

② 总控制器 Controler_Mode_Select()函数

- 5 通道为高度手动（低位）与定高模式（高位）控制通道——定高控制
- 8 通道为水平自稳（低位）到光流定点（高位）模式控制通道——定点控制
- 6 通道为非 SDK 模式（低位）到 SDK 模式（高位）控制通道——SDK 控制
- 7 通道为非一键降落（低位）到一键降落（高位）控制通道——降落控制

其中需要特别注意的是：

武汉无名创新科技有限公司

- 8 通道中光流定点模式高位起作用的前提是飞行器之前需要工作在定高模式模式，即 5 通道必须先置为高位。
- 6 通道中 SDK 模式高位起作用的前提是飞行器之前需要工作在光流定点模式，即 5、8 通道必须均先置为高位。

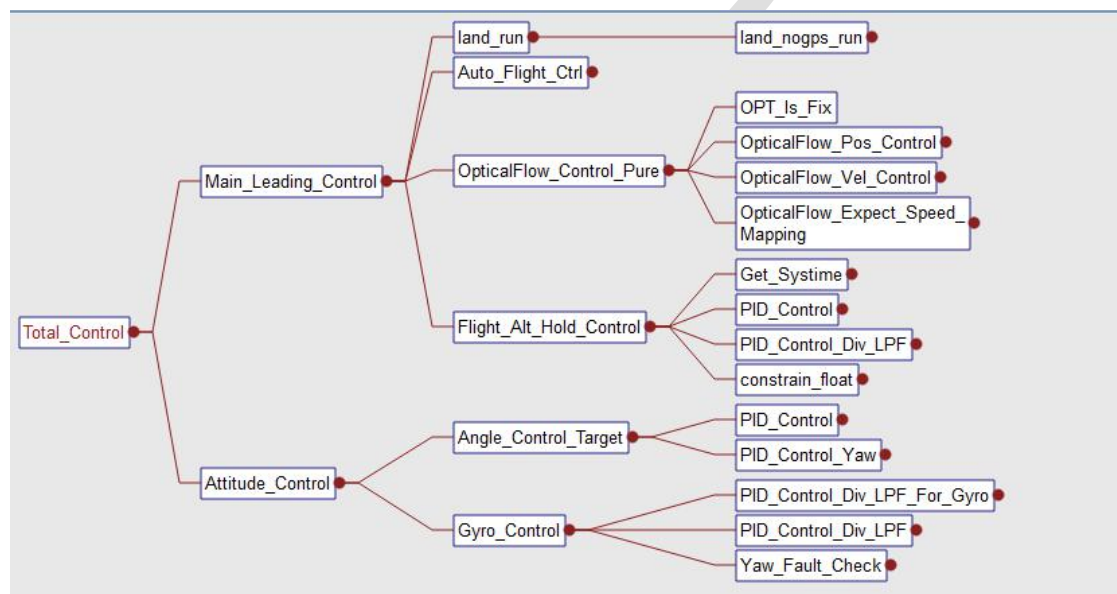
```
71 void Controller_Mode_Select()
72 {
73     Last_Controller_High_Mode=Controller_High_Mode; //上次高度控制模式
74     Last_Controller_Horizontal_Mode=Controller_Horizontal_Mode; //上次位置控制模式
75     Last_Controller_Land_Mode=Controller_Land_Mode; //上次返航模式
76
77     if (PPM_Databuf[4] >= (RC_Data.cal[4].max-RC_Data.cal[4].deadband)) Controller_High_Mode=2; //气压计、超神波定高
78     else if (PPM_Databuf[4] <= (RC_Data.cal[4].min+RC_Data.cal[4].deadband)) Controller_High_Mode=1; //纯姿态自稳
79
80     if (PPM_Databuf[5] >= (RC_Data.cal[5].max-RC_Data.cal[5].deadband))
81     {
82         Controller_SDKI_Mode=1; //SDKI模式
83     }
84     else
85     {
86         Controller_SDKI_Mode=0; //非SDKI模式
87         NCQ_SDKI_Reset();
88     }
89
90
91     if (PPM_Databuf[6] >= (RC_Data.cal[6].max-RC_Data.cal[6].deadband)) (Controller_Land_Mode=2); //返航模式
92     else if (PPM_Databuf[6] <= (RC_Data.cal[6].middle+RC_Data.cal[6].deadband)) (Controller_Land_Mode=1); //非返航模式
93
94     if (PPM_Databuf[6] >= (RC_Data.cal[6].middle-RC_Data.cal[6].deadband)
95         && PPM_Databuf[6] <= (RC_Data.cal[6].middle+RC_Data.cal[6].deadband)) //遥控器三舵开关处于中位
96     {
97     }
98     else
99     {
100     }
101
102
103
104     if (PPM_Databuf[7] >= (RC_Data.cal[7].max-RC_Data.cal[7].deadband)) Controller_Horizontal_Mode=2; //水平位置控制
105     else if (PPM_Databuf[7] <= (RC_Data.cal[7].min+RC_Data.cal[7].deadband))
106     {
107         Controller_Horizontal_Mode=1; //姿态自稳控制
108         OpticalFlow_SINS_Reset();
109         OpticalFlow_Ctrl_Reset();
110     }
111
112     if (Unwanted_Lock_Flag==1) //定高模式解锁后，无任何操作
113     {
114         Thr_Push_Over_State=Thr_Push_Over_Deadband();
115         if (Thr_Push_Over_State==0) //只要向上推过了中位死区，即把允许自动上锁操作
116         {
117             Unwanted_Lock_Flag=0;
118         }
119         else
120         {
121             Take_Off_Reset(); //清除分
122             //Throttle_Control_Reset(); //清除分
123         }
124     }
125
126     if (Controller_Land_Mode!=Last_Controller_Land_Mode)
127     {
128         Total_Controller.High_Position_Control.Expect=NamelessQuad.Position[_YAW]; //将开关拨动瞬间的诱导高度设置为期望高度
129         if (Controller_Land_Mode==1) Land_Reset(); //返航模式切回正常模式
130     }
131
132     if (Controller_High_Mode!=Last_Controller_High_Mode)
133     {
134         if (Controller_High_Mode==2) (Control_Mode_Change=1); //自稳切定高，设置基准油门值，悬停高度
135         if (Controller_High_Mode==1) (Control_Mode_Change=1); //定高切自稳
136     }
137
138     if (Controller_Horizontal_Mode!=Last_Controller_Horizontal_Mode) //位置通道有切换
139     {
140         if (Controller_Horizontal_Mode==2) Control_Mode_Change=2;
141         if (Controller_Horizontal_Mode==1) Control_Mode_Change=2;
142     }
143
144     if (Control_Mode_Change==1) //存在定高模式切换，高度只设置一次
145     {
146         if (Controller_High_Mode==High_Hold_Mode) //本次为定高模式，即自稳切定高
147         {
148             High_Hold_Throttle=Throttle_Control; //保存当前油门值，只存一次
149             //.....将当前诱导位置估计值作为目标高度.....
150             Total_Controller.High_Position_Control.Expect=NamelessQuad.Position[_YAW]; //将开关拨动瞬间的诱导高度设置为期望高度
151         }
152         else //本次为自稳模式，即定高切自稳
153         {
154             //.....
155             //Throttle_Control_Reset();
156         }
157     }
158     Control_Mode_Change=0; //将模式切换位置0，有且仅处理一次
159     else if (Control_Mode_Change==2) //存在定点模式切换，悬停位置只设置一次
160     {
161         Control_Mode_Change=0; //已响应本次定点档位切换
162     }
163
164 }
```

③ 总控制器 Total_Control()函数

总控制器分为主导控制器与姿态控制器，其中主导控制器根据遥控器设置的当前模式和预设的 SDK 状态，确定执行哪一种控制模式，包括姿态自稳、定高、定点、降落、SDK 控制等。主导控制器最后输出包括油门控制、期望姿态角度、

角速度等。姿态控制器的输入由主导控制器给出，姿态控制器内部执行的是水平姿态角度+角速度控制。用户二次开发具体功能任务时，基本是都是在 `Main_Leading_Control()` 中的 `Auto_Flight_Ctrl()` 中添加代码。

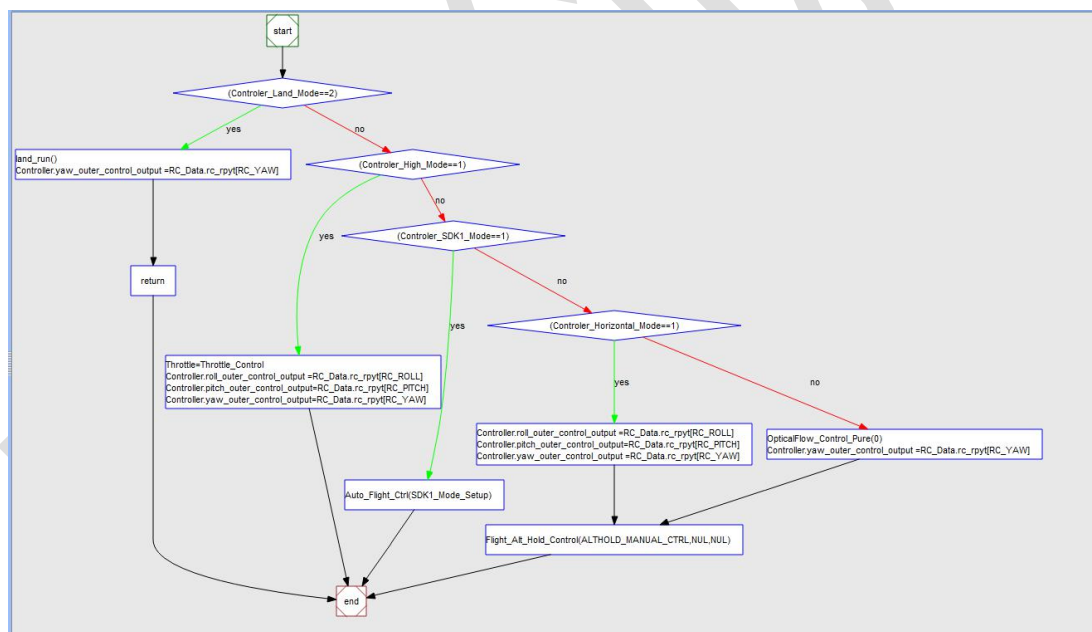
```
339 void Total_Control(void)
340 {
341     /*****主导控制器*****/
342     Main_Leading_Control();
343     /*****姿态环控制器*****/
344     Attitude_Control();
345 }
```




```

272 void Main_Leading_Control(void)
273 {
274     /*****根据遥控器切换档位，飞行进入不同模式*****/
275     if(Controleer_Land_Mode==2) //一键降落模式
276     {
277         land_run();
278         Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
279         return ;
280     }
281
282     /*****第五通道控制高度模式*****/
283     if(Controleer_High_Mode==1) //高度纯手动模式
284     {
285         Throttle=Throttle_Control; //油门直接来源于遥控器油门给定
286         Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
287         Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
288         Controller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
289     }
290     else
291     {
292         if(Controleer_SDK1_Mode==1) //用户SDK开发者自动飞行模式：水平+高度控制
293         {
294             Auto_Flight_Ctrl(SDK1_Mode_Setup);
295         }
296         else //普通定高、定点控制
297         {
298             /*****第六通道控制水平位置模式*****/
299             if(Controleer_Horizontal_Mode==1) //水平自稳
300             {
301                 Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
302                 Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
303                 Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
304             }
305             else //水平光流定点模式
306             {
307                 OpticalFlow_Control_Pure(0); //普通光流模式
308                 Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
309             }
310             Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL); //高度控制
311         }
312     }
}

```



在 Main_Leading_Control() 中一键降落拥有最高优先级，无论之前处于任何模式切第 7 通道到高位时，飞行器会执行自动降落函数，完成飞行器的降落。注意自动降落过程中遥控器对水平姿态、速度调节依然有效。

```
void Main_Leading_Control(void)
{
    /******根据遥控器切换档位，飞控进入不同模式******/
    if(Controller_Land_Mode==2)//一键降落模式
    {
        land_run();
        Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
        return ;
    }

    /******第五通道控制高度模式******/
    if(Controller_High_Mode==1)//高度纯手动模式
    {
        Throttle=Throttle_Control;//油门直接来源于遥控器油门给定
        Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
        Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
        Controller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
    }
    else
    {
        if(Controller_SDK1_Mode==1)//用户SDK开发者自动飞行模式：水平+高度控制
        {
            Auto_Flight_Ctrl(SDK1_Mode_Setup);
        }
        else//普通定高、定点控制
        {
            /******第六通道控制水平位置模式******/
            if(Controller_Horizontal_Mode==1)//水平自稳
            {
                Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
                Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
                Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
            }
            else//水平光流定点模式
            {
                OpticalFlow_Control_Pure(0);//普通光流模式
                Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
            }
            Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
        }
    }
}
```

当飞控不处于一键降落模式时，飞行器会根据 5、6、8 通道状态决策飞行模式。

- 高度手动控制模式时，油门控制、姿态期望角度、角速度直接来源于遥控器给定。用户需要实时手动调节油门、方向杆去控制无人机飞行（**不推荐新手操作**）。

```
void Main_Leading_Control(void)
{
    /******根据遥控器切换档位，飞控进入不同模式******/
    if(Controller_Land_Mode==2)//一键降落模式
    {
        land_run();
        Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
        return ;
    }

    /******第五通道控制高度模式******/
    if(Controller_High_Mode==1)//高度纯手动模式
    {
        Throttle=Throttle_Control;//油门直接来源于遥控器油门给定
        Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
        Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
        Controller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
    }
    else
    {
        if(Controller_SDK1_Mode==1)//用户SDK开发者自动飞行模式：水平+高度控制
        {
            Auto_Flight_Ctrl(SDK1_Mode_Setup);
        }
        else//普通定高、定点控制
        {
            /******第六通道控制水平位置模式******/
            if(Controller_Horizontal_Mode==1)//水平自稳
            {
                Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
                Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
                Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
            }
            else//水平光流定点模式
            {
                OpticalFlow_Control_Pure(0);//普通光流模式
                Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
            }
            Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
        }
    }
}
```

- 高度自动控制模式时，油门控制来源与竖直高度位置、速度、加速度控制器输出，无人机会根据当前状态实时调整油门控制量，自动保持高度或者期望速度飞行。

```
271 |
272 | void Main_Leading_Control(void)
273 | {
274 | /*****根据遥控器切换档位，飞控进入不同模式*****/
275 | if (Controller_Land_Mode==2) //一键降落模式
276 | {
277 |     land_run();
278 |     Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
279 |     return ;
280 | }
281 |
282 | /*****第五通道控制高度模式*****/
283 | if (Controller_High_Mode==1) //高度纯手动模式
284 | {
285 |     Throttle=Throttle_Control; //油门直接来源于遥控器油门给定
286 |     Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
287 |     Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
288 |     Controller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
289 | }
290 | else
291 | {
292 |     if (Controller_SDK1_Mode==1) //用户SDK开发者自动飞行模式：水平+高度控制
293 |     {
294 |         Auto_Flight_Ctrl (SDK1_Mode_Setup);
295 |     }
296 |     else //普通定高、定点控制
297 |     {
298 | /*****第六通道控制水平位置模式*****/
299 |         if (Controller_Horizontal_Mode==1) //水平自稳
300 |         {
301 |             Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
302 |             Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
303 |             Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
304 |         }
305 |         else //水平光流定点模式
306 |         {
307 |             OpticalFlow_Control_Pure(0); //普通光流模式
308 |             Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
309 |         }
310 |         Flight_Alt_Hold_Control (ALTHOLD_MANUAL_CTRL,NUL,NUL); //高度控制
311 |     }
312 | }
```

- 光流定点控制模式时，水平姿态控制期望来源与水平位置、速度制器输出，无人机会根据当前状态实时调整姿态期望输入，自动保持水平位置或者期望速度飞行。


```
271 -
272 void Main_Leading_Control(void)
273 {
274     /*****根据遥控器切换档位，飞控进入不同模式*****/
275     if(Controller_Land_Mode==2)//一键降落模式
276     {
277         land_run();
278         Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
279         return ;
280     }
281
282     /*****第五通道控制高度模式*****/
283     if(Controller_High_Mode==1)//高度纯手动模式
284     {
285         Throttle=Throttle_Control;//油门直接来源于遥控器油门给定
286         Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
287         Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
288         Controller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
289     }
290     else
291     {
292         if(Controller_SDK1_Mode==1)//用户SDK开发者自动飞行模式：水平+高度控制
293         {
294             Auto_Flight_Ctrl(SDK1_Mode_Setup);
295         }
296         else//普通定高、定点控制
297         {
298             /*****第六通道控制水平位置模式*****/
299             if(Controller_Horizontal_Mode==1)//水平自稳
300             {
301                 Controller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
302                 Controller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
303                 Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
304             }
305             else//水平光流定点模式
306             {
307                 OpticalFlow_Control_Pure(0);//普通光流模式
308                 Controller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
309             }
310             Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
311         }
312     }
```

- SDK 控制模式时，水平姿态控制期望来源与外部视觉位置控制、水平位置、速度制器输出，无人机会根据当前状态实时调整姿态期望输入，自动完成循迹、追踪物块、APrilTag 定位、自定义轨迹等自主飞行任务。


```

272 void Main_Leading_Control(void)
273 {
274     /*****根据遥控器切换档位，飞控进入不同模式*****/
275     if(Contoller_Land_Mode==2)//一键降落模式
276     {
277         land_run();
278         Contoller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
279         return ;
280     }
281
282     /*****第五通道控制高度模式*****/
283     if(Contoller_High_Mode==1)//高度纯手动模式
284     {
285         Throttle=Throttle_Control;//油门直接来源于遥控器油门给定
286         Contoller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
287         Contoller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
288         Contoller.yaw_outer_control_output=RC_Data.rc_rpyt[RC_YAW];
289     }
290     else
291     {
292         if(Contoller_SDK1_Mode==1)//用户SDK开发者自动飞行模式：水平+高度控制
293         {
294             Auto_Flight_Ctrl(SDK1_Mode_Setup);
295         }
296         else//普通定高、定点控制
297         {
298             /*****第六通道控制水平位置模式*****/
299             if(Contoller_Horizontal_Mode==1)//水平自稳
300             {
301                 Contoller.roll_outer_control_output =RC_Data.rc_rpyt[RC_ROLL];
302                 Contoller.pitch_outer_control_output=RC_Data.rc_rpyt[RC_PITCH];
303                 Contoller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
304             }
305             else//水平光流定点模式
306             {
307                 OpticalFlow_Control_Pure(0);//普通光流模式
308                 Contoller.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
309             }
310             Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
311         }
312     }

```

