全方位平面定位系统产品手册

OPS-9

沈阳艾克申机器人技术开发有限责任公司

2021-06

关于本文档

版权声明

历史修订

版本号	修改日期	摘要	撰稿人	审核
V1.1	2016.08		Billy	3000KG
V1.2	2017.03		Billy	3000KG
V1.3	2017.11	增加清零等命令	Nillong Maxwell	3000KG
V1.5	2018.03	改版、更新参数	Billy	3000KG
V2.0	2021.06	改版、增加示例	CZR	3000KG
V3.3	2021.10	改版、增加状态指示灯	LD	3000KG

1、简介

全方位平面定位系统集成了陀螺仪、加速度计、编码器等多种传感器，通过滤波、融合等方法，解算出定位系统中心在其坐标系中的位姿信息。正交全方位轮的浮动设计，保证了里程测量的准确性，解决了因打滑导致的测量不准问题。本模块采用 5V 供电，通过 RS232 串口进行配置以及数据输出。

该产品电子器件全部采用国际进口品牌，PCB印刷、SMT由嘉立创代工，质量有保障。

机械机构原材料选用金属铝，质轻，通过精密机床车铣工艺成型，表面采用阳极氧化喷砂处理，实现科技感十足的亮黑色。

目前，全方位平面定位系统已经被百余所院校、研究所以及 AGV 公司采用。

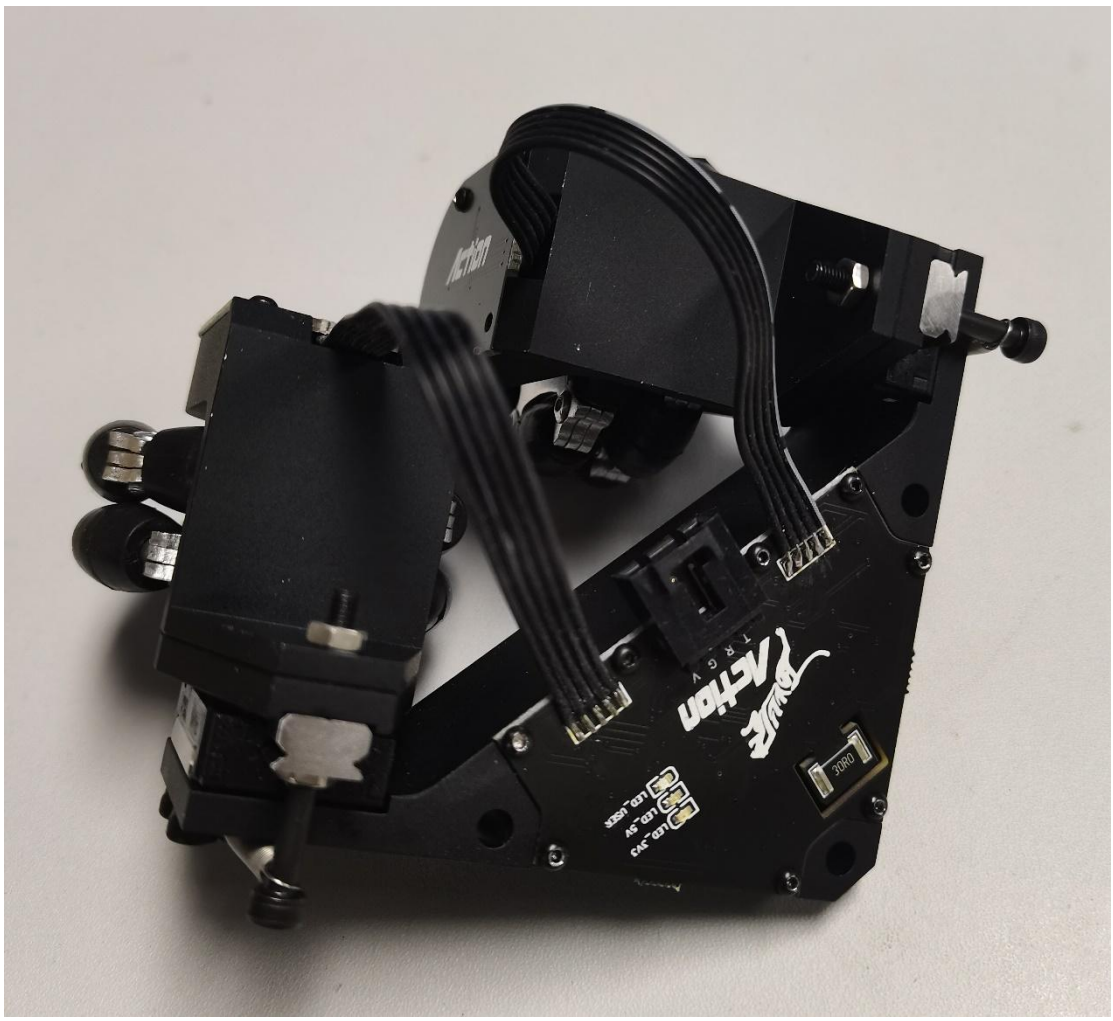


图1 定位系统近景

2、性能参数

表格1 性能参数

	条件	最小值	典型值	最大值	单位
供电电压		4.5	5	5.5	V
工作电流			0.1	0.3	A
上电后静止时间			15		s
波特率			115200		bps
数据帧率			200		fps
角速度量程			±500		dps
角速度分辨率（Z）	25 摄氏度		0.15		dps
里程计分辨率			0.03896		mm
输出角度范围			±180		度
角度静态累积误差	25 摄氏度		10		度/小时
坐标误差	直线前进 5m		±30		mm
连续工作测试	静态		48		小时
	动态		12		小时
轮径磨损测试		0.08%	0.1%	0.14%	/百公里
越障碍（沟槽）能力	沟槽宽度			10	mm
	越障次数	100000			次
环境温度范围		0	25	40	℃

3、功能介绍

3.1 硬件接口

表格2 接口信号

名称	描述
VCC	5V 电源
GND	5V 地
RS-232 TxD	RS-232 发送信号
RS-232 RxD	RS-232 接收信号

3.2 指示灯

全方位平面定位系统板载3个状态指示灯，用来指示系统状态。

LED_5V 为 5V 电源指示灯，此指示灯发光（绿色）表示外接 5V 电源正常；

LED_3V3 为板载 3.3v 稳压电路输出指示灯，此指示灯亮（红色）表示 3.3v 稳压输出正常；

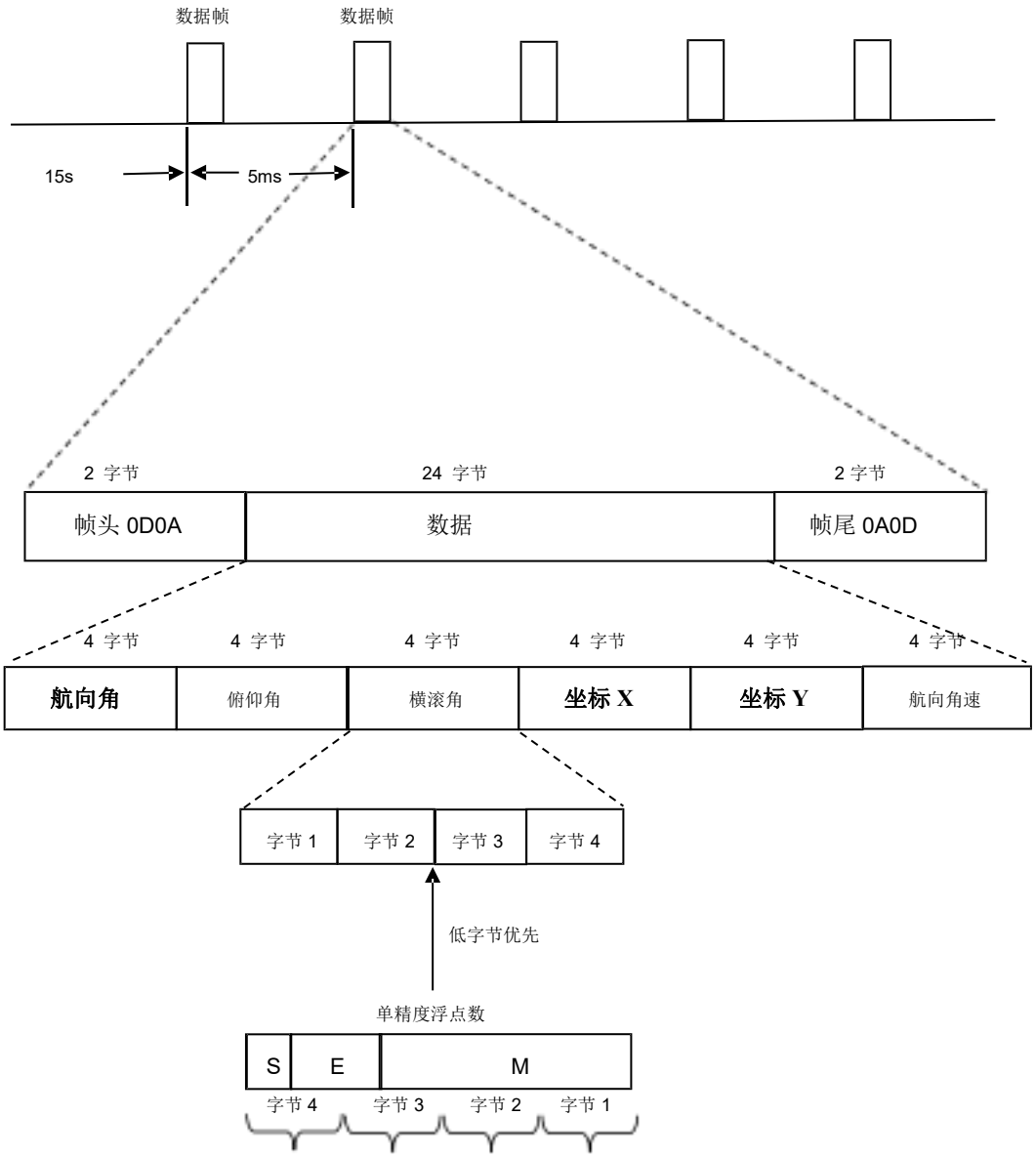
LED_USER 为新版定位系统加入的系统状态指示灯（蓝色），系统上电大约 3 秒后，此指示灯闪烁，表示系统正在初始化，此过程需要保持定位系统完全静止!! 当指示灯停止闪烁，代表定位系统初始化完成，大约 3 秒后，开始正常输出数据。

3.3 通讯协议

表格3 协议参数

UART 通信协议	
波特率	115200bps
起始位	1 位
数据位	8 位
停止位	1 位
校验位	0 位

3.4 数据帧格式



系统以数据包的形式发送数据，每个数据包中包含 28 个字节，其中包头 2 个字节，包尾 2 个字节。中间的 24 字节为数据，以 4 个字节为单位（LSB），分别为三轴角度值，两个坐标值，和一个角速度值。这些数据的存储方式为单精度浮点数（754 标准，即浮点数的二进制表示，S 为符号位，E 为阶码，M 为尾数），在 C 语言里为 float 型数据，解析请参考附录例程。

3.5 常用命令

3.5.1 校准

本产品具有较好的零漂性能，但其性能的保持需要每隔一段时间校准一次，建议在环境温度发生一定变化时（如季节变换），或定位系统角度信息发生偏差时校准一次，具体校准方法如下:**（注意：本产品在发货前都已校准完成，买家无需再自己进行校准，如有问题需要校准，请先联系技术售后）**

在供电正常工作的情况下使用串口向该模块发送字符串“ACTR”，校准开始；首先会返回类似“start2653”的数据，表示现在是从 26 度 53 开始校准的，建议从 25 度到 27 度之间开始校准，继续保持供电，大概十五分钟后，串口会返回“check”表示校准结束。校准过程中请保持该模块处于绝对静止状态中。

3.5.2 清零

在使用该模块过程中，如用户借助其它传感器可确定零点或使用环境中存在固定位置可标记零点，可通过向该模块发送“ACTO”对其进行清零操作。发送后，该模块输出数据如角度，坐标值将全部变为零，并在此刻重新开始数据计算融合。用户可在一次使用中多次发送字符串“ACTO”对其进行位置校正。

3.5.3 更新航向角

在使用该模块过程中，如用户借助其它传感器可确定固定可标记位置或使用环境中存在固定位置可进行标定，可使用此功能校准该模块角度与坐标。具体标定模式如下：

若需要标定 Z 轴(航向角)角度信息，可通过串口向定位系统发送“ACTJ****”（例如根据其它传感器信息或某一标记位置可确定某一时刻该模块输出应为 90 度，可设置变量 angle，angle 必须为 float 型数据，且为 90.0 度，可发送“ACTJangle”，注意不可直接发送“ACTJ90.0”，且通过串口发送数据时需将数据按字节发送，可设置共用体）

3.5.4 更新 X 坐标

若需要标定 X 轴坐标信息，可通过串口向定位系统发送“ACTX****”（例如根据其它传感器信息或某一标记位置可确定某一时刻该模块 X 轴位置输出应为 1000 毫米，可设置变量 position_x，position_x 必须为 float 型数据，且为 1000.0，可发送“ACTXposition_x”，注意不可直接发送“ACTX1000.0”，且通过串口发送数据时需将数据按字节发送，可设置共用体）

3.5.5 更新 Y 坐标

若需要标定 Y 轴坐标信息，可通过串口向定位系统发送“ACTY****”（例如根据其他传感器信息或某一标记位置可确定某一时刻该模块 Y 轴位置输出应为 1000 毫米，可设置变量 position_y，position_y 必须为 float 型数据，且为 1000.0，可发送“ACTYposition_y”，注意不可直接发送“ACTY1000.0”，且通过串口发送数据时需将数据按字节发送，可设置共用体）

3.4.3、3.4.4 和 3.4.5 的发送命令示例如下图所示：

```
/*字符串拼接*/
void Stract(char strDestination[],char strSource[],int num)
{
    int i = 0,j = 0;

    while(strDestination[i]!='\0')  i++;

    for(j = 0;j < num;j++)
        strDestination[i++] = strSource[j];
}
```

```
/*更新X坐标*/
void Update_X(float New_X)
{
    int i = 0;
    char Update_x[8] = "ACTX";

    static union
    {
        float X;
        char data[4];
    }New_set;

    New_set.X = New_X;
    Stract(Update_x,New_set.data,4);

    for(i = 0;i < 8;i++)
    {
        while(USART_GetFlagStatus(USART3, USART_FLAG_TXE) == RESET);
        USART_SendData(USART3,Update_x[i]);
    }
}
```

上图是更新 X 坐标的参考函数，更新 Y 坐标和更新航向角的函数只需要将其中的 Update_x[8] = "ACTX"对应改成 update_y[8] = "ACTY"和 update_A[8] = "ACTJ"即可。调用方式如下：

```
Update_X(1000.0);
delay_ms(10);
Update_Y(2000.0);
delay_ms(10);
Update_A(90);
```

为保持稳定的进行更新坐标，每次发送更新坐标命令之间要间隔 10ms。

4 安装

4.1 坐标系定义

定位系统坐标系被定义为一个绝对坐标系，定位系统一旦被安装到机器人上，在定位系统上电后，这个坐标系就确定了。该坐标系可以和用户定义的绝对坐标系不一致。由于定位系统支持实时更新角度以及坐标，因此每次数据更新，定位系统坐标系会随之改变。

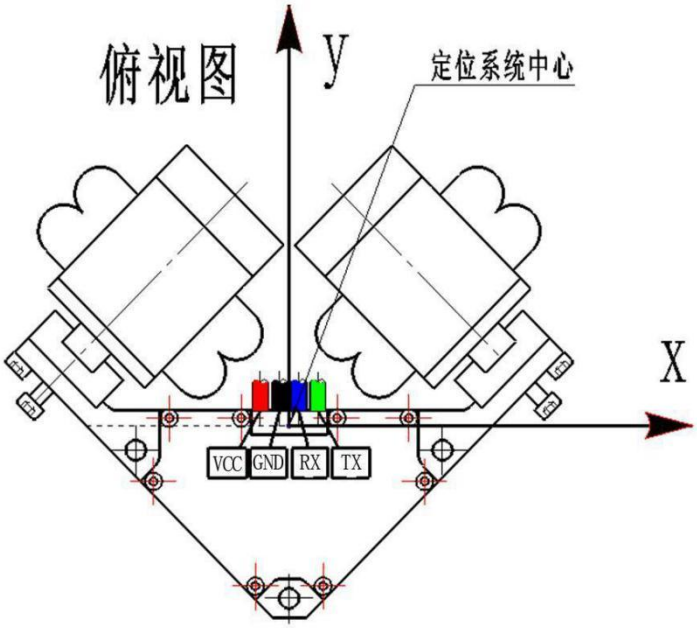


图2 定位系统坐标系定义

4.2 装配

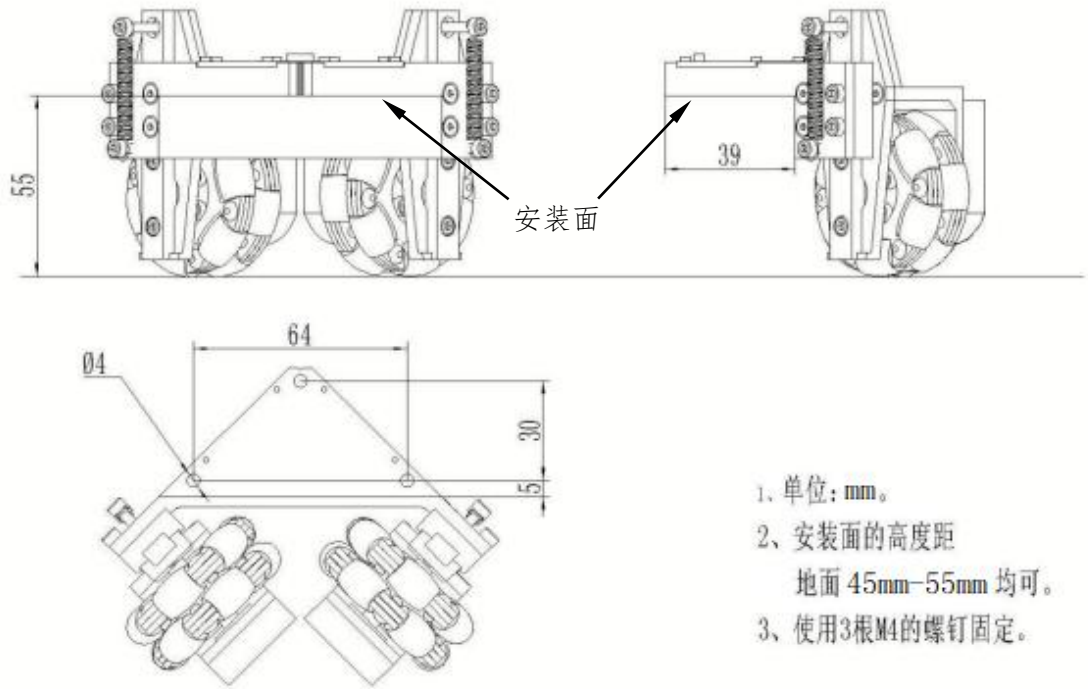


图3 装配图

该定位系统属于高精度传感设备，安装位置的准确度与输出的角度坐标信息紧密相关，如果实际安装位置与理论安装位置不符，会造成较大偏差，故请严格按照以下建议安装。

（1）定位系统具有广泛的适用性，可以安装在移动单元的任何位置，通过公式将定位系统的坐标角度信息转换为整体的坐标角度信息，为了保证更好的性能。优先推荐定位系统中心与移动单元几何中心重合，Y 轴方向为移动底盘正方向；其次推荐定位系统中心在移动单元的中轴线上；最后安装在底盘的任意位置，但该位置与移动单元中心的相对距离及角度必须确定。

（2）定位系统本身具有弹簧悬挂，可以保证编码器轮时刻与地面接触。浮动距离为 30mm，安装面距离地面的高度为 45mm-55mm 之间，为了达到最佳的浮动效果，推荐安装面距离地面 45mm。

（3）螺钉仅作为连接使用，为了保证多次安装的一致性，建议在设备安装位置加卡位设计。

（4）远离振动源。

（5）水平安装。

5、使用教程

5.1 在电脑上获取定位系统数据

该模块支持定位数据的直接显示，定位系统输出数据为 RS232 电平，可通过 RS232 转 USB 串口线将定位系统与电脑 USB 端口连接，RS232（公头）接口定义和定位系统输出接口如下图所示。需保证“RXD 接 TX，TXD 接 RX，GND 与定位系统的 GND 共地，定位系统 VCC 之间接足够功率的 5V 直流电源”。之后打开 定位系统调试软件，按使用步骤进行测试。

注意：定位系统的供电端一定要和数据端保持共地。

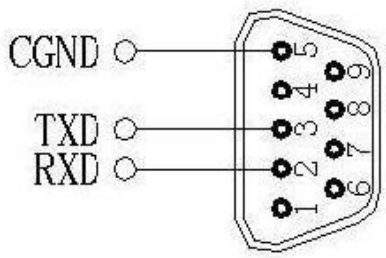


图4 DB9公头定义参考

图5 USB转rs232串口线参考

图6 定位系统调试软件

定位系统调试软件使用步骤如下：

（1）选择指定串口设备后，点击打开串口。

（2）定位系统初始化大约 15s 后，左侧的“角度”、“X 坐标”、“Y 坐标”文本框会显示定位系统输出的位姿信息。注意定位系统初始化 15s 时间内，需保证定位系统处于静止状态，请勿将定位系统置于震动环境中初始化。

（3）“数据清零”按钮用来测试定位系统是否可以进行清零操作，按下该按钮后，左侧的“角度”、“X 坐标”、“Y 坐标”文本框都会变为数值 0。

（4）“修改 X 坐标”、“修改 Y 坐标”、“修改角度”按钮用来测试定位系统是否可以正常进行修改坐标操作，在对应的文本框内写上需要修改的坐标，例如：在修改 X 坐标后面的文本框内输入 1000，点击“修改 X 坐标”，左侧的“X 坐标”文本框就会变为 1000。

5.2 在STM32F4平台上获取定位系统数据

将定位系统 VCC 与 GND 分别接到开发板 5V 与 GND，上（保证开发板电源供电，开发板若只用电脑 USB 供电会功率不足），定位系统 TX 接开发板 RS232 端 RX（切记不可接 TTL 引脚输出端），RX 接开发板 RS232 端 TX，保证硬件连接无误后，编写接收程序，并根据程序显示定位数据信息（可使用串口输出函数通过串口调试助手观察，或通过在线调试窗口观察数据）。参考程序可见附录部分。

6、故障分析与排除

6.1 接收数据乱码

◎ 请确认使用了 RS-232 进行数据通信

◎ 请确认定位系统和对端通信设备共地

◎ 请确认波特率设置为 115200，1 个停止位，无奇偶校验

◎ 第三方通信转换模块质量问题

6.2 角度漂移严重

◎ 请确保定位系统在上电初始化过程中保持静止

◎ 请确保初始化时间够长

◎ 请确保定位系统在校准过程中保持静止

◎ 请确保定位系统处于无风、环境温度稳定的工作环境

6.3 数据输出异常

◎ 都是0表示：校准开始不久就断电了，需要重新校准

◎ 都是NAN表示：校准失败，需要重新校准

6.4 角度不准

◎ 角速度超出陀螺仪角速度范围是±500dps，尤其是控制不好的快速震荡以及震动

◎ 使用环境中，温度发生严重变化，建议重新校准

◎ 机架件因意外撞击变形

6.5 坐标不准

◎ 装配误差大

◎ 未安装到机器人中心，并且未进行坐标转换

◎ 地面平坦度差

◎ 机架件因意外撞击变形

6.6 重启

◎ 电源电压不稳

◎ 静电干扰

6.7 无输出

◎ 电源驱动器能力差，导致定位系统无法启动，比如用普通 IO 供电

◎ 主控板和编码器小板之间连线产生断裂，导致无法读取编码器数据

◎ 静电干扰

6.8 数据异常跳动

◎ 数据处理不及时，发生数据覆盖，需提高处理性能或者增加缓冲区

7、注意事项

◎ 输入电源要稳定，达到要求

◎ 装配尽量准确

◎ 校准过程尽量保持静止，时间够长，环境稳定

◎ 初始化过程时间要足够长，并且保持静止

◎ 外壳尽量接地，或者接机器人机壳

◎ 使用 RS-232 进行数据通信

◎ 主控板和编码器小板之间的连线无法承受过大的力

◎ 防止水淋

8、固件升级

艾克申平面定位系统支持固件升级（部分早期版本不支持），使用固件升级软件即可通过 RS-232 接口完成固件升级。操作步骤如下：

（1）在未通电的情况下，将定位系统通过 USB 转 RS-232 连接到电脑

（2）打开串口并选择产品-双轮定位系统

（3）点击准备升级按钮，再点击设备上电按钮，

（4）然后再给定位系统上电，标题会出现“已连接xxxx”字样

（5）选择固件，固件为二进制文件，由艾克申公司发布

（6）点击开始升级按钮，待进度条完成后，断电，关闭软件

图7 固件升级上位机软件

9、联系我们

感谢您使用我们的产品，如有需要，请联系技术支持。

QQ 群：772109542

代码附录（STM32F407）

USART 配置程序

```
1.  void USART1_Init(int baudrate)
2.  {
3.      GPIO_InitTypeDef GPIO_InitStructure;
4.      USART_InitTypeDef USART_InitStructure;
5.      NVIC_InitTypeDef NVIC_InitStructure;
6.
7.      RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA,ENABLE);
8.      RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,ENABLE);
9.
10.     GPIO_PinAFConfig(GPIOA,GPIO_PinSource10,GPIO_AF_USART1);
11.     GPIO_PinAFConfig(GPIOA,GPIO_PinSource9, GPIO_AF_USART1);
12.
13.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10 | GPIO_Pin_9;
14.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
15.     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
16.     GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
17.     GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
18.     GPIO_Init(GPIOA,&GPIO_InitStructure);
19.
20.     USART_InitStructure.USART_BaudRate = baudrate;
21.     USART_InitStructure.USART_WordLength = USART_WordLength_8b;
22.     USART_InitStructure.USART_StopBits = USART_StopBits_1;
23.     USART_InitStructure.USART_Parity = USART_Parity_No;
24.     USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowCon
trol_None;
25.     USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
26.     USART_Init(USART1, &USART_InitStructure);
27.
28.     USART_Cmd(USART1, ENABLE);
```

```
29.
30.     USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
31.
32.     NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
33.     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=1;
34.     NVIC_InitStructure.NVIC_IRQChannelSubPriority =1;
35.     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
36.     NVIC_Init(&NVIC_InitStructure);
37. }
```

数据解析程序

```
1. float pos_x=0;
2. float pos_y=0;
3. float zangle=0;
4. float xangle=0;
5. float yangle=0;
6. float w_z=0;
7. void USART1_IRQHandler(void)
8. {
9.   static uint8_t ch;
10.      static union
11.      {
12.       uint8_t data[24];
13.       float ActVal[6];
14.      }posture;
15.      static uint8_t count=0;
16.      static uint8_t i=0;
17.
18.      if(USART_GetITStatus(USART1, USART_FLAG_RXNE)==SET)
19.      {
20.         USART_ClearITPendingBit( USART1, USART_FLAG_RXNE);
21.         ch=USART_ReceiveData(USART1);
22.          switch(count)
23.          {
24.             case 0:
25.                if(ch==0x0d)
26.                   count++;
27.                else
28.                   count=0;
29.                break;
```

```
30.             case 1:
31.                 if(ch==0x0a)
32.                 {
33.                     i=0;
34.                     count++;
35.                 }
36.                 else if(ch==0x0d);
37.                 else
38.                     count=0;
39.                 break;
40.             case 2:
41.                 posture.data[i]=ch;
42.                 i++;
43.                 if(i>=24)
44.                 {
45.                     i=0;
46.                     count++;
47.                 }
48.                 break;
49.             case 3:
50.                 if(ch==0x0a)
51.                     count++;
52.                 else
53.                     count=0;
54.                 break;
55.             case 4:
56.                 if(ch==0x0d)
57.                 {
58.                  zangle=posture.ActVal[0];
59.                  xangle=posture.ActVal[1];
60.                  yangle=posture.ActVal[2];
61.                  pos_x =posture.ActVal[3];
62.                  pos_y =posture.ActVal[4];
63.                  w_z   =posture.ActVal[5];
64.                 }
65.               count=0;
66.                 break;
67.             default:
68.                 count=0;
69.               break;
70.         }
71.     }
72. }
```