

3.0 内核使用说明

V1.3



信科电子

电话：0335-7096100

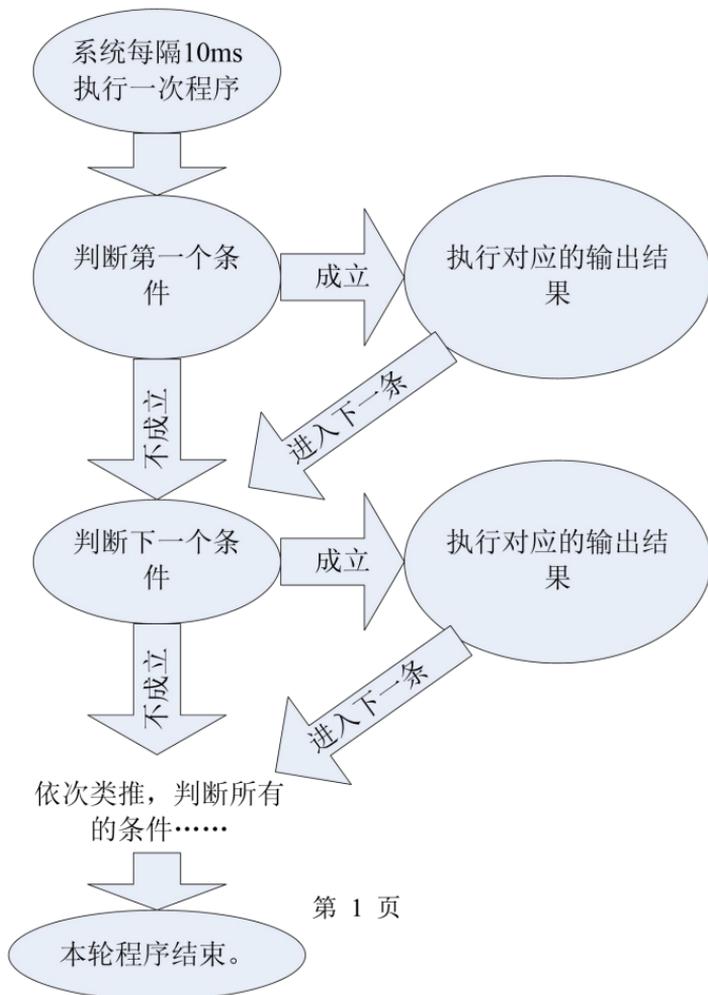
网址：www.qhdxk.com

让控制随心所欲



一、内核简介

用户使用内核来操作模块的各种功能，例如输入与输出的映射关系，或者收发字符串或者其他数据，使用定时器做延迟或者其他定时处理，内部变量的格式转换，等等。内核运行流程：





一、外部输入

外部输入：检测外部输入事件，不同的模块有不同的外部输入的数量，可以选择上升沿触发（断开到闭合）或者下降沿触发（闭合到断开），或者两者都触发，也可以选择状态触发以及单击或者双击触发。

选择状态触发的情况下，有可能这个条件会一直成立。

选择单击或者双击触发，以及在一定的时间内按下了多少次触发注意事项，系统特殊参数里面可以设置时间常数（默认 500ms）。

二、输出状态变化

对于指令操作面板，输出指的是输出 led 指示灯。对于继电器模块，输出指的是继电器的状态。对于 mos 输出的产品，输出指的是的 mos 管的开关状态。

用户可以打开，关闭，翻转某一路或者某几路。也可以点动某一路或者全部打开或者全部关闭。

三、中间继电器

中间继电器是虚拟的继电器，程序可以打开或者关闭，也可以的检测中间继电器的状态。还可以检测中间继电器的上升沿或者下降沿。



典型使用中根据外部的状态设置中间继电器的状态，然后根据中间继电器的边沿变化将外部的连续事件转化为内部的状态变化事件。

四、内部变量

内部的有符号变量，用来记录中间的数据。内部变量可以作为输入，也可以作为输出。

内部变量作为输入使用，变量与某一个数字进行比较得到一个结果，这个结果可以作为条件使用。变量比较的方式可以是大于小于或者等于，数字可以是固定的程序里设置的数字，也可以是某一个变量的值。

内部变量作为输出使用，内部变量作为输出使用是一个很重要的用法。可以将定时器，时间，日期，继电器状态，输入状态，本机地址，模拟量输入的值，环境监测的值等等各种各样的信息赋值为内部变量，然后再使用内部变量统一作出判断。内部变量还可以进行简单的运算，方便作为计数器使用。比如外部输入每闭合一次，内部变量加 1，这样，内部变量就可以记录外部输入闭合的次数。

五、定时器

定时器简介：定时器的值只有正整数，默认属于非 0 减 1 定时器。程序每执行完毕一次检查一次



定时器的值并做非 0 减 1 操作（如果不是 0，自动减一）。

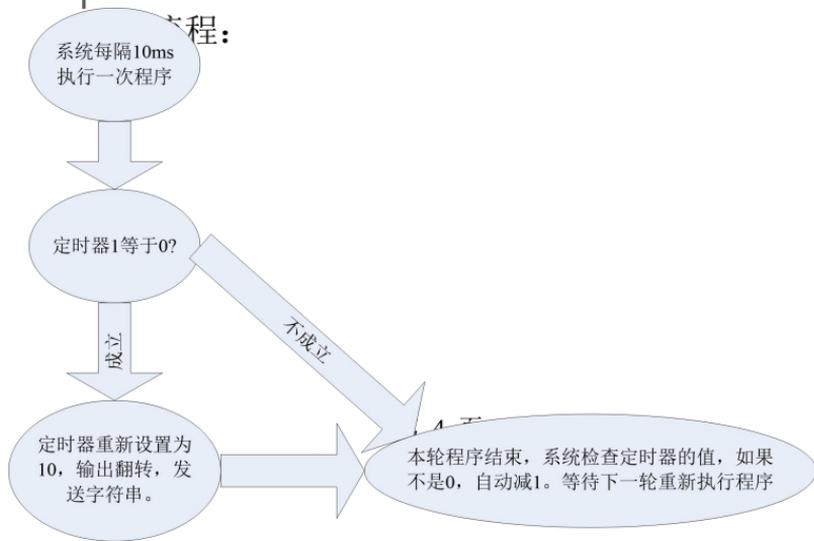
定时器用法：添加输出结果，类型选定时器，将定时器设置为等于某一个值（给定时器设置初始值）。然后添加输入，类型也选定时器，定时器等于某一值触发其他输出。例如输出里面将定时器等于 100（设置初值 100），然后使用定时器等于 80 触发另一个输出结果，设置定时器操作和定时器触发其他动作操作时间间隔就是 20（200 毫秒）。

举例说明：

以下程序每隔 100 毫秒第一路输出翻转一次，并且数据口 1 向外发送字符串：“123”。

```
1 如果 定时器1 等于0  
-> 定时器1 等于 10  
-> 输出操作 翻转第1路  
-> 数据口1 UART1 发送字符串：123
```

流程图：





六、函数使用说明

函数说明：模块包含 100 个函数，从函数 1 到函数 100，函数列表在设置菜单里调出。

函数的使用方法：输出选项里面，可以选择“函数调用”选项，参数 1 选择**函数调用**，参数 2 选择某一个函数，然后确定。

函数的意义：程序易读性，同一个功能使用一个函数替代，主程序中只是使用一个函数代替，可以大大增强主程序的易读性。避免了主程序过长让人眼花缭乱。

多级判断：如果遇到一个条件满足的情况下再去判断另一个条件，或者在一个条件满足了以后再去做一些其他的判断处理，就会用到函数功能。

节省空间：同一个功能使用同一个函数模块，可以反复调用，从而避免同一个功能代码重复。

七、函数使用方法

函数内容：通过菜单-设置-函数列表，启动函数编辑器。函数可以单独存储为函数文件，函数文件的后缀名为.hanshu。函数文件可以被导入到主程序界面，也可以导入到函数界面。

函数调用示例：

```
1 如果 第1路输入 断开到闭合  
-> 函数调用 函数1
```



如上图程序，第一路输入断开到闭合的瞬间，函数**执行一次**。

```
1 如果 定时器1 等于0
   -> 定时器1 等于 100
   -> 函数调用 函数1
```

如上图程序，函数每隔一秒钟执行一次。

函数参数：

函数调用：选择要执行的函数，需要确保函数列表里面有这个函数，否则不执行。

跳转到函数开始：在函数里面执行这个指令可以重新执行函数。

函数返回：在函数里面执行这个函数可以放弃执行函数里面本条命令后的命令。

八、系统简单设置

操作联机继电器数据口(默认数据口1)：用户在使用“操作联机继电器模块(发送命令)”这个功能时，模块通过向总线发送命令的方式实现，此设置定义里使用哪个数据口来发送命令，默认是用数据口1。（不同的模块有不同数量的数据口）

操作联机继电器返回码(默认无返回码)：用户在使用“操作联机继电器模块(发送命令)”这个功能时，模块通过向总线发送命令的方式实现。实现同样的功能，可以有两种命令，一种是带有回



码，一种不带回码（根据从机接收是否发送返回码来区分）。

擦除内部变量存储区 ROM：向数据变量存储区存储数据之前，必须首先擦除对应的扇区。

擦除数组存储区 ROM：向数组存储区存储数据之前，必须首先擦除对应的扇区。

九、系统特殊设置

点动操作联机继电器时间设置(单位 ms)：使用“操作联机继电器模块(发送命令)”这个功能来操作联机继电器模块，其中有一个命令是点动某一路，在此处设置点动时间（继电器闭合一段时间后自动断开）。

临时数据接收到时间间隔：接收临时数据的时候，模块是通过两个字节之间的时间间隔来确定的，超过这个时间间隔就认为是两个数据包，出厂默认设置为 10ms。

临时数据包间隔的毫秒数：系统在接收临时数据包的时候，作为数据包的边界使用的间隔时间，如果两个字节之间的接收时间差超过这个时间间隔，就会被当作是两个不同的数据包处理。系统默认这个间隔时间是 10ms。可以在此重新设定。使用方法一般是设备上电作为输入，然后设定为需要的值。也可以使用其他方法启动这个设置。



输入连击和长按的时间常数(ms)，输入条件-外部输入，有单击、双击、连续按下多次的输入选项，这种输入的意思是在某一个固定的时间内按键被连续按下了几次。这个时间常数默认是 500 毫秒。也就是说，输入端在 500 毫秒以内被连续按下 2 次，双击这个条件就会成立。这个时间常数可以在此更改。

程序运行间隔(ms, 数字越小越快)：程序默认每间隔 10ms 运行一次，在此可以更改。

十、临时数据包处理

数据口接收到的临时数据包放在临时数据缓冲区里面，用户可以把它赋值给数组，然后对数组进行处理，比如删除或者增加字节，提取内部的某几个字节作为变量，或者通过数据口把数组发送出去。

具体应用：通过临时数据包的处理功能，可以使模块兼容市面上的绝大部分 485 或者串口或者 ttl 的通讯协议，通过协议转发可以让不同厂家的模块互相配套兼容，或者可以用来当作主机控制不同的厂家的产品。

十一、数组简单处理

参数 1 选择将要被处理的是哪个数组，参数 2 选择将要进行的处理方式，参数 3 输入具体的处理



内容。每个处理方式的含义如下：

数组无参数处理，参数 3 内部包含了几个处理方式

加上其他数组：参数 3 选择输入被加入的数组编号，然后将这个数组添加加到参数 1 代表的数组的尾部，并且数组的长度做出相应的更改。

赋值为其他数组：将参数 3 代表的数组赋值给参数 1 代表的数组，这样会删除数组 1 原有的数据内容，并用参数 3 代表的的数据代替。

删除字节：删除某一个字节，参数 1 代表序号，序号从 1 开始，不论是字符串还是数组同样有效。

在最前面添加一个变量字节：在数组的最前面添加一个字节，这个字节增加了数组的长度并且把第一个字节改为这个字节。

十二、内部变量的使用方法

用户可以将系统内部的其他变量的值赋值给内部变量，也可以将内部变量的值赋值给其他的变量，然后再通过读写内部变量的值来读写系统内其他变量的值。

内部变量不够的时候，有的板卡内部有单独的存储区，用来存储内部变量的值，使用如下的命令来操作：

-> 内部变量 1 保存值到RAM1

通过这条指令，内部 RAM 区地址 1 保存了内部变



量 1 的值。同样的道理，保存到 ROM1 就是保存到掉电不丢失的区域（此功能需要有非易失存储器的板卡支持）。

内部变量可以进行位操作，可以读写内部变量的某一位。也可以根据某一位的值作为输入来判断。

十三、定时处理，需要带有时钟芯片的硬件支持

- 1 如果 时间设定：进入时刻 07时 30分 00秒
或者 时间设定：进入时刻 14时 30分 00秒
-> 输出操作 闭合第1路

通过这个程序，每天早上和下午定时打开第一路继电器输出。

- 1 如果 时间设定：大于 08时 00分 00秒
或者 时间设定：进入时刻 12时 00分 00秒
-> 输出操作 闭合第1路
若条件不满足-> 输出操作 断开第1路

通过这个程序，每天 8 点到 12 点之间第一路是闭合的，其他时间是断开的。

十四、格式转换

内部变量可以与内部的数组进行相互转换，可以有多种转换方式如下：

十进制字符串与内部变量的相互转化：

-> 内部变量1转换为数据包变量1字符串



通过这条指令的执行，如果内部变量十进制的值为123，则数组1的内容为“123”，也就是数组：0x31 0x32 0x33，数据包的长度为3，转换方式为根据需要长度转换（默认）。如果转换的长度设置为4（可以设置），则数组1的内容为“0123”，也就是数组：0x30 0x31 0x32 0x33。

-> 数据包变量1字符串转换为内部变量1

字符串转换为内部变量不受字符串长度的影响，如果是十进制字符串：“12345”，则内部变量的值为12345。

数组与内部变量的相互转换：

数组为高位在前，低位在后，系统默认数组十六进制的数组，一个字节代表两个十六进制位。用户也可以更改系统转换的设置，比如设置数组为十进制的数组，一个字节一个十进制数据，在这种情况下，0x01 0x02 0x03 0x04 转化为内部变量后的值为1234十进制。

十五、实际应用，提取临时数据中的变量
接收到临时数据后赋值给数组1，然后执行：

1 如果 接收到临时数据包

并且 临时数据包共 8个字节

并且 临时数据包从第1个字节为:55 01

-> 临时数据包赋值给数据包变量1

-> 数据包变量2 = 从第3个字节开始的4个字节

-> 数据包变量2数组转换为内部变量1



此程序的功能，先判断临时数据包的字节数，然后判断前两个字节的值为 0x55 0x01，上面的两个判断成立后，将临时数据包从第三个字节开始的四个字节（第 3，4，5，6 字节）的值转换为内部变量 1 的值（中间使用的数组作为转换用）。

十六、实际应用，变量按照固定格式发送出去

- > 设置类型转换时字节的数量为:固定4字节
- > 内部变量1转换为数据包变量2数组
- > 数据包变量1赋值为数组:22 01 10
- > 数据包变量1=数据包变量1+数据包变量2
- > 求数据包变量1累加和并赋值给内部变量1
- > 在数据包变量1最后面添加一个字节,字节的值为内部变量1
- > 通过数据口1 UART1发送数据包变量1

此程序的功能，就是将内部变量 1 转化为四个字节的数组，然后再最左面添加 22 01 10 三个字节，在最后面添加和校验字节，然后通过数据口 1 发送出去。

十七、实际应用，协议转换器

使用内部变量可以将不同厂家的协议对接，使用不同的 485 接口连接不同的厂家的产品，然后模块可以对总线信息进行编程从而将不同厂家的产品连接在一起。前提是厂家的通讯协议开放。

十八、实际应用，信科电子协议举例

- 1 如果 临时数据包和校验成功
 - 并且 临时数据包从第1个字节为:55
 - 并且 临时数据包从第2个字节为:01
 - > 函数调用 函数1

接收到数据后，判断第一个和第二个字节以及和校验成功后，执行相应的协议内容，执行函数1。

函数库	
导入	导出
添加	删除
删除全部	退出
函数1 ✓	1 如果 临时数据包从第3个字节为:10 ->输出 数据包变量1赋值为数组:22 01 10
函数2	->输出 内部变量1赋值为本机开关量输出
函数3	->输出 设置类型转换时字节的数量为:固定4字节
函数4	->输出 内部变量1转换为数据包变量2数组
函数5	->输出 数据包变量1=数据包变量1+数据包变量2
函数6	->输出 求数据包变量1累加和并赋值给内部变量1
函数7	->输出 在数据包变量1最后面添加一个字节,字节的值为内部变量1
函数8	->输出 通过数据口1 UART1发送数据包变量1
函数9	
函数10	2 如果 临时数据包从第3个字节为:11 ->输出 临时数据包赋值给数据包变量1
函数11	->输出 数据包变量2 = 从第7个字节开始的1个字节
函数12	->输出 数据包变量2数组转换为内部变量1
函数13	->输出 输出操作 断开内部变量1路
函数14	
函数15	3 如果 临时数据包从第3个字节为:12 ->输出 临时数据包赋值给数据包变量1
函数16	->输出 数据包变量2 = 从第7个字节开始的1个字节
函数17	->输出 数据包变量2数组转换为内部变量1
函数18	->输出 输出操作 闭合内部变量1路
函数19	
函数20	4 如果 临时数据包从第3个字节为:13 ->输出 临时数据包赋值给数据包变量1
函数21	->输出 数据包变量2 = 从第7个字节开始的1个字节
函数22	->输出 数据包变量2数组转换为内部变量1
函数23	->输出 输出操作 赋值为内部变量1的值
函数24	
函数25	
函数26	
函数27	
函数28	
函数29	
函数30	
函数31	
函数32	



通过函数列表可以看到协议的内容,函数 1 内部,通过继续判断其他字节,来执行相应的操作。如果有需要,函数可以继续调用其他函数。