ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ

имени М. Е. ЕВСЕВЬЕВА»

Физико-математический факультет

Кафедра информатики и вычислительной техники

Направление магистерской подготовки

44.04.01 Педагогическое образование

Конкурс свободной робототехники «**Робофабрика**»

Номинация «**Изделие**»

Название работы:

«**Пульт дистанционного управления роботом**»

Саранск 2018

Содержание

[Введение 3](#_Toc509272742)

[1 Обзор комплектующих для пульта дистанционного управления роботом 4](#_Toc509272743)

[1.1 Плата Arduino Nano 4](#_Toc509272744)

[1.2 Радио модуль NRF24L01 PA LNA 8](#_Toc509272745)

[1.3 Однооборотный потенциометр 11](#_Toc509272746)

[1.4 Тумблер трехпозиционный с фиксацией 12](#_Toc509272747)

[1.5 Переключатель двухпозиционный 12](#_Toc509272748)

[1.6 Батарея «крона» 13](#_Toc509272749)

[1.7 Светодиод 14](#_Toc509272750)

[2 Конструирование пульта дистанционного управления роботом 16](#_Toc509272751)

[3 Программирование пульта дистанционного управления роботом 18](#_Toc509272752)

[3.1 Среда разработки Arduino IDE 18](#_Toc509272753)

[3.2 Программа для работы пульта дистанционного управления 21](#_Toc509272754)

[Заключение 27](#_Toc509272755)

[Список использованной литературы 28](#_Toc509272756)

# Введение

Робототехника в настоящее время вызывает большой интерес, являясь движущий силой прогресса. Роботы находят свое применение и в крупном серийном производстве, и в образовании (для обучения детей применяются робототехнические комплекты).

Образовательная робототехника делится на несколько направлений, одним из которых является соревновательная робототехника и отдельным её ответвлением являются бои роботов. Ни один робот без участия человека пока не может полностью функционировать и совершать бой на арене без оператора. Поэтому каждому участнику требуется средство управления своим роботом. Подобным средством является пульт дистанционного управления. Современные пульты имеют цену в достаточно широких пределах от 5 до 50 тысяч рублей. Стоимость зависит от производителя и возможностей пульта (количество каналов, дальность действия, чувствительность джойстиков, наличие жидкокристаллических дисплеев и т.д.).

Объектом данной работы является пульт дистанционного управления роботом.

Предметом данной работы являются функциональные особенности элементов электрической цепи, с целью их объединения в единую систему управления.

Целью работы являются создание пульта дистанционного управления роботом с применением платы Arduino Nano.

Для достижения этой цели поставлены следующие задачи:

**–** изучить комплектующие для пульта;

– сделать электрическую схему пульта;

– собрать пульт из подобранных комплектующих;

– написать программу для работы пульта;

– загрузить программный код в плату Arduino Nano;

– проверка работоспособности пульта.

Научная новизна заключается в разработке электрической схемы пульта, эргономичности пульта, написании уникального программного кода.

Практическую значимость имеет созданный пульт, способный управлять множеством роботизированных систем, работающих на частоте 2,4 ГГц.

# 1 Обзор комплектующих для пульта дистанционного управления роботом

## 1.1 Плата Arduino Nano

*Общие сведения*

Платформа Nano, построенная на микроконтроллере ATmega328 (Arduino Nano 3.0) или ATmega168 (Arduino Nano 2.x), имеет небольшие размеры и может использоваться в лабораторных работах. Она имеет схожую с Arduino Duemilanove функциональность, однако отличается сборкой. Отличие заключается в отсутствии силового разъема постоянного тока и работе через кабель Mini-B USB. Nano разработана и продается компанией Gravitech.

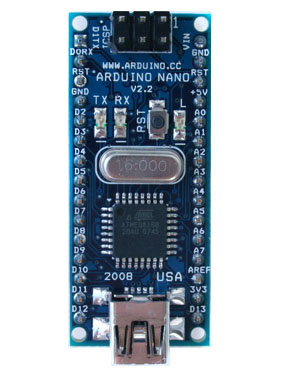


Рисунок 1.1. Внешний вид платы Arduino Nano

Таблица 1.1. Краткие характеристики

|  |  |
| --- | --- |
| Микроконтроллер | Atmel ATmega168 или ATmega328 |
| Рабочее напряжение (логическая уровень) | 5 В |
| Входное напряжение (рекомендуемое) | 7-12 В |
| Входное напряжение (предельное) | 6-20 В |
| Цифровые Входы/Выходы | 14 (6 из которых могут использоваться как выходы ШИМ) |
| Аналоговые входы | 8 |
| Постоянный ток через вход/выход | 40 мА |
| Флеш-память | 16 Кб (ATmega168) или 32 Кб (ATmega328) при этом 2 Кб используются для загрузчика |
| ОЗУ | 1 Кб (ATmega168) или 2 Кб (ATmega328) |
| EEPROM | 512 байт (ATmega168) или 1 Кб (ATmega328) |
| Тактовая частота | 16 МГц |
| Размеры | 1.85 см x 4.2 см |

*Питание:*

Arduino Nano может получать питание через подключение Mini-B USB, или от нерегулируемого 6-20 В (вывод 30), или регулируемого 5 В (вывод 27), внешнего источника питания. Автоматически выбирается источник с самым высоким напряжением.

Микросхема FTDI FT232RL получает питание, только если сама платформа запитана от USB. Таким образом при работе от внешнего источника (не USB), будет отсутствовать напряжение 3.3 В, генерируемое микросхемой FTDI, при этом светодиоды RX и TX мигаю только при наличие сигнала высокого уровня на выводах 0 и 1.

*Память*

Микроконтроллер ATmega168 имеет 16 кБ флеш-памяти для хранения кода программы, а микроконтроллер ATmega328, в свою очередь, имеет 32 кБ (в обоих случаях 2 кБ используется для хранения загрузчика). ATmega168 имеет 1 кБ ОЗУ и 512 байт EEPROM (которая читается и записывается с помощью [библиотеки EEPROM](http://arduino.ru/Reference/Library/EERPOM)), а ATmega328 – 2 кБ ОЗУ и 1 Кб EEPROM.

*Входы и Выходы*

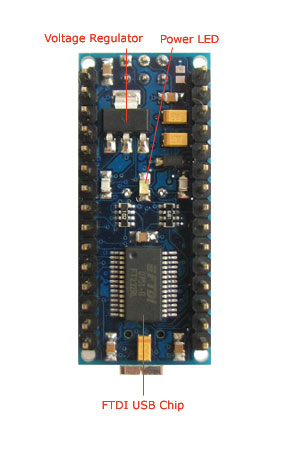
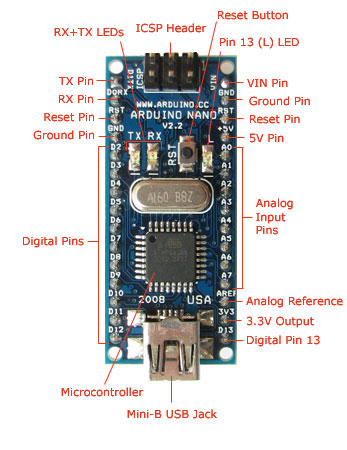


Рисунок 1.2. Распиновка платы Arduino Nano

Каждый из 14 цифровых выводов Nano, используя функции [pinMode()](http://arduino.ru/Reference/PinMode), [digitalWrite()](http://arduino.ru/Reference/DigitalWrite), и [digitalRead()](http://arduino.ru/Reference/DigitalRead), может настраиваться как вход или выход. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:

* **Последовательная шина: 0 (RX) и 1 (TX).** Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины FTDI USB-to-TTL.
* **Внешнее прерывание: 2 и 3.** Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt().
* **ШИМ: 3, 5, 6, 9, 10, и 11.** Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции analogWrite().
* **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)**. Посредством данных выводов осуществляется связь SPI, которая, хотя и поддерживается аппаратной частью, не включена в язык Arduino.
* **LED: 13.** Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

На платформе Nano установлены 8 аналоговых входов, каждый разрешением 10 бит (т.е. может принимать 1024 различных значения). Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством функции analogReference(). Некоторые выводы имеют дополнительные функции:

* **I2C: A4 (SDA) и A5 (SCL).** Посредством выводов осуществляется связь I2C (TWI). Для создания используется библиотека Wire (информация на сайте Wiring).

Дополнительная пара выводов платформы:

* **AREF.** Опорное напряжение для аналоговых входов. Используется с функцией analogReference().
* **Reset**. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

*Связь*

На платформе Arduino Nano установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega168 и ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема FTDI FT232RL направляет данный интерфейс через USB, а драйверы FTDI (включены в программу Arduino) предоставляют виртуальный COM порт программе на компьютере. Мониторинг последовательной шины (Serial Monitor) программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Светодиоды RX и TX на платформе будут мигать при передаче данных через микросхему FTDI или USB подключение (но не при использовании последовательной передачи через выводы 0 и 1).

Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Nano.

ATmega168 и ATmega328 поддерживают интерфейсы I2C (TWI) и SPI. В Arduino включена библиотека Wire для удобства использования шины I2C. Более подробная информация находится в документации. Для использования интерфейса SPI обратитесь к техническим данным микроконтроллеров ATmega168 и ATmega328.

*Программирование*

Платформа программируется посредством ПО Arduino. Из меню Tools > Board выбирается «Arduino Diecimila, Duemilanove или Nano w/ ATmega168» или «Arduino Duemilanove или Nano w/ ATmega328» (согласно установленному микроконтроллеру). Подробная информация находится в справочнике и инструкциях.

Микроконтроллеры ATmega168 и ATmega328 поставляются с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через выводы блока ICSP (внутрисхемное программирование). Подробная информация находится в данной инструкции.

*Автоматическая (программная) перезагрузка*

Nano разработана таким образом, чтобы перед записью нового кода перезагрузка осуществлялась самой программой, а не нажатием кнопки на платформе. Одна из линий FT232RL, управляющих потоком данных (DTR), подключена к выводу перезагрузки микроконтроллеров ATmega168 или ATmega328 через конденсатор 100 нФ. Активация данной линии, т.е. подача сигнала низкого уровня, перезагружает микроконтроллер. Программа Arduino, используя данную функцию, загружает код одним нажатием кнопки Upload в самой среде программирования. Подача сигнала низкого уровня по линии DTR скоординирована с началом записи кода, что сокращает таймаут загрузчика.

Функция имеет еще одно применение. Перезагрузка Nano происходит каждый раз при подключении к программе Arduino на компьютере с ОС Mac X или Linux (через USB). Следующие полсекунды после перезагрузки работает загрузчик. Во время программирования происходит задержка нескольких первых байтов кода во избежание получения платформой некорректных данных (всех, кроме кода новой программы). Если производится разовая отладка скетча, записанного в платформу, или ввод каких-либо других данных при первом запуске, необходимо убедиться, что программа на компьютере ожидает в течение секунды перед передачей данных [8].

## 1.2 Радио модуль NRF24L01 PA LNA

*Описание*

Радио модуль nRF24L01 PA LNA (Nordic Radio Frequency 2.4) - предназначен для приёма и передачи данных по радиоканалу, на разрешённом ISM (Industrial, Scientific, Medical) диапазоне радиочастот. В модуле nRF24L01 PA LNA, данный диапазон разбит на 128 каналов, с шагом 1 МГц: от 2,400 ГГц до 2,527 ГГц. Например, канал 55 означает, что приём и передача будет происходить на частоте 2,455 ГГц. Канал 99 будет передавать/принимать данные на частоте 1,499 ГГц, а канал 0 на частоте 2,400 ГГц. Модуль позволяет выбрать любой из 128 каналов для приёма и (или) передачи данных. На каждом канале можно сформировать сеть из 6 передатчиков и 1 приёмника.

*Общие сведения:*

* Модуль подключается к [аппаратной](http://wiki.iarduino.ru/page/opredelenie-apparatnyh-vyvodov-arduino/) шине SPI.
* Данные передаются по радиоканалу на расстоянии до 1 км в пределах прямой видимости (указано производителем)
* Модулю программно задается роль передатчика или приемника, но в ходе выполнения программы, эту роль можно менять.
* Имеется возможность задавать: мощность передачи (-18 дБм, -12 дБм, -6 дБм, 0 дБм), скорость передачи (250 кб/с, 1 Mб/с, 2 Мб/с), номер канала (0-127), идентификационные номера (5 байт) и т.д.
* Модуль укомплектован внешней антенной.
* Потребляемые модулем токи не превышают 13 мА (как при передаче, так и при приёме).
* Достоверность принимаемых данных обеспечивается передачей циклически избыточного кода CRC
* Реализован функционал контроля доставки данных. Приемник, после успешного получения пакета данных, отправляет передатчику пакет подтверждения приёма. А если передатчик не получил подтверждение от приёмника, то он повторно отправляет пакет данных (этот функционал настраивается в скетче).

*Подключение:*

Модуль подключается к [аппаратной](http://wiki.iarduino.ru/page/opredelenie-apparatnyh-vyvodov-arduino/) шине SPI (выводы: [SCK, MOSI, MISO, SS](http://blog.iarduino.ru/page/podrobnoe-opisanie-vseh-portov-na-raznyh-versiyah-arduino/)), вход модуля CSN (выбор режима) подключается к любому выводу [Arduino](http://iarduino.ru/shop/arduino/), а выход прерывания IRQ не используется (при управлении [библиотекой RF24](http://iarduino.ru/file/27.html)). Модуль поддерживает логические уровни 5 В на информационных выводах, но на выводы питания Vcc и GND подаётся напряжение 3,3 В постоянного тока. Если подключить модуль к напряжению питания 5 В, то он может выйти из строя!

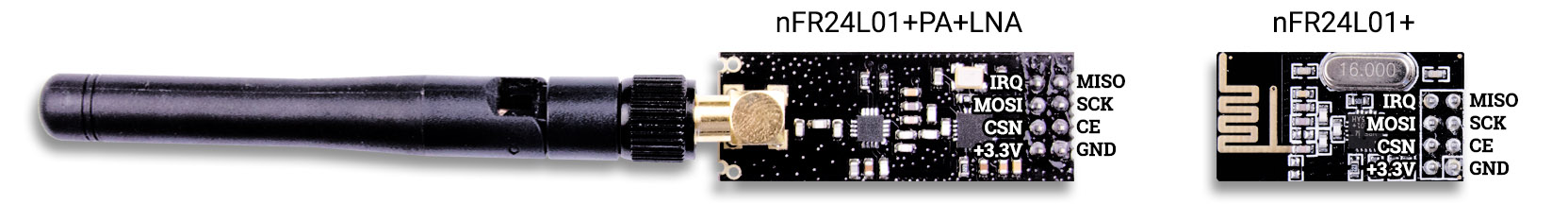
[](http://iarduino.ru/img/upload/6e9385c06b2638158f8c37548db32b56.jpg)

Рисунок 1.3. Внешний вид радио модуля NRF24L01 PA LNA

Для подключения модуля nRF24L01 PA LNA к [Arduino](http://iarduino.ru/shop/arduino/), можно воспользоваться [адаптером](http://iarduino.ru/shop/Expansion-payments/adapter-dlya-nrf24l01.html) в котором имеется собственный стабилизатор напряжения 3,3 В, а так же подписанная однорядная колодка выводов, позволяющая подключить модуль к [Arduino](http://iarduino.ru/shop/arduino/) используя однорядный шлейф. [Адаптер](http://iarduino.ru/shop/Expansion-payments/adapter-dlya-nrf24l01.html) подключается к напряжению питания 5 В постоянного тока.

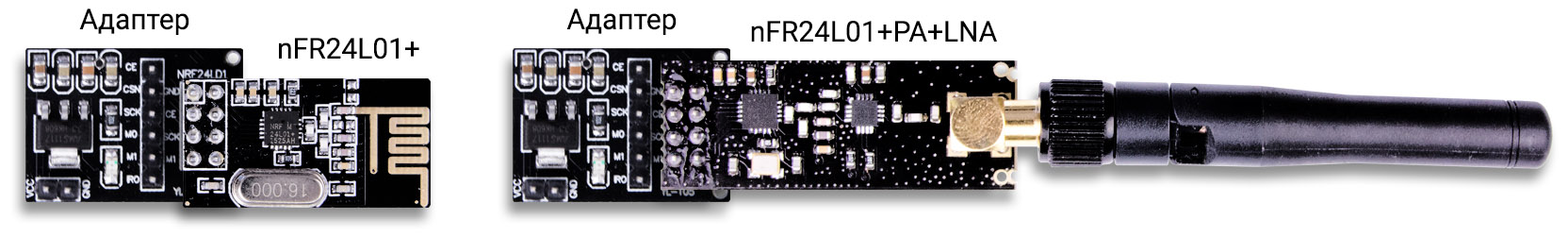
[](http://iarduino.ru/img/upload/6f1818c93a9ec16c5fafd9a18981b7c5.jpg)

Рисунок 1.4. Подключение радио модуля nRF24L01 PA LNA через адаптер

Таблица 1.2. Назначение выходов на адаптере

| **nRF24L01** | **Arduino Uno/Nano** | **Назначение** |
| --- | --- | --- |
| CE / SS | 9 (меняется в скетче) | Шина SPI - выбор устройства |
| CSN | 10 (меняется в скетче) | Выбор режима: приёмник / передатчик |
| SСK | 13 ([SCK](http://blog.iarduino.ru/page/podrobnoe-opisanie-vseh-portov-na-raznyh-versiyah-arduino/)) | Шина SPI - линия тактирования |
| MOSI | 11 ([MOSI](http://blog.iarduino.ru/page/podrobnoe-opisanie-vseh-portov-na-raznyh-versiyah-arduino/)) | Шина SPI - линия данных (от мастера к ведомому) |
| MISO | 12 ([MISO](http://blog.iarduino.ru/page/podrobnoe-opisanie-vseh-portov-na-raznyh-versiyah-arduino/)) | Шина SPI - линия данных (от ведомого к мастеру) |
| IRQ | Не используется | Прерывание |

*Питание:*

Входное напряжение 3,3 В постоянного тока, подаётся на выводы Vcc и GND. Модуль можно подключить через [адаптер](http://iarduino.ru/shop/Expansion-payments/adapter-dlya-nrf24l01.html), на который подаётся напряжение питания 5 В постоянного тока. Если подключить модуль, без адаптера, к напряжению питания 5 В, то он может выйти из строя!

*Управление:*

Так как на одном канале могут одновременно «вещать» до 6 передатчиков, то каждому передатчику нужно задать уникальный идентификатор (pipe ID) - идентификатор трубы. А приемнику задаются все идентификаторы труб (pipe ID) тех передатчиков, данные которых требуется принимать. Таким образом у каждого передатчика только один pipe. А приемнику указываются от одного до шести идентификаторов передатчиков (pipe0 - pipe5). По этим идентификаторам приёмник «понимает» данные какого передатчика он получил. Вы сами придумываете номера идентификаторов труб, они состоят из 5 байт. Но есть несколько условий: • идентификатор каждого передатчика на одном канале должен быть уникальным. • приёмнику задаются идентификаторы передатчиков. • идентификаторы передатчикам задаются так, что у приёмника pipe0 и pipe1 могут отличаться всеми байтами, а pipe2 - pipe5 должны отличаться от pipe1 только последним байтом [4].

Модулям задается не только канал и идентификатор, но и мощность передатчика, скорость передачи данных и т.д.

*Спецификация:*

* Частотный диапазон: ISM (2,400 ... 2,527 ГГц)
* Количество поддерживаемых каналов: 128 (с шагом 1МГц)
* Модуляция: GFSK
* Расстояние между передатчиком и приёмником: до 1 км (в пределах прямой видимости)
* Скорость передачи данных: 0.25, 1, 2 Мб/с (указывается в скетче), на скорости 2 Мб/с используется два канала.
* Мощность передатчика: -18, -12, -6, 0 дБм (указывается в скетче)
* Чувствительность приемника: -82 дБм
* Коэффициент усиления антенны: 2 дБм
* Интерфейс: SPI
* Напряжение питания: 3,3 В (минимально допустимое 1,9 В)
* Напряжение логической «1»: 3,3 ... 5 В
* Потребляемый ток в режиме передачи данных: 11,3 мА (при максимальной мощности передачи 0 дБм)
* Потребляемый ток в режиме приёма данных: 12,3 мА (при максимальной скорости передачи 2 Мб/с)
* Потребляемый ток в режиме энергосбережения:
* Рабочая температура: -40 ... 85 °C
* Габариты: 41х16х20 мм (без учёта антенны)
* Вес: 5 г (без учёта антенны)

## 1.3 Однооборотный потенциометр

*Устройство и работа*

На выводы резистивного элемента подается напряжение, которое предполагается регулировать. Подвижный контакт является регулирующим элементом, который приводится в действие вращением ручки. От подвижного контакта снимается напряжение, которое может находиться в диапазоне от нуля до наибольшей величины, равной входному напряжению на потенциометр, и зависит от текущей позиции подвижного контакта.

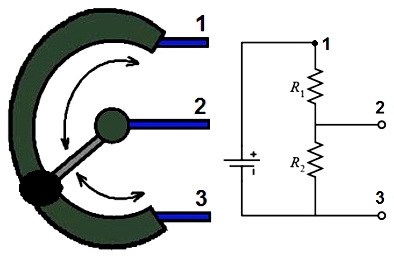


Рисунок 1.5. Принцип работы потенциометра.

Потенциометр действует по типу переменного резистора, однако выполняет функции делителя напряжения. Его резистивный компонент представляет собой два резистора, которые соединены последовательно. Положение скользящего контакта является определяющим в определении отношения величины сопротивления 1-го резистора ко 2-му.

Наиболее популярным стал переменный однооборотный резистор. Он широко применяется в радиотехнике в качестве регулятора громкости, и в других устройствах. При изготовлении потенциометров применяются разные материалы для изготовления резистора: металлическая пленка, токопроводящий пластик, проволока, металлокерамика, углерод.



Рисунок 1.6. Внешний вид однооборотного потенциометра

Скользящий элемент может поворачиваться на один оборот, а точнее, около 270 градусов. На полный оборот поворот невозможен, так как на остальной части сектора поворота размещены клеммы контактов. Наиболее популярными однооборотные переменные резисторы стали в устройствах, не требующих для регулировки более одного оборота [3].

## 1.4 Тумблер трехпозиционный с фиксацией

**Низковольтные переключатели** предназначены для работы в электрических цепях транспортных средств, бытовых и общепромышленных сетях постоянного тока напряжением 12-24В с коммутируемой нагрузкой мощностью до 240Вт [7].

Название продукта: MTS-103

Тип: spdt (ON-OFF-ON)  
Диаметр головки (прибл.): 0.57 см/0.22"  
Шпилька номер: 3  
Размеры: 13 х 8 х 10 мм/0.5 "х 0.31" х 0.4 "(Д \* Ш \* h)  
Мощность: AC 125 В 6А, ac 250 В 3A

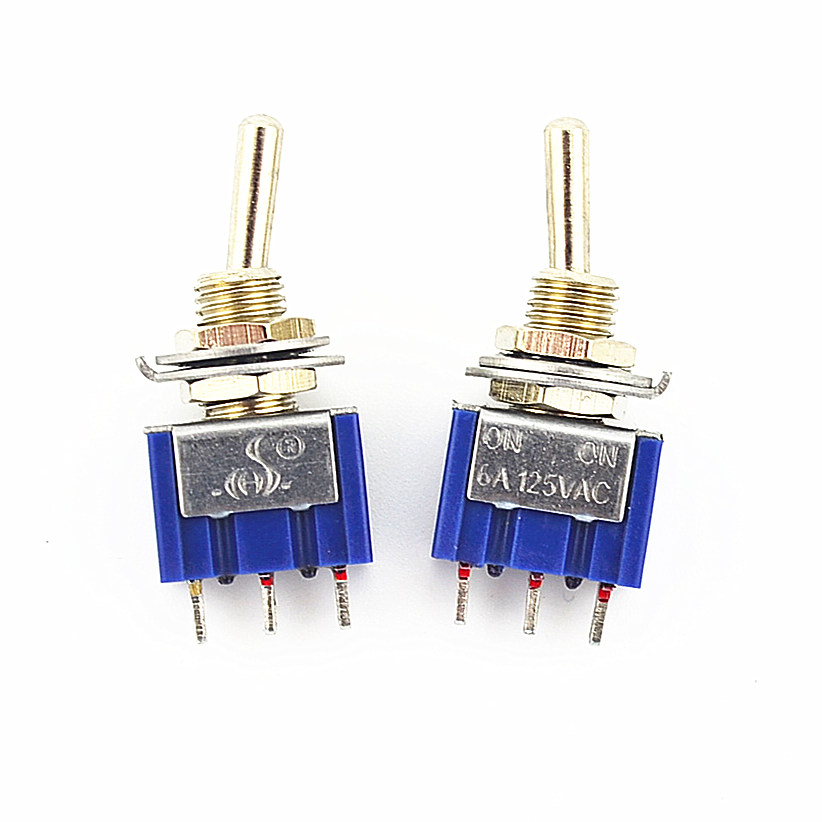


Рисунок 1.7. Внешний вид трехпозиционного тумблера с фиксацией

## 1.5 Переключатель двухпозиционный

Переключатель 2-х позиционный ON-OFF фиксируемый. Корпус и клавиша изготовлены из пластика, контакты - из латуни. Рабочий ток 15 А.



Рисунок 1.8. Внешний вид двухпозиционного переключателя

## 1.6 Батарея «крона»

Батарейка «крона» (также 1604, 6F22, 6R61 (солевая/цинковая); 1604A, 6LF22, 6LR61, MN1604, MX1604 (щелочная/алкалиновая); PP3, E-Block, 9V Brick Battery, AM6, Корунд, 522, 6AM6, CR-9V, ER9V (литиевая)) — типоразмер [батареек](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D1%82%D0%B0%D1%80%D0%B5%D0%B9%D0%BA%D0%B0). Название «крона» происходит от марки выпускавшихся в [СССР](https://ru.wikipedia.org/wiki/%D0%A1%D0%A1%D0%A1%D0%A0) угольно-марганцевых батареек этого типоразмера.

*Технические характеристики*

* Размеры: 48,5 мм × 26,5 мм × 17,5 мм.
* Масса обычно около 53 граммов.
* [Напряжение](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BD%D0%B0%D0%BF%D1%80%D1%8F%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) — 9 [В](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%BB%D1%8C%D1%82).
* Типичная [ёмкость](https://ru.wikipedia.org/w/index.php?title=%D0%81%D0%BC%D0%BA%D0%BE%D1%81%D1%82%D1%8C_%D1%85%D0%B8%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B0_%D1%82%D0%BE%D0%BA%D0%B0&action=edit&redlink=1) щелочной батарейки — 625 [мА·ч](https://ru.wikipedia.org/wiki/%D0%90%D0%BC%D0%BF%D0%B5%D1%80-%D1%87%D0%B0%D1%81).

Батарея типа «крона» имеет ёмкость (по паспорту) 0,5 А·ч.

Существуют [аккумуляторы](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D0%BA%D1%83%D0%BC%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80) данного [форм-фактора](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC-%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80_(%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)). При рабочем расчётном напряжении в 8,4 В, свежезаряженными они могут короткое время давать 11,5 В и выше, что обусловлено особенностями составляющих их [Ni-MH](https://ru.wikipedia.org/wiki/Ni-MH) аккумуляторных элементов.

Аккумуляторные батареи 7Д-0,125 (7Д-0,1, 7Д-0,11) отечественного производства (диаметр 24 мм, высота 62 мм, масса 53 г, номинальная ёмкость 100…125 мА•ч, номинальное напряжение 8,75 В), совместимые по форме контактов с «кроной», набирались из дисковых [никель-кадмиевых аккумуляторов](https://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%BA%D0%B5%D0%BB%D1%8C-%D0%BA%D0%B0%D0%B4%D0%BC%D0%B8%D0%B5%D0%B2%D1%8B%D0%B9_%D0%B0%D0%BA%D0%BA%D1%83%D0%BC%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80) Д-0,125, Д-0,1, Д-0,11 соответственно, были цилиндрической формы, корпус — пластмассовый. Их габариты немного больше стандартной «Кроны», и многие отечественные малогабаритные [радиоприёмники](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE%D0%BF%D1%80%D0%B8%D1%91%D0%BC%D0%BD%D0%B8%D0%BA) изготавливались с учётом этого. Некоторые приёмники (например, [«Селга»](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BB%D0%B3%D0%B0)) продавались в комплекте с таким аккумулятором и зарядным устройством, а карманные приёмники «Мир» 1962 года и «Сокол» 1963 года имели встроенное бестрансформаторное зарядное устройство.

Обычные одноразовые батареи «крона», в отличие от химических элементов питания других типов, допускали ограниченное количество дозарядок, хотя это и не рекомендовалось изготовителем, поэтому во многих журналах и книгах, например, в «Энциклопедии юного техника» 1977 года, публиковались схемы зарядных устройств с током особой формы. Подобные схемы имели некоторую популярность в связи с дефицитностью батарей [2].

## 1.7 Светодиод

Светодиоды для человечества стали одним из наиболее распространенных источников света для промышленных и бытовых нужд. Этот полупроводниковый прибор имеет один электрический переход, он преобразует электроэнергию в энергию видимого светового излучения. Явление открыто Генри Джозефом Раундом в 1907 году. Первые эксперименты были поставлены советским физиком-экспериментатором О.В. Лосевым, которому в 1929 году удалось получить рабочий прототип современного светодиода.

Первые современные светодиоды (СД, СИД, LED) были созданы в начале шестидесятых годов. У них было слабое красное свечение, их применяли в качестве индикаторов включения в самых разных приборах. В 90-х появились синие, желтые, зеленые и белые светодиоды. Их стали выпускать в промышленных масштабах многие компании. Сегодня LED-диоды применяются повсеместно: в светофорах, лампочках, автомобилях и так далее.

*Устройство*

Светодиод представляет полупроводниковый прибор с электронно-дырочным переходом, который создает оптическое излучение при прохождении через него тока в прямом направлении.

Стандартный индикаторный светодиод выполнен из следующих частей;

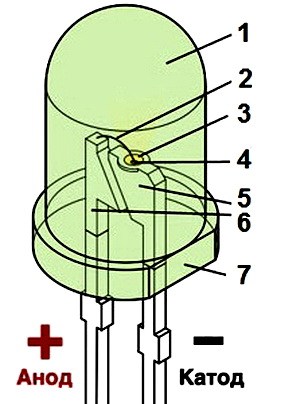


Рисунок 1.9. Схема светодиода

1 — Эпоксидная линза  
2 — Проволочный контакт  
3 — Отражатель  
4 - Полупроводник (Определяет цвет свечения)  
5 и 6 — Электроды  
7 — Плоский срез

В основании светодиода закрепляются катод и анод. Все устройство сверху герметично закрыто линзой. На катоде установлен кристалл. На контактах имеются проводники, которые подсоединены к кристаллу p-n-переходом (проволока соединения для объединения двух проводников с различными типами проводимости). Для создания стабильной работы светодиода применяется теплоотвод, который необходим для осветительных приборов. В индикаторных приборах тепло не имеет решающего значения.

DIP-диоды имеют выводы, которые монтируются в отверстия печатной платы, они при помощи пайки подсоединяются на электрический контакт. Имеются модели с несколькими кристаллами различного цвета в одном корпусе [5].

# 2 Конструирование пульта дистанционного управления роботом

Конструирование пульта дистанционного управления исходит из задач, которые данный пульт будет выполнять. Назначением пульта является управление двумя бесколлекторными двигателями постоянного, двумя коллекторными двигателями постоянного тока и одним сервоприводом. Причем коллекторные двигатели имеют возможность вращения в двух направлениях.

Для реализации поставленной задачи были использованы следующие комплектующие:

– плата Arduino Nano (1 шт.);

– радио модуль NRF24L01 PA LNA (1 шт.);

– адаптер для радио модуля NRF24L01 PA LNA (1 шт.);

– однооборотный потенциометр (4 шт.);

– тумблер трехпозиционный с фиксацией (1 шт.);

– переключатель двухпозиционный (1 шт.);

– батарея «Крона» (1 шт.);

– светодиод (2 шт.);

– рукоятки для тумблеров (4 шт.);

– корпус (1 шт.);

– провода соединительные.

Корпус был изготовлен из подручных материалов (органическое стекло, оцинкованное железо, стальные уголки, дерево).

Все электрические элементы спаяны и соединены между собой электрическими проводами сечением 0,5 мм2 и закреплены на органическом стекле. Рукоятки для тумблеров сделаны из стального прутка. Органическое стекло крепится стальными болтами длиной 60 мм. Тумблер и светодиоды соединены таким образом, что при выборе одного крайнего положения тумблера загорался первый светодиод, а при выборе второго крайнего положения тумблера загорался второй светодиод. Переключатель двухпозиционный разрывает контакт питания платы Arduino от батареи «крона».

Внешний вид пульта дистанционного управления представлен на рисунке 2.1. Электрическая схема пульта представлена на рисунке 2.2.



Рисунок 2.1. Внешний вид пульта дистанционного управления.

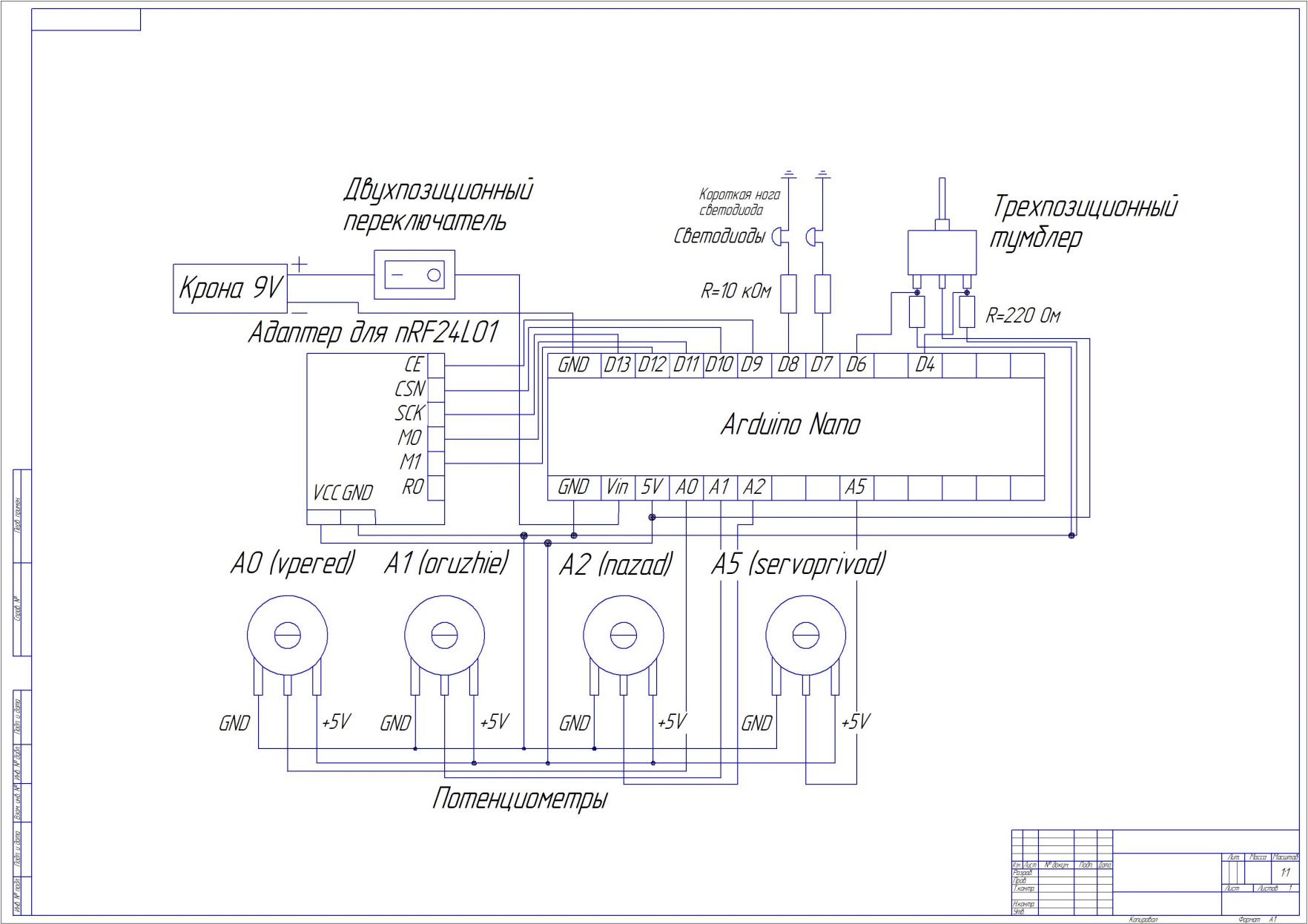


Рисунок 2.2. Электрическая схема пульта дистанционного управления

# 3 Программирование пульта дистанционного управления роботом

## 3.1 Среда разработки Arduino IDE

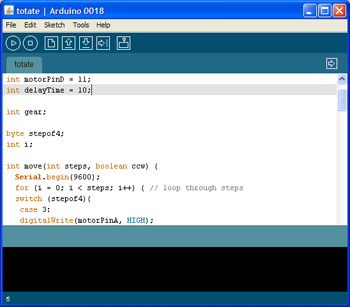


Рисунок 3.1. Внешний вид окна программы

Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino.

Программа, написанная в среде Arduino, называется скетч. Скетч пишется в текстовом редакторе, имеющем инструменты вырезки/вставки, поиска/замены текста. Во время сохранения и экспорта проекта в области сообщений появляются пояснения, также могут отображаться возникшие ошибки. Окно вывода текста(консоль) показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины:

компиляция кода*Verify/Compile*  
Проверка программного кода на ошибки, компиляция.

остановка мониторинга*Stop*  
Остановка мониторинга последовательной шины(Serial monitor) или затемнение других кнопок.

[создание нового скетча](http://arduino.ru/sites/default/files/aruino_environment/new.gif)*New*  
Создание нового скетча.

[открыть файл скетча](http://arduino.ru/sites/default/files/aruino_environment/open.gif)*Open*  
Открытие меню доступа ко всем скетчам в блокноте. Открывается нажатием в текущем окне.

*Примечание*: из-за наличия ошибки в Java данное меню не может прокручиваться; при необходимости открыть скетч из этого списка проследуйте в меню File | Sketchbook.

[сохранение скетча](http://arduino.ru/sites/default/files/aruino_environment/save.gif)*Save*  
Сохранение скетча.

[загрузка скетча в контроллер](http://arduino.ru/sites/default/files/aruino_environment/export.gif)*Upload to I/O Board*  
Компилирует программный код и загружает его в устройство Arduino. Описание загрузки приведено ниже.

[Serial monitor](http://arduino.ru/sites/default/files/aruino_environment/serial_monitor.gif)*Serial Monitor*  
Открытие мониторинга последовательной шины (Serial monitor).

Дополнительные команды сгруппированы в пять меню: File, Edit, Sketch, Tools, Help. Доступность меню определяется работой, выполняемой в данный момент.

*Edit*

* *Copy for Discourse*  
  Копирует в буфер обмена подходящий для размещения на форуме код скетча с выделением синтаксиса.
* *Copy as HTML*

Копирует код скетча в буфер обмена как HTML код, для размещения на веб-страницах.

*Sketch*

* *Verify/Compile*  
  Проверка скетча на ошибки.
* *Import Library*

Добавляет библиотеку в текущий скетч, вставляя директиву #include в код скетча. Подробная информация в описании библиотек ниже (Libraries).

* *Show Sketch Folder*  
  Открывает папку, содержащую файл скетча, на рабочем столе.
* *Add File...*

Добавляет файл в скетч (файл будет скопирован из текущего места расположения). Новый файл появляется в новой закладке в окне скетча. Файл может быть удален из скетча при помощи меню закладок.

*Tools*

* *Auto Format*

Данная опция оптимизирует код, например, выстраивает в одну линию по вертикали открывающую и закрывающую скобки и помещает между ними утверждение.

* *Board*  
  Выбор используемой платформы. Список с описанием платформ приводится ниже.
* Serial Port

Меню содержит список последовательных устройств передачи данных (реальных и виртуальных) на компьютере. Список обновляется автоматически каждый раз при открытии меню Tools.

* *Burn Bootloader*

Пункты данного меню позволяют записать Загрузчик (Bootloader) в микроконтроллер на платформе Arduino. Данное действие не требуется в текущей работе с Arduino, но пригодится, если имеется новый ATmega (без загрузчика). Перед записью рекомендуется проверить правильность выбора платформы из меню. При использовании AVR ISP необходимо выбрать соответствующий программатору порт из меню Serial Port.

*Блокнот (Sketchbook)*

Средой Arduino используется принцип блокнота: стандартное место для хранения программ (скетчей). Скетчи из блокнота открываются через меню File > Sketchbook или кнопкой Open на панели инструментов. При первом запуске программы Arduino автоматически создается директория для блокнота. Расположение блокнота меняется через диалоговое окно Preferences.

*Закладки, Файлы и Компиляция*

Позволяют работать с несколькими файлами скетчей (каждый открывается в отдельной закладке). Файлы кода могут быть стандартными Arduino (без расширения), файлами С (расширение \*.с), файлами С++ (\*.срр) или головными файлами (.h).

*Загрузка скетча в Arduino*

Перед загрузкой скетча требуется задать необходимые параметры в меню Tools > Board и Tools > Serial Port. Платформы описываются далее по тексту. В ОС Mac последовательный порт может обозначаться как dev/tty.usbserial-1B1 (для платы USB) или /dev/tty.USA19QW1b1P1.1 (для платы последовательной шины, подключенной через адаптер Keyspan USB-to-Serial). В ОС Windows порты могут обозначаться как COM1 или COM2 (для платы последовательной шины) или COM4, COM5, COM7 и выше (для платы USB). Определение порта USB производится в поле Последовательной шины USB Диспетчера устройств Windows. В ОС Linux порты могут обозначаться как /dev/ttyUSB0, /dev/ttyUSB1.

После выбора порта и платформы необходимо нажать кнопку загрузки на панели инструментов или выбрать пункт меню File > Upload to I/O Board. Современные платформы Arduino перезагружаются автоматически перед загрузкой. На старых платформах необходимо нажать кнопку перезагрузки. На большинстве плат во время процесса будут мигать светодиоды RX и TX. Среда разработки Arduino выведет сообщение об окончании загрузки или об ошибках.

При загрузке скетча используется Загрузчик (Bootloader) Arduino, небольшая программа, загружаемая в микроконтроллер на плате. Она позволяет загружать программный код без использования дополнительных аппаратных средств. Загрузчик (Bootloader) активен в течении нескольких секунд при перезагрузке платформы и при загрузке любого из скетчей в микроконтроллер. Работа Загрузчика (Bootloader) распознается по миганию светодиода (13 пин) (напр.: при перезагрузке платы).

*Библиотеки*

Библиотеки добавляют дополнительную функциональность скетчам, например, при работе с аппаратной частью или при обработке данных. Для использования библиотеки необходимо выбрать меню Sketch > Import Library. Одна или несколько директив #include будут размещены в начале кода скетча с последующей компиляцией библиотек и вместе со скетчем. Загрузка библиотек требует дополнительного места в памяти Arduino. Неиспользуемые библиотеки можно удалить из скетча убрав директиву #include.

На Arduino.cc имеется [список библиотек](http://out.arduino.ru/?redirect=http%3A%2F%2Farduino.cc%2Fen%2FReference%2FLibraries&baseU=http%3A%2F%2Farduino.ru%2FArduino_environment). Некоторые библиотеки включены в среду разработки Arduino. Другие могут быть загружены с различных ресурсов. Для установки скачанных библиотек необходимо создать директорию «libraries» в папке блокнота и затем распаковать архив. Например, для установки библиотеки DateTime ее файлы должны находится в подпапке /libraries/DateTime папки блокнота [6].

## 3.2 Программа для работы пульта дистанционного управления

Программа для работы пульта дистанционного управления представлена ниже.

//пульт// название программы

#include <SPI.h> // библиотека для работы с шиной SPI

#include "nRF24L01.h" // библиотека радио модуля

#include "RF24.h" // ещё библиотека радио модуля

RF24 radio(9, 10); // "создать" модуль на пинах 9 и 10

const byte PIN\_TUMBLER\_VPERED = 6; // к 6-ому пину подключен вывод тумблера, отвечающий за движение вперед

const byte PIN\_TUMBLER\_NAZAD = 4; // к 4-ому пину подключен вывод тумблера, отвечающий за движение назад

int led\_vpered = 7; // к 7-ому пину подключен светодиод, загорающийся при переключении тумблера в положение «вперед»

int led\_nazad = 8; // к 8-ому пину подключен светодиод, загорающийся при переключении тумблера в положение «назад»

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

byte transmit\_data[6]; // массив, хранящий передаваемые данные в шести ячейках

void setup() {

Serial.begin(9600); // открываем порт для связи с компьютером

radio.begin(); // активировать радио 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 сигнал

radio.setPayloadSize(32); //размер пакета, в байтах

radio.openWritingPipe(address[0]); //мы - труба 0, открываем канал для передачи 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

// Далее описываем тумблер и светодиоды

pinMode(led\_vpered, OUTPUT); // пин, на который подключен светодиод считаем выходом

pinMode(led\_nazad, OUTPUT); // пин, на который подключен светодиод считаем выходом

pinMode(PIN\_TUMBLER\_VPERED, INPUT); // пин, к которому подключен тумблер считаем выходом

pinMode(PIN\_TUMBLER\_NAZAD, INPUT); // пин, к которому подключен тумблер считаем выходом

}

void loop() {

byte gotByte;

// Блок включения светодиодов. Начало

if (digitalRead(PIN\_TUMBLER\_VPERED) == HIGH) { // если на пине тумблера есть высокий сигнал,

digitalWrite(led\_vpered, HIGH); // то на светодиод также подается напряжение и он включется

}

else { // если иначе

digitalWrite(led\_vpered, LOW); // то подается низкий сигнал и светодиод не горит

}

if (digitalRead(PIN\_TUMBLER\_NAZAD) == HIGH) { // если на пине тумблера есть высокий сигнал,

digitalWrite(led\_nazad, HIGH); //то на светодиод также подается напряжение и он включется

}

else { //если иначе

digitalWrite(led\_nazad, LOW); //то подается низкий сигнал и светодиод не горит

}

// Блок включения светодиодов. Конец

// Выводим данные в монитор порта

int TUMBLER\_VPERED = digitalRead(PIN\_TUMBLER\_VPERED); // переменной присваиваем данные с пина TUMBLER\_VPERED

Serial.print("TUMBLER\_VPERED: "); // выводим надпись "TUMBLER\_VPERED: " в монитор порта

Serial.print(TUMBLER\_VPERED); // выводим значения переменной в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

int TUMBLER\_NAZAD = digitalRead(PIN\_TUMBLER\_NAZAD); // переменной присваиваем данные с пина TUMBLER\_NAZAD

Serial.print("TUMBLER\_NAZAD: "); // выводим надпись "TUMBLER\_NAZAD: " в монитор порта

Serial.print(TUMBLER\_NAZAD); // выводим значения переменной в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

Serial.print("motor\_vpered:"); // выводим надпись "motor\_vpered:" в монитор порта

Serial.print(analogRead(0)); // выводим значения переменной с нулевого пина в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

Serial.print("motor\_oruzhie:"); // выводим надпись "motor\_oruzhie:" в монитор порта

Serial.print(analogRead(1)); // выводим значения переменной с 1-го пина в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

Serial.print("motor\_nazad:"); // выводим надпись "motor\_nazad:" в монитор порта

Serial.print(analogRead(2)); // выводим значения переменной со 2-го пина в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

Serial.print("servoprivod:"); // выводим надпись "servoprivod:" в монитор порта

Serial.print(analogRead(5)); // выводим значения с 5-го пина в монитор порта

Serial.print(" "); // выводим пробел в монитор порта, чтобы был отступ между столбцами

Serial.println(); // новую серию данных выводим с новой строки

transmit\_data[0] = map(analogRead(0), 0, 1023, 0, 180); // получить значение переменной записываем на нулевое место в массиве

transmit\_data[1] = map(analogRead(1), 0, 1023, 0, 180); // получить значение переменной записываем на первое место в массиве

transmit\_data[2] = map(analogRead(1), 0, 1023, 0, 180); // получить значение переменной записываем на второе место в массиве

transmit\_data[3] = map(analogRead(1), 0, 1023, 0, 180); // получить значение переменной записываем на третье место в массиве

transmit\_data[4] = TUMBLER\_VPERED; // получить значение переменной записываем на первое место в массиве

transmit\_data[5] = TUMBLER\_NAZAD; // получить значение переменной записываем на второе место в массиве

radio.write(&transmit\_data, sizeof(transmit\_data)); // отправить по радио массив данных

}

Загружаем написанную программу в плату Arduino Nano. Для этого подключаем пульт дистанционного управления к компьютеру через кабель USB-миниUSB. Для этого на передней панели пульта предусмотрено отверстие для подключения пульта к компьютеру [1].

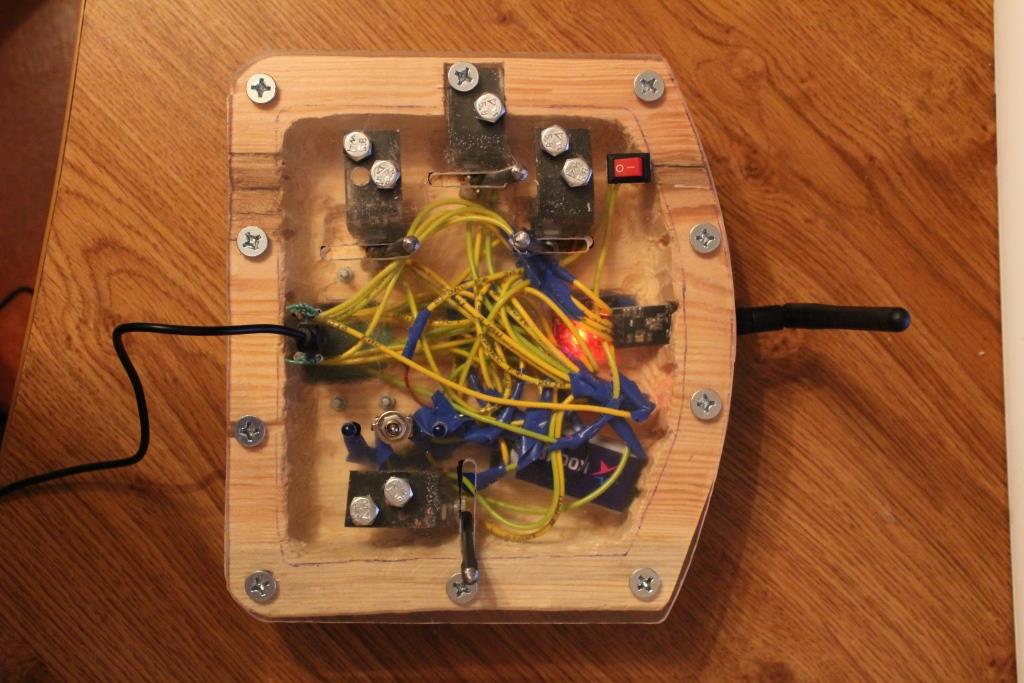


Рисунок 3.2. Подключение пульта к компьютеру

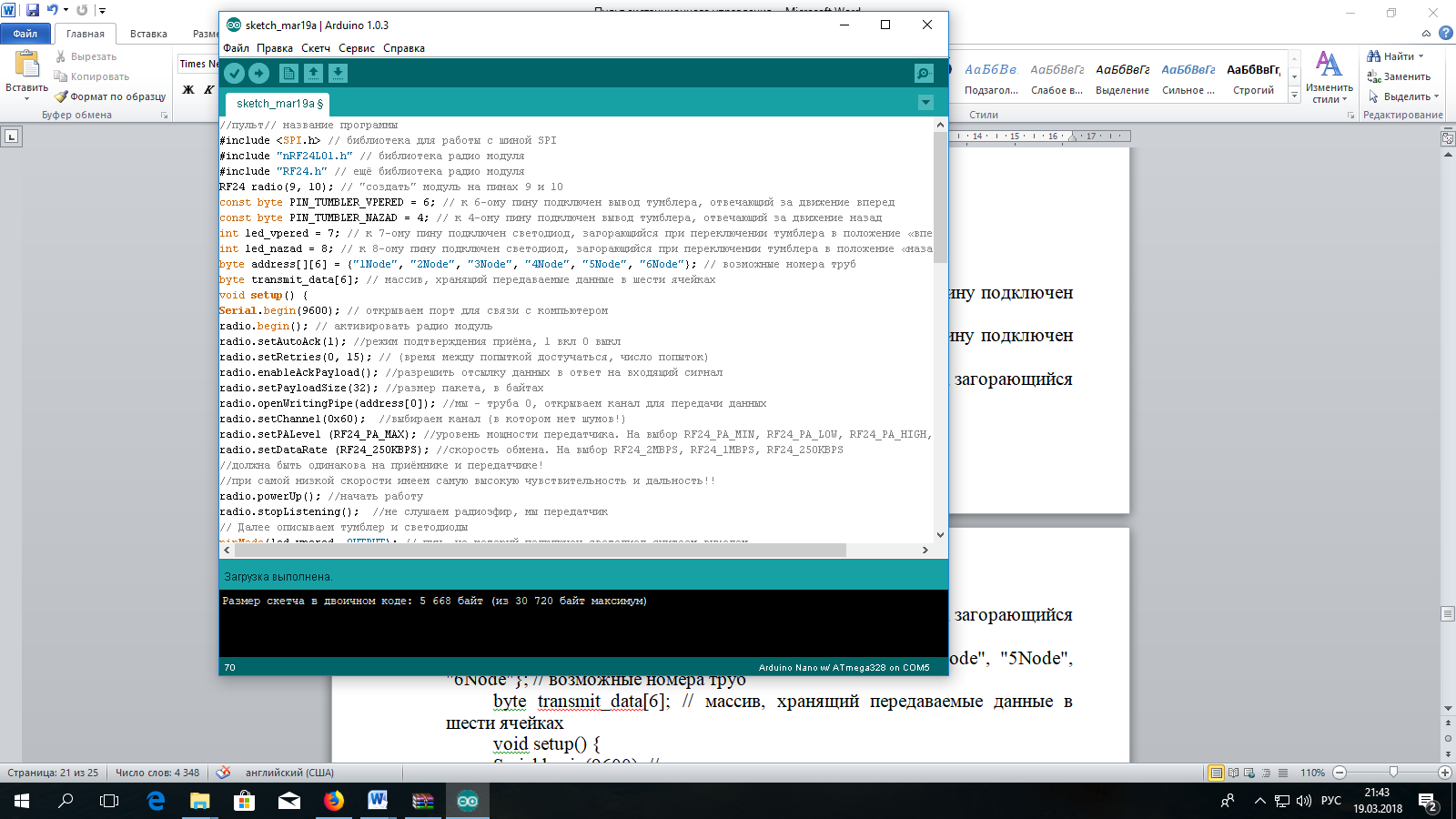


Рисунок 3.3. Загрузка скетча в плату через Arduino IDE 1.0.3

Все данные из массива плата передает в монитор порта и их можно просмотреть.

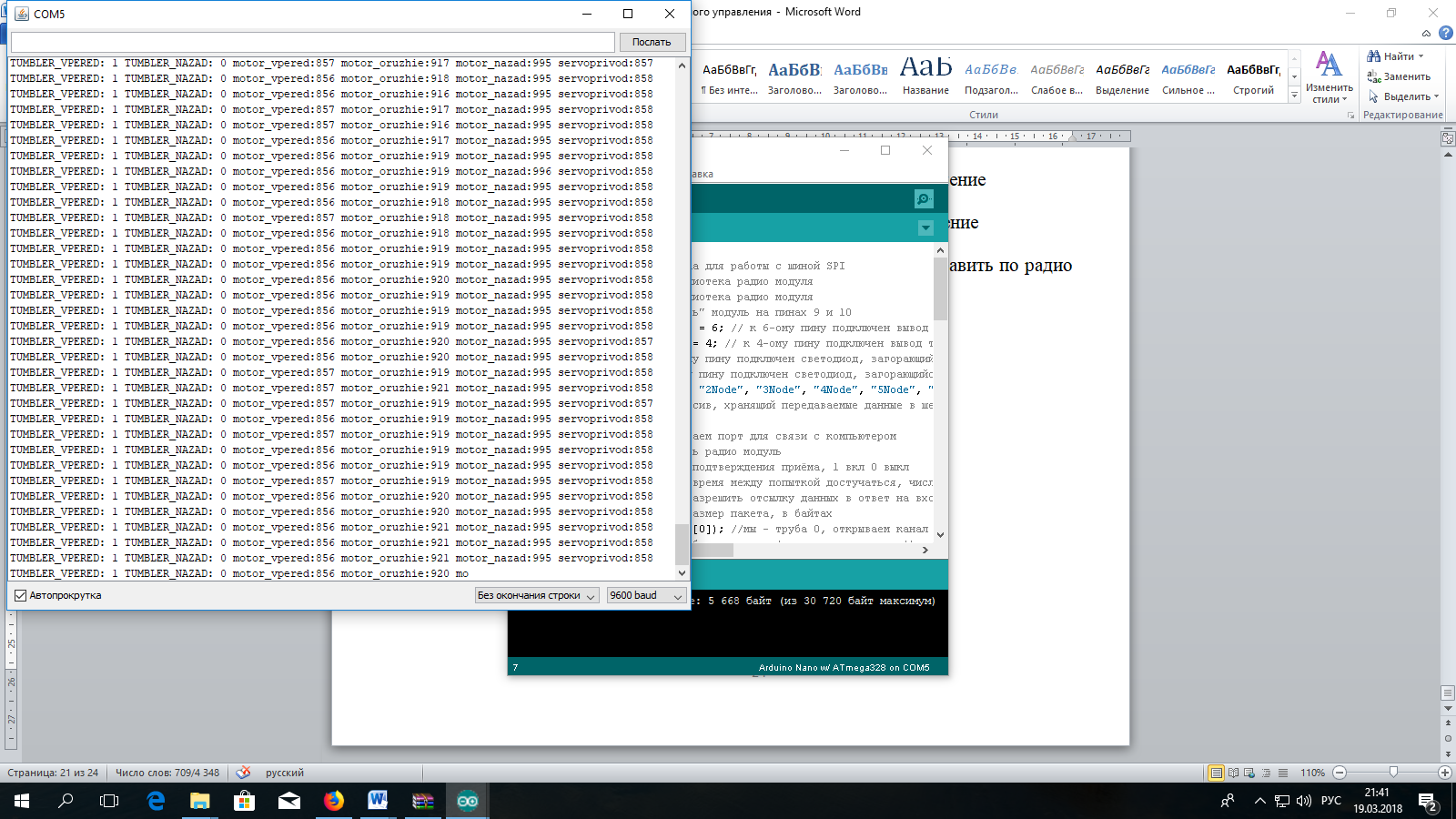


Рисунок 3.4. Данные массива в мониторе порта

# Заключение

В результате проведенной работы был создан пульт дистанционного управления роботом. С помощью которого можно управлять множеством роботизированных систем, работающих на частоте 2,4 ГГц.

В ходе работы были решены все поставленные задачи:

**–** изучить комплектующие для пульта;

– сделать электрическую схему пульта;

– собрать пульт из подобранных комплектующих;

– написать программу для работы пульта;

– загрузить программный код в плату Arduino Nano;

– проверка работоспособности пульта.

В последующем для демонстрации работы данного пульта будет разработан робот. Данный робот найдет свое применение в соревновательной робототехнике.

# Список использованной литературы

* 1. Соммер У. Программирование микроконтроллерных плат Arduino/Freeduino. – СПб.: БХВ-Петербург, 2012. – 256 с. ил – (Электроника)
  2. Батарея «Крона» [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Батарея\_«Крона»](https://ru.wikipedia.org/wiki/Батарея_)]
  3. Потенциометры. Виды и устройство. Работа и особенности [Электронный ресурс]. – Режим доступа: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/potentsiometry/>
  4. Радио модуль 1100м. 2.4G NRF24L01+PA+LNA для Arduino [Электронный ресурс]. – Режим доступа: http://iarduino.ru/shop/Expansion-payments/radio-modul-rf-2400-wireless-module-2-4g-1000m.html
  5. Светодиоды. Устройство. Виды. Работа. Применение [Электронный ресурс]. – Режим доступа: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/svetodiody/>
  6. Среда разработки Arduino [Электронный ресурс]. – Режим доступа: http://arduino.ru/Arduino\_environment
  7. Тумблеры КЭП 51 серия [Электронный ресурс]. – Режим доступа: http://keptm.ru/article/svodnaya-tablitsa-tumblerov-kep
  8. Arduino Nano [Электронный ресурс]. – Режим доступа: http://arduino.ru/hardware/arduinoboardnano