

创新实验课

课题总结报告

课题名称：智能风扇

课题名称（英文）：Smart Fan

课题组长：剡恺

课题组成员：刘威良、李月妍

报告时间：2016 年 9 月 23 日

1. 课题研究的选题背景和意义

夏日炎炎，但是寝室中又没有空调，大多数同学只能依靠普通的电风扇来达到消暑降温的目的，但是目前市场上的风扇普遍存在以下问题：

1. 普通电风扇只有 2 档风力，无法调整到令人满意的状态；
2. 无法遥控操作，风扇距离较远时及其不方便；
3. 风扇没有定时关闭功能，使用者在睡着之后可能因为风扇无法自动关闭而伤风感冒；
4. 风扇功能单一，缺乏交互性，无法显示当前温度、时间等，使得桌面需要摆放许多工具如闹钟等，繁琐而零乱。

针对以上问题实际问题，结合已有材料，我们希望打造一个能够集合智能闹钟、显示实时时间和实时温湿度，同时又能够自主远程控制风速和方向的智能风扇。

2. 课题研究的主要内容

风扇主要实现功能如下：

1. 远程遥控：利用遥控器和红外接收装置完成对风扇的远程操作；
2. 实时显示：通过 RTC、OLED 和温湿度传感器，将时间、温度、湿度等必要信息动态显示在屏幕上，方便使用者获取最新环境信息；
3. 智能送风与定时开关机。利用 RTC 和温湿度传感器，可以根据环境自动启动/关闭风扇，也可以手动设置风扇启动/关闭时间；
4. 特色闹钟：利用 RTC、OLED 完成闹钟功能，同时在普通闹钟的基础上增加通过游戏关闭闹钟的功能。闹钟响起后会播放单词，同时会在 OLED 上出现选项，用户根据听到的单词选择正确选项，连续答对 3 题后方可关闭闹钟；
5. 劲凉模式：通过继电器模块控制水泵，利用雾化喷头喷出水雾，实现加湿与降温功能；
6. 板间通信：将风扇机械控制与闹钟控制分离在不同核心板上，核心板间使用蓝牙模块通信。

3. 课题研究的特色

1. 项目贴近生活。灵感和创作源于生活，希望能生活变得更加简单和舒适。
2. 多功能集成。充分利用所给材料，将闹钟、温湿度监测显示、自动风扇与加湿模块整合在一起，最大限度发挥模块的作用。
3. 简单有趣。改变传统风扇与闹钟形式，加入远程控制功能，使操作简单便捷，并充满趣味性。

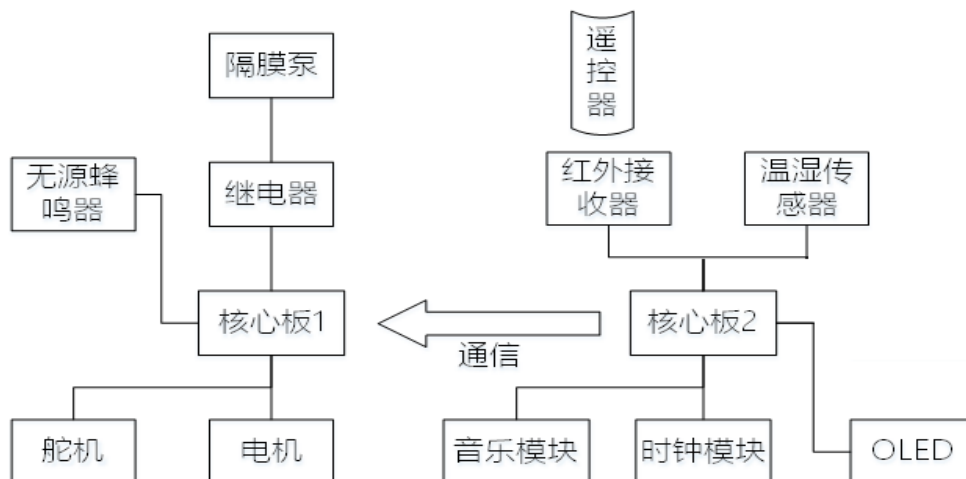
4. 课题的设计思路、总体架构、及主要步骤

■ 设计思路：

设计并制作配有闹钟的智能风扇设备：

1. 在普通风扇的基础上，配备 OLED 显示屏与时钟模块，在实现了时间与温湿度显示的同时，还可通过时间设定来启动与关闭风扇，实现风扇的智能定时功能；
2. 风扇上配有大量传感器如温湿传感器、红外接收器等，实现了风扇智能启动、智能跟随、远程遥控等功能；
3. 对闹钟系统进行改造，使用蜂鸣器播放闹钟，增加音乐模块播放单词，配合 OLED 显示，实现游戏关闭闹钟基本功能；
4. 使用继电器模块，结合程序控制水泵开关时间，实现水雾降温功能。

■ 总体架构：

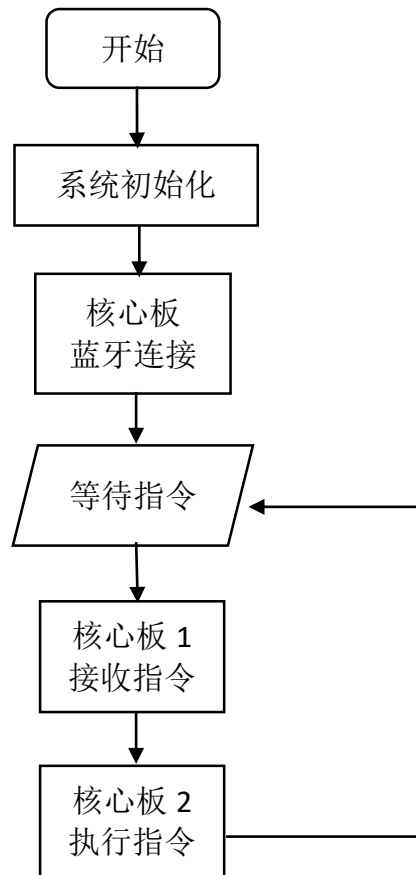


由于项目所完成功能较多，考虑到核心板内存大小及程序间的互相影响，采用多核心板控制的方案。其中：

1. 核心板一控制风扇电机、舵机、蜂鸣器、音乐播放模块，主要负责各个模块的电源控制；
2. 核心板二与红外传感器、温湿度传感器相连接，配合 RTC 与 OLED 模块，实现信息显示与远程命令接收功能。

两块核心板依靠蓝牙模块进行通信，其中核心板二为主机，核心板一为从机。核心板二收到指令后向核心板一发送相应命令，核心板一产生相应动作，实现特定功能。

■ 系统流程：



5. 各模块详细设计、实现及调试说明与分析

1. 蓝牙通信：

- 串口支持，共2路：
 - USB虚拟串口 Serial;
 - 在模块上标示为 Serial1[D0(RX), D1(TX)]。

经官方文档查询，Microduino 的 CoreUSB 共有 2 个串口，一个用于 USB，故将另一串口用于蓝牙通讯。代码如下：

```
1. /*蓝牙通信*/
2. #define my_Serial Serial1 //定义蓝牙通讯为串口 1
3. String msg = ""; //定义一个字符串用于接收信息
4.
5. //初始化中加入如下代码
6. void setup(){
7.   Serial.begin(9600);
```

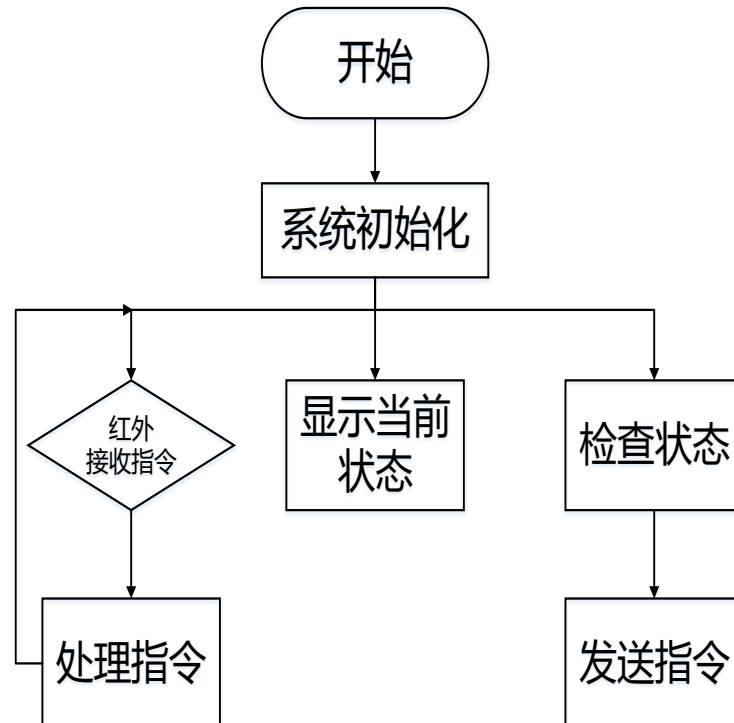
```
8. my_Serial.begin(9600);  
9. }
```

在主循环中加入如下代码实现利用串口控制蓝牙模块，使用 AT 指令集将核心板二蓝牙模块设置为主机，自动连接从机。

```
1. if (Serial.available())           //监视到串口监视器的数据  
2.   my_Serial.write(Serial.read()); //将数据写入 BT  
3. if (my_Serial.available())        //监视到 BT 串口的数据  
4.   Serial.write(my_Serial.read()); //将数据在串口监视器打印出来
```

2. 核心板二：

核心板二需要实现遥控指令接收和实时显示当前状态信息，并将相应指令发送给核心板一，多个任务间需要同时进行。这里采用 Arduino 官方提供的 Timer 库来实现多任务处理。Timer 库可以实现定时执行任务功能，最多实现 10 个任务定时执行，满足项目要求。



主要代码如下：

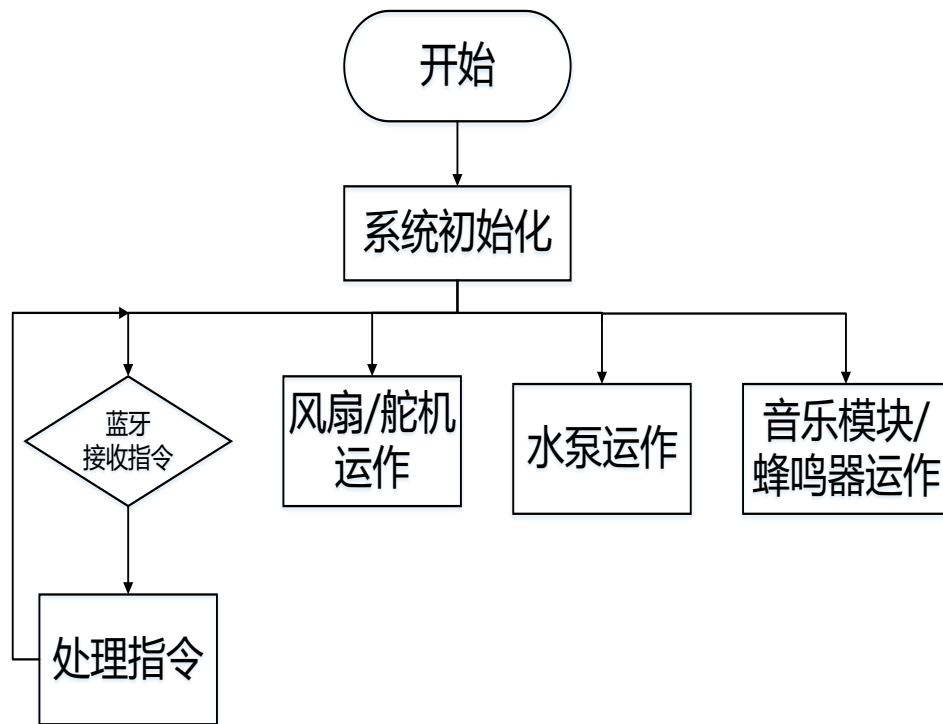
```

1 Timer t;
2 unsigned short int timeID;    //时间显示任务 ID
3 unsigned short int checkID;   //检查状态任务 ID
4 unsigned short int waterID;   //水泵状态 ID
5
6 void setup(){
7     timeID = t.every(500, timeshow);
8     checkID = t.every(1000, checkState);
9     waterID = t.every(1000, cool);
10 }
11
12 void loop(){
13     t.update();
14
15     if (irrecv.decode(&ir_res)) {
16         switch (ir_res.value) {
17             ...//判断命令
18         }
19     }
20 }
21
22 //各种任务函数
23 void timeshow(){
24     ...
25 }
26
27 void checkState(){
28     ...
29 }
30
31 void cool(){
32     ...
33 }

```

3. 核心板一：

核心板一用于控制各个模块的开关，当收到核心板二的指令后，根据对应指令执行相应命令。结合现实情况，可能会同时运行水雾喷洒、风扇和闹钟功能，故采用 **Timer** 库进行各任务控制，保证各任务不会互相影响。



主要代码如下：

```

1  Timer t;
2  unsigned short int alarmID;    //闹钟ID
3  unsigned short int fanID;      //风扇任务ID
4  unsigned short int servoID;    //舵机任务ID
5  unsigned short int alarm_eID;  //闹钟单词ID
6  unsigned short int delayID;    //水泵ID
7
8  void setup(){
9      alarmID = t.every(1000, playMusic);
10     alarm_eID = t.every(3000, english);
11     //fanID = t.every(1000, onFan);
12     servoID = t.every(15, onServo);
13     delayID = t.every(5000, water);
14     t.every(10, recMsg);        //接收指令
15 }
16
17 void loop(){
18     t.update();
19
20     if(flag_fan == 1){ //舵机控制
21         if(pos == 120){

```

```

22         pos = 0;
23     } else {
24         pos++;
25     }
26     delay(50);
27 }
28 }
29
30 void recMsg() { //接收指令
31     if (my_Serial.available() > 0) { //如果串口有数据输入
32         msg = my_Serial.readStringUntil('\n'); //获取换行符前所有的内容
33         Serial.println(msg); //在串口监视器显示收到的
34         msg中的字符串
35         fz = atoi(msg.c_str());
36         //my_Serial.println(fz); //打印当前风力
37         if (fz == 0) {
38             order = msg[0];
39             //my_Serial.println(order); //打印命令
40             switch (order) {
41                 ...//指令处理
42             }
43         }
44     }
45 }
46
47 //其余各项任务函数

```

6. 课题成果及其性能分析

已经完成的功能有：

1. 基本控制：

可通过遥控器控制风扇启动、摇头、喷出水雾，设定闹钟等。

按下遥控器电源键可控制风扇开启/关闭。

2. 实时显示：

可将当前时间、系统功能开关（如闹钟开关，定时启动开关，劲凉模式开关等）及当前环境温度与环境湿度显示在液晶屏上；



显示状态含义如下：

字母	含义
F	风扇开启，水泵关闭
AF	风扇开启且水泵在自动模式
KF	风扇开启且水泵在持续模式
A	闹钟开启
T	风扇定时启动开启
数字	风扇定时关闭开启，数字代表风扇持续时间（单位：小时）

3. 智能送风：

按下遥控器 E 键可设置风扇启动时间，当风扇启动后按下静音键可设置风扇关闭时间。

当时间到用户设定值时，风扇可自动启动；当运行时间到用户设定值时，风扇可自动关闭。

4. 劲凉模式：

风扇启动后，按下 B 键可控制水泵开关，水泵有 3 种工作模式：关闭/自动(屏幕显示 A)/持续(屏幕显示 K)。当处于自动状态时，水泵将根据环境温湿度喷水；当处于持续状态时，水泵将持续喷水（每 5 秒喷一次）。

当温湿度达到设定值时(测试条件为温度高于 25 度,湿度低于 50%), 喷出水雾；

5. 特色闹钟：

按 A 键开启/关闭闹钟，按 D 键设定闹钟时间。时间到时闹钟响起，同时播放英文单词读音，屏幕显示 4 个选项，用户用遥控器选择正确选项，连续选对 3 题方可停止闹钟。

7. 课题人员分工与合作情况说明

项目期间组员关系融洽，能积极参与项目建设中，任务分配合理，每个人都能够及时完成任务，且完成质量较高。分工详情见下表：

姓名	主要任务	贡献度
刻恺	组长，负责整体项目规划与人员调度。负责整个项目的代码编写与调试工作。	40%
刘威良	负责项目的硬件设计、材料采购与硬件拼接、调试工作。	30%
李月妍	负责项目的硬件设计与硬件拼接、调试工作，负责团队资料整理工作。	30%

8. 课题所用器材列表及说明（包括课程提供的器材及自行购买的器材）

序号	名称	型号	数量	单价	主要功能	购买方式
1	彩色开发板		4	1.33	拼装为风扇底座	淘宝
2	雾化喷头		1	1.49	用于水的雾化	淘宝
3	塑料条		4	0.79	拼装风扇模块等	淘宝
4	水泵		1	44.75	用于实现喷射水雾	淘宝
5	胶水		1	6.93	整体粘结	淘宝
6	绑扎带		若干	0.60	捆绑固定	淘宝
7	热缩管		1	0.90	导线连接处绝缘	淘宝
8	插线板		1	48.90	集合整体项目供电	京东
9	核心板	mCookie-CoreUSB	2		运行程序	
10	连接板	mCookie-Hub	2		提供传感器接口	
	显示模块	mCookie-OLED	1		显示环境信息与系统状态	
	时钟模块	mCookie-RTC	1		提供时间功能	
	蓝牙模块	MCookie-BT	2		用于 2 块核心板间通信	
	应用扩展板	MCookie-Duo-V2	1		降低核心板 2 堆叠高度	

	电源管理模块	mCookie-BM	1		为特定模块供电	
	电池充放电电源管理模块	mCookie-BM-shield	1		安装电池	
	音频模块	mCookie-Audio	1		闹钟单词存储	
	音频功率放大模块	mCookie-Amplifier	1		播放单词	
	无源蜂鸣器	Sensor-Buzzer	1		闹钟播放音乐	
	继电器	Sensor-Relay	2		控制水泵与风扇开关	
	舵机	Sensor-Servo Connector	1		控制风扇转向	
	温湿度传感器	Sensor-Tem&Hum-S2	1		检测环境温湿度	
	红外接收传感器	Sensor-IR Receiver	1		接收遥控器指令	
	遥控器	mCookie	1		发送指令	
	喇叭	mCookie	2		播放闹钟单词	
	连接线	4pin	若干		连接传感器与模块	
	USB 转 MicroUSB		3		用于 2 块核心板和风扇供电	

9. 课题成果链接（包括视频、代码）

代码：<https://github.com/kkkyan/microduino>

视频：http://v.youku.com/v_show/id_XMTczNDk2MzM3Ng==.html

10. 参加本课程的收获、体会及对课程的建议

剡恺：

在参加创新实验之前我曾在模电实验中接触过 Arduino 的最小系统，也就是在那时对 Arduino 产生了一些兴趣。之前听老师介绍过 Microduino 项目，觉得能把模块变成这种吸附式的拼接很神奇，恰巧学校提供了这个创新实验课，于是积极报名，也很荣幸自己能被选中。整个项目过程中自己收获了很多，从刚开始单个模块的简单测试，到将各个模块综合后整体调试；从简单的写入循环，到明白单片机的中断逻辑，运用各种库来解决冲突问题。同时，

这次课程不仅让我熟悉了 **Microduino** 相关系统的编程方法，也让我明白了团队协作的重要性。整个课程遇到不少硬件上的问题，组员在一起协作讨论，动手实践，很多问题都能找到合适的解决方法。

愿意担任助教工作。

刘威良：

在本次创新实验中，对于 Arduino 单片机之前并不是很了解，有很多之前未涉及的知识，好在 **microduino** 的库十分完善，网上的例程和代码很完善，这对我的学习起到了很大的帮助。但在真正的学习实践过程中，在代码和单片机的衔接上曾遭遇瓶颈，有的问题一时难以解决且难以寻求帮助，好在我们团结协作，克服了困难。因为我们制作的智能风扇是没有模具的，所有的模具需要自己购买制作，这个对我的动手能力是不小的考验，没一块板子每一个模块都需要自己手动设计位置并粘连，在这个过程中极大地提高了我的动手能力和实践能力。之实践过程中，风扇的转速一直达不到要求，我们尝试了很多办法，从最早的 **motor** 模块到直接连接在电池两端，从直接连接在 **usb** 端口的两端到电流过大自己一步步测量焊接电阻，这对我的硬件水平又是一个极大的提升。总的来说，觉得参加这次创新实践是一个非常愉快有收获很大的事情，比一般的小学期要充实很多，这次实践不仅提升了我对单片机等硬件的兴趣，更让我的知识水平有了很大提升，很感谢老师能给我这次机会，我以后会继续努力的。

愿意担任助教工作。

李月妍：

之前去参加创客展览，发现开源电子硬件正逐渐成为创新创业领域及电子设计领域的一大重要角色，正如这次创新实验所用的 **microduino** 套件，几块功能各异的集成电路模块和开源程序库就能产生极大的创造力。通过这次课程，我深刻了解了从零到一完成一个项目的过程，从项目设计、查阅相关参考资料、制定实现方案、修整方案到最后的实现，这一经历十分宝贵。在这次实验要感谢我们的老师、助教的指导和组员们的努力，令我们最初的设想得以实现。我觉得进行项目之前老师可以先开一些课程让同学们对 **microduino** 有足够的了解去入手，希望这一课程越办越好。

不愿意担任助教工作。

11. 参考文献

[1]lascchen@gmail.com.Wiki-Microduino[G/OL].

<https://wiki.microduino.cn/index.php/%E9%A6%96%E9%A1%B5>, 2016-9-8

[2]Arduino.Arduino – Reference[G/OL]. <https://www.arduino.cc/en/Reference/HomePage>,2016

附件 1: Microduino 及 mCookie 套件学习和使用中的问题及解决方法

Q1: 代码无逻辑问题，但是在 mCookie 上执行出现错误或执行不完整？

A1: 请保证动态内存存在 500 字节以上，低于此字节可能会出现程序不稳地现象。

Q2: 核心板出现无法上传程序现象？

A2: 请将 CoreUSB 置于 Duo-2 扩展板上，连接 USB，在上传前按住扩展板的复位键，点击上传后松开复位键，上传即可成功。

Q3: 蓝牙模块无法通信，AT 指令无效？

A3: 首先确认串口波特率与蓝牙硬件波特率一致，串口指令使用/r/n 结尾，若通电后串口可打印 Hello World 则蓝牙通讯正常；使用 AT 指令集建议在核心板中只上传通讯代码，可防止其他程序进行干扰，同时，确保蓝牙模块没有与其他蓝牙模块互连，只有在未连接前才可使用 AT 命令。

Q4: 模块 5V 引脚翘起？

A4: 请勿使用 IO 口直接进行 5V 供电！

附件 2: 课题执行过程中的问题及解决方法

Q1: 示例程序上传后不工作？

A1: 有些示例程序较长时间未更新，最好自己翻阅 wiki 检查元件接口方法与调用函数。

Q2: 模块工作时好时坏？

A2: 由于 mCookie 使用磁力吸附方式，接触没有杜邦线稳定。所以不工作时请尝试调节模块接触情况。

附件 3: 有用文档或链接

1、软重启: <https://github.com/WickedDevice/SoftReset>

2、Aruidno 官方函数手册: <https://www.arduino.cc/en/Reference/HomePage>

3、Arduino 多任务调度: <http://playground.arduino.cc/Code/Timer>