

## Функциональные характеристики RED CODE 3

«RED CODE 3» — это программное обеспечение, устанавливаемое на персональный компьютер, работающий на операционных системах:

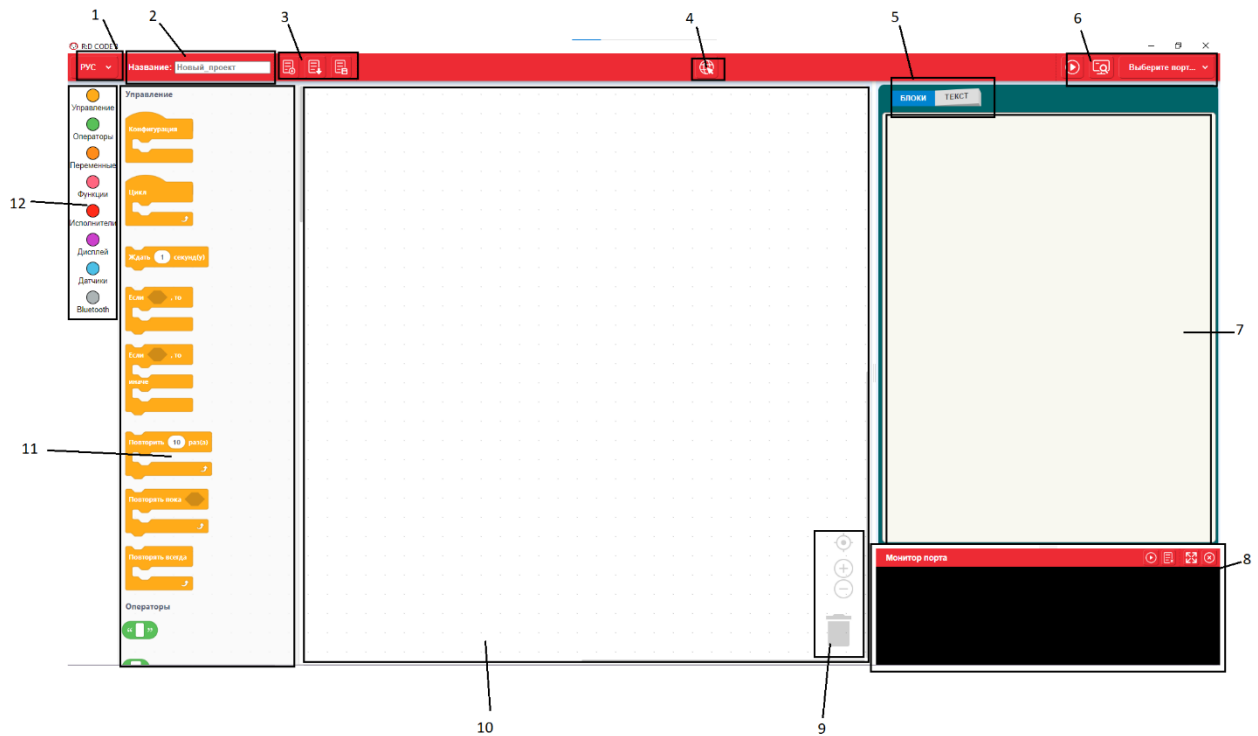
- Windows 7/ 8/ 8.1/10
- Astra Linux 1.7
- РЕД ОС 7.3

Представляет собой визуально-событийную среду программирования, с возможностью составления программного кода с помощью блоков (расширенный набор), которые описывают действия электронных устройств и полностью соответствуют всем возможностям построения логики и математических операций на контроллере, входящих в состав расширенного робототехнического набора R:ED X <https://store.r-ed.world/sets#redx>. Также с помощью данной программы можно проводить прошивку контроллера, входящего в состав расширенного робототехнического набора R:ED X. Приложение позволяет сохранять и загружать составленные пользователем программы для контроллера из набора R:ED X и редактировать данные программы с помощью встроенного текстового редактора.

Минимальные требования для ПК:

- Место на диске: 1 Гб
- Оперативная память: 2Гб (РЕД ОС или Astra Linux), 4Гб (Windows)
- USB интерфейс
- Подключение к Интернету

# Основной пользовательский интерфейс

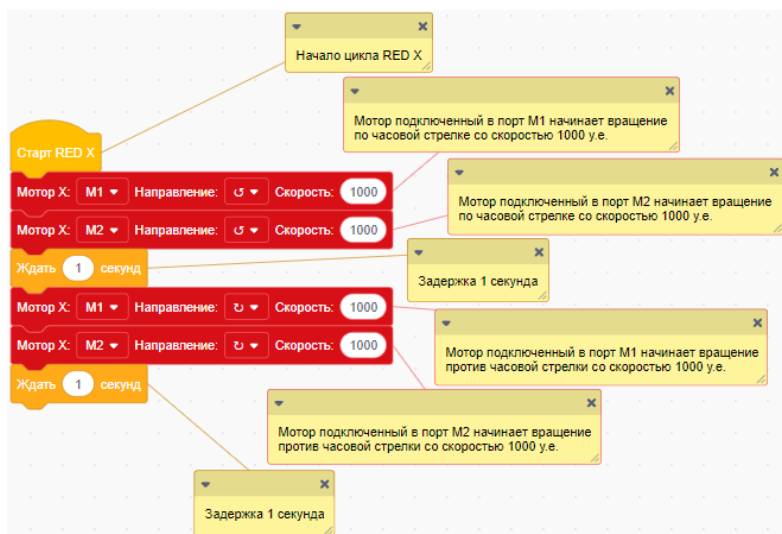


Список элементов основного окна программирования:

1. Выбор языка интерфейса
2. Название проекта
3. Меню работы с файлами проекта
4. Переход на основной сайт R:ED
5. Меню работы со сгенерированным кодом с возможностью его сохранения и загрузки
6. Меню управления сборкой проекта и подключения к контроллеру
7. Поле для работы с кодом
8. Меню монитора порта подключения к контроллеру
9. Меню управления поля программирования
10. Поле блоков программирования
11. Меню блоков программирования
12. Меню выбора группы блоков

## Пример кода - Мотор X

### Пример блочного кода



### Пример текстового кода

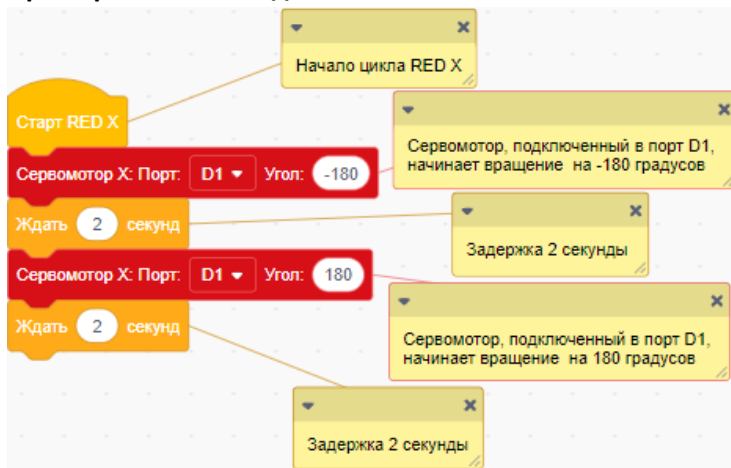
```
\\Строки с данным символом это комментарии, их  
\\прописывать не нужно  
\\Подключаем библиотеку Ардуино  
#include <Arduino.h>  
\\Подключаем библиотеку Моторов  
#include <Motors.h>  
\\Объявляем функцию инициализации периферии  
void setup() {  
  \\Инициализируем моторы  
  motorsInit();  
}  
\\Объявляем бесконечный цикл  
void loop(){  
  \\Вызываем функцию для вращения мотора (M1) по  
  \\часовой стрелке со скоростью 1000 у.е.  
  M1(FORWARD,1000);  
  \\Вызываем функцию для вращения мотора (M2) по  
  \\часовой стрелке со скоростью 1000 у.е.  
  M2(FORWARD,1000);  
  \\Вызываем функцию задержки в 1 секунду  
  delay(1000*1);  
  \\Вызываем функцию для вращения мотора (M1)  
  \\против часовой стрелки со скоростью 1000 у.е.  
  M1(BACKWARD,1000);  
  \\Вызываем функцию для вращения мотора (M2)  
  \\против часовой стрелки со скоростью 1000 у.е.  
  M2(BACKWARD,1000);  
  \\Вызываем функцию задержки в 1 секунду  
  delay(1000*1);  
}
```

Фактически данный алгоритм и созданный программный код обеспечивает такой режим работы моторов, подключённых к портам M1 и M2, что они поворачивают вал по часовой стрелке 1 секунду, затем моторы, подключённые к портам M1 и M2, поворачивают вал против часовой стрелки 1 секунду, после этого весь порядок действий повторяется бесконечно.

Таким образом, подготовленный и загруженный с помощью программы RED CODE 3 в контроллер R:ED X данный программный код обеспечит движение робота R:ED X с смонтированной колесной частью вперед и назад с секундным интервалом с бесконечным повторением.

## Пример кода - Сервомотор X

### Пример блочного кода



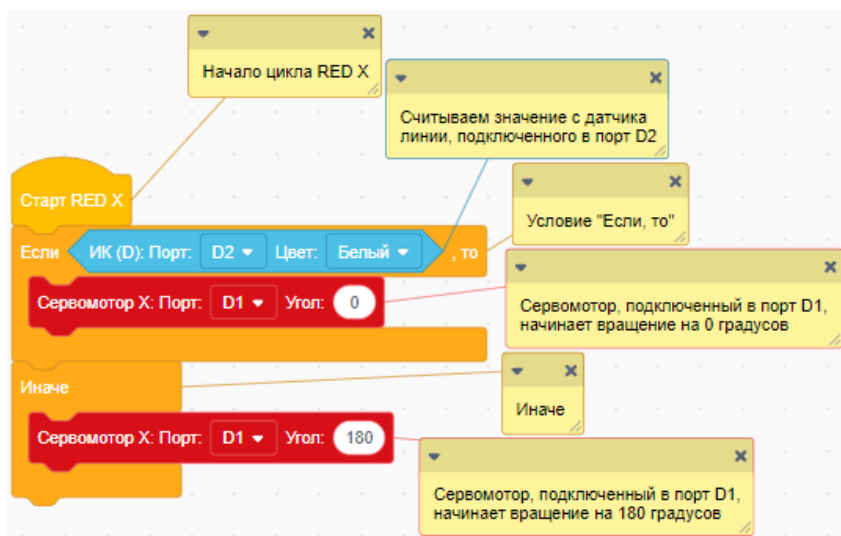
### Пример текстового кода

```
\\Строки с данным символом это комментарии, их
\\прописывать не нужно
\\Подключаем библиотеку Аруино
#include <Arduino.h>
\\Подключаем библиотеку сервомоторов
#include <Servo.h>
\\Объявляем экземпляр класса Servo
Servo servo1;
\\Объявляем функцию инициализации периферии
void setup() {
  \\Назначаем порт D1 выходом сигнала управления
  \\сервомотором
  servo1.attach(D1);
}
\\Объявляем бесконечный цикл
void loop() {
  \\Вызываем функцию поворота сервомотора на -180
  \\градусов
  servo1.write(-180);
  \\ Вызываем функцию задержки в 2 секунды
  delay(1000*2);
  \\Вызываем функцию поворота сервомотора на 180
  \\градусов
  servo1.write(180);
  \\Вызываем функцию задержки в 2 секунды
  delay(1000*2);
}
```

Фактически данный алгоритм и созданный программный код обеспечивает такой режим работы сервомотора, подключенного к порту D1, что он поворачивает вал на -180 градусов, ожидает 2 секунды, затем поворачивает вал на +180 градусов, ожидает 2 секунды, после этого весь порядок действий повторится бесконечно.

## Пример кода - Датчик линии X

### Пример блочного кода



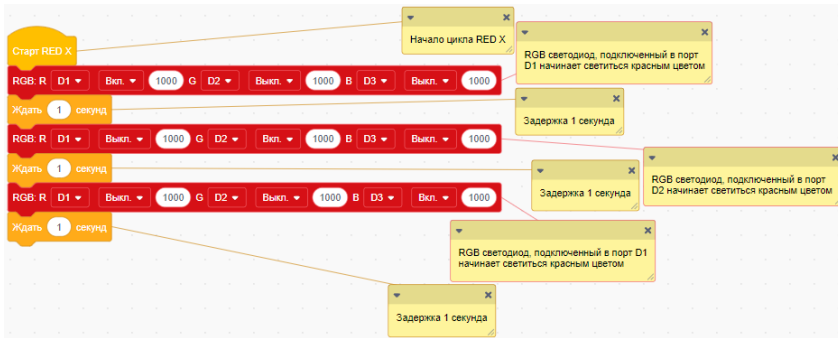
### Пример текстового кода

```
\\Строки с данным символом это комментарии, их
\\прописывать не нужно
\\Подключаем библиотеку Ардуино
#include <Arduino.h>
\\Подключаем библиотеку сервомоторов
#include <Servo.h>
\\Подключаем библиотеку датчика линии
#include <IR.h>
\\Объявляем экземпляр класса Servo
Servo servol;
\\Объявляем экземпляр класса IR
IR irD1;
\\Объявляем функцию инициализации периферии
void setup() {
  \\Назначаем порт D1 выходом сигнала управления
  \\сервомотором
  servol.attach(D1);
  \\Назначаем порт D2 приёмником сигнала с
  \\инфракрасного датчика
  irD1.attachDigital(D2);
}
\\Объявляем бесконечный цикл
void loop(){
  \\Проверяем состояние инфракрасного датчика
  if(irD1.readDigital() == WHITE){
  \\Вызываем функцию поворота сервомотора на 0
  \\градусов
  servol.write(0);
  }
  else{
  \\Иначе вызываем функцию поворота сервомотора на 180
  \\градусов
  servol.write(180);
  }
}
```

Если датчик линии, подключенный к порту D2, считывает белый цвет, то сервомотор, подключенный к порту D1, поворачивается на нулевой градус, иначе сервомотор поворачивается на 180 градусов. Таким образом для робота R:ED X может быть обеспечено отслеживание и движение по линии.

# Пример кода - RGB светодиод X

## Пример блочного кода



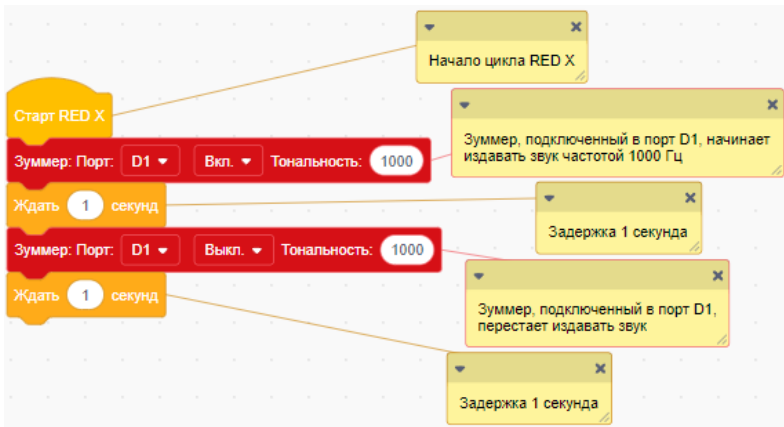
## Пример текстового кода

```
\\Строки с данным символом это комментарии, их
\\прописывать не нужно
\\Подключаем библиотеку Ардуино
#include <Arduino.h>
\\Подключаем библиотеку RGB-светодиодов
#include <RGB.h>
\\Объявляем экземпляр класса RGB
RGB rgb1;
\\Объявляем функцию инициализации периферии
void setup() {
  \\Назначаем порты D1,D2,D3 выводами сигнала
  \\управления RGB-светодиодом
  rgb1.R.attach(D1);
  rgb1.G.attach(D2);
  rgb1.B.attach(D3);
}
\\Объявляем бесконечный цикл
void loop(){
  \\Устанавливаем значение канала R в 1000u.e.
  rgb1.R.write(1000);
  \\Устанавливаем значение канала G в 0u.e.
  rgb1.G.write(0);
  \\Устанавливаем значение канала B в 0u.e.
  rgb1.B.write(0);
  \\Вызываем функцию задержки в 1 секунду
  delay(1000*1);
  \\Устанавливаем значение канала R в 0u.e.
  rgb1.R.write(0);
  \\Устанавливаем значение канала G в 1000u.e.
  rgb1.G.write(1000);
  \\Устанавливаем значение канала B в 0u.e.
  rgb1.B.write(0);
  \\Вызываем функцию задержки в 1 секунду
  delay(1000*1);
  \\Устанавливаем значение канала R в 0u.e.
  rgb1.R.write(0);
  \\Устанавливаем значение канала G в 0u.e.
```

RGB светодиод, подключенный в порты D1, D2, D3, начинает светиться красным цветом 1 секунду, затем зелёным цветом 1 секунду, затем синим цветом 1 секунду, после этого весь порядок действий повторяется бесконечно.

# Пример кода - Зуммер X

## Пример блочного кода



## Пример текстового кода

```
//Строки с данным символом это комментарии, их  
//прописывать не нужно  
//Подключаем библиотеку Ардуино  
#include <Arduino.h>  
//Подключаем библиотеку Зумера  
#include <Tone.h>  
//Объявляем функцию инициализации периферии  
void setup() {  
}  
//Объявляем бесконечный цикл  
void loop(){  
  //Вызываем функцию включения звука частотой в  
  //1000 Гц  
  tone(D1, 1000);  
  //Вызываем функцию задержки в 1 секунду  
  delay(1000*1);  
  //Вызываем функцию выключения звука  
  tone(D1, 0);  
  //Вызываем функцию задержки в 1 секунду  
  delay(1000*1);  
}
```

Зуммер, подключенный к порту D1, начинает издавать звук 1 секунду, затем перестает издавать звук на 1 секунду, после этого весь порядок действий повторяется бесконечно.

## Пример кода - Ультразвуковой датчик X

### Пример блочного кода



### Пример текстового кода

```
#include <Arduino.h>
#include <Servo.h>
#include <Ultrasonic.h>

Servo servol;
Ultrasonic us1;

void setup() {
  servol.attach(D1);
  us1.attachTrig(D3);
  us1.attachEcho(D2);
}

void loop() {
  if((us1.readDistance() < 10)){
    servol.write(0);
  }
  else{
    servol.write(180);
  }
}
```

Если ультразвуковой датчик, подключенный к порту D2, D3 считывает белый цвет, то сервомотор, подключенный к порту D1, поворачивается на нулевой градус, иначе сервомотор поворачивается на 180 градусов.

Таким образом все компоненты робототехнического набора R:ED X, имеющие возможность программного управления, что отражено в конечном наборе датчиков и исполнительных механизмов в основном меню выбора компонентов набора к программированию, могут быть запрограммированы с использованием всех возможным логических, математических, событийных, исполнительных и других функций. Совместимость по применению обеспечивается автоматически.