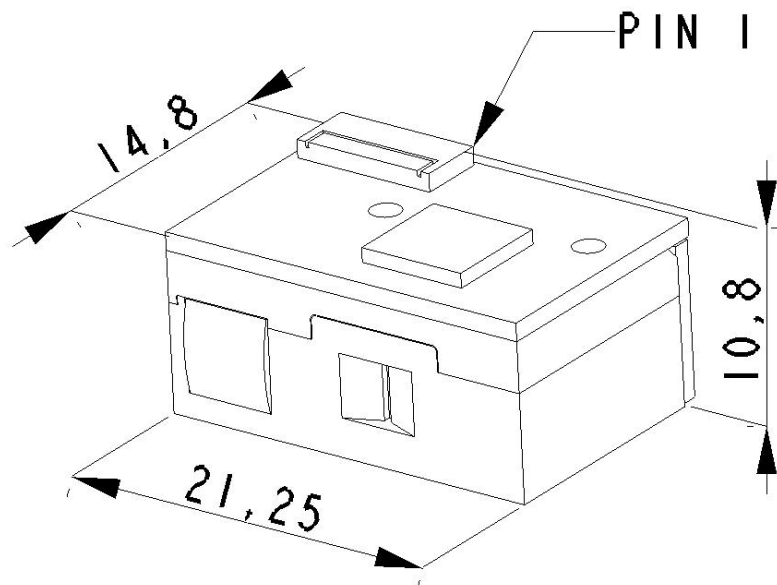


钛码 TM932 微型条码扫描模块

一、扫描模块特性

TM932 是一款专门为移动数据终端设计的超小体积的高速条形码扫描模组，可扫描支付宝、微信、电脑显示器等屏幕条码。



1. 内置解码器

模块内置32位解码器，解码完成后直接输出条码数据的ASCII码，因此用户无需编写条码解码程序，有助于缩短您的产品的开发周期。

2. 解码能力强

采用自主研发的解码软件，解码能力强，能解超长、弯曲或者磨损的条码。

2. 性能稳定

模块的软件和硬件都经过精心设计和严格测试，具有稳定的性能，适合长时间连续工作。

3. 丰富的条码码制

模块支持所有常见的一维码，并可根据用户的需求添加自定义码制。

4. 接口简单，使用方便

模块以RS232的形式输出条码数据。在RS232设置成无握手协议的情况下，用户只需要接两根电源线（GND和VCC）和一根串口数据线（TXD）即可以获取条码数据，非常适合各种嵌入式系统和自动控制系统使用。同时具备标准USB HID接口，无需安装驱动，即插即用。

二、扫描模块技术指标及参数

电源电压	直流3.3V
电源电流	工作：27毫安 读码瞬间：73毫安 休眠：6毫安
CPU	32位解码器
光源	红色LED
解码速度	最快达到300 次/秒
解析度	3mil
支持码制	UPC-A, UPC-E, EAN-13, EAN-8, ISBN/ISSN, 39 码, 39 码（ASCII 全码），32码, 交叉25 码, 工业25 码, 矩阵25 码, 库德巴码, 128 码, 93 码, 11 码, MSI/Plessey, UCC/EAN 128, 中国邮政码, GS1 DataBar（是最新的国际标准条码，前身是RSS）系列
提示方式	外接蜂鸣器和LED 指示灯
数据接口	USB、RS-232 串口 (3.3V电平)
触发方式	外部脉冲触发/串口命令触发/条码靠近自动扫描
尺寸	宽×长×高： 14.8mm×21.25mm×10.8mm

三、扫描模块接口说明

引脚号	信号名称	类型	说明
1	UPDATA	输入	保留， 请保持悬空或者低电平
2	VCC	输入	模组供电电源(3.3V)
3	GND	输入	地线
4	RXD	输入	接收数据：串行端口输入线
5	TXD	输出	发送数据：串行端口输出线
6	USB_D-	输入	USB 接口 D-
7	USB_D+	输出	USB 接口 D+
8	PWRDWN	输出	休眠指示：高电平时表示模组处于休眠模式
9	BPR	输出	蜂鸣器：PWM 低电流蜂鸣输出，电流放大后用于驱动无源蜂鸣器
10	DLED	输出	LED：低电流解码 LED 信号输出。成功扫描条码后，该引脚输出一个负脉冲，用于驱动外置的 LED 指示灯
11	WAKE	输入	唤醒：引擎在低功率模式时，此引脚下降沿唤醒引擎
12	TRIG	输入	触发：拉低此引脚电平，扫描模组启动一次条码扫描

提示：以下任意一个都可以唤醒扫描模组。

- 1. WAKE 针脚下降边沿；
- 2. 发送 0x00 到条码扫描引擎的 RXD 端口；
- 3. TRIG 针脚下降沿；

四、扫描模组串口命令协议

目前支持两种命令格式，命令 1 格式需扫码打开，命令 2 格式一直有效；



打开命令 1 模式



关闭命令 1 模式

1、set Mode Command :

0X16 CMD DATAH DATAL SUM (SUM= CMD+ DATAH+ DATAL)

CMD	DATA	TM2240	CMD
0xF1	0x00 0x00	Start scan	0x16 0xF1 0x00 0x00 0xF1
0xF2	0x00 0x00	Set Auto Mode	0x16 0xF2 0x00 0x00 0xF2
0xF3	0x00 0x00	Set Continue Mode	0x16 0xF3 0x00 0x00 0xF3

2、other Command :

LEN CMD MSG STA DATA SUMH SUML (SUM=0xFFFF-(LEN+CMD+MSG+STA+DATA)+1)

CMD	MSG	DATA	TM2240	CMD
0xA4	0x00		getversion	0x05 0xA4 0x000x000x00 0xFF 0x57
0xE4	0x00		Start scan	0x05 0xE4 0x00 0x000x00 0xFF 0x17
0xE6	0x00	0:Low Frequency 1:Medium Frequency 2:High Frequency	Beeper Tone	0x05 0xE6 0x00 0x000x00 0xFF 0x15
0xC8	0x00		Set default	0x05 0xC8 0x00 0x000x00 0xFF 0x33