```
//Einbinden der Bibliotheken


#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <Servo.h>
#include <EEPROM.h>

Adafruit_MPU6050 mpu;

Servo myservo;
Servo myservo2;
//Definieren Servos

int pos = 0;
int pos2 = 0;

//SETTINGS  Diese Variablen sind für den Betrieb
einzustellen
float filterfaktor = 0.1;
float gyroswitch = 0.1;
int xoffset = -120;
int yoffset = 0;
float gyrodriftx;
float gyrodrifty;

bool gyroswitchflag = false;

float xvals[3];
float yvals[3];
```

```
float diffaccx = 0;
float diffaccy = 0;
float diffgyrox = 0;
float diffgyroy = 0;

double cycletime;


void setup(void) {

  Serial.begin(115200);

  myservo.attach(9);
  myservo2.attach(6);

//Servos werden eingerichtet, der MPU6050 Chip
verbunden

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050
chip");
    while (1) {
      delay(10);
    }
  }
  mpu.
setAccelerometerRange(MPU6050_RANGE_8_G);
//Erdbeschleunigung
  mpu.
setGyroRange(MPU6050_RANGE_250_DEG);
//250°/s beschleunigung
```

```
  mpu.
setFilterBandwidth(MPU6050_BAND_5_HZ);
//zusätzlicher Filter

}

void loop() {

  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  //erfassung eines Sensor-events und
abspeicherung in den Variablen

  if (!gyroswitchflag) {
    xvals[1] = accx_filtered(a.acceleration.x);
    yvals[1] = accy_filtered(a.acceleration.y);
  }


//dient dazu bei stillstand den Gyro Drift zu
erfassen und zu filtern

  if (((abs(a.acceleration.x) - diffaccx) == 0)
&& ((abs(a.acceleration.y) - diffaccy) == 0)) {
    gyrodriftx = (1 - 0.05) * gyrodriftx +
g.gyro.x * 0.05 * 57;
    gyrodrifty = (1 - 0.05) * gyrodrifty +
g.gyro.y * 0.05 * 57;

  }
```

```
//Wenn eine schnelle Bewegung wahrgenommen wird,
wird auf die Werte des Gyros zurückgegriffen
(Genauer)

  if (((abs(g.gyro.x) - diffgyrox) >=
gyroswitch) || ((abs(g.gyro.y) - diffgyroy) >=
gyroswitch)) {
    gyroswitchflag = true;

    xvals[1] += map(((g.gyro.x * 57 -
gyrodriftx) * ((micros() - cycletime) /
1e6)),-180,180,-10,10);
    yvals[1] += map(((g.gyro.y * 57 -
gyrodrifty) * ((micros() - cycletime) /
1e6)),-180,180,-10,10);

  } else {
    gyroswitchflag = false;   //Zurücksetzten
auf Accellerometerbetrieb
  }
  cycletime = micros();



  setservos(xvals[1], yvals[1]);   //setzten der
Servos

  diffaccx = abs(a.acceleration.x);
  diffaccy = abs(a.acceleration.y);
  diffgyrox = abs(g.gyro.x);
```

```
  diffgyroy = abs(g.gyro.y);
}


void setservos(float accx, float accy) {
  pos = map((accx) * 1000, -8000, 8000, 1100,
1900);
  myservo. writeMicroseconds(pos + xoffset);
  pos2 = map((accy) * 1000, -8000, 8000, 1100,
1900);
  myservo2.writeMicroseconds(pos2 + yoffset);
}

//Filterfunktionen für die Servoausgänge

float accx_filtered(float accx) {
  xvals[0] = (1 - filterfaktor) * xvals[0] +
accx * filterfaktor;
  return xvals[0];
}

float accy_filtered(float accy) {
  yvals[0] = (1 - filterfaktor) * yvals[0] +
accy * filterfaktor;
  return yvals[0];
}
```