传感器与控制实验报告2018

班级:2016级14班

学号:21161402

姓名:张资卿

**实验六 超声波测距**

1. **实验目的和要求**

进一步掌握使用C51语言编写程序的方法。掌握方波的输出方法，掌握使用定时器测量时间的方法。掌握超声波距离测量的原理与方法。

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。

**三、实验步骤**

**1.** 根据实验原理，本实验的基本思路如下：

① 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。

② 发射出去的超声波遇到障碍反射回来，又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，触发外部中断。

③ 使用内部定时器来进行计时，计算出从信号发出到产生中断之间的时间，通过计算得到距离，并显示到数码管上。

④ 上述过程不断循环重复。

**2.** 产生40kHz的方波，需要在每半个周期将P1.0的输出进行翻转。这个定时需要精确实现，可以使用一个内部定时器来控制（方式2更加精确）。如果使用程序延时的话，需要查阅相关手册，并结合使用的单片机型号来确认指令的运行时间。只需要输出若干个（5-10个）脉冲即可。

**3.** 在方波输出结束此之前不能开启外部中断允许，这是因为输出的超声波会同时反馈到输入，干扰中断。为保险起见，应该在方波输出结束后稍微延迟一段时间（0.1ms即可，不用精确控制）再开启。

**4.** 使用另外一个定时器完成时间的测量。当开始输出时启动定时器（初值为0），在外部中断处理中停止定时器，并将定时器的值读出，就可以换算为距离，并显示到数码管上。

**5.** 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。时间测量定时器的溢出是检验测量无效的关键时间点。

**6.** 每次测量完成后，延时一段时间，继续下一次测量。

**四、实验原理**

1.振动频率超过20kHz，不能被人耳所接收的声波称作“超声波”。超声波频率高，波长短，在一定距离内沿直线传播，具有良好的束射性和方向性，在介质中传播的距离较远，因而超声波经常用于距离的测量，如测距仪等。

2.当超声波在发射后遇到障碍物反射回来之后，根据发射和接收的时间差计算出发射点到障碍物的实际距离。在不考虑温度等因素的情况下，超声波在空气中的传播速度约为340m/s，根据记录的时间差t，就可以计算出发射点距障碍物的距离(s)，即：s=340\*t/2。

3.超声波测距装置分为两部分：超声波发生器和超声波接受器。本实验中采用压电式超声波发生器。它利用压电晶体的谐振原理来进行电能和机械能之间的相互转换。超声波发生器内部有两个压电晶片和一个共振板。当压电晶片外加一定电压的脉冲信号，并且其频率等于压电晶片的固有振荡频率时，压电晶片将会发生共振，并带动共振板振动，将电能转换为机械能，产生超声波。反之，如果两电极间未外加电压，而共振板接收到超声波时，将压迫压电晶片作振动，将机械能转换为电信号，这时它就成为超声波接收器了。因此同一种装置即可作为发生器，也可以作为接收器。本实验使用的压电晶片的固有振荡频率为40kHz左右。因此我们需要一个内部定时器来进行控制输出方波。因为要产生40kHz的方波。每半个周期将P1.0的输出进行翻转，一个周期是1000/40=25微秒，半个周期取整为12微秒，由于选择常数自动重新载入的工作方式2更精准，而且也可以自动加载，因此我们将TH1和TL1分别设置为0xf4即可设置12微秒的定时。我们用count计算发送了多少脉冲，一般4个脉冲即可。

**五、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？

由于定时器1的最大范围是2的16次方=65536，最大定时范围（单位为秒）为=65536\*12/晶振频率。

因为实验单片机采用12MHz晶振，机器周期为1微秒，取20°C的声速为344m/s，则有：

65536\*12/12000000=0.065536s

S=344\*0.065536/2=11.272192m=1127.2192cm

2 怎样通过编程实现更远距离的测量？

增加定时器的定时时间。利用软件中断的方法，定时器采用预填充，在中断程序中加入定时器，波速不变的情况下，增加时间，可以实现更远距离的测量。

3 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。

在方波输出结束此之前不能开启外部中断允许，这是因为输出的超声波会同时反馈到输入，干扰中断。所以为保险起见，在方波输出结束后稍微延迟了一段时间(大约0.1ms)再开启。这就是测距器会有一个最小可测距离的原因。

4 怎样改进距离测量公式，以使得得到更精确的距离？

(1)在定时时间里减去防止输出的超声波会同时反馈到输入，干扰中断的延时时间。

(2)用计算脉冲个数的方式精确计算总时间，被测物体距离D理论公式为:D=1/2c(t-(2n-1)/2f)。

（c是超声波声速，n是设定的脉冲数，f回波信号频率）

1. **实验流程图**

**主函数流程：**

开始

数码管和超声波发送方波位初始化

设置定时器0为工作方式2，定时器1为工作方式1，进行定时器初值设置，由于定时器1用来计时间，因此TH1和TL1都设置为0x00，定时器0辅助发送方波，因此设置12微秒，由于是8位的，所以通过计算得TH0和TL0设置为0xf4。

关闭外部中断，关闭定时器1，打开定时器0，开始发送方波

N

定时器0结束方波发送标志位t0\_end是否为1？

Y

关闭定时器0并延时100微秒

打开外部中断，打开定时器1

是否有外部中断？

N

Y

Y

关闭定时器1，count=999，说明定时器1溢出

定时器1中断标志位t1是否为1

N

关闭定时器1，计算距离

显示count

1. **实验心得**

超声波测距的实验实际上并不难，但是在学习资料之后掌握了全部过程还是出现了问题，我的数码管的示数一直在二十多就停下不再变小。在问老师会不会出现超声波示数不太精准的问题时，了解到做超声波测距实验的学长学姐们测到的最小的数可以达到个位，于是在想自己是不是在计算距离的公式上出现问题，改代码的过程中发现自己的公式并不影响数算不出来太小的值，后来和同学进行讨论想出来其实是自己发送方波时本应该在方波输出结束后稍微延迟一段时间（0.1ms即可）再开启，因为输出的超声波会同时反馈到输入，干扰中断，为保险起见我们必须延时，可是我延时的时间有点长，导致多余的时间也被算进了定时器1的计时范围内，所以才会有20cm左右的偏移，因此需要将延时小一些，就可以了。

1. **实验代码**

#include <reg52.h>

#include<intrins.h>

typedef unsigned char uchar;

typedef unsigned intuint;

sfr P4=0xC0;

sfr P4SW=0xBB;

sbitsclk = P4^4; //Ê±ÖÓÏß

sbitsda = P4^5;//Êý¾ÝÏß

sbit out = P1^0;

bit t0\_end;

bit e0;

bit t1;

uint count = 0;

unsigned long int temp = 0;

uchar code tab[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};

unsigned long int result = 999;

voidinit()

{

P4=0xFF;

P4SW=0x70;

TMOD = 0x12;

TH0 = 0xf4;

TL0 = 0xf4; //¶¨Ê±Æ÷0ÉèµÄ³£Êý×Ô¶¯ÖØÐÂÔØÈëµÄ¹¤×÷·½Ê½2

TH1 = 0x00;

TL1 = 0x00; //¶¨Ê±Æ÷1ÉèµÄ¹¤×÷·½Ê½1£¬ÓÃÀ´¼ÆÊý£¬³õÊ¼ÖµÎª0

IT0 = 1;

EA = 1;

EX0 = 0; //³õÊ¼»¯µÄÊ±ºòÍâ²¿ÖÐ¶Ï±ØÐë¹Ø±Õ

ET0 = 1;//¶¨Ê±Æ÷ÖÐ¶ÏÔÊÐí£«¿ª¶¨Ê±Æ÷

ET1 = 1;

EX0 = 0;

}

voidsendbyte(ucharch){

uchar shape, c;

shape = tab[ch];

for (c = 0; c < 8; c++){

sclk = 0;//±ßÑØ´¥·¢£¬ÓÉµÍµ½¸ßÊ±Êý¾ÝÓÒÒÆÒ»Î»ÊäÈëµ½Q0£¨Ê±¼ä¿Õ£©

sda = shape & 0x80; //shape&1000 0000

sclk = 1;

shape<<= 1;

}

}

void display(uint n){ //ÏÔÊ¾Êý×Ö

sendbyte(n%10);

n /= 10;

sendbyte(n%10);

sendbyte(n/10);

}

void delay(uchari,uchar j)

{

while(i--){

while(j--);

j=255;

}

}

void e0\_int() interrupt 0 //Íâ²¿ÖÐ¶ÏÍê³É¹Ø¶¨Ê±Æ÷1 ½«e0=1ËµÃ÷¼ÆÊýÆ÷Íê³É

{

delay(1,255);

//delay1(1);

if(INT0==0){

TR1=0;//¹Ø±Õ¶¨Ê±Æ÷1

e0=1; //±êÖ¾Î»

}

}

void t0\_int() interrupt 1

{

out = ~out; //Êä³öÒ»¸öÖÜÆÚµÄ·½²¨

if(count != 0x00)

count-=1; //·¢ÉäÍêËùÓÐ·½²¨

else

t0\_end=1; //¹Ø±ÕÍâ²¿ÖÐ¶Ï

}

void t1\_int() interrupt 3 //Òç³öËµÃ÷½á¹û²»¶Ô

{

e0 = 1; //±êÖ¾Î»

t1 = 1; //ËµÃ÷È¡²»µ½ ½á¹ûÎª0

}

void main()

{

unsigned long t;

unsignedinti;

init(); //ÉèÖÃºÃ¶¨Ê±Æ÷

while(1){

t0\_end=0; //Íâ²¿ÖÐ¶ÏµÄ±êÖ¾Î»

count=4; //5µ½10¸öÂö³å

out=0;

t1=0;

t=0;

e0=0;

TH0=0xf4;

TL0=0xf4;

TH1=0x00;

TL1=0x00;

TR0=1; //¿ª¶¨Ê±Æ÷

TR1=1;

while(!t0\_end); //µÈ´ýt0\_endÔÚ·½²¨Íê³ÉÊä³öÖ®ºóÖÃÒ» £¬ËµÃ÷·½²¨Íê³É

TR0=0;

delay(1,120); //ÑÓÊ±Ò»Ð¡»áÔÙ¿ªÍâ²¿ÖÐ¶Ï

EX0=1;

while(!e0); //µÈ´ýÍâ²¿ÖÐ¶Ï

EX0=0; //¹Ø±ÕÍâ²¿ÖÐ¶Ï

if(t1){ //ËµÃ÷¶¨Ê±Òç³ö

TR1=0;

result=999;

}

else{

t=TH1;

t = t << 8;

t |= TL1;

result=t\*0.017;

}

display(result);

for(i=0;i<10;i++){

delay(255,255);//ÏÔÊ¾Ò»¶ÎÊ±¼ä

delay(255,255);

delay(255,255);

}

}

}