sketch\_241211SFO4servoCODECh8RearWElevTrim

//231011 version3 for ZIP added Rear wing Aileron same move 221208 Version2 added ch7 RearWingTrim　210916 15-4th Code 4servo Rudder ElevatorUP FlapAmpUP not delaytime control PPMRX Ch1-4 writeMicroseconds(1000-2000uS) use 5V3A  FrontServoLate RearServo Ailron act only Front servo dt and elapsed time by K.Kakuta

//241211 Ch8 rearWElevtrim

#include <Servo.h>

#include "src/PPMReader/PPMReader.h"// <PPMReader.h>

//#include <InterruptHandler.h>// 2022/01/27 Delete for more good move

//#define DOPRINTS // if you print on PC screen Data, delete”//”

// In use of Flap system,cut "Serial.print "(put ”//”)

//and write code to Arduino board.

int interruptPin = 2;

int channelAmount = 8;

PPMReader ppm(interruptPin, channelAmount);

int servo\_right\_pin = 5;

int servo\_left\_pin = 6;

int servo\_right2\_pin = 8;

int servo\_left2\_pin = 9;

volatile int elevator = 0;

volatile int flapamp = 0;

volatile int rearwingtrim =0;

volatile int rearWElevtrim =0;

volatile float delaytime = 100;// Servo speed low-increase Servo speed high-decrease this.

                          // unit: micro second

volatile float delaytime2 = 100;

volatile float delaytime3=100; // Servo speed low-increase Servo speed high-decrease this.

                          // unit: micro second

volatile float delaytime4 = 100;

float elapsed\_time = 0;

float dt;

unsigned long current\_time, prev\_time;

volatile int ch3value = 1000;//Ch3

volatile int ch1value = 1500;//Ch1

volatile int ch2value = 1500;//Ch2

volatile int ch4value = 1500;//Ch4

volatile int ch5value = 1500;//Ch5

volatile int ch6value = 1500;//Ch6 Right LeverSwitch

volatile int ch7value = 1500;//Ch7 rear wing aileron trim

volatile int ch8value = 1500;//Ch8 rear wing elevator trim

static int servo\_comm1 = 0;// Left or Right Servo high point and low point

static int servo\_comm2 = 0; // Left or Right Servo high point and low point

static int servo\_comm3 = 0;// Left or Right Servo high point and low point no rudder

static int servo\_comm4 = 0; // Left or Right Servo high point and low point no rudder

volatile int rudder = 0;

float glide\_deg = 0; // Gliding angle 0=0 degree 500=90degree

static float servo\_zero1 = 0;//flap angle adjust

static float servo\_zero2 = 0; //flap angle adjust

Servo servo\_left, servo\_right, servo\_right2, servo\_left2; // create servo object to control a servo

void setup() {

  Serial.begin(9600);

  pinMode(servo\_left\_pin, OUTPUT);

  pinMode(servo\_right\_pin, OUTPUT);

  pinMode(servo\_left2\_pin, OUTPUT);

  pinMode(servo\_right2\_pin, OUTPUT);

  servo\_left.attach(servo\_left\_pin);//output pin No

  servo\_right.attach(servo\_right\_pin); //output pin No

  servo\_left2.attach(servo\_left2\_pin);//output pin No

  servo\_right2.attach(servo\_right2\_pin); //output pin No

delay(2000);//Avoid abnormal positions at startup-wait 2 second until RX starts220306

}

void loop() {

prev\_time = current\_time;

  current\_time = micros();

  dt = (current\_time - prev\_time)/1000000.0;

  elapsed\_time = elapsed\_time + dt; // total time spent in the main loop since beginning one upstroke/downstroke

  ch3value = ppm.rawChannelValue(3);//Ch3

  ch1value = ppm.rawChannelValue(1);//Ch1

  ch2value = ppm.rawChannelValue(2);//Ch2

  ch4value = ppm.rawChannelValue(4);//Ch4

  ch5value = ppm.rawChannelValue(5);//Ch5

ch6value = ppm.rawChannelValue(6);//Ch6

ch7value = ppm.rawChannelValue(7);//Ch7

ch8value = ppm.rawChannelValue(8);//Ch8

    //Serial.print("ch3value ");Serial.print(ch3value);

    //Serial.print(",\t");

    //Serial.print("ch1value ");Serial.print(ch1value);

    //Serial.print(",\t");

    //Serial.print("ch2value ");Serial.print(ch2value);

//Serial.print(",\t");

    //Serial.print("ch4value ");Serial.print(ch4value);

//Serial.print(",\t");

    //Serial.print("ch5value ");Serial.print(ch5value);

    //Serial.println(",\t");

    //Serial.print("ch6value ");Serial.print(ch6value);

    //Serial.println(",\t");

 rudder=(int)(ch1value-1500);//Ch1  Flap angle incline

 elevator=(int)(ch2value-1500);//Ch2 Flap Angle bilateral UP&Down

 flapamp=(int)(ch4value-1500);//Ch4 Right and left Flap angle difference

delaytime=( ch5value -950)/5;//Ch5 Flapping frequency

 delaytime2=( ch6value -950)/10;//Ch6 Delay frontServo-RearServo

 rearwingtrim =( ch7value - 1500);//Ch7 Right and left angle trim difference -0.5to0.5

 rearWElevtrim =( ch8value - 1500);//Ch8 Right and left angle trim -0.5to0.5

delaytime4 = delaytime \* delaytime2 / 200;

delaytime3 = delaytime- delaytime4;

// you can change UP or Down direction by your transmitter Reverse setting of each Channel

    //Serial.print("rudder");Serial.print(rudder);

    //Serial.print(",\t");

    //Serial.print("elevator");Serial.print(elevator);

    //Serial.print(",\t");

    //Serial.print("flapamp");Serial.print(flapamp);

//Serial.print(",\t");

    //Serial.print("delaytime");Serial.print(delaytime);

    //Serial.print(",\t");

//Serial.print("delaytime2");Serial.print(delaytime);

    //Serial.print(",\t");

 if (ch3value > 1080) {

if (elapsed\_time < delaytime4/1000) {

  servo\_comm1 = (int)( (ch3value -1000)/2+1500 + rudder - elevator+ servo\_zero1+ flapamp);

  servo\_comm2 = (int)(1000 + (2000 - ((ch3value -1000)/2+1500)) + rudder + elevator- servo\_zero2+flapamp);

  servo\_comm3 = (int)( (ch3value -1000)/2+1500 + rudder - elevator + rearwingtrim - rearWElevtrim + servo\_zero1+ flapamp);

  servo\_comm4 = (int)(1000 + (2000 - ((ch3value -1000)/2+1500)) + rudder + elevator + rearwingtrim + rearWElevtrim - servo\_zero2+flapamp);

   // Serial.print("servo\_comm1 ");Serial.print(servo\_comm1);

   // Serial.print(",\t");

   // Serial.print("servo\_comm2 ");Serial.print(servo\_comm2);

   // Serial.print(",\t");

   // Serial.print("servo\_comm3 ");Serial.print(servo\_comm3);

//Serial.print(",\t");

   // Serial.print("servo\_comm4 ");Serial.print(servo\_comm4);

  servo\_left2.writeMicroseconds(servo\_comm3); // Rear left servo position

  servo\_right2.writeMicroseconds(servo\_comm4); // Rear right servo position

}

if ((elapsed\_time > delaytime4/1000) && ( elapsed\_time < (delaytime4 + delaytime3)/1000)) {

  servo\_left.writeMicroseconds(servo\_comm1); // Front left servo position

  servo\_right.writeMicroseconds(servo\_comm2); // Front right servo position

}

if ((elapsed\_time > (delaytime4 + delaytime3)/1000) && (elapsed\_time < (delaytime4 + delaytime3 + delaytime4)/1000)) {

  servo\_comm1 = (int)( (ch3value -1000)/2+1500 + rudder + elevator+ servo\_zero1-flapamp);

  servo\_comm2 = (int)(1000 + (2000 - ((ch3value -1000)/2+1500)) + rudder - elevator- servo\_zero2-flapamp);

  servo\_comm3 = (int)( (ch3value -1000)/2+1500 + rudder + elevator+ rearwingtrim + rearWElevtrim + servo\_zero1-flapamp);

  servo\_comm4 = (int)(1000 + (2000 - ((ch3value -1000)/2+1500)) + rudder - elevator + rearwingtrim - rearWElevtrim - servo\_zero2-flapamp);

  servo\_left2.writeMicroseconds(servo\_comm4); // Rear left servo position

  servo\_right2.writeMicroseconds(servo\_comm3); // Rear right servo position

}

   if (elapsed\_time > (delaytime4 + delaytime3 + delaytime4)/1000)  {

  servo\_left.writeMicroseconds(servo\_comm2); // Front left servo position

  servo\_right.writeMicroseconds(servo\_comm1); // Front right servo position

    //Serial.print("servo\_comm1");Serial.print(servo\_comm1);

    //Serial.print(",\t");

    //Serial.print("servo\_comm2");Serial.print(servo\_comm2);

    //Serial.println(",\t");

  }

  }

else{

  servo\_comm1=(int)(1500+rudder-elevator+glide\_deg);

  servo\_comm2=(int)(1500+rudder+elevator-glide\_deg);

servo\_comm3=(int)(1500 + rearwingtrim - rearWElevtrim + rudder -elevator+glide\_deg);

  servo\_comm4=(int)(1500 + rearwingtrim + rearWElevtrim + rudder +elevator-glide\_deg);

  servo\_left.writeMicroseconds(servo\_comm1); // servo position in variable 'pos'

  servo\_right.writeMicroseconds(servo\_comm2); // servo position in variable 'pos'

servo\_left2.writeMicroseconds(servo\_comm3); // servo position in variable 'pos'

  servo\_right2.writeMicroseconds(servo\_comm4); // servo position in variable 'pos'

    //Serial.print("servo\_comm1");Serial.print(servo\_comm1);

    //Serial.print(",\t");

    //Serial.print("servo\_comm2");Serial.print(servo\_comm2);

    //Serial.println(",\t");

}

       if (elapsed\_time > (delaytime4 + delaytime3 + delaytime4 + delaytime3)/1000) {// one full flap is finisheddt

         elapsed\_time = 0; // start next flapping cycle

}

}