

3.0 内核工具软件使用说明及解析

V1.5

信科电子



本文档适用于信科电子出品的
3.0 内核工具软件的使用
(适用于 V1.317 及以上版本)

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·····	5
(2) 类型：输出状态变化·····	6
(3) 类型：定时器·····	6
(4) 类型：位变量·····	7
(5) 类型：全局整数（全局整数/全局变量）·····	8
(6) 类型：固定输入·····	9
(7) 类型：无线遥控器·····	9
(8) 类型：时间设定·····	9
(9) 类型：日期设定·····	10
(10) 类型：联机开关量·····	11
(11) 类型：模拟量输入·····	11
(12) 类型：全局数组简单判断·····	11
(13) 类型：全局数组高级判断·····	12
(14) 类型：局部数组·····	13
(15) 类型：局部整数·····	13
(16) 类型：表达式/函数判断（118）·····	13
(17) 类型：数据口接收到数据（118）·····	14

2、输出结果

(1) 类型：系统软件资源 (118)	16
(2) 类型：输出操作.....	27
(3) 类型：定时器.....	30
(4) 类型：位变量.....	30
(5) 类型：全局整数（原内部变量）	31
(6) 类型：数据口发数据 (118)	36
(7) 类型：系统简单设置.....	37
(8) 类型：系统特殊设置.....	38
(9) 类型：通讯参数设置.....	40
(10) 类型：批量操作内部资源.....	40
(11) 类型：调用函数.....	41
(12) 类型：操作联机继电器模块（发送命令）	41
(13) 类型：全局数组简单处理.....	42
(14) 类型：全局数组复杂处理.....	47
(15) 类型：脉冲调制输出.....	50
(16) 类型：局部整数.....	50
(17) 类型：局部数组.....	51
(18) 类型：红外发射.....	52
(19) 类型：表达式/用户函数 (118)	53
(20) 类型：局部数组 (118)	57
(21) 类型：全局数组 (118)	58
(22) 类型：系统高级设置（版本 118 慎用）	59
(23) 类型：临时数据包设置 (118)	59

一、概述

3.0 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

线程功能

软件新增线程功能，支持多线程同时运行。线程阻塞延迟，仅限于本线程内有效，不会影响其他线程以及全局运行。

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

3.0 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：

只有条件满足，输出结果的程序才会执行。输出结果可以是单一结果，也可以是多个结果。

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



多个输出结果



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）。

***系统特殊设置里面可以设置输入连击和长按的时间常数。**

★参数 1：输入 1-输入 200

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

例：参数 1：输入 2 参数 2：断开到闭合

则代表输入 2 断开到闭合 为条件成立

例：参数 1：输入 7 参数 2：长按

则代表输入 7 长按（按下后 500ms 内未抬起）为条件成立

●以上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：连续按下多次 全局整数 1 代表次数

参数 3：空

例：参数 1：输入 8 参数 2：连续按下多次

如果 全局整数 1=3

则代表输入 8 在 500ms 内连续按下 3 次 为条件成立

●以上功能用法可以参考程序示例：1-外部输入-输出操作-1

（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：输出 1-输出 200

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

例：参数 1：输出 1 参数 2：闭合

则代表输出 1 闭合 为条件成立

例：参数 1：输出 5 参数 2：状态翻转

则代表输出 5 状态翻转 为条件成立

●以上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 根据需要进行选择，参数 3 手动输入定时器的值。

定时器的值为 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 1。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 1 或者停止定时。

*如果定时器处在单位时间减 1 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 1 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：定时器 1-定时器 199

用表达式方式写定时器，支持：定时器 0-定时器 199

参数 2：大于/小于/等于

参数 3：手动输入数值（0 - 2147483647）

例：参数 1：定时器 1 参数 2：大于 参数 3：500
则代表定时器 1 大于 500 为条件成立

参数 2：大于全局整数/小于全局整数/等于全局整数

参数 3：手动输入全局整数的序号

例：参数 1：定时器 2 参数 2：小于全局整数 参数 3：1
则代表定时器 2 小于全局整数 1 为条件成立

参数 2：大于整数/小于整数/等于整数

参数 3：手动输入整数的序号

例：参数 1：定时器 3 参数 2：等于整数 参数 3：0
则代表定时器 3 等于整数 0 为条件成立

参数 2：减到 0

参数 3：空

例：参数 1：定时器 3 参数 2：减到 0
则代表定时器 3 减到 0 为条件成立

●以上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: 位变量 1-位变量 199

用表达式方式写位变量，支持：位变量 0-位变量 199

参数 2: =0/=1

参数 3: 空

例：参数 1: 位变量 1 参数 2: =1

则代表位变量 1 =1 为条件成立

参数 2: 0 到 1 跳变/1 到 0 跳变/状态翻转

参数 3: 空

例：参数 1: 位变量 1 参数 2: 0 到 1 跳变

则代表位变量 1, 0 到 1 跳变 为条件成立

●以上功能用法可以参考程序示例：4-位变量-1

(5) 类型：全局整数（原内部变量）

全局整数可以作为输入，也可以作为输出。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，我们就可以用到全局整数功能。

*全局整数取值范围：-2147483648 — 2147483647

*全局整数作为输入使用，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。

*全局整数作为输出使用，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。

*全局整数功能，全局有效。

★参数 1: 全局整数 1-全局整数 199

用表达式方式写全局整数，支持：全局整数 0-全局整数 199

参数 2: 大于参数 3 的值/小于参数 3 的值/等于参数 3 的值

参数 3: 手动输入数值

例：参数 1: 全局整数 1 参数 2: 大于参数 3 的值 参数 3: 50

则代表全局整数 1 > 50 为条件成立

参数 2: 大于全局整数的值/小于全局整数的值/等于全局整数的值

参数 3: 手动输入全局整数的序号

例: **参数 1:** 全局整数 3 **参数 2:** 小于全局整数的值 **参数 3:** 2
则代表全局整数 3 < 全局整数 2 为条件成立

参数 2: 位判断

参数 3: 手动输入要判断的位数

例: **参数 1:** 全局整数 5 **参数 3:** 3
则代表全局整数 5 的第 3 位为 1 为条件成立;
全局整数 5 的第 3 位为 0 为条件不成立。

● **以上功能用法可以参考程序示例: 5-全局整数-1**

(6) 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。

● **以上功能用法可以参考程序示例: 6-固定输入-1**

(7) 类型: 无线遥控器

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:** 无线接收模块 315M

参数 2: 无

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: 无线参数:4570 无线数据:1612229

● **以上功能用法可以参考程序示例: 7-无线遥控器-1**

★ **参数 1:** 接收到万能红外发射对码信号

参数 2: 无

此功能未开发, 请忽略。

(8) 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。

★ **参数 1:** 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。

参数 2: 固定格式 XX 时 XX 分 XX 秒

参数 3: 手动输入时间

例：**参数 3:** 11 时 10 分 00 秒

则代表进入时刻 11 时 10 分 00 秒时 为条件成立；

★参数 1: 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。

参数 2: 固定格式 XX 时 XX 分 XX 秒

参数 3: 手动输入时间

例：**参数 1:** 大于 **参数 3:** 11 时 10 分 10 秒

则代表大于 11 时 10 分 10 秒时 为条件成立

●以上功能用法可以参考程序示例：9-时间设定-1

（9）类型：日期设定

日期设定指的是对当前日期进行判断。

★参数 1: 年设定/月设定/日设定/星期设定

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 3 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。

参数 2: 进入时刻

参数 3: 选择对应要设置的年/月/日/星期

例：**参数 1:** 年设定 **参数 3:** 2026 年

则代表日期从 2025 年 12 点 59 分 59 秒过度到
2026 年 0 时 0 分 0 秒 为条件成立。

参数 2: 大于/小于/等于 根据数字的大小作出判断

参数 3: 选择对应要设置的年/月/日/星期

例：**参数 1:** 星期设定 **参数 2:** 等于 **参数 3:** 星期 3

则代表当前星期 = 星期 3 时 条件成立

●以上功能用法可以参考程序示例：10-日期设定-1

10-日期设定-2

（10）类型：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：**地址 0-地址 254

参数 2：第 1 路-第 32 路

参数 3：断开/闭合/断开到闭合/闭合到断开/状态翻转

例：**参数 1：**地址 2 **参数 2：**第 1 路 **参数 3：**闭合

则联机开关量地址 2 的第 1 路闭合状态 为条件成立

例：**参数 1：**地址 2 **参数 2：**第 1 路 **参数 3：**状态翻转

则联机开关量地址 2 的第 1 路状态翻转 为条件成立

●**以上功能用法可以参考程序示例：11-联机开关量-1**

（11）类型：模拟量输入

此功能未开发，请忽略。

（12）类型：全局数组简单判断

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

*一个全局数组中可以存储若干字节，一个模块中含有若个全局数组。具体可以通过读取模块硬件信息来查看。

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

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

*全局数组功能，全局有效。

★参数 1: 全局数组 1-全局数组 100

用表达式方式写全局数组，支持：全局数组 0-全局数组 199

参数 2: 字节数判断

参数 3: 0 个字节-199 个字节

例：参数 1: 全局数组 1 参数 3: 6

则全局数组 1 的字节数等于 6 字节 为条件成立

参数 2: 校验判断

参数 3: 和校验

例：参数 1: 全局数组 1

则全局数组 1 和校验成功 为条件成立

参数 3: CRC16 校验

例：参数 1: 全局数组 1

则全局数组 1 CRC16 校验成功 为条件成立

参数 2: 等于其他全局数组的内容

参数 3: 全局数组 0-全局数组 199

例：参数 1: 全局数组 1 参数 3: 全局数组 2

则全局数组 1 等于全局数组 2 的内容 为条件成立

参数 2: 查找其他全局数组的内容

参数 3: 全局数组 0-全局数组 199

例：参数 1: 全局数组 1 参数 3: 全局数组 3

则全局数组 1 包含全局数组 3 的内容 为条件成立

●以上功能用法可以参考程序示例：13-全局数组简单判断-1

(13) 类型：全局数组高级判断

★参数 1: 全局数组 1-全局数组 100

用表达式方式写全局数组，支持：全局数组 0-全局数组 199

参数 2: 等于某几个 16 进制数据

参数 3: 手动输入 16 进制数据

例：参数 1: 全局数组 1 参数 3: 01 02 03

则全局数组 1 等于 16 进制数据：01 02 03 为条件成立

参数 2: 等于某几个字符串

参数 3: 手动输入字符串

例: **参数 1:** 全局数组 1 **参数 3:** abc

则全局数组 1 等于字符串: abc 为条件成立

参数 2: 包含某几个 16 进制数据

参数 3: 手动输入 16 进制数据

例: **参数 1:** 全局数组 1 **参数 3:** 04 05 06

则全局数组 1 包含 16 进制数据: 04 05 06 为条件成立

参数 2: 包含某几个字符串

参数 3: 手动输入字符串

例: **参数 1:** 全局数组 1 **参数 3:** 789

则全局数组 1 包含字符串: 789 为条件成立

● **以上功能用法可以参考程序示例: 13-全局数组高级判断-2**

(14) 类型: 局部数组

*局部数组功能同全局数组功能类似, 可参考全局数组说明。

*局部数组 0-局部数组 199。

*局部数组功能, 局部有效。

● **以上功能用法可以参考程序示例: 22-局部数组-1;**

22 局部数组-2; 22 局部数组-3。

(15) 类型: 局部整数

*局部整数功能同全局整数功能类似, 可参考全局整数说明。

*局部整数 0-局部整数 199。

*局部整数功能, 局部有效。

● **以上功能用法可以参考程序示例: 12-局部整数-1;**

12-局部整数-2; 12-局部整数-3;

12-局部整数-4; 12-局部整数-5。

(16) 类型: 表达式/函数判断 (118)

***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: 数据口 1={01}

则数据口 1 接收到的数据等于 16 进制数据: 01 为条件成立

●以上功能用法可以参考程序示例: 14-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: 函数 1-函数 100

参数 3: 手动输入表达式

例: 参数 2: 函数 1 参数 3: {03} = ()

则函数 1 返回的数据等于 16 进制数据: 03 为条件成立

●以上功能用法可以参考程序示例: 14-表达式-5

(17) 类型: 数据口接收到数据 (118)

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: 需要注意字节与字节之间使用空格作间隔。

*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: 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: 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/等于表达式

参数 3：手动输入数据内容或者表达式

例：参数 1：数据口 1 参数 3：01 02

则数据口 1 接收到的数据等于 16 进制数据：01 02 为条件成立

例：参数 1：数据口 1 参数 3：123

则数据口 1 接收到的数据等于字符串：123 为条件成立

例：参数 1：数据口 1 参数 3：数组 1

则数据口 1 接收到的数据等于数组 1 的内容 为条件成立

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 3：5

则数据口 1 接收到的数据等于 5 个字节 为条件成立

●以上功能用法可以参考程序示例：15-数据口接收到数据-1

参数 2：包含

参数 3：手动输入表达式

例：参数 1：数据口 1 参数 3：{01 02}

则数据口 1 接收到的数据包含 16 进制数据：01 02 为条件成立

参数 2：位置等于

参数 3：手动输入表达式

例：参数 1：数据口 1 参数 3：2，数组 2

则数据口 1 接收到的数据位置 2 等于数组 2 为条件成立

●以上功能用法可以参考程序示例：15-数据口接收到数据-2

参数 2：和校验判断（V126）

参数 3：空

例：参数 1：数据口 1

则数据口 1 接收到的数据和校验成功 为条件成立

参数 2：CRC16 校验判断（V126）

参数 3：空

例：参数 1：数据口 1

则数据口 1 接收到的数据 CRC16 校验成功 为条件成立

●以上功能用法可以参考程序示例：15-数据口接收到数据-3

2、输出结果

(1) 类型：系统软件资源（118）

★参数 1：系统运行

参数 2：运行间隔

参数 3：手动输入数值

例：参数 3：6000 则程序间隔 6000 毫秒运行一次

●以上功能用法可以参考程序示例：23-系统软件资源-1

参数 2：线程阻塞延迟

线程阻塞延迟：是指线程因等待特定条件满足而处于阻塞状态的时间长度。从线程进入阻塞状态到离开该状态所经历的这段时间，就是线程阻塞延迟。线程阻塞延迟，仅限于本线程内有效，不会影响其他线程以及全局运行。

参数 3：手动输入数值

例：参数 3：500 则线程阻塞延迟 500 毫秒

●以上功能用法可以参考程序示例：23-系统软件资源-2

参数 2：系统阻塞延迟

系统阻塞延迟：是指单片机处于休眠状态，从阻塞开始到阻塞解除所经历的时间。在此期间，程序无法继续执行有效操作，直到阻塞解除才会被唤醒并继续运行，这段等待的时间就是系统阻塞延迟。系统阻塞延迟，全局有效。

参数 3：手动输入数值

例：参数 3：5000 则系统阻塞延迟 5000 毫秒

●以上功能用法可以参考程序示例：23-系统软件资源-3

参数 2：获取唯一识别码

参数 3：手动输入表达式

例：参数 3：数据口 1 则获取芯片唯一识别码赋值给数据口 1

例：参数 3：数组 1 则获取芯片唯一识别码赋值给数组 1

●以上功能用法可以参考程序示例：23-系统软件资源-1

参数 2: 设置全局数组边界 此选项为无效功能, 请忽略。

参数 2: 设置局部数组边界 此选项为无效功能, 请忽略。

参数 2: 启动线程

参数 3: 手动输入数值

例: 参数 3: 1 则启动线程 1

参数 2: 暂停线程

参数 3: 手动输入数值

例: 参数 3: 4 则暂停线程 4

参数 2: 停止线程

参数 3: 手动输入数值

例: 参数 3: 2 则停止线程 2

● 以上功能用法可以参考程序示例: 23-系统软件资源-2

★ 参数 1: 数组相关运算

参数 2: 提取数组内容(起始 0 位置 数量)

参数 3: 手动输入表达式

例: 参数 3: 全局数组 1, 2, 5, 全局数组 10

则提取全局数组 1 的内容, 位置 2 开始的 5 个字节, 再将结果赋值给全局数组 10 (位置的起始位为 0, 依次分别是: 第 0 位、第 1 位、第 2 位、第 3 位……)

如 果: 全局数组 1 = {01 02 03 04 05 06 07 08}

处理后: 全局数组 10 = {03 04 05 06 07}

参数 2: 提取数组尾部

参数 3: 手动输入表达式

例: 参数 3: 全局数组 2, 1, 4, 全局数组 11

则提取全局数组 2 的内容, 尾部位置 1 开始的 4 个字节, 再将结果赋值给全局数组 11 (位置的起始位为 0, 依次分别是: 第 0 位、第 1 位、第 2 位、第 3 位……)

如 果: 全局数组 2 = {01 02 03 04 05 06}

处理后: 全局数组 11 = {02 03 04 05}

参数 2: 提取数组长度

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 3, 整数 0

则提取全局数组 3 的长度, 赋值给整数 0

如 果: 全局数组 3 = {01 02 03 04 05 06}

处理后: 整数 0 = 6

参数 2: 提取数组非连续字节

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 5, 全局数组 4, 0, 1, 5, 8, 9

则全局数组 5= 提取 全局数组 4 的第 0, 1, 5, 8, 9 字节

如 果: 全局数组 4 = {01 02 03 04 05 06 07 08 09 10}

处理后: 全局数组 5 = {01 02 06 09 10}

●以上功能用法可以参考程序示例: 23-系统软件资源-4

参数 2: 数组中插入数组

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 1, 2, 全局数组 3, 全局数组 2

则全局数组 1 的第 2 字节处插入全局数组 3, 再将结果赋值给全局数组 2

如 果: 全局数组 1 = {01 02 03 04 05} 全局数组 3 = {0A 0B}

处理后: 全局数组 2 = {01 02 0A 0B 03 04 05}

●以上功能用法可以参考程序示例: 23-系统软件资源-5

参数 2: 删除数组连续字节(起始 0 位置 数量)

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 1, 2, 3, 全局数组 2

则从全局数组 1 的第 2 字节开始删除 3 个字节, 再将结果赋值给全局数组 2

如 果: 全局数组 1 = {01 02 03 04 05 06 07 08}

处理后: 全局数组 2 = {01 02 06 07 08}

参数 2: 删除数组非连续字节

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 4, 全局数组 3, 0, 3, 5, 9

则全局数组 4= 把 全局数组 3 删除第 0, 3, 5, 9 字节后的结果
如 果：全局数组 3 = {01 02 03 04 05 06 07 08 09 10}
处理后：全局数组 4 = {02 03 05 07 08 09}

参数 2：删除数组末尾字节（数量）

参数 3：手动输入表达式

例：**参数 3：**全局数组 5, 3

则删除 全局数组 5 尾部的 3 个字节

如 果：全局数组 5 = {01 02 03 04 05 06}

处理后：全局数组 5 = {01 02 03}

●以上功能用法可以参考程序示例：23-系统软件资源-6

参数 2：尾部添加累加和校验

参数 3：手动输入表达式

例：**参数 3：**全局数组 1 则全局数组 1 尾部添加和校验字节

如 果：全局数组 1 = {01 01 00 00 00 40}

处理后：全局数组 1 = {01 01 00 00 00 40 42}

参数 2：尾部添加异或和校验

参数 3：手动输入表达式

例：**参数 3：**全局数组 2 则全局数组 2 尾部添加异或和校验字节

如 果：全局数组 2 = {01 02 03 04 05}

处理后：全局数组 2 = {01 02 03 04 05 01}

参数 2：尾部添加 CRC16 校验

参数 3：手动输入表达式

例：**参数 3：**全局数组 3 则全局数组 3 尾部添加 CRC16 校验字节

如 果：全局数组 3 = {01 01 00 00 00 40}

处理后：全局数组 3 = {01 01 00 00 00 40 3D FA}

参数 2：尾部添加 CRC8 校验

参数 3：手动输入表达式

例：**参数 3：**全局数组 4 则全局数组 4 尾部添加 CRC8 校验字节

如 果：全局数组 4 = {01 02 03 04}

处理后：全局数组 4 = {01 02 03 04 E3}

●以上功能用法可以参考程序示例：23-系统软件资源-7

参数 2: 字节倒序

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 1 则全局数组 1 字节倒序

如 果: 全局数组 1 = {01 02 03 04 05 06 07 08 09}

处理后: 全局数组 1 = {09 08 07 06 05 04 03 02 01}

参数 2: 每个字节前后四位互换

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 2 则全局数组 2 每个字节前后四位互换

如 果: 全局数组 2 = {01 02 03 04 05 06}

处理后: 全局数组 2 = {10 20 30 40 50 60}

参数 2: 每个字节的位倒序

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 3 则全局数组 3 每个字节的位倒序

如 果: 全局数组 3 = {01 01 00 00 00 40}

处理后: 全局数组 3 = {80 80 00 00 00 02}

参数 2: 字节转化为 16 进制编辑字符串

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 4

则全局数组 4 字节转化为 16 进制编辑字符串

如 果: 全局数组 4 = 16 进制: 50 51 52 53 54

处理后: 全局数组 4 = 字符串: 50 51 52 53 54

参数 2: 固定长度, 不足补 0, 多余删除

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 6, 10

则全局数组 6 固定长度为 10, 不足补 0, 多余删除

如 果: 全局数组 6 = {06 01 02 03 04 05}

处理后: 全局数组 6 = {06 01 02 03 04 05 00 00 00 00}

如 果: 全局数组 6 = {06 07 08 09 0A 01 02 03 04 05 06 07}

处理后: 全局数组 6 = {06 07 08 09 0A 01 02 03 04 05}

● 以上功能用法可以参考程序示例: 23-系统软件资源-8

★参数 1: 字符串相关运算

参数 2: 格式化字符串

参数 3: 手动输入表达式

例: 参数 3: 数据口 1, "目标数据%d", 整数 0

如果整数 0=156 , 则数据口 1 发送格式化字符串: 目标数据 156

参数 2: 转化为小写

参数 3: 手动输入表达式

例: 参数 3: 全局数组 1, 数据口 1

则把全局数组 1 字符串转为小写, 再通过数据口 1 发送出去

例: 参数 3: 全局数组 1, 全局数组 2

则把全局数组 1 字符串转为小写, 再将结果赋值给全局数组 2

参数 2: 转化为大写

参数 3: 手动输入表达式

例: 参数 3: 全局数组 1, 数据口 1

则把全局数组 1 字符串转为大写, 再通过数据口 1 发送出去

例: 参数 3: 全局数组 1, 全局数组 2

则把全局数组 1 字符串转为大写, 再将结果赋值给全局数组 2

●以上功能用法可以参考程序示例: 23-系统软件资源-9

参数 2: 提取整数

参数 3: 手动输入表达式

例: 参数 3: 全局数组 1, 2, 整数 0

则提取字符串 全局数组 1 中的第 2 个整数, 赋值给整数 0

如 果: 全局数组 1 = "12 34 56 78"

处理后: 整数 0 = 56

参数 2: 提取浮点数

参数 3: 手动输入表达式

例: 参数 3: 全局数组 2, 2, 浮点数 0

则提取浮点数 全局数组 2 中的第 2 个浮点数, 赋值给浮点数 0

如 果: 全局数组 2 = "12.34 56.78 98.56"

处理后: 浮点数 0 = 98.560000

●以上功能用法可以参考程序示例: 23-系统软件资源-10

★参数 1: 整数相关运算

参数 2: 等于

参数 3: 手动输入表达式

例: 参数 3: 整数 1, 32 则整数 1=32

参数 2: ←移

参数 3: 手动输入表达式

例: 参数 3: 整数 1, 2 则把整数 1 左移 2 位

如 果: 整数 1 = 32

处理后: 整数 1 = 128

处理过程: 10 进制 32, 转为 2 进制: 100000, 左移 2 位后,
等于 2 进制: 10000000, 再转为 10 进制, 即: 128

参数 2: →移

参数 3: 手动输入表达式

例: 参数 3: 整数 2, 3 则把整数 2 右移 3 位

如 果: 整数 2 = 104

处理后: 整数 2 = 13

处理过程: 10 进制 104, 转为 2 进制: 1101000, 右移 3 位后,
等于 2 进制: 1101, 再转为 10 进制, 即: 13

●以上功能用法可以参考程序示例: 23-系统软件资源-11

参数 2: 与运算

参数 3: 手动输入表达式

例: 参数 3: 全局整数 2, 全局整数 1, 4

则全局整数 2 = 全局整数 1 与 4 的运算结果

参数 2: 或运算

参数 3: 手动输入表达式

例: 参数 3: 全局整数 2, 全局整数 1, 4

则全局整数 2 = 全局整数 1 或 4 的运算结果

参数 2: 异或运算

参数 3: 手动输入表达式

例: 参数 3: 全局整数 2, 全局整数 1, 4

则全局整数 2 = 全局整数 1 异或 4 的运算结果

●以上功能用法可以参考程序示例：23-系统软件资源-12

参数 2：取反运算

参数 3：手动输入表达式

例：**参数 3：**全局整数 1，全局整数 2

则全局整数 1 取反 然后赋值给全局整数 2

参数 2：位取反 (bit0 - bit31)

参数 3：手动输入表达式

例：**参数 3：**全局整数 1，3

则把全局整数 1 的第 3 位 取反

如 果：全局整数 1 = 29

处理后：全局整数 1 = 21

处理过程：10 进制 29，转为 2 进制：11101，第 3 位取反后，
等于 2 进制：10101，再转为 10 进制，即：21

参数 2：写整数位

参数 3：手动输入表达式

例：**参数 3：**全局整数 1，2，0

则把全局整数 1 的第 2 位 设置为 0

如 果：全局整数 1 = 29

处理后：全局整数 1 = 25

处理过程：10 进制 29，转为 2 进制：11101，第 2 位设置为 0 后，
等于 2 进制：11001，再转为 10 进制，即：25

参数 2：读整数位

参数 3：手动输入表达式

例：**参数 3：**全局整数 2，全局整数 1，1

则全局整数 2 = 全局整数 1 的第 1 位

如 果：全局整数 1 = 29

处理后：全局整数 2 = 0

处理过程：10 进制 29，转为 2 进制：11101，第 1 位为 0

●以上功能用法可以参考程序示例：23-系统软件资源-13

★参数 1: 通讯协议命令(信科 55 协议)

参数 2: 使能本机数据口检测 55 协议(0 或者无代表全部)

参数 3: 手动输入数据口的序号

例: 参数 3: 1 则使能本机数据口 1 检测 55 协议

参数 2: 失能本机数据口检测 55 协议(0 或者无代表全部)

参数 3: 手动输入数据口的序号

例: 参数 3: 2 则失能本机数据口 2 检测 55 协议

●以上功能用法可以参考程序示例: 23-系统软件资源-14

参数 2: 操作联机继电器使用无回码命令

参数 3: 空

●以上功能用法可以参考程序示例: 23-系统软件资源-15

参数 2: 操作联机继电器使用有回码命令

参数 3: 空

●以上功能用法可以参考程序示例: 23-系统软件资源-16

参数 2: 发送本机状态

参数 3: 手动输入数据口的序号

例: 参数 3: 1 则通过数据口 1 发送本机状态

●以上功能用法可以参考程序示例: 23-系统软件资源-17

参数 2: 查询联机状态

参数 3: 手动输入表达式

例: 参数 3: 2, 2 则通过数据口 2 查询联机地址 2 的状态

●以上功能用法可以参考程序示例: 23-系统软件资源-16

参数 2: 断开

参数 3: 手动输入表达式

例: 参数 3: 2, 2, 1

则通过数据口 2 操作联机 断开 地址 2 第 1 路

参数 2: 闭合

参数 3: 手动输入表达式

例：参数 3：2， 2， 1

则通过数据口 2 操作联机 闭合 地址 2 第 1 路

参数 2：翻转

参数 3：手动输入表达式

例：参数 3：2， 2， 2

则通过数据口 2 操作联机 翻转 地址 2 第 2 路

参数 2：点动

参数 3：手动输入表达式

例：参数 3：2， 2， 3， 3000

则通过数据口 2 操作联机 点动 地址 2 第 3 路 3000 毫秒

参数 2：顺序断开

参数 3：手动输入表达式

例：参数 3：2， 2， 8， 3， 500

则通过数据口 2 操作联机 顺序断开 地址 2 的第 8 路-第 3 路
间隔时间 500 毫秒

参数 2：顺序闭合

参数 3：手动输入表达式

例：参数 3：2， 2， 3， 8， 1000

则通过数据口 2 操作联机 顺序闭合 地址 2 的第 3 路-第 8 路
间隔时间 1000 毫秒

●以上功能用法可以参考程序示例：23-系统软件资源-18

★参数 1：芯片内部存储器

参数 2：从存储扇区读取缓冲区

参数 3：手动输入存储扇区的序号

例：参数 3：0 则读取扇区 0 到缓冲区

参数 2：清零缓冲区 所有字节赋值为 0

参数 3：空

此选项功能，为清零缓冲区

参数 2: 保存缓冲区到存储扇区

参数 3: 手动输入存储扇区的序号

例: **参数 3:** 0 则写缓冲区到存储扇区 0

参数 2: 写缓冲区数组 占用空间数组长度+1

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 0, 2046

则写缓冲区数组 全局数组 0 到缓存区内部地址 2046

参数 2: 读缓冲区 数组

参数 3: 手动输入表达式

例: **参数 3:** 全局数组 1, 2046

则读缓冲区数组 全局数组 1 = 读缓存区内部地址 2046

● **以上功能用法可以参考程序示例: 23-系统软件资源-19**

★ **参数 1:** modbus 通讯协议

参数 2: 读 modbus 从机寄存器 (功能码 01 02 03)

参数 3: 手动输入表达式

例: **参数 3:** 2, 2, 10001, 8

则通过数据口 2 读 modbus 从机寄存器, 从机模块地址 2

寄存器地址 10001, 寄存器数量 8 个

即读取地址 2 的 1-8 路输入状态

● **以上功能用法可以参考程序示例: 23-系统软件资源-20**

参数 2: 写单个 modbus 从机寄存器 (功能码 05 06)

参数 3: 手动输入表达式

例: **参数 3:** 2, 2, 3, 1

则通过数据口 2 写单个 modbus 从机寄存器, 从机模块地址 2

寄存器地址 3, 寄存器的值为 1

即控制地址 2 的第 3 路继电器闭合

参数 2: 写多个 modbus 从机寄存器 (功能码 15 16)

参数 3: 手动输入表达式

例: **参数 3:** 2, 2, 1, 5, 21

则通过数据口 2 写多个 modbus 从机寄存器, 从机模块地址 2

寄存器地址起始 1，数量 5，寄存器的值为 21

10 进制：21 转为 2 进制：10101

即控制地址 2 的 第 1 路、3 路、5 路继电器闭合

●以上功能用法可以参考程序示例：23-系统软件资源-21

参数 2：提取本机接收到的 modbus 从机寄存器的值

参数 3：手动输入表达式

例：参数 3：2，2，10001，输出 1，8

则通过数据口 2 提取本机接收到的 modbus 从机寄存器的值，
从机模块地址 2，寄存器地址起始 10001，数量 8 的值

赋值给本机输出 1-输出 8

即地址 2 的 1-8 路输入状态，同步本机 1-8 路继电器输出

●以上功能用法可以参考程序示例：23-系统软件资源-20

★参数 1：脉冲输出控制

参数 2：设置频率

参数 3：手动输入整数数值

参数 2：设置占空比（浮点数 0.0-1.0）

参数 3：手动输入浮点数数值

此功能未开发，请忽略。

（2）类型：输出操作

用来操作继电器输出的选项，使用相对简单。

★参数 1：断开某一路/闭合某一路/翻转某一路

参数 2：输出 1-输出 32

参数 3：空

例：参数 1：闭合某一路 参数 2：输出 1 则闭合第 1 路

★参数 1：点动某一路

参数 2：输出 1-输出 32

参数 3：手动输入要点动的时间，单位：ms

例：参数 2：输出 3 参数 3：1000 则点动第 3 路 1000ms

★参数 1：断开某几路/闭合某几路/翻转某几路

参数 2：参数 3 代表第几路 中间使用空格间隔

参数 3: 手动输入继电器的路数

例: **参数 1:** 断开某几路 **参数 3:** 2 3 4
则组断开第 2 路、第 3 路、第 4 路

★**参数 1:** 全部断开/全部闭合/全部翻转

参数 2: 全部断开/全部闭合/全部翻转

参数 3: 空

例: **参数 1:** 全部断开 **参数 2:** 全部断开
则本机的所有继电器全部断开

●**以上功能用法可以参考程序示例: 1-外部输入-输出操作-1**

★**参数 1:** 断开某一路固定时间然后闭合

参数 2: 输出 1-输出 32

参数 3: 手动输入时间参数, 单位: ms

例: **参数 2:** 输出 1 **参数 3:** 1000
则断开输出 1, 1000ms 后自动闭合

★**参数 1:** 赋值为参数 3 的值

参数 2: 参数 3 输入具体数值

参数 3: 手动输入十进制数值

例: **参数 3:** 167 则把 10 进制: 167 转为 2 进制: 10100111,
0 代表断开, 1 代表闭合, 即第 1 2 3 6 8 路闭合

★**参数 1:** 赋值为全局整数的值

参数 2: 参数 3 代表全局整数的编号

参数 3: 手动输入全局整数的编号

例: **参数 3:** 3 如果全局整数 3 = 255
则把 10 进制: 255 转为 2 进制: 11111111
0 代表断开, 1 代表闭合, 即第 1-8 路继电器全部闭合

★**参数 1:** 断开全局整数指向的某一路/闭合全局整数指向的某一路/翻转全局整数指向的某一路

参数 2: 参数 3 代表全局整数的编号

参数 3: 手动输入十进制数值

例: **参数 1:** 翻转全局整数指向的某一路 **参数 3:** 6

如果全局整数 6 = 2 则翻转第 2 路继电器

★参数 1: 点动全局整数指向的某一路 (默认点动时间 50ms)

参数 2: 参数 3 代表全局整数的编号

参数 3: 手动输入十进制数值

例: 参数 3: 4 如果全局整数 4 = 4 则点动第 4 路 50ms

●以上功能用法可以参考程序示例: 1-外部输入-输出操作-2

★参数 1: 输出 = 联机开关量

参数 2: 输出 1-输出 32

参数 3: =联机开量 地址 0 第 0 路

例: 参数 2: 输出 1 参数 3: =联机开量 地址 2 第 1 路

则本机输出 1 = 联机开量 地址 2 第 1 路的状态

★参数 1: 输出 X = 输出 Y

参数 2: 输出 1-输出 32

参数 3: =输出 0

例: 参数 2: 输出 2 参数 3: =输出 1

则本机输出 2 = 输出 1 的状态

★参数 1: 输出 = 输入

参数 2: 输出 1-输出 32

参数 3: =输入 0

例: 参数 2: 输出 3 参数 3: =输入 3

则本机输出 3 = 输入 3 的状态

★参数 1: 输出 = 位变量

参数 2: 输出 1-输出 32

参数 3: =位变量 0

例: 参数 2: 输出 4 参数 3: =位变量 1

则本机输出 4 = 位变量 1 的状态

★参数 1: 顺序动作

参数 2: 顺序断开/顺序闭合

参数 3: 第 0 路 到 第 0 路 间隔 0 毫秒

例: 参数 2: 顺序闭合 参数 3: 第 8 路 到 第 1 路间隔 500 毫秒

则顺序闭合第 8 路到第 1 路，间隔时间 500 毫秒

●以上功能用法可以参考程序示例：1-外部输入-输出操作-3

(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。

*定时器默认倒计时形式，即单位时间减 1。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 1 或者停止定时。

*如果定时器处在单位时间减 1 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：定时器 1-定时器 199

用表达式方式写定时器，支持：定时器 0-定时器 199

参数 2：等于

参数 3：手动输入定时器的值。

例：参数 1：定时器 1 参数 3：255 则定时器 1 等于 255

参数 2：单位时间加 1/单位时间减 1/停止定时

参数 3：空

例：参数 1：定时器 1 参数 2：单位时间加 1

则定时器 1 单位时间加 1，即向上计数

●以上功能用法可以参考程序示例：3-定时器-2

(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：=0/=1/翻转

参数 2：位变量 1-位变量 199

用表达式方式写位变量，支持：位变量 0-位变量 199

参数 3：空

例：参数 1：=1 参数 2：位变量 1 则位变量 1=1

●以上功能用法可以参考程序示例：4-位变量-2

(5) 类型：全局整数（原内部变量）

全局整数可以作为输入，也可以作为输出。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，我们就可以用到全局整数功能。

全局整数功能，全局有效。

全局整数取值范围：-2147483648 — 2147483647

★参数 1：全局整数 1-全局整数 199

参数 2：等于某值/加上某值/减去某值/乘以某值/除以某值

参数 3：手动输入整数数值

例：参数 1：全局整数 1 参数 2：加上某值 参数 3：100

则全局整数 1 = 全局整数 1+100

●以上功能用法可以参考程序示例：5-全局整数-2

参数 2：左移某值/右移某值/与运算/或运算/异或运算

参数 3：手动输入整数数值

例：参数 1：全局整数 2 参数 2：右移某值 参数 3：2

则全局整数 2 = 全局整数 2 右移 2 位

如 果：全局整数 2 = 7890

处理后：全局整数 2 = 1972

处理过程：10 进制：7890，转为 2 进制：1111011010010，向右移 2 位后，等于 2 进制：11110110100，再转为 10 进制，即：1972

●以上功能用法可以参考程序示例：5-全局整数-3

参数 2：= 定时器的值 参数 3 代表定时器序号

参数 3：手动输入定时器的编号

例：参数 1：全局整数 1 参数 3：1

则全局整数 1 = 定时器 1 的值

参数 2：= 全局整数的值 参数 3 代表全局整数序号

参数 3：手动输入全局整数的编号

例：参数 1：全局整数 2 参数 3：3

则全局整数 2 = 全局整数 3 的值

●以上功能用法可以参考程序示例：5-全局整数-4

参数 2：= 公共数据量的值 此功能未开发，请忽略。

参数 2: = 模拟量输入的值 此功能未开发, 请忽略。

参数 2: = 显示变量的值 此功能未开发, 请忽略。

参数 2: = 当前时间的值 此功能未开发, 请忽略。

参数 2: = 本机地址的值

参数 3: 空

例: 参数 1: 全局整数 1, 则全局整数 1 = 本机地址的值

参数 2: = 当前年的值/ 当前月的值/ 当前日的值/当前星期的值/ 当前时的值/ 当前分的值/ 当前秒的值/ 本机开关量输入的值/ 本机开关量输出的值

参数 3: 空

例: 参数 1: 全局整数 3 参数 2: = 当前年的值
则全局整数 3 = 当前年的值

参数 2: = 联机开关量

参数 3: 手动输入联机地址的值

例: 参数 1: 全局整数 12 参数 3: 2

则全局整数 12 = 联机开关量地址 2 的值

●以上功能用法可以参考程序示例: 5-全局整数-6

参数 2: 加上全局整数的值/减去全局整数的值/乘以全局整数的值/除以全局整数的值

参数 3: 手动输入全局整数的编号

例: 参数 1: 全局整数 1 参数 2: 加上全局整数的值 参数 3: 2
则全局整数 1 = 全局整数 1 + 全局整数 2

参数 2: 保存值到 ROM 扇区 0

参数 3: 手动输入扇区位置编号

例: 参数 1: 全局整数 1 参数 3: 5

则全局整数 1 保存值到扇区 0 位置 5

参数 2: 从 ROM 扇区 0 中读取

参数 3: 手动输入扇区位置编号

例: 参数 1: 全局整数 1 参数 3: 5

则全局整数 1 读取值，从扇区 0 位置 5

参数 2: 位运算 置 1 / 位运算 置 0 / 位运算 翻转

参数 3: 手动输入位编号

例: **参数 1:** 全局整数 6 **参数 2:** 位运算 置 1
 参数 3: 5

则全局整数 6 BIT5=1

●以上功能用法可以参考程序示例: 5-全局整数-7

参数 2: 保存值到 RAM (此功能不支持掉电保存)

参数 3: 手动输入 RAM 编号

例: **参数 1:** 全局整数 1 **参数 3:** 1
则全局整数 1 保存值到 RAM1

参数 2: 从 RAM 中读取 (此功能不支持掉电保存)

参数 3: 手动输入 RAM 编号

例: **参数 1:** 全局整数 1 **参数 3:** 1
则全局整数 1 读取值，从 RAM1

参数 2: 设置随后操作 ROM 扇区编号

参数 3: 手动输入 ROM 编号

例: **参数 1:** 全局整数 2 **参数 3:** 1
则设置全局整数 2 读写 ROM 扇区 1

参数 2: 保存到 ROM (此功能支持掉电保存)

参数 3: 手动输入 ROM 位置编号

例: **参数 1:** 全局整数 2 **参数 3:** 1
则全局整数 2 保存值到 ROM 位置 1

参数 2: 从 ROM 读取 (此功能支持掉电保存)

参数 3: 手动输入 ROM 位置编号

例: **参数 1:** 全局整数 2 **参数 3:** 1
则全局整数 2 读取值，从 ROM 位置 1

●以上功能用法可以参考程序示例: 5-全局整数-8

参数 2: 某一位 = 全局整数的某一位

参数 3: BIT0 = 全局整数 0 BIT0

例: **参数 1:** 全局整数 3 **参数 3:** BIT0 = 全局整数 32 BIT5
则全局整数 3 BIT0 = 全局整数 32 BIT5

参数 2: 某一位 = 局部整数的某一位

参数 3: BIT0 = 整数 0 BIT0

例: **参数 1:** 全局整数 3 **参数 3:** BIT1 = 整数 1 BIT4
则全局整数 3 BIT1 = 整数 1 BIT4

参数 2: 某一位 = 联机开关量的某一路

参数 3: BIT0 = 联机开关量地址 0 第 0 路

例: **参数 1:** 全局整数 3 **参数 3:** BIT2=联机开关量地址 2 第 5 路
则全局整数 3 BIT2 = 联机开关量地址 2 第 5 路

参数 2: 某一位 = 输入

参数 3: BIT0 = 输入 0

例: **参数 1:** 全局整数 3 **参数 3:** BIT3 = 输入 1
则全局整数 3 BIT3 = 输入 1

参数 2: 某一位 = 输出

参数 3: BIT0 = 输出 0

例: **参数 1:** 全局整数 3 **参数 3:** BIT4 = 输出 1
则全局整数 3 BIT4 = 输出 1

参数 2: 某一位 = 位变量

参数 3: BIT0 = 位变量 0

例: **参数 1:** 全局整数 3 **参数 3:** BIT5 = 位变量 1
则全局整数 3 BIT5 = 位变量 1

参数 2: = 局部整数

参数 3: = 整数 0

例: **参数 1:** 全局整数 1 **参数 3:** = 整数 1
则全局整数 1 = 整数 1 (整数即局部整数)

● 以上功能用法可以参考程序示例: 5-全局整数-9

参数 2: = 全局数组的某一个字节(16 进制)

参数 3: = 全局数组 0 第 0 个字节

例: **参数 1:** 全局整数 1 **参数 3:** = 全局数组 1 第 8 个字节

则全局整数 1=全局数组 1 第 8 个字节

如 果: 全局数组 1 = {01 02 03 04 05 06 07 0B 09}

第 8 个字节为: {09}, 转为 10 进制: 9

处理后: 全局整数 0 = 9

参数 2: = 全局数组的某几个字节(16 进制)

参数 3: = 全局数组 0 起始 0 数量 0

例: **参数 1:** 全局整数 2 **参数 3:** = 全局数组 1 起始 2 数量 3

则全局整数 2=全局数组 1 字节 起始 2 数量 3

如 果: 全局数组 1 = {01 02 03 04 05 06 07 0B 09}

全局数组 1 起始 2 数量 3 为: {03 04 05}, 转为 10 进制: 197637

处理后: 全局整数 2 = 197637

参数 2: = 全局数组的某几个字符(10 进制)

参数 3: = 全局数组 0 起始 0 数量 0

例: **参数 1:** 全局整数 3 **参数 3:** = 全局数组 3 起始 8 数量 3

则全局整数 3=全局数组 3 字符 起始 8 数量 3

如 果: 全局数组 3 = 字符串: 123 458 769 147 258

处理后: 全局整数 3 = 字符串: 769

参数 2: = 提取全局数组中的数字(10 进制)

参数 3: = 全局数组 0 第 0 组数字

例: **参数 1:** 全局整数 4 **参数 3:** = 全局数组 2 第 2 组数字

则全局整数 4=全局数组 2 第 2 组数字

如 果: 全局数组 2 = 字符串: 785 235 461 479

处理后: 全局整数 4 = 字符串: 461

参数 2: = 局部数组的某一个字节(16 进制) 功能及用法同上

参数 2: = 局部数组的某几个字节(16 进制) 功能及用法同上

参数 2: = 局部数组的某几个字符(10 进制) 功能及用法同上

参数 2: = 提取局部数组中的数字(10 进制) 功能及用法同上

●以上功能用法可以参考程序示例: 5-全局整数-10

参数 2: = 临时数据包的某一个字节(16 进制) 功能及用法同上

参数 2: = 临时数据包的某几个字节(16 进制) 功能及用法同上

参数 2: = 临时数据包的某几个字符(10 进制) 功能及用法同上

参数 2: = 提取临时数据包中的数字(10 进制) 功能及用法同上

●以上功能用法可以参考程序示例：5-全局整数-11

参数 2: 通过数据口发送(16 进制 高位在前)

参数 3: 通过数据口 0 16 进制发送

例：参数 1: 全局整数 1 参数 3: 通过数据口 1 16 进制发送
则全局整数 1 通过数据口 1 16 进制发送

如果，全局整数 1 = 5964034

10 进制：5964034，转 16 进制：5B 01 02

那么，通过数据口 1 发送 16 进制数据：5B 01 02

参数 2: 通过数据口发送(10 进制字符串)

参数 3: 通过数据口 0 字符串发送

例：参数 1: 全局整数 1 参数 3: 通过数据口 1 字符串发送
则全局整数 1 通过数据口 1 字符串发送

如果，全局整数 1 = 5964034

那么，通过数据口 1 发送 10 进制数据：5964034

●以上功能用法可以参考程序示例：5-全局整数-12

(6) 类型：数据口发送数据(118)

★参数 1: 数据口 1-数据口 32

参数 2: 发送 16 进制数据

参数 3: 手动输入数据内容

例：参数 1: 数据口 1 参数 3: 01 02 03 04 05 06 07 08

则通过数据口 1 发送 16 进制数据：01 02 03 04 05 06 07 08

参数 2: 发送字符串

参数 3: 手动输入数据内容

例：参数 1: 数据口 1 参数 3: 12345678

则通过数据口 1 发送字符串：12345678

参数 2: 发送表达式

参数 3: 手动输入表达式内容

例: **参数 1:** 数据口 1 **参数 3:** 全局数组 1

则通过数据口 1 发送 全局数组 1

例: **参数 1:** 数据口 1 **参数 3:** 整数 1

则通过数据口 1 发送 整数 1

例: **参数 1:** 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 1:** 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: 手动输入表达式

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

如果整数 0 = 198, 则通过数据口 1 发送字符串: 目标数据 198

参数 2: 延迟等待

参数 3: 手动输入延迟等待时间, 单位: ms

例: **参数 1:** 数据口 1 **参数 3:** 5000

则数据口 1 延迟等待 5000ms 后, 才可以正常接收或者发送数据

●以上功能用法可以参考程序示例: 8-数据口发送数据-1

(7) 类型: 系统简单设置

★**参数 1:** 操作联机继电器数据口(默认数据口 1)

参数 2: 数据口 1-数据口 32

参数 3: 空

例: **参数 2:** 数据口 2 则操作联机继电器使用数据口 2

★**参数 1:** 操作联机继电器返回码(默认无返回码)

参数 2: 通过无回码命令实现操作

参数 3: 空

参数 2: 通过有回码命令实现操作

参数 3: 空

●以上功能用法可以参考程序示例: 16-系统简单设置-1

★参数 1: 擦除全局整数存储区 ROM

参数 2: 掉电保存 ROM 扇区 0-99

参数 3: 空

★参数 1: 擦除数组存储区 ROM

参数 2: 掉电保存 ROM 扇区 0-99

参数 3: 空

●以上功能用法可以参考程序示例: 16-系统简单设置-2

★参数 1: 表达式数组与变量默认转换方式

参数 2: 字节转换

参数 3: 空

参数 2: 字符串转换

参数 3: 空

●以上功能用法可以参考程序示例: 16-系统简单设置-3

★参数 1: 小数转字符串默认小数点位数

参数 2: 默认 0 位小数-默认 9 位小数

参数 3: 空

●以上功能用法可以参考程序示例: 16-系统简单设置-4

★参数 1: 设置 485 通讯地址

参数 2: 地址 0-地址 254 (地址 245 为广播地址, 请勿用)

参数 3: 空

●以上功能用法可以参考程序示例: 16-系统简单设置-5

(8) 类型: 系统特殊设置

★参数 1: 点动操作联机继电器时间设置(单位 ms)

参数 2: 参数 3 输入 单位 ms

参数 3: 手动输入数值

例: 参数 3: 2000 则点动操作联机继电器时间为 2000 毫秒

●以上功能用法可以参考程序示例: 17-系统特殊设置-1

★参数 1: 作为公共数据量广播的数据来源

此功能未开发, 请忽略。

★参数 1: 临时数据包间隔的毫秒数

参数 2: 数据口 1-数据口 32

参数 3: 手动输入数值

例: 参数 2: 数据口 1 参数 3: 3000

则数据口 1 接收临时数据包的间隔时间为 3000 毫秒

即数据口 1 在 3000 毫秒内接收到的数据，理解为一个数据包

●以上功能用法可以参考程序示例: 17-系统特殊设置-2

★参数 1: 输入连击和长按的时间常数(ms)

参数 2: 全部输入

参数 3: 手动输入数值

例: 参数 3: 2000

则全部输入连击和长按的时间常数为 2000 毫秒

比如: 输入 1 长按 2000ms 未抬起，则输入 1 长按触发成功

●以上功能用法可以参考程序示例: 17-系统特殊设置-1

参数 2: 输入 1-输入 32

参数 3: 手动输入数值

例: 参数 2: 输入 8 参数 3: 3000

则输入 8 连击和长按的时间常数为 3000 毫秒

比如: 输入 8 长按 3000ms 未抬起，则输入 8 长按触发成功

●以上功能用法可以参考程序示例: 17-系统特殊设置-3

★参数 1: 程序运行间隔 (ms, 数字越小越快)

参数 2: 无

参数 3: 手动输入数值

例: 参数 3: 5000 则程序每 5000ms 运行一次

●以上功能用法可以参考程序示例: 17-系统特殊设置-4

★参数 1: 使能或者失能控制指令

参数 2: 根据需求选择 使能或失能的指令

参数 3: 空

●以上功能用法可以参考程序示例: 17-系统特殊设置-5

17-系统特殊设置-6

(9) 类型：通讯参数设置

★参数 1：数据口 1-数据口 32

参数 2：UART 波特率设置

参数 3：手动输入要设置的波特率参数，波特率支持：
2400/4800/9600/19200/38400/57600/115200

●以上功能用法可以参考程序示例：18-通讯参数设置-1

参数 2：UART 奇偶校验位设置(0 无 1 奇 2 偶)

参数 3：手动输入数值

参数 2：UART 数据位设置(8 或 9)

参数 3：手动输入数值

参数 2：UART 停止位设置(0 代表 1 1 代表 1.5 2 代表 2)

参数 3：手动输入数值

●以上功能用法可以参考程序示例：18-通讯参数设置-2
18-通讯参数设置-3

参数 2：发送数据延迟等待

参数 3：手动输入数值，单位：ms

例：参数 1：数据口 1 参数 3：10000

则数据口 1 发送数据延迟等待 10000 毫秒

●以上功能用法可以参考程序示例：18-通讯参数设置-1

(10) 类型：批量操作内部资源

★参数 1：全部定时器

参数 2：清零

参数 3：空

此选项功能，将全部定时器清零

参数 2：赋值为参数 3 的值

参数 3：手动输入数值

例：参数 3：300 则全部定时器赋值为 300

参数 2：赋值为全局整数

参数 3：手动输入全局整数的编号

例：参数 3：1 则全部定时器赋值为全局整数 1 的值

●以上功能用法可以参考程序示例：19-批量操作内部资源-1

★参数 1: 全部全局整数

参数 2: 清零

参数 3: 空

此选项功能，将全部全局整数清零

参数 2: 赋值为参数 3 的值

参数 3: 手动输入数值

例：参数 3: 500 则全部全局整数赋值为 500

参数 2: 赋值为全局整数

参数 3: 手动输入全局整数的编号

例：参数 3: 2 则全部全局整数赋值为全局整数 2 的值

●以上功能用法可以参考程序示例：19-批量操作内部资源-2

(11) 类型：调用函数

★参数 1: 调用函数

参数 2: 函数 1-函数 100

参数 3: 空

★参数 1: 返回函数开始

参数 2: 空

参数 3: 空

★参数 1: 函数返回

参数 2: 空

参数 3: 空

●以上功能用法可以参考程序示例：20-调用函数-1

20-调用函数-2

(12) 类型：操作联机继电器模块(发送命令)

★参数 1: 地址 0-地址 254

参数 2: 查询状态

参数 3: 空

例：参数 1: 地址 2 则查询地址 2 的开关量状态

●以上功能用法可以参考程序示例：11-联机开关量-1

参数 2: 断开某一路/闭合某一路/翻转某一路/点动某一路

参数 3: 手动输入要操作的继电器路数序号

例: **参数 1:** 地址 2 **参数 2:** 点动某一路 **参数 3:** 3

则点动地址 2 的第 3 路 (默认点动时间 500ms)

参数 2: 断开某几路(组断开)/闭合某几路(组闭合)/
翻转某几路(组翻转)

参数 3: 手动输入要操作的继电器路数序号, 每两个序号之间
用空格做间隔

例: **参数 1:** 地址 2 **参数 2:** 翻转某几路(组翻转) **参数 3:** 4 5 6

则组翻转地址 2 的第 4 路、第 5 路、第 6 路

参数 2: 全部断开/全部闭合/全部翻转

参数 3: 空

例: **参数 1:** 地址 2 **参数 2:** 全部闭合

则闭合地址 2 的全部输出

● **以上功能用法可以参考程序示例: 11-操作联机继电器模块-2**

(13) 类型: 全局数组简单处理

★ **参数 1:** 全局数组 1-全局数组 100

参数 2: 数据包无参数处理

参数 3: 清空

例: **参数 1:** 全局数组 1, 则全局数组 1 清空

参数 3: 字节倒序

例: **参数 1:** 全局数组 1, 则全局数组 1 字节倒序

如 果: 原全局数组 1 = 16 进制: 01 02 03

则倒序后的全局数组 1 = 16 进制: 03 02 01

参数 3: 每个字节的前后四位互换

例: **参数 1:** 全局数组 1

则全局数组 1 每个字节的前后四位互换

如 果: 全局数组 1 = 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全局数组 1 = 16 进制:

10 20 30 40 50 60 70 80

参数 3: 字符串全部转化为小写/大写

例: 全局数组 2 = 字符串: 123ABC456efd

则字符串全部转化为小写后，全局数组 2 = 123abc456ef

字符串全部转化为大写后，全局数组 2 = 123ABC456EF

参数 3: 尾部添加和校验字节

例：全局数组 1 = 16 进制：01 02 03 04

则尾部添加和校验后，全局数组 1 = 16 进制：01 02 03 04 0A

参数 3: 尾部添加 CRC16 校验字节

例：全局数组 3 = 16 进制：01 01 00 00 00 05

则尾部添加 CRC16 校验后，

全局数组 3 = 01 01 00 00 00 05 FC 09

参数 3: 每个字节的位倒序

例：全局数组 1 = 16 进制：01 02 03 04

则每个字节的位倒序后，全局数组 1 = 16 进制：80 40 C0 20

●以上功能用法可以参考程序示例：13-全局数组简单处理-1

参数 2: 加上其他数组

参数 3: 全局数组 1-全局数组 100

例：参数 1: 全局数组 1 参数 3: 全局数组 2

则全局数组 1 = 全局数组 1+全局数组 2

如 果：全局数组 1 = 16 进制：01 02 03

全局数组 2 = 16 进制：04 05 06

处理后：全局数组 1 = 16 进制：01 02 03 04 05 06

参数 2: 赋值为其他数组

参数 3: 全局数组 1-全局数组 100

例：参数 1: 全局数组 1 参数 3: 全局数组 3

则全局数组 1 = 全局数组 3

如 果：全局数组 3 = 16 进制：07 08 09

则全局数组 1 = 16 进制：07 08 09

参数 2: 删除首字节

参数 3: 空

例：参数 1: 全局数组 1 则删除全局数组 1 的首个字节

如 果：全局数组 1 = 16 进制：07 08 09

处理后：全局数组 1 = 16 进制：08 09

参数 2: 删除尾字节

参数 3: 空

例: **参数 1:** 全局数组 1 则删除全局数组 1 的最后一个字节

如 果: 全局数组 1 = 16 进制: 08 09

处理后: 全局数组 1 = 16 进制: 08

参数 2: 删除某字节 (直接表达)

参数 3: 第 1 个字节-第 200 个字节

例: **参数 1:** 全局数组 2 **参数 3:** 第 2 个字节

则删除全局数组 2 的第 2 个字节

如 果: 全局数组 2 = 16 进制: 04 05 06

则删除第 2 个字节后, 全局数组 2 = 16 进制: 04 05

参数 2: 删除某字节 (变量表达)

参数 3: 全局整数 1-全局整数 100

例: **参数 1:** 全局数组 2 **参数 3:** 全局整数 1

则删除全局数组 2 的全局整数 1 位置的字节

如 果: 全局数组 2 = 16 进制: 04 05 全局整数 1=1

则删除全局整数 1 位置的字节后, 全局数组 2 = 16 进制: 04

参数 2: 首部添加字节 (变量表达)

参数 3: 全局整数 1-全局整数 100

例: **参数 1:** 全局数组 3 **参数 3:** 全局整数 2

则在全局数组 3 的首部添加一个字节, 字节值为全局整数 2 的值

如 果: 全局数组 3 = 16 进制: 07 08 09 全局整数 2 = 3

则首部添加全局整数 2 的字节值后,

全局数组 3 = 16 进制: 03 07 08 09

参数 2: 尾部添加字节 (变量表达)

参数 3: 全局整数 1-全局整数 100

例: **参数 1:** 全局数组 3 **参数 3:** 全局整数 3

则在全局数组 3 的尾部添加一个字节, 字节值为全局整数 3 的值

如 果: 全局数组 3 = 16 进制: 03 07 08 09 全局整数 3 = 5

则尾部添加全局整数 3 的字节值后,

全局数组 3 = 16 进制: 03 07 08 09 05

●以上功能用法可以参考程序示例：13-全局数组简单处理-2

参数 2: 全局整数 = 数组累加和

参数 3: 全局整数 1-全局整数 100

例：参数 1: 全局数组 1 参数 3: 全局整数 1
则全局数组 1 的累加和赋值给全局整数 1

参数 2: 全局整数 = 数组异或和

参数 3: 全局整数 1-全局整数 100

例：参数 1: 全局数组 1 参数 3: 全局整数 2
则全局数组 1 的异或和赋值给全局整数 2

参数 2: 全局整数 = 数组 crc16 运算

参数 3: 全局整数 1-全局整数 100

例：参数 1: 全局数组 1 参数 3: 全局整数 3
则全局数组 1 进行 crc16 运算后赋值给全局整数 3

参数 2: 全局整数 = 数组长度

参数 3: 全局整数 1-全局整数 100

例：参数 1: 全局数组 1 参数 3: 全局整数 4
则全局数组 1 的字节长度赋值给全局整数 4

●以上功能用法可以参考程序示例：13-全局数组简单处理-3

参数 2: 通过数据口发送出去

参数 3: 数据口 1-数据口 32

例：参数 1: 全局数组 1 参数 3: 数据口 1
则通过数据口 1 发送全局数组 1

●以上功能用法可以参考程序示例：13-全局数组简单处理-2

参数 2: 设置字符串类型转换时的进制

参数 3: 10 进制(默认)/16 进制/2 进制

●以上功能用法可以参考程序示例：13-全局数组简单处理-4

参数 2: 设置数组转换时字节代表的意义

参数 3: 一个字节 2 个 16 进制位(默认)/一个字节 1 个 16 进

制位/一个字节 1 个 10 进制位/一个字节 1 个 2 进制位

●以上功能用法可以参考程序示例：13-全局数组简单处理-5

参数 2：设置类型转换时字节的数量

参数 3：根据需要

●以上功能用法可以参考程序示例：13-全局数组简单处理-5

参数 3：固定 1 字节-32 字节

●以上功能用法可以参考程序示例：13-全局数组简单处理-4

参数 2：全局整数->字符串

参数 3：全局整数 1-全局整数 100

参数 2：字符串->全局整数

参数 3：全局整数 1-全局整数 100

参数 2：全局整数->为全局数组

参数 3：全局整数 1-全局整数 100

参数 2：全局数组->全局整数

参数 3：全局整数 1-全局整数 100

●以上功能用法可以参考程序示例：13-全局数组简单处理-6

参数 2：全局数组保存到 ROM 扇区 0

参数 3：位置 0-位置 251

参数 2：全局数组从 ROM 扇区 0 读取

参数 3：位置 0-位置 251

●以上功能用法可以参考程序示例：13-全局数组简单处理-7

参数 2：删除头部的 N 个字节

参数 3：N=1-200

例：参数 1：全局数组 1 参数 3：N=3

则删除全局数组 1 的头部 3 个字节

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

处理后：全局数组 1 = 16 进制：04 05 06

参数 2：删除尾部的 N 个字节

参数 3：N=1-200

例：参数 1：全局数组 1 参数 3：N=3

则删除全局数组 1 的尾部 3 个字节

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

处理后：全局数组 1 = 16 进制：01 02 03

●以上功能用法可以参考程序示例：13-全局数组简单处理-8

(14) 类型：全局数组复杂处理

★参数 1：全局数组 1-全局数组 100

参数 2：全局整数 = 某一字节(全局整数代表序号)

参数 3：将全局整数 0 位置的字节赋值给全局整数 0

例：参数 1：全局数组 1

参数 3：将全局整数 1 位置的字节赋值给全局整数 2

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

全局整数 1 = 3

处理后：全局整数 2 = 10 进制：4

参数 2：某一字节(全局整数代表序号) = 全局整数

参数 3：将全局整数 0 赋值给全局整数 0 位置的字节

例：参数 1：全局数组 1

参数 3：将全局整数 1 赋值给全局整数 2 位置的字节

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

全局整数 1 = 3 全局整数 2 = 4

处理后：全局数组 1 = 16 进制：01 02 03 04 03 06

参数 2：插入一个字节 前(变量表达)

参数 3：将全局整数 0 插入到全局整数 0 位置的字节之前

例：参数 1：全局数组 1

参数 3：将全局整数 1 插入到全局整数 2 位置的字节之前

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

全局整数 1 = 3 全局整数 2 = 4

处理后：全局数组 1 = 16 进制：01 02 03 04 03 05 06

参数 2：插入一个字节 后(变量表达)

参数 3：将全局整数 0 插入到全局整数 0 位置的字节之后

例：参数 1：全局数组 1

参数 3: 将全局整数 1 插入到全局整数 2 位置的字节之后
如 果: 全局数组 1 = 16 进制: 01 02 03 04 05 06
全局整数 1 = 3 全局整数 2 = 4
处理后: 全局数组 1 = 16 进制: 01 02 03 04 05 **03** 06

●以上功能用法可以参考程序示例: 13-全局数组复杂处理-1

参数 2: 其他全局数组 = 本数组某几个字节

参数 3: 全局数组 0 = 从第 0 个字节开始的 0 个字节

例: **参数 1:** 全局数组 1

参数 3: 全局数组 2 = 从第 2 个字节开始的 3 个字节

如 果: 全局数组 1 = 16 进制: 01 02 03 04 05 06

处理后: 全局数组 2 = 16 进制: 03 04 05

参数 2: 其他全局数组 = 本数组某几个字节(变量表达)

参数 3: 全局数组[全局整数 0] = 全局整数 0 开始的全局整数
0 个字节

例: **参数 1:** 全局数组 1

参数 3: 全局数组[全局整数 2] = 全局整数 2 开始的全局整
数 3 个字节

如 果: 全局数组 1 = 16 进制: 01 02 03 04 05 06

全局整数 2 = 2 全局整数 3 = 3

处理后: 全局数组 2 = 16 进制: 03 04 05

参数 2: 插入一个字节 前(变量表达)

参数 3: 在第 0 个字节之前插入字节 0

例: **参数 1:** 全局数组 1

参数 3: 在第 2 个字节之前插入字节 55

如 果: 全局数组 1 = 16 进制: 01 02 03 04 05 06

10 进制: 55, 转为 16 进制: 37

处理后: 全局数组 2 = 16 进制: 01 02 **37** 03 04 05 06

参数 2: 插入一个字节 后(变量表达)

参数 3: 在第 0 个字节之后插入字节 0

例: **参数 1:** 全局数组 1

参数 3: 在第 3 个字节之后插入字节 66

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

10 进制：66，转为 16 进制：42

处理后：全局数组 2 = 16 进制：01 02 03 04 42 05 06

●以上功能用法可以参考程序示例：13-全局数组复杂处理-2

参数 2：全局整数=全局数组的某几个字节

参数 3：全局整数 0 = 第 0 个开始的 0 个字节

例：参数 1：全局数组 1

参数 3：全局整数 1 = 第 2 个开始的 3 个字节

如 果：全局数组 1 = 16 进制：01 02 03 04 05 06

处理后：全局整数 1 = 16 进制：03 04 05

参数 2：全局整数=字符串的某几个字符

参数 3：全局整数 0 = 第 0 个开始的 0 个字符

例：参数 1：全局数组 2

参数 3：全局整数 2 = 第 2 个开始的 3 个字符

如 果：全局数组 2 = 字符串：147258

处理后：全局整数 2 = 字符串：725

参数 2：保存到 ROM

参数 3：保存到 ROM 扇区 0 位置 0

例：参数 1：全局数组 1

参数 3：保存到 ROM 扇区 0 位置 1

则全局数组 1 保存到 ROM 扇区 0 位置 1

参数 2：从 ROM 读取

参数 3：从 ROM 读取 扇区 0 位置 0

例：参数 1：全局数组 1

参数 3：从 ROM 读取 扇区 0 位置 1

则全局数组 1 从 ROM 读取 扇区 0 位置 1

●以上功能用法可以参考程序示例：13-全局数组复杂处理-3

参数 2：全局整数 = 字符串的第几个数字

参数 3：全局整数 0 = 第 0 组数字

例：参数 1：全局数组 1

参数 3：全局整数 1 = 第 2 组数字

如 果：全局数组 1 = 字符串：147 258 369

处理后：全局整数 1 = 字符串：369

参数 2：其他全局数组=某几个字节（倒数）

参数 3：全局数组 0 = 本数组倒数第 0 个字节开始的 0 个字节
(1 开始)

例：**参数 1：**全局数组 3 **参数 3：**全局数组 2 = 本数组倒数
第 4 个字节开始的 4 个字节(1 开始)

如 果：全局数组 3 = 16 进制：01 02 03 04 05

处理后：全局数组 2 = 16 进制：02 03 04 05

参数 2： = 局部数组

参数 3： = 局部数组 0

例：**参数 1：**全局数组 1 **参数 3：** = 局部数组 1

如 果：局部数组 1 = 16 进制 05 35 38 20

处理后：全局数组 1 = 16 进制 05 35 38 20

●以上功能用法可以参考程序示例：13-全局数组复杂处理-4

(15) 类型：脉冲调制输出 此功能未开发，请忽略。

(16) 类型：局部整数

局部整数功能与全局整数功能类似，可参考全局整数说明

*局部整数 0-局部整数 199

*局部整数功能 局部有效

★**参数 1：**赋值运算

●以上功能用法可以参考程序示例：12-局部整数-6
12-局部整数-7

（注意：赋值为全局小数、局部小数功能不支持）

★**参数 1：**位运算

●以上功能用法可以参考程序示例：12-局部整数-8；
12-局部整数-9；12-局部整数-10；12-局部整数-11。

★**参数 1：**操作内部资源

●以上功能用法可以参考程序示例：12-局部整数-12

★参数 1: 赋值为时间日期

●以上功能用法可以参考程序示例: 12-局部整数-13

★参数 1: 赋值为开关量

●以上功能用法可以参考程序示例: 12-局部整数-14

★参数 1: 掉电存储

●以上功能用法可以参考程序示例: 12-局部整数-15

★参数 1: 与数组互转

●以上功能用法可以参考程序示例: 12-局部整数-16

12-局部整数-17

12-局部整数-18

★参数 1: 向外发送

●以上功能用法可以参考程序示例: 12-局部整数-6

(17) 类型: 局部数组

局部数组功能与全局数组功能类似, 可参考全局数组说明

*局部数组 0-局部数组 199

*局部数组功能 局部有效

★参数 1: 无参数简单处理

●以上功能用法可以参考程序示例: 22-局部数组-4;

22-局部数组-4; 22-局部数组-4; 22-局部整数-7。

★参数 1: 增加字节/字符

●以上功能用法可以参考程序示例: 22-局部数组-8

★参数 1: 删除字节/字符

●以上功能用法可以参考程序示例: 22-局部数组-9

★参数 1: 与其他数组的运算

●以上功能用法可以参考程序示例: 22-局部数组-10

22-局部数组-11

★参数 1: 赋值为局部整数

●以上功能用法可以参考程序示例: 22-局部数组-12
22-局部数组-13

★参数 1: 赋值为全局整数

●以上功能用法可以参考程序示例: 22-局部数组-14
22-局部数组-15

★参数 1: 掉电存储

●以上功能用法可以参考程序示例: 22-局部数组-16

★参数 1: 通过数据口发送

●以上功能用法可以参考程序示例: 22-局部数组-16

(18) 类型: 红外发射

红外码说明: 红外码是一种独立的万能红外编码, 使用一组数据组合来表达一个红外信号, 不同的红外信号对应不同的数据组合。

★参数 1: 万能红外发射(固定码)

参数 2: 红外通道 1-红外通道 8

参数 3: 手动学习红外码

此操作直接发送参数 3 所代表的红外码。此选项可以学习红外码, 学码的方式为光标移到参数 3 的位置, 按下红外遥控器, 电脑上会弹出对话框显示红外码(前提是串口号和地址正确, 可以正常写入和读取内核程序)。

★参数 1: 万能红外发射(根据数组的内容)

参数 2: 红外通道 1-红外通道 8

参数 3: 全局数组 1-全局数组 32

此操作要先给全局数组赋值固定红外码, 模块再根据全局数组的内容进行发送。

★参数 1: 万能红外发射进入对码状态

参数 2: 空

参数 3: 空

此操作一般为接收到某数据，或者通过其他方式触发，从而使红外模块进入对码状态。

★**参数 1**：将接收到的临时红外码赋值给数组

参数 2：全局数组 1-全局数组 200

参数 3：空

此操作可以将接收到的红外码赋值给全局数组，方便客户建立红外码库。

●以上功能用法可以参考程序示例：21-红外发射-1

★**参数 1**：设置红外芯片参数（部分红外芯片支持）

此功能仅供开发者使用，客户请忽略。

（19）类型：表达式/用户函数（118）

*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：表达式运算

参数 3：手动输入表达式

例：**参数 3**：全局数组 1={01 02 03 04 05}

则 全局数组 1 赋值为 16 进制数据：01 02 03 04 05

●以上功能用法可以参考程序示例：14-表达式-2

参数 2：线程等待等式成立，若多条则并且，用分号(;)间隔

参数 3：手动输入表达式

例：**参数 3**：输入 1=1;输出 1=1;定时器 1=1;全局整数 1=100

则参数 3 的 4 个条件全部满足后，线程才会执行。

参数 2：线程等待等式成立，若多条则或者，用分号(;)间隔

参数 3：手动输入表达式

例：**参数 3**：输入 1=1;输出 1=1;定时器 1=1;全局整数 1=100

则参数 3 的 4 个条件任意满足其中 1 个，线程都会执行。

●以上功能用法可以参考程序示例：14-表达式-3

*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：函数 1-函数 100

参数 3：()

例：参数 2：函数 1 则调用函数 1

●以上功能用法可以参考程序示例：14-表达式-4

★参数 1：其它函数动作

参数 2：返回表达式

参数 3：手动输入表达式内容

例：参数 3：{03} 则函数返回 16 进制数据：03

●以上功能用法可以参考程序示例：14-表达式-5

参数 2：返回空

参数 3：空

●以上功能用法可以参考程序示例：14-表达式-6

参数 2：跳转到函数开始

参数 3：空

●以上功能用法可以参考程序示例：14-表达式-7

参数 2：表达式方式调用函数

参数 3：手动输入表达式内容

●以上功能用法可以参考程序示例：14-表达式-8

★参数 1: 表达式转换默认设置

参数 2: 数组转变量为默认整数

参数 3: 空

参数 2: 数组转变量为默认浮点数

参数 3: 空

●以上功能用法可以参考程序示例: 14-表达式-9

★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

参数 2: 数组转整数采用字符串转换模式

参数 3: 空

参数 2: 字节数组转整数低位字节在前

参数 3: 空

参数 2: 字节数组转整数高位字节在前

参数 3: 空

●以上功能用法可以参考程序示例: 14-表达式-10

参数 2: 字符串转整数采用 10 进制(默认)

参数 3: 空

参数 2: 字符串转整数采用 16 进制

参数 3: 空

参数 2: 字符串转整数采用 2 进制

参数 3: 空

●以上功能用法可以参考程序示例: 14-表达式-11

★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

参数 2: 整数转数组采用字符串转换模式

参数 3: 空

参数 2: 整数转字节数组低位字节在前

参数 3: 空

参数 2: 整数转字节数组高位字节在前

参数 3: 空

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

●以上功能用法可以参考程序示例: 14-表达式-12

参数 2: 整数转字符串采用 10 进制(默认)

参数 3: 空

参数 2: 整数转字符串采用 16 进制

参数 3: 空

参数 2: 整数转字符串采用 2 进制

参数 3: 空

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

●以上功能用法可以参考程序示例: 14-表达式-13

★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

参数 2: 数组转浮点数采用字符串转换模式

参数 3: 空

●以上功能用法可以参考程序示例: 14-表达式-14

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

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

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

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

●以上功能用法可以参考程序示例: 14-表达式-15

★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

●以上功能用法可以参考程序示例: 14-表达式-16

参数 2: 浮点数转数组采用字符串转换模式

参数 3: 空

●以上功能用法可以参考程序示例: 14-表达式-15

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

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

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

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

●以上功能用法可以参考程序示例: 14-表达式-16

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

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

●以上功能用法可以参考程序示例: 14-表达式-17

(20) 类型: 局部数组 (118)

★参数 1: 局部数组 1-局部数组 199

用表达式方式写局部数组, 支持: 局部数组 0-局部数组 199

参数 2: 等于 16 进制数据

参数 3: 手动输入

例: 参数 1: 局部数组 1 参数 3: 01 02 03 04 05

则局部数组 1=16 进制: 01 02 03 04 05

参数 2: 等于字符串

参数 3: 手动输入

例: 参数 1: 局部数组 2 参数 3: 123

则局部数组 2 = 字符串: 123

参数 2: 等于表达式

参数 3: 手动输入

例: **参数 1:** 局部数组 3 **参数 3:** {06 07 08}

则局部数组 3 = 16 进制: 06 07 08

例: **参数 1:** 局部数组 3 **参数 3:** "678"

则局部数组 3 = 字符串: 678

例: **参数 1:** 局部数组 3 **参数 3:** 全局数组 0

则局部数组 3 = 全局数组 0

参数 2: 等于格式化字符串

参数 3: 手动输入

例: **参数 1:** 局部数组 199 **参数 3:** "目标数据%d", 全局整数 0

如果全局整数 0 = 156

则局部数组 199 = 目标数据 156

● **以上功能用法可以参考程序示例: 22-局部数组-17**

(21) 类型: 全局数组 (118)

★ **参数 1:** 全局数组 1-全局数组 199

用表达式方式写全局数组, 支持: 全局数组 0-全局数组 199

参数 2: 等于 16 进制数据

参数 3: 手动输入

例: **参数 1:** 全局数组 1 **参数 3:** 01 02 03 04 05

则全局数组 1 = 16 进制: 01 02 03 04 05

参数 2: 等于字符串

参数 3: 手动输入

例: **参数 1:** 全局数组 2 **参数 3:** 123

则全局数组 2 = 字符串: 123

参数 2: 等于表达式

参数 3: 手动输入

例: **参数 1:** 全局数组 3 **参数 3:** {06 07 08}

则全局数组 3 = 16 进制: 06 07 08

例: **参数 1:** 全局数组 3 **参数 3:** "678"

则全局数组 3 = 字符串: 678

例: **参数 1:** 全局数组 3 **参数 3:** 数组 0

则全局数组 3 = 数组 0

参数 2: 等于格式化字符串

参数 3: 手动输入

例: **参数 1:** 全局数组 199 **参数 3:** "目标数据%d", 全局整数 0
如果全局整数 0 = 156

则全局数组 199 = 目标数据 156

● **以上功能用法可以参考程序示例: 13-全局数组-1**

(22) 类型: 系统高级设置(版本 118 慎用)

此功能仅供开发者使用, 客户请忽略。

(23) 类型: 临时数据包设置(118)

★ **参数 1:** 数据口 1-数据口 10

参数 2: 数据头设置

参数 3: 手动输入表达式

例: **参数 1:** 数据口 1 **参数 3:** {0C}

则数据口 1 接收到的数据, 必须满足数据头为 16 进制: 0C
若条件不满足, 数据忽略, 不接收。

参数 2: 数据尾设置

参数 3: 手动输入表达式

例: **参数 1:** 数据口 1 **参数 3:** {22}

则数据口 1 接收到的数据, 必须满足数据尾为 16 进制: 22
若条件不满足, 数据忽略, 不接收。

参数 2: 数据长度设置(字节数)

参数 3: 手动输入

例: **参数 1:** 数据口 1 **参数 3:** 5

则数据口 1 接收到的数据, 必须满足数据的长度为: 5 个字节
若条件不满足, 数据忽略, 不接收。

参数 2: 数据间隔时间设置(ms)

参数 3: 手动输入

例: **参数 1:** 数据口 1 **参数 3:** 3000

则数据口 1 接收数据，在 3000 毫秒内接收到的数据，认定为一个数据包。

参数 2: 数据超时接收 (1 使能 0 失能 上电默认 1)

参数 3: 手动输入 1 或者 0

例：参数 1: 数据口 1 参数 3: 1

则数据口 1 接收数据，允许超时接收。

例：参数 1: 数据口 1 参数 3: 0

则数据口 1 接收数据，不允许超时接收。

参数 2: 数据连续接收 (1 使能 0 失能 上电默认 1)

参数 3: 手动输入 1 或者 0

例：参数 1: 数据口 1 参数 3: 1

则数据口 1 接收数据，允许连续接收。

例：参数 1: 数据口 1 参数 3: 0

则数据口 1 接收数据，不允许连续接收。

● 以上功能用法可以参考程序示例：24-临时数据包设置-1

此说明书基本涵盖了 3.0 内核工具软件 V1.317 及以上版本的各个选项说明及解析，可同步参考配套程序示例，会更方便理解和学习！