



HD1668

概述

HD1668 是一种带键盘扫描接口的 LED 驱动控制专用电路，内部集成有 MCU 数字接口、数据锁存器、LED 驱动、键盘扫描等电路。

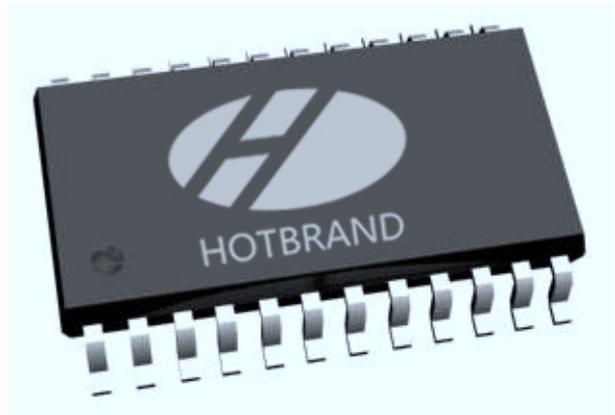
特性说明

- ◆ 采用 CMOS 工艺
- ◆ 显示模式：7 位×10 段
- ◆ 按键扫描：10×2 位
- ◆ 辉度调节电路（占空比 8 级可调）
- ◆ 串行接口（CLK，STB，DI/O）
- ◆ 振荡方式：内置 RC 振荡
- ◆ 内置上电复位电路
- ◆ 封装形式：SOP24

应用领域

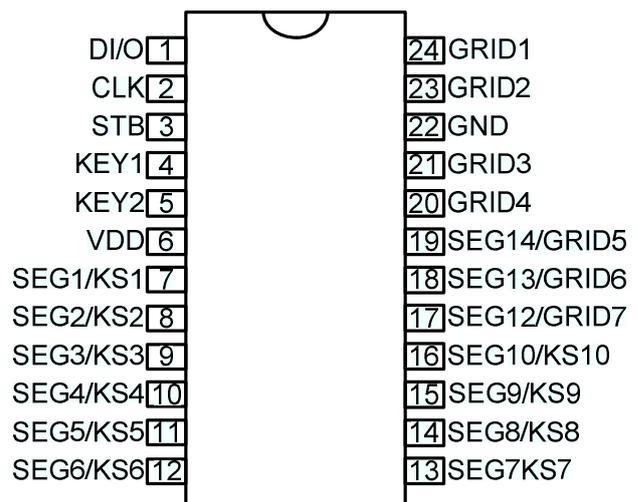
- ◆ VCD/DVD/DVB 显示
- ◆ 电磁炉显示
- ◆ 电饭煲显示
- ◆ 空调显示

封装图

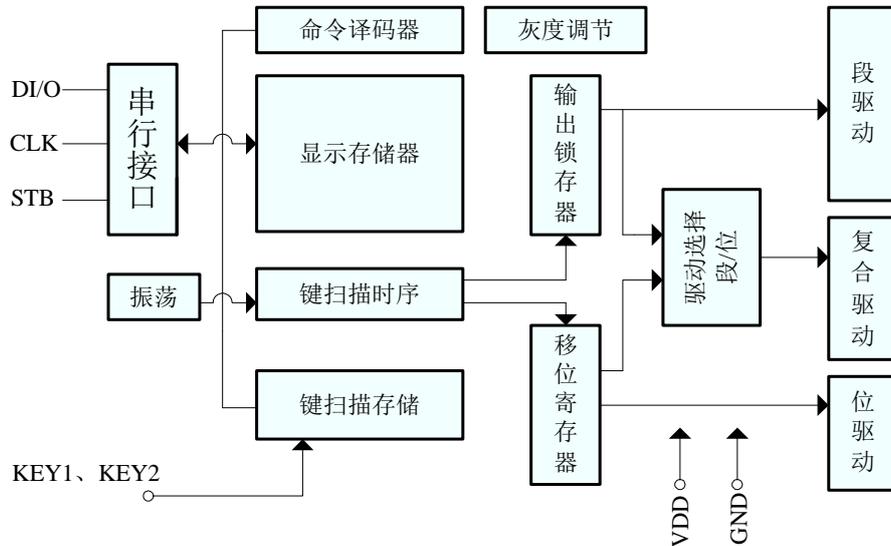


SOP24

管脚定义



内部功能框图



管脚定义说明

符号	管脚名称	管脚号	说明
DI/O	数据输入/输出	1	在时钟下降沿输出串行数据，从低位开始。在时钟上升沿输入串行数据，从低位开始
CLK	时钟输入	2	在上升沿读取串行数据，下降沿输出数据
STB	数据传输控制脚	3	在下降沿初始化串行接口，随后等待接收指令，STB 为低后的第一个字节作为指令。当 STB 为高时，CLK 被忽略
KEY1—KEY2	键扫信号输入	4—5	键扫信号在显示周期结束后被锁存
SEG1/KS1—SEG10/KS10	段输出	7—16	P 管开漏输出，管脚也用作键扫描
SEG12/GRID7—SEG14/GRID5	段/位输出	17—19	段/位驱动输出
VDD	逻辑电源	6	5V±10%
GRID3—GRID4	位输出	20—21	N 管开漏输出
GND	逻辑地	22	芯片地
GRID1—GRID2	位输出	23—24	N 管开漏输出



电气参数

极限参数 (Ta = 25°C)

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	V _{CLK} , V _{STB} , V _{DI/O}	-0.5——VDD + 0.5	V
LED SEG 驱动输出电流	I _{SEG}	-50	mA
LED GRID 驱动输出电流	I _{GRID}	500	mA
工作温度	T _{OPT}	-40——+80	°C
储存温度	T _{STG}	-65——+150	°C

电气特性 (Ta = 25°C)

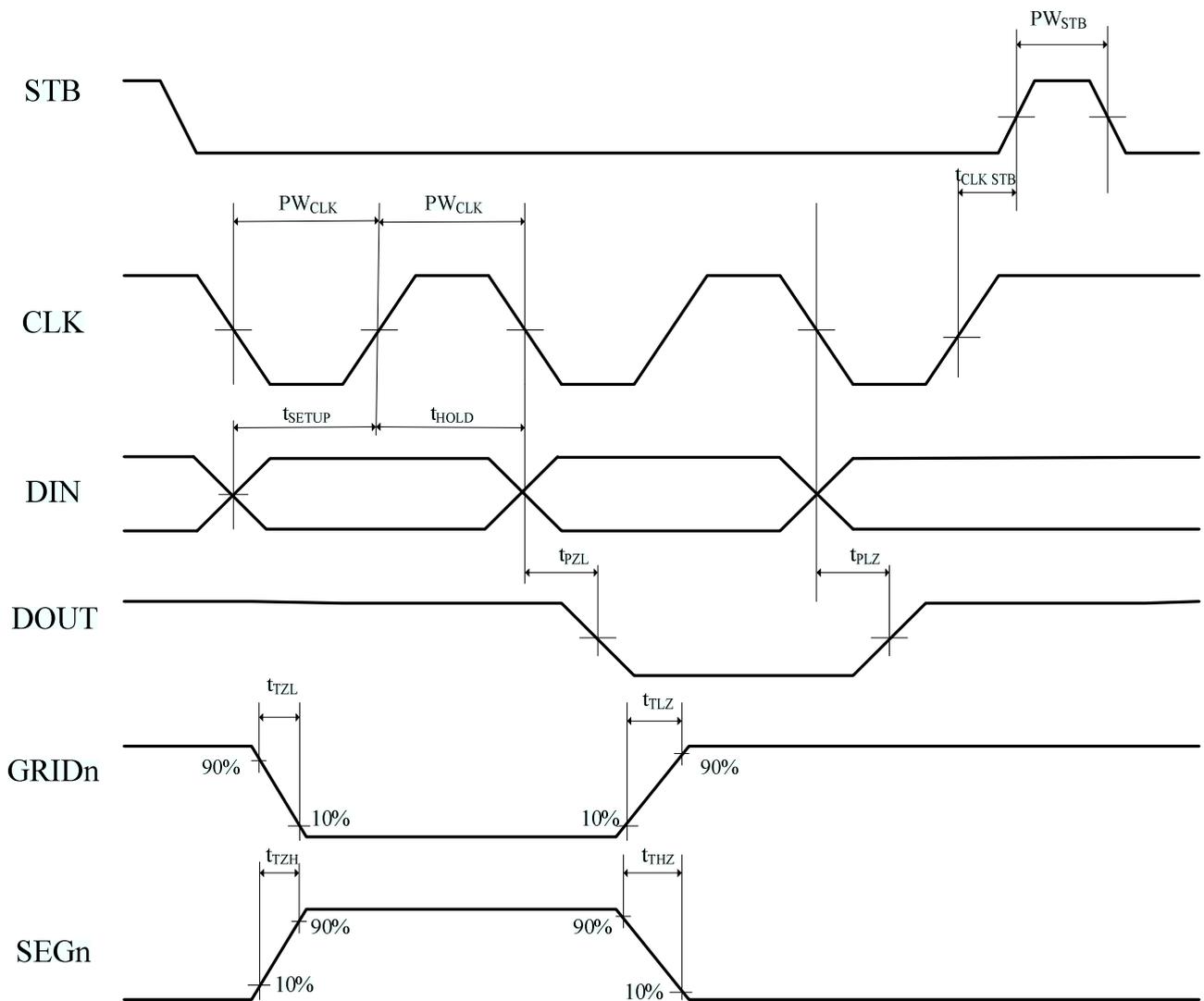
参数	符号	测试条件	最小	典型	最大	单位
逻辑电源电压	VDD		4.5	5.0	5.5	V
高电平输入电压	V _{IH}		0.7VDD	-	VDD	V
低电平输入电压	V _{IL}		0	-	0.3VDD	V
静态电流	IDD	VDD=5.0V, 无负载, 显示关	-	-	1.2	mA
SEG 驱动电流	I _{SEG}	VDD=5.0V, SEG 接 2Ω 电阻对 GND	-	-47	-	mA
GRID 驱动电流	I _{GRID}	VDD=5.0V, GRID 接 2Ω 电阻对 VDD	-	650	-	mA
KEY 下拉电阻	R _{KEY}	KEY1、KEY2	10	-	25	KΩ
GRID 振荡频率	f _{GRID}	VDD=5.0V	-	210	-	Hz
传输延迟时间	t _{PLZ}	VDD=5.0V, CLK→DI/O	-	75	-	ns
	t _{PZL}	R _{UP} = 10 KΩ, C _L =15pF	-	50	-	ns
上升时间	t _{TZH(SEG)}	VDD=5.0V, SEGn 接 100Ω 下拉电阻, GRIDn 接 10Ω 上拉电阻, C _L =15pF	-	6	-	ns
	t _{TLZ(GRID)}		-	14	-	ns
下降时间	t _{THZ(SEG)}	C _L =15pF	-	10	-	ns
	t _{TZL(GRID)}		-	27	-	ns
最大时钟频率	f _{CLK(max)}	占空比 50%	-	-	1	MHz



时序特性 (VDD=5.0V, Ta = 25°C)

参数	符号	测试条件	最小	典型	最大	单位
时钟脉冲宽度	PW _{CLK}		500	-	-	ns
选通脉冲宽度	PW _{STB}		1	-	-	μs
数据建立时间	t _{SETUP}		100	-	-	ns
数据保持时间	t _{HOLD}		100	-	-	ns
CLK → STB 时间	t _{CLK STB}	CLK↑→ STB↑	1	-	-	μs

时序波形图



*注: DIN 和 DOUT 为芯片内部 PAD 位, 通过封装合并成 DI/O。

功能描述

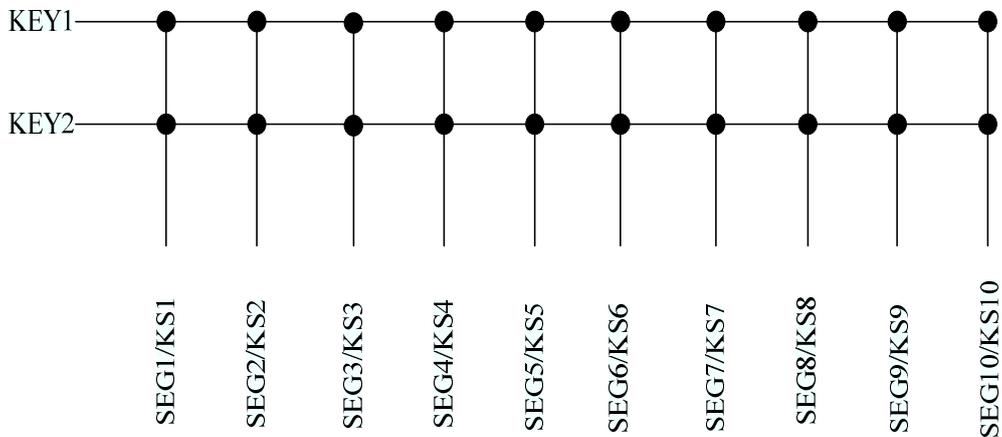
◆ 显示寄存器地址和显示模式

该寄存器存储通过串行接口从外部器件传送到 HD1668 的数据，地址分配如下：

SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	X	SEG12	SEG13	SEG14	X	X	
b0	b1	b2	b3	b4	b5	b6	b7	b0	b1	b2	b3	b4	b5	b6	b7	显示字节
显示地址 00H								显示地址 01H								GRID1
显示地址 02H								显示地址 03H								GRID2
显示地址 04H								显示地址 05H								GRID3
显示地址 06H								显示地址 07H								GRID4
显示地址 08H								显示地址 09H								GRID5
显示地址 0AH								显示地址 0BH								GRID6
显示地址 0CH								显示地址 0DH								GRID7

◆ 键扫描和按键扫描数据寄存器

按键扫描矩阵为 10×2bit，如下所示：



按键数据储存地址如下所示，用读指令读取，从最低位开始读取：

b0	b1	b2	b3	b4	b5	b6	b7	按键字节
KEY1	KEY2	-	KEY1	KEY2	-	-	-	KEY 端口
SEG1/KS1		X	SEG2/KS1		X	X	X	读取字节顺序， 从上到下
SEG3/KS1		X	SEG4/KS1		X	X	X	
SEG5/KS1		X	SEG6/KS1		X	X	X	
SEG7/KS1		X	SEG8/KS1		X	X	X	
SEG9/KS1		X	SEG10/KS1		X	X	X	

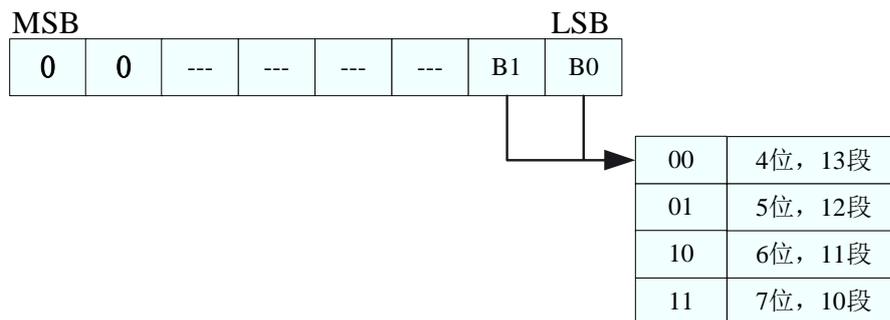


指令说明

指令用来设置显示模式和 LED 驱动器的状态。在 STB 下降沿后由 DI/O 输入的第一个字节作为一条指令。如果在指令或数据传输时 STB 被置为高电平，串行通讯将被初始化，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）。

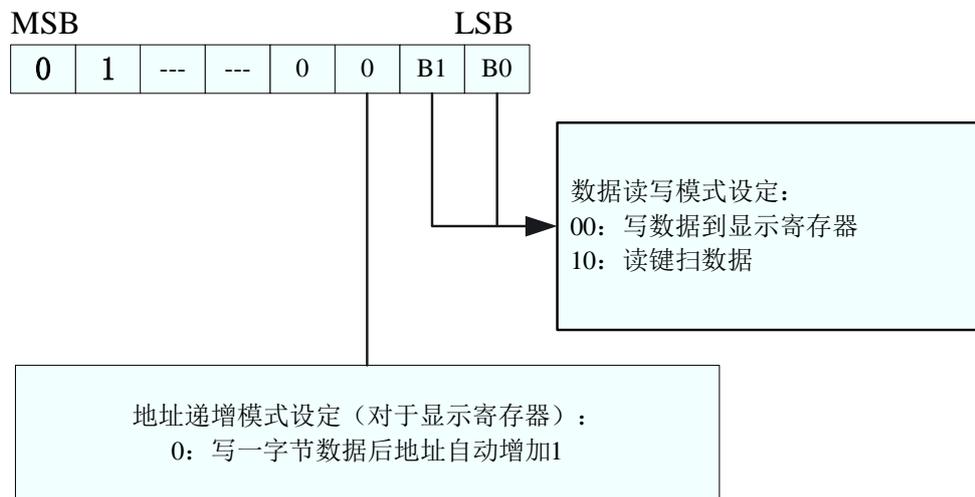
◆ 显示模式设置

该指令用来设置选择段和位的个数（4—7 位，10—13 段）。当指令执行时，显示被强制终止，同时键扫信号也将停止。如要重新显示，显示开/关指令“ON”必需被执行，当同样的模式被设置时，则上述情况不会发生。



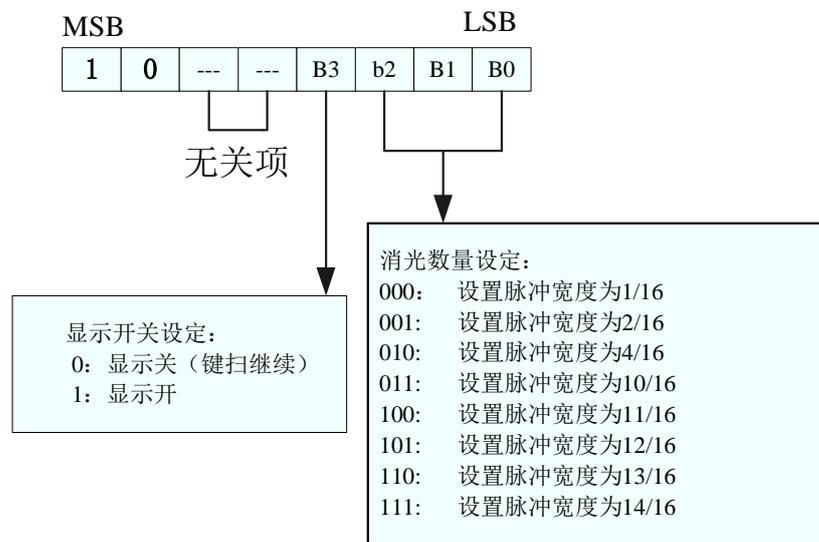
◆ 数据设置

该指令用来设置数据写和读。

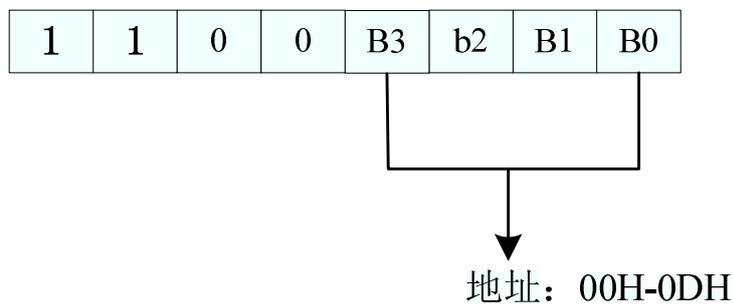




◆ 显示控制

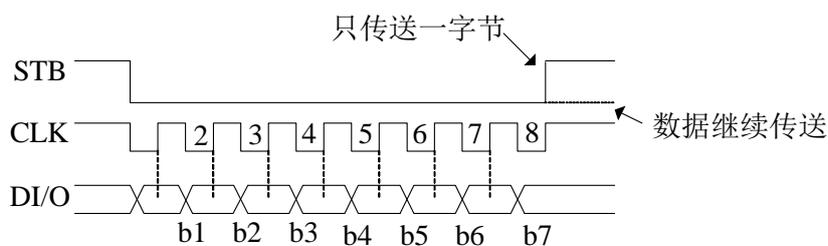


◆ 地址设置命令



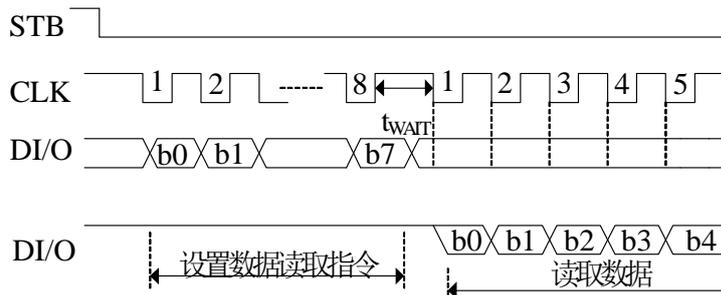
◆ 串行数据传输格式

数据接收 (写数据)





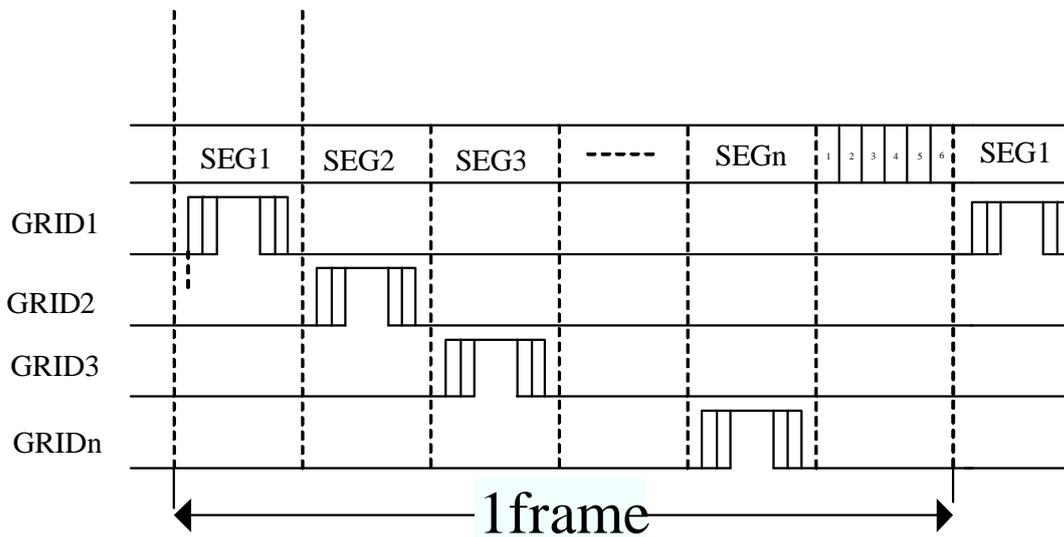
数据读取



注: $t_{WAIT} \geq 1\mu s$

DI/O 管脚用作输出功能时, 为 N 管开漏输出, 所以该脚要连接一个外部上拉电阻 (1KΩ 到 10KΩ)。

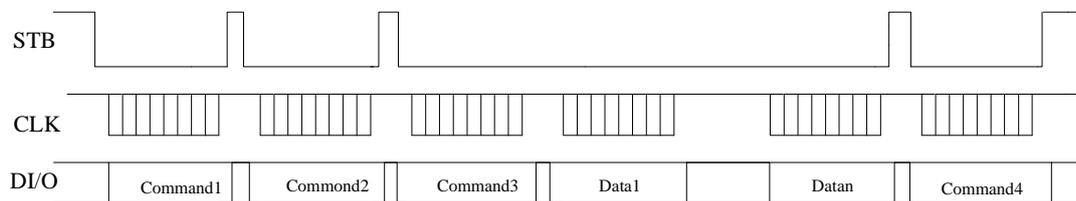
◆ 显示和键扫周期



注意: 2 frame 为一按键扫描周期

◆ 应用时串行数据传输

地址增加模式



Command1: 设置显示模式

Command2: 设置数据

Command3: 设置地址

Data1~n : 传输显示数据 (最多 14bytes)

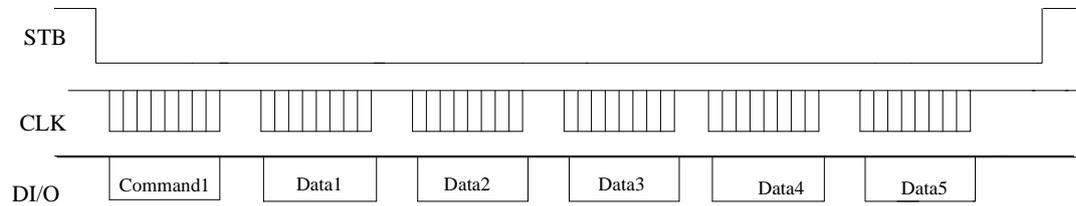
Command4: 控制显示

注: 显示寄存器地址范围为: 00H-0DH, 总共 14 个字节。连续地址模式写时, 如果地址为 00H 时, 后面



紧跟 14 个显示数据字节；地址为 01H 时，后面紧跟 13 个；以此类推，如果地址为 n，那么紧跟后面的显示字节为 14-n 个。

读按键模式



Command1: 设置读按键模式

DATA1~n: 按键数据



参考程序

```
/******
```

功能： 控制 HD1668 使数码管全点亮

单片机型号： AT89C2051

开发环境： Keil uVision3

晶振频率： 12MHz

```
*****/
```

```
#include<reg52.h>
```

```
#include<intrins.h>
```

```
typedef unsigned char uchar;
```

```
//定义控制端口
```

```
sbit DIO =P1^7;
```

```
sbit CLK =P1^6;
```

```
sbit STB =P1^5;
```

```
//定义数据
```

```
uchar KEY[5]={0}; //存储按键值用
```

```
*****/
```

```
//功能：  向 HD1668 发送 8 位数据，从低位开始
```

```
*****/
```

```
void send_8bit(uchar dat)
```

```
{
```

```
  uchar i;
```

```
  for (i=0;i<8;i++)
```

```
  {
```

```
    CLK=0;
```

```
    if(dat&0x01)
```

```
      DIO=1;
```

```
    else
```

```
      DIO=0;
```

```
    CLK=1;
```

```
    dat=dat>>1;
```

```
  }
```

```
}
```

```
*****/
```



//功能：向 HD1668 发送命令

/*

*/

void command(uchar com)

```
{  
    STB=1;  
    _nop_();  
    STB=0;  
    send_8bit(com);  
}
```

/*

*/

//功能：数码管全显示

/*

*/

void display()

```
{  
    uchar i;  
    command(0x03);           //设置为 7*10 模式  
    command(0x40);          //设置数据命令，采用地址自动加 1 模式  
    command(0xc0);          //设置显示地址，从 00H 开始  
    for(i=0;i<14;i++)  
    {  
        send_8bit(0xff);    //发送 14 字节全 1 数据  
    }  
    command(0x8f);          //显示控制命令，打开显示并设置为最亮  
    STB=1;  
}
```

/*

*/

//功能：读取按键数据并存入 KEY[]数组中，从低字节、低位开始

/*

*/

void read_key()

```
{  
    uchar i,j;  
    command(0x42);          //读按键命令  
    DIO=1;                  //释放 DIO 数据总线  
    for(j=0;j<5;j++)       //读取 5 个字节按键数据
```

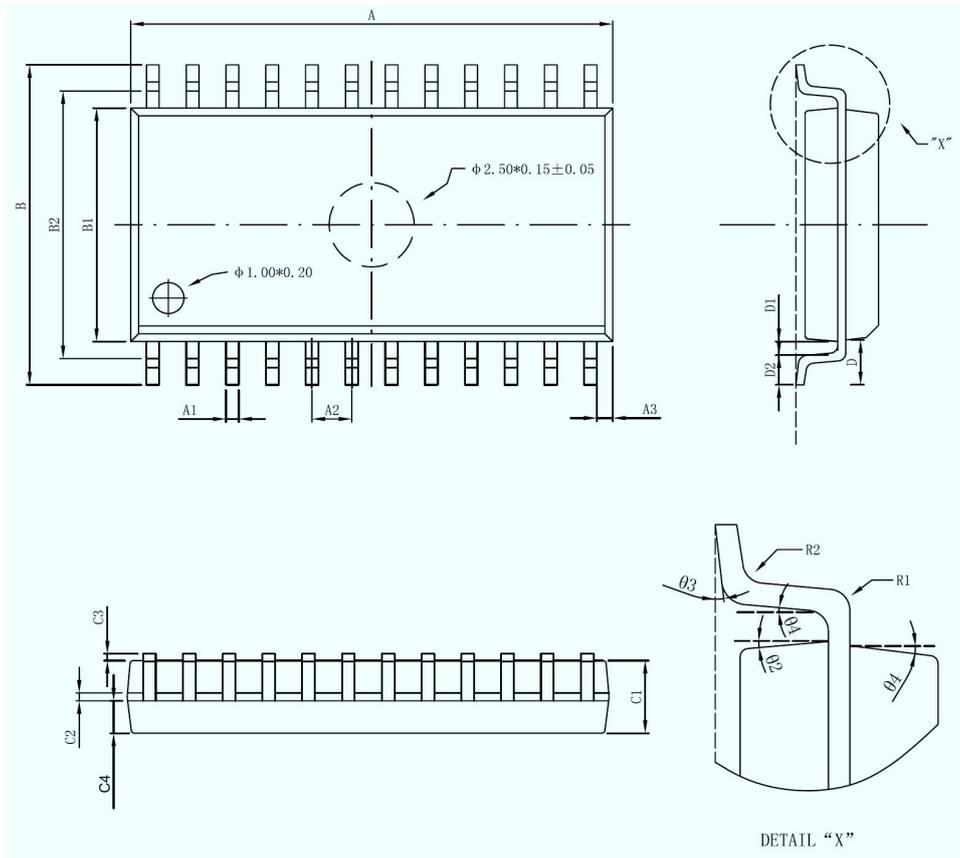


```
for(i=0;i<8;i++)
{
    KEY[j]=KEY[j]>>1;
    CLK=0;
    CLK=1;
    if(DIO)
        KEY[j]=KEY[j]&0x80;
}
STB=1;
}
/*****/
//主函数
/*****/
void main()
{
    while(1)
    {
        display();           //显示
        read_key();         //读按键数据
        {
            //按键处理,由用户自行编写
        }
    }
}
```



封装形式

SOP24



标注	尺寸		标注	尺寸	
	最小 (mm)	最大 (mm)		最小 (mm)	最大 (mm)
A	15.28	15.48	C4	0.86TYP	
A1	0406TYP		D	1.34TYP	
A2	1.27TYP		D1	0.33TYP	
A3	0.50TYP		D2	0.70	0.90
B	9.90	10.50	R1	0.25TYP	
B1	7.42	7.62	R2	0.25TYP	
B2	8.7TYP		#1	7°	
C1	2.13	2.23	#2	7°	
C2	0.204	0.33	#3	4°	
C3	0.10	0.23	#4	10°	