowTo/how_to_enable_bidi.html

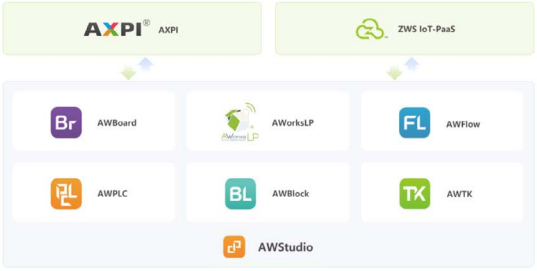
图 9 bidi属性自动排序阿拉伯文本

【技术平台】EsDA，一站式嵌入式软件

ZLG 致远电子 2024-09-14 11:36:42

EsDA 是一套面向工业智能物联领域的嵌入式系统设计自动化工具集，包含实时操作系统 AWorksLP、低代码开发平台 AWStudio、资源管理平台 AXPI、跨平台 GUI 引擎 AWTK 和云服务平台 ZWS，旨在提高嵌入式软件开发的效率、性能和可扩展性。

EsDA 全称是嵌入式系统设计自动化，它是一个由 AWorksLP、AWStudio、AXPI、AWTK、ZWS 等多种软件组成的软件过程管理和开发工具，致力于解决工业智能物联行业嵌入式软件开发过程中的各种问题。了解更多：https://www.zlg.cn/esda/esda/index.html

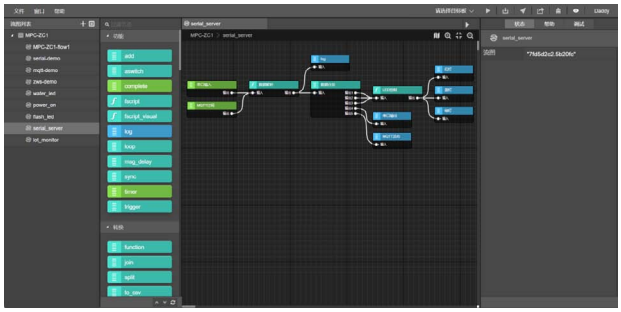


1. AWorksLP

AWorksLP 是公司自主研发的物联网实时操作系统，同时也是 ESDA 生态底层的基础技术平台，具备轻量级硬实时、低延时、低功耗等特性。一方面作为产品开发平台服务于公司的研发活动，另一方面嵌入公司产品中作为软件生态发挥作用，从而通过底层架构的优化为产品开发效率及产品性能水平的提升奠定了基础。在时延、功耗及对各类协议、硬件的兼容性上达到了国内先进水平。了解更多：
https://www.zlg.cn/aworks

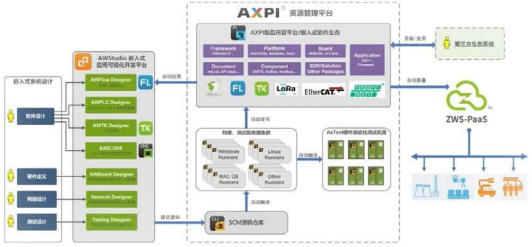
2. AWStudio

AWStudio 嵌入式和自动化应用开发平台，可实现应用的可视化、自动化及组件化，低代码开发以及业务逻辑的图形化拖拽式开发，有效降低了开发门槛，提高了开发效率和软件的复用性。以 AWStudio 的逻辑开发组件 AWFlow 为例，它以拖拽节点绘制数据流图的方式将硬件设备、软件模块、网络服务等连接在一起，最终完成整个应用业务的设计和开发，旨在帮助开发者通过图形化界面和少量编码来快速构建应用程序，使更多非技术人员能够参与应用程序开发。了解更多：
https://awtk.zlg.cn/pro/docs



3. AXPI

AXPI 资源管理平台是 EsDA 生态的软件包管理和分发中心，为开发、构建、测试、发布、部署等软件开发活动的所有环节提供各种软件制品的管理和分发服务。AXPI 可实现跨平台的代码复用，在外设、协议多样化的工业物联场景，对产品开发过程及产品的兼容性均具有重要的价值。了解更多：
https://axpi.zlgcloud.com/#/Container/HomePage



4. AWTK

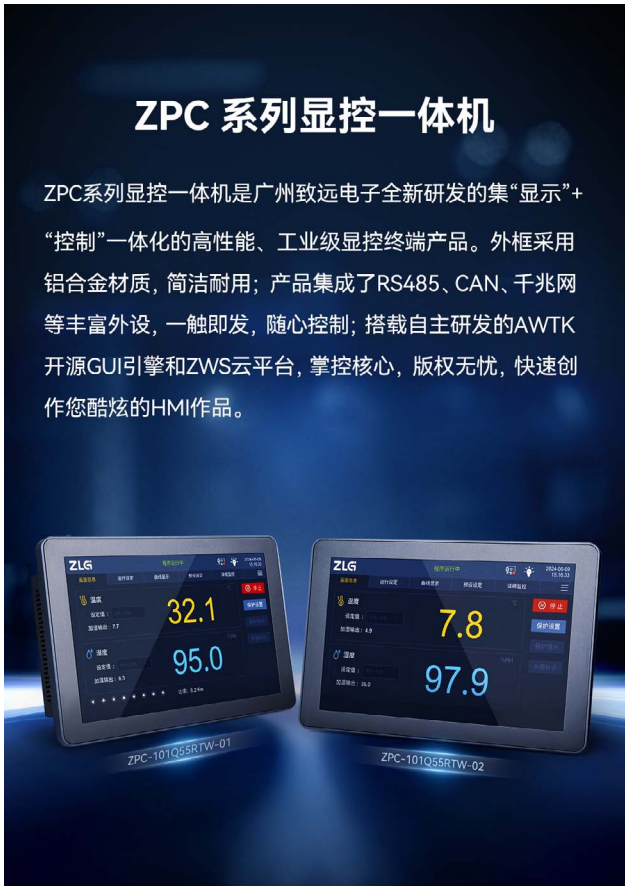
AWTK 是针对工业智能物联场景推出的图形用户界面引擎，可兼容 Android、iOS、Windows、Linux、MacOS 等平台，并基于公司自研算法在各类平台利用较小的资源即可实现较好的显示与交互效果，并且提供配套的界面设计器，通过简单的控件拖拽和配置即可轻易地创建出人机交互界面应用。了解更多：https://www.zlg.cn/index/pub/awtk.html

5. ZWS

万物互联的时代，嵌入式设备普遍需要接入云端统一管理，ZWS 云平台基于强大的物模型建模能力，可以为客户提供多协议设备的快速接入、设备管理、事件告警、数据统计分析、组态形象化数据展示等功能，帮助企业快速低成本搭建行业 SaaS 应用系统。了解更多：
https://www.zlgcloud.com/website/home/main

【新品发布】ZPC系列高性能显控一体机震撼登场！

ZLG 致远电子 2024-09-02 11:38:41



选型表

参数 \ 型号	ZPC-101Q55RTW-01	ZPC-101Q55RTW-02
处理器内核	4*Arm® Cortex®-A55	4*Arm® Cortex®-A55
主频	2.0 GHz	2.0 GHz
NPU	1 TOPs	1 TOPs
内存	4GB LPDDR4	2GB LPDDR4
电子硬盘	32GB eMMC	16GB eMMC
操作系统	Debian	Debian
屏幕尺寸	10.1 英寸	10.1 英寸
显示分辨率	1280*800	1280*800
亮度	500nit	500nit
触摸屏	4 线电阻	4 线电阻
WiFi/BT	支持	支持
4G/5G	支持	不支持
SSD	支持	支持
RS232	2 路（调试 + 通讯）	1 路（默认调试）
RS485	4 路，隔离电压 3500V DC	2 路，隔离
CAN-Bus	3 路，隔离电压 2500VDC	不支持
NET	2 路千兆 +1 路百兆	2 路千兆
USB2.0	1 路 A 口	1 路 A 口
USB3.0	1 路 A 口	1 路 A 口
Type-C	1 路（共用 USB3.0）	1 路（共用 USB3.0）
HDMI	1 路 标准 A 型母座	1 路 标准 A 型母座
TF 卡	1 路	1 路
Nano SIM 卡	1 路	不支持
DO	8 路（继电器隔离）	2 路（继电器隔离）
DI	8 路（5 路湿接点 +3 路干接点）	2 路（湿接点）
RTC 时钟	支持	支持
独立看门狗	支持	支持
掉电检测	支持	支持
掉电续航	支持	支持
供电电压	9V~24V DC	9V~24V DC
机械尺寸	255*169*70.5mm	255*169*57.5mm
工作环境	-10℃~+70℃	-10℃~+70℃

开源GUI引擎，带来非凡交互体验

ZPC 系列高性能显控一体机搭载致远电子自主研发的 AWTK 开源 GUI 引擎。AWTK 基于 C 语言开发，旨在为用户提供一个功能强大、高效可靠、可轻松做出高流畅性、交互效果好的 GUI 引擎，并开创性的支持跨平台同步开发，实现一次编程，终生使用。

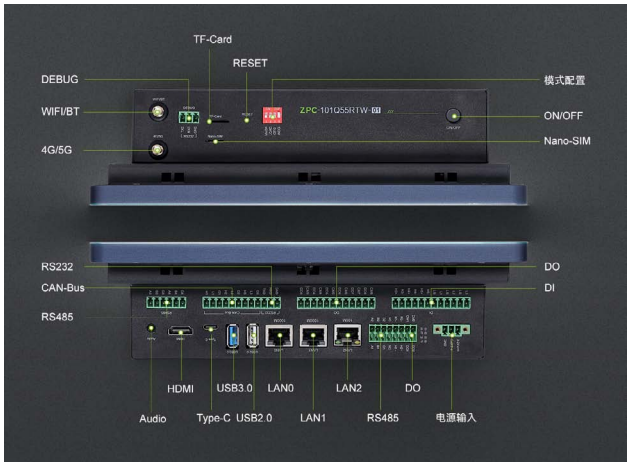
支持ZWS物联网云平台，快速实现系统智能化升级

基于致远电子自主研发的 ZWS 物联网云平台，让设备实现智能化升级，支持远程固件升级、日志文件快速召回业务数据监控、组态可视化呈现、海量数据分析等功能。

集成多种工业通信协议，让工业互联更轻松

海量外设，随心连接

集成隔离 RS485、RS232、隔离 CAN、USB、以太网等丰富外设，支持可选 4G/5G、SSD 海量存储，满足不同的应用场景需求。



IP54防护，使用很安心

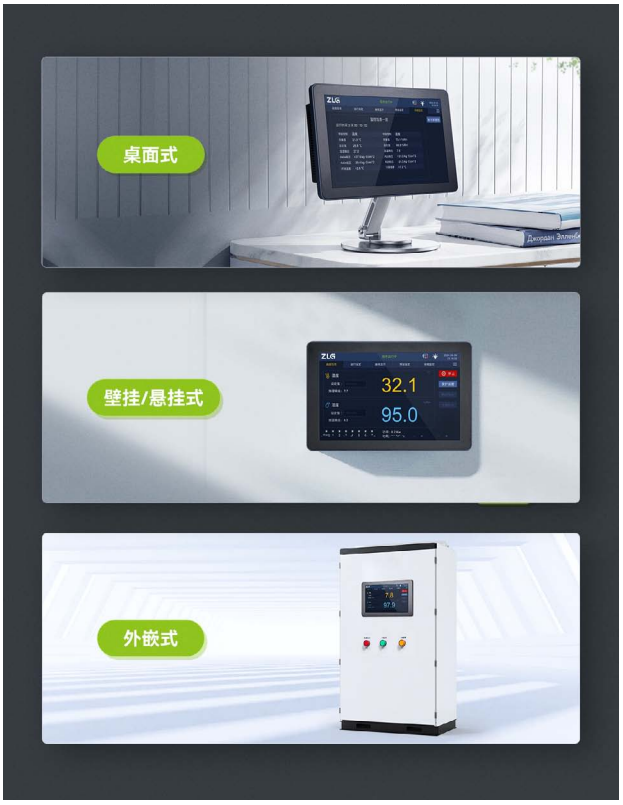
产品显示面板部分经过 GB/T 4208-2017 标准测试，具备 IP54 防尘防水等级标准，用户使用更安心。

无限视界，双面精彩

ZPC 系列高性能显控一体机支持 HDMI 输出 4K@60fps 显示配合自带的 10.1 英寸高清屏可实现双屏异显，精彩不止一面！

边缘计算▼

多种安装方式，满足各类场景



工控触摸屏ZPC系列

点击购买

行业应用

【技术分享】一文教你在ZPC上快速移植AWTK

ZLG 致远电子 2024-09-03 11:36:56

ZPC 是 ZLG 全新研发的显控一体机。开源 AWTK，版权无忧！多种工业通信协议，工业互联无壁垒！ ZWS 数据上云很轻松！更有 AWFlow，应用开发很简单！本文将介绍如何在 ZPC 上快速移植 AWTK。

ZPC简介

ZPC 系列显控一体机 是广州致远电子全新研发的集“显示”+“控制”一体化的高性能显控终端产品。外框采用铝合金材质，简洁耐用；产品集成了多路隔离 RS485、多路隔离 CAN 总线、多路千兆以太网等丰富外设。一触即发，随心控制。产品支持多种工业通信协议，工业互联无壁垒！支持 ZWS，数据上云很轻松！支持 AWFlow，应用开发很简单！还支持开源 AWTK GUI，版权无忧！可以快速创作您的 HMI 作品。

AWTK简介

AWTK 全称为 Toolkit AnyWhere，是 ZLG 倾心打造的一套基于 C 语言开发的开源 GUI 框架。旨在为用户提供一个功能强大、高效可靠、简单易用、可轻松做出炫酷效果的GUI引擎; 支持跨平台同步开发，一次编程，随处编译，跨平台终身使用，无版权费用担忧！

准备工作

- 装有 Ubuntu 系统或 Ubuntu 虚拟机的可连接外网的 PC 1 台；
- ZPC-101Q55RTW-01 显控一体机 1 台；
- USB 转 RS232 调试串口 1 个；
- 12V@2A 的电源适配器 1 个；
- MobaXterm 串口调试上位机；
- TF 卡或 U 盘或网线等。

移植过程

1. 环境准备

AWTK 代码已在 gitee 和 github 上开源，国内推荐访问速度更快的 gitee：

https://gitee.com/zlgopen/awtk

按照以下步骤，在 PC 虚拟机上进行操作：

```
host$ sudo apt install gcc-aarch64-linux-gnu git scons
// 使用 apt 指令下载所需的工具
host$ mkdir awtk-demo && cd awtk-demo
// 创建一个新的文件夹
host$ git clone https://gitee.com/zlgopen/awtk.git
// 使用 git 克隆 awtk 仓库到本地，awtk 为 AWTK 代码本仓，包含了 AWTK 最主要的代码
host$ git clone https://gitee.com/zlgopen/awtk-linux-fb.git
// 使用 git 克隆 awtk-linux-fb 仓库到本地，awtk-linux-fb 为针对嵌入式的 Frame buffer 机制和 DRM 机制的仓库
host$ ls
awtk  awtk-linux-fb
host$ cd awtk-linux-fb
```

2. 安装依赖库

Ubuntu 上编辑 /etc/apt/sources.list 文件，将它改为如下内容：

```
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal universe
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-updates universe
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal multiverse
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-updates multiverse
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-security universe
deb [arch=amd64] https://mirrors.ustc.edu.cn/ubuntu/ focal-security multiverse
```

Ubuntu 上编辑 /etc/apt/sources.list.d/ubuntu-ports.list ，如果文件不存在，则创建，将它改为如下内容：

```
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal main restricted
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-updates main restricted
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal universe
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-updates universe
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal multiverse
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-updates multiverse
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-backports main restricted universe multiverse
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-security main restricted
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-security universe
deb [arch=arm64,armhf] https://mirrors.ustc.edu.cn/ubuntu-ports/ focal-security multiverse
```

由于 AWTK 触摸功能依赖于 tslib 触摸库，所以 Ubuntu 需要下载 ARM64 架构的 tslib。

```
host$ sudo dpkg --add-architecture arm64
host$ sudo apt update
host$ sudo apt install libts0:arm64 libts-dev:arm64 libdrm-dev:arm64
```

3. 修改代码

注：以下修改在awtk-linux-fb目录下

打开 awtk_config.py 文件，该文件会描述并配置 AWTK 工程构建时所需环境变量，按照以下内容进行修改，将构建工程配置为 DRM 模式而非 Frame buffer 模式。

```
...
#LCD_DEVICES='fb'
LCD_DEVICES='drm'
```

修改 tslib 依赖库在本机的位置，Ubuntu 默认位置如下：

```
...
TSLIB_LIB_DIR='/usr/lib/aarch64-linux-gnu/ts0'
TSLIB_INC_DIR='/usr/include'
...
```

修改编译工具链的指令，再次选择 Ubuntu 下的 arm64 交叉编译工具链：

```
...
TOOLS_PREFIX='aarch64-linux-gnu-'
...
```

可以使用以下指令确认交叉工具链版本，在 Ubuntu 20.04 下默认的交叉工具链版本为 9.4.x：

```
host$ aarch64-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc-cross/aarch64-linux-gnu/9/lto-wrapper
Target: aarch64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 9.4.0-1ubuntu1~20.04.2' --with-bugurl=file:///usr/share/doc/gcc-9/README.Bugs --enable-langu
```

```
ages=c,ada,c++,go,d,fortran,objc,obj-c++,gm2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-9 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libquadmath --disable-libquadmath-support --enable-plugin --enable-default-pie --with-system-zlib --without-target-system-zlib --enable-libpth-m2 --enable-multiarch --enable-fix-cortex-a53-843419 --disable-werror --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=aarch64-linux-gnu --program-prefix=aarch64-linux-gnu- --includedir=/usr/aarch64-linux-gnu/include
    Thread model: posix
    gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.2)
```

config 文件夹下有文件 devices.json.in 这一配置文件，该文件配置了 AWTK 运行时所需物理设备的路径，但在名为 devices.json.in 时不生效，需要按照以下命令修改名称为 devices.json；再打开 config/devices.json，修改为以下内容，其中 event1 为触摸屏设备路径，类型为 ts（表示为触摸屏）。

```
host$ mv config/devices.json.in config/devices.json
{
  "/dev/fb0" : {
    "type" : "fb"
  },
  "/dev/dri/card0" : {
    "type" : "drm"
  },
  "/dev/input/event1" : {
    "type" : "ts"
  }
}
```

AWTK 默认使用鼠标设备进行操作，所以默认情况下界面会有光标显示，在使用触摸屏操作的时候会影响用户体验，如若需要关闭鼠标光标，则需要在 scons_argv.py 文件中，找到 ENABLE_CURSOR 定义，并将其 value 值改为 False。

```
...
 'ENABLE_CURSOR' : { 'value' : False, 'type' : bool.__name__, 'desc' : ['enable cursor mouse'], 'help_info' : 'set enable cursor mouse, value is true or false'},
...
```

至此，代码修改完成。

4. 编译代码

执行 scons 指令进行编译工程。

```
host$ scons
```

编译完成后会输出 log 下图所示，如果编译过程中出现寻找不到对应头文件，请返回上文安装依赖库。

接下来执行 ./release.sh 脚本对文件资源进行打包，执行完成后，可以看到生成 release 文件夹和 release.tar.gz 压缩文件。release 文件夹中，包含了编译生成的可执行文件、awtk 库以及相关配置文件，也就是应用程序运行时所需的文件；而 release.tar.gz 就是对 release 文件夹的打包。

```
host$ awtk-linux-fb ./release.sh
EXE_NAME = demoui
APP_ROOT = ../awtk
================================================
EXE_NAME:demoui
ASSETS_DIR:../awtk/res/assets
OUTPUT_DIR:/home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/release
BIN_DIR:/home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/bin
================================================
/home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/bin/demoui->/home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/release/bin/demoui
copy shared lib: /home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/bin/libtkc.so ==> /home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/release/bin/libtkc.so
copy shared lib: /home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/bin/libawtk.so ==> /home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/release/bin/libawtk.so
../awtk/res/assets->/home/felix/Workspace/Testcodes/awtk-demo/awtk-linux-fb/release/assets

host$ awtk-linux-fb ls -l release.tar.gz
-rw-rw-r-- 1 felix felix 8262509 7 月  9 09:52 release.tar.gz
```

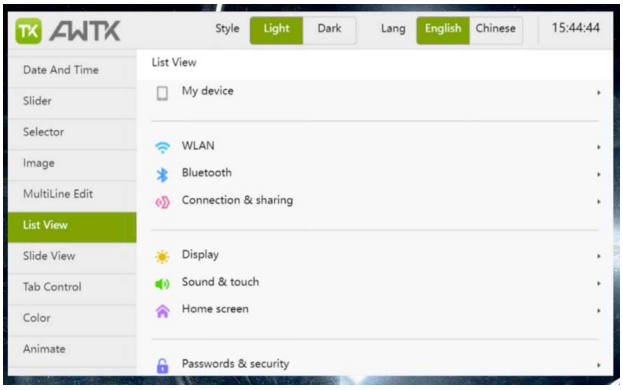
执行示例

拷贝 release.tar.gz 文件到 ZPC 一体机的 /root 文件夹下，并按照以下指令分别执行对应解压命令解压包、设置执行权限等。

```
[root@zpc:~]# tar -xvf release.tar.gz
......
[root@zpc:~]# chmod +x release/bin/demoui
```

执行以下指令，即可运行示例 demo 如下图所示。

```
[root@zpc:~]# ./release/bin/demoui
```



设置开机自启动

默认系统会进入 Debian 系统的 LXDE 界面，当使用 AWTK 进行开发时，不需要原生桌面的情况下，可以执行以下指令关闭原生桌面，提高启动速度。

```
[root@zpc:~]# systemctl disable lightdm
Synchronizing state of lightdm.service with SysV service script with /lib/systemd/systemd-....
Executing: /lib/systemd/systemd-sysv-install disable lightdm
Removed /etc/systemd/system/display-manager.service.
```

基于 systemd 创建属于 awtk 的运行服务，创建 /usr/lib/systemd/system/awtkdemo.service 文件并修改为以下内容：

```
[Unit]
Description=AWTK demo
After=network.target

[Service]
Type=simple
ExecStart=/root/release/bin/demoui # 这里是运行路径，根据实际情况修改
Restart=always
User=root
Group=root

[Install]
WantedBy=multi-user.target
```

然后执行以下指令开启 awtk 的运行服务。

```
[root@zpc:/usr/lib/systemd/system]# systemctl enable awtkdemo
Created symlink /etc/systemd/system/multi-user.target.wants/awtkdemo.service → /lib/system...
```

执行 reboot 指令后，则可看到系统启动时没有进入本来的 LXDE 桌面，而是直接启动 AWTK 示例。

【技术分享】一文搞懂用ZPC轻松拿捏数据上云

ZLG 致远电子 2024-09-04 11:36:27

ZPC 是 ZLG 全新研发的显控一体机。开源 AWTK，版权无忧！ AWFlow 流图编程，开发很简单！多种通信协议，设备互联超便捷！更有 ZWS，数据上云很轻松！本文将介绍 ZPC 轻松拿捏数据上云。

ZPC简介

ZPC 系列显控一体机 是广州致远电子全新研发的集“显示”+“控制”一体化的高性能显控终端产品。外框采用铝合金材质，简洁耐用；产品集成了多路隔离 RS485、多路隔离 CAN 总线、多路千兆以太网等丰富外设。一触即发，随心控制。产品支持多种工业通信协议，工业互联超便捷！支持拖拽式开源 AWTK GUI，版权无忧！支持 AWFlow 流图编程，应用开发很简单！还支持 ZWS 云，数据上云很轻松！

ZWS简介

ZWS IoT-PaaS 云平台 ，是一个开放的物联网云平台，可以为用户提供多种协议设备的快速接入、设备管理、事件告警、数据统计分析等功能，帮助企业快速实现低成本搭建行业 SaaS 应用系统，助力企业实现数字化转型。

原理介绍

ZPC 拥有多路隔离 RS485、多路隔离 CAN 总线、多路 DI/DO 以及多路千兆以太网等丰富的外设接口资源。为了避免大家手头没有 RS485 传感器，本次我们将演示 ZPC 一机搞定数据轻松上云。原理是我们在 ZPC 上编程从 RS-485-3 接口输出 1~100 的数据，RS-485-4 接口采集这些数据然后通过以太网接口采用 MQTT 协议上传到 ZWS 云。

准备工作

- ZPC-101Q55RTW-01 显控一体机 1 台；
- 安装 Ubuntu 虚拟机或 Ubuntu 系统的 PC 机 1 台；
- USB 转 RS232 调试串口 1 个；
- 12V@2A 的电源适配器 1 个；
- MobaXterm 上位机软件；
- 双公头网线 1 根；
- 杜邦线若干。

在开始前，我们先使用杜邦线将 RS485-3 和 RS485-4 接口连通，如下图所示位置（A3 接 A4，B3 接 B4）。

调试串口连接到 ZPC，网线连接到可以上网的电脑，然后给 ZPC 插上适配器上电。通过 MobaXterm 或者其它串口调试上位机登录 ZPC，波特率为 115200，8 位数据位，1 位停止位，无奇偶校验，无流控。超级用户账号密码同为 root。然后使用 ping 指令进行测试对外网的连通情况，如下图所示即为可以上网。

```
[root@zpc:~]$ ping www.baidu.com
PING www.a.shifen.com (183.2.172.42) 56(84) bytes of data.
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=1 ttl=49 time=12.5 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=2 ttl=49 time=42.4 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=3 ttl=49 time=44.4 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=4 ttl=49 time=39.4 ms
```

ZWS建模

首先，需要准备一个 ZWS 云平台账号，注册并登录进入官网后，可以看到类似界面如下图所示。ZWS 云平台官网链接：

https://www.zlgcloud.com/website/home/main

（注册账号的过程在这里暂不展开，需要可以联系对应的销售工程师）

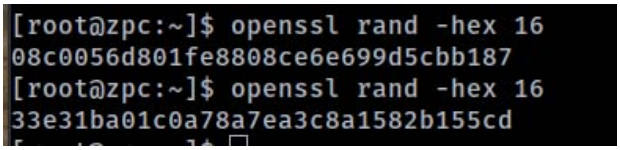
接着需要对设备进行建模，建模的意义在于创建一个设备类型，方便后续对同一类型设备和数据的管理。我们可以将 ZPC 一体机视为一个网关设备，也可以作为一个普通设备。如下图所示，依次点击设备建模、设备类型、自定义、添加设备类型。

如下图所示填写所需的设备类型信息。其中类型名称只可为英文、数字或者一些特殊符号。

填写完成后，即可在界面看到所创建的设备类型。

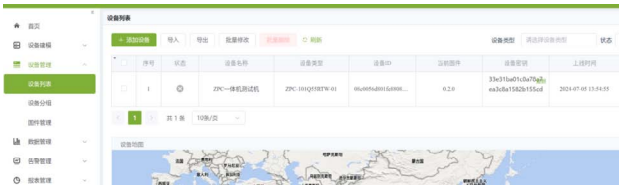
创建完成设备类型后，还需要添加一个实际的设备。如下图所示，我们依次点击设备管理、设备列表、添加设备。

然后这里需要填写设备类型、设备名称、设备 ID 和设备密钥。其中，ID 和密钥我们可以通过串口助手使用 openssl rand -hex 16 命令在 ZPC 一体机上生成一个长度为 32 的十六进制随机数来使用，如下图所示。



最后填写完毕设备信息如下图所示。（实际生产过程中，推荐使用机器序列号或其他唯一标识作为 ID）。

填写完成并确认后，可以在设备列表中看到创建完成的设备。



接下来，需要对上报的数据进行编辑。我们点击设备建模、设备类型、自定义，找到刚才创建的设备类型，点击编辑。

这里显示有两个功能组，其中 common_fn 是全局通用功能模块，其中包含上报原始数据（raw_data）的功能（本文不涉及，暂不展开）；ZPC_101Q55RTW_01_fn 是针对本次创建的设备类型的功能模块，点击 ZPC_101Q55RTW_01_fn 的编辑按钮。

创建对应的数据组和数据字段。

数据组可以将所有的同类型数据作为一个数据组。比如使用 ZPC 一体机进行检测被测设备的压力值，在此将压力值 (pressure) 作为一个数据组如下图所示。

而数据字段代表了数据组下的一个数据字段内容。比如检测多个被测设备时，可以将某一节点作为一个字段，在此模拟将节点 1 的压力值作为一个 pressure_node1 数据字段，并将其设置为 int 类型数据，如下图所示。

至此，在 ZWS 上的准备工作完毕。

应用编程

1. 搭建数据上云

首先下载 ZWS SDK 包到 Ubuntu 虚拟机下（建议使用 ubuntu20.04），并安装 gcc-aarch64-linux-gnu 包。解压 SDK 包到工作目录后，进入到 zws_sdk_r2/mqttproto_V2.0.3.240626 目录。在根目录下创建一个新的 zpc.mk 文件，用于构建工程使用，程序清单如下。ZWS SDK 包下载地址：

https://zlgcloud.oss-cn-shenzhen.aliyuncs.com/zws_sdk_r2.zip

```
MESSAGE(STATUS "BUILD FOR ZPC")
#across compiler setting
#include(CMakeForceCompiler)

set(CMAKE_SYSTEM_NAME Linux)
set(CMAKE_SYSTEM_PROCESSOR arm)

set(CMAKE_C_COMPILER aarch64-linux-gnu-gcc)
set(CMAKE_CXX_COMPILER aarch64-linux-gnu-g++)

#set(CMAKE_FIND_ROOT_PATH ${TOOL_CHAIN_DIR})
set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)
set(CMAKE_FIND_ROOT_PATH_MODE_LIBRARY ONLY)
set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)
set(CMAKE_FIND_ROOT_PATH_MODE_PACKAGE ONLY)
```

在工程根目录的 CMakeLists.txt 内添加以下内容，用于调用上文中的 zpc.mk。

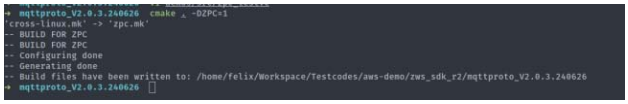

```
...
# 内置对 zpc-101q55rtw-01/zpc-101q55rtw-02 的支持
if(ZPC)
#ADD_DEFINITIONS(-DARM_LINUX=1) # 经 验 证 ADD_DEFINITIONS 定义的变量，然后同一个文件引用该变量是无效的
set(ARM_LINUX 1)
execute_process(COMMAND ln -svf zpc.mk cross-linux.mk)
endif()
...
```

创建测试代码文件，路径为 demos/src/zpc_test.c，详见在线文档 https://manual.zlg.cn/web/#/331/13023 程序清单 2 的内容，实现了从 RS-485-4（对应串口 /dev/ttyS8）读取数据，并上报到 ZWS，详细可以参考代码注释。

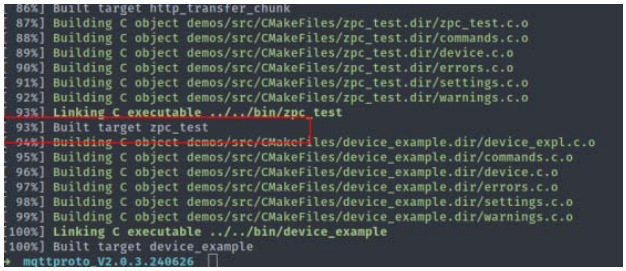
并在 demos/src/CMakeLists.txt 中添加如下内容用于编译测试 demo。

```
...
set(PRJ zpc_test)
add_executable(${PRJ} "")
target_sources(${PRJ} PRIVATE zpc_test.c ${shared_src})
set_target_properties(${PRJ} PROPERTIES FOLDER "demos")
target_link_libraries(${PRJ} client network ${SYS_LIBS})
```

接下来执行 cmake . -DZPC=1，构建编译工程。



执行 make 指令，编译工程。可以看到成功构建 zpc_test 文件（如果不成功，请检查上述步骤）。



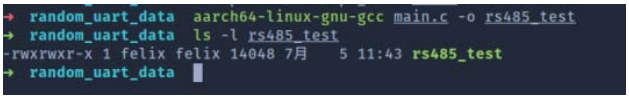
2. 搭建模拟数据

接下来，还需要实现从 RS-485-3 生成随机数据并传输。另外创建一个文件夹（最好和前文的工程分开），在文件夹下创建 main.c 文件，程序清单如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <fcntl.h>
#include <unistd.h>
#include <termios.h>
// 配置串口
int configure_serial_port(int fd) {
struct termios tty;
if (tcgetattr(fd, &tty) != 0) {
perror("tcgetattr");
return -1;
}
// 设置波特率
cfsetospeed(&tty, B9600);
cfsetispeed(&tty, B9600);
// 设置字符大小、无奇偶校验、一个停止位
tty.c_cflag &= ~PARENB; // 无奇偶校验
tty.c_cflag &= ~CSTOPB; // 一个停止位
tty.c_cflag &= ~CSIZE;
tty.c_cflag |= CS8; // 8 个数据位
// 设置为非规范模式
tty.c_lflag &= ~ICANON;
tty.c_lflag &= ~ECHO;
tty.c_lflag &= ~ECHOE;
tty.c_lflag &= ~ISIG;
// 禁用软件流控制
tty.c_iflag &= ~(IXON | IXOFF | IXANY);
// 禁用硬件流控制
tty.c_cflag &= ~CRTSCTS;
// 设置读取阻塞行为
tty.c_cc[VMIN] = 1;
tty.c_cc[VTIME] = 0;
// 刷新串口设置
if (tcsetattr(fd, TCSANOW, &tty) != 0) {
perror("tcsetattr");
return -1;
}
return 0;
}
int main() {
int fd;
int random_number;
char buffer[4];
// 打开串口设备
fd = open("/dev/ttyS7", O_WRONLY | O_NOCTTY);
if (fd == -1) {
perror("open");
return -1;
}
// 配置串口
if (configure_serial_port(fd) != 0) {
close(fd);
return -1;
}
// 初始化随机数生成器
srand(time(NULL));
while (1) {
// 生成 1 到 100 之间的随机数
random_number = rand() % 100 + 1;
snprintf(buffer, sizeof(buffer), "%d\n", random_number);
// 向串口发送随机数
```

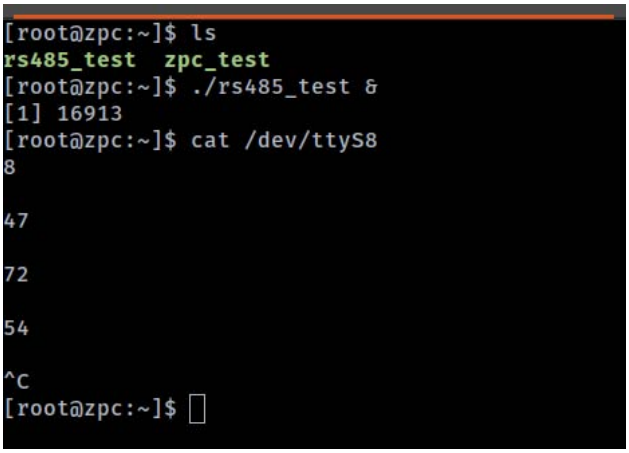
```
if (write(fd, buffer, sizeof(buffer)) == -1) {
perror("write");
close(fd);
return -1;
}
// 延时 1 秒
sleep(1);
}
// 关闭串口设备
close(fd);
return 0;
}
```

保存文件后，执行 aarch64-linux-gnu-gcc main.c -o rs485_test 生成测试文件。

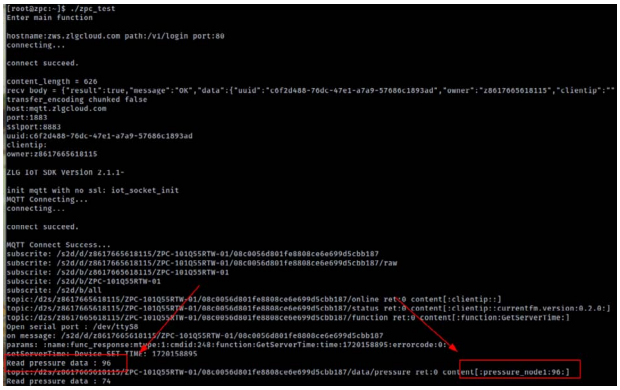


验证数据上云

将上文中生成的测试程序 zpc_test 和 rs485_test 拷贝至 ZPC 一体机，并执行 ./rs485_test &，该程序将从 RS485-3 随机发送 1~100 的数据。由于 RS-485-3（/dev/ttyS7）和 RS-485-4（/dev/ttyS8）短接，可以直接使用 cat /dev/ttyS8 查看采集的数据，如下图所示。



最后执行 ./zpc_test，便开始将 RS-485-4 读取到的数据上报到 ZWS。下图中标注的两点，为采集到的数据。



打开 ZWS IoT 设备管理平台界面，并按下图所示选择对应类别，即可看到上报的实时数据。

也可以使用历史数据功能查看历史数据。

【技术分享】ZPC是如何轻松拿捏严苛工况的?

ZLG 致远电子 2024-09-05 11:34:38

随着新能源的快速发展，储能柜的应用越来越普及。显控一体机作为储能柜中的重要部件之一，在严苛的工况环境中面临着诸多挑战。我们该如何选择合适的显控一体机产品呢?

ZPC简介

ZPC 系列显控一体机 是广州致远电子全新研发的集“显示”+“控制”一体化的高性能显控终端产品。外框采用铝合金材质，简洁耐用；产品集成了多路隔离 RS485、多路隔离 CAN 总线、多路千兆以太网等丰富外设。一触即发，随心控制。产品支持多种工业通信协议，工业互联超便捷！支持拖拽式开源 AWTK GUI，版权无忧！支持 AWFlow 流图编程，应用开发很简单！还支持 ZWS 云，数据上云很轻松！

EMS简介

EMS 即能量管理系统 (Energy Management System)，它是一套集成了软硬件的解决方案，主要实现对储能设备的数据采集和处理、可视化监控、数据存储和统计分析、报警管理、安全防护等功能。它是储能系统中不可或缺的重要组成部分，为电力储能系统的稳定运行和能源的高效利用提供了有力保障。

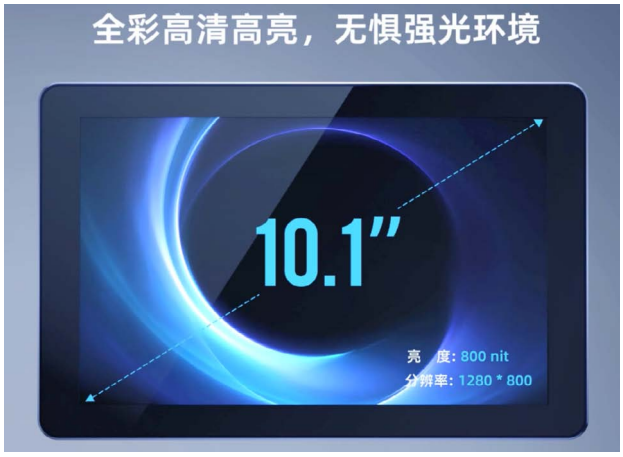
工况因素

1. 高温

储能柜一般都是在户外运行，工作环境都比较恶劣！特别是严酷的夏天，环境温度经常可达 40°C及以上。ZPC 系列显控一体机严格通过 GB/T 2423.2-2008 试验标准测试，产品的元器件都选用的工业级品质，整机可长时间稳定运行在 -20~+70°C的环境中。

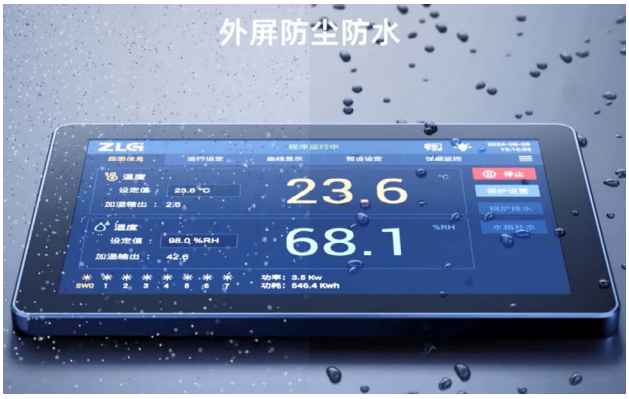
2. 强光

你是否也会遇到在户外手机屏幕看不清的情况？储能柜中的显控一体机也会存在这种挑战。致远电子研发的 ZPC 系列显控一体机选用的行业知名品牌液晶模组，1280*800 的高分辨率，1000:1 的对比度以及高达 800nit 的亮度，远远大于市面常见的四五百尼特，可以在户外强光下有效呈现界面内容。



3. 灰尘雨水

储能柜产品，有一些会安装在偏僻的户外。日晒雨淋、尘土飞扬都是家常便饭。致远电子研发的 ZPC 系列显控一体机产品，外屏采用全贴合压边工艺，严格通过 GB/T4208-2017 防护等级标准测试，达到 IP54 防尘防水等级，可以在户外长时间稳定运行，不受雨水尘埃的影响。



以上便是 3 点常见的选择储能显控一体机的考虑工况因素，你还知道哪些因素吗？快来评论区一起交流吧！

【技术分享】ZPC显控一体机，精彩不止一面！

ZLG 致远电子 2024-09-06 11:35:24

显控一体机的应用，有很多场景会遇到自带显示屏固定不灵活、尺寸不够大等问题。扩展屏幕便是一个很好的解决方案！本文将带您解锁 ZPC 显控一体机的“多面精彩”。

ZPC简介

ZPC 系列显控一体机 是广州致远电子全新研发的集“显示”+“控制”一体化的高性能显控终端产品。外框采用铝合金材质，简洁耐用；产品集成了多路隔离 RS485、多路隔离 CAN 总线、多路千兆以太网等丰富外设。支持多种工业通信协议，工业互联超便捷！支持拖拽式开源 AWTK GUI，版权无忧！支持 AWFlow 流图编程，应用开发很简单！还支持 ZWS 云，数据上云很轻松！更有 HDMI 输出 4K@60fps 显示，配合自带的 10.1 英寸高清屏可实现双屏异显，精彩不止一面！

背景说明

显控一体机的应用中，一般的场景下自带的显示屏即可满足。不过也有很多使用场景局限于自身屏幕固定不灵活、尺寸不够大等，得不到更好的使用。假如显控一体机也支持屏幕扩展，那这个问题将会迎刃而解。致远电子全新研发的 ZPC 显控一体机便具有该功能。

ZPC 自身携带一块 10.1 英寸 1280*800 高分辨率屏幕，同时支持 HDMI 扩展输出 4K@60fps 画面，双屏同显、双屏异显都可以轻松拿捏。

双屏同显

插入 HDMI 外扩屏幕后，执行以下指令，将显示设置为双屏同显。双屏同显情况下两个显示屏会显示一样的内容，可以用于设备屏幕损坏临时调试，或需要外接显示器时使用。

```
xrandr --output HDMI-1  --auto --same-as LVDS-1
```

双屏异显

插入 HDMI 外扩屏幕后，执行以下指令，将显示设置为双屏同显。双屏同显情况下两个显示屏会显示不同的内容，默认情况下，HDMI 显示屏会位于 ZPC 自带屏幕的右边，此时触摸屏无法直接操作 HDMI 屏幕内容，需要外接鼠标进行操作。

```
xrandr --output HDMI-1 --auto --right-of LVDS-1
```

双屏异显模式下支持右侧、左侧、上方、下方四种模式，设置方式分别如下：

```
/* 右侧异显 */
xrandr --output HDMI-1 --right-of LVDS-1 --auto
/* 左侧异显 */
xrandr --output HDMI-1 --left-of LVDS-1 --auto
/* 上方异显 */
xrandr --output HDMI-1 --above LVDS-1 --auto
/* 下方异显 */
xrandr --output HDMI-1 --below LVDS-1 --auto
```

配置修改

可以通过修改 /vendor/scripts/setup-dual-display.sh，修改默认 HDMI 的规则来达成，修改 DIFF_DIS 的值即可。如果要修改异显模式，也可以按照下方示例修改。

```
#!/bin/bash

export DISPLAY=:0
export XAUTHORITY=/home/zlg/.Xauthority

DIFF_DIS=1 # 为 1 是为双屏异显，为其他值时为双屏同显

status=$(cat /sys/class/drm/card0-HDMI-A-1/status)
if [ "$status" == "disconnected" ];then
   echo "HDMI disconnected"
else
```

```
HDMI_OUTPUT=$(xrandr | grep " connected " | grep "HDMI" | awk
'{ print $1 }')
  if [ "${DIFF_DIS}" = "1" ];then
    xrandr --output ${HDMI_OUTPUT} --auto --right-of LVDS-1 # 如
果需要修改异显模式，修改此处即可
  else
    xrandr --output ${HDMI_OUTPUT} --same-as LVDS-1 --auto
  fi
  echo "HDMI connected"
fi

xinput map-to-output ns2009_ts LVDS-1
```

异显实例

在 Debian 系统下通常使用 X11 窗口框架进行编程，以下是一个异显（右侧）的示例，在 ZPC 主屏上显示绿色全屏窗口，在 HDMI 副屏上显示蓝色全屏窗口。代码如程序清单 1 所示。

其中，窗口管理需要借助 wmctrl 工作进行，需要在 zpc 一体机上安装 wmctrl 工具。

程序清单1　测试程序清单

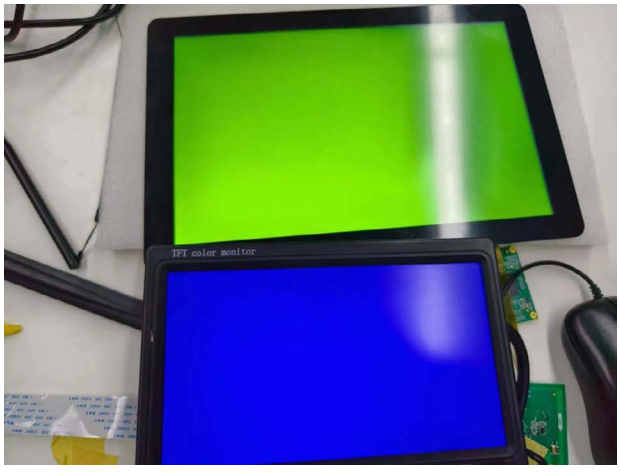
```
[root@zpc:~]# apt update
[root@zpc:~]# apt install -y wmctrl
#include <X11/X.h>
#include <X11/Xlib.h>
#include <X11/extensions/Xrandr.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
// create_window: 创建窗口
Window create_window(Display *display, int screen, int x, int y,
unsigned long background, int width, int height) {
  Window window;
  XSetWindowAttributes attributes;
  attributes.background_pixel = background;
  window = XCreateWindow(
    display, RootWindow(display, screen),
    x, y, width, height, 0,
    CopyFromParent, InputOutput, CopyFromParent,
    CWBackPixel, &attributes
  );
  XMapWindow(display, window);
  XFlush(display);
  return window;
}
// set_window_pos：设置窗口位置
void set_window_pos(Window window, int x, int y)
{
  char command[256];
  snprintf(command, 256, "wmctrl -i -r 0x%lx -e 0,%d,%d,-1,-1",
(unsigned long) window, x, y);
  system(command);
}
// set_window_fullscreen：设置窗口为全屏模式
void set_window_fullscreen(Window window)
{
  char command[256];
  system(command);
   snprintf(command, 256, "wmctrl -ir 0x%lx -b add,fullscreen",
(unsigned long) window);
  system(command);
}
int main() {
  Display *display;
  int screen;
  Window root;
  Window window1, window2;
  XRRScreenResources *res;
  XRROutputInfo *info;
  XRRCrtcInfo *crtc_info;
  int num_outputs;
  int screen_width, screen_height;
  // 获取 X dispay
  display = XOpenDisplay(NULL);
  if (display == NULL) {
    fprintf(stderr, "Unable to open X display\n");
    exit(1);
  }
  // 获取 screen，注意这里两个显示器都位于同一 screen 下
  screen = DefaultScreen(display);
  // 获取根窗口，也就是桌面
  root = RootWindow(display, screen);
// 获取屏幕资源个数，对应显示器个数
  res = XRRGetScreenResources(display, root);
  num_outputs = res->noutput;
  if (num_outputs < 2) {
    fprintf(stderr, "Less than 2 screens detected\n");
    XCloseDisplay(display);
    exit(1);
  }
  // 获取第一个显示器的 CrtcInfo
  info = XRRGetOutputInfo(display, res, res->outputs[0]);
  crtc_info = XRRGetCrtcInfo(display, res, info->crtc);
  screen_width = crtc_info->width;
  screen_height = crtc_info->height;
   printf("first screen width = %d, screen height = %d\n", screen_
width, screen_height);
  // 在第一个显示器上创建窗口，并显示绿色
   window1 = create_window(display, screen, 0, 0, 0x00ff00,
screen_width, screen_height);
  // 获取第二个显示器的 CrtcInfo
 info = XRRGetOutputInfo(display, res, res->outputs[1]);
```

```
 crtc_info = XRRGetCrtcInfo(display, res, info->crtc);
   screen_width = crtc_info->width;
   screen_height = crtc_info->height;
    printf("second screen width = %d, screen height = %d\n",
screen_width, screen_height);
   // 在第二个显示器上创建窗口并显示蓝色
   // 但此时两个窗口会被窗口管理器叠加到主显示器
    window2 = create_window(display, screen, crtc_info->x, crtc_
info->y, 0x0000ff, screen_width, screen_height);
   // 将第二个窗口定位到 HDMI 显示屏
   set_window_pos(window2, crtc_info->x, crtc_info->y);
   // 将两个窗口设置为全屏
   set_window_fullscreen(window1);
   set_window_fullscreen(window2);
   // 保持应用程序运行
   while (1) {
     sleep(1);
   }
   XCloseDisplay(display);
   return 0;
}
```

将程序清单 1 保存为 main.c，并在 Ubuntu（主机或虚拟机）上安装对应库并进行编译。

```
host$ sudo apt install x11-xserver-utils:arm64 libx11-dev:arm64
host$ aarch64-linux-gnu-gcc -o dual_screen main.c -lX11 -lXrandr
```

将编译出来的测试程序 dual_screen 拷贝到 ZPC 一体机上，并执行测试。测试结果如下图所示。



【技术分享】灵活连接，无限可能—探索EtherCAT的拓扑艺术

ZLG 致远电子 2024-09-10 11:42:19

EtherCAT 技术具备快速响应和高效率的特点，在工业自动化领域显得至关重要，其灵活的拓扑结构是其核心优势，支持多样化的网络布局，无需交换机或集线器，简化布线，降低成本，提高系统可靠性和灵活性。

拓扑结构在工业自动化网络中扮演着关键角色，影响着通信的稳定性、速度以及系统的冗余和恢复能力。EtherCAT 技术所展现的灵活的拓扑结构，能够适应不同工业环境的特定需求，通过支持线性、星型、树形等多样化的网络布局，实现高度的网络可靠性和扩展性。

1. EtherCAT支持的拓扑结构

线性拓扑：所有设备串行连接在一条主线上，优点是布线简单，成本较低；但缺点是如果主线损坏，则整个网络可能瘫痪，适用于直线布局的设备连接。

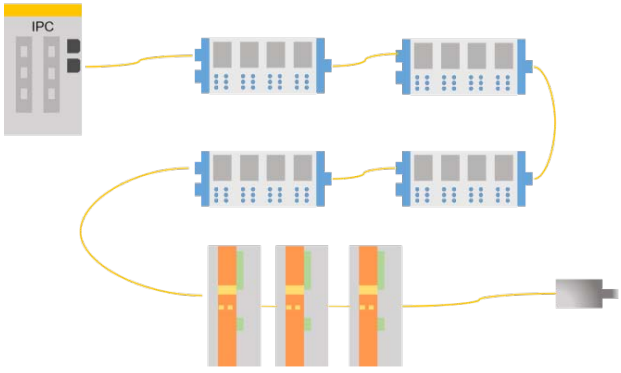


图2 星型拓扑示意图

环形拓扑：设备形成一个闭合的环，每个设备连接到两个相邻设备。环形拓扑的优点是具有容错性，一个节点的故障不会影响到整个网络；缺点是如果环中的某个连接断裂，整个网络将无法正常工作。适用于对网络稳定性有较高要求的场景。

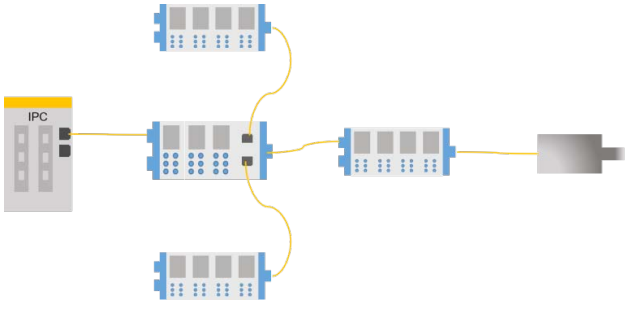


图3 环形拓扑示意图

树形拓扑：类似于树的结构，从一个根节点分出多个分支。树形拓扑的优点是易于扩展，可以覆盖更大的范围；缺点是根节点的故障将导致整个网络中断，适用于需要扩展性复杂的网络。

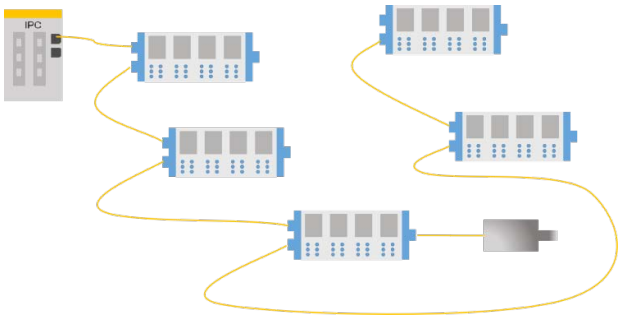


图4 树形拓扑示意图

网状拓扑：网络中的节点通过多条路径相互连接。网状拓扑的优点是具有极高的冗余性和可靠性，缺点是成本较高，布线复杂，适用于对网络稳定性和可靠性要求极高的大型系统。

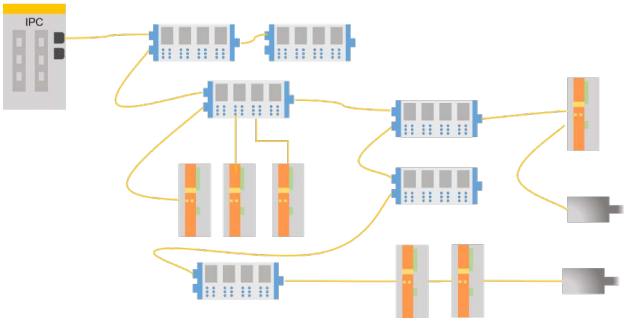


图5 网状拓扑示意图

混合拓扑：结合了两种或以上的拓扑结构，提供设计上的灵活性，优化网络性能和可靠性。适用于需要综合考虑性能、成本和可靠性的复杂应用场景。

EtherCAT在拓扑网络中的数据传输机制与关键技术

1. On the Fly数据传输机制

数据帧在通过网络传输的过程中，能够被各个从站节点实现读取和插入数据，而不需要等待整个数据帧的接收和处理完成。这种方式极大地提高了数据传输的效率和速度，因为它允许数据中即时处理，从而减少了等待时间，并提高了整体的通信性能。

2. 热连接技术(Hot Connect)

热连接技术允许 EtherCAT 网络中的从站在不中断网络通信的情况下，动态地加入或离开网络。这种技术特别适用于需要高灵活性和可扩展性的工业自动化系统。例如，在某些应用场景中，用户可能需要在不停止整个生产线的情况下更换或添加设备，热连接技术就能够实现这一点。

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。这些 EtherCAT 主站控制器和 PCIe EtherCAT 通讯卡具有丰富的接口资源，可灵活适配不同的网络拓扑结构。配套的 AWStudio 软件具有网络硬件设备接入管理，PDO 配置，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。

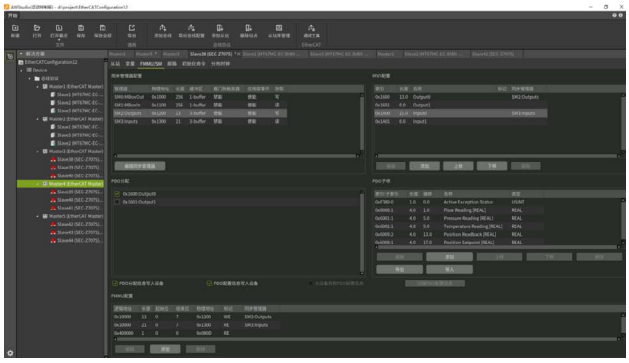


图6 AWStudio PDO配置界面

型号	内核	主频	内存	存储	以太网	无线	CAN	RS485	M.2	DI/DO	AI/AO	HDMI
ZMC300E	单核Cortex®-A8	800MHz	256MB	256MB	1*EtherCAT+2*NET	--	1路	1路	--	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	1GB	4GB	1*EtherCAT+3*NET	--	2路	2路	--	16/16	2/2	--
ZMC602E	Cortex®-A53*2+R5F*4	1GHz	1GB	4GB	1*EtherCAT+5*NET	--	3路	2路	--	16/16	--	--
ZMC60E	Cortex®-A53*2+R5F*4	1GHz	1GB	4GB	1*EtherCAT+2*NET	Wi-Fi/4G	2路	2路	--	8/8		--
ZMC61E	Cortex®-A53*2+R5F*4	1GHz	1GB	4GB	1*EtherCAT+2*NET	Wi-Fi/4G	1路	2路	--	8/8	6/2	--
ZMC900E	Cortex®-A55*4+R5f*3	2GHz	4GB	8GB	2*EtherCAT+2*NET	Wi-Fi/4G	1路	1路	支持	16/16	--	支持

表1 EtherCAT主站控制器

图7 EtherCAT主站控制器系统框图

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	PCIe 全高	2 路主站，4 网口	支持	支持	--	--	--	--
PCIe-2E	PCIe 全高	1 路主站，2 网口	支持	支持	--	--	--	--
MiniPCIe-2E	MiniPCIe	1 路主站，2 网口	支持	--	--	--	--	--

表2 PCIe EtherCAT主站通讯卡

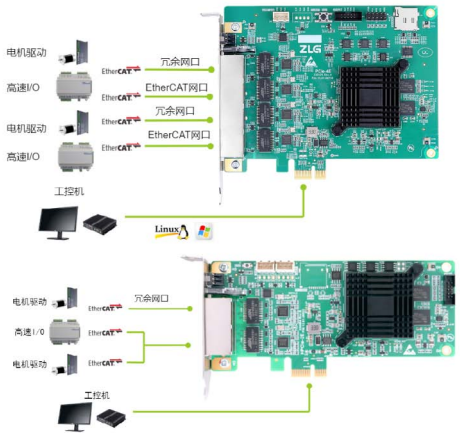


图8 PCIe EtherCAT主站通讯卡

【新品发布】
数字能源EMS管理再掀新篇章

ZLG 致远电子 2024-09-23 11:33:51

致远电子 EM 系列工商业储能网关累计装机容量突破 2GWh！聚焦数字综合能源应用，全新一代 EM-800/EM-1000G 发布，见证光储充时代的来临！

早在 2008 年，致远电子的工程师在为国内某新能源企业设计光伏通讯管理机方案时，或许并没有想到，以新能源为首的数字能源互联网进程，在当下已触手可见了。

在过 去的 16 年，致远电子陪伴中国电力电子行业走过了光伏、风电、UPS、充电桩、储能等发展周期，以自身深厚的嵌入式软硬件实力服务着每一位客户。

2022 年，致远电子在传统工业板卡、通讯模组、电源模块产品的基础上，进一步拓展了储能行业的产品线，推出了专为 EMU 和 EMS 管理设计的能源控制器——EM 系列储能网关，这一创新产品迅速赢得了市场的广泛赞誉。到了 2023 年，随着工商业储能应用的兴起，致远电子已经帮助用户累计安装了超过 2GWh 的储能设备。

今天，面向越发复杂的光储充数字能源管理场景，致远电子在 EM 系列储能网关基础上，针对综合能源管理 EMU 应用，推出全新的 EM-1000G 及 EM-800 系列新品，继续陪伴用户迎接新时代的挑战。

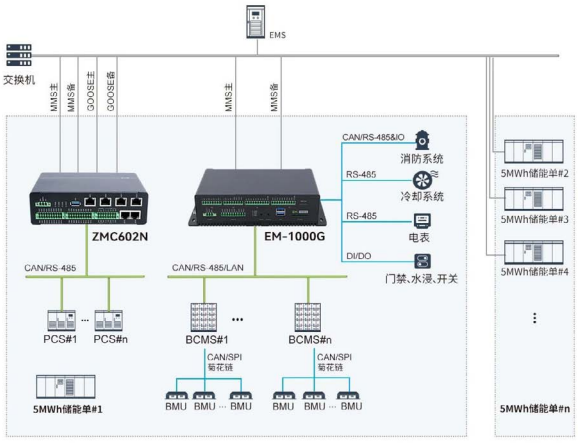
EM-1000G：大型电站EMU国产化主控+边缘算力新方案

目前，大型储能电站应用领域主要在源网侧，作为电力基础设施对系统的安全稳定性要求非常高。而随着技术的不断进步和成熟，为了实现降本增效和信息安全可控，行业内对于推动储能关键部件国产化的呼声愈发高涨。

EM-1000G

凭借在储能 EMU 产品设计领域的多年经验，致远电子隆重推出了 EM-1000 的升级版——EM-1000G。这款新品采用了先进的国产嵌入式主控平台，不仅保留了 EM-1000 原有的丰富外设接口，还在 CPU 主频上实现了 45% 的性能提升，内存和存储配置也升级至 4G+32G，从而更有效地应对源网侧储能单元 EMU 在大数据运算方面的挑战。

此外，EM-1000G 还内置了 NPU 算力，这使得它能够满足创新性储能边缘计算应用的需求，进一步推动储能技术的发展。



典型大型储能电站储能EMU应用框图

EM-800：分布式综合能源EMS新方案

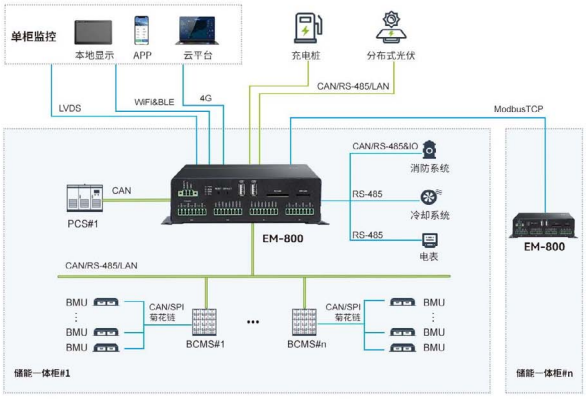
经历完 2023 年这个工商业储能“元年”，工商业储能以令人惊讶的速度进入了白热化竞争时代。除了激烈的价格竞争，将储能一体柜融合光伏、

边缘计算▼

充电桩的综合能源项目，已成为当下厂家展示技术创新能力的全新战场。相较于标准化的储能一体柜，光储充项目通常需要更高的定制化程度，涉及逆变器、MPPT 模块、充电桩等多种外设的集成管理，这就需要更强大的 EMS 硬件作为支撑。

为了满足未来光储充应用的高标准需求，致远电子推出了 EM-500 的升级版——EM-800。在工商业储能 EMS 的成本预算范围内，EM-800 采用了全新的双核 Cortex-A55 处理器，并升级了内存和存储，以更强大的性能迎接更复杂、更高要求的光储充应用挑战。

在硬件方面，EM-800 增加了内置 4G、WiFi 和蓝牙功能，并支持更多的 RS-485 接口，显著提升了产品的适用性和灵活性。这使得用户能够轻松地用一套方案来应对多样化的光储充应用场景。



光储充系统应用示例框图

矩阵式产品系列设计

除了 EM 系列网关产品，致远电子还精心打造了一套家族式产品矩阵，全面覆盖从储能到综合能源的全场景应用需求。这些产品之间的业务软件移植过程快速且便捷，协助用户快速响应瞬息万变的市场动态需求。

【CAN总线知识】如何接好CAN的“地”

ZLG 致远电子 2024-09-12 11:38:32

工业现场 CAN 环境复杂多变，工程师面对信号的杂、乱、差却是束手无策，追根溯源对于信号的各种地你接对了吗？

CAN 总线以其高可靠性、实时性、灵活性以及严谨的数据处理机制等特点，在工业现场和汽车行业得到广泛应用，但随着环境干扰以及节点数目的增加等对CAN 总线的稳定性提出更高的要求，而面对电源地、信号地、屏蔽地、外壳地不同的接地方式又该如何处理呢？

如图 1 分别是电源地、信号地、屏蔽地以及大地四种不同地的常见符号。

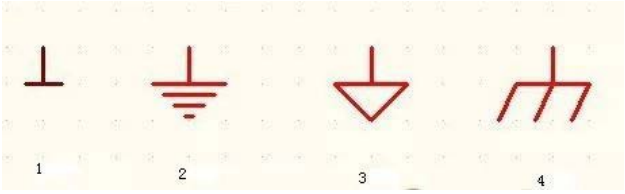


图1 四种接地符号

电源地（GND）

1. 电源地概念

电源地也为供电地，是为保证供电电源形成完整的电流回路设置的供电地，即 GND。

2. 电源地处理

与单电源供电的负极相连。

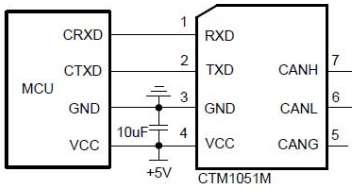


图2 CAN收发器电源地（GND）接线

信号地（CAN-GND）

1. 信号地概念

信号地也称为隔离地，为使电子设备工作时有一个统一的参考电位，避免有害电磁场的干扰，使设备稳定可靠的工作，设备中的信号电路统一参考地，即 CAN-GND。

2. 信号地处理

许多实际应用中，设计者常直接将每个节点的参考地接于本地的大地，作为信号的返回地，看似正常可靠的做法，却存在极大的隐患！

信号地（CAN-GND）正确的接法主要分为两种：

单屏蔽层线缆：如果线缆是单屏蔽层，信号地理想接法是使用专门的信号线将所有节点信号地连接，起到参考地的作用。但如果缺少信号地线，亦可将所有节点信号地都连接到屏蔽层，但这样屏蔽效果亦差强人意。

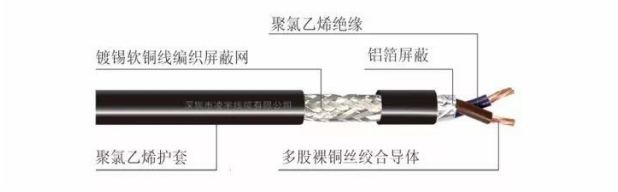


图3 带有屏蔽层双绞线

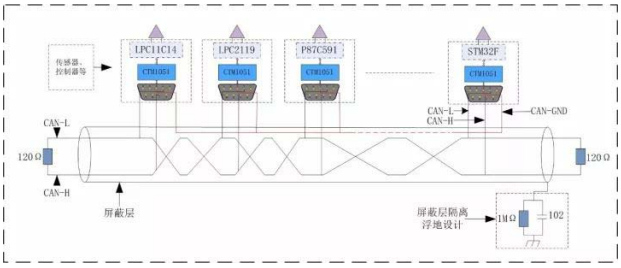


图4 含信号地线双绞线连接方式

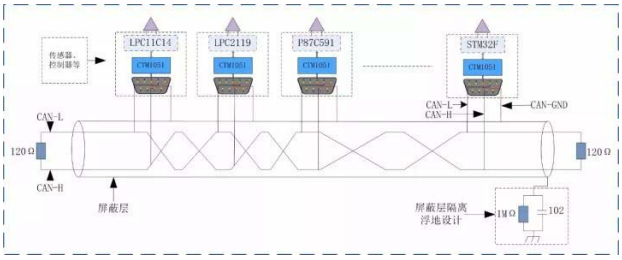


图5 信号地与屏蔽层连接方式

双屏蔽层线缆：当使用双层屏蔽电缆时，需要将所有节点信号地连接到内屏蔽层，若使用非屏蔽线进行数据传输时，请保持信号地管脚悬空处理。

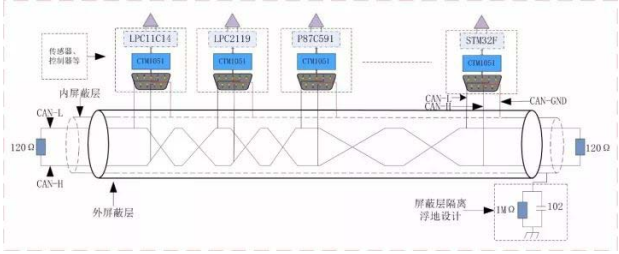


图6 双屏蔽层信号地处理方式

所有节点信号地接到屏蔽层或者双屏蔽层的内层后，屏蔽层处理方式注意为单点接地，不可多点接地，否则会在信号地线上形成地环流。

另外，单点接地时为了加大供电地和信号地之间的隔离电阻，阻止共地阻抗电路耦合产生的电磁干扰，注意采用隔离浮地设计，通过阻容方式将屏蔽层与外壳隔离。

工作环境温度 -40°C ~+85°C。

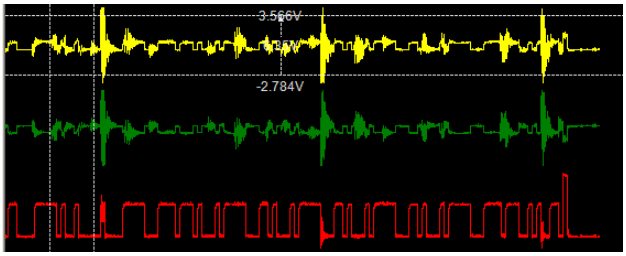


图7 未进行单点接地处理的报文受到电磁干扰

屏蔽地（CAN-Shield）

1. 屏蔽地概念

屏蔽地（CAN-Shield）也可理解为 CAN 屏蔽层，部分场合也标为 FG。导体外部有导体包裹的导线叫屏蔽线，包裹的导体叫屏蔽层，一般为编织铜网或铜泊（铝），屏蔽层需要接地处理，保证外来的干扰信号可被该层导入大地。

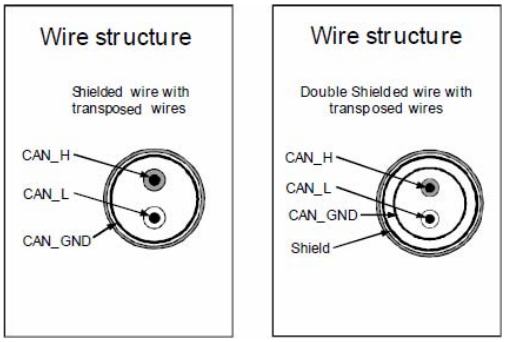


图8 单屏蔽层和双屏蔽层电缆剖析

2. 屏蔽地处理

当使用双层屏蔽电缆时，CAN-Shield 连接到外屏蔽层和 DB9 连接器的屏蔽壳。并且，使用 DB9 针式连接器时外屏蔽层会被连接到 pin 5 以保证当使用没有屏蔽连接的连接器时，可靠的接地。

多节点总线同样要求屏蔽地采用单点接地，防止形成回路，并且为浮地设计。

如下图 9 所示处理方式，CTM1051 模块 3 脚为屏蔽地，5 脚为信号地。

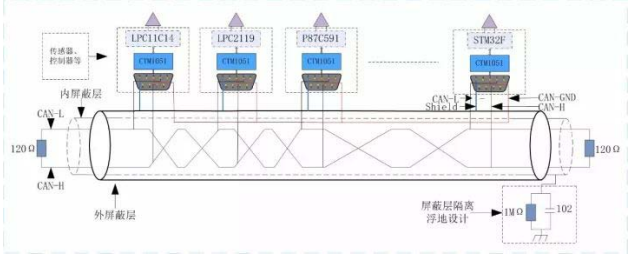


图9 双屏蔽层线中信号地、屏蔽地处理方式

外壳地

1. 外壳地概念

静电的电荷集聚在物体的表面，一旦遇到可以释放的回路就可以形成电流。有时候产生的电压非常高，特别是在干燥的环境里。电子产品的外壳地就是用来快速地将电荷释放到大地。

2. 外壳地处理

外壳接地既是对人体安全的保护，也是防干扰的一种手段，因为一般情况下机壳是金属的，是非常好的屏蔽体，绝大部分辐射干扰都可以阻挡在机壳之外。通过地线引入的干扰（也叫共阻抗干扰），处理方法一般采用地线隔离技术，在外壳接地时接入阻抗，加入滤波等。

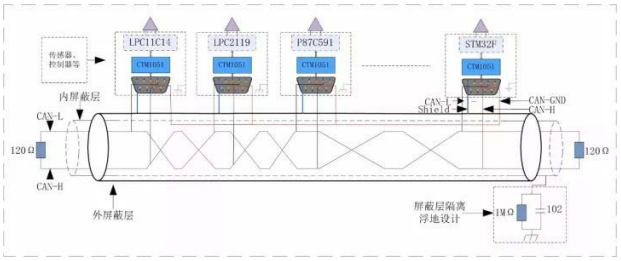


图10 信号地、屏蔽地、外壳接地连接推荐电路

3. 改进方案建议

如果您在使用 CAN 总线进行调试时，遇到过偶尔通信出错，或者接收不到数据，再者一直正常使用的总线，突然出现大范围的错误，或者节点损坏。

如果您还是在使用单纯的CAN收发器，那么请换成隔离CAN收发器吧！ZLG 致远电子的 CTM 隔离模块内部包含隔离 DC-DC、信号隔离电路、CAN 总线收发电路、基础的总线防护等。

隔离收发器可将总线和控制电路进行电气隔离，将高压阻挡在控制系统之外，可以有效地保证操作人员的人身及系统安全。不仅如此，隔离可以抑制由接地电势差、接地环路引起的各种共模干扰，保证总线在严重干扰和其它系统级噪声存在的情况下不间断、无差错运行。

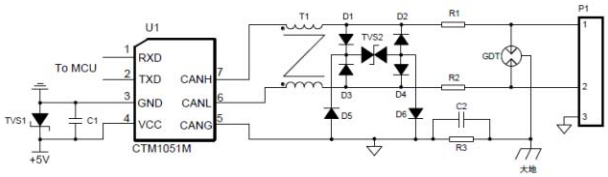


图11 CAN隔离收发器推荐设计电路

那么，您的这些地处理好了吗？

【新品发布】多通道车载以太网分析仪震撼首发！

ZLG 致远电子 2024-09-23 11:33:51

选型表

型号	VBNET-4302
车载以太网路数	4 路
车载以太网接口形式	专用车载以太网高速连接器（兼容 TE 的 MATEnet 车载连接器）
标准以太网路数	3
标准以太网速率	100M/1000M 自适应
标准以太网接口形式	RJ45
CAN FD 路数	2 路
CAN FD 接口形式	标准 DB9 接口
Master/Slave 模式	拨码开关
电气隔离	√
工业级	√
自适应速率协商	√
数据转发	支持 4 路 100/1000BASE-T1 端口任意组合数据转发，以及 2 路 RJ45 端口之间报文路由转发和监听
主从模式	支持车载以太网自适应或手动配置主从模式
速率模式	支持车载以太网自适应或手动配置连接速度
MAC 过滤	支持
MAC 配置	支持动态、静态配置 MAC 地址转发列表
映射	支持 IP/MAC 地址映射
唤醒	支持 TC10 睡眠唤醒
精度	微秒级时延动态监测 100/1000BASE-T1 节点间的全双工通信数据
LED 显示	支持 显示 100/1000BASE-T1 的 Master/Slave 模式、链接状态和数据运行状态等
供电	9-48V 宽电压供电
整机功耗	<7W
工作温度	-40℃ ~+85℃

车载以太网全场景测试解决方案

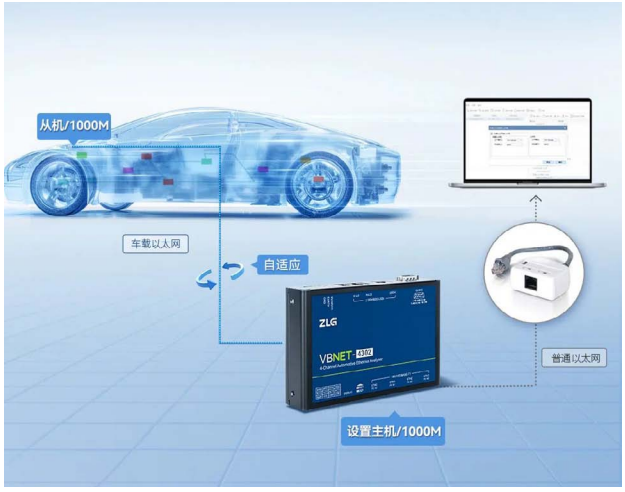
随着智能网联汽车的快速发展，车载以太网仿真测试需求日益强烈。ZLG 致远电子推出多通道车载以太网和 CAN FD 仿真测试工具，可进行车载以太网数据监控和解析、通信测试、ECU 刷写、产线升级等全场景的仿真分析测试。

多个ECU之间的数据交互和报文监听，应用于ECU系统测试。

互联互通▼

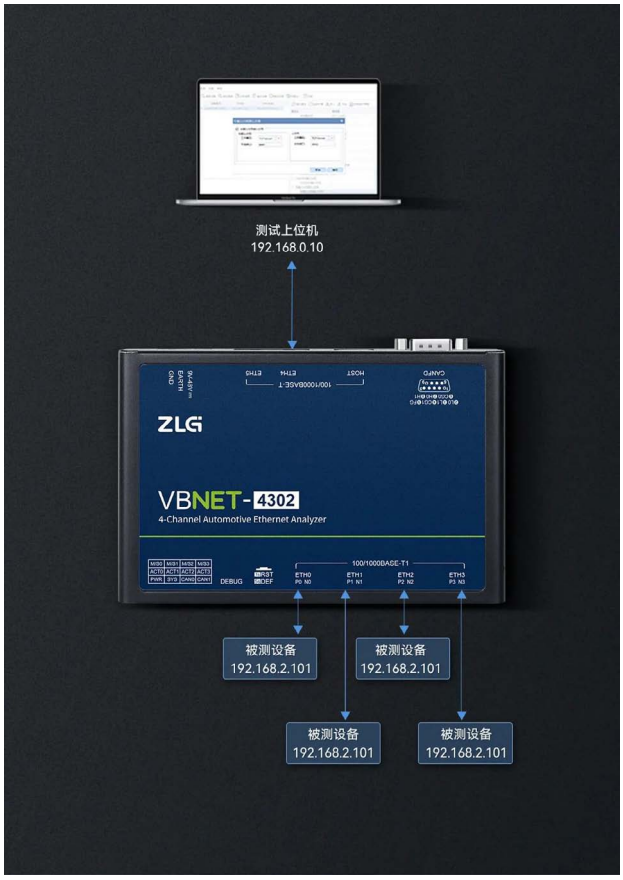
主从/速率自适应:测试刷写更轻松

VBNET 系列产品可实现主从模式和速率自适应大大降低操作时间和试错成本，让测试与刷写更高效简单。



支持多台ECU刷写，无需修改IP和MAC

VBNET-4302 首创支持多台 ECU 刷写，可在产线一对多测试，实现多个相同 ECU 刷写升级测试，彻底解决网络冲突问题。



支持VLAN网络隔离，精准保证数据安全

VBNET-4302 产品支持 VLAN 应用，更精准的网络划分，实现更高效的分析测试。

支持TC10休眠唤醒测试，真实模拟整车工况

VBNET-4302 支持 TC10 睡眠唤醒模式，允许设备在不进行数据传输时进入低功耗状态，以及当网络中的某个设备需要通信时，可以进行特定唤醒，从而实现 ECU 休眠唤醒功能的测试。

两路CAN FD并行通讯,精确时钟同步网络

VBNET-4302 同时集成 2 路标准 CAN FD 接口可高效进行 CAN FD 通讯，且实现车载以太网数据和 CAN FD 数据的时间同步，满足多总线应用需求。

全状态LED显示，现场维保更轻松

VBNET 系列产品支持 LED 状态指示，可实时显示车载以太网 Link 连接、通信速率、Master/Slave 和 USB 等状态。

核心器件全国产设计，自主可控us级监测

VBNET 系列产品采用国产 FPGA 架构平台，可支持微秒级时延动态监测 100/1000BASE-T1 节点间的全双工通信，确保数据准确性和可靠性。

互联互通▼

CAN底层报文抓到了，却不知怎么解析？以及如何看到信号运行状态？

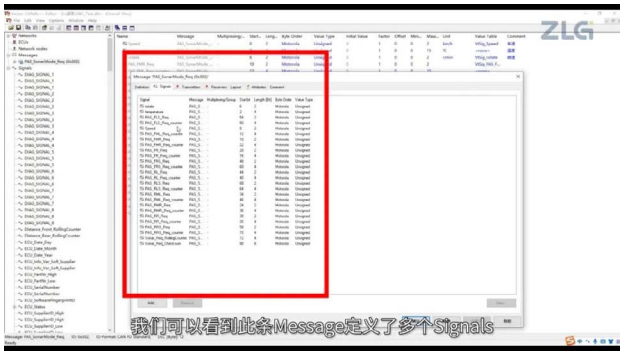
ZLG 致远电子 2024-09-29 11:42:21

本文介绍 CAN 总线中 DBC 文件的重要性及 ZCANPRO 如何实现 DBC 解析、数据发送和实时曲线分析，帮助您更有效地分析和利用 CAN 总线数据。

在 CAN 总线的实际应用中，我们经常需要解析底层的报文，以获取其实际的物理值并分析总线问题。ZCANPRO 工具的 DBC 解析和实时曲线分析功能都是基于 DBC 文件的解析结果。在演示这些功能之前，我们首先需要了解 DBC 文件到底是什么？

DBC文件简介

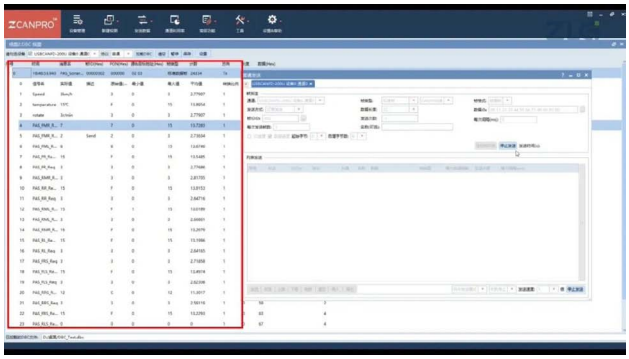
DBC 文件是一种定义 CAN 总线通信的数据格式文件，简单来说，包含了 Message（帧 ID）和 Signals（帧 data）的定义。一条 Message 可以定义多个 Signals，并且可以为这些 Signals 添加中文注释和单位。



DBC解析演示

1. 打开ZCANPRO并进行硬件连接：

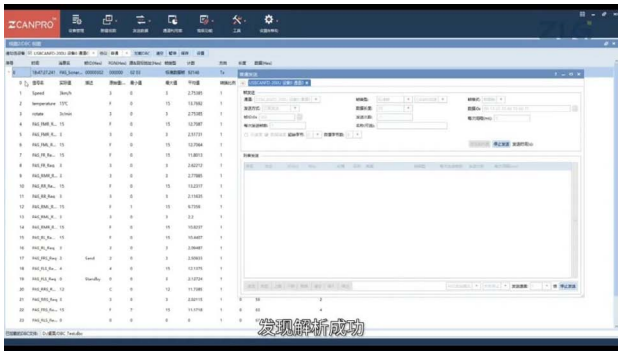
使用 USBCANFD-200U 设备进行收发模拟和解析。需要注意的是，DBC 解析功能仅解析 DBC 文件中定义的数据。例如，如果 DBC 文件定义了一条 ID 为 302，DLC（数据长度码）为 12 的 CANFD 标准帧，ZCANPRO 能够顺利解析出这些数据。



2. 解析不同DLC值的效果：

- 当 DLC 设置为 8 时，由于数据长度小于 DBC 定义的长度，解析失败。
- 当 DLC 设置为 16 时，数据长度大于等于 DBC 定义的长度，解析成功。

这说明，只有当帧数据长度大于等于 DBC 文件定义的 Message 的 DLC 时，才能正常解析。



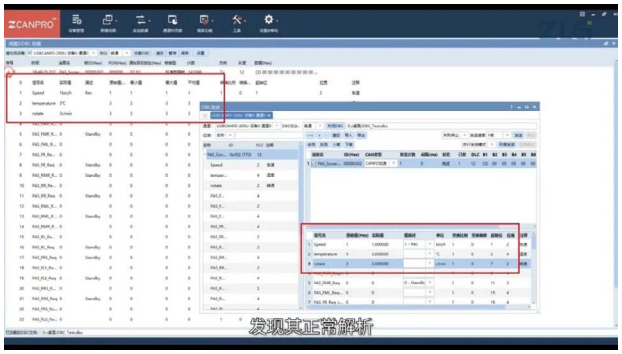
3. 帧类型的影响：

如果更改帧类型，解析同样会失败，这强调了 DBC 解析必须完全匹配 DBC 定义的数据才能成功。

DBC发送演示

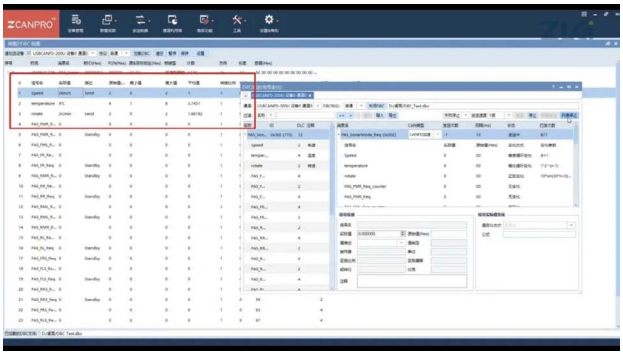
1. DBC发送：

加载 DBC 文件，可以更改实际值进行发送。如果选择 CAN 类型发送而 DBC 文件定义为 CANFD 类型，发送将失败。只有选择正确的 CANFD 加速类型，发送才能成功。



2. DBC变化发送：

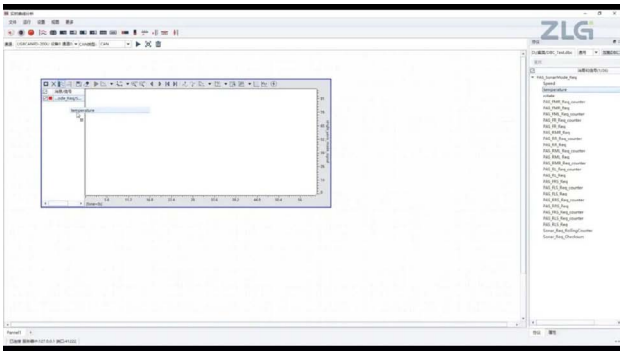
用户可以根据需要更改信号的实际值。选择 CANFD 加速类型并点击发送，可以观察到信号值的周期性变化，满足不同工况的需求。



曲线分析演示

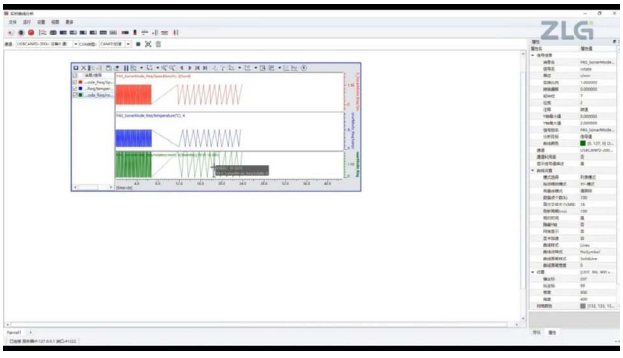
1. 加载DBC并进行曲线分析：

加载 DBC 文件，将信号值拖入分析区域。需要注意的是，曲线分析仅针对接收方向进行解析，且帧类型必须与接收帧的类型一致。



2. 观察波形变化：

通过选择 USBCANFD-200U 的一通道进行 CANFD 加速（0 通道发送，1 通道接收），点击发送后可以看到波形线的变化。通过滚动鼠标，可以改变曲线的横纵坐标。调整周期，曲线的幅值也会相应变化。



全新ZXDoc

为满足汽车电子用户需求，我们全新升级了国产汽车总线工具链软件ZXDoc，支持 XCP/CCP 标定、SOME/IP、DoIP 诊断、仿真等多项应用层功能，欢迎咨询我们了解！

关注我们！我们将持续推出更多相关内容，分享新功能的详细解读和使用指南。敬请期待！

互联互通▼

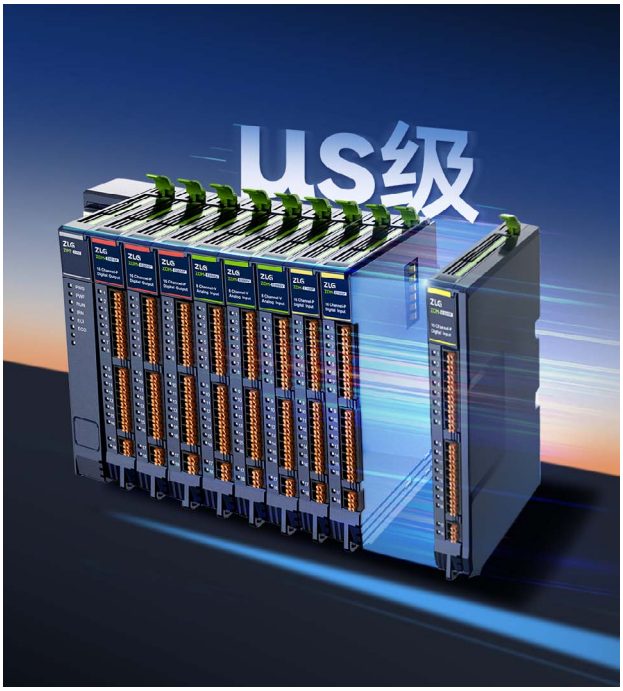
【新品发布】ZLG致远电子E系列I/O模块正式发布，让高速控制触手可得

ZLG 致远电子 2024-09-09 11:41:50

丰富的模块类型可灵活拓展

稳定高速的背板总线技术

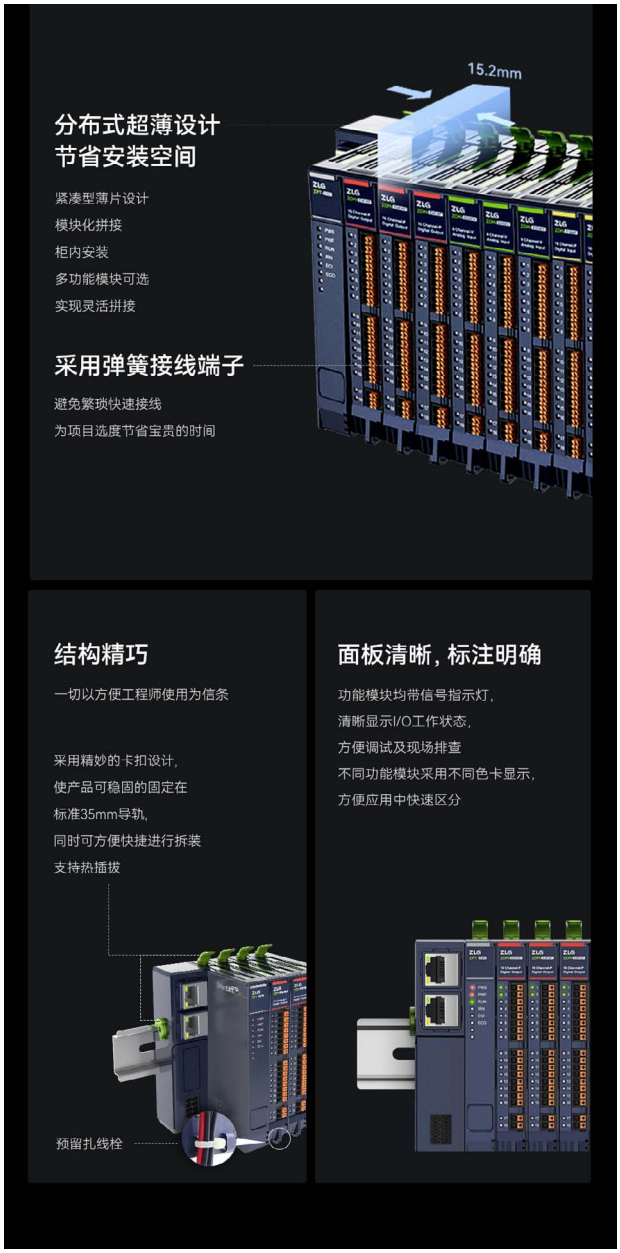
刷新周期可达 us 级。



规格参数

模块功能	型号	规格
耦合器	ZPT-8080	高性能版，EtherCAT 总线耦合器套件，μs 级模块刷新周期（含电源，尾板）
数字量输入模块	ZDM-E1600P	高性能版，16 通道数字量输入，PNP 型，直插端子
	ZDM-E1600	高性能版，16 通道数字量输入，PNP/NPN 兼容，直插端子
	ZDM-E1600N	高性能版，16 通道数字量输入，NPN 型，直插端子
数字量输出模块	ZDM-E0016P	高性能版，16 通道数字量输出，PNP 型，0.5A，直插端子
	ZDM-E0016N	高性能版，16 通道数字量输出，NPN 型，0.5A，直插端子
	ZDM-E0008M	高性能版，8 通道机械继电器输出，2A，继电器隔离，直插端子
	ZDM-E0008S	高性能版，8 通道固态继电器输出，1A，继电器隔离，直插端子
模拟量输入模块	ZDM-E0800V	高性能版，8 通道模拟量电压输入，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZDM-E0800I	高性能版，8 通道模拟量电流输入，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
模拟量输出模块	ZMD-E0008V	高性能版，8 通道模拟量电压输出，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZMD-E0008I	高性能版，8 通道模拟量电流输出，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
温度采集	ZDM-E0400P3	高性能版，4 通道 RTD 温度信号采集，3 线制，PT100/PT1000，±0.5℃精度
	ZDM-E0800TC	高性能版，8 通道 16 位热电偶温度信号采集，K 型 /T 型，±0.5℃精度

产品细节



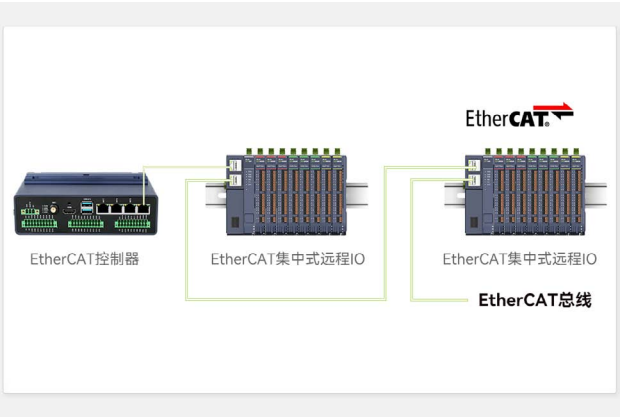
互联互通 ▼

工业级品质，通讯稳定

完全自主设计，研发到生产层层把关，稳定可靠，底部金属接地，抗干扰强。

应用行业

系统架构



其他自动化产品推荐

ZIO系列

EtherCAT插板式I/O模组

支持紧凑型电机驱动模块
支持DI、DO、AI、AO等功能模块
适用于批量标准化生产

- 预制底板即插即用
- 用线末避免了接线错误
- 用防呆接口连接器防止插错
- 可以简化接线工作，同时极大缩短电气装配时间
- 简化工作流程，降低批量的安装生产成本

8通道热电偶温度采集器

支持8通道K型热电偶信号采集
就近布置，高效测试、稳定保障，
有效降低测试成本

- 灵活选择Wi-Fi、RS-485、CAN进行通信
- 支持TF卡存储，保障数据
- 测温精度可达0.02%±0.3℃
- 速率可达100SPS

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

【应用方案】
E系列I/O模块在锂电装备制造系统的应用

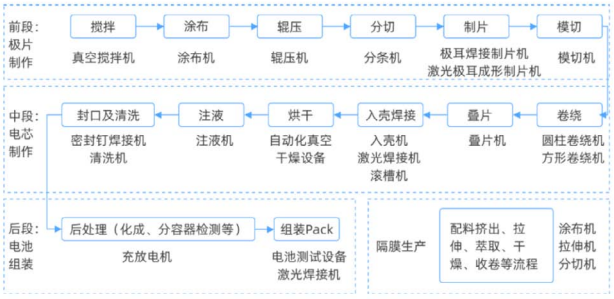
ZLG 致远电子 2024-09-24 11:31:38

为了满足电池生产线对稳定性和生产效率的严苛要求，ZLG 致远电子推出高速 I/O 应用方案，它不仅稳定可靠，而且速度快，能够迅速响应生产需求。

锂电池的生产工艺较为复杂，大致分为三个主要阶段：极片制作、电芯制作以及检测组装。

- 极片制作：此阶段的目标是将原材料加工成极片。主要工序包括搅拌、涂布、辊压、分切、制片、模切等工序；
- 电芯制作：在这一阶段，极片被进一步加工成电池电芯。主要工序包括卷绕 / 叠片、入壳焊接、烘干、注液、封口等工序；
- 检测组装：最后阶段的目的在于激活并检测电芯并经过 Pack 封装使之成为成品。

锂电池生产工艺流程



此次我们介绍 ZLG 致远电子 I/O 在 Pack 封装线的应用方案，简单来说，该条线的作用就是通过对电芯、端板、胶垫绝缘罩、侧板进行清洗、涂胶，然后经过组装、焊接、包装、打标等工序，最终形成标准电池模组的过程。

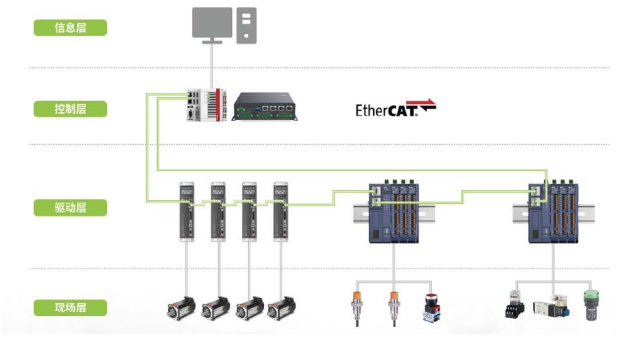
电池生产设备对稳定性以及生产效率提出几乎苛刻的要求，为了挑战更高效的生产节奏，整个系统的实时性能便格外重要。

为此，系统设计采用高速 EtherCAT 主控外，用致远电子 E 系列远程 I/O 产品进行通讯，E 系列远程 I/O 采用稳定高速的背板总线技术，刷新周期短至 μs 级，作为一个高速的桥梁，从现场到控制层的交互中实现完美提速。

系统主要使用 DI、DO、AI、AO 功能模块：

- 数字量输入模块（DI）：主要用于光电传感器、气缸磁性开关、门禁感应、各种启停信号、报警输入等信号的接入采集。
- 数字量输出模块（DO）：主要用于气缸、激光器、机器人等开关信号输出。
- 模拟量输入模块（AI）：主要用于压力检测、风速检测、气压检测等信号的采集。
- 模拟量输出模块（AO）：主要用于比例阀的控制。

系统框图



致远电子E系列远程I/O产品介绍

致远电子I/O模块优势特点：

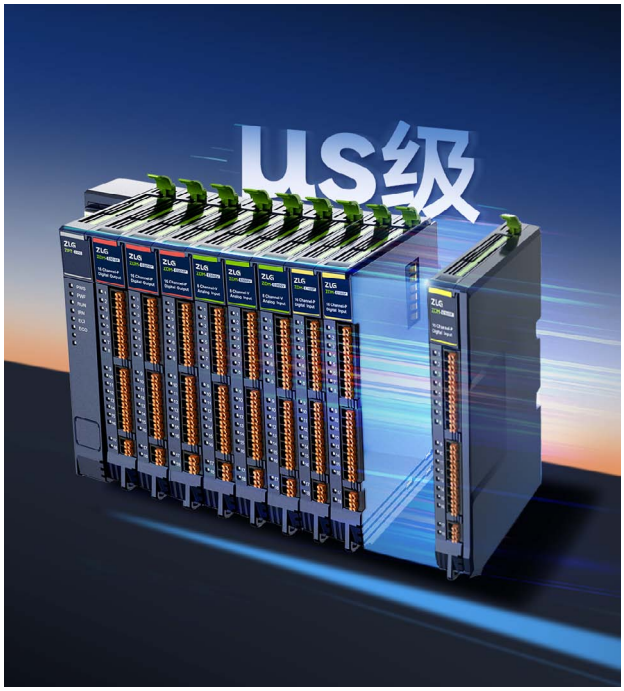
- EtherCAT 耦合 + 稳定高速的背板总线技术，实现 μs 级刷新周期；
- 功能模块均带信号指示灯，清晰显示 I/O 工作状态，方便调试查看；
- 模拟量分辨率 16 位，可选电压型和电流型，且提供多种量程；
- 测温模块有 PT100/PT1000 热电阻和 K/T 型热电偶两款可选，测温精度 ±0.5℃；
- 完全自主设计，研发到生产层层把关，工业级品质，通讯稳定；
- 拆装接线方便快捷。

互联互通 ▼

丰富的模块类型可灵活拓展

稳定高速的背板总线技术

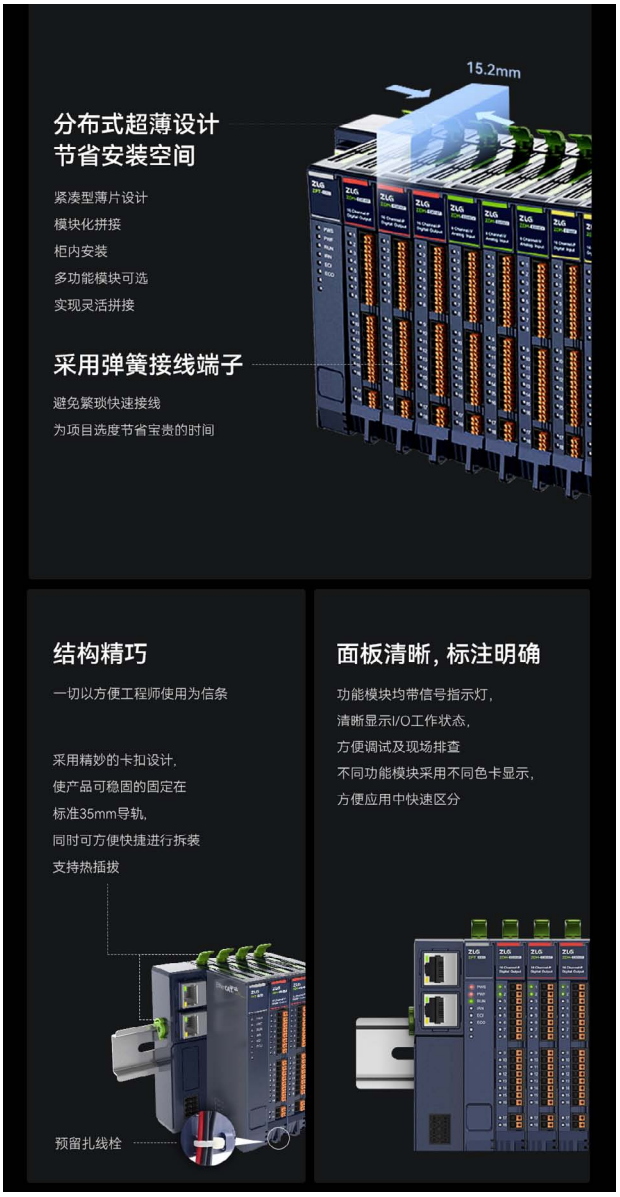
刷新周期可达 us 级。



工业级品质，通讯稳定

完全自主设计，研发到生产层层把关，稳定可靠，底部金属接地，抗干扰强。

产品细节



规格参数

模块功能	型号	规格
耦合器	ZPT-8080	高性能版，EtherCAT 总线耦合器套件，μs 级模块刷新周期（含电源，尾板）
数字量输入模块	ZDM-E1600P	高性能版，16 通道数字量输入，PNP 型，直插端子
	ZDM-E1600	高性能版，16 通道数字量输入，PNP/NPN 兼容，直插端子
	ZDM-E1600N	高性能版，16 通道数字量输入，NPN 型，直插端子
数字量输出模块	ZDM-E0016P	高性能版，16 通道数字量输出，PNP 型，0.5A，直插端子
	ZDM-E0016N	高性能版，16 通道数字量输出，NPN 型，0.5A，直插端子
	ZDM-E0008M	高性能版，8 通道机械继电器输出，2A，继电器隔离，直插端子
	ZDM-E0008S	高性能版，8 通道固态继电器输出，1A，继电器隔离，直插端子
模拟量输入模块	ZDM-E0800V	高性能版，8 通道模拟量电压输入，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZDM-E0800I	高性能版，8 通道模拟量电流输入，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
模拟量输出模块	ZMD-E0008V	高性能版，8 通道模拟量电压输出，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZMD-E0008I	高性能版，8 通道模拟量电流输出，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
温度采集	ZDM-E0400P3	高性能版，4 通道 RTD 温度信号采集，3 线制，PT100/PT1000，±0.5℃精度
	ZDM-E0800TC	高性能版，8 通道 16 位热电偶温度信号采集，K 型 /T 型，±0.5℃精度

其他自动化产品推荐

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

互联互通▼

【应用方案】E系列I/O模块在光伏制绒设备的应用

ZLG 致远电子 2024-09-25 11:36:36

光伏设备产线较长，各分布点若采用 PLC+IO 扩展的方式将会大大增加系统成本及开发难度，方案推荐采用 E 系列远程 I/O 模块，仅需和远程 PLC 通过总线 / 工业以太网连接，就可轻松实现分布式 I/O 控制系统。

光伏产业和风电一起是国家鼓励发展的新能源行业，作为新兴可再生能源，有效缓解了能源紧缺问题，同时达到节能减排目的。

随着自动化技术在国内的推广及广泛应用，通过自动产线制造太阳能电池片可以大大提升生产效率，对光伏行业的蓬勃发展起到了积极的作用。

制绒是光伏电池片制造的关键工序，主要功能是去除机械损伤层，增加电池片表面面积，降低电池片表面反射率，清理表面油污及杂质。

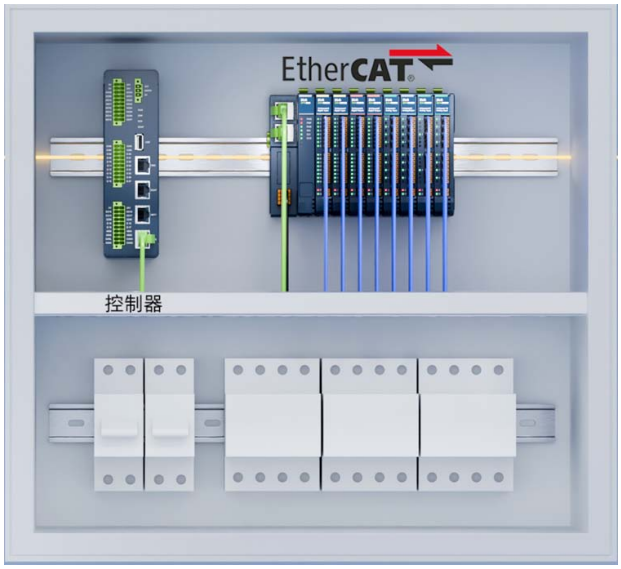
本方案将重点介绍 ZLG 致远电子 E 系列高速 EtherCAT 协议集中式 I/O 模块在光伏制绒机上下料设备上的应用。

光伏制绒上下料机设备工艺描述

人工在装料工位将原材料放入指定位置→由 AGV 小车自动取料并运送到制绒上料机→上料机将物料输送至制绒机进行一系列表面处理→作业完毕后物料通过制绒下料机下料→由 AGV 小车运料到下料指定位置。

上下料机主体由传送带、吸盘、花篮等机构组成，有大量的光电开关、接近开关、气缸等信号，特点是信号数量多、传感器执行器分布位置分散，所以采用致远 E 系列 I/O 模块来处理解决。

系统采用 EtherCAT PLC 作为主控，配置 ZPT-8080 与之通讯，设备运转过程中所有的光电传感、接近开关等检测信号由 ZDM-E1600N 模块进行采集，气缸等执行器开关信号由 ZDM-E1600N 传输，伺服系统直接由 PLC 控制，最终实现通过传送带运转传输物料。



致远电子E系列远程I/O产品介绍

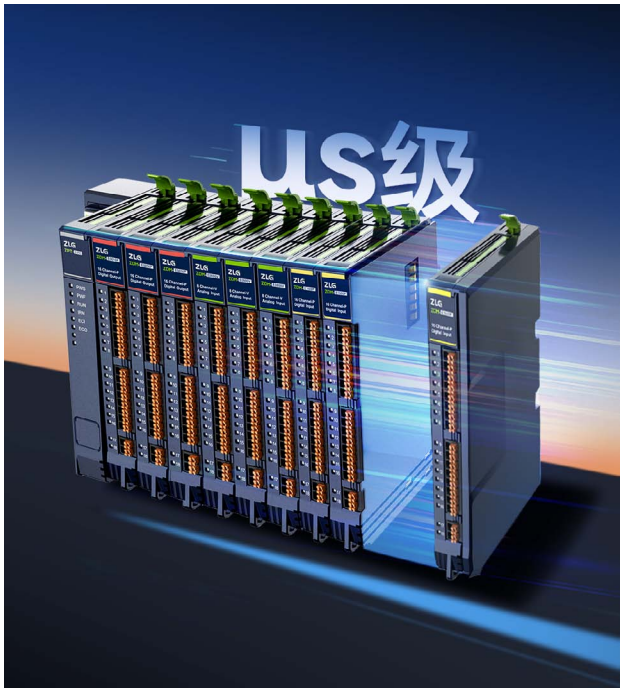
致远电子I/O模块优势特点：

- EtherCAT 耦合 + 稳定高速的背板总线技术，实现 μs 级刷新周期；
- 功能模块均带信号指示灯，清晰显示 I/O 工作状态，方便调试查看；
- 模拟量分辨率 16 位，可选电压型和电流型，且提供多种量程；
- 测温模块有 PT100/PT1000 热电阻和 K/T 型热电偶两款可选，测温精度 ±0.5℃；
- 完全自主设计，研发到生产层层把关，工业级品质，通讯稳定；
- 拆装接线方便快捷。

丰富的模块类型可灵活拓展

稳定高速的背板总线技术

刷新周期可达 us 级。

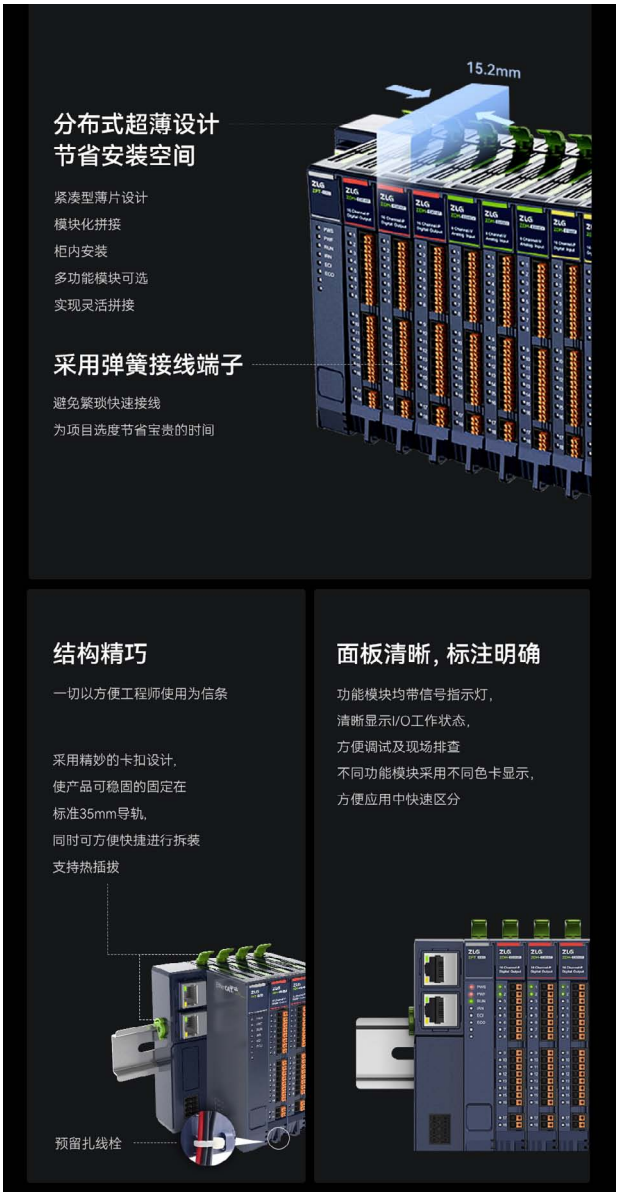


工业级品质，通讯稳定

完全自主设计，研发到生产层层把关，稳定可靠，底部金属接地，抗干扰强。

互联互通▼

产品细节



规格参数

模块功能	型号	规格
耦合器	ZPT-8080	高性能版，EtherCAT 总线耦合器套件，μs 级模块刷新周期（含电源，尾板）
数字量输入模块	ZDM-E1600P	高性能版，16 通道数字量输入，PNP 型，直插端子
	ZDM-E1600	高性能版，16 通道数字量输入，PNP/NPN 兼容，直插端子
	ZDM-E1600N	高性能版，16 通道数字量输入，NPN 型，直插端子
数字量输出模块	ZDM-E0016P	高性能版，16 通道数字量输出，PNP 型，0.5A，直插端子
	ZDM-E0016N	高性能版，16 通道数字量输出，NPN 型，0.5A，直插端子
	ZDM-E0008M	高性能版，8 通道机械继电器输出，2A，继电器隔离，直插端子
	ZDM-E0008S	高性能版，8 通道固态继电器输出，1A，继电器隔离，直插端子
模拟量输入模块	ZDM-E0800V	高性能版，8 通道模拟量电压输入，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZDM-E0800I	高性能版，8 通道模拟量电流输入，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
模拟量输出模块	ZMD-E0008V	高性能版，8 通道模拟量电压输出，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZMD-E0008I	高性能版，8 通道模拟量电流输出，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
温度采集	ZDM-E0400P3	高性能版，4 通道 RTD 温度信号采集，3 线制，PT100/PT1000，±0.5℃精度
	ZDM-E0800TC	高性能版，8 通道 16 位热电偶温度信号采集，K 型 /T 型，±0.5℃精度

其他自动化产品推荐

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。

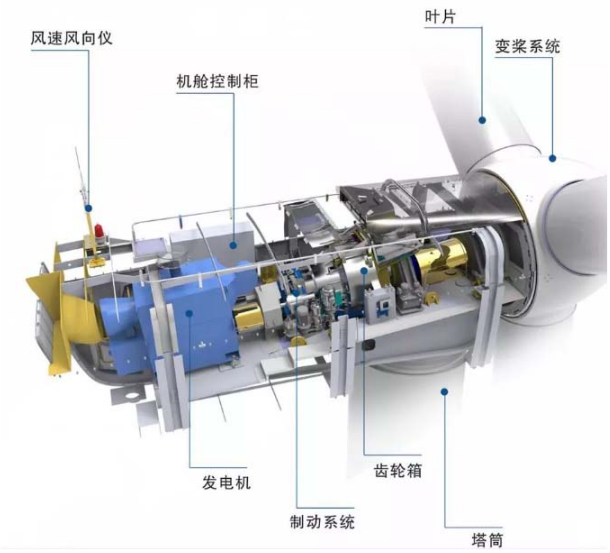
点击申请

【应用方案】
E系列IO模块在风机控制系统的应用

ZLG 致远电子 2024-09-26 11:39:35

风力发电机将风能转化为电能，面临恶劣环境和高可靠性要求的挑战，电气系统稳定性及其重要，致远电子 E 系列远程 I/O 模块，旨在助力风力发电机电控系统的稳定性和维护便捷性，为您可靠的 IO 系统保驾护航。

风力发电机是将风能转换为机械功的动力机械，其原理基于风推动叶片旋转，再通过传动系统增速，达到发电机的转速后驱动发电机发电，实现风能到电能转化的目的。



风力发电机电控系统主要由由机舱控制柜、变桨控制柜、塔基控制柜、变流器机柜等控制机柜组成，作为作为整个系统的神经中枢，电气控制系统元件器、设备等面临诸多挑战：

1.环境恶劣：

风机通常位于环境条件异常恶劣的地方，如海上盐雾、高湿环境，隔壁风沙大、高低温差大等应用环境，这对设备耐久度和稳定性提出了极高要求。

2.维护困难：

高空设备，作业困难，维护成本高。

3.系统可靠性要求高：

风电机组需要长时间不间断运行，对控制系统可靠性以及稳定性有极高的要求，任何停机可能导致巨额经济损失。

针对以上挑战，致远电子推出 E 系列远程 I/O 产品，系统所需 I/O 模块型号：

- ZPT-8080 EtherCAT 耦合器（用于和 EtherCAT PLC 系统通讯）；
  ZDM-E1600 数字量输入信号（用于叶片等位置信号读取，各系统开关状态读取、限位开关读取等）；
- ZDM-E0016 数字量输出信号（用于电机保护、系统抱闸、风扇启停控制、加热器开合控制等）；
- ZDM-E0800I 模拟量输入信号（用于采集系统传感器信号）；
- ZDM-E0400P3 PT100/PT1000 温度信号采集（用于监测系统中各组件温度）；
- ZDM-E0008I 模拟量信号输出（用于调节转速等控制需求）。

致远电子E系列远程I/O产品介绍

- EtherCAT 耦合 + 稳定高速的背板总线技术，实现 μs 级刷新周期；
- 功能模块均带信号指示灯，清晰显示 I/O 工作状态，方便调试查看；
- 模拟量分辨率 16 位，可选电压型和电流型，且提供多种量程；
- 测温模块有 PT100/PT1000 热电阻和 K/T 型热电偶两款可选，测温精度 ±0.5°C；
- 完全自主设计，研发到生产层层把关，工业级品质，通讯稳定；
- 拆装接线方便快捷。

丰富的模块类型可灵活拓展

互联互通 ▼

EtherCAT

单个耦合器最多可扩展32个功能模块

需要扩展更多，请选择多个耦合器

可根据项目需求灵活搭配选择功能模块

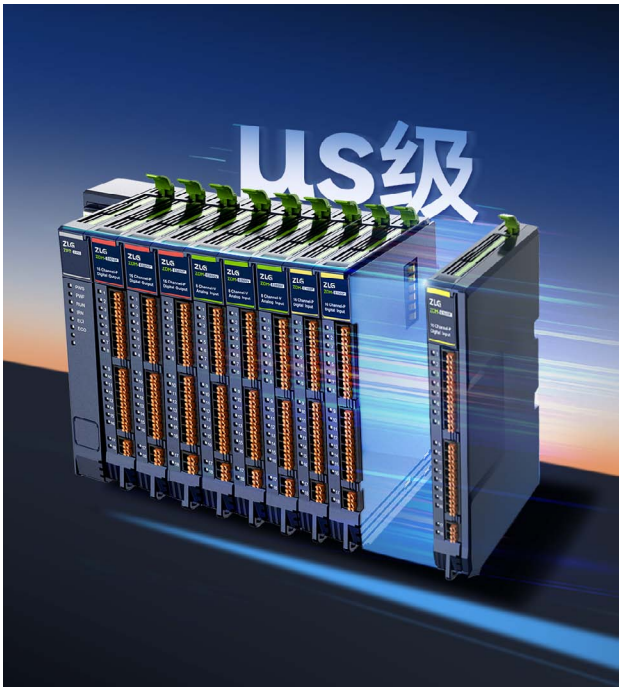
- DI、DO模块，NPN、PNP可选，DO继电器输出可选
- 模拟量分辨率16位，可选电压型和电流型，且提供多种量程
- 测温模块有PT100/PT1000热电阻和K/T型热电偶两款可选，测温精度±0.5℃

工业级品质，通讯稳定

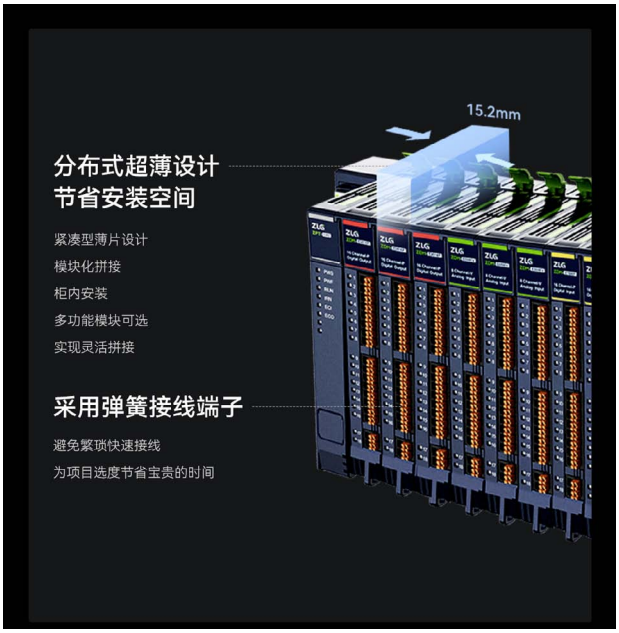
完全自主设计，研发到生产层层把关，稳定可靠，底部金属接地，抗干扰强。

稳定高速的背板总线技术

刷新周期可达 us 级。



产品细节



规格参数

模块功能	型号	规格
耦合器	ZPT-8080	高性能版，EtherCAT 总线耦合器套件，μs 级模块刷新周期（含电源，尾板）
数字量输入模块	ZDM-E1600P	高性能版，16 通道数字量输入，PNP 型，直插端子
	ZDM-E1600	高性能版，16 通道数字量输入，PNP/NPN 兼容，直插端子
	ZDM-E1600N	高性能版，16 通道数字量输入，NPN 型，直插端子
数字量输出模块	ZDM-E0016P	高性能版，16 通道数字量输出，PNP 型，0.5A，直插端子
	ZDM-E0016N	高性能版，16 通道数字量输出，NPN 型，0.5A，直插端子
	ZDM-E0008M	高性能版，8 通道机械继电器输出，2A，继电器隔离，直插端子
	ZDM-E0008S	高性能版，8 通道固态继电器输出，1A，继电器隔离，直插端子
模拟量输入模块	ZDM-E0800V	高性能版，8 通道模拟量电压输入，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZDM-E0800I	高性能版，8 通道模拟量电流输入，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
模拟量输出模块	ZMD-E0008V	高性能版，8 通道模拟量电压输出，16 位，0 ~ 5VDC/0 ~ 10VDC/±5VDC/±10VDC，±0.1% 精度
	ZMD-E0008I	高性能版，8 通道模拟量电流输出，16 位，0 ~ 20mA/4 ~ 20mA，±0.1% 精度
温度采集	ZDM-E0400P3	高性能版，4 通道 RTD 温度信号采集，3 线制，PT100/PT1000，±0.5℃精度
	ZDM-E0800TC	高性能版，8 通道 16 位热电偶温度信号采集，K 型 /T 型，±0.5℃精度

其他自动化产品推荐

【RS-485总线】RS-485网络该如何加终端电阻?

ZLG 致远电子 2024-09-13 11:37:49

RS-485 总线具有结构简单、成本低等优点，但各位工程师在组建 RS-485 总线网络时，为提升整个网络通信的可靠性，想必会经常会遇到一个问题：需不需要加终端电阻呢？本文将为你解答。

终端电阻的作用

对于 RS-485 总线，终端电阻主要是为了匹配通信线的特性阻抗，防止信号反射，提高信号质量。

在组建 RS-485 总线网络时，通常使用特性阻抗为 120Ω 的屏蔽双绞线，由于 RS-485 收发器输入阻抗一般较高（例如 RSM485ECHT 输入阻抗为 96kΩ，最多可连接 256 个节点），在信号传输到总线末端时会由于受到的瞬时阻抗发生突变（以 RSM485ECHT 为例，阻抗由 120Ω 变为 96kΩ），导致信号发生反射，影响信号的质量。RSM485ECHT 在 1200m，500kbps 通信速率的情况下不加终端电阻和加终端电阻的波形如图 1 和图 2 所示，终端电阻明显改善了信号的质量。

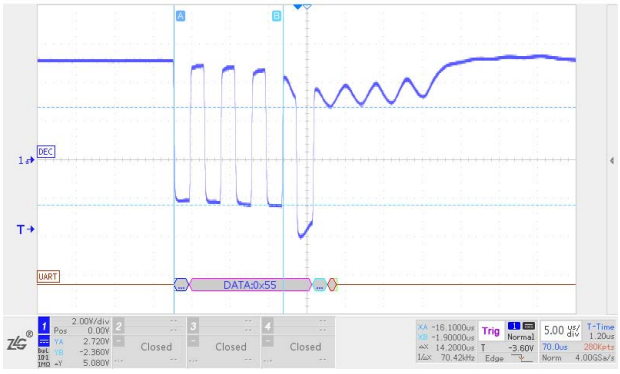
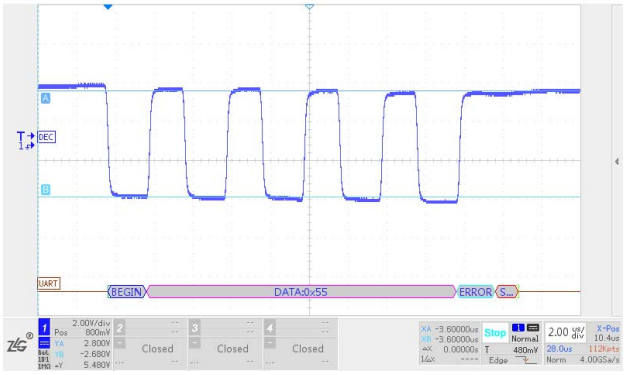


图1 RSM485ECHT 1200m 500kbps 不加终端电阻



终端电阻带来的问题

终端电阻虽然可以提高信号质量，但还具有以下几个问题：

1. 降低了驱动信号的幅值

RS-485 总线上的负载越大，RS-485 收发器输出差分电压幅值越低，RSM485ECHT 在 5m，500kbps 的情况下不加终端电阻和加终端电阻的波形如图 3 和图 4 所示，可以看出驱动信号在增加终端电阻后降低了 2V 左右。

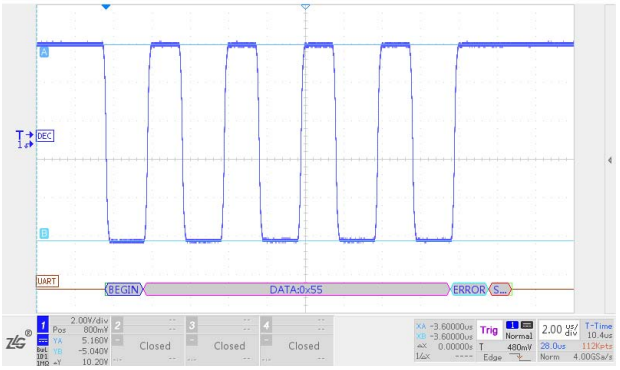


图3 RSM485ECHT 5m 500kbps 不加终端

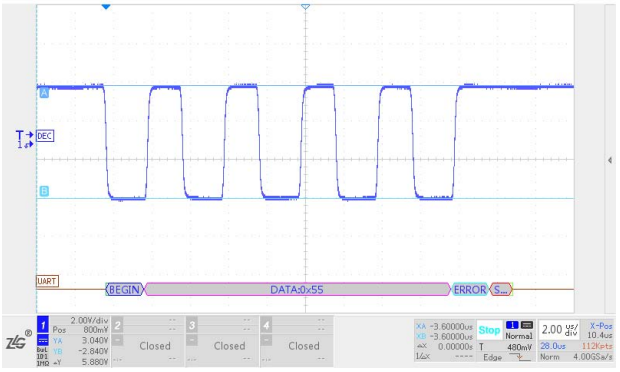


图4 RSM485ECHT 5m 500kbps 加终端

2. 增大了通信线上的压降

增加终端电阻使通信线缆上的电流增大，产生了较大的压差，降低了接收端的信号幅值。RSM485ECHT 在 1200m,115.2kbps 首端和末端的信号波形如图 5 和图 6 所示（0.75mm2 通信线），末端信号与首端信号相比下降了 0.7V 左右。

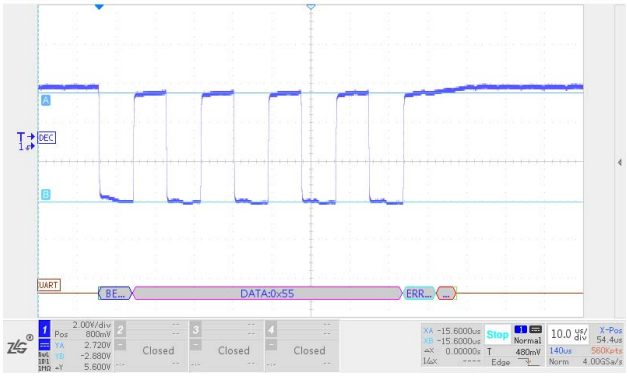


图5 RSM485ECHT 1200m 115.2kbps
加终端电阻 首端波形

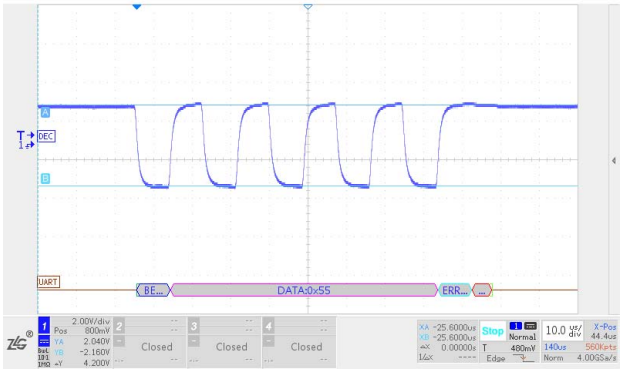


图6 RSM485ECHT 1200m 115.2kbps
加终端电阻 末端波形

3. 增大了收发器的功耗

增加终端电阻对于接收状态时的工作电流影响不大，但会大大增加驱动状态时的工作电流。以 RSM485ECHT 为例，RSM485ECHT 处于接收状态时工作电流为 20mA 左右，在驱动状态不加终端电阻时工作电流为 27mA 左右，在驱动状态加终端电阻时工作电流为 83mA 左右，可以看出终端电阻大大增加了 RS-485 收发器的功耗，对于有功耗要求的应用场合，应谨慎使用终端电阻。

4. 降低总线空闲时的差分电压

如图 7 所示为两个 RSM485ECHT 通信示意图。

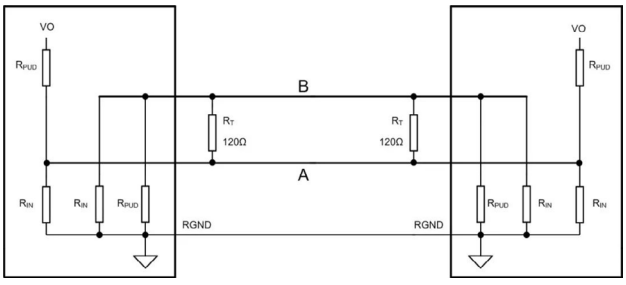


图7 RSM485ECHT通信等效示意图

当两个模块都处于接收状态时，可以根据基尔霍夫电流定律对节点 A 和节点 B 列出下列公式：

$$\frac{VO-V_A}{R_{PUD}\parallel R_{PUD}}=\frac{V_A-V_B}{R_T\parallel R_T}+\frac{V_A}{R_{IN}\parallel R_{IN}}$$

$$\frac{V_A-V_B}{R_T\parallel R_T}=\frac{V_B}{R_{PUD}\parallel R_{PUD}}+\frac{V_B}{R_{IN}\parallel R_{IN}}$$

其中：RPUD 为 RSM485ECHT 内置上下拉电阻，120kΩ；
RIN 为 RSM485ECHT 输入阻抗，96kΩ；
根据上述公式可以计算 AB 之间的差分电压为：

$$V_A-V_B=\frac{VO}{R_{PU}\parallel R_{PU}}\times\frac{1}{\frac{2}{R_T\parallel R_T}+\frac{1}{R_{PUD}\parallel R_{PUD}}+\frac{1}{R_{IN}\parallel R_{IN}}}=\frac{5}{60k}\times\frac{1}{\frac{2}{60}+\frac{1}{60k}+\frac{1}{48k}}=2.5mV$$

由于 RSM485ECHT 的门限电平为 -200mV～-40mV，所以在上述情况下，模块仍然输出高电平，保证总线空闲时不会误接收数据。但对于门限电平为 -200mV～+200mV 的 RS-485 收发器，输出电平为不确定状态，此时有可能误接收数据。

如何解决增加终端电阻后空闲状态的问题？

对于空闲状态的问题有两个解决方法：

1. 使用类似 RSM485ECHT 的模块（门限电平为 -200mV～-40mV），当 RS-485 总线的差分电压大于 -40mV 时 RS-485 收发器的输出即为高电平。

2. 使用 RSM485PCHT 或 RSM485PHT 等带有输出隔离电源的模块，可以通过在外部增加较小的上下拉电阻将 RS-485 总线的空闲状态时的电压拉到 +200mV 以上（一般要留有 100mV 或 200mV 以上的裕量），保证空闲时 RS-485 总线差分电压不处于门限电平范围内，但上下拉电阻值不能太小，一般总线上拉（或下拉）并联值要大于 375Ω。

什么时候需要加终端电阻？

1. 通信速度低或者通信距离近的情况下建议不加终端电阻

通信速度低或者通信距离近的情况下，信号反射对通信信号的影响不大，而且不加终端电阻可以大大降低功耗，并且通过加较大上下拉电阻值即可保证 RS-485 总线空闲时具有较高的差分电压幅值，提高了通信的可靠性。

2. 通信距离较长且通信速度较快，对信号质量要求较高的情况

此时可以增加终端电阻，防止阻抗突变引起的信号反射问题，提高信号质量，但应确保在总线空闲时总线的差分电压不处于门限电平范围内。

3. 对功耗有要求且通信距离较长的情况

一般在一个位的中间时间对信号进行采样，由于低通信速度的情况下，每一个位的时间较长，所以在到达采样点时反射信号已被消耗掉，对通信已无影响。RSM485ECHT 在 1200m 9600bps 不加终端电阻首端和末端的波形如图 8 和图 9 所示，可以看出反射信号在到达每一个位中间前就已经被消耗掉了。

所以对 RS-485 的收发器的功耗有较高要求且通信距离较长的应用，应适当降低通信的速度。

感知控制 ▼

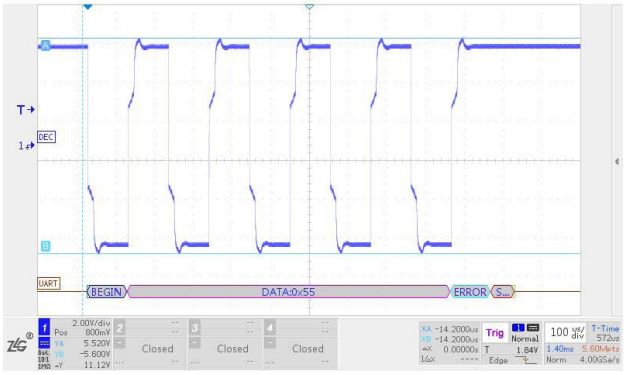


图8 RSM485ECHT 1200m 9600bps
不加终端 首端波形

图9 RSM485ECHT 1200m 9600bps
不加终端 末端波形

【RS-485总线】详解RS-485上下拉电阻的选择

ZLG 致远电子 2024-09-20 11:37:00

RS-485 总线广泛应用于通信、工业自动化等领域，在实际应中，通常会遇到是否需要加上下拉电阻以及加多大的电阻合适的问题，下面我们将对这些问题进行详细的分析。

为什么需要加上下拉电阻？

根据 RS-485 标准，当 485 总线差分电压大于 +200mV 时，485 收发器输出高电平；当 485 总线差分电压小于 -200mV 时，485 收发器输出低电平；当 485 总线上的电压在 -200mV ～ +200mV 时，485 收发器可能输出高电平也可能输出低电平，但一般总处于一种电平状态，若 485 收发器的输出低电平，这对于 UART 通信来说是一个起始位，此时通信会不正常。

当 485 总线处于开路（485 收发器与总线断开）或者空闲状态（485 收发器全部处于接收状态，总线没有收发器进行驱动）时，485 总线的差分电压基本为 0，此时总线就处于一个不确定的状态。同时由于目前 485 芯片为了提高总线上的节点数，输入阻抗设计的比较高，例如输入阻抗为 1/4 单位阻抗或者 1/8 单位阻抗 (单位阻抗为 12kΩ，1/4 单位阻抗为 48kΩ)，在管脚悬空时容易受到电磁干扰。

因此为了防止 485 总线出现上述情况，通常在 485 总线上增加上下拉电阻（通常 A 接上拉电阻，B 总线下拉电阻）。若使用隔离 RS-485 收发模块（例如 RSM485PCHT），由于模块内部具有上下拉电阻（对于 RSM485PCHT，内部上下拉电阻为 24kΩ），因此在模块外部一般不需要增加上下拉电阻。

什么情况下需要加上下拉电阻？

当遇到信号反射问题时，通常会通过增加匹配电阻来避免信号反射，以 1 对 1 通信为例，如图 1 所示。由于 485 总线通常使用特性阻抗为 120Ω 的双绞线，因此在 485 总线的首尾两端增加 120Ω 终端电阻来避免信号反射问题。

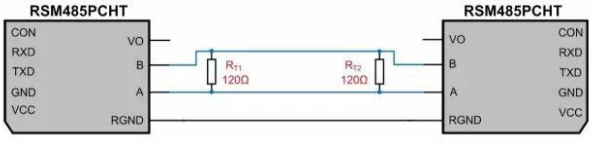


图1 两个RSM485PCHT模块通信电路

根据 RSM485PCHT 的具体参数（如表 1）可以得到如图 2 所示等效电路，其中 RPU、RPD 为模块内部在 485 总线上加的上下拉电阻，RIN 为模块的输入阻抗。

参数名称	测试条件	符号	参数值			单位
			min	typ	max	
输入电压	--	VCC	4.75	5.00	5.25	V
输出电压	5V 输入电压	VO	4.95	5.05	5.15	V
收发器输入阻抗	--	R_{IN}	120	150	--	kΩ
内置上下拉电阻	--	R_{PU}、R_{PD}	--	24	--	kΩ
门限电平	--	V_{AB}	-200	--	+200	mV
节点数	--	n	2	--	64	个

表1 RSM485PCHT参数

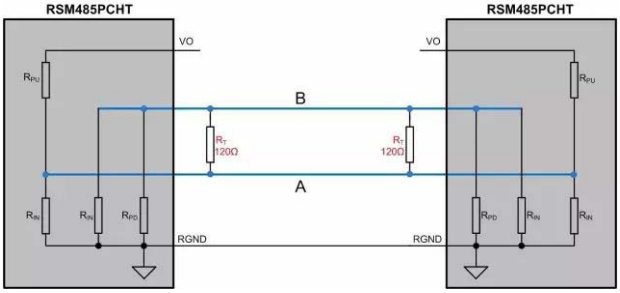


图2 RSM485PCHT通信等效示意图

当两个模块都处于接收状态时，可以根据基尔霍夫电流定律对节点 A 和节点 B 列出下列公式：

$$\frac{VO-V_A}{R_{PU}\parallel R_{PU}}=\frac{V_A-V_B}{R_T\parallel R_T}+\frac{V_A}{R_{IN}\parallel R_{IN}}$$

$$\frac{V_A-V_B}{R_T\parallel R_T}=\frac{V_B}{R_{PD}\parallel R_{PD}}+\frac{V_B}{R_{IN}\parallel R_{IN}}$$

$$R_{PU}=R_{PD}$$

根据上述公式可以计算 AB 之间的差分电压为：

$$V_A-V_B=\frac{VO}{R_{PU}\parallel R_{PU}}\times\frac{1}{\frac{2}{R_T\parallel R_T}+\frac{1}{R_{PU}\parallel R_{PU}}+\frac{1}{R_{IN}\parallel R_{IN}}}=\frac{5}{12k}\times\frac{1}{\frac{2}{60}+\frac{1}{12k}+\frac{1}{60k}}=12.46mV$$

此时模块已处于不确定状态，模块接收器可能输出为高电平，也可能输出为低电平，这时就需要在模块外部增加上下拉电阻保证模块在空闲时不处于不确定状态。

上下拉电阻如何取？

假设模块的输出电源电压 VO 相同，由于 RGND 接在一起，因此可以认为模块内部的上拉电阻是并联在一起的，为了方便解释，对图 2 的电路进行整理，如图 3 所示，在模块外部增加上下拉电阻可以选择只增加一组，也可以选择在每个模块都增加上下拉电阻，为了解释方便，我们在 485 总线上增加一组上下拉电阻。

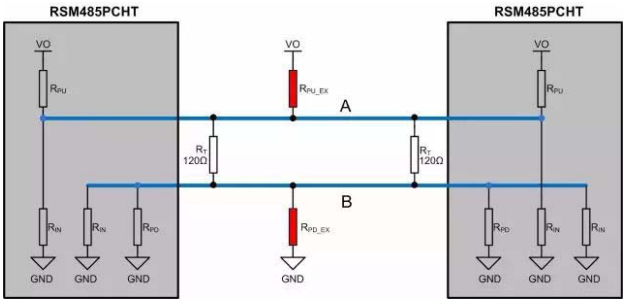


图3 RSM485PCHT通信等效电路图

其中：

- RPU 为模块内部上拉电阻，RPD 为模块内部的下拉电阻，本例中为 24kΩ；
- RIN 为模块接收器输入阻抗，本例取最小值为 120kΩ；
- RT 为终端电阻，本例取 1 20Ω；
- RPU_EX 为模块外部所加的上拉电阻，RPU_EX 为模块外部所加的下拉电阻。

由于 RSM485PCHT 的门限电平为 -200mV ～ +200mV，一般留有 100mV 或 200mV 的电压裕量，本例留有 100mV 的电压裕量，根据前面所推导的差分电压公式，可以得到下面计算公式：

$$V_A - V_B = \frac{VO}{R_{PU} \parallel R_{PU} \parallel R_{PU\_EX}} \times \frac{1}{\frac{2}{R_T \parallel R_T} + \frac{1}{R_{PU} \parallel R_{PU} \parallel R_{PU\_EX}} + \frac{1}{R_{IN} \parallel R_{IN}}} \geq 0.3V$$

由于 RSM485PCHT 在供电电压范围为 4.75V ～ 5.25V，取 VO=4.75V（最低输入电压 VCC=4.75V 情况下），可得：

$$R_{PU} \parallel R_{PU} \parallel R_{PU\_EX} \leq \frac{\frac{VO}{V_A - V_B} - 1}{\frac{2}{R_T \parallel R_T} + \frac{1}{R_{IN} \parallel R_{IN}}} = \frac{\frac{4.75}{0.3} - 1}{\frac{2}{60} + \frac{1}{60k}} = 444.78\Omega$$

由 RPU=24kΩ，可得 RPU_EX=RPU_EX=461.9Ω，由于计算出的电阻值为最大值，因此可以选择在 485 总线上仅加一组 410Ω 或 390Ω 的上下拉电阻，或者加两组 910Ω 上下拉电阻。

如何验证上下拉电阻取值？

上述计算仅考虑了 485 总线空闲状态时不处于不确定状态，并没有考虑 485 收发器的驱动能力和所用元器件的功耗等问题。外部所加上下拉电阻越小，可以将 485 总线空闲状态差分电压保持的越高，但与此同时，终端电阻和上下拉电阻的功耗也越大，对 485 收发器的驱动能力要求也越高，当超过 485 收发器的驱动能力时，也会导致通信失败。

根据 RS-485 标准，当接收器的输入阻抗为单位阻抗时（最小为 12k），总线上最多可以接 32 个节点，485 的差分负载最大为 54Ω，此时差分输出电压最小为 1.5V。

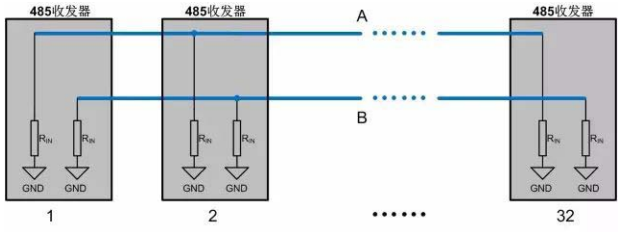


图4 485总线连接32个节点等效示意图

如图 4 所示，我们可以看到当 485 总线上接有 32 个节点时，总线 A 或 B 的共模负载为：

$$R_{LCM\_A} = R_{LCM\_B} = \frac{R_{IN}}{n} = \frac{12k}{32} = 375\Omega$$

由此可见，对于 RS-485 的标准来说，A 总线或 B 总线的最大共模负载为 375Ω。

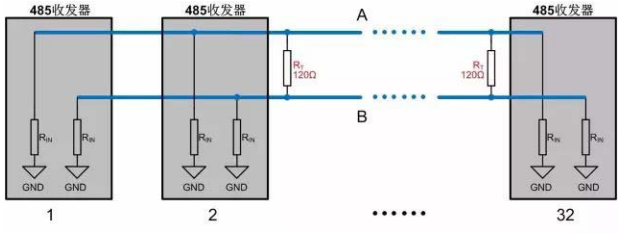


图5 485总线增加终端电阻等效示意图

当增加终端电阻后，可以发现 485 总线的共模负载没有发生变化，但差模负载急剧减小，差模负载为：

$$R_{LDM\_AB} = \frac{R_T}{2} \parallel \left(\frac{R_{IN}}{n} + \frac{R_{IN}}{n}\right) = 60 \parallel (375 + 375) = 55.56\Omega$$

因此当 485 总线的节点数达到最多以及增加终端电阻后，485 总线的差模负载仍大于 54Ω，根据 RS-485 的标准，差分输出电压最小为 1.5V。

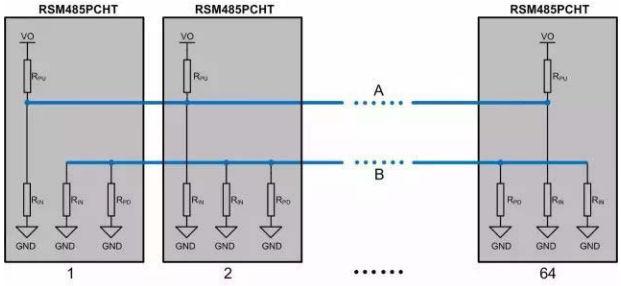


图6 RSM485PCHT 64个节点等效示意图

以 RSM485PCHT 为例说明增加上下拉电阻的情况，如图 6 所示，总线 A 或 B 的共模负载为：

$$R_{LCM\_A} = R_{LCM\_B} = \frac{R_{IN}}{n} \parallel \frac{R_{PU}}{n} = \frac{120k}{64} \parallel \frac{24k}{64} = 1875 \parallel 375 = 312.5\Omega$$

实际测试上述情况，驱动输出的最小差分电压 3.02V，这个电压远大于 RS-485 标准规定的最小差分输出电压 1.5V。

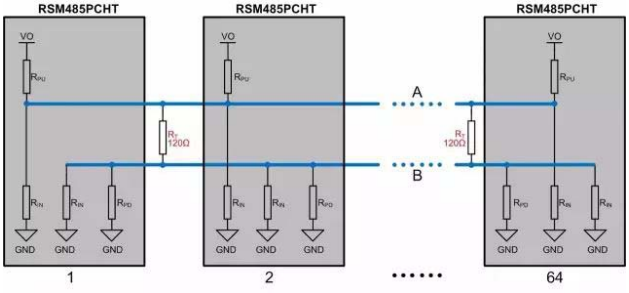


图7 RSM485PCHT 64个节点增加终端电阻示意图

当在 485 总线上增加终端电阻时，可以看出总线 A 或 B 的共模负载并没有发生变化，而差分阻抗有了较大的变化，此时差模负载为：

计算出的差模负载要略大于 RS-485 标准规定的最大负载为 54Ω，我们对 RSM485PCHT 进行实际测试，其输出差分电压 1.58V，略大于标准规定的最小电压。

当差模负载为 54Ω（485 总线接两个 120Ω 终端电阻并且上拉电阻（下拉电阻）与收发器内阻的并联值为 270Ω）时，RSM485PCHT 的差分输出电压为 1.52V（实测值），基本和 RS-485 标准相同。当差模负载为 41.54Ω（485 总线接两个 120Ω 终端电阻并且上拉电阻（下拉电阻）与收发器内阻的并联值为 135Ω）时，RSM485PCHT 的差分输出电压在 1.17V 左右 (实测值)，在这种情况下可以通信。但 485 收发芯片手册中规定的最大差模负载通常为 54Ω，即在 485 总线上增加两个 120Ω 后，上拉电阻（下拉电阻）与收发器输入阻抗的并联值应大于 270Ω。同时为了保证稳定可靠通信，一般 485 总线的上拉电阻（下拉电阻）与收发器输入阻抗的并联值应大于 375Ω。

总结

1. 通信线应选用屏蔽双绞线，屏蔽层应单点接大地；
2. 当我们没有遇到信号反射问题时，尽量不要使用终端电阻；
3. 如果使用终端电阻，我们可以通过上下拉电阻调节 485 总线在空闲状态的电压值，保证不处于门限电平（-200mV ～ +200mV 或 -200mV ～ -40mV）范围内；
4. 当我们增加上下拉电阻时，上拉电阻（下拉电阻）与收发器输入阻抗的并联值应大于 375Ω。

【技术分享】RS-485保护电路结电容对信号质量的影响

ZLG 致远电子 2024-09-27 11:37:55

RS-485 总线被广泛应用在工业环境，可能有高等静电或浪涌干扰，工程师通常会使用气体放电管和 TVS 管搭建防护电路，但该电路的结电容较高，应用不当将会影响通讯。本文将为大家介绍一种低结电容的外围电路。

常用RS-485保护电路

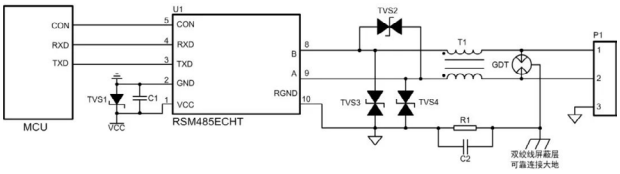


图1 保护电路1

如图 1 所示的保护电路，气体放电管将接口处的大部分浪涌电流泄放，共模电感滤除共模信号的干扰，TVS 进一步降低气体放电管后的残压，从而保护后级电路。RSM485ECHT 模块应用图 1 所示保护电路可以达到接触静电 ±8kV，共模浪涌 ±4kV，差模浪涌 ±2kV，满足大部分工业现场对 RS-485 节点静电和浪涌等级的要求。

图 1 所示保护电路虽然保护能力较强，但其结电容较大，A-RGND 或 B-RGND 结电容为 2.5nF 左右，当总线上有较多节点均使用图 1 保护电路进行组网时，总线的电容量较大，信号反射以及信号边沿趋于平缓使信号质量变差，甚至会导致通信异常。

总线电容导致的信号反射问题

当信号在通信线上传输，到达 RS-485 节点上的保护电路时，保护电路的结电容使信号受到的瞬时阻抗发生变化，一部分信号将被反射，另一部分发生失真并继续传播下去。

图 2 所示为 RSM485ECHT 单节点发送波形，图 3 为 RS-485 总线接 6 个保护电路的示意图，每个节点之间的距离在 30cm 左右，使用双绞线手拉手连接，图 4 和图 5 分别为在总线上接 6 个图 1 所示电路的波形测试点 1 和波形测试点 6（图 3 中标注的位置）的波形，波形的上升 / 下降时间变长，并且波形测试点 1 波形变成了台阶形状。

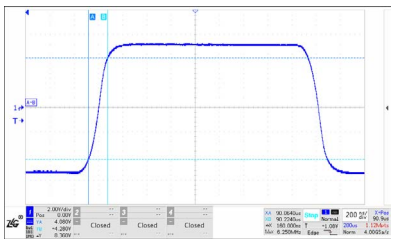


图2 RSM485ECHT单节点RS-485接口差分波形

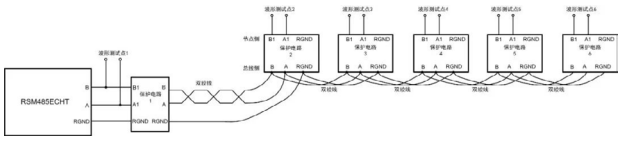


图3 总线接6个保护电路连接示意图

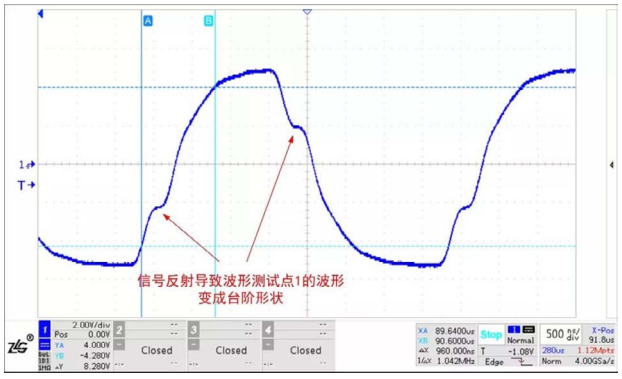


图4 RSM485ECHT接6个保护电路波形测试点1波形

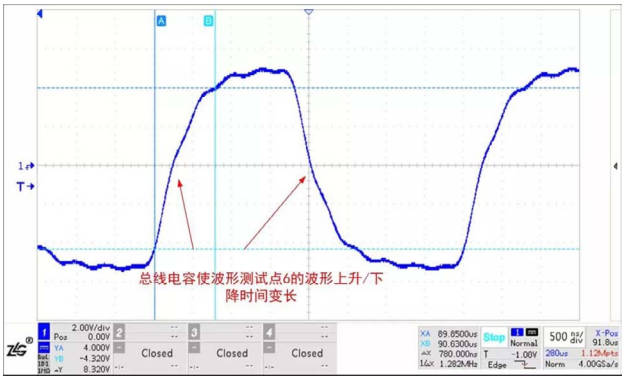


图5 RSM485ECHT接6个保护电路波形测试点6波形

RSM485ECHT 的 RS-485 接口驱动能力较强，如下为使用相同测试条件测试市场上常用的 RS-485 收发器芯片测试波形，可以看出其波形已被严重干扰，且反射波形已到达 RS-485 芯片门限电平附近，有可能引起通信异常。因此在实际应用中应选择驱动能力较强的收发器。

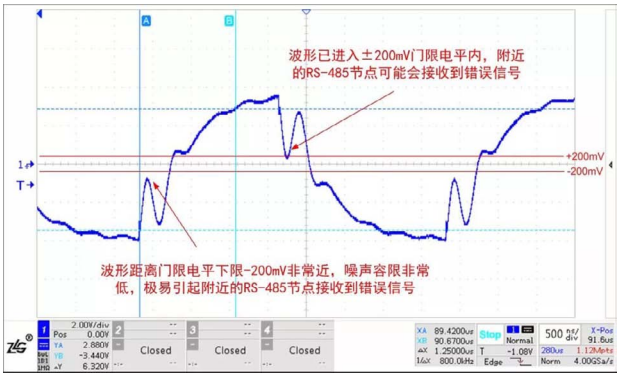


图7 某RS-485收发器接6个保护电路波形测试点6波形

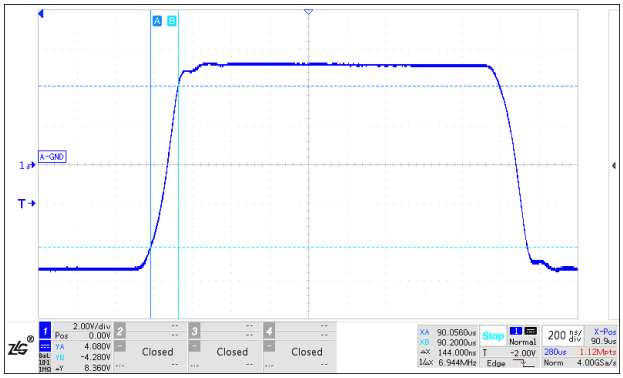


图10 RSM485ECHT接6个保护电路2波形测试点6波形

低结电容保护电路

当通信节点数较多，可以使用如图 8 所示保护电路，其 A-RGND 或 B-RGND 的结电容仅为 20pF，虽然 TVS 结电容较大，但普通二极管结电容非常小，TVS 与普通二极管的结电容为串联关系，因此可以减小保护电路的结电容。使用图 8 进行图 3 所示的组网，测试点 1 的波形如图 9 所示，测试点 6 波形如图 10 所示，波形基本未发生变化。

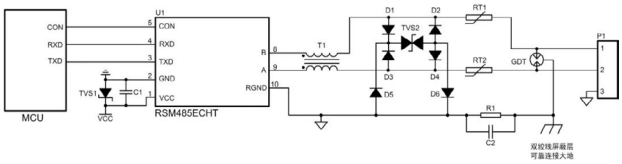


图8 保护电路2（低结电容）

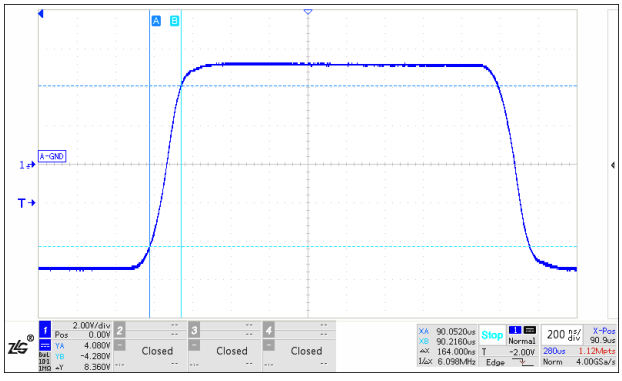


图9 RSM485ECHT接6个保护电路2波形测试点1波形

总结

总线上挂载的保护电路会使信号受到的瞬时阻抗发生变化，导致信号反射，当总线上的节点数较多，总线的电容量较大，会对总线波形造成干扰，影响通信信号质量，因此为减小保护电路对总线通信的影响，在实际应用可以选择驱动能力较强的收发器，并且保护电路若使用图 1 所示保护电路，应选择低结电容 TVS，也可选择使用如图 8 所示的低结电容保护电路。

2024/9 第9期

微文摘

ZLG MICRO DIGEST

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 www.zlg.cn
欢迎拨打全国服务热线 400-888-4005