

04

第四章 超声波测距



DFROBOT
DRIVE THE FUTURE

前情回顾

在上个章节里，学习Max的按键；if语句的使用。

本章内容

使用超声波测距，用串口打印出距离数值。

本章知识点

1. 介绍超声波测距的原理；
2. 超声波测距函数的使用；
3. 学习使用串口打印的函数；
4. Serial.print()和Serial.println()的区别；



一、编写并下载程序

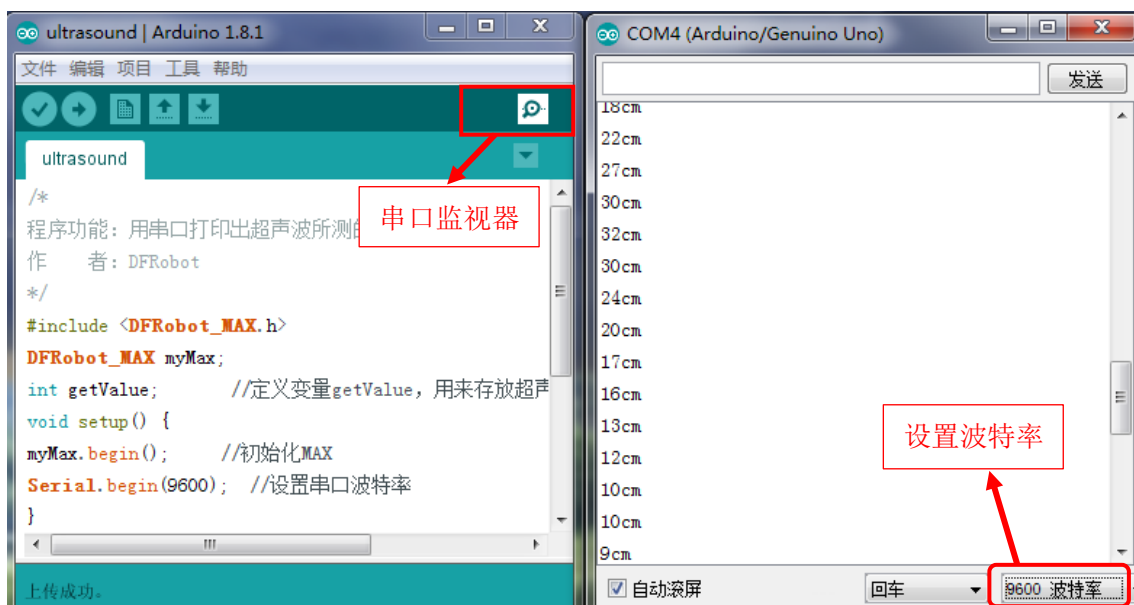
打开 **ArduinoIDE**，将下面的代码输入到编辑区中

```
/*
程序功能：用串口打印出超声波所测的距离。
作 者：DFRobot
*/
#include <DFRobot_MAX.h>
DFRobot_MAX myMax;
int getValue;          //定义变量 getValue，用来存放超声波读取到的数据

void setup() {
  myMax.begin();      //初始化 MAX
  Serial.begin(9600); //设置串口波特率
}

void loop() {
  getValue = myMax.distanceValue(); //读取超声波的值，单位 cm
  delay(300);                       //延时 300ms
  Serial.print(getValue);            //串口打印出超声波传感器的值
  Serial.println("cm");             //串口打印出“cm”
}
```

上传成功后，打开 MAX 的开关，然后点击“串口监视器”设置波特率为 9600，用手挡住 MAX 的前面就能在串口监视器里看见距离值（如下图）。注：串口打印时，必须要用 USB 线将 MAX 连接到电脑上。



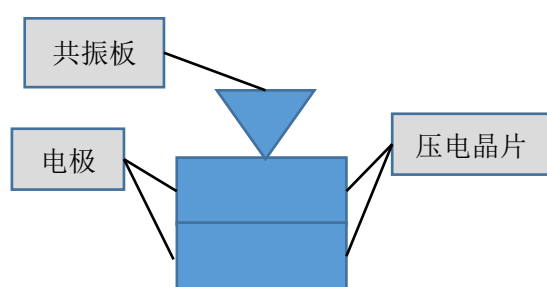
二、超声波测距的原理

测距的原理

超声波测距原理是通过超声波发射器向某一方向发射超声波,在发射时刻的同时开始计时,超声波在空气中传播时碰到障碍物就立即返回来其原理如图 1,超声波接收器收到反射波就立即停止计时。

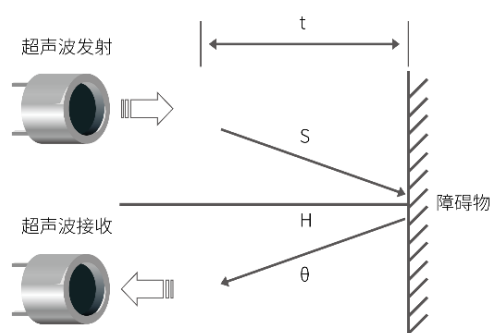
超声波传感器的原理

超声波传感器是将超声波信号转化成电信号,一般常用的超声波传感器是由压电晶片组成的。其内部结构和工作原理如图 1 和图 2。



超声波传感器内部结构

图 1



超声波传感器工作原理

图 2

来看看 MAX 身上的超声波传感器是不是很像一双水汪汪的大眼睛啊。如图 3

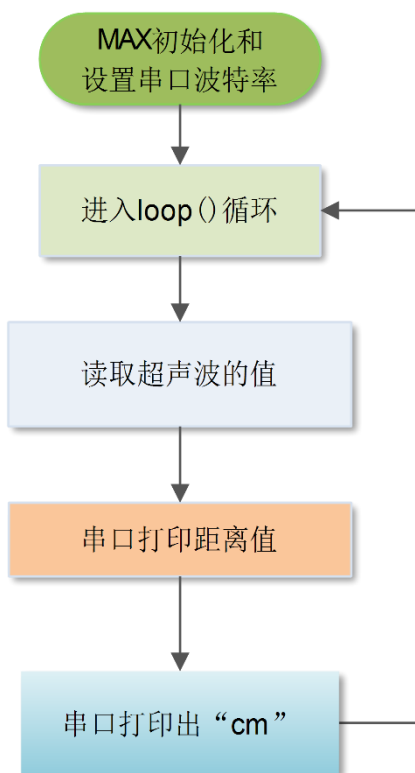


图 3. MAX 身上的超声波传感器

三. 代码回顾

我们已经大致了解了超声波的原理，那么在代码中是怎样来实现超声波测距，并让它通过串口显示出来呢？

首先我们还是先看看程序流程图（如右图），用此来了解这段程序的大致运行流程。



回到代码中；

首先定义一个变量，主要用来存放超声波读取到的数据：

```
int getValue;
```

然后在 `setup()` 函数里初始化串口：

```
Serial.begin(9600);
```

这个函数用于初始化串口波特率，也就是我们常说的数据传输的速率。设置串口波特率就是直接在括号里输入相应的数值就可以了，除了特定的无线模块对波特率有特殊要求，其他的波特率设置只需和串口监视器保持一致就行了。

接着进入 `loop()` 函数后就要开始读取超声波的值了；用函数 `myMax.distanceValue()` 读取超声波传感器的值。

示例代码如下：

```
getValue = myMax.distanceValue();
```

这句代码意思就是将读取到的超声波数值赋值给 `getValue` 。

那串口收到数据后，要如何在串口监视器上显示呢？这就要用到下面的两句语句了：

```
Serial.print(getValue)
```

```
Serial.println("cm");
```

下面我们就来了解一下这两个串口函数的用法。

四. Serial.print 和 Serial.println 串口函数的用法

串口是 Arduino 与其他设备进行通信的接口，我们需要很好的掌握它的使用。本章节主要使用了 `Serial.print` 和 `Serial.println` 串口函数。

其中 `print()` 可以这样解释：以我们可读的形式从串口输出来。而且这条命令的输出形式有很多种：

1、数字则是按照位的方式输出。

例：`Serial.print(56)`；输出“56”。

2、浮点型数据输出时，只会保留小数点后两位。

例：`Serial.print(1.23456)`；输出“1.23”。

3、字符和字符串就是按照原样输出，但是需要注意的是字符需要加上单引号（例 1），字符串就需要加上双引号（例 2）。

例 1：`Serial.print('N')`；输出“N”。

例 2：`Serial.print("Hello")`；输出“Hello”。

上面这些输出形式仅仅是我们常用的输出形式，想要了解更多的 Arduino 串口使用的相关函数可以参考[语法手册](#)。

然而 `println()` 与 `print()` 区别就是，`println()` 比 `print()` 多了回车换行，其他完全相同。除此之外，在代码中可能还会有一处不明白：

```
Serial.print(getValue);
```

可能大家会问，`getValue` 不是字符串吗？怎么输出的是数字呢？可是大家不要忘了，我们在代码的开头就定义了变量，它就是代表数字了，所以输出的就是数字。

课后思考

将下面的代码输入到编辑区中，编译并上传到 MAX 上，打开开关后，在 MAX 的面前挥动挥动你的小手，看看 MAX 会有什么变化？然后再试着分析这些代码的含义。（注：小手挥动的频率不要过快哦）

```
#include <DFRobot_MAX.h>
DFRobot_MAX myMax;
int i, getValue;

void setup() {
  myMax.begin();
}

void loop() {
  getValue = myMax.distanceValue();
  for(i=0;i<4;i++) {
    if(getValue>100)
    {
      myMax.rgbLed(i,0,255,0);
      delay(200);
      myMax.rgbLed(i,0,0,0);
    }
    else
    {
      myMax.rgbLed(i,0,255,255);
      delay(200);
      myMax.rgbLed(i,0,0,0);
    }
  }
}
```

体验了上面代码的功能后，你是否能试着修改上面的代码，做出当 MAX 在前进的途中遇到障碍物后做出的一些提示，例如：当 MAX 与障碍物的距离>200cm 的时候 RGB 灯循环亮绿色；当 MAX 与障碍物的距离>100cm 的时候 RGB 灯循环亮黄色。