

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

Профиль «АВТОНОМНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ»

Баханов Артем Григорьевич

Класс: 10

Город: Черноголовка

Школа: МОУ СОШ № 82

Регион: Московская область

Уникальный номер участника: 524

**Команда на заключительном
этапе:** Фортуна

Результаты заключительного этапа:

№	Индивидуальный этап													Командный этап			ИТОГ
	Физика				Информатика								За задачи	коэффициент 40%	баллы	коэффициент 60%	
524	4	6	0	0	6	9	0	0	0	0	0	25	6,66	61.0	14.64	21.307	

Индивидуальная часть
Персональный лист участника с номером 524:



Олимпиада НТИ

ФИО Баташов Артём Григорьевич

Город Черноголовка

Школа № 82

Физика

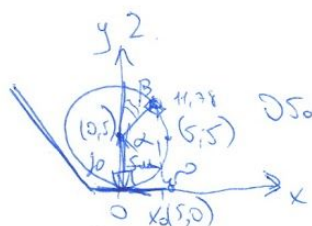
Командная инженерная олимпиада «Олимпиада НТИ»

Направление АЭС

Предмет Физика

Номер участника 524

1	2	3	4	Σ
4	6	0	0	10



Дано:

$$R = 5 \text{ м};$$

$$L = 11.78 \text{ м}$$

Определим точку, в которой тело попало в ширину

летно, или начало декарт. сист. координат.

1-й. - 1-й. ; Соотв. центр. летно имеет коорд.

(0, 5). α - угол α между осью x и радиусом R = 5 м.

Используем формулу

$$L_0 = 2\pi R \approx 31.4 \text{ м}; \text{ отсюда}$$

$$\alpha = 360^\circ \cdot \frac{11.78}{31.4} = 135^\circ$$

$$\beta = 180^\circ - \alpha = 45^\circ$$

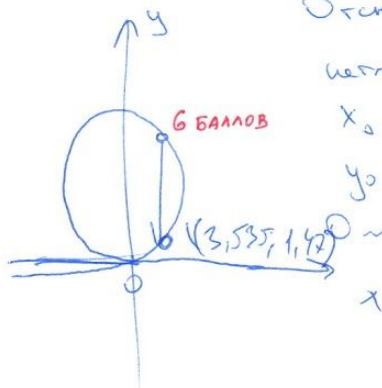
Отсюда можно считать коорд. точки на

летно, или н. ун. летно.

$$x_0 = R \cdot \sin \beta = 5 \cdot 0.707 \approx 3.535$$

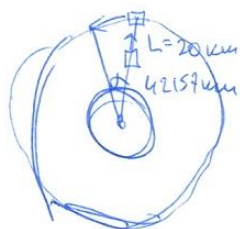
$$y_0 = -R \cdot \cos \beta + R = -3.535 + 5 = 1.47$$

Итак: (3.535; 1.47)



ОБАЛЛОВ

1.



Дано:

$$v_{\text{пр}} = 1.77 \text{ м/с}$$

$$L = 20 \text{ м}$$

Решение:

Используем формулу, по которой радиус

горизонт. β - радиус орбиты.

$$\beta = L / v_{\text{пр}} = 20 / 1.77 = 11.25 \text{ с.}$$

Вектор скорости $v_{\text{пр}}$ направлен по касательной к орбите.

$$R_{\text{г}} = 35786 = 42157 \text{ м}$$

Угловая скорость ω = угл. см. / с

$$1^\circ / \text{с} \approx 0.004167^\circ / \text{с} \quad \text{значит } \alpha = 0.004167 \cdot 11.25 = 0.0469^\circ \text{ см. на обрете} \rightarrow$$

Командная инженерная олимпиада «Олимпиада НТИ»

Направление ИТ

Предмет Физика

Номер участника 524

2. Вырожден.

То есть, по какой-то мере α $\sin \alpha =$
 $= \frac{\alpha}{360} \cdot R_p = 5,49 \text{ мм}$

Но, косинусов.

$$L = \sqrt{2R_p^2 - 2R_p^2 \cos \alpha} =$$

$$= \sqrt{2R_p^2 (1 - \cos \alpha)} = R_p \cdot \sqrt{2(1 - \cos \alpha)} =$$

$$= 34,5 \text{ мм}$$

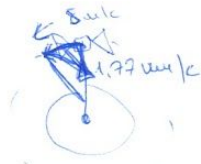
Ответ: 34,5 мм.



4 балла

4

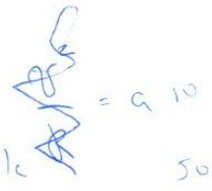
1.



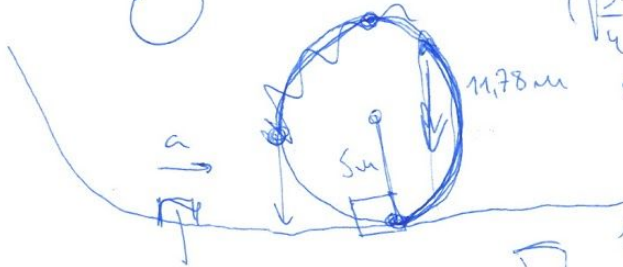
$$h = 35786 \text{ mm}$$

1.3c.
50.4

1.3c



$$R_2 = 6371 \text{ mm}$$



$$15^\circ/4, \quad 0,0041666$$

$$0,047^\circ$$

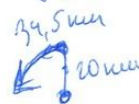
249. a



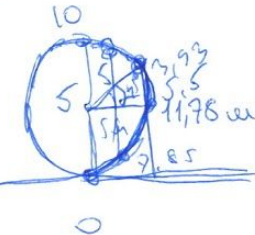
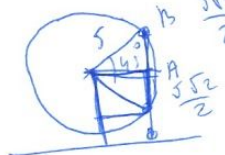
$$c^2 = a^2 + b^2 - 2ab \cos C$$

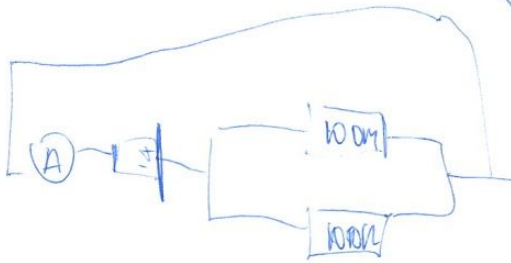
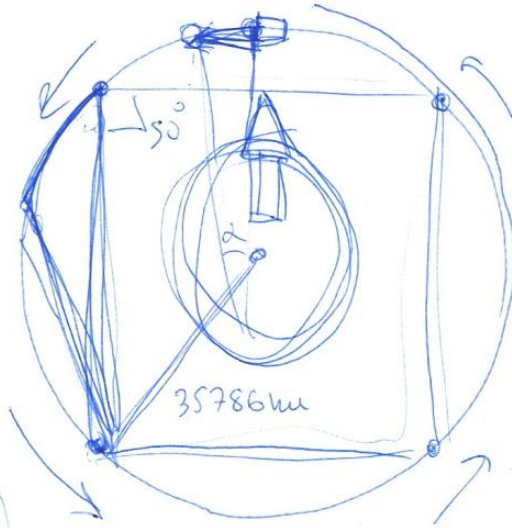


2πR



$$L_5 = 31.4 \text{ mm} \quad 0,375$$





Информатика

Задача 1.1

Код программы на языке Python:

```
1     a = input().split()
2     n, k, m = int(a[0]), int(a[1]), int(a[2])
3     a = input().split()
4     for i in range(len(a)):
5         a[i] = int(a[i])
6     n1 = n - k + 1 ## как мы достаем с левой стороны
7     leftHand = a[m:]
8     rightHand = a[1: m-1]
9     #print(leftHand)
10    #print(rightHand)
11    '''if len(leftHand) == 0 or len(rightHand) == 0:
12        print(a[m-1])
13    else:'''
14    leftsum = 0
15    rightsum = 0
16    for s in leftHand:
17        while len(leftHand) > k-1:
18            t = min(leftHand)
19            leftsum += t
20            leftHand.remove(t)
21    for s in rightHand:
22        while len(rightHand) > k-1:
23            t = min(rightHand)
24            rightsum += t
25            rightHand.remove(t)
26    print(min(rightsum, leftsum)+int(a[m-1]))
```

Программа верно решает задачу (6 баллов).

Задача 1.2

Код программы на языке Python:

```
1     # put your python code here
2     a = input().split()
3     n, k, m = int(a[0]), int(a[1]), int(a[2])
4     a = input().split()
5     for i in range(len(a)):
6         a[i] = int(a[i])
7     n1 = n - k + 1 ## как мы достаем с левой стороны
8     leftHand = a[m:]
9     rightHand = a[1: m-1]
10    #print(leftHand)
11    #print(rightHand)
```

```

12     '''if len(leftHand) == 0 or len(rightHand) == 0:
13         print(a[m-1])
14     else:'''
15     leftsum = 0
16     rightsum = 0
17     for s in leftHand:
18         while len(leftHand) > k-1:
19             t = min(leftHand)
20             leftsum += t
21             leftHand.remove(t)
22     for s in rightHand:
23         while len(rightHand) > k-1:
24             t = min(rightHand)
25             rightsum += t
26             rightHand.remove(t)
27     print(min(rightsum, leftsum)+int(a[m-1]))

```

Программа верно решает задачу (9 баллов).

Задача 1.3

Код программы на языке Python:

```

1     a = input().split()
2     n, k, m = int(a[0]), int(a[1]), int(a[2])
3     a = input().split()
4     for i in range(len(a)):
5         a[i] = int(a[i])
6         n1 = n - k + 1 ## как мы достаем с левой стороны
7         leftHand = a[m:]
8         rightHand = a[1: m-1]
9         #print(leftHand)
10        #print(rightHand)
11        '''if len(leftHand) == 0 or len(rightHand) == 0:
12            print(a[m-1])
13        else:'''
14        leftsum = 0
15        rightsum = 0
16        #for s in leftHand:
17        while len(leftHand) > k-1:
18            t = min(leftHand[0:k])
19            ind = leftHand[0:k].index(t)
20
21            leftsum += t
22            leftHand.pop(ind)
23        #for s in rightHand:
24        while len(rightHand) > k-1:
25            t = min(rightHand[0:k])
26            ind = rightHand[0:k].index(t)
27            rightsum += t
28            rightHand.pop(ind)
29        print(min(rightsum, leftsum)+int(a[m-1]))

```

Ошибка при выполнении теста №28 (превышено время выполнения программы).

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”

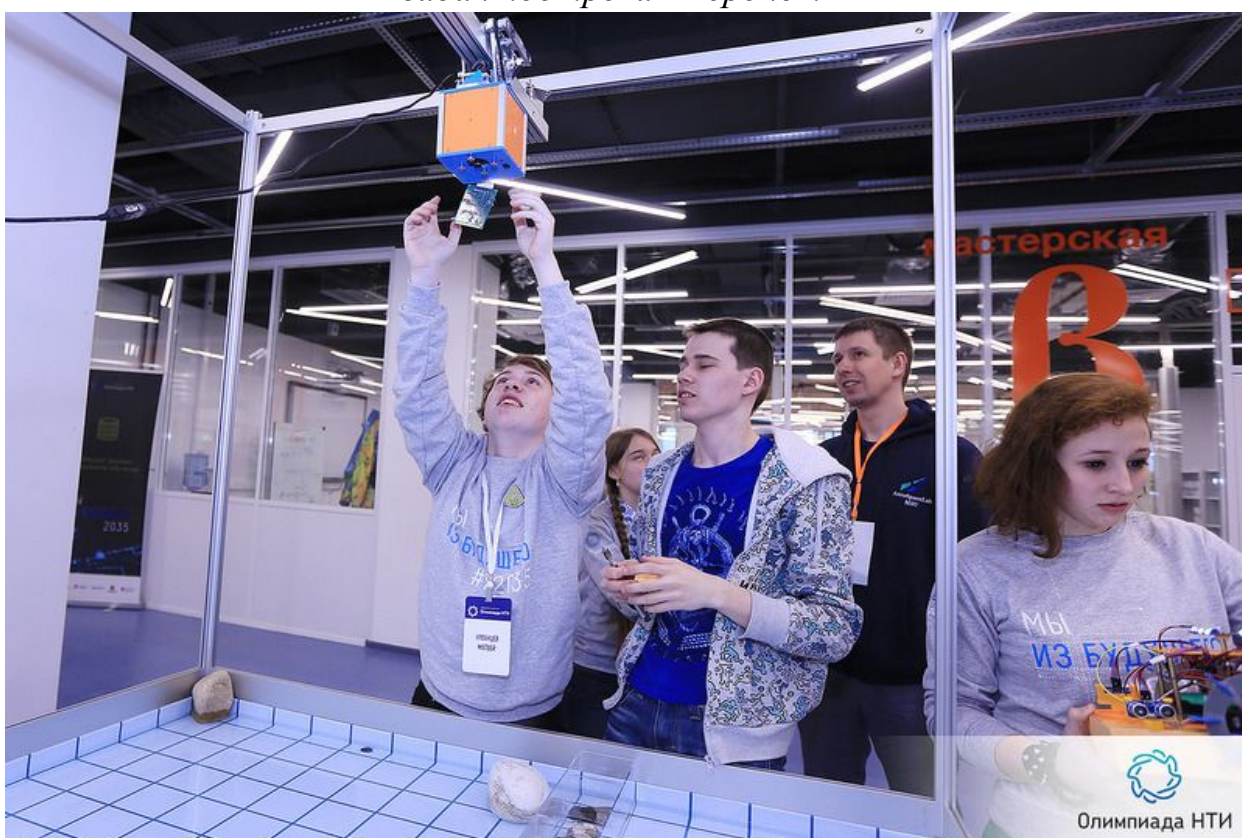


Рис.3 - Команда “Фортуна” занимается отладкой на подтреке “Спейснет”, на заднем плане - судья

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);
}
```