

一、要求

- 1、测距显示在数码管上。
- 2、复位。
- 3、校准功能，通过按键开关设置在当前测得距离基础上加或减，以调整当前测得距离的数值使其接近真实距离。

二、总体方案

本设计包括硬件和软件设计两个部分。模块划分为数据采集、按键控制、四位数码管显示、报警等子模块。电路结构可划分为：超声波传感器、蜂鸣器、单片机控制电路。就此设计的核心模块来说，单片机就是设计的中心单元，所以此系统也是单片机应用系统的一种应用。单片机应用系统也是有硬件和软件组成。硬件包括单片机、输入/输出设备、以及外围应用电路等组成的系统，软件是各种工作程序的总称。单片机应用系统的研制过程包括总体设计、硬件设计、软件设计等几个阶段。系统采用 STC89C52 单片机作为核心控制单元，当测得的距离小于设定距离时，主控芯片将测得的数值与设定值进行比较处理。然后控制蜂鸣器报警。系统总体的设计方框图如图 1 所示。

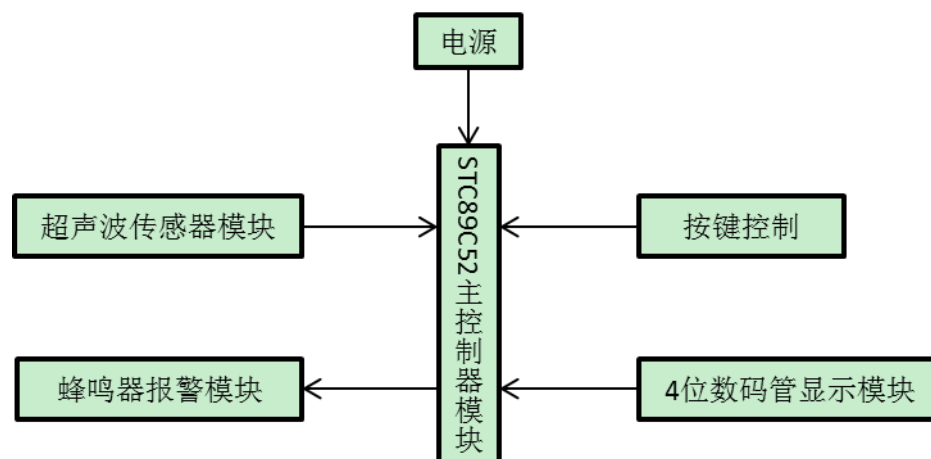
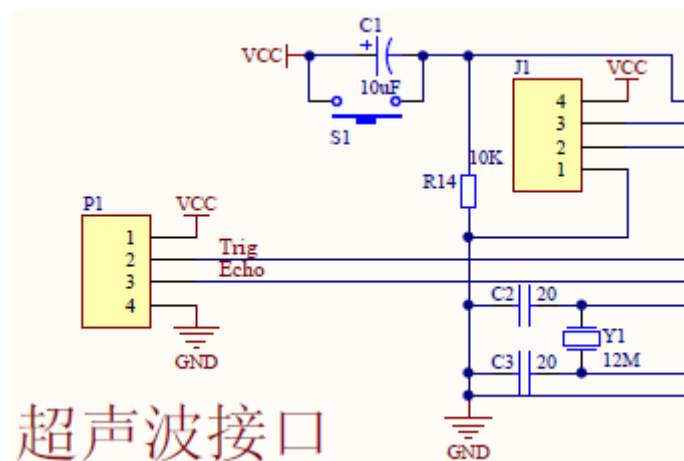


图 1 系统总体设计方框图

三、分模块电路原理及连接图

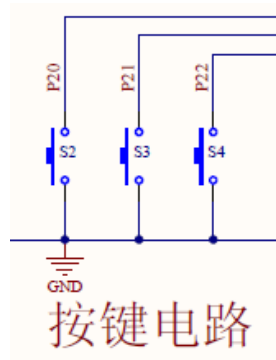
1.超声波传感器模块

超声波传感器模块用于发射超声波并接收物体反射回来的超声波，从而输出持续时间等于超声波从发射到返回的时间的高电平。



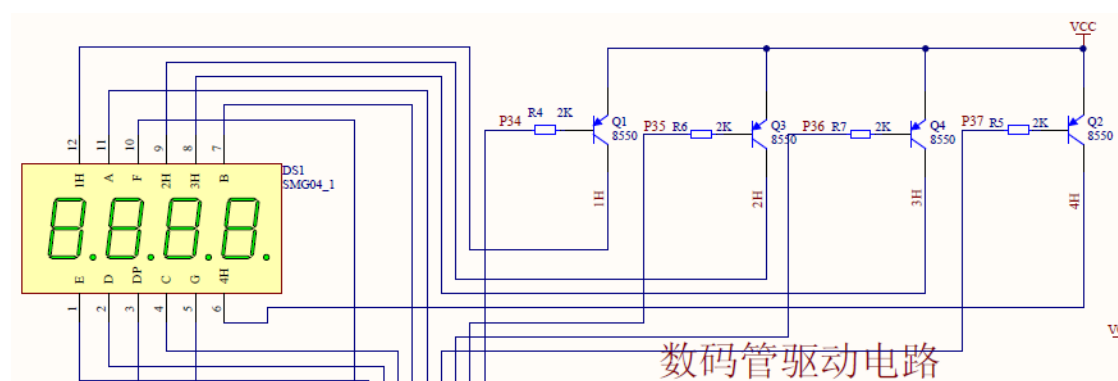
2.按键控制模块

按键控制模块有三个按键,分别为复位按键、增加距离按键和减小距离按键,分别用于复位测距电路、增加测量距离和减小测量距离。



3.四位数码管显示模块

四位数码管显示模块用于显示测量出来的距离，显示精度为 0.1cm。



4.蜂鸣器报警模块

当测量距离小于设定值时，蜂鸣器报警模块用来发声进行报警。

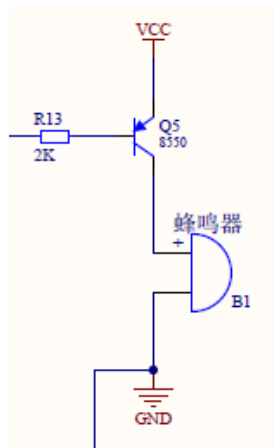


图5 蜂鸣器报警模块

四、总体电路原理连接图

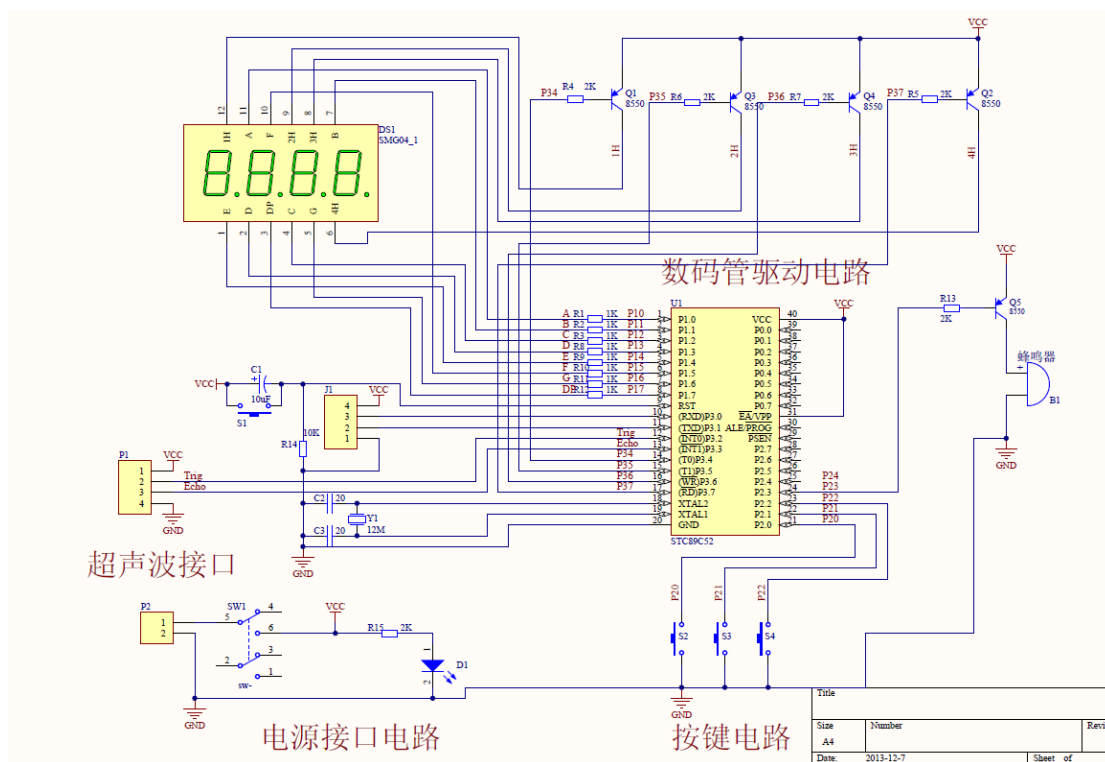


图6 总体电路原理连接图

五、超声测距原理及计算公式

- 1、超声波模块采用 I/O 触发测距，给至少 10us 的高电平信号；
- 2、模块自动发送 8 个 40kHz 的方波，自动检测是否有信号返回；
- 3、若有信号返回，则通过 I/O 输出一高电平，高电平持续的时间就是超声波从发射到返回的时间。

测试距离=(高电平时间*声速(340m/s))/2。

时序波形图如图 7 所示。

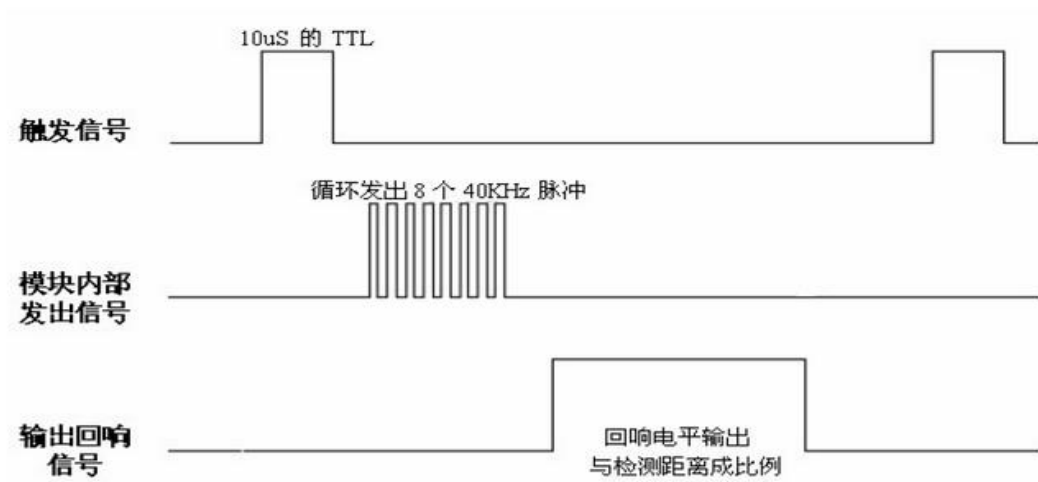


图 7 超声波模块时序图

六、程序原理图

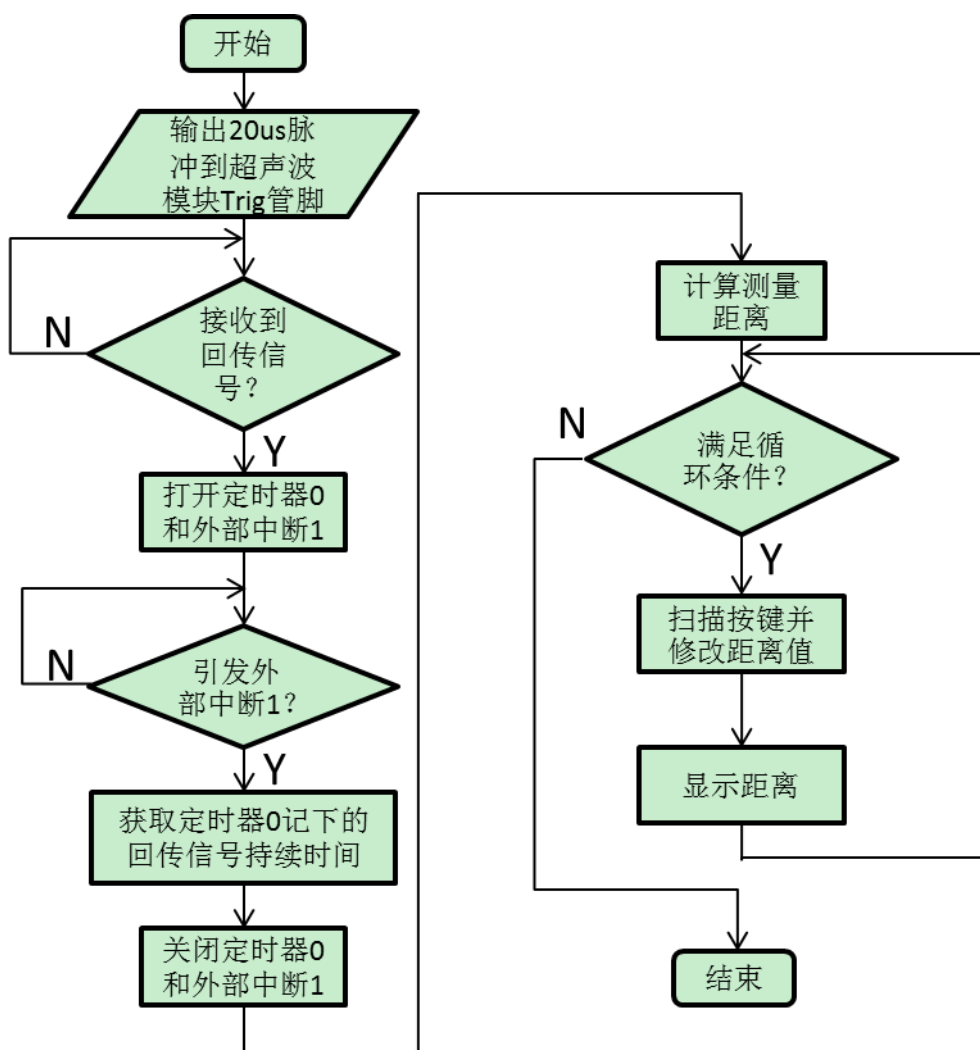


图 8 程序原理框图

七、超声模块原理分析及原理图

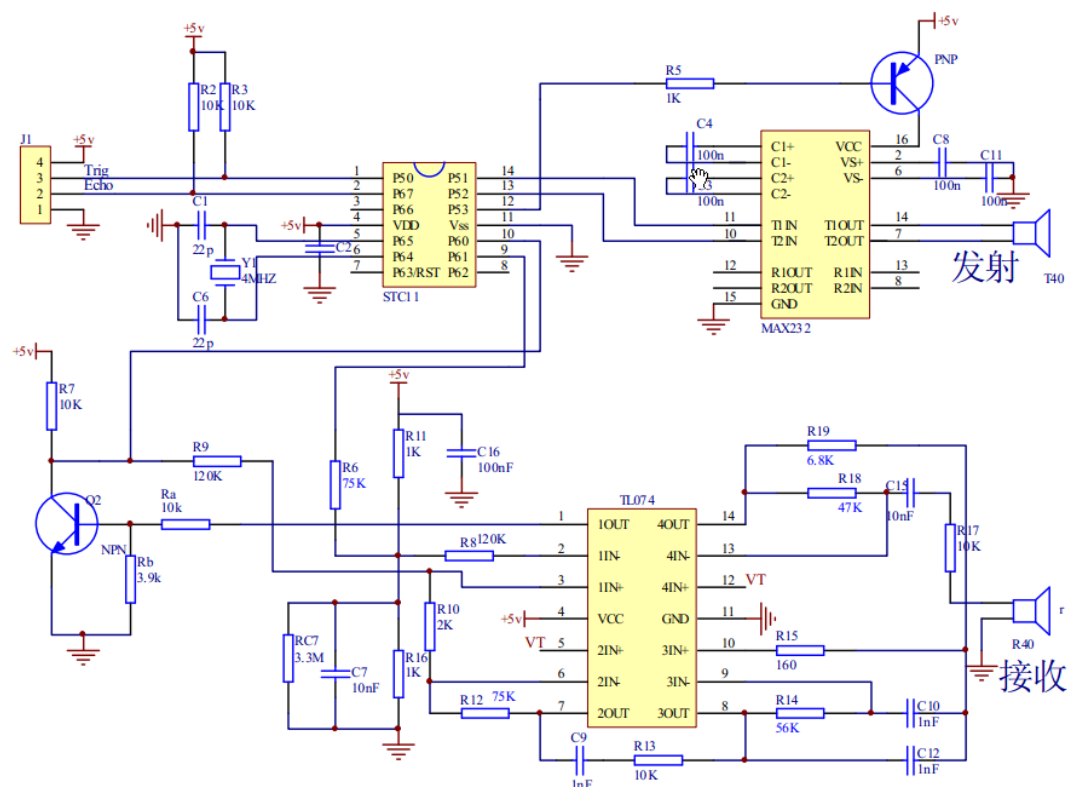


图 9 超声波模块 HC-SR04 电路原理图

八、源程序

```
/******
```

实验名称； 超声波测距实验

小组成员： 张宇 王晗丁

时 间： 2015/9/14-2015/9/25

```
*****/
```

```
#include <reg52.h>
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
sbit Trig = P3^2; //启动超声波测距的引脚
sbit Echo = P3^3; //接收超声波回传信号的引脚
sbit Qian = P3^4; //数码管最高位选信号引脚
sbit Bai = P3^5; //数码管次高位选信号引脚
sbit Shi = P3^6; //数码管次低位选信号引脚
sbit Ge = P3^7; //数码管最低位选信号引脚
sbit S2 = P2^0; //校准开关引脚
sbit S3 = P2^1; //加距离开关引脚
sbit S4 = P2^2; //减距离开关引脚
```

```

sbit Beep = P2^3;           //蜂鸣器控制引脚
int succeed_flag;           //接收到超声波回传信号与否的标志
int time;                   //回传信号持续时间
float speed = 0.340;        //声速，单位 mm/us
uint STDDISTANCE = 1000;    //校准用标准距离，1000mm
uint distance;              //存放测量好的距离
uchar timeH;                //16 位定时器高 8 位数据
uchar timeL;                //16 位定时器低 8 位数据
uchar code Table[10] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
                           //4 位共阳极数码管数字编码数组

void delayms(uint z); //延时 z ms
void delay_20us();     //延时 20us
void keyscan();         //按键扫描
void display();         //数码管动态显示距离

//按下一次复位键，进行一次测距
void main()
{
    TMOD = 0x01;         //定时器 0，工作方式 1，16 位计数
    Trig = 0;            //首先拉低脉冲输入引脚
    EA = 0;              //关总中断
    Trig = 1;            //超声波输入端送入高电平
    delay_20us();        //延时 20us
    Trig = 0;            //产生一个 20us 的脉冲
    while(Echo == 0);    //等待 Echo 回波引脚变高电平
    succeed_flag = 0;    //清测量成功标志
    EA = 1;              //开总中断
    IT1 = 1;             //外部中断 1 触发方式为下降沿触发
    EX1 = 1;             //打开外部中断 1
    TH0 = 0;             //定时器 0 清零
    TL0 = 0;             //定时器 0 清零
    TF0 = 0;             //计数溢出标志
    TR0 = 1;             //启动定时器 0
    delayms(20);         //等待测量的结果
    TR0 = 0;             //关闭定时器 0
    EX1 = 0;             //关闭外部中断 1
    if(succeed_flag == 1)
    {
        time = timeH * 256 + timeL;
        distance = time/2*speed; //毫米
    }
    if(succeed_flag == 0)
    {

```

```

        distance = 0;           //没有回波则清零
    }
    //循环检测按键并显示距离
    while(1)
    {
        keyscan();
        if(distance < 500)
            Beep = 0;           //距离小于 500mm，则蜂鸣器报警
        if(distance >= 500)
            Beep = 1;
        display();
    }
}

//外部中断 1，用做判断回波电平
void int1() interrupt 2        //外部中断 1 中断号是 2
{
    EX1=0;                     //关闭外部中断
    timeH =TH0;                 //取出定时器的值
    timeL =TL0;                 //取出定时器的值
    succeed_flag=1;            //至成功测量的标志
}

//定时器 0 中断,用做超声波测距计时
void timer0() interrupt 1      //定时器 0 中断号是 1
{
    TH0=0;
    TL0=0;
}

//延时函数
void delays(uint z)
{
    uint x,y;
    for(x = z;x > 0;x--)
        for(y = 110;y > 0;y--);
}

//延时 20us
void delay_20us()
{
    uchar a;
    for(a = 0;a < 100;a++);
}

```

```

//按键扫描
void keyscan()
{
    //加距离，一次加 10mm
    if(S3 == 0)
    {
        delayms(10);        //按键防抖
        if(S3 == 0)
        {
            distance += 10;
            while(!S3);      //等待按键释放
        }
    }
    //减距离，一次减 10mm
    if(S4 == 0)
    {
        delayms(10);
        if(S4 == 0)
        {
            distance -= 10;
            while(!S4);
        }
    }
    //校准按键，测出实际声速
    if(S2 == 0)
    {
        delayms(10);
        if(S2 == 0)
        {
            speed = 2 * STDDISTANCE / time;
            while(!S2);
        }
    }
}

```

```

//4 位数码管显示测量距离
void display()
{
    uint nTemp;
    uint nIndex;
    //显示个位，位选低电平有效
    nTemp = distance;
    Ge = 0;
}

```



```

nIndex = nTemp % 10;
P1 = Table[nIndex];
delayms(5);
Ge = 1;
//显示十位
nTemp /= 10;
Shi = 0;
nIndex = nTemp % 10;
P1 = Table[nIndex];
delayms(5);
Shi = 1;
//显示百位
nTemp /= 10;
Bai = 0;
nIndex = nTemp % 10;
P1 = Table[nIndex];
delayms(5);
Bai = 1;
//显示千位
nTemp /= 10;
Qian = 0;
nIndex = nTemp % 10;
P1 = Table[nIndex];
delayms(5);
Qian = 1;
}

```

九、测试结果及分析（包含实物图片）

测试完成了实验要求的三个功能，具体测试结果如下。

测试时，测距模块距离目标距离大约为 600mm，测试结果显示在四位数码管上，数码管上显示距离的单位是 mm。如果所测距离小于 500mm，则蜂鸣器进行蜂鸣报警。

按下复位键，则模块重新进行测距，并将所测距离显示在数码管上。

通过按下加距离按键或减距离按键，测量距离每次增加或减少 10mm，并将调整后的距离显示在四位数码管上。

实物照片如图 10 所示。

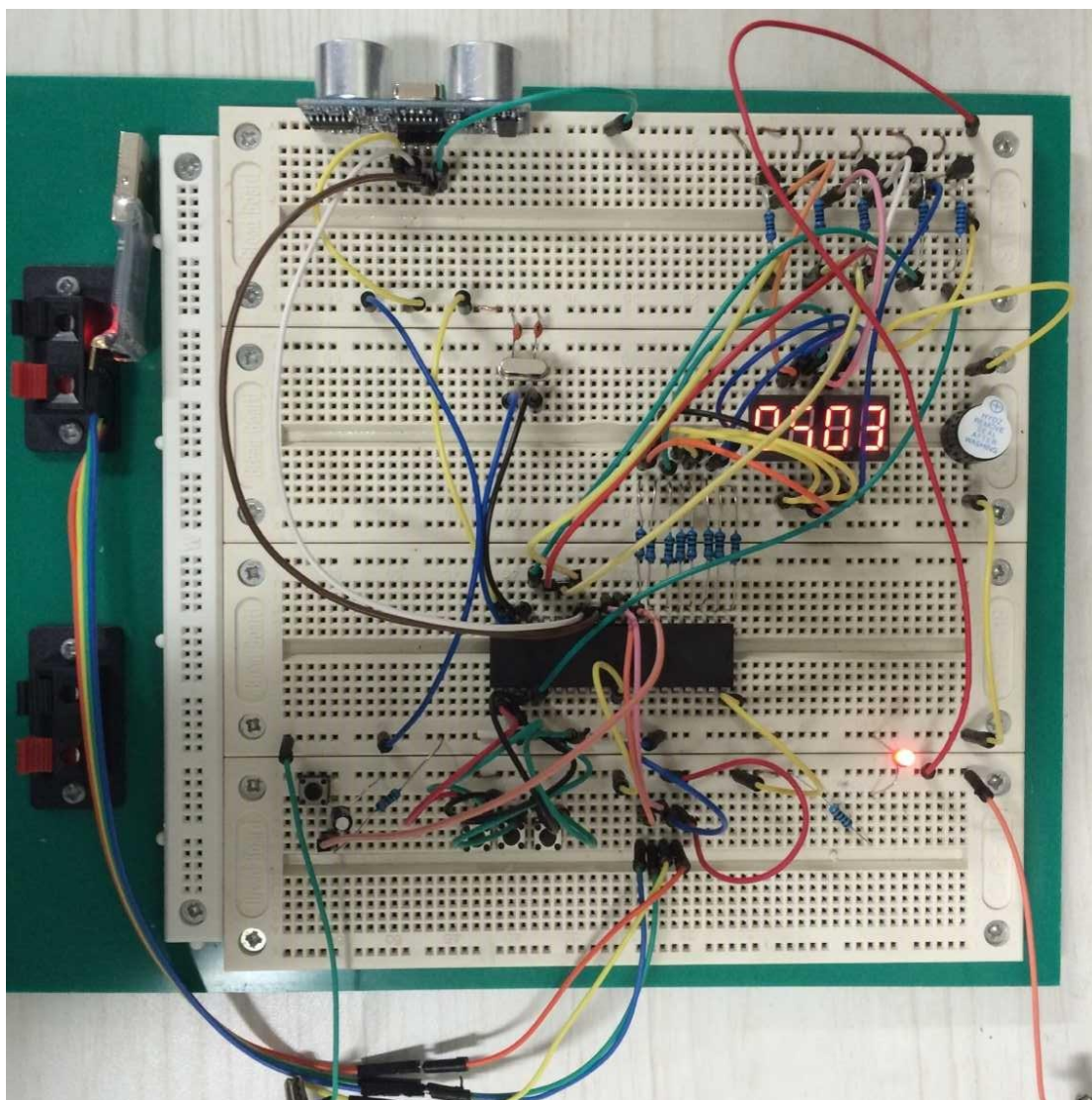


图 10 实物照片

十、遇到的问题及解决方案

问题 1：在电路插好之后，我们试着往单片机里下载程序，但无法成功。

解决方案：后来检查了才发现，原来是原理图中需要接+5V 电源的 40 和 31 号引脚没有接上电源，于是我们给这两个引脚接上了电源，就可以顺利将.hex 文件下载到单片机中了。

问题 2：第一版本的程序下载到单片机后，数码管能够实时显示测量出来的距离，但是单片机不能识别按键动作。

解决方案：首先分析一下数码管如何实现一直显示。由于执行一次数码管显示函数，只会让四位数码管依次点亮一下，所以为了让数码管持续显示，就要把数码管显示函数写在一个循环里。我们第一个版本的程序，只把数码管显示函数写在了循环里，而没有把按键扫描函数写进来，程序一直在执行显示函数，当然检测不到按键动作。后来我们把按键扫描函数也写到了循环了，这样单片机就能检测到按键动作并作出相应处理了。