



许昌学院
XUCHANG UNIVERSITY

本科生毕业设计

基于 NB-IoT 的分布式通讯线路监测护线桩

学 院	信息工程学院
专 业	物联网工程专业
班 级	2016 级本科 1 班
学 号	5006160067
学 生 姓 名	贾泽
联 系 方 式	13069509009
指 导 教 师	于洋 职称： 讲师

2020 年 5 月

独 创 性 声 明

本人郑重声明：所呈交的毕业论文（设计）是本人在指导老师指导下取得的研究成果。除了文中特别加以注释和致谢的地方外，论文（设计）中不包含其他人已经发表的研究成果。与本研究成果相关的所有人所做出的任何贡献均已在论文（设计）中作了明确的说明并表示了谢意。

签名： _____
_____年____月____日

授权声明

本人完全了解许昌学院有关保留、使用本科生毕业论文（设计）的规定，即：有权保留并向国家有关部门或机构送交毕业论文（设计）的复印件和磁盘，允许毕业论文（设计）被查阅和借阅。本人授权许昌学院可以将毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编论文（设计）。

本人论文（设计）中有原创性数据需要保密的部分为（如没有，请填写“无”）：

学生签名： _____
_____年____月____日
指导教师签名： _____
_____年____月____日

基于 NB-IoT 的分布式通讯线路监测护线桩

摘 要

鉴于智慧城市的需求，智慧城市的发展已经成为大多数城市的发展目标。窄带物联网是一种低能耗的宽带网络，具有广泛的地理覆盖、安全性高和低成本，这些特点可以帮助城市更加物联化和智能化。基于此，本文通过研究NB-IoT网络的基本结构和优势，以护线桩为例，研究NB-IoT网络支持下的应用布置，这不仅可以降低了巡线运行成本，而且还实现了护线桩的智能管理和大数据管理。

本设计以NB-IoT为网络基础，采用STM32BC20核心板等模组进行数据采集处理，实现了温湿度数据、定位信息、物体偏移角度的功能设计。

关键词： NB-IoT；窄带物联网；护线桩

Distributed Communication Line Monitoring Protection Pile Based on NB-IoT

ABSTRACT

In view of the demand of smart city, the development of smart city has become the development goal of most cities. Narrowband the Internet of things is a low-energy broadband network with extensive geographical coverage, high security and low cost, which can help cities become more connected and intelligent. Therefore, it is necessary to promote the development of the Internet of everything. Based on this, this paper studies the basic structure and advantages of NB-IoT network. Taking the wire retaining pile as an example, this paper studies the application layout under the support of NB-IoT network. Its practical application value is very high.

Based on NB-IoT network, the design adopts modules such as STM32BC20 core plate for data acquisition and processing, and realizes the functional design of temperature and humidity data, positioning information, and object migration Angle, and achieves the successful test of line retaining pile.

Key words: NB-IoT; Narrowband the Internet of things; Convenient pile

目 录

1. 绪论.....	1
1.1 课题研究的背景与意义	1
1.2 国内外研究现状	2
1.3 研究目的和主要内容	2
1.4 章节安排	3
2. 方案论证与设计.....	4
2.1 基于 NB-IoT 的分布式通讯线路监测护线桩需求分析	4
2.2 基于 NB-IoT 的分布式通讯线路监测护线桩概要设计	5
3. 系统硬件设计.....	9
3.1 微控制器电路的设计	9
3.2 护线桩姿势检测模块电路设计	9
3.3 温湿度检测模块电路设计	10
4. 系统软件设计.....	12
4.1 软件架构	12
4.2 开发环境与开发工具	12
4.3 手机终端软件设计	13
5. 系统测试.....	16
5.1 运行状态测试	16
5.2 定位状态测试	16
5.3 温湿度状态测试	16
5.4 护线桩姿态测试	17
6. 总结与展望.....	18
6.1 总结	18
6.2 展望	18

参考文献.....	19
附 录.....	20
致 谢.....	32

1. 绪论

1.1 课题研究的背景与意义

随着城市化建设的发展，城市内公共建设得到了迅速发展。大多数电力和通信部门都有自己管理的护线桩。随着城市面积扩大，护线桩的使用越来越广泛、数量也越来越多，使得护线桩的监管难度大，偷盗护线桩的犯罪行为越来越严重。护线桩是城市建设的重要环节，一旦损坏或被盗，如果不及时处理，可能会造成安全问题。

据统计，每年护线桩被破坏现象十分严重。这些护线桩的破坏现象，影响了正常设施功能的发挥，并造成巨大的经济损失和安全问题。目前，大量的护线桩没有有效的实时监察手段，人工巡线既昂贵又低效。物联网简称 IoT，是互联网、传统电信网等信息的承载体，让所有能行使独立功能的普通物体实现互联互通的网络。物联网借助四大网络，实现万物互联。四大网络包括：短距离有线通讯、长距离有线通讯、短距离无线通讯、长距离无线通讯。一般来说，物联网具备十大基本功能，其最基本功能特征是提供“无处不在的连接和在线服务”，相对于普通的设备来说，物联网化的设备具备强大的管理优势。基于蜂窝的窄带物联网 NB-IoT 是一种低能耗、高连接、高覆盖、低成本的物联网技术，使其可以广泛应用于多种垂直探测行业，如设备检测、远程抄表、资产追踪、智能泊车、农业智慧化等^[1]。

逐渐有一些厂商已经跟护线桩厂商进行合作研发软硬件结合度高的产品，但现有方案大都基于 ZigBee、LoRa 或传统 2G、3G、4G 蜂窝网。前两者需要通过自建基站或网关的方式将传感器接入到互联网中，安放范围相对复杂，并且传统蜂窝网络传输又面临不适合低功耗场景的问题。现在，为了实现实时并且高效的监管，进行统一管理是十分必要的。传统的护线桩采用的是普通物理护线桩，并没有智能化功能，缺乏有效的统一管理。部分带智能化控制的护线桩系统，智能化程度较低。

对于国内运营商来说，对通信护线桩设施进行集中智能化管理已变得十分有必要。创建一个物联网监控平台通过利用 NB-IoT 技术监测，当护线桩被挪动或者丢失时，通过传感器检测到角度的变化并向上报告平台，让维修人员及时去现场进行处理^[2]。

NB-IoT 自身具备的低功耗、广覆盖、低成本、大容量等优势，由于 NB-IoT 技术自身的低成本特性，在户外设备新建或智能化升级改造方面也不会花费太多成本，因此，实现对护线桩的物联化管理具备突出的现实意义。

1.2 国内外研究现状

我国护线桩的设置起步较晚，从上世纪 80 年代我国才开始研制护线桩产品。进入 90 年代后，国内主要厂家大多是学习国外的基础设计，并且市场也刚刚开始进行初步阶段。随着政府的开放，外国企业开始大量进入中国城市基础设施市场，带来先进技术，促进了市场的成熟。这期间，我国设计护线桩的企业发展十分迅速，一些企业进行了技术合作和融资扩张，规模逐渐扩大，取得了良好的成果。目前已知的主要低功耗宽带网络包括物联网、LoRa 和 Sigfox 等。物联网在授权频段内运行，与 LoRa 和 Sigfox 相比具有实时、安全和快速交互的优势。该技术将护线桩下的传感器信号与传感器连接起来，解决了传感器信号等问题。因此，保护桩的管理具有重要的实际意义^[3]。

自从从 20 世纪 60 年代以来，由于安全意识和环境安全要求的提高，国外对护线桩存在的研究迅速发展。在国外，比如日本，大多使用 RFID 技术进行监测。使用以上方法均存在覆盖率低、功耗高、产品体较大、成本高、射频识别极易被障碍物阻断等问题。由于人们对能源、成本以及技术的要求越来越高，此种方法慢慢无法满足此类工程应用。近些年来，窄带物联网技术逐渐兴起，在城市建设、智能家居等诸多方面解决了覆盖面积小、信号弱、成本高、功耗高等问题，因此 NB-IoT 技术逐渐出现在人们的视野中，并应用于很多领域。随着传感器的生产设计制作水平蒸蒸日上，变得越来越精密的传感器和护线桩功能多样性逐渐增加，因而增加了护线桩的种类，使得它们更有利于生活应用。

1.3 研究目的和主要内容

为了更好地解决部分护线桩管理的问题，根据近年来国内三大运营商运维部门提供的信息显示，在护线桩的日常管理维护中，存在不少管理困难或隐患。一是护线桩的数量大、分管的部门比较多、监管起来有一定难度，一旦护线桩出现问题或隐患，部门之间就会发生相互推责的现象，给管理带来麻烦；二是护线桩发生损坏、缺失、堵塞、移位等现象时，维护部门不能及时获取相关信息，所以就不能及时处理问题和消除隐患，容易造成事故发生；三是人工巡线效率较低，需要投入大量的维护人员，增加维护部门的管理成本；四是特别是近年来，随着网络的迅速发展，通信管线资源变得越来越重要，运营商之间利用非属于自己的通信管道进行私穿线路、放置通信设施的较多，给合法利用该通信管道的用户带来困扰甚至发生纠纷^[4]。因此，需要对通信护线桩目前的管理现状进行升级。本文主要研究将分布式护线桩进行有效管理。

本文的主要流程是首先对护线桩进行需求分析。根据分析后的结果，对系统进行框

架设计；对支持 NB-IoT 的智能护线桩进行硬件设计。总方案实现了电源电路、ST-LINK 仿真器、NB-ANT、GPS 北斗模块、三轴加速度传感器、温湿度传感器。接下来在硬件兼容的前提下，设计支持 NB-IoT 技术的护线桩的软件，实现需求功能。再则在物联网开放平台上创建护线桩管理应用程序。在 NB-IoT 支持下接收和统一管理的智能化保护请求。最后对于该系统和硬件设施进行测试，验证是否满足需求。

1.4 章节安排

本文的重点是在公共基础设施建设中智能护线桩管理问题。基于物联网的开放平台，开发了基于 NB-IoT 的智能护线桩管理系统，解决了上述问题，实现了上述目标。

第一章绪论。主题研究的背景与意义。

第二章方案论证与设计。首先进行了需求分析，以了解需求功能。然后进行总体的设计，提出了一种利用 NB-IoT 系统的智能护线桩。最后，进行接口设计，指定系统和模块是如何连接和工作的。

第三章支持 NB-IoT 的分布式通讯线路监测护线桩硬件设计。主要内容为按照结构设计方案中实现了电源设计、ST-LINK 仿真器、NB-ANT、GPS 北斗模块、三轴加速度传感器、温湿度传感器。

第四章支持 NB-IoT 的分布式通讯线路监测护线桩软件设计。主要内容为软件结构的设计、搭建开发环境，以硬件电路为基础，通过软件实现智能护线桩系统所需功能。

第五章系统测试。主要目的是测试 NB-IoT 分布式通信线路监测护线桩的各个方面，用来验证系统功能。

第六章总结与展望。总结研究结果摘要。分析了本文设计的基于 NB-IoT 的分布式通讯线路监测护线桩的优缺点，并提出了下一步改进的方向。

2. 方案论证与设计

本章将进行分析需求, 并将开展一项研究的总体设计和分布式监控系统通信线路的设计基于 NB-IoT 领域的使用方法、系统工程和基本框架, 以构建一个分布式监控系统通信线路基于 NB-IoT 技术。

2.1 基于 NB-IoT 的分布式通讯线路监测护线桩需求分析

本节将对基于 NB-IoT 的分布式通讯线路监测护线桩系统进行需求分析, 以确定功能需求、性能需求以及运行需求, 为概要设计打下基础。

2.1.1 功能需求

本研究采用智能护线桩管理系统。该系统与 NB-IoT 通信技术相结合, 旨在支持基于 NB-IoT 的分布式通讯线路监测护线桩来管理分布式护线桩的集中管理。本研究需要设计支持 NB-IoT 的智能护线桩管理系统。

支持 NB-IoT 的智能护线桩需要具备测量温湿度信息, 实现护线桩的环境监测; 与护线桩管理应用程序进行信息交互, 提供护线桩状态信息, 响应使用说明, 使保护线桩成为护线桩管理的主要管理平台的功能。

支持 NB-IoT 的智能护线桩需要具备定位能力, 实现护线桩的固定位置控制; 将数据传输到服务器上, 提供护线桩位置状态信息, 使保护护线桩成为护线桩管理的主要管理平台的功能。

支持 NB-IoT 的智能护线桩需要具备防侧翻能力, 实现护线桩的限定角度控制, 防止护线桩进行超过规定限度的角度偏移, 使保护护线桩成为护线桩管理的主要管理平台的功能。

2.1.2 运行需求

本文实现的护线桩管理系统的定位是一个护线桩管理平台, 需要服务器平台和相对应的终端软件进行配合, 并且组成一个对用户提供服务的系统。本系统的硬件电路系统框架如下图 2-1 所示。

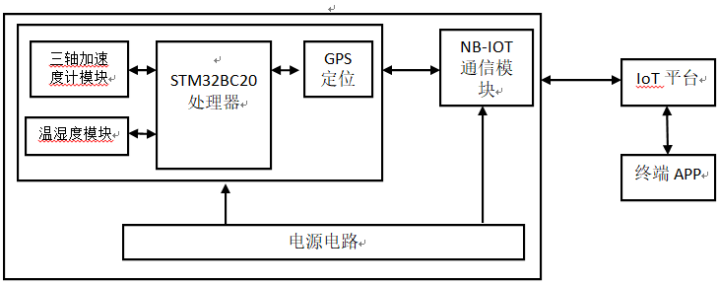


图 2-1 硬件电路系统框架图

该系统由支持 NB-IoT 的智能护线桩和用于护线桩管理的应用程序组成。每一个护线桩的数据与状态归入统一管理，实行保护，并且接收护线桩管理应用的命令。护线桩管理应用程序基于开放的运营商物联网平台。

2.2 基于 NB-IoT 的分布式通讯线路监测护线桩概要设计

本节主要介绍基于 NB-IoT 的智能护线桩管理系统的设计概要，包括单元的结构和分布情况。

2.2.1 总体结构与单元划分

根据本研究的目的，护线桩管理系统基于运营商的开放物联网平台，提供功能支持。本研究基于平台提供的功能来管理护线桩应用的发展，故总体结构支持 NB-IoT 的智能护线桩系统被划分为护线桩主体、湿温度检测模块、通信模块、微控制器模块和姿态调整模块。下面简要介绍一下各部分情况，护线桩主体是支持 NB-IoT 的智能护线桩系统功能实现的载体；温湿度检测模块是测量温湿度信息，实现护线桩的环境控制；通信模块需要实现智能护线桩的通信和接收护线桩信息的传播。微控制器模块用于逻辑控制和信息处理；姿态调整模块用于护线桩角度变化测量，需要实现“倾斜报警”这一功能需求。

2.2.2 相关结构与模块的选型

(1) 护线桩主体

现在市场里护线桩种类齐全，不需要设计新的机械结构。大体的分类有地面安装和地下安装。大多数样式的护线桩如图 2-2 所示。



图 2-2 普通护线桩

由于多数普通的护线桩只具备标示功能，效果不佳。因此本研究将基于 NB-IoT 技术进行改造护线桩，使其功能多样化。

（2）温湿度检测模块

将温湿度传感器固定在护线桩之中，地表温度湿度信息被收集和分享，从而使城市管理部门了解信息并采取降温湿度措施，通过单片机采集生成温湿度变化曲线。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。由于温湿度传感器 DHT11 的芯片紧凑而精确，芯片可以通过单个总线进行通信，数据能精确到小数点后两位，操作简单。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^[5]。因此，DHT11 具有灵敏度高、性价比高、稳定性强等优点。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。例如，DHT11 传感器都在非常精确的湿度控制室中进行测试。单线串行接口允许系统的简单和快速集成。它们很小，耗能很少，可以传输 20 米以下的信号。DHT11 如图 2-3 所示。

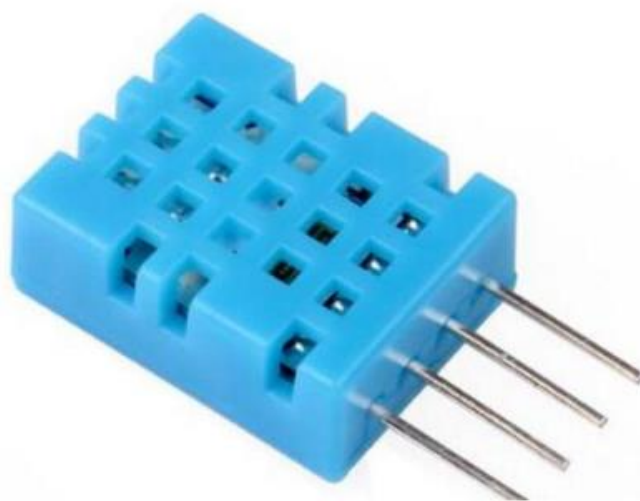


图 2-3 DHT11 温湿度传感器

（3）通信模块

通信模块，也就是 NB-IoT 通信模组。NB-IoT 部分将中央控制器处理后的信息转发到网络中。建立在蜂窝网络基础上，只需要耗费 180kHz 带宽，在 GSM 网、LTE 网基础上进行部署，减少部署成本，完成升级。NB-IoT 智慧系统需要从感知层，NB 网络接入到数据汇聚平台，最后服务上层的行业垂直应用。这种部署可以满足人们高效数据传输的需要，让人们需求得到满足。NB-IoT 信号覆盖范围广，能够使数据顺利传输到互联网；网速可达到每秒 20K 左右，足够传输数据；使用串口通信协议，操作简单；与短信相比，

利用 NB-IoT 传输数据费用相对较低，性价比高。

（4）定位模块

定位模块也是 GPS+北斗定位模块，GPS 模块是集成了 RF 射频芯片、基带芯片和核心 CPU，并加上相关外围电路而组成的一个集成电路。

目前 GPS 模块的 GPS 芯片大部分采用全球市占率第一的 SiRFIII 系列为主。由于 GPS 模块采用的芯片组不一样，性能和价格也有区别，采用 SIRF 三代芯片组的 GPS 模块性能最优，价格也要比采用 MTK 或者 MSTAR 等 GPS 芯片组的贵很多。现阶段也持续在芯片升级，比方 sirf4，然后又是 sirf5，总体灵敏度提高了不少，缩短了定位时间，同时也帮助了客户快速的进入了定位应用状态。GPS 可以提供车辆定位、防盗、反劫、行驶路线监控及呼叫指挥等功能。要实现以上所有功能必须具备 GPS 终端、传输网络和监控平台三个要素。GPS 导航系统的基本原理是测量出已知位置的卫星到用户接收机之间的距离，然后综合多颗卫星的数据就可知道接收机的具体位置。GPS 系统的特点主要是高精度、全天候、高效率、多功能、操作简便、应用广泛等。

而本研究采用 GPS+北斗双模定位可以充分利用两套卫星定位系统的长处和卫星资源，精度和可靠性更强，也可以互相验证。

（5）微控制器模块

微控制器模块负责护线桩的逻辑控制和信号处理。目前主流的微控制有 MSP430 系列、STM32、MSP430 系列以及恩智浦的 Kinetis 系列等。考虑到可以提供最全面、最易用和中小型开发商来说便宜，以及由于护线桩里保护电池上运行能耗、敏感，并且不需要芯片性能高。所以选择 STM32L 系列产品^[6]。单片机软件设计编程采用 C 语言编写，在 MDK 上进行编译，生成 hex 文件。系统启动后，单片机先进行各个功能模块的初始化，然后进入低功耗模式运行。当有突发情况后，唤醒系统，开始进入中断事情处理，单片机将报警信号传送给服务器。如果没有中断，单片机将按照定时器时间进行循环监测，将上报数据给服务器。

（6）护线桩姿态检测模块

陀螺仪模块应用于检测护线桩的角度变化，以确定护线桩是否松动还是无故拆卸的，以实现防盗和检测松动功能。MPU6050 具有非常精确的角加速度和加速度检测功能，可以检测微小的角变化，采用 I2C 通信协议，操作简单。^[7]MPU6050 可以水平安装在护线桩上，当护线桩被拆卸走时，模块检测出角度变化并返回信息。在固定护线桩的情况下，

当有人通过护线桩时，会产生轻微的角度变化，通过分析 MPU6050 提供的信息，信号闪烁则判断为晃动^[8]。MPU6050 如图 2-4 所示。



图 2-4 MPU6050

3. 系统硬件设计

3.1 微控制器电路的设计

微控制器负责控制各个模块并且对相关的信息进行处理。根据总体设计相关内容，微控制器的型号确定为 STM32L。微控制器及其外围电路如图 3-1 所示。其中 C7-C13 为 STM32L 的电源引脚的滤波电容。J4 为 SWD 调试的接口，用于连接 JLINK。J5 为 UART1 接口，用于打印日志信息，方便调试。Y1 为 32.768kHz 的晶振，与 C4、C5 构成晶振电路，为微控制器提供低速时钟信号。微控制器高速时钟信号使用的是内部的 RC 振荡电路产生的时钟信号。K1 用来给微控制器发送复位信号^[9]。微控制器电路如图 3-1 所示。

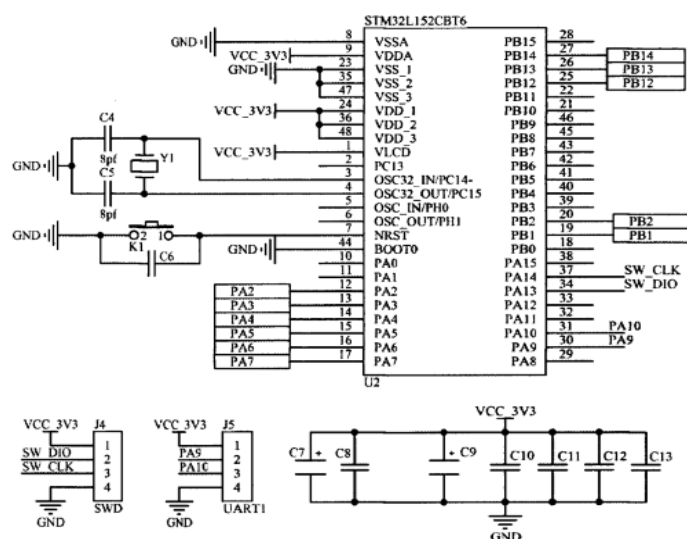


图 3-1 微控制器电路

3.2 护线桩姿势检测模块电路设计

MPU6050 可检测微小的角变化。陀螺模块应使用用于检测护线桩桩的角度变化，以确定其是否松开或无故打开，以实现护线桩和压力调节器的检测功能。MPU6050 及其外围电路如图 3-2 所示。

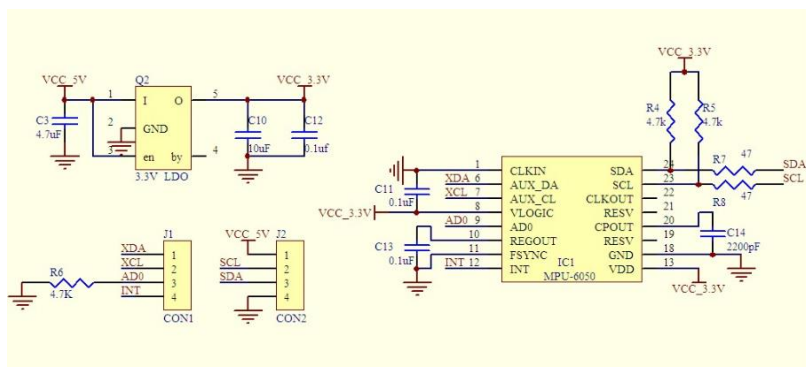


图 3-2 三轴加速度传感器电路

3.3 温湿度检测模块电路设计

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。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。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。^[10]单线制串行接口，使系统集成变得简易快捷。

VCC 表示电源，VSS 表示接地。P0 端口（P0.0~P0.7，39~32 引脚）：P0 口是一个漏极开路的 8 位双向 I/O 口。作为输出端口，每个引脚能驱动 8 个 TTL 负载，对端口 P0 写入“1”时，可以作为高阻抗输入。在访问外部程序和数据存储器时，P0 口也可以提供低 8 位地址和 8 位数据的复用总线。此时，P0 口内部上拉电阻有效。P1 端口（P1.0~P1.7，1~8 引脚）：P1 口是一个带内部上拉电阻的 8 位双向 I/O 口。P1 的输出缓冲器可驱动 4 个 TTL 输入。对端口写入 1 时，通过内部的上拉电阻把端口拉到高电位，这是可用作输入口。P1 口作输入口使用时，因为有内部上拉电阻，那些被外部拉低的引脚会输出一个电流。^[11]

此外，P1.0 和 P1.1 还可以作为定时器/计数器 2 的外部技术输入（P1.0/T2）和定时器/计数器 2 的触发输入（P1.1/T2EX）。P2 端口（P2.0~P2.7，21~28 引脚）：P2 口是一个带内部上拉电阻的 8 位双向 I/O 端口。P2 的输出缓冲器可以驱动 4 个 TTL 输入。对端口写入 1 时，通过内部的上拉电阻把端口拉到高电平，这时可用作输入口。P2 作为输入口使用时，因为有内部的上拉电阻，那些被外部信号拉低的引脚会输出一个电流。

在对 Flash ROM 编程和程序校验期间，P2 也接收高位地址和一些控制信号。P3 端口（P3.0~P3.7，10~17 引脚）：P3 是一个带内部上拉电阻的 8 位双向 I/O 端口。P3 的输出缓冲器可驱动 4 个 TTL 输入。对端口写入 1 时，通过内部的上拉电阻把端口拉到高电位，这时可用作输入口。P3 做输入口使用时，因为有内部的上拉电阻，那些被外部信号拉低的引脚会输入一个电流。

在对 Flash ROM 编程或程序校验时，P3 还接收一些控制信号。RST（9 引脚）：复位输入。当输入连续两个机器周期以上高电平时为有效，用来完成单片机的复位初始化操作。ALE/PROG（30 引脚）：地址锁存控制信号（ALE）是访问外部程序存储器时，锁存低 8 位地址的输出脉冲。在 Flash 编程时，此引脚也用作编程输入脉冲。

在一般情况下，ALE 以晶振六分之一的固定频率输出脉冲，可用来作为外部定时器或时钟使用。然而，特别强调，在每次访问外部数据存储器时，ALE 脉冲将会跳过。如果需要，通过将地址位 8EH 的 SFR 的第 0 位置“1”，ALE 操作将无效。这一位置“1”，ALE 仅在执行 MOVX 或 MOV 指令时有效。否则，ALE 将被微弱拉高。这个 ALE 使能标志位（地址位 8EH 的 SFR 的第 0 位）的设置对微控制器处于外部执行模式下无效。

PSEN(29 引脚): 外部程序存储器选通信号是外部程序存储器选通信号。当 AT89C51RC 从外部程序存储器执行外部代码时，PSEN 在每个机器周期被激活两次，而访问外部数据存储器时，PSEN 将不被激活。

EA/VPP (31 引脚): 访问外部程序存储器控制信号。为使能从 0000H 到 FFFFH 的外部程序存储器读取指令，EA 必须接 GND。注意加密方式 1 时，EA 将内部锁定位 RESET。为了执行内部程序指令，EA 应该接 VCC。XTAL1 (19 引脚): 振荡器反相放大器和内部时钟发生电路的输入端。 XTAL2 (18 引脚): 振荡器反相放大器的输入端。DHT11 外围电路如图 3-3 所示。^[12]

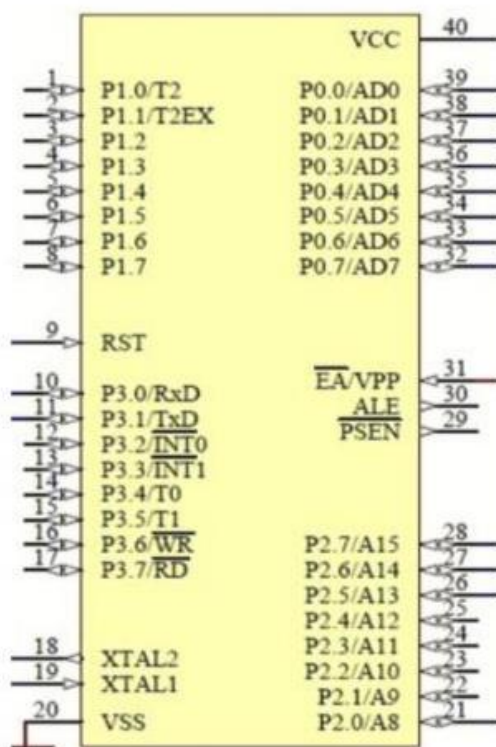


图 3-3 温湿度传感器 DHT11 电路

4. 系统软件设计

4.1 软件架构

根据上述具体功能，整个软件分为两个功能块，即“护线桩的管理”和“通信服务”。软件体系结构是通过集成硬件设计的元素来分层的，软件体系结构是作为一个整体来设计的。

应用层主要是护线桩的管理和通信。护线桩管理的主要任务是定位功能和报警功能，获得护线桩的状态。通信任务允许实时获取设备状态信息，并进行声明和接收信息，管理系统和通信模块进行相互通信。

底层由微控制器外围设备的外部模块驱动。HAL library 为加密和 USB 等中间组件提供了第二套支持，不需要用户。相反，它允许用户专注于功能开发。HAL 库也是意法半导体目前广泛开发的固件库。底层驱动程序是执行相应模块功能的甲板。其中，MPU6050 模块具有角加速度和加速度检测功能，且精度很高，可以检测到微小的角度变化。陀螺模块应使用用于检测护线桩的角度变化，确定护线桩是松开的还是无故打开的，以确保护线桩和检测松开的情况。NB-IoT 模块驱动则是为应用层提供信息的发送与接收，实现上报与接收信息的功能^[13]。

4.2 开发环境与开发工具

目前主流的微控制器开发环境有三种，是 Keil、IAR 与 GNU ArmEmbedded Toolchain。本研究使用的意法半导体的微控制器。^[14]本次选用的 Keil 是一款应用于编程的软件开发工具，它拥有强大的仿真器、调试器、宏汇编器、单板计算机以及符合生产标准的 Keil 编译器，在嵌入式软件开发的基础上有深层次的研发进展，能够在嵌入式开发上准确模拟硬件配置，以详尽的测试报告对定量的数据进行简单合理的分析。

其中 BL51 链接器/定位器是组合由 C51 和 A51 产生的可重定位的目标模块，生成绝对目标模块。A51 宏汇编是从 89C51 汇编源代码产生可重定位的目标模块。OH51 目标文件至 HEX 格式的转换器，从绝对目标模块生成 Intel Hex 文件。μVision4 for Windows 是一个集成开发环境，它将项目管理、程序调试、源代码编辑等组合在一个功能强大的环境中。RTX-51 实时操作系统简化了复杂的实时应用软件项目的设计。LIB51 库管理器从目标模块生成连接器可以使用的库文件。Keil 生成的目标代码效率非常之高，多数语句生成的汇编代码很紧凑，容易理解。在开发大型软件时更能体现高级语言的优势。与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易

用。所以选用 Keil 用于本研究的开发，生成代码也便于理解。^[15]

4.3 手机终端软件设计

该系统的移动终端主要是为巡线人员设计的，是在 Android 系统上开发的，可以接收异常信息并与服务器数据进行交互。在护线桩发生故障时，能保证定位和报警功能。以下设计的是手机终端软件设计流程图，首先利用安卓模拟器进行登录操作，并开启后台服务，同时向服务器发送请求，等待接收服务器响应。在实现功能过程中，GPS+北斗进行地图定位，将定位过的地点描绘成地图轨迹图显示出来。同时接收服务器端的数据响应，进行信息显示，其中包括温湿度显示和已经设定好的每 5 分钟一次的历史数据查询功能。具体的手机终端软件设计流程图如图 4-1 所示^[16]。

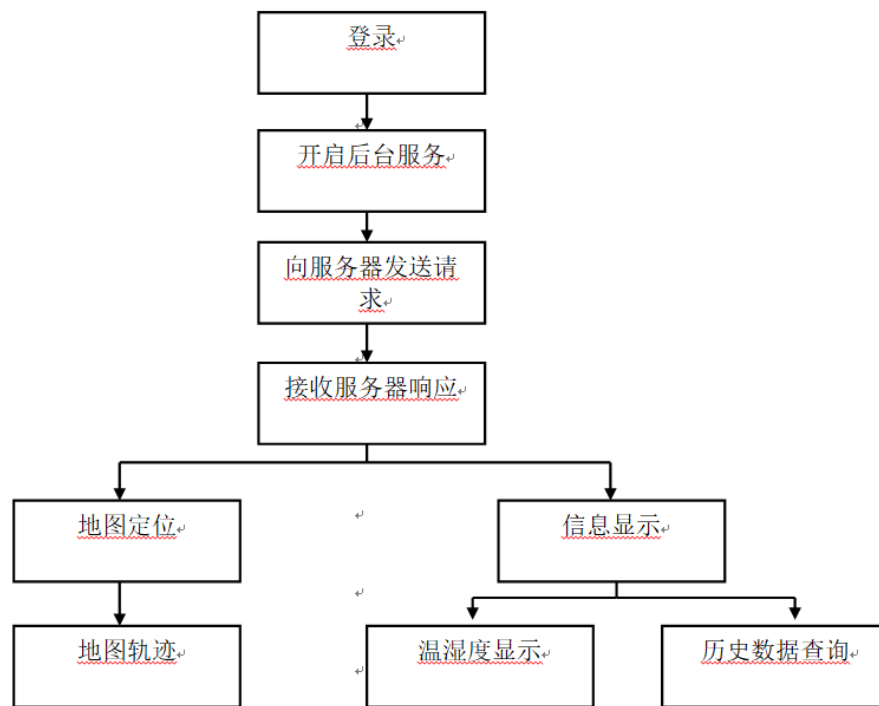


图 4-1 手机终端软件设计流程图

App 主要分为三大模块，首先登录模块将用户信息存储到数据库，在进行登录操作时，调用该数据库信息进行登录。如图 4-2 所示。

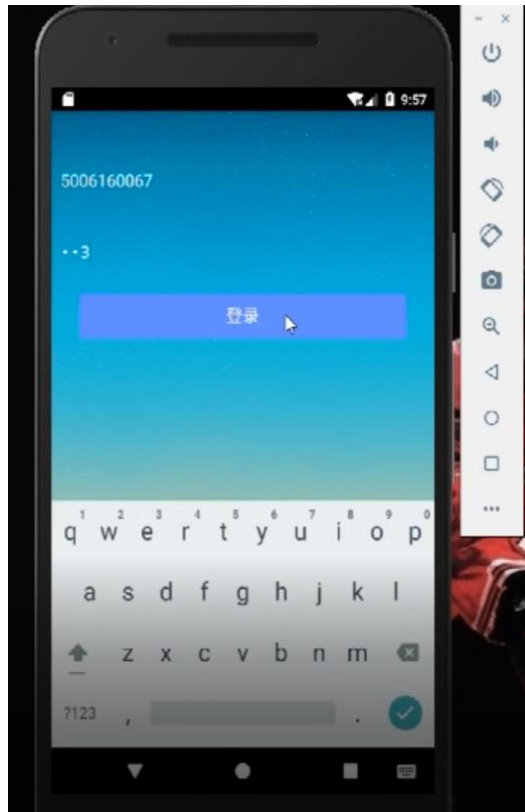


图 4-2 手机登录界面

其次是菜单模块，主要进行导航功能。如图 4-3 所示。



图 4-3 手机菜单界面

再则是定位显示模块，主要进行护线桩定位显示。如图 4-4 所示。

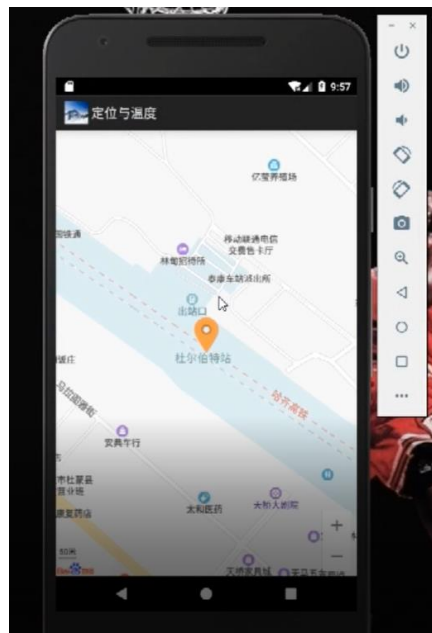


图 4-4 定位界面

最后是温湿度显示模块，主要进行护线桩周围环境的显示。如图 4-5 所示。



图 4-5 温湿度采集界面

5. 系统测试

5.1 运行状态测试

运行状态测试将验证智能护线桩是否能够识别设备故障，测试将是故障定位测试。以模拟定位异常时，第一步是 GPS +北斗模块连接在一个密闭空间，模拟定位，再发起申请，看看是否有异常，根据智能护线桩显示结果；第二步是在智能保护面板运行时，将控制面板与 GPS+ 北斗模块断开。测试结果如图 5-1 所示。

测试状态	测试结果
断开无法定位异常	检测到无法定位
使用时无法定位	检测到定位失败

图 5-1 故障检测测试

5.2 定位状态测试

定位状态测试将对智能护线桩的 GPS+北斗模块电路进行测试，测试条件为正常状态下，为定位模块进行定位变化，用该模块检测电路测量定位位置，得到的定位信息如图 5-2 所示。

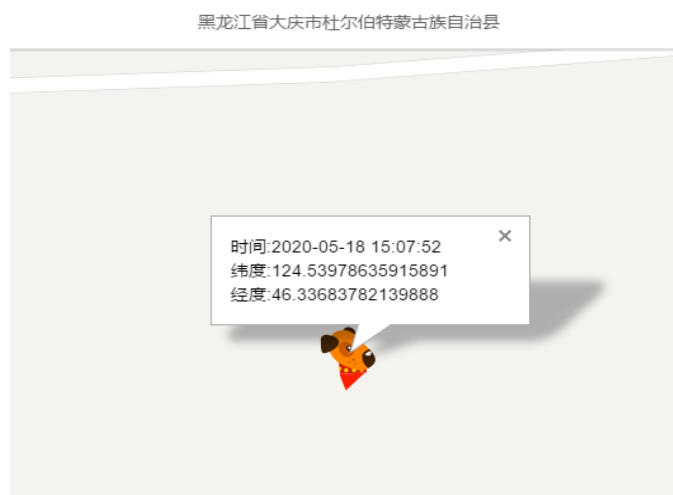


图 5-2 实时定位信息图

5.3 温湿度状态测试

温湿度测试将对智能护线桩的温湿度模块电路进行测试，测试条件为正常状态下，为温湿度模块进行人为温湿度变化，用该模块检测电路测量温湿度，得到的温湿度变化曲线如图 5-3 所示。



图 5-3 温湿度实时曲线图

5.4 护线桩姿态测试

护线桩姿态测试将对智能护线桩的三轴加速度传感器 MPU6050 模块进行测试，测试条件为正常状态下，为姿态测试模块进行角度变化，护线桩姿态信息如图 5-4 所示。



图 5-4 实时角度信息变化图

6. 总结与展望

6.1 总结

本文主要研究了基于 NB-IoT 的分布式通讯线路监测护线桩。为了收集和分析护线桩的状态信息，本文提出了一种护线桩管理系统。该系统通过创建一个智能护线桩管理应用程序来管理大量的智能护线桩，从而实现了物联网开放平台上的智能护线桩管理。

本研究主要需要进行基于 NB-IoT 的护线桩管理的方案演示与设计。在对基于 NB-IoT 的护线桩管理进行功能需求与运行基础上，确定了该系统的组成；支持 NB-IoT 的智能护线桩系统硬件设计与实现。以 STM32L 为微控制器设计控制模块。连接 MPU6050 三轴加速度传感器等模块；系统测试应包括所需的各种性能指标，包括设备性能指标、通信质量和应用护线桩管理的延迟。所有的测试结果都符合预期，系统能够满足保护桩管理的所有要求。

6.2 展望

本文设计的基于 NB-IoT 的分布式通讯线路监测护线桩系统进行管理，但是在以下几个方面仍需要进一步的探索：

对通信护线桩的智能化改造与管理的想法以及形成具体实施方案，是近几年来，伴随着相关运营商管理、运维部门的要求而逐步逐一实现的，至今为止，文中所描述的解决方案已经在多个地方的运营商实现了试点运营和部分批量商用。由于每个地方 NB 网络的信号差异大、软件管理平台的优化程度不同、不同设备厂家产品参数的差异化所带来的兼容问题，在后续的运营过程中会显现出很多的问题需要去解决，这就需要不断优化产品，着实解决运维过程中发生的实际问题。但是总体上看，由于 NB-IoT 技术的独特优势，结合智能护线桩的高效管理要优于传统护线桩管理的优势，既能满足了物联网发展的实际应用的需求，并拓宽了物联网的应用范围，又能为进一步探索 NB-IoT 技术在其他行业项目应用提供了相关应用案例，进而为智慧城市建设工作开展创造了标杆示范作用。

参考文献

- [1] 任小强. 基于 NB-IoT 技术的智能井盖监测系统研究[J]. 邮电设计技术, 2019(3):56-57.
- [2] 朱豪坤. 基于姿态传感器 MPU6050 的卡尔曼滤波应用[J]. 中国科技纵横, 2016(19):21-22.
- [3] 周云生. 物联网模式下通信井盖智能化升级解决方案[J]. 信息技术与信息化, 2019(12):5-8.
- [4] 余丹莉. 移动通信运营商集团客户满意度影响因素分析研究[J]. 市场研究, 2015(2):14-16.
- [5] 王小祥. 基于 ESP8266 的 DHT11 温湿度检测[J]. 数字技术与应用, 2019(8):34-35.
- [6] 电子信息产业网. 意法半导体(ST)免费的开发工具让开发者在所有桌面环境使用 STM32 微控制器[N]. 中国电子报, 2016-05-24(1).
- [7] 马爱丽, 曹梦宇, 等. 基于智能井盖的物联网+市政一体化系统[J]. 物联网技术, 2016, 6(3):105-107.
- [8] 张汇锋. 基于 NB-IoT 的车位管理系统研究与设计[D]. 山东: 山东大学, 2019:51-54.
- [9] 雷小燕. 基于移动边缘计算和 NB-IoT 的智能停车系统研究[J]. 信息通信 2019(6):190-192.
- [10] 朱代先, 刘冰冰, 等. 基于 NB-IoT 的智慧井盖监测系统设计与实现[J]. 计算及测量与控制, 2019, 27(10):55-29.
- [11] LG Electronics Inc.; Patent Issued for Method For Attempting Network Access From NB-IoT RAT (USPTO 10,694,447)[J]. Technology News Focus, 2020.
- [12] Donny Jackson. How NB-IoT bombed: An IoT tale of hubris and no-show[J]. Urgent Communications, 2020.
- [13] 刘利琴, 刘红梅. 窄带物联网下通信节点信息传输防碰撞仿真[J]. 计算机仿真, 2020, 37(06):165-168+424.
- [14] Telefonaktiebolaget LM (publ); Patent Application Titled "Enhanced NB-IoT CE Level Selection" Published Online (USPTO 20200163120)[J]. Telecommunications Weekly, 2020.
- [15] MediaTek Inc.; Patent Issued for Method And Apparatus For Uplink Power Consumption Reduction In NB-IoT (USPTO 10,645,658)[J]. Computer Technology Journal, 2020.
- [16] Naveen Kumar Maurya, Rajarshi Bhattacharya. Design of compact dual-polarized multiband MIMO antenna using near-field for IoT[J]. AEUE - International Journal of Electronics and Communications, 2020:117.

附录

支持 NB-IoT 的智能护线桩 PCB 板如图 1。

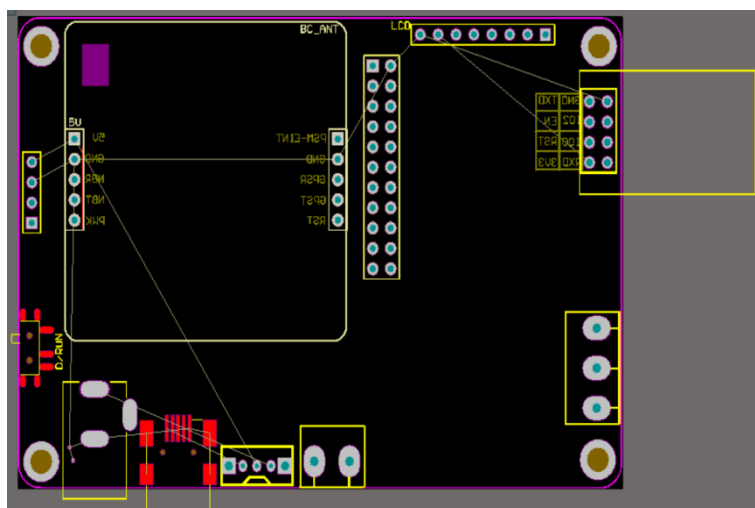


图 1 STM32 底板 PCB 图

核心板尺寸及模块封装如图 2。

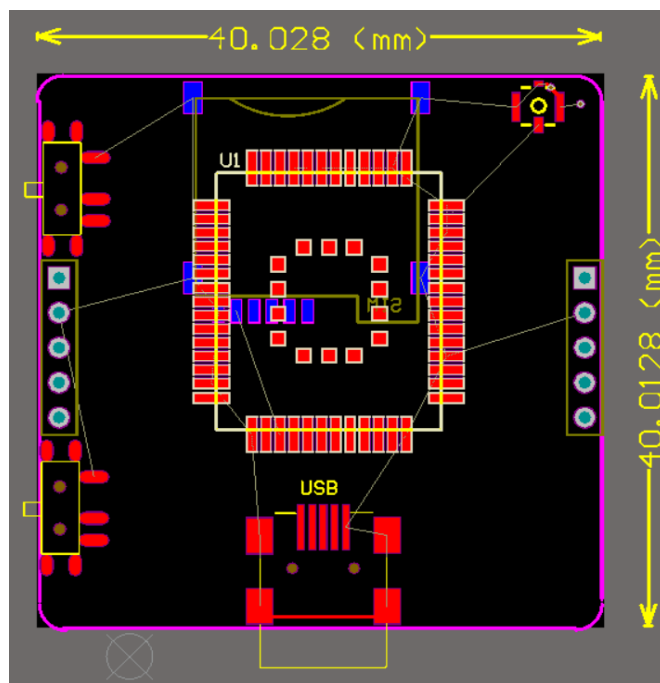


图 2 核心板尺寸及模块封装图

STM32L 发送数据核心代码如下所示：

```
#define DTH11DEV_ID          "hpwlwdwRr6EPdMe6JJ7KDc" //定义为温湿度的账号
#define DTH11DEV_KEY        "2S4SSGnhJRrCNa3EtkwD7FMpd2DhwwKh"
//定义为温湿度的密码
#define GPSDEV_ID           "hpwldws3wTJPabCjfGPncr"//定义为定位的账号
#define GPSDEV_KEY          "wKtZEGCwHH66y7xZQbFFXECQd586YkKX"
//定义为定位的密码
#define UploadtempID       "Uploadtemp"
#define gpsdataID          "gpsdata"
u8 HTTP_PostDataPkt(char *pkt,char *temp, char *humi,char *lat, char *lon);
u8 HTTP_PostPkt(char *pkt,char *udpdataid,char *devid,char *devkey,char *val1, char
*val2);
void ClearRAM(u8* ram,u32 n);
int main(void)
{
    char struart1[BUFLen];
    char struart1OK[BUFLen];
    char tempstr[5];
    char humistr[5];

    char HTTP_Buf[400];
    char HTTP_BufHEX[450];
    char latstr[BUFLen];
    char lonstr[BUFLen];
    u8 http_len=0;      //HTTP 组包数据长度
    if (SysTick_Config(SystemCoreClock / 1000))
    {
        /* Capture error */
        while (1);
    }
    LED_Init();
    uart1_init(115200);
    uart2_init(115200);
    uart3_init(115200);
    printf("\r\n ##### http://www.csgsm.com/ #####\r\n
#####("__DATE__ " - "__TIME__ ")#####\r\n");
    delay_init();
    //NBIoT_RESET();
    LED_RUN();
    while(DHT11_Init());
//液晶屏的显示驱动//
    OLED_Init();
    OLED_Clear();
```

```

    OLED_ShowString(0,0,"STM32+BC20+WIFI");
    OLED_ShowString(0,3,"NB IoT GPS BOARD");
    Uart1_SendStr("init stm32L COM1 \r\n"); // 'òÓ; ÐÃİç
    while(BC20_Init());
    BC20_INITGNSS();
    BC20_PDPACT();
    BC20_ConTCP();
    Delay(2000);
    while (1)
    {
        Clear_Buffer();
        Uart2_SendStr("AT+QGNSSRD=\"NMEA/RMC\" \r\n");
        Delay(2000);
        if(strstr((char *)buf_uart2.buf, "$GNRMC"))
        {
            ClearRAM((u8*)struart1, BUFLen);
            ClearRAM((u8*)struart1OK, BUFLen);
            ClearRAM((u8*)latstr, BUFLen);
            ClearRAM((u8*)lonstr, BUFLen);
            sprintf(struart1, "%s", buf_uart2.buf+34);
            Clear_Buffer();
            if(struart1[17]=='A')
            {
                strncpy(struart1OK, struart1, strlen(struart1)-8);
                struart1OK[2]='P';
                strncpy((char*)(latstr+strlen((char*)latstr)), &struart1[19], 2);
                strncpy((char*)(latstr+strlen((char*)latstr)), ".", 1);
                strncpy((char*)(latstr+strlen((char*)latstr)), &struart1[21], 2);
                strncpy((char*)(latstr+strlen((char*)latstr)), &struart1[24], 4);

                strncpy((char*)(lonstr+strlen((char*)lonstr)), (char*)&struart1[31], 3);
                strncpy((char*)(lonstr+strlen((char*)lonstr)), ".", 1);
                strncpy((char*)(lonstr+strlen((char*)lonstr)), (char*)&struart1[34], 2);
                strncpy((char*)(lonstr+strlen((char*)lonstr)), (char*)&struart1[37], 4);
                http_len=0;

                http_len=HTTP_PostPkt(HTTP_Buf, gpsdataID, GPSDEV_ID, GPSDEV_KEY, lonstr, latstr);
                printf("HTTP_Buf=%s\r\n", (char*)HTTP_Buf);
                Str2Hex((char*)HTTP_Buf, (char*)HTTP_BufHEX);
                if(BC20_Status.BC20conn_ok)
                {
                    BC20_SenddataHEXInt(http_len, (u8*)HTTP_BufHEX);
                }
                printf("HTTP_BufHEX=%s\r\n", (char*)HTTP_BufHEX);
            }
        }
    }

```

```

        BC20_RECData();
        GPIO_ToggleBits(GPIOC,GPIO_Pin_8);

        printf("\r\n");
    }
    else if(struart1[17]=='V')
    {
        printf("重连服务器! \r\n");
    }
    if(BC20_Status.BC20conn_ok)
    {
        ClearRAM((u8*)humistr,5);
ClearRAM((u8*)tempstr,5);
        ClearRAM((u8*)HTTP_Buf,400);
ClearRAM((u8*)HTTP_BufHEX,450);
        sprintf(tempstr, "%d",temp);
        sprintf(humistr, "%d",humi);
        http_len=0;

http_len=HTTP_PostPkt(HTTP_Buf,UploadtempID,DTH11DEV_ID,DTH11DEV_KEY,temp
pstr,humistr);
        printf("HTTP_Buf=%s\r\n",(char*)HTTP_Buf);
Str2Hex((char*)HTTP_Buf,(char*)HTTP_BufHEX);
        BC20_SenddataHEXInt(http_len,(u8*)HTTP_BufHEX);
        printf("HTTP_BufHEX=%s\r\n",(char*)HTTP_BufHEX);
    }
}
BC20_RECData();
}
void ClearRAM(u8* ram,u32 n)
{
    u32 i;
    for (i = 0;i < n;i++)
    {
        ram[i] = 0x00;
    }
}
/**
 * @brief   Decrements the TimingDelay variable.
 * @param   None
 * @retval  None
 */
void TimingDelay_Decrement(void)
{

```

```

    if (TimingDelay != 0x00)
    {
        TimingDelay--;
    }
}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t* file, uint32_t line)
{
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */

    /* Infinite loop */
    while (1)
    {
    }
}
#endif

```

App 核心代码：

```

public class BaseActivity extends FragmentActivity{
    String new1;
    String new2;
    public Dialog loadDialog;
    private ProgressDialog progressDialog;

    protected Context mContext;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mContext=this;
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);

    }
    public void showToast(int textResources){
        Toast toast= Toast.makeText(mContext,textResources,Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }

    public void showToast(String text){
        Toast toast= Toast.makeText(mContext, text, Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.CENTER, 0, 0);
    }
}

```

```

        toast.show();
    }

    public void resetTitle(String s) {

        TitleView titleView=(TitleView)findViewById(R.id.t);
        titleView.setTitle(s);
    }

    public static void into(Context context,String news1,String news2){
        Intent intent =new Intent(context, BaseActivity.class);
        intent.putExtra("news1",news1);
        intent.putExtra("news2",news2);
        context.startActivity(intent);
    }

    public void showProgressDialog(String title, String message) {
        if (progressDialog == null) {

            progressDialog = ProgressDialog.show(this, title,
                message, true, false);
        } else if (progressDialog.isShowing()) {
            progressDialog.setTitle(title);
            progressDialog.setMessage(message);
        }

        progressDialog.show();

    }

    /*
     * 隐藏提示加载
     */
    public void dissProgressDialog() {

        if (progressDialog != null && progressDialog.isShowing()) {
            progressDialog.dismiss();
        }

    }

}

```

登录部分：

```
public class LoginActivity extends BaseActivity {

    EditText editText;
    EditText editText2;
    public static String user;
    public static String userLike;

    public static  ArrayList<String> baoxius=new ArrayList<String>();
    public static  ArrayList<String> liuyans=new ArrayList<String>();
    public static  ArrayList<String> pingluns=new ArrayList<String>();

    @SuppressWarnings("rawtypes")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.longin_activity);
//    Bmob.initialize(this, getResources().getString(R.string.bomb));
        editText = (EditText) findViewById(R.id.editText1);
        editText2 = (EditText) findViewById(R.id.editText2);
        setTitle(getResources().getString(R.string.app_name));

    }

    public void denglu(View v) {

        Intent intent=new Intent(LoginActivity.this,MenuActivity.class);
        startActivity(intent);
    }

    public void zhuce(View v) {

    }

    public void luru(View v) {

    }

}

public static void into(Context context) {
    Intent intent = new Intent(context, LoginActivity.class);
```



```

        context.startActivity(intent);
    }
}

```

百度地图接入部分：

```

import java.util.List;

public class MapActivity extends Activity {
    MapView mMapView = null;
    BaiduMap mBaiduMap;
    public LocationClient mLocationClient = null;
    MyLocationListener listener = new MyLocationListener();
    boolean isFirstLoc = true;
    String permissionInfo;
    private LinearLayout toolback;
    TextView messBack;
    FrameLayout backHome;

    private final int SDK_PERMISSION_REQUEST = 127;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        SDKInitializer.initialize(this);

        mMapView = (MapView) this.findViewById(R.id.bmapView);
        mBaiduMap = mMapView.getMap();
        mBaiduMap.setMapType(BaiduMap.MAP_TYPE_NORMAL);

        //定位到自己的位置

        mLocationClient = new LocationClient(this);
        //声明 LocationClient 类
        mLocationClient.registerLocationListener(listener);
        initLocation();
        mLocationClient.start();
    }
    private void initLocation() {
        LocationClientOption option = new LocationClientOption();
        option.setLocationMode(LocationClientOption.LocationMode.Hight_Accuracy);
        //可选，默认高精度，设置定位模式，高精度，低功耗，仅设备

```

```

option.setCoorType("bd09ll");
//可选，默认 gcj02，设置返回的定位结果坐标系

int span = 1000;
option.setScanSpan(span);
//可选，默认 0，即仅定位一次，设置发起定位请求的间隔需要大于等于 1000ms
才是有效的

option.setIsNeedAddress(true);
//可选，设置是否需要地址信息，默认不需要

option.setOpenGps(true);
//可选，默认 false,设置是否使用 gps

option.setLocationNotify(true);
//可选，默认 false，设置是否当 GPS 有效时按照 1S/1 次频率输出 GPS 结果

option.setIsNeedLocationDescribe(true);
//可选，默认 false，设置是否需要位置语义化结果，可以在
BDLocation.getLocationDescribe 里得到，结果类似于“在北京天安门附近”

option.setIsNeedLocationPoiList(true);
//可选，默认 false，设置是否需要 POI 结果，可以在 BDLocation.getPoiList 里
得到

option.setIgnoreKillProcess(false);
//可选，默认 true，定位 SDK 内部是一个 SERVICE，并放到了独立进程，设置
是否在 stop 的时候杀死这个进程，默认不杀死

option.SetIgnoreCacheException(false);
//可选，默认 false，设置是否收集 CRASH 信息，默认收集

option.setEnableSimulateGps(false);
//可选，默认 false，设置是否需要过滤 GPS 仿真结果，默认需要

mLocationClient.setLocOption(option);
}
public class MyLocationListener implements BDLocationListener {

    @Override
    public void onReceiveLocation(BDLocation location) {
        if (isFirstLoc) {
            isFirstLoc = false;
            //获取定位结果

```

```

StringBuffer sb = new StringBuffer(256);

sb.append("time : ");
sb.append(location.getTime());    //获取定位时间

sb.append("\nerror code : ");
sb.append(location.getLocType());    //获取类型类型

sb.append("\nlatitude : ");
sb.append(location.getLatitude());    //获取纬度信息

sb.append("\nlontitude : ");
sb.append(location.getLongitude());    //获取经度信息

sb.append("\nradius : ");
sb.append(location.getRadius());    //获取定位精准度

if (location.getLocType() == BDLocation.TypeGpsLocation) {

    // GPS 定位结果
    sb.append("\nspeed : ");
    sb.append(location.getSpeed());    // 单位： 公里每小时

    sb.append("\nsatellite : ");
    sb.append(location.getSatelliteNumber());    //获取卫星数

    sb.append("\nheight : ");
    sb.append(location.getAltitude());    //获取海拔高度信息，单位米

    sb.append("\ndirection : ");
    sb.append(location.getDirection());    //获取方向信息，单位度

    sb.append("\naddr : ");
    sb.append(location.getAddrStr());    //获取地址信息

    sb.append("\ndescribe : ");
    sb.append("gps 定位成功");

} else if (location.getLocType() == BDLocation.TypeNetWorkLocation) {

    // 网络定位结果
    sb.append("\naddr : ");
    sb.append(location.getAddrStr());    //获取地址信息

```

```

        sb.append("\noperationers : ");
        sb.append(location.getOperators());    //获取运营商信息

        sb.append("\ndescribe : ");
        sb.append("网络定位成功");

    } else if (location.getLocType() == BDLocation.TypeOffLineLocation) {

        // 离线定位结果
        sb.append("\ndescribe : ");
        sb.append("离线定位成功，离线定位结果也是有效的");

    } else if (location.getLocType() == BDLocation.TypeServerError) {

        sb.append("\ndescribe : ");
        sb.append("服务端网络定位失败，可以反馈 IMEI 号和大体定位
时间到 loc-bugs@baidu.com，会有人追查原因");

    } else if (location.getLocType() == BDLocation.TypeNetWorkException)
    {

        sb.append("\ndescribe : ");
        sb.append("网络不同导致定位失败，请检查网络是否通畅");

    } else if (location.getLocType() == BDLocation.TypeCriteriaException) {

        sb.append("\ndescribe : ");
        sb.append("无法获取有效定位依据导致定位失败，一般是由于手
机的原因，处于飞行模式下一般会造成这种结果，可以试着重启手机");

    }

    sb.append("\nlocationdescribe : ");
    sb.append(location.getLocationDescribe());    //位置语义化信息
    List<Poi> list = location.getPoiList();    // POI 数据
    if (list != null) {
        sb.append("\npoilist size = ");
        sb.append(list.size());
        for (Poi p : list) {
            sb.append("\npoi= ");
            sb.append(p.getId() + " " + p.getName() + " " + p.getRank());
        }
    }
}

```

```

        LatLng lpoint = new LatLng(46.876709, 124.45484);
        BitmapDescriptor bitmapDescriptorFactory =
        BitmapDescriptorFactory.fromResource(R.drawable.point);
        OverlayOptions options = new
        MarkerOptions().position(lpoint).icon(bitmapDescriptorFactory);
        mBaiduMap.addOverlay(options);
        MapStatus mapstat = new
        MapStatus.Builder().target(lpoint).zoom(18).build();
        MapStatusUpdate mapStatusUpdate =
        MapStatusUpdateFactory.newMapStatus(mapstat);
        mBaiduMap.setMapStatus(mapStatusUpdate);
    }
}
}
}

```

致 谢

在校的这四年时间里我很感谢老师们对我的悉心指导，在这段大学时光里，不仅使我的知识结构和学习能力提升了不少，更重要的是综合素质得到了提高。值此论文顺利完成之际，我真诚地感谢给我无私帮助的指导教师，他的教导给本文的论述提供了理论基础。从论文的选题到技术路线的选择到系统的开发构造，每一个部分老师的耐心教导对我影响很大，老师认真的工作态度，给我留下了深刻的印象，在我的心中树立了榜样。

感谢朋友们的四年陪伴，他们灵活思考问题的方式，严谨的态度和扎实的专业知识都给我留下了深刻的印象。没有你们就没有这四年美好的回忆，虽然时有吵闹，但是更多还是陪伴是共同成长。

最后深深地感谢陪我成长的父母，每次当我遇到困难的时候，父母总是坚定地鼓励我，让我继续坚持，克服困难。这么多年来，他们在精神上和物质上的无私帮助，让我坚定了人生理想的信念，谢谢你们的支持。如果没有你们，我的学习生涯也不会顺利通过。只有做出成绩让你们骄傲，相信自己能达到目标。

我谨向在大学四年中帮助过我的所有老师和同学表示衷心的感谢和美好的祝愿！