

## 基于 ARM-LINUX 的 GPS 导航系统

朱衡

(物理与电子信息学院 应用电子技术教育专业 06 级 指导老师: 周永宏)

**摘要 :** 本系统由 GPS 数据采集并显示、地图显示两大部分组成, 模拟 GPS 导航的过程。通过 GPS 模块、arm9 硬件环境、嵌入式 LINUX 操作系统、TFT 触摸屏等模块来完成整个体系的运作, 使其实时采集 GPS 数据, 然后将 GPS 数据对应的地图显示出来, 再利用串口将数据送入 PC 机数据库中实现查询等一系列功能。本设计的特点在于界面友好、用户易操作、功耗低、便于长时间户外导航。

**关键词 :** ARM;LINUX;GPS;C++语言;SHELL 编程;嵌入式操作系统

### The ARM-LINUX-based GPS navigation system

Zhu Heng

(Institute: Institute of Physics and Electronic Information Professional: Applied Electronic Technology Education

Year: 06 Instructor: ZHOU Yong-hong)

**Abstract:** This system consists of GPS data collection, map display composed of two parts, simulated GPS navigation process. Through the GPS module, arm9 hardware environment, embedded LINUX operating system, TFT touch screen and other modules to complete the operation of the entire system to collect real-time GPS data, GPS data and then display the corresponding map, re-use the data into the PC serial port machine in the database queries, and so to achieve a series of functions. This design feature is user-friendly, user easy to

operate, low power consumption, easy to navigate a long time outdoors.

**Keywords:** ARM; LINUX; GPS; C ++ language; SHELL programming; embedded operating system

## 0 引言

GPS 是英文 Global Positioning System（全球定位系统）的简称，而其中文简称为“球位系”。GPS 是 20 世纪 70 年代由美国陆海空三军联合研制的新一代空间卫星导航定位系统。其主要目的是为陆、海、空三大领域提供实时、全天候和全球性的导航服务，并用于情报收集、核爆监测和应急通讯等一些军事目的。随着人民生活水平的法杖，GPS 技术被越来越多的应用在个人 PDA、个人车载终端、手机等个人设备上。人们通过手持 GPS，能准确知道自己所在的位置，从实现到导航、确定旅游路线、获取地理信息等功能。本文介绍的 GPS 导航系统，以 ARM 作为主控芯片，配以 GPS、TFT 触摸屏、嵌入式 LINUX 操作系统，构建了一个集 GPS 信息显示、地图显示、语音导航为一体的完整系统。本系统在一定程度上推动了个人手持 GPS 导航设备研究的发展。

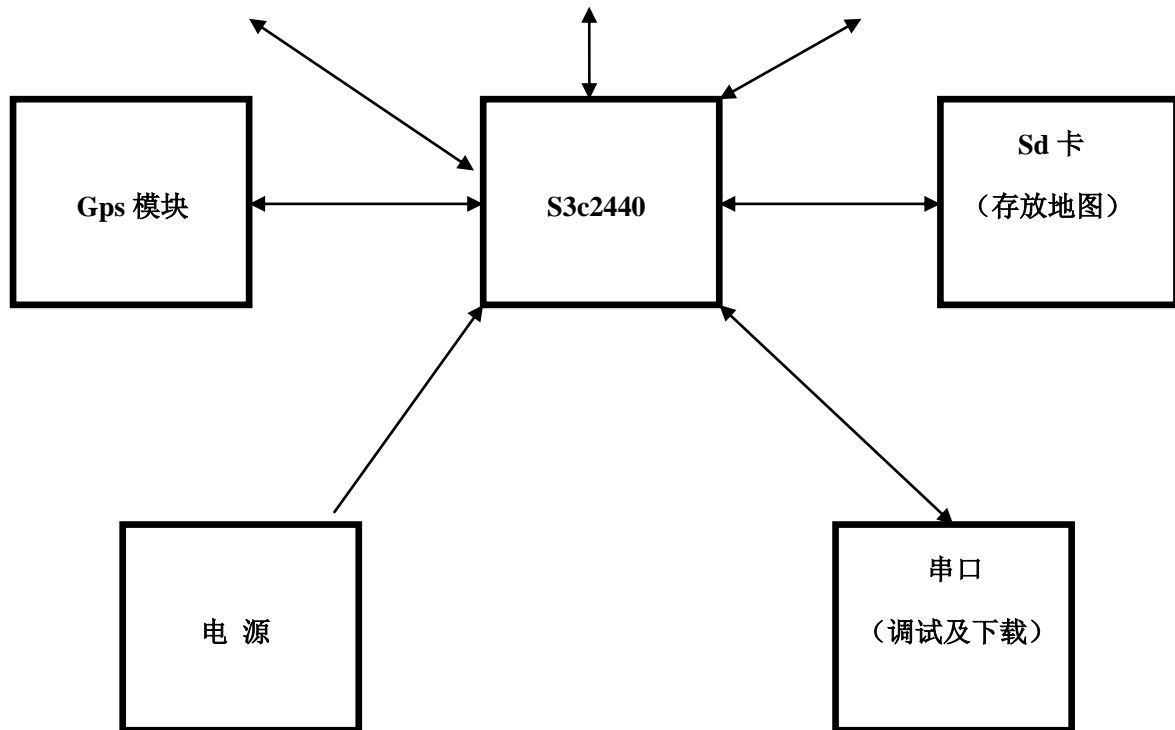
## 1 系统方案选择和论证

### 1.1 总体设计方案

个人手持设备要求界面美观、功耗低、易于操作。系统应有彩色触摸屏；低功耗、高速度的处理芯片；带有嵌入式操作系统；GPS 信息接收模块；sd 卡用以存放地图；

模块框图分别如图 1 所示：





方案一：采用凌阳公司的 16 位单片机，它是 16 位控制器，具有体积小、驱动能力高、集成度高、易扩展、可靠性高、功耗低、结构简单、中断处理能力强等特点。用于语音处理和识别等领域。满足系统 GPS 模块和语音导航功能的需要。由于本系统需要彩色人机交互界面，对处理速度要求较高，故不采用。

方案二：采用 samsung 公司的 s3c2440。S3c2440 采用 ARM920t 内核，实现了 MMU、ARBA、BUS 和 Harvard 高速缓冲体系结构。是一块高性能、低功耗的嵌入式芯片。可植入操作系统。在个人 PDA 领域应用及广。植入操作系统后，在用户 GUI 编程方面具有很大优势。内置 AD/DA，便于扩展语音模块。故选用此芯片为主控芯片。

### 1.2.2 操作系统选择方案

Linux 操作系统广泛应用于嵌入式设备中，其源代码开放、免费；内核性能高效、多任务、可制定；完善的图形管理、文件管理机制；良好的开发环境、强大的工具集。可移植 QT 开发环境，可实现跨平台的编译，代码重用率高。

### 1.2.3 GPS 模块选择方案

方案一：采用串口接口 GPS 模块。优点：便于驱动，价格便宜。缺点：嵌入式 linux 的调试需要用到串口，若 GPS 模块占用了串口会造成调试不便。故不采用

方案二：采用 usb 接口 GPS 模块。优点：便于携带、连接开发板。缺点：驱动相对较难、价格相对较贵。为了便于调试和携带，采用 usb 接口 GPS 模块。

### 1.2.4 显示模块设计方案

方案一：采用 LCD 显示。液晶显示屏（LCD）具有轻薄短小，低耗电量。但不能实现触摸功能，故部采用。

方案二：采用 TFT 显示。TFT 屏幕带有四线式电阻触摸膜，能实现触摸功能、便于用户操作，故采用。

### 1.2.5 地图文件存放方案

方案一：采用 s3c2440 核心板内置 nand flash 存放。Nand flash 读取速度快。但容量较小，只有 128M, 不适合存放大量地图数据，故不采用。

方案二：采用外置 sd 卡。Sd 卡读取速度相对较慢，且会增加成本。但能存放大量数据（1 G ~ 8 G），地图更新升级也很方便。故采用。

## 1.3 总体方案确定

- |                |                      |
|----------------|----------------------|
| 1) 控制芯片：       | 采用 s3c2440 arm9 处理器  |
| 2) 液晶显示模块：     | 采用 NEC TFT 3.5 寸屏幕   |
| 3) GPS 模块：     | 采用三星公司的 GPS-USBII 模块 |
| 4) nand flash： | 采用 K9F1208；          |
| 5) nor flash：  | 采用 AM29LV160DB       |
| 5) 音频处理芯片：     | UDA1341TS；           |
| 6) SD 卡 模块：    | 采用金士顿公司的 4GB SD 卡；   |
| 7) 电源模块：       | 采用两个 8V 可充电锂电池；      |
| 8) 操作系统：       | 采用嵌入式 linux 操作系统     |



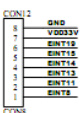


图 2.1 主控芯片电路

Fig.2.1 Circuit of ultrasonic transmitting and receiving

## 2.2 flash 芯片电路

### 2.2. 1 nand flash 芯片电路

Nand flash 是操作系统的程序存储区，该芯片掉电不丢失数据、寿命长，可反复擦写。芯片容量 128M。适合存放高密度数据结构

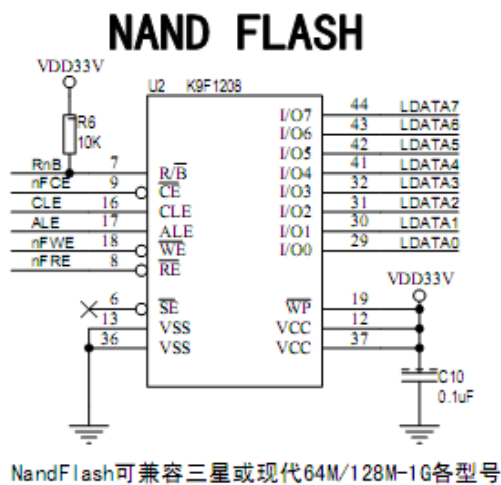


图 2.2.1 主控芯片电路

Fig.2.2.1 Circuit of ultrasonic transmitting and receiving

### 2.2. 2 nor flash 芯片电路

Nor flash 是系统 BIOS 存储区。该芯片掉电不丢失数据、寿命长，可反复擦写。芯片容量 8M。适合高速度、低数据密度的数据结构。

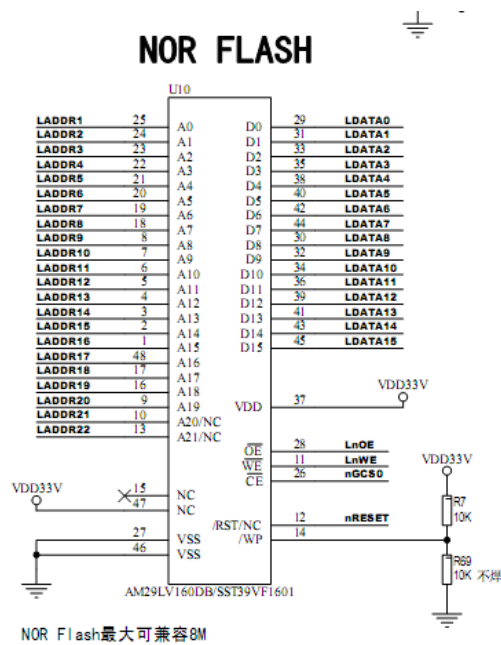


图 2.2.2 主控芯片电路

Fig.2.2.2 nor flash chip circuitry

## 2.3 串口电路

采用 max3232 电平转换芯片与计算机通信

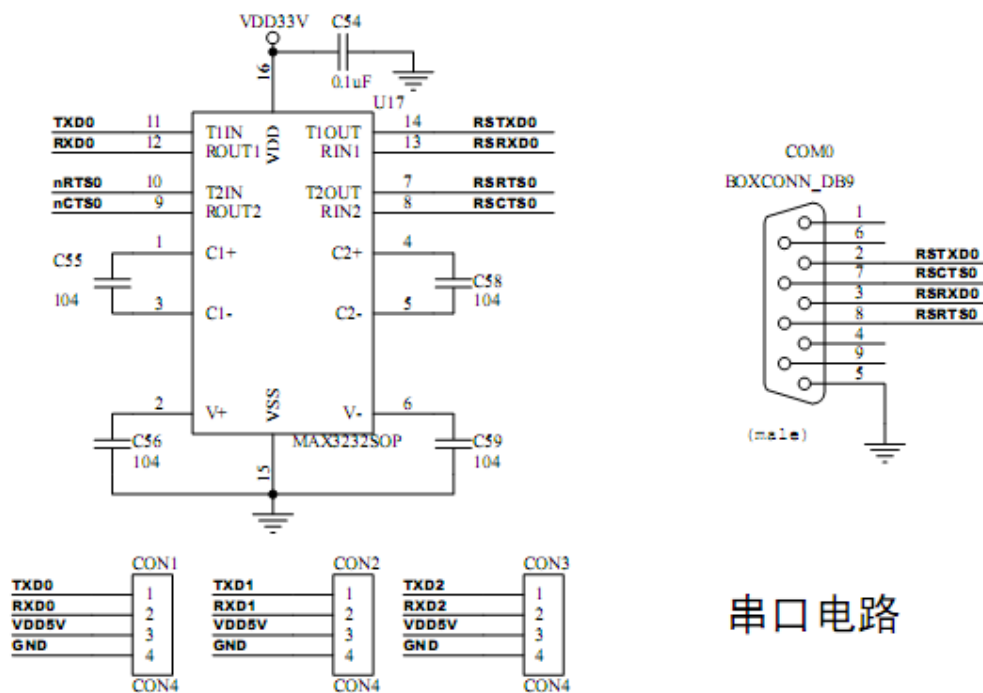




图 2.3 串口电路

Fig.2.3 Serial circuit

## 2.4 usb 端口电路

采用 usb2.0 协议，usb 接口用以连接 gps 模块

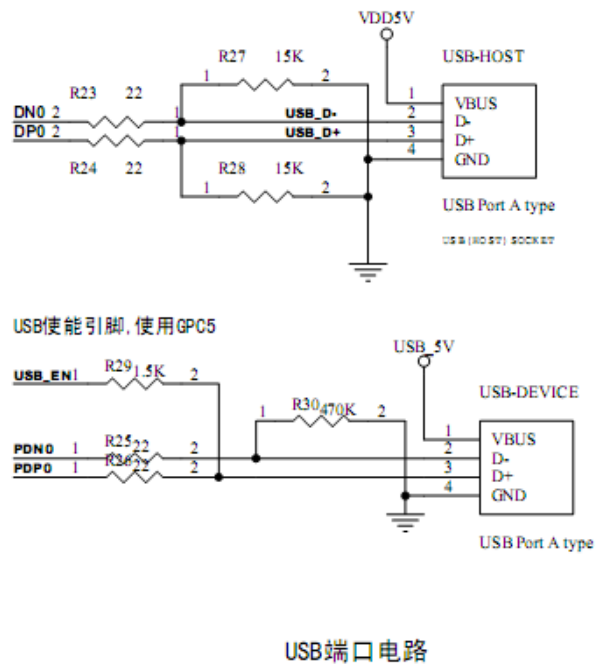


图 2.4 usb 端口电路

Fig.2.4 Driving circuit

## 2.5 系统总线及 lcd 接口电路

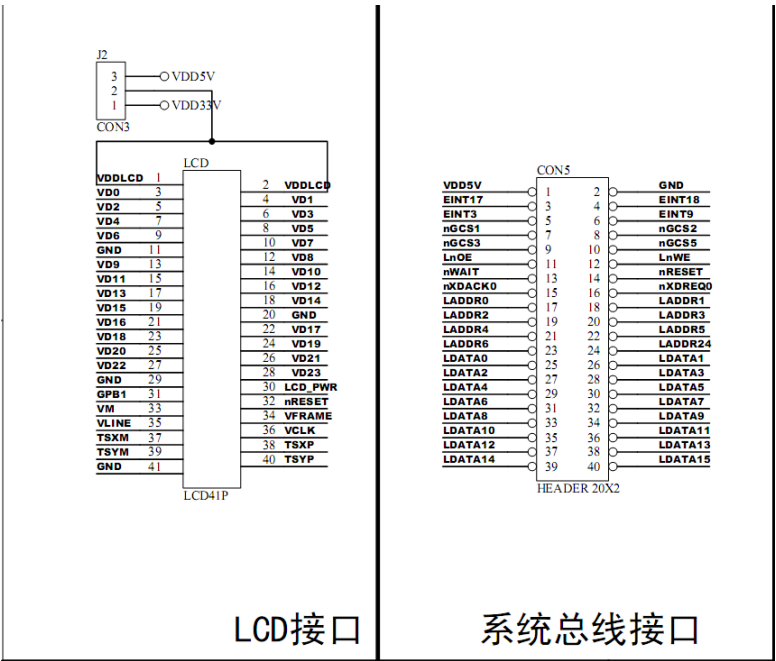


图 2.5 系统总线及 lcd 接口电路

Fig.2.5 Driving circuit

## 2.6 电源电路

系统输入电压为 DC5VC，通过 LM1117-1.8、LM1117-33、MAX8869EU18 芯片转换成系统需要的三种电压：1.8V、3.3V、1.25V

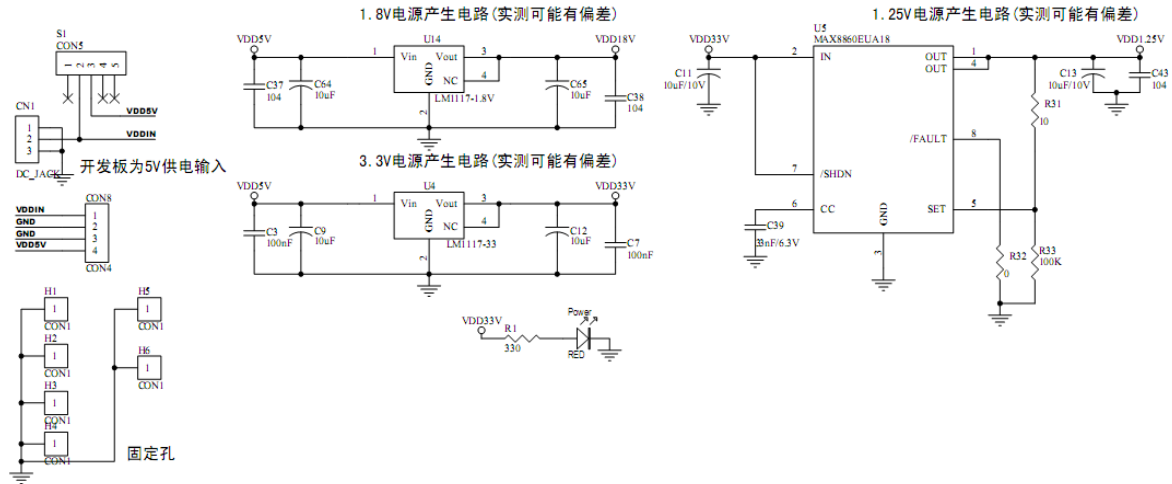


图 2.6 电源电路

Fig.6 Power circuit

### 3 系统的软件设计

本系统的所有程序均采用 C++ 语言编写，开发工具为 QT embed 2.0。程序定义了许多类来分别管理各个模块，通过对各个类的操作，来构建整个程序框架。

### 3.1 程序整体结构框图

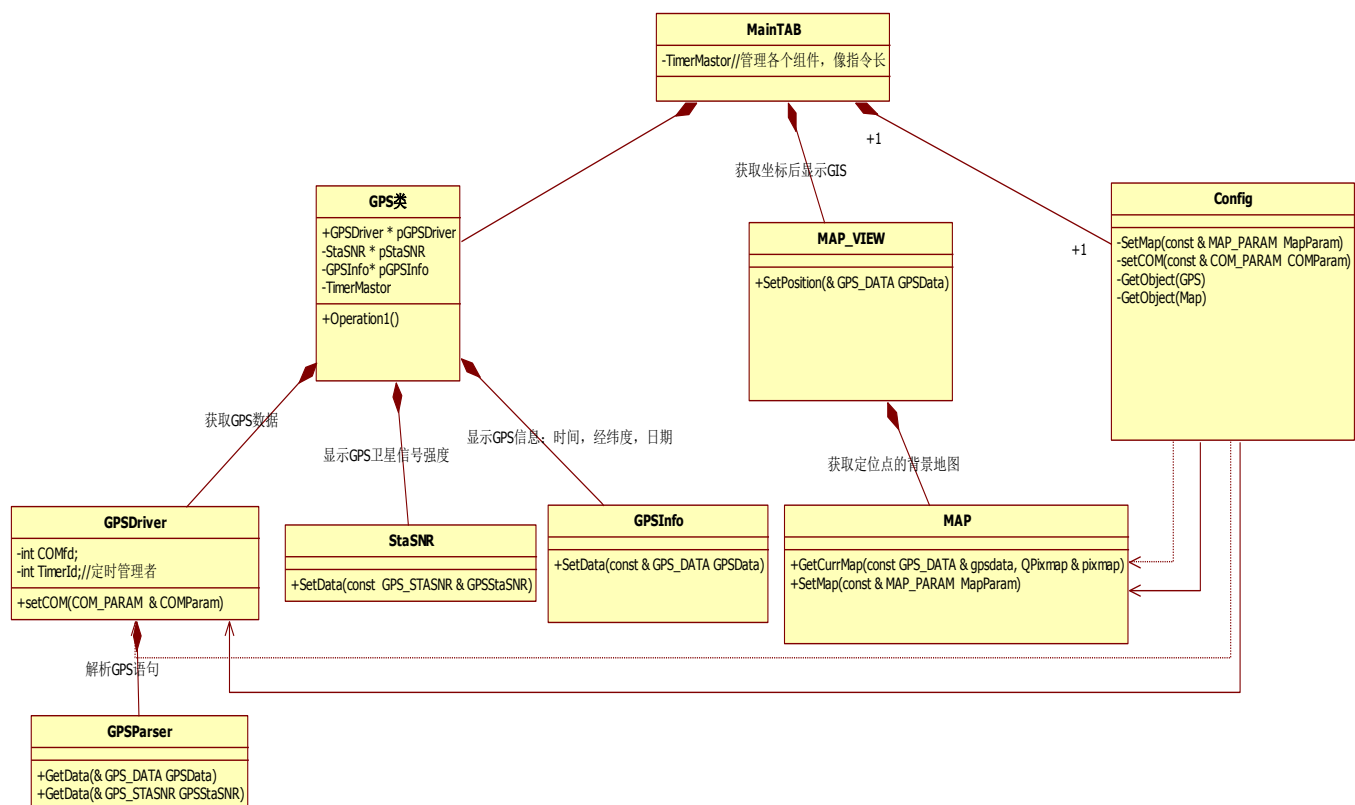
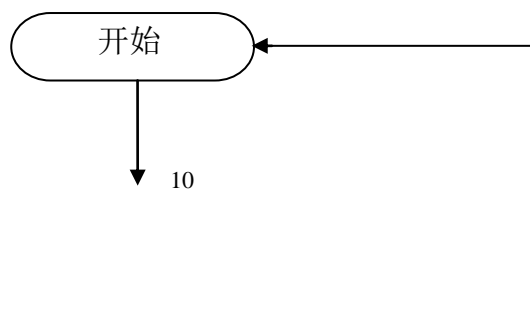


图 3.1 maintab 类流程图

Fig.3.1 Diagram of the overall structure of program

### 3.2 Maintab 类

## Maintab 类负责程序外框基本定义及程序初始化



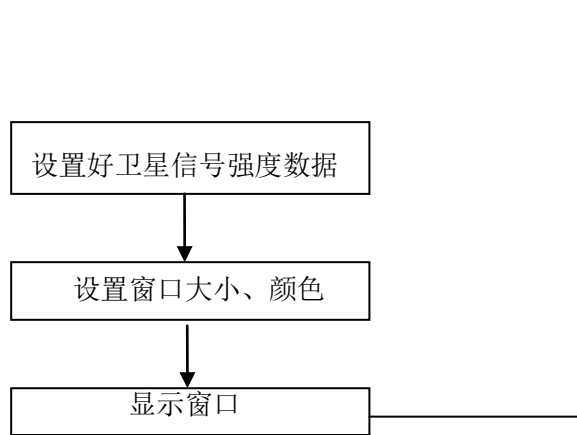


图 3.2 maintab 类流程图

Fig.3 .2 maintab Class Flowchart

### 3.3 mainwidgt 类

Mainwidgt 类负责对各按钮部件、页面部件的初始化

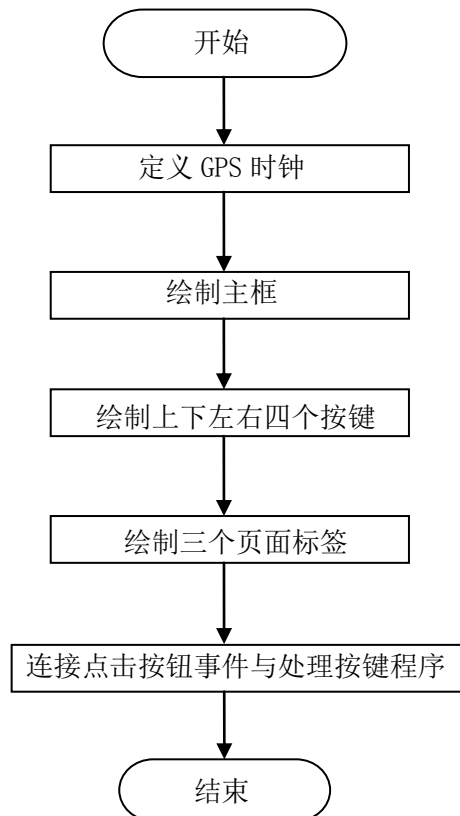


图 3.3 mainwidget 类流程图

Fig.3.3 mainwidget class Flowchart

### 3.4 config 类流程图

Config 类用以实现对地图路径、gps 模块波特率的设置

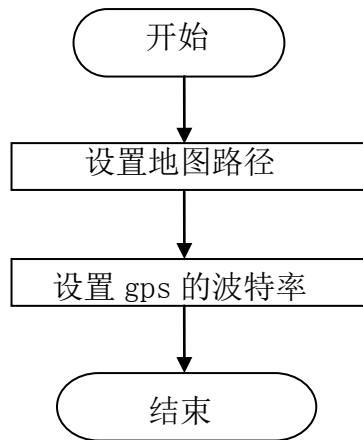


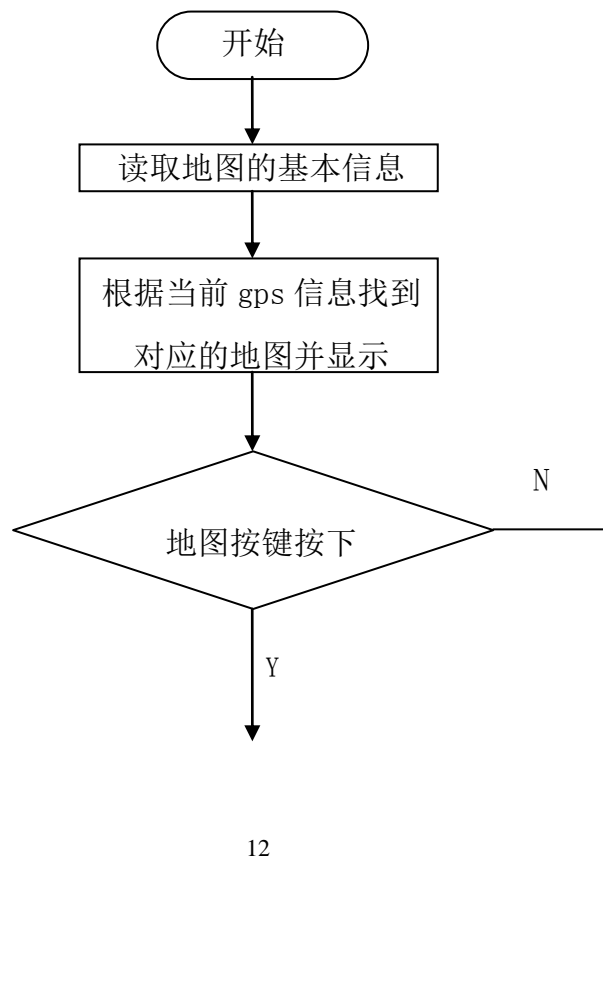
图 3.4 config 类流程图

Fig.3.4 config class Flowchart

### 3.5 map 类

Map 类负责读取 mapinfo 文件里的地图，并显示对应地图。

地图显示算法：首先把地图分割成  $n*n$  小块。然后系统通过得知整张地图四角的坐标，计算出地图每个像素点所对应的坐标。最后根据读取的 gps 信息，找到对应的地图，并显示到 lcd 上。



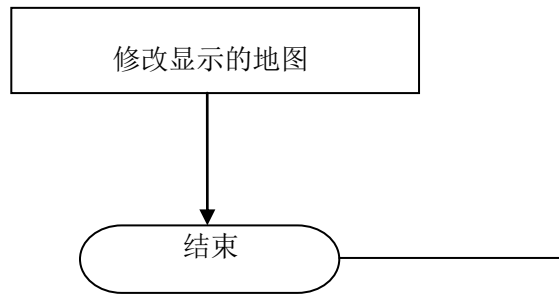


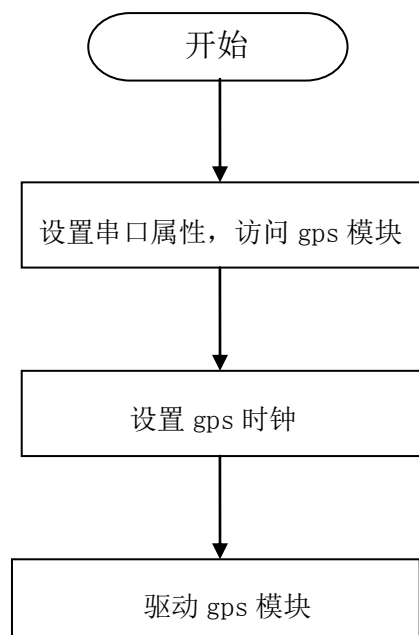
图 3.5 map 类流程图

Fig.3.5 map class Flowchart

### 3.6 gps\_diver 类

gps\_diver 类负责驱动 gps 模块，gps 驱动原理如下所述：

GPS 卫星发射两种频率的载波信号，即频率为 1575.42MHz 的 L1 载波和频率为 1227.60MHz 的 L2 载波。L1 为民用频率，L2 为军用。民用 GPS 信号调试成 C/A 码发送，C/A 码又被称为粗捕获码，是 1MHz 的伪随机噪声码（PRN 码），其码长为 1023 位（周期为 1ms）。用户机接受到 C/A 码后，通过 gps 接受器，解调为发送次 C/A 码的卫星时间。并通过多个卫星发送的 C/A 码时间差（至少 3 个），来确定当前位置。并以串口的形式把经纬度数据发送给上位机。



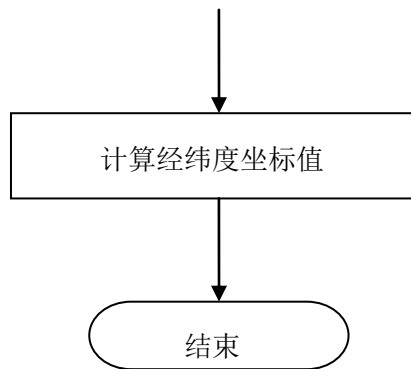


图 3.6 gps\_driver 类流程图

Fig.3.6 gps\_driver class Flowchart

### 3.7 staSNR 类

StaSnR 类负责对检测卫星数量，并告知 gps\_driver 类。

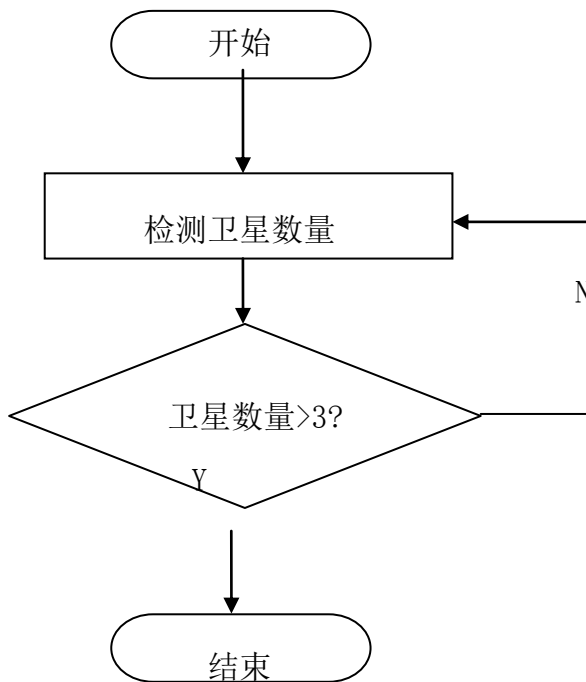


图 3.7 strSNR 类流程图

Fig.3.7 strSNR class Flowchart

### 3.8 debugPDF 类

debugPDF 类负责提供调试信息，错误编号等内容。调试程序时，配合 linux 中 gdb 命令。能实现断点调试、逐步调试等功能并显示对应的错误信息。

定义如下:

```
#include <stdlib.h>
#include <stdio.h>
#ifndef DEBUG_PDF_H
#define DEBUG_PDF_H
//=====debug level for info trace bug
#define debug 1
#define POP3DEBUG 1
#ifdef POP3DEBUG
#define PDF(level, fmt, args...) \
if (debug >= level) printf("[%s:%d] " fmt, __PRETTY_FUNCTION__, \
__LINE__ , ## args)
#else
#define PDF(level, fmt, args...) do {} while(0)
#endif
```

0: 没有任何的显示!

1: 正常工作的, 能显示工作运行主流程 error exit fatal show it

2: 更进一显示信息

3: 完全跟踪!

### 3.9 Makefile 文件

Makefile 文件为整个工程的管理文件, 作用是连接各个头文件、c 文件, 最终编译成可执行文件。

由于使用 qt 开发环境, 程序可以在 x86 平台上与 arm 平台上分别运行。只需要修改 Makefile 文件中的编译期即可。

### 3.10 地图文件 (mapinfo.txt 文件)

地图文件由一大张成都市区地图分割成许多小地图。Map.cfg 文件里包含了整张地图信



息。mapinfo:103.831787,30.687817,104.314155,30.473525,1500,1200,300。分别代表整个地图的左上角经纬度、右下角经纬度、地图像素的长度、地图像素的宽度、每张地图的像素宽度。软件通过读取 map.cfg 的内容，来获取地图的基本信息。

地图软件获取方法：采用专业地图下载软件 UMD (universal maps download)，用户只需在此软件里输入想要地图的四角经纬度坐标。即可得到所需地图。保证 mapinfo.txtL 里的地图坐标与实际坐标一致。

## 4 测试

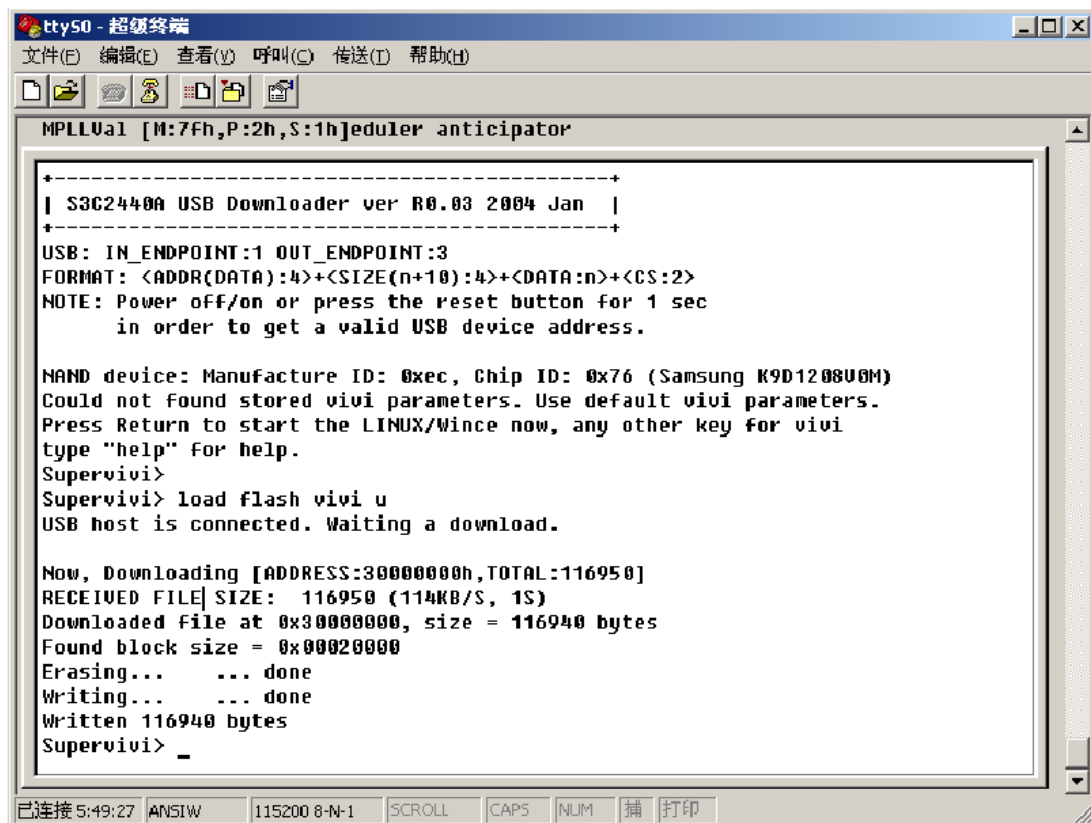
### 4.1 操作系统烧写测试

安装操作系统测试的目的是搭建系统的软件环境，便于后续工作

安装 linux 有如下步骤

1. 格式化 nand flash
2. 安装 bootloader
3. 安装内核文件
4. 安装文件系统

如下图所示



```
MPLLUa1 [M:7Fh,P:2h,S:1h]eduler anticipator

+-----+
| S3C2440A USB Downloader ver R0.03 2004 Jan |
+-----+
USB: IN_ENDPOINT:1 OUT_ENDPOINT:3
FORMAT: <ADDR(DATA):4>+<SIZE(n+10):4>+<DATA:n>+<CS:2>
NOTE: Power off/on or press the reset button for 1 sec
      in order to get a valid USB device address.

NAND device: Manufacture ID: 0xec, Chip ID: 0x76 (Samsung K9D1208U0M)
Could not found stored vivi parameters. Use default vivi parameters.
Press Return to start the LINUX/Wince now, any other key for vivi
type "help" for help.
Supervivi>
Supervivi> load flash vivi u
USB host is connected. Waiting a download.

Now, Downloading [ADDRESS:30000000h, TOTAL:116950]
RECEIVED FILE SIZE: 116950 (114KB/S, 1S)
Downloaded File at 0x30000000, size = 116940 bytes
Found block size = 0x00020000
Erasing...    ... done
Writing...    ... done
Written 116940 bytes
Supervivi> _
```

图 4.1.1 烧写操作系统--vivi 界面

Fig.4.1.1 Flow chart of program of cooling automatically

安装好操作系统后，重启系统，将出现如下画面。

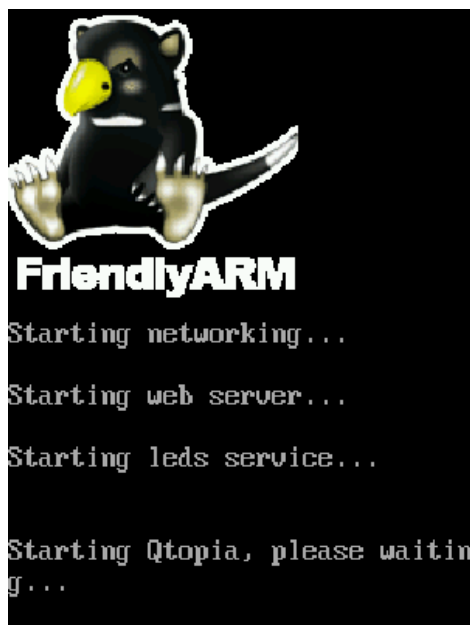


图 4.1.2 操作系统启动画面

Figure 4.1.2 operating system to start screen

系统烧写成功

## 4.2 卫星接收测试

卫星接收测试的目的是检测接收到卫星的数量，以便在接收到相同卫星的情况下，检测 gps 信号。测试位置：四川省成都市。

第一次测试：位置一室内 卫星数量 0 如下图：

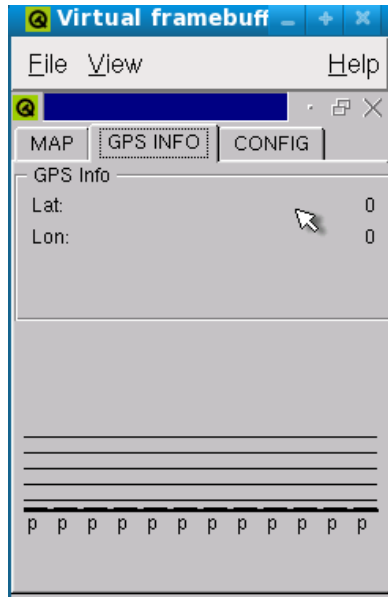


图 4.2.1 第一次卫星接收测试

Figure 4.2.1 The first test satellite reception

第二次测试：位置一室外空旷地带 卫星数量 7 如下图：

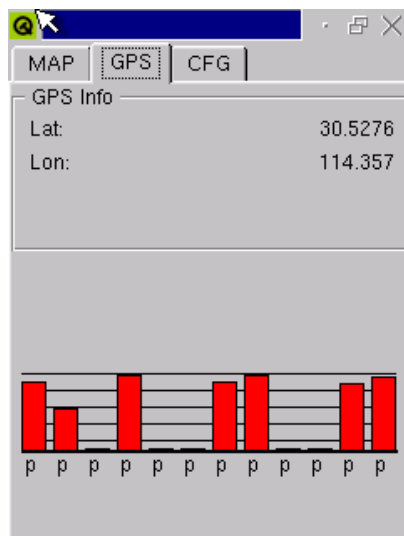


图 4.2.2 第二次卫星接收测试

Figure 4.2.2 The second test satellite reception

第三次测试：位置一街道 卫星数量 6 如下图：

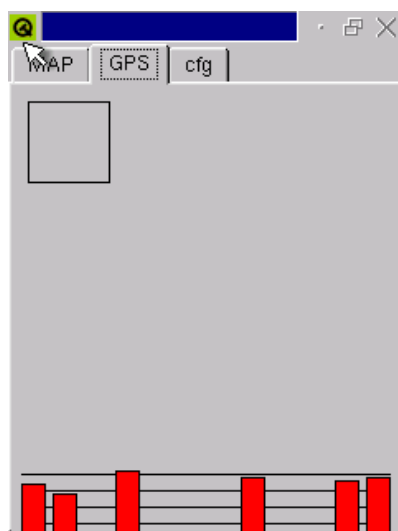


图 4.2.3 第三次卫星接收测试

Figure 4.2.2 The third test satellite reception

经测试，在室内不能接收到卫星的 gps 信号。故此系统只能在户外使用。

### 4.3 地图显示测试

地图显示测试的目的是测试载入地图的正误、验证地图上下左右移动的功能。

第一次测试： 能实现上下左右移动，显示地图如下图：



图 4.3.1 第一次地图显示测试

Figure 4.3.1 The first map shows test

第二次测试：能实现上下左右移动，显示地图如下图：



图 4.3.2 第二次地图显示测试

Figure 4.3.2 The second map shows test

第三次测试：能实现上下左右移动，显示地图如下图：



图 4.3.3 第三次地图显示测试

Figure 4.3.3 The third map shows test

经测试地图能准确的载入，并能实现上下左右移动的功能

#### 4.4 gps 信号检测测试

Gps 信号测试的目的是测试 gps 信号的精确度，测试数据列于表 3 中。

表 3 gps 数据测量表

测试次数（次）	第一次	第二次	第三次

显示经纬度	东经：30.797415 北纬：106.080517	东经：30.813544 北纬：106.09684	东经：30.804534 北纬：106.08542
实际经纬度	东经：30.797367 北纬：106.080235	东经：30.797415 北纬：106.09851	东经：30.797415 北纬：106.08846
误差计算	0.013%	0.012%	0.0011%

从上表可以看出，gps 数据精确度较高，误差都在 0.01%左右

#### 4.5 功耗测试

测试环境：户外

系统使用硬件：gps 模块、sd 卡、液晶屏

测试次数（次）	第一次	第二次	第三次
电流	300 毫安	350 毫安	320 毫安
电压	3.32V	3.31V	2.95V
功耗	0.99W	1.15W	0.94W

从上表可看出，系统功耗在 1w 左右。能满足设计技术指标。

#### 4.6 其他测试

1) 地图路径变更测试 。 功能实现，测试通过

2) gps 串口波特率设置。 功能实现，测试通过

#### 4.7 测试结果分析

由以上的测试数据可以看出：本 GPS 导航系统能很好地实现题目要求的各项功能。测试期间运行稳定，基本达到了设计要求。

### 5 总结

本系统成功的模拟了个人手持 gps 导航系统。用户通过点击触摸屏幕、方便的使用软件。实现了地图的显示、拖动；gps 信息、卫星数量的显示；地图路径的更改；gps 波特率的更改的功能。系统低功耗、高处理速度、操作简易、界面有好。已具有一定的实用价值。

通过这次毕业设计，我在更加牢固、灵活地掌握了所学的专业知识的同时。大量学习了嵌入式方面的相关知识：arm 体系结构、操作系统、c++编程、shell 编程、硬件驱动程序。

### 参考文献：

- [1] 杜春雷 ARM 体系结构与编程 清华大学出版社 2003-08-12
- [2] c++语言程序设计（第三版）郑莉 董渊 张瑞丰 编著 清华大学出版社 2001-10-11
- [3] 嵌入式系统开发 齐宇 徐俊 编著 人民邮电出版社 2005-06-05
- [4] [美]Douglas Boling 嵌入式 linux 编程 北京科彦科技发展公司 译 北京大学出版社 2003-03-12
- [5] 张洪斌 QT 编程指南 电子科技大学出版社 2006-06-12
- [6] 李先力 文苍茂 GPS 原理及应用清华大小而出版社 2004-08-27
- [8] 李新峰 何广生 赵秀文 基于 ARM9 的嵌入式 Linux 开发技术 电子工业出版社 2003-03-18
- [9]Thomas L.Floyd.Electronic Devices [M]. Prentice-Hall Inc, New jersey 1999-05-19

### 致谢

至此，我的毕业设计已经基本完成。回想过去的点点滴滴，如果没有学校、学院、老师、同学的关心和帮助，我也无法按期完成这令我满意的作品。

在此，我首先感谢学校和学院为我提供优越的实验场所和器材，使我能够更加方便、顺利地进行作品设计与调试，减少了设计周期；

其次，我要感谢指导老师周永宏老师在百忙之中抽出时间同我一起探讨系统构架和软件的编写，并不厌其烦地替我修改论文，给了我莫大的帮助，并使我学到了很多。周老师严谨负责的治学态度，给我留下了深刻的印象，将使我终身受益。我再次对周老



师这一年多来对我的关心和帮助表示最诚挚的谢意；

最后，向在整个设计过程中关心和帮助过我的其他老师和同学致谢。

## 附录：

### 附录 1：测试仪器

表 7 测试仪器设备清单

序号	型号、名称	用途	数量
1	PC 机	编译程序	1
2	UT33D 数字万用表	测量各电路工作情况	1
3	Gps 导航仪	测试显示的 gps 正误	1
4	vm 虚拟机	测试操作系统	1
5	成都、南充市区地图	测试显示地图正误	2
6	精密电流、电压表	测试系统功耗	2

### 附录 2：部分参考源程序

Maintab 类程序：

```
#include <qapplication.h>
#include <qcanvas.h>
#include <qfont.h>
#include <qimage.h>
#include <qpainter.h>
#include <qrect.h>

#include <qwidget.h>
#include <qimage.h>
#include <qpixmap.h>
#include <qpushbutton.h>
#include <qlabel.h>
#include <qdialog.h>
#include <qpainter.h>

#include <qapplication.h>
#include <qwidget.h>
#include <qglobal.h>
#include <qcolor.h>
```

```
#include <qlayout.h>
#include <stdlib.h>
#include <stdio.h>
#include <qpalette.h>
#include <qcolor.h>

//=====
#include "MainWidget.h"
#include "zhMap.h"
#include "gps_driver.h"
#include "maintab.h"
#include "GPS.h"

MainTab::MainTab(QWidget *parent, const char * name):QTabWidget(parent, name)
{

    GpMainLayout=new MainLayout;
    GpGPS=new GPS;
    GpConfig=new Config;
    GpConfig->GetObject(GpMainLayout);
    GpConfig->GetObject(GpGPS);
    GpConfig->setCOMBDRate();
    GpConfig->setMapPath();
    addTab(GpMainLayout, QString("MAP"));
    addTab(GpGPS, QString("GPS INFO"));
    addTab(GpConfig, QString("CONFIG"));
    timerid=startTimer(1000);
}

void MainTab::timerEvent(QTimerEvent * event) {
    GPS_DATA mGPSData;
    int i;
    PDF(3, "TABMaster timer alarm!\n");

    //=====设置好卫星信号强度数据!

    GpGPS->mpSatSNR->SetData(GpMainLayout->pzhDrawRect->pGpsdriver->mGPSStaSNR)
    ;
    //=====
    GpMainLayout->pzhDrawRect->pGpsdriver->GetData(mGPSData);
```

```
GpGPS->mpGPSinfo->SetData(mGPSData);
//=====
PDF(3,"TABMaster timer end\n");
}
```

```
int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    MainTab t;
    app.setMainWidget(&t);
    t.show();
    return app.exec();
}
```

Gps\_driver 类程序:

```
#include <string.h>
#include "gps_driver.h"
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
```

/\*NMEA 0183 的 Checksum 只能作 8 位异或包括", "的运算, 但不包括"\$" 及 "\*" 定义符号的运算。

最后的两位是检验码的 hex(16 进制)转成 ASCII 两个符号 (0-9, A-F)。

```
*/
char nmea_checksum(char *sentence)
/* is the checksum on the specified sentence good? */
{
    unsigned char sum = '\0';
    char c, *p = sentence, csum[2];
    c = *p;
    while (c != '*')
    { sum ^= c; p++; c=*p; }
    //=====
    csum[0]=sum&0x0f;
    if(csum[0]>=10) csum[0]+='A';
    else csum[0]+='0';
    //=====
    csum[1]=(sum&0xf0)>>4;
    if(csum[1]>=10) csum[1]+='A';
    else csum[1]+='0';
}
```

```
c= (csum[1]==toupper(p[1]) && csum[0]==toupper(p[2]) );
return c;// 0 is error
}

int zh_gpsDriver::fifoctrl(FifoType * priv ,int data ,const unsigned char flag)
{
    int temp = -1;
    unsigned int widx = 0;
    widx = priv->w_idx;    /* Save the old index before proceeding */
    if (flag == WR) //write data
    {
        /* Save it to buffer */
        if (((priv->w_idx + 1) % BUFSIZE) == priv->r_idx)
        {
            /* Adjust read index since buffer is full */
            /* 无法写入来处理 fifo 满。can wd to fifo */
            //priv->r_idx = (priv->r_idx + 1) % BUFSIZE;
            //保证是环状的 FIFO。
            PDF(2, "buffer full\n");
            return -1;
        }
        priv->buffer[priv->w_idx] = data;
        priv->w_idx = (priv->w_idx + 1) % BUFSIZE;
        //保证是环状的 FIFO。
        return 0;
    }
    else //read data
    {
        if ((priv->r_idx == widx))
        { PDF(2, "buffer empty\n");return -1;}
        else
        {
            temp= priv->buffer[priv->r_idx];
            priv->r_idx = (priv->r_idx + 1) % BUFSIZE;
            //保证是环状的 FIFO。
            return temp;
        }
    }
    PDF(2, "flag error\n");
    return -1;
}
```

```

zh_gpsDriver::zh_gpsDriver(QObject      *      parent, const      char      *
name):QObject(parent, name)
{

    COM_PARAM mCOMParam;
    mCOMParam.SerialDevicePath="/dev/ttyUSB0";
    mCOMParam.BaudRate=9600;
    id=startTimer(200);
    buf.pos=0;
    buf.sw=0;
    mpGPS_GPRMC=&mGPS_GPRMC;
    mpGPS_GPRMC->gps_sw='V';
    mpGPS_GPGSV=&mGPS_GPGSV;
    PDF(1,"gps driver construct!\n");
    /* Open modem device for reading and writing and not as controlling
    * tty. | O_NONBLOCK*/
    fd=-1;
    setCOM(mCOMParam);
}

int zh_gpsDriver::GPRS_GetTextLineFromFIFO(  CMD_BUF  * pcmd_buf)
{
    //      去掉换行符
    int j=0;
    int idata;
    do
    {
        idata=fifoctl(&fifo, idata, RD);
        if(idata==-1) {return 0;}
        else if(idata=='\n')
        {
            pcmd_buf->buf[pcmd_buf->pos]=(char)(idata&0x00ff);
            pcmd_buf->pos++;pcmd_buf->sw=1;
            break;
        }
        else
        {pcmd_buf->buf[pcmd_buf->pos]=idata;pcmd_buf->pos++;}
    }
    while(j++<BUFSIZE);
    /// PDF(4,"end of a line !\n");
    if((pcmd_buf->pos)>(BUFSIZE-5))
    {PDF(2,"potential fiflo overflow!\n");CMDBUF_Init(pcmd_buf);}
    PDF(2,"pos=%d\n",pcmd_buf->pos);
}

```

```

pcmd_buf->buf[pcmd_buf->pos]='\0';
// PDF(4, "pos=%d  ibufpos=%d  a msg=%s\n", pos, ibufpos, pStrBuffer);
return 1;
}

```

```

int  zh_gpsDriver::zh_getGPSdata(GPS_DATA & gpsdata)
{
    PDF(3, "zh_getGPSdata!\n");
    gpsdata.lati=mpGPS_GPRMC->latitude;
    gpsdata.lont=mpGPS_GPRMC->longitude ;
    return mpGPS_GPRMC->gps_sw;
}

```

```

double ddmm2dddd(double ddmm)
{
    double fi, ffz, ffx;
    int fz;
    fi=ddmm;
    fz=(int)fi;
    ffz=(double)fz;//取整数部分
    ffx=fi-ffz;//取小数部分
    fi=ffz+ffx/0.6;
    PDF(2, "ddmm=%lf  dddd:%lf\n" , ddmm, fi );
    return fi;
}

```

```

int  zh_gpsDriver::GPGSV_Analysis( GPS_GPGSV * pGPS_GPGSV)
{
    switch(pGPS_GPGSV->GSV_ID) {
        case
1:memcpy(&(mGPSStaSNR.Sta[0]), &(pGPS_GPGSV->Sta1), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[1]), &(pGPS_GPGSV->Sta2), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[2]), &(pGPS_GPGSV->Sta3), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[3]), &(pGPS_GPGSV->Sta4), sizeof(STA_INFO));
    break;
        case
2:memcpy(&(mGPSStaSNR.Sta[4]), &(pGPS_GPGSV->Sta1), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[5]), &(pGPS_GPGSV->Sta2), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[6]), &(pGPS_GPGSV->Sta3), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[7]), &(pGPS_GPGSV->Sta4), sizeof(STA_INFO));
    break;
        case
3:memcpy(&(mGPSStaSNR.Sta[8]), &(pGPS_GPGSV->Sta1), sizeof(STA_INFO));
    memcpy(&(mGPSStaSNR.Sta[9]), &(pGPS_GPGSV->Sta2), sizeof(STA_INFO));

```

```

memcpy(&(mGPSStaSNR.Sta[10]), &(pGPS_GPGSV->Sta3), sizeof(STA_INFO));
memcpy(&(mGPSStaSNR.Sta[11]), &(pGPS_GPGSV->Sta4), sizeof(STA_INFO));
break;
default:break;
}
}

int zh_gpsDriver::CMD_Analysis( CMD_BUF * pcmd_buf ){
char* pcmd;
int i;
if( pcmd_buf->sw ==1)
{
PDF(3, "sentence: %s\n end\n", pcmd_buf->buf);
pcmd=strstr(&pcmd_buf->buf[0], "$GPRMC");//只对此语句解析
if (pcmd!=NULL)
{
if (!nmea_checksum(&pcmd[1]))
{
PDF(2, "check sum error\n");
/*CMDBUF_Init(&buf);
return 1;*/
}

//2: 判断有效性。
i= sscanf(pcmd, "%*[^AV]%c", &mpGPS_GPRMC->gps_sw);
PDF(3, "gpsSwFiled=%c\n", mpGPS_GPRMC->gps_sw);

if(mpGPS_GPRMC->gps_sw=='V') { PDF(3, "V invalid data\n");
CMDBUF_Init(&buf); return 2; }

//sscanf 并不能由空值来输入浮点数。
// printf("Gpsid=%s\n utc_time=%s\n gps_sw=%c\n latitude=%lf\n
lati_ns=%c\n longitude=%lf\n longi_ew=%c\n spd=%lf",
//Gpsid, utc_time, gps_sw, latitude, lati_ns, longitude, longi_ew, spd);

i=
sscanf(pcmd, "%[^,],%[^,],%c,%lf,%c,%lf,%c,%lf,%lf,%[^,],%lf,%c,%s",
mpGPS_GPRMC->Gpsid, mpGPS_GPRMC->utc_time, &mpGPS_GPRMC->gps_sw,
&mpGPS_GPRMC->latitude, &mpGPS_GPRMC->lati_ns,
&mpGPS_GPRMC->longitude, &mpGPS_GPRMC->longi_ew, /*7*/

&mpGPS_GPRMC->speed, &mpGPS_GPRMC->angle, mpGPS_GPRMC->utc_date, &mpGPS_GPR
MC->magneDeclinat, &mpGPS_GPRMC->magneDeclinatAngle, mpGPS_GPRMC->sumCheck);

```

```
//请检查 sscanf 的返回值，可能会失败，如果小于要收得的数目。
if(i<13) //转换失败，
{
    PDF(2,"sscanf error\n");
    mpGPS_GPRMC->gps_sw='V';
CMDBUF_Init(&buf);return 3;}
//转换成度制，而不是度，分，十进制分形式。
mpGPS_GPRMC->latitude=ddmm2dddd((mpGPS_GPRMC->latitude/100));
mpGPS_GPRMC->longitude=ddmm2dddd((mpGPS_GPRMC->longitude/100));
PDF(3,"gps_sw=%c\n latitude=%lf\n lati_ns=%c\n",
mpGPS_GPRMC->gps_sw,mpGPS_GPRMC->latitude, mpGPS_GPRMC->lati_ns);
PDF(3,"longitude=%lf\n longi_ew=%c\n",
mpGPS_GPRMC->longitude,mpGPS_GPRMC->longi_ew);
goto out;
}
///=====
pcmd=strstr(&pcmd_buf->buf[0],"$GPGSV");//只对此语句解析
if (pcmd!=NULL)
{
    if (!nmea_checksum(&pcmd[1]))
    {
        PDF(2,"check sum error\n");
        /*CMDBUF_Init(&buf);
        return 1;*/
    }
    //sscanf 并不能由空值来输入浮点数。
    i=
    sscanf(pcmd,"%[^,],%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d",
        mpGPS_GPGSV->GPGSV,&mpGPS_GPGSV->GSV_NOB,&mpGPS_GPGSV->GSV_ID,
&mpGPS_GPGSV->StaNOB,
        &mpGPS_GPGSV->Sta1.StaID,&mpGPS_GPGSV->Sta1.StaA,&mpGPS_GPGSV->Sta1.StaB,
&mpGPS_GPGSV->Sta1.StaSNR,
        &mpGPS_GPGSV->Sta2.StaID,&mpGPS_GPGSV->Sta2.StaA,&mpGPS_GPGSV->Sta2.StaB,
&mpGPS_GPGSV->Sta2.StaSNR,
        &mpGPS_GPGSV->Sta3.StaID,&mpGPS_GPGSV->Sta3.StaA,&mpGPS_GPGSV->Sta3.StaB,
&mpGPS_GPGSV->Sta3.StaSNR,
        &mpGPS_GPGSV->Sta4.StaID,&mpGPS_GPGSV->Sta4.StaA,&mpGPS_GPGSV->Sta4.StaB,
&mpGPS_GPGSV->Sta4.StaSNR,
        mpGPS_GPGSV->sumCheck);
    //请检查 sscanf 的返回值，可能会失败，如果小于要收得的数目。
```



```
        GPGSV_Analysis(mpGPS_GPGSV); //存入数据中!
        i++;
        goto out;
    }

out:CMDBUF_Init(&buf);
}
return 0;
}

void    zh_gpsDriver::CMDBUF_Init(    CMD_BUF    * pcmd_buf )
{
    pcmd_buf->pos=0;
    pcmd_buf->sw=0;
}

int zh_gpsDriver::GetData(GPS_DATA    &GPSData)
{
    PDF(3, "zh_getGPSdata!\n");
    GPSData.lati=mpGPS_GPRMC->latitude;
    GPSData.lont=mpGPS_GPRMC->longitude ;
    GPSData.sw=mpGPS_GPRMC->gps_sw;
    return mpGPS_GPRMC->gps_sw;
}

int zh_gpsDriver::GetData(GPS_STASNR & GPSStaSNR)
{
    GPSStaSNR=mGPSStaSNR;
}

int zh_gpsDriver::setCOM(COM_PARAM    & COMParam)
//它是唯一控制串口的,
{
    int i;
    struct termios attr;
    // 打开串口
    if(fd==-1)
    {
        //close(fd);
        COMParam.COMfd = open(COMParam.SerialDevicePath, O_RDWR | O_NONBLOCK);
//
        fd=COMParam.COMfd;
```

```

        if (COMParam.COMfd == -1)
        {printf("open device:%s error!\n", COMParam.SerialDevicePath);return
-1;}
        else printf("open device:%s ok!\n", COMParam.SerialDevicePath);
    }
    // 读取串口当前属性
    i=tcgetattr(fd , &attr);
    if(i!=0) {printf("error for get attr!\n");return -1; }
    //=====设置好波特率
    if(COMParam.BaudRate==4800)
    {
        i=cfsetispeed(&attr,B4800);
        i=cfsetospeed(&attr,B4800);
        if(i!=0) {printf("error for get attr!\n");return -1; }
    }
    else if(COMParam.BaudRate==9600)
    {
        i=cfsetispeed(&attr,B9600);
        i=cfsetospeed(&attr,B9600);
        if(i!=0) {printf("error for get attr!\n");return -1; }
    }
    else printf("error BaudRaet=%d,only support BaudRate 4800/9600!
\n", COMParam.BaudRate);
    if((tcsetattr(fd, TCSANOW, &attr))!=0) {printf("active set attr
error!\n");return -1; }
    //=====
    return fd;
}

void zh_gpsDriver::WfifoFixString(char data)
{
    static int flag=0;
    if(data!='(',')') flag=0;
    else {if(flag==1) fifoctl(&fifo,'0',WR);
    else flag=1;
}
fifoctl(&fifo,(int) data,WR);
}

void zh_gpsDriver::timerEvent(QTimerEvent * event)
{
    int readbytes,i,j;
    char c;

```

```

if(event->timerId()==id)
{
    PDF(3,"zh_gpsDriver timer is alarm!\n");
    //=====

    for(j=0;j<7;j++)
    {
        readbytes = (int)read( fd, rdBuf, sizeof(rdBuf));
        if ( readbytes > 0 )
        {
            PDF(2,"tty readbytes1=%d\n",readbytes);
            for ( i = 0 ; i < readbytes ; i++ ) {c = rdBuf[i];WfifoFixString(c);}
            //=====
            while(GPRS_GetTextLineFromFIFO(&buf))
            {CMD_Analysis(&buf);} // 解析完所有的语句。
        }
    }
    PDF(3,"zh_gpsDriver timer out!\n");
    //=====
}
else PDF(3,"not this timer is alarm!\n");
}

```

Map 类程序:

```
#include "zhMap.h"
```

```
//=====经纬值
```

```
#define zhPR 8 //use for the zh point size
```

```
int zhMap::CMD_Analysis( char * pcmd_buf )
```

```

{
    char*    pcmd;
    int i;
    char msgid[20];
    //=====
    pcmd=strstr(&pcmd_buf[0],"mapinfo:");//只对此语句解析
    if (pcmd!=NULL)
    {
        i= sscanf(pcmd,"%[^:]:%lf,%lf,%lf,%lf,%d,%d,%d",
        msgid, &A_lont,&A_lati, &B_lont,& B_lati, &Width,&Height,/*7*/
        &subWidth);
        //请检查 sscanf 的返回值，可能会失败，如果小于要收得的数目。
        if(i<8) {PDF(2,"sscanf error mapcfg\n"); return 1;}
    }
}

```

```

        PDF(2, "A_lont=%lf, A_lati=%lf, B_lont=%lf, B_lati=%lf\n", A_lont, A_lati,
B_lont, B_lati);
        PDF(2, "Width=%d, Height=%d, subWidth=%d\n", Width, Height, subWidth);
        return 0;
    }
    //=====
    pcmd=strstr(&pcmd_buf[0], "xxx");//只对此语句解析
    if (pcmd!=NULL) {}
    return 0;
}
int zhMap::readMapCfg(QString Mapstr)
{
    Mapstr+="/map.cfg";
    const                                     char                                     *
OfilePtr=Mapstr.toLatin1(); //Mapstr.toString(); //"/.config.txt";
    FILE * infile;
    char msg[512];
    PDF(2, "OfilePtr=%s\n", OfilePtr);
    if ((infile = fopen(OfilePtr, "rb")) == NULL) {
        PDF(1, "can't open map.cfg file\n");
        return -1; }
    //=====
    while(fgets(msg, sizeof(msg), infile)!=NULL)
    {
        PDF(2, "mapcfg=%s\n", msg);
        CMD_Analysis(msg);
    }
    fclose(infile);
    return 0;
    //=====
}
zhMap::zhMap(QWidget * parent ,const char * name):QWidget(parent, name)
{
    /* A_lont=114.350683; //左上点经度 小数度数制
    A_lati=30.5360; //左上点纬度

    B_lont=114.3684; //右下点经度
    B_lati=30.521867; //右下点纬度
    Width=1080; //像素的
    Height=978;
    subWidth=300; //分割时, 子图的尺寸*/
    //=====设置好地图路径
    QStrmappath="/home/map";

```

```

mapPath=QStrmappath. latin1();
readMapCfg(QStrmappath);
//=====
WidthNumber=Width/subWidth+1;//图像水平分割的个数
HeightNumber=Height/subWidth+1;//图像垂直分割的个数
pixPerLont=((double)Width)/(B_lont-A_lont);
pixPerLati=((double)Height)/(A_lati-B_lati);
//=====
PDF(2, "WidthNumber=%d           HeightNumber=%d           pixPerLont=%lf
pixPerLati=%lf\n", WidthNumber, HeightNumber, pixPerLont, pixPerLati);
}
int zhMap::SetMapPath(QString & Qstr)
{
    QStrmappath=Qstr;
    mapPath=QStrmappath. latin1();
    readMapCfg(QStrmappath);
    //=====
    WidthNumber=Width/subWidth+1;//图像水平分割的个数
    HeightNumber=Height/subWidth+1;//图像垂直分割的个数
    pixPerLont=((double)Width)/(B_lont-A_lont);
    pixPerLati=((double)Height)/(A_lati-B_lati);
    //=====
    PDF(2, "WidthNumber=%d           HeightNumber=%d           pixPerLont=%lf
pixPerLati=%lf\n", WidthNumber, HeightNumber, pixPerLont, pixPerLati);
}

int zhMap::getMapABinfo(GPS_DATA & A_gpsdata, GPS_DATA & B_gpsdata)
{
    A_gpsdata. lati=A_lati;
    A_gpsdata. lont=A_lont;
    B_gpsdata. lati=B_lati;
    B_gpsdata. lont=B_lont;
}

int zhMap::getCurrMap(const GPS_DATA & gpsdata, QPixmap & pixmap)
{
    int xcenter, ycenter;
    int Ax, Ay, Bx, By, Cx, Cy, Dx, Dy;
    int idA, idB, idC, idD;
    int idi, idj;//
    int AxW, AyH;//用于处理 A 点的图块，表明取了多少走

```

```
QString fileStr;
//判断是否在范围

if(!(gpsdata.lont<B_lont && A_lont<gpsdata.lont))
{
    PDF(1,"out of lont range!\n");
    return -1;
} //out of range!

if(!(gpsdata.lati<A_lati && B_lati<gpsdata.lati))
{
    PDF(1,"out of lati range!\n");
    return -1;
} //out of range!!
//中心点转换成像素点
xcenter=(int)((gpsdata.lont-A_lont)*pixPerLont);
ycenter=(int)((A_lati-gpsdata.lati)*pixPerLati);
//==计算出请求图的四点像素坐标
Ax=xcenter-(zhGpsWidth)/2;
Ay=ycenter-(zhGpsHeight)/2;
Bx=Ax+zhGpsWidth;
By=Ay;
Cx=Ax;
Cy=Ay+zhGpsHeight;
Dx=Bx;
Dy=Cy;
//由四点像素坐标计算要装载的图片号。
idi=Ax/subWidth+1;
idj=Ay/subWidth;
idA=idj*WidthNumber+idi;

idi=Bx/subWidth+1;
idj=By/subWidth;
idB=idj*WidthNumber+idi;

idi=Cx/subWidth+1;
idj=Cy/subWidth;
idC=idj*WidthNumber+idi;

idi=Dx/subWidth+1;
idj=Dy/subWidth;
idD=idj*WidthNumber+idi;
//装载备用图片，如果已经存在就不要装了
```

```
fileStr.sprintf("%s/_%02d.jpg",mapPath,idA);
picloadFlag=picmapA.load(fileStr);

fileStr.sprintf("%s/_%02d.jpg",mapPath,idB);
picloadFlag=picmapB.load(fileStr);

fileStr.sprintf("%s/_%02d.jpg",mapPath,idC);
picloadFlag=picmapC.load(fileStr);

fileStr.sprintf("%s/_%02d.jpg",mapPath,idD);
picloadFlag=picmapD.load(fileStr);
//===以左上点为标准，计算并装载的各区域
paintmap.begin(&picmap,this);

AxW=subWidth-Ax%subWidth;
AyH=subWidth-Ay%subWidth;
if(AxW>zhGpsWidth && AyH>zhGpsHeight)
{
    paintmap.drawPixmap(0,0,picmapA,(Ax%subWidth),(Ay%subWidth),(zhGpsWidth),
(zhGpsHeight));//org x y pic pic x y w h
}
else if(AxW>zhGpsWidth && AyH<=zhGpsHeight)
{
    paintmap.drawPixmap(0,0,picmapA,(Ax%subWidth),(Ay%subWidth),(zhGpsWidth),
AyH);//org x y pic pic x y w h
    paintmap.drawPixmap(0,AyH,picmapC,(Ax%subWidth),0,(zhGpsWidth),Cy%subWidth);
//org x y pic pic x y w h
}
else if(AxW<=zhGpsWidth && AyH<=zhGpsHeight)
{
    paintmap.drawPixmap(0,0,picmapA,(Ax%subWidth),(Ay%subWidth),
AxW,AyH);//org x y pic pic x y w h
    paintmap.drawPixmap(AxW,0,picmapB,0,(By%subWidth),
(zhGpsWidth-AxW),AyH);//org x y pic pic x y w h
    paintmap.drawPixmap(0,AyH,picmapC,(subWidth-AxW),0,
AxW,(zhGpsHeight-AyH));//org x y pic pic x y w h
    paintmap.drawPixmap(AxW,AyH,picmapD,0,0,
(zhGpsWidth-AxW),(zhGpsHeight-AyH));//org x y pic pic x y w h
}
else if(AxW<=zhGpsWidth && AyH>zhGpsHeight)
{
    paintmap.drawPixmap(0,0,picmapA,(Ax%subWidth),(Ay%subWidth),
AxW,zhGpsHeight);//org x y pic pic x y w h
```

```

        paintmap.drawPixmap(AxW, 0,                                pixmapB, 0, (By%subWidth),
(zhGpsWidth-AxW), zhGpsHeight); //org x y pic pic x y w h
    }
    //=====
    paintmap.end();
}

View ::View(QCanvas& canvas, QWidget * parent ,const char * name):
QCanvasView(&canvas, parent, name)
{
    //===paint a comb pic
    pcanvas=&canvas;
    zhmap.resize(QSize(zhGpsWidth, zhGpsHeight));
    mainmap.getMapABInfo(MapInfoApositon, MapInfoBpositon);
    maingps.lati=MapInfoBpositon.lati+(MapInfoApositon.lati-MapInfoBpositon.lati)/2;
    maingps.lont=MapInfoApositon.lont+(MapInfoBpositon.lont-MapInfoApositon.lont)/2;
    mainmap.getCurrMap(maingps, zhmap);
    canvas.setBackgroundPixmap(zhmap);
    canvas.resize(zhGpsWidth, zhGpsHeight);
    //setFixedSize(zhGpsWidth, zhGpsHeight);
    r = new QCanvasRectangle(zhGpsWidth/2, zhGpsHeight/2, zhPR, zhPR, &canvas); //
    //r->setVelocity(0.5, 0); // speed
    r->setBrush(QColor(250, 10, 10));
    r->setZ(10);
    r->show();
}

void View::setPosition(double lontchange, double latichange, int sw)
{
    maingps.lati=latichange;
    maingps.lont=lontchange;
    maingps.sw=sw;
    mainmap.getCurrMap(maingps, zhmap);
    pcanvas->setBackgroundPixmap(zhmap);
    //pcanvas->update();
}

void View::changePosition(double lontchange, double latichange)
{
    maingps.lati+=latichange;
    maingps.lont+=lontchange;
}

```



```
PDF(2, "change lat=%lf lont=%lf\n", maingps.lati, maingps.lont);
//=====
mainmap.getCurrMap(maingps, zhmap);
pcanvas->setBackgroundPixmap(zhmap);
//pcanvas->update();
}
```

```
void View::contentsMouseEvent(QMouseEvent *e)
{
/*if(e->pos().x())
maingps.lati+=0.001;
maingps.lont+=0.001;
mainmap.getCurrMap(maingps, zhmap);
pcanvas->setBackgroundPixmap(zhmap);
r->move(e->pos().x(), e->pos().y()); */
}
```

strSNR 类程序

```
#include "staSNR.h"
#include <qpainter.h>
#include <qtooltip.h>
```

```
#define X_ADJ(a) ((int)(float(a)*xScreenCorr))
#define Y_ADJ(a) ((int)(float(a)*yScreenCorr))
#define S_ADJ(a) ((int)(float(a)*xyScreenCorr))
```

```
float xScreenCorr=1, yScreenCorr=1, xyScreenCorr=1; // global, bad but easy
```

```
SatSNR::SatSNR( QWidget * parent, const char * name, WFlags f):
    QWidget(parent, name, f)
```

```
{
    setFixedSize(240, 150);

    int i;
    for(i=0; i<12; i++)
    {d_pSatInfo[i].d_snr=68;
    d_pSatInfo[i].d_updated=true;

    d_pSatInfo[i].d_satName="p";
    }
}
```

```
SatSNR::~~SatSNR()
{
}

void SatSNR::SetData(const GPS_STASNR & GPSStaSNR) {

for (int i = 0; i < 12; ++i)
{
    SatInfo & satInfo = d_pSatInfo[i];
    STA_INFO STAINfo=GPSStaSNR.Sta[i];
    satInfo.d_snr=STAINfo.StaSNR;
    satInfo.d_updated=true;
}
update();
}

void SatSNR::paintEvent( QPaintEvent * )
{
    p.begin(this);
    QPainter * painter=&p;
    int xSPC = X_ADJ(4);
    int ySPC = Y_ADJ(4);
    int WIDTH = X_ADJ(15);
    int offset = X_ADJ(2);
    //=====设定好一些参数
    int x = xSPC + offset;
    int y = minimumHeight() - ySPC;
    int min = x;
    int max = min + 11*(WIDTH + xSPC) + WIDTH - 1;
    int threshold = Y_ADJ(7); // 7% mark
    //===计算好一些参数
    painter->setPen( QPen( Qt::black, 0, QPen::SolidLine) );

    for (int i = 0; i < 12; ++i)
    {
        SatInfo & satInfo = d_pSatInfo[i];
        //获取一个卫星信号
        int strength = satInfo.d_snr; //scaled 50pixel == 100%    //50 个像素表示满强度
        int clip = x + WIDTH + xSPC - 1;
        if (clip > max) clip = max;
        if (satInfo.d_satName.isEmpty())
```

```

{
    painter->eraseRect(x, y+Y_ADJ(10), WIDTH, Y_ADJ(-99));

    // draw the marker lines
    for (int p = threshold; p <= Y_ADJ(50); p+=Y_ADJ(10) )
        painter->drawLine ( x, y-Y_ADJ(12)-p, clip, y-Y_ADJ(12)-p );

}
else if (satInfo.d_updated)
{
    painter->eraseRect(x, y+Y_ADJ(10), WIDTH, Y_ADJ(-99));

    // repair the marker lines
    for (int p = threshold; p <= Y_ADJ(50); p+=Y_ADJ(10) )
        painter->drawLine ( x, y-Y_ADJ(12)-p, clip, y-Y_ADJ(12)-p );

    painter->drawText(x+X_ADJ(2), y, satInfo.d_satName);

    // when signal is lower than threshold, gps does not use it for
reading
    //如果信号强度低于保持值, gps 不会使用它, 所以用红色表示。
    const QColor &c = Qt::red;
    //(gpsData->d_aliveGPS ? (strength < threshold ? Qt::red : Qt::green) :
Qt::red);
    painter->setBrush( QBrush( c, QBrush::SolidPattern) );

    painter->drawRect(x, y-Y_ADJ(13), WIDTH, Y_ADJ(-strength));

    // unmark
    // satInfo.d_updated = false;
}

    x = x + WIDTH + xSPC;
}

// 0% mark
painter->setPen( QPen( Qt::black, 2, QPen::SolidLine) );
painter->drawLine ( min, y-Y_ADJ(12), max+1, y-Y_ADJ(12) );
p.end();
}

```