

MAY3S

MAY5S

# 智能控制面板

# 说明书

版权所有：深圳市顾美科技有限公司

V8.41 版

# 目录

一、产品概述.....	4
1.1 主要特点.....	4
1.2 产品型号参数.....	5
二、硬件参数.....	6
2.1 外形安装尺寸.....	6
2.1.1 MAY3S 外形安装尺寸.....	6
2.1.2 MAY5S 外形安装尺寸.....	6
2.2 接线说明.....	7
2.2.1 接线示意图.....	7
2.2.2 接线方式.....	7
2.3 继电器/SSR 输出编程地址.....	7
三、协议说明.....	8
3.1 CANBUS 协议说明.....	8
3.1.1 通讯参数设置.....	8
3.1.2 智能控制面板与 coolmay 厂家 CX3G 系列 PLC 进行 CAN 通讯设置.....	8
3.1.3 智能控制面板中参数设置.....	10
3.2 Modbus RTU 协议说明.....	15
3.2.1 智能控制面板做主机只带一台从机的设置.....	15
3.2.2 智能控制面板做主机带多台从机的设置.....	16
3.2.3 智能控制面板做从机的设置.....	19
3.2.4 其他特殊寄存器.....	21
3.3 屏与屏 CAN 口通讯说明.....	21
四、宏指令功能说明.....	22
4.1 宏的种类.....	22
4.3 宏的编辑.....	23
4.4 宏的操作数.....	25
4.5 错误讯息.....	31

## 一、产品概述

### 1.1 主要特点

- 1、6 万真彩，TFT 高清真彩显示器；
- 2、支持 WiFi 功能，轻松实现其他设备无线接入；
- 3、支持 MODBUS、CANBUS，TCP/IP 等协议，可实现内部组网；
- 4、支持在线编程，自定义在家、一键“外出”物联网智能家居；
- 5、提供大量的矢量图库及支持自建图库，支持各种格式静态图片及 GIF 动画，可制定不同风格的控制界面，专业用户可真正实现自己的智能家居 DIY；
- 6、MAY3S 采用 86 底盒安装方式；MAY5S 采用 T144 底盒安装方式；
- 7、MAY3S 可选 4 路继电器/SSR 输出，4 路同时最大 10A，每路最大 5A；MAY5S 可选 8 路继电器/SSR 输出，8 路同时最大 10A，每路最大 5A；
- 8、外观高亮银色切边，磨砂太空铝边框；
- 9、超强功能，可直接代替传统墙壁开关，可控制照明、插座、窗帘等，支持手机 APP 控制。

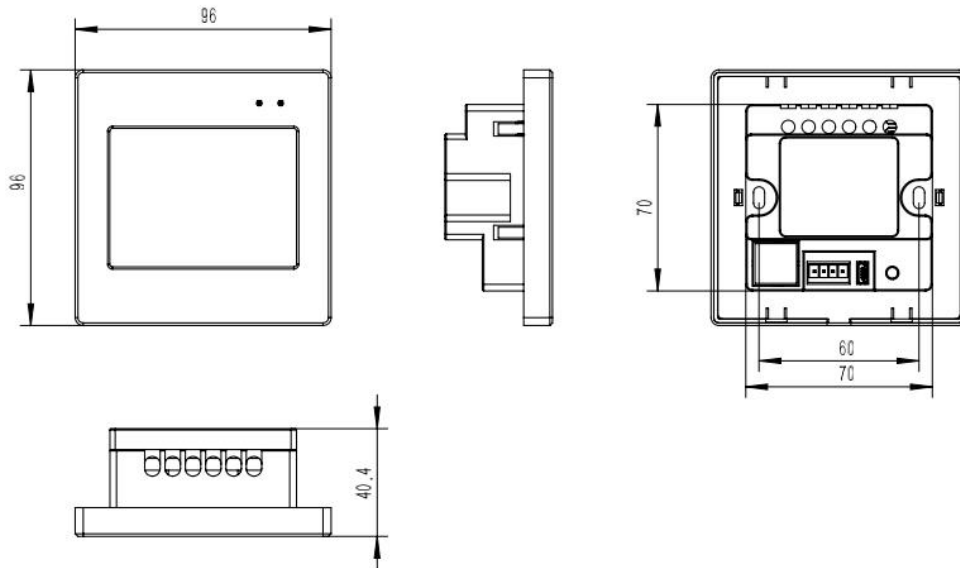
## 1.2 产品型号参数

### 二、硬件参数

产品类型		MAY3S	MAY5S
产品图片	正面		
	背面		
规格	外形尺寸	96*96*40.4mm	154*96*40.4mm
	开孔尺寸	70*70mm	134*70mm
	显示尺寸	73*56mm	108*65mm
	重量	约 0.3 kg	约 0.33 kg
显示	显示类型	3.5" TFT	5" TFT
	分辨率	320*240	800*480
	显示亮度	300cd/m <sup>2</sup>	
	对比度	400:1	
	背光类型	LED	
	背光寿命	60,000 小时	
	显示色彩	65536 真彩	
内存	触控类型	4 线电阻屏	
	存储内存	128MB	
	运行内存	32MB	
通讯接口	CPU	ARM9 内核 216MHz	
	USB Host	有	
	USB Client	USB 2.0×1	
	以太网口	可选装	
	通讯端口	RS485 口/RJ45 网口	
通讯协议	支持 MODBUS 协议、CANBUS、自由口和常规的 PLC 通讯协议		
继电器输出	4 路 10A (最大)	8 路 10A (最大)	
负载	4 路 1000W 开关负载	8 路 1000W 开关负载	
安装底盒	国标 86 底盒	国标 T144 底盒	
万年历	有		
通讯方式	WiFi, 有线 RS485, CANBUS		
输入电压	零火线 AC220V		
防护等级	IP65 (前面板)		
环境温度	工作环境 0~50℃; 存储环境 -20~70℃		
环境湿度	20%~90%RH		
认证	通过 CE 认证		
安装方式	墙面嵌入式安装		
墙外厚度	6mm		
软件	CoolMay HMI 智能控制面板编辑软件 详见《CoolMayHMI 智能控制面板使用手册》《智能控制面板用户手册》		

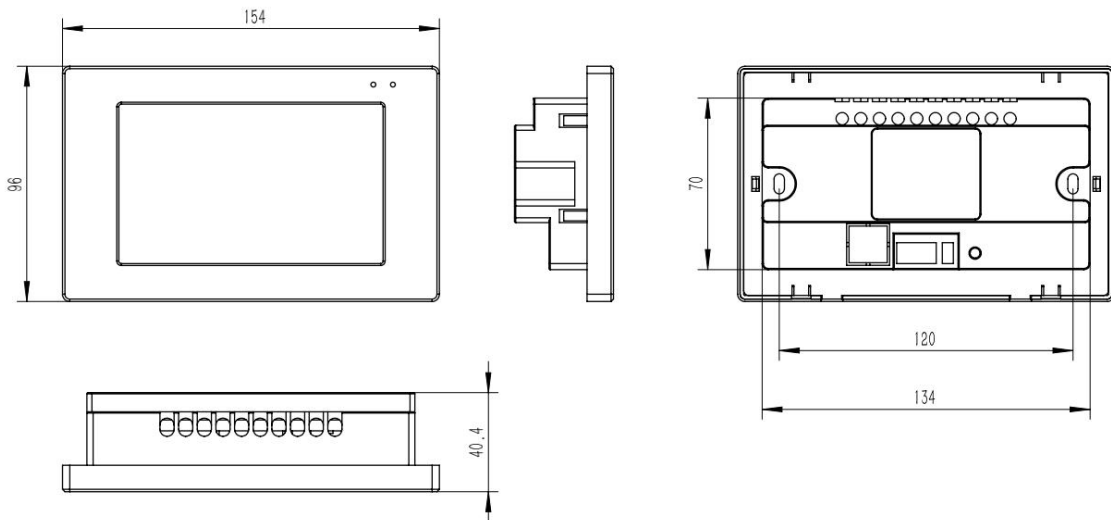
## 2.1 外形安装尺寸

### 2.1.1 MAY3S 外形安装尺寸



图一：MAY3S 外形安装尺寸

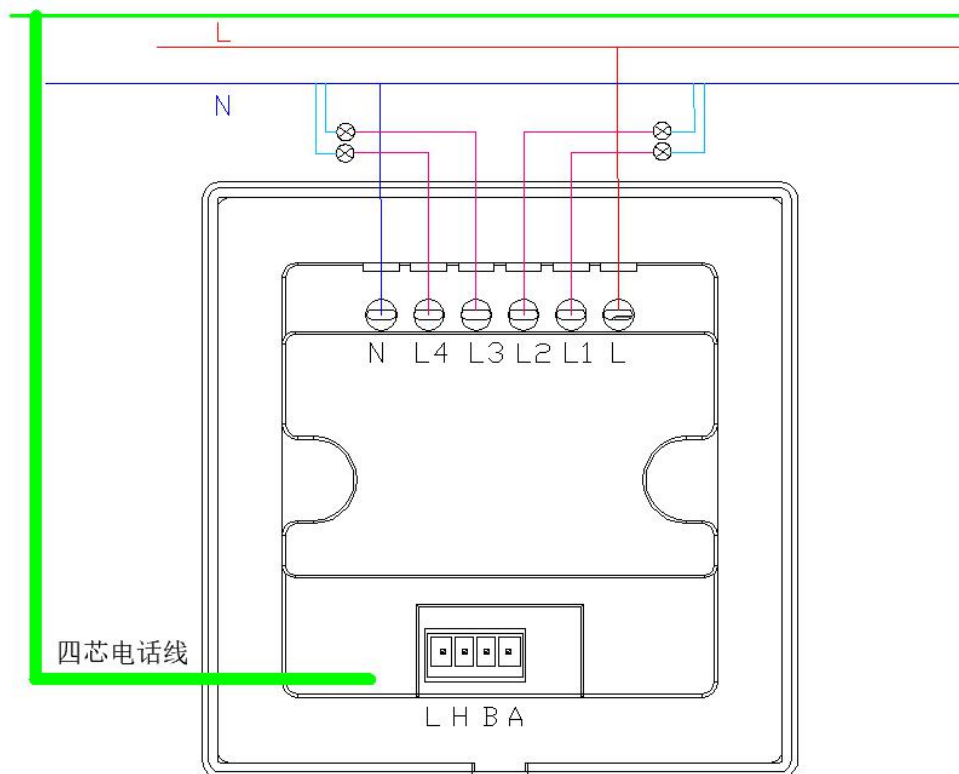
### 2.1.2 MAY5S 外形安装尺寸



图二：MAY5S 外形安装尺寸

## 2.2 接线说明

### 2.2.1 接线示意图



图三：接线示意图

### 2.2.2 接线方式

背后 6 个接线柱，两边分别为零火线。中间为设备的控制线，设备的零线与控制面板的零线接通。另外控制面板背后的通讯端口处有四个接线排。从此图看从左至右依次是 L、H（CAN 口）。B、A（485 口）。

注：网口上的 12 针脚为 CAN 口的 L B；45 针脚是 RS485 口的 A B。

## 2.3 继电器/SSR 输出编程地址

继电器输出地址对应表：

继电器输出	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
寄存器地址	LW8265.0	LW8265.1	LW8265.2	LW8265.3	LW8265.4	LW8265.5	LW8265.6	LW8265.7

## 三、协议说明

### 3.1 CANBUS 使用说明

#### 3.1.1 通讯参数设置

OP 参数设置中:通讯端口: CAN, 通讯速率: 250K, CAN 格式: 2.0B, RTR 类型: 数据, 如下图:



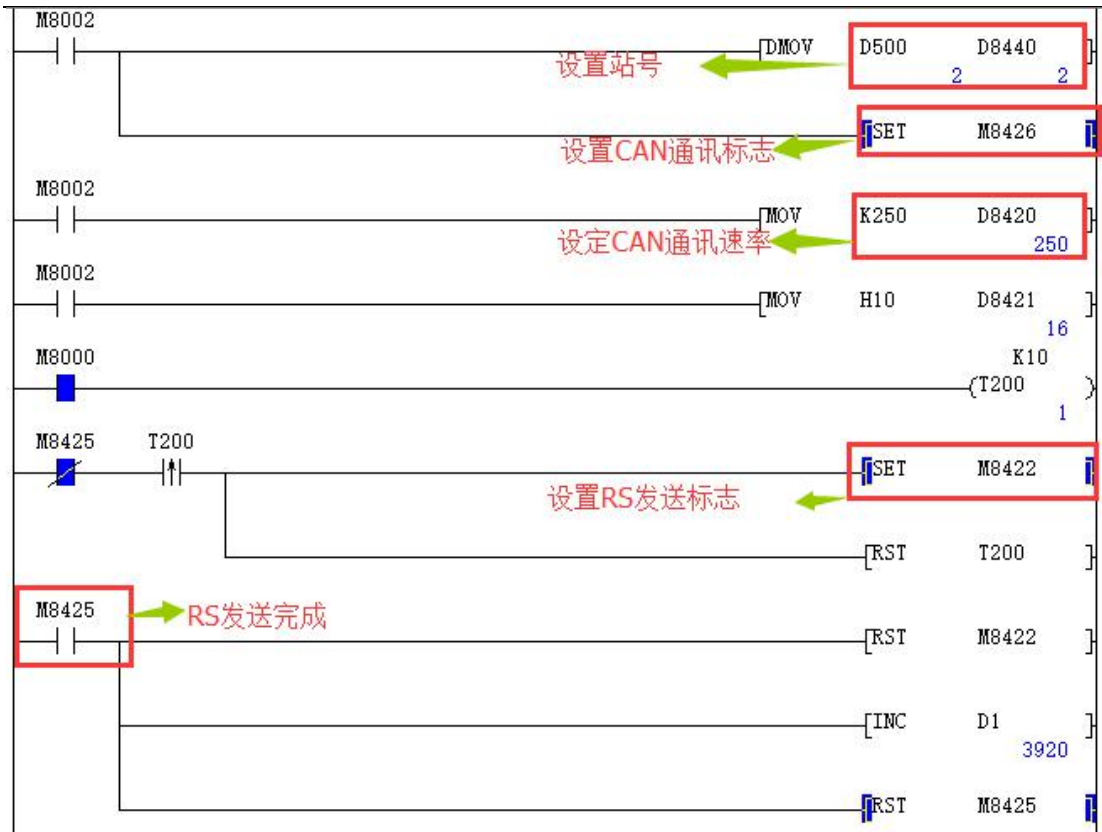
**注: 设备 ID 只能设置为 0。**

若设置为其它值, 与 PLC 进行 CAN 通讯, 不能发送数据, 能接收数据, 但只接收数据, 不能接收到 PLC 的站号。

若设置为其它值, 进行 CAN 与 CAN 通讯, 不能发送数据, 因为不能发送数据, 故没有数据可以接收。

#### 3.1.2 智能控制面板与 coolmay 厂家 CX3G 系列 PLC 进行 CAN 通讯设置

PLC 部分 CAN 通讯程序参考 coolmay 厂家 CX3G 系列 PLC 编程手册 CAN 通讯部分。以下截图为该 PLC 程序设置说明。如下所示:



PLC 中部分设定发送数据程序如下所示:





### 3.1.3 智能控制面板中参数设置

注意：设备 ID 必须设置为 0。

设定OP工作参数

通讯设置 网络设置 报警/其它 字形/语言 记录缓冲区 配方设置

**一般**

人机界面参数: MT6037H (320\*240) 人机界面匹配型号选择表

移动U盘数据上载/下载权限: 超级  Link2 使用

**Link1 设定**

通讯端口: CAN 设备类型: Free Protocol

通讯速率: 250K 通讯超时: 100 ms 设备 ID: 0

CAN 格式: 2.0B RTR 类型: 数据

尝试次数: 8 快速读取区D: 0 数据长度: 5

宏设置如下所示：

1	#条件判断
2	IF LW100.0 == ON THEN GOTO LABEL 1
3	IF LW100.1 == ON THEN GOTO LABEL 2
4	END
5	#
6	LABEL 1
7	CLRB LW100.0
8	#发送帧8/12字节
9	TXD( LW1000, 12)
10	END
11	#
12	#
13	#接收数据到LW1500..LW1755
14	LABEL 2
15	CLRB LW100.1
16	#RXD的参数A2若设置为LW寄存器, 则需A2>=(A1+255).
17	#接收数据长度大于255时, 会清除掉最先接收的255个数据
18	RXD( LW1500, LW1756)
19	END
20	

发送数据的宏指令为 TXD。

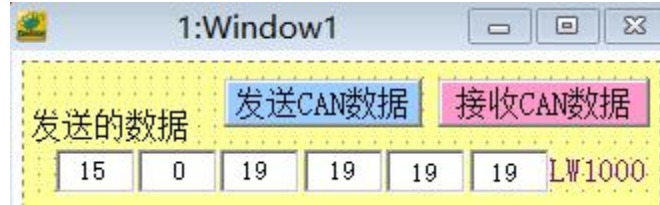
TXD(A1, A2)

A1 表示发送数据的起始地址，此处一般指 LW 寄存器。

A2 表示发送数据的字节数。

注：发送数据时，按字(Word)发送。(1 个 Word = 2 字节)此处的字可以理解为 1 个 LW 寄存器。

设置发送的数据如下图所示：



点击【发送 CAN 数据】按钮，（发送 12 个字节，其中前两个寄存器占用的是 ID 号，后面四个寄存器共 8 个字节的数据传送的 PLC），具体传送结果如下图所示。

软元件	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
D800	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 1	19
D801	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
D802	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 1	19
D803	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
D804	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 1	19
D805	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
D806	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 1	19
D807	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

接收数据的宏指令为 RXD。

RXD(A1, A2)

A1 表示接收数据到 A1 指定的开始地址，此处一般指 LW 寄存器。

A2 设置为 K 时：表示读取 K 个字节。

A2 设置为 LW 时：表示读取缓冲区的全部字节。

A2 的地址值取值：A2 >= A1+255

A2 地址中存放读取的字节数；

A2+1 地址中存放资料格式；

设置 A2+1=0：表示收到的数据按字节排列。

设置 A2+1=1：表示收到的数据按字排列，高字节在前。

设置 A2+1=2：表示收到的数据按字排列。低字节在前

以下设置按照 RXD(LW1500, LW1755)进行数据返回。

1、将 LW1756 设置为 0 时，接收的数据按字节排列

点击【接收 CAN 数据】按钮，程序界面如下所示：



PLC 的 ID(占 4 个寄存器)及 D600-D607(占 8 个寄存器)共 12 个字节接收，存放在 LW1500 开始的寄存器中。

解释：ID 号为 15，占了 4 个寄存器。PLC 中 D600-D607 的数据为：11 12 11 12 11 12 11 12.

2、将 LW1757 设置为 1 时。

点击【接收 CAN 数据】按钮，数据接收如下图所示。



前 2 个寄存器显示的为 ID 号。

后面 4 个寄存器显示的为 PLC 中 D600~D607 的数据。

数据显示方式：10 进制

数据显示内容：1 个 LW 寄存器显示 2 个 D 寄存器的数据，高字节在前。

解释：此时 PLC 中 D600~D607 的数据为

10 进制表示：11 12 11 12 11 12 11 12

16 进制表示：0B 0C 0B 0C 0B 0C 0B 0C

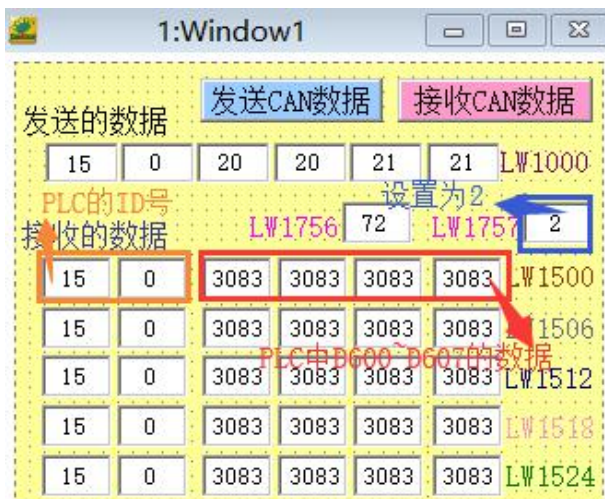
低位在前高位在后存入 LW 的 4 个寄存器中

LW 的 16 进制表示为：B0C B0C B0C B0C

转换为 10 进制表示为：2828 2828 2828 2828

3、将 LW1757 设置为 2 时。

点击【接收 CAN 数据】按钮，数据接收如下图所示。





前 2 个寄存器显示的为 ID 号。

后面 4 个寄存器显示的为 PLC 中 D600~D607 的数据。

数据显示方式：10 进制

数据显示内容：1 个 LW 寄存器显示 2 个 D 寄存器的数据，低字节在前。

解释：此时 PLC 中 D600~D607 的数据为

10 进制表示：11 12 11 12 11 12 11 12

16 进制表示：0B 0C 0B 0C 0B 0C 0B 0C

低位在前高位在后存入 LW 的 4 个寄存器中

LW 的 16 进制表示为：C0B C0B C0B C0B

转换为 10 进制表示为：3083 3083 3083 3083

以下举例将 RXD 参数 A2 设置为 K

13	#接收数据到LW1500..LW1755
14	LABEL 2
15	CLRB LW100.1
16	#RXD的参数A2若设置为LW寄存器，则需A2>=(A1+255).
17	#接收数据长度大于255时，会清除掉最先接收的255个数据
18	RXD(LW1500, 24)
19	END
20	

此处设置为24，表示只接收24个字节的数据

点击【接收 CAN 数据】按钮，数据接收如下图所示。



## 3.2 Modbus RTU 协议说明

### 3.2.1 智能控制面板做主机只带一台从机的设置

#### 1、通讯参数设置：

1) 打开软件菜单栏的“应用——设定工作参数——通讯设置”。

2) 在 link1 或 link2 设定中如下：

**通讯端口：**使用 MAY3S 或者 MAY5S 通讯的时候，选 COM1 表示使用的是智能控制面板的 RS485 口，选 COM2 表示使用的是智能控制面板的 CAN 口。

**设备类型：**Modbus RTU Slave。

**通讯速率、通讯超时、检查位元、资料位元、停止位元：**根据自己的通讯需求，主机从机设置一致即可。

**设备 ID：**即要读取的从站号。

设定OP工作参数

**一般**

人机界面参数: MT6037H (320\*240)

移动U盘数据上载/下载权限: 超级  Link2 使用

---

**Link1 设定**

通讯端口: COM1      设备类型: Modbus RTU Slave  
 通讯速率: 9600      通讯超时: 200 ms      设备 ID: 1  
 检查位元: 无      资料位元: 8 b      停止位元: 1 b  
 尝试次数: 8      快速读取区4x: 0      数据长度: 0

---

**Link2 设定**

通讯端口: COM2      设备类型: Mitsubishi MELSEC-Q00 (CPU)  
 通讯速率: 9600      通讯超时: 200 ms      设备 ID: 1  
 检查位元: 无      资料位元: 8 b      停止位元: 1 b  
 尝试次数: 8      快速读取区D: 0      数据长度: 0

## 2、多机通信设置：

1) 打开软件菜单栏的“应用---设定工作参数---网络设置”。

控制器 ID 地址模式：选择默认的标准模式。



## 3、智能控制面板做 modbus 主机时支持的功能码：

- 01 号功能：读取线圈状态，取得一组逻辑线圈的当前状态（ON/OFF）
- 03 号功能：读取保持寄存器，在一个或多个保持寄存器中取得当前二进制值
- 05 号功能：强置单线圈，强置一个逻辑线圈的通断状态（写位）
- 06 号功能：把具体二进值装入一个保持寄存器（写寄存器）
- 16 号功能：预置多寄存器，把具体的二进制值装入一串连续的保持寄存器（写多个寄存器）

### 3.2.2 智能控制面板做主机带多台从机的设置

#### 注意：

- 1、智能控制面板做主机带多台从机时，智能控制面板上或智能控制面板 PLC 一体机的智能控制面板上都必须使用 RS485 口，且必须在 link1 设定中设置通讯端口为 COM1（即 RS485）；
- 2、Link2 设定不支持做主机带多台从机的模式。

## 1、通讯参数设置：

1) 打开软件菜单栏的“应用---设定工作参数---通讯设置”。

2) 在 link1 设定中如下：

**通讯端口：**COM2。

**设备类型：**Modbus RTU Slave。

**通讯速率、通讯超时、检查位元、资料位元、停止位元：**根据自己的通讯需求,主机从机设置一致即可。

**尝试次数：**默认为 8 次，范围 1-99，即读取每台从机的次数。

若尝试次数为 8 次，当有任一从机未连接时，主机都会尝试读取 8 次，每次时间为通讯超时设置的时间（默认为 200ms）。在读取 8 次后，还未通讯成功的，则主机不再访问该站号的从机，需要重启主机之后才会再次访问 8 次该站号的从机。若通信失败提示勾选时，会提示通信失败；

若尝试次数修改为 1，则无论从机是否连接，每次都访问 1 次从机，提高主机的通讯速度。若通信失败提示勾选时，不提示连接失败。

设定OP工作参数

通讯设置 网络设置 报警/其它 字形/语言 记录缓冲区 配方设置

一般

人机界面参数: MT6037H (320\*240) 人机界面匹配型号选择表

移动U盘数据上载/下载权限: 超级  Link2 使用

**Link1 设定**

通讯端口: COM1 设备类型: Modbus RTU Slave

通讯速率: 9600 通讯超时: 200 ms 设备 ID: 1

检查位元: 无 资料位元: 8 b 停止位元: 1 b

尝试次数: 8 快速读取区 4x: 0 数据长度: 0

**Link2 设定**

通讯端口: COM2 设备类型: Mitsubishi MELSEC-Q00 (CPU)

通讯速率: 9600 通讯超时: 200 ms 设备 ID: 1

检查位元: 无 资料位元: 8 b 停止位元: 1 b

尝试次数: 8 快速读取区 D: 0 数据长度: 0

确认 (Y) 应用 (A) 取消 (N)

## 2、多机通信设置：

1) 打开软件菜单栏的“应用---设定工作参数---网络设置”。

**控制器 ID 地址模式：**选择扩展模式。

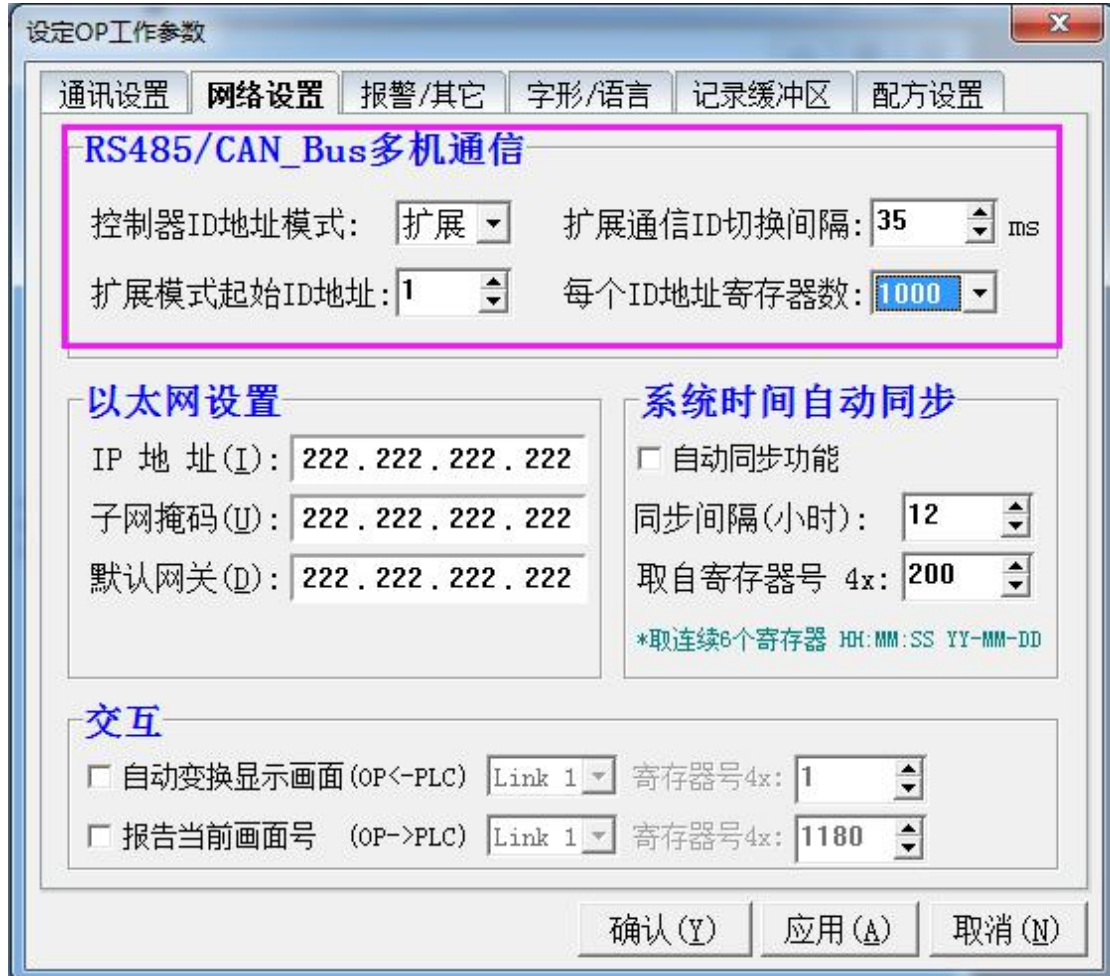
**扩展通信 ID 切换间隔：**默认为 35ms，可根据实际通讯调整。

**扩展模式起始 ID：**默认为 1，即屏连接的从机的第一个从机站号。



**每个 ID 地址寄存器数：**100-30000 范围可设，根据每个从站的实际寄存器范围设置。

如下图设置的表示：屏连多从机，第一个从机站号是从 1。每个 ID 地址寄存器数设置 1000 时，4x0-4x999 表示 1 号从机 0-999 的地址寄存器，4x1000-4x1999 表示 2 号从机的 0-999 的寄存器地址，4x2000-4x2999 表示 3 号从机的 0-999 的寄存器地址……以此类推。



### 1.2.3 智能控制面板做 modbus 主机时支持的功能码：

- 01 号功能：读取线圈状态，取得一组逻辑线圈的当前状态 (ON/OFF)
- 03 号功能：读取保持寄存器，在一个或多个保持寄存器中取得当前二进制值
- 05 号功能：强置单线圈，强置一个逻辑线圈的通断状态(写位)
- 06 号功能：把具体二进值装入一个保持寄存器(写寄存器)
- 16 号功能：预置多寄存器，把具体的二进制值装入一串连续的保持寄存器(写多个寄存器)



### 3.2.3 智能控制面板做从机的设置

#### 1、通讯参数设置：

1) 打开软件菜单栏的“应用---设定工作参数---通讯设置”。

2) 在 link1 或 link2 设定中如下：

**通讯端口：**使用 MAY3S 或者 MAY5S 通讯的时候，选 COM1 表示使用的是智能控制面板的 RS485 口，选 COM2 表示使用的是智能控制面板的 CAN 口。

**设备类型：**Modbus RTU Master。

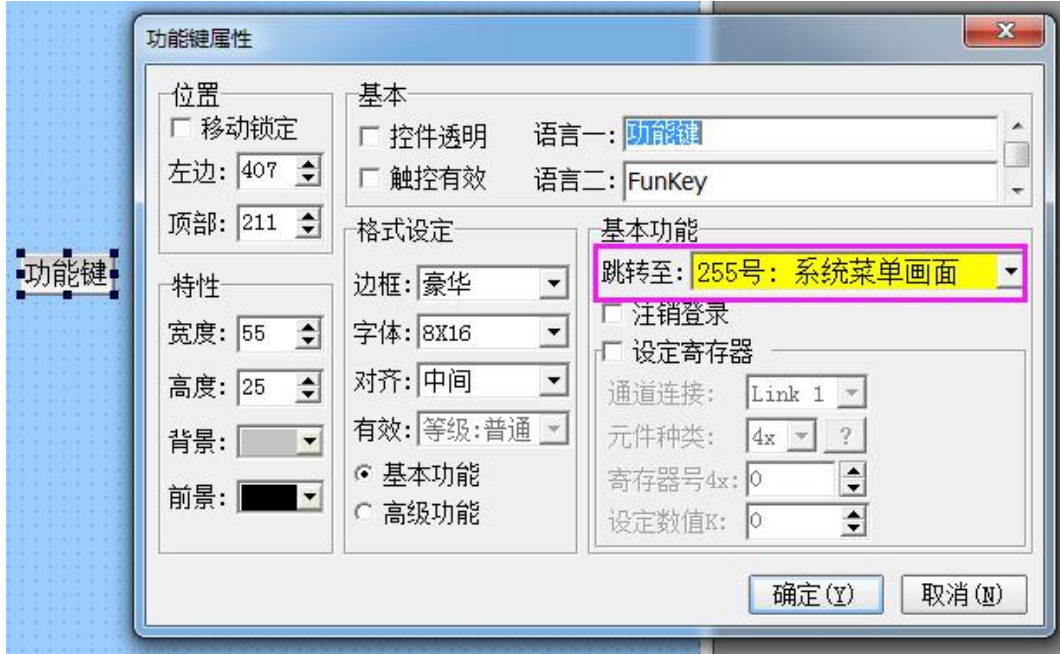
**通讯速率、通讯超时、检查位元、资料位元、停止位元：**根据自己的通讯需求，主机从机设置一致即可。



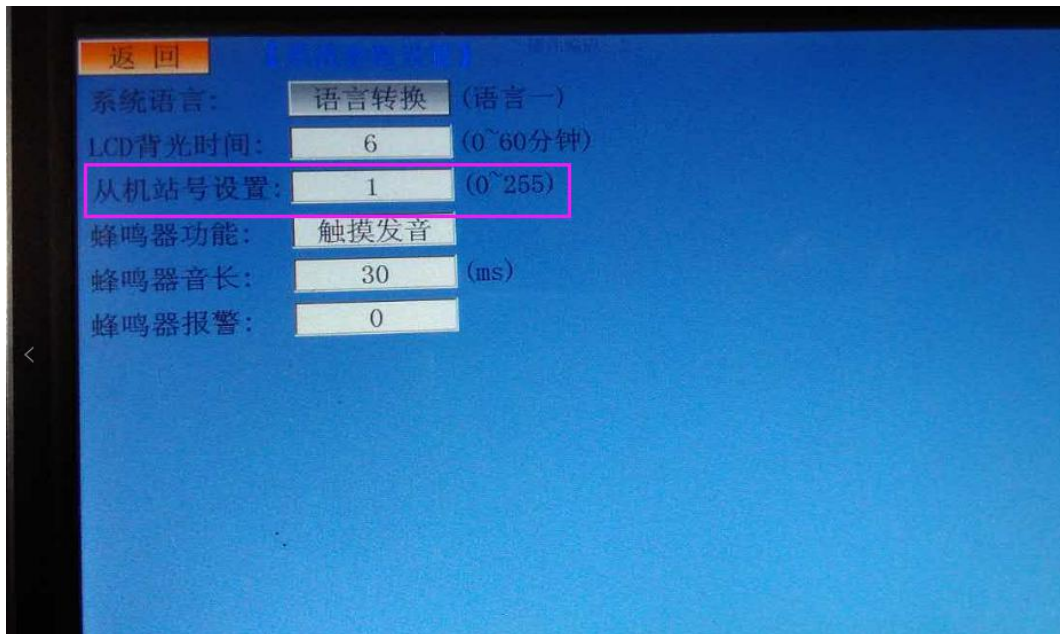
## 2、从机站号设置:

### 设置方法一:

1) 在画面程序上放一个功能键跳转到 255 号系统菜单页面, 然后把程序编译下载到智能控制面板, 如下图所示:



2) 在智能控制面板上点该功能键跳转到 255 号系统菜单画面, 即可设置从站号。



**设置方法二：**

内部寄存器：LW8206……多机通信间隔，即从机站号设定地址，范围：1-255；在智能控制面板上添加寄存器连接该地址，如下图：



**3、从站寄存器地址范围：**

- 1) 数据寄存器地址范围：4x0-4x65000
- 2) 位地址范围：0x0-0x65000

**4、智能控制面板做从机时支持的功能码：**

- 01 号功能：读取线圈状态，取得一组逻辑线圈的当前状态 (ON/OFF)
- 03 号功能：读取保持寄存器，在一个或多个保持寄存器中取得当前二进制值
- 05 号功能：强置单线圈，强置一个逻辑线圈的通断状态(写位)
- 06 号功能：把具体二进值装入一个保持寄存器(写寄存器)
- 16 号功能：预置多寄存器，把具体的二进制值装入一串连续的保持寄存器(写多个寄存器)

**3.2.4 其他特殊寄存器**

- 内部寄存器：LW8246……Link1 通信超时计数器(仅限于 ModBus RTU Master 时使用)
- 内部寄存器：LW8247……Link2 通信超时计数器(仅限于 ModBus RTU Master 时使用)
- 内部寄存器：LW8248……Link1 通信成功计数器
- 内部寄存器：LW8249……Link2 通信成功计数器

### 3.3 面板与面板之间的 CAN 口通讯

#### 3.3.1 CAN 口通讯无主机通讯设置

##### 1、通讯参数设置：

1) 打开软件菜单栏的“应用——设定工作参数——通讯设置”。

2) 在 link1 或 link2 设定中如下：

**通讯端口：**使用 MAY3S 或者 MAY5S 通讯的时候，选 COM1 表示使用的是智能控制面板的 RS485 口，选 COM2 表示使用的是智能控制面板的 CAN 口，此时选择 COM2 口。

**设备类型：**Modbus RTU Master。

**通讯速率、通讯超时、检查位元、资料位元、停止位元：**根据自己的通讯需求，主机从机设置一致即可。

设定OP工作参数

通讯设置 | 网络设置 | 报警/其它 | 字形/语言 | 记录缓冲区 | 配方设置

一般

人机界面参数: MT6037H (320\*240) | 人机界面匹配型号选择表

移动U盘数据上载/下载权限: 超级 |  Link2 使用

**Link1 设定**

通讯端口: COM2	设备类型: Modbus RTU Master	
通讯速率: 115200	通讯超时: 200 ms	设备 ID: 1
检查位元: 无	资料位元: 8 b	停止位元: 1 b
尝试次数: 8	快速读取区 4x: 0	数据长度: 0

**Link2 设定**

通讯端口: COM2	设备类型: Modbus RTU Slave	
通讯速率: 9600	通讯超时: 200 ms	设备 ID: 1
检查位元: 无	资料位元: 8 b	停止位元: 1 b
尝试次数: 8	快速读取区 4x: 0	数据长度: 0

确认(Y) | 应用(A) | 取消(N)



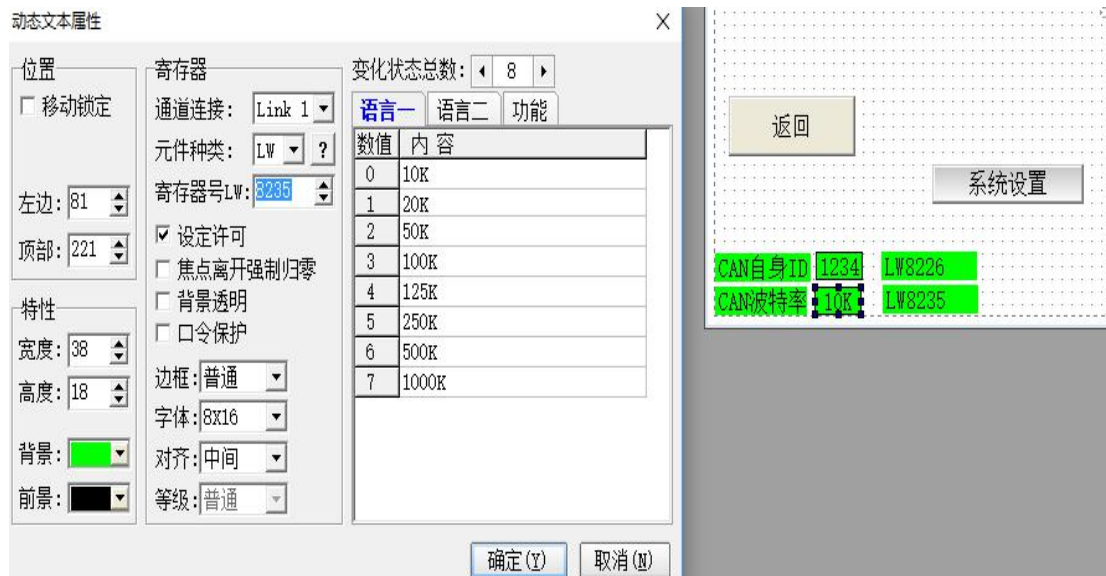


通讯波特率在工作参数设定界面设定之后，还需要在各个通讯中的屏内添加寄存器加以设置，添加一个 LW8226，这个寄存器用于更改 CAN 通讯中设备自身的 ID 号。另外添加一个多态开关 LW8226，此开关用于调节通讯波特率。

通讯时要注意：**1 LW8226 也就是自身 ID 号，ID 号必须  $\geq 100$  才能实现通讯**

**2 LW8235 通讯波特率需要统一。**

**2 LW8206 (MODBUS 通讯站号)，必须等于 255。**



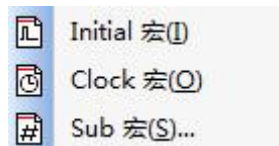
## 四、宏指令功能说明

宏是一个相当方便且强大的功能，如果撰写宏的人没有留心的话，也很容易造成大的错误，因此在这里特别强调，使用者撰写宏后，要先在计算机上做离线或是线上的模拟，并且一定要模拟一段时间之后，才开始真正的在人机端执行宏。如果你能正确的使用宏的话，不但能帮你很多的忙，也能节省你很多的时间，如果再搭配一些感应组件或是硬设备的话，甚至能帮你节省一些人力（类似自动化的过程）。每个宏最多只能编写 512 行，如果是在一行内写入注解或是字符串的话，每行最多 50 个字。而子宏最多允许 16 个，编号为 1-16（请参考下图），使用方法为呼叫“CALL 子宏编号”。



上图可以依照子巨集功能的不同，分别写上此子巨集的功能解释，如此在巨集的管理、调试以及运用上就更加方便了。而所有子巨集的初始名称为 Sub-macro #n，n 为 1 到 16。

### 4.1 宏的种类



#### 1、Initial 宏

Initial 宏初始宏，上电执行一次。在整个程序或是机器中只有一个，是程序一开始时便会执行的宏，也因此可以在这里将整个过程中必须先起始或是设定的值，先行放入，不但可以省掉设定的麻烦，也可以做类似初期设定的控制，避免因为初始值未知所造成的问题。如果在 PLC 某些位置上有特殊的设定，这里便能够设定它，如果宏设计够好的话，相信能节省你不少的工作时间。

#### 2、Clock 宏

Clock 宏，时钟宏，100ms 为一个周期，也是整个程序或机器中只有一个，会一直重复执行，而且是一次执行完毕，完毕后，于下一个 Clock 触发时则再一次重复执行。

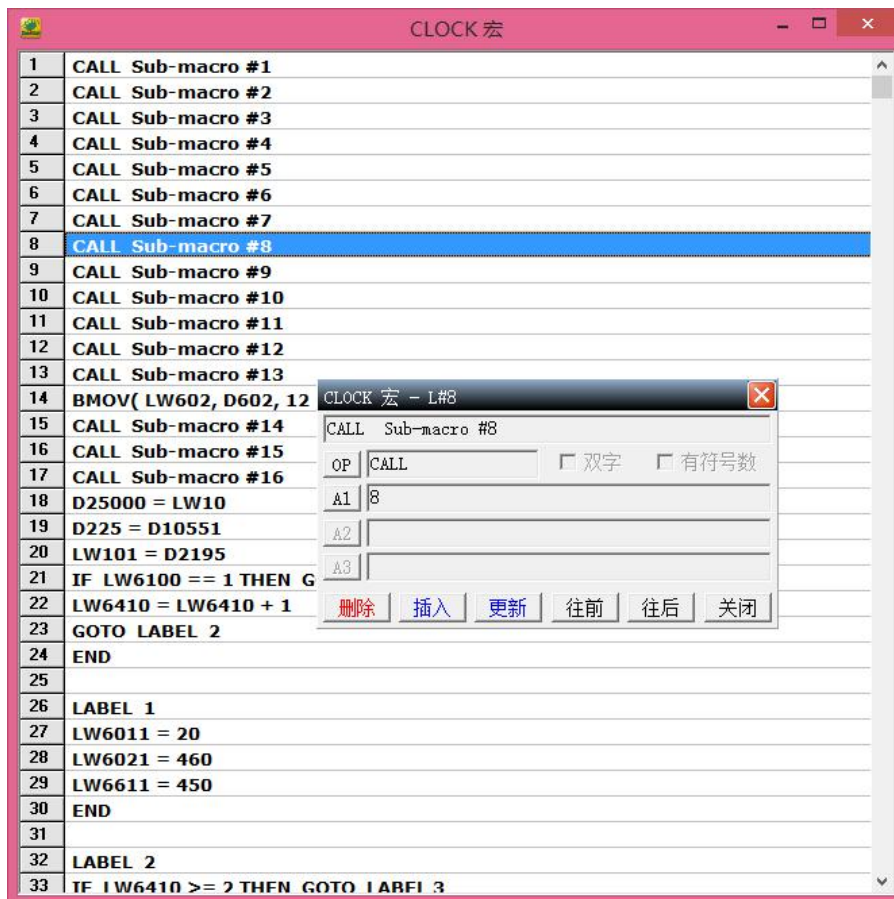
#### 3、Sub 宏

共有 16 个子宏可供编写，子宏类似程序中的子程序一样，使用者可以把重复性高的动作或是功能，放入子宏中，如此不但可以节省写宏的时间，而且调试容易。举例来说：如果有 10 个操作都使用到某一个功能，那么就可以把它写成一个子宏，比如写到子宏 1，需要

此功能的宏只要写一行“CALL 1”便可以解决了；而且如果此功能需要修改的话，只需修改子宏 1，便完成了，并不需要修改 10 个宏，而且子宏可以编写名字来表示此子宏功能，也比较方便管理。

### 4.3 宏的编辑

选好所要编写的宏后，点选后进入编辑画面，就可以开始编辑了，随便点选一行，宏指令编辑窗口便会自动浮现，并且此输入窗口，会跟着你所点选的位置而改变，以方便使用者编辑，最左边的号码为每行的编号。



当你开始编辑宏时，只要点选宏内任意一行，便会出现编辑窗口（如下图）



此时点击【OP】按钮来决定你所要使用的宏指令，点击【OP】按钮后便会弹出如下指令窗口，



这时只需要移动鼠标到所需指令的位置，按确定按钮返回。然后点击【A1】【A2】【A3】按钮即可以编辑宏，即对所选择的宏参数进行设置。

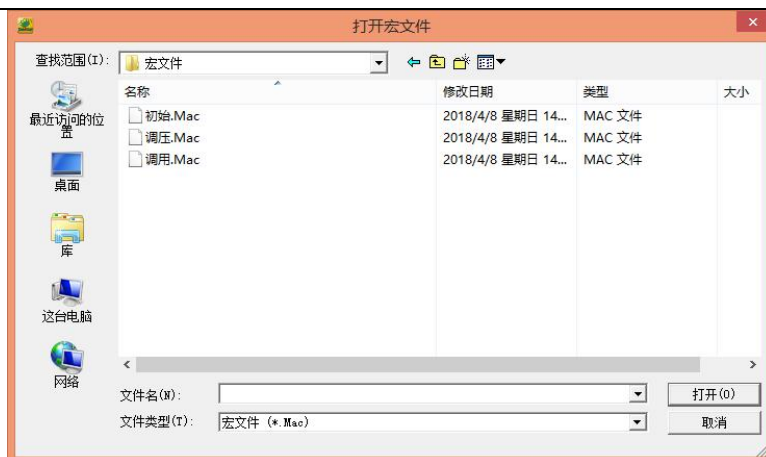


### 打开宏文件

为了方便使用者编辑宏，我们提供了开启旧文件的功能，使用者可以将所储存的宏打开，不管是那一个 PLC 厂商底下，都可以用此功能将所储存的档案开启，如此重复性高的宏文件便不需要再次的重新输入，大大的缩减编辑宏的时间，开启时在宏窗口右键选择

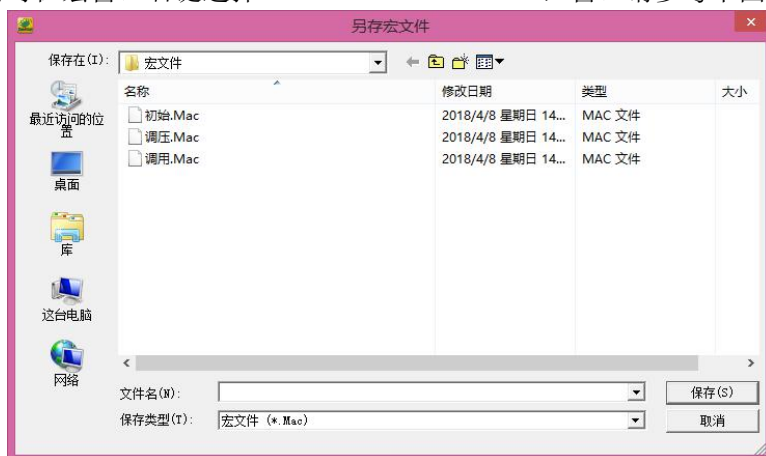
打开宏文件(O)...    Ctrl+O，窗口请参考下图：





### 另存宏文件

同样的，为了方便使用者编辑宏，我们提供了另存为新文件的功能，使用者可以将正在编辑的宏储存起来，不管是为了备份，或是为了减少其它宏的重新输入，都可以用此功能将宏储存起来。开启时在宏窗口右键选择 **另存宏文件(V)... Ctrl+S**，窗口请参考下图：



## 4.4 宏的操作数

### 算术运算

算术运算：一共有 ADD(加法)、SUB(减法)、MUL(乘法)、DIV(除法)、以及 MOD(取余数) 五种，每种都有三个操作数，每个操作数(输出只能是内部存储器)都可以是内部存储器或是常数，单位格式可以是 Word、Double Word、Signed、Signed Double Word，详情请参考以下表格。

指令	操作数	资料型态	可用格式	其它
ADD	A1、A2、A3	记、常	W、D、S	如果Word(dWord)的单位所得到的值超过Word(dWord)的长度，将仅纪录Word(dWord)范围内的数值，范围外的值一律舍弃。
SUB	A1、A2、A3	记、常	W、D、S	
MUL	A1、A2、A3	记、常	W、D、S	
DIV	A1、A2、A3	记、常	W、D、S	
MOD	A1、A2、A3	记、常	W、D、S	

记=内部存储器、常=常数、W=Word、D=Double Word、S=Signed。

► ADD→加法，公式如  $A1=A2+A3$ 。指 A2 和 A3 相加运算结果存储在 A1 中。

例如：

$A1(\text{Word})=A2(\text{Word}) + A3(\text{Word})。$

$A1(\text{Double Word})=A2(\text{Double Word}) + A3(\text{Double Word})。$

$A1(\text{Signed})=A2(\text{Signed}) + A3(\text{Signed})。$

$A1(\text{Signed Double Word})=A2(\text{Signed Double Word}) + A3(\text{Signed Double Word})。$

►SUB→减法，公式如  $A1=A2-A3$ 。指 A2 和 A3 相减运算结果存储在 A1 中。

例如：

$A1(\text{Word})=A2(\text{Word}) - A3(\text{Word})。$

$A1(\text{Double Word})=A2(\text{Double Word}) - A3(\text{Double Word})。$

$A1(\text{Signed})=A2(\text{Signed}) - A3(\text{Signed})。$

$A1(\text{Signed Double Word})=A2(\text{Signed Double Word}) - A3(\text{Signed Double Word})。$

►MUL→乘法，公式如  $A1=A2*A3$ 。指 A2 和 A3 相乘运算结果存储在 A1 中。

例如：

$A1(\text{Word})=A2(\text{Word}) * A3(\text{Word})。$

$A1(\text{Double Word})=A2(\text{Double Word}) * A3(\text{Double Word})。$

$A1(\text{Signed})=A2(\text{Signed}) * A3(\text{Signed})。$

$A1(\text{Signed Double Word})=A2(\text{Signed Double Word}) * A3(\text{Signed Double Word})。$

►DIV→除法，公式如  $A1=A2/A3$ 。指 A2 和 A3 相除运算结果存储在 A1 中。A3 不可为 0。

例如：

$A1(\text{Word})=A2(\text{Word}) / A3(\text{Word})。$

$A1(\text{Double Word})=A2(\text{Double Word}) / A3(\text{Double Word})。$

$A1(\text{Signed})=A2(\text{Signed}) / A3(\text{Signed})。$

$A1(\text{Signed Double Word})=A2(\text{Signed Double Word}) / A3(\text{Signed Double Word})。$

►MOD→取余数，公式如  $A1=A2\%A3$ 。指 A2 和 A3 相除余数存储在 A1 中。A3 不可为 0。

例如：

$A1(\text{Word})=A2(\text{Word}) \% A3(\text{Word})。$

$A1(\text{Double Word})=A2(\text{Double Word}) \% A3(\text{Double Word})。$

$A1(\text{Signed})=A2(\text{Signed}) \% A3(\text{Signed})。$

$A1(\text{Signed Double Word})=A2(\text{Signed Double Word}) \% A3(\text{Signed Double Word})。$

### 逻辑运算

逻辑运算：一共有 OR、AND、XOR、SHL 以及 SHR 五种，每种都有三个操作数，每个操作数(输出只能是内部存储器)都可以是内部存储器、或是常数，单位格式可以是 Word、Double Word，请参考以下表格。

指令	操作数	资料型态	可用格式	其它
OR	A1、A2、A3	记、常	W、D	
AND	A1、A2、A3	记、常	W、D	
XOR	A1、A2、A3	记、常	W、D	
SHL	A1、A2、A3	记、常	W、D	
SHR	A1、A2、A3	记、常	W、D	

记=内部存储器、常=常数、W=Word、D=Double Word。

►OR→逻辑或，公式如  $A1=A2|A3$ 。

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

A2 (Word) 和 A3 (Word) 相对 bit 做 OR 运算，结果存在 A1 (Word)。

A2 (DWord) 和 A3 (DWord) 相对 bit 做 OR 运算，结果存在 A1 (DWord)。

➤ AND→逻辑积，公式如  $A1=A2\&A3$ 。

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

A2 (Word) 和 A3 (Word) 相对 bit 做 AND 运算，结果存在 A1 (Word)。

A2 (DWord) 和 A3 (DWord) 相对 bit 做 AND 运算，结果存在 A1 (DWord)。

➤ XOR→逻辑互斥或，公式如  $A1=A2\^A3$ 。

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

A2 (Word) 和 A3 (Word) 相对 bit 做 XOR 运算，结果存在 A1 (Word)。

A2 (DWord) 和 A3 (DWord) 相对 bit 做 XOR 运算，结果存在 A1 (DWord)。

➤ SHL→左移，公式如  $A1=A2\ll A3$ 。

A2 (Word) 的地址左移 A3 个 bit 运算，其结果存在 A1 (Word)。左移位元指令是一边填入 0 到 bit0，一边移出 bit15。假如移位量 A3 大于 16，则 A1 为 0。

A2 (DWord) 的地址左移 A3 个 bit 运算，其结果存在 A1 (DWord)。左移位元指令是一边填入 0 到 bit0，一边移出 bit31。假如移位量 A3 大于 32，则 A1 为 0。

➤ SHR→右移，公式如  $A1=A2\gg A3$ 。

A2 (Word) 的地址右移 A3 个 bit 运算，其结果存在 A1 (Word)。右移位元指令是一边填入 0 到 bit15，一边移出 bit0。假如移位量 A3 大于 16，则 A1 为 0。

A2 (DWord) 的地址右移 A3 个 bit 运算，其结果存在 A1 (DWord)。右移位元指令是一边填入 0 到 bit31，一边移出 bit0。假如移位量 A3 大于 32，则 A1 为 0。

### 资料搬移

资料搬移：一共有 MOV、BMOV、FILL、CHR、STR、S.DATE、S.TIME 七种，详情请参考以下表格（输出只能是内部存储器）。

指令	操作数	资料型态	可用格式	其它
MOV	A1、A2	P、记、常	W、D	A1 只有 P、记
BMOV	A1、A2、A3	P、记、常	W、D	A1、A2 只有 P、记
FILL	A1、A2、A3	记、常	W、D	A1 只有记
CHR	A1、A2	记、常	W	A2 是输入字符串
STR	A1、A2、A3	记、常	W	A1 只有记
S.DATE	A1	记	W	A1 只有记
S.TIME	A1	记	W	A1 只有记

P=PLC、记=内部存储器、常=常数、W=Word、D=Double Word。

➤ MOV→搬移，如  $A1(\text{Word})=A2(\text{Word})$ ，或者  $A1(\text{DWord})=A2(\text{DWord})$

搬移指令是将缓存器 A2 的数据拷贝到目标缓存器 A1 中，而缓存器 A2 的资料不会因搬移指令而改变。搬移指令中如果 A1 使用为 PLC 地址，则表示将人机内部缓存器 A2 的资料由通讯方式写入 PLC 位地址。反之如果 A2 使用为 PLC 地址，则表示由通讯方式将 PLC 位地址 A2 的资料读出并搬移给人机内部缓存器 A1 地址。

►BMOV→区块搬移，如 BMOV(A1, A2, A3)

就是从 A2 地址开始搬移到 A1 地址，共搬移 A3 个数目，资料格式只有 Word。区块搬移指令是指从缓存器地址 A2 为起点，将连续 A3 个缓存器资料区块拷贝到目标缓存器 A1 为起始地址的连续 A3 个缓存器中，而缓存器 A2 的资料不会因区块搬移指令而改变。**A3(个数目)最大有效数值为 30；不管 A3 是直接或是间接数据，如果有效数值大于 30，则系统只按 30 处理**；区块的长度如果超过内部存储器或是 PLC 的最大值，该指令将被放弃执行。

►FILL→内存填满，如 FILL(A1, A2, A3)

就是从 A1 地址开始填满 A2 的数值，共 A3 个数目，资料格式只有 Word。填满指令是指将缓存器 A2 的资料为起点，填满到目标缓存器 A1 为起始地址的连续 A3 个缓存器中，而缓存器 A2 的资料不会因填满指令而改变。区块的长度如果超过内部存储器或是 PLC 的最大值，编译将不会通过。

►CHR→将文字转 ASCII，如 CHR(A1, "A2")

将 A2 内的文字转换为 ASCII 数字码存在 A1。字符串的长度最大为 50 个字符。

►STR→将整数转换为 ASCII 字符串，如 STR(A1, A2, A3)。多用于打印机功能。

就是把 A2 地址里面的整数，转换为 ASCII 字符串放置到 A1 指定的地址，由 A3 指定小数点位数，资料格式只有 Word(1 个 Word = 2 字符/1 个中文)

►S.DATE→取系统日期字符串，如 A1= S.DATE()

取系统当前日期转换为字符串存放到 A1 指定的寄存器地址，格式为“YYYY-MM-DD”占用 5 个 Word。建议 A1 使用可变文本元件。

►S.TIME→取系统时间字符串，如 A1= S.TIME()

取系统当前时间转换为字符串存放到 A1 指定的寄存器地址，格式为“HH:MM:SS”占用 4 个 Word。建议 A1 使用可变文本元件

### 比较

比较宏指令，一共有 IF==、IF!=、IF>、IF>=、IF<、IF<=、IF AND==0、IF AND!=0、IF == ON、IF ==OFF 等 10 种，其它详情请参考以下表格。

指令	操作数	资料型态	可用格式	其它
IF==	A1、A2、A3	记、常	W、D、S	A3 只有常
IF!=	A1、A2、A3	记、常	W、D、S	A3 只有常
IF>	A1、A2、A3	记、常	W、D、S	A3 只有常
IF>=	A1、A2、A3	记、常	W、D、S	A3 只有常
IF<	A1、A2、A3	记、常	W、D、S	A3 只有常
IF<=	A1、A2、A3	记、常	W、D、S	A3 只有常
IF AND==0	A1、A2、A3	记、常	W、D	A3 只有常
IF AND!=0	A1、A2、A3	记、常	W、D	A3 只有常
IF == ON	A1、A2	记、常	B	A1 只有 P、记，A2 只有常
IF ==OFF	A1、A2	记、常	B	A1 只有 P、记，A2 只有常

P=PLC、记=内部存储器、常=常数、W=Word、D=Double Word、S=Signed、B=Bit。

►IF== →即比较两个操作数，若相等就跳到某个标签。如 IF A1==A2 THEN GOTO LABEL A3。可使用 Signed DW 的形式。

►IF!= →即比较两个操作数，若不相等就跳到某个标签。如 IF A1!=A2 THEN GOTO LABEL A3。

可使用 Signed DW 的形式。

➤IF> →即比较两个操作数，若大于就跳到某个标签。如 IF A1>A2 THEN GOTO LABEL A3。

可使用 Signed DW 的形式。

➤IF>= →即比较两个操作数，若大于或等于就跳到某个标签。如 IF A1>=A2 THEN GOTO LABEL A3。可使用 Signed DW 的形式。

➤IF< →即比较两个操作数，若小于就跳到某个标签。如 IF A1<A2 THEN GOTO LABEL A3。

可使用 Signed DW 的形式。

➤IF<= →即比较两个操作数，若小于或等于就跳到某个标签。如 IF A1<=A2 THEN GOTO LABEL A3。可使用 Signed DW 的形式。

➤IF AND==0 →即比较两个操作数的逻辑和，等于 0 就跳到某个标签。如 IF (A1&A2)==0 THEN GOTO LABEL A3。可使用 DW 的形式。

➤IF AND!=0 →即比较两个操作数的逻辑和，不等于 0 就跳到某个标签。如 IF (A1&A2) !=0 THEN GOTO LABEL A3。可使用 DW 的形式。

➤IF == ON →即比较操作数 A1 的 BIT，若等于 1 (ON) 就跳到某个标签。如 IF A1==ON THEN GOTO LABEL A2。

➤IF == OFF →即比较操作数 A1 的 BIT，若等于 0 (OFF) 就跳到某个标签。如 IF A1==OFF THEN GOTO LABEL A2。

### 流程控制

流程控制一共有 GOTO、LABEL、CALL、RET、END 共五种，详情请参考以下表格。

指令	操作数	资料型态	可用格式	其它
GOTO	A1	常		
LABEL	A1	常		
CALL	A1	常		
RET	无			
END	无			

#### 常=常数

➤GOTO →无条件跳跃到某个标签，如 GOTO LABEL A1。

无条件跳跃指令会产生一个分支跳到该程序内部所指定的标签 (LABEL A1)，所指定的 LABEL A1 必须同在该程序内部。

➤LABEL→标签，如 LABEL A1

注意：同一宏程序内部所指定的标签代码不能重复，但不同宏程序间则可重复指定标签代码。

➤CALL →呼叫子宏 (Sub-macro) 程序，如 CALL A1。

呼叫子宏程序可将控制权转移到子宏程序中的程序指令，通常使用子宏程序来执行特定功能的控制设计或传递参数表或复杂运算集合指令等等。注意子宏程序必须存在，且其程序最后须由 RET 指令返回，由 RET 指令将控制权转移回到原呼叫子宏程序处的下一行程序指令。子宏程序编号可由 01~16 组，子宏程序名称可自定。

➤RET →回主宏程序

仅用于子宏 (Sub-macro) 程序中，而 CALL 放在主程序中。每个 RET 指令必须要有一个 CALL 指令与之对应。

➤END →结束宏程序

本 END 指令代表宏程序结束，其后面的宏程序不会被执行，下次重新从第一行指令开始。重要注意事项：END 指令代表宏程序结束。

### 位设定



指令	操作数	资料型态	可用格式	其它
SETB	A1	P、记	B	
CLRB	A1	P、记	B	
INVB	A1	P、记	B	

P=PLC、记=内部存储器、B=Bit。

- SETB → 设定特定 BIT 状态为 ON，如：SETB A1。
- CLRB → 设定特定 BIT 状态为 OFF，如：CLRB A1。
- INVB → 反置特定 BIT 状态，如：INVB A1。

#### 其它

其它有：TIMETICK、Comment、CMP、TXD、RXD、Disk. SPA、MOVLink2、MOVLink1、FP32INT、INTFP32 共 10 个指令。

指令	操作数	资料型态	可用格式	其它
TIMETICK	A1	记	W、D	
Comment	A1	字		
CMP	A1、A2、A3	记、常	W、D、S	
TXD	A1、A2	记、常	W	
RXD	A1、A2	记、常	W、D	
Disk. SPA	A1	记	W	
MOVLink2	A1、A2	P、记	W、D	
MOVLink1	A1、A2	P、记	W、D	
FP32INT	A1、A2	P、记	W、D、F	
INTFP32	A1、A2	P、记	W、D、F	

- TIMETICK → 得到系统的时间 (CPU 系统内部运算时间)，放入选定的地址里面，每增加 1，则表示增加 100ms。
- Comment → 宏注释，增加对宏的解释，对宏并无任何影响。
- CMP → 区块比较，如：CMP (A1, A2, A3)

指从 A1 地址开始的连续 A3 个数目寄存器与 A2 地址开始的连续 A3 个数目寄存器进行比较，比较结果存放在 A1+A3 地址寄存器，比较完全匹配结果==1，反之结果==0；A3 设置最大值==50。

- TXD → 发送数据 如：TXD (A1, A2)。<通讯协议必须为 Free Protocol>

指从 A1 地址开始发送数据到 LINK1/LINK2 选择对应的 UART 端口，共发送 A2 个字节，资料格式只有 Word (1 个 Word=2 字节)

- RXD → 接收数据 如：RXD (A1, A2)。<通讯协议必须为 Free Protocol>

指从 LINK1/LINK2 选择对应的 UART 端口接收数据到 A1 指定的开始地址，A2 指读取字节数。如果 A2 指定 K，则指定读取 K 个字节；如果 A2 指定 LW，则读取缓冲区 (256 个字节的大小) 的全部字节，故 A2 的地址取值必须  $\geq A1+255$ 。A2+1 地址中存放资料格式，设置 A2+1=0：表示收到的数据按字节排列；设置 A2+1=1：表示收到的数据按字排列，高字节在前；设置 A2+1=2：表示收到的数据按字排列。低字节在前；

- Disk. SPA → 取磁盘剩余空间 如：A1=Disk. SPA ()

取内部磁盘剩余可用空间值存放到 A1 指定的寄存器地址，单位为 0.1MB 点用 1 个 Word。

- MOVLink2 → 传递 Link2 的数据 如：A1 = A2 [Link2]

将与屏通讯的 Link2 中的数据，传给屏内部寄存器或传递给与屏通讯的 Link1 中的寄存器。

如下图所示，该智能控制面板通过 Link2 与 PLC 通讯，当满足屏内部寄存器 LW200 = 111 此条件时，程序会将 Link2 PLC 中寄存器 D10 的数据，传递给屏内部的寄存器 LW100。



➤ MOVLink1 → 传递 Link1 的数据 如: A1[Link2]=A2

将屏中数据或与屏通讯的 Link1 中的数据, 传递给与屏通讯的 Link2 中设定的寄存器中。

如下图所示, 该智能控制面板接了两个 PLC, 以 Link1 Link2 区分, 当满足屏内部寄存器 LW200 = 11 此条件时, 程序会将 Link1 PLC 中寄存器 D10 的数据, 传递给 Link2 PLC 中的寄存器 D0。



➤ FP32INT → 浮点数转换为整数 如: FP32INT(A1, A2)

将 A2 的浮点数转换为整数放入 A1 中。A1 必须设置为整数类型。

➤ INTFP32 → 整数转换为浮点数 如: INTFP32(A1, A2)

将 A2 的浮点数转换为整数放入 A1 中。A1 必须设置为浮点数类型。

## 4.5 错误讯息

1) 无此 LABEL

此讯息表示找不到 GOTO 所要到的 LABEL1。如下图:



2) LABEL 重复

此讯息表示同一个宏程序内部所指定的标签代码重复。如下图：

