

集成光电器件工程训练

|  |  |  |
| --- | --- | --- |
| 题 | 目： | 基于 LabView 和单片机的温湿度实  时监测网络平台设计 |
| 院 | 系： | 物理与光电工程学院 2019 级光电信息科学与工程  B01914005  苏柯宁  李劲松  集成光电器件工程训练  (2022-2023-1)-SJ32034-13098-1 |
| 专 | 业： |
| 学 | 号： |
| 姓 | 名： |
| 指导教师： | |
| 课程名称 | |
| 课 | 程 号 |

目录

[1. 研究背景及意义 3](#_Toc117506409)

[2. 温湿度传感系统整体设计 3](#_Toc117506410)

[3. 硬件设计及功能说明 3](#_Toc117506411)

[3.1 ESP8266 开发板 3](#_Toc117506412)

[3.2 DHT11 温湿度传感器 4](#_Toc117506413)

[3.2.1 DHT11 的内部测量元件及主要特性 4](#_Toc117506414)

[3.3 TFT 显示屏 5](#_Toc117506415)

[4. 软件设计 6](#_Toc117506416)

[4.1 Arduino IDE 程序 6](#_Toc117506417)

[4.2 LabView 软件和 VISA 串口通讯库函数介绍及程序设计 6](#_Toc117506418)

[5. 温湿度传感系统集成与调试 8](#_Toc117506419)

[5.1 集成过程 8](#_Toc117506420)

[5.2 调试运行 9](#_Toc117506421)

[5.3 WIFI 通讯 10](#_Toc117506422)

[5.3.1 ESP8266 芯片WIFI模块介绍 10](#_Toc117506423)

[5.3.2 WIFI 通讯过程设计 10](#_Toc117506424)

[5.3.4 拓展实验调试运行过程 11](#_Toc117506425)

[6. 实验总结 12](#_Toc117506426)

[7. 参考文献 13](#_Toc117506427)

[8. 附录 13](#_Toc117506428)

[8.1 单片机程序代码 13](#_Toc117506429)

[8.2 网页代码 18](#_Toc117506430)

[8.2.1 HTML 网页代码 18](#_Toc117506431)

[8.2.3 JavaScript 代码 20](#_Toc117506432)

1. 研究背景及意义

温度和湿度的控制对于目前工业生产、农业生产等众多方面都有着重要的意义，不仅限于保障各种机械正常运行，对于保存各种资源来说也是十分重要的。但是人工查看温度计和湿度计已经满足不了目前大规模的生产和仓储需求。随着计算机及虚拟仪器的发展，人工已经逐渐被传感器和软件系统所取代，利用虚拟仪器对常见参数进行测控已经得到了广泛的应用。LabView 本身就是一款模板化的虚拟仪器软件，对于不同的需求可以开发使用不同的模板，使用 LabView 和有着体积小、集成度高且功能强大、价格实惠的 ESP8266 开发板能够进一步降低成本，提高生产效率[1]。

实验设计旨在基于 ESP8266 开发板和 LabView 设计出一套温湿度的实时监测好反馈系统，并要能够查看一段时间内的温湿度变化情况，方便实际情况中不同地点、不同用途中根据温湿度要求进行调整。软件模拟系统，硬件组装系统要可以达到实时显示温湿度数值和波形的目的。

2. 温湿度传感系统整体设计

在目前工农业和仓储等领域智能化的趋势下，针对实际情况中的温湿度控制需求，本课程使用 ESP8266 开发板、DHT11 温湿度传感器和 TFT 显示屏构成硬件系统，使用 LabView 和 Arduino IDE 作为软件系统开展了温湿度实时监控传感系统的设计研究[2]。温湿度由 DHT11 温湿度传感器采集输出，经过开发板处理后将数据提交给 TFT 屏幕进行实时显示，并通过数据线传送至上位机 LabView 中，温湿度实时测量系统硬件和软件相结合整体设计框架流程示意图如图 1 所示。

TFT屏幕显示

ESP8266开发板

接收处理并发送数据

DHT11传感器

监测和发送数据

LabView 进行数据处理和画图显示

串口通讯

图 1.温湿度传感器系统整体结构设计框图

3. 硬件设计及功能说明

3.1 ESP8266 开发板

ESP8266-NodeMCU 是一个开源硬件开发板，如图2所示为 ESP8266-NodeMCU 开发板实物图。由于它支持WIFI功能，所以在物联网(IOT)领域，Arduino 开发板最大的对手之一就是 ESP8266 开发板。ESP8266 尺寸与 Nano 类似, 他并不是 Arduino 团队开发的，但是我们也可以使用 Arduino IDE 对他进行开发。Arduino IDE 编程语言为 C/C++。ESP8266实际为一颗国产的芯片，由于体积小巧，整颗芯片上只有10个 IO 引脚，但是全部都可以进行脉冲宽度调制(PWM)，在进行数据的处理方面并不亚于 Arduino 开发板，并且有着三对3.3V电源脚和地脚，刚好足够本次设计中 TFT 屏幕和温湿度传感器的需求。另外由于 ESP8266 有着WIFI模块,可以根据需要再添加一些功能，因此本次课程设计采用此开发板来完成部分功能。ESP8266-NodeMCU 开发板主要技术参数如表1所列。



图2. ESP8266-NodeMCU 开发板实物

|  |  |
| --- | --- |
| 核心模块 | ESP8266 |
| 工作电压 | 5伏特 |
| 输入电压(推荐) | 5伏特 |
| 输入电压(极限) | 4.5 ～ 10 伏特 |
| 数字输入输出引脚 | 10个(全部可以作为PWM引脚) |
| PWM引脚 | 10个 |
| 模拟输入引脚 | 1个 |
| WIFI标准 | 802.11 b / g / n |
| 工作模式 | STA / AP / STA + AP |
| 重量 | 7g |

表1. ESP8266-NodeMCU 开发板主要技术参数

3.2 DHT11 温湿度传感器

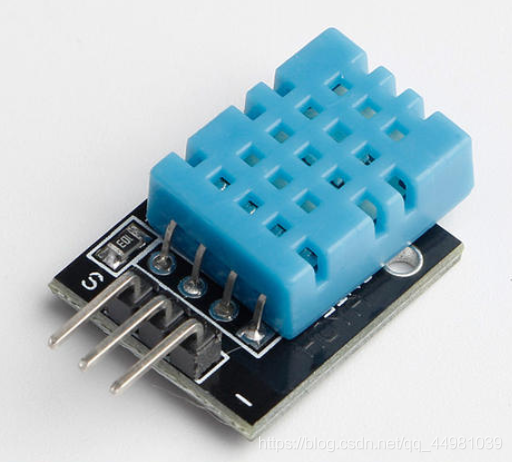
DHT11 温湿度传感器是国产的一款温湿度一体化的数字传感器，实物图如图3所示，该传感器包含了一个电阻式测湿元件和一个 NTC 测温元件。它仅需一个 I/O 口便可以和单片机进行数据通信，而且是一次性将温湿度数据发送，这使得它的应用十分方便，在编写软件程序时非常简单。DHT11 的湿度精度为 ±5% RH， 温度精度为 ±2℃，湿度量程在 5 到 95% RH，温度量程为 -20 到 +60℃，足以应对正常生产生活的需求[3]。

图3. DHT11 温湿度传感器实物图

3.2.1 DHT11 的内部测量元件及主要特性

电阻式湿度传感器是利用湿敏元件的电气特性(如电阻值)，随湿度的变化而变化的原理进行湿度测量的传感器，在湿敏元件的吸湿和脱湿过程中，水分子分解出的离子H+的传导状态发生变化，从而使元件的电阻值随湿度而变化。

NTC 测温元件其实本质为一个热敏电阻，通过检测电阻变化就能得到温度值，且灵敏度较高。

DHT11 的特点还有：

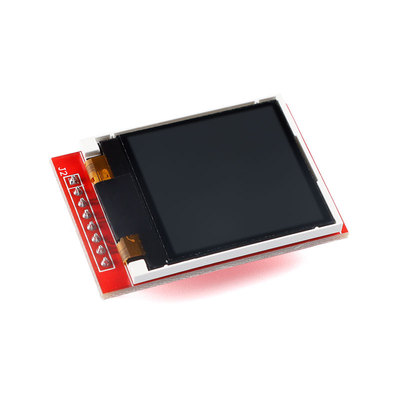
(1)兼具了温度和湿度的检测，对于需要同时测温度和湿度且精度不需要十分严格控制的情况下相较于使用两个器件来说体积更小，成本更低，且不需要额外的 I/O 接口来满足需求。

(2)DHT11 在 5V 供电条件下，运行电流只有 0.5mA，功耗非常低。

(3)在满足使用条件的情况下，DHT11 可以超长时间运行，且可以保证足够的精度，优秀的性能使其有着广泛的应用。

(4)虽然 DHT11 的一次通讯最大时间仅为 3ms ，但是其本身的采样周期为 1s ,因此在本课程设计中最短记录周期为设置的等待清除缓存区时间加上 1 秒。

3.3 TFT 显示屏

TFT(Thin Film Transistor)即薄膜场效应晶体管，它可以对屏幕上的各个独立的像素进行控制，这样可以大大提高反应时间。一般TFT的反应时间比较快，约80毫秒，而且可视角度大，一般可达到130度左右，从而可以做到高速度、高亮度、高对比度显示屏幕信息。TFT属于有源矩阵液晶显示器，在技术上采用了“主动式矩阵”的方式来驱动，方法是利用薄膜技术所制作成的电晶体电极，利用扫描的方法主动控制任意一个显示点的开与关，光源照射时先通过下偏光板向上透出，借助液晶分子传导光线，通过遮光和透光来达到显示的目的[4]。

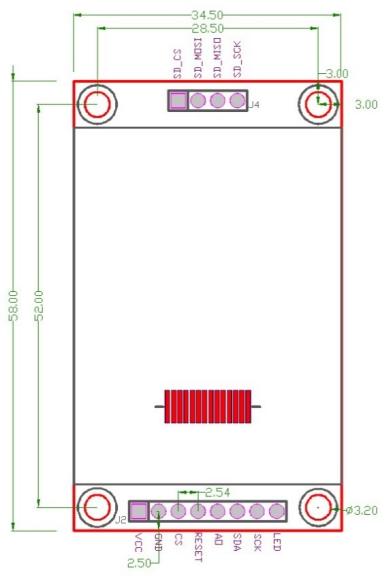
图6. TFT 显示屏实物及尺寸和引脚示意图

图6为TFT显示屏的实物图和引脚定义示意图，从引脚图上可以看到，TFT 屏幕有着两个需要电源的引脚，为 VCC 和 LED ，一个是供电引脚，一个是背光引脚，加上传感器所需要的一个电源引脚，正好是 ESP8266 所拥有的 3.3V 电源脚的数量，这也是本次采用此开发板的原因之一。 SCK 为总线时钟信号、SDA 为总线写数据信号、A0 为液晶屏寄存器与数据选择信号，本次课程设计中设为低电平，选择数据信号，RESET 为重置信号，低电平复位、CS为片选信号，低电平使能，GND 为接地引脚。

对于本次课程设计，由于在屏幕上显示温湿度两条曲线需要不同的颜色区分，而且相较于普通的 OLED 显示屏来说 TFT 屏幕可以显示更多的内容且更加清楚，所以采用了 1.8寸TFT 屏幕来满足需求。

4. 软件设计

4.1 Arduino IDE 程序

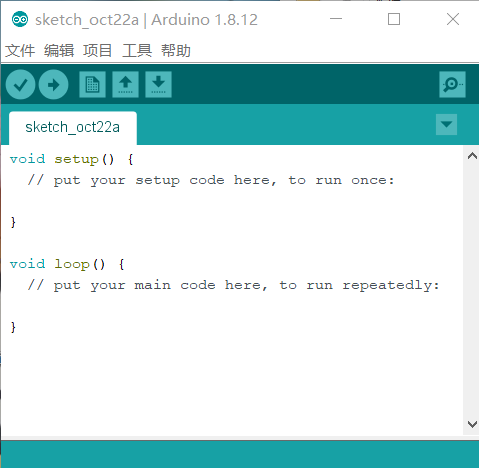
Arduino 系统的单片机和本次课程设计所使用的 ESP8266-NodeMCU 开发板在编程实现功能的时候需要使用到 Arduino IDE 软件作为编程编译以及烧录的平台，如图7所示为Arduino IDE 1.8.12 开发界面。Arduino IDE 可以一键编译烧录，查看串口监视器，还有大量的库函数可以被查询以及下载调用，极大的方便了初学者的上手过程。在本次课程设计中也采用了多个库函数来简化流程，降低了学习成本。代码见附录。Arduino 语言是建立在 C/C++基础上的，以 Setup() 开头，Loop() 循环作为主体的一个程序构架。Setup() 用来初始化变量、调整管脚输入输出模式、设置串口等，此部分只运行一次。Loop() 部分为循环函数，函数内的语句反复执行，为程序主体部分。

图7. Arduino IDE 1.8.12 开发界面

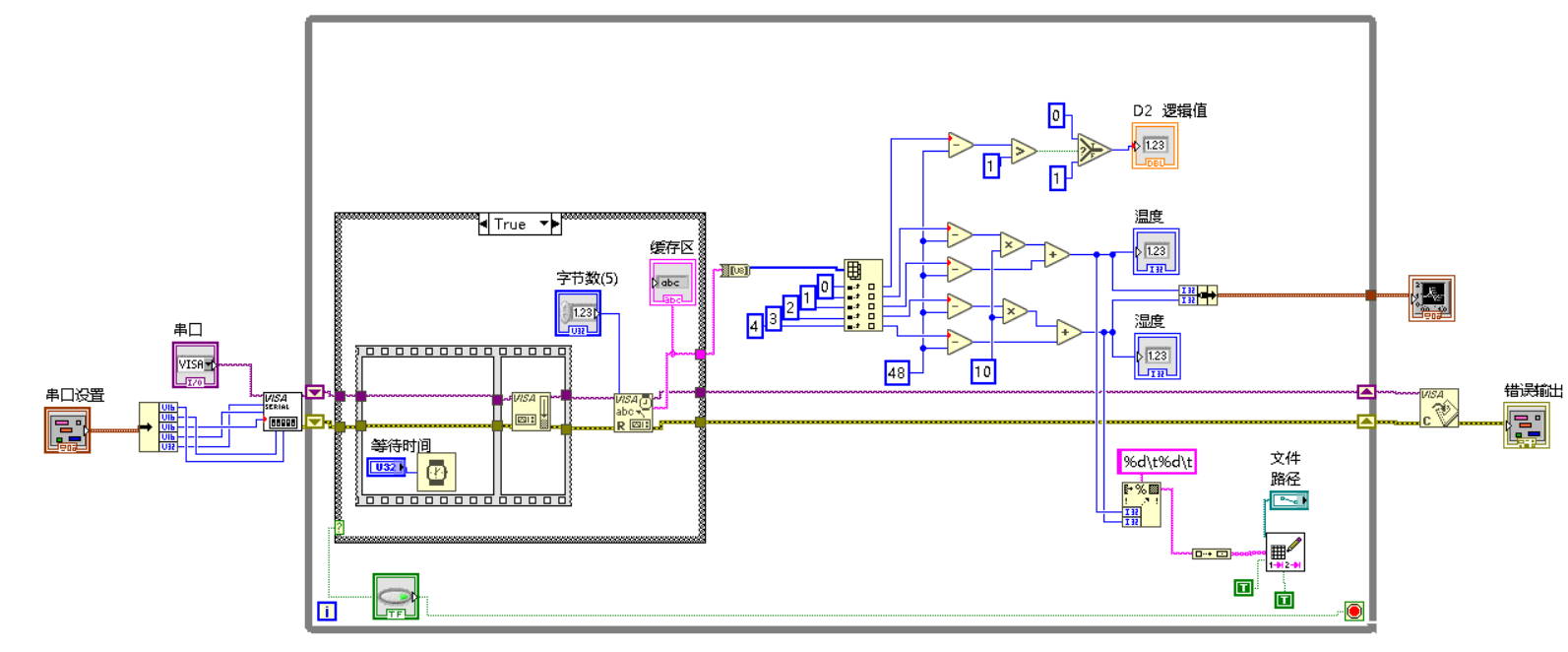
4.2 LabView 软件和 VISA 串口通讯库函数介绍及程序设计

LabView(Laboratory Virtual instrument Engineering Workbench) 是一种图形化的编程语言的开发环境，它广泛地被工业界、学术界和研究实验室所接受，视为一个标准的数据采集和仪器控制软件。它还内置了便于应用 TCP/IP、ActiveX 等软件标准的库函数。它可以增强你构建自己的科学和工程系统的能力，提供了实现仪器编程和数据采集系统的便捷途径。LabView 有一个完成任何编程任务的庞大函数库，包括数据采集、GPIB、串口控制、数据分析、数据显示及数据存储等等。LabView 也有传统的程序调试工具，如设置断点、以动画方式显示数据及其子程序(子VI)的结果、单步执行等等，便于程序的调试[5]。虚拟仪器(virtual instrument)是基于计算机的仪器，是指将仪器装入计算机。以通用的计算机硬件及操作系统为依托，实现各种仪器功能。

1993年，为了确保多厂商的仪器具有协同工作的能力以及降低包含了多厂商仪器的完整测试系统的开发时间，NI 联合许多大公司开发出来了虚拟仪器软件架构 VISA ( Virtual Instruments Software Architecture)。随着 VISA 的出现，使得一套仪器控制程序适用于种硬件接口成为可能，通过调用相同的 VISA 库函数并配置不同的设备参数，就可以编写控制各种 I/O 接口仪器的通用程序，具有很好的兼容性、拓展性和独立性。由于 NI 官网提供给用户方便的库函数管理器( NI Package Manager)以及下载支持，所以用户可以很容易的从官网下载安装 VISA 库函数以便使用(下载地址： [https://www.ni.com/zh-cn/support/downloads/drivers/download.ni-visa.html#460225](https://www.ni.com/zh-cn/support/downloads/drivers/download.ni-visa.html" \l "460225) )。

在 LabView 中使用 VISA 串口通讯仅需要了解四到五个函数：串口配置、串口读取、串口写入以及串口关闭，为了实现数据的过滤，可以在每次读取后选择清除缓存区。实现串口通讯的过程为：(1)串口初始化，利用串口配置函数配置串口的端口号、波特率、校验位、数据位等参数，本次课程设计中仅需对波特率和端口号进行设置。(2)清空缓存区，等待一定时间，从串口读取数据暂存至缓存区。(3)读取缓存区，获得数据。(4)关闭串口，利用 VISA 关闭即可关闭串口，释放串口资源，以便串口被其它程序调用[6]。系统程序设计框图见图 8。实际程序实现流程如下：

首先需要调用串口配置函数完成串口参数的设置，包括串口、波特率，流控制等使用默认设置即可。经过初始化后可使用这个串口进行数据的收发，先判断开关的布尔值，如果为假，则向接下来的程序输出 5个0，并直接关闭串口。如果为真，则使用顺序结构延时并接着清空缓存区。使用读函数来接收串口的数据并按字符串的格式写入缓存区并读取，由于程序设计中只使用串口输出了一个 5 位的数字(在程序设计中将屏幕显示情况作为一位，温湿度各作为两位拼凑为5位数字，并一次性通过串口输入计算机)，因此设置好读取的字节长度为 5。要显示成图像需要拆分数字且转换为整形数字类型，在 LabView 中串口通讯是将读取的数据全部转换为二进制，并以字符串(String)类型显示，但实际在计算机中为 ASCII 码。因此先将字符串即 ASCII 码转换为一个数组，使用拆分数组函数将数组拆为 5个数字，并减去48得到为10进制数字。第一位数字代表着开发板上 D2 脚的布尔值，如果是1，代表目前 TFT 屏幕上是数字显示温湿度，0 代表目前屏幕上是最近 20 秒内温湿度变化曲线图。剩下的 4 位表示温度和湿度，通过乘以 10 并相加来组成两个十位数，将其组合成簇后输入波形图表来显示图像,此处将图表函数放在循环外是为了避免开关关闭时其将设定输入的 5个0 作为数据无限地进行做图。将数据写入表格文件是通过将温湿度数据引入字符串连接，并以 \t 分隔便于导入表格时将温度和湿度分开写入，再次转为数组后输入写入函数，并将转置和添加至文件设为 Ture。

图8. 温湿度传感器监测系统程序框图

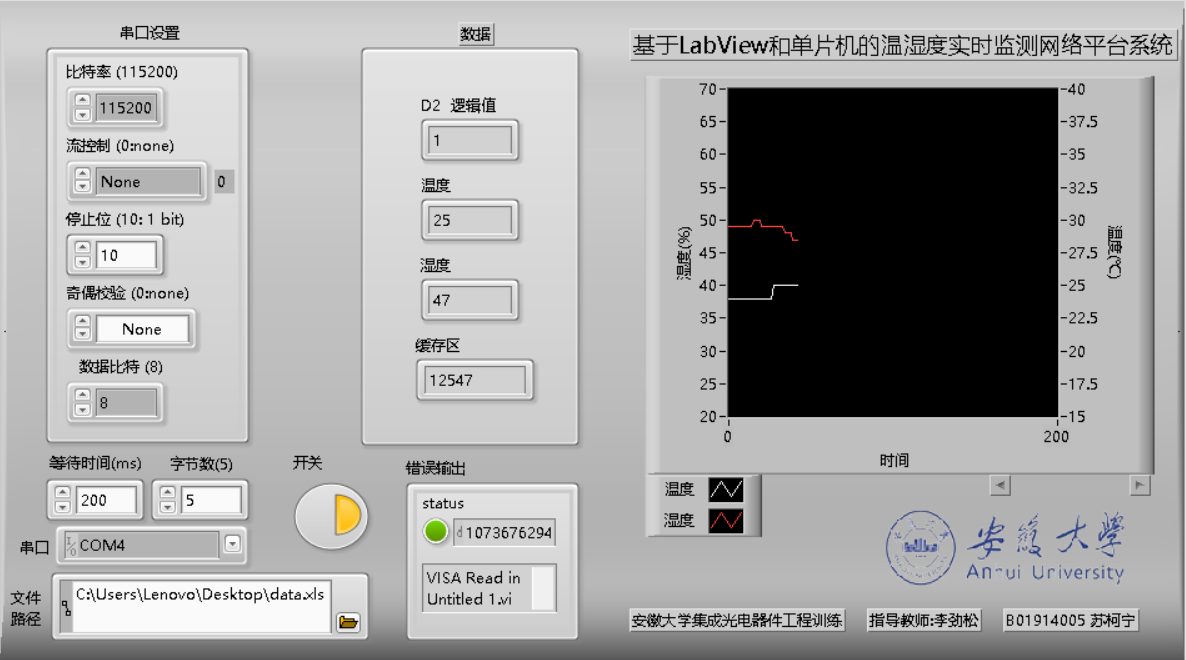
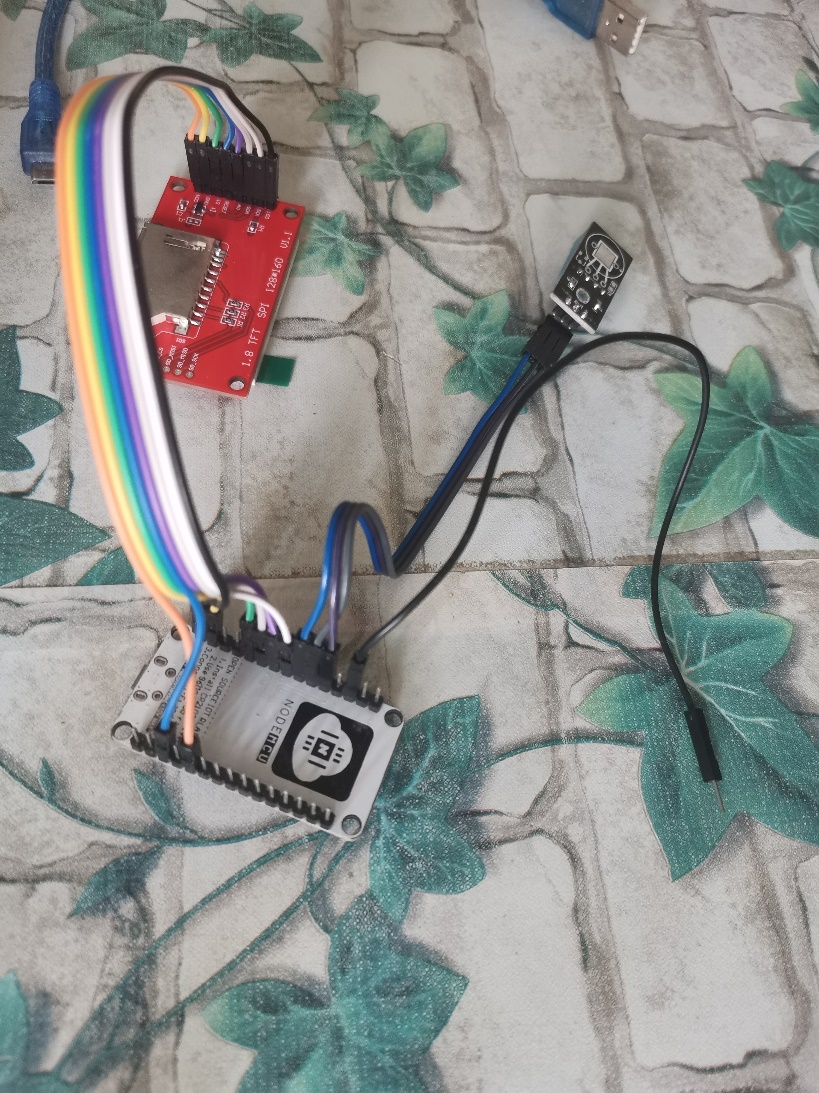
前面板设计如图 9 所示，系统前面板主要包括参数设置、数据显示和图表。需要设置的参数只有串口、循环时间和写入的文件路径，循环时间指从串口读两次数据之间的间隔时长。缓存区为目前读取到的数据，D2 逻辑值和温湿度都是其分解再组合后的数据显示。波形图表能够实时显示温度的变化情况，便于观察。

图9. 系统前面板布局图



5. 温湿度传感系统集成与调试

5.1 集成过程

图10. 温湿度传感器监测系统整体实物连线图

实物连线图如图 10 所示。该部分由 ESP8266 开发板、DHT11 温湿度传感器、 TFT 显示屏构成。在上文中提到的 TFT 屏幕的各个引脚需要连接在开发板上的位置已经提前在 TFT 屏幕的配置函数库中设置完毕，除了特殊的 RESET、LED、VCC、GND 将连接到开发板上具体位置，其他的引脚都可以在库函数中进行配置，本次课程设计将总线时钟信号(SCK)、晶屏数据选择信号(A0)、总线写数据信号(SDA)、片选信号(CS)液分别连接至开发板的 D5 到 D8 四个引脚上。对于 DHT11 温湿度传感器，在程序中将其数据传输接口设置到 D4 引脚，该引脚在开发板上紧挨着一个电源引脚和接地引脚，刚好满足 DHT11 传感器的需求。接着在 D2 接一根杜邦线用于切换屏幕的数字和图像显示，本次课程设计所需的硬件系统全部连接完毕。将 USB 数据线与电脑相连，将程序烧录至单片机中，温湿度传感器 DHT11 将采集到的数据传送给单片机，单片机对接收到的数据进行分析处理，并将数据输入显示屏和 I/O 口，随时更新数据的同时刷新屏幕图像。数据传送到 LabView 中显示且保存。

5.2 调试运行

对于 Arduino IDE 软件中单片机处理系统的代码在完成编写之后，在软件的菜单栏中选择上传，软件会自行进行编译和检查错误并将其烧录至开发板，在开始运行程序之前要先设置好串口号以及数据保存的文件路径，其他选项都使用默认值即可。系统整体运行效果实物图如图 11 所示，开始运行时，波形图表和数值可实时显示温度曲线及具体数值，此时读取间隔为 0.2 秒，传感器采样周期为 1s ，因此记录周期为 1.2s ，D2 的布尔值为 1，代表此时屏幕正处于数字显示状态，温度为 31 摄氏度，相对湿度为 62 %。

图11. 温湿度实时监测传感系统实物图

在进行长时间实验过程中，该系统记录下了从下午6点开始的15个小时内的温湿度数据。图12为使用 Origin 处理过全部数据后得到的图像中的一部分，其中从第100分钟到第350分钟之间为三次开关室内空调后测得的数据，考虑到空调和传感器的误差，测试数据是较为准确的，证明该系统在实际情况中的误差可以控制在1摄氏度和2%湿度之内，可以完全满足部分情况下的使用。在350分钟之后为将湿毛巾放置于传感器附近时系统所记录下的数据，可以看到在较长时间内湿度在不断上升，温度维持在27摄氏度，验证了温度和湿度模块的独立性和稳定性，进一步确认了系统在实际情况中的应用能力。



图 12. 实时测量温度和湿度数据变化曲线

5.3 WIFI 通讯

5.3.1 ESP8266 芯片WIFI模块介绍

ESP8266是一款超低功耗的 UART-WIFI 透传模块，拥有业内极富竞争力的封装尺寸和超低能耗技术，专为移动设备和物联网的应用设计，可将用户的物理设备连接到 WIFI 无线网络上，进行互联网或局域网通信，实现联网功能。透传就是指不需要关心 WIFI 协议是如何实现的,所需要做的就是A通过串口发数据，B通过串口收数据，整个过程中A串口和B串口就像是用导线直接连接起来了一样。则对于开发人员来看，就是完全透明的,这对于开发一些简单的功能来说是极大地简化了开发过程。

ESP8266 WIFI 模块提供了三种模式，分别是接入点(Access Point，AP)模式，无线终端(Wireless Station，WS)模式和混合模式。接入点模式是指 ESP8266 可以自行建立一个局域 WIFI 网络提供给用户连接以进行数据传输，无线终端模式是指 ESP8266 将连接到已经存在的 WIFI 网络中，并将自身作为一个终端来从网络中获取数据，混合模式即两种模式的结合。作为本次课程设计的拓展实验部分，使用接入点模式不仅简化了数据通讯的流程，不必使用已经存在的网络作为中介，而且可以让设备可以在附近没有网络的情况下可以进行正常运作，提高了实用性[7]。

5.3.2 WIFI 通讯过程设计

通讯过程首先是要保持上述实验中 ESP8266 与传感器和显示屏的数据通讯，同时要在这个基础上编写一个简单的网页来给 WIFI 通讯提供显示平台。图13为添加了WIFI通讯过程之后的简化示意图，首先是网络的建立，需要开启 ESP8266 WIFI 模块的 AP 模式，并给定网络名称和连接密码来建立一个 WIFI 局域网，将 WIFI 名称和密码以及连接这个网站所需的 IP 地址和其他运行信息打印在显示屏上，且在运行过程中持续对 80 端口进行监听，该端口是在 HTTP 传输协议中固定的虚拟端口号。然后是网页在用户端的建立，在用户端进行了数据的请求的之后，ESP8266 在从传感器获取到温湿度数据的同时响应请求，将数据先发送至显示屏进行数据更新，然后遍历闪存数据查询用户端请求的数据是否存在，如果有则同时发送数据和其数据类型，否则将返回 404 错误代码，即代表未找到数据，由于本次课程设计中编写的网页是在基本的 HTML 代码基础上调用外部 CSS 样式表和 JavaScript 代码，因此该查询和发送过程会重复多次直至将网页全部建立完毕。其次是数据通讯，用户端发送请求温湿度数据后，会发送一段字符给 ESP8266 ，在检测到该段字符后，单片机将发送一个数据给用户端，用户端接收后会检验异步响应状态码是否为 4，在JS的 XMLHttpRequest API 中4代表异步响应完成，可以进行数据获取，并同时检查网络状态码是否为200，之后用户端会获取数据，并进行处理和显示。

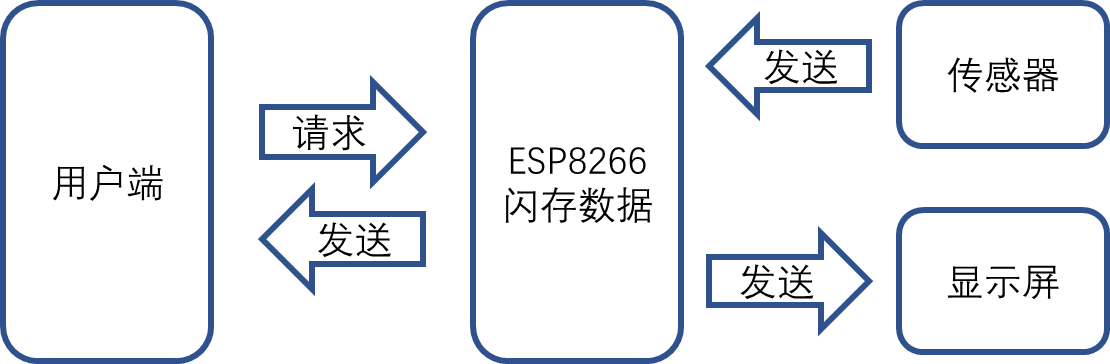


图13. ESP8266 WIFI通讯过程简化示意图

5.3.4 拓展实验调试运行过程

将程序编译烧录到单片机之后，我们需要将网页数据上传到 ESP8266 的闪存中，得益于 ESP8266 闪存工具的支持，我们可以使用 Arduino 直接上传数据到闪存。连接电源，显示屏会显示出 WIFI 局域网的名称、密码、网页IP、闪存系统启动状态以及网络服务器启动状态。图14即为启动后显示屏显示的内容，该页面会显示5秒。

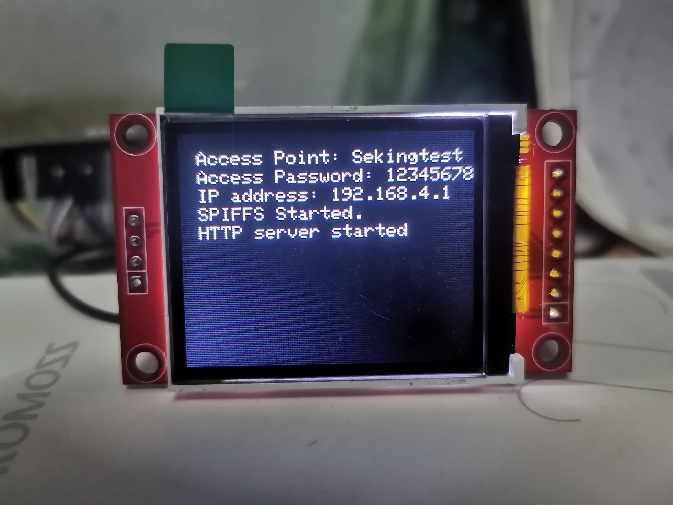


图14. 启动时显示屏内容

SPIFFS，全称为 Serial Peripheral Interface Flash File System ，串行外置接口闪存文件系统，可以使用 FS 库函数来操作 ESP8266 的闪存文件系统。

在用户完成连接后，单片机与其进行数据交换，并按照两秒的间隔开始传输温湿度数据至终端浏览器。图15和图16分别是计算机和手机的接收页面。

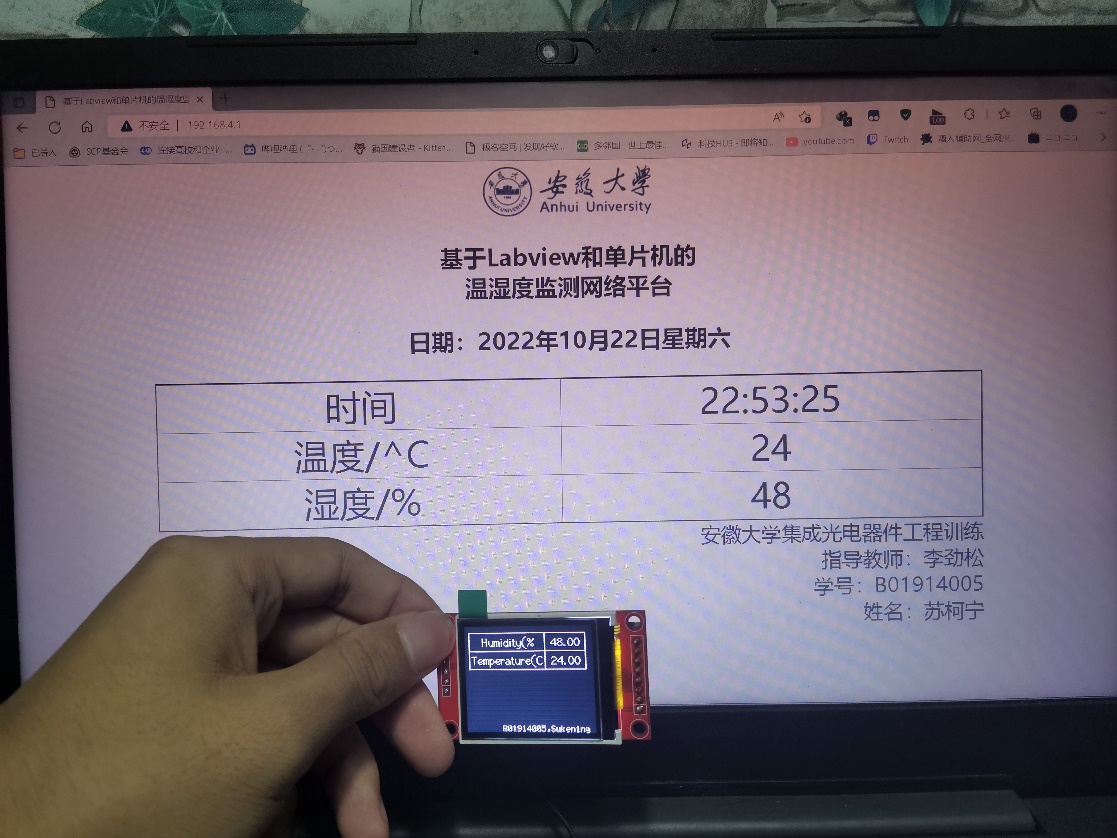


图15. 计算机 Edge 浏览器数据接收页面

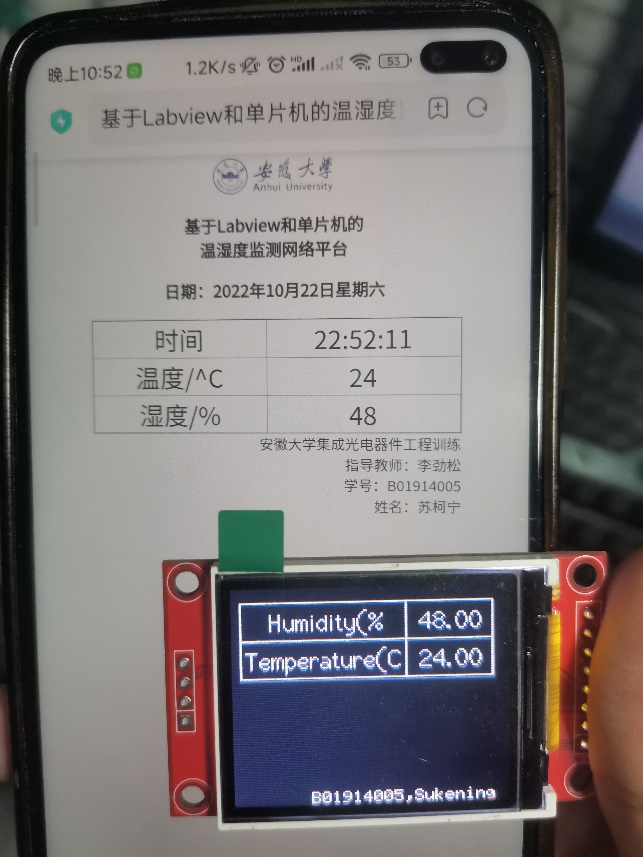


图16. 手机端浏览器数据

从图中可以看到显示屏和浏览器端的数据是同步的，且可以在浏览器端看到实时的日期和时间，便于数据的记录和分析。

# 6. 实验总结

本设计利用 LabView 软件和 ESP8266 开发板作为基础实现了温湿度的实时测量及图像表示和数据保存。使用 ESP8266、DHT11 温湿度传感器及 TFT 显示屏实现了对温湿度的测量和数字及图像的显示；在计算机上使用 LabView 进行了虚拟仪器以及前端的设计。

本次课程设计主要包括两部分，一是硬件部分的组装，将 ESP8266 开发板、DHT11 传感器及 TFT 显示屏进行相应连接，并单独连接一根杜邦线用于改变 D2 引脚的布尔值来改变屏幕的数字和曲线显示。二是软件部分，包括程序的编写和 LabView 虚拟仪器的设计。利用 Arduino IDE 进行程序代码编写后烧录进开发板中，通过数据线连接电脑后通过 LabView 虚拟仪器实现对温湿度数据的处理和显示；使用 VISA 函数库，通过串口初始化、清空缓存区、串口读取及关闭串口实现串口通讯，利用前面板的波形图表、数值显示等控件来显示数据，并实现对温湿度数据的保存。

由于先前的课程设计中接触并使用过 LabView ，所以本次实验中对于新的函数库的学习和使用并没有之前费时费力，同时也在编写代码的过程中复习并熟练了使用 C 语言的能力。本次课程设计为熟练使用各种软件提供了可贵的学习和实践机会，对于日后的学习工作有着很大的帮助。

# 7. 参考文献

1. 周新淳. 基于物联网的智能大棚数据采集系统设计[J]. 电子世界,2017(06):103-104.
2. Chananchai Wutthithanyawat and Nawadee Srisiriwat. Process Control Virtual Laboratory: Temperature Control Based on LabVIEW for Convection Heat Transfer[J]. Applied Mechanics and Materials, 2014, 3380(619-619):303-308.
3. 赵强,石富文.Arduino库的编写实例:制作基于DHT11的温湿度计[J].电子制作,2020(13):65-68+71.
4. 常爱珍,李宝辉,张凯.TFT-LCD显示原理及评判参数[J].汽车电器,2021(04):31-33.
5. 周细凤,谢波,张平华.基于 LabView 的智能温湿度检测系统[J].湖南工程学院学报(自然科学版),2018,28(01):6-9.
6. Ccalli Pacco Honorato. Temperature simulation and control in refrigerated storage chambers for fruit[J]. Procedia Manufacturing, 2020, 42(C) : 35-40.
7. 张恒强,安霆,王乙涵,胡傲,刘文静.基于ESP8266的物联网技术应用研究[J].仪表技术,2022(03):26-29.

# 8. 附录

8.1 单片机程序代码

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <FS.h>

#include <TFT\_eSPI.h>

#include <SPI.h>

#include <dht11.h>

dht11 DHT11;

#define DHT11PIN 2

TFT\_eSPI tft = TFT\_eSPI();

int temperaturedata[20];

int humiditydata[20];

int i=0,j=0,value0,value1,x;

const char \*ssid = "Sekingtest";

const char \*password = "12345678";

ESP8266WebServer esp8266\_server(80);

void setup() {

Serial.begin(115200);

tft.init();

tft.setRotation(1);

tft.fillScreen(TFT\_BLACK);

tft.setTextColor(TFT\_WHITE, TFT\_BLACK);

pinMode(4,INPUT);

WiFi.softAP(ssid, password);

tft.setCursor(10,10);

tft.print("Access Point: ");

tft.print(ssid);

tft.setCursor(10,20);

tft.print("Access Password: ");

tft.print(password);

tft.setCursor(10,30);

tft.print("IP address: ");

tft.print(WiFi.softAPIP());

tft.setCursor(10,40);

if(SPIFFS.begin()){

tft.print("SPIFFS Started.");

} else {

tft.print("SPIFFS Failed to Start.");

}

esp8266\_server.on("/getDHT",handleDHT);

esp8266\_server.onNotFound(handleUserRequet);

esp8266\_server.begin();

tft.setCursor(10,50);

tft.print("HTTP server started");

delay(5000);

}

void loop(void) {

if(digitalRead(4) == HIGH){

tft.fillScreen(TFT\_BLACK);

tft.drawLine(5,8,150,8,TFT\_WHITE);

tft.drawLine(5,30,150,30,TFT\_WHITE);

tft.drawLine(5,52,150,52,TFT\_WHITE);

tft.drawLine(5,8,5,52,TFT\_WHITE);

tft.drawLine(100,8,100,52,TFT\_WHITE);

tft.drawLine(150,8,150,52,TFT\_WHITE);

tft.setTextFont(1);

tft.setCursor(45,120);

tft.print("B01914005,Sukening");

tft.setTextFont(2);

tft.setCursor(22,12);

tft.print("Humidity(%");

tft.setCursor(8,34);

tft.print("Temperature(C");

while(1){

if(digitalRead(4) == LOW){break;}

DHT11.read(DHT11PIN);

tft.setCursor(108,12);

tft.print((float)DHT11.humidity,2);

tft.setCursor(108,34);

tft.print((float)DHT11.temperature,2);

if(i < 20){

temperaturedata[i]=DHT11.temperature;

humiditydata[i]=DHT11.humidity;

i++;

Serial.print(DHT11.temperature\*100+DHT11.humidity+digitalRead(4)\*10000);

}

else{

for(i=0;i<19;i++){

temperaturedata[i]=temperaturedata[i+1];

humiditydata[i]=humiditydata[i+1];

}

temperaturedata[i]=DHT11.temperature;

humiditydata[i]=DHT11.humidity;

Serial.print(DHT11.temperature\*100+DHT11.humidity+digitalRead(4)\*10000);

}

esp8266\_server.handleClient();

delay(1000);

}

}

else{

while(digitalRead(4) == LOW){

tft.fillScreen(TFT\_BLACK);

tft.setTextFont(1);

tft.setCursor(40,120);

tft.print("B01914005,Sukening");

tft.setTextFont(2);

tft.setCursor(20,106);

tft.print("0");

tft.setCursor(70,106);

tft.print("10");

tft.setCursor(122,106);

tft.print("20");

tft.drawLine(20,6,20,106,TFT\_WHITE);

tft.drawLine(80,106,80,101,TFT\_WHITE);

tft.drawLine(20,106,140,106,TFT\_WHITE);

tft.drawLine(20,6,140,6,TFT\_WHITE);

tft.drawLine(140,6,140,106,TFT\_YELLOW);

tft.setCursor(2,101);

tft.print(temperaturedata[0]-10);

tft.setCursor(2,1);

tft.print(temperaturedata[0]+10);

tft.setCursor(2,51);

tft.print(temperaturedata[0]);

tft.setCursor(142,106-((humiditydata[i-1]-20)\*2));

tft.print(humiditydata[i-1]);

tft.setCursor(142,101);

tft.print(20);

tft.setCursor(142,1);

tft.print(70);

for(j=6;j<110;j=j+5){

tft.drawLine(20,j,25,j,TFT\_WHITE);

tft.drawLine(135,j,140,j,TFT\_YELLOW);

}

tft.drawLine(25,56,31,106-((temperaturedata[1]+10-temperaturedata[0])\*5),TFT\_WHITE);

tft.drawLine(25,106-((humiditydata[0]-20)\*2),31,106-((humiditydata[1]-20)\*2),TFT\_YELLOW);

if(digitalRead(4) == HIGH){break;}

DHT11.read(DHT11PIN);

for(j=1,x=25;j<i-1;){

value0 = 106-((temperaturedata[j]+10-temperaturedata[j-1])\*5);

j++;

value1 = 106-((temperaturedata[j]+10-temperaturedata[j-1])\*5);

tft.drawLine(x,value0,x+6,value1,TFT\_WHITE);

j--;

value0 = 106-((humiditydata[j]-20)\*2);

j++;

value1 = 106-((humiditydata[j]-20)\*2);

tft.drawLine(x,value0,x+6,value1,TFT\_YELLOW);

x+=6;

}

if(i<20){

temperaturedata[i]=DHT11.temperature;

humiditydata[i]=DHT11.humidity;

i++;

Serial.print(DHT11.temperature\*100+DHT11.humidity+(digitalRead(4)+2)\*10000);

esp8266\_server.handleClient();

delay(1000);

}

else{

for(i=0;i<19;i++){

temperaturedata[i]=temperaturedata[i+1];

humiditydata[i]=humiditydata[i+1];

}

temperaturedata[i]=DHT11.temperature;

humiditydata[i]=DHT11.humidity;

i++;

Serial.print(DHT11.temperature\*100+DHT11.humidity+(digitalRead(4)+2)\*10000);

esp8266\_server.handleClient();

delay(1000);

}

}

}

}

void handleDHT(){

esp8266\_server.send(200, "text/plain", String(DHT11.temperature\*100+DHT11.humidity));}

void handleUserRequet() {

String webResource = esp8266\_server.uri();

bool fileReadOK = handleFileRead(webResource);

if (!fileReadOK){

esp8266\_server.send(404, "text/plain", "404 Not Found");

}

}

bool handleFileRead(String path) {

if (path.endsWith("/")) {

path = "/index.html";

}

String contentType = getContentType(path);

if (SPIFFS.exists(path)) {

File file = SPIFFS.open(path, "r");

esp8266\_server.streamFile(file, contentType);

file.close();

return true;

}

return false;

}

String getContentType(String filename){

if(filename.endsWith(".htm")) return "text/html";

else if(filename.endsWith(".html")) return "text/html";

else if(filename.endsWith(".css")) return "text/css";

else if(filename.endsWith(".js")) return "application/javascript";

else if(filename.endsWith(".png")) return "image/png";

return "text/plain";

}

8.2 网页代码

8.2.1 HTML 网页代码

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="main.css">

<title>基于Labview和单片机的温湿度监测网络平台</title>

<script type="text/javascript" src="clock.js"></script>

</head>

<body>

<center>

<a href="https://www.ahu.edu.cn/" target="\_black"><img src="/img/AHU.png" alt="安徽大学"></a>

<h2>基于Labview和单片机的<br>温湿度监测网络平台</h2>

<h3 id="date"></h3>

<script>

setInterval(function() {

var DHT=GetDHT();

document.getElementById("DHTT").innerHTML=(DHT-(DHT%100))/100;

document.getElementById("DHTH").innerHTML=DHT%100;

},2000);

function GetDHT() {

var xhttp = new XMLHttpRequest();

var DHT=0;

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

DHT = Number(this.responseText);

console.log(DHT);

}

};

xhttp.open("GET", "/getDHT", false);

xhttp.send();

return DHT;

}

</script>

<table border="2" >

<caption>安徽大学集成光电器件工程训练<br>指导教师：李劲松<br>学号：B01914005<br>姓名：苏柯宁</caption>

<tr>

<td>时间</td>

<td>

<span id="msg"></span>

</td>

</tr>

<tr>

<td>温度/^C</td>

<td id="DHTT">temperature</td>

</tr>

<tr>

<td>湿度/%</td>

<td id="DHTH"> humidty</td>

</tr>

</table>

</center>

</body>

</html>

8.2.2 CSS 外置样式表代码

body {

font-family: sans-serif;

color: #444;

}

table{

border-collapse: collapse;

width: 75%;

}

table, td, th {

boder: 5PX soild black;

font-size: 175%;

text-align: center;

}

td{

height: 10%;

}

caption {

caption-side: bottom;

text-align: right;

font-size: 100%;

}

h3, h2, a{

font-size: 200%;

}

center{

margin-top: auto;

margin-bottom: auto;

}

8.2.3 JavaScript 代码

function showDateTime(){

var sWeek=new Array("日","一","二","三","四","五","六");

var myDate=new Date();

var sYear=myDate.getFullYear();

var sMonth=myDate.getMonth()+1;

var sDate=myDate.getDate();

var sDay=sWeek[myDate.getDay()];

var h=myDate.getHours();

var m=myDate.getMinutes();

var s=myDate.getSeconds();

document.getElementById("date").innerHTML=("日期：+sYear+"年"+sMonth+"月"+sDate+"日"+"星期"+sDay+"<br/>");

h = formatTwoDigits(h);

m = formatTwoDigits(m);

s = formatTwoDigits(s);

document.getElementById("msg").innerHTML=(h+":"+m+":"+s+"<br/>");

setTimeout("showDateTime()",1000);

}

window.onload=showDateTime;

function formatTwoDigits(s) {

if (s<10)

return "0"+s;

else

return s;

}