

作业题目是“嵌入式GPS轨迹记录仪”

需要实现的基本功能是：由GPS模块接收位置数据，然后将坐标点画在LCD屏幕上，移动过程中LCD循环画点形成轨迹线。

本次任务是：

1. **GPS模块代码移植**：GPS模块资料包的代码是在STM32F103平台编写的，本次作业需要用到LCD屏，目前只有一块F4的板子可以使用LCD屏幕，所以需要将GPS的代码移植到STM32F429平台。（仅需修改main.c文件和usart.c文件，修改后的文件单独上传）。
2. **轨迹记录**：编写一个函数来记录轨迹点。每当接收到一个新的GPS位置时，将其存储在STM32的内部闪存或外部存储器中。
3. **数据显示**：编写一个函数来将解析后的GPS数据显示在LCD屏上。

移植后的GPS源码：

```
1 void errorLog(int num)
2 {
3     while (1)
4     {
5         printf("ERROR%d\r\n", num);
6     }
7 }
```

```
1 void parseGpsBuffer()
2 {
3     char *subString;
4     char *subStringNext;
5     char i = 0;
6     if (Save_Data.isGetData)
7     {
8         Save_Data.isGetData = false;
9         printf("*****\r\n");
10        printf(Save_Data.GPS_Buffer);
11
12        for (i = 0 ; i <= 6 ; i++)
13        {
14            if (i == 0)
```

```

15         if ((subString = strstr(Save_Data.GPS_Buffer, ",")) ==
NULL)
16             errorLog(1); //‰ÿŸ
17     }
18     else
19     {
20         subString++;
21         if ((subStringNext = strstr(subString, ",")) != NULL)
22         {
23             char usefullBuffer[2];
24             switch(i)
25             {
26                 case 1:memcpy(Save_Data.UTCtime,
subString, subStringNext - subString);break;
27                 case 2:memcpy(usefullBuffer, subString,
subStringNext - subString);break;
28                 case 3:memcpy(Save_Data.latitude,
subString, subStringNext - subString);break;
29                 case 4:memcpy(Save_Data.N_S, subString,
subStringNext - subString);break;
30                 case 5:memcpy(Save_Data.longitude,
subString, subStringNext - subString);break;
31                 case 6:memcpy(Save_Data.E_W, subString,
subStringNext - subString);break;
32                 default:break;
33             }
34             subString = subStringNext;
35             Save_Data.isParseData = true;
36             if(usefullBuffer[0] == 'A')
37                 Save_Data.isUsefull = true;
38             else if(usefullBuffer[0] == 'V')
39                 Save_Data.isUsefull = false;
40
41         }
42     else
43     {
44         errorLog(2); //‰ÿŸ
45     }
46 }
47 }

```

```
48     }  
49 }  
50
```

```
1  void printGpsBuffer()  
2  {  
3      if (Save_Data.isParseData)  
4      {  
5          Save_Data.isParseData = false;  
  
6          printf("Save_Data.UTCTime = ");  
7          printf(Save_Data.UTCTime);  
8          printf("\r\n");  
  
9          if(Save_Data.isUsefull)  
10         {  
11             Save_Data.isUsefull = false;  
12             printf("Save_Data.latitude = ");  
13             printf(Save_Data.latitude);  
14             printf("\r\n");  
15  
16             printf("Save_Data.N_S = ");  
17             printf(Save_Data.N_S);  
18             printf("\r\n");  
  
19             printf("Save_Data.longitude = ");  
20             printf(Save_Data.longitude);  
21             printf("\r\n");  
  
22             printf("Save_Data.E_W = ");  
23             printf(Save_Data.E_W);  
24             printf("\r\n");  
25         }  
26         else  
27         {  
28             printf("GPS DATA is not usefull!\r\n");
```

```
29         }

30     }

31 }
```

关于**LCD屏幕画运动轨迹**，目前暂且简化为：初始化时，先在屏幕上画一个**直角坐标系**，屏幕像素的（240，400）为坐标原点，在主循环函数里用**static**将**第一次**循环中接收到的经纬度坐标保存下来，然后将后续接收到的所有坐标数据都减去第一次保存的坐标，然后在LCD屏幕上循环画点形成运动轨迹（每一次的x坐标都加240，y坐标都加400，以**保证运动轨迹从坐标原点出发**）

```
1  int main(void)
2  {
3      uint8_t lcd_id[12];
4      sys_stm32_clock_init(360, 25, 2, 8);
5      delay_init(180);
6      led_init();
7      sdram_init();
8      lcd_init();
9      sprintf((char *)lcd_id, "LCD ID:%04X", lcddev.id);
10     lcd_clear(WHITE);
11
12     HAL_Init();
13     sys_stm32_clock_init(360, 25, 2, 8);
14     delay_init(180);
15     usart_init(9600);
16     led_init();
17
18     clrStruct();
19
20     lcd_draw_line(239, 0, 239, 799, BLACK);
21
22     lcd_draw_line(0, 399, 479, 399, BLACK);
23
24     lcd_draw_line(239, 0, 229, 10, BLACK);
```

```

23         lcd_draw_line(239, 0, 249, 10, BLACK);

24         lcd_draw_line(479, 399, 469, 389, BLACK);

25         lcd_draw_line(479, 399, 469, 409, BLACK);


26     while (1)
27     {
28         parseGpsBuffer();
29         printGpsBuffer();
30         if(aaa == 1)
31         {
32             LED0_TOGGLE();
33             aaa = 0;
34         }
35         // delay_ms(1000);
36         if(Save_Data.isUsefull)
37         {
38             static int x0 = (const int)Save_Data.latitude;
39             static int y0 = (const int)Save_Data.longitude;
40             int x1 = (int)Save_Data.latitude;
41             int y1 = (int)Save_Data.longitude;

42             lcd_draw_point(x1-x0+240, y1-y0+400, BLUE);


43         }
44     }
45 }

```

测试RGB显示效果：

