

Работа призера заключительного этапа
командной инженерной олимпиады школьников
Олимпиада Национальной технологической инициативы

Профиль «АВТОНОМНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ»

Цветков Иван Сергеевич

Класс: 10

Город: Москва

Школа: ГОУ СОШ № 444

Регион: Москва

Уникальный номер участника: 18

**Команда на заключительном
этапе:** Фортуна

Результаты заключительного этапа:

№	Индивидуальный этап												Командный этап		ИТОГ	
	Физика				Информатика						За задачи	Коэффициент 40%	баллы	Коэффициент 60%		
18	2	6	16	0	0	0	0	0	0	0	0	24	6,4	61,0	14,64	21,040

Индивидуальная часть

Персональный лист участника с номером 18:



Олимпиада НТИ

ФИО ЦВЕТКОВ ИВАН. СЕРГЕЕВИЧ

Город МОСКВА

Школа № 444

Физика

После апелляции участник получил +2 балла за первое задание.

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 8

1	2	3	4	5
0	6	16	0	22

7.

N1.

$$H = 35786 \text{ км.}$$

$$v_p = 7.77 \text{ км/с}$$

v_z скорость самолета на этой орбите

$$v_z = \frac{H+R}{T} \approx 0.48793 \text{ км/с}$$

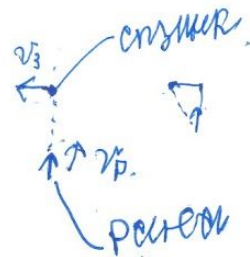
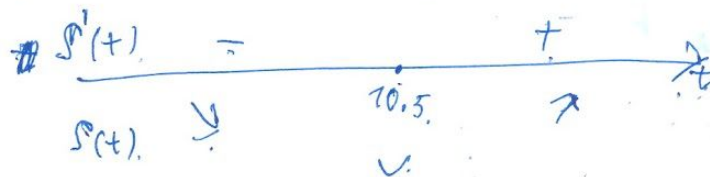
$$S(t) = (t \cdot v_p - 20)^2 + (v_z \cdot t)^2 = t^2(v_p^2 + v_z^2) - 40t v_p + 400$$

$$S'(t) = 2t(v_p^2 + v_z^2) - 40v_p$$

$$= (v_p^2 + v_z^2)t^2 - 40t v_p + 400$$

$$S'(t) = 2(v_p^2 + v_z^2)t - 40v_p$$

$$t_0 = \frac{70.8}{6.7479572} \approx 10.5$$



ошибка +2

$$S(t_0) = (t_0 \cdot v_p - 20)^2 + (v_z \cdot t_0)^2 = 2.002225 + 5.723265 =$$

$$= 7.72549 \text{ км.}$$

ответ: 7.72549 км.

я не уверен что правильно посчитал по орбите т.к. в оне не могу.

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 18

12

2

$L = 11.78 \text{ м}$
 $R = 5 \text{ м}$
 $L_0 = 2R\pi = 10\pi \approx 31.4 \text{ м} \Rightarrow$
 $\Rightarrow$ тело не достигнет ~~вершины~~ вершины точки.

$$L \cdot R = L - \frac{R\pi}{2}$$

$$\alpha = \frac{11.78 - \frac{5 \cdot 3.14}{2}}{5} =$$

$$= \frac{11.78 - 7.85}{5} = 0.786$$

$$\alpha = 0.786$$

$$H = \sin \alpha \cdot R + R = R(\sin \alpha + 1)$$

~~срочно~~
3C7

$$E_k = E_p + E_{\text{тр}}$$

$$\frac{mv^2}{2} = mgh$$

$$v = \sqrt{2gh}$$

$$\frac{3.93}{7.85} \approx 0.5$$

это означает, что тело не достигнет вершины.

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 18

ПРО РАЙСОНУ 2

3.

$$v_y = v \cdot \frac{R}{L_2} = \frac{3,14 \cdot 5}{2 \cdot 11,78} v \approx 0,6664 \cdot v$$

~~$$v_y^2 = \frac{2H}{L_2} \cdot R \cdot g \cdot 2$$~~

~~$$v_y^2 = \frac{2H}{L_2} \cdot R \cdot g \cdot 2$$~~

$$\sqrt{(H_1 - R) \cdot g \cdot 2} = h =$$

$$= \sqrt{\sin \alpha \cdot R \cdot g \cdot 2} = 10 \sqrt{\sin \alpha}$$

$g = 10$

$$v_y - v_n = v_1$$

$$v_n = \sqrt{h \cdot g \cdot 2}$$

$$v_1 = 0,6664v - 20\sqrt{5\sin \alpha}$$

~~$$H = \frac{gt^2}{2}$$~~

$$\sqrt{\frac{2H}{g}} \approx t$$

$$5(\sin \alpha + 1) \left(\pm v_1 \cdot S \right)$$

$$S = \cos \alpha \cdot R$$

$$\sqrt{\frac{2H}{g}} \cdot (0,6664v - 20\sqrt{5\sin \alpha}) \cdot \cos \alpha \cdot 5$$

и знаем отсюда мы можем узнать
то что мы не знаем

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АТС

Предмет Физика

Номер участника 18

14

4.

~~отв. от участника~~

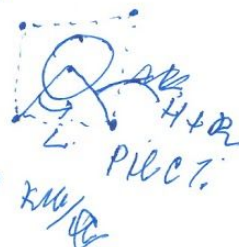
v скорость стрела

$$H = 35786$$

R_z

$$T = 24_y$$

$$v_3 = \frac{2\sqrt{H+R_z}}{T} = 0,48793 \text{ км/с}$$



$$L = \sqrt{2} (H+R_z)$$

$$v = \frac{2L}{v+v_3} = \frac{2\sqrt{2}(35786+6371)}{v+0,48793} = \frac{719238}{v+0,48793}$$

~~отв. от участника~~

мен. орм прыжок почти игнорировать т.к.
 скорость звука земная на этом уровне
 меньше чем скорость сегмента и наоборот.

11А-А-А



Командная инженерная олимпиада «Олимпиада НТИ»

Направление АЭС

Предмет Физика

Номер участника 18

1 см 78%
2 см 14.9% $R = 10 \text{ Ом}$

~~R_{пд}~~
 $R_{пд} - !$

$\$$

N3.

1 см $I_1 = \frac{E}{r+R}$

$I_2 = \frac{E}{r+R+R_{пд}}$

$I_1 = 0.82 = I_2$

2 см $I_1 = \frac{E}{\frac{R}{2} + r}$

$I_2 = \frac{E}{\frac{(R+R_{пд})/R}{2R+R_{пд}} + r}$

$I_2 = 1.149 I_1$

$\frac{0.82}{r+10} = \frac{1}{r+R_{пд}+10}$

$\frac{1}{\frac{100+10R_{пд}}{20+R_{пд}} + r} = \frac{1.149}{5+r}$

$0.82r + 0.82R_{пд} + 8.2 = r + 10$

$0.82R_{пд} - 1.8 = 0.18r$
 $r = \frac{0.82R_{пд} - 1.8}{0.18}$

$= \frac{41}{9} R_{пд} - 10$

$\frac{41}{9} R_{пд} - 5 = 1.149 \left(\frac{100 + R_{пд}}{20 + R_{пд}} + \frac{41}{9} R_{пд} - 10 \right)$

$\frac{41}{9} R_{пд} - 5 = 1.149 \left(\frac{-70}{20 + R_{пд}} + \frac{41}{9} R_{пд} - 9 \right)$

$47R_{пд} - 45 = 47.09 R_{пд} - 930.69 + \frac{-774.09}{20 + R_{пд}}$

Командная инженерная олимпиада «Олимпиада НТИ»

Направление АЭС

Предмет Физика

Номер участника 18

б.

$$\frac{+77773.8}{174.9.9} + 885.69 R_{\pi} = 8607.8 R_{\pi} + 430.09 R_{\pi}^2$$

а

б.

с

$$430.09 R_{\pi}^2 + 7776.11 R_{\pi} - 18747.9 = 0$$

это все бредовые цифры

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = R_{\pi}$$

// отсюда можно найти все что
// нужно

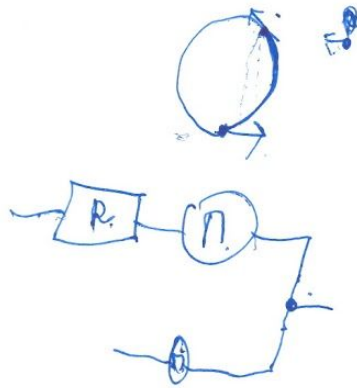
2. πR

31 μ

$$\tilde{I} = 3,14$$

$$15,7$$

$$7,85$$



$$E = I r + I R$$

$$\frac{E}{(r+R)} = I_1 \quad 28\%$$

$$\frac{E}{(R_1 + r + R)} = I_2$$

R

$R + R_1$

$$\frac{E}{\frac{R(R+R_1)}{2R+R_1} + r} = I_2$$

14%

$$\frac{E}{\frac{R_1}{2} + r} = I_2$$

$$I_2 = 1,749 I_1$$



2 3

$$\frac{6}{5}$$

Информатика

Задача 1.1

Код программы на языке C++:

```
1 // ConsoleApplication1.cpp : Defines the entry point
for the console application.
2 //
3
4
5 #include <iostream>
6
7 using namespace std;
8
9
10 int main()
11 {
12     int n, k, m;
13     cin >> n;
14     cin >> k;
15     cin >> m;
16     n--;
17     m--;
18     int A[25];
19     for (int i = 0; i <= n; i++)
20     {
21         cin >> A[i];
22     }
23     int suml = 0, sumr = 0, min = 0, b = A[0];
24     for (int i = 0; i < m - k; i++)
25     {
26         min = 0;
27         for (int j = 1; j < m; j++)
28         {
29             if (A[min] > A[j]) { min = j; }
30         }
31         sumr = sumr + A[min];
32         A[min] = 10001;
33     }
34     A[0] = b;
35     min = 0;
36     for (int i = 0; i < n + 1 - k - m; i++)
37     {
38         min = 0;
39         for (int j = n; j > m; j++)
40         {
41             if (A[min] > A[j]) { min = j; }
42         }
43         suml = suml + A[min];
```

```

44         A[min] = 10001;
45     }
46     if (suml > sumr) { cout << sumr + A[m]; }
47     else { cout << suml + A[m]; }
48     return 0;
49 }

```

Ошибка при выполнении теста №1 (Runtime Error: Segmentation fault).

Задача 1.3

Код программы на языке C++:

```

1  #include <iostream>
2
3  using namespace std;
4
5
6  int main()
7  {
8      long n, k, m;
9      cin >> n;
10     cin >> k;
11     cin >> m;
12     n--;
13     m--;
14     long A[25];
15     for (long i = 0; i <= n; i++)
16     {
17         cin >> A[i];
18     }
19     for (long i = i; i < 25; i++)
20     {
21         A[i] = 10001;
22     }
23     long suml = 0, sumr = 0, min = 0, b = A[0];
24     for (long i = 0; i < m - k; i++)
25     {
26         min = 0;
27         for (long j = 1; j < m; j++)
28         {
29             if (A[min] > A[j]) { min = j; }
30         }
31         sumr = sumr + A[min];
32         A[min] = 10001;
33     }
34     A[0] = b;
35

```

```

36     min = 0;
37     for (long i = 0; i < n + 1 - k - m; i++)
38     {
39         min = 0;
40         for (long j = n; j > m; j++)
41         {
42             if (A[min] > A[j]) { min = j; }
43         }
44         suml = suml + A[min];
45         A[min] = 10001;
46     }
47     if (suml > sumr) { cout << sumr + A[m]; }
48     else { cout << suml + A[m]; }
49     return 0;
50 }

```

Ошибка при выполнении теста №1 (Runtime Error: Segmentation fault).

Командная часть

Результаты были получены в рамках выступления команды: Фортуна

Личный состав команды:

- Акимов Артём
- Баханов Артём
- Доблер Ольга
- Урванцев Матвей
- Цветков Иван

Фото команды с собранными устройствами:



Рис.1 - Команда с четырьмя устройствами в сборке



Рис.2 - Участник команды Артём Акимов собирает коптер для выполнения задач подтрека “Аэронет”

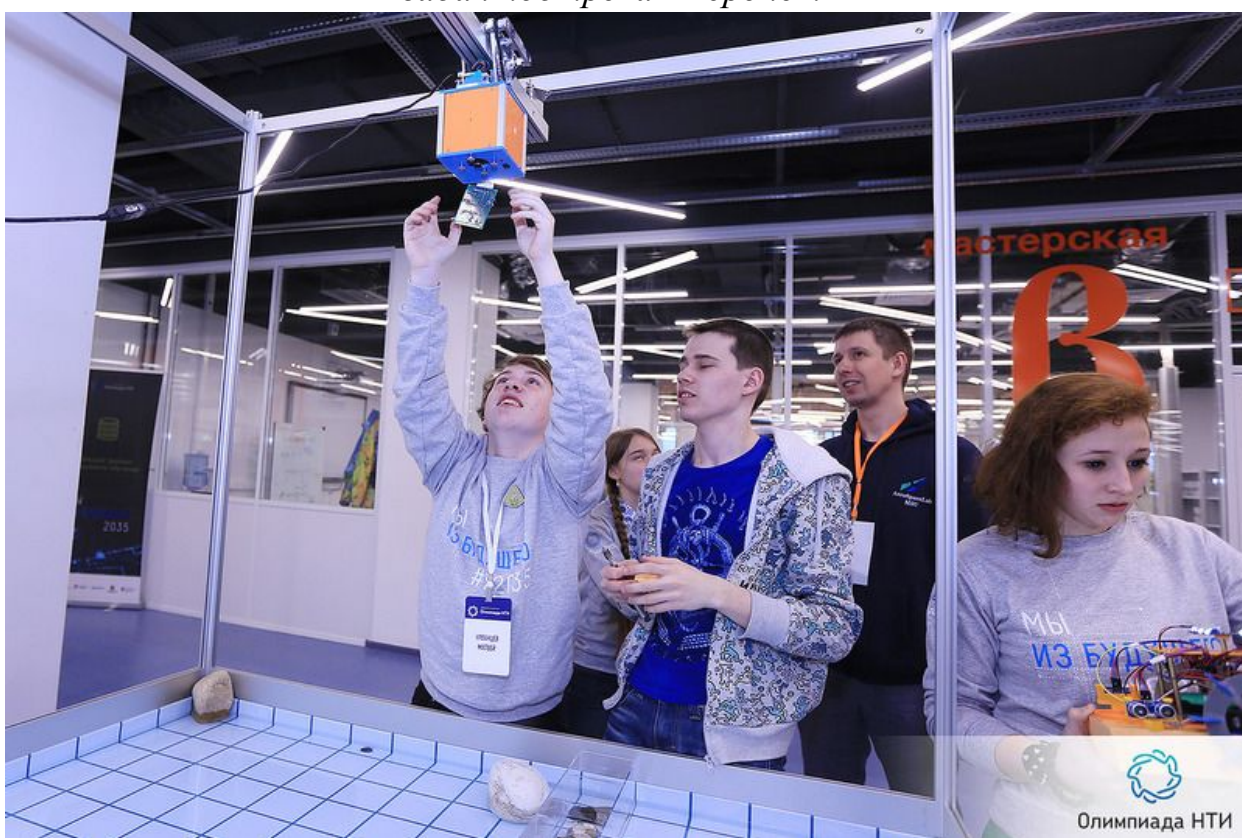


Рис.3 - Команда “Фортуна” занимается отладкой на подтреке “Спейснет”, на заднем плане - судья

Баллы командного этапа:

Название команды	МАРИНЕТ	АВТОНЕТ	АЭРОНЕТ	СПЕЙСНЕТ	КОМПЛЕКС	ИТОГО
Фортуна	8	18	6	29		61

Маринет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Фортуна	Акимов	Артём	1	9	8										8
	Баханов	Артем	2.	5											0
	Доблер	Ольга	3.	5											0
	Урванцев	Матвей	4.	17											0
	Цветков	Иван	5.	5											0
			6.	9											0
РЕЗУЛЬТАТ															8

Автонет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Фортуна	Акимов	Артём	1	10	10										10
	Баханов	Артем	2.	2	0										0
	Доблер	Ольга	3.	4	4										4
	Урванцев	Матвей	4.	2	4										4
	Цветков	Иван	5.	2											0
			6.	8											0
			7.	20											0
РЕЗУЛЬТАТ															18

Аэронет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Фортуна	Акимов	Артём	1	10	6										6
	Баханов	Артем	2.	8											0
	Доблер	Ольга	3.	12											0
	Урванцев	Матвей	4.	15											0
	Цветков	Иван	5.	10											0
			6.	5											0
РЕЗУЛЬТАТ															6

Спейснет

Название команды	Участники		№ подзадач	База	Номер попытки										лучшая
					1	2	3	4	5	6	7	8	9	10	
Фортуна	Акимов	Артём	1	10	10										10
	Баханов	Артем	2.	5	0	0	4								4
	Доблер	Ольга	3.	15	15										15
	Урванцев	Матвей													0
	Цветков	Иван													0
															0
РЕЗУЛЬТАТ															29

Комплексный проезд

Не приступали.

Решения командного этапа:

Маринет

Код программы на языке C++, обеспечивающий решение задач

```

подтрека:
uz.ino
#include <Servo.h>
#include <Ultrasonic.h>
Servo m;
Servo d;
Ultrasonic u(12, 13);
int minSpeedB = 66;
int maxSpeedB = 59;
int minSpeedF = 104;
int maxSpeedF = 105;
int righthAngle = 73;
int S = 30;
bool isBack = false;
int prevS = 0;
void setup() {
    m.attach(8);
    d.attach(9);
    Serial.begin(9600);
    prevS = u.Ranging(CM);
}

void loop() {

    //*****
    //Serial.println(m.read());
    //m.write(93.5);
    //d.write(102.6);
    //delay(100);
    //d.write(90);
    //delay(100);
    float s = u.Ranging(CM);
    float ss = prevS*0.9 + s*0.1;
    prevS = s;
    Serial.println(ss);
    if (ss > 40){
        isBack = false;
        moveFrwr(-1, 103.4);
    }
    else if (ss <= 40 && ss >= 30){
        isBack = false;
        float Kp = (ss - S)/10;
        moveFrwr(-1, minSpeedF - Kp * 2);
    }
    else if (ss < 30 && ss >= 20){
        /*if (isBack == false){
            isBack = true;
            m.write(101);
            delay(20);
            m.write(40);
        }
    }
}

```

```

        delay(150);
        m.write(90);
    }*/
    float Kp = (S - ss)/10;
    moveBack(-1, 63.8 - Kp * 2);
}
else{
    /*if (isBack == false){
        isBack = true;
        m.write(101);
        delay(20);
        m.write(40);
        delay(150);
        m.write(90);
    }*/
    moveBack(-1, 63.8);
}
//moveFrwrdsSonic(s);

}

void moveFrwrdsSonic (float s){
    float k = 1;
    while (s >= 30){
        s = u.Ranging(CM);
        if (s > 40){
            m.write (103.2);
        }
        else if (s <= 40){
            m.write (minSpeedF - (s - S)*k);
        }
    }
}

    delay(20);
    moveBackSonic(s);
    m.write(101);
    delay(20);
    m.write(40);
    delay(150);
    m.write(90);
}

void moveBackSonic (float s){
    float k = 1.7;
    while (s < 30){
        s = u.Ranging(CM);

```

```

        if (s < 20){
            m.write (66);
        }
        else if (s >= 20){
            m.write (minSpeedB - (S - s)*k);
        }

        delay(20);
        moveFrwrdsSonic(s);
    }
}

void moveFrwrds (int timing, float power){
    m.write(power);
    if (timing != -1){
        delay(timing);
    }
    else {
        delay(100);
    }
}

void moveBack (int timing, float power){

    isBack = true;
    m.write(101);
    delay(20);
    m.write(40);
    delay(150);
    m.write(90);

    m.write(power);
    if (timing != -1) {
        delay(timing);
    }
    else {
        delay(100);
    }
}

void moveStop(){
    m.write(90);
    delay(150);
}

```

Автонет

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

```

Irreciever.ino
#include <stdint.h>

```

```

#include "irreceiver.h"
#include <Servo.h>
#define leftLine A1
#define rightLine A2
#define LED 11
#define dynamic 6

const uint8_t IR_RECEIVER_PIN = 7;

IrReceiver receiver(IR_RECEIVER_PIN);

const uint8_t SIGNAL_FORWARD = 0xC0;
const uint8_t SIGNAL_STOP = 0xF0;
const uint8_t SIGNAL_LEFT = 0xA0;
const uint8_t SIGNAL_RIGHT = 0x90;

volatile int state = HIGH;
const float power = 103.9;
const float isWhite; // указать
float dynamicSound; // подобрать годный звук
int timing = 0;
bool isBlink = false;
bool isFirstLine = false;
bool isSecondLine = false;
bool isBBlack = false;
int stepp = 1;
int ser = 94;
int x;
Servo motor; //важно! установить переднюю серву на ровно 90
градусов, иначе машинка может поворачивать, что нам совершенно
не нужно
Servo servoF;
void setup() {
    motor.attach(8);
    servoF.attach(9); // (???)
    motor.write(power);
    servoF.write(ser);
    pinMode(leftLine, INPUT);
    pinMode(rightLine, INPUT);
    pinMode(LED, OUTPUT);
    pinMode(dynamic, OUTPUT);
    Serial.begin(9600);
    Serial1.begin(9600);
    Serial1.println("AT+C001");
    Serial1.setTimeout(100);
    if(Serial1.available() > 0){
        String str = Serial1.readString();
        x = str.substring(3,7).toInt();
    }
    if (x < 540){

```

```

    // слева
    while (!receiver.isDataReceived()) {

    }

    uint8_t signal = receiver.getReceivedData() >> 4;
    if (signal == SIGNAL_FORWARD){
        forwardLeft();
    }
    else if (signal == SIGNAL_RIGHT){
        right();w
    }
}
else {
    // справа
    while (!receiver.isDataReceived()) {

    }

    uint8_t signal = receiver.getReceivedData() >> 4;
    if (signal == SIGNAL_FORWARD){
        forward();

    }
    else if (signal == SIGNAL_LEFT){
        left();
    }
}
while (!receiver.isDataReceived()) {

}

uint8_t signal = receiver.getReceivedData() >> 4;

}

void forwardRight(){
    while (true){
        String str = Serial1.readString();
        x = str.substring(3,7).toInt();
        servoF.write(x);
        if (x < 270){
            servoF.write(73);
        }
        else if (x > 270){
            servoF.write(110);

        }
    }
}

```

```

}
void right(){

}
void left(){

}

void loop() {

    timing += 1;
    servoF.write(ser);
    float rLine = analogRead(leftLine);
    float lLine = analogRead(rightLine);
    Serial.println(rLine);
    Serial.println(lLine);
    if (rLine < 300 && lLine > 300){
        servoF.write(73);
    }
    else if (lLine < 300 && rLine > 300){
        servoF.write(110);

    }
    else if (rLine > 300 && lLine > 300){
        servoF.write(ser);
    }
    //Serial.print(isFirstLine, isSecondLine);
    if (isBlink && timing % 5){
        state = !state;
        digitalWrite(LED, state);
    }
    if (stepp == 1){
        Serial.println(leftLine, rightLine);
        if (rLine < 300 && lLine < 300){
            isFirstLine = true;
            stepp = 2;
            isBlink = true;
        }
    }
    else if (stepp == 2){
        float rLine = analogRead(leftLine);
        float lLine = analogRead(rightLine);
        if (rLine > 300 && lLine > 300){
            stepp = 3;
            isBBlack = true;

        }
    }
    else if (stepp == 3){

```

```

        isSecondLine = true;
        Serial.print(leftLine);
        Serial.print(" ");
        Serial.println(rightLine);
        Serial.println(1000);
        if (rLine < 300 && lLine < 300){
            stepp = 4;
            isBlink = false;
            digitalWrite(LED, HIGH);
            delay(150);
            digitalWrite(LED, LOW);
            motor.write(90);
            digitalWrite(dynamic, HIGH);
            delay(600);
            digitalWrite(dynamic, LOW);

        }
    }
    delay(50);
}
Irreceiver.cpp
#include "irreceiver.h"

#include <Arduino.h>

const unsigned int START_TIMEOUT = 2500;
const unsigned int DATA_STAGE_1_TIMEOUT = 1200;
const unsigned int DATA_STAGE_2_TIMEOUT = 900;
const unsigned int CHECKING_TIMEOUT = 10000;

static IrReceiver *active_receiver = NULL;

static void initTimer()
{
    TCCR4A = 0;
    TCCR4B = 0;
    TCCR4C = 0;
    TCCR4D = 0;
    TCCR4E = 0;
}

static void startTimer(unsigned int us)
{
    TC4H = 0;
    TCNT4 = 0;
    OCR4C = microsecondsToClockCycles(us) / 1024;
    TCCR4B = (1 << PSR4) |
              (1 << CS43) | (0 << CS42) |
              (1 << CS41) | (1 << CS40);
    TIMSK4 = (1 << TOIE4);
}

```

```

static void stopTimer()
{
    TIMSK4 = 0;
    TCCR4B = 0;
}

static void edgeHandler()
{
    if (active_receiver != NULL) {
        active_receiver->handleEdge();
    }
}

ISR(TIMER4_OVF_vect)
{
    if (active_receiver != NULL) {
        active_receiver->handleTimeout();
    }
}

void IrReceiver::init()
{
    _state = STATE_BEGIN;

    initTimer();

    active_receiver = this;

    pinMode(_pin, INPUT);

    attachInterrupt(digitalPinToInterrupt(_pin),
                    edgeHandler,
                    FALLING);
}

void IrReceiver::handleEdge()
{
    stopTimer();

    if (_state == STATE_BEGIN) {
        startTimer(START_TIMEOUT);

        _state = STATE_START;
        return;
    }

    if (_state == STATE_START) {
        _state = STATE_BEGIN;
        return;
    }
}

```

```

    if (_state == STATE_DATA_STAGE_1) {
        startTimer(DATA_STAGE_2_TIMEOUT);

        _state = STATE_DATA_STAGE_2;
        return;
    }

    if (_state == STATE_DATA_STAGE_2) {
        _state = STATE_BEGIN;
        return;
    }

    if (_state == STATE_CHECKING) {
        _state = STATE_BEGIN;
        return;
    }
}

void IrReceiver::handleTimeout()
{
    stopTimer();

    if (_state == STATE_START) {
        if (digitalRead(_pin) == HIGH) {
            startTimer(DATA_STAGE_1_TIMEOUT);
            _data = 0;
            _bit_counter = 12;

            _state = STATE_DATA_STAGE_1;
        } else {
            _state = STATE_BEGIN;
        }
        return;
    }

    if (_state == STATE_DATA_STAGE_1) {
        _state = STATE_BEGIN;
        return;
    }

    if (_state == STATE_DATA_STAGE_2) {
        if (digitalRead(_pin) == HIGH) {
            _data <<= 1;
        } else {
            _data <<= 1;
            _data |= 1;
        }

        _bit_counter--;
    }
}

```

```

        if (_bit_counter > 0) {
            startTimer(DATA_STAGE_1_TIMEOUT);

            _state = STATE_DATA_STAGE_1;
        } else {
            startTimer(CHECKING_TIMEOUT);

            _state = STATE_CHECKING;
        }
        return;
    }

    if (_state == STATE_CHECKING) {
        _state = STATE_RECEIVED;
        return;
    }
}

bool IrReceiver::isDataReceived()
{
    cli();
    State state = _state;
    sei();
    return (state == STATE_RECEIVED);
}

uint16_t IrReceiver::getReceivedData()
{
    if (isDataReceived()) {
        cli();
        uint16_t data = _data;
        _state = STATE_BEGIN;
        sei();
        return data;
    } else {
        return 0;
    }
}

#include "irreceiver.h"

#include <Arduino.h>

const unsigned int START_TIMEOUT = 2500;
const unsigned int DATA_STAGE_1_TIMEOUT = 1200;
const unsigned int DATA_STAGE_2_TIMEOUT = 900;
const unsigned int CHECKING_TIMEOUT = 10000;

static IrReceiver *active_receiver = NULL;

static void initTimer()

```

```

{
    TCCR4A = 0;
    TCCR4B = 0;
    TCCR4C = 0;
    TCCR4D = 0;
    TCCR4E = 0;
}

static void startTimer(unsigned int us)
{
    TC4H = 0;
    TCNT4 = 0;
    OCR4C = microsecondsToClockCycles(us) / 1024;
    TCCR4B = (1 << PSR4) |
              (1 << CS43) | (0 << CS42) |
              (1 << CS41) | (1 << CS40);
    TIMSK4 = (1 << TOIE4);
}

static void stopTimer()
{
    TIMSK4 = 0;
    TCCR4B = 0;
}

static void edgeHandler()
{
    if (active_receiver != NULL) {
        active_receiver->handleEdge();
    }
}

ISR(TIMER4_OVF_vect)
{
    if (active_receiver != NULL) {
        active_receiver->handleTimeout();
    }
}

void IrReceiver::init()
{
    _state = STATE_BEGIN;

    initTimer();

    active_receiver = this;

    pinMode(_pin, INPUT);

    attachInterrupt(digitalPinToInterrupt(_pin),
                    edgeHandler,

```

```

        FALLING);
    }

void IrReceiver::handleEdge()
{
    stopTimer();

    if (_state == STATE_BEGIN) {
        startTimer(START_TIMEOUT);

        _state = STATE_START;
        return;
    }

    if (_state == STATE_START) {
        _state = STATE_BEGIN;
        return;
    }

    if (_state == STATE_DATA_STAGE_1) {
        startTimer(DATA_STAGE_2_TIMEOUT);

        _state = STATE_DATA_STAGE_2;
        return;
    }

    if (_state == STATE_DATA_STAGE_2) {
        _state = STATE_BEGIN;
        return;
    }

    if (_state == STATE_CHECKING) {
        _state = STATE_BEGIN;
        return;
    }
}

void IrReceiver::handleTimeout()
{
    stopTimer();

    if (_state == STATE_START) {
        if (digitalRead(_pin) == HIGH) {
            startTimer(DATA_STAGE_1_TIMEOUT);
            _data = 0;
            _bit_counter = 12;

            _state = STATE_DATA_STAGE_1;
        } else {
            _state = STATE_BEGIN;
        }
    }
}

```

```

        return;
    }

    if (_state == STATE_DATA_STAGE_1) {
        _state = STATE_BEGIN;
        return;
    }

    if (_state == STATE_DATA_STAGE_2) {
        if (digitalRead(_pin) == HIGH) {
            _data <<= 1;
        } else {
            _data <<= 1;
            _data |= 1;
        }

        _bit_counter--;

        if (_bit_counter > 0) {
            startTimer(DATA_STAGE_1_TIMEOUT);

            _state = STATE_DATA_STAGE_1;
        } else {
            startTimer(CHECKING_TIMEOUT);

            _state = STATE_CHECKING;
        }
        return;
    }

    if (_state == STATE_CHECKING) {
        _state = STATE_RECEIVED;
        return;
    }
}

bool IrReceiver::isDataReceived()
{
    cli();
    State state = _state;
    sei();
    return (state == STATE_RECEIVED);
}

uint16_t IrReceiver::getReceivedData()
{
    if (isDataReceived()) {
        cli();
        uint16_t data = _data;
        _state = STATE_BEGIN;
        sei();
    }
}

```

```

        return data;
    } else {
        return 0;
    }
}
IrReceiver.h
#pragma once

#include <stdint.h>

class IrReceiver
{
public:
    IrReceiver(uint8_t pin) :
        _pin(pin),
        _state(STATE_BEGIN),
        _data(0),
        _bit_counter(0)
    {
    }

    void init();

    void handleEdge();

    void handleTimeout();

    bool isDataReceived();

    uint16_t getReceivedData();

private:
    enum State
    {
        STATE_BEGIN,
        STATE_START,
        STATE_DATA_STAGE_1,
        STATE_DATA_STAGE_2,
        STATE_CHECKING,
        STATE_RECEIVED
    };

private:
    uint8_t _pin;
    volatile State _state;
    volatile uint16_t _data;
    volatile int _bit_counter;
};

```

```

Car_4nfull.ino
#include <Servo.h>
#define leftLine A1
#define rightLine A2
#define LED 11
#define dynamic 6

volatile int state = HIGH;
const float power = 103.9;
const float isWhite; // указать
float dynamicSound; // подобрать годный звук
int timing = 0;
bool isBlink = false;
bool isFirstLine = false;
bool isSecondLine = false;
bool isBBlack = false;
int stepp = 1;
int ser = 94;
int x;
Servo motor; //важно! установить переднюю серву на ровно 90
градусов, иначе машинка может поворачивать, что нам совершенно
не нужно
Servo servoF;
void setup() {
    motor.attach(8);
    servoF.attach(9); //(???)
    motor.write(power);
    servoF.write(ser);
    pinMode(leftLine, INPUT);
    pinMode(rightLine, INPUT);
    pinMode(LED, OUTPUT);
    pinMode(dynamic, OUTPUT);
    Serial.begin(9600);
    Serial1.begin(9600);
    Serial1.println("AT+C001");
    Serial1.setTimeout(100);
}

void loop() {
    if(Serial1.available() > 0){
        String str = Serial1.readString();
        x = str.substring(3,7).toInt();
    }
    timing += 1;
    servoF.write(ser);
    float rLine = analogRead(leftLine);
    float lLine = analogRead(rightLine);
    Serial.println(rLine);
    Serial.println(lLine);
    if (rLine < 300 && lLine > 300){

```

```

    servoF.write(73);
}
else if (lLine < 300 && rLine > 300){
    servoF.write(110);

}
else if (rLine > 300 && lLine > 300){
    servoF.write(ser);
}
//Serial.print(isFirstLine, isSecondLine);
if (isBlink && timing % 5){
    state = !state;
    digitalWrite(LED, state);
}
if (stepp == 1){
    Serial.println(leftLine, rightLine);
    if (rLine < 300 && lLine < 300){
        isFirstLine = true;
        stepp = 2;
        isBlink = true;
    }
}
else if (stepp == 2){
    float rLine = analogRead(leftLine);
    float lLine = analogRead(rightLine);
    if (rLine > 300 && lLine > 300){
        stepp = 3;
        isBBlack = true;

    }
}
else if (stepp == 3){
    isSecondLine = true;
    Serial.print(leftLine);
    Serial.print(" ");
    Serial.println(rightLine);
    Serial.println(1000);
    if (rLine < 300 && lLine < 300){
        stepp = 4;
        isBlink = false;
        digitalWrite(LED, HIGH);
        delay(150);
        digitalWrite(LED, LOW);
        motor.write(90);
        digitalWrite(dynamic, HIGH);
        delay(600);
        digitalWrite(dynamic, LOW);

    }
}
delay(50);

```

```
}
```

Аэронет

Код программы отсутствует; команда собрала и получила зачет только за собранную конструкцию коптера.

Спейснет

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

```
Fortuna_sattelite_powered.ino
#define STEP 10
#define DIR 16
#define POWER_ENABLE 14
#define STEPCOUNT
#define END_SWITCH1 3
#define END_SWITCH2 2

int x = 0;
int y = 0;

void setup() {
  Serial1.begin(9600);
  Serial1.setTimeout(100);
  pinMode(STEP, OUTPUT);
  pinMode(DIR, OUTPUT);
  pinMode(POWER_ENABLE, OUTPUT);
  digitalWrite(DIR, HIGH);
  digitalWrite(POWER_ENABLE, LOW);
}

void loop() {
  if(Serial1.available() > 0){
    String str = Serial1.readString();
    x = str.substring(3,7).toInt();
    y = str.substring(8,12).toInt();
    Serial.print(x);
    Serial.print(" ");
    Serial.println(y);
  }

  if(y > 520){
    digitalWrite(DIR, LOW);
    digitalWrite(POWER_ENABLE, LOW);

    doStep();
  }
  if(y < 480){
    digitalWrite(DIR, HIGH);
```

```

    digitalWrite(POWER_ENABLE, LOW);
    doStep();
}
if(y > 480 && y <520){
    digitalWrite(POWER_ENABLE, HIGH);
}
if(y == 0){
    digitalWrite(POWER_ENABLE, HIGH);
}

    if(digitalRead(END_SWITCH1) == 1){
        digitalWrite(POWER_ENABLE, HIGH);
        delay(2000);
        digitalWrite(POWER_ENABLE, LOW);
    }

}

void doStep() {
    digitalWrite(STEP, HIGH);
    delayMicroseconds(800);
    digitalWrite(STEP, LOW);
    delayMicroseconds(800);
}

```