单片机解读 GPS 信息的程序设计

用单片机解读 GPS 信息是 GPS 模块使用最重要的环节，由于汲设到产品的保密问题，这里只介绍时间的处理方法，而方位、速度的处理方法不做介绍，但通过时间的处理方法，同样可以处理方位、速度。

1、 获取 GPS 模块的输出信息

由于 GPS 模块每秒输出一次：$GPGGA 、$GPGSA、$GPGSV、$GPRMC 数据。速率慢，因此必须采用中断方式接收！（采用查询会造成数据丢失），而且单片机只需要处理$GPRMC 信息，即可得到时间、方位、速度。程序采用 C51 编制，在 Keil C 中编译！

```
code unsigned char GPS_ASC[]="$GPRMC" ;   //定义特征字符串
unsigned char       idata    RsBuf[60];
//*************************************************************
//读取 GPS 模块串口数据, 采用中断方式
void GetRs232_Data() interrupt 4 {
    unsigned char i;
    unsigned int j;
    if (RI){
        RI=0;
        RsBuf[0]=SBUF;
        if (RsBuf[0] =='$'){     //是 GPS 数据的开始，进入查询接收
            for (i=1;i<sizeof(GPS_ASC)-1;i++){
```

```
                j=GetUartDat();         //接收下一个数据
                if (j<256) {
                    RsBuf[i]=(unsigned char)j;
                    if (RsBuf[i]!=GPS_ASC[i]) return;
                }
            }
//判别是否为$GPRMC 数据，是继续接收!
            for (;i<sizeof(RsBuf);i++){
                j=GetUartDat();
                if (j<256) {
                    RsBuf[i]=(unsigned char)j;
                }else{
                    break;
                }
            }
//接收完毕处理数据!
            if (1==JiaoYanDat(RsBuf)){
                FormatTimer(RsBuf);          //格式化时间
                FormatSpeed(RsBuf);          //处理速度
            }
        }
    }
```

```
}
```

2、 接收数据的处理

数据处理前，必须要再次检验是否为$GPRMC 信息。

```
unsigned char JiaoYanDat(unsigned char *p){
    unsigned char *pP;
    pP=strstr(p,GPS_ASC);
    if (pP == NULL) return 0;
    return 1;
}
```

3、接收的数据全部转换成二进制数据，便于纠错处理。

```
unsigned char ASC_To_Bin(unsigned char *p){
    unsigned char i;
    if (p[0]<'0' || p[0]>'9') return 0;
    if (p[1]<'0' || p[1]>'9') return 0;
    i=(p[0]-'0')*10;
    return (i+p[1]-'0');
}
```

4、时间调整

由于 GPS 模块采用格林威治时间，与北京时间相差 8 个时区，所以要加入 8 小时。

```
unsigned char TurnGLW(unsigned char GLW){
    return (GLW+8)%24;
```

}

5、 格式化标准时间

定义一个时间的数据结构:

```
typedef struct{
    unsigned int    ms;
    unsigned char Sec;
    unsigned char Min;
    unsigned char Hour;
    unsigned char Day;
    unsigned char Mon;
    unsigned char Week;
    unsigned char Year;
}TIMER;
```

由于 GPS 模块输出的毫秒不准，所以时间数据结构中的 ms 变量数据不准！请不要随便使用！

```
void FormatTimer(unsigned char *p){
    unsigned char i;
    Timer.ms=(p[14]-'0')*100+(p[15]-'0')*10+p[16]-'0';
    Timer.Hour=TurnGLW(ASC_To_Bin(&p[7]));
    Timer.Min=ASC_To_Bin(&p[9]);
    Timer.Sec=ASC_To_Bin(&p[11]);
    if (p[50]==','){
```

```
        Timer.Day=ASC_To_Bin(&p[51]);
        Timer.Mon=ASC_To_Bin(&p[53]);
        Timer.Year=ASC_To_Bin(&p[55]);
    }
}
```

6、 单片机接收串口的数据

由于单片机接收 GPS 后续数据，采用了查询方式！所以要注意避免死机!必须要带超时处理！

```
 unsigned int GetUartDat(void){
    unsigned int i=0,j=0;
    RI=0;
    while(!RI){
        if (i++>30000){    //超时时间
            j=256;
            return j;     //退出标志
        }
    }
    j=SBUF;
    RI=0;
    return j;
}
```

7、单片机的串口设置

采用 89C52 接收 GPS 数据的串口设置

```
#define    OSC      0          //定义使用晶振 0=11.0592 1=18.4320
#if OSC
    code char BPSAsc[]={
    0x60,0xb0,0xd8,0xec,0xf6,0xfb,0xfb,
    };  //18.4320MHz
#else
    code char BPSAsc[]={
    0xa0,0xd0,0xe8,0xf4,0xfa,0xfd,0xfd,
    };  //11.0592MHz
#endif
```

串口设置

```
void InitBps (char Bps ){
    PCON &=0x7f;    //PCON。7==0
    SCON =0x50;  //设置成串口 1 方式
    TH1=BPSAsc[Bps];    //22.1184 必须*2 2004-12-16
    TL1=TH1;
    TR1=1;
}
```