

## 华中师范大学龙岗附属中学学生论文设计

**题目:基于 Arduino 的桌面气象终端**

学习形式: 自学

专业:

层次: 高中

姓名: 邓思恒

学号:

指导老师:

**【摘要】**：基于 Arduino 的桌面气象终端，通过 Arduino 平台对 esp8266 芯片编程，借助 esp8266 的 wi-fi 通信能力，通过 NTP 获取时间信息，借助 API 获取天气信息，达到精准校时、天气预报的功能。本设计采用 Arduino+esp8266 的方案，具有成本低廉、可实用性高、拓展库丰富等优点。设计难度在于程序中的数据处理、屏幕驱动以及显示 UI 设计等方面。

**【关键词】**：Arduino、esp8266、物联网、嵌入式终端、电子时钟

## 目录

1. 概述.....	4
1.1 课题研究背景.....	4
1.2 本课题研究的内容及目的.....	4
1.3 系统结构分析.....	5
1.4 系统的控制思路与流程.....	5
2. 硬件及程序源代码介绍.....	6
2.1 硬件介绍.....	6
2.1.1 NODE-MCU 开发板介绍.....	6
2.1.1.1 esp8266 芯片介绍.....	6
2.1.2 TFT 液晶屏介绍.....	7
2.2 编程软件及源代码介绍.....	8
2.2.1 Arduino 介绍.....	8
2.2.2 Arduino IDE 介绍.....	9
2.2.3 程序源代码介绍.....	9
3. 成品展示.....	14

华中师范大学龙岗附属中学

姓名:邓思恒

## 1. 概述

### 1.1 课题研究背景

近年来物联网发展迅速，各类物联网技术正逐步普及到人民的日常生活中。物联网产品正向着轻量化、小型化、智能化发展，由于物联网的高拓展性，其产品功能不断得到发展和完善，但目前出现在社会上的方案尚存在以下几点问题：

- (1) 解决方案闭源，不利于学习交流
- (2) 现有的开源方案缺乏基本的注释和程序说明，提高了参考学习的门槛
- (3) 现有方案存在价格相对高昂、原理晦涩等问题

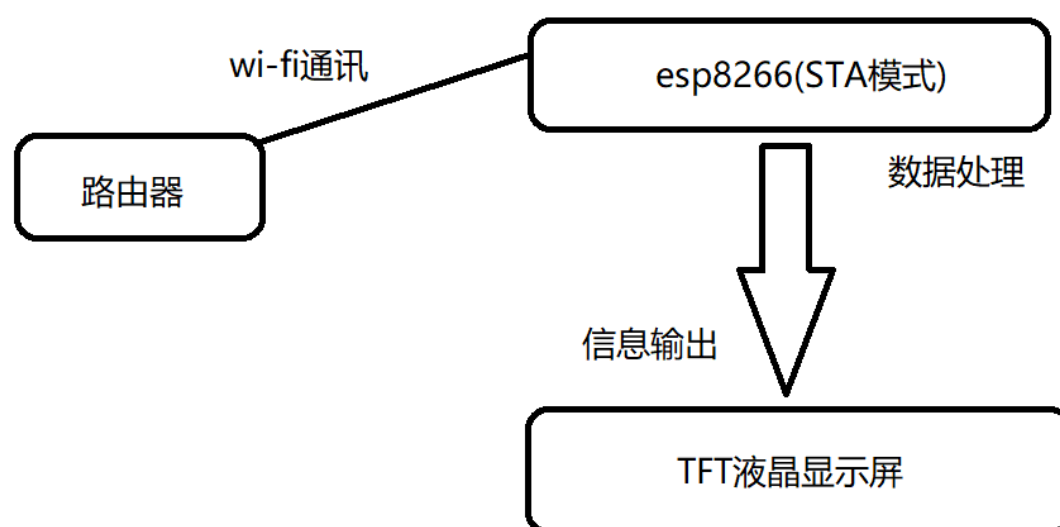
### 1.2 本课题研究的内容及目的

本项目在软件上依赖 Arduino 平台，硬件上选择具有 Wi-Fi 功能和 80MHz 主频的 esp8266MCU 方案，实现天气预报和时钟功能。研究方案综合考虑了方案难度、设计成本和产品实用性，具有成本低廉、可实用性高、程序简洁、资料完善的优点。同时，本程序将以源代码的形式在开源平台 Github 上向全社会发布，以支持相关爱好者学习。

## 1.3 系统结构分析

本项目由三部分组成（如图一），搭载 esp8266 芯片的 NODE-MCU 开发板、1.8 寸 tft 液晶屏、相关外围电路。各部分组成如下：

- （1）核心程序基于 Arduino 开发
- （2）使用 SPI 总线驱动的 TFT 液晶屏
- （3）核心模块通过 NODE-MCU 开发板与 tft 屏相连组成，其中 esp8266 作为 MCU（微控制单元）承担处理任务的同时还兼任连接 Wi-Fi 路由器获取数据的功能。



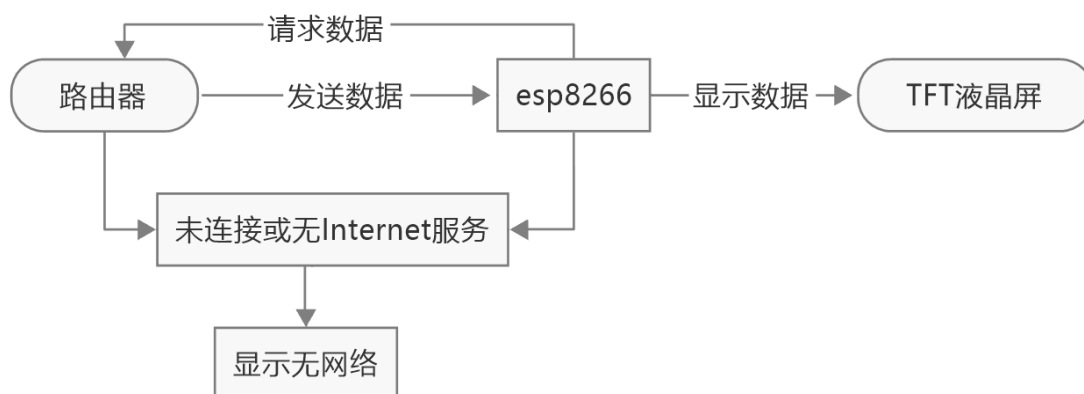
（图一）系统组成示意图

## 1.4 系统的控制思路与流程

esp8266 芯片通过 STA 模式以 UDP 通讯协议与局域网路由器（注意路由器应提供 Internet 服务）相连，芯片通过“心知天气”API 接口从互联网上获取天气信息，通过阿里云 NTP 服务器获得当前

时间。在得到所需的数据后，esp8266 将驱动 TFT 屏把可用信息显示出来供用户浏览。

流程图如下：



(图二) 系统控制流程示意图

## 2. 硬件及程序源代码介绍

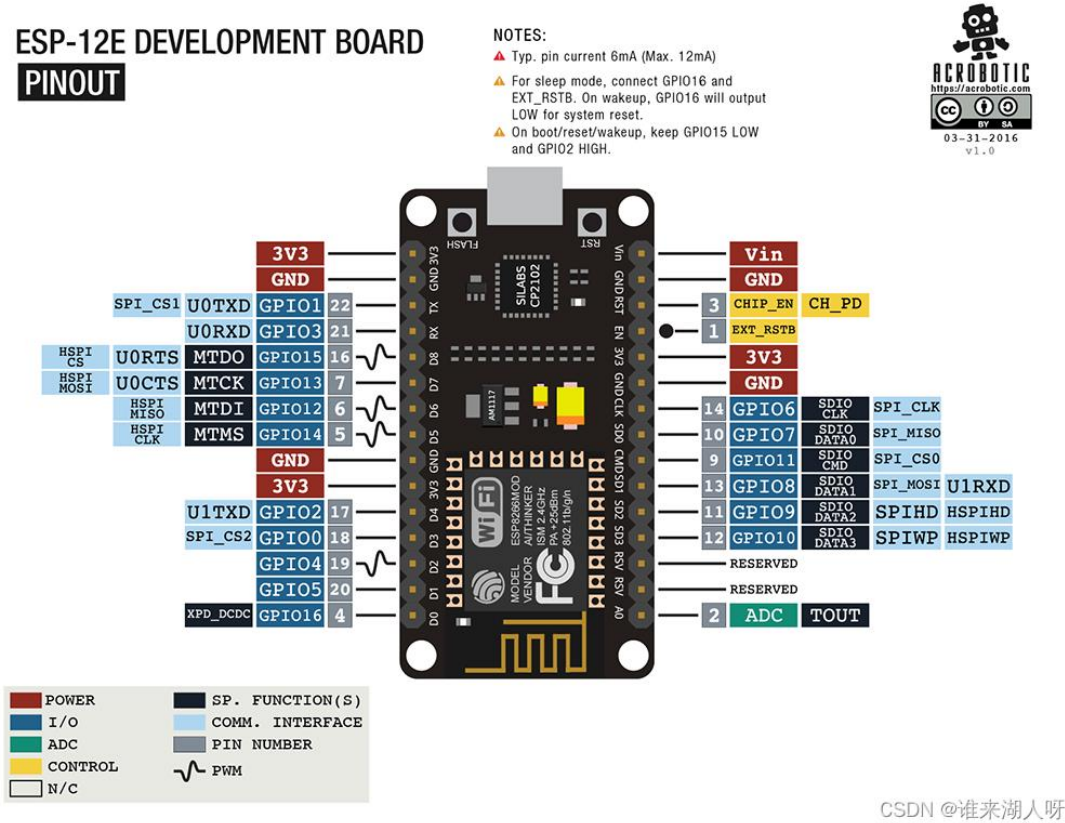
### 2.1 硬件介绍

#### 2.1.1 NODE-MCU 开发板介绍

##### 2.1.1.1 esp8266 芯片介绍

esp8266 是一块由乐鑫科技公司开发的面向物联网应用的高性价比、高度集成、具有 WI-FI 通讯功能的 MCU，它内置的超低功耗 Tensilica L106 32 位 RISC 处理器有着充足的性能支撑此次项目开发，且搭载该芯片的开发板价格区间普遍在 10~20 元人民币，成本低廉，利于复制参考和二次开发。所以选择该芯片作为主控芯片。

2.1.1.2 开发板外围电路介绍



(图三) NODE-MCU 引脚图

图三为本项目采用的开发板的引脚图。如图所示，开发板为板载的 esp8266 芯片设计了 5V-3V3 降压电路和串口下载电路。同时，NODE-MCU 为开发者提供了 I2C 引脚和 ISP 引脚，符合本次项目的需求。

2.1.2 TFT 液晶屏介绍

目前市面上主流的相关显示方案有 OLED 屏、电子墨水屏、TFT 液晶显示屏、LCD 液晶显示屏、数码管、点阵屏等。

由于本项目需要显示屏具有数字、汉字、有色图像输出功能，故不考虑 LCD 液晶显示屏、数码管、点阵屏方案。

各类屏幕的优缺点如表：

	TFT 液晶显示屏	OLED 屏	电子墨水屏
优点	1.反应时间快 2.可视角度大 3.显示效果良好 4.价格低廉	1.反应时间快 2.可视角度大 3.显示效果良好 4.相对省电	1. 省电节能  2. 保护视力
缺点	高耗电	1.长期显示相同内容会色衰、烧屏 2.闪屏现象 3.价格高昂	1.刷新率低 2.价格高昂

(表一) TFT 液晶显示屏、OLED 屏、电子墨水屏对比

综合各类优缺点，最终选择 TFT 屏的理由如下：

- (1) TFT 屏成本较为低廉，以 1.8 寸彩色为例，同等条件下 TFT 和 OLED 的差价高达 50~60 人民币。
- (2) 电子墨水屏无法做到高刷新率，同时价格也相对较高，性价比过低。
- (3) 该终端直接采用电源线供电，TFT 屏的相对高耗电问题不用考虑。

## 2.2 编程软件及源代码介绍

### 2.2.1 Arduino 介绍

Arduino 是一款便捷灵活、方便上手的开源电子原型平台。由支持 Arduino 的开发板和 Arduino IDE 组成。

Arduino 语言建立在 C 语言和 C++的基础之上开发，将一些涉及底层原理的开发过程函数化，使得开发者不必面对繁杂的底层原理，将精力投入到创意实现和产品设计上，大大简化了开发流程、提高了开发效率。



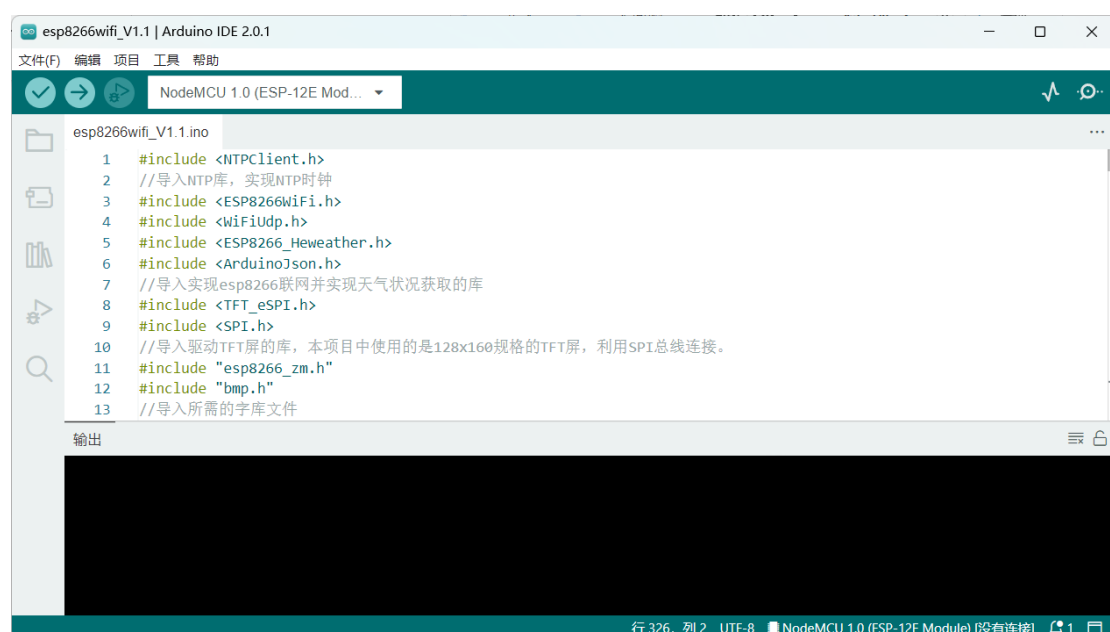
除此之外，Arduino 具有高度的开放性，在 Arduino 社区和各大开源平台上有着丰富的函数库与示例程序，降低开发难度的同时，这种开源的氛围也有利于初学者学习。

综上所述，Arduino 平台具有高度开放、简单易学、便捷灵活的特点，因此，本项目采用 Arduino 进行程序设计。

## 2.2.2 Arduino IDE 介绍

Arduino IDE 是 Arduino 官方为开发者提供的免费集成开发环境，本项目中演示所用的软件版本为：Arduino IDE 2.0.1。

Arduino IDE 2.0.1 软件界面如下：



(图四) Arduino IDE 软件界面

## 2.2.3 程序源代码介绍

### 2.2.3.1 导入的库函数介绍

```
#include <NTPClient.h>
//导入NTP库，实现NTP时钟 ↵

#include <ESP8266WiFi.h>      #include <ESP8266_Hweather.h>
#include <WiFiUdp.h>          #include <ArduinoJson.h>
//导入有关esp8266wi-fi连接的库 //导入实现天气状况获取的库 ↵

#include <TFT_eSPI.h>
#include <SPI.h>
//导入驱动TFT屏的库，本项目中使用的是128x160规格的TFT屏，利用SPI总线连接。↵
#include "esp8266_zm.h"
//导入所需的字库文件 ↵
```

(图五) 程序导入库部分代码

### 2.2.3.2 程序流程详解

#### I .NTP 时钟

网络时间协议 NTP (Network Time Protocol) 是 TCP/IP 协议族里面的一个应用层协议，用来使客户端和服务端之间进行时钟同步，提供高精度度的时间校正。NTP 服务器从权威时钟源（例如原子钟、GPS）接收精确的协调世界时 UTC，客户端再从服务器请求和接收时间。

NTP 程序实现如图：

```
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP,"ntp1.aliyun.com",60*60*8,30*60*1000);
timeClient.begin();
//开始获取时间
```

(图六) NTP 程序部分代码

作用：创建 UDP 实例以便使用 UDP 协议接受 NTP 报文、确定 NTP 服务器（在该程序中使用阿里云时钟源）、启动 NTP 服务

```
timeClient.update();  
itime = timeClient.getFormattedTime();  
//开始获取时间原始数据，并保存在变量itime中
```

(图七) NTP 程序部分代码

作用：启动 NTP 服务、在 loop 函数中不断更新时间

## II.“心知天气”API 介绍

心知天气是国内领先的商业气象服务提供商，它为开发者提供了获取准确时间的 API 接口。在开源平台 Gitee 上有本项目使用的`#include <ESP8266_Heweather.h>`

的官方文档说明，其部分截图如下：

### 获取实时天气信息

```
WeatherNow weatherNow;           // 建立weatherNow对象  
weatherNow.config(UserKey, Location, Unit, Lang); // 配置请求信息  
weatherNow.getServerCode();        // 获取API状态码  
weatherNow.getLastUpdate();        // 获取服务器更新天气信息时间  
weatherNow.getTemp();              // 获取实况温度  
weatherNow.getFeelLike();          // 获取实况体感温度  
weatherNow.getIcon();              // 获取当前天气图标代码  
weatherNow.getWeatherText();       // 获取实况天气状况的文字描述  
weatherNow.getWindDir();           // 获取实况风向  
weatherNow.getWindScale();         // 获取实况风力等级  
weatherNow.getHumidity();           // 获取实况相对湿度百分比数值  
weatherNow.getPrecip();            // 获取实况降水量,毫米
```

(图八) 心知天气官方文档部分截图

虽然该项目使用的 NTP 时钟库没有获取日期信息的函数，但在心知天气 API 中提供了 `weatherNow.getLastUpdate()` 函数，可以借助该接口来获取日期信息。其他函数介绍如下：

## 获取天气预报信息

```
WeatherForecast WeatherForecast;    // 建立WeatherForecast对象
WeatherForecast.config(UserKey, Location, Unit, Lang); // 配置请求信息
WeatherForecast.getServerCode();    // 获取API状态码
WeatherForecast.getLastUpdate();    // 获取服务器更新天气信息时间
// 以下i取值为 0,1,2, 分别代表今天, 明天和后天
WeatherForecast.getTempMax(i);      // 获取最高温度
WeatherForecast.getTempMin(i);      // 获取最低温度
WeatherForecast.getIconDay(i);      // 获取天气图标代码
WeatherForecast.getTextDay(i);      // 获取天气状况的文字描述
WeatherForecast.getWindDirDay(i);   // 获取风向
WeatherForecast.getwindScaleDay(i); // 获取风力等级
WeatherForecast.getHumidity(i);     // 获取相对湿度百分比数值
WeatherForecast.getPrecip(i);       // 获取降水量,毫米
WeatherForecast.getUvIndex(i);      // 获取紫外线强度指数
```

## 获取空气质量信息

```
AirQuality AirQuality;    // 建立AirQuality对象
AirQuality.config(UserKey, Location, Unit, Lang); // 配置请求信息
WeatherForecast.getServerCode(); // 获取API状态码
WeatherForecast.getLastUpdate(); // 获取服务器更新天气信息时间
AirQuality.getServerCode();    // 获取API状态码
AirQuality.getLastUpdate();    // 获取服务器更新天气信息时间
AirQuality.getAqi();           // 实时空气质量指数
AirQuality.getCategory();      // 实时空气质量指数级别
AirQuality.getPrimary();       // 实时空气质量的主要污染物, 优时返回值为NA
```

(图九) 心知天气官方文档部分截图

## III.TFT 屏驱动

TFT 屏的驱动方式主要有 I2C 和 SPI 两种, 本项目综合考虑程序功能拓展性和设计难度后选择 SPI 驱动 TFT 屏。

使用的函数库如图:

```
#include <TFT_eSPI.h>
#include <SPI.h>
```

(图十) TFT 屏驱动函数库导入部分程序

在使用该库时应注意项目所使用的驱动芯片和屏幕的大小，并在驱动中修改参数，屏幕驱动程序如下：

```
43
44 // Only define one driver, the ot
45 // #define ILI9341_DRIVER //
46 // #define ILI9341_2_DRIVER //
47 #define ST7735_DRIVER // Def
48 // #define ILI9163_DRIVER // D
49 // #define S6D02A1_DRIVER
50 // #define RPI_ILI9486_DRIVER // 2
51 // #define HX8357D_DRIVER
52 // #define ILI9481_DRIVER
53 // #define ILI9486_DRIVER
```

(图十) ESPI 库部分截图

```
// For ST7789, ST7735, ILI9163 and GC9A01 ONLY, de
// #define TFT_WIDTH 80
| #define TFT_WIDTH 128
// #define TFT_WIDTH 172 // ST7789 172 x 320
// #define TFT_WIDTH 240 // ST7789 240 x 240 and
| #define TFT_HEIGHT 160
// #define TFT_HEIGHT 128
// #define TFT_HEIGHT 240 // ST7789 240 x 240
// #define TFT_HEIGHT 320 // ST7789 240 x 320
// #define TFT_HEIGHT 240 // GC9A01 240 x 240
```

(图十一) ESPI 库部分截图

### 3 . 成品展示

