

南京立超电子科技有限公司

N79A8211 之 PWM 应用笔记

2009 年 3 月 15

中国南京市和燕路 251 号金港大厦 A 幢 2406 室

Room 2406, Tower A, Jingang mansion, 251 Heyan Road, Nanjing 210028, P. R. China

Tel: 0086-25-83306839/83310926 Fax: 0086-25-83737785

[Http://www.dycmcu.com](http://www.dycmcu.com)

版权申明

立超电子科技股份有限公司保留对此文件修改之权利且不另行通知。立超电子科技有限公司所提供之信息相信为正确且可靠之信息，但并不保证本文件中绝无错误。请于向立超电子科技股份有限公司提出订单前，自行确定所使用之相关技术文件及规格为最新之版本。若因贵公司使用本公司之文件或产品，而涉及第三人之专利或著作权等智能财产权之应用及配合时，则应由贵公司负责取得同意及授权，本公司仅单纯贩售产品，上述关于同意及授权，非属本公司应为保证之责任。又未经立超电子科技股份有限公司之正式书面许可，本公司之所有产品不得使用于医疗器材，维持生命系统及飞航等相关设备。

修改记录

版本	日期	作者	修订内容	对应页码
V1.0	3/15/2009	宋娴	初始版本	

目 录

1、文件概要	1
2、PWM综述.....	2
2.1、PWM技术的基本原理.....	2
2.2、N79A8211 的PWM.....	2
2.3、PWM的钳制功能	3
3、PWM的相关寄存器.....	6
3.1、PWMP计数器高位寄存器.....	6
3.2、PWMP 计数器低位寄存器.....	6
3.3、PWM 0 高位寄存器.....	6
3.4、PWMO 低位寄存器.....	6
3.5、PWM 1 高位寄存器.....	7
3.6、PWM1 低位寄存器.....	7
3.7、PWM控制寄存器 1.....	7
3.8、PWM控制寄存器 2.....	8
3.9、PWM 控制寄存器 3.....	9
3.10、中断允许寄存器 1.....	9
4、PWM的使用.....	10
5、案例及说明	13
5.1、电路图	13
5.2、流程图	14
5.3、程序	15
编后说明	19
参考资料	20

1、文件概要

该内容主要针对N79A8231的PWM的使用作相应的应用说明。

下文将用图文分别说明各项功能。

注：以下文件中Fosc是指晶振、RC或外部输入的时钟，F_{sys}为系统时钟。

2、PWM 综述

PWM的全称是Pulse Width Modulation(脉冲宽度调制)，简称脉宽调整。

2.1、PWM技术的基本原理

PWM技术的原理，简单来说就是通过调整一个周期固定的方波的占空比，来调节输出的平均电压、电流或者功率等被控量。

2.2、N79A8211 的PWM

N79A8211系列有2个PWM输出通道，分别是PWM0(P0.1)、PWM1(P1.6)。

N79A8211系列支持10-位向下PWM计数器，它的时钟源为CPU时钟。PWM计数器时钟= FOSC分频，通过设定 FP[1:0] PWMCON3[3:2] 。

PWM的频率为： $f_{PWM} = F_{CPU} / (PWMP+1)$ 。PWMP 的10- 位寄存器由PWMPH.1、PWMPH.0 和PWMP.L.7~PWMP.L.0组成。

每一个PWM输出脉冲的宽度由比较寄存器PWMOL~PWM1L和PWMOH~PWM1H决定的，当PWM比较寄存器的值大于10-位计数器寄存器时，PWM输出低。

PWM 输出高脉冲宽度是： $t_{HI} = (PWMP - PWM_{n+1})$

$$\text{Period} = (\text{pwmp} + 1) * \text{ioclock period} * 1/\text{prescaler}$$

$$\text{Duty} = \text{duty} * \text{Period}$$

PWMP寄存器写入的数据在会自动的装载到PWMRUN，CF标志为10-位向下计数器向下溢出，CF 标志在下一个周期自动清除。当计数器向下溢出后计数器的内容将自动从计数器寄存器重装。当PWMP寄存器被装载到计数器寄存器，装载位将在下个周期被自动清除。

如果第一PWM 输出周期由PWMP设定，CLR PWM把10-位计数器设为000H，CF也会被清除；然后设置PWMRUN (PCON1.7) 和装载位Load (PCON1.6) 来启动PWM。如果要输出可变的PWM脉宽，在写PWMn寄存器后，必须把装载位置Load (PCON1.6) “1”，向下溢出时把PWMn寄存器的值装载到比较寄存器。

助记符: PWMCON1

地址: DCh

Bit	7	6	5	4	3	2	1	0
名称	PWMRUN	Load	PWMP	CLR PWM	I	PWM1I	PWM0I	

7	PWMRUN	0: PWM没有运行. 1: PWM计数器运行.
6	Load	0: PWMP寄存器的值比较器不会被装载到计数器和比较器寄存器中; 1: 在计数器向下溢出后, PWMP寄存器的值将会装入计数器寄存器, 下一个周期硬件清'0'。

2.3、PWM的钳制功能

N79A8211的PWM功能一旦运行，无法关闭，只能通过钳制来控制PWM。当钳制发生时，钳制所有的PWM口，即所有PWM共用一个时钟。

N79A8211系列支持软件或外部引脚(P0.2)钳制功能。钳制控制通过PWMCON2寄存器设定。软件钳制和外部引脚钳制设定请参考钳制条件表。

助记符：PWMCON2

地址：DFh

Bit	7	6	5	4	3	2	1	0
名称	BKCH	BKPS	BPEN	BKEN			PWM1B	PWMOB

位	名称	功能
7	BKCH	见下表.
6	BKPS	0: P0.2=0, 钳制PWM 1: P0.2=1, 钳制PWM
5	BPEN	见下表
4	BKEN	0: 钳制禁止. 1: 钳制允许.

钳制条件表:

BPEN	BKCH	功能
0	0	钳制打开, 软件钳制由BKEN控制
0	1	打开: 当PWM不运行(PWMRUN=0)时, 由PWMnB设定PWM的输出条件 关闭: 当PWM为运行(PWMRUN=1)时
1	0	钳制打开, 当钳制脚指定后, 没有PWM输出, PWMRUN位被清除并把BKF标志只置'0'。
1	1	PWM 进行运行. PWM 输出不没有按照 PWMnB, 进行正常的分频输出.

1、外部钳制引脚(P0.2)钳制PWM

外部钳制引脚(P0.2)钳制PWM可以产生钳制中断，可以在中断服务程序或查询钳制标志(BKF)确定外部引脚是否钳制PWM。如果P0.2拉低，BKPS = 0，BKF(PWMCON3.0)将会被置“1”和PWNRUN将会被清除，PWM停止运行；在钳制脚释放后，PWM输出条件由PWMnB设定。钳制确定时，PWM输出由PWMnB设定。BKPS(PWMCON3.6)可以设置P0.2不同状态钳制PWM。

钳制引脚确定后，运行位将会自动清除，BKF(PWMCON3.0)标志将会自动置'1'。

2、软件钳制PWM

软件钳制，把BKEN置“1”将允许钳制功能。

由BPEN和BKCH位设定，(BPEN, BKCH) = (0, 0)，钳制确定；

(BPEN, BKCH) = (0, 1)，PWM输出与PWMRUN位一致；

当PWM不运行时也就是说PWMRUN=0，PWM输出由PWMnB设定；

当PWM正在运行时，PWMRUN = 1并保持PWM输出。

在用户程序里，可以通过检测PWMCON1.7或允许PWM的钳制中断来判断是不是由钳制引脚引起的钳制发生。

如果检测程序发现钳制的时间不够长，在引起钳制的钳制条件消失时，无论PWM处于什么状态，PWM输出在本周期内不会立即取消钳制；这样防止钳制解除后，PWM会进入混乱状态。

注：为了平缓的解除外部钳制引脚的钳制，PWM继续运行。

步骤参见下图：

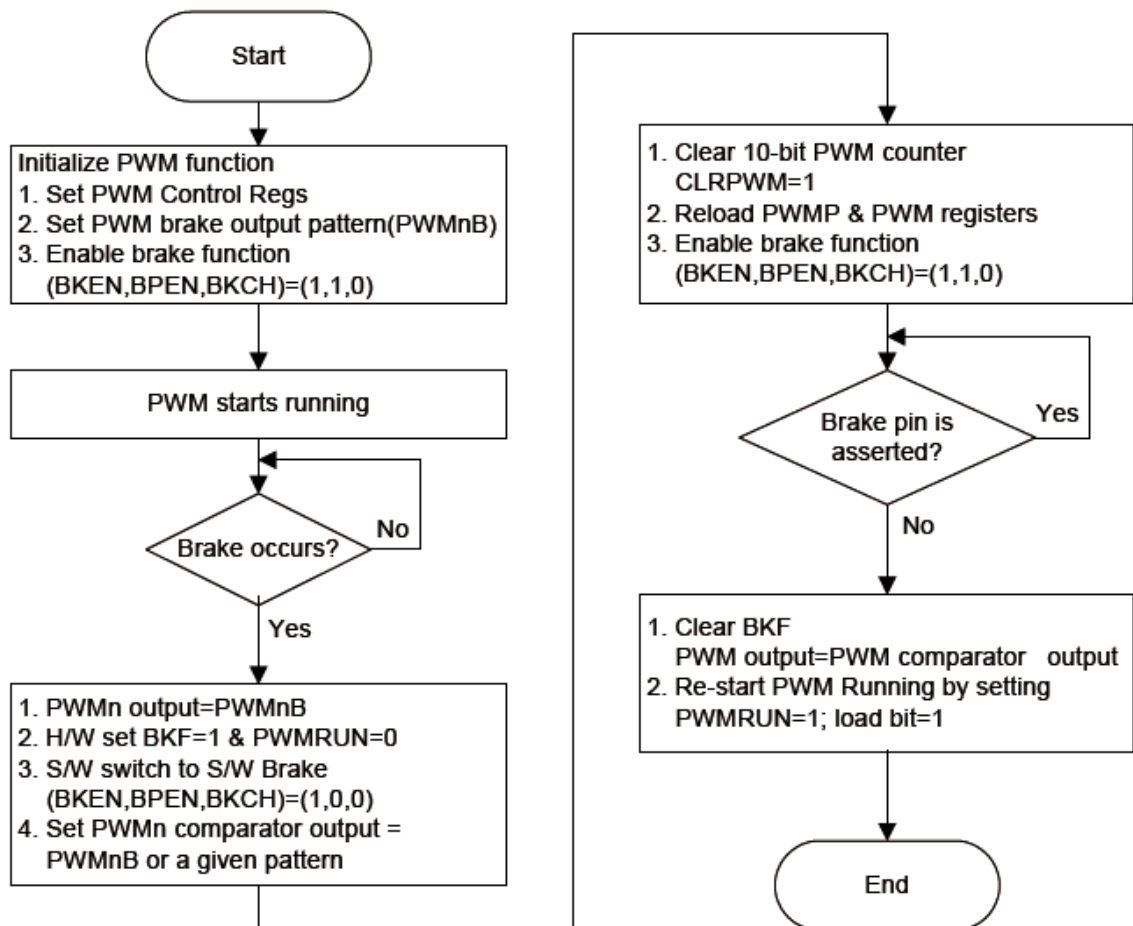


图 2-1 PWM钳制功能

由图 2-2 可知：当钳制发生时，并且PWM不运行时，PWM与IO口与操作。即：当PWM1B输出为低时，IO口输出一一直为低；当PWM1B输出为高时，IO口输出与做普通IO时IO口输出值一致。故当PWM开始运行后，想要实现普通IO功能，则钳制PWM，并设置PWMnB输出为高。PWMnB由寄存器PWMCON2[0-1]控制。

助记符：PWMCON2

地址：DFh

Bit	7	6	5	4	3	2	1	0
名称	BKCH	BKPS	BPEN	BKEN			PWM1B	PWMOB

1	PWM1B	0: 当钳制有效 PWM1输出为低 1: 当钳制有效 PWM1输出为高
0	PWMOB	0: 当钳制有效 PWMO输出为低 1: 当钳制有效 PWMO输出为高

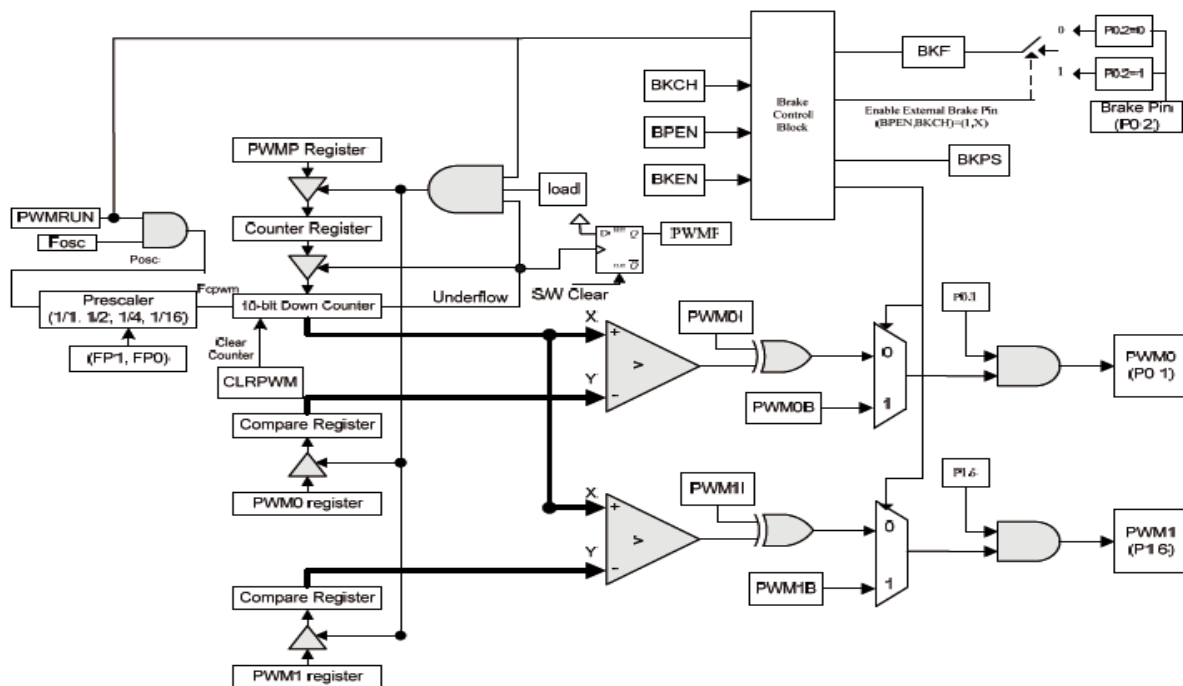


图 2-2 PWM框图

3、PWM 的相关寄存器

3.1、PWMP计数器高位寄存器

助记符: PWMPH

地址: D1h

Bit	7	6	5	4	3	2	1	0
名称	-	-	-	-	-	-	PWMP. 9	PWMP. 8

位	名称	功能
7~2	-	保留
1~0	PWMP. [9:8]	PWM计数器寄存器的9~8位.

3.2、PWMP 计数器低位寄存器

助记符: PWMPL

地址: D9h

Bit	7	6	5	4	3	2	1	0
名称	PWMP. 7	PWMP. 6	PWMP. 5	PWP. 4	PWMP. 3	PWMP. 2	PWMP. 1	PWMP. 1

位	名称	功能
7~0	PWMP. [7:0]	计数器的低位寄存器. __

3.3、PWM 0 高位寄存器

助记符: PWM0H

地址: D2h

Bit	7	6	5	4	3	2	1	0
名称	-	-	-	-	-	-	PWM0. 9	PWM0. 8

位	名称	功能
7~2	-	保留
1~0	PWM0. [9:8]	0寄存器9~8位.

3.4、PWM0 低位寄存器

助记符: PWM0L

地址: DAh

Bit	7	6	5	4	3	2	1	0
名称	PWM0. 7	PWM0. 6	PWM0. 5	PWM0. 4	PWM0. 3	PWM0. 2	PWM0. 1	PWM0. 1

位	名称	功能
7~0	PWM0. [7:0]	PWM0低位寄存器.

3.5、PWM 1 高位寄存器

助记符: PWM1H

地址: D3h

Bit	7	6	5	4	3	2	1	0
名称	-	-	-	-	-	-	PWM1.9	PWM1.8

位	名称	功能
7~2	-	保留
1~0	PWM1. [9:8]	PWM1寄存器9~8位.

3.6、PWM1 低位寄存器

助记符: PWM1L

地址: DBh

Bit	7	6	5	4	3	2	1	0
名称	PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0

位	名称	功能
7~0	PWM1. [7:0]	PWM1低位寄存器.

3.7、PWM控制寄存器 1

助记符: PWMCON1

地址: DCh

Bit	7	6	5	4	3	2	1	0
名称	PWMRUN	Load	PWMF	CLRPWM	I	PWM1I	PWMOI	

位	名称	功能
7	PWMRUN	0: PWM没有运行. 1: PWM计数器运行.
6	Load	0: PWMP寄存器的值比较器不会被装载到计数器和比较器寄存器中; 1: 在计数器向下溢出后, PWMP寄存器的值将会装入计数器寄存器, 下一个周期硬件清'0'。
5	PWMF	PWM 向下溢出标准位 0: 没有溢出. 1: PWM 10位计数器向下溢出 (产生PWM 中断请求, 当PWM中断使能).
4	CLRPWM	清PWM 计数器 1: 清 PWM 10位计数器为000H. 置位后由硬件自动清除.

3		
2		
1	PWM1 I	0: PWM1 正相输出. 1: PWM1 反相输出.
0	PWM0 I	0: PWM0 正相输出. 1: PWM0 反相输出.

3.8、PWM控制寄存器 2

助记符: PWMCON2

地址: DFh

Bit	7	6	5	4	3	2	1	0
名称	BKCH	BKPS	BPEN	BKEN			PWM1B	PWM0B

位	名称	功能
7	BKCH	见下表.
6	BKPS	0: P0.2=0, 钳制PWM 1: P0.2=1, 钳制PWM
5	BPEN	见下表
4	BKEN	0: 钳制禁止. 1: 钳制允许.
3		
2		
1	PWM1B	0: 当钳制有效 PWM1输出为低 1: 当钳制有效 PWM1输出为高
0	PWM0B	0: 当钳制有效 PWM0输出为低 1: 当钳制有效 PWM0输出为高

钳制条件表:

BPEN	BKCH	功能
0	0	钳制打开, 软件钳制由BKEN控制
0	1	打开: 当PWM不运行(PWMRUN=0)时, 由PWMnB设定PWM的输出条件 关闭: 当PWM为运行(PWMRUN=1)时
1	0	钳制打开, 当钳制脚指定后, 没有PWM输出, PWMRUN位被清除并把BKF标志只置'0'。
1	1	PWM 进行运行. PWM 输出不按照 PWMnB, 而进行正常的分频输出.

3.9、PWM 控制寄存器 3

助记符: PWMCON3

地址: D7h

Bit	7	6	5	4	3	2	1	0
名称	-	-	-	-	FP1	FP0	-	BKF

位	名称	功能										
7-4	-	保留.										
3-2	FP[1:0]	PWM频率选择位. 当 PWMRUN=1, 使能 P W M <table border="1" data-bbox="625 698 853 965"> <tr> <td>FP[1:0]</td> <td>Fpwm</td> </tr> <tr> <td>00</td> <td>FOSC</td> </tr> <tr> <td>01</td> <td>FOSC/2</td> </tr> <tr> <td>10</td> <td>FOSC/4</td> </tr> <tr> <td>11</td> <td>FOSC/16</td> </tr> </table>	FP[1:0]	Fpwm	00	FOSC	01	FOSC/2	10	FOSC/4	11	FOSC/16
FP[1:0]	Fpwm											
00	FOSC											
01	FOSC/2											
10	FOSC/4											
11	FOSC/16											
1	-	保留.										
0	BKF	0: PWM禁止钳制. 1: 使能PWM钳制. 该位由软件清零.										

3.10、中断允许寄存器 1

助记符: EIE

地址: E8h

Bit	7	6	5	4	3	2	1	0
名称	EED	EPWMUF	EPWM	EWDI	-	-	-	-

位	名称	功能
7	EED	0: 禁止边沿触发中断. 1: 使能边沿触发中断.
6	EPWMUF	0: 禁止PWM下溢中断. 1: 使能 PWM 下溢中断.
5	EPWM	0: 禁止PWM下溢中断(外部钳制). 1: 使能 PWM 下溢中断(外部钳制).
4	EWDI	0: 禁止看门狗中断. 1: 使能看门狗中断.
3-0	-	保留.

4、PWM 的使用

PWM的使用可以按照以下步骤：（以下均以P1.6/PWM1为例）

(1)、I/O口设置:

如：
 $P1M1 \&= 0xb0;$
 $P1M2 |= 0x40;$ //推挽输出，0 1
 $P1 |= 0x02;$ //初始化为1;

注：PWM功能口必须为“1”，否则无法开启PWM功能。

(2)、设置PWM频率:

如：
 $PWMCON3 = 0x00;$ // PWM频率选择：1/1
 PWM频率选择位PWMCON3.[3:2]. 当 PWMRUN=1, 使能PWM

FP[1:0]	Fpwm
00	FOSC
01	FOSC/2
10	FOSC/4
11	FOSC/16

(3)、设置PWM周期:

如：
 $PWMPH = 0x03;$
 $PWMPL = 0xff;$
 $Period = 1/Fosc * (PWMP+1) * 1/prescaler$
 $= 1/20MHZ * 4 * (1023+1) * 1$
 $= 248us$

(4)、设置占空比:

如：
 $PWM1H = 0x01;$
 $PWM1L = 0xff;$
 $Duty = (PWMP - PWMn+1) / (PWMP+1) * Period$
 $= (1023-511+1) / 1024 * 248us$
 $= 124us$

(5)、开启PWM:

$PWMCON1 = 0xe0;$

PWM控制寄存器1: PWMCON1

位	名称	功能
7	PWMRUN	0: PWM没有运行. 1: PWM计数器运行.
6	Load	0: PWMP寄存器的值比较器不会被装载到计数器和比较器寄存器中; 1: 在计数器向下溢出后, PWMP寄存器的值将会装入计数器寄存器, 下一个周期硬件清'0'。
5	PWMF	PWM 向下溢出标准位 0: 没有溢出. 1: PWM 10位计数器向下溢出 (产生PWM 中断请求, 当PWM中断使能).
4	CLRPWM	清PWM 计数器 1: 清 PWM 10位计数器为000H. 置位后由硬件自动清除.

(6)、关闭即钳制PWM:

PWM一旦开启, 无法关闭, 如果要控制PWM则采用钳制功能。

钳制状态: PWM输出与PWMRUN位一致;

当PWM不运行时也就是说PWMRUN=0, PWM输出由PWMnB设定;

当PWM正在运行时, PWMRUN = 1并保持PWM输出。

如: PWMCON1 = 0x00; // 停止运行

PWMCON2=Bin(00011111); //钳制PWM

此刻, 普通IO输出与PWMnB是与操作, 即要实现普通IO功能则必须设置: PWMnB当钳制有效 PWM1输出为高。

PWMCON2|=0x03;

一旦钳制取消, PWM继续运行。

助记符: PWMCON2

地址: DFh

Bit	7	6	5	4	3	2	1	0
名称	BKCH	BKPS	BPEN	BKEN			PWM1B	PWM0B

位	名称	功能
7	BKCH	见下表.
6	BKPS	0: P0.2=0, 钳制PWM 1: P0.2=1, 钳制PWM
5	BPEN	见下表
4	BKEN	0: 钳制禁止. 1: 钳制允许.

1	PWM1B	0: 当钳制有效 PWM1输出为低 1: 当钳制有效 PWM1输出为高
0	PWM0B	0: 当钳制有效 PWM0输出为低 1: 当钳制有效 PWM0输出为高

(6)、如果要输出可变的PWM脉宽，在写PWMn寄存器后，必须把装载位置“1”：

如：
PWM1H=0x00;
PWM1L=0x00;
PWMCON1|=0x40;

备注：

1. 如果把比较寄存器设置成000H，PWMn输出高；如果比较寄存器置成3FFH， PWMn输出低
2. 在 ICP 模式下，PWM将停止。 ICP结束后，PWM引脚将按照 SFR的值输出。
- 3、启动 PWM 之前，I/O 口必须初始化为“1”，否则 PWM 无法运行。
- 4、关闭 PWM 时，必须钳制 PWM 并关闭 PWM，当钳制取消，PWM 继续运行。

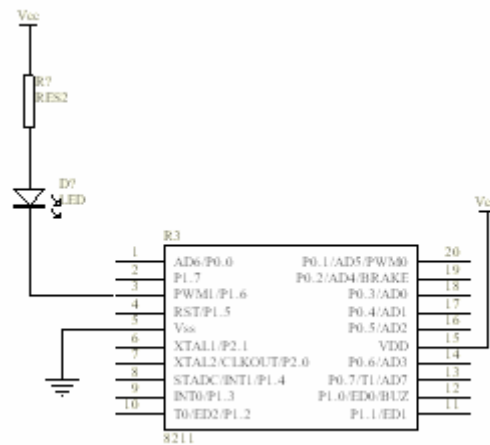
5、案例及说明

下面是以N79A8211 为例编写的PWM程序：

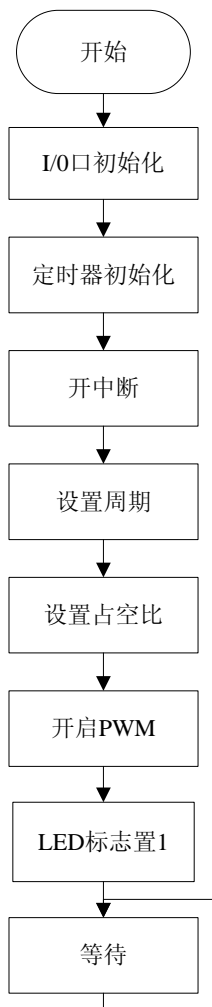
功能：（1）、通过调节占空比，输出可变脉宽。

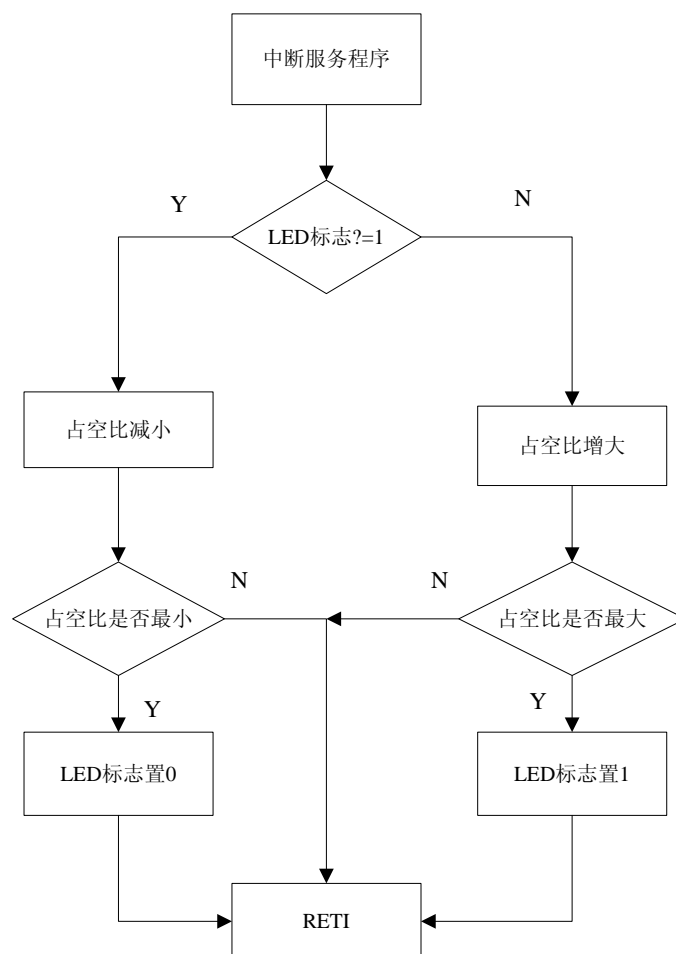
（2）、LED灯从亮减灭，从不亮到渐亮，依次循环。

5.1、电路图



5.2、流程图





5.3、程序

```
#include "config.h"
```

```
uint8 led_flag;
```

```
/**/
```

```
** 函数名称: Start_PWM
```

```
** 函数描述: 开启PWM
```

```
** 输入参数: 无
```

```
** 输出参数: 无
```

```
/**/
```

```
void Start_PWM
```

```
{
```

```
    P1M1&=0xb0;
```

```
    P1M2|=0x40;    //推挽输出, 0 1
```

```
    P1|=0x02;
```

```
PWMPH=0x03;
PWMPL=0xff;          // 频率设定
PWM1H=0x03;         // 占空比设定
PWM1L=0xff;
PWMCON3=0x0c;       // PWM频率选择: 1/16
PWMCON1=0xe0;
led_flag=1;         //1: 灯由亮到灭, 0: 灯由灭到亮
}

/*****
** 函数名称: PWM_Sub
** 函数描述: 占空比调节
** 输入参数: 无
** 输出参数: 无
*****/
void PWM_Sub()
{
    uint16 val;
    val=(PWM1H<<8)+PWM1L;
    if(led_flag==1)
    {
        if(--val==3)
        {
            led_flag=0;
        }
    }
    else
    {
        if(++val==0x3ff)
        {
            led_flag=1;
        }
    }
    PWM1H=(val>>8)&Bin(00000011);
    PWM1L=val&Bin(11111111);
}
```

```
PWMCON1|=0x40;
}
/*****
** 函数名称: Stop_PWM
** 函数描述: 钳制PWM
** 输入参数: 无
** 输出参数: 无
*****/
void Start_PWM
{
    PWMCON1 = 0x00;    // 停止运行
    PWMCON2=Bin(00011111); //钳制PWM
}
/*****
** 函数名称: Init_Timer
** 函数描述: 初始化TIMER
** 输入参数: 无
** 输出参数: 无
*****/
void Init_Timer(void)
{
    TR0=0;                //关闭TIMER0
    TF0=0;
    TMOD=Bin(00010001);   //Timer0和Timer1都做为16位手动重装定时器
    CKCON=0x08;          //时钟选择为1 / 4系统时钟
    TLO=T0_2MS_CNT&0x00ff;
    TH0=(T0_2MS_CNT>>8)&0x00ff;
    TR0=1;                //启动TIMER0
    ET0=1;
}
/*****
** 函数名称: Timer0_ISP
** 函数描述: TIMER0中断服务程序          5ms中断一次
*****/
```

** 输入参数: 无

** 输出参数: 无

*****/

```
void Timer0_ISP(void) interrupt 1
```

```
{
```

```
    EA=0;
```

```
    TF0=0;                //清T0中断标志
```

```
    TLO=T0_2MS_CNT&0x00ff;
```

```
    TH0=(T0_2MS_CNT>>8)&0x00ff;
```

```
    //在此加入您要处理的代码
```

```
    PWM_Sub();
```

```
    EA=1;
```

```
}
```

*****/

** 函数名称: main

** 函数描述: 程序主函数

** 输入参数: 无

** 输出参数: 无

*****/

```
void main(void)
```

```
{
```

```
    Init_Timer();
```

```
    EA=1;
```

```
    Start_PWM();
```

```
    while(1)
```

```
    {
```

```
    }
```

```
}
```

编后说明

NA79A8211 系列单片机PWM使用指南,是立超电子为帮助用户迅速掌握NA79A8211单片机而编写的,限于水平,难免有错误和不妥之处,恳请广大读者批评指正。请将您的建议和批评发至E-Mail:xian.song@dycmcu.com,我们深表感谢!

继本指南之后,我们已经或即将编写以下使用指南、应用设计和范例程序等,欢迎及时访问我们的网页:www.dycmcu.com,或通过电话、E-Mail 索取更多的应用指南及内部资料等。

使用指南:

NA79A8211 系列单片机A/D 转换器使用指南

NA79A8211 系列单片机PWM 使用指南

NA79A8211 系列单片机内置WDT 使用指南

NA79A8211 系列单片机中断使用指南

NA79A8211 系列单片机的低功耗设计方法

.....

应用设计:

SH69PXX 系列单片机与E2PROM 接口及程序设计

SH69PXX 系列单片机I2C 总线模拟程序包

.....

其它:

NA79A8211 系列单片机特殊功能寄存器定义库文件

NA79A8211 系列单片机特殊功能寄存器定义库文件

NA79A8211 系列单片机特殊功能寄存器定义库文件

本应用指南欢迎各相关电子网站转载,为了尊重我们付出的劳动,请您注明出处来自站点:www.dycmcu.com

立超电子 技术支持部

2009 年 3 月 15 日 星期日

编写单位

南京立超电子科技有限公司

参考资料

- 1、N79A8211 规格书 单片机规格书。