**传感器与控制实验报告**

**姓名：吴茜眉**

**学院：计算机科学与技术学院**

**班级：2016级物联网工程14班**

**学号：21161410**

**实验六 超声波测距**

1. **实验内容**

编写C51程序，使用超声波测量实验板距离障碍物的距离，将结果（以厘米计）显示到数码管上。测量距离在30cm~200cm之间。

1. **电路原理**

在该电路中，单片机与超声波接收器和超声波发生器以及数码管相连。利用单片机的P1.0发出若干40kHz的方波脉冲，经过放大后输出到超声波发生器产生超声波。发射出去的超声波遇到障碍反射回来，进入实验板的超声波接受器，内部电路将接受到的回波信号经过放大、过滤等，最后输出到锁相电路中，在INT0管脚出现下降沿会触发外部中断。我们可以使用内部定时器来进行计时，测量出从信号发出到产生中断之间的时间，通过计算得到距离，再显示到数码管上。由于实验一已经详细讲述过数码管部分，这里不再赘述。

1. **功能完成分析**

为了完成实验内容，需要完成三个子功能，分别是发出超声波、测量距离以及显示。

对于发出超声波，需要利用P1.0发出若干40KHz的方波脉冲，也就是P1.0每25us需要取反，在实验代码中将若干认为是20个。代码中采用定时器0的方式二来计时25us。

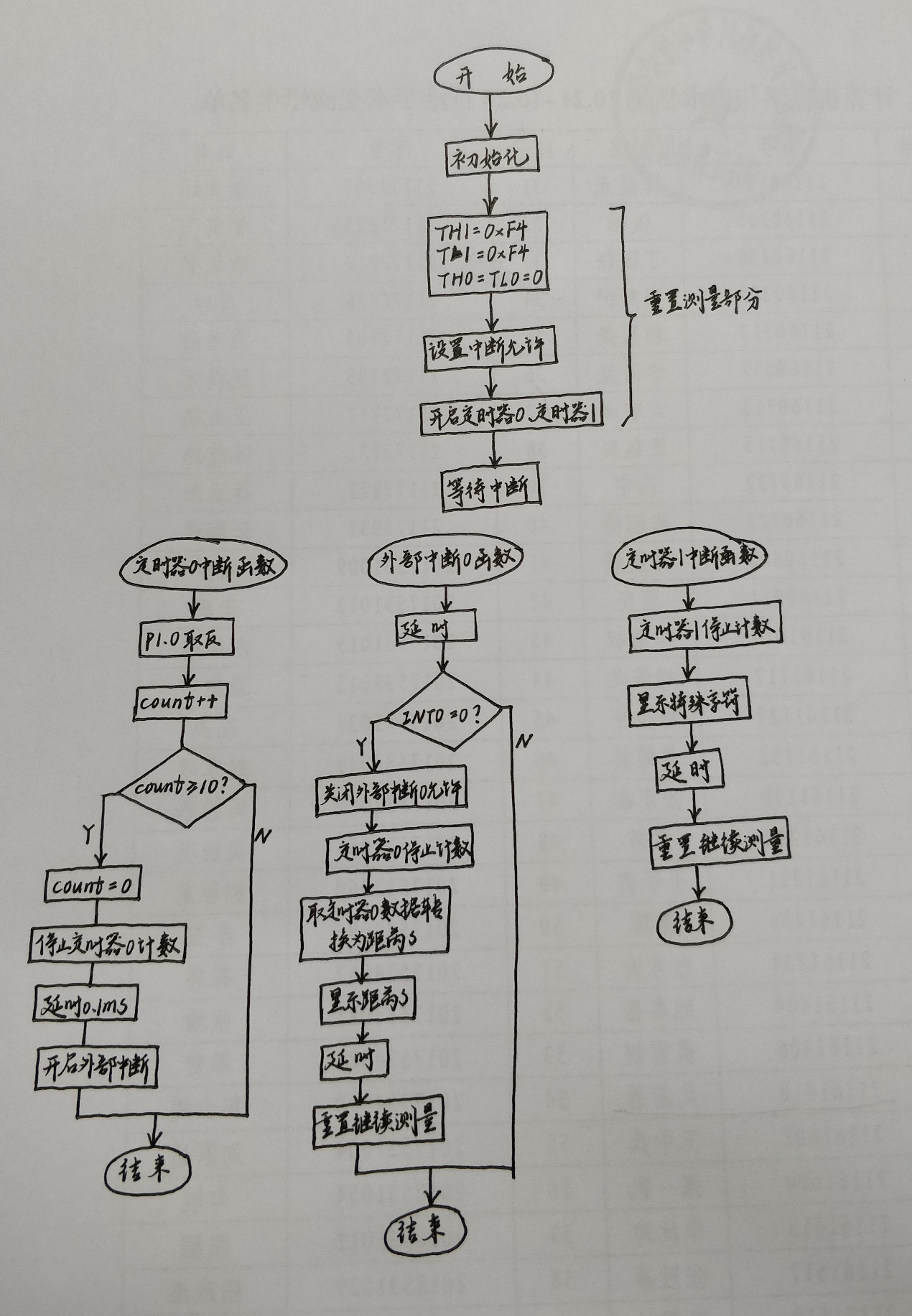
对于测量距离，需要用一个定时器测量从第一个方波发出到第一个方波到达这一过程中的时间，即外部中断一产生就需要停止计数，并对此时定时器的计数进行处理，将时间转换为距离。在代码中采用定时器1计数，定时器加1相当于1us，最后利用s = 340 \* t / 2来获得实际距离。

对于显示，可以直接调用实验一关于数码管的代码，这里不再赘述。

在本实验中，还有一些需要注意的问题。在利用定时器0产生方波时，需要关闭其他中断，因为超声波会干扰中断。因此当方波产生结束后延时一段时间再打开外部中断和定时器1中断。如果经过很长时间，仍然没有中断触发，可以认为超声波没有返回，本次测量无效。在数码管上显示一个特别的数字或符号。在代码中，定时器1的溢出是检验测量无效的关键时间点。每次测量完成后，延时一段时间，再继续下一次测量。

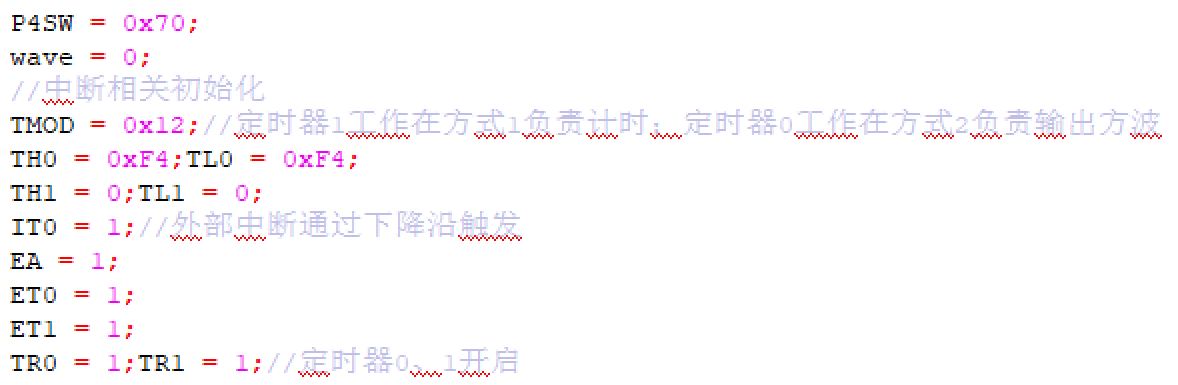
因此，在本实验中，实际有三个中断函数——定时器0中断函数、外部中断函数、定时器1中断函数。定时器0中断函数负责产生方波并对方波数进行计数，且在方波输出完成后延时一段时间打开其他中断。外部中断函数负责处理数据。定时器1中断函数负责在数码管上显示特殊字符表示此次测量无效。

1. **程序框图**

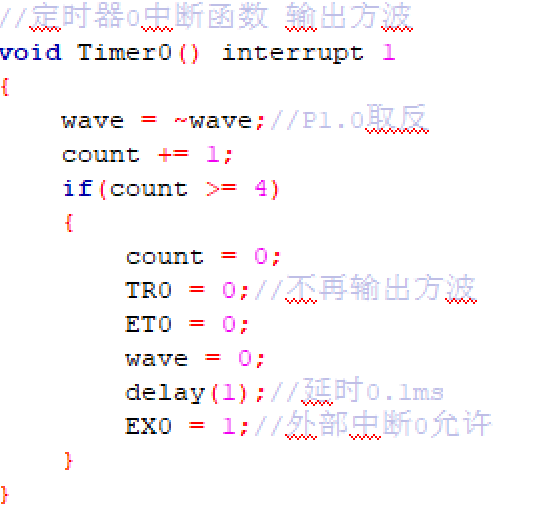
****

1. **实现代码**

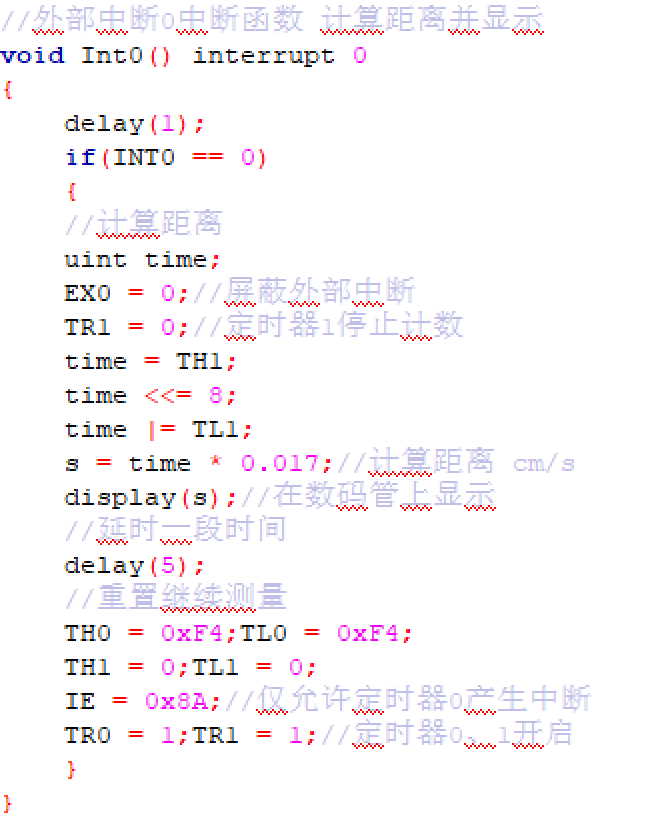
初始化、定时器赋值、中断允许、开启定时器部分：



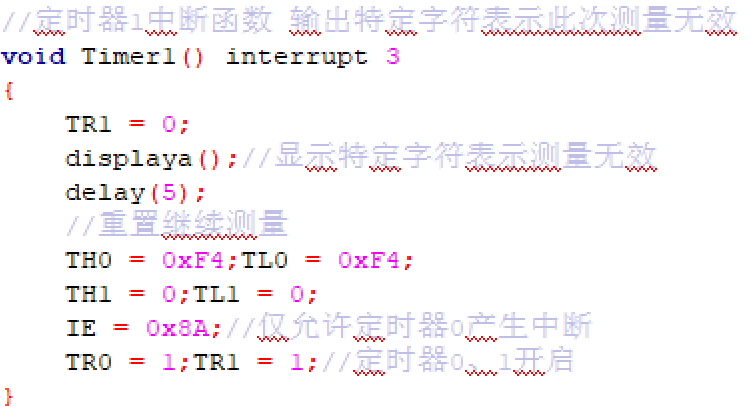
定时器0中断函数：



外部中断0中断函数：



定时器1中断函数：



1. **思考题**
   1. 当时间测量定时器溢出时，能够测量的最大距离是多少？

答：时间测量定时器的最大值为65535，能够计数的时间为65536us，最大距离s = 340m/s \* 65536us / 2 = 1114.1120cm。

* 1. 怎样通过编程实现更远距离的测量？

答：可以采用两个定时器来进行距离的测量，第一个定时器溢出会使得第二个定时器加一，这样相当于把16位的定时器扩展到32位，可以测得的最大距离扩大到s = 340m/s \* 2^32us / 2。

* 1. 不仅最大距离存在限制，设备能够测量的最小距离也有限制。讨论这个限制的来源，计算该最小距离，并讨论如何减少。

答：这个限制的来源同样是超声波对外部中断响应产生影响，单片机在产生超声波时，会先屏蔽外部中断，产生完毕之后延时0.1ms后才可开启外部中断。假设超声波产生时间忽略不计，也得是0.1ms后才可接收超声波，即进行距离的测量并显示，因此最短距离s = 340m/s \* 1 \* 10^(-4)s / 2 = 1.7cm。当然这也是忽略超声波产生后的理论值，实际值应该略微高一点。

* 1. 怎样改进距离测量公式，以使得得到更精确的距离？

答：声速 ＝ 331．4＊sqrt（1＋T／273）．sqrt表示开根号，T表示温度．距离 ＝ 0．5＊C＊t．C为声速，t为发射和接收的时间差。

1. **遇到过的问题及解决**
   1. 最开始测量出来的效果固定不变，后经过与同学讨论发现进入外部中断0函数时，需对INT0进行判断，若为0才可进行下面的操作，为1则退出中断函数。
   2. 实验最后的效果是测出较为合理的距离值后会跳回多个小于20cm的值，测量效果非常不稳定。考虑到中断函数中不宜设置太多功能，将功能放到主函数仍是同样的效果。若将方波数减少，仍然没有很大改进。考虑到延时效果也会对实验产生影响，将一般的软件延时改为\_nop\_延时后仍没有很大起色。
2. **实验最终代码**

#include<reg52.h>

#include<intrins.h>

typedef unsigned int uint;

typedef unsigned char uchar;

sfr P4SW = 0xBB;

sfr P4 = 0xC0;

sbit wave = P1^0;

sbit CLK = P4^4;

sbit DAT = P4^5;

uint count = 0;//记录方波输出个数

uint s = 0;

uint code tab[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90, 0x88};//数码管段码

uint a = 312;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*数码管相关函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//显示一个数字

void dispnum(uint num)

{

int i;

uint dm = tab[num];//段码存入dm

uint c;

for(i = 0;i < 8;i ++)

{

CLK = 0;

c = dm & 0x80;

if (c == 0) DAT = 0;

else DAT = 1;

CLK = 1;

dm <<= 1;//dm左移一位

}

}

//在数码管上显示特定字符

void displaya()

{

dispnum(10);

dispnum(10);

dispnum(10);

}

//在数码管上显示数字

void display(uint co)

{

dispnum(co%10);//显示个位

co = co / 10;

dispnum(co%10);//显示十位

dispnum(co/10);//显示百位

}

//延时函数

void delay(uint x)

{

int n = 100;

while(x--)

{

while(n--)

\_nop\_();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

P4SW = 0x70;

wave = 0;

//中断相关初始化

TMOD = 0x12;//定时器1工作在方式1负责计时；定时器0工作在方式2负责输出方波

TH0 = 0xF4;TL0 = 0xF4;

TH1 = 0;TL1 = 0;

IT0 = 1;//外部中断通过下降沿触发

EA = 1;

ET0 = 1;

ET1 = 1;

TR0 = 1;TR1 = 1;//定时器0、1开启

while(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*中断函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//定时器0中断函数 输出方波

void Timer0() interrupt 1

{

wave = ~wave;//P1.0取反

count += 1;

if(count >= 4)

{

count = 0;

TR0 = 0;//不再输出方波

ET0 = 0;

wave = 0;

delay(1);//延时0.1ms

EX0 = 1;//外部中断0允许

}

}

//外部中断0中断函数 计算距离并显示

void Int0() interrupt 0

{

delay(1);

if(INT0 == 0)

{

//计算距离

uint time;

EX0 = 0;//屏蔽外部中断

TR1 = 0;//定时器1停止计数

time = TH1;

time <<= 8;

time |= TL1;

s = time \* 0.017;//计算距离 cm/s

display(s);//在数码管上显示

//延时一段时间

delay(5);

//重置继续测量

TH0 = 0xF4;TL0 = 0xF4;

TH1 = 0;TL1 = 0;

IE = 0x8A;//仅允许定时器0产生中断

TR0 = 1;TR1 = 1;//定时器0、1开启

}

}

//定时器1中断函数 输出特定字符表示此次测量无效

void Timer1() interrupt 3

{

TR1 = 0;

displaya();//显示特定字符表示测量无效

delay(5);

//重置继续测量

TH0 = 0xF4;TL0 = 0xF4;

TH1 = 0;TL1 = 0;

IE = 0x8A;//仅允许定时器0产生中断

TR0 = 1;TR1 = 1;//定时器0、1开启

}