

ZT.SEG8B4A036A使用说明书

概述

ZT.SEG8B4A036A集成了数码管显示所需要的所有元件的条件。4位8段数码管可以按照用户的任意要求显示内容。模块接口为准IIC借口，用户可自行定义模块的设备地址，用户可自行设置显示的亮度，使用灵活方便。

特点

- 标准I2C接口，集成4.7K上拉电阻
- 模块地址可任意修改
- 自带演示功能
- 模块管脚为标准2*4,2,54间距
- 宽电源电压，3.3V~5V
- 适用于3.3V单片机和5V单片机系统
- 亮度可调

管脚功能描述

表 1. 管脚功能描述

管脚	功能
VCC	电源，3.0-5.5V
GND	电源地
SDA	I2C总线数据线，上电或复位期间为低电平时，模块进入演示模式（ARDUINO的NANO平台接A4管脚）
SCL	I2C总线时钟线，上电或复位期间为低电平时，模块进入设置新地址模式（ARDUINO的NANO平台接A5管脚）
RST	复位管脚（可悬空或MCU控制）

模块物理尺寸

单位为毫米，公差为 2MM，

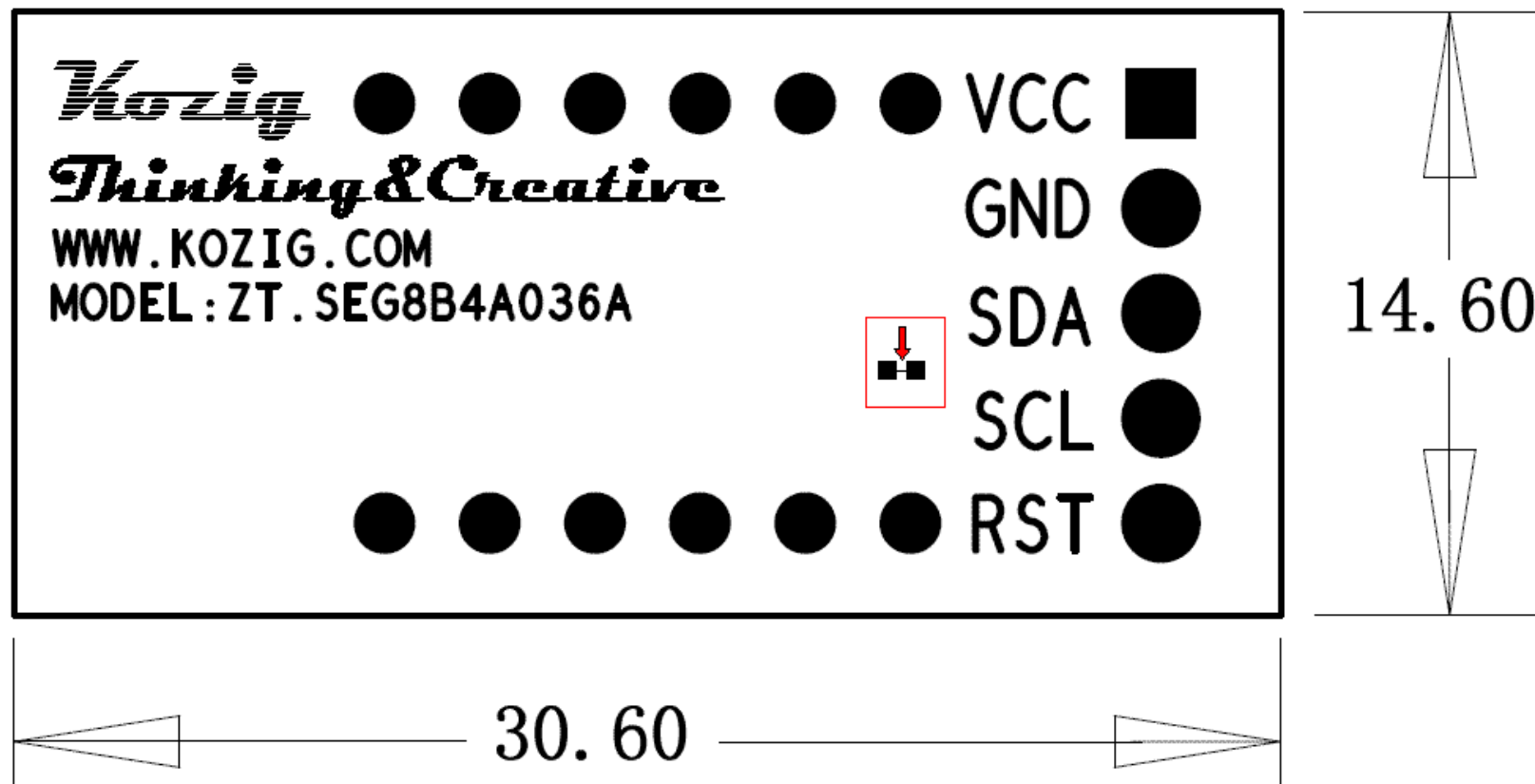


图 1. 模块底视图

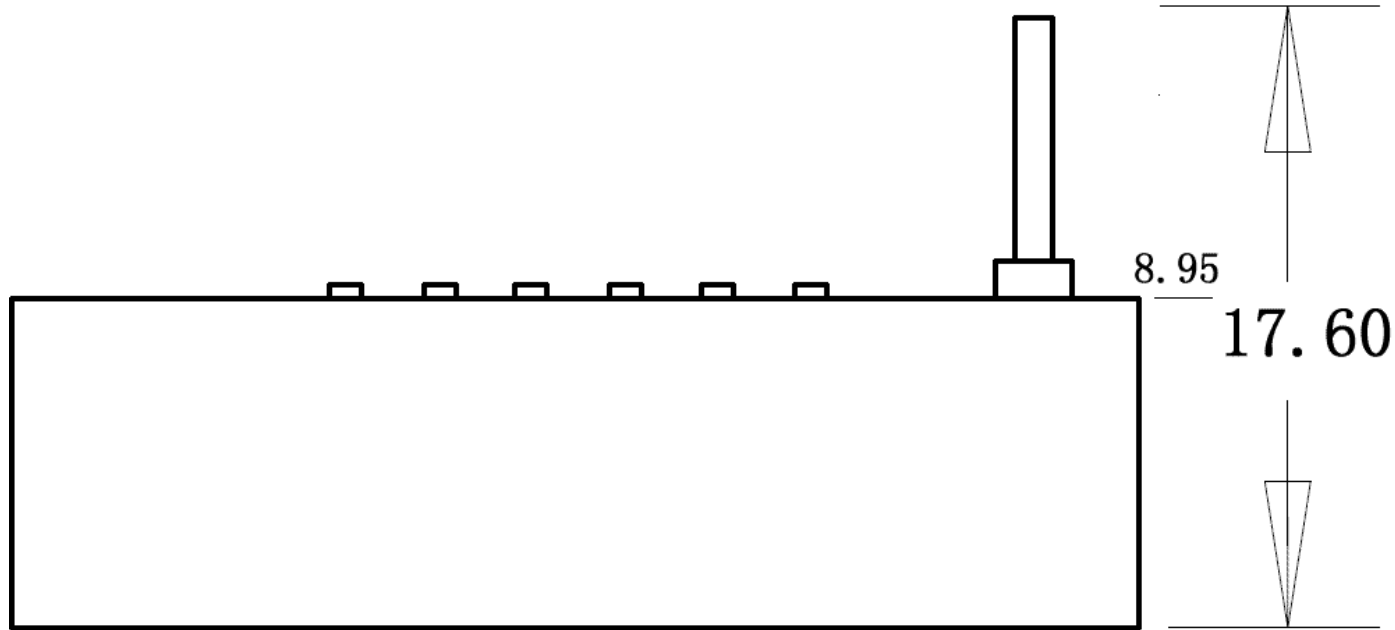


图 2. 模块侧视图

模块4. 7K上拉电阻的位置与拆除

在模块物理尺寸的图1中，红色线框框住的地方有两个很靠近的焊盘，这两个焊盘中间连有一条细线，如果不想使用模块内部的上拉电阻，把两个焊盘中间的细线割掉即可，如果想再次使用内部的上拉电阻，用锡将这两个很靠近的焊盘搭焊即可。

寄存器描述

地址	读/写	描述	最低版本号
0x02	只写	透传数据	ZT.SC-I2CMx-V1.0
0x04	读写	睡眠/唤醒，睡眠写0xA5，唤醒写0xA1，	ZT.SC-I2CMx-V1.0
0x06	共用	状态	ZT.SC-I2CMx-V1.0
0x07	只读	忙/闲	ZT.SC-I2CMx-V1.0
0x08	只写	设置地址(模块出厂默认的地址设置为b0100111x)	ZT.SC-I2CMx-V1.0
0x0A	只写	设置亮度	ZT.SC-I2CMx-V1.1

0x1F	只读	模块代码版本	ZT.SC-I2CMx-V1.0
------	----	--------	------------------

使用时寄存器的宏定义设置

```
#define REG_CMD          0x01 //透传指令寄存器
#define REG_DAT          0x02 //透传数据寄存器
#define REG_RESET        0x03 //复位寄存器
    #define RESET_OLED    0x06 //OLED复位指令
#define REG_VERSION      0x1F //模块版本寄存器
#define REG_VCOMH        0x05 //设置OLED的高压
#define REG_STATUS       0x06 //状态寄存器
    #define STATUS_RUN     0x00 //模块处于正常状态，可对模块进行操作
    #define STATUS_SET_ADDRESS 0x02 //模块处于设置地址模式
    #define STATUS_BUSY    0x10 //模块处于忙状态，对模块操作无效
```

模块进入演示模式

模块复位的时候如果SDA为低电平，模块会进入演示模式，实现方法有两种：

方法1:

- 1.1 在断电状态下将SDA设置为低电平;
- 1.2 给模块接通电源
- 1.3 给模块供电后模块将会进入演示模式，进入演示模式后可以释放SDA
- 2.1 模块在通电的情况下设置RST为低电平
- 2.2 将SDA设置为低电平
- 2.3 设置RST为高电平（设置为高电平货释放悬空均可）
- 2.4 RST为高电平后模块将会进入演示模式，进入演示模式后可释放SDA管脚

如果按照以上两种方法模块都不能进入演示模式的，说明模块有问题，可以参阅模块故障排除部分

模块进入设置地址模式并重新设置地址

ZT.SEG8B4A036A具有任意设置模块本身地址的功能，此功能可以避免I2C总线上其它设备地址与本模块地址冲突。设置模块地址的步骤为：

一. 模块进入设置地址模式有两种方法

方法1:

- 1.1 在断电情况下将SCL设置为低电平;
- 1.2 给模块接通电源
- 1.3给模块供电后模块将会进入设置地址模式，进入设置地址模式后将SDA释放

方法2:

- 2.1 模块在通电的情况下设置RST为低电平
- 2.2 将SCL设置为低电平
- 2.3 设置RST为高电平（设置为高电平货释放悬空均可）
- 2.4 RST为高电平后模块将会进入设置地址模式，进入设置地址模式后将SCL管脚释放

如果按照以上两种方法模块都不能进入演示模式的，说明模块有问题，可以参阅模块故障排除部分

二. 设置新的地址

模块进入模块设置地址模式后，设置模块的新地址要向模块发送的指令为:

模块设置地址模式下的模块当前地址**0x51** + 模块的地址寄存器 **0x08** + 模块的新地址数据**0x27**(比如模块的地址要设置为**0x27**)

要发送的数据依次为: **0x51 0x08 0x27**

在ARDUINO平台下直接调用的函数为

```
ZT.Seg8b4a036aSetAddress (0x27);
```

函数返回值为0时说明设置成功，具体可参考库函数例子

在非ARDUINO平台下实现的代码如下

```
ZtSeg8b4a036aSetAddress(0x27);
```

模块每进入设置地址模式一次只能设置一次新地址,设置新地址后，模块要重启才有效

设置模块显示亮度

模块具有256*256个亮度级别可以调试，设置屏幕亮度格式为:

模块地址 + 数码管显示亮度寄存器**0x0A** + 亮度级别亮度数据 (0~0xFF) + 亮度级别暗度数据 (0~0xFF) +0+0

在ARDUINO平台下直接调用的函数为(假设要设置的亮度级别为**0x0F**, 暗度为**0x0F**)

```
ZT.Seg8b4a036aSetBrightness (OLED_ADDRESS, 0x0F, 0x0F);
```

在非ARDUINO平台下调用的函数为(假设要设置的亮度级别为**0x0F**, 暗度为**0x0F**)

```
ZtSeg8b4a036aSetBrightness (OLED_ADDRESS, 0x0F, 0x0F)
```

模块的唤醒设置

从省电的角度考虑，模块上电后默认为睡眠状态，要点亮模块必须先唤醒模块，

在AEDUINO平台下唤醒模块的函数调用为：

Zt.Seg8b4a036aUnSleep (address); 其中参数为模块的地址.

在非Arduino平台下调用的函数为：

ZtSeg8b4a036aUnSleep (address); 其中参数为模块的地址.

IIC总线上依次送出的操作数据为：模块地址 + 休眠寄存器**0x04** + 关闭休眠参数**0xA1**

模块进入休眠设置

如果为了省电暂时不想让模块显示数据，可以让模块进入睡眠模式，

在Arduino平台下唤醒模块的函数调用为：

Zt.Seg8b4a036aSleep (address); 其中参数为模块的地址.

在非Arduino平台下调用的函数为：

ZtSeg8b4a036aSleep (address); 其中参数为模块的地址.

IIC总线上依次送出的操作数据为：模块地址 + 休眠寄存器**0x04** + 关闭休眠参数**0xA5**

读取模块的当前状态

可以在任何时间读取模块的当前状态

Zt.Seg8b4a036aReadState (address); 其中参数为模块的地址.

在非Arduino平台下调用的函数为：

Zt Seg8b4a036aReadState (address); 其中参数为模块的地址.

IIC总线上依次送出的操作数据为：模块地址 + 状态寄存器**0x06** + 读取当前状态（1个字节）

函数的返回值为模块的当前状态，返回值对应的状态值：

```
#define STATUS_RUN           0x00 //正常运行状态
#define STATUS_SLEEP        0x01 //睡眠状态
#define STATUS_SET_ADDRESS  0x02 //设置地址模式
#define STATUS_TEST         0x04 //测试模式
#define STATUS_BUSY         0x10 //忙状态
```

返回为0时表明IIC总线有错误

读取模块的版本号

可工作模式下课读取模块的版本号

在Arduino平台下唤醒模块的函数调用为：

Zt. Seg8b4a036aReadVersion (address, versionbuff); 其中address参数为模块的地址. Versionbuff为所要读取的版本号的缓存
在非Arduino平台下调用的函数为:

Zt Seg8b4a036aReadVersion (address, versionbuff); 其中参数为模块的地址.

IIC总线上依次送出的操作数据为: 模块地址 + 状态寄存器**0x1F** + 读取版本号

透传数据的实现

透传数据就是将数据直接传递到4位数码管上然后显示出来, 每次操作的数据总长度位6个BYTE, 第一个BYTE位模块的地址, 第2个BYTE位数
据寄存器的地址0x02, 第3到第6个BYTE为在数码管上要显示的断码内容:

模块地址 + 透传数据寄存器**0x02** + 数据 (4个BYTE, 每个BYTE对应1位数码管)

段码显示的原理按照8段共阴常规数码管来对吗, 以下数据表位数字1~F对应的段码 (不含小数点)

```
unsigned char codetable[] =
```

```
{
```

```
0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F, 0x77, 0x7C,0x39,0x5E,0x79,0x71,0x00
```

```
};
```

例如: 如果要在4位数码管上显示1234, 则操作的数据位 模块地址 + 0x02 + codetable[1] + codetable[2] + codetable[3] + codetable[4]
4位数码管, 每位都带有一个小数点, 如果要将小数点显示出来, 在对应的位码上或上**0x80**即可, 如要显示1.234, 则总线上依次发送的数据为:
模块地址 + 0x02 + codetable[1]|0x80 + codetable[2] + codetable[3] + codetable[4]

数据透传的对应的函数为ZT.Seg8b4a036aDisplayBuff(ZTSEG8B4A036A_ADDR,buff); 第一个参数为模块的地址, 第二个参数为4个数码管对应的数据
码, 如果要显示123.4, 可赋值为

```
Buff[0]= codetable[1];
```

```
Buff[0]= codetable[2];
```

```
Buff[0]= codetable[3]|0x80;
```

```
Buff[0]= codetable[1];
```