**基于Arduino的智能超声波垃圾桶设计**

邵帅，张伶杰，殷鑫军，吕冠华

**摘要：设计了能够通过超声波感应控制开关的智能垃圾桶，该垃圾桶以Arduino UNO 开发板为核心，由外部多个传感器采集数据，通过在Arduino IDE 上的编程控制数据采集，经处理换算后感应开关垃圾桶，无接触的开关垃圾桶在当下，更具有卫生性，安全性等特点。**

**关键词：Arduino；超声波；感应开关**

**1引言**

随

着文明的进步，人们对卫生的要求愈发的提高，特别是疫情期间，更多的提倡无接触式操作，而垃圾桶作为收集垃圾的容器，更容易滋生细菌和残留病菌，无需接触打开的感应垃圾桶可以很好的解决这些问题，本设计通过超声波传感器判断是否有人需要扔垃圾，从而自动打开垃圾桶，等待一段时间后再自动关闭，具有很大的实际效用。

2**总体设计方案**

该感应垃圾桶主要有四个模块：Arduino控制模块、超声波测距模块，舵机模块、led照明模块，其中Arduino控制模块为重要部分。Arduino开发板作为核心部件，负责接受超声波测距模块的信号，控制舵机的转动和led的开关。

**3 硬件模块设计**

**3.1Arduino控制模块**

卡通人物

描述已自动生成

图1 Arduino控制模块

Arduino Uno是一款基于ATmega328P的微控制器板。它有14个数字输入/输出引脚(其中6个可用作PWM输出)，6个模拟输入，16MHz晶振时钟，USB连接，电源插孔，ICSP接头和复位按钮。只需要通过USB数据线连接电脑就能供电、程序下载和数据通讯[1]。

表1 Arduino Uno引脚说明

|  |  |
| --- | --- |
| 工作电压 | 5v |
| 输入电压（推荐） | 7-12v |
| 数字I/O引脚 | 14(six of them provide PWM output) |
| pwm数字I/O引脚 | 6 |
| 模拟输入引脚 | 6 |
| 每个I/O引脚的直流电流 | 20mA |
| 3.3v引脚的直流电流 | 20mA |

**3.2超声波测距模块**

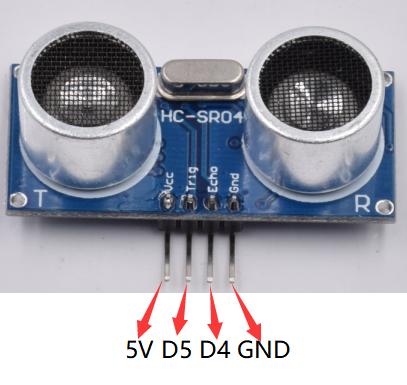


图2超声波测距模块引脚

超声波测距发射装置发射超声波的原理，可以通过接收器和超声波的时间差来了解。这和雷达测距很相似。超声波发射器向一定方向发射超声波，在定时的瞬间同时开始，超声波在空气中遇到障碍物立即返回，超声波接收器接收到反射波后立即停止定时。(超声波在空气中传播的速度是340米/秒。根据计时器的时间，可以计算出发射点到障碍物的距离，即 s = 340t/2)。

**3.3 舵机**



图3舵机引脚

舵机的转动的角度是通过调节PWM（脉冲宽度调制）信号的占空比来实现的，标准PWM（脉冲宽度调制）信号的周期固定为20ms（50Hz），理论上脉宽分布应在1ms到2ms 之间，但是，事实上脉宽可由0.5ms 到2.5ms 之间，脉宽和舵机的转角0°～180°相对应。

**3.4 RGB和led**

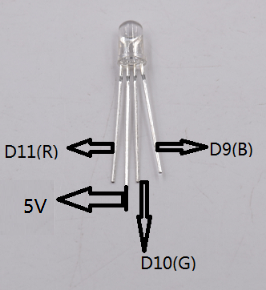


图4 RGB引脚通过接受电信号来闪烁。

图示, 示意图

描述已自动生成

图5 led样式

RGB和 Led通过接受电信号来闪烁。

**4. Arduino电路模块设计**

电路总体设计首先通过Tinkercad制作，具体电路模块如下图所示：

图示

描述已自动生成

图6 模块电路图

图形用户界面, 应用程序, Teams

描述已自动生成

图7 组件列表

实际电路按照该模拟电路进行接线。

**5 实际组装成品**

图片包含 室内, 绿色

描述已自动生成卡通人物

低可信度描述已自动生成

图8 外部样式和内部样式

通过实际测试，将感应距离设置在20cm，当超声波测距模块感应到有物体靠近垃圾桶20cm内，经过数据处理，将会控制舵机打开垃圾桶盖，并且led将会开启和闪烁，盖子开启五秒后，舵机将会恢复原位并带动盖子关闭。

**6 结束语**

该超声波智能垃圾桶，可以实现通过感应自动打开垃圾桶盖，有效解决了关于垃圾桶的卫生问题，在当下拥有很大的实际效益。在此基础上，我们可以增加更多的传感器原件来实现更多的功能，智能垃圾桶将成为未来的普遍品，以满足人们生活中对扔垃圾时的卫生要求。

参考文献；

[1] Tony Code. Arduino Uno R3开发板介绍[EB/OL]. https://www.jianshu.com/p/afe07479b2c2, 2018-7-30/2020/12/3

程序代码：

#include <Servo.h>

Servo myServo;

const int TrigPin = 2;

const int EchoPin = 3;

float cm;

void setup()

{

myServo.attach(9);

Serial.begin(9600);

pinMode(TrigPin, OUTPUT);

pinMode(EchoPin, INPUT);

pinMode(5,OUTPUT);

pinMode(6,OUTPUT);

pinMode(8, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(13,OUTPUT);

pinMode(12,OUTPUT);

}

void loop(){

digitalWrite(12, HIGH);

digitalWrite(13, HIGH);

digitalWrite(6,LOW);

digitalWrite(TrigPin, LOW);

delayMicroseconds(2);

digitalWrite(TrigPin, HIGH);//

delayMicroseconds(10);

digitalWrite(TrigPin, LOW);

cm = pulseIn(EchoPin, HIGH) / 58.0;

Serial.print("cm=");

Serial.print(cm);

if (cm<=20 && cm>=0){

lightRGB();

openDoor();

}

Serial.print("cm");

Serial.println();

delay(1000);

}

void lightRGB(){

digitalWrite(8,LOW);

digitalWrite(10,HIGH);

digitalWrite(11,HIGH);

delay(150);

digitalWrite(8,HIGH);

digitalWrite(10,LOW);

digitalWrite(11,HIGH);

delay(150);

digitalWrite(8,HIGH);

digitalWrite(10,HIGH);

digitalWrite(11,LOW);

delay(150);

}

void openDoor(){

digitalWrite(5,HIGH);

for(int i=0; i<=100;i++){

myServo.write(i);

delay(10);

}

delay(5000);

for(int i =100; i>=0; i--){

myServo.write(i);

delay(5);

}

delay(100);

digitalWrite(5,LOW);

}