

```

1 #ifndef _FLIGHT_CONTROL_H_
2 #define _FLIGHT_CONTROL_H_
3
4 #include "stm32f4xx_hal.h"
5 #include "quaternion.h"
6 #include "ahrs.h"
7 #include "motor.h"
8
9 #ifdef MOTOR_DC
10 #define MIN_THR          200          /* DC motor configuration */
11 #endif
12 #ifdef MOTOR_ESC
13 #define MIN_THR          20          /* External ESC configuration */
14 #endif
15
16 #define MAX_ADJ_AMOUNT    800
17 #define MAX_ADJ_AMOUNT_YAW 800
18 #define MAX_ACC_DIFF     10
19 #define MAX_ACC_DATA     125
20
21 #define GYRO_FIL_K        0.05
22 #define ACC_FIL_K         0.05
23 #define MAG_FIL_K         1.0
24 #define EULER_FIL_K      0.001
25 #define EULER_OFFSET_LIMIT 0.015      //5 degree 0.09
26 #define EULER_NEAR_ZERO  0.005      //1 degree 0.01745
27
28 #define ROLL_PID_KP1      3
29 #define ROLL_PID_KI1      0
30 // #define ROLL_PID_KP2    800      /* default */
31 #define ROLL_PID_KP2      100      /* test minidrone */
32 // #define ROLL_PID_KP2    200      /* test FPV250 with STEVAL-ESC001V1 */
33 // #define ROLL_PID_KP2    40       /* test FPV250 with Afro ESC12A */
34 // #define ROLL_PID_KI2    400      /* default */
35 #define ROLL_PID_KI2      100      /* test minidrone */
36 // #define ROLL_PID_KI2    200      /* test FPV250 with STEVAL-ESC001V1 */
37 // #define ROLL_PID_KI2    40       /* test FPV250 with Afro ESC12A */
38 #define ROLL_PID_KD2      10       //(x/PID_SAMPLING_TIME)
39 // #define ROLL_PID_KD2    5        /* test FPV250 with STEVAL-ESC001V1 */
40 #define ROLL_PID_I1_LIMIT 2.0      //5 degree
41 // #define ROLL_PID_I2_LIMIT (100/ROLL_PID_KI2)
42 #define ROLL_PID_I2_LIMIT 20.0
43 // #define ROLL_PID_I2_LIMIT 2
44
45
46 #define PITCH_PID_KP1     ROLL_PID_KP1
47 #define PITCH_PID_KI1     ROLL_PID_KI1
48 #define PITCH_PID_KP2     ROLL_PID_KP2
49 #define PITCH_PID_KI2     ROLL_PID_KI2
50 #define PITCH_PID_KD2     ROLL_PID_KD2

```

```

51 #define PITCH_PID_I1_LIMIT    ROLL_PID_I1_LIMIT
52 #define PITCH_PID_I2_LIMIT    ROLL_PID_I2_LIMIT
53
54 // #define YAW_PID_KP1        4.0
55 // #define YAW_PID_KI1        0.0
56 // #define YAW_PID_KP2        1500
57 // #define YAW_PID_KI2        1500
58 // #define YAW_PID_KD2        0.0
59 // #define YAW_PID_I1_LIMIT    50.0
60 // #define YAW_PID_I2_LIMIT    0.1    //6 degree/s
61 #define YAW_PID_KP1           4
62 #define YAW_PID_KI1           0
63 #define YAW_PID_KP2           1000
64 #define YAW_PID_KI2           0
65 #define YAW_PID_KD2           0        //(x/PID_SAMPLING_TIME)
66 #define YAW_PID_I1_LIMIT      2.0        //5 degree
67 // #define YAW_PID_I2_LIMIT    (100.0/YAW_PID_KI2)
68 #define YAW_PID_I2_LIMIT      2
69
70 #define PID_SAMPLING_TIME     0.00125
71
72 #define D_FILTER_COFF        0.025f
73
74 #define MOTOR_OFF1           0
75 #define MOTOR_OFF2           0
76 #define MOTOR_OFF3           0
77 #define MOTOR_OFF4           0
78
79 #define X_AXIS_OFFSET         -0.056
80 #define Y_AXIS_OFFSET         -0.029
81 // #define X_AXIS_OFFSET         -0.08
82 // #define Y_AXIS_OFFSET         0
83
84 #define GYRO_OFFSET_X         0.094
85 #define GYRO_OFFSET_Y         0.060
86 #define GYRO_OFFSET_Z         -0.064
87
88 #define FIFO_Order           5
89 #define MID_FIFO              (FIFO_Order>>1)
90 #define FIFO_Order_Recip     (1.0/FIFO_Order)
91
92 // Structure for P-PI type PID control
93 // first stage is angle stage, second stage is angle rate stage.
94 typedef struct
95 {
96     float ts;                    // sampling time
97     float x_kp1, x_ki1, x_kp2, x_ki2, x_kd2;    // stage pid parameter
98     float y_kp1, y_ki1, y_kp2, y_ki2, y_kd2;
99     float z_kp1, z_ki1, z_kp2, z_ki2, z_kd2;
100    float x_i1_limit, y_i1_limit, z_i1_limit;

```

```

101 float x_i2_limit, y_i2_limit, z_i2_limit;
102 float x_s1, x_s2; // stage output
103 float y_s1, y_s2;
104 float z_s1, z_s2;
105 }P_PI_PIDControlTypeDef;
106
107 void FlightControlPID(EulerAngleTypeDef *euler_rc, EulerAngleTypeDef *euler_ahrs, Gyro_Rad
    *gyro_rad, AHRS_State_TypeDef *ahrs, P_PI_PIDControlTypeDef *pid, MotorControlTypeDef
    *motor_pwm);
108 void FlightControlPID_OuterLoop(EulerAngleTypeDef *euler_rc, EulerAngleTypeDef
    *euler_ahrs, AHRS_State_TypeDef *ahrs, P_PI_PIDControlTypeDef *pid);
109 void FlightControlPID_innerLoop(EulerAngleTypeDef *euler_rc, Gyro_Rad *gyro_rad,
    AHRS_State_TypeDef *ahrs, P_PI_PIDControlTypeDef *pid, MotorControlTypeDef *motor_pwm);
110 void PIDControlInit(P_PI_PIDControlTypeDef *pid);
111 void PIDOuterLoopFrameTrans(P_PI_PIDControlTypeDef *pid, EulerAngleTypeDef *euler_ahrs);
112
113
114
115 #define RP_RCTRL_FA { ¥
116     -53032.F, 0.F, -107.592F, 0.F, ¥
117     0.F, -53032.F, 0.F, -107.592F ¥
118 }
119 #define RP_RCTRL_GA { ¥
120     100.F, 0.F, ¥
121     0.F, 100.F ¥
122 }
123 #define RP_RCTRL_HA { ¥
124     100.268F, 0.F, ¥
125     0.F, 100.268F ¥
126 }
127 #define RP_RCTRL_AOD { ¥
128     0.987578F, 0.F, ¥
129     0.F, 0.987578F ¥
130 }
131 #define RP_RCTRL_BOD { ¥
132     4.81981e-007F, 0.F, ¥
133     0.F, 4.81981e-007F ¥
134 }
135 #define RP_RCTRL_COD { ¥
136     1.F, 0.F, ¥
137     0.F, 1.F ¥
138 }
139 #define EGX_I_LIMIT (20.F)
140 #define EGY_I_LIMIT (20.F)
141 #define X_S2_LIMIT_0 (10.F)
142 #define Y_S2_LIMIT_0 (10.F)
143
144
145
146

```

追加：状態フィードバック制御に使用するパラメータの値

```
147 #endif
```

```
148
```