

Профиль «АВТОНОМНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ»

[illegible]

Индивидуальная часть

Персональный лист участника с номером 209:



Олимпиада НТИ

ФИО Белавенцев Валерий Евгеньевич

Город Москва

Школа № 2007

Физика

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет ФИЗИКА

Номер участника 209

1	2	3	4	Σ
2	18	8	0	28

~ 2

Дано:
 $L = 11,78 \text{ м}$
 $R = 5 \text{ м}$
 $x - ?$



$$\ell = 2\pi R = 2 \cdot 3,14 \cdot 5 = 31,4 - \text{длина}$$

Всё мертвой петли.

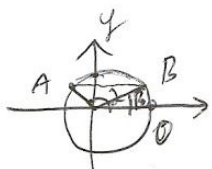
Найдём, какую часть петли прошло тело.

$$k = \frac{L}{\ell} = \frac{11,78}{31,4} \approx 0,375$$

Вся петля представляет собой угол 2π , тогда участок дуги, по которому прошло тело до падения, составляет $L = k \cdot 2\pi = 0,375 \cdot 2\pi =$

$$\approx 0,75\pi = 135^\circ \checkmark$$

открытость.



Пусть наше тело в процессе движения прошло до точки A, после падения оно приземлится

в точке B. Из рисунка видно, что α угол L

и β равные синусы. И т.к. $\sin L = \sin(\pi - L) \Rightarrow$

$$\Rightarrow \beta = \pi - L = 45^\circ \text{ Это есть после приземления}$$

1

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет ФИЗИКА

Номер участника 209

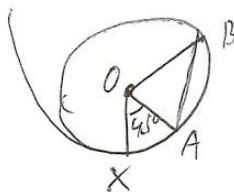
~ 2 (прод.)

мело окажется в той точке, что угол, открывающийся на дуге OA (A - точка падения) равен 45° ✓

$$\beta = 45^\circ$$

4 БАЛЛА

Ответ:



$$\angle XO A = 45^\circ$$

$$\angle XO B = 135^\circ$$

8 БАЛЛОВ

A - точка приземления.

~ 7.

Дано:

$$l = 20 \text{ км}$$

$$v = 277 \frac{\text{м}}{\text{с}}$$

x - ?

$$a_y = \frac{v^2}{R}$$

$$a_y = \frac{F_{\text{тяг}}}{m} = \frac{M \cdot G}{R^2} \quad \text{закон всемирного тяготения}$$

$$a_y = \frac{5,97 \cdot 10^{24} \cdot 6,67408 \cdot 10^{-11}}{(42157)^2 \cdot 10^3} \approx$$

$$= 0,22 \frac{\text{м}}{\text{с}^2}$$

(ан. англ. сист)

(2)

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

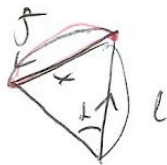
Предмет Физика

Номер участника 209

~ 1 (проф)

$$a_y = \frac{v^2}{R} = \omega^2 R$$

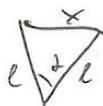
$$\omega = \sqrt{\frac{a_y}{R}} = \sqrt{\frac{0,22}{42157000}} = 0,00007 \text{ (рад/с)}$$



$$t = \frac{20}{177} \approx 11,3 \text{ (с)} \text{ - ракета пролетит } l > 20 \text{ км.}$$

$$\alpha = \omega t = 0,00007 \cdot 11,3 = 0,000791 \text{ (рад)}$$

и
сформулируем



x найдем из т. косинусов.

2 БАЛЛА

$$x = l^2 + l^2 - 2l^2 \cdot \cos \alpha = l^2 (1 - \cos \alpha) =$$

$$= 4 \cdot 10^8 (1 - 0,999987776) \approx 125138 \text{ (м)} \quad \text{Ответ: } \boxed{3} \quad (x \approx 12,5 \text{ км})$$

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 209

~ 3.

Составим систему ур-ий:

$$\frac{\mathcal{E}}{R+r} = I_1 \quad (1)$$

$$\frac{\mathcal{E}}{R+r+x} = 0,98 I_1 \quad (2)$$

$$\frac{\mathcal{E}}{\frac{R}{2}+r} = I_2 \quad (3)$$

$$\frac{\mathcal{E}}{\frac{R(R+x)}{2R+x}+r} = 114,9 I_2 \quad (4)$$

$$\frac{(2)}{(4)} = 0,98 = \frac{\mathcal{E} \cdot (R+r)}{(R+x+r) \cdot \mathcal{E}} = \frac{R+r}{R+x+r}$$

решаем эту систему ур-ий (если требуется - см. черновик, т.к. нет времени переписать)

получаем квадратное уравнение:

$$0,189x^2 - 81,87x - 88 = 0$$

$$\sqrt{D} = 82$$

$$x_{1,2} = \frac{81,87 \pm 82}{2 \cdot 0,189}$$

Ответ: 433 Ом.

Р. С. Будем считать, что напряжение на батарейке не изменяется - $R(R+x)$ есть (уменьшается),



Если же оно изменяется (напр.

из-за разрядки аккумуля-

тора, то паразитное сопротивление будет принимать меньшее значение, чем полученное в ответе.

x - это сопротивл $\Rightarrow$
 $\Rightarrow x > 0$

$$x = \frac{81,87 + 82}{2 \cdot 0,189} \approx 433 \text{ (Ом)}$$

[4]

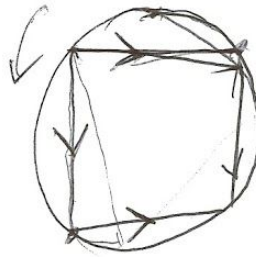
Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 209

№4



Будем считать, что скорость электромагнитной волны равна скорости света в вакууме. ?

√5

$$\left\{ \begin{array}{l} \frac{E}{\frac{R}{2} + r} = I_2 \\ \frac{E}{\frac{R(R+x)}{2R+x} + r} = 1,189 \cdot I_2 \end{array} \right.$$

$$r = \frac{x - 45,5}{4,55}$$

$$x_{1,2} = \frac{81,87 \pm 82}{2 \cdot 0,189}$$

$$1,189 = \frac{\frac{E}{\frac{R(R+x)}{2R+x} + r}}{\frac{\frac{R}{2} + r}{E}}$$

$$1,189 \cdot \frac{\frac{R}{2} + r}{\frac{R(R+x)}{2R+x} + r} = 1$$

$$100 + 5x - 118,9 + 11,89x = 45,5 + 22,75x - 54,9 + 54,9x = 328x + 918,9x^2 - 1772 - 85x$$

$$1,189 = \frac{5 + r}{\frac{100 + 10x}{20 + x} + r}$$

$$0,189x^2 - 81,87x - 16 = 0$$

$$119x^2 - 81870x - 80000 = 0$$

$$D = 566 \quad \sqrt{D} = 82$$

$$1,189 = 5 + \frac{x - 45,5}{4,55}$$

$$\frac{100 + 10x}{20 + x} + \frac{x - 45,5}{4,55} = 5 + \frac{x - 45,5}{4,55}$$

$$1,189 \cdot \frac{100 + 10x}{20 + x} + 1,189 \cdot \frac{x - 45,5}{4,55} = 5 + \frac{x - 45,5}{4,55}$$

$$\frac{118,9 + 11,89x}{20 + x} + 1,189 \left| \frac{x - 45,5}{4,55} \right| = 5 + \dots$$

$$5 = \frac{118,9 + 11,89x}{20 + x} + 0,189 \cdot \frac{x - 45,5}{4,55}$$

$$5 = \frac{100 + 5x - 118,9 + 11,89x}{20 + x} + \frac{0,189x - 8,6}{4,55} (20 + x)$$

$$F_{\text{grav}} = \frac{GMm}{R^2} = m \cdot \frac{GM}{R^2} = \frac{6,67408 \cdot 10^{-11} \cdot 5,92 \cdot 10^{24}}{(42157000)^2} =$$

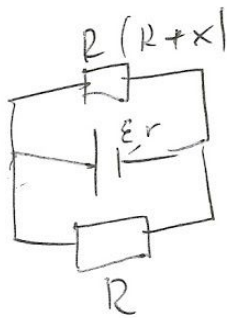
$$= (0,22) \left(\frac{\text{m}}{\text{s}^2} \right)$$

$$a_y = \frac{v^2}{R} \Rightarrow v = \sqrt{a_y \cdot R} = \sqrt{0,22 \cdot 42157000} =$$

$$\approx 3045 \left(\frac{\text{m}}{\text{s}} \right) \approx 3 \frac{\text{km}}{\text{s}} \quad R_{\text{eff}} = \frac{R \cdot (R+x)}{2R+x}$$

$$R_{\text{eff}} = \left(\frac{R}{2} \right)$$

$$R_{\text{eff}} = \frac{R \cdot R}{2R+x}$$



$$\left\{ \begin{array}{l} \frac{E}{R+r} = I_1 \quad (1) \quad r = x = \frac{45,5}{4,55} \\ \frac{E}{R+x+r} = 0,18 I_1 \quad (2) \\ \frac{E}{\frac{R}{2} + x} = I_2 \quad (3) \quad 45,5 + 4,55 r = x \\ \frac{E}{\frac{R(R+x)}{2R+x} + r} = 194,9 I_2 \quad (4) \end{array} \right.$$

$$\frac{(2)}{(1)} \Rightarrow 0,18 = \frac{E}{R+x+r} \cdot \frac{R+r}{E} = \frac{R+r}{R+x+r} = k$$

$$\frac{k}{k-x} = 0,18$$

$$k = 0,18k + 0,18x$$

$$0,82k = 0,18x$$

$$x = 4,55 (10+r)$$

$$x = 4,55 k$$



$$a_y = \frac{v^2}{R}$$

$$\begin{array}{r} 35788 \\ + 6321 \\ \hline 42157 \end{array}$$

$$m a_y = \frac{m v^2}{R}$$

$$a_y = \frac{v^2}{R}$$

$$F_m = \frac{m M}{R^2} G = m \cdot \frac{M G}{R^2}$$

умень.

$$= m \cdot \frac{5,97 \cdot 10^{24} \cdot 6,6 \cdot 10^{-11}}{1777272649} = 0,6 \cdot 22 \cdot 10^4$$

11111

ε₂

$$\frac{10 \cdot (10+x)}{20+x}$$

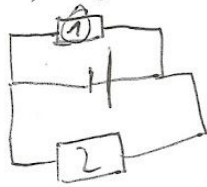
Без нар:

$$E I = \frac{E}{10+x}$$

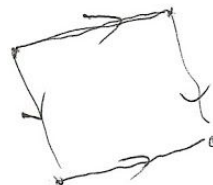
$$0,18 I = \frac{E}{10+x+r}$$

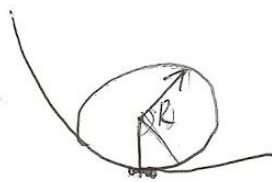
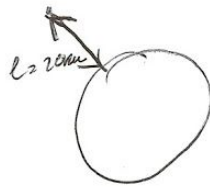
$$0,149 I = \frac{E(20+x)}{100+10x}$$

ε₁ r - нар.



$$\cos \alpha = 0,999968278$$





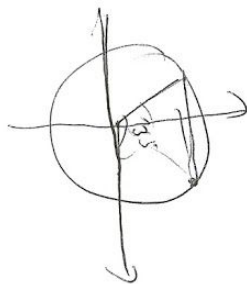
$$l = 2\pi R = 2 \cdot 314.5 \quad (314 \text{ m})$$

314

0,375.

$$2\pi \cdot 0,375$$

$$(0,75\pi) = 135^\circ$$



$$\sin 135^\circ = \sin (180^\circ - 45^\circ)$$

Информатика

Задача 1.1

Код программы на языке Python:

```
1     def rfind(a, elem):
2         pos = len(a) - 1
3         while a[pos] != elem:
4             pos -= 1
5         return pos
6
7     n, k, m = map(int, input().split())
8     m -= 1
9     INF = 10000000000
10    k_copy = k
11    nums = list(map(int, input().split()))
12    var1 = 0
13    var2 = 0
14    while k < m:
15        var = min(nums[:k + 1])
16        var1 += var
17        mpos = nums.index(var)
18        nums[mpos] = INF
19        k += 1
20    var1 += nums[m]
21    k = k_copy
22    while n - k > m:
23        var = min(nums[n - k:])
24        var2 += var
25        mpos = rfind(nums, var)
26        nums[mpos] = INF
27        k += 1
28    var2 += nums[m]
29    print(min(var1, var2))
```

Программа верно решает задачу (6 баллов).

Задача 1.2

Код программы на языке Python:

```
1     def rfind(a, elem):
2         pos = len(a) - 1
3         while a[pos] != elem:
4             pos -= 1
5         return pos
```

```

6
7     n, k, m = map(int, input().split())
8     m -= 1
9     INF = 100000000000
10    k_copy = k
11    nums = list(map(int, input().split()))
12    nums.append(nums[0])
13    nums_copy = nums.copy()
14    var1 = 0
15    var2 = 0
16    while k < m:
17        var = min(nums[:k + 1])
18        var1 += var
19        mpos = nums.index(var)
20        nums[mpos] = INF
21        k += 1
22    var1 += nums[m]
23    k = -k_copy
24    nums = nums_copy.copy()
25    while k > m - n:
26        var = min(nums[k - 1:])
27        var2 += var
28        mpos = rfind(nums, var)
29        nums[mpos] = INF
30        k -= 1
31    var2 += nums[m]
32    print(min(var1, var2))

```

Ошибка при выполнении теста №9 (неверный ответ).

Задача 1.3

Код программы на языке Python:

```

1     def rfind(a, elem):
2         pos = len(a) - 1
3         while a[pos] != elem:
4             pos -= 1
5         return pos
6
7     n, k, m = map(int, input().split())
8     m -= 1
9     INF = 100000000000
10    k_copy = k
11    nums = list(map(int, input().split()))
12    nums.append(nums[0])
13    nums_copy = nums.copy()

```

```
14     var1 = 0
15     var2 = 0
16     while k < m:
17         var = min(nums[:k + 1])
18         var1 += var
19         mpos = nums.index(var)
20         nums[mpos] = INF
21         k += 1
22     var1 += nums[m]
23     k = -k_copy
24     nums = nums_copy.copy()
25     while k > m - n:
26         var = min(nums[k - 1:])
27         var2 += var
28         mpos = rfind(nums, var)
29         nums[mpos] = INF
30         k -= 1
31     var2 += nums[m]
32     print(min(var1, var2))
```

Ошибка при выполнении теста №9 (неверный ответ).

Командная часть

Результаты были получены в рамках выступления команды: Вжух и в продакшн

Личный состав команды:

- Белавенцев Валерий
- Васильев Валерий
- Литвинов Лев
- Лямаев Михаил
- Серянин Александр

Фото команды с собранными устройствами:



Рис.1 - Команда с четырьмя устройствами в сборке

Автонет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Взвух и в продакш	Белавенцев	Валерий	1	10	0	9,5									9,5
	Васильев	Валерий	2.	2	2										2
	Литвинов	Лев	3.	4	4										4
	Лямаев	Михаил	4.	4											0
	Серянин	Александр	5.	2	2										2
			6.	8											0
			7.	20											0
РЕЗУЛЬТАТ															17,5

Аэронет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Взвух и в продакш	Белавенцев	Валерий	1	10	8										8
	Васильев	Валерий	2.	8	8										8
	Литвинов	Лев	3.	12	12										12
	Лямаев	Михаил	4.	15											0
	Серянин	Александр	5.	10											0
			6.	5											0
РЕЗУЛЬТАТ															28

Спейснет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Взвух и в продакш	Белавенцев	Валерий	1	10	10										10
	Васильев	Валерий	2.	5	5										5
	Литвинов	Лев	3.	15	15										15
	Лямаев	Михаил													0
	Серянин	Александр													0
															0
РЕЗУЛЬТАТ															30

Комплексный проезд

Не приступали.

Решения командного этапа:

Маринет

Код программы на языке C++, обеспечивающий решение задач подтрека:

```
.LodkaProg.vsarduino.h
```

```
#ifndef _VSARDUINO_H_
```

```
#define _VSARDUINO_H_
```

```
#define __AVR_ATmega32u4__
```

```
#define __AVR_ATmega32U4__
```

```
#define F_CPU 16000000L

#define ARDUINO 10608

#define ARDUINO_AVR_MICRO

#define ARDUINO_ARCH_AVR

#define USB_VID 0x2341

#define USB_PID 0x8037

#define __cplusplus 201103L

#define __AVR__

#define __inline__

#define __asm__(x)

#define __extension__

#define __inline__

#define __volatile__

#define GCC_VERSION 40801

#define volatile(va_arg)

#define _CONST

#define __builtin_va_start

#define __builtin_va_end

#define __attribute__(x)

#define NOINLINE __attribute__((noinline))

#define prog_void

#define PGM_VOID_P int

#ifndef __builtin_constant_p

    #define __builtin_constant_p __attribute__((__const__))

#endif

#ifndef __builtin_strlen

    #define __builtin_strlen __attribute__((__const__))
```

```
#endif

#define NEW_H

typedef void *__builtin_va_list;

extern "C" void __cxa_pure_virtual() {}


#include <arduino.h>

#include <pins_arduino.h>

#undef F

#define F(string_literal) ((const PROGMEM char
*)(string_literal))

#undef PSTR

#define PSTR(string_literal) ((const PROGMEM char
*)(string_literal))

#include "LodkaProg.ino"

#endif
```

LodkaProg.ino

```
#include <NewPing.h>

#include <Servo.h>


#define MAX_DISTANCE 200

#define OFFSET_IN_STEP 500


#define START 5

#define LED_PIN 13

#define L_wheel 9
```

```

#define R_wheel 10

#define GEO_CHANEL "AT+C085"

#define SAT_CHANEL "AT+C025"

#define volt_pin A0

#define HC12_SetupPin 4

#define IR_PIN A4

#define P 20

//♦♦♦♦♦♦♦♦ ♦♦♦♦♦♦♦♦♦♦

#define FFtrig 6

#define FFecho 6

NewPing sonarFF(FFtrig, FFecho, MAX_DISTANCE);

//♦♦♦♦♦♦♦♦ ♦♦♦♦♦♦ ♦ ♦♦♦♦♦♦ ♦♦♦♦♦♦ ♦♦♦♦♦♦♦♦♦♦

#define FRtrig 12

#define FRecho 12

#define RRtrig 8

#define RRecho 8

NewPing sonarFR(FRtrig, FRecho, MAX_DISTANCE);

NewPing sonarRR(RRtrig, RRecho, MAX_DISTANCE);

//♦♦♦♦♦♦♦♦ ♦♦♦♦♦♦ ♦ ♦♦♦♦♦♦ ♦♦♦♦♦♦ ♦♦♦♦♦♦♦♦♦♦

#define FLtrig 11

#define FLecho 11

#define RLtrig 7

#define RLecho 7

NewPing sonarFL(FLtrig, FLecho, MAX_DISTANCE);

NewPing sonarRL(RLtrig, RLecho, MAX_DISTANCE);

```

```
float distFF;

float distFL;

float distFR;

float distRL;

float distRR;

float LightIR;

int Voltage;

float vLeft;

float vRight;

int offset = 0;


Servo servoLeft, servoRight;

struct GEO {

    int x;

    int y;

    int grad;

};

char strbuf[40];

GEO geo_data;

GEO points[10];

float TwoPointAngle(GEO point1, GEO point2);

// the setup function runs once when you press reset or power
the board

void setup() {

    setPoints();

    servo_attach();

    Serial.begin(9600);
```

```

Serial1.begin(9600);

Serial1.flush();

pinMode(13, OUTPUT);

pinMode(START, INPUT); // ??????????
pinMode(HC12_SetupPin, OUTPUT);

delay(500);

set_transmitter_chanel(GEO_CHANEL);

Voltage = analogRead(volt_pin);

Serial.print("Voltage:");

Serial.println(Voltage*10.00 / 1023.00);


while (!digitalRead(START)) {

  if (analogRead(volt_pin) < 720) {

    digitalWrite(LED_PIN, HIGH);

    delay(250);

    digitalWrite(LED_PIN, LOW);

    delay(250);

  }

  else {

    digitalWrite(LED_PIN, LOW);

  }

}

read_geo_data();


sprintf(strbuf, "X:=%d Y:=%d grad:=%d
offset:=%d", geo_data.x, geo_data.y, geo_data.grad, offset);

Serial.println(strbuf);

```



```

    }

    digitalWrite(LED_PIN, HIGH);

    set_transmitter_chanel(GEO_CHANEL);

    //Serial.println((atan(5/4)*180)/3.14);

}

```

// the loop function runs over and over again until power down or reset

```

void loop() {

    read_geo_data();

    setServo(20, 20);

    while (!(geo_data.y + P >= points[0].y && points[0].y >=
geo_data.y - P)) {

        read_geo_data();

    }

    setServo(-70, -70);

    delay(900);

    setServo(0, 0);

    if (geo_data.grad - 180 < 0) {

        Right();

    }

    else {

        Left();

    }

}

```

```

    }

    setServo(0, 0);

    setServo(20, 20);

    while (!(geo_data.x + P >= points[1].x && points[1].x >=
geo_data.x - P)) {
        read_geo_data();
    }

    setServo(-80, -80);

    delay(900);

    setServo(0, 0);

//  SensorScan();

    while (1)
    {

    }

}

void Right()
{

    setServo(-20, 20);

    while (!(geo_data.grad + 30 >= points[1].grad &&
points[1].grad >= geo_data.grad - 30)) {

        read_geo_data();

    }

```

```

        setServo(80, -80);

        delay(500);
    }

    void Left()
    {
        setServo(20, -20);

        while (!(geo_data.grad + 30 >= points[1].grad &&
points[1].grad >= geo_data.grad - 30)) {

            read_geo_data();

        }

        setServo(-80, 80);

        delay(500);
    }

    void SensorScan()
    {

        delay(2);

        distFF = sonarFF.ping_cm();

        Serial.print("FF:");

        Serial.println(distFF);

        delay(2);

        distFR = sonarFR.ping_cm();

        Serial.print("FR:");

        Serial.println(distFR);

        delay(2);
    }

```

```
distFL = sonarFL.ping_cm();

Serial.print("FL:");

Serial.println(distFL);


delay(2);

distRL = sonarRL.ping_cm();

Serial.print("RL:");

Serial.println(distRL);

delay(2);

distRR = sonarRR.ping_cm();

Serial.print("RR:");

Serial.println(distRR);


LightIR = analogRead(IR_PIN);

Serial.print("IR:");

Serial.println(LightIR);


Voltage = analogRead(volt_pin);

Serial.print("Voltage:");

Serial.print(Voltage);

Serial.print("  ( )");

Serial.println(Voltage*10.00 / 1023.00);


Serial.println("---");
```

```

}

void setServo(float vLeft, float vRight)
{
    if (vLeft > 80) { vLeft = 80; }
    if (vRight > 80) { vRight = 80; }
    servoLeft.write(90 - vLeft);
    servoRight.write(90 + vRight);

}

void servo_attach() {

    servoLeft.attach(L_wheel);
    servoRight.attach(R_wheel);
    servo_stop();

}

void servo_stop() {

    servoLeft.write(90);
    servoRight.write(90);

}

void read_geo_data() {
    String str;
    //str = "rer";
    while (1)

```

```

{
  if (Serial1.available() > 0) {
    str = Serial1.readStringUntil('\n');
    //str = "00;1019;0609;192";
    if (str.substring(0, 1)=="22") {
      offset = str.substring(3, 7).toInt();
    }
    else {
      geo_data.x = str.substring(3, 7).toInt();
      geo_data.y = str.substring(8,
12).toInt()+(offset*OFFSET_IN_STEP);
      geo_data.grad = str.substring(13, 16).toInt();
    }
    Serial.println(str);
  }
}

```

```

void set_transmitter_chanel(char* set_chanel) {

  digitalWrite(HC12_SetupPin, LOW);
  delay(50);
  Send_command(set_chanel);
  digitalWrite(HC12_SetupPin, HIGH);

}

```

```

void Send_command(char* ATcommand) {
    String str;
    Serial1.println(ATcommand);
    Serial.println(ATcommand);
    delay(200);
    if (Serial1.available() > 0)
    {
        str = Serial1.readStringUntil('\n');// LF
        Serial.println(str);
    }
    delay(100);
}

float TwoPointAngle(GEO point1,GEO point2) {

    return atanf((point2.y-point1.y)/(point2.x-point1.x));
}

void setPoints() {
    points[0].x = 380;
    points[0].y = 180;
    points[0].grad = 270;

    points[1].x = 1000;
    points[1].y = 100;
    points[1].grad = 0;
}

```

Автонет

Код программы на языке C++, обеспечивающий решение задач

подтрека:

.MASHINA.vsarduino

/*

Editor: <http://www.visualmicro.com>

visual micro and the arduino ide ignore this code during compilation. this code is automatically maintained by visualmicro, manual changes to this file will be overwritten

the contents of the Visual Micro sketch sub folder can be deleted prior to publishing a project

all non-arduino files created by visual micro and all visual studio project or solution files can be freely deleted and are not required to compile a sketch (do not delete your own code!).

note: debugger breakpoints are stored in '.sln' or '.asln' files, knowledge of last uploaded breakpoints is stored in the upload.vmps.xml file. Both files are required to continue a previous debug session without needing to compile and upload again

Hardware: Arduino Leonardo, Platform=avr, Package=arduino

*/

#ifndef _VSARDUINO_H_

#define _VSARDUINO_H_

#define __AVR_ATmega32u4__

#define __AVR_ATmega32U4__

#define F_CPU 16000000L

#define ARDUINO 10608

#define ARDUINO_AVR_LEONARDO

#define ARDUINO_ARCH_AVR

#define USB_VID 0x2341


```
#define USB_PID 0x8036

#define __cplusplus 201103L

#define __AVR__

#define __inline__

#define __asm__(x)

#define __extension__

#define __inline__

#define __volatile__

#define GCC_VERSION 40801

#define volatile(va_arg)

#define _CONST

#define __builtin_va_start

#define __builtin_va_end

#define __attribute__(x)

#define NOINLINE __attribute__((noinline))

#define prog_void

#define PGM_VOID_P int

#ifndef __builtin_constant_p

    #define __builtin_constant_p __attribute__((__const__))

#endif

#ifndef __builtin_strlen

    #define __builtin_strlen __attribute__((__const__))

#endif

#define NEW_H

typedef void *__builtin_va_list;

extern "C" void __cxa_pure_virtual() {}
```

```
#include <arduino.h>

#include <pins_arduino.h>

#undef F

#define F(string_literal) ((const PROGMEM char
*)(string_literal))

#undef PSTR

#define PSTR(string_literal) ((const PROGMEM char
*)(string_literal))

#include "MASHINA.ino"

#endif
```

MASHINA.ino

```
// the setup function runs once when you press reset or power
the board
```

```
#include <IRremoteInt.h>

#include <IRremote.h>

#include <Servo.h>

#define SIGNAL_FORWARD 0xc00

#define SIGNAL_STOP 0xf00

#define SIGNAL_LEFT 0xa00

#define SIGNAL_RIGHT 0x900

#define MAX_DISTANCE 200

#define OFFSET_IN_STEP 500

#define MOTOR_PIN 5
```

```
#define SERVO_PIN 6

#define LINE_PIN A2

#define SONIC_PIN 3

#define IR_PIN A4

#define SAUND_PIN 7

#define LED_PIN A1

#define HC12_SetupPin 2

#define GEO_CHANEL "AT+C056"

#define SAT_CHANEL "AT+C025"

#define START A0

int offset = 0;

int dist = 0;

IRrecv irrecv(IR_PIN); // пин ик

Servo servo, motor;

//NewPing sonar(SONIC_PIN, SONIC_PIN, MAX_DISTANCE); // NewPing
setup of pins and maximum distance.

//Ultrasonic ultrasonic(SONIC_PIN, 5); // (Trig PIN, Echo PIN)

struct GEO {

    int x;

    int y;

    int grad;

};

char strbuf[40];

GEO geo_data;

decode_results results;

uint32_t ms, ms1 = 0;

bool led_stat = false;
```

```

void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    irrecv.enableIRIn();
    servo.attach(SERVO_PIN);
    motor.attach(MOTOR_PIN);
    pinMode(START, INPUT_PULLUP);
    digitalWrite(START, HIGH);
    pinMode(HC12_SetupPin, OUTPUT);
    digitalWrite(HC12_SetupPin, HIGH);
    pinMode(LED_PIN, OUTPUT);
    digitalWrite(LED_PIN, HIGH);
    pinMode(SAUND_PIN, OUTPUT);
    digitalWrite(SAUND_PIN, LOW);
    //digitalWrite(SAUND_PIN, HIGH);
    delay(500);
    set_transmitter_chanel(GEO_CHANEL);
    servo.write(90);
    motor.write(90);
    while (!digitalRead(START)) {
        read_geo_data();
        Serial.println(analogRead(LINE_PIN));
    }

    cheks();
    while (1);
}

```

```

}

// the loop function runs over and over again until power down
or reset

void loop() {

    read_geo_data();

    if (geo_data.grad > 270) {
        servo.write(80);
    }

    if (geo_data.grad < 270) {
        servo.write(100);
    }

    motorF(110);
    delay(200);
    motorF(100);
}

void IR_read() {

    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);
        switch (results.value) {
            case SIGNAL_FORWARD:
                Serial.println("Forward!");
                break;

            case SIGNAL_LEFT:

```

```

        Serial.println("Left!");
        break;

case SIGNAL_RIGHT:
    Serial.println("Right!");
    break;

case SIGNAL_STOP:
    Serial.println("Stop!");
    break;

default:
    Serial.print("Unknown signal: ");
    Serial.print(results.value, HEX);
    Serial.println("!");
}

irrecv.resume(); // Receive the next value
}

}

void motorS() {
    //Serial.print(motor.read());
    if (motor.read() > 90) {
        motor.write(40);
    }
    if (motor.read() < 90)
    {
        //    motor.write(140);
    }
}

```

```

    }

    delay(150);

    motor.write(90);

    delay(150);
}

void motorF(int speed) {

    motor.write(speed);

}

void motorB(int speed) {

    if (motor.read() > 90) {

        motorS();

    }

    motor.write(speed);

}

void read_geo_data() {

    String str;

    //str = "rer";

    while (1)

    {

        if (Serial1.available() > 0) {

            str = Serial1.readStringUntil('\n');

            //str = "00;1019;0609;192";

            if (str.substring(0, 1) == "22") {

                offset = str.substring(3, 7).toInt();

            }

        }

    }

}

```

```

        else {
            geo_data.x = str.substring(3, 7).toInt();
            geo_data.y = str.substring(8, 12).toInt() +
(offset*OFFSET_IN_STEP);
            geo_data.grad = str.substring(13, 16).toInt();
        }
        Serial.println(str);
        return;
    }
}

```

```

void set_transmitter_chanel(char* set_chanel) {

```

```

    digitalWrite(HC12_SetupPin, LOW);
    delay(50);
    Send_command(set_chanel);
    digitalWrite(HC12_SetupPin, HIGH);

```

```

}

```

```

void Send_command(char* ATcommand) {

```

```

    String str;
    Serial1.println(ATcommand);
    Serial.println(ATcommand);
    delay(200);
    if (Serial1.available() > 0)

```



```

    {
        str = Serial1.readStringUntil('\n');// LF
        Serial.println(str);
    }

    delay(100);
}

int Distanse() {
    pinMode(SONIC_PIN, OUTPUT); //Ультразвук

    unsigned int impulseTime = 0;
    digitalWrite(SONIC_PIN, HIGH);

    /* Подаем импульс на вход trig датчика */
    delayMicroseconds(10); // равный 10 микросекундам
    digitalWrite(SONIC_PIN, LOW); // Отключаем

    pinMode(SONIC_PIN, INPUT); //Ультразвук

    impulseTime = pulseIn(SONIC_PIN, HIGH); // Замеряем длину
импульса

    //Serial.print(impulseTime);

    return impulseTime / 58; // Пересчитываем в сантиметры
}

void cheks() {

    motorF(105);

    delay(1000);

    motorF(100);

    //Serial.println("F");
}

```

```

while (analogRead(LINE_PIN) > 500);

//Serial.println("S");

//motorS();

motorF(105);

delay(500);

motorF(100);

while (analogRead(LINE_PIN) > 500)
{
    ms = millis();

    // Событие срабатывающее каждые 500 мс
    if ((ms - ms1) > 200 || ms < ms1) {
        ms1 = ms;

        // Инвертируем светодиод
        digitalWrite(LED_PIN, led_stat);

        led_stat = !led_stat;
    }
}

digitalWrite(LED_PIN, HIGH);

motorS();

digitalWrite(SAUND_PIN, HIGH);

delay(500);

digitalWrite(SAUND_PIN, LOW);

}

```

Аэронет

Код программы на языке C++, обеспечивающий решение задач подтрека:

Copter01.ino

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

```
#include <Ultrasonic.h>
```

```
Servo Roll;
```

```
Servo Pitch;
```

```
Servo Throt;
```

```
Servo Yaw;
```

```
Ultrasonic sForward(11, 12);
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
    pinMode(2, INPUT);
```

```
    Roll.attach(3, 1000, 2000);
```

```
    Pitch.attach(5, 1000, 2000);
```

```
    Throt.attach(6, 900, 2000);
```

```
    Yaw.attach(9, 1000, 2000);
```

```
    delay(6000);
```

```
    while(!digitalRead(2))
```

```
    {
```

```
        Serial.println(sForward.Ranging(CM));
```

```
        delay(100);
```

```
}  
  
Throt.writeMicroseconds(900);  
Roll.writeMicroseconds(1500);  
Pitch.writeMicroseconds(1500);  
Yaw.writeMicroseconds(2000);  
delay(4000);  
Yaw.writeMicroseconds(1900);  
delay(100);  
Yaw.writeMicroseconds(1800);  
delay(100);  
Yaw.writeMicroseconds(1700);  
delay(100);  
Yaw.writeMicroseconds(1600);  
delay(100);  
Yaw.writeMicroseconds(1500);  
delay(100);  
  
}
```

```
void loop()  
{  
  
    Throt.writeMicroseconds(1000);  
    delay(1000);  
    Throt.writeMicroseconds(1100);  
    delay(1000);  
    Throt.writeMicroseconds(1200);  
    delay(1000);  
}
```

```
    Throt.writeMicroseconds(1300);  
    delay(1000);  
    Throt.writeMicroseconds(1400);  
    delay(1000);  
    Throt.writeMicroseconds(1500);  
    delay(2000);  
    //Roll.writeMicroseconds(1500);  
    //Pitch.writeMicroseconds(1500);  
    //Yaw.writeMicroseconds(1500);  
  
}
```

Copter02.ino

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

```
#include <NewPing.h>
```

```
Servo Roll;
```

```
Servo Pitch;
```

```
Servo Throt;
```

```
Servo Yaw;
```

```
NewPing sForward(4, 4, 100);
```

```
NewPing sBack(7, 7, 100);
```

```
NewPing sLeft(8, 8, 100);
```

```
NewPing sRight(10, 10, 100);
```

```
NewPing sDown(11, 11, 100);
```

```
void setup()
```

```
{  
  
  Serial.begin(9600);  
  
  pinMode(2, INPUT);  
  
  Roll.attach(3, 1000, 2000);  
  
  Pitch.attach(5, 1000, 2000);  
  
  Throt.attach(6, 900, 2000);  
  
  Yaw.attach(9, 1000, 2000);  
  
  delay(5000);  
  
  
  while(!digitalRead(2))  
  {  
  
    delay(10);  
  
  }  
  
  
  Throt.writeMicroseconds(900);  
  
  Roll.writeMicroseconds(1500);  
  
  Pitch.writeMicroseconds(1500);  
  
  Yaw.writeMicroseconds(2000);  
  
  delay(4000);  
  
  Yaw.writeMicroseconds(1900);  
  
  delay(100);  
  
  Yaw.writeMicroseconds(1800);  
  
  delay(100);  
  
  Yaw.writeMicroseconds(1700);  
  
  delay(100);  
  
  Yaw.writeMicroseconds(1600);  
  
  delay(100);
```

```
    Yaw.writeMicroseconds(1500);

    delay(100);
}

void loop()
{
    Serial.println(sonar.ping_cm());

    delay(100);

}
```

Copter03.ino

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

```
#include <NewPing.h>
```

```
Servo Roll;
```

```
Servo Pitch;
```

```
Servo Throt;
```

```
Servo Yaw;
```

```
NewPing sForward(4, 4, 100);
```

```
NewPing sBack(7, 7, 100);
```

```
NewPing sLeft(8, 8, 100);
```

```
NewPing sRight(10, 10, 100);
```

```
NewPing sDown(11, 11, 100);
```

```
int rollVal = 1500;
```

```
int pitchVal = 1500;

int throtVal = 1000;

int yawVal = 2000;


void control(int roll, int pitch, int throt, int yaw)
{
    Roll.writeMicroseconds(roll);
    Pitch.writeMicroseconds(pitch);
    Throt.writeMicroseconds(throt);
    Yaw.writeMicroseconds(yaw);
}


void arming()
{
    rollVal = 1500;
    pitchVal = 1500;
    throtVal = 1000;
    yawVal = 2000;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(5000);
    yawVal = 1900;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1800;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1700;
```



```

    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1600;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1500;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
}

void takeOff()
{
    rollVal = 1500;
    pitchVal = 1500;
    throtVal = 1000;
    yawVal = 1500;
    while(throtVal <= 1300)
    {
        throtVal += 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);

    while(throtVal <= 1500)
    {
        throtVal += 50;

```

```

        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);

    while(throtVal <= 1600)
    {
        throtVal += 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);

    while(throtVal <= 1750)
    {
        throtVal += 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);
}

void landing()
{
    rollVal = 1500;
    pitchVal = 1500;
    throtVal = 1500;

```

```

yawVal = 1500;
while(throtVal >= 1300)
{
    throtVal -= 50;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(500);
}
delay(1000);
}

void setup()
{
    Serial.begin(9600);
    pinMode(2, INPUT);

    Roll.attach(3, 1000, 2000);
    Pitch.attach(5, 1000, 2000);
    Throt.attach(6, 1000, 2000);
    Yaw.attach(9, 1000, 2000);
    delay(3000);

    while(!digitalRead(2))
    {
        delay(10);
    }

    arming();

```

```

    takeOff();

    delay(1000);

    landing();

}

void loop()
{
    if (!digitalRead(2))
    {
        throtVal = 1000;

        control(rollVal, pitchVal, throtVal, yawVal);

        delay(10);
    }

    if (sDown.ping_cm() <= 5)
    {
        throtVal = 1000;

        control(rollVal, pitchVal, throtVal, yawVal);

        delay(10);
    }

    Serial.println(sDown.ping_cm());

    delay(100);

}

```

Copter04.ino

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

```
#include <NewPing.h>

Servo Roll;

Servo Pitch;

Servo Throt;

Servo Yaw;


NewPing sForward(4, 4, 100);
NewPing sBack(7, 7, 100);
NewPing sLeft(8, 8, 100);
NewPing sRight(10, 10, 100);
NewPing sDown(11, 11, 100);


int rollVal = 1500;
int pitchVal = 1500;
int throtVal = 1000;
int yawVal = 2000;


#define rollMid 1460
#define pitchMid 1460
#define throtMid 1950
#define yawMid 1500


#define throtMax 1950


#define toWall 50
#define toFlour 70
```

```
void control(int roll, int pitch, int throt, int yaw)
{
    Roll.writeMicroseconds(roll);
    Pitch.writeMicroseconds(pitch);
    Throt.writeMicroseconds(throt);
    Yaw.writeMicroseconds(yaw);
}
```

```
void arming()
{
    Serial.println("Arming");
    rollVal = rollMid;
    pitchVal = pitchMid;
    throtVal = 1000;
    yawVal = 2000;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(5000);
    yawVal = 1900;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1800;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
    yawVal = 1700;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
}
```

```

    yawVal = 1600;

    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);

    yawVal = 1500;

    control(rollVal, pitchVal, throtVal, yawVal);
    delay(100);
}

void takeOff()
{
    Serial.println("Take Off start");

    rollVal = rollMid;
    pitchVal = pitchMid;
    throtVal = 1000;
    yawVal = yawMid;
    while(throtVal <= 1300)
    {
        throtVal += 50;

        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);

    while(throtVal <= 1500)
    {
        throtVal += 50;

        control(rollVal, pitchVal, throtVal, yawVal);

```

```
        delay(100);
    }
    delay(1000);

    while(throtVal <= 1600)
    {
        throtVal += 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1000);

    while(throtVal <= 2000)
    {
        throtVal += 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(700);

    while(throtVal > throtMid)
    {
        throtVal -= 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(100);
    }
    delay(1500);
```



```

    Serial.println("Take Off end");

}

void landing()
{
    Serial.println("Landing");
    rollVal = rollMid;
    pitchVal = pitchMid;
    throtVal = throtMid;
    yawVal = yawMid;
    while(sDown.ping_cm() >= 5)
    {
        while(throtVal > 1250)
        {
            throtVal -= 50;
            control(rollVal, pitchVal, throtVal, yawVal);
            delay(500);
        }
        delay(100);
    }

    throtVal = 1200;
    control(rollVal, pitchVal, throtVal, yawVal);
    delay(500);
    throtVal = 1100;
    control(rollVal, pitchVal, throtVal, yawVal);

```

```
    delay(500);

    throtVal = 1000;

    control(rollVal, pitchVal, throtVal, yawVal);

    delay(500);

    Serial.println("End");

    while (1)

    {

        delay(100);

    }

}

void setup()

{

    Serial.begin(9600);

    pinMode(2, INPUT);

    Roll.attach(3, 1000, 2000);

    Pitch.attach(5, 1000, 2000);

    Throt.attach(6, 1000, 2000);

    Yaw.attach(9, 1000, 2000);

    delay(3000);

    while(!digitalRead(2))

    {

        delay(10);

    }
```

```

    arming();

    takeOff();
}

int t = 0;

# define delta 75

void loop()
{
    if (!digitalRead(2))
    {
        throtVal = 1000;
    }

    if (t >= 30000)
    {
        landing();
    }

    //BMCOTA

    if (sDown.ping_cm() <= toFlour - 10)
    {
        throtVal += delta;

        throtVal = constrain(throtVal, 1200, throtMax);
    }

    if (sDown.ping_cm() >= toFlour + 10 or sDown.ping_cm() == 0)
    {
        throtVal -= delta;
    }
}

```

```

        throtVal = constrain(throtVal, 1200, throtMax);
    }

//ЛЕВО ПРАВО
if (sLeft.ping_cm() <= toWall and sLeft.ping_cm() != 0)
{
    rollVal += delta;
    rollVal = constrain(rollVal, 1000, 2000);
}
if (sRight.ping_cm() <= toWall and sRight.ping_cm() != 0)
{
    rollVal -= delta;
    rollVal = constrain(rollVal, 1000, 2000);
}

//ПЕРЕД ЗАД
if (sForward.ping_cm() <= toWall and sForward.ping_cm() != 0)
{
    pitchVal += delta;
    pitchVal = constrain(pitchVal, 1000, 2000);
}
if (sBack.ping_cm() <= toWall and sBack.ping_cm() != 0)
{
    pitchVal -= delta;
    pitchVal = constrain(pitchVal, 1000, 2000);
}

```

```
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(30);  
    t += 30;  
    Serial.println(t);  
    Serial.println(throtVal);  
  
}
```

Copter05.ino

```
#include <Servo.h>  
  
#include <NewPing.h>  
  
  
Servo Roll;  
  
Servo Pitch;  
  
Servo Throt;  
  
Servo Yaw;  
  
  
NewPing sForward(4, 4, 100);  
NewPing sBack(7, 7, 100);  
NewPing sLeft(8, 8, 100);  
NewPing sRight(10, 10, 100);  
NewPing sDown(11, 11, 100);  
  
  
int rollVal = 1500;  
  
int pitchVal = 1500;  
  
int throtVal = 1000;  
  
int yawVal = 2000;
```

```
#define rollMid 1460

#define pitchMid 1450

#define throtMid 1950

#define yawMid 1500


#define throtMax 2000


#define toWall 50

#define toFlour 120


void control(int roll, int pitch, int throt, int yaw)
{
    Roll.writeMicroseconds(roll);
    Pitch.writeMicroseconds(pitch);
    Throt.writeMicroseconds(throt);
    Yaw.writeMicroseconds(yaw);
}


void arming()
{
    Serial.println("Arming");

    rollVal = rollMid;
    pitchVal = pitchMid;
    throtVal = 1000;
    yawVal = 2000;

    control(rollVal, pitchVal, throtVal, yawVal);
}
```

```
    delay(5000);  
    yawVal = 1900;  
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(100);  
    yawVal = 1800;  
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(100);  
    yawVal = 1700;  
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(100);  
    yawVal = 1600;  
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(100);  
    yawVal = 1500;  
    control(rollVal, pitchVal, throtVal, yawVal);  
    delay(100);  
}
```

```
void takeOff()  
{  
    Serial.println("Take Off start");  
    rollVal = rollMid;  
    pitchVal = pitchMid;  
    throtVal = 1000;  
    yawVal = yawMid;  
    while(throtVal <= 1300)  
    {
```

```
    throtVal += 50;

    control(rollVal, pitchVal, throtVal, yawVal);

    delay(100);
}

delay(1000);

while(throtVal <= 1500)
{
    throtVal += 50;

    control(rollVal, pitchVal, throtVal, yawVal);

    delay(100);
}

delay(1000);

while(throtVal <= 1600)
{
    throtVal += 50;

    control(rollVal, pitchVal, throtVal, yawVal);

    delay(100);
}

delay(1000);

while(throtVal <= 2000)
{
    throtVal += 50;

    control(rollVal, pitchVal, throtVal, yawVal);

    delay(100);
```



```

    }

    delay(700);

    while(throtVal > throtMid)
    {
        throtVal -= 50;

        control(rollVal, pitchVal, throtVal, yawVal);

        delay(100);
    }

    delay(1000);

    Serial.println("Take Off end");

}

void forward()
{
    throtVal = 1950;
}

void landing()
{
    Serial.println("Landing");

    rollVal = rollMid;

    pitchVal = pitchMid;

    throtVal = throtMid;

    yawVal = yawMid;

    while(sDown.ping_cm() >= 5)

```

```

{
    while(throtVal > 1250)
    {
        throtVal -= 50;
        control(rollVal, pitchVal, throtVal, yawVal);
        delay(500);
    }
    delay(100);
}

throtVal = 1200;
control(rollVal, pitchVal, throtVal, yawVal);
delay(500);
throtVal = 1100;
control(rollVal, pitchVal, throtVal, yawVal);
delay(500);
throtVal = 1000;
control(rollVal, pitchVal, throtVal, yawVal);
delay(500);
Serial.println("End");
while (1)
{
    delay(100);
}
}

```

```

void setup()
{
    Serial.begin(9600);
    pinMode(2, INPUT);

    Roll.attach(3, 1000, 2000);
    Pitch.attach(5, 1000, 2000);
    Throt.attach(6, 1000, 2000);
    Yaw.attach(9, 1000, 2000);
    delay(3000);

    while(!digitalRead(2))
    {
        delay(10);
    }

    arming();
    takeOff();
}

int t = 0;
# define delta 75
void loop()
{
    if (!digitalRead(2))
    {
        throtVal = 1000;
    }
}

```

```

}

if (t >= 5000)

{
    landing();
}


//БЫСОТА

if (sDown.ping_cm() <= toFlour - 10)

{
    throtVal += delta;

    throtVal = constrain(throtVal, 1200, throtMax);
}

if (sDown.ping_cm() >= toFlour + 10 or sDown.ping_cm() == 0)

{
    throtVal -= delta;

    throtVal = constrain(throtVal, 1200, throtMax);
}


/*//ПЕБО ПРАВО

if (sLeft.ping_cm() <= toWall and sLeft.ping_cm() != 0)

{
    rollVal += delta;

    rollVal = constrain(rollVal, 1000, 2000);
}

if (sRight.ping_cm() <= toWall and sRight.ping_cm() != 0)

{
    rollVal -= delta;

```

```

        rollVal = constrain(rollVal, 1000, 2000);
    }*/

/* //ПЕРЕД ЗАД
    if (sForward.ping_cm() <= toWall and sForward.ping_cm() != 0)
    {
        pitchVal += delta;
        pitchVal = constrain(pitchVal, 1000, 2000);
    }
    if (sBack.ping_cm() <= toWall and sBack.ping_cm() != 0)
    {
        pitchVal -= delta;
        pitchVal = constrain(pitchVal, 1000, 2000);
    }*/
    pitchVal = 1700;

    control(rollVal, pitchVal, throtVal, yawVal);
    delay(30);
    t += 30;
    Serial.println(t);
    Serial.println(throtVal);
}

```

Спейснет

Код программы на языке C++, обеспечивающий решение задач подтрека:

SputnikProg.ino

```

#define STEP_PIN 10

#define DIR_PIN 16

```

```

#define POWER_ENABLE_PIN 14

#define END_SWITCH1 3

#define END_SWITCH2 2

#define HC12_SetupPin 4

#define LEFT_DIR 0

#define RIGHT_DIR 1

#define GEO_CHANEL "AT+C056"

#define LED1 8

#define LED2 9


#define OFFSET_IN_STEP 100

#define STEP_IN_ROTATION 400

struct GEO {

    int x;

    int y;

    int grad;

};

GEO geo_data;

const int volt_pin = A0;

float Voltage;

int dir, rotation, offset;

long steps, i;

String str;

    char strbuf[15];

void setup() {

    // put your setup code here, to run once:

```

```
Serial.begin(9600);

Serial1.begin(9600);

pinMode(STEP_PIN, OUTPUT);

pinMode(DIR_PIN, OUTPUT);

pinMode(POWER_ENABLE_PIN, OUTPUT);

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

pinMode(HC12_SetupPin, OUTPUT);

digitalWrite(POWER_ENABLE_PIN, HIGH);

digitalWrite(LED1, HIGH);

digitalWrite(LED2, HIGH);

delay(1000);

digitalWrite(LED1, LOW);

digitalWrite(LED2, LOW);

Voltage = analogRead(volt_pin);

if(Voltage < 480 ){ // мониторинг напряжения

    while(1){

        digitalWrite(LED2, HIGH);

        delay(1000);

        digitalWrite(LED2, LOW);

        delay(1000);

    }

}
```

```

    set_transmitter_chanel(GEO_CHANEL);

    move(RIGHT_DIR,10);

    offset = 0;
}

void loop() {

    // put your main code here, to run repeatedly:

    read_geo_data();

    if(geo_data.y < 350 && geo_data.y > 10){

        move(LEFT_DIR,1);

        Serial.println("left");

        offset++;

    }

    if(geo_data.y > 700 && geo_data.y > 10 ){

        move(RIGHT_DIR,1);

        Serial.println("right");

        offset--;

    }

    sprintf(strbuf,"22;%04d",offset);

    Serial1.println(strbuf);

    Serial.println(strbuf);

}

void move(int dir, unsigned int rotation) {

    digitalWrite(

```



```

    POWER_ENABLE_PIN, LOW);

digitalWrite(DIR_PIN, dir);

steps = rotation * STEP_IN_ROTATION;

doSteps(steps);

digitalWrite(POWER_ENABLE_PIN, HIGH);

}

boolean isEndSwitch() {

    if (digitalRead(END_SWITCH1) == 1) {

        Serial1.println("END_SWITCH1 (Start)");

        rollBack(RIGHT_DIR);

        return true;

    }

    if (digitalRead(END_SWITCH2) == 1) {

        Serial1.println("END_SWITCH2 (Finish)");

        rollBack(LEFT_DIR);

        return true;

    }

    return false;

}

```

```
boolean isStopInSerial() {

    if (Serial1.available() > 0) {
        if (Serial1.readStringUntil('\n').toInt() == 0) {
            return true;
        }
    }

    return false;

}
```

```
void rollBack(int dir) {

    digitalWrite(DIR_PIN, dir);
    for (i = 0; i <= 100; i++) {
        doStep();
    }
}
```

```
void doSteps(long steps) {

    for (i = 0; i <= steps; i++) {

        if (isEndSwitch()) {
            return;
        }
    }
}
```

```

    }

    if (isStopInSerial()) {
        return;
    }

    doStep();

}

Serial1.println("STOP");

}

void doStep() {

    digitalWrite(STEP_PIN, HIGH);
    delayMicroseconds(500);
    digitalWrite(STEP_PIN, LOW);
    delayMicroseconds(500);

}

void set_transmitter_chanel(String set_chanel) {

    digitalWrite(HC12_SetupPin, LOW);
    delay(50);
    Send_command(set_chanel);
}

```

```

    digitalWrite(HC12_SetupPin, HIGH);
    delay(100);
}

void Send_command(String ATcommand) {

    Serial1.println (ATcommand);
    Serial.println (ATcommand);
    delay(100);
    if (Serial1.available() > 0)
    {
        str = Serial1.readStringUntil('\n');// LF
        Serial.println (str);
    }
}

GEO read_geo_data() {

while(1){

    if (Serial1.available() > 0) {

        str = Serial1.readStringUntil('\n');
        //str = "00;1019;0609;192";
        Serial.println(str);

        geo_data.x      = str.substring(3, 7).toInt();
        geo_data.y      = str.substring(8, 12).toInt();
        geo_data.grad = str.substring(13, 16).toInt();
    }
}
}

```

```
        return geo_data;
    }
}
}
```