```
#include <Wire.h>
#include <nRF24L01.h>
#include <I2Cdev.h>
#include <MPU6050.h>
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
#define Gry_offset 0
#define Gyr_Gain 0.00763358
#define Angle_offset 2.23
#define RMotor_offset 13.0
#define LMotor_offset 13.0
#define pi 3.14159
long data;
int x, y;
float kp, ki, kd;
float r_angle, f_angle, omega;
float LOutput,ROutput;
unsigned long preTime = 0;
float SampleTime = 0.08;
unsigned long lastTime;
float Input, Output;
float errSum, dErr, error, lastErr;
int timeChange;
int TN1=3;
int TN2=4;
int ENA=9;
int TN3=5;
int TN4=6;
int ENB=10;
void setup() {
Wire.begin();
accelgyro.initialize();
pinMode(TN1,OUTPUT);
pinMode(TN2,OUTPUT);
pinMode(TN3,OUTPUT);
pinMode(TN4,OUTPUT);
pinMode(ENA,OUTPUT);
pinMode(ENB,OUTPUT);
Serial.begin(115200);
}
void loop() {
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
r_angle = (atan2(ay, az) * 180 / pi + Angle_offset); Serial.print(" r_angle=");
Serial.print(r_angle);
omega = Gyr_Gain * (gx + Gry_offset); Serial.print(" omega="); Serial.
print(omega);
if (abs(r_angle)<40){
myPID();
PWMControl();
}
else{
analogWrite(ENA, 0);
analogWrite(ENB, 0);
}
}
void myPID(){
kp = 25.0; Serial.print ( "kp ="); Serial.print (kp); // 20
kd = 4.0; Serial.print ( "kd ="); Serial.print (kd); // 1000
ki = 0.1; Serial.print(" ki=");Serial.print(ki);
unsigned long now = millis();
float dt = (now - preTime) / 1000.0;
preTime = now;
float K = 0.8;
float A = K / (K + dt);
f_angle = A * (f_angle + omega * dt) + (1 - A) * r_angle; Serial.print("
f_angle=");Serial.print(f_angle);
timeChange = (now - lastTime);
if(timeChange >= SampleTime){
Input = f_angle;
error = Input;
errSum += error * timeChange;
dErr = (error - lastErr) / timeChange;
Output = kp * error + ki * errSum + kd * dErr;
LOutput = Output ; Serial.print(" LOutput=");Serial.print(LOutput);
ROutput = Output ; Serial.print(" ROutput=");Serial.println(ROutput);
lastErr = error ;
lastTime = now ;
errSum =0;
error =0;
} }
void PWMControl(){
if(LOutput > 0){
digitalWrite(TN1, HIGH);
digitalWrite(TN2, LOW);
}
else if(LOutput < 0){
  digitalWrite(TN1, LOW);
  digitalWrite(TN2, HIGH);
}
else{
digitalWrite(TN1, HIGH);
digitalWrite(TN2, HIGH);
}
if(ROutput > 0){
digitalWrite(TN3, HIGH);
digitalWrite(TN4, LOW);
}
```

```
else if(ROutput < 0){
digitalWrite(TN3, LOW);
digitalWrite(TN4, HIGH);
}
else{
digitalWrite(TN3, HIGH);
digitalWrite(TN4, HIGH);
}
analogWrite(ENA, min(255, abs(LOutput) + LMotor_offset));
analogWrite(ENB, min(255, abs(ROutput) + RMotor_offset));
}
}
```