PWM使用说明

问题描述

PWM使用说明

影响范围

蓝牙全系列芯片产品

解决方案

TIMER 输出 PWM

AB530X

AB530x 支持3个定时器（Timer3,Timer4,Timer5,）PWM输出，每个定时器支持3路PWM输出 ，分别是PWM0 ，PWM1，PWM2，其中每个定时器支持3路周期相同，可设置不同占空比的PWM同时输出。

每组PWM对应的引脚

- Timer3 : PWM0→PB0 ,PWM1→PB1 ,PWM2→PB2;
- Timer4 : PWM0→PA5 ,PWM1→PA6 ,PWM2→PA7;
- Timer5 : PWM0→PE1 , PWM1→PE2 ,PWM2→PE3;

AB532X

AB532X 每组timer对应PWM IO说明：

- Timer3 :
  - G1：PWM0→PA2 ,PWM1→PA3 ,PWM2→PA4;
  - G2：PWM1→PB3,PWM2→PB4;
  - G3：PWM0→PB0,PWM1→PB1,PWM2→PB2;
  - G4：PWM0→PE0,PWM1→PE1;
  - G5：PWM0→PF2,PWM1→PF3,PWM2→PF4;
- Timer4 :
  - G1：PWM0→PA0 ,PWM1→PA1 ;
  - G2：PWM0→PE5,PWM→PE6,PWM2→PE7;
  - G3：PWM0→PF1,PWM1→PF0,PWM2→PF5;
- Timer5 :
  - G1：PWM0→PA5 , PWM1→PA6 ,PWM2→PA7;
  - G2：PWM0→PE2,PWM1→PE3,PWM2→PE4;
  - G3：PWM0→PB5,PWM1→PB6,PWM2→PB7;

例程

下面以timer5为例，

```
void timer5_pwm_test(void)
{
    GPIOEDIR &= ~(BIT(1) | BIT(2) | BIT(3)); //DIR OUT
    GPIOEDE |= (BIT(1) | BIT(2) | BIT(3));   //数字IO口要设置上.
    GPIOEFEN |= (BIT(1) | BIT(2) | BIT(3));  //function io enable
    FUNCMCON2 = (1 << 16); //pwm map to G1 (PE1/2/3)  timer4->12Bit  timer3->8Bit

    TMR5CON =  0x00;           //时钟源默认为系统时钟24M
    TMR5CNT = 0;
    TMR5PR  = 24000000L/500-1;  //例如为24M时, 频率为 24M/(12000*4) = 500HZ, 即周期为0.002S.   //U32
    TMR5DUTY0 = 12000 - 1;     //u16 //25% 负占空比  PE1     //注意 PR >= DUTY    //  DUTY ==PR, 全高.   -1 (也就是0xFFFF)时, 全低, 其他值表示"负占空比"
    TMR5DUTY1 = 12000*2 - 1;    //u16 //50% 负占空比  PE2
    TMR5DUTY2 = 12000*3- 1;     //u16 //75% 负占空比  PE3
    TMR5CON |= (BIT(9) | BIT(10) | BIT(11));    //timer5 pwm0, pwm1, pwm2 enable
    TMR5CON |= BIT(0);               //Timer enable (defualt work in counter mode)
}
```

占空比duty = TMR5DUTY1 /TMR5PR ;负占空比 SDK默认定时器用途：

- timer0 : 系统时钟使用
- timer1:（可选）ledseg_ajust. 7脚屏的时候使用. timer1_irq_init
- timer3:（可选）软件红外解码使用
- timer4/timer5 : tws使用

低功耗 LPWM 输出

AB530X/AB531X 还支持3组（G1，G2，G3）硬件LPWM(低功耗PWM)输出。

每组有4路PWM ，分别是PWM0 ，PWM1，PWM2，PWM3，每组PWM可独立输出。

每组PWM对应的引脚

- G1 : PWM0→PE4 ,PWM1→PE5 ,PWM2→PE6 ,PWM3→PE7;
- G2 : PWM0→PB3 ,PWM1→PB4 ,PWM2→PE0 ,PWM3→PF2;
- G3 : PWM0→PA0 ,PWM1→PA1 ,PWM2→PA2 ,PWM3→PA3;

例程

```
#define LPWM_G1                 1
#define LPWM_G2                 2
#define LPWM_G3                 3
#define LPWM_MAP                LPWM_G1

void pwm_test(void)
{
#if (LPWM_MAP == LPWM_G1)
    GPIOEDIR &= ~(BIT(4) | BIT(5) | BIT(6) | BIT(7));
GPIOEFEN |= BIT(4) | BIT(5) | BIT(6) | BIT(7);
GPIOE &=  ~(BIT(4) | BIT(5) | BIT(6) | BIT(7));
#elif (LPWM_MAP == LPWM_G2)
    USBCON0 = 0;
    USBCON1 = 0;
    GPIOBDIR &= ~(BIT(3) | BIT(4));
GPIOBFEN |= BIT(3) | BIT(4);
    GPIOEDIR &= ~BIT(0);
GPIOEFEN |= BIT(0);
    GPIOFDIR &= ~BIT(2);
GPIOFFEN |= BIT(2);
    GPIOB &= ~(BIT(3) | BIT(4));
    GPIOE &= ~BIT(0);
    GPIOF &= ~BIT(2);
#elif (LPWM_MAP == LPWM_G3)
    GPIOADIR &= ~0x0F;              // PA0 ~ PA3 对应 PWM0~PWM3
 GPIOAFEN &= ~0X0F;
    GPIOA &= ~0x0F;
#endif

    FUNCMCON1 = (LPWM_MAP << 16) | (LPWM_MAP << 20) | (LPWM_MAP << 24) | (LPWM_MAP << 28);   16-> PWM0   20->PWM1   24->PWM2  28->PWM3
    PWMPRCON = 0x658F;  //PWM period =26MHz/4/ (PWMPRCON+1) = 250Hz = 4ms
    PWM01DUTY = 0x32c84c2c;   //DUTY: 占空比PWM0 bit0~bit15,PWM1 bit 16~31bit
    PWM23DUTY = 0x32c81964;   //DUTY: 占空比PWM2 bit0~bit15,PWM3 bit 16~31bit

    PWMSTEP = 0x68686868;           //PWMSTEP PWM0(0~7) ~ PWM3(24~31);有效位7bit，最高位为符号位 （按STEP加减调节到DUTY或0） 0x68->16us
    PWMCYCNUM = 0x01010101;         //CYCNUM PWM0(0~7) ~ PWM3(24~31);当前STEP的PWM输出次数 = PWMCYCNUM + 1
    PWMCON = 0;
//  PWMCON |= BIT(5);   //PWMIVN=0:设置正占空比； PWMIVN=1:设置负占空比

    PWMCON |= BIT(4);                //PWM Auto Adjust enable
    PWMCON |= 0x0f;                  //PWM0, PWM1, PWM2, PWM3 enable
}
```