基于移动终端的温湿度调控系统

**摘要**

随着工农业生产、家庭生活的电气化水平不断提高，越来越多的电气化设备进入我们的生活圈内，对这些设备的检测、控制成为生产生活中的一个重要环节。对于一般的检测、控制操作，虽然能够在数据获取方面数字化，却需要将获得的数据信息反馈到控制人员手中，有控制人员进行决定，今儿操纵设备。从这个过程可以发现，控制过程不仅对人力依靠大，还容易造成很大的数据延迟，如果是室内的火灾监测、处理控制，容易因为人员的不到位导致延误，产生经济损失。

随着计算机网络和嵌入式技术的发展，我们完全可以通过嵌入式设备完成数据采集、数据上传、数据分析、设备控制的一系列操作。这样，不仅能够及时根据状态的变化及时采取措施，还可以通过网络传输将数据信息分享到不同的设备平台，实现数据的实时分享、设备的远程监控。这样，不仅能够及时的对设备进行控制，还可以将所有的设备、传感网络联系起来，通过统一的平台进行监视、控制，减少人力指出、提高控制效率。

本文讲述的就是通过arduino硬件环境进行数据获取、网络上传至服务器平台、完成设备控制的实现方法。

**关键词 :**控制技术 嵌入式控制 数据传输 信号处理 传感器

**目录**

**绪论.....................................................3**

**第一章 系统介绍**

1.1系统总述

1.1.1流程..............................................................................................................................**4**

1.1.2网络传输......................................................................................................................**5**

1.1.3多平台监控..................................................................................................................**6**

1.2模块介绍

1.2.1传感模块......................................................................................................................**8**

1.2.2调控设备......................................................................................................................**9**

1.2.3控制系统......................................................................................................................**10**

1.2.4设备保护......................................................................................................................**11**

1.2.5网络传输......................................................................................................................**11**

**第二章 系统应用**

2.1优势条件

2.1.1平台广泛......................................................................................................................**12**

2.1.2自动与手动结合..........................................................................................................**12**

2.1.3系统应用成本较低......................................................................................................**12**

2.2可用领域

2.2.1工农业生产..................................................................................................................**13**

2.2.2居民家庭生活..............................................................................................................**13**

**第三章 自主改进**

3.1服务器建设

3.1.1操作系统......................................................................................................................**14**

3.1.2服务软件......................................................................................................................**14**

3.1.3数据库设置..................................................................................................................**14**

3.1.4数据服务......................................................................................................................**14**

3.1.5数据上传......................................................................................................................**15**

3.2客户端开发

3.2.1设计..............................................................................................................................**15**

**绪论**

本文介绍的是一套能够进行数据感知、分析控制、网络共享、远程控制的控制系统。通过对不同嵌入式硬件设备的了解，我们采用了易于编程的arduino开源平台，采用多个开发板，分别完成数据采集、分析控制、数据上传的功能。为了能够模拟控制的效果，我们制作了一个仿真的室内环境，设计了一套降温装置、一套加温装置，通过电路驱动完成一系列控制进程。

参照现有的技术资料和应用经验，我们采用了高精度的DS18B20数字传感器、DHT11数字温湿度传感器进行数据采集；采用传感器冗余和传感器网络提高数据可靠性和稳定性。受知识水平和硬件实力的限制，我们采用的是开放网络共享平台Yeelink，作为网络数据存储和分享的终端。Arduino获得的数据信息上传到Yeelink的数据库后，可以通过平台配套的Android和iOS客户端进行远程查看、控制。

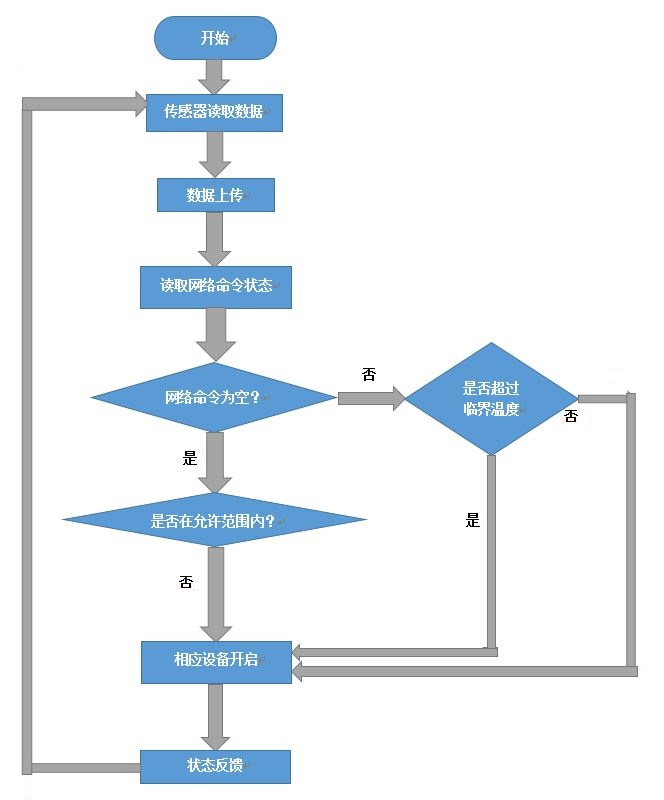
但是，受制于人并不是长远之计，在网络安全备受重视的今天，如果单纯的把设计关键位置的信息和设备控制权交给一个不完全可靠的机构，是完全无法接受的。为了破除这种限制，我们现在正在自己搭建的数据库服务器上进行自己的数据存储和传输实验，相信过不了多久，就能够完成真正意义上的自主研发。

第一章 **系统介绍**

1.1系统总体介绍

1.1.1流程

系统的处理顺序，可以概括为 **图1.1**的流程图。



**图1.1** **流程图**

系统的整个运行过程为不间断循环结构：首先，对部署的传感器网络进行数据获取，得到较为可靠的相关数据，通过网络模块传送到平台的服务器，同时连接服务器，获取用户的命令状态，如果用户进行了强制命令操作，则进行读取；如果用户没有对设备进行强制的命令操作，则根据进行之前设定好的参数进行自主控制，判断数据范围，进行设备控制条件判断，下达控制命令，交给执行部分执行控制代码，或者直接进入下一个循环状态。如果用户对特定设备进行了强制性的命令操作，则首先判断当前是否处于可控状态，目的是保证不超过安全极限，如果超过了安全极限，不仅不要执行用户的命令，还要反向执行，采取控制使状态回到安全范围，进入下个循环；如果在安全范围内，可以直接下达控制的命令，由执行部分完成操作，命令下达后，进入下个循环。控制过程完毕。

1.1.2网络传输

数据的网络共享功能，是基于Yeelink物联网平台的API接口完成的。具体的上传步骤见**表1**中的代码结构，通过发送http请求，与Yeelink服务器取得上传资格，进而确认用户身份，发送设备ID、传感器ID、传感器数据等信息，完成会话任务，数据上传结束。

借助Yeelink平台的API接口，可以在Android和iOS设备上获取服务器数据，在终端中显示当前的传感器信息；通过简单的交互按钮，也可以进行设备的远程控制。

**表1** **数据发送步骤**

void sendData(int thisData) {//数据发送过程

if (client.connect(server, 80)) {//判断链接是否可用

// 开始发送http请求

client.print("POST /v1.0/device/");

client.print(DEVICEID);client.print("/sensor/");client.print(SENSOR\_H);

client.print("/datapoints");

client.println(" HTTP/1.1");

client.println("Host: api.yeelink.net");

client.print("Accept: \*");client.print("/");client.println("\*");client.print("U-ApiKey: ");

client.println(APIKEY);//身份验证密钥

client.print("Content-Length: ");//确定数据长度

int thisLength = 10 + getLength(thisData);//数据长度求解

client.println(thisLength);//发送长度

client.println("Content-Type: application/x-www-form-urlencoded");

client.println("Connection: close");

client.println();

// 真正发送的信息

client.print("{\"value\":");

client.print(thisData);//发送的数据

client.println("}");

}

else {// 如果连接失败

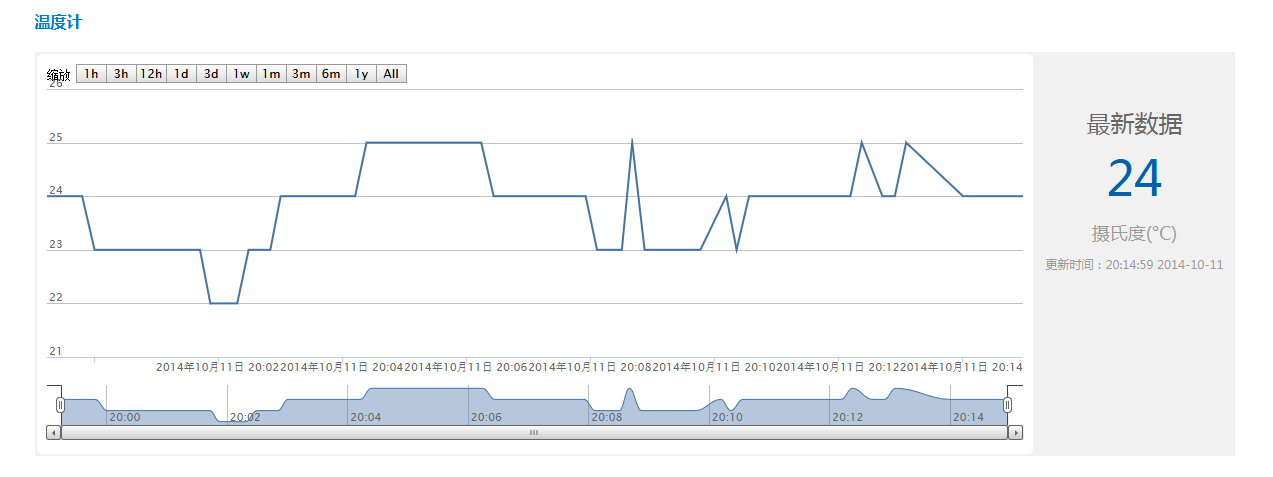
client.stop();//停止

}

}

1.1.3多平台监控

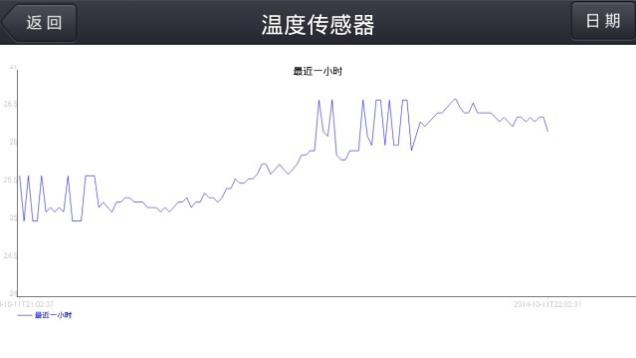
借助网络服务器接口，Yeelink提供了电脑浏览器、手机客户端分别对传感器进行监视，对设备进行控制的方式，对于小型的应用环境来说，基本可以满足安全性、实用性的需求。各个平台的界面效果如下：



**图1.2 电脑网页查看**



**图1.3 电脑网页控制**



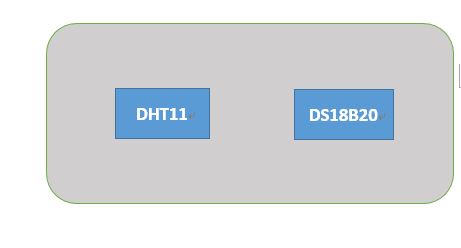
**图1.4 手机设备界面 图1.4 手机查看界面**

1.2模块介绍

1.2.1传感模块

为获取设定环境内的温湿度数据，我们采用了DHT11和DS18B20两种传感器组成的传感器冗余节点，以及多个单总线DS18B20传感器组成的传感器网路作为数据源。

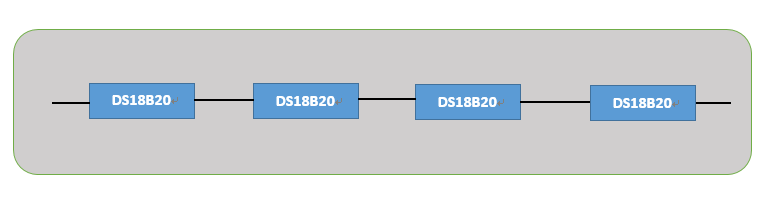
传感器冗余节点，结构如**图1.5**，是由一个整数型DHT11数字温湿度传感器和一个高精度DS18B20温度传感器共同组成的。DHT11数字温湿度传感器可以一次获得一个8bit温度数据和一个8bit湿度数据，精度均为1，在我们的实验初期发现，该传感器获得的温度数据基本可以保持在1℃—2℃的精度范围内，对控制应用来说，范围过于宽泛，很难作为可靠来源，因此，我们后期添加了DS18B20数字温度传感器作为温度数据源。



**图1.5 传感器冗余**

DS18B20数字温度传感器可以根据要求精确到不同的范围，在试验中，精确到小数点后三位的温度数据已经足够大多数条件下的温度精度要求。故我们以此型传感器的数据作为主要判断依据，以DHT11传感器的数据作为参考，以应对传感器受到干扰时导致的精度变化。

DS18B20温度传感器作为单总线设备，理论上可以在一条数据线上部署多个传感器节点，通过对内部MAC的区分，可以一次获得不同位置的温度数据。对于较大型空间的内部温度测量，可以在不同距离处设立传感器节点，最后通过总线获取数据，完成精确地环境分析。传感器线路类似计算机网络拓扑中的总线结构，示意图如 **图1.6**。



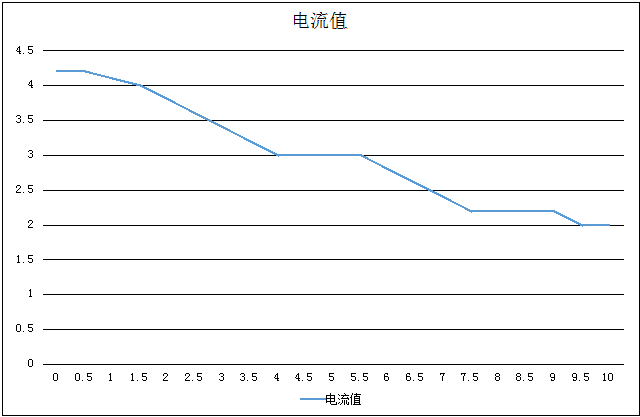
**图1.6 单总线传感器网络**

1.2.2调控设备

为了方便的进行环境模拟，我们制作了一套降温装置，一套增温装置。

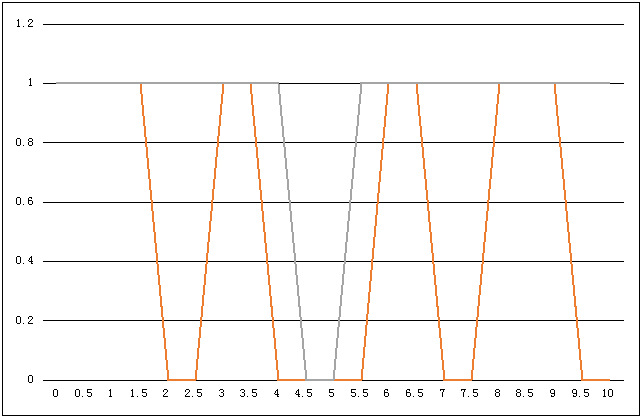
温控设备的原理，是半导体制冷技术，通过半导体片上的P\N电子流，在不同极性上产生不同的温度变化效果。我们采用温度变化显著的TEC-12706半导体制冷贴片作为热源和冷源，通过铝制散热片进行风冷进行温度传递。

在试验中，半导体贴片的工作效率受到半导体贴片本身温度的影响，所以，单个半导体贴片的持续工作效率在开始一段时间能够保持较高的水平，但是随着温度的升高，效率反而会达到一个很低的水平（如**表2**），不仅仅不能完成降温的目的，反倒会带来温度的升高。



**表2 TEC-12706电流随时间的变化规律**

为了对半导体贴片进行有效的利用，同时避免因温度过高导致设备损坏，我们设计采用两片叠加、分时工作的方式综合利用半导体贴片。具体的电流与时间关系如**表3**



**表3 TEC-12706贴片通电顺序**

1.2.3控制系统

控制系统获取传感器检测到的数据，进行比较判断和数学计算，判断变化趋势，得到进行设备控制的最佳时间点。

控制系统根据传感器得到的数据流，计算温度变化幅度和变化速率，如果变化速率以一个非常细微的速率进行，则可以等到温度变化到超出设定范围后，进行小程度的温度补偿。如果温度速率变化较大，则进行不间断地温度补偿，直到超过设定范围一部分。

（1）设备进行自动运行前，需要对温控设备及环境自然温度变化情况进行测试，得到温控设备开启时的温度变化率ΔT0，密闭环境下的温度自然变化率ΔT1。

（2）确定环境温度的设定范围 T1 - T2。

（3）确定环境温度的最大可接受范围 T3 - T4。

（4）当读取到一个温度数据时，与之前的温度数据比较，如果上次数据为空，则直接与 T1 - T2 比较，如果在范围内，则不再进行操作，直接进行下一步操作；如果不在范围内，则与最大可接受范围 T3 - T4 进行比较，如果在范围内，则暂时搁置，如果不在范围内，则直接开启控制。如果上次数据不为空，则与上次数据比较，由ΔT = T末 – T初 得到温度变化幅度，由f = ΔT / Δt （Δt为两次检测时间差）得到温度变化率。

(5)进行计算 f \*Δt + T末 是否在 T1 - T2 中，如果在，可以跳过；如果不在范围内，计算 f \*Δt + T末 + (ΔT0 +ΔT1) \*Δt ，如果在 T1 - T2 内，可以开启[(f \*Δt + T末)- (T1 ~ T2)] /ΔT0 时间的控制。

1.2.4设备保护

由于实际应用过程中，有些控制设备在运行时会产生较高的温度或者电压，超过设备的限定条件时，会对设备产生不可逆损害，所以，必须要有一套独立运行的系统进行设备的检测保护。

在我们的实例中，需要对散热片的温度进行实时监测，如果散热片温度超过一般的正常值，需要发出声、光等警示信息，同时如果可以的话，直接切断设备电源，结束运行状态，经过网络发送信息到平台，提示设备强制关闭。

1.2.5网络传输

Arduino硬件平台配套有Ethernet WR5100 Shield拓展板，经过与Arduino 的ISP接口连接，可以进行网路传输。由于拓展板对Arduino相对透明，在程序设计时可以直接进行命令操作而不必考虑设备信息。

根据不同的网络环境，可以有多种网络接入方式。当有线网络应用没有问题时，可以将所有的网络拓展板接入到有线网络；如果有线网络布局比较困难，可以使用NRF24L01无线网络传输模块作为子系统部分与网络交互部分的中间传输途径，通过这种方式，可以实现更大范围内的传感器布局。

第二章 **系统应用**

2.1优势条件

2.1.1平台广泛

借助Yeelink的互联网平台，不管是电脑的网页，还是Android的手机移动客户端，都能够完成信息查看与设备控制的任务，这是传统的控制系统不能完成的，也是最近移动互联、物联网、云计算等发展的趋势。

像一般的家庭用户来说，电脑的网页操作仅仅用来对设备状况进行修改操作就已经够繁琐了，更为方便的是手机端进行移动互联的查看与操作，毕竟现代人很难离开智能手机，手机不离身的便利也是移动监控的便利。

对于企业等专业用户来说，网页功能的丰富就可以满足企业生产维护的需求，只要信息显示的足够全面，数据存储显示准确、安全，就可以满足这些专业用户的要求了。

2.1.2自动与手动结合

用户可以预先设定变量的变化区间，如果不对设备进行强制命令的话，设备就会以设定的条件进行一系列自动调控措施。如果用户需要在某一特殊时间对设备有特殊状态要求，可以通过网络开关简单的完成设备的圆成控制行为。

对于实际生产生活中，有时仅仅需要设定范围就可以满足运行条件，比如温室大棚的自动温度控制；有时用户又需要对一些设备进行单独的操作，比方说日光灯的开、闭，是依赖用户做出决定的。这样相互结合的设计不仅不会对功能的完成带来干扰，还能满足用户的多样化需求，在生活多样化的今天，这样的设计是非常有意义的。

2.1.3系统应用成本较低

当下正是物联网市场推广的发展阶段，用户除了考虑应用的效果之外，最主要考虑的还是成本的投入问题。如果用户耗费巨额的资金，换来的只不过是将手机变为遥控器这样一个简单的改变，是不会有很大的应用前景的。

我们这个系统的设计，采用的嵌入式设备为价格低廉的8bit单片机，所有的外围设备按需添加，用户无需添加额外的材料支出；现有的网络服务器，属于一次投入，长久回报的状况，只要服务器能够承受IO压力，理论上可以并发连接的数据量是非常大的，这样就可以使得很多的用户同时进行数据上传和访问请求。所以相对而言，总体的成本平均值，还是会保持在较低的水平。

2.2 应用领域

2.2.1工农业生产

现代生产中，存在大量的关键设备和关键节点，对这些节点的监控，是正常生产的一个重要部分，如果将这些任务附加到工作人员身上，不仅难以完全掌握，也难以对正确状态进行准确判断，更难以进行不间断监测。

通过不同类型传感器节点的海量部署，可以实现一个全方位的检测网络，所有需要检测的信息，都会以数字的形式呈现在电脑屏幕上，生产部门可以在控制中心通过电脑屏幕查看、控制，大大节省人力指出，同时提高工作效率。

2.2.2居民家庭生活

在个人家庭生活的环境中，任何具有状态特征的空间都可以成为系统的部署点。大至个人的房间，小到汽车内的环境监测和电气控制，甚至是水族箱、宠物箱的信息获取、辅助电器的控制，都是非常好的应用方向。并且家庭用户带来的潜在市场相对企业来说，更加的富有潜力。

在科幻中，人类不用特殊的控制行为，各种设备就可以按照合适的进程工作，这里面涉及的就是一系列传感以及控制设计工程。我们可以想象，这种类似的系统，在生活中的广泛应用，一定会给日常生活带来翻天覆地的变化！

第三章 **自主改进**

为了摆脱对他人现有平台的依赖，提高安全性，我们正在进行网络平台的自主搭建工作，虽然开始技术有限，效果也没有Yeelink的成熟，尝试仍在进行中。

3.1 服务器建设

3.1.1操作系统

虽然说现在Windows系统占据了操作系统的绝大部分疆域，但最多的还是个人桌面操作系统，在服务器领域并没有优势

反倒是开源的Linux系统凭借着较低的资源占用，较好的移植性，成为大多数服务器的首选。其中RedHat Linux Enterprise 凭借优秀的服务，良好的设计，受到企业用户的青睐。但是这一系统确实收费的，对于我们这种小型的搭建来说，是不可接受的。

船到桥头自然直，CentOS，一款以RedHat Linux Enterprise 内核重新编译的社区操作系统，成为很多小型服务器，也是我们的最佳选择。

3.1.2服务软件

在后台设计领域，LAMP,是一个万能的搭配，Linux的操作系统，Apache服务器软件，MySQL开源数据库，PHP编程语言，使得所有网站的搭建成为可能，我们自然也采用这一搭配。

3.1.3数据库设置

对于网络数据存储这种功能来说，数据库是重要的不可分割部分。

对于我们这个系统来说，基本需要3个数据库，一个用来存储用户的用户名、密码、邮箱、用户编码等身份识别信息，一个用来存储用户对应的设备设置、传感器设置，一个用来记录传感器数据，每个传感器占据一条记录。

3.1.4数据服务

通过PHP编写的网页，用户可以完成注册的初步试用条件，注册完成登陆时，向服务器反馈用户ID，以及ID + Password字符串计算得到的MD5校验值，通过与数据库中存储数据计算得到的结果比较，即可判断用户身份，将身份正误反馈给网页，进而进入设备页面。

在设备页面，用户可以添加自己的设备集，添加完成后，数据库就会生成一张设备ID命名的表，用来存储相关传感器信息，用户进入对应的设备，即可添加传感器信息，所有基础操作完成。

当有数据传入后，用户进入浏览网页，网页将显示从数据库中查询到的最近数据信息，将结果显示出来。

3.1.5数据上传

在之前的代码介绍中，我们了解了Yeelink采用的数据上传方式，在我们的系统设计中，拟采用http头中包含ID、数据等信息的方式与服务器进行交互。

3.2客户端开发

3.2.1设计

手机客户端是直接给用户看的部分，所以首先要把数据展现的直观、形象，用户能够做到一目了然。

其次，软件的容错性要好，面对网络中断、数据异常等情况要有处理方案，不能导致死机等错误结果。

另一方面，用户使用时可能会用数据流量进行交互，这就要求我们的客户端与服务器进行数据交互的时候必须尽可能的减少字节发送数量，节省用户的流量开支。