



基于 STM32 的 PWM 音乐播放器应用设计

马志强, 王建刚, 孙少林, 胡明

(总参谋部通信训练基地, 宣化 075100)

摘要: 基于 32 位的 STM32F103, 利用 PWM 产生的音频信号驱动蜂鸣器演奏乐曲, 实现了音乐播放器的应用设计。该播放器能实现从低音到高音的 21 个音阶, 并能根据乐谱演奏完整的曲目。测试结果表明, PWM 的输出信号与各音阶对应的声音频率基本一致, 方案切实可行。这一方法也可用于电机控制、电子琴设计等方面, 具有较好的实用性。

关键词: STM32; PWM; 蜂鸣器; 定时器; 音乐

中图分类号: TP273

文献标识码: A

Application Design of PWM Music Player Based on STM32

Ma Zhiqiang, Wang Jiangang, Sun Shaolin, Hu Ming

(Communication Training Base of PLA General Staff Headquarters, Xuanhua 075100, China)

Abstract: Based on 32-bit STM32F103, using audio signals produced by PWM to drive a buzzer to play music, a music player is designed. The player can realize 21 scales from bass to treble and play complete songs according to music scores. Test results show that the output signals of PWM are consistent with the corresponding voice frequency of the scales, and the scheme is feasible. This method can also be used in motor control, keyboard design and so on, showing good practicality.

Key words: STM32; PWM; buzzer; timer; music

在科研项目开发中, 有时会遇到需要播放电话铃声、音乐等情况。简单的做法是购买专用音乐芯片, 但该方法的缺点是播放的内容不可变, 不能很好地满足项目需求。一般地, 可采用 89C51 等单片机实现音乐播放^[1-3], 其播放内容及歌曲数量都可以随时修改, 使用上相对方便。随着 STM32 系列微处理器的出现, 其基于 ARM Cortex-M 内核的 32 位闪存微控制器, 高达 72 MHz 的主频, 高集成度、实时性、数字信号处理、低功耗、低电压操作等众多特点, 使得其应用越来越广泛。本文基于 STM32 处理器, 根据乐曲简谱制作供程序识别的乐谱, 并利用内部定时器产生 PWM 输出信号, 驱动蜂鸣器完成自定义乐谱的播放。经测试, 播放效果良好。

1 乐谱简析

1.1 音阶

音阶是音乐必不可少的要素, 主要由声音的频率决定。通过给蜂鸣器不同频率的音频脉冲, 可以产生不同的音阶, 而要产生某频率的音频脉冲, 最简单的办法是算出该音频的周期, 然后将此周期除以 2 即为半周期的时间。通过程序控制单片机某引脚半周期为“高”、半周期为“低”, 不断交替变换, 即可产生该频率的矩形波, 接到蜂鸣

器上就可发出该频率的声音。若想改变音阶, 只需要改变半周期时间即可。表 1 为 C 调时音符频率对照表, 据此可产生不同音阶的音符。“#”表示半音, 用于上升或下降半个音, 乘以 2 就提升该声音一个 8 度音阶, 减半则降一个 8 度。

表 1 C 调时音符频率对照表

音符	低音/Hz	中音/Hz	高音/Hz
1	262	523	1046
2	294	578	1175
3	330	659	1318
4	349	698	1397
5	392	784	1568
6	440	880	1760
7	494	988	1976

1.2 节拍

若要构成音乐, 光有音阶是不够的, 还需要节拍, 也就是音符持续时间的长短, 一般用拍数表示。至于 1 拍是多少秒, 没有严格的规定, 只要节拍适宜, 声音悦耳即可。假如某首歌曲的节奏是每分钟 120 拍, 那么 1 拍为 0.5 s, 1/4 拍为 0.125 s, 以此类推可得到其他节拍对应的时长。这样, 利用不同的频率, 加上与拍数对应的延时, 就构成了

乐曲。

2 STM32 中的定时器^[4]

音阶的产生与声音频率有关,为了实现不同音阶,必须能为蜂鸣器提供不同频率的脉冲。为此,选择 STM32 芯片,利用其自带的定时器,通过 PWM 产生脉冲信号。STM32 中一共有 11 个定时器,包含 2 个高级控制定时器、4 个普通定时器、2 个基本定时器,以及 2 个看门狗定时器和 1 个系统滴答定时器 SysTick。其中,TIM1 和 TIM8 是高级定时器,时钟由 APB2 的输出产生。TIM2~TIM5 是普通定时器,TIM6 和 TIM7 是基本定时器,这 6 个定时器的时钟由 APB1 的输出产生。

2.1 定时时长的计算

定时器的一个主要功能就是到指定时间就会产生一个溢出事件,这个时间的设置与定时器时钟有关,在定时器时钟基础上进行预分频,设置计数溢出大小即可。

2.1.1 系统时钟设置

要保证定时的准确性,必须先确保系统时钟的设置是我们所预期的。定时器时钟分配如图 1 所示。通过编程使 SYSCLK 为 72 MHz,APB1 预分频后得到 PCLK1 为 36 MHz,再经 TIM2~TIM7 倍频器得到 TIM2~TIM7 时钟 72 MHz。时钟源多采用 HSE(外部时钟源),对于 STM32F103,其外部时钟为 8 MHz,而 STM32F107 外部时钟为 25 MHz,因此,在使用 HSE 做时钟源时,这两种器件产生 SYSCLK 的分频和倍频方式不同,需要使用者引起注意。

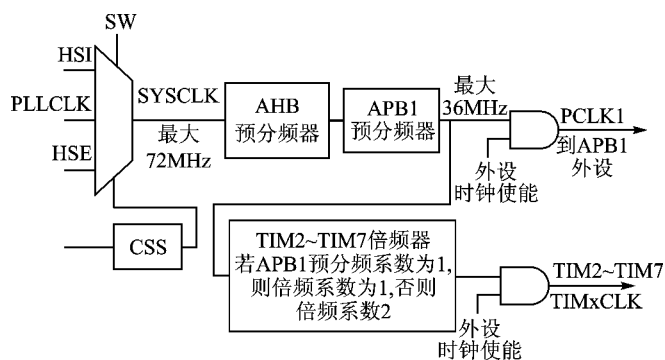


图 1 定时器时钟分配图

2.1.2 定时器相关参数设置

定时器的参数由结构体 TimeBaseInitTypeDef 定义,主要包括预分频系数、时钟分割、计数器模式、计数溢出大小等。例如,要由 TIM3(定时器 3)产生一个时长为 1 s 的定时,首先,应进行系统时钟的设置,得到 TIM3CLK=72 MHz,然后进行定时器设置。其中,预分频系数为 35 999,此时,TIM3 时钟为 72 MHz/36 000=2 kHz,无时钟分

割。设置计数溢出大小为 1 999,即每计 2 000 个数就产生一个更新事件,输出频率为 2 kHz/2 000=1 Hz。代码如下:

```
void TIMER_Config() {
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
    .....
    //预分频系数
    TIM_TimeBaseStructure.TIM_Prescaler=36000-1;
    //设置时钟分割
    TIM_TimeBaseStructure.TIM_ClockDivision=TIM_CKD_
DIV1;
    //设置计数溢出大小
    TIM_TimeBaseStructure.TIM_Period = 2000-1;
    .....
    //将配置应用到 TIM3 中
    TIM_TimeBaseInit(TIM3,&TIM_TimeBaseStructure);
}
```

2.2 STM32 的 PWM 输出

脉冲宽度调制(Pulse Width Modulation,PWM)是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术,简而言之,就是实现对输出信号脉冲宽度的控制,一般用来控制步进电机等。STM32 的定时器除了 TIM6 和 TIM7 之外,其他的定时器都可以用来产生 PWM 输出,其中,高级定时器 TIM1 和 TIM8 能够产生 3 对 PWM 互补输出,而 TIM2~TIM5 也能同时产生 4 路的 PWM 输出。

2.2.1 PWM 输出引脚

STM32 给不同的定时器分配了不同的输出引脚,考虑到引脚复用功能,STM32 还提出了一个“重映像”的概念,就是通过设置某一些相关的寄存器,使得在其他非原始指定的引脚上也能输出 PWM 波形,但是这种重映像不是随意的,使用方法可参照参考文献[1]。例如,TIM3 的通道 2,在没有重映像的时候,指定的引脚是 PA7。如果设置部分重映像之后,输出就被映像到 PB5 上了;如果设置完全重映像的话,输出就被映像到 PC7 上。

2.2.2 占空比的计算

占空比(Duty Ratio)有如下含义:在一串理想的脉冲周期序列(如方波)中,正脉冲的持续时间与脉冲总周期的比值。

$$\text{占空比} = \frac{\text{TIM_Pulse}}{(\text{TIM_Period} + 1)} \times 100\% \quad (1)$$

当 TIM_Period 为 1 999 时,若想得到占空比 50%,则 TIM_Pulse 应设置为 (1999+1)/2=1000。具体设置如下:

```
void PWM_Config() {
```

```

TIM_OCInitTypeDef TimOCInitStructure;
.....
//设置占空比
TimOCInitStructure.TIM_Pulse = 1000;
.....
//设置 TIM3 的 PWM 输出为使能
TIM_CtrlPWMOutputs(TIM3, ENABLE);
}

```

3 用 PWM 控制蜂鸣器播放音乐

3.1 硬件电路设计

蜂鸣器电路如图 2 所示。需要注意的是,有源蜂鸣器是以固定频率工作,加电即可鸣叫,无源蜂鸣器可以用不同频率输入信号来控制发声,因此,需要选择无源蜂鸣器。核心控制器器件选择 STM32F103VET6,其引脚 PB5 连接到 BEEP。由电路可知,当 PB5 为高电平时,蜂鸣器可工作,只要控制 PB5 高低电平输出形成的矩形波的频率就可以控制蜂鸣器演奏音乐。

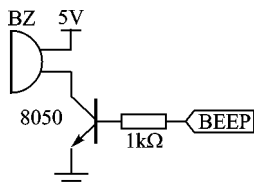


图 2 蜂鸣器电路

3.2 程序设计

3.2.1 制作乐谱

音阶的产生依赖于 PWM 输出信号的频率。为了简化设计,我们令定时器的 TIM_Period 为 1 999,且占空比始终为 50%,根据式(1)则 TIM_Pulse 为 1000。此时,PWM 输出信号频率仅与定时器预分频系数 TIM_Prescaler 有关,只需要调整该系数,即可得到所需信号频率。TIM_Prescaler 由下式得到:

$$TIM_Prescaler = \frac{72 \text{ MHz}}{(TIM_Period + 1) \times f_{\text{sound}}} - 1 \quad (2)$$

式(2)中, f_{sound} 为音阶对应的频率,如低音 Do 频率为 262 Hz。要产生该音频,TIM_Prescaler 应为 136。

乐谱由音阶和节拍组成,每两个元素为一组,前者表示音阶,后者表示节拍。节拍以 1/4 拍为基准,存放的数值为 1/4 拍的倍数。相关代码如下:

```

#define StdBeat 125 //1/4 节拍=125ms
/* 定义音阶(预分频系数) */
#define low_1 136 //低音 Do
#define low_2 121 //低音 Re
.....
#define high_7 17 //高音 Xi
/* 定义乐谱(两只老虎) */
u16 music[] = {mid_1, 4, mid_2, 4, mid_3, 4, mid_1, 4, mid_1, 4,
mid_2, 4, mid_3, 4, mid_1, 4, mid_3, 4, mid_4, 4, mid_5, 8, ...};

```

3.2.2 主程序设计

程序流程如图 3 所示。由于 STM32 的 PWM 输出引脚是 PB5,所以我们采用 TIM3 的通道 2 来产生 PWM 输出。在 GPIO 设置程序中,将 TIM3 的通道 2 引脚部分重映像到 PB5,GPIO 模式选择为复用推挽输出。程序循环读取乐谱,根据音阶修改定时器的预分频系数,并重新设置定时器和 PWM。同时,利用 STM32 的内部 SysTick 进行精确计时,根据节拍实现 ms 级延时,并减少内核消耗。图 4 为示波器测量得到的低音 So 的 PWM 输出波形。根据表 1,该波形频率应为 392 Hz,实测为 391.549 Hz,可见本方案的 PWM 输出误差较小。

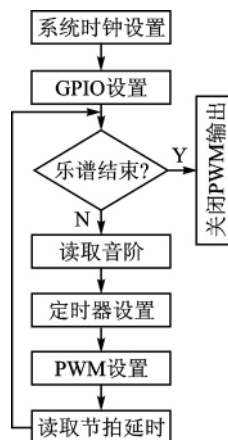


图 3 主程序流程图

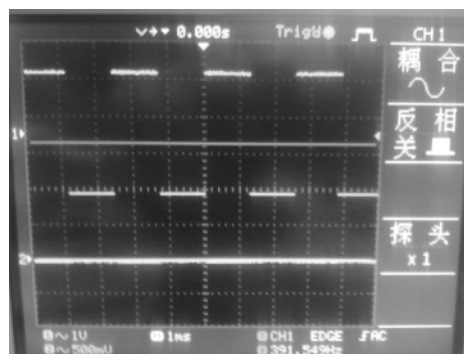


图 4 PWM 输出波形

结 语

STM32 既可以搭载 $\mu\text{C}/\text{OS}$,也可以作为单片机使用,是一款性价比比较高的处理器。本文利用 STM32 的定时器,产生 PWM 音频脉冲波形,实现了音乐播放功能。这种方法可用于电机控制、电子琴以及无线报务中的电子键设计等,具有较高的实用价值。

参考文献

- [1] 远飞. 基于单片机的音乐播放器的仿真与制作[J]. 电子元件应用, 2010, 12(12): 47-49.
- [2] 王建平, 焦国太, 陈亚星. 数字音乐盒设计[J]. 机械, 2011, 38(4): 65-67.
- [3] 马巍. 单片机制作音乐盒[J]. 职业, 2011(29): 164-165.
- [4] 意法半导体(中国)投资有限公司. STM32F10xxx 参考手册, 2010.

马志强(高级讲师),主要研究方向为无线通信、嵌入式系统和自动控制。

(责任编辑:梅栾芳 收稿日期:2012-07-09)