

HE2833

■ 产品简介

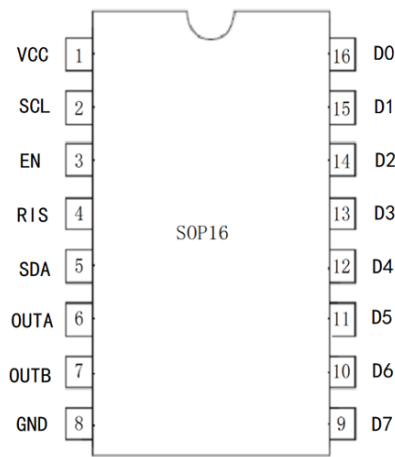
HE2833 是一款专用步进电机和 IR-Cut Removable 驱动集成电路。兼容 I²C 总线输入，八路并行输出控制驱动步进电机，两路互斥输出驱动 IR-Cut 电机。此串入并出的模式可为方案设计节约 MCU 的输出端口资源，缩减 PCB 布线面积，提高设计效率。HE2833 可驱动两路四相步进电机，或驱动八路继电器，也可用于 LED 大屏驱动等其它应用。

■ 产品特点

- 电源电压：3V~6V
 - 输出最大电流：400MA，Ta=25℃
 - 低功耗：典型值 0uA
- 兼容 I²C 总线通讯
 - 八路并行输出，两路 IR-Cut 互斥输出
 - 封装形式：SOP16

■ 封装形式和管脚功能定义

管脚定义	管脚功能描述	管脚编号	管脚定义	管脚编号	管脚功能描述
1	VCC	电源正	9	D7	并行输出 D7 端
2	SCL	数据时钟	10	D6	并行输出 D6 端
3	EN	数据输出锁存	11	D5	并行输出 D5 端
4	RIS	输出清零	12	D4	并行输出 D4 端
5	SDA	数据输入	13	D3	并行输出 D3 端
6	OUTA	IR-Cut 输出 A	14	D2	并行输出 D2 端
7	OUTB	IR-Cut 输出 B	15	D1	并行输出 D1 端
8	GND	电源地	16	D0	并行输出 D0 端



■ 极限参数

参数	符号	极限值	单位
输入电压	V _{CC}	7.0V	V
输出电压	V _{OUT}	0~V _{CC}	V
输出电流	I _{OUT}	400	mA
耗散功率	P _D	400	mW
输入电压	V _{IN}	0~V _{CC}	V
工作温度	T _A	-45~85	°C
存储温度	T _S	-65~150	°C

注：（1）极限参数是指无论在任何条件下都不能超过的极限值。如果超过此极限值，将有可能造成产品劣化等物理性损伤；同时在接近极限参数下，不能保证芯片可以正常工作。

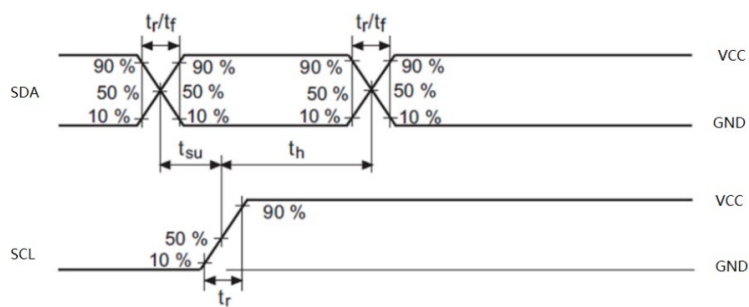
■ 电学特性

直流电学特性： T_A=25°C

符号	项目	测试条件		最小值	典型值	最大值	单位
V _{CC}	压电源电压	T _a =25° C		2.8	—	5.5	V
I _{VCC}	工作电流	V _{CC} = 3.3 V; V _{OUT} 输出全关闭		—	0	15	μ A
		V _{CC} = 5.0 V; V _{OUT} 输出全关闭		—	0	15	μ A
I _{OH}	D0~D7 输出漏电流	V _{OUT} = 7V	T _a =25° C	—	0	10	μ A
			T _a =85° C	—	0	50	μ A
V _{OL}	D0~D7 低电平有效 输出电压	I _{OUT} =350 mA, T _a =25° C		—	1.02	1.6	V
		I _{OUT} =200 mA, T _a =25° C		—	0.96	1.3	V
		I _{OUT} =100 mA , T _a =25° C		—	0.85	1.1	V
V _{OL}	OUTA/OUTB 输出低电压	V _{CC} = 5.0 V, I _{OUT} =80 mA		—	0.8	—	V
V _{OH}	OUTA/OUTB 输出高电压	V _{CC} = 5.0 V, I _{OUT} =80 mA		—	3.4	—	V
I _{IH}	输入电流	V _{CC} = 5.0 V; V _{IN} = 5.0 V		—	0	5	μ A
I _{IL}		V _{CC} = 5.0 V; V _{IN} = 0 V		—	0	5	μ A
V _{IH}	输入高电平	V _{CC} = 3.3 V		2.2	—	3.3	V
		V _{CC} = 5.0 V		2.5	—	5.0	V
V _{IL}	输入低电平	V _{CC} = 3.3 V		0	—	1.0	V
		V _{CC} = 5.0 V		0	—	1.5	V

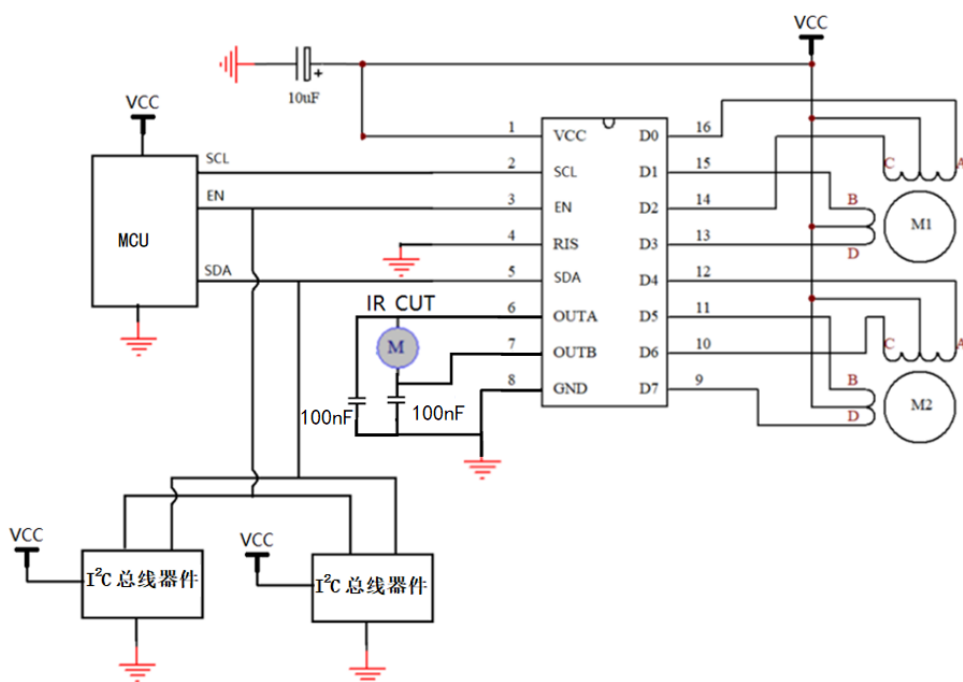
交流电学特性: **Ta=25℃** VCC=5.0V, $t_r=t_f \leq 20\text{ns}$ 见测试方法。

符号	项目	测试条件	最小值	典型值	最大值	单位
f_{Max}	SCL 最高工作频率	-	-	200	-	KHZ
t_{SU}	最小延时时间 SDA to SCL	波形图 1	-	5	-	us
t_H	最小保持时间 SDA to SCL	波形图 1	-	5	-	us



波形图 1: SDA to SCL

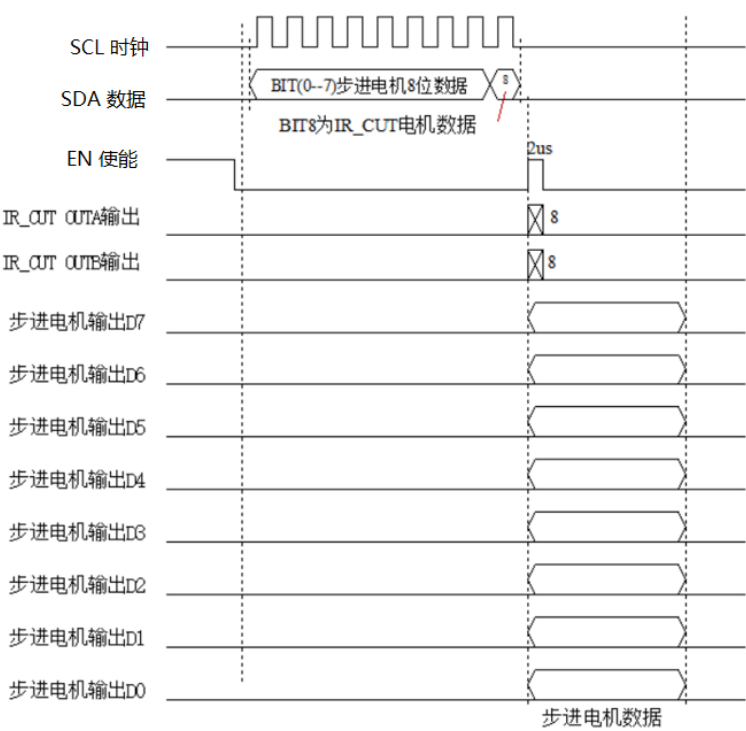
■ 典型应用线路



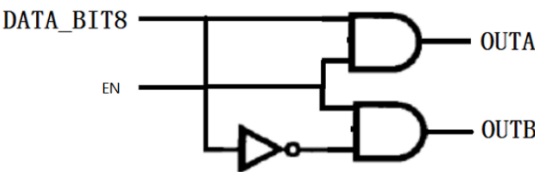
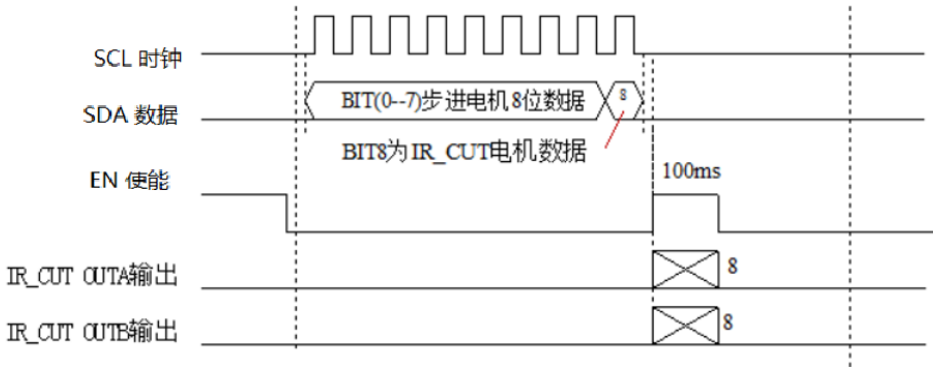
步进电机典型应用线路图

■ 时序图

1、步进电机时序



2、IR_CUT 电机时序



3、应用时序说明:

1、IR_CUT 的 OUTA 和 OUTB 输出电平由 DATA 数据的第 8 位数据决定, DATA_BIT8=1, 则 OUTA=1, OUTB=0; DATA_BIT8=0, 则 OUTA=0, OUTB=1。

2、OUTA 和 OUTB 输出时间由 EN 输入高电平脉宽决定。

3、在 IR_CUT 电机切换时, DATA_BIT7-0 步进电机 8 位数据建议设定为 0 (步进电机停止运转); EN 输入高电平脉宽一般设置为 50ms—200ms。

4、在步进电机工作时, DATA_BIT8 的数据应保持原状态不变, 同时 EN 输入高电平脉宽建议在 2-10us 内。

5、兼容 I²C 通讯, 在给其他 I²C 期间发送数据时, 需要让 EN 管脚保持低电平, 给 HE2833 发数据时, 每发完一帧数据 (bit0~bit8) 后需要发送一个 EN 时钟 (脉宽参考备注 3、4)。

■ 程序范例

1、步进电机驱动子程序

```
void wr_data(void)                //步进电机驱动子程序
{
    BYTE BIT_N=0;
    EN=0;
    SCL=0;
    .delay 20;                    //10US
    while(BIT_N<8)
    {
        SCL=0;
        .delay 20;
        SL DATA_F;              //左移一位，最高为 bit7 移入 CF
        if (CF==1) SDA=1;
        else SDA=0;
        .delay 20;
        SCL=1;
        .delay 20;
        BIT_N++;
    }
    SCL=0;
    .delay 20;
    if (DATA_icut==1) SDA=1;
    else SDA=0;
    .delay 20;
    SCL=1;
    .delay 20;
    SCL=0;
    .delay 20;
    SDA=0;
    EN=1;
    .delay 20;
    EN=0;
}
```

2、IR_CUT 驱动子程序

```
void wr_icut(void)               //IR_CUT 驱动子程序
{
    BYTE BIT_N=0;
    EN=0;
    SCL=0;
    .delay 20;                    //10US
    while(BIT_N<8)
```

```
{
    SCL=0;
    .delay 20;
    SL DATA_F;
    SDA=0;
    .delay 20;
    SCL=1;
    .delay 20;
    BIT_N++;
}
SCL=0;
.delay 20;
if(DATA_icut==1) SDA=1;
else SDA=0;
.delay 20;
SCL=1;
.delay 20;
SCL=0;
.delay 20;
SDA=0;
EN=1;
delay_n_ms=20;
delay_10ms();
EN=0;
}
```

3、步进电机向上转演示程序

```
void motor_up(void) //电机上转
{
    while (KEY_UP==0)
    {
        DATA_F=0X0C;
        wr_data ();
        .delay 20000;
        DATA_F=0X04;
        wr_data ();
        .delay 20000;

        DATA_F=0X06;
        wr_data ();
        .delay 20000;
        DATA_F=0X02;
        wr_data ();
        .delay 20000;
    }
}
```

```
DATA_F=0X03;
wr_data ();
.delay 20000;
DATA_F=0X01;
wr_data ();
.delay 20000;

DATA_F=0X09;
wr_data ();
.delay 20000;
DATA_F=0X08;
wr_data ();
.delay 20000;
}
DATA_F=0X00;          //0X00000000;
wr_data ();           //调用发送子程序
}
```

4、步进电机向下转演示程序

```
void motor_down(void)    //电机下转
{
    while (KEY_DOWN==0)
    {
        DATA_F=0X01;
        wr_data ();
        .delay 20000;
        DATA_F=0X03;
        wr_data ();
        .delay 20000;

        DATA_F=0X02;
        wr_data ();
        .delay 20000;
        DATA_F=0X06;
        wr_data ();
        .delay 20000;

        DATA_F=0X04;
        wr_data ();
        .delay 20000;
        DATA_F=0X0C;
        wr_data ();
        .delay 20000;

        DATA_F=0X08;
```



```
        wr_data ();  
        .delay 20000;  
        DATA_F=0X09;  
        wr_data ();  
        .delay 20000;  
    }  
    DATA_F=0X00;  
    wr_data ();  
}
```

5、步进电机向右转演示程序

```
Void motor_right(void)          //电机右转  
{  
    while (KEY_RIGHT==0)  
    {  
        DATA_F=0XC0;  
        wr_data ();  
        .delay 20000;  
        DATA_F=0X40;  
        wr_data ();  
        .delay 20000;  
  
        DATA_F=0X60;  
        wr_data ();  
        .delay 20000;  
        DATA_F=0X20;  
        wr_data ();  
        .delay 20000;  
  
        DATA_F=0X30;  
        wr_data ();  
        .delay 20000;  
        DATA_F=0X10;  
        wr_data ();  
        .delay 20000;  
  
        DATA_F=0X90;  
        wr_data ();  
        .delay 20000;  
        DATA_F=0X80;  
        wr_data ();  
        .delay 20000;  
    }  
    DATA_F=0X00;  
    wr_data ();
```

```
}
```

6、步进电机向左转演示程序

```
void motor_left(void)          //电机左转
{
    while (KEY_LEFT==0)
    {
        DATA_F=0X10;
        wr_data ();
        .delay 20000;
        DATA_F=0X30;
        wr_data ();
        .delay 20000;

        DATA_F=0X20;
        wr_data ();
        .delay 20000;
        DATA_F=0X60;
        wr_data ();
        .delay 20000;

        DATA_F=0X40;
        wr_data ();
        .delay 20000;
        DATA_F=0XC0;
        wr_data ();
        .delay 20000;

        DATA_F=0X80;
        wr_data ();
        .delay 20000;
        DATA_F=0X90;
        wr_data ();
        .delay 20000;
    }
    DATA_F=0X00;
    wr_data ();
}
```

7、IR_CUT 切换延时程序

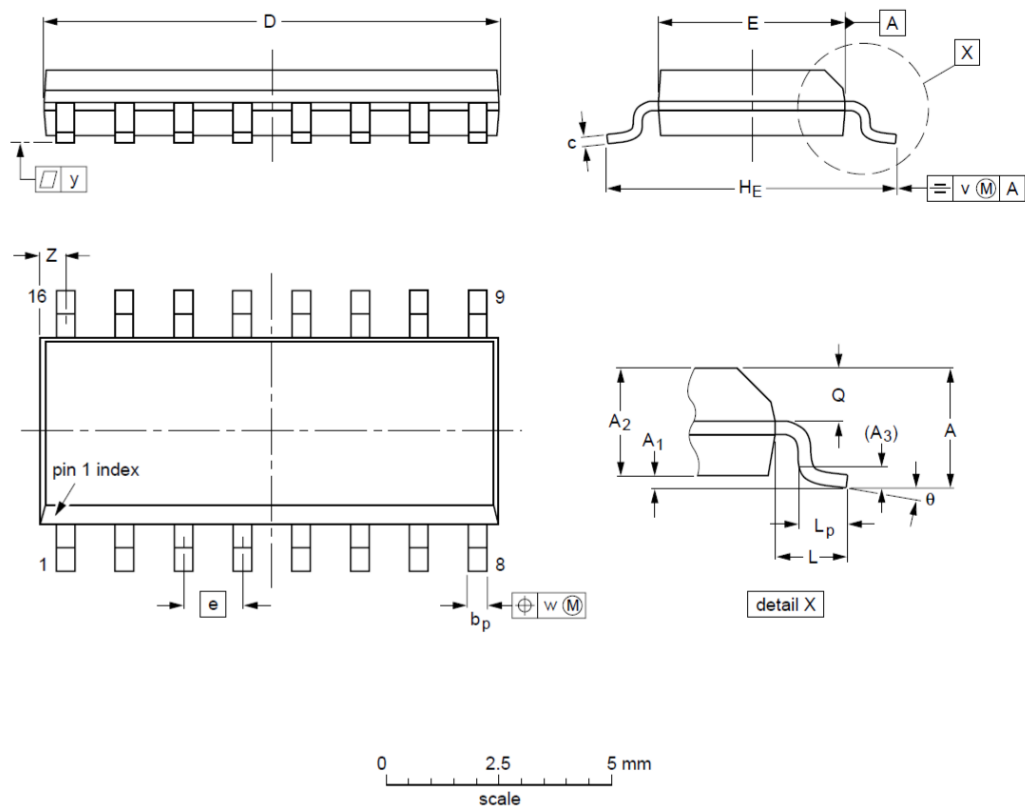
```
void motor_cut(void)          //I_CUT 驱动
{
    if(DATA_icut==0) DATA_icut=1;    //1 切上， 0 切下
    else DATA_icut=0;
```

```
    wr_icut();  
  
    delay_n_ms=20;           //延时 200MS  
    delay_10ms();  
    DATA_F=0X00;  
    wr_data ();  
}
```

■ 封装信息

单位：毫米 / 英寸

SOP16



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽¹⁾	e	H _E	L	L _p	Q	v	w	y	Z ⁽¹⁾	θ
mm	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	10.0 9.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8° 0°
inches	0.069	0.010 0.004	0.057 0.049	0.01	0.019 0.014	0.0100 0.0075	0.39 0.38	0.16 0.15	0.050	0.244 0.228	0.041	0.039 0.016	0.028 0.020	0.01	0.01	0.004	0.028 0.012	