



**NLS-EM22**

**条码识读引擎**

**用户手册**

## 免责声明

请您在使用本手册描述的产品前仔细阅读手册的所有内容，以保障产品的安全有效地使用。阅读后请将本手册妥善保存以备下次使用时查询。

请勿自行拆卸终端或撕毁终端上的封标，否则福建新大陆自动识别技术有限公司不承担保修或更换终端的责任。

本手册中的图片仅供参考，如有个别图片与实际产品不符，请以实际产品为准。对于本产品的改良更新，新大陆自动识别技术有限公司保留随时修改文档而不另行通知的权利。

本手册包含的所有信息受版权的保护，福建新大陆自动识别技术有限公司保留所有权利，未经书面许可，任何单位及个人不得以任何方式或理由对本文档全部或部分内容进行任何形式的摘抄、复制或与其它产品捆绑使用、销售。

本手册中描述的产品中可能包括福建新大陆自动识别技术有限公司或第三方享有版权的软件，除非获得相关权利人的许可，否则任何单位或者个人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反相工程、出租、转让、分许可以及其它侵犯软件版权的行为。

福建新大陆自动识别技术有限公司对本声明拥有最终解释权。

## 版本记录

版本号	版本描述	发布日期
V1.0.0	初始版本。	2020.07.14

# 目 录

版本记录.....	- 3 -
前言.....	1
简介.....	1
章节纲要.....	1
手册图例.....	1
第一章 关于 NLS-EM22.....	2
第二章 配套工具.....	3
EasySet.....	3
第三章 系统设置.....	4
简介.....	4
设置码.....	4
设置命令.....	4
EasySet 设置.....	4
设置标识.....	5
使用设置码.....	6
设置码信息.....	6
LED 灯设置.....	7
解码成功 LED 灯设置.....	7
解码成功 LED 灯持续时间设置.....	8
解码成功 LED 灯颜色.....	9
提示音.....	10
开机提示音.....	10
解码成功声音设置.....	10
解码成功声音持续时间设置.....	11
解码成功声音频率设置.....	12
解码成功声音音量设置.....	13
识读模式.....	14
一次读码超时.....	15
延迟设置.....	16
重读延迟.....	16
重读延迟时间.....	17
重读超时复位.....	17
读码成功延迟.....	18
读码成功延迟时间.....	18
解码超时.....	19
响提示音.....	20
串行触发指令.....	21
修改启动读码指令.....	21
修改停止读码指令.....	22
识读偏好.....	22
禁止/允许读码.....	22
解码中心区域.....	23
设置中心区域.....	23
图像翻转.....	26

传送读码未成功信息.....	27
修改读码未成功信息（NGR 信息）.....	27
默认设置.....	28
出厂默认设置.....	28
用户默认设置.....	28
产品信息查询.....	29
查询系统信息.....	29
查询产品名称.....	30
查询固件版本号.....	30
查询解码器版本号.....	30
查询数据格式编辑版本号.....	31
查询硬件信息.....	31
查询产品序列号.....	31
查询产品生产日期.....	32
查询产品 OEM 序列号.....	32
<b>第四章 串口通讯设置.....</b>	<b>33</b>
RS-232 通讯（4-PIN 表贴杜邦插座）.....	33
TTL-232 通讯（12-PIN 表贴插座）.....	33
波特率.....	33
奇偶校验字符.....	35
数据位传输.....	36
停止位.....	36
硬件自动流控.....	37
<b>第五章 USB 通讯设置.....</b>	<b>38</b>
简介.....	38
USB 键盘.....	39
国家键盘布局.....	40
未知字符提示音.....	44
键盘仿真输入字符.....	45
控制字符输出.....	49
控制字符对应表.....	50
控制字符对应表（续）.....	51
按键延迟.....	52
大写锁定设置.....	53
大小写转换.....	54
模拟数字小键盘.....	55
快速模式.....	57
轮询速度.....	58
USB CDC 串口.....	60
HID POS (POS HID Bar Code Scanner).....	60
简介.....	60
软件编程访问设备的方法.....	60
获取扫描数据.....	61
发送数据到设备.....	61
IBM SurePOS (Table-Top).....	62
IBM SurePOS (Hand-Held).....	62
VID 和 PID 表.....	62
<b>第六章 条码参数设置.....</b>	<b>63</b>
简介.....	63

综合设置.....	63
允许所有条码 .....	63
禁止所有条码 .....	63
允许所有一维码.....	64
禁止所有一维码.....	64
允许所有二维条码 .....	64
禁止所有二维条码 .....	64
一维条码双码 .....	65
Code 128 .....	66
恢复出厂默认 .....	66
使能/禁止识读 .....	66
设置读码长度 .....	67
QR Code .....	68
恢复出厂默认 .....	68
使能/禁止识读 .....	68
设置读码长度 .....	69
QR 双码 .....	70
QR 反相 .....	71
字符编码方式 .....	71
ECI 输出.....	72
<b>第七章 数据格式编辑.....</b>	<b>73</b>
简介.....	73
开启/关闭数据格式编辑 .....	74
添加数据格式.....	76
清除数据格式.....	78
选择数据格式.....	79
单次使用数据格式 .....	80
数据格式不匹配错误提示音 .....	80
查询数据格式.....	81
数据格式编辑命令 .....	82
发送命令.....	82
移动命令.....	85
搜索命令.....	86
其他命令.....	89
<b>第八章 前后缀设置.....</b>	<b>94</b>
简介.....	94
综合设置.....	95
所有前后缀 .....	95
前缀顺序.....	95
自定义前缀 .....	96
修改自定义前缀.....	96
AIM ID 前缀 .....	97
Code ID 前缀 .....	98
默认 Code ID.....	98
修改 Code ID.....	99
自定义后缀 .....	101
修改自定义后缀.....	101
数据打包.....	102
打包格式.....	103
结束符后缀 .....	104

修改结束符后缀.....	104
<b>第九章 批处理设置.....</b>	<b>105</b>
简介.....	105
生成批处理指令.....	106
制作批处理条码.....	107
使用批处理条码.....	108
<b>附录.....</b>	<b>109</b>
数据码.....	109
保存或取消.....	112
默认设置表.....	113
AIM ID 列表.....	116
Code ID 列表.....	117
条码序号对照表.....	118
ASCII 码表.....	119
键盘按键序号.....	123





# 前言

## 简介

本手册主要向用户介绍 NLS-EM22 的使用方法。

## 章节纲要

《第一章 关于 EM22》	
《第二章 配套工具》	介绍配套工具 EasySet 的功能
《第三章 系统设置》	介绍扫描器的主要设置方法以及系统参数的设置
《第四章 RS-232 通讯设置》	描述 RS-232 通讯参数的设置
《第五章 USB 通讯设置》	描述 USB 通讯参数的设置
《第六章 条码参数设置》	列出 EM22 支持识读的所有码制并提供了相关的参数设置码
《第七章 数据格式编辑》	介绍如何使用数据格式编辑功能自定义格式输出条码信息
《第八章 前后缀设置》	介绍如何利用前、后缀来满足用户编辑条码信息的需求
《第九章 批处理设置》	介绍如何将多项设置操作制作成一个批处理设置码
《附录》	提供常用设置码和出厂默认参数表等

## 手册图例



辅助工具，方便用户使用文档



注意提示，提示用户需要强烈注意此处的内容



小提示，帮助用户更好的理解文档内容



示例，帮助用户熟悉操作

## 第一章 关于 NLS-EM22

NLS-EM22 二维影像式嵌入应用条码识读引擎，应用了国际领先的芯片化新大陆 **UING** 智能图像识别技术，开创影像式二维条码识读引擎的新时代。

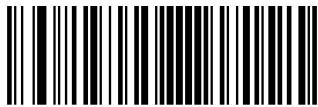
新大陆的二维解码芯片，将先进的 **UING** 图像识别算法与先进的芯片设计与制造技术完美融合，极其简化了二维条码识读产品的设计难度，树立二维影像产品高性能、高可靠、低功耗的优秀标杆。

## 第二章 配套工具

### EasySet

EasySet 是一款在 Windows 操作系统下运行的，由新大陆自动识别公司自主开发的设备条码、通讯等参数的配置软件。它可以通过 EasySet 图形界面设置或查询设备配置，也可以通过指令的方式和设备直接交互。





#SETUPE1  
【启动设置】

## 第三章 系统设置

### 简介

有三种方法可以对扫描器进行设置：

#### 设置码

扫描器通过识读一系列特殊条码来设置选项和功能。在下面的章节里，我们会详细介绍可供设置的选项和功能并提供对应的设置码。

这种设置识读的方法比较直接，由于需要手动识读每个设置码，因而容易发生误设置。

#### 设置命令

主机可以发送设置命令字符串对扫描器进行设置。在下面的章节里，除了介绍设置码，我们也将介绍设置命令字符串。

利用设置命令对扫描器进行设置是可以自动化进行的。用户可以开发一套软件，将所有相关的设置数据都载入扫描器中。

#### EasySet 设置

EasySet 是一款运行在 Windows 系统下的图像用户界面程序，为新大陆自动识别公司自主研发，用于条码识读及处理。用 EasySet 可以查看解码后的条码信息及扫描器摄取的图像，还可以很方便地对扫描器进行设置。

这种设置方法与设置命令很相似。EasySet 是专为新大陆自动识别公司的产品而设计的。

**提示：**除了一些临时性的设置会在设备重启或断电后消失，其他功能设置信息将储存在扫描器中，不会因为关机而丢失。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 设置标识



这是禁用设置码功能的标识。

该标识由四个部分组成：

1. 设置码的条码部分。
2. 与设置码相对应的设置命令字符串。
3. 设置的选项或者功能的名称，如退出设置功能。
4. \*\*表示该项设置为默认设置。



#SETUPE0

【退出设置】

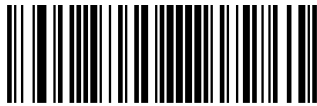


#SETUPE1  
【启动设置】

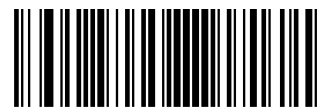
## 使用设置码

读取“启动设置”条码来激活设置码功能。可以通过读取设置码来对识读引擎进行设置。

要退出设置码功能，只要读取“退出设置”条码或设置码外的条码即可。



@SETUPE0  
\*\*【退出设置】



@SETUPE1  
【启动设置】

## 设置码信息

设置码信息可以被发送给主机。出厂默认设置是“不发送设置码信息”，此时设置码信息不会被发送给主机；通过识读“发送设置码信息”的条码，识读引擎将会把设置码信息发送给主机。



@SETUPT0  
\*\*【不发送】



@SETUPT1  
【发送】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---

## LED 灯设置

解码成功 LED 灯设置



@GRLENA1

\*\*【开启】



@GRLENA0

【关闭】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 解码成功 LED 灯持续时间设置

可选择 20ms、120ms、220ms、320ms 的快速设定。



@GRLDUR20

【短 (20ms)】



@GRLDUR120

【中 (120ms)】



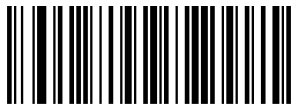
@GRLDUR220

\*\*【长 (220ms)】



@GRLDUR320

【特长 (320ms)】



@GRLDUR

【解码成功 LED 灯持续时间自定义设置

(范围 1-2500ms)】

**E**  
*sample*

设置解码成功 LED 灯持续时间为 800ms，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“解码成功 LED 灯持续时间自定义设置”
3. 数据码“8”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”



#SETUPE0

【退出设置】





#SETUPE1

【启动设置】

## 解码成功 LED 灯颜色

- ◇ **红色：** 解码成功时，亮红色 LED 灯。
- ◇ **白色：** 解码成功时，亮白色 LED 灯。
- ◇ **绿色：** 解码成功时，亮绿色 LED 灯。
- ◇ **蓝色：** 解码成功时，亮蓝色 LED 灯。
- ◇ **条码支付模式：** 微信支付条码解码成功时，亮绿色 LED 灯；支付宝支付条码解码成功时，亮蓝色 LED 灯，其他类型条码解码成功时，亮红色 LED 灯。



@GRLCLR0

【红色】



@GRLCLR1

【白色】



@GRLCLR2

【绿色】



@GRLCLR3

【蓝色】



@GRLCLR4

\*\*【条码支付模式】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## 提示音

### 开机提示音



@PWBENA1  
\*\*【开启】



@PWBENA0  
【关闭】

### 解码成功声音设置

读取“关闭”可以禁止解码成功声音响起，读取“开启”即可恢复解码成功声音提示。



@GRBENA1  
\*\*【开启】



@GRBENA0  
【关闭】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 解码成功声音持续时间设置

可以选择 40ms、80ms、120ms 的快速设定。



@GRBDUR40

【短 (40ms)】



@GRBDUR80

\*\*【中 (80ms)】



@GRBDUR120

【长 (120ms)】



@GRBDUR

【解码成功声音持续时间自定义设置

(范围 20-300ms)】

**E**

1. *sample*

设置解码成功声音持续时间为 200ms，可以按顺序识读以下条码来设置：

- 识读“解码成功声音持续时间自定义设置 (范围 20-300ms)”
- 数据码“2”，“0”，“0” (见附录-数据码)
- 读“保存” (见附录-保存或取消)
- 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 解码成功声音频率设置



@GRBFRQ800  
【最低(800Hz)】



@GRBFRQ1600  
【低(1600Hz)】



@GRBFRQ2730  
\*\*【中(2730Hz)】



@GRBFRQ4200  
【高(4200Hz)】



@GRBFRQ  
【解码成功声音频率自定义设置  
(范围: 20-20000Hz)】

**E**  
*example*

设置解码成功声音频率为 2000Hz，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“解码成功声音频率自定义设置”
3. 数据码“2”，“0”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 解码成功声音音量设置



@GRBVLL20

\*\*【大】



@GRBVLL8

【中】



@GRBVLL2

【小】



@GRBVLL

【解码成功声音音量自定义设置（范围：1-20）】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## 识读模式

- ◇ **电平触发模式：**按住触发键，启动读码；读码成功或松开触发键后，读码结束。
- ◇ **连续读码模式：**开机后一直处于读码状态。按下并松开按键可以让扫描器在读码状态和停止读码状态之间切换。在这个模式下，**重读延时**可以用来防止同一个条码被读到多次。识读设置码切换为该模式时，将会停止读码**3**秒钟，然后进入连续读码状态。



@SCNMOD0

【电平触发模式】



@SCNMOD3

\*\*【连续读码模式】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 一次读码超时

**一次读码超时：**扫描器处于读码状态的时间量。设置的时间范围是 0 到 3600000 毫秒，设置为 0 的时候，扫描器将一直处于读码状态。默认：3000 毫秒。



@ORTSET

【一次读码超时】

**E**  
*example*

设置一次读码超时时间为 1500 毫秒，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“一次读码超时”
3. 数据码“1”，“5”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## 延迟设置

### 重读延迟

- ◇ **开启**：如果扫描器读到一个条码并且在重读延迟时间内连续第二次读到这个条码，则第二次读到条码将会忽略，不会输出。
- ◇ **关闭**：重读延迟无效。扫描器在任何时候都可以连续识读同一个条码。

默认：重读延迟关闭。



@RRDENA1  
\*\*【开启】



@RRDENA0  
【关闭】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## 重读延迟时间

用于防止同一个条码被连续读到两次，重读延时设置的是允许连续识读到同一个类型和数据的条码的最小时间间隔。该延迟仅在自动读码模式和连续读码模式下使用，设置的时间范围是 1 到 3600000 毫秒，默认为 1500 毫秒。如果重读延迟时间设置大于 3000 毫秒，设置码的重读延迟会限制在 3000 毫秒。



@RRDDUR

【重读延迟时间】

**E**  
*example*

设置重读延迟时间为 1000 毫秒，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“重读延迟时间”
3. 数据码“1”，“0”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”

## 重读超时复位



@RRDREN1

\*\*【开启】



@RRDREN0

【关闭】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 读码成功延迟

开启读码成功延时，则读码成功后的一段时间（**读码成功延迟时间**）内扫描器识读任意条码均不会识读成功。默认：关闭。



@GRDNA1  
【开启】



@GRDNA0  
\*\* 【关闭】

## 读码成功延迟时间

读码成功后暂停读码的时间长度，设置的时间范围是 1 到 3600000 毫秒，默认为 500 毫秒。



【读码成功延迟时间】

**E**  
*example*

设置读码成功延迟时间为 1000 毫秒，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“读码成功延迟时间”
3. 数据码“1”，“0”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 解码超时

扫描器每次对图像解码运算的超时时间。设置的时间范围从 1 到 3000 毫秒，默认为 500 毫秒。



@DETSET

【解码超时时间】

**E**  
*xample*

设置解码超时时间为 1000 毫秒，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“解码超时时间”
3. 数据码“1”，“0”，“0”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 响提示音

通过发送**响提示音**指令可以控制扫描器发出提示音响声。提示音的频率和持续时间作为**响提示音**指令的参数需要配置。指令格式为 BEEPON xxxFyyyTnnV，其中：F 表示提示音的频率，取值范围为 1 ~ 20000Hz，xxxF 表示提示音的频率为 xxx Hz（例如 2700F 表示响声的频率为 2700Hz）；T 表示提示音的持续时间，取值范围为 1 ~ 10000ms，yyyT 表示提示音的持续时间为 yyy ms（例如，100T 表示响声持续时间为 100ms）；V 表示音量，取值范围为 1~20。

**E**  
*xample*

发出频率为 2000Hz 持续时间为 50ms 音量为 20V 的响声：

发送 ~<SOH>0000#BEEPON2000F50T20V;<ETX>

应答 <STX><SOH>0000#BEEPON2000F50T20V<ACK>;<ETX>。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 串行触发指令

识读模式设置为电平触发模式的时候，支持通过串行通讯命令“启动读码”命令触发读码，扫描器识读成功或者达到一次读码超时或者接收到“停止读码”命令后结束读码。

- ◇ 禁止：不使用串行触发指令。
- ◇ 使能：在电平触发读码模式下使用串行触发指令。



@SCNTCE0

\*\*【禁止】



@SCNTCE1

【使能】

## 修改启动读码指令

启动读码指令可以模拟按键按下。指令支持 1 个到 10 个字符，字符的取值范围为 1 到 0xFF。首字符不能为‘?’(即 0x3F)。

默认指令为：<SOH> T <EOT>



@SCNTCT

【修改启动读码指令】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 修改停止读码指令

停止读码指令可以模拟按键松开。指令支持 1 个到 10 个字符，字符的取值范围为 1 到 0xFF。首字符不能为 '?' (即 0x3F)。默认指令为: <SOH> P <EOT>

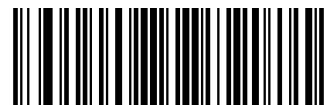


@SCNTCP  
【修改停止读码指令】

## 识读偏好



@EXPLVL0  
【普通】



@EXPLVL2  
【屏幕识读模式】



@EXPLVL5  
\*\*【条码支付模式】

## 禁止/允许读码

通过发送**禁止读码**和**允许读码**指令可以控制扫描器是否需要停止读码。默认为**允许识读**。

- ◇ **禁止读码**: 强制扫描器一直处于停止读码状态。扫描器重启或者收到允许读码指令，才会恢复为允许读码状态。禁止读码指令内容为: ~<SOH>0000#SCNENA0;<ETX>
- ◇ **允许读码**: 扫描器受配置的读码模式控制读码。允许读码指令内容为: ~<SOH>0000#SCNENA1;<ETX>



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 解码中心区域

- ◇ 全区域解码：选取拍摄的图像的全部区域（整幅图）用于解码，只输出读到的第一个条码。
- ◇ 中心区域解码：条码的中心位置必须位于设定的区域，才能成功读取。同时有多个条码处于设定的区域内的时候，只输出读取到的第一个条码（这种情况下，建议缩小中心区域的大小）。



@CADENA0

\*\*【全区域解码】



@CADENA1

【中心区域解码】

## 设置中心区域

区域的大小以图像宽度和高度的比例来设定的。需要设置中心区域顶部、底部、左侧、右侧范围，设置范围为 0 到 100，底端必须大于顶端，右端必须大于左端。默认：顶部为 40%，底部为 60%，左侧为 40%，右侧为 60%。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】



@CADTOP  
【中心区域顶部】



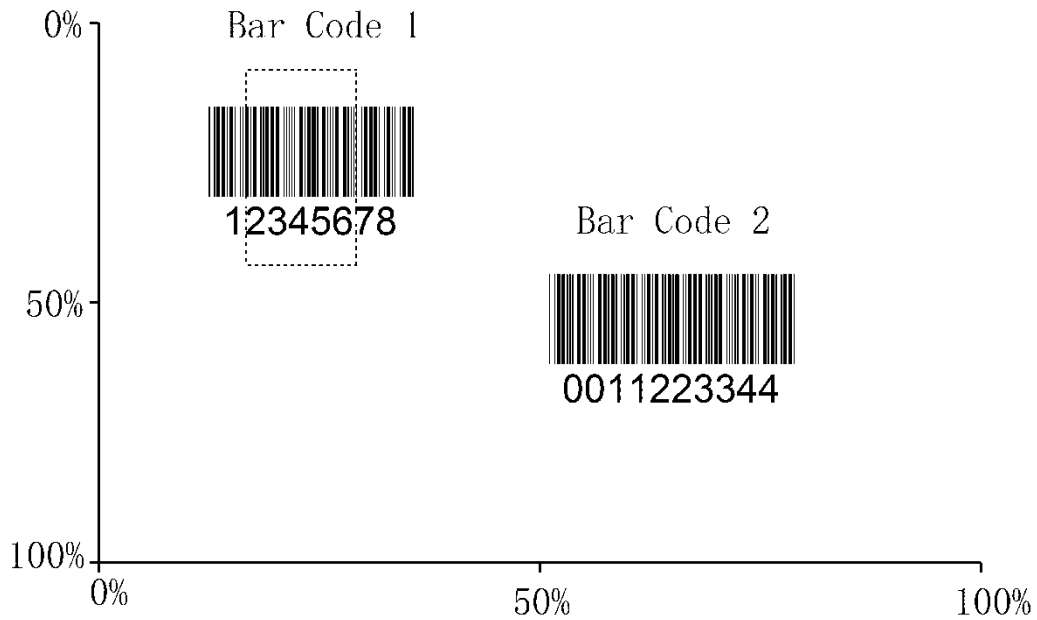
@CADLEF  
【中心区域左侧】



@CADBOT  
【中心区域底部】



@CADRIG  
【中心区域右侧】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

**E**  
*example*

设置只识读上图中的 Bar Code 1，中心区域的顶部为 10，底部为 45，左侧为 15，右侧为 30，可以按顺序识读以下条码来设置：

1. 读“启动设置”
2. 识读“中心区域顶部”
3. 数据码“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 识读“中心区域底部”
6. 数据码“4”，“5”（见附录-数据码）
7. 读“保存”（见附录-保存或取消）
8. 识读“中心区域顶部”
9. 数据码“1”，“0”（见附录-数据码）
10. 读“保存”（见附录-保存或取消）
11. 识读“中心区域左侧”
12. 数据码“0”（见附录-数据码）
13. 读“保存”（见附录-保存或取消）
14. 识读“中心区域右侧”
15. 数据码“3”，“0”（见附录-数据码）
16. 读“保存”（见附录-保存或取消）
17. 识读“中心区域左侧”
18. 数据码“1”，“5”（见附录-数据码）
19. 读“保存”（见附录-保存或取消）
20. 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## 图像翻转

图像支持水平翻转、垂直镜像输出功能。通过 EasySet 可以获取扫描器拍摄的图像。



@MIRROR0

\*\*【正常图像】



@MIRROR2

【垂直翻转】



@MIRROR1

【水平翻转】



@MIRROR3

【水平、垂直翻转】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 传送读码未成功信息

- ◇ 使能：读码不成功时，按键释放或者读码超时后发送读码未成功信息。
- ◇ 禁止：读码不成功时，不会发送读码未成功信息。



@NGRENA0

\*\*【禁止】



@NGRENA1

【使能】

## 修改读码未成功信息（NGR 信息）

读码未成功（NGR）信息支持 1 到 7 个字符，字符的取值范围为 0 到 0xff，默认：**4E47**。



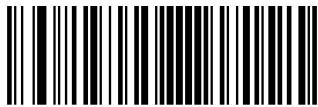
@NGRSET

【修改读码未成功信息（NGR 信息）】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

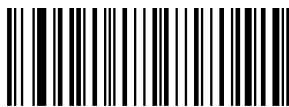
## 默认设置

### 出厂默认设置

所有扫描器都有一个出厂的默认设置，读取“加载出厂默认设置”条码，将使扫描器的所有属性恢复成出厂状态。

在以下情况下您最有可能使用到此条码：

- ◇ 扫描器设置出错，如无法识读条码。
- ◇ 您忘记了之前对扫描器做过何种设置，而又不希望受之前的设置影响。



@FACDEF  
\*\*【加载出厂默认设置】

### 用户默认设置

除了出厂的默认设置外，您可以把您经常使用的设置存成用户默认设置。

用户默认设置也包含扫描器的所有属性设置，并且用户默认设置将被保存下来不会丢失，除非重新将当前设置存为用户默认设置。读取“保存用户默认设置”将保存当前设置为用户默认设置，并且覆盖掉之前设置过的用户默认设置。读取“加载用户默认设置”将使扫描器切换到用户默认设置的状态。



@CUSSAV  
【保存用户默认设置】



@CUSDEF  
【加载用户默认设置】



读取“加载出厂默认设置”条码，扫描器中保存的用户默认设置不会被删除。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 产品信息查询

### 查询系统信息

您可以通过识读“查询系统信息”设置码来获得产品的相关信息。读此设置码后，扫描器会立即传送产品信息给主机。



@QRYSYS

【查询系统信息】

### 系统信息内容：

名称	描述
Product Name	产品名称
Firmware Version	固件版本号
Decoder Version	解码器版本号
Hardware Version	硬件版本
Serial Number	产品序列号
OEM Serial Number	产品 OEM 序列号 (ESN)
Manufacturing Date	产品生产日期



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## 查询产品名称

您可以通过识读“查询产品名称”来获得产品的名称信息。读此设置码后，扫描器会立即传送产品名称给主机。



@QRYPDN  
【查询产品名称】

## 查询固件版本号

您可以通过识读“查询固件版本号”来获得设备固件版本号。读此设置码后，扫描器会立即传送固件版本号给主机。



@QRYFWW  
【查询固件版本号】

## 查询解码器版本号

您可以通过识读“查询解码器版本号”来获得产品的解码库版本号。读此设置码后，扫描器会立即传送解码库版本号给主机。



@QRYDCV  
【查询解码器版本号】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 查询数据格式编辑版本号

您可以通过识读“查询数据格式编辑版本号”来获得产品的数据格式编辑功能的版本号。读此设置码后，扫描器会立即传送数据格式编辑功能的版本号给主机。



@QRYDFM

【查询数据格式编辑版本号】

## 查询硬件信息

您可以通过识读“查询硬件信息”来获得产品的硬件版本信息。读此设置码后，扫描器会立即传送硬件信息给主机。



@QRYHWW

【查询硬件信息】

## 查询产品序列号

您可以通过识读“查询产品序列号”来获得产品的序列号信息。读此设置码后，扫描器会立即传送产品序列号给主机。



@QRYPSN

【查询产品序列号】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## 查询产品生产日期

您可以通过识读“查询产品生产日期”来获得产品的生产日期。读此设置码后，扫描器会立即传送产品生产日期给主机。



@QRYDAT

【查询产品生产日期】

## 查询产品 OEM 序列号

您可以通过识读“查询产品 OEM 序列号”来获得产品 OEM 序列号信息。读此设置码后，扫描器会立即传送产品 OEM 序列号给主机。



@QRYESN

【查询产品 OEM 序列号】



#SETUPE0

【退出设置】





#SETUPE1

【启动设置】

## 第四章 串口通讯设置

### RS-232 通讯（4-PIN 表贴杜邦插座）

可以使用 RS-232 通讯，也叫做串口通讯方式。当扫描器与主机使用串口线连接时，双方需要设置相同的通讯参数以保证通讯的正常进行，需要设置的参数包含通讯的波特率（即传输速率），校验字符设置，数据位设置，停止位设置。



@INTERFO

RS-232

### TTL-232 通讯（12-PIN 表贴插座）

可以使 TTL-232 通讯，也叫做串口通讯方式。当扫描器与主机使用串口线连接时，双方需要设置相同的通讯参数以保证通讯的正常进行，需要设置的参数包含通讯的波特率（即传输速率），校验字符设置，数据位设置，停止位设置。

### 波特率

波特率是串口数据通讯是每秒传输的位数，扫描器和数据接收主机所使用的波特率须保持一致才能保证数据传输的准确。扫描器支持以下列出的波特率，单位是 bit/s。默认：9600bps



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---



@232BAD8

【115200】



@232BAD6

【38400】



@232BAD4

【14400】



@232BAD2

【4800】



@232BAD0

【1200】



@232BAD7

【57600】



@232BAD5

【19200】



@232BAD3

\*\*【9600】



@232BAD1

【2400】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 奇偶校验字符

扫描器在使用串口传输过程中可以选用不同的奇偶校验字符类型，但必须和主机的奇偶校验字符类型一致。

- ◇ 选择奇校验，若传输的数据中“1”的个数为奇数，则校验字符为0。
- ◇ 选择偶校验，若传输的数据中“1”的个数为偶数，则校验字符为0。
- ◇ 选择无校验，不发送奇偶校验字符。



@232PAR0

\*\*【无校验】



@232PAR1

【偶校验】



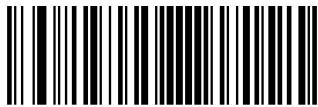
@232PAR2

【奇校验】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 数据位传输

可选择传输 7, 8 位数据, 请务必确保扫描器的数据位和数据接收主机的一致。



@232DAT1  
【7 个数据位】



@232DAT0  
\*\* 【8 个数据位】

## 停止位

停止位位于传输的每个字节的最后部分, 用来标志此字节传输完成可以开始接收下一字节数据。

默认设置 1 个停止位。如果需要停止较长时间, 可以设置 2 个停止位。



@232STP0  
\*\* 【一个停止位】



@232STP1  
【两个停止位】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 硬件自动流控

当使能这个设置的时候，扫描器会根据 CTS 信号的电平来判断是否可以发送数据，当 CTS 信号为低电平的时候表示接收端（PC 等）的串口缓存已经满了，此时阅读器不会再发送串口数据，直到 CTS 信号被接收端（PC 等）设置为高电平。当阅读器的串口接收未准备好的时候会把 RTS 设置为低电平，发送端（PC 等）检测到该信号为低电平的时候，不可以再发送数据给阅读器，否则数据将会丢失。

如果禁止硬件自动流控，则串口数据的收发不受 RTS/CTS 信号影响。



@232AFL0

\*\*【关闭】



@232AFL1

【开启】



如果需要使用硬件自动流控，请确保使用的串口通讯线缆中包含 RTS/CTS 信号线。如果串口通讯线缆不含 RTS/CTS 信号线，开启硬件自动流控将会导致串口通讯故障。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 第五章 USB 通讯设置

### 简介

当您使用 USB 方式连接扫描器和主机时，有以下四种连接方式可供选择，同时可根据客户实际需要进行默认方式设置：

- ◇ **USB 键盘：**该方式将扫描器输入虚拟成 USB 键盘输入，无需通过 USB 接口进行命令设置，且条码数据中的数据可以用键盘按键直接输入，无需驱动，主机端也可以很方便地取得数据。
- ◇ **USB CDC 串口：**符合 USB 组织定义的 CDC 规范的接口，主机端虚拟成串口，主机端操作该串口与操作物理串口行为一致。需要在主机安装驱动。
- ◇ **HID POS (POS HID Bar Code Scanner)：**该方式基于 HID 接口，不需要自定义驱动，并且比模拟键盘接口和传统的 RS-232 串口的通讯速度快。
- ◇ **IBM SurePOS：**符合 IBM（现为 Toshiba Global Commerce Solutions）4698 扫描器接口规范的 USB 接口。

当同时使用 USB 及 RS-232 方式连接扫描器和主机时，由于 USB 连接方式具有更高的优先级，因此扫描器会默认选择使用 USB 连接方式。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## USB 键盘

USB 数据线连接状况下，可以将扫描器设置成 HID Keyboard 输入模式。在这种模式下，扫描器将成为一个虚拟键盘，数据接收主机像接受真实键盘输入一样接受此虚拟键盘的输入。扫描器解码得到数据后的发送过程便是敲击虚拟键盘中与数据对应的每一个按键。



@INTERF3

【USB 键盘】



若主机的输入框可以接受键盘输入，则扫描器采用这种通讯方式可以不需要其他任何辅助程序，直接将解码后的数据输入到主机的输入框中。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## 国家键盘布局

不同国家语言对应的键盘键位排布，符号等不尽相同。因此，扫描器可以根据需要虚拟成不同国家的键盘制式。默认为美国制式的键盘。



@KBWCTY0  
\*\*【美国英语】



@KBWCTY1  
【比利时】



@KBWCTY2  
【巴西】



@KBWCTY3  
【加拿大(法语)】



@KBWCTY4  
【捷克斯洛伐克】



@KBWCTY5  
【丹麦】



@KBWCTY6  
【芬兰(瑞典语)】



#SETUPE0  
【退出设置】





#SETUPE1

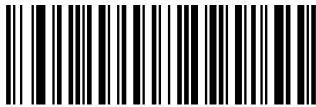
【启动设置】

---



@KBWCTY7

【法国】



@KBWCTY8

【德国/奥地利】



@KBWCTY9

【希腊】



@KBWCTY10

【匈牙利】



@KBWCTY11

【以色列(希伯来语)】



@KBWCTY12

【意大利】



@KBWCTY13

【拉丁美洲】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---



@KBWCTY14  
【荷兰 (荷兰语)】



@KBWCTY16  
【波兰】



@KBWCTY18  
【罗马尼亚】



@KBWCTY21  
【斯洛伐克】



@KBWCTY15  
【挪威】



@KBWCTY17  
【葡萄牙】



@KBWCTY19  
【俄罗斯】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---



@KBWCTY22

【西班牙】



@KBWCTY23

【瑞典】



@KBWCTY24

【瑞士(德语)】



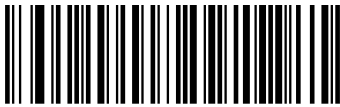
@KBWCTY25

【土耳其 F】



@KBWCTY26

【土耳其 Q】



@KBWCTY27

【英国】



@KBWCTY28

【日本】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 未知字符提示音

由于键盘布局的差异，条形码数据中包含的某些字符在选定的键盘上可能不可用。因此，扫描器无法传输未知字符。扫描下面的相应条形码，以便在检测到未知字符时启用或禁用发出蜂鸣声。



@KBWBUC0  
\*\*【关闭】



@KBWBUC1  
【开启】

## Example

假设当前扫描器虚拟键盘制式对应语言为法语（编号为7），用此扫描器识读内容为“ADF”的条码。由于条码数据中的一个字符“D”（0xD0）不在法语键盘所有可用的按键中，因此传输时扫描器将跳过此未知字符继续处理下一字符。

从实际使用场合看，如果未知字符提示音设置为“关闭”，则扫描器传输时将无任何声音提示，传输到数据接收主机的数据为“AF”；

如果未知字符提示音设置为“开启”，则扫描器处理此字符时发出错误提示音，传输到数据接收主机的数据仍然为“AF”。



若开启了“键盘仿真输入字符”功能，则此功能无效。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 键盘仿真输入字符

当键盘仿真输入字符开启后，字符将通过数字小键盘发送，会忽略国家键盘布局设置。这个模式还需要设置 **Code Page 选择** 和 **Unicode 输出**。Code Page 决定了目标语言，Unicode 输出设置控制输入到主机的编码方式是 Unicode 还是 Code Page。



@KBWALTO

\*\*【关闭】



@KBWALT1

【开启】



位于 0x00~0x1F 之间的 ASCII 字符按“控制字符输出”功能设置的方式输出。



这种发送方式能确保任何字符都能被顺利传输，但因为每传送一个字符所需要模拟的按键过多，因此速度较慢。



假设当前扫描器 Code Page 选择为 Code Page 1252（拉丁，西欧），扫描器识读内容为“ADF”（Code Page 1252 中十进制值分别为 65/208/70）的条码。

如果键盘仿真输入字符设置为“开启”，Unicode 输出设置为“关闭”，则扫描器模拟键盘操作如下：

输入“A” -- ALT 键按下，同时顺序按动数字小键盘键 0，6，5，松开 ALT 键；

输入“D” -- ALT 键按下，同时顺序按动数字小键盘键 2，0，8，松开 ALT 键；

输入“F” -- ALT 键按下，同时顺序按动数字小键盘键 0，7，0，松开 ALT 键。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## Code Page

代码页（Code Page）定义字符代码到字符的映射。如果收到的数据没有显示正确的字符，可能是因为正在扫描的条形码是使用不同于主机程序期望的代码页来创建的。如果是这种情况，请扫描以下设置码，选择创建条形码的代码页（如果是 PDF417、QR Code、Aztec、Data Matrix 等条码，还需要在条码参数设置章节设置对应的字符编码方式）。设置后，条形码数据字符应该可以正确显示。“Code Page 选择”功能仅在键盘仿真输入字符功能开启后才生效。默认：Code Page 1252（拉丁，西欧）。



@KBWCPG0

\*\*【Code Page 1252（拉丁，西欧）】



@KBWCPG1

【Code Page 1251（西里尔文）】



@KBWCPG2

【Code Page 1250（中欧）】



@KBWCPG3

【Code Page 1253（希腊语）】



@KBWCPG4

【Code Page 1254（土耳其语）】



@KBWCPG5

【Code Page 1255（希伯来语）】



@KBWCPG6

【Code Page 1256（阿拉伯语）】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---



@KBWCPG7

【Code Page 1257 (波罗的海)】



@KBWCPG8

【Code Page 1258 (越南语)】



@KBWCPG9

【Code Page 936 (简体中文, GB2312, GBK)】



@KBWCPG10

【Code Page 950 (繁体中文, Big5)】



@KBWCPG12

【Code Page 932 (日语, Shift-JIS)】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

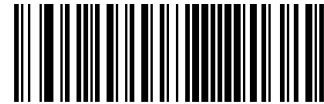
---

## Unicode 输出

不同的应用软件对于接收的字符编码也有要求，如 MS Office 的 Word 采用的是 Unicode 编码，需要将 Unicode 输出设置为“开启”；而 MS Office 的 Excel 或记事本采用的是 Code Page 编码，需要将 Unicodes 输出设置为“关闭”。“Unicode 输出”功能仅在键盘仿真输入字符功能开启后才生效。默认：关闭。



@KBWCPU0  
\*\*【关闭】



@KBWCPU1  
【开启】

## 前导“0”

开启“前导‘0’”功能，将通过数字键盘发送的字符序列发送为具有前导 0 的 ISO 字符。例如，ASCII “A” 以“ALT MAKE” 0065 “ALT BREAK” 的形式传输。此功能仅在“键盘仿真输入字符”启用时有效。



@KBWALZ0  
【关闭】



@KBWALZ1  
\*\*【开启】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## 控制字符输出

位于 0x00~0x1F 之间的 ASCII 字符可以被转义成为某个控制功能键。控制功能键的输入在虚拟键盘中的操作如下，具体的 ASCII 值与控制功能键的对应关系见下页附表。默认：关闭。



@KBWFKM0

\*\*【关闭】



@KBWFKM1

【Control + ASCII 模式】



@KBWFKM2

【Alt + Keypad 模式】

# Example

在扫描器的其它 HID Keyboard 相关设置为默认值，此项设置设置为控制字符输出“Control + ASCII 模式”时，识读数据为“A <HT>F（HT 为不可见字符，不显示在终端软件上）”（16 进制值分别为 0x41/0x09/0x46）的字符，扫描器虚拟键盘操作如下：

输入“A” -- 按下按键 A；

输入“Ctrl I” -- 由于 0x09 的数据对应控制功能键“I”，因此虚拟键盘将按住 Ctrl 不放，接着按下 I 键，最后同时松开 Ctrl 键和 I 键；

输入“F” -- 按下按键 F。

由于“Ctrl I”在某些字处理软件中对应转换字符为斜体的功能，因此完成上述操作可能会看到正常字符“A”和斜体的“F”。

如果此项设置设置为“Alt + Keypad 模式”，对于“<HT>”字符扫描器虚拟键盘操作如下：

输入“Alt 0 0 9” -- 虚拟键盘将按住 Alt 不放，接着依次按数字键盘的“0”，“0”和“9”，最后松开 Alt。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 控制字符对应表

ASCII Function	ASCII Value (HEX)	控制字符输出关闭	控制字符输出 Control + ASCII 模式
NUL	00	Null	Ctrl+@
SOH	01	Keypad Enter	Ctrl+A
STX	02	Caps Lock	Ctrl+B
ETX	03	ALT	Ctrl+C
EOT	04	Null	Ctrl+D
ENQ	05	CTRL	Ctrl+E
ACK	06	Null	Ctrl+F
BEL	07	Enter	Ctrl+G
BS	08	Left Arrow	Ctrl+H
HT	09	Horizontal Tab	Ctrl+I
LF	0A	Down Arrow	Ctrl+J
VT	0B	Vertical Tab	Ctrl+K
FF	0C	Delete	Ctrl+L
CR	0D	Enter	Ctrl+M
SO	0E	Insert	Ctrl+N
SI	0F	Esc	Ctrl+O
DLE	10	F11	Ctrl+P
DC1	11	Home	Ctrl+Q
DC2	12	Print Screen	Ctrl+R
DC3	13	Backspace	Ctrl+S
DC4	14	tab+shift	Ctrl+T
NAK	15	F12	Ctrl+U
SYN	16	F1	Ctrl+V
ETB	17	F2	Ctrl+W
CAN	18	F3	Ctrl+X
EM	19	F4	Ctrl+Y
SUB	1A	F5	Ctrl+Z
ESC	1B	F6	Ctrl+[
FS	1C	F7	Ctrl+\
GS	1D	F8	Ctrl+]
RS	1E	F9	Ctrl+6
US	1F	F10	Ctrl+-



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 控制字符对应表（续）

上表中“控制字符输出 Control + ASCII 模式”时的 0x1B~0x1F 对应的是美国键盘布局，如果是其他国家键盘布局，

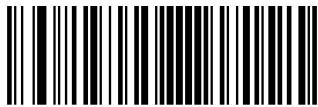
请参考下表：

国家	代码					
United	Ctrl+ [	Ctrl+\	Ctrl+]	Ctrl+6	Ctrl+-	
Belgium	Ctrl+ [	Ctrl+<	Ctrl+]	Ctrl+6	Ctrl+-	
Scandinavia	Ctrl+8	Ctrl+<	Ctrl+9	Ctrl+6	Ctrl+-	
France	Ctrl+^	Ctrl+8	Ctrl+\$	Ctrl+6	Ctrl+=	
Germany		Ctrl+Ã	Ctrl++	Ctrl+6	Ctrl+-	
Italy		Ctrl+\	Ctrl++	Ctrl+6	Ctrl+-	
Switzerland		Ctrl+<	Ctrl+..	Ctrl+6	Ctrl+-	
United Kingdom	Ctrl+ [	Ctrl+∅	Ctrl+]	Ctrl+6	Ctrl+-	
Denmark	Ctrl+8	Ctrl+\	Ctrl+9	Ctrl+6	Ctrl+-	
Norway	Ctrl+8	Ctrl+\	Ctrl+9	Ctrl+6	Ctrl+-	
Spain	Ctrl+ [	Ctrl+\	Ctrl+]	Ctrl+6	Ctrl+-	



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

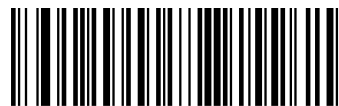
## 按键延迟

此参数指定模拟按键之间的延迟。当主机需要较慢的数据传输时，扫描下面的相应条形码以增加延迟。默认：无延迟。



@KBWDLY0

\*\*【无延迟】



@KBWDLY40

【长延迟 (40ms)】



@KBWDLY20

【短延迟 (20ms)】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 大写锁定设置

当开启时，可以反转条形码数据中包含的大小写字符。无论主机键盘上 Caps Lock 键的状态如何，都会发生此反转。默认：关闭。



@KBWCAP0

\*\*【关闭(非日文键盘)】



@KBWCAP2

【关闭(日文键盘)】



@KBWCAP1

【开启(非日文键盘)】



@KBWCAP3

【开启(日文键盘)】



若开启了“键盘仿真输入字符”或“大小写转换”功能，则此功能无效



开启此功能后，扫描器读取数据为“AbC”的条码，主机将得到“aBc”。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 大小写转换

扫描下面的相应条形码将所有条形码数据转换为所需的情况。默认：不转换。

- ◇ 若设置为“全部转换为大写字母”，则无论条码数据中字母是大写还是小写，全部转换为大写字母。
- ◇ 若设置为“全部转换为小写字母”，则无论条码数据中字母是大写还是小写，全部转换为小写字母。



@KBWCAS0  
\*\*【不转换】



@KBWCAS1  
【全转换成大写字母】



@KBWCAS2  
【全转换成小写字母】



若开启了“仿真输入字符”功能，则此功能无效。



设置“全转换成小写字母”，此时读取内容数据为“AbC”的条码，主机将得到“abc”的键盘输入。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 模拟数字小键盘



### 数字字符采用数字小键盘

不开启此功能，则所有输出均按主键盘对应键值输出。

开启此功能后，扫描器识读的条码数据中若包含数字“0~9”，则虚拟键盘将按数字小键盘对应的键值输出。

数字小键盘如上图所示，一般位于键盘的最右侧，由左上角的 Num Lock 控制其键值是数字还是功能键。虚拟键盘不独立控制 Num Lock 状态，而是与主机实际键盘的 Num Lock 状态一致，因此如果主机实际键盘关闭了 Num Lock（Num Lock 灯熄灭），则扫描器虚拟数字小键盘后，输出的是功能键而非数字。



@KBWNUM0

\*\*【关闭】



@KBWNUM1

【开启】



使用此功能前请务必先确认主机此时的 Num Lock 状态。

若已开启了“仿真输入字符”的功能，则此功能无效。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## Example

开启“模拟数字小键盘”后，扫描器读内容为“A4.5”的条码。

若主机“Num Lock”处于打开状态，则主机将得到“A4.5”的数据。

若主机“Num Lock”处于关闭状态，则主机将依次得到以下键盘输入：

首先得到“A”键，此字符不在数字键功能区中，因此正常发送；

其次得到数字“4”对应的功能键 -- 光标左移的指令；

最后得到“.”键，此字符正常发送；

由于数字“5”没有对应的功能键，因此不产生按键输入。

### 字符 ‘+’，‘-’，‘\*’，‘/’ 采用数字小键盘



@KBWNCH0  
\*\*【关闭】



@KBWNCH1  
【开启】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

---

## 快速模式

开启快速模式，扫描器可以更快地将字符发送到主机。如果主机会丢弃字符，请勿使用快速模式，或者调大**轮询速度**设置值。



@KBWFAS0

\*\*【关闭】



@KBWFAS1

【开启】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## 轮询速度

键盘轮询速度可以通过以下设置码设置成 1~10 毫秒。设置的数值越小，扫描器可以更快地将字符发送到主机。如果主机会丢弃字符，请调大轮询速度设置值。



@KBWPOR0  
【1ms】



@KBWPOR1  
【2ms】



@KBWPOR2  
【3ms】



@KBWPOR3  
\*\*【4ms】



@KBWPOR4  
【5ms】



@KBWPOR5  
【6ms】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---



@KBWPOR6

【7ms】



@KBWPOR7

【8ms】



@KBWPOR8

【9ms】



@KBWPOR9

【10ms】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## USB CDC 串口

当您使用 USB 连接，而同时又希望主机端采用串口方式接收数据，则应采用 USB 虚拟串口方式。从主机端系统接口来看，扫描器相当于通过串口方式与主机连接。



@INTERF8

\*\*【USB CDC 串口】

## HID POS (POS HID Bar Code Scanner)

### 简介

HID POS 接口被推荐为新的应用软件使用。在一个单独的 USB 报文中它就能发送 56 个字符，并且比模拟键盘接口的速度快。

特征：

- ◇ 基于 HID 接口，不需要定制驱动。
- ◇ 通讯速度比模拟键盘接口和传统的 RS-232 接口都快很多。



@INTERF5

【HID POS】

### 软件编程访问设备的方法

使用 CreateFile 把设备当成一个 HID 类型设备打开，然后使用 ReadFile 把扫描得到的数据传递给应用程序。使用 WriteFile 发送数据给设备。完整的 USB 和 HID 接口信息请参考：[www.USB.org](http://www.USB.org)。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 获取扫描数据

识读条形码成功之后，设备会发送以下的 Input 报文：

Byte	Bit							
	7	6	5	4	3	2	1	0
0	报文 ID = 0x02							
1	条码数据长度							
2-57	条码数据 (1-56)							
58-61	保留							
62	新大陆条码类型识别符号 或 无用：0x00							
63	-	-	-	-	-	-	-	解码数据继续

## 发送数据到设备

这个 Output 报文是用来发送数据到设备的。所有的通讯指令都可以采用这个方式发送到设备。

Byte	Bit							
	7	6	5	4	3	2	1	0
0	Report ID = 0x04							
1	Length of the Data("Output Data" field)							
2-63	Output Data (1-62)							



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## IBM SurePOS (Table-Top)



@INTERF6  
【IBM SurePOS (Table-Top)】

## IBM SurePOS (Hand-Held)



@INTERF7  
【IBM SurePOS (Hand-Held)】

## VID 和 PID 表

USB 使用 2 个号码来识别设备并找到正确的设备。第一个号码是 VID（厂商 ID），由 USB Implementers Forum（USB 应用厂商论坛）指派。新大陆自动识别公司的厂商 ID（VID）是 1EAB（十六进制）。第二个号码是 PID（设备 ID）。每种新大陆自动识别的产品都有一个范围的 PID，每个 PID 号码都包含一个产品类型的基数和接口类型。

设备名称	接口类型	PID（十六进制）	PID（十进制）
EM22	USB 键盘	0003	0003
	USB CDC 串口	0006	0006
	HID POS	0010	0016
	IBM SurePOS (Table-Top)	0020	0032
	IBM SurePOS(Hand-Held)	0021	0033



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 第六章 条码参数设置

### 简介

每种类型的条码都有其独特的属性，通过本章的设置码可以调整扫描器适应这些属性变化。

开启“允许识读”的条码类型越少，扫描器的识读速度越快。您可以禁止扫描器识读不会使用到的条码类型，以提高扫描器的工作性能。

### 综合设置

#### 允许所有条码

读取“允许识读所有条码”，扫描器将识读所有可识读的条码。



@ALLENA1

【使能所有条码】

#### 禁止所有条码

读取“禁止读所有条码”，扫描器将只能识读设置码，除设置码外的所有条码将无法识读。



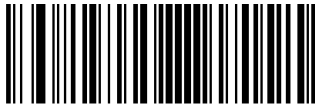
@ALLENA0

【禁止所有条码】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

### 允许所有一维码



@ALL1DC1  
【使能所有一维条码】

### 禁止所有一维码



@ALL1DC0  
【禁止所有一维条码】

### 允许所有二维条码



@ALL2DC1  
【使能所有二维条码】

### 禁止所有二维条码



@ALL2DC0  
【禁止所有二维条码】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## 一维条码双码

一维条码双码：两个上下排列的一维条码。双码的方向要一致，差异尽量小，距离尽量近。双码设置分成以下三种设置模式。

- ◇ **仅读单个一维条码：**任何时候设备一次只读取一个一维条码；
- ◇ **仅读双一维条码：**任何时候设备必须检测到一维双码，并两个码都解码成功后才发送解码信息。发送顺序为从上到下。
- ◇ **可读单双一维条码：**读码时先检测一维双码是否存在，若存在并解码成功则按照双码来发送，否则将做为单码处理。



@A1DDOU0

\*\*【仅读单个一维条码】



@A1DDOU2

【仅读双一维条码】



@A1DDOU1

【可读单双一维条码】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## Code 128

恢复出厂默认



@128DEF  
\*\*【Code 128 出厂默认设置】

使能/禁止识读



@128ENA1  
\*\*【使能】



@128ENA0  
【禁止】



若扫描器无法识别 Code 128 条码，请尝试读取“使能”设置码，然后再试一次。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 设置读码长度

扫描器可以配置为仅识读长度在（包括）最小和最大长度之间的 Code 128 条形码。



@128MIN

【最小长度（默认：1）】



@128MAX

【最大长度（默认：48）】



任何一维条码最大长度限制值不得超过 127，若最大长度小于最小长度，则为仅识读这两种长度的条码。若最大长度等于最小长度，则仅支持此长度。



限制扫描器只识读最小 8 字节，最大 12 字节的 Code128 条码：

1. 读“启动设置”
2. 读“最小长度”码
3. 读数据码“8”（见附录-数据码）
4. 读“保存”码（见附录-保存或取消）
5. 读“最大长度”码
6. 读数据码“1”
7. 读数据码“2”
8. 读“保存”码
9. 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## QR Code

恢复出厂默认



@QRCDEF  
\*\*【QR 出厂默认设置】

使能/禁止识读



@QRCENA1  
\*\*【使能】



@QRCENA0  
【禁止】



若扫描器无法识别 QR Code 条码，请尝试读取“使能”设置码，然后再试一次。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 设置读码长度

扫描器可以配置为仅识读长度在（包括）最小和最大长度之间的 QR Code 条码。



@QRDMIN

【最小长度（默认值：1）】



@QRDMAX

【最大长度（默认值：7089）】



任何二维条码最大长度限制值不得超过 65535 字节，且最大长度限制值不得小于最小长度限制值。

如果你希望只读某一个固定长度的 QR Code 条码，那么可以将最小长度限制值设置为与最大长度限制值相等。



限制扫描器只识读最小 8 字节，最大 12 字节的 QR Code 条码：

1. 读“启动设置”
2. 读“最小长度”
3. 读数据码“8”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“最大长度”
6. 读数据码“1”
7. 读数据码“2”
8. 读“保存”码
9. 读“退出设置”



#SETUPE0

【退出设置】



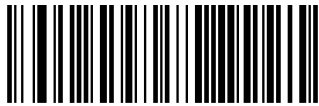
#SETUPE1  
【启动设置】

## QR 双码

QR 双码：两个上下或左右排列的 QR 条码。双码的方向要一致，差异尽量小，距离尽量近。

双码设置有三个设置模式：

- ◇ **仅读单个 QR 码：**任何时候设备一次只读取一个 QR 条码。
- ◇ **仅读双 QR 码：**任何时候设备必须检测到 QR 双码，并两个码都解码成功后才发送解码信息。发送顺序为从上到下或从左到右。
- ◇ **可读单双 QR 码：**读码时先检测 QR 双码是否存在，若存在并解码成功则按照双码来发送，否则将做为单码处理。



@QRCDU0  
\*\*【仅读单个 QR 码】



@QRCDU1  
【仅读双 QR 码】



@QRCDU2  
【可读单双 QR 码】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## QR 反相

QR 有正、反相条码之分。

- ◇ **正相条码：**浅色底，深色条的条码。
- ◇ **反相条码：**深色底，浅色条的条码。



@QRCINV0

\*\*【只识别正相条码】



@QRCINV1

【只识别反相条码】



@QRCINV2

【正反相条码都识别】

## 字符编码方式



@QRCENC0

\*\*【默认】



@QRCENC1

【UTF-8】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

## ECI 输出



@QRCECI0  
【禁止】



@QRCECI1  
\*\*【使能】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## 第七章 数据格式编辑

### 简介

您可以使用数据格式编辑功能更改扫描器的输出数据。例如，您可以使用数据格式编辑功能在条码数据的特定位置插入字符。

通常，当您扫描条形码时，会自动输出。但是创建数据格式时，必须使用“发送”命令（发送命令）才能让数据格式编辑功能输出数据。

一共可以设置四组数据格式，分别是数据格式 0，数据格式 1，数据格式 2，数据格式 3，可按实际需要四组数据格式分别进行设置。设置之后才能正确使用。数据格式包括数据格式的应用范围（如条码类型、条码长度）以及数据编辑命令。当识读的条码与使用的数据格式不匹配时，扫描器发出错误提示音（如果已开启了“数据格式不匹配错误提示音”）。

在进行数据格式编辑时，所有数据格式配置的总长度最大为 2048 字节。

数据格式编辑功能默认是关闭的，用户在使用前需要开启此功能。可以通过两种方式添加数据格式：一是通过 EasySet 软件进行配置，二是通过识读设置码。

用户在改变数据格式设置后，识读“出厂默认数据格式”设置码可清除所有格式并恢复默认设置。



@DFMDEF

\*\*【出厂默认数据格式】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 开启/关闭数据格式编辑

通过以下设置实现对数据格式编辑功能的关闭或开启。

### ◇ 关闭数据格式编辑

关闭数据格式编辑功能。

### ◇ 开启数据格式编辑，要求匹配，保留前后缀

开启数据格式编辑功能。与数据格式规则匹配的条码，按照数据格式要求输出，是否添加前后缀取决于前后缀的设置；与数据格式规则不匹配的条码不输出，扫描器发出错误提示音（如果已开启了“数据格式不匹配错误提示音”）。

### ◇ 开启数据格式编辑，要求匹配，丢弃前后缀

开启数据格式编辑功能。与数据格式规则匹配的条码，按照数据格式要求输出，不添加前后缀（忽略前后缀的设置）；与数据格式规则不匹配的条码不输出，扫描器发出错误提示音（如果已开启了“数据格式不匹配错误提示音”）。

### ◇ 开启数据格式编辑，不要求匹配，保留前后缀

开启数据格式编辑功能。与数据格式规则匹配的条码按照数据格式要求输出，是否添加前后缀取决于前后缀的设置；与数据格式规则不匹配的条码按原始信息输出，是否添加前后缀取决于前后缀的设置。

### ◇ 开启数据格式编辑，不要求匹配，丢弃前后缀

开启数据格式编辑功能。与数据格式规则匹配的条码按照数据格式要求输出，不添加前后缀（忽略前后缀的设置）；与数据格式规则不匹配的条码按原始信息输出，是否添加前后缀取决于前后缀的设置。



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---



@DFMENA0

\*\*【关闭数据格式编辑】



@DFMENA1

【开启数据格式编辑，要求匹配，保留前后缀】



@DFMENA2

【开启数据格式编辑，要求匹配，丢弃前后缀】



@DFMENA3

【开启数据格式编辑，不要求匹配，保留前后缀】



@DFMENA4

【开启数据格式编辑，不要求匹配，丢弃前后缀】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 添加数据格式

**步骤 1:** 识读“启动设置”。

**步骤 2:** 识读“添加数据格式”。



@DFMADD  
【添加数据格式】

**步骤 3:** 选择数据格式

识读数据码“0”或“1”或“2”或“3”，它们分别代表选择数据格式 0~3。

**步骤 4:** 依次识读数据码“6” “9” “9” “9”。

**步骤 5:** 选择条码序号

数据格式仅对指定类型的条码起作用。参考附录-[条码序号对照表](#)。条码序号由 3 个数据位组成。若设置数据格式对所有条码类型均有效，则设置条码序号为 **999**。例如只针对 EAN-13 做处理，则识读数据码“005”。

**步骤 6:** 设置条码数据长度

数据格式仅对指定长度的条码起作用。数据长度由 4 个数据位组成。例如设置条码数据长度为 32，则识读数据码“0032”。若设置数据格式对所有数据长度的条码均有效，则设置条码长度为 **9999**。

**步骤 7:** 设置数据格式编辑命令

扫描数据码来设置所需的数据编辑命令。例如，数据编辑命令为 F141，则识读数据码“F141”。详细命令请参考[数据格式编辑命令](#)。

**步骤 8:** 识读“保存”设置码。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---

## Example

添加数据格式 0，该数据格式仅适用于数据长度为 10 字节的 Code 128，发送条码数据并在其末尾添加字符“A”，设置步骤如下：

1. 读“启动设置”
2. 读“添加数据格式”
3. 添加数据格式 0，读数据码“0”
4. 读数据码“6” “9” “9” “9”
5. Code 128 条码序号 002，读数据码“0” “0” “2”
6. 条码长度为 10 字节，读数据码“0” “0” “1” “0”
7. 发送所有数据后发送“A”，对应的数据格式编译命令为 F141，读数据码“F” “1” “4” “1”
8. 读“保存”
9. 读“退出设置”



#SETUPE0

【退出设置】

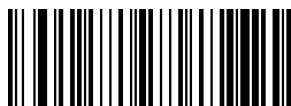


#SETUPE1  
【启动设置】

---

## 清除数据格式

清除数据格式配置有两种方法，一种是识读“清除所有数据格式”设置码，将所有数据格式全部清除；另一种是先识读“清除一组数据格式”设置码，再识读需要清除的那组数据格式的编号的数据码，再读取“保存”设置码，清除 4 组数据编辑格式配置中的一组。例如要清除“数据格式 2”，先识读“清除一组数据格式”设置码，再识读数据码“2”，最后识读“保存”。



@DFMCAL  
【清除所有数据格式】



@DFMCLR  
【清除一组数据格式】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 选择数据格式

开启数据格式编辑功能以后，可以选择要使用哪一组数据格式，通过以下的四条设置码选择使用不同的数据格式。默认：数据格式 0。



@DFMUSE0

\*\*【数据格式 0】



@DFMUSE1

【数据格式 1】



@DFMUSE2

【数据格式 2】



@DFMUSE3

【数据格式 3】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 单次使用数据格式

单次使用数据格式是指只对设置之后识读的一个条码起作用，之后使用的数据格式将恢复到原先的数据格式。可以通过以下设置码选择单次使用的数据格式。例如，已经设置扫描器选择数据格式 3，但又希望在识读某个条码时使用数据格式 1，这种情况下就需在识读那个条码前扫描“单次使用数据格式 1”设置码。



@DFMSIN0  
【单次使用数据格式 0】



@DFMSIN2  
【单次使用数据格式 2】



@DFMSIN1  
【单次使用数据格式 1】



@DFMSIN3  
【单次使用数据格式 3】

## 数据格式不匹配错误提示音

如果开启了数据格式不匹配提示音，则在识读不符合数据格式要求的条码时，扫描器会发出错误提示音。



@DFMTON0  
【关闭】



@DFMTON1  
\*\*【开启】



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## 查询数据格式

### ◇ 查询当前的数据格式

获取当前配置的数据格式 0 到数据格式 3 的配置内容。

### ◇ 查询出厂默认的数据格式

获取出厂配置的数据格式 0 到数据格式 3 的配置内容。

查询到的所有数据格式配置内容格式如下：

Data Format 0:xxxx;

Data Format 1:xxxx;

Data Format 2:xxxx;

Data Format 3:xxxx;



@DFMQCU

【查询当前的数据格式】



@DFMQFA

【查询出厂默认的数据格式】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 数据格式编辑命令

使用数据格式编辑器就如同在输入数据字符串中移动虚拟光标。下列命令既可用于移动光标至不同位置，也可用于选择、替换、将数据插入最终输出等。命令中所涉及的十六进制值参考附录的《[ASCII 码表](#)》。

初始光标位置：指向未经过数据格式编辑处理的原始条码信息的第一个字节。



如以上条码，初始光标位置在“1”的左侧。

### 发送命令

#### F1 发送全部字符

命令格式=F1xx（xx：待插入字符的十六进制值）

输出光标右侧的全部字符，然后输出字符 xx。

#### F2 发送若干个字符

命令格式=F2nngx（nn：00-99，发送的字符长度；xx：待插入字符的十六进制值）

输出光标右侧的 nn 个字符，然后输出字符 xx。

#### F2 示例：发送若干个字符



发送上述条码内容的前 10 个字符，再发送一个回车。命令字符串：**F2100D**

F2：“发送若干个字符”命令

10：发送的字符长度(从当前光标位置开始输出)

0D：回车的十六进制值

输出结果：**1234567890<CR>**



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

### F3 发送某字符前的所有字符

命令格式=F3ssxx (ss: 某字符的十六进制值; xx: 待插入字符的十六进制值)

输出当前光标右侧至字符 ss 左侧之间的所有字符 (不包括字符 ss), 其后输出字符 xx。光标向右移至字符 ss 的左侧。

### F3 示例: 发送某字符前的所有字符且在末尾添加一个字符



1234567890ABCDEFGHIJ

发送上述条码内容中“D”之前 (不包括 D) 的所有字符, 再发送一个回车。

命令字符串: **F3440D**

F3: “发送某字符前的所有字符”命令

44: D 的十六进制值

0D: 回车的十六进制值

输出结果: **1234567890ABC<CR>**

### E9 发送末尾若干字符前所有字符

命令格式=E9nn (nn: 00-99, 不输出的末尾字符的长度)

输出当前光标右侧至末尾 nn 个字符前 (左侧) 的所有字符。光标移至最后一个发送字符的右侧 (即末尾 nn 个字符的左侧)。

### F4 多次发送一个字符

命令格式=F4xxnn (xx: 待插入字符的十六进制值; nn: 00-99, 字符输出的次数)

nn 次输出字符 xx, 光标位置不变。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

**E9 和 F4 示例：发送末尾若干字符前所有字符，再发送两个 Tab**



1234567890ABCDEFGHIJ

发送上述条码内容末 9 位字符前的所有字符，再发送两个 tab。

命令字符串：**E909F40902**

E9: “发送末尾若干字符前所有字符” 命令

08: 不输出的末尾字符的长度

F4: “多次插入一个字符” 命令

09: Horizontal tab 的十六进制值

02: 发送 Tab 的次数

输出结果：**1234567890A<tab><tab>**

### B3 插入码制名称

在输出信息中插入码制名称，不移动光标。

### B4 插入条码长度

在输出信息中插入条码长度，不移动光标。条码长度以数字形式输出，且不包括前置零。

**B3 和 B4 示例：插入码制名称和长度**



1234567890ABCDEFGHIJ

发送上述条码的码制名称、长度和条码数据，间隔符使用空格，最后发送一个回车。

命令字符串：**B3F42001B4F42001F10D**

B3: “插入码制名称” 命令



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

F4: “多次插入一个字符” 命令

20: 空格的十六进制值

01: 发送空格的次数

B4: “插入条码长度” 命令

F4: “多次插入一个字符” 命令

20: 空格的十六进制值

01: 发送空格的次数

F1: “发送全部字符” 命令

0D: 回车的十六进制值

输出结果: **Code 128 20 1234567890ABCDEFGHIJ<CR>**

## 移动命令

**F5 向右移动光标若干个字符**

命令格式 = F5nn (nn: 00-99, 光标向右移动的字符数)

向右移动光标 nn 个字符。

**F5 示例: 向右移动光标并发送条码数据**



1234567890ABCDEFGHIJ

上述条码中, 向右移动光标 3 个字符, 发送光标右侧的所有条码数据, 最后发送一个回车。

命令字符串: **F503F10D**

F5: “向右移动光标若干个字符” 命令

03: 光标向右移动的字符数



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

F1: “发送全部字符”命令

0D: 回车的十六进制值

输出结果: 4567890ABCDEFGHJ<CR>

### F6 往左移动光标若干个字符

命令格式=F6nn (nn: 00-99, 光标往左移动的字符数)

向左移动光标 nn 个字符。

### F7 移动光标至起始位置

命令格式=F7

移动光标至输入信息的首字符的左侧。

### EA 移动光标至末尾位置

命令格式=EA

移动光标至输入信息的末尾字符的左侧。

## 搜索命令

### F8 向右搜索字符

命令格式=F8xx (xx: 待查找字符的十六进制值)

在输入信息中从当前光标位置开始向右搜索字符 xx, 光标移动到该字符的左侧。

### F8 示例: 发送条码信息中某个字符及其后数据



1234567890ABCDEFGHIJ



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

在上述条码信息中搜索字母“D”，发送字母“D”及其后数据，最后发送一个回车。

命令字符串：**F844F10D**

F8：“向右搜索字符”命令

44：“D”的十六进制值

F1：“发送全部字符”命令

0D：回车的十六进制值

输出结果：**DEFGHIJ<CR>**

### F9 向左搜索字符

命令格式=F9xx（xx：待查找字符的十六进制值）

在输入信息中从当前光标位置向左搜索字符 xx，光标移动到该字符的左侧。

### B0 向右搜索字符串

命令格式=B0nnnnS（nnnn：0000-9999，待查找字符串的长度；S：字符串中所有字符的十六进制值）

在输入信息中从当前光标位置向右搜索字符串 S，光标移动该字符串的左侧。例如，B0000454657374 表示向右搜索首次出现的字符串“Test”。

### B0 示例：发送条码信息中某字符串及其后数据



1234567890ABCDEFGHIJ

在上述条码信息中搜索“FGH”，发送“FGH”及其后数据，最后发送一个回车。

命令字符串：**B00003464748F10D**

B0：“向前搜索字符串”命令

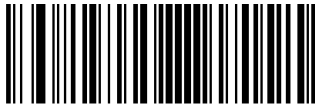
0003：字符串的长度（3个字符）

46：“F”的十六进制值



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

47: “G”的十六进制值

48: “H”的十六进制值

F1: “发送全部字符”命令

0D: 回车的十六进制值

输出结果: **FGHIJ<CR>**

### B1 向左搜索字符串

命令格式=B1nnnnS (nnnn: 0000-9999, 待查找字符串的长度; S: 字符串中所有字符的十六进制值)

在输入信息中从当前光标位置向左搜索字符串 S, 光标移动到该字符串的左侧。例如, B1000454657374 表示向左搜索首次出现的字符串“Test”。

### E6 向右搜索不匹配字符

命令格式=E6xx (xx: 待查找的不匹配字符的十六进制值)

在输入信息中从当前光标位置向右搜索首个出现的与 xx 不匹配的字符, 光标移动到该字符的左侧。

### E6 示例: 删除条码信息中的前置零



0000123abc

上述条码信息中带前置零, 要求发送前置零后面的条码数据。向右搜索首个非零字符, 发送该字符及其后数据, 最后发送一个回车。

命令字符串: **E630F10D**

E6: “向右搜索不匹配字符”命令

30: “0”的十六进制值

F1: “发送全部字符”命令

0D: 回车的十六进制值

输出结果: **123abc<CR>**



#SETUPE0  
【退出设置】





#SETUPE1  
【启动设置】

### E7 往左搜索不匹配字符

命令格式 = E7xx (xx: 待查找的不匹配字符的十六进制值)

在输入信息中从当前光标位置向左搜索首个出现的与 xx 不匹配的字符，光标移动到该字符的左侧。

## 其他命令

### FB 阻止输出字符

命令格式 = FBnnxxyy.zz (nn: 阻止输出的同一字符的数量; xxyy.zz: 阻止输出字符的十六进制值)

从当前光标右侧开始，阻止输出字符（最多可达 15 个不同字符），光标的移动由其他命令决定。

### FB 示例：删除条码信息中的空格等字符



12 34\_5\*6 78

上述条码信息中带空格、下划线和“\*”，要求删除这 3 种字符后发送其余数据。

命令字符串：**FB03205F2AF10D**

FB: “阻止输出字符”命令

03: 阻止 3 种字符

20: 空格的十六进制值

5F: “\_”的十六进制值

2A: “\*”的十六进制值

F1: “发送全部字符”命令

0D: 回车的十六进制值

输出结果：**12345678<CR>**



#SETUPE0  
【退出设置】



#SETUPE1  
【启动设置】

#### E4 替换字符

命令格式=E4nnxx1xx2yy1yy2...zz1zz2 (nn: 被替换字符数+替换字符数; xx1: 被替换字符的十六进制值, xx2: 替换字符的十六进制值, 以此类推)

从当前光标右侧开始, 替换输出字符 (最多可达 15 个字符), 不移动光标。

#### E4 示例: 将条码信息中的零替换成回车



当条码中含有主机应用不需要的字符时, 可使用 E4 命令将不需要的字符替换成别的字符。此示例要求将条码中的零替换成回车。

命令字符串: **E402300DF10D**

E4: “替换字符”命令

02: 被替换字符数+替换字符数 (0 替换成 CR, 总计 2 个字符)

30: “0”的十六进制值

0D: 回车的十六进制值 (用回车替换 0)

F1: “发送全部字符”命令

0D: 回车的十六进制值

输出结果: **123**

**456**

**78**

**AB<CR>**

#### BA 替换字符串多次

命令格式=BAAnnN1SS1NN2SS2 (nn: 替换次数, nn=00 表示替换全部 SS1; N1: 被替换字符串的长度; SS1: 被替换字符串的十六进制值; NN2: 新字符串的长度; SS2: 新字符串的十六进制值)

从当前光标右侧开始, 替换过程不移动光标。仅遍历原始信息一次, 不做反复查找替换; 字符串出现次数不足, 不报错。NN1>0, NN2>=0。



#SETUPE0  
【退出设置】



#SETUPE1  
【启动设置】

**BA 示例：将条码信息中的 23 替换成 XYZ**



1234Abc23R0123U

命令字符串：**BA020232330358595AF100**

BA: “替换字符串多次”命令

02: 替换次数

02: 字符串“23”长度

3233: 字符串“23”的十六进制值

03: 字符串“XYZ”长度(如果设为 00 则其后不加字符串)

58595A: 字符串“XYZ”的十六进制值

F1: “发送全部字符”命令

00: 空字符的十六进制值

输出结果：**1XYZ4AbcXYZR0123U**

**EF 字符间延时，用于 USB 键盘的通讯模式**

命令格式=EFnnnn (nnnn 表示延时时长为 5ms 的 nnnn 倍，用十进制表示)

在字符间插入一段延时，延时长度为 5ms 的倍数，最大可到 49995ms。最多可插入 255 段延时。

**EF 示例：在条码的第 5 个字符和第 7 个字符后插入延时**



1234567890ABCDEFGHJ

命令字符串：**F20500EF0200F20200EF0200F100**

F2: “发送若干个字符”命令

05: 发送的字符长度(从当前光标位置开始输出)



#SETUPE0  
【退出设置】



#SETUPE1  
【启动设置】

00: NUL 的十六进制

EF: “延时”命令

0200: 延时长度为 5ms 的 200 倍, 即 1s

F2: “发送若干个字符”命令

02: 发送的字符长度(从当前光标右侧开始输出)

00: NUL 的十六进制

EF: “延时”命令

0200: 延时长度为 5ms 的 200 倍, 即 1s

输出结果: 12345{1s 延时}67{1s 延时}890ABCDEFGHJI

#### B5 插入按键, 用于 USB 键盘的通讯模式

命令格式: B5nnssxx (nn: 插入的按键总数 (不包括控制键); ss: 控制键序号, 为十六进制数; xx: 表示按键序号, 为十六进制数)

按键编号参考附录-[键盘按键序号](#)中的 104 键和 105 键键盘布局, 控制键编号见下表:

控制键	
无控制键	00
左 Shift	01
右 Shift	02
左 Alt	04
右 Alt	08
左 Control	10
右 Control	20



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---

**B5 示例：美国英语 104 键盘布局模式，在条码的起始位置插入 aBc**



12345678

命令串字符：**B503001F01320030F100**

B5: “插入按键”命令

03: 插入的按键个数

00: 无控制键

1F: “a”按键序号

01: 左 Shift 控制键序号

32: “b”按键序号

00: 无控制键

30: “c”按键序号

F1: 发送全部字符

00: Nul 的十六进制数

输出结果：**aBc12345678**



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 第八章 前后缀设置

### 简介

扫描器解码成功后，获得一串数据，这串数据可以是数字，英文，符号等等，对于二维码还可以是汉字，这串数据就是条码所包含的数据信息。在实际应用中，我们可能不仅仅需要条码的数据信息，或者说条码所包含的数据信息不能满足您的需要。如您可能希望知道获得的这串数据信息是来自于哪一种类型的条码，或者想知道条码信息是在哪一天扫描的，或者您希望在扫描完一个条码后，记录条码的文本可以自动换行回车，而这些可能不包含在条码的数据信息中。

在制码时增加这些内容，势必增加条码长度且灵活性不够，不是提倡的做法。此时我们想到，人为的在条码的数据信息前面或者后面增加一些内容，而且这些增加的内容，可以根据需求实时改变，可以选择增加或者屏蔽，这就是条码数据信息的前后缀，增加前后缀的方法，即满足了需求又无需修改条码信息的内容。



条码处理的步骤：

1. 数据格式编辑
2. 添加前后缀
3. 数据打包
4. 添加结束符后缀



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 综合设置

### 所有前后缀

- ◇ 若设置为“禁止”，则解码后的信息中只有条码的数据信息，没有前后缀。
- ◇ 若设置为“使能”，则将在解码信息前后增加“CodeID 前缀”，“AIM 前缀”，“自定义前缀”，“自定义后缀”，“结束符后缀”。



@APSENA0

\*\*【禁止所有前后缀】



@APSENA1

【使能所有前后缀】

### 前缀顺序



@PRESEQ0

\*\*【Code ID +自定义前缀+AIM ID】



@PRESEQ1

【自定义前缀+Code ID+AIM ID】



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 自定义前缀

自定义前缀在解码信息之前添加用户自定义的字符串。例如，允许添加自定义前缀并设置前缀为字符串“AB”，识读数据为“123”的条码后，扫描器在“123”字符串前添加“AB”字符串，主机端接收到“AB123”。



@CPRENA0  
\*\*【禁止】



@CPRENA1  
【使能】

## 修改自定义前缀

自定义首先读取“修改自定义前缀”，然后按顺序读取要设置的前缀字符串中每个字节的 16 进制值，最后读取“保存设置”完成自定义前缀的设置。注意：自定义前缀字符串总长度不得超过 10 个字符，字符取值范围为 0x00~0xFF。



@CPRSET  
【修改自定义前缀】

### Example

设置自定义前缀为“CODE”（16 进制值为 0x43/0x4F/0x44/0x45）：

1. 读“启动设置”
2. 读“修改自定义前缀”码
3. 读以下数据码：“4”“3”“4”“F”“4”“4”“4”“5”
4. 读“保存”码
5. 读“退出设置”

这样设置完成后，只要自定义前缀设置为“使能”，则读取任何条码，扫描器都将在数据前添加自定义前缀字符串“CODE”。



#SETUPE0  
【退出设置】





#SETUPE1

【启动设置】

## AIM ID 前缀

AIM 是 Automatic Identification Manufacturers（自动识别制造商协会）的简称，AIM ID 为各种标准条码分别定义了识别代号，具体定义见下表。扫描器在解码后可以将此识别代号添加在条码数据前，即 AIM 前缀。前缀格式：“J”+AIM 前缀+数字“0”，如 Code 128 的 AIM ID 前缀为“JC0”。



@AIDENA0

\*\*【禁止】



@AIDENA1

【使能】



用户不可自定义 AIM ID。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## Code ID 前缀

除了 AIM 前缀可用于识别不同的条码类型外，用户也可以使用 Code ID 前缀来标识条码类型。与 AIM 前缀不同，每种条码类型所对应的 Code ID 前缀是可以自定义的。所有条码的 Code ID 为 1 个或 2 个字符，并且必须为字母，不能设为数字，不可见字符，或标点符号等。



@CIDENA0  
\*\*【禁止】



@CIDENA1  
【使能】

## 默认 Code ID



@CIDDEF  
【恢复默认 Code ID 设置】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

## 修改 Code ID

修改 Code ID 和恢复 Code ID 出厂设置的方法请参考下列示例。

**E**  
*sample*

修改 PDF417 的 CodeID 为“p”（16 进制值为 0x70）：

1. 读“启动设置”
2. 读“PDF417”
3. 读数据码“7”，“0”（见附录-数据码）
4. 读“保存”（见附录-保存或取消）
5. 读“退出设置”

恢复包括 PDF417 在内的所有条码的 Code ID 为默认值：

1. 读“启动设置”
2. 读“恢复默认 Code ID 设置”
3. 读“退出设置”



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

---

### 修改一维条码 Code ID



@CID002  
【Code 128】

### 修改二维条码 Code ID



@CID033  
【QR】



#SETUPE0  
【退出设置】

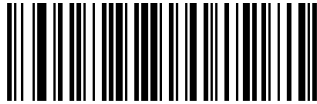


#SETUPE1  
【启动设置】

## 自定义后缀

自定义后缀就是在解码信息后添加用户自定义的字符串。

例如，允许添加自定义后缀并设置后缀为字符串“AB”，识读数据为“123”的条码后，扫描器在“123”字符串后添加“AB”字符串，主机端接收到“123AB”。



@CSUENA0  
\*\*【禁止】



@CSUENA1  
【使能】

## 修改自定义后缀

首先读取“修改自定义后缀”，然后按顺序读取要设置的后缀字符串中每个字节的 16 进制值，最后读取“保存设置”完成自定义后缀的设置。注意：自定义后缀字符串总长度不得超过 10 个字符。



@CSUSET  
【修改自定义后缀】

**E**  
*xample*

设置自定义后缀为“CODE”（16 进制值为 0x43/0x4F/0x44/0x45）：

1. 读“启动设置”
2. 读“修改自定义后缀”设置码
3. 读以下数据码：“4”“3”“4”“F”“4”“4”“4”“5”
4. 读“保存”设置码
5. 读“退出设置”

这样设置完成后，只要自定义后缀设置为“使能”，则读取任何条码，扫描器都将在数据后添加自定义后缀字符串“CODE”。



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---

## 数据打包

对于一些应用，对数据完整性、可靠性等有高要求，可使用对数据打包后输出的方式，通过内容格式的检查 and 校验的手段确保完整而可靠的数据传输。

使用打包格式的数据传输，需要主机上的软件对打包格式进行解析。



@PACKAG0

\*\*【禁止数据打包】



@PACKAG1

【使能数据打包，格式一】



@PACKAG2

【使能数据打包，格式二】



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

## 打包格式

格式一：[STX + ATTR + LEN] + [AL\_TYPE + DATA] + [LRC]

- ◇ STX: 0x02
- ◇ ATTR: 0x00
- ◇ LEN: DATA 数据的长度，使用两个字节表示，高位字节在前，取值范围为 0~65535.
- ◇ AL\_TYPE: 0x36
- ◇ DATA: 数据信息内容。
- ◇ LRC: 校验字符。

LRC: 校验字符。LRC 校验字符的算法： $0xFF \wedge LEN \wedge AL\_TYPE \wedge DATA$ （ $\wedge$ 表示 算术异或操作），所有数据按字节单位进行异或运算。即 0xFF 与 LEN 的第一个字节进行异或得到一个字节数据再与 LEN 的第二个字节进行异或，一次重复异或操作，直到全部数据异或完，最后获得的一个字节数据就是校验字符。

格式二为：[STX + ATTR + LEN] + [AL\_TYPE ] + [Symbology\_ID + DATA] + [LRC]

- ◇ STX: 0x02
- ◇ ATTR: 0x00
- ◇ LEN: Symbology ID + DATA 数据的长度，使用两个字节表示，高位字节在前，取值范围为 0~65535.
- ◇ AL\_TYPE: 0x3B
- ◇ Symbology\_ID: 条码序号，1 个字节（条码序列号参考附录“条码序列号对照表”）
- ◇ DATA: 数据信息内容。
- ◇ LRC: 校验字符。

LRC: 校验字符。LRC 校验字符的算法： $0xFF \wedge LEN \wedge AL\_TYPE \wedge Symbology\_ID \wedge DATA$ （ $\wedge$ 表示 算术异或操作），所有数据按字节单位进行异或运算。即 0xFF 与 LEN 的第一个字节进行异或得到一个字节数据再与 LEN 的第二个字节进行异或，一次重复异或操作，直到全部数据异或完，最后获得的一个字节数据就是校验字符。



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 结束符后缀

结束符后缀用于标志一段完整数据信息的结束。结束符后缀独立存在，不会参与任何其他形式的数据格式化；结束符后缀一定是一段数据发送时最后的内容，其后不会再有任何追加数据。



@TSUENA0  
\*\*【禁止】



@TSUENA1  
【使能】

## 修改结束符后缀

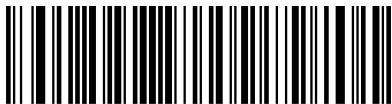
首先读取“修改结束符后缀”，然后按顺序读取要设置的后续字符串中每个字节的 16 进制值，最后读取“保存设置”完成结束符后缀的设置。注意：结束符后缀字符串总长度不得超过 2 个字符。



@TSUSET  
【修改结束符后缀】



@TSUSET0D  
\*\*【修改结束符后缀为<CR> (0x0D)】



@TSUSET0D0A  
【修改结束符后缀为<CR> <LF> (0x0D, 0x0A)】



#SETUPE0  
【退出设置】





#SETUPE1  
【启动设置】

## 第九章 批处理设置

### 简介

在识读设备需要进行多项设置时，逐个设置可能很繁琐，这时候我们可以把需要设置的所有信息存成一个条码信息，设备读取该条码后就可以完成多项设置。

以下为批处理设置准则：

- 1、 批处理命令中每条命令的格式为命令+参数。
- 2、 命令用分号结束，注意各条命令间不能有空格。
- 3、 将此命令在制码软件中制作为任意扫描器支持的条码即可，推荐用二维码。

例如：连续模式读码（命令：SCNMOD3）、一次读码超时时间 2 秒（命令：ORTSET2000）、禁止识读 Interleaved 2 of 5 码的命令码（命令：I25ENA0），则批处理的命令内容如下（@表示指令需要存储）：

@ILLSCN1;SCNMOD3;ORTSET2000;I25ENA0;

需要设置时读取“启动批处理设置”，再读取制作好的批处理码可完成设置。



@BATCHS  
【启动批处理设置】



#SETUPE0  
【退出设置】



#SETUPE1

【启动设置】

---

## 生成批处理指令

一条批处理指令可以包含很多条单独指令。每条单独指令由分号隔开。

指令结构：指令 (+设置信息)

一共有 3 种设置指令形式

### 1、设置语法 1：指令

这一形式的指令最多，就是不要借助数据码，一次设置就可以完成的设置指令。

例如：

设置波特率为 38400bps 的指令为 @232BAD6

设置自动读码的指令为 @SCNMOD2

### 2、设置语法 2：指令+数字

这一形式的指令用于设置一些参数的值，这些参数包括：条码读码最大最小长度设置、一次读码超时设置、相同读码延迟设置、灵敏度设置等。

例如：

设置一次读码延迟的时间为 3000 毫秒的指令为：@ORTSET3000

设置解码成功声音持续时间为 100 毫秒的指令为：@GRBDUR100

### 3、设置语法 3：指令+十六进制数

这一形式的指令可用于设置自定义前缀、自定义后缀、结束符后缀、CodeID 等。

注意：指令中每两个十六进制字符表示一个要设置的字符。

例如：

设置自定义前缀为“J”的指令为：@CPRSET4A

设置 Code128 的 CodeID 为“j”的指令为：@CID0026A



#SETUPE0

【退出设置】



#SETUPE1

【启动设置】

---

## 制作批处理条码

批处理条码可以采用任何扫描器支持且已使能的码制，建议采用二维条码。

例如，需要设置并保存：连续模式读码，修改一次读码超时时间为 2 秒。指令为：“@ SCNMOD3;ORTSET2000”。制作数据为“@SCNMOD3;ORTSET2000”的 PDF417 码如下：



#SETUPE0

【退出设置】



#SETUPE1  
【启动设置】

## 使用批处理条码

按如下步骤依次识读设置码和批处理条码：



@SETUPE1  
【启动设置】



@BATCHS  
【启动批处理设置】



【批处理设置码】



@SETUPE0  
【退出设置】



#SETUPE0  
【退出设置】

# 附录

## 数据码

读取数据码完成后务必读取“保存”以保存数据码设置。

**0-9**



@DIGIT0

**【0】**



@DIGIT1

**【1】**



@DIGIT2

**【2】**



@DIGIT3

**【3】**



@DIGIT4

**【4】**



@DIGIT5

**【5】**



@DIGIT6

【6】



@DIGIT8

【8】



@DIGIT7

【7】



@DIGIT9

【9】

---

A~F



@DIGITA

【A】



@DIGITB

【B】



@DIGITC

【C】



@DIGITD

【D】



@DIGITE

【E】



@DIGITF

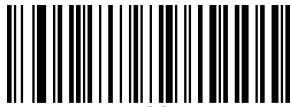
【F】

---

## 保存或取消

读完数据码后要读取保存码才能将读取到的数据保存下来。如果在读取数据码时出错，除了重新设置外，您还可以取消读取错误的码。

如读取某个设置码，并依次读取数据“1”，“2”，“3”，此时若读取“取消前一次读的一位数据”，将取消最后读的数字“3”，若读取“取消前面读的一串数据”将取消读取到的数据“123”，若读取“取消当前设置”将连设置码一起取消，但此时设备还处于启动设置状态。



@DIGSAV

【保存】



@DIGCAN

【取消当前设置】



@DIGDEL

【取消前一次读的一位数据】



@DIGDAL

【取消前面读的一串数据】



## 默认设置表

参数名称	默认设置	备注
<b>系统设置</b>		
设置码功能	关闭	
发送设置码信息	不发送	
解码成功 LED 灯	开启	
解码成功 LED 灯持续时间	长 (220 毫秒)	
开机提示音	开启	
解码成功声音	开启	
解码成功声音持续时间	中 (80 毫秒)	
解码成功 LED 灯颜色	条码支付模式	
解码成功声音频率	中 (2730HZ)	
解码成功声音音量	大	
识读模式	连续模式	
一次读码超时	3000 毫秒	0-3600000 毫秒
重读延迟	开启	
重读延迟时间	1500 毫秒	1-3600000 毫秒
重读超时复位	开启	
读码成功延迟	关闭	
读码成功延迟时间	500 毫秒	1-3600000 毫秒
解码超时	500 毫秒	1-3000 毫秒
串行触发指令	禁止	
识读偏好	条码支付模式	
禁止/允许读码	允许读码	
解码中心区域	全区域解码	
图像翻转	正常图像	
传送读码未成功信息	禁止	
修改未读码成功信息	NG	
<b>通讯接口</b>	USB CDC 串口	
<b>RS-232 通讯设置</b>		
波特率	9600	
奇偶校验	无校验	
数据位传输	8 位	
停止位	1 位	
硬件自动流控	关闭	

<b>USB 通讯设置</b>		
国家键盘布局	美国英语	USB 键盘
未知字符提示音	关闭	USB 键盘
键盘仿真输入字符	关闭	USB 键盘
Code Page	Code Page 1252 (拉丁, 西欧)	USB 键盘
Unicode 输出	关闭	USB 键盘
前导“0”	开启	USB 键盘
控制字符输出	关闭	USB 键盘
按键延迟	无延迟	USB 键盘
大小写锁定设置	关闭(非日文键盘)	USB 键盘
大小写转换	不转换	USB 键盘
模拟数字小键盘		USB 键盘
数字字符采用数字小键盘	关闭	USB 键盘
字符 '+', '-', '*', '/' 采用数字小键盘	关闭	USB 键盘
快速模式	关闭	USB 键盘
轮询速度	4 毫秒	USB 键盘
<b>条码参数设置</b>		
<b>Code 128</b>		
使能/禁止识读	使能	
最大长度	48	
最小长度	1	
<b>QR Code</b>		
使能/禁止识读	使能	
最大长度	7089	
最小长度	1	
QR 双码	仅读单个 QR 码	
QR 反相	只识别正相条码	
字符编码方式	默认	
ECI 输出	使能	
<b>数据格式编辑</b>		
开启/关闭数据格式	禁止	
选择数据格式	数据格式 0	
数据格式不匹配错误提示音	开启	
多重数据格式设置	禁止	
<b>前后缀设置</b>		
所有前后缀	禁止	

前缀顺序	Code ID+自定义前缀+AIM ID	
自定义前缀	禁止	
AIM ID 前缀	禁止	
Code ID 前缀	禁止	
自定义后缀	禁止	
数据打包	禁止数据打包	
结束符后缀	禁止	

---

## AIM ID 列表

条码类型	AIM ID	可能的 AIM ID 限定参数 (m)
Code 128	JC0	
QR Code	JQm	0-6

参考资料: ISO/IEC 15424: 2008 信息技术 – 自动识别及数据获取技术 – 数据载体标识符 (包括符号表示标识符)

---

## Code ID 列表

条码类型	Code ID
Code128	j
QR Code	s

---

## 条码序号对照表

条码类型	序号
Code 128	002
QR Code	033

## ASCII 码表

十六进制	十进制	字符
00	0	NUL (Null char.)
01	1	SOH (Start of Header)
02	2	STX (Start of Text)
03	3	ETX (End of Text)
04	4	EOT (End of Transmission)
05	5	ENQ (Enquiry)
06	6	ACK (Acknowledgment)
07	7	BEL (Bell)
08	8	BS (Backspace)
09	9	HT (Horizontal Tab)
0a	10	LF (Line Feed)
0b	11	VT (Vertical Tab)
0c	12	FF (Form Feed)
0d	13	CR (Carriage Return)
0e	14	SO (Shift Out)
0f	15	SI (Shift In)
10	16	DLE (Data Link Escape)
11	17	DC1 (XON) (Device Control 1)
12	18	DC2 (Device Control 2)
13	19	DC3 (XOFF) (Device Control 3)
14	20	DC4 (Device Control 4)
15	21	NAK (Negative Acknowledgment)
16	22	SYN (Synchronous Idle)
17	23	ETB (End of Trans. Block)
18	24	CAN (Cancel)
19	25	EM (End of Medium)
1a	26	SUB (Substitute)
1b	27	ESC (Escape)
1c	28	FS (File Separator)
1d	29	GS (Group Separator)
1e	30	RS (Request to Send)
1f	31	US (Unit Separator)
20	32	SP (Space)

21	33	! (Exclamation Mark)
22	34	" (Double Quote)
23	35	# (Number Sign)
24	36	\$ (Dollar Sign)
25	37	% (Percent)
26	38	& (Ampersand)
27	39	` (Single Quote)
28	40	( (Right / Closing Parenthesis)
29	41	) (Right / Closing Parenthesis)
2a	42	* (Asterisk)
2b	43	+ (Plus)
2c	44	,
2d	45	- (Minus / Dash)
2e	46	.
2f	47	/ (Forward Slash)
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3a	58	: (Colon)
3b	59	; (Semi-colon)
3c	60	< (Less Than)
3d	61	= (Equal Sign)
3e	62	> (Greater Than)
3f	63	? (Question Mark)
40	64	@ (AT Symbol)
41	65	A
42	66	B
43	67	C
44	68	D



45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4a	74	J
4b	75	K
4c	76	L
4d	77	M
4e	78	N
4f	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5a	90	Z
5b	91	[ ( Left / Opening Bracket )
5c	92	\ ( Back Slash )
5d	93	] ( Right / Closing Bracket )
5e	94	^ ( Caret / Circumflex )
5f	95	_ ( Underscore )
60	96	' ( Grave Accent )
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h

---

69	105	i
6a	106	j
6b	107	k
6c	108	l
6d	109	m
6e	110	n
6f	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7a	122	z
7b	123	{ (Left/ Opening Brace)
7c	124	(Vertical Bar)
7d	125	} (Right/Closing Brace)
7e	126	~ (Tilde)
7f	127	DEL (Delete)

---

## 键盘按键序号

6E	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	•	•	•		
01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0F	4B	50	55	5A	5F	64	69
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	4C	51	56	5B	60	65	6A
1E	1F	20	21	22	23	24	25	26	27	28	29	2B					5C	61	66	
2C	2E	2F	30	31	32	33	34	35	36	37	39			53			5D	62	67	6C
3A	3B	3C	3D				3E	3F	38	40			4F	54	59		63	68		

104 键美式键盘

6E	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	•	•	•		
01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0F	4B	50	55	5A	5F	64	69
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	2B	4C	51	56	5B	60	65	6A
1E	1F	20	21	22	23	24	25	26	27	28	29	1D					5C	61	66	
2C	2D	2E	2F	30	31	32	33	34	35	36	37	39		53			5D	62	67	6C
3A	3B	3C	3D				3E	3F	38	40			4F	54	59		63	68		

105 键欧式键盘