

一. 基础知识

巨控模块配置软件 GRMDEV，内置事件控制和定时器编程，可以完成各种定时，条件，顺序控制，数据处理和转发，实际上巨控模块可以做为一个 PLC 或者数据协议网关使用。

该说明书适合巨控 GRM 模块的全部型号（GRM 500 ,GRM200,GRM230 GRM530 GRM110 等）。

表达式

表达式是一个 C 语言的数值表达式，语法符合标准 C 语言规范，是 C 表达式的子集。表达式可以是单个变量名，也可以是常量，还可以是变量和操作符的运算式。

本产品支持条件表达式、参数表达式和赋值表达式。

- ◆ 条件表达式：用于报警触发条件、定时执行条件和事件触发条件中，表达式计算结果非 0 表示条件成立。

例如：(温度>20&&湿度>50)||开关 1


- ◆ 参数表达式：用于短信报警和短信查询回送信息模板中，计算结果与类型有关。

- ◆ 赋值表达式：用于短信写变量、短信控制命令、定时器和事件控制中。

操作符

本系统支持以下几种常用的操作符，手册中未提到的操作符（如++，--）不被支持。


操作符	类型	示例	说明
+	算术运算符	a+b	加法
-	算术运算符	a-b	减法
*	算术运算符	a*b	乘法
/	算术运算符	a/b	除法
%	算术运算符	a%b	取模
>	比较操作符	a>3	大于
<	比较操作符	a<b	小于
>=	比较操作符	a>=b	大于等于
<=	比较操作符	a<=b	小于等于
==	比较操作符	a==b	相等
!=	比较操作符	a!=b	不相等
&&	逻辑操作符	a&&b	逻辑与
	逻辑操作符	a b	逻辑或
~	位操作符	~a	按位取反
^	位操作符	a^b	按位异或
&	位操作符	a&b	按位与
	位操作符	a b	按位或
!	单目操作符	! a	逻辑取反
-	单目操作符	-1	产生操作数负值
()	括号操作符	(a-b)*c	改变运算顺序
=	赋值操作符	a=3	赋值操作

 **注意：**操作符双等号“==”和单个等号“=”两者区别。

变量=5，是赋值表达式，执行后变量的值改变为 5。

变量==5，是比较操作，判断变量是否等于 5，若等于 5，条件为真（1）。

操作符的优先级别如下：

操作符	优先级别
—(单目), !, ~	由高到低 
*, / , %	
+, —	
<, >, <=, >=, ==, !=	
&, , ^	
&&	
=	



注意：建议使用括号来明确优先级如：

(风机 1 故障||风机 2 故障) && (温度>50)

表达式举例

- 1) 多个条件同时成立，符号&&表示 2 个条件需要同时成立。
风机 1 故障&&风机 2 故障
- 2) 任何一个条件成立,, 符号||表示 2 个条件只需要满足一个。
风机 1 故障||风机 2 故障
- 3) 组合条件判断
(风机 1 故障||风机 2 故障) && (温度>50)

以上表达式表示任何一个风机故障而且温度>50 时成立

定时器

GRM 模块支持间隔定时器、每日定时器和星期定时器三种定时器。定时器可以实现定时发送查询短信，也可以实现定时给变量赋值。

定时器编辑器: (开机后始终按照指定的间隔执行)

定时器类型

☒ 间隔定时器 ☐ 每日定时器 ☐ 星期定时器

定时间隔(可取1秒-31天)

定时时间单位

☐ 时 ☒ 分 ☐ 秒

执行条件

定时执行的内容

触发短信查询(可以发送短信到指定目标):

写变量(最多可写入8个变量):

	变量名	变量值/表达式
1	压缩机运行分钟	压缩机运行分钟+1
2	化霜输出	0
3		
4		

◆ 定时器类型

间隔定时器: 始终按指定间隔周期执行定义的动作。

如: 上图表示, 在压缩机==1 成立后, 过一分钟后, 开始第一次执行下面写变量的操作, 然后每过一分钟重复一次。

每日定时器: 每日指定的时间段内(开始时间至结束时间) 按指定间隔周期执行定义的动作。

星期定时器: 每周若干天内的指定时间点(开始时间) 执行一次定义的动作。

◆ 执行条件

是一个条件表达式, 当表达式计算结果非 0 时才执行定时器定义的动作。

◆ 定时间隔

连续执行动作之间的时间间隔。

◆ 触发短信查询

在完成赋值操作后, 发出一条已定义的查询短信, 短信发送给该查询授权的所有用户。

◆ 定时执行内容

定时器执行的动作可以是向用户发送一个查询信息, 也可以是给变量赋值。

如下图：

表示每天上午 9:00 触发一次查询，然后 10:00 触发一次查询，一直到下午 17:00 触发最后一次查询。
也就是每隔 1 小时发出一条短信，如果模块上电时间是 9:30，那么第一条短信会 10:00 发出。

定时器编辑器: (在每天指定的时间段内按照指定间隔执行)

定时器类型

☐ 间隔定时器 ☒ 每日定时器 ☐ 星期定时器

开始时间 09:00 结束时间 17:00

定时间隔(可取1秒-31天) 1

定时时间单位

☒ 时 ☐ 分 ☐ 秒

执行条件

添加变量

定时执行的内容

触发短信查询(可以发送短信到指定目标):

查询1

写变量(最多可写入8个变量):

	变量名	变量值/表达式
1		

确认 取消

事件控制

事件控制编辑器

触发表达式

添加变量

执行条件==1

描述信息:

执行条件为真时, 按触发模式执行赋值运算

触发模式

☒ 表达式为真, 触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☐ 表达式为真, 周期性触发

触发延时(单位0.1秒, 最大60000, 0是无延时)

0

触发查询:

(无) 特殊情况可以在执行事件时触发短信发送到手机

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	执行内容1	1
2	执行内容2	运算结果

右边的值或者表达式写入到左边的变量, 完成事件执行结果

确认

取消

- ◆ 触发表达式
触发表达式是一个条件表达式。计算结果非 0 表示触发条件成立。只填入数字“1”，表示为真执行。
- ◆ 触发模式
表达式为真，触发 1 次：触发条件成立时仅执行一次事件动作。
表达式变化触发：变量值变化时执行事件动作。该类型事件会在系统启动时先执行一次。
表达式为真，周期性触发：触发条件成立时先执行一次，然后按周期性执行事件动作。
注意：尽量使用表达式为真周期性触发，程序更为清晰，易于调试。可参考后面简例 1
- ◆ 触发延时
仅对“表达式为真，触发 1 次”事件有效。在设定时间内，如果触发条件一直成立，GRM 模块执行事件动作 1 次。
- ◆ 触发周期
仅对“表达式为真，周期性触发”事件有效。触发条件成立时，系统每隔设定时间执行一次事件动作。第一次事件动作是在触发条件成立时立即执行。
- ◆ 触发查询
在完成赋值操作后，发出一条已定义的查询短信，短信发送给该查询授权的所有用户。
- ◆ 事件执行内容
变量赋值操作。可以写入多行，注意是事件触发的时候，将右边的变量值或者计算表达式，复制给左边的变量。

高级事件控制

高级事件控制实质上是多个事件控制的组合，和前面的使用方法并无区别。

使用高级事件控制可以减少模块程序里面事件控制的数量，增强可读性：

如下二个单独的事件控制：

要求错缺相或者压机过载任何一个条件成立时，关闭风机，压机，输出故障。

没有任何报警输入时，取消报警输出

事件控制编辑器

触发表达式

错缺相输入||压机过载输入

添加变量

描述信息:

触发模式

☒ 表达式为真，触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☐ 表达式为真，周期性触发

触发延时(单位0.1秒, 最大60000, 0是无延时)

0

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	压机输出	0
2	风机输出	0
3	故障输出	1

确认

取消

事件控制编辑器

触发表达式

(错缺相输入==0)&&(压机过载输入==0)

添加变量

描述信息:

触发模式

☒ 表达式为真，触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☐ 表达式为真，周期性触发

触发延时(单位0.1秒, 最大60000, 0是无延时)

0

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	故障输出	0

确认

取消

可以使用一个高级事件控制完成：

高级事件控制编辑器

触发表达式

错缺相输入||压机过载输入

添加变量

描述信息:

表达式为真执行一次

表达式为真周期性执行

表达式为假执行一次

触发延时(单位0.1秒, 最大60000, 0是无延时)

0

触发查询:

(无)

为真时执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	压机输出	0
2	风机输出	0
3	故障输出	1

确认

取消

高级事件控制编辑器

触发表达式

错缺相输入||压机过载输入

添加变量

描述信息:

表达式为真执行一次

表达式为真周期性执行

表达式为假执行一次

触发延时(单位0.1秒, 最大60000, 0是无延时)

0

触发查询:

(无)

为假时执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	故障输出	0

确认

取消

事件控制管理

对于比较复杂的逻辑，一般需要多个事件控制完成，如果全部采用单个事件控制，程序查找和编辑比较麻烦，可以建多个事件控制组合成一个程序块，程序块可以单独展开和编辑，浏览的时候也可以折叠或者展开程序块。

组合成程序块只是为了浏览和编辑方便，并不影响事件控制的执行顺序和逻辑，和多个独立的事情控制并无区别。

使用程序块时只需点击新建程序块，然后在这个程序块里面添加多个事件控制，即可。

需要使用定时器或者事件控制实现比较复杂的逻辑功能，可以参考巨控“巨控模块逻辑编程指南”，上面有各种案例和详细说明。

数据词典	短信报警	短信查询	来电动作	短信控制	定时器	事件控制	数据交换	从机地址映射
ID	触发表达式		触发类型	延时/周期	内容			
1	- 液位换算和修正			编辑				
1.01	1	为真周期性触发	10	液位=液位采样值 +液位修正值				
2	- 水泵启动逻辑			编辑				
2.01	液位>1.5	为真触发1次	0	水泵1启动=1				
2.02	液位>2.5	为真触发1次	0	水泵2启动=1				
3	- 水泵停机逻辑			编辑				
3.01	液位 <1	为真触发1次	0	水泵1启动=0				
3.02	液位 <0.5	为真触发1次	0	水泵2启动=0				
4	- 水泵故障处理			编辑				
4.01	水泵1故障标志 ==1	为真触发1次	0	水泵1启动=0				
4.02	水泵2故障标志 ==1	为真触发1次	0	水泵2启动=0				

二. 简单例子

简例 1：根据输入开关，启动电机

要求开关 1 闭合，电机输出，开关断开，电机断开

The image shows two side-by-side screenshots of the 'Event Control Editor' (事件控制编辑器) interface. Both windows have a title bar with a close button (X).

Left Window:

- 触发表达式 (Trigger Expression):** 开关==1 (Switch == 1). The expression is circled in red.
- 触发模式 (Trigger Mode):** ☒ 表达式为真, 触发1次(带延时) (Expression is true, trigger 1 time with delay). This option is circled in red.
- 触发延时 (Trigger Delay):** 0 (unit 0.1 seconds, max 60000, 0 is no delay).
- 触发查询 (Trigger Query):** (无) (None).
- 事件执行内容 (Event Execution Content):**
 - ☐ 自由模式(可多行, 每一项是 变量名=变量值:)
 - | 变量名 | 变量值/表达式 |
|------|---------|
| 1 电机 | 1 |
- Buttons:** 确认 (Confirm) and 取消 (Cancel).

Right Window:

- 触发表达式 (Trigger Expression):** 开关==0 (Switch == 0). The expression is circled in red.
- 触发模式 (Trigger Mode):** ☒ 表达式为真, 触发1次(带延时) (Expression is true, trigger 1 time with delay). This option is circled in red.
- 触发延时 (Trigger Delay):** 0 (unit 0.1 seconds, max 60000, 0 is no delay).
- 触发查询 (Trigger Query):** (无) (None).
- 事件执行内容 (Event Execution Content):**
 - ☐ 自由模式(可多行, 每一项是 变量名=变量值:)
 - | 变量名 | 变量值/表达式 |
|------|---------|
| 1 电机 | 0 |
- Buttons:** 确认 (Confirm) and 取消 (Cancel).

注意: 左边的事件控制只会在开关闭合时, 执行一次, 让电机启动, 并不会在开关断开为 0 时, 关闭电机, 需要右边的事件控制实现关电机。

也就是说执行方式和 PLC 编程的置位复位指令类似, 而不是线圈和触点。

更好的方式: 尽量使用周期执行, 只需要确保你的触发表达式条件正确, 就类似梯形图一样, 始终按周期扫描执行, 不依赖其他前后关系, 调试更为简单。特别适合习惯 PLC 编程的程序员。

The image shows two side-by-side screenshots of the 'Event Control Editor' (事件控制编辑器) interface, similar to the previous ones but with different trigger modes.

Left Window:

- 触发表达式 (Trigger Expression):** 开关 == 1 (Switch == 1).
- 触发模式 (Trigger Mode):** ☒ 表达式为真, 周期性触发 (Expression is true, periodic trigger). This option is circled in red.
- 触发周期 (Trigger Period):** 10 (unit 0.1 seconds, range 1-60000). The value 10 is circled in red.
- 触发查询 (Trigger Query):** (无) (None).
- 事件执行内容 (Event Execution Content):**
 - ☐ 自由模式(可多行, 每一项是 变量名=变量值:)
 - | 变量名 | 变量值/表达式 |
|------|---------|
| 1 电机 | 1 |
- Buttons:** 确认 (Confirm) and 取消 (Cancel).

Right Window:

- 触发表达式 (Trigger Expression):** 开关 == 0 (Switch == 0).
- 触发模式 (Trigger Mode):** ☒ 表达式为真, 周期性触发 (Expression is true, periodic trigger). This option is circled in red.
- 触发周期 (Trigger Period):** 10 (unit 0.1 seconds, range 1-60000). The value 10 is circled in red.
- 触发查询 (Trigger Query):** (无) (None).
- 事件执行内容 (Event Execution Content):**
 - ☐ 自由模式(可多行, 每一项是 变量名=变量值:)
 - | 变量名 | 变量值/表达式 |
|------|---------|
| 1 电机 | 0 |
- Buttons:** 确认 (Confirm) and 取消 (Cancel).

简例 2：温度显示修正

PLC 里面获取的测量温度和实际有偏差，要求 APP 上显示的温度加以修正，可以在模块数据词典里面把测量温度的网络读写属性取消，APP 只显示修正后的值，不显示原始值。

事件控制编辑器

触发表达式: 1 **1表示条件永远成立，每秒钟修正计算一次**

添加变量

触发模式

☐ 表达式为真，触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☒ 表达式为真，周期性触发

触发周期(单位0.1秒, 范围1-60000) 选择变量 10

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	显示温度	测量温度+温度修正

确认 取消

简例 3：根据温度启动停止风机

启动风机逻辑如下：

要求温度>设定温度，启动风机。用事件控制实现如下图：

如前所述，如下配置会在温度>温度设定时，打开风机，**温度<=温度设定时不会关闭风机。**

事件控制编辑器

触发表达式: 温度>温度设定

添加变量

描述信息:

触发模式

☒ 表达式为真，触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☐ 表达式为真，周期性触发

触发延时(单位0.1秒, 最大60000, 0是无延时) 0

触发查询: (无)

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	风机启动	1

确认 取消

如果希望**设定温度**作为参数，掉电后模块依然保存上次设定值，数据词典里面勾选上保存值即可。

变量编辑

×

变量属性

扩展Web属性

变量名称

设定温度

变量描述

变量类型

整数

变量组

(无)

IO设备

(无)

初始值

25

☒ 保存值

勾选后模块断电后该参数仍然可以保持

关闭风机的逻辑如下：

温度<设定温度-5 度 30 秒，关闭风机。用事件控制实现如下图：

注意，需要在 30 秒内持续满足条件温度<设定温度-5 度才会执行，

事件控制编辑器

×

触发表达式

温度<温度设定-5

添加变量

描述信息：

触发模式

☒ 表达式为真，触发1次(带延时)

☐ 表达式变化触发(只限单个变量, 且不可延时)

☐ 表达式为真，周期性触发

触发延时(单位0.1秒, 最大60000, 0是无延时)

300

触发查询：

(无)

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

变量名	变量值/表达式
风机启动	0

确认

取消

计算风机运行时间：

要求在风机运行时，计算风机运行时间，单位是秒。

用事件控制实现如下图，风机运行时，每秒钟执行一次运行时间的计数。

事件控制编辑器

触发表达式: 风机启动==1

添加变量

描述信息:

触发模式:

- ☐ 表达式为真, 触发1次(带延时)
- ☐ 表达式变化触发(只限单个变量, 且不可延时)
- ☒ 表达式为真, 周期性触发

触发周期(单位0.1秒, 最大60000, 不可为0): 10

触发查询: (无)

事件执行内容

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	风机运行时间	风机运行时间 +1

确认 取消

简例 4: 每周定时开关机

周 1 到周 5 每天早上 8:30 打开空调, 每天下午 17:30 关闭空调。

定时器编辑器: (在每天指定的时间段内按照指定间隔执行)

定时器类型:

- ☐ 间隔定时器
- ☒ 每日定时器
- ☐ 星期定时器

开始时间: 08:30 结束时间: 17:30

定时间隔(可取1秒-31天): 1

定时时间单位:

- ☐ 时
- ☐ 分
- ☒ 秒

执行条件: (\$WeekDay>=1)&&(\$WeekDay<=5)

添加变量

写变量:

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	<u>空调启动</u>	1

确认 取消

定时器编辑器: (在每天指定的时间段内按照指定间隔执行)

定时器类型:

- ☐ 间隔定时器
- ☒ 每日定时器
- ☐ 星期定时器

开始时间: 17:31 结束时间: 08:29

定时间隔(可取1秒-31天): 1

定时时间单位:

- ☐ 时
- ☐ 分
- ☒ 秒

执行条件: (\$WeekDay>=1)&&(\$WeekDay<=5)

添加变量

写变量:

☐ 自由模式(可多行, 每一项是 变量名=变量值:)

	变量名	变量值/表达式
1	空调启动	0

确认 取消

注意: 周一到周五的开机和关机逻辑是由上图 2 个独立的定时器分别执行的。

周末整天都需要关闭空调，配置方法如下：

定时器编辑器: (在每天指定的时间段内按照指定间隔执行)

定时器类型

☐ 间隔定时器

☒ 每日定时器

☐ 星期定时器

开始时间

00:00

结束时间

23:59

定时间隔(可取1秒-31天)

1

定时时间单位

☐ 时

☐ 分

☒ 秒

执行条件

(\$WeekDay ==6)&&(\$WeekDay==7)

添加变量

写变量:

☐ 自由模式(可多行, 每一项是 变量名=变量值;)

	变量名	变量值/表达式
1	空调启动	0

确认

取消

注意：因为模块通电运行时刻是不一定的，所以使用上图的每日定时器，在时间段内**每秒钟**检查一次当前时间是需要开空调还是关空调，可以确保正确的逻辑。

如果使用下图的星期定时，因为是每天仅在 8：30 检查一次是否需要开启空调，如果模块是 8：30 之后通电运行，当天的空调启动不会执行。

定时器编辑器: (每周的若干天内各按照指定时间执行一次)

定时器类型

☐ 间隔定时器

☐ 每日定时器

☒ 星期定时器

开始时间

08:30

每周的星期几执行

☐ 日 ☒ 一 ☒ 二 ☒ 三 ☒ 四 ☒ 五 ☐ 六

全选

全不选

反转选择

执行条件

添加变量

写变量:

☐ 自由模式(可多行, 每一项是 变量名=变量值;)

	变量名	变量值/表达式
1	空调启动	1

编辑...

确认

取消

三. 完整控制逻辑案例

案例 1：冷库控制逻辑

运行逻辑：

- 1. 当冷库温度 \geq 设定温度+温度偏差且持续 200 秒时，压机开启，制冷。
- 2. 当压机开启后，经过风机延迟时间，开启风机。
- 3. 当冷库温度 \leq 设定温度，压机将关闭。风机关闭。

停机逻辑：

按下停机，关闭风机和压机。

故障逻辑：

当压机故障输入或者错缺相闭合时，关闭风机，压机。报警输出闭合。

案例使用巨控 GRM232Q-4D4N4Q,输入输出全部使用模块本机 IO 资源，将模块当 PLC 使用。

工程选项

基本选项

短信/GPRS/4G

网络选项

本机IO选项

NTC分度表

Q1(继电器输出)

绑定变量

压机输出

Q2(继电器输出)

绑定变量

故障输出

Q3(继电器输出)

绑定变量

风机输出

Q4(继电器输出)

绑定变量

A1(开关量输入), 公共端 (GND)

绑定变量(工程值)

压机过载输入

A2(开关量输入), 公共端 (GND)

绑定变量(工程值)

错缺相输入

滤波次数(10mS/次)

64

滤波次数(10mS/次)

64

A3(开关量输入), 公共端 (GND)

绑定变量(工程值)

系统启动

A4(开关量输入), 公共端 (GND)

绑定变量(工程值)

滤波次数(10mS/次)

64

滤波次数(10mS/次)

64

B1(NTC测温), 公共端 (GND)

绑定变量(工程值)

冷库温度

B2(NTC测温), 公共端 (GND)

绑定变量(工程值)

滤波次数(10mS/次)

64

滤波次数(10mS/次)

64

NTC分度表

☒ 分度表1

☐ 分度表2

NTC分度表

☒ 分度表1

☐ 分度表2

B3(NTC测温), 公共端 (GND)

绑定变量(工程值)

B4(NTC测温), 公共端 (GND)

绑定变量(工程值)

滤波次数(10mS/次)

64

滤波次数(10mS/次)

64

NTC分度表

☒ 分度表1

☐ 分度表2

NTC分度表

☒ 分度表1

☐ 分度表2

确认

取消

具体实现请参考 Sample 目录下：冷库控制逻辑.grmprj

全部使用事件控制和高级事件控制实现。

Sample 目录下还提供了一个稍微复杂的冷库控制案例供参考：复杂案例.grmprj

案例 2：液位控制水泵

自动运行逻辑：

1. 当液位 \leq 设定液位底限,启动水泵，有 2 个水泵，要求轮换启动
2. 当液位 \geq 设定液位高限,停止水泵

手动运行逻辑：

可远程在 APP 上手工启停任何一个水泵，不受液位控制。

故障逻辑：

当电源输入错缺相闭合时，关闭水泵。

具体实现请参考 Sample 目录下：液位控制水泵.grmproj

全部使用事件控制和高级事件控制实现。