



NLS-EM1365-EX
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、反向工程、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。	2018-10-12

目 录

第一章 开始	1
简介	1
关于本手册	2
连接 EVK 与 PC	2
条码识读操作	2
设置 EM1365-EX.....	3
使用串口指令	3
读寄存器指令	3
写寄存器指令	6
EEPROM 写使能开启/关闭命令.....	9
EEPROM 进行写命令	11
EEPROM 进行读命令	14
寄存器列表	17
使用设置条码	34
开启/关闭设置码	34
发送设置码信息	34
恢复出厂默认	35
第二章 识读模式	36
手动识读模式	36
连续识读模式	37
感应识读模式	40
命令触发识读模式.....	44
第三章 提示发送	45
识读成功提示音设置.....	45
解码状态提示符	46
其他设置	47
静音设置	47
照明设置	47
第四章 通讯设置	48
串行通讯接口	49
波特率	49
校验	51

停止位.....	51
数据位.....	52
USB 接口.....	54
USB HID-KBW	54
USB DATAPIPE.....	64
USB COM Port Emulation.....	64
HID-POS	65

第五章 数据编辑67

简介	67
前缀顺序设置	67
自定义前缀	68
禁止或允许添加自定义前缀	68
设置自定义前缀	68
AIM ID 前缀	69
CODE ID 前缀	70
Code ID 出厂默认值	70
修改 Code ID	70
自定义后缀	74
禁止或允许添加自定义后缀	74
设置自定义后缀	75
结束符后缀	76
禁止或允许添加结束符后缀	76
设置结束符后缀	77

第七章 条码参数设置78

简介	78
全局设置	78
允许/禁止识读所有条码.....	78
Code 128.....	79
恢复出厂默认值	79
允许/禁止识读 Code 128	79
设置读码长度限制	80
UCC/EAN-128.....	81
恢复出厂默认值	81
允许/禁止识读 UCC/EAN-128.....	81
设置读码长度限制	82
AIM 128.....	83
恢复出厂默认值	83

允许/禁止识读 AIM 128	83
设置读码长度限制	84
EAN-8.....	85
恢复出厂默认值	85
允许/禁止识读 EAN-8	85
设置是否发送校验位	85
设置是否允许识读附加码	86
设置是否必须有附加码	87
扩展设置	87
EAN-13.....	88
恢复出厂默认值	88
允许/禁止识读 EAN-13	88
设置是否发送校验位	88
设置是否允许识读附加码	89
设置是否必须有附加码	90
ISSN.....	91
恢复出厂默认值	91
允许/禁止识读 ISSN	91
ISBN	92
恢复出厂默认值	92
允许/禁止识读 ISBN.....	92
ISBN 数据位设置	92
UPC-E.....	93
恢复出厂默认值	93
允许/禁止识读 UPC-E	93
设置是否发送校验位	93
设置是否允许识读附加码	94
设置是否必须有附加码	95
设置是否发送系统字符	95
扩展设置	96
UPC-A	97
恢复出厂默认值	97
允许/禁止识读 UPC-A.....	97
设置是否发送校验位	98
前缀字符发送设置	98
设置是否允许识读附加码	99
设置是否必须有附加码	100
Interleaved 2 of 5	101
恢复出厂默认值	101

允许/禁止识读 Interleaved 2 of 5.....	101
校验设置.....	102
设置读码长度限制	103
ITF-6	104
恢复出厂默认值	104
允许/禁止识读 ITF-6	104
ITF-14	105
恢复出厂默认值	105
允许/禁止识读 ITF-14	105
Deutsche 14.....	106
恢复出厂默认值	106
允许/禁止识读 Deutsche 14.....	106
Deutsche 12.....	107
恢复出厂默认值	107
允许/禁止识读 Deutsche 12	107
Matrix 2 of 5 (European Matrix 2 of 5)	108
恢复出厂默认值	108
允许/禁止识读 Matrix 2 of 5.....	108
校验设置.....	109
设置读码长度限制	110
Industrial 25	111
恢复出厂默认值	111
允许/禁止识读 Industrial 25.....	111
校验设置.....	112
设置读码长度限制	113
Standard 25	114
恢复出厂默认值	114
允许/禁止识读 Standard 25.....	114
校验设置.....	115
设置读码长度限制	116
Code 39.....	117
恢复出厂默认值	117
允许/禁止识读 Code 39	117
校验设置.....	118
设置是否发送起始符和终止符	119
Full ASCII 支持	119
设置读码长度限制	120
Codabar.....	121
恢复出厂默认值	121

允许/禁止识读 Codabar	121
校验设置	122
起始符和终止符设置	123
设置读码长度限制	124
Code 93	125
恢复出厂默认值	125
允许/禁止识读 Code 93	125
校验设置	126
设置读码长度限制	127
Code 11	128
恢复出厂默认值	128
允许/禁止识读 Code 11	128
校验设置	129
设置读码长度限制	130
Plessey	131
恢复出厂默认值	131
允许/禁止识读 Plessey	131
校验设置	132
设置读码长度限制	133
MSI-Plessey	134
恢复出厂默认值	134
允许/禁止识读 MSI-Plessey	134
校验设置	135
设置读码长度限制	136
RSS-14	137
恢复出厂默认值	137
允许/禁止识读 RSS-14	137
设置是否发送 AI 字符	137
RSS-Limited	138
恢复出厂默认值	138
允许/禁止识读 RSS-Limited	138
设置是否发送 AI 字符	138
RSS-Expand	139
恢复出厂默认值	139
允许/禁止识读 RSS-Expand	139

附录	140
----------	-----

默认设置表	140
AIM ID 列表	146

Code ID 列表..... 148

ASCII 码表 149

数据码 153

保存或取消 156

F1~F12..... 157



第一章 开始

简介

EM1365-EX 条码识读引擎，应用了国际领先的芯片化新大陆**UIMC**智能图像识别技术，开创一维条码识读引擎的新时代。

新大陆的一维解码芯片，将先进的**UIMC**图像识别算法与先进的芯片设计与制造技术完美融合，极其简化了一维条码识读产品的设计难度，为自动识别行业树立高性能、高可靠、低功耗的优秀标杆。

EM1365-EX 可识读各类主流一维条码，包括如下：EAN-13, EAN-8, UPC-A, UPC-E, ISSN, ISBN, Codabar, Code 128, Code 93, ITF-6, ITF-14, Interleaved 2 of 5, Industrial 2 of 5, Standard 2 of 5, Matrix 2 of 5, GS1 Databar, Code 39, Code 11, MSI-Plessey, Plessey。

EM1365-EX 小巧的设计，可嵌入于绝大多数的设备，仅仅 5.7 克的重量，几乎感觉不到它的存在。EM1365-EX 的瞬间启动功能，可允许在应用中彻底关断电流，在需要时瞬间启动进行条码识别，并以超低的工作功耗为应用带来便利和迅捷。





W010F01

** 开启设置码

关于本手册

本手册主要提供了 EM1365-EX 识读引擎的各种功能设置指令。通过扫描本手册中的设置条码或发送指令，可以更改 EM1365-EX 的功能参数如通讯接口参数、识读模式、提示方式等。

EM1365-EX 产品在出厂时已经提供了适合大多数通常应用功能的参数配置，大多数情况下用户无需做调整就可以投入使用，在本手册的附录中，列出了 EM1365-EX 的默认功能和参数，可供参考。设置码中标有“**”的选项，同样表示了默认的功能或参数。

连接 EVK 与 PC

使用辅助工具 EVK 可配套 EM1365-EX 产品的快速应用开发。用户可使用同面或异面 12-pin 柔性线缆将 EM1365-EX 安装于 EVK 上，连接 EVK 至 PC 可选择使用 USB 连接或 RS-232 连接。安装驱动后，EVK 可通过虚拟串口直接与 EM1365-EX 通讯和接收识读数据。

条码识读操作

EM1365-EX 得益于一维影像技术及新大陆优秀的 **UIMG** 技术，可以非常容易且准确地识读条码符号，即使条码倾斜 $\pm 65^\circ$ ，偏转 $\pm 60^\circ$ ，仍可具有良好的识读效果。在识读时，将 EM1365-EX 投射出的瞄准指示光束瞄准于所需读取的条码符号上即可。



W010F00

关闭设置码



设置 EM1365-EX

用户可使用设置条码或串口指令对识读引擎进行设置。

使用串口指令

用户可从主机发送串口指令对识读引擎进行设置。识读引擎与主机设备间必须在通讯参数配置上完全匹配才能实现正常通讯。识读引擎默认的串行通讯参数：波特率 9600bps，无校验，8 位数据位，1 位停止位，无流控。识读引擎使用 8 位寄存器。

读寄存器指令

对于设备寄存器的读操作最多可一次读取 256 个连续寄存器。

命令格式：

输入：{Prefix1} {Types} {Lens} {Address} {Datas} {FCS}

其中 Prefix1 : 0x7E 0x00 (2 bytes)

Types : 0x07 (1 byte)

Lens : 表示该命令中 Datas 的字节数，这里建议设置 0x01

Address: 0x0000~0xFFFF (2 bytes)，表示要读取的寄存器的起始地址

Datas : 0x00~0xFF (1 byte)，表示要连续读取的寄存器的个数，0x00 表示 256 个

FCS : CRC_CCITT 校验值 (2 bytes)。计算的范围：Types、Lens、Address、Datas 计算的方法为 CRC_CCITT，特征多项式： $X^{16}+X^{12}+X^5+1$ ，即多项式系数为 0x1021，初始值为全 0，对于单个字节来说最高位先计算，不需要取反直接发送。C 的参考代码如下：





W010F01

** 开启设置码

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i != 0; i /= 2)
        {
            crc *= 2;
            if((crc&0x10000) != 0) //上一位 CRC 乘 2 后，若首位是 1，则除以 0x11021
                crc ^= 0x11021;
            if((*ptr&i) != 0) //如果本位是 1，那么 CRC = 上一位的 CRC + 本位/CRC_CCITT
                crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```

返回：{Prefix2} {Types} {Lens} {Datas} {FCS}

1) 读成功并返回读数据

其中 Prefix2 : 0x02 0x00

Types : 0x00 (读成功)

Lens : 表示上传的 Datas 的字节个数，0x00 表示 256 个

Datas : 0x00~0xFF，表示读上来的数据

FCS : CRC_CCITT 校验值。

2) 下发 FCS 校验失败

其中 Prefix2 : 0x02 0x00

Types : 0x01 (下发 FCS 校验失败)

Lens : 0x01

Datas : 0x00，无实际意义 (1 个字节)

FCS : 0x04 0x01 (CRC_CCITT 校验值)



W010F00

关闭设置码



W010F01

** 开启设置码

3) 未知命令失败

检测到 0x7e 0x00 开头的数据包，但内容长度不够或类型不对。

其中 Prefix2 : 0x02 0x00

Types : 0x03 (未知命令失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x6A 0x61 (CRC_CCITT 校验值)

示例:

对起始地址为 0x0005 连续 3 个地址的寄存器进行读操作。

1) 读成功并返回数据，返回的数据为 0x35 0x36 0x37

输入: 0x7e 0x00 0x07 0x01 0x00 0x05 0x03 0xde 0xf6

返回: 0x02 0x00 0x00 0x03 0x35 0x36 0x37 0x2a 0xba

2) 下发的 FCS 错误

输入: 0x7e 0x00 0x07 0x01 0x00 0x05 0x03 0x33 0x34

返回: 0x02 0x00 0x01 0x01 0x00 0x04 0x01

3) 未知命令失败

输入: 0x7e 0x00 0x07 0x01 0x00 0x05 0x03 0x33

返回: 0x02 0x00 0x03 0x01 0x00 0x6A 0x61



W010F00

关闭设置码



W010F01

** 开启设置码

写寄存器指令

对于设备寄存器的写操作最多可一次写入 256 个连续寄存器。

命令格式：

输入：{Prefix1} {Types} {Lens} {Address} {Datas} {FCS}

其中 Prefix1 : 0x7E 0x00 (2 bytes)

Types : 0x08 (1 byte)

Lens : 0x00~0xFF (1 byte)，表示该命令中 Datas 字段的字节数，同时也表示要进行连续写操作的次数，而 0x00 表示有 256 个字节

Address : 0x0000~0xFFFF (2 bytes)，表示要写入的寄存器的起始地址

Datas : 0x00~0xFF (1~256 bytes)，表示写入寄存器的数据

FCS : CRC_CCITT 校验值 (2 bytes)。计算的范围：Types、Lens、Address、Datas 计算的方法为 CRC_CCITT，特征多项式： $X^{16}+X^{12}+X^5+1$ ，即多项式系数为 0x1021，初始值为全 0，对于单个字节来说最高位先计算，不需要取反直接发送。C 的参考代码如下：

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i != 0; i /= 2)
        {
            crc *= 2;
            if((crc&0x10000) != 0) //上一位 CRC 乘 2 后，若首位是 1，则除以 0x11021
                crc ^= 0x11021;
            if((*ptr&i) != 0) //如果本位是 1，那么 CRC = 上一位的 CRC + 本位/CRC_CCITT
                crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```



W010F00

关闭设置码



W010F01

** 开启设置码

返回: {Prefix2} {Types} {Lens} {Datas} {FCS}

1) 写成功

其中 Prefix2 : 0x02 0x00

Types : 0x00 (写成功)

Lens : 0x01

Datas : 0x00

FCS : 0x33 0x31 (CRC_CCITT 校验值)

2) 下发 FCS 校验失败

其中 Prefix2 : 0x02 0x00

Types : 0x01 (下发 FCS 校验失败)

Lens : 0x01

Datas : 0x00

FCS : 0x04 0x01 (CRC_CCITT 校验值)

3) 未知命令失败

检测到 0x7e 0x00 开头的数据包, 但内容长度不够或类型不对。

其中 Prefix2 : 0x02 0x00

Types : 0x03 (未知命令失败)

Lens : 0x01

Datas : 0x00

FCS : 0x6A 0x61 (CRC_CCITT 校验值)



W010F00

关闭设置码



W010F01

** 开启设置码

示例：

向起始地址为 0x000a 连续 4 个地址的寄存器写入 31、32、33、34。

1) 设置成功

输入：0x7e 0x00 0x08 0x04 0x00 0x26 0x31 0x32 0x33 0x34 0xcd 0xa4

返回：0x02 0x00 0x00 0x01 0x00 0x33 0x31

2) 下发的 FCS 错误

输入：0x7e 0x00 0x08 0x04 0x00 0x26 0x31 0x32 0x33 0x34 0x33 0x34

返回：0x02 0x00 0x01 0x01 0x00 0x04 0x01

3) 未知命令失败

输入：0x7e 0x00 0x08 0x04 0x00 0x26 0x31 0x32 0x33 0x34 0x33

返回：0x02 0x00 0x03 0x01 0x00 0x6A 0x61



W010F00

关闭设置码



W010F01

** 开启设置码

EEPROM 写使能开启/关闭命令

EEPROM 默认在上电后写使能关闭，对 EEPROM 进行写操作之前需先将写使能开启，写完之后最好能将写使能关闭，以防止误写。

命令格式：

输入： {Prefix1} {Types} {Lens} {Address} {Datas} {FCS}

其中 Prefix1 : 0x7E 0x00

Types : 0x03 (开启) / 0x00 (关闭)

Lens : 表示该命令中 Datas 的字节数，这里建议设置 0x01

Address: 这里无实际意义

Datas : 这里无实际意义

FCS : CRC_CCITT 校验值 (2 bytes)。计算的范围: Types、Lens、Address、Datas 计算的方法为 CRC_CCITT，特征多项式: $X^{16}+X^{12}+X^5+1$ ，即多项式系数为 0x1021，初始值为全 0，对于单个字节来说最高位先计算，不需要取反直接发送。C 的参考代码如下：

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i != 0; i /= 2)
        {
            crc *= 2;
            if((crc&0x10000) != 0) //上一位 CRC 乘 2 后，若首位是 1，则除以 0x11021
                crc ^= 0x11021;
            if((*ptr&i) != 0) //如果本位是 1，那么 CRC = 上一位的 CRC + 本位/CRC_CCITT
                crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```



W010F00

关闭设置码



W010F01

** 开启设置码

返回: {Prefix2} {Types} {Lens} {Datas} {FCS}

1) 写成功

其中 Prefix2 : 0x02 0x00

Types : 0x00 (写成功)

Lens : 0x01

Datas : 0x00 (1 个字节)

FCS : 0x33 0x31 (CRC_CCITT 校验值)

2) 下发 FCS 校验失败

其中 Prefix2: 0x02 0x00

Types : 0x01 (下发 FCS 校验失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x04 0x01 (CRC_CCITT 校验值)

3) 未知命令失败

检测到 0x7e 0x00 开头的数据包, 但内容长度不够或类型不对。

其中 Prefix2 : 0x02 0x00

Types : 0x03 (未知命令失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x6A 0x61 (CRC_CCITT 校验值)



W010F00

关闭设置码



W010F01

** 开启设置码

EEPROM 进行写命令

此操作和对寄存器的写操作基本一样，EEPROM 的地址长度为 512，可最多支持 256 个地址的连续写操作，写完成一个后内部地址自动加 1，有所差别的是在写之前要将写使能开启，写完之后最好将写使能关闭。但如果没有开启就进行写操作，也不会返回错误，所以写完之后需要读回进行校验，以保证正确性。

命令格式：

输入：{Prefix1} {Types} {Lens} {Address} {Datas} {FCS}

其中 Prefix1 : 0x7E 0x00

Types : 0x04

Lens : 0x00~0xff, 表示该命令中 Datas 字段的字节数，0x00 表示 256 个字节

Address: 0x0000~0xFFFF, 表示需要写入的起始地址

Datas : 0x00~0xff, 表示待写入寄存器的数据

FCS : CRC_CCITT 校验值 (2 bytes)。计算的范围: Types、Lens、Address、Datas 计算的方法为 CRC_CCITT, 特征多项式: $X^{16}+X^{12}+X^5+1$, 即多项式系数为 0x1021, 初始值为全 0, 对于单个字节来说最高位先计算, 不需要取反直接发送。C 的参考代码如下:

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i != 0; i /= 2)
        {
            crc *= 2;
            if((crc & 0x10000) != 0) //上一位 CRC 乘 2 后, 若首位是 1, 则除以 0x11021
                crc ^= 0x11021;
            if((*ptr & i) != 0) //如果本位是 1, 那么 CRC = 上一位的 CRC + 本位/CRC_CCITT
                crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```



W010F00

关闭设置码



W010F01

** 开启设置码

返回: {Prefix2} {Types} {Lens} {Datas} {FCS}

1) 写成功

其中 Prefix2 : 0x02 0x00

Types : 0x00 (写成功)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x33 0x31 (CRC_CCITT 校验值)

2) 下发 FCS 校验失败

其中 Prefix2 : 0x02 0x00

Types : 0x01 (下发 FCS 校验失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x04 0x01 (CRC_CCITT 校验值)

3) 写失败, 即 EEPROM 无应答

其中 Prefix2 : 0x02 0x00

Types : 0x02 (写失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x5d 0x51 (CRC_CCITT 校验值)

4) 未知命令失败

检测到 0x7e 0x00 开头的数据包, 但内容长度不够或类型不对。

其中 Prefix2 : 0x02 0x00

Types : 0x03 (未知命令失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x6A 0x61 (CRC_CCITT 校验值)



W010F00

关闭设置码



W010F01

** 开启设置码

示例：

对 EEPROM 中地址为 0x0000 的连续 4 个地址写入 51、52、53、54。

1) 设置成功

输入：0x7e 0x00 0x04 0x04 0x00 0x00 0x51 0x52 0x53 0x54 0xbc 0x17

返回：0x02 0x00 0x00 0x01 0x00 0x33 0x31

2) 下发的 FCS 错误

输入：0x7e 0x00 0x04 0x03 0x00 0x00 0x51 0x52 0x53 0x54 0x33 0x34

返回：0x02 0x00 0x01 0x01 0x00 0x04 0x01



W010F00

关闭设置码



W010F01

** 开启设置码

EEPROM 进行读命令

此操作和对寄存器的读操作一样，EEPROM 的地址长度为 512，可最多支持 256 个地址的连续读操作，读成一个后内部地址自动加 1。

命令格式：

输入：{Prefix1} {Types} {Lens} {Address} {Datas} {FCS}

其中 Prefix1 : 0x7E 0x00

Types : 0x05

Lens : 表示该命令中 Datas 的字节数，这里建议设置 0x01，0x00 表示 256 个字节

Address: 0x0000~0xFFFF，表示需要读出数据的起始地址

Datas : 0x00~0xff，第一个字节数据表示需要读出寄存器数据的字节数，最多支持 256 个字节连续读。其他字节无意义

FCS : CRC_CCITT 校验值（2 bytes）。计算的范围：Types、Lens、Address、Datas 计算的方法为 CRC_CCITT，特征多项式： $X^{16}+X^{12}+X^5+1$ ，即多项式系数为 0x1021，初始值为全 0，对于单个字节来说最高位先计算，不需要取反直接发送。C 的参考代码如下：

```
unsigned int crc_cal_by_bit(unsigned char* ptr, unsigned int len)
{
    unsigned int crc = 0;
    while(len-- != 0)
    {
        for(unsigned char i = 0x80; i != 0; i /= 2)
        {
            crc *= 2;
            if((crc&0x10000) != 0) //上一位 CRC 乘 2 后，若首位是 1，则除以 0x11021
                crc ^= 0x11021;
            if((*ptr&i) != 0) //如果本位是 1，那么 CRC = 上一位的 CRC + 本位/CRC_CCITT
                crc ^= 0x1021;
        }
        ptr++;
    }
    return crc;
}
```



W010F00

关闭设置码



W010F01

** 开启设置码

返回: {Prefix2} {Types} {Lens} {Datas} {FCS}

1) 读成功并返回读数据

其中 Prefix2 : 0x02 0x00

Types : 0x00 (读成功)

Lens : 表示上传的 Datas 的字节个数

Datas : 0x00~0xff, 表示读上来的数据

FCS : CRC_CCITT 校验值

2) 下发 FCS 校验失败

其中 Prefix2: 0x02 0x00

Types : 0x01 (下发 FCS 校验失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x04 0x01 (CRC_CCITT 校验值)

3) 写失败, 即 EEPROM 无应答

其中 Prefix2 : 0x02 0x00

Types : 0x02(写失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x5d 0x51 (CRC_CCITT 校验值)

4) 未知命令失败

检测到 0x7e 0x00 开头的数据包, 但内容长度不够或类型不对。

其中 Prefix2: 0x02 0x00

Types : 0x03 (未知命令失败)

Lens : 0x01

Datas : 0x00, 无实际意义 (1 个字节)

FCS : 0x6A 0x61 (CRC_CCITT 校验值)



W010F00

关闭设置码



W010F01

** 开启设置码

示例:

对 EEPROM 中地址为 0x0005 的连续 3 个地址进行读操作

1) 读成功并返回数据, 返回的数据为 0x35 0x36 0x37

输入: 0x7e 0x00 0x05 0x01 0x00 0x05 0x03 0x9a 0x75

返回: 0x02 0x00 0x00 0x03 0x35 0x36 0x37 0x2a 0xba

2) 下发的 FCS 错误

输入: 0x7e 0x00 0x05 0x01 0x00 0x05 0x03 0x33 0x34

返回: 0x02 0x00 0x01 0x01 0x00 0x04 0x01



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器列表

寄存器	0x0000
数据位	功能
Bit 7	保留
Bit 6	1: 关闭静音 0: 启动静音
Bit 5-4	保留
Bit 3-2	照明设置: 00: 无照明 01: 普通 10/11: 常亮
Bit 1-0	识读模式设置: 00: 手动模式 01: 命令触发模式 10: 连续模式 11: 感应模式
寄存器	0x0003
数据位	功能
Bit 7-0	灵敏度设置: 0x00~0xFF: 0-255 级。数值越小, 灵敏度越高。
寄存器	0x0004
数据位	功能
Bit 7-0	稳像时长设置: 0x00-0xFF: 0.0-25.5 秒
寄存器	0x0005
数据位	功能
Bit 7-0	识读间隔时长设置: 0x00-0xFF: 0.0-25.5 秒
寄存器	0x0006
数据位	功能
Bit 7-0	单次读码时长设置: 0x00: 无限长; 0x01-0xFF: 1-255 秒
寄存器	0x0007
数据位	功能
Bit 7-0	相同读码延时设置: 0x00: 无限长; 0x01-0xFF: 0.1-25.5 秒



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0009
数据位	功能
Bit 7-5	保留
Bit 4	0: 串口通讯
Bit 3-2	保留
Bit 1-0	00: USB DATAPIPE 01: USB HID-KBW 10: USB 虚拟串口 11: USB HID-POS
寄存器	0x000A
数据位	功能
Bit 7-5	保留
Bit 4	0: 相同读码不延时 1: 要求相同读码延时
Bit 3-2	保留
Bit 1-0	安全级别设置: 00-03: 0-3 级。数值越高, 误码率和效率越低。
寄存器	0x000F
数据位	功能
Bit 7-3	保留
Bit 2-1	00: 不发送设置码信息 11: 发送设置码信息
Bit 0	0: 关闭设置码 1: 启动设置码
寄存器	0x0010
数据位	功能
Bit 7-0	识读成功提示音频率设置: 0xDA: 低频 0x4B: 中频 0x25: 高频
寄存器	0x0011
数据位	功能
Bit 7-0	识读成功提示音持续时间设置: 0x1F: 40ms 0x3E: 80ms 0x5D: 120ms
寄存器	0x0012
数据位	功能
Bit 7-3	保留
Bit 2	1: 开启识读成功提示音 0: 关闭识读成功提示音
Bit 1-0	保留



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0019	
数据位	功能	
Bit 7-0	USB HID-KBW 下国家/语言键盘布局选择:	
	00: 美式键盘	01: 比利时
	03: 加拿大	04: 捷克斯洛伐克
	06: 芬兰	07: 法国
	09: 希腊	0A: 匈牙利
	0C: 意大利	0D: 拉丁美洲
	0F: 挪威	10: 波兰
	12: 罗马尼亚	13: 俄罗斯
	16: 西班牙	17: 瑞典
	19: 土耳其 1	1A: 土耳其 2
	1C: 日本	1B: 英国
	02: 巴西	05: 丹麦
	08: 奥地利	0B: 以色列
	0E: 荷兰	11: 葡萄牙
	15: 斯洛伐克	18: 瑞士
寄存器	0x001A	
数据位	功能	
Bit 7-6	键间延时设定:	
	00: 不延时	01: 延时 5ms
	10: 延时 10ms	11: 延时 15ms
Bit 5-3	强制字母大小写转换:	
	000: 不转换	001: 大小写反转
	100/101: 全转换成大写字符	110/111: 全转换成小写字符
Bit 2	1: 模拟数字小键盘	0: 不模拟数字小键盘
Bit 1-0	00: 标准键盘输入模式	01: 键盘仿真输入控制字符模式
	10/11: 键盘仿真输入字符模式	



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0029	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 8 个数据位	0: 7 个数据位
Bit 2-1	校验: 00/01: 无校验 10: 奇校验 11: 偶校验	
Bit 0	1: 2 个停止位	0: 1 个停止位
寄存器	0x002B, 0x2A	
数据位	功能	
Bit 15-13	保留	
Bit 12-0	0x09C4: 串口波特率为 1200 bps 0x04E2: 串口波特率为 2400 bps 0x0271: 串口波特率为 4800 bps 0x0139: 串口波特率为 9600 bps 0x00D0: 串口波特率为 14400 bps 0x009C: 串口波特率为 19200 bps 0x004E: 串口波特率为 38400 bps 0x0034: 串口波特率为 57600 bps 0x001A: 串口波特率为 115200 bps	
寄存器	0x0031	
数据位	功能	
Bit 7-6	保留	
Bit 5	1: 发送解码状态提示符	0: 不发送解码状态提示符
Bit 4	1: 允许添加结束符后缀	0: 禁止添加结束符后缀
Bit 3	1: 允许添加自定义后缀	0: 禁止添加自定义后缀
Bit 2	1: 允许添加自定义前缀	0: 禁止添加自定义前缀
Bit 1	1: 允许添加 CODE ID 前缀	0: 禁止添加 CODE ID 前缀
Bit 0	前缀顺序设置: 1: 自定义前缀+Code ID+AIM ID 0: Code ID +自定义前缀+AIM ID	



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0037-0x33		
数据位	功能		
Bit 39-0	设置自定义前缀值		
寄存器	0x0042-0x3E		
数据位	功能		
Bit 39-0	设置自定义后缀值		
寄存器	0x004D-0x49		
数据位	功能		
Bit 39-0	设置结束符后缀值		
寄存器	0x0061		
数据位	功能		
Bit 7-1	保留		
Bit 0	1：允许识读 Code 128	0：禁止识读 Code 128	
寄存器	0x0062		
数据位	功能		
Bit 7-2	保留		
Bit 1-0	00：禁止识读 UCC/EAN 128	01：识别成 Code 128	10/11：允许识读 UCC/EAN 128
寄存器	0x0063		
数据位	功能		
Bit 7-2	保留		
Bit 1-0	00：禁止识读 AIM 128	01：识别成 Code 128	10/11：允许识读 AIM 128
寄存器	0x0065		
数据位	功能		
Bit 7-6	00：EAN-8 条码信息不扩展 10/11：EAN-8 条码信息及类型转换成 EAN-13		
Bit 5	1：允许识读 EAN-8 的 5 位附加码 0：不允许识读 EAN-8 的 5 位附加码		
Bit 4	1：允许识读 EAN-8 的 2 位附加码 0：不允许识读 EAN-8 的 2 位附加码		
Bit 3	1：EAN-8 必须有附加码 0：EAN-8 不要求有附加码		
Bit 2	1：发送 EAN-8 校验位 0：不发送 EAN-8 校验位		
Bit 1	保留		
Bit 0	1：允许识读 EAN-8 0：禁止识读 EAN-8		



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0066		
数据位	功能		
Bit 7-6	保留		
Bit 5	1: 允许识读 EAN-13 的 5 位附加码	0: 不允许识读 EAN-13 的 5 位附加码	
Bit 4	1: 允许识读 EAN-13 的 2 位附加码	0: 不允许识读 EAN-13 的 2 位附加码	
Bit 3	1: EAN-13 必须有附加码	0: EAN-13 不要求有附加码	
Bit 2	1: 发送 EAN-13 校验位	0: 不发送 EAN-13 校验位	
Bit 1	保留		
Bit 0	1: 允许识读 EAN-13	0: 禁止识读 EAN-13	
寄存器	0x0067		
数据位	功能		
Bit 7-2	保留		
Bit 1-0	00: 禁止识读 ISSN	01: 识别成 EAN-13	10/11: 允许识读 ISSN
寄存器	0x0068		
数据位	功能		
Bit 7-3	保留		
Bit 2	ISBN 格式设置: 1: ISBN-10 0: ISBN-13		
Bit 1-0	00: 禁止识读 ISBN	01: 识别成 EAN-13	10/11: 允许识读 ISBN
寄存器	0x0069		
数据位	功能		
Bit 7-6	00: UPC-E 条码信息不扩展 10/11: UPC-E 条码信息及类型转换成 UPC-A		
Bit 5	1: 允许识读 UPC-E 的 5 位附加码	0: 不允许识读 UPC-E 的 5 位附加码	
Bit 4	1: 允许识读 UPC-E 的 2 位附加码	0: 不允许识读 UPC-E 的 2 位附加码	
Bit 3	1: UPC-E 必须有附加码	0: UPC-E 不要求有附加码	
Bit 2	1: 发送 UPC-E 校验位	0: 不发送 UPC-E 校验位	
Bit 1	保留		
Bit 0	1: 允许识读 UPC-E	0: 禁止识读 UPC-E	



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x006A		
数据位	功能		
Bit 7-6	保留		
Bit 5-4	UPC-E 系统字符发送设置： 01：不发送系统字符		



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x006E	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 ITF-14 校验位	0: 不发送 ITF-14 校验位
Bit 2	保留	
Bit 1-0	00: 禁止识读 ITF-14	01: 识别成 Interleaved 2 of 5 10/11: 允许识读 ITF-14
寄存器	0x006F	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Deutsche 14 校验位	0: 不发送 Deutsche 14 校验位
Bit 2	保留	
Bit 1-0	00: 禁止识读 Deutsche 14	01: 识别成 Interleaved 2 of 5 10/11: 允许识读 Deutsche 14
寄存器	0x0070	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Deutsche 12 校验位	0: 不发送 Deutsche 12 校验位
Bit 2	保留	
Bit 1-0	00: 禁止识读 Deutsche 12	01: 识别成 Interleaved 2 of 5 10/11: 允许识读 Deutsche 12
寄存器	0x0071	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Matrix 2 of 5 校验位	0: 不发送 Matrix 2 of 5 校验位
Bit 2	1: Matrix 2 of 5 进行校验	0: Matrix 2 of 5 不进行校验
Bit 1	保留	
Bit 0	1: 允许识读 Matrix 2 of 5	0: 禁止识读 Matrix 2 of 5
寄存器	0x0072	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Industrial 25 校验位	0: 不发送 Industrial 25 校验位
Bit 2	1: Industrial 25 进行校验	0: Industrial 25 不进行校验
Bit 1	保留	
Bit 0	1: 允许识读 Industrial 25	0: 禁止识读 Industrial 25



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0073	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Standard 25 校验位	0: 不发送 Standard 25 校验位
Bit 2	1: Standard 25 进行校验	0: Standard 25 不进行校验
Bit 1	保留	
Bit 0	1: 允许识读 Standard 25	0: 禁止识读 Standard 25
寄存器	0x0074	
数据位	功能	
Bit 7-6	保留	
Bit 5	1: 开启 Code 39 Full ASCII	0: 关闭 Code 39 Full ASCII
Bit 4	1: 发送 Code 39 校验位	0: 不发送 Code 39 校验位
Bit 3	1: Code 39 进行校验	0: Code 39 不进行校验
Bit 2	1: 发送 Code 39 起始/终止符	0: 不发送 Code 39 起始/终止符
Bit 1	保留	
Bit 0	1: 允许识读 Code 39	0: 禁止识读 Code 39
寄存器	0x0075	
数据位	功能	
Bit 7	保留	
Bit 6	1: 发送 Codabar 校验位	0: 不发送 Codabar 校验位
Bit 5	1: Codabar 进行校验	0: Codabar 不进行校验
Bit 4-3	Codabar 起始/终止符格式: 00: ABCD/ABCD 10: abcd/abcd 01: ABCD/TN*E 11: abcd/tn*e	
Bit 2	1: 发送 Codabar 起始/终止符	0: 不发送 Codabar 起始/终止符
Bit 1	保留	
Bit 0	1: 允许识读 Codabar	0: 禁止识读 Codabar



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0076	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Code 93 校验位	0: 不发送 Code 93 校验位
Bit 2	1: Code 93 进行校验	0: Code 93 不进行校验
Bit 1	保留	
Bit 0	1: 允许识读 Code 93	0: 禁止识读 Code 93
寄存器	0x0077	
数据位	功能	
Bit 7-6	保留	
Bit 5	1: 发送 Code 11 校验位	0: 不发送 Code 11 校验位
Bit 4-2	Code 11 校验设置: 000: 不校验 001: 一位校验, MOD11 010: 两位校验, MOD11/MOD11 011: 两位校验, MOD11/MOD9 100: MOD11 单校验 (Len <= 11) ; MOD11/MOD11 双校验 (Len > 11) 101: MOD11 单校验 (Len <= 11) ; MOD11/MOD9 双校验 (Len > 11)	
Bit 1	保留	
Bit 0	1: 允许识读 Code 11	0: 禁止识读 Code 11
寄存器	0x0078	
数据位	功能	
Bit 7-4	保留	
Bit 3	1: 发送 Plessey 校验位	0: 不发送 Plessey 校验位
Bit 2	1: Plessey 进行校验	0: Plessey 不进行校验
Bit 1	保留	
Bit 0	1: 允许识读 Plessey	0: 禁止识读 Plessey



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0079	
数据位	功能	
Bit 7-5	保留	
Bit 4	1: 发送 MSI- Plessey 校验位	0: 不发送 MSI- Plessey 校验位
Bit 3-2	MSI-Plessey 校验设置: 00: 无校验 01: 一位校验, MOD10 10: 两位校验, MOD10/MOD10 11: 两位校验, MOD10/MOD11	
Bit 1	保留	
Bit 0	1: 允许识读 MSI-Plessey	0: 禁止识读 MSI-Plessey
寄存器	0x007A	
数据位	功能	
Bit 7-3	保留	
Bit 2	1: 发送 RSS-14 AI 字符	0: 不发送 RSS-14 AI 字符
Bit 1	保留	
Bit 0	1: 允许识读 RSS-14	0: 禁止识读 RSS-14
寄存器	0x007B	
数据位	功能	
Bit 7-3	保留	
Bit 2	1: 发送 RSS-Limited AI 字符	0: 不发送 RSS-Limited AI 字符
Bit 1	保留	
Bit 0	1: 允许识读 RSS-Limited	0: 禁止识读 RSS-Limited
寄存器	0x007C	
数据位	功能	
Bit 7-1	保留	
Bit 0	1: 允许识读 RSS-Expand	0: 禁止识读 RSS-Expand



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0080
数据位	功能
Bit 7-0	设置 Code 128 最大解码长度
寄存器	0x0081
数据位	功能
Bit 7-0	设置 Code 128 最小解码长度
寄存器	0x0082
数据位	功能
Bit 7-0	设置 UCC/EAN-128 最大解码长度
寄存器	0x0083
数据位	功能
Bit 7-0	设置 UCC/EAN-128 最小解码长度
寄存器	0x0084
数据位	功能
Bit 7-0	设置 AIM 128 最大解码长度
寄存器	0x0085
数据位	功能
Bit 7-0	设置 AIM 128 最小解码长度
寄存器	0x0086
数据位	功能
Bit 7-0	设置 Interleaved 2 of 5 最大解码长度
寄存器	0x0087
数据位	功能
Bit 7-0	设置 Interleaved 2 of 5 最小解码长度
寄存器	0x0088
数据位	功能
Bit 7-0	设置 Matrix 2 of 5 最大解码长度
寄存器	0x0089
数据位	功能
Bit 7-0	设置 Matrix 2 of 5 最小解码长度



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x008A
数据位	功能
Bit 7-0	设置 Industrial 25 最大解码长度
寄存器	0x008B
数据位	功能
Bit 7-0	设置 Industrial 25 最小解码长度
寄存器	0x008C
数据位	功能
Bit 7-0	设置 Standard 25 最大解码长度
寄存器	0x008D
数据位	功能
Bit 7-0	设置 Standard 25 最小解码长度
寄存器	0x008E
数据位	功能
Bit 7-0	设置 Code 39 最大解码长度
寄存器	0x008F
数据位	功能
Bit 7-0	设置 Code 39 最小解码长度
寄存器	0x0090
数据位	功能
Bit 7-0	设置 Codabar 最大解码长度
寄存器	0x0091
数据位	功能
Bit 7-0	设置 Codabar 最小解码长度
寄存器	0x0092
数据位	功能
Bit 7-0	设置 Code 93 最大解码长度
寄存器	0x0093
数据位	功能
Bit 7-0	设置 Code 93 最小解码长度



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x0094
数据位	功能
Bit 7-0	设置 Code 11 最大解码长度
寄存器	0x0095
数据位	功能
Bit 7-0	设置 Code 11 最小解码长度
寄存器	0x0096
数据位	功能
Bit 7-0	设置 Plessey 最大解码长度
寄存器	0x0097
数据位	功能
Bit 7-0	设置 Plessey 最小解码长度
寄存器	0x0098
数据位	功能
Bit 7-0	设置 MSI-Plessey 最大解码长度
寄存器	0x0099
数据位	功能
Bit 7-0	设置 MSI-Plessey 最小解码长度
寄存器	0x00A1, 0x00A0
数据位	功能
Bit 15-0	设置 Code 128 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00A3, 0x00A2
数据位	功能
Bit 15-0	设置 UCC/EAN-128 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00A5, 0x00A4
数据位	功能
Bit 15-0	设置 AIM 128 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x00A9, 0x00A8
数据位	功能
Bit 15-0	设置 EAN-8 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00AB, 0x00AA
数据位	功能
Bit 15-0	设置 EAN-13 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00AD, 0x00AC
数据位	功能
Bit 15-0	设置 ISSN Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00B1, 0x00B0
数据位	功能
Bit 15-0	设置 UPC-E Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00B3, 0x00B2
数据位	功能
Bit 15-0	设置 UPC-A Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00B5, 0x00B4
数据位	功能
Bit 15-0	设置 Interleaved 2 of 5 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00B7, 0x00B6
数据位	功能
Bit 15-0	设置 ITF-6 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x00B9, 0x00B8	
数据位	功能	
Bit 15-0	设置 ITF-14 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00BB, 0x00BA	
数据位	功能	
Bit 15-0	设置 Deutsche 14 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00BD, 0x00BC	
数据位	功能	
Bit 15-0	设置 Deutsche 12 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00BF, 0x00BE	
数据位	功能	
Bit 15-0	设置 Matrix 2 of 5 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00C1, 0x00C0	
数据位	功能	
Bit 15-0	设置 Industrial 25 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00C3, 0x00C2	
数据位	功能	
Bit 15-0	设置 Standard 25 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	
寄存器	0x00C5, 0x00C4	
数据位	功能	
Bit 15-0	设置 Code 39 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值	



W010F00

关闭设置码



W010F01

** 开启设置码

寄存器	0x00C7, 0x00C6
数据位	功能
Bit 15-0	设置 Codabar Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00C9, 0x00C8
数据位	功能
Bit 15-0	设置 Code 93 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00CB, 0x00CA
数据位	功能
Bit 15-0	设置 Code 11 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00CD, 0x00CC
数据位	功能
Bit 15-0	设置 Plessey Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00CF, 0x00CE
数据位	功能
Bit 15-0	设置 MSI-Plessey Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00D1, 0x00D0
数据位	功能
Bit 15-0	设置 RSS-14 Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00D3, 0x00D2
数据位	功能
Bit 15-0	设置 RSS-Limited Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值
寄存器	0x00D5, 0x00D4
数据位	功能
Bit 15-0	设置 RSS-Expand Code ID: 一个或两个英文字母（小写或大写）的 ASCII 值



W010F00

关闭设置码



W010F01

** 开启设置码

使用设置条码

识读引擎可通过识读一个或多个设置条码来设置选项和功能。下面的章节将会详细介绍用户可设置的参数和功能以及其对应的设置条码和设置命令。



开启/关闭设置码



W010F01

** 【开启设置码】



W010F00

【关闭设置码】

发送设置码信息



W060F00

** 【不发送设置码信息】



W060F06

【发送设置码信息】



W010F00

关闭设置码



W010F01

** 开启设置码

恢复出厂默认

所有识读引擎都有一个出厂的默认设置，读取“设置所有属性恢复出厂默认”条码，将使识读引擎的所有属性设置恢复成出厂状态。

在以下情况下您最有可能使用到此条码：

- 1、识读引擎设置出错，如无法识读条码。
- 2、您忘记了之前对识读引擎做过何种设置，而又不希望受之前的设置影响。
- 3、设置了识读引擎使用某项不常使用的功能，并使用完成后。



WFFD980

【恢复出厂默认】



W010F00

关闭设置码



W010F01

** 开启设置码

第二章 识读模式

手动识读模式

手动识读模式为默认识读模式。在此模式下，识读引擎在按下触发键后开始读码，在读码成功或松开触发键或读码时间结束后停止读码。



W030000

**【手动识读】

在手动识读模式下，识读引擎可以设置单次读码时长。读码时长可设置为 0~255s，默认为 15s。当设置为 0 时，表示无限长时间。



M00031D

【设置单次读码时长】

示例：设置单次读码时长为 5s

1. 读“开启设置码”
2. 读“设置单次读码时长”码
3. 读数据码“5”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



连续识读模式

设置完毕，按下触发按钮，识读引擎在一次读码完成后间隔一段时间后（可设置）会自动开始下一次读码；直到再次单击触发键，识读引擎停止读码。默认状态下，重复识读相同条码不延时。



在连续识读模式下，识读引擎可以设置单次读码时长。读码时长可设置为 0~255s，默认为 15s。当设置为 0 时，表示无限长时间。



示例：设置单次读码时长为 5s

1. 读“开启设置码”
2. 读“设置单次读码时长”码
3. 读数据码“5”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”





W010F01

** 开启设置码

在连续识读模式下，识读引擎还可以设置识读间隔时长，该参数以 100ms 为单位，可设置为 0~25.5s，默认为 1.0s。



M00031C

【设置识读间隔时长】

示例：设置识读间隔时长为 5s

1. 读“开启设置码”
2. 读“设置识读间隔时长”码
3. 读数据码“5”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



在连续识读模式下，选择“相同读码不延时”，识读引擎在一次读码完成后会自动开始下一次读码；直到再次单击触发键，识读引擎停止读码。

选择“要求相同读码延时”，识读引擎在一次读码完成后会自动开始下一次读码，若条码与上一次被成功识读的条码完全一样，识读引擎会处于持续等待状态，直到相同读码延时结束后，方能成功解码。当条码不重复时，识读引擎会一直读码，直到再次单击触发键，识读引擎停止读码。

相同读码延时以 100ms 为单位，可设置为 0.1~25.5s；设置为 0 表示超时无限长，即不允许重复识读相同条码。



示例：设置相同读码延时为 5s

1. 读“开启设置码”
2. 读“设置相同读码延时”码
3. 读数据码“5”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”





W010F01

** 开启设置码

感应识读模式

设置完毕，无需触发，识读引擎即开始侦测窗口前环境的变化。读码完成后停止并处于监测状态等待下一次环境亮度的改变。此模式下单击触发键也可以启动读码。设备对周围环境感应的灵敏度可以设置。默认状态下，重复识读相同条码不延时。



W030003

【感应识读】

在感应识读模式下，识读引擎可以设置单次读码时长，读码时长可设置为 0~255s，默认为 15s。当设置为 0 时，表示无限长时间。



M00031D

【设置单次读码时长】

示例：设置单次读码时长为 5s

1. 读“开启设置码”
2. 读“设置单次读码时长”码
3. 读数据码“5”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

在感应识读模式下，识读引擎还可以设置稳像时长。该参数以 100ms 为单位，可设置为 0~25.5s，默认为 0.4s。



M00031B

【设置稳像时长】

示例：设置稳像时长为 5s

1. 读“开启设置码”
2. 读“设置稳像时长”码
3. 读数据码“5”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



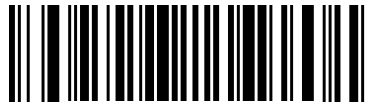
W010F01

** 开启设置码

在感应识读模式下，选择“相同读码不延时”，识读引擎在一次读码完成后会自动开始下一次读码；直到再次单击触发键，识读引擎停止读码。

选择“要求相同读码延时”，识读引擎在一次读码完成后会自动开始下一次读码，若条码与上一次被成功识读的条码完全一样，识读引擎会处于持续等待状态，直到相同读码延时结束后，方能成功解码。当条码不重复时，识读引擎会一直读码，直到再次单击触发键，识读引擎停止读码。

相同读码延时以 100ms 为单位，可设置为 0.1~25.5s，默认为 3.0s。设置为 0 表示超时无限长，即不允许重复识读相同条码。



M00031E

【设置相同读码延时】



W100A00

** 【相同读码不延时】



W100A10

【要求相同读码延时】

示例：设置相同读码延时为 5s

1. 读“开启设置码”
2. 读“设置相同读码延时”码
3. 读数据码“5”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

灵敏度是感应读码模式下识读引擎对周围环境变化剧烈程度的感应程度。用户可以根据实际使用环境来选择灵敏度，以提高读码的效率。灵敏度越高，启动识读所需的场景变化越小；反之灵敏度越低，启动识读所需的场景变化越大。



WFF0305

【高灵敏度】



WFF0310

**【中灵敏度】



WFF0330

【低灵敏度】



M00031A

【自定义灵敏度】

感应模式下灵敏度可设置 0-255 等级，数值越小，灵敏度级别越高。

示例：设置灵敏度级别为 10

1. 读“开启设置码”
2. 读“自定义灵敏度”码
3. 读数据码“1”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

命令触发识读模式

此模式状态下，需要上位机控制触发识读引擎进行解码。



W030001

【命令触发识读】

在命令触发识读模式下，识读引擎可以设置单次读码时长，读码时长可设置为 0~255s，默认为 15s。当设置为 0 时，表示无限长时间。



M00031D

【设置单次读码时长】

示例：设置单次读码时长为 5s

1. 读“开启设置码”
2. 读“设置单次读码时长”码
3. 读数据码“5”（见附录）
4. 读“保存”码（见附录）
5. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

第三章 提示发送

识读成功提示音设置



W041204

**【开启识读成功提示音】



W041200

【关闭识读成功提示音】



WFF10DA

【低频】



WFF104B

**【中频】



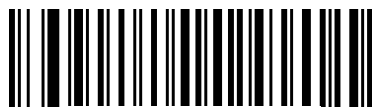
WFF1025

【高频】



WFF111F

【音长 40ms】



WFF113E

**【音长 80ms】



WFF115D

【音长 120ms】



W010F00

关闭设置码



W010F01

** 开启设置码

解码状态提示符

为了让主机能快速了解当前解码是否成功，可以开启此功能。

读取“发送解码状态提示符”开启此功能后，若识读不成功，识读引擎会发送提示字符“F”；若识别成功则在解码数据前添加提示字符“S”。



W203120

【发送解码状态提示符】



W203100

** 【不发送解码状态提示符】



W010F00

关闭设置码



其他设置

以下参数设置为临时设置。在设备重启或关机后，这些参数将自动恢复为出厂默认设置。

静音设置



提示：该设置仅对识读成功提示音有效。

照明设置





W010F01

** 开启设置码

第四章 通讯设置

EM1365-EX 识读引擎提供 TTL-232 串行通讯接口与主机进行通讯连接。经由通讯接口，可以接收识读数据、对识读引擎发出指令进行控制，以及更改识读引擎的功能参数等。



W010F00

关闭设置码



W010F01

** 开启设置码

串行通讯接口

波特率

当识读引擎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，需要设置通讯时的波特率（即传输速率）。

波特率是 232 数据通讯时每秒传输的位数（8 位一个字节），识读引擎和数据接收主机所使用波特率须保持一致才能保证数据传输准确。识读引擎支持以下波特率，单位是 bit/s。



WFFD9D3

** 【9600】



WFFD9D5

【19200】



WFFD9D0

【1200】



WFFD9D6

【38400】



WFFD9D1

【2400】



WFFD9D7

【57600】



W010F00

关闭设置码



W010F01

** 开启设置码



WFFD9D2

【4800】



WFFD9D8

【115200】



WFFD9D4

【14400】



W010F00

关闭设置码



W010F01

** 开启设置码

校验



W062900

**【无校验】



W062906

【偶校验】



W062904

【奇校验】

停止位



W012900

**【1 个停止位】



W012901

【2 个停止位】



W010F00

关闭设置码



W010F01

** 开启设置码

数据位



W082908

【8 个数据位】



W0F2908

**【8 个数据位,无校验,1 个停止位】



W0F290E

【8 个数据位，偶校验，1 个停止位】



W0F290C

【8 个数据位，奇校验，1 个停止位】



W0F2909

【8 个数据位，无校验，2 个停止位】



W0F290F

【8 个数据位，偶校验，2 个停止位】



W010F00

关闭设置码



W010F01

** 开启设置码



W0F290D

【8 个数据位，奇校验，2 个停止位】



W082900

【7 个数据位】



W0F2906

【7 个数据位，偶校验，1 个停止位】



W0F2904

【7 个数据位，奇校验，1 个停止位】



W0F2907

【7 个数据位，偶校验，2 个停止位】



W0F2905

【7 个数据位，奇校验，2 个停止位】



W010F00

关闭设置码



W010F01

** 开启设置码

USB 接口

USB HID-KBW

在使用 USB 通讯接口时，可以将识读引擎模拟成 HID-KBW 设备。在这种模式下，识读引擎将成为一个虚拟键盘向主机发送数据。



W070901

** 【切换到 USB HID-KBW 模式】



W010F00

关闭设置码



标准键盘输入模式



键盘仿真输入字符模式

为了使识读引擎能够在任何语言制式下输入任意 ASCII 字符（16 进制值在 0x00~0xFF），可以将虚拟键盘设置为键盘仿真输入字符模式。在使用这种组合方式发送字符时，因为发送的数据较多，速度会减慢。

在切换到“键盘仿真输入字符模式”后，依次识读想要输入的 ASCII 字符对应的字符代码的数据码，识读引擎在解码成功后将采用如下虚拟键盘操作：

- 1、按住“ALT”键不放
- 2、根据该字符代码，依次按数字键盘中的数字键
- 3、松开“ALT”键



注意：启用此模式建议开启主机数字小键盘上的 Num Lock。





W010F01

** 开启设置码

键盘仿真输入控制字符模式

16 进制值位于 0x00~0x1F 之间的 ASCII 值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：

- 1、按住“Ctrl”键不放
- 2、按指定的控制功能键（ASCII 值与控制功能键的对应关系可参阅下页的《键盘仿真输入控制字符对应表》）
- 3、松开“Ctrl”键和控制功能键



W031A01

【切换到键盘仿真输入控制字符模式】



W010F00

关闭设置码



W010F01

** 开启设置码

键盘仿真输入控制字符对应表

ASCII Value (HEX)	Function Key	ASCII Value (HEX)	Function Key
00	2	10	P
01	A	11	Q
02	B	12	R
03	C	13	S
04	D	14	T
05	E	15	U
06	F	16	V
07	G	17	W
08	H	18	X
09	I	19	Y
0A	J	1A	Z
0B	K	1B	[
0C	L	1C	\
0D	M	1D	]
0E	N	1E	6
0F	O	1F	.



W010F00

关闭设置码



W010F01

** 开启设置码

国家/语言键盘布局选择

不同国家语言对应的键盘键位排布、符号等不尽相同。识读引擎可以根据实际需要虚拟成不同国家的键盘制式。



WFF1900

** 【1 - 美式键盘】



WFF1901

【2 - 比利时】



WFF1902

【3 - 巴西】



WFF1903

【4 - 加拿大】



WFF1904

【5 - 捷克斯洛伐克】



WFF1905

【6 - 丹麦】



WFF1906

【7 - 芬兰】



WFF1907

【8 - 法国】



W010F00

关闭设置码



W010F01

** 开启设置码



WFF1908

【9 - 奥地利】



WFF1909

【10 - 希腊】



WFF190A

【11 - 匈牙利】



WFF190B

【12 - 以色列】



WFF190C

【13 - 意大利】



WFF190D

【14 - 拉丁美洲】



WFF190E

【15 - 荷兰】



WFF190F

【16 - 挪威】



WFF1910

【17 - 波兰】



WFF1911

【18 - 葡萄牙】



W010F00

关闭设置码



W010F01

** 开启设置码



WFF1912

【19 - 罗马尼亚】



WFF1913

【20 - 俄罗斯】



WFF1915

【21 - 斯洛伐克】



WFF1916

【22 - 西班牙】



WFF1917

【23 - 瑞典】



WFF1918

【24 - 瑞士】



WFF1919

【25 - 土耳其 1】



WFF191A

【26 - 土耳其 2】



WFF191B

【27 - 英国】



WFF191C

【28 - 日本】



W010F00

关闭设置码



W010F01

** 开启设置码

键间延时设定

虚拟键盘连续按键操作时的按键时间间隔，间隔时间为上一次按键松开到下一次按键按下。



WC01A00

** 【不延时】



WC01A40

【延时 5ms】



WC01A80

【延时 10ms】



WC01AC0

【延时 15ms】



W010F00

关闭设置码



W010F01

** 开启设置码

强制字母大小写转换

此参数仅在标准键盘输入模式和键盘仿真输入控制字符模式下有效。



W381A00

** 【不转换】



W381A20

【全部转换为大写字母】



W381A30

【全部转换为小写字母】



W381A08

【大小写反转】

示例：设置“大小写反转”，此时读取内容数据为“AbC”的条码，主机将得到“aBc”的键盘输入。



W010F00

关闭设置码



W010F01

** 开启设置码

模拟数字小键盘

不开启此功能，则所有发送均按大键盘对应键值发送。

开启此功能后，识读引擎得到的解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发送。若得到的解码数据含有“0~9”之外的也包含在数字小键盘中的“+”“_”“*”“/”“.”等符号，则仍按大键盘对应的键值发送。模拟数字小键盘功能受主机小键盘的 Num Lock 状态的影响：如果主机小键盘的 Num Lock 状态为关闭（Num Lock 灯熄灭），解码数据仍按大键盘对应的键值发送；如果主机小键盘的 Num Lock 状态为开启（Num Lock 灯点亮），解码数据则按数字小键盘对应的键值发送。



W041A04

【模拟数字小键盘】



W041A00

**【不模拟数字小键盘】

注意：启用此功能前请务必先确认主机此时的 Num Lock 处于开启状态。若已开启了“键盘仿真输入字符模式”，则此功能无效。



W010F00

关闭设置码



W010F01

** 开启设置码

USB DATAPIPE

此功能需要在主机上安装相应的驱动程序。



W070900

【切换到 USB DATAPIPE 模式】

USB COM Port Emulation

当识读引擎使用 USB 通讯接口，但主机应用程序是采用串口通讯方式接收数据，则可通过将识读引擎设置为 USB 虚拟串口通讯方式。此功能需要在主机上安装相应的驱动程序。



W070902

【切换到 USB COM Port Emulation 模式】



W010F00

关闭设置码



HID-POS

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，不需要安装驱动。
- ✧ 通讯速度比模拟键盘接口和传统的 RS-232 接口都快很多。

注意：HID-POS 接口不需要安装自定义驱动。但是，HID 接口在 Windows 98 系统需要安装驱动。当设备初次插上 Windows 98 会请求安装驱动。所有的 HID 接口都使用操作系统提供的标准的驱动。





W010F01

** 开启设置码

软件编程访问设备的方法

1. 使用 CreateFile 把设备当成一个 HID 类型设备打开。
2. 然后使用 ReadFile 把扫描得到的数据传递给应用程序。
3. 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

获取扫描数据

扫描解码一个条形码之后, 设备会发送以下的 input 报文:

	Bit							
Byte	7	6	5	4	3	2	1	0
0	报文 ID = 02							
1	条码数据长度							
2-57	条码数据 (1-56)							
58-61	保留 (1-4)							
62	00							
63	00 (没有数据) 或 01 (还有数据)							

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)
EM1365-EX	USB DATAPIPE	8001	32769
	USB HID-KBW	8003	32771
	USB COM Port Emulation	8006	32774
	HID-POS	8010	32784



W010F00

关闭设置码



第五章 数据编辑

简介

识读引擎解码成功后，获得一串数据，这串数据可以是数字，英文，符号等等，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。如您想获得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回车换行，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，既满足了需求又无需修改条码本身的信息内容。

提示：编辑条码数据：<前缀> <条码数据><后缀><结束符>

前缀顺序设置





W010F01

** 开启设置码

自定义前缀

禁止或允许添加自定义前缀

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，识读引擎在“123”字符串前添加“AB”字符串，主机端接收到“AB123”。



W043104

【允许添加自定义前缀】



W043100

**【禁止添加自定义前缀】

设置自定义前缀

自定义首先读取“设置自定义前缀”，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”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 5 个字符。



M000100

【设置自定义前缀】

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

1. 读“开启设置码”
2. 读“设置自定义前缀”码
3. 读以下数据码：“4”“3”“4”“F”“4”“4”“4”“5”（见附录）
4. 读“保存”码（见附录）
5. 读“允许添加自定义前缀”码
6. 读“关闭设置码”
- 7.



W010F00

关闭设置码



W010F01

** 开启设置码

AIM ID 前缀

AIM 是 Automatic Identification Manufacturers（自动识别制造商协会）的简称，AIM 为各种标准条码分别定义了识别代号，具体定义见附录。识读引擎在解码后可以将此识别代号添加在条码数据前，即 AIM ID 前缀。



W186018

【允许添加 AIM ID 前缀】



W186000

**【禁止添加 AIM ID 前缀】



W010F00

关闭设置码



W010F01

** 开启设置码

CODE ID 前缀

除了 AIM ID 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 ID 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 CodeID 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。



W023102

【允许添加 CODE ID 前缀】



W023100

** 【禁止添加 CODE ID 前缀】

Code ID 出厂默认值



WFFD9C2

【所有条码 Code ID 恢复出厂默认值】

修改 Code ID

修改 Code ID 的方法请参考下列示例。

示例：修改 Code 128 的 Code ID 为 “p”（16 进制值为 0x70）

1. 读“开启设置码”
2. 读“修改 Code 128 Code ID” 码
3. 读以下数据码：“7”“0”（见附录）
4. 读“保存”码（见附录）
5. 读“允许添加 Code ID 前缀”码
6. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

修改各条码类型的 Code ID 设置码列表:



M000200

【修改 Code 128 Code ID】



M000201

【修改 UCC/EAN-128 Code ID】



M000202

【修改 AIM 128 Code ID】



M000204

【修改 EAN-8 Code ID】



M000205

【修改 EAN-13 Code ID】



M000206

【修改 ISSN Code ID】



M000207

【修改 ISBN Code ID】



M000208

【修改 UPC-E Code ID】



W010F00

关闭设置码



W010F01

** 开启设置码

修改各条码类型的 Code ID 设置码列表（续）：



M000209

【修改 UPC-A Code ID】



M00020A

【修改 Interleaved 2 of 5 Code ID】



M00020B

【修改 ITF-6 Code ID】



M00020C

【修改 ITF-14 Code ID】



M00020D

【修改 Deutsche 14 Code ID】



M00020E

【修改 Deutsche 12 Code ID】



M00020F

【修改 Matrix 2 of 5 Code ID】



M000210

【修改 Industrial 25 Code ID】



W010F00

关闭设置码



W010F01

** 开启设置码

修改各条码类型的 Code ID 设置码列表（续）：



M000211

【修改 Standard 25 Code ID】



M000212

【修改 Code 39 Code ID】



M000213

【修改 Codabar Code ID】



M000214

【修改 Code 93 Code ID】



M000215

【修改 Code 11 Code ID】



M000216

【修改 Plessey Code ID】



M000217

【修改 MSI-Plessey Code ID】



M000218

【修改 RSS-14 Code ID】



M000219

【修改 RSS-Limited Code ID】



M00021A

【修改 RSS-Expand Code ID】



W010F00

关闭设置码



W010F01

** 开启设置码

自定义后缀

禁止或允许添加自定义后缀

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，识读引擎在“123”字符串后添加“AB”字符串，主机端接收到“123AB”。

注意：自定义后缀字符串总长度不得超过 5 个字符。



W083108

【允许添加自定义后缀】



W083100

** 【禁止添加自定义后缀】



W010F00

关闭设置码



设置自定义后缀

首先读取“设置自定义后缀”，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”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 5 个字符。



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

1. 读“开启设置码”
2. 读“设置自定义后缀”码
3. 读以下数据码：“4”“3”“4”“F”“4”“4”“4”“5”（见附录）
4. 读“保存”码（见附录）
5. 读“允许添加自定义后缀”码
6. 读“关闭设置码”





W010F01

** 开启设置码

结束符后缀

结束符后缀（如回车、换行）可用于标志一段完整数据信息的结束。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。

注意：结束符后缀字符串总长度不得超过 5 个字符。

禁止或允许添加结束符后缀

选择读取以下设置码，可以使识读引擎添加结束符，或不再添加结束符。



W103110

**【允许添加结束符后缀】



W103100

【禁止添加结束符后缀】



W010F00

关闭设置码



设置结束符后缀

读取以下设置码，可以快速将结束符设定为 0x0D（回车）或 0x0D,0x0A（回车换行）或 0x09（Tab），并允许添加结束符进行发送。



【结束符设为 0x0D，并使能发送】



**【结束符设为 0x0D,0x0A，并使能发送】



【结束符设为 0x09，并使能发送】



【设置结束符后缀】

用户也可以自定义结束符后缀：首先读取“设置结束符后缀”，然后按顺序读取要设置的结束符后缀的 16 进制值，最后读取“保存”。

注意：结束符后缀字符串总长度不得超过 5 个字符。

示例：设置结束符后缀为 0x0A

1. 读“开启设置码”
2. 读“设置结束符后缀”码
3. 读以下数据码：“0”“A”（见附录）
4. 读“保存”码（见附录）
5. 读“允许添加结束符后缀”码
6. 读“关闭设置码”





W010F01

** 开启设置码

第七章 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识读引擎适应这些属性变化。

开启“允许识读”的条码类型越少，识读引擎的识读速度越快。您可以禁止识读引擎识读不会使用到的条码类型，以提高识读引擎的工作性能。

全局设置

允许/禁止识读所有条码

读取“禁止识读所有条码”，识读引擎将只能识读设置码，除设置码外的所有条码将无法识读。



WFFD981

【允许识读所有条码】



WFFD982

【禁止识读所有条码】



W010F00

关闭设置码



W010F01

** 开启设置码

Code 128

恢复出厂默认值



WFFD990

【恢复 Code 128 相关设置默认值】

允许/禁止识读 Code 128



W016101

** 【允许识读 Code 128】



W016100

【禁止识读 Code 128】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 Code 128 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Code 128 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000301

【设置最小长度限制】



M000300

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Code128 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

UCC/EAN-128

恢复出厂默认值



WFFD991

【恢复 UCC/EAN-128 相关设置默认值】

允许/禁止识读 UCC/EAN-128



W036203

** 【允许识读 UCC/EAN-128】



W036200

【禁止识读 UCC/EAN-128】



W036201

【识别成 Code 128】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 UCC/EAN-128 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

UCC/EAN-128 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000303

【设置最小长度限制】



M000302

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 UCC/EAN-128 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

AIM 128

恢复出厂默认值



WFFD992

【恢复 AIM 128 相关设置默认值】

允许/禁止识读 AIM 128



W036302

【允许识读 AIM 128】



W036300

** 【禁止识读 AIM 128】



W036301

【识别成 Code 128】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 AIM 128 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

AIM 128 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000305

【设置最小长度限制】



M000304

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 AIM 128 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

EAN-8

恢复出厂默认值



WFFD994

【恢复 EAN-8 相关设置默认值】

允许/禁止识读 EAN-8



W016501

** 【允许识读 EAN-8】



W016500

【禁止识读 EAN-8】

设置是否发送校验位

EAN-8 条码数据固定为 8 字符，第 8 位字符为检验位，用于检验全部 8 个字符的正确性。



W046504

** 【发送校验位】



W046500

【不发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

设置是否允许识读附加码

附加码指在普通条码后面追加的 2 位或 5 位数字条码，如下图，其中左边蓝色线框内为普通条码，右边红色线框内为附加码。



W106510

【识读 2 位附加码】



W106500

** 【不识读 2 位附加码】



W206520

【识读 5 位附加码】



W206500

** 【不识读 5 位附加码】

设置为“识读 2 位附加码”或“识读 5 位附加码”后，识读引擎既可识读普通条码与附加码组成的新条码；也可识读不带附加码的普通条码。设置为“不识读 2 位附加码”或“不识读 5 位附加码”后，普通条码与附加码组成的新条码中附加码的部分将不能被识读，普通条码的部分仍然可以正常识读。



W010F00

关闭设置码



设置是否必须有附加码

该参数仅在识读引擎已设置为“识读 2 位附加码”或“识读 5 位附加码”时才有效。



W086508

【必须有】



W086500

**【不要求】

扩展设置

“条码信息不扩展”，即保持条码原有类型和数据位，不进行扩展。

“条码信息扩展成 EAN-13”，即把条码的数据位进行扩展（前面加 0），但条码类型不改变。

“条码信息及类型转换成 EAN-13”，即条码类型和数据位均进行扩展。



WC06540

【条码信息扩展成 EAN-13】



WC06500

**【条码信息不扩展】



WC06580

【条码信息及类型转换成 EAN-13】



W010F00

关闭设置码



W010F01

** 开启设置码

EAN-13

恢复出厂默认值



WFFD995

【恢复 EAN-13 相关设置默认值】

允许/禁止识读 EAN-13



W016601

** 【允许识读 EAN-13】



W016600

【禁止识读 EAN-13】

设置是否发送校验位

EAN-13 条码数据固定为 13 字符，第 13 位字符为检验位，用于检验全部 13 个字符的正确性。



W046604

** 【发送校验位】



W046600

【不发送校验位】



W010F00

关闭设置码



设置是否允许识读附加码

附加码指在普通条码后面追加的 2 位或 5 位数字条码，如下图，其中左边蓝色线框内为普通条码，右边红色线框内为附加码。



W106610

【识读 2 位附加码】



W106600

**【不识读 2 位附加码】



W206620

【识读 5 位附加码】



W206600

**【不识读 5 位附加码】

设置为“识读 2 位附加码”或“识读 5 位附加码”后，识读引擎既可识读普通条码与附加码组成的新条码；也可识读不带附加码的普通条码。设置为“不识读 2 位附加码”或“不识读 5 位附加码”后，普通条码与附加码组成的新条码中附加码的部分将不能被识读，普通条码的部分仍然可以正常识读。





W010F01

** 开启设置码

设置是否必须有附加码

该参数仅在识读引擎已设置为“识读 2 位附加码”或“识读 5 位附加码”时才有效。



W086608

【必须有】



W086600

** 【不要求】



W010F00

关闭设置码



W010F01

** 开启设置码

ISSN

恢复出厂默认值



WFFD996

【恢复 ISSN 相关设置默认值】

允许/禁止识读 ISSN



W036702

【允许识读 ISSN】



W036700

** 【禁止识读 ISSN】



W036701

【识别成 EAN-13】



W010F00

关闭设置码



W010F01

** 开启设置码

ISBN

恢复出厂默认值



WFFD997

【恢复 ISBN 相关设置默认值】

允许/禁止识读 ISBN



W036802

【允许识读 ISBN】



W036800

** 【禁止识读 ISBN】



W036801

【识别成 EAN-13】

ISBN 数据位设置



W086800

** 【采用 13 位】



W086808

【采用 10 位】



W010F00

关闭设置码



W010F01

** 开启设置码

UPC-E

恢复出厂默认值



WFFD998

【恢复 UPC-E 相关设置默认值】

允许/禁止识读 UPC-E



W016901

** 【允许识读 UPC-E】



W016900

【禁止识读 UPC-E】

设置是否发送校验位

UPC-E 条码数据固定为 8 字符，第 8 位字符为检验位，用于检验全部 8 个字符的正确性。



W046904

** 【发送校验位】



W046900

【不发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

设置是否允许识读附加码

附加码指在普通条码后面追加的 2 位或 5 位数字条码，如下图，其中左边蓝色线框内为普通条码，右边红色线框内为附加码。



W106910

【识读 2 位附加码】



W106900

** 【不识读 2 位附加码】



W206920

【识读 5 位附加码】



W206900

** 【不识读 5 位附加码】

设置为“识读 2 位附加码”或“识读 5 位附加码”后，识读引擎既可识读普通条码与附加码组成的新条码；也可识读不带附加码的普通条码。设置为“不识读 2 位附加码”或“不识读 5 位附加码”后，普通条码与附加码组成的新条码中附加码的部分将不能被识读，普通条码的部分仍然可以正常识读。



W010F00

关闭设置码



W010F01

** 开启设置码

设置是否必须有附加码

该参数仅在识读引擎已设置为“识读 2 位附加码”或“识读 5 位附加码”时才有效。



W086908

【必须有】



W086900

** 【不要求】

设置是否发送系统字符

UPC-E 条码供人识别字符中的第 1 个字节是系统字符。



W306A10

【不发送系统字符】



W306A20

** 【发送系统字符】



W010F00

关闭设置码



W010F01

** 开启设置码

扩展设置

“条码信息不扩展”，即保持条码原有类型和数据位，不进行扩展。

“条码信息扩展成 UPC-A”，即把条码的数据位进行扩展，但条码类型不改变。

“条码信息及类型转换成 UPC-A”，即条码类型和数据位均进行扩展。



WC06940

【条码信息扩展成 UPC-A】



WC06900

** 【条码信息不扩展】



WC06980

【条码信息及类型转换成 UPC-A】



W010F00

关闭设置码



W010F01

** 开启设置码

UPC-A

恢复出厂默认值



WFFD999

【恢复 UPC-A 相关设置默认值】

允许/禁止识读 UPC-A



W036B02

** 【允许识读 UPC-A】



W036B00

【禁止识读 UPC-A】



W036B01

【识别成 EAN-13】



W010F00

关闭设置码



W010F01

** 开启设置码

设置是否发送校验位

UPC-A 条码数据固定为 13 字符，第 13 位字符为检验位，用于检验全部 13 个字符的正确性。



W086B08

**【发送校验位】



W086B00

【不发送校验位】

前缀字符发送设置

UPC-A 条码的国家码为前导字符，该字符一般不显示在条码下方的供人识别字符中，“0”代表 USA。供人识别字符中的第 1 个字符是系统字符。



W036A00

【无前缀】



W036A01

**【系统字符】



W036A02

【系统字符与国家码】



W010F00

关闭设置码



设置是否允许识读附加码

附加码指在普通条码后面追加的 2 位或 5 位数字条码，如下图，其中左边蓝色线框内为普通条码，右边红色线框内为附加码。



W206B20
【识读 2 位附加码】



W206B00
** 【不识读 2 位附加码】



W406B40
【识读 5 位附加码】



W406B00
** 【不识读 5 位附加码】

设置为“识读 2 位附加码”或“识读 5 位附加码”后，识读引擎既可识读普通条码与附加码组成的新条码；也可识读不带附加码的普通条码。设置为“不识读 2 位附加码”或“不识读 5 位附加码”后，普通条码与附加码组成的新条码中附加码的部分将不能被识读，普通条码的部分仍然可以正常识读。





W010F01

** 开启设置码

设置是否必须有附加码

该参数仅在识读引擎已设置为“识读 2 位附加码”或“识读 5 位附加码”时才有效。



W106B10

【必须有】



W106B00

** 【不要求】



W010F00

关闭设置码



W010F01

** 开启设置码

Interleaved 2 of 5

恢复出厂默认值



WFFD99A

【恢复 Interleaved 2 of 5 相关设置默认值】

允许/禁止识读 Interleaved 2 of 5



W016C01

** 【允许识读 Interleaved 2 of 5】



W016C00

【禁止识读 Interleaved 2 of 5】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Interleaved 2of 5 条码数据中不强制包含校验符，如果有校验符，则是数据的最后 1 个字符。校验符是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验符”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验符外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验符”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验符作为正常数据最后 1 位一起传输，校验失败将提示读码失败。



W0C6C00

【无校验】



W0C6C04

** 【校验但不发送校验符】



W0C6C0C

【校验且发送校验符】



W010F00

关闭设置码



设置读码长度限制

用于设置 Interleaved 2 of 5 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Interleaved 2 of 5 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000307

【设置最小长度限制】



M000306

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Interleaved 2 of 5 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

ITF-6

ITF-6 是一种特殊的 Interleaved2 of 5 条码，即条码数据长度为 6 字符且最后 1 个字符数据为校验位的 Interleaved2 of 5 码。

恢复出厂默认值



WFFD99B

【恢复 ITF-6 相关设置默认值】

允许/禁止识读 ITF-6

识读引擎默认将 ITF-6 识别成 Interleaved 2 of 5。



W036D01

【禁止识读 ITF-6】



W0B6D02

【允许识读 ITF-6 但不发送校验符】



W0B6D0A

【允许识读 ITF-6 且发送校验符】

注意：建议在使用普通交插二五码时，禁止 ITF-6；或在需要使用 ITF-6 时，禁止识读普通的交插二五码。



W010F00

关闭设置码



ITF-14

ITF-14 是一种特殊的 Interleaved 2 of 5 条码，即条码数据长度为 14 字符且最后 1 个字符数据为校验位的 Interleaved2 of 5 码。

恢复出厂默认值



允许/禁止识读 ITF-14

识读引擎默认将 ITF-14 识别成 Interleaved 2 of 5。



注意：建议在使用普通交插二五码时，禁止 ITF-14；或在需要使用 ITF-14 时，禁止识读普通的交插二五码。





W010F01

** 开启设置码

Deutsche 14

恢复出厂默认值



WFFD99D

【恢复 Deutsche 14 相关设置默认值】

允许/禁止识读 Deutsche 14

识读引擎默认将 Deutsche 14 识别成 Interleaved 2 of 5。



W036F01

【禁止识读 Deutsche 14】



W0B6F02

【允许识读 Deutsche 14 但不发送校验位】



W0B6F0A

【允许识读 Deutsche 14 且发送校验位】

提示：由于 Deutsche14 与 ITF-14、普通 ITF 使用同样的编码方式，所以若几者同时使用会对功能设置及解码产生误解。建议不使用 Deutsche14 时将其关闭。



W010F00

关闭设置码



W010F01

** 开启设置码

Deutsche 12

恢复出厂默认值



WFFD99E

【恢复 Deutsche 12 相关设置默认值】

允许/禁止识读 Deutsche 12

识读引擎默认将 Deutsche 12 识别成 Interleaved 2 of 5。



W037001

【禁止识读 Deutsche 12】



W0B7002

【允许识读 Deutsche 12 但不发送校验符】



W0B700A

【允许识读 Deutsche 12 且发送校验符】

提示：由于 Deutsche12 与 ITF-12、普通 ITF 使用同样的编码方式，所以若几者同时使用会对功能设置及解码产生误解。建议不使用 Deutsche12 时将其关闭。



W010F00

关闭设置码



W010F01

** 开启设置码

Matrix 2 of 5 (European Matrix 2 of 5)

恢复出厂默认值



WFFD99F

【恢复 Matrix 2 of 5 相关设置默认值】

允许/禁止识读 Matrix 2 of 5



W017101

** 【允许识读 Matrix 2 of 5】



W017100

【禁止识读 Matrix 2 of 5】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Matrix 2 of 5 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验位作为正常数据最后 1 位一起传输，校验失败将提示读码失败。



W0C7100

**【无校验】



W0C7104

【校验但不发送校验位】



W0C710C

【校验且发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 Matrix 2 of 5 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎不会把该条码内容发送到主机端。

Matrix 2 of 5 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000309

【设置最小长度限制】



M000308

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Matrix 2 of 5 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Industrial 25

恢复出厂默认值



WFFD9A0

【恢复 Industrial 25 相关设置默认值】

允许/禁止识读 Industrial 25



W017201

** 【允许识读 Industrial 25】



W017200

【禁止识读 Industrial 25】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Industrial 25 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验位作为正常数据最后 1 位一起传输，校验失败将提示读码失败。



W0C7200

**【无校验】



W0C7204

【校验但不发送校验位】



W0C720C

【校验且发送校验位】



W010F00

关闭设置码



设置读码长度限制

用于设置 Industrial 25 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Industrial 25 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M00030B

【设置最小长度限制】



M00030A

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Industrial 25 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”





W010F01

** 开启设置码

Standard 25

恢复出厂默认值



WFFD9A1

【恢复 Standard 25 相关设置默认值】

允许/禁止识读 Standard 25



W017301

** 【允许识读 Standard 25】



W017300

【禁止识读 Standard 25】



W010F00

关闭设置码



校验设置

Standard 25 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验位作为正常数据最后 1 位一起传输，校验失败将提示读码失败。





W010F01

** 开启设置码

设置读码长度限制

用于设置 Standard 25 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Standard 25 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M00030D

【设置最小长度限制】



M00030C

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Standard 25 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Code 39

恢复出厂默认值



WFFD9A2

【恢复 Code 39 相关设置默认值】

允许/禁止识读 Code 39



W017401

** 【允许识读 Code 39】



W017400

【禁止识读 Code 39】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Code 39 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验位作为正常数据最后 1 位一起传输，校验失败将提示读码失败。



W187400

** 【无校验】



W187408

【校验但不发送校验位】



W187418

【校验且发送校验位】



W010F00

关闭设置码



设置是否发送起始符和终止符

Code39 条码数据前后各有一个字符的“*”作为起始符和终止符，可以设置在读码成功后是否将起始符和终止符与条码数据一同传输。



Full ASCII 支持

Code 39 的编码方法可以包括对所有 ASCII 字符的表示形式，通过设置，可以使识读引擎支持含有全 ASCII 字符集的条码。





W010F01

** 开启设置码

设置读码长度限制

用于设置 Code 39 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Code 39 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M00030F

【设置最小长度限制】



M00030E

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Code 39 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Codabar

恢复出厂默认值



WFFD9A3

【恢复 Codabar 相关设置默认值】

允许/禁止识读 Codabar



W017501

** 【允许识读 Codabar】



W017500

【禁止识读 Codabar】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Codabar 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

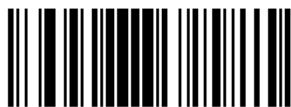
设置为“校验但不发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 1 位数据做校验，若校验通过则将校验位作为正常数据最后 1 位一起传输，校验失败将提示读码失败。



W607500

**【无校验】



W607520

【校验但不发送校验位】



W607560

【校验且发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

起始符和终止符设置



W047504

** 【发送起始符与终止符】



W047500

【不发送起始符与终止符】



W187500

** 【以 ABCD/ABCD 作为起始和终止字符】



W187508

【以 ABCD/TN*E 作为起始和终止字符】



W187510

【以 abcd/abcd 作为起始和终止字符】



W187518

【以 abcd/tn*e 作为起始和终止字符】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 Codabar 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Codabar 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000311

【设置最小长度限制】



M000310

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Codabar 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Code 93

恢复出厂默认值



WFFD9A4

【恢复 Code 93 相关设置默认值】

允许/禁止识读 Code 93



W017601

** 【允许识读 Code 93】



W017600

【禁止识读 Code 93】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Code 93 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 2 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 2 位数据做校验，若校验通过则传输除校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 2 位数据做校验，若校验通过则将校验位作为正常数据最后 2 位一起传输，校验失败将提示读码失败。



W0C7600

【无校验】



W0C7604

** 【校验但不发送校验位】



W0C760C

【校验且发送校验位】



W010F00

关闭设置码



设置读码长度限制

用于设置 Code 93 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Code 93 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000313

【设置最小长度限制】



M000312

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Code 93 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Code 11

恢复出厂默认值



WFFD9A5

【恢复 Code 11 相关设置默认值】

允许/禁止识读 Code 11



W017701

【允许识读 Code 11】



W017700

** 【禁止识读 Code 11】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Code 11 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个或 2 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。



W1C7700

【无校验】



W1C7704

**【一位校验，MOD11】



W1C7708

【两位校验 MOD11/MOD11】



W1C770C

【两位校验 MOD11/MOD9】



W1C7710

【MOD11 单校验 (Len<=11))
MOD11/MOD11 双校验 (Len>11)】



W1C7714

【MOD11 单校验 (Len<=11)
MOD11/MOD9 双校验 (Len>11)】



W207720

【不发送校验位】



W207700

**【发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 Code 11 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Code 11 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000315

【设置最小长度限制】



M000314

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Code 11 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”



W010F00

关闭设置码



W010F01

** 开启设置码

Plessey

恢复出厂默认值



WFFD9A6

【恢复 Plessey 相关设置默认值】

允许/禁止识读 Plessey



W017801

【允许识读 Plessey】



W017800

** 【禁止识读 Plessey】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

Plessey 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 2 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。

设置为“校验但不发送校验位”则识读引擎将根据条码最后 2 位数据做校验，若校验通过则传输除 校验位外的正常数据，校验失败将提示读码失败。

设置为“校验且发送校验位”则识读引擎将根据条码最后 2 位数据做校验，若校验通过则将校验位作为正常数据最后 2 位一起传输，校验失败将提示读码失败。



W0C7800

【无校验】



W0C7804

**【校验但不发送校验位】



W0C780C

【校验且发送校验位】



W010F00

关闭设置码



设置读码长度限制

用于设置 Plessey 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

Plessey 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000317

【设置最小长度限制】



M000316

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 Plessey 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”





W010F01

** 开启设置码

MSI-Plessey

恢复出厂默认值



WFFD9A7

【恢复 MSI-Plessey 相关设置默认值】

允许/禁止识读 MSI-Plessey



W017901

【允许识读 MSI-Plessey】



W017900

** 【禁止识读 MSI-Plessey】



W010F00

关闭设置码



W010F01

** 开启设置码

校验设置

MSI-Plessey 条码数据中不强制包含校验位，如果有校验位，则是数据的最后 1 个或 2 个字符。校验位是根据所有数据计算得出的值，用以校验数据是否正确。

设置为“无校验”则识读引擎将正常传输所有条码数据。



W0C7900

【无校验】



W0C7904

** 【单校验 MOD10】



W0C7908

【双校验 MOD10/MOD10】



W0C790C

【双校验 MOD10/MOD11】



W107910

【发送校验位】



W107900

** 【不发送校验位】



W010F00

关闭设置码



W010F01

** 开启设置码

设置读码长度限制

用于设置 MSI-Plessey 的有效识读长度。若读取条码长度与所设置有效长度不符，读码不成功，识读引擎将不会把该条码内容发送到主机端。

MSI-Plessey 识读长度由“最小长度”和“最大长度”构成。任何一维条码最大长度限制值不得超过 255，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。



M000319

【设置最小长度限制】



M000318

【设置最大长度限制】

示例：限制识读引擎只识读最小 8 字节，最大 12 字节的 MSI-Plessey 条码

1. 读“开启设置码”
2. 读“设置最小长度限制”码
3. 读数据码“8”（见附录）
4. 读“保存”码（见附录）
5. 读“设置最大长度限制”码
6. 读数据码“1”“2”
7. 读“保存”码
8. 读“关闭设置码”
- 9.



W010F00

关闭设置码



W010F01

** 开启设置码

RSS-14

恢复出厂默认值



WFFD9A8

【恢复 RSS-14 相关设置默认值】

允许/禁止识读 RSS-14



W017A01

** 【允许识读 RSS-14】



W017A00

【禁止识读 RSS-14】

设置是否发送 AI 字符



W047A04

** 【发送 AI 字符】



W047A00

【不发送 AI 字符】



W010F00

关闭设置码



W010F01

** 开启设置码

RSS-Limited

恢复出厂默认值



WFFD9A9

恢复 RSS-Limited 相关设置默认值

允许/禁止识读 RSS-Limited



W017B01

** 【允许识读 RSS-Limited】



W017B00

【禁止识读 RSS-Limited】

设置是否发送 AI 字符



W047B04

** 【发送 AI 字符】



W047B00

【不发送 AI 字符】



W010F00

关闭设置码



W010F01

** 开启设置码

RSS-Expand

恢复出厂默认值



WFFD9AA

【恢复 RSS-Expand 相关设置默认值】

允许/禁止识读 RSS-Expand



W017C01

** 【允许识读 RSS-Expand】



W017C00

【禁止识读 RSS-Expand】



W010F00

关闭设置码

附录

默认设置表

参数名称		默认设置	备注
系统设置			
设置码功能		开启	设置为关闭，将发送出设置码信息
设置码信息		不发送	
识读模式		手动模式	
手动模式	单次读码时长	15 秒	1-255 秒，0：无限长，步长为 1 秒
连续模式	单次读码时长	15 秒	1-255 秒，0：无限长，步长为 1 秒
	识读间隔时长	1.0 秒	0.0-25.5 秒，步长为 0.1 秒
	重复识读相同条码	不延时	可选择“要求相同读码延时”，时长设置为 0 时，则无限长延时，即不允许重复识读相同条码。
感应模式	单次读码时长	15 秒	1-255 秒，0：无限长，步长为 1 秒
	稳像时长	0.4 秒	0.0-25.5 秒，步长为 0.1 秒
	重复识读相同条码	不延时	可选择“要求相同读码延时”，时长设置为 0 时，则无限长延时，即不允许重复识读相同条码。
	灵敏度	中灵敏度	
命令触发模式	单次读码时长	15 秒	1-255 秒，0：无限长，步长为 1 秒
安全级别		0 级	
识读成功提示音		开启	
识读成功提示音频率		中频	
识读成功提示音持续时间		80ms	
解码状态提示符		不发送	
静音模式		关闭	临时设置
照明		识读时亮	临时设置

参数名称		默认设置	备注
通讯设置			
串口通讯	波特率	9600	
	校验	无校验	
	数据位	8 个	
	停止位	1 个	
	流控	无流控	
USB 通讯		USB HID-KBW	其他：DataPipe、USB COM Port Emulation、HID-POS
USB HID-KBW	输入模式	标准键盘	
	国家/语言键盘布局	第 1 种，美国英语键盘	
	键间延时	不延时	
	强制字母大小写转换	不转换	
	数字小键盘	不模拟	
数据格式设置			
前缀顺序设置		CODE ID+自定义+AIMID	
添加 AIMID 前缀		Off	]Cm 标识
添加 CODE ID		Off	1 个或 2 个字符，大写或小写字母
添加自定义前缀		Off	前缀字符最多 5 个
添加自定义后缀		Off	后缀字符最多 5 个
添加结束符后缀		On--0x0D,0x0A	允许，回车换行

参数名称	默认设置	备注
Code 128		
使能	On	
最小长度	1	不得小于 1（含校验）
最大长度	80	
UCC/EAN-128 (GS1-128)		
使能	On	
最小长度	1	不得小于 1（含校验）
最大长度	80	
AIM 128		
使能	Off	
最小长度	1	不得小于 1（含校验）
最大长度	80	
EAN-8		
使能	On	
校验符	发送	
2 位附加码	不识读	
5 位附加码	不识读	
必须有附加码	不要求	
扩展为 EAN-13	不扩展	
EAN-13		
使能	On	
校验符	发送	
2 位附加码	不识读	
5 位附加码	不识读	
必须有附加码	不要求	
ISSN		
使能	Off	
ISBN		
使能	Off	
ISBN 格式	13 位	

参数名称	默认设置	备注
UPC-E		
使能	On	
校验符	发送	
2 位附加码	不识读	
5 位附加码	不识读	
必须有附加码	不要求	
扩展为 UPC-A	不扩展	
系统字符	发送	
UPC-A		
使能	On	
校验符	发送	
2 位附加码	不识读	
5 位附加码	不识读	
必须有附加码	不要求	
前缀字符发送	系统字符	
Interleaved 2 of 5		
使能	On	
校验	On	
校验符	不发送	
最小长度	6	不得小于 3（含校验）
最大长度	100	
ITF-6		
使能	识别成 I25	
校验符	发送	
ITF-14		
使能	识别成 I25	
校验符	发送	
Deutsche 14		
使能	识别成 I25	
校验符	发送	
Deutsche 12		
使能	识别成 I25	
校验符	发送	

参数名称	默认设置	备注
Matrix 2 of 5		
使能	On	
校验	Off	
校验符	不发送	
最小长度	6	不得小于 2（含校验）
最大长度	80	
Industrial 25		
使能	On	
校验	Off	
校验符	不发送	
最小长度	6	不得小于 2（含校验）
最大长度	80	
Standard 25		
使能	On	
校验	Off	
校验符	不发送	
最小长度	6	不得小于 2（含校验）
最大长度	80	
Code 39		
使能	On	
校验	Off	
校验符	不发送	
起始/终止符	不发送	
支持 Full ASCII	On	
最小长度	4	不得小于 2（含校验）
最大长度	50	
Codabar		
使能	On	
校验	Off	
校验符	不发送	
起始/终止符	发送	
起始/终止符格式	ABCD/ABCD	
最小长度	4	不得小于 1（含校验）
最大长度	60	

参数名称	默认设置	备注
Code 93		
使能	On	
校验	On	
校验符	不发送	
最小长度	2	不得小于 1（含校验）
最大长度	80	
Code 11		
使能	Off	
校验	一位校验，MOD11	
校验符	不发送	
最小长度	4	不得小于 2（含校验）
最大长度	80	
Plessey		
使能	Off	
校验	On	
校验符	不发送	
最小长度	4	不得小于 3（含校验）
最大长度	60	
MSI-Plessey		
使能	Off	
校验	一位校验，MOD10	
校验符	不发送	
最小长度	4	不得小于 2（含校验）
最大长度	60	
RSS-14		
使能	On	
AI 字符	发送	
RSS-Limited		
使能	On	
AI 字符	发送	
RSS-Expand		
使能	On	

AIM ID 列表

条码类型	AIM ID	说明
Code 128	JC0	普通 Code 128
UCC/EAN 128 (GS1-128)	JC1	FNC1 在第 1 码词位置
AIM 128	JC2	FNC1 在第 2 码词位置
EAN-8	JE4	普通 EAN-8 数据
	JE4....JE1...	EAN-8 数据加上 2 位附加码
	JE4....JE2...	EAN-8 数据加上 5 位附加码
EAN-13	JE0	普通 EAN-13 数据
	JE3	EAN-13 数据加上 2/5 位附加码
ISSN	JX5	
ISBN	JX4	
UPC-E	JE0	普通 UPC-E 数据
	JE3	UPC-E 数据加上 2/5 位附加码
UPC-A	JE0	普通 UPC-A 数据
	JE3	UPC-A 数据加上 2/5 位附加码
Interleaved 2 of 5	JI0	无校验
	JI1	校验且发送校验字符
	JI3	校验但不发送校验字符
ITF-6	JI1	发送校验字符
	JI3	不发送校验字符
ITF-14	JI1	发送校验字符
	JI3	不发送校验字符
Deutsche 14 Deutsche 12	JX0	
Matrix 2 of 5	JX1	无校验
	JX2	有校验且发送校验字符
	JX3	有校验且不发送校验字符
Industrial 25	JS0	目前没有任何的特别指定
Standard 25	JR0	无校验

条码类型	AIM ID	说明
	JR8	MOD 7 校验但不发送校验字符
	JR9	MOD 7 校验且发送校验字符
Code 39	JA0	无校验，无 Full ASCII 扩展。所有数据如原发送
	JA1	MOD 43 校验，且发送校验字符
	JA3	MOD 43 校验，但不发送校验字符
	JA4	进行了 Full ASCII 扩展，但无校验
	JA5	进行了扩展，MOD43 校验，且发送校验字符
	JA7	进行了扩展，MOD43 校验，但不发送校验字符
Codabar	JF0	标准数据包，没有特别处理
	JF1	用于美国血液中心管理
	JF2	校验，且发送校验字符
	JF4	校验，但不发送校验字符
Code 93	JG0	目前无特别指定
Code 11	JH0	MOD11 单字符校验，且发送校验字符
	JH1	MOD11/MOD11 双字符校验，且发送校验字符
	JH3	校验，但不发送校验字符
	JH8	MOD11/MOD9 双字符校验，且发送校验字符
	JH9	无校验
Plessey	JP0	目前无特别指定
MSI Plessey	JM0	MOD10 校验，且发送校验字符
	JM1	MOD10 校验，但不发送校验字符
	JM7	MOD10 /MOD11 校验，且不发送校验字符
	JM8	MOD10 /MOD11 校验，且发送校验字符
	JM9	无校验
RSS-14/RSS-Limited RSS-Expand	Je0	标准数据包
	Je1	其它用途
	Je2	其它用途
	Je3	其它用途

参考资料：ISO/IEC 15424-2008 信息技术 – 自动识别及数据采集技术 –数据载体标识符（包括符号标识符）

Code ID 列表

条码类型	Code ID
Code 128	j
UCC/EAN-128	u
AIM 128	f
SETTING 128	t
EAN-8	g
EAN-13	d
ISSN	n
ISBN	B
UPC-E	h
UPC-A	c
Interleaved 2 of 5	e
ITF-6	r
ITF-14	q
Deutsche 14	w
Deutsche 12	l
Matrix 2 of 5(European Matrix 2 of 5)	v
Industrial 25	i
Standard 25	s
Code 39	b
Codabar	a
Code 93	y
Code 11	z
Plessey	p
MSI-Plessey	m
RSS-14	D
RSS-Limited	C
RSS-Expand	R

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)

十六进制	十进制	字符
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	, (Comma)
2d	45	- (Minus / Dash)
2e	46	. (Dot)
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)

十六进制	十进制	字符
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)

十六进制	十进制	字符
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)

数据码

0~5



D0000000

【0】



D0000001

【1】



D0000002

【2】



D0000003

【3】



D0000004

【4】



D0000005

【5】

6~9



D000006

【6】



D000007

【7】



D000008

【8】



D000009

【9】

A ~ F



D00000A

【A】



D00000B

【B】



D00000C

【C】



D00000D

【D】



D00000E

【E】



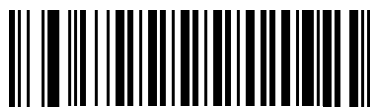
D00000F

【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，您可以取消读取错误的数据。

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”。



D000012

【保存】



D000010

【取消前一次读的一位数据】



D000011

【取消前面读的一串数据】

F1~F12

在 USB HID-KBW 模式下，方便用户输出 F1~F12 功能键值，通过以下的条码进行输出。

F1~F6



F000000

【F1】



F000001

【F2】



F000002

【F3】



F000003

【F4】



F000004

【F5】



F000005

【F6】

F7~F12



F000006

【F7】



F000007

【F8】



F000008

【F9】



F000009

【F10】



F00000A

【F11】



F00000B

【F12】



Headquarters / 总部

福建新大陆自动识别技术有限公司

地址：福建省福州市马尾区儒江西路 1 号新大陆科技园

邮编：350015

电话：+86 - (0) 591-83979222

传真：+86 - (0) 591-83979208

E-mail: marketing@nlscan.com

WEB: www.nlscan.com

Newland Europe BV/ 欧洲新大陆有限公司

Rolweg 25, 4104 AV Culemborg, The Netherlands

TEL: +31 (0) 345 87 00 33

FAX: +31 (0) 345 87 00 39

Email: info@newland-id.com

WEB: www.newland-id.com

Tech Support: tech-support@newland-id.com

Newland North America Inc. / 北美新大陆有限公司

Address: 46559 Fremont Blvd., Fremont, CA 94538, USA

TEL: 510 490 3888

Fax: 510 490 3887

Email: info@newlandna.com

WEB: www.newlandamerica.com

Newland Taiwan Inc. / 台湾新大陆资讯科技股份有限公司

7F-6, No. 268, Liancheng Rd., Jhonghe Dist. 235, New Taipei City, Taiwan

新北市 235 中和區連城路 268 號 7 樓之 6

(遠東世紀廣場 J 棟)

TEL: +886 2 7731 5388

FAX: +886 2 7731 5389

Email: info@newland-id.com.tw

WEB: www.newland-id.com.tw