

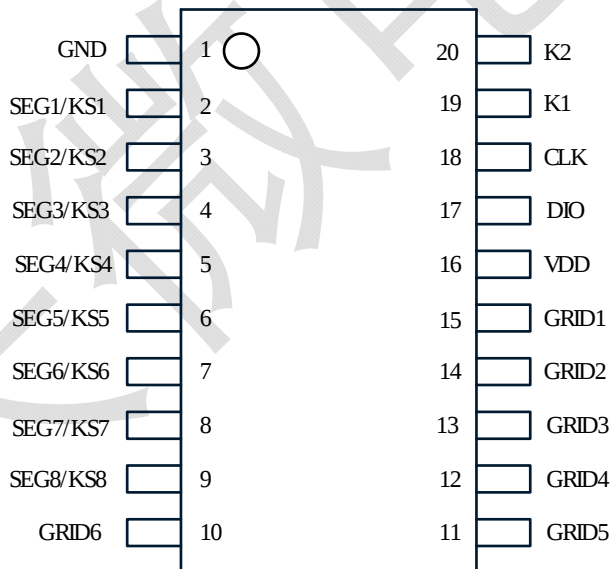
特性描述

TM1637 是一种带键盘扫描接口的LED（发光二极管显示器）驱动控制专用电路，内部集成有MCU 数字接口、数据锁存器、LED 高压驱动、键盘扫描等电路。本产品性能优良，质量可靠。主要应用于电磁炉、微波炉及小家电产品的显示屏驱动。采用DIP/SOP20的封装形式。

功能特点

- 采用功率CMOS 工艺
- 显示模式（8 段×6 位），支持共阳数码管输出
- 键扫描（8×2bit），增强型抗干扰按键识别电路
- 辉度调节电路（占空比 8 级可调）
- 两线串行接口（CLK，DIO）
- 振荡方式：内置RC 振荡（450KHz±5%）
- 内置上电复位电路
- 内置自动消隐电路
- 封装形式：DIP20/SOP20

管脚信息



管脚功能

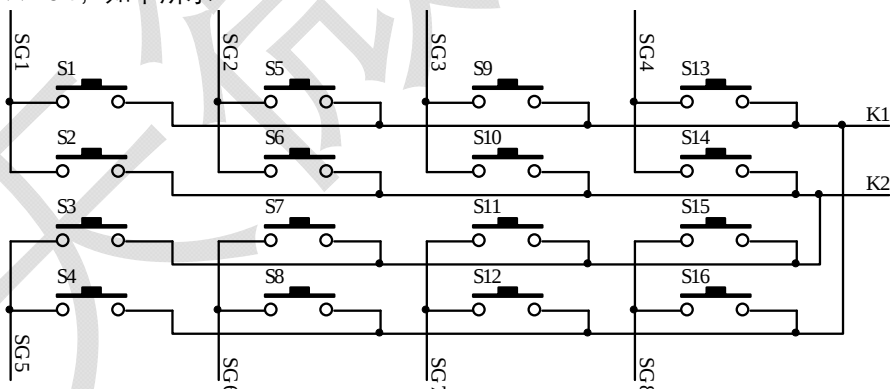
符号	管脚名称	管脚号	说明
DIO	数据输入/输出	17	串行数据输入/输出，输入数据在 SLCK 的低电平变化，在 SCLK 的高电平被传输，每传输一个字节芯片内部都将在第八个时钟下降沿产生一个 ACK
CLK	时钟输入	18	在上升沿输入/输出数据
K1~K2	键扫数据输入	19-20	输入该脚的数据在显示周期结束后被锁存
SG1~SG8	输出（段）	2-9	段输出（也用作键扫描），N 管开漏输出
GRID6~GRID1	输出（位）	10-15	位输出，P 管开漏输出
VDD	逻辑电源	16	5V ± 10%
GND	逻辑地	1	接系统地



在干燥季节或者干燥使用环境内，容易产生大量静电，静电放电可能会损坏集成电路，天微电子建议采取一切适当的集成电路预防处理措施，如果不正当的操作和焊接，可能会造成ESD损坏或者性能下降，芯片无法正常工作。

读键扫数据

键扫矩阵为 8×2bit，如下所示：



在有按键按下时，读键数据如下：

	SG1	SG2	SG3	SG4	SG5	SG6	SG7	SG8
K1	1110_11 11	0110_11 11	1010_11 11	0010_11 11	1100_11 11	0100_11 11	1000_11 11	0000_11 11
K2	1111_01 11	0111_01 11	1011_01 11	0011_01 11	1101_01 11	0101_01 11	1001_01 11	0001_01 11

注意：在无按键按下时，读键数据为：1111_1111，低位在前，高位在后。由于在电磁炉等厨房电器应用中，由于干扰较强，为改善这个问题，TM1637 采用负沿触发方式解决误触发现象，即所谓“跳键”现象。

显示寄存器地址和显示模式

该寄存器存储通过串行接口从外部器件传送到TM1637 的数据，地址00H-05H共6个字节单元，分别与芯片SGE和GRID管脚所接的LED灯对应，分配如下图：

写LED显示数据的时候，按照从显示地址从低位到高位，从数据字节的低位到高位操作。

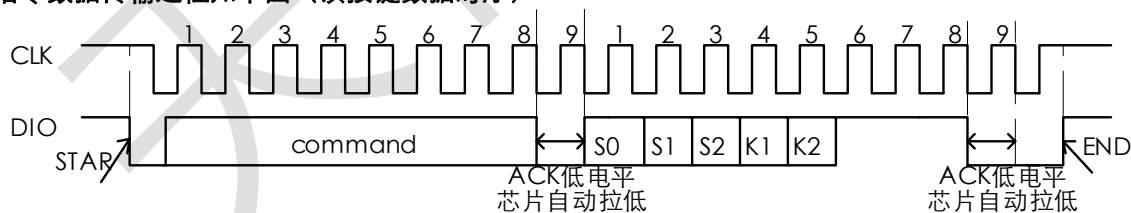
SEG8								
SEG7								
SEG6								
SEG5								
SEG4								
SEG3								
SEG2								
SEG1								
xxHL (低四位)				xxHU(高四位)				
B0	B1	B2	B3	B4	B5	B6	B7	
00HL				00HU				GRID1
01HL				01HU				GRID2
02HL				02HU				GRID3
03HL				03HU				GRID4
04HL				04HU				GRID5
05HL				05HU				GRID6

接口说明

微处理器的数据通过两线总线接口和 TM1637 通信，在输入数据时当 CLK 是高电平时，DIO 上的信号必须保持不变；只有 CLK 上的时钟信号为低电平时，DIO 上的信号才能改变。数据输入的开始条件是 CLK 为高电平时，DIO 由高变低；结束条件是 CLK 为高时，DIO 由低电平变为高电平。

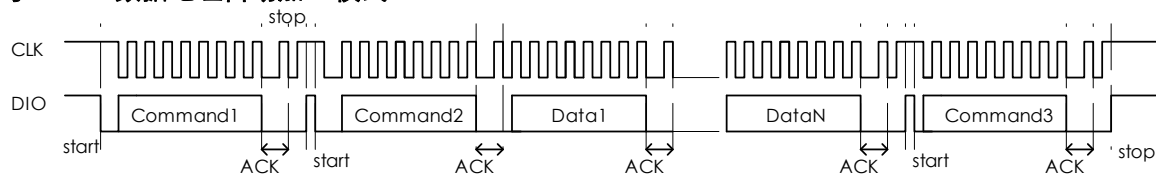
TM1637 的数据传输带有应答信号 ACK，当传输数据正确时，会在第八个时钟的下降沿，芯片内部会产生一个应答信号 ACK 将 DIO 管脚拉低，在第九个时钟结束之后释放 DIO 口线。

1、指令数据传输过程如下图（读按键数据时序）



Command: 读按键指令；S0、S1、S2、K1、K2 组成按键信息编码，S0、S1、S2 为 SGn 的编码，K1、K2 为 K1 和 K2 键的编码，读按键时，时钟频率应小于 250K，先读低位，后读高位。

2、写 SRAM 数据地址自动加 1 模式



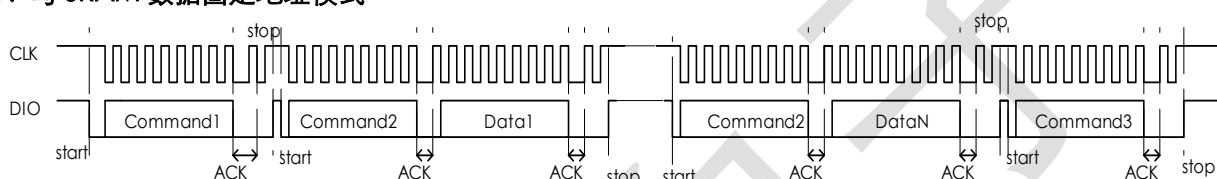
Command1: 设置数据

Command2: 设置地址

Data1~N: 传输显示数据

Command3: 控制显示

3、写 SRAM 数据固定地址模式



Command1: 设置数据

Command2: 设置地址

Data1~N: 传输显示数据

Command3: 控制显示

数据指令

指令用来设置显示模式和LED 驱动器的状态。

在CLK下降沿后由DIO输入的的第一个字节作为一条指令。经过译码，取最高B7、B6两位比特位以区别不同的指令。

B7	B6	指令
0	1	数据命令设置
1	0	显示控制命令设置
1	1	地址命令设置

如果在指令或数据传输时发送STOP命令，串行通讯被初始化，并且正在传送的指令或数据无效（之前传送的指令或数据保持有效）

1、数据命令设置

该指令用来设置数据写和读，B1和B0位不允许设置01或11。

MSB

LSB

B7	B6	B5	B4	B3	B2	B1	B0	功能	说明
0	1	无关项，填 0				0	0	数据读写模式设置	写数据到显示寄存器
0	1					1	0		读键扫数据
0	1				0			地址增加模式设置	自动地址增加
0	1				1				固定地址
0	1			0				测试模式设置（内部使用）	普通模式
0	1			1					测试模式

2、地址命令设置

MSB

LSB

B7	B6	B5	B4	B3	B2	B1	B0	显示地址
1	1	无关项，填 0		0	0	0	0	00H
1	1			0	0	0	1	01H
1	1			0	0	1	0	02H
1	1			0	0	1	1	03H
1	1			0	1	0	0	04H
1	1			0	1	0	1	05H

该指令用来设置显示寄存器的地址；如果地址设为0C6H 或更高，数据被忽略，直到有效地址被设定；上电时，地址默认设为00H。

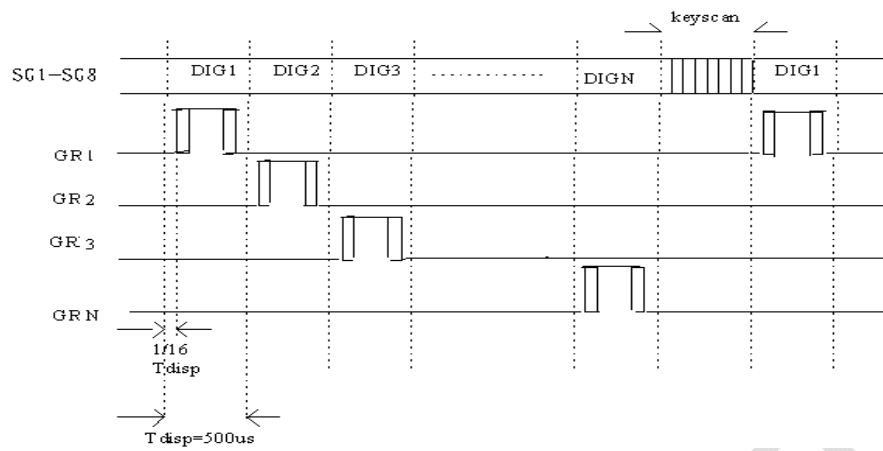
3、显示控制

MSB

LSB

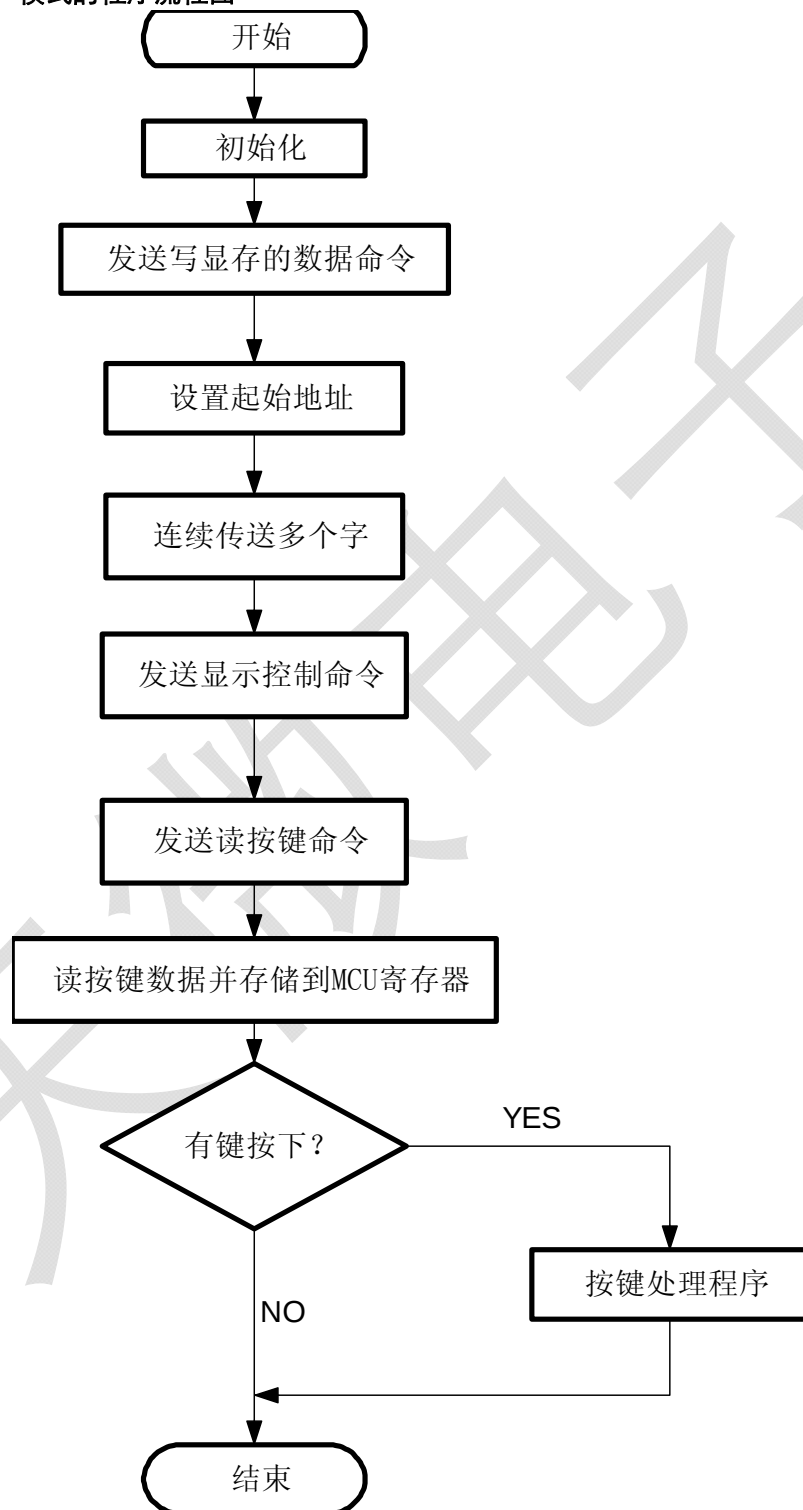
B7	B6	B5	B4	B3	B2	B1	B0	功能	说明
1	0	无关项，填 0			0	0	0	消光数量设置	设置脉冲宽度为 1/16
1	0				0	0	1		设置脉冲宽度为 2/16
1	0				0	1	0		设置脉冲宽度为 4/16
1	0				0	1	1		设置脉冲宽度为 10/16
1	0				1	0	0		设置脉冲宽度为 11/16
1	0				1	0	1		设置脉冲宽度为 12/16
1	0				1	1	0		设置脉冲宽度为 13/16
1	0				1	1	1		设置脉冲宽度为 14/16
1	0			0				显示开关设置	显示关
1	0			1					显示开

显示和键扫周期

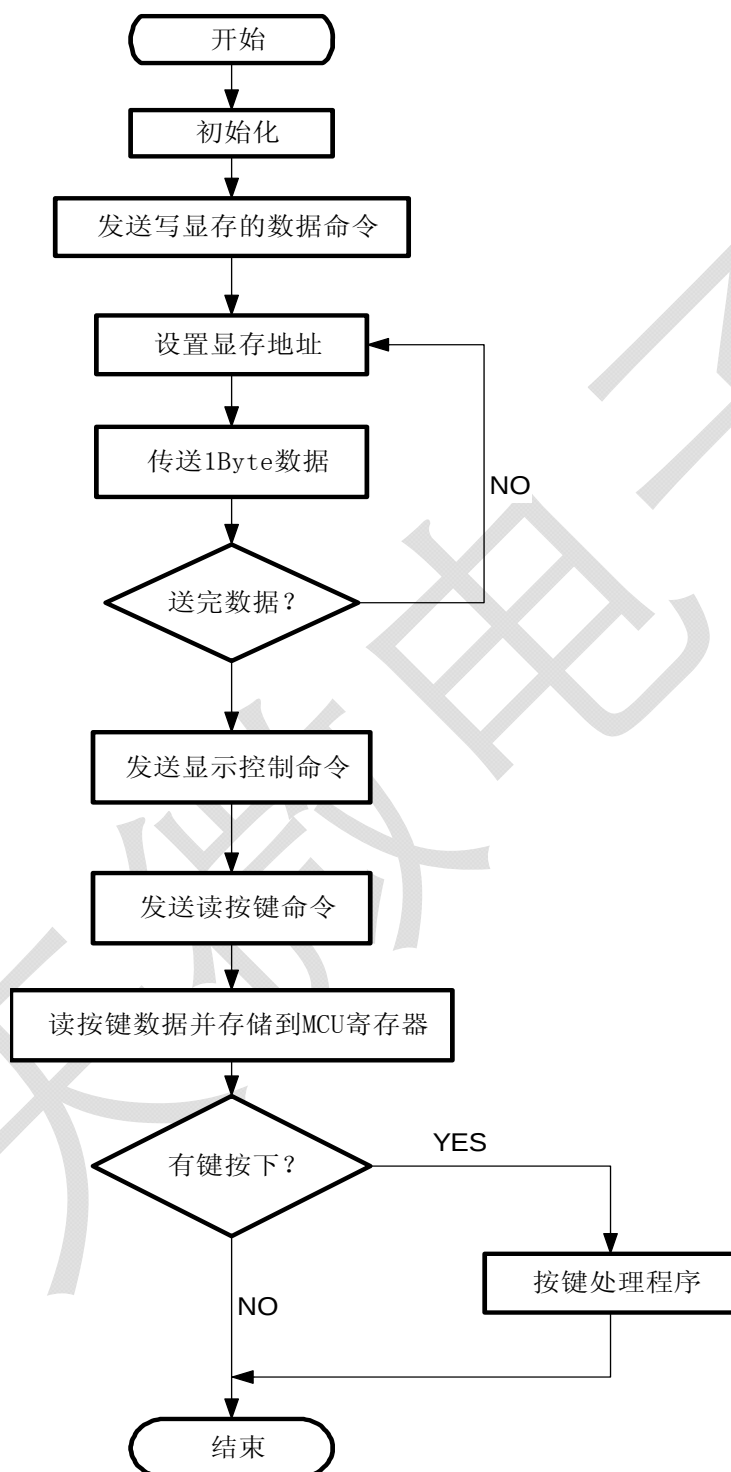


程序流程图

1、采用地址自动加一模式的程序流程图



2、采用固定地址的程序设计流程图



参考程序

```
/*
*版权信息:    深圳天微电子
*文件名:      TM1637
*当前版本:    1.0
*单片机型号:  AT89S52
*开发环境:    Keil uVision3
*晶震频率:    11.0592M
*程序功能:    把 TM1637 所有显示寄存器地址全部写满数据 0xff，并开显示，然后再读按键值。
*/
#include<reg52.h>
#include<intrins.h>
//定义端口
sbit clk = P1^2;
sbit dio = P1^1;

///=====
void Delay_us(unsigned int i)    //nus 延时
{
    for(;i>0;i--)
        _nop_();
}

///=====
void I2CStart(void)              //1637 开始
{
    clk = 1;
    dio = 1;
    Delay_us(2);
    dio = 0;
}

///=====
void I2Cask(void)                //1637 应答
{
    clk = 0;
    Delay_us(5);                  //在第八个时钟下降沿之后延时 5us，开始判断 ACK 信号
    while(dio);
    clk = 1;
    Delay_us(2);
    clk=0;
}

///=====
```

```
void I2CStop(void)                // 1637 停止
{
    clk = 0;
    Delay_us(2);
    dio = 0;
    Delay_us(2);
    clk = 1;
    Delay_us(2);
    dio = 1;
}

//=====================================================
void I2CWrByte(unsigned char oneByte) //写一个字节
{
    unsigned char i;
    for(i=0;i<8;i++)
    {
        clk = 0;
        if(oneByte&0x01)           //低位在前
        {
            dio = 1;
        }
        else
        {
            dio = 0;
        }
        Delay_us(3);
        oneByte=oneByte>>1;
        clk=1;
        Delay_us(3);
    }
}

//-----
unsigned char ScanKey(void)        //读按键
{
    unsigned char rekey,rkey,i;
    I2CStart();
    I2CWrByte(0x42);              //读按键命令
    I2Cask();
    dio=1;                        // 在读按键前拉高数据线
    for(i=0;i<8;i++)              //从低位开始读
    {
        clk=0;
        rekey=rekey>>1;
    }
}
```

```

        Delay_us(30);
        clk=1;
        if(dio)
        {
            rekey=rekey | 0x80;
        }
        else
        {
            rekey=rekey | 0x00;
        }
        Delay_us(30);
    }
    I2Cask();
    I2CStop();
    return (rekey);
}

//=====================================================
void SmgDisplay(void)                //写显示寄存器
{
    unsigned char i;
    I2CStart();
    I2CWrByte(0x40);                // 40H 地址自动加 1 模式,44H 固定地址模式,本程序采
    用自加 1 模式
    I2Cask();
    I2CStop();
    I2CStart();
    I2CWrByte(0xc0);                //设置首地址,
    I2Cask();

    for(i=0;i<6;i++)                //地址自加, 不必每次都写地址
    {
        I2CWrByte(0xff);            //送数据
        I2Cask();
    }
    I2CStop();

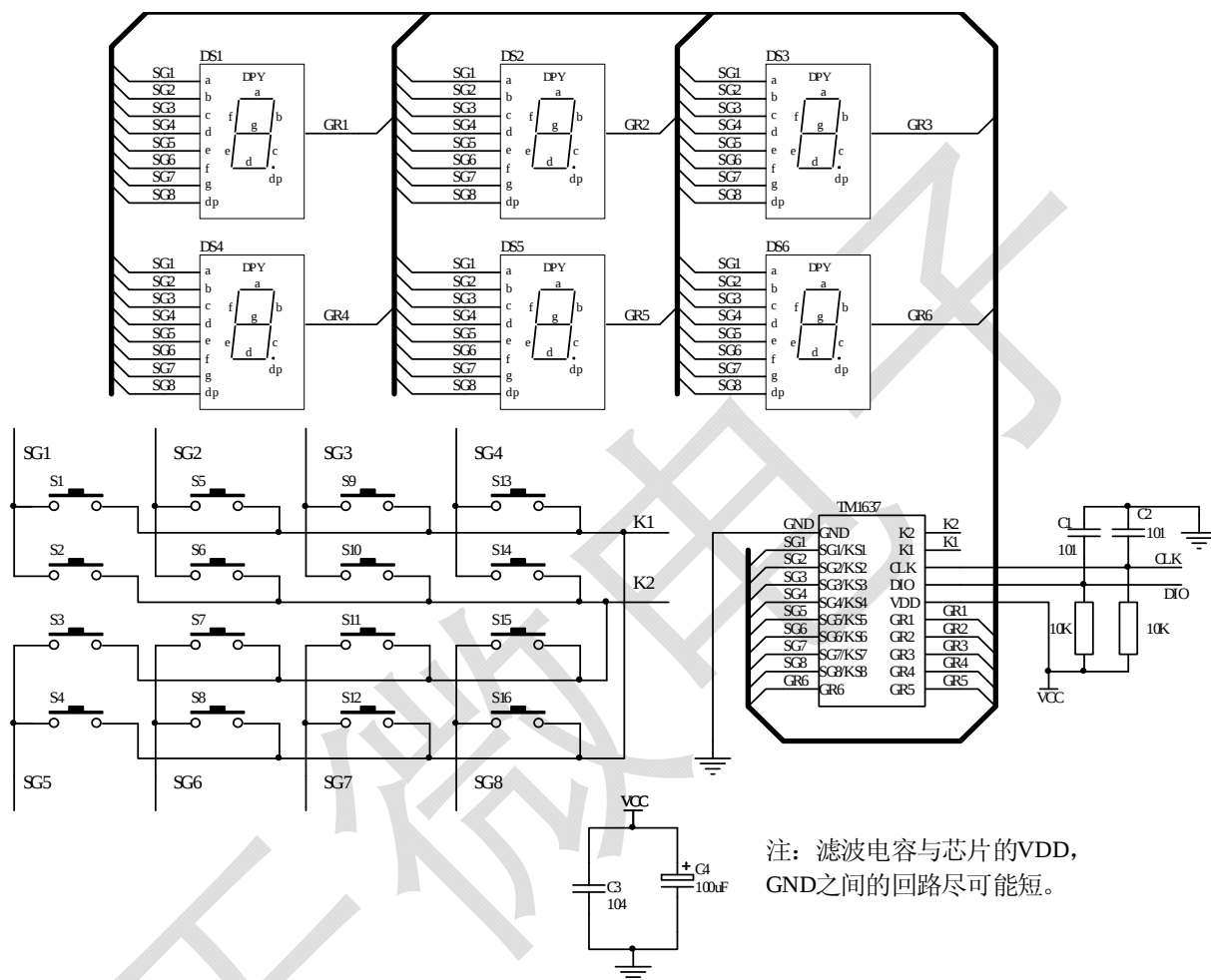
    I2CStart();
    I2CWrByte(0x8f);                //开显示 , 最大亮度
    I2Cask();
    I2CStop();
}
//=====================================================

```

```
void init()  
{  
    //初始化略  
}  
  
///  
void main(void)  
{  
    unsigned char keydate;  
    init(); //初始化  
    SmgDisplay(); //写寄存器并开显示  
    while(1)  
    {  
        keydate=Scankey(); //读按键值 ， 读出的按键值不作处理。  
    }  
}  
///  
//=====end=====
```

硬件连接图

电路图中所接数码管为共阳数码管



电气参数:
1、极限参数 (Ta = 25°C, Vss = 0 V)

参数	符号	范围	单位
逻辑电源电压	VDD	-0.5 ~ +7.0	V
逻辑输入电压	VII	-0.5 ~ VDD + 0.5	V
LED SEG 驱动输出电流	IO1	-200	mA
LED GRID 驱动输出电流	IO2	+20	mA
功率损耗	PD	400	mW
工作温度	Topt	-40 ~ +85	°C
储存温度	Tstg	-65 ~ +150	°C

2、正常工作范围 (Ta = -40~+85°C, Vss = 0 V)

参数	符号	最小	典型	最大	单位	测试条件
逻辑电源电压	VDD		5		V	-
高电平输入电压	VIH	0.7 VDD	-	VDD	V	-
低电平输入电压	VIL	0	-	0.3 VDD	V	-

3、电气特性 (Ta = -40~+85°C, VDD = 4.5 ~ 5.5 V, Vss = 0 V)

参数	符号	最小	典型	最大	单位	测试条件
高电平输出电流	Ioh1	-20	-25	-40	mA	GRID1~GRID6, Vo = vdd-2V
	Ioh2	-20	-30	-50	mA	GRID1~GRID6, Vo = vdd-3V
低电平输出电流	IOL1	80	140	-	mA	SEG1~SEG8 Vo=0.3V
低电平输出电流	Idout	4	-	-	mA	VO = 0.4V, dout
高电平输出电流容许量	ItoIsg	-	-	5	%	VO = VDD - 3V, GRID1~GRID6

输出下拉电阻	RL		10		KΩ	K1~K2
输入电流	II	-	-	±1	μA	VI = VDD / VSS
高电平输入电压	VIH	0.7 VDD	-		V	CLK, DIN
低电平输入电压	VIL	-	-	0.3 VDD	V	CLK, DIN
滞后电压	VH	-	0.35	-	V	CLK, DIN
动态电流损耗	IDDdyn	-	-	5	mA	无负载, 显示关

4、开关特性 (Ta = -40~+85℃, VDD = 4.5 ~ 5.5 V)

参数	符号	最小	典型	最大	单位	测试条件	
振荡频率	fosc	-	450	-	KHz		
传输延迟时间	tPLZ	-	-	300	ns	CLK → DIO CL = 15pF, RL = 10K Ω	
	tPZL	-	-	100	ns		
上升时间	TTZH 1	-	-	2	μs	CL = 300p F	GRID1~ GRID6
	TTZH 2	-	-	0.5	μs		SEG1~ SEG8
下降时间	TTHZ	-	-	120	μs	CL = 300pF, Segn, Gridn	
最大时钟频率	Fmax	-	-	500	KHz	占空比50%	
输入电容	CI	-	-	15	pF	-	

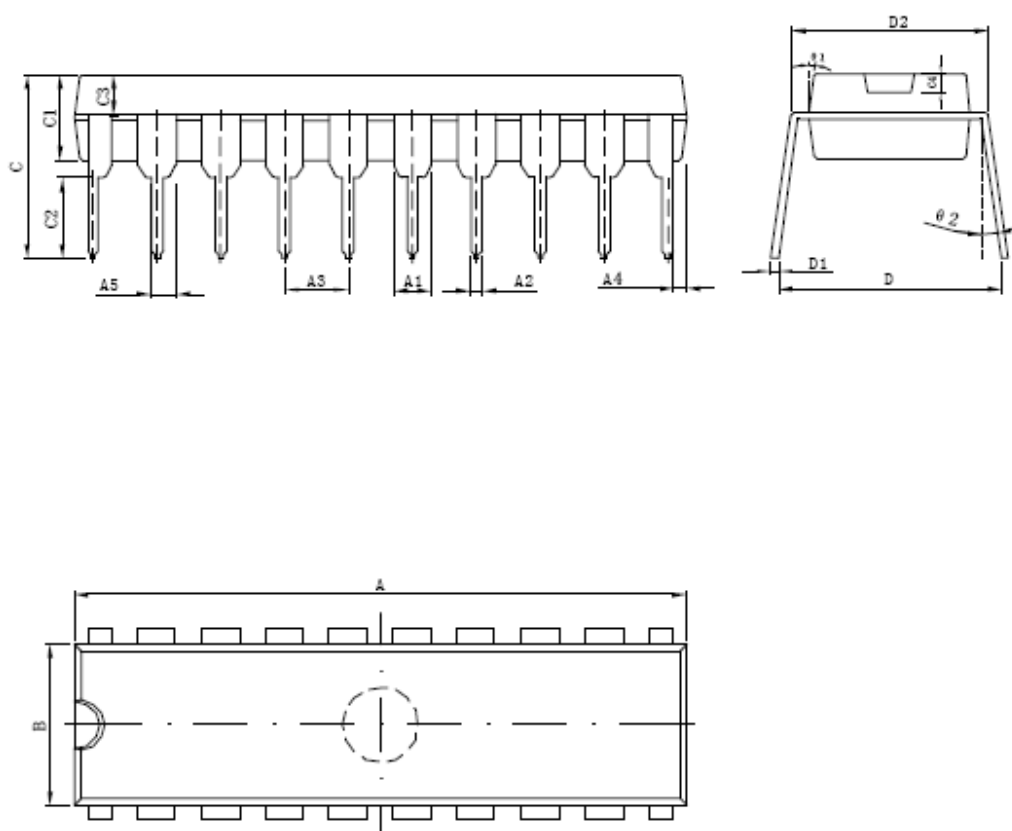
5、时序特性 (Ta = -40 ~+85℃, VDD = 4.5 ~ 5.5 V)

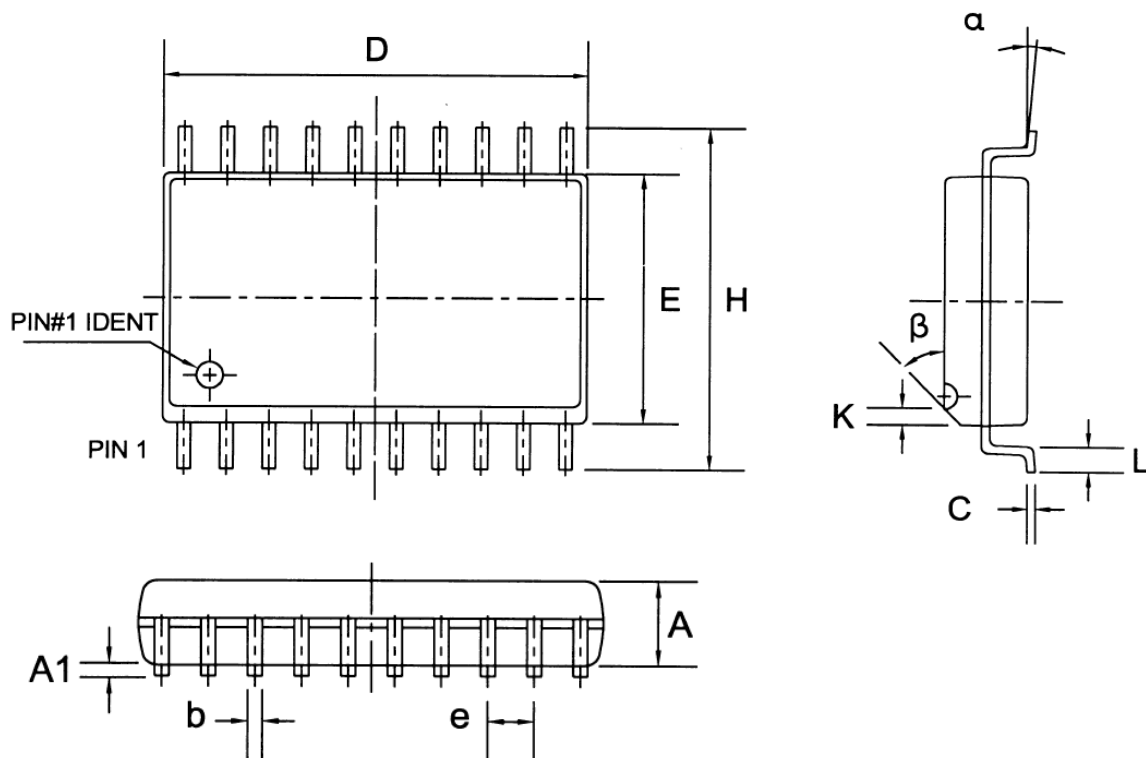
参数	符号	最小	典型	最大	单位	测试条件
时钟脉冲宽度	PWCLK	400	-	-	ns	-
数据建立时间	tSETUP	100	-	-	ns	-
数据保持时间	tHOLD	100	-	-	ns	-
等待时间	tWAIT	1	-	-	μs	CLK ↑ → CLK ↓

IC封装示意图

DIP20

尺寸	最小 (mm)	最大 (mm)	尺寸	最小 (mm)	最大 (mm)
标注			标注		
A	24.50	24.70	C2	2.9	
A1	1.40TYP		C3	1.56TYP	
A2	0.43	0.57	C4	0.80TYP	
A3	2.54TYP		D	7.87	8.60
A4	0.62TYP		D1	0.20	0.33
A5	0.95TYP		D2	7.62	7.87
B	6.3	6.5	θ1	8° TYP	
C	7.5TYP		θ2	5° TYP	
C1	3.30	3.50			





Symbol	Dimensions In Millimeters			Dimensions In Inches		
	Min	Nom	Max	Min	Nom	Max
A	2.15	2.35	2.55	0.085	0.093	0.100
A1	0.05	0.15	0.25	0.002	0.006	0.010
b	—	0.40	—	—	0.016	—
C	—	0.25	—	—	0.010	—
D	12.40	12.70	13.00	0.488	0.500	0.512
E	7.40	7.65	7.90	0.291	0.301	0.311
e	—	1.27	—	—	0.050	—
H	10.15	10.45	10.75	0.400	0.411	0.423
K	—	0.50	—	—	0.020	—
L	0.60	0.80	1.00	0.024	0.031	0.039
α	0°	—	8°	0°	—	8°
β	—	45°	—	—	45°	—

修改说明

版本	修改日期	修改说明
Ver1.0	2011-06-28	初版发行
Ver2.0	2011-09-22	1.更改对 ack 信号的说明 2.更改例程中 ack 信号的程序错误
Ver1.2	2012-08-12	1、修改排版格式 2、修改例程中 STOP 的时序 3、修改关于 ACK 信号的描述