



Проект: **Робот – тренажёр для судомodelистов  
РТС-1**

Автор: *Севастьянова Анастасия*

Организация: *МБОУ ЦДО «ИСТОКИ»  
Клуб Юных Техников «ОМЕГА»*

Город: *Балашиха*

Директор: *Тимошкина Елена Владимировна.*

## Робот-тренажёр для судомоделистов РТС-1

Наш Клуб Юных Техников часто принимает участие в соревнованиях по судомодельному спорту в городе Королёве. Там ребята запускают радиоуправляемые модели судов по траектории между буйками в бассейне. Траектория сложная, в нашем Клубе бассейна нет, и тренироваться негде. Мы тренировались на радиоуправляемых машинках ездой вокруг стаканчиков. Машинки для этого подходят плохо: очень быстро разгоняются, тормозят, не могут разворачиваться на месте. Стаканчики стоят твёрдо, и не всегда понятно, задеты самые дальние или нет. Это совсем не похоже на плавное движение корабля в бассейне между буйками, которые вращаются от малейшего касания. Нужна модель, которая двигалась бы по полу как корабль, и замечала любое касание. Родилась идея сделать для кружка роботизированный радиоуправляемый тренажёр с датчиками, звуковыми и световыми сигналами.

Мой робот-тренажёр РТС-1 сделан на базе платы ARDUINO UNO, которая является программируемым микрокомпьютером и «мозгом» робота. Сигналы от пульта поступают к роботу в стандартный приёмник, обрабатываются по написанной мной программе и передаются через плату управления на 2 мотора. При движении РТС-1 подсвечивается ярким зелёным светом. По бокам в передней части расположены 2 инфракрасных датчика, которые замечают касание предметов. При касании подсветка мигает красным цветом, и подаётся сигнал со звуковой платы. Форма робота в виде корабля. Два ведущих колеса расположены по бокам ближе к носу, а на корме - третья опора в виде шарика. РТС-1 поворачивает при изменении скорости вращения левого и правого колеса и может разворачиваться на месте. Этот манёвр полностью повторяет поворот типа «раздрай» в судомодельном спорте. Расстояние между колесами в 3 раза меньше длины робота. Из-за этого движение получается неуклюжим, на поворотах по инерции он делает размах. Всё это очень похоже на движение модели корабля в бассейне. А самое интересное, что РТС-1 может «прилипнуть» к стене. Если модель судна боком «притрётся» к бортику бассейна, то её нелегко отвести обратно. «Прилипание» портит жизнь новичкам. Робот помогает научиться «отлипнуть» от стены.

**Робот-тренажёр РТС-1 движется по полу как модель корабля в бассейне, замечает любое касание буйков, учит «отлипнуть» от стены, позволяет изменять, улучшать свою программу движения и работы датчиков.**

# Программа для РТС-1 на языке С.

## Написана в среде разработки ARDUINO IDE.

```
#include <TimerOne.h> // podkluchenie biblioteki TimerOne

#define Battery 0 // pin A0 zaryad batarei
#define RC_LEFT_RIGHT 2 // pin D2 kanal levo-pravo
#define RC_FORW_BACK 3 // pin D3 kanal vpered-nazad
#define L_MOTOR_DIR1 4 // pin D4 napravlenie 1 levogo motora
#define L_MOTOR_PWM 5 // pin D5 skorost levogo motora
#define R_MOTOR_PWM 6 // pin D6 skorost pravogo motora
#define R_MOTOR_DIR1 7 // pin D7 napravlenie 1 pravogo motora
#define L_MOTOR_DIR2 8 // pin D8 napravlenie 2 levogo motora
#define RED 9 // pin D9 krasniy LED
#define GREEN 10 // pin D10 zeleniy LED
#define R_MOTOR_DIR2 11 // pin D11 napravlenie 2 pravogo motora
#define IR_Bamper1 15 // pin D15 pravogo IK sensora
#define IR_Bamper2 16 // pin D16 levogo IK sensora
#define SOUND 17 // pin 17 zvukovaya plata

float Delt=170; // chuvstvitelnost okolo nulevogo polozeniya djoystika
volatile unsigned int time[2]; // zapominaet vremya signala resivera
volatile byte state=255; // perekluchaet prerivanie na pod'em ili spad
volatile boolean i=0; // flag kanalov preryvaniy D2->0 D3->1
boolean flag=false;
unsigned long cntLED;
byte stepLED;

void rise(){ // preryvanie po pod'emu signala resivera
    Timer1.restart();
    Timer1.start();
    state=RISING;
}
void fall(){ // preryvanie po spadu signala resivera
    time[i]=Timer1.read();
    Timer1.stop();
    state=FALLING;
}

void setup(){
    Timer1.initialize(2200); // schetchik s periodom 2.2 millisekundi
    Timer1.stop(); // sbros i start schetchika
    Timer1.restart();
    DDRD = DDRD & B11110011; // pin 2,3 na vhod
    PORTD=PORTD | B00001100; // PULL-UP resistor na pin 2 i 3
```

```

DDRC = DDRC & B11111001;           // pin 15,16 na vhod
PORTC=PORTC | B00000110;          // PULL-UP resistor na pin 15 i 16
DDRD = DDRD | B11110000;          // pin 4,5,6,7 na vihod
DDRB = DDRB | B00001111;          // pin 8,9,10,11 na vihod
pinMode(SOUND,OUTPUT);
digitalWrite(SOUND,LOW);
attachInterrupt(i,rise,RISING);    // vkluchenie prerivaniya po pod'emu signala resivera
}

```

```

void loop(){

```

```

    if(analogRead(Battery)< 950) { // esli batareya razryazena do 96% zaryada,
        digitalWrite(R_MOTOR_PWM,LOW); // to stop motory
        digitalWrite(L_MOTOR_PWM,LOW);
        while(1) { // i postoyanno
            delay(70); // migaem kak policeiskaya mashina
            cntLED++; // pri raznyh znacheniyah cntLED

        if ( 7>=cntLED && cntLED> 0) {digitalWrite(GREEN, !digitalRead(GREEN)); digitalWrite(RED,LOW);
        if (13>=cntLED && cntLED> 7) {digitalWrite(RED,!digitalRead(RED));
            digitalWrite(GREEN, !digitalRead(GREEN));}
        if (19>=cntLED && cntLED>13) {digitalWrite(GREEN, !digitalRead(GREEN)); digitalWrite(RED,LOW);}
        if (25>=cntLED && cntLED>19) {digitalWrite(GREEN, LOW); digitalWrite(RED,HIGH);}
            if (cntLED>=25) cntLED=0;
        }
    }

    if((digitalRead(IR_Bamper1)==0 || digitalRead(IR_Bamper2)==0) && !flag) { // proveryaem
        flag=true; // sensory kasaniya
        cntLED = millis(); // esli est kasaniye
    } // to migaem krasnym,
    // zhelytm i zvenim

    if(flag){
        stepLED=byte((millis()- cntLED)/40 );
        if (7 >=stepLED && stepLED> 0) {digitalWrite(RED, stepLED%2);
            digitalWrite(GREEN, LOW);
            digitalWrite(SOUND,HIGH);}
        if (13>=stepLED && stepLED> 7) {digitalWrite(GREEN, stepLED%2);
            digitalWrite(RED,stepLED%2);
            digitalWrite(SOUND,HIGH);}
        if (19>=stepLED && stepLED>13) {digitalWrite(RED, stepLED%2);
            digitalWrite(GREEN, LOW);
            digitalWrite(SOUND,HIGH);}
        if (25>=stepLED && stepLED>19) {digitalWrite(GREEN,LOW);
            digitalWrite(RED,HIGH);
            digitalWrite(SOUND,HIGH);}
        if (stepLED>25){ flag=false; digitalWrite(GREEN,LOW);
            digitalWrite(RED,LOW);
            digitalWrite(SOUND,LOW);}
    }
}

```

```

switch (state){

    case RISING:
        detachInterrupt(i);
        attachInterrupt(i, fall, FALLING);           // perekluchenie prerivaniy
        state=255;
        break;
    case FALLING:
        detachInterrupt(i);
        i=!i;
        attachInterrupt(i, rise, RISING);
        state=255;
        break;
}

float skorost2=map(time[1],1093,1946,100,-100);    // dlitelnost signala mashtabiruem
float skorost1=map(time[0],1081,1940,100,-100);    // v diapason ot -255 do 255
skorost2=constrain(skorost2, -100, 100);          // obresaem previshauchie znachenia,
skorost1=constrain(skorost1, -100, 100);          // t.e. -255<=skorost1<=255
skorost2=skorost2+155*(skorost2/abs(skorost2));
skorost1=skorost1+155*(skorost1/abs(skorost1));
byte modul_sk1= abs(skorost1);
byte modul_sk2= abs(skorost2);

if (skorost2<Delt && skorost2>-Delt){

    if(skorost1>Delt || skorost1<-Delt){
        digitalWrite(GREEN,LOW);                // esli skorost2 (vpered-nazad) vnutri +-Delta,
        digitalWrite(R_MOTOR_DIR1,(skorost1<0)); // a skorost1 (levo-pravo) ne vnutri +-Delta, to
        digitalWrite(R_MOTOR_DIR2,(skorost1>0)); // povorachivaem na meste so skorost1
        digitalWrite(L_MOTOR_DIR1,(skorost1>0));
        digitalWrite(L_MOTOR_DIR2,(skorost1<0));
        analogWrite(R_MOTOR_PWM, modul_sk1);
        analogWrite(L_MOTOR_PWM, modul_sk1);
    }
    else{

        digitalWrite(GREEN,HIGH);                // esli -Delt<skorost1<Delt (levo-pravo),
        digitalWrite(R_MOTOR_PWM, LOW);          // i -Delt<skorost2<Delt (vpered-nazad),
        digitalWrite(L_MOTOR_PWM, LOW);          // to stoim
    }
}
else if(skorost1<Delt && skorost1>-Delt){

    digitalWrite(GREEN,LOW);                    // skorost2 ne vnutri +-Delta, a skorost1
    digitalWrite(R_MOTOR_DIR1,(skorost2>0));    // (levo-pravo) -Delta<skorost1<Delt, to
    digitalWrite(R_MOTOR_DIR2,(skorost2<0));    // edem vpered ili nazad so skorost2
    digitalWrite(L_MOTOR_DIR1,(skorost2>0));
}

```

```

digitalWrite(L_MOTOR_DIR2,(skorost2<0));
analogWrite(R_MOTOR_PWM,modul_sk2);
analogWrite(L_MOTOR_PWM,modul_sk2);
}
else {
float alfa=abs(atan(skorost1/skorost2));
float medleno =((HALF_PI-alfa)*2*modul_sk2/HALF_PI)-modul_sk2/2;
digitalWrite(R_MOTOR_DIR1-3*(skorost1<0),((skorost2*medleno)>0));
digitalWrite(R_MOTOR_DIR2-3*(skorost1<0),((skorost2*medleno)<0));
digitalWrite(R_MOTOR_DIR1-3*(skorost1>0),(skorost2>0));
digitalWrite(R_MOTOR_DIR2-3*(skorost1>0),(skorost2<0));
analogWrite(R_MOTOR_PWM-(skorost1<0),int(abs(medleno)));
analogWrite(R_MOTOR_PWM-(skorost1>0),modul_sk2);
}
}

```

// obschij sluchay, skorost1 i skorost2 ne vnutri +-Delt  
// odno koleso skorost2  
// vpered-nazad, a drugoe  
// pritormaziv so skor medleno





