

**Муниципальное бюджетное общеобразовательное учреждение
«Лицей №17»**

Создание робота на Arduino

Выполнил:

Кундалев Алексей, обучающийся 11 класса,

Руководитель:

Ширяева Мария Анатольевна,

учитель информатики и ИКТ

Березовский городской округ

2019

Содержание

Введение	3
1. Робототехника. Принципы классификации.....	4
2. Основы робототехники на платформе Arduino	6
3. Сборка робота на платформе Arduino	11
Заключение	15
Список использованной литературы.....	16
Приложение 1	17
Программный код микроконтроллера	17

Введение

Роботы, микроконтроллеры, робототехника...

Роботы сегодня вошли в нашу жизнь в разных областях. Они летают в космос, исследуют другие планеты; помогают в военных целях — разминируют бомбы и разведывают обстановку с воздуха. В промышленности многие области уже немыслимы без роботов: они собирают автомобили, помогают находить новые лекарства. Для овладения наукой робототехники нужно быть специалистом во многих областях: программирование, механика, физика, электроника и т.д.

Цель работы: создать действующую модель робота на основе платы Arduino.

Для достижения поставленной цели были сформулированы следующие **задачи:**

- изучить литературу по данной теме;
- рассмотреть аппаратно-вычислительную платформу Arduino для создания устройства;
- собрать электронную систему на основе аппаратно-вычислительной платформы Arduino.

Предметом исследования: аппаратная вычислительная платформа Arduino.

Методы исследования: теоретический анализ источников, формализация, обобщение, моделирование, эксперимент, описание.

1. Робототехника. Принципы классификации

Робототехника — это достаточно молодая отрасль прикладных наук, которая занимается разработкой автоматизированных технических систем. Робототехника тесно связана с механикой, электроникой и программированием.

Само слово «робот» придумали ещё в 1920 году чешский писатель Карел Чапек и его брат Йозеф, впервые оно встретилось в их научно-фантастической пьесе «Россумские универсальные роботы», которая получила большой зрительский успех.

Там оно обозначало искусственно созданных людей, труд которых использовался вместо человеческого на тяжёлых и опасных производствах («robota» в переводе с чешского языка – каторга). Хотя в этом произведении роботы и изготавливались на фабриках по выращиванию органических тканей, само понятие впоследствии стало популярным именно в отношении механических устройств.

Роботы отличаются от простых механических автоматов. В отличие от автоматов робот не просто следует заранее вложенному в него алгоритму, а способен воспринимать внешние сигналы и в соответствии с ними адаптировать свои действия в изменяющейся ситуации.

Понятие «робототехника» впервые в своём рассказе «Лжец» использовал писатель-фантаст Айзек Азимов в 1941 году.

Уже давно прошли те времена, когда роботов можно было встретить только в фантастических фильмах. Сегодня роботы становятся частью обыденной жизни.

Робототехника, как отрасль наук, охватывает различные сферы, поэтому существует некая классификация.

По отрасли применения робототехника подразделяется на **промышленную, бытовую и экстремальную.**

Промышленная робототехника включает в себя:

- производственную робототехнику (например, автосборочное производство);
- строительную робототехнику;
- транспортную робототехнику (например, авиационная, автомобильная робототехника).

Экстремальная робототехника включает в себя:

- космическую;
- подводную;
- военную робототехнику.

По уровню применения робототехника подразделяется на **игровую, прикладную, исследовательскую и обучающую.**

В рамках обучающей робототехники используются робототехнические комплекты для детских, учебных и досуговых центров на базе Huna, Lego, Fishertechnik, Arduino. Arduino на сегодняшний день является очень популярной платформой для радиолюбителей и для прототипирования будущих серийных устройств.

2. Основы робототехники на платформе Arduino

Arduino — торговая марка аппаратно-программных средств для построения простых систем автоматики и робототехники, ориентированная на непрофессиональных пользователей.

Программная часть Arduino состоит из бесплатной программной оболочки (IDE) для написания программ, их компиляции и программирования аппаратуры.

Аппаратная часть Arduino представляет собой набор смонтированных печатных плат, продающихся как официальным производителем, так и сторонними производителями.

Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino.

Arduino может использоваться как для создания автономных объектов автоматики, так и подключаться к программному обеспечению на компьютере через стандартные проводные и беспроводные интерфейсы.

Языком программирования Arduino является стандартный C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающими новичкам написание первой работающей программы.

Программы, написанные программистом Arduino, называются *наброски* (или иногда *скетчи* — варваризм от англ. *sketch*) и сохраняются в файлах с расширением `ino`. Эти файлы перед компиляцией обрабатываются препроцессором Arduino. Также существует возможность создавать и подключать к проекту стандартные файлы C++.

Обязательную в C++ функцию `main()` препроцессор Arduino создает сам, вставляя туда необходимые «черновые» действия.

Программист должен написать две обязательные для Arduino функции **setup()** и **loop()**. Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле.

В текст своей программы (скетча) программист не обязан вставлять заголовочные файлы используемых стандартных библиотек. Эти заголовочные файлы добавит препроцессор Arduino в соответствии с конфигурацией проекта. Однако пользовательские библиотеки нужно указывать.

Менеджер проекта **Arduino IDE** имеет нестандартный механизм добавления библиотек. Библиотеки в виде исходных текстов на стандартном C++ добавляются в специальную папку в рабочем каталоге IDE. При этом название библиотеки добавляется в список библиотек в меню IDE. Программист отмечает нужные библиотеки и они вносятся в список компиляции.

Arduino IDE не предлагает никаких настроек компилятора и минимизирует другие настройки, что упрощает начало работы для новичков и уменьшает риск возникновения проблем.

Простейшая Arduino-программа состоит из двух функций:

- **setup()**: функция вызывается однократно при старте микроконтроллера;
- **loop()**: функция вызывается после setup () в бесконечном цикле все время работы микроконтроллера.

Все используемые в примере функции являются библиотечными. В комплекте Arduino IDE имеется множество примеров программ. Существует перевод документации по Arduino на русский язык. Так как платформа очень популярна, появилось огромное количество китайских копий, которые отличаются очень низкой ценой.

Любая робототехническая система, не зависимо от своей сложности исполнения, является «умным» устройством способным выполнять разного рода задачи, следуя определённому алгоритму действий– программе, написанной программистом.

Так же любая такая система имеет на борту электронно-вычислительную аппаратуру, которая контролирует работу всех узлов техники в соответствии с программой, записанной в памяти устройства. В качестве таких устройств используют специальные компьютеры, и микроконтроллеры.

Построение роботов с использованием любой технологии подразумевает изучение принципов работы специальных микросхем, которые называются микроконтроллерами. Они предназначены для управления электронными устройствами и представляют собой однокристалльный компьютер, способный выполнять простые задачи. Микроконтроллер – компьютер на одной микросхеме, предназначен для управления различными электронными устройствами и осуществления взаимодействия между ними в соответствии с заложенной программой. В отличие от микропроцессоров, используемых в персональных компьютерах, микроконтроллеры содержат встроенные устройства(см. рис. 1).

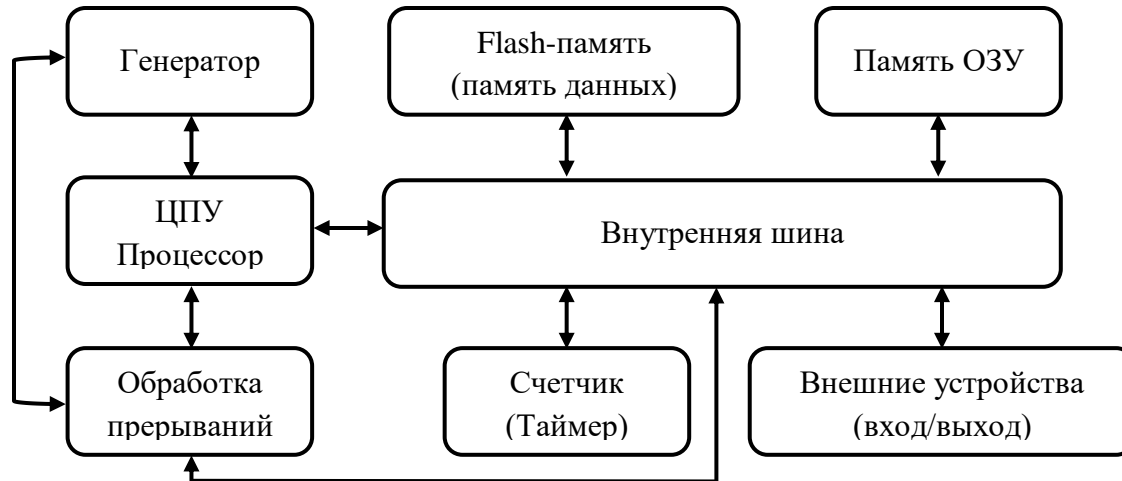


Рисунок 1. Структура микроконтроллера

Всё управление робототехническими системами сводится к подаче электрических импульсов на нужные выводы в соответствии с программным кодом микроконтроллера.

Первый прототип Arduino был разработан в 2005 году программистом Массимо Банци. На сегодняшний день платформа Arduino представлена не одной платой, а целым их семейством. Такой подход позволяет собирать всевозможные электронные устройства, работающие как автономно, так и в связке с компьютером. Платы Arduino представляют собой наборы, состоящие из готового электронного блока и программного обеспечения. Электронный блок — это печатная плата с установленным микроконтроллером. Фактически электронный блок Arduino является аналогом материнской платы компьютера. На нем имеются разъемы для подключения внешних устройств, а также разъем для связи с компьютером, по которому осуществляется программирование. Самой популярной и наиболее универсальной платформой семейства является плата ArduinoUno. Она выполнена на базе процессора с тактовой частотой 16 МГц, обладает памятью 32кБ, два из которых выделено под загрузчик, позволяющий прошивать Arduino с обычного компьютера через USB. Также имеется 2 Кбайта SRAM-памяти, которые используются для хранения временных данных (это оперативная память платформы) и 1Кбайт EEPROM-памяти для долговременного хранения данных (аналог жёсткого диска). На платформе расположены 14 контактов, которые могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от программы. Некоторые контакты обладают дополнительными ролями. Например, Serial 0-й и 1-й — используются для приёма и передачи данных по USB; LED 13-й — к этому контакту подключен встроенный в плату светодиод. Также имеется 6 контактов аналогового ввода и входной контакт Reset для сброса (см. рис. 2).

Отличительной особенностью Arduino является наличие плат расширения — «шилдов». Это дополнительные платы, которые ставятся подобно «слоям бутерброда» поверх Arduino, чтобы дать ему новые возможности. Shield подключаются к Arduino с помощью имеющихся на них штыревых разъемов.

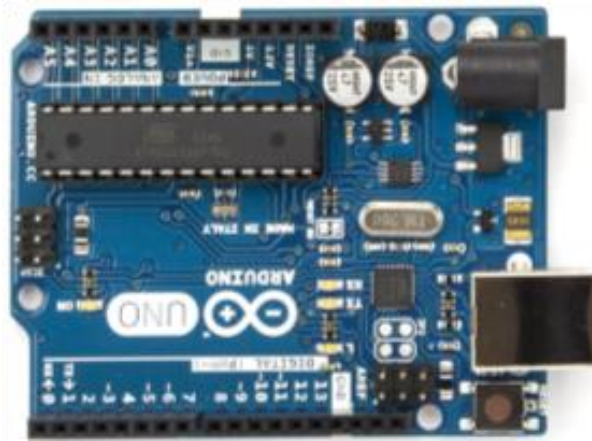


Рисунок 2. Плата ArduinoUno

Разработка приложений на базе плат Arduino осуществляется в специальной среде программирования Arduino IDE. Среда предназначена для написания и загрузки собственных программ в память микроконтроллера. Среда разработки Arduino состоит из редактора программного кода, области сообщений, окна вывода текста, панели инструментов и панели меню (см. рис.3).

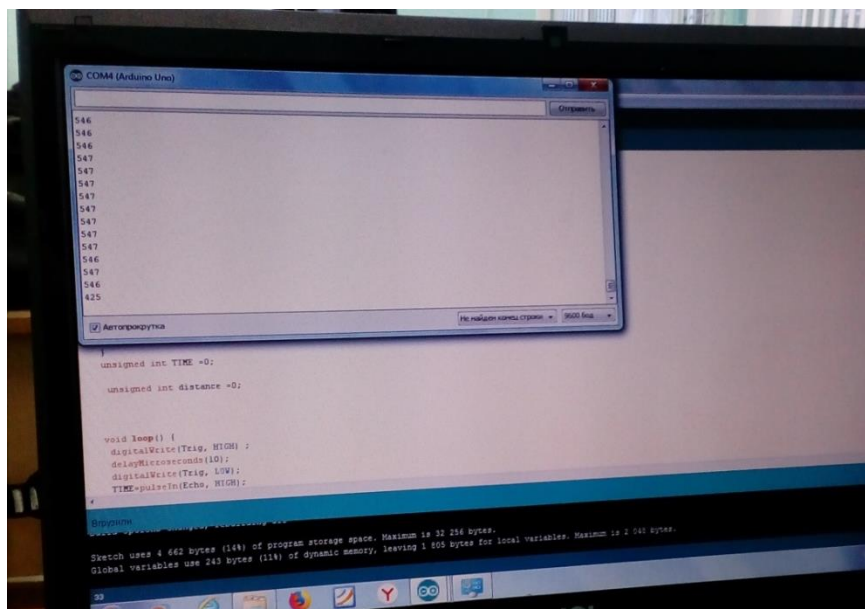


Рисунок 3. Среда разработки Arduino

3. Сборка робота на платформе Arduino

Рассмотрим практическую часть проекта — создание робота. Для этого использовались платы ArduinoNano, ArduinoMicro, отдельный радио модули, двигатели с редукторами, модуль аналогового джойстика.

Основная идея заключается в следующем: робот имеет дистанционное радиоуправление. Имеет манипулятор с «захватом», способен выполнять работу по взаимодействию с окружающей средой, способен передвигаться.

Процесс построения модели робота можно разделить на несколько этапов.

Этап I. Сборка платформы. Вначале необходимо собрать основу робота — подвижную платформу. (см. рис. 4).

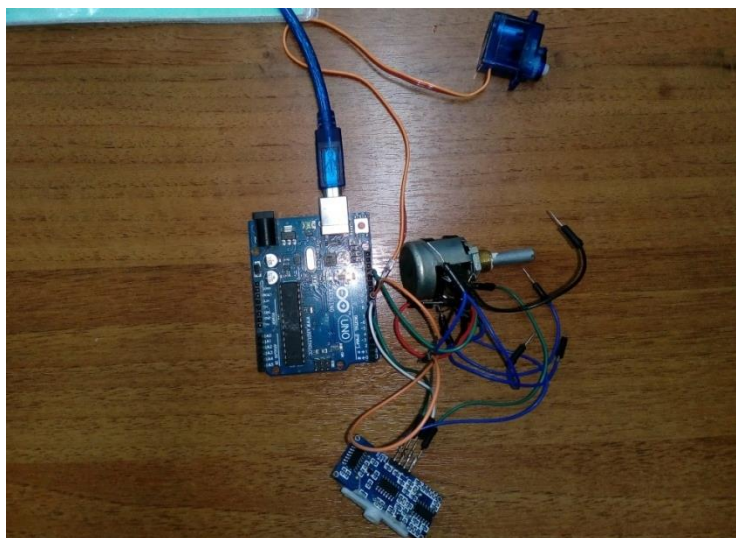


Рисунок 4. Платформа для робота

Этап II. Установка платы Arduino и подключение моторов, и сервоприводов (см. рис.5).

Этап III. Настройка оборудования. На заключительном этапе сборки необходимо добиться слаженной и бесперебойной работы всех систем робота.

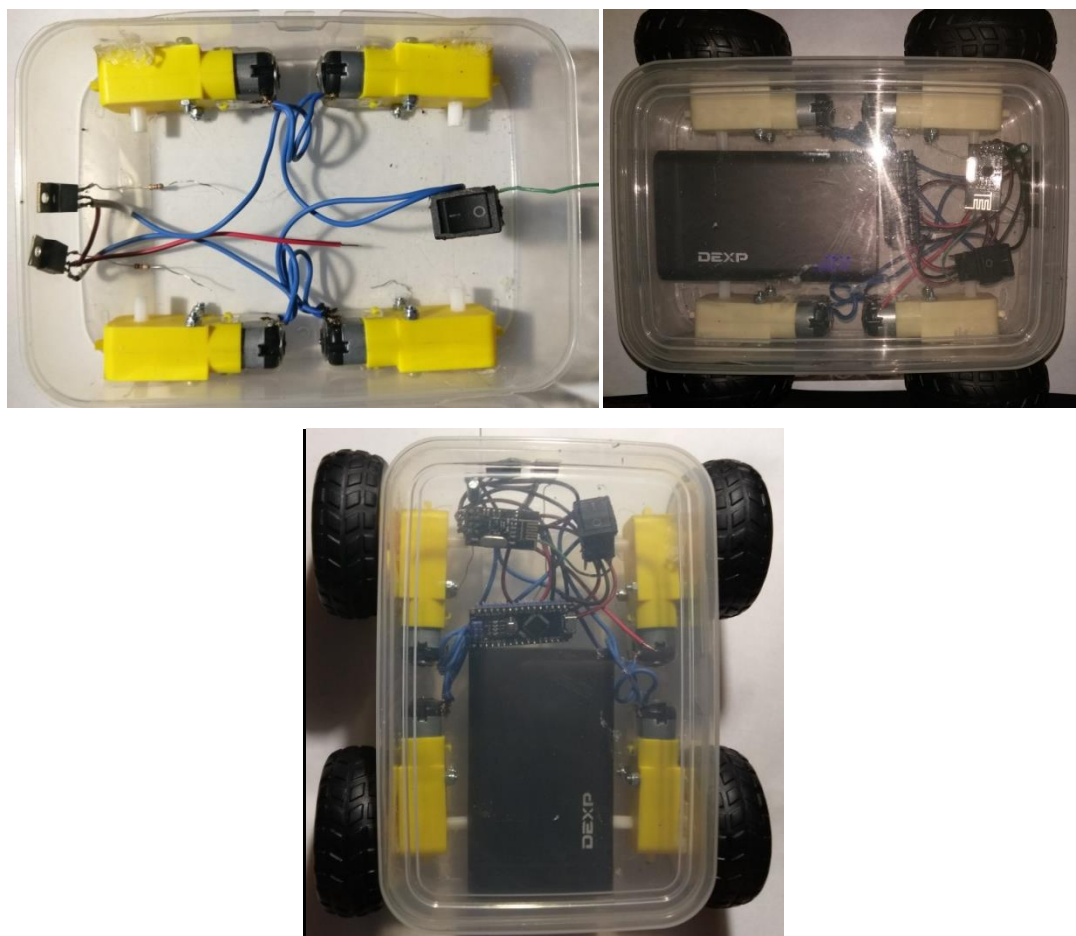


Рисунок 5. Установка моторов и приводов

Далее необходимо разработать алгоритм управления движением робота.

Происходит опрос состояния джойстика – при изменении положения джойстика, изменяется его электрическое сопротивление, изменения регистрируются аналогово-цифровым преобразователем (АЦП). В зависимости от регистрируемого изменения формируются команды, которые, посредством радиосвязи передаются на принимающее устройство робота (см. рис. 6). После получения роботом радиосообщения, принятые данные сравниваются с ключами команд, прописанными в коде. При совпадении данных, подаётся сигнал либо на включение пар двигателей через драйвер, либо на поворот вала одного из трех сервоприводов, приводящих в движение манипулятор посредством широтно-импульсной модуляции (ШИМ-сигнала) (см. прилож. 1).



Рисунок 6. Алгоритмы работы датчиков

Этап III. Установка и подключение датчиков.

Этап IV. Балансировка. На заключительном этапе сборки необходимо закрепить провода на платформе, чтобы они не мешали движению робота (см. рис. 7).

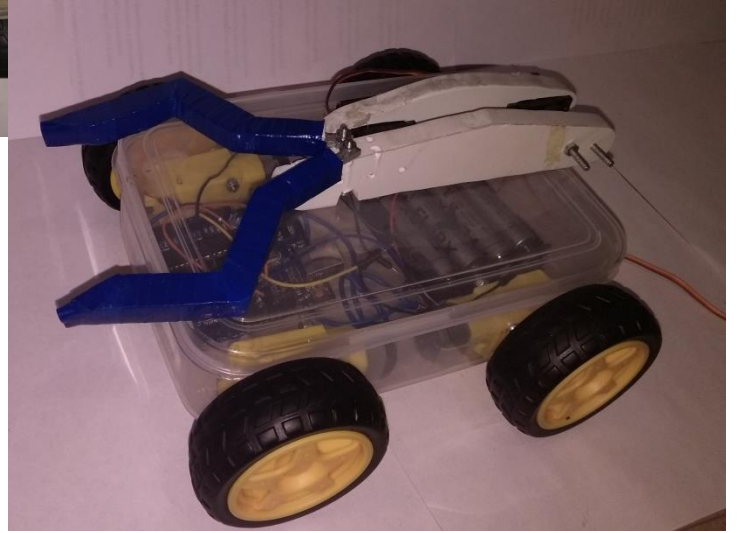
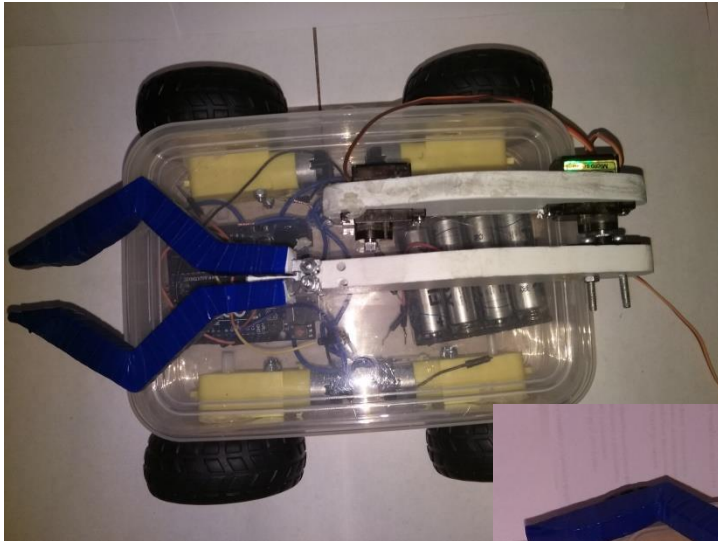


Рисунок 7. Итоговый образец робота

Заключение

В результате использования различных источников информации, я узнал о программно-вычислительной платформе Arduino и о сферах её применения, изучил язык программирования C++.

В данном исследовании были рассмотрены основы проектирования роботов на базе платформ семейства Arduino, а также построена действующая модель робота. В ходе тестирования были выявлены и устранены некоторые недостатки как технической части, так и алгоритма движения. Полученная модель способна корректно воспринимать поступающую информацию о местонахождении робота и, анализируя ситуацию, принимать решение о дальнейшем движении.

Работа по изучению и применению на практике полученных результатов данного исследования будет продолжена. В дальнейшем планируется расширить полученные знания.

/

Список использованной литературы

1. Arduino для начинающих [Электронный ресурс] / Сайт All-Arduino – режим доступа: <https://all-arduino.ru/> (Дата обращения 15.10.2018г.)
2. Блум, Д. Изучаем Arduino [Текст] /. Д.Блум — Справочник для начинающих / СПб.: БХВ-Петербург, 2015. – 105 с.
3. Исследовательские работы. Информатика [Электронный ресурс] / Сайт Обучонок - режим доступа: <http://obuchonok.ru/node/1799>(Дата обращения 22.10.2018г.)
4. Современные языки программирования микроконтроллеров и встраиваемых систем [Электронный ресурс] / Сайт Digitrobe: цифровая электроника, вычислительная техника, встраиваемые системы - режим доступа:<http://digitrobe.ru/theory/programming/> (Дата обращения 12.11.2018г.)
5. Страковский, Д. А. Создание робота-гонщика на платформе Arduino [Электронный ресурс] / Страковский Д. А., Симаков Е. Е. // Юный ученый. — 2016. — №3. — С. 120-124. URL: <http://yun.moluch.ru/archive/6/412/> (дата обращения: 10.12.2018)
6. Теория Arduino [Электронный ресурс] / Сайт учителя информатики Сиденко Андрея Григорьевича – режим доступа: <http://agsidenko.ru/category/arduino/> (Дата обращения 22.10.2018г.)

Программный код микроконтроллера

1. Программный код работы передатчика

```
#include <SPI.h>
#include "RF24.h"
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;

int msg[1];
int Pot_1 = A5;
int Pot_2 = A7;
int Pot_3 = A6;
int Pot_4 = A4;
int val_1;
int val_2;
int val_3;
int val_4;

void setup(void) {
  radio.begin();
  radio.openWritingPipe(pipe);
}

void loop() {
  if (analogRead(Pot_1) < 255) {
    val_1 = 1;}
  else{
    if(analogRead(Pot_1) < 768){
      val_1 = 2;}
    }
  msg[0] = val_1;
  radio.write(msg, 1);

  if (analogRead(Pot_2) < 255) {
    val_2 = 3;}
  else{
    if(analogRead(Pot_2) < 768){
      val_2 = 4;}
      msg[0] = val_1;
    }
  radio.write(msg, 1);

  if (analogRead(Pot_3) < 255) {
    val_1 = 5;}
  else{
```

```

        if(analogRead(Pot_3) < 768){
            val_1 = 6;}
    }
    msg[0] = val_1;
    radio.write(msg, 1);

    if (analogRead(Pot_4) < 255){
    val_2 = 7;}
    else{
        if(analogRead(Pot_4) < 768){
            val_2 = 8;}
        }
        msg[0] = val_1;
        radio.write(msg, 1);

    }

```

2. Программный код работы приемника

```

#include <Servo.h>
#include <SPI.h>
#include "RF24.h"
Servo servo1;
Servo servo2;
Servo servo3;

int pos1;
int pos2;
int pos3;

int dvig_p = 8;
int dvig_l = 7;

RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;
int msg[1];

void setup(){
    pinMode(dvig_p, OUTPUT);
    pinMode(dvig_l, OUTPUT);

    servo1.attach(6);
    servo2.attach(3);
    servo3.attach(5);
    delay(50);
    radio.begin();
    radio.openReadingPipe(1,pipe);
    radio.startListening();

}

void loop() {

```

```

if (radio.available())
{radio.read(msg, 1);

  if(msg[0] == 1){digitalWrite(dvig_p, HIGH);}
  else{digitalWrite(dvig_p, LOW);}

  if(msg[0] == 3){digitalWrite(dvig_l, HIGH);}
  else{digitalWrite(dvig_l, LOW);}

  if((msg[0] == 2) && ((pos1 + 1) >= 180)){
    pos1 = pos1 + 1;}

  if((msg[0] == 4) && ((pos1 - 1) <= 0)){
    pos1 = pos1 - 1;}

    servo1.write(pos1);

    if((msg[0] == 2) && ((pos2 + 1) >= 180)){
    pos2 = pos2 + 1;}

  if((msg[0] == 4) && ((pos2 - 1) <= 0)){
    pos2 = pos2 - 1;}

    servo2.write(pos2);

    if((msg[0] == 2) && ((pos3 + 1) >= 180)){
    pos3 = pos3 + 1;}

  if((msg[0] == 4) && ((pos3 - 1) <= 0)){
    pos3 = pos3 - 1;}

    servo3.write(pos3);

}

}

```