

5.0 内核使用说明 V1.00

表达式由参与运算参数和运算符组成。

参与运算的参数：

数据口

- 用来发送或者接收数据的接口 (uart)，有可能是串口，485，网口等等，标号默认从 1 开始，数据口 1，数据口 2，……
- 数据口[表达式]，代表某一个数据口，如果表达式=1，代表数据口 1。
- 数据口[1]与数据口 1 功能相同。
- 数据口 n[表达式]，代表数据口 n 接收到的数据包里面的某一个字节，只读，标号默认从 0 开始，如果表达式=0，代表数据口 1 接收到的第一个元素，以此类推。
- 数据口 1[2]，代表数据口 1 接收到的数据中的第 2 个字节（从 0 开始）。
- 数据口赋值，代表发送数据。
- 等式：数据口 1="abc123456"，数据口 1 发送字符串 abc123456
- 读数据口，代表提取接收到的数据。
- 举例：数组 1=数据口 1 把数据口 1 接收到的数据赋值给数组 1。
- 等式：数据口 1= 数组表达式 数据口 1 发送数组表达式所代表的内容
- 等式：数据口 1={01 02 03 A0 B0} 数据口 1 发送 16 进制数据包：01 02 03 A0 B0。
- 等式：数据口 1="123asd45ASD" 数据口 1 发送字符串：123asd45ASD。
- 等式：数据口 1= 数组 1 发送数组 1 代表的内容
- 等式：数据口 1= 整数 1 发送整数 1 转换成数组后的内容，具体转换规则可以设置与更改

整数

- 用来记录中间状态的数据值，四个字节长度，可以是正数或者负数。函数内有效，默认从 0 开始，整数 0,整数 1,.....
- 整数[表达式]，代表某一个整数，如果表达式=1，代表整数 1，以此类推。
- 整数 n[表达式]，代表整数 n 的某一个位，如果 n=1 表达式=0，代表整数 1 第 0 位。
- 整数或者整数的某一个位都可以读写
- 每个函数默认含有 10 个整数。整数 0-整数 9

定时器

- 定时器全局有效，默认从 0 开始，定时器 0,定时器 1,.....
- 在大于零的情况下，系统每运行一次，定时器减一一次，减到 0 为止。
- 定时器[表达式]，代表某一个定时器，如果表达式=1，代表定时器 1，以此类推。
- 系统默认含有 100 个定时器。定时器 0-定时器 99

全局整数

- 功能与局部整数相同，全局有效。
- 系统默认含有 100 个全局整数。全局整数 0-全局整数 99
- 其他运算功能参考局部整数，比如数组下标，位操作等

位变量

- 全局有效,位变量 0,位变量 1,位变量 2.....
- 位变量[表达式]，代表某一个位变量，如果表达式=1，代表位变量 1，以此类推。
- 系统默认含由 100 个位变量
- 位变量支持边沿检测

常数

- 十进制常数: 0,1,2,3,100,1000,1010,456,123,785,.....
- 十进制负数: -1,-2,-3,.....
- 16 进制常数: 0x12,0x7845,0x56898145.....
- 字符常数: '0','1','2','a','b','c','d',..... 代表对应的 ASC 码
- 浮点数 1.235, -5.636。小数的输入需要加上小数点, 系统会按照浮点数来处理, 系统内部采用 8 字节长度双精度浮点数。

立即字符串

- 字符串使用双引号表达: "0123dds" "werf" "短方法"

立即数组 (16 进制表达)

- 16 进制字节数组使用 {} 表达: {01 02 03 45 88}

数组

- 函数内有效, 默认使用数组表达。
- 默认从 0 开始, 数组 0, 数组 1,
- 数组 [表达式], 代表某一个数组, 如果表达式=1, 代表位数组 1, 以此类推。
- 数组 n[表达式], 代表数组 n 里面的某一个字节, n 默认从 0 开始, 表达式默认从 0 开始, 如果 n=1 表达式=0, 代表数组 1 的第 0 个字节, 以此类推。
- 每个函数默认含由 5 个数组, 最大长度 50 字节
- 数组的最大长度可以由系统函数设定, 设定的长度长了, 数组数量会变少, 反之亦然。

全局数组

- 功能与数组相同, 只是全局有效
- 下标默认从 0 开始, 全局数组 0, 全局数组 1,

io 口

- 供单片机开发者使用, 直接操作 IO 口
- 默认可以操作 4 个 port 口 PA PB PC PD
- IO 表达式: PA0, PA1, PB1, PD5, PA[0]
- 支持元素运算符 [], PA[表达式], 代表 PA 里面的某一个引脚。标号从 0 开始, 如果表达式=0, 代表 PA0, 以此类推。

函数参数

- 代表从函数人口传进来的表达式。
- 默认下标从 0 开始, 函数参数 0, 函数参数 1,
- 注意, 使用方法为只读。

时间参数

- 年 月 日 时 分 秒 星期

参与运算的运算符:

- + 加运算符
- - 减运算符
- * 乘运算符
- / 除运算符
- << 左移运算符
- >> 右移运算符
- % 取余运算符
- & 与运算符
- | 或运算符

- ^ 异或运算符
- = 等于运算符
- > 大于运算符
- < 小于运算符
- >= 大于等于
- <= 小于等于
- <> 不等于
- () 括号运算符, 括起来后是独立的参与运算的参数
- [] 中括号运算符 提取元素的子元素, 里面是表达式
- {} 用来表达立即数组 或者表达语句块

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。
- 区分不同的层次, 从而进行多级判断时候更方便
- 右键可以添加语句块。

表达式举例

- 1,2,3 整数
- 0.1,1.5,2.5568 小数
- -1, -0.1, -5.6685 负数
- 整数 0
- 整数[1]
- 整数[整数 0]
- 整数 0+整数 1
- 整数 1+整数 2+1
- 数组 1[0] 数组 1 的第 0 个元素(起始 0,字节值)
- (((整数 1+2)*3)<<4)+数组 1[0]
- 整数[整数 1]+3
- 整数 1+数组 1[2];整数 1+数组 1 的第 2 个元素(起始 0)
- 数据口 1
- 数据口 1+数组 1
- 数据口 1+{01 03 04 56 FF}
- "124sdf"+{01 02 56 0A AB} 字符串与 16 进制数组组合而成的数组

等式举例:

- 整数 1 = 表达式
- 整数[1] = 表达式
- 整数[整数 0] = 表达式
- 整数 2 = 表达式
- 整数[整数 1] = 表达式
- 数据口 1 = 表达式
- 输出 1 = 1 打开第一路输出
- 输出 1 = 输入 1 根据输入 1 的状态重置输出 1
- 输出 1 = 整数 1 整数 1 等于 0 关闭 等于 1 打开
- 整数[整数[整数 0]]=2, 此情况虽然有点极端, 但可以运行。