ZLG 致远电子

微文摘

ZLG MICRO DIGEST

2023/9 第9期

月刊

D9系列核心板

国产工业芯，性能再升级

产品型号	MD9340-2GF8GLI-T	MD9340-2GF16GLI-T	MD9350-2GF8GLI-T	MD9350-2GF16GLI-T
处理器	D9340	D9340	D9350	D9350
ARM内核	四核Cortex®-A55 +双核 Cortex®-R5	四核Cortex®-A55 + 双核 Cortex®-R5	五核Cortex®-A55 + 三核 Cortex®-R5	五核Cortex®-A55 + 三核 Cortex®-R5
主频	1.6GHz + 800MHz	1.6GHz + 800MHz	1.6GHz + 800MHz	1.6GHz + 800MHz
操作系统	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS
GPU Core	支持	支持	支持	支持
VPU Core	支持	支持	支持	支持
NPU	-	-	0.8TOPS	0.8TOPS
运行内存（SDRAM）	2GB	2GB	2GB	2GB
板载存储（eMMC）	8GB	16GB	8GB	16GB
QSPI Flash	16MB	16MB	16MB	16MB
LVDS接口	2路，4-Lane	2路，4-Lane	2路，4-Lane	2路，4-Lane
显示分辨率	单路最大支持 1920x1080@60Hz，双路最大支持 2560x1440@60Hz			
电阻触摸屏	IIC、SPI或USB扩展支持	IIC、SPI或USB扩展支持	IIC、SPI或USB扩展支持	IIC、SPI或USB扩展支持
电容触摸屏	IIC扩展支持	IIC扩展支持	IIC扩展支持	IIC扩展支持
MIPI-CSI	1路，4-Lane	1路，4-Lane	1路，4-Lane	1路，4-Lane
以太网	2路千兆	2路千兆	2路千兆	2路千兆

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

ZWS 云平台

边缘计算

核心板

工控板

工控机

互联互通

CAN-bus 总线

无线通讯

接口与协议转换

感知控制

电源与隔离

【EsDA应用】常用IO设备节点详解

ZLG 致远电子　2023-09-13 11:35:25

本文介绍如何通过 EsDA 开发套件，快速熟悉常用 IO 设备节点及其使用方法，大大缩短了用户对 EsDA 的学习开发周期。本文选用 EPC6450-AWI 开发板，以及图形化设计工具 AWFlow Designer 进行测试讲解，对常用 IO 设备节点和使用方法进行详细说明。

前期了解

对 AWFlow 工具以及 AWFlow 理论并不了解的，可以点击 AWFlow 文档进行学习：

https://awtk.zlg.cn/pro/docs/AWFlow/

对 AWFlow 有一定的了解但是对 EsDA 和相关节点的综合应用并不了解的，可以点击 Z 站的 EsDA 专栏进行学习：

https://z.zlg.cn/specialInfo?id=134

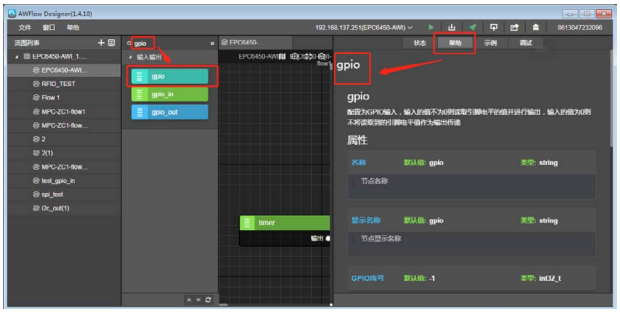
常用节点简介

本文将对 gpio 引脚电平操作的 gpio 节点，gpio_in 节点，gpio_out 节点三个节点，对 led 灯操作的 led 节点，对按键操作的 button 节点，对 adc 通道采集的 adc 节点进行详细描述。

1. gpio节点

1.1 查看帮助文档

在搜索框中输入 gpio 关键字即可检索出 gpio 节点，点击 gpio 节点即可看到右边弹出 gpio 节点的帮助文档界面，里面包含节点的简介以及属性，输入和输出的相关参数介绍。用户可以在此操作下找寻相关节点的帮助文档去了解节点的相关参数。



1.2 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- GPIO 序号：对需要操作的 GPIO 引脚序号进行选定，此引脚序号与目标板的丝印一一对应；
- 模式：设置 GPIO 序号引脚的输入模式，float 是浮空输入，pull up 是输入上拉，pull down 是输入下拉。

1.3 输入

- msg.payload ：不为 0 则表示读取属性值“GPIO 序号”的电平值，为 0 则不将读取到的电平值进行输出传递给下一个节点。

所有节点的输入参数都是由上级节点的输出提供的，且通常为 msg 对象携带的对应属性参数。

1.4 输出

- msg.payload：输出的数据，需要注意的是只有当输入的 msg.payload 不为 0 时才有输出，0 表示读取属性值“GPIO 序号”的电平值为低电平，1 表示读取到的是高电平。

1.5 加载 demo 示例

- 在相关节点的帮助文档旁边有一个示例，点击后则可以看到对应的节点 demo 示例，选中查看的 demo 示例，并加载到项目中即可查看。

1.6 自定义 demo

1.6.1 绘制流图

- 添加 timer，fscript，gpio，fscript 节点到画布中并连线如下图。

1.6.2 节点配置

- timer 节点周期性的触发 fscript 节点，需要更改周期性的时长可双击 timer 节点，修改属性值“定时周期”即可。而 fscript 节点作为 gpio 的上级节点，那么传递的则是 gpio 的输入值，在此希望能周期性的读取电平值，所以 msg.payload 的值不为 0 即可，gpio 的上级节点 fscript 的内容：

msg.payload = 666

双击 gpio 节点，选择目标板对应丝印的 GPIO 序号，在此选择的模式是浮空输入如下图。

gpio 节点的消费者 fscript 节点作为输出，直接打印 gpio 节点的输出值显示电平值即可，gpio 节点的消费者 fscript 节点内容如下：

```
print("read gpio level value is "+msg.payload)
```

关于 timer，fscript 节点的介绍可参考 AWFlow 节点使用手册：https://awtk.zlg.cn/pro/docs/AWFlow/

1.6.3 下载验证

将此流图下载到目标板后，连接调试串口后，打开串口助手，改变 UTX5 的电平值可得到如下显示：

```
[11:04:55.686]收←◆read gpio level value is 0
[11:04:56.692]收←◆read gpio level value is 1
[11:04:57.402]收←◆ : question PTR _awflow._tcp.local.
  --> answer awflow.dev._awflow._tcp.local. port 8888 (multicast)
[11:04:57.698]收←◆read gpio level value is 1
[11:04:58.711]收←◆read gpio level value is 1
```

2. gpio_in节点

将 GPIO 配置为输入模式，并轮询读取电平状态且将其作为输出值。与 gpio 节点功能类似，都是配置为输入模式且可以将读取到的电平值作为输出，只不过 gpio_in 是不间断循环的输出，并且 gpio 节点还加了一个是否开启读电平的输入。

2.1 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- GPIO 序号：对需要操作的 GPIO 引脚序号进行选定，此引脚序号与目标板的丝印一一对应；
- 模式：设置 GPIO 序号引脚的输入模式，float 是浮空输入，pull up 是输入上拉，pull down 是输入下拉；
- 轮询：是否开启以设定值去轮询的读取 GPIO 序号的电平状态，off 则关闭用设定值去轮询读取电平状态而使用默认的 50ms 去轮询读取，on 则以设定值去轮询读取电平状态值；
- 轮询间隔（ms）：以设定值去轮询读取电平状态，设置生效的前提是轮询需为 on。

2.2 输出

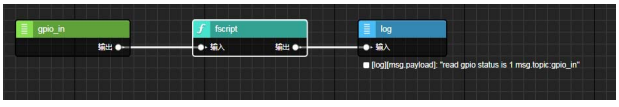
- msg.payload：属性值“GPIO 序号”的电平状态值，0 为低电平，1 为高电平；
- msg.topic：固定输出 gpio_in 字符串。

pump 类型节点无输入，相关类型节点介绍可查看 AWFlow 节点开发指南进行更多的了解。

2.3 使用方法

2.3.1 绘制流图

添加 timer，fscript，log 节点到画布中并连接如下图所示：



2.3.2 节点配置

双击 gpio_in 节点，选择目标板对应丝印的 GPIO 序号，在此选择的模式是浮空输入，开启以设定值 2000ms 去轮询读取 gpio 电平状态值如下图所示。

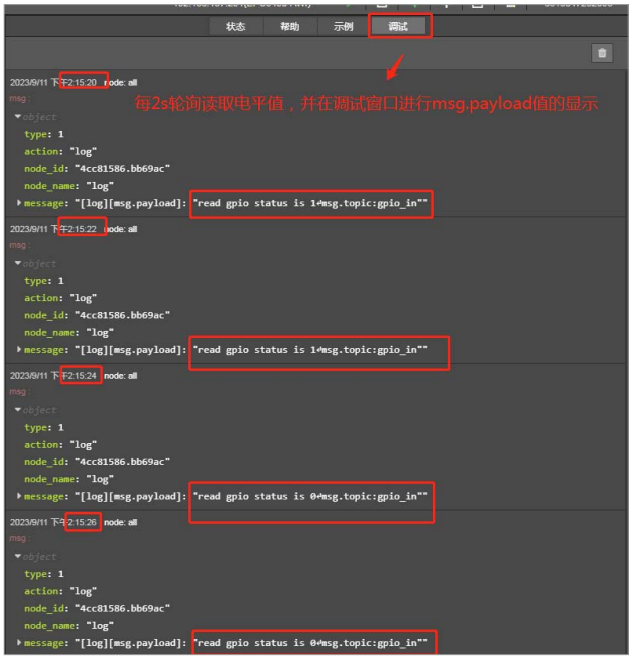
gpio_in 节点的消费者节点 fscript 是对 gpio_in 的输出参数进行输出显示，内容如下：

```
msg.payload = "read gpio status is " + msg.payload + "\n" + "msg.topic:"+msg.topic
```

用 log 节点对上一级节点的 msg.payload 输出到调试窗口进行显示。

2.3.3 下载验证

将此流图以下载运行的模式下载到目标板后，可在 AWFlow Designer 的调试窗口查看到如下信息：



3. gpio_out节点

配置为 GPIO 输出，将输入的 msg.payload 的值设置为 GPIO 的电平值。

3.1 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- GPIO 序号：对需要操作的 GPIO 引脚序号进行选定，此引脚序号与目标板的丝印一一对应；
- 模式：设置 GPIO 输出模式，push pull 为推挽输出模式，open drain 为开漏输出模式；
- GPIO 初始状态：设置 GPIO 初始输出的状态值，0 为低电平，1 为高电平。
- 3.2 输入
- msg.payload：设置属性 GPIO 序号的输出电平值，0 为低电平，1 为高电平，2 为电平翻转。
- sink 类型节点无输出。

3.2 输入

- msg.payload：设置属性 GPIO 序号的输出电平值，0 为低电平，1 为高电平，2 为电平翻转。

sink 类型节点无输出。

3.3 使用方法

3.3.1 绘制流图

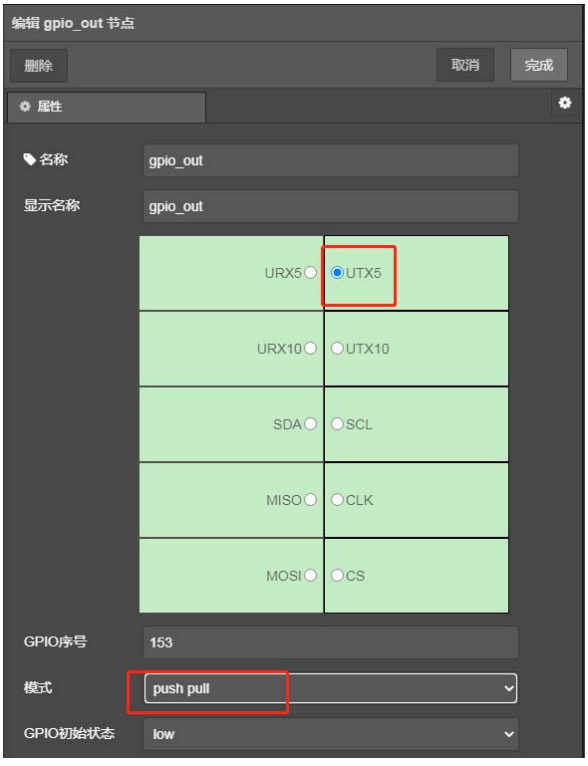
添加 timer，fscript，gpio_out 节点到画布中，并连线如下图所示。

3.3.2 节点配置

timer 节点周期性定时触发 fscript 节点，这里的 timer 的属性定时周期值配置为 3000ms，而 fscript 节点作为 gpio_out 节点的上一级节点，主要内容是设置 gpio_out 的输入参数，内容如下：

```
if(((msg.payload/1000)%2)==0) {
    msg.payload = 0
    print("low level\n")
}else {
    msg.payload = 1
    print("high level\n")
```

双击 gpio_out 节点，选择目标板对应丝印的 GPIO 序号，在此选择的模式是推挽输出。



3.3.3 下载验证

下载流图到目标板后，可以用万用表去查询 GPIO 序号对应的引脚电平值可知，电平是周期性的翻转。

4. led节点

该节点主要有两种模式，一种是设置 led 灯的状态，一种是设置快闪灯。

4.1 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- led 设备名称：Led 设备名称，可根据不同的目标板对应丝印进行设备选择；
- 初始状态：对选中的 led 的设备设置初始状态值，0 为 led 灯的初始状态为灭，1 为 led 灯的初始状态为亮。

4.2 输入

- msg.device_name：led 设备名称，将与属性值“led 设备名称”进行对比，若不同则该节点功能无效；
- msg.payload：模式一，设置 led 的状态值。0 为关闭 led 灯，1 为开启 led 灯，2 为翻转 led 灯；
- msg.timer：模式二，快闪灯。该输入参数为设置定时闪烁的时间，单位 ms，若为 0 则关闭快闪，大于 0 则设置闪烁间隔时间；
- msg.fast_blink：模式二，快闪灯。设置定时快闪的次数，在定时闪烁周期中的快闪次数，输入参数 msg.timer 与 msg.fast_blink 需同步设置才能生效。

sink 类型节点无输出。

4.3 设置灯状态示例

4.3.1 绘制流图

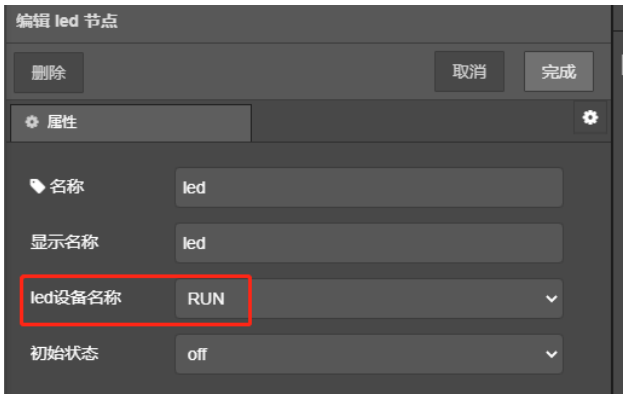
添加 timer，fscript，led 节点到画布中并连线如下图所示：

4.3.2 节点配置

timer 节点周期性的去设置 led 灯的状态值，fscript 节点作为 led 的上一级节点，其内容主要是对 led 设备状态的设置：

- msg.payload = 2

双击 led 节点，在属性“led 设备名称”的下拉框中选择对应丝印的 LED 设备如下：



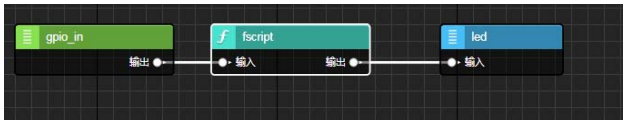
4.3.3 下载验证

下载流图后可看到 RUN 灯周期性的 3s 翻转一次。

4.4 快闪灯示例

4.4.1 绘制流图

添加 gpio_in，fscript，led 节点到画布，并连线如下图所示：



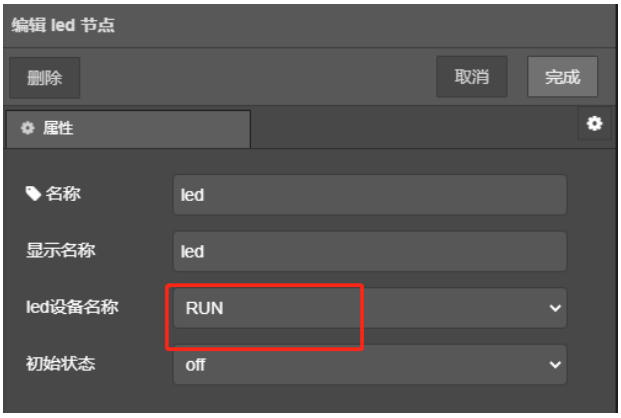
4.4.2 节点配置

双击 gpio_in 节点，本节点主要作用就是运行流图后只触发一次 fscript 节点，所以要开启定时轮询，且轮询间隔需设置尽量大一点如下图所示：

fscript 节点作为 led 的上一级节点，主要内容是 led 相关的输入参数，内容如下：

```
if(((msg.payload/1000)%2)==0) {
    msg.payload = 0
    print("low level\n")
}else {
    msg.payload = 1
    print("high level\n")
}
```

双击 led 节点，在属性“led 设备名称”的下拉框中选择对应丝印的 LED 设备如下：



4.4.3 下载验证

下载流图到目标板后，对 gpio_in 节点配置的引脚进行电平置低，5s 内切换到高电平后即可看到 led 灯 2s 闪烁 2 次的实验现象。

5. button节点

5.1 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- 对应的 GPIO 编号：对需要操作的 GPIO 引脚序号进行选定，此引脚序号与目标板的丝印一一对应；
- 触发条件：按钮按下时的触发条件，fall 表示按钮按下的触发条件为 GPIO 下降沿触发，up 表示按钮按下的触发条件为 GPIO 上升沿触发。

5.2 输出

msg.payload ： 按钮按下状态输出为 on，松开状态输出为 off，在初始化完成以及按钮状态发生改变的时候会输出当前的状态。

pump 类型节点没有输入。

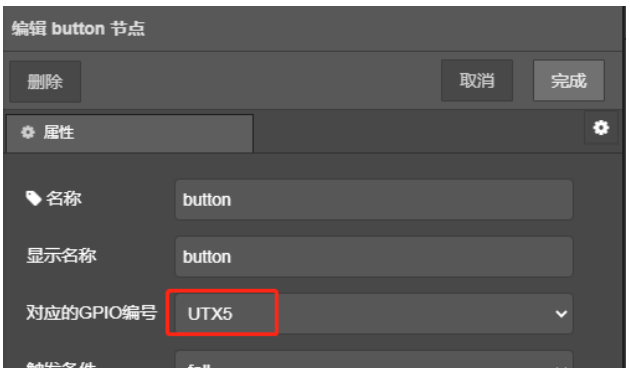
5.3 使用方法

5.3.1 绘制流图

添加 button，fscript，led 节点到画布中并连线如下图所示：

5.3.2 节点配置

双击 button 节点，对需要操作的引脚进行选择如下：



fscript 节点作为 button 的消费者节点，内容是 button 的相关输出参数:

```
if(msg.payload == "on"){
    msg.payload = 1
}else if(msg.payload == "off") {
    msg.payload = 0
}
```

双击 led 节点，选择闪灯的 led 设备。

5.3.3 下载验证

下载流图后，对 button 配置的按钮进行按下和松开的操作可以看到相应 led 灯的效果。

6. adc节点

6.1 属性

- 名称：可用于索引查找本节点；
- 显示名称：用于画布上显示的名称，仅做显示；
- 通道号：进行采样的 ADC 通道号，可根据不同的目标板对应丝印进行设备选择；
- 采集次数：ADC 通道对 ADC 值的采集次数；
- 采样率：ADC 通道号采样率（每秒 ADC 的采样次数）。

6.2 输出

- msg.payload：通道采样的最终电压值 (mv)；
- msg.vref：基准电压；
- msg.bits：ADC 通道位数；
- msg.rate：ADC 通道采样率。

pump 类型节点没有输入。

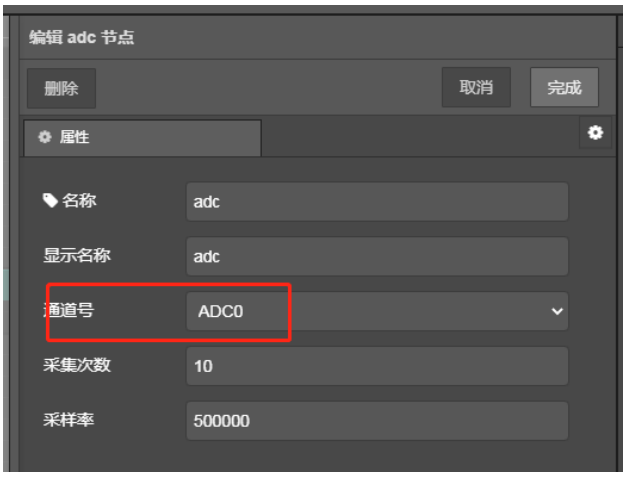
6.3 使用方法

6.3.1 绘制流图

添加 adc，fscript，log 节点到画布中并连线。

6.3.2 节点配置

双击 adc 选择对应丝印的 ADC 设备如下：



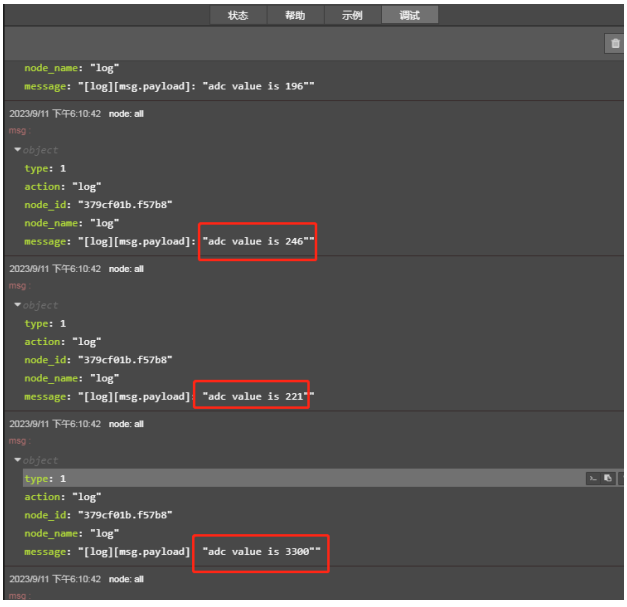
fscript 节点作为 adc 的消费者节点，其内容主要是 adc 的相关输出参数：

```
msg.payload = "adc value is "+msg.payload
```

log 节点对上一级 fscript 节点进行输出显示。

6.3.3 下载流图

下载流图后，对 adc 配置的通道号进行不同电压值的采样结果如下：



【EsDA应用】IO设备接入MQTT

ZLG 致远电子 2023-09-27 11:32:31

随着物联网的迅速发展，越来越多的 IO 设备需要与云平台进行通信，以实现远程监控和控制。本文将基于 EsDA 开发套件快速实现将 IO 设备接入 MQTT 的应用开发，帮助用户实现智能互联。

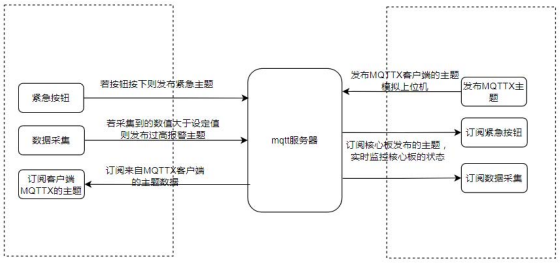
简介

在物联网应用中，接入 IO 设备是非常重要的一环。而 MQTT 作为轻量级的一种通信协议，被广泛应用于物联网领域。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 IO 设备接入 MQTT，实现实时的数据传输和远程控制。

项目概述

MQTT 是通过发布主题来上传消息，订阅相关的主题来接收消息。本文将通过对 EPC6450-AWI 核心板 上的 IO 设备执行数据采集和数据处理后，将数据以发布 MQTT 主题消息的形式进行上传；而 MQTTX 客户端以订阅相关主题来实时监控 IO 设备相关的数据，MQTTX 客户端还可以以发布主题的形式对核心板进行远端控制和消息下发。

本文的 IO 设备主要是 button 按键和 ADC 数据采集，按键主要是模拟按下紧急按钮后上传紧急戒备的状态消息到云端，MQTTX 客户端订阅紧急状态的主题就可以实时监控并显示相关状态。ADC 数据采集是将采集到的数据与设定值对比，若大于设定值则发布数值过高报警的主题，MQTTX 客户端订阅该主题就可以实时显示 ADC 数据采集的状态。本项目大概的逻辑处理可参考下图。



项目准备

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

- 【EsDA 应用】常用 IO 设备节点详解
- EsDA MPC-ZC1 入门（二）—— LED 控制
- 基于 EsDA MPC-ZC1 快速实现——按键高级应用
- EsDA MPC-ZC1 应用——串口服务器（一）

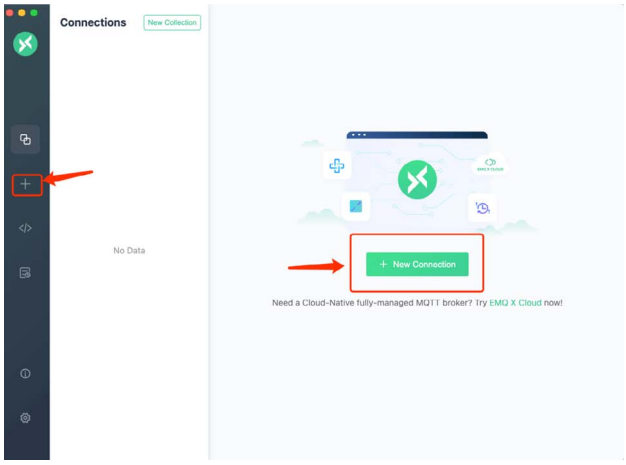
1. 上位机准备

1.1 下载 MQTTX

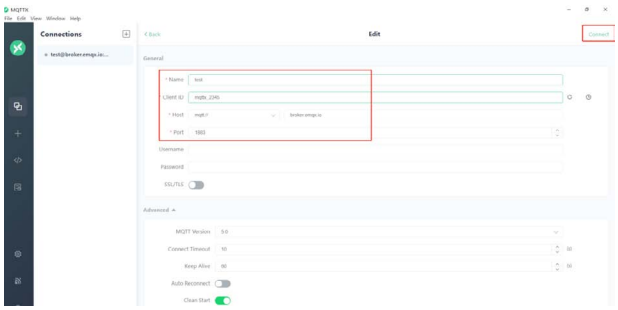
在官网下载并安装 MQTTX，一款开源跨平台 MQTT 协议的桌面客户端。在使用过程中不需要自行部署 MQTT 服务器，使用以下代理地址和端口号就能进行快速测试，MQTT 代理地址：broker.emqx.io，端口号：1883(TCP)；8883(SSL)。

1.2 快速建立连接

打开软件，点击左侧菜单栏的 “+” 按钮。如果页面为空，也可直接点击右侧的 + New Connection 来快速设定新的客户端连接。

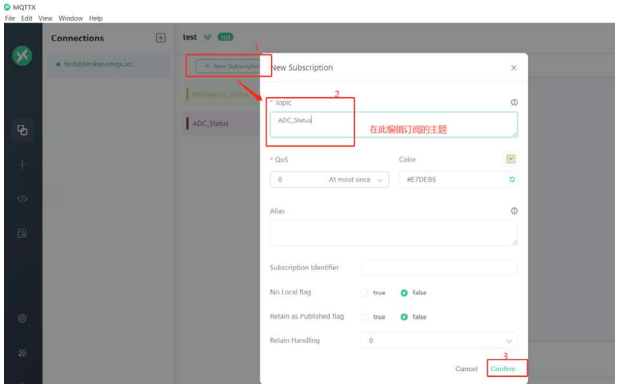


在创建连接的页面上，需要设置连接服务器的相关信息，用户只需要填写 Name(连接标识名称，客户随机设定即可)，Client ID(客户端 ID 号，客户随机设定即可)，Host(连接的服务器地址，因为不是自建服务器，所以使用公用的 broker.emqx.io)，Port(服务器对应的端口号 1883) 参数即可，设置成功后点击 connect 按钮，若网络无碍则等待连接成功即可。

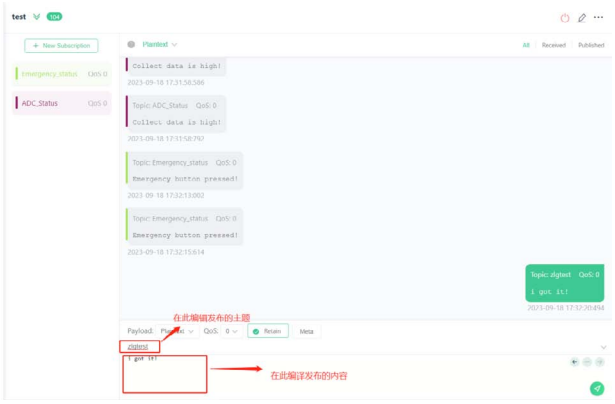


1.3 消息的订阅与发布

连接成功后，点击 New Subscription 按钮，在弹窗的 Topic 编辑框中填写客户需要订阅的主题名称，填写后点击 Confirm 确认，则订阅成功。



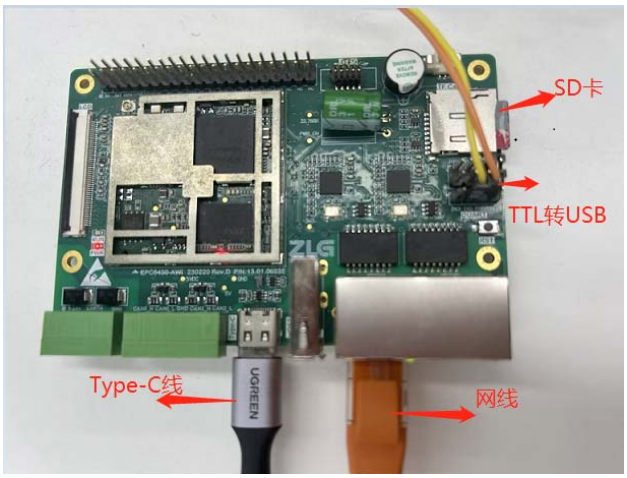
若需要发布主题，则在下图中的发布主题编辑框中填写用户的发布主题名称，内容框中填写发布主题的内容。



MQTTX 软件的更多使用操作，可以查看其官网进行阅读并学习。

2. 硬件准备

- 在标有丝印为 TF Card 的卡槽处，插入 SD 卡。
- 在标有丝印为 DUART 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 的网络接口处，插上网线头，另一端的网线头插在 PC 的网络接口处。



3. 网络搭建

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
```

因为 AWFlow 的 mqtt 节点是客户端节点，而访问的 mqtt 代理地址 broker.emqx.io 是需要流量的，所以需要通过 PC 电脑共享网络给开发板上网。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 配置为共享给以太网。

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。

项目实施

本项目业务主要分为三个部分：

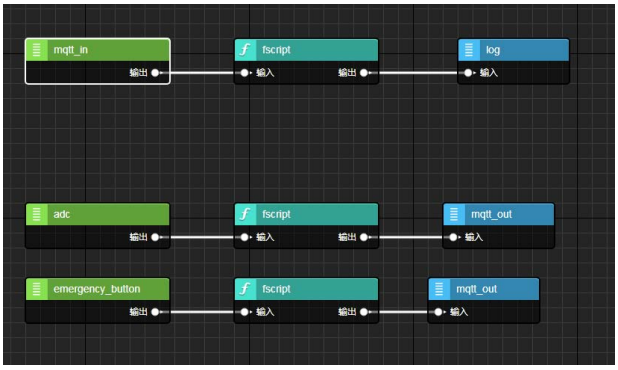
MQTTX 客户端：订阅核心板设备发布的主题，并发布对核心板控制和下发消息的主题；

按键模块：报警按钮按下则触发报警状态，并发布报警的主题；

ADC 采集模块：adc 数据采集，若采集到的数值大于设定值则触发数值过高报警，并发布 adc 数值报警的主题。

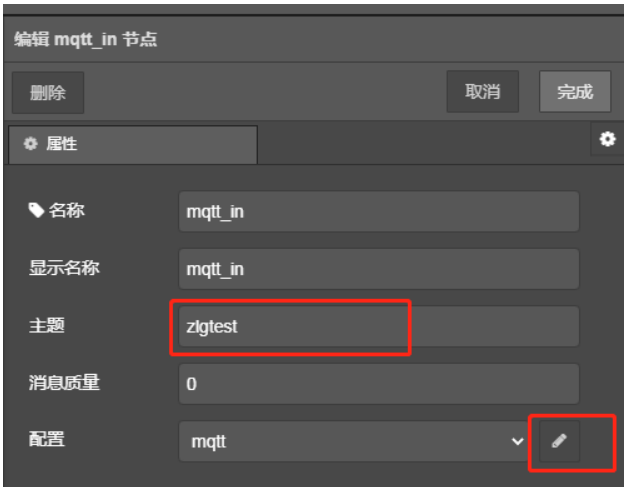
1. 流图绘制

添加 mqtt_in，fscript，log，adc，mqtt_out，button 节点到画布中并连线如下图。



2. 节点配置

双击 mqtt_in 节点，配置订阅的主题以及点击配置的铅笔图标对 mqtt_config 配置节点进行配置。

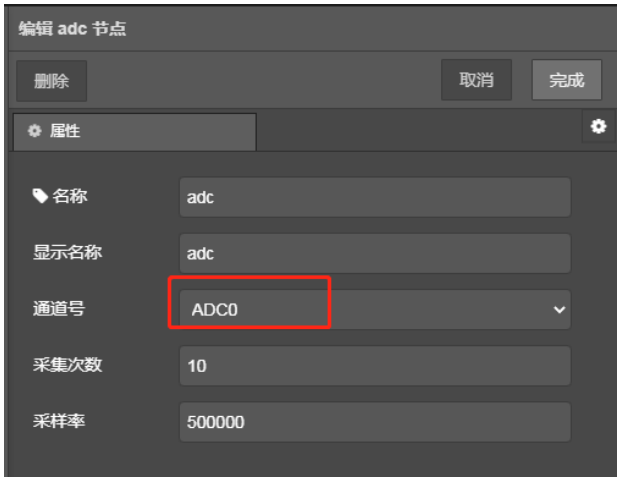


在 mqtt_config 配置节点中，可以只对主机地址，端口以及客户端 id 进行配置，如下图，因为本流图所有的 mqtt_config 节点配置值都相同，后面不再赘述。

mqtt_in 的消费者节点 fscript 的内容是对订阅主题的消息进行处理并输出，内容如下：

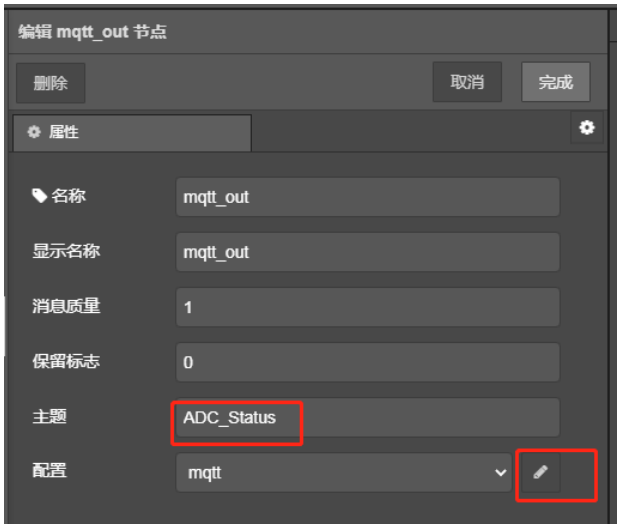
```
print("receive mqtt topic is "+msg.topic+'\n')
rb = rbuffer_create(msg.payload, msg.payloadLength)
f = "receive mqtt data is "
msg.payload = f + rbuffer_read_string(rb)
```

双击 adc 节点，选择目标板对应丝印的 ADC 通道号。

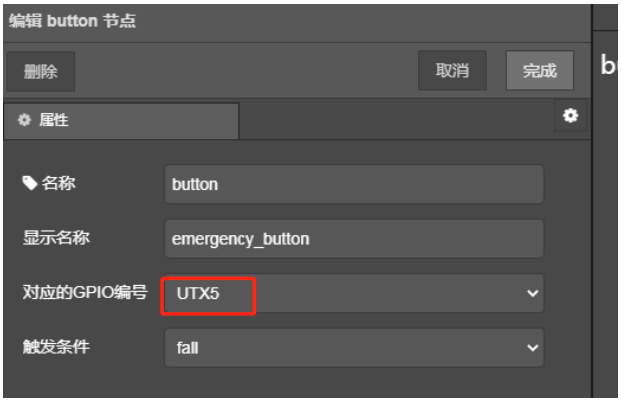



```
if (msg.payload > 2500){
  print("adc data is high")
  msg.payload = "Collect data is high!\n"
  wbuffer_write_string(wb,msg.payload)
  output.payload = wbuffer_get_data(wb)
  output.payloadLength = len("Collect data is high!\n")
}
```

接着双击 adc 排头的 mqtt_out 节点，编辑发布的 MQTT 主题，以及对 mqtt_config 进行配置。



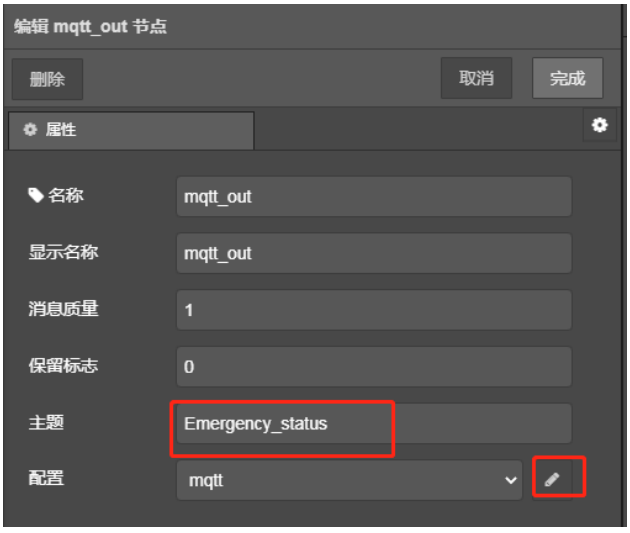
双击 button 节点，选择目标板对应按键丝印的 GPIO 编号。



双击 button 的消费者节点 fscript，本节点的主要内容是对 button 按键输出的数值进行处理后传递给 mqtt_out 节点，内容如下：

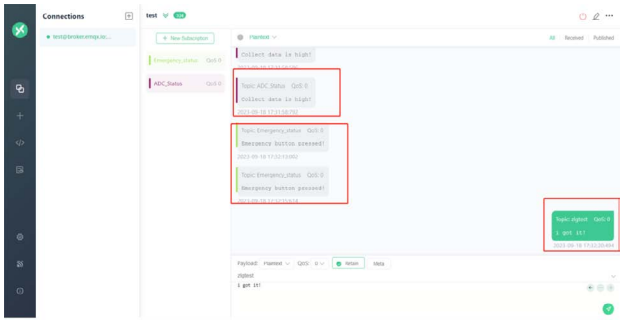
```
if (msg.payload == 'on' ){
  print("button pressed")
  msg.payload = "Emergency button pressed!\n"
  wbuffer_write_string(wb,msg.payload)
  output.payload = wbuffer_get_data(wb)
  output.payloadLength = len("Emergency button pressed!\n")
}else if(msg.payload == 'off') {
  print("button bounced")
}
```

双击 button 按键排头的 mqtt_out 节点，对发布主题和 mqtt_config 进行配置。



3. 下载验证

基于上位机准备章节搭建好 MQTTX 客户端后对 ADC_Status 和 Emergency_status 的主题进行订阅，下载 AWFlow 流图到核心板中；进行按键按下和弹起等的操作以及 adc 数据采集的操作后，可以看到 MQTTX 客户端能接收到对应主题的消息，实现了对核心板的 IO 设备进行数据和状态等的实时监控。



MQTTX 客户端发布 zlgtest 的主题，可以在核心板上的打印查看发布主题的消息内容如下，也可以根据客户端发布主题的消息对设备进行控制等操作。

```
sw_node_sink_mqtt_out_on_data:176 payload_len > 0 && payload != NULL

[17:21:34.349]收←◆button pressed

[17:21:34.447]收←◆button bounced
sw_node_sink_mqtt_out_on_data:176 payload_len > 0 && payload != NULL

[17:21:34.525]收←◆button pressed

[17:21:35.558]收←◆button bounced
sw_node_sink_mqtt_out_on_data:176 payload_len > 0 && payload != NULL

[17:21:36.775]收←◆button pressed

[17:21:36.875]收←◆button bounced
sw_node_sink_mqtt_out_on_data:176 payload_len > 0 && payload != NULL

[17:21:37.013]收←◆button pressed

[17:21:38.154]收←◆button bounced
sw_node_sink_mqtt_out_on_data:176 payload_len > 0 && payload != NULL

[17:21:41.225]收←◆receive mqtt topic is zlgtest


[17:21:41.405]收←◆[log][msg.payload]: "receive mqtt data is ,105,32,103,111,116,32,105,116,33"
[log][msg.payload]: "receive mqtt data is ,105,32,103,111,116,32,105,116,33"
tk_iostream_get_ostream:31 condition(stream != NULL && stream->get_ostream != NULL) failed!
[ERROR] [76L tk_ostream_tcp_server_write] condition((ret = (ostream) ? RET_OK : -1) == RET_OK) failed!
sw_ostream_write_payload:94 condition(ret == sizeof(header)) failed!

[17:21:44.855]收←◆receive mqtt topic is zlgtest


[17:21:45.022]收←◆[log][msg.payload]: "receive mqtt data is ,105,32,103,111,116,32,105,116,33"
[log][msg.payload]: "receive mqtt data is ,105,32,103,111,116,32,105,116,33"
tk_iostream_get_ostream:31 condition(stream != NULL && stream->get_ostream != NULL) failed!
[ERROR] [76L tk_ostream_tcp_server_write] condition((ret = (ostream) ? RET_OK : -1) == RET_OK) failed!
sw_ostream_write_payload:94 condition(ret == sizeof(header)) failed!
```

根据上述实验可知，将 IO 设备接入 MQTT 后，即可实现对设备的监控和控制等能力，为用户带来了更好的使用体验。

【产品应用】
如何通过云平台远程运维ZigBee网关及节点

ZLG 致远电子 2023-09-01 11:34:13

工业物联网应用中，节点设备可以通过 ZigBee 网关与其他设备通信，网关通过 wifi、4G 或以太网的方式与云端通信。本文将介绍通过 ZWS 云平台对 ZigBee 网关及节点进行远程运维管理。

应用场景

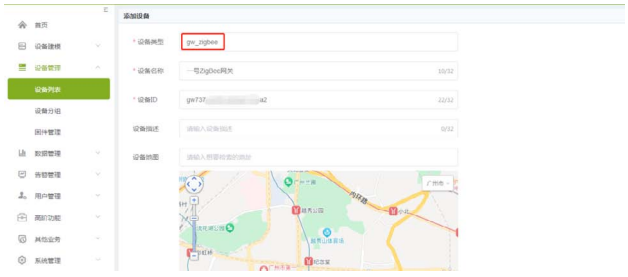
GZ32M 系列是工业级 ZigBee 物联网网关，是 ZLG 致远电子 ZigBee 生态系统的核心设备，能够实现多种有线、无线协议的传输，内置丰富的网络管理功能，且支持 MQTT 协议上云。适用于工业控制、照明、智慧楼宇、数据采集等多种应用场景

GZ32M ZigBee 网关支持 ZigBee 协议版本终端设备接入，应用于低功耗远距离物联网场景。还支持对网关自身进行远程配置、远程升级、远程监测、异常数据告警等一系列远程维护功能。本文将介绍通过 ZWS 云平台对 ZigBee 网关及节点进行远程运维管理。

ZigBee网关接入ZWS云平台

1. 打开网关的本地网页配置，上云配置中的云策略选择“sys_default”。该策略主要连接 ZWS 云，会将数据传输到 ZWS 云中。

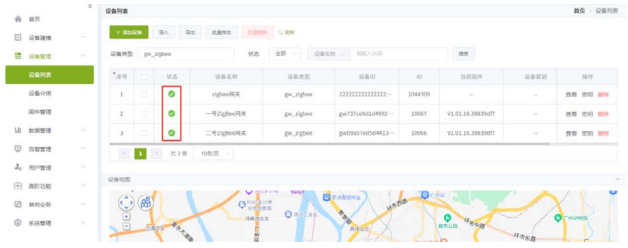
2. 登录 ZWS 物联网云平台，在云平台中添加网关。



ZigBee网关及节点云端远程运维

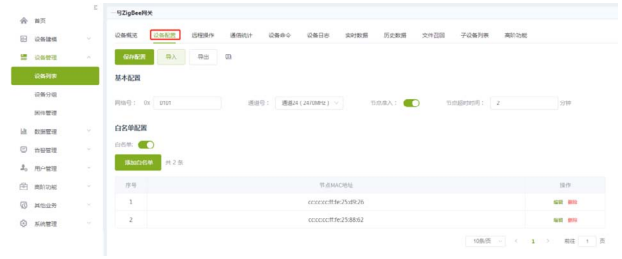
1. 远程查看ZigBee网关和节点在线情况

接入 ZWS 云后，可以远程查看网关和节点的在线情况，直观看到是否有掉线。



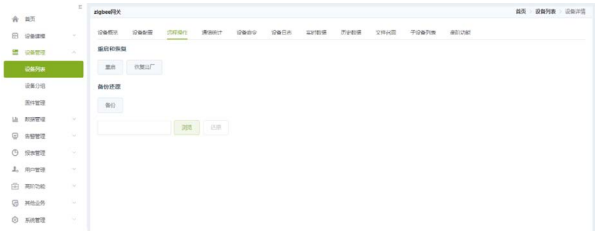
2.远程配置ZigBee网关

可以远程配置网关的 ZigBee 参数以及添加白名单等。



3. 远程重启/备份网关

可以通过云平台对 ZigBee 网关进行重启 / 复位等操作。



4. 远程给网关下发命令

可以远程给网关下发命令，进行设备控制。

5. 远程升级ZigBee网关

当 ZigBee 网关功能升级或者修改重要 bug 时，可以通过云端进行远程固件升级。

6. 网关通信统计

可以远程查看网关的信道负载率、网络拓扑图，还可以实时扫描信道绘制出信道图。

【产品应用】基于DTU&ZWS云组态，快速搭建智慧环境监测系统

ZLG 致远电子 2023-09-06 11:36:28

为了对环境数据进行精细化监测，辅助实现高效的环境治理，基于DTU&ZWS 云组态，能够集成大气、水质、土壤等环境数据，进行组态数据大屏展示，实时掌握环境污染状况。

简介

随着中国生态环境保护工作的推进，环境监测范围越来越广，气方面有 PM2.5、臭氧、碳排放和柴油货车污染治理等，水方面有黑臭水体、水库、江河治理等，土方面有山体滑坡、农业污染治理等。不断增加的环境监测需求，迫切需要一个快速响应的环境监测管理系统。

CATCOM-100 是一款智能网联 DTU 终端，它能够将串口设备的数据通过无线网络传输到云端服务器，实现设备的远程管控，常被应用于环境监测、电力配电、智慧工厂等多种行业场景。

智能网联DTU终端
CATCOM-100

点击购买

ZWS-IoT 低代码开发平台支持一站式可视化开发组态应用，分钟级所见即所得的构建业务大屏页面，具有灵活编排、“积木式”搭建、轻松发布等特点。

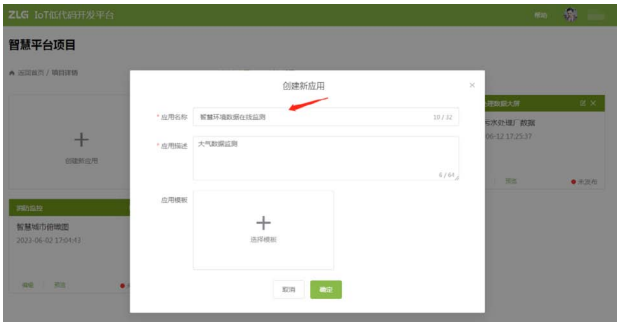
以大气环境监测系统为例，通过 CATCOM-100 将空气质量传感器接入 ZWS 云平台，能够监测大气环境数据，然后，可以用 ZWS-IoT 低代码平台自助搭建一个专属化的智慧环境监测系统。

那么，如何用 ZWS-IoT 低代码平台搭建一个环境监测系统呢？

搭建智慧环境监测系统

1. 创建web应用

登录 ZWS 物联网云平台，点击“IoT 低代码平台”入口，创建项目，添加一个新的应用。

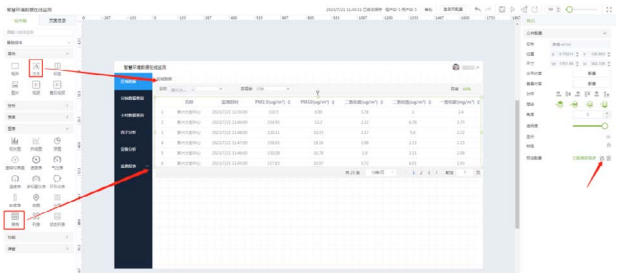


2. 顶部栏和导航栏配置

在右侧属性配置栏目中，找到【顶部栏】和【左导航】，启用对应的【排列方式】；还可以配置它的 宽度、高度、logo、颜色、模式 等基本属性。

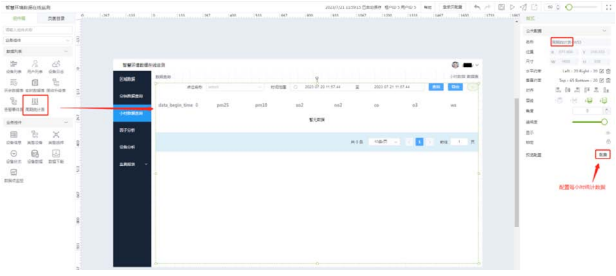
3. 区域数据页面

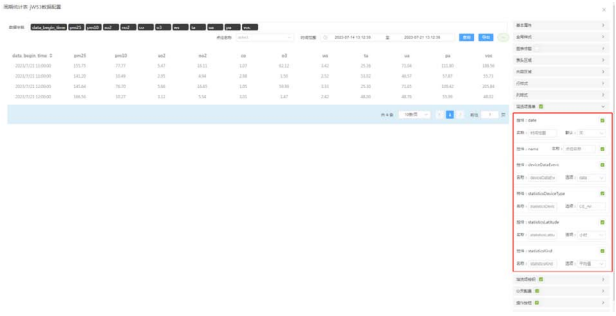
拖拽文本、下拉框、表格等组件，设置组件样式，配置表格数据源为环境数据。



4. 每小时数据查询页面

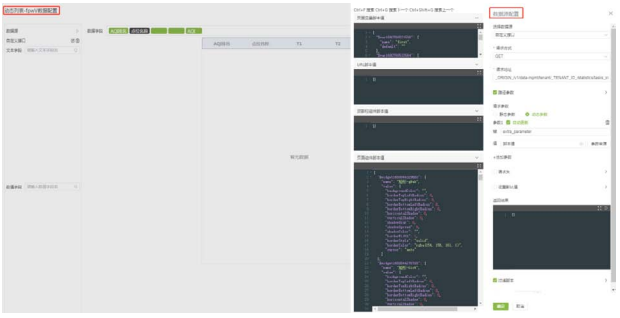
拖拽周期统计表组件，配置数据源为每小时统计数据。





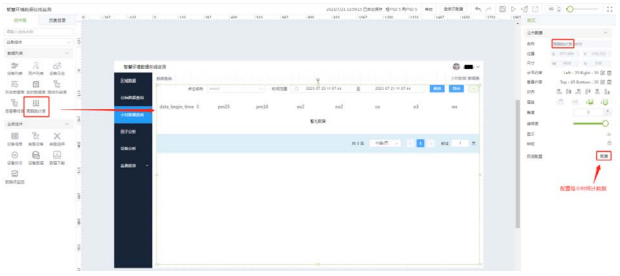
5. 报表类页面

可以拖拽动态列表组件，组成报表类页面。



6. 发布应用

完成环境监测系统页面搭建后，可以发布，生成一个唯一的链接，后续可通过该链接来访问系统。



【新品发布】ZWS云平台&CATCOM-100赋能工厂实现“云上设备管理”

ZLG 致远电子 2023-09-15 11:36:04

传统制造业数字化转型是大势所趋，然而“往哪转”、“如何转”成为了困扰众多工厂主的难题。本文将通过介绍 DTU 终端 CATCOM-100 和 ZWS 云平台，提供面向通用型工业设备的云端运维和智能控制解决方案。

场景介绍

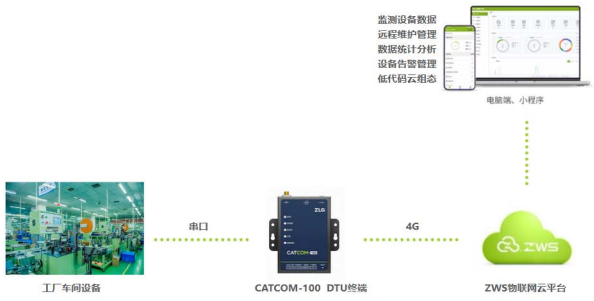
“工厂设备多，光是运维人员就要配备一大批人”、“车间内中央空调、空压机、焊接设备能耗都很大，一不留神就产生了大量的电费”、“我们需要经常监测房间内温湿度值，时刻关注车间压差和排风情况”、“经常不知道机器的内部运行情况，机器里面出了小问题也无法及时检修”……这一个个工厂主经常遇到的问题背后是传统的设备运维管理方式已经较难满足工厂的需求，设备管理混乱，运维低效已经成为了制造业企业发展的掣肘。

难道就没办法解决这些问题？答案是有的，设备管理的数字化、云端化转型能更好的帮助工厂解决这些设备管理上的问题，ZWS 物联网云平台正是帮助企业进行“云上设备管理”，实现设备管理数字化转型的“桥梁”。

ZWS 物联网云平台立足于制造业设备管理的场景，通过简化设备上云流程、云端监控与实时告警、支持设备数据快速生成管理报表等功能，赋能工厂实现设备的云端管理，助力企业实现设备精益管理和设备运营成本控制。

工业设备远程运维解决方案

工厂设备会通过串口与 CATCOM-100 连接，CATCOM-100 会将采集的设备数据直接透传上云，设备管理员和企业决策层能通过移动端的 APP 或者智慧工厂监控大屏查看到设备实时运行数据，基于数据进行下一步运维或管理决策。



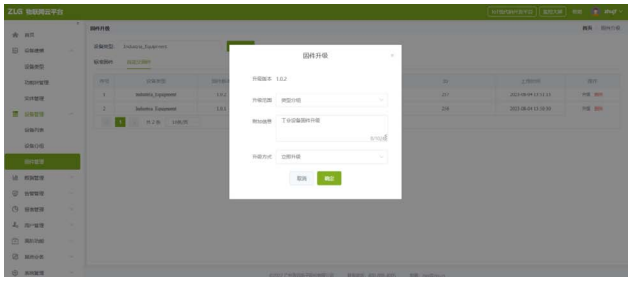
ZWS物联网云平台功能介绍

1. 设备管理

ZWS 物联网云平台能够快速接入多种协议的工业设备，进行远程数据监控和管理，比如空压机的状态监测。

2. 远程升级/配置

支持云端固件升级功能，能对设备进行远程升级和配置。



3. 数据大盘&数据统计分析

系统具备数据大盘和报表功能，数据大盘由多个看板组成，用户可以按照实际需求自行创建曲线图、折线图、柱状图等多形态看板，进行工业设备运行数据的展示；报表功能可以对数据进行多维度计算分析后得到结果报表。

4. 告警维护

系统具备告警管理功能，允许用户配置告警规则，系统依据规则对工厂设备的运行状态和数据进行实时告警。支持通过告警列表 / 手机短信 / 邮件等多种形式对设备管理员进行告警提醒。

5. 低代码搭建可视化大屏

依托 ZWS-IoT 低代码开发平台，制造业企业能够使用搭积木的方式，通过“拖拉拽”组件快速搭建出企业专属的智慧工厂数据可视化大屏和相关应用。

【产品应用】AWorksLP 样例详解（MR6750）——双核烧录

ZLG 致远电子 2023-09-20 11:35:23

AWorksLP 对外设进行了高度抽象化，为同一类外设提供了相同的接口，应用程序可以轻松跨平台。本文以 MR6750 平台为例，介绍 AWorksLP 双核烧录的方法。

简介

MR6750 双核是集成了两个 RISC-V 处理器，是两个完全独立的 CPU，故 HPM 双核工程是 Core0 工程和 Core1 工程两个独立的工程。因此 HPM 双核工程编译，其实是两个独立的单核工程的编译。用户只需要建立 core0 和 core1 的各自工程编译调试即可。

双核固件烧录

双核的例程是由 hart0 启动 hart1 的方式，所以需要两个固件，即 hart0 的固件与 hart1 的固件。采用汇编 incbin "xxx.bin" 的方式，将 hart1 的固件包含在 hart0 的固件中，所以最终 hart0 与 hart1 的固件合并成一个 elf 或者 bin 文件，由调试器下载的芯片中 hart0 的固件是在 flash 中运行，hart1 的固件是在 SDRAM 中运行。

注：HPM6750 总是从 hart0 启动，因此 hart0 是主核，hart1 是从核，hart1 作为从核不能自主启动，必须由 hart0 来启动。

{SDK}\demos\multi-core 路径下为 6750 双核例程，hello 例程是最基础的双核例程，openamp 和 rpc 是两种多核处理器框架、本小节将基于 hello 例程介绍双核的程序如何烧录。

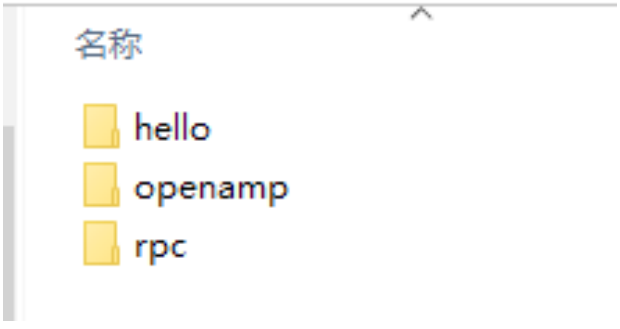


图1 例程目录

1. 创建工程

AWorksLP SDK 相关使用请参考《AWorksLP SDK 快速入门（MR6450）——开箱体验》一文，本文不在赘述。

hello 目录下有两个工程，hart0 和 hart1，分别对应主核和从核的工程、在选择板卡时需要选择 EPC6750-AWI-muti 板卡。

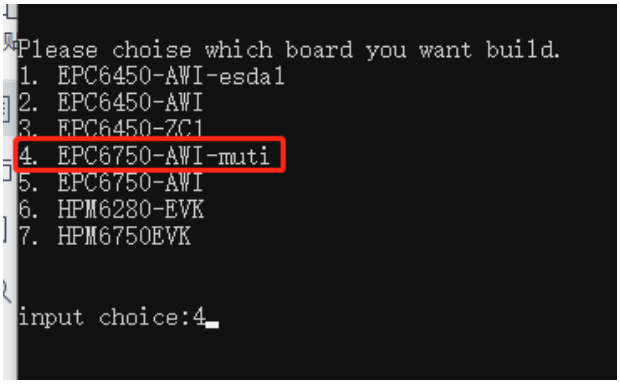


图2 选择板卡

2. 编译hart1固件

在 hart1 工程的配置中选择输出文件类型为 Raw binary。

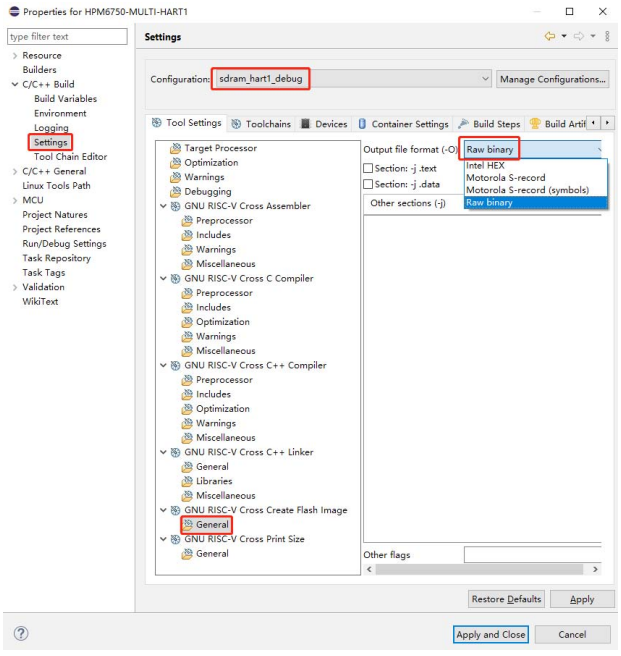


图3 Hart1工程配置

Hart1 工程选择 2 sdram_hart1_debug 编译方式，从核的程序将在 SDRAM 中运行。编译完成后将会在当前工程目录下的 hart1\project_eclipse\sdram_hart1_debug 路径下生成 HPM6750-MULTI-HART1.bin 文件。

图4 编译方式

将 HPM6750-MULTI-HART1.bin 固件拷贝到对应的板卡路径中的 source 文件中、{SDK}\platforms\platform-hpm-aworks-lp\boards\EPC6750-AWI-muti\source。

注意：hpm_hart1_image.S 文件中的名称要与从核的固件名称相同。

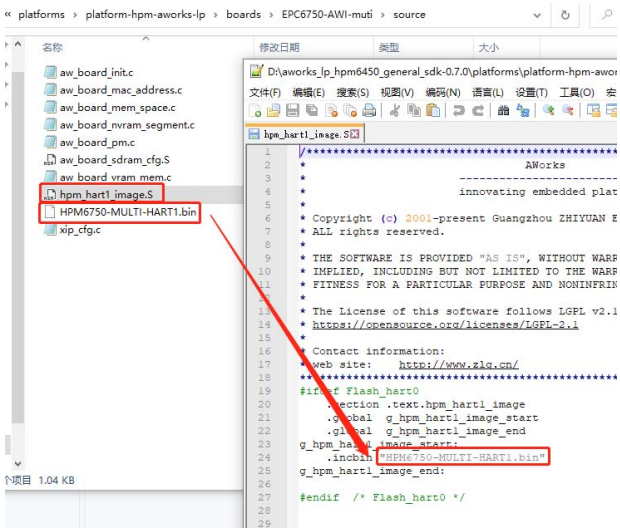


图5 拷贝固件

3. 编译hart0固件

将 hart1 固件拷贝完成后、编译 hart0 固件。主核的程序将在 flash 中运行、烧录完成后按一下复位键。

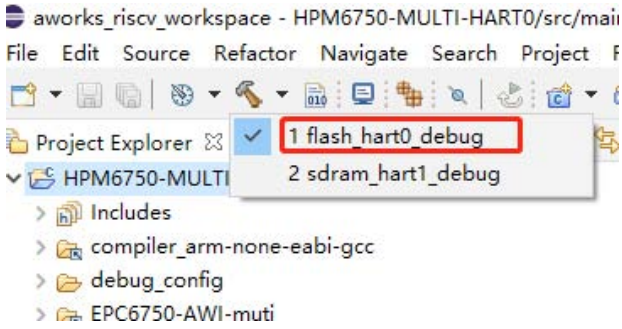


图6 hart0编译

4. 例程

将 hart1 固件拷贝完成后、编译 hart0 固件。主核的程序将在 flash 中运行、烧录完成后按一下复位键。

```
int aw_main()
{
  aw_kprintf("\r\napplication Start..... \r\n");

  while(1) {

    aw_kprintf("hart1: hello world!\n");
    aw_mdelay(1000);
  }

  return 0;

}
```

Hart0 的工程源码如下在 __start_hart1(void) 函数中使用 aw_open 接口打开核从核、打开后在对应的调试串口中打印”hart0：hello world!\n”。

```
static void __start_hart1(void)
{
  int fd;
  fd = aw_open("/dev/multi_core", AW_O_RDWR, 0);
  if (fd < 0) {
    aw_kprintf("open error, fd: %d\n", fd);
  }
  aw_kprintf("open multi_core ok!\n");
}

int aw_main()
{
  aw_kprintf("\r\napplication Start..... \r\n");

  __start_hart1();

  while(1) {

    aw_kprintf("hart0: hello world!\n");
    aw_mdelay(1000);
  }

  return 0;

}
```

在 {SDK}\platforms\platform-hpm-aworks-lp\boards\EPC6750-AWI-muti\EPC6750-AWI-muti.dts 文件中定义了 hart0 和 hart1 两个工程分别使用的调试串口号、当使用 hart1 工程时调试串口为 uart13，当使用 hart0 工程时调试串口为 uart0。

注意：若使用的时 EPC6750-AWI-L 评估板，评估板的 uart13 串口对应的硬件接的是 485，需在 gui 上使能 485 功能才能使用，也可直接将 uart13 改为 uart5 在排针上有对应接口直接使用即可。

```
/* 修改hart1 默认debug uart为uart13，并将uart0的驱动兼容性修改为普通串口 */
#ifdef SDRAM_hart1

&uart13{
    compatible = "riscv,hpm_uart_dbg";
    status = "okay";
};

&systmr {
    reg = <0xF300C000 0x4000>;
    interrupts = <IRQn_TMR3 0>;
    dev_clk = <HPM_CLK_GTMR3>;
};
#else

&uart0{
    compatible = "riscv,hpm_uart_dbg";
    default_baudrate = <115200>;
};

#endif
```

图7 调试串口

代码运行后 hart0 和 hart1 所对应的串口的打印信息分别为：

```
[15:25:13.459]收←◆
application Start.....
hart1 image start addr: 0x40000000
hart1 image size: 248972 bytes
hart1 image entry: 0x40000000
open multi_core ok!
hart0: hello world!
□[36m[AWorks->>> □[0m
[15:25:14.506]收←◆hart0: hello world!

[15:25:15.509]收←◆hart0: hello world!

[15:25:16.510]收←◆hart0: hello world!

[15:25:17.513]收←◆hart0: hello world!

[15:25:18.515]收←◆hart0: hello world!

[15:25:19.518]收←◆hart0: hello world!
```

```
[15:27:45.298]收←◆
application Start.....
hart1: hello world!

[15:27:46.303]收←◆hart1: hello world!

[15:27:47.305]收←◆hart1: hello world!

[15:27:48.307]收←◆hart1: hello world!

[15:27:49.310]收←◆hart1: hello world!

[15:27:50.311]收←◆hart1: hello world!

[15:27:51.314]收←◆hart1: hello world!

[15:27:52.316]收←◆hart1: hello world!

[15:27:53.319]收←◆hart1: hello world!

[15:27:54.326]收←◆hart1: hello world!
```

【产品应用】AWorksLP 样例详解（MR6750）——双核调试

ZLG 致远电子 2023-09-22 11:32:34

AWorksLP 对外设进行了高度抽象化，为同一类外设提供了相同的接口，应用程序可以轻松跨平台。本文以 MR6750 平台为例，介绍 AWorksLP 双核调试的基本用法。

简介

网络通信其实就是不同的设备按同一套协议来互传数据。这些协议包括了标准协议和非标准协议。其中最经典的标准协议模型是应用广泛的 TCP/IP 参考模型，涉及应用层、传输层、网络层和网络接口层。

MR6750 双核是集成了两个 RISC-V 处理器，是两个完全独立的 CPU，故 HPM 双核工程是 Core0 工程和 Core1 工程两个独立的工程。因此 HPM 双核工程编译，其实是两个独立的单核工程的编译。用户只需要建立 core0 和 core1 的各自工程编译调试即可。

多核调试

1. 修改默认工程配置

修改 hart1 工程 OpenOCD 的调试配置中的端口号。

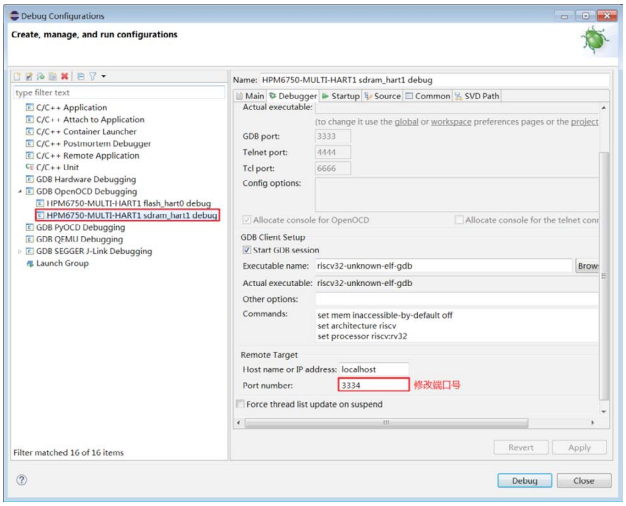


图1 修改端口号

在 Startup 配置栏中去除复位选项。

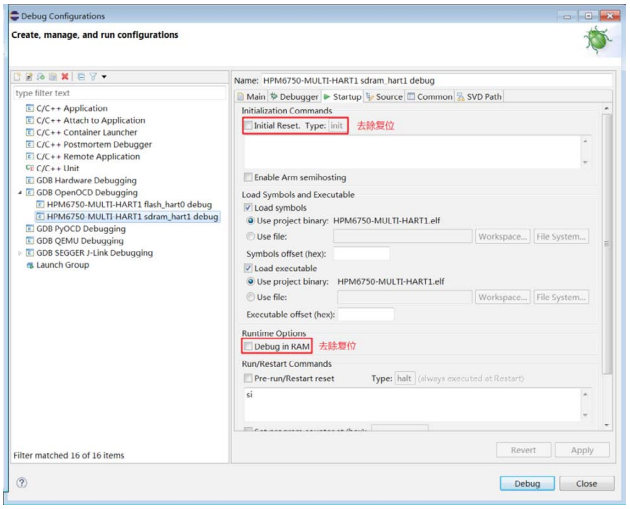


图2 去除复位选项

2. OpenOCD调试

在多核调试前建议将启动方式设置为串行启动流程或者串行下载流程。

BOOT1	BOOT0	
0	0	XIP NOR 启动
0	1	串行启动流程
1	0	串行下载流程
1	1	进入while(1)loop

图3 启动方式

多核调试下会分配两个 gdb 端口，分别是 3333 与 3334 分别对应 hart0 与 hart1，端口配置参考 4.1 小节。

```
Please choose board:
1. SmartU-HPM1000
2. HPM6750EVK
3. HPM6750EVK-MINI
4. EPC6750-AWI


your choice:2 根据实际的board选则
Multi-core support?(default N) [Y/N]:y      多核调试的支持
Multitasking debugging(default N) [Y/N]:n  多任务调试的支持
core_cfg_cmd="device_cfg/common/core1.cfg"
aworks_rtos_cfg=""
$$$$$$$$\ $$\       $$$$$$\
\____$$  |$$ |     $$  __$$\
    $$  / $$ |     $$ /  \__|
   $$  /  $$ |     $$ |$$$$\
  $$  /   $$ |     $$ |\_$$ |
 $$  /    $$ |     $$ |  $$ |
$$$$$$$$\ $$$$$$$$\\$$$$$$  |
\________|\________|\______/
Open On-Chip Debugger 0.11.0+dev-00302-ge9433e107-dirty (2022-10-10-14:38)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Info : auto-selecting first available session transport "jtag". To override use
init_sdram
force hard breakpoints
flash breakpoints: enable
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : ftdi: if you experience problems at higher adapter clocks, try the comman
Info : clock speed 10000 kHz
Info : JTAG tap: yitian.cpu tap/device found: 0x1000563d (mfg: 0x31e (Andes Tech
Info : datacount=4 progbufsize=8
Info : Examined RISC-U core; found 2 harts
Info :  hart 0: XLEN=32, misa=0x4094112d
Info :  hart 1: currently disabled
Info : [0] Found 2 triggers
release core1!
target0-event examine-end!
Info : datacount=4 progbufsize=8
Info : Examined RISC-U core; found 2 harts
Info :  hart 0: currently disabled
Info :  hart 1: XLEN=32, misa=0x4094112d
Info : starting gdb server for yitian.cpu0 on 3333
Info : Listening on port 3333 for gdb connections
Info : starting gdb server for yitian.cpu1 on 3334
Info : Listening on port 3334 for gdb connections
```

图4 OpenOCD调试

先下载 hart0 的固件，再下载 hart1 的固件。注意顺序不能错。

注：在调试状态下，hart0 不会将 hart1 的固件加载到相应的内存上，所以这里需要分别下载。

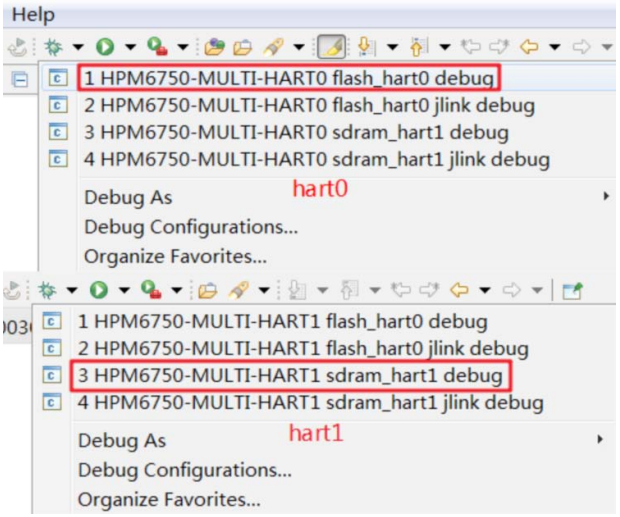


图5 OpenOCD下载

【产品应用】AWorksLP 样例详解（MR6750）——双核通信

ZLG 致远电子 2023-09-26 11:35:52

AWorksLP 对外设进行了高度抽象化，为同一类外设提供了相同的接口，应用程序可以轻松跨平台。本文以 MR6750 平台为例，介绍 AWorksLP 双核通信的基本用法。

简介

通信信箱 MBX 有 2 套寄存器访问接口，接口 A 和接口 B。A 和 B 接口都具有一套 TX FIFO 寄存器、RX FIFO

寄存器、控制寄存器和状态寄存器。用户从 A 接口的发送端 TX 发送的数据，可以在 B 接口的接收端 RX 接收 到。同理，A 接口的接收端 RX 可以接收到 B 接口发送端 TX 发送的数据。

- 双核烧录的用法请参考《AWorksLP 样例详解（MR6750）——双核烧录》
- 双核调试的用法请参考《AWorksLP 样例详解（MR6750）——双核调试》

双核通信

1. MBX信箱

{SDK}\demos\multi-core\openamp 路径下为 openamp 的例程。双核通信需要使用信箱在 gui 上勾选对应的信箱接口，hart0 和 hart1 需勾选同一个信箱的两个不同接口。例如 hart0 勾选了 mbx0a、则 hart1 需勾选 mbx0b。

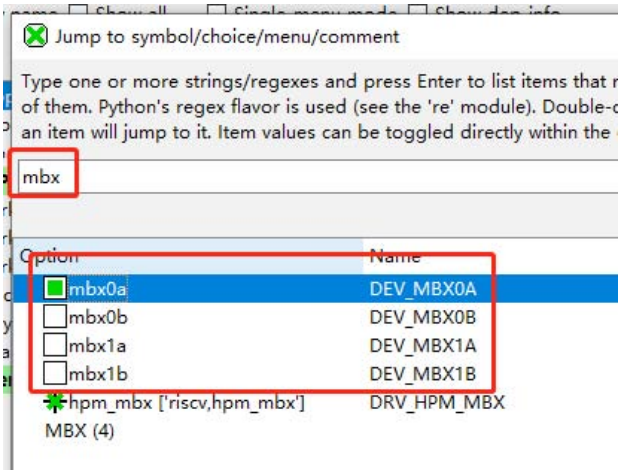


图1 mbx设备

3. 例程

```
#if CONFIG_AW_OPENAMP_MASTER
aw_local int rx_callback (struct rpmsg_endpoint *ept, void *data,
        size_t len, uint32_t src, void *priv)
{
  aw_kprintf("[Master receive]: %s\n", data);
  return 0;
}
#else
aw_local int rx_callback (struct rpmsg_endpoint *ept, void *data,
        size_t len, uint32_t src, void *priv)
{
  char sendbuf[512];

  aw_kprintf("[Slave receive]: %s\n", data);
  aw_snprintf(sendbuf, sizeof(sendbuf), "%s ACK", data);
   if (rpmsg_send(&__resmgr_ept, sendbuf, strlen(sendbuf) + 1) <
0) {
    aw_kprintf("[Slave send]: error!\n");
  }
  return 0;
}
#endif


aw_local int __mail_box_notify(void *priv, uint32_t id)
{
  uint32_t tmp;

#if CONFIG_AW_OPENAMP_MASTER
  /* master to remote */
  if (id == VRING1_ID) {
    /* send msg */
    tmp = EPT_SEND_MSG_FLAG;
  } else { /* remote to master */
    /* send ack */
    tmp = EPT_SEND_ACK_FLAG;
  }
#else
  if (id == VRING1_ID) {
    /* send ack */
    tmp = EPT_SEND_ACK_FLAG;
  } else {
      aw_read(__g_mbx_fd, &tmp, 4);
}
```

```
/* 处理其它设备发送过来的 MBX */
aw_local void __openamp_task(void *p_arg)
{

    struct rpmsg_virtio_device *p_dev = (struct rpmsg_virtio_device
*)p_arg;

    aw_kprintf("Entry OpenAMP task!\n");

    while(1) {
      uint32_t tmp;

      aw_read(__g_mbx_fd, &tmp, 4);

      /*
       * 默认 Master VRING0 是接收 , VRING1 是发送 , 从机反之
       */
#if CONFIG_AW_OPENAMP_MASTER
      if (tmp == EPT_SEND_MSG_FLAG) {

        /* 接收到来自从机的消息 */
        rproc_virtio_notified(p_dev->vdev, VRING0_ID);
      } else {

        /* 接收到来自从机的 ACK */
        rproc_virtio_notified(p_dev->vdev, VRING1_ID);
      }
#else
if (tmp == EPT_SEND_MSG_FLAG) {
        rproc_virtio_notified(p_dev->vdev, VRING1_ID);
      } else {
        rproc_virtio_notified(p_dev->vdev, VRING0_ID);
      }
#endif
    }
}

void rpmsg_demo()
{
  int ret = 0;
#if CONFIG_AW_OPENAMP_MASTER
  int i = 0;
  int RPMsgRole = 0;
#else
  int RPMsgRole = 1;
#endif

   __g_mbx_fd = aw_open(CONFIG_MBX_CHOOSE, AW_O_RDWR,
0);

    ret = aw_openamp_init(&rpmsg_dev, RPMsgRole, NULL, __
mail_box_notify);

#if CONFIG_AW_OPENAMP_MASTER
  /* 启动固件 */
  ret = aw_openamp_remoteproc_init(&__aworks_rproc_ops);
  if (ret) {
    aw_kprintf("Start processor fail!\n");
  }
#endif

  if (ret) {
    aw_kprintf("OpenAMP init error!\n");
    while(1);
  }

  aw_openamp_create_ept(&rpmsg_dev,
                &__resmgr_ept,
                 "rpmsg-client-sample",
                 0xFFFFFFFF,
                 rx_callback, NULL);

  aw_openamp_ep_poll_task_start(&rpmsg_dev);
  aw_openamp_wait_ept_ready(&__resmgr_ept);

  while(1) {

#if CONFIG_AW_OPENAMP_MASTER
    char sendbuf[512];
    aw_snprintf(sendbuf, sizeof(sendbuf), "AWorks %d", i);
    aw_kprintf("[Master send]: %s\n", sendbuf);
      if (aw_openamp_send(&__resmgr_ept, sendbuf,
strlen(sendbuf) + 1) < 0) {
        aw_kprintf("[Master send]: error!\n");
      }
      i++;
#else
    aw_kprintf("Salve is alive!\n");
#endif
    aw_mdelay(100);
  }
}
```

由于篇幅原因本文仅截取部分关键代码。

- 在 rpmsg_demo 中使用 aw_open 打开信箱、使用 aw_openamp_init 并注册 __mail_box_notify 函数；
- 在 __mail_box_notify 函数中发送相应的标记、使用 aw_openamp_remoteproc_init 函数注册 __aworks_rproc_ops，参数中是各运行阶段的函数接口；
- 使用 aw_openamp_create_ept 函数注册 rx_callback 接收回调函数，当作为主核时打印从核发送的数据，当作为从核时将收到的数据发送回去；
- 使用 aw_openamp_ep_poll_task_start 函数创建一个任务，任务的函数入口为 __openamp_task，在 __openamp_task 函数中根据读到的标记做相应的处理。

- 使用 aw_openamp_wait_ept_ready 函数等待从机准备好。
- 在 while 循环中主核使用 aw_openamp_send 函数循环的发送数据、从核在 rx_callback 回调函数中将主核发送的数据发送回去、串口打印如下图。

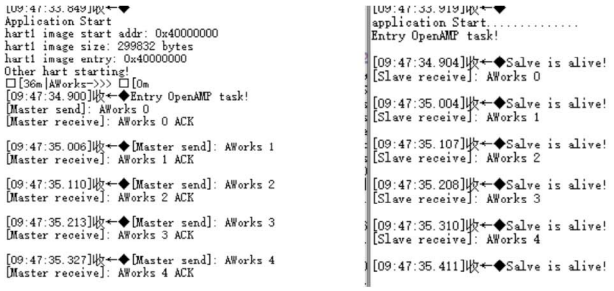


图2 串口打印

【产品应用】基于ZWS云平台对ZigBee网关的通信统计

ZLG 致远电子 2023-09-28 11:32:48

在 ZigBee 物联网应用中，ZigBee 网关是终端设备与云端通信联接的重要一环。本文将介绍 ZWS 物联网云平台支持从哪些方面对 ZigBee 网关进行通信统计分析。

应用场景

GZ32M 系列是工业级 ZigBee 物联网网关，是 ZLG 致远电子 ZigBee 生态系统的核心设备，能够实现多种有线、无线协议的传输，内置丰富的网络管理功能，且支持 MQTT 协议上云。适用于工业控制、智慧照明、智慧楼宇、数据采集等多种应用场景。GZ32M ZigBee 网关支持 ZigBee 协议终端设备接入，应用于低功耗远距离物联网场景。

GZ32M系列

点击购买

在 ZigBee 物联网应用中，ZigBee 网关是终端设备与云端通信联接的重要一环。为了让用户能实时获知 ZigBee 网关的通信情况，ZWS 物联网云平台支持网关的通信统计分析，可以从信道扫描、信道负载率、网络拓扑图三个方面进行查看分析。

前期准备

- GZ32M 系列 ZigBee 网关；
- 注册 / 登录 ZWS 物联网云平台。

首先，将网关接入 ZWS 云平台，并将网关上线，就能在云平台中对网关进行远程查看分析了。

1. 实时信道扫描

可以远程扫描周围 ZigBee 的信道，选择合适的信道，避免其他网络干扰。

2. 统计信道负载率

可以实时统计网关的信道负载率，分析上下行负载情况。

3. 查看网络拓扑图

可以在线获取网络拓扑图，远程查看组网效果。

【应用方案】核污水来了，我们能做什么？

ZLG 致远电子 2023-09-07 11:46:06

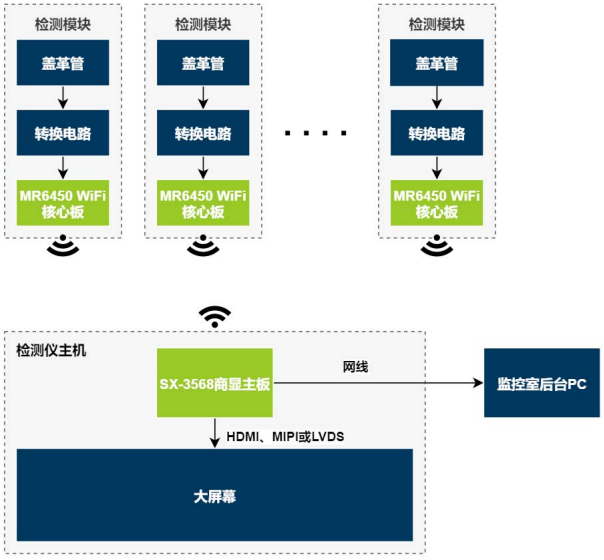
刚刚走出口罩时代的阴霾，又开始了辐射时代。面对坏消息，有人恐惧愤怒，有人砥砺向前，我们也可以做一些积极的应对。

2023 年 8 月 24 日中午 1 点，日本正式开始向太平洋中排放核污染水。核污染水一旦排海，放射性物质将在 57 天内扩散至太平洋大半区域，10 年后蔓延全球海域。目前我国已全面禁止进口日本海产品，但以前就时不时出现媒体报道无良商家伪造产地，以次充好的新闻，现在大家就更加担心自己吃到的食物有没有被污染，去海边游玩时也会担心海水是否干净。

但如果日料店、海鲜酒店、生鲜超市、滨海景区等场所，都能够实时监测核辐射并在大屏幕上显示辐射值，大家是否就能够放心地消费和游玩？

目前网上热卖的手持式辐射仪，明显很难满足场所监测的需求；传统的固定式核辐射监测报警仪，多应用于核工业与医学辐照场所，对精度和稳定性有更高的要求，而对人性化和可视化要求较低，也无法很好地满足商用核辐射仪的场景需求。

那么今天就分享一个基于 ZLG 致远电子的 SX-3568 商显主板和 MR6450 核心板的固定式核辐射检测仪方案：



如图，整个系统分为检测仪主机和检测模块。

1. 检测仪主机

主机由 SX-3568 商显主板和一块显示屏组成。SX-3568 商显主板通过 WiFi 与检测模块进行通信，并可通过网线连接后台的上位机，便于配置画面背景、显示样式等。

商显主板 SX-3568LI

点击购买

SX-3568LI/LC 是广州致远电子股份有限公司精心推出的一款基于 Rockchip 公司 RK3568 处理器的工控主板。主板标配处理器为 Cortex®-A55 四核，最高主频 2GHz 的 RK3568 处理器，内置 2/4GB DDR4 内存，8/16GB eMMC 存储。SX-3568 支持最高 4K@60fps，支持双屏异显，且有 HDMI、MIPI、LVDS、eDP 四种显示接口方式可选。

2. 检测模块

检测模块基于盖革 - 米勒计数器进行辐射的检测，这种盖革 - 米勒计数器技术成熟且简单便宜。该方案中，MR6450 将接收并监测盖革管发出的脉冲，并通过 WiFi 发送给检测仪主机，最终在主机显示屏上实时显示出各路模块的监测结果。

MR6450/6750系列

点击购买

MR6450 核心板是 ZLG 致远电子新推出的一款嵌入式核心板，内置 WiFi 模组，兼具丰富的功能外设以及多样化的入网接口方式，和极强的易用性、低功耗、实时运行以及低中断延迟特性。

方案中的 MR6450 核心板还可替换为 ZSL420/ZSL421 芯片，同时主机端也需要增加一个 ZSL420/ZSL421 芯片的模块。ZSL420/ZSL421 是致远电子推出的一款 LoRa 智能组网芯片，支持 LoRa、FSK、GFSK 等多种调制方式，结合频谱扩宽处理技术，旨在解决小数据量在复杂环境中的远距离通信问题。LoRa 传输距离更远，功耗更低，支持更多节点。

该方案非常适合 B2C 商业场景的使用，能实现专业、高档的呈现效果，WiFi 连接免于布线烦恼，大大减少装修改造对正常营业的影响，同时能够承受 -40~85°C工作温度，即使是海滩、码头等户外场所，也能稳定工作。SX-3568 商显主板接口资源丰富，方案灵活“可玩性”高，适配各种显示接口形式，还有丰富的通信总线资源如 RS485 和 CAN 总线，如果需要更高的稳定性，也可轻松实现。

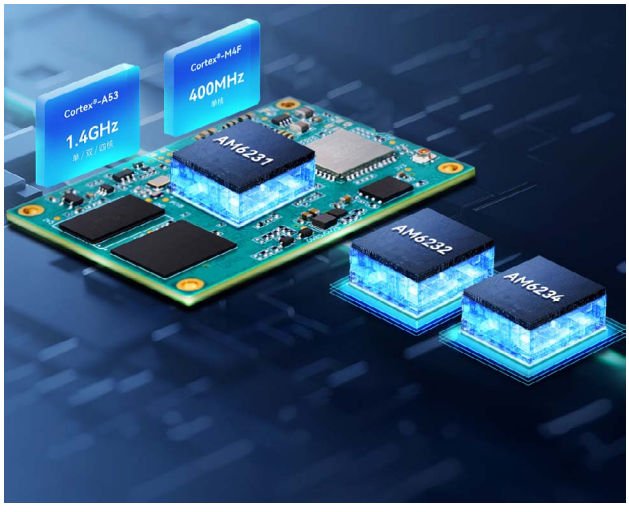
【新品发布】M62xx系列核心板：超越经典，AM335x升级之选

ZLG 致远电子 2023-09-11 11:52:31

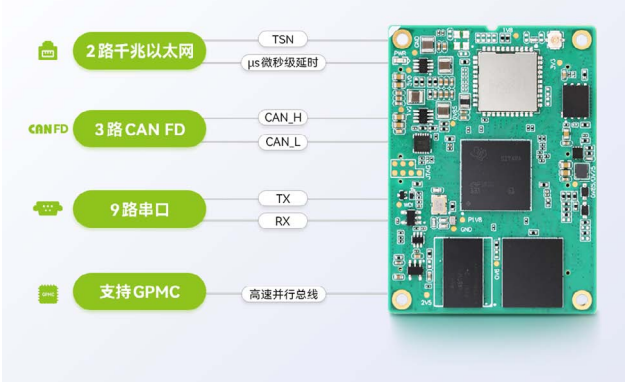
选型表

产品型号	M6231-1GF4GLI-T	M6232-1GF4GLI-T	M6234-1GF4GLI-T
处理器	AM6231	AM6232	AM6234
内核	单核 Arm® Cortex®-A53 + 单核 Arm® Cortex®-M4F	双核 Arm® Cortex®-A53 + 单核 Arm® Cortex®-M4F	四核 Arm® Cortex®-A53 + 单核 Arm® Cortex®-M4F
主频	1.4GHz + 400MHz	1.4GHz + 400MHz	1.4GHz + 400MHz
操作系统	Linux	Linux	Linux
运行内存（SDRAM）	1GB	1GB	1GB
板载存储（eMMC）	4GB	4GB	4GB
QSPI Flash	8MB	8MB	8MB
独立硬件看门狗	支持，1.6 秒溢出	支持，1.6 秒溢出	支持，1.6 秒溢出
3D 图形处理器	不支持	不支持	不支持
RGB-LCD	支持	支持	支持
LVDS	2 路，各 4 通道	2 路，各 4 通道	2 路，各 4 通道
显示分辨率	最高两路 1920x1080@60fps 或 1 路 2048x1080+ 1 路 1280x720	最高两路 1920x1080@60fps 或 1 路 2048x1080+ 1 路 1280x720	最高两路 1920x1080@60fps 或 1 路 2048x1080+ 1 路 1280x720
电阻触摸屏	IIC、SPI 或 USB 扩展支持	IIC、SPI 或 USB 扩展支持	IIC、SPI 或 USB 扩展支持
电容触摸屏	IIC 扩展支持	IIC 扩展支持	IIC 扩展支持
MIPI-CSI	1 路，4 通道	1 路，4 通道	1 路，4 通道
USB 2.0	2 路	2 路	2 路
以太网	2 路千兆	2 路千兆	2 路千兆
SD/SDIO	2 路，其中 1 路用于 TF 卡	2 路，其中 1 路用于 TF 卡	2 路，其中 1 路用于 TF 卡
UART	9 路（含 1 路调试串口）	9 路（含 1 路调试串口）	9 路（含 1 路调试串口）
CAN FD	3 路	3 路	3 路
SPI	最高 5 路	最高 5 路	最高 5 路
IIC	最高 5 路	最高 5 路	最高 5 路
IIS/MCASP	最高 3 路	最高 3 路	最高 3 路
ePWM	最高 3 路	最高 3 路	最高 3 路
eCAP 捕获	最高 3 路	最高 3 路	最高 3 路
eQEP正交编码器脉冲	最高 3 路	最高 3 路	最高 3 路
RTC 实时时钟	IIC 扩展支持	IIC 扩展支持	IIC 扩展支持
音频接口	IIS 扩展支持	IIS 扩展支持	IIS 扩展支持
供电电压	+5.0V	+5.0V	+5.0V
工作温度	-40℃ ~+85℃	-40℃ ~+85℃	-40℃ ~+85℃
机械尺寸	40mm * 55mm	40mm * 55mm	40mm * 55mm
评估底板	M62xx-EV-Board	M62xx-EV-Board	M62xx-EV-Board

多核异构，性能升级



丰富接口，连接更强大



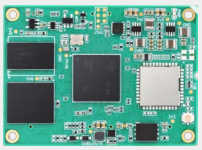
边缘计算▼

超越经典，AM325x升级之选

双屏异显，创见双倍精彩

严苛测试，工业级可靠

行业应用



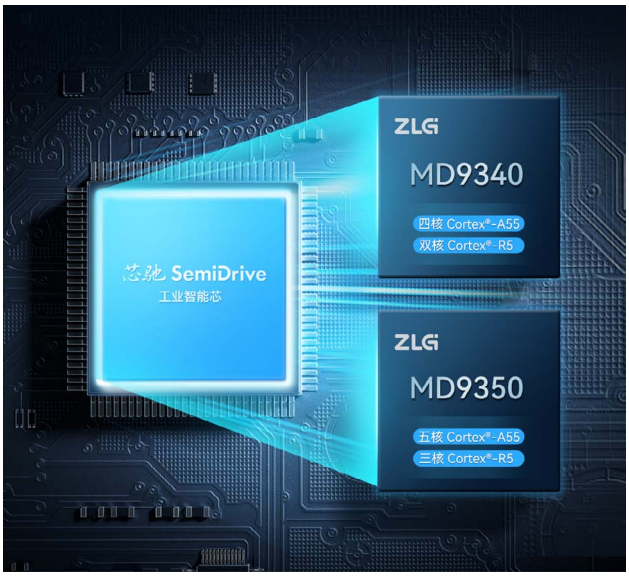
M62xx 核心板

点击购买

【新品发布】
D9核心板：强劲五核，国产芯篇章

ZLG 致远电子 2023-09-25 11:36:56

芯驰工业芯，多核异构，性能强劲



严苛测试，工业级可靠

产品型号	MD9340-2GF8GLI-T	MD9340-2GF16GLI-T	MD9350-2GF8GLI-T	MD9350-2GF16GLI-T
处理器	D9340	D9340	D9350	D9350
ARM 内核	四核 Cortex-A55® + 双核 Cortex-R5®	四核 Cortex-A55®+ 双核 Cortex-R5®	五核 Cortex-A55 ®+ 三核 Cortex-R5®	五核 Cortex-A55®+ 三核 Cortex-R5®
主频	1.6GHz + 800MHz	1.6GHz + 800MHz	1.6GHz + 800MHz	1.6GHz + 800MHz
操作系统	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS	Linux、Buildroot、Ubuntu、FreeRTOS
GPU Core	支持	支持	支持	支持
VPU Core	支持	支持	支持	支持
NPU	-	-	0.8TOPS	0.8TOPS
运行内存（SDRAM）	2GB	2GB	2GB	2GB
板载存储（eMMC）	8GB	16GB	8GB	16GB
QSPI Flash	16MB	16MB	16MB	16MB
LVDS 接口	2 路，4-Lane	2 路，4-Lane	2 路，4-Lane	2 路，4-Lane
显示分辨率	单路最大支持 1920x1080@60Hz 双路最大支持 2560x1440@60Hz	单路最大支持 1920x1080@60Hz 双路最大支持 2560x1440@60Hz	单路最大支持 1920x1080@60Hz 双路最大支持 2560x1440@60Hz	单路最大支持 1920x1080@60Hz 双路最大支持 2560x1440@60Hz
电阻触摸屏	IIC、SPI 或 USB 扩展支持	IIC、SPI 或 USB 扩展支持	IIC、SPI 或 USB 扩展支持	IIC、SPI 或 USB 扩展支持
电容触摸屏	IIC 扩展支持	IIC 扩展支持	IIC 扩展支持	IIC 扩展支持
MIPI-CSI	1 路，4-Lane	1 路，4-Lane	1 路，4-Lane	1 路，4-Lane
DVP-CSI	1 路，8 位并口	1 路，8 位并口	1 路，8 位并口	1 路，8 位并口
以太网	2 路千兆	2 路千兆	2 路千兆	2 路千兆
UART	10 路（含 1 路调试串口，核心板最大支持 16 路）	10 路（含 1 路调试串口，核心板最大支持 16 路）	10 路（含 1 路调试串口，核心板最大支持 16 路）	10 路（含 1 路调试串口，核心板最大支持 16 路）
CAN FD	4 路	4 路	4 路	4 路
SPI	1 路	1 路	1 路	1 路
IIC	最高 5 路	最高 5 路	最高 5 路	最高 5 路
IIS	1 路	1 路	1 路	1 路
PWM	5 路	5 路	5 路	5 路
SD/SDIO	2 路	2 路	2 路	2 路
独立硬件看门狗	支持，1.7 秒溢出	支持，1.7 秒溢出	支持，1.7 秒溢出	支持，1.7 秒溢出
供电电压	+5.0V	+5.0V	+5.0V	+5.0V
工作温度	-40°C ~+85°C	-40°C ~+85°C	-40°C ~+85°C	-40°C ~+85°C
机械尺寸	45mm * 65mm	45mm * 65mm	45mm * 65mm	45mm * 65mm
评估底板	MD9340-EV-Board（不含核心板）、MD9340-IO-Board			

D9系列核心板

点击购买

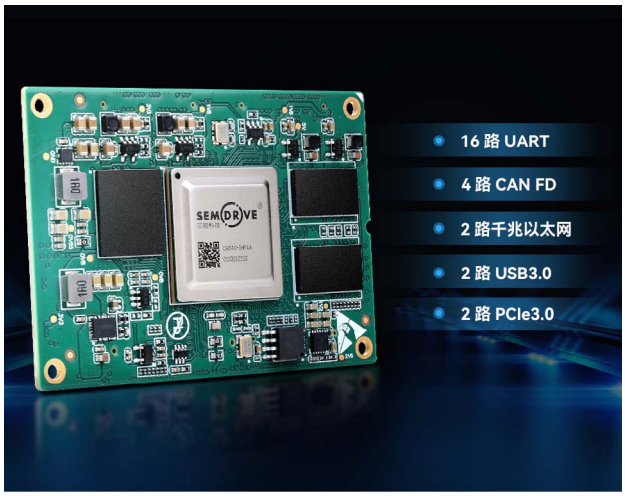
边缘计算▼

精准实时控制



工业自动化　　运动控制　　医疗设备

通讯接口丰富



双重ASIL认证呢，安全可靠

严苛测试，真正工业级

行业应用

【新品发布】EPC6750-AWI工控板搭载MR6750核心板，强势来袭

ZLG 致远电子 2023-07-03 11:36:18

MR6750 系列核心板基于先楫半导体的 HPM6750IVM2 开发，集成了两个 RISC-V 处理器，主频高，支持高速数据处理能力，具有丰富的通信接口，适合于工业控制、仪器仪表、电机控制等应用场合。

MR6750核心板介绍

1. 核心特点

双核 32 位 RISC-V 处理器，816MHz 主频，高于 9000 CoreMark™和 4500 以上的 DMIPS 性能，支持双精度浮点运算、DSP 扩展及硬件图形加速，内置大容量片上 SRAM 2MB；

- 板载 32MB SDRAM，8MB QSPI Flash；
- 支持 2.4GHz WiFi 及 Bluetooth 4.2；
- 双千兆以太网，iperf 测试带宽大于 900Mbps，也可配置成百兆模式；
- 支持 IEEE 1588 PTP 规范，时间同步精度可达 17ns；
- 四路 CANFD 接口，最高速率可达 8Mbps；
- 多种通信接口：15 路 UART，4 路 SPI，4 路 I2C；
- 双路高速 USB2.0，内置 PHY；
- 4 个 PWM 定时器，调制精度可达 2.5ns，
- 3 个 12 位 ADC，1 个 16 位 ADC。

2. 双核特点

核心板集成 2 个 RISC-V 处理器，双核采用主从结构。CPU0 和 CPU1 采用相同配置，如下：

- 支持相同指令集；
- 相同容量的 L1 指令和数据缓存：
- – 32KB L1 I-Cache，4-way，128x 64B cache line per way
- – 32KB L1 D-cache，4-way，128x 64B cache line per way
- 相同容量的指令和数据本地存储器：256 KB ILM 和 256 KB DLM。
- CPU0 和 CPU1 采用相同的存储器映射，以下为例外：
- CPU 自身的指令 / 数据本地存储器 ILM / DLM 为私有；
- FGPIO 为私有；
- 平台中断控制器 PLIC 为私有；
- 软件中断控制器 PLICSW 为私有；
- 机器定时器 MCHTMR 为私有。

MR6750核心板

点击购买

EPC6750-AWI单板介绍

EPC6750-AWI 工控单板是基于双 RISC-V 内核的 MR6750 核心板设计，板载丰富的硬件资源和接口，如 2 路百兆以太网、2 路 CAN FD、1 路 USB HOST、1 路 Type-C、1 路 TF 卡和 1 路 LCD 显示等接口。

EPC6750-AWI工控单板

点击购买

资源	配置说明
以太网	2 路 10/100Mbps（RJ45）
USB	1 路 Type-C 支持电源供电
	1 路 USB HOST 支 USB2.0 High speed
CANFD	2 路，支持 CAN FD/CAN
DVP	1 路 24PIN FFC 接口，支持 ov7725 型号摄像头
LCD	1 路 50PIN FFC 接口，支持（RGB565）
TF 卡	1 路 TF 接口
JTAG	1 路调试接口
UART	1 路显示串口接口
RTC 电池座子	1 路 CR1220 RTC 座子
PWM	蜂鸣器
BOOT	2 路模式选择
按键	支持复位
LED 灯	1 路电源显示，1 路运行显示

EPC6750-AWI 工控单板还支持 40PIN 的扩展接口，扩展接口资源如下：

接口资源	配置
RS485	1 路
UART	2 路
CAN	1 路
I2C	1 路
SPI	1 路
ADC	3 路

【产品应用】从中心到边缘：5G智能边缘计算网关为环境监测带来变革

ZLG 致远电子 2023-09-21 11:41:52

由广州致远电子股份有限公司推出的 5G 工业物联网边缘计算智能网关 EPCM3568B-LI/EPCM3568C-LI 结合 5G 和边缘计算两大技术的优势，为环境监测提供高效、灵活的解决方案。

环境监测，作为全球关注的焦点，一直在寻求技术上的突破以提高其效率和准确性。随着 5G 和边缘计算的兴起，我们正见证一个从中心到边缘的技术变革，这为环境监测带来了前所未有的机会。

传统的环境监测系统大多基于中心化的架构。这意味着所有的数据，无论来源于何处，都需要传输到中心服务器进行处理和分析。这不仅增加了数据传输的时间和成本，还可能因网络延迟或中断导致数据丢失。此外，中心服务器可能会成为瓶颈，限制系统的扩展性和响应速度。

5G 技术提供了高速、低延迟的网络连接，使得数据可以实时传输。而边缘计算则允许数据在产生地进行处理，减少了数据传输的需求。由广州致远电子股份有限公司推出的 5G 工业物联网边缘计算智能网关 EPCM3568B-LI/EPCM3568C-LI 结合了这两大技术的优势，为环境监测提供了一个高效、灵活的解决方案。

1. 数据采集与传输的高手

EPCM3568B-LI/EPCM3568C-LI 具有 6 路 RS485、2 路 CAN、4 路 ADC、8 路隔离 DO、8 路隔离 DI、2 路千兆以太网，通过丰富的通讯接口与环境监测中的常用到的如：空气采样器、颗粒物监测仪器、水质分析仪器、土壤湿度和温度计、声级计等模块直接或间接连接，实时采集并传输数据到中心服务器或云平台。这不仅确保了数据的实时性，也大大提高了数据的准确性。

2. 实时分析与响应

得益于其搭载的 RK3568 处理器，EPCM3568B-LI/EPCM3568C-LI 具有可靠的数据处理能力。当监测到某些关键参数超出正常范围时，它可以立即进行实时分析，并发出警报或与其他设备协同，如启动空气净化器或调整排放系统，实现快速响应。

3. 云端与边缘的完美结合

EPCM3568B-LI/EPCM3568C-LI 支持多种云平台，如致远 ZWS 云、阿里云、腾讯云等，实现了云端与边缘的协同工作。这意味着，无论是数据存储、分析还是共享，都可以在云端和边缘之间自由切换，满足不同的应用需求。

4. 工业级的稳定性

在环境监测领域，稳定性是关键。EPCM3568B-LI/EPCM3568C-LI 的工作温度范围为 -40°C ~ +70°C，能够适应各种恶劣环境，确保长时间稳定运行。

5. 开放与兼容

EPCM3568B-LI/EPCM3568C-LI 支持多种工业协议，如 Modbus TCP/RTU、OPC UA 等，使其可以轻松集成到现有的环境监测系统中，为系统升级提供了便利。

从中心到边缘的技术变革为环境监测开辟了新的道路，随着更多的应用案例，我们有理由相信，EPCM3568B-LI/EPCM3568C-LI 5G 工业物联网边缘计算智能网关将在环境监测领域发挥更大的作用。

【产品应用】
如何精准分析人形机器人运动数据?

ZLG 致远电子 2023-09-14 11:35:03

全球“机器换人”进程加速，人形机器人有望成为 AI 下一个重要落地应用场景；EtherCAT-Analyzer 具备分析人形机器人所有关节和电池与主站的通讯信息，快速掌握节点网络状态！

前言

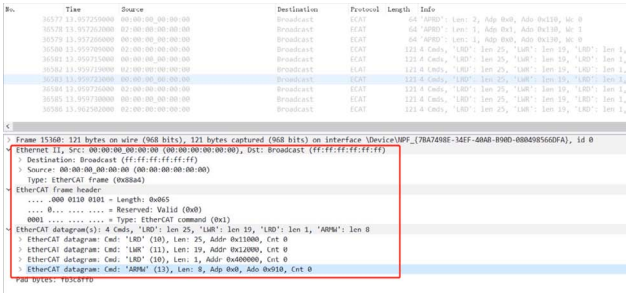
随着人形机器人行业的发展及《中国制造 2025》的全面实施，传统的脉冲模式控制很大程度上制约了机器人的性能，相反，高速工业现场总线 EtherCAT 利用以太网协议完成工业自动化控制，满足了工业控制工程中通信稳定、大数据量，低延时性的要求，成为机器人主力发展的总线解决方案，且其易于与现场其它设备组网，为实现自动化流水线及建立智能工厂提供了基础。

人形机器人关节作为 EtherCAT 从站，通过主站控制器轻松完成通信控制。但是，主从通讯过程中经常出现的通讯丢帧、延迟和断开连接等多种问题，严重影响工程师们对于机器人的调试与控制，此时迫切需要专业的分析设备快速定位问题点，“对症下药”，解决问题。

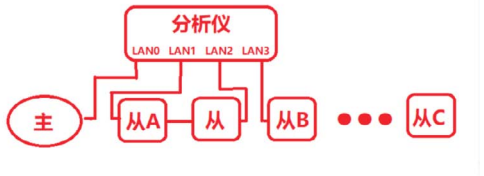
致远电子 EtherCAT 分析仪 EtherCAT-Analyzer 应运而生，其具有网络帧统计（流量、转发延时、错误帧、周期抖动）、网络帧时间分析、网络帧内容分析、GPIO 事件分析以及帧数据存储等功能，可快速分析评估人形机器人网络状态，给与“良药”快速解决问题。

wireshark解析协议,定位主从通讯问题

上位机连接上 EtherCAT 分析仪，软件映射虚拟网卡，轻松查看主从初始化交互信息，获取从站配置参数，掌握各厂家协议差异，解决适配难题；还可通过实时拓扑，判断是否为“设备丢帧”、“设备不处理数据”、“设备断开连接”等问题。



解析协议



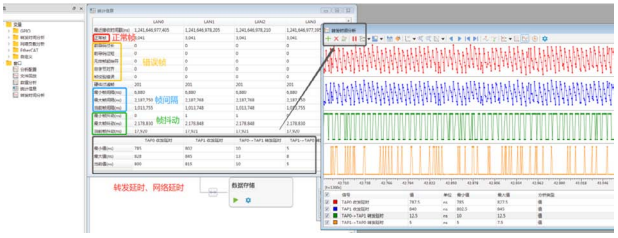
观察上位机的统计信息，可以呈现如下特征：

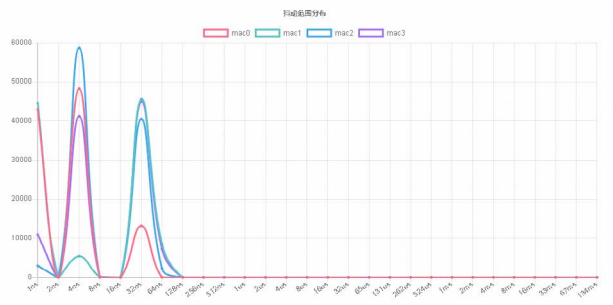
定位主从通讯问题

互联互通▼

全面统计，便于诊断

上位机连接上 EtherCAT 分析仪，支持查看四个端口的正常帧、错误帧、帧间隔、帧抖动、转发延时、网络延时等统计信息。

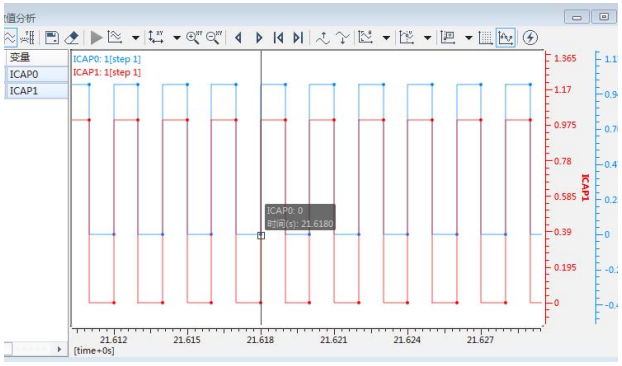




抖动分析

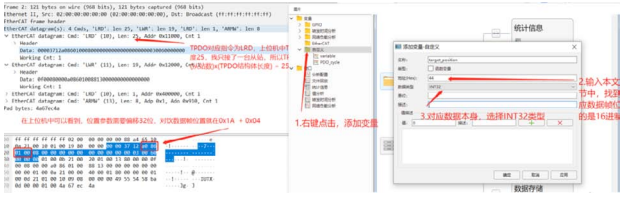
支持GPIO事件捕获，满足网络同步分析需求

设备支持 GPIO 事件监控、抓取从站 DC 中断信号，从而调整合适的偏移时间。



直观的数值曲线，轻松判断从站执行效率

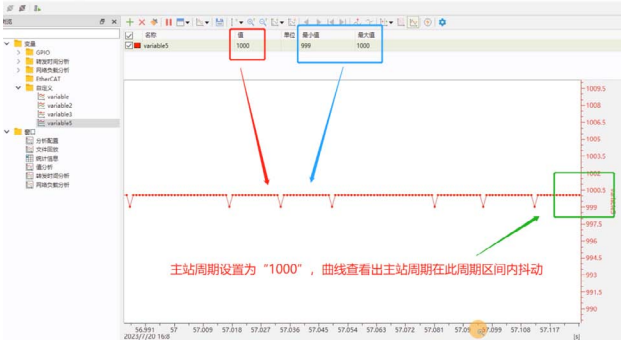
EtherCAT 分析仪支持帧内数值曲线分析，支持按帧偏移和任意扩展协议解析数值，人形机器人各关节电机和算法的跟随曲线分析，找到机械或算法性能瓶颈。



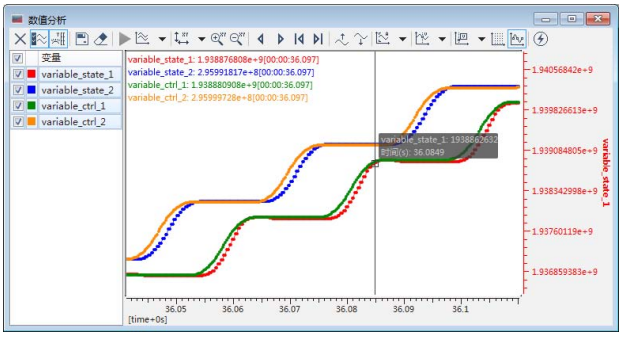
Wireshark确认PDO，上位机添加变量

<1ns的转发延迟，精准评估主站周期

EtherCAT 分析仪支持四个端口的帧转发时间分析，在多节点的 EtherCAT 网络中，不影响实际通信，精准分析主站实际 PDO 周期与抖动。



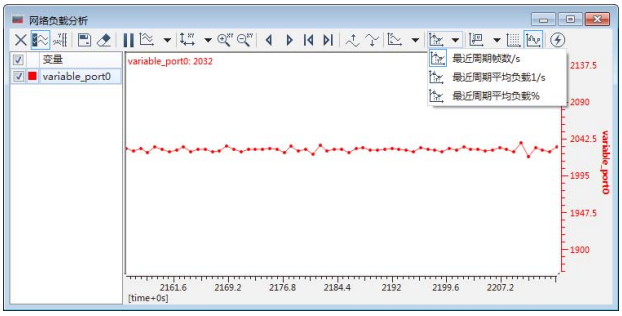
分析PDO周期



上位机显示曲线
（主从站输出曲线贴合度越高，从站执行效率越高）

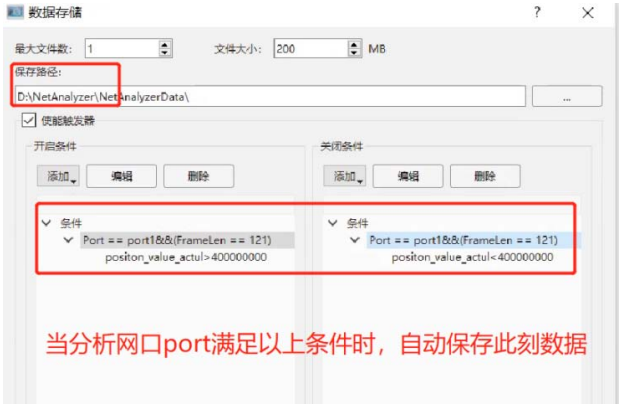
网络负载监测，实时掌握节点帧流量

EtherCAT 分析仪支持对指定数据源进行网络负载分析，同时支持设置过滤，可针对某个 PDO 进行帧流量分析。

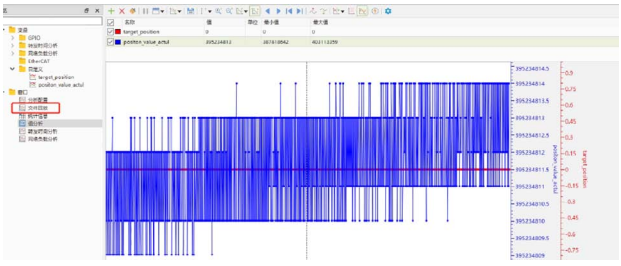


触发保存，数据回放

上位机配置触发条件，达到相应条件时候，上位机自动保存相关数据至相关路径；过后支持将保存的数据通过上位机重新回放分析。



触发保存



数据回放

搭配ZWS物联网云平台，快速实现系统智能化升级

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

互联互通▼

【技术分享】BLE蓝牙模块功能应用① — 主从一体

ZLG 致远电子 2023-09-08 11:36:31

随着科技快速发展，蓝牙技术在各个行业应用中变得非常广泛。今天，我们来分享一种非常热门的主从一体蓝牙模块。我们将对它的模式、概念、应用、选购等进行探讨，让读者深入了解这种蓝牙模块。

蓝牙的工作模式

蓝牙模块支持多种类型的运行模式，主设备模式、从设备模式、主从一体工作模式、广播站模式、Mesh 组网模式、广播模式、iBeacon 模式。在众多工作模式中，主从一体工作模式是蓝牙模块最为常用，也是当下最热门的。工作在该模式下的蓝牙产品既可以单独作为主机或者从机使用，也可以同时作为主机和从机，同时连接从设备和被其他主设备连接，极大提高蓝牙模块的灵活使用与应用。下图为广州致远电子股份有限公司推出 ZM52820P 蓝牙模块。ZM52820P 是一款低功耗、低成本、通信距离远、性能稳定的主从一体蓝牙模块。

图1 ZM52820P蓝牙模块

什么叫蓝牙主从一体

蓝牙主从一体工作模式是指蓝牙模块可以同时作为主设备和从设备。其可以在两个角色间切换，工作在从模式时，等待其它主设备来连接，需要时，转换为主模式，向其它设备发起连接调用。主从一体提供了扩展蓝牙模块的能力，在蓝牙 4.1 协议规范后，添加了“链路层拓扑 " 的功能，就可以允许蓝牙模块同时作为主设备和从设备，在任何角色组合中操作，例如蓝牙 HUB 终端。

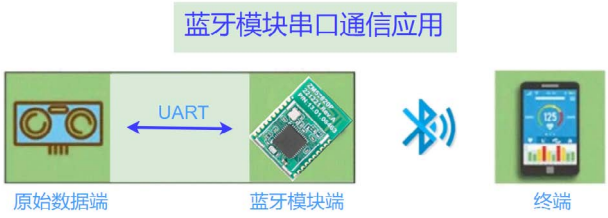
当具备主从一体的蓝牙模块工作时，该蓝牙模块可以作为主设备搜集其它外围从节点设备的信息，同时作为一个从设备将搜集到信息上报给主控终端如手机。这样的好处就是外围的从节点设备信息可以不局限在本地保存，通过主从一体的蓝牙模块发挥中继器的作用，搜集后上传给云端或集中控制器保存或显示。

主从一体额外增加了蓝牙模块的功能，成本优化和易用性。如果蓝牙模块以前在封闭系统中作为主设备工作，那现在还可以同时作为从机连接到智能手机，从而实现新的连接维度，在主从一体工作模式下，一个蓝牙模块就可以扮演两种角色，从而可以优化系统架构。

蓝牙主从一体应用

1. 串口通信

广州致远电子股份有限公司研发的蓝牙模块都是 BLE 蓝牙模块，支持数据传输，UART 接口，支持 Android、IOS 数据传输，简单易用，不需要复杂设置就可实现数据的快速传输。



2. 数据采集

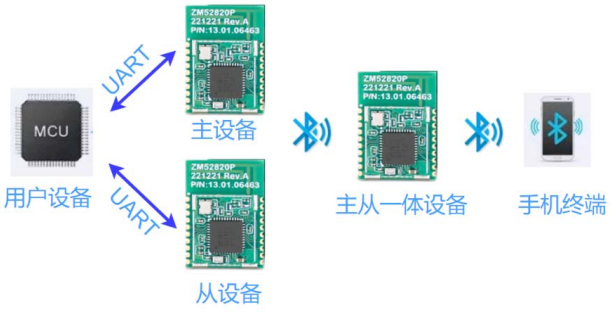
支持温度、湿度“等众多传感器数据采集应用，比如可以将温度传感器连接至蓝牙模块的 UART 接口，可以实现温度读取功能，再通过串口将采集的温度数据传输给到接收设备，如手机、平板、PC 或云平台等监控实时数据。

3. 智能控制

支持对接手机，可通过手机 APP 访问。比如可以将蓝牙模块嵌入到智能窗帘的电机或者是智能灯具中，通过手机 APP 来实现电机调速、LED 灯亮度调节。

4. 一对多数据传输

一对一传输，一对多传输，支持工作于主机模式、从机模式、主从一体，一主多从等工作模式，可满足物联网应用一对一、一对多的数据传输。



如何选择主从一体蓝牙模块

对于不同的应用场景，用户在选择主从一体蓝牙模块时，应根据设备的功能需求通讯距离、功耗要求、性能稳定等因素，综合考虑选择合适的主从一体蓝牙模块。在购买主从一体蓝牙模块的时候，用户应该选择质量可靠，性能稳定，价格合理的品牌，同时应了解产品的资质生产工艺等情况，确保所选的蓝牙模块能够满足自己的需求。

结语

主从一体蓝牙模块作为一种蓝牙通讯模块，具有功耗、高速率传输、易于集成、多种应用场景和简化开发流程等优势，已经广泛应用于各种智能设备中。对于促进智能物联网的发展，提高智能设备用户体验起到了积极的促进作用。在未来，随着智能设备市场的发展，主从一体蓝牙模块的使用会更加广泛，并为用户提供更好的服务。

互联互通▼

【新品发布】ZM5825系列工业级Wi-Fi+BLE模组全新上市

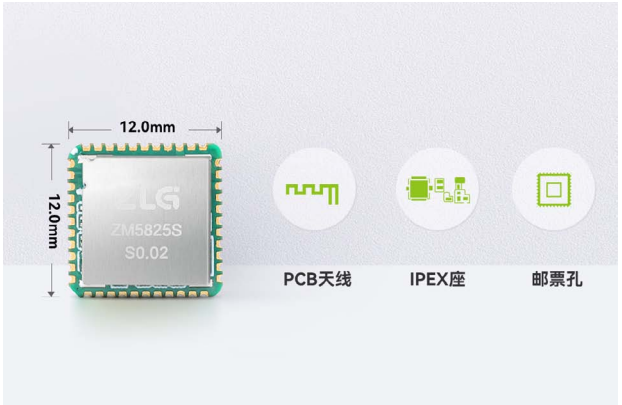
ZLG 致远电子 2023-09-18 11:41:32

选型表

型号	ZM5825P	ZM5825E	ZM5825S
产品类型	Transceiver Wi-Fi		
无线协议	IEEE802.11b/g/n & BLE5.1		
Wi-Fi 工作模式	AP/STA		
通信接口	SDIO @WiFi UART @BLE		
射频输出	PCB 天线	IPEX 连接器	邮票孔
射频输出（mm）	18.0×25.0×2.7	18.0×25.0×2.7	12.0×12.0×2.1

高速稳定通讯，数据传输无忧

高集成紧凑设计



工业级品质

典型应用场景

强大的驱动兼容性

互联互通▼

【产品应用】需要扩展CAN总线接口吗？小Z教你怎么做！

ZLG 致远电子 2023-09-04 11:38:26

您是否遇到需要使用 CAN 总线，但 MCU 没有集成 CAN 总线控制器的情况？您又是否遇到 MCU 的 CAN 口资源不够的情况？一颗 CSM330A 芯片能解决您的问题，本文将带您了解 CSM330A 的优势及应用。

CSM330A的优势

如图 1 所示，通过 SPI 或 UART 扩展隔离 CAN 接口的常规方案，一般要用到 CAN 控制器芯片、隔离 DC-DC 电源、数字隔离芯片、CAN 收发芯片等，存在电路设计复杂、器件种类多、占板面积大等缺点。

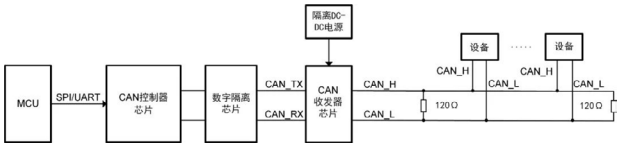


图1 常规CAN接口扩展方案

ZLG 致远电子推出的 CSM330A 可以很方便的嵌入到有 SPI/UART 接口的设备中，在不改变原有硬件结构的前提下使设备获得 CAN-bus 通信接口。CSM330A 的大小仅为 15×10×3mm，体积仅为 CSM300(A) 系列直插封装 SPI/UART 转 CAN 模块的 11%，拇指大小的模块大大减少了占板空间，更加适合嵌入到设备中。如图 2，仅需一片 CSM330A，即可扩展出一路 CAN 接口。

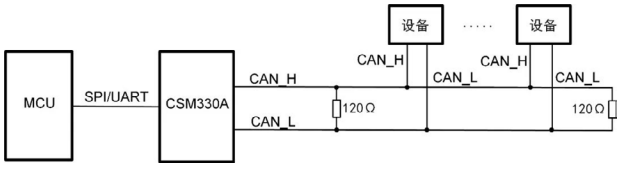


图2 CSM330A扩展CAN接口

CSM330A硬件电路设计

使用 SPI 转 CAN 功能时，需要将 MODE 引脚接至高电平。MCU 的 SPI 接口与 CSM330A 的 SPI 接口连接，同时 MCU 需要提供 GPIO 与 RST、INT、CTL0、CTL1 引脚连接，实现对 CSM330A 的有效监测与控制。若需要通过 MCU 对 CSM330A 进行配置，则需要额外的 GPIO 与 CFG 引脚连接。

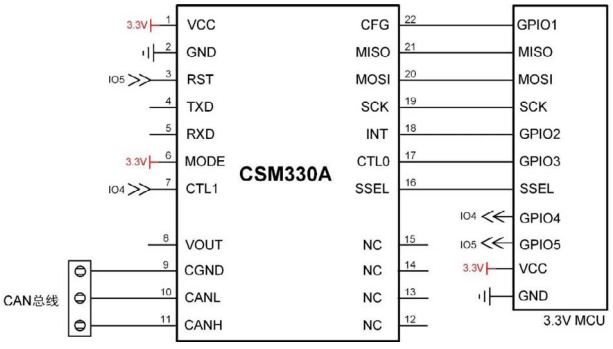


图3 SPI转CAN参考电路

使用 UART 转 CAN 功能时，需要将 MODE 引脚接至低电平。MCU 的 UART 与 CSM330A 的 UART 接口连接，同时一个 GPIO 与 RST 引脚连接。若需要通过 MCU 对 CSM330A 进行配置，则需要额外 GPIO 的与 CFG 引脚连接。

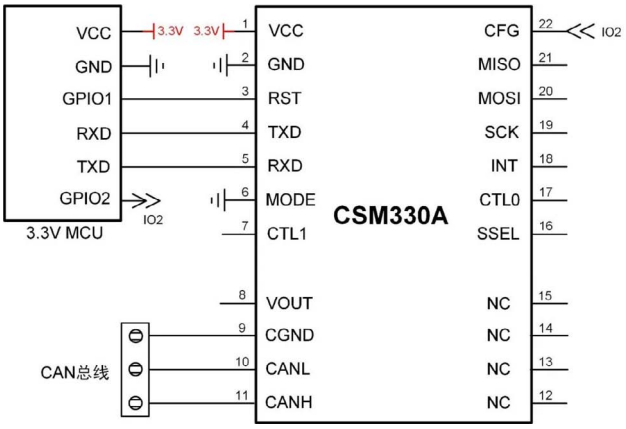


图4 UART转CAN参考电路

行业应用

用户使用 CSM330A 进行 CAN 协议转换，能很方便地在设备中扩展出 CAN 接口，可广泛使用于储能、充电桩、智慧交通、工业控制、轨道交通等领域。

例如 ZLG 致远电子面向 DMS（驾驶员监控系统）行业客户提供的司机行为监测解决方案，其中主控采用了 Cortex®-A35 M1808 AI 核心板。该核心板搭载自主研发的图像传感技术，搭配推出车载行为监测系统解决方案，可实现驾驶员身份信息识别和司机行为监测，以此向车主发出报警提醒，并可通过板载的 WiFi、4G 等模块实现远程人脸录入、管理与监控。

供电方面推荐 ZLG 致远电子的隔离 DC-DC 电源模块 E2405UHADD-10W，该电源模块 9-36V 宽压输入，±5V 稳压输出，能有效降低供电波动导致的设备损坏风险。

M1808 AI 核心板有着两路 CAN FD 和高达 8 路的 UART 接口，如果外接的车载应用较多的情况下核心板的 CAN 口资源不够用，这时就可以通过 CSM330A 来拓展 CAN 接口，其应用如图 5 所示。

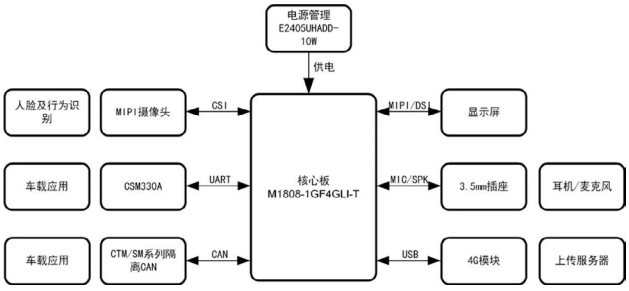


图5 司机行为监测解决方案

互联互通▼

【产品应用】快速了解ePort-G(1.8)的关键设计技巧

ZLG 致远电子 2023-09-05 11:34:50

小巧、集成的 ePort-G(1.8) 是否令你眼前一亮？下面带你快速了解 ePort(1.8) 的关键设计技巧，轻松解决信号线的各种难题

通信接口了解

ePort-G(1.8) 为千兆以太网模块，采用 RGMII 接口（千兆精简介质独立接口）与 MAC 进行数据交互，RGMII 有 12 根数据线，发送信号组和接收信号组各 6 根。发送组信号方向为 MAC 至 ePort-G(1.8)，接收组信号方向为 ePort-G(1.8) 至 MAC。RGMII 相对于 GMII（精简介质独立接口）来说数据线减少了一半，减轻了设计工作。

RGMII 分为发送组信号 TXCLK、TXCTL、TXD[3:0] 和接收组信号 RXCLK、RXCTL、RXD[3:0]，发送信号组的时钟由 MAC 提供，接收信号组时钟由 ePort-G(1.8) 提供，两者的最高频率为 125MHz。除此之外还有用于配置的 MDIO 接口、用于时间同步的 CLKOUT 以及中断 INT 和复位 RESET 引脚。

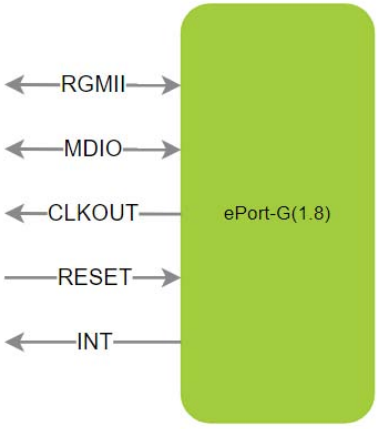


图1 信号示意图

原理图设计

1.ePort-G(1.8) 电源输入是 3.3V。

2.RGMII 的发送信号组和接收信号组的 IO 电平均为 1.8V。发送信号组需要在靠近 MAC 端串联 22Ω 电阻，对信号线进行阻抗匹配，避免信号过冲和反射导致通信失败甚至损坏器件。接收信号组在 ePort-G(1.8) 内部已集成相关匹配电阻，因此可以直接从 ePort-G(1.8) 连接 MAC 端。

3. 配置接口 MDC 和 MDIO 电平为 1.8V，与 MAC 端直连。

4. 复位和中断电平为 3.3V，直连 MAC 端，其中复位引脚是必须的，用于上电后复位。中断引脚为可选。

5.PHY 地址配置引脚与部分 RGMII 引脚复用，可使用 4.7kΩ 电阻对其进行上、下拉处理以配置所需要的地址，在上拉时，注意上拉到 1.8V 而非 3.3V。

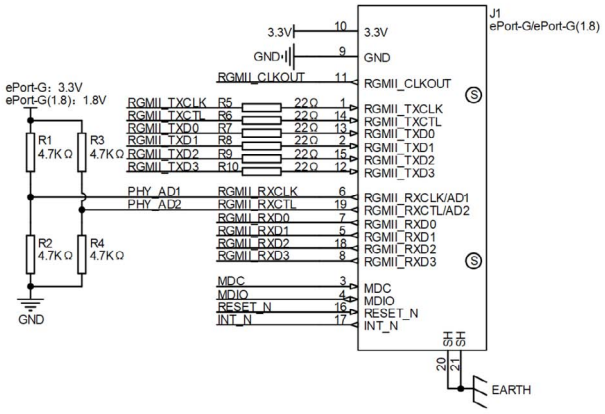


图2 ePort-G(1.8)经典应用

PCB设计

1. 发送信号组串联的 22Ω 电阻需要靠近 MAC 端放置。

2. 发送信号组以及接收信号组都需要以组内时钟信号为基准 120mil 等长处理。

3. 发送信号组、接收信号组、管理接口信号线需要控制 50Ω 单端阻抗。

4.RGMII、MDIO 和复位信号为敏感信号，走线尽可能短，需远离其他高速信号，避免干扰。

5.RGMII 需要有完整的信号参考回流平面。

6.RGMII 信号线间距需要在 3W 以上。

以上是在 PCB 设计时需要注意的几个点，我们成功将 ePort-G(1.8) 应用在 RK3568J 平台上，供大家参考。

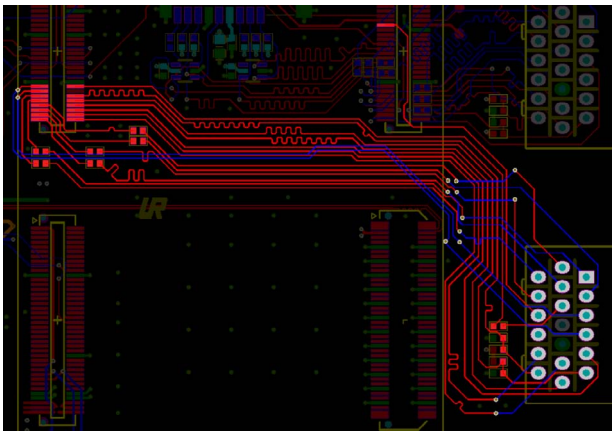


图3 ePort-G(1.8)布线参考

【技术分享】电源模块输出为何振荡?

ZLG 致远电子 2023-09-12 11:37:49

在复杂的工业控制系统中，由于涉及功能繁多，因此需要各种各样的电源进行组合使用。各电源之间往往需要设计合适的电路进行连接，中间电路设计得不合理的话就会导致电路出现问题。本文选择一种情况进行分析说明。

电源连接问题

在做 AC-DC 电源与 DC-DC 电源模块连接时，为了降低电路的启动电流，在 DC-DC 电源模块前端接入了热敏电阻，具体电路如下图 1。结果在进行电路测试时出现了问题，电路在关断的时候 DC-DC 电源模块输出电压出现振荡，无法直接关断，测试波形如图 2。

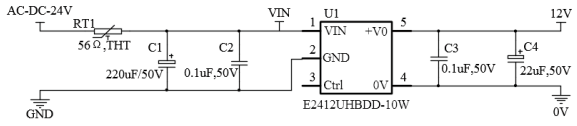


图1 应用电路图

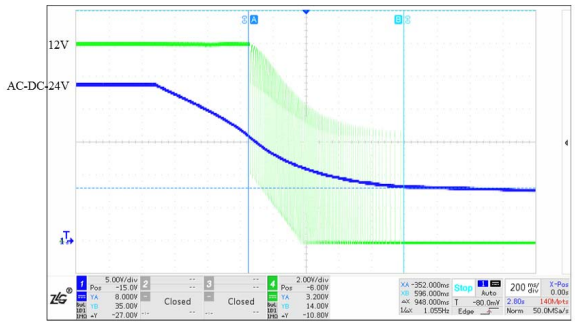


图2 模块关断波形图

问题分析

对图 2 波形进行分析，发现输出振荡波形在 AC-DC-24V 电压下降至低于 8V 时消失了，该电源模块具有输入欠压保护功能，推测可能是模块输入端电压在 8V 左右波动，引发了欠压保护，导致输出端在不断开启和关闭间形成振荡。

对模块输入端电压波形进行检测如图 3，可见在 AC-DC-24V 电压关断下降后，VIN 电压下降至 6.8V 左右时出现了波动，VIN 电压波动区间与输出 12V 电压振荡区间重合。由此可知，模块关闭时输出振荡是由于模块输入端电压波动使得模块反复进入欠压保护状态所导致的。

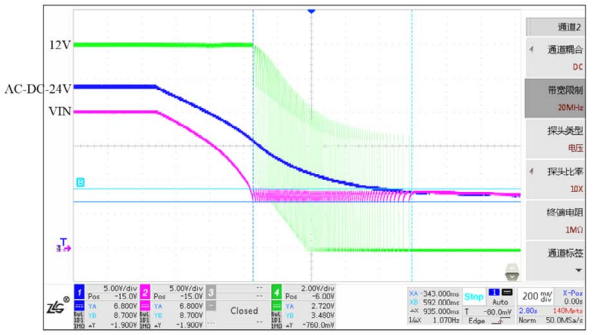


图3 测试波形图

进一步分析图 3 中波形，可以看到 VIN 电压与 AC-DC-24V 电压在输出震荡结束前均存在压差，结合电路图中 VIN 与 AC-DC-24V 之间存在热敏电阻 RT1，可知 RT1 的存在使得电路存在压差。在 AC-DC-24V 关断时，由于压差的存在，VIN 电压先下降至低于 6.8V，此时模块进入欠压保护状态，模块输出关闭输出电压下降。此时 AC-DC-24V 电压仍旧高于 VIN 电压，故 AC-DC-24V 电压会对电容 C1、C2 进行充电，使得 VIN 电压抬高至 8V 以上，模块脱离欠压保护状态，输出重新开启输出电压上升。模块重新启动后会消耗掉电容 C1、C2 提供的能量，使得 VIN 电压再次下降至 6.8V，模块再次进入欠压保护状态。模块重复多次进入和退出欠压保护状态，使得输出电压一直处于振荡中，直至 AC-DC-24V 下降至低于 8V 后，不再进入欠压保护状态，从而结束输出振荡状态。

解决方案及验证

根据分析可知，因为电路中的热敏电阻 RT1 使得电路存在压差，重复触发了欠压保护功能，从而导致输出电压振荡。故可将热敏电阻 RT1 去掉，电路原理图如图 4，测试波形如图 5。

分析图 5 波形，去掉热敏电阻 RT1 后，AC-DC-24V 电压关断下降后，VIN 电压同步下降且不存在压差，模块输出电压没有出现振荡。由此可知通过去热敏电阻 RT1 便能解决此电路输出振荡问题。

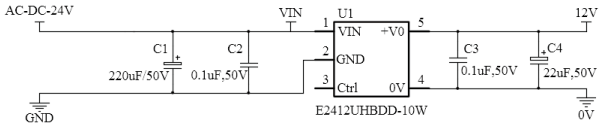


图4 去掉RT1电路

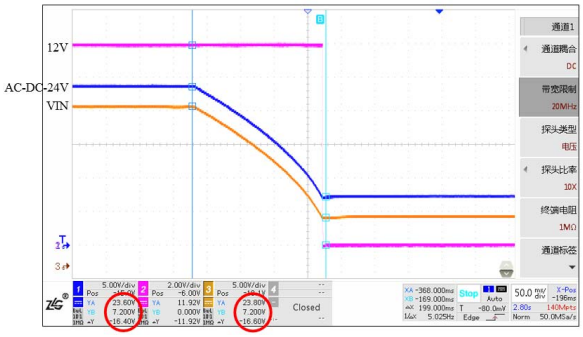


图5 修改电路测试波形图

启动电流的限制

E_UHBDD-6W、E_UHBDD-10W、E_UHBCS-6W 这些产品内部均设计有软启动电路，用于降低模块启动电流，能够有效地限制启动电流尖峰，无需再增加额外的限流电路。

2023/9 第9期

微文摘

ZLG MICRO DIGEST

ZLG致远电子官方微信

广州致远电子股份有限公司
更多详情请访问 www.zlg.cn
欢迎拨打全国服务热线 400-888-4005