

BALLUFF

C-Macro Builder™
技术手册，用户指南



中文



C-Macro Builder™

适用于 **BIS M-6xx** 系列 **RFID** 控制器

www.balluff.com

1	基本信息	5
1.1	印刷规则	5
1.2	符号	6
1.3	缩写	6
2	概述	7
2.1	C-Macro Builder GUI	7
2.2	保存宏	9
3	宏命令	10
3.1	宏配置命令	10
4	单标签命令	18
4.1	单标签搜索	18
4.2	单标签 ID 读取	19
4.3	单标签数据读取	19
4.4	单标签起始连续读取数据	20
4.5	单标签停止连续读取数据	21
4.6	单标签数据写入	22
4.7	单标签数据填充	23
4.8	单标签写入 时间戳	25
4.9	单标签增量/减量数据	26
4.10	单标签 AFI 写入	27
4.11	单标签 AFI 锁定	28
4.12	单标签锁定存储数据块	28
4.13	单标签 EAS 操作	29
5	多标签命令	31
5.1	多标签遍询	31
5.2	多标签搜索	32
5.3	多标签数据读取	33
5.4	多标签数据写入	35
5.5	多标签数据填充	36
5.6	多标签 AFI 写入	38
5.7	标签 AFI 锁定	39

5.8	多标签 EAS 操作	41
6	设置/清除输出命令	43
6.1	设置输出	43
6.2	清除输出	44
7	宏专用命令	46
7.1	标记	46
7.2	分支	47
7.3	条件分支	49
7.4	传输最后响应	54
7.5	传输用户定义字符串	55
7.6	执行宏	56
7.7	等待	58
7.8	注意	58
7.9	等待	59
7.10	结尾	59
附录		61
ASCII 表		61

1 基本信息

Balluff GmbH 保留对其产品和/或文档进行修改和改进的权利，恕不另行通知。

Balluff GmbH 不对本文所含的技术或编辑错误或遗漏负责，亦不对因使用本手册所致的附随或后续损失负责。

只有在满足以下所有条件的情况下才能使用、印刷和分发本出版物的文字和图形内容：

- 首先从 Balluff GmbH 获得许可。
- 内容仅用于非商业目的。
- 内容未经修改。

以下是 Balluff GmbH 的商标和/或注册商标：Subnet16™、RFID AT WORK™、C-Macro™、C-Macro Builder™、CBx™和 Balluff Dashboard™。

本文档中提及的第三方产品名称仅用于识别目的，可能是其各自公司的商标和/或注册商标：飞利浦、罗克韦尔自动化 (ControlLogix、RSLogix)、德州仪器、英飞凌、Modbus®TCP/IP、莫迪康、百通、微软和开放式设备网络供货商协会 (ODVA)。

1.1 印刷规则

本手册使用了以下惯例：

列举

使用英文连接号进行列举。

- 列举 1。
- 列举 2。

语法

数字：

- 十进制数字显示没有附加指示符（如：123）。
- 十六进制数字与附加指示符 0x 一起显示（如：0x7B）。

参数：

参数或命令以斜体显示（例如 *CRC_16*）。

1 基本信息

1.2 符号



注意

有关可用软件和附件的更多信息，请访问 www.balluff.com。

1.3 缩写

AFI	应用族标识符
BIS	巴鲁夫识别系统
EAS	电子物品监控系统
GUI	图形用户界面
LSB	最低有效字节
MSB	最高有效字节
PC	个人电脑
RFID	射频识别
标签	数据载体
ID	标识符

2 概述

C-Macro Builder 是一种软件工具，允许快速创建和编译宏或简单的指令程序，可以加载到巴鲁夫 BIS M-6xx 和 BIS U-6xx 系列 RFID 控制器中。

在执行时，这些宏可以指示 RFID 控制器执行各种操作序列，例如向 RFID 标签写入和读取数据，比较读取数据的结果，填充或清除标签，根据数据的逻辑比较结果传输自定义字符串信息，或设置/清除输出，以及一些其他功能。

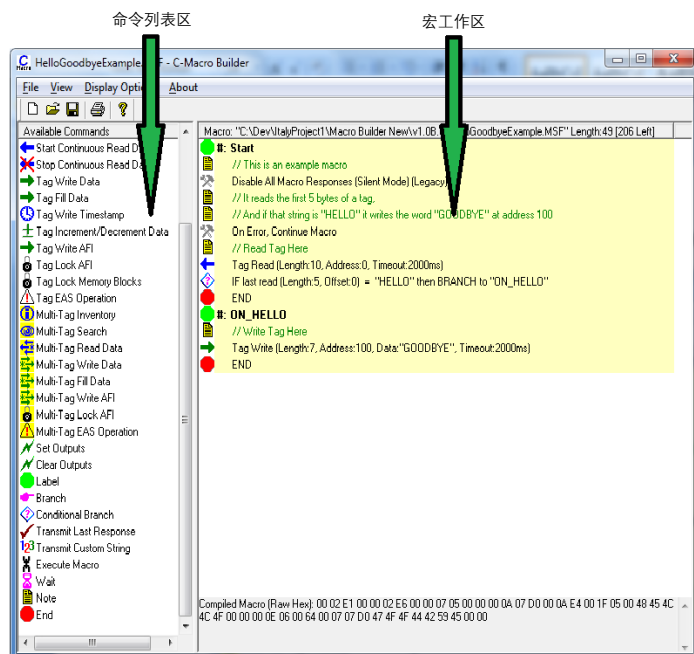
C-Macro Builder 工具可让用户使用直观的拖放界面创建这些宏，该界面以简单的宏编程语言将宏显示为人可读的指令。

宏创建完成后，C-Macro Builder 工具可以将宏的编译版本（BIS M RFID 控制器可以解析和执行的一串字节值）保存到用户 PC 上的文件中。然后，可通过其他程序（如巴鲁夫 Dashboard 配置工具）将编译后的宏加载到 BIS M-6xx 或 BIS U-6xx 系列 RFID 控制器中。

每个 BIS M-6xx 和 BIS U-6xx 系列 RFID 控制器都有空间容纳八个宏，每个宏最多 255 个字节，编号为 1 到 8。由于宏还具有执行其他宏的能力，因此可以将它们链接在一起以创建更大的程序。

2.1 C-Macro Builder GUI

C-Macro Builder GUI 是一个窗口，分为两个主要部分，命令列表和宏工作区。



C-Macro Build GUI

2 概述

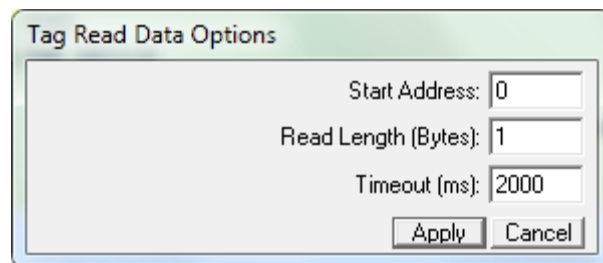
C-Macro Builder GUI

窗口的左侧部分是命令列表区域，列出了用于构建宏的可用命令。后文将单独说明这些命令。

窗口的右侧部分是宏工作区，即创建宏的区域。

要创建宏，将命令列表区中的项目（通过鼠标）拖放到宏工作区中相应的位置，一次拖一个项目。

将命令放入工作区后，将出现一个弹出对话框，可让用户输入该命令的特定选项。例如，如果用户将**标签数据读取**命令放入工作区，将出现以下对话框：



然后，用户在对话框的字段中输入相应的值 – 在本例中，是标签数据读取命令的起始地址、读取长度和超时值。输入相应的值后，用户单击**应用**，将命令插入宏。如果用户单击**取消**，则不会插入命令。

插入命令后，可双击项目来编辑参数。某些项目没有可编辑的参数，因此没有关联的弹出对话框。

也可以通过在宏中拖动鼠标重新定位命令。先用鼠标选择命令并按删除键，可以删除命令。

程序由 *RFID 控制器* 最终执行时，从顶部开始进行宏执行，并按顺序执行每个命令，将命令的响应发送回主机。后文将讨论可以将执行重新引导到宏的其他部分的各种分支命令。

创建宏时，在工作区上方的标题栏中显示宏编译字符串的总长度。此状态信息可让用户查看此特定宏剩余的物理空间，最大为 255 个字节。

2 概述

2.2 保存宏

创建宏时，C-Macro Builder 将宏编译成其字节值字符串。可以在工作区底部窗口的小灰色栏中看到此字节字符串。可以从“显示选项”菜单中启用或禁用此显示区域。它主要用于诊断或高级用户，大多数用户不会关心创建的实际字节字符串。

要保存宏的源文件（C-macro Builder 用于创建已编译宏的文件），用户可单击“文件”菜单中的**将宏另存为**。这将提示用户保存两个文件，首先是文件扩展名为 **.MSF** 的宏源文件，然后是文件扩展名为 ***.MACRO** 的已编译宏文件。

使用另一个程序将宏加载到 BIS M-6xx 或 BIS U-6xx *RFID 控制器*中。C-Macro Builder 工具本身只创建宏。请参阅其他程序的文档，例如巴鲁夫 *Dashboard 配置工具*，它可以将编译后的宏加载到 BIS M-6xx 和 BIS U-6xx 系列 *RFID 控制器*。

3 宏命令

对于本文档，命令将分为五种类型：

宏配置命令

用于配置宏行为、处理和自定义响应的命令。

标签读取、写入、锁定和 EAS 命令（单标签）

当预期射频场中只有一个标签时，用于从标签读取/写入数据、锁定标签某些部分或对标签执行 EAS 操作的命令。

多标签命令

用于在超时期间预期射频场中有多个标签的情况下使用的命令。

设置/清除输出命令

用于设置或清除 RFID 控制器（或 Hub）输出的命令

宏专用命令

用于完成与宏执行相关的各种特殊功能的命令，例如对结果进行条件检查、将宏执行控制转移到宏的其他部分（甚至其他宏）、标记宏的执行点、传输自定义消息、插入延迟、插入描述性注释或结束宏执行。

3.1 宏配置命令

以下命令用于配置宏的行为：

- 宏响应选项
- 宏响应选项（旧版）
- 错误行为选项
- 用户定义报头/报尾

宏响应选项

此命令可让宏配置宏的响应选项，并确定启用、禁用或自定义的响应类型。

BIS M-6xx 或 BIS U-6xx RFID 控制器有五种响应类型：

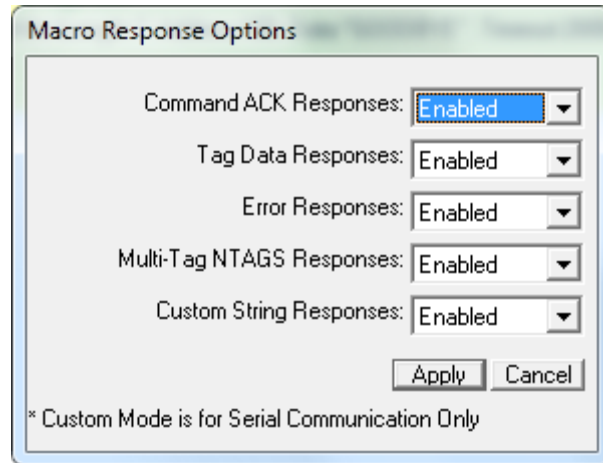
响应类型	响应的描述
命令成功响应 (ACK)	对命令代码作出回显以表明命令已成功响应
标签数据响应	包含数据的响应
错误响应	包含错误代码的响应
多标签 NTAG 响应	在多标签命令结束后发送的一种响应，该命令只包括对其执行操作的标签数。
用户定义字符串响应	发送用户定义字符串的响应

3 宏命令

宏响应选项

默认情况下，当宏开始时，*RFID 控制器* 在宏执行期间生成的所有五种类型的响应都将发送回主机。此命令可让宏更改每种类型响应的处理方式。

将项目插入宏时，会出现以下对话框：



五种响应类型的每一种都有三个选项供选择：

选项	说明
已启用	已启用此类响应并将发送到主机
已禁用	已禁用（“静默”）此类响应并且不会发送到主机
用户定义	这种类型的响应将使用 <i>用户定义报头/报尾</i> 宏命令定义的 <i>用户定义报头/报尾</i> 字节

示例 1：

在以下的示例中，此命令用于隐藏未发现标签错误，如果未发现标签，反而让 *Hub* 输出 *A* 闪烁。如果成功读取 *标签 ID*，则会使用 *传输最后响应* 命令将其发送到主机。

3 宏命令

宏响应选项

```

● #: Start
📄 // This Macro shows how certain errors can be "hidden"
📄 // Only successful Tag Serial #'s will be reported, and
📄 // Any "Tag Not Found" errors will be silenced
🔧 Set Macro Response Options: ACKS:Yes, DATA:Yes, ERRS:No, NTAGS:Yes, STRINGS:Yes
🔧 On Error, Continue Macro
👉 Tag Read ID (Timeout:2000ms)
🔍 IF last RF command caused an error then BRANCH to "ON_ERROR"
✅ Transmit Last Response
● END
● #: ON_ERROR
📄 // Toggle Hub Output 1
⚡ Set Hub Outputs {A }
⚡ Clear Hub Outputs {A }
● END

```

示例 2:

在以下示例中，此命令与*用户定义报头/报尾*命令结合使用以更改读取*标签 ID*响应的报头和报尾字符

宏无限循环，如果成功读取*标签 ID*，则将其发送到主机，并带有用户定义报头“<”和用户定义报尾“>” – 不会向主机发送其他响应（即*未发现标签*错误）

```

● #: Start
📄 // Tell macro to continue even on errors
🔧 On Error, Continue Macro
📄 // Disable all responses except Data Responses
🔧 Set Macro Response Options: ACKS:No, DATA:Custom, ERRS:No, NTAGS:No, STRINGS:No
📄 // Set the Custom Header/Terminator for Data Responses
🔧 Set Custom DATA Output Header="<", Terminator=">"
● #: Loop1
📄 // Read the Tag ID, with a timeout of 1 second
👉 Tag Read ID (Timeout:1000ms)
📄 // Loop Forever, until stopped
👉 Branch To "Loop1"

```



注意

宏响应选项命令可用于将这些设置“飞速”更改为不同设置，无论在宏中的任何位置。



注意

只有使用 ABx 协议的具有串行接口（RS232、USB 等）的 RFID 控制器才能使用此命令将响应设置为*用户定义*。无法自定义或更改 CBx 响应的报头和报尾。

3 宏命令

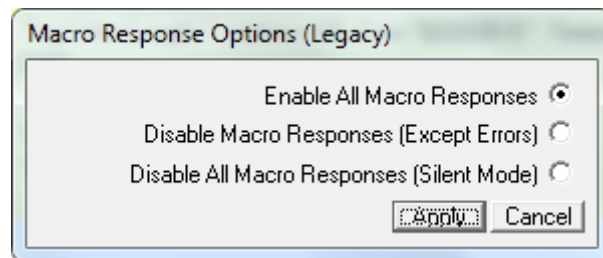
宏响应选项
(旧版)

选项	值
旧版响应模式	启用所有宏响应 禁用宏响应 (错误除外) 禁用所有宏响应 (静默模式)

这是一个旧版命令，可让宏更改模式，或者使所有五种类型的响应静默，或者重新启用它们。此命令的新版本能更好地控制这些响应，但是为旧版宏保留了此命令。

默认情况下，当宏开始时，执行宏期间 *RFID 控制器* 生成的所有响应 (成功响应或错误响应) 都会发送回主机。

将项目插入宏时，会出现以下对话框：



用户可以选择三种旧版响应模式中的一种。

选项	说明
启用所有宏响应	在选择此选项的情况下遇到此命令时，宏进入所有响应 (包括错误) 都发送到主机的模式。这是默认设置。
禁用宏响应 (错误除外)	在选择此选项的情况下遇到此命令，宏进入不将成功响应发送回主机但发送回错误响应的模式。
禁用所有宏响应 (静默模式)	在选择此选项的情况下遇到此命令时，宏将进入静默模式 (成功响应和错误响应都不会发送回主机)。

3 宏命令

宏响应选项
(旧版)

示例 1:

在以下示例中，此命令在选择禁用所有宏响应（静默模式）的情况下用于隐藏未发现标签错误，如果未发现标签，反而让 Hub 输出 A 闪烁。如果成功读取标签 ID，则会使用传输最后响应命令将其发送到主机。

```

● #: Start
  // This Macro shows how certain errors can be "hidden"
  // Only successful Tag Serial #'s will be reported, and
  // Any "Tag Not Found" errors will be silenced
  Disable All Macro Responses (Silent Mode) (Legacy)
  On Error, Continue Macro
  Tag Read ID (Timeout:2000ms)
  IF last RF command caused an error then BRANCH to "ON_ERROR"
  Transmit Last Response
  END
● #: ON_ERROR
  // Toggle Hub Output 1
  Set Outputs {Hub Output A=HIGH}
  Clear Outputs {Hub Output A=LOW}
  END

```

示例 2:

在以下的示例中，此命令使用两次 – 首先在读取标签的前 5 个字节时使宏静默，然后重新启用宏响应，读取并报告标签序列号，但前提是标签的前 5 个字节的字符串小于字符串“44444”

```

● #: Start
  // This Macro shows how certain errors can be "hidden"
  // Only successful Tag Serial #'s will be reported, and
  // Any "Tag Not Found" errors will be silenced
  Disable All Macro Responses (Silent Mode) (Legacy)
  On Error, Continue Macro
  Tag Read ID (Timeout:2000ms)
  IF last RF command caused an error then BRANCH to "ON_ERROR"
  Transmit Last Response
  END
● #: ON_ERROR
  // Toggle Hub Output 1
  Set Outputs {Hub Output A=HIGH}
  Clear Outputs {Hub Output A=LOW}
  END

```

此命令可以插入宏中的任意位置，当遇到此命令时，从该点开始将宏置于该模式。

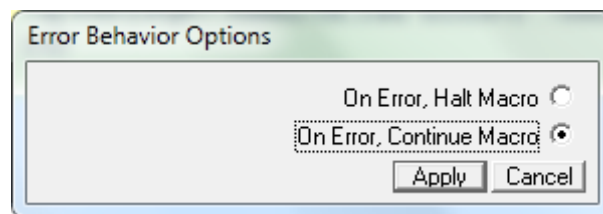
3 宏命令

错误行为选项

选项	值
错误行为模式	出错，停止宏或 出错，继续宏

此命令可在出错时指令宏继续执行还是停止。

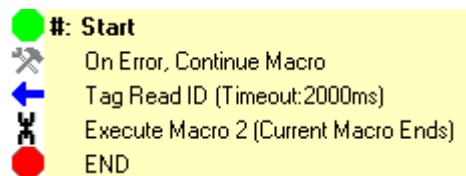
将项目插入宏时，会出现以下对话框：



默认情况下，当宏开始后，在执行宏过程中由命令生成的所有错误都会导致宏在该点停止 - 这包括未发现标签之类的错误。

示例 1:

在以下示例中，此命令用于在标签 ID 读取命令失败的情况下继续执行宏。如果没有此命令，未发现标签错误将导致宏立即结束，并且执行宏 2 命令永远不会执行。



用户定义报头/
报尾

选项	值
设置（响应类型） 用户定义报头/报尾	命令 ACK 响应 标签数据响应 错误响应 多标签 NTAG 响应 用户定义字符串响应
用户定义报头	十六进制或 ASCII 数据的 0 到 10 字节
用户定义报头长度（字节）	值 0-10
用户定义报尾	十六进制或 ASCII 数据的 0 到 10 字节
用户定义报尾长度（字节）	值 0-10
显示模式（仅限数据输入）	十六进制或 ASCII

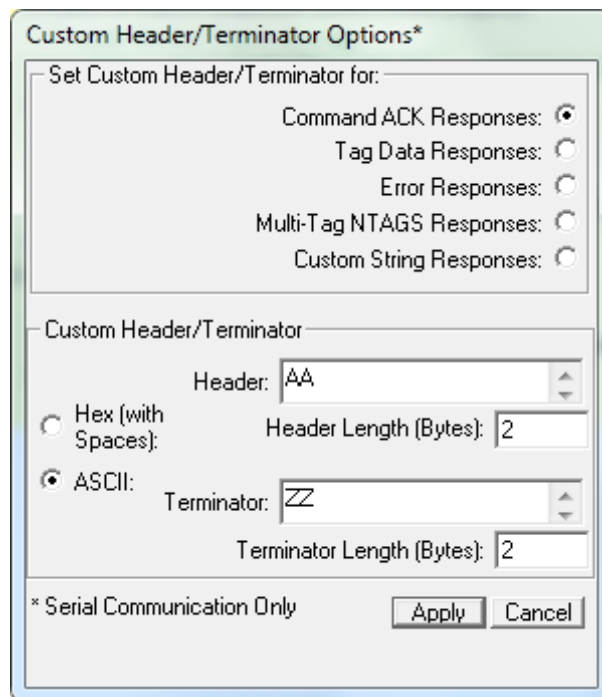
3 宏命令

用户定义报头/
报尾

默认情况下，当宏开始后，RFID 控制器在宏执行过程中生成的所有五种类型的响应（在 [3.1 宏响应选项](#) 中描述）都会用标准的 ABX 报头/报尾字符发送回主机。此命令与宏响应选项宏命令一起，可让宏更改每种响应类型的报头和报尾字符。如果特定响应类型的行为已在宏响应选项命令中设置为用户定义，则此命令用于定义用户定义报头和报尾字符。

用户定义报头/
报尾

将项目插入宏时，会出现以下对话框：



对话框的上部是设置用户定义报头/报尾部分：选择要自定义的响应类型。

对话框的下部是用户定义报头/报尾部分，可以十六进制或 ASCII 指定用户定义报头和报尾字节。用户定义报头和报尾的长度可以是 0 到 10 个字节中的任何字节。



注意

每个命令只能更改 1 个响应类型的报头/报尾字符。如果要为所有五种类型的响应更改报头/报尾，则需要 5 个命令。



注意

只有采用 ABx 协议的具有串行接口（RS232、USB 等）的 RFID 控制器才能使用此命令。无法自定义或更改 CBx 响应的报头和报尾。

3 宏命令

用户定义
报头/
报尾

示例 1:

在以下示例中，此命令与宏响应选项命令结合使用，以更改读取标签 ID 响应的报头和报尾字符

宏无限循环，如果成功读取标签 ID，则将其发送到主机，并带有用户定义报头“<”和用户定义报尾“>” – 不会向主机发送其他响应（即“未发现标签”错误）

```

● #: Start
📄 // Tell macro to continue even on errors
🔧 On Error, Continue Macro
📄 // Disable all responses except Data Responses
🔧 Set Macro Response Options: ACKS:No, DATA:Custom, ERRS:No, NTAGS:No, STRINGS:No
📄 // Set the Custom Header/Terminator for Data Responses
🔧 Set Custom DATA Output Header="<", Terminator=">"
● #: Loop1
📄 // Read the Tag ID, with a timeout of 1 second
🔍 Tag Read ID (Timeout:1000ms)
📄 // Loop Forever, until stopped
🔍 Branch To "Loop1"

```



注意

用户定义报头/报尾命令可用于在宏中的任何位置将响应类型的用户定义报头/报尾飞速更改为不同的值。

4 单标签命令

预期射频场中只有一个标签时，以下命令用于从标签读取/写入数据、锁定标签某些部分或对标签执行 EAS 操作。

- 🔍 Tag Search
- ← Tag Read ID
- ← Tag Read Data
- ← Start Continuous Read Data
- ✖ Stop Continuous Read Data
- Tag Write Data
- Tag Fill Data
- 🕒 Tag Write Timestamp
- ± Tag Increment/Decrement Data
- Tag Write AFI
- 🔒 Tag Lock AFI
- 🔒 Tag Lock Memory Blocks

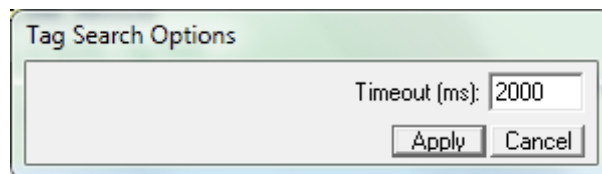
4.1 单标签搜索

选项	值
超时 (毫秒)	0-65535

此命令执行 *标签搜索* 命令。

执行此命令时，将向主机发送结果，要么是 *发现标签* 消息，要么是 *未发现标签* 错误。

将项目插入宏时，会出现以下对话框：



示例 1：

在以下示例中，*标签搜索* 命令用于确定是否存在标签，无需实际执行读取或写入操作。如果发现标签，则向主机发送 *发现标签* 消息。如果未发现标签，则会向主机发送 *未发现标签* 错误。

- **#: Start**
- ✖ On Error, Continue Macro
- ✖ Disable All Macro Responses (Silent Mode) (Legacy)
- 🔍 Tag Search (Timeout:2000ms)
- ◇ IF last RF command caused a "Tag Not Found" error then BRANCH to "No_Tag"
- 123 Transmit Custom String ("Tag Found!")
- END
- **#: No_Tag**
- 123 Transmit Custom String ("Tag Not Found!")
- END

4 单标签命令

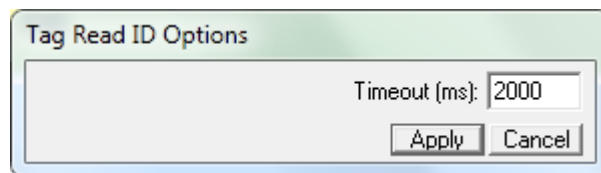
4.2 单标签 ID 读取

选项	值
超时 (毫秒)	0-65535

此命令执行 *标签 ID 读取* 命令。

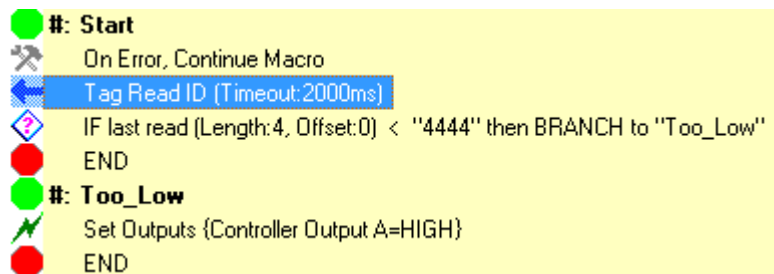
执行此命令时，将读取标签的 ID。如果成功读取 *标签 ID*，则将其发送到主机。如果未读取到 *标签 ID*，则向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：



示例 1：

在以下的示例中，*标签 ID 读取* 命令用于读取已发送到主机的 *标签 ID*。如果 *标签 ID* 的前 4 个字节小于“4444”，则设置控制器输出 A，否则宏结束。



4.3 单标签数据读取

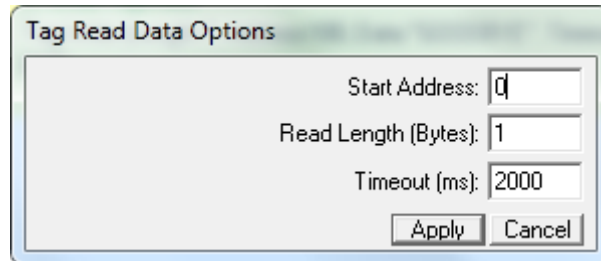
选项	值
起始地址	0-65535
读取长度 (字节)	0-247
超时 (毫秒)	0-65535

此命令执行 *标签数据读取* 命令。

执行此命令时，将从标签的起始地址开始读取指定长度的数据。如果成功读取数据，则将其发送到主机。如果未读取到数据，则向主机发送相应的错误。

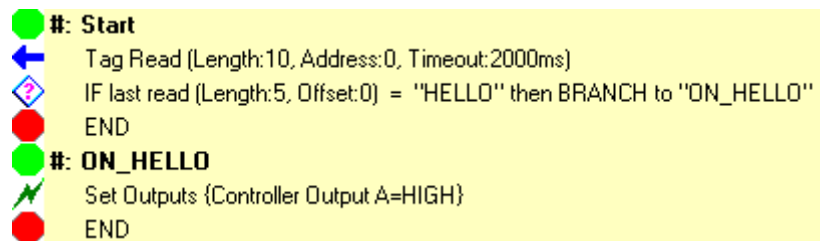
4 单标签命令

单标签数据读取 将项目插入宏时，会出现以下对话框：



示例 1:

在以下示例中，*标签数据读取* 命令用于从标签地址 0 开始读取 10 个字节，并将结果发送回主机。如果读取的前 5 个字节为“HELLO”，则设置控制器输出 A，否则宏结束。



4.4 单标签起始 连续读取数据

选项	值
起始地址	0-65535
读取长度 (字节)	0-247
超时 (毫秒)	0-65535

此命令执行*起始连续读取* 命令。

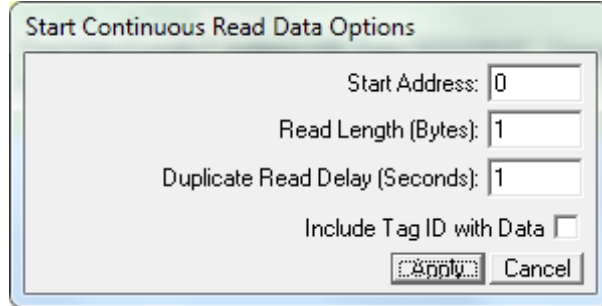
执行此命令时，*RFID 控制器* 开始在*连续读取模式* 下运行 – 但是在宏完成之前不会开始连续读取。

此命令不用于读取宏内部处理的实际数据 - 它仅指令 *RFID 控制器* 在宏终止后开始连续读取。

4 单标签命令

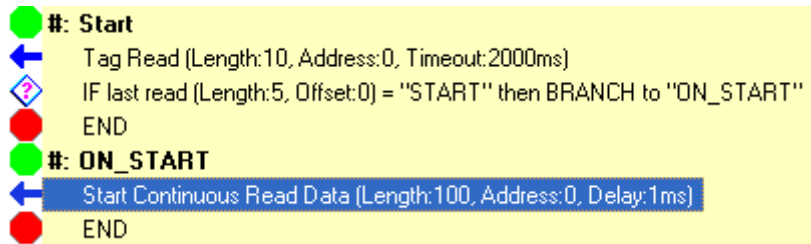
单标签起始
连续读取数据

将项目插入宏时，会出现以下对话框：



示例 1：

在以下的示例中，从标签的地址 0 读取 5 个字节，如果这些字节为“START”，则宏启动连续读取模式，从地址 0 读取 100 个字节。连续读取在宏结束时开始。



4.5 单标签停止 连续读取数据

选项	值
无	

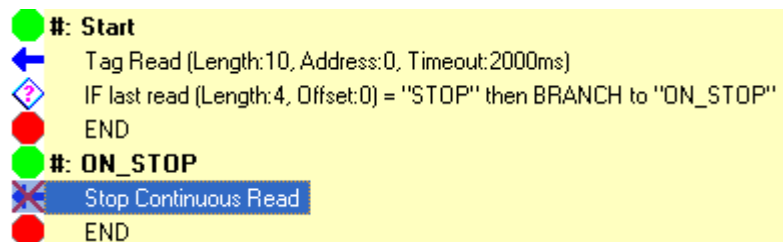
此命令执行停止连续读取命令。

执行此命令时，RFID 控制器停止在连续读取模式下的运行 – 当宏终止时，RFID 控制器将不会恢复连续读取。

由于此命令没有可编辑项目，因此当项目插入宏时，不会出现对话框。

示例 1：

在以下示例中，从标签的地址 0 读取 10 个字节，并将其发送到主机。如果前 4 个字节为“STOP”，则宏停止连续读取模式。



4 单标签命令

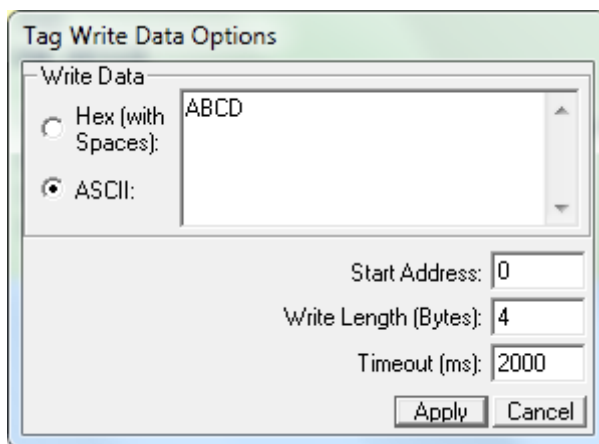
4.6 单标签数据写入

选项	值
起始地址	0-65535
写长度 (字节)	0-241
超时 (毫秒)	0-65535
写数据	ASCII 或十六进制数据的 0 到 241 字节
显示模式	十六进制或 ASCII

此命令执行标签数据写入命令。

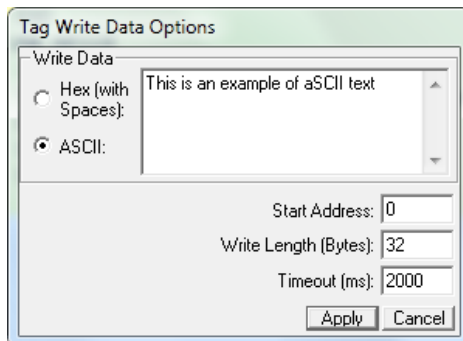
执行此命令时，数据将以指定的写入长度写入标签指定地址。如果成功写入数据，则向主机发送标签数据写入 ACK 响应。如果写入失败，将向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：

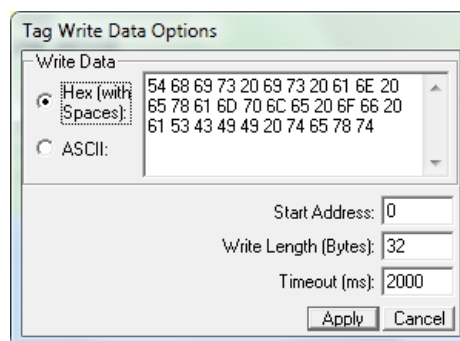


要输入要写入标签的数据，可以在写入数据框中键入 ASCII 文本或以空格分隔的 2 位十六进制值。

ASCII 数据示例：



十六进制数据示例：



4 单标签命令

单标签数据写入

无论哪种方式，写入标签的数据都是相同的 - 这只是为了方便数据输入，或者为了方便人读取。

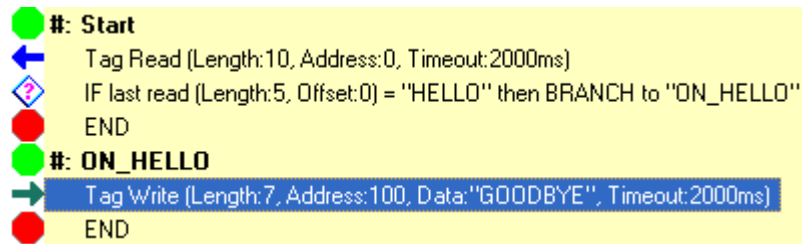


注意

将 ASCII 字符串“1”“2”“3”“4”“5”写入标签会写入十六进制值 31 32 33 34 35 – 必须认识到，在 ASCII 模式下输入数字时，这些数字本身就是 ASCII 字符，而不是等效的数值。如果要写入实际字节值 0x01 0x02 0x03 0x04 0x05，则需要使用十六进制数据输入模式。

示例 1:

在以下的示例中，读取标签的前 10 个字节，并将其发送到主机。如果前 5 个字节为“HELLO”，则使用 *标签数据写入* 命令将字符串“GOODBYE”写入标签中的标签地址 100。



4.7 单标签数据填充

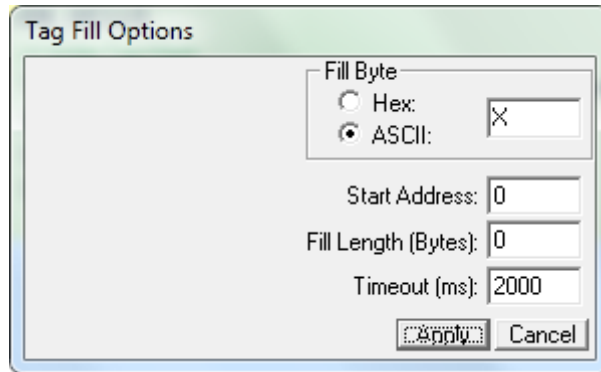
选项	值
起始地址	0-65535
填充长度 (字节)	0-241
超时 (毫秒)	0-65535
填充字节	ASCII 或十六进制数据的 1 字节
显示模式	十六进制或 ASCII

此命令执行 *标签数据填充* 命令。

执行此命令时，指定的填充字节将以指定长度写入标签的指定地址。如果成功写入数据，则向主机发送 *标签数据填充 ACK* 响应。如果写入失败，将向主机发送相应的错误。

4 单标签命令

单标签数据填充 将项目插入宏时，会出现以下对话框：



要输入要写入标签的填充字节，可以在填充字节框中键入 ASCII 字符或 2 位十六进制值。

无论哪种方式，写入标签的填充字节值都是相同的 – 这只是为了方便数据输入，或者为了方便人读取。

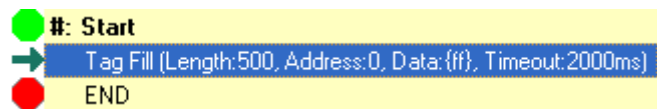


注意

将 ASCII 字符“1”写入标签将写入十六进制值 0x31 – 必须认识到，在 ASCII 模式下输入数字时，这些数字本身就是 ASCII 字符，而不是等效的数值。如果要写入实际字节值 0x01，则需要使用十六进制数据输入模式。

示例 1:

在以下示例中，使用 *标签数据填充* 命令从标签的地址 0 到地址 499 写入字节值 0xFF。



注意

将以填充长度为 0 且起始字节为 0 填充整个标签。

4 单标签命令

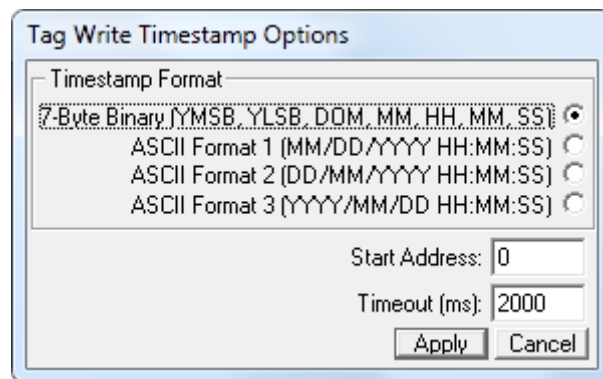
4.8 单标签写入 时间戳

选项	值
时间戳格式	字节二进制 ASCII 格式 1 ASCII 格式 2 ASCII 格式 3
起始地址	0-65535
超时 (毫秒)	0-65535

此命令执行 *标签写入时间戳* 命令。

执行此命令时，时间戳将以四种不同格式之一写入标签。时间取自 *RFID 控制器* 的内部实时时钟（如有）。如果写入时间戳，则向主机发送 *标签写入时间戳 ACK* 响应。如果标签写入失败，则向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：



四种时间戳格式为：

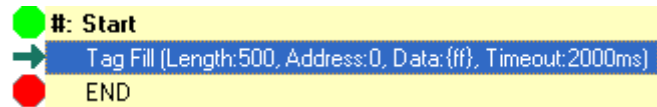
格式	说明
7-字节二进制 (格式 0)	此格式从指定的起始地址开始以 7 个字节写入时间戳，格式为年 MSB、年 LSB、日、月、时、分和秒。
ASCII 格式 1	此格式从指定的起始地址开始以 19 个字节写入时间戳，格式为 <i>MM/DD/YYYY HH:MM:SS</i>
ASCII 格式 2	此格式从指定的起始地址开始以 19 个字节写入时间戳，格式为 <i>DD/MM/YYYY HH:MM:SS</i>
ASCII 格式 3	此格式从指定的起始地址开始以 19 个字节写入时间戳，格式为 <i>YYYY/MM/DD HH:MM:SS</i>

4 单标签命令

单标签写入
时间戳

示例 1:

在以下示例中，*标签写入时间戳* 命令用于将当前时间戳以 *MM/DD/YYYY HH:MM:SS* 格式写入标签



4.9 单标签增量/ 减量数据

选项	值
操作	减量的增量
起始地址	0-65535
数据长度	1、2 或 4 字节值
按值	0-255 表示 1 字节值 0-65535 表示 2 字节值 0-4294967296 表示 4 字节值
超时 (毫秒)	0-65535

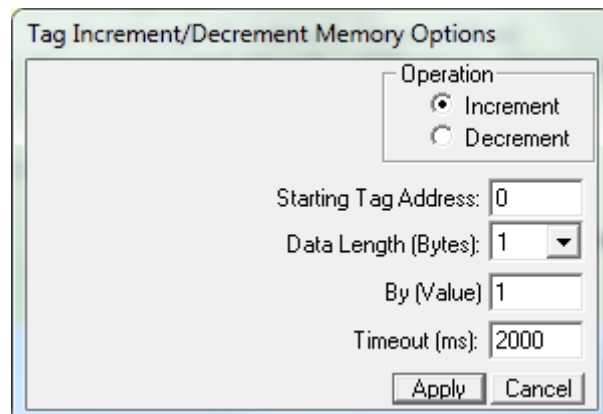
此命令执行 *标签增量* 或 *标签减量* 命令。

当执行此命令时，*RFID 控制器* 将在指定的 *起始地址* 读取指定长度的数据，并按指定的 *按值* 量递增或递减该值，然后将新值写回同一地址。如果成功写入结果，则向主机发送 *标签增量 (或减量) 数据 ACK* 响应。如果操作失败，则向主机发送相应的错误。

数据长度 选项指定递增或递减值的字节宽度。1 字节值将递增到 0xFF (255)，然后回绕到 0x00，而 2 字节值将递增到 0xFFFF (65535)，然后回绕到 0x0000。4 字节值将递增到 0xFFFFFFFF (4294967296)

这些值首先以 *MSB* 排列。

将项目插入宏时，会出现以下对话框：



4 单标签命令

单标签增量/
减量数据

示例 1:

在以下示例中，标签地址 0 处的 2 字节值递增。如果结果值超过 0x200 (512)，则将消息 *标签递增超过 512 倍* 发送到主机。

```

# Start
  Tag Increment Memory (Address:0, Length:2, By:1, Timeout:2000ms)
  Tag Read (Length:2, Address:0, Timeout:2000ms)
  IF last read (Length:2, Offset:0) > {02 00} then BRANCH to "Over_512_Times"
  END
# Over_512_Times
  Transmit Custom String ("This tag has been incremented over 512 times")
  END
    
```

4.10 单标签 AFI 写入

选项	值
新 AFI 字节	0-255
超时 (毫秒)	0-65535

此命令执行 *标签 AFI 写入* 命令。

执行此命令时，一个新的 AFI 字节值将写入标签。如果成功写入新 AFI 字节值，则向主机发送 *标签 AFI 写入 ACK* 响应。如果写入失败，将向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：

Tag Write AFI Options

New AFI Byte:

Timeout (ms):

示例 1:

在下面的示例中，值 128 作为新的 AFI 字节值写入标签。

```

# Start
  Tag Write AFI (New AFI:128, Timeout:2000ms)
  END
    
```

4 单标签命令

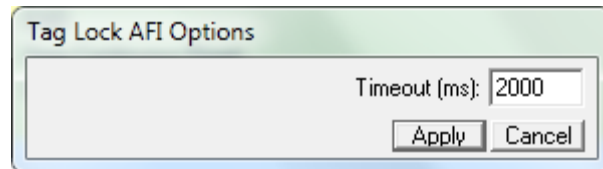
4.11 单标签 AFI 锁定

选项	值
超时 (毫秒)	0-65535

此命令用于锁定标签的 *AFI* 字段。

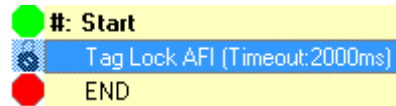
成功执行此命令后，标签的 *AFI* 字段将被永久锁定，并且永远无法更改 *AFI* 字节值。如果成功锁定 *AFI* 字节值，则向主机发送 *标签 AFI 锁定 ACK* 响应。如果命令失败，则向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：



示例 1:

在以下示例中，*AFI* 字节锁定在标签中。



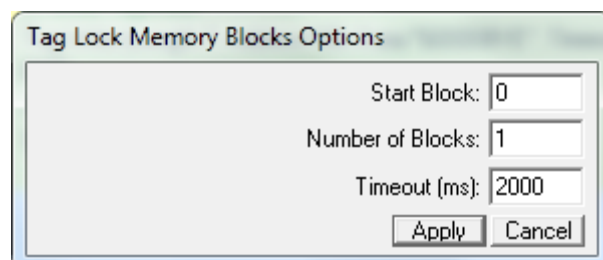
4.12 单标签锁定存储数据块

选项	值
起始数据块	0-255
数据块数	0-255
超时 (毫秒)	0-65535

此命令用于锁定标签的 *存储数据块*。

成功执行此命令后，标签的指定存储数据块将被永久锁定，并且永远无法更改数据块。如果命令成功，则向主机发送 *标签锁定存储数据块 ACK* 响应。如果命令失败，则向主机发送相应的错误。

将项目插入宏时，会出现以下对话框：



4 单标签命令

单标签锁定存储
数据块

示例 1:

在以下示例中，锁定标签从数据块 20 开始的 10 个存储数据块。

```
#: Start  
Tag Lock Memory Blocks (Number of Blocks:10, Starting Block:20, Timeout:2000ms)  
END
```



注意

不同标签类型的存储数据块大小不同。

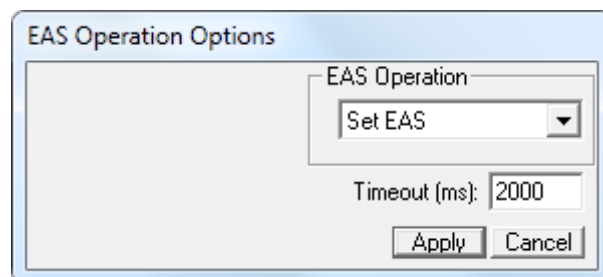
4.13 单标签 EAS 操作

选项	值
EAS 操作	设置、清除、锁定或警报
超时 (毫秒)	0-65535

此命令用于对标签执行以下 EAS 操作之一：

- 设置 EAS
- 清除 EAS
- 锁定 EAS
- 警报 EAS

将项目插入宏时，会出现以下对话框：



4 单标签命令

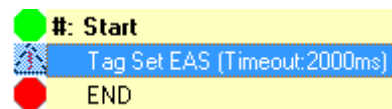
单标签 EAS 操作

四种 EAS 操作是：

操作	说明
设置 EAS	此操作设置标签的 EAS 位。设置此位后，标签将在执行警报 EAS 操作时做出响应。
清除 EAS	此操作清除标签的 EAS 位。清除此位后，标签将在执行警报 EAS 操作时不做出响应。
锁定 EAS	此操作将标签的 EAS 位永久锁定在其当前设置（设置或清除）。一旦锁定，就永远无法更改。
警报 EAS	此操作会导致其 EAS 位已设置的所有标签做出响应。没有 EAS 功能或清除了 EAS 位的标签将不做出响应。如果有任何标签响应，则响应将是标准命令 ACK 响应。否则，将生成“未发现标签”响应。

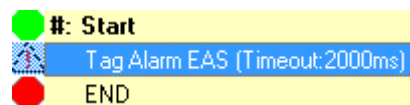
示例 1：

在以下示例中，标签中的 EAS 位被清除。



示例 2：

在以下示例中，警报 EAS 命令被发送到标签。如果已设置 EAS 位（如前一示例中所示），则标签将以警报响应，并且将警报 EAS ACK 响应返回主机。如果 EAS 位未设置（清除）或者不被标签支持，则会向主机返回未发现标签响应。



5 多标签命令

以下多标签命令用于在超时期间预期射频场中有多个标签的情况。

-  Multi-Tag Inventory
-  Multi-Tag Search
-  Multi-Tag Read Data
-  Multi-Tag Write Data
-  Multi-Tag Fill Data
-  Multi-Tag Write AFI
-  Multi-Tag Lock AFI
-  Multi-Tag EAS Operation

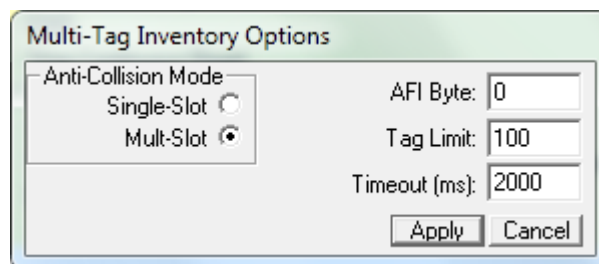
5.1 多标签遍询

选项	值
防冲突模式	单时隙或多时隙
AFI 字节	0-255
标签限制	0-255
超时 (毫秒)	0-65535

此命令执行多标签遍询命令。

执行此命令时，将为发现的每个标签生成一个包含标签 ID 的单独响应，并将其发送到主机。超时到期后，向主机发送包含整个超时期间发现的标签总数的最终 NTAGS 响应。

将项目插入宏时，会出现以下对话框：



AFI 字节值可用于筛选响应此命令的标签。AFI 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 AFI 字节值的标签

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要报告的最大标签数。

5 多标签命令

多标签遍询

示例 1:

在以下示例中，*多标签遍询* 命令与以下选项一起使用：

- 多时隙防冲突
- 标签限制为 10 个标签
- AFI 字节值为 25
- 超时 5 秒。

控制器先发现的 AFI 字节值为 25 的标签（最多 10 个）将向主机生成包含 *多标签遍询* 操作码和标签 ID 的响应。当总共发现 10 个标签，或达到 5 秒 *超时*（以先到者为准）时，将向主机返回包含发现的标签总数的最终 *NTAGS 响应*。

```

● #: Start
● Multi-Tag Inventory (AFI:25, Anti-Collision mode:Multi-Slot, Tag Limit:10, Timeout:5000ms)
● END
  
```

示例 2:

在以下示例中，*多标签遍询* 命令与以下选项一起使用

- 多时隙防冲突
- 标签限制为 100 个标签
- AFI 字节值为 0
- 超时 5 秒。

控制器先发现的标签（最多 100 个）将向主机生成包含 *多标签遍询* 操作码和标签 ID 的响应。当总共发现 100 个标签，或达到 5 秒 *超时*（以先到者为准）时，将向主机返回包含发现的标签总数的最终 *NTAGS 响应*。请注意，标签的 AFI 字节值可以是任意值，因为 AFI 字节使用了 0。

```

● #: Start
● Multi-Tag Inventory (AFI:0, Anti-Collision mode:Multi-Slot, Tag Limit:100, Timeout:5000ms)
● END
  
```

5.2 多标签搜索

选项	值
防冲突模式	单时隙或多时隙
AFI 字节	0-255
标签限制	0-255
超时（毫秒）	0-65535

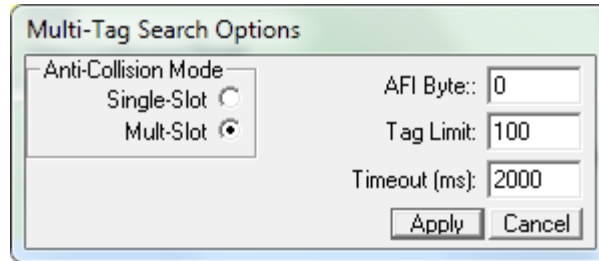
此命令执行 *多标签搜索* 命令。

此命令与上述 *多标签遍询* 命令完全相同，只是没有向主机返回的 *标签 ID*，只有包含发现的标签数的最终 *NTAGS 响应*。当仅需要射频场中的标签总数时，使用此命令。

5 多标签命令

多标签搜索

将项目插入宏时，会出现以下对话框：



AFI 字节值可用于筛选响应此命令的标签。*AFI* 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 *AFI* 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

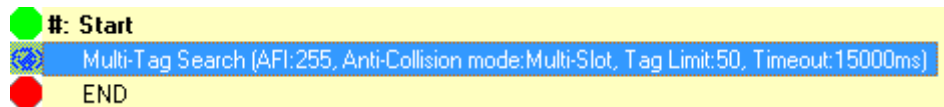
标签限制 指定要搜索的最大标签数。

示例 1:

在以下示例中，多标签搜索命令与以下选项一起使用：

- 多时隙防冲突
- 标签限制为 50 个标签
- *AFI* 字节值为 255
- 超时 15 秒。

控制器将查找 *AFI* 字节值为 255 的标签，直到发现 50 个标签，或达到 15 秒超时。包含发现的标签总数的 *NTAGS* 响应将返回到主机。



5.3 多标签数据读取

选项	值
防冲突模式	单时隙或多时隙
起始地址	0-65535
读取长度 (字节)	0-243
<i>AFI</i> 字节	0-255
标签限制	0-255
超时 (毫秒)	0-65535
包含 ID 和数据	选中或未选中

5 多标签命令

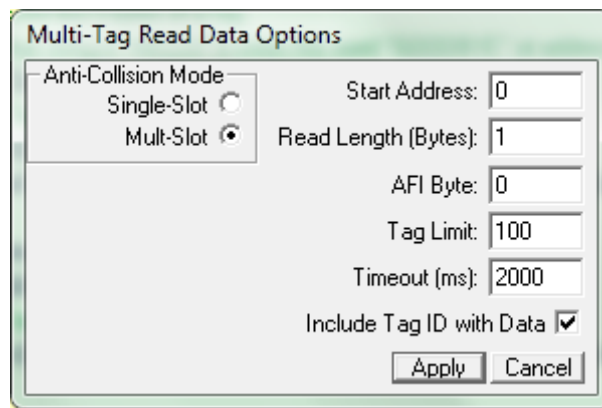
多标签数据读取

此命令执行多标签数据读取命令。

此命令用于从射频场中的多个标签读取数据。执行此命令时，将从射频场中发现的具有指定 AFI 字节值的每个标签的指定起始地址读取指定读取长度的数据。对于发现并读取的每个标签，将生成包含从标签读取的数据的单独响应并将其发送到主机。如果选中了包含 ID 和数据，则标签 ID 也将包含在每个响应中。

超时到期后，向主机发送包含整个超时期间读取的标签总数的最终 NTAGS 响应。

将项目插入宏时，会出现以下对话框：



AFI 字节值可用于筛选使用此命令读取的标签。AFI 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 AFI 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要写入的最大标签数。

示例 1：

在以下示例中，使用多标签数据读取命令从多个标签的字节 0 开始读取 20 个字节。命令中的其他选项包括：

- 多时隙防冲突
- 标签限制为 100 个标签
- AFI 字节值为 128
- 超时 15 秒
- 数据响应中包含的标签 ID

执行命令时，控制器先发现的 AFI 字节值为 128 的标签（最多 100 个）将生成包含多标签数据读取操作码、标签 ID 和从每个标签读取的指定数据的响应。

当总共发现 100 个标签，或达到 15 秒超时（以先到者为准）时，将向主机返回包含发现并读取的标签总数的最终 NTAGS 响应。

5 多标签命令

多标签数据读取

```

# Start
Multi-Tag Read Data (AFI:128, Anti-Collision mode:Multi-Slot, Length:20, Address:0, Tag Limit:100, Timeout:15000ms, Include ID:Yes)
END
    
```

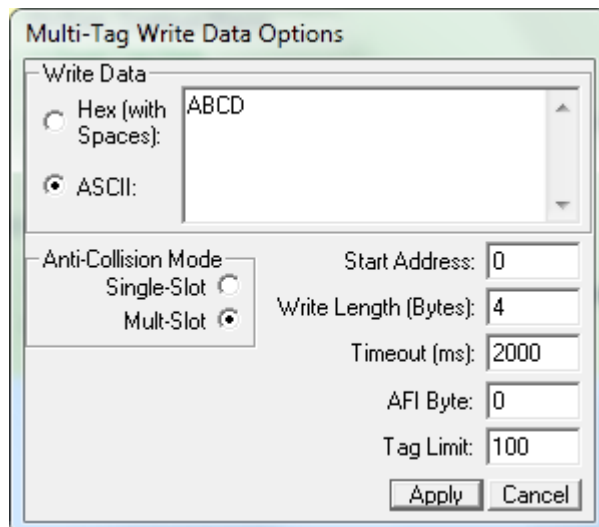
5.4 多标签数据写入

选项	值
防冲突模式	单时隙或多时隙
起始地址	0-65535
写长度 (字节)	0-247
AFI 字节	0-255
标签限制	0-255
超时 (毫秒)	0-65535
写数据	ASCII 或十六进制数据的 0 到 237 字节
显示模式	ASCII 或十六进制

此命令执行多标签数据写入命令。

此命令用于将数据写入射频场中的多个标签。执行此命令时，指定的写入数据将以指定的写入长度在指定的起始地址写入到射频场中发现的具有指定 AFI 字节值的每个标签。对于发现并写入的每个标签，将生成包含标签 ID 的单独响应并发送到主机。超时到期后，向主机发送包含整个超时期间写入的标签总数的最终 NTAGS 响应。

将项目插入宏时，会出现以下对话框：



5 多标签命令

多标签数据写入

AFI 字节值可用于筛选使用此命令写入的标签。*AFI* 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 *AFI* 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要写入的最大标签数。

示例 1:

在以下示例中，使用多标签数据写入命令从多个标签的字节 0 开始写入“HELLO”。命令中的其他选项包括：

- 多时隙防冲突
- 标签限制为 100 个标签
- *AFI* 字节值为 128
- 超时 15 秒。

执行命令时，控制器先发现并成功写入“HELLO”的 *AFI* 字节值为 128 的标签（最多 100 个）将生成包含多标签数据写入操作码、和标签 *ID* 的响应。

当总共发现 100 个标签，或达到 15 秒超时（以先到者为准）时，将向主机返回包含发现并写入的标签总数的最终 *NTAGS* 响应。

```

● #: Start
● Multi-Tag Write Data (AFI:128, Anti-Collision mode: Multi-Slot, Length:5, Address:0, Data:"HELLO", Tag Limit:100, Timeout:15000ms)
● END
  
```

5.5 多标签数据填充

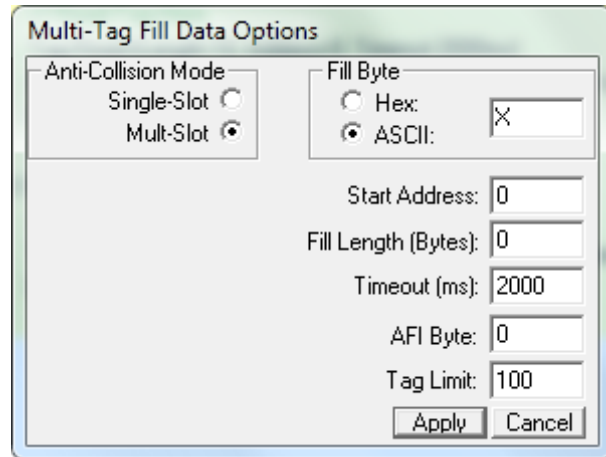
选项	值
防冲突模式	单时隙或多时隙
填充地址	0-65535
填充长度（字节）	0-65535
<i>AFI</i> 字节	0-255
标签限制	0-255
超时（毫秒）	0-65535
填充字节	! ASCII 或十六进制填充字节

此命令执行多标签数据填充命令。

此命令用于填充射频场中的多个标签的某些部分。执行此命令时，指定的填充字节值将以指定的填充长度字节在指定的起始地址写入到射频场中发现的具有指定 *AFI* 字节值的每个标签。对于发现并写入的每个标签，将生成包含标签 *ID* 的单独响应并发送到主机。超时到期后，向主机发送包含整个超时期间写入的标签总数的最终 *NTAGS* 响应。

5 多标签命令

多标签数据填充 将项目插入宏时，会出现以下对话框：



起始地址 为 0 和 填充长度为 0 表示填充整个标签。

AFI 字节值可用于筛选使用此命令写入的标签。AFI 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 AFI 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要写入的最大标签数。

示例 1：

在以下示例中，多标签数据填充命令用于将填充长度为 50 字节的填充字节 0x00 从起始地址 100 写入多个标签。命令中的其他选项包括：

- 多时隙防冲突
- 标签限制为 100 个标签
- AFI 字节值为 0
- 超时 15 秒。

执行命令时，控制器先发现并从地址 100 开始成功写入 50 个字节的 0x00 的标签（最多 100 个）将生成包含多标签数据填充操作码和标签 ID 的响应。

当总共发现 100 个标签，或达到 15 秒超时（以先到者为准）时，将向主机返回包含发现并写入的标签总数的最终 NTAGS 响应。

```
##: Start
Multi-Tag Fill Data (AFI:0, Anti-Collision mode:Multi-Slot, Length:50, Address:100, Data:{00}, Tag Limit:100, Timeout:15000ms)
END
```

5 多标签命令

多标签数据填充

示例 2:

在以下示例中，使用多标签数据填充命令以 255 (0xFF) 字节完全填充多个标签。命令中的其他选项包括：

- 多时隙防冲突
- 标签限制为 100 个标签
- AFI 字节值为 0
- 超时 15 秒。

执行命令时，控制器先发现并成功填充 0xFF 的标签（最多 100 个）将生成包含多标签数据填充操作码和标签 ID 的响应。

当总共发现 100 个标签，或达到 15 秒超时（以先到者为准）时，将向主机返回包含发现并填充的标签总数的最终 NTAGS 响应。

```

# Start
Multi-Tag Fill Data (AFI:0, Anti-Collision mode:Multi-Slot, Length:0, Address:0, Data:{FF }, Tag Limit:100, Timeout:15000ms)
END
    
```

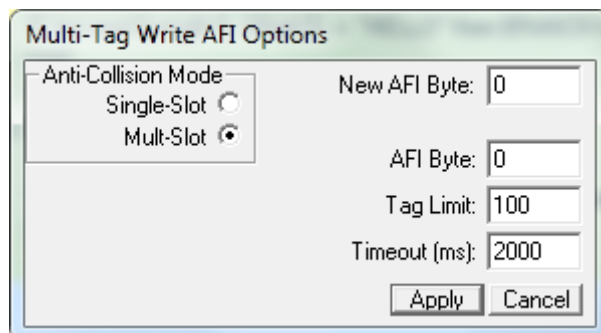
5.6 多标签 AFI 写入

选项	值
防冲突模式	单时隙或多时隙
新 AFI 字节	0-255
AFI 字节	0-255
标签限制	0-255
超时（毫秒）	0-65535

此命令执行多标签 AFI 写入命令。

此命令用于将新 AFI 字节写入射频场中的多个标签。执行此命令时，指定的新 AFI 字节将写入射频场中具有指定 AFI 字段值的每个标签。对于发现并写入的每个标签，将生成包含标签 ID 的单独响应并发送到主机。超时到期后，向主机发送包含整个超时期间写入的标签总数的最终 NTAGS 响应。

将项目插入宏时，会出现以下对话框：



5 多标签命令

多标签 AFI 写入

AFI 字节值可用于筛选使用此命令写入的标签。AFI 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 AFI 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要写入的最大标签数。

示例 1:

在以下示例中，多标签 AFI 写入命令用于将多个标签的 AFI 字节值从 1 更改为 2。命令中的其他选项包括：

- 多时隙防冲突
- 标签限制为 100 个标签
- 超时 15 秒。

执行命令时，控制器先发现并成功写入新 AFI 字节值 2 的 AFI 字节值为 1 的标签（最多 100 个）将生成包含多标签 AFI 写入操作码和标签 ID 的响应。

当总共发现 100 个标签，或达到 15 秒超时（以先到者为准）时，将向主机返回包含发现并写入的标签总数的最终 NTAGS 响应。

```

# Start
Multi-Tag Write AFI (AFI:1, Anti-Collision mode:Multi-Slot, New AFI:2, Max Tags:100, Timeout:15000ms)
END
    
```

5.7 标签 AFI 锁定

选项	值
防冲突模式	单时隙或多时隙
AFI 字节	0-255
标签限制	0-255
超时（毫秒）	0-65535

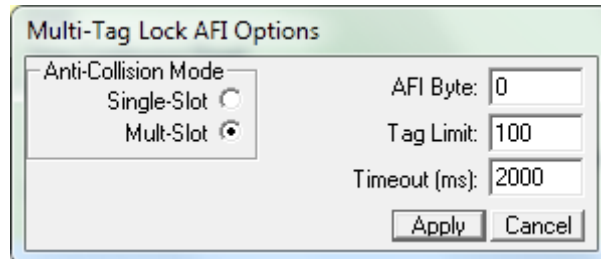
此命令执行标签 AFI 锁定命令。

此命令用于锁定射频场中多个标签的 AFI 字节值。执行此命令时，射频场中发现的具有指定 AFI 字节值的每个标签中的 AFI 字节将被锁定。对于发现并锁定的每个标签，将生成包含标签 ID 的单独响应并发送到主机。超时到期后，向主机发送包含整个超时期间锁定的标签总数的最终 NTAGS 响应。

5 多标签命令

标签 AFI 锁定

将项目插入宏时，会出现以下对话框：



AFI 字节值可用于筛选使用此命令锁定的标签。*AFI* 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 *AFI* 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要锁定的最大标签数。

示例 1：

在以下示例中，*标签 AFI 锁定* 命令用于永久锁定多个标签的 *AFI* 字节值。命令中的其他选项包括：

- *多时隙防冲突*
- *标签限制* 为 100 个标签
- *AFI* 字节值为 0
- *超时* 15 秒。

执行命令时，控制器先发现并成功锁定 *AFI* 字节值的标签（最多 100 个）将生成包含 *标签 AFI 锁定* 操作码和 *标签 ID* 的响应。

当总共发现 100 个标签，或达到 15 秒 *超时*（以先到者为准）时，将向主机返回包含发现并锁定的标签总数的最终 *NTAGS* 响应。

```

# Start
Multi-Tag Lock AFI (AFI:0, Anti-Collision mode:Multi-Slot, Tag Limit:100, Timeout:15000ms)
END
  
```

5 多标签命令

5.8 多标签 EAS 操作

选项	值
防冲突模式	单时隙或多时隙
EAS 操作	设置、清除、锁定或警报
AFI 字节	0-255
标签限制	0-255
超时 (毫秒)	0-65535

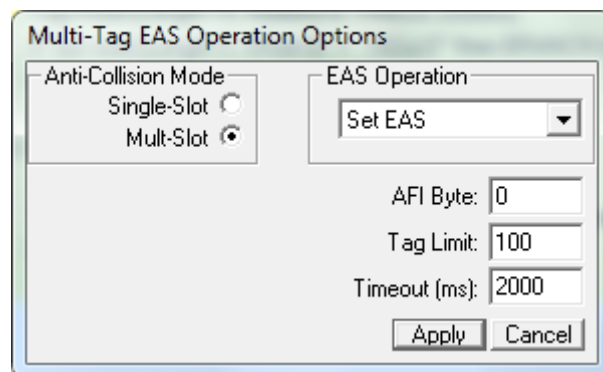
此命令对射频场中的多个标签执行四个多标签 EAS 操作命令之一。

执行此命令时，将对射频场中发现的每个标签执行指定的 EAS 操作。对于发现并操作的每个标签，将生成包含标签 ID 的单独响应并发送到主机。超时到期后，向主机发送包含整个超时期间对其执行操作的标签总数的最终 NTAGS 响应。

四种 EAS 操作是：

操作	说明
设置 EAS	此操作设置标签的 EAS 位。设置此位后，标签将在执行警报 EAS 操作时做出响应。
清除 EAS	此操作清除标签的 EAS 位。清除此位后，标签将在执行警报 EAS 操作时不做出响应。
锁定 EAS	此操作将标签的 EAS 位永久锁定在其当前设置（设置或清除）。一旦锁定，就永远无法更改。
警报 EAS	此操作会导致其 EAS 位已设置的所有标签做出响应。没有 EAS 功能或清除了 EAS 位的标签将不做出响应。如果有任何标签响应，响应将包含标签 ID（不同于单个标签 EAS 操作命令）

将项目插入宏时，会出现以下对话框：



5 多标签命令

多标签 EAS 操作

AFI 字节值可用于过滤根据此命令操作的标签。*AFI* 字节值为 0 表示所有标签，而任意其他非零值表示仅具有此 *AFI* 字节值的标签。

防冲突模式 确定使用的 ISO15693 防冲突模式。

标签限制 指定要在其上执行操作的最大标签数。

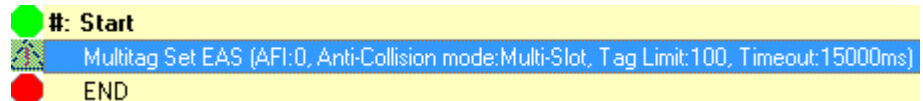
示例 1:

在以下示例中，使用 *多标签设置 EAS* 命令设置多个标签的 EAS 位。命令中的其他选项包括：

- *多时隙防冲突*
- *标签限制* 为 100 个标签
- *AFI* 字节值为 0
- *超时* 15 秒。

执行命令时，控制器先发现并成功设置 EAS 位的标签（最多 100 个）将生成包含 *多标签设置 EAS* 操作码和 *标签 ID* 的响应。

当总共发现 100 个标签，或达到 15 秒 *超时*（以先到者为准）时，将向主机返回包含发现并设置了其 EAS 位的标签总数的最终 *NTAGS* 响应。



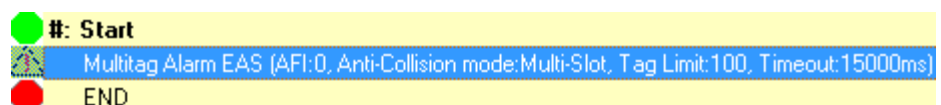
示例 2:

在以下示例中，使用 *多标签 EAS 警报* 命令查找射频场中设置了 EAS 位的所有标签。命令中的其他选项包括：

- *多时隙防冲突*
- *标签限制* 为 100 个标签
- *AFI* 字节值为 0
- *超时* 15 秒。

执行命令时，控制器先发现并设置了其 EAS 位的标签（最多 100 个）将生成包含 *多标签 EAS 警报* 操作码和 *标签 ID* 的响应。

当总共发现 100 个标签，或达到 15 秒 *超时*（以先到者为准）时，将向主机返回包含发现并设置了其 EAS 位的标签总数的最终 *NTAGS* 响应。



6 设置/清除输出命令

以下输出命令用于设置或清除 RFID 控制器（或 Hub）的输出。

- ⚡ Set Outputs
- ⚡ Clear Outputs

6.1 设置输出

选项	值
设置 Hub 输出 A、B、C、D	选中或未选中
设置控制器输出 A、B	选中或未选中

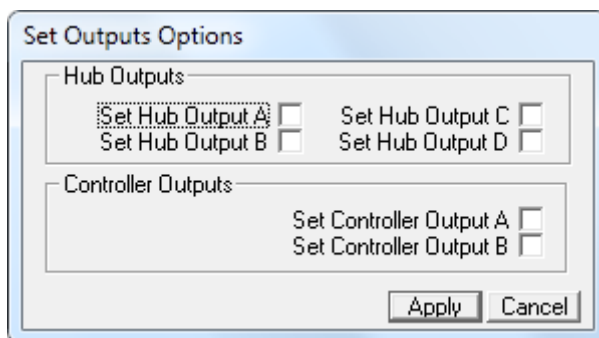
此命令设置输出。

可以使用设置输出命令设置两种类型的输出：

Hub 输出 或 **控制器输出**。对于有输出的 RFID 控制器，此命令将指定的输出信号设置为高。对于 Hub，此命令将向 Hub 发送 Subnet16 消息，指示 Hub 将指定的输出信号设置为高。

这可让宏直接控制连接到控制器或 Hub 的外部硬件，如传送带、分流器、多层指示灯等。

将项目插入宏时，会出现以下对话框：



注意

可以使用此命令设置 Hub 输出或控制器输出，但不能在同一命令中同时设置这两种类型。



注意

如果未设置输出，则此命令不会将其清除。此命令仅影响明确设置的输出。

6 设置/清除输出命令

设置输出

示例 1:

在以下示例中，读取标签 ID。如果 ID 的第一个数字小于 5，则设置 Hub 输出 A。如果第一个数字大于 8，则设置 Hub 输出 B。

```

#: Start
// This macro sets hub output a if the first byte of a tag is less than the character "5".
// And sets hub output b if the first byte is greater than the character "8"
On Error, Halt Macro
Tag Read ID (Timeout:2000ms)
IF last read (Length:1, Offset:0) < "5" then BRANCH to "LessThanFive"
IF last read (Length:1, Offset:0) > "8" then BRANCH to "GreaterThanEight"
END
#: LessThanFive
Set Outputs {Hub Output A=HIGH}
END
#: GreaterThanEight
Set Outputs {Hub Output B=HIGH}
END
    
```

示例 2:

在以下示例中，读取标签 ID。如果出现错误，则设置控制器输出 A。

```

#: Start
On Error, Continue Macro
Tag Read ID (Timeout:2000ms)
IF last RF command caused an error then BRANCH to "Error"
END
#: Error
Set Outputs {Controller Output A=HIGH}
END
    
```

6.2 清除输出

选项	值
设置 Hub 输出 A、B、C、D	选中或未选中
设置控制器输出 A、B	选中或未选中

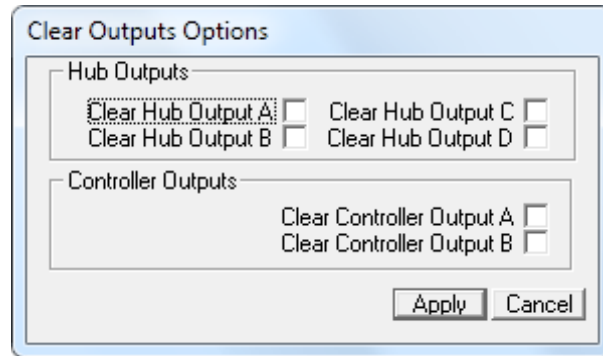
此命令清除输出。

此命令与设置输出完全相同，只是用于清除输出 - 将其设置为低。

6 设置/清除输出命令

清除输出

将项目插入宏时，会出现以下对话框：



可以使用此命令清除 *Hub* 输出 或 *控制器输出*，但不能在同一命令中同时清除这两种类型。

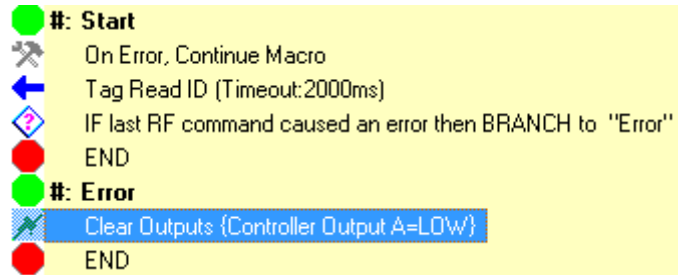


注意

如果未清除输出，则此命令不会设置该输出。此命令仅影响明确清除的输出。



示例 1:

在以下示例中，读取 *标签 ID*。如果出现错误，则清除 *控制器输出 A*。



7 宏专用命令

以下命令是宏专用命令，用于完成与宏执行相关的各种特殊功能的命令，例如对结果进行条件检查、将宏执行控制转移到宏的其他部分（甚至其他宏）、标记宏的执行点、传输自定义消息、插入延迟、插入描述性注释或结束宏执行。

-  Label
-  Branch
-  Conditional Branch
-  Transmit Last Response
-  Transmit Custom String
-  Execute Macro
-  Wait
-  Note
-  End

7.1 标记

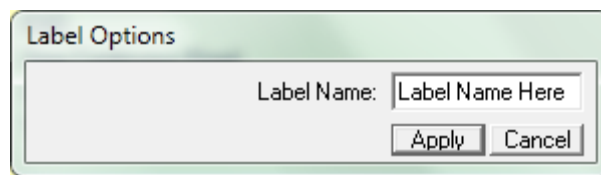
选项	值
标记名称	1-255 个字符

标记是宏程序中特定点的名称。标记使程序能够将宏的执行分支或重新引导到宏的其他部分。

每个宏都有一个固定在宏的开头名为开始的初始标记。这是唯一无法从工作区中删除的项目。

标记不占用任何程序空间，也不执行任何任务 - 您可以添加任意数量的标记。它们的存在只是为了让宏编译器通过分支指令（本章后面将介绍）识别分支到（或转到）的位置。

将项目插入宏时，会出现以下对话框：



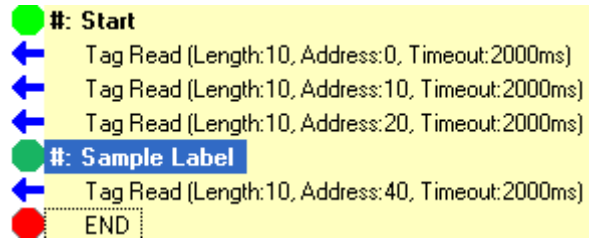
标记名称最多可包含 255 个字符 - 但实际上，标记名称应短，长度为 1 到 16 个字符左右。它们可以包含任何可键入的字符（包括空格）并且区分大小写。

7 宏专用命令

标记

示例 1:

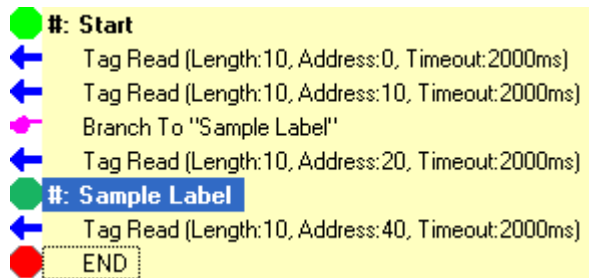
在以下的示例中，一个简单宏中已插入一个名为“示例标记”的标记。



在此示例中，标记没有实际用途。宏将简单地跳过标记并按顺序执行所有四个读取指令。

示例 2:

在此宏中，还插入了一条简单的分支指令：



在此宏中，将执行前两次读取，但随后该宏将分支并跳过第三次读取，直接进行第四次读取。在此宏中，在任何情况下都不会执行第三次读取。

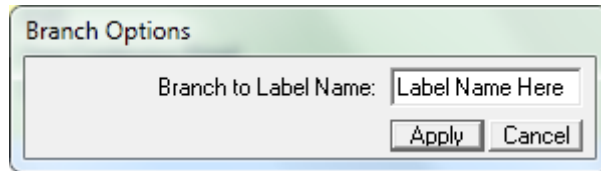
7.2 分支

选项	值
分支标记名称	1-255 个字符

此命令使宏执行流无条件地分支到宏的另一个位置。这要求已在分支命令的目标处创建标记。这相当于 *GoTo* 命令。

7 宏专用命令

分支 将项目插入宏时，会出现以下对话框：



注意

如果在您插入此命令时标记实际上不存在，那没关系 – 您会在工作区中的命令项旁边看到一条警告消息，上面写着 “(** NOT FOUND)” – 这表明宏编译器还不知道您实际上要在哪里分支 – 您仍然需要插入具有相应名称的标记。

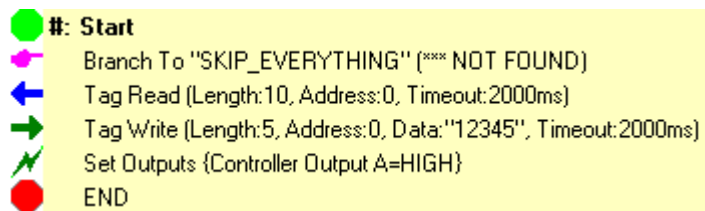


注意

对于所有分支命令，应注意不要使宏进入无限循环，除非这是需要的特性。

示例 1：

在本例中，在宏的开头插入了一个转到名为 “SKIP_EVERYTHING” 的标记的分支。由于没有名为 “SKIP_EVERYTHING” 的标记，因此会显示警告消息 (** NOT FOUND)。



注意

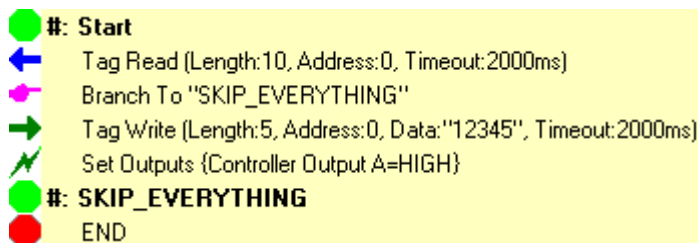
如果加载并执行此宏，将发生致命错误。

7 宏专用命令

分支

示例 1:

在此示例中，已在宏结束之前插入了名为“SKIP_EVERYTHING”的相应标记。执行此宏时，分支指令将使宏跳过接下来的三个命令，并在标记处继续执行 - 在本例中，只是结束宏。



7.3 条件分支

选项	值
分支条件	符合下列条件之一： 如果上次“读取”结果 < 比较数据 如果上次“读取”结果 > 比较数据 如果上次“读取”结果 = 比较数据 如果上次“读取”结果 != 比较数据 如果上次“读取”结果“包含”比较数据（忽略偏离） 如果上次“读取”或“写入”导致任何错误 如果上次“读取”导致“读取”错误 如果上次“写入”导致“写入”错误 如果上次“读取”或“写入”导致“未发现标签”错误 如果上次“读取”或“写入”导致（旧版） “未发现标签”错误 如果控制器“输入” A = 低 如果控制器“输入” A = 高 如果控制器“输入” B = 低 如果控制器“输入” B = 高 如果控制器“输入” A = 低且“输入” B = 低 如果控制器“输入” A = 高且“输入” B = 低 如果控制器“输入” A = 低且“输入” B = 高 如果控制器“输入” A = 高且“输入” B = 高 如果控制器“输出” A = 低 如果控制器“输出” A = 高 如果控制器“输出” B = 低 如果控制器“输出” B = 高 如果控制器“输出” A = 低且“输出” B = 低 如果控制器“输出” A = 高且“输出” B = 低 如果控制器“输出” A = 低且“输出” B = 高 如果控制器“输出” A = 高且“输出” B = 高

7 宏专用命令

条件分支

续表：

选项	值
比较起始	0-65535
比较长度	0-255
比较数据	ASCII 或十六进制比较数据的 0 到 255 字节
分支到标记	现有标记名称 (1-10 个字符)

此命令导致宏执行流根据先前 RFID 命令的结果或错误有条件地分支到宏的另一个位置：

- 标签搜索
- 标签 ID 读取
- 标签数据读取
- 标签数据写入
- 标签数据填充
- 控制器输入/输出状态

如果满足条件, 此命令要求已在分支的目标位置创建标记。这相当于 *If-Then-GoTo* (如果 - 那么 - 转到) 命令。

将项目插入宏时, 会出现以下对话框：

上部名为分支条件的部分是一个包含 26 种不同条件、一些涉及的数据、一些涉及的错误、一些涉及的输入或输出的列表框。

前五个选项允许基于先前的标签数据读取或标签 ID 读取命令的结果进行分支。

7 宏专用命令

条件分支

接下来的五个选项允许基于先前 RFID 命令生成的错误进行分支：

- 标签搜索
- 标签 ID 读取
- 标签数据读取
- 标签数据写入
- 标签数据填充

最后 16 个选项允许根据 *控制器输入/输出* 的状态组合进行分支。

在中间，有一个用于输入 *比较数据串* 以及 *偏离* 和 *长度* 的 *数据输入框*。比较数据仅用于前 5 个条件选项。

注意：这里的偏离是上次读取结果中的偏离，而不是标签地址偏离。因此，例如，如果您只是从标签的地址 100 读取 8 个字节，并且您想要比较这些字节中的最后 2 个，那么您将使用的比较起始（偏离）为 6，比较长度为 2。

注意：所有比较（如“小于”和“大于”）都是逐字节比较的。因此，实际十六进制字节值为 41 42 43 44 的字符串“ABCD”比实际字节值为 61 62 63 64 的字符串“ABCD”小。同样，字符串“Cat”（43 61 74）比字符串“CAT”（43 41 54）“小”

最后，还有一个指示在满足条件时宏要分支到标记的“分支到标记”名称。



注意

注意：这里的 *偏离* 是上次读取结果中的 *偏离*，而不是标签地址 *偏离*。

因此，例如，如果您只是从标记的 *起始地址* 100 读取 8 个字节，并且您想要比较这些字节中的最后 2 个，那么您将使用的 *比较起始(偏离)* 为 6，*比较长度* 为 2。



注意

所有比较，（如 *小于* 和 *大于*）都是逐字节比较的。因此，实际十六进制字节值为 0x41 0x42 0x43 0x44 的字符串“ABCD”比实际字节值为 0x61 0x62 0x63 0x64 的字符串“ABCD”小。同样，字符串“Cat”（0x43 0x61 0x74）*小于* 字符串“CAT”（0x43 0x41 0x54）。

7 宏专用命令

条件分支

最后，还有一个指示在满足条件时宏要分支到 *标记* 的分支到 *标记名称*。



注意

如果在您插入此命令时 *标记* 实际上不存在，那没关系 – 您会在工作区中的命令项旁边看到一条警告消息，上面写着 “(*** NOT FOUND)” – 这表明宏编译器还不知道您实际上要在哪里分支 – 您仍然需要插入具有相应名称的 *标记*。

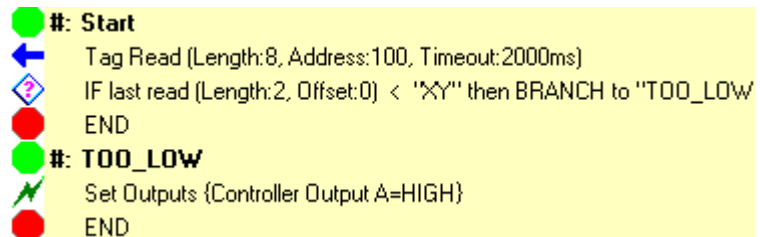


注意

对于所有分支命令，应注意不要使宏进入 *无限循环*，除非这是需要的特性。

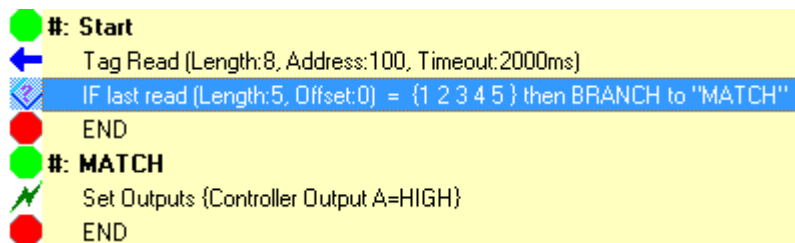
示例 1:

在以下示例中，如果读取的前 8 个字节中的最后 2 个字节 *小于* 字符串 “XY”，则宏将使用 *条件分支* 命令分支到 “TOO_LOW” 并设置 *控制器输出 A*，否则宏将结束。



示例 2:

在以下示例中，如果读取的前 8 个字节中的前 5 个字节为字节值 0x01 0x02 0x03 0x04 0x05，则宏将使用 *条件分支* 命令分支到 “MATCH” 并设置 *控制器输出 A*，否则宏将结束。

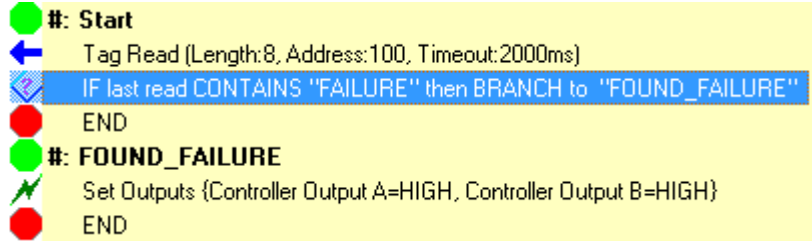


7 宏专用命令

条件分支

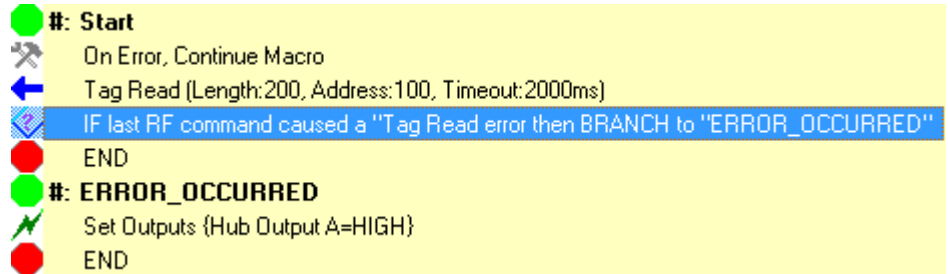
示例 3:

在下面示例中，如果在读取的前 100 个字节的任意位置发现字符串“FAILURE”，宏将使用条件分支命令分支到“FOUND_FAILURE”并设置控制器输出 A 和 B，否则宏将结束。



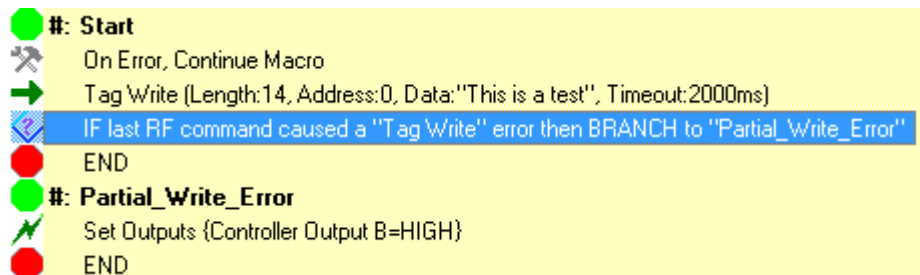
示例 4:

在以下示例中，如果尝试的标签数据读取命令因任何原因而失败，宏将使用条件分支命令分支到“ERROR_OCCURRED”并设置 Hub 输出 A，否则宏将结束。请注意，需要将出错-继续宏命令插入宏中，否则宏只会在出现错误时结束，而不会设置输出。



示例 5:

在以下示例中，如果尝试的标签写入命令因标签写入错误而失败，宏将使用条件分支命令分支到“Partial_Write_Error”并设置控制器输出 B，否则宏将结束。请注意，需要将出错-继续宏命令插入宏中，否则宏只会在出现错误时结束，而不会设置输出。未发现标签错误不会导致设置 Hub 输出—只会导致实际的标签写入失败。这在根本未写入标签不是问题但标签被部分写入的应用中非常有用。



7 宏专用命令

7.4 传输最后响应

选项	值
无	

此命令可使以下命令的最后结果/响应再次发送到主机：

- 标签搜索
- 标签 ID 读取
- 标签数据读取
- 标签数据写入
- 标签数据填充

此命令通常与禁用所有主机响应一起使用，以使命令可以执行（例如标签读取），以便宏在决定是否应将结果实际发送回主机前对结果进行一些比较。从而过滤响应，并且可以提高应用的效率。

由于此命令没有可编辑项目，因此当项目插入宏时，不会出现对话框。

示例 1：

在此示例中，禁用所有响应类型，读取标签 ID，如果标签 ID 的最后一个字节小于 5，则只有在那时，才通过传输最后响应命令将标签 ID 发送到主机。

```

#: Start
// This macro sets controller output A, if the first byte of a tag is less than the character "5",
// otherwise it just returns the read data
Set Macro Response Options: ACKS:No, DATA:No, ERRS:No, NTAGS:No, STRINGS:No
// First, disable all responses
Tag Read (Length:10, Address:0, Timeout:2000ms)
IF last read (Length:1, Offset:0) < "5" then BRANCH to "LessThanFive"
Transmit Last Response
END

#: LessThanFive
Set Outputs {Controller Output A=HIGH}
END
    
```

7 宏专用命令

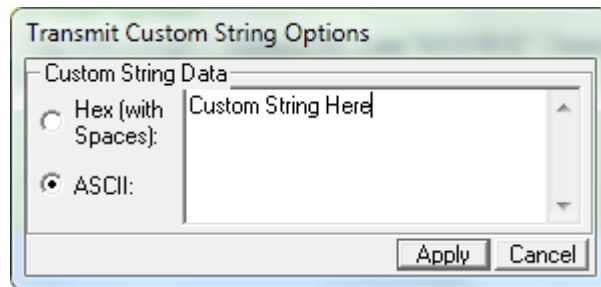
7.5 传输用户定义字符串

选项	值
用户定义字符串数据	ASCII 或十六进制数据的 0 到 247 字节

此命令使 *RFID 控制器* 将数据的 *用户定义字符串* 传输回主机。这使得发送回主机的信息更具灵活性，并能够在某些可以将数据视为文本的应用中发送人可读的信息数据。

此命令在调试宏时也非常有用，通过在宏的各个点插入许多 *传输自定义字符串* 命令，您可以跟踪宏的执行情况。

将项目插入宏时，会出现以下对话框：



此外，此命令的 *显示模式* 可以设置为十六进制（带空格）或 ASCII。

要输入要返回的 *用户定义字符串*，可以在 *写入数据* 框中键入 ASCII 文本或以空格分隔的 2 位十六进制字节值。

示例 1：

在以下示例中，从标签中读取两个字节。如果这两个字节是 0x01 0x02，则将 *自定义字符串* “批处理通过” 发送到主机。如果两个字节为 0x03 0x04，则将 *用户定义字符串* “批处理失败” 发送到主机。

```

● #: Start
✖ On Error, Continue Macro
✖ Set Macro Response Options: ACKS:No, DATA:No, ERRS:No, NTAGS:No, STRINGS:No
← Tag Read (Length:2, Address:0, Timeout:2000ms)
◇ IF last read (Length:2, Offset:0) = {01 02} then BRANCH to "PASSED"
◇ IF last read (Length:2, Offset:0) = {03 04} then BRANCH to "FAILED"
● END
● #: PASSED
123 Transmit Custom String ("Batch Passed")
● END
● #: FAILED
123 Transmit Custom String ("Batch Failed")
● END
    
```


7 宏专用命令

传输用户定义
字符串

示例 2

下面的示例只是一个大型宏的一段，其中插入了许多 *传输用户定义字符串* 作为调试辅助工具来跟踪宏的流程。

```

● #: Start
// This is an example macro, used to show some simple logic and hub capabilities.
// The first 6 bytes of a tag are read, and the data compared to 5 different strings
// Based on the text string, a s different combination of hub outputs is set.
//
// Enable all macro output here (this is the default, so we really dont need to do this)
Set Macro Response Options: ACKS:Yes, DATA:Yes, ERRS:Yes, NTAGS:Yes, STRINGS:Yes
// Set Maco to continue even if an error occurs (such as tag not found)
On Error, Continue Macro
// Here is where we read the first 6 bytes of the tag
Tag Read (Length:6, Address:0, Timeout:2000ms)
Transmit Custom String ("About to read tag")
// First make sure we really have a tag. If not, go to NO_TAG
IF last RF command caused a "Tag Not Found" error then BRANCH to "NO_TAG"
// Here is where we start the comparisons and branch
// to the different sections based on the tag data
IF last read (Length:3, Offset:0) = "RED" then BRANCH to "ON_RED"
IF last read (Length:4, Offset:0) = "BLUE" then BRANCH to "ON_BLUE"
IF last read (Length:5, Offset:0) = "GREEN" then BRANCH to "ON_GREEN"
IF last read (Length:6, Offset:0) = "YELLOW" then BRANCH to "ON_YELLOW"
IF last read (Length:5, Offset:0) = "BLACK" then BRANCH to "ON_BLACK"
// Tag data didnt match any of those five colors, so lets clear all outputs, and END
Clear Outputs {Hub Output A=LOW, Hub Output B=LOW, Hub Output C=LOW, Hub Output D=LOW}
END
● #: ON_RED
// "Red" tag = Output 1
Clear Outputs {Hub Output B=LOW, Hub Output C=LOW, Hub Output D=LOW}
Set Outputs {Hub Output A=HIGH}
END
    
```

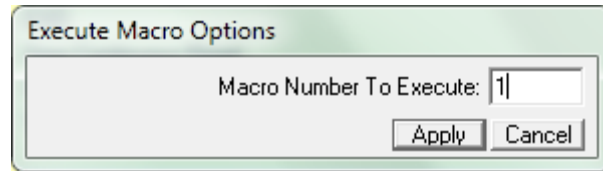
7.6 执行宏

选项	值
要执行的宏编号	1-8

此命令使 *RFID 控制器* 结束当前宏并立即开始执行另一个宏。当该宏完成时，控制不会返回到当前宏。此命令允许将宏链接在一起以形成更大的程序。

7 宏专用命令

执行宏 将项目插入宏时，会出现以下对话框：



使用此命令可以执行从 1 到 8 的任意宏。



注意

编译后的宏没有与之关联的内部宏编号 – 它们可以加载到 BIS M-6xx 和 BIS U-6xx 系列 RFID 控制器中可用的八个宏隙中的任意一个中。因此，由用户负责跟踪各个宏，并事先知道（例如）每个宏编号将加载什么。



注意

应注意不要使宏进入无限循环，除非这是需要的特性。



注意

如果执行的宏不存在，宏终止将结束。

示例 2

在以下示例中，读取标签的前 6 个字节。如果前 3 个字节为“RED”，则执行宏 5。如果前 4 个字节为“BLUE”，则执行宏 6。否则执行宏 7。

```
● #: Start
✂ Disable Macro Responses (Except Errors)
← Tag Read (Length:6, Address:0, Timeout:2000ms)
◇ IF last read (Length:3, Offset:0) = "RED" then BRANCH to "ON_RED"
◇ IF last read (Length:4, Offset:0) = "BLUE" then BRANCH to "ON_BLUE"
⌘ Execute Macro 7 (Current Macro Ends)
● END
● #: ON_RED
⌘ Execute Macro 5 (Current Macro Ends)
● END
● #: ON_BLUE
⌘ Execute Macro 6 (Current Macro Ends)
● END
```

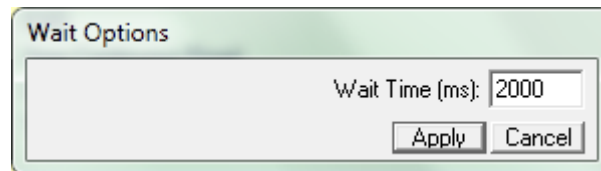
7 宏专用命令

7.7 等待

选项	值
等待时间（以毫秒为单位）	0-65535

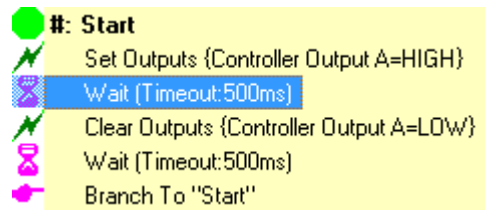
此命令会导致宏执行流程暂停指定的持续时间。它可用于需要延迟的情况。

将项目插入宏时，会出现以下对话框：



示例 1：

在以下示例中，控制器输出 A 在无限循环中设置为高 500 ms，然后设置为低 500 ms。

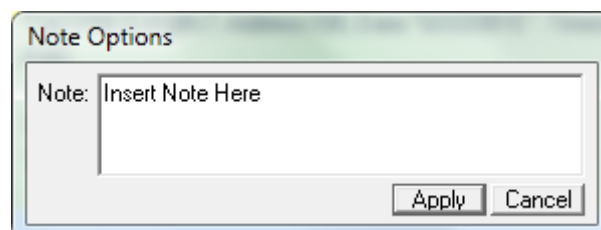


7.8 注意

选项	值
注意	0-255 个字符的文本

注意命令实际上只是一个在宏中放置文本符号的地方，用于注释或其他信息性目的。它们不会影响宏执行的流程。它们可以包括程序逻辑的描述，或版本号，或用户认为应该注意的任何内容。

将项目插入宏时，会出现以下对话框：



7 宏专用命令

7.9 等待

每个字符串的长度最多可达 255 个字节。它们不占用宏空间。

示例 1:

以下示例显示了一个大型宏的片段，其中插入了许多注意，用于描述宏的流程并提供其他信息。

```

#: Start
// This is an example macro, used to show some simple logic and hub capabilities.
// The first 6 bytes of a tag are read, and the data compared to 5 different strings
// Based on the text string, a s different combination of hub outputs is set.
//
// Enable all macro output here (this is the default, so we really dont need to do this)
Set Macro Response Options: ACKS:Yes, DATA:Yes, ERRS:Yes, NTAGS:Yes, STRINGS:Yes
// Set Maco to continue even if an error occurs (such as tag not found)
On Error, Continue Macro
// Here is where we read the first 6 bytes of the tag
Tag Read (Length:6, Address:0, Timeout:2000ms)
// First make sure we really have a tag. If not, go to NO_TAG
IF last RF command caused a "Tag Not Found" error then BRANCH to "NO_TAG"
// Here is where we start the comparisons and branch
// to the different sections based on the tag data
IF last read (Length:3, Offset:0) = "RED" then BRANCH to "ON_RED"
IF last read (Length:4, Offset:0) = "BLUE" then BRANCH to "ON_BLUE"
IF last read (Length:5, Offset:0) = "GREEN" then BRANCH to "ON_GREEN"
IF last read (Length:6, Offset:0) = "YELLOW" then BRANCH to "ON_YELLOW"
IF last read (Length:5, Offset:0) = "BLACK" then BRANCH to "ON_BLACK"
// Tag data didnt match any of those five colors, so lets clear all outputs, and END
Clear Outputs {Hub Output A=LOW, Hub Output B=LOW, Hub Output C=LOW, Hub Output D=LOW}
END
#: ON_RED
    
```

7.10 结尾

选项	值
无	

此命令使宏立即终止。此命令应插入在宏中要停止宏执行的任意点。

无论您是否手动插入了一个宏，宏编译器会自动在每个宏的末尾插入一个隐含的结束。不过，在所有宏的末尾加上一个结束可能是一种很好的编程方法。

由于此命令没有可编辑项目，因此当项目插入宏时，不会出现对话框。

7 宏专用命令

结尾

示例 1:

在以下示例中，读取标签的前 6 个字节。如果前 3 个字节为“RED”，则设置 *Hub* 输出 A，宏结束。如果前 4 个字节为“BLUE”，则设置 *Hub* 输出 B，宏结束。否则设置 *Hub* 输出 C，宏结束。

```

● #: Start
⚡ Set Macro Response Options: ACKS:No, DATA:No, ERRS:No, NTAGS:No, STRINGS:No
← Tag Read (Length:6, Address:0, Timeout:2000ms)
◇ IF last read (Length:3, Offset:0) = "RED" then BRANCH to "ON_RED"
◇ IF last read (Length:4, Offset:0) = "BLUE" then BRANCH to "ON_BLUE"
⚡ Set Outputs {Hub Output C=HIGH}
● END
● #: ON_RED
⚡ Set Outputs {Hub Output A=HIGH}
● END
● #: ON_BLUE
⚡ Set Outputs {Hub Output B=HIGH}
● END

```

示例 2:

以下示例显示了一个常见的编程错误：在此示例中，如果标签的前几个字节与“RED”或“BLUE”不匹配，则设置 *Hub* 输出 C。但是，设置 *Hub* 输出 C 后宏执行不会停止，宏将继续执行，越过“ON_RED”标记，并设置 *Hub* 输出 A，然后结束。在希望宏停止的每个点插入 *结束* 命令非常重要。

```

● #: Start
⚡ Set Macro Response Options: ACKS:No, DATA:No, ERRS:No, NTAGS:No, STRINGS:No
← Tag Read (Length:6, Address:0, Timeout:2000ms)
◇ IF last read (Length:3, Offset:0) = "RED" then BRANCH to "ON_RED"
◇ IF last read (Length:4, Offset:0) = "BLUE" then BRANCH to "ON_BLUE"
⚡ Set Outputs {Hub Output C=HIGH}
● #: ON_RED
⚡ Set Outputs {Hub Output A=HIGH}
● END
● #: ON_BLUE
⚡ Set Outputs {Hub Output B=HIGH}
● END

```

附录

ASCII 表

十进制	Hex	控制代码	ASCII	十进制	Hex	ASCII	十进制	Hex	ASCII
0	00	Ctrl @	NUL	43	2B	+	86	56	V
1	01	Ctrl A	SOH	44	2C	,	87	57	W
2	02	Ctrl B	STX	45	2D	-	88	58	X
3	03	Ctrl C	ETX	46	2E	.	89	59	Y
4	04	Ctrl D	EOT	47	2F	/	90	5A	Z
5	05	Ctrl E	ENQ	48	30	0	91	5B	[
6	06	Ctrl F	ACK	49	31	1	92	5C	\
7	07	Ctrl G	BEL	50	32	2	93	5D]
8	08	Ctrl H	BS	51	33	3	94	5E	^
9	09	Ctrl I	HT	52	34	4	95	5F	_
10	0A	Ctrl J	LF	53	35	5	96	60	`
11	0B	Ctrl K	VT	54	36	6	97	61	a
12	0C	Ctrl L	FF	55	37	7	98	62	b
13	0D	Ctrl M	CR	56	38	8	99	63	c
14	0E	Ctrl N	SO	57	39	9	100	64	d
15	0F	Ctrl O	SI	58	3A	:	101	65	e
16	10	Ctrl P	DLE	59	3B	;	102	66	f
17	11	Ctrl Q	DC1	60	3C	<	103	67	g
18	12	Ctrl R	DC2	61	3D	=	104	68	h
19	13	Ctrl S	DC3	62	3E	>	105	69	i
20	14	Ctrl T	DC4	63	3F	?	106	6A	j
21	15	Ctrl U	NAK	64	40	@	107	6B	k
22	16	Ctrl V	SYN	65	41	A	108	6C	l
23	17	Ctrl W	ETB	66	42	B	109	6D	m
24	18	Ctrl X	CAN	67	43	C	110	6E	n
25	19	Ctrl Y	EM	68	44	D	111	6F	o
26	1A	Ctrl Z	SUB	69	45	E	112	70	p
27	1B	Ctrl [ESC	70	46	F	113	71	q
28	1C	Ctrl \	FS	71	47	G	114	72	r
29	1D	Ctrl]	GS	72	48	H	115	73	s
30	1E	Ctrl ^	RS	73	49	I	116	74	t
31	1F	Ctrl _	US	74	4A	J	117	75	u
32	20		SP	75	4B	K	118	76	v
33	21		!	76	4C	L	119	77	w
34	22		"	77	4D	M	120	78	x
35	23		#	78	4E	N	121	79	y
36	24		\$	79	4F	O	122	7A	z
37	25		%	80	50	P	123	7B	{
38	26		&	81	51	Q	124	7C	
39	27		'	82	52	R	125	7D	}
40	28		(83	53	S	126	7E	~
41	29)	84	54	T	127	7F	DEL
42	2A		*	85	55	U			

 **www.balluff.com**

巴鲁夫自动化（上海）有限公司
上海市浦东新区成山路 800 号
云顶国际商业广场 A 座 8 层
热线电话：400 820 0016
传真：400 920 2622
邮箱：sales.sh@balluff.com.cn