**西安电子科技大学优研面试**

**科研实践报告**



**题目：电动车定位系统**

**学校：河北科技大学**

**专业：物联网工程**

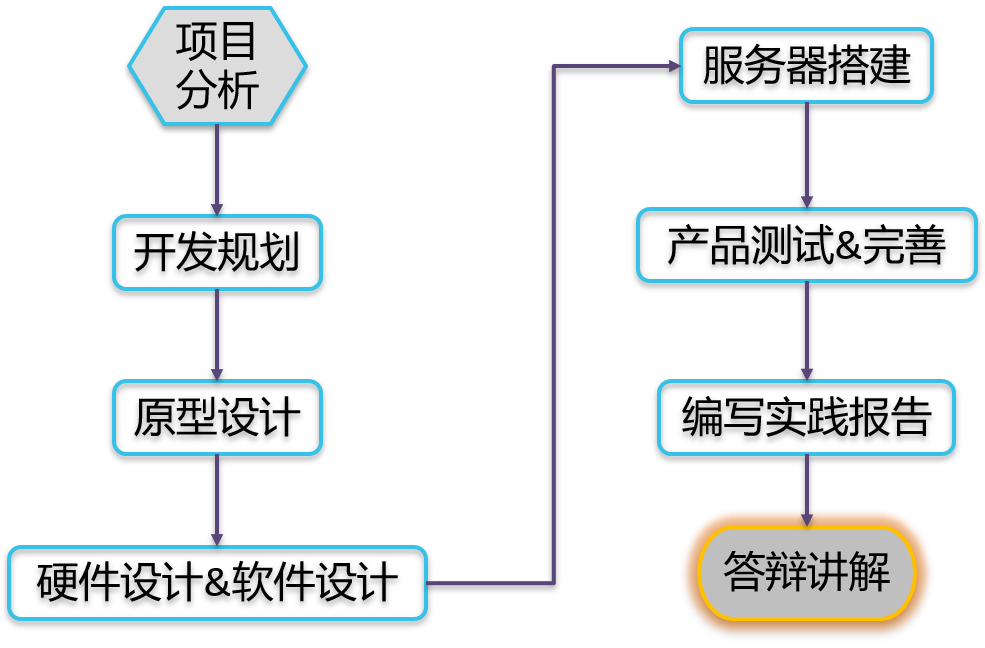
**作者：李圭印**

**电动车定位系统**

**介绍**

该系统是一款以 STM32 单片机和 A9G 模块为底层组件，阿里云服务器为中间件，APP为服务程序的物联网生活应用。本项目分为硬件端、服务器端和软件端。硬件端用于获取定位数据并将其发送至远程服务器，核心处理模块为 STM32F103C8T6，通过 A9G 模块中自带的GPS和GPRS进行卫星定位和2.5G通信。 APP 端通过用户操作向远程服务器请求电动车的位置信息， 并根据返回的位置信息，在手机地图页面上进行定位的追踪。服务器端负责远程存储定位系统发来的位置信息，并向APP端提供有效的位置信息，降低系统耦合度，提高开发效率。该系统可对大学生电动车丢失问题提供一定的安全保障。

**执行总结**



**目录**

[**电动车定位系统** 2](#_Toc50664093)

[**介绍** 2](#_Toc50664094)

[**执行总结** 2](#_Toc50664095)

[一、功能需求 1](#_Toc50664096)

[1.1 功能描述 1](#_Toc50664097)

[1.2 主要功能结构 1](#_Toc50664098)

[二、原型设计 2](#_Toc50664099)

[2.1 APP原型设计 2](#_Toc50664100)

[2.1.1 系统主界面 2](#_Toc50664101)

[2.1.2 系统二级页面 3](#_Toc50664102)

[2.2 定位系统原型设计 3](#_Toc50664103)

[2.2.1 A9G模块 3](#_Toc50664104)

[2.2.2 STM32模块 4](#_Toc50664105)

[2.2.3 电源模块 4](#_Toc50664106)

[2.2.4 外壳设计 5](#_Toc50664107)

[三、技术路线 6](#_Toc50664108)

[3.1 APP设计 6](#_Toc50664109)

[3.1.1 页面 6](#_Toc50664110)

[3.1.2 地图 6](#_Toc50664111)

[3.1.3 技术选型 6](#_Toc50664112)

[3.2 硬件技术 7](#_Toc50664113)

[3.2.1 AT指令 7](#_Toc50664114)

[3.2.2 串口通信 7](#_Toc50664115)

[3.2.3 技术选型 7](#_Toc50664116)

[3.3 服务器配置 8](#_Toc50664117)

[四、地图设计 8](#_Toc50664118)

[4.1 准备工作 8](#_Toc50664119)

[4.1.1 申请百度地图密钥 8](#_Toc50664120)

[4.1.2 下载百度LBS Android SDK 8](#_Toc50664121)

[4.2 地图的导入 8](#_Toc50664122)

[4.3 地图的定位 9](#_Toc50664123)

[4.4 定位请求 10](#_Toc50664124)

[4.5 目标追踪 10](#_Toc50664125)

[4.6 阶段成果 11](#_Toc50664126)

[4.6.1启动界面 11](#_Toc50664127)

[4.6.2 进入主界面 12](#_Toc50664128)

[4.6.3寻车页面 12](#_Toc50664129)

[4.6.4 用户须知 12](#_Toc50664130)

[4.6.5 警告图 13](#_Toc50664131)

[4.6.6 定位到我的位置 14](#_Toc50664132)

[4.6.7 定位请求 14](#_Toc50664133)

[4.6.8 追踪目标 14](#_Toc50664134)

[五、服务器设计 17](#_Toc50664135)

[5.1 开发说明 17](#_Toc50664136)

[5.2 数据库 17](#_Toc50664137)

[5.3 服务器 17](#_Toc50664138)

[5.4 通信说明 19](#_Toc50664139)

[六、定位系统设计 20](#_Toc50664140)

[6.1 A9G模块 20](#_Toc50664141)

[6.2 程序设计 20](#_Toc50664142)

[6.2.1 主函数设计 20](#_Toc50664143)

[6.2.2 GPRS设计 20](#_Toc50664144)

[6.2.3 GPS数据获取 21](#_Toc50664145)

[6.3 阶段成果 22](#_Toc50664146)

[6.3.1 数据获取&通信 22](#_Toc50664147)

[6.3.2 电路设计 22](#_Toc50664148)

[6.3.3 成品完善 22](#_Toc50664149)

[七、项目测试 23](#_Toc50664150)

[7.1 硬件测试结果 23](#_Toc50664151)

[7.1.1 A9G输出信息 23](#_Toc50664152)

[7.1.2 STM输出信息 23](#_Toc50664153)

[7.2 软件测试结果 23](#_Toc50664154)

[7.2.1 一键寻车 23](#_Toc50664155)

[7.2.2获取当前位置 24](#_Toc50664156)

[八、使用说明 26](#_Toc50664157)

[8.1 硬件使用说明 26](#_Toc50664158)

[8.2 软件使用说明 26](#_Toc50664159)

一、功能需求

1.1 功能描述

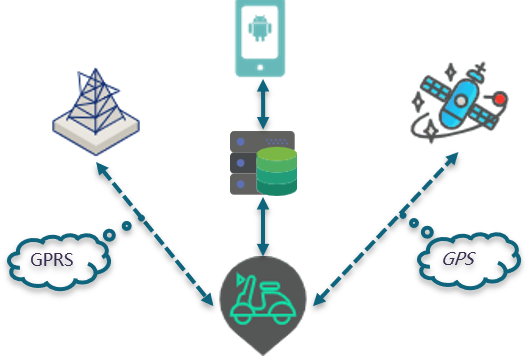
本项目分为硬件端和软件端。硬件端用于获取定位数据并将其发送至远程服务器。APP端通过用户操作向远程服务器请求电动车的位置信息，从而实现电动车远程实时定位。

1.2 主要功能结构

本产品顺应物联网时代发展潮流，采用GPS卫星定位、GPRS2.5G通信、计算机网络、Android等技术，其示意图如图2-1所示。

本项目的研究内容是基于32单片机的A9G模块以及Android-APP开发与使用。利用GPS具有全天候、高精度和自动测量的特点，来实现丢失电动车的校园定位。目前已有A9G模块，usb-tll接口和手机卡，能够借助APP实现定位功能。利用AiThinker Serial Tool V1.2.3将A9G模块、单片机和电脑连接起来,利用AT指令对A9G进行基本属性的配置（比如网络注册、配置IP地址和SIM信息），使电脑能够与该GPS模块匹配

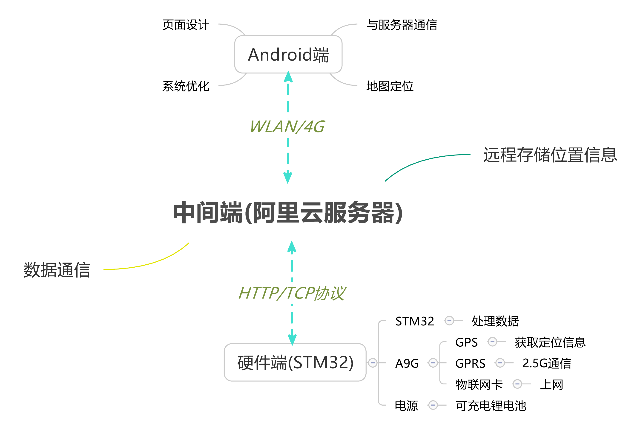
以阿里云服务器为中间媒介，定位系统将获取到得定位信息通过GET请求发送至服务器，APP端可以随时向服务器请求信息，获取电动车的位置信息，以便实现定位和轨迹追踪。



*图2-1*

二、原型设计

原型框架如图3-1所示



*图3-1*

2.1 APP原型设计

2.1.1 系统主界面

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 系统主界面 | | | |
| 操作角色 | 用户 | | | |
| 功能描述 | 1. 寻车宝系统主页。 2. 标题栏是寻车宝系统。 3. 导航栏Logo是自行车，TextView实时显示当前时间。 4. 图片是河北科技大学。 5. 两个按钮“通知公告”和“用户需知”，点击出现弹窗显示。 6. 底部导航栏有三个按钮，地图、我的和首页。 7. 点击“地图”跳转至相应界面。 | | | |
| 样片展示 |  |  |  |  |

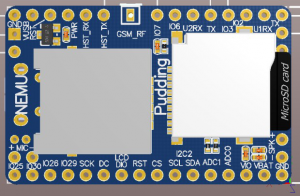
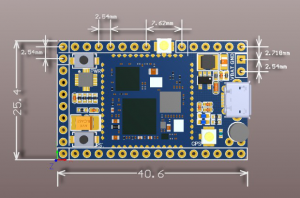
2.1.2 系统二级页面

|  |  |  |
| --- | --- | --- |
| 名称 | 系统二级页面 | |
| 操作角色 | 用户 | |
| 功能描述 | 1、寻车宝系统主页。  3、Logo是自行车。  5、底部导航栏有三个按钮，获取位置、一键寻车和我的主页。  6、点击“我的主页”跳转至系统首页面。  7、界面中间包含一个地图，是程序主功能块。 | |
| 样片展示 |  |  |

2.2 定位系统原型设计

2.2.1 A9G模块

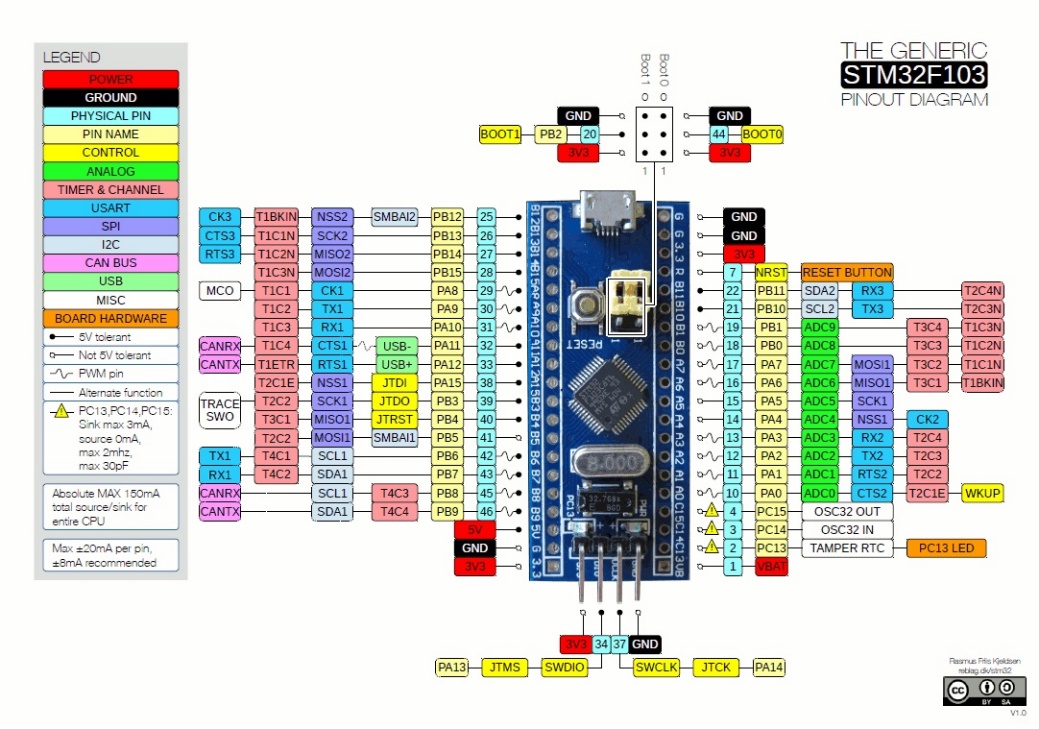
核心模块采用安信可科技的A9G模块，集GPS、GPRS为一体，物联网卡上网通信，节省空间占比，功能完善，是本系统核心模块的不二选择。其外观图如图3-2所示



*图3-2*

2.2.2 STM32模块

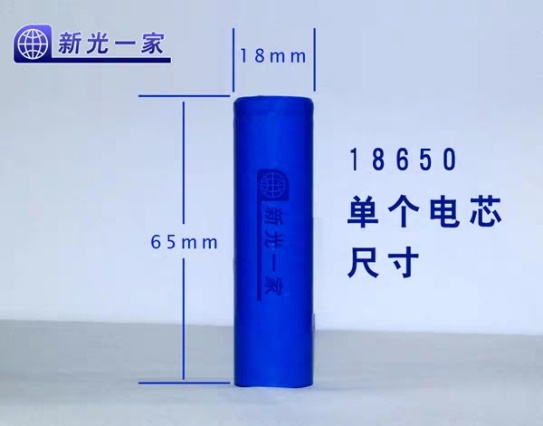
数据处理模块采用STM32F103C8T6，性价比高，功耗低，其引脚图如图3-3所示。本系统主要使用它的串口和电源模块。



*图3-3*

2.2.3 电源模块

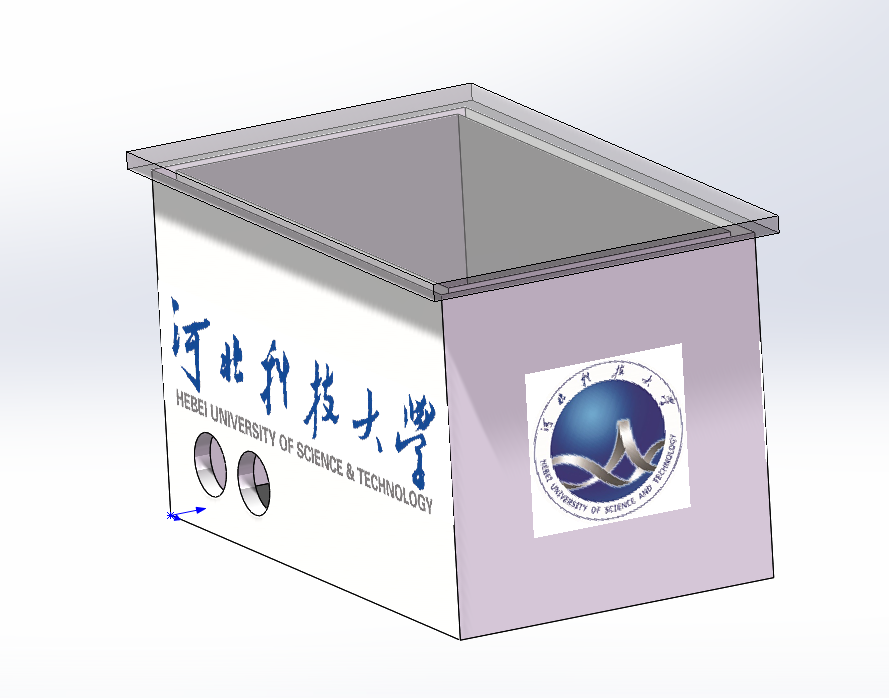
考虑到待电问题，本系统使用如下型号的可充电锂电池(2节)，充电方式为交流充电。待电时长为一周左右(系统正常工作情况下)。



*图3-4*

2.2.4 外壳设计

我们在保证体积小、坚硬度高的情况下，对其外观进行了设计，设计图如图3-5所示。外表刻有“河北科技大学”以及科大校徽，并有接口便于充电和拆卸。内部用来放置硬件系统。



*图3-5*

三、技术路线

3.1 APP设计

3.1.1 页面

(1) 线性布局

将控件组织在一个垂直或水平的形式。当布局方向设置为垂直时，它里面的所有子控件被组织在同一列中；当布局方向设置为水平时，所有子控件被组织在一行中。

(2) ImageView

在xml里面编写控件，背景是ImageView，给src资源；

(3) 实时更新时间

Java语句中System.currentTimeMillis()可以获取系统当前的时间，这里要实时显示就可以开启一个线程，然后通过handler发消息，来实时的更新TextView上显示的系统时间。具体就是写一个线程，线程里面无限循环，每隔一秒发送一个消息,在主线程里面处理消息并更新时间。

(4) 对话框

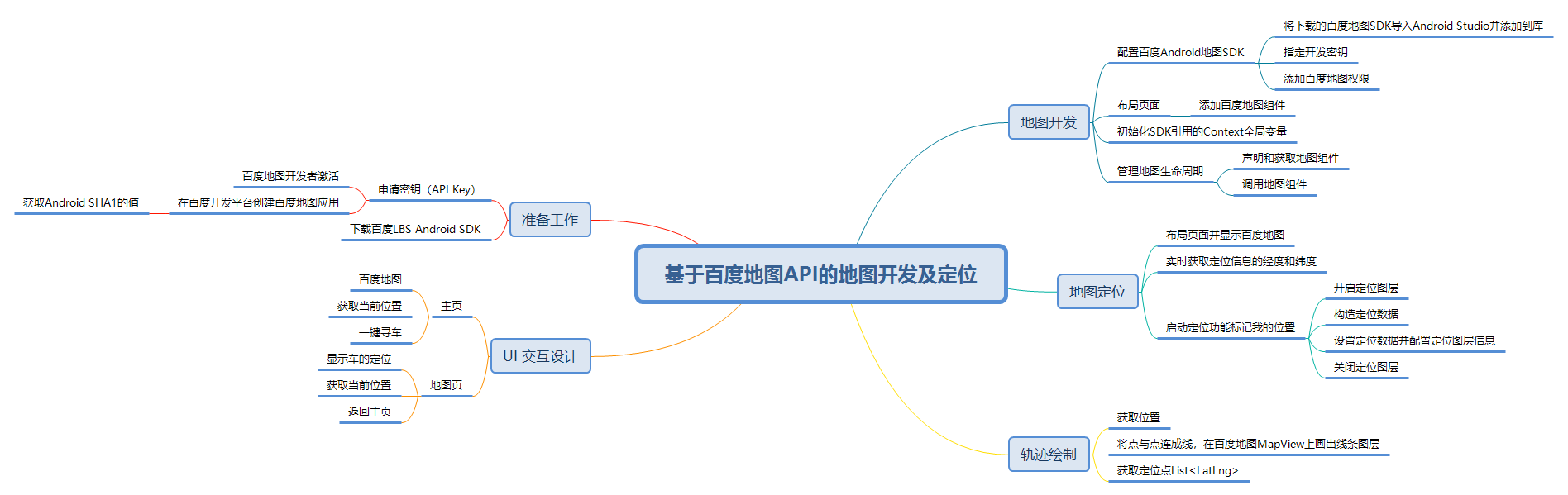
Button按钮用来显示通知公告和用户需知，Button按钮的点击时间，设置弹出对话框，使用AlertDialogBuilder类构建，再用AlertDialog类具体化。

(5) 页面跳转

Button的点击事件，创建Intent意图，设置好当前activity.class以及要跳转的页面activity.class。

3.1.2 地图

详细地图设计思路如图4-1所示



*图4-1*

3.1.3 技术选型

(1) 开发软件

Android Studio

(2) 开发环境

Win10、笔记本

(3) 开发语言

JAVA、XML

(4) 插件

LBS Android SDK

(5) 测试运行环境

安卓操作系统手机

3.2 硬件技术

3.2.1 AT指令

AT指令为STM32和A9G进行交互的串口通信语言。

所有AT指令必须以“AT”或“at”为开头，以回车作为结尾。返回响应通常紧随命令之后，样式为<回车><换行><响应内容><回车><换行>。

AT指令共有基础类、参数类和扩展类三种命令。

3.2.2 串口通信

STM32单片机的串口通信为全双工异步串行通信，可满足与A9G的通信需求。在此系统中使用了usart2和usart3。二者均采用定时中断接收和轮询发送。

1. usart2

该串口用于和GPRS进行通信，波特率为115200bps。

1. usart3

该串口用于和GPS进行通信，波特率为9600bps。其中GPS返回命令代号如图4-2所示



*图4-2*

3.2.3 技术选型

(1) 开发软件

Keil MDK5。

(2) 开发环境

Win10、笔记本

1. 调试工具

Keil-Debug、AiThinker Serial Tool V1.2.3.exe、J-Scope。

1. 模块选择

STM32F103C8T6、A9G、锂电池、7805稳压模块。

(5) 开发语言

C。

(6) 库函数

Keil::STM32F1xx\_DFP。

3.3 服务器配置

采用阿里云ECS算力服务器。其详细配置参数如表4-1所示

|  |  |
| --- | --- |
| CPU | 2核 |
| 内存 | 4GB |
| 实例类型 | I/O优化 |
| 操作系统 | Ubuntu 16.04 64位 |
| 公网IP | 39.97.235.93 |
| 端口 | 80 |
| 当前使用带宽 | 1Mbps |

*表4-1*

四、地图设计

4.1 准备工作

因为此APP是基于百度地图API开发的地图APP，所以在开发之前需要申请百度地图密钥（API Key），申请密钥需要进行百度地图开发者激活，并在百度地图平台创建百度地图应用。密钥申请成功之后，需要在百度地图开发平台下载百度LBS Android SDK,并根据需求勾选相应的功能资源。

4.1.1 申请百度地图密钥

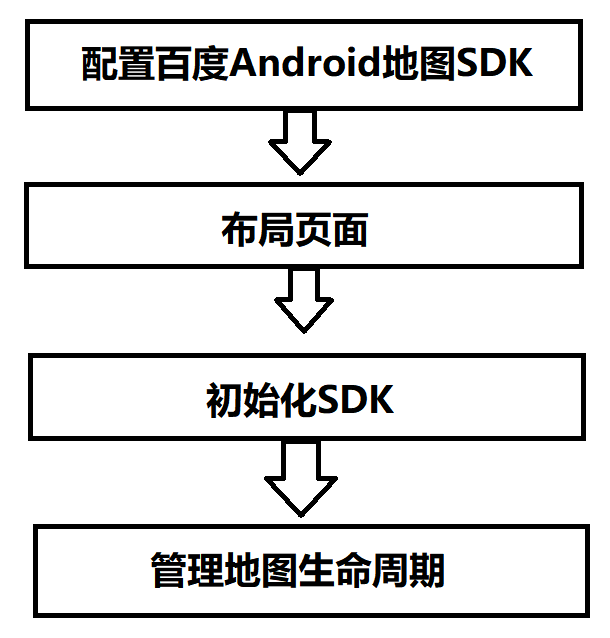
百度地图开发者激活，并在百度地图平台创建百度地图应用。

4.1.2 下载百度LBS Android SDK

采用百度提供的地图开发插件进行本系统的地图设计

4.2 地图的导入

首先将下载好的Android地图SDK添加到库中，在全局页面中指定开发密钥并添加百度地图权限。接下来需要在布局页面中添加百度地图组件。然后在onCreate方法下初始化地图SDK并获取地图组件。最后需要在生命周期方法中管理地图的生命周期。其大致过程如图4-1所示。



*图4-1*

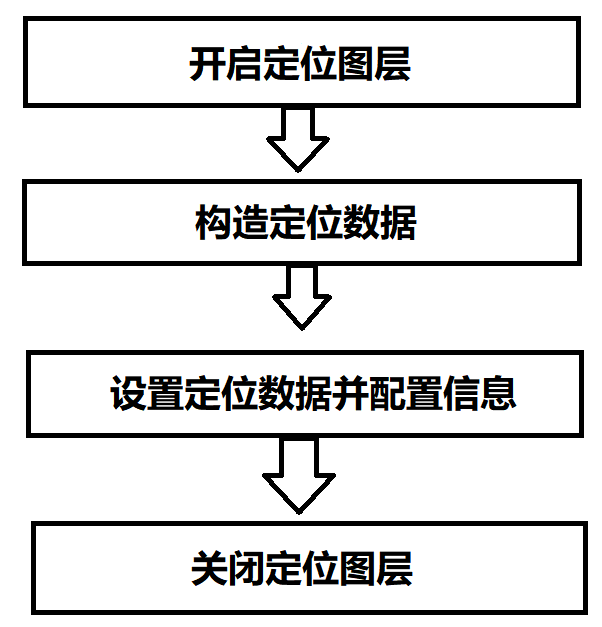
地图导入完成后，手机端显示的地图样式如图4-2所示。



*图4-2*

4.3 地图的定位

首先用 getSystemService方法获取系统的LocationManager对象，然后用LocationUpdates方法注册监听器，用于实现每隔一秒钟获取一次定位信息，如果location不为空，获取用户当前的经纬度，否则输出没有获取到GPS信息。接下来需要构造定位数据，设置方向信息和经纬度坐标。然后需要设置定位数据和模式，并在地图生命周期的onStart方法中开启定位图层，onStop方法中停止定位图层。其大致过程如图4-3所示。



*图4-3*

①首先用 getSystemService方法获取系统的LocationManager对象。

②然后调用LocationManager对象的LocationUpdates方法注册监听器，用于实现每隔一秒钟获取一次定位信息并将最新的定位信息传递给创建的locationUpdates方法当中，如果location定位信息不为空，则保存到LatLng对象中（获取用户当前的经纬度），之后通过日志输出经纬度的值；若location定位信息为空，则通过日志输出没有获取到GPS信息。

③接下来需要构造定位数据，设置方向信息（direction方法）、经度坐标（getLongitude方法获取）和纬度坐标（getLatitude方法获取）。

④然后需要设置定位数据和模式，并在地图生命周期的onStart方法中开启定位图层，onStop方法中停止定位图层。

4.4 定位请求

使用HttpClient进行Get通信，发送HTTP Get请求，服务器收到请求后，传回Json数据，再通过对Json数据的解析，得到时间以及经纬度。如图4-4。

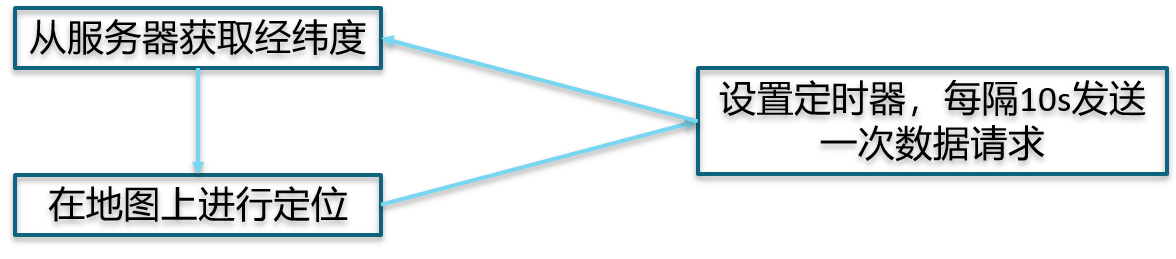


*图4-4*

通过readNet（）方法将服务器的地址（http:// 39.97.235.93:80）以及请求数据的参数（“00”），进行Get通信。

服务器接收到合法的请求后，给APP返回数据，这时通过 EntityUtils.toString（）方法，接收服务器传来的数据并转化为字符串形式。再通过 JSON.parse（）方法将Json数据解析。

4.5 目标追踪

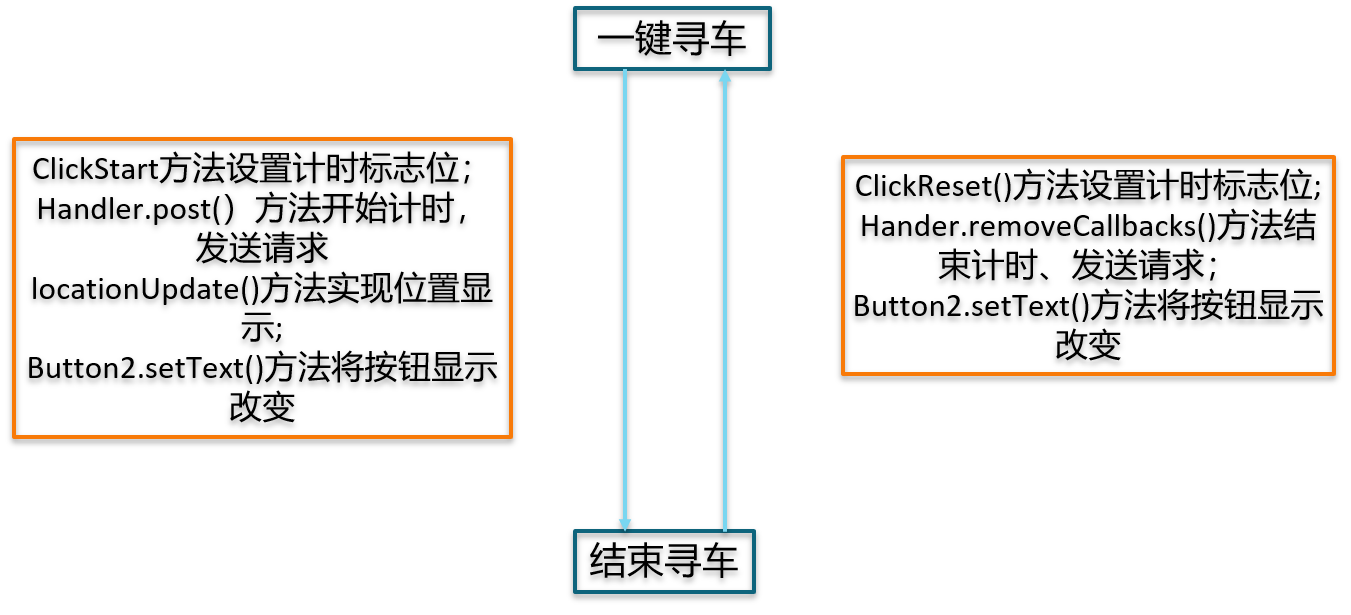


*图4-5*

追踪目标功能是在请求定位的基础上实现的。

首先通过locationUpdate（）方法把从服务器获取的经纬度在地图上进行定位。

设置一个定时器，延时10s,向服务器发送HTTP请求，不断的请求数据。



*图4-6*

如图4-6所示，设置一个按钮，并添加监听事件。当跳转到这个定位界面时，这个按钮显示为“一键寻车”，点击按钮，通过ClickStart()方法设置计数标志位running，调用handler.post()方法，使计时器开始计数，并发送HTTP Get请求，收到数据后通过locationUpdate（）方法实现位置显示，同时使用button2.setText()方法将按钮的显示由“一键寻车”改为“结束寻车”。此时APP就会根据服务器传来的不同的经纬度，进行位置显示，根据地图上的位置显示，从而对设备进行追踪。当点击“结束寻车”时，通过调用ClickReset()方法，将计数标志位running设置为false，调用handler.removeCallbacks()方法结束计时，同时使用button2.setText（）方法，将按钮的显示由“结束寻车”改为“一键寻车”。在点击“一键寻车”后，用户不可以在进行其他操作，如果点击其他按钮，APP就会出现一个警告的提示框。

4.6 阶段成果

4.6.1启动界面



*图4-7*

4.6.2 进入主界面



*图4-8*

4.6.3寻车页面

百度地图导入APP后默认定位到北京天安门（经度116.38，纬度39.90）。如图5-1地图导入定位图。



*图4-9*

4.6.4 用户须知



*图4-10*

4.6.5 警告图

当一键寻车时首先应该结束寻车才能获取现在的位置，当寻车时获取位置时会弹出警告图，如图4-11。



*图4-11警告图*

4.6.6 定位到我的位置

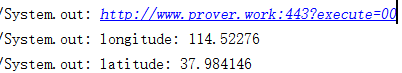
当点击获取现在的位置时，自动定位到用户当前的位置，如图4-12。



*图4-12*

4.6.7 定位请求

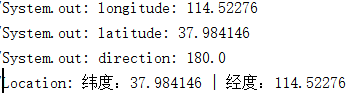
对HTTP请求内容和服务器返回内容的输出，如图5-12所示。



***图4-13.对HTTP请求内容和服务器返回内容的输出***

4.6.8 追踪目标

（1）从服务器中获取的经纬度，以及地图上定位的经纬度，如图4-13所示。



***4-13.从服务器中获取的经纬度，以及地图上定位的经纬度***

（2）加上定时器后从服务器连续的获取经纬度，如图4-14,4-15所示。



***图4-14.连续获取经纬度（1）***



***图4-15.连续的获取经纬度（2）***

（3）APP地图上位置信息的变化，如图所示。

*图4-16.位置1*  *图4-17.位置2*

** **

*图4-18.位置3**图4-19.位置4*

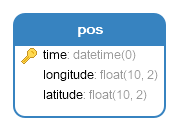
五、服务器设计

5.1 开发说明

* 服务器采用阿里云ECS服务器，操作系统为ubuntu16.04
* 开发工具为pycharm、Xshell
* Python开发环境为python3.8.0
* 依赖件为flask、pymysql、nwsgi
* 日志文件：flask.log，uwsgi.log

5.2 数据库

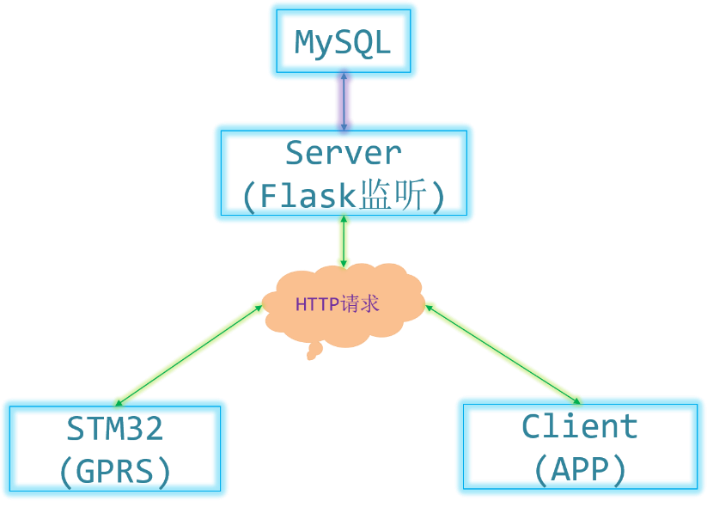
数据库采用MySQL，将时间、经度、纬度信息作为一条记录存储到远程服务器中，便于硬件端向该数据库存储数据，APP端从数据库中获取数据。其设计内容如图5-1所示。该数据库以时间为索引，每次服务器从里面获取数据时，根据时间排序，获取最最近一次的定位数据。APP端未设置历史记录删除功能，数据库数据由开发人员定期清除。除此之外，time这个字段，是由服务器自己获取并添加的，并不是由硬件端发送的，这样可以减轻硬件端功耗。



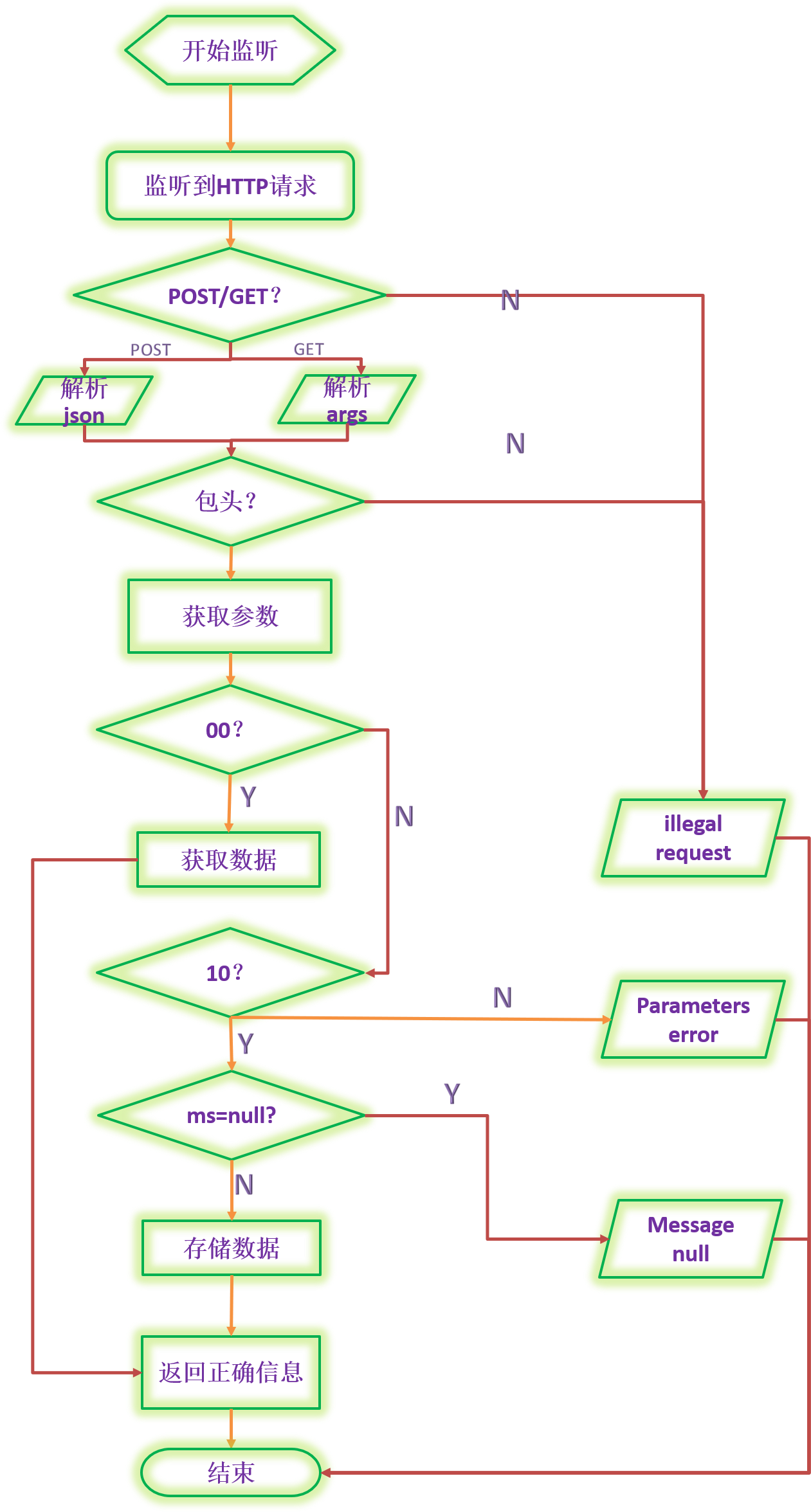
*图5-1*

5.3 服务器

服务器采用阿里云ECS服务器，操作系统为ubuntu16.04。其整体框架如图6-2所示。服务器端采用Flask框架监听硬件端和APP端的GET请求，并根据请求信息返回正确数据，从而完成APP端和硬件端的通信。主要通过flask实现功能，uwsgid使flask项目在服务器后台运行,并分别对数据库操作和服务器范围操作惊进行日志监控。其流程图如图5-3所示。



*图5-2*



*图5-3*

5.4 通信说明

完成上述设计后，在接口测试网站<https://getman.cn/>进行通信测试，并根据硬件端和APP端的功能，设计了一套通信协议，一方面为了正确处理数据，另一方面也是为了防止其他IP恶意攻击该web服务，造成服务器瘫痪。其通信规则如图5-4所示。



*图5-4*

六、定位系统设计

6.1 A9G模块

核心模块采用安信可科技的A9G模块，集GPS、GPRS为一体，物联网卡上网通信，节省空间占比，功能完善，是本系统核心模块的不二选择。利用AT指令来调用GPRS部分，确定链接网络能够获得GPS的返回值。同时使用AT指令来对GPS进行设置。

6.2 程序设计

6.2.1 主函数设计

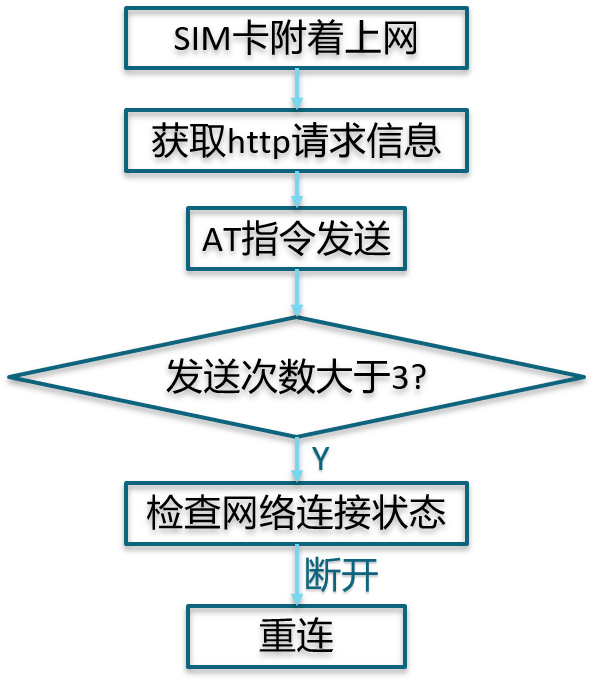
初始化数据，设置中断，电脑测试，AT通信，GPS通信。进入主循环，判断STA高位是否为1，如果接受到了(为1)，首先得到数据长度，自动添加结束符，由串口3发送，数组发送生变化（即地理发生变化），数据发送给GPRS,分析字符串获取并发送信息，同时开启下一次接收。



*图6-1*

6.2.2 GPRS设计

发送设备号，串口显示将要发送的信号。如果接受到了服务返回的信息，结束通信。如果没有接受到服务返回的信息，显示提示信息，显示网络中断。延延时1s，系统退出GET请求，检测已退出GET请求后，结束延时等待。LED闪烁2秒一次，表示服务器断开；正常状况LED的闪烁情况小于2秒一次。使用AT指令“AT+CGACT=1,1”,如果返回失败，则表示SIM链接失败，链接SIM，反之则是网络链接失败。

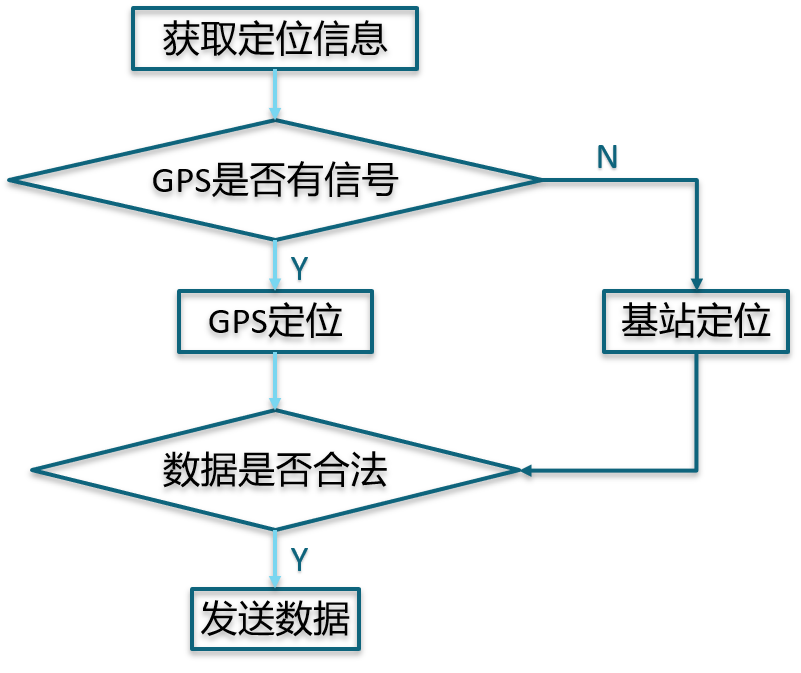


*图6-2*

6.2.3 GPS数据获取

获取GPS定位消息，获得经度，维度和航向的字符串，打印要发送的定位的信息(用GPGSA解析定位类型GPRMC解析经纬度GPVTG解析航向)。

向服务器发送GPS定位信息，判断是否连上GPS，如果没连上显示GPS无信号。如果连上判断信息，使用HTTPGET获得信息，或者信息错误。



*图6-3*

6.3 阶段成果

6.3.1 数据获取&通信

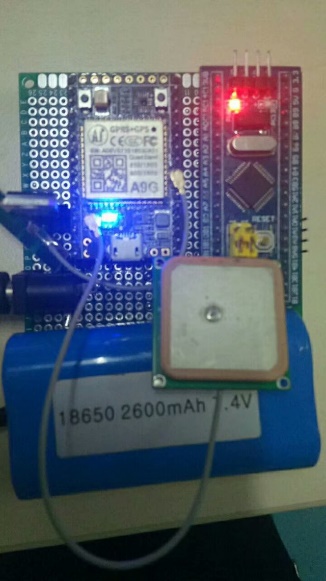
编写基础程序，获取GPS数据，并将获取的数据发送至服务器，成果如图6-1所示。



*图6-1*

6.3.2 电路设计

完成功能之后，进行电路焊接，成果如图6-2所示。

**

*图6-2*

6.3.3 成品完善

设计外壳，纠正问题，完善硬件，成果如图6-3所示。

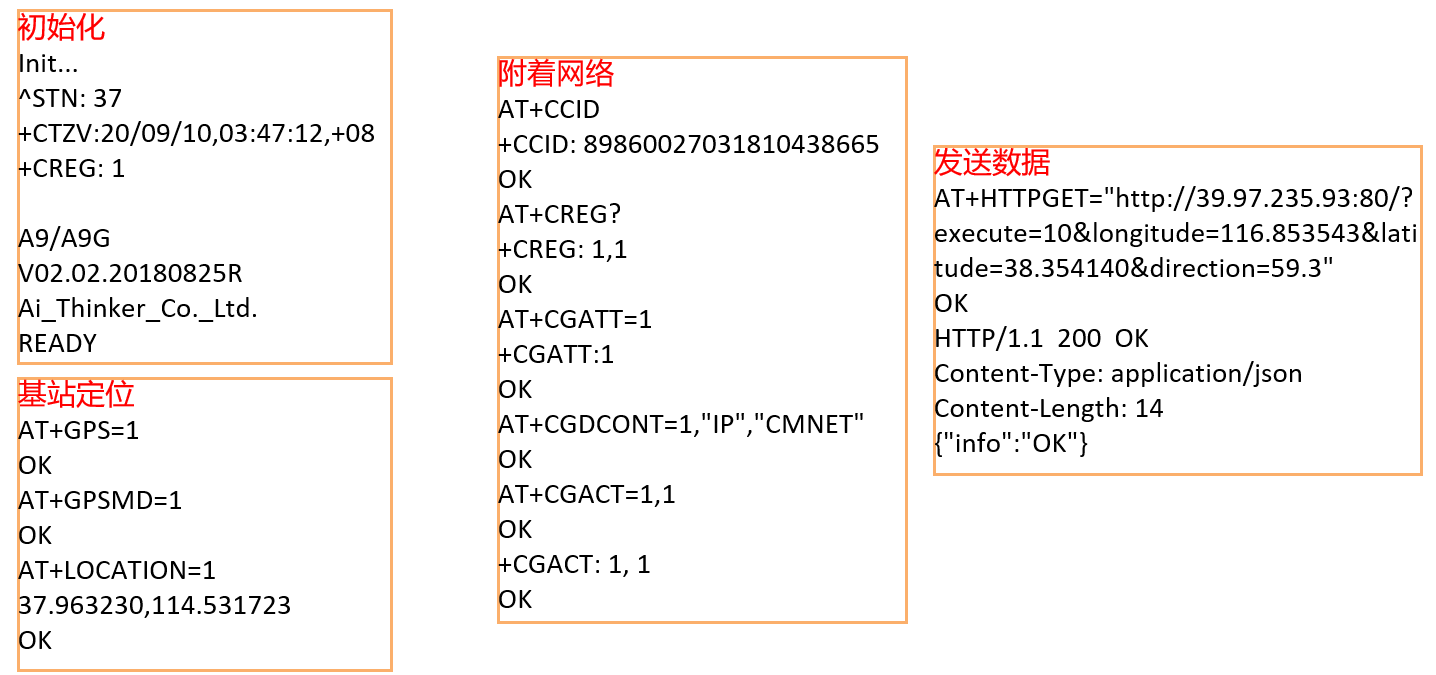


*图6-3*

七、项目测试

7.1 硬件测试结果

7.1.1 A9G输出信息



*图7-1*

7.1.2 STM输出信息



*图7-2*

7.2 软件测试结果

7.2.1 一键寻车

测试数据如图7-1所示



*图7-1*

7.2.2获取当前位置

|  |  |  |
| --- | --- | --- |
| IMG_256 | | |
| 位置一 | 纬度：39.48 | IMG_256 |
| 经度：115.85 |
| 位置二 | 纬度：39.95 | F6E477C5E3D9936D5D1459DEE0DD1CA2 |
| 经度：116.83 |

八、使用说明

8.1 硬件使用说明

|  |  |  |  |
| --- | --- | --- | --- |
| ***硬件使用说明*** | | | |
| 工作参数 | | | |
| 工作环境 | 最小值 | 典型值 | 最大值 |
| 工作温度 | -20 | 25 | 75 |
| 存储温度 | -30 | 25 | 85 |
| 供电电压 | 3.5 | 4.0 | 4.2 |
| LED指示灯 | | | |
| STM32-LED13 | 系统状态指示灯 | 闪烁 | 系统运行正常 |
| A9G\_GPS-LED1 | GPS状态指示灯 | 间隔1s闪烁一次 | 定位正常 |
| A9G\_GPRS-LED2 | GPRS状态指示灯 | 间隔1s闪烁2次 | 连接服务器 |
| 其他 | | | |
| GPRS通信 | SIM卡/物联网卡(2.5G) | | |
| 充电方式 | 锂电池充电器220V交流充电 | | |

*表9-1 硬件使用说明*

8.2 软件使用说明

|  |  |
| --- | --- |
| ***APP使用说明*** | |
| 主页面按钮 | |
| 开始寻车 | 跳转到功能页面 |
| 用户须知 | 跳转到用户须知界面 |
| 功能页面按钮 | |
| 一键寻车 | 根据车辆位置信息自动引导寻找车辆 |
| 获取现在位置 | 返回用户当前的位置信息 |
| 其他 | |
| 注意事项 | 寻车时若想获取用户的当前位置需要结束寻车 |
| 请使用者谨遵用户须知说明 |

*表9-2 APP使用说明*