**部分代码：**

/\*!

\* program for 四旋翼飞行器

\*

\* @file main.c

\* @brief stm32f103 平台

\* @author J-pcy

\* @version v1.0 平稳飞行

\* @date 2014-01-9

\*/

/\* Includes ------------------------------------------------------------------\*/

#include "config.h"

\_\_IO u8 MS5611\_Cal\_Flag = 0;

\_\_IO u8 AK8975\_Cal\_Flag = 0;

\_\_IO u16 DEBUG\_CNT;

int main(void)

{

Usart1\_Init(115200);

Usart2\_Init(19200);

Tim\_Pwm\_In\_Init(); //输入捕获

Tim\_Pwm\_Out\_Init();

Timer\_Sys\_Init(500);

I2C1\_RESET();

I2C1\_INIT(0X00,400000,1,1,1,1);

Led\_Init();

Spi1\_Init();

Nvic\_Init();

Nrf24l01\_Init(MODEL\_TX2,40);

if(Nrf24l01\_Check()) Uart1\_Put\_String("NRF24L01 IS OK !\r\n");

else Uart1\_Put\_String("NRF24L01 IS NOT OK !\r\n");

MPU6050\_Init();

Uart1\_Put\_String("Welcome to ANO TC!\r\n");

if(SD\_Card\_Init())

Uart1\_Put\_String("SD\_Card OK!\r\n");

else

Uart1\_Put\_String("SD\_Card Error!\r\n");

Data\_Read();

while (1)

{

if(AK8975\_Cal\_Flag)

{

AK8975\_Cal\_Flag = 0;

AK8975\_Cal();

}

if(MS5611\_Cal\_Flag)

{

MS5611\_Cal\_Flag = 0;

MS5611\_Cal();

}

if(Data\_Check)

{

Data\_Check = 0;

Send\_Data();

}

}

}

/\*!

\* program for 四旋翼飞行器

\*

\* @file Control.c

\* @brief stm32f103 平台

\* @author J-pcy

\* @version v1.0 平稳飞行

\* @date 2014-01-9

\*/

#include "control.h"

#include "Rc.h"

#include "imu.h"

#include "tim\_pwm\_out.h"

PID PID\_ROL,PID\_PIT,PID\_YAW,PID\_ALT,PID\_POS,PID\_PID\_1,PID\_PID\_2;

int16\_t getlast\_roll=0,geilast\_pitch=0;

float rol\_i=0,pit\_i=0,yaw\_p=0;

vs16 Moto\_PWM\_1=0,Moto\_PWM\_2=0,Moto\_PWM\_3=0,Moto\_PWM\_4=0,Moto\_PWM\_5=0,Moto\_PWM\_6=0,Moto\_PWM\_7=0,Moto\_PWM\_8=0;

void CONTROL(float rol, float pit, float yaw)

{

rol=rol-(Rc\_Data.ROLL-1500)/20;

rol\_i+=rol;

if(rol\_i>2000)

rol\_i=2000;

if(rol\_i<-2000)

rol\_i=-2000;

PID\_ROL.pout = PID\_ROL.P \* rol;

PID\_ROL.dout = PID\_ROL.D \* MPU6050\_GYRO\_LAST.X;

PID\_ROL.iout = PID\_ROL.I \* rol\_i;

pit=pit+(Rc\_Data.PITCH-1500)/20;

pit\_i+=pit;

if(pit\_i>2000)

pit\_i=2000;

if(pit\_i<-2000)

pit\_i=-2000;

PID\_PIT.pout = PID\_PIT.P \* pit;

PID\_PIT.dout = PID\_PIT.D \* MPU6050\_GYRO\_LAST.Y;

PID\_PIT.iout = PID\_PIT.I \* pit\_i;

if(Rc\_Data.YAW<1480||Rc\_Data.YAW>1520)

{MPU6050\_GYRO\_LAST.Z=MPU6050\_GYRO\_LAST.Z+(Rc\_Data.YAW-1500)\*10;}

yaw\_p+=MPU6050\_GYRO\_LAST.Z\*0.0609756f\*0.002f;// +(Rc\_Get.YAW-1500)\*30

PID\_YAW.pout=PID\_YAW.P\*yaw\_p;

PID\_YAW.dout = PID\_YAW.D \* MPU6050\_GYRO\_LAST.Z;

if(Rc\_Data.THROTTLE<1200)

{

pit\_i=0;

rol\_i=0;

yaw\_p=0;

}

PID\_ROL.OUT = (-PID\_ROL.pout) - PID\_ROL.iout +PID\_ROL.dout;

PID\_PIT.OUT = PID\_PIT.pout + PID\_PIT.iout + PID\_PIT.dout;

PID\_YAW.OUT = PID\_YAW.pout + PID\_YAW.iout + PID\_YAW.dout;

if(Rc\_Data.THROTTLE>1200&&ARMED)

{

Moto\_PWM\_1 = Rc\_Data.THROTTLE - 1000 - PID\_ROL.OUT - PID\_PIT.OUT + PID\_YAW.OUT;

Moto\_PWM\_2 = Rc\_Data.THROTTLE - 1000 + PID\_ROL.OUT - PID\_PIT.OUT - PID\_YAW.OUT;

Moto\_PWM\_3 = Rc\_Data.THROTTLE - 1000 + PID\_ROL.OUT + PID\_PIT.OUT + PID\_YAW.OUT;

Moto\_PWM\_4 = Rc\_Data.THROTTLE - 1000 - PID\_ROL.OUT + PID\_PIT.OUT - PID\_YAW.OUT;

}

else

{

Moto\_PWM\_1 = 0;

Moto\_PWM\_2 = 0;

Moto\_PWM\_3 = 0;

Moto\_PWM\_4 = 0;

}

Moto\_PwmRflash(Moto\_PWM\_1,Moto\_PWM\_2,Moto\_PWM\_3,Moto\_PWM\_4);

}