

基于ESP32与phyphox的DIS实验制作（4）-使用超声波位移传感器研究弹簧振子的运动

原创

MakeMaker 于 2021-12-25 09:41:58 发布 1871 收藏

分类专栏: [基于ESP32与phyphox的DIS实验制作](#) 文章标签: [单片机](#) [嵌入式硬件](#)

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/qq_33378028/article/details/122139429

版权



[基于ESP32与phyphox的DIS实验制作](#) 专栏收录该内容

6 篇文章 1 订阅

订阅专栏

通过上一篇文章你应该已经了解了基于ESP32与phyphox的DIS实验大概是个什么模样了, 只是上一篇文章里仅仅使用了ESP32自带的模数转换(也就是电压传感器), 程序也是用了它自带的示例, 感觉还缺点意思。那么这一篇, 我们将会让ESP32连接外接的传感器, 通过编写程序驱动传感器测量数据并发送到手机上的phyphox。

好了, 还是和之前一样, 先来段视频:

(观看视频请关注微信公众号: 宁中物理创新实验室)

想不想也来一个视频中的位移传感器呢, 跟着下面的教程, 一起动手来制作吧!

1、超声波测距模块

(下图来自某宝截屏)



要注意一点, 大家购买的时候一定要买标注了宽电压的, 因为常见的模块都是5V电压的, 但是ESP32其实是一个工作在3.3V电压下的单片机, 不要觉得我们用USB给它供电的时候用的是5V, 那是因为里面有个降压模块将5V降低到了3.3V。所以切记, 一定要买能支持3.3V的超声波测距模块(其实貌似更便宜, 我买的才2.7元一个, 自己某宝去搜了, 免得我给别人打广告)

首先呢, 我们来学习一下这个超声波测距模块的原理。

“超声波: 人类耳朵能听到的声波频率为280HZ~20KHz。当声波的振动频率大于20KHz或小于20Hz时, 我们便听不见了。因此, 我们把频率高于20000赫兹的声波称为“超声波”。因其方向性好, 穿透能力强, 易于获得较集中的声能, 在水中传播距离远, 可用于测距、测速、清洗、焊接、碎石、杀菌消毒等。

最常用的超声测距的方法是回声探测法, 如下图, 超声波发射器向某一方向发射超声波, 在发射的同时计数器开始计时, 超声波在空气中传播, 途中碰到障碍物阻挡就立即反射回来, 超声波接收器收到反射回的超声波就立即停止计时。超声波在空气中的传播速度为340m/s, 根据计时器记录的时间 t , 就可以计算出发射点距障碍物面的距离 s , 即: $s=340t/2$ 。*



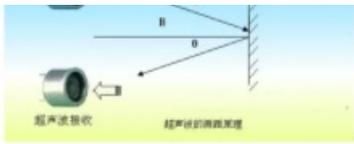
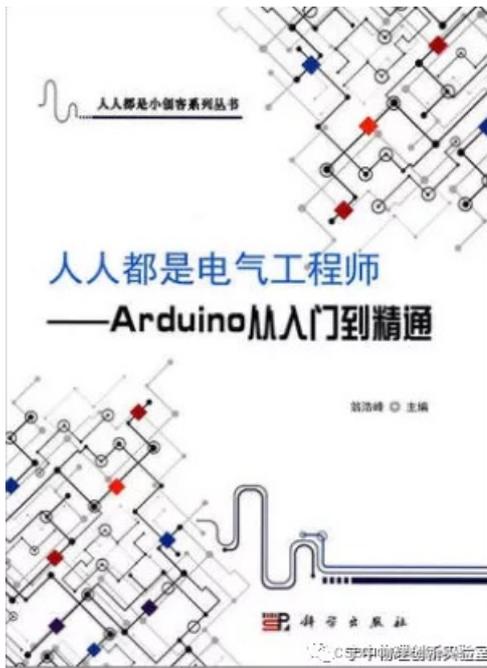


图 18-2 超声波测距原理图

超声波传感器HC-SR04上有4个针脚，分别是“VCC”、“Trig”、“Echo”和“GND”，“VCC”和“GND”不再解释，在“Trig”端施加一个10微秒高电压信号会触发超声波发射口发射8个超声波脉冲，当超声波接受口接收到超声波时“Echo”端会输出一段高电压，高电压持续的时间就是超声波从发射到接收的时间。”

以上内容来自我自己曾经写过的一本书（顺便做做书的广告了，有想买的同学自己去某宝或者某当，我手里的也是自己买来的，不过里面用的是Arduino，程序也是用图形化的，适合小学生或者刚入门者）：



大概了解了这个模块的原理后，就可以想办法让ESP32来驱动它了，其实也很简单，就是让ESP32给模块的“Trig”端加一个10微秒的高电压信号，然后再读取一下模块“Echo”端口上高电压持续的时间就可以了。

2、ESP32驱动超声波测距模块

在做正式版之前我们先编一个小程序玩玩，让ESP32驱动超声波测距模块，然后将测得的时间转换成距离发送给电脑，最后在电脑上来显示这个距离值。下面便是程序：

```

int trig_pin=12,echo_pin=13;//将超声波模块的trig接P12端口，echo接P13
float dis;//定义一个用来记录距离的变量
float checkdistance() { //这里为带有返回值的自定义函数，用来完成超声波模块的测量
    digitalWrite(trig_pin, LOW);//将trig端拉低
    delayMicroseconds(2);//等待2微秒
    digitalWrite(trig_pin, HIGH);//将trig端拉高
    delayMicroseconds(10);//等待10微秒
    digitalWrite(trig_pin, LOW);//将trig端拉低
    float distance = pulseIn(echo_pin, HIGH) / 58.00;//测量echo高电平持续时间，并将它转化为距离值，单位厘米
    return distance;//返回距离值
} //自定义函数结束

void setup() { //初始化函数
    pinMode(trig_pin,OUTPUT);//将trig端定义为输出端
    pinMode(echo_pin,INPUT);//将echo端定义为输入端
    Serial.begin(115200);//开启串口
}

void loop() { //主函数
    dis=checkdistance();//调用自定义函数测距，并将距离值赋值给dis变量
    Serial.println(dis);//通过串口输出dis变量值
    delay(100);//等待100毫秒
}

```

具体的接线为：

超声波模块 ESP32

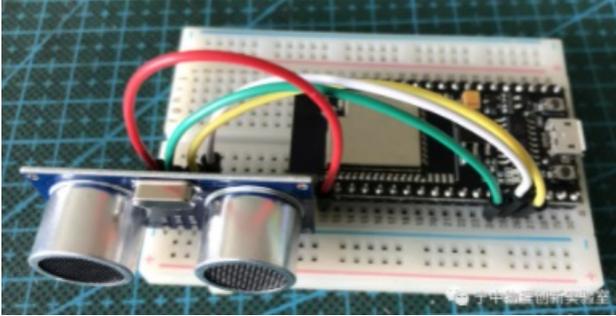
Vcc ----- 3V3

Gnd ----- GND

Trig ----- P12

Echo ----- P13

下图是我用一个面包板来帮助接线（不知道面包板是什么，请自行百度，或者买我的书来看），由于这块ESP32端口号都印在了背面，插到面包板后就看不到了，所以请先拍照保存，接线时请断开电源并仔细认真，不要接错了以免烧板！



接好线烧完程序后，点击Arduino IDE右上角的串口监视器，就像放大镜的那个图标。

```
sketch_jun25a | Arduino 1.8.13
文件 编辑 项目 工具 帮助
串口监视器
sketch_jun25a $
int trig_pin=12,echo_pin=13;//将超声波模块的trig接P12
float dis;//定义一个用来记录距离的变量
float checkdistance() { //这里为带有返回值的自定义函数,
    digitalWrite(trig_pin, LOW);//将trig端拉低
    delayMicroseconds(2);//等待2微秒
    digitalWrite(trig_pin, HIGH);//将trig端拉高
    delayMicroseconds(10);//等待10微秒
    digitalWrite(trig_pin, LOW);//将trig端拉低
}
Leaving...
Hard resetting via RTS pin...
CSDN @MakeMaker
```

点击后就可以打开串口监视器，注意将右下角的波特率设置为115200，之后就可以看到从ESP32发过来的距离数据了。

```
COM3
发送
11.22
11.22
11.28
14.29
10.55
11.59
9.78
8.86
10.50
14.90
9.84
9.17
9.47
28.52
29.41
CSDN @MakeMaker
```

3、基于ESP32和phyphox的位移传感器

接下去进入今天的最后一段，开始来完成视频中的位移传感器制作。

其实接线已经不需要改变了，只要改一下程序就可以了，以下是程序，注意阅读程序里的注释部分来理解程序，开始学的时候可以抄别人的程序，读懂程序，并尝试做一些小的修改，例如改一下采样频率，在phyphox上显示的文字啦之类的。

```
// 本程序由宁中物理创新实验室所有，如需使用请注明出自宁中物理创新实验室！
#include <phyphoxBle.h> // 加载phyphoxBle库函数
int trig_pin=12,echo_pin=13;//将超声波模块的trig接P12端口，echo接P13
float dis;//定义一个用来记录距离的变量
float checkdistance() { //这里为带有返回值的自定义函数，用来完成超声波模块的测量
    digitalWrite(trig_pin, LOW);//将trig端拉低
    delayMicroseconds(2);//等待2微秒
    digitalWrite(trig_pin, HIGH);//将trig端拉高
    delayMicroseconds(10);//等待10微秒
    digitalWrite(trig_pin, LOW);//将trig端拉低
    float distance = pulseIn(echo_pin, HIGH) / 58.00;//测量echo高电平持续时间，并将它转化为距离值，单位厘米
    return distance;//返回距离值
} // 自定义函数结束

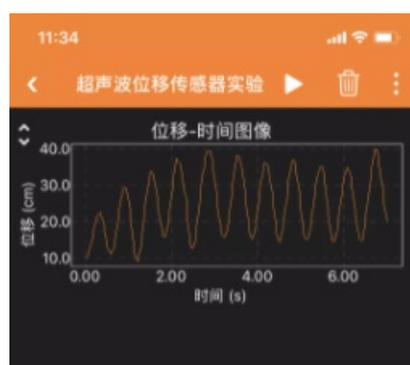
void setup() { // 初始化函数从“{”开始到“}”结束
    // setup是初始化函数，也就是esp32刚通电时会运行一遍
    pinMode(trig_pin,OUTPUT);//将trig端定义为输出端
    pinMode(echo_pin,INPUT);//将echo端定义为输入端
    Serial.begin(115200); // 开启esp32的串口，用来发数据给电脑，这里可以不用
    PhyphoxBLE::start("位移传感器");//开启蓝牙，引号内的文字会出现在phyphox搜索蓝牙设备时
    PhyphoxBleExperiment Voltmeter;//开启一个实验，取名为Voltmeter
    Voltmeter.setTitle("超声波位移传感器实验");//会显示在phyphox主页面上的实验名称
    Voltmeter.setCategory("宁中物理创新实验室");//会显示在phyphox的实验类别
    Voltmeter.setDescription("用来测位移的！");//会显示在phyphox的实验介绍里
    PhyphoxBleExperiment::View firstView;//在实验界面里添加一个显示界面
    firstView.setLabel("位移");//界面上显示的标签
    PhyphoxBleExperiment::Graph firstGraph; //在phyphox里添加一个图像
    firstGraph.setLabel("位移-时间图像");//图像的名称
    firstGraph.setUnitX("s");//x轴坐标单位
    firstGraph.setUnitY("cm");//y轴坐标单位
    firstGraph.setLabelX("时间");//x轴物理量名称
    firstGraph.setLabelY("位移");//y轴物理量名称
    firstGraph.setChannel(0, 1);//开启一个蓝牙数据通道
    firstView.addElement(firstGraph); //将图像添加到显示界面上
    Voltmeter.addView(firstView); //将显示界面添加到实验
    PhyphoxBLE::addExperiment(Voltmeter); //将实验设置通过蓝牙发送到phyphox
} // 初始化函数结束，在初始化函数里其实设置了一些phyphox中要显示的元素
// 以下为主程序，esp32在运行时主程序内的程序是循环运行的
// 也就是第一遍结束第二遍又开始，周而复始，直到断电

void loop() { // 主程序开始
    dis=checkdistance();//调用自定义函数测距，并将距离值赋值给dis变量
    delay(1); // 运行暂停1毫秒，让蓝牙模块准备好
    PhyphoxBLE::write(dis);//将voltage通过蓝牙发送到手机端的phyphox
    delay(18);//暂停18毫秒，这里经测试最小可以填4，再小esp32就会卡死，也就是差不多50Hz
} // 主程序结束
```

将程序上传到ESP32后，打开手机上的phyphox，用上一篇中的方法添加实验：

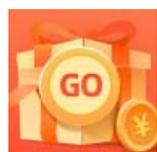


然后打开超声波位移传感器实验，连接名为“位移传感器”的蓝牙设备，测试一下吧：



如果你想用它来研究弹簧振子的运动，请参照第二篇中的内容，将数据共享到电脑端，然后用excel进行数据拟合处理。由于超声波测距仪的原理所致，因为声速会受到温度、风速、湿度、气压等等各种因素的影响，所以为了更精确的测量距离还需要测量这些值，并都声速进行修正，当然除了这些因素外，ESP32本身的计时精度也会影响距离的测量，所以超声波测距模块其实精度并不高，一般情况下只能精确到1cm左右，所以用它来做物理的位移传感器其实并不是很好，当然也有贵点的超声波测距模块可以选用，可以适当增加精度。在之后的篇章中，我会为大家带来更精确的位移测量模块，敬请期待吧！这篇到这里就结束了，回顾一下这篇的内容：

- (1) 认识了超声波测距模块。
- (2) 使用ESP32读取超声波测距模块的小实验。
- (3) 基于ESP32和phyphox的超声波位移传感器制作。



[创作打卡挑战赛](#)

[赢取流量/现金/CSDN周边激励大奖](#)