

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«Челябинский государственный педагогический университет»

Т.В. Никитина

**ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА
КАК НАПРАВЛЕНИЕ ИНЖЕНЕРНО-ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА УЧАЩИХСЯ**

Учебное пособие

Челябинск
2014

УДК 6(07):373.04(021)
ББК 74.263:74.200.585.01я73
Н 62

Никитина Т.В. Образовательная робототехника как направление инженерно-технического творчества школьников [Текст]: учебное пособие / Т.В. Никитина. – Челябинск: Изд-во Челяб. гос. пед. ун-та, 2014. – 169 с.

ISBN 978-5-906777-21-8

Учебное пособие содержит материалы, позволяющие студентам педвуза (будущим учителям информатики, технологии, физики, начальных классов) освоить предметную область «Образовательная робототехника» и эффективно использовать эти знания в своей профессиональной деятельности. Изложенный материал нацелен на раскрытие потенциала данной предметной области в российском школьном образовании.

Издание предназначено для организации самостоятельной внеаудиторной работы студентов и рекомендовано учителям школ, педагогам дополнительного образования, методистам, специалистам ИПКРО.

Рецензенты:

Беспаль И.И., канд. физ.-мат. наук, доцент

Девяткин О.Е., канд. техн. наук, доцент

ISBN978-5-906777-21-8

© Никитина Т.В., 2014

© Издательство Челябинского государственного педагогического университета, 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА I. ТЕХНОЛОГИИ И МЕТОДЫ ОБУЧЕНИЯ ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ	
§ 1. Робототехника как прикладная наука.....	8
§ 2. Образовательная робототехника в средней школе.....	15
§ 3. Методика обучения образовательной робототехнике	37
ГЛАВА II. УСТРОЙСТВО УЧЕБНОГО РОБОТА	
§ 4. Устройство управления учебного робота	42
§ 5. Сенсорная система учебного робота	44
§ 6. Исполнительная система учебного робота	66
§ 7. Модели учебных роботов на базе <i>LegoMindstormsNXT</i> и <i>EV3</i>	71
ГЛАВА III. РЕШЕНИЕ СТАНДАРТНЫХ ЗАДАЧ ДЛЯ УЧЕБНОГО РОБОТА	
§ 8. Создание программ для учебного робота.....	77
§ 9. Задача движения робота	89
§ 10. Следование по линии	97
ГЛАВА IV. СОРЕВНОВАНИЯ ПО ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ	
§ 11. Классические соревнования роботов.....	105
§ 12 Всемирная олимпиада по робототехнике.....	109
§ 13. Соревнования по правилам <i>FIRST</i>	111
§ 14. Методика организации тренерской работы учителя по подготовке школьной команды к соревнованиям по робототехнике	114
ЛАБОРАТОРНЫЙ ПРАКТИКУМ	
Лабораторная работа 1. Состав конструктора и программное обеспечение <i>Lego Mindstorm NXT</i>	119

Лабораторная работа 2. Движение робота по заданной траектории	122
Лабораторная работа 3. Управление поведением робота с помощью датчиков касания, звука, расстояния	126
Лабораторная работа 4. Управление движением робота с датчиком освещенности и датчиком цвета	129
Лабораторная работа 5. Создание программ для робота с использованием переменных	134
Лабораторная работа 6. Программирование <i>NXT</i> в графической среде <i>Robolab</i>	138
Лабораторная работа 7. Организация проектной деятельности школьников с набором <i>GreenCity</i> (Зелёный город)	143
Лабораторная работа 8. Кегельринг	148
Лабораторная работа 9. Изучение конструкций роботов	152
ЗАКЛЮЧЕНИЕ	154
ГЛОССАРИЙ	155
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	160
ПРИЛОЖЕНИЕ	164

ВВЕДЕНИЕ

Стремительное развитие научно-технического прогресса находит своё адекватное отражение в школьном обучении. Так в школе начинают изучаться новые перспективные предметные области, одной из которых является образовательная робототехника. Знания из данной предметной области востребованы сегодня как на профессиональном, так и на бытовом уровне, поскольку робототехника всё увереннее входит в жизнь человека. Роботы используются на производстве, в военном деле, в чрезвычайных ситуациях, в медицине и даже в быту. Во многих странах мира (США, Германия, Сингапур, Япония, Китай, Корея, Австралия и др.) наблюдается значительный рост интереса к образовательной составляющей данного научно-технического направления.

На сегодняшний день образовательная робототехника в российском образовании осваивается учащимися в школьных кружках, а также на элективных курсах посредством образовательных конструкторов: *Lego WeDo*, *Lego Mindstorms NXT*, *Lego Mindstorms EV3*, *Fischertechnik*, *Arduino*, *Roborobo*, *Bioid* и др. При этом конструктор *Lego Mindstorms NXT* является наиболее распространённым, поскольку позволяет решать широкий класс задач для школьников от 8 до 18 лет.

В основе образовательной робототехники лежат игровые технологии, этим в значительной степени обусловлена её популярность у школьников. В то же время образовательная робототехника – это интегративная предметная область, как и для любого интегративного курса для неё характерно соединение нескольких учебных предметов в единое целое и выделение ведущего предмета. Во всех известных нам разработках ведущим школьным предметом для изучения образовательной робототехники является информатика. Это может быть связано с тем, что знакомство учащихся с робототехникой начинается именно

с создания простейших программ-алгоритмов для робота на основе базовой модели (двухмоторной тележки). Данный учебный материал не вызывает особых затруднений у учителей информатики, поскольку является естественной сферой их профессиональной деятельности. Все же думаем, что среда графического программирования, предназначенная специально для изучения основ образовательной робототехники (например, *NXT-G* для *Lego Mindstorms NXT*), доступна и учителям физики, технологии, а также учителям начальных классов.

Образовательная робототехника новое для отечественного образования направление, в настоящее время она интенсивно развивается как в нашей стране, так и в мировой образовательной практике. В настоящее время происходит быстрое обновление робототехнических конструкторов (за период с 1998 по 2013 годы компания *Lego* трижды обновила состав робототехнического конструктора и программное обеспечение к нему), в сети Интернет неуклонно растет число информационных и учебных материалов по образовательной робототехнике, в то же время большая часть этих материалов представлена на английском языке и требует адаптации к российской образовательной системе. Поэтому методическая подготовка студентов педвуза (будущих учителей) по образовательной робототехнике должна быть направлена на повышение самостоятельности в ходе учебно-познавательной деятельности, стимулирование мотивов учения и самообразования.

Цель данного пособия – оказать помощь студенту (будущему учителю) или начинающему учителю в освоении данной предметной области. Работа с учебным пособием будет способствовать поддержке профессионального самообразования и личностного роста студентов, позволит проектировать дальнейший образовательный маршрут и профессиональную карьеру по направлениям:

1) организация внеурочной деятельности в школе и учреждениях дополнительного образования;

2) обучение образовательной робототехнике в школе: элективные курсы, встраивание элементов образовательной робототехники в предмет (информатика, физика, технология, окружающий мир (начальная школа));

3) подготовка обучающихся к олимпиадам по образовательной робототехнике;

4) культурно-просветительская деятельность (чтение научно-популярных лекций, проведение ознакомительных мастер-классов и др.);

5) профориентация школьников.

Учебное пособие содержит:

- теоретический материал, включающий информацию о современных достижениях прикладной науки робототехники, о возможностях демонстрации этих достижений средствами образовательного конструктора *LegoMindstormsNXT*;

- систему заданий для самостоятельной внеаудиторной работы студентов;

- лабораторный практикум на оборудовании *Lego-MindstormsNXT* и стандартном программном обеспечении к нему;

- систему контрольных вопросов к лабораторным работам, позволяющую понять интегративный характер данной предметной области;

- список литературы (печатные издания и Интернет-источники) для более глубокого изучения материала.

ГЛАВА I

ТЕХНОЛОГИИ И МЕТОДЫ ОБУЧЕНИЯ ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ

§ 1. РОБОТОТЕХНИКА КАК ПРИКЛАДНАЯ НАУКА

1.1. Робототехника в системе наук

Опираясь на самые современные достижения науки и техники, робототехника развивается непрерывно и стремительно.

Робототехника – область техники, связанная с разработкой и применением роботов, а также компьютерных систем для управления ими, сенсорной обратной связи и обработки информации [30]. Роботы и робототехнические системы предназначены для выполнения рабочих операций от микро- до макро-размерностей, в том числе с заменой человека на тяжелых, утомительных и опасных работах. Существует много типов робототехнических устройств: роботы-манипуляторы, мобильные роботы, шагающие роботы, средства помощи инвалидам и др.

Робот представляет собой машину автоматического действия, которая объединяет в себе свойства машин рабочих и информационных, являясь таким образом принципиально новым видом машин [47]. К функциям, которыми в первую очередь должен обладать робот, можно отнести следующие четыре. Первая – функция манипулирования и перемещения. Манипулируя предметами, робот должен выполнять полезную работу и обеспечивать свое перемещение. Второй функцией является функция определения состояния внешней среды. Робот должен обладать органами чувств, позволяющими ему «видеть», «осознать», «чувствовать» силу, «распознавать» объекты и др. К третьей функции относится функция осмысления и принятия решения. Она даёт возможность планировать последователь-

ность операций, необходимых для достижения цели, заданной человеком, а также для осуществления соответствующего управления. И наконец, четвертой функцией является функция диалога, при реализации которой обеспечивается эффективная связь человека с роботом. С помощью комплексной реализации этих четырех функций можно создать работа [13].

При создании первых роботов вплоть до наших дней образцом для них служат возможности человека. Именно стремление заменить человека на тяжелых и опасных работах породило идею создания робота. Своим появлением роботы обязаны, в частности, компьютеризации производства, автоматизации технологических процессов, а также огромному опыту, накопленному в процессе эксплуатации станков для механической обработки с числовым программным управлением [19].

Возникнув на основе кибернетики и механики, робототехника в свою очередь породила новые направления развития самих этих наук. В кибернетике это связано прежде всего с интеллектуальным управлением и бионикой как источником новых, заимствованных у живой природы идей, а в механике – многостепенными механизмами типа манипуляторов [47].

Робототехника призвана заниматься вопросами построения технических систем, функционально эквивалентных некоторым из важнейших систем человеческого организма. Задачи, стоящие перед робототехникой, естественным образом определяют направления её развития. Так созданием и использованием искусственных чувств робототехника занимается такой раздел робототехники как сенсорная технология, а обучением робота элементам мыслительной деятельности человека – искусственный интеллект и информатика. Робототехника в качестве самостоятельного раздела включает область механики, связанную с разработкой механических конечностей. Раздел робототехники, имеющий дело с конструированием искусственных рук или ног робота для выполнения конкретных

производственных операций, а также приспособлением технологии производства изделий к возможностям конечностей конкретного робота, получил название промышленная робототехника. Из всех разделов робототехники промышленная робототехника развивается самыми быстрыми темпами. Именно здесь достигнуты наиболее значительные промышленные результаты: внедрение промышленных роботов позволило автоматизировать многие промышленные операции [19].

Робототехника занимается также изучением вопросов имитационности человека и животного свободно перемещаться в любом направлении. Сюда относятся не только проблемы построения мобильных роботов-тележек на колесах, но и проблемы создания шагающих роботов, а также технических устройств, способных перемещаться подобно змее и морским моллюскам. Последнее свидетельствует о том, что робототехника является развивающейся дисциплиной. Многие подходы известны уже сегодня, но вряд ли кто-нибудь может сказать, какие технологии будут использоваться в дальнейшем [19; 47].

Наряду с робототехникой часто употребляют термин «мехатроника». Появился он примерно в 80-х годах XX в. в известном смысле как развитие понятия «электромеханика». Различие этих, на первый взгляд тождественных, понятий заключается в разном уровне используемых электронных решений. Мехатроника предполагает непосредственное внедрение микроэлектронных систем в систему управления, в некотором роде синтез электроники и механики, позволяющий создавать эффективные регуляторы. В дальнейшем термин «мехатроника» был существенно обобщен, в результате чего к мехатронным системам стали относить практически любые сложные технические системы, содержащие механическую и электронную части и управляемые компьютером [8]. Мехатронику как научную дисциплину можно определить следующим образом:

область науки и техники, основанная на системном объединении узлов точной механики, датчиков состояния внешней среды и самого объекта, источников энергии, исполнительных механизмов, усилителей, вычислительных устройств (ЭВМ и микропроцессоры) [42].

Мехатроника и робототехника имеют много точек соприкосновения. Обе дисциплины основаны на применении электронных элементов и механических деталей и узлов. Общеизвестным является факт, что роботы являются мехатронными системами [6;8;13;42]. Поэтому термин «мехатроника», конечно, шире. К этой области относятся также системы, которые не являются робототехническими, например: системы управления комплексами технологического оборудования, обрабатывающие центры, системы поддержания заданной температуры в помещении. Функционирование таких систем связано с выполнением разнообразных измерений, для чего их также оснащают соответствующими сенсорными устройствами [7].

1.2. История развития робототехники

Современными предшественниками роботов явились различного рода устройства для манипулирования на расстоянии объектами, непосредственный контакт с которыми для человека невозможен или опасен. Впервые манипуляторы были созданы в 1940–1950 гг. для атомных исследований, а затем и для атомной промышленности. Подобные устройства стали применяться в глубоководной технике, металлургии и ряде других отраслей промышленности [47].

Первый промышленный робот *Unimate* был выпущен в конце 50-х годов фирмой *Unimation* (Д. Деволом совместно с Д. Энгельбергером). Этот робот содержал контур обратной связи по положению, в котором перемещения звеньев манипулятора измеряли установленные в них датчики. Датчики

углового и линейного перемещения и сегодня составляют основу информационного обеспечения робототехнических и мехатронных систем.

В середине 60-х годов стало очевидно, что гибкость программируемых роботов может быть повышена при использовании систем осязания, основанных на применении датчиков среды. Первая система такого типа – тактильная – была разработана Х. Эрнстом в рамках проекта «*Mechanical Hand-1*». Она позволяла роботу укладывать кирпичные блоки без помощи оператора. Наконец, в начале 70-х годов по проекту «*Stanford Arm*» группа Р. Пола создала мультимодальную информационную систему «глаз – ухо – рука», содержащую тактильные, локационные и визуальные датчики. Эти разработки заложили основы для использования информационных средств в автоматическом и роботизированном производствах [8].

Сегодня развитие робототехники подходит к важному этапу – развитию искусственного интеллекта. «Интеллект», заключенный в роботе, принимает одну из двух форм: программно поддерживаемый интеллект (экспертная система) и интеллект в форме нейронной сети [16].

Экспертные системы основаны на системе решающих правил. Это системы-решатели, системы-консультанты определенной предметной области, заключающие в себе знания специалиста-эксперта [38].

Нейронными сетями называют искусственные компьютерные системы (на базе аппаратного программного обеспечения), которые функционируют и «обучаются» на основе моделей, созданных по аналогии с биологическими системами человеческого мозга. Моделирование по образцу биологических мозговых структур привело и к успешному решению некоторых частных проблем, необходимых для создания искусственного интеллекта, таких как машинное зрение и распознавание речи. Таким образом, использование технологий нейросетей позво-

ляет создавать системы контроля функций робота, превосходящих возможности стандартных ЦПУ и программного обеспечения [30].

Таким образом, каждое последующее поколение роботов обладает большими возможностями и совершенством, но не исключает предыдущего; они взаимно дополняют друг друга и находят применение соответственно своим функциональным возможностям и условиям экономической целесообразности.

1.3. Классификация роботов

По назначению роботы делятся на промышленные, поисковые, военные, бытовые, исследовательские.

Промышленные роботы наиболее широко распространены. По характеру выполнения технологических операций промышленные роботы делятся на технологические, вспомогательные и универсальные. Технологические роботы непосредственно участвуют в технологическом процессе в качестве оборудования: гибка, сварка, окраска, сборка и др. Вспомогательные (подъемно-транспортные) роботы выполняют функции переноса объекта в вертикальной и горизонтальной плоскостях. Они применяются для обслуживания основного технологического оборудования. Универсальные роботы выполняют разнообразные технологические операции – основные и вспомогательные.

Поисковые роботы применяются для космических исследований, при чрезвычайных ситуациях. Достаточно широк спектр применения военных роботов: для наземной и воздушной разведки, для наземных и воздушных тактических операций, для космического базирования, подводные работы.

Бытовые роботы все увереннее входят в жизнь человека. Сегодня на рынке представлены роботы-игрушки, роботы для обслуживания детей, престарелых людей и др.

Исследовательские роботы служат для поиска, сбора, переработки и передачи информации об исследуемых объектах.

К числу таких объектов относятся космическое пространство, поверхности планет, подводное пространство, подземные полости (шахты, пещеры и др.), Арктика и Антарктика, пустыни, зараженная местность и другие труднодоступные для человека области. Примерами таких роботов являются беспилотные летательные аппараты, роботы-саперы и др.

По виду и конструкции механики выделяют роботы-манипуляторы, роботы-тележки (или колесные роботы), шагающие роботы (насекомоподобные, гуманоиды и др.), специализированные роботы (роботы-футболисты, космические аппараты, летающие роботы и др.), нанороботы. Развитие нанотехнологий оказало существенное влияние на всю технологию производства роботов как микроскопических, так и обычных, макроскопических, спектр возможностей которых неизмеримо расширился, начиная от задач уборки помещений и заканчивая автоматизированным производством продукции. С нанотехнологиями связаны большие надежды в создании роботов, имеющих микроскопические размеры. Такие роботы будут способны перемещаться по кровотоку, удаляя холестериновые бляшки, отыскивать раковые опухоли и удалять в них все пораженные клетки, бороться с процессами старения в организме и др.

По принципу управления роботы делятся на программные, адаптивные, интеллектуальные. Программные роботы работают по жесткой, заранее заложенной программе; допускается изменение порядка выполнения программы. Адаптивные роботы приспосабливаются к изменениям внешней среды (рабочей зоны), имеют сенсорные датчики, при этом программы адаптации должны быть заранее составлены. Интеллектуальные роботы отличаются тем, что конечная цель операции заранее неизвестна, но известен набор объектов с их характеристиками и алгоритм работы с каждым объектом, имеется решающее правило, которое позволяет перепланировать действия.

По способу организации управления роботы бывают автономные и дистанционно управляемые. Автономные роботы работают по программе, без участия человека. Дистанционно управляемые роботы управляются с участием человека (с помощью задающих рукояток, традиционно – трёхступенных).

Задания для самостоятельной внеаудиторно работы

Подготовьте презентации по темам:

1. Перспективные направления развития робототехники;
2. Робототехника и кибернетика;
3. Робототехника и бионика;
4. Применение роботов;
5. Промышленные роботы;
6. Космические роботы;
7. Военные роботы;
8. История развития робототехники;
9. Искусственный интеллект и робототехника.

§ 2. ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА В СРЕДНЕЙ ШКОЛЕ

2.1. Образовательная робототехника как предметная область в средней школе

Образовательная робототехника представляет собой дидактическую модель робототехнической науки. Элементы этой модели не являются научным и инженерно-техническими знаниями в области роботостроения и могут быть использованы для организации пропедевтического обучения школьников основам инженерной деятельности с целью привлечения их интереса к инженерно-техническим специальностям. Круг задач, решаемых образовательной робототехникой, достаточно широк, поскольку робот может выступать не только объектом для изу-

чения, нои средством учебного моделирования и конструирования. Во втором случае открываются широкие возможности для встраивания образовательной робототехники в различные школьные предметы.

Образовательная робототехника как направление учебно-познавательной деятельности пользуется высоким познавательным интересом у школьников. Изучение робототехники в школе осуществляется посредством образовательных конструкторов: *LegoWeDo*, *Lego Mindstorms NXT*, *Lego Mindstorms EV3*, *Tetrix*, *Matrix*, *Fischertechnik*, *Arduino*, *Roborobo*, *Bioloid* и др. В основе образовательной робототехники лежат игровые технологии, этим в значительной степени обусловлена её популярность. Безусловно, игра является эффективным методом и формой организации обучения, она позволяет школьникам учиться, не замечая процесса обучения.

В то же время образовательная робототехника – это интегративная предметная область, отражающая современный уровень развития науки и техники. Она включает в себя знания из школьных предметов: информатики, физики, математики. Информатика как ведущий учебный предмет сохраняет свою специфику, а физика и математика выступают в качестве вспомогательной основы. Можно выделить два вида интегративных связей образовательной робототехники с названными учебными предметами: 1) элементы *предметных* знаний, необходимые для изучения робототехники; 2) элементы *межпредметных* знаний, необходимые для изучения робототехники. Раскроем эти связи (рис. 1).

Создание и отладка алгоритмов для робота – задача из курса информатики. В то же время программирование устройств (моторов и датчиков), которыми оснащен робот, затрагивает и область физики. При создании программ необходимо понимать суть работы датчика (физические закономерности, на которых основана его работа), учитывать погрешности измерения датчи-

ка и др. Физика всегда занимала ведущее место как научная основа техники, поскольку она лежит в основе всех наиболее значимых направлений технического прогресса. Для образовательной робототехники особо важными разделами физической науки являются механика и электроника. Математика как инструмент научного познания позволяет в образовательной робототехнике решать задачи с углами, градусами, коэффициентами и пропорциями. В сумме физические и математические знания дают возможность рассчитывать траекторию движения робота, измерять и рассчитывать значения физических величин. Наконец, в совокупности с информатикой математика позволяет создавать достаточно сложные алгоритмы для робота с использованием переменных величин и математических вычислений.



Рис. 1. Интегративные связи образовательной робототехники

Разъясним изложенное на примере проекта по созданию робота, рисующего многоугольник. С помощью несложного циклического алгоритма (ехать прямо, повернуть (действия повторяются многократно)) роботу задается траектория движения; в зависимости от заданного угла поворота робот может двигаться по траектории треугольной, квадратной, пятиугольной форм и др. При создании такого робота используются следующие дидактические понятия: дроби, пропорции, окружность, радиус, диаметр (математика); движение, скорость тела, пройденный путь, механическая энергия, теплота, трение (физика); линейный алгоритм, циклический алгоритм, программа, отладка программы (информатика). При создании программы для робота, рисующего многоугольник, ключевым моментом является вычисление угла, на который поворачивается робот. Совокупность физических и математических знаний в решении этой задачи делает возможным вывод формулы для расчета длительности работы мотора, этот параметр закладывается в программу.

Характеризуя образовательную робототехнику как интегративный курс для средней школы, можно выделить целевой, содержательный, деятельностный, воспитательный, развивающий аспекты её преподавания.

Целевой аспект: образовательная робототехника рассматривается как средство реализации ФГОС общего образования [41], проектная деятельность на занятиях по образовательной робототехнике способствует эффективному формированию у школьников всего комплекса универсальных учебных действий (познавательных, регулятивных, личностных, коммуникативных);

Содержательный аспект: в ходе изучения образовательной робототехники у учителя появляется возможность эффективной реализации межпредметных связей по основным школьным предметам «Информатика», «Физика», «Математика» (рис. 2). Нельзя не отметить и межпредметные связи образовательной

робототехники с биологией. Так, зачастую биологические механизмы сенсорных и двигательных функций живых организмов являются прототипам сенсорных и двигательных систем робота [19].

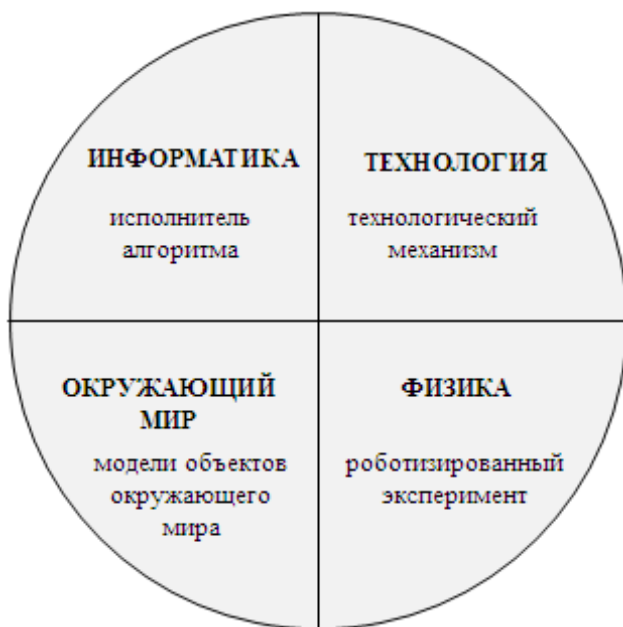


Рис. 2. Робот как средство обучения

Если рассматривать робота как средство обучения какому-либо предмету, то целесообразно включать элементы образовательной робототехники в уроки по информатике, технологии, физике и окружающему миру (начальная школа) [20].

Проекты роботов, предлагаемых *LegoMindstorms*, могут использоваться на уроках школьного курса «Технология» в рамках направления «Технический труд» по темам «Машины и механизмы. Графическое представление и моделирование» (механизмы технологических машин, сборка моделей технологических машин из деталей конструктора по эскизам и чертежам)

и «Электротехнические работы» (устройства с элементами автоматики, электропривод, простые электронные устройства)[11]. В начальных классах робототехника может использоваться на уроках по окружающему миру. Работая с роботизированными моделями, младшие школьники воссоздают жизненные ситуации и объекты окружающего мира наиболее приближенно к реальной действительности, и, следовательно, лучше осваивают результаты в данной предметной области[6]. На уроках информатики робот выступает реальным исполнителем созданного учащимся алгоритма. На уроках физики возможно применение роботизированного эксперимента, когда из деталей робототехнического конструктора собирается демонстрационная или лабораторная установка, т.е. конструктор используется как измерительная система с обработкой и фиксацией результатов. При этом возможна интеграция оборудования кабинета физики и робототехнического оборудования [11].

Деятельностный аспект связан с освоением в рамках курса образовательной робототехники видов деятельности, присущих предметам естественнонаучного цикла: систематическое наблюдение, выдвижение гипотезы, прогнозирование, сбор и интерпретация данных, анализ полученных результатов, формулировка выводов и др. Ведущим методом при обучении школьников образовательной робототехнике является метод проектов, ориентированный на самостоятельную деятельность учащихся – индивидуальную, парную, групповую, которую учащиеся выполняют в течение определенного отрезка времени.

Воспитательный аспект образовательной робототехники связан как с профориентационной функцией курса (на занятиях представляются образцы инженерной деятельности), так и с культурологической (знания по робототехнике как «значимые формы социокультурного опыта человечества»). Это зависит от «глубины» освоения курса. Думаем, что изучение образовательной робототехники на базовом уровне имеет важное культуроло-

гическое значение, поскольку учащиеся должны не только обладать основополагающими знаниями в классических дисциплинах, но и уметь ориентироваться в новых реалиях, одной из которых является тенденция к повсеместному распространению роботов и управляемых встраиваемых систем. Робототехника на сегодняшний день является одним из ключевых направлений научно-технического прогресса. Изучение образовательной робототехники на углублённом уровне предполагает решение достаточно сложных задач, например, использование ПИД-регулятора в программировании движения робота по лабиринту и по черной линии. Такой курс может рассматриваться как пропедевтический курс (от греческого *«propaideuo»* – «обучаю предварительно»), представляющий школьникам образцы инженерной деятельности, готовящий их к выбору будущей профессии.

Развивающий аспект образовательной робототехники заключается в том, что синтез конструирования и программирования в одном курсе позволяет решать задачи развития у обучающихся психических познавательных процессов (восприятия, мышления и речи, памяти, воображения), развитие форм мышления (анализ, синтез, сравнение и др.), развитие качеств личности (поведение и поступки, интеллектуальные, особенности, организационно-волевые качества, творческий потенциал и др.).

Таким образом, образовательная робототехника как интегративный курс обладает значительным потенциалом в школьном обучении, отвечая требованиям современного производства, способствуя углублению и систематизации знаний учащихся по основным школьным предметам, позволяя сориентироваться в выборе будущей профессии. С помощью многосторонних межпредметных связей образовательной робототехники с базовыми школьными предметами задачи обучения, развития и воспитания учащихся решаются на качественно новом уровне, закладывается фундамент для комплексного подхода в решении сложных проблем реальной действительности.

2.2. Оборудование для изучения образовательной робототехники

Изучение робототехники в школе осуществляется посредством образовательных конструкторов: *Lego WeDo*, *LegoMindstorms NXT*, *Lego Mindstorms EV3*, *Tetrix*, *Matrix*, *Fischertechnik*, *Arduino*, *Roborobo*, *Bioloid* и др. Ниже приведено краткое описание данного оборудования.

2.2.1. LegoWeDo

Данный конструктор предназначен в первую очередь для начальной школы (2–4 класс). Позволяет строить модели машин и животных, программировать действия моделей («поведение»). Конструктор содержит более 150 деталей, важнейшими из них являются: коммутатор (для управления датчиками и моторами при помощи программного обеспечения *WeDo*), мотор, датчик наклона, датчик движения [12; 26].



Рис. 3. Образовательный конструктор *Lego WeDo*



Рис. 4. Модель учебного робота «Аллигатор»

2.2.2. Индустрия развлечений. ПервоРобот RCX

Первая версия конструктора «Индустрия развлечений. ПервоРобот» (на базе микрокомпьютера *RCX*) была представлена компанией *LEGO* в 1998 г. Конструктор объединял в себе не только обычные для *LEGO* кубики, пластины, колёса и шестерёнки, но и микрокомпьютер в виде небольшого ярко-жёлтого «кирпича»; этот набор получил название *Mindstorms*. Программируемый блок, названный *RoboticControlExplorer (RCX)*, позволял подключить к нему датчики и моторы, написать программу и превратить любую конструкцию в самостоятельного робота, умеющего видеть, чувствовать, принимать решения и выполнять определённые действия. Название «*Mindstorms*», которое стало именем всего семейства конструкторов на базе программируемых контроллеров *LEGO*, было взято из названия книги «*Mindstorms: Children, Computers and Powerful Ideas*», написанной профессором математики и компьютерных технологий Сеймуром Пейпертом, где он предложил уникальную концепцию объединения компьютерных технологий и образовательного процесса[23; 50].

Микрокомпьютер *RCX* представляет собой программируемый микропроцессорный блок с памятью, жидкокристаллическим дисплеем и инфракрасным интерфейсом, предназначенным для связи с компьютером *PC* или *Mac*, а также с другими микрокомпьютерами.



Рис. 5. Микрокомпьютер *RCX*

RCX функционирует как автономный компьютер и является «мозгом» всех моделей роботов серии «ПервоРобот». К нему подключаются датчики (до трех одновременно), получающие информацию об окружающей среде. Действия роботов определяются программами, которые разрабатываются на настольном компьютере с помощью программного обеспечения *RoboLab* и загружаются в *RCX* посредством инфракрасного передатчика.



Рис. 6. Модели учебных роботов на базе конструктора «Индустрия развлечений. ПервоРобот»

2.2.3. LEGOMindstormsNXT

Новая серия робототехнического конструктора – *MindstormsNXT*, выпущенная компанией «LEGO» в 2006 г., является, по сути, полностью переработанной и улучшенной преемницей *RCX*, были разработаны новая среда и язык программирования *NXT-G*. Новый блок *NXT* оказался совместимым со старыми датчиками *RCX* (через переходник).

В 2007 г. была выпущена образовательная версия *NXT* – «*LEGOMindstormsEducationNXTBaseSet*» в пластиковой коробке с лотками для деталей. Позднее в 2008 г. производители предложили ресурсные наборы «*LEGOMindstormsEducationResourceSet*», которые содержат только детали *LEGO*, в них нет электронных компонентов.



Рис. 7. Базовый (v.9797) и ресурсный наборы *LEGO Mindstorms NXT*

В 2009 г. появилась ещё одна версия *LEGOMindstormsNXT* 2.0 (см. рис. 7). Кроме незначительных изменений в составе мелких деталей, набор *NXT* 2.0 получил новый датчик цвета взамен монохромного датчика освещённости, но из набора был исключён датчик звука [25; 50].

Основным компонентом конструкторов *LEGO-MindstormsNXT* является микрокомпьютер *NXT*, снабженный

входными портами для датчиков и выходными портами для исполнительных устройств, делающий робота программируемым, интеллектуальным, способным принимать решения. В микрокомпьютер *NXT* можно загружать программу, созданную с помощью программного обеспечения для настольного компьютера, а можно обойтись и без помощи компьютера – используя меню *NXTProgram*, например, запрограммировать робота таким образом, чтобы он двигался вперёд и назад при нажатии кнопки датчика касания. Для обмена данными между персональным компьютером и микрокомпьютером *NXT* можно воспользоваться *USB*-портом, а можно установить беспроводное соединение между *NXT* и другими устройствами, поддерживающими *Bluetooth*-связь, например, с другими *NXT*, с мобильными телефонами или с компьютерами.

NXT снабжен тремя разъёмами для подключения моторов и ламп, четырьмя разъёмами для датчиков (сенсоров), встроенным динамиком для воспроизведения звука.

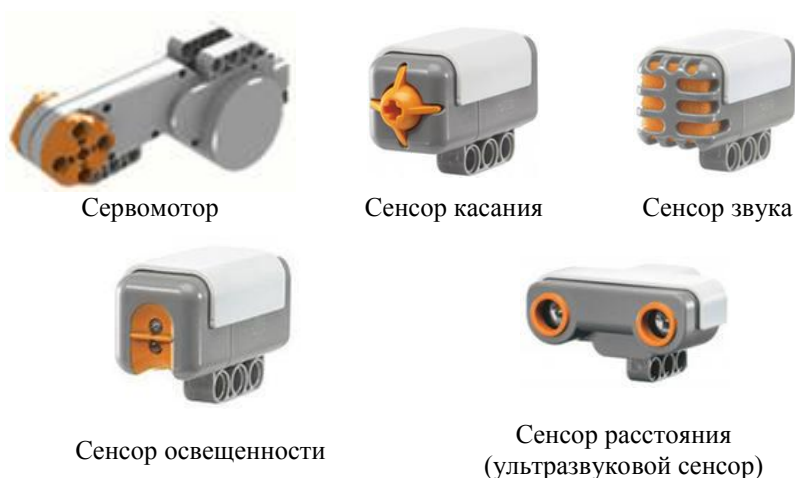


Рис. 8. Стандартные сенсоры для *LEGO Mindstorms NXT*(v.9797)



Сервомотор



Сенсор касания



Сенсор цвета











Сенсор расстояния

Рис. 9. Стандартные сенсоры для *LEGO Mindstorms NXT 2.0*

Таблица 1

Конструкционные элементы *LEGO Mindstorms NXT*

Шестеренки					
					
8-зубое зубчатое колесо	16-зубое зубчатое колесо	24-зубое зубчатое колесо	Червячное зубчатое колесо	4-зубое зубчатое колесо	Зубчатое колесо 24 зуба внутри, 56 зубьев снаружи
Ремни, оси, втулки для осей					
					
Ремень, 33 мм	Втулка, удлинитель оси	Втулка	Ось, 5-ти модульная	Ось, 6-ти модульная	Ось, 12-ти модульная

Колеса с шинами и шкивы различного диаметра					
					
Ступица	Колесо	Ступица	Колесо	Ступица	Колесо
Ориентирующие и комбинированные соединители					
					
Соединительный штифт 2-модульный двойной	Соединительный штифт 3-модульный двойной	Фиксатор 2-модульный	Фиксатор 3-модульный	Угловой фиксатор	
Цилиндрические фиксаторы					
					
Соединительный штифт полуось	Соединительный штифт с выступами / полуось	Соединительный штифт со втулкой	Соединительный штифт с выступами	Соединительный штифт с выступами	
Балки					
					
Балка 3-модульная	Балка 5-модульная	Изогнутая балка 3*5-модульная	Балка с шипами 1*2	Балка с шипами 1*6	Пластина с отверстиями 2*4

2.2.4. LEGO Mindstorms EV3

Третья версия *LEGO Mindstorms* была представлена в 2013 г. Обновился состав электронных компонентов конструктора, но основные изменения коснулись программного блока (рис. 10).



Рис. 10. Программируемый блок EV3

Как и предыдущая модель конструктора *NXT*, новый *EV3* представлен в двух комплектациях: «потребительская» (*retail version*) и образовательная (*educational version*) [26]. Программируемый блок и три мотора одинаковы в обоих наборах, но состав датчиков различается [25; 33; 49]:

Большой сервомотор



Средний сервомотор



Сенсор касания



Сенсор цвета



Гироскоп



Датчик расстояния
(ультразвуковой)



Рис. 11. Стандартные сенсоры для *LEGO Mindstorms EV3*
(*educational version*)

Большой сервомотор



Средний сервомотор



Сенсор касания



Сенсор цвета



Датчик расстояния
(инфракрасный)



1 инфракрасный маячок / блок управления



Рис. 12. Стандартные сенсоры для *LEGO Mindstorms EV3* (retail version)

В комплекте с *EV3* поставляется новая графическая среда разработки на базе *LabView*, похожая на *NXT-G* (рис. 13).



Рис. 13. Графическая среда для разработки робота *LEGO Mindstorms EV3*

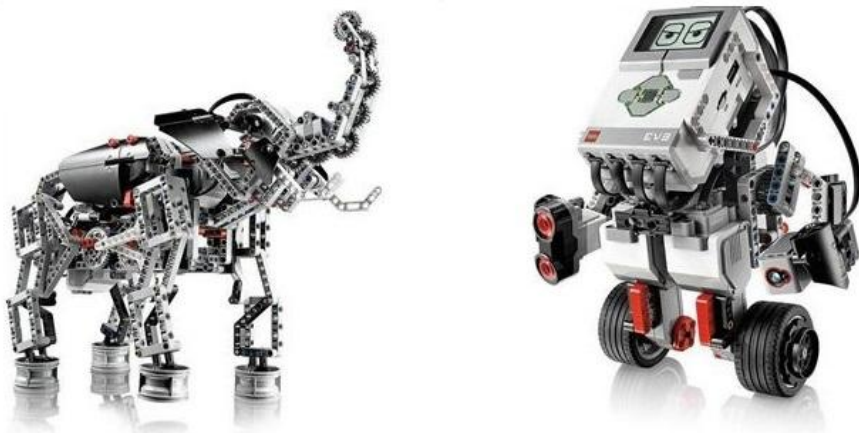


Рис. 14. Модели учебных роботов *LEGO Mindstorms EV3*
«Слон», «Сегвей»

2.2.5. Tetrrix

Tetrrix – основной конструктор международных соревнований *FIRST Tech Challenge* (см. § 14). *Tetrrix* включает в себя все необходимое для создания металлических роботов, которые могут управляться микрокомпьютером *NXT*.

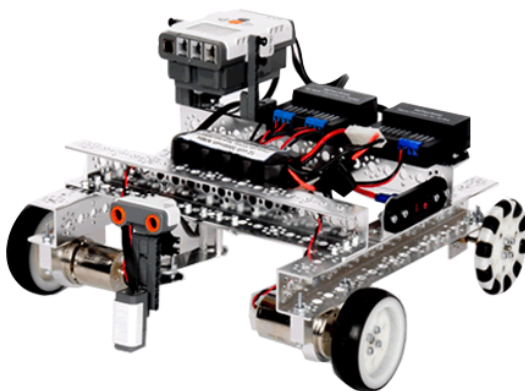


Рис. 15. Модель учебного робота на основе конструктора *Tetrrix*
с микрокомпьютером *NXT*

2.2.6. Roborobo

Серия конструкторов *Roborobo* представлена пятью комплектами, отличающимися по уровню сложности. В каждый из них входят: текстолитовые платы, металлические соединители и уголки, моторы и сервоприводы, колеса и гусеницы, контроллер для моторов и сервоприводов, пульт дистанционного управления, программное обеспечение, кабели для подключения к компьютеру. Для программирования *Roborobo* используется программное обеспечение *ROGIC*, оно позволяет начинающим легко освоить азы программирования.



Рис. 16. Модели учебных роботов Roborobo

2.2.7. Fischertechnik

Робототехнические конструкторы *Fischertechnik* созданы немецким ученым – профессором Артуром Фишером. Эти конструкторы во многом не уступают *LEGO Mindstorms*, они имеют аналогичные составляющие элементы. Робототехнические конструкторы *Fischertechnik* рассчитаны на учащихся старше 10 лет. В комплекты конструкторов входят программируемые контроллеры (*Robo TX*, *Robotics TXT*), двигатели, различные датчики и блоки питания, что позволяет приводить механические конструкции в движение, создавать роботов и программировать их с помощью компьютера. Для разработки управляющих про-

грамм для контроллера *Robo TX* используется среда программирования *Robo Pro*. Программы составляются на графическом языке в виде блок-схем. Готовые программы загружаются в контроллер через интерфейсы *USB* или *Bluetooth*.



Рис. 17. Учебная модель «Автоматический высотный стеллажный склад из серии» «*RoboTX*. Автоматические роботы»



Рис. 18. Модель учебного мобильного робота из серии «*Robo TX*. Учебная лаборатория»

2.2.7. Arduino

Arduino – удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Представляет собой небольшую плату с собственным процессором, памятью. На плате имеется достаточно большое количество контактов, к которым можно подключать всевозможные компоненты: лампочки, датчики, моторы другие электронные компоненты.

Программы для *Arduino* пишутся на языке C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Для удобства работы с *Arduino* существует бесплатная официальная среда программирования «*Arduino IDE*», работающая под *Windows*, *Mac OS* и *Linux*. Для

изучения образовательной робототехники можно использовать образовательный набор «Амперка», в комплект которого входит в том числе и микроконтроллер *Arduino* [2; 3].



Рис. 19. Платформа *Arduino*



Рис. 20. Образовательный набор «Амперка»

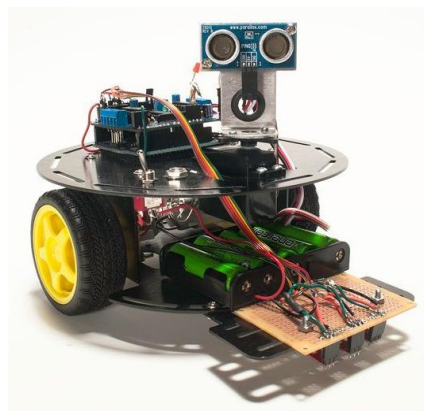


Рис. 21. Модели учебных мобильных роботов на базе *Arduino*

2.2.8. Bioloid

Это семейство образовательных робототехнических конструкторов, предназначенных для обучения в области разработки и отладки сложных робототехнических систем. Название *Bioloid* получено путем сложения слов «bio» + «all» + «oid», что отражает концепцию данного конструктора: любое живое суще-

ство может быть представлено в виде робота. Для программирования микроконтроллера CM-5 служит среда *RoboPlus* (см.рис. 22). Движение созданных роботов обеспечивается сервоприводами *Dynamixel AX-12A* или *AX-18A* [32].

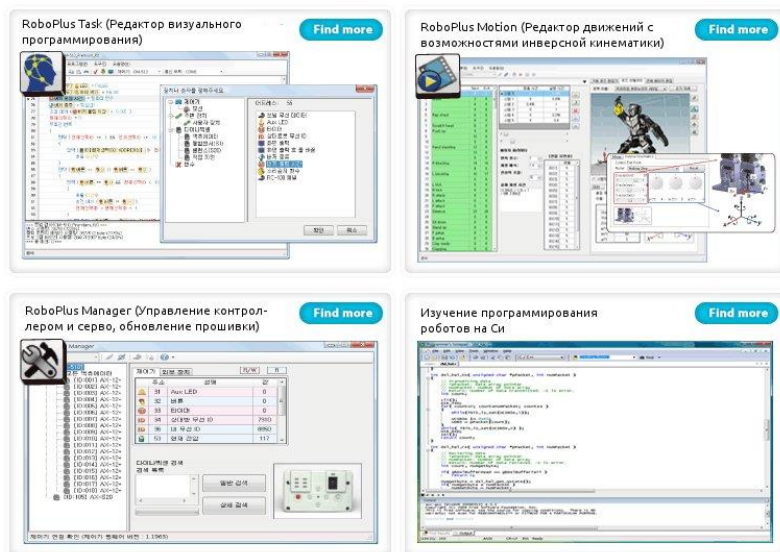


Рис. 22. RoboPlus (среда разработки роботов Bioloid)

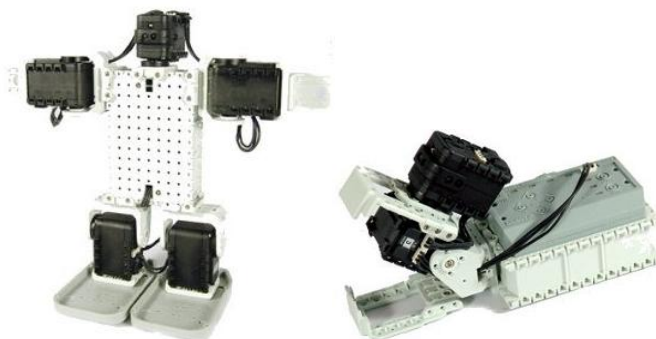


Рис. 23. Модели учебных роботов «Пингвин», «Крокодил» серия «*RoboticsBioloid* новичок (*BeginnerKit*)»



Рис. 24. Модели учебных роботов «Королевский паук», «Гуманоид» серия «*Robotis Bioloid* комплексный (*Comprehensive Kit*)»

Задания для самостоятельной внеаудиторной работы

1. Подготовьте реферат по одной из тем:
 - 1.1. Использование элементов робототехники на уроках физики;
 - 1.2. Использование элементов робототехники на уроках информатики;
 - 1.3. Использование элементов робототехники на уроках математики;
 - 1.4. Использование элементов робототехники на уроках технологии;
 - 1.5. Использование элементов робототехники на уроках биологии;
 - 1.6. Образовательная робототехника в начальной школе.
2. Используя рекомендованные печатные издания и интернет-источники составьте сводную таблицу по оборудованию для изучения образовательной робототехники (название конструктора, образовательные задачи, решаемые при использовании данного конструктора, предполагаемый возраст учащихся, возможности использования в учебном процессе, преимущества, недостатки).

3. Используя интернет-источники, самостоятельно соберите информацию о конструкторах (отечественных разработках), которые можно было бы использовать на занятиях по образовательной робототехнике в школе.

§3. МЕТОДИКА ОБУЧЕНИЯ ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ

В основе всех продуктов *LEGO* лежит принцип «*Learning by making*» («обучение через действие»), предполагающий реализацию циклической модели обучения, основанной на четырех образовательных составляющих: взаимосвязь, конструирование, рефлексия, развитие (рис. 25).

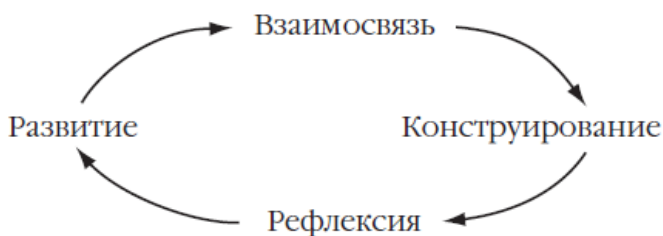


Рис. 25. Циклическая модель обучения на основе образовательного оборудования Lego [28; 29]

«Взаимосвязь» предполагает, что пополнение багажа знаний происходит, когда вновь приобретенные опыт и знания удастся соединить с уже имеющимися или сделать их стимулом, отправной точкой для нового этапа обучения.

«Конструирование» подразумевает создание моделей и генерацию идей. Предлагаются три вида «Конструирования»:

1) свободное «зондирование» проблемы – учащиеся знакомятся с новым понятием, самостоятельно модифицируя простые модели и управляя ими;

2) исследование по инструкции – учащиеся, следуя подробным инструкциям, создают модели, которые служат для получения количественных результатов, пригодных для математической обработки;

3) свободное решение проблемы – учащиеся создают модель собственной конструкции, способную выполнить поставленную задачу.

«Рефлексия» предполагает осмысление того, что сделано, создано, модифицировано, поиск словесной формулировки полученного знания, способов представления результатов опыта, путей его применения в комплексе с другими идеями и решениями.

Поддержка творческой атмосферы, эмоциональной и физической радости от успешно выполненной работы реализуется на этапе «Развитие» при выполнении более сложных заданий, способствующих углублению полученного опыта, развитию креативных и исследовательских навыков.

По содержанию в начальном курсе изучения образовательной робототехники можно выделить два раздела: «Основы конструирования и механики» и «Программирование робототехнических устройств» [46]. Раздел «Основы конструирования и механики» направлен на изучение базовых механических принципов и элементарных технических решений, лежащих в основе всех современных конструкций и устройств (устойчивость и прочность конструкции, зубчатые и ременные передачи, рычаги, колеса, оси, блоки, преобразование движения). Освоение этих знаний дает учащимся возможность разработки и сборки модели для реализации поставленной задачи.

Раздел подготовки «Программирование робототехнических устройств» должен основываться на требованиях к результатам изучения предметной области «Математика и информатика», включающих в себя «развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать

алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической» [41]. Любая робототехническая конструкция выступает в роли исполнителя программы (алгоритма).

В процессе обучения образовательной робототехнике педагог сталкивается с необходимостью:

- организации проектной деятельности;
- использования метода научного познания в ходе выполнения проекта;
- подготовки учащихся к соревнованиям роботов.

Метод проектов является ведущим методом при обучении образовательной робототехнике. Этот метод предполагает использование широкого спектра проблемных, исследовательских, поисковых методов, ориентированных на реальный практический результат. Использование метода проектов в обучении предполагает наличие значимой в исследовательском, творческом плане проблемы (задачи, требующей интегрированного знания, исследовательского поиска для её решения), а также организацию самостоятельной (индивидуальной, парной, группой) деятельности учащихся на уроке или во внеурочное время.

Создание действующей модели робота представляется в виде проекта[28]. При этом созданная модель робота одновременно сочетает в себе функции как практико-ориентированного, так и исследовательского проекта.

Практико-ориентированный проект отличает обозначенный с самого начала результат деятельности участников проекта – робот, выполняющий определенные функции (например, проект «Парковка», проект «Измерение уровня шума» и др.). Исследовательский характер такой проект приобретает при отладке работы робота. Суть исследовательского проекта в том, что он подчинен логике пусть небольшого, но исследования и

имеет структуру, приближенную к подлинным научным исследованиям. Испытание и отладка работы робота также предполагает уяснение цели и обоснование гипотезы, проектирование серии опытов для испытания модели, подготовка условий для испытаний, осуществление испытаний, внесение исправлений в конструкцию и программу робота на основе сделанных выводов.

Подготовка к соревнованиям роботов ставит необходимость перед учителем работать в качестве тренера команды (§ 8). Соревнования как метод организации учебно-познавательной деятельности школьников по робототехнике способствует формированию качеств конкурентоспособной личности. Этот метод опирается на естественные склонности учащегося к лидерству, к соперничеству. В процессе соревнования школьник достигает определенного успеха в отношениях с товарищами, приобретает новый социальный статус. Соревнование вызывает не только активность, но и формирует у него способность к самоактуализации, которую можно рассматривать как метод самовоспитания.

При подготовке команд к соревнованиям широко используются методы сотрудничества, которые основаны на совместной деятельности школьников и тренера команды «на равных». К ним относятся: совместные обсуждения, дискуссии, активизирующее общение в парах «учитель-ученик», «ученик-ученик», в коллективе «тренер-операторы» [34].

Соревнования роботов являются своеобразным смотрам достижений для учащихся участие в соревнованиях является своеобразным результатом их деятельности по изучению робототехники. Соревнования направлены на повышение уровня мотивации у учащихся к занятиям робототехникой, на популяризацию этого вида технического творчества.

Соревнования затрагивают не только аспекты технической подготовки команд, но и коллективной деятельности уча-

щихся, эмоционального воздействия. Работая над проектом, члены команды учатся распределять ответственность, взаимодействовать, определять приоритеты и прогнозировать возможные затруднения.

Тренерская работа при подготовке к соревнованиям включает следующие виды деятельности [38]:

- формирование состава команды;
- специальная подготовка к конкретным соревнованиям (см. глава IV);
- морально-психологический настрой команды на предстоящие соревнования.

Этапы подготовки модели робота к соревнованиям [38]:

- 1) изучение регламента (требования к конструкции и средствам программирования, принцип судейства);
- 2) составление плана подготовки;
- 3) поиск идеи решения (использование своего и чужого опыта, ресурсов сети интернет);
- 4) анализ идеи решения;
- 5) создание эскиза модели;
- 6) сборка конструкции робота согласно выбранной идее;
- 7) составление программы для робота согласно выбранной идее;
- 8) тестирование модели робота и выявление слабых мест;
- 9) доработка и совершенствование модели.

Задания для самостоятельной внеаудиторной работы

Подготовьте рефераты по темам:

1. Проектная деятельность школьников на занятиях по образовательной робототехнике;
2. Исследовательская деятельность школьников на занятиях по образовательной робототехнике;
3. Игровые методы обучения образовательной робототехнике.

ГЛАВА II УСТРОЙСТВО УЧЕБНОГО РОБОТА

У робота можно выделить три основные части. Одна часть объединяет в себе исполнительные органы, базой которых служат механические элементы. С помощью исполнительных органов робот активно воздействует на внешнюю среду. Другая важная часть объединяет органы чувств (чувствительные элементы), с помощью которых выполняется сбор необходимой информации о внешней среде. Чувствительные элементы в своей работе используют технику измерений. Третья часть объединяет элементы обработки информации, она базируется на вычислительной технике [47].

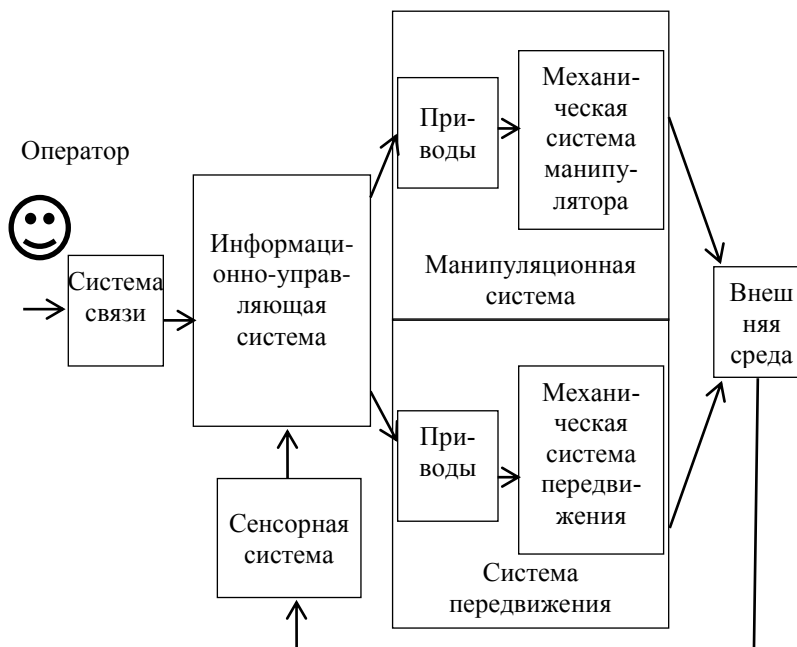


Рис. 26. Функциональная схема робота [47]

§ 4. Устройство управления учебного робота

Информационно-управляющая система – это «мозг» робота, который служит для выработки законов управления механизмами исполнительской системы на основе заложенной программы с учетом сигналов обратной связи от сенсорной системы. Информационно-управляющая система Lego-робота представляет собой блок с микрокомпьютером – *RCX*, *NXT* или *EV3* – в зависимости от поколения конструктора. К данному блоку в определенном порядке подключаются моторы, отвечающие за исполнительную систему, и датчики, составляющие сенсорную систему робота.



Рис. 27. Информационно-управляющая система LegoMindstormsNXT

Таблица 2

Порядок подключения стандартных датчиков и моторов к блоку *NXT*

Порт	Устройство
A, B, C	Моторы
1	Датчик касания (<i>Touch Sensor</i>)
2	Датчик звука (<i>Sound Sensor</i>)
3	Датчик освещенности (<i>Light Sensor</i>)
4	Ультразвуковой датчик расстояния (<i>UltrasonicSensor</i>)

Задания для самостоятельной внеаудиторной работы

Используя список рекомендованной литературы и интернет-источники, составьте сравнительную таблицу функциональных характеристик блоков *RCX*, *NXT* или *EV3*.

§ 5. Сенсорная система учебного робота

5.1. Датчик как чувствительный элемент в системе робота

В датчике как устройстве для измерения физических величин происходят три процесса: восприятие входной физической величины, её преобразование в промежуточную (или сразу же выходную) величину той же или иной физической природы, формирование измерительного сигнала, передаваемого вдоль измерительной цепи, сопрягаемой с датчиком [17].

Поскольку любой датчик является средством измерения, то важно понимать смысл терминов «точность» и «погрешность» измерений. Точность – это характеристика качества средства измерений. Мы можем высказывать пожелания об измерении физической величины с большей или меньшей точностью. Погрешность средства измерения – это количественная оценка получаемого результата измерений, выражаемая в разнице между измеренным и истинным значением физической величины [17].

Датчик как средство измерения характеризуется диапазоном измерений, т.е. диапазоном значений измеряемой физической величины, в пределах которого датчик способен преобразовать её в измерительный сигнал при соблюдении функциональной зависимости между входной и выходной величинами. Функциональная зависимость между входной и выходной величинами называется функцией преобразования датчика, она может выражаться в виде формулы, графика или таблицы. Чаще всего стремятся иметь линейную функцию преобразования, т.е.

прямую пропорциональность между изменением входной величины и соответствующим приращением выходной величины датчика (рис. 28).

Для описания линейной функции преобразования $y = f(x)$, где x – входная, а y – выходная величина, достаточно начального значения y_0 , именуемой нулевым уровнем, соответствующего нулевому (или какому-либо другому характерному) значению входной величины x , и показателя относительного наклона прямой $s = \Delta y / \Delta x$, называемого чувствительностью датчика [17].

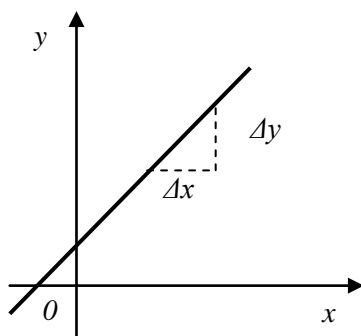


Рис. 28. Линейна функция преобразования

Система чувствительных датчиков наделяет робота способностью воспринимать информацию окружающей среды, зачастую датчики являются прототипами сенсорной системы человека.

Восприятие внешнего мира человеком осуществляется через шесть основных каналов сенсорной рецепции, образующих распределенную информационную сеть. Это каналы органов чувств – слух, зрение, осязание, терморепция, обоняние и вкус. В каждом из них возбуждение регистрируется системой рецепторов, специфических для разных сенсорных модальностей, и передается по каналу связи (нервному волокну) в виде потенциалов действий. Система рецепторов каждой модально-

сти связана с определенными отделами центральной нервной системы.

Распознавание сенсорного образа у человека является результатом совместной работы информационной системы и мозга. Значительная часть информации обрабатывается уже на уровне рецепторов. При этом возможность рецепторов по передаче информации существенно превышает возможности информационной системы по ее переработке, а, следовательно, пропускная способность рецепторных нейронов определяет максимальный поток информации. Заметим, что благодаря высокой пропускной способности зрение играет основную роль в адаптации человека к окружающей среде. Можно предположить такую же роль искусственного зрения при адаптации робота в условиях неопределенности. В то же время слух и особенно осязание обладают значительно более высокой чувствительностью и используются для тонкой коррекции движений при взаимодействии с внешней средой.

Биологические механизмы сенсорных функций являются прототипом сенсорной системы робота. Заметим, что во многих случаях одна и та же поведенческая задача может быть решена путем объединения нескольких сенсорных функций. Хорошим примером такого взаимодействия является движение. При ходьбе человек использует зрение, тактильную, кинестетическую, а также слуховую сенсорные функции. Однако, как правило, предоставляемая этими системами информация оказывается избыточной. Например, чтобы пройти по улице, достаточно использовать всего три информационных канала. Замещение одной сенсорной функции другой получило название сенсорной компенсации. В частности, тактильная рецепция слепого частично замещает зрение. Принцип замещения широко используется в робототехнике [8].

Сенсорная система роботов (система чувствительных датчиков) обычно копирует функции органов чувств человека

(зрение, слух, обоняние, осязание и вкус). Чувство равновесия и положения тела в пространстве, как функция внутреннего уха, иногда считаются шестым чувством. Сенсорные датчики могут обнаружить какие-то внешние сигналы и определить их величину, что выражается в появлении на выходе пропорционального электрического сигнала. Информация, содержащаяся в сигнале, должна быть считана и обработана «интеллектом» робота (например, микроконтроллером) или нейронной сетью. Как правило, классы сенсорных устройств выделяются по типу воздействия, на которое данный сенсор реагирует: свет, звук, тепло и др. Типы сенсоров, встроенных в робота, определяются целями и местом их применения.

Таблица 3

Назначение датчиков для *Lego MindstormsNXT* и *EV3*

№	Датчик	Назначение
1	Датчик вращения мотора	Движение, подсчет числа оборотов мотора
2	Датчик касания	Осязание (прикосновение)
3	Датчик звука	Слух (уровень громкости звука)
4	Датчик цвета / света	Зрение (определение освещенности помещения, яркости объектов, различение цветов)
5	Датчик расстояния	Зрение (отслеживание препятствий, определение расстояния до препятствий)
6	Инфракрасный датчик	Зрение (отслеживание препятствий, определение расстояния до препятствий)
7	Датчик-гироскоп	Определение вращательного движения
8	Датчик ускорения	Определение изменения скорости
9	Датчик-компас	Определение положение тела в пространстве (ориентация по сторонам света)

5.2. Сервомотор

Сервомотор NXT – это сочетание электродвигателя, шестерёнчатого редуктора и датчика вращения, объединённых в одном корпусе своеобразной формы.



Рис. 29. Сервомотор *NXT*

Малогабаритные двигатели постоянного тока обычно вращаются слишком быстро и имеют не очень большую мощность на валу, поэтому нет особого смысла подключать их напрямую к колёсам или другим механизмам. Чтобы понизить скорость вращения и заодно увеличить вращающий момент, обычно используют какой-либо редуктор. Редуктор сервомотора содержит восемь шестерёнок, имеющих суммарное передаточное отношение 1:48 [20; 50].

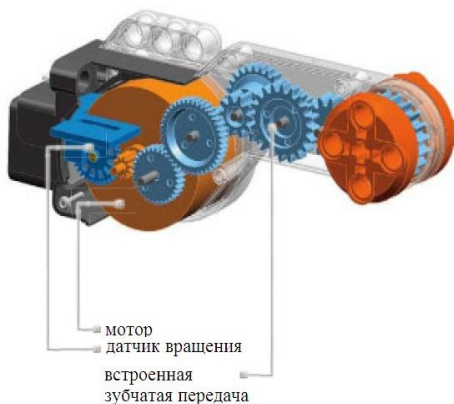


Рис. 30. Внутреннее устройство сервомотора *NXT*

В каждый мотор встроен датчик вращения. Он позволяет точнее вести управление моторами. Датчик вращения измеряет вращение мотора в градусах с точностью $\pm 1^\circ$. Полный оборот мотора составляет 360° . Встроенный в каждый мотор датчик позволяет также выставлять разные скорости для моторов (разные параметры мощности в программном обеспечении).

Одной из важнейших характеристик мотора является крутящий момент (или вращательный момент, момент силы). Момент силы – векторная физическая величина, равная произведению радиус-вектора, проведенного от оси вращения к точке приложения силы, на вектор этой силы (рис. 31). Момент силы характеризует вращательное действие силы на твёрдое тело. В системе СИ единицами измерения для момента силы является Ньютон-метр.

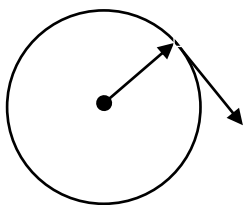


Рис. 31. Момент силы

5.3. Датчик касания

Осязание и чувство давления робота может быть реализовано с помощью датчика касания (*Touch Sensor*). Он представляет собой кнопку, у которой возможно два состояния – «Нажат» (*pressed*) и «Отпущен» (*released*). Программно датчик распознает еще одно состояние «Щелчок» (*Bumped*).

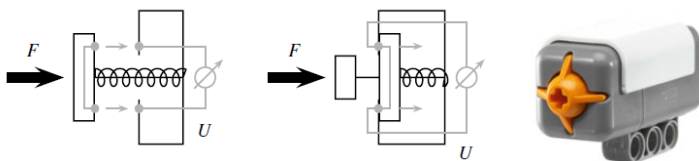


Рис. 32. Схема [48] и внешний вид датчика касания NXT

Датчики касания обычно используются в роботах для обнаружения препятствий на пути следования, что позволяет роботу избегать столкновений. Более совершенные датчики прикосновения и давления используются в конструкциях рук и кистей. Такие датчики позволяют «руке» робота захватывать и удерживать предметы с достаточным усилием без риска их повреждения.

5.4. Датчик расстояния

Ультразвуковой датчик (*Ultrasonic Sensor*) наделяет робота способностью видеть и распознавать объекты, избегать препятствия, измерять расстояния и обнаруживать движение. Расстояние датчиком измеряется в сантиметрах и дюймах. Диапазон измерений составляет от 0 до 2,5 м с точностью в ± 3 см.

По своему физическому устройству датчик расстояния можно представить как передатчик и приемник: передатчик испускает волну в ультразвуковом диапазоне, волна, отраженная от удаленной поверхности, улавливается приемником (рис. 33).

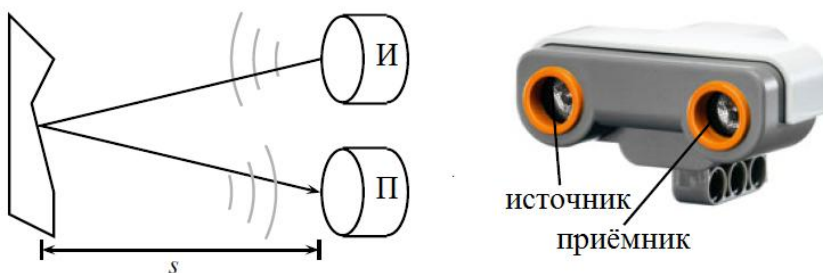


Рис. 33. Функциональная схема датчика расстояния.
Внешний вид датчика расстояния *NXT*

Поскольку путь до препятствия и обратно у звуковой волны займет какое-то время Δt , программа получит данные с сенсора не сразу, а как минимум через это самое время. Рассчитаем

теоретический предел задержки [33]. Стандартный ультразвуковой сенсор, поставляемый с набором *Mindstorms*, позволяет определять расстояние S до 2,5 метров. Т.е., зная скорость звука $v_{звукa}$ (пусть будет $v_{звукa}=330$ м/с), можно рассчитать, что максимальное время ожидания вернувшейся волны Δt :

$$\Delta t = \frac{v_{звукa}}{2 \cdot s} = \frac{330 \text{ м/с}}{2 \cdot 2,5 \text{ м}} = \frac{1}{66} \text{ с} \approx 0,15 \text{ с}$$

Большие объекты с твердой, хорошо отражающей звук поверхностью, дают самые надежные показания. Хуже всего регистрируется сигнал от маленьких либо тонких объектов с искривленной поверхностью (например, шарик). Мягкие объекты, такие как ткань, могут поглощать ультразвуковые волны и не обнаруживаться этим датчиком. Два и более ультразвуковых датчика, работающих в одном помещении могут интерферировать и снижать точность результатов измерений.

5.5. Датчик освещенности

Освещенность – физическая величина, численно равная световому потоку, падающему на единицу освещенной поверхности; в СИ за единицу освещенности принят люкс (лк) (от лат. *lux* – свет). Освещенность прямо пропорциональна силе света источника, она зависит от расстояния от источника света до освещаемого предмета, а также от угла, под которым свет падает на поверхность. Если пучок света падает перпендикулярно к поверхности, то световой поток распределяется на минимальной площади, освещенность при этом максимальна.

Освещенность уменьшается, если в воздухе есть частички пыли, тумана, дыма, так как они отражают и рассеивают определенную часть световой энергии.

Существует большое количество различных типов датчиков освещенности фоторезисторы, фотоэлектрические устройства, фотодиоды и фототранзисторы. Простейшие из них фоторе-

зисторы и фотоэлементы – относятся к классу полупроводниковых приборов, в который входят также транзисторы и интегральные микросхемы. Полупроводниковые материалы по своей электропроводности занимают промежуточное положение между электроизоляционными и проводящим материалами. Носителями тока в полупроводниках являются электроны, но характер их движения в некоторых полупроводниках таков, что для удобства описания природы их электропроводности за ток принимают движение положительных зарядов, равных по величине заряду электрона. Такие проводники обладают электропроводностью р-типа (от лат. *positivus* – положительный), а те, в которых ток образован движением электронов, обладают электропроводностью n-типа (от лат. *negative* – отрицательный).

Основными полупроводниковыми материалами, получившими широкое практическое применение, являются германий и кремний: они имеют кристаллическую структуру. Проводимость чистых полупроводников очень мала, однако при внесении в кристаллическую решетку полупроводников атомов других элементов, например мышьяка, она резко повышается. Такие добавочные элементы называются примесями. Меняя их концентрацию и используя разные элементы, можно в широких пределах менять их электропроводность и ее тип (р или n), т. е. придавать полупроводнику нужные свойства.

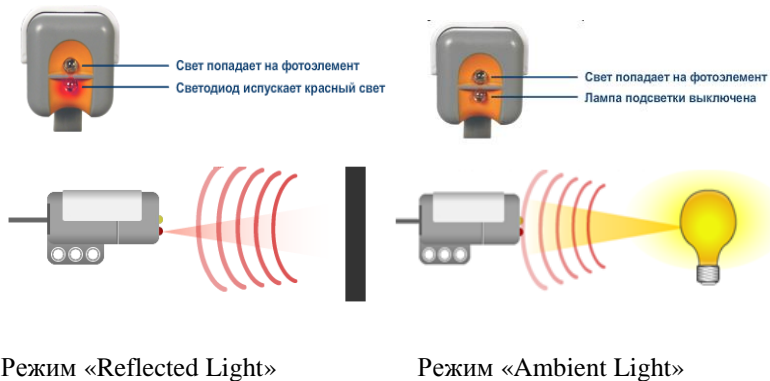
Датчик освещённости (*LightSensor*) содержит в себе красный светодиод, который можно включить и выключить программно, а также фотоэлемент, который измеряет яркость падающего на него света (рис. 34).

В режиме измерения окружающей освещенности (*Ambient Light*), количество света, попавшее на светочувствительный элемент, преобразуется в цифровое значение, которое уже используется в программе. Например, с датчиком, работающем в этом режиме, можно собрать робота, который ищет самое освещенное место в комнате.

В режиме измерения отраженного цвета (*Reflected Light*), помимо светочувствительного элемента, активируется светоиспускающий элемент (светодиод). Свет, выпущенный этим элементом, отражается от какой-нибудь поверхности и попадает обратно в светочувствительный элемент [25; 33; 49].



Рис. 34. Функциональная схема фотодиода [48].
Внешний вид датчика освещенности



Режим «Reflected Light»

Режим «Ambient Light»

Рис. 35. Режимы измерения освещенности датчиком света

Чем светлее отражающая поверхность, тем больше света приходит в светочувствительный элемент. И наоборот, чем темнее поверхность, тем меньше света приходит на светочувствительный элемент. Данное количество света преобразуется в цифровое значение и передается в программу. Если датчик на-

ходится над темной поверхностью – в программу приходят маленькие значения; чем светлее поверхность, тем больше света приходит – программа оперирует с большими значениями [33]. Для корректного уровня освещенности необходимо держать сенсор под прямым углом на расстоянии 1 см над поверхностью.

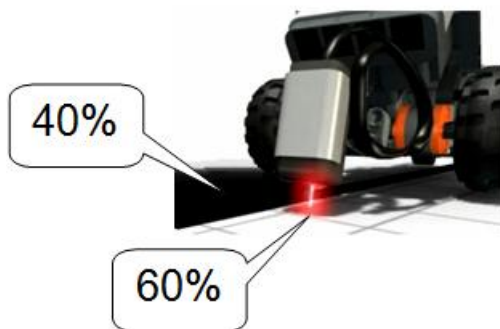


Рис. 36. Показания датчика света (в режиме отраженного света) на светлой и на темной поверхности

Принимая во внимание то, что физическая природа цвета объясняется отражательной способностью какой-либо поверхности, можно понять, каким образом робот с помощью датчика освещенности будет различать не только темную и светлую поверхности, но и цвета (желтый, зеленый, синий и др.). При этом следует учитывать, что фотозлемент в датчике освещенности более чувствителен к инфракрасному излучению, чем к обычному видимому спектру света. График на рис. 37 показывает различия в спектральном диапазоне датчика освещенности и человеческого глаза.

На рис. 38 показаны значения освещенности, выдаваемые датчиком в зависимости от яркости освещения, выраженной в люксах (лк). Из графика заметно, что характеристика датчика далека от линейной, но это не мешает использовать датчик по назначению.

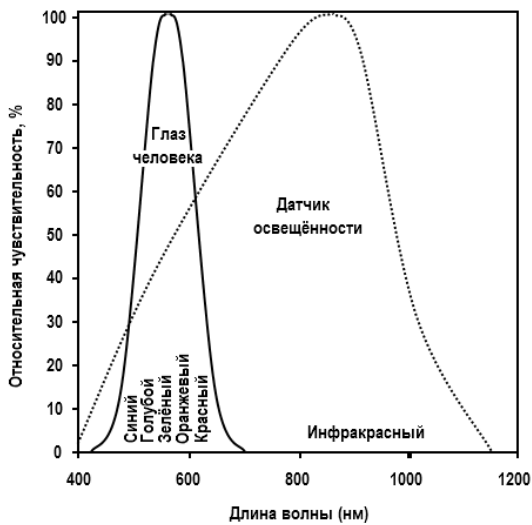


Рис. 37. Сравнение спектральных диапазонов датчика освещённости и глаза человека [25; 50]

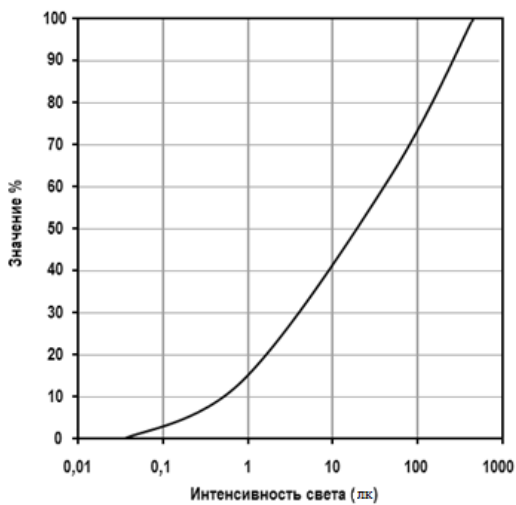


Рис. 38. Функция преобразования датчика света Lego [25; 50]

Одним из способов управления движением роботов является движение по разметке, для этого используется датчик освещенности. Он устанавливается вертикально вниз. Для создания алгоритма движения по разметке необходимо знать пороговое значение срабатывания датчика.

Пороговое значение – это число, разделяющее два диапазона значений, например, на «светлые» и «тёмные». Используя пороговое значение, робот определяет, какие значения показаний датчика света относятся к «светлым», а какие – к «тёмным». Это позволяет разделить все возможные показания датчика (от 0 до 100) на два диапазона. В зависимости от того, к какому из них относится очередной сигнал, поступивший от датчика, робот поворачивает налево или направо, чтобы не «потерять» линию. Пороговое значение для различных роботов будет различаться, оно может получиться либо слишком высоким, либо слишком низким – в зависимости от уровня освещённости в помещении, а также от отражающей способности поверхности рабочего поля и черной линии (разметки). Пороговое значение определяется как среднее арифметическое между показаниями датчика над темной и светлой областями [27].

5.6. Датчик цвета

Датчик цвета (*Color Sensor*) от *Lego* представляет собой усовершенствованный датчик освещенности и может различать 6 различных цветов. Помимо своей основной задачи – различать цвета, этот датчик полностью дублирует функции датчика освещённости. Датчик определяет цвета, последовательно измеряя отражённый свет, излучаемый красным, зелёным и синим светодиодами. Результаты этих измерений обрабатываются особым образом, и по результатам обработки датчик пытается определить ближайшее значение цвета и выдать его код [25]. Для корректного определения цвета необходимо держать сенсор под прямым углом на расстоянии 1 см над поверхностью.



Рис. 39. Датчик цвета NXT 2.0

5.7. Датчик звука

Датчик звука (*Sound Sensor*) представляет собой микрофон, он измеряет уровень громкости звукового сигнала в дБа (звуки диапазона 20–20000 Гц, воспринимаемые ухом человека) и дБ [весь диапазон, включая инфразвук (ниже 20 Гц) и ультразвук (свыше 20 000 Гц)]. Максимальное звуковое давление, которое может измерить датчик – около 90 дБ, что соответствует громкости газонокосилки. Показания датчика звука выражаются в процентах [%] от максимальной громкости, которую он способен зафиксировать.

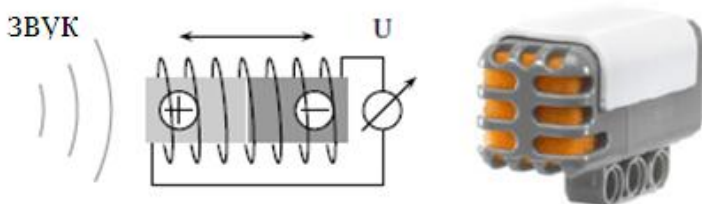


Рис. 40. Функциональная схема датчика звука [48].
Внешний вид датчика звука

Громкость звука, или уровень звукового давления измеряется в единицах, называемых децибелами (дБ). Децибелы показывают, насколько громкость звука больше или меньше относительно другого звука, в данном случае 0 дБ – это самый тихий звук на пределе слышимости среднестатистического человека.

Датчик звука *Lego* выдает следующие показания: 4–5% соответствуют уровню шума в тихой жилой комнате; 5–10% – обычная речь, слышимая на среднем расстоянии; 10–30% – нормальный разговор вблизи датчика или музыка, воспроизводимая на нормальном уровне громкости; 30–100% – громкие крики или музыка. Подразумевается, что датчик расположен на расстоянии 1 м от источника звука.

Звуковая чувствительность человеческого уха сильно зависит от частоты звука. Пик чувствительности приходится на значение 3–5 кГц, а значения ниже 20 Гц и выше 20 кГц находятся уже за пределами физических возможностей. Датчик звука может быть переключен в дБА-режим, в котором значения измерений наиболее соответствуют диаграмме чувствительности человеческого уха.

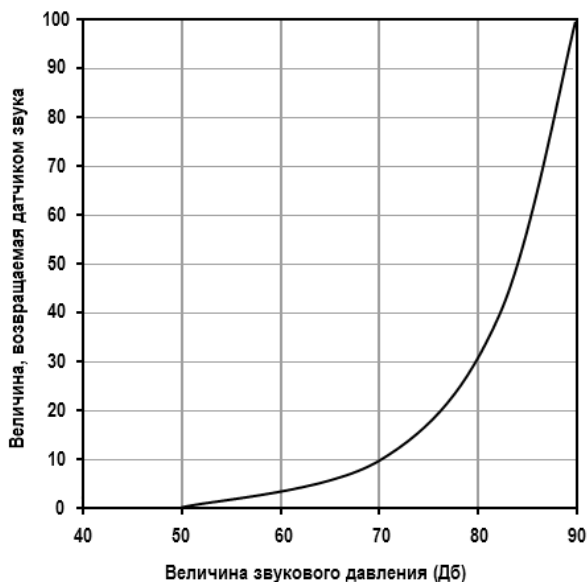


Рис. 41. Показания датчика звука и соответствующая величина звукового давления [25; 50]

5.8. Датчик ускорения

Датчик ускорения (*Accelerometer Sensor*) позволяет роботу определять результирующее ускорение по изменению скорости в каждом из трёх направлений, а также определять своё «отклонение» от горизонтали. Датчики ускорения относятся к датчикам внутреннего состояния, поскольку они не записывают данные из окружающей среды.

Работа датчика основана на пьезоэффекте – возникновении электрического заряда на поверхности диэлектрика (пьезоэлектрика) под действием механических напряжений. Когда робот изменяет скорость (ускоряется или замедляется), его инертность позволяет пьезоэлементу растягиваться или сжиматься [48]. Зная массу m , длину l , коэффициент пропорциональности q и измерив напряжение U , ускорение может быть рассчитано по формуле:

$$U \cdot q = m \cdot l \cdot a \quad \Rightarrow \quad a = \frac{U \cdot q}{m \cdot l}$$

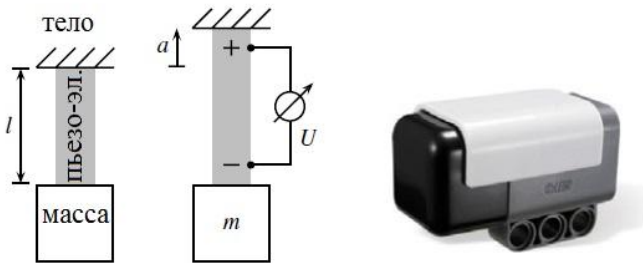


Рис. 42. Принцип работы датчика ускорения [48]. Внешний вид датчика ускорения HiTechnic

Датчик ускорения, разработанный компанией HiTechnic, позволяет измерять ускорение в диапазоне $\pm 19,62 \text{ м/с}^2$, т.е. вдвое превышающий ускорение свободного падения g ($g=9,81 \text{ м/с}^2$).

5.9. Датчик-компас

Датчик-компас (*Magnetic Compass Sensor*) определяет текущую ориентацию робота. Это электронный аналог обычного компаса, дает информацию об ориентации в пространстве в цифровой форме.

Работа электронного компаса (рис. 43) основана на магниторезистивном эффекте, т.е. изменении электрического сопротивления какого-либо материала под действием внешнего магнитного поля. Наиболее сильно подвержены этому явлению полупроводники.

Магниторезистивное сопротивление R_ϕ изменяет свое значение в зависимости от направления силовых линий магнитного поля, вследствие чего возникает разность потенциалов. Чем больше угол поворота датчика ϕ относительно силовых линий магнитного поля Земли, тем больше сопротивление. Точность измерения датчика составляет 1° , диапазон измеряемых значений от 0 до 359° .

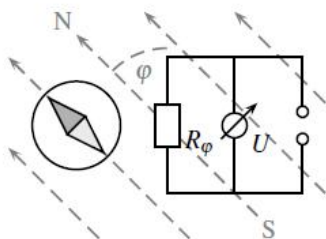


Рис. 43. Принцип работы датчика-компаса.
Внешний вид датчика-компаса *HiTechnic*

5.10. Датчик-гироскоп

Гироскопический датчик *Lego* может определять движение только вокруг одной оси вращения. С помощью датчика-гироскопа можно определить скорость вращения в градусах в

секунду, которую в свою очередь можно использовать для определения времени поворота робота или его части. Кроме того, гироскопический датчик регистрирует общий угол вращения в градусах. Таким образом можно управлять движением робота. Также с помощью этого датчика можно определить скорость вращения робота в градусах в секунду.



Рис. 44. Внешний вид датчиков-гироскопов Lego (EV3) и HiTechnic

Датчик для определения скорости вращения *HiTechnic* представляет собой также одноосный гироскоп, связанный с кварцевым резонатором [51]. С его помощью можно построить робота, который сможет реагировать на скорость поворота. Датчик *HiTechnic* позволяет обнаруживать вращательные движения транспортного средства *NXT* к скорости вращения $\pm 360^\circ / c$. Измеренная величина представляет собой скорость вращения в градусах в секунду.

Простейший механический гироскоп состоит из ротора (рис. 45) и карданного подвеса. При работе гироскопа его ротор с большой скоростью вращается вокруг оси *X*, закрепленной на подвижной внутренней рамке. Эта рамка, в свою очередь, может вращаться по оси *Y*, закрепленной на наружной рамке. Наружная рамка вместе с внутренней и ротором может вращаться по оси *Z*, которая прикреплена на шарикоподшипниках к основанию. Таким образом, ротор гироскопа имеет возможность свободно вращаться вокруг трех осей, т.е. он имеет три степени свободы. Система двух подвижных рамок и закрепленной в них

оси вращения ротора образует карданный подвес. Ось X , вокруг которой вращается ротор, называется главной осью гироскопа.



Рис. 45. Схема трехстепенного механического гироскопа

5.11. Инфракрасный датчик

Инфракрасный датчик от Lego для EV3 позволяет определять сигналы в инфракрасном диапазоне волн, посылаемые, например, с удаленного инфракрасного маяка (ИК-маяка) и посылать собственный инфракрасный сигнал и определять его отражение от других объектов.



Рис. 46. Внешний вид инфракрасного датчика и ИК-маяка для EV3

Инфракрасное излучение занимает в электромагнитном спектре диапазон между красной границей видимого спектра и микроволнами (от 750 нм до 1 мм), его часто называют тепло-

вым излучением. Этот тип сигнала используется в большинстве телевизионных пультов управления. Как и видимый свет, инфракрасное излучение блокируется находящимися на его пути объектами, поэтому между ИК-маяком и инфракрасным датчиком должно быть свободное пространство. Солнечный свет может влиять на инфракрасные сигналы, но обычное комнатное освещение не должно оказывать влияния.

Инфракрасный датчик имеет три разных режима: «Приближение», «Маяк» и «Дистанционное управление».

В режиме «Приближение» инфракрасный датчик посылает свой инфракрасный сигнал и может определить отражение этого сигнала от объекта, находящегося перед датчиком. Сила отраженного сигнала может использоваться для определения расстояния до объекта.

Значение приближения прямо не соответствует определенному расстоянию. Если объект находится на расстоянии менее 1 см, то датчик не в состоянии его зафиксировать. Датчик выдает относительные значения расстояния до объекта (0 значит очень близко, 100 – далеко). На показания датчика влияют цвет и материал объекта, а также другие факторы. Это обусловлено тем, что цветные объекты отражают инфракрасный свет лучше, чем темные объекты.

В режиме «Маяк» инфракрасный датчик от Lego может обнаруживать приблизительное положение дистанционного инфракрасного маяка (ИК-маяка) перед датчиком. Датчик может дать значение приближения маяка (относительного расстояния от датчика) и его направление (угол от направления, которое указывает датчик). Режим «Маяк» можно использовать, например, для того, чтобы заставить робота искать ИК-маяк и двигаться к нему.

В режиме «Дистанционное управление» инфракрасный датчик может определять нажатие кнопок на ИК-маяке.



Рис. 47. Детектор инфракрасного излучения *HiTechnic*.
Инфракрасный мяч *HiTechnic*

Датчик для обнаружения источника инфракрасного излучения *HiTechnic (InfraredSeeker)* посредством пяти инфракрасных сенсоров, направленных в разные стороны через каждые 60° , определяет, откуда исходят ИК-сигналы и их интенсивность. Инфракрасный мяч *HiTechnic* генерирует электромагнитное излучение в инфракрасном спектральном диапазоне. Питание мяча осуществляется от четырёх элементов питания типа ААА. Мяч функционирует в четырёх режимах работы: режим *A* – режим излучения затухающий, частота импульсов 1200 Гц, период – 833 м/с, время работы в данном режиме не менее 7 часов 30 мин; режим *B* – постоянное ИК-излучение, время работы в данном режиме не менее 1 часов 20 мин; режим *C* – частота импульсов 600 Гц, период – 1667 м/с, время работы в данном режиме не менее 3 часов 30 мин; режим *D* – частота импульсов 1200 Гц, период – 833 м/с, время работы в данном режиме не менее 3 часов 30 мин [51].

5.12. Датчик для обнаружения объектов

Датчик для обнаружения объектов *HiTechnic (EOPDSensor)* даёт возможность построить робота, который сможет обнаруживать объекты при помощи отраженных световых сигналов. Предусмотрено две настройки чувствительности в зависимости

от удаленности объекта. Встроенная функция автоматической коррекции окружающего освещения позволяет использовать его как при ярком освещении, так и в затемнённых помещениях [51].



Рис. 48. Датчик для обнаружения объектов HiTechnic

Задания для самостоятельной внеаудиторной работы

1. Постройте функции преобразования для датчиков звука и света *LegoMindstormsNXT*.
2. Заполните таблицу (для датчиков, используемых в лабораторном практикуме):

Таблица 4

Характеристики датчиков учебного робота

№	Датчик	Диапазон измерений	Погрешность измерений	Единицы измерения выходной величины

3. Используя список рекомендованной литературы и интернет-источники, составьте описание одного из датчиков (ультразвуковой датчик, инфракрасный датчик, датчик касания, датчик звука, датчик света, датчик цвета, датчик-компас, датчик-гироскоп, датчик ускорения), согласно приведенному ниже плану.

План описания датчика

1. Назначение датчика;
2. Принцип действия датчика (явление или закономерность, положенные в основу работы датчика);
3. Схема устройства датчика (его основные части, их назначение);
4. Правила использования датчика (условия корректной работы датчика);
5. Метрологические характеристики датчика (точность измерений, диапазон измерений, функция преобразования);
6. Область применения датчика.

§ 6. Исполнительная система робота

Исполнительная система робота включает манипуляционную систему и систему передвижения.

Основным типом манипуляционных систем являются механические манипуляторы, прототипом для разработки которых послужила рука человека. Рука представляет собой многозвенный механизм, имеющий несколько степеней свободы. Рука заканчивается кистью, предназначенной для захвата предметов.

По аналогии манипуляторы представляют собой пространственные механизмы в виде кинематических цепей из звеньев, образующих кинематические пары с угловым или поступательным относительным движением и системой приводов обычно раздельных для каждой степени подвижности. Манипуляторы заканчиваются как правило захватным устройством, которое служит для непосредственного взаимодействия с объектами внешней среды [19; 47].

Захватные устройства предназначены для того, чтобы брать объект, удерживать его в процессе манипулирования и освободить по окончании этого процесса. Из всех живых существ наиболее искусным в выполнении операция взятия са-

мых разнообразных предметов является человек. Приведем некоторые примеры захвата предметов кистью человека (рис. 49).



Пример
двухпальцевого
захвата

Пример
многопальцевого
захвата

Пример
ладонного
захвата

Рис. 49. Примеры захвата предметов кистью человека

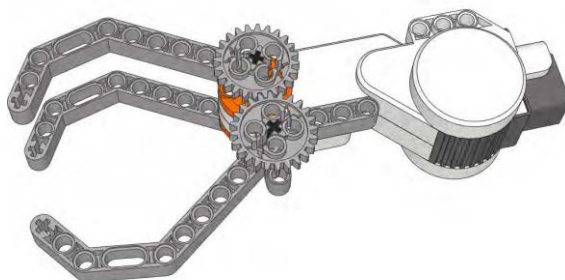


Рис. 50. Пример захватного устройства учебного робота [55]



Рис. 51. Пример захватного устройства учебного робота
в виде карабина [55]



Рис. 52. Примеры учебного робота с захватным устройством в виде крюка, «щеткой» [55]

Системы передвижения роботов также относятся к исполнительным системам наряду с манипуляционными системами. В современных мобильных роботах нашли применение практически все известные транспортные средства. Кроме того, предметом робототехники являются различные бионические способы передвижения, заимствованные у живой природы и не освоенные ещё в технике.

Мобильные роботы, предназначенные только для транспортных операций по перевозке грузов, – робокары – часто не имеют манипуляторов, а снабжены упрощенными погрузочно-разгрузочными устройствами. Последние операции могут выполняться также стационарными манипуляторами, находящимися в местах остановки транспортных роботов [47].

Можно выделить четыре принципиально различных типа мобильных роботов – наземные, летающие (воздухоплавающие), водоплавающие и подземные. Наиболее распространенными являются мобильные роботы, способные перемещаться каким-либо образом по поверхности земли. Они в свою очередь подразделяются на: колесные, гусеничные, шагающие, гибридные роботы. Последние в зависимости от обстоятельств могут либо шагать, либо катиться, либо одновременно шагать и катиться, т.е. пользоваться и колесами и ногами. Существует

также большое количество специализированных мобильных роботов, ориентированных на ограниченное применение. К их числу относятся рельсовые роботы, адсорбционные роботы (способные передвигаться по крутым участкам, цепляясь за поверхность с помощью вакуумных присосок) и др. [19].

Колесо является наиболее распространенным механизмом не только для передвижения робота, но и в искусственных транспортных средствах в целом. Использование колес позволяет достичь наибольшей эффективности при относительно простом конструктивном исполнении. При этом колесные мобильные роботы постоянно находятся в равновесии в результате постоянного контакта колес с поверхностью. Вследствие этого отпадает необходимость решения задачи устойчивости. Трех колес достаточно для обеспечения устойчивости.

Роботы на гусеничном ходу обладают значительно большей маневренностью при перемещении по пересеченной местности, чем роботы на колесной базе, так как они имеют большое количество точек контакта с опорной поверхностью. Однако, в связи с большой площадью контакта с землей, при изменении ориентации необходимо обеспечить скольжение между гусеницей и поверхностью, что требует больших усилий и приводит к невозможности прогнозирования точного положения центра вращения робота. При этом характеристики движения и точность ориентации очень сильно зависят от характеристик опорной поверхности. Таким образом, роботы на гусеничном ходу обладают хорошими маневренностью и устойчивостью, но плохой управляемостью. С точки зрения эффективности передвижения гусеничный ход рационально использовать при перемещении по рыхлому или сыпучему грунту.

Шагающие роботы в настоящее время не нашли широкого применения в промышленности, но они являются важным направлением долгосрочного исследования. Как и их биологические прототипы, шагающие роботы могут иметь разное количе-

ство ног (например, крупные млекопитающие и рептилии имеют четыре ноги, насекомые – шесть и более ног, а люди – две).

В последние годы успешно развиваются конструкции двуногих мобильных роботов. Они могут бегать, прыгать, ходить по лестнице и даже выполнять воздушные трюки, такие как сальто. Для этого разработаны специальные компактные серводвигатели для суставов, которые имеют большую мощность и развивают значительные крутящие моменты. Эти «умные» двигатели обеспечивают согласованное срабатывание различных суставов и имеют управление с обратной связью.



Рис. 53. Двуногий робот от *Sony* Рис. 54. Двуногий робот от *Honda*

Четвероногие роботы, стоя на месте, обладают статической устойчивостью, в то же время при перемещении у них возникают сложности, связанные с необходимостью перемещать центр тяжести робота во время движения.

Большой популярностью среди мобильных роботов пользуются шестиногие роботы благодаря их статической устойчивости при ходьбе.



Рис. 55. Четвероногий робот *AIBO* (*Sony*)



Рис. 56. Шестиногий робот *PhantomX*

§ 7. Модели учебных роботов на базе конструктора *LegoMindstormsNXT*

Робот, собранный из конструктора *LegoMindstorms*, представляет собой интеллектуальный блок (*NXT* или *EV3*), присоединенные к нему электронные компоненты (моторы и датчики),

а также конструкционные элементы (оси, балки, соединительные штифты, шестеренки и др.). Используя данные элементы можно собирать различные конструкции мобильных роботов и манипуляторов.

Следует учитывать, что первоначальное освоение робототехнического конструктора требует наличия готовых шаблонов [12]. Компания *Lego* предлагает большое количество моделей для сборки самых разнообразных типов роботов.

У новичков в образовательной робототехнике робот ассоциируется, как правило, с человекоподобным существом. Примеры таких моделей представлены ниже (рис. 57).



Рис. 57. Двухногие роботы *NXT* и *EV3*

Как правило, обучение образовательной робототехнике начинается с курса программирования мобильного робота – двухмоторной тележки. Имеется достаточно много вариантов построения такой тележки. Приведем некоторые из них (рис. 58, 59).

Имеется и более простой вариант двухмоторной тележки (рис. 60) [10; 43], конструирование которой занимает гораздо меньше времени и позволяет быстрее перейти к изучению приемов программирования робота. Корпус тележки расположен горизонтально, рулевое колесико вращается вокруг вертикальной оси свободно.

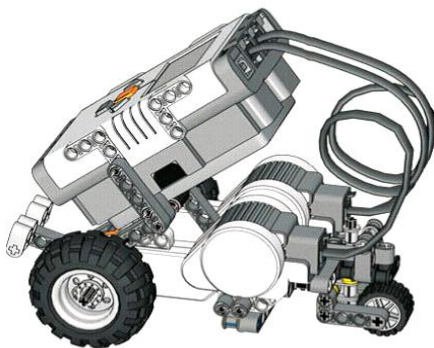


Рис. 58. Стандартная основа для робота из конструктора версии 9797



Рис. 59. Основа для робота из пособия «Введение в робототехнику» [28]



Рис. 60. Основа для робота («робот-пятиминутка») [9; 43]

В руководстве пользователя к конструктору версии *NXT* 2.0 представлена модель двухмоторного гусеничного робота (рис. 61).



Рис. 61. Двухмоторный гусеничный учебный робот

Кроме двуногих роботов разработчики предлагают собрать и четвероногого и шестиногого робота (рис. 62, 63).



Рис. 62. *NXTAPOD*
(автор *Daniele Benedettelli*)

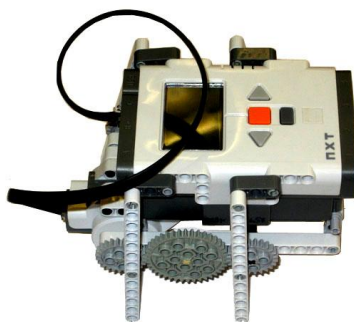


Рис. 63. Шагающий четвероногий
робот [43]

Организуя проектную деятельность по сборке различных моделей роботов, школьники могут собрать различных представителей животного мира (рис. 64).

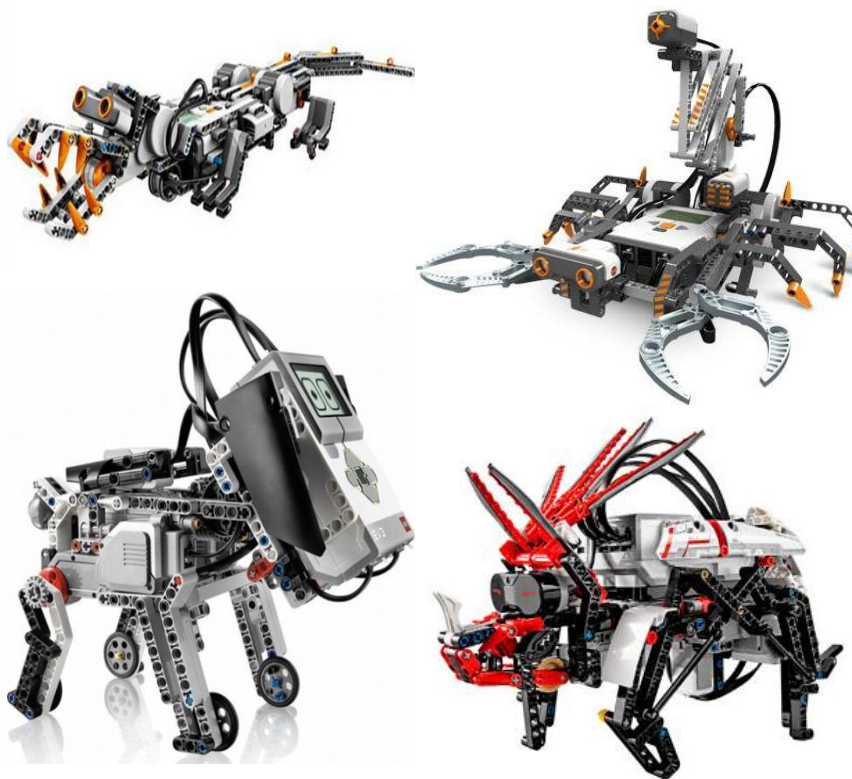


Рис. 64. Модели роботов-животных *NXT* и *EV3*
(аллигатор, скорпион, собака, динозавр)

Промышленные роботы являются одним из ключевых направлений развития робототехники. На рис. 65 представлены модели манипуляторов на основе *LegoMindstorms*.

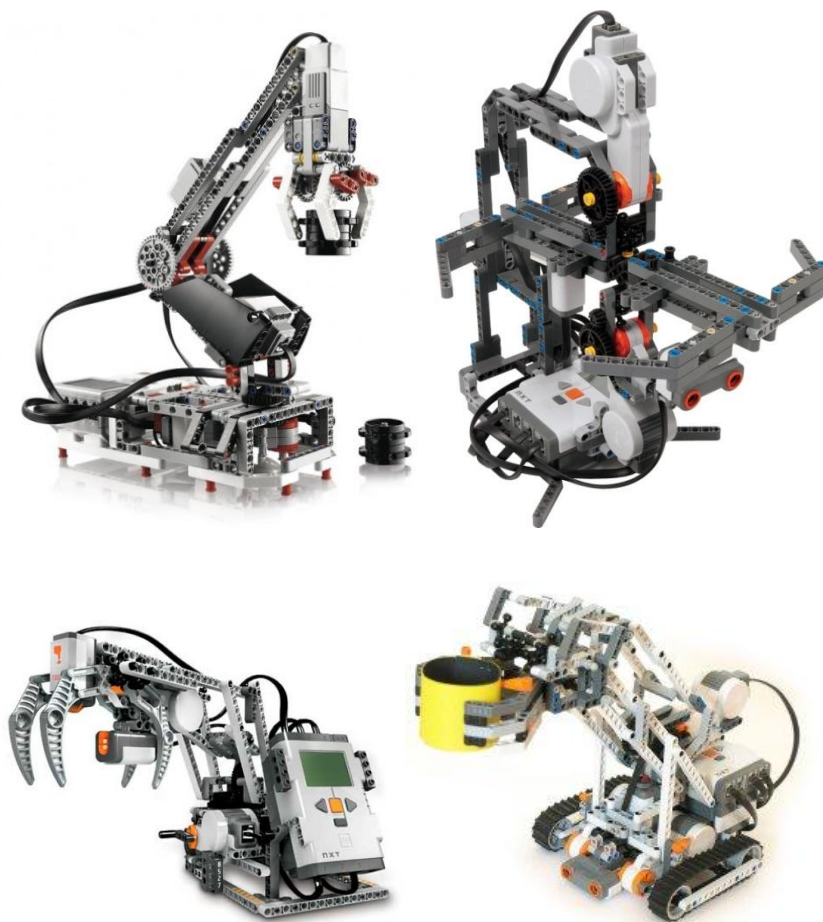


Рис. 65. Модели роботов-манипуляторов *NXT* и *EV3*

ГЛАВА III

РЕШЕНИЕ СТАНДАРТНЫХ ЗАДАЧ ДЛЯ УЧЕБНОГО РОБОТА

§ 8. Создание программ для робота

8.1. Свойства алгоритма.

Базовые алгоритмические конструкции

Программирование – неотъемлемый элемент процесса создания даже самого простого робота. Начальным этапом создания программы является алгоритм – точное и полное описание последовательности действий, позволяющее получить конечный результат.

Основные требования, предъявляемые к алгоритмам:

- дискретность: алгоритм должен представлять решение задачи в виде последовательности простых (или ранее определенных) шагов;
- определенность: шаги алгоритма должны допускать однозначную трактовку и быть понятными для исполнителя алгоритма;
- массовость: алгоритм должен давать решение не только для конкретного набора значений, а для целого класса задач, который определяется диапазоном возможных исходных данных;
- результативность: алгоритм должен давать конкретный результат, т.е. должны быть рассмотрены все возможные ситуации и для каждой из них получен результат;
- конечность: количество шагов алгоритма должно быть конечным.

Для описания алгоритма распространены следующие методы:

- словесный способ: изложение алгоритма ведется на обычном языке с разделением на последовательные шаги;

- блок-схемы (графический способ): графическое изображение алгоритма с помощью специальных значков-блоков, блоки соединяются между собой линиями переходов, которые определяют очередность выполнения действий;

- формальные алгоритмические языки (языки программирования): при записи алгоритмов используют строго определенный набор символов и составленных из них специальных зарезервированных слов, имеют строгие правила построения языковых конструкций;

- псевдокод: синтез алгоритмического и обычного языков. Элементы некоторого базового языка используются для строгой записи базовых структур алгоритма.

Любой алгоритм может быть построен комбинацией трех базовых логических структур:

- 1) следование;
- 2) ветвление (развилка);
- 3) цикл.

Логическая структура «следование» указывает на то, что управление передается последовательно от одного действия к другому (рис. 66).



Рис. 66. Последовательное выполнение шагов алгоритма на языке блок-схем

Логическая структура «ветвление» («развилка») используется, когда выполнение программы может измениться в зависимости от результата проверки условия и пойти двумя альтерна-

тивными путями. Условие является некоторым высказыванием и может быть истинным или ложным. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Различают полное и неполное ветвление. В первом случае, если условие истинно, выполняется действие 1, если ложно – действие 2. В случае неполного ветвления, если условие выполняется (является истинным), то выполняется действие 1, иначе ничего не происходит.

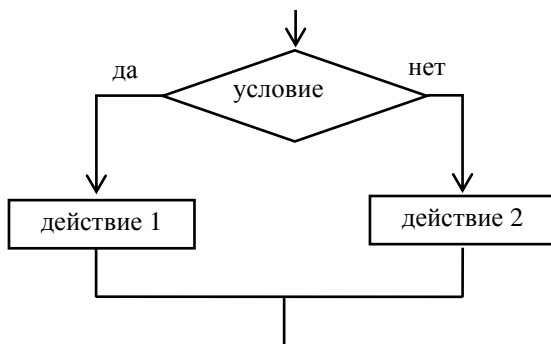


Рис. 67. Полное ветвление на языке блок-схем

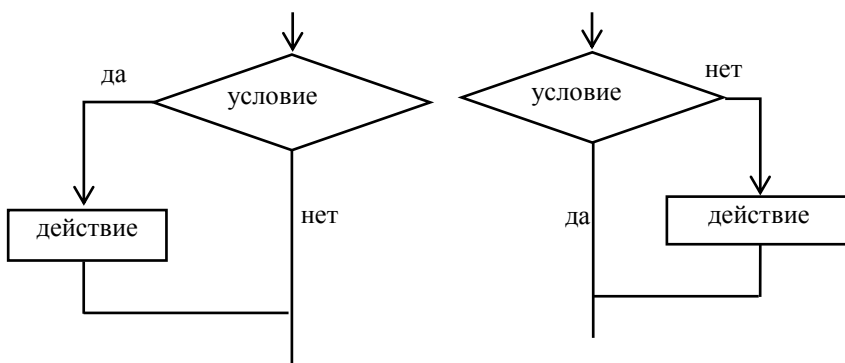


Рис. 68. Неполное ветвление на языке блок-схем

Одним из вариантов логической структуры «ветвление» в алгоритме служит оператор выбора. При этом может быть как полный, так и неполный выбор.

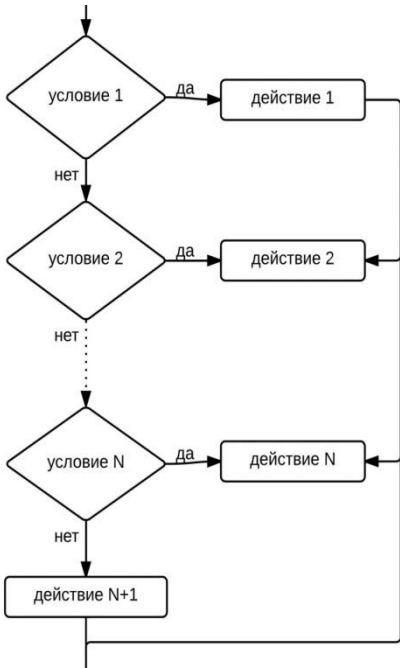


Рис. 69. Полный выбор на языке блок-схем

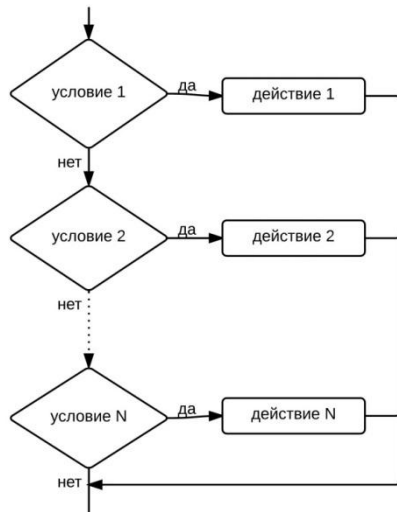


Рис. 70. Неполный выбор на языке блок-схем

Логическая структура «цикл» служит для записи алгоритмов, в которых некоторая часть алгоритма (тело цикла) должна повторяться несколько раз. Количество повторений цикла может определяться несколькими способами, поэтому различают три вида циклов.

1. Цикл с предусловием (или цикл «пока»). При реализации этого цикла сначала проверяется условие его выполнения, пока оно выполняется, будут происходить повторения тела цикла. Если условие не выполняется при первой проверке, то тело

цикла не будет выполняться вообще. После выхода из цикла управление передается следующей структуре.

2. Цикл с постусловием (или цикл «до»). При реализации этого цикла условие завершения цикла проверяется после выполнения тела цикла. Поэтому тело цикла выполняется всегда хотя бы один раз. Пока условие не выполнено, тело цикла будет повторяться (выполнение условия является условием окончания цикла).



Рис. 71. Цикл с предусловием на языке блок-схем



Рис. 72. Цикл с постусловием на языке блок-схем

3. Цикл с параметром. Этот вид цикла удобно использовать, когда заранее известно количество повторений цикла. Вводится понятие счетчика цикла, который используется для организации этого цикла. Необходимо задать верхнюю и нижнюю границы изменения счетчика цикла. В зависимости от значения верхней и нижней границы определяется шаг (1 или -1), т.е. значение счетчика цикла.

Операция присваивания является важной при составлении алгоритмов. Эта операция обозначается «:=», она служит для вычисления выражений, стоящих справа, и присваивания их значений переменным, стоящим слева. Например, исходное зна-

чение переменной x составляет 5, после выполнения команды $x:=x+1$, значение переменной x изменяется на 6.

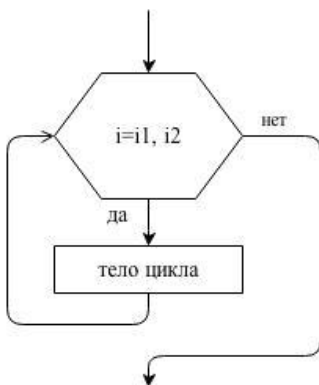


Рис. 73. Цикл с параметром на языке блок-схем

Алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. В этом случае язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись языка на этом языке – программой.

Разработка алгоритмов решения задач на компьютере требует определённых навыков. При построении алгоритма нужно стремиться к тому, чтобы запись алгоритма была понятной и наглядной. Внося изменения в алгоритм, желательно не перестраивать его полностью. Эти требования можно соблюсти, если придерживаться структурного подхода. Самым важным преимуществом структурного подхода является возможность нисходящего программирования, благодаря которой можно двигаться от крупных задач к более мелким. Крупная задача будет разбиваться на менее крупные блоки, те в свою очередь – на меньшие блоки и т.д. Каждый блок алгоритма должен быть максимально самостоятельным и логически завершённым. Разбиение на блоки должно определяться внутренней логикой задачи.

Процесс разбиения завершается, когда решение исходной задачи сводится к решению ряда простых задач, для которых легко построить соответствующие алгоритмы. На каждом шаге этого процесса происходит детализация, т.е. переход от общих задач к частным, которые в свою очередь детализируются в виде конкретных подзадач. Такой метод называется методом пошаговой детализации. Таким образом, при структурном подходе можно комбинировать не только базовые структуры (следование, ветвление, цикл), но и подключать алгоритмы, написанные ранее. Алгоритмы, которые целиком используются в составе других алгоритмов, называются вспомогательными. Если вспомогательный алгоритм в процессе работы программы выполняется многократно, то обычно прибегают к оформлению вспомогательного алгоритма в виде подпрограммы.

«Процесс восприятия» компьютером программы, написанной на некотором формальном языке (высокого уровня) называется трансляцией. В процессе трансляции программы происходит её компиляция и интерпретация. Компиляция – трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду. Интерпретация программы – покомандный анализ, обработка и тут же выполнение исходной программы (в отличие от компиляции, при которой программа транслируется без её выполнения).

8.2. Программное обеспечение для образовательной робототехники

Для программирования роботов *Lego Mindstorms NXT* используются как графические, так и текстовые языки программирования. К графическим относятся такие языки как *NXT-G*, *Robolab*, *LabView Education Edition*, к текстовым – *RobotC*, *NXC*, *Phyton* и др. Графические языки программирования предпола-

гают создание программ для робота в виде наглядных блок-схем и позволяют даже школьникам младшего возраста обучаться основам программирования. Текстовые языки программирования предоставляют больше возможностей для работы с формулами, большими математическими выражениями, они являются адаптированными версиями языков программирования используемых в профессиональной деятельности инженерами и программистами.

Базовым программным обеспечением робота *Lego Mindstorms NXT* является среда программирования *Lego Mindstorms NXT Software* (язык программирования *NXT-G*). Программное обеспечение *Lego Mindstorms NXT* было полностью разработано в среде графического проектирования *LabVIEW*, созданного компанией *National Instruments* для построения автоматизированных систем измерений и управления.

Создание программы управления роботом при использовании *NXT-G* очень похоже на создание блок-схемы. Программист создает схему поведения робота с помощью имеющихся блоков и задает характеристики его поведения. Данная среда наглядна и очень проста в использовании, что позволяет быстро освоить её человеку с любым уровнем подготовки. Однако функциональные возможности *NXT-G* весьма ограничены. Кроме того, она требует для своей работы значительных ресурсов компьютера. Это является существенной преградой для использования этой среды при разработке сложных программ.

Robolab – ещё одна графическая среда программирования, используемая для преподавания образовательной робототехники в школах. Эта среда была создана для *Lego*-робота на базе микрокомпьютера *RCX*, до сих пор пользуется большой популярностью для программирования *Lego*-робота на базе микрокомпьютера *NXT*.

На сегодняшний день популярной становится среда для программирования *LabVIEW* [от англ. *Laboratory Virtual Instru-*

mentationEngineeringWorkbench (Среда разработки виртуальных приборов)]. Эта среда применяется во многих отраслях промышленности, включая космические. Школьная версия программной среды графического программирования *LabVIEW Education Edition* представляет широкие возможности для проектной и научно-исследовательской деятельности учителя и учащихся, поскольку является простой и интуитивно понятной системой пользования.

RobotC for Mindstorms – текстовая среда программирования, позволяющая разрабатывать программы для управления *Lego Mindstorms NXT*, используя популярный язык программирования *C*. *RobotC* ориентирована как на новичков, так и на продвинутых пользователей. Она имеет два режима работы: базовый режим и режим для специалистов. *RobotC* имеет мощный интерактивный отладчик в режиме реального времени, что значительно сокращает время, необходимое для отладки программ. Однако, несмотря на то, что с помощью этой среды можно составлять довольно сложные и эффективные программы, для тех, кто не знаком с языком *C*, текстовое программирование может оказаться недостаточно наглядным.

При всем своем различии все эти языки имеют много общего и, в принципе, эквиваленты с точки зрения потенциальной возможности написать одну и ту же программу на любом из них.

8.3. Среда программирования Lego Mindstorms NXT Software

Lego Mindstorms NXT Software – это программное обеспечение, которым комплектуется конструктор *LegoMindstorms NXT*. Данное программное обеспечение базируется на широко известном пакете графического программирования *LabVIEW*. Основным применяемый метод – «*drag-and-drop*» (перетащи-и-

бросил) – предполагает составление алгоритма функционирования робота из логических блоков в специальном визуальном редакторе (рис. 74).

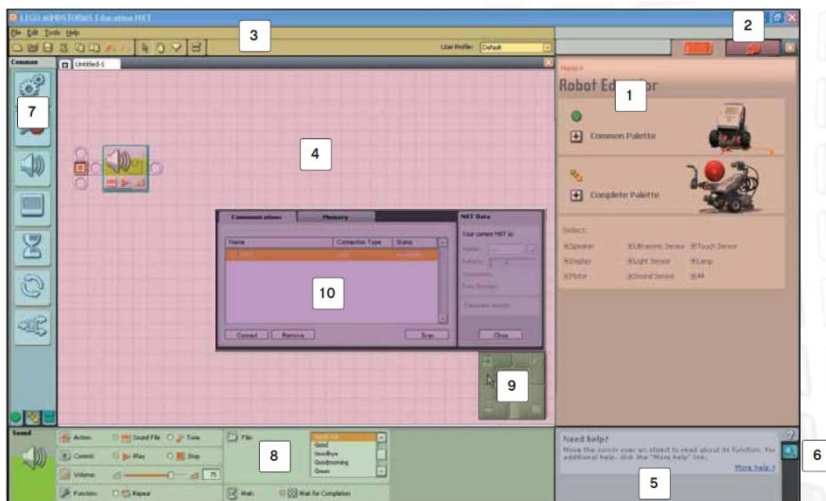


Рис. 74. Внешний вид окна стандартного программного обеспечения для *Lego Mindstorms NXT* (v. 9797)

Окно программы содержит следующие разделы:

1. Самоучитель *RobotEducator* – инструкции по сборке и программированию.
2. Мой портал – ссылка на сайт www.mindstormseducation.com, где можно скачать инструменты, загрузить программы и получить информацию;
3. Панель инструментов предназначена для быстрого вызова часто используемых команд;
4. Рабочее поле – область экрана, где составляются программы. Перетаскивайте блоки из Палитры на рабочее поле и выстраивайте из них программу;
5. Окно подсказок – область, где появляются контекстные подсказки;

6. Карта рабочего поля – служит для перемещения по рабочему полю и для получения общего вида программы;

7. Палитра команд – область размещения блоков команд, из которых создаются программы;

8. Панель настроек – отображается для каждого программного блока, позволяет описать команду, которую этот блок должен выполнить;

9. Пульт управления – пять кнопок, которые позволяют загружать программы (или части программ) в микрокомпьютер *NXT*, а также изменять его настройки;

10. Окно *NXT* – всплывающее окно, в котором отображается информация о памяти *NXT* и его соединениях.

В каждой версии стандартного программного обеспечения *Lego Mindstorms NXT* имеется «Самоучитель» (*RobotEducator*) или «Робоцентр» (*RoboCenter*), где представлена серия примеров для сборки и программирования моделей роботов.

Конструктор v.9797 предназначен для образовательных целей, там представлен раздел «Самоучитель» (рис. 75), где собрана серия учебных материалов, которые показывают, как запрограммировать двухмоторного робота, используя основные возможности программного обеспечения *NXT-G*. В разделе «Основная палитра» представлены примеры программ:

1) «Воспроизводить звук»; 2) «Применить дисплей»; 3) «Двигаться вперед»; 4) «Двигаться назад»; 5) «Ускорить»; 6) «Разворот»; 7) «Поворот»; 8) «Ехать по квадрату» и др.

В наборе *NXT 2.0* представлен раздел «Робоцентр», где предлагается построить 4 типовые модели, каждая из них программируется по-своему:

- *ShooterBot*: робот, стреляющий шариком по движущимся объектам;

- *Robogator*: похож на аллигатора. Он щелкает пастью на все попадающиеся на пути объекты;

- *ColorSorter*: сортирует шарики в зависимости от их цвета по разным дорожкам;
- *AlphaRex*: робот-гуманоид, который может ходить вперед, назад, поворачиваться, говорить, брать предметы, танцевать.



Рис. 75. Раздел «Самоучитель»
(ПО для v.9797)

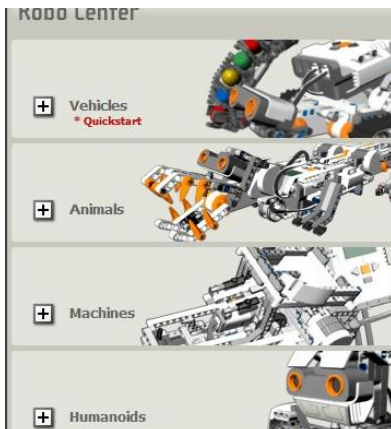


Рис. 76. Раздел «Робоцентр»
(ПО для NXT 2.0)

Задания для самостоятельной внеаудиторной работы

1. Используя рекомендованную литературу и интернет-источники, составьте сравнительную характеристику сред программирования *Lego*-робота (возраст учащихся, преимущества, недостатки, степень сложности освоения, скорость работы и др.).

2. Используя раздел «Справка» *Lego Mindstorms NXT Software*, а также рекомендованную литературу и интернет-источники, ознакомьтесь с назначением блоков «Основной» и «Полной палитры команд», с назначением команд Меню и Панели инструментов.

§ 9. Задача движения робота

9.1. Физические характеристики механического движения

Движение – одна из важнейших задач для робота. С физической точки зрения движение (механическое движение) – это изменение положения тела в пространстве относительно других тел с течением времени. Механическое движение может быть прямолинейным или криволинейным, равномерным или неравномерным (равноускоренным или равнозамедленным). Механическое движение в физике рассматривается для двух идеализированных объектов материальной точки и абсолютно твердого тела, при этом механическое движение робота будет рассматриваться как движение абсолютно твердого тела.

Твердое тело можно определить как систему материальных точек, расстояния между которыми неизменны. К основным видам движения твердого тела относятся поступательное и вращательное.

Поступательное движение – движение, при котором все точки тела имеют одинаковые траектории (рис. 77). Вращательное движение – это движение, при котором траектории различных точек тела (или системы тел) представляют собой окружности (или дуги окружностей) с общей осью.

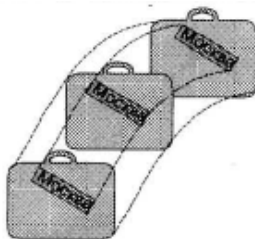


Рис. 77. Поступательное движение

Механическое движение характеризуется такими физическими величинами, как скорость, ускорение, пройденный путь, перемещение, траектория.

Траектория – линия, вдоль которой движется тело. По виду траектории движения разделяются на прямолинейные и криволинейные. Пройденный путь – длина траектории движущегося тела, измеряется в метрах. Перемещение – это вектор, соединяющий начальное и последующее положения тела. Траектория и перемещение измеряются в метрах (система СИ), наглядно разница между этими терминами представлена на рис. 78.



Рис. 78. Соотношение понятий «перемещение» и «траектория»

Скорость характеризует быстроту изменения положения тела, рассчитывается как отношение перемещения к промежутку времени, за которое оно было совершено. Ускорение – характеризует быстроту изменения скорости тела, вычисляется как отношение изменения скорости к промежутку времени, за которое оно было совершено.

При движении робот (точнее моторы робота) совершает работу. Эта работа идет на преодоления сил трения и на совершение оборотов двигателями. Любое механическое устройство характеризуется такой физической величиной, как мощность – величина, показывающая быстроту совершения работы по отношению к отрезку времени, за который она была совершена. Мощность моторов *NXT* зависит от уровня заряда батареи *NXT*,

значительно увеличить или уменьшить мощность робота можно с помощью механических передач.

9.2. Блок «Движение»

Робот, на примере которого мы будем рассматривать реализацию движения, представляет собой двухмоторную тележку, имеющую два ведущих колеса и одно рулевое. Движение в *NXT-G* реализуется с помощью блока «Движение», основные характеристики которого, задаваемые на панели настройки блока, отображаются и на самом блоке: 1) буквы в правом верхнем углу блока показывают, какие из портов устройства *NXT* будут контролироваться; 2) пиктограмма показывает направление движения робота; 3) пиктограмма показывает уровень мощности; 4) пиктограмма показывает, какое значение установлено для характеристики «Длительность движения»: градусы, обороты, секунды, «без ограничения».

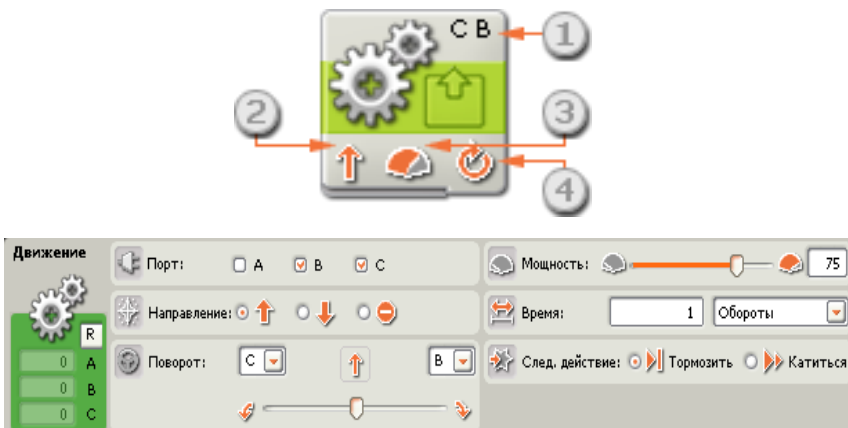


Рис. 79. Блок «Движение» в среде *NXT-G* и панель настроек блока «Движение» (по умолчанию)

Блок «Движение» позволяет управлять движением одновременно трех моторов. По умолчанию всегда установлено управления двумя моторами (*B* и *C*), что характерно для классической двухмоторной тележки. Моторы могут быть запрограммированы на вращение вперед, назад, остановку.

Рулевое управление осуществляется перемещением ползунок влево или вправо. В зависимости от того насколько сдвинут ползунок, можно осуществить плавный или резкий поворот:

Мощность мотора задается в условных единицах в значениях от 0 до 100. Этот параметр позволяет регулировать скорость вращения мотора. Следует иметь в виду, что при низких скоростях двигателя может не хватить крутящего момента, чтобы заставить робота двигаться. Вообще говоря, уровни мощности 10–100 подходят для большинства конструкций роботов.

Наиболее важным свойством в окне конфигурации блока «Движение» является «Длительность движения», которая может задаваться в градусах, в секундах, в оборотах колеса и «без ограничения». Таким образом, расстояние, на которое должен перемещаться робот, не может задаваться в явном виде. Также расстояние это напрямую будет зависеть от радиуса колес робота.

9.3. Кинематика учебного мобильного робота

Кинематика – раздел механики, изучающий математическое описание (средствами геометрии, алгебры, математического анализа) движения идеализированных тел (материальная точка, абсолютно твердое тело, идеальная жидкость) без рассмотрения причин движения (массы, сил и др.). Кинематика является одним из основных этапов исследований при проектировании мобильных роботов. Результатом кинематического анализа является математическое описание поведения механической системы для дальнейшей разработки программного управления движением учебного робота.

Рассмотрим модель учебного робота в виде двухмоторной тележки на базе конструктора *LegoMindstormsNXT* (рис. 80). Робот моделируется как твердое тело на колесах, перемещающееся по горизонтальной плоскости.

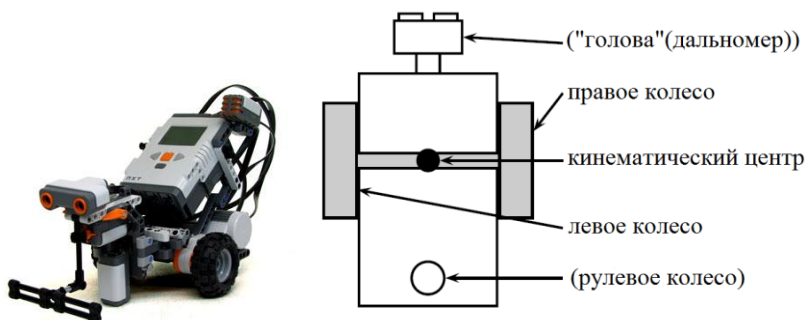


Рис. 80. Базовое шасси для учебного робота *Lego Mindstorms NXT*.
Схема базового шасси робота для кинематического анализа [48]

Для написания программы движения данного робота в *LegoMindstormsNXTSoftware* необходимо указывать длительность работы моторов, которая может задаваться в градусах, оборотах, секундах. Рассмотрим пример выполнения роботом задания: проехать прямо 50 см и повернуть на угол 90° .

Начнем с задачи прямолинейного движения робота. Важный параметр – длительность работы моторов – подбирается методом проб и ошибок либо рассчитывается по формуле, полученной на основе кинематического анализа. Для обеспечения наибольшей точности движения робота будем рассчитывать длительность работы моторов в градусах. Так за один оборот колеса мотор совершает поворот 360° , т.е. $2 \cdot \pi \cdot r = 360^\circ$ (где, r – радиус колеса).

Длительность работы моторов (в градусах) $\varphi_{пр}$ и $\varphi_{лев}$ рассчитывается по формуле согласно чертежу (см. рис. 81),

где s – расстояние, на которое должен переместиться робот (в нашем случае $s = 50$ см), r – радиус колеса ($r = 2,8$ см для модели робота представленной на рис.). В результате получаем $\varphi_{np} = \varphi_{лев} = 1023^\circ$.

$$\varphi_{np} = \varphi_{лев} = s \cdot \frac{360^\circ}{2 \cdot \pi \cdot r}$$

Для того чтобы робот повернулся на 90° ($\varphi_P = 90^\circ$) влево, нам необходимо знать d – расстояние между осями колес робота ($d = 11,5$ см для модели робота представленной на рис. 81), r – радиус колеса ($r = 2,8$ см для модели робота представленной на рис. 81). Если робот поворачивается вокруг левого колеса, то в этом случае $\varphi_{лев} = 0$, а φ_{np} рассчитывается по формуле:

$$\varphi_{np} = \varphi_P \cdot \frac{d}{r}$$

В результате расчётов получаем продолжительность работы правого мотора 370° .

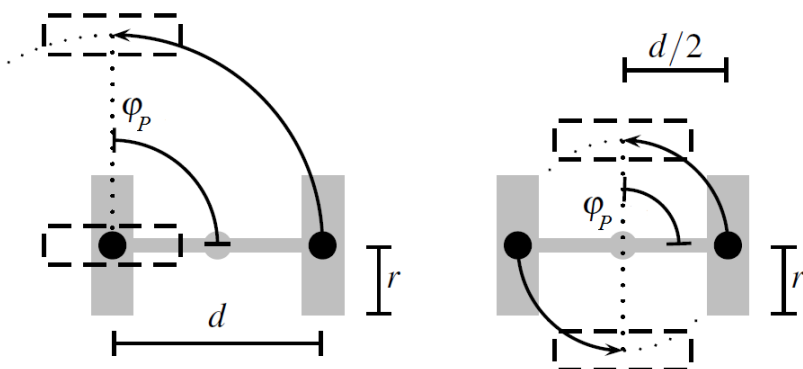


Рис. 81. Схема осуществления поворота роботом вокруг левого колеса, вокруг собственного центра (на месте)

Поворот робота вокруг собственного центра (поворот на месте) осуществляется таким образом, что правый мотор работает в прямом направлении, а левый в обратном, при этом длительность работы моторов будет одинаковой. Согласно схеме, представленной на рисунке, длительность работы моторов будет рассчитываться по формулам:

$$\varphi_{np} = \varphi_P \cdot \frac{d/2}{r}; \quad \varphi_{лев} = -\varphi_P \cdot \frac{d/2}{r}.$$

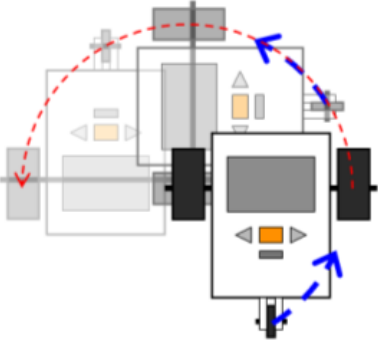
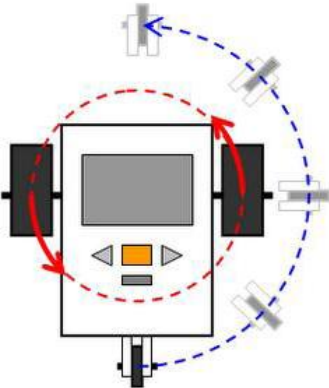
В результате расчетов получим $\varphi_{np} = 188^\circ$, $\varphi_{лев} = -188^\circ$. Знак « \leftarrow » означает, что левому мотору в программе необходимо задать направление «назад».

Полученные математическим путем значения длительности работы моторов не обеспечивают максимальную точность движения робота. Необходимо учитывать инерционность моторов, силы трения, которые возникают между колесами робота и поверхностью, по которой он перемещается, уровень заряда аккумулятора (или батареек) *NXT* блока и др. Поэтому, как правило, полученные значения корректируются в конкретных заданных роботу условиях.

Выбор способа поворота вокруг одного из колес либо на месте зависит от характера решаемой задачи. Если робот совершает поворот вокруг одного колеса, то точность такого поворота намного выше, в то же время он занимает гораздо больше пространства для маневра, и скорость его выполнения гораздо ниже, чем при повороте на месте.

Таблица 5

Сравнение способов выполнения поворота двухмоторной тележкой

Поворот вокруг одного колеса	Поворот на месте
	
$\varphi_{np} = \varphi_P \cdot \frac{d}{r} \varphi_{лев} = 0$	$\varphi_{np} = \varphi_P \cdot \frac{d/2}{r} \varphi_{лев} = -\varphi_P \cdot \frac{d/2}{r}$
Пространство поворота	
больше	Меньше
Скорость поворота	
меньше	Больше
Точность поворота	
больше	Меньше

Задание для самостоятельной внеаудиторной работы

1. Используя рекомендованную литературу и интернет-источники, подготовьте реферат на тему «Виды механических передач».

2. Используя рекомендованную литературу и интернет-источники, подготовьте реферат на тему «Движение робота по квадрату».

3. Предложите план решения следующих экспериментальных заданий:

- демонстрация равномерного и равноускоренного движения робота;

- определение расстояния, которое проходит робот за время одного оборота колеса;
- измерение величины силы трения при движении двухмоторной тележки на различных поверхностях.

§ 10. СЛЕДОВАНИЕ ПО ЛИНИИ

10.1. Базовый алгоритм следования по линии (релейный регулятор)

Задача следования вдоль черной линии наиболее часто встречается в робототехнических соревнованиях как отдельный вид состязания «Траектория» и как одна из составляющих общей задачи состязаний (см. § 8).

Алгоритм следования робота по линии изучается на примере простейшей двухмоторной тележки с двумя ведущими колесами и одним рулевым колесом. Датчик освещенности, позволяющий отслеживать роботу край линии, направлен вниз (рис. 82).

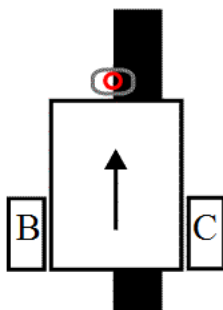


Рис. 82. Робот, следующий по линии

Наиболее очевидный и часто используемый алгоритм на языке *NXT-G* заключается в том, что робот ориентируясь на показания датчика освещенности (цвета), т.е. постоянно сравнивая измеренное им значение с пороговым значением

(обозначающим границу черного и белого участков поля), попеременно работает то левым, то правым моторами, совершая пошатывающиеся движения вперед. Такой алгоритм называют «классическим», с него начинают изучение движения по черной линии. Если рассматривать его как регулятор, то его называют релейным регулятором, т.е. управление осуществляется по принципу открыт/закрыт (в каждый момент времени оба мотора находятся попеременно в двух состояниях – включен/выключен). Графичеки идею релейного регулятора в алгоритме следования по черной линии можно представить следующим образом(рис. 83).

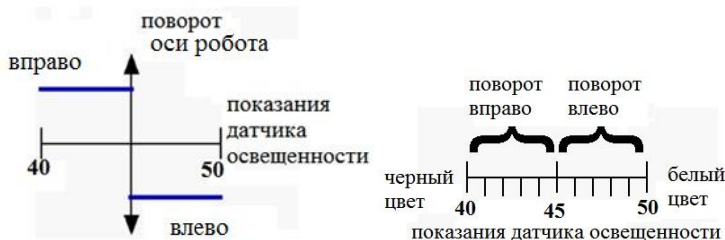


Рис.83. Идея алгоритма релейного регулятора для следования двухмоторного робота с одним датчиком освещенности по линии [54]

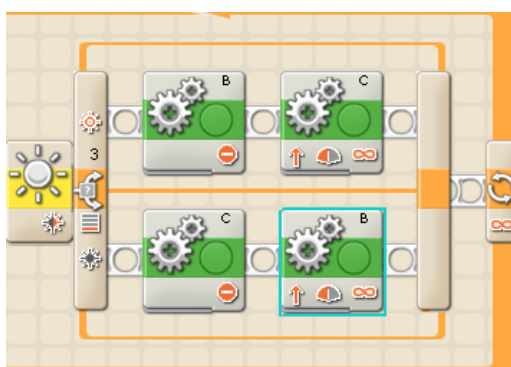


Рис. 84. Программа алгоритма следования по линии
(Lego Mindstorms NXT Software)

Данный алгоритм обладает рядом недостатков, которые влияют на качество выполнения роботом поставленной ему задачи (быстрее остальных роботов добраться по намеченной траектории до финиша) [43;44]:

- 1) в каждый момент времени один из моторов не движется;
- 2) робот может проскакать черную линию из-за высокой скорости моторов;
- 3) в результате изменения уровня освещенности помещения робот может проскакать линию и крутится на месте, требуется дополнительная отладка программы (установка подходящего внешним условиям порогового значения уровня освещенности).

10.2. Оптимизация базового алгоритма следования по линии

Улучшить работу робота по релейному алгоритму можно исходя из следующих соображений. Необходимо учитывать отличия между зрением робота и зрением человека. Робот воспринимает не цвет, а интенсивность отраженного света (см. § 3), попадающего на светочувствительный элемент датчика освещенности.



Рис. 85. Отличия между зрением робота и человека [43; 44]

Пороговое значение (среднее арифметическое между показаниями датчика на темной и светлой поверхности) воспринимается условно как «серый цвет». Белую поверхность робот определяет, получая с датчика освещенности значения меньше порогового (заложенного в программе), темные – выше порогового. Можно выделить определенный диапазон значений, которые робот воспринимает как «серый цвет», в этом случае робот должен ехать прямо, т.е. работают оба мотора в одном направлении с одинаковой мощностью (рис. 85).



Рис. 86. Графическое представление оптимизации классического алгоритма следования по линии [54]

Тогда алгоритм следования по линии немного усложнится, но робот проедет заданную траекторию быстрее (рис. 53).

10.3. Использование ПИД-регулятора в алгоритме следования по линии

ПИД-регулятор (пропорциональный интегрально-дифференциальный регулятор) – это метод, широко используемый для улучшения работы различных технических устройств (роботы, транспортные средства, ракеты и др.). Математическое описание данного метода приводится, как правило, в теории автоматического управления. Регулирование является частным случаем управления, цель его заключается в поддержании на

заданном уровне одного или нескольких выходов объекта управления, т.е. системы, в которой происходит подлежащий управлению процесс. Выходной сигнал ПИД-регулятора ($u(t)$) зависит от ошибки регулирования ($e(t)$), от интеграла от этой ошибки и от производной от этой ошибки.

$$u(t) = K_1 \cdot e(t) + K_0 \int_0^t e(t) \cdot dt + K_2 \cdot \frac{de(t)}{dt},$$

где K_1 – коэффициент усиления пропорциональной части, K_0 – коэффициент усиления интегральной части, K_2 – коэффициент усиления дифференциальной части.

Если какие-то из составляющих не используются, то регулятор называют пропорционально-интегральным (ПИ-), пропорционально-дифференциальным (ПД-), пропорциональным (П-).

Усовершенствуем с помощью ПИД-регулятора исходный алгоритм движения по черной линии, описанный в п. 11.1. Для этого рекомендуется по отдельности добавлять в алгоритм каждую из трех поправок (пропорциональную, интегральную, дифференциальную) и на каждом шаге отлаживать работу робота.

Пропорциональный регулятор – это устройство, оказывающее управляющее воздействие $u(t)$ на объект пропорционально его линейному отклонению $e(t)$ от заданного состояния $x_0(t)$ [43; 44]:

$$e(t) = x_0(t) - x(t),$$

где $x(t)$ – состояние в данный момент времени;

$$u(t) = k \cdot e(t),$$

где k – усиливающий коэффициент (коэффициент пропорциональности).

В случае классического алгоритма в каждый момент времени работает только один мотор (левый или правый). В случае пропорционального регулятора: чем дальше робот отклоняется от заданного курса, тем активнее должны работать

моторы, выравнивая его. Представим идею пропорционального регулятора графически (рис. 87):

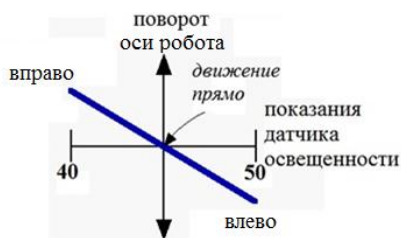


Рис. 87. Графическое представление идеи пропорционального регулятора в алгоритме следования по линии [54]

По оси X отложены показания датчика освещенности (от 40 до 50), в пределах которых его показания будут считаться серыми. В качестве порогового значения выбрано 45. Таким образом, ошибка будет составлять ± 5 . Ось Y , проходящая через пороговое значение, отвечает за графическое изображение поворота оси робота (влево или вправо). В первом случае «крутизна» поворота робота постоянна, во втором она меняется по линейному закону (т.е. пропорционально показаниям датчика).

Очевидно, что «крутизна» поворота робота является линейной функцией, зависящей от показаний датчика освещенности. Уравнение прямой описывается выражением $y = k \cdot x + b$, в нашем случае $y = k \cdot x$ ($b = 0$). Коэффициент k – это угол наклона прямой, рассчитываемый по формуле $k = dy/dx$, с другой стороны – это усиливающий коэффициент пропорциональности. Установим требование для значений этого коэффициента, он должен изменяться в пределах от 0 до 1.

Построим диаграмму, отражающую суть пропорционального регулятора. На оси X будет отложена ошибка в интервале от -5 до 5 , по оси Y – коэффициент «крутизны» поворота в интервале от -1 (крутой поворот налево) до 1 (крутой поворот направо).

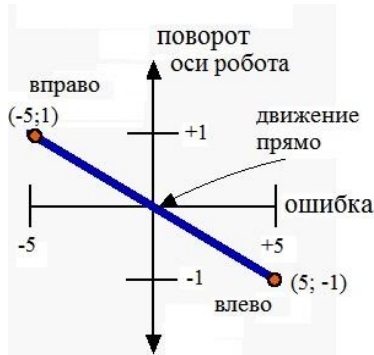


Рис. 88. Диаграмма, отражающая математическое описание пропорционального регулятора [54]

Рассчитаем значение коэффициента k для использования его в программе. Возьмем крайние точки графика $(-5; 1)$ и $(5; -1)$:

$$k = (\Delta y) / (\Delta x) = (1 - (-1)) / (-5 - 5) = 2/10 = 0,2.$$

Как видно, k зависит от выбранного диапазона ошибки (он выбран нами условно). Вне диапазона от -5 до $+5$ мы не можем сказать, как далеко датчик освещенности от края линии. Как только датчик освещенности становится слишком далеко от края линии, показания датчика больше не пропорциональны ошибке.

Диапазон, в котором датчик дает пропорциональный ответ, называется «пропорциональный диапазон». В нашем случае аналогом пропорционального диапазона для датчика освещенности являются его показания в интервале от 40 до 50 и ошибке от -5 до $+5$. Для двигателей пропорциональный диапазон от -100 (полная мощность в обратном направлении) до $+100$ (максимально мощное движение вперед).

Таким образом, пропорциональный регулятор эффективно работает только для малых углов отклонения, поэтому робота надо ставить в направлении движения так, чтобы датчик оказался по левую сторону от черной линии [43; 44].

Изложенное позволяет утверждать, что пропорциональное управляющее воздействие на моторы можно осуществить по следующему закону:

$$e = s_I - grey,$$

где s_I – текущие показания датчика, а $grey$ – заданное значение, например $grey = 45$.

$$u = k * e,$$

$$MotorB = N + u,$$

$$MotorC = N - u,$$

где N – базовая мощность двигателей, например $N = 50$.

Нетрудно заметить, что движение по линии на П-регуляторе отличается плавностью, и на некоторых участках робот движется практически прямолинейно или точно повторяя изгибы линии. При достаточно большом усиливающем коэффициенте движение робота напоминает движение при релейном регуляторе (робот заносит из стороны в сторону). При низком коэффициенте робот может потерять линию. Для достижения максимального эффекта от П-регулятора следует подобрать оптимальные соотношения параметров конструкции робота, базовой мощности моторов N и усиливающего коэффициента k [43; 44].

Включение интегральной и дифференциальной составляющих в алгоритм следования по линии описано в источниках [43; 44; 54].

Задание для самостоятельной внеаудиторной работы

1. Используя рекомендованную литературу и интернет-источники, подготовьте реферат на тему «Изучение алгоритма следования по линии».

2. Используя рекомендованную литературу и интернет-источники, подготовьте реферат на тему «Использование ПИД-регулятора в образовательной робототехнике».

ГЛАВА IV СОРЕВНОВАНИЯ ПО ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ

§ 11. Классические соревнования роботов

Это соревнования для начинающих, их правила предельно просты и не меняются год от года. В общем регламенте соревнований имеются ограничения по использованию оборудования (например, только детали *Lego*, только 2 мотора и др.), по габаритам и массе, которые не должен превышать робот и др. К классическим соревнованиям относятся: кегельринг, сумо, траектория, лабиринт.

11.1. Кегельринг

Участникам необходимо подготовить автономного робота, способного выталкивать кегли за пределы ринга. Поле представляет собой белый круг с черной границей. Внутри круга равномерно устанавливается 8 кеглей[14].

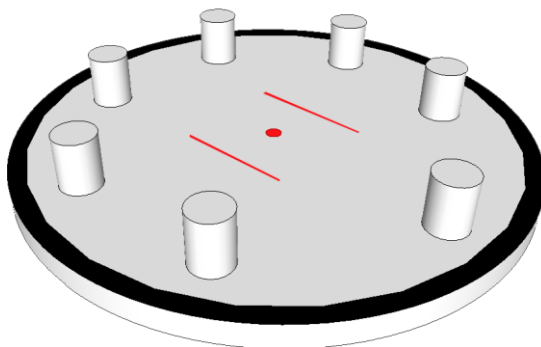


Рис. 89. Поле для игры «Кегельринг»

11.2. Сумо

Состязание проходит между двумя роботами. Цель состязания – вытолкнуть робота-противника за черную линию ринга.

Роботы должны проехать прямо и столкнуться друг с другом, после столкновения роботы могут маневрировать по рингу как угодно [15].

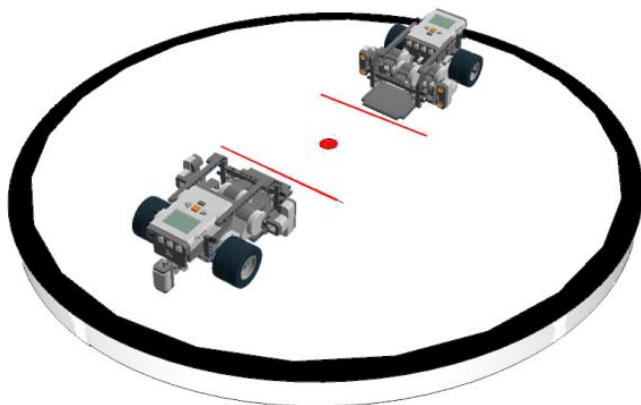


Рис. 90. Поле для игры «Сумо»

11.3. Траектория

В этом состязании участникам необходимо подготовить автономного робота, способного проехать от зоны старта до зоны финиша по траектории, составленной из типовых элементов, преодолевая препятствия [15].

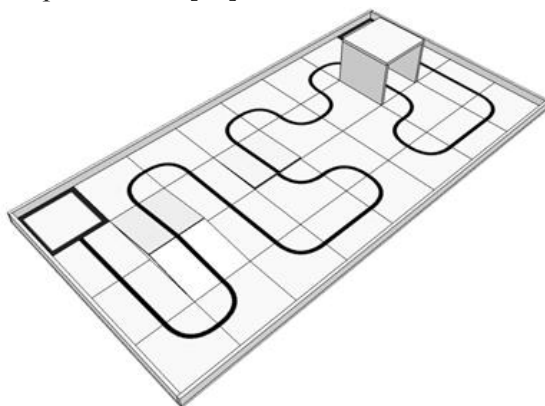


Рис. 91. Поле для игры «Траектория»

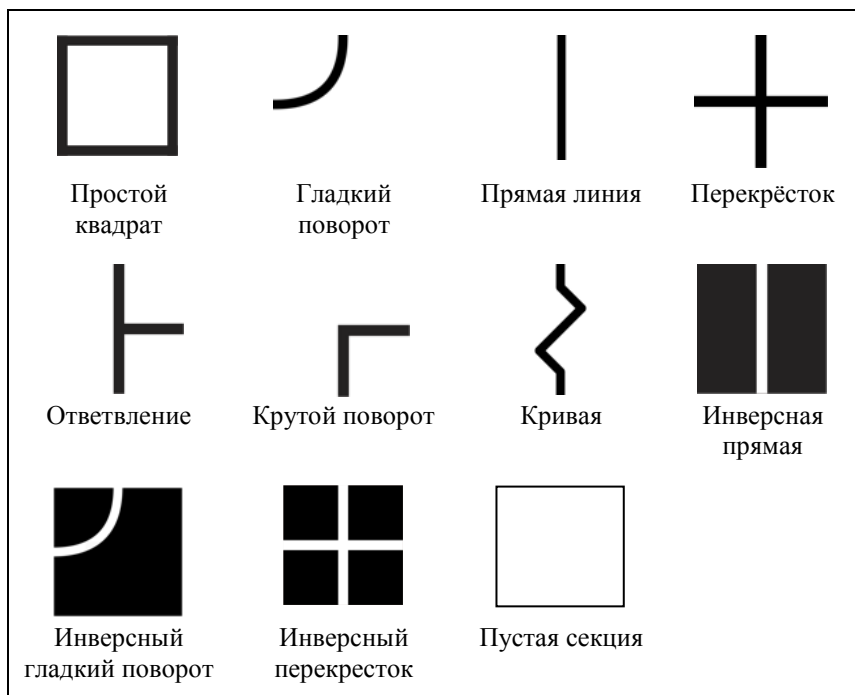


Рис. 92. Элементы, из которых может быть составлено поле для соревнования «Траектория»

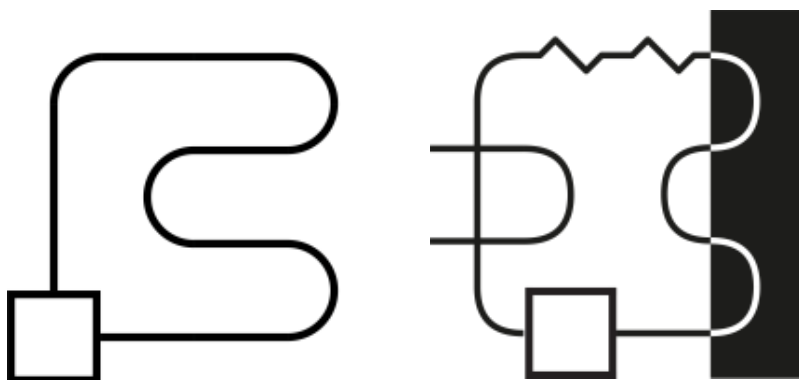


Рис. 93. Примеры траекторий

На траектории возможно использование дополнительных элементов: горок, трамплинов, препятствий, туннелей, банок и др. (рис. 94):

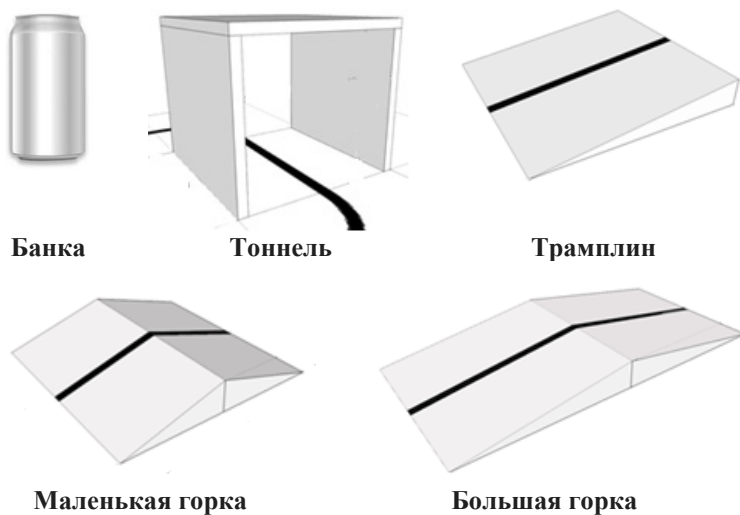


Рис. 94. Варианты дополнительных элементов к соревнованию «Траектория»

11.4. Лабиринт

В этом состязании участникам необходимо подготовить автономного робота, способного наиболее быстро проехать от зоны старта до зоны финиша по лабиринту, составленному из типовых элементов [15].

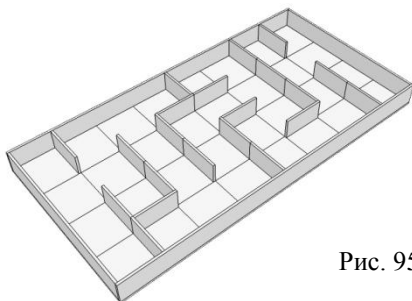


Рис. 95. Поле для игры «Лабиринт»

§ 12. Всемирная олимпиада по робототехнике

Всемирная олимпиада по робототехнике (*WorldRobotOlympiad (WRO)*) проводится в 40 странах по всему миру. Сначала *WRO* появилась в Азии, а далее распространилась на Ближнем Востоке, Европе, Африке, Южной Америке и Австралии.

Все турниры *WRO* ставят целью бросить участникам вызов, чтобы они задействовали свои творческие способности для построения и программирования автономных роботов. Оргкомитет *WRO* считает, что платформа *Lego Mindstorms* является для этого наиболее подходящей. При этом допускается использование любого из трех поколений конструкторов (*RCX*, *NXT*, *EV3*).

Всемирная олимпиада по робототехнике проводится в трех конкурсных категориях: основная, творческая категории и футбол роботов.

12.1. Основная и творческая категории

Основная категория была первой в соревнованиях *WRO*. В этой категории команды строят и программируют роботов, которые должны решить конкретную задачу на предметном столе. Например, это может быть робот, который должен в начале пути захватить шарик и пройти с ним обозначенную траекторию, преодолев все препятствия, и доставить шарик к финишу (рис. 96). Содержание испытаний разрабатывается организаторами олимпиады и должны быть распространены среди региональных организаторов. В рамках конкурса действует три возрастные подгруппы: младшая, средняя и старшая, для каждой из которых составляются задания разного уровня сложности.

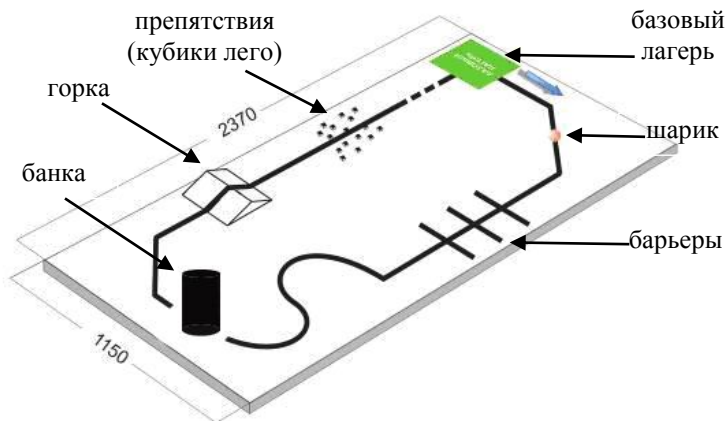


Рис. 96. Поле для соревнований «Траектория». Основная категория, младшая возрастная группа (2010 год)

Творческая категория также представлена в трех возрастных группах. Соревнование проводится в форме презентации участниками робототехнических проектов по заданной теме. Например, темой 2014 года является «Роботы и космос». Команды должны разработать и создать проекты, моделирующие использование роботов в космическом пространстве, на космических станциях, на поверхности и в недрах планет и др. Также допустимы проекты роботов, помогающих человеку на Земле в изучении космоса: телескопы, сборщики ракет и др.

12.2. Футбол роботов

Данная категория является сравнительно новой и впервые была представлена в 2010 году. Команды из двух роботов играют в футбол на игровом поле в течение 10 минут. В данной категории нет деления на возрастные группы, допускаются все участники в возрасте от 10 до 19 лет.

Для построения роботов нужно использовать платформу *Lego Mindstorms*, контроллеры *RCX*, *NXT* или *EV3*, сенсоры и

мячи от *HiTechnic*. Игры проводятся в соответствии с регламентом кубка *RoboCupJuniorAustralia (RCJA)*. В этом состязании участникам нужно создать автономных роботов, которые смогут выиграть у соперников футбольный матч [15; 33].

§ 13. Соревнования по правилам FIRST

Организация *FIRST* («За подъем и признание Науки и Технологий») была основана изобретателем Дином Кэменом для того, чтобы вдохновлять молодёжь на участие в научно-технической деятельности. В *FIRST* существует четыре программы:

- *FIRSTRoboticsCompetition (FRC)* – соревнования для учащихся в возрасте 14-18 лет;
- *FIRST Tech Challenge (FTC)* – соревнования для учащихся в возрасте 14-18 лет;
- *FIRST LEGO League (FLL)* – соревнования для учащихся в возрасте 9-14 лет;
- *Junior FIRST LEGO League (Jr. FLL)* – соревнования для учащихся младших классов возрастом 6–9 лет [31; 35; 39].

13.1. FLL

Программа *FIRST LEGO League (FLL)* предназначена для учащихся в возрасте от 9 до 14 лет. Тематика соревнований FLL ежегодно обновляется, тема задается в соответствии с какой-либо неразрешенной проблемой, стоящей перед человечеством. В качестве примера приведем три темы: *FoodFactor* (2011) – «Пищевой фактор»; *Senior Solutions* (2012) – «Решение проблем старшего поколения. Независимость. Занятость. Общение»; *Nature'sFurySM* (2013) – «Ярость природы». Соревнования по правилам *FLL* включают составление командой проекта, в котором предлагается инновационный подход к решению про-

блемы,

а также состязания роботов. Эти состязания заключаются в выполнении роботом миссий, т.е. определенных заданий, символически соответствующих решению заданной проблемы. За выполнение миссий робот зарабатывает очки. Порядок и количество выполнения миссий выбирают участники самостоятельно, закладывая эти параметры в программы робота.

Набор «Зелёный город», выпускаемый компанией *Lego* в дополнение к конструктору *LegoMindstormsNXT*, позволяет познакомиться с правилами этих соревнований [27]. Главная задача игры – активировать «Зеленый город». Для этого необходимо выполнить 4–6 миссий, в результате которых будут собраны 4–6 энергетических блока, с помощью которых будет выполнена последняя миссия – активизация «Зеленого города».

Миссии «Зеленого города»:

1. Запуск Ветрогенератора.
2. Размещение Солнечной батареи.
3. Безопасность Цветочной Леди.
4. Сортировка мусора.
5. Закрытие дамбы.
6. Развертывание новой дымовой трубы.
7. Активизация Зеленого города – финальная миссия

(обязательна к выполнению).

13.2. FIRST Tech Challenge (FTC)

Команды *FIRST Tech Challenge* каждый год получают задание во время сентябрьского анонса соревнований. В соревнованиях могут принимать участие школьники от 14 до 18 лет, команда включает от 3 до 10 человек. Для конструирования робота разрешены наборы *Lego*, *Tetrix*, *Matrix*, для программирования – *LabVIEW*, *RobotC*.

Матчи проходят на игровом поле, установленном, как показано на рисунке ниже. Два альянса: один альянс «красных», другой – альянс «синих». Альянсы, состоящие из двух команд каждый, соревнуются в каждом матче. Цель игры – получить более высокие баллы, чем альянс оппонентов. Игра состоит из двух периодов: автономный период и период с участием операторов. Правила и нормативные документы соревнований находятся на сайте www.usfirst.org.

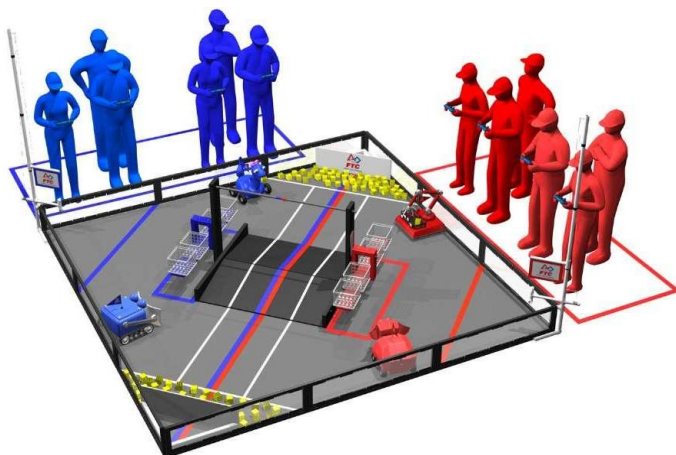


Рис. 97. Принцип соревнований *FTC*

Ценности, провозглашаемые программой *FIRSTTechChallenge(FTC)*:

- уважать друг друга в лучших традициях командной работы;
- всегда относиться с уважением и пониманием к другим;
- уважать дух дружественного соревнования;
- работать открыто и честно;
- демонстрировать благородный профессионализм;
- поощрять других к принятию этих ценностей.

Понятие «благородный профессионализм» понимается как часть философии осмысленной жизни: «принести пользу обществу и испытать удовлетворение от того, что ты поступил честно и отзывчиво». Благородный профессионал вносит свой вклад, принося радость другим и самому себе, он уважает других и проявляет это уважение в своих поступках. Благородные профессионалы обладают специальными знаниями, общество доверяет им ответственное их использование [31].

§ 14. Методика организации тренерской работы учителя по подготовке школьной команды к соревнованиям по робототехнике

Каждая команда, участвующая в соревнованиях, состоит из операторов и тренера. Операторами являются школьники, почти во всех соревнованиях организаторы накладывают возрастные ограничения и делят команды по возрастным категориям. Общепринятое возрастное ограничение для тренера – «не моложе 18 лет». Хорошими примерами тренеров являются родители, учителя, педагоги дополнительного образования, инженеры, студенты вузов и др.

Основная задача тренера: направлять процесс, которому следует команда, чтобы решить поставленные в игре задачи, при этом не давая готовых решений со своей стороны. Тренер для успешной работы должен обладать базовыми знаниями среды программирования и основ роботостроения. Необходимо быть в курсе новинок и идей, освещаемых в интернет-сообществе (на форумах, в блогах, социальных сетях и др.). Полезной бывает поддержка преподавателей технических дисциплин вузов и среднепрофессиональных учебных заведений, людей с инженерным образованием и навыками программирования, с тем, чтобы они могли поделиться опытом с командой.

Для организации подготовки к соревнованиям необходимо составить расписание занятий команды. Рекомендуется начинать с двух собраний в неделю в течение периода создания команды. Наиболее продуктивными обычно считаются занятия продолжительностью 2–3 часа. В ходе таких занятий учащиеся учатся решать проблемы, находя решения самостоятельно, и лучшей помощью тренера будет способствовать этому процессу.

В ходе работы над созданием робота для соревнований необходимо поощрять школьников к вежливому оспариванию мнения тренера и мнений других участников, если они не понимают или не согласны с ними. Для организации работы команды необходимо распределить ответственность и роль при создании робота между членами команды. Обычно участники команды имеют некоторое представление о том, чем они хотели бы заниматься – программирование, конструирование, разработка и др. В то же время могут быть учащиеся, которых вытеснили из той области, в которой они хотели бы участвовать или те, кто старается избегать определённых задач. Часто напоминайте участникам команды о важности сотрудничества, командной работе и распределении обязанностей. Например, участники команды работают вместе над механизмами, которые при небольшой модификации могут быть собраны в один узел. Необходимо убедиться в том, что ребята сообщают друг другу об изменениях.

При создании робота, как правило, используется метод мозгового штурма, для того чтобы построить робота, который смог бы выполнить задачу соревнований. Необходимо принять решение относительно главной идеи конструкции и постараться достигнуть согласия между членами команды относительно механического дизайна робота. Важно обеспечить согласованную работу всех членов команды, чтобы все механизмы могли быть собраны воедино. Аналогичную работу тренер должен осуществить при написании программы для робота.

Далее необходимо провести контроль качества созданного командой робота. Для этого:

- проводятся независимые испытания работы робота, с тем, чтобы установить потенциальные возможности для усовершенствования модели;
- проверяются функции, которые не работают надёжно, готовятся рекомендации по их улучшению.

Необходимо помнить, что разработка робота – это пошаговый, всегда развивающийся процесс. Важно регулярно делать какие-либо изменения для улучшения работы робота (его конструкции и программы).

Разработку конструкции робота для соревнований можно разделить на две основных части – привод (базовое шасси) и рабочий механизм (манипулятор).

Одной из основных систем робота является привод – система, которая перемещает робота по игровому полю. Существует множество конфигураций приводов, но все они состоят из: 1) одного и более моторов; 2) некоторого механизма передачи вращающего момента (например, колесо); 3) устройства управления. Наиболее распространённым является базовое шасси на основе двух ведущих колес и одного рулевого колеса [43]. Ведущие колеса закреплены на отдельных моторах, каждый из которых управляется независимо друг от друга. Существует много вариантов конструкции базового шасси, важно, чтобы эта конструкция позволяла выполнить поставленные задачи.

В дополнение к способности перемещаться по игровому полю робот в соревнованиях *WRO* и *First* должен манипулировать различными объектами. Манипулирование, возможно, самая сложная часть в соревнованиях, особенно для новых команд. Необходимо минимизировать вес и сложность манипуляторов. Тяжёлые большие конструкции затрудняют перемещение робота, приводят к чрезмерному расходу батареи, делают навигацию менее предсказуемой и повторяемой. Возможность отказа во

время соревнований тем выше, чем сложнее конструкция. Необходимо поощрять команду на неустанную работу в поиске простых решений, которые будут надёжно работать во время выступлений и постепенно будут только улучшаться.

Задания для самостоятельной внеаудиторной работы

I. Используя рекомендованные печатные издания и интернет-источники, создайте презентацию о подготовке команды к одному из видов соревнований (Сумо, Кегельринг, Траектория, Лабиринт, *WRO*, *FLL*, *FTC*).

II. Посетите какие-либо соревнования по образовательной робототехнике. Составьте отчёт о посещении соревнований по плану:

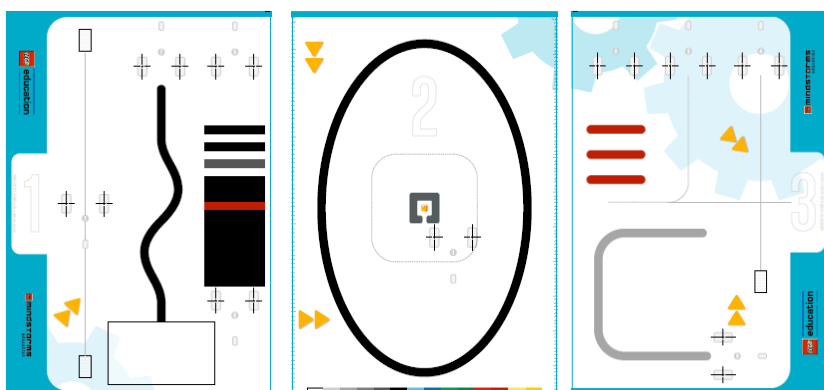
1. Название соревнований, возраст участников, разрешенное оборудование;
2. Основные положения регламента соревнований (кратко);
3. Особенности конструкций и стратегии поведения, созданных участниками роботов. Анализ их преимуществ и недостатков;
4. Особенности конструкций и стратегии поведения роботов, занявших призовые места.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Лабораторный практикум по курсу «Методика обучения образовательной робототехнике» позволяет студентам получить навыки конструирования и программирования робота на базе конструктора *LegoMindstormsNXT*. Данный практикум содержит:

- 5 лабораторных работ, в рамках которых студенты знакомятся с приемами графического программирования двухмоторной тележки с помощью стандартных датчиков, входящих в набор;
- 2 лабораторные работы, в которых рассматривается принцип организации классических соревнований робота (на примере соревнований Кегельринг) и соревнований по типу *FLL* (на примере набора «Зелёный город» (*GreenCity*));
- лабораторная работа, в рамках которой студенты имеют возможность познакомиться с различными готовыми моделями роботов (шагающие роботы, роботы-животные, манипуляторы и др.).

Для отладки программ робота предлагается использовать тренировочные поля из набора «Зеленый город» (см. рис. 98).



Тренировочное поле 1

Тренировочное поле 2

Тренировочное поле 3

Рис. 98. Тренировочные поля из набора «Зеленый город», используемые на лабораторном практикуме

Лабораторная работа 1. Состав конструктора и программное обеспечение LegoMindstormsNXT

Цель: познакомиться с конструктором и программным обеспечением *Lego Mindstorms NXT*.

Теоретическое введение

Для робота характерны следующие свойства, отличающие его от других технических устройств:

- очувствление – робот способен воспринимать различные факторы окружающей среды, для этого используются датчики – устройства сбора информации;
- планирование – робот способен использовать полученную от датчиков информацию для выбора своих действий, которые определяются согласно заложенной программе;
- действие – робот должен выполнять свой план, например, с помощью моторов, которые являются исполнительными механизмами робота.

«Мозгом» робота является микрокомпьютер *NXT*. Робот *NXT* имеет «органы чувств» – датчик касания, датчик звука, датчик освещенности, датчик расстояния. Исполнительные механизмы робота – моторы – позволяют ему перемещаться в пространстве, совершать различные действия и манипуляции. В каждый мотор встроены датчик вращения, он позволяет оценить длительность работы моторов. Подключив датчики и моторы *NXT* в установленном по умолчанию порядке (табл. 6), можно проверить, каким образом *NXT* воспринимает информацию от датчиков. Для этого можно воспользоваться меню *View* («Просмотр»).

Имеется возможность создавать для робота программы с использованием микрокомпьютера *NXT*. Эти программы могут быть составлены максимально из 5 блоков, имеют простую последовательную структуру выполнения, возможна организация бесконечного цикла. При этом датчики и моторы должны быть подключены к блоку *NXT*, как указано в таблице 6.

Практические задания

Задание 1. Ознакомьтесь с элементами конструктора *Lego Mindstorms NXT*.

Задание 2. Подключите электрические элементы конструктора к блоку *NXT* (табл. 6) и с помощью меню *View* получите текущие данные с каждого устройства.

Таблица 6

Порядок подключения моторов и датчиков к блоку *NXT*

Порт	Устройство
A, B, C	Моторы (сервоприводы), лампы, переходники на датчики старого образца
1	Датчик касания (<i>Touch Sensor</i>)
2	Датчик звука (<i>Sound Sensor</i>)
3	Датчик освещенности (<i>Light Sensor</i>)
4	Ультразвуковой датчик расстояния (<i>UltrasonicSensor</i>)

Задание 3. Соберите простейшую модель робота – двухмоторной тележки (рис. 60, § 3).

Задание 4. Создайте для робота программы (рис. 99) и проверьте их выполнение. Опишите словами, какие действия выполняет робот (рис. 100).

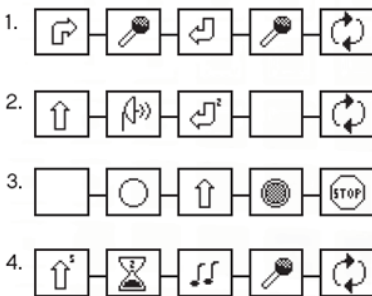


Рис. 99. Примеры программ для робота (к заданию 4)

1.		Робот поворачивает вправо,
		пока «не услышит» громкий звук;
		затем поворачивает влево,
		пока «не услышит» громкий звук.
		Программа выполняется бесконечно.

Рис. 100. Поясняющий рисунок к заданию 4

Задание 5. Пользуясь инструкциями «Робоцентра» раздел «Основная палитра» (рис. 101), составьте для робота программы:

- а) «Воспроизводить звук»;
- б) «Применить дисплей»;
- в) «Двигаться вперед»;
- г) «Двигаться назад»;
- д) «Ускорить»;
- е) «Разворот»;
- ж) «Поворот»;

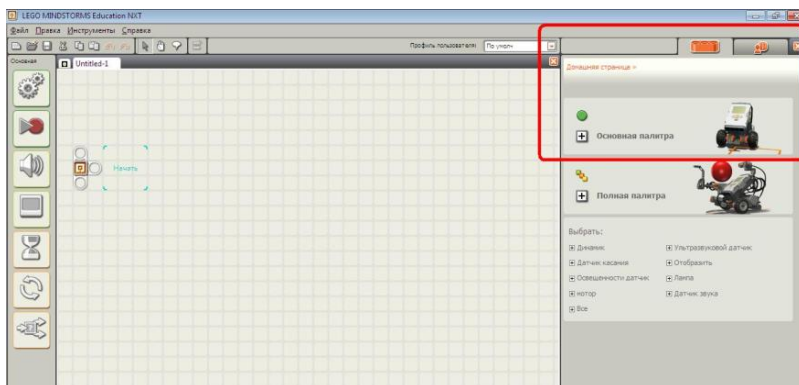


Рис. 101. Поясняющий рисунок к заданию 5.

Задание 6. Используя тренировочное поле 2 (рис. 98), протестируйте выполнение роботом программы «Ехать по квадрату». Отладьте программу таким образом, чтобы робот точно совершал повороты и вернулся на исходную позицию.

Контрольные вопросы

1. Каков порядок подключения моторов и датчиков к блоку *NXT*?
2. В каких единицах измерения можно снять показания датчиков, присоединенных к блоку *NXT*?

3. Какие алгоритмические конструкции используются при программировании робота с помощью блока *NXT*?

4. Какие элементы знаний по физике необходимы при знакомстве с датчиками *LegoMindstormsNXT*?

Лабораторная работа 2. Движение робота по заданной траектории

Цель: научиться управлять движением двухмоторной тележки.

Теоретическое введение

Двухмоторная тележка на базе конструктора *Lego Mindstorms NXT* является прототипом мобильного робота. Она имеет, как правило, три точки опоры: две из них – ведущие колеса, соединенные с моторами, и третья – волокуша (свободно вращающееся колесико, его также называют рулевым колесом). Центр масс двухмоторной тележки должен находиться не над волокушей, а ближе к ведущим колесам [43; 44]. Двухмоторную тележку часто называют также базовым шасси или основой для робота.

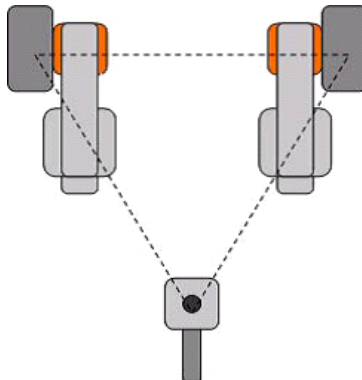
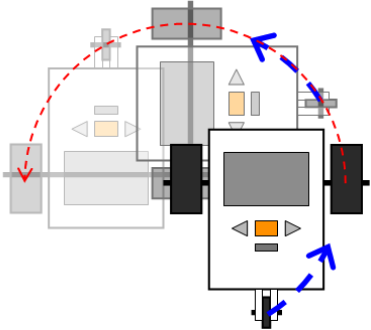
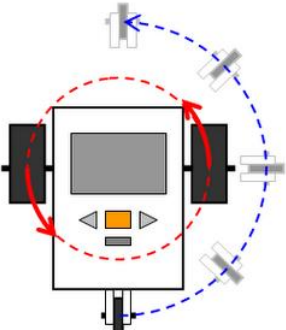
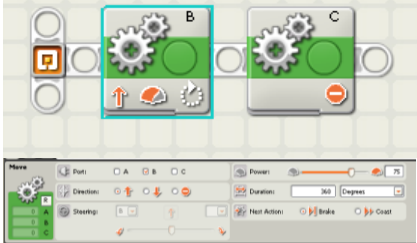
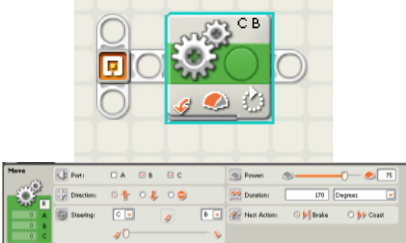


Рис. 102. Схема двухмоторной тележки с двумя ведущими колесами и одним рулевым колесом

Для решения задачи движения робота используются сервомоторы, они подключаются, как правило, к портам В,С. Для программирования движения робота используется блок «Движение». Двухмоторная тележка может осуществлять повороты двумя способами: вокруг одного колеса и «на месте», т.е. вокруг себя (табл. 7).

Таблица 7

Выполнение поворотов двухмоторным роботом

Поворот одним колесом	Поворот двумя колесами
	
	

Практические задания

Задание 1. Составить программу «Движение вперед» таким образом, чтобы робот перемещался вперед по прямой линии на один оборот колеса. Запустить программу и проверить её выполнение.

Задание 2. Модифицировать программу «Движение вперед» таким образом, чтобы робот перед началом движения издавал звук, например, «Hello», после перемещения на один оборот колеса выводил на экран *NXT* изображение, например, смайл.

Задание 3. Модифицировать программу «Движение вперед» таким образом, чтобы робот двигался назад по прямой линии на 420° (поворот оси колеса).

Задание 4. Составить программу для робота, выполняющего поворот на 90° : а) поворот выполняется влево одним колесом; б) поворот выполняется влево двумя колесами.

Задание 5. Составить программу для робота, который будет двигаться по квадрату. Отладить программу, используя тренировочное поле 2.

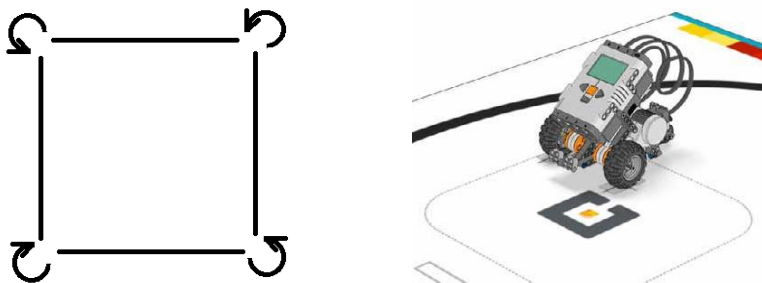


Рис. 103. Идея алгоритма движения робота по квадрату

Задание 6. Составить программу для робота, который описывает «восьмерку» (рис. 105). Для обеспечения контроля за правильностью движения робота по заданной траектории прикрепите к нему фломастер и проводите испытание программ на листе ватмана.

Задание 7. Модифицировать предыдущую программу для робота таким образом, чтобы траектория его движения соответствовала рис. 106.

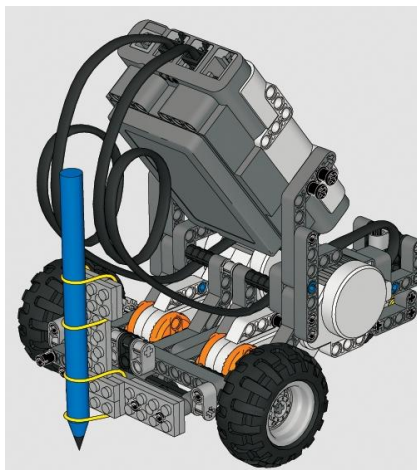


Рис. 104. Пример закрепления фломастера на двухмоторной тележке

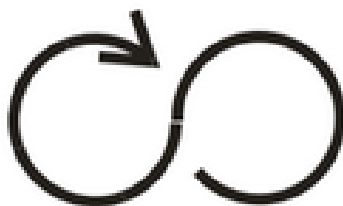


Рис. 105. Траектория движения «Восьмерка»



Рис. 106. Траектория движения «Змейка»

Задание 8. Модифицировать программу «Движение по квадрату» таким образом, чтобы робот двигался: а) вдоль сторон треугольника; б) вдоль сторон шестиугольника.

Задание 9. Составить программу для робота, движущегося по «восьмерке» по одному из вариантов (рис. 107).

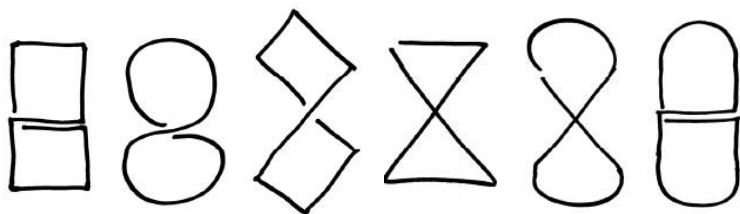


Рис. 107. Варианты траектории движения робота «Восьмерка» [52; 53]

Контрольные вопросы

1. Какими единицами измерения может задаваться длительность работы мотора в *NXT-G*?
2. Какие способы осуществления поворота двухмоторной тележкой Вы можете назвать?
3. Можно ли однозначно утверждать, что представленный фрагмент программы для двухмоторной тележки предназначен для движения робота по квадрату? Ответ обоснуйте.

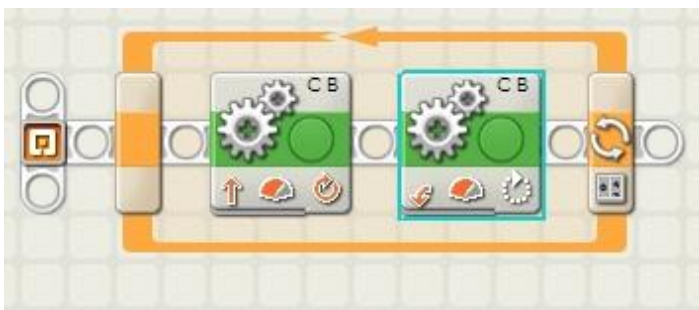


Рис.108. Программа движения робота (к вопросу 3)

4. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?
5. Какие элементы знаний по физике и математике необходимы при решении задач движения робота?

Лабораторная работа 3. Управление поведением робота с помощью датчиков касания, звука, расстояния

Цель: научиться использовать датчики касания, звука, расстояния в работе робота.

Теоретическое введение

Робот, собранный из конструктора *LegoMindstormsNXT*, способен выполнять следующие действия:

- движение всего устройства или перемещение его отдельных частей;
- запоминание движения (запоминание состояния моторов с помощью блока запись / воспроизведение);
- воспроизведение звуков;
- вывод изображения на экран NXT.

Робот способен изменять свое поведение в зависимости от внутренних часов (таймер) внешних событий. Внешние события робот воспринимает через сенсоры (датчики), от оператора и других роботов (через *Bluetooth*).

При программировании робота говорят о смене состояния робота в результате наступления какого-либо события. То есть робот находится в режиме ожидания события, при его наступлении изменяет свое состояние. Например: робот находится в состоянии движения вперед, ему приходит сигнал от сенсора о наличии впереди препятствия, и робот останавливается (рис. 109).

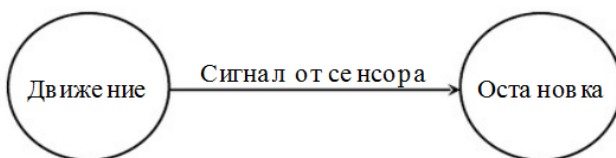


Рис. 109. Смена состояний робота

Программирование ожидания событий (команда «Жди...») осуществляется с помощью блока «Ожидание» (*Wait*), который находится в Основной палитре команд (рис. 110).



Рис. 110. Блок «Ожидание» и его модификации в «Основной палитре команд»

Источниками событий могут быть:

- сенсоры (датчики);
- внутренние часы и таймер;
- кнопки NXT (нажатие оператором на эти кнопки);
- сообщения, переданные по *Bluetooth* (от других роботов или от оператора через *Bluetooth*-устройства).

Практические задания

Задание 1. Подготовьте для выполнения заданий двухмоторную тележку, присоедините к ней датчики касания, звука, расстояния.

Задание 2. Составьте программу таким образом, чтобы робот ехал вперед, пока не услышит громкий звук (хлопок в ладоши). Затем робот должен остановиться.

Задание 3. Измените созданную программу таким образом, чтобы робот начинал движение только после звукового сигнала.

Задание 4. Составьте программу для робота, который изменяет скорость своего движения в зависимости от громкости окружающей среды (чем громче звук, тем он быстрее едет, чем тише звук, тем он едет медленнее).

Задание 5. Составьте программу для робота, который движется вперед до обнаружения препятствия (с помощью датчика касания) и останавливается.

Задание 6. Составьте программу для робота, который движется вперед до обнаружения предмета (с помощью датчика касания), после этого робот издает звук (например, «Hello»), поворачивается на некоторый угол и снова едет вперед.

Задание 7. Составьте программу для робота, который выполняет роль охранной сигнализации. Например, когда он обнаруживает препятствие впереди себя, он издает звук сирены.

Задание 8. Составьте программу для робота-прилипалы, который держится на определенном расстоянии перед препятст-

вием. Если это расстояние увеличивается, он подъезжает вперед, если уменьшается – отъезжает назад.

Контрольные вопросы

1. В чем заключается ожидание роботом события?
2. На какие события способен реагировать робот на основе *LegoMindstormsNXT*?
3. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?
4. Какие элементы знаний по физике необходимы при использовании датчиков касания, звука, расстояния в работе робота?

Лабораторная работа 4. Управление движением робота с датчика освещенности и датчиком цвета

Цель: научиться использовать датчики освещенности и цвета для движения робота по разметке.

Теоретическое введение

I. Движение робота с датчиком освещённости

Показания датчика освещенности зависят от интенсивности освещения в месте измерения. Для того чтобы всегда быть уверенным в достоверности показаний датчика освещенности, необходимо выполнить калибровку своего датчика применительно к рабочим условиям окружающей среды.

Это можно осуществить тремя способами:

- используя меню *View* в *NXT*;
- используя поле обратной связи в *NXT-G*;
- используя блок «Калибровка» в *NXT-G*.

Калибровка с помощью меню *View* в *NXT* осуществляется в следующей последовательности:

- 1) выбирается датчик освещенности *ReflectedLight*;
- 2) выбирается порт, к которому подключен датчик;

3) робот помещается на поле, на котором ему предстоит выполнить задание;

4) снимаются показания датчика освещенности в требуемых условиях.

В поле обратной связи текущие показания датчика отображаются, если робот подключен к компьютеру и активным является блок датчика освещенности.

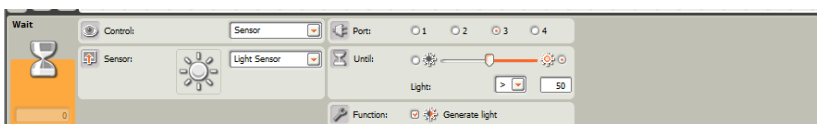


Рис. 111. Поле обратной связи в панели настроек блока «Ожидание», настроенного на показания датчика освещенности

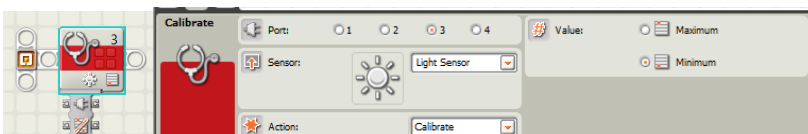


Рис. 112. Значок и панель настроек блока «Калибровка»

Параметр *Action* позволяет выбрать два действия для блока: калибровать (*Calibrate*) и сбросить настройки (*Delete*). Параметр *Value* определяет какое значение необходимо откалибровать – минимальное (*Minimum*) или максимальное (*Maximum*).

Для того, чтобы робот мог обнаружить край черной линии, необходимо «границу» между областями различного цвета. Сигналом о «нахождении границы» служит пороговое значение срабатывания датчика освещённости. Оно находится как среднее арифметическое между показаниями датчика на области одного цвета и показаниями на области другого цвета. Это значение заносится в программу.

II. Движение робота с датчиком цвета

Датчик цвета может работать в двух режимах:

- 1) как датчик цвета;
- 2) как датчик освещенности.



Рис. 113. Выбор режима работы датчика цвета NXT

Режим работы датчика устанавливается в панели настроек блока, соответствующего датчику цвета. Панель настроек блока Ожидания, настроенного на датчик цвета, соответствует работе датчика в режиме датчика цвета. С помощью регуляторов устанавливается диапазон значений, на которые будет реагировать датчик.



Рис. 114. Панель настроек датчика цвета. Режим «датчик цвета»

Панель настроек блока Ожидания, настроенного на датчик цвета, соответствует работе датчика в режиме датчика освещенности. С помощью регулятора устанавливается пороговое значение, на которое будет реагировать датчик.



Рис. 115. Панель настроек датчика цвета.
Режим «датчик освещенности»

Практические задания

Задание 1. Составить программу для робота, который должен найти черную линию, и отладить её, используя тренировочное поле 1 (рис. 116).



Рис. 116. Отладка программы поиска черной линии на тренировочном поле 1

Задание 2. Модифицировать программу так, чтобы робот находил третью по счету черную линию, и отладить её, используя тренировочное поле 1.

Задание 3. Составить программу для робота, который обнаруживает красную линию, и отладить её, используя тренировочное поле 1.

Задание 4. Создайте программу «Движение вдоль черной линии» и отладьте её, используя тренировочное поле 1.

Задание 5. Модифицируйте программу «Движение вдоль черной линии» таким образом, чтобы робот двигался по ней:

- а) пока он не совершит 1,5 оборота колеса;
- б) пока он «не услышит» громкий звук.

Задание 6. Составить программу для робота, который обнаруживает вторую по счету красную линию. Отладить программу, используя тренировочное поле 3.

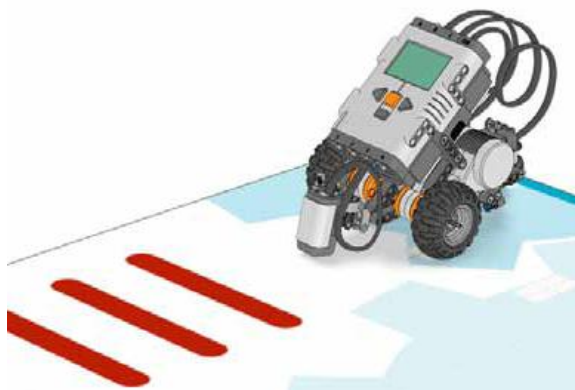


Рис. 117. Отладка программы поиска черной линии на тренировочном поле 3

Задание 7. Составить программу для робота, который движется вдоль серой линии. Отладить программу, используя тренировочное поле № 3.

Задание 8. Протестировать работу датчика цвета на различных поверхностях с помощью меню *View* в *NXT*.

Задание 9. Составить программу для робота, оснащенного датчиком цвета. При нажатии на датчик касания на датчике цвета должен загораться синий светодиод.

Задание 10. Запрограммировать датчик цвета так, чтобы по очереди мигали все три светодиода.

Задание 11. Составить программу для робота, который издает звуковой сигнал при нахождении зеленого кубика *Lego*.

Задание 12. Составить программу для робота, который движется вдоль черной линии с помощью датчика цвета.

Контрольные вопросы

1. Какими способами осуществляется калибровка датчика освещенности?
2. Почему при использовании датчика освещенности, роботу «сложно увидеть» красную линию?
3. В каких режимах может работать датчик цвета из набора *LegoMindstormsNXT*?
4. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?
5. Какие элементы знаний по физике необходимы при решении задач движения робота по разметке?

Лабораторная работа 5. Создание программ для робота с использованием переменных

Цель: научиться использовать возможности блоков «переменная», «Математика», «Логика» в программной среде *LegoMindstormsNXTSoftware*.

Теоретическое введение

Переменную придумали математики в XVII веке для того, чтобы «забронировать» в формуле место, на которое в нужный момент можно подставить конкретное значение. Поэтому с точки зрения математики переменная – это величина, характеризующаяся множеством значений, которое она может принимать.

В любом языке программирования используются переменные. Это делает программы более сложными, но в то же время позволяет повысить их эффективность, хранить промежуточные значения во время исполнения программы. Некоторые задачи в программировании вообще невозможно решить без использования переменных. В информатике под переменной понимают область памяти, к которой мы можем обратиться с помощью имени переменной. Переменная нужна для того, чтобы записывать и считывать из нее значения. При этом

каждая переменная имеет имя и свой тип (числовой, символьный, текстовый и др.), который задаётся в каждой конкретной программе. Объявленной переменной можно присваивать значения соответствующего типа. Таким образом:

- переменная – это область памяти;
- каждая переменная имеет тип;
- переменной можно присваивать значения этого типа;
- переменные нужны для того, чтобы записывать и считывать из них значения.

В языке *NXT-G* можно задать три типа переменных: числовой, логический, текстовый. Имя переменной – это последовательность латинских букв и цифр, причем первая – буква. Желательная длина имени – не больше 3-4 символов.

Для объявления переменной используется меню *Edit / Правка*, команда *DefineVariables / Создать переменную* (рис.118). В окне редактора переменных задаем имя переменной / *Name* и выбираем её тип (числовой / *Number*, логический / *Logic*, текстовый / *Text*). Нажимаем *Create / Определить переменную*.

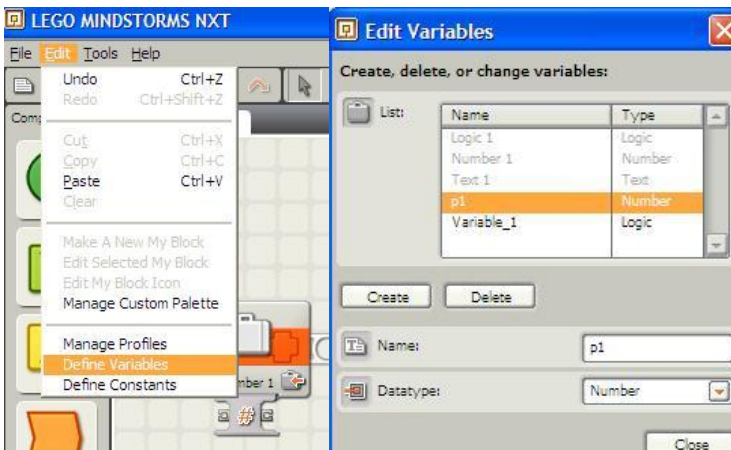


Рис. 118. Этапы создания переменной в *Lego Mindstorms NXT Software*

Для работы с переменными служит блок «Переменная» в «Полной палитре команд» *NXT-G*.

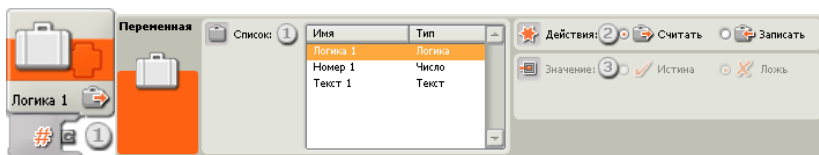


Рис. 119. Блок «Переменная». Панель настроек блока «Переменная»

Использование переменной в программе *LegoMindstormsNXTSoftware* предполагает всегда использование тройки команд:

- *записать* значение в переменную (автоматически в начале программы, с блока «Датчик» и др.);
- *считать* значение переменной (например, с блока «Датчик»);
- *изменить* значение (с помощью блоков «Математика», «Логика», «Сравнение» и др.).

Для переменных числового типа предназначен блок «Математика». Он выполняет такие простые арифметические действия, как сложение, вычитание, умножение и деление.

Блок «Сравнение» может определить, является ли число больше (>) или меньше (<), чем другое число, или равно (=) другому числу.

Блок «Логика» позволяет работать с переменными логического типа, блок принимает только два возможных значения данных – «истина» и «ложь» – для входа и для выхода. Часто эти значения записываются в виде чисел «1» и «0», где любое истинное утверждение записывается как «1», а каждое ложное утверждение записывается как «0». Четыре логических операции (*And* – И, *Or* – ИЛИ, *Xor* – Исключающее ИЛИ и *Not* – НЕ), которые может выполнять этот блок, позволяют вам произвести

ряд сравнений на основе таблиц истинности математической логики.



Рис. 120. Блоки «Математика», «Сравнение», «Логика» в *Lego Mindstorms NXT Software*

Практические задания

Задание 1. Используя инструкции «Робоцентра», создайте программу для робота, который подсчитывает количество щелчков (нажатий на датчик касания) в течение 20 с.

Задание 2. Используя блок «Математика», создайте программу для робота, движущегося по спирали. При этом количество витков спирали, радиус каждого витка задаются математически.

Задание 3. Составьте программу для робота, который вращается вокруг одного колеса, а при приближении некоторого объекта, останавливается и издает звук (для составления программы используйте переменную логического типа).

Задание 4. Составьте программу для робота, который движется вдоль черной линии. В конце пути помещен мяч на подставке, как только робот видит мяч, он должен ударить по нему клюшкой.

Задание 5. Робот должен посчитать количество предметов, расставленных по прямой. По окончании движения робот

должен каким-то образом передать человеку информацию о количестве предметов.

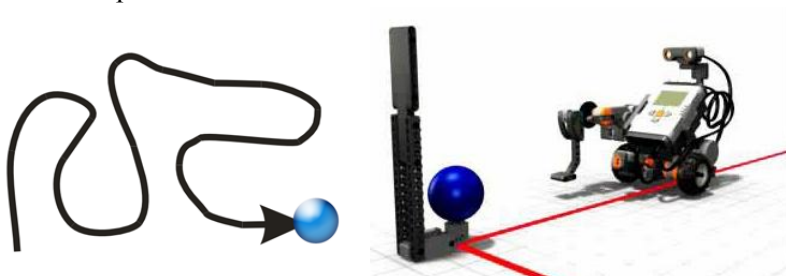


Рис. 121. Поясняющий рисунок к заданию 4

Лабораторная работа 6. Программирование NXT в программной среде Robolab

Цель: получить навыки работы со средой программирования Robolab.

Теоретическое введение

Основные элементы интерфейса программной среды *Robolab* представлены на рис. 122.

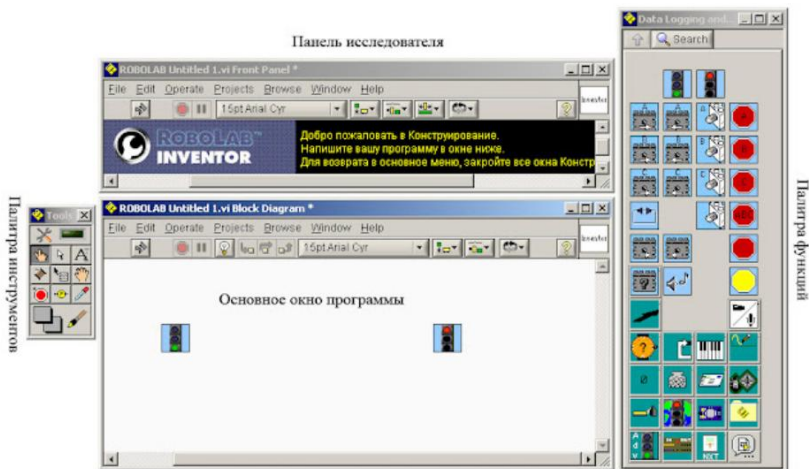


Рис. 122. Окно программной среды *Robolab*

Команды в программе задаются в виде пиктограмм, которые находятся в Палитре команд (функций). Там представлены команды действия, ожидания, управляющие структуры, модификаторы.

Соединение пиктограмм между собой имеет ряд свойств:

- щелчок в любом месте окна, кроме пиктограммы, фиксирует точку для поворота провода;
- двойным щелчком в области окна программы заканчивается прокладка начатого провода;
- при нажатии клавиши пробел провод поворачивается.

Программная среда *Robolab* позволяет создавать линейные алгоритмы с ветвлением и циклом. Два светофора обозначают начало и конец программы.

Цикл применяется для многократного повторения команд. Он может иметь параметр, который определяет, сколько раз выполняются команды между началом и концом цикла. Команды Прыжок (стрелка вверх) и Метка (стрелка вниз) служат для изменения порядка выполнения команд. Они позволяют делать в программе петли (возвращаться назад) или пропускать команды (прыгать вперед).

Ветвление применяется для проверки различных условий. Когда программа достигает развилки, она проверяет значение, указанное на развилке. Если значение больше указанного, программа дальше следует вдоль провода, присоединенного к знаку «больше» (>), если меньше или равно, то по проводу знака «меньше или равно» (<=).

Практические задания

Задание 1. Запустите программу *Robolab*. При запуске программы пользователю предлагается три режима «Администратор», «Программист», «Исследователь». Выберите режим «Администратор» и измените прошивку *NXT*. Вернитесь к вы-

бору режимов работы в *Robolab*, выберите режим «Программист», затем в разделе Конструирование – уровень *Inverter 4*.

Задание 2. Составьте программу для движения робота по квадрату:

- повторять 4 раза:
 - моторы *B, C* вперед в течение 1 секунды;
 - мотор *C* назад (*B* вперед) в течение $N/100$ секунды;
- стоп моторы.



Рис. 123. Программа к заданию 2

Задание 3. Составьте программу для движения робота в круге, ограниченном черной линией.

Повторяй бесконечно:

- моторы *B, C* вперед;
- жди темнее на датчике освещенности (среднее арифметическое между показаниями датчика освещенности на темной и на светлой областях поля);
- изменить направление моторов *B, C*;
- жди $N/100$ секунды;
- мотор *C* вперед (*B* назад);
- жди $N/100$ секунды.

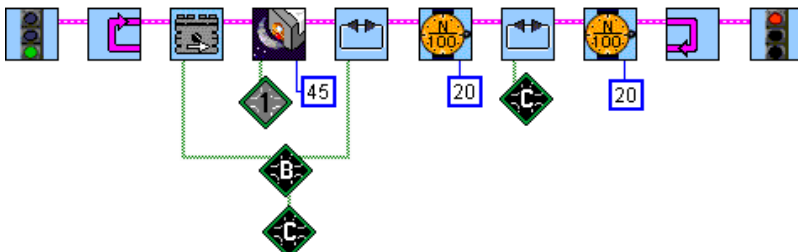


Рис. 124. Программа к заданию 3

Задание 4. Составьте программу для робота, который движется по черной линии (перед запуском программы установить робота слева от черной линии). В качестве граничного значения для датчика освещенности используется среднее значение между его показаниями на светлом участке поля и на темном.

Повторяй бесконечно:

- плавный поворот направо до черного;
- плавный поворот налево до белого.

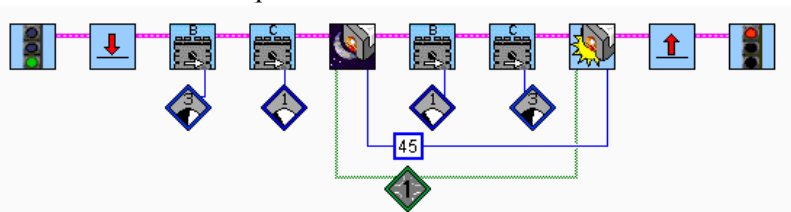


Рис. 125. Программа к заданию 4

Задание 5. Составьте программу для робота, который путешествует по комнате (датчик расстояния смотрит вперед).

Повторяй бесконечно:

- моторы *B, C* вперед;
- жди ближе 30 на датчике расстояния;
- изменить направление моторов *B, C*;
- жди $N/100$ секунды;
- мотор *C* вперед (*B* назад);
- жди $N/100$ секунды.

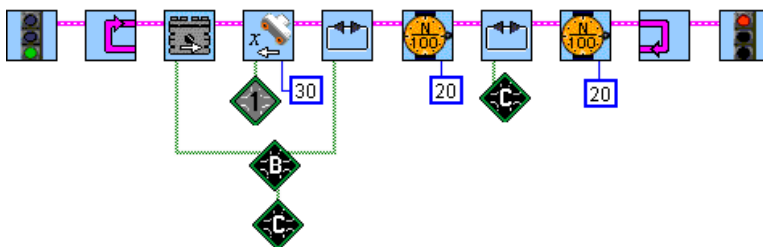


Рис. 126. Программа к заданию 5

Задание 6. Составьте программу для робота, который движется по черной линии. В качестве граничного значения для датчика освещенности используется среднее значение между его показаниями на светлом участке поля и на темном.

Повторяй бесконечно:

- если показания датчика освещенности выше среднего значения, поворачивай колесом B ;
- если показания датчика освещенности ниже среднего значения, поворачивай колесом C ;
- жди $N/1000$ секунды.

Перед запуском программы установить робота слева от черной линии.

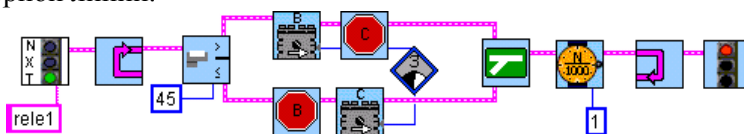


Рис. 127. Программа к заданию 6

Задание 7. Составьте программу для робота прилипалы: если расстояние до ближайшего объекта находится в заданном интервале, робот остается на месте, если это расстояние больше, робот едет вперед, иначе – едет назад.

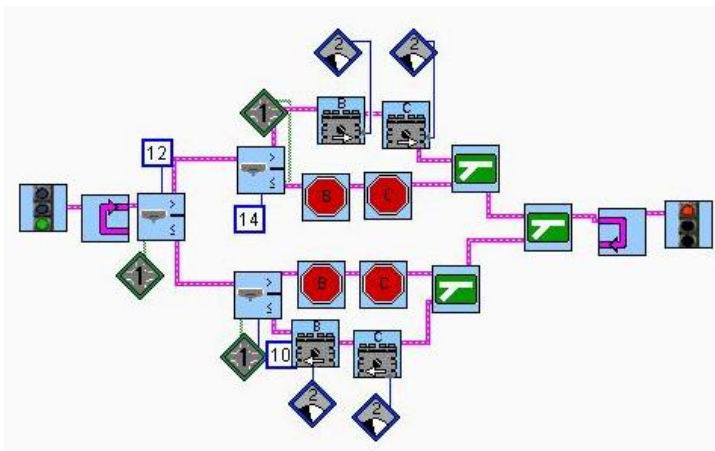


Рис. 128. Программа к заданию 7

Контрольные вопросы

1. Какие преимущества и недостатки имеет среда программирования *Robolab* по сравнению с *Lego-MindstormsNXTSoftware*?
2. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?
3. Какие элементы знаний по физике и математике использовались Вами при выполнении заданий лабораторной работы?

Лабораторная работа 7. Организация проектной деятельности школьников с набором *GreenCity* (Зелёный город)

Цель: познакомиться с правилами проведения соревнований по типу *FLL*.

Описание миссий набора «Зеленый город»

Запуск Ветрогенератора

Для активизации необходимо нажать или ударить по красному рычагу и будет выпущен Энергетический блок. Выпущенный Энергетический блок роботу нужно забрать и принести на Базу (рис. 130).



Рис. 130

Размещение Солнечной батареи

Робот должен поместить Солнечную батарею на крышу дома, чтобы высвободить Энергетический блок. Выпущенный Энергетический блок роботу нужно забрать и принести на Базу (рис. 131).



Рис. 131

Безопасность Цветочной Леди

Робот не должен двигать или опрокидывать Цветочную Леди во время выполнения миссий (рис. 132).



Рис. 132

Сортировка мусора

Робот должен собрать и отсортировать городской мусор, помещая Желтый Мусор в Желтый Контейнер и Черный Мусор в Черный Контейнер. Когда выпущены Энергетические блоки, они должны быть собраны и принесены роботом на Базу. Не разрешается приносить на Базу Мусор и Контейнеры (рис. 133).

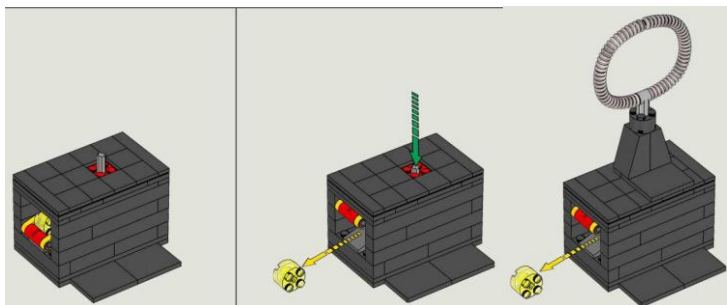


Рис. 133

Закрытие дамбы

Робот должен найти и использовать Блок Дамбы, чтобы закрыть Отверстие в Дамбе и выпустить Энергетический блок. Выпущенный Энергетический блок роботу нужно забрать и принести на Базу (рис. 134).



Рис. 134

Развертывание новой дымовой трубы

Робот должен поднять Белую Дымовую трубу и положить Черную Дымовую трубу. Когда выпущен Энергетический блок, он должен быть принесен роботом на Базу (рис. 135).

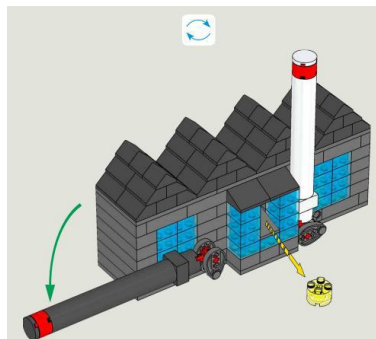


Рис. 135

Активизация Зеленого города

Робот должен положить в лоток по крайней мере четыре Энергетических Блока и задвинуть лоток, тем самым активизировать Зеленый город (рис. 136).

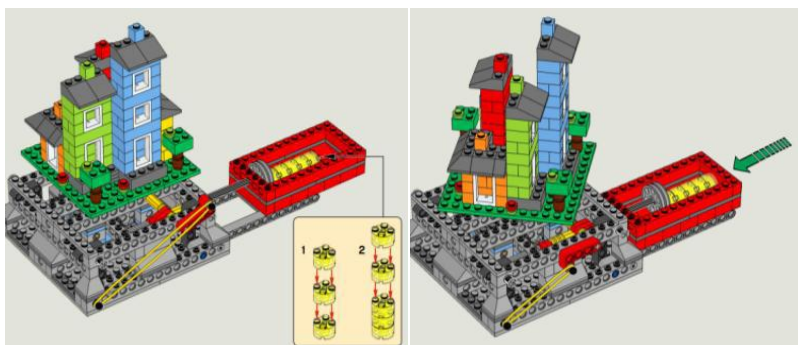


Рис . 136

Практические задания

Задание 1. Используя интернет-источники, ознакомьтесь с основными правилами *FLL*.

Задание 2. Пользуясь методическими инструкциями, представленными на диске «Зеленый город», выполните следующее:

- соберите базовую модель робота и насадки к нему;
- загрузите в блок *NXT* программы для выполнения миссий (или разработайте данные программы самостоятельно);
- ознакомьтесь с правилами подсчета баллов за выполнение миссий.

Задание 3. Проведите в своей группе соревнования «Зеленый город».

Контрольные вопросы

1. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?

2. Какие элементы знаний по физике, математике необходимы при проектировании робота для выполнения миссий Зеленого города?

Лабораторная работа 8. Кегельринг

Цель: научиться создавать робота для соревнований, познакомиться с методикой организации и проведения соревнований «Кегельринг».

Теоретическое введение

Задачей робота в соревнованиях «Кегельринг» является освобождение ринга от кеглей. Тот робот, который справляется с этой задачей за наименьшее время, объявляется победителем.

Краткое содержание правил соревнований «Кегельринг»:

- ринг имеет круглую форму, основной цвет белый, края ринга ограничены черной линией;
- перед началом соревнования на ринге устанавливается 8 кеглей;

- робот помещается в центр ринга и включается, после этого в его работу нельзя вмешиваться. Максимальные размеры робота 20 см×20 см×20 см (возможны иные ограничения);

- робот должен выталкивать кегли только своим корпусом;

- кегля считается вытолкнутой, если никакая ее часть не находится внутри круга, ограниченного линией.

Советы по конструированию:

- использовать колеса большего диаметра, т.к. маленькое колесо за поворот двигателя на 180° , проедет гораздо меньшее расстояние, чем проедет большое колесо при повороте двигателя на эти же 180° ;

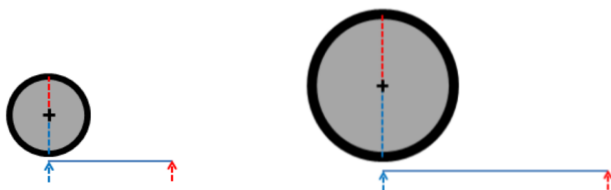


Рис. 137

- использовать датчик для определения местонахождения кеглей;

- использовать датчик для контроля, который определяет, выехал ли робот за пределы ринга.



Рис. 138. Пример робота для соревнований «Кегельринг» [43; 44]

Стратегии поведения робота

Стратегия 1. Пусть робот (двухмоторная тележка) оснащен датчиком расстояния, чтобы отслеживать кегли, и датчиком освещенности, чтобы не выехать за пределы поля. Робот ставится в центр поля, он поворачивается вокруг себя, пока не заметит кеглю. Робот едет вперед, выталкивая кеглю за край поля, и возвращается обратно в центр (рис. 139).

Стратегия 2. Пусть робот (двухмоторная тележка) оснащен датчиком расстояния, чтобы отслеживать кегли, и двумя датчиком освещенности, чтобы отслеживать край поля и не выехать за его пределы. Робот ставится в центр поля, он поворачивается вокруг себя, пока не заметит кеглю. Робот едет вперед, выталкивая кеглю за край поля, и едет в обратную сторону до противоположного края поля, выталкивая кеглю установленную напротив. Далее робот возвращается в центр и поворачивается вокруг себя, отслеживая следующую кеглю [33] (рис. 140).

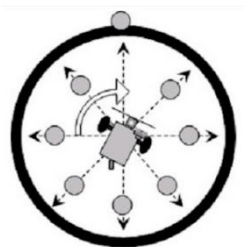


Рис. 139. Пояснение к стратегии 1

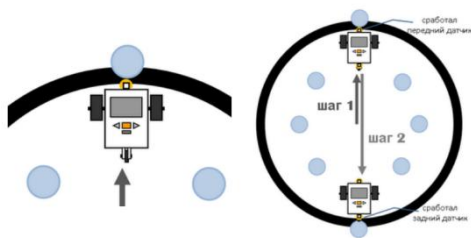


Рис. 140. Пояснение к стратегии 2

Возможны и другие варианты стратегий поведения робота [33] (рис. 141).

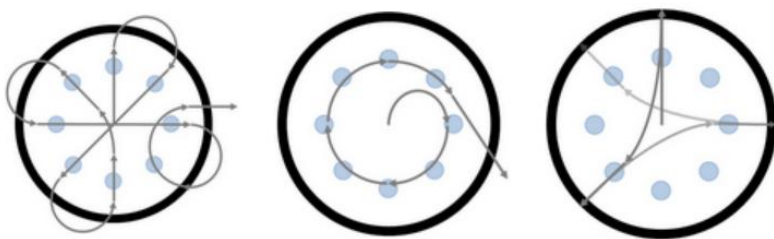


Рис. 141. Варианты стратегий поведения робота [33]

Датчик оборотов двигателя очень часто используется при программировании робота для игры «Кегельринг». Как правило, требуется указать в градусах, на какой угол должны повернуться моторы, чтобы робот вернулся в центр поля. При многократном использовании датчика все обороты суммируются. Поэтому необходимо обнулять счётчик, чтобы обеспечить правильные действия робота на втором и последующих этапах выполнения программы.

Поясним на примере: робот должен проехать вперед на 720° , а затем вернуться обратно в исходное положение.

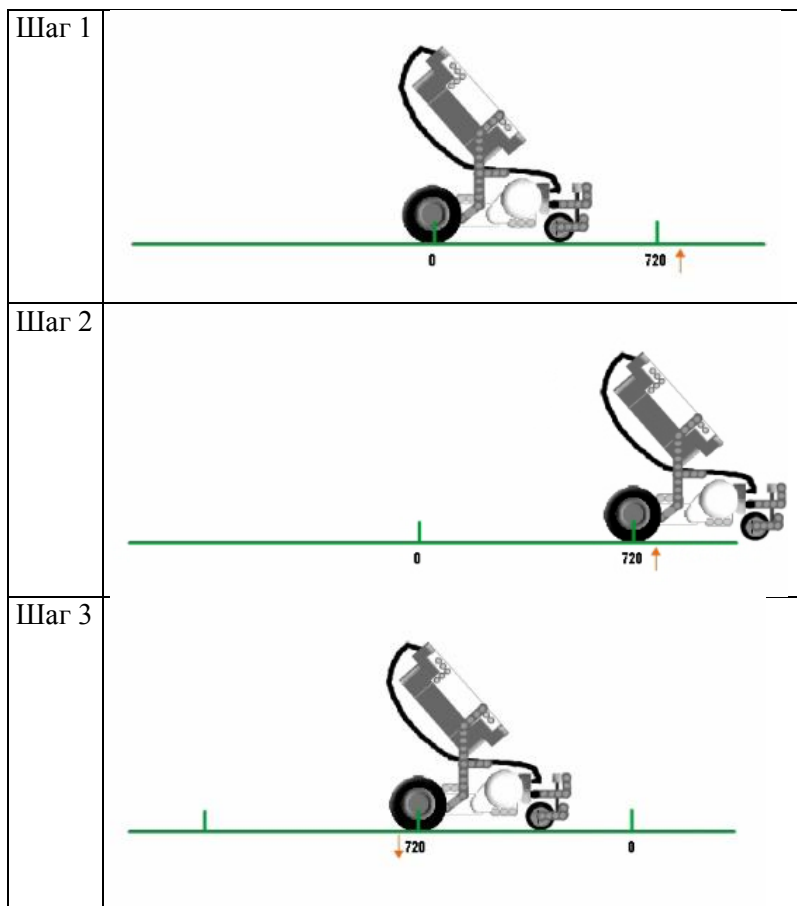


Рис. 136. Пояснение к программированию датчика оборотов

Реализация первого этапа не вызывает затруднений. На втором шаге, если запрограммировать робота так, чтобы он двигался назад на то же число оборотов, робот будет ожидать, пока датчик оборотов не зафиксирует поворот больше, чем на 720° , в обратном направлении, что эквивалентно повороту вперёд менее, чем на -720° . Фактически робот совершит движение назад на 1440° . Если же обнулить показания датчика

оборотов после завершения первого этапа, робот будет начинать отсчёт с нуля и продвинется назад на расстояние, соответствующее повороту на -720° . Это и будет заданным поведением робота [28].

Практические задания

Спроектируйте собственного робота для игры в «Кегельринг» и проведите такую игру в вашей группе.

Контрольные вопросы

1. С помощью каких приемов можно увеличить мощность робота (скорость его движения)?
2. Почему, когда робот начинает свою «работу», операторам необходимо отойти от края поля не менее, чем на 1 м?
3. Почему необходима калибровка датчика освещенности перед началом старта?
4. Какие алгоритмические конструкции использовались Вами при выполнении заданий лабораторной работы?
5. Какие элементы знаний по физике и математике необходимы при решении задачи движения робота в круге?

Лабораторная работа 9. Изучение конструкций учебных Lego-роботов

Цель: познакомиться с различными конструкциями *Lego*-роботов, представленными в сети Интернет.

Практические задания

Задание 1. Используя ресурсы сети Интернет, соберите какую-либо модель *Lego*-робота. Рекомендуемые источники:

- <http://mindstorms.lego.com/>

Лего-сообщество, расположенное на официальном сайте *Lego*, где люди со всего мира делятся информацией о придуманных и собранных ими моделях роботов.

- <http://nxtprograms.com/>

Хранилище инструкций по сборке различных моделей роботов с примерами программ.

- <http://odno-lego.ru>

Сайт посвящён роботам *Lego Mindstorms NXT* и *NXT 2.0*. Здесь Вы найдёте множество пошаговых инструкций по сборке роботов *Mindstorms* из *Lego*, программ, видео, обзоры наборов 8527 и 8547 и др.

- <http://www.prorobot.ru/>

Курс «Робототехника и ЛЕГО-конструирование» в школе (для учеников 6–8 класса). Основное внимание уделяется конструированию различных моделей роботов. На данном сайте есть инструкции по роботам *LEGO Mindstorms NXT* на русском языке, готовые инструкции для сборки простых роботов, статьи и обзоры разных версий конструкторов *LEGO*, которые можно использовать на уроках робототехники.

Задание 2. Усовершенствуйте собранного по готовой инструкции робота, внесите изменения в конструкцию и программу.

Контрольные вопросы

1. Назовите модели *Lego*-роботов, которые можно собрать, используя готовые инструкции в сети Интернет?
2. Какой тип шагающих роботов обладает наилучшим равновесием?
3. Предложите идеи тематических занятий, на которых ученикам будет предложено собрать модели *Lego*-роботов по инструкциям из сети Интернет.

ЗАКЛЮЧЕНИЕ

Образовательная робототехника как интегративный курс обладает значительным потенциалом в школьном обучении, отвечая требованиям современного производства, способствуя углублению и систематизации знаний учащихся по основным школьным предметам, позволяя сориентироваться в выборе будущей профессии. С помощью многосторонних межпредметных связей образовательной робототехники с школьными предметами задачи обучения, развития и воспитания учащихся решаются на качественно новом уровне, закладывается фундамент для комплексного подхода в решении сложных проблем реальной действительности.

Задача педагогических вузов заключается в качественной подготовке специалистов по образовательной робототехнике, обладающих системными знаниями в данной области, владеющих современными приёмами и способами обучения, умеющих соединять традиционное с инновационным, алгоритмические предписания с творческими методами и средствами обучения.

ГЛОССАРИЙ

Алгоритм – точное и полное описание последовательности действий, позволяющее получить конечный результат.

Базовое программное обеспечение – программное обеспечение, поставляемое с роботом и предназначенное для организации его функционирования.

Бионика – прикладная наука о применении в технических устройствах и системах принципов организации, свойств, функций и структур живой природы, то есть формы живого в природе и их промышленные аналоги.

Вращательное движение – это движение, при котором траектории различных точек тела представляют собой окружности (или дуги окружностей) с общей осью.

Вспомогательный алгоритм – алгоритм, который целиком используются в составе другого алгоритма.

Датчик – это средство измерений, размещаемое в месте отбора информации, исполняющее функцию первичного преобразователя измеряемой величины в электрическую или электромагнитную величину.

Звук – физическое явление, представляющее собой распространение в виде упругих волн механических колебаний в твёрдой, жидкой или газообразной среде.

Инфракрасное излучение – не видимое глазом электромагнитное излучение, занимающее спектральную область между красным концом видимого света и коротковолновым радиоизлучением.

Кибернетика – наука об управлении, связи и переработке информации.

Кинематика учебного мобильного робота – один из основных этапов исследований при проектировании мобильных роботов. Результатом кинематического анализа является математическое описание поведения механической системы для

дальнейшей разработки программного управления движением учебного робота.

Манипулятор – управляемое устройство, оснащенное рабочим органом для выполнения двигательных функций, аналогичным движениям руки человека при перемещении объектов в пространстве.

Механическая передача – механизм, служащий для передачи и преобразования механической энергии от энергетической машины до исполнительного механизма, как правило, с изменением характера движения (изменения направления, скоростей и др.).

Мехатроника – область науки и техники, основанная на системном объединении узлов точной механики, датчиков состояния внешней среды и самого объекта, источников энергии, исполнительных механизмов, усилителей, вычислительных устройств.

Обратная связь – канал, по которому в систему вводятся данные о результатах управлении.

Органы рабочие манипулятора – различные инструменты, закрепляемые на конце манипулятора, с помощью которых последний выполняет конкретные производственные операции.

Освещенность – световая величина, равная отношению светового потока, падающего на малый участок поверхности, к его площади.

Очувствление – использование информации об окружающей среде в качестве сигналов обратной связи, позволяющих, например, роботу реагировать на изменение среды.

ПИД-регулятор (пропорциональный интегрально-дифференциальный регулятор) – это метод, широко используемый для улучшения работы различных технических устройств.

Полупроводниковые приборы – электронные приборы, действие которых основано на электронных процессах в полупроводниках. Наиболее распространенными из них являются

полупроводниковые фото- и терморезисторы, диоды, транзисторы, полупроводниковые интегральные микросхемы и др.

Поступательное движение – движение, при котором все точки тела имеют одинаковые траектории.

Привод робота – часть исполнительного устройства робота, предназначенная для приведения в движение его звеньев и функциональных элементов.

Программирование – процесс подготовки задач для решения их на компьютере (микрокомпьютере).

Программирование робота – процесс формирования управляющей программы робота.

Программное обеспечение робота – программное обеспечение, предназначенное для организации процесса программирования и исполнения управляющей программы.

Робот – многофункциональная перепрограммируемая машина для полностью или частично автоматического выполнения двигательных функций аналогично живым организмам, а также некоторых интеллектуальных функций человека.

Робот адаптивный – робот, управляющая программа которого целенаправленно изменяет последовательность или характер действий в зависимости от контролируемых факторов рабочей среды и/или функционирования самого робота.

Робот жесткопрограммируемый – робот, действия которого, заданные управляющей программой, не могут быть целенаправленно изменены в процессе работы в зависимости от функционирования робота и/или контролируемых параметров рабочей среды.

Робот интеллектуальный – робот, управляющая программа которого может полностью или частично формироваться автоматически в соответствии с поставленным заданием и в зависимости от состояния рабочей среды.

Робот манипуляционный – робот для выполнения двигательных функций, аналогичных функциям руки человека.

Робот мобильный – робот, способный перемещаться в рабочей среде в соответствии с управляющей программой.

Роботизация – автоматизация ручного или рутинного видов умственного труда человека с применением роботов.

Робототехника – научно-техническое направление, занимающееся проектированием, изготовлением и использованием роботов.

Робототехника – область науки и техники, связанная с созданием, исследованием и применением роботов. Робототехника охватывает вопросы проектирования, программного обеспечения, очувствления роботов, управления ими, а также роботизации промышленной и непромышленной сферы.

Роботы второго поколения – адаптивные роботы, которые могут изменять своё поведение в зависимости от изменения внешних условий.

Роботы первого поколения – программируемые роботы, не имеющие органов очувствления.

Роботы третьего поколений – роботы, наделенные элементами искусственного интеллекта.

Сервомотор – силовой элемент исполнительного механизма, преобразующий энергию вспомогательного источника в механическую энергию перемещения в соответствии с сигналом управления.

Система информационно-управляющая – комплекс измерительно-информационных и управляющих средств, автоматически производящих сбор, обработку и передачу информации, и формирующих различные управляющие сигналы.

Система исполнительная – это устройства, предназначенные для непосредственного воздействия на объекты окружающей среды или взаимодействия с ними в соответствии с управляющими сигналами, формулируемыми информационно-управляющей системой или непосредственно оператором. В качестве элементов исполнительной системы используются двига-

тели, передаточные устройства (передачи), связанные с ними манипуляторы, механические ноги, тележки с колесным, гусеничным и иными шасси и др.

Система сенсорная – это искусственные органы чувств робота, предназначенные для восприятия и преобразования информации о состоянии внешней среды и самого робота.

Система управления роботом – система, состоящая из комплекса аппаратных и программных средств и обеспечивающая формирование и выдачу управляющих воздействий исполнительным устройствам в соответствии с задаваемыми целями и с учетом состояния внешней среды.

Траектория – линия, вдоль которой движется тело.

Ультразвук – звуковые волны, имеющие частоту выше воспринимаемой человеческим ухом (20 000 Герц).

Управляющая программа – программа, задающая действия робота по выполнению им требуемых функций.

Функция преобразования – математическое (или графическое) описание связи изменения выходного сигнала датчика в зависимости от изменения входного сигнала. Функция преобразования датчика может быть как линейной, так и нелинейной.

Lego Mindstorms NXT Software – базовое программное обеспечение робота *Lego Mindstorms NXT*.

Robolab – графическая среда программирования, используемая для программирования *Lego*-роботов на базе *RCX* и *NXT*.

RobotC for Mindstorms – текстовая среда программирования, позволяющая разрабатывать программы для управления *Lego Mindstorms NXT*.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев А.П. Робототехника: учеб. пособие для 8–9 кл. сред. шк. / А.П. Алексеев, А.Н. Богатырев, В.А. Серенко. – М.: Просвещение, 1993. – 160 с.
2. Амперка. Официальный сайт [Электронный ресурс]. – Режим доступа: <http://amperka.ru/>, свободный. – Загл. с экрана.
3. Ардуино [Электронный ресурс]. – Режим доступа: <http://arduino.ru/>, свободный. – Загл. с экрана.
4. Белиовская Л.Г. Программируем микрокомпьютер *NXT* в *LabVIEW* / Л.Г. Белиовская, А.Е. Белиовский. – М.: ДМК Пресс, Москва, 2010. – 280 с.
5. Борковский А.Б. Англо-русский словарь по программированию и информатике (с толкованиями) / А.Б. Борковский. – М.: Русский язык, 1990. – 335 с.
6. Брага Н. Создание роботов в домашних условиях / Н. Брага; пер с англ. Е.А. Добролежина. – М.: НТ Пресс, 2007. – 568 с.
7. Власова, О.С. Технологии образовательной робототехники как средство освоения предметной области «Математика и информатика» / О.С. Власова // Начальная школа плюс До и после. – 2013. – № 10. – С. 61–67.
8. Воротников С.А. Информационные устройства робототехнических систем: учеб. пособие / С.А. Воротников; под. ред. С.Л. Зенкевича, А.С. Ющенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 384 с.
9. Всероссийский учебно-методический центр образовательной робототехники [Электронный ресурс]. – Режим доступа <http://фгосигра.рф/>, свободный. – Загл. с экрана.
10. Дистанционный курс «*LEGO Mindstorms NXT*: основы конструирования и программирования роботов» [Электронный ресурс]. – Режим доступа <http://learning.9151394.ru/course/view.php?id=280>, свободный. – Загл. с экрана.
11. Ершов М.Г. Использование робототехники в преподавании физики / М.Г. Ершов // Информационные компьютерные технологии в образовании. Вестник ПГГПУ. – Вып. 8. – С. 77–85.
12. Злаказов А.С. Уроки Лего-конструирования в школе: методическое пособие / А.С. Злаказов, А.Г. Горшков, С.Г. Шевалдина. – М.: БИНОМ. Лаборатория знаний. – 2011. – 120 с.
13. Ииси Т. Мехатроника / Т. Ииси, И. Симояма, Х. Иноуэ [и др]; пер. с япон. – М.: Мир, 1988. – 318 с.
14. Информатика. Универсальный справочник. / И.А. Трофимова, О.В. Яровая. – 2010. – 288 с.

15. Информация о международных состязаниях роботов [Электронный ресурс]. – Режим доступа: <http://wroboto.ru/>, свободный. – Загл. с экрана.
16. Копосов Д.Г. Первый шаг в робототехнику: практикум для 5–6 классов / Д.Г. Копосов. – М: БИНОМ. Лаборатория знаний, 2012. – 286 с.
17. Котюк А.Ф. Датчики в современных измерениях / А.Ф. Котюк. – М.: Радио и Связь, Горячая линия – Телеком, 2006. – 96 с.
18. Ловин Д. Создаем робота-андроида своими руками / Д. Ловин; пер. с англ. Г. Мельникова. – М.: Издат. дом «ДМК Пресс», 2007. – 312 с.
19. Накано Э. Введение в робототехнику / Э. Накано; пер. с яп. А.И. Логинов, А.М. Филатов. – М.: Мир, 1998. – 334 с.
20. О встраивании робототехники в образовательный процесс в образовательных учреждениях Челябинской области в 2010–2011 учебном году. Приложение к письму Министерства образования и науки Челябинской области от 23.08.2010 г. № 103/3976 [Электронный ресурс]. – Режим доступа <http://oimozlat.edusite.ru/DswMedia/vstraiivanierobototexnikivobrazovatel-nyiyprocess.pdf>, свободный. – Загл. с экрана.
21. Образовательная робототехника в начальной школе: учебно-методич. пособие / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 152 с.
22. Образовательная робототехника во внеурочной деятельности: учебно-методическое пособие / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 96 с.
23. Образовательная робототехника на уроках информатики и физики в средней школе / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011.
24. Опалёва Д.А. Языки программирования и методы трансляции / Д.А. Опалева, В.П. Самойленко. – СПб: БХВ-Петербург, 2005. – 480 с.
25. Оренбургский клуб фанатов *LegoMindstorms*, а также *LegoTechnic* и других конструкторов *Lego* [Электронный ресурс]. – Режим доступа: <http://lego56.ru/>, свободный. – Загл. с экрана.
26. Официальный сайт компании *Lego* [Электронный ресурс]. – Режим доступа: <http://mindstorms.lego.com>, свободный. – Загл. с экрана.
27. ПервоРобот *LEGO WeDo*: книга для учителя [Электронный ресурс]. – Дания: *LEGO Group*. – 2009. – (опт. диск CD-ROM).
28. ПервоРобот *NXT 2.0* Введение в робототехнику [Электронный ресурс]. – *CarnegieMellon. RoboticsAcademie*. – (опт. диск CD-ROM).
29. ПервоРобот *NXT*: Экоград. Комплект заданий: книга для учителя / *The LEGO Group*. – Дания: *ICOM ApS*, 2011. – 102 с.

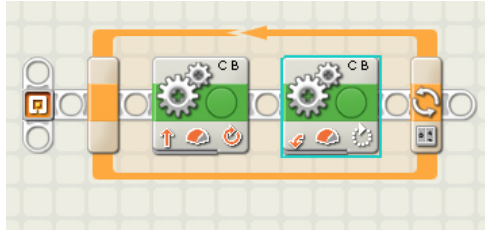
30. Предко М. 123 эксперимента по робототехнике / М. Предко; пер. с англ. В.П. Попова. – М.: НТ Пресс, 2007. – 544 с.
31. Программа «Робототехника: инженерно-технические кадры инновационной России» [Электронный ресурс]. – Режим доступа <http://www.russianrobotics.ru/>, свободный. – Загл. с экрана.
32. Робототехника для начинающих с *Robotis Bioloïd* [Электр.й ресурс]. – Режим доступа: <http://www.wertech.ru>, свободный. – Загл. с экрана.
33. Робототехника для школ и вузов Нижнего Новгорода [Электронный ресурс]. – Режим доступа <http://nnxt.blogspot.ru/>, свободный. – Загл. с экрана.
34. Российский этап Всемирной олимпиады роботов 2014 [Электронный ресурс]. – Режим доступа: <http://wro2014.ru/about/organizing-committee.php>, свободный. – Загл. с экрана.
35. Руководство тренера 2010–2011 // *FIRSTTechChallenge*, 2010. – 52 с.
36. Сайт «Робототехника» [Электронный ресурс]. – Режим доступа: <http://robot-prz.blogspot.com/>, свободный. – Загл. с экрана.
37. Сайт о роботах *LegoMindstormsNXT* и *NXT 2.0* [Электронный ресурс]. – Режим доступа: <http://odno-lego.ru>, свободный. – Загл. с экрана.
38. Семакин И.Г. Преподавание базового курса информатики в средней школе: метод. пособие / И.Г. Семакин, Т.Ю. Шеина. – 2-е изд., испр. и доп. – М.: БИНОМ. Лаборатория знаний, 2004. – 540 с.
39. Соревновательная деятельность региональных ресурсных центров технического творчества для детей и молодежи на базе социально ориентированных НКО на примере Программы «Робототехника» [Электронный ресурс]. – 2013. – 38 с. – Режим доступа: http://www.russianrobotics.ru/netcat_files/userfiles/Documents/upload/Methodichka_blokChB_3_v%20pechat.pdf, свободный. – Загл. с экрана.
40. Терехов Н.А. Образовательный кибернетический конструктор для использования в школах и вузах [Электронный ресурс] – Н.А. Терехов, Р.М. Лучин, С.А. Филипов. – Режим доступа http://www.math.spbu.ru/user/ant/All_articles/096_Terekhov_Filipov_Luchin_robots.pdf, свободный. – Загл. с экрана.
41. Федеральный государственный образовательный стандарт [Электронный ресурс]. – Режим доступа <http://standart.edu.ru>, свободный. – Загл. с экрана.
42. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки «221000 Мехатроника и робототехника (квалификация (степень) «бакалавр»)» утвержден приказом МОиН РФ № 545 от 9 ноября 2009 года [Электронный ресурс]. – Режим доступа http://www.edu.ru/db-minobr/mo/Data/d_09/prm545-1.pdf, свободный. – Загл. с экрана.

43. Филиппов С.А. Робототехника для детей и родителей / С.А. Филиппов. – СПб.: Наука, 2013. – 319 с.
44. Филиппов С.А. Основы робототехники на базе конструктора *Lego Mindstorms NXT*. Занятие 5. Движение по линии / С.А. Филиппов // Компьютерные инструменты в школе. – № 5. – 2010. – С. 34–43. [Электронный ресурс]. – Режим доступа: <http://technocontext.ru/>, свободный. – Загл. с экрана.
45. Хранилище инструкций по сборке различных моделей роботов с примерами программ [Электронный ресурс]. – Режим доступа: <http://nxtprograms.com/>, свободный. – Загл. с экрана.
46. Шимов И.В. Применение робототехнических устройств в обучении программированию школьников / И.В. Шимов // Педагогическое образование в России, 2013. – № 1. – С. 185–188.
47. Юревич Е.И. Основы робототехники / Е.И. Юревич. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2005. – 416 с.
48. Berns K., Programmierung mit LEGO MINDSTORMSNXT / K. Berns, D. Schmidt. – eXamen.press, The LEGO Group 2010. – 246 s.
49. CarnegieMellon. RoboticsAcademie [Электронный ресурс]. –Режим доступа: <http://www.education.rec.ri.cmu.edu/content/lego/curriculum/>, свободный. – Загл. с экрана.
50. Extreme NXT: Extending the LEGO Mindstorms NXT to the Next Level / M. Gasperi, P.E. Hurbain, I.L. Hurbain. – Apress, 2007. – 286 p.
51. HiTechnic. Официальный сайт [Электронный ресурс]. – Режим доступа: <https://www.hitechnic.com>, свободный. – Загл. с экрана.
52. KeeD. ClassroomActivitiesfortheBusyTeacher: NXT [Электронный ресурс]. – Режим доступа: <http://www.damienkee.com/classroom-activities-nxt/>, свободный. – Загл. с экрана.
53. KeeD. Technologyineducation[Электронный ресурс] / D. Кее. – Режим доступа: <http://www.damienkee.com>, свободный. – Загл. с экрана.
54. Sluka J. A PID Controller For LEGO® Mindstorms Robots (Part 1) [Электронныйресурс] / J. Sluka // Hispabrick Magazine 008. EnglishEdition. – P. 20–26. – Режим доступа: // www.hispabrickmagazine.com,свободный. – Загл. с экрана.
55. Trobaugh J.J. Winning Design! Lego Mindstorms NXT / J.J. Trobaugh. – 2010. – 292 p.

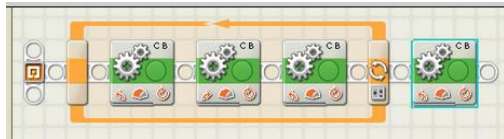
ПРИЛОЖЕНИЕ 1

Скриншоты программ, входящих в практические задания к лабораторным работам

Лабораторная работа 2.
Задание
«Движение по квадрату»



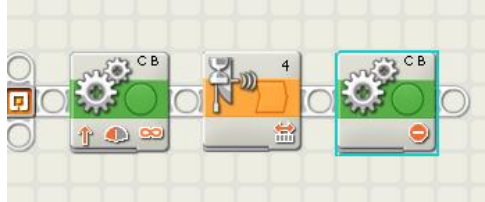
Лабораторная работа 2.
Задание
«Робот описывает
восьмерку»



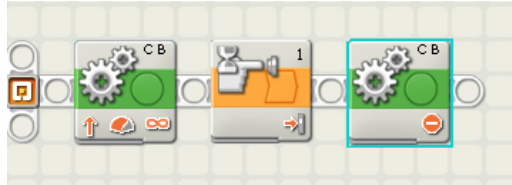
Лабораторная работа 2.
Задание
«Активация робота
звуком»



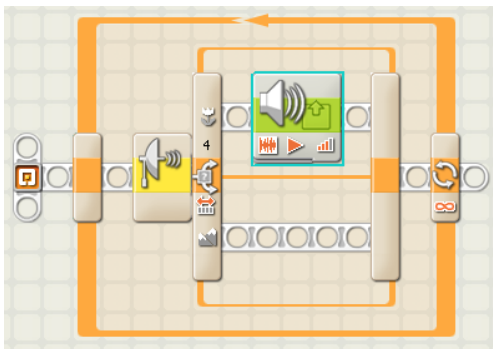
Лабораторная работа 2.
Задание
«Обнаружение препятст-
вия с помощью
датчика расстояния»



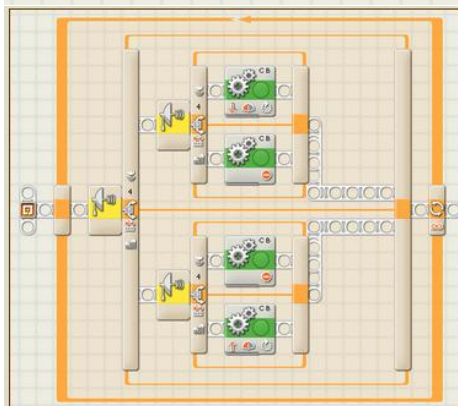
Лабораторная работа 2.
Задание
«Обнаружение препятст-
вия с помощью
датчика касания»



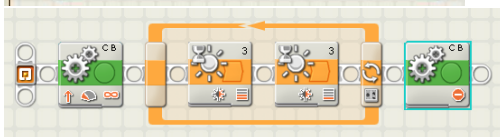
Лабораторная работа 2.
Задание
«Охранная
сигнализация»



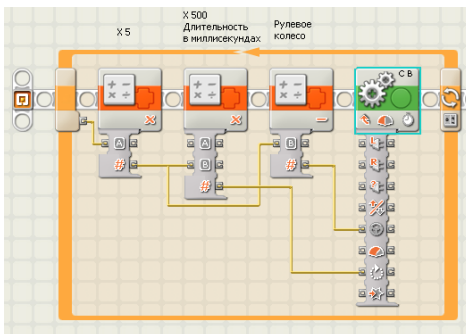
Лабораторная работа 2.
Задание
«Робот-прилипала»



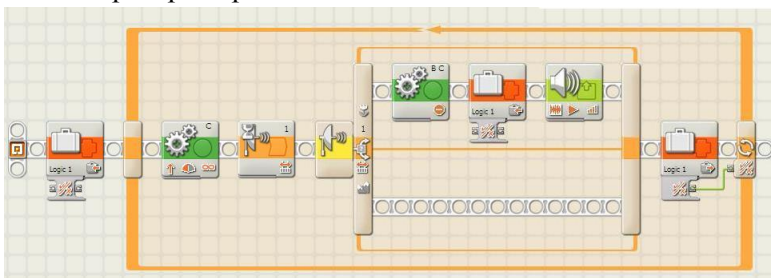
Лабораторная работа 2.
Задание
«Поиск заданной
линии»



Лабораторная работа 2.
Задание
«Движение по спира-
ли»

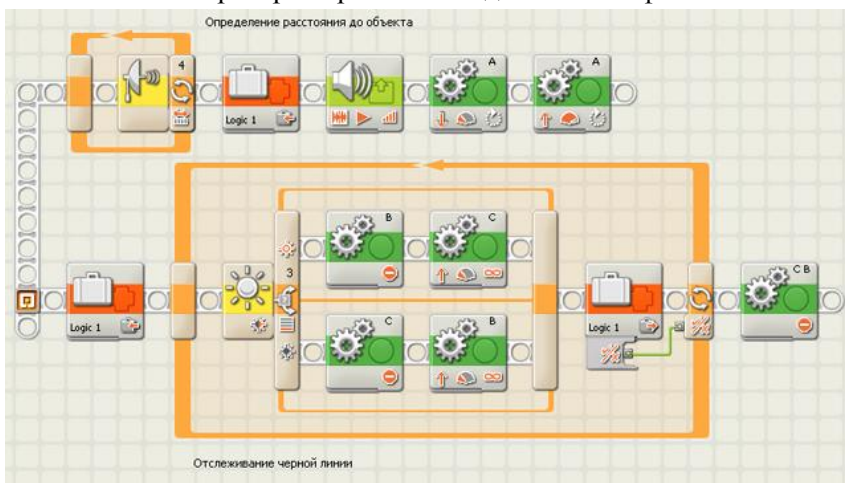


Лабораторная работа 2. Задание «Робот-наблюдатель»



Комментарий: в программе использована логическая величина – *Logic1*, начальное значение которой *False*, при наступлении отслеживаемого события – значение меняется на *True*.

Лабораторная работа 2. Задание «Гольф»



Комментарий: Датчик освещенности должен смотреть вниз и следить за линией в процессе движения робота. Одновременно с этим робот должен контролировать расстояние до препятствия. При обнаружении мяча робот должен нанести удар. Так как трасса извилистая, робот может обнаружить мяч и

попытается нанести удар раньше, чем дойдет до ее конца. Также, двигаясь вдоль черной линии с датчиком освещенности по классическому алгоритму, робот, заметив препятствие (верхний край подставки), по инерции может проехать одним колесом в сторону и ударить таким образом мимо цели.

Возможно следующее решение задачи: в программе используется возможность *NXT* выполнять параллельно несколько программ или процессов. Верхняя ветвь программы определяет расстояние до объекта, в нашем случае – подставки с мячом. Нижняя ветвь управляет двигателями, отслеживая черную линию. Обмен информацией между этими параллельными процессами происходит через специальную логическую переменную *Logic 1*.

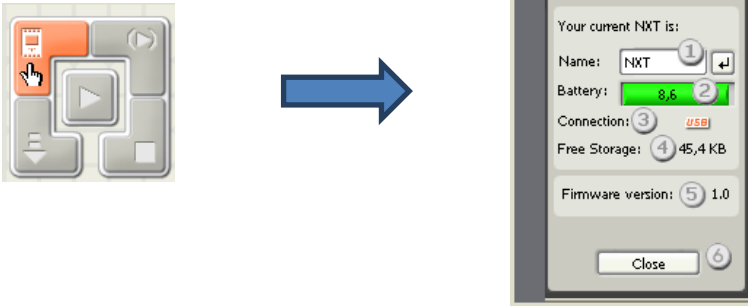
На верхней ветке в ней записано значение «Ложь». Это дает знать нижней ветви программы, что робот приблизился к мячу на расстояние удара.

На нижней ветке изначально переменной *Logic 1* присваивается значение *True*. Это означает, что робот еще не достиг мяча и что он должен двигаться, следуя черной линии. Третий раз мы обращаемся к переменной *Logic 1* для считывания её текущего значения. Если ее значение *False*, цикл прекращает свою работу, после чего программа выключает двигатели и робот останавливается.

ПРИЛОЖЕНИЕ 2

Панель диагностики блока NXT

Для диагностики блока NXT предназначена специальная панель, вызываемая при нажатии кнопки в верхнем левом углу пульта управления.



Пользователь сразу получает доступ к следующим параметрам *NXT*:

1. Имя *NXT*. Его можно ввести в это поле и нажать «Ввод»;
2. Уровень заряда батареи *NXT*;
3. Тип подключения к компьютеру (*Bluetooth* или *USB*);
4. Объем свободной памяти;
5. Версия операционной системы *NXT*;
6. Кнопка *Close* закрывает панель управления и диагностики *NXT*.

Иногда возникает необходимость в управлении памятью *NXT* и файлами, хранящимися в ней. Это можно сделать, выбрав вкладку Память (*Memory*) на панели *NXT*, где в графическом виде показано использование памяти *NXT*. Здесь можно обмениваться программами, звуками, графикой и другими файлами между компьютером и *NXT*, копировать и удалять файлы, хранящиеся в памяти *NXT*.

Учебное издание

Татьяна Владимировна Никитина

**ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА
КАК НАПРАВЛЕНИЕ ИНЖЕНЕРНО-ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА ШКОЛЬНИКОВ**

Учебное пособие

Работа рекомендована РИСом ЧГПУ

Протокол № _____, 2014 г.

ISBN978-5-906777-21-8

Издательство ЧГПУ

454080, г. Челябинск, пр. Ленина, 69

Редактор О.В. Максимова

Технический редактор Т.Н. Никитенко

Подписано в печать 28.11.2014

Объем 8,2 уч.-изд. л. Формат 60×84/16

Бумага офсетная. Тираж 100 экз.

Заказ № _____

Отпечатано с готового оригинал-макетав типографии ЧГПУ

454080, г. Челябинск, пр. Ленина, 69