

ZLG 致远电子

微文摘

ZLG MICRO DIGEST

2023/3 第3期

月刊



ePort系列集成式RJ45插座

为解决以太网PHY芯片开发痛点而生

芯片短缺?

网络链接不稳定?

电路设计不合理?

EMC测试不过?



货源保障

稳定的货源保障，
无需担心货源问题



稳定性强

稳定性强，一致性好，
无需担心设计问题



体积小

高集成体积小，节省占板面积，
有效降低综合成本

选型表

产品型号	ePort-M	ePort-G
速率	10M/100Mbps 自适应	10M/100M/1000Mbps 自适应
供电电压	3.3V±3%	3.3V±3%
工作环境	-40°C ~+85°C	-40°C ~+85°C
存储温度	-45°C ~+90°C	-45°C ~+90°C
相对湿度	5%~95% (非冷凝)	5%~95% (非冷凝)
电磁兼容	静电放电抗扰度试验: 4 级 雷击 (浪涌) 抗扰度试验: 3kV 电快速瞬变脉冲群抗扰度试验: 3 级 射频场感应的传导骚扰抗扰度试验: 10V	静电放电抗扰度试验: 4 级 雷击 (浪涌) 抗扰度试验: 3kV 电快速瞬变脉冲群抗扰度试验: 3 级 射频场感应的传导骚扰抗扰度试验: 10V



致远电子官方网站



致远电子官方微信

CONTENTS

目录

技术平台

EsDA 平台

【EsDA 应用】如何用 5 分钟快速建立 CAN 通信	04
【EsDA 应用】Modbus 应用详解	08
【EsDA 协议转换】Modbus 转 OPC UA 协议	14
【EsDA 协议转换】Modbus 转 CAN/CANFD 协议	20

ZLG 云平台

【解决方案】如何通过 IoT 云对设备数据进行自定义统计?	25
【解决方案】设备如何上报数据到 ZWS 云平台?	26
【产品应用】如何通过 IoT 云平台对 CAN 数据进行实时 DBC 解析?	27
【产品应用】IoT 云平台中告警触发规则的应用	29

边缘计算

核心板

【技术分享】基于 M6Y2C 平台实现 LoRa 远距离传输	30
【产品应用】M1126 高清摄像头应用实例	32

行业控制器

【技术分享】为何 EtherCAT 在运动控制总线中如此受欢迎?	33
【深度解析】EM-1000 储能网关的 CAN 性能有多强?	36
【产品应用】ZLM3100S 应用 -- 快速驱动永磁同步电机风机	40

互联互通

无线通讯

【产品应用】ZigBee 技术的典型应用方案介绍	42
【产品应用】如何使用 ZM602 模组实现公里级 Wi-Fi 通信?	43

接口与协议转换

【深度解析】你知道我们的 ePort 模块经历了多少刀山火海 才能交付客户吗?	45
---	----

感知控制

电源与隔离

【解决方案】充电桩从电源隔离到温度监测的一体化方案设计	48
【技术分享】电源模块的本手与应用妙手	50
【产品应用】小器件助力舞台灯光可靠连接	52
【技术分享】电源模块应用疑惑与解惑	53

数据采集

【解决方案】致远电子隔离测温技术为精密恒温箱赋能	55
--------------------------	----

致远微电子芯片

【技术分享】"喂狗", 也要讲究方法	57
【产品应用】ZL6300 在加载时间过长系统的巧妙应用	59

【EsDA应用】 如何用5分钟快速建立CAN通信

ZLG 致远电子 2023-03-01 11:32:44

CAN 总线的高可靠性、高性能等优点使其在工业测控以及工业自动化等领域中被大量广泛使用。本文基于 EsDA 开发平台，为您简单介绍如何用 5 分钟快速建立 CAN 通信。

简介

CAN 是 ISO 国际标准化的串行通信协议。由于性能高，可靠性高现被广泛应用在汽车，医疗，船舶，工业自动化等其他方面。因为只需要两根线就可在多个设备间进行数据通信所以被广泛使用。本文基于 EsDA 开发平台，为您简单介绍如何用 5 分钟快速建立 CAN 通信。

基本工作原理：通过 USBCANFD-200U 接口卡实现主板与 PC 实现 CAN(FD) 通信。

本应用选用 EPC6450-AWI 开发板，USBCANFD-200U 接口卡与 PC 连接，使其能与接入 CAN(FD) 网络的开发板进行数据收发，USBCANFD 不同系列产品支持 1~8 路 CAN(FD) 通信，可根据不同需求进行选择，本文采用的 200U 系列仅支持 2 路通信。该实验用到了 timer，fscript，canfd_out，canfd_in，fscript 和 log 节点。



图1 EPC6450-AWI开发板



图2 USBCANFD-200U接口卡

VS Code调试环境搭建

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

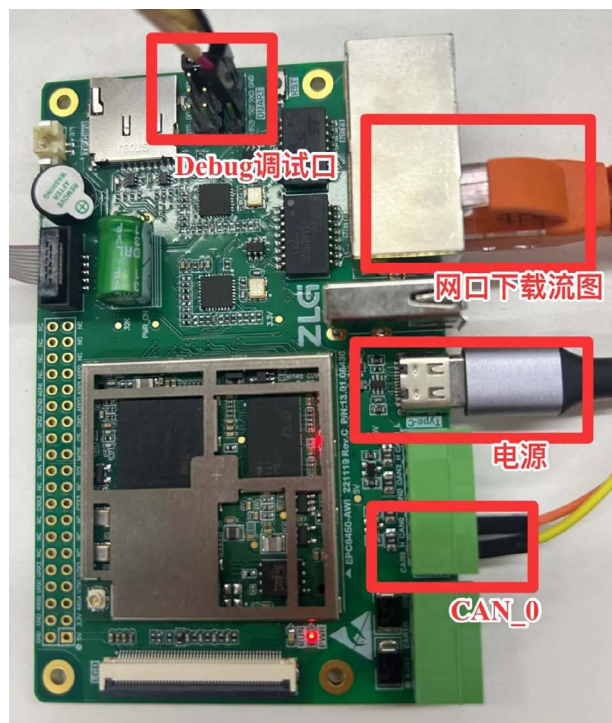
- EsDA MPC-ZC1 入门（一）—— 软件安装
- EsDA MPC-ZC1 入门（二）—— LED 控制
- 【EsDA 应用】5 分钟实现一个串口通信业务

1. 硬件准备

1.1 准备一个 USBCANFD-200U 接口卡，并在 PC 上下载并安装 ZCANPRO 的上位机软件。



1.2 准备 EPC6450-AWI 开发板，并按照下图所示连接好硬件。



将 EPC6450-AWI 开发板上的 CAN_0 与 USBCANFD-200U 随机一路进行连接，并将 USBCANFD-200U 接口卡的 USB2.0 连接到 PC 上，建立好硬件连接。

节点介绍

实现 CAN(FD) 总线通信，需要使用到 canfd 相关的 AWFlow 节点，包括 canfd_out，canfd_in 和 canfd 三个节点。

2.1 canfd

CANFD 通信协议配置节点，属于隐式节点，不会显示在画布中，需要配套相应功能节点一起使用。

2.1.1 属性

The image shows a configuration window for a CANFD node. It includes fields for port (/dev/can0), CAN controller type (CANFD), non-standard ISO mode (enable), arbitration rate (1000000), arbitration sampling point precision (80), arbitration sampling point range (75-85), arbitration sampling point denominator (100), data rate (500000), data sampling point precision (80), data sampling point range (75-85), data sampling point denominator (100), filter enable (disable), and a filter table (60 - 3200).

端口 (port): 设备端口路径, 用于系统打开设备;

CAN 控制器类型 (can_type): 设置 CAN 控制器类型。值为 0 表示 CAN; 值为 1 表示 CANFD;

非标准 ISO 模式 (non_iso_mode): 开启 / 关闭非标准 ISO 模式。值为 0 表示关闭; 值为 1 表示开启;

仲裁域波特率 (nominal_bps): 设置仲裁域波特率, 单位是 bps, 值域是 0~1000000;

仲裁域采样点精确的分子值 (nominal_spf): 设置仲裁域采样点精确的分子值, 采样点用分数来表示, 如需要将采样点设置为 87.5%, 则设置采样点的分母为 1000, 分子为 875 即可。因硬件原因可能导致不能精确地配置采样点, 增加了两个分子值来表示范围。值域是 0~10240;

仲裁域采样点最小的分子值 (nominal_spf_min): 设置仲裁域采样点最小分子值, 值域是 0~10240;

仲裁域采样点最大的分子值 (nominal_spf_max): 设置仲裁域采样点最大分子值, 值域是 0~10240;

仲裁域采样点分母值 (nominal_spd): 设置仲裁域采样点分母值, 值域是 0~10240 数据域波特率 (data_bps): 设置数据域波特率, 值域是 0~5000000;

数据域采样点精确的分子值 (data_spf): 设置数据域采样点精确的分子值, 采样点用分数来表示, 如需要将采样点设置为 87.5%, 则设置采样点的分母为 1000, 分子为 875 即可。因硬件原因可能导致不能精确地配置采样点, 增加了两个分子值来表示范围。值域是 0~10240;

数据域采样点最小的分子值 (data_spf_min): 设置数据域采样点最小的分子值。值域是 0~10240;

数据域采样点最大的分子值 (data_spf_max): 设置数据域采样点最大的分子值。值域是 0~10240;

数据域采样点分母值 (data_spd): 数据域采样点分母值。值域是 0~10240;

滤波器使能 (filter_enable): 开启 / 关闭滤波器。值为 0 表示关闭; 值为 1 表示开启;

滤波表 (filter_table): 滤波表, 滤波表采用 json 数组的形式设置, 最大支持 32 条, 符合滤波表条件的报文将被接收, 不符合的报文会被过滤, 但错误报文不会被过滤。滤波表格式如下: id 表示报文 ID; msk 表示 ID 掩码, ID 掩码部分相同的报文将被接收; recv_std 表示是否接收标准帧; recv_ext 表示是否接收扩展帧; recv_data 表示是否接收数据帧; recv_rmt: 是否接收远程帧。

* 配置节点 (config类型) 只存在节点属性, 不具备输入输出功能。

2.2 canfd_in

2.2.1 属性

The image shows the configuration window for the 'canfd_in' node. It includes fields for name (canfd_in), display name (显示名称), and associated configuration (can_0).

- 名称 (name): 节点名称, 用于索引查找本节点;
- 显示名称 (displayName): 用于画布上显示的名称;
- 关联配置 (config): 绑定一个 canfd 的配置节点。

2.2.2 输出

- payload: CAN 报文, 1-8 字节为时间戳, 9-12 字节为 ID, 13-14 字节为报文标志位【位 0: 发送帧标志 (单次发送)。位 1: CANFD 帧标志, 位 2: 远程帧标志, 位 3: 扩展帧标志, 位 4: 错误帧标志, 位 5: 转换可变速率, 位 6: 错误状态指示 (0: 被动错误, 1: 主动错误)】, 15-16 字节为通道号 (低 5 位) 和数据长度 (高 11 位), 17 字节到末尾为报文数据;
- payloadLength: 负载长度。
- * pump 类型节点通常不具备数据输入。

2.3 canfd_out

2.3.1 属性



- 名称 (name): 节点名称, 用于索引查找本节点;
- 显示名称 (displayName): 用于画布上显示的名称;
- 关联配置 (config): 绑定一个 canfd 的配置节点。

2.3.2 输入

- payload: CAN 报文, 1-8 字节为时间戳, 9-12 字节为 ID, 13-14 字节为报文标志位【位 0: 发送帧标志 (单次发送), 位 1: CANFD 帧标志, 位 2: 远程帧标志, 位 3: 扩展帧标志, 位 4: 错误帧标志, 位 5: 转换可变速率, 位 6: 错误状态指示 (0: 被动错误, 1: 主动错误)】, 15-16 字节为通道号 (低 5 位) 和数据长度 (高 11 位), 17 字节到末尾为报文数据;
- payloadLength: 负载长度。
- * sink 类型节点通常不具备数据输出。

业务处理

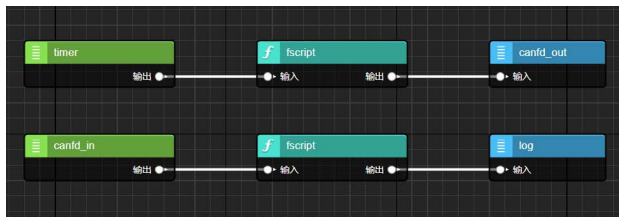
本应用主要通过 USBCANFD-200U 接口卡将 PC 与接入 CAN 网络的 EPC6450-AWI 开发板进行数据收发, 具体实现如下:

1. 添加节点并连线

本实验主要执行两个步骤:

- EPC6450-AWI 开发板上的 CAN_0 通过 CAN 通信协议, canfd_out 节点定时向 USBCANFD-200U 接口卡发送 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 八个字节的数据, 发送数据可在 PC 上的工具 ZCANPRO 上查看。
- PC 上的工具 ZCANPRO 通过 CAN 总线通信协议定时发送 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77 八个字节的数据, EPC6450-AWI 开发板上通过 canfd_in 这个节点去接收数据, 并通过 log 最终打印可查看。

采用的节点有 timer, fscript, canfd_out, canfd_in, fscript 和 log 节点。



2. 配置节点

双击 timer 节点, 设置 canfd_out 节点发送数据的定时时间。

双击 canfd_out 节点的前驱节点 fscript, 结合 2.2 关于 canfd_out 节

点的介绍可知 canfd_out 节点有两个输入参数: payload;payloadLength。一个传递通信数据内容, 一个传递通信数据长度, 内容如下:

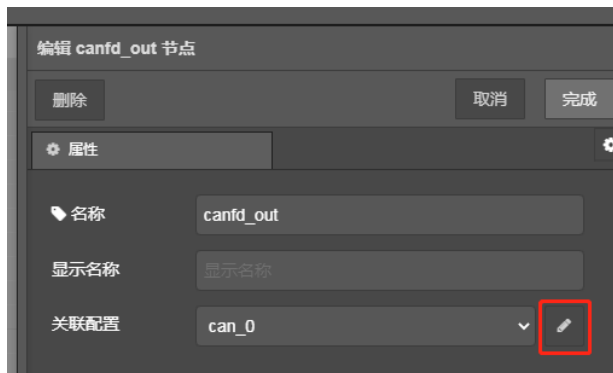
```
buf = wbuffer_create()
wbuffer_write_int8(buf, 202)
wbuffer_write_int8(buf, 150)
wbuffer_write_int8(buf, 9)
wbuffer_write_int8(buf, 124)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
```

```
wbuffer_write_int8(buf, 17)
wbuffer_write_int8(buf, 1)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 1)
```

```
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 1)
wbuffer_write_int8(buf, 2)
wbuffer_write_int8(buf, 3)
wbuffer_write_int8(buf, 4)
wbuffer_write_int8(buf, 5)
wbuffer_write_int8(buf, 6)
wbuffer_write_int8(buf, 7)
```

```
msg.payload = wbuffer_get_data(buf)msg.payloadLength = str(24)
```

双击 canfd_out 和 canfd_in 节点, 并点击配置节点名旁边的铅笔图标去配置 canfd 的属性。



因为 canfd_in 和 canfd_out 节点的配置节点都是同一个，所以配置信息一致。根据 2.2 中对配置节点 canfd 的属性介绍，配置如下：

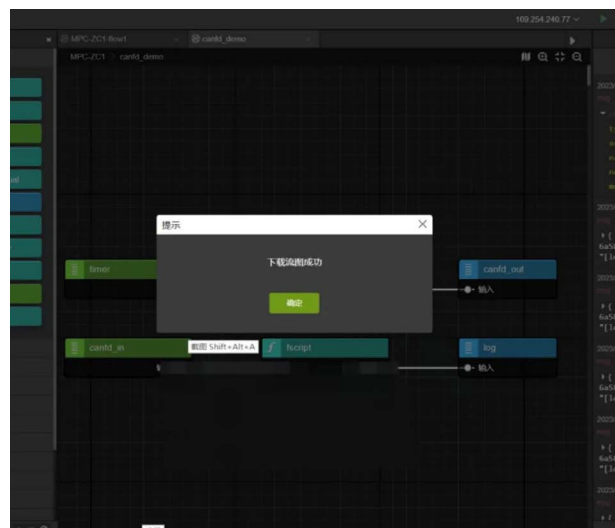
双击 canfd_in 的消费者节点 fscript，根据 2.2 节点介绍中对 canfd_in 的输出参数描述，fscript 的内容如下：

```
data_str = "data:"
len = msg.payloadLength
rbuf = rbuffer_create(msg.payload, msg.payloadLength)
for (var i = 0; i < len; i = i + 1) {
    data = rbuffer_read_uint8(rbuf);
    //print("data:" + data)
    data_str = data_str + data
    if (i < (len - 1)) {
        data_str = data_str + ","
    }
}
output.payload = data_str
```

3. 下载验证

因为 EPC6450-AWI 开发板是通过网口去下载流图，所以打开 AWFlow_Designer 后需要配置目标板 IP 和端口号。点击流图下载，显示下载流图成

功即可。



打开上位机 ZCANPRO 并开启对应的通道号，将波特率等参数选择与流图上的 canfd 配置节点参数同步一致。上位机 CAN 视图可以查看从上位机发送出去的数据和接收到的数据如下。

序号	时间戳	源通道	帧ID	CAN类型	方向	长度	数据
0	0.000000	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
1	2.999600	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
2	5.999600	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
3	8.999400	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
4	11.999300	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
5	14.999300	0	0x111	CAN	Rx	8	00 01 02 03 04 05 06 07
6	16.625700	0	0x100	CANFD	Tx	8	00 11 22 33 44 55 66 77
7	17.626900	0	0x100	CANFD	Tx	8	00 11 22 33 44 55 66 77
8	18.627800	0	0x100	CANFD	Tx	8	00 11 22 33 44 55 66 77
9	19.628400	0	0x100	CANFD	Tx	8	00 11 22 33 44 55 66 77
10	20.628400	0	0x100	CANFD	Tx	8	00 11 22 33 44 55 66 77

根据流图中 log 节点的打印信息可知：数据内容和长度与上位机发送的数据是一致的。

```
[11:12:48.159]收+◆[log][msg payload]: "data:248,50,209,112,61,0,0,0,1,0,0,2,0,0,0,17,34,51,68,85,102,119"
[log][msg payload]: "data:248,50,209,112,61,0,0,0,1,0,0,2,0,0,0,17,34,51,68,85,102,119"
tk_iostream_get_outstream:31 condition 0x00:0x11:0x22:0x33:0x44:0x55:0x66:0x77
[ERROR] [TOL tk_iostream_top_server_write] condition (get == sizeof(header)) failed!
we_outstream_write_payload:94 condition (get == sizeof(header)) failed!

[11:12:51.201]收+◆[log][msg payload]: "data:2,244,111,172,61,0,0,0,1,0,0,2,0,0,0,17,34,51,68,85,102,119"
[log][msg payload]: "data:2,244,111,172,61,0,0,0,1,0,0,2,0,0,0,17,34,51,68,85,102,119"
tk_iostream_get_outstream:31 condition (stream != NULL && stream->get_outstream != NULL) failed!
[ERROR] [TOL tk_iostream_top_server_write] cond
```

综上测试结果可知：实验效果已达成。

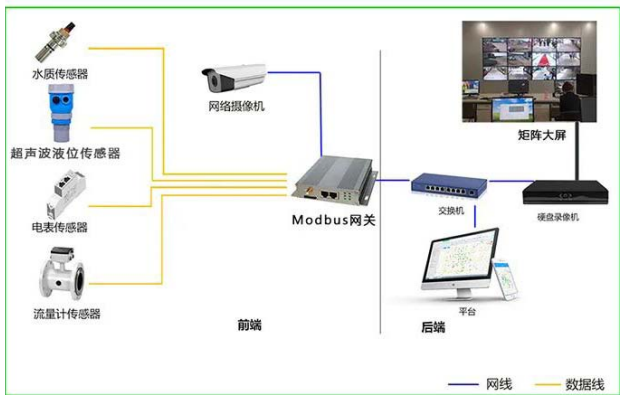
【EsDA应用】 Modbus应用详解

原创 研发部 ZLG 致远电子 2023-03-08 11:31:28

Modbus 作为一种公开、免费的现场总线，被广泛应用于工业电子领域。本文基于 EsDA 开发平台，为您详细介绍 Modbus 主机功能节点的使用方式，以及如何快速实现读取 Modbus 从机数据。

Modbus简介

Modbus 是一种串行通信协议，由于其公开、免费、易于部署和维护的优点，被广泛应用于工业电子领域，并且已经成为工业领域通信协议的业界标准。



图片来源于网络，侵权

以往我们在使用 Modbus 协议进行应用开发时，通常需要自己实现诸多的 Modbus 功能码，或者移植开源的第三方库，这个过程往往比较费时费力，耽误项目的进展。

基于 EsDA 设计的 Modbus 主机功能节点，可以通过简单的拖拽、连线方式，快速实现读取 Modbus 从机数据，搭配其他功能节点，可以快速搭建出一个集数据采集上报、远程控制于一体的物联网应用。

Modbus相关节点介绍

目前和 Modbus 主机相关的节点主要有 6 个，分别是 modbus_master_rtu、modbus_master_in、modbus_master_dynamic_in、modbus_master_out、modbus_parse_in 和 modbus_parse_out 节点。其中，modbus_master_rtu 节点属于配置节点，用于配置 Modbus 通信设备的参数信息，该节点提供了 Modbus RTU 和 Modbus TCP 的主机通信服务；modbus_master_in 和 modbus_master_dynamic_in 是 Modbus 的输入节点，主要用于读取从机设备的线圈量和寄存器数据；modbus_master_out 是数据的输出节点，用于写线圈量和寄存器；modbus_parse_in 和 modbus_parse_out 是扩展的 Modbus 功能节点，一般用来对输入 / 输出数据进行处理。

modbus_master_rtu节点

Modbus 主机的配置节点，用于配置与 Modbus 主机通信的从设备的通信参数，提供了 Modbus RTU 和 Modbus TCP 主机功能。该节点需要和对应的功能节点搭配使用，不会在画布中显示。

1. 属性

modbus_master_rtu 节点包含了 RTU 和 TCP 两种模式的配置，不同模式需要设置的属性不同。

1.1 RTU 模式配置属性

编辑 modbus_master_in 节点 > 编辑 modbus_master_rtu 节点

删除 取消 更新

属性

名称

modbus_master_rtu

显示名称

modbus_master_rtu

传输类型

rtu

串口设备名

/dev/Uart1

波特率

9600

数据位

8bits

校验位

none

停止位

1bits

响应时间

0

最大请求数量

3

- 名称：节点名称，用于索引查找本节点；
- 显示名称：用于画布上显示的名称；
- 传输类型：用于设置链路层的传输模式（rtu/tcp 可选）；
- 串口设备名：用于与从设备通信的串口设备名；
- 波特率：串口波特率参数；
- 数据位：串口数据位参数；
- 校验位：串口奇偶校验位参数；
- 停止位：串口停止位参数；
- 响应时间：从机应答超时时间，单位 ms；
- 最大请求数量：用于配置 Modbus 主机读写请求的最大数量。

1.2 TCP 模式配置属性

编辑 modbus_master_in 节点 > 编辑 modbus_master_rtu 节点

删除 取消 更新

属性

名称 modbus_master_rtu

显示名称 modbus_master_rtu

传输类型 tcp

IP地址 192.168.88.90

端口 502

响应时间 0

最大请求数量 3

- IP 地址：从机设备（服务器）的 IP 地址；
- 端口：从机设备的端口号；
- 响应时间：从机应答超时时间，单位 ms；
- 最大请求数量：用于配置 Modbus 主机读写请求的最大数量。

2. 使用方法

该节点的使用依附于 modbus_master_in、modbus_master_dynamic_in 以及 modbus_master_out 等节点，使用时选择对应的通信模式，根据从机信息配置相应的配置属性即可。

modbus_master_in节点

modbus_master_in 是 Modbus 的输入节点，主要用于读取从机设备的线圈量和寄存器数据并输出给消费者节点。其输出是原始数据的缓冲区，后续可连接 modbus_parse_in 节点对数据进行处理。

1. 属性

- 主机参数配置：输入节点依赖 modbus_master_rtu 节点，选择对应的配置节点即可；
- 读取模式：选择节点的触发方式，可选择以用户设定的输出周期定时向消费者节点输出数据；也可根据输入的信息（来自 push 节点）进行数据读取并输出；
- 从机 ID：从机设备的 ID 地址；
- 输出周期：用于周期读取模式设置读取和输出的周期；
- 寄存器地址：需要被读取的寄存器 / 线圈的起始地址；
- 读取数量：需要读取的寄存器 / 线圈数量；
- 寄存器类型：用于选择读取目标的类型，可选线圈量、离散量、输入寄存器、保持寄存器。

编辑 modbus_master_in 节点

删除 取消 完成

属性

名称 modbus_master_in

显示名称 modbus_master_in

主机参数配置 modbus_master_rtu

读取模式 periodic read data

从机ID 1

输出周期 1000

寄存器地址 0

读取数量 1

寄存器类型 read input registers

2. 输入

该节点属于 pump 类型节点，一般不需要数据输入，但可以使用 push 节点来实现数据输入。

- slaveID：从机设备的 ID 地址；
- address：需要被读取的寄存器 / 线圈的起始地址；
- reg_num：需要读取的寄存器 / 线圈数量；
- reg_type：需要读取的寄存器 / 线圈类型。

3. 输出

- slaveID：从机设备的 ID 地址，通常用于后级节点区分设备；
- address：读取的寄存器起始地址；
- reg_num：读取到的寄存器 / 线圈数量；
- payload：缓冲区，存储读取到的数据；
- payloadLength：读取到的数据长度；
- poll_result：指明读取是否成功；

4. 使用方法

这里我们借助 ZC1 平台和一个 RS485 型的温湿度变送器来说明节点的使用方法。按照下图所示分别给 ZC1 开发板和传感器供电，并连接好 RS485 的 A、B 两线。



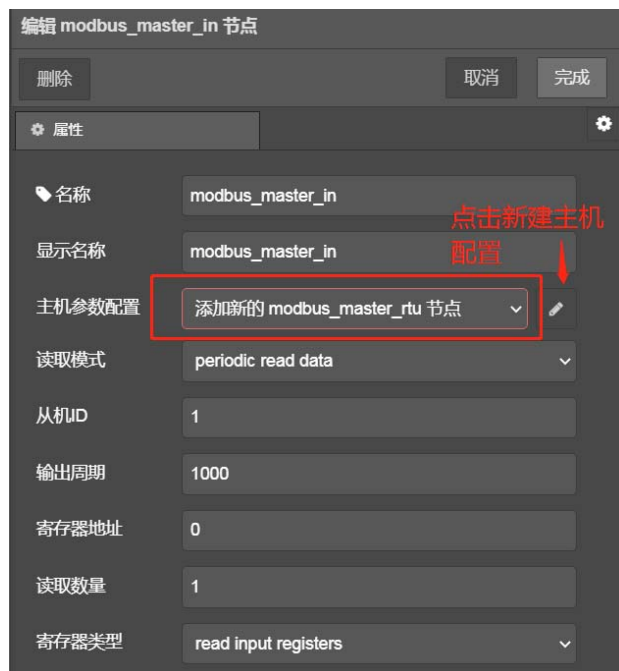
4.1 添加节点

添加 modbus_master_in、modbus_parse_in、fscript 以及 log 节点到画布上并连接节点。

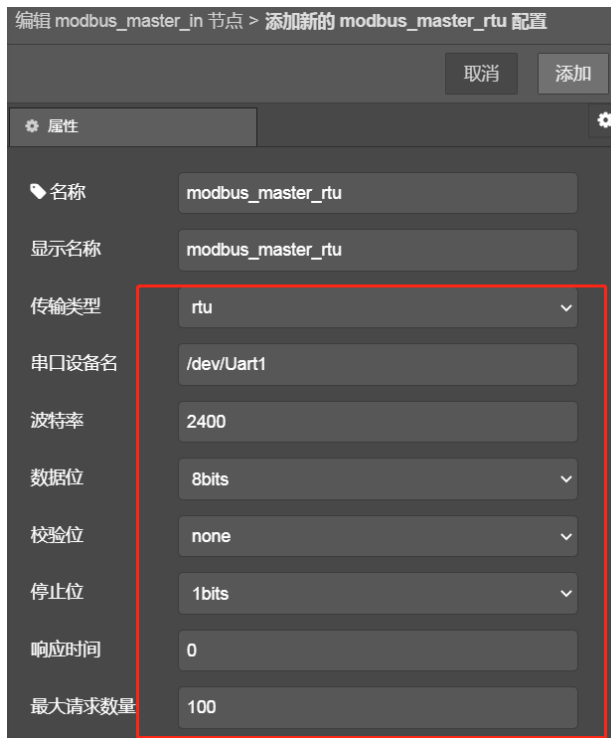


4.2 配置节点

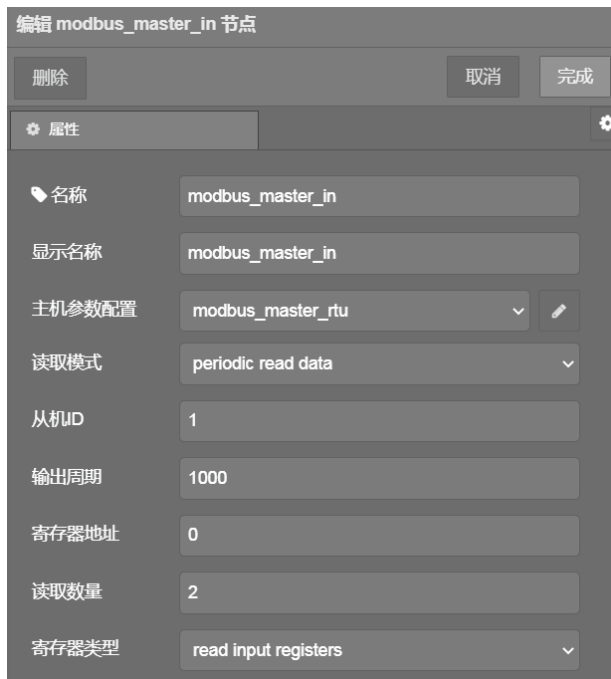
双击 modbus_master_in 节点打开属性配置面板。



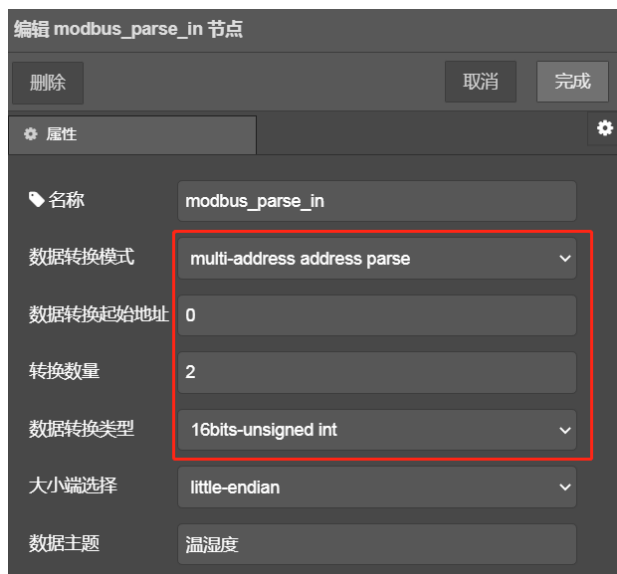
选择“添加新的 modbus_master_rtu 节点”，进入配置主机参数面板。



根据实际情况配置完相应的参数后，点击添加，回到 modbus_master_in 节点配置界面。



可以看到已经创建了一个新的主机参数配置，同时配置读取模式为周期读取，从机 ID、输出周期、寄存器地址等参数按照实际设备进行配置。然后双击 modbus_parse_in 节点打开属性配置面板，对 modbus_parse_in 进行转换规则设置。



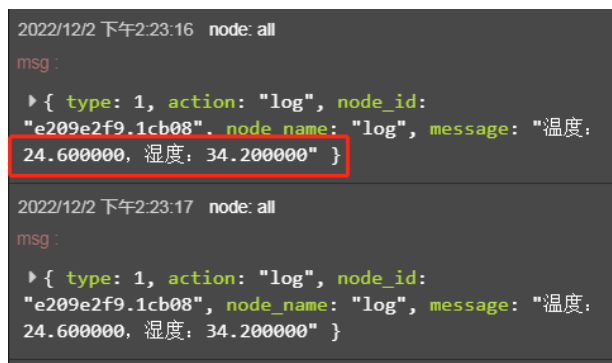
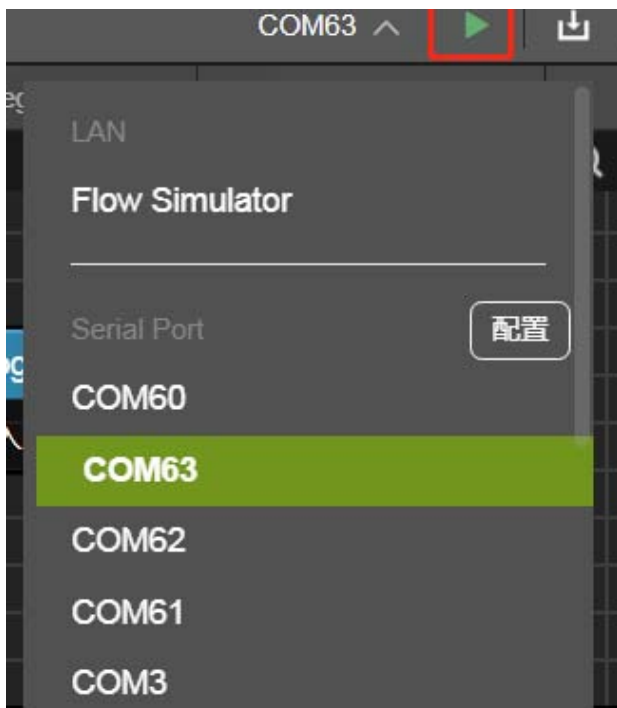
这里我们选择多地址转换模式，分别对温度和湿度两个寄存器进行转换，因此转换数量设置为2，转换类型设置为16位。

接下来设置 modbus_parse_in 节点的后级节点 fscript，主要是对后续的打印信息进行格式化。由于 modbus_parse_in 节点的输出 payload 是 array 型对象，这里我们可以通过 fscript 的内置方法 array_get 来获取数据。fscript 的内容如下：

```
a = msg.payload
msg.payload = "温度: "+array_get(a,0)/10 +", 湿度: "+array_get(a,1)/10
```

4.3 下载验证

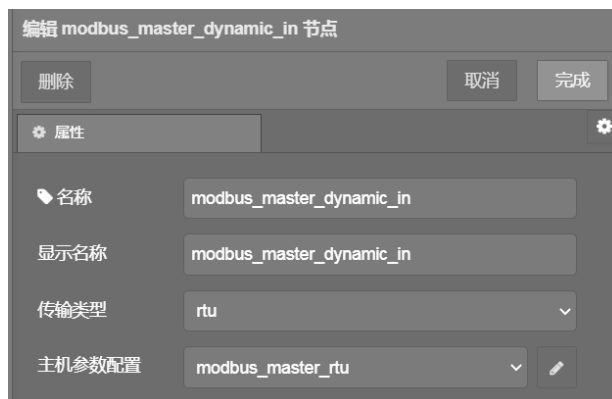
连接好硬件，通过下载接口下载流图进行验证。



通过调试面板可以看到读回的温湿度数据。

modbus_master_dynamic_in节点

modbus_master_dynamic_in 同样是 Modbus 的输入节点，与 modbus_master_in 的区别是：modbus_master_dynamic_in 节点是 filter 类型的节点，其根据前级节点输入的信息向从机设备读取数据，并输出给消费者节点。



1. 属性

- 传输类型：用于设置链路层的传输模式（rtu/tcp 可选）；
- 主机参数配置：输入节点依赖 modbus_master_rtu 节点，和使用 modbus_master_in 节点时一样，选择对应的配置节点即可；

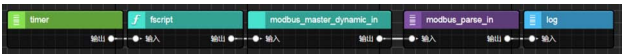
2. 输入

- slaveID：从机设备的 ID 地址；
- address：待读取的寄存器的起始地址；
- reg_num：待读取的寄存器的数量；
- reg_type：待读取寄存器的类型。

3. 输出

- slaveID：从机设备的 ID 地址；
- address：待读取的寄存器的起始地址；
- reg_num：读取到的寄存器的数量；
- payload：数据缓冲区，存储读取到的数据；
- payloadLength：读取到的数据长度；
- poll_result：指明读取是否成功。

4. 使用方式



modbus_master_dynamic_in 节点一般是由 fscript 节点指明需要读取的从机设备的信息，节点本身只需要创建并配置相应的主机即可。对于 modbus_master_dynamic_in 的输出缓冲区，我们同样使用 modbus_parse_in 节点进行处理。

在 fscript 中指明读取内容时，只需提供以下信息：

```
msg.slaveID = 1
msg.address = 0
msg.reg_num = 2
msg.reg_type = 4
```

modbus_master_out节点

modbus_master_out 节点是 Modbus 的输出节点，用于写线圈量和寄存器。

1. 属性



- 主机参数配置：输出节点依赖 modbus_master_rtu 节点，选择对应的配置节点即可；
- 从机 ID：待写入的从机设备 ID 地址；
- 错误重试次数：发送错误情况下的重试次数；
- 寄存器类型：待写入寄存器的类型。

2. 输入

- slaveID：从机设备的 ID 地址，如果输入中包含此参数，则忽略属性中的地址；
- address：待写入的寄存器的起始地址；
- reg_num：待写入的寄存器的数量；
- payload：写入缓冲区，存储待写入的数据，一般由 modbus_parse_out 输入；
- payloadLength：写入缓冲区的字节长度。

3. 输出

modbus_master_out 节点是 sink 类型的节点，一般没有输出。

4. 使用方式

modbus_master_out 节点通常由 modbus_parse_out 节点进行输入。使用时和 modbus_master_in 一样创建一个主机配置，指明待写入设备的 ID 地址以及寄存器类型即可。



其中，modbus_parse_out 节点的输入一般来自 fscript 节点，我们可以在 fscript 中借助 array 对象完成数据输入。比如我们需要将从机地址为 1 的设备的寄存器 0 设置为 2，我们只需在 fscript 中添加如下内容：

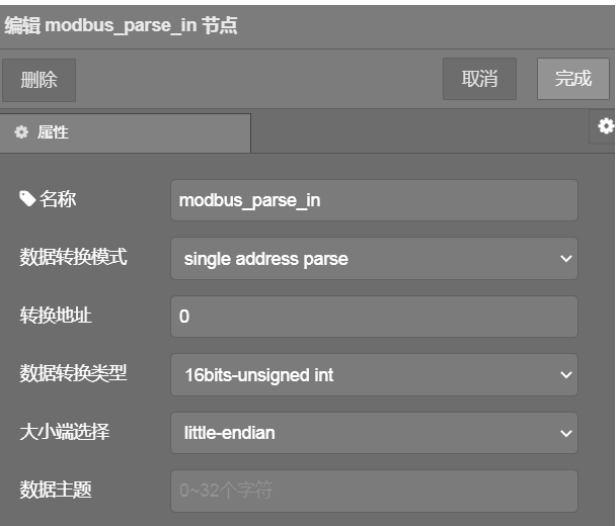
```
var a = array_create();
array_insert(a,0,2);
```

```
msg.payload = a;
msg.slaveID = 1;
msg.address = 0;
```

modbus_parse_in节点

Modbus 输入数据的转换节点，这个节点一般用于接收 Modbus 输入节点的原始数据，将其转换成单个 value 或者 array 对象，以便于后续处理。

1. 属性



- 数据转换模式：可以选择对输入中的单个地址的数据进行转换，也可以选择对指定数量的寄存器、或者对输入的所有地址进行转换；
- 转换地址：选择单个寄存器转换时，填写需要转换的寄存器地址；
- 数据转换起始地址：选择转换多个寄存器时，指明转换的起始地址；
- 转换数量：选择转换多个寄存器时，指明转换数量；
- 数据转换类型：选择按照哪种类型对数据进行转换；
- 大小端选择：大小端系统选择；
- 数据主题：数据地址的主题（可选）。

2. 输入

modbus_parse_in 的输入来自 modbus_master_in 或 modbus_master_dynamic_in 节点。

- slaveID：从机设备的 ID 地址；
- address：上一级节点读取的寄存器起始地址；
- reg_num：上一级节点读取到的寄存器数量；
- payload：数据缓冲区指针，存储读取到的数据；
- payloadLength：读取到的字节数据长度；
- poll_result：上一级节点读取数据是否成功。

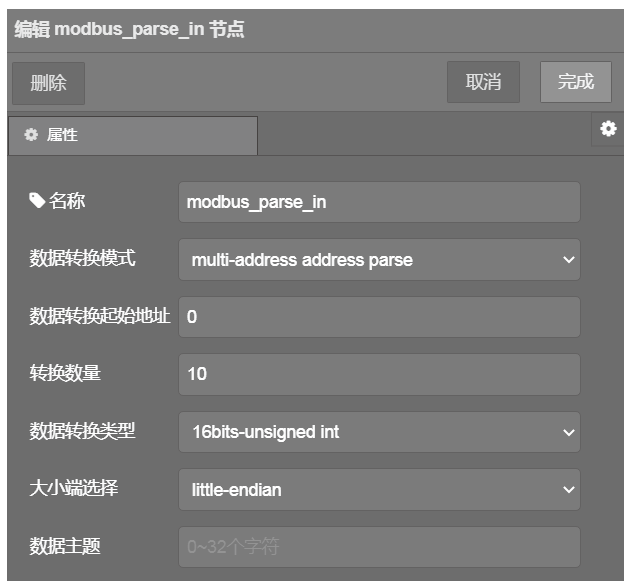
3. 输出

- slaveID：从机设备 ID 地址；
- address：转换单个寄存器时，指示数据的寄存器地址；在转换多个寄存器模式下，指示起始的寄存器的地址；

- payloadType: 指示缓冲区的类型是 value 还是 array;
- payload: 转换后的数据缓冲区, 在转换单个寄存器时, payload 的类型是 value, 可直接读取; 当转换多个寄存器时, payload 是个 array 对象, 可以使用 rbuffer 进行读取;
- topic: 属性中设定的数据主题;
- poll_result: 上一级节点读取数据是否成功。

4. 使用方法

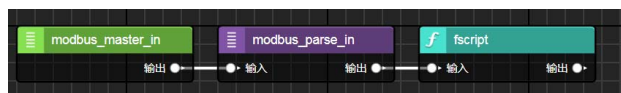
modbus_parse_in 节点使用时需要指定转换的模式、待转换的寄存器地址和数量、待转换寄存器的数据类型以及系统的大小端模式。



作为 filter 类型节点, modbus_parse_in 通常是从 modbus_master_in 节点获取输入。其输出有两种类型, 在转换单个寄存器时, payload 的类型是 value, 可直接读取。



当转换多个寄存器时, payload 是个 array 对象, 可以在 fscript 中通过 array 对象的方法进行处理。



其中, array 对象的使用方式如下:

```
a = msg.payload
```

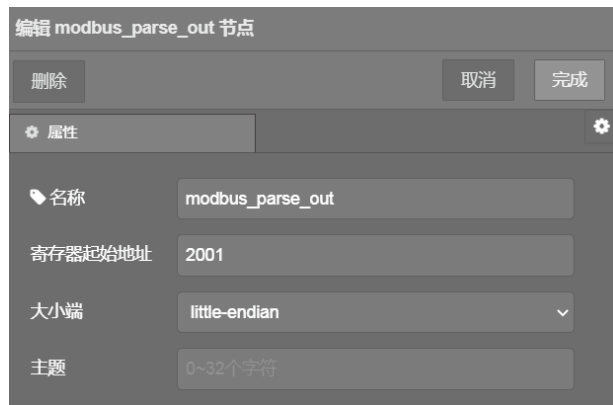
```
print(" 湿度 " + array_get(a, 0))
```

```
print(" 温度 " + array_get(a, 1))
```

modbus_parse_out节点

modbus_parse_out 节点通常用于将 fscript 节点输出的 value 或 array 对象转换成 Modbus 标准的十六进制格式, 后续可以将相应的数据指针传入 Modbus 的输出节点 modbus_master_out 进行使用。

1. 属性



- 寄存器起始地址: 指明需要 modbus_master_out 写入的寄存器的起始地址, 如果节点的输入不包含 msg.address, 则使用此属性指向的地址;
- 大小端: 系统大小端模式选择;
- 主题: 如果设定了主题, 则节点只处理属于自己的主题消息。

2. 输入

- slaveID: 从机设备 ID 地址;
- address: 待写入的寄存器起始地址;
- payload: 待写入的值, 可以是 value 类型, 也可以是 array 对象;
- topic: 输入的数据主题, 如果本节点设置了主题属性, 则只有输入的主题与节点的主题属性匹配, 节点才会对下一节点输出;

3. 输出

- slaveID: 从机设备 ID 地址;
- address: 待写入的寄存器的起始地址;
- reg_num: 待写入的寄存器的数量;
- payload: 转换后的 Modbus 标准十六进制数组缓冲区指针;
- payloadLength: 数组缓冲区的字节长度。

4. 使用方法

modbus_parse_out 节点的前级通常是 fscript, 使用时在 fscript 中指明待写入的从机设备 ID 地址、并传递构建好的 array 对象即可。其后级节点一般是 modbus_master_out 节点, 通过 modbus_master_out 节点将数据写入目标设备。



其中, 在 fscript 中构建 array 对象的方法如下:

```
a = array_create();  
array_insert(a, 0, u16(1))  
array_insert(a, 1, u16(2))  
array_insert(a, 2, u16(3))
```

```
output.payload = a;  
output.slaveID = 10;  
output.address = 0;
```



MPC-ZCI卡片式工控主板
MPC-ZCI

点击购买

【EsDA协议转换】 Modbus转OPC UA协议

ZLG 致远电子 2023-02-08 11:51:18

OT/IT 系统中的设备使用了各种各样的通讯协议，如何让这些设备实现信息共享，是当今工业互联网的一大难点。本文将讲解如何使用 EsDA 工具图形化低代码地开发一个 Modbus 与 OPC UA 的“转换器”。

简介

随着工业互联网和产业数字化的加速推进，工业企业的 IT(Information Technology) 系统与 OT (Operation Technology) 系统也在加速融合。但 OT/IT 系统中的设备使用了各种不同的通讯协议，如何将它们互联起来，让所有的信息能够实现共享，是当前工业互联网的一大难点。

如图 1，在实际项目中可能会存在这么一种场景，需要采集支持各种标准或非标准协议的工业设备的数据并将其发送到支持 OPC UA/OPC DA 协议的 SCADA(Supervisory Control And Data Acquisition 数据采集与监视控制) 系统中。但由于 SCADA 系统支持的是 OPC UA/OPC DA 协议，这些设备不能直接接入到系统，这个时候就需要一个“转换器”。

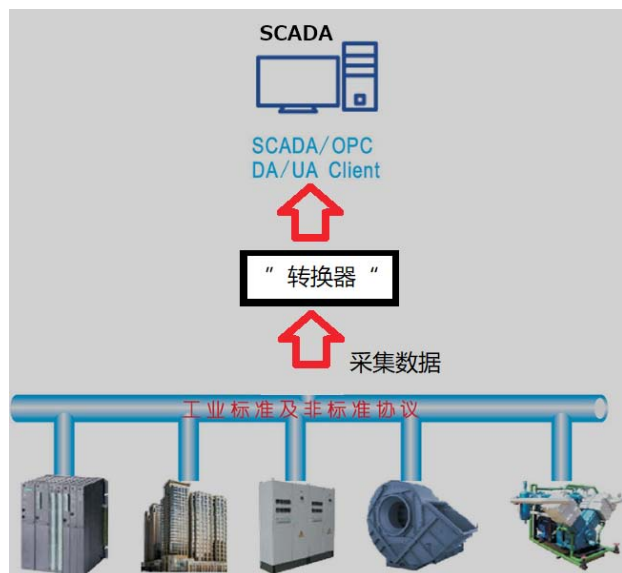


图1

若需要采集数据的设备仅支持 Modbus 协议，此时就需要一个 Modbus 转 OPC UA /OPC DA 的“转换器”，如下图 2, 本文以实现 Modbus 转 OPC UA /OPC DA 协议“转换器”为目标，展开讲解，介绍如何通过 EsDA 家族中的 AFlow Designer 和 EPC6450-AWI 开发板套件，进行图形化低代码应用开发，快速完成一个“转换器”。

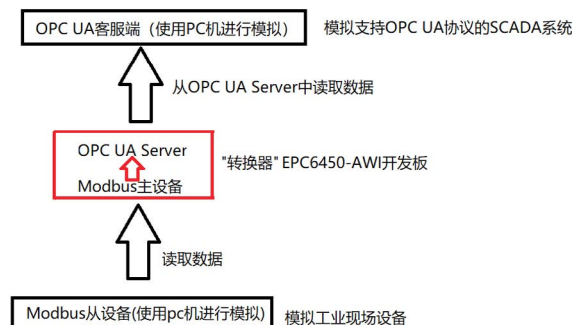


图2

1. 本次实验主要步骤

- 在 PC 机上配置 Modbus 模拟从设备；
- 业务开发, 使用 EsDA 家族中的 AFlow Designer, 进行图形化开发完成“转换器”业务；
- 在 PC 机上打开 OPC UA Client 模拟软件，读取 OPC UA 服务器数据。

2. EsDA方案与传统方案对比

“转换器”的传统的解决方案是采购一个协议转换网关，但与传统方案相比，使用 EsDA 家族中的 AFlow Designer(嵌入式应用开发利器——【EsDA】AFlow) 与配套开发板方案主要有以下优点：

- 灵活性高支持二次开发，传统网关一般功能比较单一，使用 EsDA 方案，可以使用 AFlow Designer 对硬件进行可视化配置，且可根据实际需求自由更改应用逻辑（且大多数情况无需重新编译和烧写固件），可同时做多种协议的“转换器”，尽可能降低成本，甚至可以做协议转换之外的其它业务；
- 二次开发简单，使用 AFlow Designer 简单“拖拽、连线”即可完成二次开发；
- 可复用性强、模块化程度高，AFlow 将常用模块封装为“节点”，高度模块化，并支持自定义节点；
- 硬件成本比传统方案低。

零基础EsDA入门

若是刚开始接触 EsDA，可先阅读 EsDA MPC-ZC1 系列文章，从零开始搭建环境和掌握基本开发流程，已有基础的可以跳过：

- EsDA MPC-ZC1 入门（一）—— 软件安装
- EsDA MPC-ZC1 入门（二）—— LED 控制
- EsDA MPC-ZC1 应用—— 串口服务器（一）
- EsDA MPC-ZC1 应用—— 串口服务器（二）

1. 实验准备工作

1.1 PC 机软件安装

- 下载并安装 OPC UA Demo Client (OPC UA 客户端模拟工具)
<https://pan.baidu.com/s/1wiIv8eKsX7ypv4nnYyB7w?pwd=pit2>
- 下载并安装 ModbusSlave (Modbus 从机模拟工具)
<https://pan.baidu.com/s/1JWDGgQbGPpDxc4oHNwg?pwd=cyr2>

1.2 硬件准备工作

如图 3, 准备一块 EPC6450-AWI 开发板, 并按照下图所示连接好硬件 (另一端接到 PC 机上)。

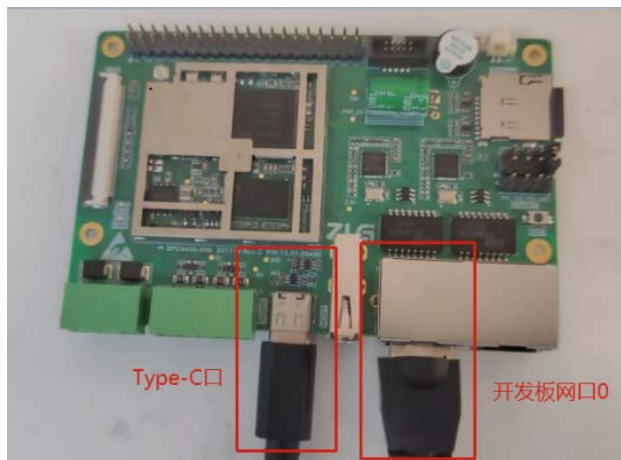


图3

1.3 配置 PC 机

注意: 需要将 PC 机与开发板设置为同一网段 (例如本实验选用 EPC6450-AWI 开发板的 ip 为 169.254.240.77 故需要将 PC 机的 ip 改为 169.254.240.xx), 否则实验可能会失败, 本文的 PC 机 ip 更改为 169.254.240.136。

1.3.1 配置模拟 Modbus 从设备

1.3.1.1 打开刚才安装的 Modbus Slave 软件, 如下图 4。

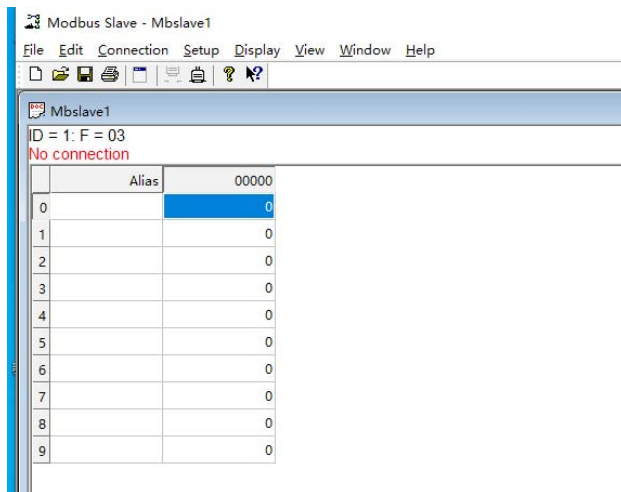


图4

1.3.1.2 依次点击 Connection->Connection Setup, 并根据图 5 配置从设备参数, 并点击 OK 保存设置。

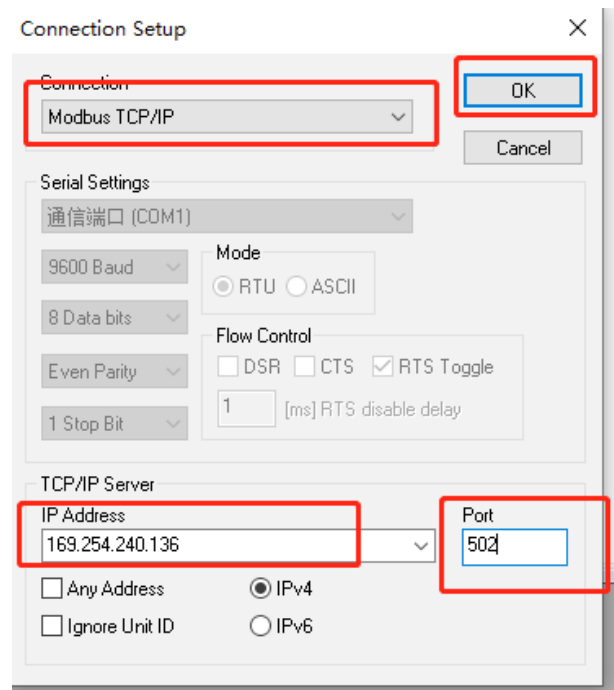


图5

1.3.1.3 按 F8 弹出 Slave Definition 界面, 并如图 6 所示进行配置, 点击 OK 保存。

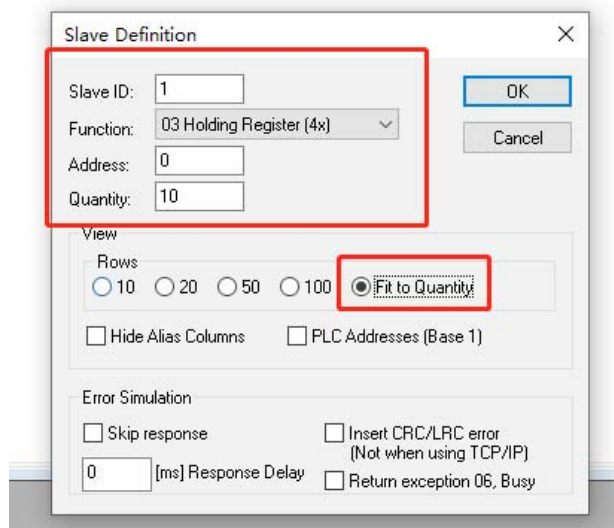


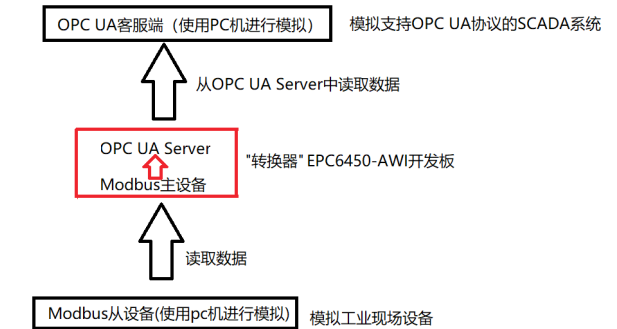
图6

1.3.1.4 双击红框位置，配置 Modbus 从设备对应寄存器的值，如图 7 所示。

	Alias	00000
0		0
1		18
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

图 7

2. EsDA业务开发

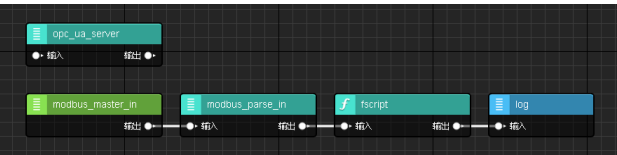


本次实验主要完成的是上图红框部分，即 EPC6450-AWI 作为一个 “转换器”，其业务主要分为两大部分：

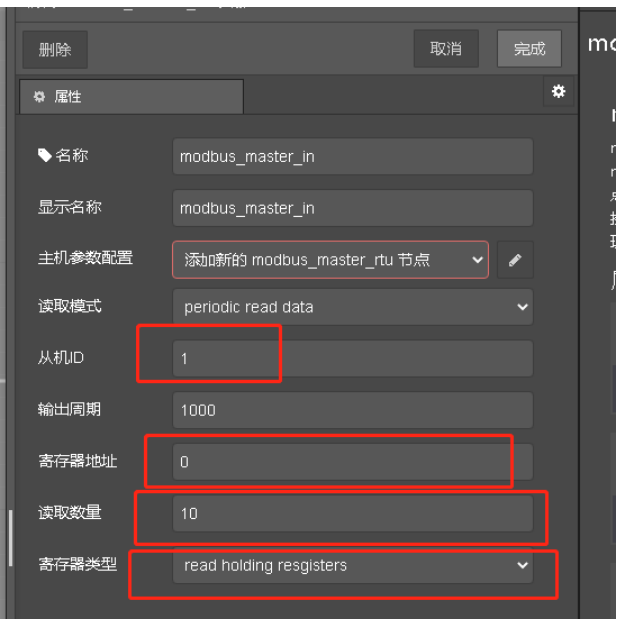
- 从 Modbus 设备读取数据；
- 启动 OPC UA 服务器，创建变量，并将上一步读取到的数据的值写入 (让 OPC UA 模拟客户端读取)。

2.1 应用流图绘制

2.1.1 依次将画布拖出 modbus_master_in、modbus_parse_in、fscript、log、opc_ua_server 节点，并依次连线。

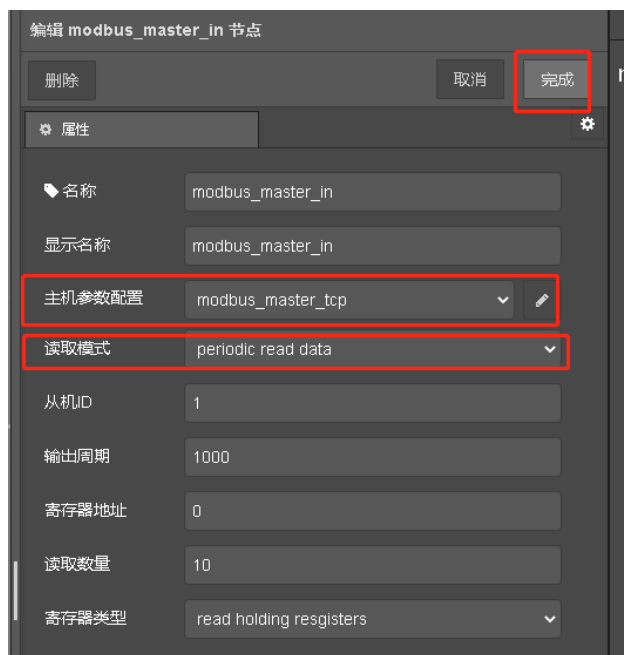


2.1.2 双击 modbus_master_in 节点，对应之前在 PC 机上 Modbus Slave 模拟软件的参数，进行配置。



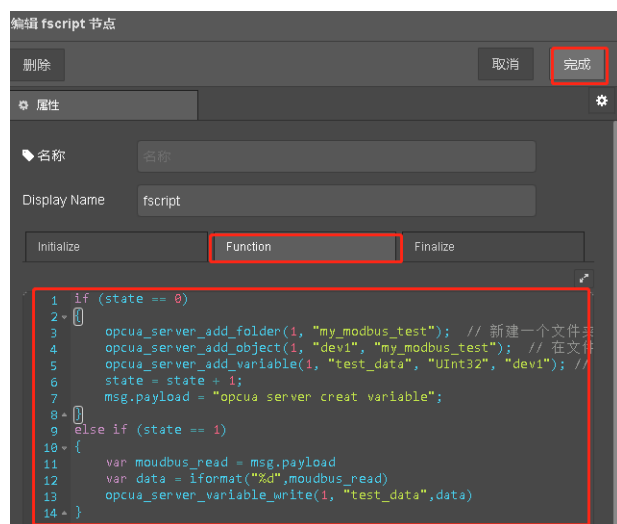
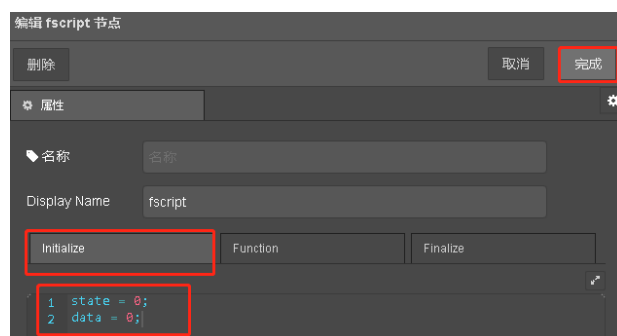
2.1.3 配置 modbus_master_in 节点连接参数与数据读取方式，并点击完成。





2.1.4 配置 modbus_parse_in 节点，配置 Modbus 数据解析方式，并点击完成。

2.1.5 在 fscript 节点编写脚本，配置 OPC UA 服务器初始参数，并将从 Modbus 设备读取到的数据写入 OPC UA 服务器。



// 脚本如下:

```
if (state == 0) {
//opcua_server: 新建一个文件夹:my_modbus_test
opcua_server_add_folder(1, "my_modbus_test");
//opcua_server: 在文件夹 zlg_gateway1 里新建一个对象 com1.m1
opcua_server_add_object(1, "dev1", "my_modbus_test");
//opcua_server: 在对象 dev1 里新建一个变量 test_data, 类型
UInt32, 只读
```

```
opcua_server_add_variable(1, "test_data", "UInt32", "dev1");
// 标记已经初始化过了
state = state + 1;
msg.payload = "opcua server creat variable";
}
else if (state == 1)
{
var moudbus_read = msg.payload
var data = iformat("%d",moudbus_read)
// 将读取到的数据写入 opcua_server 的 test_data 变量中
opcua_server_variable_write(1, "test_data",data)
}
```

至此, "转换器" 业务流程图已经开发完成。

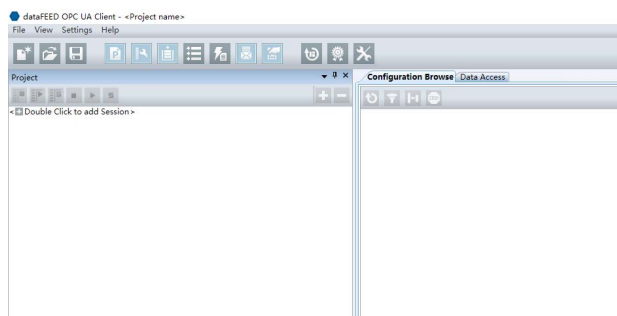
2.1.6 将开发板与 AWEFlow Designer 建立连接并将应用流图下载到开发板。



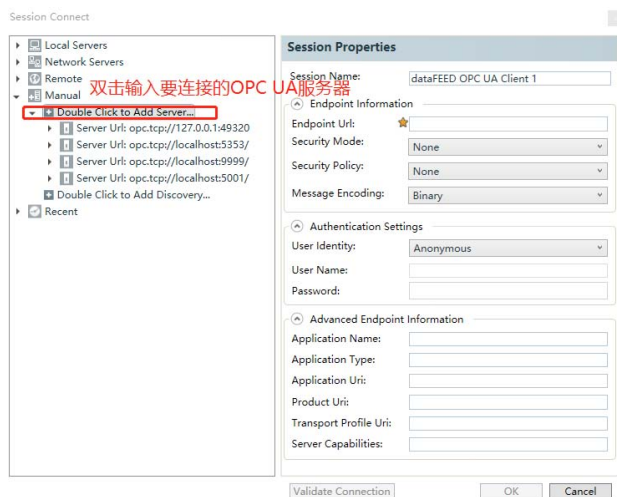
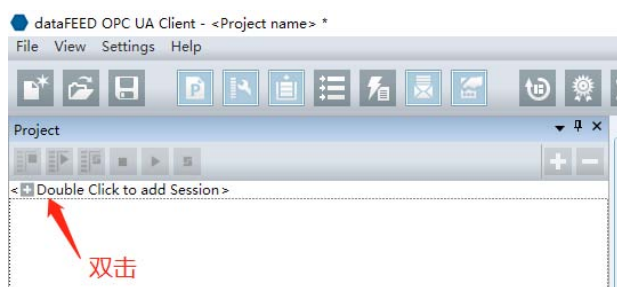
实验结果

配置模拟 OPC UA 客户端, 读取数据。

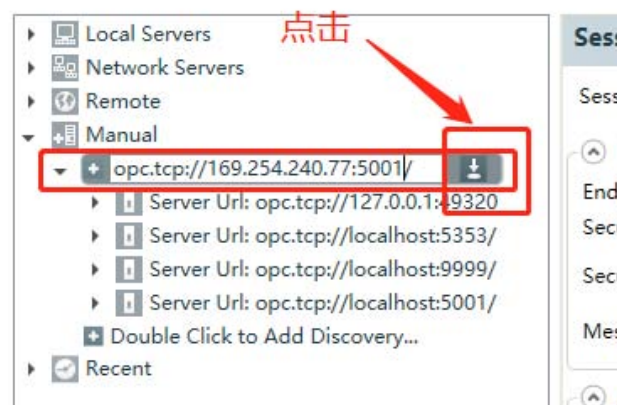
1. 打开刚才安装的 OPC UA Demo Client 软件。

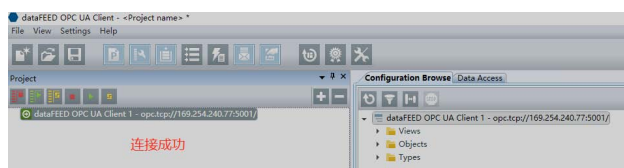
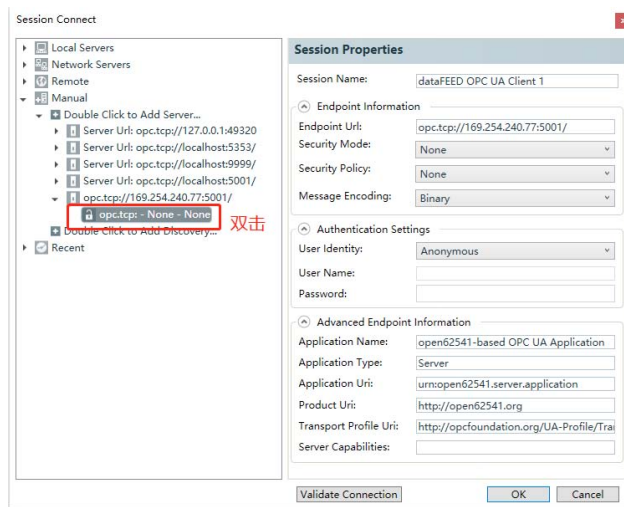


2. 双击打开对话配置菜单, 输入服务器地址, 并建立连接。

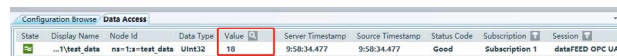
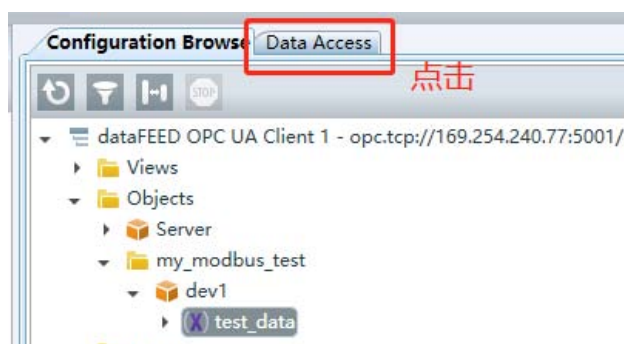
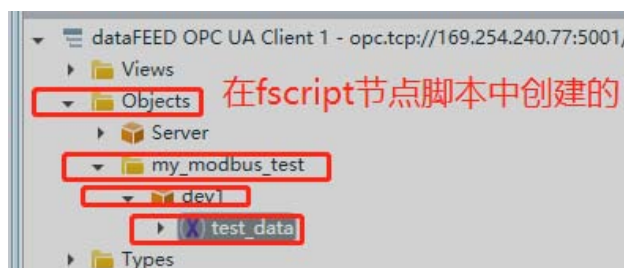


Session Connect





3. 查看转换器创建的文件夹与数据，数值为 18 正是之前所配置的 Modbus 从设备寄存器中的数值。



4. 修改 Modbus 模拟从设备的数值为 38，片刻后，OPC UA Client 模拟软件上的数值也变为 38，实验成功。

	Alias	00000
0		0
1		38
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0





工控板EPC6450-AWI
EPC6450-AWI

[点击购买](#)

【EsDA协议转换】 Modbus转CAN/CANFD协议

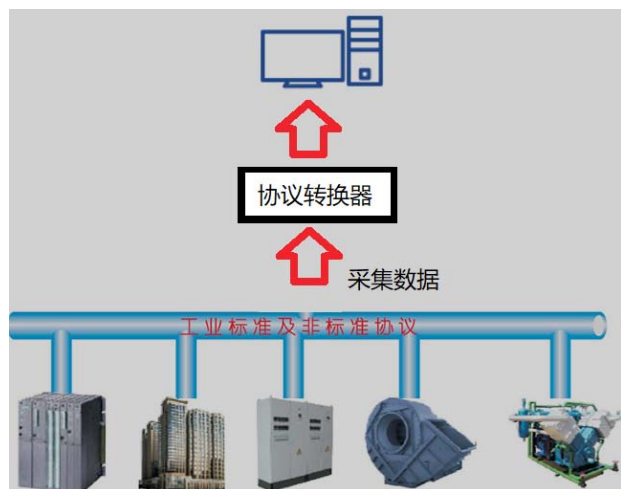
ZLG 致远电子 2023-03-29 11:32:18

实施 IIoT，协议转换在某些场合下必不可少。本文介绍如何通过 EsDA 开发套件，快速完成一个 Modbus 转 CAN 的协议转换器。

简介

工业物联网感知层的数据来源非常多样，通常来自各种多源异构设备和系统，因此如何从这些设备和系统中获取数据，是工业物联网面临的第一道门槛。

但实际上在工业现场，会采用不同厂商生产的设备，而这些设备也可能采用不同的工业协议。有些时候，由于有些设备甚至已经使用了若干年，出于系统稳定性和集成时间的考虑，更换设备并非上策。因此实施 IIoT，协议转换在某些场合下必不可少。



如何解决协议转换中的"痛点"

工业物联网感知层的数据来源非常多样，通常来自各种多源异构设备和系统，因此如何从这些设备和系统中获取数据，是工业物联网面临的第一道门槛。

但实际上在工业现场，会采用不同厂商生产的设备，而这些设备也可能采用不同的工业协议。有些时候，由于有些设备甚至已经使用了若干年，出于系统稳定性和集成时间的考虑，更换设备并非上策。因此实施 IIoT，协议转换在某些场合下必不可少。

1. 灵活性高支持二次开发，传统网关一般功能比较单一或难以实现复杂逻辑，使用 EsDA 方案，可以使用 AWWFlow Designer 对硬件进行可视化配置，且可根据实际需求自由更改应用逻辑（且大多数情况无需重新编译和烧写固件），可同时做多种协议的转换器，甚至可以做协议转换之外的业务；
2. 二次开发简单，使用 AWWFlow Designer 简单“拖拽、连线”即可完成二次开发；

3. 可复用性强、模块化程度高，AWFlow 将常用模块封装为“节点”，高度模块化，并支持自定义节点；
4. 硬件成本比传统方案低。

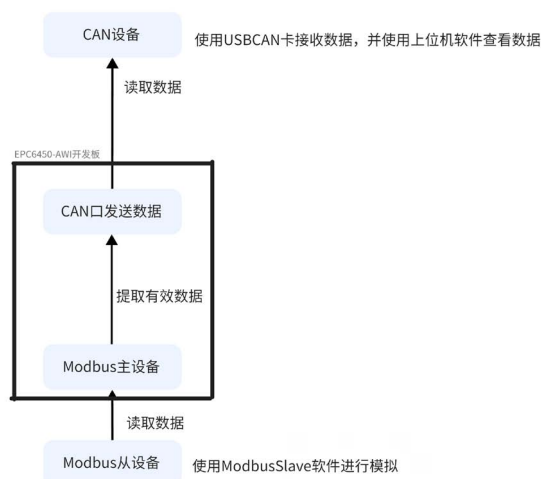
零基础EsDA入门

若是刚开始接触 EsDA，可先阅读 EsDA MPC-ZC1 系列文章，从零开始搭建环境和掌握基本开发流程，已有基础的可以跳过：

- EsDA MPC-ZC1 入门（一）—— 软件安装
- EsDA MPC-ZC1 入门（二）—— LED 控制
- EsDA MPC-ZC1 应用—— 串口服务器（一）
- EsDA MPC-ZC1 应用—— 串口服务器（二）

使用EsDA完成Modbus转CAN协议实验

在实际项目中，有时会遇到需要读取 Modbus 从设备的数据到不支持 Modbus 的设备的情况，本文以实现 Modbus 转 CAN 协议转换器为目标（即读取 Modbus 从设备数据并发送到指定 CAN 设备以完成转换），展开讲解，介绍如何通过 EsDA 家族中的 AWWFlow Designer 和 EPC6450-AWI 开发板套件，进行图形化低代码应用开发，快速完成一个协议转换器。



1. 本次实验主要步骤

准备一块 EPC6450-AWI 开发板与 USBCANFD-200U CANFD 接口卡，并接线；

在 PC 机上配置 Modbus 模拟从设备；

业务开发，使用 EsDA 家族中的 AWWFlow Designer，进行图形化开发完成转换器业务；

在 PC 机上打开 ZCANPRO 软件 (USBCANFD-200U CANFD 接口卡上位机软件)，读取来自协议转换器的数据。

1.1 PC 机软件安装

1.1.1 下载并安装 ModbusSlave (Modbus 从机模拟工具)

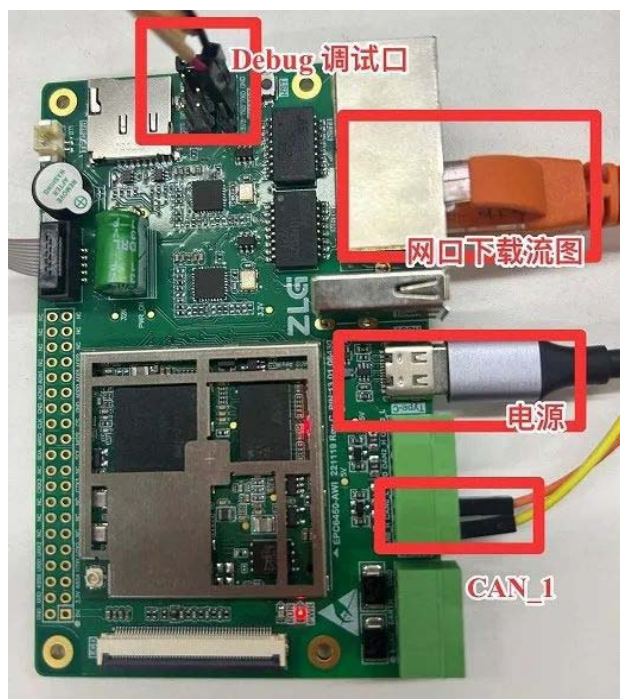
附：https://www.modbustools.com/modbus_slave.html

1.1.2 下载 USBCANFD-200U 接口卡驱动和上位机软件 (CAN-bus-ZCANPRO)，并安装。

附: <http://www.zlgcan.com/zlgcan-usbcanfd-200u/927/>

1.2 硬件准备工作

1.2.1 准备一块 EPC6450-AWI 开发板，并按照下图所示连接好硬件 (电源与网口接到 PC 机上, CAN_1 两根线接到 USBCANFD-200U 接口卡上, Debug 调试口可不接)。



1.2.2 将第一步的 CAN_1 线连接到 USBCANFD-200U CANFD 接口卡的 CAN0, USB 线接 PC。

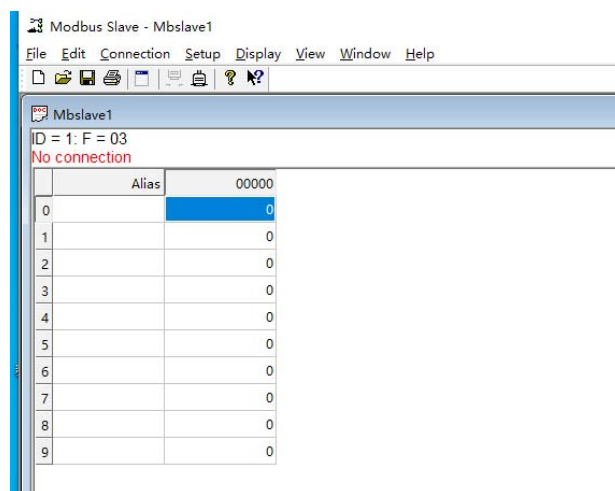


1.3 配置 PC 机

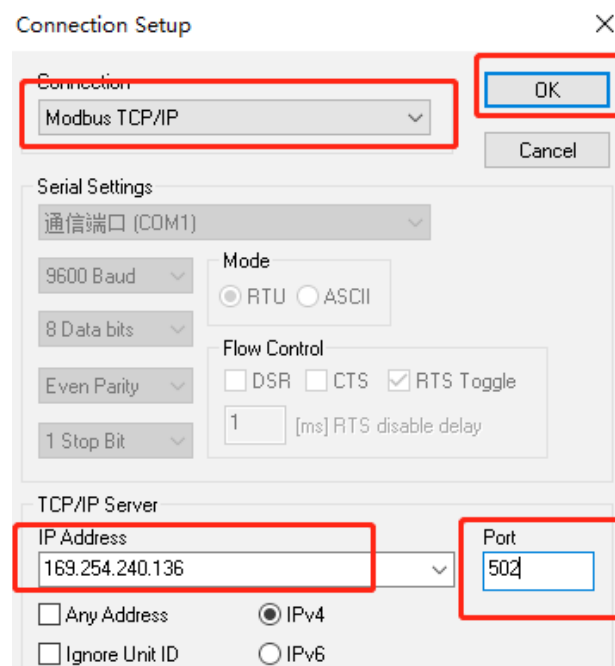
注意: 需要将 PC 机与开发板设置为同一网段 (例如本实验选用 EPC6450-AWI 开发板的 ip 为 169.254.240.77 故需要将 PC 机的 ip 改为 169.254.240.xx)，否则实验可能会失败, 本文的 PC 机 ip 更改为 169.254.240.136。

1.3.1 配置模拟 Modbus 从设备

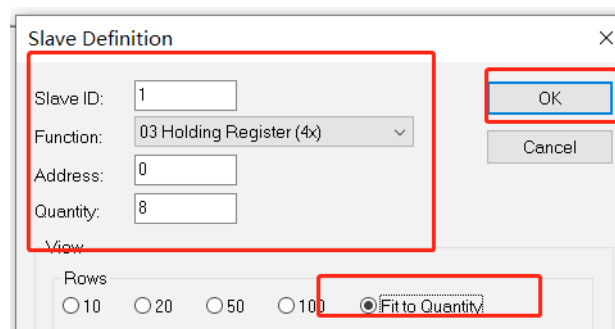
1.2.1.1 打开刚才安装的 Modbus Slave 软件。



1.2.1.2 依次点击 Connection->Connection Setup, 并根据图 5 配置从设备参数, 并点击 OK 保存设置。



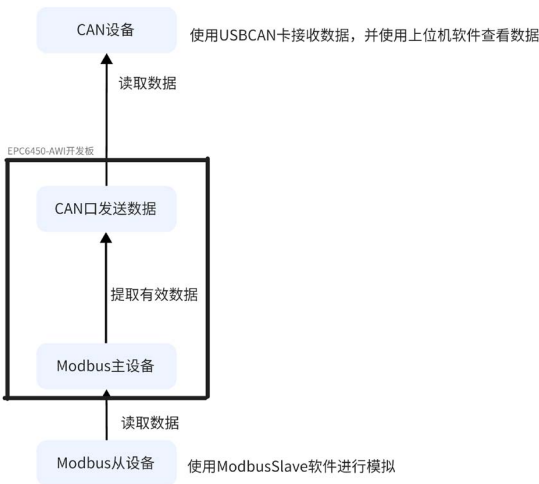
1.2.1.3 按 F8 弹出 Slave Definition 界面, 并如图 6 所示进行配置, 点击 OK 保存。



1.2.1.4 配置 Modbus 从设备对应寄存器的值如下图。

	Alias	00000
0		0
1		1
2		2
3		3
4		4
5		5
6		6
7		7

2. EsDA业务开发

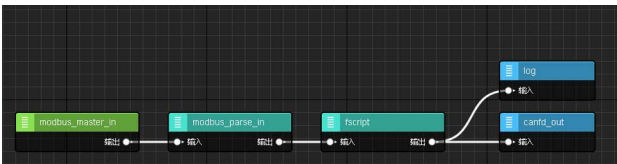


本次实验选用的开发板为 EPC6450-AWI，其作为一个协议转换器，其业务主要分为两大部分：

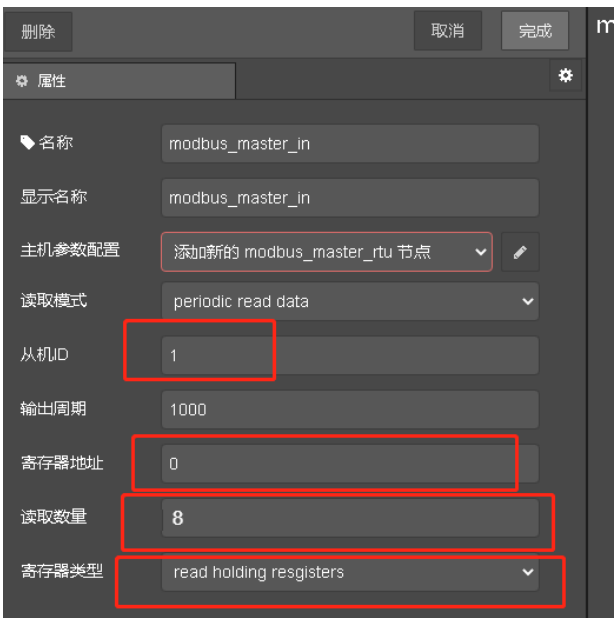
- 从 PC 机模拟的 Modbus 设备读取数据；
- 将读取到的数据通过 CAN 接口发送到 USBCANFD-200U CANFD 接口卡。

2.1 应用流图绘制

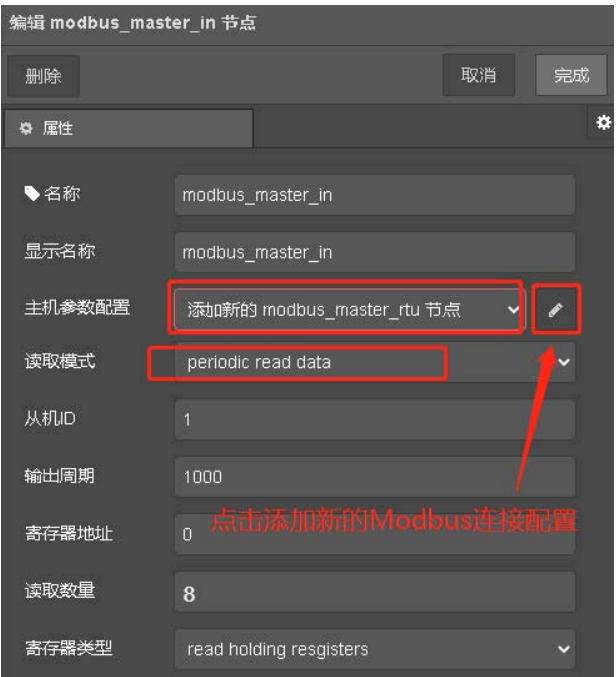
2.1.1 依次将画布拖出 modbus_master_in(Modbus 主设备读取从设备数据节点)、modbus_parse_in、fscript、log、canfd_out(CAN 发送数据)节点，并依次连线。

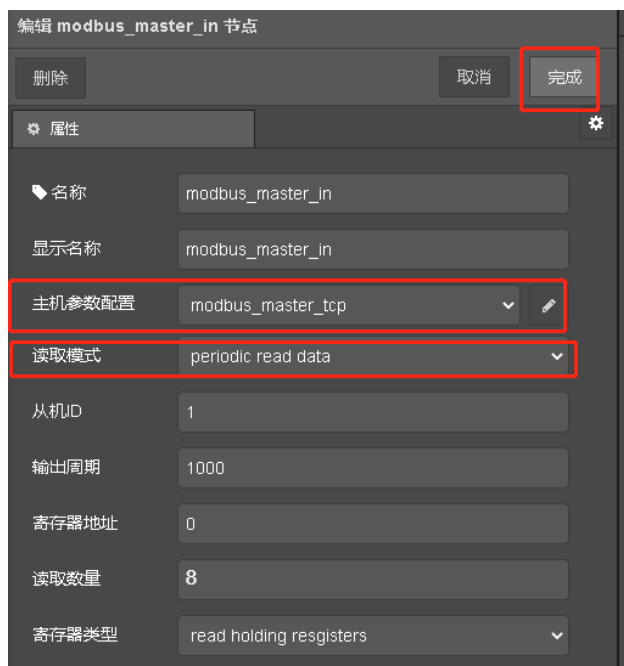
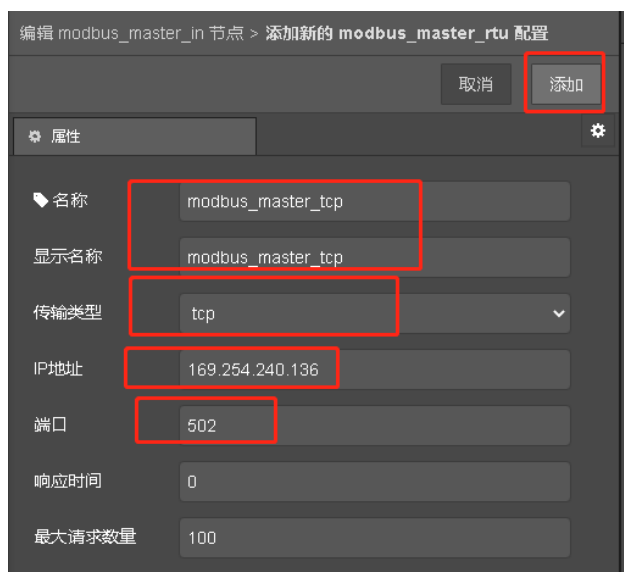


2.1.2 双击 modbus_master_in 节点，对应之前在 PC 机上 Modbus Slave 模拟软件的参数，进行配置。



2.1.3 配置 modbus_master_in 节点连接参数与数据读取方式，并点击完成。

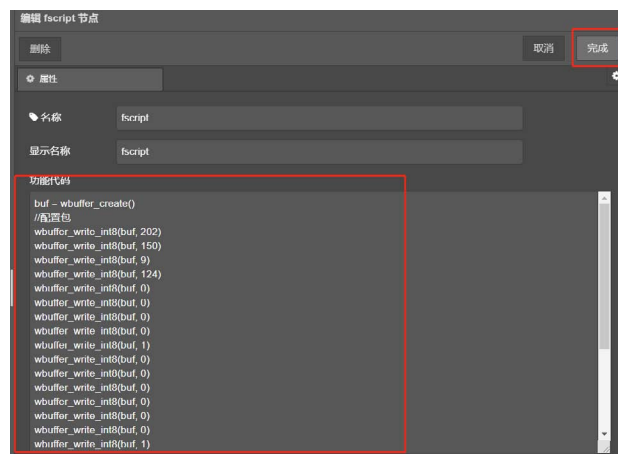




2.1.4 配置 modbus_parse_in 节点，配置 Modbus 数据解析方式，并点击完成。



2.1.5 在 fscript 节点编写脚本，将从 Modbus 设备读取到的数据写入 buff，并使用 CAN 发送节点发送 buff 中的数据。



2.1.4 配置 modbus_parse_in 节点，配置 Modbus 数据解析方式，并点击完成。

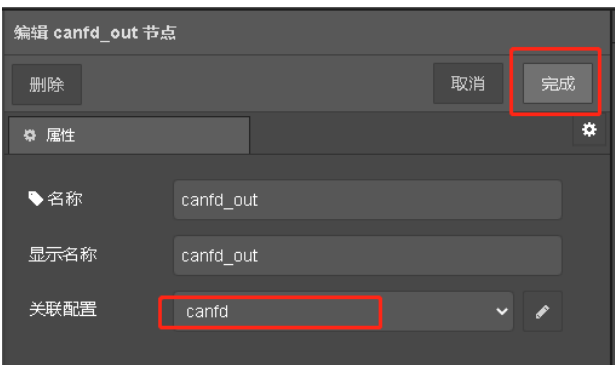
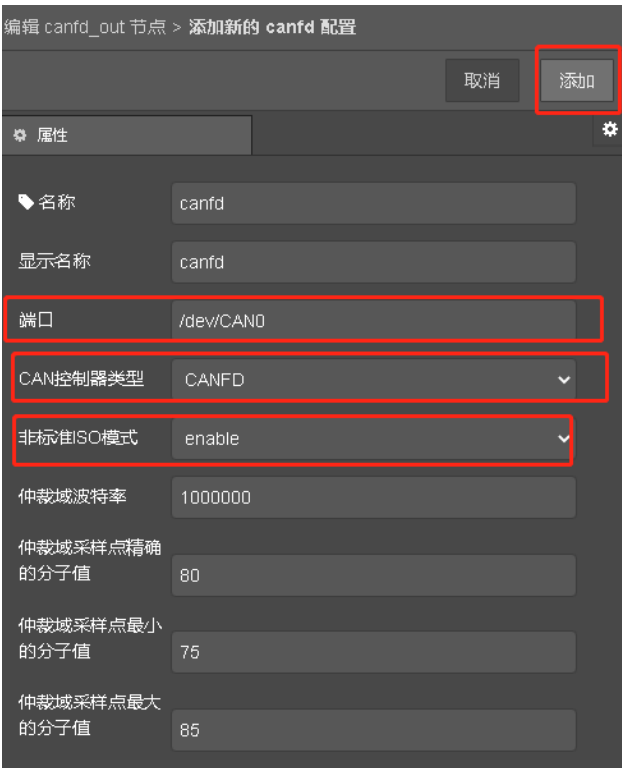
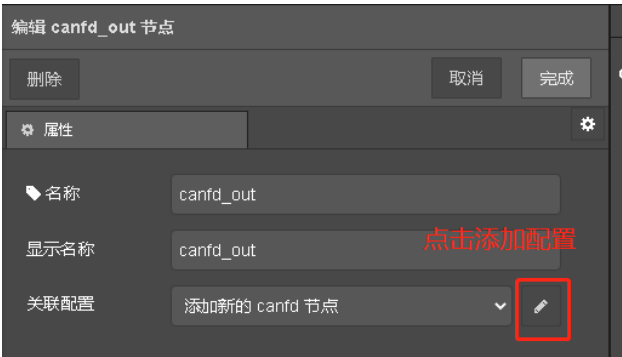
```
// 具体脚本如下：  
buf = wbuffer_create()  
//CAN 配置包 (前 16 个字节)  
wbuffer_write_int8(buf, 202)  
wbuffer_write_int8(buf, 150)  
wbuffer_write_int8(buf, 9)  
wbuffer_write_int8(buf, 124)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 1)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 0)  
wbuffer_write_int8(buf, 1)
```



```
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 0)
wbuffer_write_int8(buf, 1)

// 从 Modbus 从设备读取到的数据包 (八字节)
data=msg.payload
for(var i=0;i<data.size,i++)
{
    // 将数据写入 CAN 发送 buff
    wbuffer_write_int8((buf, array_get(data,i))
}
msg.payload = wbuffer_get_data(buf)
//3 * 8 = 24 个字节
msg.payloadLength = 24
```

2.1.6 双击 CAN 节点，进行配置。



至此，"转换器" 业务流程图已经开发完成。
2.1.7 将开发板与 AFlow Designer 建立连接并将应用流图下载到开发板。



3. 实验现象
打开 USBCANFD-200U 上位机软件（CAN-bus-ZCANPRO），选择打开设备（本实验所用的是 USB-CAN FD-200U），即可查看 CAN 卡读取的数据，与之前我们配置的 Modbus Slave 软件的 Modbus 从设备的数据一致，实验成功。





工控板EPC6450-AWI
EPC6450-AWI

[点击购买](#)



USBCANFD系列CANFD接口卡
USBCANFD-200U

[点击购买](#)

技术平台 ▼

【解决方案】 如何通过IoT云对设备数据进行自定义统计？

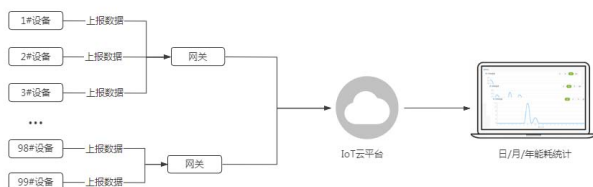
原创 研发部 ZLG 致远电子 2023-03-02 11:30:13

从设备→网关→云服务器，可以实现将数据采集到物联网云平台，但业务需求各式各样，有时需要不同的统计数据，那么，如何做好物联网数据自定义统计呢？

物联网数据自定义统计场景

物联网数据主要是指传感器或设备上报到云平台的数据，比如：电量、电压、电流、温度等动态数据。随着设备数量和时间增加，云平台会采集到大量的物联网数据。然而，在实际应用中我们发现，单纯采集到的数据，有时不能产生实际的业务价值，需要不同的统计才能满足业务需求。

例如，某工业园区对用电侧的设备数据监测，平台采集每台设备的电量数据，设备上报数据频率为 30 分钟，但单台用能设备数据无法直接反映能源消耗情况，需要从日、月、年三个时间维度，对不同区域用能设备的能耗进行统计。



数据自定义统计功能

ZWS 云平台是 ZLG 致远电子研发的通用物联网云平台，支持数据自定义统计功能。用户可以根据不同的业务需求，自定义编写统计算法，并创建不同的实例，来完成业务数据的不同统计规则。

1. 统计算法

指对设备数据的运算方式，可以自定义算法。比如，定义一个算法用来统计分组设备的数据在一段时间的总和。



2. 统计实例

指根据算法创建出具体某个设备数据的实例进行统计。比如，可以创建电量实例用于统计能耗，也可以创建出水量实例用于统计用水量。



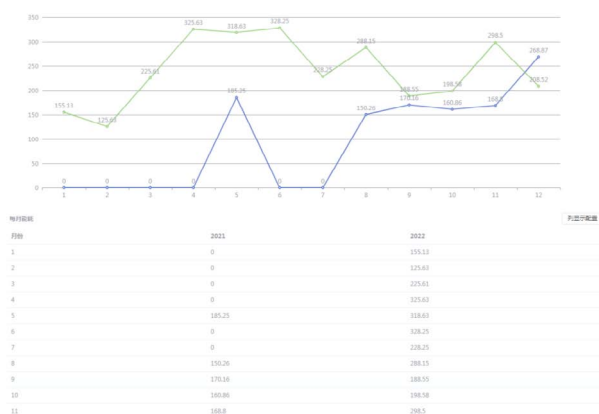
3. 统计方式

支持立即统计和定时统计两类不同实例。立即统计是指对历史数据进行立即统计，比如，对近 1 月的数据累加；定时统计是指设置统计周期进行统计，比如，定时每日 / 每月统计。



4. 统计结果

支持以列表、图表的形式展示统计结果，也可以用不同格式下载统计结果。





工业级CAT.1智能网联终端
CAT.1系列

[点击购买](#)

【解决方案】 设备如何上报数据到ZWS云平台？

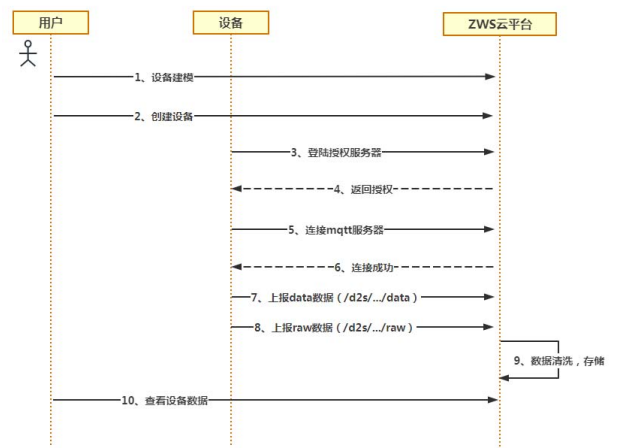
原创 研发部 ZLG 致远电子 2023-03-10 11:32:53

在物联网中，监测设备数据已是一个基础必备功能，可那些不同种类的物联设备究竟是如何上报数据到云平台呢？本文将介绍设备上报数据到 ZWS 云的整个流程，让您对设备上报数据有一个清晰的了解。

介绍

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

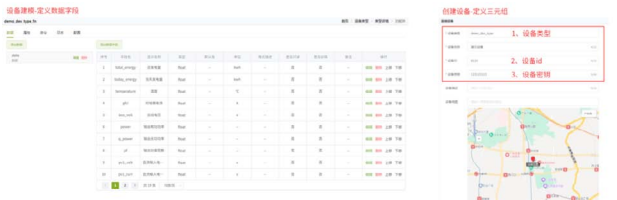
不同种类的设备采集的数据点不同，比如：逆变器采集电量，水表采集水量。若设备开发者要上报设备数据到云端，首先，为了区分种类，要在云端定义设备的数据模型，也就是对设备进行建模。其次，创建设备，定义设备三元组。最后，设备连接云服务器，通过 MQTT 将数据上报到云，云端对数据进行清洗，用户就能在云端查看数据了。流程如下所示：



设备建模&创建设备

设备建模是在云端定义一类设备的模型 schema，比如：设备的数据字段电量、电压、电流等，方便后续云端清洗存储。

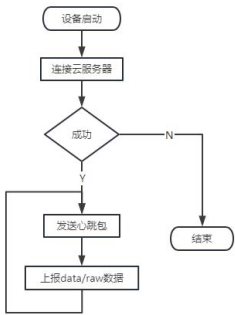
创建设备是在云端添加设备，定义设备的三元组：设备类型、设备 ID、设备密钥。（设备需要使用三元组连接云服务器）。



设备上报数据

设备固件要集成“ZWS_SDK”，SDK 里面提供了设备连接云服务关于网络层的封装同时也提供了一些 demo，其中 device_example 这个 demo 就是演示设备怎么连接云然后上报数据的。用户可以直接通过 make 指令编译 SDK 里的 device_expl.c，编译成功后，把设备的三元组传给运行程序，进行数据上报即可。

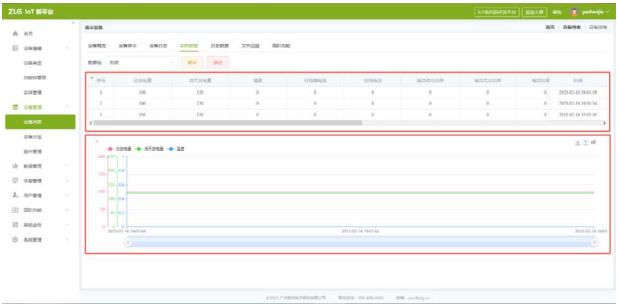
linux 系统，在 SDK 所在目录执行以下命令：
./bin/device_example demo_dev_type d123 123123123
流程图如下：



- 说明：云端支持的数据上报方式有两种
- 1.data 数据：以键值对形式上传，例如：“\0key1\0value1\0key2\0value2\0”；
- 2.raw 数据：使用 raw 主题上报，云端需按照设备的协议，用解析脚本或者解析配置进行数据解析后，再存储到服务器。

云端查看数据

返回到 ZWS 云平台，就可以看到设备上报的数据了。





工业级CAT.1智能网联终端
CAT.1系列

点击购买

技术平台 ▼

【产品应用】 如何通过IoT云平台对CAN数据进行实时DBC解析？

原创 研发部 ZLG 致远电子 2023-03-20 11:30:02

汽车路试中，遇到各种类型的故障问题，常规需要对 CAN 原始报文数据进行采集、下载、特定软件进行分析，一连串繁琐的步骤。如今，通过 CAN 智慧云平台，可以一步完成实时 DBC 解析，快速定位问题。

背景

DBC 文件是用来描述 CAN 总线通讯信息数据的一种文件，其中包含了 CAN 总线中协议数据及其所代表的具体意义。在汽车路试中，对于车辆错误故障信息，需对 CAN 报文数据进行实时 DBC 解析，定位问题点。

云端解析DBC数据

CAN 智慧云是基于 ZWS 云平台衍生而出的行业应用系统，适配致远电子推出的 CANDTU 系列产品，服务于汽车路试场景中，致力于解决汽车路试应用的痛点。

在 CAN 智慧云平台中进行实时 DBC 解析，只需以下三步：



1. 上报CAN原始数据到云

通过 CANDTU 系列设备，上报 CAN 原始数据到 CAN 智慧云。

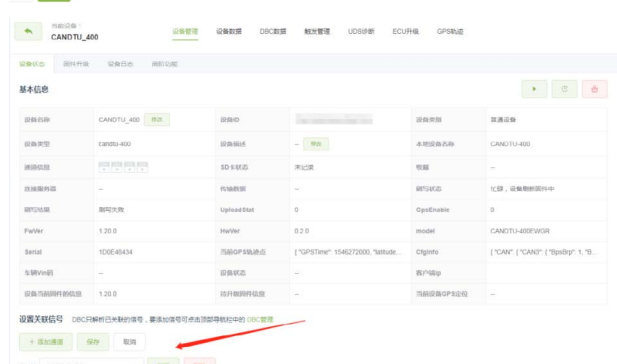
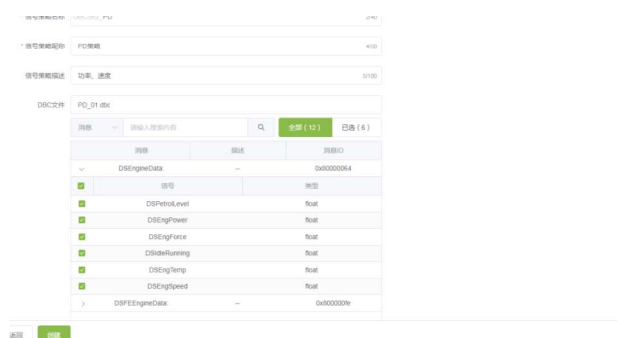
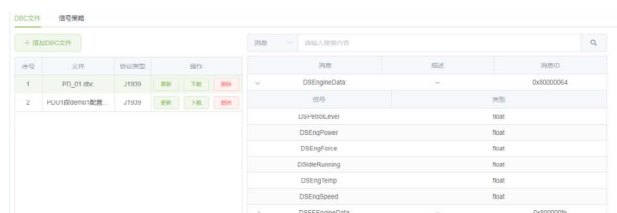
序号	报文ID	报文标志	帧格式	帧类型	报文通道	报文数据长度	报文数据	时间	设备时间
1	0a64	64	数据帧	R 接收	3	8	0x 01 00 19 01 00 00 65 00	2023-02-23 14:34:28.983	2023-02-23 22:34:26.525
2	0a64	64	数据帧	R 接收	3	8	0x 00 00 19 00 00 00 00 00	2023-02-23 14:34:27.959	2023-02-23 22:34:25.905
3	0a64	64	数据帧	R 接收	3	8	0x 01 00 19 01 00 00 65 00	2023-02-23 14:34:26.030	2023-02-23 22:34:24.484
4	0a64	64	数据帧	R 接收	3	8	0x 00 00 10 00 00 00 F4 01	2023-02-23 14:34:25.010	2023-02-23 22:34:23.453
5	0a64	64	数据帧	R 接收	3	8	0x 0A 00 1C 0A 00 00 C8 00	2023-02-23 14:34:24.491	2023-02-23 22:34:22.443
6	0a64	64	数据帧	R 接收	5	8	0x 0F 00 30 0F 00 00 DC 05	2023-02-23 14:34:19.464	2023-02-23 22:34:17.425
7	0a64	64	数据帧	R 接收	3	8	0x 12 00 22 12 00 00 4A 07	2023-02-23 14:34:22.441	2023-02-23 22:34:20.402
8	0a64	64	数据帧	R 接收	3	8	0x 14 00 23 14 00 00 D0 07	2023-02-23 14:34:21.416	2023-02-23 22:34:19.381
9	0a64	64	数据帧	R 接收	3	8	0x 12 00 22 12 00 00 4A 07	2023-02-23 14:34:20.396	2023-02-23 22:34:18.360
10	0a64	64	数据帧	R 接收	3	8	0x 0F 00 30 0F 00 00 DC 05	2023-02-23 14:34:19.378	2023-02-23 22:34:17.340

2. 配置DBC信号策略

首先，上传 DBC 文件。

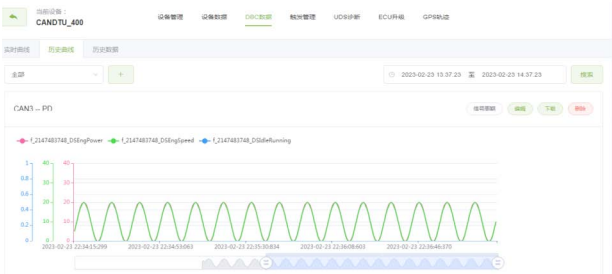
其次，设置信号策略。（解析 DBC 数据时，关注的信号值都有差异，信号策略就是通过选定 DBC 文件的信号值，生成相应的策略）

最后，通道关联信号策略。每个通道号都可关联不同的策略，可在“设备管理—设备状态—设置关联信号”中进行关联，关联成功后云端会将该通道的数据按策略进行解析。

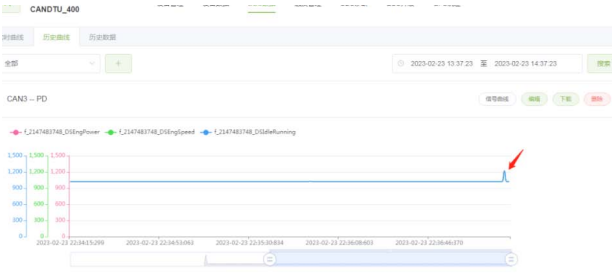


3. 可视化查看数据

在 DBC 数据 - 历史曲线中，可以查看分析一段时间内的信号值，分析信号值之间的关系与影响。点击右侧【下载】，对特定的时间数据进行下载保存。



可切换至信号周期，信号周期可查看设备是否存在掉帧的情况，一般信号会按周期发送数据，若发送正常，信号之间的时差应该是平稳的，但是若出现掉帧等异常情况，时间差曲线会出现波动。



关联产品

ZLGi

车载多通道CAN(FD)-bus
数据记录仪

CANFDDTU 系列

- 智能网联汽车总线
三合一测试利器
- 总线记录回放
可模拟现场数据
- 严格测试 稳定可靠



车载多通道CAN(FD)-bus
数据记录仪

点击购买

【产品应用】 IoT云平台中告警触发规则的应用

原创 研发部 ZLG 致远电子 2023-03-27 11:52:25

物联网中告警触发该怎么做？如何实现物联设备智能化预警？本文将介绍如何通过 ZWS 云平台完成告警触发需求。

应用场景

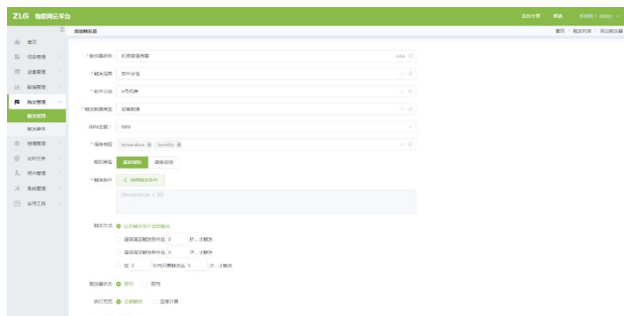
场景1

物联网终端采集设备会在运行时，定时上报数据，特定的数据或状态可能不会被及时发现。如某粮食仓库内部需要保持恒温恒湿的状态，遇到高温高湿情况要及时处理，那么，用户如何能及时获知特定数据状态？

场景2

设备安装在不同位置，告警阈值不同。如温度传感器在机房内和在冷藏库内的温度阈值会要求不同，那么，如何对不同设备分别设置不同的告警阈值条件？

针对以上场景，用户可以通过物联网云平台设置告警触发规则，灵活配置触发范围来解决。当云平台监控到设备上报的数据 / 状态满足用户通过规则设置的触发条件时，云平台会触发告警通知到用户，方便用户定位问题设备，及时解决问题。



告警触发

ZWS 云平台是 ZLG 致远电子研发的通用物联网云平台，支持告警触发功能。用户可以根据自己的业务场景和设备特性，在云平台创建自定义触发规则，设置触发范围和触发条件等。当监控数据满足条件触发后，支持以事件列表、邮件、短信等方式通知用户，帮助用户快速处理故障。

下面我们温度超限为例，看看在 ZWS 云平台中如何配置告警触发。

1. 创建触发规则

登录云平台，点击菜单“触发规则”，点击“添加触发器”，输入触发器名称。



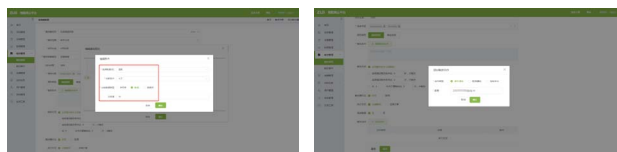
2. 设置触发范围

支持两种触发范围：“设备类型”和“软件分组”。按设备类型范围是指该设备类型下的设备满足条件就会触发告警。同理，软件分组是以某一分组设备作为触发范围进行告警。



3. 设置触发条件

可以自定义触发条件。编辑条件时，选择数据点“温度”，判断比较设置为“大于 30”。

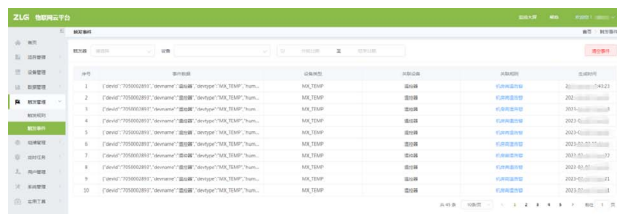


4. 设置触发动作

支持邮件、短信、控制命令三种触发动作，可以添加一个或多个触发动作。（若没有设置触发动作，默认是以事件列表在云端通知）。

5. 查看触发事件

成功创建触发规则后，可在触发事件列表内查看告警触发情况。



【技术分享】 基于M6Y2C平台实现LoRa远距离传输

原创 研发部 ZLG 致远电子 2023-03-03 11:31:00

想要使用 LoRa 远距离稳定可靠的传输数据？又不知道从何下手？
本文将向您介绍 ZSL421 LoRa 智能组网芯片，以及在 M6Y2C 使用 ZSL421 LoRa 模块进行数据通信的实操案例。

移植准备

1. 简介

LoRa 全称为远距离无线电（Long Range Radio）是一种线性调频扩频调制技术，具有传输距离远、低功耗、组网灵活等诸多优势。

ZSL421 LoRa 智能组网芯片是致远电子自主研发的 LoRa 智能组网芯片，该产品集成无线收发器，超低功耗 MCU，射频收发匹配电路和滤波电路。支持自组网透传协议、LoRaNET、LoRaWAN、LinkWAN、CLAA 等软件组网协议。



2. 平台选择



M6Y2C 是一款工业控制核心板，如图 2 所示，采用 NXP Cortex®-A7 800MHz 主频的处理器，以先进的电源管理架构带来更低功耗。M6Y2C 标配 8 路 UART、2 路 USB OTG、2 路 CAN-Bus、2 路以太网等接口；标配 128/256/512MB DDR3 和 128/256MB NAND Flash、硬件看门狗等。

使用的 LoRa 模块是基于 ZSL42x LoRa 智能组网芯片的 PCIE-ZSL421 LoRa 模块，如图 1 所示。



图1

3. 硬件连接

将 PCIE-ZSL421 插入 PCIE 座子即可，如图 2 所示。



图2

开始适配

1. 设备树修改

为了方便开发使用，PCIE-ZS L421 模块里面已有出厂固件，我们只需要配置相应的串口，对该模块进行配置及通信即可，打开核心板对应的设备树文件，使能硬件对应的串口，如下：

```
&uart5 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_uart5>;
    status = "okay";
};

在 iomux 节点下添加对应引脚的电气属性。
pinctrl_uart5: uart5grp {
    fsl,pins = <
        MX6UL_PAD_UART5_TX_DATA__UART5_DCE_TX 0x1b0b1
        MX6UL_PAD_UART5_RX_DATA__UART5_DCE_RX 0x1b0b1
    >;
};
```

配置完后编译烧录即可。

模块调试

烧录完成后，我们需要通过刚刚适配的串口对 PCIE-ZSL421 模块进行调试，调试时我们借助 ZSL42x-EVB Board 板子以及 Wireless Tool APP，使用 USB 线将电脑与 ZSL42x-EVB Board 板子连接后，打开 Wireless Tool 连接 ZSL42x-EVB Board，点击红框的按钮然后连接即可，如图 3 所示。（注：该工具可以在 ZSL42x 资料下载页面获取）

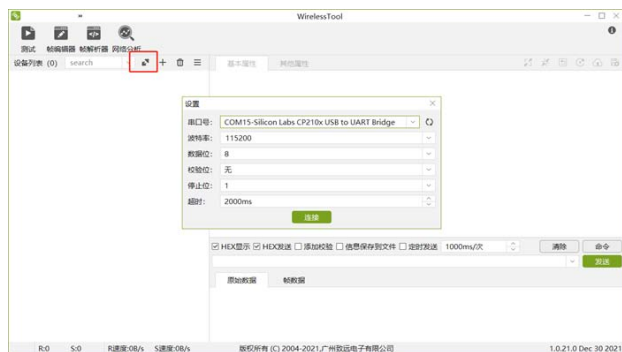


图3

连接后将设备地址设置为 20 02，设置信道为信道 1，使能透传功能，设置目标地址为 20 01，如图 4 所示，设置完后点击右上角的保存图标进行保存。



图4 Wireless Tool配置

使用串口连接到核心板的调试串口，输入以下命令对 PCIE-ZSL421 进行配置。

设置串口波特率：

```
stty -F /dev/ttymx4 raw speed 115200 cs8 -parenb -cstopb -echo min 0 time 50
```

设置 PCIE-ZSL421 地址为 20 01：

```
echo -n -e "\x7e\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttymx4
```

```
echo -n -e "\x7e\x00\x00\x06\x00\x03\x06\x02\x20\x01\x00\x0b" > /dev/ttymx4
```

使能 PCIE-ZSL421 透传模式：

```
echo -n -e "\x7e\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttymx4
```

```
echo -n -e "\x7e\x00\x00\x04\x00\x03\x09\x01\x00\x8f" > /dev/ttymx4
```

设置 PCIE-ZSL421 目标地址为 20 02：

```
echo -n -e "\x7e\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttymx4
```

```
echo -n -e "\x7e\x00\x00\x08\x00\x03\x1b\x00\x02\x20\x02\x04\x00\xcc" > /dev/ttymx4
```

设置 PCIE-ZSL421 信道为信道 1：

```
echo -n -e "\x7e\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttymx4
```

```
echo -n -e "\x7e\x00\x00\x04\x00\x03\x07\x01\x00\x8d" > /dev/ttymx4
```

配置完成后就可以当串口来使用了。

现在开始进行通信，首先在 PC 端关闭 Wireless Tool，打开串口工具，设置波特率为 115200，8n1，如图 7 所示，就可以进行测试，测试结果如图 5 所示。

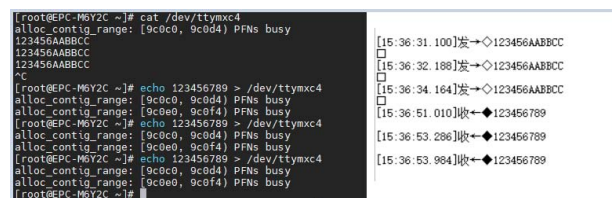


图5 PCIE-ZSL421通信结果

【产品应用】 M1126高清摄像头应用实例

原创 研发部 ZLG 致远电子 2023-03-14 11:37:58

这些年物联网高速发展，需要用到摄像头的场景越来越多。本文讲述以 M1126-EV-Board 和 MIPI camera-demo-6188 为配套的高清摄像头方案，以满足用户对高清录像的需求。



M1126-EV-Board 评估板是广州致远电子精心推出的集工控与评估于一身的评估底板。该评估底板可选搭配 M1126-L LGA 系列核心板，根据核心板的选型提供丰富的外围接口做评估和扩展使用，方便灵活。评估板提供 2 路 4 通道 MIPI 摄像头接口、1 路 4 通道 MIPI 显示屏接口、1 个 MiniPCIE 座子分别用于评估广域与局域无线模块、1 个 SIM 卡座、1 个 TF 卡座、1 路千兆以太网接口、1 路 USB Type C 接口、2 路扩展 IO 接口（包含 1 路 SPI、5 路 UART、1 路 I2C、1 路 CAN、1 路 PDM、9 路 GPIO、5 路 ADC）、1 个带麦克风的耳机座、1 个 UART 调试接口，满足 M1126-L 核心板的接口评估，也可以直接用于医疗电子、工业控制、人脸识别、人工智能等应用场合。

M1126-EV-Board 将 1 路 4 通道的 MIPI-CSI 数字摄像头引出，数字信号通讯采用 0.5mm 32Pin 的 FFC 连接器，连接器如图 1 所示，信号定义如表格 1 所示。

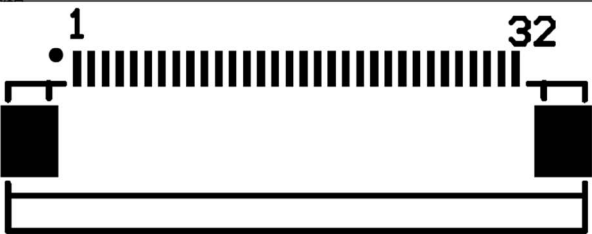


图1 MIPI-CSI摄像头接口

引脚	信号名称	功能	引脚	信号名称	功能
1	SV_VDD	5V 电源	2	SV_VDD	5V 电源
3	GND	地	4	Flash_LED+/NC	闪光灯+/悬空
5	Flash_LED-/NC	闪光灯-/悬空	6	SCL	I2C 时钟
7	SDA	I2C 数据	8	SYS_EN	系统使能
9	SYS_RST	系统复位	10	SYS_CLK	系统时钟
11	GND	地	12	GPIO/CAM_PCLK	GPIO/像素同步时钟
13	GPIO/CAM_HS	GPIO/水平同步信号	14	GPIO/CAM_VS	GPIO/垂直同步信号
15	GND	地	16	GPIO/CAM_D0	GPIO/数据位 0
17	GPIO/CAM_D1	GPIO/数据位 1	18	GND	地
19	CSI_CLK_P/CAM_D2	CSI 时钟正/数据位 2	20	CSI_CLK_N/CAM_D3	CSI 时钟负/数据位 3
21	GND	地	22	CSI_D0_P/CAM_D4	CSI 数据通道 1 正/数据位 4
23	CSI_D0_N/CAM_D5	CSI 数据通道 1 负/数据位 5	24	GND	地
25	CSI_D1_P/CAM_D6	CSI 数据通道 2 正/数据位 6	26	CSI_D1_N/CAM_D7	CSI 数据通道 2 负/数据位 7
27	GND	地	28	CSI_D2_P/CAM_D8	CSI 数据通道 3 正/数据位 8
29	CSI_D2_N/CAM_D9	CSI 数据通道 3 负/数据位 9	30	GND	地
31	CSI_D3_P/CAM_D10	CSI 数据通道 4 正/数据位 10	32	CSI_D3_N/CAM_D11	CSI 数据通道 4 负/数据位 11

表格 1 MIPI-CSI摄像头接口引脚定义

本文将使用 MIPI camera-demo-6188 摄像头作为 M1126-EV-Board 评估板的配件，该摄像头支持 1~4 路串行摄像头输入，1 路 MIPI CSI 输出。如图 2 所示。



图2 MIPI camera-demo-6188摄像头配件

nvp6188 支持两种分辨率 1920x1080, 1280x720，默认 1920x1080，可以通过下面命令设置分辨率：

```
$ media-ctl -vvv -d /dev/media0 --set-v4l2 '"m01_b_nvp6188 3-0030":0[fmt:YUYV8_2X8/1280x720]'
```

```
$ media-ctl -vvv -d /dev/media0 --set-v4l2 '"m01_b_nvp6188 3-0030":0[fmt:YUYV8_2X8/1920x1080]'
```

可用下列命令在板上预览画面：

```
$ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=Nv12,width=1280,height=720,framerate=25/1 ! queue ! videoconvert ! \
waylandsink
```

预览画面如图 3 所示：



图3 M1126&NVP6188预览画面

预览画面需要接屏，当未接 AHD 模拟摄像头时，看到的是 nvp6188 输出的测试彩条，如果接了 AHD 模拟摄像头，则正常情况应该看到摄像头画面。nvp6188 对应的 4 路视频设备节点分别是：/dev/video0、/dev/video1、/dev/video2、/dev/video3。

如果未接屏，还可以录制视频数据，然后复制到 PC 上回放：

```
$ v4l2-ctl -d /dev/video1 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=5 --stream-skip=3 \
--stream-to=/userdata/myvideo.nv12 --stream-count=250 --stream-poll
# dffplay -i myvideo.nv12 -f rawvideo -pixel_format nv12 -s 1920x1080d /* 在 Ubuntu 终端执行 */
```


边缘计算 ▼

【技术分享】 为何EtherCAT在运动控制总线中如此受欢迎？

原创 研发部 ZLG 致远电子 2023-03-16 11:39:53

现场总线通讯将逐渐替代传统运动控制器中的脉冲控制，追求更高效率的数据传输。当前国内流行的驱动器通讯有 CANOPEN、RS232、RS485、EtherCAT 等，本文与您一起了解 EtherCAT 总线。

什么是EtherCAT

EtherCAT(以太网控制自动化技术)是一个以以太网为基础的开放架构的现场总线系统，最初由德国倍福研发。EtherCAT 为系统的实时性能和拓扑的灵活性树立了新的标准，是一种开放性的协议，开放标准由现在的 EtherCAT 技术组 (ETG) 管理。

为什么选择EtherCAT

1. 精简的通信结构

OSI 模型各层间关系和通讯时的数据流向如图 1 所示。

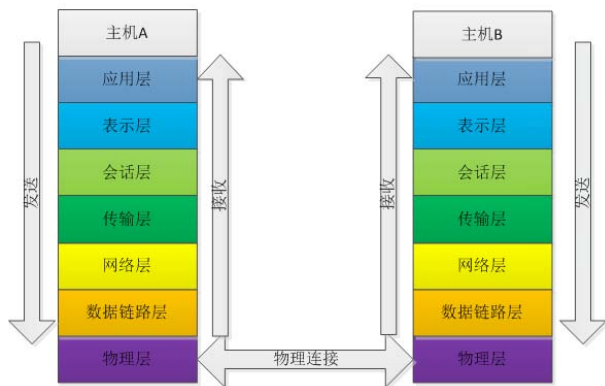


图1 OSI模型示意图

RS232、RS485 都属于 ISO 七层结构中的物理层，实现完整的通讯，需要加入数据链路层和应用层。RS232、RS485 一般需要使用串口作为链路层对物理层进行指挥。RS485 应用层协议通常使用 MODBUS，CANOPEN 也是应用层协议。CANOPEN 和 EtherCAT 通常分别结合 can、ethernet 实现链路层和物理层，EtherCAT 在 ISO 模型设计上也仅使用应用层、数据链路层、物理层。

在软件设计层次中，物理层向上走，将上一层帧再封装成特定帧后传输，通信过程将存在组包拆包的过程，越往顶层走帧格式越复杂，相比较其它的以太网协议，可见 EtherCAT 设计实现的精简之处，体现出了更优越的实时性。

2. 优越的综合性能

2.1 传输高效

EtherCAT 的物理层中使用光信号进行传输，传输速率达 100Mbit/s(100 base-Tx)，且为全双工传输，相比于上述总线均为电气传输。RS232 抗干扰能力差，RS485 半双工，can 一帧只能传输 8 个字节，而以太网传输一帧可以高达 1514 字节。以太网使用光纤为传输媒介，线束简洁，同时抗干扰能力更强，传输距离更远。

2.2 拓扑灵活

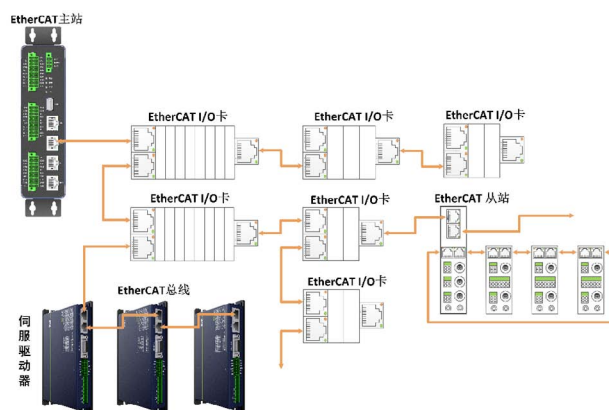


图2 EtherCAT 拓扑结构图

EtherCAT 符合以太网标准，支持多种拓扑结构：线型、星型、树型，以一种“逻辑闭环”的方式传播，灵活性更强。主站实现只需要一张普通网卡，不需要交换机与路由器的参与，解决传统以太网的交换机延时、堆栈延时、带宽利用率等问题。

2.3 时钟同步

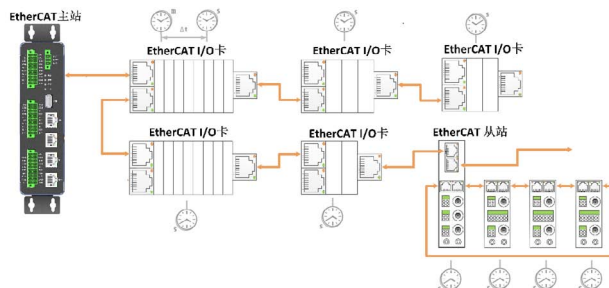


图3 DC 分布式时钟图

在多轴运动控制中，时钟同步的精准性能对数据传输产生直接影响，可能会导致丢帧，运动不协调，从而无法控制对各轴设备的同步执行。EtherCAT 支持 DC 分布式时钟，且基于硬件的产生的时钟进行校准和补偿，大大减小了系统的抖动时间。

EtherCAT 实现了数据传输中速率更快、实时性更高、更可靠、传输数据量更大、抗干扰能力更强的等特点，这恰是市场日益要求的性能指标，您心动了吗？

EtherCAT帧结构

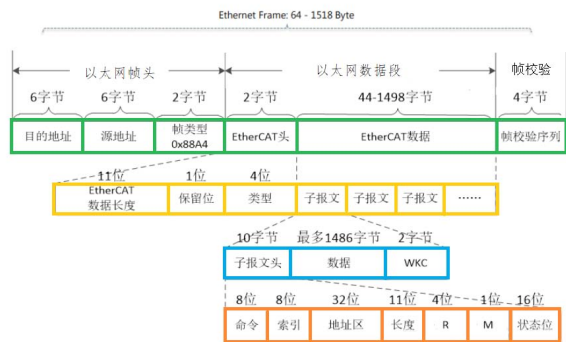


图4 以太网帧格式图

名称	含义
目的地址	接收方 MAC 地址。
源地址	发送方 MAC 地址。
帧类型	0x88A4
EtherCAT 头部	长度:报文总长度 类型:1 表示与从站通信。
EtherCAT 数据段	该段存放着应用层发送的数据。
帧类型	帧校验

表2 Ethercat状态机含义表

EtherCAT 帧组成结构如上图所示，以太网帧头中使用的帧类型是 EtherCAT 专有的 0x88a4 类型，后面接着就是 EtherCAT 的数据内容了，我们的应用层传输的内容就存放在子报文结构中的数据段，容纳了更大的数据传输量。数据段可以包含一个或多个 EtherCAT 子报文构成。当然，EtherCAT 也是支持嵌入 UDP/IP 格式的数据帧格式，数据段中会让出 28 个字节长度以供 UDP/IP 使用。

EtherCAT应用层

1. ESM状态机

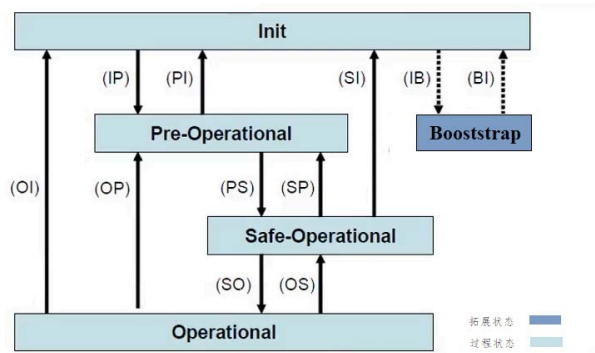


图5 状态机转换关系图

状态	说明
Init	主站和从站之间没有应用层上的通讯，主站可以访问从站 DL 相关状态寄存器信息。
Pre-Operation	如果从站支持邮箱通信，主站和从站可以使用邮箱和相关协议进行应用层次的初始化和参数配置，此状态下不能进行过程数据通信。
Safe-Operation	可以进行过程数据通信，从站可以进行数据的输入，但是不允许数据输出，数据输出处于“安全”状态。
Operation	从站可以进行数据的输入输出操作。

表2 Ethercat状态机含义表

EtherCAT 状态机定义了每一个 EtherCAT 从站设备的分布设置，并指示了可用的功能。ESM 状态的改变是由主站请求发起的，从站需要一级一级的向下进行转换状态。而上行可以降级切换，如状态 op 切换到 Init 状态，如上图 5 箭头方向所示。

2. 寻址方式

- 广播寻址

将数据广播到各个从站设备。

- 自增寻址

根据设备连接的顺序进行编号，从 0 开始，每经过一个从站数值 -1，以此递推。

- 设置寻址

相当于给每一个从站分配地址，主站通过从站地址找到对应的从站进行通信。

- 逻辑寻址

主站配备 4G 的数据逻辑地址空间，FMMU 将数据逻辑地址和从站中的物理地址进行映射，主站在帧中只需携带某个逻辑地址，即可寻址到任意从站中的任意数据，是一种更加灵活的寻址方式。

3. 数据传输

3.1 同步管理器 SM

同步管理器 (Syncmangers) 用来协调应用层和主机的数据交互，实现数据同步而不是时间同步，确保数据正确读取和写入，并且同步管理可以以中断的形式通知主机和应用程序更新事件，然后进入中断服务函数中处理即可。SM 通道一共用到 4 个，SM0 和 SM1 用于邮箱数据输出和输入，SM2 和 SM3 用于过程数据的输出和输入。

3.2 应用层协议

EtherCAT 应用层协议支持 COE、VOE、FOE、EOE 等协议，而在电机控制领域我们常用到的是 COE 协议，COE 全称为 CanOpen Over EtherCAT，相当于 CanOpen 协议通过 EtherCAT 方式来实现，都是基于 cia402 协议进行编写。主站可以通过邮箱 (SDO) 或 PDO 对从站的对象字典进行读写操作，实现数据通讯。邮箱常用对从站的配置和读写，采用应答方式，而 PDO 为过程数据，实现数据通信的实时和快速传输。

4.2 配置软件

本文使用的配置软件是 ZLG 致远电子团队开发的 EtherCAT 网络配置工具，开发者导入 esi 从站描述文件后进行相关需求配置，生成相应的 eni 文件，EtherCAT 网络运行时将对所有设备进行初始化。EtherCAT 配置软件支持以下功能供帮助开发者快速使用与开发。

- 浏览设备信息

显示主站和从站的设备信息，如制造商信息，网络配置信息等内容。

- 浏览拓扑结构

配置软件显示主站和从站的连接拓扑图。

- FMU/SM 与 DC 配置

FMU、SM、PDO 相关的信息与配置，DC 分布时钟的操作模式选择。

- 浏览输入与输出变量

显示 PDO 数据的输入与输出数据信息，如通道、名称、类型、位长等信息。

- 邮箱功能

支持选择配置邮箱轮询与读取对象字典的方式，显示相关的 COE 对象字典列表。

- 内存信息

浏览内存偏移信息与 eeprom 的参数信息等内容。

- 集成帮助文档

浏览帮助文档，查阅更多的功能介绍与 demo 示例，帮助使用与开发。

EtherCAT 配置软件主界面如下图 6 所示。

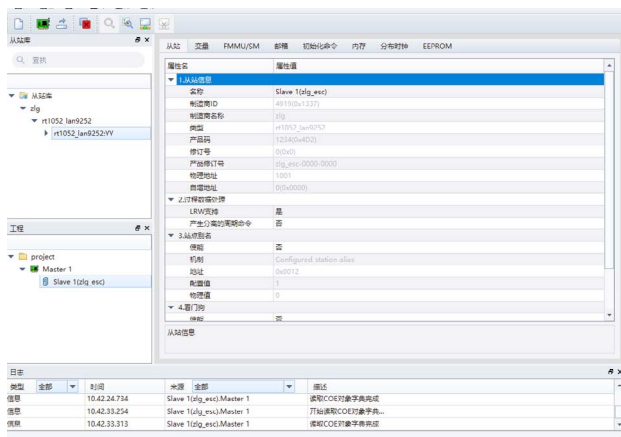


图6 EtherCAT 配置软件主界面图

产品案例



图7 ZMC600E EtherCAT 主站控制器

1. 产品介绍

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

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

2. EtherCAT性能优势

- ZMC600E 集成了商业授权的 EtherCAT 主站解决方案；
- Cortex-R5F 协处理器独立处理 EtherCAT 数据收发，实时性更强；
- 精准分布式时钟，抖动 $<1\mu s$ ；
- 最多支持 128 个从站节点，节点间距最大 100m；
- 支持 CoE 对象读写，SoE IDN 读写；
- 循环周期最小达 $125\mu s$ ，支持线形、树形、星型拓扑结构。

【深度解析】EM-1000储能网关的CAN性能有多强？

原创 研发部 ZLG 致远电子 2023-03-23 11:33:16

针对储能应用，致远电子推出 EM 系列储能边缘智能网关，其中 EM-1000 储能网关是一款高性能、多接口通讯，并拥有 3 路 CAN-bus 的管理设备。接下来让我们看看它的 CAN-bus 性能有多强！

EM-1000储能网关介绍



EM-1000 储能网关是致远电子新推出的 EM 系列储能边缘智能网关之一，该系列是专为新能源储能系统设计的一款高性能、多接口通讯管理设备。可在储能系统应用中作为边缘 EMS(能源管理系统) 总控、通讯管理机、規約转换器或 BAU(电池管理总控) 使用。该系列产品集成丰富的外设接口，支持各类 BMS、PCS、空调、电表、屏显等设备的通讯传输，且软件上支持 RT-Linux、Ubuntu 等操作系统，支持 IEC-61850/IEC-104/EtherCAT 等专用协议，可广泛满足各类储能系统的本地能源管理应用需求。

EM-1000 储能网关具有三路 CAN 总线，包括 2 路原生 CAN FD 接口和 1 路扩展 CAN 2.0 接口。在储能应用中，CAN 总线的性能是重要的评估条件之一，以下将对 EM-1000 储能网关的 3 路 CAN-bus 接口进行性能测试。

测试方法

本次测试是为了测试 EM-1000 储能网关上的 3 路 CAN 的性能（包括 CAN FD），主要指收发速率，具体测试内容如下：

1. EM-1000 储能网关发送和接收 100,000 帧 CAN 帧所需的时间，并计算出其收发速率。包括 50kbps、100kbps、125kbps、250kbps、500kbps、800kbps 和 1Mbps 共 7 个仲裁域波特率；
2. EM-1000 储能网关发送和接收 100,000 帧 CAN FD 帧所需的时间，并计算出其收发速率。包括 50kbps、100kbps、125kbps、250kbps、500kbps、800kbps 和 1Mbps 共 7 个仲裁域波特率。

测试准备

1. 本次性能测试需准备一台 PC、一台 EM-1000 以及一个 USB 转 CAN 模块。USB 转 CAN 模块使用致远电子的 USBCANFD-100U-mini 模块。
2. 在 PC 上安装 USBCANFD-100U-mini 的驱动以及上位机软件 ZCANPRO。软件和驱动可在 ZL G 官网的 USBCANFD 系列页面的资料下载页面中下载。

USBCANFD系列CANFD接口卡			
用户手册	功能特点	参数与配件	资料下载
【用户手册】USBCANFD-100U-mini用户手册 V1.00 [下载: 1787次, 大小: 2MB, 更新日期: 2021-01-26]			
【用户手册】USBCANFD-100U-mini用户手册 (简体中文) V1.02 [下载: 334次, 大小: 953KB, 更新日期: 2022-05-31]			
【用户手册】USBCANFD-100U-mini用户手册 (简体中文) V1.02 [下载: 86次, 大小: 936KB, 更新日期: 2022-05-31]			
【用户手册】USBCANFD-800U产品用户手册 V1.00 [下载: 256次, 大小: 980KB, 更新日期: 2022-07-25]			
应用软件			
【应用软件】CAN-bus综合应用软件ZCANPRO_Setup_V2.2.5 [下载: 45755次, 大小: 39.4MB, 更新日期: 2023-02-03]			
【应用软件】ZLGCAN上位机例程 [下载: 25361次, 大小: 60MB, 更新日期: 2023-02-16]			
驱动库			
【驱动库】CAN接口卡二次开发接口函数库 [下载: 39114次, 大小: 6.99MB, 更新日期: 2022-10-20]			
【驱动库】USBCANFD-100(200)-H驱动安装 [下载: 11145次, 大小: 6MB, 更新日期: 2020-07-08]			
【驱动库】USBCANFD-Linux-X86 驱动包及安装说明 [下载: 1581次, 大小: 47KB, 更新日期: 2020-11-28]			

图1 驱动以及上位机下载页面

3. 完成硬件连线，将 USBCAN 模块的 USB 口插入 PC，CAN 接口的 CAN_H 和 CAN_L 插入 EM-1000 储能网关的 CAN_H 和 CAN_L。



图2 USBCAN模块与EM-1000储能网关的接线

4. 打开 ZCANPRO 软件，此时弹出设备管理页面，在页面中选择 USBCANFD-MINI 类型，点击打开设备。

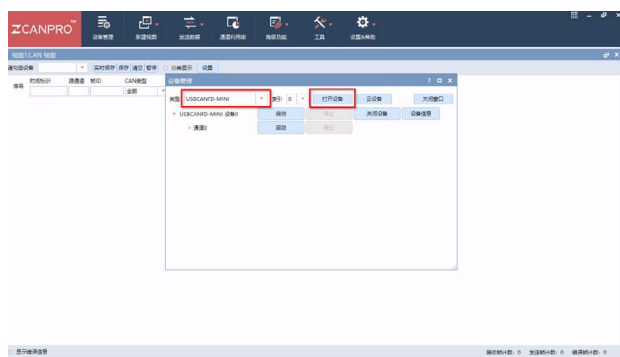


图3 ZCANPRO软件打开USBCAN设备

至此，准备工作完成。

测试步骤

1. CAN的速率测试

1.1 在设备管理页面中，点击启动，选择 CAN 协议为 CAN，选择需测试的仲裁域波特率，然后点击确认。



图4 ZCANPRO软件的CAN协议启动

1.2 在 EM-1000 储能网关中运行以下命令启动 can 设备：

```
# ifconfig can0 down  
# ip link set can0 type can bitrate 1000000  
# ifconfig can0 up
```

1.3 测试发送帧速率时，需先清空 ZCANPRO 的接收帧数据，然后在 EM-1000 储能网关上运行以下命令发送 100,000 帧：

```
# cangen can0 -g 0 -p 10 -L 8 -n 1000000
```

等待发送结束后，查看接收帧计数是否为 100,000，以及错误帧是否为 0，若一切正常，则可将最后一帧的时间表示作为发送时间，计算发送帧速率。

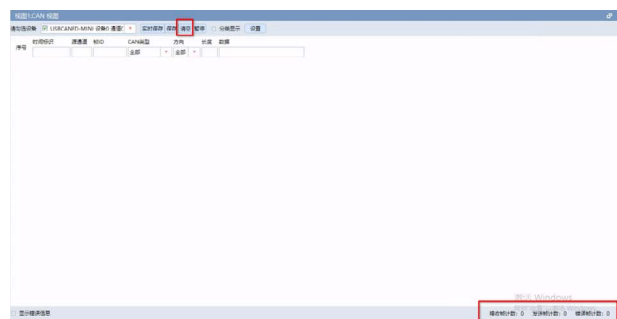


图5 测试CAN发送速率前清空帧数据

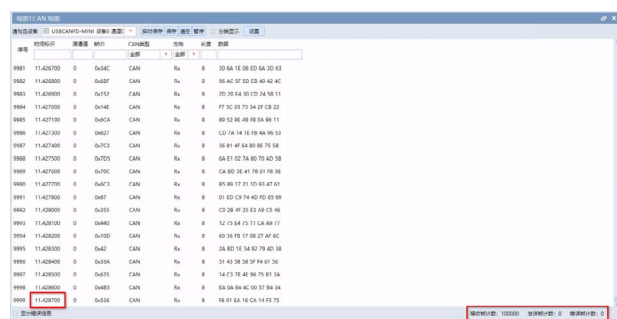


图6 测试CAN发送速率的结果

1.4 测试接收帧速率时，先在 EM-1000 储能网关上运行命令：

```
# candump can0 -n 1000000 -l
```

该命令的主要作用是，在 can0 设备上接收 100,000 帧的数据，记录到文件中。如果未接收到 100,000 帧数据，命令不会停止。然后，在 ZCANPRO 软件中打开发送数据中的普通发送页面，配置发送次数为 100 以及每次发送帧数为 1000，即为 100,000 帧。最后，点击立即发送即可。

发送成功后，EM-1000 储能网关的命令会自动退出，并且 ZCANPRO 软件的发送帧计数为 100,000 帧，错误帧计数为 0。此时，可根据 ZCANPRO 的发送时间作为接收时间来计算接收帧速率。

```
root@EM1000:~# candump can0 -n 100000 -l  
Disabled standard output while logging.  
  
Enabling Logfile 'candump-2023-03-09_095423.log'  
  
root@EM1000:~# cat candump-2023-03-09_095423.log -n | tail -n 5  
99996 (1678326879.109314) can0 100#0011223344556677  
99997 (1678326879.109434) can0 100#0011223344556677  
99998 (1678326879.109556) can0 100#0011223344556677  
99999 (1678326879.109679) can0 100#0011223344556677  
100000 (1678326879.109798) can0 100#0011223344556677  
root@EM1000:~#
```

图7 EM-1000储能网关接收CAN帧结果

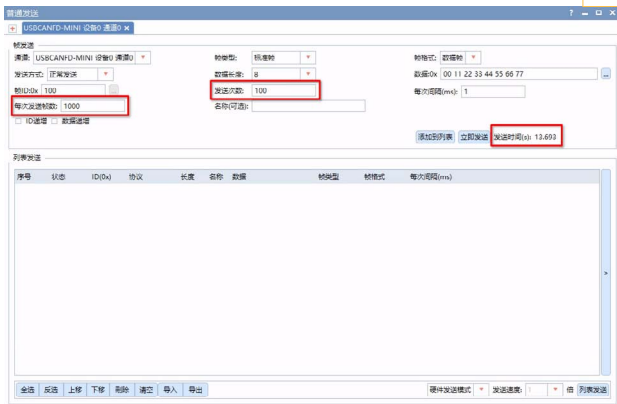


图8 ZCANPRO软件的发送CAN帧结果

2. CAN FD的速率测试

2.1 在设备管理页面中，点击启动，选择CAN协议为CAN FD，选择需测试的数据域波特率，然后点击确认。



图9 ZCANPRO软件的CAN FD协议启动

2.2 在EM-1000 储能网关中运行以下命令启动 CAN FD:

```
# ifconfig can0 down
# ip link set can0 type can bitrate 10000000 dbitrte 5000000 fd on
# ifconfig can0 up
```

注意：can0 为第 1 路 CAN 设备，第 2 路和第 3 路 CAN 设备为 can1 和 can2。命令中的 1000000 是仲裁域波特率，单位为 bps。1000000bps 为 1Mbps，dbitrte 为数据域波特率，单位为 bps。5000000bps 为 5Mbps。

2.3 测试发送帧速率时，需先清空 ZCANPRO 的接收帧数据，然后在 EM-1000 储能网关上运行以下命令发送 100,000 帧：

```
# cangen can0 -f -g 0 -p 10 -L 8 -n 100000
```

等待发送结束后，查看接收帧计数是否为 100,000，以及错误帧是否为 0，若一切正常，则可将最后一帧的时间表示作为发送时间，计算发送帧速率。

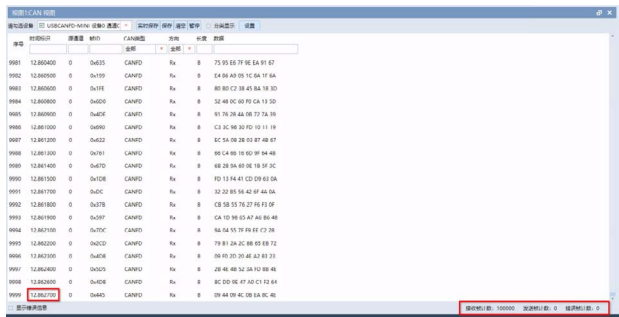


图10 测试CAN FD发送速率的结果

2.4 测试接收帧速率时，先在 EM-1000 储能网关上运行命令：

```
# candump can0 -n 100000 -l
```

该命令的主要作用是，在 can0 口上接收 100,000 帧的数据，记录到文件中。如果未接收到 100,000 帧数据，命令不会停止。然后，在 ZCANPRO 软件中打开发送数据中的普通发送页面，配置帧类型为 CAN FD，发送次数为 100 以及每次发送帧数为 1000，即为 100,000 帧。最后，点击立即发送即可。

发送成功后，EM-1000 储能网关的命令会自动退出，并且发送帧数为 100,000 帧，错误帧计数为 0。此时，可根据 ZCANPRO 的发送时间作为接收时间来计算接收帧速率。

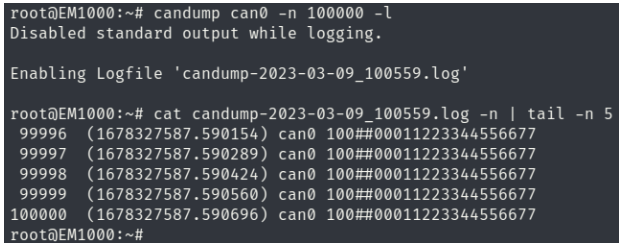


图11 EM-1000储能网关接收CAN FD帧结果



图12 ZCANPRO软件的发送CAN FD帧结果

3. 测试结果

仲裁域波特率	接收时间(秒)	接收速率(帧/秒)	发送时间(秒)	发送速率(帧/秒)
1Mbps	13.772	7261.109	11.430	8748.906
800kbps	16.621	6016.485	14.287	6999.370
500kbps	25.183	3970.933	22.860	4374.453
250kbps	47.910	2087.247	45.719	2187.274
125kbps	93.893	1065.042	91.430	1093.733
100kbps	116.909	855.366	114.290	874.967
50kbps	232.199	430.665	228.567	437.508

图13 CAN0设备的CAN速率 (100,000帧)

仲裁域波特率	接收时间(秒)	接收速率(帧/秒)	发送时间(秒)	发送速率(帧/秒)
1Mbps	15.637	6395.089	12.863	7774.236
800kbps	18.790	5321.980	16.078	6219.679
500kbps	28.542	3503.854	25.725	3887.269
250kbps	54.185	1845.529	51.448	1943.710
125kbps	106.190	941.708	102.904	971.780
100kbps	132.166	756.624	128.618	777.496
50kbps	262.125	381.497	257.237	388.747

图14 CAN0设备的CAN FD速率 (100,000帧)

仲裁域波特率	接收时间(秒)	接收速率(帧/秒)	发送时间(秒)	发送速率(帧/秒)
1Mbps	13.831	7230.135	11.430	8748.906
800kbps	16.529	6049.972	14.287	6999.370
500kbps	24.993	4001.120	22.859	4426.993
250kbps	47.772	2093.276	45.717	2187.370
125kbps	93.761	1066.542	91.432	1093.709
100kbps	116.689	856.979	114.278	875.059
50kbps	231.662	431.663	228.584	437.475

图15 CAN1设备的CAN速率 (100,000帧)

仲裁域波特率	接收时间(秒)	接收速率(帧/秒)	发送时间(秒)	发送速率(帧/秒)
1Mbps	15.531	6438.735	12.863	7774.236
800kbps	18.700	5347.594	16.078	6219.679
500kbps	28.423	3518.277	25.725	3887.269
250kbps	54.073	1849.352	51.447	1943.748
125kbps	106.089	942.605	102.893	971.883
100kbps	132.035	757.375	128.620	777.484
50kbps	262.003	381.675	257.228	388.760

图16 CAN1设备的CAN FD速率 (100,000帧)

仲裁域波特率	接收时间(秒)	接收速率(帧/秒)	发送时间(秒)	发送速率(帧/秒)
1Mbps	26.666	3750.094	16.615	6018.658
800kbps	27.976	3574.492	17.187	5818.351
500kbps	31.482	3176.418	25.133	3978.833
250kbps	53.411	1872.274	47.885	2088.337
125kbps	98.737	1012.792	93.846	1065.576
100kbps	121.446	823.411	116.886	855.534
50kbps	235.435	424.746	231.884	431.250

图17 CAN2设备的CAN速率 (100,000帧)

【产品应用】 ZLM3100S应用--快速驱动永磁同步电机风机

原创 研发部 ZLG 致远电子 2023-03-30 11:41:41

《电机能效提升计划（2021-2023）》中提到，加快高效节能电机推广应用，推进电机系统智能化，电机系统数字化应用。ZLM3100S 电机驱动器为此量身设计，助力工业风机、水泵行业电机能效提升改造。

永磁同步电机（PMSM）在转子上使用永磁体提供励磁，因此转子上无铜损和铁损，另一方面无励磁电流，不需要从电网吸收无功电流，所以电机在很宽的负载范围内能保持接近于 1 的功率因数，高效节能。

随着近十年来高耐热性、高磁性能永磁体的成功研发并产业化，集成电路和电力电子元件技术的快速发展，永磁同步电机迎来了一个黄金时代，在新能源汽车、轨道交通、电梯、医疗器械、石油化工、家电、风机、水泵等行业得到了广泛的应用。风机主要应用于冶金、电力、城市轨道交通、养殖、工厂、高层建筑等各种场所通风换气，本章将带大家快速使用 ZLM3100S 驱动器驱动永磁同步电机风机。

驱动器工作原理简述

永磁同步电机的定子有三相绕组，三角形的联结方式由于三相绕组不完全对称，内部会产生环流，所以基本上都为星形联结方式，将定子三相绕组接到示波器测量它的电压，匀速转动它的转子，可以在示波器上看到 3 根正弦曲线，而且三根曲线两两相位差为 120°，这是发电机。那么反过来我们在永磁同步电机定子的三相线圈上输入三相正弦电压，相位差两两相差 120°，就可以让永磁同步电机平稳高效地运转，就是电动机啦。

驱动器的工作原理是先将三相（或单相）电压整流、滤波得到直流电，然后使用 SVPWM 技术产生调制信号，驱动三相逆变电路，合成出等效的三相正弦电压来驱动电机。

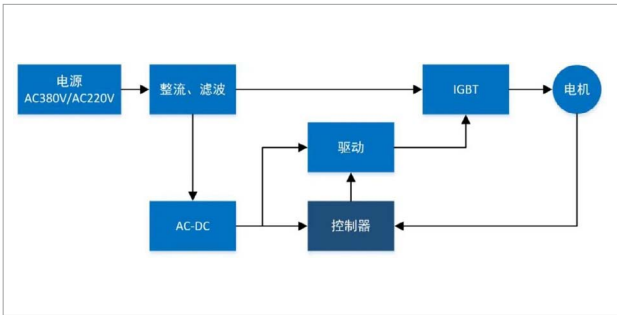


图1 驱动器系统框图

ZLM3100S驱动器介绍



图2 ZLM3100S驱动器

1. 简介

ZLM3100S 是新一代风机、水泵高效节能的永磁同步电机驱动器，该驱动器可以装在电机尾部，和电机进行全封闭一体式结构设计，结构紧凑，防护能力强；高功率密度，节能、高效，符合国家能效等级标准；该电机驱动器采用无感矢量变频技术，具有节能、安静、高可靠等特点；采用无位置传感器的技术，零速启动，自动弱磁控制，最大转矩控制等功能；其中支持“电压自适应”算法，硬件无高压电解电容器设计和“高压电源变换”技术，主板体积小、发热量低、抗干扰能力强、寿命长。

2. 驱动器参数

型号	ZLM3100S
输入电源	三相 AC380V±15%，50/60Hz
通讯功能	RS485，Modbus 通讯协议
可调转速范围	500RPM~1320RPM
保护功能	欠压、过压、过流、堵转、过温等
其他功能	顺逆风启动，缺相保护
散热方式	壳体散热，自冷
编码器	无
启动延时时间	20s(可设置)
工作/存储温度	-20℃~60℃/-40℃~85℃
海拔高度	2000m 以下

ZLM3100S驱动器快速使用

ZLM3100S 驱动器使用无感 FOC 控制，通过精确地控制磁场大小与方向，使得电机的运动转矩平稳、噪声小、效率高，并且具有高速的动态响应等优点，其缺点在于需要进行参数整定，匹配电机参数。

1. 标配电机参数

使用标配电机参数差不多的电机，可直接使用 ZLM3100S 驱动器平稳驱动。

参数	典型值	单位
额定电压	AC380	V
额定功率	1100	W
额定转速	1320	RPM
极对数	5	对极
线电阻	7.2	Ω
D轴电感	35	mH
Q轴电感	40	mH
反电动势常数	234	V/KRPM

2. 安装散热器

IGBT 模块是大功率器件，在工作时会产生大量的热量，IGBT 模块正常工作时结温的范围是 $-40^{\circ}\text{C} \sim 150^{\circ}\text{C}$ ，驱动器输出功率较大时需要安装散热器辅助散热，才能将 IGBT 模块的温度控制在工作范围内。一般将驱动器安装在电机后端盖，利用电机壳体散热，自然冷却。

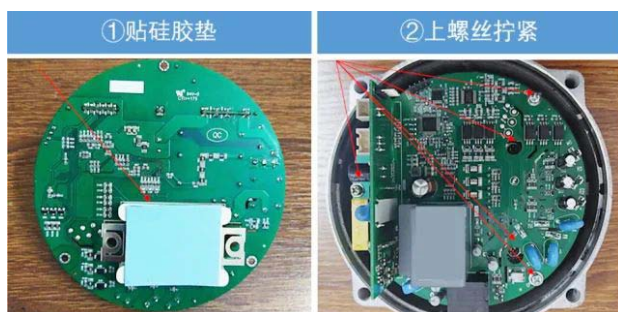


图3 散热器安装过程

3. 驱动器接线

ZLM3100S 驱动器单台设备接线图如下图所示，多台设备连接可根据单台设备接线图扩展。示例中使用空气开关作为主开关，由于空气开关不能频繁操作，所以这适用于不需要频繁通断电的使用场景，若需要频繁通断电，需要在空气开关和驱动器电源之间加装接触器。驱动器作为从设备，RS485 通讯接口默认不焊接 120Ω 终端电阻，多台设备组网时，需要在最后一台设备的 RS485 接口处添加 120Ω 电阻 ($1/4\text{W}$)，以保证通讯质量。在复杂的工业环境和长距离通讯时，RS485 通讯线使用双绞屏蔽线。

ZLM3100S 驱动器的 RS485 总线默认最大支持 32 个节点，若需要支持更多的节点需要联系我们修改电路配置，或使用 RS485 中继器。

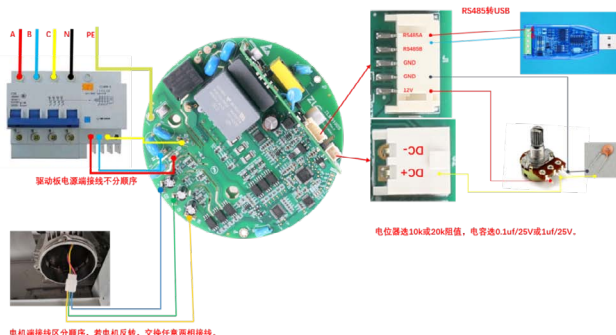


图4 ZLM3100S接线图

4. 电位器调速

接好线之后接通电源，电机就能运转起来了，默认以 1320RPM 转速运行，如果想要调节转速，有两种方式，RS485 调速和电位器调速，电位器调速功能默认使能，接好电位器后，旋转电位器即可让电机在 500RPM~1320RPM 转速范围内运转。



图5 风机运行

若只使用 485 调速，为防止误动作，用串口助手软件以 16 进制发送命令：01 06 00 06 00 00 69 CB，即可将模拟调速功能失能。



5. RS485调速

示例 - 设置转速

设备地址：1

期望转速：1320RPM

• 请求帧

进制	地址	功能码	寄存器地址	数据	CRC
DEC	1	6	1	1320	17626
HEX	01	06	00 01	05 28	DB 44

• 应答帧
与请求帧一致。

6. 智能风机系统

EsDA + MPC-ZC1 工控单板 + ZLM3100S 电机驱动器可快速搭建智能风机控制系统。具体实现过程请查阅文章《EsDA MPC-ZC1 应用——IoT 监测控制系统（三）》

【产品应用】 ZigBee技术的典型应用方案介绍

原创 研发部 ZLG 致远电子 2023-03-09 11:30:34

ZigBee 无线通信技术已广泛应用于社会的各个行业领域，那么 ZigBee 产品都适用于什么具体场景？给行业带来哪些不一样的使用体验？本期文章我们一起来了解致远电子 ZigBee 产品的几个典型应用场景。

智慧路灯

据统计，截至 2022 年中国城市道路长度已经超过 57 万公里，对应照明灯用量已经达到 3440 万盏。对于占用全国照明用电总量 30% 的道路照明，ZigBee 技术给路灯带来了以下优势：

1. 节能

通过独立控制每个路灯的照明时间、亮度等状态，较传统方案节能 70% 以上。

2. 维护便捷

通过 ZigBee 构成路灯局域网，即可实时监控每盏灯的状态，无需定期巡检。

3. 功能扩展

目前，电子广告牌、摄像等设备不断集成在路灯上，给路灯增加无线控制单元，意味着智慧城市的更多可能。



图1 ZigBee广泛应用于智慧路灯行业

致远电子研发的 ZM21 系列和 ZM32 系列是高性能 ZigBee 模块，以高达 20dBm 的发射功率和 -103dBm 的接收灵敏度保证了模块的穿墙能力和超过 3.3 公里的视距通信距离。在 Mesh 组网方面，Zigbee 模块更是能在百秒以内完成 100 台设备的自组网，且支持短地址、长地址、数据帧地址三种通信方式动态切换，使用更加灵活。凭借产品优秀的射频性能、便捷的组网应用，以及稳定的控制功能，致远电子的 Zigbee 方案现已作为行业经典案例应用于智慧路灯行业。



光伏领域

传统的太阳能发电站和光伏发电站中，太阳能板、定日镜等通信节点均采用有线形式。但随着光伏领域的高速发展，激增的节点数量带来巨大的布线成本。所以目前的光能发电站，急需转向无线形式的数据交互。

致远电子开发的 AW5161 系列 ZigBee 模块，涵盖陶瓷天线和 IPEX 座的无线输出方式，并且实现了产品的最小尺寸仅为 13.5mm×16.5mm，所以除了组网灵活便捷外，该系列在光伏领域产品的小型化和轻量化方面有显著优势，现已应用于多个光能发电通信系统。



图2 ZigBee技术助力光伏发电

智慧灌溉

在农业领域，由于农田覆盖面积较大，供电难度高，想要做到每个灌溉点的精准控制，使用有线的方式将带来极高的成本。此时就需要通过无线通讯方式，控制每个灌溉点的工作状态，根据不同的天气、季节情况进行实时调整，有效节约资源。



图3 智慧灌溉中的ZigBee应用

致远电子开发的 AW5161 系列低功耗 Zigbee 模块为智慧灌溉领域提供了优秀的解决方案。其通信范围可达 3 公里，而且在产品功耗方面，AW5161 系列低功耗性能非常优异，休眠功耗低至 100nA，完全实现电池长期供电，目前已在多地大规模使用。

互联互通 ▼

【产品应用】 如何使用ZM602模组实现公里级Wi-Fi通信？

原创 研发部 ZLG 致远电子 2023-03-22 11:32:20

常规的 Wi-Fi 产品，通信距离一般都在百米以内，致远电子推出的 ZM602 是一款公里级通信距离的 Wi-Fi 模组，它在实际环境中的通信距离测试情况如何？本期文章带你一起了解。

通信距离的影响因素

在往期的文章《无线通信究竟如何传得更远？》中，我们介绍了无线通信中影响通信距离的三个主要因素：信号强度、最小接收灵敏度、信噪比。其中，信号强度和最小接收灵敏度主要受产品的性能参数影响，而信噪比则主要受通信环境的影响。

产品的射频性能是实现远距离通信的基础。ZM602 系列 Wi-Fi 模组拥有优异的射频性能，802.11b 模式下，发射功率高达 18dBm，最小接收灵敏度可达 -98dBm，高达 116dB 的链路预算，为实现超远距离 Wi-Fi 通信提供了可能。



图1 空中的无线通信及干扰

设备参数配置



ZM602 系列模组是一款 Wi-Fi+BLE 的 IoT 串口透传模组，主要应用于低速率、远距离应用场景，支持 Wi-Fi、BLE 双串口数据传输，支持 TCP/UDP/MQTT/HTTP 等常用通信协议，Wi-Fi 支持 Station 模式（客户端）和 SoftAP 模式（热点）。

本次通信距离测试，我们使用两个 ZM602 模块，选择 TCP 通信方式进行数据收发。参数配置如下：

- ZM602A：SoftAP 模式、TCP 服务器（本地地址 172.16.1.1，本地端口 4000）、115200 波特率；
- ZM602B：Station 模式、TCP 客户端（远程地址 172.16.1.1，远程端口 4000）、115200 波特率。

进行以上简单的配置以后，两个模块建立 Wi-Fi 连接，即可实现串口数据透传。



图2 串口数据透传

通信距离的影响因素

通信环境会极大地影响通信效果，主要原因是环境中的障碍物以及信号干扰的影响。本次测试我们选择了湖面开阔环境，尽可能减小障碍物和空间信号的干扰。天线的选择也会极大地影响通信效果。为了实现最优的通信效果，本次测试我们选用了增益为 5dBi 的棒状天线，天线通过 ZM602 的 IPEX 连接器进行连接。

我们首先将配置好参数的 ZM602A 模组放置在湖的一端，上电启动，模组会正常工作在 SoftAP 模式。然后我们沿着湖边，移动到 500 米开外的位置，将 ZM602B 模块放置好。给 ZM602B 上电，模组工作在 Station 模式，启动 Wi-Fi 扫描，我们可以找到 ZM602A 发出的热点信号，并进行连接。Wi-Fi 连接成功后，两个模组之间即可实现串口数据透传。在 500 米距离下，通信效果测试情况如下：

应用场景	产品型号	发送字节	接收字节	丢包率
湖面 500m	ZM602P2S31E	10364	10364	0%
		10184	10184	0%
		12844	12844	0%
		10868	10868	0%

图3 500米通信距离测试结果

测试结果显示，在 500 米距离下，ZM602 通信效果良好，没有任何丢包。

接着我们移动到湖的对岸，此时两者距离约为 850 米。在该位置按照同样的操作进行测试，通信效果测试情况如下：

通信场景	产品型号	发送字节	接收字节	丢包率
湖面 850m	ZM602P2S31E	10412	10412	0%
		10678	10678	0%
		10512	10512	0%
		10868	10868	0%

图4 通信距离测试结果

测试结果显示，在 850 米距离下，ZM602 仍然可以保持良好的通信效果，没有任何丢包。

由于环境的限制，我们没有进行更远距离的通信效果测试。从测试结果来看，ZM602 凭借其优异的射频性能，在开阔湖面环境下，850 米通信距离下通信效果良好，完全可以实现公里级 Wi-Fi 通信。



图5 通信环境图示



Wi-Fi+BLE二合一无线模块

ZM602

 点击购买

互联互通 ▼

【深度解析】 你知道我们的ePort模块经历了多少刀山火海 才能交付客户吗？

原创 研发部 ZLG 致远电子 2023-03-22 11:32:20

致远电子新推出的 ePort 系列产品从工业控制领域到其它各大领域正逐步被大量广泛使用，那您知道我们的 ePort 模块经历了多少刀山火海才能交付客户吗？

ePort系列产品简介



ePort 系列集成式 RJ45 插座模块（点击了解更多）是致远电子推出的一款高集成度、高可靠性产品，旨在简化以太网应用电路设计，降低客户电路开发的复杂度。使用 ePort 系列产品，不仅可以解决供应链问题和产品设计问题，还因其内部集成 PHY 芯片以及网络变压器，能够有效地缩小以太网电路的占板面积，其高可靠性更能够让广大用户放心使用。

ePort 系列产品可应用在电力监控、工业控制、医疗电子、仪器仪表、石油化工、轨道交通、智能家居等各大领域，且具有以下产品优点：

- 符合 IEEE 802.3/az/u/ab 标准；
- 分立元件模块化，简化电路设计；
- 采用 THT 式封装，易于更换；
- 提高 PCB 板空间利用率；
- 外壳材料符合 UL94 V-0 标准；
- 具有极低电磁辐射和高的抗电磁干扰性；
- 高低温特性好，满足工业级产品要求。

ePort系列产品的高可靠性

ePort 系列产品是满足工业级产品的要求，具有高可靠性的产品。对此，致远电子针对其使用环境和性能，相应做了以下的可靠性试验：电磁兼容性、环境适应性和其它可靠性试验。

1. 电磁兼容性

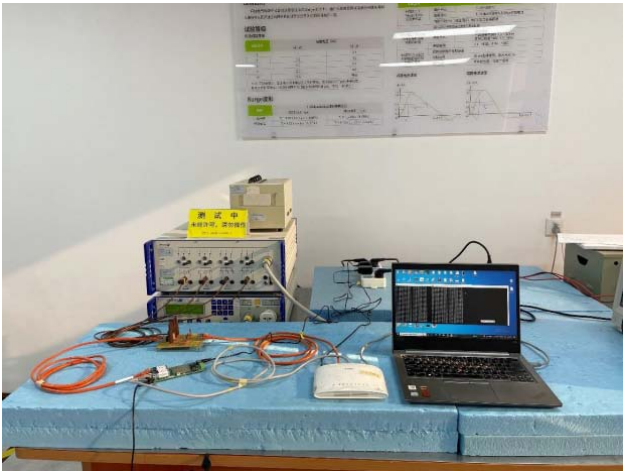
针对电磁兼容性，根据符合工业级产品的标准，相应做了静电放电抗扰度试验、电快速瞬变脉冲群抗扰度试验、雷击（浪涌）抗扰度试验、射频场感应的传导骚扰抗扰度试验、辐射骚扰试验等。



静电放电抗扰度试验



电快速瞬变脉冲群抗扰度试验



雷击（浪涌）抗扰度试验



射频场感应的传导骚扰抗扰度试验



辐射骚扰试验

通过相关试验，我们得到以下相应的测试结果：

产品型号	试验项目	测试标准	试验等级	性能判据
ePort-M	静电放电抗扰度试验	GB/T 17626.2-2018/IEC61000-4-2:2008	3级 ±6kV	A
			4级 ±8kV	B
	电快速瞬变脉冲群抗扰度试验	GB/T 17626.4-2018/IEC61000-4-4:2012	3级 ±1kV/5kHz	B
			3级 ±1kV/100kHz	B
			4级 ±2kV/5kHz	
			4级 ±2kV/100kHz	B
ePort-G	静电放电抗扰度试验	GB/T 17626.2-2018/IEC61000-4-2:2008	3级 ±6kV	B
			4级 ±8kV	B
	电快速瞬变脉冲群抗扰度试验	GB/T 17626.4-2018/IEC61000-4-4:2012	3级 ±1kV/5kHz	A
			3级 ±1kV/100kHz	B
			4级 ±2kV/5kHz	
			4级 ±2kV/100kHz	B
	雷击（浪涌）抗扰度试验	GB/T 17626.5-2019/IEC61000-4-5:2014	3级 ±4kV	A
	射频场感应的传导骚扰抗扰度试验	GB/T 17626.6-2017/IEC61000-4-6:2013	3级 10V	A
	30MHz~1000MHz 辐射骚扰	GB/T 9254-2008 (CISPR 22:2006, IDT)	A 级限值	A
	射频场感应的传导骚扰抗扰度试验	GB/T 17626.6-2017/IEC61000-4-6:2013	3级 10V	A

图1 电磁兼容性通过试验等级

2. 环境适应性

针对环境适应性，根据符合工业级产品的标准，相应做了低温启动与运行试验、高温启动与运行试验、温度变化试验和恒定温湿试验等。

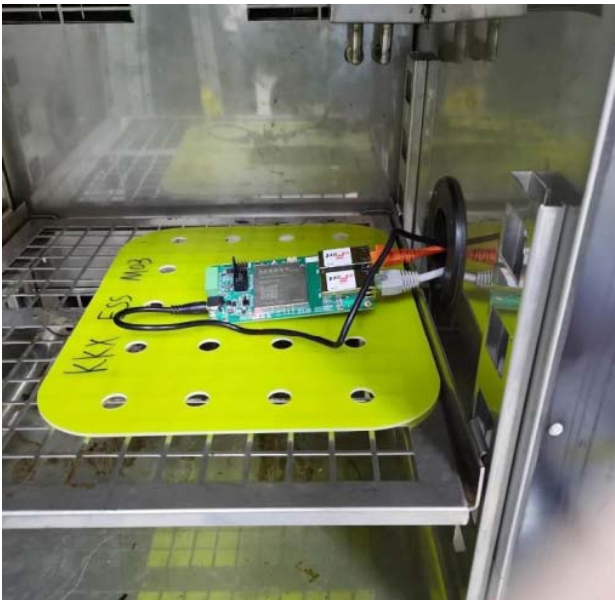


图2 环境试验

通过相关试验，我们得到以下相应的测试结果：

产品型号	试验项目	测试标准	试验温度湿度	性能判据
ePort-M	低温启动与运行试验	GB/T 2423.1-2008	-42℃	A
	高温启动与运行试验	GB/T 2423.2-2008	87℃	A
	恒定湿热试验	GB/T 2423.3-2016	+40℃/93%RH	A
	温度变化试验	GB/T 2423.22-2012	-42℃~+87℃	A
ePort-G	低温启动与运行试验	GB/T 2423.1-2008	-42℃	A
	高温启动与运行试验	GB/T 2423.2-2008	87℃	A
	恒定湿热试验	GB/T 2423.3-2016	+40℃/93%RH	A
	温度变化试验	GB/T 2423.22-2012	-42℃~+87℃	A

图3 环境适应性通过试验等级

更多 ePort 模块产品详情资料:

<https://www.zlg.cn/ethernet/down/down/id/307.html>

基于采用双核 64 位 A53 和四核 R5F 异构处理器的 EtherCAT 主站控制器 ZMC600E, 使用了 2 个 ePort-M 和 2 个 ePort-G 产品模块:

ePort系列产品应用案例

ePort 系列产品模块, 一经推出后, 其它各大产品都在广泛使用。包括有工控单板、医疗主板、EtherCAT 主站控制器等等。

基于采用致远电子 M6Y2C (A7 处理器) 工业控制核心板的工控单板 EPC6Y2C-LI, 使用了 2 个 ePort-M 产品模块:



图4 EPC6Y2C-LI工控单板

基于采用 M1126 (四核 32 位 A7 处理器) AI 核心板的工控单板 EPC1126-LI, 使用了 1 个 ePort-G 产品模块:

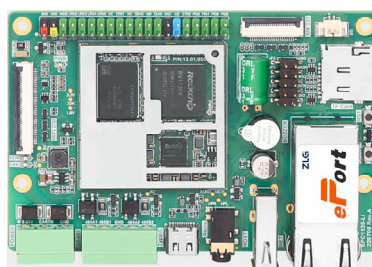


图5 EPC1126-LI工控单板

基于采用 RK3568 (四核 64 位 A55 高性能处理器) 的医疗主板, 使用了 1 个 ePort-M 和 1 个 ePort-G 产品模块:

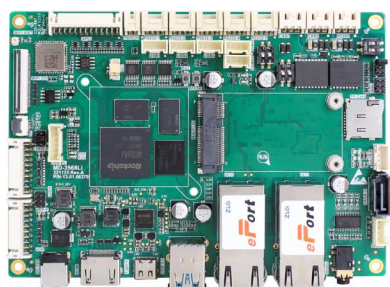


图6 RK3568医疗主板

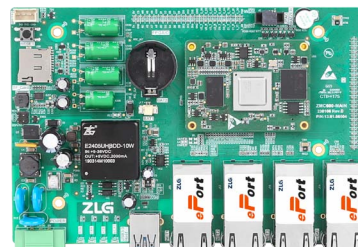


图7 ZMC600E控制器



ePort系列集成式
RJ45插座模块

 点击购买

【解决方案】充电桩从电源隔离到温度监测的一体化方案设计

原创 研发部 ZLG 致远电子 2023-03-08 11:31:28

在全球能源危机和环境危机的大背景下，电动汽车近年来行业增长迅速，充电桩也迎来了快速发展阶段。本文将提供一种充电桩从电源隔离到温度监测的一体化设计方案，有效提高充电桩的抗干扰能力、稳定性和安全性。

充电桩的介绍及使用场景

充电桩是用于电动汽车的充电设备，相当于汽车的加油机，从充电方式上可分为直流充电桩和交流充电桩，从使用环境上可分为室外充电桩和室内充电桩。充电桩通常由一个主控系统和多个子系统构成，子系统一般包括电源供电系统、充电枪、电表、显示屏、读卡模块、温度检测系统等，子系统与主控系统之间由 CAN、RS485 等隔离器件连接通信。

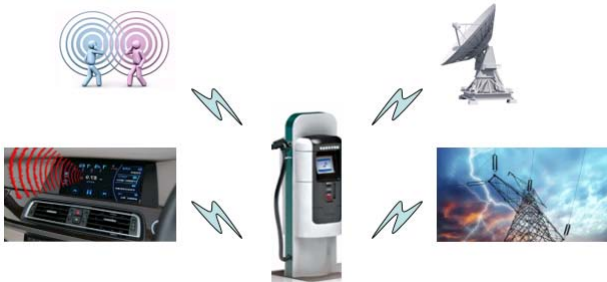


图1 充电桩受到外界干扰示例

一般来说，交流充电桩（慢充）是通过车载充电机为电动汽车提供电能，充电速度较慢，电流电压较小；直流充电桩（快充）则可以直接为电动汽车的蓄电池提供直流电能，充电速度快，电流电压较大。不论是直流桩还是交流桩，充电桩内部的 CAN、RS485、RS232 的通信距离都不会很长，所以节点数并不多，一般的隔离通信器件都能满足波特率、节点数量的需求。但是充电桩本身是一个强弱电结合的电子系统，每个子系统之间还会相互串扰和存在共模噪声影响，并且充电桩常常置于室外工作，易受雷击、浪涌等外界环境影响。所以选择具备更好 EMC 防护、可靠性更高的隔离通信方案尤为重要。



图2 过温引起火灾险情（图片源自网络，侵权）

另外，充电桩的使用环境也相对较严格，室外充电桩受自然环境影响更容易出现线路老化、漏电、短路等问题；室内充电桩大多位于地下车库内，空间狭小，车辆密集，容发生火灾事故。所以对充电桩的过温故障实时监测、预防火灾险情是充电桩必不可少的一项功能。

针对上述情况，我司提供了一种从电源隔离到温度监测一体化的设计方案，可以有效地解决充电桩的电磁干扰、高温烧损等问题，提高了充电桩的抗干扰能力、稳定性和安全性。

方案设计

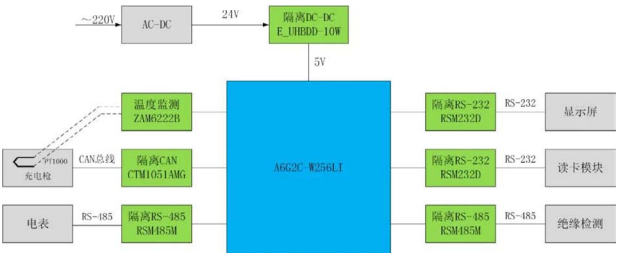
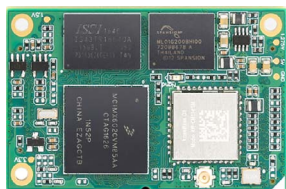


图3 充电桩整体方案框图

该方案使用了致远电子的 A6G2C 系列 Cortex-A7 核心板、CTM 系列 CAN 隔离、RSM 系列 RS485、RS232 隔离、ZAM 系列 PT1000 测温，可以完全满足上述要求。高隔离耐压性能、高精度温度采集、小体积模块与核心板一体化设计，帮助用户轻松快速实现功能，同时满足复杂的电磁环境、宽泛的工作温度范围，有效保障充电环境的安全与稳定。

方案使用模块及性能优点如下：

模块	性能优点	工作温度
A6G2C-W256LI	高集成、低功耗的 Cortex-A7 核心板，支持 Linux、AWorks IoT 操作系统，板载 Wi-Fi、蓝牙模块。	-30℃~+85℃
E_UHBD-10W	具备优异的 EMS 性能的电源隔离模块，高输入范围，自带输出过压、过流保护。	-40℃~+85℃
CTM1051AMG	超小体积封装 CAN 总线隔离模块，具备极低电磁辐射和高的抗电磁干扰性，有效防止 ESD、电火花等干扰。	-40℃~+105℃
RSM485M	超小体积封装 485 隔离通信模块，集成电源隔离和信号隔离，可支持 64 个节点连接。	-40℃~+105℃
RSM232D	具备极低电磁辐射和高的抗电磁干扰性的 232 隔离通信模块，超宽压输入电源供电。	-40℃~+85℃
ZAM6222B	PT1000 测温模块，兼容二线制、三线制 PT1000 测温，最高速率可达 10Hz/通道，具备断线检测功能。	-40℃~+85℃



A6G2C-W256LI 核心板可作为充电桩的主控板，板载高性能应用处理器，并集成 Wi-Fi、zigbee、LoRa 或 NFC 无线功能，搭载 Linux 或 AWorks IoT 操作系统。核心板上自带 Wi-Fi、蓝牙模块，可以实时远程收发数据，对充电桩的系统进行监控、升级，及时了解充电桩的全面情况。

A6G2C-W256LI 产品特性：

- 内核 Cortex-A7；
- 主频 528MHz；
- 内存 256M；
- 低功耗、高性能；
- Linux 或 AWorks IoT 操作系统；
- 工作环境温度 -30°C ~+85°C。



E_UHBDD-10W 电源隔离模块，用于主控板的隔离电源供电，可以提供优异的 EMS 性能。

E_UHBDD-10W 产品特性：

- 隔离电压 1500VDC；
- 输入电压范围 9V~75V；
- 效率高达 87%；
- 可持续短路，自恢复；
- 输出过压保护、输出过流保护；
- 工作环境温度 -40°C ~+85°C。



CTM1051AMG 隔离模块，用于充电枪和主控板之间的 CAN 总线隔离通信，可以有效防止充电枪与汽车的连接与断开时产生的 ESD、瞬态的电压变化、电火花等干扰，提供可靠的总线防护功能。

CTM1051AMG 产品特性：

- 隔离电压 2500VDC；
- 支持 CAN FD，最高速率 5Mbps；
- 单网络最多可连接 110 个节点；
- 超小体积，仅为一般产品的 40%；
- 具备极低电磁辐射和高的抗电磁干扰性；
- 工作环境温度 -40°C ~+105°C。



RSM485M 隔离模块，用于电表、绝缘检测模块与主控板之间的隔离通信，实现稳定电量的统计与计费，漏电检测。

RSM485M 产品特性：

- 隔离电压 2500VDC；
- 超小体积；
- 最多可连接 64 个节点；
- 电磁辐射 EME 极低；
- 电磁干扰 EMS 极高；
- 集成电源隔离和信号隔离；
- 工作环境温度 -40°C ~+105°C。



RSM232D 隔离模块，用于显示系统、读卡模块与主控板之间的隔离通信，完成身份识别、扣费等功能。

RSM232D 产品特性：

- 隔离电压 2500VDC；
- 3.15V~5.25V 超宽压输入电源供电；
- 电磁辐射 EME 极低；
- 电磁干扰 EMS 极高；
- 工作环境温度 -40°C ~+85°C。
- ZAM6222B 测温模块，用于充电枪接头的温度监测。充电接头是充电桩与电动汽车的充电连接点，易发生过温现象。ZAM6222B 监测该点的温度，配合主控系统及电子锁模块，可以预防充电过程中过压、短路、漏电等故障引起充电枪的高温损坏，同时也可以作为充电桩内部的实时温度监测模块，起到火灾预警作用。



- ZAM6222B 产品特性：
- 两通道 PT1000 热电阻测量；
- -100°C ~300°C 测温范围；
- 0.02%±0.1°C 测温误差（三线制）；
- 0.1%±0.2°C 测温误差（二线制）；
- 工作环境温度 -40°C ~+85°C；
- 断线检测功能。

【技术分享】 电源模块的本手与应用妙手

原创 研发部 ZLG 致远电子 2023-03-13 12:03:21

现如今电是不可或缺的能源，给电器设备注入了电，设备才有了活力，有电的设备就需要用到电源模块，而一个好的电源模块是本手，简单外围电路能够使模块进一步安全稳定的运行是应用中的妙手？

电源模块抗浪涌的本手

电源模块应用过程中，部分客户会发现电源模块用着会“无缘无故”就坏了，特别是在室外应用场景和室内恶劣的工业电网场景，这种情况很可能是因为浪涌电压的产生而损坏。

浪涌电压产生有雷电，电网上大型负荷接通或断开；浪涌电压的产生是不可控的，浪涌电压破坏对象首当其冲是电源模块，然后是电源模块后端电路。模块电源应用过程中，抗浪涌的本手是提升模块本身的抗浪涌能力，模块内部选用抗浪涌能力强的器件，浪涌能量泄放后，模块不受损。



图1 雷电产生浪涌

电源模块抗浪涌应用妙手

由于客户的应用场合众多，有些客户的应用场景模块的本手受空间局限性抗浪涌能力没法满足需求，需要通过妙手外接电路保护模块，通过外接器件可以提供浪涌能量泄放路径，保护了后端的模块电源。加了外围电路的模块电源能够在室外环境或室内恶劣环境应用。

常用的几种抗浪涌器件：压敏电阻、气体放电管、TVS 管。

压敏电阻的选型最重要的几个参数为：最大允许电压、最大钳位电压、能承受的浪涌电流。

首先应保证压敏电阻最大允许电压大于电源输出电压的最大值；其次应保证最大钳位电压不会超过后级电路所允许的最大浪涌电压；最后应保证流过压敏电阻的浪涌电流不会超过其能承受的浪涌电流。

气体放电管属于开关型器件，常和压敏电阻串并联使用 TVS 与压敏电阻和气体放电管相比，响应速度更快，耐浪涌冲击能力较差，属于钳位器件，钳位电压更稳。可以和压敏电阻、气体放电管配合使用，作为分级防护释放浪涌能量。

如下图是几种常用的抗浪涌外接电路图，图2中 D1 是肖特基二极管，此电路是模块在无外围电路情况下 GND 端注入浪涌能量易损坏时用到。

图2 图3 的外接电路缺点明显，成本不如图4 低，需要用图2 图3 这

种外接电路的模块，模块电源本手抗浪涌能力是稍差的，只能通过外围堆积器件实现更高的抗浪涌能力。我们的宽压 E-UHBDD-10W/6W 用到图 4 的外接电路就能实现很好的抗浪涌能力。图5是按图4 的电路参数接线测试结果，按 IEC/EN 61000-4-5 ±2KV line to line 标准测试。

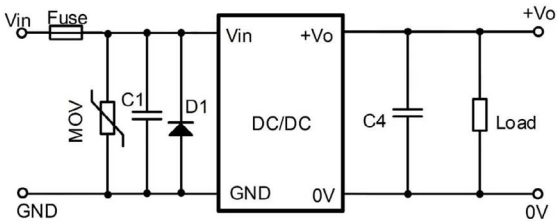


图2 外接抗浪涌电路图

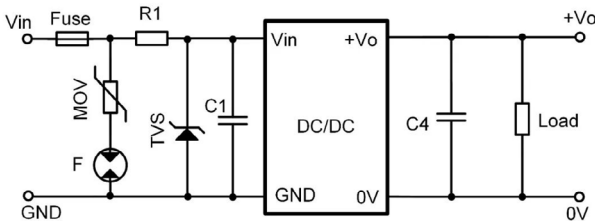


图3 外接抗浪涌电路图

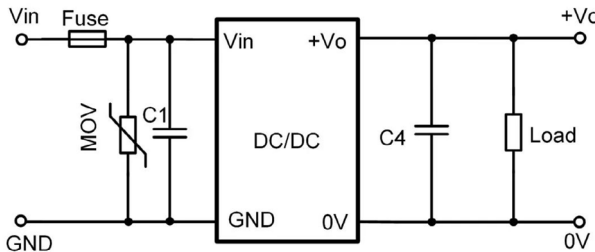


图4 外接抗浪涌电路图

型号	Vin: 24VDC	Vin: 48VDC
FUSE	依照客户实际输入电流选择	
MOV	14D560K	14D101K
C1	330μF/50V	330μF/100V

表 1 E-UHBDD-10W推荐电路参数



图5 测试浪涌图

产品推荐



**6W宽压隔离
DC-DC电源模块**

 [点击购买](#)



**10W宽压隔离
DC-DC电源模块**

 [点击购买](#)

【产品应用】 小器件助力舞台灯光可靠连接

原创 研发部 ZLG 致远电子 2023-03-17 11:31:23

绚丽夺目的灯光表演，能给我们带来不一样的视觉感受，如广州塔的灯光秀、演唱会的灯光特效等。宏大的灯光效果需要众多灯具的精妙配合，而这都离不开一个小小的器件——RS-485收发器。

大规模网络化灯光控制广泛采用 DMX512 协议，此协议由美国舞台灯光协会提出并最终成为国际标准，采用该标准能够对舞台、剧场等场所的灯具进行数字控制，以实现多种设备的实时配合，达到设想的光影特效。DMX512 协议适用于一主多从的控制网络，其物理层采用的是 RS-485 总线收发器，通过 485 总线实现灯光控制器与灯具的连接。

案例分享

一个灯光控制网络可连接多个灯具，多的可达几十个。实际使用中，在同一网络的灯具可能仅需使用几台，而不需要使用的灯具不通电。在这种场景中，有可能会因 RS-485 收发器原因造成灯具无法控制的情况。

我司一客户曾遇到类似异常,其灯光网络如图 1 所示,由一台灯光控制器控制多台灯具,灯具执行摇头、调光等动作,一个网络最多需要连接 32 台灯具。测试发现,当有个别灯具不通电时,灯光控制器发出的控制信号无法被灯具正确接收。



图1 灯光控制组网示意图

经分析,客户所使用的 RS-485 收发器在断电后,仍能通过 485 总线获得少量的能量,使芯片保持在一个非完全关闭状态,并拉低了总线信号幅度导致数据错误,最终造成控制命令下发失败,表现出灯光控制台无法正确控制灯具的现象。

实测波形如图 2, CH4 为灯具断电后, 485 芯片电源引脚电压波形, 可见电压缓慢下降, 并稳定在 1.45V 左右。CH1 为 485 总线 AB 差分波形, 可见随着 CH4 电压下降, AB 差分信号幅值也随之下降, 从而导致异常出现。

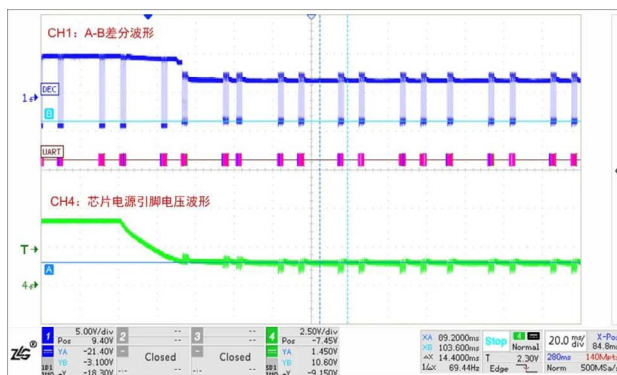


图2 个别灯具断电后总线异常波形

通过更换 RS-485 收发器型号后,异常现象消失。经确认,更换后的 485 收发器芯片在灯具断电后完全关闭,不影响总线通讯。实测波形如图 3,当灯具断电后,CH4 芯片电源电压逐步下降到 0,CH1 的 AB 差分信号也不受影响。

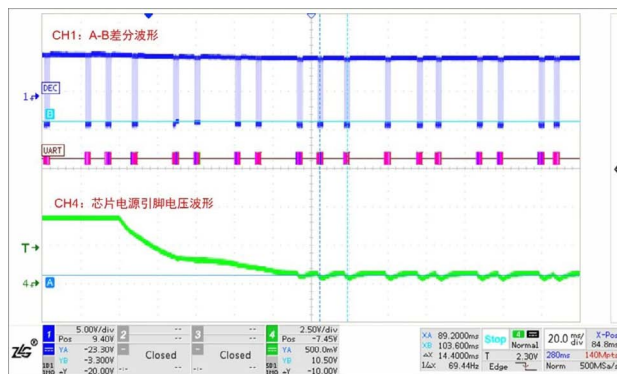
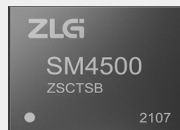


图3 个别灯具断电后总线正常波形

致远电子的 RS-485 收发器经过多年的耕耘，现已广泛应用于国内灯光行业产品中，赋能灯具可靠连接。致远电子推出的 SM4500 贴片 RS-485 收发器，采用了 SiP 塑封工艺，体积小，控制延时低，工作速率高，且进行针对性优化，通过了多节点组网断电测试，非常适合应用于舞台灯光行业，可为灯光的可靠控制保驾护航。



SM系列 全隔离RS-485收发芯片



【技术分享】 电源模块应用疑惑与解惑

原创 研发部 ZLG 致远电子 2023-03-31 11:32:25

在电源模块选型应用中，曾经是否会因选用隔离电源模块还是非隔离电源模块而难做抉择？本文可以让你快速做出判断，选择合理的供电方案，能够保证系统安全平稳运行。

电源模块应用疑惑

客户在电源模块应用过程中曾经出现过这样的应用场景，如下图1所示，客户选了一个带隔离变压器的电源模块，然后把输入输出的地连在一起使用，可以正常工作，不影响使用，但客户考虑长久使用的可靠性进行了技术交流沟通。

客户这个应用是隔离电源用成了非隔离电源，这样应用的结果是隔离电源和非隔离电源的缺点都占有了，而非隔离电源应用的优点又没沾上，

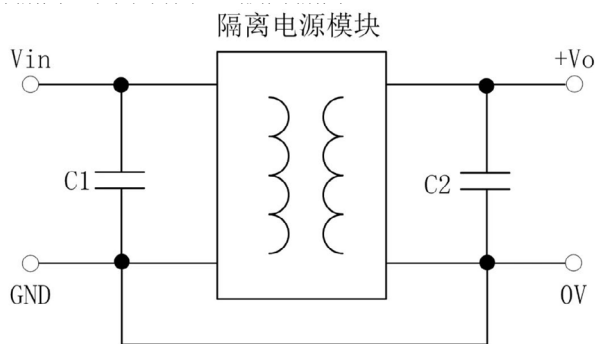


图1 客户的应用电路图

电源模块应用解惑

客户上面的应用就是涉及隔离和非隔离电源模块的问题，要解决客户的应用疑惑，首先需要对电源模块的隔离和非隔离有所了解，下面两个电路图可以直观的做出隔离和非隔离电路的辨识，图2是隔离电路，图3是非隔离电路。

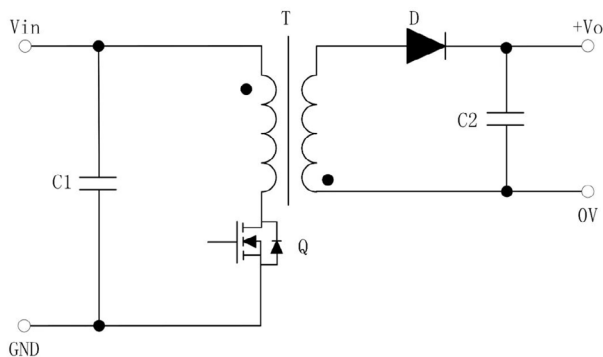


图2 常见隔离电路图

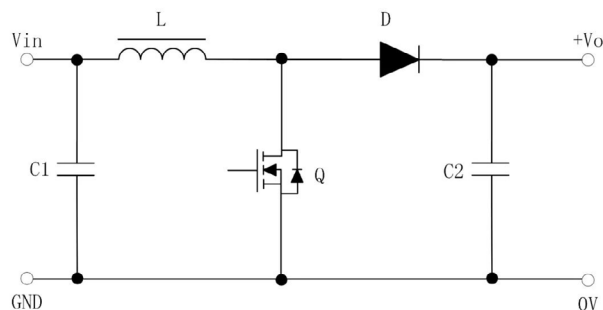


图3 常见非隔离电路图

隔离电源的特点是输入输出通过隔离器件隔离，输入到输出的能量是隔离变压器进行传递，信号传递是通过光电耦合器；输入输出间没有直接的电气连接，存在高阻抗的隔离。隔离电源的优点是抗干扰能力强，安全性好；缺点是转换效率低，体积大。

非隔离电源的特点是输入输出共地，存在直接的电气连接。非隔离电源的优点是转换效率高，体积可以做小；缺点是因为输入输出共地，安全性较差，抗干扰能力差。

隔离电源和非隔离电源涉及安全和抗干扰性，所以应用场合还有些许差异。

1. 隔离电源的应用场景

隔离电源的安全性高，所以适用于涉及人身安全的需防触电的场合，比如医疗行业电源，高压转低压场合；需求电源前端损坏，能保护负载电路的场合。

需要提升抗干扰的场合，比如系统前端电源，如图4所示。远程工业通信场合，每个通信节点用隔离电源供电可以降低因导线耦合的干扰。



图4 隔离电源应用图

2. 非隔离电源的应用场景

非隔离产品效率高，抗干扰能力差，适合纯净的用电环境和损耗需求较低的场景，比如家庭灯具供电场合，由电池供电的仪器仪表需求空载损耗小，长续航的场合。系统的后级电路，给后级的系统芯片供电。

很多场合下隔离电源和非隔离电源是一起用的，这种场合隔离电源用在前端电路，非隔离电源用在后级供电电路，如参考下图 5，这是一个工控板的隔离电源和非隔离电源的一个应用场景。

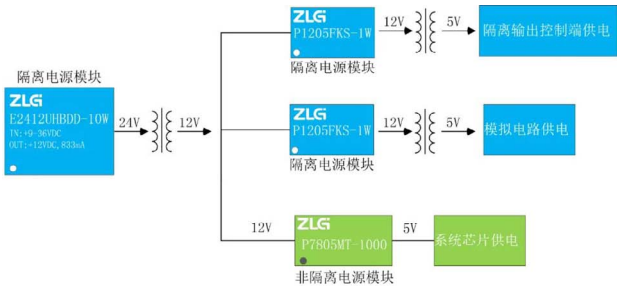


图5 工控板电源模块应用场景框图

有些要求不高的场合，可用非隔离电源也可以用隔离电源，这种情况就要结合实际，选用最合适的方案。

产品系列	E-UHBDD-10W	P-FKS-1W	P7805MT-1000
隔离耐压	1500VDC	3000VDC	--
效率	87%	83%	90%
保护功能	输入欠压保护，输出过压、过流、短路保护	输出短路保护	输出短路保护

表1 电源选型表

产品推荐



**10W宽压隔离
DC-DC电源模块**

 点击购买



**1W定压隔离
DC-DC电源模块**

 点击购买

感知控制 ▼

【解决方案】 致远电子隔离测温技术为精密恒温箱赋能

原创 研发部 ZLG 致远电子 2023-03-27 11:52:25

工业领域对温度的测量精度要求较高，本文介绍一种恒温箱应用精密温度传感电路的设计方案：用热电阻 PT100 温度传感器和致远的隔离测温模块可实现高精度的温度采集，对环境干扰不敏感。

恒温箱的介绍及使用场景

恒温箱可对产品的物理或其他相关特性进行环境模拟测试，以判断产品的性能，是否仍然能够符合预定要求，以便供产品设计、鉴定及出厂检验用。恒温箱一般由箱体、发热体、制冷器、测温仪或温度采集器、温控器、空气循环系统和信号系统等组成，特殊用途的恒温箱还有加湿器、压缩机、防爆装置等。

例如在实验室应用中，为了得到更加准确的实验数据，对于恒温实验环境要求严格。所以针对实验室来说，恒温箱的作用显得相当重要，对于实验室的研究过程以及研究结果将产生很大的影响。

同时在工业生产过程中对于恒温箱的要求也相对更加严格，比如质量的可靠性、恒温箱的维护管理费用、恒温箱的本身价位、温度采集及控温精度等，这些对于工业上的批量生产产品的企业都是相当重要的。随着恒温箱的技术进步，以及用户对智能化的需求，这种设备对温度的控制要求不仅越来越高，而且也希望具备远程传输及控制功能。

方案设计

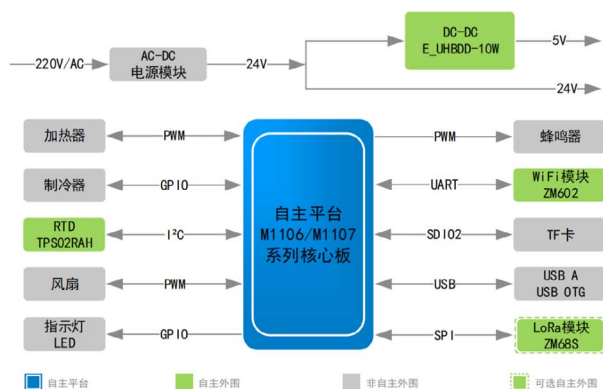


图1 恒温箱整体方案框图

该方案使用了致远的 M1106/M1107 系列核心板、DC-DC 隔离电源系列 E_UHBD0-10W、ZM602 WiFi 模块、LoRa 射频模块 ZM68S、TPS02RAH 双通道热电阻隔离测温模块，可以实现恒温箱内完全电气隔离，见图 2。

方案使用模块及性能优点如下：

模块	M1106/M1107 系列	E_UHBD0-10W	ZM602	ZM68S	TPS02RAH
性能优点	高集成、低功耗的 ARM926 核心板，支持 Linux 操作系统。	具备优异的 EMS 性能，电源隔离模块，高输入范围，自带输出过压、过流保护。	ZM602 支持 IEEE 802.11b/g/n 协议，支持 UART 串口透明传输，用户可实现设备的快速联网。	ZM68S 具备小体积、低功耗、高稳定性、远距离传输等特点，是一款低成本、高性能的 LoRa 射频模块。	PT100 隔离测温模块，三线制 PT100 测温，高隔离耐压。
工作温度	-40℃~+80℃	-40℃~+85℃	-30℃~+85℃	-40℃~+80℃	-40℃~+85℃

图2 方案使用模块及性能优点



M1106/M1107 系列核心板可作为恒温箱的主控板，板载低功耗应用处理器，搭载 Linux 操作系统。核心板上搭配外围的 Wi-Fi、LoRa，可以实时远程收发数据，对恒温箱进行监控、升级，及时了解温度曲线的全面情况。

M1106/M1107 系列产品特性：

- 内核 ARM926EJ-S；
- 主频 804MHz；
- 内存 64M DDR2；
- 低功耗、高性能；
- 工作环境温度 -40℃ ~ +80℃。



E_UHBDD-10W隔离电源模块

E_UHBDD-10W 隔离电源模块，用于主控板的隔离电源供电，可以提供优异的 EMS 性能。

E_UHBDD-10W 产品特性：

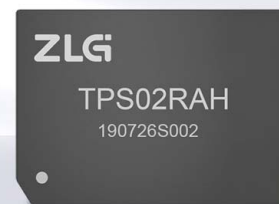
- 隔离电压 1500VDC；
- 输入电压范围 9V~36V；
- 效率高达 87%；
- 可持续短路，自恢复；
- 输出过压保护、输出过流保护；
- 工作环境温度 -40°C ~+85°C。

ZLG

双通道热电阻隔离测温模块

TPS02RAH

- 2通道
- 隔离耐压4000Vrms
- 高精度



TPS02RAH 双通道热电阻隔离测温模块，只需接入 PT100 热电阻，即可完成温度的采集，采用标准 IIC 接口直接输出以°C为单位的温度数据。模块的测温精度高达 0.02%±0.1°C，分辨率高达 0.01°C。模块内置电气隔离，保障测量结果不受干扰。采用超小的体积设计，更易于集成到各种测温设备中。

TPS02RAH 产品特性：

- 两通道 PT100 热电阻测量
- 0.01°C测温分辨率；
- 0.02%±0.1°C测温误差；
- 10ppm±0.1°C温漂；
- I2C 通信接口；
- 隔离电压 4000Vrms；
- 3.3V 供电电压；
- DIP16 小体积封装；
- 温度报警输出；
- 工作环境温度 -40°C ~+85°C。

原创 研发部 ZLG 致远电子 2023-03-27 11:52:25

看似乖巧的看门狗，也有比较调皮的一面，在烧录无法喂狗的特殊情况下仍会毫不留情的复位芯片，造成烧录失败，该如何避免这种情况的发生吗？

初识看门狗

嵌入式产品设备在电源瞬变、电磁干扰、静电放电、极端温湿度等复杂环境的干扰下可能会造成程序异常，从而使系统陷入瘫痪状态，这时候我们就希望设备能够自动复位，而不需要人工干预，看门狗定时器可以帮助捕获和重置已经“失控”的设备，实现无人值守，提高系统可靠性。

看门狗又叫 watchdog timer，是一个定时器电路，一般有一个输入端用来喂狗，一个输出端 RST 接到芯片的复位引脚，芯片正常工作的时候，每隔一段时间输出一个信号到喂狗端，给看门狗清零，如果超过规定的时间不喂狗，看门狗溢出，就会给出一个复位信号，拉低 RST 引脚使得芯片复位。

ZL6300 是广州致远微电子有限公司自行设计的一款集欠压监测复位、手动复位和超时复位功能的看门狗芯片，具有以下特性：

- 上电复位脉冲 200mS;
- 支持手动复位;
- 看门狗定时器溢出周期 1.7S;
- 低功耗仅 9uA;
- 欠压复位。

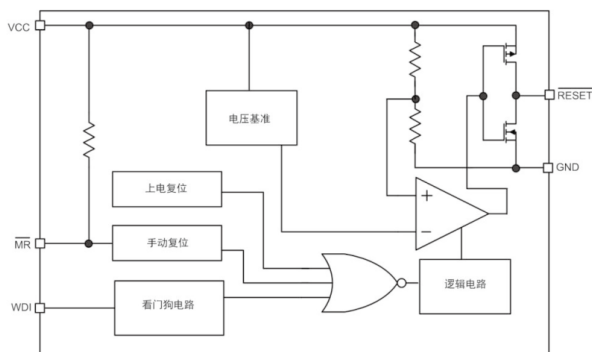
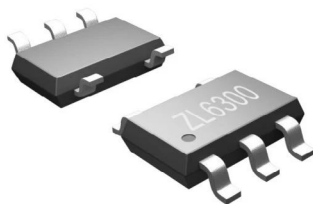


图1 ZL6300内部结构



应用电路

通过之前对看门狗的初步认识，好的看门狗设计可以大大提升产品的稳定可靠性，我司基于该理念，核心板上均板载独立看门狗，为适应各种复杂环境保驾护航。下面以我司性能卓越的 M3568 核心板为例，看下它如何灵活设计看门狗电路，对应的原理图如下图 2 所示，看门狗芯片选型 ZL6300，支持手动按键复位和看门狗溢出复位，当系统正常运行时从芯片上电启动到加载看门狗喂狗驱动所需时长约为 1S 左右，ZL6300 溢出周期为 1.7S，可以在看门狗溢出周期内及时喂狗清除定时器。

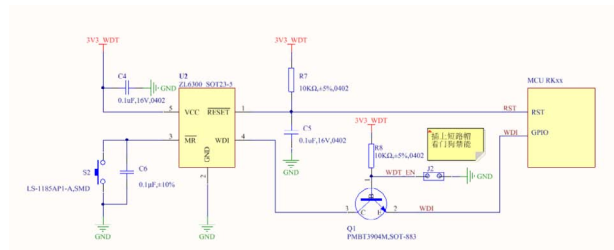


图2 看门狗电路

当芯片处于烧录状态时因内核停止工作，喂狗程序不能正常运行，不能及时喂狗，看门狗溢出拉低芯片复位引脚，导致程序下载失败。为应对这种特殊情况，预留了跳线冒选项，插上短路帽即可禁用看门狗，当 J2 插上短路帽时，WDO_EN 被拉至低电平，Q1 三级管处于关断状态，WDI 引脚处于高阻态，当 WDI 引脚检测到它处于高阻抗状态时将产生自己的 WDI 脉冲。对应的喂狗波形如下图 3 所示。

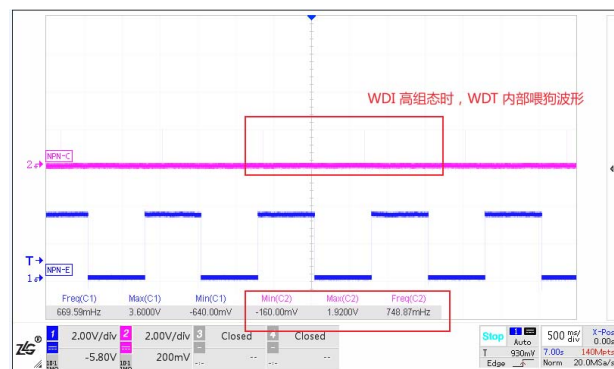


图3 看门狗禁能时喂狗波形

当烧录完成后拔掉跳线帽，Q3 处于导通状态，芯片产生的喂狗脉冲 WDI_GPIO 可通过 Q3 传递到 WDI 引脚，实现正常喂狗，对应的喂狗波形如下图所示。

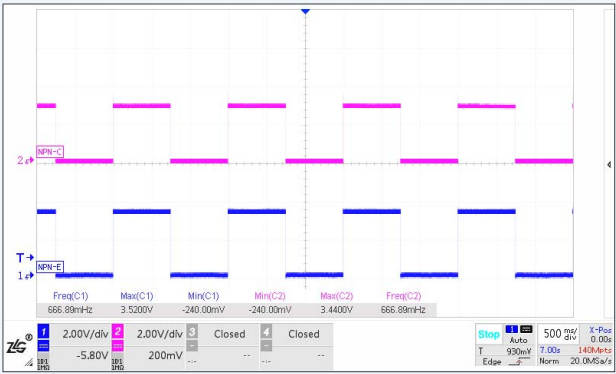


图4 看门狗使能时喂狗波形

总结

- 下面简单汇总看门狗电路设计注意事项
- 看门狗复位脉冲时长要大于 CPU 的最少复位时间；
 - 看门狗溢出周期要大于系统启动到加载看门狗驱动时长，并留有一定的余量；
 - 充分利用 WDI 引脚在高阻态时看门狗定时器功能禁用；
 - 避免通过使能看门狗的电源来控制看门狗工作与否，否则会造成系统一直重启，看门狗芯片上电后会输出一个时长200mS的POR信号。

【产品应用】 ZL6300在加载时间过长系统的巧妙应用

原创 研发部 ZLG 致远电子 2023-03-28 11:44:05

采用单片机为核心的控制系统常常会受到各种干扰的影响，打乱正常程序的运行，这时就需要我司产品登场了，ZL6300 集成了欠压监测复位、手动复位和超时复位等功能。本文将介绍 ZL6300 的巧妙应用！

低压复位

产品工作过程中，保持电源的稳定尤其重要，电源电压的不稳定会导致一些难以预测的问题，比如说样品测试的时候没有问题，量产时就会出现。这样问题排查起来也麻烦，而我司产品 ZL6300 能时刻精准的监控系统工作电压，无论是电源故障，还是偶发性掉电 ($t_{RD}>10\mu s$)，ZL6300 都能监控到。



程序“跑飞”复位

针对程序“跑飞”，ZL6300 也有很好的应对方案：由微处理器输出的重复喂狗信号作用于 ZL6300 看门狗的 WDI 引脚。当 WDI 输入在典型 1.7s 内没有从低到高或从高到低的转换时，看门狗定时器功能强制把 RESET 信号拉低复位。当 RESET 为高电平复位无效时，看门狗超时计时开始。如果在看门狗超时之前在 WDI 输入引脚上发生电平转换，看门狗定时器将复位并再次开始进行计时，如此重复，一旦程序“跑飞”，微处理器没有喂狗，ZL6300 将产生复位信号使微处理器复位，有效地在程序“跑飞”后系统自动复位，使系统自动回归正常工作。

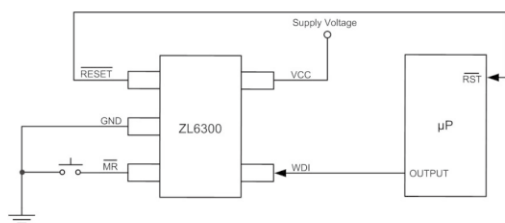


图1 ZL6300常规应用

应用于初始化加载时间长的系统

利用 ZL6300 的 WDI 引脚确实可以防止系统死机。但有一种特殊情况：就是对于初始化加载时间长的系统，加载时无法喂狗，超过 ZL6300 看门狗定时器溢出时间，系统就会复位。那么是不是就不能用 ZL6300 了？答案是否定的，下面我们就来讨论下对于初始化加载时间长的系统的应用示例：

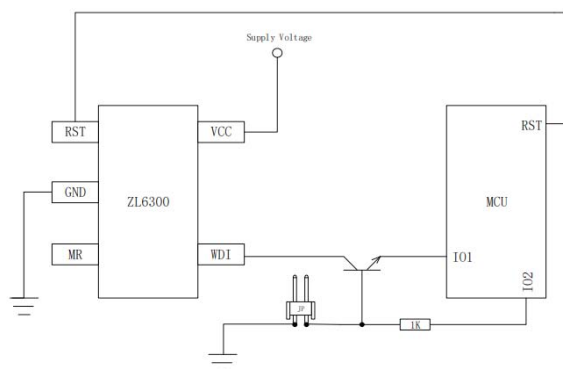


图2 ZL6300应用于加载时间长的系统

示例说明：JP 短接时，三极管截止，WDI 引脚处于高阻态，ZL6300 内部自动喂狗，便于烧录调试时，ZL6300 不会输出复位信号，顺利完成系统固件烧录。当系统运行时，JP 开路，三极管的状态受控于 IO2，系统加载时间过长时，在 MCU 加载系统前就需要将 IO2 拉低，此时三极管截止，WDI 引脚也处于高阻态，IO1 不需要喂狗，ZL6300 也不会输出复位信号。等系统加载完成后，先把 IO2 拉高，三极管导通，再用 IO1 周期性喂狗，系统进入正常运行状态。这样就可以应用于加载时间长的系统了。

有关 ZL6300 的详细资料，请到官网地址下载：

<https://www.zlg.cn/microelectronics/down/down/id/296.html>

2023/3 总第3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