实验 6　定时器 PWM 输出及应用

一．实验目的

1. 掌握利用定时器输出 PWM 波形方法；
2. (提高)掌握低功耗模式控制方法。

二．实验任务

1. 利用定时器 A 比较输出功能，输出 PWM 波形

1） 阅读 L6_PWM.c 程序，了解定时器 PWM 波形实现原理；

2） 将 P2.1 和 P2.4 两引脚分别连到两发光二级管上，运行 L6_PWM.c 程序，通过观察二级管闪烁的频率，以及亮、灭时间的相对长度，了解 PWM 输出波形的频率、占空比控制方法，其中占空比是指高电平占一个周期中的比例。可在 DEBUG 下，如图 6-1，在 View/Registers 串口下，修改 TA1CCR0、TACCR1、TACCR2 的置，观察现象，理解程序实现原理；（特点:同一个定时器的两个比较器）。

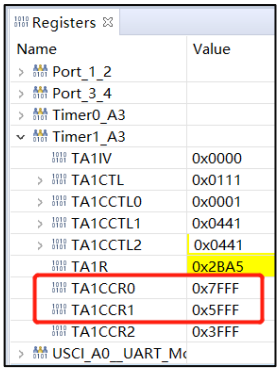

图 6-1 修改定时器设置值

3） 在 L6_PWM.c 的基础上，编程在 P2.1 、P2.4 上输出频率为 0.5Hz、占空比为 90%、10%的方波；（特点：同一个定时器的两个比较器，改频率，改占空比）。

4） 编程在 P1.6 引脚上输出频率为 1Hz、占空比为 50%的 PWM 波形;
在 P2.5 引脚上输出频率为 0.5Hz、占空比为 50%的 PWM 波形。
(特点：不同定时器，不同比较器)。

注意：PWM 波形的输出本身与中断无关，
只与定时器的工作设置有关。

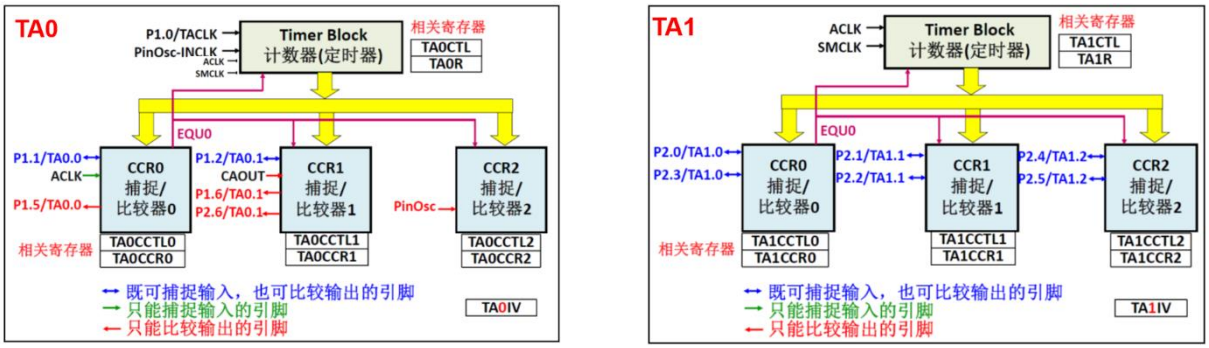

图 6-2 msp430g2553 定时器 A 引脚分配

L6_PWM.c

```
#include "msp430.h"
int main ( void )
{    WDTCTL = WDTPW + WDTHOLD;          //关闭看门狗
     P2SEL |=BIT1+BIT4;                 //置 P2.1 和 P2.4 为定时器 TA1 的 PWM 输出引脚
     P2SEL2 &=~(BIT1+BIT4);        //P2.1 为比较器 1 的 PWM 输出引脚
     P2DIR |=BIT1+BIT4;            //P2.4 为比较器 2 的 PWM 输出引脚

     TA1CTL |=TASSEL0;             //选择 TA1 计数时钟为 ACLK，使用上电复位设置，即 32768Hz
     TA1CCR0=32767;                //设置 PWM 周期，周期= (TA1CCR0+1)*T =(TA1CCR0+1)/计数时钟频率
                                   //即 PWM 频率=1/PWM 周期=计数时钟频率/(TA1CCR0+1)
     TA1CCTL1|=OUTMOD1;            //设置 TA1 比较器 1 的 PWM 输出为模式 2：  计数到 CCR1 值翻转，到 CCR0 值置 0
     TA1CCR1=24575;                //设置 TA1 比较器 1 设定值 CCR1，TA1CCR1=TA1CCR0*(1-PWM 波形占空比 25%)

     TA1CCTL2|=OUTMOD1;            //设置比较器 2 的 PWM 输出为模式 2：  计数到 CCR2 值翻转，到 CCR0 值置 0
     TA1CCR2=16383;                //设置 TA1 比较器 2 设定值 CCR2，TA1CCR2=TA1CCR0*(1-PWM 波形占空比 50%)

     TA1CTL |=TACLR+MC0;           //设置增计数方式，使计数器从 0 开始计数，计数到 TA1CCR0 后又从 0 计数。

       while(1){   };              // 主循环，CPU 可做其他事情
}
```

2. 蜂鸣器的发生控制

在任务 1 的基础上，将定时器输出的 PWM 波连接到蜂鸣器的控制引脚上，编程控制蜂鸣器发出低、中、高三个音域 1、2、3、4、5、6、7 等不同音符的音调。可分别用无源和有源蜂鸣器完成，比较二者的不同之处。

蜂鸣器的发声原理：

蜂鸣器分有源和无源蜂鸣器两种。这里的源，指的是振荡源。

有源蜂鸣器，内部有振荡源，在它的两个引脚接上电压就会发出固定频率的声音；

无源蜂鸣器，也称电磁式蜂鸣器，内部没有振荡源，主要由永磁体、线圈、振荡片构成。需要由外部提供一定频率的方波信号作为振荡信号(源)，作用于内部线圈，使内部振荡片产生振动，发出声音。

两种蜂鸣器的发声可由输入的方波信号来改变，其中音调的高低由方波的频率决定，音长由方波的个数，即输出方波的时间长短决定。调节各音的音调和音长，可以使蜂鸣器发出不同旋律的音乐。

表 6-1 是钢琴中央 C 开始的一个 8 度的各音符频率，高 8 度和低 8 度各音的频率分别是表中对应唱名音的 2 倍（高音区）和 1/2（低音区），更多音符对应的相应频率可参看图 6-3。

有源蜂鸣器的发声除控制方波的频率外，还需控制方波的占空比为 90%以上，减小其固有频率带来的杂声，音质效果才较好。

表 6-1　C 大调各音符参考频率

唱名	简谱	C 调(Hz)	C 调(Hz)(取整)
do	1	261.6	262
re	2	293.7	294
mi	3	329.6	330
fa	4	349.2	349
sol	5	392.0	392
la	6	440.0	440
si	7	493.9	494
do(高)	i	523.2	523

3. 调节小车转动速度

将定时器 TA 输出的两路 PWM 波分别连接小车的控制端 PWMA、PWMB，完成：

1) 控制小车以低速、中速、快速 3 个档位速度前行或后退；
2) 控制小车在前行的过程中左转、右转；
3) 控制小车在后退的过程中左转、右转；

4. (提高)低功耗模式学习

用跳线将 P2.3 与发光二极管 L4 短接，将 P2.4 用长杜邦线与蜂鸣器 buzz 短接，P1.1 与按键 K2 短接，利用程序 L4_LPM.c，完成下面操作，了解低功耗模式的进入和退出。

1) 运行程序，观察现象，记录进入低功耗前、进入低功耗后、响应中断后、退出中断后的发光二极管和蜂鸣器状态，并做分析。
2) 如果中断程序中有 LPM4_EXIT 语句，运行的结果会有什么不同？请分析。

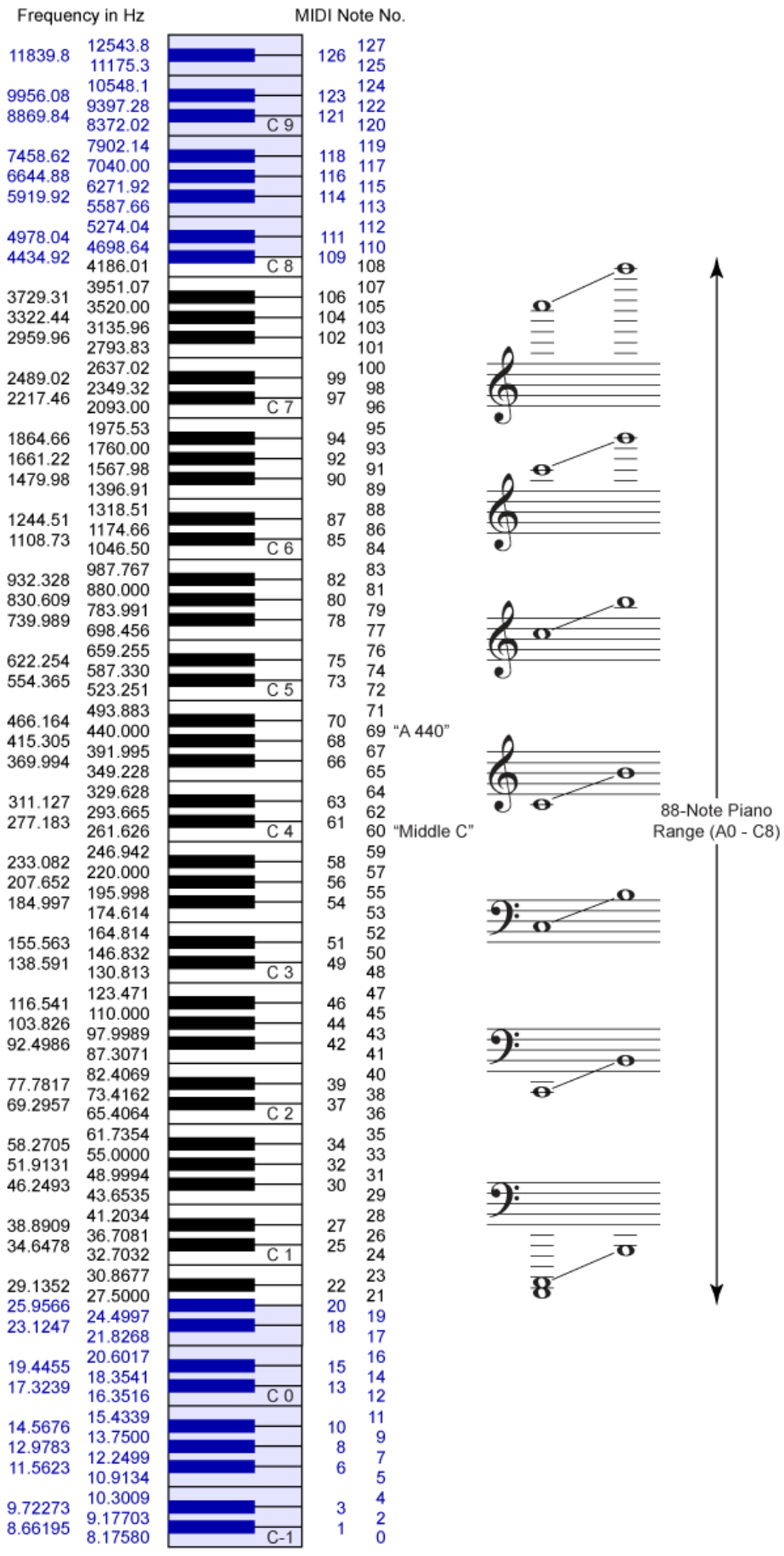
Frequency in Hz
MIDI Note No.
11839.8
12543.8
11175.3
9956.08
10548.1
8869.84
9397.28
8372.02
C 9
7458.62
7902.14
6644.88
7040.00
5919.92
6271.92
5587.66
4978.04
5274.04
4434.92
4698.64
4186.01
C 8
3729.31
3951.07
3322.44
3520.00
2959.96
3135.96
2793.83
2489.02
2637.02
2217.46
2349.32
2093.00
C 7
1864.66
1975.53
1661.22
1760.00
1479.98
1567.98
1396.91
1244.51
1318.51
1108.73
1174.66
1046.50
C 6
932.328
987.767
830.609
880.000
739.989
783.991
698.456
622.254
659.255
554.365
587.330
523.251
C 5
466.164
493.883
415.305
440.000
369.994
391.995
349.228
311.127
329.628
277.183
293.665
261.626
C 4
233.082
246.942
207.652
220.000
184.997
195.998
174.614
155.563
164.814
138.591
146.832
130.813
C 3
116.541
123.471
103.826
110.000
92.4986
97.9989
87.3071
77.7817
82.4069
69.2957
73.4162
65.4064
C 2
58.2705
61.7354
51.9131
55.0000
46.2493
48.9994
43.6535
38.8909
41.2034
34.6478
36.7081
32.7032
C 1
29.1352
30.8677
25.9566
27.5000
23.1247
24.4997
21.8268
19.4455
20.6017
17.3239
18.3541
16.3516
C 0
14.5676
15.4339
12.9783
13.7500
11.5623
12.2499
10.9134
9.72273
10.3009
8.66195
9.17703
8.17580
C-1
126
127
125
123
124
122
121
120
118
119
117
116
115
114
113
111
112
110
109
108
106
107
105
104
103
102
101
99
100
98
97
96
94
95
93
92
91
90
89
87
88
86
85
84
82
83
81
80
79
78
77
75
76
74
73
72
70
71
69
"A 440"
68
67
66
65
63
64
62
61
60
"Middle C"
58
59
57
56
55
54
53
51
52
50
49
48
46
47
45
44
43
42
41
39
40
38
37
36
34
35
33
32
31
30
29
27
28
26
25
24
22
23
21
20
19
18
17
15
16
14
13
12
10
11
9
8
7
6
5
3
4
2
1
0
88-Note Piano
Range (A0 - C8)

图 6-3 钢琴各音频率表

任务 4：L6_LPM.c（提供电子版）：

```
#include "msp430.h"
void delay( unsigned int i)          //延时函数
{    unsigned int j;                 //定义局部变量
     for (j=0;j<i;j++);
}
void Blink( )                        //LED 闪烁
{    unsigned int i;
     for (i=0;i<3;i++)
     { P2OUT &= ~BIT3;
          delay(0xe000);
        P2OUT |=BIT3;
           delay(0xe000);
     };
}
void Buzz( )                         //蜂鸣响
{    unsigned int i;
     for (i=0;i<2;i++)
     { P2OUT &= ~BIT4;
          delay(0xf800);
        P2OUT |=BIT4;
          delay(0xf800);
     };
}
int   main ( void )
{
    WDTCTL = WDTPW + WDTHOLD;              //关闭看门狗

//设置端口 P2.3 输出，控制 LED,P2.4 输出，控制蜂鸣器
     P2SEL &=~(BIT3+BIT4);
     P2SEL2&=~(BIT3+BIT4);
     P2OUT |= BIT3+BIT4;
     P2DIR |= BIT3+BIT4;

//设置端口 P1.1 允许中断
     P1SEL &=~BIT1;
     P1SEL2 &=~BIT1;
     P1REN |=BIT1;
     P1OUT |=BIT1;
     P1DIR &=~BIT1;
     P1IES |=BIT1;
     P1IFG &=~BIT1;
     P1IE |=BIT1;
     _EINT();

//P1.0 输出时钟 ACLK, P1.4 输出时钟 SMCLK
     P1SEL |=BIT0+BIT4;
     P1SEL2 &=~(BIT0+BIT4);
     P1DIR |=BIT0+BIT4;

     Blink();
     Buzz();

     for (;;)   //主循环
     {    LPM4;
          Blink();
     }
}

#pragma   vector=PORT1_VECTOR          //中断向量设置
__interrupt   void   port_ISR( )       //中断函数
{    Buzz();
     P1IFG&=~(BIT1);                   //清中断标志
//   LPM4_EXIT;
}
```