

模块供电

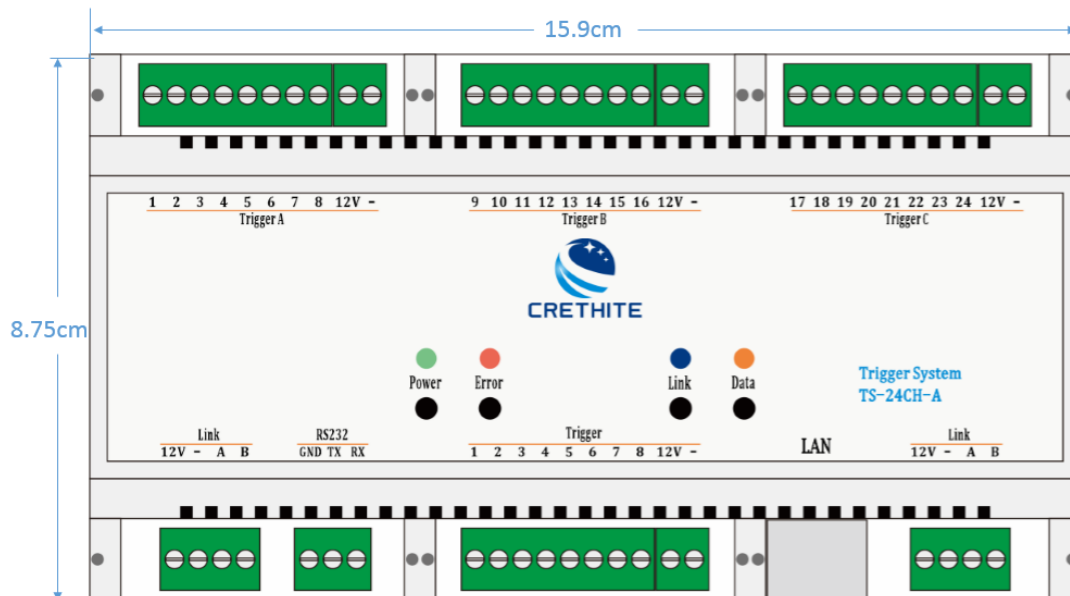
需选用 12V 1A 以上电源

● 安装方式及尺寸

标准导轨式安装

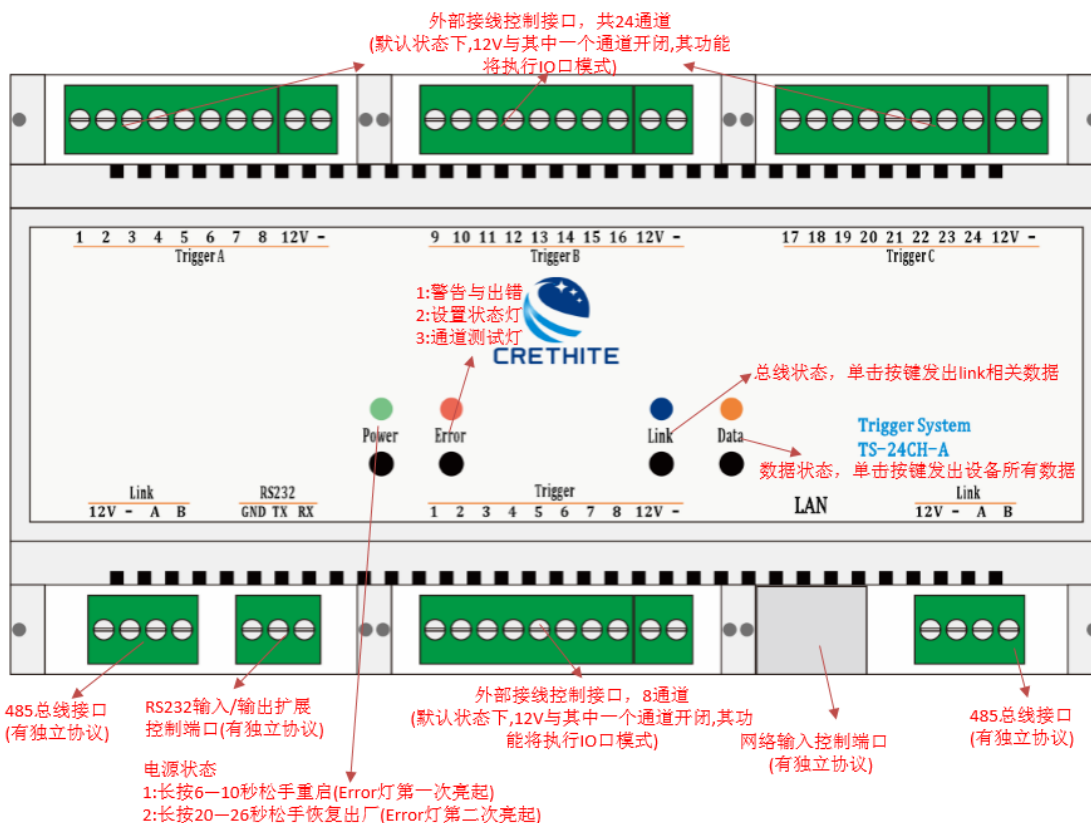
尺寸 15.9cmx8.75cm

外壳 PCV 防火材质



● 型号及功能

1. TS-24CH-A 输入 RS232、LAN*1、感应接口*8,扩展输出为感应接口*24



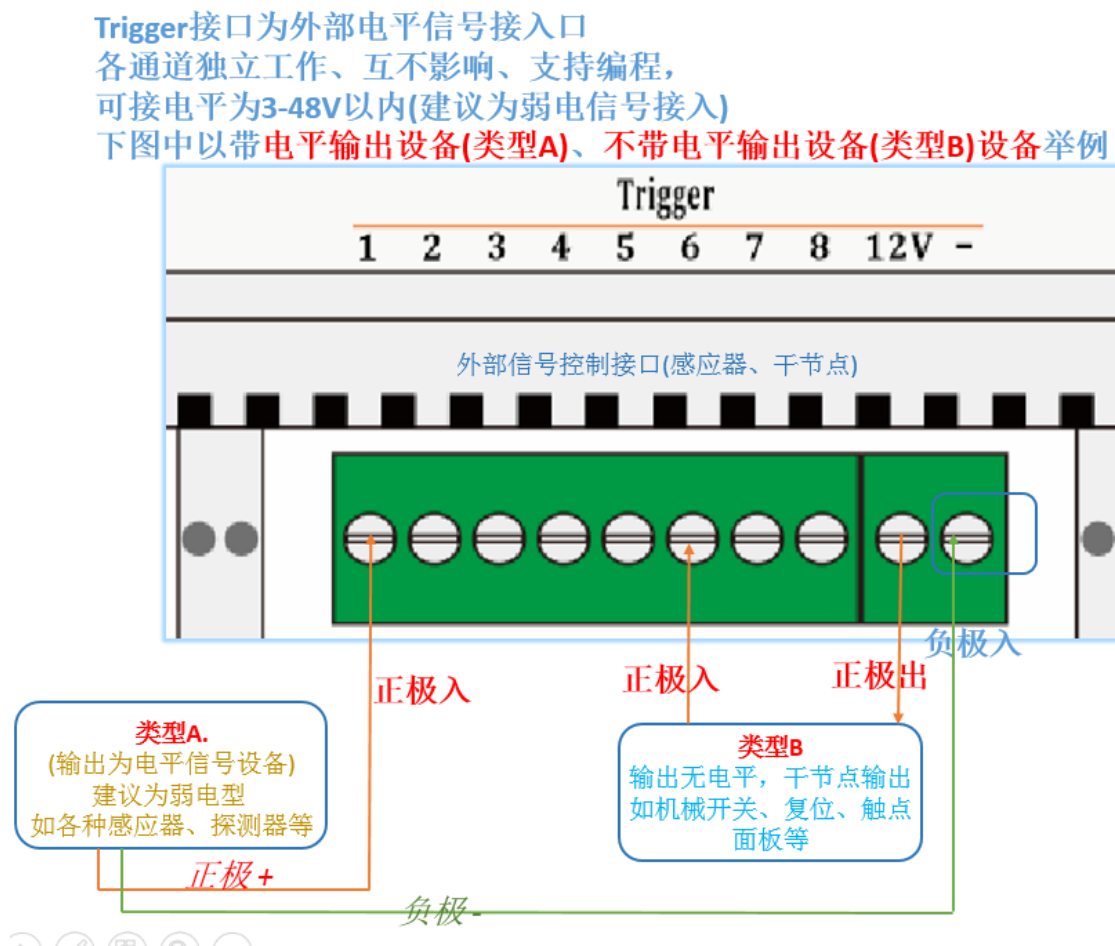
- 接口功能

创思特各模块上的 Trigger 接口功能一致，可检测电平信号为 3-48V，为公地型输入
可接机械开关、触点面板、感应器、等干节点或者电平信号，系统将信号经过处理后，不仅能轻松实现对自身模块功能的调用，还可通过总线发出自定义指令，实现对总线上任意模块的控制，如总线上加入串口服务器等设备，可对 Trigger 接口进行编程实现对串口服务器上设备的直接控制。

- 支持的触发事件

1. 仅按下(接通)
2. 仅松开(断开)
3. 按下(接通)及松开(断开)
4. 长按 3 秒及松开
5. 每按住 1.54 秒及松开

- 接线示例



- 功能示例

复位/机械开关接入 Trigger 简单设置后可实现对灯光控制或者使指令发送至接入总线的第三方中控或受控设备，实现控制目的。

感应器、探测器、消防信号等接入 Trigger 接口可根据需要进行设置，如有消防信号触发，关闭电器设备，打开应急灯。如感应器触发后自动将单路/多路灯光调节至合适亮度。

设置 Trigger 接口协议

● 串口通信

波特率 19200,数据位 8,停止位 1,无校验

1. 485 类型模块接线方式

A 为 + ----- 接控制器 A(+)

B 为 - ----- 接控制器 B(-)

2. 232 类型模块接线方式

TX 为设备往外发送针脚 ----- 接控制器(PC) RX

RX 为设备接收外部命令针脚 ----- 接控制器(PC) TX

- 为公共地 ----- 接控制器(PC) GND

● 控制指令

指令均为 ASCII 码,不区分大小写,且无回车换行。

设备具有超强处理能力,1 秒冷启动。支持连续收码及快速处理,控制无需延时发码

1. 启用或者禁用缓存(如总线上模块较多,建议启用缓存机制,可防止数据冲突)

setSYS(id,5,isOn)

设置设备号为 12 的模块启用缓存机制(设备号为 0 将设置总线上所有模块)

setSYS(12,5,1)

设置设备号为 12 的模块禁用缓存机制(设备号为 0 将设置总线上所有模块)

setSYS(12,5,0)

2. 设置地址指令(如用 0 代替现有地址,需先将该模块连接到其他模块的总线断开,以免误将其他模块地址改变)多台级联需设置从 1 开始的不同地址

setSYS(old_id,1,new_id)

如将设备号为 1 的模块设备号改为 2.

setSYS(1,1,2)

如总线上模块设备号改为 2.(旧设备用 0 代替,但防止误操作请其他模块从总线上断开)

setSYS(0,1,2)

a) 查询设备地址(查看该台模块地址,模块需已禁用缓存机制)

getSYS(0,1)

模块返回信息如下 **msgSYS(0,1,12)** 代表该模块地址为 12

3. 根据模块序列号设置地址 (模块需已禁用缓存机制,按下模块 Link 按钮,模块会将自身信息从总线接口发出)

setSYS(0,7,id,"00122F3CDDE3EB")

如模块发出 SN:"00122F3CDDE3EB" 代表该模块序列号为"00122F3CDDE3EB"

将该模块地址设置为 6 指令为

setSYS(0,7,6,"00122F3CDDE3EB")

4. 设置设备扩展串口波特率及校验信息

setDev(id,10,baud,parity)

id=设备号 默认为 0 *id=0 时将控制所有总线上模块*

baud=为波特率 2400-115200

parity=校验位 0:无校验 1:奇校验 2:偶校验 例

如将所有设备扩展串口波特率设置为 9600,无校验命令为

setDev(0,10,9600,0)

如将所有设备扩展串口波特率设置为 38400,偶校验命令为

setDev(0,10,38400,2)

查询指令 **getDev(id,10)**

5. 通过扩展串口发送 ASICC 格式代码(使用该口已经默认的波特率及校验位)

`setDev(id,1,"DATAS")`

id=设备号 默认为 0

DATAS 为需要发送的数据

例如 2 号设备扩展串口发送数据 Power on

`setDev(2,1,"Power on")`

查询指令 `getDev(id,1)`

6. 通过扩展串口发送 16 进制格式代码(使用该口已经默认的波特率及校验位)

`setDev(id,4," DATAS")`

id=设备号 默认为 0

DATAS 为需要发送的数据

例如 2 号设备扩展串口发送 16 进制数据 706F776572206F6E

`setDev(2,4,"706F776572206F6E")`

查询指令 `getDev(id,4)`

7. 通过扩展串口发送 ASICC 格式代码(临时指定波特率及校验位)

`setDev(id,3, baudID, parity , "DATAS")`

id=设备号 默认为0

baudID =为波特率类型 0 代表默认, 1 为

2400, 2 为 4800, 3 为 9600, 4 为 19200, 5 为

38400, 6 为 57600, 7 为 115200

parity=校验位 0:无校验 1:奇校验 2:偶校验

DATAS 为需要发送的数据

例如 2 号设备扩展串口发送数据 Power on(波特率 19200 无校验)

`setDev(2,3,4,0,"Power on")`

8. 通过扩展串口发送 16 进制格式代码(临时指定波特率及校验位)

`setDev(id,6,baudID , parity , " DATAS")`

id=设备号 默认为0

baudID=为波特率类型 0 代表默认, 1 为

2400, 2 为 4800, 3 为 9600, 4 为 19200, 5 为

38400, 6 为 57600, 7 为 115200

parity=校验位 0:无校验 1:奇校验 2:偶校验

DATAS 为需要发送的数据

例如 2 号设备扩展串口发送 16 进制数据 A9172E0000003F9A (波特率 38400 偶校验)

`setDev(2,6,5,2,"A9172E0000003F9A")`

9. 设置扩展串口返回数据的接收方式

`setDev(id,11,comReturnType)`

id=设备号 默认为 0

comReturnType

==0: 纯数据, 无格式

==1: ASCII 码, 有格式, 但未转义

==2: ASCII 码, 有格式, 已转义

==3: 十六进制码, 有格式

==4: 禁止反馈

查询指令 `getDev(id,11)`

10. 启用 Trigger 接口反馈系统代码

`setDev(id,30,isOn)`

如设置设备号为 12 的模块仅主串口反馈系统代码功能

`setDev(12,30,1)`

如设置设备号为 12 的模块仅扩展串口反馈系统代码功能

`setDev(12,30,2)`

如设置设备号为 12 的模块主/扩展串口均启用反馈系统代码功能

`setDev(12,30,3)`

如设置设备号为 12 的模块禁用系统代码功能

`setDev(12,30,0)`

查询命令 `getDev(id,30)`

11. 启用 Trigger 接口动作类型 (该协议适用于 24 通道感应器及触摸面板)

`setDev(id,25,ChGroup,CH,clickType)`

`id`=设备号 默认为 0 *id=0 时将控制所有总线上模块*

`ChGroup` 为接口组号

`==1` 为 A 组

`==2` 为 B 组

`==3` 为 C 组

`==4` 为主 IO

`CH`=需要控制的通道 *CH=0 时为全部通道控制*

`clickType` =1 为复位模式, 每按一次循环执行(默认)

`clickType` =2 为自锁模式, 按下、松开两种事件都执行

`clickType` =3 仅执行按下事件

`clickType` =4 代表禁用按下事件, 启用长按事件, 按下 3 秒后, 发出按下代码, 松开时执行对应指令

`clickType` =5 代表禁用按下事件, 启用长按事件, 每按住 1.5 秒重复发出按下代码, 松开时执行对应指令

例如将 12 号设备 A 组 Trigger 2 口设置为类型 3(只执行按下事件)

`setDev(12,25,1,2,3)`

查询命令 `getDev(id,25,chGroup,CH)`

12. 设置设备模式存储 ASICC 格式存档 (该协议适用于 24 通道感应器模块及面板)

`setDev(id,22,modelId,sendTpye,"Command")`

`modelId` 为需要存储的模式编号

`sendTpye` =1 为仅主串口发送

`sendTpye` =2 为仅扩展串口发送

`sendTpye` =3 为主/扩展串口都发送

`Command` 为 ASICC 格式的指令, 最多为 64 个字符

例如将设备号为 12 的模块模式 2 设置为主串口发送指令 `push:1`

`setDev(12,22,2,1,"push:1")`

例如将设备号为 12 的模块模式 1 设置为主/扩展串口同时发送指令 `push:1`

`setDev(12,22,1,3,"push:1")`

查询命令 `getDev(id,22,modelId)`

13. 设置设备模式存储 16 进制格式存档 (该协议适用于 24 通道感应器模块及面板)

`setDev(id,23,modelId,sendTpye,"Command")`

`modelId` 为需要存储的模式编号

`sendTpye` =1 为仅主串口发送

`sendTpye` =2 为仅扩展串口发送

`sendTpye` =3 为主/扩展串口都发送

Command 为 16 进制格式的指令, 最多为 64 个字符

例如将设备号为 12 的模块模式 2 设置为主串口发送指令 31585948

`setDev(12,23,2,1," 31585948")`

例如将设备号为 12 的模块模式 1 设置为主/扩展串口同时发送指令 31585948

`setDev(12,23,1,3," 31585948")`

查询命令 `getDev(id,23,modelId)`

14. 调用设备单个模式

`setDev(id,24,mode)`

mode 编号为 1-80

例如调用 12 号设备第 2 场景

`setDev (12,24,2)`

15. 调用设备多个模式

`setDev(id,28,{modeArr})`

modeArr 为模式编号组, 最大长度为 16, 多模式用","逗号隔开

例如调用 12 号设备第 2、3、4、5、6、7、10 场景

`setDev (12,28,{2,3,4,5,6,7,10})`

16. 设置 Trigger 接口触发后需调用的模式

`set_RS(id,27, ChGroup,ch, clickType,{ModeArr})`

id=设备号 默认为 0 id=0 时将控制所有总线上模块

ChGroup 为接口组号

==1 为 A 组

==2 为 B 组

==3 为 C 组

==4 为主 IO

Ch 为 1-8 即各 Trigger 组内 8 个通道

clickType =0 代表设置按下与松开模式

clickType =1 代表仅设置按下模式

clickType =2 代表仅设置松开模式

ModeArr 为需调用的模式组(1-8 个, 多个用","逗号隔开)

如设置 12 号设备 Trigger B 组 3 口设置按下调用模式 8

`set_RS(12,27,2,3,1,{8})`

如设置 12 号设备 Trigger B 组 3 口设置按下调用模式 1,3,5,7

`set_RS(12,27,2,3,1,{1,3,5,7})`

查询命令 `get_RS(id,27,ch,clickType)`

17. 通过指令调用 Trigger 接口事件

`setDev(id,29,ChGroup,CH, clickType)`

id=设备号 默认为 0 id=0 时将控制所有总线上模块

ChGroup 为接口组号

==1 为 A 组

==2 为 B 组

==3 为 C 组

==4 为主 IO

CH=需要调用的通道 CH=0 时为全部通道

clickType =1 代表需调用按下事件对应模式

clickType =2 代表需调用松开事件对应模式

例如 1 号地址模块调用 Trigger B 组 3 口按下事件, 将已存档的命令发出;

`setDev(1,29,2,3,1)`

更多详细说明请上官方网站查阅 www.crethite.com

淘宝企业店铺(经销商折扣请联系客服)

<https://shop155265173.taobao.com/?spm=2013.1.0.0.x1GetC>

关注公众号
可直接购买

