



湖南大學
HUNAN UNIVERSITY

课程设计报告

课程设计名称: 基于 gps 和北斗的定位器

专 业 班 级: 计科 2002 班

姓 名: 杨鹏宇

学 号: 202004061409

完 成 时 间: 2022 年 8 月 31 日

信息科学与工程学院

摘要

实时定位功能广泛应用于我们的生活当中, 定位信息为我们提供了多种便捷的服务。实现准确的定位不仅能便利我们的生活, 也能满足各类场景下的需要。

本设计在 STC 学习板上进行 GPS+北斗双模定位报文解析, 完成校时, 定位及坐标记录, 电子围栏, 测距共四种卫星定位的实际应用, 结合定位模块可提供基本的卫星定位服务。

关键词: 定位; GPS 和北斗定位报文; STC

目录

- 第一章 设计背景及目的 1
 - 1.设计背景 1
 - 2.设计目的..... 1
- 第二章 设计与实现 1
 - 1.整体设计 1
 - 2.功能设计..... 2
 - 3.按键设计及相应的状态转换 3
 - 4.功能实现..... 4
 - 4.1 使用的变量及函数 4
 - 4.2 串口接收回调..... 4
 - 4.3 按键回调..... 5
 - 4.4 显示函数 6
 - 4.5 导航按键回调..... 7
 - 4.6 距离计算..... 9
- 第三章 功能测试..... 10
 - 1.时间校准测试 10
 - 2.定位测试..... 10
 - 3.防走失/防盗功能测试 11
 - 4.测距功能测试 11
 - 5.户外实测..... 12
- 第四章 总结 12

第一章 设计背景及目的

1.设计背景

在日常生活中，我们通常使用手机就可以实现定位，并使用基于位置的各类服务。除了手机定位提供的导航服务等外，其他情景下的定位也有广泛的应用，如车载定位(提供导航，防盗)，农林业的坐标采集，地址勘探，还有可以保障儿童或老人安全活动范围的电子围栏等，卫星定位的校时也用于各类调度系统。

通常广泛使用的定位系统为 GPS 系统。近年来，我国的北斗卫星导航系统也已能够为全球提供全天候，高精度的定位。目前的许多可定位设备都支持 GPS 与北斗双模定位。

2.设计目的

实现在 STC 学习板上对 GPS 和北斗双模定位报文的解析，完成校时，定位及坐标记录，电子围栏，测距四个定位器的基本功能，实现一个能够提供基本服务的卫星定位器。

通过完成该定位器的设计和实现，熟悉 STC 学习板的 BSP 库中各功能模块的使用，包括显示、按键、adc、串口通信等。并进一步熟悉此类电子系统的设计，实现，测试，的过程，掌握相关方法和技能。

第二章 设计与实现

1.整体设计

STC 学习板通过串口 2 连接定位模块，获取定位报文，并对定位报文进行解析，实现以下四个功能：

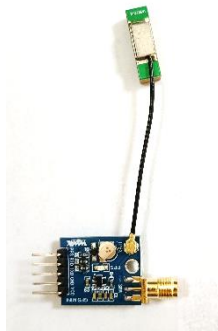
- 校时功能。用定位信息的UTC时间更新实时时钟并显示在数码管上。可用于各类调度系统。
- 定位功能。显示当前位置的经纬度，可将当前坐标信息存储，用于记录行动轨迹或标点。记录的坐标信息可以通过串口发送到PC。
- 防走失/防盗电子围栏功能。可设定特定距离(单位：米)，当定位与设定点超出该特定距离时，蜂鸣器发出警报，需要选出设定好的密码才能停止警报。能够设定密码。

- 测距功能。可测算两坐标点间的距离。起点开启后移动到另一坐标点，可显示距离。

以上功能通过按键进行切换，并将模式信息，操作信息及需要显示的时间，距离，经纬度信息显示在 led 和数码管上。使用的 BSP 库中的模块为：

- 串口 1：向计算机发送记录的坐标信息。在完成具体实现时，由于定位信息只能在户外获取，为了方便调试，使用串口 1 由计算机模拟发送定位报文信息。
- 串口 2：接收定位报文信息。
- 显示模块：在数码管上显示数据信息，led 显示模式信息。
- 按键模块与 ADC 模块：模式切换与设置。
- 实时时钟模块：UTC 校时。
- 蜂鸣器：用于电子围栏功能的超范围警报。
- 24C02 非易失性存储器：用于存储定位坐标和其他信息

使用的定位模块为 GPS 和北斗双模定位模块，接 IPEX 短天线，可在户外接收到定位信息。



2.功能设计

2.1 校时功能

按下 Key1 切换到时钟模式 (led 最低位亮起) 后，数码管显示实时时钟，并更新从定位信息中获取的时间。其中，获取的定位时间为 UTC 时间，+8 调整为北京时间。实际上应该由经度判断所在时区进行时间调整，为了保证完成所有功能，此处简化为校准到北京时间。

2.2 定位功能

按下 Key1 切换到定位模式 (led 第二位亮起) 后，数码管显示 8 位经度信息，导航按键向右或向左可以切换显示纬度信息。导航按键向上或向下可以选择存储坐标的位置，位置会显示在 led 的高四位，即共可记录 16 个坐标。按下 Key2 即

可记录当前坐标在非易失性存储器中，且位置值将自动+1。按下 Key3 将所有记录的坐标信息通过串口 1 发送。

2.3 防丢失/防盗功能

按下 Key1 切换到该模式 (led 第三位亮起) 后，数码管显示该功能是否打开 (显示 ON/OFF)，按下导航按键可以开启/关闭该功能。按下 Key2 将进入距离设定 (led 第 5 位)，通过导航按键向上向下，向左向右可以设定距离值 (单位 10m，最大 10km)。按下 Key3 将进入密码设定 (led 第 6 位亮)，通过导航按键向上向下，向左向右可以设定密码 (四位)，按下导航按键确认。

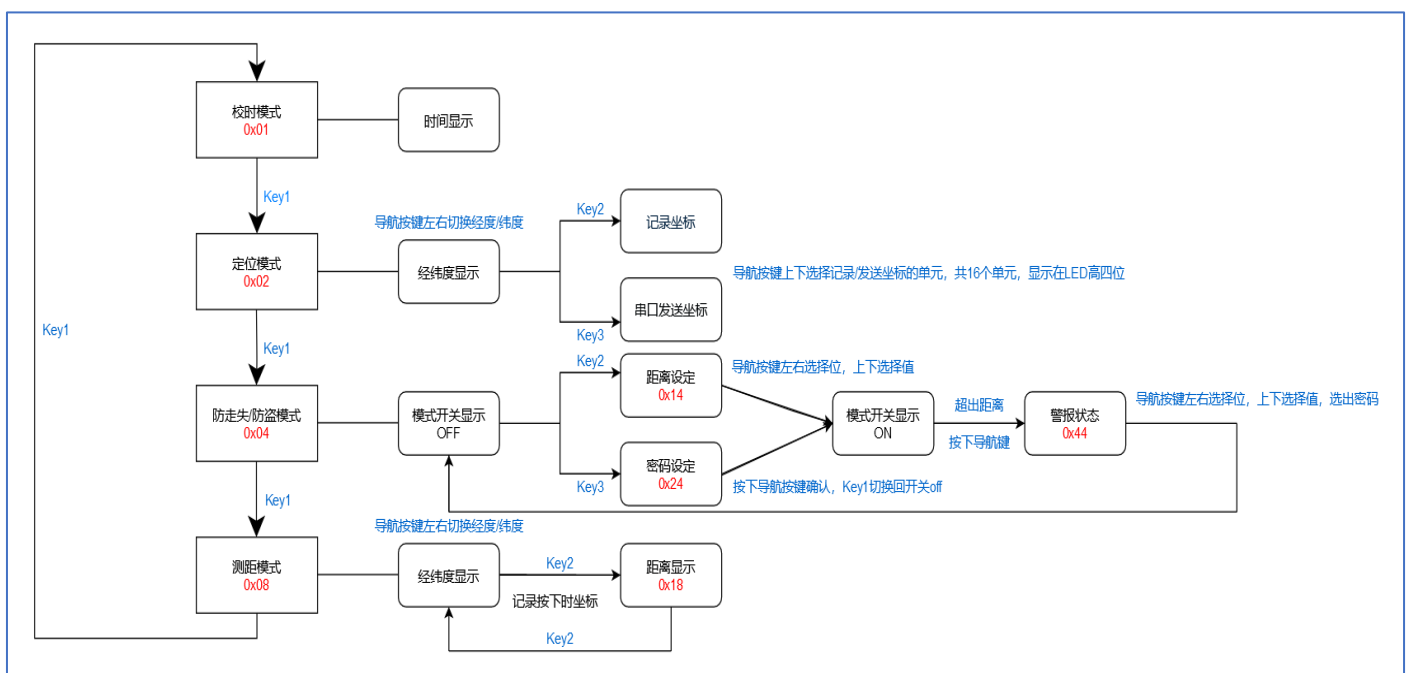
当该功能开启后，STC 会存储开启时的坐标位置，并在接收到定位信息时计算获取到的坐标与记录的坐标的距离，如果超过了设定的距离，蜂鸣器将产生蜂鸣，并可以通过导航按键向上向下更改数码管的数字，与设定的密码一致时，蜂鸣器将停止蜂鸣。

2.4 测距功能

按下 Key1 切换到该模式 (led 第四位亮起) 后，数码管显示与定位功能相同，按下 Key2 后，数码管将显示当前获取的定位坐标与按下 Key2 时的坐标距离。再次按下 Key2，数码管又会重新显示坐标，此时可以重新设置起点进行测距。

3.按键设计及相应的状态转换

根据整体设计，应共有四个模式，其中后两个模式还有不同的状态。根据各个功能需要的操作，对按键及相应的状态转换如下：



其中的状态 status 取值共有 8 种：

status	说明
0x01	表示处于校时模式，时间显示状态
0x02	表示处于定位模式，坐标显示状态
0x04	表示处于防盗防走失模式，开关显示状态
0x08	表示处于测距模式，坐标显示状态
0x14	表示处于距离设定状态
0x24	表示处于密码设定状态
0x44	表示处于警报状态
0x18	表示处于测距显示状态

4.功能实现

四个功能主要在报文处理的基础上，通过按键，数码管及 LED 写入，非易失性存储的读写等模块实现。

4.1 使用的变量及函数

实现过程中使用的变量包括模式的状态标记，时间，获取的经纬度数据，需要记录的经纬度数据，数据接收区等。如下：

```
code char matchhead[6] = "$GNGGA";
struct_DS1302_RTC time;           //时间
bit lon_lat = 0;                 //显示经度或纬度
uchar status = 0;                //模式选择
uchar i, weixuan=1;              //用于数字选择
uchar lock_flag=0;               //防盗/防走失是否开启, 存储在0x80, 上电时读取
uchar hour_temp;                 //小时, 用于时区调整
uchar lon_h, lon_m, lon_s, lat_h, lat_m, lat_s; //接收经纬度
uchar xlon_h, xlon_m, xlon_s, xlat_h, xlat_m, xlat_s; //警报/测距功能记录的坐标点经纬度
uchar point_addr = 0;            //坐标存储在非易失性存储的位置
xdata char buf[72];              //数据接收区
xdata char send_buf[19];         //数据发送区
xdata uint safety_dis, dis_temp, dis=0; //安全距离, 存储在0x81, 上电时读取; 设定距离; 实际距离
xdata uint lockcode, code_temp=0, trycode=0; //密码, 存储在0x85, 上电时读取; 设定时输入的密码; 尝试解锁输入的密码
```

使用的函数包括串口接收回调函数，按键和 adc 回调函数，更新数码管和 led 显示的函数以及距离计算函数。如下：

```
void uart_callback();             //串口接收回调
void key_callback();              //按键回调
void nav_callback();              //导航按键回调
void display();                   //数码管及led显示, 1s更新显示或按下按键后更新显示
void Delay10ms();                //延时函数, 用于写入非易失性存储器
int cal_dis();                    //距离计算
double radian(float);             //角度转弧度
```

4.2 串口接收回调

串口接收回调完成对接收报文的处理，更新经纬度坐标值及时间信息。

GPS 和北斗双模定位报文遵守 NMEA-0183 协议，报文有\$GNGGA 语句的定位信息，\$GPGSA 语句的卫星信息，\$GPGSV 语句的可见卫星数信息等多种语句信息。完成需要实现的功能，只需要接收\$GPGGA 语句的定位信息。

\$GPGGA 语句的基本格式如下（其中 M 指单位 M，hh 指校验和）：

\$GPGGA,(1),(2),(3),(4),(5),(6),(7),(8),(9),M,(10),M,(11),(12)*hh

(1)UTC 时间，格式为 hhmmss.sss；

(2)纬度，格式为 ddmm.mmmmm（度分格式）；

(3)纬度半球，N 或 S（北纬或南纬）；

(4)经度，格式为 dddmm.mmmmm（度分格式）；

(5)经度半球，E 或 W（东经或西经）；

(6)-(12)包括 GPS 状态，海拔高度，卫星数量和大地水准面高度等，不需要使用。

报文最大帧长为 72 字节，其中需要使用的只有 UTC 时间和经纬度信息。格式确定，可直接提取数值。需要注意的是 UTC 时间还要转为北京时间，且时钟数据结构中的时间为 BCD 码，先转换为数值调整时区后，再转为 BCD 码。而经纬度信息直接按照度分秒保存到定义好的变量当中。

保存好经纬度信息后，如果开启了防盗防丢失功能或测距功能，还要调用 cal_dis()更新记录的坐标点与当前坐标距离。如果这个距离超过了设定的安全距离，则切换到防盗防丢失模式的警报状态。

```
//报文接收处理
void uart_callback() {
    //UTC时间校准
    hour_temp = (buf[7]-'0')*10+(buf[8]-'0')+8;
    if(hour_temp>24) hour_temp -= 24;
    time.hour = (hour_temp/10)*16+hour_temp%10;
    time.minute = (buf[9]-'0')*16+(buf[10]-'0');
    time.second = (buf[11]-'0')*16+(buf[12]-'0');
    RTC_Write(time);
    //经纬度数据
    lat_h = (buf[18]-'0')*10+(buf[19]-'0');
    lat_m = (buf[20]-'0')*10+(buf[21]-'0');
    lat_s = ((buf[23]-'0')*1000+(buf[24]-'0')*100+(buf[25]-'0')*10+(buf[26]-'0'))*6/1000;
    lon_h = (buf[31]-'0')*100+(buf[32]-'0')*10+(buf[33]-'0');
    lon_m = (buf[34]-'0')*10+(buf[35]-'0');
    lon_s = ((buf[37]-'0')*1000+(buf[38]-'0')*100+(buf[39]-'0')*10+(buf[40]-'0'))*6/1000;
    //距离更新
    if(lock_flag == 1 || status == 0x18)
        dis = cal_dis();
    if(lock_flag==1 && dis>safety_dis)
        status = 0x44;
}
```

4.3 按键回调

按键回调处理各个模式下的按键操作，由于使用了 adc 模块，按键回调只处理 Key1 和 Key2 按下的情况。

按下 Key1 只进行模式切换。status 的低四位表示模式。

```

if(GetKeyAct(enumKey1) == enumKeyPress){
    //模式切换
    if(((status&0x0f) == 0) | ((status&0x0f) == 0x08)){
        status = 0x01;
    }
    else{
        status = (status&0x0f) * 2;
    }
}
}

```

对于 Key2 按下的情况，校时模式下无反应；定位模式下将记录当前坐标在非易失性存储器中（写入操作之间要延时 10ms）；防盗防走失模式下将进入距离设置状态；测距模式下则进入测距状态或回到坐标显示状态。

```

//定位功能下按下K2
if((status&0x0f) == 0x02){
    if(GetKeyAct(enumKey2) == enumKeyPress){
        M24C02_Write(point_addr, lat_h);
        Delay10ms();
        M24C02_Write(point_addr+0x01, lat_m);
        Delay10ms();
        M24C02_Write(point_addr+0x02, lat_s);
        Delay10ms();
        M24C02_Write(point_addr+0x03, lon_h);
        Delay10ms();
        M24C02_Write(point_addr+0x04, lon_m);
        Delay10ms();
        M24C02_Write(point_addr+0x05, lon_s);
        status += 0x10; //记录坐标的单元+1
        point_addr = ((status>>4)&0x0f)*6;
    }
}
//防盗防走失按下K2, 进入距离设置
else if((status&0x0f) == 0x04){
    if(GetKeyAct(enumKey2) == enumKeyPress){
        status = 0x14;
    }
}
//距离测量功能, 按下K2, 记录当前坐标并开始测量
else if((status&0x0f) == 0x08){
    if(GetKeyAct(enumKey2) == enumKeyPress){
        if(status == 0x18){
            status = 0x08;
        }
        else{
            status = 0x18;
            xlon_h=lon_h, xlon_m=lon_m, xlon_s=lon_s, xlat_h=lat_h, xlat_m=lat_m, xlat_s=lat_s; //更新当前坐标为记录坐标
        }
    }
}
}

```

4.4 显示函数

显示函数更新不同模式及状态下的数码管和 led 显示。显示函数每 1s 调用一次，按下按键或导航按键后也会调用该函数，更新显示。Led 固定显示 status 的值。在校时模式下，数码管显示 RTC 时间（每次接收到报文，RTC 时间都将校准）；在定位模式及测距模式的坐标显示状态，数码管将显示经度或纬度。

```

//时间校准显示
if((status&0x0f) == 0x01){
    time = RTC_Read();
    Seg7Print(time.hour/16, time.hour%16, 12, time.minute/16, time.minute%16, 12, time.second/16, time.second%16);
}
//经纬度显示
else if((status&0x0f) == 0x02 || status == 0x08){
    if(lon_lat == 0){
        Seg7Print(10, lon_h/100, (lon_h/10)%10, lon_h%10+20, lon_m/10, lon_m%10+20, (int)lon_s/10, (int)lon_s%10);
    }
    else{
        Seg7Print(10, 10, lat_h/10, lat_h%10+20, lat_m/10, lat_m%10+20, (int)lat_s/10, (int)lat_s%10);
    }
}
}

```

防盗防走失模式下，初始状态显示 ON/OFF，按下 Key2 后将显示可修改的距离值（有小数点的位为修改位），按下 Key3 后将显示可修改的密码值，若处于报警状态，将显示尝试输入的密码。

```
//防盗防走失模式的值显示
else if((status&0x0f) == 0x04){
    if(status == 0x04){
        if(lock_flag == 0)
            Seg7Print(10,10,10,10,10,10,0,30,30);
        else
            Seg7Print(10,10,10,10,10,10,0,31);
    }
    else if(status == 0x14){
        Seg7Print(10,10,10,10,(dis_temp/1000)+((weixuan&0x08)>>3)*20,(dis_temp/100)+((weixuan&0x04)>>2)*20,(dis_temp/10)+((weixuan&0x02)>>1)*20,dis_temp%10+(weixuan&0x01)*20);
    }
    else if(status == 0x24){
        Seg7Print(10,10,10,10,(code_temp/1000)+((weixuan&0x08)>>3)*20,(code_temp/100)+((weixuan&0x04)>>2)*20,(code_temp/10)+((weixuan&0x02)>>1)*20,code_temp%10+(weixuan&0x01)*20);
    }
    else if(status == 0x44){
        Seg7Print(10,10,10,10,(trycode/1000)+((weixuan&0x08)>>3)*20,(trycode/100)+((weixuan&0x04)>>2)*20,(trycode/10)+((weixuan&0x02)>>1)*20,trycode%10+(weixuan&0x01)*20);
    }
}
```

测距模式的距离显示状态下，将显示计算出的距离。

```
//测距显示
else if(status == 0x18){
    Seg7Print((dis/1000000)%10,(dis/100000)%10,(dis/10000)%10,(dis/1000)%10,(dis/100)%10,(dis/10)%10,dis%10);
}
```

警报状态下蜂鸣器的警报也在显示函数中设置，由于显示函数每秒调用一次，若处于警报状态则蜂鸣器将持续产生蜂鸣。

```
if(status == 0x44)
    SetBeep(1000, 20);
LedPrint(status);
```

4.5 导航按键回调

导航按键回调对各个模式及状态下导航按键的事件进行处理。

校时模式下，导航按键无反应。

定位模式下，导航按键向左或向右切换经纬显示，向上向下则对存储坐标的单元值进行修改，按下 Key3 则从选定单元读取非易失性存储中的坐标，并通过串口 1 发送到计算机。

```
//定位功能下使用导航按键
if((status&0x0f) == 0x02){
    if(GetAdcNavAct(enumAdcNavKeyRight)==enumKeyPress || GetAdcNavAct(enumAdcNavKeyLeft)==enumKeyPress){
        lon_lat=!lon_lat;
    }
    else if(GetAdcNavAct(enumAdcNavKeyUp)==enumKeyPress){
        status += 0x10;
        point_addr = ((status>>4)&0x0f)*6;
    }
    else if(GetAdcNavAct(enumAdcNavKeyDown)==enumKeyPress){
        status = ((status&0xf0) - 0x10) | (status&0x0f);
        point_addr = ((status>>4)&0x0f)*6;
    }
    else if(GetAdcNavAct(enumAdcNavKey3) == enumKeyPress){
        i = M24C02_Read(point_addr);
        send_buf[0] = i/10+'0', send_buf[1] = i%10+'0', send_buf[2] = '.';
        i = M24C02_Read(point_addr+1);
        send_buf[3] = i/10+'0', send_buf[4] = i%10+'0', send_buf[5] = '.';
        i = M24C02_Read(point_addr+2);
        send_buf[6] = i/10+'0', send_buf[7] = i%10+'0', send_buf[8] = '.';
        i = M24C02_Read(point_addr+3);
        send_buf[9] = i/100+'0', send_buf[10] = (i/10)%10+'0', send_buf[11] = (i%10)+'0', send_buf[12] = '.';
        i = M24C02_Read(point_addr+4);
        send_buf[13] = i/10+'0', send_buf[14] = i%10+'0', send_buf[15] = '.';
        i = M24C02_Read(point_addr+5);
        send_buf[16] = i/10+'0', send_buf[17] = i%10+'0', send_buf[18] = '.';
        Uart1Print(send_buf,19);
    }
}
```

防盗防走失模式下，若处于 ON 显示状态，按下导航按键将打开该功能，将标志位写入存储器（保证该功能重启仍有效），并将记录坐标更新为最新坐标；若处于 OFF 显示状态，将状态修改为警报状态，输入设定好的密码后关闭。

```

//防盗防走失功能下使用导航按键
else if((status&0x0f) == 0x04) {
    //开关显示
    if(status == 0x04) {
        if(GetAdcNavAct(enumAdcNavKeyCenter) == enumKeyPress) {
            if(lock_flag == 0x00) { //打开
                lock_flag=0x01;
                M24C02_Write(0x80, lock_flag);
                xlon_h=lon_h, xlon_m=lon_m, xlon_s=lon_s, xlat_h=lat_h, xlat_m=lat_m, xlat_s=lat_s;
                trycode = 0;
            }
            else //关闭需要密码, 进入密码输入状态
                status = 0x44;
        }
    }
}

```

该模式下如果按下 Key3, 则将进入密码设置状态。

```

//进入密码设置
if(GetAdcNavAct(enumAdcNavKey3) == enumKeyPress)
    status = 0x24;

```

该模式下在距离设置状态, 导航按键向左向右选择数值的位, 向上向下修改值, 按下导航按键将安全距离值更新为用数码管选出的值, 并将其写入非易失性存储 (0x81)。

```

//距离设置
if(status == 0x14) {
    if(GetAdcNavAct(enumAdcNavKeyLeft)==enumKeyPress) {
        weixuan = (weixuan<<1)&0x0f;
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyRight)==enumKeyPress) {
        weixuan = (weixuan>>1);
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyUp)==enumKeyPress) {
        dis_temp += (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(dis_temp>9999)
            dis_temp = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyDown)==enumKeyPress) {
        dis_temp -= (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(dis_temp<0)
            dis_temp = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyCenter)==enumKeyPress) {
        safety_dis = dis_temp;
        M24C02_Write(0x81, safety_dis);
    }
}

```

密码设置与距离设置相同。

```

//密码设置
else if(status == 0x24) {
    if(GetAdcNavAct(enumAdcNavKeyLeft)==enumKeyPress) {
        weixuan = (weixuan<<1)&0x0f;
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyRight)==enumKeyPress) {
        weixuan = (weixuan>>1);
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyUp)==enumKeyPress) {
        code_temp += (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(code_temp>9999)
            code_temp = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyDown)==enumKeyPress) {
        code_temp -= (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(code_temp<0)
            code_temp = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyCenter)==enumKeyPress) {
        lockcode = code_temp;
        M24C02_Write(0x85, lockcode);
    }
}

```


警报状态下通过导航按键输入密码解除警报, 与密码设置和距离设置基本相同, 只有按下导航按键的处理不同。在警报状态下按下导航按键将判断选择的密码是否与设定的密码相同, 若相同则会关闭警报, 回到显示 OFF 的状态。

```
//输入密码
else if(status == 0x44) {
    if(GetAdcNavAct(enumAdcNavKeyLeft)==enumKeyPress) {
        weixuan = (weixuan<<1)&0x0f;
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyRight)==enumKeyPress) {
        weixuan = (weixuan>>1);
        if(weixuan == 0)
            weixuan = 1;
    }
    else if(GetAdcNavAct(enumAdcNavKeyUp)==enumKeyPress) {
        trycode += (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(trycode>9999)
            trycode = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyDown)==enumKeyPress) {
        trycode -= (weixuan&0x08)*125+(weixuan&0x04)*25+(weixuan&0x02)*5+(weixuan&0x01);
        if(trycode<0)
            trycode = 0;
    }
    else if(GetAdcNavAct(enumAdcNavKeyCenter)==enumKeyPress) {
        if(trycode == lockcode) {
            lock_flag = 0;
            M24C02_Write(0x80, lock_flag);
            status = 0x04;
            trycode = 0;
        }
    }
}
}
```

测距模式下处于坐标显示状态, 则导航按键左右可切换经纬度显示。导航按键事件处理结束后, 调用显示函数更新数码管和 led 的显示。

```
//测距功能使用导航按键
else if(status == 0x08) {
    if(GetAdcNavAct(enumAdcNavKeyRight)==enumKeyPress || GetAdcNavAct(enumAdcNavKeyLeft)==enumKeyPress)
        lon_lat=!lon_lat;
}
//按下导航按键后更新led与数码管的显示
display();
```

4.6 距离计算

通过经纬度计算两坐标点的距离, 根据资料, 使用 Haversine 公式进行计算。在功能测试中会对计算精度进行分析, 此处只列出计算方法。

- d: 两点间的距离
- r: 球的半径
- $\theta_1\theta_2$: 点 1 的纬度和点 2 的纬度, 以弧度制度量
- $\lambda_1\lambda_2$: 点 1 的经度和点 2 的经度, 以弧度制度量

Haversine 公式如下:

$$d = 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right)$$

$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

根据公式，距离计算函数如下：

```
//距离计算
int cal_dis() { //计算距离，单位为10m
    xdata double X1, Y1, X2, Y2, res;
    xdata double a;
    xdata double b;
    X1 = radian(xlat_h + ((float)xlat_m)/100.0 + ((float)xlat_s)/10000.0);
    X2 = radian(lat_h + ((float)lat_m)/100.0 + ((float)lat_s)/10000.0);
    Y1 = radian(xlon_h + ((float)xlom_m)/100.0 + ((float)xlom_s)/10000.0);
    Y2 = radian(lon_h + ((float)lon_m)/100.0 + ((float)lon_s)/10000.0);
    a = X1-X2;
    b = Y1-Y2;
    res = 2 * EARTH_RADIUS * asin((sqrt(pow(sin(a / 2), 2) + cos(X1) * cos(X2) * pow(sin(b / 2), 2) )));
    return (int)(res/10);
}
//角度转弧度
double radian(float angle) {
    return angle*PI/180;
}
```

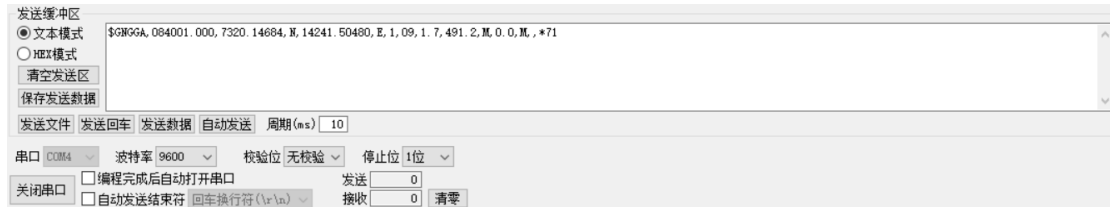
第三章 功能测试

以下测试均使用串口1向STC发送定位数据。

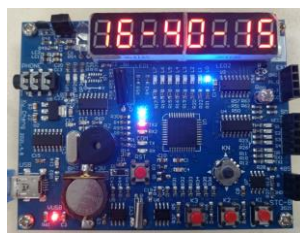
1.时间校准测试

通过串口发送定位信息，包含UTC时间。测试时间为16:40，由于定位信息中包含的是UTC时间，因此向串口发送的时间信息为084001.000。其他信息取任意值，发送后，数码管显示的时间被校准为当前时间16:40。

发送数据：



STC 的数码管显示：

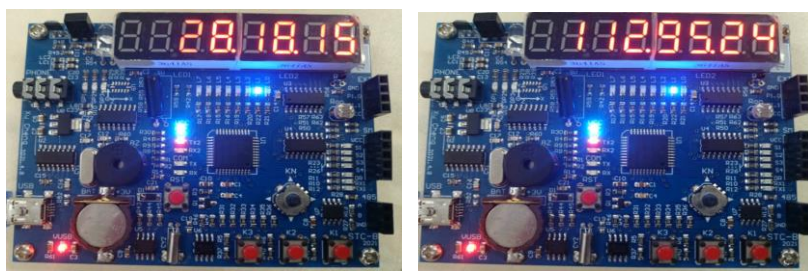


2.定位测试

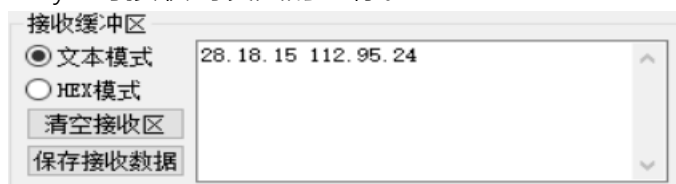
按下 Key1 切换到定位模式，通过串口发送坐标 112.954062,28.182611，即以下报文：

\$GNGGA,084001.000,2818.26110,N,11295.40620,E,1,09,1.7,491.2,M,0.0,M,,*71

发送后数码管可显示经纬度，并可通过导航按键左右切换经纬度显示。



通过导航按键上下可选择存储单元，选择 8 号存储单元，按下 Key2 存储当前坐标，再按下 Key3 可接收到发回的坐标。



3.防走失/防盗功能测试

选定坐标点（11295.14000,2817.34000）天马学生公寓和坐标点（11295.19000,2818.63000）体育场进行测试。这两个坐标点在测距功能测试时测距为 1.3km。先发送以下定位信息：

```
$GNGGA,084001.000,2817.34000,N,11295.14000,E,1,09,1.7,491.2,M,0.0,M,,*71
```

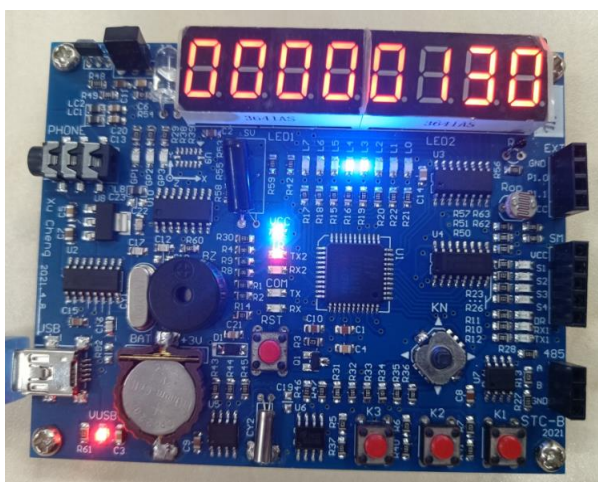
切换到防盗防走失模式，按下 Key2 设置安全距离为 129 (1.29km)，按下 Key3 设置密码为 0001,分别按下导航按键确认。确认后按 Key1 切换至显示 OFF，再按下导航按键。最后发送以下定位信息：

```
$GNGGA,084001.000,2818.63000,N,11295.19000,E,1,09,1.7,491.2,M,0.0,M,,*71
```

发送后蜂鸣器响起，通过导航按键选择密码 0001，按下导航按键后蜂鸣器停止发声，数码管显示 OFF。

4.测距功能测试

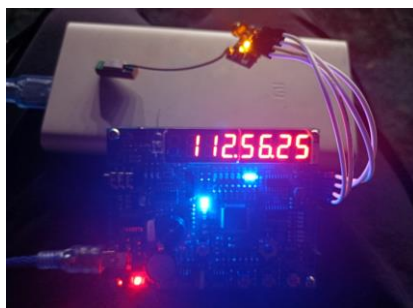
选定坐标点（11295.14,2817.34000）天马学生公寓和坐标点（11295.19,2818.63000）体育场进行测试。按下 Key1 切换到测距模式下，先从串口发送第一个坐标，按下 Key2 开始测距，然后发送第二个坐标，此时数码管显示两点距离为 130，单位为 10m，即距离为 1.3km。打开手机地图，两点步行距离为 1.5km。



对结果进行分析。经纬度计算距离为直线距离，因此小于步行距离，此外，定位信息中经纬度单位为分，小数点后 5 位经计算得到秒，在计算时只使用了三位进行计算，因此可能有一定误差。定位本身也存在一定误差。因此这个测距功能存在误差，在直线距离上保证大致准确，具体的精度需要进行进一步的测算。

5.户外实测

将定位模块连接到 STC 的 ext 接口，在户外(荫马塘体育场)等待定位模块启动，按下 Key2 后时间被校准，收到了卫星的定位信息，切换到定位模式，数码管显示了经纬度：



这个坐标和手机中该位置的坐标不同，这是因为不同地图使用了不同的坐标系，如百度地图使用了 BD-09 坐标系，而腾讯地图使用的是 GCJ-02 坐标系，因相同地点在不同地图上的经纬度不同。该定位器的测距功能，放盗防走失功能在同一个坐标系下计算距离，因此不受此影响，定位功能可能需要在不同地图上进行校准（定位器应添加一个校准坐标的功能）。

第四章 总结

本设计的目标是实现一个定位信息实际应用相关的定位器，最终实现的定位器实现了基本功能。但由于处理报文时保留的精度，以及计算距离的公式（使用了球体公式而不是椭圆体公式），定位本身精度不足，实际测距精度不确定，只

能保证大致准确。因时间有限,没有尝试使用 ESP8266 连接网络传输定位数据,因此功能还不完整。

通过完成本次创新设计作业,熟悉了 BSP 库的使用,实现了用 STC 实现一个具有基本的校时,定位,测距等功能的定位器。进一步加强了相关的设计,调试,测试等基本方法和技能。也在实现过程中熟悉了 STC 学习板各个器件的用途和应用方法,能够在学习板上,以 BSP 库为基础开发实现一些具体的功能。