



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

《单片机原理及应用》作业报告

实验报告 5 第二部分：三角波生成

学院 卓越学院

学号 23040447

姓名 陈文轩

专业 智能硬件与系统(电子信息工程)

2025 年 5 月 30 日

原题目：DAC0832 与单片机的接线如课堂上所接，参考电压为-10V，请编程实现三角波的波形，三角波的周期为 20+ 作业号，单位是 ms。

1 实验代码

Code Listing 1: 实验程序

```
1
2 #include <reg51.h>
3 #define DATA P2
4 #define STUDENT_ID 47 // 学号定义为47
5
6 // 波形输出总时间 = (学号 + 20)ms
7 #define TOTAL_TIME (STUDENT_ID/1.375 + 20) // 单位: ms
8 unsigned char mode = 2; // 0:自定义波形, 1:锯齿波, 2:三角波
9
10 // 自定义波形表 - 从+1V到+2V再到0V
11 // 0V对应值为0, 1V对应值为128, 2V对应值为255
12 unsigned char code customWave[256] = {
13     // 0-15: 起始点(+1V)到上升过程
14     128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180,
15     184, 188,
16     // 16-31: 继续上升
17     192, 196, 200, 204, 208, 212, 216, 220, 224, 228, 232, 236, 240, 244,
18     248, 252,
19     // 32-47: 到达峰值(+2V)并开始下降
20     255, 255, 250, 245, 240, 235, 230, 225, 220, 215, 210, 205, 200, 195,
21     190, 185,
22     // 48-63: 继续下降
23     180, 175, 170, 165, 160, 155, 150, 145, 140, 135, 130, 125, 120, 115,
24     110, 105,
25     // 64-79: 继续下降
26     100, 95, 90, 85, 80, 75, 70, 65, 60, 55, 50, 45, 40, 35, 30, 25,
27     // 80-95: 接近0V
28     20, 15, 10, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
29     // 96-111: 保持0V一段时间
30     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
31     // 112-127: 开始回升
32     5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80,
33     // 128-143: 继续回升
34     85, 90, 95, 100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155,
35     160,
```

```
31 // 144-159: 继续回升至起始点(+IV)
32 165, 170, 175, 180, 185, 190, 195, 200, 205, 210, 215, 220, 225, 230,
    235, 240,
33 // 160-255: 保持IV电平直到循环结束
34 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128,
35 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128,
36 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128,
37 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128,
38 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128,
39 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
    128, 128
40 };
41
42
43 unsigned int timer_count = 0; // 定时器计数
44
45 // 计算锯齿波数据
46 unsigned char getSawWave(unsigned char i)
47 {
48     // 锯齿波是线性上升的波形
49     return i; // 值从0线性增加到255
50 }
51
52 // 计算三角波数据
53 unsigned char getTriangleWave(unsigned char i)
54 {
55     if(i < 128)
56         return i << 1; // 0-255上升段
57     else
58         return 255 - ((i - 128) << 1); // 255-0下降段
59 }
60
61 // 定时器0初始化
62 void Timer0Init()
63 {
64     TMOD = 0x01; // 设置定时器0为模式1(16位定时器)
65     EA = 1;      // 开总中断
66     ET0 = 1;     // 开定时器0中断
```

```
67     TRO = 1;          // 启动定时器0
68 }
69
70 // 设置定时器初值, 使每个点的延时为 TOTAL_TIME/256 ms
71 void setTimer0()
72 {
73     unsigned int delay_us = (TOTAL_TIME * 1000UL) / 256; // 每个点的延时
74     // (微秒)
75     unsigned int reload_value;
76
77     // 假设使用12MHz晶振, 每个机器周期为1us
78     reload_value = 65536 - delay_us;
79     TH0 = (unsigned char)(reload_value >> 8);
80     TLO = (unsigned char)reload_value;
81 }
82 // 定时器0中断服务函数
83 void Timer0_ISR() interrupt 1
84 {
85     timer_count = 1; // 设置标志, 表示延时完成
86     setTimer0();     // 重新设置定时器初值
87 }
88
89 void main()
90 {
91     unsigned char i;
92     unsigned char waveData;
93
94     Timer0Init(); // 初始化定时器
95     setTimer0(); // 设置定时器初值
96
97     while(1)
98     {
99         for(i=0;i<256;i++)
100         {
101             // 根据mode选择波形
102             switch(mode)
103             {
104                 case 0: // 自定义波形 (使用预先计算好的波形表)
105                     waveData = customWave[i];
106                     break;
107                 case 1: // 锯齿波
108                     waveData = getSawWave(i);
```

```
109         break;
110     case 2: // 三角波
111         waveData = getTriangleWave(i);
112         break;
113     default:
114         waveData = customWave[i]; // 默认使用自定义波形
115         break;
116     }
117     DATA = waveData;
118
119     // 等待定时器中断发生（延时结束）
120     timer_count = 0;
121     while(!timer_count);
122 }
123 }
124 }
```

2 实验效果

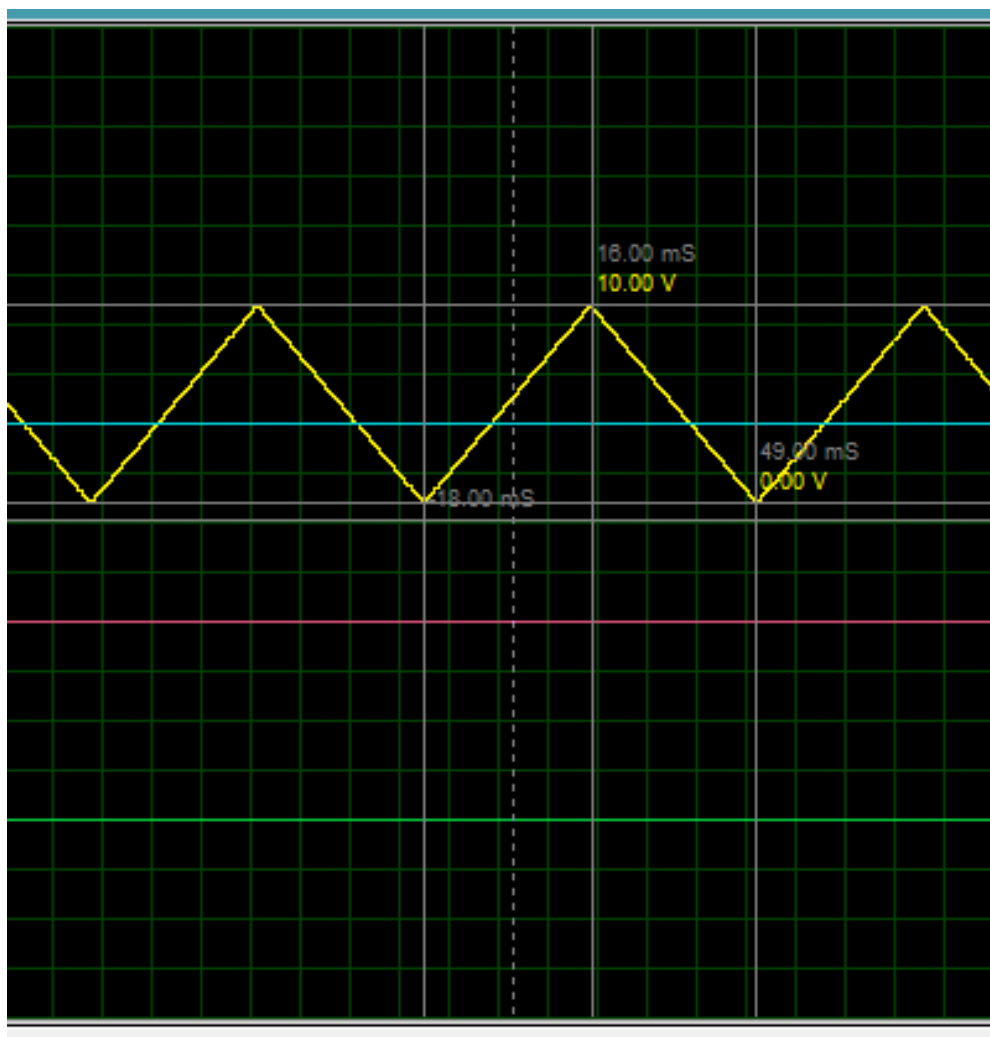


图 1 代码控制效果

示波器光标测量得：周期为 67ms，符合要求；DC 挡位测量电压变化为 0 至 10V，符合实验预期。

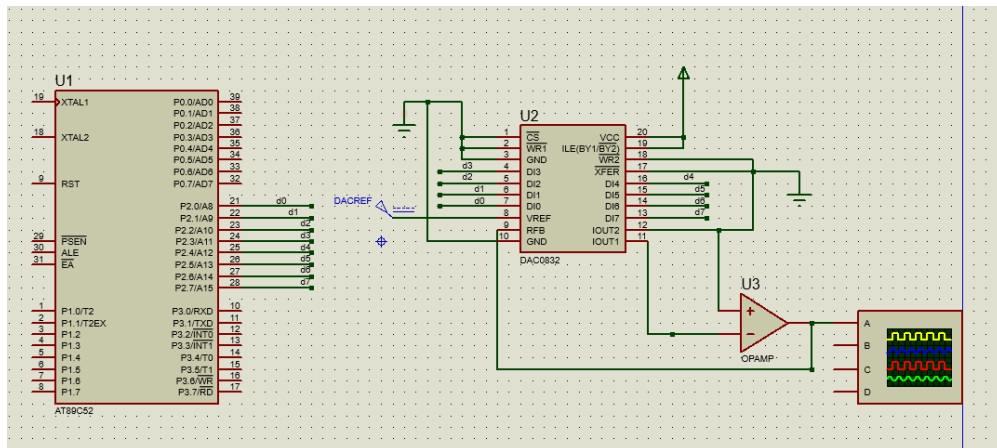


图 2 电路结构

3 流程图

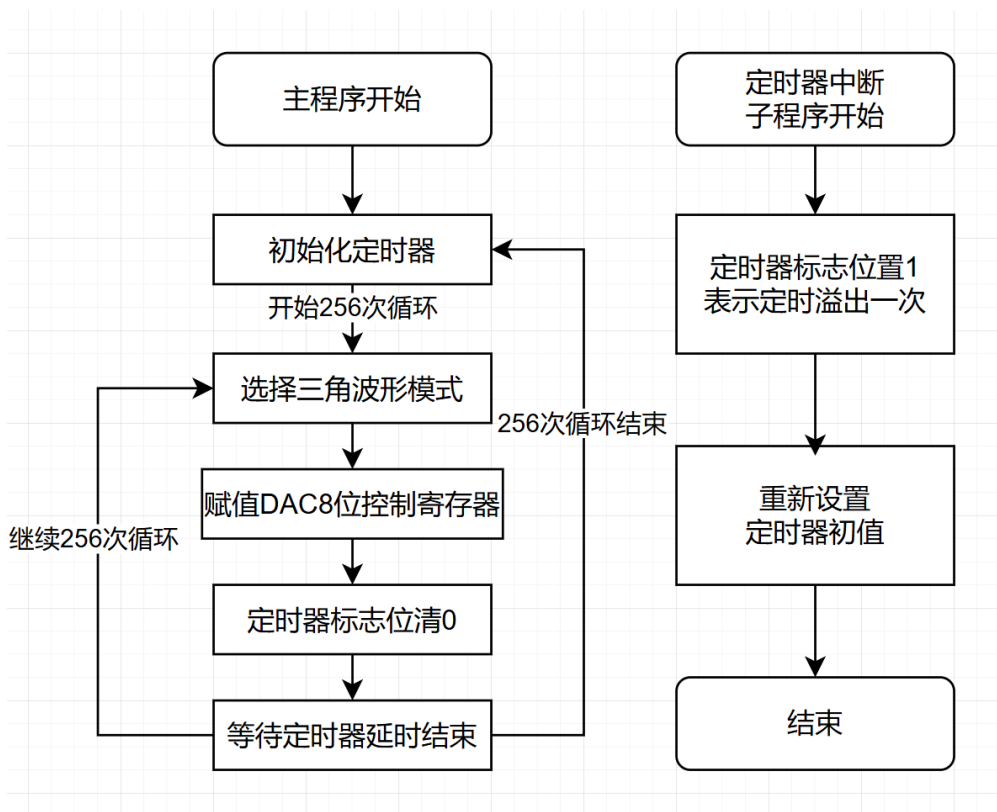


图 3 系统控制流程图

4 实验体会

通过本次三角波生成实验，我深入理解了单片机与 DAC0832 的接口及应用原理。本实验的收获和体会如下：

1. **三角波生成原理：**实验使我清晰地理解了数模转换的基本原理，特别是如何通过分段函数算法生成三角波。在代码中，通过判断当前点在波形前半段还是后半段，分别实现了上升和下降过程，这让我对数字合成波形的原理有了直观认识。
2. **波形精度控制：**学会了如何在有限的 8 位数字量下实现高质量的三角波形。通过 256 个采样点的精确控制，实现了平滑的三角波输出，观察到实际波形与理论预期十分接近，这加深了我对 DAC 分辨率与波形质量关系的理解。
3. **定时器精确控制：**掌握了如何利用定时器中断实现波形的精确周期控制。为了生成周期为 $(20+47)=67\text{ms}$ 的三角波，需要精确计算每个采样点的持续时间约为 $262\mu\text{s}$ ，这锻炼了我对硬件定时资源的精确使用能力。
4. **模块化设计思想：**通过实现不同波形生成函数(`getSawWave`、`getTriangleWave` 和自定义波形表)，学习了良好的软件设计方法。这种函数封装和模块化设计使得程序易于理解、维护和扩展，为以后实现更复杂的信号生成系统打下基础。
5. **实际测量与分析能力：**通过示波器观察和测量实际波形，验证了周期为 67ms ，幅值为 5V ，符合实验预期。这锻炼了我对电子系统进行测试、验证和分析的能力，培养了严谨的工程素养。
6. **数模混合系统理解：**本实验是典型的数模混合系统应用，通过软件生成数字信号，再经 DAC 转换为模拟信号。这种数字控制模拟输出的方法在实际工程中应用广泛，如音频处理、信号发生器等，对我理解现代混合信号系统有很大帮助。

总体而言，这次实验不仅让我掌握了三角波生成的具体技术，更重要的是培养了我将理论知识与实际应用相结合的能力。通过亲自设计程序、连接电路并测量结果，我对单片机在波形生成中的应用有了更为系统和深入的理解，为今后在嵌入式系统、信号处理等领域的学习和实践奠定了坚实基础。