

ZLG 致远电子

微文摘

ZLG MICRO DIGEST

2023/11 第11期

月刊



小型高性能边缘计算网关

EPCM3568A-LI

- 四核64位Cortex-A55内核处理器RK3568J
- 内置NPU，支持搭载轻量级运算
- 支持WiFi，Bluetooth，LoRa，ZigBee无线通讯
- 丰富的外设接口：双千兆以太网，USB2.0*2，USB3.0*1，RS485*2，HDMI*1，RS232*1，CAN*1，DI*2，DO*2，耳机口*1
- 支持TF卡、U盘存储
- 金属机身，无风扇设计
- 预安装Debian开源系统,支持Ubuntu系统
- 支持docker容器技术
- 支持Modbus TCP/RTU、OPC UA等工业协议
- 支持致远ZWS云、阿里云、腾讯云等多种云平台



规格参数

类别	名称	主要参数
最小系统	处理器	RK3568, 四核 64 位 Cortex®-A55, 主频 2.0GHz
	内存	2GB DDR4
	存储	8GB eMMC
电源	工作电压	DC 9V-24V
有线通讯	RS485	2 路
	RS232	1 路
	CAN	1 路隔离
	以太网	2 路, 10/100/1000Mbps (RJ45)
无线通讯	WiFi	1 路, 支持 2.4G
	Bluetooth	1 路, WiFi 模块支持
扩展存储	TF 卡	1 路, 标准 TF 卡座
USB 接口	USB2.0	2 路
	USB3.0	1 路
显示接口	HDMI	1 路, HDMI2.0
音频接口	Audio	1*Headphone, 3.5mm 音频插座

类别	名称	主要参数
其他	DI	2 路无源型, 均带光电隔离
	DO	2 路输出, 光耦隔离, 用户侧需接外部上拉
	miniPCle 插座	可支持扩展 LoRa、ZigBee 模块
	LED 指示灯	4 路, 其中 1 路可编程
工作环境	工作温度	-40° C ~ +60° C
	存储温度	-40° C ~ +60° C
	相对湿度	5% ~ 95% (无凝露)
软件规格	系统	默认 Debian, 支持 Ubuntu
	可支持编程框架	TensorFlow/Caffe/Darknet 等
	网络服务	支持 DHCP 服务器 / 客户端、动态域名 DDNS、SSH、Telnet、HTTPS、TFTP
	数据安全	OpenVPN、OpenSSL、WireGuard 等
	支持多种云平台	致远 ZWS 云、阿里云、腾讯云等
物理规格	支持多种工业协议	Modbus TCP/RTU 协议、OPC UA 协议等
	安装方式	壁挂式, 导轨式
	机械尺寸	110x73x44.5mm

CONTENTS

目录

技术平台

EsDA 平台

- 【技术分享】AWorksLP 应用笔记：重定向 printf 函数 04
- 【EsDA 应用】快速实现串口转 HTTP 请求 07
- 【技术分享】Modbus RTU Master 转 MQTT 14
- 【从 0 开始创建 AWTk 应用程序】开发及调试环境搭建 19
- 【从 0 开始创建 AWTk 应用程序】创建应用程序并在模拟器运行 21

ZWS 云平台

- 【产品应用】基于 ZWS 云低代码平台的智慧工厂监控大屏应用 24
- 【产品应用】基于 ZWS 云对 LoRa 网关的远程运维管理 25
- 【产品应用】ZWS 工商储能管理云平台，实现智能化运营，拓展收益 26
- 【产品应用】基于 ZWS 云对 LoRa 网关的远程升级 28
- 【产品应用】ZWS 工商储能云平台，实现智能化削峰填谷 29
- 【产品应用】如何将工业设备快速接入到 ZWS 物联网云平台 31

边缘计算

工控板 / 工控机

- 【新品发布】小型高性能边缘计算网关，小身板，大能量！ 32
- 【新品发布】ZLG 致远电子 PCIe EtherCAT 通讯卡产品正式发布！ 34
- 【技术分享】小型边缘计算网关设备如何快速上云 36

行业控制器

- 【产品应用】基于 EPCM3568A-LI 的 LoRa 快速使用指南来了 39

互联互通

接口与协议转换

- 【产品应用】快速实现 PLC 的 Modbus 数据转换为 CAN/CAN FD 数据？ 42

感知控制

电源与隔离

- 【新品发布】宽电压输入稳压电源模块 E48_UHFCS-3W 系列 45
- 【技术分享】振动场合的电源模块该如何选型？ 46
- 【产品应用】SM 系列全隔离 RS-485 收发芯片“全家福”来啦！ 47

数据采集

- 【技术分享】三线制 PT100 测温容易忽略的设计细节 48
- 【产品应用】多通道测温还需要搭电路？一个模块全搞定！ 49

【技术分享】

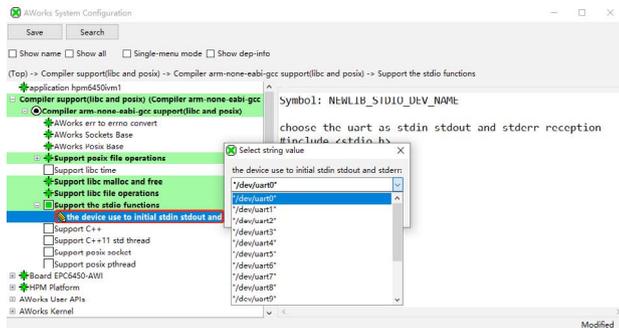
AWorksLP应用笔记：重定向printf函数

ZLG 致远电子 2023-11-24 11:38:09

printf 函数作为标准库定义的格式化输出方式，本文将介绍其在 AWorksLP 下默认适配以及重映射至热拔插设备端口的实现。

默认适配

AWorksLP 中默认已经对 printf 函数完成相关适配工作，且默认被适配在 UART 设备。用户可以在图形化配置界面中使能 support the stdio functions，并选择期望 UART 设备进行输出，具体配置如下图所示。

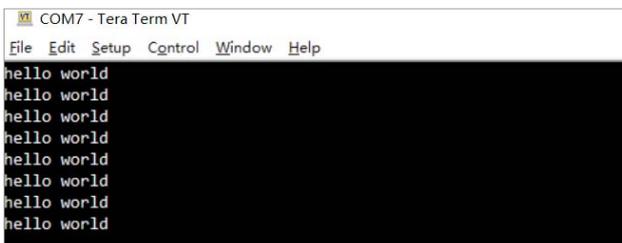


注：若用户未使能 stdio function 时，调用 printf 函数时，将不会有任何输出。

本文将使用 EPC6450-AWI 平台，选择标有丝印为 DUART 的调试串口（UART0 设备）进行 printf 功能演示测试。将 TTL 转 USB 串口模块的 TXD 与板子的 RXD 丝印连接，RXD 与板子的 TXD 丝印，将另一端的 USB 口接入电脑。

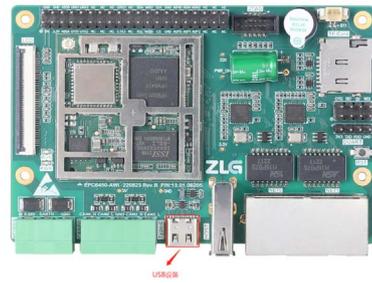


启动串口调试助手，搜索并打开串口模块的设备端口号后，在工程中调用 printf 函数，根据下图可知，printf 函数适配 UART0 设备成功。



重定向至其他设备

嵌入式的诸多应用在 UART 设备资源受限的情况下，可能存在将 printf 函数重定向到其他设备需求。为此，笔者将以 EPC6450-AWI 平台的 USB 串口设备为例进行说明。



1. 实施步骤

与 UART 设备不同，USB 设备为动态设备，因此重定向 printf 函数时，需要注意以下几个关键步骤：

1.1 支持NEWLIB标准库函数

由于 AWorksLP 中利用 posix file 相关操作接口对 printf 函数进行适配，故在重映射端口时，需将 support libc file operations 使能，并取消默认选择 UART 设备作为 printf 函数的适配，具体如下图所示。



1.2 检测动态设备

USB 设备为动态设备，因此需要持续检测设备的是否存在情况。可通过初始化一个动态设备检测任务，对设备的是否存在情况进行周期性检测。

```
while true:
    access (device)
    delay()
```

1.3 关联标准文件流

在检测到 USB 设备存在时，仅需将设备与标准文件流（stdio 中的 stdin、stdout、stderr，且在 C 库中被假定为交互设备，并约定了这些设备的文件描述符依次为 0、1、2）关联起来。故在使用时，我们仅需将描述符 0、1、2 与 USB 串口设备即可，其伪代码如下所示。

```
while true:
    if access (device):
        0 = open (device)
        duplicate 1 to 0
        duplicate 2 to 0
        delay()
```

1.4 清理文件描述符

检测到 USB 设备不存在时，需及时取消设备与标准文件流的关联。即根据设备的打开情况，对文件描述符进行清理，以便之后重新关联标准文件流。

```
while true:
    if access (device):
        0 = open (device)
        duplicate 1 to 0
        duplicate 2 to 0
    else:
        close (device)
        delay()
```

2. 基础配置

在 EPC6450-AWI 平台标有丝印为 Type-C 的接口处，插上 Type-C 线，将 Type-C 线的另一端 USB 口连接电脑。并在图形化配置界面，将 USB 设备选择为 CDC 串口设备。



3. 简单示例

在 EPC6450-AWI 平台标有丝印为 Type-C 的接口处，插上 Type-C 线，将 Type-C 线的另一端 USB 口连接电脑。并在图形化配置界面，将 USB 设备选择为 CDC 串口设备。

```
static int __dynamic_stdin_fd = -1;
static aw_err_t __dynamic_stdout_ret = -AW_EBADF;
static aw_err_t __dynamic_stderr_ret = -AW_EBADF;

aw_err_t aw_printf_redirect_dynamic_dev(void)
{
    int find = -AW_ENODEV;

    // 检测动态设备
```

```
find = aw_access(AW_DYNAMIC_DEV_PATH, AW_F_OK);

if(find == AW_OK) {
    // 关联标准文件流
    if(__dynamic_stdin_fd < 0)
    {
        __dynamic_stdin_fd = \
            aw_open_at(AW_DYNAMIC_DEV_PATH, AW_O_
RDWR, 0, 0);
        __dynamic_stdout_ret = aw_dup2(0, 1);
        __dynamic_stderr_ret = aw_dup2(0, 2);
        return AW_OK;
    }
}
else {
    // 清理文件描述符
    if(__dynamic_stdin_fd >= 0) {
        aw_close(0);
        __dynamic_stdin_fd = -1;
    }
    if (__dynamic_stdout_ret == AW_OK) {
        aw_close(1);
        __dynamic_stdout_ret = -AW_EBADF;
    }
    if (__dynamic_stderr_ret == AW_OK) {
        aw_close(2);
        __dynamic_stderr_ret = -AW_EBADF;
    }
}

return -AW_ENODEV;
}
```

```
int aw_main(void)
{
    int ret;

    aw_kprintf("hello world\n");
    printf("hello world\n");

    while(1) {
        ret = aw_printf_redirect_dynamic_dev();
        if (AW_OK == ret)
            break;

        // 设置检测周期
        AW_TASK_DELAY(100);
    }

    aw_kprintf("hello world, ZLG\n");
    printf("hello world, ZLG\n");

    return 0;
}
```

启动串口调试助手，搜索并打开 DEBUG UART 设备与 CDC 串口设备的端口号后，运行上文示例程序。根据下图可知，USB 设备枚举后，printf 函数成功重定向到了 CDC 串口设备。

```
COM7 - Tera Term VT
File Edit Setup Control Window Help
hello world
HPM USBD: USBD suspend
hello world, ZLG
```

```
COM5 - Tera Term VT
File Edit Setup Control Window Help
hello world, ZLG
```

CDC串口设备只输出了“hello world, ZLG”。说明，在printf函数重定向之前，填入的内容不会输出到CDC串口设备。

总结

实现重定向 printf 函数时主要关注以下两个关键点：

1. 重写 NEWLIB 标准库中 printf 函数的底层实现；
2. 将指定设备以标准文件流约定的文件描述符打开。

扩展阅读

本文所演示平台使用的是 GCC 编译器，其对应 C 库为 NEWLIB 标准库。在 AWorksLP 中 printf 函数的底层输出接口在 AWorksLP 中实现为 _write_r 函数，其具体代码实现如下所示。

```
__attribute__((__used__)) _ssize_t
_write_r(struct _reent *ptr, int fd, const void *buf, size_t nbytes)
{
    return aw_write(fd, buf, nbytes);
}
```

需要值得注意的是，上述适配方式仅兼容 NEWLIB，若是其他编译器，其实现以及接口不尽相同，下表仅给出部分以供参考，在使用时需根据实际情况进行调整。

工具链	标准库	底层接口
GCC	NEWLIB	_write_r
ARMCC	ARMCLIB	_sys_write
ARMCLANG	ARMCLIB	_sys_write



EPC6450-AWI开发板

[点击购买](#)

【EsDA 应用】 快速实现串口转HTTP请求

ZLG 致远电子 2023-11-08 11:42:02

本文将基于 EsDA 开发套件快速实现将串口数据转换为 HTTP 请求的应用开发，实现串口设备与网络服务的互联互通，提供便捷和高效的串口数据转换功能。

项目简介

HTTP 协议是互联网应用最为广泛的一种网络协议，由客户端发送请求消息，服务端针对客户端的请求进行响应回复。本文将基于 EsDA 开发平台，使用 EPC6450-AWI 开发板，以及图形化设计工具 AWFlow Designer 实现将串口数据转换为 HTTP 请求发送给云端，随后云端针对 HTTP 请求进行处理响应的功能。该项目主要用到了 serial_in_ex, fscript, http_request, fileout, timer, filein 和 serial_out_ex 节点，具体实现请见下文。

项目概述

该项目是将串口数据通过作为 HTTP 客户端的 EPC6450-AWI 开发板去发送 HTTP 请求到云端服务器，在云端服务器上处理 HTTP 请求并下发响应数据。

该项目的主要步骤如下：

1. 配置 EPC6450-AWI 的网口设备并连接到互联网；
2. 配置串口通信参数，包括波特率，数据位，停止位和校验等；
3. 从串口读取 HTTP 请求参数，在本项目中串口的输入数据如下：

POST 方法主要是输入 body 消息正文，GET 方法主要是输入 url 统一资源定位符；

4.http_request 节点向云端服务器发起 HTTP 请求，云端服务器处理数据后发送响应到客户端，客户端接收来自云端服务器的响应数据；

5.HTTP 客户端接收响应数据后，将响应数据中转到文本中保存并输出到串口进行显示。



项目准备

在本文将不再赘述串口设备在 EsDA 的基础通信，读者可以阅读以下文章对串口节点和 EsDA 的一些基础项目进行熟练：

- [【EsDA 应用】5 分钟实现一个串口通信业务](#)
- [【EsDA 应用】常用 IO 设备节点详解](#)
- [EsDA MPC-ZC1 应用——串口服务器\(一\)](#)

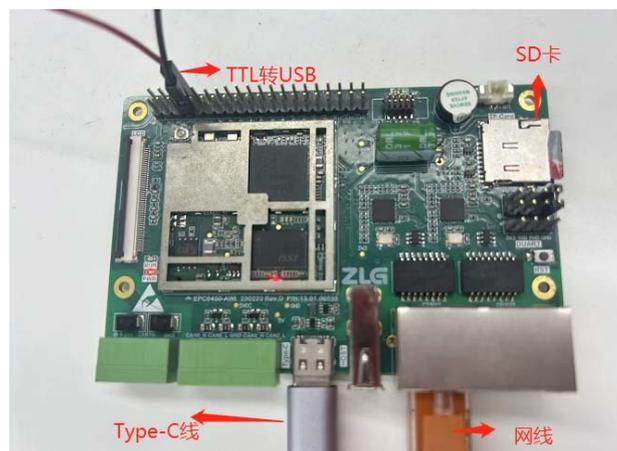
1. 硬件准备

在标有丝印为 TF Card 丝印的卡槽处，插入 SD 卡。

用户可随机选择 EPC6450-AWI 的可用串口设备，在本文将用 UART5 进行串口通信。在标有丝印为 UTX5,URX5 的串口模块上，将 TTL 转 USB 串口模块的 TX 与板子丝印为 URX5 连接，TTL 转 USB 串口模块的 RX 与板子丝印为 UTX5 相连；并将 TTL 转 USB 串口模块另一端的 USB 口接入电脑。

在标有丝印为 Type-C 的接口处，插上 Type-C 线，并将 Type-C 线的另一端 USB 口插入电脑。

在标有丝印为 NET0 或 NET1 的 RJ45 插座处接上水晶头，网线另一端水晶接头插在 PC 的网络插座上。



2. 网络搭建

本文的网络搭建是将电脑 wifi 通过以太网与开发板进行网络共享，以此达到开发板的以太网联网的目的。

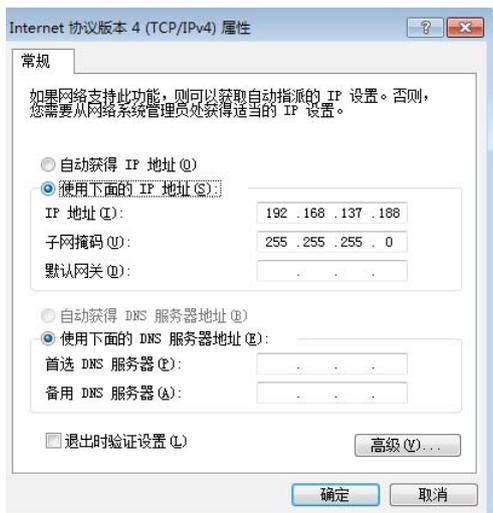
将 TTL 转 USB 串口模块接在丝印为 DUART 的调试串口上 (TX 接 RX, RX 接 TX)。

打开串口调试助手，检索并打开 TTL 转 USB 串口模块的设备端口号后，使用 shell 命令 ip addr，查看网口的 ip 地址，根据下图可知，本文使用的网口设备 ip 地址是 192.168.137.251。

```

[19:24:25.817]发->◇ip addr
□
[19:24:25.823]收<-◇ip addr
1 :eth0: <BROADCAST, MULTICAST, UP> mtu 1500 state UP
  link/ether 00:14:97:0f:01:22 brd ff:ff:ff:ff:ff:ff
  inet 192.168.137.251/24 brd 192.168.137.255
  inet gateway: 192.168.137.1
  inet dhcp: Off
2 :eth1: <NO-CARRIER, BROADCAST, MULTICAST, UP> mtu 1500 state UP
  link/ether 00:14:97:0f:01:23 brd ff:ff:ff:ff:ff:ff
  inet 192.168.137.252/24 brd 192.168.137.255
  inet gateway: 192.168.137.1
  inet dhcp: Off
  
```

配置 PC 上的以太网的 IP 与开发板的 IP 地址在同一局域网下。



将 PC 上的 WLAN 配置共享给与开发板连接的以太网。



在串口调试助手输入 shell 指令 ping www.baidu.com, ping 成功即开发板联网成功。

```
[19:36:25.800]发->ping www.baidu.com
[19:36:25.810]收->ping www.baidu.com
ping www.baidu.com(14.119.104.189) with 56 bytes
56 bytes from www.baidu.com:icmp_seq=5 ttl=53 time=14ms
[19:36:26.708]收->tk_led_toggle:72 condition(sw_led_toggle(TK_LED_PRIV(p_this)->Fd) == AW_OK) failed!
[19:36:26.685]收->56 bytes from www.baidu.com:icmp_seq=6 ttl=53 time=13ms
[19:36:27.676]收->56 bytes from www.baidu.com:icmp_seq=7 ttl=53 time=13ms
[19:36:28.311]收->tk_led_toggle:72 condition(sw_led_toggle(TK_LED_PRIV(p_this)->Fd) == AW_OK) failed!
[19:36:28.901]收->56 bytes from www.baidu.com:icmp_seq=8 ttl=53 time=14ms
```

项目实施

本项目业务主要分为两个部分：

POST 方法请求项目：模拟传感器数据通过串口转 HTTP 客户端传送到 HTTP 服务器，服务器进行数据分析和处理。将串口数据作为 HTTP 请求的 body 参数，http_request 节点在整理属性和输入参数后对自建的 HTTP 服务器发起 POST 方法的 HTTP 请求；服务器收到请求后，将串口数据保存到本地的文本文件以便后续查看，下发响应数据给客户端；客户端收到服务器的响应数据经过数据处理后打印到串口助手的界面进行查看。

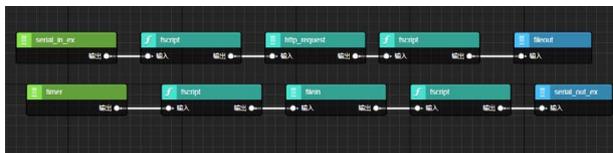
GET 方法请求项目：请求访问百度服务器。串口提供 HTTP 请求参数，http_request 节点整理属性和输入参数对百度服务器发起 GET 方法的 HTTP 请求；http_request 节点接收来自百度服务器的响应数据，将数据保存到指定的文本文件中存储再从文本中将数据输出到串口助手的界面上。

1. POST方法请求

本项目由串口助手模拟将采集的传感器数据通过开发板的串口传输到开发板创建的 HTTP 客户端，HTTP 客户端再通过 POST 方法将传感器数据作为 body 参数去请求本地搭建的 HTTP 服务器，HTTP 服务器接收到请求后，将传感器数据存储在本地的 index.html 文件中，并将数据处理后作为响应体回发给客户端，HTTP 客户端接收到响应消息后打印到串口助手上进行显示。

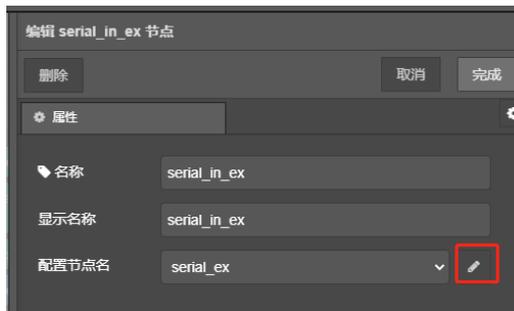
1.1 流程图绘制

添加 serial_in_ex, fscript, http_request, fileout, timer, filein 和 serial_out_ex 节点到画布中并连线如下图。

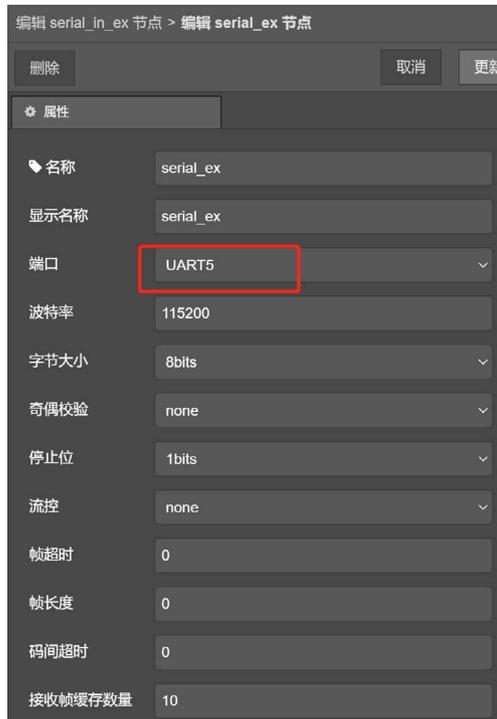


1.2 节点配置

在本文的 serial_in_ex 和 serial_out_ex 节点配置参数和操作一致，后面不再赘述 serial_out_ex 节点的配置操作。双击 serial_in_ex 节点，点击配置节点名旁边的铅笔图标。



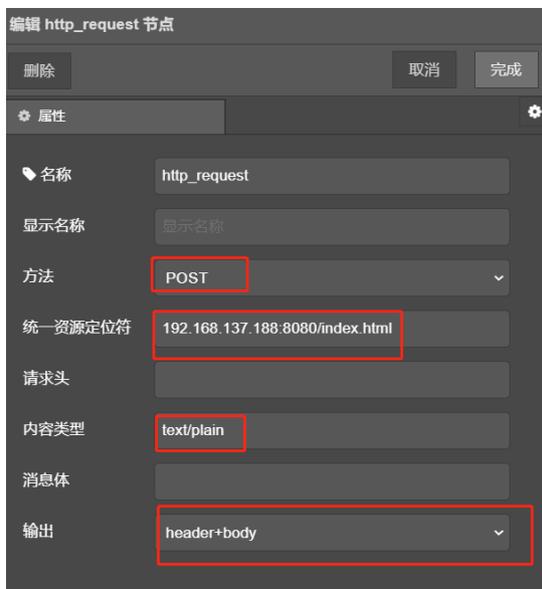
选择用户使用的串口设备，根据实际需求配置波特率，奇偶校验等串口配置参数，本项目中的串口配置参数如下图所示。



双击 serial_in_ex 的消费者节点 fscript，因为本项目主要是将采集到的传感器数据作为消息体参数发送到 HTTP 服务器进行处理，所以该 fscript 主要是将读取到的 serial_in_ex 串口数据赋值给 http_request 节点的 body 参数如下：

```
msg.body = istream_read_string(msg.istream, 100)
```

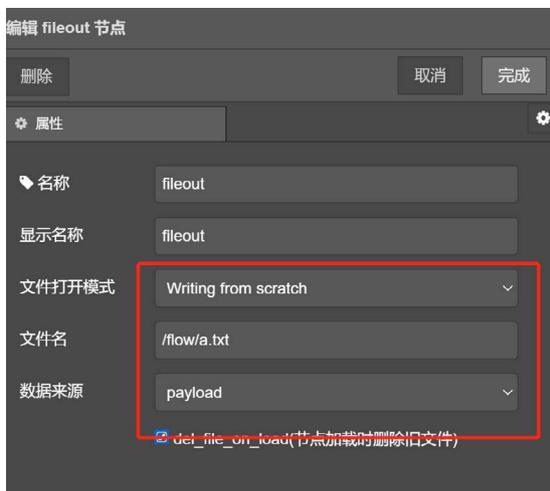
双击 http_request 节点，配置方法为 POST，并设置用户想要访问的 HTTP 服务器 URL(这里的 HTTP 服务器是笔者本地用 python 搭建的一个简易 HTTP 服务器)，根据需要选择输出的内容类型，其他参数按需配置即可。



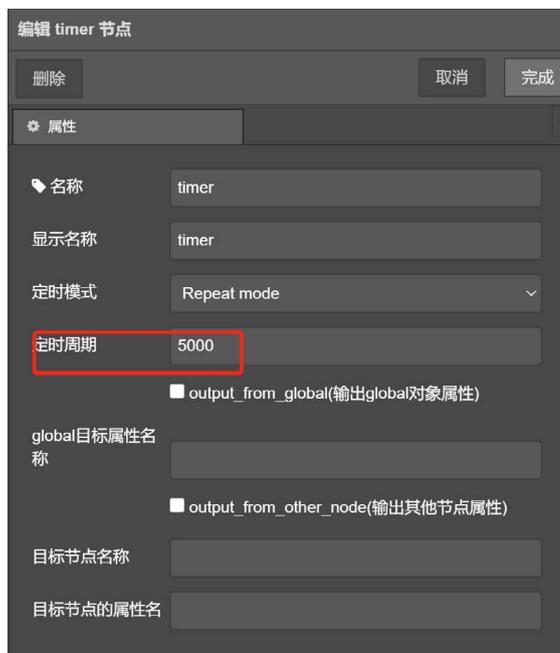
双击 http_request 节点的消费者节点 fscript，该节点主要是存储 http_request 节点的输出参数 msg.payloadLength，用于后续赋值给 filein 节点的输入参数读取的数据长度。

```
set(global.length, msg.payloadLength)
```

双击 fileout 节点，配置属性参数如下，在本项目中配置文件打开模式为从头写入且丢弃源文件内容，文件名选择开发板上自动挂载的 /flow 目录下的文件，数据来源选择 payload 形式。



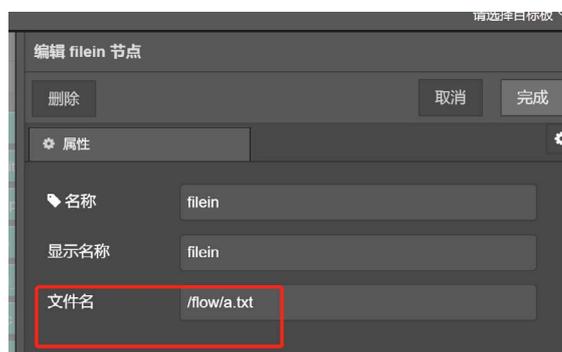
双击 timer 节点，配置定时周期时长，定时用 filein 节点去读取存放 HTTP 响应的数据。



双击 timer 的消费者节点 fscript，配置 filein 节点的输入参数如下：

```
set(msg.topic, "exec:read_data");
var length = global.length
set(msg.payload, length);
```

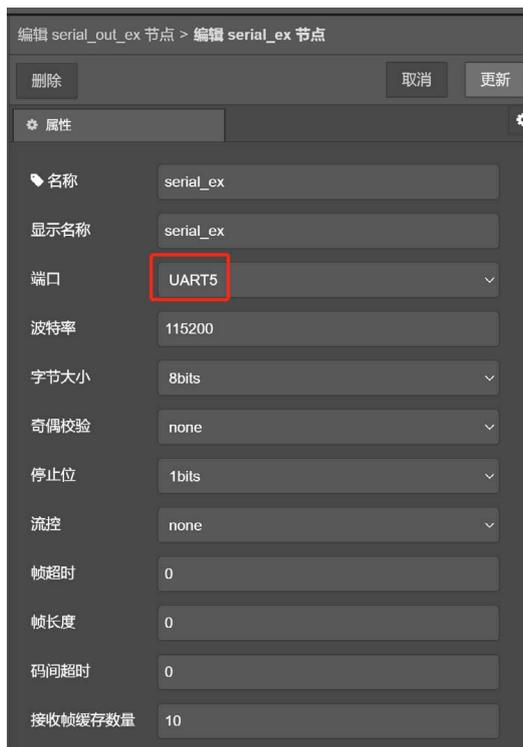
双击 filein 节点，配置需要读取的文件名。



双击 filein 的消费者节点 fscript，该节点主要将从 filein 节点读取到的数据转换给 serial_out_ex 节点。

```
set(output.payload, str(msg.payload, true));
```

双击 serial_out_ex 节点，该节点配置参数与 serial_in_ex 节点一样。



1.3 本地HTTP服务器搭建

笔者用 python 脚本搭建了一个简易的 HTTP 服务器，主要功能是收到的 POST 请求中的消息体数据存储到本地的 index.html 文本中，并将收到的消息体数据处理后作为响应体回发给客户端。在 PC 端执行以下 http_server.py 的脚本即开启了本地的 HTTP 服务端。

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import logging

class S(BaseHTTPRequestHandler):
    def do_HEAD(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()

    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        post_data = self.rfile.read(content_length)

        logging.info("POST request,\nPath: %s\nHeaders:\n%s\n\n\nBody:\n%s\n",
                    str(self.path), str(self.headers), post_data.decode('utf-8'))

        res = "You Input: " + post_data.decode('utf-8')
```

```
with open("index.html", "a+") as f:
    f.write(post_data.decode('utf-8'))

self.do_HEAD()
self.wfile.write("{}".format(res).encode('utf-8'))

def respond(self, opts):
    response = self.handle_http(opts['status'], self.path)
    self.wfile.write(response)

def handle_http(self, status_code, path):
    self.send_response(status_code)
    self.send_header('Content-type', 'text/html')
    self.end_headers()
    content = ""
    <html><head><title>Title goes here.</title></head>
    <body><p>This is a test.</p>
    <p>You accessed path: {}</p>
    </body></html>
    "".format(path)
    return bytes(content, 'UTF-8')
```

```
def run(server_class=HTTPServer, handler_class=S, port=8080):
    print("run()")
    logging.basicConfig(level=logging.INFO)
    server_address = ('', port)
    httpd = server_class(server_address, handler_class)
    logging.info('Starting http server...\n')
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        pass
    httpd.server_close()
    print("httpd.server_close()")
    logging.info('Stopping http server...\n')
```

```
if __name__ == '__main__':
    from sys import argv

    if len(argv) == 2:
        run(port=int(argv[1]))
    else:
        run()
```

1.4 下载运行

将流程图下载到目标开发板后，在串口调试助手输入 HTTP 请求的消息体（模拟的传感器数据）后，就可以在串口助手输出界面看到返回的响应内容如下：

```
[17:11:30.013]发-> temperature:55 humidity:32
[17:11:32.582]收<- HTTP/1.0 200 OK
Server: BaseHTTP/0.6 Python/3.7.9
Date: Wed, 01 Nov 2023 09:10:34 GMT
Content-type: text/html

You Input:
[17:11:37.553]收<- temperature:55 humidity:32
```

本地的 HTTP 服务器端收到的来自客户端的请求信息如下：

```

D:\>python http-server1.py
run()
INFO:root:Starting httpd...

INFO:root:POST request,
Path: /index.html
Headers:
Host: 192.168.137.188:8080
Content-Length: 28
Content-Type: 1875444760

Body:
temperature:55 humidity:32

192.168.137.251 -- [01/Nov/2023 17:23:35] "POST /index.html HTTP/1.1" 200 -
    
```

打开 PC 本地的 index.html 文本，可以看到保存的传感器数据如下：

```

temperature:55 humidity:32 temperature:56 humidity:74
temperature:21 humidity:63 temperature:27 humidity:89
temperature:24 humidity:12 temperature:18 humidity:23
temperature:17 humidity:52 temperature:17 humidity:24
temperature:54 humidity:45 temperature:24 humidity:49
temperature:24 humidity:51 temperature:24 humidity:58
temperature:25 humidity:63 temperature:25 humidity:27
temperature:35 humidity:56 temperature:54 humidity:72
temperature:54 humidity:54
    
```

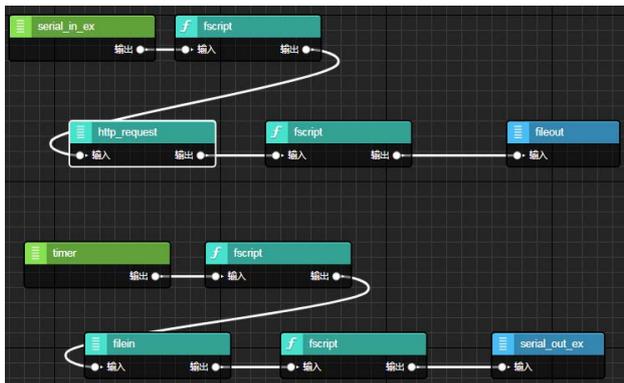
至此，串口转 HTTP 客户端的传感器数据上云的实验就已完全结束，接下来开启访问百度浏览器的 HTTP 请求实验。

2. GET方法请求

该项目主要是通过 GET 方法请求百度服务器，最后将百度服务器响应的消息进行打印显示，主要是将串口数据转换为 HTTP 请求，发送到服务器端，以实现与远程服务器的通信。

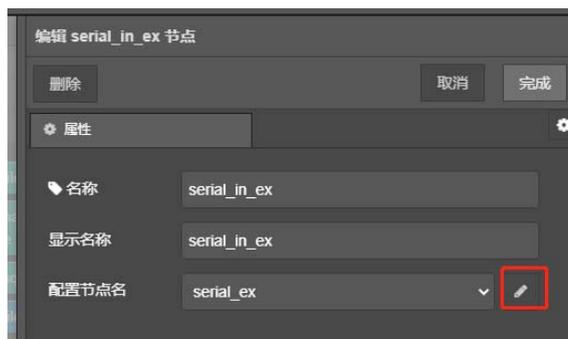
2.1 流图绘制

添加 serial_in_ex, fscript, http_request, fileout, timer, filein 和 serial_out_ex 节点到画布中并连线如下图。

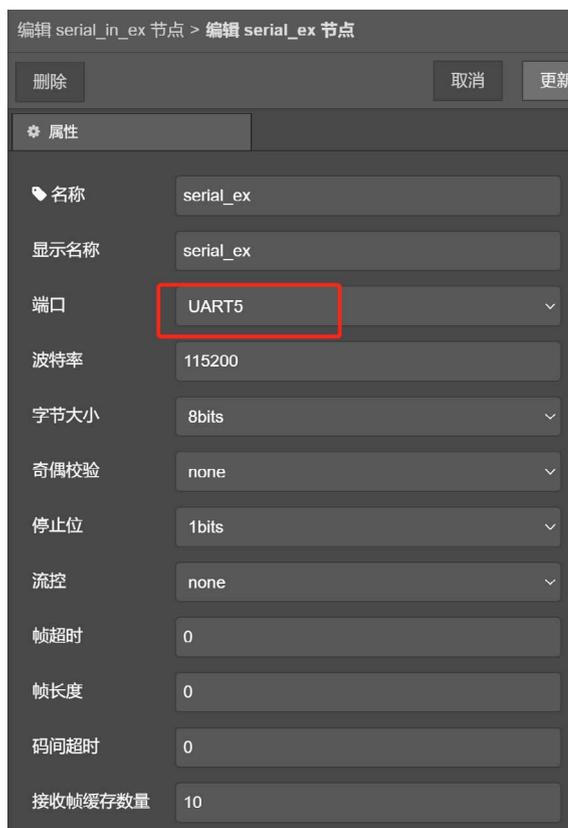


2.2 节点配置

在本文的 serial_in_ex 和 serial_out_ex 节点配置参数和操作一致，后面不再赘述 serial_out_ex 节点的配置操作。双击 serial_in_ex 节点，点击配置节点名旁边的铅笔图标。



选择用户使用的串口设备，根据实际需求配置波特率，奇偶校验等串口配置参数，本项目中的串口配置参数如下图所示。

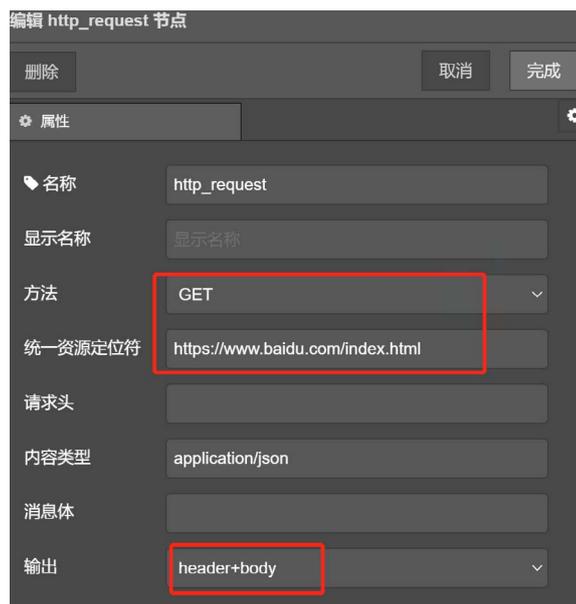


双击 serial_in_ex 的消费者节点 fscript，因为本项目主要是对 HTTP 协议的 GET 方法进行数据请求，而 GET 的请求参数是拼接在 URL 的后面，所以串口输入的主要是 http_request 节点的 URL 配置项，该 fscript 主要是读取 serial_in_ex 的串口数据，如下：

```

var str = istream_read_string(msg.istream, 100)
msg.url = str
    
```

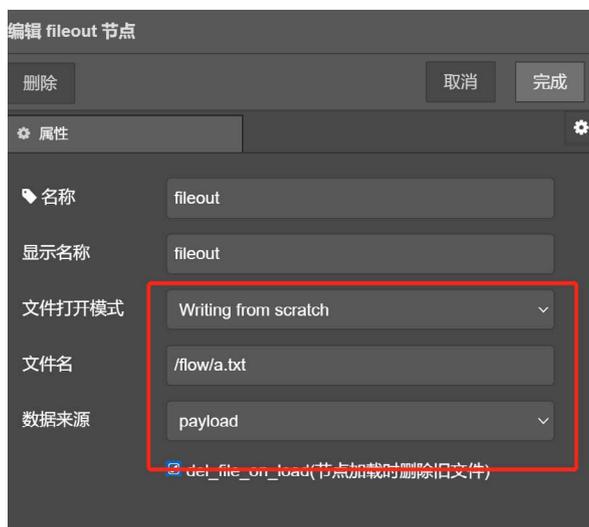
双击 http_request 节点，配置方法为 GET，并设置用户想要访问的云端服务器 URL，根据需要选择输出内容类型，其他参数按需配置即可。



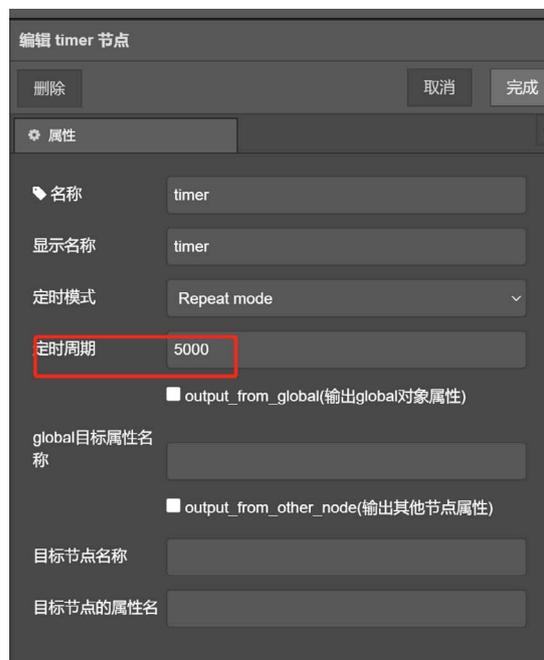
双击 http_request 节点的消费者节点 fscript，该节点主要是存储 http_request 节点的输出参数 msg.payloadLength，用于后续赋值给 filein 节点的输入参数读取的数据长度。

```
set(global.length, msg.payloadLength)
```

双击 fileout 节点，配置属性参数如下，在本项目中配置文件打开模式为从头写入且丢弃源文件内容，文件名选择开发板上自动挂载的 /flow 目录下的文件，数据来源选择 payload 形式。



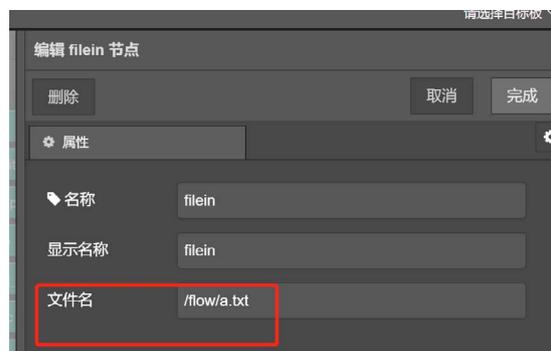
双击 timer 节点，配置定时周期时长，定时用 filein 节点去读取存放 HTTP 响应的数据。



双击 timer 的消费者节点 fscript，配置 filein 节点的输入参数如下：

```
set(msg.topic, "exec:read_data");
var length = global.length
set(msg.payload, length);
```

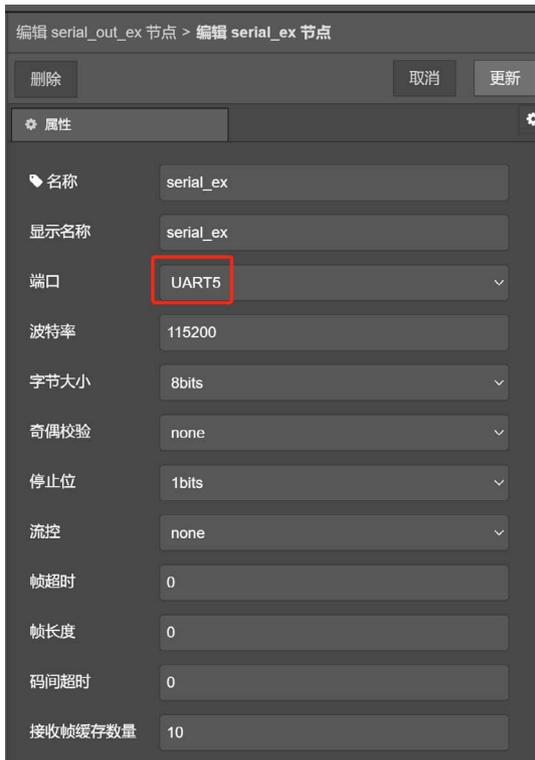
双击 filein 节点，配置需要读取的文件名。



双击 filein 的消费者节点 fscript，该节点主要将从 filein 节点读取到的数据转换给 serial_out_ex 节点。

```
set(output.payload, str(msg.payload, true));
```

双击 serial_out_ex 节点，该节点配置参数与 serial_in_ex 节点一样。



2.3 下载验证

将该流图下载到目标开发板上，在串口调试助手输入 HTTP 请求的云端 URL 后，可以看到返回的响应消息，具体如下图。

```

[14:14:53.795] [E] -> [O] https://www.baidu.com/index.html
[14:15:18.462] [E] -> [O] https://www.baidu.com/index.html 通过串口发送的get方法请求的url网址
[14:15:28.793] [E] -> [O] HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 9908
Content-Type: text/html
Date: Fri, 12 Oct 2023 06:13:53 GMT
Psp: CF= OTI DSP CDR IVA 008.130 COM * 接收到的http响应的header+body
Psp: CF= OTI DSP CDR IVA 008.130 COM *
Pragma: no-cache
Server: BE/1.1
Set-Cookie: BAIDUID=A4E44F7E544B342E27E217418B4F4D:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: BIDISID=A4E44F7E544B342E27E217418B4F4D; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: E2781697177632; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: BAIDUID=A4E44F7E544B342E27E217418B4F4D:FG=1; max-age=31536000; expires=Sat, 12-Oct-24 06:13:53 GMT; path=/; domain=baidu.com; version=1; comment=CN
TraceId: 1697177632318507941216499421633712893
Vary: Accept-Encoding
X-UA-Compatible: IE=Edge, chrome=1

<!DOCTYPE html><html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"><meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><meta content="always" name="referrer"><meta name="description"
[14:15:31.842] [E] -> [O] HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 9908
Content-Type: text/html
Date: Fri, 12 Oct 2023 06:14:22 GMT
Psp: CF= OTI DSP CDR IVA 008.130 COM *
Psp: CF= OTI DSP CDR IVA 008.130 COM *
Pragma: no-cache
Server: BE/1.1
Set-Cookie: BAIDUID=A8F94F764B42A76C38F4815017A714:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: BIDISID=A8F94F764B42A76C38F4815017A714; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: E2781697177632; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=baidu.com
Set-Cookie: BAIDUID=A8F94F764B42A76C38F4815017A714:FG=1; max-age=31536000; expires=Sat, 12-Oct-24 06:14:22 GMT; path=/; domain=baidu.com; version=1; comment=CN
TraceId: 169717763231850794144600666010294570
Vary: Accept-Encoding
X-UA-Compatible: IE=Edge, chrome=1

<!DOCTYPE html><html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"><meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><meta content="always" name="referrer"><meta name="description"

```

至此该实验到此结束。

【技术分享】 Modbus RTU Master 转 MQTT

ZLG 致远电子 2023-11-22 11:34:38

随着物联网的迅速发展，越来越多的工业自动化设备需要上报数据到云。本文将基于 EsDA 开发套件快速实现将 Modbus RTU 协议的数据转换为 MQTT 格式，实现更灵活的数据传输和远程控制。

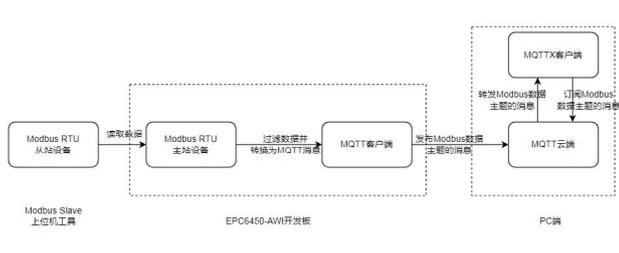
项目简介

在物联网领域中，Modbus RTU 是一种串行通信协议通常用于工业自动化领域，MQTT 则是基于发布 / 订阅模式的消息传递的协议，并广泛应用于物联网设备间的通信。本文将介绍如何使用 EPC6450-AWI 开发板，以及图形化设计工具 AWFlow Designer 将 Modbus RTU 协议转换为 MQTT 协议，实现物联网设备的数据传输与远程控制。

项目概述

本项目主要将 Modbus RTU 数据转换为 MQTT 消息，然后通过 MQTT 协议将 Modbus 采集到的数据以主题消息的形式发送到 MQTT 服务器 (在 PC 本地自搭建的 emqx 代理服务器)，最后在 PC 端的 MQTTX 客户端上显示 (MQTTX 订阅相关主题的消息后，即可收到由 MQTT 服务器转发的 Modbus 数据主题的消息)。具体可以分为以下几个步骤：

- Modbus RTU 数据采集：通过 modbus_master_in 节点去采集 Modbus RTU 从站设备的数据；
- 数据转换：将采集到的 Modbus RTU 从站设备的数据，通过 modbus_parse_in 和 fscript 节点转换为有效且能被 mqtt_out 节点接收的数据；
- MQTT 发布：mqtt_out 节点可发布 Modbus 主题消息到 MQTT 云端服务器；
- MQTT 订阅：由 MQTTX 客户端工具订阅 Modbus 相关主题，接收并显示发布的 Modbus 主题消息。



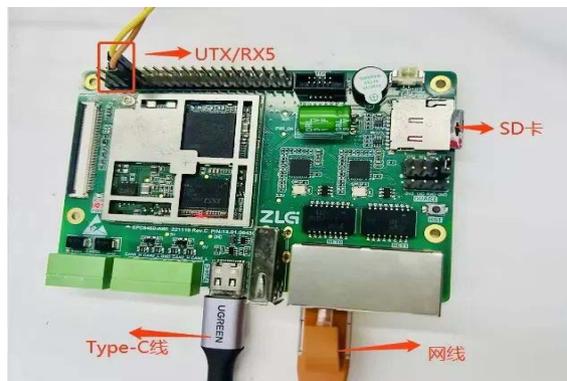
项目准备

客户可阅读以下相关文章，对 Modbus 协议和 MQTT 协议在 EsDA 上的使用操作上可以有更多的了解：

- [Modbus RTU Master 转 TCP Server](#)
- [Modbus TCP Master 转 ZWS 云](#)
- [IO 设备接入 MQTT](#)
- [EsDA MPC-ZC1 应用——串口服务器 \(一\)](#)

1. 硬件准备

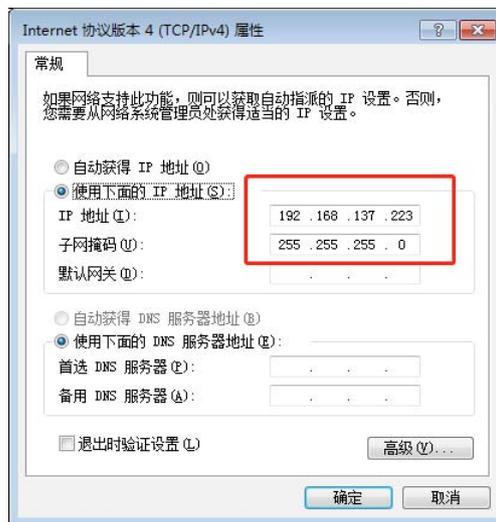
- 在标有丝印为 TF Card 丝印的卡槽处，插入 SD 卡。
- 在标有丝印为 UTX5 和 URX5 的模块上，将 TTL 转 USB 串口模块的 TX 与板子丝印为 RXD 连接，TTL 转 USB 串口模块的 RX 与板子丝印为 TXD 相连；并将串口模块的 USB 端口接入电脑。
- 在标有丝印为 Type-C 的接口处，插上 Type-C 线，并将 Type-C 的 USB 端口插入电脑。
- 在标有丝印为 NET0 或 NET1 的 RJ45 网络插座处，插上网线水晶头，另一端的网线水晶头插在 PC 的网络插座上。



2. 软件准备

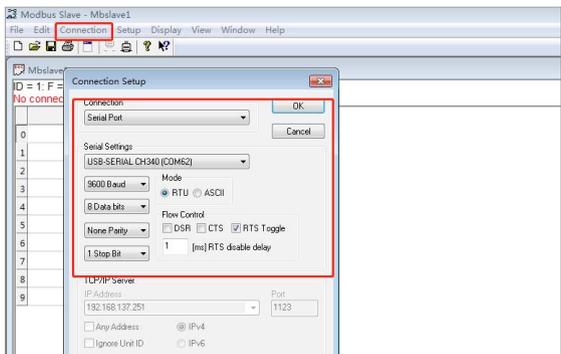
2.1 网络搭建

- 因为 MQTT 服务器是在 PC 端本地搭建的，所以与开发板通信的网口网络和 PC 的以太网网络在同一局域网下即可，不需要接入互联网。
- 出厂固件的 NET0 默认网口 IP 为：192.168.137.251，NET1 默认网口 IP 为：192.168.137.252。所以配置 PC 上的以太网的 IP 与开发板的 IP 地址在同一局域网下如图所示。

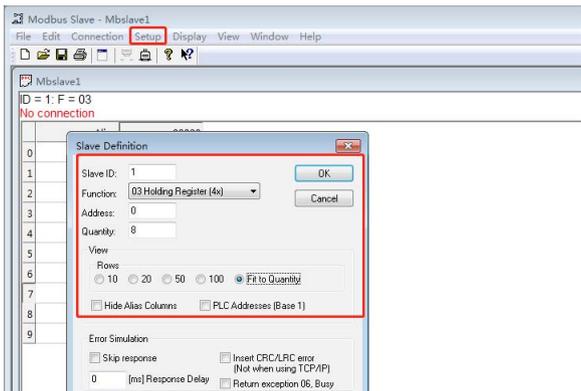


2.4.2 配置Modbus Slave工具

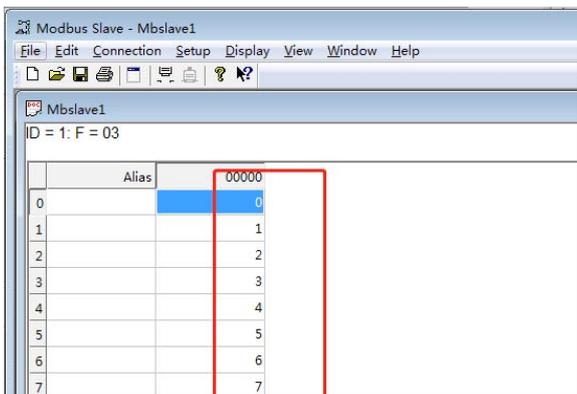
打开刚才安装的 Modbus Slave 软件。点击 Connection->Connection Setup, 根据用户实际使用情况配置 Modbus 串通信的设备参数, 随后点击 OK 即可保存设置。



点击 Setup->Slave Definition, 根据用户实际使用情况配置 Modbus 通信协议的从设备参数, 点击 OK 保存。



配置 Modbus 从设备对应寄存器的值如下图。



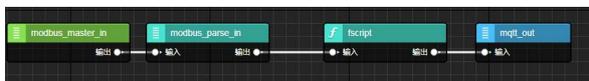
项目实施

本项目主要分为以下四个部分:

1. Modbus RTU Master 采集 Modbus 从站设备寄存器的值;
2. 将 Modbus 数据转换为 MQTT 协议消息内容的主体;
3. 开发板上搭建的 MQTT 客户端发送 Modbus 主题消息到 MQTT 服务器;
4. MQTT 服务器将消息转发给订阅 Modbus 主题的 MQTT 客户端。

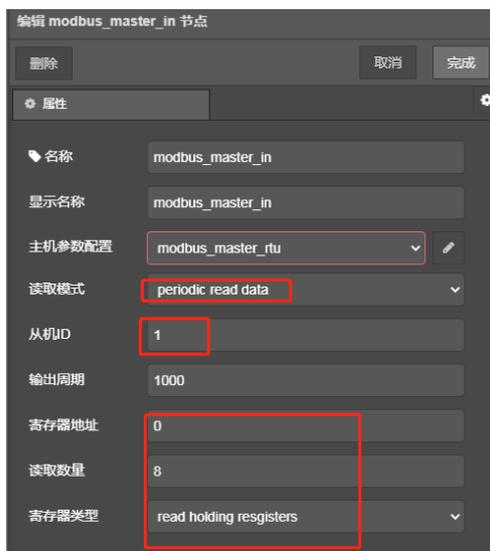
1. 流图绘制

添加 modbus_master_in, modbus_parse_in, fscript 和 mqtt_out 节点到画布中并连线如下图。

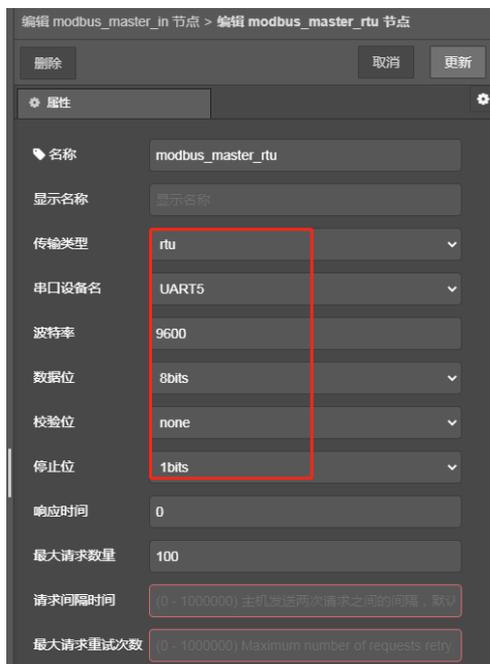


2. 节点配置

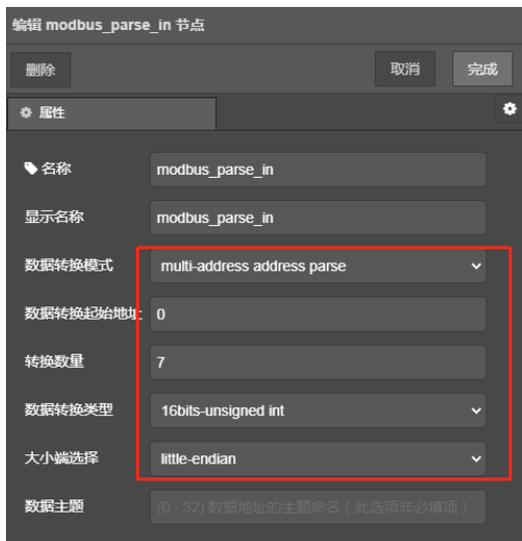
双击 modbus_master_in 节点, 配置需要读取的 Modbus 从机 ID 和寄存器的参数, 该节点的相关属性需要与前面配置的 Modbus Slave 模拟从机工具的参数一致。点击完成即可保存配置。



双击 modbus_master_in 节点的主机参数配置属性旁边的铅笔图标配置 Modbus 主机的参数配置, 根据用户实际需要选择相关参数。本项目主要是使用 rtu 传输数据, EPC6450-AWI 开发板用丝印为 UTX/RX5 的串口设备与从站进行 Modbus 通信。点击更新即可保存配置。



双击 modbus_parse_in 节点，根据用户实际需要选择进行数据转换的方式，该节点接收 modbus 输入形节点的原始数据，支持将不同数量的寄存器地址的值按序向其消费者节点进行输出。点击完成即可保存配置。



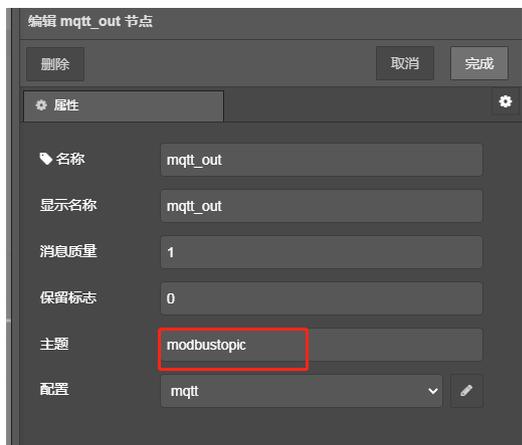
双击 fscript 节点，将读到的 Modbus 从站寄存器的值转换为 mqtt_out 节点可以接收的数据类型的值。点击完成即可保存配置。

```
var a = "modbus data is "

for(var i=0; i<array_size(msg.payload); i=i+1)
{
  a = join(", ", a, array_get(msg.payload, i))
}

wb = wbuffer_create()
wbuffer_write_string(wb, a)
output.payload = wbuffer_get_data(wb)
output.payloadLength = len(a)
```

双击 mqtt_out 节点，填写需要发布的主题名。点击完成即可保存配置。



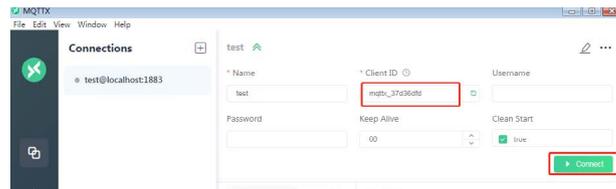
双击 mqtt_out 节点的配置属性旁边的铅笔图标，相关参数根据用户实际使用情况填写，需注意的是主机地址填写与开发板网口进行 MQTT 通信的以太网 IP 地址，若使用 emqx 服务器则端口号通用为 1883。点击更新按钮即可保存配置。



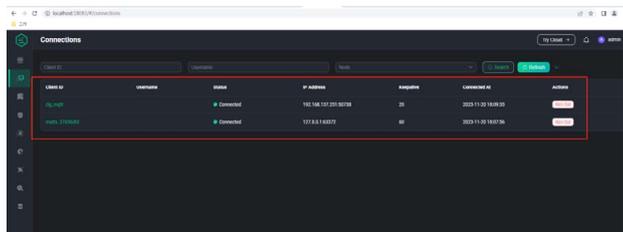
3. 下载验证

下载 AWFlow 流图到核心板中。

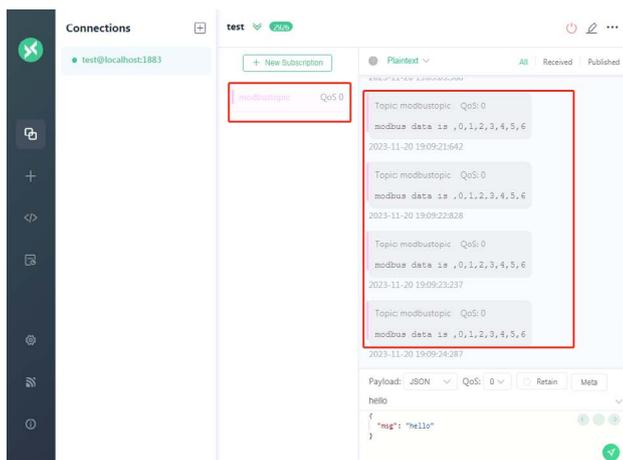
用户在 MQTTX 上位机上创建一个随机的 MQTT 客户端如下图所示，点击连接按钮。



打开 EMQX 服务器网址，可以看到所有客户端的连接情况。由下图可知 EPC6450-AWI 开发板上的 MQTT 客户端和 MQTTX 上位机创建的 MQTT 客户端都已连接到 EMQX 服务器上了。



在 MQTTX 上创建的客户端上订阅开发板上发布的 Modbus 主题名，就可以收到来自 MQTT 服务器转发的 Modbus 主题消息如下图所示。至此，Modbus Master RTU 转 MQTT 协议的项目就已实现。



MR6450/6750系列核心板

[点击图片](#)

【从0开始创建AWTK应用程序】 开发及调试环境搭建

ZLG 致远电子 2023-11-10 11:46:25

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

安装AWStudio

AWStudio 是用来整合 AWTK Designer 的工具，可以管理 AWTK Designer 创建的项目，我们可以通过 AWStudio 来安装 AWTK 的开发环境。

1. 下载与安装AWStudio

首先，我们使用浏览器打开 AWStudio 云平台，点击下面的“注册”按钮进行账号的注册，如下图所示：



图1 AWStudio云平台注册账号

需要注意的是，在注册过程中填写完整的公司名称，并需要绑定好邮箱，这样后台才可以审核通过并正确发放续期卡券到账号上，如果之前注册的账号没有绑定到邮箱，可以通过 AWStudio 云平台中的“个人信息”——“账号安全”选项卡中进行编辑，如下图所示：



图2 绑定邮箱与填写正确的公司名称



图3 编辑AWStudio云平台账号邮箱信息

注册完成在 AWStudio 云平台界面点击右上角的“客户端”跳转至 AWStudio 下载界面，并根据自己系统选择合适的版本进行下载。下载完成后，双击“AWStudio-Community-x64-xxx-Setup.exe”打开安装包，并根据提示完成安装，由于安装过程比较简单，这里不过多赘述。

2. 激活与续期AWStudio

在首次登录进 AWStudio 时，系统会提示需要选择一个许可证进行激活，这里我们选择其中一个许可证点击“激活”按钮即可。如果许可证到期可以在 AWStudio 云平台的许可管理页面点击“续期”按钮选择续期卡券来续期许可证，如下图所示：



图4 许可证激活



图5 许可证续期

使用AWStudio环境检测工具安装AWTK开发环境

在 AWStudio 主界面点击上方“帮助”——“检测 AWTK 开发环境”按钮可以打开 AWStudio 自带的环境检测工具，如下图：



图6 环境检测工具

这个列表中所列出的环境都推荐安装，其中 Python 与 Node.js 是负责打包 AWTK 程序所用到的资源文件，而 SCons 与 编译器是负责将 AWTK 应用程序编译成可以在桌面系统（如：Windows 和 Ubuntu）模拟器上执行的应用程序，如果用户不需要在模拟器运行可以选择只安装 Python 和 Node.js。点击环境检测工具下方的“检测”按钮可以快速检测当前环境是否完整，如果检测到不完整的话点击下方的“一键安装”按钮可以一键安装 Python、SCons、Node.js 环境。

1. 安装编译器

AWTK 是跨平台的 GUI，可以在 Windows 等桌面系统运行，也可以在嵌入式系统运行，运行前均需要使用对应平台的编译器进行编译。安装 Windows 版的 AWStudio 后，如需要在 Windows 模拟运行 AWTK 程序，需要用户自行安装编译器，AWTK 支持 Visual Studio C++ 编译器（推荐使用）与 MinGW 编译器，我们只需要二选一选择其中一个安装即可，并不需要两个编译器都进行安装，安装教程可以点击对应编译器的“安装指南”按钮进行查看。

需要注意的是如果选择安装 Visual Studio C++ 编译器的话，需要安装 Visual Studio 2015 或以上的版本。如果安装 Visual Studio 2015 需要勾选“Visual C++ 编程语言”，如果安装 Visual Studio 2019 则需要选择“使用 C++ 的桌面开发”，如下图所示：

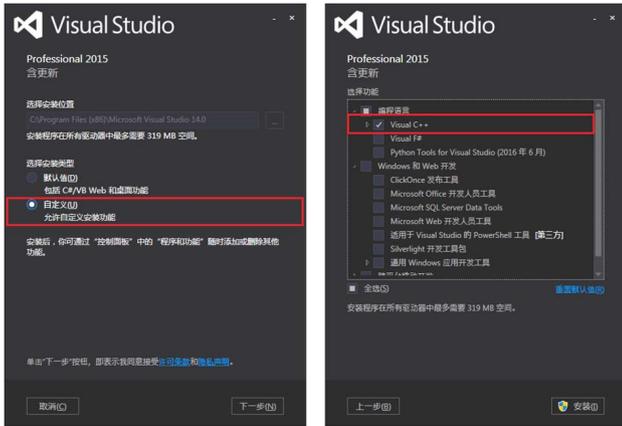


图7 安装Visual Studio 2015



图8 安装Visual Studio 2019

编译AWTK

安装好所需编译环境之后，我们在 AWStudio 新建一个工程并打开，此时点击上方工具栏的“项目”-“编译”按钮时会提示我们 AWTK 未进行编译，

这时候根据提示点击“是”即可进行编译。除了上面方式，还可以点击左侧导航栏的“插件管理”按钮进入插件页面也可以进行 AWTK 编译，如下图所示：



图9 编译AWTK的方式



图10 编译AWTK的方式

等待一段时间后可以看到如下图所示的 AWTK 编译完成提示。



图11 AWTK编译完成

到这里，我们的 AWTK 开发环境便搭建好了，如果你在安装过程中遇到了问题，欢迎加入交流群讨论。

【从0开始创建AWTK应用程序】 创建应用程序并在模拟器运行

ZLG 致远电子 2023-11-30 11:35:58

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

创建工程

本篇文章我们来创建第一个 AWTK 程序，也就是 HelloWorld 程序，它总共包含两个页面，先来看下最终效果图：

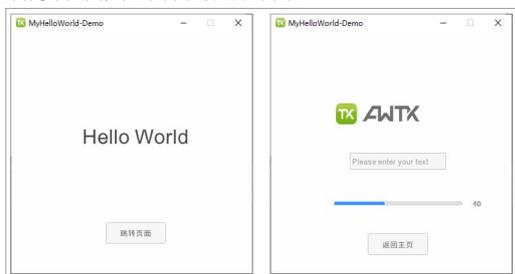


图1 HelloWorld工程两个页面的效果预览

首页包含两个控件，一个是显示“Hello World”的 label 控件，另一个是显示“跳转页面”文本的 button 控件，点击按钮时会跳转到第二个页面。

第二个页面包含显示图片的 image 控件、edit 控件、进度条 progress_bar 控件、label 控件以及按钮 button 控件。

其中 image 控件具有位移动画，会循环地左右移动。edit 控件在修改值时会同步地改变进度条的值和 label 控件的值。

1. 使用AWStudio创建项目

在开始设计 UI 前，我们在之前安装好的 AWStudio 中新建一个项目，注意新建项目时类型要选择“AWTK Application”。

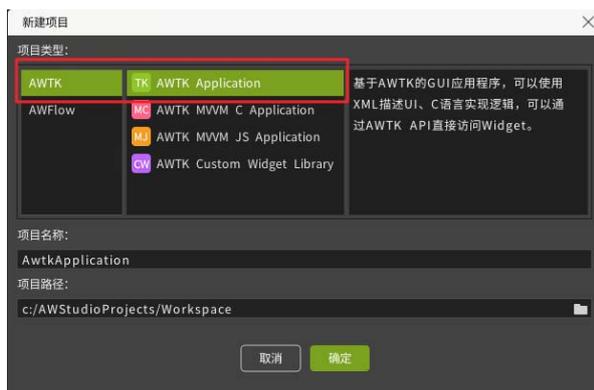


图2 使用AWStudio创建项目

AWTK Designer 是 AWStudio 中内置的工具，专门用来制作 AWTK 应用程序 UI 界面的实用工具，下图是创建好项目后的界面：

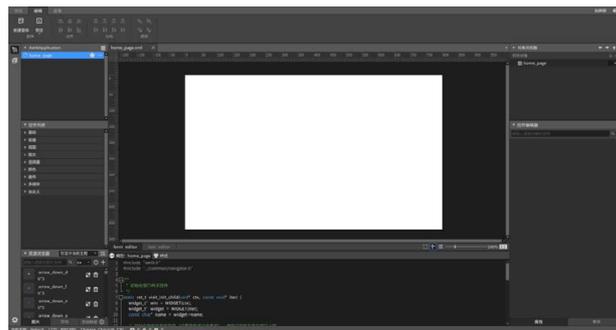


图3 创建项目后进入AWTK Designer的UI设计界面

使用AWTK Designer设计UI界面

首先参照预览图，我们需要一个 label（静态文本）控件和 button（按钮）控件，我们在左侧的控件列表中找到这两个控件并拖到中间的控件编辑栏中：

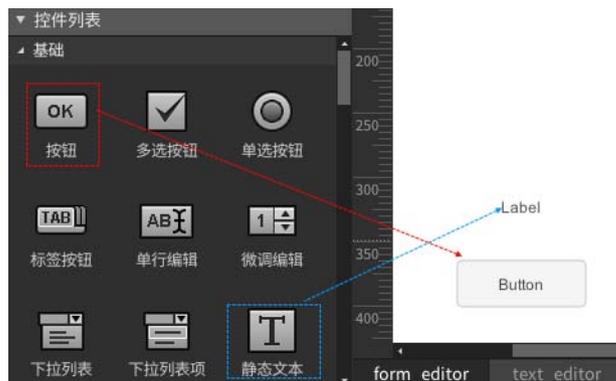


图4 放置首页的label控件与button控件

1. 设置首页label控件与button控件文本及样式

首先我们修改控件的文本以及字体样式，修改控件文本有两种方式，一种是双击控件直接修改，另一种是修改控件的 text 属性。

除了 text 属性之外，还可以通过设置 font_size 属性修改字体的字号，修改完毕后效果图如下：

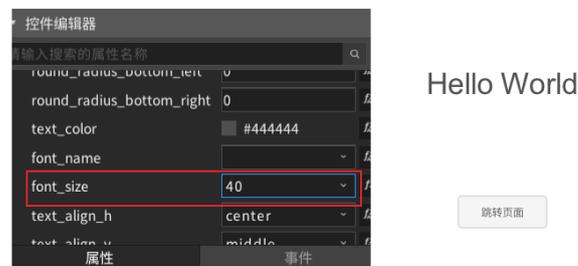


图5 首页label控件和button控件的字号设计与显示效果

2. 新建第二个页面并进行布局

接下来我们设计第二个页面，我们通过 AWTK Designer 上方的编辑 - 新建窗体选项新建一个窗口 (Window)，设置好窗体设置栏中的名称，操作如下图：

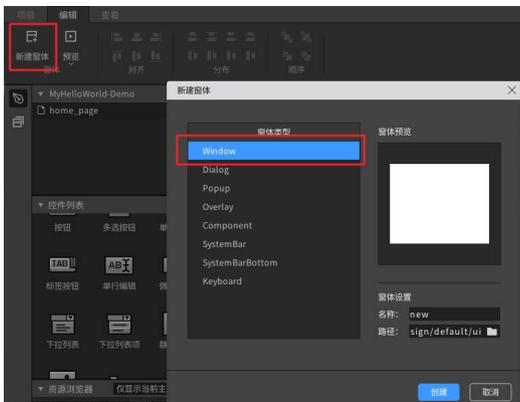


图6 使用AWTK Designer新建窗口

按照图 1 的效果图将第二个页面所需控件拖拽出来，并进行属性调整。

另外，由于我们的 edit 只用来改变进度条的值，这个值是非负整数，所以将 edit 的 input_type 属性修改成 "uint"。

image 控件显示图片前需要将图片添加到资源浏览器，再到控件中设置 image 属性为该图片，如下图：



图7 添加图片资源并应用到image控件中

3. 设置控件的位移动画

接下来给图片添加位移动画，我们选中 image 控件，在右侧“控件编辑器”-“动画”选项中添加一个动画，动画类型选择“位移”，然后对下图设置动画参数即可。

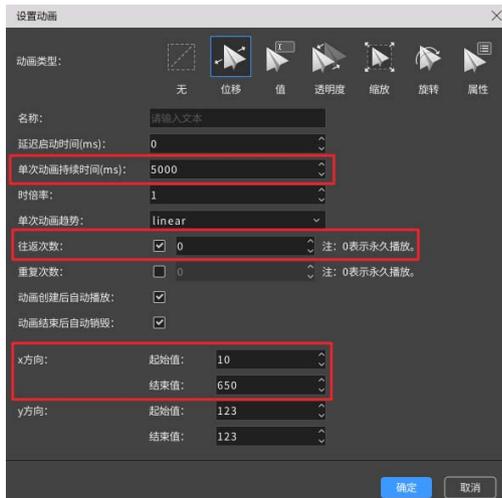


图8 使用AWTK Designer设置控件动画

在完成界面设计之后，我们点击 AWTK Designer 上方的“打包”按钮即可将页面布局文件、资源文件进行打包。

添加控件事件并编写事件处理代码

项目中我们需要用到按钮的跳转以及修改 edit 控件值同步改变进度条值和 label 控件显示文本的效果，我们可以用事件的方式来实现它们。

1. 使用AWTK Designer为控件添加事件

首先选择主页中的 button 控件，并在右侧的“控件编辑器”中选择“事件”选项，接着点击选项卡右上角的加号并选择“click”事件，添加后效果如图所示。

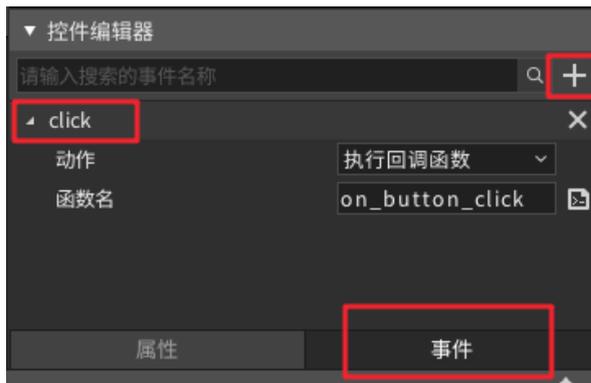


图9 添加控件click事件

第二个界面中的 button 控件也是像上面一样添加 click 事件、edit 控件添加 value_changed (值改变) 事件，最后不要忘了保存与打包操作。

2. 在VSCode编写控件事件处理代码

我们在 VSCode 打开 Hello 工程的目录，然后找到首页“home_page.c”代码文件，打开之后我们只需在 AWTK Designer 自动生成的事件代码中写处理逻辑即可。

因为点击按钮要跳转界面，所以 on_button_click 要用到 navigator_to 接口，参数填的是要打开窗口的 name 属性值。

```
static ret_t on_button_click(void* ctx, event_t* e) {
    // 打开另一个window, 参数填窗口的name属性值
    return navigator_to("new_window");
}

/**
 * 初始化窗口的子控件
 */
static ret_t visit_init_child(void* ctx, const void* iter) {
    widget_t* win = WIDGET(ctx);
    widget_t* widget = WIDGET(iter);
    const char* name = widget->name;

    // 初始化指定名称的控件 (设置属性或注册事件), 请保证控件名称在窗口上唯一
    if (name != NULL && *name != '\0') {
        if (tk_str_eq(name, "button")) {
            widget_on(widget, EVT_CLICK, on_button_click, win);
        }
    }
}
```

图10 事件处理函数和注册事件代码

在新窗口的按钮也是类似，要用到 `navigator_back_to_home` 接口，代表回到主页。

接下来编写 `edit` 控件的 `value_changed` 值改变事件处理函数代码，本质是获取用户输入的值，并将这个值设置到进度条和 `label` 控件中，代码如下：

```
9 static ret_t on_edit_value_changed(void* ctx, event_t* e) {
10     char str[20] = {'\0'};
11     widget_t* win = WIDGET(ctx);
12     // 获取label控件的widget_t对象
13     widget_t* label = widget_lookup(win, "label", TRUE);
14
15     // 获取进度条控件的widget_t对象
16     widget_t* progress_bar = widget_lookup(win, "progress_bar", TRUE);
17
18     // 获取edit的text属性值
19     value_change_event_t* evt = value_change_event_cast(e);
20     uint32_t value = value_uint32(&evt->new_value);
21
22     // 将value设置为进度条进度
23     progress_bar_set_value(progress_bar, (double)value);
24
25     // 更新label显示的值
26     tk_itoa(str, 20, (int)value);
27     widget_set_text_utf8(label, str);
28
29     return RET_OK;
30 }
```

图11 edit控件value_changed事件处理代码

到这里，我们就完成了 Hello 工程界面设计、业务逻辑代码编写以及调试运行，如果你在开发过程中遇到了问题，欢迎在评论区进行讨论。



运行与调试

1. 运行AWTK程序

在完成项目的界面布局设计以及业务逻辑代码编写之后，我们依次点击 AWTK Designer 上方的“打包”、“编译”和“模拟运行”按钮即可运行程序。

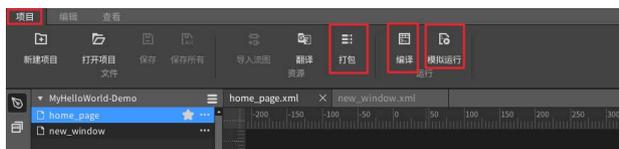


图12 打包、编译以及模拟运行项目

2. 调试AWTK程序

可以使用 Visual Studio Code 调试应用程序，具体操作步骤可以浏览官方开发实践文档的《2.7 应用调试》。

需要注意在 `launch.json` 配置 demo 程序的启动路径，如下图：

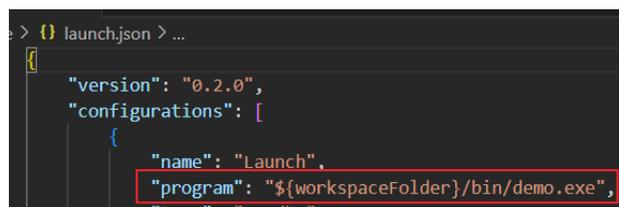


图13 配置VSCode的调试程序路径

【产品应用】基于ZWS云低代码平台的智慧工厂监控大屏应用

ZLG 致远电子 2023-11-02 11:41:51

智慧工厂是现代工厂信息化发展的新阶段，利用物联网服务，能够清楚掌握生产流程，提高生产过程的可控性。运用致远电子的 ZWS 云低代码平台，搭建工厂数据监控大屏，能够实时掌握工厂生产信息，使生产透明化。

应用场景

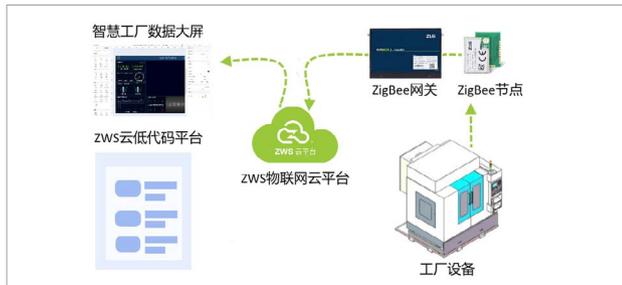
工厂可视化管理，是以工厂实际数据情况出发，衍生数据化、可视化大屏看板，打造数字可视化场景，实现了虚拟与现实工厂的实时同步。



工厂集中可视化监控的缺失，导致管理人员无法实时、全面、准确的得知各生产及相关环节的实际状况，更无法及时的排查处理故障，造成工厂管理的窘境。

智慧工厂是现代工厂信息化发展的新阶段，利用物联网服务，能够清楚掌握生产流程，提高生产过程的可控性，减少生产线上人工的干预，及时正确地采集生产线数据，以及合理的生产计划编排与生产进度。

致远电子的 ZigBee 网关和节点可以将工业设备的数据采集到 ZWS 物联网云平台，建立数字化智慧工厂。此外，通过致远电子的 ZWS 云低代码平台，可以自助搭建工厂不同业务的可视化监控大屏，实时掌握生产现场的生产进度和生产计划达成情况，以及生产的设备、物料、质量等相关信息，使生产完全透明化。



ZigBee网关及节点

ZLG 致远电子工业级 ZigBee 物联网网关 GZ32M，支持 MQTT 协议上云，适用于工业控制、智慧照明等多种应用场景。



点击购买

ZLG 致远电子 ZM32x 是基于 Silicon Labs EFR32 系列芯片设计的高性能 ZigBee 透传模组，模组有着强大硬件性能，高达 20dBm 的发射功率和 -103dBm 的接收灵敏度保证了模组的穿墙能力和超过 3.3 公里的视距通信距离。



点击购买

ZWS云低代码开发平台

ZWS 云低代码开发平台是一个以图形化、拖拽方式并辅以少量代码就能搭建数字化系统应用的工具平台。提供了常用的基础组件、工业组件（用于工业生产场景）、大屏组件（用于展示大量数据）以及业务组件（物联网相关业务），无需深入了解编程语言和开发技术，只需简单地拖拽组件即可快速搭建物联网数据监控大屏。



智慧工厂可视化监控大屏

通过低代码平台，自定义绑定数据源，自选组件，可以搭建出不同工厂数据可视化大屏。

1. 便于管理

作为数字化的一种表现方式，工厂可以查看不同版块的数据信息，可视化的数据将整体信息进行统一汇总整理，通过直观易懂的监控大屏呈现给用户，对智慧工厂的建设和管理有帮助。

2. 对外展示

数据监控大屏可以展示工厂自身的各项数据，还可以展示工厂报表，对客户展示工厂的生产力，从而获得市场的后续发展。

3. 改善维护

关联 ZWS 物联网云平台，可实时查询监控告警数据，能及时对设备发生故障告警，列出告警设备信息，并进行醒目提醒，以此提高对产品的改善和维护。



【产品应用】 基于ZWS云对LoRa网关的远程运维管理

ZLG 致远电子 2023-11-14 11:37:40

LoRa 网关作为工业物联网设备的接入和数据传输的重要组成部分，需要具备易于管理和维护的特点，方便管理员进行设备的配置、监控和管理。本文将介绍基于 ZWS 物联网云平台对 LoRa 网关的远程运维管理。

应用场景

GLCOM-NET 是致远电子研发的一款智能无线数据通信网关，采用 LoRa 无线技术，实现了自组网、无线数据透明传输至 TCP、UDP、MQTT、HTTP 及 RS485/232 等功能，广泛应用于工业物联网。



[点击购买](#)

LoRa 网关可用于连接各种物联网设备，如传感器、智能终端等，通过与这些设备进行通信，实现对工业环境、设备状态的监测和采集。LoRa 网关作为工业物联网设备的接入和数据传输的重要组成部分，需要具备易于管理和维护的特点，方便管理员进行设备的配置、监控和管理，例如，支持远程可视化配置、信道负载率分析等。

基于 ZWS 物联网云平台，GLCOM-NET 网关支持远程配置、远程升级、远程监测、通信统计等一系列远程维护功能。本文将介绍基于 ZWS 物联网云平台对 LoRa 网关的远程运维管理。

基于ZWS云平台远程运维管理LoRa网关

1. 在线集中管理LoRa网关

可以远程查看 lora 网关的在线、离线情况，统一管理网关。

序号	状态	设备名称	设备类型	设备ID	ID	上线时间	操作
1	●	lora网关3	LORAWAN_GATEWAY	bc12578112175	1053765	2023-09-26 16:16	查看 密钥 删除
2	●	lora网关2	LORAWAN_GATEWAY	bc12578112155	1053764	2023-09-26 16:16	查看 密钥 删除
3	●	lora网关1	LORAWAN_GATEWAY	bc12578111368	1053762	2023-09-26 16:16	查看 密钥 删除
4	●	lora网关	LORAWAN_GATEWAY	b1257811029	1029	2023-09-26 16:16	查看 密钥 删除

2. 远程配置LoRa网关

可以在线远程配置 LoRa 网关的频段、功率等参数。



3. 远程升级LoRa网关

可以通过云端远程升级 LoRa 网关的固件，进行功能更新或修复漏洞。

序号	设备类型	固件版本	固件类型	描述	ID	↓	操作
1	LORAWAN_GATEWAY	1.0.2	自身固件	支持协议栈新功能	390	2023-09	升级 删除
2	LORAWAN_GATEWAY	1.0.0	自身固件	支持本地配置	389	2023-09	升级 删除

4. 远程下发指令

可以通过云端给 LoRa 网关下发控制命令。



5. 网关通信统计

可以远程统计 LoRa 网关的上下行数据帧、信道负载率以及空闲信道扫描。



【产品应用】ZWS工商储能管理云平台，实现智能化运营，拓展收益

ZLG 致远电子 2023-11-16 11:38:07

数字化，云服务化，已经成为储能产品的刚需。那 ZWS 工商储能管理云平台，是如何赋能于工商业储能产品的？如何实现智能化运营，拓展业主单位收益的？我们一起来看看吧。

工商储能行业的火热

过去两年，国内储能市场的火热程度有目共睹。随着各地峰谷电价差持续拉大与电池价格下降，使工商业储能的经济性，引起工业用电业主的关注。从而众多的储能企业，加大对于工商储能产品的研发与市场的投入，各类工商储能产品投入市场。

如何让业主单位更智能、更安全的运营电站，获取更多的收益，也成了储能企业考虑的核心问题。ZWS 工商储能管理云平台，适配致远电子推出的 EM 系列储能控制器，为用户提供云端数字化运维的工商储能管控方案，助力推动储能系统安全运维，拓展收益。



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](#)

工商储能现状与痛点

在储能电站的运营中，主要通过“低电价为储能系统进行充电，高电价时放电用于企业自行用电”，降低成本使业主收益最大化。因这差价形成的经济收益，对于业主的运营管理，极为重要，需要考虑的因素居多。低电价时充电，需要充多少电，功率需要多少？一天能完成多少次充放电，才达到收益最大化？如何根据这些运营数据，进行统计分析，并可实时进行调整？

储能系统，运维也相对重要，储能系统对于实时监控性要求较高的系统产品，需要动态监控储能系统的运行状况和告警信息，做到预测性运维，保障储能系统的运行安全。

以下为工商储能的痛点：

- 储能电站现场，需人力轮值监控，巡检记录纸笔化，运营效率低效；
- 缺少对储能电站的统一监测、管理、运维；
- 需根据电站的运营情况，实时按需调整电站运行策略；
- 电站安全监控需求高，电池健康度需时刻关注；
- 电站运营数据难以统计并分析，运营信息延迟。

ZWS工商储能管理云平台赋能

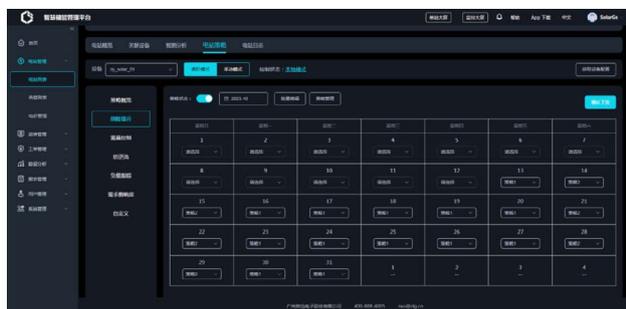
ZWS 工商储能管理云平台，为工商储能用户赋能，支持远程监控电站各设备数据、统计分析、远程安全运维、削峰填谷、需量控制、防逆流、自定义策略等，支持通过 PC 端、移动端 APP，查看电站中的设备（电表、PCS、BMS、空调等）实时运行参数，减少实地运维的困难性，实现运维降本增效。



1. 统计电站的充放电、能耗、收益等，以日 / 月 / 年维度，对储能数据进行可视化分析。



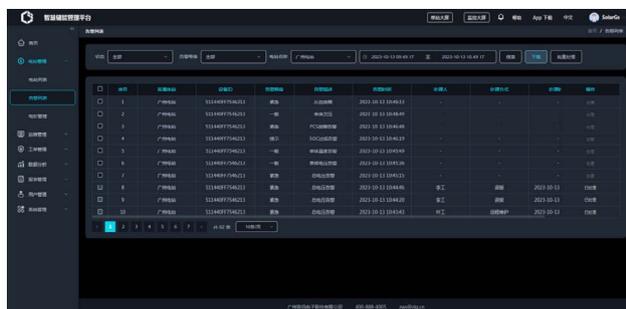
2. 可以提供削峰填谷、需量控制、防逆流等策略，从而最大化业主收益，并进一步提升储能系统稳定性。



3. 可以进行电池分析，提供储能电池的电压、电流与温度分析，查看健康度低的单体，对电池进行维护。



4. 可以动态监控电站运行，预测性故障报错，第一时间获取电站故障信息，让电站时刻处于安全运行状态。



【产品应用】

基于ZWS云对LoRa网关的远程升级

ZLG 致远电子 2023-11-17 11:37:00

当设备出现软件程序问题或者更新时，如果按照传统的解决方式，是一项很繁重的任务。为了使得软件更新更迅速，致远电子的LoRa网关GLCOM-NET，支持远程升级固件。

应用场景

GLCOM-NET 是致远电子研发的一款智能无线数据通信网关，采用LoRa无线技术，实现了自组网、无线数据透明传输至TCP、UDP、MQTT、HTTP及RS485/232等功能，广泛应用于工业物联网。



点击购买

当设备出现软件程序问题或者更新时，如果按照传统的解决方式，是一项很繁重的任务。以某智慧工厂LoRa网关出现软件问题为例，按照传统的解决方案，需要派人到现场更新软件，这样，一方面影响用户体验和满意度，另一方面又要耗费大量的人力和物力来修复问题。

为了使得软件更新更迅速，致远电子的LoRa网关GLCOM-NET，支持远程升级固件。远程升级网关分三步：

1. 将待升级的固件包上传到ZWS云；
2. 云端对比版本并下发升级通知；
3. 网关下载升级包进行自动升级。

远程升级的优势

OTA在线升级，空中下载技术，通过无线传输的方式对设备软件进行在线更新，解决传统升级方式的痛点，可以看出远程升级的优势：

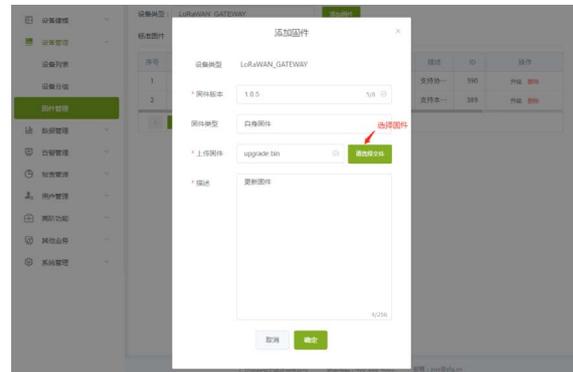
1. 可以有效提升用户体验和满意度；
2. 可以大范围大批量升级设备；
3. 可以快速修复设备故障；
4. 可以有效降低售后维护成本。

如何远程升级LoRa网关

首先，登录ZWS物联网云平台，进入“固件管理”页面。



然后，在自定义固件页面，点击“添加固件”，将升级包上传到云端。



成功上传固件后，点击“升级”，选择升级的范围和升级方式。



比如，可以对单个设备升级，或者分组批量升级，还有立即升级和定时升级可选。



最后，设备端收到升级通知后，就会下载固件包进行自动升级。成功升级后，设备上报当前版本，可以在设备详情中查看版本是否为需升级的版本。



【产品应用】 ZWS工商储能云平台，实现智能化削峰填谷

ZLG 致远电子 2023-11-23 11:37:12

ZWS 工商储能云平台，根据分时电价，统计储能系统收益，让收益透明化，并可实现云端智能化削峰填谷，拓展收益。

削峰填谷介绍

目前国内广泛推行分时电价，根据尖、峰、平、谷的分时段分电价，实现不同的时段不同的收费电价。其中电价收费按尖、峰、平、谷的价格，依次递减。对于工商储能系统而言，可以在谷的时候进行充电，在尖或峰时段，进行放电，赚取电价的差值，拓展业主单位的收益。

ZWS 工商储能云平台，搭配致远电子 EM-1000 储能网关，可根据不同时段不同的收费电价，统计储能系统的收益，让收益透明化，并可实现云端智能化削峰填谷，拓展收益。



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

ZWS工商储能云平台赋能

ZWS 工商储能云平台，为工商储能用户赋能，支持远程监控电站各设备数据、统计分析、远程安全运维、削峰填谷、需量控制、防逆流、自定义策略等，支持通过 PC 端、移动端 APP，查看电站中的设备（电表、PCS、BMS、空调等）实时运行参数，减少实地运维的困难性，实现运维降本增效。

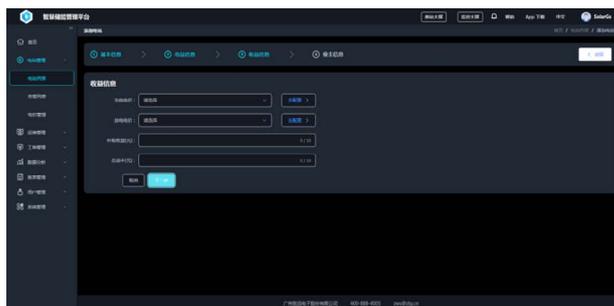


1. 统计电站收益

1.1 用户可在电价管理中加相关地区的收费规则，其中填写充、放电的尖、峰、平、谷的相关时间段与电价。



1.2 在创建或编辑电站时，让电价规则与电站进行关联，以电站的维度，对储能系统的数据进行统计。



1.3 可在云端对电站的收益情况，以日/月/年，不同维度的查看，并支持以报表形式进行下载。

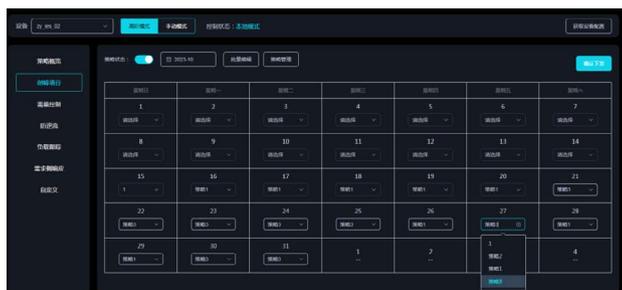


2. 电站策略-削峰填谷

2.1 以天的维度，设定具体的时间段与充放功率等，形成具体当天策略。



2.2 相应的日期，选定相应的策略，将该削峰填谷边缘策略下发至储能系统中，储能系统监控相应的 PCS 运行，完成边缘策略。



【产品应用】如何将工业设备快速接入到ZWS物联网云平台

ZLG 致远电子 2023-11-29 11:38:13

工业设备作为工业互联网的底层终端，承担着数据感知和智能控制的重要作用。本文将介绍如何将工业设备快速接入到ZWS物联网云平台。

应用场景

工业现场，分布着许多类型的工业设备，比如生产设备有机床、冲压机、注塑机、机器人等；感知设备有温度传感器、压力传感器、流量计等。这些设备在运行过程中会产生大量的数据，如何将这些数据能够实时传输到云端，利用数据驱动精益管理和生产，是一个亟待解决的问题。ZWS物联网平台，搭配致远电子CATCOM-100智能网联终端产品，能将工业设备数据快速接入上云，从设备建模、设备管理、数据分析、云组态搭建等多功能，赋能企业实现设备精益管理和生产。

CATCOM-100是一款工业级智能网联终端，通过RS-485等接口与工业设备连接，然后通过Cat.1与物联网云平台通信，可快速实现设备联网。终端采用Cat.1通信标准，最高支持上行5Mbps、下行10Mbps的通信速率，与Cat.4(LTE 4G)网络共用基站，保障信号覆盖范围。CATCOM-100适用于环境监测、电力配电、智慧工厂、智慧农业等多种应用场景。



[点击购买](#)

工业设备接入ZWS物联网云平台方案

工业设备通过串口与致远电子CATCOM-100智能网联终端连接，CATCOM-100会将设备上传数据透传至ZWS物联网云平台，设备管理员能够在PC端和移动端监测到设备实时运行数据，企业管理层可以通过智慧监控大屏掌握工厂全局设备运行状况。



将工业设备接入ZWS云平台操作流程

1. 创建设备类型

登录ZWS物联网云平台，点击系统左侧菜单“设备建模-设备类型”，选择“自定义设备类型”，点击“添加设备类型”创建工业设备类型。



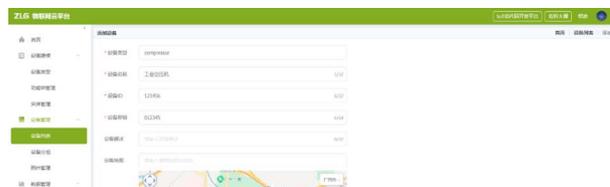
2. 添加功能块

给创建好的工业设备添加功能块，同时在功能块里面配置好设备的数据、状态、命令、日志等信息。例如，在“工业空压机”功能块里添加排气量、排气压力、功率等数据项。



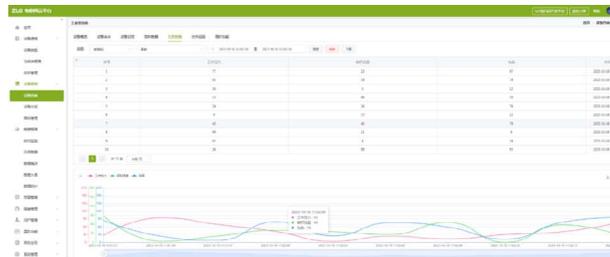
3. 添加设备

点击系统左侧菜单“设备管理-设备列表”，进入设备列表页面，点击页面左上角“添加设备”按钮，进入到添加设备页面，在“设备类型”选项选择刚刚创建好的工业设备类型、输入“设备名称”、“设备ID”、“设备密钥”和“描述”。点击“确定”按钮，即可在设备列表查看到对应工业设备。



4. 查看实时/历史数据

添加完设备后回到设备列表页面，选择要查看的工业设备，点击“查看”按钮，进入到设备详情页。点击“实时数据”，选择好数据组和数据项，系统会展示工业设备当前的实时数据和数据变化曲线图。点击“历史数据”，还可以查看工业设备的历史数据。



【新品发布】 小型高性能边缘计算网关，小身板，大能量！

ZLG 致远电子 2023-11-20 11:38:01

自 SX-3568LI 系列主板面市后，受到了众多客户的热烈追捧，同时也收到了客户的进一步需求，于是我们设计了全新的 MD-3568LI 系列主板，以适应更多的客户应用需求。



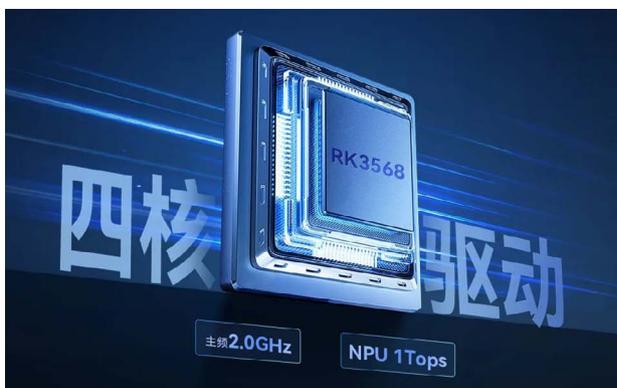
选型表

类别	名称	主要参数
最小系统	处理器	RK3568, 四核 64 位 Cortex®-A55, 主频 2.0GHz
	内存	2GB DDR4
	存储	8GB eMMC
电源	工作电压	DC 9V-24V
有线通讯	RS485	2 路
	RS232	1 路
	CAN	1 路隔离
	以太网	2 路, 10/100/1000Mbps (RJ45)
无线通讯	WiFi	1 路, 支持 2.4G
	Bluetooth	1 路, WiFi 模块支持
扩展存储	TF 卡	1 路, 标准 TF 卡座
USB 接口	USB2.0	2 路
	USB3.0	1 路
显示接口	HDMI	1 路, HDMI2.0
音频接口	Audio	1*Headphone, 3.5mm 音频插座

类别	名称	主要参数
其他	DI	2 路无源型, 均带光电隔离
	DO	2 路输出, 光耦隔离, 用户侧需接外部上拉
	miniPCIe 插座	可支持扩展 LoRa、ZigBee 模块
	LED 指示灯	4 路, 其中 1 路可编程
工作环境	工作温度	-40° C ~ +60° C
	存储温度	-40° C ~ +60° C
	相对湿度	5% ~ 95% (无凝露)
软件规格	预装系统	Debian
	可支持编程框架	TensorFlow/Caffe/Darknet 等
	网络服务	支持 DHCP 服务器 / 客户端、动态域名 DDNS、SSH、Telnet、HTTPS、TFTP
	数据安全	OpenVPN、OpenSSL、WireGuard 等
	支持多种云平台	致远 ZWS 云、阿里云、腾讯云等
	支持多种工业协议	Modbus TCP/RTU 协议、OPC UA 协议等
物理规格	安装方式	壁挂式、导轨式
	机械尺寸	110x73x44.5mm

四核驱动，边缘计算很强劲

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



多样接口，数据采集很轻松



支持多种云平台，数据上云很便捷

EPCM3568A-LI 支持对接多种云平台，实现云端与本地协同管理和控制。如：ZWS 云、阿里云、腾讯云、涂鸦云、机智云等云平台。默认支持致远 ZWS 云接入。



主流开源系统，多通讯协议

预安装 Debian 系统

支持 C/C++、python、Java、node.js 等众多开发语言

支持第三方 SDK 接入

预安装 openVPN、openssl 库，面向数据保障

预安装 sqlite3 数据库，支持数据存储服务，

精致小巧，安装很灵活

EPCM3568A-LI 具有小巧精致的外观，安装方式支持导轨式和挂耳式两种，非常方便用户的灵活使用安装。



典型应用场景



能源应用场景



工业能源

监测各类设备（如锅炉、热泵、照明系统等）的能耗、效率等参数，实时监测能源的消耗情况，提供能源管理和优化建议，降低能源成本 and 环境影响。

光伏发电站

监测每个光伏组件的发电功率、电流、电压等参数，以及整个场站的总发电量、能源利用效率等，实时监控光伏系统的运行情况，提供故障诊断和性能评估。

暖通领域

连接暖通设备，实时监测设备的能耗和运行状态，优化能源利用效率。

EPCM3568A-LI 边缘计算网关

点击购买

【新品发布】

ZLG致远电子PCIe EtherCAT通讯卡产品正式发布!

ZLG 致远电子 2023-11-06 11:41:36

自 SX-3568LI 系列主板面市后，受到了众多客户的热烈追捧，同时也收到了客户的进一步需求，于是我们设计了全新的 MD-3568LI 系列主板，以适应更多的客户应用需求。

PCIe EtherCAT通讯卡

ZLG 致远电子 PCIe EtherCAT 通讯卡适用于半导体设备、机器人、激光加工、生物医疗、3C 制造和装配、锂电池包装、数控机床、流水线工作站、非标自动化装备、特种机床等不同智能制造设备应用需求。



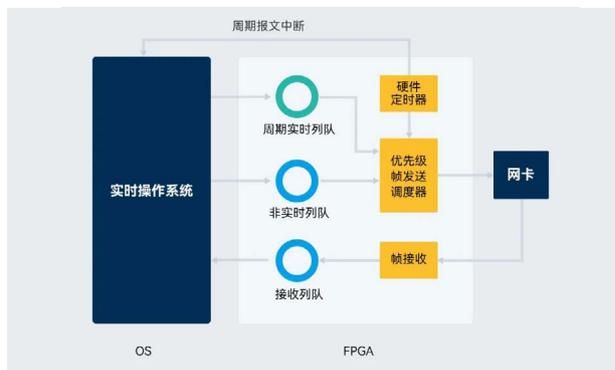
高实时性的网络技术，助您抢占先机，决胜未来!

PCIe EtherCAT 通讯卡控制周期最快可达 125us，满足智能设备的高实时性控制要求。



商业授权的EtherCAT协议栈，稳定性更有保证

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



选型表

名称	HP PCIe-2E	PCIe-2E	PCIe-4E	MiniPCIe-2E
EtherCAT	1 路主站, 2 网口	1 路主站, 2 网口	2 路主站, 4 网口	1 路主站, 2 网口
线路冗余	支持	支持	支持	支持
热插拔	支持	支持	支持	-
CAN	-	-	-	-
用户 I/O 输入	-	-	-	-
用户 I/O 输出	-	-	-	-
编码器	-	-	-	-
PWM 输出	-	-	-	-
安装方式	固定孔螺丝安装	固定孔螺丝安装	固定孔螺丝安装	固定孔螺丝安装
结构	PCIe 半高	PCIe 全高	PCIe 全高	MiniPCIe
产品尺寸	137.5mm × 68.92mm	137.5mm × 111.15mm	137.5mm × 111.15mm	30mm × 50.95mm

使用EtherCAT电缆冗余构建强大，且故障安全的工业以太网网络

ZLG 致远电子 PCIe EtherCAT 通讯卡支持 EtherCAT 线缆冗余功能，即使电缆在某一点物理中断，也可以保持主站设备和从站设备之间的通讯。



AWStudio EtherCAT配置软件

配套的AWStudio (运动控制版本) 软件提供了EtherCAT的配置功能, 用户通过设备扫描或者手动配置EtherCAT从机设备来快速得到ENI的配置配置文件, 降低用户使用EtherCAT设备的门槛, 提高开发效率。



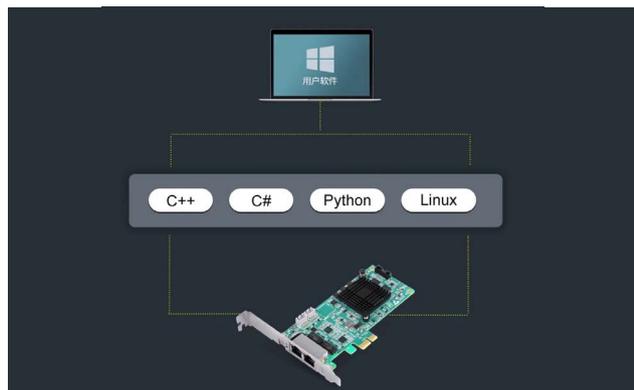
严格测试稳定可靠, 品质更有保证

PCIe EtherCAT 通讯卡均通过六大科研实验室严格的实践验证, 保证在多种恶劣环境下仍然稳定可靠。



高效易用的二次开发函数

PCIe EtherCAT 通讯卡提供二次开发库, 可支持各类开发环境, 如VC、VB、C#、Linux、Python、Windows 等主流开发环境, 用户只需修改设备类型即可实现不同接口的无缝切换, 高效易用。



【技术分享】 小型边缘计算网关设备如何快速上云

ZLG 致远电子 2023-11-21 11:35:57

本文教你使用 ZWS 云平台提供的 SDK，快速将小型边缘计算网关设备接入 ZWS 云平台。话不多说，跟着操作起来吧！

前期准备

1. 下载 ZWS 云平台提供的 ZWSR2_MQTT_SDK 【SDK 下载地址：zlgcloud.oss-cn-shenzhen.aliyuncs.com/sdk/zws_sdk_r2.zip】；
2. 下载 EPCM3568A-LI 开发环境 【下载地址：manual.zlg.cn/web/#/269/10177】；
3. EPCM3568A-LI 小型边缘计算网关；
4. 网线。

编译ZWSR2_MQTT_SDK

1. SDK目录说明

名称	修改日期	类型	大小
3rd	2023/6/27 10:12	文件夹	
demos	2023/6/27 10:12	文件夹	
vs_build	2023/6/27 10:12	文件夹	
zws_sdk	2023/6/27 10:12	文件夹	
arm-linux.mk	2023/9/8 14:00	Makefile 源文件	1 KB
CMakeLists.txt	2023/9/12 11:39	文本文档	4 KB
em-500.mk	2023/9/12 11:21	Makefile 源文件	1 KB
em-1000.mk	2023/9/6 14:58	Makefile 源文件	1 KB
epcm3568.mk	2023/9/12 11:21	Makefile 源文件	1 KB
Makefile	2023/4/6 11:38	文件	13 KB
msvc_gen.bat	2023/1/31 17:16	Windows 批处理...	1 KB
ReleaseNote.md	2023/9/13 16:20	Markdown File	2 KB
ZWSR2_MQTT_SDK使用手册.pdf	2023/9/12 19:04	WPS PDF 文档	4,943 KB

MQTT SDK 是用 C 语言封装了一些连接云的底层接口和提供了一些必要的工具接口，用户可以利用这些封装好的接口可以快速开发自己应用到 ZWS 云。同时 SDK 也提供了不同类型的设备接入云的例子，可供用户参考，从而更加方便用户了解设备接入云的流程。

- 3rd, SDK 中使用到第三方库的目录。
- zws_sdk, 接入 ZWS 的 mqtt sdk 目录。
- demos, 演示 mqtt sdk 的例子。
- vs_build, 指的是工程文件目录。

2. ARM Linux交叉编译MQTT SDK

- 将 zws_sdk_r2.zip 压缩包拷贝到提前下载好的 Ubuntu20.04 虚拟机开发环境下并进行解压，然后进入源码目录执行如下 cmake 命令：

```
zlg@arm-multicross:~/ws/zws_sdk_r2/mqttproto_V2.0.1.230913$  
cmake -DEPCM3568=1
```

然后执行 make 进行编译即可。

```
zlg@arm-multicross:~/ws/zws_sdk_r2/mqttproto_V2.0.1.230913$  
make
```

编译成功后，可以看到 ./bin 目录下生成多个二进制文件，示例如下：

```
zlg@arm-multicross:~/ws/zws_sdk_r2/mqttproto_V2.0.1.230913/bin$ ll  
总用量 2448  
drwxrwxr-x 2 zlg zlg 4096 11月 10 16:41 /  
drwxrwxr-x 8 zlg zlg 4096 11月 10 17:33 /  
-rwxrwxr-x 1 zlg zlg 257792 11月 10 16:41 appuser_example*  
-rwxrwxr-x 1 zlg zlg 259744 11月 10 16:41 device_example*  
-rwxrwxr-x 1 zlg zlg 225072 11月 10 16:41 download_firmware_example*  
-rwxrwxr-x 1 zlg zlg 232832 11月 10 16:41 download_package_example*  
-rwxrwxr-x 1 zlg zlg 36512 11月 10 16:41 echoc*  
-rwxrwxr-x 1 zlg zlg 264624 11月 10 16:41 gateway_example*  
-rwxrwxr-x 1 zlg zlg 70472 11月 10 16:41 http_auth*  
-rwxrwxr-x 1 zlg zlg 52616 11月 10 16:41 http_download*  
-rwxrwxr-x 1 zlg zlg 227216 11月 10 16:41 http_transfer_chunk*  
-rw-rw-r-- 1 zlg zlg 100218 11月 10 16:41 libclient.a  
-rw-rw-r-- 1 zlg zlg 28440 11月 10 16:41 libjsmn.a  
-rw-rw-r-- 1 zlg zlg 28440 11月 10 16:41 libjsmn_wrapper.a  
-rw-rw-r-- 1 zlg zlg 354436 11月 10 16:41 libnetwork.a  
-rwxrwxr-x 1 zlg zlg 119744 11月 10 16:41 mqtt_demo*  
-rwxrwxr-x 1 zlg zlg 218208 11月 10 16:41 upload_file_example*
```

将 bin 目录打包好，稍后拷贝到 EPCM3568A-LI 设备上。

```
zlg@arm-multicross:~/ws/zws_sdk_r2/mqttproto_V2.0.1.230913$  
tar cvf bin.tar bin/
```

设备接入ZWS云端

1. 设置ZWS云端

1.1 云账号注册

首先打开 zws.zlgcloud.com 网站，若还未注册帐号，则点击用户注册，注册一个新账号。注册界面如下：





1.2 创建设备类型

帐号注册好之后登录, 来到 ZWS 物联网云平台主页, 依次点击左侧菜单的“设备建模”->“设备类型”->“自定义”->“添加设备类型”进行自定义创建设备类型。如下图所示。



注意: 设备类型名称在整个系统中是全局唯一不允许重名, 因此用户在参考此例子添加类型时要自己起一个新的名字, 否则创建不成功。

1.3 创建设备

添加好设备类型后, 依次点击左侧菜单栏“设备管理”->“设备列表”->“添加设备”。如下图所示。



选择刚刚创建好的设备类型, 设备名称、设备 ID 可以由用户自行定义, 设备 ID 不能与已存在的设备 ID 重复。



设备创建完成后, 即可在设备列表看到刚创建的设备了, 此时设备处于离线状态。

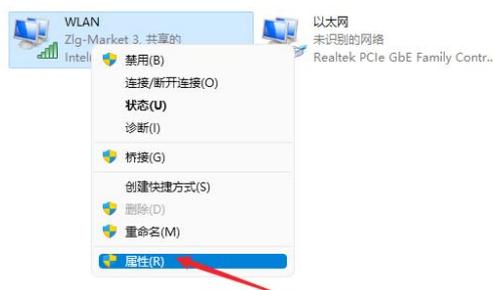


2. 设备端接入ZWS云

2.1 设备端网络配置

EPCM3568A-LI 设备上电, 使用网线将 EPCM3568A-LI 设备的 NET1 网口与电脑 (可以连接外网的主机) 连接, 并按如下步骤配置上网。

2.1.1 电脑端连接 wifi, 配置将无线网卡共享给以太网。





2.1.2 将以太网 IP 地址修改为 192.168.0.1。



2.1.3 使用 SSH 登录 EPCM3568A-LI 设备后配置以下路由：

```
zlgmcu@EPCM3568:~$ sudo route add default gw 192.168.0.1
```

2.1.4 配置完成后，EPCM3568A-LI 即可连接外网了。

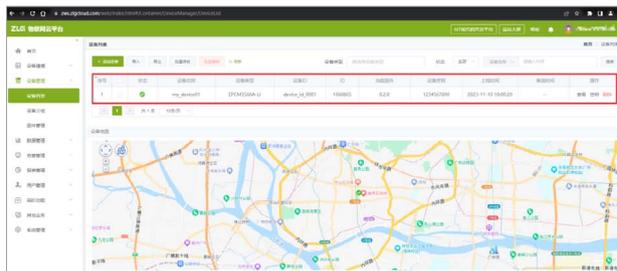
```
zlgmcu@EPCM3568:~$ sudo ping www.baidu.com
PING www.a.shifen.com (14.119.104.254) 56(84) bytes of data.
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=1 ttl=53
time=10.1 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=2 ttl=53
time=42.1 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=3 ttl=53
time=19.0 ms
```

2.2 设备接入演示

将打包好的 bin.tar 拷贝到设备上并解压，然后进入 bin 目录执行如下命令启动 device_example 程序。

```
zlgmcu@EPCM3568:~/bin$ ./device_example EPCM3568A-LI device_id_0001 1234567890
```

程序启动后，此时在云端便可以看到设备处于在线状态，至此设备已经成功接入 ZWS 云平台了。



【产品应用】 基于EPCM3568A-LI的LoRa快速使用指南来了

ZLG 致远电子 2023-11-28 11:34:28

本文采用 EPCM3568A-LI 小型边缘计算网关设备搭载 PCIE-ZSL421 模块，给大家演示实现设备间的透明数据传输、无线空中配置、自组网功能操作过程，帮助用户快速入门 LoRa 应用开发。

简介

EPCM3568A-LI 是 ZLG 致远电子面向小体积工控机应用市场，研发的国产化小型高性能边缘计算网关产品。

EPCM3568A-LI 主控具有 4 个主频高达 2GHz 的 Cortex-A55 内核，处理器内置独立的 NPU，板载 2GB DDR4 内存，8GB eMMC 存储器。EPCM3568A-LI 拥有丰富的功能接口，包括 WiFi、蓝牙、双路千兆以太网、双路 USB2.0 HOST、一路 USB3.0 HOST、HDMI、3.5mm 耳机、TF 卡接口，2 路 RS485、1 路 RS232、1 路 CAN、2 路光耦隔离 DI。



ZSL420/ZSL421 是广州致远微电子研发的两款 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。芯片采用 LGA68 封装，尺寸 11*11*1mm，非常适合用于无线通信组网节点。



演示一下

1. 前期准备

EPCM3568A-LI 小型边缘计算网关设备；
PCIE-ZSL421 LoRa 模块，实物图如图 1 所示；

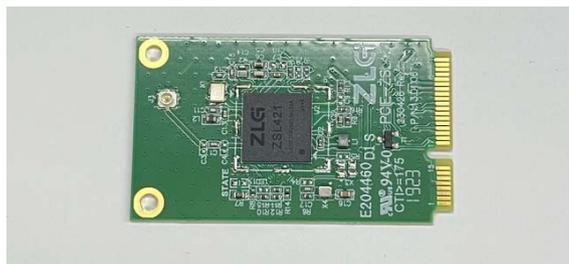


图1

为了方便演示，本文还将借助 ZSL42x-EVB Board 评估板以及与其配套的无线配置工具进行操作。ZSL42x-EVB Board 评估板实物图如图 2 所示；

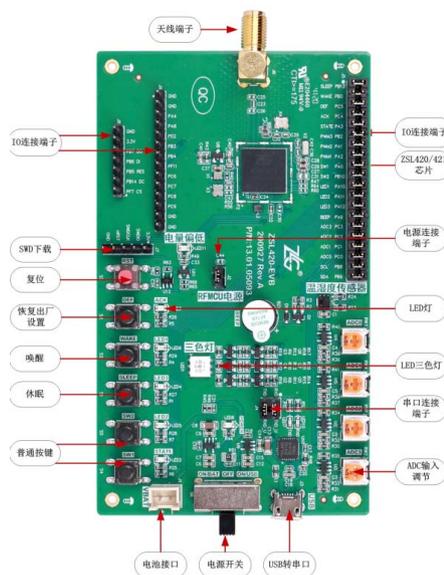


图2

无线模块配置工具【下载地址：https://www.zlg.cn/data/upload/software/Wireless/WirlessTool_V1.0.24.rar】。

2. 配置工具

2.1 把 ZSL42x-EVB Board 评估板通过 USB 线连接电脑，并安装驱动，在电脑上将出现 USB 转串口的设备。安装并打开无线配置工具 Wireless Tool，点击【连接串口】按钮，选择连接 ZSL42x-EVB Board 评估板的串口，串口参数按照模块配置的串口参数设置（模块出厂的默认参数波特率是 115200、8 位数据位、1 位停止位、无校验），点击【连接】按钮，如图 3 所示。

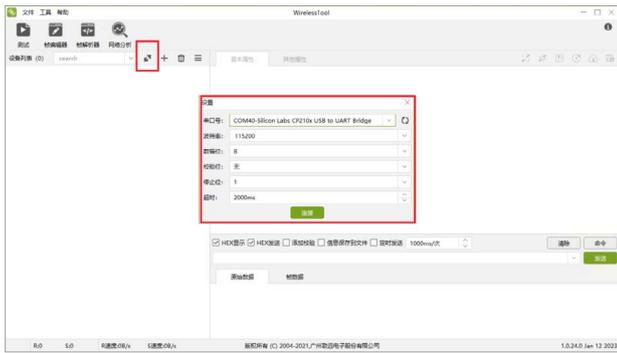


图3

2.2 设备连接成功，在左边设备栏本地设备里可以看到连接成功的设备，通过双击左边的设备，可获取模块的配置信息，如图4所示。将设备地址设置为20 01，设置信道为信道1，使能透传功能，设置目标地址为20 02，点击【保存设置】即完成模块的配置，如图5所示。

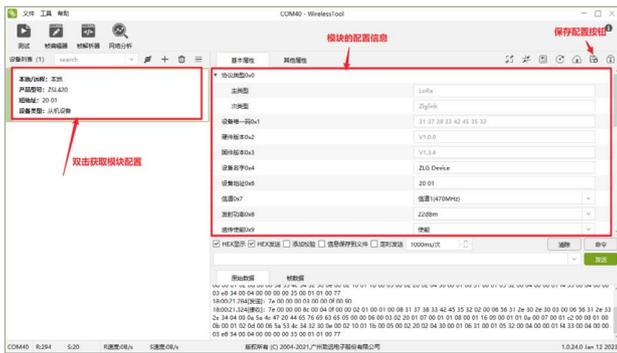


图4

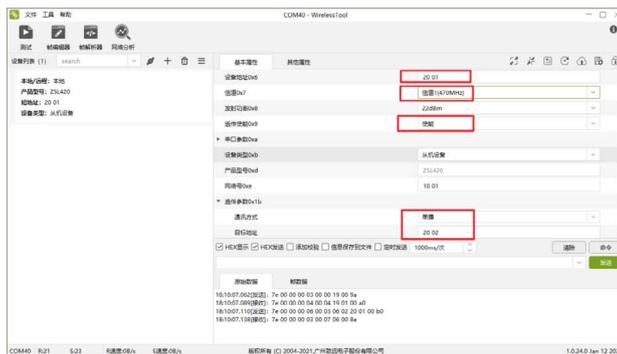


图5

3. 数据透传

3.1 将 PCIE-ZSL421 模块插入 EPCM3568A-LI 设备后，上电启动。PCIE-ZSL421 模块出厂自带固件，因此设备上电后，模块即进入正常工作状态，此时可以看到 EPCM3568A-LI 设备的 STA 灯闪烁。

3.2 EPCM3568A-LI 设备使用 /dev/ttyS9 串口设备与 PCIE-ZSL421 模块进行通信，输入以下命令对 PCIE-ZSL421 模块进行配置。

```
# 设置串口波特率
stty -F /dev/ttyS9 raw speed 115200 cs8 -parenb -cstopb -echo min 0
time 100

# 设置设备地址：20 02
echo -n -e "\x7e\x00\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttyS9
echo -n -e "\x7e\x00\x00\x00\x06\x00\x03\x06\x02\x20\x02\x00\xb1"
> /dev/ttyS9

# 打开数据透传模式
echo -n -e "\x7e\x00\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttyS9
echo -n -e "\x7e\x00\x00\x00\x04\x00\x03\x09\x01\x00\x8f" > /dev/
ttyS9

# 设置目标地址：20 01
echo -n -e "\x7e\x00\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttyS9
echo -n -e "\x7e\x00\x00\x00\x08\x00\x03\x1b\x00\x02\x20\x01\x04\
x00\xcb" > /dev/ttyS9

# 设置信道为信道 1
echo -n -e "\x7e\x00\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttyS9
echo -n -e "\x7e\x00\x00\x00\x04\x00\x03\x07\x01\x00\x8d" > /dev/
ttyS9
```

3.3 配置完成后，EPCM3568A-LI 设备与 ZSL42x-EVB Board 评估板间就可以当做串口一样进行数据透明传输了，如图6所示。



图6

4. 无线空中配置

4.1 通过以上数据透传的配置步骤后，此时可以在无线配置工具页面点击【添加远程设备】，在弹出的搜索设备对话框里填入搜索的超时时间，点击【搜索】按钮，便可以搜索到 EPCM3568A-LI 这个设备节点，如图7所示。

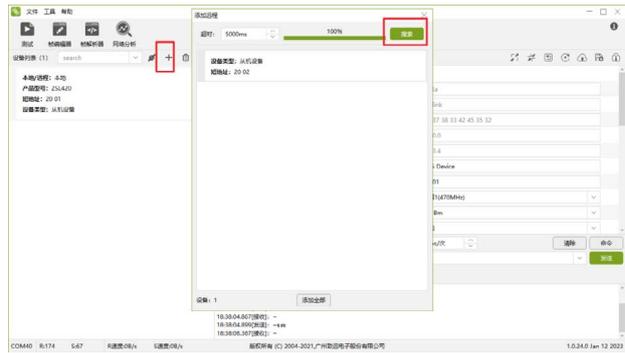


图7

4.2 等待本地模块搜索完成，点击【添加全部】把远程设备添加到设备列表栏里，此时便可以对设备进行远程配置了，如图8所示。

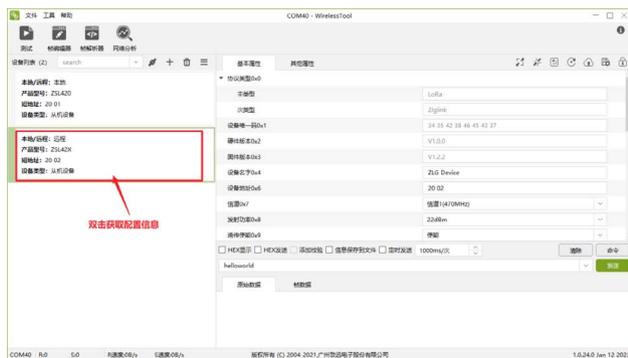


图8

5. 自组网功能

5.1 将 ZSL42x-EVB Board 评估板的设备类型配置成主机设备、使能自组网功能，如图 9 所示。

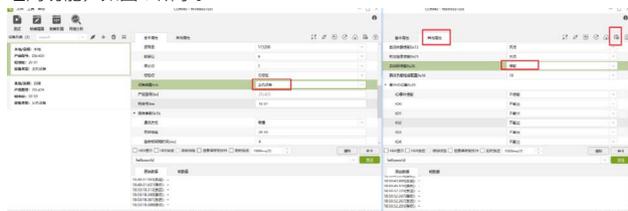


图9

5.2 将 EPCM3568A-LI 通过远程配置使能自组网功能，如图 10 所示。使能自组网后可看到 EPCM3568A-LI 设备的 STA 灯间隔 3 秒双闪。

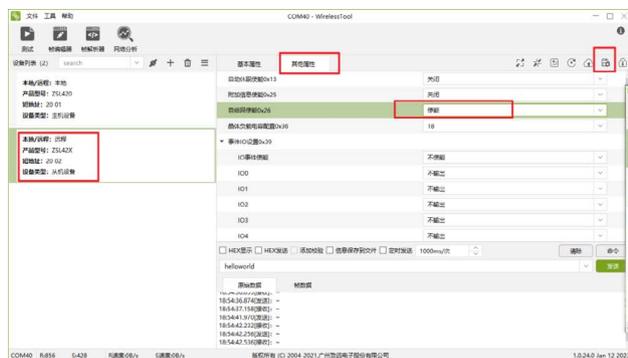


图10

或者通过如下命令集设置使能自组网功能。

```
# 使能自组网
echo -n -e "\x7e\x00\x00\x03\x00\x00\x19\x00\x9a" > /dev/ttyS9
echo -n -e "\x7e\x00\x00\x00\x04\x00\x03\x26\x01\x00\xac" > /dev/ttyS9
```

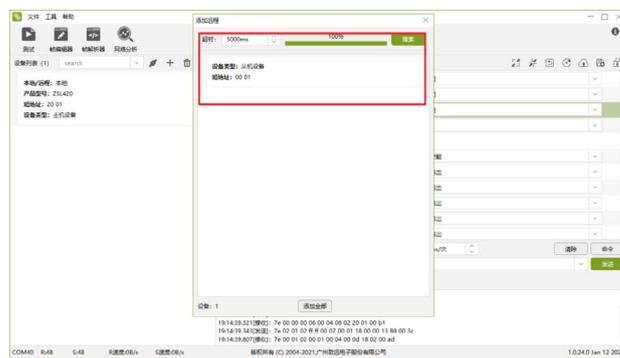
5.3 在 EPCM3568A-LI 设备端输入如下命令导出 JOIN 引脚。

```
echo 508 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio508/direction
echo 1 > /sys/class/gpio/gpio508/value
```

5.4 按住 ZSL42x-EVB Board 评估板 SW1 按键不松手，此时 ZSL42x-EVB Board 评估板主机模块允许从机模块加入网络。同时在 EPCM3568A-LI 设备端输入如下命令，拉低 JOIN 引脚小于 3s 进入入网申请状态。

```
echo 0 > /sys/class/gpio/gpio508/value
sleep 2
echo 1 > /sys/class/gpio/gpio508/value
```

5.5 当 EPCM3568A-LI 设备作为从机设备成功入网后 STA 灯间隔 3 秒单闪，此时在无线配置工具页面删除原本的远程节点，重新搜索添加，会发现 EPCM3568A-LI 设备节点的设备地址被自动重新分配了，如图 11 所示。至此，设备便组网成功了。



演示到此结束，快开始用 LoRa 实现自己的物联网应用吧！



EPCM3568A-LI边缘计算网关

[点击图片](#)



**LoRa智能组网芯片
ZSL420/421**

[点击图片](#)

【产品应用】快速实现PLC的Modbus数据转换为CAN/CAN FD数据?

ZLG 致远电子 2023-11-07 11:36:27

在现代工业自动化中，经常会有 PLC 直接控制 CAN/CAN FD 接口设备的需求，但是一般的小型或中型 PLC 很少有 CAN 接口，一般标配的是 RS485 接口，那该如何解决呢？

PXB-6020 是致远电子新推出的一款 Modbus to CAN FD 协议转换器，拥有 1 路 CAN FD 接口，1 路 100M 以太网接口和 1 路 RS485 接口。该产品可支持 Modbus RTU 及 Modbus TCP，既可以当 Modbus 主站，也可以当 Modbus 从站。PXB-6020 最大支持 128 组 CAN/CAN FD 报文发送和 128 组白名单接收，支持 Modbus 所有类型的寄存器。同时还可以自定义方式发送数据，具有毫秒级转换和简单灵活等特点，广泛应用于 PLC 外扩 CAN/CAN FD 等场合。



接下来我们将使用 PXB-6020 协议转换器快速实现西门子 S7-200 SMART（以下简称：PLC）的 Modbus 数据转换为 CAN FD 数据，达到 PLC 的 RS485 接口也可以与 CAN FD 设备通信的效果。PLC 运行 Modbus RTU 主站，而 PXB-6020 协议转换器运行 Modbus RTU 从站。

为方便文章说明，PXB-6020 的 CAN FD 接口接入的是一个 USBCANFD-200U（USBCANFD 接口卡），用户可根据实际需求接入其他 CAN/CAN FD 设备。

硬件接线

1. PXB-6020 的 CAN FD 接口的 CAN_H 和 CAN_L 分别连接到 USBCANFD-200U 的 CAN FD 接口的 CAN_H 和 CAN_L；
 2. PXB-6020 的 RS485 接口的 A 和 B 分别连接到 PLC 的 RS485 接口的 A 和 B；
 3. PXB-6020 和 PLC 的以太网接口使用以太网线连接电脑；
 4. 接上 PLC、PXB-6020 和 USBCANFD-200U 的电源。
- 设备接线示意如图 1 所示：



图1 设备接线示意图

PXB-6020参数配置

接下来打开 AWPX Tools 软件对 PXB-6020 进行配置。

1. 配置 Modbus 参数，将 PXB-6020 的工作模式配置为 Modbus RTU 从站，如图 2 所示。



图2 Modbus参数设置

2. 配置 CAN FD 参数，如图 3 所示。

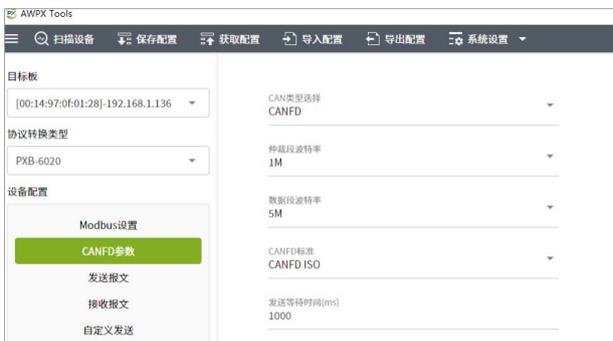


图3 CAN FD参数配置

3. 配置发送报文参数，如图 4 所示。



图4 发送报文配置

4. 配置接收报文参数，如图 5 所示。



图5 接收报文配置

PLC配置

打开 STEP 7-Micro/WIN SMART 软件, 使用该软件对 PLC 进行软件编程、配置和程序下载。

1. 双击页面左侧的“通信”按钮, 在弹出的通信界面点击“查找 CPU”, 查找到对应的 CPU 后点击“确定”, 如图 6 所示。



图6 查找CPU

2. 使用页面左侧的“指令”->“库”->“Modbus RTU Master (V2.0)”下的“MBUS_CTRL”和“MBUS_MSG”指令使 PLC 运行行为 Modbus RTU 主站, 实现对 Modbus RTU 从站的读写。

3. MBUS_CTRL 指令用于初始化主站, 必须在每次扫描时调用 MBUS_CTRL 指令, 否则 Modbus 主站协议不能正常工作。MBUS_MSG 指令用于向 Modbus 从站发起读或写请求。

4. Modbus RTU Master (V2.0) 指令库需要 286 字节的全局 V 存储器, 所以需要指定该库可使用的 V 存储器量的地址。点击并右键界面左侧的“程序块”->“库存储器...”, 会弹出如图 7 所示界面。在弹出的界面选择“建议地址”, 再点击“确定”即可。建议地址不能与参数“DatePtr”指向的地址有重合。

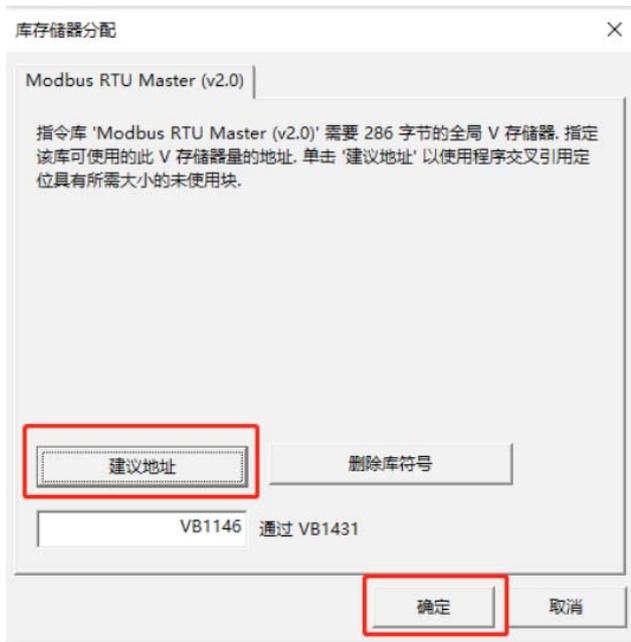


图7 库存储器分配

5. 库存储器设置完成后, 使用 MBUS_CTRL 指令与 MBUS_MSG 组态 PLC, 使其作为 Modbus RTU 主站并对 PXB-6020 进行读写。一个简单的读写 PXB-6020 的梯形图如图 8-10 所示, 此梯形图程序仅供参考。

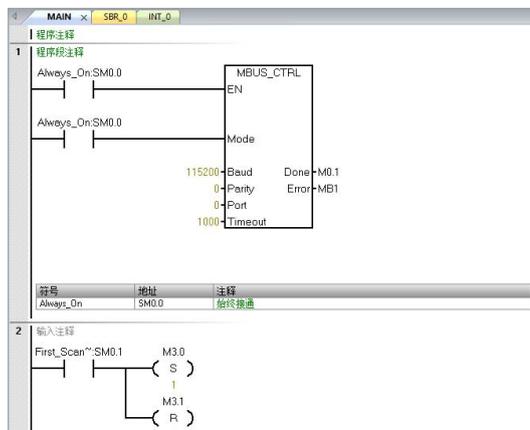


图8 梯形图1

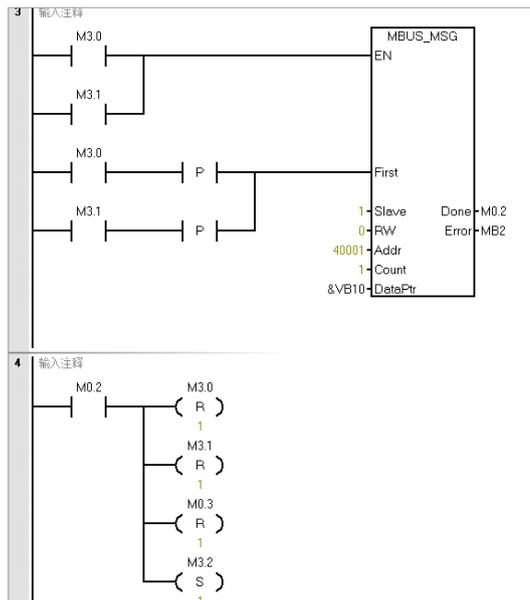


图9 梯形图2

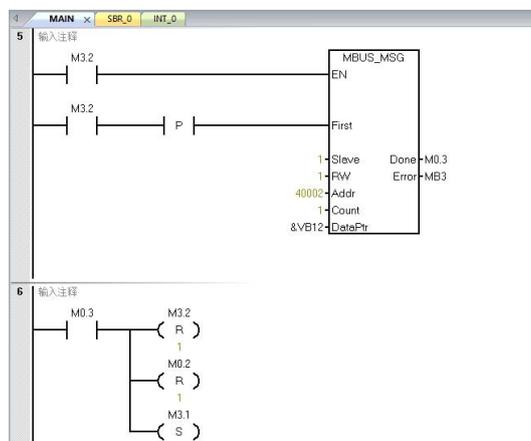


图10 梯形图3

互联互通 ▼

6. 以上梯形图实现对 Modbus 从站保存寄存器的循环读写，读取 Modbus 从站保持寄存器 0 地址一个字的数据，并将一个字数据写入 Modbus 从站保持寄存器 1 地址。

数据转换

将以上梯形图下载至 PLC 后，点击“启动程序监视”按钮后，即可在如图 11 所示的“状态图表”页面对 Modbus 从站进行读写操作。

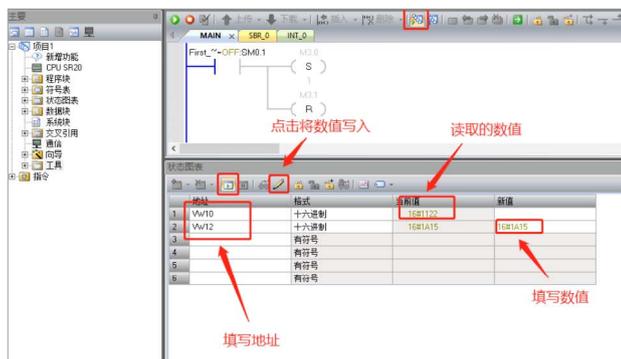


图11 读写Modbus从站数据

1. Modbus数据转换为CAN FD数据

根据图 4-5 的发送报文和接收报文的配置，在“状态图表”里面的 Vw12 地址写入 0x1122，可使 PXB-6020 发送如图 12 所示的 CAN FD 标准帧，可在 ZCANPRO 软件观测 PXB-6020 发送的 CAN FD 标准帧。



图12 发送的CAN FD帧

2. CAN FD数据转换为Modbus数据

使用 USBCANFD-200U 设备向 PXB-6020 发送帧 ID 为 0x02，帧数据段为 0x8899 的 CAN FD 标准帧，则在“状态图表”里面的 Vw10 地址可读取到数据 0x8899，如图 13 所示。



图13 读取Modbus从站数据

至此，我们成功实现了 PLC 的 Modbus 数据与 CAN FD 数据的双向转换。



【新品发布】 宽电压输入稳压电源模块E48_UHFCS-3W系列

ZLG 致远电子 2023-11-27 11:37:35

在高集成度的控制系统上，电源模块体积越做越小，为满足需求，致远电子推出一款小体积、宽压输入电源模块。E48_UHFCS-3W 系列 18~72VDC 宽输入电压范围，小体积产品。

E48_UHFCS-3W 系列产品拥有 18~72VDC 宽输入电压范围，单路输出电压 5V、12V、24V，可满足 -40°C ~+85°C 的工作温度范围，隔离耐压 4000VDC，EMC 性能优越，具有输入欠压保护，输出短路保护。



[点击购买](#)

产品特点

- 4:1 超宽输入电压范围：18~72VDC；
- 尺寸为 22.00x9.50x12.00mm；
- 隔离耐压高达 4000VDC；
- 输入欠压保护、输出短路保护等；
- 工作温度范围：-40°C ~+85°C；
- 国际标准引脚方式。

EMI性能

E48_UHFCS-3W 系列产品有优越的 EMC 性能指标，以产品 E4805UHFCS-3W 为例，增加外围电路（详情见规格书），测试的传导测试传导性能如图 1 所示，测试辐射结果如图 2、3 所示。

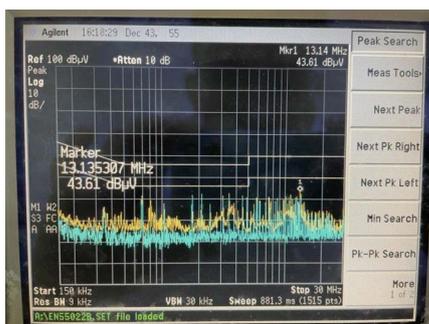


图1 E4805UHFCS-3W加外围传导测试数据 (QP&AVG值)

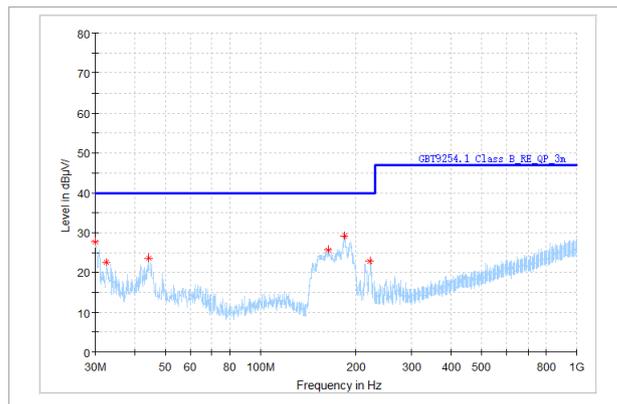


图2 产品加外围电路水平方向测试数据

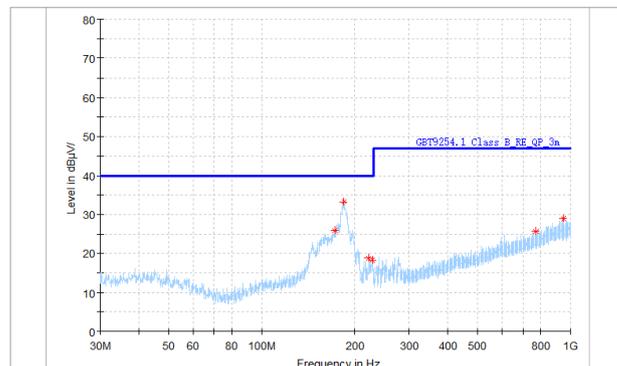


图3 产品加外围电路垂直方向测试数据

由以上可知，E4805UHFCS-3W 产品具有传导和辐射性能非常好，余量也充足，特别是辐射性能。

宽压 3W 系列产品有优秀的 EMC 特性，只需要简单的外围电路，就不用担心产品影响产品的 EMC 性能，是一款高性价比的小功率电源模块。欢迎大家测试与选用。

公司网址：www.zlg.cn

产品应用

可广泛用于工控、电力、仪器仪表、通信等领域。



【技术分享】 振动场合的电源模块该如何选型？

ZLG 致远电子 2023-11-03 11:39:35

在不同的应用环境，该如何进行电源模块的应用选型？选择合适电源模块可以延长模块的使用寿命，本文主要是介绍在振动场合电源模块可能出现的一些失效形态和如何选用可靠的电源模块。

常见存在振动的应用环境

在现实应用中，很多应用环境是存在振动情况的，需要考虑电源模块的选型，存在较大振动的常见场合如下：

- 轨道交通：地铁，高铁，城际列车；
- 工程机械：机械手、爆破机械，激光机器等。

振动环境中，常见电源模块失效的形态

振动环境中，有固定频率，固定方向的摆幅振动，器件在这样的环境中应用可能存在疲劳损伤或是掉件，导致功能丧失。如图 1 所示，单列直插的电源模块在振动环境比较恶劣的情况下应用，金属引脚受到振动的摆幅会大，在反复的摆动下金属引脚与焊点接触点产生了金属疲劳，引脚出现了折断，产品功能失效。

还有一类电源模块外壳是单靠紧配合的卡扣组装起来的，由于紧配合也会存在公差，上下外壳有轻微松动情况。这类电源模块在长时间的振动环境中应用，紧配合的上下外壳在长时间的振动磨损，卡扣会被磨掉，卡口会被磨大，最终会导致外壳脱落。

开板式的电源，PCBA 上的器件在振动应用场合中有脱落的风险。



图1 振动场合导致SIP产品的金属引脚疲劳折断图

电源模块振动实验的标准

目前电源模块的振动实验标准是参照 IEC 60068-2-6:2007 对应的国家标准是 GB/T 2423.10-2019 《环境试验 第 2 部分：试验方法 试验 Fc：振动（正弦）》，振动实验中涉及到几个关键的实验参数：振动频率范围，振幅，加速度，扫描速率，振动轴向。具体参数要求查阅振动实验标准，不再赘述。

有些行业有专门的振动实验标准，比如轨道交通行业标准 IEC/EN61373 车体 1B 级，振动实验要按行业标准。

选型与应用处理

从电源模块在振动场合应用过程中可能出现的失效形态，考虑电源模块选型从封装和外壳装配方式等着手。选择 DIP 封装和 SMD 封装的会比 SIP 封装的抗振动好，DIP 的双列针脚固定，避免了引脚疲劳折断的情况，SMD 封装的焊点与 PCB 结合牢固，不会脱落。

当然并不是在振动应用场合不能选 SIP 封装电源模块，如果振动条件不苛刻的话，SIP 封装的模块一样能够选用，应用的时候需要做一些处理，比如给电源模块点胶固定，如图 2 所示，这样可以比未做处理的电源模块使用寿命更长。

如选开板式的电源，在振动场合中应用，需要选已经对容易脱落的器件进行点胶固定的开板式电源，才能够延长使用寿命。

致远电子新推出的 E_UHBKD-3W 系列产品，E_UHBKD-3W 是 DIP 封装，由外壳加 PCBA 和灌密封胶组装而成，灌密封胶起到了固定内部器件和外壳的作用，封装尺寸 14.00*14.00*10.00mm，具备较强的抗振动能力，适用于各种具有振动的场景。



图2 SIP封装电源模块点胶固定图



3W宽压输入隔离DC-DC电源模块

[点击购买](#)

【产品应用】SM系列全隔离RS-485收发芯片“全家福”来啦!

ZLG 致远电子 2023-11-13 11:36:51

ZLG 致远电子基于近二十年的总线隔离技术及工艺经验，推出 SM 系列集成电源隔离、RS-485 收发电路和信号隔离电路“三合一”的高集成度全隔离 RS-485 收发芯片。下面就带大家看下它们的“全家福”吧!

在选择隔离与非隔离电源之前，我们先了解下两者之间的优缺点，然后根据需求，参考 P 系列隔离电源模块与 ZYM 系列非隔离电源芯片产品性能特点，选择一款合适的产品。

“大合照”来了

如图 1，ZLG 致远电子目前推出的全隔离 RS-485 收发芯片有 SM4500、SM4510、SM4500H、SM4510H。



图1 “全家福”合照

该系列 RS-485 收发芯片相较于传统模块方案，在超小、超薄的 DFN 封装内部集成了完整的 RS-485 总线隔离电路。

相似的“外貌”与共同的“内涵”

1. 它们有着共同的外表，不细看，相信你一定认不出来

四款芯片体积大小完全相同，尺寸为 12.45×9.85×3.00mm，全部采用 DFN 贴片封装，集成度高、物理尺寸小，极大地节省了占板空间。

2. 内涵1

全部支持 10Mbps 的最高波特率，工作温度覆盖 -40°C ~125°C。相较于传统的 RSM 系列模块，通讯速率显著提高、温度范围更广，满足各类复杂恶劣的工业现场 RS-485 总线隔离需求。

3. 内涵2

完全兼容 3.3V 和 5V 电路系统。四款芯片均支持 3.15V~5.25V 的宽电压供电，同时带独立的逻辑电源引脚 VIO。逻辑电源电压可跟随供电电压 Vcc，也可不跟随，以 SM4500(H) 为例，如下图 2。

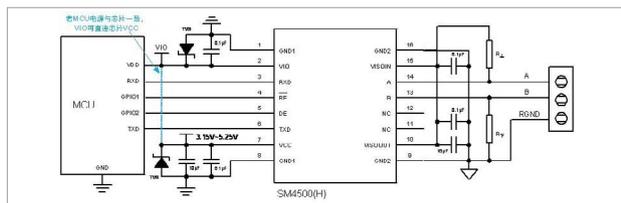


图2 典型应用

4. 共同点汇总

表1 全隔离RS-485收发芯片共同点

产品尺寸	12.45×9.85×3.00 mm
工作电压	3.15 V~5.25 V
工作温度	-40°C ~+125°C
通讯速率	10M bps
节点数	256 pcs
系统兼容性	带独立的逻辑电源引脚：VIO

独特的“性格”

为满足不同的应用环境、不同的市场需求，全隔离 RS-485 收发芯片有着各自的特点，如下表 2。

表2 全隔离RS-485收发芯片不同点

产品型号	静态电流	最大工作电流	封装	隔离耐压	ESD 等级
SM4500	10 mA	90 mA	DFN16	3500 VDC	Contact ±6kV
SM4510	10 mA	90 mA	DFN20	3500 VDC	Contact ±6kV
SM4500H	20 mA	100 mA	DFN16	5000 VDC	Contact ±8kV
SM4510H	20 mA	100 mA	DFN20	5000 VDC	Contact ±8kV

选型与应用

这四款芯片的相似程度还是非常高的，那么如何去选用呢？对于大多数的场合来说可以直接选用 SM4500 或者 SM4510，这两款只有封装引脚的不同，其他参数均一致。当应用环境要求隔离较高、抗干扰能力较强时，则可以选用 SM4500H 或者 SM4510H。

SM 系列全隔离 RS-485 收发芯片适用于储能、光伏电力、工业控制、智慧交通等领域。例如在交通信息检测设备中应用，如图 3，ZLG 面向交通信息检测应用提供主控、无线通讯及通信接口防护产品方案，主控方面提供 Cortex®-A7 平台 M6Y2C 系列核心板，产品拥有丰富的接口资源和处理能力，可实现接入各类传感器及远程维护，降低运维成本；无线方面提供 ZigBee 或 470 等多种无线通讯方案，实现与地面设备的组网及状态采集；接口防护方面提供 SM4500、SM4500H 等的通讯接口防护产品，可有效提升用户整体设备的总线通讯稳定性。

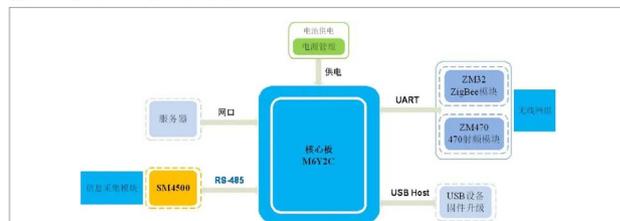


图3 交通信息检测设备方案

ZLG
SM4500
ZSCTS8
2107

全隔离RS-485收发芯片SM4500

[点击购买](#)

【技术分享】

三线制PT100测温容易忽略的设计细节

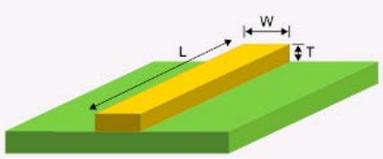
ZLG 致远电子 2023-11-01 11:42:39

高精度三线制 PT100 测温电路容易遇到 PCB 走线电阻差异导致的测量精度下降问题，看似简单的走线，可能引入 0.1°C及以上偏差。本文进行量化分析，并探讨相应的解决方法。

PCB走线电阻的影响程度

PT100 三线制测量方法，通过引线电阻相减的方式，抵消引线电阻。当三根引线的阻值不相等时，则引入测量误差。PCB 的输入端子至 ADC 走线，是 PT100 引线的一部分，同样会引入测量误差。误差值可以通过走线阻值的差值和 PT100 的平均温度系数换算。

PT100 的平均温度系数为 0.385Ω/°C，38.5mΩ 对应 0.1°C 的温度变化。PCB 走线考虑常用的 10mil 线宽、1oz 铜层厚度，如果走线长度差异为 25mm，其直流电阻为 42.6mΩ，换算成温度偏差为 0.11°C，是一个明显的误差。走线长度差异为 5mm，直流电阻变为 8.5mΩ，换算成温度偏差为 0.02°C，则基本可以忽略。

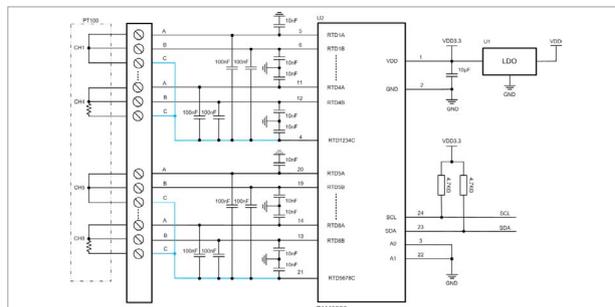


Inputs	
Trace width (W)	10 mil
Length (L)	25 mm
Trace thickness (T)	1 oz/ft ²
Ambient temperature	27 °F
Output	
Resistance	0.04263 Ω

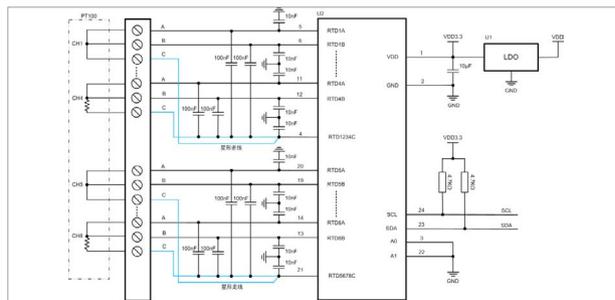
星形走线和等长走线

PT100 的三线制测量电路，在多通道应用时，激励电流的返回路径为多个通道共用。该返回路径在 PCB 上共用走线时，走线长度将更长。以高精度测温模块 ZAM6228 为例，下图中，如果通道 CH1 离 ADC 近，它的 A、B 线 PCB 走线短，C 线由于 4 个通道共用，需要在 PCB 上需要额外经过另外 3 个通道 CH2~CH4 的接线端子，走线更长。使用常规的

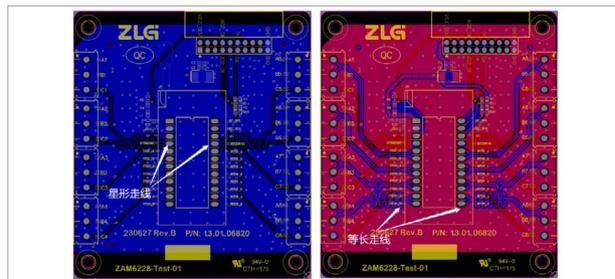
5.08mm 间距工业端子时，每个通道有 3 个位端子，C 线 PCB 走线长度多出 3*3*5.08=45.7mm，如果用 10mil 走线宽度，对应的温度偏差约为 0.2°C。



在 PCB 上使用星形走线，并通过 PCB 软件做等长走线，可以轻松解决走线电阻引入温度偏差的问题。以高精度测温模块 ZAM6228 为例，下图中，星形走线是指每个通道的 C 线单独从模块引脚走线至对应的输入端子。而每个通道的 A、B、C 三根线，通过 PCB 软件做等长走线，可以使得走线长度差异小于 5mm。



ZAM6228 模块的评估板使用了星形走线和等长走线，如下图所示。经测试，PCB 走线引入偏差不到 0.01°C，对于模块自身 0.02%+0.1°C 的测温精度，可以忽略，可以完整的得到高精度。



ZAM6228 产品资料: www.zlg.cn/tm/tm/product/id/328.html



8通道PT100热电阻温度测温模块

[点击图片](#)

【产品应用】 多通道测温还需要搭电路？一个模块全搞定！

ZLG 致远电子 2023-11-15 11:58:31

随着中国工业化进程的稳步推进，各行各业对温度测量越来越重视，传统单一通道的电路已经不能满足现有需求，全面立体多点测量温度的需求正悄然出现在市场中，多通道热电阻采集模块在市场需求下顺势推出！

传统热电阻测温电路由 ADC 与外围电路组成，可灵活搭配出所需电路。当多通道测温时，存在占用空间大、设计成本高、元器件供货不稳定、编程复杂等问题。而 ZLG 致远电子最新推出的 ZAM6228 测温模块，具有测温通道多、精度好、电路设计简单、编程方便等特点，可有效降低用户的设计成本，在锂电化成分容、电池测试设备、多通道环境监测等行业都有广泛应用。

传统热电阻测温电路如何设计

传统热电阻测温电路设计分为电桥式和恒流源式，以下分别介绍这两种测温方式：

电桥式电路主要的设计思路：选择三个阻值匹配的电阻与热电阻搭成电桥电路，当测温电阻变化时电桥信号端会出现压差，使用 ADC 采集电桥两端信号的压差后通过计算得出所需结果。此方法设计简单，使用方便，但若想在复杂环境下测得精准的温度，电桥的固定电阻要选择低温漂电阻。参考电路如图 1 所示。

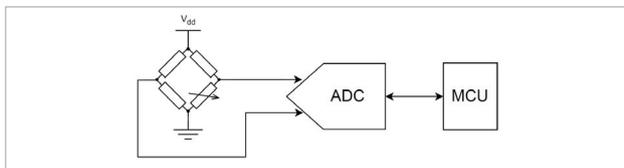


图1 电桥式测温电路

恒流源式电路一般由运放、三极管和电阻组成，可输出恒定电流。将热电阻串联进恒流源电路中，此时热电阻两端会产生压差，用 ADC 采集热电阻两端电压后通过 MCU 计算得到对应温度。为防止测量电路对热电阻两端的电压信号产生影响，推荐选择具有差分输入的高阻抗 ADC。参考电路如图 2 所示。

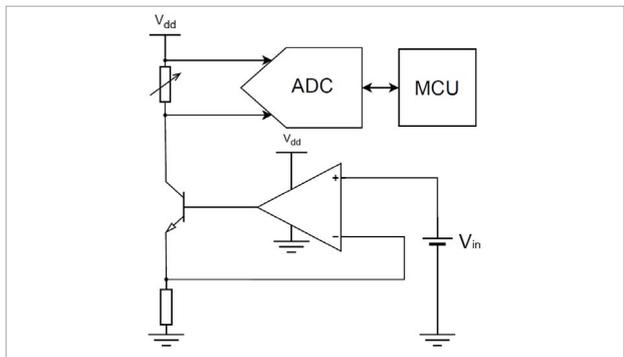


图2 恒流源式测温电路

如何让传统热电阻测温电路实现多通道设计

1. 堆叠电路法，按照实际通道需求去堆叠电路：

- 优点：电路结构一致，原理图设计与 PCB 绘画方便。
- 缺点：当通道多时会严重占用电路板空间，MCU 引脚占用较多，影响其他功能。

2. 通道切换法，使用模拟开关或继电器进行通道切换：

- 优点：节省电路板空间，使用精简的测温电路实现多通道测量。
- 缺点：使用元器件较多，电路相对复杂，软件工作量大，调试周期长。

8通道测温模块，多通道设计不二之选

ZAM6228 是致远电子面向多通道工业测温应用开发的一款 8 通道三线制 PT100 测温模块，无需外围调理电路，PT100 可直接接入引脚，即可通过 I2C 输出以 °C 为单位的测试结果，模块每通道采集速率 2SPS，测量精度 0.02%±0.1°C，分辨率可达 0.01°C，温漂 10PPM/°C，该模块易于集成，抗干扰性强，具备断线检测功能，支持多模块级联，可广泛应用于锂电化成分容、电池测试柜、多通道环境监测等。

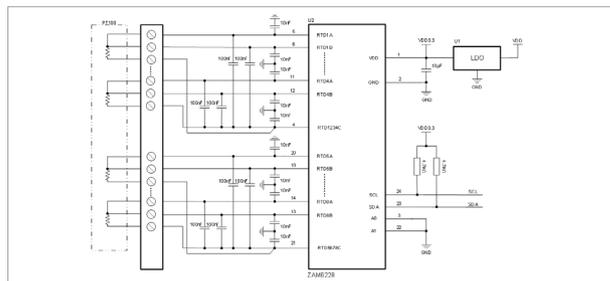
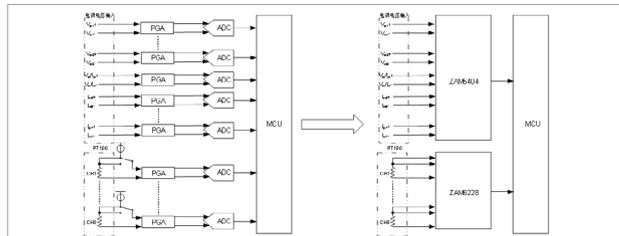


图3 ZAM6228 外围电路

电路设计中的模块应用

现如今测温应用常常伴随着测量电压、测量电流、IO 拓展等其他需求，应用设计中测量电路由于电路设计复杂、占用位置多、软件调试困难等问题而让人头疼不已，时常我们在想，是否有几种现成的模块直接使用以节省出宝贵的时间调试更重要的电路中呢？故致远电子根据实际需求出发，研发的 ZAM6228 测温模块、ZAM5404 高速模拟采集模块等多种高精度模块解决此类问题，模块体积小、方便调试、外围电路设计简单，使得应用设计方面更加轻松愉快。



8通道PT100热电阻温度测温模块

[点击购买](#)

2023/11 第11期

微文摘

ZLG MICRO DIGEST



ZLG致远电子官方微信