摘要

本设计以AT89C51 单片机为核心器件，程序设计采用C 语言，Keil 软件编译程序，配以相关外围接口电路，实现了方波、锯齿波、正弦波、三角波、梯形波五种特定波形的产生。仿真结果表明，改变C 程序设计中递增变量大小，可以得到不同频率的波形，从而验证了设计。

**关键字：**at89c51，波形发生器，频率可调

目录

[摘要 I](#_Toc535003462)

[目录 II](#_Toc535003463)

[1 波形发生器设计方案 1](#_Toc535003464)

[1.1 基于51单片机的波形发生器系统结构 1](#_Toc535003465)

[1.2 系统模块划分 1](#_Toc535003466)

[2 系统的硬件设计 2](#_Toc535003467)

[2.1 总述 2](#_Toc535003468)

[2.2 时钟电路 2](#_Toc535003469)

[2.3 按键电路 3](#_Toc535003470)

[2.4 数模转换以及放大电路 3](#_Toc535003471)

[3 系统软件设计 5](#_Toc535003472)

[3.1 主程序设计 5](#_Toc535003473)

[3.2 延时程序设计 5](#_Toc535003474)

[3.3 外部中断0子程序设计 5](#_Toc535003475)

[3.4 波形产生程序设计 6](#_Toc535003476)

[4 调试及仿真 7](#_Toc535003477)

[4.1 产生的波形截图 7](#_Toc535003478)

[4.1.1 方波波形截图： 7](#_Toc535003479)

[4.1.2 锯齿波波形截图： 8](#_Toc535003480)

[4.1.3 正弦波仿真截图 10](#_Toc535003481)

[4.1.4 三角波仿真截图 11](#_Toc535003482)

[4.1.5 梯形波仿真截图 13](#_Toc535003483)

[4.2 调试中遇见的问题 14](#_Toc535003484)

[5 设计总结 15](#_Toc535003485)

[6 参考文献 16](#_Toc535003486)

[7 附录 17](#_Toc535003487)

[7.1 完整电路原理图 17](#_Toc535003488)

[7.2 源程序 18](#_Toc535003490)

1. 波形发生器设计方案
   1. 基于51单片机的波形发生器系统结构

本设计中的波形发生器系统要求基于51单片机，因此选用以AT89C51单片机作为整个系统的控制核心，应用其强大的接口功能，构成整个波形发生器的硬件系统。使用C 语言对单片机编程可产生相应的正弦波，方波，三角波，锯齿波梯形波波形信号。在程序运行时，当接收到按键信息后，需要输出某种波形时，调用相应的中断服务子程序和波形发生程序，经电路的数/模转换器和运算放大器处理后，从信号发生器的输出端口输出即可得到要求的波形。

按键输入

单片机at89c51

数模转换器

放大器

图 1-1 波形发生器系统结构

当需要改变频率时只需要改变单片机的波形发生程序中的递增或者递减变量即可。

* 1. 系统模块划分

根据波形发生器的系统结构可以划分为五大模块，它们分别为按键电路，时钟电路，单片机，数模转换电路，运算放大器电路。

1. 系统的硬件设计
   1. 总述

前述的不同功能模块相互联系，相互协调，以单片机为核心构成一个统一的整体。其整体原理框图如图所示。

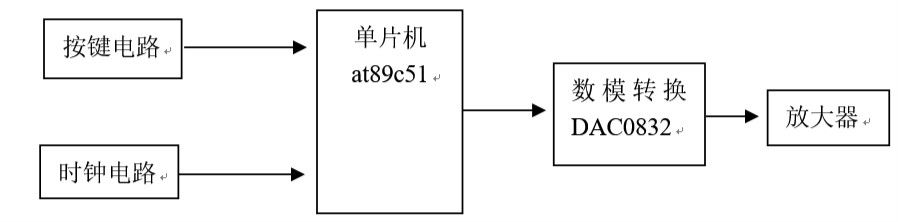


图 2-1 功能模块联系

At89C51单片机是该波形发生器的核心，它有2个定时器， 5个中断源。[1]在单片机最小系统中，单片机从P1的0、1、2、3、4、5、6口接收来自键盘的信号，并通过P0口输出控制信号，再通过D/A转换芯片最终由示波器显示输出波形，所有按键接与门后接单片机的外部中断0接口，相当于每按下某个按键就会产生一个外部中断0，而不需要时时刻刻查询每个接口的状态。

单片机引脚分配如下:

XTAL1，XTAL2：外接晶振，产生时钟信号；

P0 口：输出信号，需外接上拉电阻；

P1的0、1、2、3、4、5、6依次接5个波形选择按键和频率+、-减按键；

* 1. 时钟电路

单片机的时钟信号通常用两种电路形式得到：内部振荡和外部振荡方式。在引脚XTAL1 和XTAL2 外接晶体振荡器，构成内部振荡方式。由于单片机内部有一个高增益的反相放大器，当外接晶振后，就构成了自激振荡，并产生振动时钟脉冲。晶振通常选用6MHZ、12MHZ、24MHZ。本设计选择12MHZ晶振分别接引脚XTAL1 和XTAL2，电容C1，C2 均选择为10pF，对振荡器的频率有稳定作用。

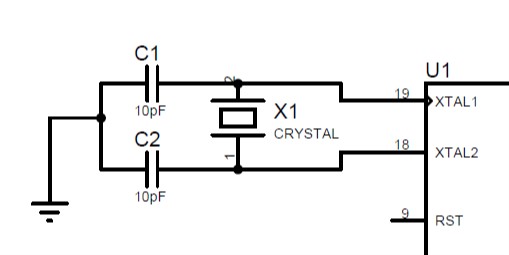


图 2-2 时钟电路

* 1. 按键电路

本次设计采用独立按键，P1^5和P1^6为频率调整按键，分别为增加和减少。P1^0 、P1^1 、P1^2 、P1^3、P1^4分别为方波、锯齿波、正弦波、三角波、梯形波的选择按键。这7个按键还要同时接入与门，然后接入单片机的外部中断0接口，当按下这7个按键中的一个后就会产生外部中断0，然后调动相关的程序执行相关操作。[3]

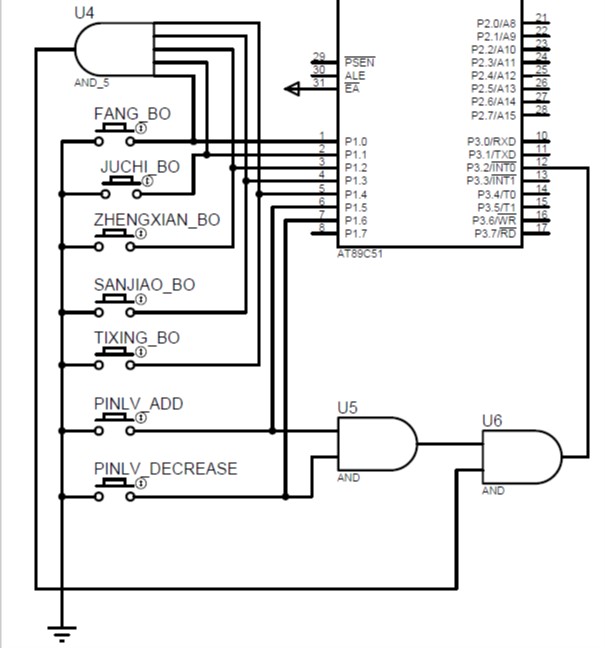


图 2-3 按键电路

* 1. 数模转换以及放大电路

由于单片机输出的是数字信号，要得到模拟信号的波形就需要对其进行数模换。本设计采用了DAC0832 数模转换器，该芯片由8 位输入锁存器、8 位DAC 寄存器、8 位D/A 转换器及转换控制电路四部分构成。由于其输出为电流输出，因此外加运算放大器LM358使之转换为电压输出。最后通过示波器显示输出的波形。

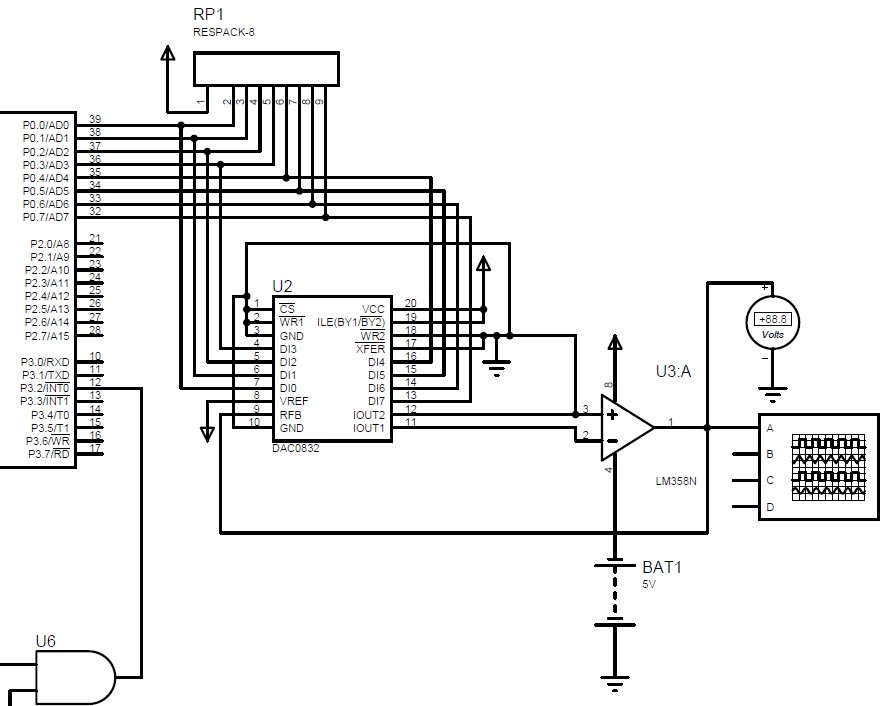


图 2-4 数模转换及放大电路

1. 系统软件设计
   1. 主程序设计

首先用Time0\_Init( )子函数初始化定时器，初始化过程：先写入控制字，再写入计数初值，最后启动定时器。初始化完毕后再将外部中断0设置为下降沿触发，然后开定时器中断和外部中断0，最后设置一个while（1）死循环，使cpu一直等待中断的产生而去执行相应操作。流程图如下：

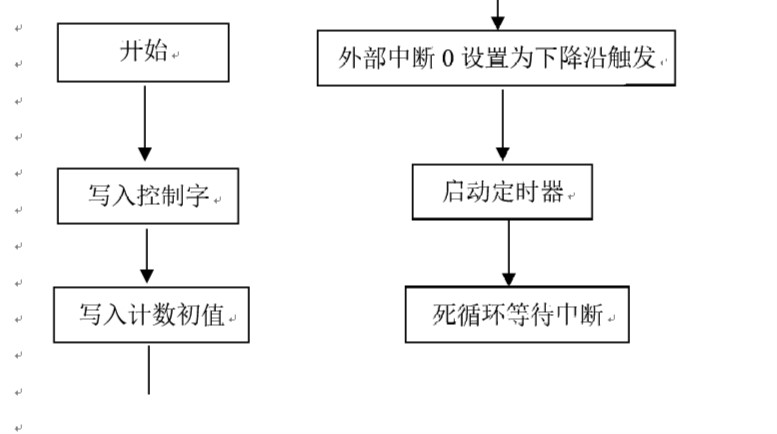


图 3-1 主程序流程图

* 1. 延时程序设计

查资料知道Cpu从1加到125大约耗时1毫秒，故此在外部套上一个while（--a！=0），那么这个程序块可以延时a毫秒，具体输入可以根据要求来定。程序块截图如下：

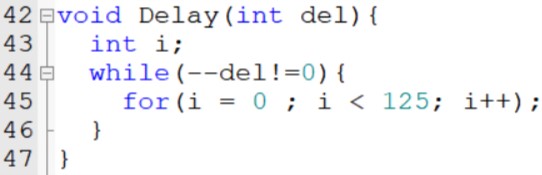


图 3-2 延时程序

* 1. 外部中断0子程序设计

此子程序用于处理波形选择按键或者频率调整按键被按下时的操作。程序中定义了unsigned int 型标志flag，当方波按键被按下后flag置1；当锯齿波按键按下，flag置2；当正弦波按键按下时flag置3；当三角波按键按下，flag置4；当梯形波按键按下，flag置5；当频率调整按键按下时，根据要求改变波形产生程序中用到的递增或递减变量即可改变频率。

* 1. 波形产生程序设计

第一个是正弦波。正弦波的产生只需要先输出一段时间的全0，再输出相同时间的全一即可。锯齿波的产生，需要先不断递增，到达峰值后突然置零，然后不断循环即可。正弦波的产生，事先准备好了数组，数组中的数根据计算拟合得来，只要不断输出数组中的数即可产生正弦波信号。三角波的产生，首先不断递增，到达峰值后再不断递减即可，递增量或者递减量根据频率要求来定。梯形波的产生类似三角波，但是需要在到达峰值后延时一段时间。

1. 调试及仿真
   1. 产生的波形截图
      1. 方波波形截图：

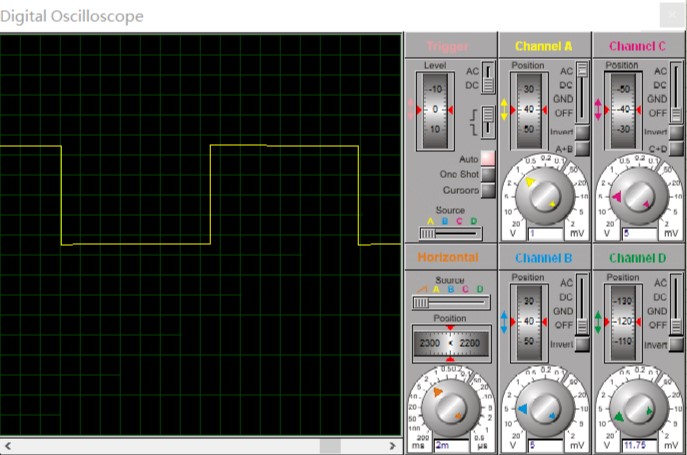


图 4-1 方波波形

变频后的方波:

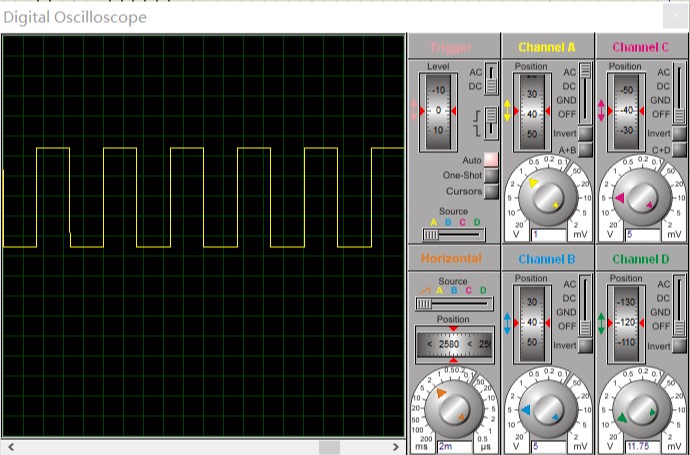


图 4-2 频率增加后的方波

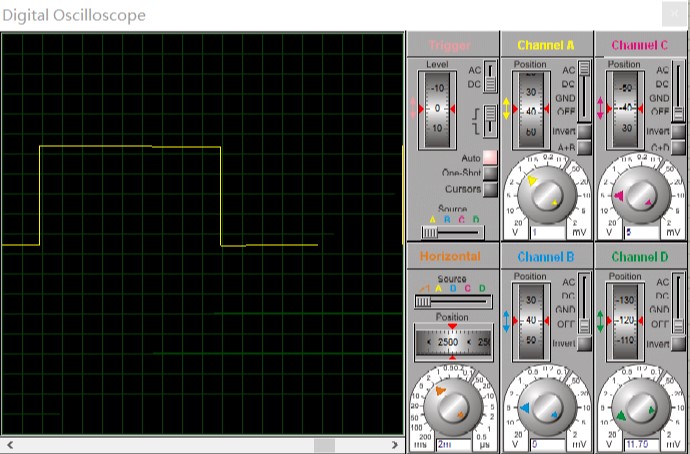


图 4-3 频率减小后的方波

* + 1. 锯齿波波形截图：

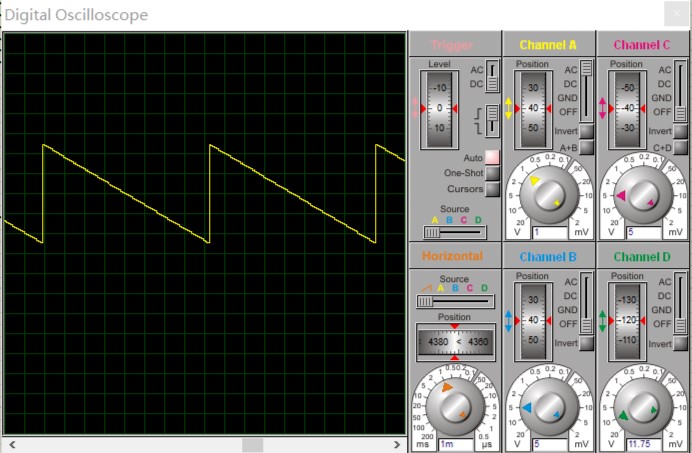


图 4-4 锯齿波波形

变频后的锯齿波：

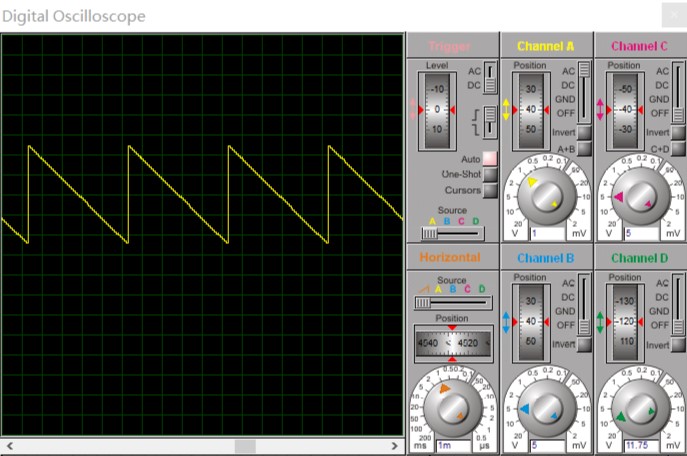


图 4-5 频率增加后的锯齿波

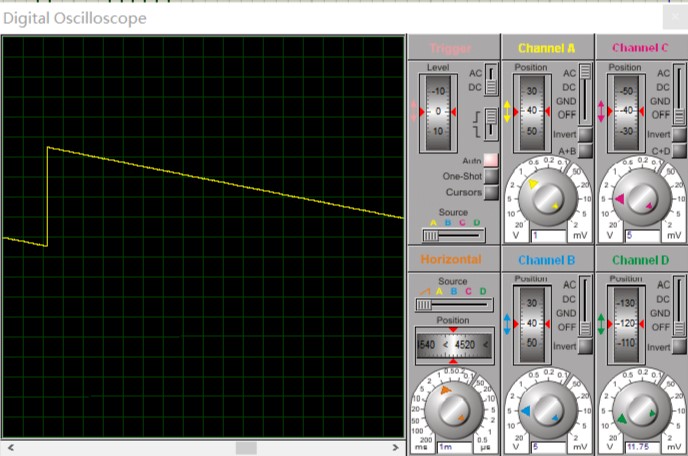


图 4-6 频率减小后的锯齿波

* + 1. 正弦波仿真截图

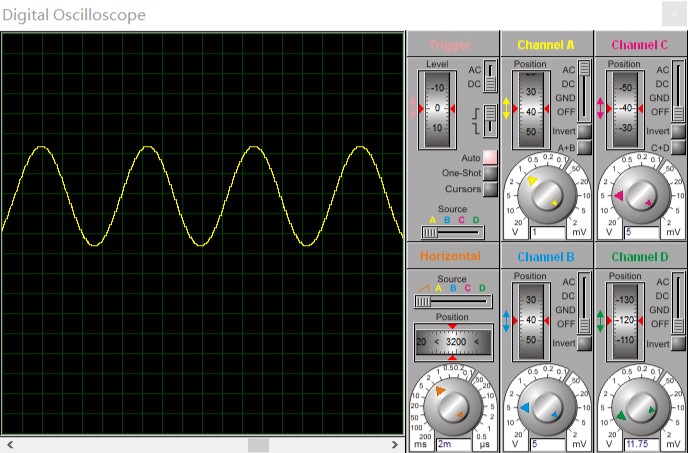


图 4-7 正弦波波形

变频后的正弦波：

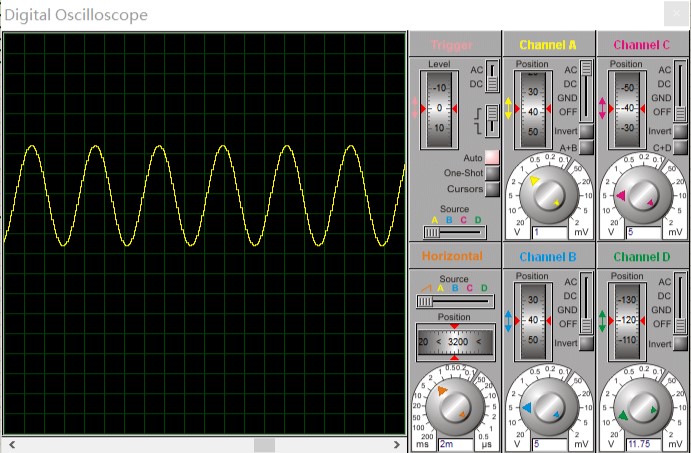


图 4-8 频率增加后的正弦波

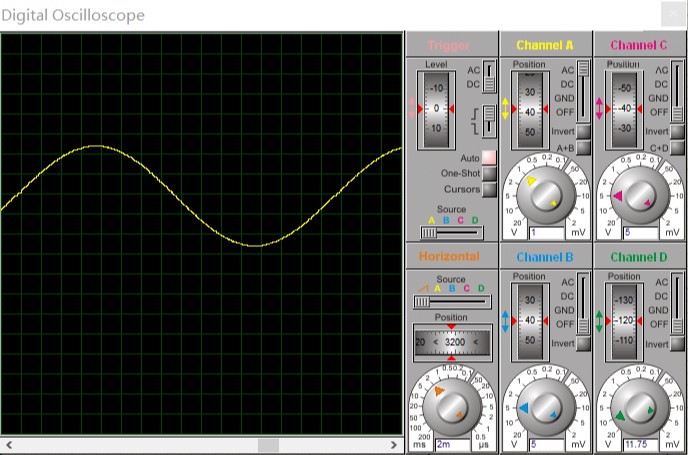


图 4-9 频率减小后的正弦波

* + 1. 三角波仿真截图

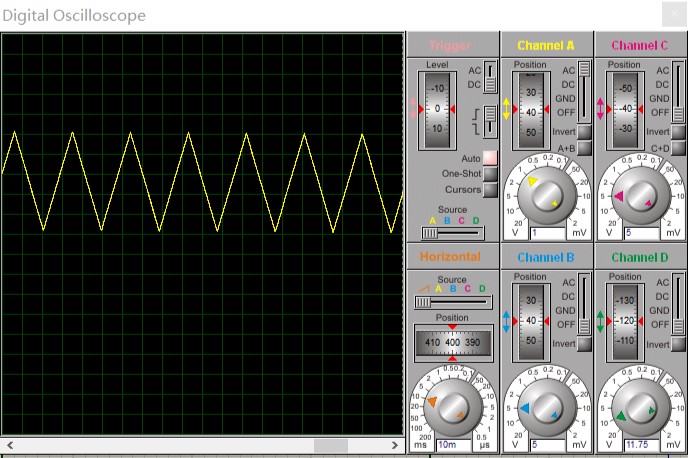


图 4-10 三角波波形

变频后的三角波：

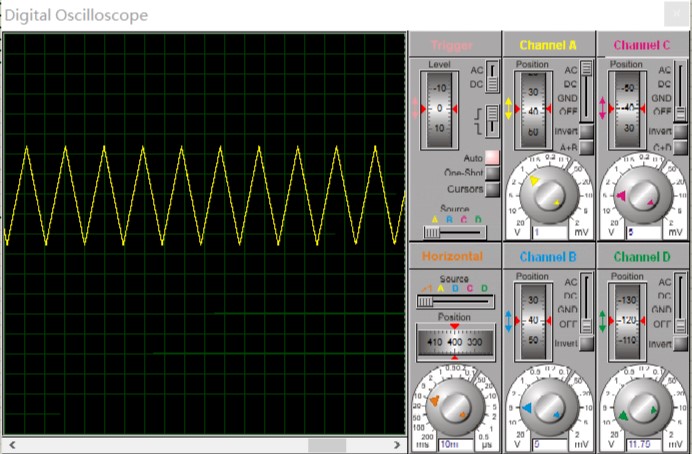


图 4-11 频率增加后的三角波

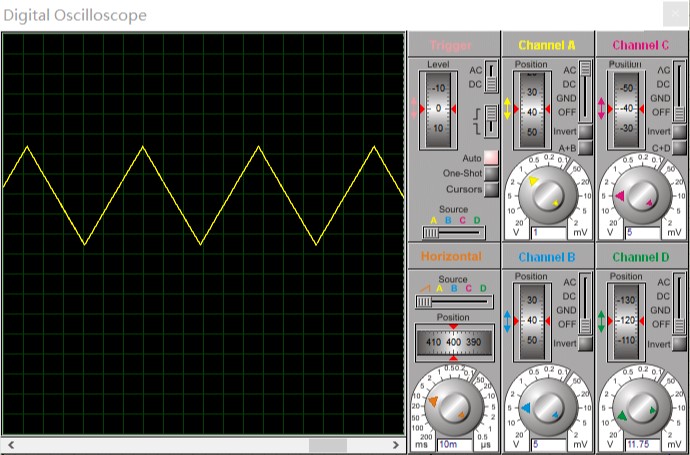


图 4-12 频率减小后的三角波

* + 1. 梯形波仿真截图

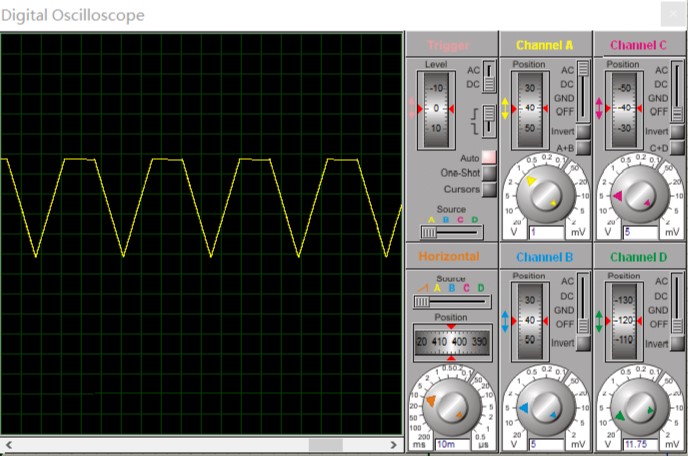


图 4-13 梯形波波形

变频后的梯形波：

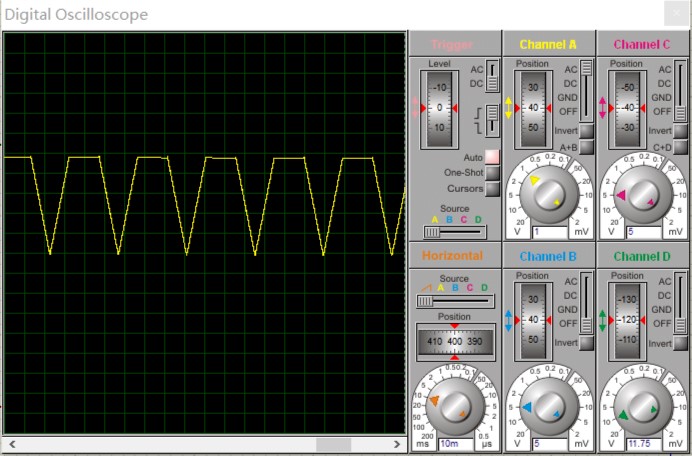


图 4-14 频率增加后的梯形波

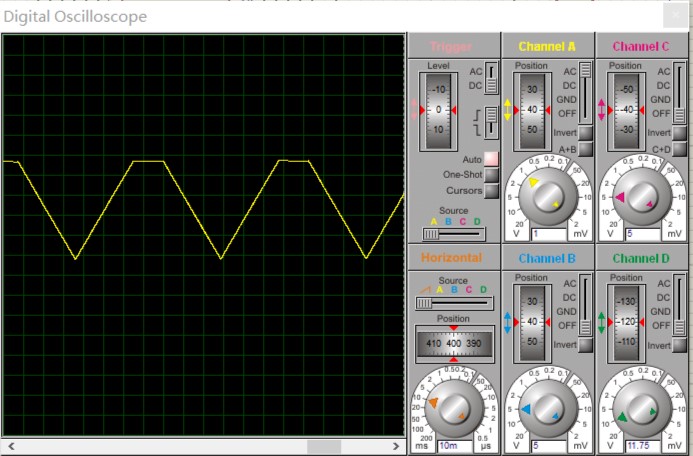


图 4-15 频率减小后的梯形波

* 1. 调试中遇见的问题

首先，因为设计的正弦波产生程序中有一个数组存在，所以最初在keil中编译后出现如下问题：

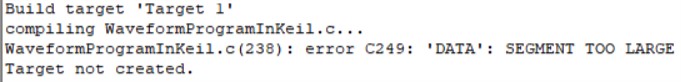


图 4-16 错误一

经过查找资料后发现是因为数组太大导致放不下的原因，因此将数组定义时前缀加了一个code，放入程序段即解决问题。

然后又发现一个问题，在百般检查后不能找到问题甚至一度觉得是keil软件的bug，最后在同学的提醒下发现是数组定义时少了分号，这提醒我在工作学习中都应该认真仔细。

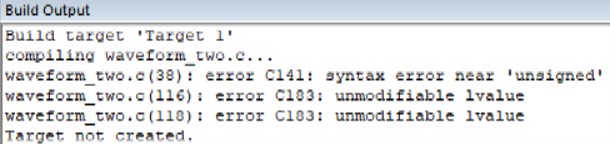


图 4-17 错误二

1. 设计总结

这次课程设计，遇到了一些问题，在自己的努力下，很好的解决了这些问题。第一次尝试把学习的51单片机知识综合起来，解决某一个问题，还学会了DAC0832的使用。这些东西不仅仅只是知识方面的，也有能力方面的——提高了自己查阅各种资料，分析和解决问题的能力以及一种实事求是的精神。

这份设计的优点在于：硬件设计和编程不复杂，能使输出频率有较好的稳定性，充分体现了模块化设计的要求，而且这些芯片及器件均为通用器件。

这份设计的缺点在于：1.频率调整范围较窄，调频率的时候不能过多按下调频按钮，否则因为内部参数的问题程序将会崩溃。 2.频率不可见，当前频率和调频后的频率都不可见，只能根据波形图得出一个大致的结论，如果可以，可以后续加入显示屏实时显示频率值。

1. 参考文献
2. 余发山.微机原理与单片机接口技术.北京.煤炭工业出版社.2013
3. 谭浩强.C语言程序设计.北京：清华大学出版社，1999
4. 陈辉.基于AT89C51单片机波形发生器的Proteus设计.≪自动化与仪器仪

表≫.2012第3期

1. 附录
   1. 完整电路原理图

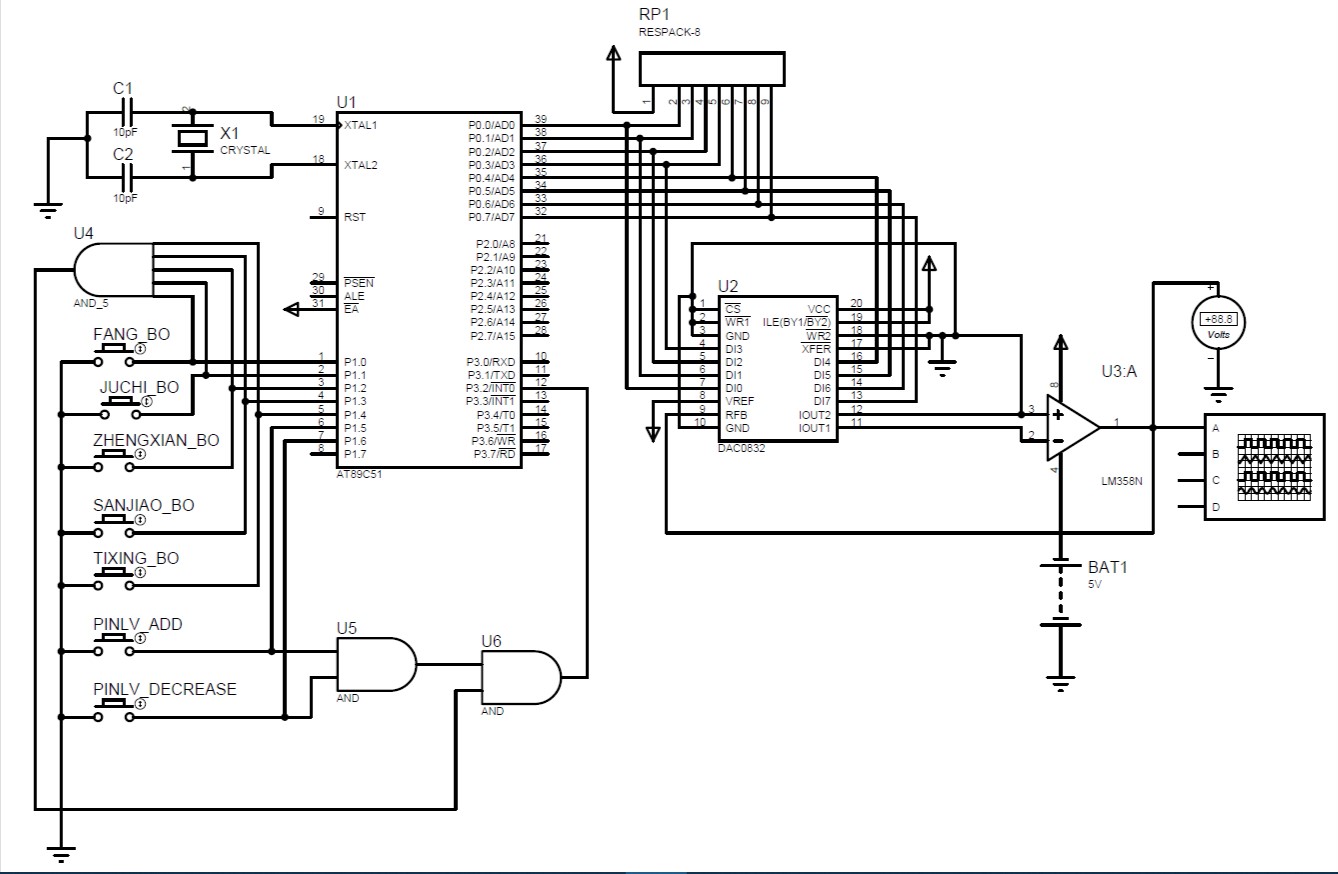


图 7-1 总电路图

* 1. 源程序

#include<reg52.h>

#define uchar unsigned char

#define uint unsigned int

uint flag = 1;//1.方波 2.锯齿波 3.正弦波 4.三角波 5.梯形波

uint juchif = 0;//波形产生用到的变量

uint sanjiaof = 0;

uint tixingf = 0;

uint f = 0;//波形产生时是否到达最大值标识

uint fangbo\_delay\_halfT = 10;//方波的半个周期时间，单位为ms，增大时周期增大频率减小

uint juchi\_delay\_add = 10;//锯齿波递增时的增量，默认为1，增大时周期减小频率增大

uint zhengxian\_delay\_add = 10;//正弦波数组指针增加时的增量，增大时频率增大

uint sanjiao\_delay\_addorde = 1;//三角波增量或者递减量，增大时周期减小频率增大

uint tixing\_delay\_f\_addorde = 1;//梯形波增量或者减少量，增大时频率增大

uint tixing\_delay\_f\_peak;//梯形波到达顶峰时的延时时间，增大时频率减小

sbit fangbo = P1^0;//端口分配

sbit juchi = P1^1;

sbit zhengxian = P1^2;

sbit sanjiao = P1^3;

sbit tixing = P1^4;

sbit p\_add = P1^5;//频率增加

sbit p\_decrease = P1^6;//频率减少

uchar code sin1[256]={//正弦波函数数组，放在程序存储器

0x80,0x83,0x86,0x89,0x8D,0x90,0x93,0x96,0x99,0x9C,0x9F,0xA2,0xA5,0xA8,0xAB,0xAE,

0xB1,0xB4,0xB7,0xBA,0xBC,0xBF,0xC2,0xC5,0xC7,0xCA,0xCC,0xCF,0xD1,0xD4,0xD6,0xD8,

0xDA,0xDD,0xDF,0xE1,0xE3,0xE5,0xE7,0xE9,0xEA,0xEC,0xEE,0xEF,0xF1,0xF2,0xF4,0xF5,

0xF6,0xF7,0xF8,0xF9,0xFA,0xFB,0xFC,0xFD,0xFD,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,

0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0xFD,0xFD,0xFC,0xFB,0xFA,0xF9,0xF8,0xF7,0xF6,

0xF5,0xF4,0xF2,0xF1,0xEF,0xEE,0xEC,0xEA,0xE9,0xE7,0xE5,0xE3,0xE1,0xDE,0xDD,0xDA,

0xD8,0xD6,0xD4,0xD1,0xCF,0xCC,0xCA,0xC7,0xC5,0xC2,0xBF,0xBC,0xBA,0xB7,0xB4,0xB1,

0xAE,0xAB,0xA8,0xA5,0xA2,0x9F,0x9C,0x99,0x96,0x93,0x90,0x8D,0x89,0x86,0x83,0x80,

0x80,0x7C,0x79,0x78,0x72,0x6F,0x6C,0x69,0x66,0x63,0x60,0x5D,0x5A,0x57,0x55,0x51,

0x4E,0x4C,0x48,0x45,0x43,0x40,0x3D,0x3A,0x38,0x35,0x33,0x30,0x2E,0x2B,0x29,0x27,

0x25,0x22,0x20,0x1E,0x1C,0x1A,0x18,0x16,0x15,0x13,0x11,0x10,0x0E,0x0D,0x0B,0x0A,

0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x02,0x01,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x02,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,

0x0A,0x0B,0x0D,0x0E,0x10,0x11,0x13,0x15,0x16,0x18,0x1A,0x1C,0x1E,0x20,0x22,0x25,

0x27,0x29,0x2B,0x2E,0x30,0x33,0x35,0x38,0x3A,0x3D,0x40,0x43,0x45,0x48,0x4C,0x4E,

0x51,0x55,0x57,0x5A,0x5D,0x60,0x63,0x66,0x69,0x6C,0x6F,0x72,0x76,0x79,0x7C,0x80};

uchar \*p = sin1;//指针，指向数组头

uchar \*p1 = sin1+255;//指向数组尾

//延时函数，del ms

void Delay(int del){

int i;

while(--del!=0){//晶振为12MHZ 延时 del ms

for(i = 0 ; i < 125; i++);//i加到125，耗时1ms

}

}

void Time0\_Init( ){//定时器初始化,12MHZ晶振,定时50ms

TMOD = 0x01;//0000 0001

TH0 = 0x3C;//装入计数初值3CB0H 15536

TL0 = 0xB0;

TR0 = 1;//启动

}

void Outside\_Int(void) interrupt 0{//外部中断0

if(fangbo == 0) flag = 1;

if(juchi == 0) flag = 2;

if(zhengxian == 0) flag = 3;

if(sanjiao == 0) flag = 4;

if(tixing == 0) flag = 5;

if(p\_add == 0){//增加频率

if(flag == 1){//方波

fangbo\_delay\_halfT = fangbo\_delay\_halfT-0.5;

}

if(flag == 2){//锯齿波

juchi\_delay\_add = juchi\_delay\_add+1;

}

if(flag == 3){//正弦波

zhengxian\_delay\_add = zhengxian\_delay\_add+1;

}

if(flag == 4){//三角波

sanjiao\_delay\_addorde = sanjiao\_delay\_addorde+1;

}

if(flag == 5){//梯形波

tixing\_delay\_f\_addorde = tixing\_delay\_f\_addorde+1;

}

}

if(p\_decrease == 0){//减小频率

if(flag == 1){//方波

fangbo\_delay\_halfT = fangbo\_delay\_halfT+1;

}

if(flag == 2){//锯齿波

juchi\_delay\_add = juchi\_delay\_add-1;

}

if(flag == 3){//正弦波

zhengxian\_delay\_add = zhengxian\_delay\_add-1;

}

if(flag == 4){//三角波

sanjiao\_delay\_addorde = sanjiao\_delay\_addorde-1;

}

if(flag == 5){//梯形波

tixing\_delay\_f\_addorde = tixing\_delay\_f\_addorde-1;

}

}

}

void Time0\_Int( ) interrupt 1{//定时器中断

TH0 = 0xff;//改变了计数初值，65504 定时 8 us 每这么多时间输出一个数字量或者延时相应时间

TL0 = 0xf8;

if(flag == 1){//方波

P0=0x01;

Delay(fangbo\_delay\_halfT);

P0 = 0xfe;

Delay(fangbo\_delay\_halfT);

}

if(flag == 2){//锯齿波

P0 = juchif;

juchif = juchif + juchi\_delay\_add;

if(juchif >= 0xfe) juchif = 0x00;

}

if(flag == 3){//正弦波

P0 = \*p;

p = p + zhengxian\_delay\_add;

if(p>= p1)

p = sin1;

}

if(flag == 4){//三角波

P0=sanjiaof;

if( f == 0) sanjiaof = sanjiaof + sanjiao\_delay\_addorde;

if(sanjiaof >= 0xfe) f = 1;

if( f == 1) sanjiaof = sanjiaof - sanjiao\_delay\_addorde;

if(sanjiaof <= 0x01) f = 0;

}

if(flag == 5){//梯形波

P0=tixingf;

if(f == 0) tixingf = tixingf + tixing\_delay\_f\_addorde;

if(tixingf >= 0xfe){

f = 1;

}

if( f == 1) tixingf = tixingf - tixing\_delay\_f\_addorde;

if(tixingf <= 0x01){

f = 0;

Delay(tixing\_delay\_f\_peak);

}

}

}

/\*主函数\*/

void main(){

Time0\_Init( );

IT0 = 1;//外部中断0下降沿触发

IE = 0x87;//中断允许寄存器1000 0111 开定时器中断和外部0

tixing\_delay\_f\_peak=10\*tixing\_delay\_f\_addorde;

while(1);

}