

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования



**«Пермский национальный исследовательский  
политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 15.03.06 Мехатроника и робототехника

## **О Т Ч Е Т**

### **по учебной практике**

Выполнили студенты гр. МИР-22-26

Гребенщиков Кирилл Владимирович  
(фамилия, имя, отчество)

---

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

---

(оценка)

---

(подпись)

---

(дата)

**Пермь 2023**

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования



**«Пермский национальный исследовательский  
политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 15.03.06 Мехатроника и робототехника

## **О Т Ч Е Т**

### **по учебной практике**

Выполнили студенты гр. МИР-22-26

Тятенков Артём Александрович  
(фамилия, имя, отчество)

---

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

---

(оценка)

---

(подпись)

---

(дата)

**Пермь 2023**

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования



**«Пермский национальный исследовательский  
политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 15.03.06 Мехатроника и робототехника

## **О Т Ч Е Т** **по учебной практике**

Выполнили студенты гр. МИР-22-26

Стрельников Максим Романович  
(фамилия, имя, отчество)

---

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

---

(оценка)

---

(подпись)

---

(дата)

**Пермь 2023**

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования



**«Пермский национальный исследовательский  
политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 15.03.06 Мехатроника и робототехника

## **О Т Ч Е Т**

### **по учебной практике**

Выполнили студенты гр. МИР-22-26

Козлов Павел Дмитриевич  
(фамилия, имя, отчество)

---

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

---

(оценка)

---

(подпись)

---

(дата)

**Пермь 2023**

## **РЕФЕРАТ**

Лодка на радиоуправлении.

Объём отчёта 16 страниц.

Количество иллюстраций 12.

Количество таблиц 1.

Количество используемых источников 4.

Объектом исследования является

Ключевые слова: лодка, аэровинт, электронные компоненты, радиомодуль, NRF24L01, L298N, сервопривод, MR996R, схемы.

Предметом исследования является лодка на радиоуправлении.

Целью исследования является создание лодки на радиоуправлении.

В результате проведенного исследования разработан план по созданию лодки на радиоуправлении.

## Содержание

Введение.....	4
1.Необходимые материалы.....	5
2.Организационно-штатная структура.....	12
Заключение.....	13
Список литературы.....	14

## Введение

Лодка на радиоуправлении которая будет носить название "Титаник". Лодка будет работать на базе микроконтроллера Arduino Nano с сервоприводом и управляться при помощи пульта дистанционного управления по воде за счёт воздушных винтов, соединённых с электродвигателем. А также лодка будет способна поворачивать за счёт рулей, которые будут расположены за аэровинтами, соединёнными с сервоприводом при помощи рычага. Держаться на плаву лодка будет благодаря воздушной камере от велосипеда.

Цель исследования: спроектировать и запрограммировать лодку на радиоуправлении на базе микроконтроллера Arduino Nano.

Задачи исследования:

1. Составить список комплектующих.
2. Составить схему подключения электронных компонентов устройства.
3. Собрать схему электронных компонентов.
4. Установить аппаратную часть в корпус.
5. Написать код в Arduino IDE для передвижения лодки.

## Необходимые материалы

Мы составили список комплектующих, которые нам понадобятся для создания лодки ду.

Список комплектующих:

1. Плата Arduino nano (2шт) (рис.1)
2. Сервопривод (1шт) (рис.2)
3. Электромотор (5В) (2шт) (рис.3)
4. Аккумулятор (9В) (рис.4)
5. Н-мост (L298В) (рис.5)
6. Радиомодуль (2шт) (рис.6)

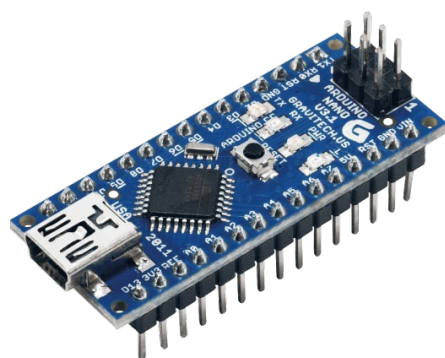


Рисунок №1 – Arduino Nano[Источник №1]





Рисунок №2 – Сервопривод[Источник №2]



Рисунок №3 – Электромотор[Источник №3]



Рисунок №4 – Аккумулятор[Источник №4]

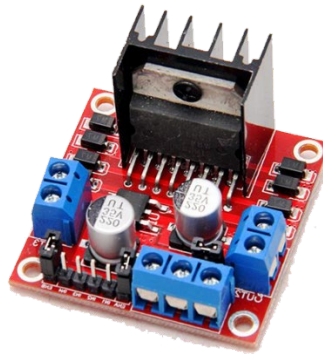


Рисунок №5 – Драйвер L298N[Источник №5]



Рисунок №6 – радиомодуль NRF24L01[Источник №6]

В ходе работы, мы использовали следующие схемы (рис.7) и (рис.8)

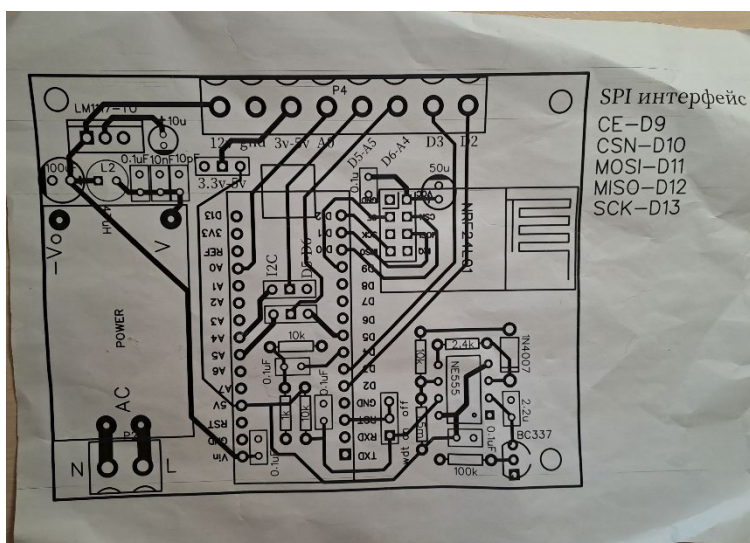


Рисунок 8 – схема платы

Рисунок 9 – распиновка контроллера

Пользуясь информацией в интернете, мы смогли собрать нашу схему для передвижения лодки. Готовая модель (рис. 10) , (рис.11),(рис.12)

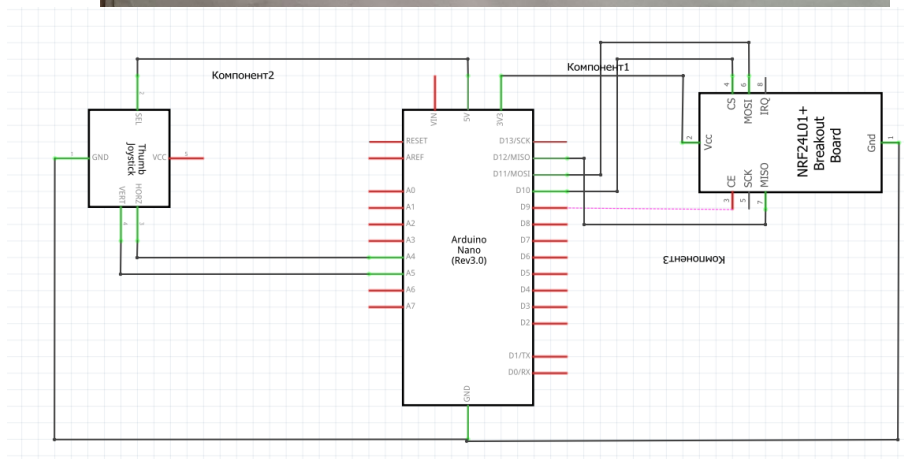
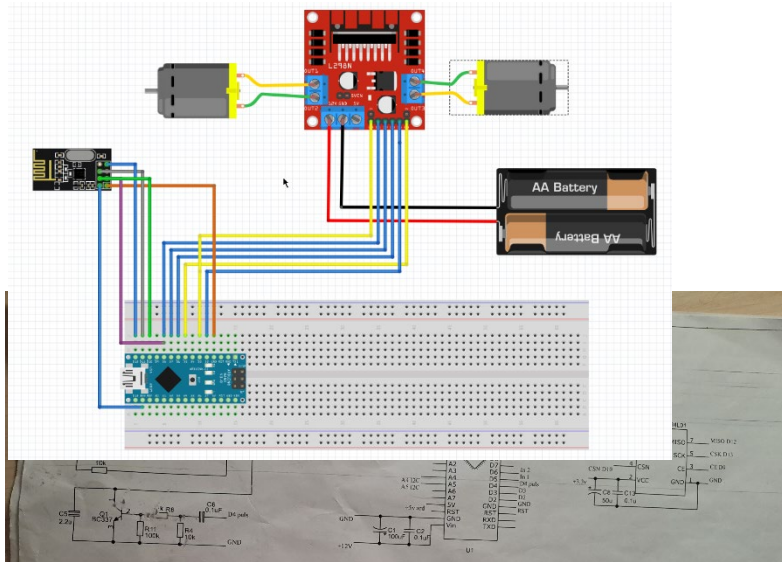


Рисунок 11 – макетная схема устройства лодки

Рисунок 12 – принципиальная схема пульта управления

Для того, чтобы схема правильно работала, в среде программирования Arduino IDE был написан следующий код для приёмника:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x40);

#define SERVOMIN 150 // изменять значения пока не откалибруете сервопривод на по-
ложение в 0 градусов
```

```

#define SERVOMAX 600 // изменять значения пока не откалибруете сервопривод на по-
ложение в требуемое кол-во градусов
#define SERVO_1 0

#include <SPI.h> // Подключаем библиотеку для работы с SPI-интерфейсом
#include <nRF24L01.h> // Подключаем файл конфигурации из библиотеки RF24
#include <RF24.h> // Подключаем библиотеку для работа для работы с модулем
NRF24L01
#define IN1 8 // Input1 подключен к выводу 8
#define IN2 7 // Input2 подключен к выводу 7
#define IN3 6 // Input3 подключен к выводу 6
#define IN4 2 // Input4 подключен к выводу 2
#define EN1 3 // Input4 подключен к выводу 3
#define EN2 5 // Input4 подключен к выводу 5
#define CE 10 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define CSN 9 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля
RF24 radio(CE,CSN); // Создаём объект radio с указанием выводов CE и CSN

byte recieved_data[2]; // массив принятых данных
byte vint = 3; // EN1 на 3 цифровом

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"}; //воз-
можные номера труб

void setup() {
    Serial.begin(9600);
    pinMode (EN1, OUTPUT);
    pinMode (IN1, OUTPUT);
    pinMode (IN2, OUTPUT);
    pinMode (EN2, OUTPUT);
    pinMode (IN4, OUTPUT);
    pinMode (IN3, OUTPUT);
    pwm1.begin(); // активируем PCA9685
    pwm1.setPwmFreq(50); // Задаём частоту подаваемую на сервопривод
    radio.begin(); //активировать модуль
    radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл
    radio.setRetries(0, 15); // (время между попыткой достучаться, число попыток)
    radio.enableAckPayload(); // разрешить отсылку данных в ответ на входящий сиг-
нал
    radio.setPayloadSize(32); // размер пакета, в байтах

    radio.openReadingPipe(1, address[0]); // хотим слушать трубу 0
    radio.setChannel(0x60); // выбираем канал (в котором нет шумов!)

    radio.setPALevel (RF24_PA_MAX); // уровень мощности передатчика. На выбор
RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH, RF24_PA_MAX

```

```

    radio.setDataRate (RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS,
    RF24_1MBPS, RF24_250KBPS
    //должна быть одинакова на приёмнике и передатчике!
    //при самой низкой скорости имеем самую высокую чувствительность и дальность!!

    radio.powerUp();           // начать работу
    radio.startListening();     // начинаем слушать эфир, мы приёмный модуль
}

void loop() {
    byte pipeNo;
    while ( radio.available(&pipeNo)) { // есть входящие данные
        // читаем входящий сигнал
        radio.read(&recieved_data, sizeof(recieved_data));

        // подать на драйвер сигнал с 0 места массива
        analogWrite(vint, recieved_data[0]);

        // повернуть серво на угол 0..180
        // значение получено с 1 элемента массива
        pwm1.Write(recieved_data[1]);
    }
}

```

Для передатчика:

```

#include <SPI.h> // Подключаем библиотеку для работы с SPI-интерфейсом
#include <nRF24L01.h> // Подключаем файл конфигурации из библиотеки RF24
#include <RF24.h> // Подключаем библиотеку для работы с модулем
NRF24L01
#define CE 10 // Номер пина Arduino, к которому подключен вывод CE радиомодуля
#define CSN 9 // Номер пина Arduino, к которому подключен вывод CSN радиомодуля
RF24 radio(CE,CSN); // Создаём объект radio с указанием выводов CE и CSN

byte address[][6] = {"1Node", "2Node", "3Node", "4Node", "5Node", "6Node"}; //воз-
можные номера труб

byte OX = 4; // Ось X джостика на 4 аналоговом
byte OY = 5; // Ось Y джостика на 5 аналоговом

byte transmit_data[2]; // массив, хранящий передаваемые данные
byte latest_data[2]; // массив, хранящий последние переданные данные
boolean flag; // флажок отправки данных

void setup() {
    Serial.begin(9600); //открываем порт для связи с ПК

    radio.begin(); // активировать модуль
    radio.setAutoAck(1); // режим подтверждения приёма, 1 вкл 0 выкл

```

```

    radio.setRetries(0, 15);    // (время между попыткой достучаться, число попыток)
    radio.enableAckPayload();    // разрешить отсылку данных в ответ на входящий сиг-
нал
    radio.setPayloadSize(32);    // размер пакета, в байтах

    radio.openWritingPipe(address[0]); // мы - труба 0, открываем канал для пере-
дачи данных
    radio.setChannel(0x60);      // выбираем канал (в котором нет шумов!)

    radio.setPALevel (RF24_PA_MAX); // уровень мощности передатчика. На выбор
RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH, RF24_PA_MAX
    radio.setDataRate (RF24_250KBPS); // скорость обмена. На выбор RF24_2MBPS,
RF24_1MBPS, RF24_250KBPS
    //должна быть одинакова на приёмнике и передатчике!
    //при самой низкой скорости имеем самую высокую чувствительность и дальность!!

    radio.powerUp();            //начать работу
    radio.stopListening();      //не слушаем радиоэфир, мы передатчик
}

void loop() {

    transmit_data[0] = map(analogRead(OX), 509, 1023, 0, 255 ); // получить значение
// в диапазоне 0..1023, перевести в 0..255, и записать на 1 место в массиве
    transmit_data[1] = map(analogRead(OY), 509, 1023, 0, 255);

    for (int i = 0; i < 2; i++) { // в цикле от 0 до числа каналов
        if (transmit_data[i] != latest_data[i]) { // если есть изменения в trans-
mit_data
            flag = 1; // поднять флаг отправки по радио
            latest_data[i] = transmit_data[i]; // запомнить последнее изменение
        }
    }

    if (flag == 1) {
        radio.powerUp(); // включить передатчик
        radio.write(&transmit_data, sizeof(transmit_data)); // отправить по радио
        flag = 0; //опустить флаг
        radio.powerDown(); // выключить передатчик
    }
}

```

## **Организационно-штатная структура**

Тятенков Артём Александрович - руководитель группы. Во время проведения летней практики в центре робототехники занимался аппаратной составляющей радиоуправляемой лодки.

Стрельников Максим Романович - программный инженер группы. Во время проведения летней практики в центре робототехники занимался созданием кода.

Козлов Павел Дмитриевич – ответственный за сбор информации. Во время проведения летней практики в центре робототехники занимался сбором необходимой информации для дальнейшего развития проекта.

Гребенщиков Кирилл Владимирович – ответственный за обработку и оформление информации. Во время проведения летней практики в центре робототехники занимался обработкой собранной информации и её конспектированием.

## **Заключение**

Нам удалось выполнить все поставленные задачи:

1. Составили список комплектующих.
2. Разработали схему подключения электронных компонентов.
3. Собрать схему для передвижения лодки.
4. Написать программу в Arduino IDE для передвижения лодки.

В ходе работы над поставленными задачами были выявлены следующие проблемы:

1. Слишком большая схема подключения компонентов.
2. Не оптимизированный код.
3. Много работы с установкой компонентов в готовый корпус.

При дальнейшей работе с данным проектом эти проблемы можно решить так:

1. Упростить схему подключения компонентов.
2. Оптимизировать код.
3. Создать корпус с выделенными местами под каждый компонент, напечатанный на 3D-принтере.



## Список источников

1. <https://www.pngwing.com/ru/free-png-xzcpk> - Arduino Nano
2. <https://www.pngwing.com/ru/free-png-xvdhm> - Сервопривод
3. <https://rcdrive.ru/unit.php?unit=24180&pPage=11> - Электромотор
4. <https://www.ozon.ru/product/2-sht-akkumulyator-9v-500mah-6f22-466858374/?sh=6oYUDYFd0Q> - Аккумулятор
5. <https://www.pngwing.com/ru/free-png-zdljf> - Драйвер L298N
6. <https://voltiq.ru/shop/nrf24l01-pa-lna/> - Радиомодуль NRF24L01
7. <https://arduinomaster.ru/uroki-arduino/shema-raboty-n-mosta-dlya-upravleniya-dvigatelyami/> - Datasheet драйвера L298N
8. [http://arduino-kid.ru/arduino\\_nano\\_datasheet](http://arduino-kid.ru/arduino_nano_datasheet) - Datasheet Arduino Nano
9. <http://wiki.amperka.ru/articles:servo> - Datasheet сервопривода