

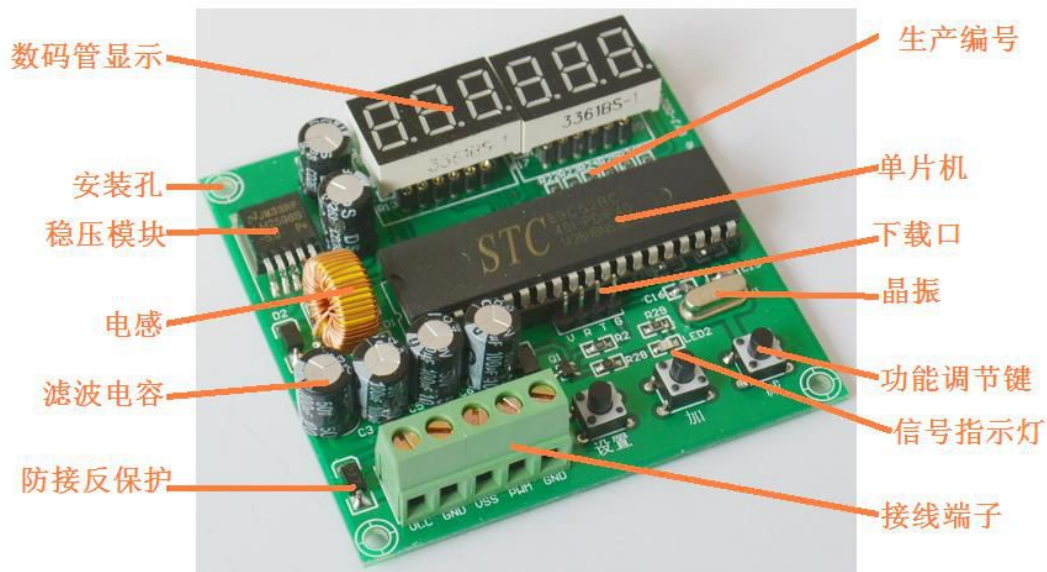
频率及占空比均可调带数码管模块使用手册

【简要说明】

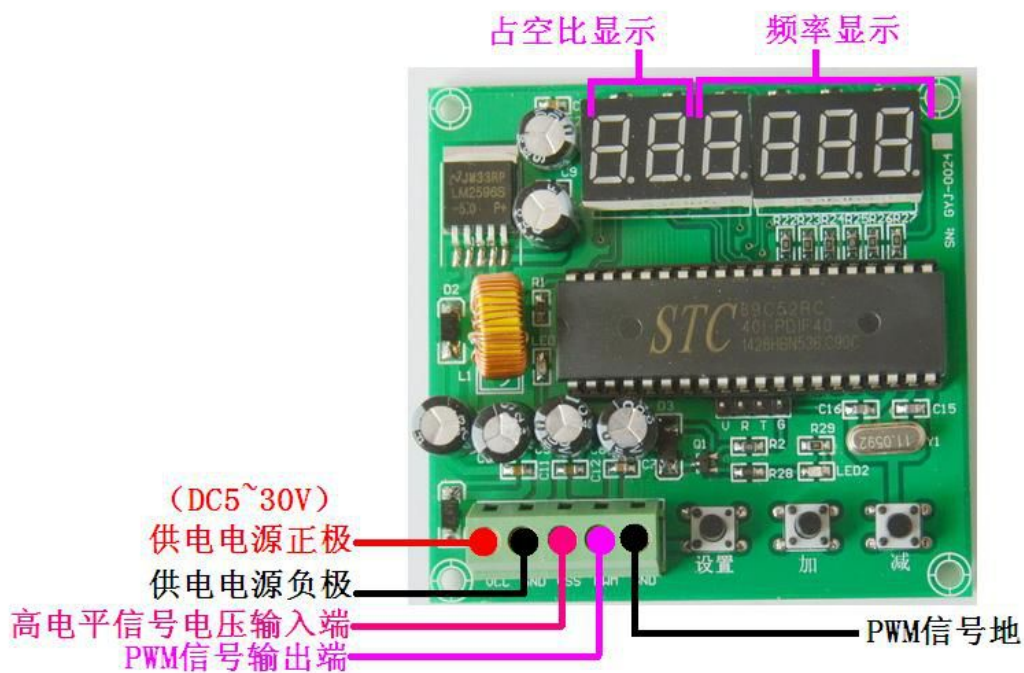
- 一、 尺寸：长 74mmX 宽 72mmX 高 18mm
- 二、 主要芯片：单片机，数码管、稳压器
- 三、 工作电压：6V 至 40V, 功耗小于 1W
- 四、 特点：
 - 1、具有稳压电路，输入电压广，具有电源指示灯。
 - 2、具有 6 位数码管显示，前两位显示占空比，后四位显示频率。
 - 3、具有按键调节功能，设置参数带记忆功能。
 - 4、输出占空比：0~100%可调。
 - 5、输出频率：小于 10KHZ 可调。
 - 6、输出 PWM 信号，电压 3.3V~30V, 电压值有输入 VSS 电压决定。输出电流最大 1.5A，输出最大信号功率 1.5W。
 - 7、单片机编程，客户可以自己更改，提供源代码
 - 8、有无信号输出，具有 LED 灯指示。
 - 9、端子采用螺旋压接端子
 - 10、工作温度-40 度至 +70 度
 - 11、工作湿度 40% ~ 80%RH
 - 12、具有电源防接反保护
 - 13、具有续流保护
 - 14、具有电磁抗干扰能力
 - 15、板子稳定工作可靠
 - 16、板子可安装在 DIN 导轨上面

使用说明：

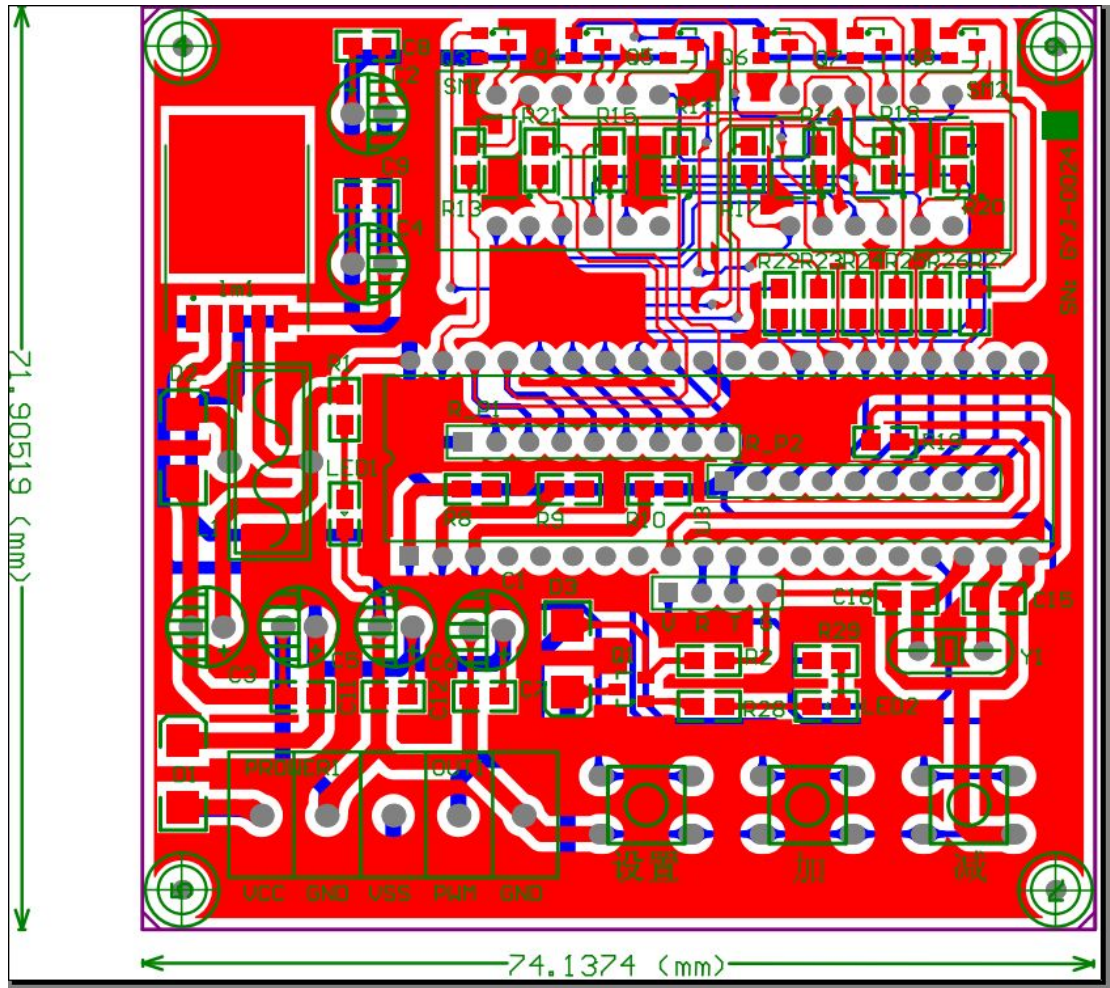
【标注说明】



【功能描述】



【原理图】

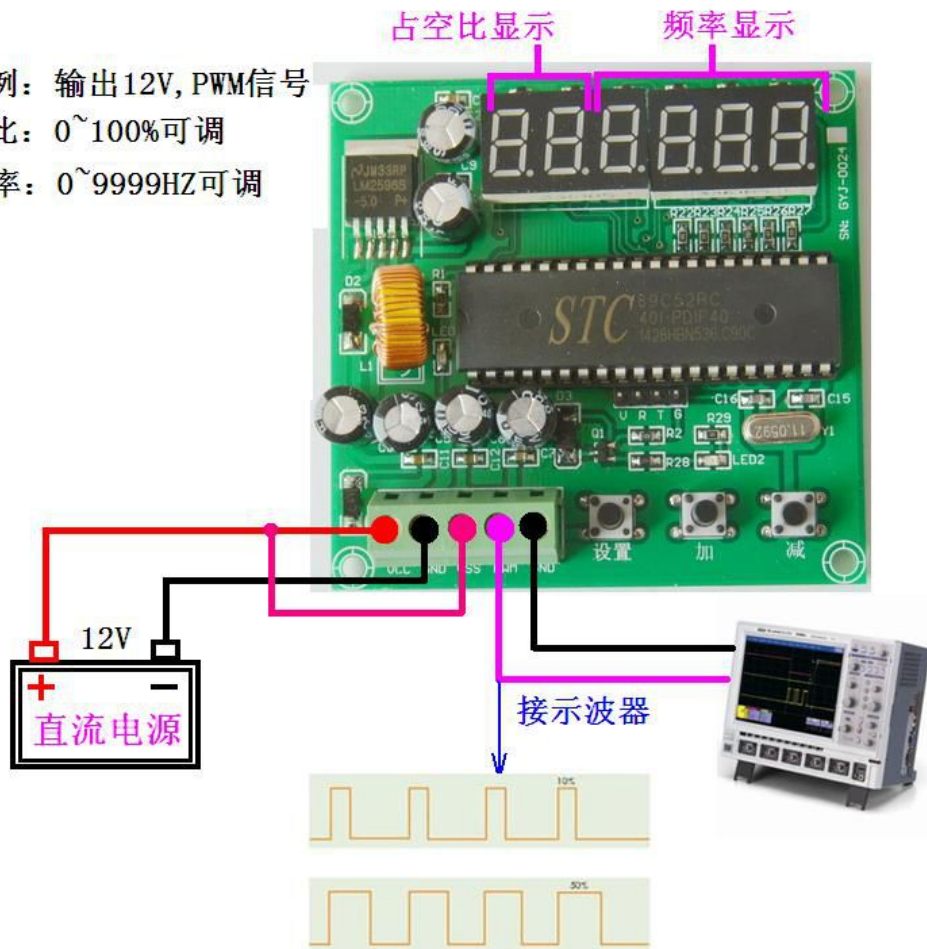


【元件清单】

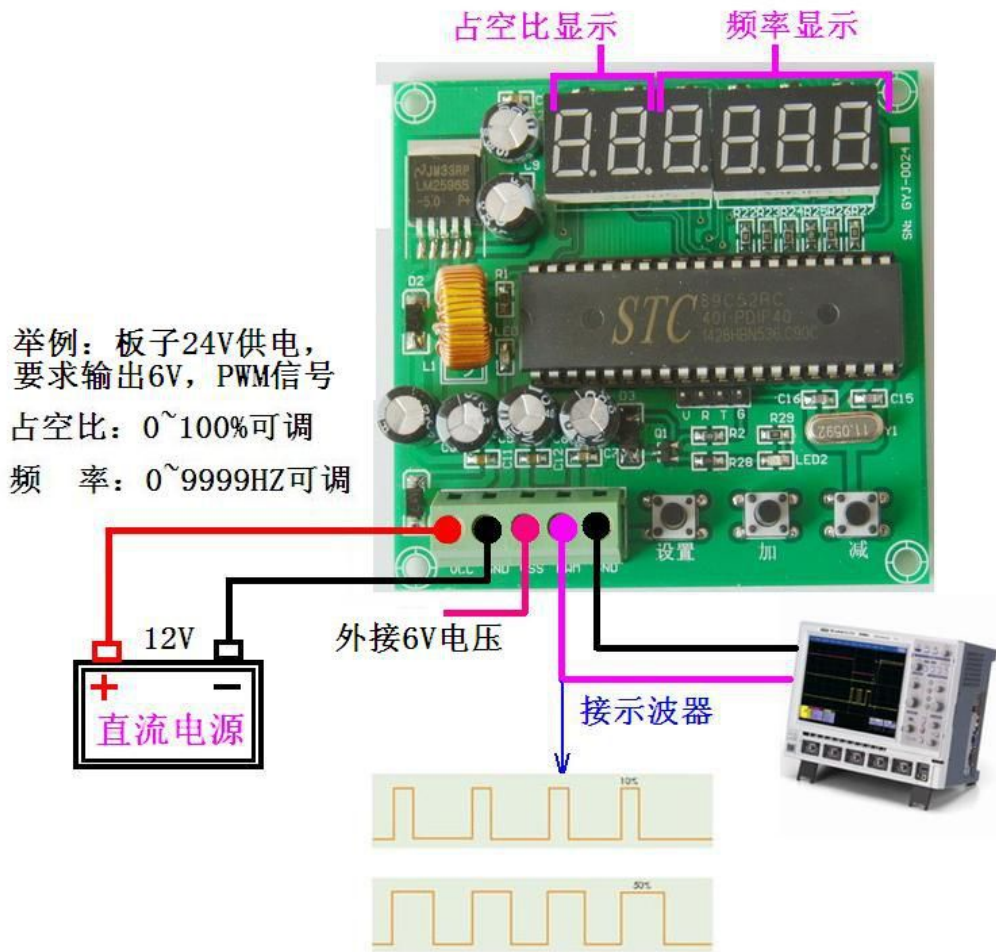
Comment	Description	Designator	Footprint	LibRef	Q...	Value
DJDR	电解电容	C1, C2, C3, C4, C5, C6	C-DJ-6	DJDR	6	100UF/50V
CPDR	瓷片电容	C7, C8, C9, C10, C11, C12, C15, C16	0805	CPDR	8	30, 104
ZL2JG	整流二极管	D1, D3	7227	ZL2JG	2	
WY2JG	稳压二极管	D2	7227	WY2JG	1	
DG	电感	L1	L_DG -fang	DG	1	100UH
LED	发光二极管	LED1	0805 - LED	LED	1	
	发光二极管	LED2	0805 - LED	LED	1	
LM2576T5.0	Simple Switcher 3A Ste	lm1	LM2596_TP	LM2576HVT-3.3	1	
PZ_3	排针—3	OUT1	JK128_3	PZ_3	1	
PZ_2	排针—2	PROWER1	JK128_2	PZ_2	1	
NPN	NPN Bipolar Transistor	Q1	SO-G3	NPN	1	
TPSJG_PNP	贴片三极管—PNP	Q3, Q4, Q5, Q6, Q7, Q8	SO-G3	TPSJG_PNP	6	
Res	电阻	R1, R8, R9, R10, R13, R14, R15, R16,	0805	Res	19	1K, 10K
Res2	电阻	R2, R29	0805	Res2	2	10K
	电阻	R22	0805	Res	1	1K
	8位排阻	R_P1, R_P2	HDR-9	Res PZ_8	2	10K
AJKG	按键开关	S1, S2, S3	an-6*6	AJKG	3	
三联数码管	四联数码管	SM1, SM2	SMG_1	SLSUMG_1	2	
STC89S52		U3	DIP-40	89c51	1	
XTAL	Crystal Oscillator	Y1	JZ_ZC	XTAL	1	

【应用举例】

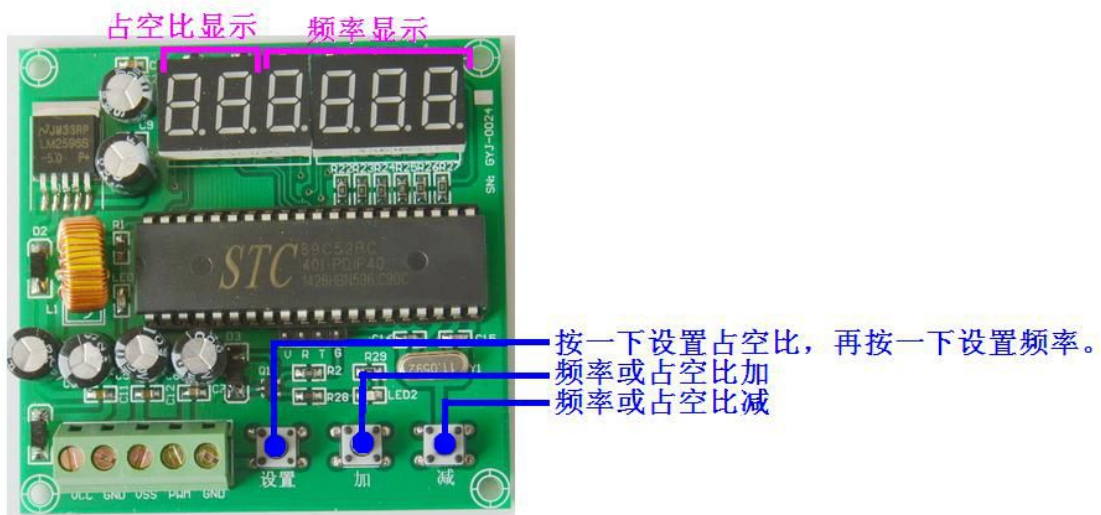
举例：输出12V, PWM信号
占空比：0~100%可调
频率：0~9999HZ可调



【应用举例 2】



【功能调节设置】



【测试参考程序】

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

汇诚科技

实现功能:频率及占空比均可调带数码管模块

使用芯片: STC12C5A60S2

晶振: 11.0592MHZ

编译环境: Keil

作者: zhangxinchun

淘宝店: 汇诚科技

【声明】此程序仅用于学习与参考, 引用请注明版权和作者信息!

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

```
/*
```

注意 单片机必须是 52rc 不能用 60s2

```
*/
```

```
#include"main.h"
```

```
#include"peizhi.h"
```

```
#include"smg.h"
```

```
unsigned char HighRH = 0; //高电平重载值的高字节
```

```
unsigned char HighRL = 0; //高电平重载值的低字节
```

```
unsigned char LowRH = 0; //低电平重载值的高字节
```

```
unsigned char LowRL = 0; //低电平重载值的低字节
```

```
/******按键定义*****/
```

```
bit d1 = 1;
```

```
bit d2 = 1;
```

```
bit d3 = 1;
```

```
uint8 ci;
```

```
uint8 ca = 0;
```

```
uint8 ca1 = 0;
```

```
uint16 cb = 0;
```

```
void key(); //按键函数声明
```

```
void ConfigPWM(unsigned int fr, unsigned char dc); //频率和占空比调节函数
```

```
void main()
```

```
{
```

```
    bit q1 = 1;
```

```
    bit q2 = 1;
```

```
    bit q3 = 1;
```

```
EA = 1;          //开总中断
peizhit1(1);    //配置 T0 定时 2ms
```

```
while(1)
{
```

```
    ConfigPWM(cb, ca);    //频率 100Hz, 占空比 10%
```

```
    if(d1 != q1)
```

```
    {
        q1 = d1;
        if(d1 == 0)
        {
```

```
            ca1++;
            if(ca1 >= 2)
            {
                ca1 = 0;
            }
        }
    }
```

```
    if(ca1 == 1)
```

```
    {
        if(d2 != q2)
        {
            q2 = d2;
            if(d2 == 0)
            {
```

```
                ca++;
                if(ca >= 99)
                {
                    ca = 99;
                }
            }
        }
    }
```

```
    if(d3 != q3)
```

```
    {
```

```
    q3 = d3;
    if(d3 == 0)
    {
        if(ca>0)
        {
            ca--;
        }
    }
}
```

```
if(ca1 == 0)
{
    if(d2 == 0)
    {
        cb++;
        if(cb >= 9999)
        {
            cb = 9999;
        }
    }
}
```

```
if(d3 == 0)
{
    if(cb>0)
    {
        cb--;
    }
}
```

```
    }
    xianshi1(cb);
    xianshi2(ca);
```

```
    }
}
```

```
void key()
```

```

{
    static uint8 saomiaozhi[] = {1,1,1,1};
    saomiaozhi[0] = (saomiaozhi[0]<<1) | in1;
    saomiaozhi[1] = (saomiaozhi[1]<<1) | in2;
    saomiaozhi[2] = (saomiaozhi[2]<<1) | in3;

    if(saomiaozhi[0] == 0x00)
    {
        d1 = 0;
    }
    if(saomiaozhi[0] == 0xff)
    {
        d1 = 1;
    }
    if(saomiaozhi[1] == 0x00)
    {
        d2 = 0;
    }
    if(saomiaozhi[1] == 0xff)
    {
        d2 = 1;
    }
    if(saomiaozhi[2] == 0x00)
    {
        d3 = 0;
    }
    if(saomiaozhi[2] == 0xff)
    {
        d3 = 1;
    }
}

/* 配置并启动 PWM， fr-频率， dc-占空比 */
void ConfigPWM(unsigned int fr, unsigned char dc)
{
    unsigned int  high, low;
    unsigned long tmp;

    tmp  = (11059200/12) / fr; //计算一个周期所需的计数值
    high = (tmp*dc) / 100;    //计算高电平所需的计数值
    low  = tmp - high;       //计算低电平所需的计数值
    high = 65536 - high + 12; //计算高电平的重载值并补偿中断延时
}

```

```

low = 65536 - low + 12; //计算低电平的重载值并补偿中断延时
HighRH = (unsigned char)(high>>8); //高电平重载值拆分为高低字节
HighRL = (unsigned char)high;
LowRH = (unsigned char)(low>>8); //低电平重载值拆分为高低字节
LowRL = (unsigned char)low;
TMOD &= 0xF0; //清零 T0 的控制位
TMOD |= 0x01; //配置 T0 为模式 1
TH0 = HighRH; //加载 T0 重载值
TL0 = HighRL;
ET0 = 1; //使能 T0 中断
TR0 = 1; //启动 T0
PWMOUT = 1; //输出高电平
}

```

/* T0 中断服务函数，产生 PWM 输出 */

```
void InterruptTimer0() interrupt 1
```

```

{
    if(PWMOUT == 1) //当前输出为高电平时，装载低电平值并输出低电平
    {
        TH0 = LowRH;
        TL0 = LowRL;
        PWMOUT = 0;
    }
    else //当前输出为低电平时，装载高电平值并输出高电平
    {
        TH0 = HighRH;
        TL0 = HighRL;
        PWMOUT = 1;
    }
}

```

/* T0 中断服务函数，完成数码管、按键扫描与秒表计数 */

```
void t1() interrupt 3
```

```

{

    TH1 = T1RH; //重新加载重载值
    TL1 = T1RL;
    ci++;
    ssmg(); //数码管扫描

    if(ci>=2)
    {
        ci = 0;
    }
}

```

```
    key();  
  }  
}
```

【图片展示】



