1.1简介

相对于上一章节，略有难度，**如果您是初学者而且现在手头很忙可以跳过此节**，这章涉及到一些变量等的操作，但其实并不难，任何一门编程语言都会涉及到变量，即使现在火热的 scratch 编程，比这个难多了，而他主要面向的是 6~10 岁的普通小学生，此节虽是选读篇，还是建议您可以在以后的时间慢慢消化。等熟练使用变量了，您会发现机器会为你做事了，省出时间让你去忙更有意义的事。

1.2变量

1.2.1 变量有什么用

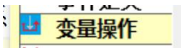
变量是随时变化的量，编程的核心就是变量的操作，大部分指令都支持变量的操作，当使用变量以后可以让你的编程效率变得很高，先选择变量操作指令，再双击某个变量，会弹出以下表：

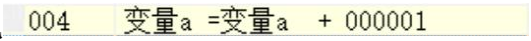
左侧列表是变量的分类，右侧是变量名，其中分类中 变量,菜单,A 盘,B 盘,C 盘,D 盘,是用户随意调用和赋值的。其他都为系统变量，允许读，但不要随意写，当写入某个系统变量是，会执行一个动作，比如写入 输出 00=1 会让打开端口 00 的动作，读 输出 00 会返回端口 00 的状态(0 关闭，1 开启)，不管什么类型的数据，都是纯中文调用。无需像 PLC 一样计算或者背诵变量地址。这里有几个注意点：

A)有别于其他高级语言，所有变量不需要提前定义，而且是全局有效。
B)所有变量不支持负数（负数统一为 0），小数，最大值为 4,294,967,296

1.2.2 用户变量 a~z

系统内置 26 个用户变量 abcd....z，重启后会自动变成 0，通过变量控制可以让您的程序更灵活，更方便，用户变量非常重要，大家应掌握它的用法。

在指令板中选择 

指令，出现 

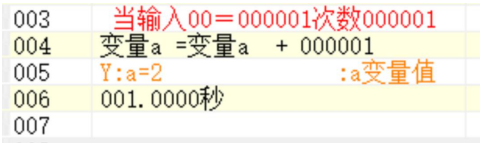
，他的意思是每运行这行程序，变量 a 变大 1，既第一次执行后是 1，第二次是 2，第三次是 3....

以下是变量使用举例：

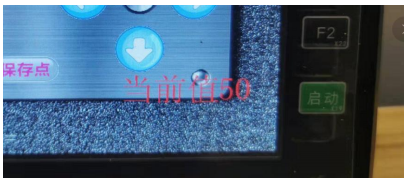
示例 1:第一次按下 X1 按键，Y0 亮，第 2 次按下 Y1 亮，第 3 次 Y2 亮.....

因为他点亮的不是固定的口。因此要采用 Y:a 模式，当 a=1, Y:a 等价于 Y01

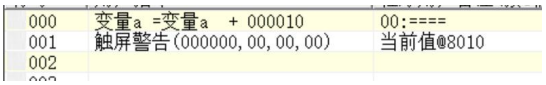
编程:

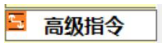


示例 2：每次运行 a 加 10，并在屏幕显示 a 的值



编程:



 **触屏警告** 语句是从 

中的电脑调试指令中按出来，他会把注释区的文字打印在屏幕上(位置不能更改)，同时注释区支持文字和变量的混合显示。此功能一般用作调试。

1.2.3 掉电保存型变量

掉电保存型变量指，关机以后数据还保存的（无需纽扣电池），正常情况下可以保存一百年，每个位置能至少能擦除 1000 万次。系统有二类掉电保存型变量，一类是 菜单，一类是硬盘数据，既 A 盘 B 盘，C 盘，D 盘。

菜单：

面向一线工人输入数据，最多 44 行，每行支持 18 个自定义中文名。

菜单变量读取示例：

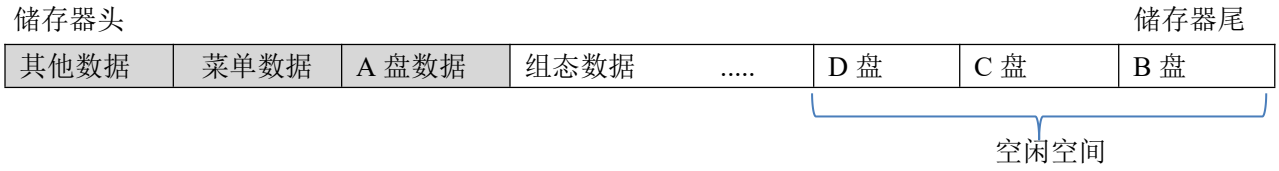
```
000    0机正转菜单00 速010000 等停
001    等：菜单00秒
002    0机正转菜单01 速010000 等停
```

运行效果：0 机先正转一段距离，再延迟一段时间，再次正转一段距离后停止，正转的长度和延迟的时间是不确定的，由用户输入的菜单值决定。而且菜单数据断电后会自动保存在硬盘中。菜单数据也同样和普通变量一样支持运算。


所有变量都不支持小数点,菜单比较特殊,他展现给客户虽然是带小数点的,其实内部保存的是去除小数点后的值,比如 1.0 内部值为 10;　001.00 内部值为 100

硬盘数据：

硬盘数据和菜单数据很像，区别是，菜单数据带可自定义的菜单名，硬盘数据没有菜单名。本系统把硬盘分为 ABCD 盘，其中 A 盘空间是固定 100 个数据，BCD 三个盘把空闲空间等分了。 数据存储结构 如下图所示：



上表中的灰色项为固定地址，组态以后的空间为剩余空间，注意 BCD 盘为从尾部开始保存，假设空闲空间为 3000，那么 BCD 每个盘都为 1000 个数据，B 盘第一个数据保存在储存器的倒数第一位置，C 盘的第一个数据在倒数第 1000 的位置，D 盘的第一个数据在倒数第 2000 的位置，但组态的数据每个项目大小不一，如果组态数据过大，会覆盖 D 盘数据，如果再大，会覆盖 C 盘数据。因此空闲硬盘的大小取决于组态占用的多少。

在空闲空间 3000 情况下，假设 NP 参数(每个盘大小)设置了 1500，那么就意味着，D 盘将无法使用。因为 B 盘和 C 盘已经占用了全部的 3000，C 盘以后的数据将会是组态数据而无法读写（屏幕上也会有报警信息）


硬盘变量适合保存示教时候的点位数据，他没有数据格式，在初始化后，可保存 3000 个数字，如果一个坐标占用 2 个数字，那么共可以保存 1500 个点，请看 **示教参考视频**。

1.2.4 系统变量

变量具体含义表：

变量大类	含义
输出	读写端口的输出 00～32
输入	读写端口的输入 00～32
坐标	预留
系统	当前系统变量，比如已循环的次数,CPU 使用率等等
变量	用户自由设定的变量 a~z
菜单	读写用户的菜单数据，掉电不丢失数据
设置	修改系统内部的设置，比如屏幕亮度，蜂鸣器等，掉电不丢失
...	

每个变量类中包含变量的名称（属性和方法），就拿系统这个变量类来举例，当选中系统类后会跳出如下变量名：

变量名	备注
00 本次工作次数	
01 永久工作次数	
02 开机总时间	(毫秒)
03 开机总时间	(秒)
04 当前时间	单位:时
05 当前时间	单位:分
06 当前时间	单位:秒
07 当前时间	单位:0.1 秒
08 运行中线程数	0~4
09CPU 运行速度	
10 当前页面	
11 最后感应是哪个口	
12 最后感应口状态	
13 本循环体已运行次数	第一次运行时，读数为 0
14 进程号	0~4
15 本事件已运行次数	第一次运行时，读数为 0
16 本事件剩余次数	当在运行最后一遍时，读数为 0
17 本循环体剩余次数	当在运行最后一次时，读数为 0
21 运行锁(0:解锁 1:锁)	上锁后无法运行程序(Z 除外)。当硬件运行锁闭合后，此参数无效
22 距离上次触摸的时间(秒)	
23 端口(0~23 号)状态(二进制)	以二进制的形式同时控制或读取 24 位端口输出状态
24 输入（0~23 号）状态	比如 012 号感应器有信号，此值为(二进制) 1110 0000 0000 0000 0000 0000
25 有效菜单数	用户自定义了几行菜单
26 电机手动速度全局开关(1 快 0 慢)	当 1 的时候，手动测试时所有电机都临时调用快速速度
27 最后触碰的按钮序号	一般用作事件触发

28 总共按下几次触屏	一般用作事件触发
29 总共弹起几次触屏	一般用作事件触发
30 总共几次长按	一般用作事件触发
31 总共几次键盘数据保存	一般用作事件触发
32 键盘输入的数字	一般用作事件触发

1.2.5 变量运算

在指令板选择变量操作命令就能对变量进行一些运算，注意这里只支持一次运算，如需要多次运算，请多行输入。变量运算其实比较简单，大部分情况下只用到加减乘除，详细说明请点击这里

1.2.6 变量赋值

比如 a=00000;就是变量赋值，其实赋值和变量运算是同一个指令，**只要把运算符设置为空格**就可以，需要注意的是，对于某些系统变量，比如你写：
输出 01=1；端口 1 就会输出,用高级语言的术语说就是执行一个“过程”。
输入 01=1;那么系统内部会模拟输入被触发一次，这可实现某些进程的”软启动”,比较有用。
系统 10=1;跳到用户页面 1；系统 10 指当前的页面,可读可写。变量的用法非常灵活，以下是变量操作的一些例子，大家可以举一反三:

A)实现当循环最后一次后，打开端口 1：

```
当输入21＝000001次数000001
循环开始 次数:000010
如系统17＝000000打开Y000001    本循环体剩余次数
001.0000秒
循环结束
```

程序中系统 17 代表本层循环体剩余次数,而传统写法就必须浪费一个变量来实现了，如以下程序：

```
当输入21＝000001次数000001
变量a =000010
循环开始 次数:000010
变量a =变量a - 000001
如变量a ＝000000打开Y000001
001.0000秒
循环结束
```

B)流水灯

用端口输出指令 Y00 只能单个端口，如果要实现群控，就必须使用系统变量功能了，以下程序实现灯 0~19 之间流水显示。非常简洁。

行号	用户指令
000	循环开始 次数:000020
001	系统23=000001 < 系统13
002	000.1000秒
003	循环结束
004	

系统 23 代表 24 位输出端口状态

系统 13 代表当前的已循环次数，会从 0 累加到 19

< 代表移位运算 ，类似 C 语言中<<操作符号

视频演示

A) 当 X1 闭合,1 灯亮；当 X2 闭合,2 灯亮；当 X3 闭合,3 灯亮；.....

行号	用户指令	程序用户备注(按@输入变量)
000	当系统11±000001次数000001	当端口有变化了
001	变量a =系统11	读取最后感应口号赋值给a
002	输出a =000002	开灯

B) 当用户修改菜单 00 行后，模拟量端口 D1 输出设定的电压

行号	用户指令	程序用户备注(按@输入变量)
000	当菜单00±000001次数000001	假如菜单第0行有变动
001	DA1.11=菜单01	将菜单值写入DA值输出电压
002		

C) 通过外部按钮修改屏幕亮度

行号	用户指令	程序用户备注(按@输入变量)
000	当输入21=000001次数000001	按下F1设亮度最暗
001	设置08=000001	屏幕亮度=1
002	当输入22=000001次数000001	按下F1设亮度最亮
003	设置08=000100	屏幕亮度=100
004		

D) 通过外部按钮设置蜂鸣器开关(触摸震动)

行号	用户指令	程序用户备注(按@输入变量)
000	当输入21=000001次数000001	
001	设置03=000001	蜂鸣器开
002	当输入22=000001次数000001	
003	设置03=000000	蜂鸣器关

E) 按 X0,Y0 端口输出 / 按 X1,Y1 端口输出 / 按 X2,Y2 端口输出 / 按 X3,Y3 端口输出.....

假如用传统的方法写，每个按键写一个事件，如果控制 24 个口，那么程序要接近 48 行。用变量来编程就很优雅，仅需 3 行！而且有变量提示和自动产生注释，你完全不用像 PLC 一样，哪个寄存器对应哪个功能。

行号	用户指令	程序用户备注(按@输入变量)
000	当系统12=000001次数000001	最后感应口状态为按下时触发
001	变量a =系统11	最后感应是哪个口赋值给a
002	Y:a=2	

当然你可以加上条件判断，比如只对前 4 个按键有用。也是很简单的。

F) 档位开关转到 0 跳到第 0 页//转到 1 跳到第 1 页//转到 2 跳到第 2 页。档位开关如下:

以上的档位开关就 2 个触点，假设到 **1** 的位置 X21 导通，到 **2** 的位置 X22 导通，到 **0** 的位置 2 个触点都不导通，我们可以这样写程序:

行号	用户指令	程序用户备注(按@输入变量)	调试窗口
000	当输入21＝000001次数000001	1号档位	255
001	系统10=000001	跳到第1页	
002	004行空白		
003	当输入22＝000001次数000001	2号档位	
004	系统10=000002	跳到第2页	
005			
006	当输入21＝000000次数000001	从1号档位转到0号	
007	系统10=000000	跳到首页	
008	当输入22＝000000次数000001	从1号档位转到0号	
009	系统10=000000	跳到首页	
010			
011			

提示：具体数字代表哪个页面如下图所示：

页面号	数值	注释
用户可编辑页面	0～19	实际上可用是 0~9
手动测试第 1 页	20	系统内置，无法更改
手动测试第 2 页	21	系统内置，无法更改
系统设置页	22	系统内置，无法更改
代码浏览页	23	系统内置，无法更改
电机设置	24	系统内置，无法更改
登录界面	25	系统内置，无法更改

易控派的页面时仅需 80 毫秒的，就屏幕刷新速度这块可以比肩高性能的文本显示器。

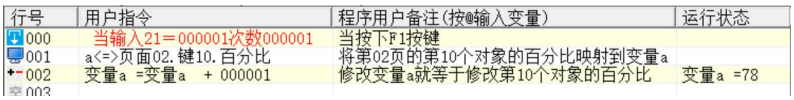
I)在首页按 F1 能控制端口 0，在其他页按了没反应

那首先要判断当前页面号，通过读取 系统 10

行号	用户指令	程序用户备注(按@输入变量)
000	当输入21＝000001次数000001	当接受到启动信号
001	如系统10＝000000	判断当前页是否在首页
002	Y00=2 零端口●	只有在首页才能运行
003	判断结束	

1.2.7 变量映射

可以简单理解为改名(其实内部机制比较复杂)，既把一个比较长的变量名给他别名比如 a,后期对 a 的读写操作其实就是操作原来的变量。他和变量赋值有点像，如要了解想看后面的指令表中的变量映射指令。用法演示：



行号	用户指令	程序用户备注(按@输入变量)	运行状态
000	当输入21＝000001次数000001	当按下F1按键	
001	a<=>页面02.键10.百分比	将第02页的第10个对象的百分比映射到变量a	
002	变量a =变量a + 000001	修改变量a就等于修改第10个对象的百分比	变量a =78
003			

当 a 变化后会实时反应到进度条的百分比上，记住当映射以后，不管是读和写都是针对刚刚映射的对象，这个指令主要用于控制屏幕对象，也可以是其他变量(比如菜单，电机坐标等等，但不太建议，因为会影响阅读），用法比较灵活，需自己摸索。