

2.2 要求

1. 无人机可按照现场设置的目标位置信息，对 2 个指定位置的目标地点完成送货作业。目标地点位置信息坐标可用无人机上携带的键盘设置。
 - (1) 无人机在“十”字起降点垂直起飞，升空至 $150 \pm 10\text{cm}$ 的巡航高度；（5 分）
 - (2) 根据现场设置的送货目标，先后依次飞行抵达目标地点上方，无人机降低飞行高度到 $80 \pm 10\text{cm}$ ；（16 分）
 - (3) 无人机放出吊舱，吊舱降至距地面 $20 \pm 5\text{cm}$ 高度，并保持稳定悬停 5 秒，完成送货作业，期间播放提醒目标点收货的语音；悬停期间，标识无人机位置的激光笔光斑落在以目标中心为圆心、半径 15cm 的圆内；（20 分）
 - (4) 作业完成后飞行到起降点稳定准确降落，无人机几何中心点与起降点中心距离偏差不大于 $\pm 10\text{cm}$ ；（4 分）
 - (5) 送货过程必须要在 180 秒内完成，用时越少越好。（5 分）
2. 无人机在作业区外学习识别某一种指定目标特征（颜色、形状），然后寻找具有此特征的两个目标地点，完成送货作业。
 - (1) 无人机从起降点起飞到 150cm 巡航高度，先后寻找 2 个上述已识别的目标，飞行抵达目标地点上方，降低飞行高度到 $80 \pm 10\text{cm}$ 左右；（20 分）
 - (2) 放出下降吊舱至距地面 $20 \pm 5\text{cm}$ 左右高度，稳定保持悬停 5 秒完成送货作业，期间播放提醒目标点收货的语音；送货期间，标识无人机位置的激光笔光斑落在以目标中心为圆心、半径 15cm 的圆内；送货完成即恢复巡航高度；（10 分）
 - (3) 送货作业完成后无人机降落到起降点；送货过程用时越少越好，需在 270 秒内完成；（5 分）
3. 无人机找到放置在 A 附近区域的红色圆框，并从圆框中穿越而过。（10 分）
4. 其他自主发挥。（5 分）
5. 设计报告

根据赛题要求，我们可以将需要设计的软件部分拆解为以下三个部分：

- 机器视觉部分：需要识别底部目标的颜色和形状并通过 SDK 串口实现和飞控的数据交互；
- 飞控人机交互界面部分：按键、显示屏配合实现目标航点坐标的录入、学习到的目标特征录入、圆环的坐标与角度信息录入，将设置好的参数存储在飞控 EEPROM 内；
- 无人机自动飞行任务设计：完成自动起飞、根据现场录入或者学习到的航点目标进行作业（包括作业高度调整、自动投放/回收装置、蜂鸣器报警提示等）、执行完毕后自动返航。

2.1 机器视觉部分软件的设计

底部目标特征点的颜色为红色和蓝色，可以通过 OPENMV 机器视觉模组的寻找色块函数 `find_blob()` 对底部目标进行识别，我们可以判断识别到的色块的密度参数即 `blob.density()`，可以很方便的区分出矩形、圆形、三角形。

无名创新

零基础学习竞赛无人机积木式编程指南

```
1 def opv_find_color_blobs():
2     img=sensor.snapshot()
3     target.flag=0
4     target.img_width=IMAGE_WIDTH
5     target.img_height=IMAGE_HEIGHT
6     pixels_max=0
7     type_id=0
8     for b in img.find_blobs(thresholds_r_b,pixels_threshold=30,merge=True,margin=50):
9         if b.code()==1:
10             print('颜色为红色')
11         else:
12             print('颜色为蓝色')
13         #density色块面积和外接矩形的比值
14         if b.density()>0.8:#当检测的对象为矩形时,此值接近1
15             type_id=2
16             img.draw_rectangle(b.rect())
17             print('矩形',b.cx(),b.cy(),b.density(),type_id)
18         elif b.density()>0.65:#当检测的对象为圆形时,此值大于0.65
19             type_id=1
20             img.draw_circle((b.cx(), b.cy(),int((b.w()+b.h())/4)))
21             print('圆形',b.cx(),b.cy(),b.density(),type_id)
22         elif b.density()>0.40:#当检测的对象为三角形时,此值大于0.4
23             type_id=3
24             img.draw_cross(b.cx(), b.cy())
25             print('三角形',b.cx(),b.cy(),b.density(),type_id)
26         else: #忽略掉其它图形或者噪声干扰
27             print("未识别")
28     if pixels_max<b.pixels():
29         pixels_max=b.pixels()
30         target.x = b.cx()
31         target.y = b.cy()
32         target.pixel=pixels_max
33         target.reserved1=b.w()>>8
34         target.reserved2=b.w()
35         target.reserved3=b.code()#颜色
36         target.reserved4=type_id #形状
37         target.flag = 1
```

最后将识别到的色块坐标、色块形状等关键参数填入到目标类中，编码后通过串口发送到飞控 SDK 串口，飞控通过解析到的数据帧就能判断出目标点的形状和颜色用于控制决策。

```
41 |HEADER=[0xFF,0xFC]
42 |MODE=[0xF1,0xF2,0xF3]
43 |#
44 |def package_blobs_data(mode):
45 |    #数据打包封装
46 |    data=bytearray([HEADER[0],HEADER[1],0xA0+mode,0x00,
47 |                    target.x>>8,target.x, #将整形数据拆分成两个8位
48 |                    target.y>>8,target.y, #将整形数据拆分成两个8位
49 |                    target.pixel>>8,target.pixel, #将整形数据拆分成两个8位
50 |                    target.flag, #数据有效标志位
51 |                    target.state, #数据有效标志位
52 |                    target.angle>>8,target.angle, #将整形数据拆分成两个8位
53 |                    target.distance>>8,target.distance, #将整形数据拆分成两个8位
54 |                    target.apriltag_id>>8,target.apriltag_id, #将整形数据拆分成两个8位
55 |                    target.img_width>>8,target.img_width, #将整形数据拆分成两个8位
56 |                    target.img_height>>8,target.img_height, #将整形数据拆分成两个8位
57 |                    target.fps, #数据有效标志位
58 |                    target.reserved1, #数据有效标志位
59 |                    target.reserved2, #数据有效标志位
60 |                    target.reserved3, #数据有效标志位
61 |                    target.reserved4, #数据有效标志位
62 |                    target.range_sensor1>>8,target.range_sensor1,
63 |                    target.range_sensor2>>8,target.range_sensor2,
64 |                    target.range_sensor3>>8,target.range_sensor3,
65 |                    target.range_sensor4>>8,target.range_sensor4,
66 |                    target.camera_id,
67 |                    target.reserved1_int32>>24&0xff,target.reserved1_int32>>16&0xff,
68 |                    target.reserved1_int32>>8&0xff,target.reserved1_int32&0xff,
69 |                    target.reserved2_int32>>24&0xff,target.reserved2_int32>>16&0xff,
70 |                    target.reserved2_int32>>8&0xff,target.reserved2_int32&0xff,
71 |                    target.reserved3_int32>>24&0xff,target.reserved3_int32>>16&0xff,
72 |                    target.reserved3_int32>>8&0xff,target.reserved3_int32&0xff,
73 |                    target.reserved4_int32>>24&0xff,target.reserved4_int32>>16&0xff,
74 |                    target.reserved4_int32>>8&0xff,target.reserved4_int32&0xff,
75 |                    0x00])
76 |    #数据包的长度
77 |    data_len=len(data)
```


2.2 飞控人机交互界面软件设计

依靠 Tiva LaunchPad 核心板上板载的两个独立按键 SW1、SW2，同时配合 OLED 显示屏实现对赛题中相关参数的进行修改、存储，新加功能需要兼容原来按键的翻页切换的功能，就必须设置相应的菜单界面配合按键的长按、短按响应来予以实现。

```

1  ssd1306_clear_display();
2  ssd1306_draw_line(0,32,128,32,WHITE);
3  ssd1306_draw_line(64,0,64,64,WHITE);
4  uint16_t x,y;
5  y=32-32*constrain_float(0.5*Opv_Top_View_Target.height-Opv_Top_View_Target.y,-50,50)/50.0f;
6  x=64-32*constrain_float(0.5*Opv_Top_View_Target.width-Opv_Top_View_Target.x,-50,50)/50.0f;
7  if(Opv_Top_View_Target.flag==1) ssd1306_fill_circle(x,y,2,WHITE);
8  else ssd1306_draw_circle(x,y,2,WHITE);
9
10 display_6_8_string(0,0,"px:"); write_6_8_number(80,0,Opv_Top_View_Target.x);
11 write_6_8_number(105,0,Page_Number+1);
12 display_6_8_string(0,1,"py:"); write_6_8_number(80,1,Opv_Top_View_Target.y);
13 write_6_8_number(105,1,Opv_Top_View_Target.sdk_mode-0xA0);
14 display_6_8_string(0,2,"sq:"); write_6_8_number(80,2,Opv_Top_View_Target.pixel);
15 display_6_8_string(0,3,"fg:"); write_6_8_number(80,3,Opv_Top_View_Target.flag);
16 display_6_8_string(0,4,"cmx:"); write_6_8_number(60,4,Opv_Top_View_Target.sdk_target.x);
17 write_6_8_number(110,4,Opv_Top_View_Target.fps);
18 display_6_8_string(0,5,"cmx:"); write_6_8_number(60,5,Opv_Top_View_Target.sdk_target.y);
19 display_6_8_string(0,6,"type:");
20
21 if(Opv_Top_View_Target.reserved3==1) display_6_8_string(30,6,"red");
22 else if(Opv_Top_View_Target.reserved3==2) display_6_8_string(30,6,"blue");
23 else display_6_8_string(30,6,"unk");
24
25 if(Opv_Top_View_Target.reserved4==1)
26 {
27     display_6_8_string(65,6,"circular");
28     ssd1306_draw_circle(64,32,16,WHITE);
29 }
30 else if(Opv_Top_View_Target.reserved4==2)
31 {
32     display_6_8_string(65,6,"rectangle");
33     ssd1306_draw_rect(48,16,32,32,WHITE);
34 }
35 else if(Opv_Top_View_Target.reserved4==3)
36 {
37     display_6_8_string(65,6,"triangle");
38     ssd1306_draw_triangle(48,48,80,48,64,21,WHITE);
39     ssd1306_draw_triangle(48,48,80,48,64,21,WHITE);
40 }
41 else display_6_8_string(65,6,"unk");
42 ssd1306_display();
43
44 display_6_8_string(0,7,"color");
45 display_6_8_number(80,7,param_value[7]);
46 display_6_8_number(110,7,param_value[8]);

```

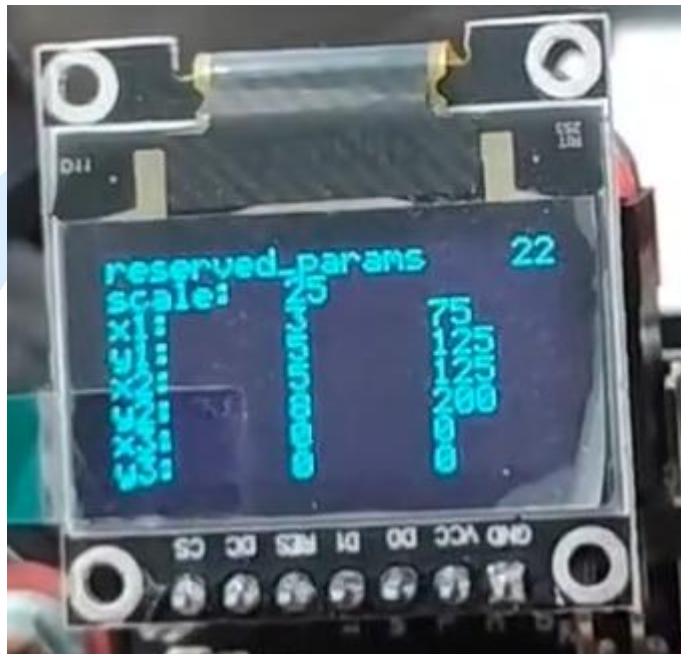
零基础学习竞赛无人机搭积木式编程指南

```
47 if(_button.state[UP].press==LONG_PRESS)
48 {
49     _button.state[UP].press=NO_PRESS;
50     hor_choose++;
51     if(hor_choose>3) hor_choose=1;
52     Blink_Set(&Light_2,500,50,0.2,0,GPIO_PORTF_BASE,GPIO_PIN_2,0); //蓝色
53 }
54
55 //通过下一页按键长按可以实现选中的参数行自增加调整
56 if(_button.state[DOWN].press==LONG_PRESS)
57 {
58     _button.state[DOWN].press=NO_PRESS;
59     Blink_Set(&Light_2,500,50,0.2,0,GPIO_PORTF_BASE,GPIO_PIN_2,0); //蓝色
60     switch(hor_choose)
61     {
62         case 1://将OPENMV识别到的模板特征作为待检测目标
63         {
64             param_value[7]=Opv_Top_View_Target.reserved3; //颜色
65             param_value[8]=Opv_Top_View_Target.reserved4; //形状
66         }
67         break;
68         case 2://可对模板颜色做进一步修改
69         {
70             param_value[7]+=1;
71             if(param_value[7]>2) param_value[7]=1;
72         }
73         break;
74         case 3://可对模板形状做进一步修改
75         {
76             param_value[8]+=1;
77             if(param_value[8]>3) param_value[8]=1;
78         }
79         break;
80     }
81     //按下后对参数进行保存
82     WriteFlashParameter(REERVED_PARAM+7,param_value[7],&Flight_Params); //颜色
83     WriteFlashParameter(REERVED_PARAM+8,param_value[8],&Flight_Params); //形状
84 }
85
92 case 24:
93 {
94     static uint8_t ver_item=1;
95     static uint16_t step=1;
96     LCD_clear_L(0,0);display_6_8_string(25,0,"loop_setup"); write_6_8_number(105,0,Page_Number+1);
97     LCD_clear_L(0,1);display_6_8_string(0,1,"step:"); write_6_8_number(80,1,step);
98     LCD_clear_L(0,2);display_6_8_string(0,2,"ax:"); write_6_8_number(80,2,param_value[9]);
99     LCD_clear_L(0,3);display_6_8_string(0,3,"ay:"); write_6_8_number(80,3,param_value[10]);
100    LCD_clear_L(0,4);display_6_8_string(0,4,"ang:"); write_6_8_number(80,4,param_value[11]);
101    LCD_clear_L(0,5);display_6_8_string(0,5,"save:"); display_6_8_string(40,5,"long press");
102
103    display_6_8_string(30,ver_item,"*");
104    //通过上一页按键长按,来实现换行选中待修改参数
105    if(_button.state[UP].press==LONG_PRESS)
106    {
107        _button.state[UP].press=NO_PRESS;
108        ver_item++;
109        if(ver_item>5) ver_item=1;
110        Blink_Set(&Light_2,500,50,0.2,0,GPIO_PORTF_BASE,GPIO_PIN_2,0); //蓝色
111    }
112
113    //通过下一页按键持续长按可以实现选中的参数行自增加调整
114    if(_button.state[DOWN].press==LONG_PRESS)
115    {
116        _button.state[DOWN].press=NO_PRESS;
117        switch(ver_item)
118        {
119            case 1:
120            {
121                step*=2;
122                if(step>100) step=1;
123            }
124            break;
125            case 2:
126            {
127                param_value[9]+=step;
128                if(param_value[9]>500) param_value[9]=0;
```

零基础学习竞赛无人机搭积木式编程指南

```
130     break;
131     case 3:
132     {
133         param_value[10]+=step;
134         if(param_value[10]>500) param_value[10]=0;
135     }
136     break;
137     case 4:
138     {
139         param_value[11]+=step;
140         if(param_value[11]>180) param_value[11]=0;
141     }
142     break;
143     case 5:
144     {
145         //按下后对参数进行保存
146         WriteFlashParameter(REERVED_PARAM+9 ,param_value[9],&Flight_Params);
147         WriteFlashParameter(REERVED_PARAM+10,param_value[10],&Flight_Params);
148         WriteFlashParameter(REERVED_PARAM+11,param_value[11],&Flight_Params);
149         Bling_Set(&Light_2,500,50,0.2,0,GPIO_PORTF_BASE,GPIO_PIN_2,0);//蓝色
150     }
151     break;
152 }
153 }
154 }
155 break;
```

菜单界面的包括换行参数选中、参数加减调整，由于按键功能复用的关系，操作起来相关比较繁琐，用户后期可以通过外接 ADC 按键的方式，用扩展按键实现本例中的按键操作功能，人机交互部分操作视频教程见下方链接。



TI 电赛 B 题送货无人机通过按键、菜单对坐标、模板进行设置

https://www.bilibili.com/video/BV1PB4y1t7y9/?vd_source=fa3e626a57e95e09ecf1b8f1627e58ac

无名创新

2.3 无人机自动飞行任务的软件设计

● 第一阶段——自动起飞到航巡高度

uint8_t Auto_Takeoff(float target)//自动起飞到某一高度

```

1643 uint8_t Auto_Takeoff(float target)
1644 {
1645     static uint8_t n=1;
1646     Vector3f target_position;
1647     basic_auto_flight_support();//基本飞行支持软件
1648     if(flight_subtask_cnt[n]==0)
1649     {
1650         //不加此行代码，当后续全程无油门上下动作后，飞机最后自动降落到地面不会自动上锁
1651         Unwanted_Lock_Flag=0;//允许飞机自动上锁，原理和手动推油起飞类似
1652
1653         //记录下初始起点位置，实际项目中可设置为某一基准原点
1654         base_position.x=VIO_SINS.Position[_EAST];
1655         base_position.y=VIO_SINS.Position[_NORTH];
1656         base_position.z=NamelessQuad.Position[_UP];
1657
1658         //execute time ms[n]=10000/flight_subtask_delta;//子任务执行时间
1659         target_position.x=base_position.x;
1660         target_position.y=base_position.y;
1661         target_position.z=base_position.z+target;
1662         Horizontal_Navigation(target_position.x,
1663                             target_position.y,
1664                             target_position.z,
1665                             GLOBAL_MODE,
1666                             MAP_FRAME);
1667         flight_subtask_cnt[n]=1;
1668         return 0;
1669     }
1670     else if(flight_subtask_cnt[n]==1)
1671     {
1672         //判断是否起飞到目标高度
1673         if(flight_global_cnt[n]<400)//持续400*5ms满足
1674         {
1675             if(ABS(Total_Controller.High_Position_Control.Err)<=10.0f) flight_global_cnt[n]++;
1676             else flight_global_cnt[n]/=2;
1677             return 0;
1678         }
1679         else//持续200*5ms满足，表示到达目标高度
1680         {
1681             return 1;
1682         }
1683     }
1684     return 0;
1685 }

```

函数输入参数 **target** 为目标高度，自动起飞任务分为两个线程，第一步为记录当前 3 维位置信息，作为导航初始原点位置。并且通过导航控制函数设置期望目标高度位置。第二步为实时检测高度偏差值，连续 2S 满足位置偏差在 10cm 以内后，函数返回值置 1 后，自动起飞到目标高度任务完成，用法参照 Developer_Mode.c 开发者模式中 case 11 用法，自主起飞任务完成后会进入 case 15/16/17 执行航点遍历作业任务。

```

129 case 11://自动起飞到某一高度
130 {
131     if (Auto_Takeoff(Target_Height)==1)//WORK_HEIGHT_CM
132     {
133         // *mode+=1;//到达目标高度后，切换到下一SDK任务
134         *mode+=4;//到达目标高度后，切换2022年7月省赛第一部分任务
135         // *mode+=5;//到达目标高度后，切换2022年7月省赛第二部分任务
136         // *mode+=6;//到达目标高度后，切换2022年7月省赛第三部分任务
137     }
138 }
139 break;

```

```
162 | case 15:
163 | {
164 |     if(openmv_work_mode==0)//只配置一次
165 |     {
166 |         openmv_work_mode=0x10;
167 |         SDK_DT_Send_Check(openmv_work_mode,UART3_SDK); //起飞完毕之后，将底部OPENMV设置成色块、形状检测模式
168 |     }
169 |     //2022年月电子设计竞赛B题送货无人机—第1部分
170 |     Deliver_UAV_Basic();
171 | }
172 | break;
173 | case 16:
174 | {
175 |     if(openmv_work_mode==0)//只配置一次
176 |     {
177 |         openmv_work_mode=0x10;
178 |         SDK_DT_Send_Check(openmv_work_mode,UART3_SDK); //起飞完毕之后，将底部OPENMV设置成色块、形状检测模式
179 |     }
180 |     //2022年月电子设计竞赛B题送货无人机—第2部分
181 |     Deliver_UAV_Innovation();
182 | }
183 | break;
184 | case 17:
185 | {
186 |     //2022年月电子设计竞赛B题送货无人机—第3部分
187 |     Deliver_UAV_Hulahoop();
188 | }
189 | break;
```

● 第二阶段——航点遍历作业任务

void Deliver_UAV_Basic(void)

第一步将高度期望设置成第一作业高度 150cm，水平位置期望为初始起飞时候的水平位置，并设置激光笔为持续闪烁，便于裁判判断飞机机身中心在地面上的投影位置，起飞点上方悬停时间设置为 5S。

```
1783 | void Deliver_UAV_Basic(void)
1784 | {
1785 |     static uint8_t n=12;
1786 |     Vector3f target_position;
1787 |     float x=0,y=0,z=0;
1788 |     if(flight_subtask_cnt[n]==0)//起飞点作为第一个悬停点
1789 |     {
1790 |         basic_auto_flight_support();//基本飞行支持软件
1791 |         deliver_work_waypoint_generate();//备用
1792 |
1793 |         //记录下初始起点位置，实际项目中可设置为某一基准原点
1794 |         //base_position.x=VIO_SINS.Position[EAST];
1795 |         //base_position.y=VIO_SINS.Position[NORTH];
1796 |         base_position.z=First_Working_Height;//第一作业高度
1797 |
1798 |         x=base_position.x;
1799 |         y=base_position.y;
1800 |         z=First_Working_Height;
1801 |         target_position.x=x;
1802 |         target_position.y=y;
1803 |         target_position.z=z;
1804 |         Horizontal_Navigation(target_position.x,
1805 |                               target_position.y,
1806 |                               target_position.z,
1807 |                               GLOBAL_MODE,
1808 |                               MAP_FRAME);
1809 |
1810 |         flight_subtask_cnt[n]=1;
1811 |         flight_global_cnt[n]=0;
1812 |         execute_time_ms[n]=5000/flight_subtask_delta;//子任务执行时间
1813 |
1814 |         laser_light_1.reset=1;
1815 |         laser_light_1.times=50000;//闪烁50000次
1816 |         laser_light_1.period=200;
1817 |         laser_light_1.light_on_percent=0.98;
```

起飞点上方悬停 5S 后，会将键盘输入的第一个作业点的水平坐标填入到期望的目标位置，随后飞机会执行从起飞点正上方飞向第一作业点的动作，在执行本任务中对水平位置误差进行实时检测，连续 N 次水平位置误差小于某一阈值，即可以认为到达第一作业点正上方附近，满足达到第一作业点条件后会将高度期望设置成第二作业高度，飞机高度会下降。

无名创新


```

1818 else if(flight_subtask_cnt[n]==1)//起飞之后原定悬停5后再执行航点任务
1819 {
1820     basic_auto_flight_support();//基本飞行支持软件
1821     if(execute_time_ms[n]>0) execute_time_ms[n]--;
1822     if(execute_time_ms[n]==0)//悬停时间计数器归零，悬停任务执行完毕
1823     {
1824         x=param_value[0]*param_value[1];
1825         y=param_value[0]*param_value[2];
1826         z=First_Working_Height;
1827         target_position.x=base_position.x+x;
1828         target_position.y=base_position.y+y;
1829         target_position.z=z;
1830         Horizontal_Navigation(target_position.x,
1831                               target_position.y,
1832                               target_position.z,
1833                               GLOBAL_MODE,
1834                               MAP_FRAME);
1835
1836         flight_subtask_cnt[n]++;
1837         flight_global_cnt[n]=0;
1838         execute_time_ms[n]=0;
1839     }
1840 }
1841 else if(flight_subtask_cnt[n]==2)//检测起飞点悬停完毕后，飞向第一个目标点
1842 {
1843     basic_auto_flight_support();//基本飞行支持软件
1844
1845     //判断是否到达目标航点位置
1846     if(flight_global_cnt[n]<Times_Fixed)//持续10*5ms=0.05s满足
1847     {
1848         float dis_cm=pythagorous2(OpticalFlow_Pos_Ctrl_Err.x,OpticalFlow_Pos_Ctrl_Err.y);
1849         if(dis_cm<=Fixed_CM) flight_global_cnt[n]++;
1850         else flight_global_cnt[n]/=2;
1851     }
1852     else//持续10*5ms满足，表示到达目标航点位置，后降低目标高度
1853     {
1854         x=param_value[0]*param_value[1];
1855         y=param_value[0]*param_value[2];
1856         z=Second_Working_Height;
1857         target_position.x=base_position.x+x;
1858         target_position.y=base_position.y+y;
1859         target_position.z=z;
1860         Horizontal_Navigation(target_position.x,
1861                               target_position.y,
1862                               target_position.z,
1863                               GLOBAL_MODE,
1864                               MAP_FRAME);
1865         flight_subtask_cnt[n]++;
1866         flight_global_cnt[n]=0;
1867     }
1868 }

```

第三步是判断无人机是否到达第二作业高度，检测高度是否达到的原理和水平方向一样，同样是检测高度方向上位置误差持续 N 次满足某一阈值。

```

1869 else if(flight_subtask_cnt[n]==3)
1870 {
1871     basic_auto_flight_support();//基本飞行支持软件
1872     //判断是否到达第二作业高度
1873     if(flight_global_cnt[n]<400)//持续400*5ms满足
1874     {
1875         if(ABS(Total_Controller.Height_Position_Control.Err)<=10.0f) flight_global_cnt[n]++;
1876         else flight_global_cnt[n]/=2;
1877     }
1878     else
1879     {
1880         flight_subtask_cnt[n]++;
1881         flight_global_cnt[n]=0;
1882         execute_time_ms[n]=deliver_down_time/flight_subtask_delta;
1883     }
1884 }

```

第四步飞机处于第一作业点正上方附件，并且高度已经调整为第二作业高度，此时由于定位传感器会存在一定位置误差，这个时候水平方向位置控制会引入 OPENMV 识别到的色块坐标信息，无人机会对水平位置进行二次修正，使得无人机中心投影处于色块的中心区域。相当于用 SLAM 定位提供的位置信息做

粗对准，在接近作业点时使用机器视觉实现进一步地精确对准。

```

1885 else if(flight_subtask_cnt[n]==4) //执行塔吊下方任务
1886 {
1887     //basic_auto_flight_support();//基本飞行支持软件
1888     Color_Block_Control_Pilot();//俯视OPENMV视觉水平追踪
1889     Flight.yaw_ctrl_mode=ROTATE;
1890     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
1891     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
1892
1893     if(execute_time_ms[n]>0)
1894     {
1895         execute_time_ms[n]--;
1896         Reserved_PWM1_Output(deliver_down_pwm_us);//放下塔吊
1897     }
1898
1899     if(execute_time_ms[n]==0) //悬停时间计数器归零，悬停任务执行完毕
1900     {
1901         Reserved_PWM1_Output(deliver_stop_pwm_us);//停止转动
1902         //不需要刷新位置期望，和上个线程保持一致
1903         flight_subtask_cnt[n]++;
1904         flight_global_cnt[n]=0;
1905         execute_time_ms[n]=5000/flight_subtask_delta;
1906
1907         //蜂鸣器语音提示
1908         laser_light_2.period=200;//200*5ms
1909         laser_light_2.light_on_percent=0.5f;
1910         laser_light_2.reset=1;
1911         laser_light_2.times=4;//闪烁4次
1912     }
1913 }

```

与此同时会执行吊舱下放动作指令，利用的是 360 度转动的舵机，以某一角速度执行一定时间予以实现，具体的时间角速度、时间参数和自己舵机选型、绞盘的半径有关，这两个参数在实际调试也比较容易确定。塔吊下方执行完毕后会驱动高分贝蜂鸣器进行报警提示，原地悬停 5S 之后抬升塔吊装置并恢复到第一作业高度

```

1914 else if(flight_subtask_cnt[n]==5) //原地悬停5s，后恢复到第一巡航高度
1915 {
1916     //basic_auto_flight_support();//基本飞行支持软件
1917     Color_Block_Control_Pilot();//俯视OPENMV视觉水平追踪
1918     Flight.yaw_ctrl_mode=ROTATE;
1919     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
1920     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
1921
1922     if(execute_time_ms[n]>0) execute_time_ms[n]--;
1923     if(execute_time_ms[n]==0) //悬停时间计数器归零，悬停任务执行完毕
1924     {
1925         OpticalFlow_Control_Pure(1);//强制刷新悬停位置
1926         flight_subtask_cnt[n]++;
1927         flight_global_cnt[n]=0;
1928         execute_time_ms[n]=deliver_up_time/flight_subtask_delta;;
1929     }
1930 }

```

无名创新

```
1931 else if(flight_subtask_cnt[n]==6)//收起塔吊
1932 {
1933     basic_auto_flight_support();//基本飞行支持软件
1934     if(execute_time_ms[n]>0)
1935     {
1936         execute_time_ms[n]--;
1937         Reserved_PWM1_Output(deliver_up_pwm_us);//收起塔吊
1938     }
1939
1940     if(execute_time_ms[n]==0)//悬停时间计数器归零，悬停任务执行完毕
1941     {
1942         Reserved_PWM1_Output(deliver_stop_pwm_us);//停止转动
1943
1944         x=param_value[0]*param_value[1];
1945         y=param_value[0]*param_value[2];
1946         z=First_Working_Height;
1947         target_position.x=base_position.x+x;
1948         target_position.y=base_position.y+y;
1949         target_position.z=z;
1950         Horizontal_Navigation(target_position.x,
1951                               target_position.y,
1952                               target_position.z,
1953                               GLOBAL_MODE,
1954                               MAP_FRAME);
1955
1956         flight_subtask_cnt[n]++;
1957         flight_global_cnt[n]=0;
1958         execute_time_ms[n]=5000/flight_subtask_delta;;
1959     }
1960 }
```

第五步是无人机执行完第一作业点任务并回到第一作业高度后，会想任务线程发布第二作业点的位置，与第一作业点类似无人机会到第二作业点正上方附近，随即下降高度到第二作业高度，执行视觉二次对准、下放/抬升塔吊、蜂鸣器报警，之后重新返回到第一作业高度，准备返航。

```
1961 else if(flight_subtask_cnt[n]==7)//第一个航点执行完毕后，恢复到第一飞行高度
1962 {
1963     basic_auto_flight_support();//基本飞行支持软件
1964     //判断是否到达第一作业高度
1965     if(flight_global_cnt[n]<200)//持续200*5ms满足
1966     {
1967         if(ABS(Total_Controller.Height_Position_Control.Err)<=10.0f) flight_global_cnt[n]++;
1968         else flight_global_cnt[n]/=2;
1969     }
1970     else
1971     {
1972         //达到第一飞行高度后，继续执行第二个航点任务
1973         x=param_value[0]*param_value[3];
1974         y=param_value[0]*param_value[4];
1975         z=First_Working_Height;
1976         target_position.x=base_position.x+x;
1977         target_position.y=base_position.y+y;
1978         target_position.z=z;
1979         Horizontal_Navigation(target_position.x,
1980                               target_position.y,
1981                               target_position.z,
1982                               GLOBAL_MODE,
1983                               MAP_FRAME);
1984
1985         flight_subtask_cnt[n]++;
1986         flight_global_cnt[n]=0;
1987         execute_time_ms[n]=0;
1988     }
1989 }
```

无名创新

零基础学习竞赛无人机搭积木式编程指南

```
1990 else if(flight_subtask_cnt[n]==8)//检测起飞点悬停完毕后，飞向第二个目标点
1991 {
1992     basic_auto_flight_support();//基本飞行支持软件
1993
1994     //判断是否到达目标航点位置
1995     if(flight_global_cnt[n]<Times_Fixed)//持续10*5ms=0.05s满足
1996     {
1997         float dis_cm=pythagorou2(OpticalFlow_Pos_Ctrl_Err.x,OpticalFlow_Pos_Ctrl_Err.y);
1998         if(dis_cm<=Fixed_CM) flight_global_cnt[n]++;
1999         else flight_global_cnt[n]/=2;
2000     }
2001     else//持续10*5ms满足，表示到达目标航点位置，后降低目标高度
2002     {
2003         x=param_value[0]*param_value[3];
2004         y=param_value[0]*param_value[4];
2005         z=Second_Working_Height;
2006         target_position.x=base_position.x+x;
2007         target_position.y=base_position.y+y;
2008         target_position.z=z;
2009         Horizontal_Navigation(target_position.x,
2010                             target_position.y,
2011                             target_position.z,
2012                             GLOBAL_MODE,
2013                             MAP_FRAME);
2014         flight_subtask_cnt[n]++;
2015         flight_global_cnt[n]=0;
2016     }
2017 }
2018 else if(flight_subtask_cnt[n]==9)
2019 {
2020     basic_auto_flight_support();//基本飞行支持软件
2021
2022     //判断是否到达第二作业高度
2023     if(flight_global_cnt[n]<200)//持续200*5ms满足
2024     {
2025         if(ABS(Total_Controller.Height_Position_Control.Err)<=10.0f) flight_global_cnt[n]++;
2026         else flight_global_cnt[n]/=2;
2027     }
2028     else//放出吊仓
2029     {
2030         flight_subtask_cnt[n]++;
2031         flight_global_cnt[n]=0;
2032         execute_time_ms[n]=deliver_down_time/flight_subtask_delta;
2033     }
2034 }
2035 else if(flight_subtask_cnt[n]==10)//执行塔吊下方任务
2036 {
2037     //basic_auto_flight_support();//基本飞行支持软件
2038     Color_Block_Control_Pilot();//俯视OPENMV视觉水平追踪
2039     Flight.yaw_ctrl_mode=ROTATE;
2040     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
2041     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
2042
2043     if(execute_time_ms[n]>0)
2044     {
2045         execute_time_ms[n]--;
2046         Reserved_PWM1_Output(deliver_down_pwm_us);//放下塔吊
2047     }
2048
2049     if(execute_time_ms[n]==0)//悬停时间计数器归零，悬停任务执行完毕
2050     {
2051         Reserved_PWM1_Output(deliver_stop_pwm_us);//停止转动
2052
2053         flight_subtask_cnt[n]++;
2054         flight_global_cnt[n]=0;
2055         execute_time_ms[n]=5000/flight_subtask_delta;
2056
2057         //蜂鸣器语音提示
2058         laser_light_2.period=200;//200*5ms
2059         laser_light_2.light_on_percent=0.5f;
2060         laser_light_2.reset=1;
2061         laser_light_2.times=4;//闪烁4次
2062     }
2063 }
```

零基础学习竞赛无人机搭积木式编程指南

```
2064 else if(flight_subtask_cnt[n]==11)//原地悬停5s, 后恢复到第一巡航高度
2065 {
2066     //basic_auto_flight_support();//基本飞行支持软件
2067     Color_Block_Control_Pilot();//俯视OPENMV视觉水平追踪
2068     Flight.yaw_ctrl_mode=ROTATE;
2069     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
2070     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
2071
2072     if(execute_time_ms[n]>0) execute_time_ms[n]--;
2073     if(execute_time_ms[n]==0)//悬停时间计数器归零, 悬停任务执行完毕
2074     {
2075         OpticalFlow_Control_Pure(1);//强制刷新悬停位置
2076         flight_subtask_cnt[n]++;
2077         flight_global_cnt[n]=0;
2078         execute_time_ms[n]=deliver_up_time/flight_subtask_delta;
2079     }
2080 }

2081 else if(flight_subtask_cnt[n]==12)//收起塔吊
2082 {
2083     basic_auto_flight_support();//基本飞行支持软件
2084     if(execute_time_ms[n]>0)
2085     {
2086         execute_time_ms[n]--;
2087         Reserved_PWM1_Output(deliver_up_pwm_us);//收起塔吊
2088     }
2089
2090     if(execute_time_ms[n]==0)//悬停时间计数器归零, 悬停任务执行完毕
2091     {
2092         Reserved_PWM1_Output(deliver_stop_pwm_us);//停止转动
2093
2094         x=param_value[0]*param_value[3];
2095         y=param_value[0]*param_value[4];
2096         z=First_Working_Height;
2097         target_position.x=base_position.x+x;
2098         target_position.y=base_position.y+y;
2099         target_position.z=z;
2100         Horizontal_Navigation(target_position.x,
2101                               target_position.y,
2102                               target_position.z,
2103                               GLOBAL_MODE,
2104                               MAP_FRAME);
2105
2106         flight_subtask_cnt[n]++;
2107         flight_global_cnt[n]=0;
2108         execute_time_ms[n]=0;
2109     }
2110 }

2111 else if(flight_subtask_cnt[n]==13)//第二个航点执行完毕后, 恢复到第一飞行高度
2112 {
2113     basic_auto_flight_support();//基本飞行支持软件
2114     //判断是否到达第二作业高度
2115     if(flight_global_cnt[n]<200)//持续400*5ms满足
2116     {
2117         if(ABS(Total_Controller.Height_Position_Control.Err)<=10.0f) flight_global_cnt[n]++;
2118         else flight_global_cnt[n]/=2;
2119     }
2120     else
2121     {
2122         //达到第一飞行高度后, 继续执行返航任务航点任务
2123         target_position.x=base_position.x;
2124         target_position.y=base_position.y;
2125         target_position.z=First_Working_Height;
2126         Horizontal_Navigation(target_position.x,
2127                               target_position.y,
2128                               target_position.z,
2129                               GLOBAL_MODE,
2130                               MAP_FRAME);
2131
2132         flight_subtask_cnt[n]++;
2133         flight_global_cnt[n]=0;
2134         execute_time_ms[n]=0;
2135     }
2136 }
```

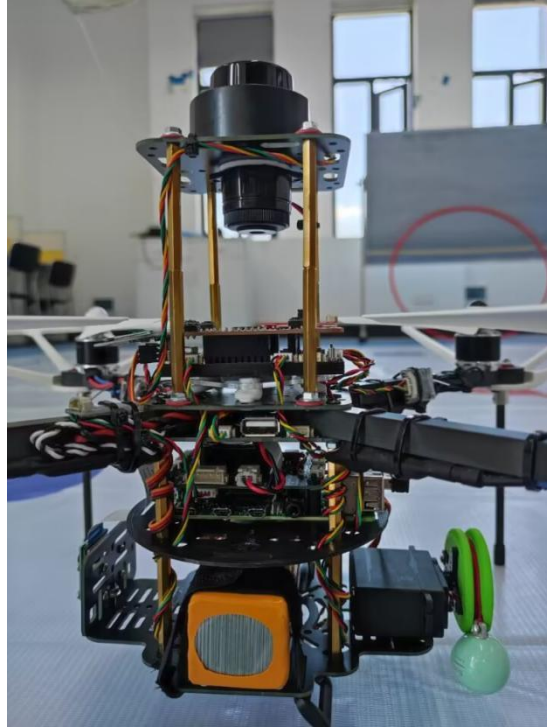

完成预设的两个作业送货后，无人机会执行返航动作，首先无人机飞到起飞点正上方，连续 N 次水平位置误差小于某一阈值即可认为达到起飞点正上方附近，满足水平抵达之后会执行原地降落至地面的任务，到达地面后无人机会满足地面检测条件自动上锁。

```

2137 | else if(flight_subtask_cnt[n]==14) //飞向起飞点正上方
2138 | {
2139 |     basic_auto_flight_support(); //基本飞行支持软件
2140 |
2141 |     //判断是否到达目标航点位置
2142 |     if(flight_global_cnt[n]<Times_Fixed) //持续10*5ms=0.05s满足
2143 |     {
2144 |         float dis_cm=pythagorous2(OpticalFlow_Pos_Ctrl_Err.x,OpticalFlow_Pos_Ctrl_Err.y),
2145 |         if(dis_cm<=Fixed_CM) flight_global_cnt[n]++;
2146 |         else flight_global_cnt[n]/=2;
2147 |     }
2148 |     else //持续10*5ms满足，表示到达目标航点位置
2149 |     {
2150 |         flight_subtask_cnt[n]++;
2151 |         flight_global_cnt[n]=0;
2152 |         execute_time_ms[n]=0;
2153 |         /*
2154 |         以下复位操作的作用：空中到地面不同高度，环境分布陈设变化造成的误差
2155 |         不同高度梯度上陈设变化不大时，以下特殊处理部分可以去掉
2156 |         */
2157 |         //特殊处理开始
2158 |         send_check_back=4; //重置slam
2159 |         VIO_SINS.Position[_EAST] = 0;
2160 |         VIO_SINS.Position[_NORTH]= 0;
2161 |         OpticalFlow_Pos_Ctrl_Expect.x=0;
2162 |         OpticalFlow_Pos_Ctrl_Expect.y=0;
2163 |         //特殊处理结束
2164 |     }
2165 | }
2166 | else if(flight_subtask_cnt[n]==15) //原地下降
2167 | {
2168 |     Flight.yaw_ctrl_mode=ROTATE;
2169 |     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
2170 |     OpticalFlow_Control_Pure(0);
2171 |     Flight_Alt_Hold_Control(ALTHOLD_AUTO_VEL_CTRL,NUL,-30); //高度控制
2172 | }
2173 | else
2174 | {
2175 |     basic_auto_flight_support(); //基本飞行支持软件
2176 | }
2177 | }
    
```

这里需要注意的是程序在此处有一段特殊处理部分为 SLAM 建图复位指令，在之所以需要这么操作的原因是减小定位误差，NC360 竞赛无人机平台使用的是 2D 激光雷达，理论上只适合平面运动机器人平台的 SLAM 建图，无人机在空中的线性运动为 3 维的，无人机在不同高度上，由于周围环境的陈设布置变化会造成一定的定位误差，通常在 3~5cm 以内，该误差范围能满足无人机在竞赛中的控制精度要求。对于不同高度空间环境变化较大的情况，可以采用本方案中的特殊处理形式，本特殊处理方式非必须，用户可以自己二次开发过程中根据实际情况来决策是否选用，同时对于末端降落过程，依然可以利用底部视觉特征精准对准的方法，实现精准降落。

针对创新+发挥部分的自动学习和穿越圆环部分内容，大体过程和本例程类似，差别在于坐标是通过现场识别到的模板特征录入的，这个处理过程主要工作量在于 OPENMV 部分，整个飞行任务执行过程和基础部分基本一样，同时对于穿越圆环部分，通过手动输入圆环的坐标和大致角度后，整个飞行过程就是航点遍历、机头对准圆环、穿越的过程，执行完毕后返航降落，整个过程相比前两项只是多了一个航向控制，相对来讲实现难度并没有拔高，仍然属于常规考察项，用户结合程序注释可以比较容易理解，故在这里不做展开。



2022 年 7 月电赛 B 题送货无人机加装硬件介绍

https://www.bilibili.com/video/BV1re4y1D7zJ/?vd_source=fa3e626a57e95e09ecf1b8f1627e58ac

2022 年 TI 电赛 B 题一送货无人机开源方案

https://www.bilibili.com/video/BV1PB4y1t7eM/?vd_source=fa3e626a57e95e09ecf1b8f1627e58ac

任意位置、角度自主飞行穿越圆框——2022 年 TI 电赛飞行器 B 题送货无人机

https://www.bilibili.com/video/BV14S4y1474j/?vd_source=fa3e626a57e95e09ecf1b8f1627e58ac



2021 年电赛 G 题植保无人机国奖标准方案学习样例

https://www.bilibili.com/video/BV11T4y1v7uW?spm_id_from=333.999.0.0

https://www.bilibili.com/video/BV1QR4y1F7vj?spm_id_from=333.999.0.0

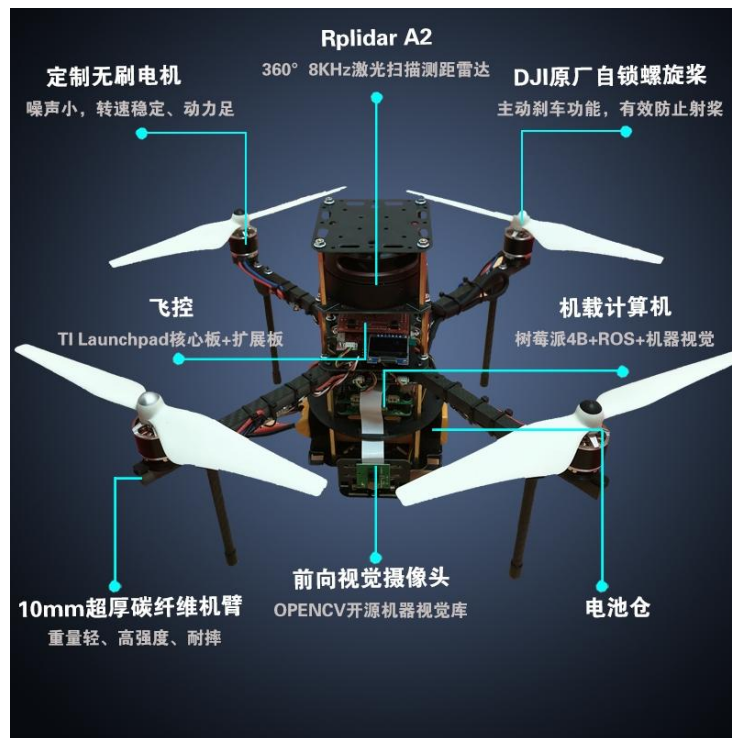


5_竞赛无人机搭积木式编程 ——以 2021 年电赛国奖标准完整复现为例学习

<https://www.bilibili.com/read/cv15844252>

无名创新

NC360 深度开源竞赛无人机开发平台



功能概览



1 送货无人机（7 月份 B 题）

1.1 任务

设计一基于多旋翼飞行器的送货无人机，能够根据不同的要求，向指定的目标地点运送货物。图 2 为作业区域示意图，有起飞降落点和多个具有不同特征的目标地点。

送货无人机上需安装一可升降吊舱，吊舱重量 $50 \pm 5\text{g}$ ，升降范围 $60 \pm 10\text{cm}$ ；起飞、飞行过程中，吊舱紧贴无人机腹；到达目标地点上方，无人机下降悬停，并将吊舱降至距离地面一定高度，送货操作完成后恢复到巡航高度飞行。无人机上需安装扬声器，可播放语音提示信息；无人机安装垂直向下的激光笔，用以标识航迹。

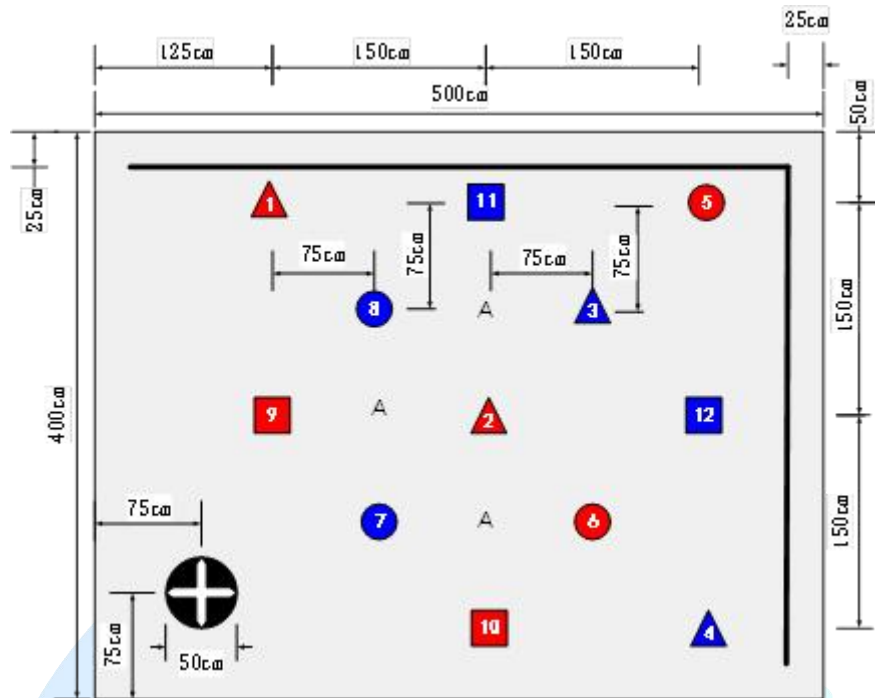


图2 送货无人机作业区示意图

1.2 要求

1. 无人机可按照现场设置的目标位置信息，对 2 个指定位置的目标地点完成送货作业。目标地点位置信息坐标可用无人机上携带的键盘设置。

- （1） 无人机在“十”字起降点垂直起飞，升空至 $150 \pm 10\text{cm}$ 的巡航高度；（5 分）
- （2） 根据现场设置的送货目标，先后依次飞行抵达目标地点上方，无人机降低飞行高度到 $80 \pm 10\text{cm}$ ；（16 分）
- （3） 无人机放出吊舱，吊舱降至距地面 $20 \pm 5\text{cm}$ 高度，并保持稳定悬停 5 秒，完成送货作业，期间播放提醒目标点收货的语音；悬停期间，标识无人机位置的激光笔光斑落在以目标中心为圆心、半径 15cm 的圆内；（20 分）
- （4） 作业完成后飞行到起降点稳定准确降落，无人机几何中心点与起降点中心距离偏差不大于 $\pm 10\text{cm}$ ；（4 分）
- （5） 送货过程必须要在 180 秒内完成，用时越少越好。（5 分）

2. 无人机在作业区外学习识别某一种指定目标特征（颜色、形状），然后寻找具有此特征的两个目标地点，完成送货作业。

- (1) 无人机从起降点起飞到 150cm 巡航高度，先后寻找 2 个上述已识别的目标，飞行抵达目标地点上方，降低飞行高度到 $80 \pm 10\text{cm}$ 左右；（20 分）
- (2) 放出下降吊舱至距地面 $20 \pm 5\text{cm}$ 左右高度，稳定保持悬停 5 秒完成送货作业，期间播放提醒目标点收货的语音；送货期间，标识无人机位置的激光笔光斑落在以目标中心为圆心、半径 15cm 的圆内；送货完成即恢复巡航高度；（10 分）
- (3) 送货作业完成后无人机降落到起降点；送货过程用时越少越好，需在 270 秒内完成；（5 分）

3. 无人机找到放置在 A 附近区域的红色圆框，并从圆框中穿越而过。（10 分）

4. 其他自主发挥。（5 分）

5. 设计报告

项 目	主要内容	满分
系统方案	技术路线、系统结构，方案描述、比较与选择	3
设计与计算	控制方法描述及参数计算	5
电路与程序设计	系统组成，原理框图与各部分电路图 系统软件设计与流程图	7
测试方案与测试结果	测试方案及测试条件；测试结果完整性；测试结果分析	3
设计报告结构及规范性	摘要、报告正文结构、公式、图表的完整性和规范性	2
小计		20

1.3 说明

1. 送货作业现场说明

- (1) 参赛队在赛区提供的场地测试，不得擅自改变测试环境条件。
- (2) 送货作业区域铺设亚光喷绘布为淡灰色（R-240、G-240、B-240），目标地点的形状有三种（圆、正方形、三角形，最大边长或直径为 25cm），颜色有红、蓝两种（红 R-255、G-0、B-0，蓝 R-0、G-0、B-255）；作业区上、右两侧有1.8cm宽黑色标志线；应考虑到材料及颜料导致颜色存在差异的可能性。
- (3) 测评将现场准备图 1 所示 6 种送货目标的特征样板，如“红色三角形”、“蓝色正方形”等，以备给无人机识别。
- (4) 送货目标中的数字并非给无人机识别用，仅为了描述、记录方便，颜色与目标底色相近。
- (5) 作业区域中标志“A”所在附近区域可放置供无人机穿越的圆框，见图 3，圆框可采用外径约 110cm 的红色呼啦圈，呼啦圈固定在地面支架上，圆心高度约 150cm，支架为黑色。

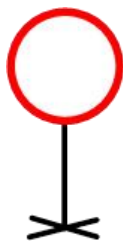


图3 圆框示意图

- (6) 400cm×500cm 作业区四周及顶部设置安全网，安全网支架在安全网外。
- (7) 测试现场避免阳光直射，但不排除顶部照明灯及窗外环境光照射，参赛队应考虑到测试现场会受到外界光照或室内照明不均等影响因素；测试时不得提出光照条件要求。

2. 飞行器要求

- (1) 参赛队使用无人机时应遵守中国民用航空局的相关管理规定。
- (2) 无人机最大轴间距不大于 45cm。
- (3) 无人机桨叶必须全防护，否则不得测试。
- (4) 无人机上的激光笔垂直向下安装，不得移动、转动。
- (5) 起飞前，无人机可手动放置到起降点；可手动一键启动后起飞，起飞后整个飞行过程中不得人为干预；若采用无人机以外的启动或急停操作装置，一键启动起飞操作后必须立刻将装置交给工作人员。
- (6) 调试及测试时必须佩戴防护眼镜，穿戴防护手套。

3. 测试要求与说明

- (1) 吊舱可用软线悬吊 50g 砝码来模仿。
- (2) 提醒目标点收货的语音可自行设定，时长 1~3 秒。
- (3) 要求 1 送货前，可连续输入两个目标地点的位置信息；如，若以起降点为原点，编号 11 号目标的位置可为 (200, 275)；目标的位置信息格式可自己定义。
- (4) 在要求 2 送货前，将现场指定形状及颜色的样板（如“红色三角形”），在场外手持给无人机学习识别将要送货的目标特征。
- (5) 要求 1 的送货过程必须在 180 秒内完成，超时不得分。
- (6) 要求 2 的送货过程必须在 270 秒内完成，超时不得分。
- (7) 要求 1 的 (1) ~ (4) 必须连续完成，期间不得人为干预；要求 2 的 (1) ~ (3) 必须连续完成，期间不得人为干预。
- (8) 每次测试全过程中不得更换电池；两次测试之间允许更换电池，更换电池时间不大于 2 分钟。
- (9) 飞行期间，无人机触及地面后自行恢复飞行的，扣 5 分；触地后 5 秒内不能自行恢复飞行视为失败，失败前完成动作仍计分。

平稳降落是指在降落过程中无明显的跌落、弹跳及着地后滑行等情况出现。