



# DRON

Súťaž RBA

Kategória: vlastný model

28.5.2019

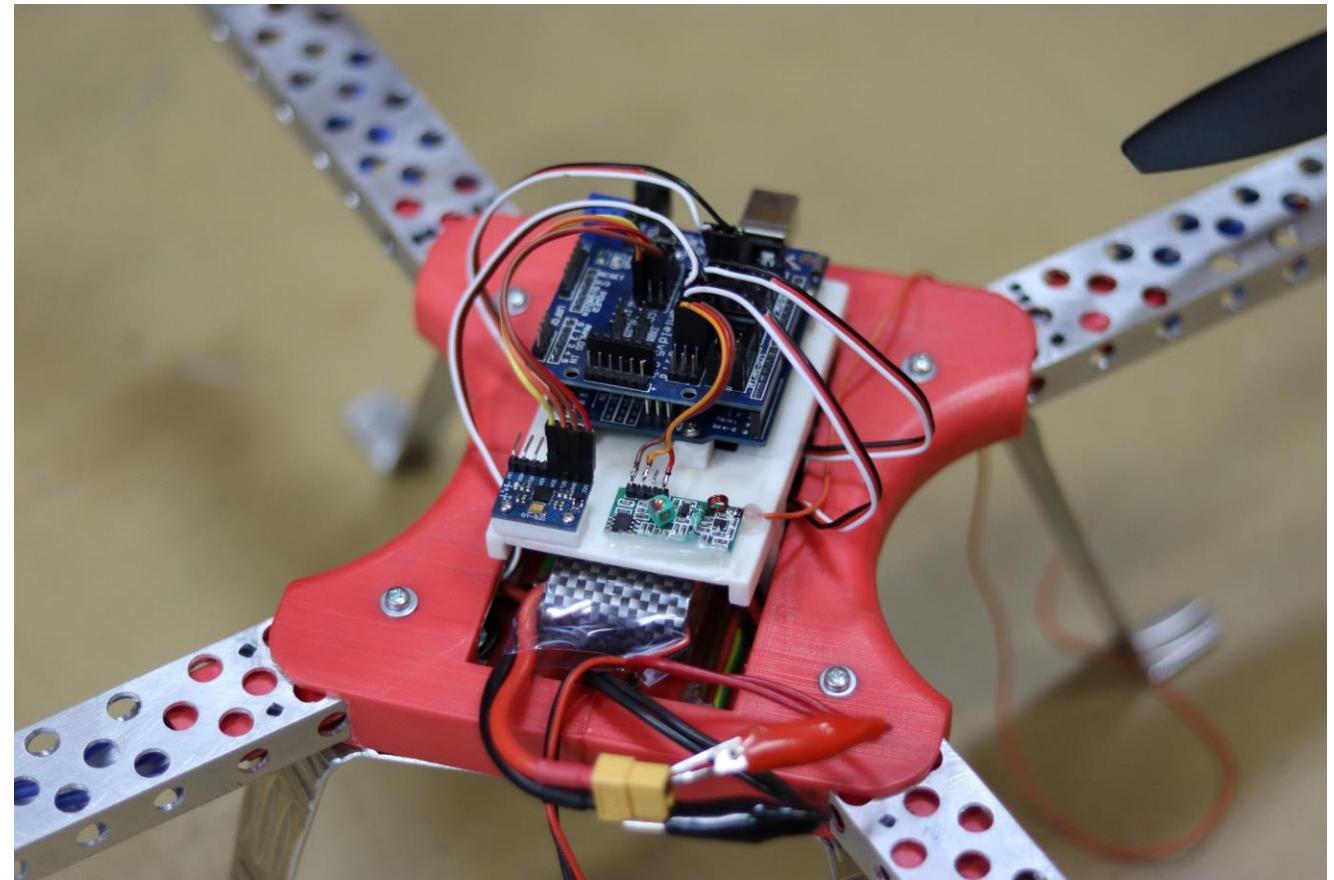
## Srobarka-dron

- Členovia tímu:
  - Stanislav Guskov
  - Jozef Sabo
  - Bianka Šimková



# Čo dokáže?

- komunikácia
- stabilizácia
- dizajn
- výdrž



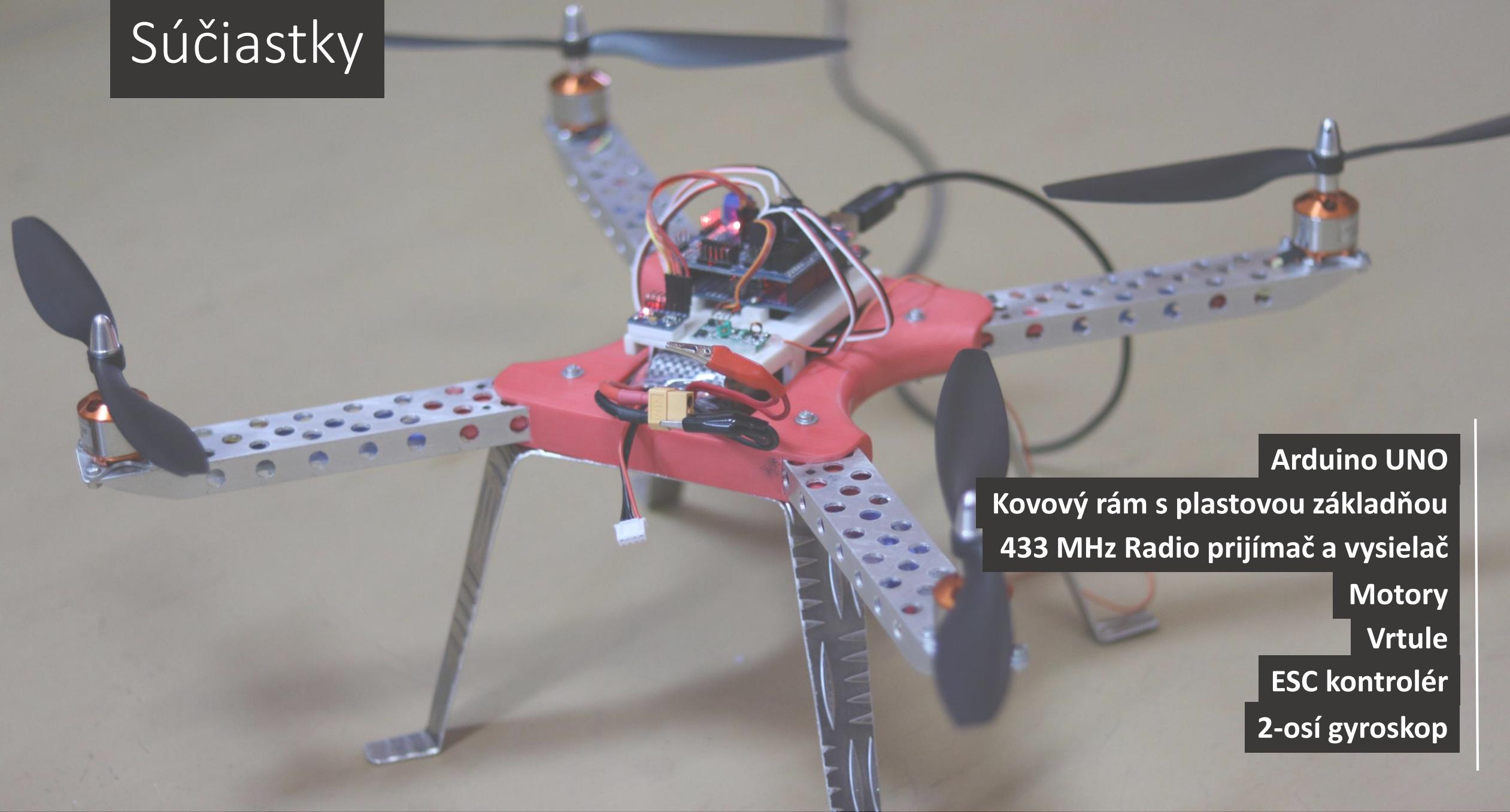
# Na čo ho vieme využiť?

---

- Snímanie zemského povrchu z vtáčej perspektívy
- Transport predmetov
- Zameriavanie pozemkov
- Kontrola rozľahlých plôch
- Drone racing
- Výučba algoritmizácie a proramovania



# Súčiastky



Arduino UNO

Kovový rám s plastovou základňou

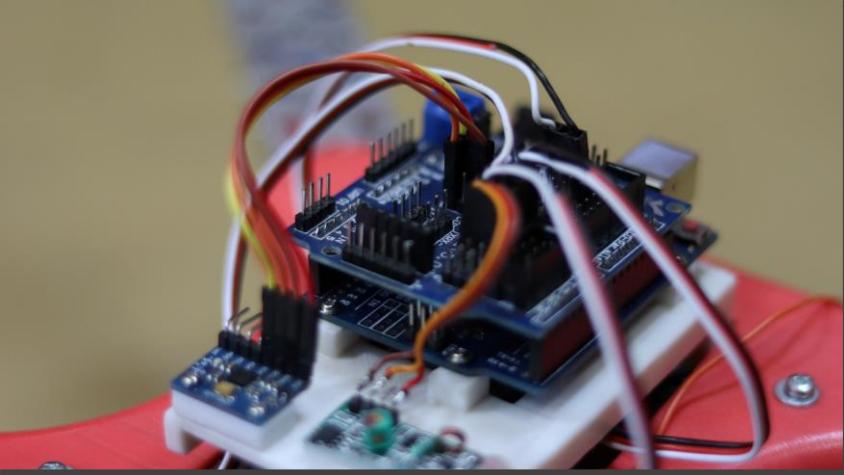
433 MHz Radio prijímač a vysielač

Motory

Vrtule

ESC kontrolér

2-osí gyroskop



Fotografie

# Čo bolo najnáročnejšie?

- program
- stabilizácia
- prvé vzlety
- ESC kontrolér

# Program

```

// Arduino piny | MPU6050
// 5V | Vcc
// GND | GND
// A4 | SDA
// A5 | SCL

#include <Wire.h> // Knižnica pre MPU6050
#include <Servo.h> // V RH_ASK.cpp definovat: #define RH_ASK_ADUINO_USE_TIMER2
#include <RH_ASK.h>

RH_ASK driver(2000, 2, 0, 0);
/*
#include <NewPing.h> // Knižnica pre ultrazvukový senzor
#define TRIGGER_PIN 9 // Porty pre ultrazvukový senzor
#define ECHO_PIN 8
#define MAX_DISTANCE 600
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
*/
int vzdialenosCm;

String txt;
char pismeno;
Servo motor1; // M1 ----- M2
Servo motor2; // | | Rozmiestnenie
Servo motor3; // | | motorov
Servo motor4; // M3 ----- M4
int motor1_power;
int motor2_power;
int motor3_power;
int motor4_power;
float allmotors_power = 100;
float kor_naklon_x, kor_naklon_y;
int i, riadenie[6], smer, riad_vys[5];
float elapsedTime, time, timePrev;
int gyro_error=0;
float Gyr_rawX, Gyr_rawY, Gyr_rawZ;
float Gyro_angle_x, Gyro_angle_y;
float Gyro_raw_error_x, Gyro_raw_error_y;

int acc_error=0;
float rad_to_deg = 180/3.141592654;
float Acc_rawX, Acc_rawY, Acc_rawZ;
float Acc_angle_x, Acc_angle_y;
float Acc_angle_error_x, Acc_angle_error_y;
float Total_angle_x, Total_angle_y;

void setup() {
    Wire.begin(); // Spustiť komunikáciu
    Wire.beginTransmission(0x68); // Odoslať slave adresu (v tomto pripade 68)
    Wire.write(0x6B); // Reset (umiestnite 0 do registra 6B)
    Wire.write(0x00);
    Wire.endTransmission(true); // Ukončiť prenos
    // Gyro config
    Wire.beginTransmission(0x68);
    Wire.write(0xB); // Zápis do registra (1B hex)
    Wire.write(0x10); // Nastavíť bity registrov 00010000 (1000dps)
    Wire.endTransmission(true);
    // Acc config
    Wire.beginTransmission(0x68);
    Wire.write(0x1C);
    Wire.write(0x10); // Nastavíť bity registrov ako 00010000 (+/- 8g rozsahu)
    Wire.endTransmission(true);

    // Rýchlosť pre sériový port
    Serial.begin(9600);
    if (!driver.init())
        Serial.println("init failed");
}

```

```

if (!driver.init())
    Serial.println("init failed");

time = millis();
// Pôrty motorov
motor1.attach(3);
motor2.attach(5);
motor3.attach(6);
motor4.attach(12);
// Vypnutie motorov
motor1.write(0);
motor2.write(0);
motor3.write(0);
motor4.write(0);

// Výpočet gyroskopickej chyby pred spustením slučky, priemer z 200 hodnôt
if(acc_error==0)
{
    for(int a=0; a<200; a++)
    {
        Wire.beginTransmission(0x68); // Požiadat o registráciu 0x3B - odpovedať na AcX
        Wire.write(0x3B);
        Wire.endTransmission(false);
        Wire.requestFrom(0x68,6,true);

        Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0; // Každá hodnota potrebuje dva registre
        Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0;
        Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0;

        Acc_angle_error_x = Acc_angle_error_x + ((atan(Acc_rawX)/sqrt(pow((Acc_rawX),2) + pow((Acc_rawZ),2)))*rad_to_deg);
        Acc_angle_error_y = Acc_angle_error_y + ((atan(-1*(Acc_rawY))/sqrt(pow((Acc_rawY),2) + pow((Acc_rawZ),2)))*rad_to_deg);

        if(a==199)
        {
            Acc_angle_error_x = Acc_angle_error_x/200;
            Acc_angle_error_y = Acc_angle_error_y/200;
            acc_error=1;
        }
    }
}

// Výpočet gyroskopickej chyby pred spustením slučky, priemer z 200 hodnôt
if(gyro_error==0)
{
    for(int i=0; i<200; i++)
    {
        Wire.beginTransmission(0x68); // Odoslať slave adresu (v tomto pripade 68)
        Wire.write(0x43); // Prvá adresa údajov Gyro
        Wire.endTransmission(false);
        Wire.requestFrom(0x68,4,true); // Žiadame len 4 registre

        Gyr_rawX=Wire.read()<<8|Wire.read();
        Gyr_rawY=Wire.read()<<8|Wire.read();

        Gyro_raw_error_x = Gyro_raw_error_x + (Gyr_rawX/32.8);
        Gyro_raw_error_y = Gyro_raw_error_y + (Gyr_rawY/32.8);

        if(i==199)
        {
            Gyro_raw_error_x = Gyro_raw_error_x/200;
            Gyro_raw_error_y = Gyro_raw_error_y/200;
            gyro_error=1;
        }
    }
}

pinMode(13, OUTPUT);
pinMode(11, OUTPUT);

}

void loop() {
    smer = 0;
    if (i>5)
        smer = 1;
    if (i>10)
        smer = 2;
    if (i>15)
        smer = 3;
    if (i>20)
        smer = 4;
    if (i>25)
        smer = 5;
    if (i>30)
        smer = 6;
    if (i>35)
        smer = 7;
    if (i>40)
        smer = 8;
    if (i>45)
        smer = 9;
    if (i>50)
        smer = 10;
    if (i>55)
        smer = 11;
    if (i>60)
        smer = 12;
    if (i>65)
        smer = 13;
    if (i>70)
        smer = 14;
    if (i>75)
        smer = 15;
    if (i>80)
        smer = 16;
    if (i>85)
        smer = 17;
    if (i>90)
        smer = 18;
    if (i>95)
        smer = 19;
    if (i>100)
        smer = 20;
    if (i>105)
        smer = 21;
    if (i>110)
        smer = 22;
    if (i>115)
        smer = 23;
    if (i>120)
        smer = 24;
    if (i>125)
        smer = 25;
    if (i>130)
        smer = 26;
    if (i>135)
        smer = 27;
    if (i>140)
        smer = 28;
    if (i>145)
        smer = 29;
    if (i>150)
        smer = 30;
    if (i>155)
        smer = 31;
    if (i>160)
        smer = 32;
    if (i>165)
        smer = 33;
    if (i>170)
        smer = 34;
    if (i>175)
        smer = 35;
    if (i>180)
        smer = 36;
    if (i>185)
        smer = 37;
    if (i>190)
        smer = 38;
    if (i>195)
        smer = 39;
    if (i>200)
        smer = 40;
}
}

```

```

pinMode(13, OUTPUT);
pinMode(11, OUTPUT);

}

void loop() {
    smer = 0;
    for (i=0;i<=6;i++) riadenie[i]=0;
    //*****
    uint8_t buf[8]; // velkosť prijateho textu (okrem znaku \0, ktoré su ukoncovacie a debugovacie)
    uint8_t bufLen = sizeof(buf);txt = "";
    if (driver.recv(buf, &bufLen))txt = (char*)buf; // Non-blocking
    if (int(txt[0])>0){
        riad_vys[1]=int(txt[1]); //Vysielac PX
        riad_vys[2]=map(int(txt[2]),1,127,1,2000); //Vysielac PY
        riad_vys[3]=map(int(txt[3]),1,127,-127,150); //Vysielac LX
        riad_vys[4]=map(int(txt[4]),1,127,-127,150); //Vysielac LY
        riad_vys[0]=int(txt[0]); //Vysielac PT
    }
    //Riadenie[i], pre i= 0-Hore/Dole, 2-Vlavo/Vpravo, 4-Dopredu/Dozadu
    if (riad_vys[3]<-10||riad_vys[3]>10){
        riadenie[2]=riad_vys[3]; //Vlavo/Vpravo
        smer = 1;
    }
    if (riad_vys[4]<-10||riad_vys[4]>10){
        riadenie[4]=riad_vys[4]; //Dozadu/Dopredu
        smer = 1;
    }
    if (riad_vys[2]>1000)riadenie[0]=10; //Hore
    if (riad_vys[2]>0 && riad_vys[2]<800)riadenie[0]=-10; //Dole
    /*
    Serial.print("PX: ");Serial.print(riad_vys[1]);Serial.print(" ");
    Serial.print("PY: ");Serial.print(riad_vys[2]);Serial.print(" ");
    Serial.print("LX: ");Serial.print(riad_vys[3]);Serial.print(" ");
    Serial.print("LY: ");Serial.print(riad_vys[4]);Serial.print(" ");
    Serial.print("PT: ");Serial.print(riad_vys[0]);Serial.println(" ");
    */
    if(smer == 1){
        digitalWrite(13, HIGH);
    }else{
        digitalWrite(13, LOW);
    }

    timePrev = time; // Uloženie predchádzajúceho času
    time = millis(); // Čítanie času
    elapsedTime = (time - timePrev) / 1000; //Delit 1000, prevod na sekundy
    //////////////////////////////////////////////////////////////////Gyro read////////////////////////////////////////////////////////////////

    Wire.beginTransmission(0x68); //Odoslať slave adresu (68)
    Wire.write(0x43); //Prvá adresa údajov Gyro
    Wire.endTransmission(false);
    Wire.requestFrom(0x68,4,true); //Žiadať len 4 registre

    Gyr_rawX=Wire.read()<<8|Wire.read();
    Gyr_rawY=Wire.read()<<8|Wire.read();
    // Úprava gyro v stupňoch/sek
    Gyr_rawX = (Gyr_rawX/32.8) - Gyro_raw_error_x;
    Gyr_rawY = (Gyr_rawY/32.8) - Gyro_raw_error_y;

    //Vypočet uhlov
    Gyro_angle_x = Gyr_rawX*elapsedTime;
    Gyro_angle_y = Gyr_rawY*elapsedTime;
    //////////////////////////////////////////////////////////////////Acc read////////////////////////////////////////////////////////////////
    Wire.beginTransmission(0x68); //Odoslať slave adresu (v tomto prípade 68)
    Wire.write(0x3B); //Požiaľať o registráciu 0x3B - odpovedať na AcX
    Wire.endTransmission(false);
    Wire.requestFrom(0x68,6,true); //Požiaľať dalších 6 registrov z 3B

    Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0; //Každá hodnota potrebuje dva registre

```

```

    Acc_rawY=(Wire.read()<<8|Wire.read()); //Fodržať prenos a pokračovať
    Acc_rawZ=(Wire.read()<<8|Wire.read()); //Požiaľať dalších 6 registrov z 3B

    Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0; //Každá hodnota potrebuje dva registre
    Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0;
    Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0;
    //Aby sme získali uhly Acc, použijeme euklerov vzorec s hodnotami zrýchlenia

    Acc_angle_x = (atan((Acc_rawY)/sqrt(pow((Acc_rawX),2) + pow((Acc_rawZ),2)))*rad_to_deg) - Acc_angle_error_x;
    Acc_angle_y = (atan(-1*(Acc_rawX)/sqrt(pow((Acc_rawY),2) + pow((Acc_rawZ),2)))*rad_to_deg) - Acc_angle_error_y;
    ////////////////////////////////////////////////////////////////// Výpočet uhlov x a y //////////////////////////////////////////////////////////////////
    Total_angle_x = 0.98 *(Total_angle_x + Gyro_angle_x) + 0.02*Acc_angle_x;
    Total_angle_y = 0.98 *(Total_angle_y + Gyro_angle_y) + 0.02*Acc_angle_y;
    //Kontrola vzdialenosť
    //vzdialenosťCm = sonar.ping_cm();

    //Regulovanie otáčok motorov M1, M2, M3, M4
    /*
    Serial.print("x: ");
    Serial.print(Total_angle_x);
    Serial.print(" | ");
    Serial.print("y: ");
    Serial.print(Total_angle_y);
    Serial.println(" ");
    */
    kor_naklon_x = Total_angle_x;
    kor_naklon_y = Total_angle_y;
    allmotors_power = allmotors_power + riadenie[0]; //Motory Hore/Dole
    //Obmedzenie výšky
    //if (vzdialenosťCm > 100)(allmotors_power = allmotors_power - 10);

    /* RIADENIE 1. verzia, samostatne stabilizacia i riedenie
    if(smer == 0)
    {
        //Len stabilizácia
        motor1_power = allmotors_power - kor_naklon_x + kor_naklon_y;
        motor2_power = allmotors_power + kor_naklon_x + kor_naklon_y;
        motor3_power = allmotors_power - kor_naklon_x - kor_naklon_y;
        motor4_power = allmotors_power + kor_naklon_x - kor_naklon_y;
    }
    else
    {
        //Zmena smeru bez stabilizácie
        motor1_power = allmotors_power + riadenie[2] - (riadenie[4]);
        motor2_power = allmotors_power - (riadenie[2]) - (riadenie[4]);
        motor3_power = allmotors_power + riadenie[2] + riadenie[4];
        motor4_power = allmotors_power - (riadenie[2]) + riadenie[4];
    }
    */
    // RIADENIE 2. verzia, stabilizacia + riadenie
    motor1_power = allmotors_power - kor_naklon_x + kor_naklon_y
    + riadenie[2] - (riadenie[4]);
    motor2_power = allmotors_power + kor_naklon_x + kor_naklon_y
    - (riadenie[2]) - (riadenie[4]);
    motor3_power = allmotors_power - kor_naklon_x - kor_naklon_y
    + riadenie[2] + riadenie[4];
    motor4_power = allmotors_power + kor_naklon_x - kor_naklon_y
    - (riadenie[2]) + riadenie[4];

    if (allmotors_power <= 0)
    {
        allmotors_power = 0;
        motor1_power = 0; motor2_power = 0; motor3_power = 0; motor4_power = 0;
    }
    if (allmotors_power >2000)
    {
        allmotors_power = 2000;
    }
    //delay(1000);
    */

```

```
        }
        if (allmotors_power >2000)
        {
            allmotors_power = 2000;
        }
        //delay(1000);
        //Kalibrovanie motorov
        motor1_power = motor1_power+0;
        motor2_power = motor2_power+0;
        motor3_power = motor3_power+0;
        motor4_power = motor4_power+0;

        Serial.print("M1: ");
        Serial.print(motor1_power);
        Serial.print(" | ");
        Serial.print("M2: ");
        Serial.print(motor2_power);
        Serial.print(" | ");
        Serial.print("M3: ");
        Serial.print(motor3_power);
        Serial.print(" | ");
        Serial.print("M4: ");
        Serial.print(motor4_power);
        Serial.print(" | ");
        Serial.print("Vyska: ");
        Serial.print(vzdialenosCM);
        Serial.println(" ");

        Ovl_motorov ();
    }
    //Ovládanie motorov dĺžkou impulzu
    void Ovl_motorov (){
        // M1 ----- M2
        // |           | Rozmiestnenie
        // |           | motorov
        // M3 ----- M4
        motor1.writeMicroseconds(motor1_power);
        motor2.writeMicroseconds(motor2_power);
        motor3.writeMicroseconds(motor3_power);
        motor4.writeMicroseconds(motor4_power);
    }
}
```



Ďakujeme za pozornosť