

Конструирование и программирование роботов на платформе Lego Mindstorms

Якушин Алексей Валериевич, к.п.н, доцент
Лаборатория ОИТ ВМК МГУ
Гладких Илья Олегович, аспирант
ТГПУ им. Л.Н. Толстого

Направления развития

Космос



Биотехнологии



Робототехника

LEGO Mindstorms EV3



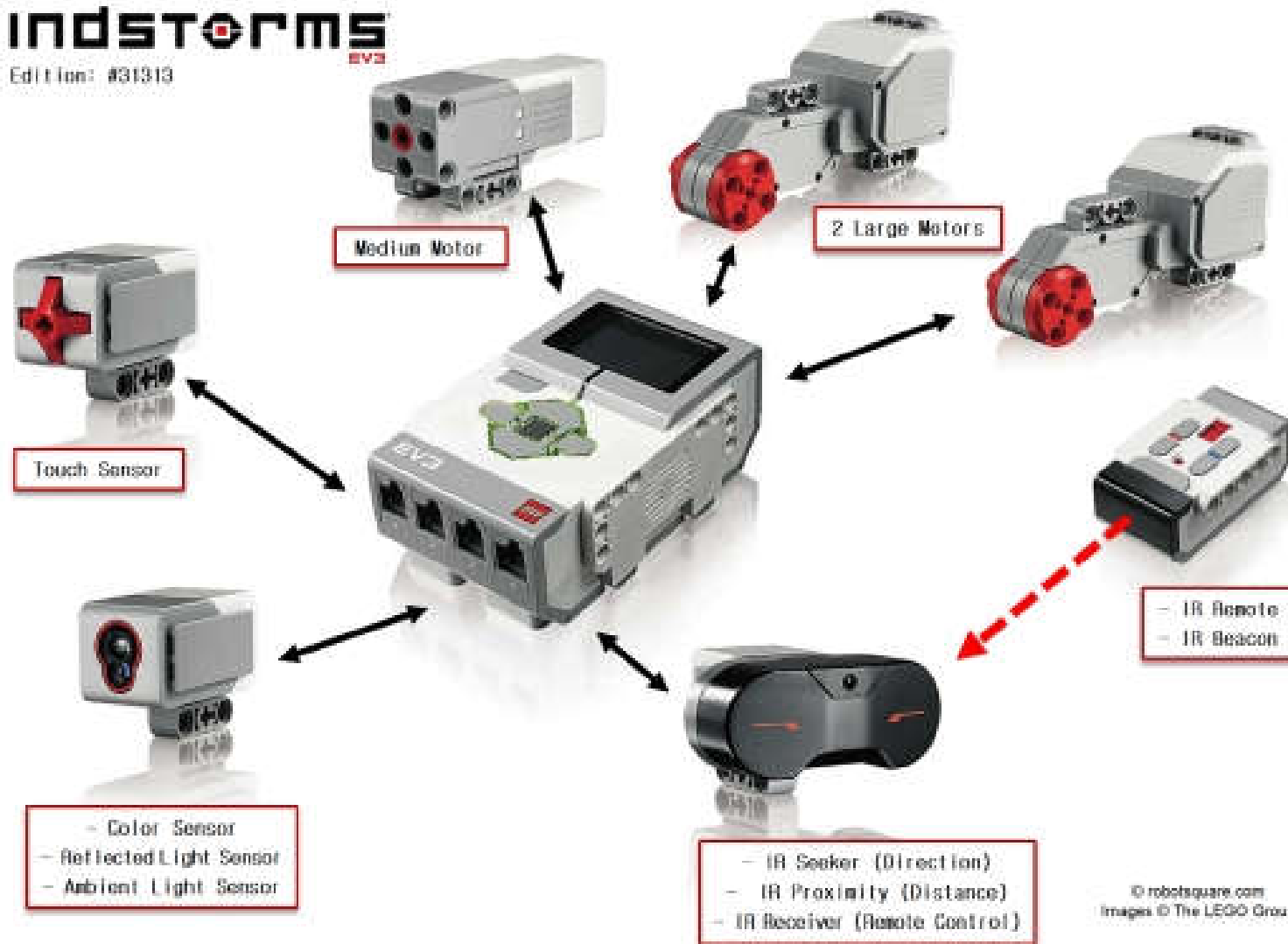
Общие сведения

- Набор предназначен для пользователей в возрасте от 10 лет до 21 года. В наборе свыше 500 деталей, совместимых с деталями серии LEGO Technic. Набор выпускается в нескольких комплектациях: для класса, для индивидуального пользователя, ресурсная.

Электронные компоненты EV3

MINDSTORMS
EV3

Home Edition: #31313



Микроконтроллер EV3 – интерфейсы



6 кнопок с LED-подсветкой
4 порта для моторов
Дисплей 178x128
Слот для карт памяти mini SD
Порт USB 2.0 тип А
Bluetooth, Wi-Fi
Громкий динамик

Микроконтроллер EV3 – параметры

Процессор ARM 9, 300 МГц с ОС Linux, 4 порта ввода для получения данных с частотой 1000 отсчетов/сек, 4 порта вывода для выполнения команд, программная память 16 Мб флэш и 64 Мб ОЗУ, мини SDHC карт ридер для 32 Гб внешней памяти, питание 6 батареек AA или Li-Ion аккумулятор 2050 mAh, возможно автономное программирование



Большой сервомотор

Мощный мотор со
встроенным датчиком
угла поворота с
точностью до 1 градуса;
160-170 об/мин; момент
вращения 335 г*см,
момент удержания 670
г*см; может быть
синхронизирован с
другим мотором для
движения строго по
прямой; автоматически
распознается
встроенным ПО



Средний сервомотор

Скорость вращения
240-250 об/мин;
встроенный датчик
угла поворота с
точностью до 1
градуса; момент
вращения 115 г*см;
момент удержания
170 г*см;
автоматически
распознается
встроенным ПО.



Ультразвуковой датчик расстояния

Измеряет расстояние до отражающего звук предмета в интервале 3-250 см с точностью +/- 1 см; может использоваться как датчик звука; индикатор светится непрерывно, когда датчик излучает, и мигает, когда слушает; автоматически распознается встроенным ПО.



Датчик касания (кнопка)

Определяет, нажата или отпущена кнопка, умеет подсчитывать одиночные и многократные нажатия; автоматически распознается встроенным ПО.



Гироскопический датчик

Цифровой гироскопический датчик. В режиме «угол» измеряет угловое положение с точностью ± 3 градуса; в режиме «гироскоп» измеряет скорость вращения до 440 град/сек; автоматически распознается встроенным ПО.



Датчик цвета/света

- Цифровой датчик цвета различает 8 цветов и определяет освещенность в широком диапазоне: от темноты до яркого солнечного дня. Меряет отраженный красный свет и общий фоновый. Различает синий, зеленый, желтый, красный белый и коричневый, а также различает цветное и черно-белое изображение. Частота опроса 1 кГц; автоматически распознается встроенным ПО.



Инфракрасный датчик-поисковик

Цифровой инфракрасный датчик-поисковик определяет близость к роботу и считывает сигналы инфракрасного маяка. Близостью считается расстояние в 50-70 см. Рабочая дистанция от маяка до 2 метров. Поддерживает 4 сигнальных канала. Принимает команды с пульта управления. Автоматически распознается встроенным ПО.



Инфракрасный маяк – пульт ДУ

4 ИК-канала; кнопка активации/деактивации; зеленый светодиод сигнализирует об активности маяка; автоматическое выключение, если нет активности в течение часа; рабочее расстояние до двух метров; питание от двух батареек ААА.

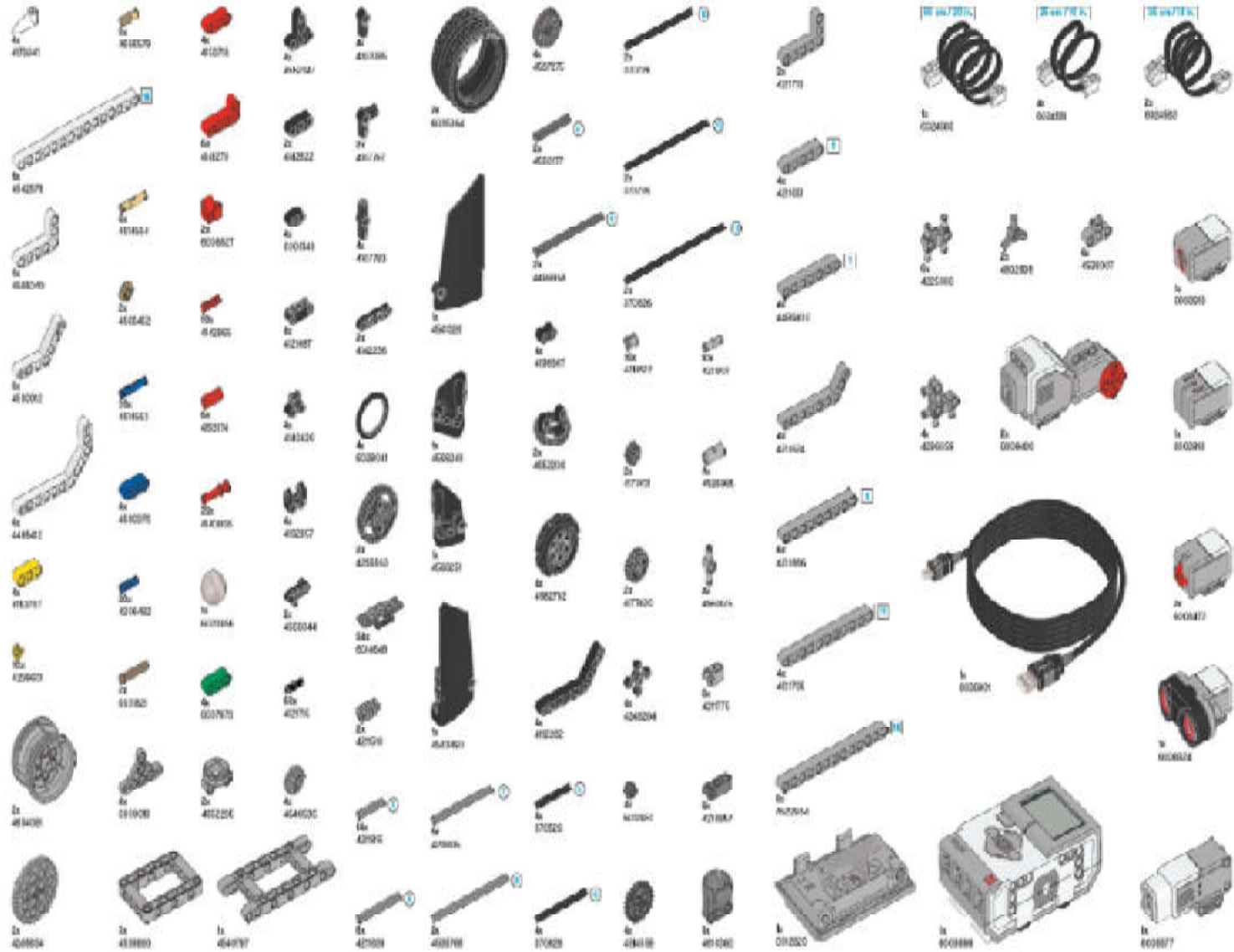


Базовый набор EV3

541 деталь



Состав базового набора EV3

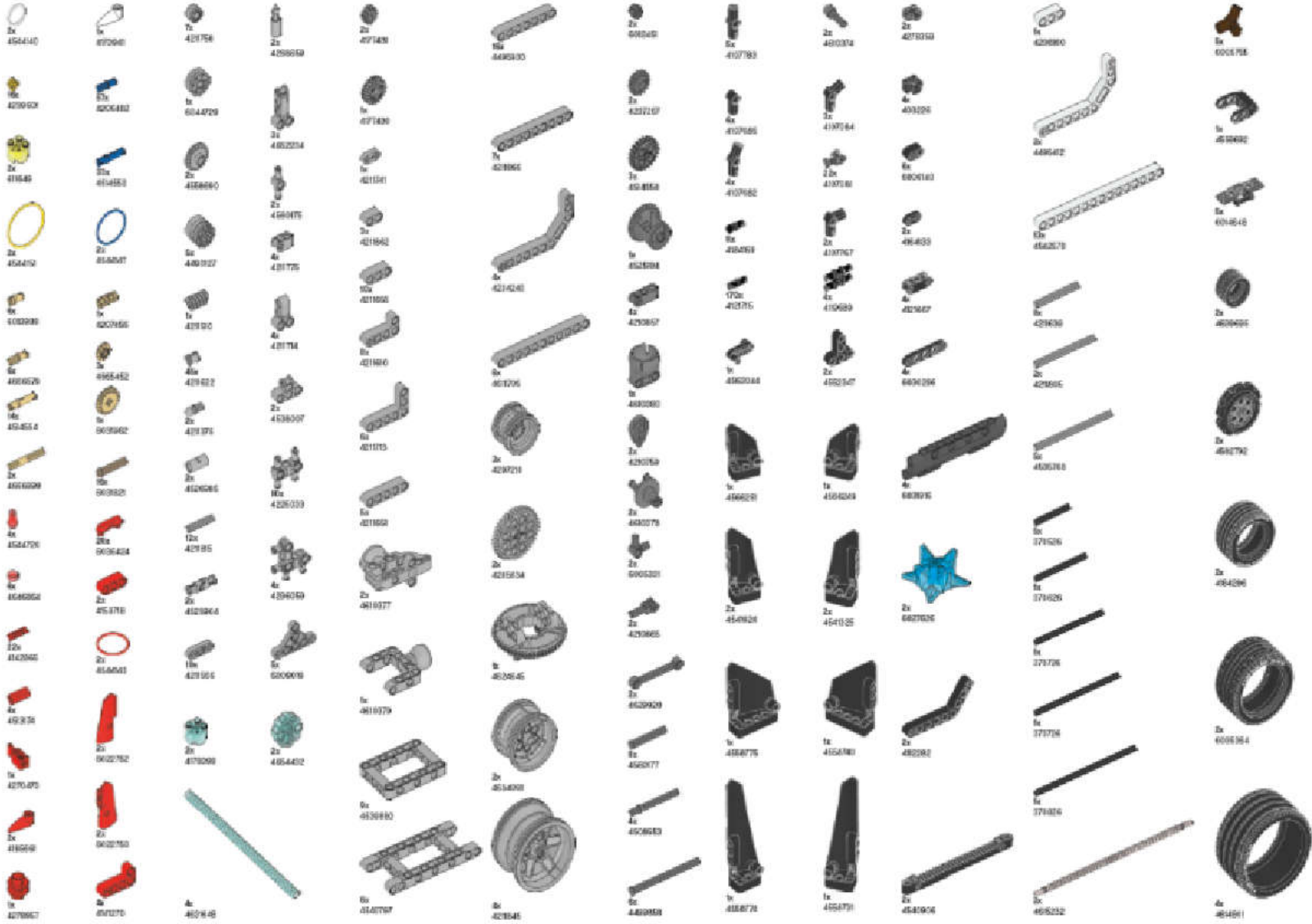


Ресурсный набор EV3

853 детали



Состав ресурсного набора EV3



Этапы подготовки и построения робототехнической платформы

Анализ

- 1) Определение цели и задачи проекта**
- 2) Определение общей концепции проекта**

Дифференциация

3) Выделение подзадач

**4) Выделение независимых в
разработке элементов проекта**

Требования к конструкции

5) Определение конструктивных требований к проекту

5.1) Определение требований к конструкции всего проекта

5.1) Определение требований к конструкции отдельных элементов

Требования к функциям

6) Определение функциональности

6.1) Определение требований к функциям всего проекта

6.1) Определение требований к функциям отдельных элементов

Анализ требований

7) Анализ заявленных конструктивных и функциональных требований

7.1) Определение приоритетных функций и конструктивных задач

7.2) Коррекция поставленных требований по функциям и конструкции в соответствии с пунктом 6.1

7.3) Возвращение к шагу 3, при необходимости.

Подготовка

8) Создание виртуальной* модели исполнителя

9) Создание информационной схемы работы исполнителя

Сборка и тестирование

10) Сборка модели исполнителя

**11) Реализация информационной схемы
в виде программы**

12) Тестирование проекта

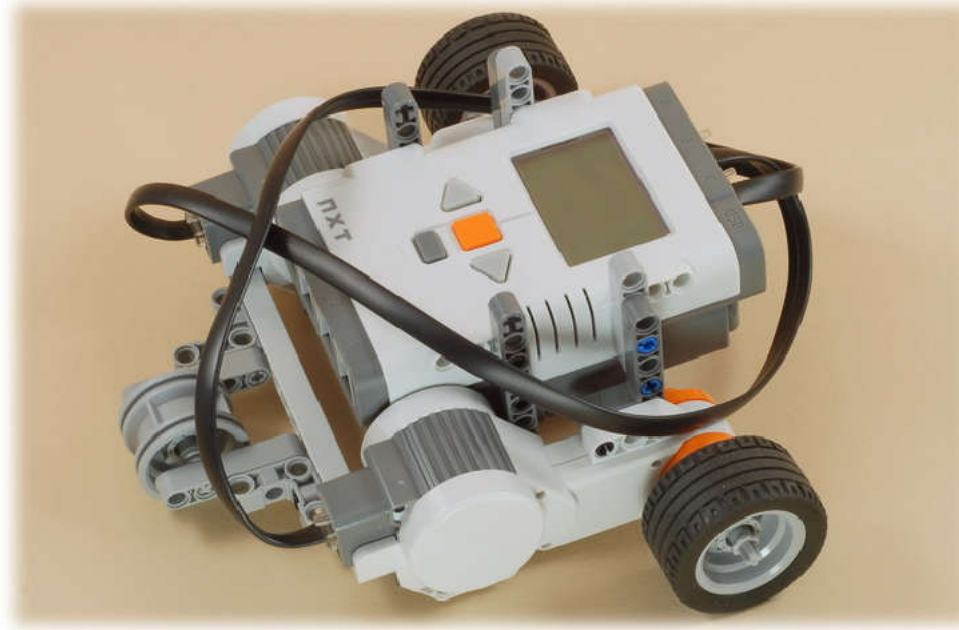
Робототехника.
Алгоритмические
конструкции.

Роботы и алгоритмы

- **Абстракции и реальные действия**
- **Выявление семантических ошибок**
- **Форма записи и выполнения алгоритма**

Выбор исполнителя (робота-модели)

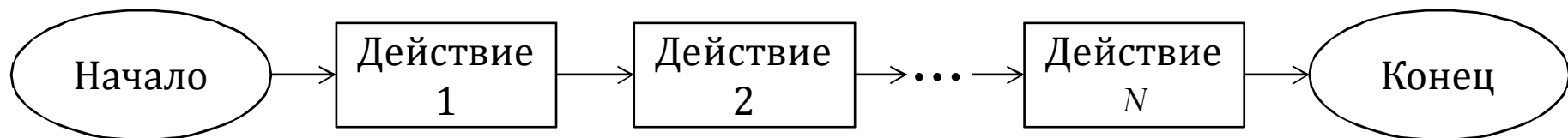
- Для демонстрации базовых алгоритмических конструкций достаточно простой модели с минимально необходимым набором деталей и приспособлений



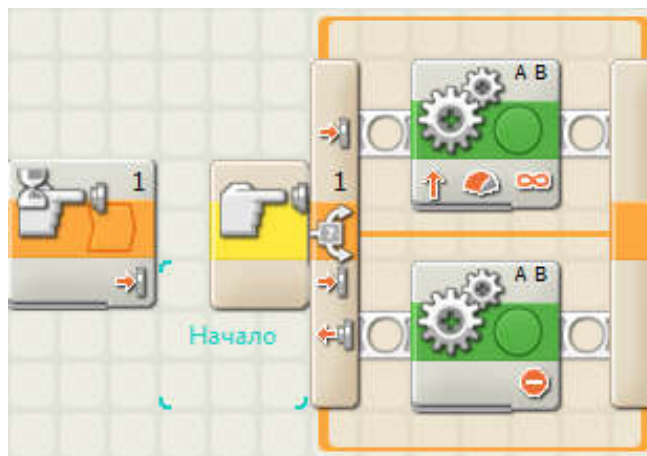
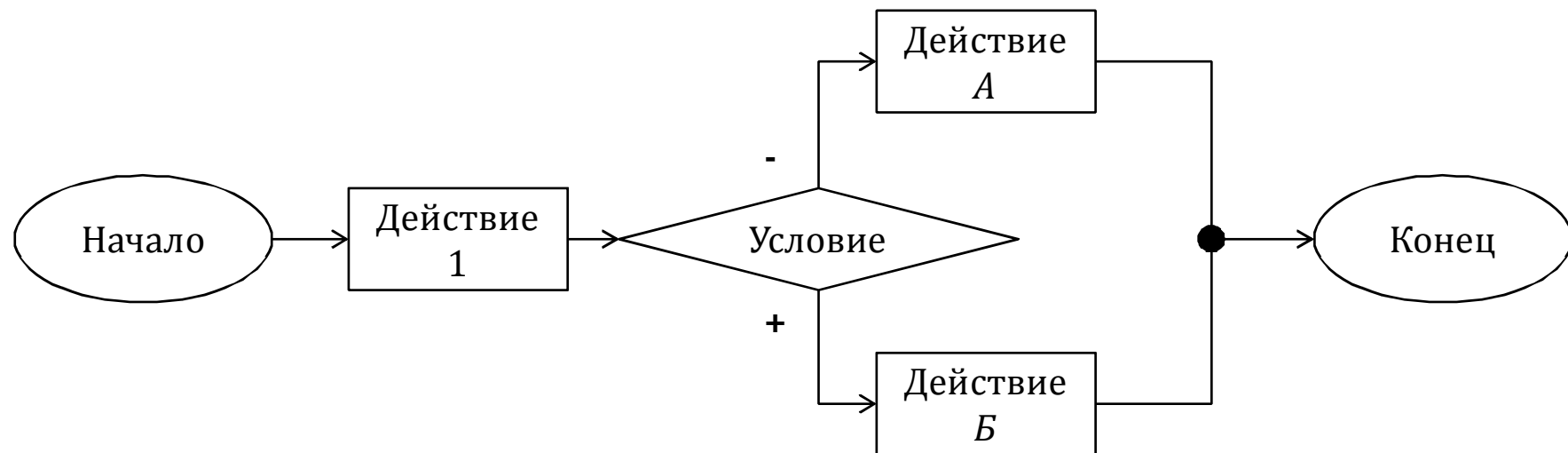
Требования к исполнителю

- ***Способность воспроизведения звуковых файлов** *или вывода на дисплей*
- ***Способность движений вперед, назад и поворота**
- **Работа с датчиками касаний**, *достаточно двух датчиков*

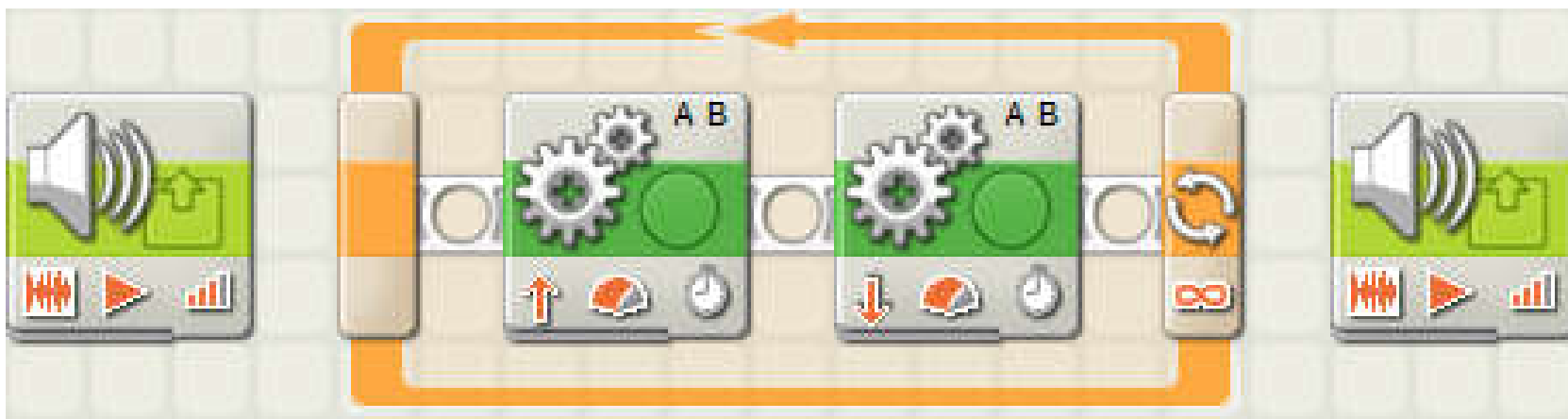
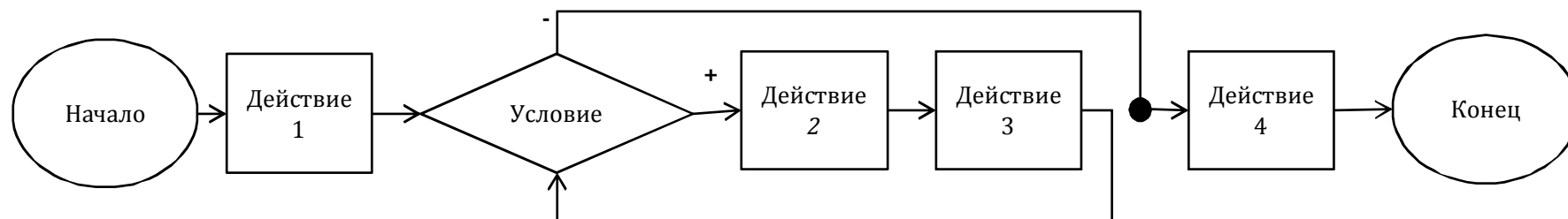
Линейный алгоритм



Ветвящийся алгоритм

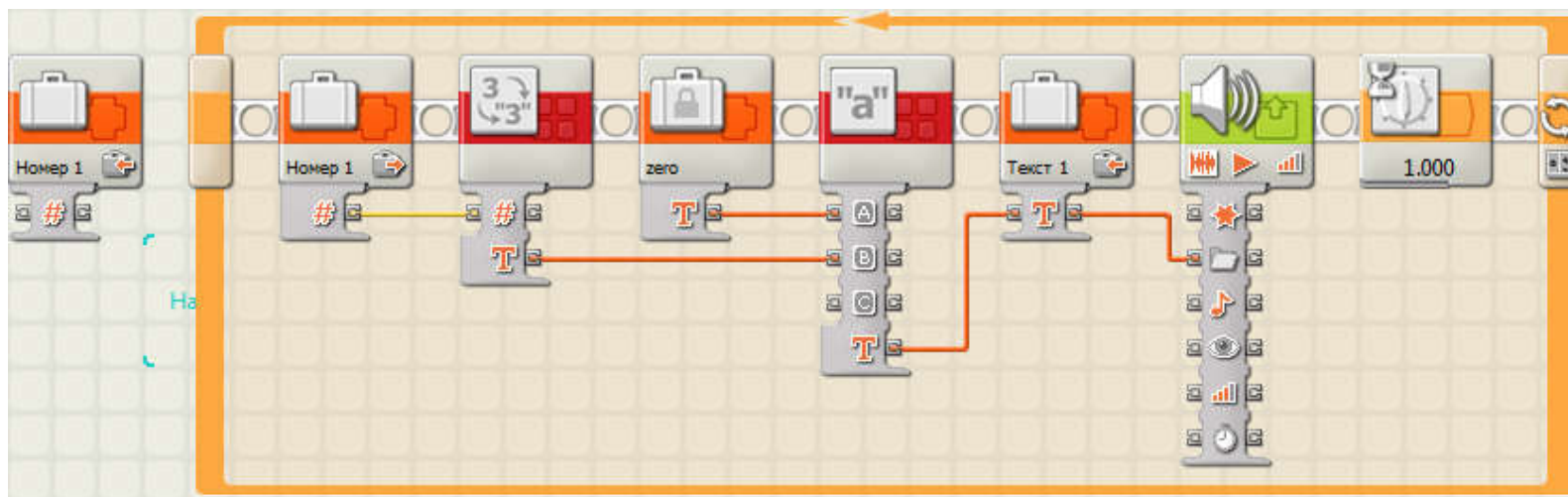
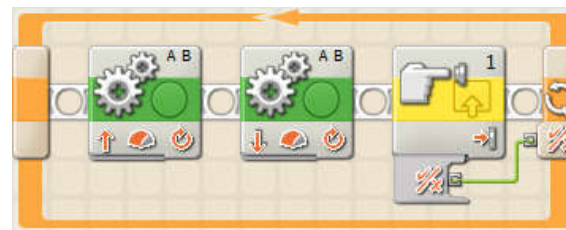
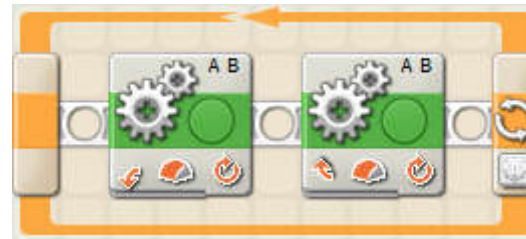


Циклический алгоритм

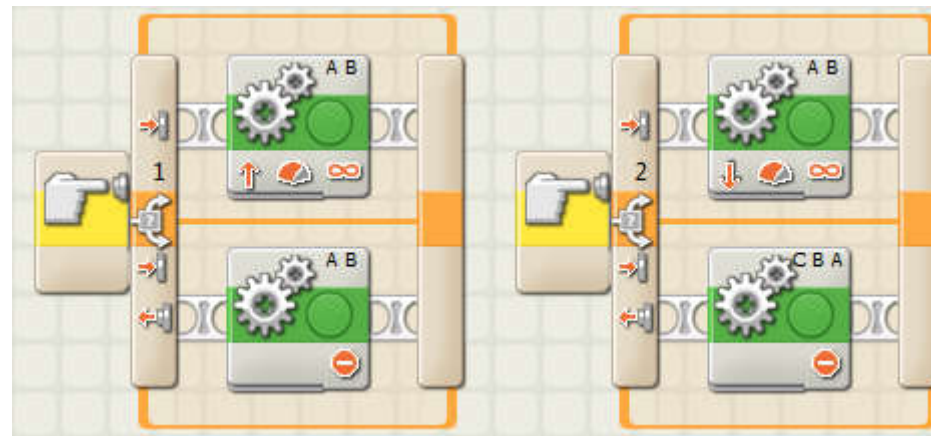
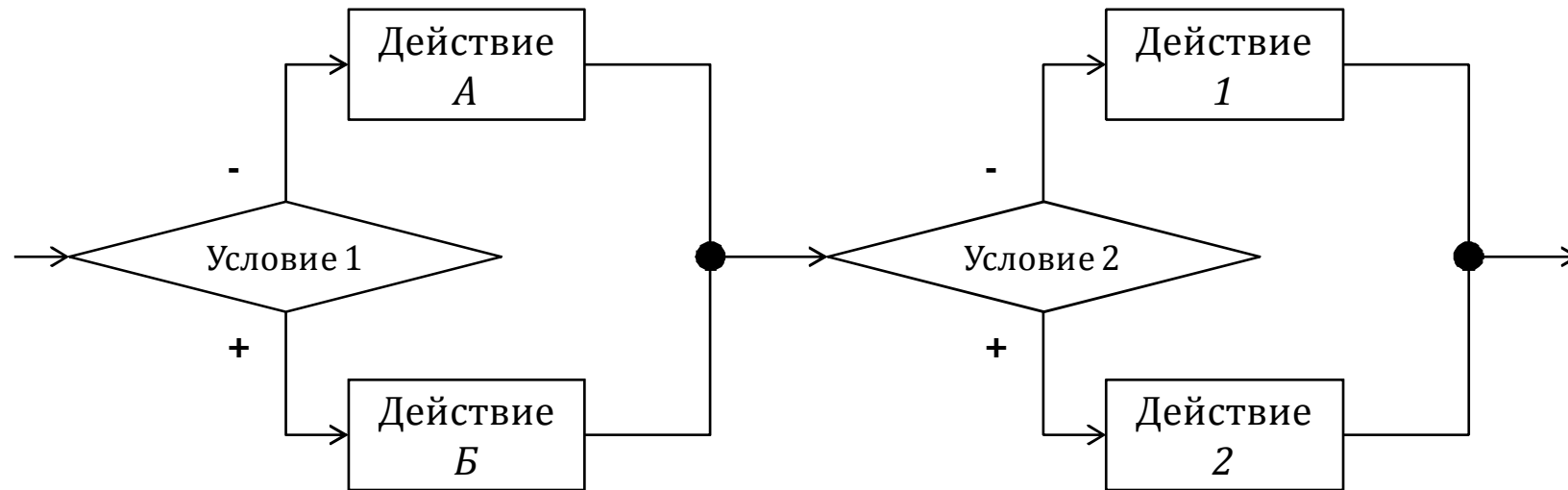


Условия цикла

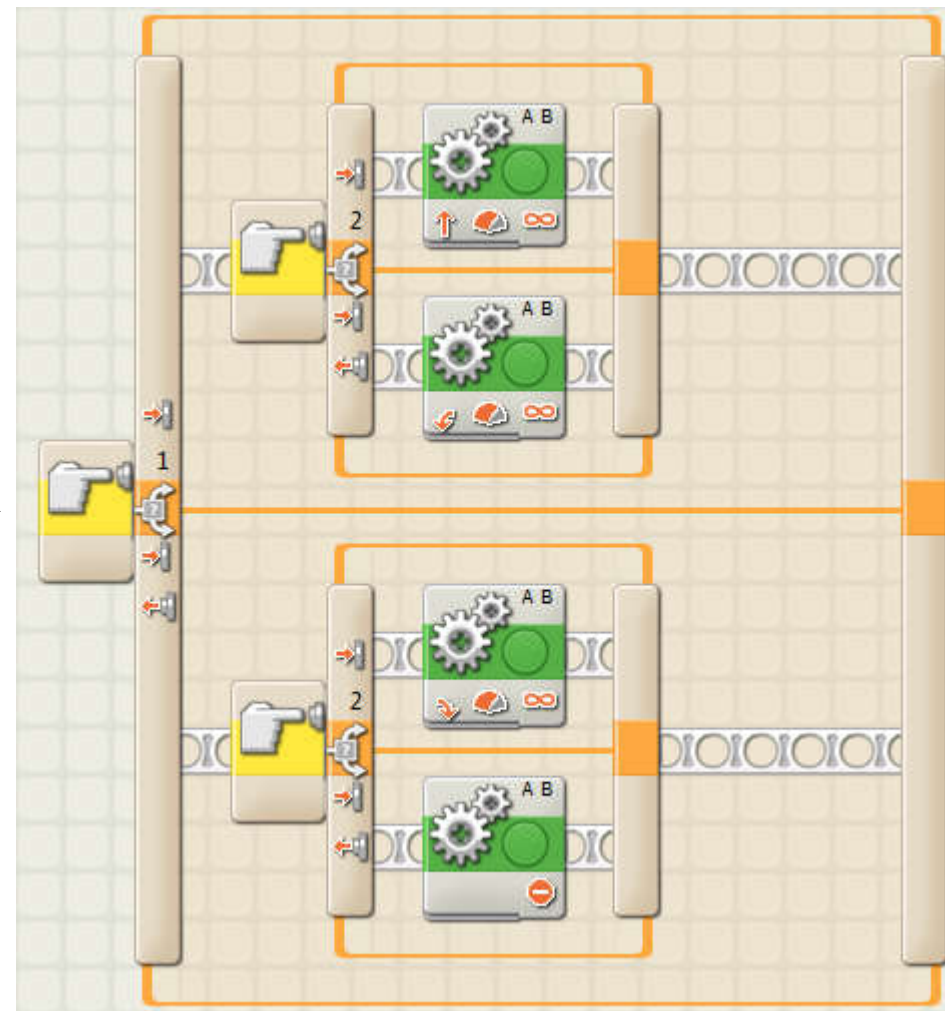
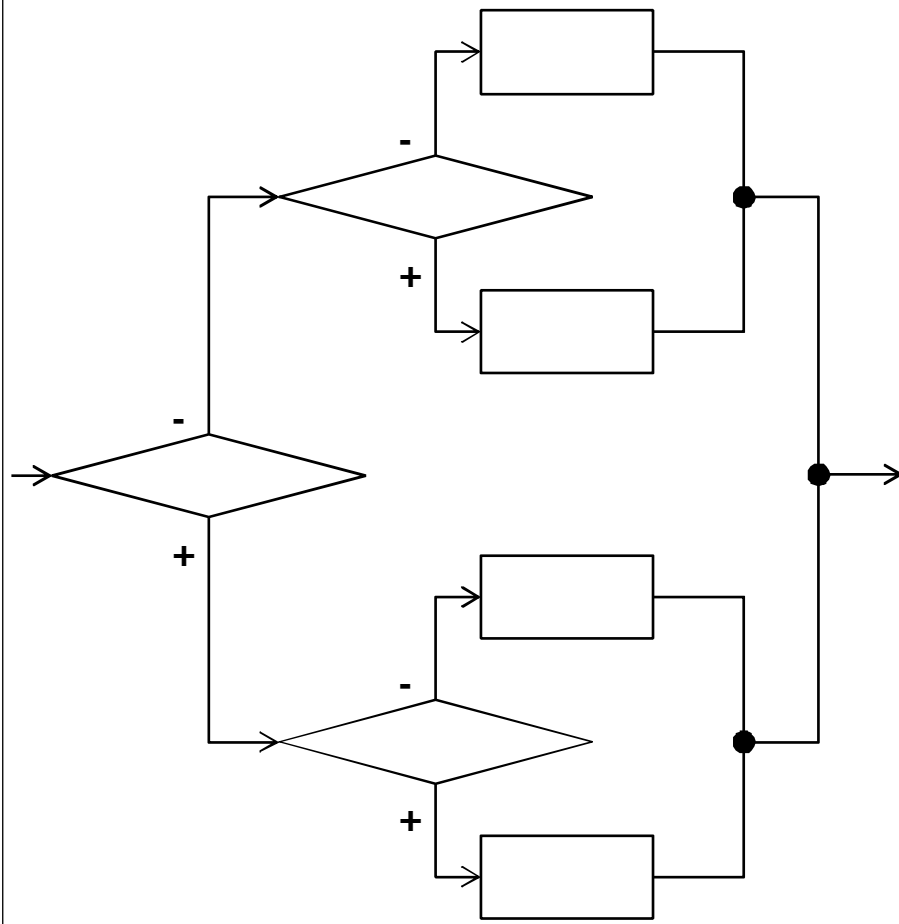
- От времени
- От логического значения
- От счетчика
- От показаний датчика
- Бесконечный цикл



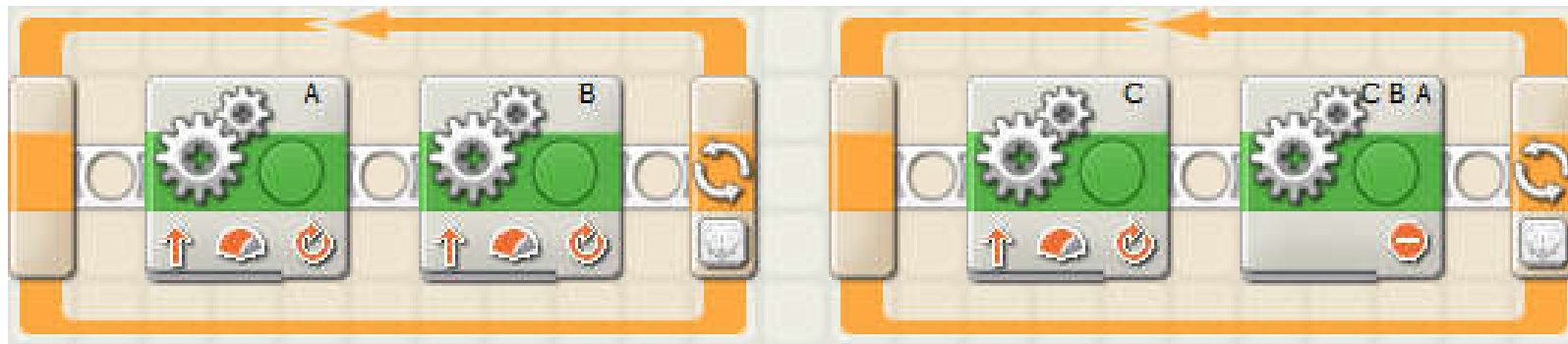
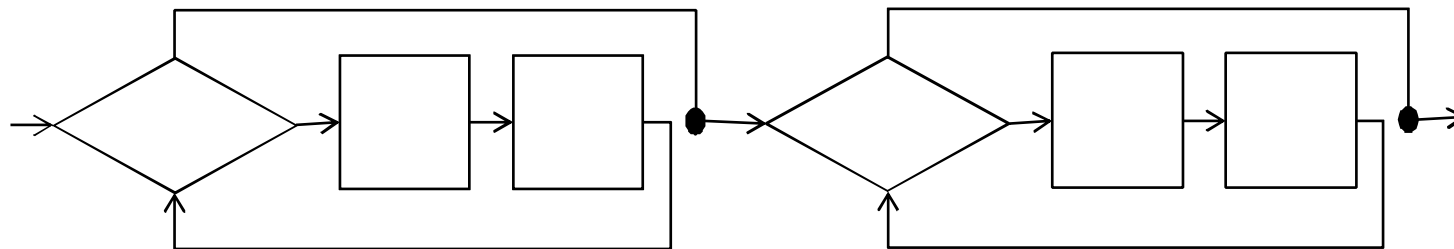
Последовательные ветвления



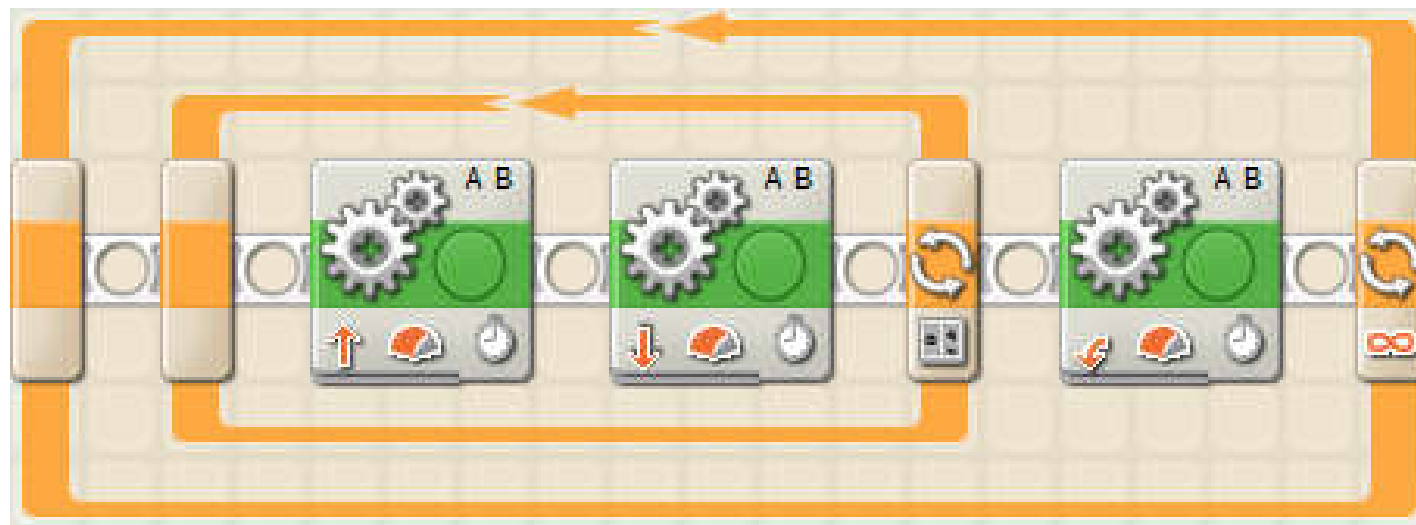
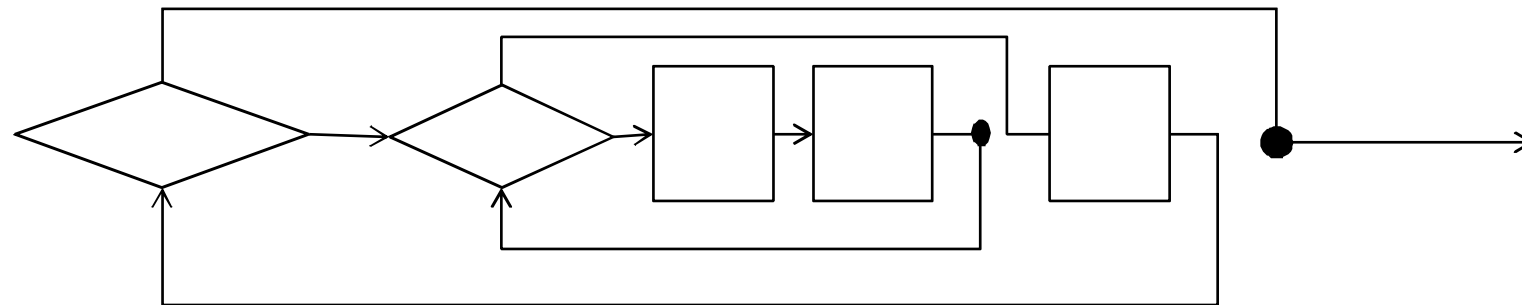
Вложенное ветвление



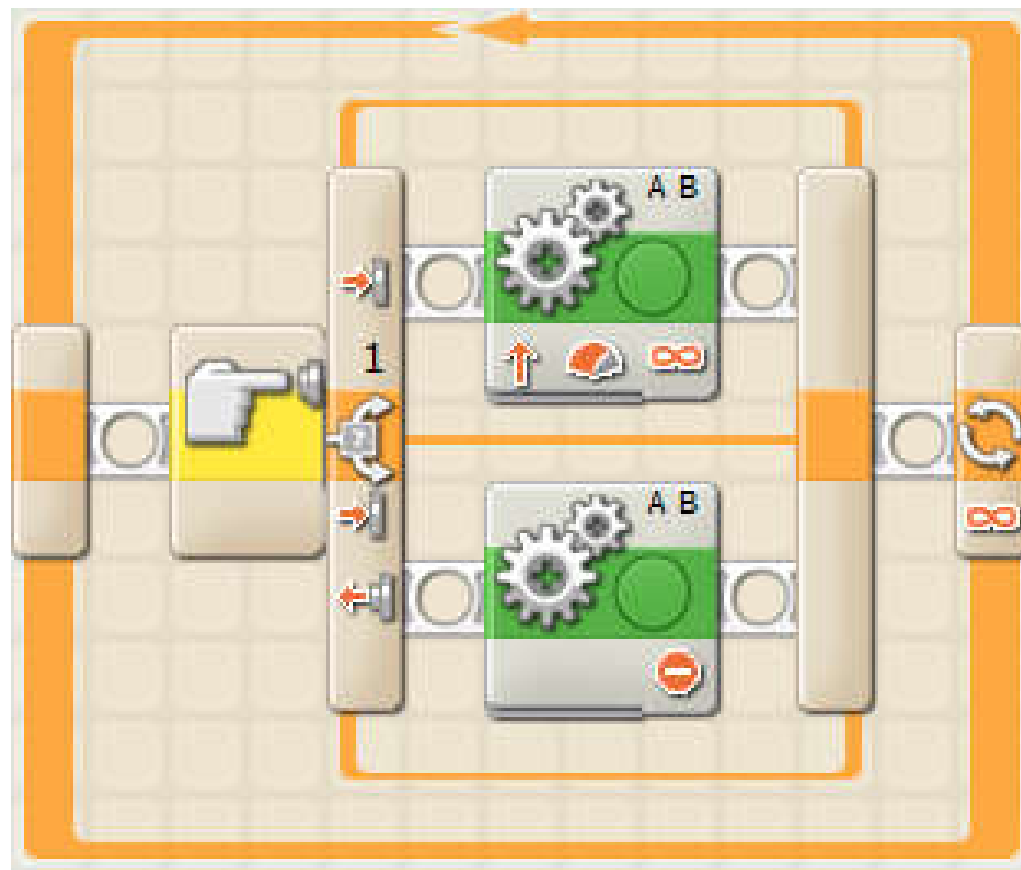
Последовательные циклы



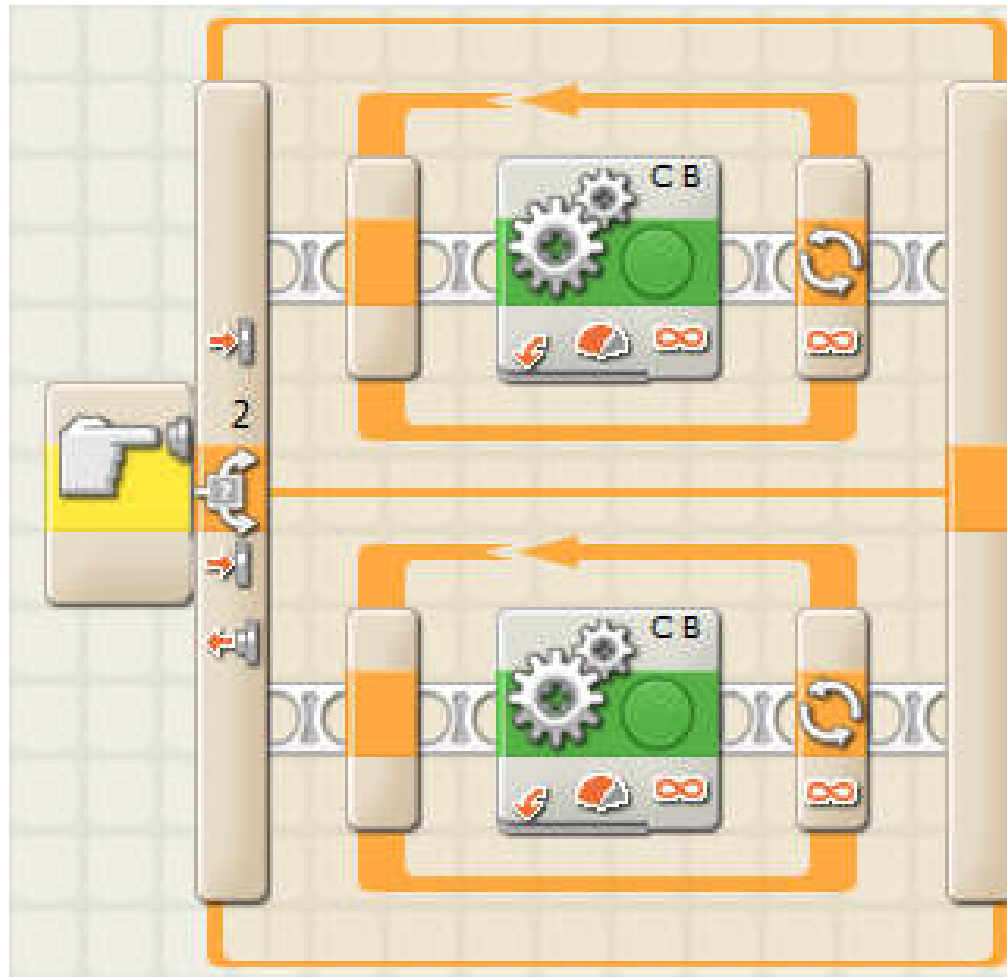
Вложенные циклы



Ветвление в цикле



Циклы во ветвлении



Робот-сортировщик
Создание, Программирование
и
Использование

Цели

- «Материализация» (овеществление) алгоритмов
 - перемещения объектов в памяти
 - поиска
 - сортировок

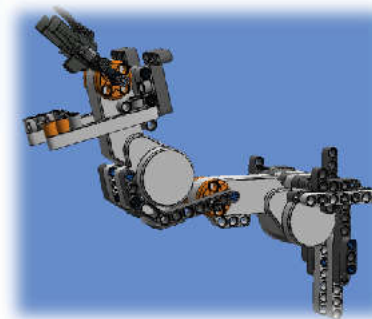
Задачи конструирования

- Конструирование робота

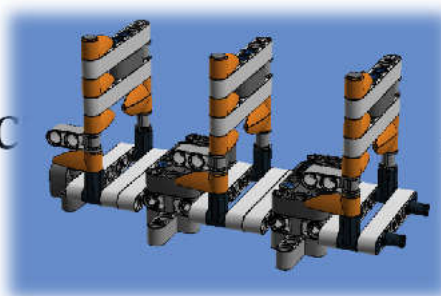
- Движущая часть плат



- Манипулятор

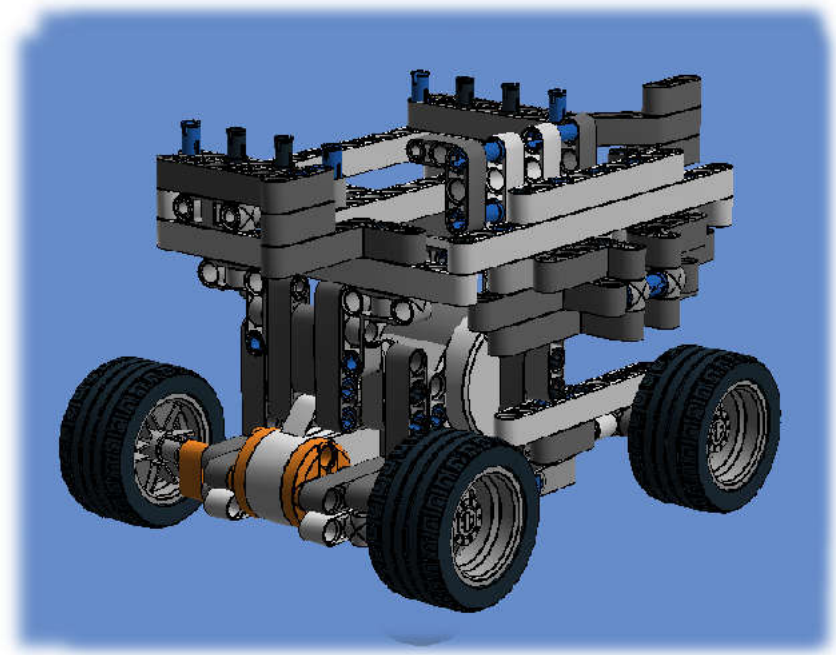


- Конс ячеек памяти

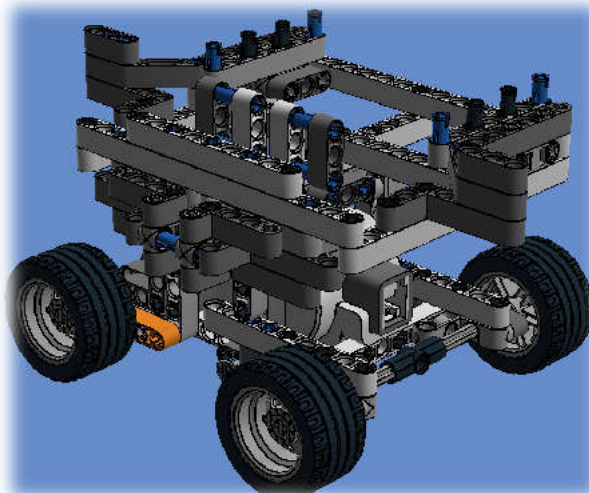
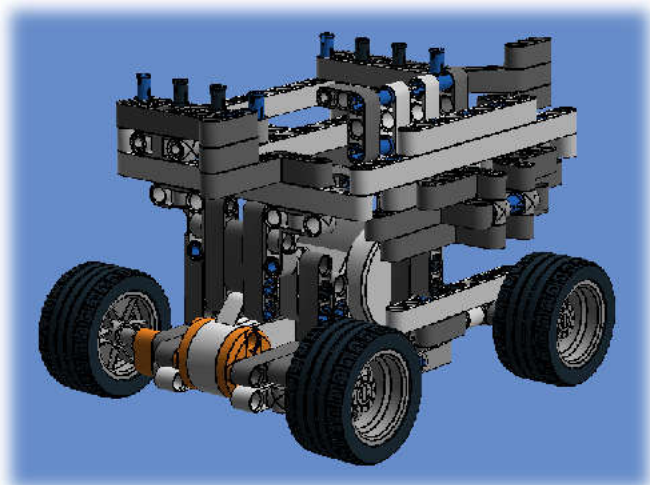
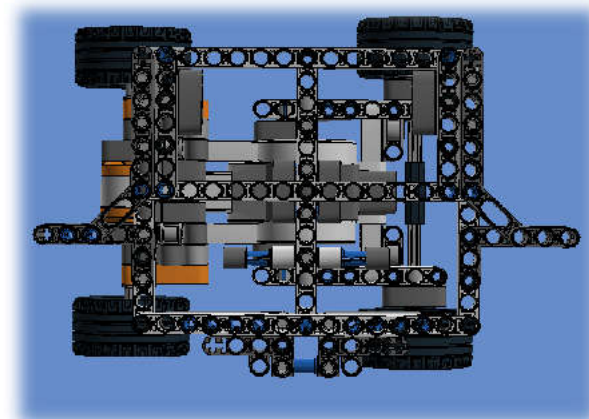
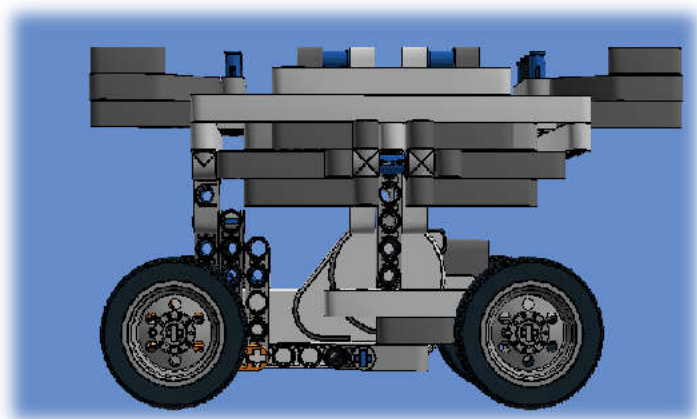


Требования к конструкции подвижной платформы

- Сбалансированность
- Жесткость каркаса
- Устойчивость к вибрациям
- Компактность
- Модульность

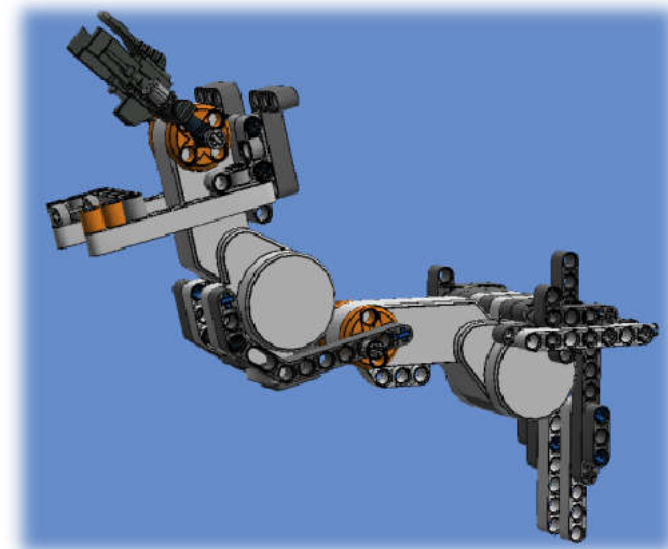


Подвижные платформы

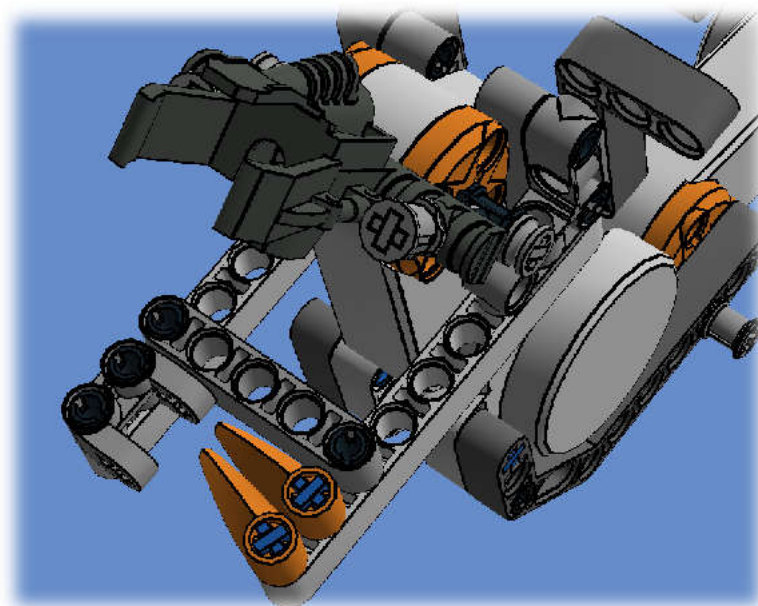
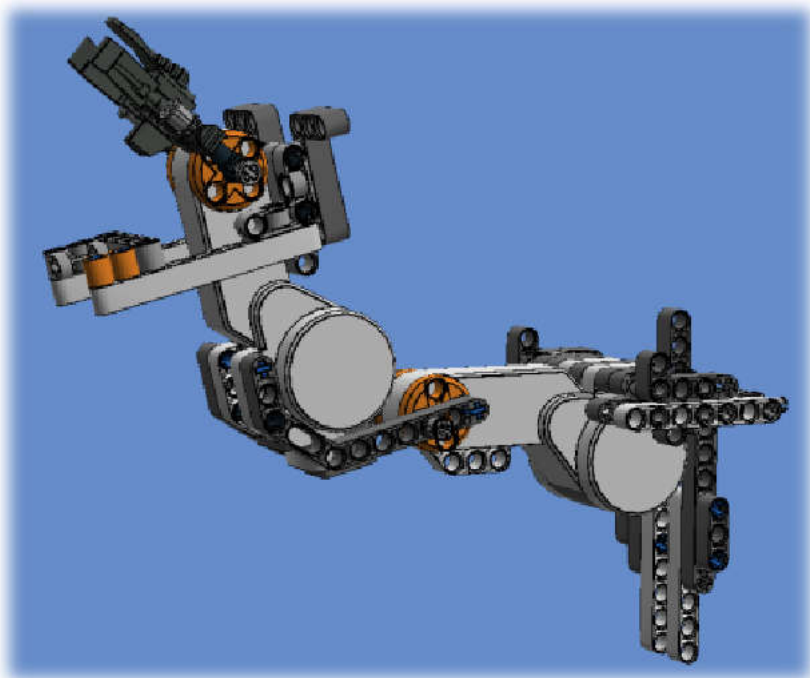


Требования к конструкции манипулятора

- Сбалансированность
- Устойчивость к вибрациям
- Компактность
- Надежность соединений
- Подвижность
- Соответствие устройства захвата ячейкам и объектам

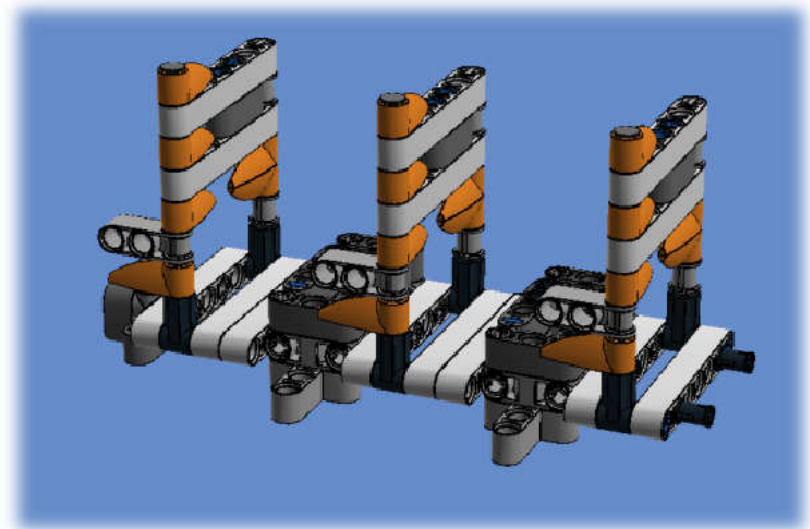


Манипулятор



Требования к конструкции ячеек

- Модульность
- Потенциально-
бесконечность
- Экономия деталей
- Сохранность объекта
внутри
- Доступность сортировщиком
- Распознаваемость



Требования к функциональности робота

- **Аккуратность и плавность движений**
- Способность определять ячейки
- Способность определять объект
- Способность определять характеристики объекта
- **Манипуляции с объектом**
 - Захват объекта
 - Движение с объектом
 - Размещение объекта в ячейке
- **Определение ошибки**

Плавность движений. Необходимость.

- Уменьшение вибраций при движении – уменьшение ошибок в действиях
 - Увеличение точности движения
 - Уменьшение вероятности неправомерного позиционирования объекта при размещении в ячейке
 - Уменьшение влияния движений на другие функции
- Уменьшение затрат энергии на движение

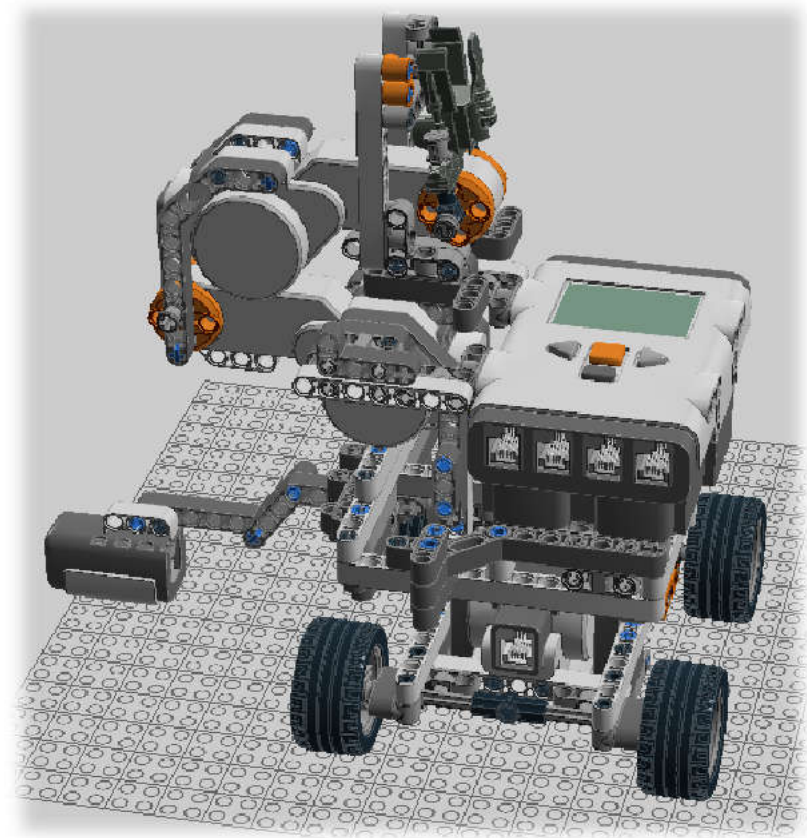
Плавность движений.

Общий алгоритм

1. Определить текущую и конечную позиции
2. Определить направление движения
3. Задать мощность движения минимально возможной
4. Пока текущая позиция не совпала с конечной
 - 1) Сдвинуться на одну позицию (переместится на минимально допустимую величину отличную от нуля)
 - 2) Если текущая позиция не приблизилась ко конечной
 - Увеличить мощность движения
 - Иначе
 - Уменьшить мощность движения
5. Зафиксировать позицию

Стартовое положение

- Манипулятор прижат к роботу
- Захват сомкнут
- Приведение к стартовому положению из любого другого подразумевает движение манипулятора к роботу и закрытие захвата до стартового положения



Захват объекта. Общий алгоритм.

1. Приведение к стартовому положению
2. Открытие захвата
3. Перемещение манипулятора в ячейку
4. Закрытие захвата
5. Прижать манипулятор к роботу

Размещение объекта в ячейке. Общий алгоритм.

1. Приведение к стартовому положению
2. Перемещение манипулятора в ячейку
3. Открытие захвата
4. Прижать манипулятор к роботу
5. Закрытие захвата

Определение ошибки. Общий алгоритм.

I. Определение невозможности выполнения движения

1. Запустить таймер
2. Обнулить таймер перед началом любого движения
2. Пока не закончится программа
 - 1) Если таймер превысил лимит
 - *1. Прервать движение
 - *2. Остановить и обнулить таймер
 - *3. Вернуться в стартовую позицию
 - *4. Внести корректировки позиции робота
 - *5. Повторить движение

II. Определение неудачного захвата

1. Оценить стартовое состояние ячейки
2. Произвести операцию размещения объекта
3. Оценить текущее состояние ячейки
4. Если текущее состояние ячейки совпадает со стартовым
Повторить с шага 2.