




超声波+51单片机测距实验

原创

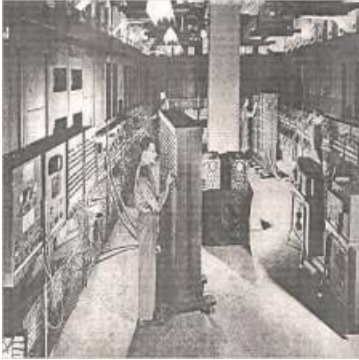
浮夸,  于 2020-09-14 17:46:15 发布  3850  收藏 80

分类专栏: [电子制作](#) 文章标签: [单片机](#) [电子制作](#) [传感器](#) [超声波](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44678969/article/details/108583295

版权



[电子制作](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

学习食谱

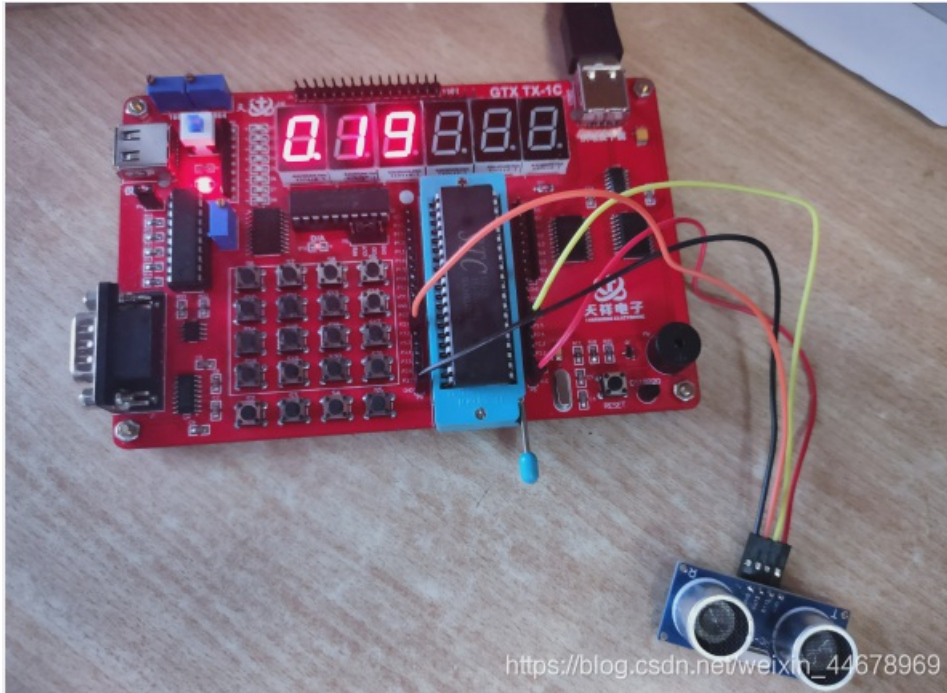
前言

- 一、超声波即测距原理
- 二、源代码
- 三、总结及扩展

前言

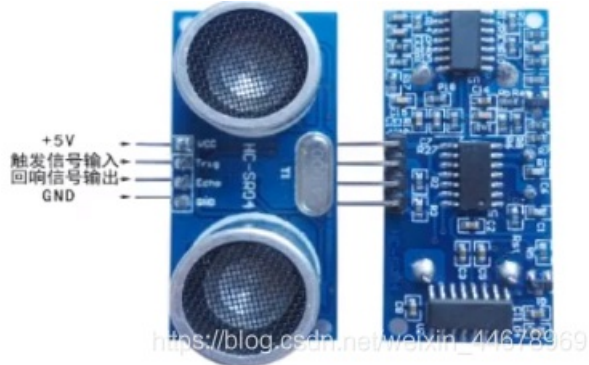
提示：

我这里选的超声波模块是 HC-SR04 ， 单片机使用的是51单片机。



一、超声波即测距原理

原理很简单就是依靠脉冲， 只要控制脉冲的时间就能触发其驱动。



看相关文档记录有：

- 采用IO口Trig触发测距，至少给 10us 的高电平信号触发（利用单片机的定时器可实现）
- 当有了19us的高电平后模块会自动发送8个40 khz 的方波，自动检测是否用信号返回
- 有信号返回，通过Echo输出，输出的高电平时间就是测距所用的时间
- 公式
测试距离 = (高电平持续时间*声速 (340M.S))/2

二、源代码

```
// 超声波测距
#include<AT89X51.H>
#include<intrins.h>
//引脚定义根据自己所连接的
```

```

#define RX P3_2
#define TX P2_5

// 数码管的段，位选端
sbit dula = P2^6;
sbit wela = P2^7;

#define uchar unsigned char
#define uint unsigned int

uint timer=0,time=0,index=0;
unsigned long length=0;
bit flag;

// 用个数组来保存测出来的距离
uchar disbuff[4] = {0,0,0,0};
// 数码管共阴极
uchar const table[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,
0x7d,0x07,0x7f,0x6f,0x40};
// 数码管的位选端
uchar const post[3] = {0xfe,0xfd,0xfb};

void Count();
void Show();

// 定时器初始化,
void init()
{
    TMOD = 0x11;
    TH0 = 0;
    TL0 = 0;
    TH1 = 0xf8; //2us 的时间
    TL1 = 0x30;
    ET0 = 1;
    ET1 = 1;
    TR1 = 1;
    EA = 1;
}

void main()
{
    init();
    while(1)
    {
        while(!RX); // 如果每接收到数据，则等待
        TR0 = 1; // 启动定时器0开始记录高电平的持续时间
        while(RX);
        TR0 = 0;
        Count();
    }
}

void Count()
{
    timer = TH0*256+TL0; // 存储TR0测得时间
    // 重新赋值
    TH0 = 0;
    TL0 = 0;
    // 还记得上面那个公式吗？ 这里 (*1.085) 是为了减少误差；

```

```

length = (float)(timer*1.085)*0.017;
//控制测量的距离, 如果超出距离则数码管显示 [-]
if((length>=700) || flag ==1)
{
    flag = 0;
    disbuff[0] = 10;
    disbuff[1] = 10;
    disbuff[2] = 10;
}
else
{
    disbuff[0] = length/100;
    disbuff[1] = length%100/10;
    disbuff[2] = length%10;
}
}

//TR0 定时器溢出都没信号返回, 肯定超出测量范围了
void stop() interrupt 1
{
    flag = 1;
}

//利用定时器TR1发送10us的方波驱动超声波工作
void start() interrupt 3
{
    TH1 = 0xf8;
    TL1 = 0x30;
    time++;
    //数码管显示函数
    Show();
    // 2*400=800ms, 这里是控制下一次的测量时间间隔, 不可能还没收到Echo就继续发送方波对吧?
    if(time >= 400)
    {
        time = 0;
        TX = 1;
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        TX = 0;
    }
}

void Show()
{
    dula = 0;
    if(index == 0)
        //这里 | 0x80 是为了显示 x.XX 的形式
        {P0 = (table[disbuff[index]] | 0x80);}
    else

```

```
CISC
{ P0 = table[disbuff[index]];}
dula = 1;
dula = 0;

wela = 0;

P0 = post[index];
wela = 1;
wela = 0;
if(++index >= 3)
    index = 0;
}
```

三、总结及扩展

到这对超声波模块就有了一个基本的了解了，其实原理非常简单，只要懂了原理看起代码来也就不那么费力了。

扩展：我们这里使用数码管显示的测量距离，那是不是也可以用1602的液晶显示呢？惭愧的是我没写出来，主要在写入的时间把握不好，这样就无法显示测量的距离。

在扩展一下可以做一个智能垃圾桶：

结合舵机云台，我们可以当到达一定距离时，利用云台驱动开垃圾桶。