



SimFAS 深克斯中控系统

语法函数编程手册

中控系统-默认引擎 Sim V7.2

建议使用代码助手功能,您只需填参数,AI 智能生成代码

内部文档,仅供合作伙伴使用,请勿公开.

上海深克斯科技有限公司 Copyright@2008-2023

SimFAS 语法函数编程手册

<适用于中控默认引擎 v7.0+ 以上>

- 该模式适合基础用户使用, 具备自动容错功能(出现语法参数错误会尝试修复并继续运行)
- 如果您有编程基础, 但该引擎还不能满足您的逻辑需求, 请使用专家模式.
- 双减号是代表备注, 双减号后面的当前行内容不执行, 字符串用双引号引起来.

-----串口发送数据-----

```
com1.set(9600,8,"n",1); --设置 com1 波特率 --串口支持 com1-com8,  
com1.send("hello"); --com1 发送 ASCII 数据  
com1.sendhex("00 01 02 03"); --com1 发送 16 进数据 (大部分控制指令用这个函数)
```

-----网络应用 TCP/UDP/http-----

```
--TCP: 向设备 192.168.2.100 的端口 8800 发送 tcp 数据 hello ,发完后立刻断开  
tcp.send("192.168.2.100",6688,"hello");  
--发送 16 进 tcp 数据(其它参数参照上面)  
tcp.sendhex("192.168.2.100",6688,"01 0B 00 0D 0A");  
--带超时参数,连上后等待 1000ms 后发送 tcp 数据并等 1000ms 后再断开,连接 2 秒超时.  
ret=tcp.send("192.168.2.100",6688,"hello",1000,2); --ret 记录成功发送的字节数.  
res=tcp.sendrcv("192.168.2.100",6688,"chk"); --发送并接收,res 记录服务器返回的数据.  
res=tcp.sendhexrcv("192.168.2.100",6688,"010203"); --发送 hex 并接收,res 是返回的数据.
```

```
--UDP: 向设备 192.168.2.100 的端口 6699 发送 udp 数据 hello ,发完后立刻断开  
udp.send("192.168.2.100",6699,"hello");  
--发送 16 进 udp 数据(其它参数参照上面)  
udp.sendhex("192.168.2.100",6699,"01 0B 00 0D 0A");
```

```
--HTTP 客户端 get 和 post 交换数据  
http.get("http://www.simfas.com/get/");  
http.post("http://www.simfas.com/post/","a=1;b=2");  
--注意: 如果表单或者 url 含有中文空格等参数,可能需要使用 str.EncodeURL 函数进行编码  
Req=str.EncodeURL("hello=你好");  
url=str.cat("http://www.simfas.com/get/?", Req);  
res=http.get(url);
```

-----电脑控制-----

```
--网络唤醒网段内的 MAC 为 01:03:0A:F2:DD:08 的计算机(注意,IP 地址是广播地址)  
pc.wakeup("192.168.2.255","01:03:0A:F2:DD:08"); --因为电脑关机了没有 IP 了.  
pc.shutdown("001"); --关闭 SimCTRL 软件 ID 为 001 的电脑  
pc.shutdown("*","S","192.168.2.101"); --关闭电脑 IP 为 192.168.2.101,SimCTRL  
pc.reboot("001"); --重启 SimCTRL 软件 ID 为 001 的电脑  
pc.SendKey("001","31"); --发送按键 (第三个参数为 IP 地址, 可选填)  
pc.open("001","d:\\abc.txt"); --打开文件(第三个参数为 IP 地址, 可选填)  
pc.close("001","abc.exe"); --关闭软件(任务管理器找到 exe 的全名)
```

-----红外控制-----

`ir1.send("按键 A");` --把红外[按键 A]通过通道 1 发送出去 ,其中[按键 A]是通过红外学习进去。
--红外支持 ir1-ir8 ,
--红外学习在 我的程序—红外学习 里面,使用前,需先把遥控器功能学习到中控里面。

-----变量-----

--局部变量规则:字母开头+字母(数字) (当前程序访问,程序执行完后不再保留)
--全局变量规则:_开头+字母(所有程序可访问, 程序执行完后值会保留)
--长度不大于 8 个字符

`a=10` --整数赋值
`b=-5` --负数赋值 (负号必须紧跟数字, 中间不要有空格)
`c=0xff` --16 进赋值
`_abc=1;` --全局变量,下划线_开始
`ip="192.168.2.194"` --字符串赋值
`str="Hello SimFAS"` --字符串赋值,用双引号引起来。
`port=8899 ;` --整数赋值
`port2=port;` --变量赋值给变量

-----系统常用函数-----

`sleep(1);` ---延时 1 秒
`msleep(500);` ---延时 500 毫秒
`usleep(100);` ---延时 500 微秒
`beep(10);` ---蜂鸣器响一下
`print("IP 地址:",ip);` --打印调试信息(所有参数会合并起来显示)
`printhex(data);` --打印 hex 调试信息(data 会以 16 进制显示)
--开新的线程立刻运行 `abc.tsk` 任务 (不等 `abc.tsk` 运行完毕,立刻运行后面的代码)
`call("abc.tsk");`
--在当前线程运行 `abc.tsk` 任务 (运行完 `abc.tsk` 任务后,继续运行后面的代码)
`call_in("abc.tsk");`
`time.Limit(1000);` --如果中控开机超过 1000 小时,那么后面的代码不执行。
`exitIf("程序已经运行");` --如果当前程序已经运行了,那么后面的代码不执行。
`break();` ---退出当前循环,如果没有循环,那么退出当前程序, ,那么后面的代码不执行。

-----触摸屏指令-----

--该指令用于通过中控去修改所有触摸屏平板界面上的某个对象的文字,透明度等.

--把平板 SimAPP 上,名称为 `abc` 的按钮,文字改成 系统已打开 ,透明度改成 80
`touch.send("abc","系统已打开",80);`
--先保存起来,等调用 `touch.send` 再一起发,用于一次性更新多个 UI
`touch.sendPre("abc","系统已打开",80);`
`touch.SendOnce("tips","灯光已打开",80);` --只显示一次,切换页面后,该内容不会被保存
`touch.remove("abc");` --删除 `abc` 的内容。
`touch.disable("abc") ;` --禁用触摸屏平板的"abc"按钮
`touch.enable("abc") ;` --启用触摸屏平板的"abc"按钮
`touch.open("next.ui") ;` --触摸屏平板切换页面到"next.ui"
`touch.speak("你好,系统正在打开") ;` --向触摸屏平板发送语音提示

-----投影 Pjlink-----

```

pjlink.on(ip,pass);          --根据 IP 和密码打开投影机
pjlink.off(ip,pass);         --根据 IP 和密码关闭投影机

--通用接口函数,data 可以根据 pjlink 协议自行组合(支持 class1/class2/等)
pjlink.send(ip,port,data,pass);
--根据 IP 和密码切换输入通道,ch 是字符型 "11" ...."2Z"等具体参照详细列表
pjlink.input(ip,ch,pass);
res=pjlink.get(ip,id,pass);   --获取投影机状态, id 参照列表,res 是投影返回的数据.
pjlink.set(ip,id,pass);       --设置投影机, id 参照列表

```

id 值	说明	ch 值	说明	说明
2	查询电源状态	11	切换通道到: RGB 1	1 代表 RGB,1 代表通道
3	查询投影机名称	1Z	切换通道到: RGB Z	通道支持 1,2,3...A...Z
4	查询投影机型号	21	切换通道到: VID 1	2 代表 VID,1 代表通道
5	查询投影机制造商	2Z	切换通道到: VID Z	通道支持 1,2,3...A...Z
6	查询输入通道	31	切换通道到: DIG 1	3 代表 DIG,1 代表通道
7	查询状态	3Z	切换通道到: DIG Z	通道支持 1,2,3...A...Z
8	查询序列号	41	切换通道到: STR 1	3 代表 STR,1 代表通道
9	查询分辨率	4Z	切换通道到: STR Z	通道支持 1,2,3...A...Z
10	设置 FREZ ON	51	切换通道到: NET 1	3 代表 STR,1 代表通道
11	设置 FREZ OFF	5Z	切换通道到: NET Z	通道支持 1,2,3...A...Z
12	设置 SVOL +			
13	设置 SVOL -			
14	设置 MVOL +		列表未列出的指令,您可以查看 pjlink 协议,使用 pjlink.send 指令操作,或联系我们.	
15	设置 MVOL -	注:		

-----逻辑判断 if (更多例子见附录)-----

--逻辑判断: >(大于) <(小于) =(等于) ~(不等于)

-- 逻辑表达式需要用括号概起来,逻辑里面支持变量判断,不支持直接函数判断.

-- 不支持 if 内嵌 if, 不支持 or and 等(如需完整功能,请使用专家模式)

```

a=10;
if (a>0)
    print("a 大于 0");
end
--没有比较符,该变量不为 0,那么就成立
if (a)
    print("a 不等于 0");
end

```

-----数学运算-----

```

a++; --a 加 1
a--; --a 减 1
a!!; --a 取反 (结果是一次 0,下一次 1,如此反复)

```

-----while 循环-----

--逻辑判断: >(大于) <(小于) =(等于) ~(不等于)
-- 不支持 if 内嵌 if,
-- 不支持 if 内嵌 while
-- 逻辑表达式需要用括号括起来
-- 不支持 or and
-- 支持 while 内嵌 if else
-- 支持 break(); 退出循环,注意 break();要带括号.

```
while (a<10);  
  
    if (a>5)  
        print("a 大于 5 的值是:",a);  
    else  
        print(a);  
    end  
  
    a++;  
  
end
```

-----数学函数-----

```
ret=math.add (1,2,5); --ret 的结果是: 1+2+5  
ret=math.take (a,b) ; -- ret 的结果是:a-b  
ret=math.times(a,b) ; -- ret 的结果是:a*b  
ret=math.div (a,b) ; -- ret 的结果是:a/b  
ret=math.divf (a,b) ; -- ret 的结果是:a/b <小数>  
ret=math.max (a,b,c); -- ret 的结果是:a,b,c 的 最大数  
ret=math.min (a,b,c); -- ret 的结果是:a,b,c 的最小数  
ret=math.abs (-10); -- ret 的结果是-10 的绝对值:10
```

-----网络应用 tcp (高级) -----

```
--创建 tcp 客户端  
sock=tcp.new();  
--连接 tcp 服务器,没连接成功 0+1 秒后超时  
ret=tcp.connect(sock, "192.168.2.194",7788,0);  
if (ret)  
    --发送数据  
    si=tcp.sendto(sock,"hello kitty");  
    --接收数据, 3000ms 后没有数据退出  
    res=tcp.recv(sock,3000);  
end  
--关闭 tcp 客户端  
tcp.close(sock);
```

-----字符函数-----

```
bool=str.cmp(result,"OK");      --两个字符参数是否一样
bool=str.contain(result,"OK");  --result 是否包含 "OK"
res=str.new(64);                 --res 申请 64 个字节长度
res=str.cat(result," : ", "Great"); --把所有参数都连起来,保存到 res
ret=str.len(res);                --获取 res 字符串的长度
res=str.tochar("31 32 33 34 35"); --把 16 进制字符串或整数转成字符串(char)
res=str.getChar("abcde",4);      --获取字符串/整数的第 n 个 char,返回 char(左 1 为第一位)
res=str.sub(ip,2,4);             --截取字符串 ip 的第 2 位开始,截取 4 个字节
res=str.md5(ip);                 --把 IP 字符串 MD5 加密
res=str.sprintf("ip=%s,port=%d,md5=%s",ip,port,res); ---格式化字符串.(最多 3 个参数)
res=str.Before(res,"port")      --截取 res 中字符串",port"前的内容
res=str.After(res,"port")       --截取 res 中字符串",port"后的内容
url=str.EncodeURL("abc?=中文"); --把 http 需要提交的数据 url 编码.
res=str.indexof(ip, "9");        --获取字符 9 在 ip 中的位置,返回 (int)
res=str.getBit(0xf0,4);          --获取整数 0xf0 第 4 位的值,返回(0/1),共 32 位,右 1 为第 1 位
res=str.setChar(data,2,0xf1);    --设置字符串 data 的第二位为 0xf1
res=modbus.new(cmd,id,addr,val); --创建 modbus RTU 的协议数据,
--创建好 Modbus 的数据后,用 com1.send(res); 发送生成的 Modbus 数据,
-- cmd 代表指令(1-5 是读, 6 是写), id 是 RS485 设备 ID, addr 是寄存器地址, val 是要写入的数据.
```

-----系统函数-----

```
ret=sys.uptime(); --系统启动时间,单位 s
res=sys.now();     --系统当前时间
log.save("用户执行了系统总开"); --用户记录操作日志
relay.set(ch,val); --控制中控自带的 8 路继电器,其中 ch 是通道,范围是 1-8, val 是开关值(0 或 1)
return();         --退出当前程序
printhex(data); --打印变量 data 的 16 进数据.
```

-----校验算法等函数-----

```
ret=crc16.toint(data); --计算 data 的 crc16 的值,结果返回整数 (CRC16 为 Modbus 高低位结果)
res= crc16.tochar(data);--计算 data 的 crc16 的值,结果返回字符,2 个字节的字符串.
res=str.md5(data);     --把 data 字符串 MD5 加密,结果返回给 res
```

-----更多说明-----

基础模式中,所有的功能基本是一行完成,无需创建,无需释放等.

如果您需要更复杂的功能,请使用 专家模式,

专家模式拥有完整的功能,能实现 c 语言,C++,java 等语言的全部逻辑.

-----网络应用 tcp 客户端,长连接,带错误处理 -----

```
--创建 tcp 客户端
sock=tcp.new();
--连接 tcp 服务器,没连接成功 2 秒后超时
ret=tcp.connect(sock, "192.168.2.194",7788,2);
if (ret)
    si=tcp.sendto(sock,"hello kitty"); --发送数据
end
---循环接收数据
while (ret)
    --接收数据, 3000ms 后没有数据退出
    res=tcp.recv(sock,3000);
    if (res="$Close$") --这个是系统返回的错误指令,说明和服务端断开了 .
        print("无法连接到服务器!");
        break(); --退出循环
    end
    ----收到服务器数据
    if (res)
        print("收到数据:",res);
    end
end
--关闭 tcp 客户端
tcp.close(sock);
print("tcp 客户端退出");
```

-----逻辑判断例子-----

```
a=10;
if (a>0)
    print("a 大于 0");
else
    print("a 小于 0");
end

s="hello"
if (s="hello")
    print("字符串相等");
end

c=str.contain(s,"llo");
if (c) print("字符串包含 llo"); end
```

常用功能说明

-----限制用户使用时间 time.limit -----

该函数用于限制程序使用的时间, 只需要把 `time.limit()` 这个函数放在你要限制的程序的第一行就可以了, 后面的参数是中控的总通电时间, 该时间可以在 中控的管理页面的一系统状态 `time.Limit(1000)`; --如果中控开机超过 1000 小时, 那么后面的代码不执行

--您的代码.



-----调用其它程序 call / call_in -----

这两个函数可以在当前程序中调用其它程序, 只需要通过程序的控制代码进行调用, 区别是: Call 是打开新的线程运行要调用的程序, 然后立刻运行当前程序后面的代码.

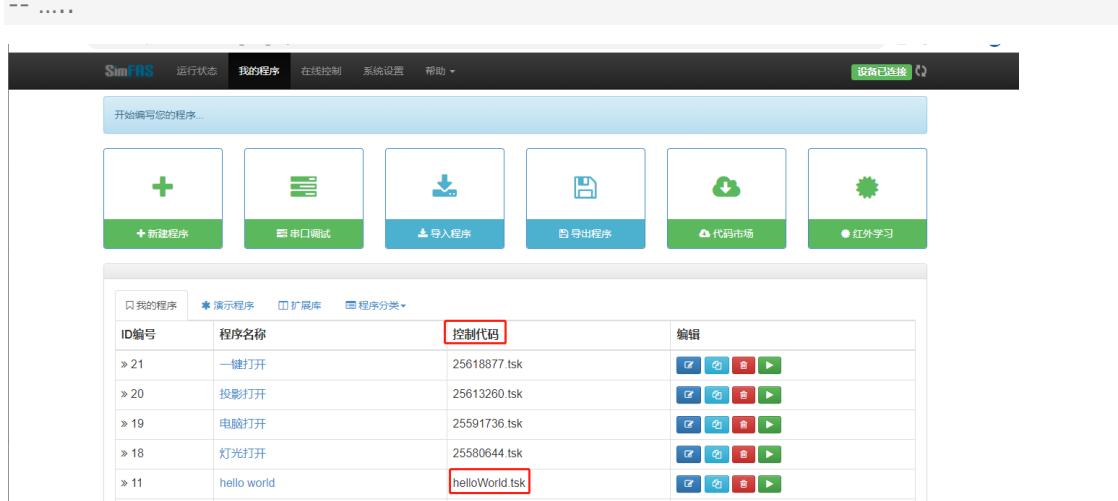
Call_in 是在当前程序中运行要调用的程序, 调用的程序运行完毕后, 继续运行程序后面的代码.

--您的代码.

`call("helloWorld.tsk");`

--您的代码.

`call_in("helloWorld.tsk");`



-----调试用途 beep / print -----

这两个函数可以用于您调试程序, beep 是让中控响一下, print 是打印调试信息

--您的代码.

`beep(10);`

--您的代码.

`print("程序已经运行");`

--

记录用户操作日志

中控可以记录用户的操作日志, 通过程序指令 `log.save` 进行记录.具体用法是在您要记录的程序最后一行,加上该指令,`log.save("日志提示信息");` 这样,每次用户执行了该程序,都会在系统生成一条带时间的日志,注意:系统的时间需要中控 WAN 口连接外网,中控自动更新.



查看用户操作日记

管理人员可以通过中控的管理界面---我的程序---帮助---用户日志 Log 查询



注意:用户日志是滚动保存的,最早的记录会被覆盖掉,日志总数默认是 128KB-256KB 动态保存,, 用户可以根据需要使用,合计可以保存日志记录在 5000-16000 条左右的记录.

特别注意,该文档只适用 V7.0+的中控系统



中控程序调用和带参数调用

中控和中控之间可以相互调用程序,

其它程序,包括 SimAPP 等也可以调用中控程序.

其中 SimAPP 是通过 tcp 和中控通讯的,其中按钮的动作,

就是要发送给中控的调用程序指令,结构就是:(中控程序的) 控制代码.task?参数

如下图,其中参数会直接被中控直接运行

例如,调用中控时候的参数是: 77206615.ts?a=1&b=20 ,?后的红色部分参数会被当做中控代码直接运行,这个参数可用于一个程序实现多个功能或者矩阵切换等,配合 if 使用,例如:

```
If (a=1)  print("执行 a=1 的内容");  end
```

```
If (a=2)   print("执行 a=2 的内容");  end
```

程序引擎	默认模式
控制代码	77206615.tsk
定时控制	格式: 2014-01-23 * 12:00:01
语音控制	格式: [关键字1][关键字2][开电视][打开电视]
程序分类	默认
程序图标	Default.png

Restore

更多选项 0

保存 保存并关闭

强烈建议您使用代码助手功能

自动生成代码,您只需要填写参数,无需记忆函数,实时调试.

如您有疑问，请联系我们：

提供实时免费的各种服务



编程协助



APP制作协助



调试协助



远程Debug



177-0180-5710



simfas



cs@simfas.com



上海深克斯科技有限公司

<http://www.simfas.cn/>