实验名称： 超声波测距仪

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生： | 姓名： | | 陈浩宇 | | 学号： | 20375173 | | 班级： | | 210614 | | 报告成绩： | | | |  | |
| 任课教师： | | 李成 | | | | | 验收时间： | | 2023 | | 年 | | 5 | 月 |  | | 日 |
| 验收小组签字： | | | |  | | | | | | | | | | | | | |

## 实验需求分析

1.实验1:Arduino控制超声波模块

实验内容：Arduino给超声波模块发送触发信号，并接收超声波模块的回响信号，根据公式计算出测量的距离，将结果通过串口上传到计算机上；当目标距离小于10厘米时，LED灯闪烁报警；按照实验记录表进行记录与分析

实验目的：掌握使用Arduino控制超声波传感器测距的方法，测出与障碍物的距离

实验步骤：将超声波模块的四个引脚与Arduino相连，其中超声波模块的Vcc引脚与5V供电连接，GND引脚与GND连接，Trig和Echo所接引脚要与程序中设置端口保持一致；将LED的负极通过电阻接地，正极与程序中预设的引脚相连

实验效果：当目标距离小于10厘米时，LED快速闪烁报警，同时距离计算结果也将通过串口上传到Arduino监视器里

2.实验2：Arduino超声波测距仪

实验内容：在实验1的基础上将测量结果循环显示在1位8段数码管上，整数部分为4位，以0开头，小数部分为1位；使用DHT11模块测量环境温湿度，根据温湿度值按照声速温湿度表修正声速值，进一步提升测量精度，将温湿度、声速和距离计算值通过串口上传，结果保留2位小数；对修正后的距离数据分别进行平均数和中位数滤波提升测量精度，可通过按键key1和key2进行选择数码管显示的是平均数和中位数，同时将原始距离数据、平均数滤波结果/中位数滤波结果通过串口绘图器画出波形图并截图记录

实验目的：考虑温度与湿度因素进而提高超声波测距的精度

实验步骤：超声波测距模块与LED灯的接法与实验1一致，注意LED引脚要与程序对应；将8个电阻连接到面包板上，电阻的一端与8位显示数码管相连，另一端分别插入互不连接的插孔中；将数码管的公共端通过导线连接到电源负极；用导线将数码管的各个段对应的引脚与Arduino相连，相连时各个引脚要与程序中预设的引脚一一对应；程序中预设的数码管为2-9号引脚

实验效果：数码管显示测量的距离，通过按键key1、key2可以选择数码管输出的是平均数还是中位数，原始距离数据、平均数滤波、中位数滤波结果波形显示在串口绘图器中

## 实验原理论证

①Vcc和GND为供电端口，连接+5V电源；Trig为超声波模块的输入（对应为Arduino的输出引脚），用来触发超声波测距功能，需要一个至少10微秒的高电平信号来进行触发；触发测距后，超声波模块自动发送8个40kHz的方波，并自动检测是否有信号返回；当有信号返回，超声波模块通过Echo输出一个高电平，高电平持续的时间就是超声波从发射到返回的时间，根据公式s = vt，可以得到：距离(s) =高电平持续时间(T) × 声速(v) / 2；当无信号返回或返回信号未超过检测门限，超声波模块通过Echo输出一个38ms的高电平；

②使用超声波模块发射和接收40kHz的超声波脉冲，通过测量脉冲从发射到接收所需的时间，计算出探头到物体表面之间的距离。同时，由于声速会受到温湿度的影响，使用DHTI1模块测量环境温湿度，便于后续根据声速温湿度表对声速进行修正

③使用Arduino单片机对超声波模块和DHT11模块的信号进行读取分析和处理，根据声速温湿度表修正声速值，计算出准确的距离值。

④使用Arduino单片机将处理后的距离值通过串口输出到计算机，使用数码管将距离值以厘米为单位实时显示出来或者使用计算机的串口监视器来查看距离值。

⑤传输的数据通过串口监视器、串口绘图器、数码管等方式来进行显示。

## 方案选择与电路设计

实验一：先将超声波模块与串口相连，包括接地的引脚、接电源的引脚、输出数据的引脚。在超声波模块所输出的数据传输进去之后，判断需要LED亮/灭之后，将输出的高低电平连接到LED灯，并串联上一个限流电阻

实验二：在实验一的基础上，将温湿度传感器接入整个系统当中去，输出温湿度的数据传输进整个系统，对声速进行修正。此外通过设计，将获得的数据输出到数码管上显示，每一个输出的串口控制数码管的一段

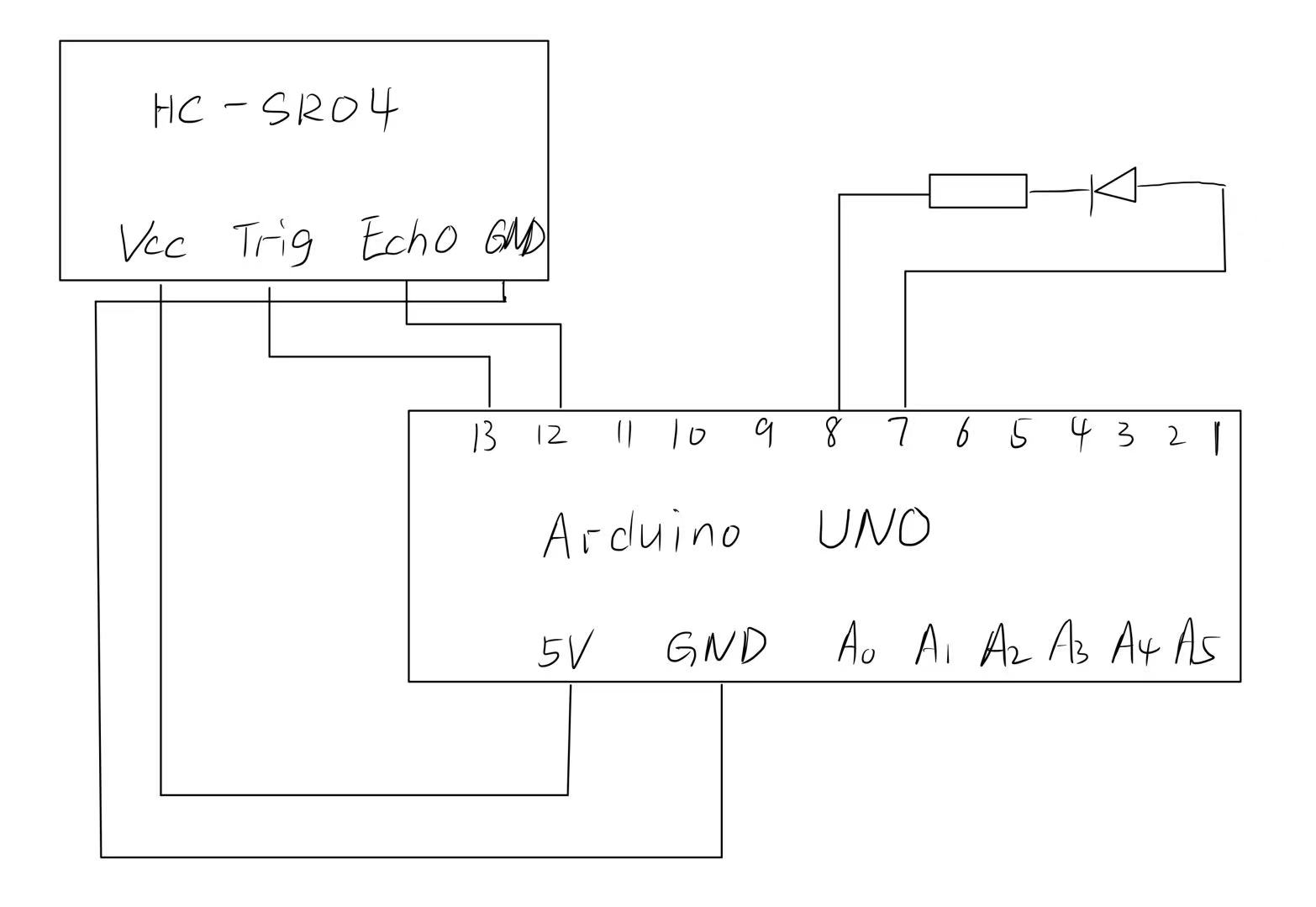


图1 实验1电路图

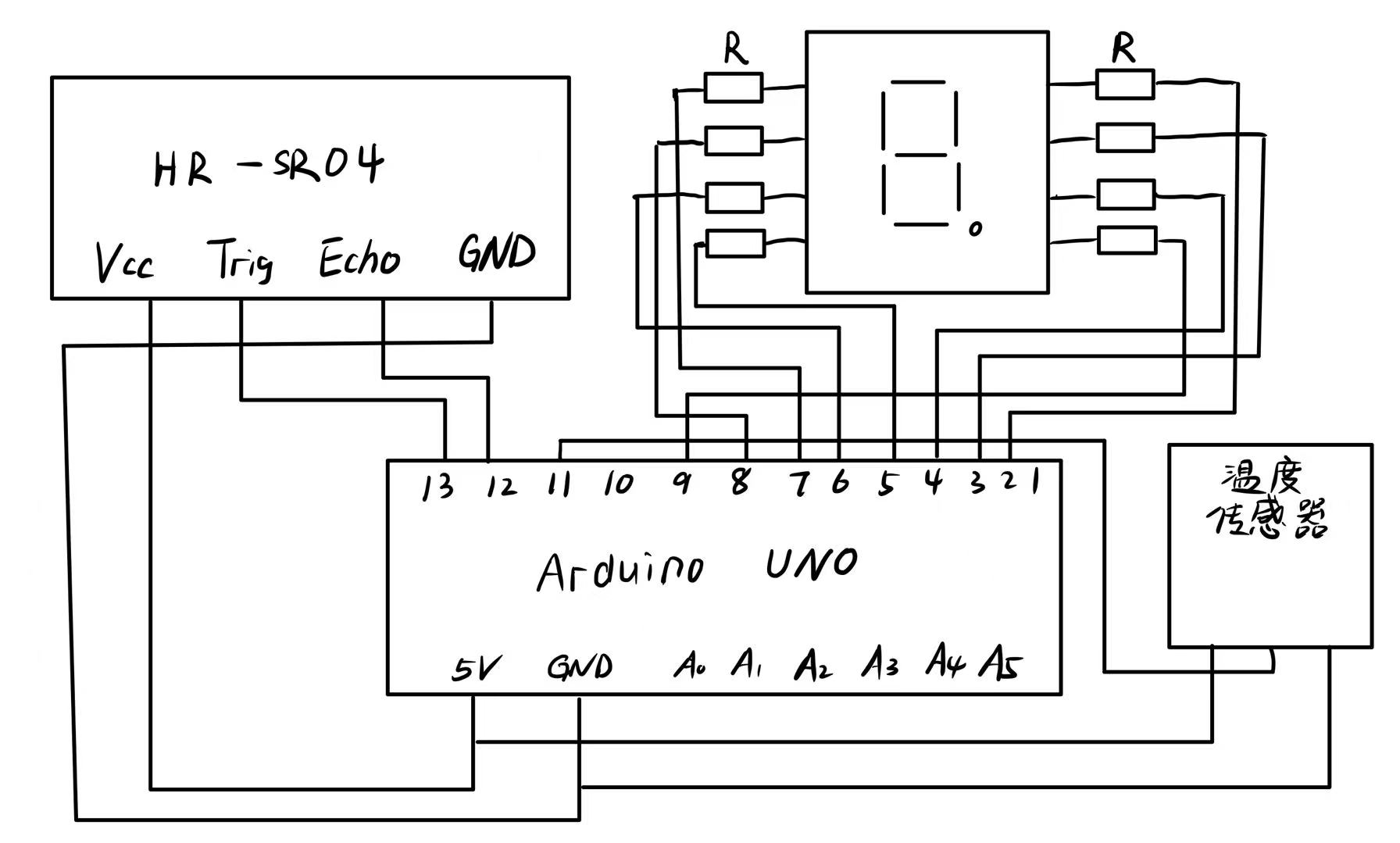


图2 实验2电路图

## 实验代码编写

实验一伪代码：

void setup() {

设置波特率和引脚

初始化Timer

}

void loop() {

通过pulseIn函数读取时间

利用公式计算距离

将距离与阈值比较来控制LED亮灭

}

实验二伪代码：

void setup() {

设置波特率和引脚

初始化Timer

初始化温湿度传感器相关函数

}

void inputData() {

读取10组数据及温度湿度

利用修正的声速计算出当前距离

}

float Filter\_Av() {

计算平均数

}

float Filter\_Md() {

计算中位数

}

void loop() {

调用inputData()函数读取一组距离

计算中位数、计算平均数

根据按键选择要输出中位数还是平均数

将距离与阈值比较来控制LED亮灭

控制数码管的亮灭

}

## 实验过程与结果

在实验1中按照表1中所列距离放置障碍物，使用搭建的Arduino超声波测距系统测量，将测量结果填入实验表1，并计算平均数和中位数。

表1 超声波测距数据记录分析表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 设置 | 1次 | 2次 | 3次 | 4次 | 5次 | 6次 | 7次 | 平均数 | 中位数 |
| 5cm | 4.89 | 4.88 | 4.79 | 4.77 | 4.90 | 4.77 | 4.89 | 4.84 | 4.84 |
| 20cm | 20.02 | 20.30 | 20.28 | 20.32 | 20.28 | 20.30 | 20.27 | 20.25 | 20.28 |
| 50cm | 48.72 | 48.89 | 48.71 | 48.60 | 48.89 | 48.87 | 48.98 | 48.81 | 48.81 |
| 80cm | 77.56 | 77.93 | 77.96 | 77.85 | 77.20 | 76.96 | 78.02 | 77.64 | 77.64 |

在实验2中按照表2中所列距离放置障碍物，使用搭建的Arduino超声波测距仪测量，将测量结果填入实验表2，并进行分析。

表2 超声波测距仪数据记录分析表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 设置 | 温度 | 湿度 | 声速 | 1次 | 2次 | 3次 | 4次 | 5次 | 平均数 | 中位数 |
| 5cm | 29.3 | 16% | 347.76 | 5.01 | 5.02 | 5.01 | 4.99 | 5.02 | 5.01 | 5.01 |
| 20cm | 29.3 | 16% | 347.76 | 20.01 | 20.02 | 19.99 | 20.03 | 20.01 | 20.01 | 20.01 |
| 50cm | 29.3 | 16% | 347.76 | 50.10 | 49.93 | 50.05 | 49.88 | 49.90 | 49.97 | 49.93 |

按照表2所设置的距离，将对应的原始数据-平均数滤波和原始数据-中位数滤波的波形图截取粘贴至此处，并对现象进行分析。

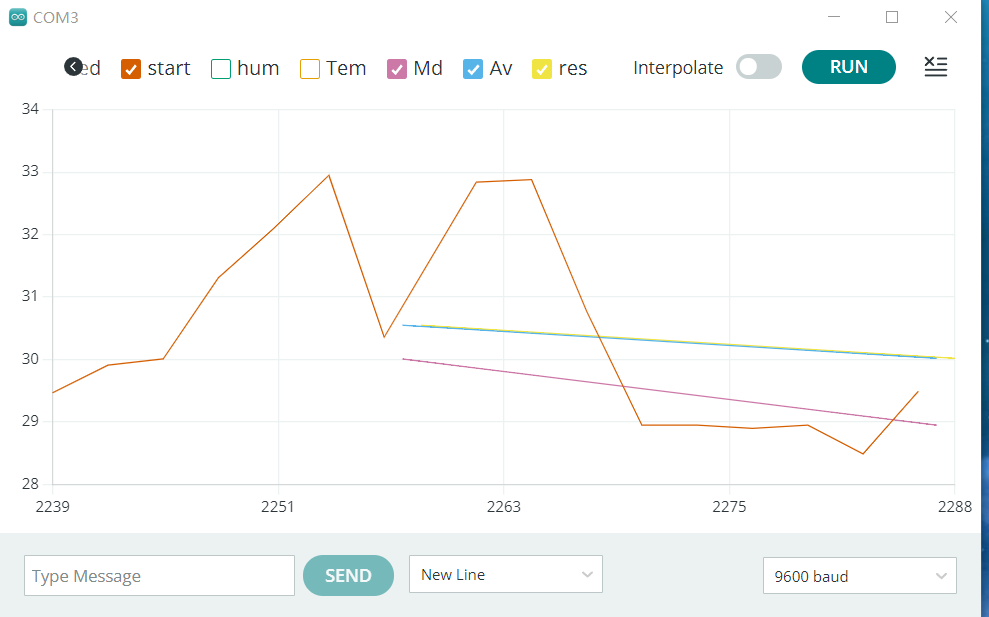
**

图3 key1输出平均数

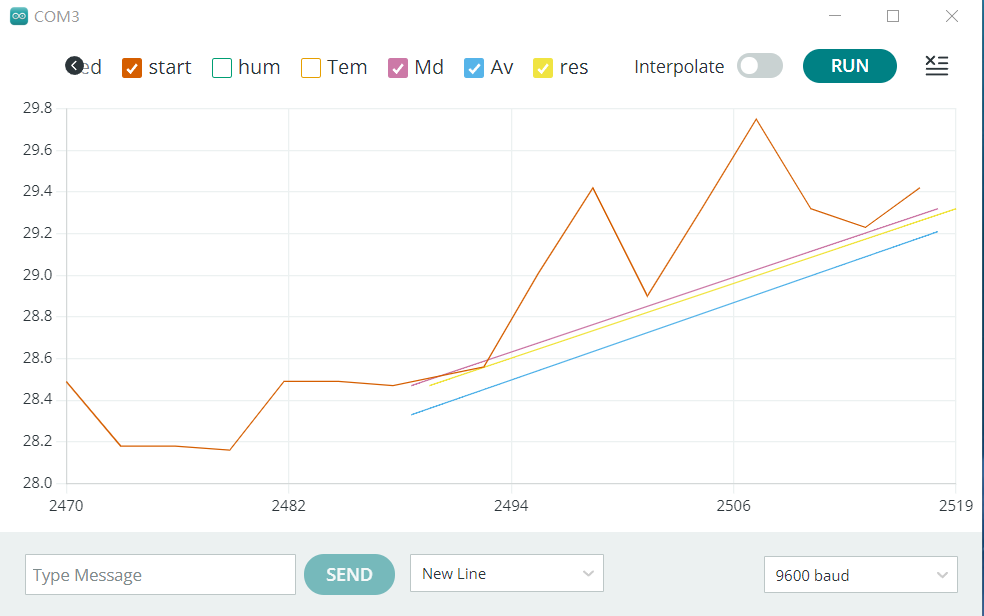
**

图4 key2输出中位数

进行温湿度修正后，大部分数据的平均数及中位数均有一定程度的增加，相较于实际值的相对误差和绝对误差均有减少，精度得到了改善。而且在加入了滤波之后，可以看见最后测得的数据明显与理论值更加接近，尽管只是一个小小的滤波，对于实验的准确性有非常大的提高

## 实验结果总结

问题一：通过按键key1和key2控制数码管显示中位数和平均数，key1和key2无法与引脚6，7相连，与数码管引脚发生冲突，导致显示异常

解决办法：外接导线至A0、A1引脚，控制A0、A1处的高低电平来模拟key1与key2

问题二：实验电路过于复杂，出现错误时不容易观察

解决办法：事先规划好电路的连接，使用两块面包板进行连接，避免导线间的相互干扰

## 实验思考题

1、超声波模块的最小和最大测试距离和什么参数相关？

换能器性能、输出功率、变压器参数、激励电压

2、有哪些影响超声波距离测量精度的因素？

空气湿度、空气温度、气压、介质均匀性、噪音、振动

3、编程过程中数据格式转换受哪些因素影响？

数据格式的转换主要受到转换类型的影响。转为类型即两个要转换的数据格式，在内存中，存储的位数不一样，如将浮点型(32bit)转换为整形(8bit)要去掉小数部分

4、通过实验，你掌握了哪些科研方法？

文献查阅法：在本实验中，为了得到更精确的声速进而测出更精准的距离，需要查阅温湿度修正速度表

数据滤波与可视化：通过使用DHT11模块测量环境温湿度，并根据声速温湿度表修正声速值，了解到了如何提高测量精度；修正优化的思想在以后的实验中也会产生好的收益

实验数据分析：将实验数据进行分析和整理，可以让我们更直观的分析出实验数据的变化趋势，数据可视化

## 实验感想

通过这个实验我学习到了超声波测距的原理和方法，了解到了超声波模块的使用方法，并进一步锻炼了动手实践的能力；通过运用Arduino编写程序控制硬件，我学会了使用pulseIn()函数来检测高低电平的脉冲宽度，同时复习了串口通信和测量环境温湿度的相关知识。

附录 源代码

实验一：

const int trig = 13;

const int echo = 12;

const int led = 10;

float distance\_cm;

void setup() {

  Serial.begin(9600);

  pinMode(trig, OUTPUT);

  pinMode(echo, INPUT);

  pinMode(led, OUTPUT);

}

void loop() {

  digitalWrite(trig, LOW);

  delay(10);

  digitalWrite(trig, HIGH);

  delay(10);

  digitalWrite(trig, LOW);

  distance\_cm = pulseIn(echo, HIGH) / 58.8;

  distance\_cm = (int (distance\_cm \* 100.0)) / 100.0;

  if (distance\_cm < 10.00) {

    digitalWrite(led, HIGH);

    delay(100);

    digitalWrite(led, LOW);

    delay(100);

  } else {

    digitalWrite(led, LOW);

  }

  Serial.print("Distance:");

  Serial.print(distance\_cm);

  Serial.println("cm");

  delay(100);

}

实验二：

#include <TimerOne.h>

#include "DHT.h"

#define FLITER\_N 9

float TempHumdTable[15][11] = { { 331.4313, 334.4107, 337.2659, 340.0071, 342.6441, 345.1864, 347.6434, 350.0246, 352.3395, 354.5979, 356.8101 },

                                { 331.447, 334.4334, 337.2982, 340.0525, 342.707, 345.2724, 347.7595, 350.1796, 352.5442, 354.8655, 357.1564 },

                                { 331.4627, 334.4561, 337.3306, 340.0979, 342.7698, 345.3583, 347.8756, 350.3346, 352.7488, 355.1329, 357.5024 },

                                { 331.4783, 334.4787, 337.3629, 340.1433, 342.8327, 345.4442, 347.9916, 350.4895, 352.9533, 355.4, 357.848 },

                                { 331.494, 334.5014, 337.3952, 340.1887, 342.8955, 345.5302, 348.1076, 350.6443, 353.1577, 355.667, 358.1933 },

                                { 331.5097, 334.5241, 337.4275, 340.234, 342.9584, 345.616, 348.2236, 350.799, 353.3619, 355.9338, 358.5383 },

                                { 331.5253, 334.5468, 337.4598, 340.2794, 343.0212, 345.7019, 348.3395, 350.9537, 353.566, 356.2003, 358.8829 },

                                { 331.541, 334.5694, 337.4921, 340.3248, 343.084, 345.7878, 348.4554, 351.1083, 353.7701, 356.4667, 359.2272 },

                                { 331.5567, 334.5921, 337.5244, 340.3701, 343.1468, 345.8736, 348.5713, 351.2629, 353.9739, 356.7329, 359.5712 },

                                { 331.5723, 334.6148, 337.5567, 340.4155, 343.2096, 345.9594, 348.6871, 351.4174, 354.1777, 356.9988, 359.9148 },

                                { 331.588, 334.6375, 337.589, 340.4608, 343.2724, 346.0452, 348.8028, 351.5718, 354.3814, 357.2646, 360.2581 },

                                { 331.6036, 334.6601, 337.6213, 340.5062, 343.3352, 346.1309, 348.9186, 351.7261, 354.5849, 357.5302, 360.6011 },

                                { 331.6193, 334.6828, 337.6536, 340.5515, 343.3979, 346.2167, 349.0343, 351.8804, 354.7884, 357.7955, 360.9437 },

                                { 331.6349, 334.7054, 337.6859, 340.5968, 343.4607, 346.3024, 349.1499, 352.0346, 354.9917, 358.0607, 361.2861 },

                                { 331.6506, 334.7281, 337.7182, 340.6421, 343.5234, 346.3881, 349.2655, 352.1887, 355.1949, 358.3257, 361.6281 } };

boolean blinkLED = false;

int ledState = LOW;

int echo = 12;

int trig = 13;

int led = 10;

const int sega = 2;

const int segb = 3;

const int segc = 4;

const int segd = 5;

#define key0 A0

#define key1 A1

const int sege = 6;

const int segf = 7;

const int segg = 8;

const int segdp = 9;

const int sensor = 11;

DHT dht(sensor, DHT11);

float distance\_cm = 0.0;

  int key = -1;

int mode = 0;

float filter\_buf[FLITER\_N];

void timerIsr() {

  if (blinkLED) {

    ledState = !ledState;

    digitalWrite(led, ledState);

  }

}

void setup() {

  Serial.begin(9600);

  pinMode(echo, INPUT);

  pinMode(trig, OUTPUT);

  pinMode(led, OUTPUT);

  pinMode(sega, OUTPUT);

  pinMode(segb, OUTPUT);

  pinMode(segc, OUTPUT);

  pinMode(segd, OUTPUT);

  pinMode(sege, OUTPUT);

  pinMode(segf, OUTPUT);

  pinMode(segg, OUTPUT);

  pinMode(segdp, OUTPUT);

  pinMode(sensor, INPUT);

  pinMode(key0,INPUT);

  pinMode(key1,INPUT);

  dht.begin();

  Timer1.initialize(50000);

  Timer1.attachInterrupt(timerIsr);

}

void inputData() {

  int i, j;float Hum, Tem, speed;

  for (i = 0; i < FLITER\_N; i++) {

    Hum = dht.readHumidity();

    Tem = dht.readTemperature();

    Serial.print("hum:");

    Serial.print(Hum);

    Serial.print("% Tem:");

    Serial.print(Tem);

    Serial.println("°C");

    Hum = (Hum - 10.0) / 5.0;

    Tem = Tem / 5.0;

    speed = TempHumdTable[(int)Hum][(int)Tem];

    Serial.print("speed:");

    Serial.print(speed);

    Serial.println("m/s");

    digitalWrite(trig, LOW);

    delayMicroseconds(2);

    digitalWrite(trig, HIGH);

    delayMicroseconds(10);

    digitalWrite(trig, LOW);

    distance\_cm = (pulseIn(echo, HIGH) / 10000.0) \* speed / 2.0;

    distance\_cm = (int(distance\_cm \* 100.0)) / 100.0;

    filter\_buf[i] = distance\_cm;

    Serial.print("start:");

    Serial.println(distance\_cm);

    delay(1);

  }

}

float Filter\_Md() {

  int i, j;

  float filter\_temp;

  float Hum, Tem, speed;

  for (j = 0; j < FLITER\_N - 1; j++) {

    for (i = 0; i < FLITER\_N - 1 - j; i++) {

      if (filter\_buf[i] > filter\_buf[i + 1]) {

        filter\_temp = filter\_buf[i];

        filter\_buf[i] = filter\_buf[i + 1];

        filter\_buf[i + 1] = filter\_temp;

      }

    }

  }

  return filter\_buf[(FLITER\_N - 1) / 2];

}

float Filter\_Av() {

  int i, j;

  float filter\_temp=0.0;

  float Hum, Tem, speed;

  for (int i = 0; i < FLITER\_N; i++) {

    filter\_temp += filter\_buf[i];

  }

  return filter\_temp / FLITER\_N;

}

void loop() {

  int n[11][8] = {

    { 0, 0, 1, 1, 1, 1, 1, 1 },

    { 0, 0, 0, 0, 0, 1, 1, 0 },

    { 0, 1, 0, 1, 1, 0, 1, 1 },

    { 0, 1, 0, 0, 1, 1, 1, 1 },

    { 0, 1, 1, 0, 0, 1, 1, 0 },

    { 0, 1, 1, 0, 1, 1, 0, 1 },

    { 0, 1, 1, 1, 1, 1, 0, 1 },

    { 0, 0, 0, 0, 0, 1, 1, 1 },

    { 0, 1, 1, 1, 1, 1, 1, 1 },

    { 0, 1, 1, 0, 1, 1, 1, 1 },

    { 1, 0, 0, 0, 0, 0, 0, 0 },

  };

  int a, b, c, d;

  float distance\_Md, distance\_Av, res;

  inputData();

  distance\_Md = Filter\_Md();

  distance\_Av = Filter\_Av();

  Serial.print("Md:");

  Serial.print(distance\_Md);

  Serial.print(" Av:");

  Serial.println(distance\_Av);

  int distance;

    if(digitalRead(key0) == 1){

      key = 0;

  }

  if(digitalRead(key1) == 1){

      key = 1;

  }

  if (key == 0) distance = int((distance\_Md+0.05)\*10);

  else if (key == 1) distance = int((distance\_Av+0.05)\*10);

  Serial.print("res:");

    if (key == 0) {

      res = distance\_Md;

      Serial.println(res);

    } else if (key == 1) {

      res = distance\_Av;

      Serial.println(res);

    }

  a = distance % 10;

  b = distance / 10 % 10;

  c = distance / 100 % 10;

  d = distance / 1000 % 10;

  if (distance\_cm < 10.00) {

    blinkLED = true;

  } else  {

    blinkLED = false;

    digitalWrite(led, LOW);

  }

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[0][9 - pin]);

  }

  delay(400);

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[d][9 - pin]);

  }

  delay(400);

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[c][9 - pin]);

  }

  delay(400);

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[b][9 - pin]);

  }

  delay(400);

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[10][9 - pin]);

  }

  delay(400);

  for (int pin = 2; pin <= 9; pin++) {

    digitalWrite(pin, n[a][9 - pin]);

  }

  delay(500);

}