毕业论文

|  |  |
| --- | --- |
| 学生姓名： |  |
| 学 号： |  |
| 学院、系： | 机电工程学院 |
| 专 业： | 电子科学与技术 |
| 题 目： | 基于STM32的户外环境监测系统的设计 |
| 指导教师： |  |

2016年5月1日

目录

[1](#_Toc13705) **[引言](#_Toc13705)** [2](#_Toc13705)

**[1.1 本课题的研究意义](#_Toc396)** [2](#_Toc396)

**[1.2 本课题的研究现状](#_Toc25397)** [2](#_Toc25397)

**[1.3本课题的发展趋势和研究可行性](#_Toc20622)** [2](#_Toc20622)

**[1.4本课题主要研究工作](#_Toc2272)** [3](#_Toc2272)

**[2 系统的概述和相关原理](#_Toc15802)** [4](#_Toc15802)

**[2.1 系统的概述](#_Toc15721)** [4](#_Toc15721)

**[2.1.1 总体设计的方案](#_Toc461)** [4](#_Toc461)

**[2.1.2 总体框图](#_Toc22906)** [4](#_Toc22906)

**[2.2 相关理论](#_Toc22102)** [5](#_Toc22102)

**[2.2.1 STM32平台](#_Toc15563)** [5](#_Toc15563)

**[2.2.2 WIFI模块](#_Toc30785)** [6](#_Toc30785)

**[3 硬件电路设计](#_Toc15802)** [8](#_Toc15802)

**[3.1 主控电路](#_Toc10126)** [8](#_Toc10126)

**[3.2 WIFI通信模块](#_Toc10764)** [8](#_Toc10764)

**[3.3报警模块](#_Toc29025)** [10](#_Toc29025)

**[3.4温湿度模块](#_Toc11641)** [11](#_Toc11641)

**[3.5 光照强度模块](#_Toc1408)** [12](#_Toc1408)

**[3.6 PM2.5模块](#_Toc18892)** [12](#_Toc18892)

**[3.7液晶模块](#_Toc6986)** [13](#_Toc6986)

**[3.8电源模块](#_Toc7699)** [14](#_Toc7699)

**[4 软件设计](#_Toc18125)** [14](#_Toc18125)

**[4.1 总体程序流程图](#_Toc17624)** [14](#_Toc17624)

**[4.2 WIFI模块](#_Toc19154)** [16](#_Toc19154)

**[4.3 LCD5110模块](#_Toc19075)** [17](#_Toc19075)

**[4.5 DHT11模块](#_Toc16216)** [27](#_Toc16216)

**[4.7 ADC-DMA模块](#_Toc23346)** [29](#_Toc23346)

**[5 Android系统](#_Toc18125)** [3](#_Toc18125)0

**[5.1 Android系统简述](#_Toc17624)** [3](#_Toc17624)1

**[5.2 Android开发环境的搭建](#_Toc17624)** [3](#_Toc17624)6

**[6 运行与调试](#_Toc18125)** [3](#_Toc18125)7

**[6.1 测试](#_Toc17624)** [3](#_Toc17624)8

[结论 39](#_Toc17873)

[参考文献 40](#_Toc16658)

基于STM32的户外环境监测系统的设计

**摘 要:**随 着 无 线 局 域 网 技 术 的 快 速 发 展 ， 无 线 终 端 已 经 融 入 了 我 们 的 生 活 ， 无 论 是 智 能 手 机 还 是 笔 记 本 ， WiFi 功 能 几 乎 是 必 不 可 少 的 。 伴随着电子产品的快速发展，电子测量的应用也越来越广泛，不再局限于军事，转向民用发展。更加使得电子测量技术的到极大的发展。数字信号有着良好的抗材料本身干扰和环境干扰的能力，所以，现在市面上的模拟信号产品逐渐被数字信号产品代替，并且使得测量产品越来越方便。根据市场调查，虽然市场上有很多关于环境监测系统测试仪，但大多数是应用于空气，湖泊，海洋，河流等大型检测系统，关于检测环境的小型简单方便使用的检测系统。本文开发并实现基于一种嵌入式开发平台的 STM32 的 WiFi 模块，结合以上物联网新型概念，实现用户通过网络对环境的实时监控。该系统可以使用户能够对想要知道的情况进行实时的掌握以及控制。通过各种传感器获取家庭内信息（温湿度信息、光照信息、PM2.5），用户在Android界面上可以对这些信息进行掌控。

**关键词：**环境监测；STM32；传感器；WIFI；Android

# 1 引言

在日常生活中，关乎我们的生活质量，也在农业大棚中关乎大棚的产量，合适的，会极大地提高产量。合适的也会极大地提高我们的生活质量。但是。传统的监控系统监控耗时间极长，效率挤满，而且需要人工不间断的监控，费事，费力，才能达到每时 每刻监控的目的。我们采用stm32对外界环境的抗干绕能力也强，可以应用在场合,并且配合Android可以有效的采集多个传感器和信息分析，分析等信息，并置设报响声的系统，数据超过设定的数值，就可以利用响声系统来提醒操作人员进行调控。该系统对可以系统的监控和控制，有不错的效果。

* 1. **[本课题的研究意义](#目录)**

随着现代社会人们的生活环境越来越复杂，人们对环境的要求也越来越严格，环境对生活的影响已经成为一个热点问题。市面上的环境监测装置功能单一化，一般只能监测某一种环境因素，不能够很好的满足人们的需求。再加上物联网技术的快速发展，物联网是新一代信息技术的重要组成部分，也是“信息化”时代的重要发展阶段。为此设计出由物联网技术、传感器技术和电子信息技术组成的多功能环境监测系统，可以同时监测多路信息，比如温度、湿度、光照强度、大气压强、PM2.5、氨气、硫化物、烟雾和苯系蒸汽等。帮助人们更准确的监测环境参数，力求最大限度改善环境，使人们的生活更加美好。

**1.2 [本课题的研究现状](#目录)**

国内环境监测系统发展于 20 世纪 70 年代。目前国内现在对环境监测还停留在传统的人工环境监测方法，几乎还基于单片机终端监测很少具有实时通信功能。产品的现场安装受使用环境所限，环境应用适用性差，测控手段单一。很难进行远距离的监控和报警。我国环境监测工作是随着人们对环境保护认识的深化和环保工作的需要逐步发展起来的。国外环境监测系统发展比较早。目前国外的环境监测系统已经广泛应用在各个领域，并且发展比较成熟完善。许多国家的环境监测已经很先进了，可以同时进行多路采集各种环境参数。美国的一些公司已经开始利用太空技术进行环境监测作业。随着国内外物联网技术的快速发展，将物联网技术和电子信息技术合并发展的趋势已经开始展现出来了。目前已经有许多家企业研究院已经开始在研究物联网如何应用在我们的生活当中。但是，目前市场上类似的环境监测电子产品多为单一功能系统，限制了应用范围，而随着“系统集成化”的发展趋势一步步推进，只能实现上述单一功能的单一系统将逐步被淘汰，设计出能实现综合功能的系统则是大势所趋

**1.3本课题的发展趋势****和研究可行性**

物联网这个行业依然存在一些问题需要解决。但在科学技术高速发展的大环境下，物联网也会朝着越来越好的方向发展，主要的发展趋势有以下几个方面：  
（ 1）低成本。物联网还做不到普及，最大的原因就是成本问题。随着电子元器件的不断更新、改进，未来的低成本产品也是必然的趋势；  
（ 2）标准化。无规矩不成方圆，没有一个统一业内标准的支持，物联网的发展终究会受到阻碍。建立统一的标准，有利于新产品在旧版本的基础上进行下一步改进，形成良性的发展过程。  
（ 3）稳定化。稳不稳定是客户使用产品最直观的感受。如果一个产品花费大把时间安装

**1.4本课题****主要研究工作**

本论文完成了基于 STM32 的环境检测，主要实现环境监控和远程监控等功能，其中涉及到多方

面的研究，如对 STM32 进行深刻的了解和学习才能进行下一步的硬件和软件设计，实际情况设计出相应的应用软件。研究、发展现状、目前存在的问题、未来的发展趋势和本文的主要研究内容。

1. 检测和控制系统方案设计 主要介绍了系统的设计过程和实现的主要功能。课题的设计要求，先初步设计出系统框架，再在功能的基础上，进行下一步的具体设计。

2. 系统硬件电路搭建 先介绍了主要芯片的选择，然后设计了系统整体硬件方案并针对各个模块进行详细介绍，分析各部分的功能及设计过程。最后简单介绍了电路PCB 版图设计。

3. 系统硬件驱动程序 先介绍了系统整体软件设计。接着介绍了软件开发环境 KEIL的配置，在此基础上，对主控制器及各个硬件模块进行驱动设计。

4. Android 应用软件的设计 先对 Android 系统进行简要介绍，然后叙述了如何搭建Android 开发平台。在前两者的基础上，对应用程序进行设计。分两部分介绍，其一描述了整体设计框架，其二具体介绍界面的设计和通信的实现。

5. 系统测试结果 这一章对检测和控制系统进行测试，各模块均正常工作。

**2** **系统的概述和相关原理**

**2.1 系统的概述**

**2.1.1 总体设计的方案**

本设计包括了主控制器、WIFI模块、温湿度监测、光照监测、PM2.5监测、手机客户端以及电源模块。电源模块为整个系统提供供电，主控制器模块通过串口协议与WIFI模块进行通信，利用串口发送AT指令使其与手机的wifi通信，通过STM32采集传感器检测到的数据，并且通过wifi发送到手机客户端，并可通过手机控制阈值，当数据值超过了阈值，单片机控制语音播报模块进行语音信息的播放。

该系统主要应具备以下功能：

（1）可实现环境监测系统的正常功能；

（2）Arduino手机app远程显示控制能够进行设置报警数值；

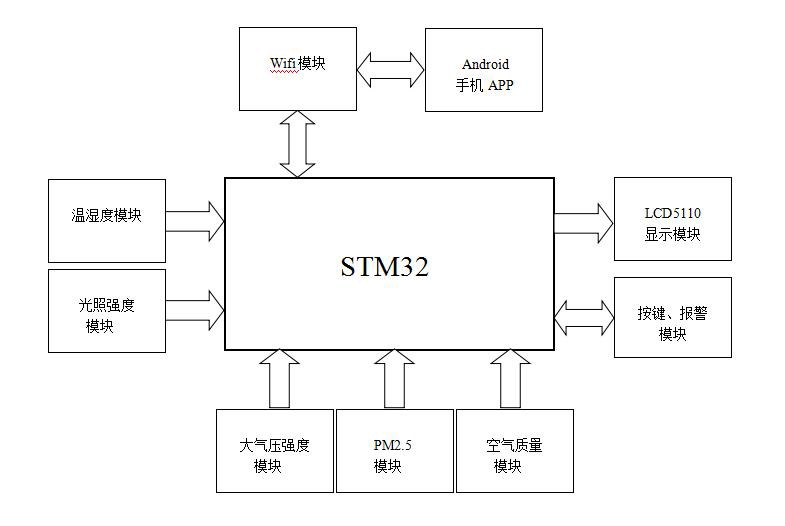
（3）STM32通过各种接口控制温湿度传感器，光照强度传感器，空气质量传感器，PM2.5传感器以及大气压强传感器，进而获取环境参数，并显示在LCD5110屏幕上；

（4）能够实现AC-DC降压电路给STM32系统供电

**2.1.2 总体框图**

该系统采用的是5V供电，主要是以STM32平台作为主控制器，，主控制器模块通过串口协议与WIFI模块进行通信，利用串口发送AT指令使其与手机的wifi通信，通过STM32采集传感器检测到的数据，并且通过wifi发送到手机客户端，并可通过手机控制阈值，当数据值超过了阈值，单片机控制语音播报模块进行语音信息的播放。

本系统总体框图如下：



**2.2 相关理论**

**2.2.1 STM32平台**

目前嵌入式系统中32位微控制器(MCU)的使用率正逐年增长,32位ARM体系结构己经成为一种事实上的标准，随着高端 32位嵌入式微处理器价格的不断下降和开发环境的成熟，使得 32 位嵌入式处理器正日益挤压原先由 8 位微控 制器主导的应用空间。随着 ARM 处理器在全球范围的流行，32 位的 RISC 嵌入 式处理器已经开始成为高中端嵌入式应用和设计的主流。使用 32 位架构不仅能 提升性能，还能降低相同成本下的系统功耗和节约总成本以及缩短产品上市时 间。并使得嵌入式系统可随着产品的性能和需求不断扩展而升级。

本文选用了一款基于 ARM Cortex-M3 内核芯片—由意法半导体（ST)推出的 STM32F103 芯片。STM32F103 系列微处理器是首款基于 ARMv7-M 体系结构的 32 位标准 RISC（精简指令集）处理器，很高的代码效率，在 8 位和 16 位系统的存 储空间上发挥了 ARM 内核的高性能。该系列微处理器工作频率为 72MHz，内置 高达128K 字节的 [Flash](http://www.dzsc.com/stock-ic/FLASH.html) [存储器](http://www.dzsc.com/product/searchfile/310.html)和 20K字节的[SRAM](http://www.dzsc.com/icstock/221/SRAM.html)，具有丰富的通用I/O 端口。

作为最新一代的嵌入式 ARM 处理器，它为实现 MCU 的需要提供了低成本的平 台、缩减的引脚数目、降低的系统功耗，同时提供了卓越的计算性能和先进的 中断响应系统。丰富的片上资源使得 STM32F103系列微处理器在多种领域如电机驱动、实时控制、手持设备、PC 游戏外设和空调系统等都显示出了强大的发 展潜力。

STM32F103 系列微处理器主要资源和特点如下：

1.多达 51 个快速 I /O 端口， 所有 I/O 口均可以映像到 16 个外部中断， 几 乎所有端口都允许 5V 信号输入。每个端口都可以由软件配置成输出（推挽或开 漏）、输入（带或不带上拉或下拉）或其它的外设功能口。

2.2 个 12 位模数转换器，多达 16 个外部输入通道，转换速率可达 1MHz,转 换范围为 0-36V，具有双采样和保持功能。内部嵌入有[温度传感器](http://www.dzsc.com/product/searchfile/1950.html)，可方便的测 量处理器温度值。

3.灵活的 7 路通用 DMA 可以管理存储器到存储器、设备到存储器和存储器 到设备的数据传输，无须 CPU 任何干预。通过 DMA 可以使数据快速地移动 CPU 的 资源来进行其他操作。DMA 控制器支持环形缓冲区的管理，避免了控制器传输 到达缓冲区结尾时所产生的中断。它支持的外设包括：定时器、ADC、SPI、I2C 和 USART 等。

1. .调试模式：支持标准的 20 脚 JTAG 仿真调试以及针对 Cortex- M3 内核的 串行单线调试（SWD）功能。通常默认的调试接口是 JTAG 接口。
2. .含有丰富的通信接口，三个 USART 异步串行通信接口、两个 I2C 接口、两 个 SPI 接口、一个 CAN 接口和一个 USB 接口，为实现数据通信提供了保证。

除工业可编程逻辑控制器(PLC)、家电、工业及家用安全设备、消防和暖气 通风空调系统等传统应用，智能卡和生物测定等消费电子应用外，新的 STM32 系列还特别适合侧重低功耗的设备，如血糖和血脂监测设备。

**2.2.2 WIFI模块**

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应用设计，可将用户的物理设备连接到Wi-Fi 无线网络上，进行互联网或局域网通信，实现联网功能。

ESP8266封装方式多样，天线可支持板载PCB天线，IPEX接口和邮票孔接口三种形式；

ESP8266可广泛应用于智能电网、智能交通、智能家具、手持设备、工业控制等领域。

* 支持无线802.11 b/g/n 标准
* 支持STA/AP/STA+AP 三种工作模式
* 内置TCP/IP协议栈，支持多路TCP Client连接
* 支持丰富的Socket AT指令
* 支持UART/GPIO数据通信接口
* 支持Smart Link 智能联网功能
* 支持远程固件升级（OTA）
* 内置32位MCU，可兼作应用处理器
* 超低能耗，适合电池供电应用
* 3.3V 单电源供电

|  |  |  |
| --- | --- | --- |
| **模块** | **型号** | ESP8266-01 |
| **主芯片** | ESP8266 |
| **无线参数** | **无线标准** | IEEE 802.11b/g/n |
| **频率范围** | 2.412GHz-2.484GHz |
| **发射功率** | 802.11b: +16 +/-2dBm (@11Mbps) |
| 802.11g: +14 +/-2dBm (@54Mbps) |
| 802.11n: +13 +/-2dBm (@HT20, MCS7) |
| **接收灵敏度** | 802.11b: -93 dBm (@11Mbps ,CCK) |
| 802.11g: -85dBm (@54Mbps, OFDM) |
| 802.11n: -82dBm (@HT20, MCS7) |
| **天线形式** | 外置：邮票孔接口 |
| 外置：I-PEX 连接器、SMA 连接器 |
| 内置：板载PCB 天线 |
| **硬件参数** | **硬件接口** | UART，IIC，PWM，GPIO，ADC |
| **工作电压** | 3.3V |
| **GPIO驱动能力** | Max：15ma |
| **工作电流** | 持续发送下=>  平均值：~70mA,峰值: 200mA  正常模式下=>  平均: ~12mA,峰值: 200mA  待机：<200uA， |
| **工作温度** | -40℃~125℃ |
| **存储环境** | 温度：<40℃，相对湿度：<90%R.H. |
| **尺寸** | 板载PCB天线：14.3mm\*24.8mm\*1mm； |
| **串口透传** | **传输速率** | 110-921600bps |
| **TCP Client** | 5个 |
| **软件参数** | **无线网络类型** | STA/AP/STA+AP |
| **安全机制** | WEP/WPA-PSK/WPA2-PSK |
| **加密类型** | WEP64/WEP128/TKIP/AES |
| **固件升级** | 本地串口，OTA远程升级 |
| **网络协议** | IPv4, TCP/UDP/FTP/HTTP |
| **用户配置** | AT+指令集, Web 页面 Android/iOS 终端, Smart Link 智能配置APP |

**3 硬件电路设计**

**3.1 主控电路**

STM32F103x8和STM32F103xB增强型系列使用高性能的ARM® Cortex™-M3 32位的RISC内核，工 作频率为72MHz，内置高速存储器(高达128K字节的闪存和20K字节的SRAM)，丰富的增强I/O端口 和联接到两条APB总线的外设。所有型号的器件都包含2个12位的ADC、3个通用16位定时器和1个 PWM定时器，还包含标准和先进的通信接口：多达2个I2C接口和SPI接口、3个USART接口、一个 USB接口和一个CAN接口。

STM32F103xx中等容量增强型系列产品供电电压为2.0V至3.6V，包含-40°C至+85°C温度范围和-40°C至+105°C的扩展温度范围。一系列的省电模式保证低功耗应用的要求STM32F103xx中等容量增强型系列产品提供包括从36脚至100脚的6种不同封装形式；根据不同的封 装形式，器件中的外设配置不尽相同。

在本设计中，采用STM32F103C8T6最小系统当做主控，图3-1为最小系统电路的原理图。

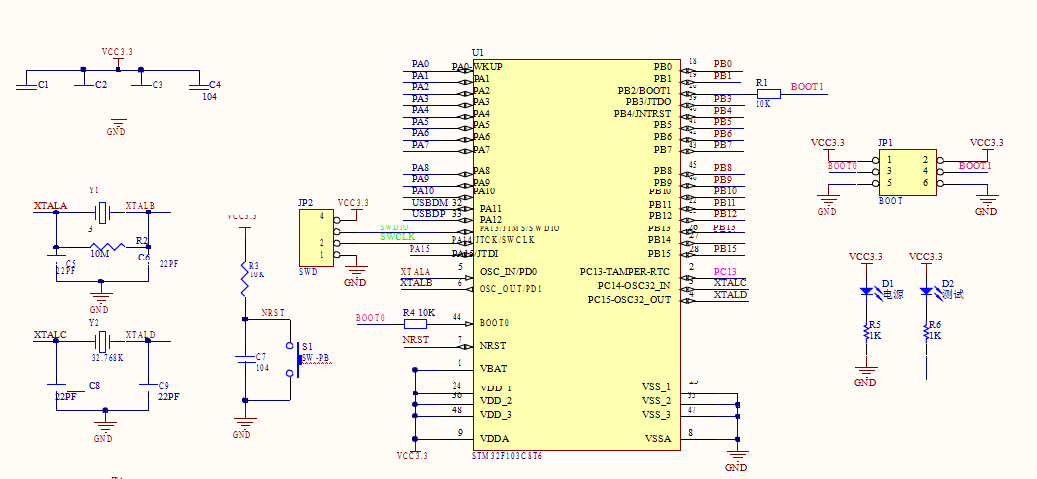


图3-1 STM32F103C8T6最小系统原理图

**3.2 WIFI通信模块**

WIFI模块又名串口WIFI模块，属于物联网传输层，功能是将串口或TTL电平转为符合WIFI无线网络通信标准的嵌入式模块，内置无线网络协议IEEE802.11b.g.n协议以及TCP/IP协议。传统的硬件设备嵌入WIFI模块可以直接利用WIFI联入互联网，是实现无线智能家居、M2M等物联网应用的重要组成部分。ESP8266是一款超低功耗的模块，拥有业内极富竞争力的封装尺寸和超低能耗技术，专为移动设备和互联网的应用设计，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。

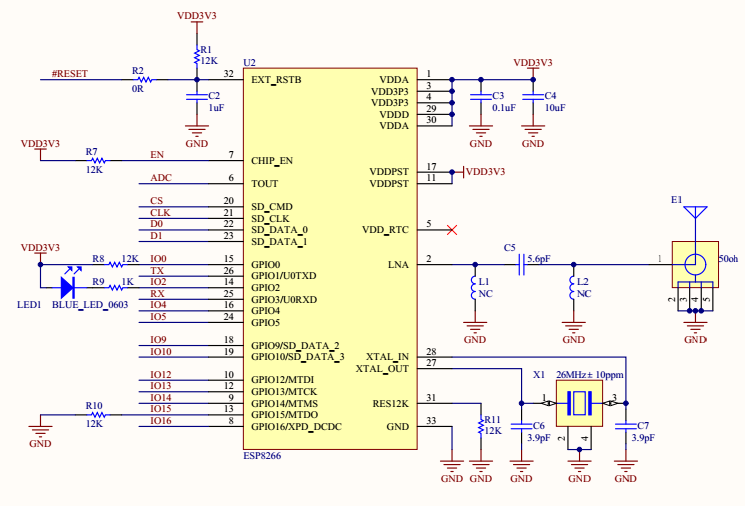


图3-2WIFI模块电路图

## 3.3报警模块

报警电路主要由蜂鸣器和LED组成，当环境数据超过阈值时，蜂鸣器报警，LED点亮。

该模块的电路连接图如下图3-3所示。

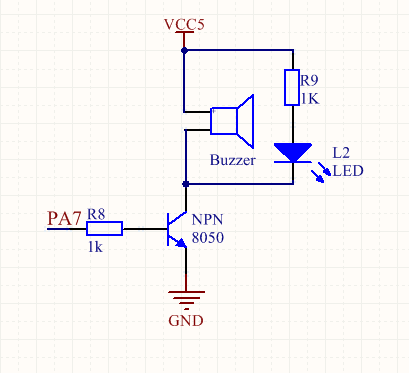


图3-3语音播报模块电路图

## 3.4温湿度模块

DHT11数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传 感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高 的可靠性与卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个NTC测 温元件，并与一个高性能8位单片机相连接。因此该产品具有品质卓越、超快 响应、抗干扰能力强、性价比极高等优点。每个DHT11传感器都在极为精确的 湿度校验室中进行校准。校准系数以程序的形式储存在OTP内存中，传感器内 部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集 成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达20米以上，使 其成为各类应用甚至最为苛刻的应用场合的最佳选则。产品为 4 针单排引脚 封装。连接方便，特殊封装形式可根据用户需求而提供。该模块在电路连接中图如下图3-5所示。

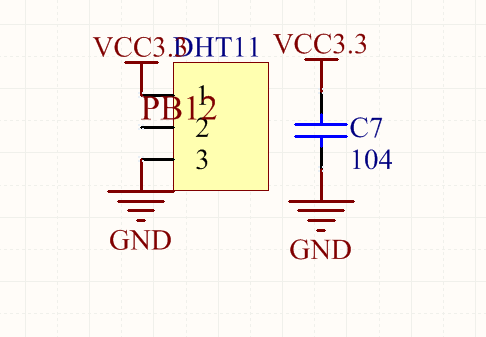


图3-4 DHT11模块电路图

## 3.5 光照强度模块

采用光敏电阻作为信号采集器件。光敏电阻是基于光电导效应的一种光电器件，无光照时, 光敏电阻值（暗电阻）很大, 电路中电流（暗电流）很小； 当受到光照时，半导体材料电导率增加，电阻减小。其阻值随光照增强而减小。光敏电阻作为光电式传感器的一种，它具有灵敏度高、光谱响应范围宽；体积小、重量轻、机械强度高、耐冲击、耐震动、抗过载能力强和寿命长等特点。所以选择光敏电阻采集光照信号，并把不同的光照强度转化为不同的电阻值。把光敏电阻串联在直流电路中即可把不同的电阻值转化为不同的电压值，电路如原理图所示。于是，就把对光照信号的处理转化为对电压信号V的处理。该模块在电路连接中图如下图3-5所示。

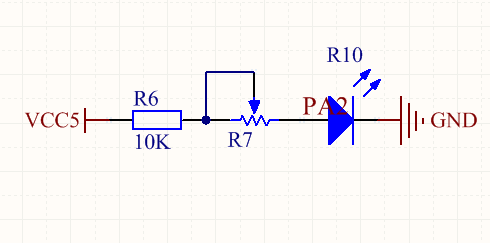


图3-5 光照强度模块电路图

## 3.6 PM2.5模块

粉尘采集电路采用夏普GP2Yl051AUOF空气质量传感器采集空气中的PM2.5颗粒，GP2Yl051AUOF 利用光电来测定空气质量的一个传感器，利用发光二极管发射光线，利用晶体管接收二极管发射的光引起电压的变化，这时候传感器里的空气传导设备就会产生一定的作用使空气流通，此设计能够检测到小的微粒，甚至非常小的烟草颗粒也能感应到。即使在一个地方测量到比较高的PM2.5值，到另外一个地方空气质量较好，PM2.5值低的地区测量的结果也是比较准确的。

GP2Yl051AUOF粉尘传感器的性能特点：

1 灵敏度：较高

2 其重量比较轻，能够很快的找到安装的方法

3 消耗的电量比较小一般采用5V的电压，有利于信号的储量过程

4 能够实现空气的流通，大大的增大了外部的大气的流通

5 保养简单，能够长时间的保持其模拟量，有利于AD转化

6 输出电压0. 5V

7 消耗电流相对比较小，降低了其功率达到11lmA;

该模块在电路连接中图如下图3-6所示。



图3-6 PM2.5模块电路图

## 3.7 空气质量模块

**MQ135气体传感器所使用的气敏材料是在清洁空气中电导率较低的二氧化锡(SnO2)。当传感器所处环境中存在污染气体时，传感器的电导率随空气中污染气体浓度的增加而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。MQ135传感器对氨气、硫化物、苯系蒸汽的灵敏度高，对烟雾和其它有害的监测也很理想。这种传感器可检测多种有害气体，是一款适合多种应用的低成本传感器。**该模块的电路连接图如下图3-7所示。

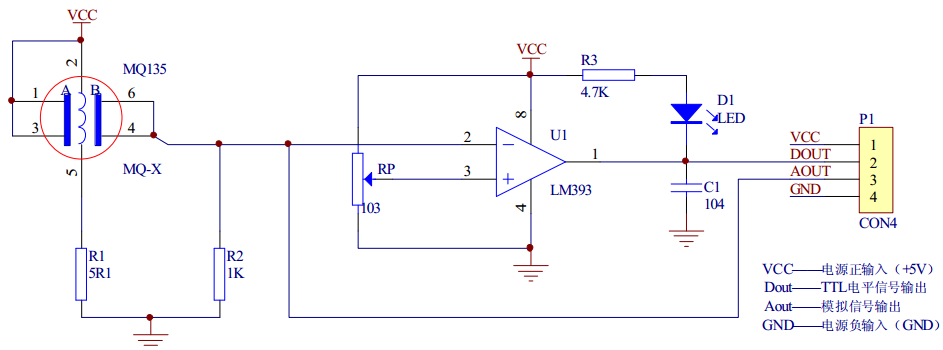


图3-7 空气质量模块电路图

## 3.8液晶模块

Nokia/诺基亚5110 LCD原理应用资料:LPH7366 是NOKIA 公司生产的可用于其5110、6150，6100 等系列移动电话的液晶显示模块，国内厂家也生产有类似的兼容产品。该产品除应用于移动电话外，也可广泛应用于各类便携式设备的显示系统。与其它类型的产品相比，该模块具有以下特点：

●84x48 的点阵LCD，可以显示4 行汉字，

●采用串行接口与主处理器进行通信，接口信号线数量大幅度减少，包括电源和地在内的信号线仅有9 条。支持多种串行通信协议（如AVR 单片机的ＳＰI、MCS51 的串口模式０等），传输速率高达4Mbps，可全速写入显示数据，无等待时间。

●可通过导电胶连接模块与印制版，而不用连接电缆，用模块上的金属钩可将模块固定到印制板上，因而非常便于安装和更换。

●LCD 控制器／驱动器芯片已绑定到LCD 晶片上，模块的体积很小。

●采用低电压供电，正常显示时的工作电流在200μA 以下，且具有掉电模式。 LPH7366 的这些特点非常适合于电池供电的便携式通信设备和测试设备中。

该模块的电路连接图如下图3-8所示。



图3-8 液晶模块电路图

## 3.9电源模块

该部分主要由电源线引入5V1A电源给系统供电，经ASM117降压为3.3V给STM32最小系统和3.3V的传感器供电。该模块在电路连接中图如下图3-9所示。



图3-9电源部分原理图

**4 软件设计**

**4.1 总体程序流程图**

该系统主要由STM32通过DMA采集多通道的ADC接口来连接光照强度、PM2.5模块、空气质量模块，采集到传感器的模拟量后，通过算法，进而得到光照强度、PM2.5和空气质量的值；并通过I2C和BMP180通信得到大气压的数值;通过单总线和DHT11通过得到温湿度的数值；采集的环境数据在LCD5110液晶上显示；并通过串口来与WIFI模块进行通信，通过串口配置与WIFI模块一样的波特率，然后发送特定的AT指令，可以对WIFI模块进行配置及控制，使得可以与手机进行通信，最后通过串口将采集传感器的数值经过WIFI模块发送到手机APP上，进而就可以实现在Android平台上，手机对环境数据的检测；最后，本作品还可通过手机或者按键对温湿度、光照强度、PM2.5的阀值做了设定，当数值超出这个界限的时候，语音播放模块将会发送语音信息来警告此时的数值已经超过了所设定的界限。

具体的工作流程为：

（1）初始化系统设置，包括初始化系统的时钟，设设置中断的优先级，初始化各模块,初始化串口1和串口3（设置通信的波特率位115200），初始化定时器3的频率为100HZ；

（2）STM32发送相关的AT指令，控制WIFI模块创造一个局域网；

（3）如果手机连接成功，就将STM32平台采集到的环境数据每30秒通过WIFI发送给手机，并且在手机端显示出当前的环境数值；

（4）用户可以根据自己的需求通过手机端按键设置报警的阀值，当环境数值超过设定的阈值是，主控制器就会控制语音模块播放出相应的报警信息。

其具体的工作流程图如下图4-1所示。



图4-1总体程序流程图

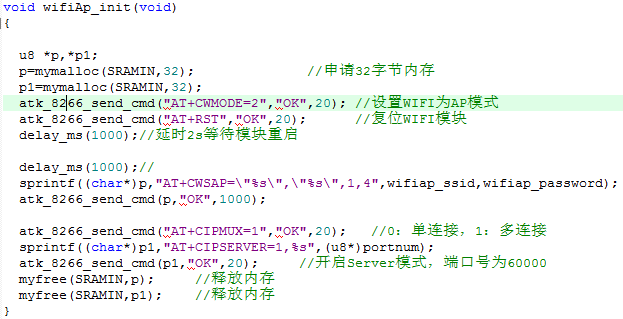
**4.2 WIFI模块**

ESP8266模块支持STA/AP/STA+AP 三种工作模式。

* STA 模式：ESP8266模块通过路由器连接互联网，手机或电脑通过互联网实现对设备的远程控制。
* AP 模式：ESP8266模块作为热点，实现手机或电脑直接与模块通信，实现局域网无线控制。
* STA+AP 模式：两种模式的共存模式，即可以通过互联网控制可实现无缝切换，方便操作。

本系统此次使用AP模式，实现局域网的无线控制。WIFI模块主要是负责系统与手机WIFI的通信，从而实现两者之间的数据传输。MCU与WIFI通信主要是通过串口来实现的，通过发送相关的AT指令，系统就可以完成对WIFI模块的操作以及相应的控制。

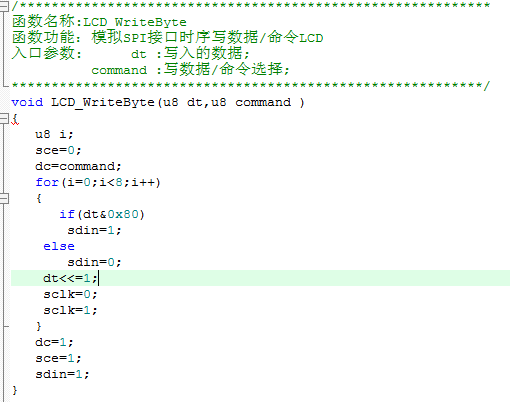
其实现的部分软件代码如下：



**4.3 LCD5110模块**

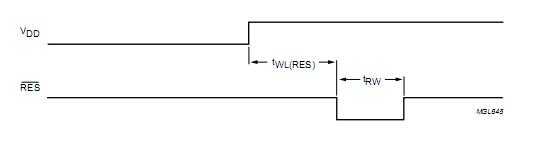
Nokia5110的通信协议是一个没有MISO只有MOSI的SPI协议，如果MCU有富裕的SPI接口，也可以利用硬件SPI，但通常没有必要，只需要软件程序模拟即可。本程序使用软件来模拟SPI,

程序如下：

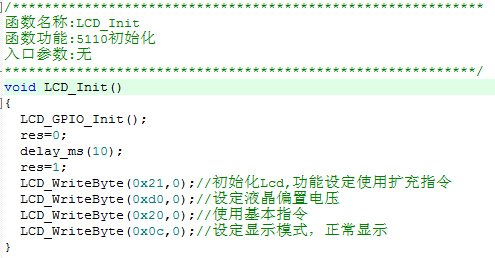


管脚D/C(LCD\_DC)用于选择写的是命令(D/C=0)还是数据(D/C=1)。

Nokia5110的初始化接通电源后，内部寄存器和RAM的内容是不确定的，这需要一个RES低电平脉冲复位一下。当VDD变为高电平，达到VDDmin(或更高)之后，最多100ms，RST输入低电平(电平幅度<0.3VDD)。

[](http://space.ednchina.com/upload/2010/4/12/a0dfd4b2-8a6d-4017-9d61-cf47ad114abd.JPG" \t "_blank)

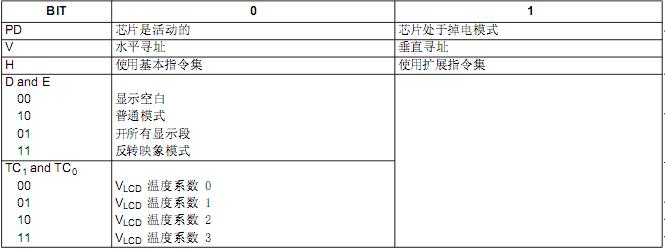
5110复位时序



设置Nokia5110液晶的坐标Nokia5110(PCD8544)的命令集：

[](http://space.ednchina.com/upload/2010/4/12/c66c94ee-4c72-4637-8f5c-30732d131d98.JPG" \t "_blank)

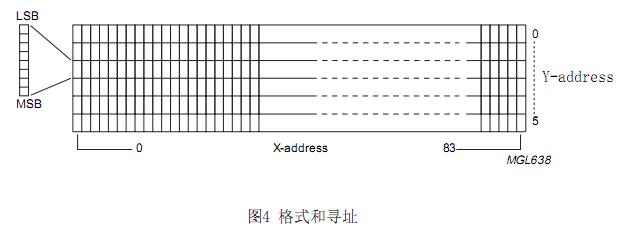
命令字符解释：

[](http://space.ednchina.com/upload/2010/4/12/0b3e87ac-4598-47ea-a906-ce66013958ca.JPG" \t "_blank)

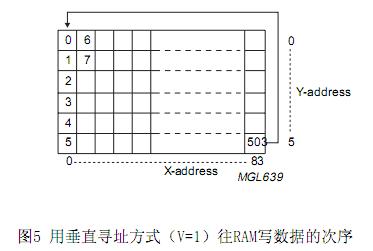
要注意的是“功能设置”命令的“H”和“V”。

H=0使用基本指令集命令，H=1使用扩展指令集命令。

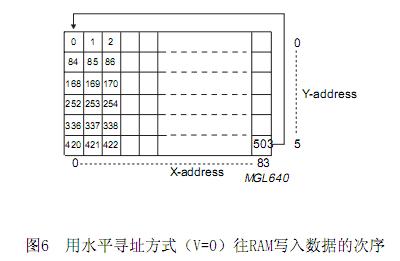
V=0水平寻址，V=1垂直寻址

[](http://space.ednchina.com/upload/2010/4/12/5bc89e17-628e-4e2a-ad9e-e2a6ece7b55e.JPG" \t "_blank)

列通过地址指针寻址，地址范围为X 0~83(1010011)，Y 0~5(101)。地址不允许超过这个范围。

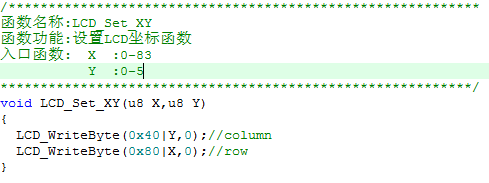


在垂直寻址(V=1)，Y地址在每个字节之后递增，经最后的Y地址(Y=5)之后，Y绕回0，X递增到下一列的地址。



在水平寻址模式(V=0)，X地址在每个字节之后递增，经最后的X地址(X=83)之后，X绕回0，Y递增到下一行的地址。

经每一个最后地址之后(X=83,Y=5)，地址指针绕回地址(X=0,Y=0)。



**4.4 BMP180模块**

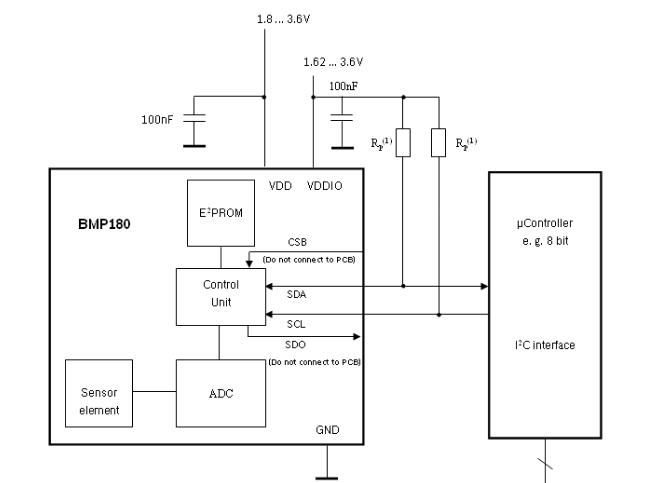
\* BMP180被设计成直接连接到经由I2C总线的移动设备的微控制器。

压力和温度数据具有由BMP180的E2PROM中的校准数据来补偿。

一般功能和应用原理图

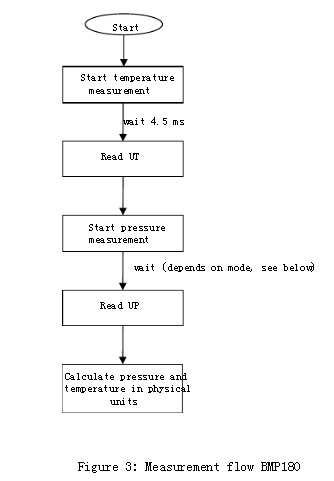
该BMP180由一个压阻传感器，模拟 - 数字转换器，并与E2PROM和一个串行I 2 C接口的控制单元。该BMP180提供的压力和温度补偿的值。 E2PROM中已存储的个人校准数据的176位。这是用于补偿偏移量，温度依赖性和传感器的其他参数。

UP=压力数据（16——19位）UT=温度数据（16位）

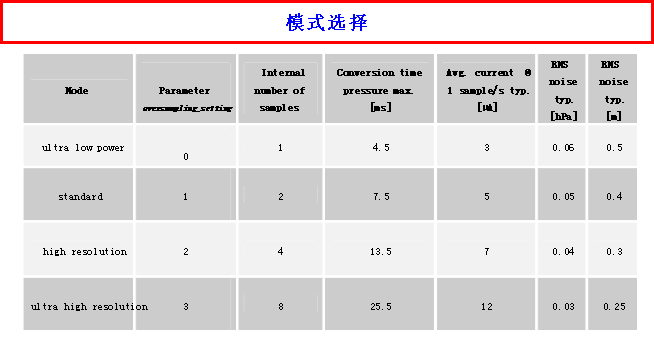


温度与压力的测量对于所有的计算这里给出一个ANSI C代码可从博世传感器（“BMP180\_API”）。

微控制器发送一个启动序列启动压力或温度测量。转换时间后，结果值（UP或UT）可通过I2C接口读取。用于计算温度℃并在帕斯卡的压力，该校准数据已被使用。这些常数可以从BMP180 E2PROM通过在软件初始化I2C接口被读出。 采样率可以提高到128个样本每秒（标准模式），用于动态测量。在这种情况下，它足以对测量温度只有每秒一次，在同一期间使用该值对于所有的压力测量。



硬件压力采样精度模式通过使用不同的模式下的功耗，速度和分辨率之间的最佳折衷，可以选择，见下表。

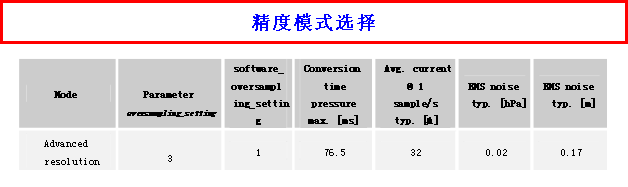


关于噪声特性的进一步信息，请参阅相关的应用笔记“噪音压力传感器应用”。所有模式都可以在更高的速度下进行，例如高达每秒128次为标准模式，与消耗电流成比例地增加的采样率。

软件件压力采样精度模式

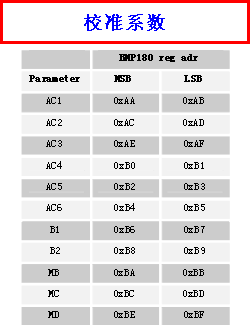
对于应用中的低噪声水平是至关重要的，平均化，建议如果低带宽是可以接受的。过采样可以使用该软件的API驱动程序（OSR=3）。

BMP180软件精度模式下，通过变量software\_oversampling\_setting通过驱动程序软件中选择模式。



校准系数

176位E2PROM划分在每个16位的11个字。这些包含11个校准系数。每个传感器模块具有独立的系数。温度和压力的第一个计算之前，主机读出E2PROM中的数据。数据通信可通过检查，没有的话具有值0或0xFFFF的检查。

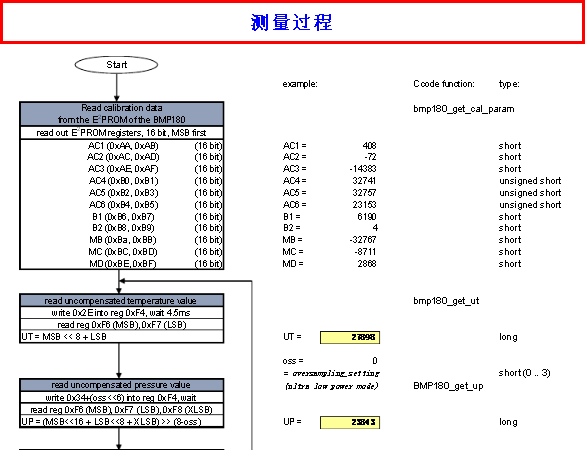


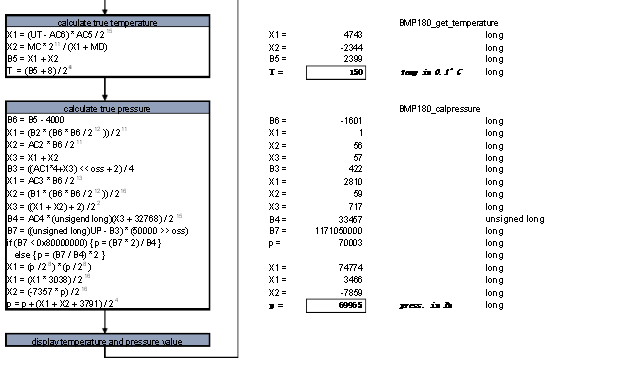
计算压力和温度

模式（超低功率，标准型，高，超高分辨率）可通过可变oversampling\_setting选择（0，1，2，3）。

真正的温度和压力的步骤为1pA（=0.01hPa=0.01mbar）和温度步长为0.1°C的计算

下图显示了详细的算法对压力和温度测量。

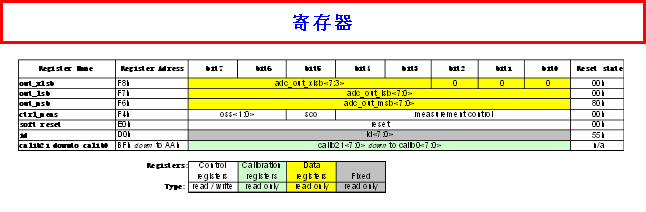




内存映射

下面的内存映射显示需要哪些操作BMP180所有的外部访问的数据寄存器。左边的列显示的内存

地址。中间的列描述每个寄存器位的内容。这些位的颜色表示他们是否只读，只写或读可写。存储器是易失性，使得该可写内容已被重新写入每次上电之后。



寄存器F4H 控制寄存器

Control(第0-4位）测量控制位

Sco (第5位）开始转换。该位的值保持为“1”的转换过程中和复位为“0”转换完成后（数据寄存器被填充）。

Oss (第6.7位）控制压力测量的过采样率（00B：单，01B：2次，10B：4倍，11B：8倍）。

寄存器E0H 软复位寄存器

只写 写0XB6复位

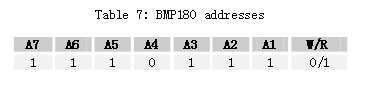
寄存器D0H ID寄存器

只读 值为固定值0X55

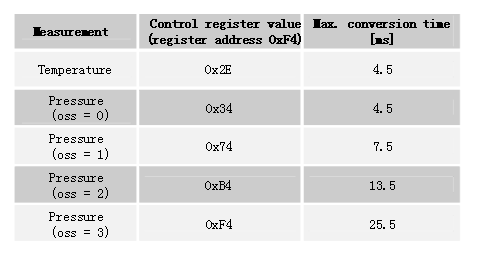
转换后，数据寄存器可以任意顺序读出（即MSB优先或LSB在前）。使用突发读取并不是强制性的。

4 I2C总线协议请参考其他资料

5 设备和寄存器地址

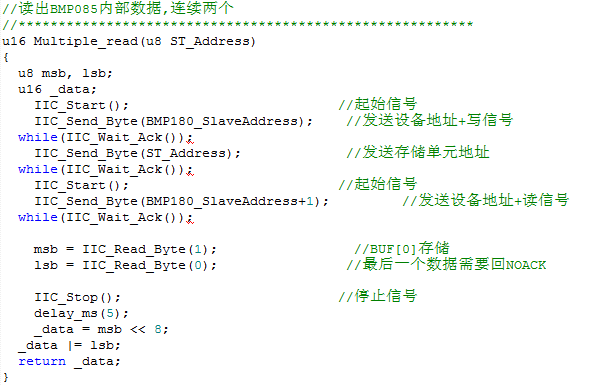


控制寄存器的值不同的内部oversampling\_setting（OSS）



读取A/ D转换结果或E2PROM数据读出温度数据字UT（16位），压力数据字UP（16—19位）和数据进行如下操作E2PROM中：启动条件之后，主机发送模块地址写命令和寄存器地址。该寄存器地址选择读寄存器E2PROM数据寄存器0xAA到0xBF温度值UT或压力值UP 0XF6（MSB）的0xF7（LSB），可选为0xF8（XLSB）然后主机发送一个启动条件和随后的模块地址读将由BMP180（ACKS）进行确认。该BMP180首先发送的 MSB，由主（ACKM）承认，那么 LSB。主机发送一个“不承认”（NACKM），最后一个停止条件。可选的超高分辨率，在XLSB注册地址为0xF8可读取16位字扩展到了19位。

部分程序如下：



**4.5 DHT11模块**

DATA 用于微处理器与 DHT11之间的通讯和同步,采用单总线数据格式,一次 通讯时间4ms左右,数据分小数部分和整数部分,具体格式在下面说明,当前小数 部分用于以后扩展,现读出为零.操作流程如下:

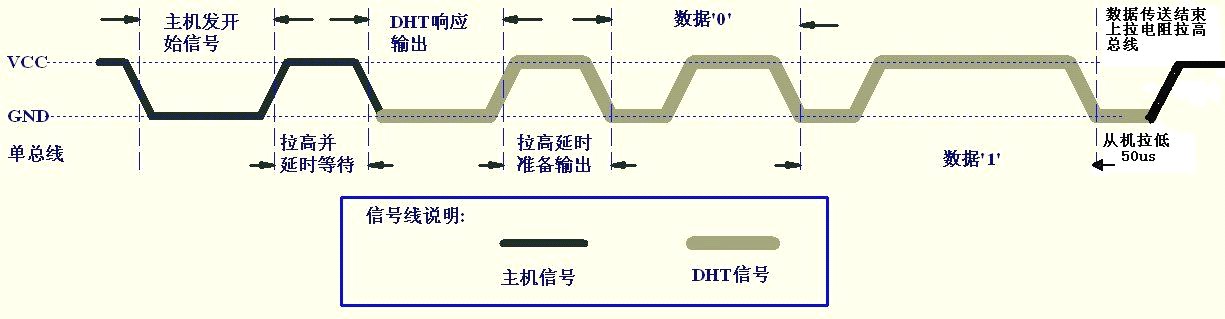
一次完整的数据传输为40bit,高位先出。 数据格式:8bit湿度整数数据+8bit湿度小数数据

+8bi温度整数数据+8bit温度小数数据

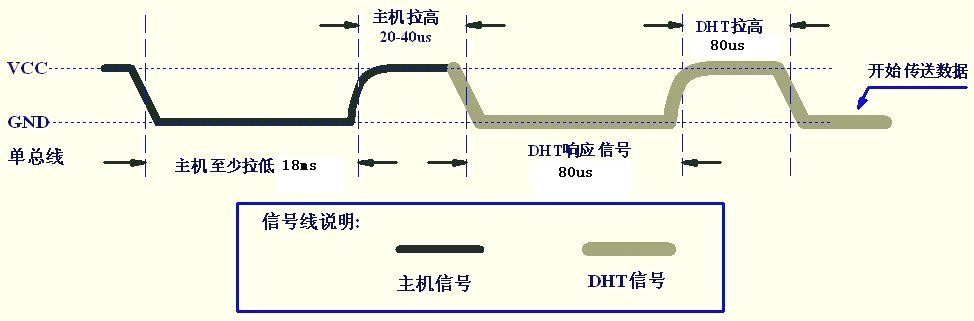
+8bit校验和

数据传送正确时校验和数据等于“8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据”所得结果的末8位。

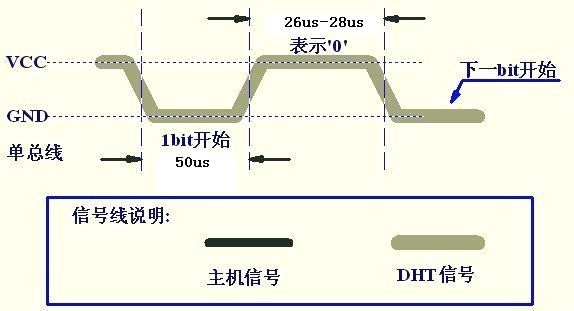
用户MCU发送一次开始信号后,DHT11从低功耗模式转换到高速模式,等待主机开始信号结束后,DHT1发送响应信号,送出40bit的数据,并触发一次信号采集,用户可选择读取部分数据.从模式下,DHT11接收到开始信号触发一次温湿度采集,如果没有接收到主机发送开始信号,DHT11不会主动进行温湿度采集.采集数据后转换到低速模式。

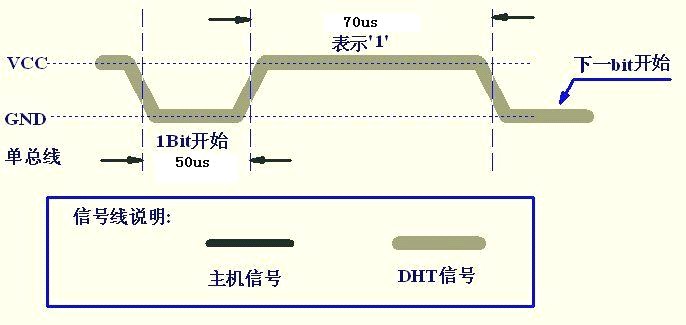
1.通讯过程如下图所示

总线空闲状态为高电平,主机把总线拉低等待DHT11响应,主机把总线拉低必须大于18毫秒,保证DHT11能检测到起始信号。DHT11接收到主机的开始信号后,等待主机开始信号结束,然后发送80us低电平响应信号.主机发送开始信号结束后,延时等待20-40us后, 读取DHT11的响应信号,主机发送开始信号后,可以切换到输入模式,或者输出高电平均可,总线由上拉电阻拉高。



数字0信号表示方法如下图所示



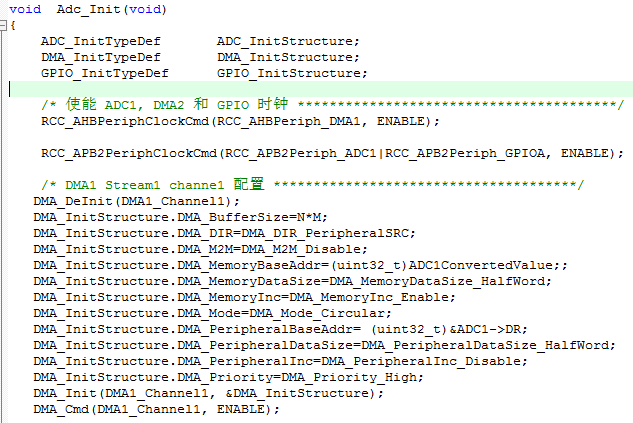
 数字1信号表示方法.如下图所示

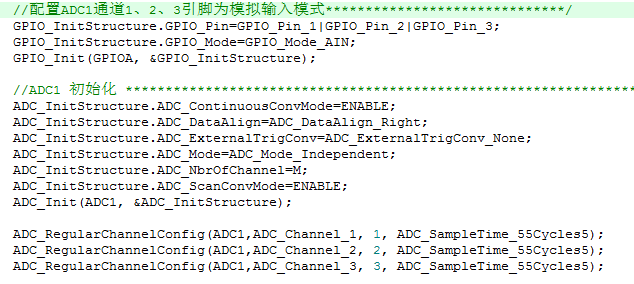
部分程序如下：



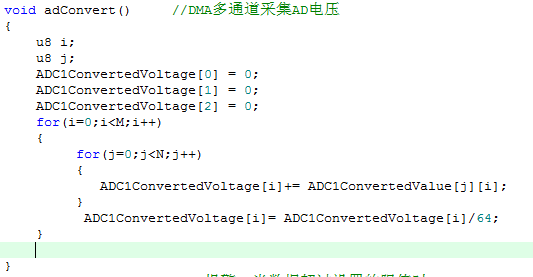
**4.7 ADC-DMA模块**

本系统采用DMA采集多通道ADC,DMA就相当与CPU的一个秘书，他的作用就是帮CPU减轻负担的。说的再具体点就是帮CPU来转移数据的。我们都知道，AD每次转换结束后会将转换的结果放到一个固定的寄存器里，以往我们如果想将该寄存器中的值赋给某一变量时会用到赋值语句，如果不用DMA，则赋值语句便要CPU来完成，CPU本来就要忙着处理其他事情，现在还要来解决赋值语句这么简单的问题，会影响CPU效率。所以需要DMA这个秘书来帮他解决这个问题。下图就是初始化DMA和AD来开启多通道ADC:





采集到AD后将采集的数据放到数组ADC1ConvertedValue[N][M]中，然后经过算术平均算法的到模块具体所用的AD数值，程序如下：



得到的数值在经过各个模块具体的算法最终得到用户所看到的各传感器数值，程序如下：

空气质量MQ135模块：



光照强度模块：



PM2.5模块：



**5** **Android系统**

**5.1 Android系统简述**

“ Android”来源于一名法国作家的科幻小说，意思是“外表像人的机器”，在国内通过音译取名“安卓”。 Android公司在成立 22 个月的 2005 年被互联网巨头Google（谷歌）收购。至 2007 年， Google宣布与软件开发商、硬件制造商、电信营运商组建开放手机联盟，并发布了Android移动操作系统[50]。在开发团队的努力下，加入Android阵营的手机厂商越来越多，如HTC、三星、 LG等， Android的市场占有率也越来越高。特别在国内，曾经占有绝对优势的塞班系统，原有的市场份额也逐渐被Android替代。 Android手机操作系统能够在短时间内获得成功，主要有以下几个优势：  
①开放性。 Android 是基于 Linux 平台开发的手机系统，是市场上难得的一款完全免费的开源平台。开源是吸引众多手机制造厂商选择 Android 的重要原因，也在一定程度上促进了手机开发领域的技术创新。  
②资源丰富。 Android 操作系统提供了大量免费的应用程序，如搜索、地图、浏览器等，开发人员可以合理利用资源进行设计。同时，官方网页 http://developer.android.com/index.html也为开发提供详尽的资料，不仅可以方便下载到系统不同版本的开发包，还可以在 Reference（参考）中找到各个组件的介绍与范例，更是准备了各种开发工具供设计者使用。  
③不受运营商限制。早前的手机市场，无论在国内还是国外，运营商往往决定了手机究  
竟使用什么样的网络，这样显得十分被动。随着 3G网络的到来，且运营商同样加入了开放手机联盟， Android手机的网络不必受制于运营商[51]，有了更加开阔自由的开发环境。  
5.1.1 Android系统架构  
Android系统的内部架构图如图 6-1 所示，架构清晰、层次分明，自上而下可以分为应用程序层、应用程序框架层、 Android运行时库及系统核心库层、 Linux内核层四层[52]：  
 应用程序层  
Android 内部包含大量丰富、便捷的应用软件，如通讯录、短消息、浏览器、电子邮件等，这些程序均放在应用程序层。该层中的所有程序都由 JAVA 编程语言实现，也是在最大程序体现面向对象的设计理念。  
 应用程序框架层  
该层用于存放开发人员在开发过程中调用的一些标准接口[53]。开发人员可以平等地使用  
内部框架提供的应用接口，也可以依据硬件基础、后台服务等，自行设计适合的应用程序框架。各类管理器组成了Android系统这个强大的框架层。  
 Android 运行时库及系统核心库层

Android的运行时库主要由核心库和虚拟机组成。核心库不仅包含了像Android.net等核心API，还为JAVA提供了大部分需要调用的功能函数。 Dalvik虚拟机提供了可以同时运行的多个虚拟系统的功能[54]。系统库是内核层与应用框架层进行联系的桥梁， 这些库透过框架层为开发人员提供函数库给应用程序的组件使用，这些库多数为C或C++语言。  
 Linux 内核层  
Android 是基于 Linux 的操作系统。从图中可以看出，内核层提供了显示、蓝牙、视频等模块的驱动。除此之外，该层还负责内存、进程、电源管理等，是其他层强有力的支撑。

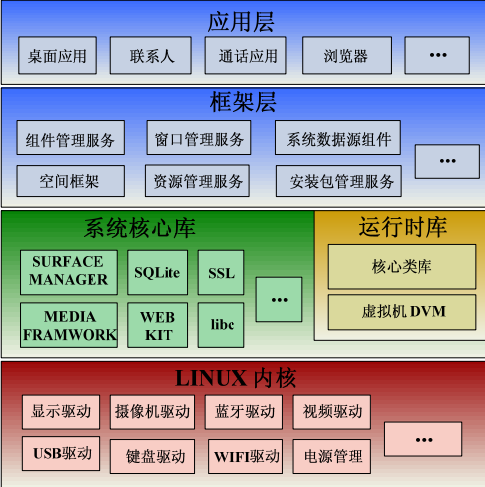


图 5-1 Android 系统内部架构图

6.1.2 Android组件模型  
组件是 Android 应用程序的基本组成， 经过内部联系将这些看似零散的组件组成一个个应用程序。四大组件如下所介绍：  
 Activity  
Activity是最直观体现在用户面前的组件。在一个界面上，依据具体需求放置系统提供的控件。一个应用系统可以有一个Activity，也可以有多个Activity，但通常有一个Activity标记成默认开启的界面。 Activity不仅在本程序中完成工作，同时允许被其他程序调用[55]。  
 Service  
Service 可以理解成运行在后台的事件，本身没有显示的界面，需通过其他组件进行调用。平时的使用中遇到下载文件时，依然可以执行其他操作，这便是 Service。音乐在后台播放时便是继承了 Service 基类的服务。  
 Content Provider

Content Provider 可译为内容提供器，为 Android 实现了共享功能，为其他程序存储或读取本程序共享数据提供了一种标准接口。 Android 提供了多种存储方式，如 SQLite 数据库等。内容提供器将共享的数据放入存储文件中，其他程序通过系统提供的 Content Resolver 提取或放入数据。  
 Broadcast receiver  
Broadcast receiver，广播接收器，是用来接收和处理广播信息。当它接收到消息后，通过声音、震动、图标来提醒用户。该组件也没有界面显示，可以调用 Activity 等来响应。应用程序需要通知其他程序的时候，也可以使用广播完成通知。

**5.2 Android开发环境的搭建**  
要进行 Android 应用程序开发，必需有 Android 开发环境的支持，搭建开发环境是必不可少的一步。主要分为以下几个步骤：  
 Java 开发环境的搭建

Android 应用程序开发依赖于 Java 语言， Java 是 Sun（ Stanford University Network）公司于 1995 年推出的跨平台面向对象的程序语言与 JavaSE、 JavaEE、 JavaME 集合平台的总称。Java 开发环境的搭建主要有三个步骤：a） 下载 JDK（ Java Development Kit）。先登录甲骨文官方网站： http://www.oracle.com/，  
在 Downloads 目录下选择【 Java for Developers】—【 Java Platform（ JDK）】，在此页面下根据电脑的配置选择对应的 JDK 下载。  
b） 安装 JDK。与普通应用软件的安装类似，按照页面提示一步步执行。完成 JDK 安装后，会自动跳出 JRE 安装页面。 JRE（ Java Runtime Environment）是 Java 的开发环境，一样按提示完成安装。安装路径中不要出现中文字符和空格。  
c） 设置环境变量。右键进入【我的电脑】—【高级】—【环境变量】，在系统变量中设置三个变量：①找到 Path 变量，输入之前安装的 JDK 中 bin 的路径（与其他路径用“ ;”隔开），若没有该变量则新建一个。此变量使系统能够识别 Java 命令；②以同样方法将 JDK 根目录设置在“ JAVA\_HOME”变量内，以此指明 JDK 的安装路径；③在变量“ CLASSPATH” 中指明 Java 类库的路径，可以只输入“ .”，系统便在之前的 Path 路径中搜索，也可以将具体路径输入。经过以上步骤， Java 环境搭建完毕。 Windows 系统可以通过命令提示符来确认安装情况。如果输入“ java -version”能正确显示 Java 版本信息，输入“ JAVAC”能显示 JAVAC 的用法  
说明安装正确，如图 5-2 所示：

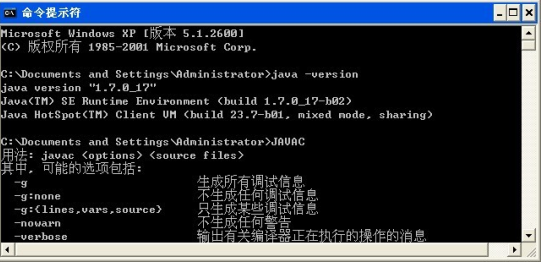


图 5-2 命令窗口确认 Java 安装情况

 Android SDK 的安装  
SDK（ Software Development Kit）为开发提供了类库的支持，安装可分为以下几个步骤：  
a） 下载 SDK。进入官方开发网站： http://developer.android.com/index.html，点击【 Get theSDK】—【 Download the SDK】进入下载页面，同意官方的条款并选择计算机的位数，变可以下载 SDK 了。  
b） 安装 SDK。本课题选择在线安装，解压下载的 SDK 安装包，在文件夹中找到 SDK  
Manager.exe 并双击执行，等待安装完毕。  
c） 安装好 SDK，选择所需的 Android 版本及 Google 提供的一些工具执行下载安装，这部分耗时较长。本课题下载的版本为 Android 4.2.2（ API 17）。  
 Eclipse 及 ADT 插件的安装



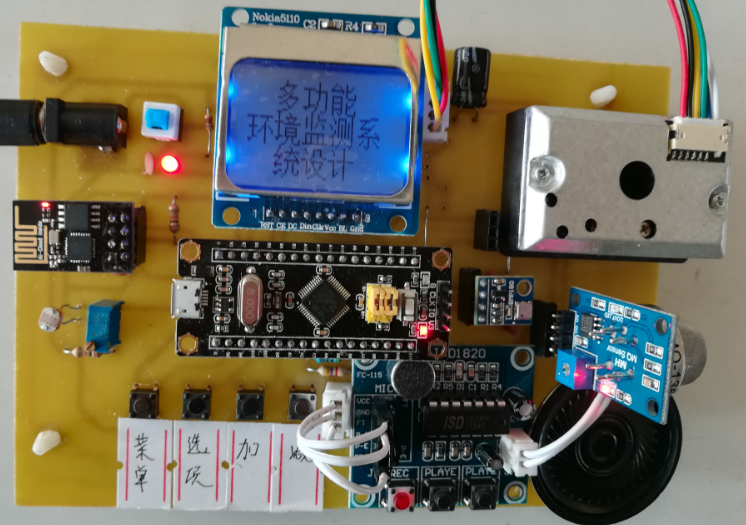
图 5-3 Android 模拟器界面

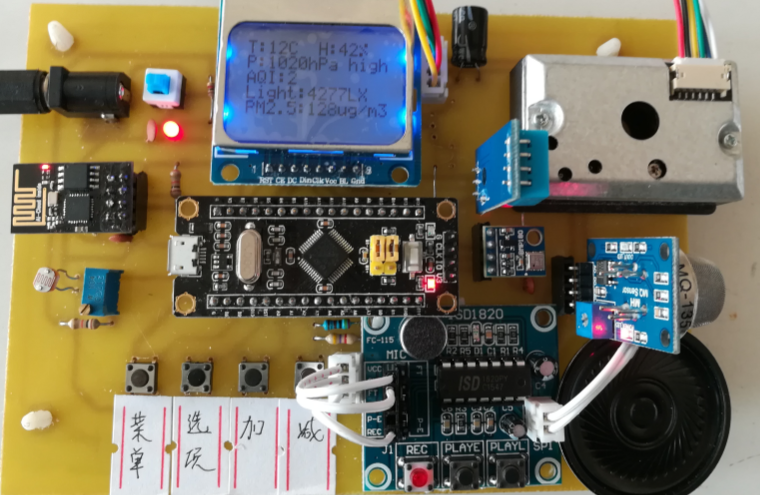
之前下载的 SDK 安装包也包含了 Eclipse 的文件夹，可以直接使用该文件夹中的  
Eclipse.exe。打开 Eclipse 后，先配置工作区（ Workplace）。 Eclipse 可以安装多种插件，其中ADT 是 Google 提供的开发 Android 的工具。点击菜单栏里的 Virtual Device Manager（虚拟设备管理器），创建新的模拟器：设置好名字、设备 CPU、内部存储空间大小、 SD 卡大小等点击确定。之后就可以使用该模拟器来模拟 Android 手机来实现开发，图 6-4 显示了该虚拟器的界面，左边为 Android 手机界面，右边为键盘：经过以上几步，本课题完成了 Android 开发环境的搭建。

**6 运行与调试**

**6.1 测试**

在硬件调试的过程中，利用AD绘制完原理图，根据原理图设计好电路板后，对板进行了检查，确认整体的电路设计没有问题后，确定无误后再进行元器件的焊接。焊接完成后，对焊接的地方进行检查，检查是否有虚焊或者漏焊，确认没问题后。再针对电源供电部分进行检查，看MCU等各个部分的供电是否正常。然后下载之前在开发板上调试过的程序，看是否在本系统中是否能正常运行。在软件调试时，利用MDK5进行了程序的开发，分步进行了各个模块的初步调试，待每个模块都可完成基本的功能后，再对整体的功能进行调试，最后再和Android进行通讯测试。虽然在调试的过程中遇到了一些问题，但是经过各种排查和解决，终于达到了设定的目标功能。







# 

# 结论

在信息高度发展的今日社会，人们对环境的要求比过去有了大幅度的提高，期望能够拥有更安全的环境。本文先介绍了物联网在国内外的发展状况、目前存在的问题和未来的发展方向。在此基础上设计了整体架构，并最终完成环境检测系统。经过实际测试，主控制器能够准确地测量环境变量。 整个系统工作稳定、准确。

将本文完成的工作总结如下：  
（一） 整个检测和控制系统分为主控制器和客户端两部分。主控制器用于实现硬件操作和WIFI通信，包括硬件设计、硬件驱动和通信协议三个部分。客户端由 Android 应用程序实现，为用户提供远程监控的界面。  
（二）主控制器的硬件设计主要包括 STM32 最小系统、电源模块、WIFI模块、大气压模块、光照强度模块、空气质量模块、PM2.5模块、温湿度模块、语音模块和显示模块。传感器负责测量环境温度、湿度、光照度、大气压、PM2.5和空气质量，测量结果精度、准确率都很高；语音可输出报警的声音，使系统更加人性化。  
（三） STM32 拥有丰富的外设资源，在驱动设计中使用到了 GPIO、 USART、 ADC、 I2C等。对于一些简单的功能，使用宏定义来代替函数提高了代码的可修改性和可读性。  
（四） Android 客户端用于远程控制和检测环境。主控制界面通过 Socket 实现与主控制器的通信，由输入流获得命令并解析为相应的消息进行处理。 Socket 通信运行于单独的线程中，减少了主线程的负担并保证了通信质量。由于在设计初期专业技能知识不够全面，以及时间和个人能力的限制，系统在整体上依然存在一些不尽如人意的地方，有些问题需要进一步改进。

**致谢语**

衷心感谢许老师对我耐心的指导和悉心的帮助，无论是在选择课题上还是在毕业设计的实现上以及在论文的撰写定稿方面，他都给予了我极大地帮助和指导。在这里我想向许老师致以最崇高的敬意！他渊博的学识、严谨的工作态度、务实的工作作风以及认真细致都给我留下了极为深刻的印象，而我在今后的工作中都将以许老师为榜样。同时，也感谢在毕业设计过程中给予过我帮助的同学们。

最后，衷心地感谢在百忙之中来评阅论文和参加答辩的各位老师!

参考文献

[1]尹育新,戎宏娜.浅谈一种全新的网络技术一物联网[J]．信息系统工程,2OlO(2)：105-106

[2]王宝云.物联网技术研究综述[J].电子测量与仪器学报,2009,(12):1-7.

[3]邵文佳.物联网技术应用展望[A].企业技术开发,2010年7月，第29卷第14期.

[4]徐涛.物联网发展现状及问题研究[A].中国市场,2010年第32期。

[5]古丽萍.对于我国物联网应用与发展的思考[N].通信世界周刊,2009—10-27.

[6]王光辉.物联网战略的国际观察与思考[EB/OL].(2OLO-O6-22)

[7]郭川.中国工程院副院长邬贺铨:物联网不是网络是应用[N].人民邮电报,2010-3-1. [8]张毅,唐红.重庆邮电大学通信与信息学院:物联网综述[J].数字通信,2010

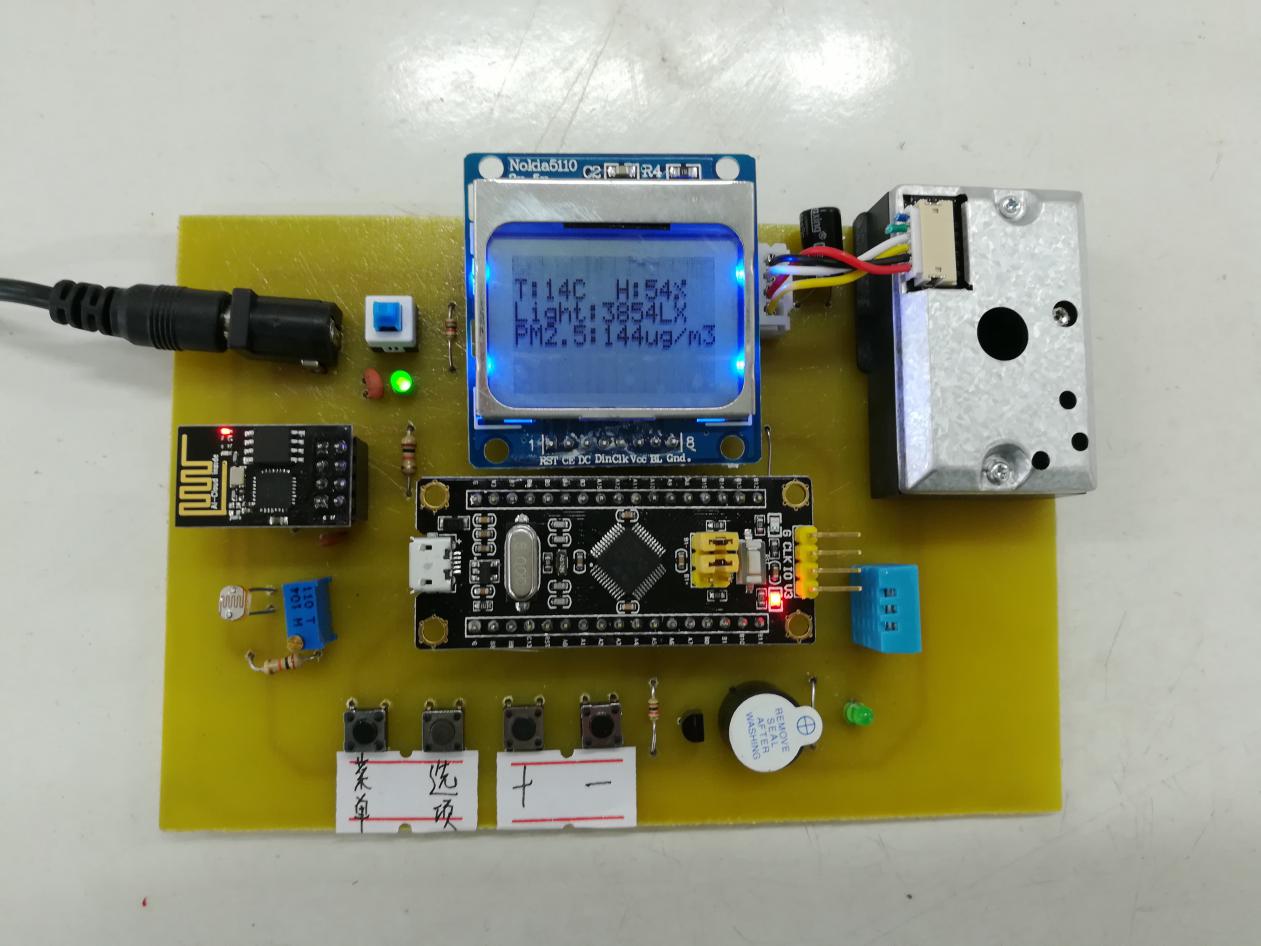
[8]李光辉,赵军,王智. 基于无线传感器网络的森林火灾监测预警系统[J]. 传感技术学报. 2006(06)

附录一：总体电路图





附录2 实物图



附录三：程序部分代码

int main(void)

{

delay\_init();

NVIC\_Configuration();

TIM\_Test(); //定义频率为100HZ的定时器

uart\_init(115200); //初始化串口

usart3\_init(115200);

IIC\_Init(); //初始化I2C

DHT11\_Init();

Adc\_Init(); //初始化ADC

KEY\_Init();

LCD\_Init();

LCD\_Clear();

lcd\_start();

wifiAp\_init();

sendWifi=1;

LCD\_Clear();

while(1)

{

if(COUNT%200==0) //500ms更新一次数据

{

adConvert(); //读取AD值

DHT11\_Read\_Data(&temperature,&humidity); //读取温湿度值

}

key\_service(); //按键服务

display\_service(); //显示界面

Beep();

if(COUNT>=4000)

{

if(sendWifi==1)

{

LCD\_Write\_Char(8,5,'s');

LCD\_Write\_Char(10,5,'e');

LCD\_Write\_Char(12,5,'n');

LCD\_Write\_Char(14,5,'d');

LCD\_Write\_Char(16,5,'.');

LCD\_Write\_Char(18,5,'.');

LCD\_Write\_Char(20,5,'.');

LCD\_Write\_Char(22,5,'.');

atk\_8266\_send\_cmd("AT+CIPSEND=0,14","OK",200); //发送数据指令

atk\_8266\_send\_data(TxBuffer,"OK",400); //发送指定长度的数据

LCD\_Write\_Char(8,5,' ');

LCD\_Write\_Char(10,5,' ');

LCD\_Write\_Char(12,5,' ');

LCD\_Write\_Char(14,5,' ');

LCD\_Write\_Char(16,5,' ');

LCD\_Write\_Char(18,5,' ');

LCD\_Write\_Char(20,5,' ');

LCD\_Write\_Char(22,5,' ');

}

COUNT=0;

}

}

}