

Функциональные характеристики RED CODE 2.0

«RED CODE 2.0» - это программное приложение, устанавливаемое на персональный компьютер, работающий на операционных системах Windows 7, Windows 8 или Windows 10, и представляет собой визуально-событийную среду программирования, с возможностью составления программного кода с помощью блоков, которые описывают действия электронных устройств и полностью соответствуют всем возможностям построения логики и математических операций на контроллере, входящих в состав робототехнического набора R:ED X <https://store.r-ed.world/sets#redx>. Также с помощью данной программы можно проводить прошивку контроллера, входящего в состав робототехнического набора R:ED X <https://store.r-ed.world/sets#redx>. Приложение позволяет сохранять и загружать составленные пользователем программы для контроллера из набора R:ED X <https://store.r-ed.world/sets#redx> и редактировать данные программы с помощью встроенного текстового редактора.

Программа распространяется бесплатно и поддерживает все функциональные и программные возможности выбранного робототехнического набора R:ED X <https://store.r-ed.world/sets#redx>

Такой подход к распространению обеспечен исполнением требований открытой лицензии библиотеки Scratch https://en.scratch-wiki.info/wiki/Scratch_Source_Code_License

Минимальные требования для ПК:

Microsoft Windows XP, VISTA, 7, 8 и 10

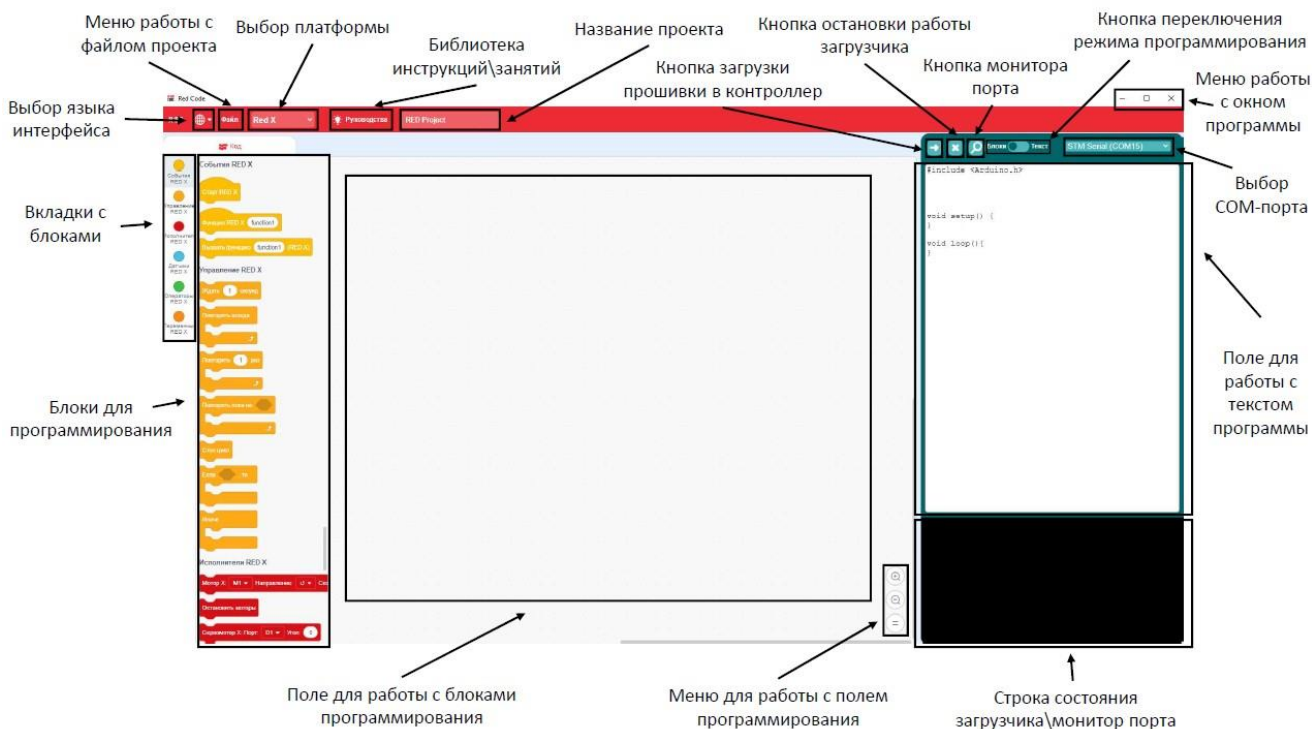
3 Гб ОЗУ

300 Мб свободного пространства на жестком диске

USB-соединение

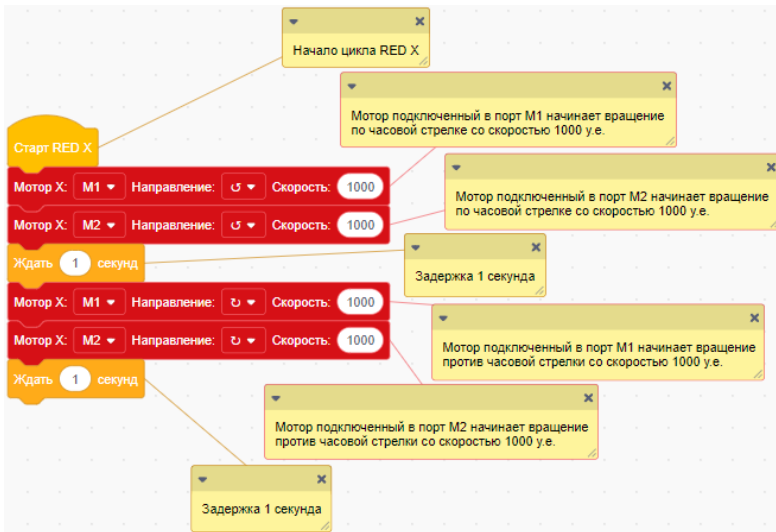
подключение к сети Интернет

Основной пользовательский интерфейс RED CODE 2.0



Пример кода RED Code 2.0 (Мотор X)

Пример блочного кода



Пример текстового кода

```
STM Serial (COM14)
//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 2.0 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 RED Code 2.0 (Сервомотор X)

Пример блочного кода



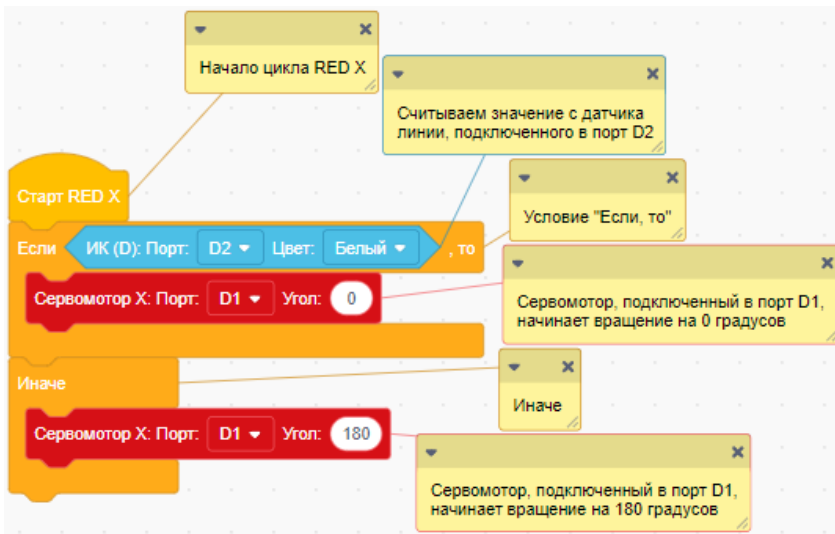
Пример текстового кода

```
STM Serial (COM14)
//Строки с данным символом это комментарии, их
//прописывать не нужно
//Подключаем библиотеку Ардуино
#include <Arduino.h>
//Подключаем библиотеку сервомоторов
#include <Servo.h>
//Объявляем экземпляр класса Servo
Servo serv01;
//Объявляем функцию инициализации периферии
void setup() {
  //Назначаем порт D1 выходом сигнала управления
  //сервомотором
  serv01.attach(D1);
}
//Объявляем бесконечный цикл
void loop() {
  //Вызываем функцию поворота сервомотора на -180
  //градусов
  serv01.write(-180);
  //Вызываем функцию задержки в 2 секунды
  delay(1000*2);
  //Вызываем функцию поворота сервомотора на 180
  //градусов
  serv01.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 повторяется бесконечно.

Пример кода RED Code 2.0 (Датчик линии X)

Пример блочного кода



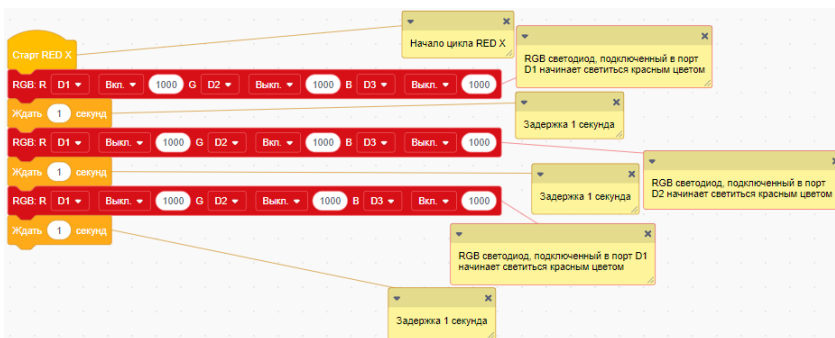
Пример текстового кода

```
STM Serial (COM14)
//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 serv01;
//Объявляем экземпляр класса IR
IR irD1;
//Объявляем функцию инициализации периферии
void setup() {
  //Назначаем порт D1 выходом сигнала управления
  //сервомотором
  serv01.attach(D1);
  //Назначаем порт D2 приемником 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
    serv01.write(0);
  }
  else{
    //Иначе вызываем функцию поворота сервомотора на 180
    //градусов
    serv01.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 RED Code 2.0 (RGB – светодиод X)

Пример блочного кода



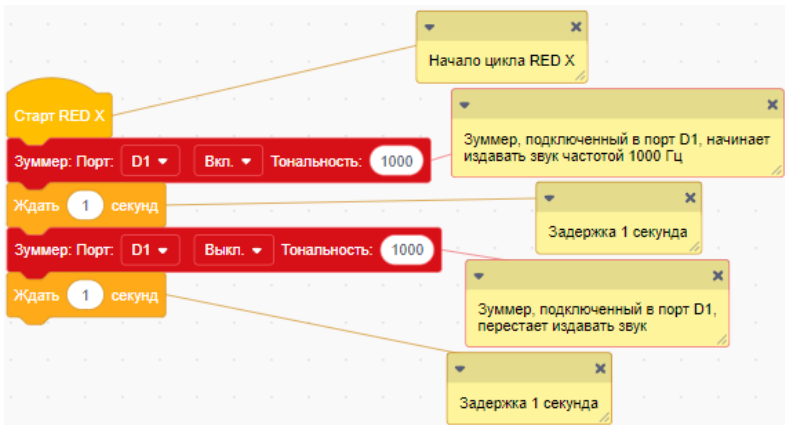
Пример текстового кода

```
STM Serial (COM14)
//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 выходами 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 1000y.e.
  rgb1.R.write(1000);
  //Устанавливаем значение канала G в 0y.e.
  rgb1.G.write(0);
  //Вызываем функцию задержки в 1 секунду
  delay(1000*1);
  //Устанавливаем значение канала R в 0y.e.
  rgb1.R.write(1000);
  //Устанавливаем значение канала G в 1000y.e.
  rgb1.G.write(0);
  //Устанавливаем значение канала B в 0y.e.
  rgb1.B.write(0);
  //Вызываем функцию задержки в 1 секунду
  delay(1000*1);
  //Устанавливаем значение канала R в 0y.e.
  rgb1.R.write(1000);
  //Устанавливаем значение канала G в 0y.e.
  //Устанавливаем значение канала B в 1000y.e.
  rgb1.B.write(1000);
}
```

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 RED Code 2.0 (Зуммер X)

Пример блочного кода



Пример текстового кода

```
STM Serial (COM14)
//Строки с данным символом это комментарии, их
//прописывать не нужно
//Подключаем библиотеку Ардуино
#include <Arduino.h>
//Подключаем библиотеку Зуммера
#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 повторится бесконечно.

Пример кода RED Code 2.0 (Ультразвуковой датчик X)

Пример блочного кода



Пример текстового кода

```
STM Serial (COM15)
#include <Arduino.h>
#include <Servo.h>
#include <Ultrasonic.h>

Servo servo1;
Ultrasonic us1;

void setup() {
  servo1.attach(D1);
  us1.attachTrig(D3);
  us1.attachEcho(D2);
}

void loop(){
  if(us1.readDistance() < 10){
    servo1.write(0);
  }
  else{
    servo1.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.