



Муниципальное бюджетное общеобразовательное учреждение  
"Лицей-интернат"  
Великий Новгород



# Робот РОБОНЯША

Николаева Валерия, Нырка Оксана  
10 Л класс

# Цели и задачи проекта

**Цель:** Создание робота на плате Iskra JS, который управляется с помощью ИК-пульта.

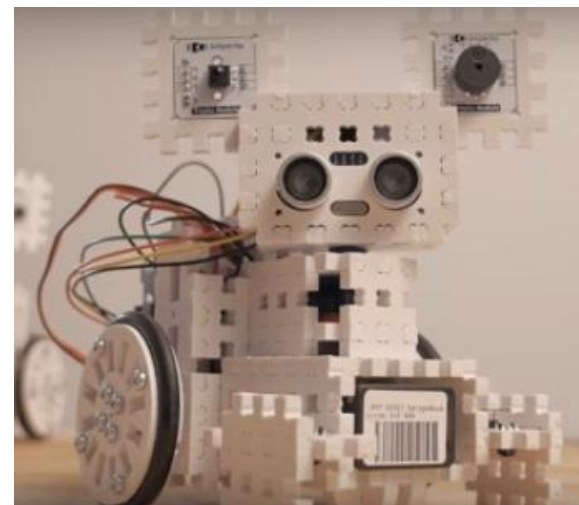
## **Задачи:**

1. Изучить строение платы Iskra JS
2. Изучить работу двигателей
3. Изучить строение платы Motor Shield по управлению двигателями
4. Собрать робота с помощью структора и установить на него необходимые платы и датчики
5. Написать программу для управления роботом

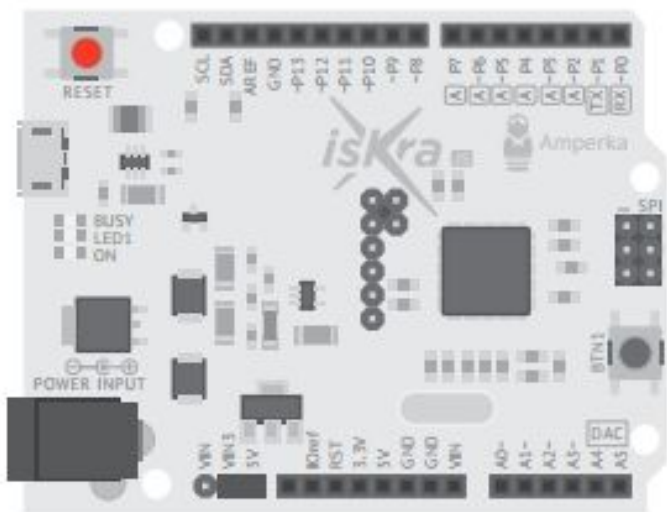
## **Практическая значимость проекта:**

На основании проекта можно:

- создать робота, который будет участвовать в конкурсах по робототехнике.
- создать умного робота, под управлением ИК – пульта или под управлением датчиков (ультразвуковой дальномер, цифровые и аналоговые датчики линий)



# Плата Iskra JS



Плата Iskra JS  
Мозг Робоняши. Выполняет программы робота

Iskra JS — это маленький компьютер, мозг нашей Робоняши. Он умеет измерять напряжение, умеет его выдавать. Этого достаточно, чтобы взаимодействовать с внешним миром: считывать всевозможные сенсоры, выдавать команды на реле, моторы, светодиоды, дисплеи.

Чтобы получилось законченное устройство, нужно описать его поведение. Это поведение мы можем запрограммировать на языке JavaScript, загрузить программу в плату и таким образом получить думающего робота!

# Работа двигателей



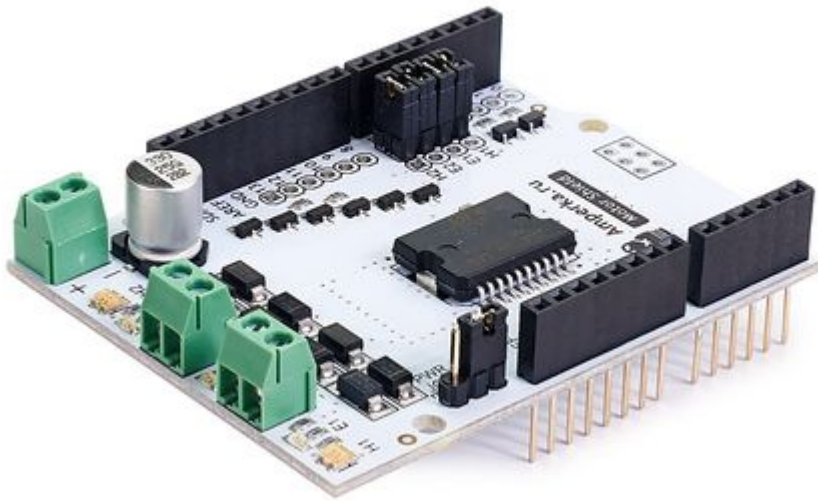
Крепим колёса к моторам, которые заставят двигаться нашего робота.

Для работы нам понадобится микромотор с передаточными шестернями, формирующими редуктор, и D-образным шпинделем; он подойдёт для приведения в движение колёс (рабочее напряжение 3-6 вольт).

Мотор к микроконтроллеру напрямую подключать нельзя— большой ток выведет его из строя! Мы будем использовать плату с драйвером для мотора - Motor Shield.

# Плата Motor Shield

## Motor Shield



Плата позволяет управлять скоростью и направлением вращения мотора с помощью логических сигналов микроконтроллера.

При разгоне и торможении двигатели сами индуцируют кратковременный обратный ток большой величины, который может выжечь контакты микроконтроллера.

На Motor Shield установлены возвратные диоды, которые это предотвращают.

На плате расположены светодиоды-индикаторы, показывающие направление и скорость по каждому из каналов и подачу питания.

# Сборка робота



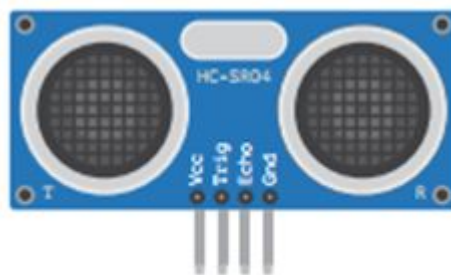
Тройка-модуль светодиод  
Светит по команде Iskra JS



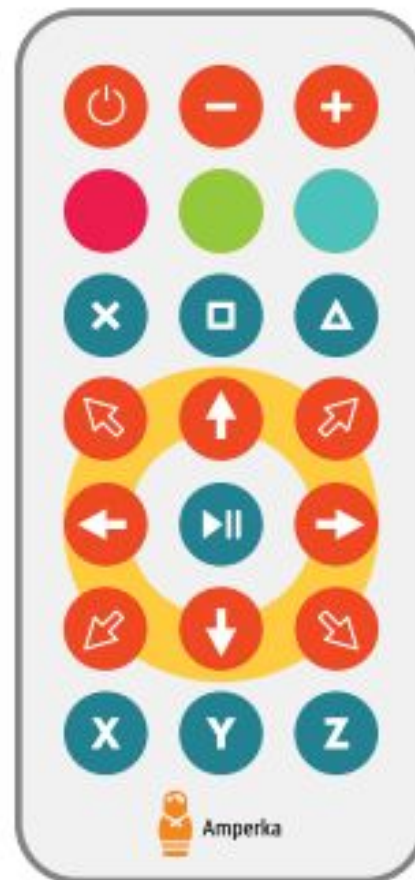
Тройка-модуль  
ИК-приёмник  
Принимает команды  
ИК-пульта



Кабель micro-USB  
Соединяет Iskra JS  
с компьютером



Ультразвуковой дальномер

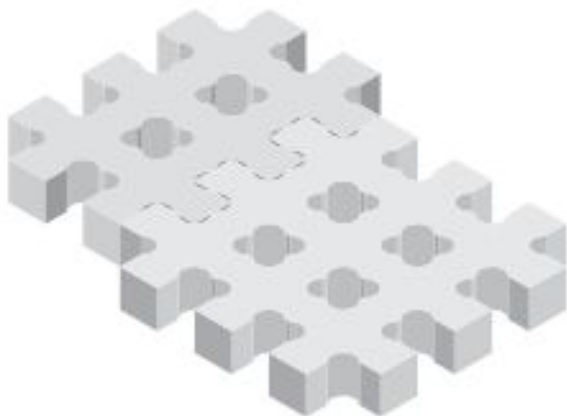


ИК-пульт

# Структор

## **КОНСТРУКТОР ДЛЯ КОРПУСА**

# Структор — это решётчатый конструктор из ПВХ.  
# Структор можно склеивать, ломать и резать.

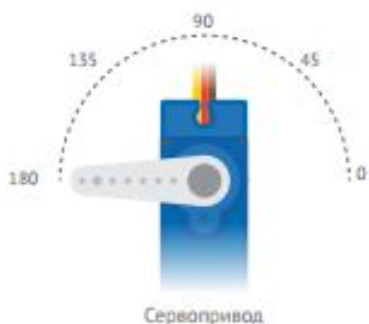


Детали можно соединять  
в плоскость.



Можно ставить стенки.

# Поворот головы

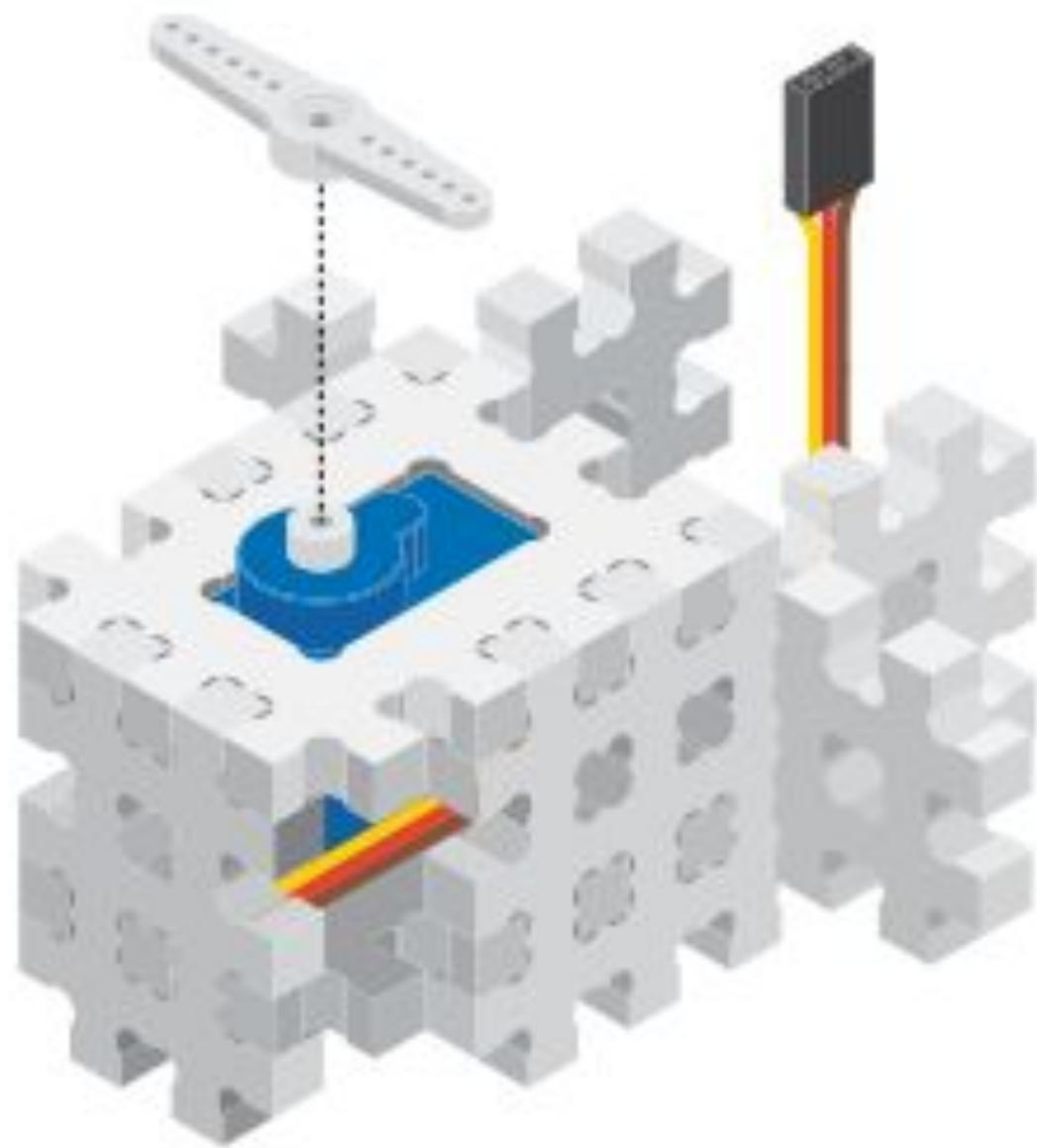


Для этого нам потребуется двигатель, но обычные двигатели вращаются непрерывно с заданной скоростью. Для головы это не подходит, она должна поворачиваться влево или вправо только до заданного угла.

Будем использовать сервопривод.

Сервопривод (или просто — серво) не умеет вращаться непрерывно, зато он умеет устанавливать свой вал на заданный угол. Например, мы можем задать угол в  $34^\circ$ . Серво переведёт вал на этот угол и будет удерживать его. Сервопривод может поворачиваться на угол от  $0^\circ$  до  $180^\circ$ .





# Программа для управления роботом

Ни один робот не работает без внутренней программы. Программу для нашей Робоняши будем писать в специальном редакторе кода в Espruino IDE, этот редактор умеет загружать написанную программу в Iskra JS.



Пишем программу на языке JavaScript. Это популярный язык программирования. JavaScript обычно используют для создания веб-приложений, но его также может исполнять Iskra JS.

# Основные компоненты Робоняши, и соответствующие им программы

Дадим всем знать о приближении робота. Установим на ушке Робоняши мигающий сигнал – светодиод.

```
var notice = require('@gamperka/led')
  .connect(P3);

notice.blink(0.1, 0.9);
```

Заставим Робоняшу поворачивать шею, с помощью микросервопривода, чтобы она могла смотреть из стороны в сторону и напишем для этого соответствующий код.

# Поворот головы

```
1 { var neck = require('@amperka/servo').connect(P8);
2
3   var angle = 90;
4   var STEP = 5;
5
6   setInterval(function() {
7     if (angle <= 30 || angle >= 150) {
8       STEP = -STEP;
9     }
10    angle = angle + STEP;
11    neck.write(angle);
12  }, 100);
```

**a** С помощью библиотеки '@amperka/servo' подключаем наш сервопривод к пину P8.

**b** Задаём первоначальный угол серво на 90° и шаг поворота серво. Пусть шаг будет 5°. Поэкспериментируй с разными значениями.

**c** Когда вал повернётся максимально влево или максимально вправо, меняем знак шага с плюса на минус и наоборот. Так мы заставим серво изменить направление поворота.

**d** Прибавляем к установленному углу 1 шаг. Если значение шага отрицательное, угол будет уменьшаться.

**e** Функция `write()` устанавливает угол вала сервопривода.

# Основные компоненты Робоняши, и соответствующие им программы

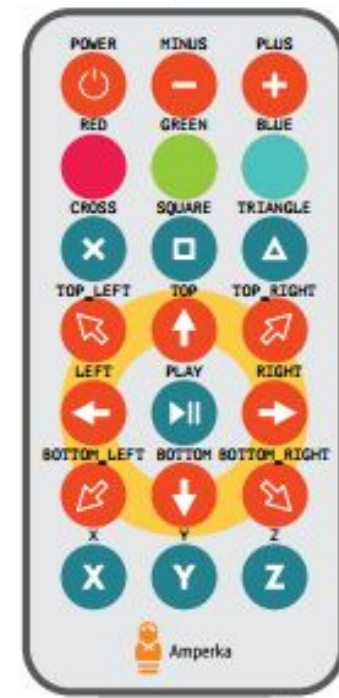
Будем управлять с помощью пульта на инфракрасных лучах. Точь-в-точь как телевизор или кондиционер. Пульт ИК при нажатии кнопки излучает кодированную посылку, а приёмник принимает её.

Кодированный сигнал — это последовательность импульсов разной длительности. Ноль — короткий сигнал, единица — длинный.

Приёмник получает кодированный сигнал и передаёт его на Iskra JS. Каждая кнопка на пульте имеет свой собственный код.

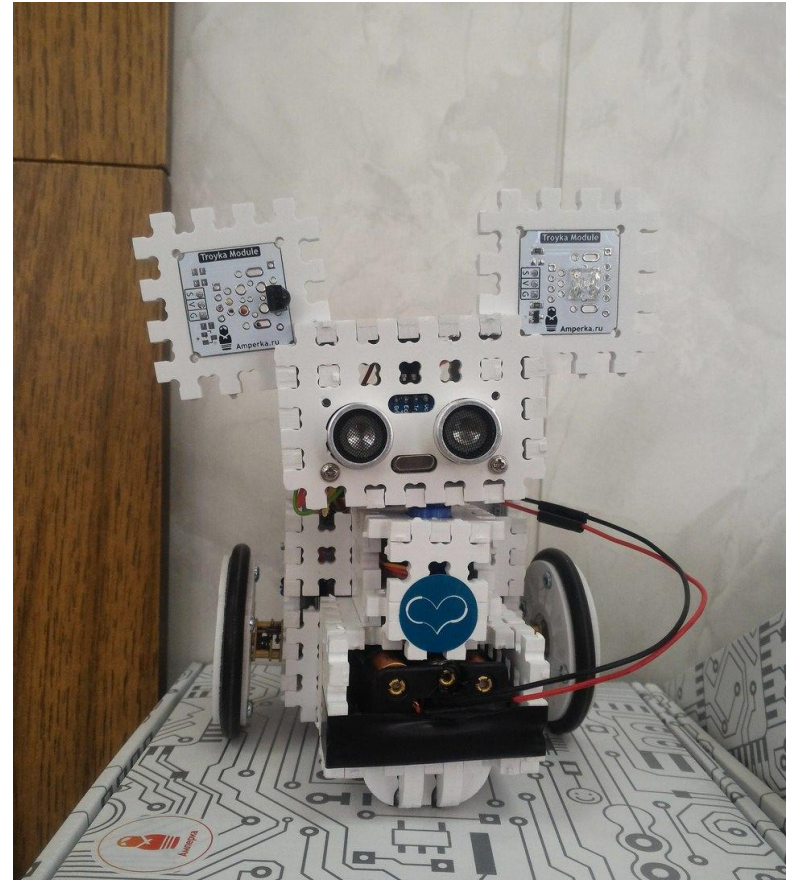
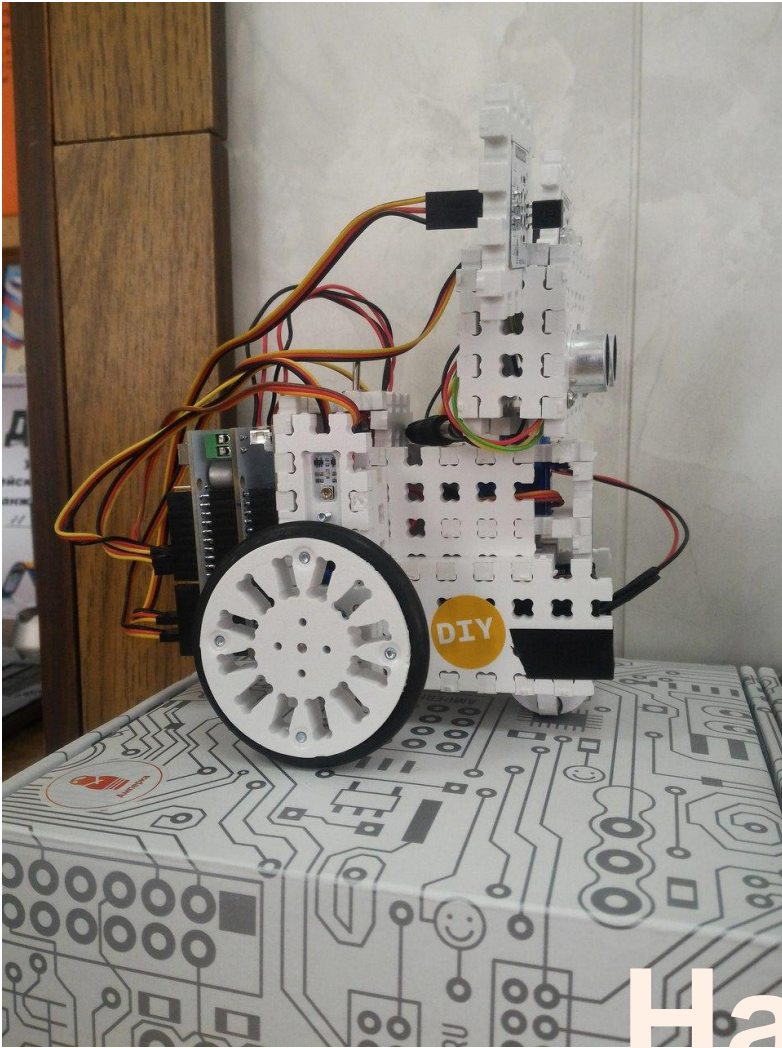


001010..



# Полная программа для управления роботом с ПОМОЩЬЮ ПУЛЬТА

```
1 var notice = require('@amperka/led').connect(P1);
2 notice.blink(0.1, 0.9);
3 var neck = require('@amperka/servo').connect(P8);
4 var angle = 90;
5 var STEP = 5;
6 setInterval(function() {
7   if (angle <= 60 || angle >= 120) {
8     STEP = -STEP;
9   }
10  angle = angle + STEP;
11  neck.write(angle);
12 }, 100);
13
14 var marsohod = require('@amperka/robot-2wd')
15   .connect();
16 var receiver = require('@amperka/ir-receiver')
17   .connect(P3);
18 receiver.on('receive', function(code) {
19   if (code === receiver.keys.TOP) {
20     marsohod.go({l: 1, r: 1});
21   }
22   if (code === receiver.keys.POWER) {
23     marsohod.stop();
24   }
25 });
26 var SPEED = 0.3;
27 var marsohod = require('@amperka/robot-2wd')
28   .connect();
29 var receiver = require('@amperka/ir-receiver')
30   .connect(P3);
31 receiver.on('receive', function(code) {
32   if (code === receiver.keys.TOP) {
33     marsohod.go({l: SPEED, r: SPEED});
34   }
35   if (code === receiver.keys.BOTTOM) {
36     marsohod.go({l: -SPEED, r: -SPEED});
37   }
38   if (code === receiver.keys.LEFT) {
39     marsohod.go({l: SPEED, r: 0});
40   }
41   if (code === receiver.keys.RIGHT) {
42     marsohod.go({l: 0, r: SPEED});
43   }
44 });
```



# Наша Робоняша



# Выводы

В данном проекте мы закрепили свои знания в области конструирования Arduino. Научились управлять моторами, изучили строение плат MotorShield и TroykaShield для управления роботами. На основании этого проекта мы построили умного робота – Робоняшу, которым можно управлять с помощью ИК-пульта. Представленная программа для управления Робоняшей оставляет множество возможностей для улучшения и оптимизации! Скорость поворота можно менять плавно, можно добавить контроль заноса, можно менять расположения сенсоров и центров масс, что даст другой результат. В конце концов можно получить непобедимого на трекке робота!



