

# 5.0 内核工具软件使用说明及解析

V1.1

信科电子



本文档适用于信科电子出品的  
5.0 内核工具软件的使用

# 目 录

## 一、概述

## 二、软件特色

## 三、程序结构

## 四、各选项说明及解析

### 1、输入条件

(1) 类型：外部输入·····	6
(2) 类型：输出状态变化·····	6
(3) 类型：定时器·····	7
(4) 类型：位变量·····	8
(5) 类型：整数·····	8
(6) 类型：全局整数·····	9
(7) 类型：浮点数·····	9
(8) 类型：全局浮点数·····	9
(9) 类型：数组·····	9
(10) 类型：全局数组·····	11
(11) 类型：固定输入·····	11
(12) 类型：时间设定·····	11
(13) 类型：日期设定·····	11
(14) 类型：数据口接收到数据·····	11
(15) 类型：其他输入数据·····	13
(16) 类型：函数表达式·····	14
(17) 类型：内部总线接收到数据·····	17
(18) 类型：网络数据·····	17

## 2、输出结果

(1) 类型：输出继电器·····	19
(2) 类型：定时器·····	20
(3) 类型：位变量·····	20
(4) 类型：整数·····	21
(5) 类型：全局整数·····	23
(6) 类型：浮点数·····	23
(7) 类型：全局浮点数·····	25
(8) 类型：数组·····	25
(9) 类型：全局数组·····	32
(10) 类型：系统设置·····	32
(11) 类型：数据口发数据·····	39
(12) 类型：表达式·····	40
(13) 类型：函数处理·····	40
(14) 类型：通讯协议发命令·····	42
(15) 类型：掉电存储·····	45
(16) 类型：内部总线数据处理·····	47
(17) 类型：操作 SD 卡·····	47
(18) 类型：点阵液晶显示·····	47
(19) 类型：其他输出数据·····	47
(20) 类型：网络通讯·····	47

## 一、概述

5.0 内核工具软件，是一款全中文编程软件，通过“添加指令”选项，选择输入条件或输出结果，完成编程，实现逻辑运行。

## 二、软件特色

### 语句块功能

语句块用来表达一个单独的语句结构。语句块内的语句受到条件的影响，只有条件满足，语句块内部的语句才会执行。语句块还可以用来区分不同的层次，从而进行多级判断时更方便。

### 表达式功能

表达式由参与运算参数和运算符组成。用户可手动输入要执行的表达式，使用更灵活。

### 调用函数功能

同一个功能做成一个函数，主程序中再调用函数，可以大大增强主程序的易读性，避免主程序过长让人眼花缭乱。

函数功能还能节省空间，同一个功能使用同一个函数模块，可以反复调用，从而避免主程序中同一个功能代码的重复出现。

### 重命名功能

软件中的任意选项，都可以使用重命名功能，或者自定义的某组固定数据也可以用重命名功能。重命名功能可以方便客户记忆，更直观的显示程序代表的设备及功能。

## 三、程序结构

5.0 内核工具软件写程序，一般是通过判断输入条件，来执行输出结果，实现逻辑运行。

### 输入条件：

单一条件、多个条件 并且关系、多个条件 或者关系。

### 输出结果：

输出结果必须用语句块括起来。语句块用来表达一个单独的语句结构。语句块内的语句受到条件的影响，只有条件满足，语句块内部的语句才会执行。语句块还可以用来区分不同的层次，从而进行多级判断时候更方便。鼠标在空白区域右键可以添加语句块。

语句块效果为：{ }

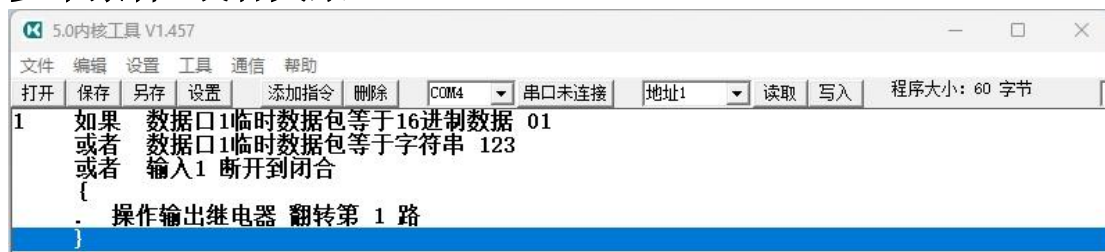
## 程序界面展示： 单一条件



## 多个条件 并且关系



## 多个条件 或者关系



## 多级判断



## 四、各选项说明及解析

### 1、输入条件

#### (1) 类型：外部输入

检测外部输入端情况，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（断开、闭合）以及单击、双击、长按、连续按下多次触发。

\*状态触发（断开、闭合）

注意事项：有可能这个条件会一直成立。

\*单击、双击、长按、连续按下多次触发

注意事项：默认时间常数为 500ms（比如在 500ms 内连续按下两次并抬起，则双击触发成功）。

\*系统设置里面可以设置外部输入连击或者长按的时间常数。

★参数 1：选择输入的序号 输入 1-输入 32

参数 2：选择要判断的条件

断开/闭合/断开到闭合/闭合到断开/状态翻转/单击/双击/长按

参数 3：空

●此功能用法可以参考程序示例：1-外部输入-输出操作-1；  
1-外部输入-输出操作-2。

参数 2：连续按下多次（参数 3 表达式）

参数 3：手动输入表达式，代表要连续按下的次数

例：参数 1：输入 1， 参数 2：连续按下多次， 参数 3： 3  
则代表输入 1 在 500ms 内连续按下 3 次为条件成立

参数 2：等于参数 3 表达式

参数 3：手动输入表达式或整数，1 代表闭合，0 代表断开

例：参数 1：输入 3， 参数 2：等于参数 3 表达式， 参数 3： 1  
则代表输入 3 闭合为条件成立

●此功能用法可以参考程序示例：1-外部输入-输出操作-2

#### (2) 类型：输出状态变化

输出状态变化代表继电器输出，根据输出的状态触发，与根据输入状态触发功能相似，系统单独检测输出的状态并根据输出的状态（断开、闭合）或者状态变化（断开到闭合、闭合到断开、状态翻转）来确定这个条件是否成立。

★**参数 1:** 选择输出的序号 输出 1-输出 32

**参数 2:** 选择要判断的条件

断开/闭合/断开到闭合/闭合到断开/状态翻转

**参数 3:** 空

**参数 2:** 等于参数 3 表达式

**参数 3:** 手动输入表达式或整数，1 代表闭合，0 代表断开

例：**参数 1:** 输出 3， **参数 2:** 等于参数 3 表达式，

**参数 3:** 1

则代表输出 3 闭合为条件成立

例：**参数 1:** 输出 3， **参数 2:** 等于参数 3 表达式，

**参数 3:** 输出 1

则代表输出 3=输出 1 为条件成立

●此功能用法可以参考程序示例：2-输出状态变化-1

### (3) 类型：定时器

定时器的使用非常广泛，定时器既可以作为输入，也可以作为输出。程序每执行一次，都会检查定时器的状态并进行相应的操作。

\*参数 1 和参数 2 根据需求选择，参数 3 手动输入定时器的值。

定时器的值是 0 或者正整数（取值范围：0 - 2147483647）

\*定时器默认倒计时形式，即单位时间减一。通过计算定时器的差值来算出间隔时间，比如：定时器差值 100=1 秒，也可将定时器设置为单位时间加一或者暂停计数以及继续计数。

\*如果定时器处在单位时间减一的状态，并且定时器的值不等于 0，系统就会自动让定时器减 1 一次，每个周期（系统默认周期 10ms）都会检查并执行一次，直到定时器的值为 0。

\*当程序检测到定时器条件时，会检测定时器是不是等于或者大于或者小于这个给定的值或者内部数据变量的值，如果检测的结果成立，就会认为此条件成立

\*注意事项：定时器的值为 0 时，将不再进行单位时间减一操作，定时器将一直等于 0。因此使用定时器 = 0 为触发条件时，有可能会持续触发，因为这个条件有可能会持续成立。

★**参数 1:** 选择定时器的序号 定时器 0-定时器 32

**参数 2:** 选择要判断的条件

等于某值/大于某值/小于某值/大于等于/小于等于/不等于

**参数 3:** 手动输入整数数值（0 - 2147483647）

参数 2: 减到 0

参数 3: 空

●此功能用法可以参考程序示例：3-定时器-1

#### (4) 类型：位变量

位变量可以理解为中间继电器，是虚拟的继电器，位变量既可以作为输入，也可以作为输出使用。程序可以打开或者关闭中间继电器，也可以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。

★参数 1: 选择位变量的序号 位变量 0-位变量 32

参数 2: 选择要判断的条件

=0/=1/0 到 1 跳变/1 到 0 跳变/状态翻转

参数 3: 空

参数 2: 等于参数 3

参数 3: 手动输入表达式或整数，1 代表闭合，0 代表断开

●此功能用法可以参考程序示例：4-位变量-1

#### (5) 类型：整数

整数可以作为输入，也可以作为输出。不同产品，整数的数量不同。很多情况会用到数值变量来存储一些中间值或者中间状态，这时，我们就可以用到整数功能。

\*整数取值范围：-2147483648 — 2147483647

\*整数作为输入使用，数值变量可以与其他数值变量做比较，或与某一个固定数值做比较，比较的结果用来作为输入条件的判断。

\*整数作为输出使用，数值变量可以存储一些临时的值，比如数据口收到的临时数据中的某个字节的值，然后将这些值作为输入使用与其他的数值做比较来触发输出或者定时器等。

\*整数功能，局部有效。

★参数 1: 选择整数的序号 整数 0-整数 19

参数 2: 选择要判断的条件

等于某值/大于某值/小于某值/大于等于/小于等于/不等于

参数 3: 手动输入整数数值（-2147483648 — 2147483647）

参数 2: 判断某一位

参数 3: 手动输入要判断的位数

●此功能用法可以参考程序示例：5-整数-1



### (6) 类型：全局整数

全局整数功能与整数功能类似，可参考整数说明。

\*全局整数 0-全局整数 99

\*全局整数取值范围：-2147483648 — 2147483647

\*全局整数功能，全局有效。

●此功能用法可以参考程序示例：6-全局整数-1

### (7) 类型：浮点数

\*浮点数的运算包括加法、减法、乘法、除法等。

\*默认浮点数精确到小数点后 6 位。

\*可以通过系统设置，来设置浮点数转字符串的小数点位数。

\*浮点数功能，局部有效。

★参数 1：选择浮点数的序号 浮点数 0-浮点数 19

参数 2：选择要判断的条件

等于某值/大于某值/小于某值/大于等于/小于等于/不等于

参数 3：手动输入浮点数数值

●此功能用法可以参考程序示例：7-浮点数-1

### (8) 类型：全局浮点数

全局浮点数与浮点数功能类似，可参考浮点数说明。

\*全局浮点数 0-全局浮点数 31。

\*全局浮点数功能，全局有效。

●此功能用法可以参考程序示例：8-全局浮点数-1

### (9) 类型：数组

数组是内部的 16 进制字节组合或字符串组合。数组既可以作为输入，也可以作为输出使用。

\*一个数组中可以存储若干字节，一个模块中含有若个数组。

不同种类的模块内部数组的数量不一样，每个数组中最多存放的字节数也不一样。具体可以通过读取模块硬件信息来查看。

\*参数 1 和参数 2 根据需要选择，参数 3 手动输入数组的内容。

\*数组作为输入使用，可以判断每个数组的字节数大小或者判断各种校验是否通过。也可以判断两个数组的内容是否完全相同。

\*数组作为输出使用，是一个很重要的功能，尤其是将临时数据包的内容赋值给数组，并对数组进行各种处理。可以用来处理各

种通讯协议数据。

\*数组功能，局部有效。

★**参数 1**：选择数组的序号 数组 0-数组 9

**参数 2**：等于 16 进制数据/等于字符串/等于格式化字符串

**参数 3**：手动输入表达式

例：**参数 1**：数组 0 **参数 3**：01 02

则 判断数组 0 是否等于 16 进制数据：01 02

例：**参数 1**：数组 1 **参数 3**：123

则 判断数组 1 是否等于字符串：123

例：**参数 1**：数组 2 **参数 3**：“目标数据%d”，156

则 判断数组 2 是否等于格式化字符串：“目标数据%d”，156

**参数 2**：和校验判断

**参数 3**：空

例：**参数 1**：数组 3 则 判断数组 3 是否和校验成功

**参数 2**：crc 校验判断

**参数 3**：空

例：**参数 1**：数组 4 则 判断数组 4 是否 crc 校验成功

**参数 2**：字节数判断

**参数 3**：手动输入表达式或者数值

例：**参数 1**：数组 6 **参数 3**：5

则 判断数组 6 的字节数是否等于 5 字节

●此功能用法可以参考程序示例：9-数组-1

**参数 2**：等于表达式

**参数 3**：手动输入表达式

例：**参数 1**：数组 0 **参数 3**：数组 1

则 判断数组 0 是否等于数组 1

**参数 2**：包含表达式

**参数 3**：手动输入表达式

例：**参数 1**：数组 1 **参数 3**：数组 2

则 判断数组 1 是否包含数组 2

**参数 2**：某位置开始等于表达式

**参数 3**：手动输入表达式

例：**参数 1**：数组 1 **参数 3**：1，数组 3

则 判断数组 1 位置 1 是否等于数组 3

●此功能用法可以参考程序示例：9-数组-2

### （10）类型：全局数组

\*全局数组功能同数组功能类似，可参考数组说明。

\*全局数组 0-全局数组 38。

\*全局数组功能，全局有效。

●此功能用法可以参考程序示例：10-全局数组-1；  
10-全局数组-2。

### （11）类型：固定输入

有一些特殊的输入，可以用固定输入来解决，比如：设备上电，此条件为仅模块上电时，执行一次。

●此功能用法可以参考程序示例：11-固定输入

### （12）类型：时间设定

根据预先设定好的时间，执行想要的输出结果。

\*进入时刻，指的是这一秒钟的进入的时刻，比如说，10:15:30 的进入时刻，在时钟由 10:15:29 变化到 10:15:30 的那一瞬间，这一条件就会成立。

\*大于、小于、等于，这是对时间进行的运算结果，00:00:00 为最小值，23:59:59 为最大值。

●此功能用法可以参考程序示例：12-时间设定

### （13）类型：日期设定

日期设定指的是对当前日期进行判断。

\*进入时刻，指的是由条件不成立到条件成立的那一时刻，比如星期 4 的进入时刻，就是星期三的晚上 23 点 59 分 59 秒过渡到星期 4 的 00 点 00 分 00 秒的那一时刻。年月日星期都可以这样。

\*大于、小于、等于，根据数字的大小作出判断。

●此功能用法可以参考程序示例：13-日期设定-（1-3）

### （14）类型：数据口接收到数据

数据口接收到数据，是很常用的功能，可以用来判断哪个数据口接收到数据，也可以判断数据的全部内容或者部分内容。

\*16 进制数据：参数 3 手动输入数据内容，需要注意字节与字节之间使用空格作间隔。

\*字符串：参数 3 手动输入数据内容，字符串可以是字母、数字、符号、空格等单个字符或多个字符组成。

**\*表达式:** 参数 3 手动输入表达式内容, 表达式可以是固定数据, 也可以是预先设置好的数组或整数。

★**参数 1:** 选择数据口的序号    数据口 1-数据口 10

**参数 2:** 有无判断

**参数 3:** 空

例: **参数 1:** 数据口 1    则代表判断数据口 1 是否接收到数据

**参数 2:** 等于 16 进制数据/等于字符串/等于格式化字符串

**参数 3:** 手动输入表达式

例: **参数 1:** 数据口 1    **参数 3:** 01 02

则 判断数据口 1 接收到的数据, 是否等于 16 进制数据: 01 02

例: **参数 1:** 数据口 1    **参数 3:** 123

则 判断数据口 1 接收到的数据, 是否等于字符串: 123

例: **参数 1:** 数据口 1    **参数 3:** "目标数据%d", 整数 0

则 判断数据口 1 接收到的数据, 是否等于格式化字符串: "目标数据%d", 整数 0

**参数 2:** 和校验判断

**参数 3:** 空

例: **参数 1:** 数据口 1

则 判断数据口 1 接收到的数据, 是否和校验成功

**参数 2:** CRC16 校验判断

**参数 3:** 空

例: **参数 1:** 数据口 1

则 判断数据口 1 接收到的数据, 是否 CRC16 校验成功

**参数 2:** 字节数判断

**参数 3:** 手动输入表达式或者数值

例: **参数 1:** 数据口 1    **参数 3:** 6

则 判断数据口 1 接收到的数据, 是否等于 6 个字节

●**此功能用法可以参考程序示例: 14-数据口接收到数据-1**

**参数 2:** 等于表达式

**参数 3:** 手动输入表达式

例: **参数 1:** 数据口 1    **参数 3:** {03 04}

则 判断数据口 1 接收到的数据, 是否等于 16 进制数据: 03 04

**参数 2:** 包含表达式

**参数 3:** 手动输入表达式

例：参数 1：数据口 1            参数 3：数组 0  
则 判断数据口 1 接收到的数据，是否包含数组 0

参数 2：某位置开始等于表达式

参数 3：手动输入表达式

例：参数 1：数据口 1            参数 3：2，数组 0  
则 判断数据口 1 接收到的数据，位置 2 是否等于数组 0

●此功能用法可以参考程序示例：14-数据口接收到数据-2

### （15）类型：其他输入数据

**\*联机开关量：**这个选项是为了支持本公司的开关量产品而定的。根据信科通讯协议，信科的产品在接收到有回码指令控制或者查询命令后，会返回一个 0x22 开头的回码。其他模块的回码会被本模块获取到，模块会将总线上所有 0x22 开头的继电器板的回码记录下来，并且以此为依据记录下总线上所有的继电器模块的输入和输出状态。只要总线上有其他继电器模块的状态返回码数据，本模块就会更新本地的关于这个地址的状态的记录信息。联机开关量判断就是根据这个记录信息作出的。实际使用中，往往向总线发送某一个地址的查询码去激发目标模块发送状态信息，然后就可以使用联机开关量判断了。

★参数 1：联机开关量

参数 2：断开

参数 3：手动输入地址及路数

例：参数 3：地址 2    第 1 路  
则 判断联机开关量地址 2 的第 1 路是否为断开状态

参数 2：闭合

参数 3：手动输入地址及路数

例：参数 3：地址 2    第 1 路  
则 判断联机开关量地址 2 的第 1 路是否为闭合状态

参数 2：断开到闭合

参数 3：手动输入地址及路数

例：参数 3：地址 2    第 1 路  
则 判断联机开关量地址 2 的第 1 路是否 断开到闭合

参数 2：闭合到断开

参数 3：手动输入地址及路数

例：参数 3：地址 2    第 1 路



则 判断联机开关量地址 2 的第 1 路是否 闭合到断开

**参数 2:** 状态翻转

**参数 3:** 手动输入地址及路数

例: **参数 3:** 地址 2 第 1 路

则 判断联机开关量地址 2 的第 1 路是否 状态翻转

●此功能用法可以参考程序示例: 15-其他输入数据-联机开关量

**\*无线遥控信号:** 一般通过接收到无线遥控器信号作为输入, 来触发一些输出动作。无线遥控器的对码, 在通讯成功的情况下(可以正常读取和写入程序), 可以进行遥控器对码。

★**参数 1:** 无线遥控信号

**参数 2:** 接收到无线遥控信号

**参数 3:** 鼠标点击参数 3 对话框后, 在十秒内按下遥控器, 电脑上会弹出一个对话框表示接收到遥控信号, 点击确定, 这个遥控信号的遥控码会自动填写到无线参数里面。

例: 接收到无线遥控信号: {00 00 11 EF 00 18 99 C5}

**参数 2:** 接收到任意无线遥控信号

**参数 3:** 空

即接收到任意无线遥控信号都可以触发成功

**参数 2:** 接收到无线遥控对码信号

**参数 3:** 空

即接收到无线遥控对码信号触发成功

●此功能用法可以参考程序示例: 15-其他输入数据-无线遥控信号

**\*公共数据量:** 此功能未开发, 请忽略。

**\*模拟量输入:** 此功能未开发, 请忽略。

**\*万能红外信号:** 此功能未开发, 请忽略。

## (16) 类型: 函数表达式

**\*表达式**

表达式是由数字、运算符、变量等组成的式子, 能明确数学关系并可通过计算得出结果。

★参数 1: 表达式

参数 2: 一般计算表达式

参数 3: 手动输入表达式

例: 参数 3: 数组 1={01 02}

则 判断数组 1 是否等于 16 进制数据: 01 02

●此功能用法可以参考程序示例: 16-函数表达式-1

### \*用户函数

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。

函数的意义: 程序易读性, 同一个功能做成一个函数, 主程序中再调用函数, 可以大大增强主程序的易读性, 避免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出现。

★参数 1: 用户函数

参数 2: 选择函数的序号

参数 3: 手动输入表达式

例: 参数 2: 函数 1    参数 3: {01 02} = ( )

则 判断函数 1 返回的数据是否为 16 进制数据: 01 02

●此功能用法可以参考程序示例: 16-函数表达式-2

### \*循环表达式

循环表达式是编程中用于控制循环结构的重要组成部分, 它决定了循环何时开始、继续和结束。

★参数 1: 循环表达式

参数 2: for 循环

参数 3: 手动输入表达式

例: 参数 3: for(整数 0=1; 整数 0<9; 整数 0=整数 0 + 1)

则 整数 0 为 0-8 的任意整数

●此功能用法可以参考程序示例: 16-函数表达式-3

## \*系统函数

系统函数是指由 5.0 内核编程软件提供的预先定义好的函数，通常用于完成一些特定的功能。

### ★参数 1：系统函数

参数 2：提取数组

参数 3：手动输入表达式

例：参数 3：{01 02 03} = (数组 1, 0, 3)

则 判断数组 1 的第 0 个字节开始的共 3 个字节，是否等于 16 进制数据：01 02 03

参数 2：提取数组尾部

参数 3：手动输入表达式

例：参数 3：{05} = (数组 1, 1)

则 判断数组 1 尾部的 1 个字节是否等于 16 进制数据：05

参数 2：提取长度

参数 3：手动输入表达式

例：参数 3：5 = (数组 1)

则 判断数组 1 的字节数是否等于 5 字节

参数 2：获取芯片唯一识别码

参数 3：手动输入表达式

例：参数 3：{6A 00 5E 00 14 50 4D 59 30 39 32 20} = ( )

则 判断芯片唯一识别码是否等于 16 进制数据：

6A 00 5E 00 14 50 4D 59 30 39 32 20

●此功能用法可以参考程序示例：16-函数表达式-4

## \*外部输入相关函数

此选项功能与 (1) 类型：外部输入功能相似，可以参考其说明。

●此功能用法可以参考程序示例：16-函数表达式- (5-6)

## \*输出操作相关函数

此选项功能与 (2) 类型：输出操作功能相似，可以参考其说明。

●此功能用法可以参考程序示例：16-函数表达式-7

## \*位变量相关函数

此选项功能与 (4) 类型：位变量功能相似，可以参考其说明。

●此功能用法可以参考程序示例：16-函数表达式-8



- \*定时器相关函数：此功能未开发，请忽略。
- \*数据口相关函数：此功能未开发，请忽略。
- \*数学相关函数：此功能未开发，请忽略。

### (17) 类型：内部总线接收到数据

此功能未开发，请忽略。

### (18) 类型：网络数据

此功能仅适用于 V6 版网络产品的产品，可用来判断网络接收数据的数据源，比如 IP 地址、端口、数据内容等。

#### ★参数 1：UDP 控制器

参数 2：判断接收到 UDP 数据

参数 3：手动输入 UDP 控制器的序号

例：参数 3： 0      则 判断 UDP 控制器 0 是否接收到数据

参数 2：判断 UDP 数据源 IP 地址

参数 3：依次填写 UDP 控制器的序号、IP 地址

例：参数 3： 1, "192.168.1.8"

则 判断 UDP 控制器 1 数据源 IP 是否为：192.168.1.8

参数 2：判断 UDP 数据源端口

参数 3：依次填写 UDP 控制器的序号、端口号

例：参数 3： 1, 4001

则 判断 UDP 控制器 1 数据源端口号是否为：4001

参数 2：判断接收数据字节数等于

参数 3：依次填写 UDP 控制器的序号、字节数量

例：参数 3： 2, 5

则 判断 UDP 控制器 2 数据源字节数是否为：5 个字节

#### ●此功能用法可以参考程序示例：24-网络数据-1

#### ★参数 1：TCP 控制器

参数 2：判断接收到 TCP 数据

参数 3：手动输入 TCP 控制器的序号

例：参数 3： 0      则 判断 TCP 控制器 0 是否接收到数据

参数 2：判断 TCP 连接成功

参数 3：手动输入 TCP 控制器的序号

例：参数 3： 0      则 判断 TCP 控制器 0 是否连接成功

**参数 2:** 判断 TCP 发送数据成功

**参数 3:** 手动输入 TCP 控制器的序号

例: **参数 3:** 0 则 判断 TCP 控制器 0 发送数据是否成功

**参数 2:** 判断连接对方 IP 地址

**参数 3:** 依次填写 TCP 控制器的序号、IP 地址

例: **参数 3:** 0, "192.168.1.110"

则 判断 TCP 控制器 0 对方 IP 是否为: 192.168.1.110

**参数 2:** 判断连接对方端口

**参数 3:** 依次填写 TCP 控制器的序号、端口号

例: **参数 3:** 0, 6000

则 判断 TCP 控制器 0 数据源端口号是否为: 6000

●此功能用法可以参考程序示例: 24-网络数据-2

★**参数 1:** TCP 服务端口

**参数 2:** 判断服务端口接收到数据

**参数 3:** 服务端口号

例: **参数 3:** 3000

则 判断 TCP 服务端口 3000 是否接收到数据

**参数 2:** 判断服务端口接收的数据的来源 IP

**参数 3:** 依次填写 服务端口号、IP 地址

例: **参数 3:** 3000, "192.168.1.8"

则 判断 TCP 服务端口 3000 接收到数据的对方 IP 是否为:  
192.168.1.8

**参数 2:** 判断服务端口接收的数据的来源端口

**参数 3:** 依次填写 服务端口号、对方端口号

例: **参数 3:** 3000, 6001

则 判断 TCP 服务端口 3000 接收到数据的对方端口是否为: 6001

**参数 2:** 判断服务端口接收到数据的内容

**参数 3:** 依次填写 服务端口号、数据内容

例: **参数 3:** 3000, {03 04}

则 判断 TCP 服务端口 3000 接收到的数据是否为 16 进制数据:  
03 04

**参数 2:** 判断服务端口接收到数据的字节数

**参数 3:** 依次填写 服务端口号、字节数量

例: **参数 3:** 3000, 5

则 判断 TCP 服务端口 3000 接收到的数据是否为：5 个字节

**参数 2：**判断服务端口接收到数据的某几个字节

**参数 3：**依次填写 服务端口号、位置、数据内容

例：**参数 3：** 3000, 1, {04 05}

则 判断 TCP 服务端口 3000 接收到的数据的第 1 个字节开始是否为 16 进制数据：04 05

●此功能用法可以参考程序示例：24-网络数据-3

## 2、输出结果

### (1) 类型：输出继电器

用来操作继电器输出的选项，使用相对简单。

★**参数 1：**简单动作

代表继电器路数的参数使用 10 进制表示

**参数 2：**闭合某路/断开某路/翻转某路

**参数 3：**手动输入要操作的继电器路数。

●此功能用法可以参考程序示例：1-外部输入-输出操作-1

**参数 2：**点动某路

**参数 3：**手动输入要点动的路数及点动的时间，时间单位 ms

例：**参数 3：** 3, 500      则点动第 3 路 500ms

**参数 2：**全部闭合/全部断开/全部翻转

**参数 3：**空

●此功能用法可以参考程序示例：1-外部输入-输出操作-2

**参数 2：**全部点动

**参数 3：**手动输入点动时间，单位 ms

**参数 2：**依次闭合全部/依次断开全部

**参数 3：**手动输入间隔时间，单位 ms

**参数 2：**依次闭合某几路/依次断开某几路

**参数 3：**依次填写 间隔时间，具体路数，用逗号做间隔

例：**参数 3：** 300, 1, 5, 8

则间隔 300ms 依次闭合第 1 路，第 5 路，第 8 路

●此功能用法可以参考程序示例：1-外部输入-输出操作-3

★参数 1: 赋值

参数 2: 选择输出的路数

参数 3: 手动输入表达式

例: 参数 2: 输出 1

参数 3: 0 , 则输出 1=0, 0 代表断开, 即输出 1 断开

参数 3: 1 , 则输出 1=1, 1 代表闭合, 即输出 1 闭合

例: 参数 2: 输出 2                      参数 3: 输入 2

则输出 2=输入 2, 可实现输出 2 状态与输入 2 状态同步。

例: 参数 2: 输出 5                      参数 3: 整数 1

则输出 5=整数 1, 整数 1 需要提前赋值为 0 或者 1。

●此功能用法可以参考程序示例: 1-外部输入-输出操作-4

(2) 类型: 定时器

定时器的使用非常广泛, 定时器既可以作为输入, 也可以作为输出。程序每执行一次, 都会检查定时器的状态并进行相应的操作。

\*定时器默认倒计时形式, 即单位时间减一。通过计算定时器的差值来算出间隔时间, 比如: 定时器差值 100=1 秒, 也可将定时器设置为单位时间加一或者暂停计数以及继续计数。

\*如果定时器处在单位时间减一的状态, 并且定时器的值不等于 0, 系统就会自动让定时器减 1 一次, 每个周期 (系统默认周期 10ms) 都会检查并执行一次, 直到定时器的值为 0。

\*定时器的值是 0 或者正整数 (取值范围: 0 - 2147483647 )

★参数 1: 选择定时器的序号, 支持定时器 0-定时器 31。

参数 2: 赋值为参数 3

参数 3: 手动输入定时器的值。

参数 2: 单位时间加 1/单位时间减 1/暂停计数/继续计数

参数 3: 空

●此功能用法可以参考程序示例: 3-定时器-2

(3) 类型: 位变量

位变量可以理解为中间继电器, 是虚拟的继电器, 位变量既可以作为输入, 也可以作为输出使用。程序可以打开或者关闭中间继电器, 也可以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。

★参数 1: 简单动作

参数 2: 置 0 位变量/置 1 位变量

参数 3: 手动输入位变量序号

●此功能用法可以参考程序示例: 4-位变量-1

参数 2: 翻转位变量

参数 3: 手动输入位变量序号

参数 2: 点动位变量

参数 3: 依次填写 位变量序号、点动时间

例: 参数 3: 1, 1000 则点动位变量 1 时间为 1000ms

参数 2: 全部置 0/全部置 1/全部翻转

参数 3: 空

●此功能用法可以参考程序示例: 4-位变量-2

★参数 1: 赋值

此功能不完善, 只能给位变量 1 赋值为 0 或者 1, 可忽略不用

#### (4) 类型: 整数

整数可以作为输入, 也可以作为输出。不同产品, 整数的数量不同。很多情况会用到数值变量来存储一些中间值或者中间状态, 这时, 我们就可以用到整数功能。整数功能, 局部有效。

整数取值范围: -2147483648 — 2147483647

★参数 1: 选择整数的序号

参数 2: 赋值(表达式)

参数 3: 手动输入整数的数值或表达式

例: 参数 3 可以输入: 500、整数 1、输入 1、输出 1……

●此功能用法可以参考程序示例: 5-整数-1

参数 2: 提取字符串中的数字(表达式)

参数 3: 手动输入表达式

例: 参数 1: 整数 0 参数 3: 数据口 1, 1

则提取数据口 1 接收到的数据中第 1 个数字

例: 参数 1: 整数 0 参数 3: 数组 1, 1

则提取数组 1 中第 1 个数字

参数 2: =16 进制字符串转换(表达式)

参数 3: 手动输入表达式



例：参数 1：整数 1                      参数 3：1002

则整数 1=16 进制字符串：10 02

参数 2：=2 进制字符串转换(表达式)

参数 3：手动输入表达式

例：参数 1：整数 2                      参数 3：101101

2 进制转 10 进制，则整数 2=10 进制字符串：45

参数 2：=16 进制数组转换 低位在前(表达式)

参数 3：手动输入表达式

例：参数 1：整数 3                      参数 3：5964034

10 进制转 16 进制低位转换，则整数 3=16 进制：02 01 5B

参数 2：=16 进制数组转换 高位在前(表达式)

参数 3：手动输入表达式

例：参数 1：整数 4                      参数 3：5964034

10 进制转 16 进制低位转换，则整数 4=16 进制：5B 01 02

参数 2：位赋值

参数 3：手动输入表达式

例：参数 1：整数 5                      参数 3：5, 1

2 进制第 5 位为 1，其余为 0，2 进制：100000 转为 10 进制，  
则整数 5=10 进制字符串：32

●此功能用法可以参考程序示例：5-整数-2

参数 2：向外发送 10 进制字符串

参数 3：手动输入数据口序号

例：参数 1：整数 1                      参数 3：1

如果 整数 1=25

则整数 1 通过数据口 1 以 10 进制字符串格式发送出去

数据口 1 发送字符串：25

参数 2：向外发送 16 进制字符串

参数 3：手动输入数据口序号

例：参数 1：整数 2                      参数 3：1

如果 整数 2=4098，10 进制：4098 转为 16 进制数据：10 02

则整数 2 通过数据口 1 以 16 进制字符串格式发送出去

数据口 1 发送字符串：10 02

参数 2：向外发送 2 进制字符串

参数 3：手动输入数据口序号

例：参数 1：整数 3

参数 3：1

如果 整数 3=25，10 进制：25 转为 2 进制数据：11001

则整数 3 通过数据口 1 以 2 进制字符串格式发送出去

数据口 1 发送字符串：11001

**参数 2：**向外发送 16 进制数组 低位在前

**参数 3：**手动输入数据口序号

例：参数 1：整数 4

参数 3：1

如果 整数 4=258，10 进制：258 转为 16 进制数据：01 02

则整数 4 通过数据口 1 以 16 进制数组 低位在前 发送出去

数据口 1 发送 16 进制：02 01

**参数 2：**向外发送 16 进制数组 高位在前

**参数 3：**手动输入数据口序号

例：参数 1：整数 5

参数 3：1

如果 整数 5=258，10 进制：258 转为 16 进制数据：01 02

则整数 5 通过数据口 1 以 16 进制数组 高位在前 发送出去

数据口 1 发送 16 进制：01 02

●此功能用法可以参考程序示例：5-整数-3

### （5）类型：全局整数

全局整数功能与整数功能类似，可参考整数说明。

\*全局整数 0-全局整数 99

\*全局整数取值范围：-2147483648 — 2147483647

\*全局整数功能，全局有效。

●此功能用法可以参考程序示例：6-全局整数-1；

6-全局整数-2；

7-全局整数-3。

### （6）类型：浮点数

\*浮点数的运算包括加法、减法、乘法、除法等。

\*默认浮点数精确到小数点后 6 位。

\*可以通过系统设置，来设置浮点数转字符串的小数点位数。

\*浮点数功能，局部有效。

★参数 1：选择浮点数的序号 浮点数 0-浮点数 19

**参数 2：**赋值

**参数 3：**手动输入浮点数数值

●此功能用法可以参考程序示例：7-浮点数-1

**参数 2:** 提取 10 进制字符串中的数字 表达式

**参数 3:** 输入表达式

例: **参数 1:** 浮点数 0                      **参数 3:** 数组 1, 1

如果数组 1=字符串"123abc456ef"

数组 1 的字符串中第 0 组数字为: 123, 第 1 组数字为: 456

提取数组 1 中的第 1 组数字: 456, 将该数据赋值给浮点数 0

则浮点数 0=456.000000

**参数 2:** 单精度 4 字节转换 低位在前 表达式

**参数 3:** 输入表达式

例: **参数 1:** 浮点数 0                      **参数 3:** {00 00 E0 3F}

16 进制转为浮点数, 则浮点数 0=1.750000

**参数 2:** 单精度 4 字节转换 高位在前 表达式

**参数 3:** 输入表达式

例: **参数 1:** 浮点数 0                      **参数 3:** {3F E0 00 00}

16 进制转为浮点数, 则浮点数 0=1.750000

**参数 2:** 双精度 8 字节转换 低位在前 表达式

**参数 3:** 输入表达式

例: **参数 1:** 浮点数 0                      **参数 3:** {00 00 00 00 00 00 FC 3F}

16 进制转为浮点数, 则浮点数 0=1.750000

**参数 2:** 双精度 8 字节转换 高位在前 表达式

**参数 3:** 输入表达式

例: **参数 1:** 浮点数 0                      **参数 3:** {3F FC 00 00 00 00 00 00}

16 进制转为浮点数, 则浮点数 0=1.750000

**参数 2:** 向外发送 10 进制字符串

**参数 3:** 手动输入数据口序号

例: **参数 1:** 浮点数 0                      **参数 3:** 1

如果浮点数 0=1.750000

则通过数据口 1 发送字符串: 1.750000

●此功能用法可以参考程序示例: 7-浮点数-2

**参数 2:** 向外发送单精度 4 字节 低位在前

**参数 3:** 手动输入数据口序号

例: **参数 1:** 浮点数 0                      **参数 3:** 1

如果浮点数 0=1.750000, 浮点数转 16 进制 4 字节低位在前,

则数据口 1 发送 16 进制数据: 00 00 E0 3F



**参数 2:** 向外发送单精度 4 字节 高位在前

**参数 3:** 手动输入数据口序号

例: **参数 1:** 浮点数 0      **参数 3:** 1

如果浮点数 0=1.750000, 浮点数转 16 进制 4 字节高位在前,  
则数据口 1 发送 16 进制数据: 3F E0 00 00

**参数 2:** 向外发送双精度 8 字节 低位在前

**参数 3:** 手动输入数据口序号

例: **参数 1:** 浮点数 0      **参数 3:** 1

如果浮点数 0=1.750000, 浮点数转 16 进制 8 字节低位在前,  
则数据口 1 发送 16 进制数据: 00 00 00 00 00 00 FC 3F

**参数 2:** 向外发送双精度 8 字节 高位在前

**参数 3:** 手动输入数据口序号

例: **参数 1:** 浮点数 0      **参数 3:** 1

如果浮点数 0=1.750000, 浮点数转 16 进制 8 字节高位在前,  
则数据口 1 发送 16 进制数据: 3F FC 00 00 00 00 00 00

●此功能用法可以参考程序示例: 7-浮点数-3

### (7) 类型: 全局浮点数

全局浮点数与浮点数功能类似, 可参考浮点数说明。

\*全局浮点数 0-全局浮点数 31。

\*全局浮点数功能, 全局有效。

●此功能用法可以参考程序示例: 8-全局浮点数-1;  
8-全局浮点数-2;  
8-全局浮点数-3。

### (8) 类型: 数组

数组是内部的 16 进制字节组合或字符串组合。数组既可以作为输入, 也可以作为输出使用。

\*一个数组中可以存储若干字节, 一个模块中含有若个数组。

不同种类的模块内部数组的数量不一样, 每个数组中最多存放的字节数也不一样。具体可以通过读取模块硬件信息来查看。

\*数组作为输出使用, 是一个很重要的功能, 尤其是将临时数据包的内容赋值给数组, 并对数组进行各种处理。可以用来处理各种通讯协议数据。

\*数组 0-数组 9。

\*数组功能, 局部有效。

★参数 1: 赋值表达式

参数 2: 选择数组的序号

参数 3: 手动输入表达式内容

例: 参数 2: 数组 4      参数 3: {01 01 00 00 00 05 FC 09}

则 数组 4=16 进制数据: 01 01 00 00 00 05 FC 09

例: 参数 2: 数组 4      参数 3: “123”

则 数组 4=字符串: 123

★参数 1: 赋值字符串

参数 2: 选择数组的序号

参数 3: 手动输入数组内容

例: 参数 2: 数组 1      参数 3: 123

则 数组 1=字符串: 123

★参数 1: 赋值 16 进制输入

参数 2: 选择数组的序号

参数 3: 手动输入数组内容

例: 参数 2: 数组 0      参数 3: 01 02

则 数组 0=16 进制数据: 01 02

★参数 1: 赋值格式化字符串

参数 2: 选择数组的序号

参数 3: 手动输入数组内容

例: 参数 2: 数组 2      参数 3: “目标数据%d”, 整数 0

则 数组 2=格式化字符串“目标数据%d”, 整数 0

●此功能用法可以参考程序示例: 9-数组-1

★参数 1: 无参数简单处理

参数 2: 清空

参数 3: 手动填写数组的序号

例: 参数 3: 0 , 则数组 0 清空

参数 2: 字节倒序

参数 3: 手动填写数组的序号

例: 参数 3: 1, 则数组 1 字节倒序

如果原数组 1 赋值为 16 进制: 01 02 03

则 倒序后的数组 1 为 16 进制: 03 02 01

参数 2: 每个字节的前后四位互换

参数 3: 手动填写数组的序号

例：参数 3： 2，则数组 2 每个字节的前后四位互换  
例：原数组 2 赋值为 16 进制：01 02 03 04 05 06 07 08  
则 每个字节的前后四位互换后的数组 2 为 16 进制数据：  
10 20 30 40 50 60 70 80

●此功能用法可以参考程序示例：9-数组-3

参数 2：字符串全部转化为小写/大写

参数 3：手动填写数组的序号

例：原数组 0 赋值为字符串：“123abc456EF”  
则 数组 0 字符串全部转化为小写后，数组 0=“123abc456ef”  
数组 0 字符串全部转化为大写后，数组 0=“123ABC456EF”

参数 2：尾部添加和校验字节

参数 3：手动填写数组的序号

例：数组 1=16 进制数据：01 02 03 04  
则 数组 1 尾部添加和校验后为：01 02 03 04 0A

参数 2：尾部添加 CRC16 校验字节

参数 3：手动填写数组的序号

例：数组 2=16 进制数据：01 01 00 00 00 05  
则 数组 2 尾部添加 CRC16 校验后为：01 01 00 00 00 05 FC 09

●此功能用法可以参考程序示例：9-数组-4

参数 2：每个字节的位倒序

参数 3：手动填写数组的序号

例：数组 0=16 进制数据：01 02 03 04  
则 数组 0 每个字节的位倒序后为 16 进制数据：80 40 C0 20

参数 2：字节转化为 16 进制编辑字符串

参数 3：手动填写数组的序号

例：数组 0=16 进制数据：01 02 03 04  
则数组 0 字节转化为 16 进制编辑字符串后为字符串：01 02 03 04

参数 2：尾部添加 CRC8 校验字节

参数 3：手动填写数组的序号

例：数组 0=16 进制数据：01 02 03 04  
则 数组 0 尾部添加 CRC8 校验字节后为：01 02 03 04 E3

参数 2：尾部添加 CRC8\_MAXIM 校验字节

参数 3：手动填写数组的序号

例：数组 0=16 进制数据：01 02 03 04

则 数组 0 尾部添加 CRC8\_MAXIM 校验字节后为：01 02 03 04 F4

●此功能用法可以参考程序示例：9-数组-5

★参数 1：增加字节

参数 2：给定位置插入表达式

参数 3：依次填写 数组的序号，位置号，表达式内容

例：数组 0=16 进制数据：01 02 04 05， 参数 3：0， 2， {03}

则代表给数组 0 位置 2 插入 16 进制数据：03

给定位置插入表达式后，数组 0=16 进制数据：01 02 03 04 05

参数 2：给定位置插入格式化字符串

参数 3：依次填写 数组的序号，位置号，格式化字符串内容

例：数组 1=字符串“123ab456”，整数 0=10

参数 3：1， 0，“目标数据%d”，整数 0

则代表给数组 1 位置 0 插入格式化字符串“目标数据%d”，整数 0

给定位置插入格式化字符串后，

数组 1=字符串：目标数据 10123ab456

参数 2：头部添加表达式

参数 3：依次填写 数组的序号，表达式内容

例：数组 0=16 进制数据：01 02 03 04， 参数 3：0， {0A 0B}

则头部添加表达式后，数组 0=16 进制数据：0A 0B 01 02 03 04

参数 2：尾部添加表达式

参数 3：依次填写 数组的序号，表达式内容

例：数组 0=16 进制数据：01 02 03 04， 参数 3：0， {06 07}

则尾部添加表达式后，数组 0=16 进制数据：01 02 03 04 06 07

参数 2：延长数组长度到某一长度

参数 3：依次填写 数组的序号，数组长度数值

例：数组 0=16 进制数据：01 02 03 04， 参数 3：0， 6

则延长数组长度为 6 字节后，

数组 0=16 进制数据：01 02 03 04 00 00

●此功能用法可以参考程序示例：9-数组-6

★参数 1：删除字节

参数 2：删除头部多个字节

参数 3：依次填写 数组的序号，删除头部字节的数量

例：数组 0=16 进制数据：01 02 03 04 05 06，**参数 3**：0，2  
则数组 0 删除头部 2 个字节后，数组 0=16 进制数据：03 04 05 06

**参数 2**：删除尾部多个字节

**参数 3**：依次填写 数组的序号，删除尾部字节的数量

例：数组 0=16 进制数据：01 02 03 04 05 06，**参数 3**：0，2  
则数组 0 删除尾部 2 个字节后，数组 0=16 进制数据：01 02 03 04

**参数 2**：删除中间多个字节

**参数 3**：依次填写 数组的序号，位置号，删除字节的数量

例：数组 0=16 进制数据：01 02 03 04 05，**参数 3**：0，2，2  
则删除数组 0 位置 2 开始的 2 个字节后，  
数组 0=16 进制数据：01 02 05

●此功能用法可以参考程序示例：9-数组-7

★**参数 1**：=整数转换

**参数 2**：10 进制字符串

**参数 3**：依次填写 数组的序号，整数的数值

例：**参数 3**：1，15，则数组 1=整数 15 转为 10 进制字符串：15

**参数 2**：16 进制字符串

**参数 3**：依次填写 数组的序号，整数的数值

例：**参数 3**：1，25，则数组 1=整数 25 转为 16 进制字符串：19

**参数 2**：2 进制字符串

**参数 3**：依次填写 数组的序号，整数的数值

例：**参数 3**：1，15，则数组 1=整数 15 转为 2 进制字符串：1111

**参数 2**：16 进制数组 低位在前

**参数 3**：依次填写 数组的序号，整数的数值

例：**参数 3**：1，815

则数组 1=整数 815 转为 16 进制数组 低位在前：2F 03

**参数 2**：16 进制数组 高位在前

**参数 3**：依次填写 数组的序号，整数的数值

例：**参数 3**：1，815

则数组 1=整数 815 转为 16 进制数组 高位在前：03 2F

●此功能用法可以参考程序示例：9-数组-8

★**参数 1**：=浮点数转换

**参数 2**：10 进制字符串



**参数 3:** 依次填写 数组的序号, 浮点数的数值

例: **参数 3:** 1, 12.450000

则数组 1=浮点数 12.450000 转为 10 进制字符串: 12.450000

**参数 2:** 单精度 4 字节数组 低位在前

**参数 3:** 依次填写 数组的序号, 浮点数的数值

例: **参数 3:** 1, 12.450000

则数组 1=浮点数 12.450000 转为单精度 4 字节数组 低位在前  
即 16 进制数据: 33 33 47 41

**参数 2:** 单精度 4 字节数组 高位在前

**参数 3:** 依次填写 数组的序号, 浮点数的数值

例: **参数 3:** 1, 12.450000

则数组 1=浮点数 12.450000 转为单精度 4 字节数组 高位在前  
即 16 进制数据: 41 47 33 33

**参数 2:** 双精度 8 字节数组 低位在前

**参数 3:** 依次填写 数组的序号, 浮点数的数值

例: **参数 3:** 1, 12.450000

则数组 1=浮点数 12.450000 转为双精度 8 字节数组 低位在前  
即 16 进制数据: 66 66 66 66 66 E6 28 40

**参数 2:** 双精度 8 字节数组 高位在前

**参数 3:** 依次填写 数组的序号, 浮点数的数值

例: **参数 3:** 1, 12.450000

则数组 1=浮点数 12.450000 转为双精度 8 字节数组 高位在前  
即 16 进制数据: 40 28 E6 66 66 66 66 66

●此功能用法可以参考程序示例: 9-数组-9

★**参数 1:** 通过数据口发送

**参数 2:** 选择数据口 1-数据口 10

**参数 3:** 手动填写数组的序号

例: **参数 2:** 数据口 1 **参数 3:** 1

则 数组 1 通过数据口 1 发送出去

●此功能用法可以参考程序示例: 9-数组-9

★**参数 1:** 查找和替换

**参数 2:** 查找

**参数 3:** 依次填写 数组的序号, 查找的内容, 查找内容的位

置号赋值给某表达式

例：数组 0={01 02 03 04 05 06} **参数 3**: 0, {05}, 整数 0  
则在数组 0 中查找 16 进制数据：05，将 {05} 的位置号赋值给整数 0。位置从 0 开始数，数组 0 的第 0 个字节为 {01}，第 1 个字节为 {02}，第 4 个字节为 {05}，即整数 0=4

**参数 2: 替换**

**参数 3**: 依次填写 数组的序号，查找的内容，替换的内容

例：数组 0={01 02 03 04 05 06} **参数 3**: 0, {03 04}, {AA BB}  
则在数组 0 中查找 {03 04}，并替换为 {AA BB}，  
处理后，数组 0={01 02 AA BB 05 06}

●此功能用法可以参考程序示例：9-数组-10

★**参数 1: 提取数据/信息**

**参数 2**: 当前字节数 赋值给表达式

**参数 3**: 依次填写 数组的序号，表达式

例：数组 0=16 进制数据：01 02 03 04 05 06 07 08 09 10

**参数 3**: 0, 数组 1

则把数组 0 的字节数赋值给数组 1，数组 0 共 10 个字节，  
即数组 1=16 进制：0A

**参数 2**: 最大允许长度 赋值给表达式

**参数 3**: 依次填写 数组的序号，表达式

例：默认数组的最大字节数为：50 字节，**参数 3**: 0, 数组 1

则数组 0 的最大长度为 50 字节，即数组 1=16 进制：32

用户可以通过“系统运行设置”设置数组边界(允许的最大长度)

**参数 2**: 提取数组某几个元素 赋值表达式

**参数 3**: 依次填写 数组的序号，位置号，数量，表达式

例：数组 0=16 进制数据：01 02 03 04 05 06

**参数 3**: 0, 2, 3, 数组 1

则提取数组 0 中，位置 2 开始的，数量 3 个字节，赋值给数组 1  
即数组 1=16 进制数据：03 04 05

**参数 2**: 设置数组长度 不足补 0 多余删除

**参数 3**: 依次填写 数组的序号，数组长度

例：**参数 3**: 0, 8

即数组 0 的长度设置为 8 个字节，不足补 0 多余删除

例：如果数组 0=16 进制数据：01 02 03 04 05 06

则处理后，数组 0=16 进制数据：01 02 03 04 05 06 00 00  
例：如果数组 0=16 进制数据：01 02 03 04 05 06 07 08 09 0A  
则处理后，数组 0=16 进制数据：01 02 03 04 05 06 07 08

**参数 2：**清零数组每一个元素

**参数 3：**依次填写 数组的序号，

例：**参数 3：** 0

即清零数组 0 中所有元素的值，长度不变

如果数组 0=16 进制数据：01 02 03 04 05 06

则处理后，数组 0=16 进制数据：00 00 00 00 00 00

**参数 2：**直接赋值数组的内容

**参数 3：**依次填写 数组的序号，数组内容

例：**参数 3：** 0, {01 02 03 04 05}

则数组 0 赋值为 16 进制数据：01 02 03 04 05

●此功能用法可以参考程序示例：9-数组-11

### (9) 类型：全局数组

\*全局数组功能同数组功能类似，可参考数组说明。

\*全局数组 0-全局数组 38。

\*全局数组功能，全局有效。

●此功能用法可以参考程序示例：10-全局数组-（1-11）

### (10) 类型：系统设置

★**参数 1：**输入参数设置

**参数 2：**外部输入连击或者长按的时间常数(ms)

**参数 3：**依次填写 输入的序号，时间常数

例：**参数 3：** 1, 3000

即设置输入 1 的连击或长按时间为 3000ms，输入 1 长按 3000ms 未抬起，则长按触发成功

●此功能用法可以参考程序示例：17-系统设置-1

★**参数 1：**系统运行设置

**参数 2：**程序运行间隔(ms, 数字越小越快)

**参数 3：**手动输入时间常数

例：**参数 3：** 5000 即程序每 5000ms 运行一次

●此功能用法可以参考程序示例：17-系统设置-2



参数 2: 程序延迟 (单位 ms)

参数 3: 手动输入时间常数

例: 参数 3: 10000 即程序暂停 10000ms 后, 再继续运行

●此功能用法可以参考程序示例: 17-系统设置-3

参数 2: 设置数组/全局数组边界(允许的最大长度)

参数 3: 手动输入长度值, 可填整数范围: 1-200

●此功能用法可以参考程序示例: 17-系统设置-4

参数 2: 程序延迟 (阻塞, 单位 ms)

参数 3: 手动输入时间常数

例: 参数 3: 10000 即 CPU 原地等待, 10000ms 后, 再恢复运行

●此功能用法可以参考程序示例: 17-系统设置-3

★参数 1: 使能或者失能控制指令

参数 2: 根据需要进行选择 使能或失能的指令

参数 3: 空

●此功能用法可以参考程序示例: 17-系统设置-5

★参数 1: 通讯波特率设置

参数 2: 根据需要进行选择要设置的数据口, 数据口 1-数据口 10

参数 3: 手动输入要设置的波特率参数, 波特率支持:

2400/4800/9600/19200/38400/57600

●此功能用法可以参考程序示例: 17-系统设置-6

★参数 1: 通讯数据位 校验位 停止位设置

参数 2: 根据需要进行选择要设置的数据位、校验位、停止位

参数 3: 手动输入数据口序号

●此功能用法可以参考程序示例: 17-系统设置- (7-10)

★参数 1: 临时数据包设置

参数 2: 根据需要进行选择要设置的功能

参数 3: 手动输入表达式

●此功能用法可以参考程序示例: 17-系统设置- (11-12)

★参数 1: 表达式转换默认设置

参数 2: 根据需求选择要设置的功能

参数 3: 空

●此功能用法可以参考程序示例: 17-系统设置-13

★参数 1: 数组/字符串转整数设置

参数 2: 数组转整数采用字节数组转换模式

参数 3: 空

例: 全局数组 1=字符串: 25

转换后 全局整数 1=16 进制数据: 32 35

参数 2: 数组转整数采用字符串转换模式

参数 3: 空

例: 全局数组 1=字符串: 19, 全局数组 1 转全局整数 1 后,  
全局整数 1=10 进制字符串: 19

参数 2: 字节数组转整数低位字节在前

参数 3: 空

例: 全局数组 1=16 进制: 01 02, 全局数组 1 转全局整数 1 后,  
全局整数 1=16 进制数据: 02 01

参数 2: 字节数组转整数高位字节在前

参数 3: 空

例: 全局数组 1=10 进制: 772, 全局数组 1 转全局整数 1 后,  
全局整数 1=16 进制数据: 03 04

●此功能用法可以参考程序示例: 17-系统设置-14

参数 2: 字符串转整数采用 10 进制(默认)

参数 3: 空

例: 全局数组 1=字符串: 19, 全局数组 1 转全局整数 1 后,  
全局整数 1=10 进制字符串: 19

参数 2: 字符串转整数采用 16 进制

参数 3: 空

例: 全局数组 1=字符串: 19, 全局数组 1 转全局整数 1 后,  
全局整数 1=10 进制字符串: 25

参数 2: 字符串转整数采用 2 进制

参数 3: 空

例: 全局数组 1=字符串: 10011, 全局数组 1 转全局整数 1 后,

全局整数 1=10 进制字符串: 19

●此功能用法可以参考程序示例: 17-系统设置-15

★参数 1: 整数转数组/字符串设置

参数 2: 整数转数组采用字节数组转换模式

参数 3: 空

例: 全局整数 1=25, 全局整数 1 转全局数组 1 后,  
全局数组 1=16 进制数据: 19

参数 2: 整数转数组采用字符串转换模式

参数 3: 空

例: 全局整数 1=25, 全局整数 1 转全局数组 1 后,  
全局数组 1=10 进制字符串: 25

参数 2: 整数转字节数组低位字节在前

参数 3: 空

例: 全局整数 1=258, 全局整数 1 转全局数组 1 后,  
全局数组 1=16 进制数据: 02 01

参数 2: 整数转字节数组高位字节在前

参数 3: 空

例: 全局整数 1=258, 全局整数 1 转全局数组 1 后,  
全局数组 1=16 进制数据: 01 02

参数 2: 整数转字节数组字节数(默认 0 根据需要)

参数 3: 手动输入数值, 数值范围: 1-4

例: 全局整数 1=258, 参数 3: 4, 全局整数 1 转全局数组 1 后,  
全局数组 1=16 进制数据: 00 00 01 02

●此功能用法可以参考程序示例: 17-系统设置-16

参数 2: 整数转字符串采用 10 进制(默认)

参数 3: 空

例: 全局整数 1=258, 全局整数 1 转全局数组 1 后,  
全局数组 1=字符串: 258

参数 2: 整数转字符串采用 16 进制

参数 3: 空

例: 全局整数 1=258, 全局整数 1 转全局数组 1 后,  
全局数组 1=字符串: 102

参数 2: 整数转字符串采用 2 进制

**参数 3:** 空

例：全局整数 1=258，全局整数 1 转全局数组 1 后，  
全局数组 1=字符串：100000010

**参数 2:** 整数转字符串场宽(总位数)设置

**参数 3:** 手动输入数值，数值范围：1-32

例：全局整数 1=258，全局整数 1 转全局数组 1 后，  
全局数组 1=字符串：000000000000000000000000100000010

●此功能用法可以参考程序示例：17-系统设置-17

★**参数 1:** 数组/字符串转浮点数设置

**参数 2:** 数组转浮点数采用字节数组转换模式

**参数 3:** 空

例：全局数组 1={3F E0 00 00}，全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：1.750000

**参数 2:** 数组转浮点数采用字符串转换模式

**参数 3:** 空

例：全局数组 1="19.78"，全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：19.780000

●此功能用法可以参考程序示例：17-系统设置-18

**参数 2:** 字节数组转浮点数 单精度 4 字节 低位在前

**参数 3:** 空

例：全局数组 1={00 00 E0 3F}，全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：1.750000

**参数 2:** 字节数组转浮点数 单精度 4 字节 高位在前

**参数 3:** 空

例：全局数组 1={3F E0 00 00}，全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：1.750000

**参数 2:** 字节数组转浮点数 双精度 8 字节 低位在前

**参数 3:** 空

例：全局数组 1={00 00 00 00 00 00 FC 3F}，  
全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：1.750000

**参数 2:** 字节数组转浮点数 双精度 8 字节 高位在前

**参数 3:** 空

例：全局数组 1={3F FC 00 00 00 00 00 00}，  
全局数组 1 转全局浮点数 1 后，  
全局浮点数 1=字符串：1.750000

●此功能用法可以参考程序示例：17-系统设置-19

★参数 1：浮点数转数组/字符串设置

参数 2：浮点数转数组采用字节数组转换模式

参数 3：空

例：全局浮点数 1=12.450000  
浮点数转字节数组 单精度 4 字节 高位在前  
全局浮点数 1 转全局数组 1 后，  
全局数组 1={41 47 33 33}

●此功能用法可以参考程序示例：17-系统设置-20

参数 2：浮点数转数组采用字符串转换模式

参数 3：空

例：全局浮点数 1=1.750000  
全局浮点数 1 转全局数组 1 后，  
全局数组 1=字符串：1.750000

●此功能用法可以参考程序示例：17-系统设置-19

参数 2：浮点数转字节数组 单精度 4 字节 低位在前

参数 3：空

例：全局浮点数 1=12.450000  
全局浮点数 1 转全局数组 1 后，  
全局数组 1=16 进制：33 33 47 41

参数 2：浮点数转字节数组 单精度 4 字节 高位在前

参数 3：空

例：全局浮点数 1=12.450000  
全局浮点数 1 转全局数组 1 后，  
全局数组 1=16 进制：41 47 33 33

参数 2：浮点数转字节数组 双精度 8 字节 低位在前

参数 3：空

例：全局浮点数 1=12.450000  
全局浮点数 1 转全局数组 1 后，

全局数组 1=16 进制: 66 66 66 66 66 E6 28 40

**参数 2:** 浮点数转字节数组 双精度 8 字节 高位在前

**参数 3:** 空

例: 全局浮点数 1=12.450000

全局浮点数 1 转全局数组 1 后,

全局数组 1=16 进制: 40 28 E6 66 66 66 66 66

●此功能用法可以参考程序示例: 17-系统设置-20

**参数 2:** 浮点数转字符串场宽(总位数) 设置

**参数 3:** 手动输入数值

例: 全局浮点数 1=12.450000

**参数 3:** 10 , 即浮点数转字符串场宽(总位数) 设置为: 10

则处理后, 全局浮点数 1=012.450000

●此功能用法可以参考程序示例: 17-系统设置-21-1

**参数 2:** 浮点数转字符串小数点位数设置

**参数 3:** 手动输入数值

例: 全局浮点数 1=12.450000

**参数 3:** 5 , 即浮点数转字符串小数点位数设置为: 5

则处理后, 全局浮点数 1=12.45000

●此功能用法可以参考程序示例: 17-系统设置-21-2

★**参数 1:** 设置单片机 I/O 口

此功能仅供开发者使用, 即公司内部使用, 客户请忽略此功能。

★**参数 1:** 设定当前的日期与时间

**参数 2:** X 年 X 月 X 日 星期 X X 时 X 分 X 秒 (固定选项)

**参数 3:** 依次填写 日期与时间的数值, 用逗号做间隔

例: **参数 3:** 2024, 7, 15, 1, 14, 4, 20

则校准当前的时间为:

2024 年 7 月 15 日 星期 1 14 时 4 分 20 秒

●此功能用法可以参考程序示例: 17-系统设置-22

★**参数 1:** 网络参数设置

**参数 2:** 设置网卡参数(静态 IP) (固定选项)



**参数 3:** 依次填写 IP 地址、子网掩码、网关

例: **参数 3:** "192.168.1.111", "255.255.255.0", "192.168.1.1"

则设置网卡的静态 IP 为: 192.168.1.111

子网掩码: 255.255.255.0, 网关: 192.168.1.1

●此功能用法可以参考程序示例: 17-系统设置-23

★**参数 1:** 数据口参数设置

**参数 2:** 设置数据口波特率 (固定选项)

**参数 3:** 依次填写 数据口序号, 波特率数值

例: **参数 3:** 1, 115200

则设置数据口 1 的波特率为: 115200

●此功能用法可以参考程序示例: 17-系统设置-24

★**参数 1:** 开机启动参数设置(重启有效)

此功能仅供开发者使用, 即公司内部使用, 客户请忽略此功能。

(11) 类型: 数据口发数据

★**参数 1:** 选择数据口 数据口 1-数据口 10

**参数 2:** 发送 16 进制数据

**参数 3:** 手动输入 16 进制数据内容

例: **参数 2:** 数据口 1 **参数 3:** 01 02

则通过数据口 1 发送 16 进制数据: 01 02

**参数 2:** 发送字符串

**参数 3:** 手动输入字符串内容

例: **参数 2:** 数据口 1 **参数 3:** 123

则通过数据口 1 发送字符串: 123

**参数 2:** 发送表达式

**参数 3:** 手动输入表达式内容

例: **参数 2:** 数据口 1 **参数 3:** {03 04}

则通过数据口 1 发送 16 进制数据: 03 04

例: **参数 2:** 数据口 1 **参数 3:** "456"

则通过数据口 1 发送字符串: 456

**参数 2:** 发送格式化字符串

**参数 3:** 手动输入格式化字符串内容

例: 整数 0=25

**参数 2:** 数据口 1      **参数 3:** "目标数据: %d", 整数 0  
则数据口 1 发送字符串: 目标数据: 25

**参数 2:** 发送数据延迟等待

**参数 3:** 手动输入延迟等待时间, 单位: ms

例: **参数 2:** 数据口 1      **参数 3:** 2000

则数据口 1 发送数据延迟等待 2000ms

●此功能用法可以参考程序示例: 18-数据口发数据-1

## (12) 类型: 表达式

\*表达式是由数字、运算符、变量等组成的式子, 能明确数学关系并可通过计算得出结果。

★**参数 1:** 表达式计算

**参数 2:** 表达式计算

**参数 3:** 手动输入表达式及计算规则

例: 整数 0=10 ; 整数 1=20

**参数 3:** 数据口 1=(整数 0 + 整数 1)

则通过数据口 1 发送字符串: 30

★**参数 1:** 语句块开始:{

★**参数 1:** 语句块结束:}

\*语句块用来表达一个单独的语句结构。

\*语句块内的语句受到条件的影响, 只有条件满足, 语句块内部的语句才会执行。

\*语句块还可以用来区分不同的层次, 从而进行多级判断时候更方便。

\*鼠标在空白区域右键可以添加语句块。

★**参数 1:** 否则, 条件翻转

**参数 2:** 空

**参数 3:** 空

●此功能用法可以参考程序示例: 19-表达式-1

## (13) 类型: 函数处理

\*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。

\*函数的意义: 程序易读性, 同一个功能做成一个函数, 主程序



中再调用函数，可以大大增强主程序的易读性，避免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出现。

★参数 1：选择用户函数

参数 2：选择函数的序号

参数 3：可以为空，也可以是表达式

●此功能用法可以参考程序示例：20-函数处理-1

★参数 1：函数返回

参数 2：返回表达式

参数 3：手动输入表达式

●此功能用法可以参考程序示例：20-函数处理-2（函数 1）

参数 2：返回空，直接返回

参数 3：空

●此功能用法可以参考程序示例：20-函数处理-3（函数 1）

参数 2：跳转到函数开始

参数 3：空

●此功能用法可以参考程序示例：20-函数处理-4（函数 1）

★参数 1：指定用户函数

参数 2：表达式=表达式(表达式, ...)

参数 3：手动输入表达式

例：参数 3：整数 1( ) 即调用序号为整数 1 的函数

●此功能用法可以参考程序示例：20-函数处理-5（函数 1）

★参数 1：系统函数

参数 2：提取数组

参数 3：手动输入表达式

例：全局数组 1=16 进制数据：01 02 03 04 05 06

**参数 3:** 全局数组 2 = (全局数组 1, 2, 3)  
即提取全局数组 1 第 2 字节开始的 3 个字节, 赋值给全局数组 2  
则全局数组 2=16 进制数据: 03 04 05

**参数 2:** 提取数组尾部

**参数 3:** 手动输入表达式

例: 全局数组 3=16 进制数据: 02 03 04 05 06

**参数 3:** 全局数组 4 = (全局数组 3, 4)

即提取全局数组 3 尾部的 4 个字节, 赋值给全局数组 4

则全局数组 4=16 进制数据: 03 04 05 06

**参数 2:** 提取长度

**参数 3:** 手动输入表达式

例: 全局数组 5=16 进制数据: 03 04 05 06

**参数 3:** 全局数组 6 = ( 全局数组 5 )

即提取全局数组 5 的字节长度, 赋值给全局数组 6

则全局数组 6=16 进制数据: 04

**参数 2:** 获取芯片唯一识别码

**参数 3:** 手动输入表达式

例: **参数 3:** 全局数组 7 = ( )

则提取芯片唯一识别码赋值给全局数组 7

●此功能用法可以参考程序示例: 20-函数处理-6

- ★参数 1: IIC 相关系统函数 此功能未开发, 请忽略。
- ★参数 1: SPI 相关系统函数 此功能未开发, 请忽略。
- ★参数 1: 脉冲输出相关系统函数 此功能未开发, 请忽略。
- ★参数 1: SD 卡相关系统函数 此功能未开发, 请忽略。
- ★参数 1: 数学函数 此功能未开发, 请忽略。
- ★参数 1: 外部输入 此功能未开发, 请忽略。
- ★参数 1: 输出操作 此功能未开发, 请忽略。
- ★参数 1: 定时器操作 此功能未开发, 请忽略。
- ★参数 1: 字符串操作 此功能未开发, 请忽略。

#### (14) 类型: 通讯协议发命令

★参数 1: 信科 55 协议

**参数 2:** 闭合某路联机继电器

**参数 3:** 依次填写 数据口序号、地址几、第几路

例：参数 3： 2, 2, 1

则代表通过数据口 2 闭合联机继电器 地址 2 第 1 路

参数 2： 断开某路联机继电器

参数 3： 依次填写 数据口序号、地址几、第几路

例：参数 3： 2, 2, 1

则代表通过数据口 2 断开联机继电器 地址 2 第 1 路

参数 2： 翻转某路联机继电器

参数 3： 依次填写 数据口序号、地址几、第几路

例：参数 3： 2, 2, 2

则代表通过数据口 2 翻转联机继电器 地址 2 第 2 路

参数 2： 点动某路联机继电器

参数 3： 依次填写 数据口序号、地址几、第几路

例：参数 3： 2, 2, 3

则代表通过数据口 2 点动联机继电器 地址 2 第 3 路

默认点动时间 500ms

参数 2： 全部闭合联机继电器

参数 3： 依次填写 数据口序号、地址几

例：参数 3： 1, 2

则代表通过数据口 1 全部闭合联机继电器 地址 2

参数 2： 全部断开联机继电器

参数 3： 依次填写 数据口序号、地址几

例：参数 3： 2, 1

则代表通过数据口 2 全部断开联机继电器 地址 1

参数 2： 全部翻转联机继电器

参数 3： 依次填写 数据口序号、地址几

例：参数 3： 2, 2

则代表通过数据口 2 全部翻转联机继电器 地址 2

●此功能用法可以参考程序示例：21-通讯协议发命令-1

参数 2： 发送本机状态

参数 3： 手动输入数据口序号

例：参数 3： 1

则代表通过数据口 1 发送本机状态

参数 2： 发送公共数据量状态 此功能未开发，请忽略。

参数 2： 使用无回码命令（默认）

参数 3: 空

即操作联机继电器使用无回码命令

参数 2: 使用有回码命令

参数 3: 空

即操作联机继电器使用有回码命令

●此功能用法可以参考程序示例: 21-通讯协议发命令-2

参数 2: 使用协议回码更新本地联机开关量的状态

参数 3: 手动输入表达式, 此功能仅限 V6 版网络产品使用

例: 参数 3: UDPO, 1

则使用协议的回码 UDPO 更新本地联机开关量 1 的状态

●此功能用法可以参考程序示例: 24-网络数据-1

参数 2: 执行协议 并获取回码 (固件版本 9 或以上)

参数 3: 依次填写 协议内容、表达式

例: 参数 3: {55 01 20 00 00 00 01 77}, 数组 1

则执行 55 协议命令 {55 01 20 00 00 00 01 77}

将回码赋值给数组 1

●此功能用法可以参考程序示例: 21-通讯协议发命令-3

★参数 1: modbus 协议

参数 2: 功能码 01 读取单线圈

参数 3: 依次填写 数据口序号、模块地址、起始位置及数量

例: 参数 3: 1, 1, 0, 10

即通过数据口 1 发送 modbus 指令, 读取单线圈, 模块地址 1, 起始位置 0, 数量 10

则数据口 1 发送 16 进制数据: 01 01 00 00 00 0A BC 0D

参数 2: 功能码 02 读取外部输入

参数 3: 依次填写 数据口序号、模块地址、起始位置及数量

例: 参数 3: 1, 1, 0, 15

即通过数据口 1 发送 modbus 指令, 读取输入, 模块地址 1, 起始位置 0, 数量 15

则数据口 1 发送 16 进制数据: 01 02 00 00 00 0F 38 0E

参数 2: 功能码 03 读寄存器

参数 3: 依次填写 数据口序号、模块地址、寄存器地址及数

量

例：参数 3：1, 1, 10, 3

即通过数据口 1 发送 modbus 指令，读取寄存器，模块地址 1，寄存器地址 10，数量 3

则数据口 1 发送 16 进制数据：01 03 00 0A 00 03 25 C9

参数 2：功能码 05 控制单线圈

参数 3：依次填写 数据口序号、模块地址、位置及值

例：参数 3：1, 1, 5, 65280

即通过数据口 1 发送 modbus 指令，控制单线圈，模块地址 1，位置 5，值 65280

则数据口 1 发送 16 进制数据：01 05 00 05 FF 00 9C 3B

参数 2：功能码 06 写寄存器

参数 3：依次填写 数据口序号、模块地址、寄存器地址及值

例：参数 3：1, 1, 15, 100

即通过数据口 1 发送 modbus 指令，写单寄存器，模块地址 1，寄存器地址 15，值 100

则数据口 1 发送 16 进制数据：01 06 00 0F 00 64 B8 22

参数 2：执行协议 并获取回码（固件版本 9 或以上）

参数 3：依次填写 协议内容、表达式

例：参数 3：{01 0F 00 01 00 04 01 FF 43 16}，数据口 1

即执行 modbus 协议命令 {01 0F 00 01 00 04 01 FF 43 16}

并将回码赋值给数据口 1

●此功能用法可以参考程序示例：21-通讯协议发命令-4

### （15）类型：掉电存储

★参数 1：芯片内部存储器

参数 2：写缓冲区数组 占用空间数组长度+1

参数 3：依次填写数组表达式、缓冲区内部地址

例：参数 3：全局数组 0, 0

即写数组 全局数组 0 到缓冲区 缓冲区内部地址 0

参数 2：保存缓冲区到存储扇区

参数 3：手动输入存储扇区序号

例：参数 3：0 即保存缓冲区到存储扇区 0

参数 2：从存储扇区读取缓冲区

参数 3：手动输入存储扇区序号



例：参数 3： 0            即从存储扇区 0 读取到缓冲区

参数 2：读缓冲区 数组

参数 3：依次填写数组表达式、缓冲区内部地址

例：参数 3：全局数组 0, 0

即全局数组 0=读取缓冲区数组 缓冲区内部地址 0

参数 2：清零缓冲区 所有字节赋值为 0

参数 3：空                即清零缓冲区

●此功能用法可以参考程序示例：22-掉电存储-1

参数 2：写缓冲区字节 占用空间 1 字节

参数 3：依次填写 数组表达式、缓冲区内部地址

例：参数 3：数组 0, 0

即写字节 数组 0 到缓冲区 缓冲区内部地址 0

参数 2：读缓冲区 字节

参数 3：依次填写 数组表达式、缓冲区内部地址

例：参数 3：数组 1, 0

即数组 1=读取缓冲区字节 缓冲区内部地址 0

●此功能用法可以参考程序示例：22-掉电存储-2

参数 2：写缓冲区整数 占用空间 4 字节

参数 3：依次填写 整数表达式、缓冲区内部地址

例：参数 3：整数 0, 0

即写整数 整数 0 到缓冲区 缓冲区内部地址 0

参数 2：读缓冲区 整数

参数 3：依次填写 整数表达式、缓冲区内部地址

例：参数 3：整数 1, 0

即整数 1=读取缓冲区整数 缓冲区内部地址 0

●此功能用法可以参考程序示例：22-掉电存储-3

参数 2：写缓冲区单精浮点数 占用空间 4 字节

参数 3：依次填写 浮点数表达式、缓冲区内部地址

例：参数 3：浮点数 0, 0

即写单精度浮点数 浮点数 0 到缓冲区 缓冲区内部地址 0

参数 2：读缓冲区 单精度浮点数

参数 3：依次填写 浮点数表达式、缓冲区内部地址



例：参数 3：浮点数 1, 0

即浮点数 1=读取单精度浮点数 缓冲区内部地址 0

●此功能用法可以参考程序示例：22-掉电存储-4

参数 2：写缓冲区双精浮点数 占用空间 8 字节

参数 3：依次填写 浮点数表达式、缓冲区内部地址

例：参数 3：浮点数 0, 0

即写双精度浮点数 浮点数 0 到缓冲区 缓冲区内部地址 0

参数 2：读缓冲区 双精度浮点数

参数 3：依次填写 浮点数表达式、缓冲区内部地址

例：参数 3：浮点数 1, 0

即浮点数 1=读取双精度浮点数 缓冲区内部地址 0

●此功能用法可以参考程序示例：22-掉电存储-5

★参数 1：芯片外部铁电存储器 此功能未开发，请忽略。

(16) 类型：内部总线数据处理

此功能未开发，请忽略。

(17) 类型：操作 SD 卡

此功能未开发，请忽略。

(18) 类型：点阵液晶显示

此功能未开发，请忽略。

(19) 类型：其他输出数据

此功能未开发，请忽略。

(20) 类型：网络通讯

★参数 1：UDP 控制器

参数 2：UDP 控制器 绑定目标 IP 和端口

参数 3：依次填写 UDP 控制器序号、目标 IP 和端口

例：参数 3： 0, "192.168.1.110", 6000

则代表设置 UDP 控制器 0 连接（绑定）目标 IP：192.168.1.110

目标端口：6000

**参数 2:** 发送 UDP 数据

**参数 3:** 依次填写 UDP 控制器序号、数据内容

例: **参数 3:** 0, {55 01 20 00 00 00 01 77}

则代表 UDP 控制器 0 发送 16 进制数据: 55 01 20 00 00 00 01 77

**参数 2:** UDP 控制器 绑定本地端口(作为客户端 IP=本机)

**参数 3:** 依次填写 UDP 控制器序号、本地端口

例: **参数 3:** 1, 3001

则代表 UDP 控制器 1 绑定本地端口 3001 作为客户端

●此功能用法可以参考程序示例: 24-网络通讯-4

**参数 2:** UDP 控制器 绑定本地端口(作为服务器 IP=0)

**参数 3:** 依次填写 UDP 控制器序号、本地端口

例: **参数 3:** 0, 3000

则代表 UDP 控制器 0 绑定本地端口 3000 作为服务器

●此功能用法可以参考程序示例: 24-网络通讯-1

★**参数 1:** TCP 控制器

**参数 2:** TCP 控制器连接目标 IP 和端口

**参数 3:** 依次填写 TCP 控制器序号、目标 IP 和端口

例: **参数 3:** 0, "192.168.1.110", 6000

则代表设置 TCP 控制器 0 连接目标 IP: 192.168.1.110

目标端口: 6000

**参数 2:** 发送 TCP 数据

**参数 3:** 依次填写 TCP 控制器序号、数据内容

例: **参数 3:** 0, {55 01 20 00 00 00 01 77}

则代表 TCP 控制器 0 发送 16 进制数据: 55 01 20 00 00 00 01 77

**参数 2:** TCP 控制器断开连接

**参数 3:** TCP 控制器序号

例: **参数 3:** 0

则代表 TCP 控制器 0 断开连接

●此功能用法可以参考程序示例: 24-网络通讯-2

★**参数 1:** 网络参数

**参数 2:** 修改网卡当前参数(静态 IP)

**参数 3:** 依次填写 IP 地址、子网掩码和网关

例: **参数 3:** "192.168.1.111", "255.255.255.0", "192.168.1.1"  
则代表设置网卡当前参数 IP 地址: 192.168.1.111  
子网掩码: 255.255.255.0 网关: 192.168.1.1  
此参数掉电后会丢失, 仅代表当前参数。

**参数 2:** 重新设置网卡上电默认参数(静态 IP)

**参数 3:** 依次填写 IP 地址、子网掩码和网关

例: **参数 3:** "192.168.1.110", "255.255.255.0", "192.168.1.1"  
则代表设置网卡上电参数 IP 地址: 192.168.1.110  
子网掩码: 255.255.255.0 网关: 192.168.1.1  
写完程序, 需重新上电, 参数有效。此参数掉电后不会丢失。

**参数 2:** 获取网卡网络参数(字符串)

**参数 3:** 手动输入表达式

例: **参数 3:** 数据口 1

则代表获取网络参数字符串, 并通过数据口 1 发送出去

**参数 2:** 获取网卡物理地址(字符串)

**参数 3:** 手动输入表达式

例: **参数 3:** 数组 1

则代表获取网络物理地址 字符串赋值给数组 1

**参数 2:** 获取 TCP 控制器数量

**参数 3:** 手动输入表达式

例: **参数 3:** 数据口 1

则代表获取 TCP 控制器的数量, 并通过数据口 1 发送出去

**参数 2:** 获取 UDP 控制器数量

**参数 3:** 手动输入表达式

例: **参数 3:** 数据口 1

则代表获取 UDP 控制器的数量, 并通过数据口 1 发送出去

**参数 2:** 获取 TCP 服务端口数量

**参数 3:** 手动输入表达式

例: **参数 3:** 数据口 1

则代表获取 TCP 服务端口的数量, 并通过数据口 1 发送出去

●此功能用法可以参考程序示例: 24-网络通讯-5

★**参数 1:** TCP 服务端口

**参数 2:** 打开 TCP 服务端口

**参数 3:** 手动输入要打开的服务端口

例：参数 3：3000

则代表打开 TCP 服务端口 3000

参数 2：关闭 TCP 服务端口

参数 3：手动输入要关闭的服务端口

例：参数 3：3000

则代表关闭 TCP 服务端口 3000

参数 2：关闭本地端口对应的连接

参数 3：手动输入本地端口

例：参数 3：3000

则代表关闭 TCP 服务端口 3000 对应的连接

●此功能用法可以参考程序示例：24-网络通讯-6

参数 2：提取 TCP 服务端口接收到的数据

参数 3：依次填写 TCP 服务端口、表达式

例：参数 3：3000，数组 1

则代表提取 TCP 服务端口 3000 接收到的数据，并赋值给数组 1

参数 2：提取 TCP 服务端口接收到数据的对方 IP（数组）

参数 3：依次填写 TCP 服务端口、表达式

例：参数 3：3000，数组 2

则代表提取 TCP 服务端口 3000 接收到数据的对方 IP 字符串，并赋值给数组 2

参数 2：提取 TCP 服务端口接收到数据的对方 IP 整数

参数 3：依次填写 TCP 服务端口、表达式

例：参数 3：3000，整数 1

则代表提取 TCP 服务端口 3000 接收到数据的对方 IP 整数，并赋值给整数 1

参数 2：提取 TCP 服务端口接收到数据的对方端口

参数 3：依次填写 TCP 服务端口、表达式

例：参数 3：3000，整数 2

则代表提取 TCP 服务端口 3000 接收到数据的对方端口，并赋值给整数 2

参数 2：提取 TCP 服务端口接收到数据的对方客户端的指针

参数 3：依次填写 TCP 服务端口、表达式

例：参数 3：3000，整数 0

则代表提取 TCP 服务端口 3000 接收到数据的对方客户端的指针，

并赋值给整数 0

**参数 2:** 根据指针向客户端发送数据

**参数 3:** 依次填写 指针、数据内容

例: **参数 3:** 整数 0, {01 02 03}

则代表根据指针 整数 0 客户端, 发送 16 进制数据: 01 02 03

●此功能用法可以参考程序示例: 24-网络通讯-7

**参数 2:** 服务端口提取临时数据的客户端并发送数据

**参数 3:** 依次填写 TCP 服务端口、数据内容

例: **参数 3:** 3000, {01 02}

则代表服务端口 3000 提取临时数据的客户端

并发送 16 进制数据: 01 02

**参数 2:** 服务端口向最近的客户端发送数据

**参数 3:** 依次填写 TCP 服务端口、数据内容

例: **参数 3:** 3000, {03 04}

则代表服务端口 3000 向最近的客户端, 发送 16 进制数据: 03 04

**参数 2:** 服务端口向所有客户端发送数据

**参数 3:** 依次填写 TCP 服务端口、表达式

例: **参数 3:** 3000, {05 06}

则代表服务端口 3000 向所有客户端, 发送 16 进制数据: 05 06

**参数 2:** 服务端口向某一个客户端发送数据(ip, 端口)

**参数 3:** 依次填写 TCP 服务端口、IP 地址、端口、数据内容

例: **参数 3:** 3000, "192.168.1.7", 4000, {07 08}

则代表服务端口 3000 向客户端 IP: 192.167.1.7, 端口: 4000

发送 16 进制数据: 07 08

●此功能用法可以参考程序示例: 24-网络通讯-8