



ТЕХНОЛОГИИ РОБОТОТЕХНИКИ

ПЕРВЫЕ ШАГИ

Учебно-методическое пособие

Бюджетное учреждение высшего образования
Ханты-Мансийского автономного округа – Югры
«Сургутский государственный университет»

А.С. Буеров

ТЕХНОЛОГИИ РОБОТОТЕХНИКИ
первые шаги

Учебно-методическое пособие

Ульяновск
Зебра
2018

УДК 7.73.738
ББК 74.268.5
Б 90

Рецензенты:

доктор педагогических наук, профессор Ф.Д. Рассказов
доктор педагогических наук, профессор Э.Ф. Насырова

Б 90 Буеров, А.С. Технологии робототехники: первые шаги: учебно-методическое пособие. – Ульяновск: Зебра, 2018. – 79 с.

Учебно-методическое пособие предназначено для получения базового уровня знаний по робототехнике. Содержит теоретическую часть для самостоятельного изучения материала. Раскрыты основные аспекты в области робототехники, а так же представлены разработки практических занятий, контрольно-измерительные материалы. Данное пособие повысит интеллектуальный уровень, расширит кругозор у обучающихся, будет способствовать развитию самостоятельности, внимательности, инициативности, а так же даст ориентир на профессиональное самоопределение.

Учебно-методическое пособие будет интересно учителям информатики, технологии, педагогам дополнительного образования, учителям начальных классов, а так же тем, кто интересуется моделированием и конструированием в области робототехники.

ISBN 978-5-6042184-1-9

УДК 7.73.738
ББК 74.268.5

© Буеров, А.С., 2018
© БУ ВО «Сургутский государственный университет, 2018.

СОДЕРЖАНИЕ

Пояснительная записка.....	5
1. Знакомство с конструктором.....	6
2. Программирование движения робота.....	14
3. Знакомство с вычислительными возможностями робота.....	24
4. Датчик касания.....	31
5. Датчик цвета.....	38
6. Датчик цвета. Продолжение.....	44
7. Ультразвуковой датчик.....	51
8. Инфракрасный датчик.....	57
9. Инфракрасный датчик. Продолжение.....	65
10. Гироскопический датчик.....	72
Заключение.....	75
Используемая литература.....	76

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Научно-технический прогресс последних десятилетий неразрывно связан с интеллектуальным продуктом, открытиями и изобретениями, получаемыми в результате инновационной деятельности. Их роль в экономике значительно возрастает день ото дня. Создание конкурентоспособной продукции, имеющей высокую степень наукоёмкости и новизны, практически невозможно без применения инноваций, которые занимают одно из центральных мест в современной рыночной экономике, так как ведут к созданию новых потребностей, снижению себестоимости продукции, притоку инвестиций. Для потребителя продукты инновационной деятельности представляют собой максимально простые и удобные решения многих проблем. Одним из ведущих направлений современной прикладной науки является робототехника, которая занимается созданием и внедрением в жизнь человека автоматических машин, способных намного облегчить как промышленную сферу жизни, так и бытовую.

Роботостроение сегодня – довольно развитая отрасль промышленности: огромное количество роботов выполняют работу на различных предприятиях, изучение космического пространства или подводных глубин уже не обходится без использования робототехнических манипуляторов подводных или летательных аппаратов с высоким уровнем интеллекта. В стенах лабораторий создается все большее количество роботов бытового назначения, «умные машины» все чаще заменяют человека на рабочем месте.

ЦЕЛЬ: является саморазвитие и формирование конструкторских компетенций личности каждого учащегося в процессе освоения мира через его собственную творческо-техническую деятельность.

ЗАДАЧИ ПРОГРАММЫ:

- ознакомление с основными принципами механики;
- формирование у учащихся конструкторских компетенций и целостного представления о той части окружающей их действительности, которая создается человеческим обществом;
- формирование внутреннего плана деятельности на основе поэтапной отработки предметно-преобразовательных действий;
- формирование умения искать и преобразовывать необходимую информацию на основе различных информационных технологий (графических – текст, рисунок, схема; информационно-коммуникативных).

1. ЗНАКОМСТВО С КОНСТРУКТОРОМ

Введение:

Задача данного курса – познакомить вас с конструктором Lego mindstorms. Научить собирать базовые конструкции роботов, программировать их под определенные задачи, разобрать с вами базовые решения наиболее распространенных задач-соревнований.

Курс рассчитан на делающих первые шаги в мир робототехники с помощью конструктора Lego mindstorms. Хотя все примеры роботов в этом курсе сделаны с помощью конструктора Lego mindstorms EV3, программирование роботов объясняется на примере среды разработки Lego mindstorms EV3, тем не менее, владельцы Lego mindstorms NXT тоже могут присоединиться к изучению данного курса, и, надеемся, найдут для себя тоже полезное.

1.1. Что в наборе? Классификация деталей, крепление деталей между собой, главный блок, моторы, датчики.

Давайте начнем знакомиться с конструктором Lego mindstorms EV3. Распечатав конструктор, мы найдем в нем множество разнообразных деталей. Если вы знакомы с традиционными кирпичиками Lego, но раньше вам не приходилось сталкиваться с наборами Lego серии Technic, то, возможно, вы будете слегка обескуражены видом непривычных деталей. Однако, разобраться с ними совсем несложно. Итак, условно разделим все детали на несколько категорий. На рисунке представлены детали, называемые балками (иногда для этих деталей можно встретить название – бим (beam)) Балки исполняют роль каркаса (скелета вашего робота).



Рис.1

Следующая группа деталей служит для соединения балок между собой, с блоком и датчиками. Детали, имеющие крестообразное сечение, называются осями (иногда штифтами) и служат для передачи вращения от моторов к колесам и шестерням. Детали, похожие на цилиндры (имеющие в сечении окружность) называются пинами (от англ. pin - шпилька).



Рис.2

Представленный ниже рисунок демонстрирует вам различные варианты соединения балок с помощью пинов.



Рис.3

Следующую группу деталей называют коннекторами. Их главная задача – соединение балок в различных плоскостях, изменение угла соединения деталей и подсоединение датчиков к роботу.

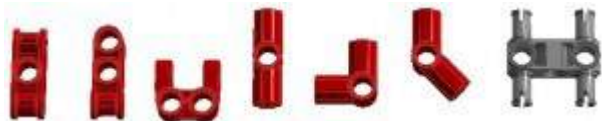


Рис.4

Переходим к следующей группе деталей. Шестерни предназначены для передачи вращения от моторов к другим элементам конструкции робота. Как правило, это колеса, но в тоже время шестерни могут широко применяться и в различных конструкциях роботов, не предполагающих вращение.

С ними мы непременно еще не раз встретимся при конструировании сложных механизмов.



Рис.5

Ну и, конечно же, движение в пространстве нашему роботу обеспечиваются различные колеса и гусеницы, представленные в наборе.



Рис.6

Следующая группа деталей несет в себе декоративные функции. С их помощью мы можем украсить нашего робота, придать ему неповторимый вид.



Рис.7

В набор Lego mindstorms EV3 входят два больших мотора. Моторы выполняют роль мышц или силовых элементов нашего робота. Большие моторы, наиболее часто используются для передачи вращения на колеса, тем самым, обеспечивая движение робота. Можно сказать, что эти моторы выполняют ту же роль, что и ноги человека.



Рис.8

Один средний мотор, который также входит в набор Lego mindstorms EV3 выполняет роль движущей силы для различного навесного оборудования робота (клешни, модули захвата, различные манипуляторы) По аналогии с большими моторами отведем среднему мотору ту же роль, которую у нас выполняют руки.

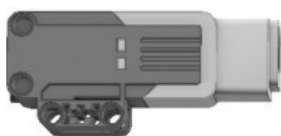


Рис.9

Датчики, входящие в набор Lego mindstorms, представляют роботу необходимую информацию из внешней среды. Главная задача программиста - научиться извлекать и анализировать информацию, поступающую с датчиков, а затем подавать верные команды на моторы для выполнения определенных действий.



Рис.10

Ну и основным элементом нашего конструктора является главный блок EV3. В этом корпусе заключен мозг нашего робота. Именно здесь выполняется программа, получающая информацию с датчиков, обрабатывающая её и передающая команды моторам.

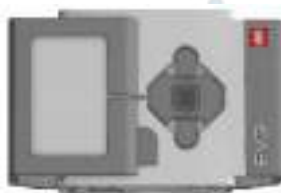


Рис.11

1.2. Собираем робота, с помощью которого будем изучать данный курс

Настало время – собрать нашего первого робота.

На первом этапе конструкция нашего робота будет следующей:

- Два больших мотора для того, чтобы мы смогли научить нашего робота поворачивать.
- Два ведущих колеса, на которые будут передаваться усилия моторов.
- Одно свободно вращающееся колесо или шаровая опора, которая будет придавать устойчивость нашему роботу.
- Один главный блок EV3, который будет хранить и выполнять нашу программу.
- Некоторое количество деталей для придания конструкции законченного вида.

Такой простейший робот называется роботом-тележкой.

Вы можете попробовать поэкспериментировать или собрать робота по предложенной инструкции в зависимости от версии вашего набора EV3:

Lego mindstorms EV3 Home



Lego mindstorms EV3 Education



Как только наш робот будет готов – начнем изучение среды программирования.

1.3. Знакомство со средой программирования

Первым делом загружаем среду программирования Lego mindstorms EV3. В главном меню программы выбираем: «Файл» – «Новый проект» или нажимаем «+», показанный на рисунке стрелкой.



Рис.12

В одном проекте может находиться множество программ. Для того, чтобы проект корректно загружался в нашего робота необходимо в названии проекта и программ использовать только буквы латинского алфавита. Давайте назовем наш проект lessons (уроки), а первую программу – lesson-1 (урок-1). Для того, чтобы дать название проекту, воспользуемся главным меню программы: «Файл» - «Сохранить проект как...» Чтобы изменить название программы – следует сделать двойной щелчок мышью на её названии (program) и вписать свое название.

Включим центральный блок нашего робота. Для этого нажмем на центральную (самую темную) кнопку блока. С помощью USB-кабеля, идущего в комплекте с конструктором, подключим робота к компьютеру. Успешное подключение робота отразится на вкладке аппаратных средств программного обеспечения EV3 в правом нижнем углу программы.



Рис.13

Если подключение робота прошло успешно, то приступим к программированию и создадим нашу первую программу.

1.4. Наша первая программа

Давайте научим нашего робота двигаться вперед на определенное расстояние. В нижней части экрана находится палитра программирования, каждому цвету палитры соответствуют различные группы программных блоков. Выберем зеленую палитру «Действие». Она содержит блоки управления моторами, блок вывода информации на экран, блок управления звуком и кнопка контроллера EV3 (главного блока). Выберем блок «Рулевое управление» и перетащим его в область программирования (центральная область программы).



Рис.14

Каждая программа состоит из цепочки блоков, задающих определенное действие или проверяющих различные условия. Каждый блок имеет множество различных параметров. Первый, оранжевый блок с зеленым треугольником внутри называется – «Начало». Именно с него начинается любая программа для нашего робота. Второй блок установили мы. Повторю – он называется «Рулевое управление». Его назначение - одновременное управление двумя моторами.



Рис.15

Но, если вы собирали робота по инструкции, предложенной выше, то, наверное, обратили внимание, что в ней отсутствует схема подключения моторов и датчиков. Настало время с этим разобраться. Блок EV3 имеет 4 порта, обозначенных цифрами: 1, 2, 3, и 4. Эти порты служат для подключения только датчиков. Для подключения моторов служат порты, обозначенные буквами: А, В, С и D. Можно подключать моторы в любые свободные порты, предназначенные для них. Но в случае управляемой тележки рекомендовано подключать моторы в порты: В и С.

Давайте сейчас возьмем два соединительных кабеля длиной 25 см, левый мотор подключим к порту В, а правый - к порту С. Именно это подключение выбрано по умолчанию в блоке «Рулевое управление». Специальная кнопка, обозначенная стрелкой, отвечает за режим работы блока. Для первой программы выберем режим: «Включить на количество оборотов». Значение 0 под черной стрелочкой на блоке означает прямолинейное движение, когда оба мотора крутятся с одинаковой скоростью. Число 75 задает мощность моторов, чем больше это значение, тем быстрее поедет наш робот. Цифра 2 задает количество оборотов каждого из моторов, на которое они должны прокрутиться.

Итак, наша первая программа готова. Загружаем ее в нашего робота. Для этого нажимаем кнопку «Загрузить» на вкладке аппаратных средств и отсоединяем USB-кабель от робота.



Рис.16

Устанавливаем робота на ровную поверхность. С помощью стрелок на блоке EV3 заходим в папку нашего проекта, выбираем программу lesson-1 и центральной кнопкой блока EV3 запускаем ее на выполнение.

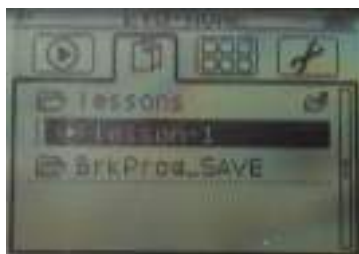


Рис.17

1. ПРОГРАММИРОВАНИЕ ДВИЖЕНИЯ РОБОТА

Введение:

На втором занятии мы подробнее познакомимся со средой программирования и подробно изучим команды, задающие движение нашему роботу-тележке, собранному на первом занятии. Итак, давайте запустим среду программирования Lego mindstorms EV3, загрузим наш проект lessons.ev3, созданный ранее и добавим в проект новую программу – lesson-2-1. Программу можно добавить двумя способами:

- Выбрать команду «Файл» – «Добавить программу» (Ctrl+N).
- Нажать «+» на вкладке программ.



Рис.2.1

2.1. Палитры программирования и программные блоки

Давайте теперь обратим свой взгляд в нижний раздел среды программирования. Из материала первого занятия мы уже знаем, что здесь находятся команды для программирования робота. Разработчики применили оригинальный прием и, сгруппировав программные блоки, присвоили каждой группе свой цвет, назвав группы палитрами.

Зеленая палитра называется: «Действие»:



Рис.2.2

На данной палитре расположены программные блоки управления моторами, блок вывода на экран, блок управления индикатором состояния модуля. Сейчас мы начнем изучение этих программных блоков.

2.2. Зеленая палитра – блоки действия

Первый программный блок зеленой палитры предназначен для управления средним мотором, второй блок – для управления большим мотором. Так как параметры этих блоков идентичны - рассмотрим настройку на примере блока – большой мотор.



Рис.2.3

Для правильной настройки блока управления большим мотором мы должны:

1. Выбрать порт, к которому подключен мотор (А, В, С или D) (Рис.2.3 поз.1).
2. Выбрать режим работы мотора (Рис. 2.3 поз. 2).
3. Настроить параметры выбранного режима (Рис. 2.3 поз. 3).

Чем же отличаются режимы? Режим: «Включить» включает мотор с заданным параметром «Мощность» и после этого управление передается следующему программному блоку программы. Мотор будет продолжать вращаться, пока не будет остановлен следующим блоком «Большой мотор» с режимом «Выключить» или следующий блок «Большой мотор» не будет содержать другие параметры выполнения. Режим «Включить на количество секунд» включает большой мотор с установленной мощностью на указанное количество секунд, и только по завершению времени мотор остановится, а управление в программе перейдет к следующему программному блоку. Аналогично поведет мотор себя в режимах «Включить на количество градусов» и «Включить на количество оборотов»: только после выполнения установленного вращения мотора, он остановится и управление в программе перейдет к следующему блоку.

Параметр мощность (на Рис.2.3 мощность установлена в 75) может принимать значения от -100 до 100. Положительные значения мощности задают вращение мотора по часовой стрелке, отрицательные – против часовой. При значении мощности равном 0 мотор вращаться не будет, чем «выше» значение мощности, тем быстрее вращается мотор.

Параметр мощность задается только целыми значениями, параметры: секунды, градусы, обороты могут принимать значения с десятичной дробью. Но следует помнить, что минимальный шаг вращения мотора равен одному градусу.

Отдельно следует сказать о параметре «Тормозить в конце». Данный параметр, если установлен в значение «Тормозить» заставляет мотор тормо-

зять после выполнения команды, а если установлен в значение «Двигаться назад», то мотор будет вращаться по инерции, пока сам не остановится.

Следующие два программных блока «Рулевое управление» и «Независимое управление моторами» реализуют управление парой больших моторов. По умолчанию левый большой мотор подключается к порту «В», а правый – к порту «С». Но вы можете в настройках блока поменять порты подключения в соответствии с требованиями вашей конструкции (Рис.2.4 поз. 1).



Рис.2.4

Параметр «Рулевое управление» (Рис.2.4 поз.2) может принимать значения от -100 до 100. Отрицательные значения параметра заставляют робота поворачивать налево, при значении равном 0 робот движется прямо, а положительные значения заставляют робота поворачивать направо. Стрелка над числовым параметром меняет свою ориентацию в зависимости от значения, подсказывая тем самым направление движения робота (Рис.2.5).



Рис.2.5

Программный блок «Независимое управление моторами» похож на программный блок «Рулевое управление». Он также управляет двумя большими моторами, только вместо параметра «Рулевое управление» появляется возможность независимого управления мощностью каждого мотора. При равном значении параметра «Мощность» для левого и правого мотора робот будет двигаться прямолинейно. Если на один мотор подать отрицательное значение мощности (например -50), а на второй – положительное значение (например 50), то робот будет разворачиваться на месте (Рис.2.6).



Рис.2.6

Режимы работы этих блоков аналогичны режимам блока управления одним мотором, поэтому дополнительного описания не требуют.

2.3. Прямолинейное движение, повороты, разворот на месте остановка

Итак, теперь мы можем написать программу движения робота по какому-либо маршруту.

Задача 1: Проехать прямолинейно вперед на 4 оборота двигателя. Развернуться. Проехать на 720 градусов.

Решение (Рис.2.7):

1. Используя программный блок «Рулевое управление» проехать вперед на 4 оборота.

2. Используя программный блок «Независимое управление моторами» развернуться на месте (значение градусов придется подобрать экспериментально).

3. Используя программный блок «Рулевое управление» проехать вперед на 720 градусов.

Примечание: Почему при развороте пришлось подбирать значение градусов в блоке 2? Разве не 360 градусов – искомая величина? Нет, если мы зададим значение параметра «Градусы» равным 360, то тем самым заставим на искомую величину повернуться валы левого и правого моторов нашего робота. На какой угол повернется робот вокруг своей оси – зависит от размера (диаметра) колес и расстояния между ними. На Рис. 2.7 значение параметра «Градусы» равно 385. Данное значение позволяет роботу, собранному по инструкции small-robot 45544 развернуться вокруг своей оси. Если у вас другой робот, то вам придется подобрать другое значение. Можно ли это значение найти математически? Можно, но об этом мы поговорим позднее.

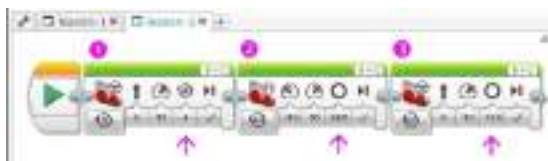


Рис.2.7

Задача 2: Установите на ровной поверхности какое-либо препятствие (банку, кубик, небольшую коробку), отметьте место старта вашего робота. Создайте в проекте новую программу: lesson-2-2, позволяющую роботу объехать вокруг препятствия и вернуться к месту старта.

Сколько программных блоков вы использовали? Поделитесь своим успехом в комментарии к уроку.

2.4. Экран, звук, индикатор состояния модуля

Программный блок «Экран» позволяет выводить текстовую или графическую информацию на жидкокристаллический экран блока EV3. Какое это может иметь практическое применение? Во-первых, на этапе программирования и отладки программы можно выводить на экран текущие показания датчиков во время работы робота. Во-вторых, можно выводить на экран название промежуточных этапов выполнения программы. Ну а в-третьих, с помощью графических изображений можно «оживить» экран робота, например с помощью мультипликации.



Рис.2.8

Программный блок «Экран» имеет четыре режима работы: режим «Текст» позволяет выводить текстовую строку на экран, режим «Фигуры» позволяет отображать на экране одну из четырех геометрических фигур (прямая, круг, прямоугольник, точка), режим «Изображение» может вывести на экран одно изображение. Изображение можно выбрать из богатой коллекции изображений или нарисовать свое, используя редактор изображений. Режим «Окно сброса настроек» сбрасывает экран модуля EV3 к стандартному информационному экрану, показываемому во время работы программы.



Рис.2.9

Рассмотрим параметры программного блока «Экран» в режиме «Текст» (Рис.2.9 поз.1). Строка, предназначенная для вывода на экран, вводится в специальное поле (Рис.2.9 поз.2). К сожалению, в поле ввода текста можно вводить только буквы латинского алфавита, цифры и знаки препинания. Если режим «Очистить экран» установлен в значение «Истина», то экран перед выводом информации будет очищен. Поэтому, если вам требуется объединить текущий вывод с информацией уже находящейся на экране, то устано-

вите этот режим в значение «Ложь». Режимы «X» и «Y» определяют точку на экране, с которой начинается вывод информации. Экран блока EV3 имеет 178 пикселей (точек) в ширину и 128 пикселей в высоту. Режим «X» может принимать значения от 0 до 177, режим «Y» может принимать значения от 0 до 127. Верхняя левая точка имеет координаты (0, 0), правая нижняя (177, 127).

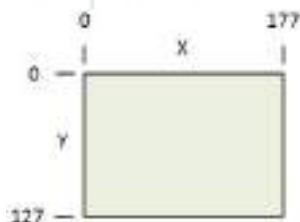


Рис.2.10

Во время настройки программного блока «Экран» можно включить режим предварительного просмотра (Рис.2.9 поз.3) и визуально оценить результат настроек вывода информации.

В режиме «Фигуры» (Рис.2.11 поз.1) настройки программного блока меняются в зависимости от типа фигуры. Так при отображении круга необходимо будет задать координаты «X» и «Y» центра окружности, а также значение «Радиуса». Параметр «Заполнить» (Рис.2.11 поз.2) отвечает за то, что будет отображен либо контур фигуры, либо внутренняя область фигуры будет заполнена цветом, заданным в параметре «Цвет» (Рис.2.11 поз.3).



Рис.2.11

Для отображения прямой необходимо задать координаты двух крайних точек, между которыми располагается прямая.



Рис.2.12

Чтобы отобразить прямоугольник следует задать координаты «X» и «Y» левого верхнего угла прямоугольника, а также его «Ширину» и «Высоту».



Рис.2.13

Отобразить точку проще всего. Укажите лишь её координаты «X» и «Y». Режим «Изображение», наверное, самый интересный и самый используемый режим. Он позволяет выводить на экран изображения. Среда программирования содержит огромную библиотеку изображений, отсортированную по категориям. В дополнение к имеющимся изображениям вы всегда можете создать свой рисунок и, вставив его в проект, вывести на экран. («Главное меню среды программирования» – «Инструменты» – «Редактор изображений»). Создавая своё изображение, вы можете также вывести на экран символы русского алфавита.



Рис.2.14

Как вы видите – отображению информации на экране главного модуля EV3 среда программирования придает огромное значение. Давайте рассмотрим следующий важный программный блок «Звук». С помощью этого блока

мы можем выводить на встроенный динамик блока EV3 звуковые файлы, тона произвольной длительности и частоты, а также музыкальные ноты. Давайте рассмотрим настройки программного блока в режиме «Воспроизвести тон» (Рис.2.15). В этом режиме необходимо задать «Частоту» тона (Рис.2.15 поз.1), «Продолжительность» звучания в секундах (Рис. 2.15 поз.2), а также громкость звучания (Рис.2.15 поз.3).



Рис.2.15

В режиме «Воспроизвести ноту» вам вместо частоты тона необходимо выбрать ноту на виртуальной клавиатуре, а также установить длительность звучания и громкость (Рис.2.16).



Рис.2.16

В режиме «Воспроизвести файл» вы можете выбрать один из звуковых файлов из библиотеки (Рис.2.17 поз.1), либо, подключив к компьютеру микрофон, с помощью Редактора звука («Главное меню среды программирования» – «Инструменты» – «Редактор звука») записать собственный звуковой файл и включить его в проект.



Рис.2.17

Давайте отдельно рассмотрим параметр «Тип воспроизведения» (Рис.2.17 поз.2), общий для всех режимов программного блока «Звук». Если данный параметр установлен в значение «Ожидать завершения», то управление следующему программному блоку будет передано только после полного воспроизведения звука или звукового файла. В случае установки одного из двух следующих значений начнется воспроизведение звука и управление в программе перейдет к следующему программному блоку, только звук или звуковой файл будет воспроизведен один раз или будет повторяться, пока не его не остановит другой программный блок «Звук».

Нам осталось познакомиться с последним программным блоком зеленой палитры – блоком «Индикатор состояния модуля». Вокруг кнопок управления модулем EV3 смонтирована цветовая индикация, которая может светиться одним из трех цветов: зеленым, оранжевым или красным. За включение – выключение цветовой индикации отвечает соответствующий режим (Рис.2.18 поз.1). Параметр «Цвет» задает цветовое оформление индикации (Рис.2.18 поз.2). Параметр «Импульсный» отвечает за включение – отключение режима мерцания цветовой индикации (Рис.2.18 поз.3). Как можно использовать цветовую индикацию? Например, можно во время различных режимов работы робота использовать различные цветовые сигналы. Это поможет понять: так ли выполняется программа, как мы запланировали.

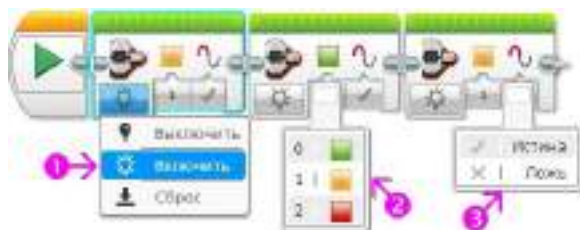


Рис.2.18

Давайте используем полученные знания на практике и немного «раскрасим» нашу программу из Задачи 1.

Задача 3:

1. Воспроизвести сигнал «Start».
2. Включить зеленую немигающую цветовую индикацию.
3. Отобразить на экране изображение «Forward».
4. Проехать прямолинейно вперед на 4 оборота двигателя.
5. Включить оранжевую мигающую цветовую индикацию.
6. Развернуться.
7. Включить зеленую мигающую цветовую индикацию.
8. Отобразить на экране изображение «Backward».
9. Проехать на 720 градусов.
10. Воспроизвести сигнал «Stop».

Решите задачу 3 самостоятельно, не подглядывая в решение. Удачи.

Решение задачи 3



3. ЗНАКОМСТВО С ВЫЧИСЛИТЕЛЬНЫМИ ВОЗМОЖНОСТЯМИ РОБОТА

Введение:

Наше третье занятие мы посвятим изучению вычислительных возможностей модуля EV3 и разберем примеры практических решений задач на вычисление траектории движения. Снова запускаем среду программирования Lego mindstorms EV3, загружаем наш проект lessons.ev3 и добавляем в проект новую программу – lesson-3-4. Добавлять новую программу в проект мы научились с вами на предыдущем уроке.

3.1. Красная палитра – операции с данными

Программные блоки, необходимые для выполнения различных операций над числовыми, логическими или текстовыми данными, сосредоточены в красной палитре среды программирования Lego mindstorms EV3. Красная палитра содержит 10 программных блоков. В отличие от зеленой палитры – с программными блоками красной палитры мы будем знакомиться постепенно, по мере продвижения по курсу программирования и возникновения необходимости в новых программных конструкциях.



Рис.3.1

3.2. Числовые значения. Блок «Константа», блок «Переменная»

Среда программирования Lego mindstorms EV3 позволяет нам обрабатывать в своих программах пять различных типов данных: «Текст», «Числовое значение», «Логическое значение», «Числовой массив», «Логический массив». В сегодняшнем уроке мы научимся оперировать с числовыми данными. Тип данных «Числовое значение» позволяет нам выполнять различные математические операции над числами. Числа в программе могут быть как положительными, так и отрицательными, быть целыми значениями или содержать десятичную дробь. Примеры: -15; 3,145; 8; -247,34.

Перед тем, как начать обрабатывать различные типы данных в наших программах, нам надо научиться их создавать и хранить. Для этих целей среда программирования Lego mindstorms EV3 предоставляет два вида программных блоков: «Переменная» и «Константа». Эти блоки позволяют создать в памяти робота специальные ячейки, позволяющие записывать, извлекать и редактировать различные типы данных. Программный блок «Константа» (Рис.3.2) позволяет создавать ячейку памяти для хранения одно-

го из пяти типов данных (Рис.3.2 поз.1). Требуемое значение записывается в ячейку на этапе создания программы (Рис.3.2 поз.2) и остается неизменным во время выполнения всей программы. Для получения значения, записанного в блок «Константа» используется «Вывод» (Рис. 3.2 поз.3). Подробнее с извлечением данных из программных блоков мы познакомимся ниже при решении практической задачи Урока №3.



Рис.3.2

В отличие от программного блока «Константа» – в блоке «Переменная» присутствуют два режима «Считывание» и «Записать» (Рис.3.3 поз.1). Перед первым использованием необходимо задать имя переменной, выбрав параметр блока «Добавить переменную» (Рис.3.3 поз.2). Имя переменной может содержать только заглавные и строчные буквы латинского алфавита, цифры, а также символы «_» и «-». Задать значение переменной можно, записав или передав число в параметр «Значение» (Рис.3.3 поз.3).



Рис.3.3

3.3. Блок математика, блок округление

Для выполнения математических вычислений служит программный блок «Математика». Он позволяет выполнить выбранную математическую операцию (Рис.3.4 поз.1) над двумя числами, заданными параметрами «a» и «b». В режимах «Абсолютная величина» и «Квадратный корень» для вычисления доступен только один параметр «a».



Рис.3.4

Отдельно следует остановиться на режиме «Дополнения». В этом режиме количество параметров для расчета увеличивается до четырех: «a», «b», «c» и «d». В параметр «Уравнение» (Рис.3.5 поз.1) можно вписать любую произвольную формулу, производящую вычисления с этими параметрами.



Рис.3.5

Иногда возникает необходимость произвести округление результата вычисления. Например: при отладке программы, можно выводить на экран модуля EV3 округленные промежуточные расчеты, чтобы легче было визуально контролировать ход выполнения программы. Для этого предназначен программный блок «Округление» (Рис.3.6). Режимы «До ближайшего», «Округлить к большему» и «Округлить к меньшему» производят округление до целого значения. В режиме «Отбросить дробную часть» можно задать количество остающихся знаков дробной части после запятой.

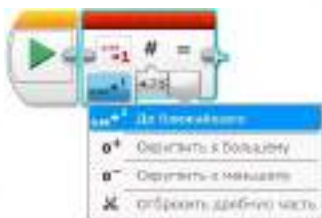


Рис.3.6

3.4. Примеры выполнения вычислений в программе

Настало время применить полученные знания на практике.

Задача №4: необходимо написать программу прямолинейного движения для проезда роботом расстояния в 1 метр.

Решение:

За один полный оборот мотора робот проезжает расстояние, равное длине окружности колеса. Это расстояние можно найти, умножив число Пи ($\pi \approx 3,14159$) на диаметр колеса. Диаметр колеса из образовательного набора Lego mindstorms EV3 равен 56 мм, а – из домашнего набора Lego mindstorms EV3 равен 43,2 мм. Если переведем расстояние в 1 метр в миллиметры (1000 мм) и разделим на расстояние, которое робот проходит за один оборот мотора, то узнаем: сколько оборотов мотора необходимо для проезда всего заданного расстояния.



Рис.3.7

Приступим к созданию программы:

- Используя программный блок «Константа», заведем в программу постоянное число Пи, равное примерно 3,14159.



- Используя программный блок «Переменная», создадим в программе переменную D и занесем в нее значение диаметра колеса в зависимости от используемого конструктора (если вы использовали другие колеса, то самостоятельно измерьте диаметр и внесите значение в программный блок).

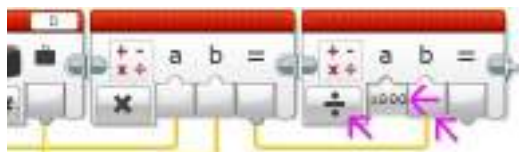


- Используя программный блок «Математика», умножим значение блока «Константа» на значение переменной D. Для передачи значения из переменной D в программный блок «Математика» используем второй программный блок «Переменная» в режиме «Считывание». (Для передачи значений между программными блоками используются шины данных. Чтобы установить шину данных, необходимо «потянуть» выходной параметр одного программного

блока и «присоединить» его к входному параметру другого программного блока).



- Используя программный блок «Математика», разделим значение пути (1000 мм) на значение, полученное в шаге 3.



- Полученное в шаге 4 значение, округлив до двух знаков после запятой, выведем на экран модуля EV3.



- Полученное в шаге 4 значение подадим в параметр «Обороты» блока «Рулевое управление».



Загрузим полученную программу в нашего робота. Поставим робота на ровную свободную площадку и запустим программу. Измерив расстояние, пройденное роботом, убедимся в правильности нашей программы.



Задача №5: необходимо написать программу, рассчитывающую значение параметра «Градусы» для разворота нашего робота (Урок №2, Задача №1).

Данная задача имеет сходство с предыдущей – нам только требуется найти расстояние, которое должны проехать колеса нашего робота. Для того, чтобы наш робот развернулся на 180 градусов – необходимо, чтобы правое и левое колеса, проехав определенный путь по окружности, поменялись местами. Как видим из Рис.3.8 – каждое колесо при этом проедет ровно половину окружности с диаметром, равным расстоянию между центрами колес (красная линия на Рис.3.8). Подходящей линейкой померяем расстояние между центрами колес. Для робота, собранного по инструкции small-robot-45544, это расстояние равно 120 мм. Следовательно, умножив это значение на число Пи (3,14159) и разделив на 2, мы найдем расстояние, которое должно проехать каждое из колес нашего робота. Как найти соответствующее этому расстоянию число оборотов мотора – мы разобрали в Задаче 4 данного урока. Для того, чтобы перевести полученное число оборотов в градусы – вспомним соотношение: 1 оборот мотора = 360 градусов. Следовательно, если мы, воспользовавшись программным блоком «Математика», умножим полученное значение оборотов на 360 и подадим результат в параметр «Градусы» программного блока «Независимое управление моторами» (Урок №2 Рис.2.7 поз. 2), то решим требуемую задачу.

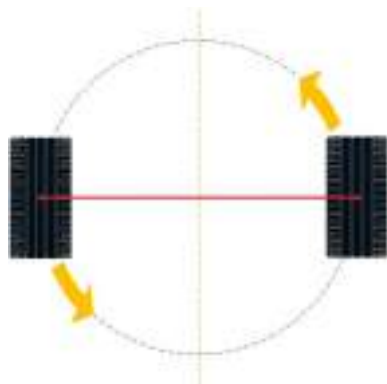


Рис.3.8

Попробуйте написать программу для решения задачи №5 самостоятельно, не подглядывая в решение.

Решение задачи №5



Оптимизация решения задачи 5

Одна из важнейших задач программиста, это не только решение поставленной задачи, а поиск оптимального решения, которое позволяет экономить память и вычислительные ресурсы. Давайте еще раз вернемся к решению задачи №5.

Обозначим через D_1 – диаметр колеса нашего робота, а через D_2 – расстояние между центрами колес.

Тогда расстояние, пройденное каждым колесом при развороте, может быть найдено по формуле:

$$\frac{\text{Пи} * D_2}{2}$$

Расстояние, которое проезжает робот за один оборот, выражается формулой:

$$\text{Пи} * D_1$$

Разделив первую формулу на вторую, вычислим значение оборотов двигателя для разворота робота:

$$\frac{\text{Пи} * D_2}{2 * \text{Пи} * D_1} = \frac{D_2}{2 * D_1}$$

Чтобы получить необходимое нам значение градусов, умножим предыдущую формулу на 360:

$$\frac{D_2}{2 * D_1} * 360 = \frac{D_2}{D_1} * 180$$

Итак: для того, чтобы развернуть нашего робота на 180 градусов, мы должны расстояние между центрами колес разделить на диаметр колеса и умножить на 180. Наша формула значительно упростилась и имеет замечательное следствие: если мы вместо 180 поставим в нашу формулу произвольное значение, то именно на этот угол в итоге повернет вокруг своей оси наш робот.

Решим нашу Задачу №5 с помощью программного блока «Математика» в режиме «Дополнения»:



4. ДАТЧИК КАСАНИЯ

Введение:

В состав конструктора Lego mindstorms EV3 входят различные датчики. Главная задача датчиков – представлять информацию из внешней среды модулю EV3, а задача программиста – научиться получать и обрабатывать эту информацию, подавая необходимые команды моторам робота. На протяжении ряда уроков мы будем последовательно знакомиться со всеми датчиками, входящими и в домашний, и в образовательный наборы, научимся взаимодействовать с ними и решать наиболее распространенные задачи управления роботом.

4.1. Изучаем первый датчик – датчик касания

Для подключения датчиков к модулю EV3 предназначены порты, обозначенные цифрами «1», «2», «3» и «4». Таким образом, к одному модулю EV3 одновременно можно подключить до четырех различных датчиков. Все порты абсолютно равнозначны и вы можете подключать датчики к любым портам, главное – будьте внимательны при указании номера порта для соответствующих датчиков в ваших программах.



Рис.4.1

Первым датчиком, который мы изучим, будет датчик касания (Рис.4.2).



Рис.4.2

Этот датчик, по сути, представляет собой специальную кнопку, которая может находиться в двух состояниях: «Нажатие» (Рис.4.3 поз.1) или «Освобождение» (Рис.4.3 поз.2). Также, последовательный переход в состояние "Нажатие", а затем «Освобождение» называется: «Щелчок» (Рис.4.3 поз.3) и может обрабатываться программой, как самостоятельное событие.



Рис.4.3

4.2. Оранжевая палитра – Управление операторами

Какие же инструменты представляет нам среда программирования для получения информации с датчиков и реагирования на эту информацию в программе? Давайте начнем знакомиться с программными блоками, расположенными в Оранжевой палитре, которая называется «Управление операторами» (Рис.4.4).



Рис.4.4

Программные блоки оранжевой палитры, не смотря на свою малочисленность, очень важны. С помощью этих блоков мы можем обрабатывать массу событий и условий и сложно представить практическую программу, которая может обойтись без этих блоков.

- С самым первым блоком Оранжевой палитры мы уже с вами знакомы: он называется «Начало». Именно с него начинаются все программы для роботов.

- Второй программный блок называется «Ожидание». Этот блок заставляет программу ожидать выполнения какого-либо условия или наступления какого-либо события. Пока не выполнится условие, установленное в этом блоке, программа не перейдет к выполнению следующих программных блоков. Если перед тем, как начнется выполнение блока «Ожидание» были включены, какие-либо моторы, то они будут продолжать вращаться с установленной скоростью.

- Третий программный блок называется «Цикл». Этот блок многократно выполняет программные блоки, вложенные внутрь его, пока не будет выполнено условие завершения цикла, заданное в настройках блока.

- Следующий программный блок называется «Переключатель». Он служит для того, чтобы в зависимости от заданных условий – выполнить одну последовательность программных блоков, вложенных в один из своих кон-

тейнеров.

- Заключительный программный блок называется «Прерывание цикла». Его предназначение – досрочное прекращение выполнения заданного цикла.

Программные блоки «Ожидание», «Цикл» и «Переключатель» имеют множество режимов и соответствующих настроек,знакомиться с которыми мы будем на практических примерах, последовательно и с наглядными пояснениями.

4.3. Оранжевая палитра, программный блок «Ожидание»

Перед тем, как приступить к решению практических задач, давайте закрепим датчик касания на нашем роботе, как показано на Рис.4.5, и подключим его кабелем к порту «1» модуля EV3.

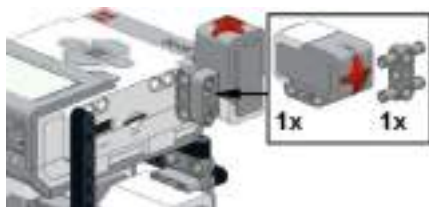


Рис.4.5

Задача №6: необходимо написать программу, запускающую движение робота по щелчку кнопки.

Решение:

Само условие задачи подсказывает нам возможное решение: перед началом движения – необходимо дождаться нажатия-отпускания кнопки датчика касания. Возьмем программный блок «Ожидание», изменим режим программного блока на «Датчик касания» – «Сравнение» (Рис.4.6).



Рис.4.6

Как можно увидеть – программный блок «Ожидание» сменил свое отображение. Рядом с песочными часами появилось изображение датчика касания (Рис.4.7 поз.1), помогающее в программе визуально оценивать установленный режим работы. Настройка программного блока «Состояние» задает требуемое состояние датчика, достижение которого прекратит выполнение блока «Ожидание» (Рис.4.7 поз.2). Настройка «Состояние» может принимать следующие значения: «0» – «Отпущено», «1» – «Нажатие», «2» – «Щелчок». Для решения нашей задачи выберем состояние «Щелчок». Вывод «Измеренное значение» (Рис.4.7 поз.3) при необходимости позволяет передать окончательное состояние датчика для обработки в другой программный блок.

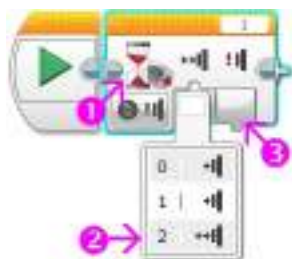


Рис.4.7

Итак: при такой настройке блока ожидания выполнение нашей программы будет остановлено до нажатия-отпускания кнопки датчика касания. Только после «Щелчка» выполнение будет передано следующему программному блоку. Установим после блока ожидания один программный блок «Рулевое управление», загрузим программу в робота и убедимся в правильности её выполнения (Рис. 4.8).

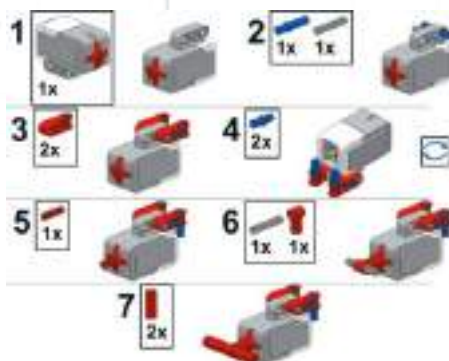


Рис.4.8

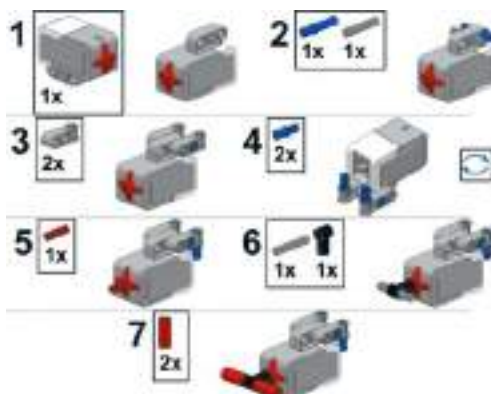
Задача №7: необходимо написать программу, останавливающую робота, столкнувшегося с препятствием.

Из датчика касания давайте соберем небольшой бампер, который будет нам сигнализировать о том, что наш робот столкнулся с препятствием. Ниже приведены подробные инструкции для сборки, как из домашней, так и из образовательной версии конструктора Lego mindstorms EV3. Можете поэкспериментировать и придумать собственный вариант конструкции.

Lego mindstorms EV3 home

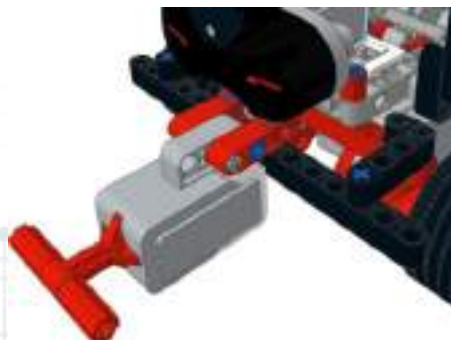


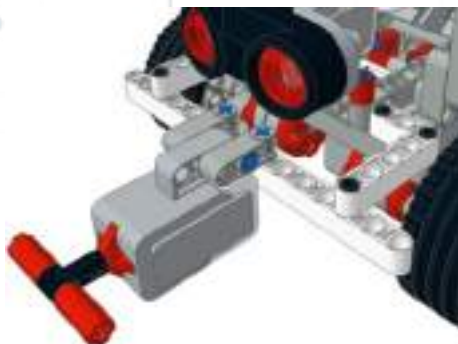
Lego mindstorms EV3 education



Получившийся элемент закрепим на передней балке нашего робота и соединим датчик касания с портом «1» модуля EV3.

Lego mindstorms EV3 Home





Конструкция готова. Приступим к созданию программы. По условию задачи: робот должен двигаться вперед, пока не наткнется на препятствие. В этом случае датчик касания будет нажат! Для решения снова воспользуемся программным блоком «Ожидание».

Решение:

4. Начать прямолинейное движение вперед (Рис.4.9 поз.1).
5. Ждать, пока датчик касания не будет нажат (Рис.4.9 поз.2).
6. Прекратить движение вперед (Рис.4.9 поз.3).



Рис.4.9

Для решения следующей задачи нам понадобится программный блок «Цикл» Оранжевой палитры.

Задача №8: необходимо написать программу, заставляющую робота двигаться вперед, при наезде на препятствие – отъезжать назад, поворачивать вправо на 90 градусов и продолжать движение вперед до следующего препятствия.

Подсказка: напишите и протестируйте программу движения – отъезда – поворота, а затем поместите эти блоки внутри программного блока «Цикл».

Решение задачи №8

1. Включаем моторы для прямолинейного движения вперед (Рис.4.10 поз.1).
2. Ожидаем нажатия датчика касания (Рис.4.10 поз.2).
3. Выключаем моторы (Рис.4.10 поз.3).
4. Отъезжаем немного назад (Рис.4.10 поз.4).
5. Рассчитываем значения параметра для поворота робота вправо на 90 градусов (диаметр колес робота равен 56 мм (образовательная версия конструктора)) (Рис.4.10 поз.5).

6. Поворачиваем вправо на 90 градусов (Рис.4.10 поз.6).
- 7.Пункты 1–6 повторяем в бесконечном цикле (Рис.4.10 поз.7).



Рис.4.10

5. ДАТЧИКА ЦВЕТА

Введение:

На этом уроке мы продолжаем знакомство с датчиками набора Lego mindstorms EV3. На очереди - датчик цвета, очень важный и полезный датчик! В большинстве конструкций он является, тем, чем у человека являются глаза. Поэтому изучению датчика цвета мы посвятим два последовательных урока, но в дальнейшем курсе еще вернемся к его изучению и использованию.

5.1. Изучаем второй датчик – датчик цвета

Датчик цвета может работать в трех различных режимах:

- в режиме «Цвет» датчик может определить цвет поднесенного к нему предмета;
- в режиме «Яркость отраженного света» датчик направляет световой луч на близкорасположенный предмет и по отраженному пучку определяет яркость предмета;
- в режиме «Яркость внешнего освещения» датчик может определить – насколько ярко освещено пространство вокруг.



Рис.5.1

5.2. Датчик цвета. Режим «Цвет»

В режиме «Цвет» датчик цвета достаточно точно умеет определять семь базовых цветов предметов, находящихся от него на расстоянии примерно в 1 см. Это следующие цвета: «черный» = 1, «синий» = 2, «зеленый» = 3, «желтый» = 4, «красный» = 5, «белый» = 6 и «коричневый» = 7. Если предмет удален от датчика или некорректно определяется цвет предмета – датчик информирует об этом состоянием «Без цвета» = 0.

Давайте, не затягивая, перейдем к практическому занятию.

Задача №9: необходимо написать программу, называющую цвета предметов, подносимых к датчику цвета.

Если вы собирали своего тренировочного робота по инструкции этого курса, то у вас датчик цвета уже размещен внутри робота и направлен вниз.

Потребуется приложить некоторые усилия, может быть даже слегка разобрать-собрать нашу конструкцию, чтобы подключить кабелем датчик цвета, например к порту «2» модуля EV3. Для отладки программы нам также понадобятся несколько цветных предметов: это могут быть кирпичики конструктора Lego, полоски цветной бумаги или цветные кубики. Для лучшего результата следует взять цвета, максимально приближенные к основным, но датчик довольно неплохо справляется с распознаванием подходящих оттенков. Чтобы не снимать датчик цвета и не крепить его в другом месте, во время выполнения программы можно держать робота перевернутым вверх колесами.



Рис.5.2

5.3. Оранжевая палитра, программный блок «Переключатель»

В решении Задачи №9 нам поможет программный блок «Переключатель» Оранжевой палитры. Этот блок в зависимости от настроек выбирает для выполнения программные блоки, расположенные в одном из своих контейнеров. Рассмотрим настройку этого блока в режиме работы с датчиком цвета.

Создадим новую программу «lesson-5-9», установим в программе блок «Переключатель», выберем режим «Датчик цвета» – «Измерение» – «Цвет» (Рис.5.3). В отличие от программного блока «Ожидание», программный блок «Переключатель» не ждет, пока наступит определенное событие, а проверяет текущее состояние и выполняет программные блоки, находящиеся в контейнере, сопоставленном текущему состоянию.



Рис.5.3

Рассмотрим подробнее настройки программного блока «Переключатель»:

7. Выбранный режим устанавливает изображение датчика цвета в блоке (Рис.5.4 поз.1).

8. Порт, к которому подключен датчик, отображается в соответствующем поле блока (Рис.5.4 поз.).

9. В настройках каждого программного контейнера выбирается значение, в соответствии с которым будут выполняться программные блоки, вложенные в этот контейнер (Рис.5.4 поз.3).

10. Один из контейнеров должен быть объявленным «Вариантом по умолчанию» – в случае, если значению, полученному от датчика, не соответствует ни один контейнер, то выполняется контейнер, объявленный «Вариантом по умолчанию» (Рис.5.4 поз.4).

11. Кнопка «+» добавляет программный контейнер в блок «Переключатель» (Рис.5.4 поз.5).

12. Программный блок «Переключатель» может автоматически растягиваться, чтобы вместить все блоки, помещаемые внутрь. С помощью меток, помеченных красными стрелками, можно самому изменять размеры блока (Рис.5.4).

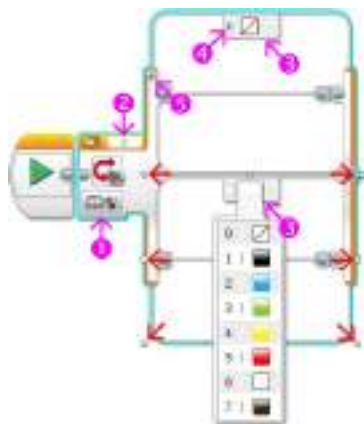


Рис.5.4

Продолжим формирование программного блока «Переключатель»:

1. Создадим необходимое количество контейнеров, соответствующее количеству цветов для распознавания «+» вариант «Без цвета»,

2. В настройках контейнеров установим распознаваемые цвета,

3. Вариантом по умолчанию выберем вариант «Без цвета»,

4. В каждый контейнер кроме варианта «Без цвета» (этот контейнер останется пустым) поместим программный блок «Звук» зеленой палитры.

5. Каждому цвету сопоставим соответствующий звуковой файл.

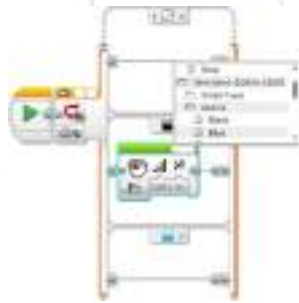


Рис.5.5

Наш программный блок «Переключатель» значительно увеличился в размерах. Специальная кнопка (Рис.5.6 поз.1) позволяет переключить режим отображения блока на экране на «Вид с вкладками». Изменим размеры блока для комфортного визуального отображения.



Рис.5.6

Осталось вставить наш настроенный программный блок «Переключатель» внутрь программного блока «Цикл» Оранжевой палитры. Программа готова! Загрузим её в робота и протестируем работу (Рис.5.7).



Рис.5.7

5.4. Оранжевая палитра, программный блок «Прерывание цикла»

Добавим в нашу программу движение. Сделаем следующее поле для выполнения задания:

11. Возьмем белый лист бумаги формата А4 или А3.
12. Нанесем на него последовательно, на равном расстоянии несколько цветных полос. Полосы можно наклеить из цветной бумаги, цветной изолянтной или нарисовать и закрасить.
13. Можете также загрузить подготовленное изображение и распечатать его на цветном принтере.
14. Последнюю полосу сделаем черного цвета (Рис.5.8).



Рис.5.8

Задача №10: необходимо написать программу прямолинейного движения робота, называющего цвета полос, над которыми он проезжает. При достижении черной полосы робот проговаривает «Stop» и останавливается.

За основу решения данной задачи возьмем программу, решающую Задачу №9. При решении Задачи №10 нам потребуется прервать выполнение цикла. Этой цели служит программный блок «Прерывание цикла» Оранжевой палитры. С помощью данного блока можно организовать выход из цикла, заданного параметром «Имя прерывания» (Рис.5.9 поз.1).



Рис.5.9

Попробуйте решить Задачу №10 самостоятельно, не подглядывая в решение.

Решение Задачи №10

1. Внутри цикла перед программным блоком «Переключатель» добавим программный блок «Рулевое управление», тем самым заставим нашего робота двигаться (Рис.5.10 поз.1). Во время движения робот будет проверять текущее состояние датчика цвета и произносить название цвета. Если полосы будут широкими, а робот будет двигаться медленно, то, возможно, он станет произносить название цвета более одного раза, так как проверка цвета будет происходить неоднократно. Если такое положение дел вас не устроит – увеличьте скорость робота, чтобы он быстрее проезжал цветные полосы.

2. В соответствии с условием задачи нам надо изменить поведение контейнера программного блока «Переключатель» для черного цвета.

3. В программном блоке «Звук» изменим звуковой файл «Black» на «Stop» (Рис.10 поз.2).

4. Добавим в контейнер программный блок, выключающий моторы (Рис.5.10 поз.3).

5. Нам требуется прервать выполнение программного блока «Цикл», чтобы завершить выполнение программы. Для этого поместим в контейнер программный блок «Превышение цикла» Оранжевой палитры (Рис.5.10 поз.4). У данного программного блока существует только одна настройка – название прерываемого цикла. В сложной программе со множеством циклов важно правильно устанавливать эту настройку, чтобы остановить выполнение нужного цикла (Рис.5.10 поз.5). В нашей программе за программным блоком отсутствуют другие программные блоки, поэтому программа завершится.

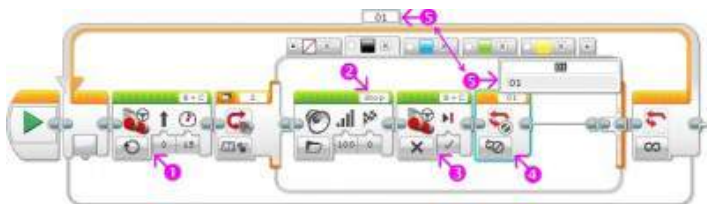


Рис.5.10

6. ДАТЧИКА ЦВЕТА. ПРОДОЛЖЕНИЕ

Введение:

На этом уроке мы продолжим изучать использование датчика цвета. Материал, изложенный ниже, очень важен для дальнейшего изучения курса робототехники. После того, как мы научимся использовать все датчики конструктора Lego mindstorms EV3, при решении множества практических задач, будем опираться на знания, полученные на этом занятии.

6.1. Датчик цвета – режим «Яркость отраженного света»

Итак, мы приступаем к изучению следующего режима работы датчика цвета, который называется «Яркость отраженного света». В этом режиме датчик цвета направляет поток красного света на близкорасположенный предмет или поверхность и измеряет количество отраженного света. Более темные предметы будут поглощать световой поток, поэтому датчик будет показывать меньшее значение, по сравнению с более светлыми поверхностями. Диапазон значений датчика измеряется от 0 (очень темный) до 100 (очень яркий). Данный режим работы датчика цвета используется во множестве задач по робототехнике, например, для организации движения робота по заданному маршруту вдоль черной линии, нанесенной на белое покрытие. При использовании этого режима рекомендуется располагать датчик таким образом, чтобы расстояние от него до исследуемой поверхности составляло примерно 1 см (Рис.6.1).



Рис.6.1

Перейдем к практическим занятиям: датчик цвета уже установлен на нашем роботе и направлен вниз к поверхности покрытия, по которому будет передвигаться наш робот. Расстояние между датчиком и полом соответствует рекомендуемому. Датчик цвета уже подключен к порту «2» модуля EV3. Давайте загрузим среду программирования, подключим робота к среде и для проведения замеров воспользуемся полем с цветными полосами, изготовлен-

ным нами для выполнения заданий Раздела 5.4 Урока №5. Установим робота, таким образом, чтобы датчик цвета расположился над белой поверхностью. «Страницу аппаратных средств» среды программирования переключим в режим «Просмотр портов» (Рис.6.2 поз.1). В этом режиме мы можем наблюдать все выполненные нами подключения. На Рис.6.2 отображено подключение к портам «В» и «С» двух больших моторов, а к порту «2» – датчика цвета.

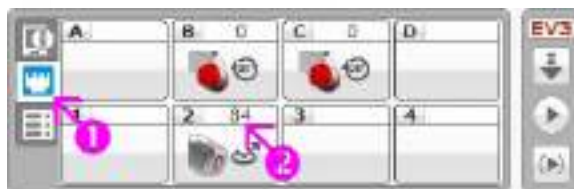


Рис.6.2

Для выбора варианта отображения показаний датчиков необходимо нажать на изображение датчика и выбрать нужный режим (Рис.6.3).



Рис.6.3

На Рис.6.2 поз.2 мы видим, что значение показания датчика цвета над белой поверхностью равно 84. В вашем случае может получиться другое значение, ведь оно зависит от материала поверхности и освещения внутри помещения: часть освещения, отражаясь от поверхности, попадает на датчик и влияет на его показания. Установив робота таким образом, чтобы датчик цвета расположился над черной полосой, зафиксируем его показания (Рис.6.4). Попробуйте измерить самостоятельно значения отраженного света над оставшимися цветными полосами. Какие значения у вас получились? Напишите ответ в комментарии к этому уроку.



Рис.6.4

Давайте теперь порешаем практические задачи.

Задача №11: необходимо написать программу движения робота, останавливающегося при достижении черной линии.

Решение:

Проведенный эксперимент показал нам, что при пересечении черной линии, значение датчика цвета в режиме «Яркость отраженного света» равняется 6. Значит, для выполнения Задачи №11 наш робот должен двигаться прямолинейно, пока искомое значение датчика цвета не станет меньше 7. Воспользуемся уже знакомым нам программным блоком «Ожидание» Оранжевой палитры. Выберем требуемый условию задачи режим работы программного блока «Ожидание» (Рис.6.5).

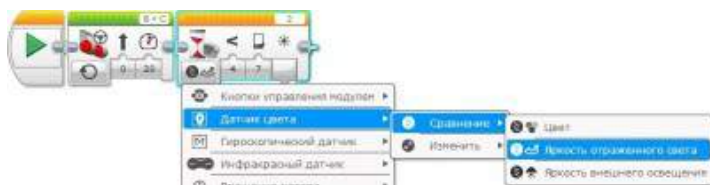


Рис.6.5

Необходимо также настроить параметры программного блока «Ожидание». Параметр «Тип сравнения» (Рис.6.6 поз.1) может принимать следующие значения: «Равно»=0, «Не равно»=1, «Больше»=2, «Больше или равно»=3, «Меньше»=4, «Меньше или равно»=5. В нашем случае установим «Тип сравнения» в значение «Меньше». Параметр «Пороговое значение» установим равным 7 (Рис.6 поз. 2).

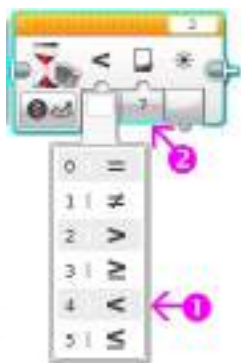


Рис.6.6

Как только установится значение датчика цвета меньше 7, что случится, когда датчик цвета окажется расположенным над черной линией, нам необходимо будет выключить моторы, остановив робота. Задача решена (Рис.6.7).



Рис.6.7

Для продолжения занятий нам понадобится изготовить новое поле, представляющее собой черную окружность диаметром примерно 1 метр, нанесенную на белое поле. Толщина линии окружности равняется 2-2,5 см. Для основы поля можно взять один лист бумаги размером A0 (841x1189 мм), склеить вместе два листа бумаги размером A1 (594x841 мм). На этом поле разметить линию окружности и закрасить её черной тушью. Можете также скачать макет поля, выполненный в формате Adobe Illustrator, а затем заказать его печать на баннерной ткани в типографии. Размер макета равен 1250x1250 мм. (Просмотреть скачанный ниже макет можно, открыв его в программе Adobe Acrobat Reader).



Рис.6.8

Данное поле пригодится нам для решения нескольких классических задач курса робототехники.

Задача №12: необходимо написать программу для робота, передвигающегося внутри круга, окантованного черной окружностью по следующему правилу:

- робот движется вперед прямолинейно;
- достигнув черной линии, робот останавливается;
- робот отъезжает назад на два оборота моторов;
- робот поворачивает вправо на 90 градусов;
- движение робота повторяется.

Знания, полученные на предыдущих уроках, помогут вам самостоятельно создать программу, решающую Задачу №12.

К работе датчика цвета в режиме «Яркость отраженного света» мы еще неоднократно вернемся, когда будем рассматривать алгоритмы движения вдоль черной линии. А пока разберем третий режим работы датчика цвета.

6.2. Датчик цвета – режим «Яркость внешнего освещения»

Режим работы датчика цвета «Яркость внешнего освещения» очень похож на режим «Яркость отраженного света», только в этом случае датчик не излучает освещение, а измеряет естественное световое освещение окружающей среды. Визуально данный режим работы датчика можно определить по слабо светящемуся синему светодиоду. Показания датчика изменяются от 0 (отсутствие света) до 100 (самый яркий свет). При решении практических задач, требующих измерения внешнего освещения, рекомендуется располагать датчик, так, чтобы датчик оставался максимально открытым и не загороживался другими деталями и конструкциями.

Давайте закрепим датчик цвета на нашем роботе так же, как мы крепили датчик касания в Уроке №4 (Рис.6.9). Подключим датчик цвета кабелем к порту «2» модуля EV3. Перейдем к решению практических задач.

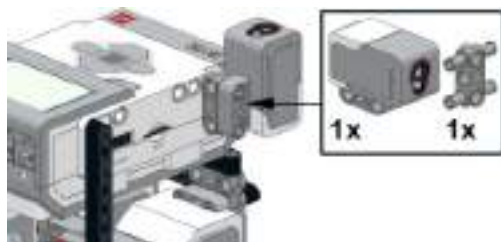


Рис.6.9

Задача №13: необходимо написать программу, изменяющую скорость движения нашего робота в зависимости от интенсивности внешнего освещения.

Чтобы решить эту задачу, нам надо узнать, как получать текущее значение датчика. А поможет нам в этом Желтая палитра программных блоков, которая называется «Датчики».

6.3. Желтая палитра – «Датчики»

Желтая палитра среды программирования Lego mindstorms EV3 содержит программные блоки, позволяющие получать текущие показания датчиков для дальнейшей обработки в программе. В отличие, например, от программного блока «Ожидание» Оранжевой палитры, программные блоки Желтой палитры сразу же передают управление к следующим за ними программным блокам.

Количество программных блоков Желтой палитры отличается в домашней и образовательной версии среды программирования. В домашней версии среды программирования отсутствуют программные блоки для датчиков, не входящих в домашнюю версию конструктора. Но, при необходимости, их можно самостоятельно подключить.



Желтая палитра. Домашняя версия среды программирования.

Образовательная версия среды программирования содержит программные блоки для всех датчиков, которые можно использовать с конструктором Lego mindstorms EV3.



Желтая палитра. Образовательная версия среды программирования.

Вернемся же к решению Задачи №13 и посмотрим, как можно получать и обрабатывать показания датчика цвета. Как мы уже знаем: диапазон значений датчика цвета в режиме «Яркость внешнего освещения» находится в пределах от 0 до 100. Такой же диапазон у параметра, регулирующего мощность моторов. Попробуем показанием датчика цвета регулировать мощность моторов в программном блоке «Рулевое управление».

Решение:

13. Считать текущее показание датчика цвета в режиме «Яркость внешнего освещения».



14. Подать полученное с датчика цвета значение на вход параметра «Мощность» программного блока «Рулевое управление»;

15. Повторять команды 1-2 в бесконечном цикле.

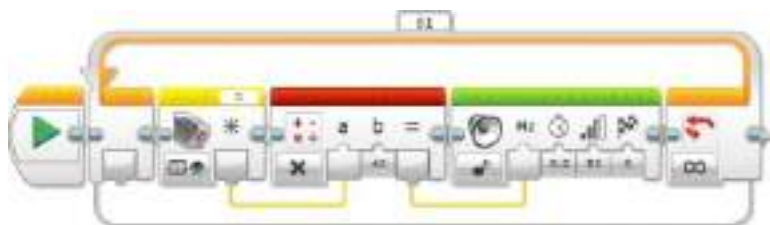


Рис.6.10

Давайте загрузим получившуюся программу в робота и запустим её на выполнение. Робот поехал медленно? Включим светодиодный фонарик и попробуем подносить его к датчику цвета на разном расстоянии. Что происходит с роботом? Закроем датчик цвета ладонью – что случилось в этом случае? Напишите ответы на эти вопросы в комментариях к уроку.

Задача – Bonus

Загрузите в робота и запустите на выполнение задачу, изображенную на рисунке ниже. Повторите эксперименты со светодиодным фонариком. Поделитесь впечатлениями в комментариях к уроку.



7. УЛЬТРАЗВУКОВОЙ ДАТЧИК

Введение:

Текущий урок мы посвятим изучению ультразвукового датчика. Данный датчик присутствует только в образовательной версии набора Lego mindstorms EV3. Тем не менее, пользователям домашней версии конструктора советуем тоже обратить внимание на данный урок. Возможно, что прочитав о назначении и использовании этого датчика, вы пожелаете его приобрести в дополнение к своему набору.

7.1. Изучаем ультразвуковой датчик

Главное назначение ультразвукового датчика, это определение расстояния до предметов, находящихся перед ним. Для этого датчик посылает звуковую волну высокой частоты (ультразвук), ловит обратную волну, отраженную от объекта и, замерив время на возвращение ультразвукового импульса, с высокой точностью рассчитывает расстояние до предмет



Рис.7.1

Ультразвуковой датчик может выдавать измеренное расстояние в сантиметрах или в дюймах. Диапазон измерений датчика в сантиметрах равен от 0 до 255 см, в дюймах - от 0 до 100 дюймов. Датчик не может обнаруживать предметы на расстоянии менее 3 см (1,5 дюймов). Так же он не достаточно устойчиво измеряет расстояние до мягких, тканевых и малообъемных объектов. Кроме режимов измерения расстояния в сантиметрах и дюймах датчик имеет специальный режим «Присутствие/слушать». В этом режиме датчик не излучает ультразвуковые импульсы, но способен обнаруживать импульсы другого ультразвукового датчика.

У нашего робота, собранного по инструкции small-robot-45544, ультразвуковой датчик уже закреплен впереди по ходу движения. Подключим его кабелем к порту «3» модуля EV3 и приступим к разбору практических примеров использования ультразвукового датчика.

Задача №14: написать программу, останавливающую прямолинейно движущегося робота, на расстоянии 15 см до стены или препятствия.

Для решения задачи воспользуемся уже знакомым нам программным блоком «Ожидание» Оранжевой палитры, переключив его в Режим: «Ультразвуковой датчик» – «Сравнение» – «Расстояние в сантиметрах» (Рис.7.2). Само решение будет похоже на решение Задачи №7.



Рис.7.2

Решение:

- Начать прямолинейное движение вперед (Рис.7.3 поз.1).
- Ждать, пока значение ультразвукового датчика не станет меньше 15 см (Рис.7.3 поз.2).
- Прекратить движение вперед (Рис.7.3 поз.3).



Рис.7.3

Задача решена!

Задача №15: написать программу для робота, держащего дистанцию в 15 см от препятствия.

Решение:

Поведение робота будет следующим:

16. при значении показания ультразвукового датчика больше 15 см робот будет двигаться вперед, стараясь приблизиться к препятствию;
17. при значении показания ультразвукового датчика меньше 15 см робот будет двигаться назад, стараясь удалиться от препятствия.

Мы уже знаем, что за организацию выбора выполняемых блоков в зависимости от условия отвечает программный блок «Переключатель» Оранжевой палитры. Установим для блока «Переключатель» режим «Ультразвуковой

датчик» – «Сравнение» – «Расстояние в сантиметрах» (Рис.7.4 поз1). Параметр «Тип сравнения» блока «Переключатель» установим в значение «Больше»=2, а «Пороговое значение» определим равным 15 (Рис.7.4 поз.2). Такие настройки программного блока «Переключатель» приведут к следующему поведению программы: При показаниях ультразвукового датчика больше 15 см будут выполняться программные блоки, помещенные в верхний контейнер (Рис.7.4 поз.3), в противном случае будут выполняться программные блоки, помещенные в нижний контейнер (Рис.7.4 поз.4).

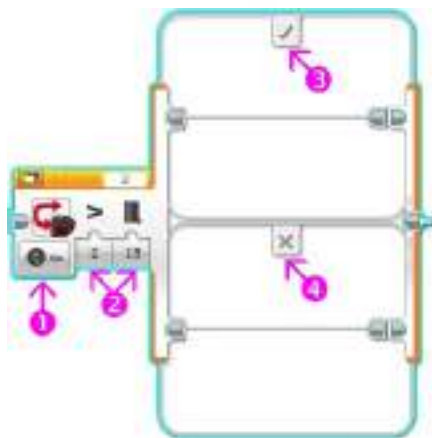


Рис.7.4

Поместим в эти контейнеры программные блоки, включающие движение вперед и назад. Для того чтобы программный блок «Переключатель» выполнялся многократно, поместим его внутрь программного блока «Цикл» Оранжевой палитры (Рис.7.5).



Рис.7.5

Загрузите получившуюся программу в робота и запустите ее на выполнение. Если перед роботом отсутствует препятствие, то он поедет вперед. Поднесите руку близко к ультразвуковому датчику, попробуйте отводить - приближать руку. Как ведет себя робот? Ждем ваши комментарии к этому уроку.

7.2. Робот-полицейский

Принцип работы ультразвукового датчика очень похож на радар, который применяется для измерения скорости движущихся автомобилей. Как радар узнаёт скорость автомобиля? Он измеряет расстояние до движущегося объекта, ждёт заданное небольшое время и повторяет измерение. Разность расстояний - это пройденный путь автомобиля. Разделив пройденный путь на время между двумя измерениями, можно найти скорость, с которой двигался объект измерения.

Давайте же научим и нашего робота работе радара!



Рис.7.6

Последовательность действий, выполняемых роботом, будет следующей:

- Робот ждёт появления в зоне контроля движущегося объекта;
- измеряет расстояние до объекта;
- ждёт 1 секунду;
- повторно измеряет расстояние до объекта;
- находит пройденное расстояние и сравнивает его с пороговым значением;
- выводит на экран результат и подает тревогу в случае превышения скорости.

Начнём создавать программу для нашего робота-полицейского.

1. С помощью программного блока «Ожидание» ждём появления объекта в зоне контроля робота (Рис.7.7 поз.1). Расстояние до объекта передаем в программный блок «Математика» (Рис. 7 поз. 4).

2. С помощью программного блока «Ожидание» ждем 1 секунду.

3. Второй раз снимаем показание ультразвукового датчика (Рис.7.7 поз.3) и передаем полученное значение в программный блок «Математика» (Рис.7.7 поз.4).

4. В программном блоке «Математика» находим расстояние, пройденное объектом измерения за 1 секунду. Полученное значение передаем в программный блок «Сравнение» (Рис.7.7 поз.5) и выводим на экран (Рис.7.7 поз.6).

5. С помощью программного блока «Сравнение» (Рис.7.7 поз.5) сравниваем пройденное расстояние с пороговым значением, равным 10. Результат сравнения двух чисел представляет собой логический вывод. Логический вывод может принимать одно из двух значений: «Да» или «Нет». Этот вывод

мы передаем в программный блок «Переключатель» (Рис.7.7 поз.7), настроив его на прием логических значений. Обратите внимание: шины данных, передающие логические значения, окрашены в зеленый цвет, в отличие от желтых шин данных, передающих числовые значения. (В дальнейшем мы подробнее ознакомимся с принципами обработки логических значений).

6. С помощью программного блока «Переключатель» мы организуем две ветки поведения программы в зависимости от скорости объекта. Если объект за 1 секунду приблизился к роботу, больше чем на 10 см, значит, будем считать его приближение критическим и подадим сигналы тревоги (Рис.7.7 поз.8). В противном случае будем считать, что объект движется медленно, в этом случае робот включит зеленую подсветку клавиш модуля EV3 и произнесёт «Окау».

7. В конце программы еще раз воспользуемся программным блоком «Ожидание» (Рис.7.7 поз.10) и «придержим» завершение программы на 5 секунд, чтобы успеть прочитать информацию на экране модуля EV3.

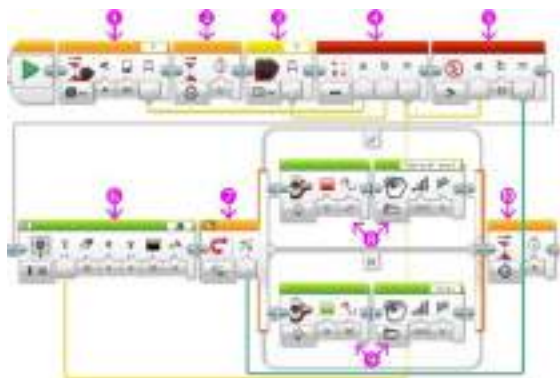


Рис.7.7

Загрузите программу в робота, расположите робота так, чтобы перед ним на расстоянии 60 сантиметров отсутствовали другие предметы, запустите программу на выполнение. Перемещайте в направлении к роботу игрушечный автомобиль или объемный предмет, наблюдайте за реакцией робота. Попробуйте изменять пороговые значения в программе. Как изменяется поведение робота? Опишите свои наблюдения в комментарии к этому уроку.

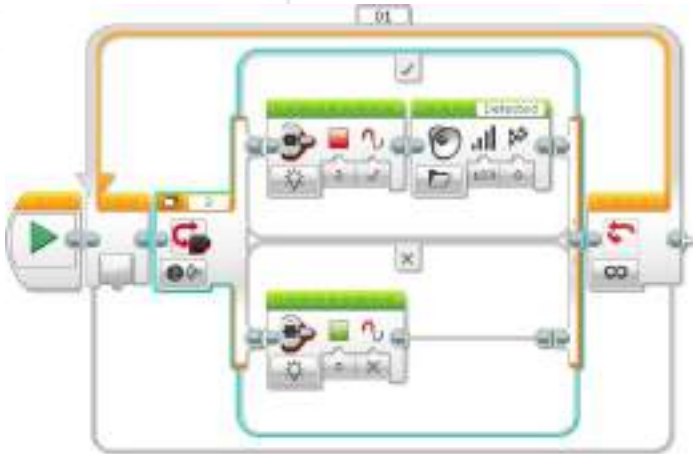
7.3. Ультразвуковой датчик – режим «Присутствие/слушать»

Как уже отмечалось выше, в этом режиме ультразвуковой датчик способен обнаруживать излучение другого ультразвукового датчика. Результатом обнаружения является логическое значение: «Да», если найдено ультразвуковое излучение, или «Нет», если ничего не найдено. Данный режим можно использовать, например, в состязаниях роботов-шпионов (описание режима уже говорит о том, что для его использования необходимо минимум два робота).

Задача № 16: необходимо написать программу, обнаруживающую другого робота, с работающим ультразвуковым датчиком.

Решение Задачи №16

Используя знания, полученные ранее, попробуйте самостоятельно разработать вариант решения Задачи №16. Предложите свой вариант решения.



8. ИНФРАКРАСНЫЙ ДАТЧИК

Введение:

Инфракрасный датчик входит домашнюю версию набора Lego mindstorms EV3. Это единственный датчик, который может применяться как самостоятельно, так и в паре с инфракрасным маяком, тоже являющимся частью домашнего набора. Следующие два урока мы посвятим изучению этих двух устройств, а также их взаимодействию между собой.

8.1. Изучаем инфракрасный датчик и инфракрасный маяк

Инфракрасный датчик (Рис.8.1) в своей работе использует световые волны, невидимые человеку – инфракрасные волны*. Такие же волны используют, например, дистанционные пульты управления различной современной бытовой техникой (телевизорами, видео и музыкальными устройствами). Инфракрасный датчик в режиме «Приближение» самостоятельно посылает инфракрасные волны и, поймав отраженный сигнал, определяет наличие препятствия перед собой. Еще два режима работы инфракрасный датчик реализует в паре с инфракрасным маяком (Рис.8.2). В режиме «Удаленный» инфракрасный датчик умеет определять нажатия кнопок инфракрасного маяка, что позволяет организовать дистанционное управление роботом. В режиме «Маяк» инфракрасный маяк посылает постоянные сигналы, по которым инфракрасный датчик может определять примерное направление и удаленность маяка, что позволяет запрограммировать робота таким образом, чтобы он всегда следовал в сторону инфракрасного маяка. Перед использованием инфракрасного маяка в него необходимо установить две батарейки AAA.



Рис.8.1



Рис.8.2

8.2. Инфракрасный датчик. Режим «Приближение»

Этот режим работы инфракрасного датчика похож на режим определения расстояния ультразвуковым датчиком. Разница кроется в природе световых волн: если звуковые волны отражаются от большинства материалов практически без затухания, то на отражение световых волн влияют не только материалы, но и цвет поверхности. Темные цвета в отличие от светлых сильнее поглощают световой поток, что влияет на работу инфракрасного датчика. Диапазон работы инфракрасного датчика также отличается от ультразвукового - датчик показывает значения в пределах от 0 (предмет находится очень близко) до 100 (предмет находится далеко или не обнаружен). Еще раз подчеркнем: инфракрасный датчик нельзя использовать для определения точного расстояния до объекта, так как на его показания в режиме «Приближение» оказывает влияние цвет поверхности исследуемого предмета. В свою очередь это свойство можно использовать для различия светлых и темных объектов, находящихся на равном расстоянии до робота. С задачей же определения препятствия перед собой инфракрасный датчик справляется вполне успешно.

Решим практическую задачу, похожую на Задачу №14 Урока №7, но, чтобы не повторяться, усложним условие дополнительными требованиями.

Задача №17: написать программу прямолинейно движущегося робота, останавливающегося перед стеной или препятствием, отъезжающего немного назад, поворачивающего на 90 градусов и продолжающего движение до следующего препятствия.

У робота, собранного по инструкции small-robot-31313, впереди по ходу движения установлен инфракрасный датчик. Соединим его кабелем с портом «3» модуля EV3 и приступим к созданию программы.

Рассмотрим программный блок «Ожидание» Оранжевой палитры, переключив его в Режим: «Инфракрасный датчик» – «Сравнение» – «Приближение» (Рис.8.3). В этом режиме программный блок «Ожидание» имеет два входных параметра: «Тип сравнения» и «Пороговое значение».

Настраивать эти параметры мы уже умеем.

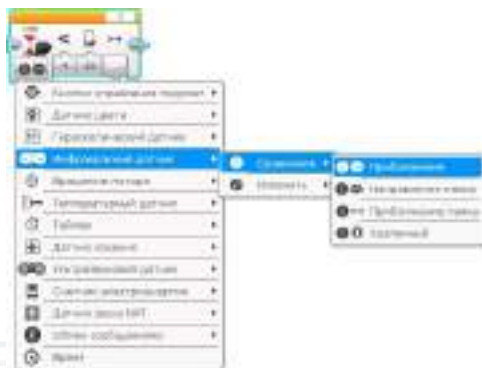


Рис.8.3

Решение:

- Начать прямолинейное движение вперед.
- Ждать, пока пороговое значение инфракрасного датчика станет меньше 20.
- Прекратить движение вперед
- Отъехать назад на 1 оборот двигателей
- Повернуть вправо на 90 градусов (воспользовавшись знаниями Урока №3, рассчитайте необходимый угол поворота моторов).
 - Продолжить выполнение пунктов 1-5 в бесконечном цикле.

Попробуйте решить Задачу № 17 самостоятельно, не подглядывая в решение.

Решение Задачи №17

А теперь для закрепления материала попробуйте адаптировать решение Задачи №15 Урока №7 к использованию инфракрасного датчика! Получилось? Поделитесь впечатлениями в комментарии к уроку.

8.3. Дистанционное управление роботом с помощью инфракрасного маяка.

Инфракрасный маяк, входящий в домашнюю версию конструктора Lego mindstorms EV3, в паре с инфракрасным датчиком позволяет реализовать дистанционное управление роботом. Познакомимся с маяком поближе:

- Пользуясь инфракрасным маяком, направляйте передатчик сигнала (Рис.8.5 поз.1) в сторону робота. Между маяком и роботом должны отсутствовать любые препятствия! Благодаря широкому углу обзора инфракрасный датчик уверенно принимает сигналы, даже если маяк располагается позади робота.

- На корпусе маяка расположены 5 серых кнопок (Рис.8.5 поз.2), нажатия которых распознает инфракрасный датчик, и передает коды нажатий в программу, управляющую роботом.

- С помощью специального красного переключателя (Рис.8.5 поз.3) можно выбрать один из четырех каналов для связи маяка и датчика. Сделано это для того, чтобы в непосредственной близости можно было управлять несколькими роботами.



Рис.8.4

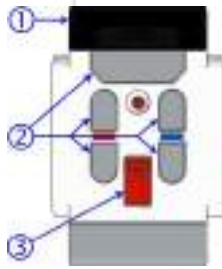


Рис.8.5

Задача №18: написать программу дистанционного управления роботом с помощью инфракрасного маяка.

Мы уже знаем, что для реализации возможности выбора выполняющихся блоков необходимо воспользоваться программным блоком «Переключатель» Оранжевой палитры. Установим режим работы блока «Переключатель» в «Инфракрасный датчик» – «Измерение» – «Удалённый» (Рис.8.6).

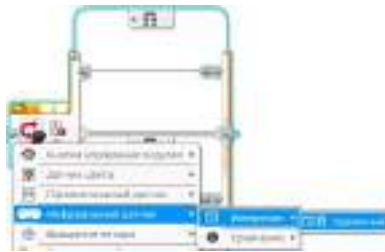


Рис.8.6

Для активации связи между инфракрасным датчиком и маяком необходимо установить правильное значение параметра «Канал» (Рис.8.7 поз.1) в соответствии с выбранным каналом на маяке. Каждому программному контейнеру блока «Переключатель» необходимо сопоставить один из возможных вариантов нажатия серых клавиш (Рис.8.7 поз.2). Заметьте: некоторые варианты включают одновременное нажатие двух клавиш (нажатые клавиши помечены красным цветом). Всего в программном блоке «Переключатель» в этом режиме можно обрабатывать до 12 различающихся условий (одно из условий должно быть выбрано условием по умолчанию). Добавляются программные контейнеры в блок «Переключатель» нажатием на «+» (Рис.8.7 поз.3).

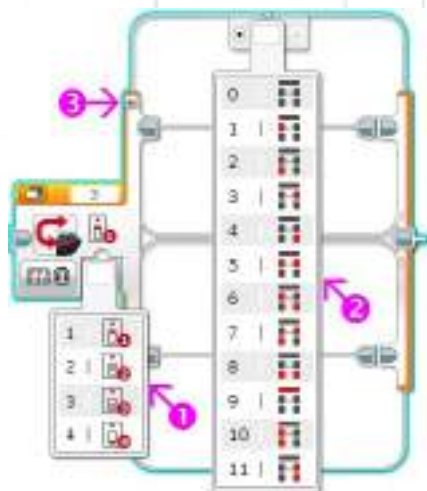


Рис.8.7

Предлагаем реализовать следующий алгоритм управления роботом:

- Нажатие верхней левой кнопки включает вращение левого мотора, робот поворачивает вправо (Рис.8.7 поз.2 значение: 1).
- Нажатие верхней правой кнопки включает вращение правого мотора, робот поворачивает влево (Рис.8.7 поз.2 значение: 3).
- Одновременное нажатие верхних левой и правой кнопок включает одновременное вращение вперед левого и правого мотора, робот движется вперед прямолинейно (Рис.8.7 поз.2 значение: 5).
- Одновременное нажатие нижних левой и правой кнопок включает одновременное вращение назад левого и правого мотора, робот движется назад прямолинейно (Рис.8.7 поз.2 значение: 8).
- Если не нажата ни одна кнопка маяка – робот останавливается (Рис.8.7 поз.2 значение: 0).

При разработке алгоритма дистанционного управления вы должны знать следующее: когда нажата одна из комбинаций серых кнопок - инфракрасный маяк непрерывно посылает соответствующий сигнал, если кнопки отпущены, то отправка сигнала прекращается. Исключение составляет отдельная горизонтальная серая кнопка (Рис.8.7 поз.2 значение: 9). Эта кнопка имеет два состояния: «ВКЛ» – «ВЫКЛ». Во включенном состоянии маяк продолжает посылать сигнал, даже если вы отпустите кнопку (о чём сигнализирует загорающийся зеленый светодиод), чтобы выключить отправки сигнала в этом режиме – нажмите горизонтальную серую кнопку еще раз.

Приступим к реализации программы:

Наш алгоритм дистанционного управления предусматривает 5 вариантов поведения, соответственно наш программный блок «Переключатель» будет состоять из пяти программных контейнеров. Займемся их настройкой.

1. Вариантом по умолчанию назначим вариант, когда не нажата ни одна кнопка (Рис.8.7 поз.2 значение: 0). Установим в контейнер программный блок «Независимое управление моторами», выключающий моторы «В» и «С».

2. В контейнер варианта нажатия верхней левой кнопки (Рис.8.7 поз.2 значение: 1) установим программный блок «Большой мотор», включающий мотор «В».

3. В контейнер варианта нажатия верхней правой кнопки (Рис.8.7 поз.2 значение: 3) установим программный блок «Большой мотор», включающий мотор «С».

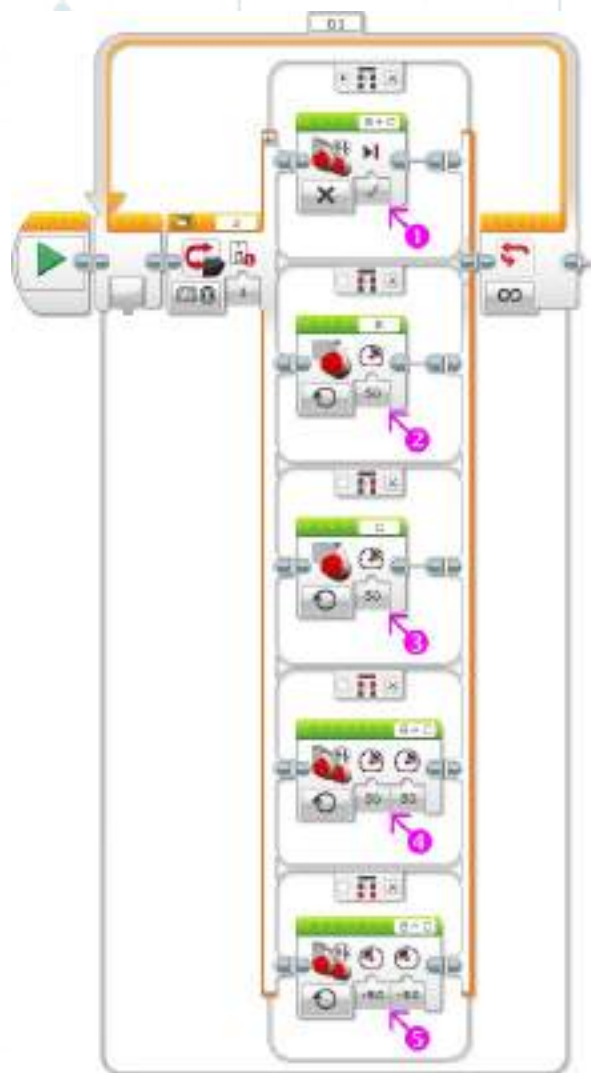
4. В контейнер варианта одновременного нажатия верхних левой и правой кнопок (Рис.8.7 поз.2 значение: 5) установим программный блок «Независимое управление моторами», включающий вращение моторов «В» и «С» вперед.

5. В контейнер варианта одновременного нажатия нижних левой и правой кнопок (Рис.8.7 поз.2 значение: 8) установим программный блок «Независимое управление моторами», включающий вращение моторов «В» и «С» назад.

6. Поместим наш настроенный программный блок "Переключатель" внутрь программного блока «Цикл».

По предложенной схеме попробуйте создать программу самостоятельно, не подглядывая в решение.

Решение Задачи №18



9. ИНФРАКРАСНОГО ДАТЧИК. ПРОДОЛЖЕНИЕ

Введение:

Наш очередной урок мы посвятим последнему режиму совместной работы инфракрасного датчика и инфракрасного маяка - режиму **«Маяк»**. В этом режиме инфракрасный датчик способен обнаруживать излучение инфракрасного маяка, а также определять примерное направление и расстояние до него.

9.1. Инфракрасный датчик. Режим «Маяк»

Для того, чтобы использовать инфракрасный маяк в этом режиме, следует, нажав отдельную горизонтальную серую кнопку, перевести маяк в режим непрерывного излучения сигнала (Рис.9.1 поз.1). При этом на инфракрасном маяке загорится зеленый индикатор.

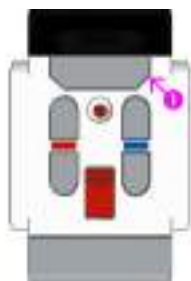


Рис.9.1

Давайте рассмотрим программный блок «Инфракрасный датчик» Желтой палитры в режиме «Измерение» – «Маяк». (Рис.9.2) В этом режиме программный блок имеет один входной параметр, определяющий номер канала работы инфракрасного маяка (Рис. 2 поз.1), а также три выходных параметра: «Направление» (Рис.9.2 поз.2), «Приближение» (Рис. 2 поз.3) и «Обнаружено» (Рис.9.2 поз.4).



Рис.9.2

- Параметр «Обнаружено» выдает логическое значение «Да» – если инфракрасный маяк обнаружен датчиком, и «Нет» – в противном случае.

- Параметр «Приближение» выдает числовое значение в диапазоне от 0 до 100. Значение, равное 0, означает, что инфракрасный маяк находится очень близко. Значения, меньшие 100, сигнализируют о том, что инфракрасный маяк уверенно обнаруживается датчиком и находится в относительной удаленности от робота. Значение, равное 100 говорит нам о том, что инфракрасный маяк находится очень далеко или не обнаружен датчиком.

- Параметр «Направление» выдает числовое значение в диапазоне от -25 до 25. Значение, равное 0, означает, что инфракрасный маяк находится строго напротив инфракрасного датчика. Отрицательные значения свидетельствуют об отклонении инфракрасного датчика в левую сторону от направления датчика (против часовой стрелки), а положительные – в правую сторону от направления датчика (по часовой стрелке) (Рис.9.3).

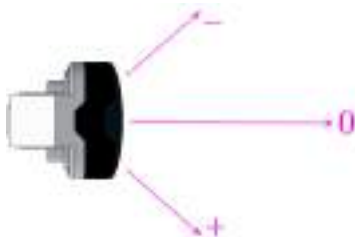


Рис.9.3

Попробуем применить полученные знания на практике и создать программу для робота, позволяющую ему находить инфракрасный маяк и следовать за ним.

9.2. Поиск инфракрасного маяка

На первом этапе приступим к созданию программы поиска роботом инфракрасного маяка.

Задача № 19: написать программу для робота, вращающегося вокруг своей оси и останавливающегося в направлении инфракрасного датчика.

Решение:

1. Используя программный блок «Независимое управление моторами», начать вращение робота вокруг своей оси против часовой стрелки (Рис.9.4 поз.1).

2. Используя программный блок «Ожидание» в режиме «Инфракрасный датчик» – «Сравнение» – «Приближение маяка» (Рис.9.4 поз.2) с пороговым значением равным 80 (Рис.9.4 поз.3), ожидаем, пока робот не обнаружит инфракрасный маяк (значение параметра «Приближение» станет меньше 100).

3. Так как наш робот вращается против часовой стрелки, то, когда инфракрасный датчик обнаружит маяк, его параметр «Направление» примет отрицательное значение. Поэтому, следующий программный блок «Ожидание»

в режиме «Инфракрасный датчик» – «Сравнение» – «Направление маяка» (Рис.9.4 поз.4) даст возможность роботу вращаться до тех пор, пока робот не окажется напротив инфракрасного маяка (значение параметра «Пороговое значение» превысит 0 (Рис.9.4 поз.5)).

4. Так как наш робот, вращаясь с большой скоростью, может повернуть чуть больше в результате сил инерции, то, на малой скорости, используя следующие два программных блока, повернем робота по часовой стрелке (Рис.9.4 поз.6, 7).

5. Выключим моторы робота (Рис.9.4 поз.8).

Загрузите программу в робота, поверните робота спиной к инфракрасному маяку, включите маяк и запустите программу на выполнение. Наш робот должен остановиться напротив маяка. Получилось? Для того чтобы лучше понять принцип работы нашей программы, попробуйте, изменив направление начального вращения робота (по часовой стрелке), отредактировать необходимые программные блоки и добиться правильной её работы.

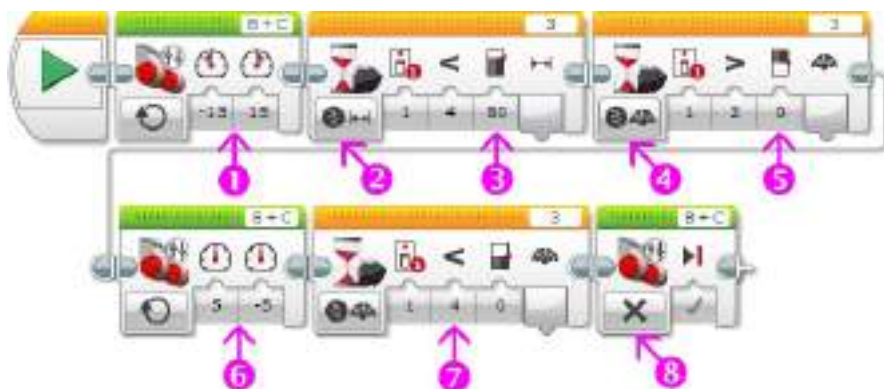


Рис.9.4

9.3. Следование за инфракрасным маяком

Задача №20: написать программу следования робота за инфракрасным маяком.

Решение:

Наш робот научился уверенно определять направление инфракрасного маяка и поворачивать в его сторону. Осталось только доехать до него и остановиться напротив. Решить эту задачу можно различными способами. Предлагаю вам воспользоваться возможностями, предоставляемыми программным блоком «Инфракрасный датчик» Желтой палитры, который мы рассмотрели в первой части сегодняшнего урока. Как поведет себя наш робот, если в бесконечном цикле мы подадим значение параметра «Приближение» (Рис.9.5 поз.1) программного блока «Инфракрасного датчика» на вход параметра «Мощность» (Рис.9.5 поз.2) программного блока «Рулевое управление» Зеленой палитры?

Поместим включенный маяк непосредственно перед роботом. Загрузим получившуюся программу (Рис.9.5) в робота и запустим её на выполнение. Приближаясь к маяку, наш робот будет постепенно замедляться. Подъехав вплотную к маяку, робот остановится. Если отодвинуть маяк, то робот снова устремится в его сторону.



Рис.9.5

К сожалению, пока наш робот не умеет изменять направление движения. Если инфракрасный маяк переместить влево - вправо от направления движения робота, то, наш робот проедет мимо. Давайте научим движущегося робота поворачивать в сторону маяка. Для этого возьмем параметр «Направление» (Рис.9.6 поз.1) программного блока «Инфракрасный датчик», умножим его на 2 (Рис.9.6 поз.2) и подадим на вход параметра «Рулевое управление» (Рис.9.6 поз.3) программного блока «Рулевое управление». Для чего нам потребовалось умножение? Диапазон значений параметра «Направление» от -25 до 25. Получая значения из этого диапазона, робот будет поворачивать на недостаточный угол – умножение же расширяет диапазон значений от -50 до 50, что позволит роботу уверенно следовать за перемещениями маяка.

Снова поместим включенный инфракрасный маяк напротив робота. Загрузим исправленную программу (Рис.9.6) в робота и запустим её на выполнение. Как только робот устремится в сторону инфракрасного маяка,

начнем перемещать маяк влево или вправо от направления движения робота. Наш робот будет уверенно поворачивать в сторону маяка.



Рис.9.6

Теперь наша программа отлично справляется с задачей следования за инфракрасным маяком. Но, если выключить маяк, то робот начинает вести себя непредсказуемо. Ведь мы никак не учитываем показание параметра «Обнаружено» (Рис.9.7 поз.1) программного блока «Инфракрасный датчик». Если робот теряет маяк, то значение параметра «Обнаружено» становится равным «Нет». Подадим значение параметра «Обнаружено» на вход программного блока «Переключатель», установленного в режим «Логическое значение». В контейнер логического значения «Нет» установим программный блок «Прерывание цикла» Оранжевой палитры (Рис.9.7 поз.2). Параметр «Имя прерывания» программного блока «Прерывание цикла» должно соответствовать имени цикла.

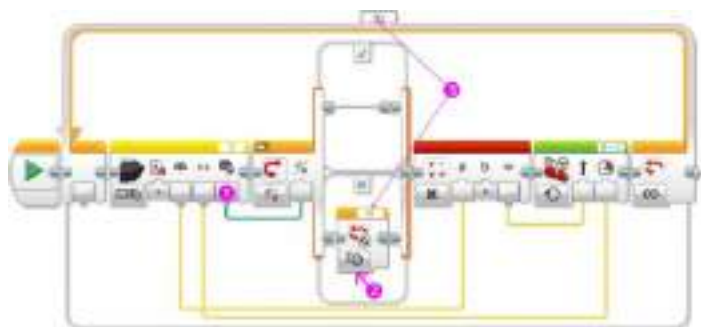


Рис.9.7

Протестируем получившуюся программу (Рис.9.7) – стоит выключить инфракрасный маяк, как программный блок «Прерывание цикла» фактически остановит выполнение нашей программы.

9.4. Поиск и следование за инфракрасным маяком

Задача №21: написать программу поиска и следования за инфракрасным маяком.

Решение:

Настало время объединить знания, полученные в разделе 9.2. Поиск инфракрасного маяка и 9.3. Следование за инфракрасным маяком. Внутри бесконечного цикла сначала поместим программу поиска маяка (Рис.9.4), а затем - программу следования за маяком (Рис.9.7). Получившуюся программу (Рис.9.8) загрузим в робота и запустим на выполнение. Сначала наш робот будет вращаться на месте, пока не повернется в сторону инфракрасного маяка, а затем устремится в его сторону. Если маяк выключить, то робот остановится и снова начнет вращаться на месте, пока не обнаружит инфракрасный маяк.

Задача решена!

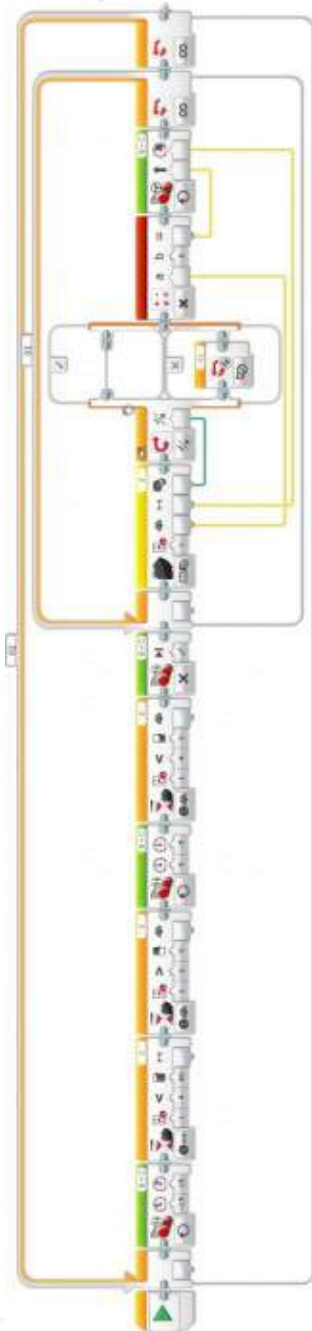


Рис.9.8

10. ГИРОСКОПИЧЕСКИЙ ДАТЧИК

Введение:

Завершающий урок цикла "Первые шаги" посвятим изучению гироскопического датчика. Данный датчик, как и ультразвуковой, присутствует только в образовательной версии набора Lego mindstorms EV3. Тем не менее, пользователям домашней версии конструктора советуем тоже обратить внимание на данный урок. Возможно, что прочитав о назначении и использовании этого датчика, вы пожелаете его приобрести в дополнение к своему набору.

10.1. Изучаем гироскопический датчик

Гироскопический датчик (**Рис.10.1**) предназначен для измерения угла вращения робота или скорости вращения. Сверху на корпусе датчика нанесены две стрелки, обозначающие плоскость, в которой работает датчик. Поэтому важно правильно установить датчик на робота. Также для более точного измерения крепление гироскопического датчика должно исключать его подвижность относительно корпуса робота. Даже во время прямолинейного движения робота гироскопический датчик может накапливать погрешность измерения угла и скорости вращения, поэтому непосредственно перед измерением следует осуществить сброс в 0 текущего показания датчика. Вращение робота против часовой стрелки формирует отрицательные значения измерений, а вращение по часовой стрелке – положительные.



Рис.10.1

Рассмотрим программный блок «Гироскопический датчик» (Рис.10.2) Желтой палитры. Этот программный блок имеет три режима работы: «Измерение», «Сравнение» и «Сброс». В режиме «Измерение» можно измерить «Угол», «Скорость» или одновременно «Угол и скорость».

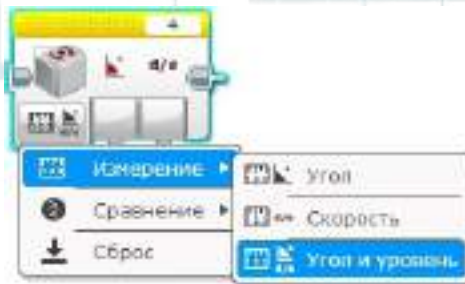


Рис.10.2

Давайте закрепим гироскопический датчик на нашем роботе (Рис.10.3), подсоединим его кабелем к порту 4 модуля EV3 и рассмотрим примеры использования.

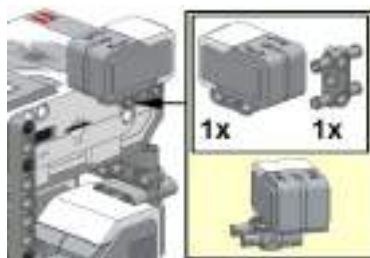


Рис.10.3

Задача №22: написать программу движения робота по квадрату с длинной стороны квадрата, равной длине окружности колеса робота.

Решение:

- Перед началом движения сбросим датчик в 0, используя программный блок «Гироскопический датчик» Желтой палитры;
 - Мы уже знаем: чтобы проехать прямолинейно требуемое расстояние – необходимо, воспользовавшись программным блоком «Независимое управление моторами», включить оба мотора на 1 оборот.
 - Для поворота робота на 90 градусов в этот раз воспользуемся гироскопическим датчиком:
 - Используя программный блок «Независимое управление моторами», заставим робота вращаться вправо вокруг своей оси.
 - Используя программный блок «Ожидание» в режиме «Гироскопический датчик», будем ждать, пока значение угла поворота не достигнет 90 градусов.
 - Выключим моторы.
 - Используя программный блок «Цикл» в режиме «Подсчет», повторим шаги 1-3 четыре раза.
- Попробуйте решить Задачу №22 самостоятельно, не подглядывая в решение.

Решение Задачи №22



ЗАКЛЮЧЕНИЕ

Происходящие в настоящее время изменения социально-экономической сферы требуют кардинального пересмотра системы приоритетов не только в экономике, но и в социальной политике современной России.

Ориентация на модель социально ориентированной рыночной экономики предусматривает формирование нового поколения специалистов высокого класса, способных самостоятельно выстраивать частно-государственное партнерство, производить конкурентоспособную на мировых рынках продукцию и определять структуру ее потребления. Соревновательное содержание Программы «Технология робототехники» отвечает духу времени и обеспечивает ей статус базового образовательного модуля всех существующих центров детского и юношеского технического творчества. Задания для соревнований в рамках программы сформированы так, чтобы участники имели возможность показать свой профессионализм и практические навыки в той или иной отрасли (механика, электроника, схемотехника автоматизированных устройств, работа с датчиками, управление приводами, работа с технической документацией и пр.).

В результате реализации программы «Технология робототехники» в системе дополнительного образования будет способствовать возвращению престижа инженерных профессий в глазах молодежи, формированию у ее участников широкого спектра профессиональных навыков и личных качеств – умения работать в команде, ответственности, целеустремленности. Помимо этого, Программа предоставит возможности для самореализации представителям всех слоев населения вне зависимости от образовательного, материального и социального уровня.

ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. Аленина, Т.И. Образовательная робототехника во внеурочной деятельности младших школьников в условиях введения ФГОС НОО: пособие для учителя / сост.: Аленина Т.И., Енина Л.В., Колотова И.О., Сичинская Н.М., Смирнова Ю.В., Шаульская Е.Л. – Челябинский Дом печати, 2012. – 208 с.
2. Зайцева, Н. Н. Образовательная робототехника в начальной школе: пособие для учителя / Зайцева Н.Н., Зубова Т.А., Копытова О.Г., Подкорытова С.Ю. – Челябинск: Обл. центр информ. и мат.-тех. обесп. ОУ Челяб. обл. – 192 с.
3. Косенок, С.М., Рассказов Ф.Д. Современные педагогические технологии в учебном процессе вуза: уч.-метод. пособие / С.М. Косенок, Ф.Д. Рассказов. – Сургут, 2018.
4. Мирошина, Т. Ф. Образовательная робототехника в начальной школе: пособие для учителя / Мирошина Т.Ф., Соловьева Л.Е., Могилева А.Ю., Перфириева Л.П. – Челябинск: Взгляд. – 2011. – 150 с.
5. Мирошина, Т.Ф. Образовательная робототехника на уроках информатики и физики в средней школе: пособие для учителя / Мирошина Т.Ф., Соловьева Л.Е., Могилева А.Ю., Перфириева Л.П. – Челябинск: Взгляд, 2011. – 150 с.
6. Перфириева, Л.П., Трапезникова Т.В., Шаульская Е.Л., Выдрина Ю.А. Образовательная робототехника во внеурочной деятельности: методическое пособие / Перфириева Л.П., Трапезникова Т.В., Шаульская Е.Л., Выдрина Ю.А. – Челябинск: Взгляд. – 2011. – 94 с.
7. Процесс формирования конструкторской компетентности студентов педагогического направления в области робототехники. Буеров А.С., Рассказов Ф.Д. В сборнике: Тенденции развития образования: педагог, образовательная организация, общество – 2018. Материалы Всероссийской научно-практической конференции. Под редакцией Ж.В. Мурзиной, – 2018. – С.25-26.
8. Рогов Ю.В. Робототехника для детей и их родителей: уч.-метод. пособие / Ю.В. Рогов. – Челябинск, 2012. – 72 с.: ил.
9. Сагритдинова Н.А. Fischertechnik – основы образовательной робототехники: уч.-метод. пособие / Н.А. Сагритдинова. – Челябинск, 2012. – 40 с.: ил.
10. Федеральный закон «О некоммерческих организациях» от 12.01.1996 N 7-ФЗ: в действующей редакции от 14.07.2013.

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕТОК

Учебное издание

Буеров Алексей Сергеевич

**ТЕХНОЛОГИИ РОБОТОТЕХНИКИ
первые шаги**

Учебно-методическое пособие

В авторской редакции

Подписано в печать 10.12.2018. Формат 60x84/16

Печать оперативная. Усл. п.л. 4,9

Тираж 500 экз. Заказ №94–18–07.

Отпечатано с готового оригинал-макета в издательстве ЗЕБРА
432072, Россия, г. Ульяновск, ул. Жуковского, 83.

ISBN 978-5-6042184-1-9



9 785604 218419