上海大学　计算机学院

《单片机技术》自选设计报告

姓名　严昕宇　　学号　20121802

时间　2022.11.6　　指导教师　支小莉老师

单片机应用系统名称:　空气质量综合检测仪

一、预期系统功能

改革开放 40 年来，随着我国经济社会的快速发展和综合国力的显著增强，城乡居民生活水平显著提高。但与此同时，由于对环境保护的不重视，空气质量急剧下降，大气污染防治仍任重而道远。传统的室内环境检测设施实时性差、精度低、体积大、功能不齐全等，难以适应人们的要求。

基于以上背景以及在《单片机技术》课程内外所学到的知识，本文设计了基于单片机的空气质量综合检测仪，它能实时自动地检测与采集空气中 $PM_{2.5}$、PM_{10}、温度与湿度数据，并将处理后的数据传输展示到需要的界面。

二、系统设计

1．硬件设计

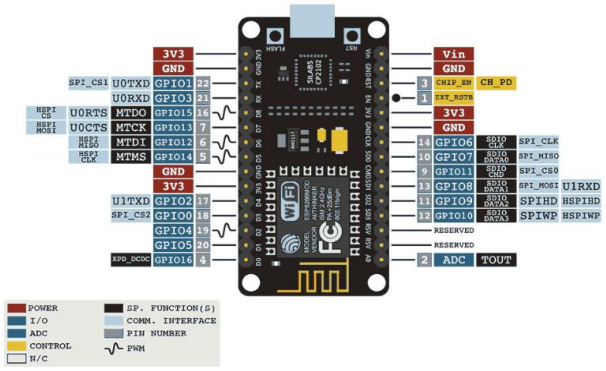
在空气质量综合检测仪的各个部件中，NodeMCU 芯片是整个系统的主控制器，整个检测系统都是在它的控制下完成的，并且为其他设备提供 GPIO 外部接口，如 OLED 显示屏、数据采集设备灯。首先通过 HTU21D 温湿度传感器与攀藤 G5 $PM_{2.5}$ 激光传感器进行温湿度和 $PM_{2.5}$ 浓度信号的采集，再将采集到的信号送至 NodeMCU 进行信号转换、放大和处理。通过基于 Lua 语言的编程，控制整个检测系统的工作流程，并在 OLED 液晶屏上实时显示数据值。

1.1 单片机 NodeMCU

本是基于 ESP8266 的 NodeMCU 进行设计。NodeMCU 是一款集成了 WiFi 功能的 MCU 开发板，可以直接连接 WiFi，开发环境多元化,也是表较受欢迎的物联网芯片。

其主要特点如下：

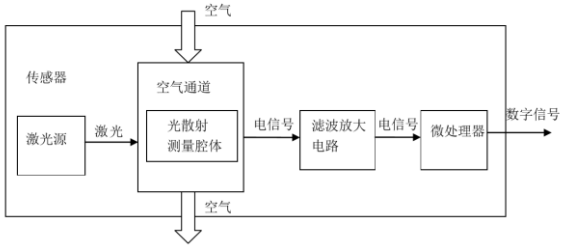
- 像 Arduino 一样操作硬件 IO，即提供硬件的高级接口，可以将应用开发者从繁复的硬件配置、寄存器操作中解放出来。使用交互式 Lua 脚本，编写硬件代码轻松。
- 用 NodeJS 类似语法，极大的方便了用户进行网络应用开发
- 集成了 WIFI 芯片，提供性价比最高的物联网应用开发平台。



1.2 攀藤 G5 PMS5003 $PM_{2.5}$ 激光传感器

本系统采用攀藤科技的 G5 PMS5003 $PM_{2.5}$ 激光传感器。PMS5003 是一款基于激光散射原理的数字式通用颗粒物传感器，可连续采集并计算单位体积内空气中不同粒径的悬浮颗粒物个数，即颗粒物浓度分布，进而换算成为质量浓度，并以通用数字接口形式输出。本传感器可嵌入各种与空气中悬浮颗粒物浓度相关的仪器仪表或环境改善设备，为其提供及时准确的浓度数据，符合本系统的设计要求。。

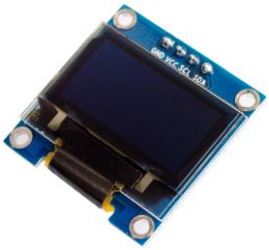
本传感器采用激光散射原理。即令激光照射在空气中的悬浮颗粒物上产生散射，同时在某一特定角度收集散射光，得到散射光强度随时间变化的曲线。进而微处理器基于米氏(MIE)理论的算法，得出颗粒物的等效粒径及单位体积内不同粒径的颗粒物数量。



1.3 HTU21D 温度与湿度传感器

本系统采用的传感器为 HTU21D。HTU21D 是一款新型数字湿度传感器，带温度输出，在规模和智能方面树立了新的标准。该传感器以数字 IIC 格式提供经过校准的线性信号。

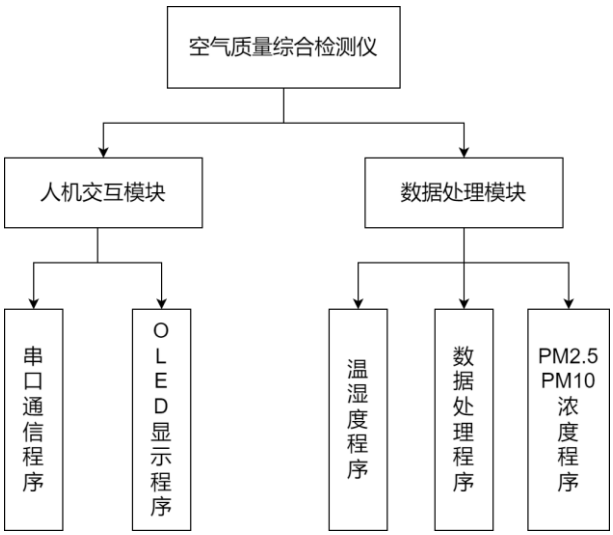
1.4 0.96 寸 OLED 显示屏



OLED，即有机发光二极管(Organic Light Emitting Diode)。OLED 同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、使用温度范围广、构造及制程较简单等优异特性。本系列 OLED 显示屏的内部驱动芯片为 SSD1306，共有 4 种接口的版本，包括:6800、8080 两种并行接口方式、3 线或 4 线的串行 SPI 接口方式、IIC 接口方式本项目中使用基于 IIC 的 OLED 显示屏。

2．软件设计

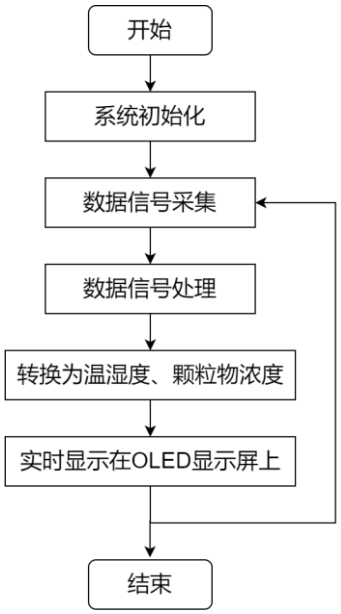
系统主要实现温湿度以及 $PM_{2.5}$ 浓度的检测与显示功能，通过传感器不断采集数据，同时对这些数据进行处理、修正和补偿，将数据转换为温湿度以及颗粒物浓度进行显示，让用户可以知道室内的空气质量。与硬件系统一样分模块设计，软件设计分成人机交互模块和数据处理模块两部分。



系统上电后，先对系统的各模块传感器、OLED 显示屏等外设进行初始化，设置标志位和定时器。之后读取温湿度对应的模拟电流电压等电信号参数值，将数据进行转换，对数字量处理后转换为对应的温湿度和 $PM_{2.5}$ 以及 PM_{10} 浓度显示出来。

2.1 温湿度检测程序设计

温湿度检测程序首先采集传感器反馈的数据，通过固定的计算公式将数据信号转化为真实的温度值，并返回给主程序，用以输出至 OLED 显示屏和网页。



代码如下：

```
-- read humidity from si7021
local function read_humi()
   dataH = read_data(Si7021_ADDR, CMD_MEASURE_HUMIDITY_HOLD, 2)
   UH = string.byte(dataH, 1) * 256 + string.byte(dataH, 2)
--  转换公式
   h = ((UH*12500+65536/2)/65536 - 600)
   UH = nil
   dataH = nil
   return(h)
end

-- read temperature from si7021
local function read_temp()
   dataT = read_data(Si7021_ADDR, CMD_MEASURE_TEMPERATURE_HOLD, 2)
   UT = string.byte(dataT, 1) * 256 + string.byte(dataT, 2)
--  转换公式
   t = ((UT*17572+65536/2)/65536 - 4685)
   UT = nil
   dataT = nil
   return(t)
end
```

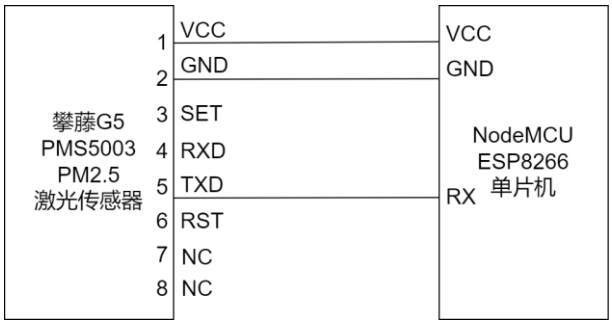
2.2 颗粒物浓度检测程序设计

颗粒物浓度检测程序的设计与温湿度检测程序大致相似，因而在此不过多赘述。

三、系统实现

3.1 硬件实现

由硬件设计可知，系统选用了攀藤 G5 PMS5003 $PM_{2.5}$ 激光传感器传感器，通过该模块可以对颗粒物浓度进行测量，传感器通过 TXD 向 NodeMCU 发送数据，接线图如下所示：



其余模块的连接与之类似，如因为 OLED 显示屏使用的是 IIC 总线，所以将 SCL、SDA 连接至 NodeMCU 的 GPIO 引脚即可，在此不过多赘述。

3.2 软件实现

此部分主要参考了乐为物联的相关代码与开发文档，实现了 OLED 显示屏输出与网页数据显示，此处贴出部分重要代码。

OLED 显示屏初始化代码：

```
-- OLED 显示屏初始化
disp = nil
function init_OLED(sda,scl) --Set up the u8glib lib
      sla = 0x3c
      i2c.setup(0, sda, scl, i2c.SLOW)
      disp = u8g.ssd1306_128x64_i2c(sla)
      disp:setFont(u8g.font_6x10)
      disp:setFontRefHeightExtendedText()
      disp:setDefaultForegroundColor()
      disp:setFontPosTop()
end

init_OLED(5,6) --Run setting up
```

IIC 总线初始化代码：

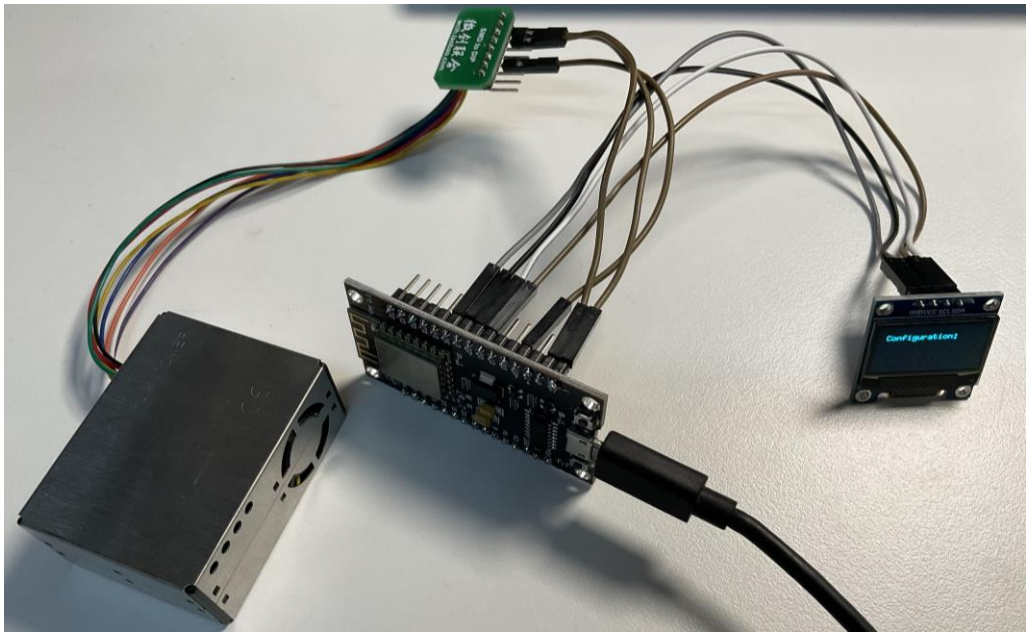
```
-- read data from si7021
-- ADDR: slave address
-- commands: Commands of si7021
-- length: bytes to read
local function read_data(ADDR, commands, length)
  i2c.start(id)
  i2c.address(id, ADDR, i2c.TRANSMITTER)
  i2c.write(id, commands)
  i2c.stop(id)
  i2c.start(id)
  i2c.address(id, ADDR,i2c.RECEIVER)
  tmr.delay(20000)
  c = i2c.read(id, length)
  i2c.stop(id)
  return c
end
-- initialize module
-- sda: SDA pin
-- scl SCL pin
function M.init(sda, scl)
  i2c.setup(id, sda, scl, i2c.SLOW)
  --print("i2c ok..")
  init = true
end
```

主程序部分代码：

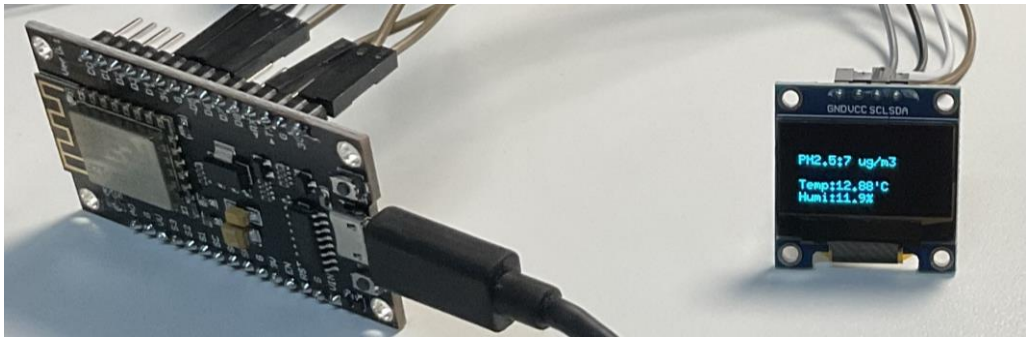
```
disp:firstPage()
repeat
disp:drawFrame(15,15,100,25)
disp:drawStr(25,25,"PM2.5 Detector")
until disp:nextPage() == false

uart.setup( 0, 9600, 8, 0, 1, 0 )
uart.on("data", 0,
 function(data)
      if((string.len(data)==32) and (string.byte(data,1)==0x42) and (string.byte(data,2)==0x4d))
then
            pm25 = (string.byte(data,13)*256+string.byte(data,14))
            --socket:send(pm25..'\n\r')
            local si7021 = require("si7021")

            SDA_PIN = 5 -- sda pin, GPIO14
            SCL_PIN = 6 -- scl pin, GPIO12

            si7021.init(SDA_PIN, SCL_PIN)
            si7021.read(OSS)
            Hum = si7021.getHumidity()
            Temp = si7021.getTemperature()
            --print(Hum)
            --print(Temp)
            -- release module
            si7021 = nil
            _G["si7021"]=nil
             disp:firstPage()
             repeat
                  disp:drawStr(10,20,"PM2.5:"..pm25.." ug/m3")
                  disp:drawStr(10,40,"Temp:"..Temp.."'C")
                  disp:drawStr(10,50,"Humi:"..Hum.."%")
             until disp:nextPage() == false
      end
end, 0)
```

四、系统测试

整体样貌：

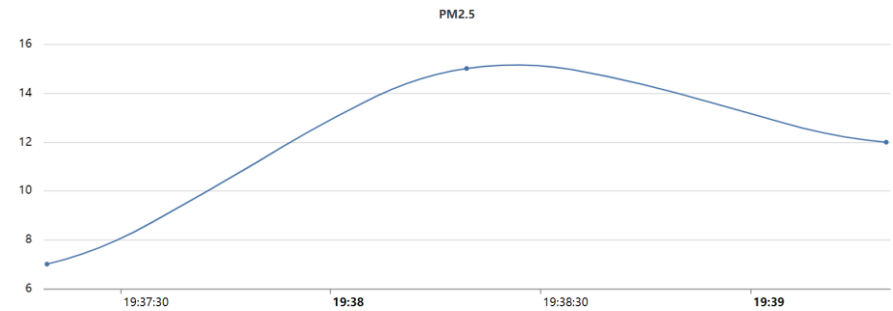


数据显示：



云端显示：

实时统计



状态	设备	名称	最新数值	正常值范围	最后更新	类型	操作
	Air Quality Index	温度	12.88 ℃	-	2022/11/7 19:39:19	温度监控	查看
	Air Quality Index	湿度	11.9 %	-	2022/11/7 19:39:19	湿度监控	查看
	Air Quality Index	PM2.5	12 ug/m3	-	2022/11/7 19:39:19	颗粒物	查看
	Air Quality Index	空气质量	17	-	2022/11/7 19:39:19	其他类型	查看

五、建议和体会

单片机技术是我大学的第三门硬件类专业课。相比数字逻辑、计算机组成原理，单片机技术的相关知识难了不少，但我在其中收获颇丰，并能在最后利用课内课外知识，设计并制作出一个空气质量综合检测仪，对我的日常生活带来帮助。

有人可能会问，学习这课程有什么用呢？在我看来，对于芯片开发或者嵌入式设备开发而言，这部分知识可以为我们提供工程实践的基本理论与实践支持，同时，该门课程也是整个计算机组成原理与体系结构的回顾与实践，掌握好这门课程对于我们从事计算机软硬件方面的工作非常重要。即使将来从事纯软件开发，如果通过实验了解硬件的底层运行原理、工作逻辑，可以设计出更适配的高质量代码，提升效率。

更重要的是，通过本课程的学习，可以掌握不少学习的方法。在单片机技术的实验课程学习过程中，通过同学间互相探讨，我逐步提高了发现问题、分析问题和解决问题的能力；同时，通过撰写这篇报告，我也提高了文字编辑、排版等方面的能力。

当然实验课临近尾声，我想对该课程的授课形式、实验内容提出自己的建议。从一名学生的角度来看，单片机技术这门课带给我从理论走向实践的关键步骤，让我通过在单片机上动手操作，感受计算机的精密美妙。但是，实验课其内容有时与书本知识的结合程度不是最好，有时存在实验内容超前于课本讲授的情况。因此在我看来，如果两者能同步进行，能通过实验课帮助我更好理解单片机技术的原理。而且受限于可课时量，我无法有更多的时间在单片机上操作，这也是一点遗憾。

最后，在此我想感谢支老师的认真教授，帮助我能通过实验，更好掌握课程的知识点。也感谢同学们对我的帮助。