

Plane Autopilot

Aim of project: To stabilize plane in air from unwelcomed wind

Remark: This code is still lagging UI and remote control.

Code:

```
/*
```

Algorithm:

- 1.Start
- 2.Callibrate Gyro and set initial angle all axes x, y, z to 0
- 3.Get gyro reading
- 4.Adjust Pitch, Roll and Yaw Servo accordingly to keep all x, y, z angles 0
- 5.Stop

Additional Features(have bugs):

- 1)pre command plane to steer.

EX-

- 1)Start
- 2)Forward(100)
- 3)Turn Left(90 degree)
- 4)Forward(100)
- 5)Stop

This will turn plane left, similarly more steps can be imitated

```
*/
```

```
//Including Libraries
```

```
#include <Servo.h>
```

```
#include <Wire.h>
```

```
#include <MPU6050_light.h>
```

```
//Defining Instances
```

```
MPU6050 mpu(Wire);
```

```
Servo rollControl;
```

```
Servo pitchControl;
```

```
Servo yawControl;
```

```
//Defining Pins
```

```
const int rollPin=5;
```

```
const int pitchPin=6;
```

```
const int yawPin=8;
```

```
const int sigPin=4;
```

```
const int rollInputPin=2;
```

```
const int pitchInputPin=3;
```

```
//Defining Variable
```

```
const int rollExtreme=30;
```

```
const int pitchExtreme=30;
```

```
const int yawExtreme=30;
```

```
const int initialRollPosition=90;
```

```
const int initialPitchPosition=90;
```

```
const int initialYawPosition=90;
```

```
float rollY;
```

```
float pitchZ;
```

```
float yawX;
```

```
int state;
```

```
float currentYaw;
```

```
float pitchError;
```

```
int pitchControlAngle;
```

```
float rollError;
```

```
int rollControlAngle;

int com=0;


//test

unsigned long rollInput,pitchInput,sigData;


void setup() {

    //Initializing of Components

    //Control Servos

    rollControl.attach(rollPin);
    pitchControl.attach(pitchPin);
    yawControl.attach(yawPin);


    //Serial Monitor

    Serial.begin(9600);


    //Libraries

    Wire.begin();

    mpu.begin();

    //Calibration

    mpu.calcOffsets();


    //Defining Pins

    pinMode(sigPin,INPUT);
    pinMode(rollInputPin,INPUT);
    pinMode(pitchInputPin,INPUT);
}


void loop() {
```

```

//NEw

sigData=pulseIn(sigPin, HIGH);
if(sigData>1000){
rollInput = pulseIn(rollInputPin, HIGH);
pitchInput =pulseIn(pitchInputPin,HIGH);
if(pitchInput>1000){
    pitchControlAngle=map(pitchInput,993,1800,60,120);
    pitchControlAngle=constrain(pitchControlAngle,60,120);
    pitchControl.write(pitchControlAngle);

// rollControlAngle=map(rollInput,1022,1919,60,120);
//// Serial.println(rollInput);
//// Serial.println(rollControlAngle);
// rollControlAngle=constrain(rollControlAngle,60,120);
// rollControl.write(rollControlAngle);
}

if(rollInput>1000){
    rollControlAngle=map(rollInput,1022,1919,60,120);
// Serial.println(rollInput);
// Serial.println(rollControlAngle);
    rollControlAngle=constrain(rollControlAngle,60,120);
    rollControl.write(rollControlAngle);
}
}

//rollControl.write(70);
// pitchControl.write(90);

//To Test

//printRawMPU();

//meanPosition();

```

```

//rollControl.write(70);

//pitchControl.write(70); //to go up

//printServoAngles();

//pitchErrorCalc();

//left();

//ch4 yaw

//state=digitalRead(sigPin);
//while(state==0){
// printServoAngles();
// state=digitalRead(sigPin);
//}

// //main code

// state=digitalRead(sigPin);
//if(state==0){
//left();
//meanPosition();
//delay(1)
//if (com == 0){
// right();
// meanPosition();
// com++;
//}

// //Do not Delete
//}

//Some Delay for smooth Functioning
//delay(10);

}

void printRawMPU(){
    mpu.update();

    //Roll Angle
    rollY = mpu.getAngleY();

```

```

//Pitch Angle

pitchZ = mpu.getAngleZ();

//Yaw Angle

yawX = mpu.getAngleX();

Serial.print("Roll: ");

Serial.print(rollY);

Serial.print("\t");

Serial.print("Pitch: ");

Serial.print(pitchZ);

Serial.print("\t");

Serial.print("Yaw: ");

Serial.println(yawX);
}

void right(){
  mpu.update();

  yawX = mpu.getAngleX();

  currentYaw =yawX;

  while(yawX>currentYaw-80){

    //Servo Control

    rollControl.write(70);

    pitchControl.write(75);

    //=====

    mpu.update();                                //Do not touch

    yawX = mpu.getAngleX();

    //=====

  }

  Serial.println("Turned Right");

```

```
}
```

```
void left(){  
    mpu.update();  
    yawX = mpu.getAngleX();  
    currentYaw =yawX;  
    while(yawX<currentYaw+80){  
        //Servo Control  
        rollControl.write(110);  
        pitchControl.write(100);  
  
        //=====  
        mpu.update();                //Do not touch  
        yawX = mpu.getAngleX();  
        //=====  
    }  
    Serial.println("Turned Left");  
}
```

```
void pitchErrorCalc(){  
    mpu.update();  
  
    pitchZ =mpu.getAngleZ();  
    pitchError=pitchZ;  
  
    pitchControlAngle=map(pitchError,-pitchExtreme,pitchExtreme,initialPitchPosition-  
pitchExtreme,initialPitchPosition+pitchExtreme);  
  
    pitchControlAngle=constrain(pitchControlAngle,initialPitchPosition-  
pitchExtreme,initialPitchPosition+pitchExtreme);  
  
    pitchControl.write(pitchControlAngle);  
}  
void rollErrorCalc(){
```

```
mpu.update();
```

```
rollY =mpu.getAngleY();
```

```
rollError=rollY;
```

```
rollControlAngle=map(rollY,-rollExtreme,rollExtreme,initialRollPosition-  
rollExtreme,initialRollPosition+rollExtreme);
```

```
rollControlAngle=constrain(rollControlAngle,initialRollPosition-  
rollExtreme,initialRollPosition+rollExtreme);
```

```
rollControl.write(rollControlAngle);
```

```
}
```

```
void printServoAngles(){
```

```
pitchErrorCalc();
```

```
rollErrorCalc();
```

```
//Roll
```

```
Serial.print("rollControlAngle:");
```

```
Serial.println(rollControlAngle);
```

```
//Pitch
```

```
Serial.print("\tPitchControlAngle:");
```

```
Serial.print(pitchControlAngle);
```

```
}
```

```
void meanPosition(){
```

```
pitchErrorCalc();
```

```
rollErrorCalc();
```

```
}
```

```
//V.9.0
```