

STC15F2K60S2 芯片 PWM 的应用

1 . 目的

脉宽调制（ PWM ， Pulse Width Modulation ）是一种使用程序来控制波形占空比、周期、相位波形的技术，在三相电机驱动、 D/A 转换等场合有广泛的应用。

STC15 系列单片机的 PCA 模块可以通过设定各自的寄存器 PCA_PWMn（ n=0,1,2.下同）中的位 EBSn_1/PCA_PWMn.7 及 EBSn_0/PCA_PWMn.6，使其工作于 8 位 PWM 或 7 位 PWM 或 6 位 PWM 模式。

2 . 与 CCP/PWM/PCA 应用有关的特殊功能寄存器

符号	描述	位地址及其符号									复位值
		地址	B7	B6	B5	B4	B3	B2	B1	B0	
CCON	PCA Control Register	D8H	CF	CR	-	-	-	CCF2	CCF1	CCF0	00xx xx00
CMOD	PCA Mode Register	D9H	CIDL	-	-	-	CPS2	CPS1	CPS0	ECF	0xxx 0000
CCAPM0	PCA Module 0 Mode Register	DAH	-	ECOM0	CAPP0	CAPN0	MAY0	TOG0	PWM0	ECCF0	x000 000
CCAPM1	PCA Module 1 Mode Register	DBH	-	ECOM1	CAPP1	CAPN1	MAY1	TOG1	PWM1	ECCF1	x000 0000
CCAPM2	PCA Module 2 Mode Register	DCH	-	ECOM2	CAPP2	CAPN2	MAY2	TOG2	PWM2	ECCF2	x000 0000
CL	PCA Base Timer Low	E9H									0000 0000
CH	PCA Base Timer High	F9H									0000 0000
CCAP0L	PCA Module-0 Capture Register Low	EAH									0000 0000
CCAP0H	PCA Module-0 Capture Register High	FAH									0000 0000
CCAP1L	PCA Module-1 Capture Register Low	EBH									0000 0000
CCAP1H	PCA Module-1 Capture Register High	FBH									0000 0000
CCAP2L	PCA Module-2 Capture Register Low	ECH									0000 0000
CCAP2H	PCA Module-2 Capture Register High	FCH									0000 0000
PCA_PWM0	PCA PWM Mode Auxiliary Register 0	F2H	EBS0_1	EBS0_0	-	-	-	-	EPC0H	EPC0L	00xx xx00
PCA_PWM1	PCA PWM Mode Auxiliary Register 1	F3H	EBS1_1	EBS1_0	-	-	-	-	EPC1H	EPC1L	00xx xx00
PCA_PWM2	PCA PWM Mode Auxiliary Register 2	F4H	EBS2_1	EBS2_0	-	-	-	-	EPC2H	EPC2L	00xx xx00
AUXR1_P_SW1	Auxiliary Register 1	A2H	S1_S1	S1_S0	CCP_S1	CCP_S0	SP1_S1	SP1_S0	-	DPS	0100 0000

2.1.PCA 工作模式寄存器 CMOD

CMOD

PCA 工作模式寄存器的格式如下：
CMOD：PCA 工作模式寄存器

SFR name	Address	bit	B7	B6	B5	B4	B3	B2	B1	B0
CCON	D9H	name	CIDL	-	-	-	CPS2	CPS1	CPS0	ECF

CIDL：空闲模式下是否停止 PCA 计数的控制位。
当 CIDL = 0 时，空闲模式下 PCA 计数器继续工作；
当 CIDL = 1 时，空闲模式下 PCA 计数器停止工作；
CPS2、CPS1、CPS0：PCA 计数脉冲源选择控制位。
PCA 计数脉冲选择如下表所示。

CPS2	CPS1	CPS0	选择 PCA/PWM 时钟源输入
0	0	0	0，系统时钟，SYSclk/12
0	0	1	1，系统时钟，SYSclk/2
0	1	0	2，定时器 0 的溢出脉冲。由于定时器 0 可以工作在 T1 模式，所以可以达到记一个时钟就溢出，从而达到最高频率 CPU 工作时钟 SYSclk，通过改变定时器 0 的溢出率，可以实现可调频率的 PWM 输出
0	1	1	3，ECI/P1.2(或 P4.1)脚输入的外部时钟（最大速率 =SYSclk/2）
1	0	0	4，系统时钟，SYSclk
1	0	1	5，系统时钟 /4，SYSclk/4
1	1	0	6，系统时钟 /6，SYSclk/6
1	1	1	7，系统时钟 /8，SYSclk/8

例如，CPS2/CPS1/CPS0 = 1/0/0 时，CCP/PCA/PWM 的时钟源是 SYSclk，不用定时器 0，PWM 的频率为 SYSclk/256。
如果要用系统时钟 /3 来作为 PCA 的时钟源，应选择 T0 的溢出作为 CCP/PCA/PWM 的时钟源，此时应让 T0 工作在 1T 模式，计数 3 个脉冲即产生溢出。用 T0 的溢出可对系统时钟进行 1~65536 级分频（T0 工作在 16 为重装载模式）。

ECF：PCA 计数溢出中断使能位。
当 ECF = 0 时，禁止寄存器 CCON 中 CF 位的中断；
当 ECF = 1 时，允许寄存器 CCON 中 CF 位的中断。

CCON

2.2 . PCA 控制寄存器 CCON
PCA 控制寄存器的格式如下：
CCON：PCA 控制寄存器

SFR name	Address	bit	B7	B6	B5	B4	B3	B2	B1	B0
CCON	D8H	name	CF	CR	-	-	-	-	CCF1	CCF0

CF：PCA 计数器阵列溢出标志位。当 PCA 计数器溢出时，CF 由硬件置位。如果 CMOD 寄存器的 ECF 位置位，则 CF 标志可用来产生中断。CF 位可通过硬件或软件置位，但只能通过软件清零。
CR：PCA 计数器阵列运行控制位。该位通过软件置位，用来启动计数器阵列计数。该位通过软件清零，用来关闭 PCA 计数器。
CCF2：PCA 模块 2 中断标志。当出现匹配或捕捉时该位由硬件置位。该位必须通过软件清零。
CCF1：PCA 模块 1 中断标志。当出现匹配或捕捉时该位由硬件置位。该位必须通过软件清零。
CCF0：PCA 模块 0 中断标志。当出现匹配或捕捉时该位由硬件置位。该位必须通过软件清零。

CCAPM0

2.3.PCA 比较/捕获寄存器 CCAPM0、CCAPM1 和 CCAPM2
PCA 模块 0 的比较/捕获寄存器的格式如下：
CCAPM0：PCA 模块 0 的比较/捕获寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
CCAPM0	DAH	name	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0

B7：保留为将来之用。
ECOM0：允许比较器功能控制位。

CCAPM1

CCAPM2

当 ECOM0 = 1 时，允许比较器功能。

CAPP0：正捕获控制位。

当 CAPP0 = 1 时，允许上升沿捕获。

CAPN0：负捕获控制位。

当 CAPN0 = 1 时，允许下降沿捕获。

MAT0：匹配控制位。

当 MAT0 = 1 时，PCA 计数值与模块的比较 / 捕获寄存器的值的匹配将置位 CCON 寄存器的中断标志位 CCF0。

TOG0：翻转控制位。

当 TOG0 = 1 时，工作在 PCA 高速脉冲输出模式，PCA 计数器的值与模块的比较 / 捕获寄存器的值的匹配将使 CCP0 脚翻转。

PWM0：脉宽调制模式。

当 PWM0 = 1 时，允许 CCP0 脚用作脉宽调节输出。

ECCF0：使能 CCF0 中断。使能寄存器 CCON 的比较 / 捕获标志 CCF0,用来产生中断。

PCA 模块 1 的比较 / 捕获寄存器的格式如下：

CCAPM1：PCA 模块 1 的比较 / 捕获寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
CCAPM1	DBH	name	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1

B7：保留为将来之用。

ECOM1：允许比较器功能控制位。

当 ECOM1 = 1 时，允许比较器功能。

CAPP1：正捕获控制位。

当 CAPP1 = 1 时，允许上升沿捕获。

CAPN1：负捕获控制位。

当 CAPN1 = 1 时，允许下降沿捕获。

MAT1：匹配控制位。

当 MAT1 = 1 时，PCA 计数值与模块的比较 / 捕获寄存器的值的匹配将置位 CCON 寄存器的中断标志位 CCF1。

TOG1：翻转控制位。

当 TOG1 = 1 时，工作在 PCA 高速脉冲输出模式，PCA 计数器的值与模块的比较 / 捕获寄存器的值的匹配将使 CCP1 脚翻转。

PWM1：脉宽调制模式。

当 PWM1 = 1 时，允许 CCP1 脚用作脉宽调节输出。

ECCF1：使能 CCF1 中断。使能寄存器 CCON 的比较 / 捕获标志 CCF1,用来产生中断。

PCA 模块 2 的比较 / 捕获寄存器的格式如下：

CCAPM2：PCA 模块 2 的比较 / 捕获寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
CCAPM2	DCH	name	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2

B7：保留为将来之用。

ECOM2：允许比较器功能控制位。

当 ECOM2 = 1 时，允许比较器功能。

CAPP2：正捕获控制位。

当 CAPP2 = 1 时，允许上升沿捕获。

CAPN2：负捕获控制位。

当 CAPN2 = 1 时，允许下降沿捕获。

MAT2：匹配控制位。

当 MAT2 = 1 时，PCA 计数值与模块的比较 / 捕获寄存器的值的匹配将置位 CCON 寄存器的中断标志位 CCF2。

TOG2：翻转控制位。

当 TOG2 = 1 时，工作在 PCA 高速脉冲输出模式，PCA 计数器的值与模块的比较 / 捕获寄存器的值的匹配将使 CCP2 脚翻转。

PWM2：脉宽调制模式。

当 PWM2 = 1 时，允许 CCP2 脚用作脉宽调节输出。

ECCF2：使能 CCF2 中断。使能寄存器 CCON 的比较 / 捕获标志 CCF2,用来产生中断。

2.4.PCA 的 16 位计时器 — 低 8 位 CL 和高 8 位 CH

CL 和 CH 地址分别为 E9H 和 F9H，复位值均为 00H，用于保存 PCA 的装载值。

2.5.PCA 捕捉 /比较寄存器 — CCAPnL（低位字节）和 CCAPnH（高位字节）

当 PCA 模块用于捕获或比较时，它们用于保存各个模块的 16 位捕捉计数值；当 PCA 模块用于 PWM 模式时，它们用来控制输出的占空比。其中，n = 0、1、2，分别对应模块 0、模式 1 和模块 2。复位值均为 00H。它们对应的地址分别为：

CCAP0L — EAH、CCAP0H — FAH：模块 0 的捕捉 /比较寄存器。

CCAP1L — EBH、CCAP1H — FBH：模块 1 的捕捉 /比较寄存器。

CCAP2L — ECH、CCAP2H — FCH：模块 2 的捕捉 /比较寄存器。

2.6.PCA 模块 PWM 寄存器 PCA_PWM0、PCA_PWM1 和 PCA_PWM2

PCA 模块 0 的 PWM 寄存器的格式如下：

PCA_PWM0：PCA 模块 0 的 PWM 寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
PCA_PWM0	F2H	name	EBS0_1	EBS0_0	-	-	-	-	EPC0H	EPC0L

EBS0_1,EBS0_0：当 PCA 模块 0 工作于 PWM 模式时的功能选择位。

- 0，0：PCA 模块 0 工作于 8 位 PWM 功能；
- 0，1：PCA 模块 0 工作于 7 位 PWM 功能；
- 1，0：PCA 模块 0 工作于 6 位 PWM 功能；
- 1，1：无效，PCA 模块 0 工作于 8 位 PWM 模式。

EPC0H：在 PWM 模式下，与 CCAP0H 组成 9 位数。

EPC0L：在 PWM 模式下，与 CCAP0L 组成 9 位数。

PCA 模块 1 的 PWM 寄存器的格式如下：

PCA_PWM1：PCA 模块 1 的 PWM 寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
PCA_PWM1	F3H	name	EBS2_1	EBS2_0	-	-	-	-	EPC1H	EPC1L

EBS1_1,EBS1_0：当 PCA 模块 1 工作于 PWM 模式时的功能选择位。

- 0，0：PCA 模块 1 工作于 8 位 PWM 功能；
- 0，1：PCA 模块 1 工作于 7 位 PWM 功能；
- 1，0：PCA 模块 1 工作于 6 位 PWM 功能；
- 1，1：无效，PCA 模块 1 工作于 8 位 PWM 模式。

EPC1H：在 PWM 模式下，与 CCAP1H 组成 9 位数。

EPC1L：在 PWM 模式下，与 CCAP1L 组成 9 位数。

PCA 模块 2 的 PWM 寄存器的格式如下：

PCA_PWM2：PCA 模块 2 的 PWM 寄存器

SFR name	Address	Bit	B7	B6	B5	B4	B3	B2	B1	B0
PCA_PWM2	F4H	name	EBS2_1	EBS2_0	-	-	-	-	EPC2H	EPC2L

EBS2_1,EBS2_0：当 PCA 模块 2 工作于 PWM 模式时的功能选择位。

- 0，0：PCA 模块 2 工作于 8 位 PWM 功能；
- 0，1：PCA 模块 2 工作于 7 位 PWM 功能；
- 1，0：PCA 模块 2 工作于 6 位 PWM 功能；
- 1，1：无效，PCA 模块 2 工作于 8 位 PWM 模式。

EPC2H：在 PWM 模式下，与 CCAP2H 组成 9 位数。

EPC2L：在 PWM 模式下，与 CCAP2L 组成 9 位数。

PCA 模块的工作模式设定表如下表所列：

ECOMn	CAP Pn	CAPNn	MATn	TOGn	PWMn	ECCFn	模块功能
0	0	0	0	0	0	0	无此操作
1	0	0	0	0	1	0	8 位 PWM，无中断
1	1	0	0	0	1	1	8 位 PWM 输出，由低变高产生中断
1	0	1	0	0	1	1	8 位 PWM 输出，由高变低产生中断
1	1	1	0	0	1	1	8 位 PWM 输出，由高变低或由低到高
X	1	0	0	0	0	X	16 位捕获模式，由 CEXn/PCAn 的上升沿触发

X	0	1	0	0	0	X	16 位捕获模式，由 CEXn/PCAn 的下降沿触发
X	1	1	0	0	0	X	16 位捕获模式，由 CEXn/PCAn 的跳变触发
1	0	0	1	0	0	X	16 位软件定时器
1	0	0	1	1	0	X	16 位高速输出

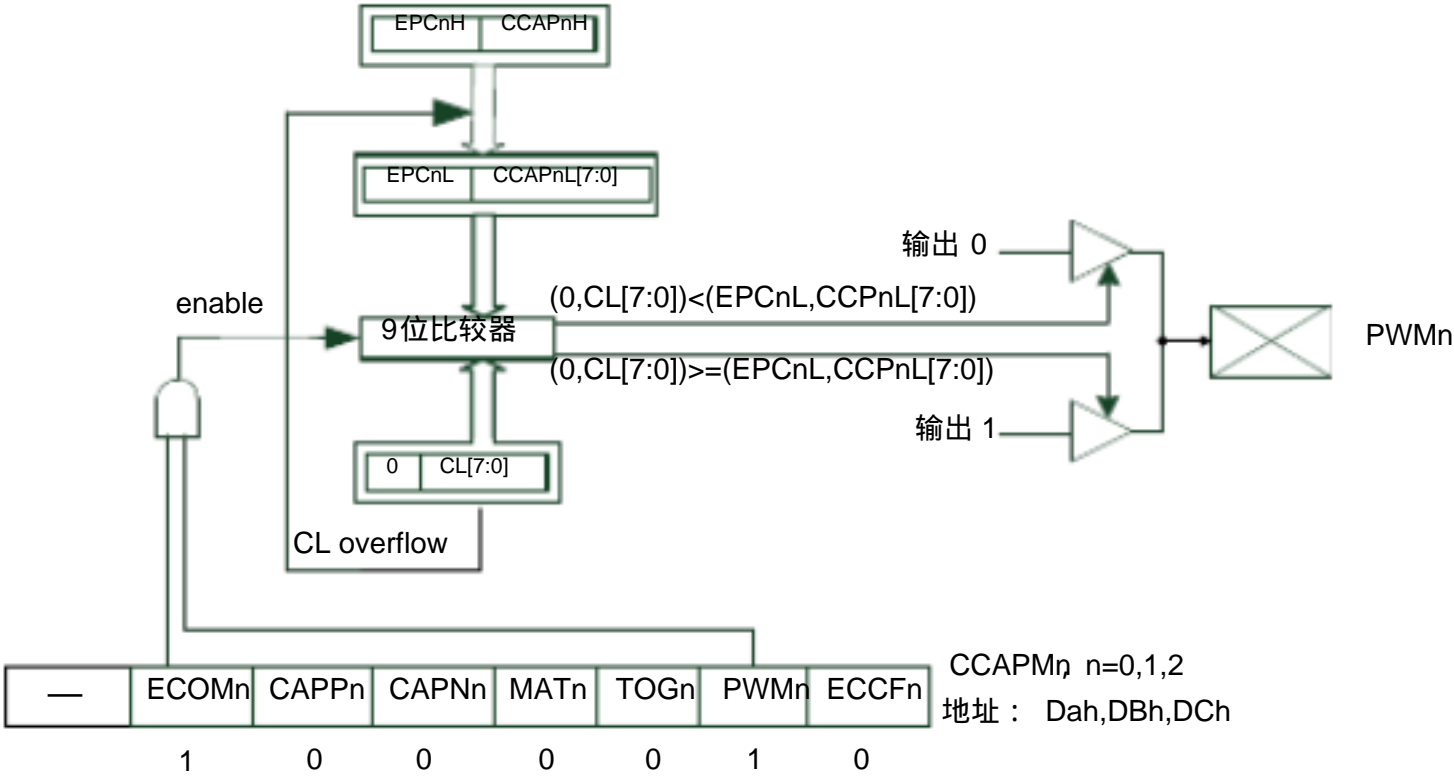
PCA 模块工作模式设定（ CCAPMn 寄存器， n = 0、 1、 2 ）

3 . 原理图

3.1 8 位脉宽调节模式（ PWM ）

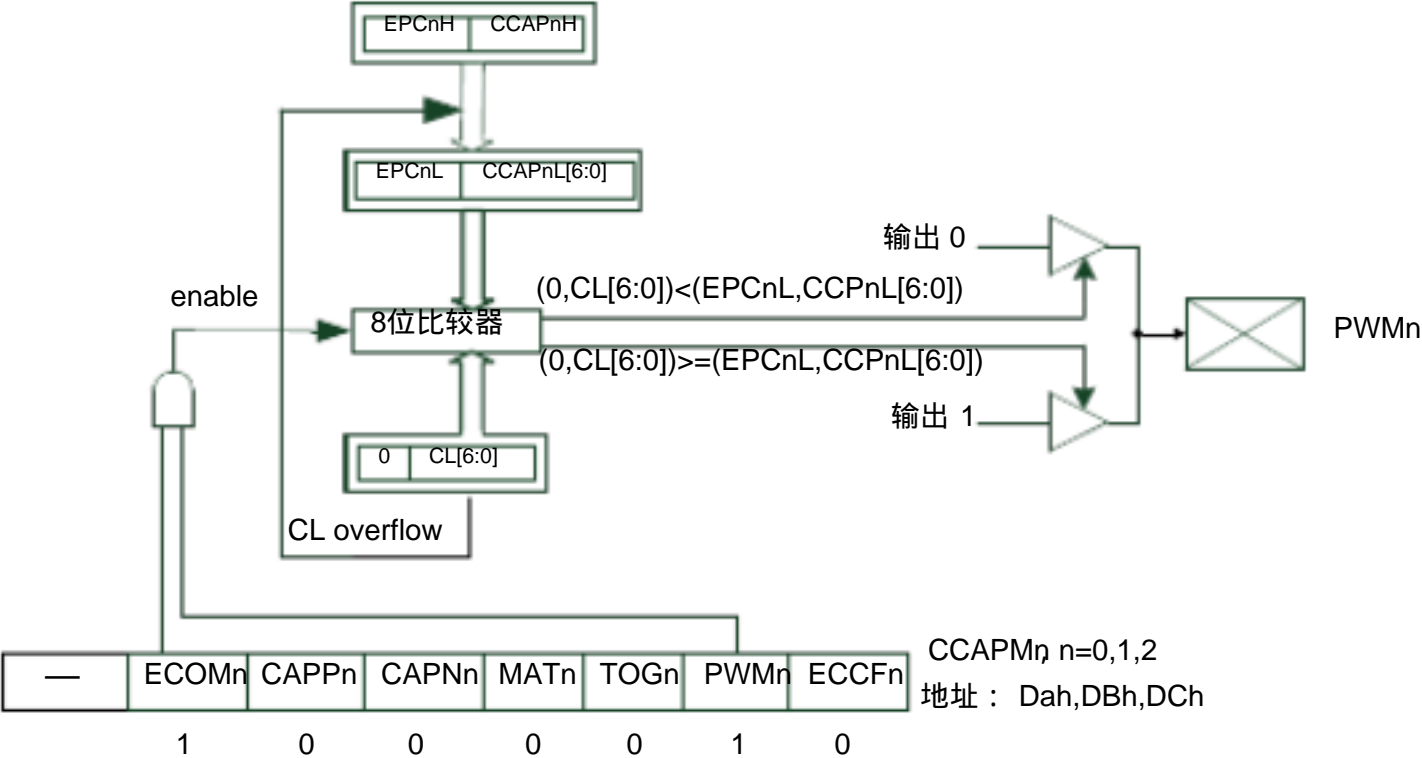
当[EBSn_1,EBSn_0]=[0,0] 或[1,1] 时 ,PCA 模块 n 工作 8 位 PWM 模式 ,此时将 {0 ,CL[7:0]} 与捕获寄存器 [EPCnL,CCAPnL[7:0]] 进行比较。

PWM 模式的结构如下图所示。



3.2 7 位脉宽调节模式（ PWM ）

当[EBSn_1,EBSn_0]=[0,1] 时，PCA 模块 n 工作 7 位 PWM 模式，此时将 {0，CL[6:0]} 与捕获寄存器 [EPCnL,CCAPnL[6:0]] 进行比较。PWM 模式的结构如下图所示。



PCA PWM mode/ 可调制脉冲宽度输出模式结构图（ PCA 模块工作于 7 位 PWM 模式）
当 PCA 模块工作于 7 位 PWM 模式时，由于所有模块共用仅有的 PCA 定时器，所有它们的输出频率相同。各个模块的输出占空比是独立变化的，与使用的捕获寄存器 {EPCnL,CCAPnL[6:0]} 有关。当 {0，CL[6:0]} 的值小于 {EPCnL，CCAPnL[6:0]} 时，输出为低；当 {0，CL[6:0]} 的值等于或大于 {EPCnL,CCAPnL[6:0]} 时，输出为高。当 CL 的值由 7F 变为 00 溢出时，{EPCnH,CCAPnH[6:0]} 的内容装载到 {EPCnL,CCAPnL[6:0]} 中。这样就实现无干扰地更新 PWM。要使能 PWM 模式，模块 CCAPMn 寄存器的 PWMn 和 ECOMn 位必须置位。

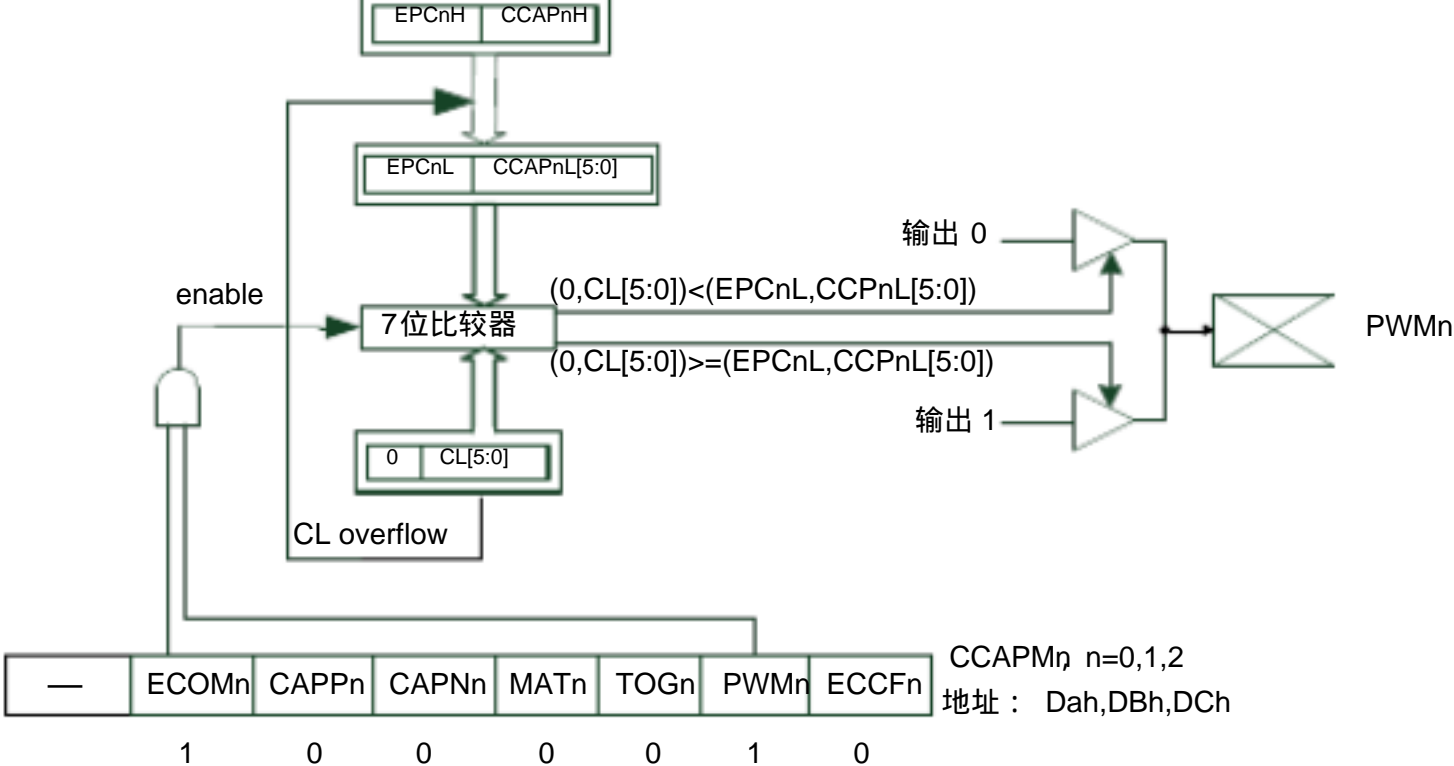
PCA时钟输入源频率

当 PWM是 7 位的时： PWM的频率 = $\frac{128}{\text{PCA时钟输入源频率}}$

PCA 时钟输入源可以从以下 8 中中选择一种： SYSclk，SYSclk/2，SYSclk/4，SYSclk/6，SYSclk/8，SYSclk/12，定时器 0 的溢出，ECI/P1.2 输入。
如果要实现可调频率的 PWM 输出，可选择定时器 0 的溢出率或者 ECI 脚的输入作为 PCA/PWM 的时钟输入源
当 EPCnL = 0 及 CCAPnL = 80H 时，PWM 固定输出高
当 EPCnL = 1 及 CCAPnL = 0FFH 时，PWM 固定输出低

3.2 6 位脉宽调节模式（ PWM ）

当[EBSn_1,EBSn_0]=[1,0] 时，PCA 模块 n 工作 6 位 PWM 模式，此时将 {0，CL[5:0]} 与捕获寄存器 [EPCnL,CCAPnL[5:0]] 进行比较。PWM 模式的结构如下图所示。



PCA PWM mode/ 可调制脉冲宽度输出模式结构图（ PCA 模块工作于 6 位 PWM 模式）
当 PCA 模块工作于 6 位 PWM 模式时，由于所有模块共用仅有的 PCA 定时器，所有它们的输出频率相同。各个模块的输出占空比是独立变化的，与使用的捕获寄存器 {EPCnL,CCAPnL[5:0]} 有关。当 {0，CL[5:0]} 的值小于 {EPCnL，CCAPnL[5:0]} 时，输出为

低；当 {0, CL[5:0]} 的值等于或大于 {EPCnL, CCAPnL[5:0]} 时，输出为高。当 CL 的值由 3F 变为 00 溢出时，{EPCnH, CCAPnH[5:0]} 的内容装载到 {EPCnL, CCAPnL[5:0]} 中。这样就实现无干扰地更新 PWM。要使能 PWM 模式，模块 CCAPMn 寄存器的 PWMn 和 ECOMn 位必须置位。

PCA时钟输入源频率

当 PWM 是 6 位的时：PWM 的频率 = $\frac{\text{PCA 时钟输入源频率}}{64}$

PCA 时钟输入源可以从以下 8 种中选择一种：SYSclk, SYSclk/2, SYSclk/4, SYSclk/6, SYSclk/8, SYSclk/12, 定时器 0 的溢出, ECI/P1.2 输入。

如果要实现可调频率的 PWM 输出，可选择定时器 0 的溢出率或者 ECI 脚的输入作为 PCA/PWM 的时钟输入源

当 EPCnL = 0 及 CCAPnL = 0C0H 时，PWM 固定输出高

当 EPCnL = 1 及 CCAPnL = 0FFH 时，PWM 固定输出低

4. 程序

//P3.7 输出 PWM 波形

```
#include<reg51.h>
```

```
sfr CCON = 0xd8;
```

```
sfr CMOD = 0xd9;
```

```
sfr CCAPM2 = 0xdc;
```

```
sfr CL = 0xe9;
```

```
sfr CH = 0xf9;
```

```
sfr CCAP2L = 0xec;
```

```
sfr CCAP2H = 0xfc;
```

```
sfr PCA_PMW2 = 0xf4;
```

```
code unsigned char seven_seg[] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
```

```
unsigned char cp1, cp2 = 255;
```

```
unsigned int cp0;
```

```
sbit P3_4 = P3^4;
```

```
sbit P3_5 = P3^5;
```

```
sbit P3_6 = P3^6;
```

```
void timer0_isr(void) interrupt 1
```

```
{
```

```
    TH0 = (65536 - 1000) / 256; // 重装初值
```

```
    TL0 = (65536 - 1000) % 256; // 重装初值
```

```
    cp0++; // 中断 1 次，变量加 1
```

```
    if(cp0 >= 100) // 1 秒到了
```

```
    {
```

```
        cp0 = 0;
```

```
        cp2--; // cp2 为脉宽控制变量
```

```
        if(cp2 <= 1)
```

```
        {
```

```
            cp2 = 255;
```

```
        }
```

```
    P2 = 0xff;
```

```
    P3 = 0xff;
```

```
    switch(cp1)
```

```
    {
```

```

    case 0: P2 = seven_seg[cp2 % 100 % 10]; P3_6 = 0; break;
    case 1: P2 = seven_seg[cp2 % 100 / 10]; P3_5 = 0; break;
    case 2: P2 = seven_seg[cp2 / 100]; P3_4 = 0; break;
}
cp1++;
if(cp1 >= 3) cp1 = 0;
}
/*****Timer0          初始化函数 *****/
void timer0_init(void)
{
    TMOD = 0x01;          //T0 工作方式 1
    TH0 = (65536 - 1000) / 256; // 对机器脉冲计数 1000 个计满溢出引发中断
    TL0 = (65536 - 1000) % 256;
    EA = 1;                //开总中断
    ET0 = 1;                //开 T0 中断
    TR0 = 1;                //启动定时器 T0
}
void pwm0_init(void)
{
    CCON = 0x00;
    CL = 0x00;
    CH = 0x00;
    PCA_PMW2 = 0x00; //8 位 PMW , 占空比的第九位 EPC0L 为 0
    CCAPM2 = 0x42; //允许比较, P3.7 输出
    CCON = 0x40; //允许 PAC 计数
}
void main(void)
{
    timer0_init();
    pwm0_init();
    while(1)
    {
        if(CL == 0xff)
        {
            CCAP2L = cp2;
            CCAP2H = cp2;
        }
    }
}

```