

# 基于自定义PWM的乐曲演奏实验



# 目录

- 实验内容与目的
- 大长今乐谱
- 实验原理
- 硬件系统构建
- 蜂鸣器连接
- C编程

# 实验内容与目的

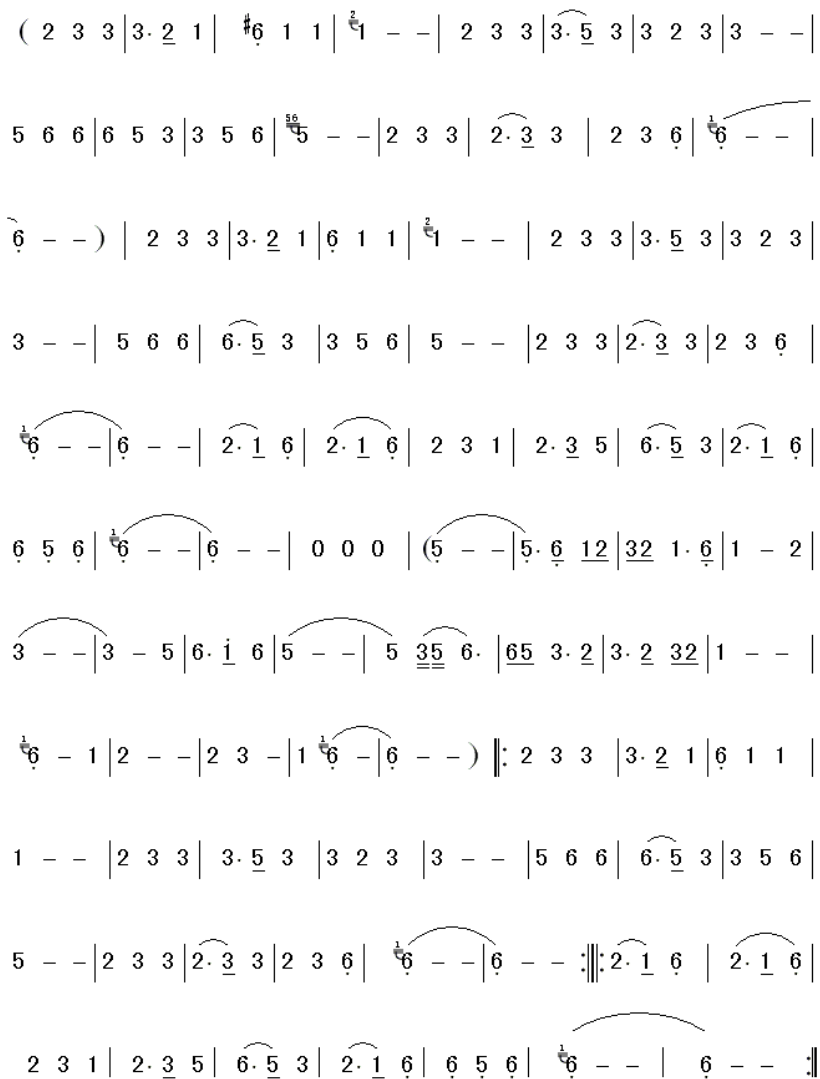
- 利用自定义的**PWM**外设和蜂鸣器，完成乐谱的演奏，如大长今，最好自己找一个。
- 熟悉自定义外设的设置与使用，初步体会软硬件协同设计。

# 大长今乐谱

## 大长今

韩国电视连续剧《大长今》主题曲

1=G  $\frac{3}{4}$   
♩=100



# 实验原理1

表 9.13 简谱中的音名与频率的关系

音 名	频 率/Hz	音 名	频 率/Hz	音 名	频 率/Hz
低音 1	261.6	中音 1	523.3	高音 1	1046.5
低音 2	293.7	中音 2	587.3	高音 2	1174.7
低音 3	329.6	中音 3	659.3	高音 3	1319.5
低音 4	349.2	中音 4	699.5	高音 4	1396.9
低音 5	392	中音 5	784	高音 5	1568
低音 6	440	中音 6	880	高音 6	1760
低音 7	493.9	中音 7	987.8	高音 7	1975.5

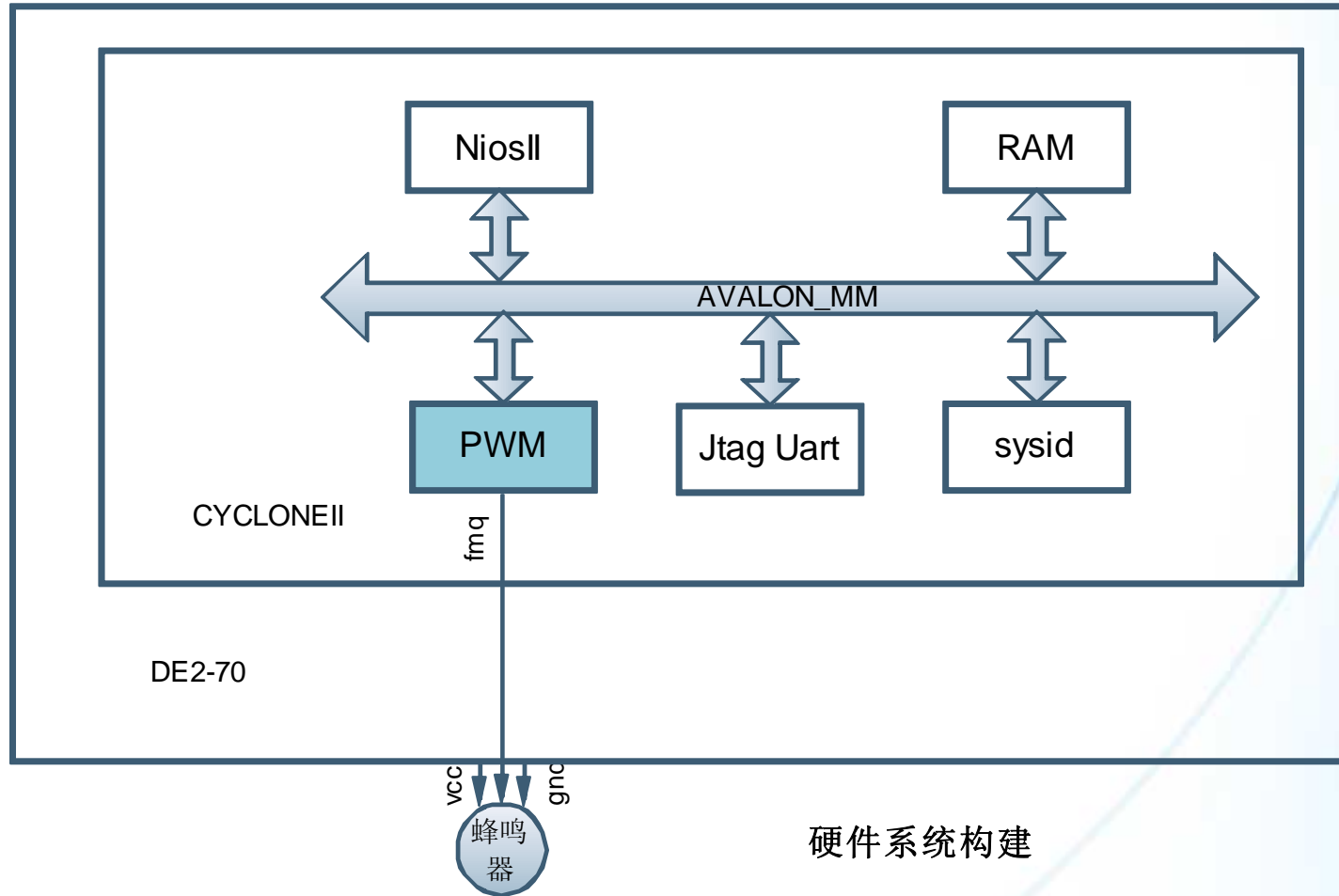


# 实验原理2

表 9.14 各音阶频率对应的分频比及预置数（从 3 MHz 频率计算得出）




音 名	分 频 比	预 置 数	音 名	分 频 比	预 置 数
低音 1	11 468	4 915	中音 5	3 827	12 556
低音 2	10 215	6 168	中音 6	3 409	12 974
低音 3	9 102	7 281	中音 7	3 037	13 346
低音 4	8 591	7 792	高音 1	2 867	13 516
低音 5	7 653	8 730	高音 2	2 554	13 829
低音 6	6 818	9 565	高音 3	2 274	14 109
低音 7	6 073	10 310	高音 4	2 148	14 235
中音 1	5 736	10 647	高音 5	1 913	14 470
中音 2	5 111	11 272	高音 6	1 705	14 678
中音 3	4 552	11 831	高音 7	1 519	14 864
中音 4	4 289	12 094	休止符	0	16 383

# 硬件系统构建



# 蜂鸣器连接

Named:  Edit:   PIN\_AJ6

		Node Name	Location
1		iCLK	PIN_AD15
2		iKEY[0]	PIN_T29
3		LEDR[0]	PIN_C30
4		<<new node>>	

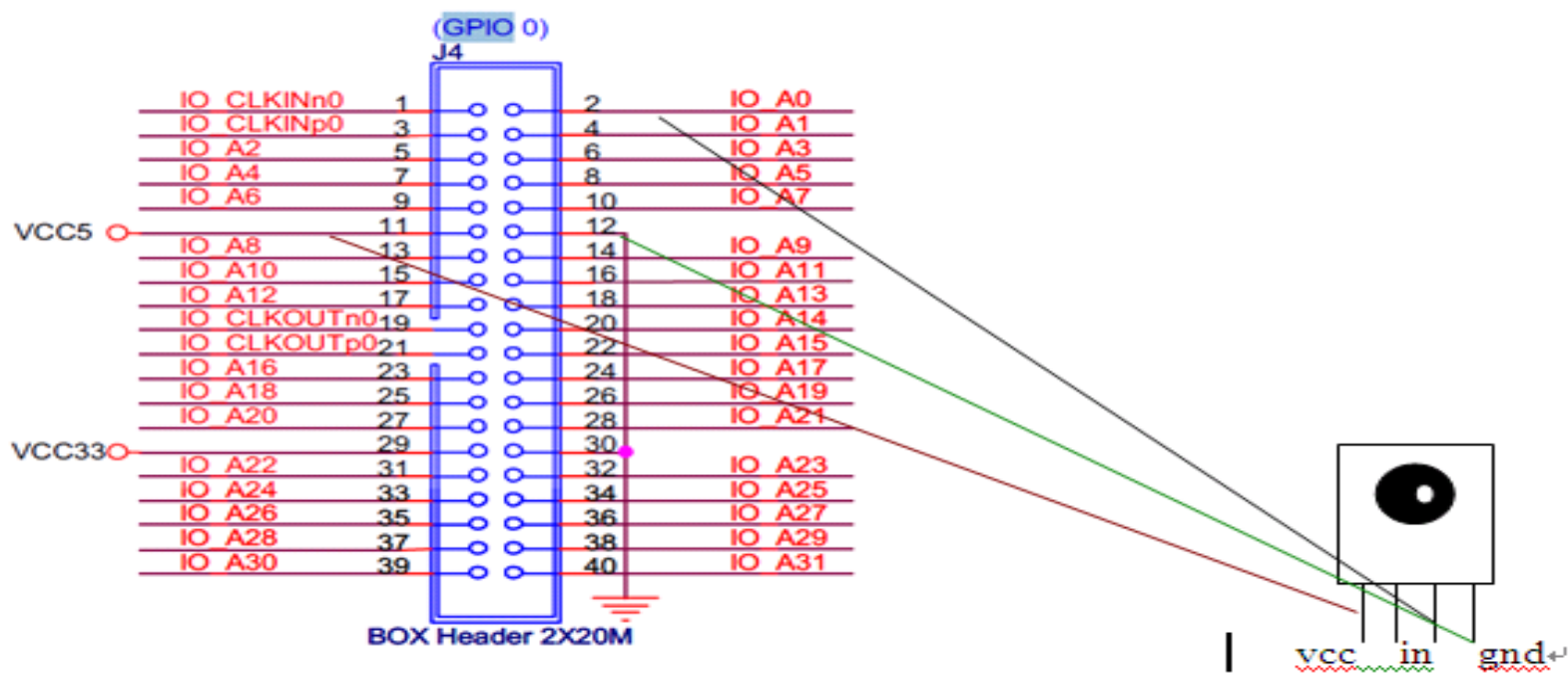


图 1-1 扩展插槽 J4 引脚图



# C编程：头文件

- **#include "system.h"**
- **#include <io.h>**

# C编程：音长参数定义

- `#define f 1000000`
- `#define rhythm 5000000`
- `#define ff f*50`
  
- `#define _1 rhythm*4`
- `#define _1d rhythm*6`
- `#define _2 rhythm*2`
- `#define _2d rhythm*3`
- `#define _4 rhythm*1`
- `#define _4d (rhythm*3)/2`
- `#define _8 (rhythm*1)/2`
- `#define _8d (rhythm*3)/4`
- `#define _16 rhythm*1/4`
- `#define _16d (rhythm*3)/8`
- `#define _32 (rhythm*1)/8`

# C编程：音色参数—低音

- `#define f 1000000`
- `#define rhythm 5000000`
- `#define ff f*50`
  
- `#define _1do (ff/131)/2`
- `#define _1dor (ff/139)/2`
- `#define _1re (ff/147)/2`
- `#define _1rer (ff/155)/2`
- `#define _1mi (ff/165)/2`
- `#define _1fa (ff/175)/2`
- `#define _1far (ff/185)/2`
- `#define _1sol (ff/196)/2`
- `#define _1solr (ff/207)/2`
- `#define _1la (ff/220)/2`
- `#define _1lar (ff/233)/2`
- `#define _1si (ff/247)/2`

# C编程：音色参数—中音

- `#define f 1000000`
- `#define rhythm 5000000`
- `#define ff f*50`
  
- `#define _do (ff/262)/2`
- `#define _dor (ff/277)/2`
- `#define _re (ff/294)/2`
- `#define _rer (ff/311)/2`
- `#define _mi (ff/330)/2`
- `#define _fa (ff/349)/2`
- `#define _far (ff/370)/2`
- `#define _sol (ff/392)/2`
- `#define _solr (ff/416)/2`
- `#define _la (ff/440)/2`
- `#define _lar (ff/466)/2`
- `#define _si (ff/492)/2`

# C编程：音色参数—高音

- `#define f 1000000`
- `#define rhythm 5000000`
- `#define ff f*50`
  
- `#define _do1 (ff/523)/2`
- `#define _do1r (ff/554)/2`
- `#define _re1 (ff/579)/2`
- `#define _re1r (ff/740)/2`
- `#define _mi1 (ff/651)/2`
- `#define _fa1 (ff/695)/2`
- `#define _fa1r (ff/740)/2`
- `#define _sol1 (ff/784)/2`
- `#define _sol1r (ff/830)/2`
- `#define _la1 (ff/880)/2`
- `#define _la1r (ff/932)/2`
- `#define _si1 (ff/983)/2`

## C编程：音高参数

- **#define** mute(tone) (tone)>>3
- **#define** loud(tone) (tone)>>2
- **int** dachangjin[song\_size][3]={
- {\_la,\_4,loud(\_la)},
- {\_si,\_4,mute(\_si)}, ◦ ◦ ◦



# C编程：乐曲编程（第一段）

■ **int** dachangjin[song\_size][3]={

■     ○   ○   ○   ○

■     { \_re1, \_4, loud(\_re1) },

■     { \_mi1, \_4, mute(\_mi1) },

■     { \_mi1, \_4, mute(\_mi1) },

■     { \_mi1, \_4d, loud(\_mi1) },

■     { \_re1, \_8, mute(\_re1) },

■     { \_si, \_4, mute(\_si) },

■     { \_si, \_4, loud(\_si) },

■     { \_re1, \_4, mute(\_re1) },

■     { \_mi1, \_8, mute(\_mi1) },

■     { \_re1, \_32, mute(\_re1) },

■     { \_mi1, \_32, mute(\_mi1) },

■     { \_re1, \_2d, mute(\_re1) },   ○   ○   ○

# C编程：乐曲演奏

```
■ int main()
■ {
■   int i;
■   while(1)
■   {
■     for(i=0;i<song_size;i++)
■     {
■       IOWR(MYPWM_0_BASE,2,1); //start PWM
■       IOWR(MYPWM_0_BASE,0,dachangjin[i][0]);
■       IOWR(MYPWM_0_BASE,1,dachangjin[i][2]);
■       delay(dachangjin[i][1]);

■       IOWR(MYPWM_0_BASE,2,0); //stop PWM
■       delay(500000); //100ms
■     }
■   }
■   return 0;
■ }
```

# C编程：延时函数

- //延时函数（延时时间为 $(2+2*i)us$ ）
- **void** delay(**alt\_u32** cnt)
- {
- **alt\_u32** i =0;
- **while**(i < cnt)
- {
- i++;
- }
- }