

文章编号: 2096-1472(2016)-03-49-02

## 基于Linux+ARM的GPS模块设计

周小仁

(黄冈职业技术学院, 湖北 黄冈 438002)

**摘要:** 随着移动终端的普及, 嵌入式Linux系统越来越得到广泛的应用, 同时作为定位领域最为常用的GPS信号, 也需要逐步应用到基于Linux的移动便携终端中来, 使得各种移动终端的位置信息更加可靠, 同时使得各设备之间互联及数据传输变得高效和方便, 本文旨在设计一款基于Linux+ARM的GPS通信模块。

**关键词:** 嵌入式; Linux; GPS; ARM

**中图分类号:** TP391 **文献标识码:** A

## Design of GPS Module Based on Linux+ARM

ZHOU Xiaosa

(Huanggang Polytechnic College, Huanggang 438002, China)

**Abstract:** With the popularity of mobile terminal, embedded Linux system, more and more widely used, at the same time as the location of most commonly used GPS signal, also need to gradually applied to Linux mobile portable terminal based on, makes a variety of mobile terminal location information more reliable at the same time between each equipment interconnection and data transmission more efficient and convenient. The aim of this paper is to design a GPS communication module based on Linux+arm is being paid.

**Keywords:** embedded; Linux; GPS; ARM

### 1 引言(Introduction)

何谓GPS呢, 就是全球定位系统, 英文为Global Positioning System。换言之, 是一个由覆盖全地球的24颗卫星组成的卫星定位系统。此项技术可作为航空器、船只、汽车和人的定位和导向, 能够较为安全和准确地沿着既定的路线, 准确的到达所需目的地。美国主导的全球定位系统简称GPS, 是20世纪70年代末期美军三军联合研制的军用卫星来进行导航和定位的系统。它研制的主要目的是为三军的三个大的领域提供全球性、长时间的和实时的导航和定位信号, 还可用于军事情报的收集、核爆炸的监测和应急通讯等, 是美国控制全球战略的重要组成部分之一。历经30多年的发展和壮大, 消耗资金超过400亿美金, 至今其全球覆盖率已经达到99%的二十余颗GPS卫星都已经全部布置完成。上述GPS全球卫星定位系统主要由以下几大部分构成, 它们是空间部分, 即GPS星座, 地面控制部分, 即地面监控系统, 用户终端接收设备部分, 即GPS信号接收器<sup>[1]</sup>。

### 2 定位系统的构成(Composition of positioning system)

#### 2.1 定位系统的三大构成部分

地球周边太空空间部分, 定位系统需要21颗工作卫星组成, 后备的卫星为三颗。地球上的后台支撑系统, 主要有主控站点、三个注入站点、五个监测站点共同组成。终端用户的信号接收设备部分, 用来接收GPS卫星所发射来的信号, 来获取所需的导航和定位数据, 经过信号的分析 and 处理, 从而来实现导航和定位的目的。本定位系统的接收机硬件主要由主控机、天线和电源部分所构成。

#### 2.2 全球定位系统的定位思路

依据高速运动的卫星瞬时的位置作为已知的初始数据, 来达到定位的目的, 利用空间距离后方交会的方式, 从而来确定待检测点的位置数据。当前定位系统所提供的定位精度是9米以内, 误差较大, 为了提高定位的精度, 经常利用差分的定位技, 在基准站点之上安放一台GPS接收器进行检测。依据该基准站点的精准坐标值, 来计算卫星到基准站点的距离误差修正数据, 然后由基准站点把此数据实时发送至终端用户。终端接收器进行系统定位检测的时候, 同时接收到基准站点所发来的修正数据, 综合对定位结果数据进行修正, 从而来提高定位精度。差分定位系统主要分两大类, 它们是伪距差分 and 载波相位差分<sup>[2]</sup>。伪距差分的思想, 该方法应用最为广泛的差分方式。即在基准站点之上, 检测所有卫星数据, 依据基准站已有的坐标数据和诸卫星的坐标数值, 解出每颗卫星在每一个时刻距离基准站点的实际距离值。然后与测得的伪距离相比较之后, 计算出伪距修正数, 把它发送至用户终端接收器, 以此来提高定位精度。此差分方式, 能得到米级定位精度, 例如, 我国开放地区大量应用的信标差分。载波相位差分思路, 该差分技术简称RTK, 即Real Time Kinematic, 是及时计算若干个测站载波相位观测值的差分方式。就是把基准站点采样到的数据发送给终端用户接收器, 从而来求解坐标值。此方式能使定位精度达到厘米级。其广泛应用在需要高精度定位的领域。此方式经历了十多年的发展, 随着GPS技术的不断改良, 该系统的软件和硬件技术也在不断完善, 不断的拓展其应用领域, 至今已经遍及经济社会的各个领域, 同时逐步深入到老百姓的日常生活当中。本全

球定位系统的强大的作用还会继续方便人们的生活<sup>[3]</sup>。

### 3 GR-87模块介绍(GR-87 module introduction)



Pin	Pin Name	Function description
1	VCC_5V	+3.5-5.5Vdc power input
2	TXA	Serial Data output port A(TTL Level: Voh≥2.4V Vol≤0.4V Ioh=Iol=2mA)
3	RXA	Serial Data input port A(TTL Level Vih≥0.7VCC Vil≤0.3VCC)
4	RXB	Serial Data input port B(TTL Level Vih≥0.7VCC Vil≤0.3VCC)
5	GND	Power ground
6	TIMEMARK/RESET (option)	TIMEMARK: 1PPS Time mark output(Vil≤0.2V Pulse Width 10ms) RESET: Reset Input (Active Low)(option function)

该模块与MCU的接口是普通TTL电平串口，非常方便。另外模块使用SiRF第二代高效能晶片，大大缩小体积。快速定位及追踪23颗卫星的能力。晶片内建1920次/频率硬体，提高接收传送搜寻卫星信号。GR-87模块输出的数据格式遵循标准GPS接收机定位的数据格式——“NMEA-0183”数据格式。

### 4 具体实现步骤(Specific implementation steps)

(1)插上GPS模块后上电,启动开发平台

用普通交叉串口线连接开发平台的DBGU串口到PC的串口,用USB电缆连接开发平台的USB Slave口到PC,使用网线可以连接开发平台的100M网卡到PC的网口。

(2)使用时要注意GPS模块串口的波特率

程序所要做的工作是通过串口从GR-87模块接收数据然后进行分析,得到经纬度,时间等信息。部分核心代码如下:

```
int main(int argc,char *argv[])
{int i,char huanchong[GPS_RECV_CMD_MAX_BUF]; //receive huanchong
char tmp;XINXI xinxi,char Jiaodu[4];
char Fenzhong[8];char Shizhong[4];char Minu[4];
char Miao[4];char Zhuangtaishu[4];int number=0; //
open the COM
bsp_init();int recv_count;rs485_open();
printf("test GPS...\r\n");while(1){while(!Is_startGP(fd_rs485));
for(i=0;i<256;i++){read(fd_rs485,&tmp,1);huanchong[i]=tmp;if(tmp=='\n')break;}
if(Is_GGA(huanchong)){xinxi.bIsGPGA=1;number=6;
for(i=0;i<4;i++) Shizhong[i]=0;
Shizhong[0]=huanchong[number];Shizhong[1]=huanchong[number+1];number += 2;for(i=0;i<4;i++)Minu[i]=0;
Minu[0]=huanchong[number];Minu[1]=huanchong[number+1];number+=2;for(i=0;i<4;i++)Miao[i]=0;
Miao[0]=huanchong[number];Miao[1]=huanchong[number+1];xinxi.shizhong=atoi(Shizhong);
xinxi.min=atoi(Minu);xinxi.sec=atoi(Miao);
xinxi.secFrac=0;xinxi.bjshizhong=xinxi.shizhong+8;
if(xinxi.bjshizhong>24)xinxi.bjshizhong-=24;number=17;
for(i=0;i<4;i++)
```

```
Jiaodu[i]=0;Jiaodu[0]=huanchong[number];Jiaodu[1]=
huanchong[number+1];number+=2;for(i=0;i<8;i++)
Fenzhong[i]=0;
```

```
for(i=0;i<7;i++)Fenzhong[i]=huanchong[number+i];
xinxi.latitude=(atof(Fenzhong))/60.0 + atoi(Jiaodu);
```

```
xinxi.latNS=huanchong[27];number=29;
```

```
for(i=0;i<4;i++)Jiaodu[i]=0;
```

```
Jiaodu[0]=huanchong[number];Jiaodu[1]=huanchong[
umber+1];Jiaodu[2]=huanchong[number+2];number+=3;for
(i=0;i<8;i++) Fenzhong[i]=0;
```

```
for(i=0;i<7;i++)Fenzhong[i]=huanchong[number+i];
```

```
xinxi.longitude=(atof(Fenzhong))/60.0+atoi(Jiaodu);
```

```
xinxi.lgtEW=huanchong[40];number=44;
```

```
for(i=0;i<4;i++)Zhuangtaishu[i]=0;Zhuangtaishu[0]=
huanchong[number];Zhuangtaishu[1]=huanchong[
umber+1];xinxi.satellites=atoi(Zhuangtaishu);
```

```
printf("UTC Time:%d h%d m%d s\n",xinxi.
shizhong,xinxi.min,xinxi.sec);
```

```
printf("BJ Time:%d h%d m%d s\n",xinxi.
bjshizhong,xinxi.min,xinxi.sec);
```

```
printf("Latitude:%c wei %f\n",xinxi.latNS,xinxi.
latitude);
```

```
printf("Longitude:%c jing %f\n",xinxi.lgtEW,xinxi.
longitude);
```

```
printf("Satellites:%d\n",xinxi.satellites);printf("\n");
```

```
}else continue;sleep(1);return 0;}
```

(3)编译生成目标代码下载至目标板然后运行调试

直接使用arm-linux-gcc命令进行编译,也可先编好Makefile文件,然后用make命令编译。我们采用后一种方法,较规范。

(4)运行之前先进行如下操作

①cd dev;②mkdir tts;③ln s/dev/tq2440\_serial1/dev/tts/1然后运行目标程序,调试发送和接受数据。

### 5 结论(Conclusion)

本文是作者在多年从事Linux和嵌入式ARM开发的实践过程中,就基于Linux+ARM的GPS模块的开发所得到的一点心得体会,希望对读者有所帮助,由于本人能力有限,技术细节不够完善,尚未实现多GPS模块的多点通讯,没能把GPS模块驱动程序加载至Linux内核中开机自动运行,不足之处,希望读者朋友多多的指正。

### 参考文献(References)

- [1] 田世君.高灵敏度GPS定位及组合导航技术研究[D].电子科技大学,2009.
- [2] 唐康华.GPS/MIMU嵌入式组合导航关键技术研究[D].国防科学技术大学,2008.
- [3] 公磊.基于Android的GPS测量系统开发[D].南昌大学,2008.

### 作者简介:

周小仁(1980-),男,硕士,讲师.研究领域:嵌入式,EDA,单片机技术开发。