МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.П. АСТАФЬЕВА»
(КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики

Выпускающая кафедра технологии и предпринимательства

Трясина Михаила Владимировича

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема «Выбор языка программирования роботов в контексте повышения сложности и разнообразия решаемых задач»

Направление подготовки 44.03.01 Педагогическое образование

Направленность (профиль) образовательной программы Технология

ДОПУСКАЮ К ЗАЩИТЕ
Зав. кафедрой технологии
и предпринимательства,
к.т.н., доцент
С. В. Бортновский
« 09 » июня 2020

Руководитель
к.т.н., доцент
И.В. Шадрин

Дата защиты «03» июля 2020

подпись

09 июня 2020 г.
Оценка отлично

Красноярск 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Слово «робототехника» (от анг. «robotics»), впервые было употреблено при печати писателем Айзеком Азимовым в научно-фантастическом рассказе под названием «Лжец», который был опубликован в 1941 году. Этим же автором в рассказе «Хоровод» (1942) были впервые успешно сформулированы три закона робототехники — обязательные правила, которым должен следовать робот — это базовые правила робототехники.

Изучение основ робототехники очень перспективно и важно, т.к. в настоящее время руководство страны четко сформулировало первоочередной социальный заказ в сфере образования: в настоящее время нашей стране не хватает профессиональных инженеров. Для этого нужно активно начинать популяризацию профессии инженера уже при преподавании с начальных классов. Робототехника является популярным и эффективным связующим звеном, позволяющим рассматривать широкий круг вопросов из разных областей науки, позволяющим воплотить в жизнь самые смелые инженерные замыслы на уроках технологии.

На занятиях с образовательными конструкторами в учебных учреждениях учащиеся строят действующие модели реальных механизмов, живых организмов и машин, проводят естественнонаучные эксперименты, осваивают основы информатики, алгоритмики и робототехники, попутно укрепляя свои знания по математике и физике и приобретая навыки работы в творческом коллективе.

На сегодняшний день 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, Fischertechnic, Arduino, Roborobo, Bioloid и др.

Однако, очень часто школы ограничиваются приобретением одного конструктора, например, Lego Mindstorms EV3 или Arduino. Не смотря на то, что эти наборы являются наиболее распространёнными и методически

проработанными, педагог не должен ограничивать кругозор рассмотрением единственной робототехнической платформы. С повышением уровня образования возрастает сложность задач, и ширина спектра используемого аппаратного и программного обеспечения будет определять эффективность разрабатываемых решений.

Актуальность работы обусловлена разрозненностью подходов к выбору языков программирования робототехнических устройств на занятиях по Технологии, отсутствием целостного представления о задачах, ставящихся на разных уровнях обучения, и понимания необходимости применения различных инструментов для их решения.

Объект исследования: содержание учебного курса по Робототехнике в рамках уроков Технологии в средней школе.

Предмет исследования: выбор среды программирования роботов в контексте повышения сложности и разнообразия решаемых задач.

Цель исследования: разработка методических рекомендаций для реализации комплексного подхода к программированию робототехнических устройств на уроках Технологии.

В соответствии с поставленной целью, объектом и предметом исследования были поставлены следующие **задачи** исследования:

1. Провести анализ научной и методической литературы по организации учебной деятельности обучающихся при изучении робототехники на уроках Технологии.
2. Установить круг доступного и распространенного аппаратного и программного обеспечения роботов для использования на уроках Технологии.
3. Провести сравнительный анализ языков программирования роботов в контексте решаемых учебных задач.
4. Разработать методические рекомендации по использованию языков программирования роботов в контексте повышения сложности и разнообразия решаемых задач.

Для решения поставленных задач использованы следующие методы исследования:

1. Системный подход;
2. Комплексная методика, включающая теоретическое изучение и анализ научно-педагогической литературы;
3. Стандартизированные методики изучения проектной деятельности (анализ, синтез, сравнение, наблюдение);
4. Методы и алгоритмы управления роботами и робототехническими системами;

Теоретическую основу выпускной работы составили исследования по:

- вопросам включения основ робототехники в обучении детей в школьном и дополнительном образовании (Х.Х. Абушкин, Д.В. Андреев, О.С. Власова, К.А. Вегнер, Р.А. Галустов, Л.Н. Гостева, А.В. Дадонова, А.Н. Дахин, М.Г. Ершов, А.С. Злаказов, О.С. Нетесова, Т.В. Никитина, Н.П. Петрова, С.А. Филиппов, В.Н. Халамов, И.В. Шимов и др.);

- основам программирования на уроках робототехники (Байктал Дж., Ю.А. Винницкий, К.Ю. Поляков, В.Г. Сафули, Н.Г. Дорожкина, Е.И Рыжая, М.А. Стерхова, В.В. Тарапата, А.А. Салахова, А.В. Красных и др.).

Теоретическая значимость работы заключается в определении места и роли робототехники в рамках образовательного процесса, обосновании форм и методов обучения, способствующих развитию навыков программирования роботов.

Практическая значимость: заключается в том, был сделан качественный отбор программно-технического обеспечения элементов конструирования и робототехники, направленный на развитие навыков программирования роботов на уроках технологии курса робототехники.

Структура работы - соответствует логике исследования и включает в себя введение, две главы, заключение, библиографический список.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРЕПОДАВАНИЯ РОБОТОТЕХНИКИ В ШКОЛЬНОМ КУРСЕ ТЕХНОЛОГИИ.

1.1. Робототехника в системе школьного образования.

Образовательная робототехника - это междисциплинарная область школьного образования, объединяющая знания физики, мехатроники, машиностроения, математики, электроники, кибернетики и информационно-коммуникационных технологий, позволяющая учащимся разного возраста участвовать в процессе инновационного научно-технического творчества. Робототехника - одна из самых передовых областей науки и техники. Она направлена на развитие популярности научно-технического творчества и повышение имиджа инженерных профессий среди молодых людей, развитие способности молодежи решать реальные инженерные задачи и работать с оборудованием.

Представители системы общего образования подразумевают под робототехникой особенный вид образовательного комплекса оборудования, использование которого позволяет проводить уроки с детьми школьного возраста на достаточно интересном и современном уровне.

Высшие учебные заведения рассматривают робототехнику как отдельный предмет, который связан с специалистами, которых они готовят.

Рассмотрим подходы к определению понятия «образовательная робототехника».

В. Н. Халимов определяет робототехнику как «универсальный инструмент» общего образования. По его мнению, «робототехника прекрасно вписывается в дальнейшее образование, внеклассную деятельность, а также преподавание предметов и в соответствии с требованиями ФГОС. Он подходит для всех возрастов - от дошкольников до студентов. А использование роботизированных устройств на занятиях - это еще и тренировка технического творчества одновременно, что способствует воспитанию активных, увлеченных людей с инженерным и дизайнерским

мышлением. Образовательная робототехника позволяет выявить технические тенденции студентов на ранней стадии и развить их в этом направлении».

А. А. Крицын в своей статье утверждает, что «робототехника-это еще и спорт, хобби и отдых для российских школьников. Он также убежден, что образовательная робототехника - это не только класс, но и комплекс мероприятий, направленных на развитие навыков и технического творчества, мотивацию студентов к изучению точных наук и обеспечение их своевременной профессиональной ориентации, объединяющий проектирование и программирование в единое целое».

А. В. Литвин рассматривает образовательную робототехнику как средство формирования проектной компетентности студентов. В своей статье он пишет, что учебная робототехника является эффективным материалом и инструментом для проектной деятельности студентов, поскольку каждый робот уже является мини-проектом.

М. В. Кузьмина считает, что образовательная робототехника способна сочетать мехатронику, проектирование и программирование, что способствует интеграции STEM-образования, обучению информатике, математике, физике, рисованию и естественным наукам с развитием приемов мышления через техническое творчество.

Д. С. Кошева, И. О. Ефремова раскрыли понятие образовательной робототехники как «цикл мероприятий в общеобразовательных школах или образовательных учреждениях непрерывного образования, в ходе которых программирование и проектирование позволяет сформировать навыки технического творчества, мотивацию учащихся к изучению точных наук и обеспечить их своевременную профессиональную ориентацию».

И. Х. Саламов и З. Б. Алхатова считают, что образовательная робототехника способствует развитию навыков, способствующих реализации основных задач научно-технического прогресса.

Таким образом, становится ясно, что робототехника - это универсальная сфера деятельности, которая может быть реализована как в основном, так и в дальнейшем образовании.

С. А. Филиппов считает, что ситуация с робототехникой очень похожа на проблему введения информатики в качестве школьного предмета. Злые языки говорили: «компьютеры-это для инженеров, и нет никакой необходимости заставлять обычных людей подвергаться воздействию радиации, мониторам». Но мы все знаем, что информатика стала школьной темой. В робототехнике есть все, что нужно считать наукой, достойной изучения. Кроме того, эта наука носит практический характер и не оторвана от жизни: дети могут использовать полученные ими знания по физике, математике, информатике, биологии и химии.

О. В. Петракова и Р. Ю. Ракитин считает, что «внедрение робототехники в школьные предметы позволит заинтересовать учащихся, разнообразить учебную деятельность, использовать групповые активные методы обучения и решать практические задачи. Программирование настоящего робота, который поможет вам увидеть законы математики не на страницах тетради или учебника, а в окружающем нас мире».

Введение предмета робототехники в школах, однако, приносит не только пользу, но и некоторые недостатки, о которых писали В. А. Морозов и К. И. Ивкина в своих статьях. «Образовательные программы должны быть полностью перестроены, и часть времени должна быть потрачена на изучение новых технологий. В то же время многие преподаватели не могут похвастаться знаниями в этой области, но все возникающие проблемы можно решить».

Поэтому главная проблема при внедрении робототехники как школьного предмета трудоемка: «Внедрение роботов в основной учебный процесс требует много времени на подготовку учителей. Не хватает учебников и методических рекомендаций для преподавателей и студентов.»

Образовательная робототехника в данный момент недостаточно укомплектована учебниками для учащихся. Дети в роботизированных классах получают много информации. Чтобы систематизировать и закрепить полученные знания, необходимо дать им домашнюю самостоятельную работу. К. Д. Ушинский считал, что только самостоятельная работа создает условия для глубокого усвоения знаний и развития мышления учащихся.

Одним из основных результатов обучения в школе, согласно требованиям нового ФГОС, является умение учиться самостоятельно.

И. А. Зима представляет самостоятельную работу школьника как более широкое понятие, чем только одни домашние задания. Самостоятельная работа должна включать в свою деятельность внеклассные уроки. Самостоятельная работа должна вытекать из правильно организованной учебной деятельности учащегося на уроках технологии. Это специфический вид учебной деятельности школьника. Это высшая форма учебной деятельности ученика, форма самооценки, которая связана с его работой на уроке. Это один из видов форм работы над индивидуальными планами, которые необходимы для того, что бы дополнить, расширить и углубить знания, получаемые ребенком на уроках технологии. Автор в одной из своих статей пишет, что в определении деятельности «самостоятельная работа должна быть организована самим учащимся в соответствии с его внутренними познавательными темами, на наиболее удобное, рациональное с его точки зрения время, руководимое им в процессе и в результате его деятельности, на основе системы внешкольного косвенного управления преподавателем».

Ю. М. Кулюткин определяет общую интеллектуальную самостоятельность как важное качество личности, имеющее существенное значение для творческой направленности личности и продуктивности ее деятельности. По его мнению, духовная самостоятельность проявляется в умении учащихся ставить цели, определять свои задачи, выбирать средства и методы решения этих задач. Самостоятельность проявляется также в

способности человека планировать, организовывать, регулировать свою деятельность, а также в хорошо развитой деятельности самоконтроля и самоуважения. Автор обобщает свои мысли, пишет, что психологическая независимость - это способность человека самостоятельно управлять своей деятельностью.

Ученые выделяют разные степени психологической независимости. Таким образом, в зависимости от сложности решаемых учениками образовательных задач выделяют три уровня. Первая степень самостоятельности развития характеризуется тем, что учащиеся способны решать типовые задачи, а именно те, которые требуют простого воспроизведения способов использования полученных на курсах знаний. Вторая степень означает, что ученики демонстрируют способность самостоятельно решать нестандартные задачи, прибегая к самостоятельным методам поиска, деятельности, операций, необходимости установления межсистемных связей, обобщения, классификации и доказательства. Третий и высший уровень самостоятельности проявляется в умении школьника решать творческие задачи.

Согласно теории постепенного формирования умственной деятельности П. Я. Гальперина и Н. Ф. Талызина, способности проявляются в деятельности и формируются в ней. Это означает, что сегодня образовательный процесс должен быть «активным»: дети не должны получать какие-то готовые знания, а" добывать " их сами в процессе своей деятельности. При этом важна не только деятельность детей, но и сам процесс обучения, в ходе которого эффективно формируются необходимые общеобразовательные умения и навыки: умение ставить цели, принимать решения, делать выбор и доводить его до конца, рефлексировать и так далее. Иными словами, формируется именно способность к обучению, способность к самоконтролю и самореализации.

Образовательные конструкторы для робототехники обычно снабжены базовым набором инструкций для быстрого начала обучения. Технология 4С

включает в себя подключение, установку, обсуждение и совершенствование робототехнической модели, что способствует постепенному развитию и самостоятельности учащихся.

1.2. Аппаратное обеспечение уроков «Основы робототехники».

Конструктор LEGO Education Wedo 2.0.

Конструктор Wedo 2.0 (Рис. 1) поставляется вместе с программным обеспечением и представляет собой готовое решение для развития научной деятельности, навыков проектирования, абстрактного мышления и грамотности изложения. Главные отличия от предыдущей версии конструктора: автономность от компьютера, увеличенное количество деталей и бесплатное программное обеспечение.

Рис.1. Конструктор LEGO Education Wedo 2.0.

Рис. 2. СмартХаб.

СмартХаб (Рис. 2) работает как беспроводной соединитель между вашим устройством и другими электронными компонентами. Используется технология Bluetooth Low Energy. В него передаются программные строки от устройства и робот исполняет их.

В качестве источника питания в СмартХаб используются батарейки AA или присоединяется дополнительная аккумуляторная батарея.

Рис. 3.Средний мотор.

Средний мотор (Рис. 3) – это мотор, которые заставляется двигаться другие компоненты. Ось среднего мотора приводится в движение при помощи электричества. Мотор возможно запускать с требуемой скоростью (мощностью) в обоих направлениях бесконечно, но только на определенное время или до выполнения конкретного условия.

Рис.4. Датчик наклона.

Датчик наклона (Рис. 4) необходим для отслеживания изменения положения в шести различных позициях:

• наклон в одну сторону;

• наклон в другую сторону;

• наклон вверх;

• наклон вниз;

• без наклона;

• любой наклон.

Рис.5. Датчик перемещения.

Датчик перемещения (Рис. 5) необходим для определения изменения в расстоянии до объекта в его радиусе действия тремя способами:

• объект приближается;

• объект удаляется;

• объект изменяет положение.

Конструктор LEGO MindStorms EV3

Данный набор конструктора распространяется в двух версиях. Первый - набор для домашнего пользования (Рис. 6) и второй - набор для обучения (Рис. 7). Различие этих наборов заключается в том, что набор для обучения является более универсальным в использовании из-за большего количества датчиков, которые при покупке идут в комплекте.

Так же существуют ресурсные наборы, которые расширяют возможности конструирования технической основы робота. Они включают в себя балки, шестерни, колеса и другие, в том числе декоративные, элементы.

Рис.6. Домашний комплект.

Рис.7. Обучающий комплект.

Рассмотрим набор для обучения и разберем все его составляющие. В основе набора лежит набор под названием LEGO Technic, который состоит из следующих деталей: балки различных форм и размеров, крепежные элементы, кронштейны, различные инструменты, декоративные детали (бамперы и крылья). Их цель-создать каркас робота, на котором в процессе сборки будут установлены двигатели, датчики и модуль управления.

Модуль управления EV3(Рис. 8), который служит центром управления и энергетической станцией для робота.

Рис.8. Модуль управление EV3.

На экране отображается то, что воспроизводится внутри модуля управления контроллером EV3, и позволяет использовать встроенный интерфейс модуля. Также в нём имеется возможность выводить текст, числовые или графические ответы в программу, или эксперименты.

При помощи кнопок управления модуля можно перемещаться по интерфейсу EV3. В том числе их возможно использовать в качестве программируемых активаторов.

Порт ПК Mini USB используется для связи с персональным компьютером, он находится рядом с портом D и используется для соединения модуля управления EV3 с компьютером.

Порты входа под номерами 1, 2, 3 и 4 необходимы для подключения дополнительных датчиков к модулю управления EV3.

Порты входа A, B, C и D необходимы для подключения моторов к модулю управления EV3.

Все звуки, которые воспроизводятся в модуле управления EV3, исходят из динамика, включая все звуки, которые воспроизводятся при выполнении программы робота. Если же качество звука имеет весомую значимость, то необходимо стараться не закрывать динамик при выполнении программы роботом.

Разберем моторы, датчики, которые входят в комплект набора.

Рис.9. Большой мотор.

Большой мотор (Рис. 9) — самый сильный мотор по мощности, который имеется в наборе. В нем встроен датчик вращения с разрешением 1 градус, необходимый для контроля. Большой мотор нужен в роли привод-

ной платформы в собираемых роботах. Используя такие программные блоки, как «Рулевое управление» или «Независимое управление моторами» в программном обеспечении EV3, возможно скоординировать работу двух моторов одновременно.

Рис.10. Средний мотор.

Средний мотор (Рис. 10) так же имеет встроенный датчик вращения, но имеет меньший вес, нежели большой мотор. Это означает, что время реакции данного мотора намного выше. Средний мотор имеет возможность быть запрограммированным таким образом, чтобы он имел возможность включаться или выключаться, контролировал уровень питания блока управления, осуществлял свою работу в течение определенного времени или выполнял запрограммированное число оборотов двигателя.

Рис.11. Датчик света.

Датчик цвета (Рис. 11) — это цифровой датчик, который осуществляет проверку цвета или яркость света, который поступает в небольшое окошко, расположенное на лицевой стороне. Датчик работает в трех режимах: «Цвет», «Яркость отраженного света» и «Яркость внешнего освещения».

При работающем режиме «Цвет» датчик цвета может распозновать семь различных цветов: черный, синий, зеленый, желтый, красный, белый и коричневый, а также информировать об отсутствие цвета. Данная возможность означает, что программируемый робот может осуществлять такие действия, как сортировка цветных мячей или кубиков, озвучивал найденные им цвета или останавливал свою работу в тот момент, когда найдёт определенный цвет, например красный.

При работающем режиме «Яркость отраженного света» датчик цвета способен определить яркость света. Датчик при подсчёте использует шкалу от нуля (очень темный) до ста (очень светлый). Получается, что робота можно запрограммировать так, чтобы он передвигался по белой поверхности до того момента, пока не будет обнаружен на его пути черная линия, или запрограммировать его так, что бы он преобразовывал идентификационную карточку с цветовым кодом.

При работающем режиме «Яркость внешнего освещения» датчик цвета способен определить силу света, который поступает в окошко из внешней среды. Датчик использует шкалу от нуля (очень темный) до ста (очень светлый). Это значит, что работу робота можно запрограммировать так, чтобы он издавал звуковой сигнал утром, когда встаёт солнце, или он завершал своб работу, если свет потух.

Рис.12. Датчик касания.

Датчик касания (Рис. 12) — это аналоговый датчик, который имеет способность определять, когда кнопка датчика нажата, а когда отпущена.

Рис.13. Инфракрасный датчик.

Инфракрасный датчик (Рис. 13) — это цифровой датчик, который имеет способность реагировать на инфракрасный цвет, который был отражен от расположенных рядом объектов.

Так же имеет возможность находить инфракрасные световые сигналы, которые были излучены удаленным инфракрасным маяком. Инфракрасный датчик осуществляет свою работу в трёх режимах: режиме приближения, в режиме маяка и дистанционном режиме. В режиме приближения инфракрасный датчик использует световые волны, которые были отражены от объекта. Это необходимо, чтобы рассчитать расстояние между датчиком и объектом. Он сообщает о расстоянии, используя значения от нуля (очень близко) до ста (далеко). Датчик не может дать ответ в точном расстоянии, например в сантиметрах. Датчик может осуществить обнаружение объектов длиной до 70 см, в зависимости от того, какие размеры и формы он собой представляет. Так же имеется возможность воспринимать сигналы от инфракрасного пульта дистанционного управления, что позволит разрабатывать роботов на пульте дистанционного управления.

Рис.14. Ультразвуковой датчик.

Ультразвуковой датчик конструктора EV3 (Рис. 14) осуществляет генерацию звуковых волн и фиксирует их отражение от объектов, при этом производится измерение расстояния. Данный датчик также может быть подключен в режиме сонара, который осуществляет излучение отдельными волнами. Кроме этого, датчик способен обнаруживать звуковые волны, которые будут триггерами для запуска запрограммированных программ. Например, учащиеся смогут использовать датчик для создания системы мониторинга дорожного движения и измерения расстояния между приближающимися транспортными средствами. Благодаря использованию данного датчика становится понятными и увлекательными принципы ультразвуковой технологии и ее применения в автоматических дверях, машинах и заводских системах.

Рис.15. Гироскопический датчик.

Гироскопический датчик конструктора EV3 (Рис. 15). Цифровой гироскопический датчик конструктора EV3 даёт возможность измерить движение вращения робота, а также уловить двадцать три изменения положения и движения робота. Измерение углов производится с точностью до ± 3 градуса; встроенный гироскоп способен уловить вращения с моментом до 440 град/с; частота ответа до 1 кГц.

Плата Arduino.

Рис.16. Основная плата Arduino.

Благодаря тому, что Arduino открыта, его устройство известно и позволяет свободное изменение и дополнение. Поэтому любой из производителей плат имеет возможность производить аналоговую плату Arduino, вносить изменения в плата, а так же производить выпуск свободной комплектации, которые имеют название kits.

Arduino — это бренд производителя, именно поэтому выпущенные аналоги носят другое название, но обычно осуществляется с созвучным с оригинальным — Freduino, Freeduino, DCcduino, Xdruino, Funduino, Robotale. Некоторые производителе не дают название своим платам, а указывают лишь надпись типа for Arduino. Оригинальные платы Arduino (Рис. 16) производятся в Италии, а большинство аналогов данных плат — в Китае. Так же есть российские разработки.

Большинство плат Arduino выглядит одинаково (у оригинальной платы Arduino имеется фирменный логотип) и производятся в одной цветовой гамме, как оригинал (есть некоторые исключения, Funduino, например, красный). Аналоговые платы Arduino обычно создаются с теми же префиксами в содержании названиях, что и сам Arduino, например, DFrduino UNO соответствует Arduino UNO.

Производитель рекомендует начинать изучение Arduino с набора под названием Arduino Starter Kit (Рис. 17). Этот комплект включает в себя плату

Arduino UNO и другие электронные составляющие, которые необходимы для того, что бы поверхностно изучить Arduino: светодиоды, резисторы, сервопривод, мотор, кнопку, жидкокристаллический экран, пьезоэлемент, датчики и другие компоненты.

Рис.17. Arduino Starter Kit.

1.3. Базовые алгоритмические конструкции при создании программ для роботов.

Программирование является неотъемлемой частью процесса создания даже самого простого робота. На ранних этапах проектирования программы разрабатывается алгоритм – это точное и полное описание последовательности действий, которые позволяют получить конечный результат.

Основные требования, которые предъявляются к алгоритмам:

1. Алгоритм должен представлять конечное решение;
2. Задачи в виде последовательности простых задач;
3. Определенность: подзадачи алгоритма должны быть однозначными и быть понятными для выполнения алгоритма;

4. Массовость: алгоритм должен выполнять решение не только для одного набора значений, а для целого класса задач, который реализуется диапазоном возможных исходных данных;
5. Результативность: алгоритм должен выдавать конкретный результат, т.е. должны быть предусмотрены все возможные варианты и для каждого из них получен свой результат;
6. Конечность: количество выполнений шагов алгоритма должно быть ограниченным.

Для описания алгоритма распространяются следующие методы:

1. Словесный способ: описание алгоритма ведется на обычном языке с разбиением на пошаговые действия;
2. Блок-схемы (графический способ): изображение алгоритма в виде графика при помощи специальных значков-блоков, блоки соединены между собой линиями, которые отображают поочередность выполняемых действий;
3. Формальные алгоритмические языки (языки программирования): при программировании алгоритмов используется строго определенный набор из символов и составленных из них специальных зарезервированных слов, формируются из строгих правил построения языковых конструкций;
4. Псевдокод: объединение алгоритмических и обычных языков.

Элементы некоторых базовых языков используются для строгого написания стандартных структур алгоритма. Каждый алгоритм может быть выполнен с использованием комбинации из трех основных логических структур:

1. Следование;
2. Ветвление;
3. Цикл.

Логическая структура «следование» означает то, что выполнение задачи передается поочерёдно от одного действия к следующему (Рис. 18).

Рис.18. Последовательное выполнение шагов алгоритма на языке блок-схем.

Логическая структура «ветвление» применяется, когда схема выполнения программы может быть изменена в соответствии с результатом проверки условий и принимает два альтернативных пути. Условие является определенным утверждением и может быть истинным или ложным. Каждый путь ведет к общему выходу, поэтому алгоритм будет продолжать работать, независимо от того, какой путь выбран. Существует полное и неполное ветвление. В полном ветвлении, если условие истинно, то реализуется условие 1, а если оно неверно, то реализуется условие 2 (Рис. 19). В случае неполного ветвления, если условие выполнено (оно истинно), условие 1 реализуется, в противном случае ничего не происходит (Рис. 20).

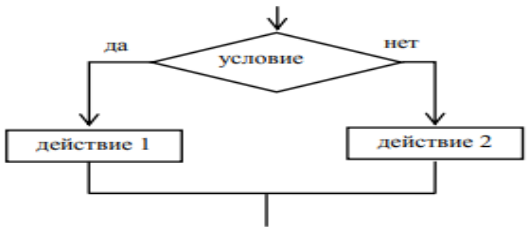

Рис.19.Полное ветвление на языке блок-схем.

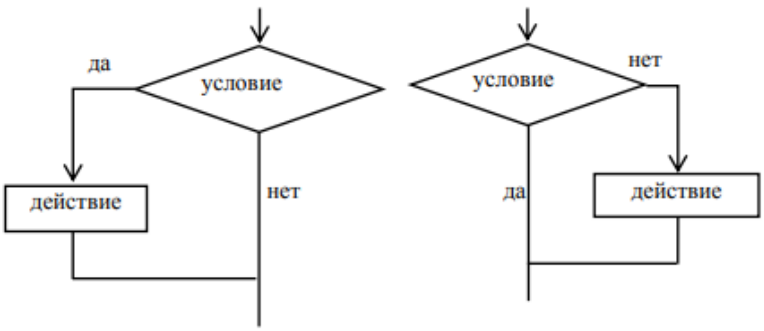

Рис.20. Неполное ветвление на языке блок-схем.

Одним из вариантов логической структуры «ветвления» в алгоритме является оператор выбора. Это может быть полное ветвление (Рис. 21) или неполный ветвление (Рис. 22).

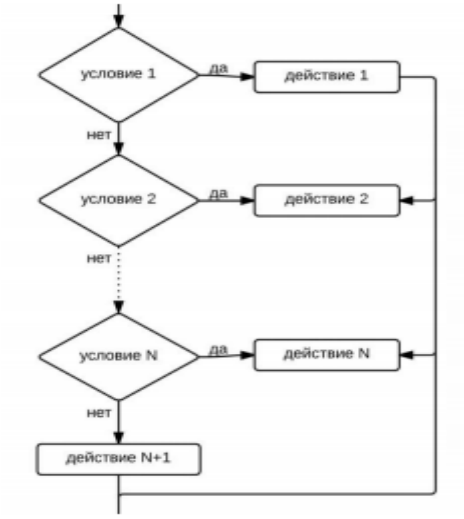

Рис.21. Полный выбор на языке блок-схем.

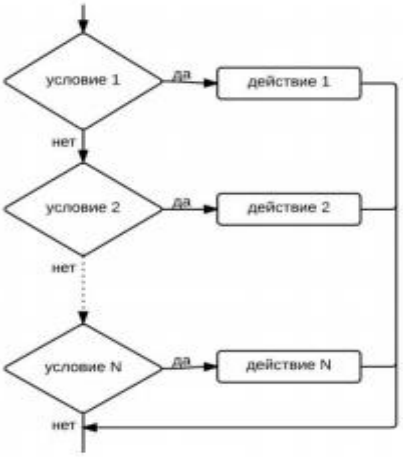

Рис.22. Неполный выбор на языке блок-схем.

Логическая структура «цикл» используется для написания алгоритмов, в которых часть алгоритма (тело цикла) должна быть повторена несколько раз. Число повторов циклов может быть определено несколькими способами, и поэтому существует три типа циклов.

1. Цикл с предусловием. При выполнении этого цикла в первую очередь проверяется условие производительности и повторяется тело цикла. Если условие не будет выполнено во время первого цикла, то тело

второго цикла не будет выполнено вообще. После выхода из контура управление должно быть передано в следующую структуру.

2. При выполнении этого цикла состояние завершения цикла должно быть проверено после того, как тело было зациклено. Вот почему тело цикла всегда выполняется в программе как минимум один раз. До тех пор, пока условие не будет выполнено, тело цикла должно быть повторено (конец цикла является условием).
3. Цикл с параметром. Этот тип цикла подходит для использования, когда количество повторений цикла известно заранее. Представлена концепция счетчика циклов, используемого для организации этого цикла. Чтобы изменить счетчик циклов, необходимо установить верхний и нижний пределы. В зависимости от значений верхнего и нижнего пределов определяется уровень (1 или – 1), т. е. значение счетчика цикла.

Алгоритм, используемый для запуска программы на компьютере, должен быть написан на языке, который он понимает. В этом случае язык, используемый для написания алгоритмов, должен быть формализован. Этот язык обычно называют языком программного обеспечения, и запись на этом языке называется программой.

Разработка алгоритмов решения компьютерных задач требует определенных навыков. Когда необходимо создать алгоритм, написание должно быть ясны и наглядны. При изменении алгоритма рекомендуется не завершать его. Эти требования могут быть выполнены, если принять во внимание структурный подход. Наиболее важным преимуществом структурного подхода является возможность перемещать программы сверху вниз, что позволяет перейти от основной задачи к более мелким задачам. Большая задача будет разделена на маленькие блоки, эти блоки так же разделены на более мелкие блоки и так далее. Каждый блок алгоритма должен быть максимально независимым и логически завершенным. Разделение на блоки должно осуществляться с учетом внутренней логики

задачи. Процесс деления завершается, после того как решение исходной задачи сводится к решению нескольких простых задач, для которых легко разработать соответствующие алгоритмы. На каждом этапе этой процедуры происходит детальный переход от общих задач, в частности тех, которые детализированы в виде конкретного подправила. Такой метод называется пошаговой детализацией. Таким образом, используя структурный подход, имеется возможность комбинировать не только лишь базовые структуры (следование, разветвленные и цикл), но и подключаться к ранее написанным алгоритмам. Алгоритмы, которые полностью используются как часть других алгоритмов, называются вспомогательными. Если вспомогательный алгоритм выполняется более одного раза в течение программы, вспомогательный алгоритм обычно используется в качестве подпрограммы.

«Процесс восприятия» компьютерной программой, написанной на формальном (высоком) языке, называется трансляцией. Программа должна быть собрана и интерпретирована во время выполнения. Перевод - это перевод программы, написанной на высоком языке происхождения, в эквивалентную программу на более низком языке, близком к машинному коду. Интерпретационная программа-покомандный анализ, обработка и мгновенное выполнения исходной программы (в отличие от компиляции, при которой программа транслируется без ее выполнения).

Выводы по первой главе.

Изучение основ робототехники имеет перспективность и важность, т.к. в настоящее момент руководством нашей страны было четко сформулировало социальный заказ в сфере образования: в нашей стране на данный момент имеется нехватка инженеров.

Для того, что бы исправить данную проблему нужно активно начинать популяризацию профессии инженера уже в средней школе. Детям необходимы образы для подражания в области инженерной деятельности, чтобы «разбудить» в них интерес и дать возможность ощутить важность и необычность в работе инженера. Исходя из этого, остро стоит вопрос выбора программного обеспечения для тех или иных образовательных целей.

Для реализации занятий по робототехнике одним из главных вопросов при реализации методики является верный подбор конструктора, а также среды разработки, которые позволят решать современные образовательные задачи. На данный момент имеется большое количество сред разработки, которые используются на уроках «Основы робототехники». Каждая среда разработки по своему сложная и подходит для решения определённых учебных задач, в зависимости от того, какие именно задачи стоят перед учащимися. Исходя из этого, перед педагогами стоит задача выбрать подходящую среду разработки для изучения учащимся, научить детей сначала основам программирования, переходить к более сложным системам программирования.

Решением данной задачи является разработка методических рекомендация по изучению среды разработки, которая позволит развивать навыки программирования роботом с учётом их возраста и полученных знаний на других уроках. Данная преемственность изучения позволит школьникам на основе существующих знаний программирования осваивать более сложные алгоритмы.

ГЛАВА 2. ПРОГРАММИРОВАНИЕ РОБОТОВ НА УРОКАХ ТЕХНОЛОГИИ.

2.1. Обзор программного обеспечения для образовательной робототехники.

Одним из основных вопросов при реализации методики обучения робототехнике является выбор конструктора, а также среда разработки, позволяющая решать современные образовательные задачи.

Для программирования роботов используются как графические, так и текстовые языки программирования. К графическим относятся такие языки как LegoWedo, NXT-G, LegoMindstorms EV3, LabViewEducationEdition, к текстовым – RobotC, ArduinoIDE, и др. Графические языки программирования предполагают 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.

Wedo 2.0

Wedo 2.0 – в качестве основного языка программирования для платформ WeDo2.0 (Рис. 22) используется графическая нотация LabVIEW от компании NationalInstruments. В отличии от платформы EV3, в WeDo 2.0 этот язык сильно упрощен в соответствии с возрастом обучающихся, работающих с WeDo 2.0.

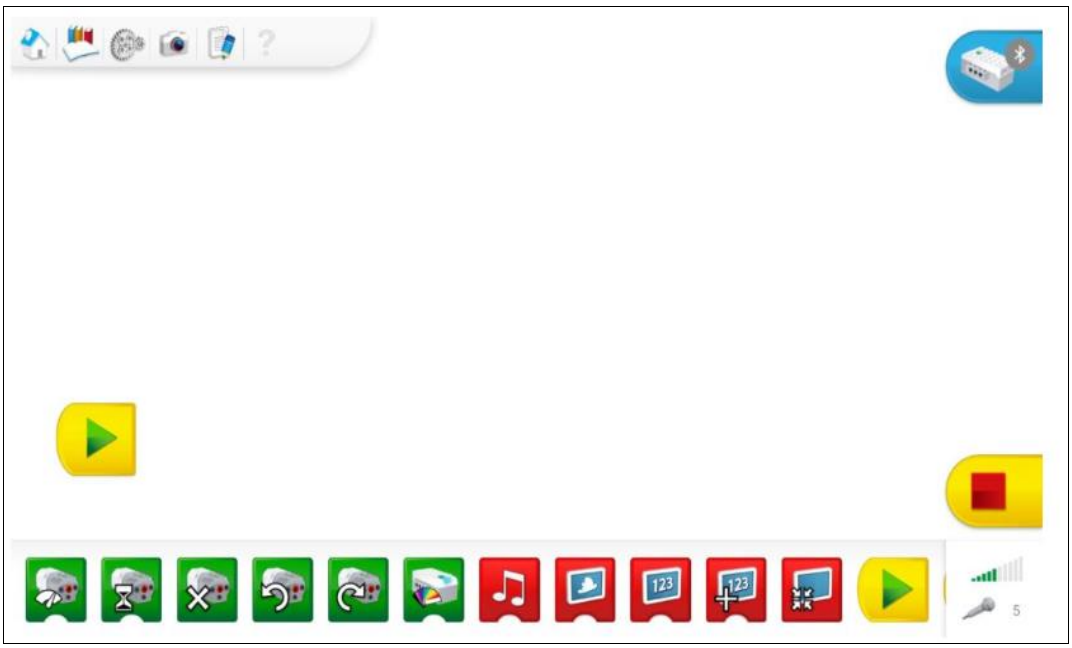

Рис.23. Общий вид Wedo 2.0.

Как уже упоминалось, программная платформа WeDo 2.0 основана на графическом языке программирования "G", заимствованном из среды программирования LabView. Но если исходные блоки соединены проводами (по аналогии с проводами в физическом мире), то для этой среды был выбран более простой подход. Блоки соединены между собой по принципу "вагонов" в поезде – один за другим, а расширители блоков будут иметь пазлообразный вид и даже ребенок интуитивно поймет, что и где нужно соединять.

Ещё одна идея разработчика также помогает "новичку" привыкнуть к ней в кратчайшие сроки. Это разделение программных блоков по цветовой палитре:

- Зеленная политра - блок регулирования двигателем и интеллектуальный индикатор смартхаба.
- Красная палитра - блок работы с дисплеем, аудио и арифметикой.
- Желтая политра - блок программного обеспечения (пуск, режим ожидания, повтор серий).
- Оранжевая палитра - блок работы с датчиками.
- Синяя палитра - расширение блоков.

Все блоки регулирования двигателем и индикатором смартхаба (Рис. 24) имеют визуальную подсказку – на них нарисован мотор или смартхаб – поэтому сразу понятно каким элементом мы будем управлять, добавив этот блок в программу.

Рис.24. Блок управления мотором и индикатором смартхаба.

Первый блок с символом, похожим на спидометр, задает мощность (скорость вращения) двигателя. На практике чаще всего используют оба понятия, как взаимозаменяемые. Детям нравится, что в основных настройках можно установить значение мощности в десятки тысяч, но это не имеет смысла, потому что программное ограничение устанавливается на отметку "10", и все значения, превышающие эту отметку, воспринимаются как "10".

Блок с песочными часами задает время работы мотора. Единица измерения времени – секунда.

Следующие два блока отвечают за задание направления вращения оси, подключенной к мотору – по часовой стрелке или против.

Блок с крестом отвечает за остановку двигателя. Остановить двигатель можно несколькими способами: установив блок питания с уставкой «0» в нужном месте алгоритма или полностью остановив программу.

И последний блок в палитре отвечает за изменение цвета свечения индикатора смартхаба. Причём эту функцию можно использовать как в мультимедийных целях (реализация светофора), так и для отладки алгоритма и установки в «контрольных точках» программы.

Красная палитра, в первую очередь, включает в себя блоки управления экраном:

Рис.25. Блок работы с экраном, звуками и математикой.

Блок экрана с облаком (Рис. 25) позволяет задать фон экрану из встроенной библиотеки изображений, которая содержит 28 доступных картинок различных категорий: природа (горы, океан).

Блок экрана с цифрами «123» позволяет работать с текстовыми и числовыми данными. При добавлении блока расширения «abc» мы переходим в режим вывода текстовых сообщений – на экране отобразится информация для пользователя, введённая в блок расширения. В случае добавления блока расширения «123» (цифры на белом фоне) активируется режим работы с числами. При этом введённое значение не только отображается на экране, но еще и запоминается в памяти экрана. Последнее записанное значение хранится в блоке расширений «123» (полностью синий блок). Таким образом, получается аналог переменной из классического программирования.

Блок математики выполняет привычную для него роль – складывает, вычитает, умножает и делит. Отлично подходит для реализации таймеров и счетчиков, инверсии сигналов от датчиков.

Последний блок отвечает за размер экрана – его можно развернуть на всю рабочую зону программы, уменьшить, либо свернуть.

Блок с изображенной нотой – блок звуковых эффектов. Настоящая боль любого преподавателя робототехники, поскольку дети, узнав про этот блок, стараются установить его в каждую свою программу. При этом регулятор громкости выкручивается в максимум. У блока имеется встроенная библиотека разнообразных звуков, а также функция записи своего звукового файла.

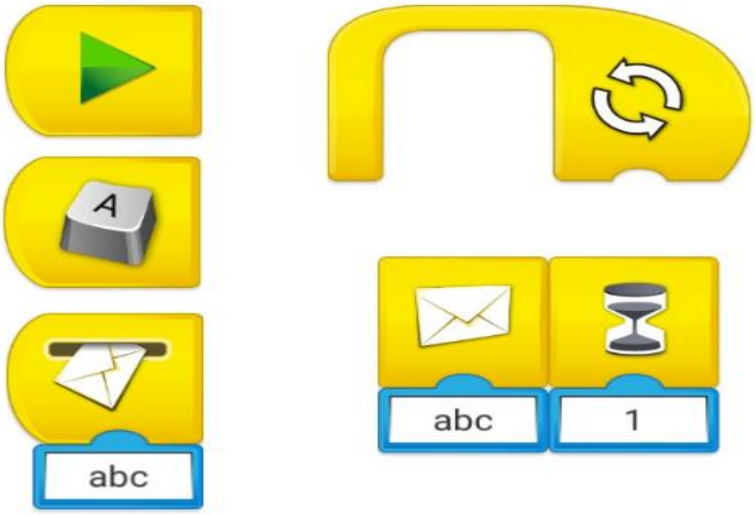

Рис.26. Блоки управления программой (запуск, ожидание, цикл).

У любой программы должна быть кнопка её запуска (Рис. 26) – за эту функцию в WeDo 2.0 отвечает сразу несколько блоков:

Блок запуска с символом «Play» появляется сразу в рабочей области программы, как бы призывая сразу написать свой первый алгоритм управления собранной моделью.

Следующий блок, которым можно запустить выполнение клавиатуры – это блок «Клавиша» — по умолчанию установлена клавиша A, но можно выбрать любую другую клавишу как на латинице, так и на кириллице. Изменить клавишу возможно после клика на блок, удержания его в «нажатом состоянии» – блок перейдет в режим изменения параметров – выбирайте любой понравившийся символ с клавиатуры.

Ещё один блок, который может быть стартовым – это блок «Получение сообщения» (работает в связке с блоком «Отправка сообщения»). Данный блок используется для перехода из одной ветки алгоритма в другую при достижении заданных параметров. Например, в основном алгоритме у вас выполняется программа, в которой содержится блок отправки сообщения «Stop». Вы устанавливается в рамках подпрограммы блок получения сообщения с аргументом «Stop» – и выполняете требуемую последовательность действий параллельно с выполнением основного кода.

Этот блок достаточно часто применяется при опросе датчиков в режиме реального времени – под конкретное значение датчика пишется своя подпрограмма с аргументом, соответствующим этим числовым значениям.

Один из самых часто-используемых блоков на уроках робототехники– это «Цикл» – функционал у этого блока тот же, что и в классическом программирование – повторять программу или её часть определенное число раз, по наступлению какого-либо события или же бесконечно. По умолчанию блок «Цикл» работает в режиме бесконечного, для того чтобы задать ему ограничение по числу выполнений достаточно подключить блок расширения (например, числовой блок или датчик расстояния).

Последний в обзоре блоков управления, но в то же время один их самых важных при написании программ – блок «Ожидание». По умолчанию это таймер, который останавливает выполнение программы на время, заданное в блоке расширения (отчёт ведётся в секундах). Расширить функционал блока можно, подключив к нему блоки расширения из оранжевой и синей палитр. Например, при добавлении датчика расстояния блок «Ожидания» останавливает программу до момента, пока не произойдет срабатывание датчика. То же самое справедливо для блоков расширения датчика наклона и микрофона. Таким образом, блок «Ожидание» — это основной блок программ, которые подразумевают реагирование роботов на события внешнего мира – наклон, изменение расстояния, увеличение шума и др.

Одно из основных отличий образовательного конструктора LegoWeDo 2.0 от обычного конструктора – это наличие датчиков, позволяющих роботам взаимодействовать с окружающим миром. Появление препятствий, удаление объектов, изменение наклона плоскости или управление джойстиком – все эти события внешнего мира нужно уметь понимать на программном уровне.

Для этого в среде программирования WeDo 2.0 предусмотрены блоки расширения (Рис.27), которые считывают информацию с датчиков.

Датчик расстояния может работать в трех режимах:

- Объект приближается (блок расширения со стрелкой, указывающей на датчик)
- Объект отдаляется (блок расширения со стрелкой, указывающей от датчика)
- Объект изменяет свое положение (блок расширения со стрелкой, указывающей в обе стороны)

Рис.27. Блок работы с датчиками.

Также имеется блок расширения без стрелок, изображающий датчик расстояния – он используется в случаях, когда требуется получить числовое значение датчика в конкретный момент времени.

Датчик расстояния считывает расстояние по шкале от 0 до 10 условных единиц, максимальная граница соответствует 15-18 сантиметрам.

В свою очередь датчик наклона считывает наклон в двух плоскостях, при этом разработчик закодировал каждое положение соответствующей цифрой:

- Наклон носом вверх (к себе)
- Наклон носом вниз (от себя)
- Наклон влево
- Наклон вправо
- Отсутствие наклона (датчик расположен горизонтально) «0»
- Наклон в любую сторону (режим «тряска»)

Ещё одна группа блоков представляет собой блоки расширения (Рис. 28). Их цветовая гамма — синяя.

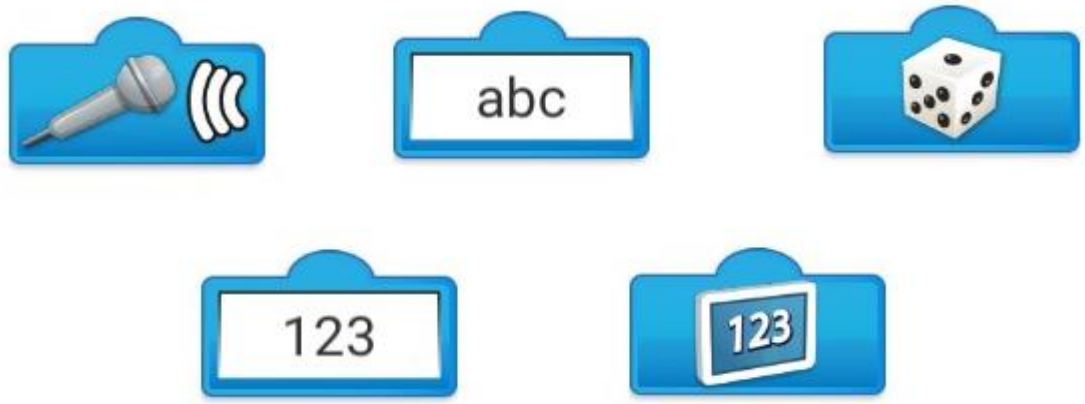

Рис.28. Блоки расширения.

1. Блок с изображением микрофона является простейшим датчиком звука. Если этот блок расширения добавить к блоку ожидания («Песочные часы»), то программа будет ожидать увеличения громкости звука — это может быть, например, хлопок.
2. Блок с буквенными символами «abc» является блоком ввода текстовых данных. Подключается как правило к блокам «Экран» и «Отправка/получение сообщения».
3. Блок с символом игральной кости — это генератор случайных чисел от 0 до 10. Возможно подключение ко всем блокам, которые имеют «разъём» расширения.
4. Блок с числовыми символами «123» является блоком ввода числовых данных. Используется в случаях, когда нужно определённому блоку присвоить некое значение, например, задать мощность на уровне «6».
5. Блок с символом экрана «123» — хранит текущее значение, которое записано в память блока экрана с цифрами «123». По сути своей этот блок является переменной в чистом виде.

Lego Mindstorms EV3

Простое в освоении и использовании мульти платформенное образовательное программное обеспечение EV3 создано специально для применения в учебной деятельности. ПО делает возможным программировать модели роботов, созданные обучающими, используя

графический язык программирования LabVIEW, в котором программа основана на разработке программных блоков, которая является наиболее важной частью программы. Помимо этого, программное обеспечение для образовательной платформы EV3 обладает целым рядом дополнительных функций, призванным обеспечить использование EV3 в широком спектре предметных областей.

Рис.29. Среда программирования LEGO MINDSTORMS Education EV3.

Среда программирования EV3 (рис. 29) состоит из следующих основных областей:

1. Область программирования – бласть размещение своей программы.

2. Поллитры - здесь размещены модули для программы.

3. Аппаратная часть - здесь вы можно связаться и контролировать модуль EV3 и посмотреть, как подключены двигатели и датчики. Также можно скачать программы для EV3 Brick здесь.

4. Content Editor - электронная книга, установленная в программном обеспечении. Получить инструкции или документировать свой проект с текстом, изображениями и видео.

5. Окно программирования - здесь можно найти основные инструменты для работы с программой. Для получения дополнительной информации об этих инструментах см. меню справки программного обеспечения EV3.

Все программные блоки, используемые для управления роботом, расположены в палитрах программного обеспечения в программной среде в области программирования. Программа разбита на категории в зависимости от типа и типа блоков, что упрощает поиск нужного блока.

Если вы заинтересованы, узнайте и поймете больше о среде программирования EV3, как начать работу с первой программой, то можно посмотреть видео "Программирование" и "Общая информация о программировании", которое находится в разделе "Первые шаги" на главной странице при запуске программы.

Для получения дополнительной информации о программировании с помощью EV3 разработчики программного обеспечения рекомендуют перейти в меню "Справка".

Блоки действия (изображены слева направо на Рис. 30) отображаются зеленым цветом. На данной палитре расположены программные блоки управления моторами, блок вывода на экран, блок управления индикатором состояния модуля.

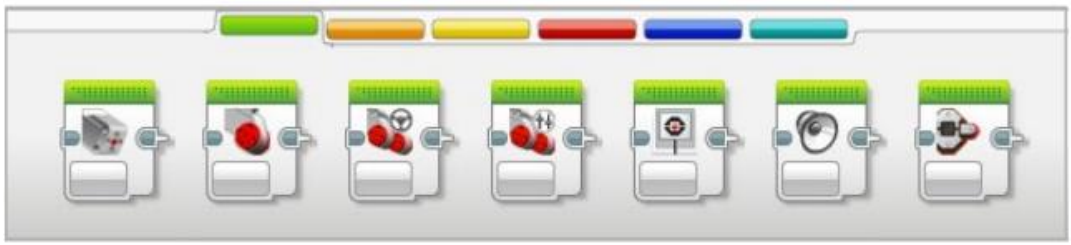

Рис.30. Блоки действия.

- Средний двигатель
- Большой двигатель
- Рулевое управление
- Независимое управление двигателями
- Дисплей
- Аудио
- Индикатор состояния модуля

Блоки-операторы (изображены слева направо на Рис. 31), отображаются оранжевым цветом, Этот блок в зависимости от настроек выбирает для выполнения программные блоки, расположенные в одном из своих контейнеров.

Рис.31. Блоки-операторы.

- Начать
- Ожидать
- Повтор
- Переключатель
- Конец повтора

Блоки датчиков (изображены слева направо на Рис. 32), отображаются желтым цветом, помогают получить информацию с датчиков и способствуют реагированию на эту информацию в программе.

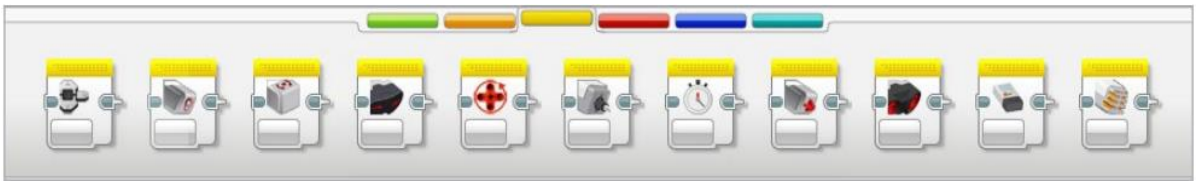

Рис.32. Блоки датчиков.

- Кнопки управления модулем
- Прибор измерения цвета
- Гироскопический измеритель
- Инфракрасный измеритель
- Обороты двигателя
- Прибор измерения температуры
- Время(таймер)

- Сенсорный
- Ультразвуковой датчик
- Электрический счетчик
- NXT датчик звука

Блоки данных (изображены слева направо на Рис. 33), отображаются красным цветом. На данной палитре расположены программные блоки, необходимые для выполнения различных операций над числовыми, логическими или текстовыми данными.

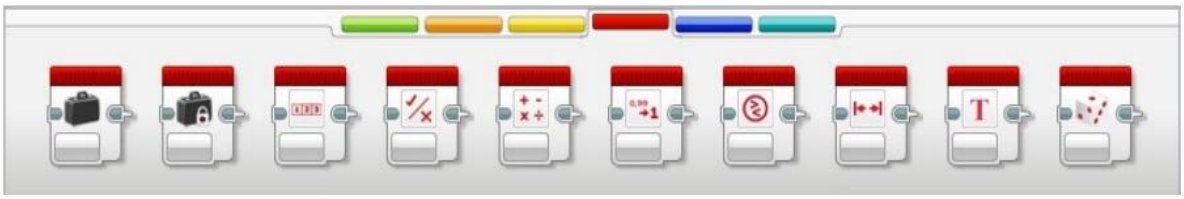

Рис.33. Модуль данных.

- Переменная
- Константа
- Операции над массивом
- Логические операции
- Математика
- Округление
- Сравнение
- Интервал
- Текст
- Случайное значение

Расширенные модули (изображены слева направо на Рис. 34), отображаются синим цветом.

Рис.34. Расширенные модули.

- Доступ к файлу

- Регистрация данных
- Обмен сообщениями
- Подключение Bluetooth
- Поддерживать в активном состоянии
- Необработанное значение датчика
- Нерегулируемый мотор
- Инвертировать вращение мотора
- Остановить программу
- Комментарий

Мои модули. Если многократно использовать один и тот же сегмент программы во многих программах, имеет смысл создать Мой Блок. Создав Мой блок, можно просто вставлять этот единый блок в будущие программы этого проекта. Отображаются бирюзовым цветом (Рис. 35).

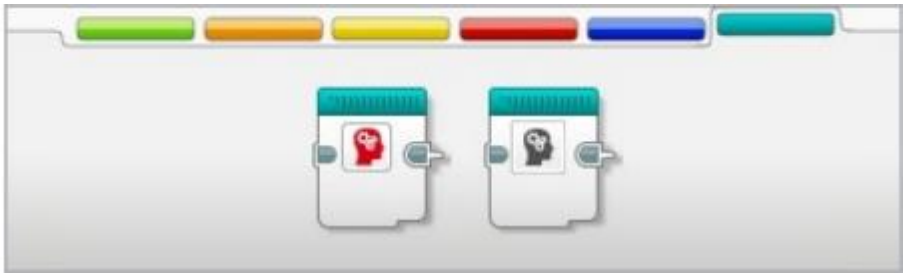

Рис.35. Мои блоки.

На специальной странице с подключенным устройством вы можете отслеживать состояние и вызывать значения с датчиков в режиме реального времени.

Благодаря функции автоматической идентификации устройство автоматически распознается при подключении (автоматическое определение устройства). Т.е. необходимости указывать, подключен ли такой датчик или двигатель к такому соединению.

Выделение блоков при выполнении программы позволяет определить, где в данный момент выполняется условие.

Если датчик или двигатель с соответствующим подключением используется неправильно, специальный символ отображается в

соответствующем программном блоке. Он снова доступен с функцией автоматической идентификации.

Вы можете отобразить значения, передаваемые по каналам данных (кабель для передачи данных).

С точки зрения программирования можно кратко отметить следующие нововведения:

- Интенсивная интеграция между p-модулем (новое имя вместо блока Nxt) и средой программирования;
- Вы можете отслеживать состояние на специальной странице с подключенными устройствами и вызывать значения в режиме реального времени с помощью датчиков.
- Благодаря функции Auto-ID устройство автоматически распознается при подключении. Это означает, что вам не нужно указывать, какой порт подключен к определенному датчику или двигателю.
- Новый режим отладки:
- Освещение места выполнения алгоритма определяет, какая именно программа запущена.
- Если датчик или двигатель неправильно используются с соответствующим соединением, специальный символ отображается в соответствующем программном блоке. Это также достигается с помощью функции Auto-ID;
- Может видеть значения, переданные через Datavires);
- Новые функции программных блоков:
- Блокируя блоки, блоки могут отбрасывать «исполнительный луч», расположенный в среде Nxt-g.
- Нет ничего похожего на поле настройки поведения блока, которое теперь настраивается непосредственно в блоке, что приводит к увеличению. Программное обеспечение для юзабилити теперь легче читать, поэтому вы можете сразу увидеть, как реагирует на прикосновения или двигатель.

- Есть блоки «ожидание изменений», с помощью которых вы не можете изменить определенное значение, а просто реагировать на изменения.
- Вы можете работать с сериями;
- Ранний доступ к циклу стало возможным.

Arduino IDE

Среда разработки Arduino IDE (рис. 36) предлагает код текстового редактора, окно сообщения, окно вывода текста (консоль), панель инструментов и несколько меню. Среда разработки для прямой загрузки программ и коммуникации добавлена во флот Arduino.

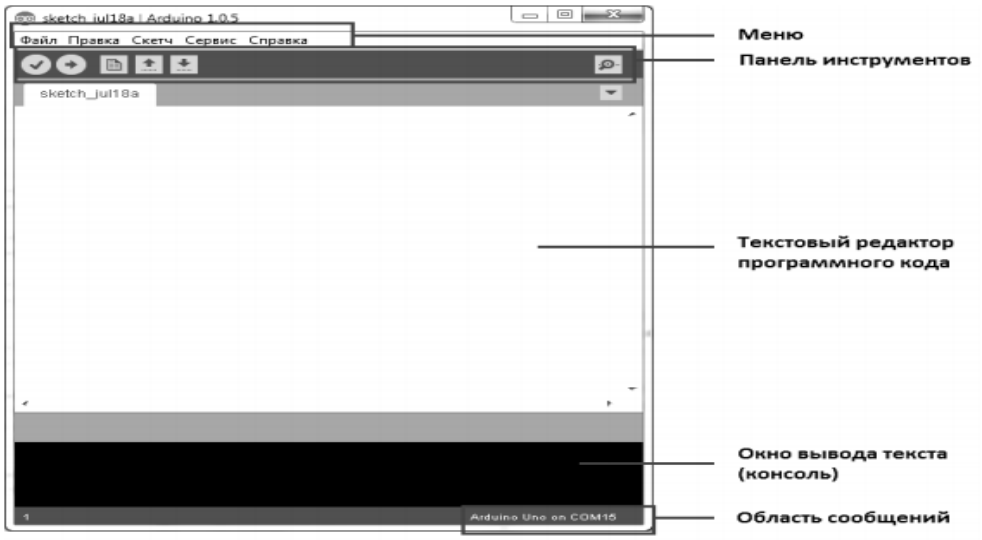

Рис.36. Общий вид Arduino IDE.

Меню "Файл" (Рис. 37). Меню включает в себя разнообразные вкладки по работе с файлом. Элемент Thumb Folder: по умолчанию среда IDE Arduino хранит каждый эскиз в отдельной папке. При сохранении имени папки оно совпадает с именем, указанным для миниатюры. Для небольших папок вы можете изменить рабочую папку через меню настроек.

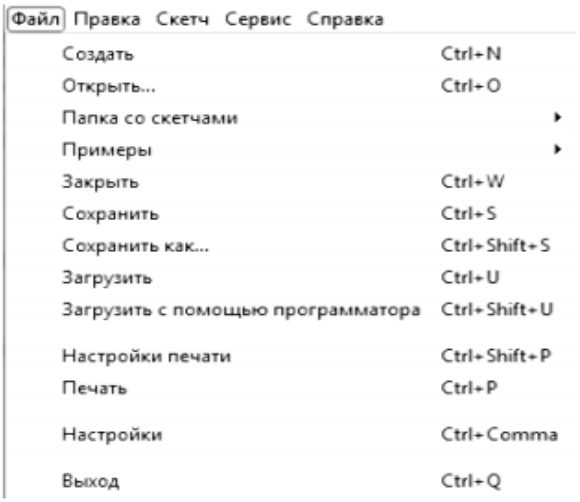

Рис.37. Меню «Файл».

Меню «Правка» (рис. 38) организовано в соответствии с командами для работы с кодом программы. Команды, которые часто используются, удобны тем, что есть комбинации для быстрого доступа к клавиатуре. Практические функции предоставляют возможность копировать форум и код HTML в формате, который позволяет обмениваться миниатюрами, в то время как форматирование, такое как коды BB или HTML, четко отформатировано.

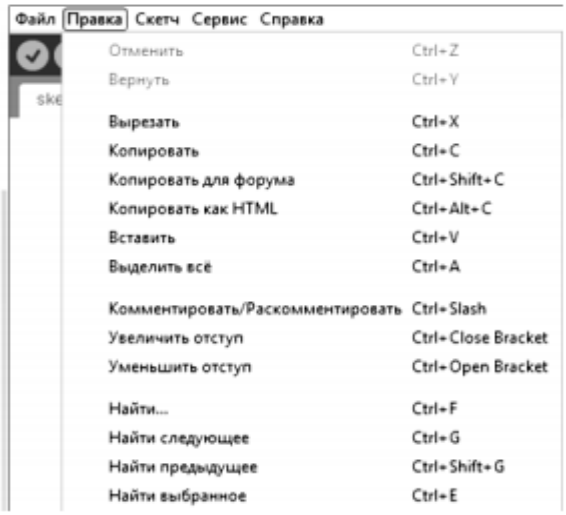

Рис.38. Меню «Правка».

Небольшое меню «Скетч» (Рис. 39). Команда в этом контекстном меню дублируется с помощью панели управления «Проверка / Компиляция». Выполнение этой команды проверяет код на наличие ошибок и компилирует приложение, если оно не существует.

Пункт меню «Показать небольшую папку «Скетчей»» открывает рабочую папку Arduino IDE, указанную в настройках.

«Как добавить файл ...» - это текстовый файл (или небольшой файл), который можно использовать для открытия разработки в отдельной среде с вкладками.

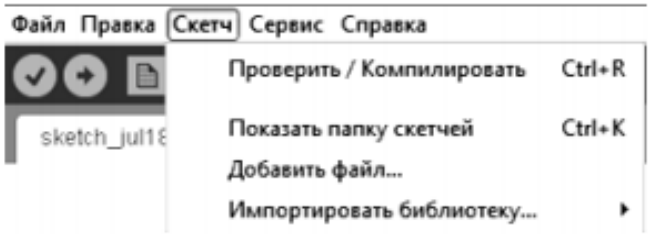

Рис.39. Меню «Скетч».

Пункт меню «Импорт библиотеки»: содержит множество предустановленных библиотек с Arduino IDE. Библиотеки предлагают дополнительные функции при работе с устройствами или обработке данных, например, Б. Миниатюры. Инструкции из одного или нескольких символов #include помещаются в начале кода эскиза, что сопровождается компиляцией библиотек и эскиза. Загрузка библиотек требует дополнительного места в Arduino. Чтобы загрузить другие библиотеки, вы можете использовать команду «Импорт библиотеки» ...

В меню «Сервис» (фото) 40) должна быть указана модель используемой материнской платы Arduino и подключенного COM-порта.

Практическая функция - функция автоматического форматирования, с помощью которой вы можете исправлять ошибки при написании эскизов и делать их читабельными.Особенно актуальной эта функция является при копировании сторонних программ в среду разработки AdruinoIDE.

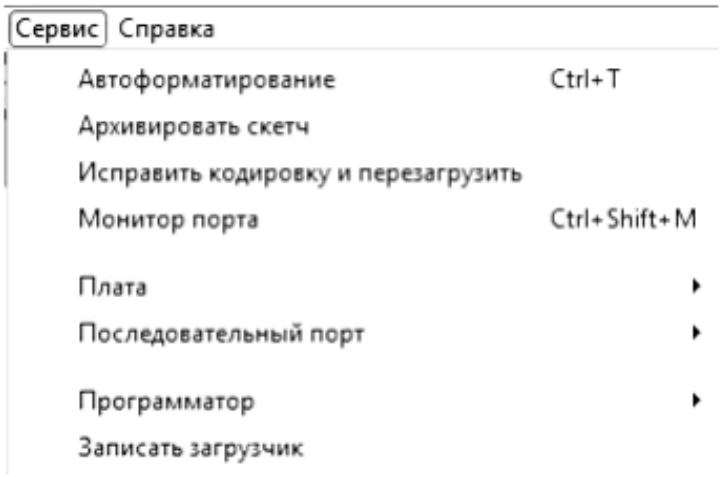

Рис.40. Меню «Сервис».

Пункт меню «Монитор порта» (Рис. 41) вызывает окно для обмена сообщениями с arduino через COM порт:

Рис.41. Пункт меню «Монитор порта».

Команды панели управления (Рис. 42), дублируют наиболее актуальные пункты меню, которые часто используются при использовании в среде разработки.

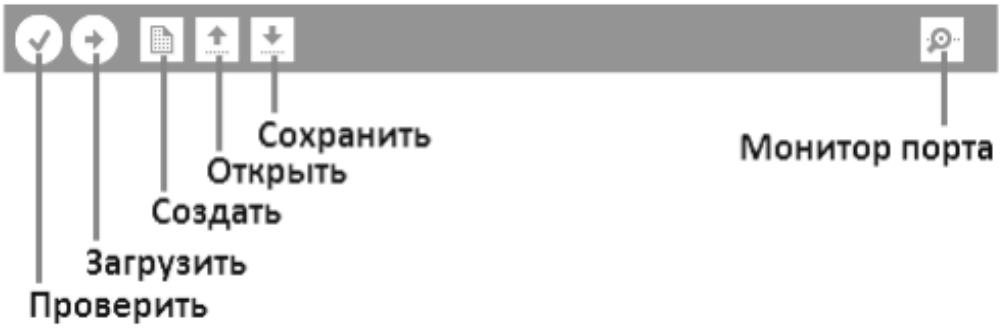

Рис.42. Команды панели управления.

LabVIEW

На сегодняшний день наиболее популярной становится среда для программирования LabVIEW (Рис. 43) (Среда разработки виртуальных приборов). Эта среда программирования используется во многих отраслях промышленности, включая космическую. Школьная версия графической среды программирования LabVIEW Education Edition представляет собой большие возможности для проектной и исследовательской деятельности преподавателей и учащихся, поскольку представляет собой достаточно простую и интуитивно понятную системупри её использовании.

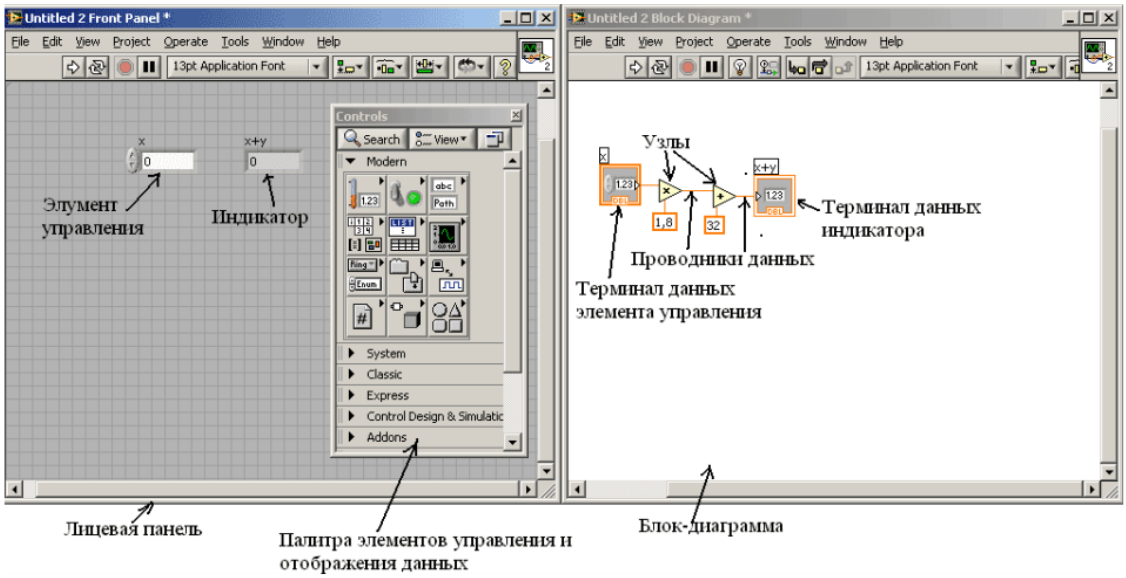

Рис.43. Графическая среда программирования LabVIEW

Программы создаются в графической среде программирования в двух окнах в лабиринте. Это приложение называется:

- Передняя (передняя панель);
- Блок-схема.

Панель управления улучшает внешний вид будущего виртуального устройства (все программы, разработанные Labvi, называются VirtualInstruments), которое создает элементы управления и отображения из интерактивных инструментов ввода и вывода для этого виртуального инструмента.

Кнопки управления и другие устройства ввода.

Элементы дисплея (дисплеи) - графика, светодиоды и др. дисплеи.

Элементы управления управляют модельными устройствами ввода данных и передают данные на блок-схему виртуальных устройств. Элементы экрана имитируют вывод и показывают устройства, которые получают или генерируют потоковую диаграмму.

На блок-схеме приведен исходный код будущего виртуального устройства. В отличие от классических языков программирования высокого уровня, таких как C, Pascal, Fortran, исходный код в Labv представляет собой блок-схему, в которой все команды, операторы цикла и сравнения представлены графическими символами. Блок-схема состоит из узлов, терминалов и проводников данных.

Узлы - это объекты в блок-схеме, которые имеют одно или несколько полей ввода / вывода данных и выполняют алгоритмические операции на виртуальном устройстве. Они похожи на операторы, функции и процедуры текстовых языков программирования. Узлы включают функции, подпрограммы (SubVI) и структуры. Подпрограмма - это виртуальное устройство, которое можно использовать в качестве подпрограммы в другой блок-схеме VI. Структуры Плоская структура, корпус, петля состояния (белая) и т. Д. Являются такими элементами управления процессом.

На блок-схеме объекты отображаются спереди как терминалы данных. Соединения для передачи данных обеспечивают обмен данными между передней панелью и блок-схемой. Существуют следующие типы терминалов данных: терминалы управления и отображения, терминалы узлов. Терминалы управления и индикации подключены к управлению и отображению данных на передней панели. Информация, которая поступает на органы управления на передней панели, вводится в блок-схему через клеммы управления.

В программе данные передаются между объектами схемы блокчейна по соединительным линиям (проводам). Data Explorer похож на переменные в текстовых языках программирования. Каждый проводник данных имеет источник данных, но многие функции могут передавать его. Потоки данных

различаются по цвету, стилю и ширине линии в зависимости от того, какой тип информации передается из одной блок-схемы в другую.

В среде Labvi объекты подключаются к проводам данных после их вставки в потоковую диаграмму.

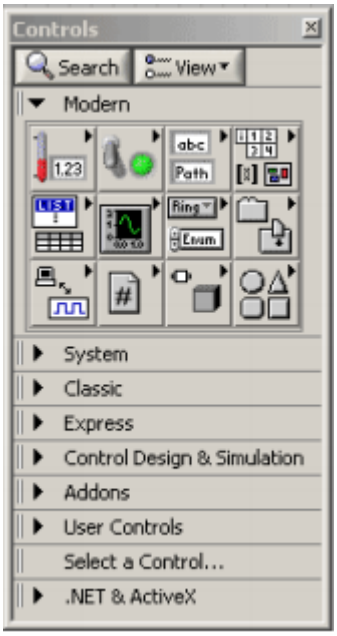

Рис.44. Палитра элементов управления и отображения.

Панель управления и экран (изображение) используются для размещения элементов управления и отображения информации на панели управления. 44). Элементы управления и палитра доступны только спереди. Чтобы отобразить палитру на экране, щелкните правой кнопкой мыши рабочую область спереди.

Все элементы управления и изображения в палитре разделены на следующие разделы:

- Современные элементы управления и дисплеи имеют современный стиль.
- Панель управления и отображение определенной операционной системы (кнопок, ползунков и т. д.). Имеет соответствующий стиль.);
- Классическое управление и классический экран;
- Экспресс общее управление и отображение;

Каждый раздел разделен на шесть частей (числовой, линейный дисплей, кнопки и т. Д.).

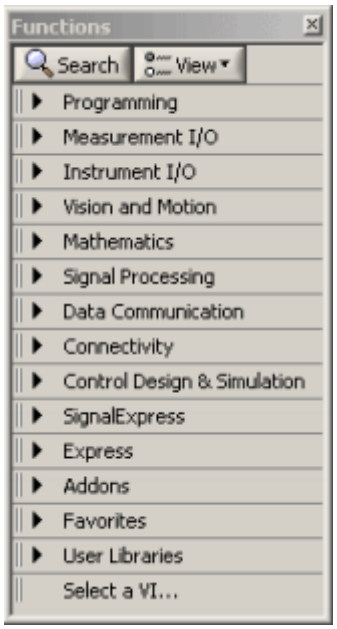

Рис.45. Палитра функций.

Как уже упоминалось в блокчейне, исходный код программы разрабатывается. В программировании используется ряд функций (функция паллета, фото). 45). Чтобы отобразить палитру на экране, щелкните правой кнопкой мыши рабочую область потоковой диаграммы.

Все элементы палитры разделены на следующие разделы:

- Программирование - Здесь вы найдете все основные функции, циклические структуры, сравнения, дополнительные операторы и вычитания, необходимые для создания большинства простых приложений, которые были изучены в учебном году.
- Здесь собрана утилита ввода / вывода для работы с различными устройствами ввода / вывода (COM-порт и т. Д.)
- Функции математической оси для математических расчетов (решение системы уравнений, решение дифференциальных уравнений и т. Д.). Сборник.);
- Здесь собраны функции обработки сигналов, связанные с цифровой обработкой и анализом дискретных сигналов (цифровые фильтры, быстрое преобразование Фурье и т. Д.);

- Выражать общие характеристики, связанные с обработкой данных и обработкой данных;
- Выберите VI ... - Пользовательские функции.

Рис.46. Палитра инструментов.

С помощью палитры инструментов (рисунок) вы можете создавать, изменять и отлаживать виртуальные инструменты. 46). Палитра инструментов доступна как спереди, так и на блок-схеме. Термин инструмент означает специальную функцию указателя мыши. Когда вы выбираете инструмент из списка, значок курсора меняется на значок выбранного инструмента. Панель инструментов вызывается при одновременном нажатии кнопки и правой кнопки мыши. Палитра инструментов, блок-схема и передняя панель могут быть размещены в любой области.

Если функция автоматического выбора инструмента активна и вы наводите указатель мыши на объект лицевой панели или блок-схему Labvi, соответствующий инструмент автоматически выбирается из палитры инструментов. Опция автоматического инструмента активируется нажатием кнопки Palitra на инструменте автоматического выбора инструмента (зеленый - вкл. Палитра отображается в виде прямоугольника в верхнем правом углу). Автоматический выбор инструмента Black-Off.) Или нажатием Shift + Tab.

Инструмент управления (называемый пальцем) используется для управления или изменения значения ввода текста. При наведении указателя мыши на указанный элемент управления в виде строки значок инструмента меняется.

Инструмент движения (обозначенный стрелкой) используется для выбора, перемещения или изменения размера объектов. Указатель мыши изменяется, когда вы наводите курсор мыши на объект со сменным символом инструмента.

Инструмент для ввода текста (называемый буквой «A») используется для редактирования текста и создания свободных тегов. Символ пишущего инструмента свободно изменяется при создании.

Инструмент контактов (показанный в виде катушки) создает проводники данных, которые соединяют объекты в блок-схеме.

Инструмент вызова контекстного меню (отображается как меню) вызывает контекстное меню соответствующего объекта, щелкая левой кнопкой мыши.

Инструмент быстрой прокрутки (называемый рукой) используется для отображения окна без использования ползунка.

С помощью инструмента ввода контрольных точек (отображается как контрольная точка) вы можете размещать контрольные точки в виртуальных инструментах, функциях, узлах, направлениях данных и структурах и прекращать выполнение программ в них.

Вы можете использовать инструмент набора индикаторов отладки (показанный в качестве образца), чтобы проверить поток данных в потоковых диаграммах. Эти виртуальные инструменты используются для отображения промежуточных значений при наличии подозрительных или неожиданных результатов.

Средство для удаления краски (показанное как капельница) используется для удаления краски, а затем склеивает ее вместе с краской (кистью).

Используйте инструмент рисования (называемый кистью), чтобы изменить цвет объекта. Текущие настройки для переднего цвета и фона также отображаются.

Если автоматический выбор инструмента деактивирован, вы можете изменить инструменты палитры инструментов с помощью клавиши табуляции. Пробел - Вам нужно нажать клавишу на клавиатуре, которая отвечает за перемещение и переключение между диаграммами блок-схем или перемещение передней панели и подключение инструмента управления.

Несмотря на все свои различия, все эти среды разработки имеют много общего и в основном эквивалентны с точки зрения потенциала для написания определенной программы в любой из этих сред.

2.2. Плюсы и минусы сред разработки на уроках технологии курса робототехники.

Среда разработки Wedo 2.0

Достоинства:

- Поддерживает все доступные платформы для установки программного обеспечения;
- Бесплатное программное обеспечение;
- Встроенная интерактивная версия для преподавателей;
- Связывается с устройством посредством Bluetooth версии 4.0, соединение стало беспроводным;
- Большое количество встроенных проектов для начала изучения конструктора;
- Встроен редактор отчёта.

Недостатки:

- Нестабильная беспроводная связь;
- Не большое количество блоков для программирования;
- Нет возможности повторно запустить программу при помощи смартхаба;

- Нет совместимости с другими средами графического программирования.

Среда разработки Lego Mindstorms EV3

Достоинства:

- Система является бесплатной;
- Интуитивно понятный и удобный интерфейс;
- Возможность интеграции модуля машинного зрения;
- Мощная методическая поддержка;
- Возможность реализации сложных алгоритмов.

Недостатки:

- Отсутствие возможности отладить программу на симуляторе;
- Громоздкость диаграмм при создании больших и сложных программ;
- При написании сложных и разветвленных алгоритмов программное обеспечение сильно грузит систему;
- Поддерживает не все операционные системы.

Среда разработки Arduino IDE

Достоинства:

- Подсветка кода
- Быстрая заливка скетча в плату ардуино;
- Для того, что бы обозначить определённый пин порта как вход выход достаточно написать функцию pinMode (Имя порта, OUTPUT/INPUT);
- Можно быстро присвоить номер порта к определённой переменной;
- Быстрое обозначение состояние порта (LOW или HIGH);
- Имеются стандартные библиотеки, которые могут помочь разработать программу за короткий промежуток времени;
- Открытый исходный код программы. Кодирование программы может расширяться на платформе C++;
- Программировать и обмениваться сообщениями с Arduino возможно всего лишь при помощи одного USB кабеля.

Недостатки:

- Работа только с платами Arduino;
- ОтсувствиеCodeGuard'a, который помогает быстро дописывать переменные, функции и т.д.;
- Необходим определённый bootloader, который будет разпозновать плату подключённую через USB как COM-порт;
- Некоторые встроенные функции не всегда удобны, так как используют некоторую аппаратную часть микроконтроллера;
- Данная среда разработки поддерживает только определённые микроконтроллеры atmel;
- Отсутствие какого-либо виртуального симулятора;
- Пустой проект Arduino занимает большое количество места в памяти платы.

Среда разработки LabVIEW

Достоинства:

- LabVIEW изначально создавалась для инженеров, а не для программистов. Поэтому эта среда (и язык программирования G) сделаны максимально интуитивно понятным.
- LabVIEW графический язык программирования, поэтому не нужно тратить время на проверку синтаксиса, поиск ошибок и опечаток. LabVIEW оперирует "объектами" (Терминалы, узлы, функции и т.д.).
- LabVIEW изначально создано для написания программа для автоматизации в промышленности и учебных лабораторий, поэтому для работы с периферией в своем составе имеет очень большой набор библиотек (работа с RS-232 (COM порт), LPT, USB, протоколы TCP/IP, UDP и т.д.). Сейчас очень многие известные производители плат сбора данных, камер машинного зрения, промышленного оборудования поставляются вместе с ним готовый набор библиотек для их оборудование в LabVIEW, с большим количеством

примеров. Что позволяет в сжатые сроки начать работу с оборудование и подстроить его под свои задачи.

- LabVIEW поддерживает работы с оборудование компании NationalInstruments.
- В LabVIEW имеется коллосальный набор библиотек для обработки и анализа сигнала, различные преобразования, фильтры и т.д.
- LabVIEW модульная система. Поэтому установив нужный модуль или тулкит, можно расширить ее возможности. Так есть модуль для машинного зрения, модуль работы со звуком, шаговыми двигателями, модуль для генерации отчетов в MS office, Internet модуль для передачи и публикации данных в сети.
- LabVIEW позволяет создавать законченные приложения и компилировать полноценное exe приложение.
- LabVIEW кроссплатформенная среда. Помимо написания программ под разные виды операционных систем, она позволяет писать ПО для WindowsMobile, встраиваемых система (CompactRIO, систем на основе Linux и WindowsEmbeded), поддерживает ARM микроконтроллеры), Blackfin, есть модуль для FPGA (в основном для оборудования NationalInstruments, однако можно писать ПО и для некоторых FPGA от Xilinx).
- LabVIEW позволяет создавать законченные приложение, в достаточно сжатые сроки, в ней очень проста отладка и тестирования программ.
- LabVIEW очень легко работает с параллельными потоками, они очень легко создаются и управляются, что играет не маловажную роль при разработке достаточно объемных проектов.

Недостатки:

- Несмотря на огромную гибкость данной среды, есть некоторые ограничения, например утилиты в LabVIEW писать достаточно не удобно.
- LabVIEW несколько уступает по производительности другим языкам программирования (это цена за скорость и гибкость), но не значительно.

- Чтобы запустить приложение созданное с помощью LabVIEW на компьютере в котором сама среда не установлена, нужно ставить LabVIEWRun-Time (или делать инсталлятор, при установке которого Run-Time поставиться автоматически), он бесплатный и всегда доступен для скачивания с сайта www.ni.com. Но имеет достаточно большой объем.
- LabVIEW предназначена для решения различных задач, связанных с автоматизацией, сбором и обработкой сенсорных данных и т.д. Поэтому в более общих задачах данная среда программирования проигрывает.

2.3. Методические рекомендации по изучению курса робототехники на уроках технологии.

Во время урока учащиеся могут дополнить готовые модели различными датчиками, что позволит им внести разнообразие в урок и сохранить интерес учащихся к решению проектных и конструкторских задач. Разработка собственного уникального устройства позволит ученикам отклоняться от инструкций, в том числе и с помощью собственного воображения. Умение создавать оригинальные модели будет способствовать повышению мотивации и активности каждого учащегося, что приведет к выводу обучения на новый продуктивный уровень.

Способ работы на технологических уроках курса робототехники основан на парной работе учащихся, поэтому рекомендуется разбивать один набор конструктора на двоих учеников. Это позволяет учащимся приобретать навыки сотрудничества и одновременно выполнять индивидуальные задания.

Основной принцип обучения робото-конструирования – «шаг за шагом» – обеспечивает учащимся возможность работать в собственном темпе.

Помимо учебных наборов программируемых устройств, важно оборудовать технологический кабинет путями для передвижения робота. Траектория, как правило, представляет собой черную линию на белом фоне.

Чем больше путей, тем разнообразнее круг задач, которые необходимо решать в рамках представленной дисциплины.

При организации занятий с использованием робототехнических устройств возникает следующий ряд вопросов:

- Планирование процедуры занятия;
- Результаты занятия;
- Оценка результатов занятия.

Планирование занятия с использованием робототехнических устройств, имеет ряд отличительных особенностей. В первую очередь – это значительные затраты времени на сборку устройств.

Вторая особенность связана с тем, что каждая группа учащихся на занятии будет работать в индивидуальном темпе. Часть класса будет организовывать свою деятельность размеренно, решая поставленную задачу, не выходя за рамки занятия, оставшаяся часть учащихся может справляться с заданиями гораздо быстрее. В последнем случае перед учителем встает вопрос об инвариантности поставленной задачи.

Итоговый контроль знаний и умений может быть реализован через мини-соревнования, где каждая группа учащихся продемонстрирует результаты решения поставленной задачи. Он также может учитывать время и скорость перемещения приборов, точность выполняемых действий, точность калибровки датчика и многое другое.

Для повышения эффективности обучения необходимо грамотно сочетать используемые технические средства обучения с применением следующих методов:

- Объяснительно-иллюстративный метод – предъявление информации ученикам различными словесными и наглядными способами;
- Эвристический метод – метод творческой деятельности;
- Проблемный метод – постановка проблемы учителем и самостоятельный поиск её решения обучающимися;

- Программированный метод – установка набора операций, которые необходимо совершить учащимся в ходе выполнения практических работ;
- Репродуктивный метод – воспроизводство знаний и способов деятельности по примеру учителя;
- Частично-поисковый метод – решение проблемных задач с помощью учителя;
- Поисковый метод – самостоятельное решение проблем учащимися;
- Метод проблемного изложения – постановка проблемы педагогом, решение ее самим педагогом, соучастие обучающихся при решении

Опираясь на ФГОС важно выделить универсальные учебные действия, формируемые в результате образовательного процесса:

- Мотивационная основа деятельности;
- Умение планировать своё действие в соответствии с поставленной задачей и условиями её реализации;
- Оценивать правильность выполнения действия;
- Осуществлять анализ объекта с выделением существенных признаков и несущественных;
- Осуществлять синтез как составление целого из частей;
- Допускать возможность существования у людей различных точек зрения, в том числе не совпадающих с его собственной, ориентироваться на позицию парнёрства в общении и взаимодействии;
- Договариваться и приходить к общему решению совместной деятельности.

Таким образом, можно составить следующую таблицу (Таблица 1), которая наглядно покажет рекомендации по изучению сред программирования на уроках технологии курса робототехники, основываясь на исследования данной работы.

Среда программирования	Рекомендуемый класс
Lego Education Wedo 2.0	1-3 класс
LegoMindstormsEV3	4-7 класс
Arduino IDE	8-9 класс
LabView	10-11 класс

Таблица 1. Рекомендации по изучению сред программирования.

Задачи, которые будут решены в ходе изучения каждых из сред программирования:

Lego Education Wedo 2.0 – научиться программировать простые действия и реакции механизмов.

LegoMindstormsEV3 – сформировать навыки конструирования и программирования роботов.

AdruinoIDE – научиться программировать роботов при помощи высокоуровневых языков программирования.

LabView – позволяет использовать аппаратную платформу Legoи Arduino для повышения эффективности решения задач встречавшихся ранее, поиска новых алгоритмов их решения, а так же расширения спектра проектируемых алгоритмов.

Выводы по второй главе.

Во второй главе проведен обзор сред программирования на уроках «Основы робототехники», таких как: Lego Education Wedo 2.0, Lego Mindstorms EV3, Arduino IDE, LabView для учеников, которые изучают этот курс дисциплины. Представлены к ознакомлению плюсы и минусы каждой из сред разработки.

Основной целью данной дисциплины является развитие алгоритмического мышления и обучения программированию в различных средах разработки, как на визуальном языке программировании, так и на высокоуровневом (текстовом). Так же курс «Основы робототехники» необходим для привитие интереса школьников к техническому творчеству, раскрытие талантов тех учеников, которые в дальнейшем могут стать первоклассными конструкторами, технологами, инженерами.

Для реализации дисциплины были разработаны методические рекомендации, в основе которых лежат принципы сознательности, активности обучающихся, принцип наглядности, систематичности и последовательности, прочного овладения знаниями и навыками, доступности и посильности, учета возрастных особенностей обучаемых. Применение разнообразных методов и форм способствует хорошему усвоению обучающимися материала и их заинтересованности.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы были разработаны методические рекомендации по использованию языков программирования в контексте повышения сложности и разнообразия решаемых задач, предназначенные для педагогов. Это необходимо, что бы у учащихся освоение основ по решению задач связанных с предметом «Основы робототехники» было на достаточно высоком уровне.

На первом этапе выполнения выпускной квалификационной работы был проведен анализ научной и методической литературы по организации учебной деятельности обучающихся при изучении робототехники на уроках технологии. Так же установлен круг доступного и распространенного аппаратного и программного обеспечения роботов для использования на уроках технологии.

На втором этапе выполнения выпускной квалификационной работы был проведен сравнительный анализ языков программирования роботов в контексте решаемых учебных задач, выявлены достоинства и недостатки каждой из сред программирования, а так же представлены методические рекомендации при проведении уроков «Основы робототехники».

Цель выпускной квалификационной работы достигнуты, были разработаны методические рекомендации по использованию языков программирования в контексте повышения сложности и разнообразия решаемых задач.

Список использованных источников

1. 123 эксперимента по робототехнике / М. Предко; пер. с англ. В. П. Попова. - М.: НТ Пресс, 2007. 544 с: ил.
2. Белоусов, И.Р. Дистанционное обучение механике и робототехнике через сеть Интернет [Текст] / И.Р. Белоусов, Д.Е. Охоцимский, А.К. Платонов [и др.] // Компьютерные инструменты в образовании.– 2003.– №2.– с. 34-41
3. Беспалько В.П. Основы теории педагогических систем. - Воронеж: изд-во ВГПУ, 1977. – 298 с.
4. Гершунский Б.С. Философия образования: Учебное пособие для студентов высших и средних педагогических учебных заведений. - М.: Московский психолого-социальный институт, 1998.- 432 с.: ил.
5. Книга: Системы искусственного интеллекта в машиностроении. Учебное пособие. Бровкова Б.В., 2004.
6. Мартыненко, Ю.Г. Динамика мобильных роботов // Соровский образовательный журнал.– 2000.– №5.– с. 110-116
7. Мякушко А.А. Основы образовательной робототехники: Учебнометодическое пособие для учителя.- М.,2010.- 80 с.
8. Николаев А.Б., Васюгова С.А. Программирование роботовманипуляторов: Методические указания к лабораторным работам по дисциплине «Интеллектуальные системы» - М.: Изд-во МАДИ. 2015.- 96 с.
9. Николаев А.Б., Остроух А.В. Интеллектуальные системы: учебное пособие - М.: МАДИ, 2012. – 271 с.
10. Остроух А.В., Николаев А.Б. Интеллектуальные системы в науке и производстве / Учебно-методическое пособие. – Saarbrucken, Germany: PalmariumAcademicPublishing, 2012. - 312 с.
11. Остроух А.В. Основы построения систем искусственного интеллекта для промышленных и строительных предприятий. Монография. – М.: ООО «Техполиграфцентр». 2008. - 280 с.

12. ПервоРобот NXT. Введение в робототехнику [Электронный ресурс] // LEGO MINDSTORMS Education. – Режим доступа: www.MINDSTORMSeducation.com
13. Программируемые роботы. Создаем робота для своей домашней мастерской / Дж. Вильяме; пер. с англ. А. Ю. Карцева. - М.: НТ Пресс, 2006. -240 с.: ил.
14. Программируемый робот, управляемый с КПК / Д. Вильяме; пер. с англ. А. Ю. Карцева. - М.: НТ Пресс, 2006. - 224 с.: ил
15. Психолого-педагогический словарь. / Сост. Рапацевич Е.С. – Минск, 2006.– с. 184-185
16. Руководствопользователя. LEGO MINDSTORMS Education EV3 .The LEGO GROUP. 2013.с. 98.
17. Русова Н.Ю. Теоретические основы моделирования дидактического материала: автореф. к. п. н. - Н.Новгород, 2001, - 26 с.
18. Сборка и программирование мобильных роботов в домашних условиях. Жимарши Ф., 2008.
19. Создание роботов в домашних условиях / Брага Ньютон; пер. с англ. Е. А. Добролежина. - М.: НТ Пресс, 2007. - 368 с.: ил.
20. Сайт компании LEGO [Электронный ресурс].– Режим доступа: http://www.lego.com/ru-ru/
21. Селевко Г.К. Современные образовательные технологии Учебное пособие. — М.: Народное образование, 1998. – 256 с.
22. Ситаров В.А. Дидактика. Учеб. пособие для студ. высш. пед. учеб. заведений / Под ред. В. А. Сластенина. – 2-е изд., стереотип. – М.: Издательский центр «Академия», 2004. – 368 с.
23. Устройства управления роботами. Схемотехника и программирование. Предко М., 2004.
24. Уроки Лего-конструирования в школе. Методическое пособие. Злаказов А.С., Горшков Г.А., Шевалдина С.

25. Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 02.03.2016) "Об образовании в Российской Федерации"//СПС КонсультантПлюс. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_140174/
26. Филиппов С.А. Робототехника для детей и родителей 3-е изд., доп. и испр. – СПб.: Наука, 2013. – 319 с. – (Шаги в кибернетику)
27. Хуторской А. В. Современная дидактика: учебник для вузов.-Спб.: Питер. 2007.- 639
28. Юревич Е. И. Основы робототехники. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2005. - 416 с: ил.
29. Федеральный государственный образовательный стандарт основного общего образования // Министерство образования и науки Российской Федерации URL: http://минобрнауки.рф/документы/543
30. ПервоРобот LEGO® WeDo™. Книга для учителя [Электронный ресурс]. - Режим доступа :http://robot.edu54.ru/sites/default/files/rukovodstvo_dlya_uchitelya_lego_education _wedo.pdf
31. Толстова Н. А. Образовательная робототехника как составляющая инженерно-технического образования / Н. А. Толстова, Д. А. Бондаренко, К. Ю. Ганьшин // Наука. Инновации. Технологии. -2013. - № 3. -С.171-177.
32. Шимов И.В. Применение робототехнических устройств в обучении программированию школьников // Педагогическое образование в России. - 2013. - С. 184. К.А.Вегнер Внедрение основ робототехники в современной школе // Вестник Новгородского Государственного Университета. 2013. № №74 Т.2. С