**湘潭大学**

**毕业设计说明书**

**题 目：基于单片机的GPS定位系统设计**

**学 院： 信息工程学院**

**专 业： 通信工程**

**学 号： 2014551003**

**姓 名： 张薇**

**指导教师： 印峰**

**完成日期： 2018年5月**

目 录

[摘要 I](#_Toc18589)

[Abstract II](#_Toc28658)

[第1章 系统方案选择和论证 1](#_Toc7133)

[1.1 引言 1](#_Toc29834)

[1.2 设计背景 1](#_Toc20134)

[1.3 方案说明 2](#_Toc26526)

[1.3.1 原理说明 2](#_Toc27584)

[1.3.2 系统功能说明 5](#_Toc23720)

[1.4 模块设计 5](#_Toc19172)

[1.4.1 主控制模块设计方案 5](#_Toc16966)

[1.4.2 显示系统选择方案 6](#_Toc26125)

[1.4.3 GPS模块选择方案 6](#_Toc11361)

[1.5 整体方案的选择 7](#_Toc28087)

[1.5.1 总体设计框架 7](#_Toc1966)

[1.5.2 硬件方案 7](#_Toc31208)

[1.5.3 软件设计与编程 8](#_Toc15246)

[第2章 系统硬件的设计与实现 10](#_Toc25485)

[2.1 主控芯片电路 10](#_Toc17130)

[2.2 电源电路 11](#_Toc1077)

[2.3 复位电路 11](#_Toc12769)

[2.4 晶振电路 12](#_Toc25887)

[2.5 GPS模块电路 12](#_Toc9171)

[2.6 LCD1602显示电路 14](#_Toc30436)

[2.7 整体原理图及实物图 15](#_Toc18135)

[第3章 系统软件的设计与实现 17](#_Toc22396)

[3.1 程序框图 17](#_Toc20547)

[3.2 代码详述 17](#_Toc8502)

[3.2.1 头文件 17](#_Toc2615)

[3.2.2 IO口设置 18](#_Toc21146)

[3.2.3 延时函数 18](#_Toc27249)

[3.2.4 1602初始化函数 18](#_Toc7769)

[3.2.5 GPS解析函数 19](#_Toc25866)

[第4章 系统整体测试及结果 22](#_Toc11184)

[4.1 测试过程 22](#_Toc451)

[4.2 测试问题及解决方案 22](#_Toc21225)

[4.3 测试结果 22](#_Toc12553)

[5总结与展望 24](#_Toc32480)

[5.1 总结 24](#_Toc1507)

[5.2 展望 24](#_Toc25526)

[5.3毕业要求指标点达成情况分析 24](#_Toc25687)

[6致谢 27](#_Toc31769)

[参考文献 28](#_Toc28693)

[附录 A 程序 30](#_Toc2504)

**基于单片机的GPS定位系统设计**

**摘要：**随着手机应用软件的普及，AI智能产品如智能家居、VR眼镜、智能手表等智能穿戴产品也层出不穷，成为21世纪发展热潮。人们在体验到这些技术给生活带来了极大地方便的同时，应该也要考虑如何将这些技术融入到生活中，结合现在的社会现状以及存在的社会问题，这是我们研究和发展硬件技术的意义之一，用技术改变生活。针对目前大多数定位系统都为手机定位软件应用，使用时需要实时开启手机GPS定位，下载城市地图等不便之处，结合智能手环可计算步数，测试心率等特点，设计一款实时显示位置的智能定位硬件系统，更方便人们的出行。本次设计开发一个成本低的，方便携带、操作方便的基于单片机开发的一套GPS定位系统，主要完成电源电路、晶振电路、复位电路、GPS模块电路和显示电路，利用GPS模块接受模块卫星数据进行处理，获取当前时间和经纬度，在系统上通过LCD1602显示，提供方位与时间，此系统可运用于智能手环产品，通过将经纬度转换为地图信息，实现实时定位并显示可查询时间功能，成为真正的掌上地图应用。本文从系统设计框架开始，完成硬件设计，器件芯片的选择，原理图的绘制，软件程序设计，整体系统的调试与问题分析，结合各方面要求完成一个单片机定位系统，并讨论研究加深定位精度的方法与措施。

**关键词：**定位；单片机；显示屏

**GPS Positioning System Design**

**Based on Micro Control Unit**

**Abstract:** With the popularity of mobile application software, AI smart products,such as smart home, VR glasses and smart watches are also emerging, becoming a hot trend in the 21st century.People bring to the life in the experience of these technologies have greatly convenient at the same time, should also consider how these techniques can be incorporated into to life, combined with the present social situation and the existing social problem, it is one of the significance of research and development of hardware technology, using technology to change life.Aiming at most of the positioning system for the mobile phone positioning software applications, the need when using GPS real-time open mobile phones and download the city map, such as inconvenience, combined with intelligent bracelet steps can be calculated, the characteristics of heart rate, design a real-time display the location of the intelligent hardware system, people travel more convenient.The design and development of a low cost, convenient to carry, easy to operate a GPS positioning system based on single chip microcomputer development, the main power supply circuit, crystals circuit, reset circuit, a GPS module circuit and display circuit, using GPS satellite data processing module accept module, access to the current time and latitude and longitude, and through LCD1602 display in the system, provide the location and time, this system can be applied to intelligent hand ring products, by latitude and longitude into map information, real-time positioning and display can query time function, become the real palm maps.In this paper, starting from the system design framework, complete hardware design, device chip selection, schematic drawing, software program design, the overall system debugging and problem analysis, combined with various aspects required to complete a single chip positioning system, and discuss the positioning precision of the methods and measures.

Keywords: GPS; MCU; 1602

第1章 系统方案选择和论证

## 1.1 引言

贴合社会发展，结合当今社会生活现状，针对目前大部分定位软件都应用于手机上，而智能穿戴产品成为未来发展趋势，设计一款基于目前主流的定位系统GPS的定位硬件系统，给出解决社会问题的实际解决方案，希望能够更利于人们的生活。本文主要介绍研究本次课题的原因，所要解决的社会问题，使用到的技术，以及设计的开发过程和系统使用说明。还有总结设计开发过程中遇到的困难，以及实际解决问题方法思路。

目前，美国的GPS、欧洲的“伽利略”、俄罗斯的“格洛纳斯”和中国的北斗四大定位系统齐聚太空[3]。另有基站定位，wifi定位及蓝牙定位等定位技术，其中基站定位主要用于手机服务，wiFi定位及蓝牙定位为室内定位，本设计为用于室外的手环定位扩展系统，采用应用较广、参考文献多的GPS定位技术。通过GPS模块，接收卫星的定位信息并进行数据处理，提取模块所在位置的经纬度信息和当地时间，可以实现低成本、低功耗完成精准的导航定位。本论文以GPS技术为导向，基于单片机系统设计GPS定位系统，并显示屏实时显示用户所在经纬度及时间，由于GPS定位精度较高，结合wiFi定位或蓝牙定位等，得到更精确的位置信息，对照地图信息的对照读取完成实时显示所在具体区域地点，同时此次设计系统为单片机系统，具有可扩展性，如结合语音助手提示信息可便利盲人的出行。

## 1.2 设计背景

目前手机地图的应用的普及，如百度地图，高德地图等使人们的出行得到便利，人们通过手机便可以获取地图信息，车载导航便是通过连接手机地图应用实现定位导航，然而国内大部分定位系统都是利于手机APP实现定位的，而小型定位硬件系统是少之又少，室内定位基本采用基站手机定位，WiFi定位或蓝牙定位，室外定位也有我国自主研发的北斗卫星定位，北斗卫星定位主要用于一些特殊用途，如抗洪救灾等，很少用于民事，故使用率不高，国内大部分室外定位都为GPS定位，再加上传感器辅助算法等，弥补GPS室外定位精度不高，不稳定等缺陷。由于此设计系统为单片机系统，可扩展性强，也可通过增加传感器，增强定位的精度，不仅实现室外高精度定位，在一些偏僻的地方，如树林，桥洞，隧道等亦能完成定位，使定位更稳定。将GPS定位显示系统结合智能手表实现定位智能手环，为出行提供便利。人们可以及时的获取自己的位置信息，针对一些

特殊群体，如不擅用网络手机的老人，出游旅行团，若在技术允许的条件下，将定位信息通过语音播放，成为个人行走导航助手，帮助盲人出行。实时显示或语音提示位置信息的系统将会是一个可满足多方面人群需求的系统。

## 1.3 方案说明

### 1.3.1 原理说明

GPS是Global Positioning System(全球定位系统)的缩写，该系统由三部分构成: GPS卫星星座(空间部分)、地面监控系统(控制部分)和用户接收机(用户部分)[4-5]。本文主要讨论GPS定位原理，GPS模块工作定位时，通过接受卫星信号并进行运算处理，当卫星数量为3颗时，得到的定位数据不稳定，是2D定位，只有卫星数大于4时才实现3D定位，并且随着卫星数目的增多，定位越精确，GPS定位原理以高速运动的卫星的瞬间位置作为起始位置数据，假设为（x,y,z），假设需要最少四颗卫星定位，则起始位置分别为（x1,y1,z1）、（x2,y2,z2）、（x3,y3,z3）、（x4,y4,z4）。假设GPS模块接收卫星数据耗费△t时间，卫星与模块的距离为D，则由图1-1可得到以下四个方程式:

 式（1-1）

 式（1-2）

 式（1-3）

 式（1-4）

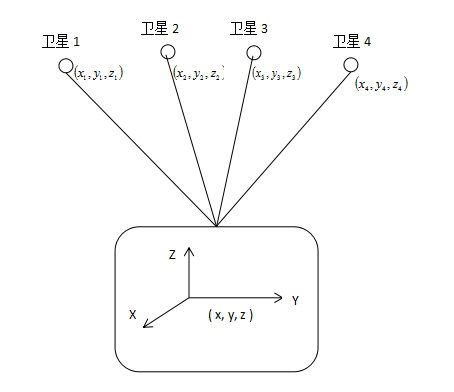


图1-1 卫星定位算式图

通过解四个方程式即可知道GPS模块接收机的地理位置（x,y,z）。以上为得到位置的算法，同理时间也可由路程、速度和时间差得到星历数据，再编写程序将收集到的数据按照UTC标准转换为北京时间。算法公式结合GPS模块对数据的处理和定义才能获得经纬度信息和时间信息。

在GPS系统中，卫星位置的计算采用的是世界大地坐标系，即WGS-84 (World Geodetic System)[18,21,19] 。WGS-84可以看作是协议地球坐标系CTS的一种物理实现，是目前世界上最高精度的全球大地测量参考系统。WGS-84坐标系原点位于地球质量的中心，Z轴平行于国际时间局BIH1984.0时元定义的协议地球极轴(CTP)方向，X轴指向国际时间局BIH 1984.0时元定义的零子午面和国际时间局BIH1984.0时元定义的协议地球赤道的交点，Y轴指向国际时间局BIH 1984.0时元定义的协议地球东向而垂直于X轴方向，构成了地心地固正交坐标系[21]。如下图1-2所示。

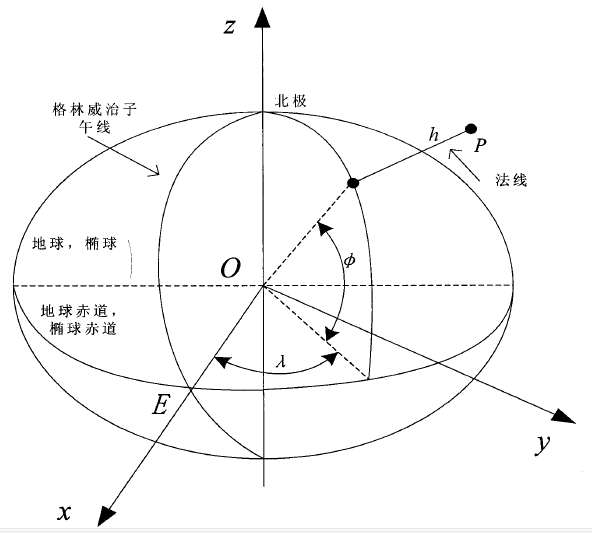


图1-2 WGS-84坐标系

在坐标系中一共平均分配180个纬度，计算大约一个纬度为12756×PI /360 = 111.133 KM，经纬度转换公式为经度 1°长度=111.413cosφ，在纬度φ处，经纬度数据表示方法为度.分.秒，将数据进行编程时，GPS模块采用NEMA -0183协议[1]，利用ASCII码传递定位信息，需要了解一下四种帧表示含义与方法

表1-1 GPS模块帧含义

|  |  |
| --- | --- |
| GGA | 时间、位置、定位类型 |
| GLL | UTC时间、经度、纬度 |
| GSA | 操作模式、卫星、DOP值 |
| GSV | 卫星信息、仰角、方位角、信噪比 |
| RMC | 时间、日期、位置、速度 |

GPS定位信息除了经纬度外，还能显示海拔高度，GPS 输出的高度是垂直于椭球表面的高度而非海平面高度，图中 h 是 GPS 测得的相对于椭球表面的高度；H 表示正高；N 表示大地水准偏差，即地球实际形状与参考椭球体的偏差，范围在正负 100m 之间，它随着地球重力分布变化，没有唯一的确定数值。所以，GPS 直接输出的海拔数据也就会始终存在一个“误差”了。

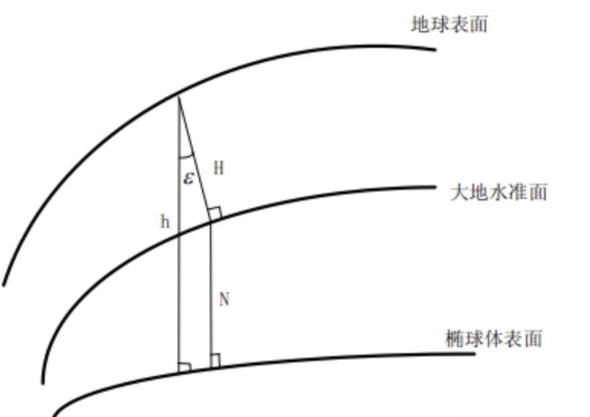


图1-3 海拔高度的计算方法

1.3.2 系统功能说明

此次设计得单片机系统主要有两个功能：定位显示功能和时间显示功能。定位显示功能是由GPS模块，在室外通过分布在地球外六个轨道面的24颗卫星获得定位信息传送给单片机，再通过LCD1602显示出来经纬度。由卫星的信号接收定位信息精度可达10米，若再加上室内定位技术或增加传感器算法来增强精度可达厘米级甚至毫米级，故此系统可用性非常大。除此之外，通过卫星传输信号的时间差与信号传输往返路径，得到的公式算法可求得时间，按照UTC标准转换为北京时间，在LCD显示屏上将同时显示系统位置信息及当地北京时间。

## 1.4 模块设计

1.4.1 主控制模块设计方案

单片机主控芯片主要有AT系列单片机和STC系列单片机，其中AT系列单片机主要是AT89S51和AT89S52，AT89S52是在AT89S51的基础上，增加了一倍Flash程序存储器的大小，即S52程序存储大小为8k。同理STC系列单片机有STC89C51和STC89C52，这两者的程序存储器大小对应跟AT系列两者芯片存储大小相同。除此52还多了一个定时器。

将AT系列单片机和STC单片机芯片对比，对比如下表格，综合考虑存储空间及系统需求等要求，本系统最终选用STC89C51芯片作为系统主控芯片。

表1-2 AT单片机与STC单片机对比

|  |  |  |
| --- | --- | --- |
|  | AT系列单片机 | STC系列单片机 |
| 程序存储空间 | 8k字节 | 8k字节 |
| 数据存储空间 | 256字节 | 512字节 |
| EEPROM存储空间 | 无 | 2k字节 |
| 下载串口 | 特定编程器 | 可在线编程，232串口 |
| 速度 | 较快，6T模式 | 稍慢，12T模式 |
| 工作电压 | 5.5~3.3v | 5.25~4.75v |
| 价格 | 贵 | 便宜 |

### 1.4.2 显示系统选择方案

目前市场所用显示模块主要有LCD1602和12864两种显示屏，12864能显示128\*64个字符或8\*4个汉字，也可以显示图形，功能比1602强大，而1602只能显示字母、数字和符号，能显示16\*2个字符，显示内容简单，但是两种显示屏原理简单且相仿，使用难度不大，编程写指令、地址和数据等都大致相同。以下表格为两者主要区别与其优缺点。结合系统需要显示的经纬度和时间为字符，无需汉字与图形的显示，考虑性价比和性能，最终选择较为合适的LCD1602。

表1-3 1602与12864优缺点对比

|  |  |  |  |
| --- | --- | --- | --- |
| LCD类型 | 分辨率 | 优点 | 缺点 |
| 1602显示屏 | 字5\*8点阵  字符16\*2点阵 | 1. 字符型液晶，能显示字符和数字。2.操作控制简单。3.成本较低大约6块钱 | 1. 显示的字体有大小限制。2.不能显示图形等。3.不能显示曲线 |
| 12864显示屏 | 128\*64点阵 | 1. 功耗低 2. 体积小不占面积 3. 重量轻，超薄 | 1. 液晶显示信息量大，相当来说程序和电路都比较复杂。2.价格较高最少40块钱一块 |

### 1.4.3 GPS模块选择方案

目前市面上主要销售GPS模块为环天、达伽马、Ublox等，主要本次设计使用GPS模块为Ublox NEO-6M 模组，GPS模块上电后会有冷启动、暖启动和热启动三种模式，模块首次通电工作会有冷启动反应，即模块会计算一次星历数据，这需要花费一定时间，大概一分钟左右完成，模块即可完成定位。

在模块选择方面，对比市场上达伽马S-87和Ublox NEO-6M模块，得到如下对比表格。

## 表1-4 达伽马与Ublox模块对比

|  |  |  |
| --- | --- | --- |
|  | 达伽马S-87 | Ublox NEO-6M |
| 灵敏度 | -159dbm | 跟踪：-165dbm 捕获：-148dbm |
| TTFF首次定位时间 | 1~42s | 1~23s |
| 定位精度 | 10m | 2.5m~3m |
| 电源 | 3v | 3-3.6v |

## 1.5 整体方案的选择

### 1.5.1 总体设计框架

单片机的最小系统有电源电路、复位电路和晶振电路就可以工作了。本系统再加上STC89C51单片机芯片作为主控芯片，GPS模块选用Ublox-NEO-6M模块，显示模块选用LCD1602显示屏即可构成整个系统硬件部分。本设备本着低功耗，低成本，高实用性等要求，提出了下图的框架设计。

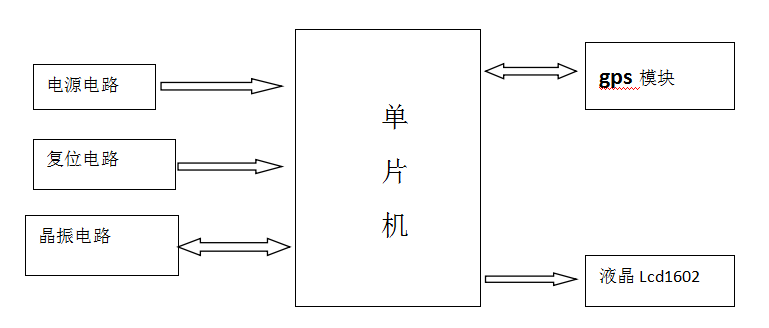


图1-4 总体框架

1.5.2 硬件方案

完成系统仿真及焊接图，使用protel99se软件完成仿真的绘制，根据下图的系统框架，在protel99se软件上完成系统的原理图绘制，在proteus软件上完成仿真，在软件工具上搭建好系统，计算各电阻电容的数值，仿真实现系统功能后，实现系统功能需要完成硬件系统的搭建，购买原器件芯片及模块等材料，焊接单片机系统，完成硬件电路部分的测试。



图1-5 硬件框架图

### 1.5.3 软件设计与编程

本系统采用STC51芯片，是基于51单片机的定位系统，所用编程软件为keil4软件，keil uvision4 软件是一款单片机应用开发软件程序，支持多个公司的不同芯片编程，用户界面友好，支持多种语言，主要语言为C语言，集编辑、编译、调试和仿真于一体的软件。编程环境如下：

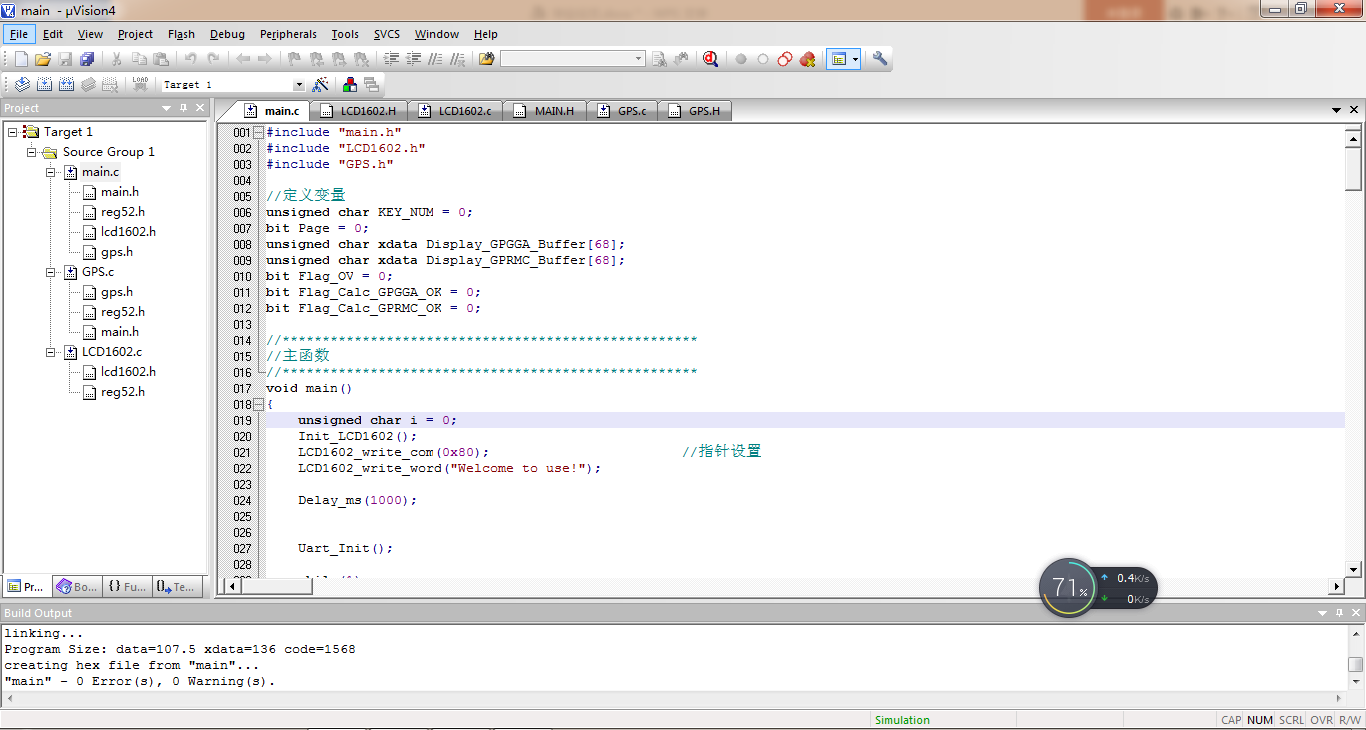


图1-6 keil4编程软件

在Kiel4软件下完成编程和编译调试，生成.HEX文件后保存，在STC-ISP中完成程序的烧录，如下图

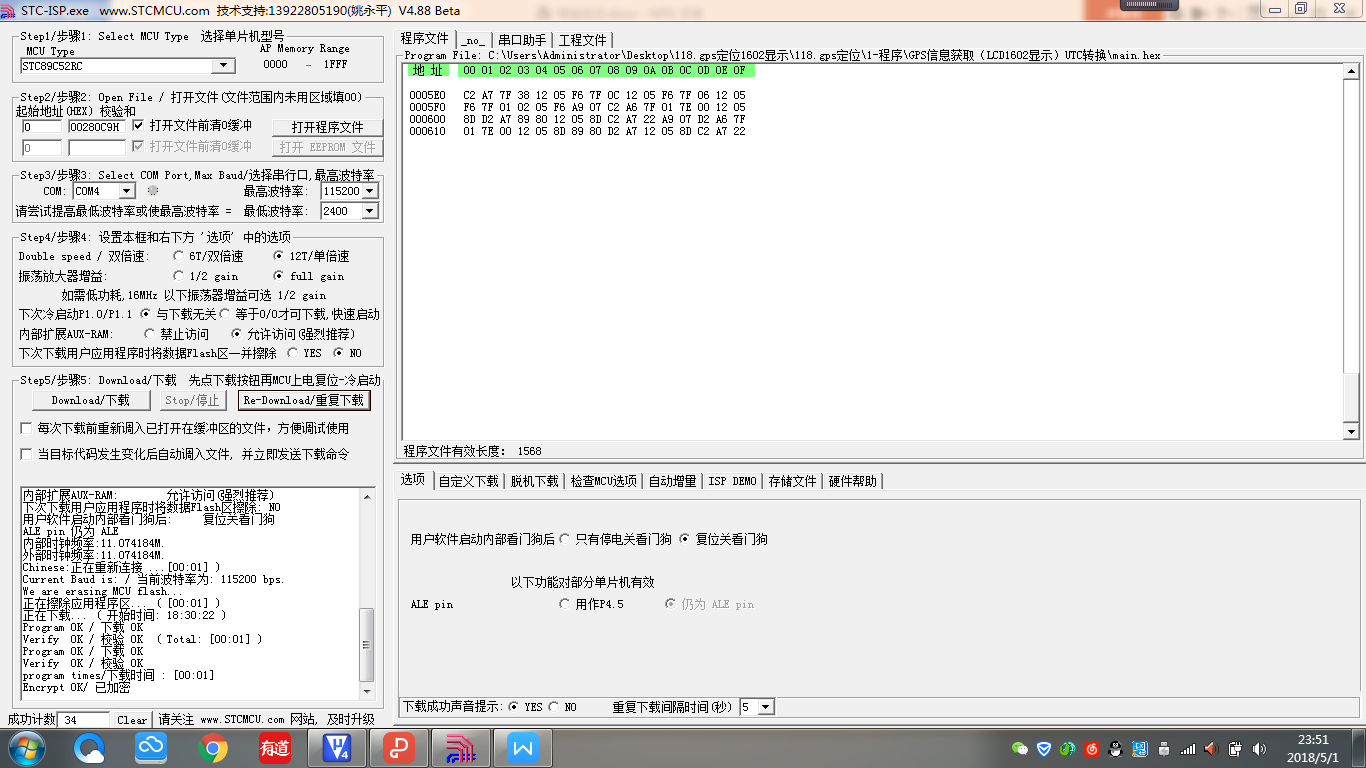


图1-7 STC-ISP烧录软件

# 第2章 系统硬件的设计与实现

## 2.1 主控芯片电路

主控芯片选用STC89C51单片机，51单片机是STC公司生产的一款可编程芯片。使用率高，性能稳定，资料多也便于学习操作，主要功能有512字节RAM，可实现在线编程，通过串口直接下载，具有EEPROM功能，有两个定时器等。下图2-1为STC89C51单片机引脚图。

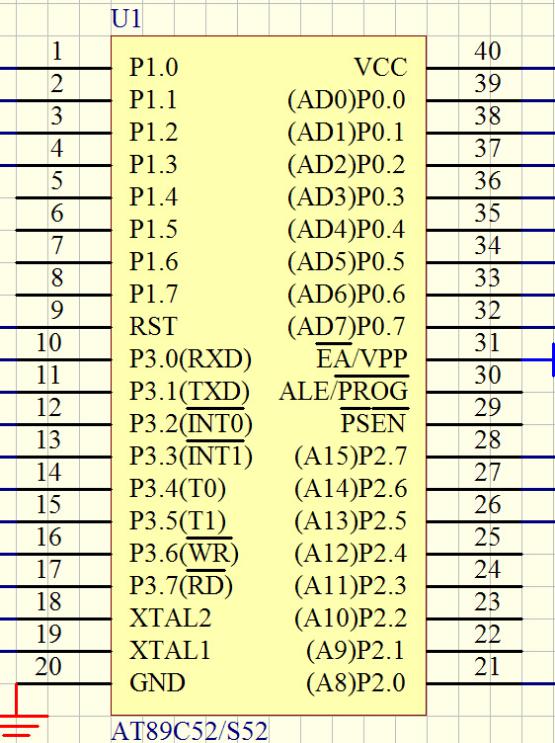


图2-1 STC89C51芯片引脚图

表2-1为芯片引脚说明，19、18引脚可以有外部方式即将XTAL2悬空，XTAL1接外部震荡信号，也可以内部方式，外接晶振电路。

表2-1 STC89C51引脚说明

| 引脚类型 | 引脚号 | 引脚功能说明 |
| --- | --- | --- |
| 电源引脚 | 40 | VCC |
| 20 | GND |
| 时钟引脚 | 18 | 反相放大器输入口 |
| 19 | 反相放大器输出口 |
| 控制引脚 | 9 | 复位口 |
| 31 | 内外ROM的选择端和对片内EEPROM的编程电源 |

续表2-1 STC89C51引脚说明

|  |  |  |
| --- | --- | --- |
|  | 30 | P0口低八位地址的锁存允许和片内EEPROM编程脉冲 |
| 29 | 外部ROM读选通信号 |
| P0口 | 32~39 | 八位双向IO口 |
| P1口 | 1~8 |
| P2口 | 21~28 |
| P3口 | 10~17 |

## 2.2 电源电路

电源电路由一个自锁开关和一个DC电源插口组成，原理图如下，自锁开关作用仅是电源开关作用，由于本次设计使用电源是外接电源，不是电池电源，故还结合电源插口便于供电。

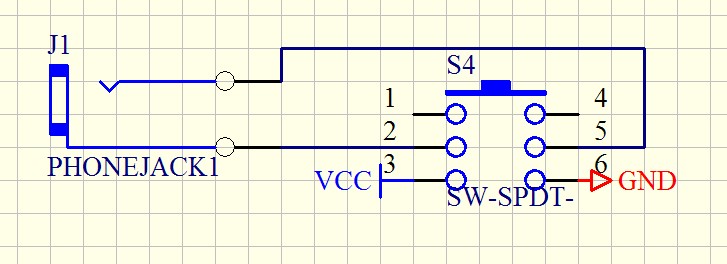


图2-2 开关电路仿真

## 2.3 复位电路

复位电路原理图如下，选用10uf的电容和10k电阻组成复位电路，复位条件及振荡器运用时，有24个震荡周期以上的高电平出现，即可实现复位。则由公式计算得电容充电时间为0.1s，在单片机通电初始，电容C1充电为3.5v，则电阻R4电压由5v降低至1.5v，单片机系统持续为高电平，系统完成复位。

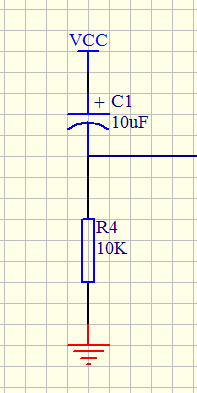


图2-3 复位电路仿真

## 2.4 晶振电路

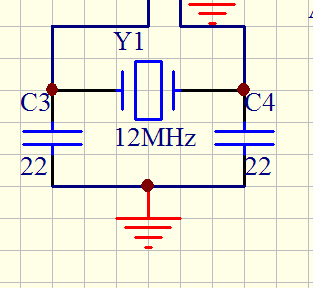
晶振电路给单片机提供信号脉冲，晶振电路原理图如下，由两个22pf电容并联一个12MHZ晶振构成。晶振电路所接接口分别为单片机的19引脚XTAL1和18引脚XTAL2，单片机这两个引脚内部为一个反向放大器，两者连接构成正弦波震荡电路。可以使用外部方式和内部方式，此晶振电路采用内部方式，时钟频率为震荡脉冲二分频，晶振电路采用12MHZ晶振，时钟频率为6MHZ，电容取22pf即可。

图2-4 晶振电路仿真

## 2.5 GPS模块电路

GPS模块选用Ublox NEO-6M GPS 模块，该模块灵敏度高，功耗低，尺寸适中，且该模块能兼容3.3v和5v电压，默认波特率为9600，适合用于一般小型移动设备的定位。图2-5为Ublox NEO-6M GPS 模块图片。



图2-5 为Ublox NEO-6M GPS 模块图

模块引脚功能如下表2-2.

表2-2 GPS模块引脚说明

|  |  |  |
| --- | --- | --- |
| 1 | VCC | 电源 |
| 2 | GND | 接地 |
| 3 | RXD | 接收脚 |
| 4 | TXD | 发送脚 |
| 5 | PPS | 脉冲输出 |

为了实现GPS模块与单片机信息的交换与传输，需要将GPS模块RXD接收口接单片机TXD口发送，GPS模块TXD发送口接单片机RXD接收口，即将GPS模块收发口与单片机收发口反接。在连接线路模块第五引脚可不接单片机，第五引脚内接模块的指示灯，接通电源后，指示灯常亮则模块工作，等待一分钟后，指示灯闪烁则说明模块成功定位。仿真图如下图2-6。

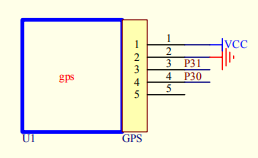


图2-6 GPS模块仿真图

## 2.6 LCD1602显示电路

目前市场最常用显示屏有LCD1602显示屏和12864显示屏，12864显示屏不仅可以显示数字字符等，还可以显示图形和汉字，可显示字符数也比1602多，而1602只能显示32个字符，但是在此系统中暂可不用显示图形及汉字，故考虑各需求及性价比分析，选用LCD1602即可。用于显示经纬度与时间。

本设计系统选用16引脚的LCD1602。引脚图及原理图如下图2-7。各引脚功能如下表2-3.

表2-3 LCD1602引脚说明

|  |  |  |
| --- | --- | --- |
| 引脚 | 符号 | 功能 |
| 1 | VSS | 接地 |
| 2 | VDD | 接电源 |
| 3 | V0 | 对比度调整 |
| 4 | RS | 寄存器选择，1为数据，0为指令 |
| 5 | R/W | 读写信号，1为读操作，0为写操作 |
| 6 | E | 使能端 |
| 7~14 | DB0~DB7 | 低4位与高4位，双向数据线0~7位 |
| 15 | A | 接电源 |
| 16 | K | 接地 |

由引脚表功能可知，绘制原理图时，要使3（V0）引脚VO通过一个2.4k电阻调整对比度，防止显示屏出现对比度过高或过低现象，导致显示重影或模糊。

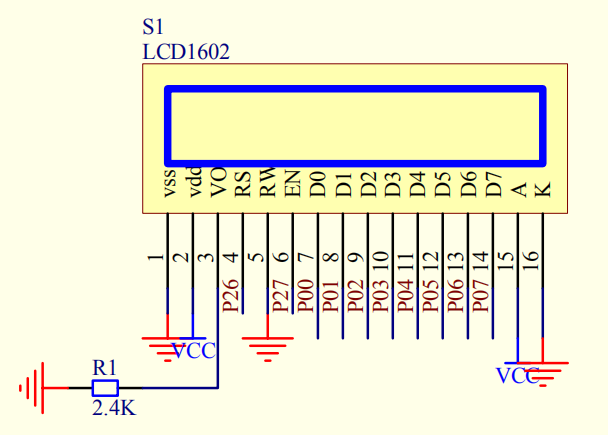


图2-7 LCD1602仿真图

## 2.7 整体原理图及实物图

将单片机主控芯片电路和GPS模块电路、显示电路、晶振电路、开关电路和复位电路连接在一起，绘制整体仿真图为图2-8.通过开关电路，接通电源后，按下自锁开关，GPS模块开始工作，单片机最小系统驱动LCD显示初始化字符，等待两三分钟后，GPS模块接收卫星数据，将数据传入主控芯片，最小系统将数据信息的经纬度信息进行处理和时间信息按照UTC标准转换为北京时间，再将这两个信息传入LCD显示。通过仿真图连接各模块电路制作硬件电路实物图为下图2-9.

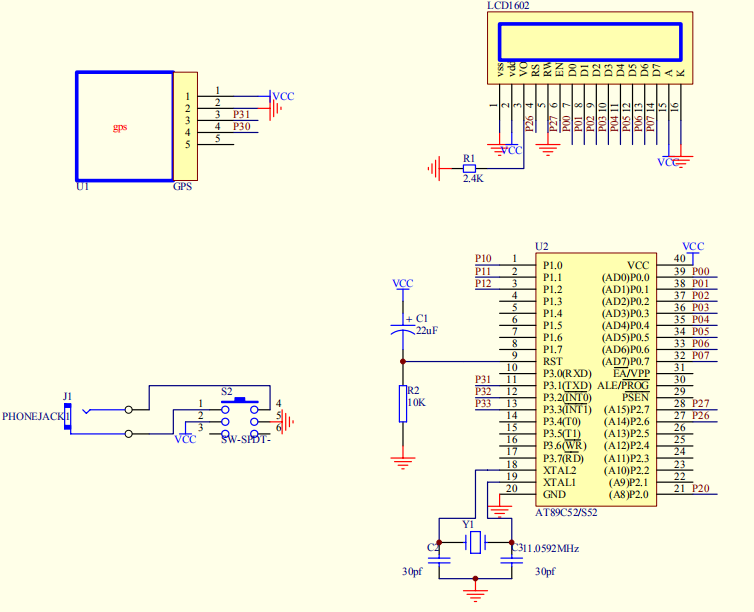


图2-8 整体电路仿真图



图2-9 硬件电路实物图

# 第3章 系统软件的设计与实现

3.1 程序框图

程序开始，对LCD1602进行初始化，对GPS模块初始化，单片机从GPS模块串口接收读取GPS数据，写入缓冲，当读到换行符时进行GPS数据解析，数据解析中BUF|5|=’c’时提取经纬度和时间信息并存入结构体GPS-DATA中，传入LCD进行显示，程序结束。程序流程图如下。

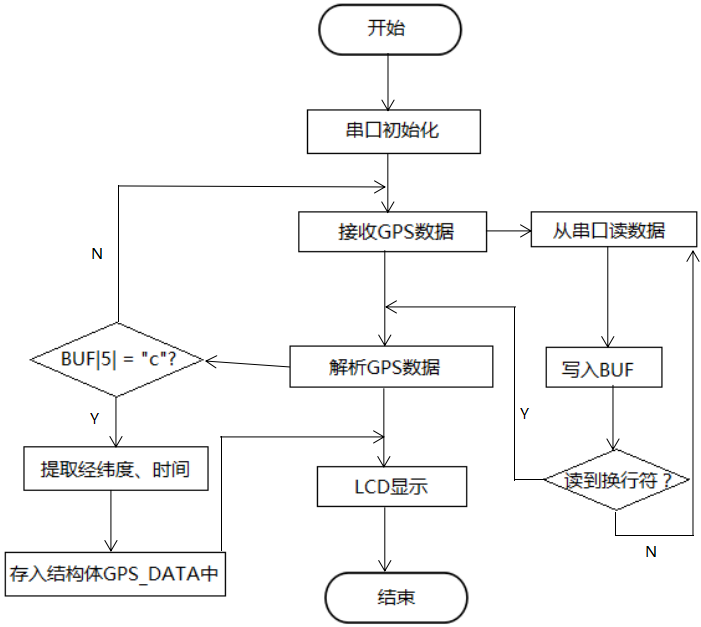


表3-1 程序设计框图

3.2 代码详述

### 3.2.1 头文件

头文件主要包括主函数头文件、显示头文件和GPS模块头文件。

#include "main.h"

#include "LCD1602.h"

#include "GPS.h"

### 3.2.2 IO口设置

sbit LCD1602\_RS = P2^6;

sbit LCD1602\_EN = P2^7;

sbit LED1 = P2^0; //LED IO设置

sbit KEY4 = P3^6; //按键口设置

### 3.2.3 延时函数

延时函数是在12M晶振下测试程序

void Delay\_ms(unsigned int n)

{

unsigned int i,j;

for(i=0;i<n;i++)

for(j=0;j<123;j++);

}

### 3.2.4 1602初始化函数

按键扫描程序如下，

void Init\_LCD1602()

{

LCD1602\_EN = 0;

LCD1602\_write\_com(0x38); //显示模式设定

LCD1602\_write\_com(0x0c); //开关显示、光标有无设置、光标闪烁设置

LCD1602\_write\_com(0x06); //写一个字符后指针加一

LCD1602\_write\_com(0x01); //清屏指令

}

### 3.2.5 GPS解析函数

检测是否收到"GPGGA"这一帧数据代码如下：

if(Flag\_GPS\_OK == 1 && RX\_Buffer[4] == 'G' && RX\_Buffer[6] == ',' && RX\_Buffer[13] == '.')

{

for( i = 0; i < 68 ; i++)

{

Display\_GPGGA\_Buffer[i] = RX\_Buffer[i];

}

将UTC时间转换到北京时间代码如下：

(Display\_GPGGA\_Buffer[7]-0x30)\*10+(Display\_GPGGA\_Buffer[8]-0x30)+8; // UTC+8，0x30为ASCII转换为数字

if( Hour >= 24) //溢出

{

Hour %= 24; //获取当前Hour

Flag\_OV = 1; //日期进位

}

else

{

Flag\_OV = 0;

}

Min\_High = Display\_GPGGA\_Buffer[9];

Min\_Low = Display\_GPGGA\_Buffer[10];

Sec\_High = Display\_GPGGA\_Buffer[11];

Sec\_Low = Display\_GPGGA\_Buffer[12];

Flag\_Calc\_GPGGA\_OK = 1;

}

3.2.6 日期转换函数

获取日期的程序如下：

Day\_High = Day/10 + 0x30; //转换日期值为ASCII

Day\_Low = Day%10 + 0x30;

Month\_High = Month/10 + 0x30; //转换月份值为ASCII

Month\_Low = Month%10 + 0x30;

Year\_High = Year%100/10 + 0x30; //转换年份值为ASCII

Year\_Low = Year%10 + 0x30;

将日期值转换为ASCII代码如下：

void UTCDate2LocalDate(void)

{

Day = (Day\_High - 0x30) \* 10 + (Day\_Low-0x30) + 1; //日期加一

Month = (Month\_High - 0x30) \* 10 + (Month\_Low - 0x30);

Year = 2000 + (Year\_High - 0x30) \* 10 + (Year\_Low - 0x30);

MaxDay = GetMaxDay(Month,Year); //获取当月天数最大值

if(Day > MaxDay) //溢出

{

Day = 1;

Month += 1;

if(Month > 12)

{

Year+=1;

}

}

检测特殊日期，检测是否闰年，及根据闰年判断2月28天还是29天代码如下：

//闰年检测：

bit IsLeapYear(unsigned int uiYear)

{

return (((uiYear%4)==0)&&((uiYear%100)!=0))||((uiYear%400)==0);

}

//2月检查：

case 2:

{

iDays = IsLeapYear(Year\_Value)?29:28;

}

break;

# 第4章 系统整体测试及结果

4.1 测试过程

测试过程步骤如下：

1. 打开keil软件编写程序后点击编译，若编译出错，则根据相关提示错误信息进行更改，直至编译无误
2. 将编译好的程序生成HEX文件
3. 将STC89C51芯片放入开发板，开发板通过USB线连接电脑，查看串口号，打开STCIP烧录软件，通过烧录软件下载HEX文件进单片机主控芯片中根据相关提示信息给MCU上电，等待烧录成功
4. 将烧录号程序的芯片放入单片机系统，给单片机上电，将单片机放置室外，按下自锁开关，观察模块现象，进行测试。问题可能存在于电路的设计上，也可能存于数据处理程序设计缺陷问题上或者数据信号线的排版上。

4.2 测试问题及解决方案

测试过程中遇到的问题：

1. 硬件系统的搭建出现问题，没有在一开始对所有器件先进行检测就焊接，有的器件在焊接过程中受到了损坏，对于电路线路排版设计上也不是很完美，出现线路很多交叉重复导致焊接过程复杂化，焊接电路板过程中也出现很多问题，比如虚焊，焊错等问题，这些问题都在搭建后期进行检测的时候利用万用表检查出来，并进行了修正，最终完成硬件电路的搭建。
2. 软件编程对GPS模块解析程序的不了解，在烧录HEX文件中烧录不进芯片，在反复确认连接线，串口，芯片等完好的情况下完成程序的下载。

## 4.3 测试结果

单片机接上电源放置室外进行测试，打开自锁开关，LCD1602显示“welcome to use!”初始化字符，GPS模块指示灯常亮，说明模块能工作，等待1分钟左右，LCD1602上排显示当地北京时间和海拔高度，下排显示所在地的经纬度，指示灯闪烁，模块完成定位并成功，设计满足要求。测试结果图4-1如下



图4-1 测试结果图

# 第5章 总结与展望

## 5.1 总结

通过这次毕业设计，我不仅对专业知识的掌握更深刻了，使我更深入的了解了单片机，对GPS都有了更深刻的理解。在通过基于单片机的GPS定位这个作品的设计与实现。它具备了基本的定位能力，可以初步满足和个人定位和简单的导航功能需求。时间较短，加上我个人能力还有所欠缺，设计出来的这套定位系统装置难免存在一些不足，还有优化个完善的空间，实现更精确的定位功能，更方便使用，更低的材耗和能耗，拥有更高效，更稳定的性能。例如系统响应性能方面，在打开GPS定位的时候响应就非常慢，可能是在在程序设计上存在一定的问题或者硬件平台处理能力不足，可调用资源有限，在这两方面就有提升空间。在GPS定位上有时接受信号不稳定，定位出现误差大于允许误差范围，这些都是我应该之后要入手调整，完善的方面。能够通过我个人的不断实践最终完成这个毕业设计，对作品的整体设计制作流程也有课更深刻的理解，我对知识的理解和整合应用能提也有了显著的提升。对其中遇到的问题进行分析，我知道如何通过测试，定位解决这些问题，这些都是今后在社会上实实在在用到的技能。总之在今后的工作生活中，我会不断努力前行，好好应用这些技能，提升自身价值，成为社会更需要的人才。

## 5.2 展望

通过设计这个毕业作品，解决期间遇到的难题，我提升了不少，除了要通过自身设计系统的程序和电路板的设计上加进对定位信息的精确度和反应速度外，在此设计系统还可以扩展更多的功能，将此定位系统与智能手表结合在一起，不仅弥补GPS模块只能室外定位的缺陷，还可以结合手机卡的定位以上，完成多重定位结合，使定位方式更多元化，也可以使系统不再是单一的定位系统，成为现代流行需求的智能穿戴产品。

## 5.3毕业要求指标点达成情况分析

本设计达成毕业要求指标点情况如表5.1所示。

表5.1 毕业设计达成毕业要求指标点情况自评表

| 毕业设计教学大纲的能力指标点 | 本毕业设计对该项能力指标点的具体体现 | 自评分数 |
| --- | --- | --- |

续表5.1 毕业设计达成毕业要求指标点情况自评表

|  |  |  |
| --- | --- | --- |
| 1-3 具备针对复杂工程问题进行硬件分析与设计的专业知识和能力。 | 本设计运用电路基础，模拟电子技术，数字电子技术等硬件电路知识，对硬件系统进行整体框架的设计与仿真图的绘制。（1.5，p8） | 3 |
| 1-4 具备针对复杂工程问题进行软件分析与设计的专业知识和能力。 | 本设计能够运用protel 99 SE软件绘制系统的放真图（2.7，P15），也能运用C语言编程软件keil4对程序进行编程，（1.5，P8）能运用STC-ISP烧录软件进行程序的下载。（1.5，P9） | 3 |
| 2-2能够对复杂工程问题所涉及的内容进行文献检索、整理和研究。 | 在研究本设计的应用背景和发展前景时查阅大量文献资料，参考了二十多篇文献资料，从中不仅了解到GPS定位原理方面的的知识（1.3，P2-P5），还对设计在国内外的发展和前景有所了解。（1.1，1.2，P1） | 4 |
| 3-3 能够进行系统结构组成设计和参数计算。 | 本设计在单元电路模块中，复位电路和晶振电路的电阻电容的参数计算，复位电路中充放电时间又电容电阻决定，（2.3，P11）在晶振电路中又晶振频率决定电容取值范围，这些器件的计算与选择都是离不开系统的设计要求。（2.4,P12） | 4 |
| 3-4 能够集成单元过程进行流程设计，对流程设计方案进行优选，体现创新意识。 | 在程序设计中，通过硬件框架及仿真图，完成各模块的程序设计流程，对流程之间的衔接也设置相应的条件，对LCD显示程序和GPS模块接收信号程序在主函数的连接有很大帮助。（3.1，P17） | 3 |
| 7-2 能针对实际的工程项目，分析并判断产品周期中可能对人类和环境造成损害的隐患，并对污染源处置方案和安全防范措施做出评价。 | 本设计电路板的使用工作电压3.3-5.5V，电压范围在人体所能承受的范围之内，且外部电路结构简单，电路板所产生的辐射对人体造成伤害可忽略；电路板芯片，GPS模块和LCD显示屏都可重复利用，其他电容、电阻等外围电路器件可分类回收。（1.4,P5-7） | 3 |
| 10-1 具有英语听说能力；具备英语专业文献的阅读理解能力，具备一定的国际视野，能在跨文化背景下进行沟通与交流。 | Ublox NEO-6M GPS模块是Ublox公司下一款产品，在设计过程中研究过它的相关英文文献资料和说明书等，（2.1，P10）查阅有关STC89C51单片机芯片的英文资料以及GPS定位系统在国内外的发展状况。（1.4，P5） | 3 |
| 11-3 具有安全管理知识，在设计、实验及项目实施中注重安全管理工作。 | 在实现硬件电路焊接过程中，注重实施安全，在拿到原器件之后都会对器件进行短路、断路的检查工作，首先焊接单片机芯片，其次按器件由小到大进行焊接，也能正确区分器件引脚的正负，在焊接过程中正确使用和放置焊笔，结束后也能马上对焊笔进行断电处理。（2.7，P16,4.1,P22） | 4 |

续表5.1 毕业设计达成毕业要求指标点情况自评表

|  |  |  |
| --- | --- | --- |
| 12-1 能认识不断探索和学习的必要性，具有自主学习和终身学习的意识。 | 通过课题的选择，到一步步实现课题设计，完成论文的编写，我查阅了很多资料，看过相关的硕士论文，也通过这些资料了解到自己所需的知识，在设计过程中回顾过去学到的知识并将它们综合起来，整个过程都让我受益匪浅，并且培养了我自主学习的意识，有助于我将来的工作与学习。（5,P23） | 4 |

# 致谢

时光荏苒，日月如梭。四年美好的大学生活时光悄悄的走到了尾声，四年来的我刻苦努力，也在不断的努力成长，离不开我身边的来时同学们的帮助。感谢所有我身边的老师们、同学们、朋友们。是你们的支持和帮助我才有了今天的成长。

经过将近两个月的学习和实践，我最后独立完成了我《基于单片机的GPS定位系统》的毕设作品和论文，从选题到实现整个作品，期间遇到很多困难与挑战，感谢印峰老师孜孜不倦的悉心指导，帮助我战胜这么多的困难和挑战。作品的设计之初，我要将我课堂上学到的零散的知识组织起来，完成这么一个完整的作品，我感到十分的混乱和艰难，是印峰老师帮我缕清了思路，并协助我完成了整个作品的整体框架设计。我才能顺利跨过开发期间遇到的困难和挑战。也是老师无私的奉献精神，指导我完成了论文的编写、修稿、定稿，还有毕设作品的设计，实现和调试，如果没有老师的负责任的、严谨的指导我将无法如此顺利的完成论文的写作。我向我的导师印峰老师表示深深的感谢！

我也向我这四年遇到的所有老师表示感谢，感谢所有任课老师和实验老师对我的栽培，是老师们的谆谆教诲，无私的奉献，才有我今天的成长，你们不仅教会了我知识技能，并在毕设作品中得以体现，顺利完成毕业论文写作。还教会了我为人处世的道理。使我往后离开学校能很好的走向社会，运用这些所学更好的生存，更好的报效祖国。

在此，我同样要感谢我的同学，感谢我的室友，在毕业设计开发期间，你们给了我很多启发，给我提了很多宝贵意见，感谢你们乐于分享经验，感谢你们和我探讨其中的难题，使我面对困难不再觉得孤单。感谢我的同学们对我的帮助支持，在此我深深的表示感谢！

同样感谢所有我借阅的书籍的作者，是你们将你们的智慧凝聚于书上，才带给我那么多新知识和心得，感谢你们的智慧分享让我顺利完成毕业作品。对于你们的帮助与支持，在此我向你们表示由衷的感谢。最后我也要感谢学校给我提供这么好的学习环境，让我在学校里能学到这么的知识技能，学会这么多道理，收获如此珍贵的友谊。感谢！

# 参考文献

[1]何香玲,郑钢.GPS通信的NEMA协议及定位数据的提取[J].计算机应用与软件,2004(12):121-122.

[2]Xin Liu,Zhen Feng,Yao Yuan Zhang,Si Qi Liu. GPS Positioning System Design Based on Micro Control Unit[J]. Advanced Materials Research,2014,3140(915).

[3]林建军. GPS星历解调和UTC时间转换的算法与实现[D].北京交通大学,2009.

[4]李明峰，冯宝红，刘三枝.GPS定位技术及其应用.北京.国防工业出版社.2006.2-6,18

[5]李天文.GPS原理与应用.北京.科学出版社.2003.5-8

[6]Bilich, Andria L. Improving the precision and accuracy of geodetic GPS: Applications to multipath and seismology [Dissertation]. Colorado.University of Colorado at Boulder.2006.2

[7]Jiung-yaoHuang, Chung-Hsien, Tsai.Improve GPS Positioning Accuracy with Context Awareness.IEEE. Ubi-Media Computing, 2008.1-3

[8]美国国防科学委员会报告.崔志译GPS在军事任务和国家基础设施中的作用.卫星应用.2005.1-3

[9]刘春保.GPS系统现状与未来发展.北京.卫星应用简报.2008.1-3

[10]KaiBorre, Dennis M.Akos, Nicolaj Bertelsen著，杨东凯，张飞舟，张波译软件定义的GPS和伽利略接收机.北京.国防工业出版.2009.1-6

[11]王惠南.GPS -T-航原理与应用.北京.科学出版社.2003.1-3,30-35

[12]熊志昂，李红瑞，赖顺香.GPS技术与工程应用.北京.国防工业出版.2005.1-6

[13]Choi, Kyuhong.Improvements in GPS precision: 10 Hz to one day [Dissertation]. Colorado .University of Colorado at Boulder.2007.7

[14]徐绍锉，张华海，杨志强.GPS测量原理及应用.武汉.武汉大学出版社，2008.1-5

[15]杨俊，武奇生.GPS基本原理及其Matlab仿真.西安.西安电子科技大学出版社2006.4-8

[16]Tsoi, Raymond. Precise GPS relative navigation for future geodetic satellite formation flying missions [Dissertation]. Calgary (Canada). University of Calgary.2007.2-4

[17]刘基余.GPS卫星导航定位原理与方法.北京科学出版社.2008.54-57

[18]JAMES, BAO-YEN,TSUI著.陈军，潘高峰译.杨小牛审校.GPS软件接收机基础.北京.电子工业出版社2007.68-70,74-81

[19]Elliott D., Kaplan.Chrustopher J., Hegarty主编.寇艳红译.GPS原理与应用.北京电子工业出版社.2007.103-105

[20]沈学标，吴向阳.GPS定位技术.北京.中国建筑工业出版社.2003.41-45

[21]深圳湾，“GPS 的海拔数据为何不准确？——详解无人机导航中的位置坐标系”，shenzhenware.com，2016.07.04.

[22]周建郑.GPS测量定位技术.北京.化学工业出版社2004.8

[23]Nguyen, Thao Q. Efficient GPS position determination algorithms (Dissertation].Air Force Institute of Technology. 2007.1-2

[24]Gollakota, Manorama. GPS receiver improvements for low signal-to-noise ratio applications [Dissertation]. California.University of Southern California.2007

[25]Kouba clockJ, Heroux P Precise point positioning using IGS orbit andproducts.GPS Solutions.2000,12-15

附录 A 程序

1. GPS接收信号函数程序

#include "GPS.h"

unsigned char RX\_Buffer[68];

unsigned char RX\_Count = 0;

unsigned char Hour = 0,Min\_High = 0,Min\_Low = 0,Sec\_High = 0,Sec\_Low = 0;

unsigned char Month = 0,Day = 0,Month\_High = 0, Month\_Low = 0,Day\_Low = 0 ,Day\_High = 0, Year\_High = 0,Year\_Low = 0;

unsigned int Year = 0;

bit Flag\_GPS\_OK = 0;

unsigned char MaxDay = 0;

void Uart\_Init()

{

SCON = 0X50; //UART方式1；8位UART

REN = 1; //允许串行口接收数据

PCON = 0x00; //SMOD=0;波特率不加倍

TMOD = 0x20; //T1方式2，用于产生波特率

TH1 = 0xFD; //装初值

TL1 = 0xFD;

TR1 = 1; //启动定时器1

EA = 1; //打开全局中断控制

ES = 1; //打开串行口中断

}

void RECEIVE\_DATA(void) interrupt 4 using 3

{

unsigned char temp = 0;

ES=0;

temp = SBUF;

RI = 0;

// LED1 = ~LED1;

if(temp == '$')

{

RX\_Count = 0;

Flag\_GPS\_OK = 0;

}

RX\_Buffer[RX\_Count++] = temp;

if(RX\_Count >= 59)

{ RX\_Count = 59;

Flag\_GPS\_OK = 1;

}

ES=1; }

1. LCD显示数据函数

#include "LCD1602.h"

void LCD1602\_delay\_ms(unsigned int n) //延时函数(12M晶振下测试)

{

unsigned int i,j;

for(i=0;i<n;i++)

for(j=0;j<123;j++); }

void LCD1602\_write\_com(unsigned char com) //写指令

{

LCD1602\_RS = 0;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 1;

LCD1602\_PORT = com;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 0; }

void LCD1602\_write\_data(unsigned char dat) //写数据

{

LCD1602\_RS = 1;

LCD1602\_delay\_ms(1);

LCD1602\_PORT = dat;

LCD1602\_EN = 1;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 0; }

void LCD1602\_write\_word(unsigned char \*s) //连续写字符

{ while(\*s>0)

{ LCD1602\_write\_data(\*s);

s++; }}

void Init\_LCD1602()

{

LCD1602\_EN = 0;

LCD1602\_write\_com(0x38); //显示模式设定

LCD1602\_write\_com(0x0c); //开关显示、光标有无设置、光标闪烁设置

LCD1602\_write\_com(0x06); //写一个字符后指针加一

LCD1602\_write\_com(0x01); //清屏指令 }

1. 主函数

#include "main.h"

#include "LCD1602.h"

#include "GPS.h"

unsigned char KEY\_NUM = 0; //定义变量

bit Page = 0;

unsigned char xdata Display\_GPGGA\_Buffer[68];

unsigned char xdata Display\_GPRMC\_Buffer[68];

bit Flag\_OV = 0;

bit Flag\_Calc\_GPGGA\_OK = 0;

bit Flag\_Calc\_GPRMC\_OK = 0;

void main() //主函数

{

unsigned char i = 0;

Init\_LCD1602();

LCD1602\_write\_com(0x80); //指针设置

LCD1602\_write\_word("Hello 2014551003");

Delay\_ms(1000);

Uart\_Init();

while(1)

{

Scan\_Key();

if(Flag\_GPS\_OK == 1 && RX\_Buffer[4] == 'G' && RX\_Buffer[6] == ',' && RX\_Buffer[13] == '.') //确定是否收到"GPGGA"这一帧数据

{

for( i = 0; i < 68 ; i++)

{ Display\_GPGGA\_Buffer[i] = RX\_Buffer[i]; }

Hour = (Display\_GPGGA\_Buffer[7]-0x30)\*10+(Display\_GPGGA\_Buffer[8]-0x30)+8; //UTC时间转换到北京时间UTC+8，0x30为ASCII转换为数字

if( Hour >= 24) //溢出

{ Hour %= 24; //获取当前Hour

Flag\_OV = 1; //日期进位 }

else

{ Flag\_OV = 0; }

Min\_High = Display\_GPGGA\_Buffer[9];

Min\_Low = Display\_GPGGA\_Buffer[10];

Sec\_High = Display\_GPGGA\_Buffer[11];

Sec\_Low = Display\_GPGGA\_Buffer[12];

Flag\_Calc\_GPGGA\_OK = 1; }

if(Page == 0 && Flag\_Calc\_GPGGA\_OK == 1)

{

LED1 = ~LED1;

Flag\_Calc\_GPGGA\_OK = 0;

LCD1602\_write\_com(0x80); //设置指针

LCD1602\_write\_data(Hour/10+0x30);

LCD1602\_write\_data(Hour%10+0x30);

LCD1602\_write\_data(':');

LCD1602\_write\_data(Min\_High);

LCD1602\_write\_data(Min\_Low);

LCD1602\_write\_data(':');

LCD1602\_write\_data(Sec\_High);

LCD1602\_write\_data(Sec\_Low);

LCD1602\_write\_word(" ");

LCD1602\_write\_data(Display\_GPGGA\_Buffer[54]);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[55]);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[56]);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[57]);

LCD1602\_write\_word("m");

LCD1602\_write\_com(0x80+0x40); //设置指针

LCD1602\_write\_data(Display\_GPGGA\_Buffer[28]); //N或者S

LCD1602\_write\_data(Display\_GPGGA\_Buffer[17]); //纬度

LCD1602\_write\_data(Display\_GPGGA\_Buffer[18]); //纬度

LCD1602\_write\_data(0xdf); //度

LCD1602\_write\_data(Display\_GPGGA\_Buffer[19]); //纬度

LCD1602\_write\_data(Display\_GPGGA\_Buffer[20]); //纬度

LCD1602\_write\_word("'"); //秒

LCD1602\_write\_data(Display\_GPGGA\_Buffer[42]); //E或者W

LCD1602\_write\_data(Display\_GPGGA\_Buffer[30]); //经度

LCD1602\_write\_data(Display\_GPGGA\_Buffer[31]);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[32]);

LCD1602\_write\_data(0xdf);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[33]);

LCD1602\_write\_data(Display\_GPGGA\_Buffer[34]);

LCD1602\_write\_word("'"); }

if(Flag\_GPS\_OK == 1 && RX\_Buffer[4] == 'M' && RX\_Buffer[52] == ',' && RX\_Buffer[59] == ',') //确定是否收到"GPRMC"这一帧数据

{

for( i = 0; i < 68 ; i++)

{ Display\_GPRMC\_Buffer[i] = RX\_Buffer[i]; }

Year\_High = Display\_GPRMC\_Buffer[57];

Year\_Low = Display\_GPRMC\_Buffer[58];

Month\_High = Display\_GPRMC\_Buffer[55];

Month\_Low = Display\_GPRMC\_Buffer[56];

Day\_High = Display\_GPRMC\_Buffer[53];

Day\_Low = Display\_GPRMC\_Buffer[54];

if(Flag\_OV == 1) //有进位

{

UTCDate2LocalDate(); //UTC日期转换为北京时间 }

Flag\_Calc\_GPRMC\_OK = 1;}

if(Page == 1 && Flag\_Calc\_GPRMC\_OK == 1)

{

LED1 = ~LED1;

Flag\_Calc\_GPRMC\_OK = 0;

LCD1602\_write\_com(0x80); //设置指针

LCD1602\_write\_word("20");

LCD1602\_write\_data(Year\_High);

LCD1602\_write\_data(Year\_Low);

LCD1602\_write\_data('-');

LCD1602\_write\_data(Month\_High);

LCD1602\_write\_data(Month\_Low);

LCD1602\_write\_data('-');

LCD1602\_write\_data(Day\_High);

LCD1602\_write\_data(Day\_Low);

LCD1602\_write\_com(0x80+0x40); //设置指针

LCD1602\_write\_word("Speed:"); //显示内容

LCD1602\_write\_data(Display\_GPRMC\_Buffer[46]);

LCD1602\_write\_data(Display\_GPRMC\_Buffer[47]);

LCD1602\_write\_data(Display\_GPRMC\_Buffer[48]);

LCD1602\_write\_data(Display\_GPRMC\_Buffer[49]);

LCD1602\_write\_data(Display\_GPRMC\_Buffer[50]);

LCD1602\_write\_word("m/s");

}

}

}

void UTCDate2LocalDate(void) //UTC日期与当地日期转换

{

Day = (Day\_High - 0x30) \* 10 + (Day\_Low-0x30) + 1; //日期加一

Month = (Month\_High - 0x30) \* 10 + (Month\_Low - 0x30);

Year = 2000 + (Year\_High - 0x30) \* 10 + (Year\_Low - 0x30);

MaxDay = GetMaxDay(Month,Year); //获取当月天数最大值

if(Day > MaxDay) //溢出

{ Day = 1;

Month += 1;

if(Month > 12)

{ Year+=1; } }

Day\_High = Day/10 + 0x30; //转换日期值为ASCII

Day\_Low = Day%10 + 0x30;

Month\_High = Month/10 + 0x30; //转换月份值为ASCII

Month\_Low = Month%10 + 0x30;

Year\_High = Year%100/10 + 0x30; //转换年份值为ASCII

Year\_Low = Year%10 + 0x30; }

//获取当月日期最大值

unsigned char GetMaxDay(unsigned char Month\_Value,unsigned int Year\_Value)

{ unsigned char iDays;

switch(Month\_Value) {

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

{ iDays = 31; }

break;

case 2:

{ iDays = IsLeapYear(Year\_Value)?29:28; //判断2月是28天还29天 }

break;

case 4:

case 6:

case 9:

case 11:

{ iDays = 30; }

break;

default : break; }

return(iDays); }

bit IsLeapYear(unsigned int uiYear) //闰年检测

{ return (((uiYear%4)==0)&&((uiYear%100)!=0))||((uiYear%400)==0); }

void Scan\_Key() //按键扫描程序

{

if( KEY4 == 0 ) //按键1扫描

{

Delay\_ms(10); //延时去抖

if( KEY4 == 0 )

{

while(KEY4 == 0); //等待松手

KEY\_NUM = 3;

Page = ~Page;

LCD1602\_write\_com(0X01); //清屏

}

}

}

void Delay\_ms(unsigned int n) //MS延时函数(12M晶振下测试)

{

unsigned int i,j;

for(i=0;i<n;i++)

for(j=0;j<123;j++);

}