

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаров Алексей Владимирович
Должность: И.о. директора технологического колледжа
Дата подписания: 19.01.2024 10:54:36
Уникальный программный ключ:
7f14295cc243663512787b1375f9c1207c9a75d

Приложение к ППССЗ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный аграрный университет –
МСХА имени К.А. Тимирязева»
(ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева)
ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по дисциплине «ОП.11 Основы программирования робототехнических устройств»

специальность: 44.02.03 Педагогика дополнительного образования

форма обучения: очная

Москва, 2022

СОДЕРЖАНИЕ

1. Пояснительная записка	4
2. Тематика и задания практической работы	6
3. Критерии оценки	46
4. Список рекомендуемой литературы	47

1. Пояснительная записка

1.1. Методические указания для обучающихся по выполнению заданий практической работы по ОП.11 Основы программирования робототехнических устройств, предназначены для реализации Федерального государственного образовательного стандарта среднего профессионального образования по специальности 44.02.03 Педагогика дополнительного образования.

Дисциплина входит в профессиональный цикл, в состав общепрофессиональных дисциплин.

Учебная дисциплина ОП.11 Основы программирования робототехнических устройств в структуре ППССЗ входит в профессиональный цикл, в состав общепрофессиональных дисциплин.

ОК 3. Оценивать риски и принимать решения в нестандартных ситуациях.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ПК 1.3. Демонстрировать владение деятельностью, соответствующей избранной области дополнительного образования.

В результате освоения учебной дисциплины обучающийся должен **уметь:**

- уметь использовать основные алгоритмические конструкции для решения задач;
- уметь конструировать различные модели, использовать созданные программы;
- уметь применять полученные знания в практической деятельности

В результате освоения учебной дисциплины обучающийся должен **знать:**

- конструктивные особенности различных роботов;
- основные алгоритмические конструкции, этапы решения задач с использованием ЭВМ;
- конструктивные особенности различных моделей, сооружений и механизмов

Количество часов отведенное на выполнение практических работ обучающихся 38 часов.

1.2. Практическая работа обучающихся

Наименование темы	Объем часов	Вид работы	Формируемые результаты освоения
Тема 2. Устройство роботов.	16	Практическое занятие №1. Знакомство с вычислительными возможностями робота. Практическое занятие №2. Изучаем датчик касания	ОК 3,4,5,84 ПК 1.3
Тема 3. Манипуляционные системы роботов.	8	Практическое занятие №3. Структурный и кинематический анализ манипуляторов.	ОК 3,4,5,84 ПК 1.3
Тема 6. Динамика роботов. Основные принципы организации движения роботов.	14	Практическое занятие №4. Знакомство с конструктором Практическое занятие № 5. Программирование движения робота	ОК 3,4,5,84 ПК 1.3
Итого	38		

2. Тематика и задания практической работы

Практическая работа № 1

Знакомство с вычислительными возможностями робота

Цель работы —ознакомиться с вычислительными возможностями робота.

Введение:

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.

1.1. 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.

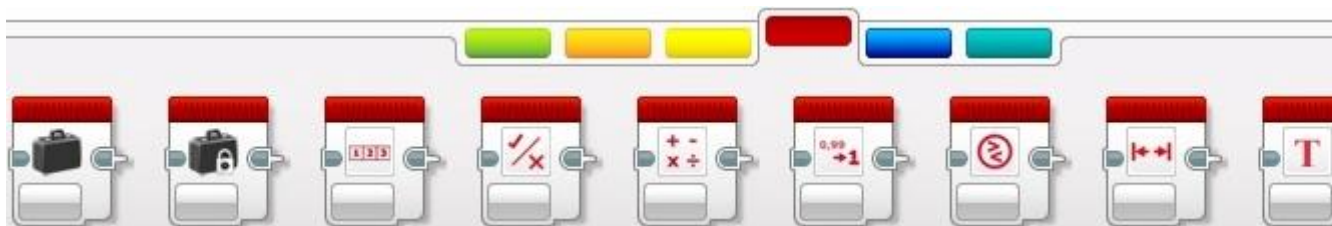


Рис.1

1.2. 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. 2) позволяет создавать ячейку памяти для хранения одного из пяти типов данных (Рис. 2 поз. 1). Требуемое значение записывается в ячейку на этапе создания программы (Рис. 2 поз. 2) и остается неизменным во время выполнения всей программы. Для получения значения, записанного в блок "Константа" используется "Вывод" (Рис. 2 поз. 3). Подробнее с

извлечением данных из программных блоков мы познакомимся ниже при решении практической задачи Урока №3.

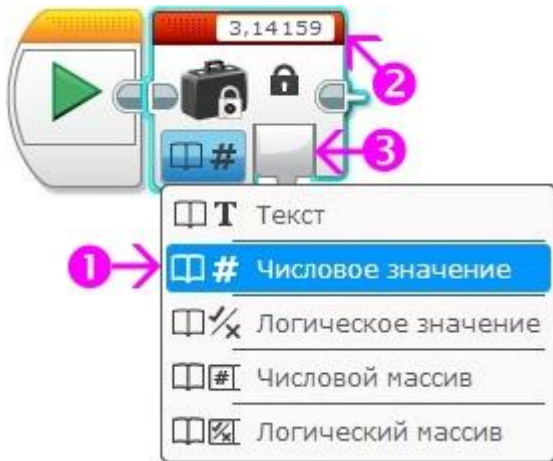


Рис. 2

В отличие от программного блока "Константа" - в блоке "Переменная" присутствуют два режима "Считывание" и "Записать" (Рис. 3 поз. 1). Перед первым использованием необходимо задать имя переменной, выбрав параметр блока "Добавить переменную" (Рис. 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 поз. 3).

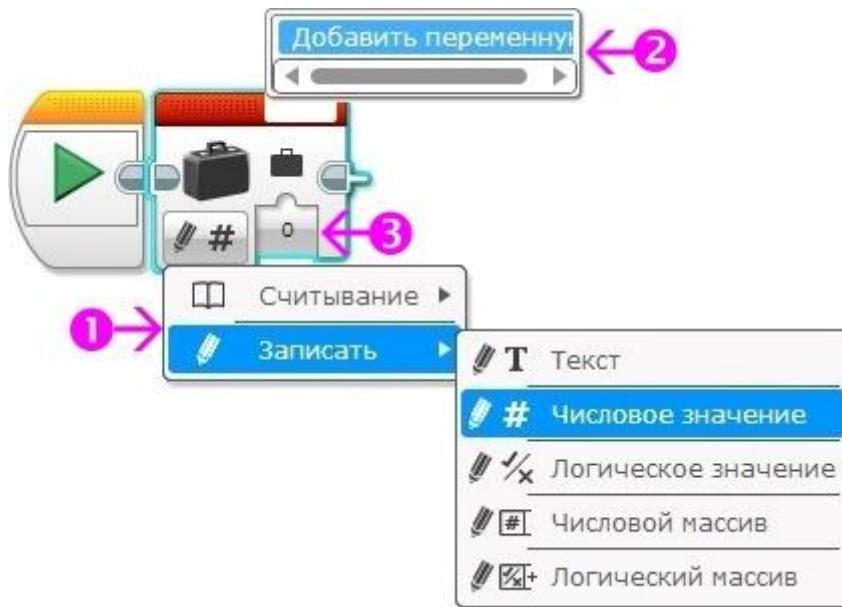


Рис. 3

1.3. Блок математика, блок округление

Для выполнения математических вычислений служит программный блок "Математика". Он позволяет выполнить выбранную математическую операцию (Рис. 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".

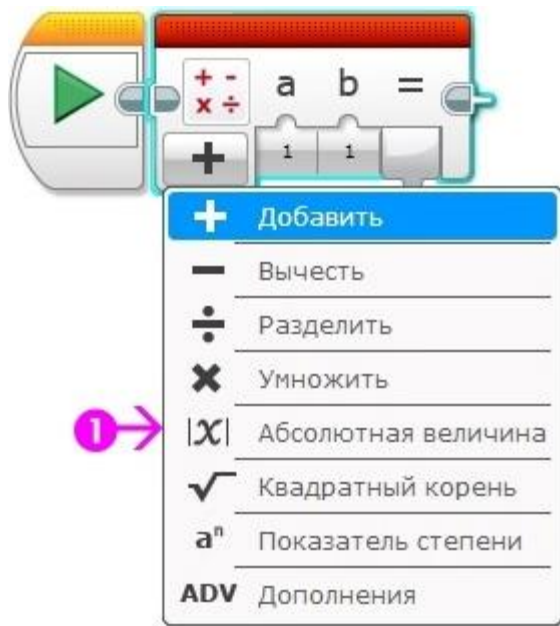


Рис. 4

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. 5 поз. 1) можно вписать любую произвольную формулу, производящую вычисления с этими параметрами.

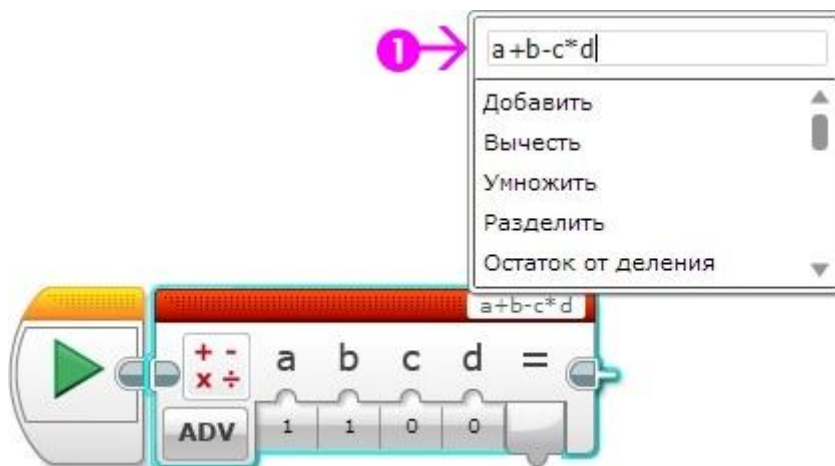


Рис. 5

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. 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. 6

1.4. 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** ($\approx 3,14159$) 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. 7

Приступим к созданию программы:

1. Используя программный блок "Константа", заведем в программу постоянное число



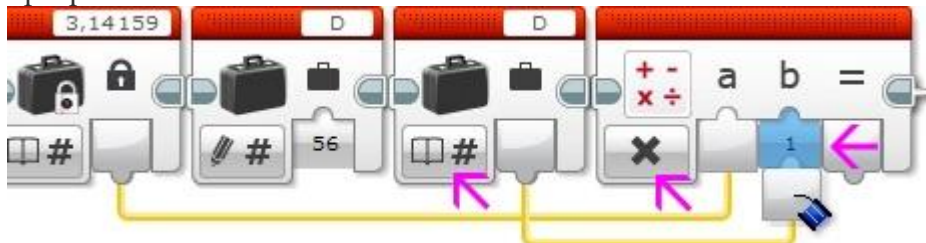
Пи, равное примерно **3,14159**.

2. 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

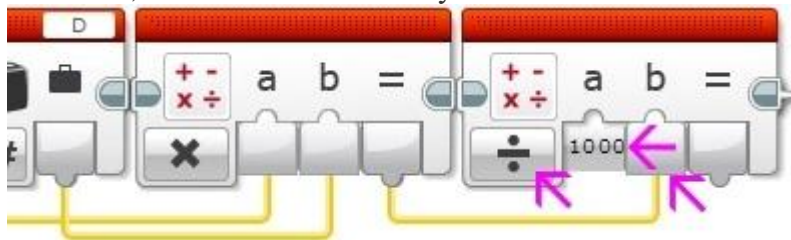
самостоятельно измерьте диаметр и внесите значение в программный блок).



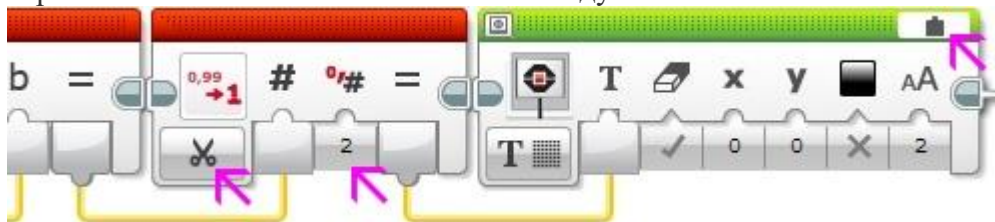
3. Используя программный блок "Математика", умножим значение блока "Константа" на значение переменной **D**. Для передачи значения из переменной **D** в программный блок "Математика" используем второй программный блок "Переменная" в режиме "Считывание"! (Для передачи значений между программными блоками используются шины данных. Чтобы установить шину данных, необходимо "потянуть" выходной параметр одного программного блока и "присоединить" его к входному параметру другого программного блока)



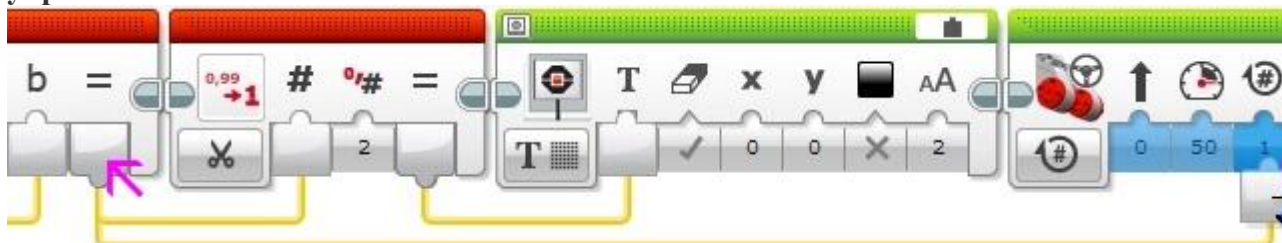
4. Используя программный блок "Математика", разделим значение пути (1000 мм) на значение, полученное в шаге 3.



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



6. 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

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. 8** - каждое колесо при этом проедет ровно половину окружности с диаметром, равным расстоянию между центрами колес (**красная линия на Рис. 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.7 поз. 2**), то решим требуемую задачу.

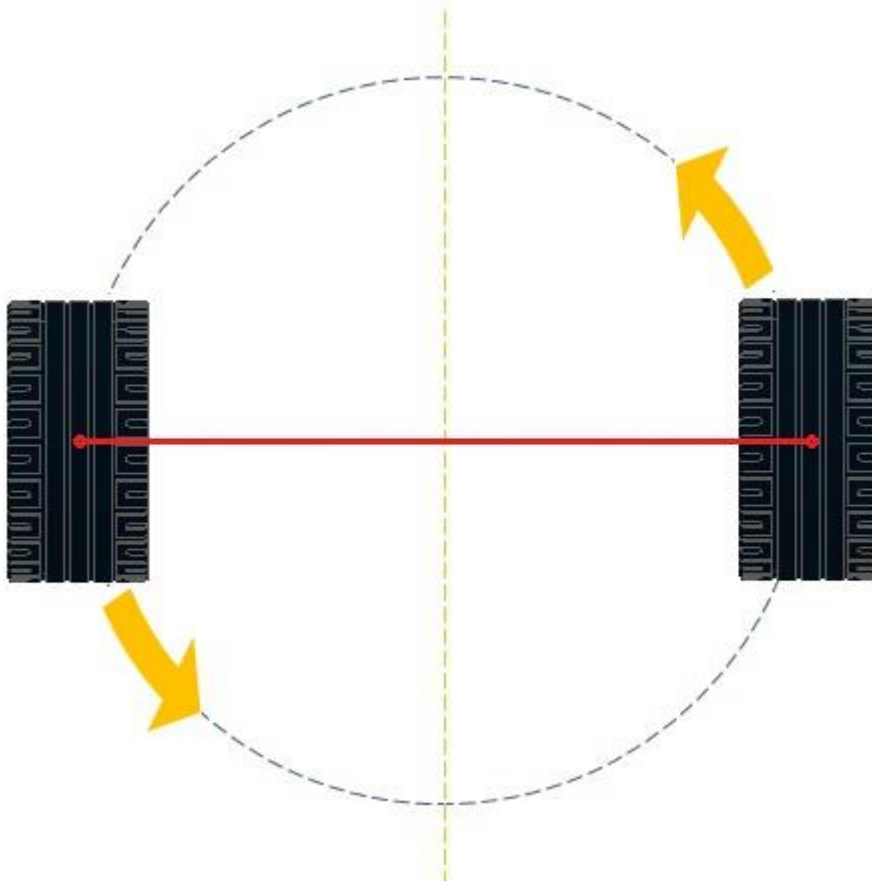


Рис. 8

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

Практическое занятие №2

Изучаем датчик касания

Цель работы —ознакомиться с датчиком 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.

2.1. 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.

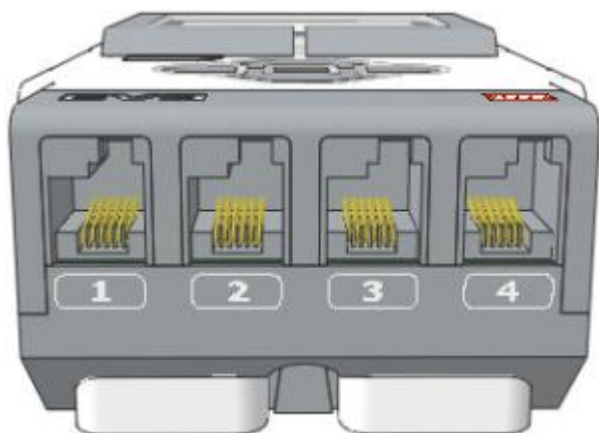


Рис. 1

Первым датчиком, который мы изучим, будет датчик касания (Рис. 2).

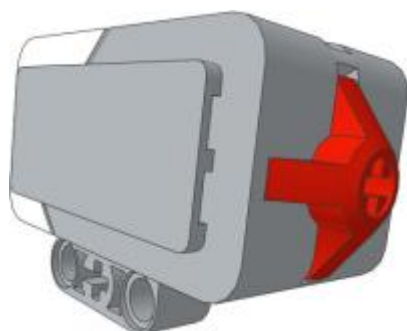


Рис. 2

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

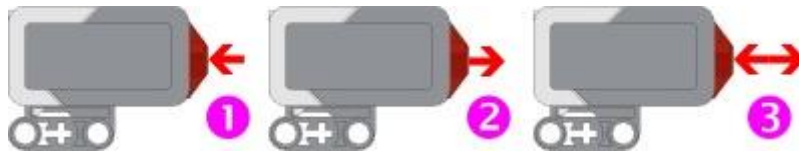


Рис. 3

2.2. 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)

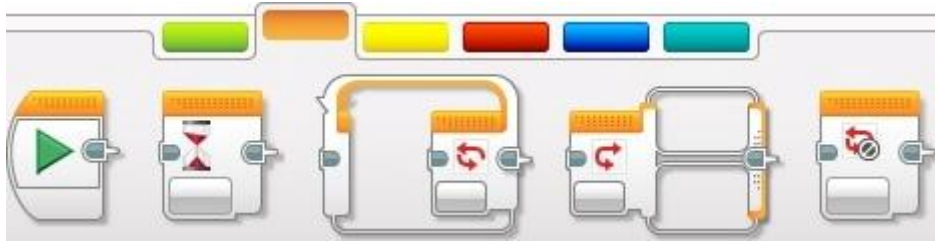


Рис. 4

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 контейнеров.
- 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.

2.3. Оранжевая палитра, программный блок "Ожидание"

Перед тем, как приступить к решению практических задач, давайте закрепим датчик касания на нашем роботе, как показано на **Рис. 5**, и подключим его кабелем к порту **"1"** модуля EV3.

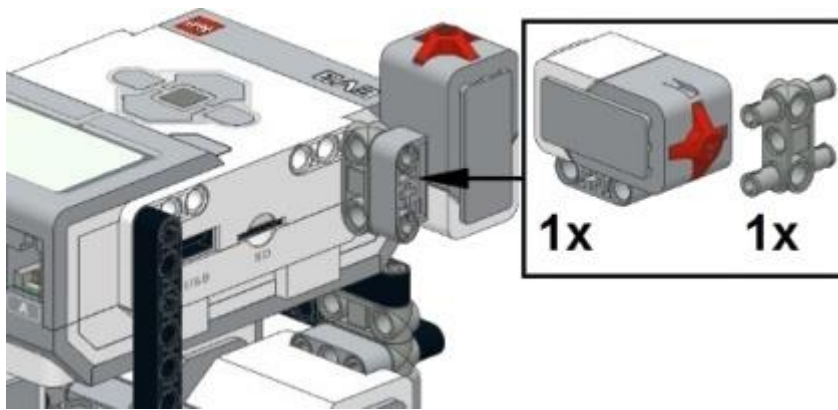


Рис. 5

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. 6).

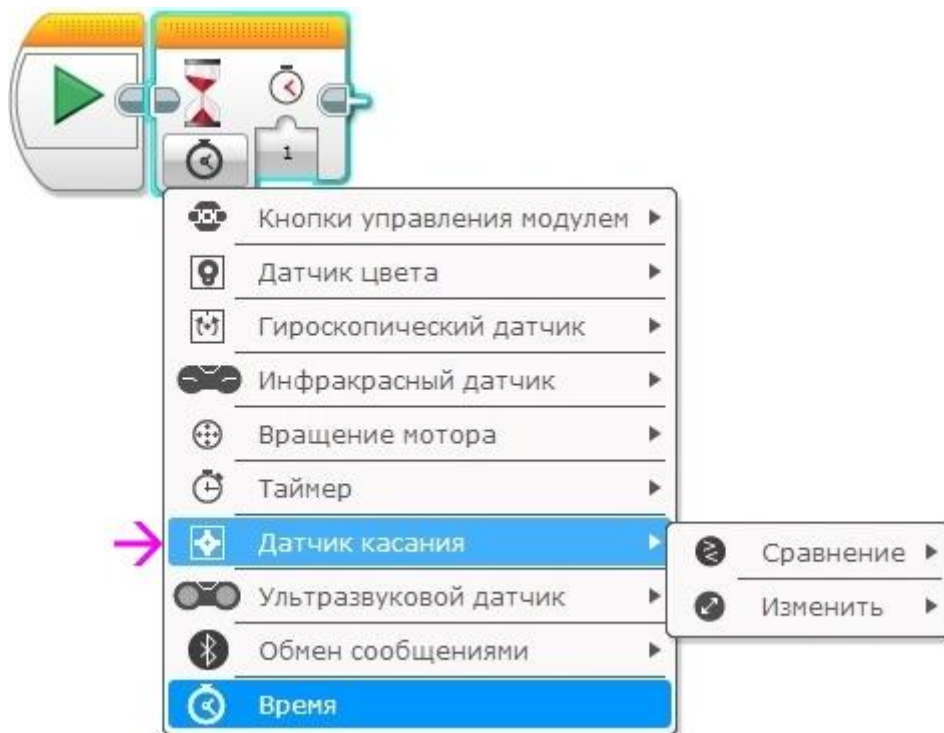


Рис. 6

Как можно увидеть - программный блок "Ожидание" сменил свое отображение! Рядом с песочными часами появилось изображение датчика касания (Рис. 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. 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. 7 поз.

3) при необходимости позволяет передать окончательное состояние датчика для обработки в другой программный блок.

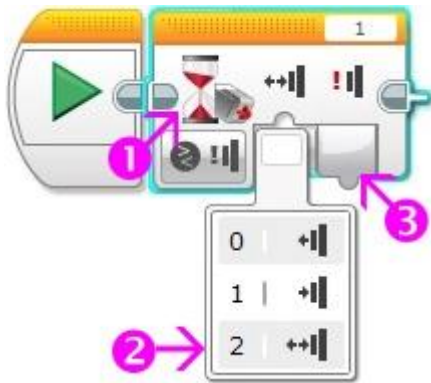


Рис. 7

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. 8)

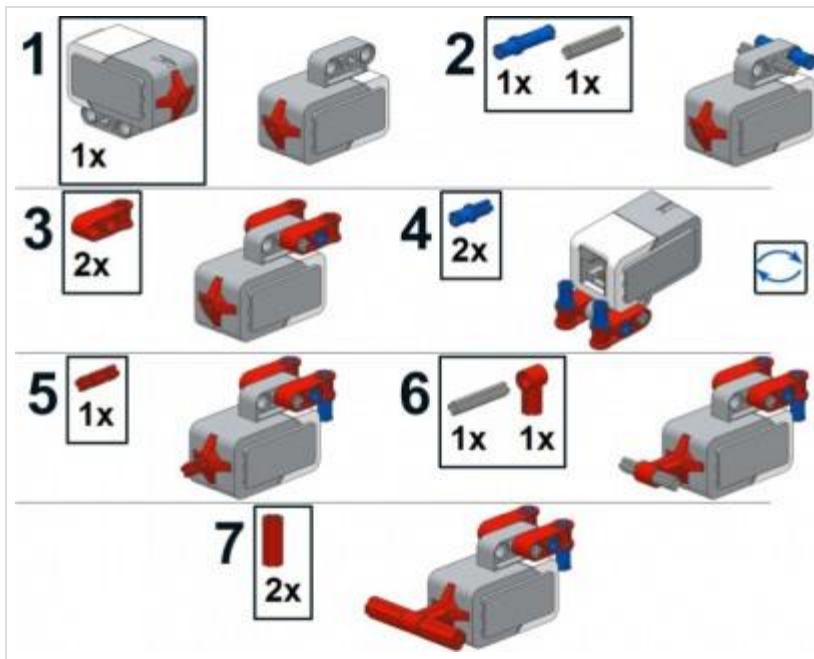


Рис. 8

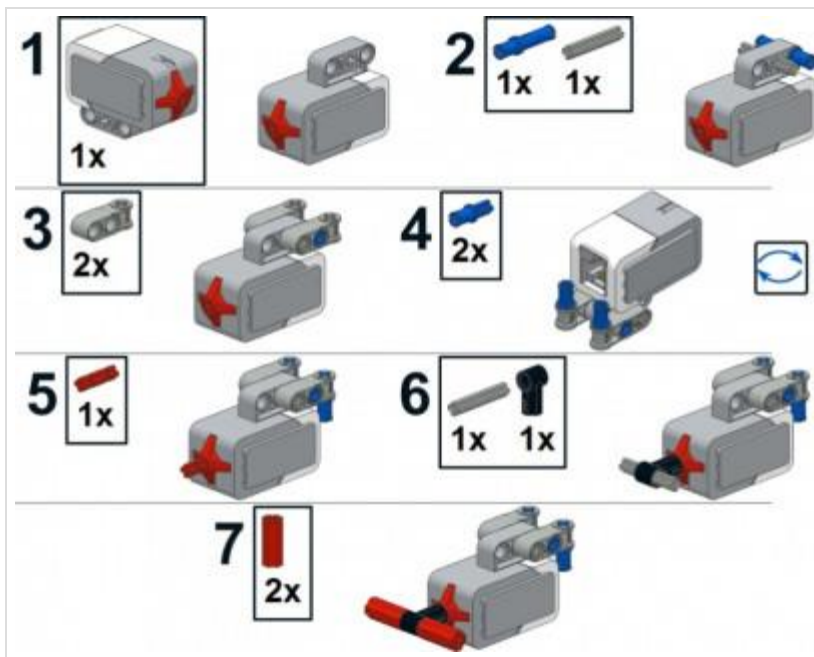
Задача №7: необходимо написать программу, останавливающую робота, столкнувшегося с препятствием.

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

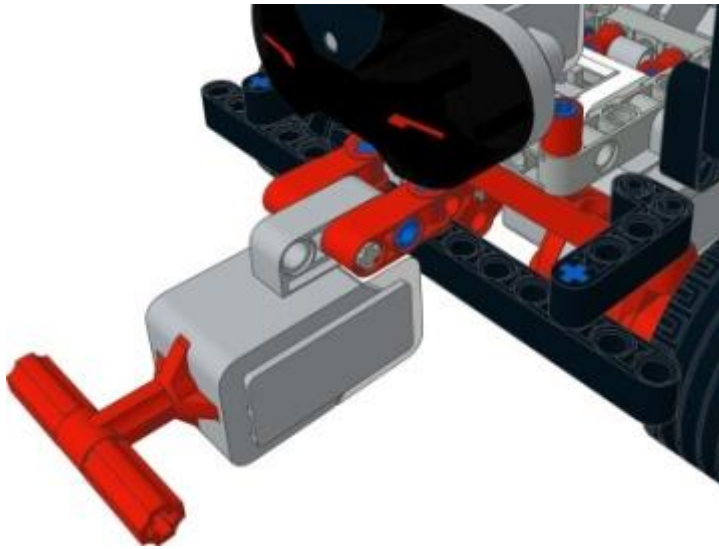


Lego mindstorms EV3 education

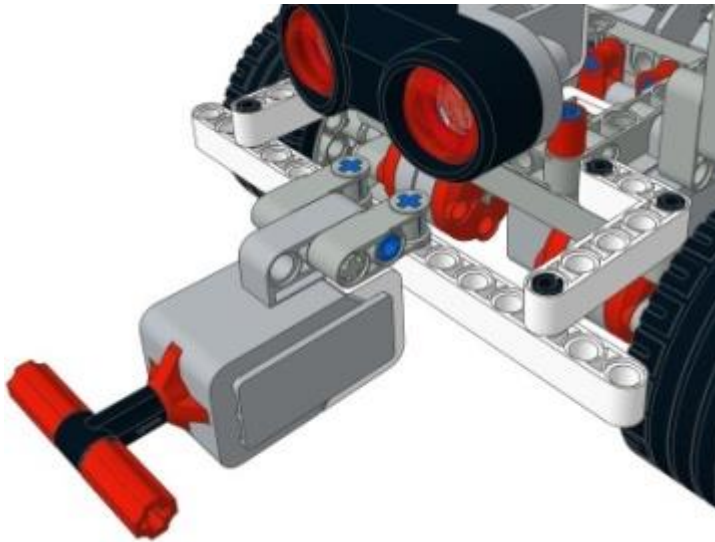


Получившийся элемент закрепим на передней балке нашего робота и соединим датчик касания с портом "1" модуля EV3.

Lego mindstorms EV3 Home



Lego mindstorms EV3 Education



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:

1. Начать прямолинейное движение вперед (Рис. 9 поз. 1).
2. Ждать, пока датчик касания не будет нажат (Рис. 9 поз. 2).
3. Прекратить движение вперед (Рис. 9 поз. 3).



Рис. 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".

Практическая работа № 3 **Структурный и кинематический анализ манипуляторов**

Цель работы — провести структурный анализ и ознакомиться с основными свойствами кинематических цепей копирующего манипулятора и промышленных роботов.

3.1 Промышленные роботы и манипуляторы.

Промышленный робот – программируемая автоматическая машина, состоящая из манипулятора и устройства программного управления его движением, предназначенная для замены человека при выполнении основных и вспомогательных операций в производственных процессах. *Манипулятор* – совокупность пространственного рычажного механизма и системы приводов, осуществляющая под управлением программируемого автоматического устройства или человека-оператора действия (манипуляции), аналогичные действиям руки человека.

3.2 Назначение и область применения.

Промышленные роботы предназначены для замены человека при выполнении основных и вспомогательных технологических операций в процессе промышленного производства. При этом решается важная социальная задача - освобождения человека от работ, связанных с опасностями для здоровья или с тяжелым физическим трудом, а также от простых монотонных операций, не требующих высокой квалификации. Гибкие автоматизированные производства, создаваемые на базе промышленных роботов, позволяют решать задачи автоматизации на предприятиях с широкой номенклатурой продукции при мелкосерийном и единичном производстве. Копирующие манипуляторы, управляемые человеком-оператором, необходимы при выполнении различных работ с радиоактивными материалами. Кроме того, эти устройства незаменимы при выполнении работ в космосе, под водой, в химически активных средах. Таким образом, промышленные роботы и копирующие манипуляторы являются важными составными частями современного промышленного производства.

3.3. Классификация промышленных роботов.

Промышленные роботы классифицируются по следующим признакам:

- 1) по характеру выполняемых технологических операций: основные, вспомогательные, универсальные;
- 2) по виду производства: литейные, сварочные, кузнечно-прессовые, для механической обработки, сборочные, окрасочные, транспортно-складские;

- 3) по системе координат руки манипулятора: прямоугольная, цилиндрическая, сферическая, сферическая угловая (ангулярная), другие;
- 4) по числу подвижностей манипулятора;
- 5) по грузоподъемности: сверхлегкие (до 10 Н), легкие (до 100 Н), средние (до 2000 Н), тяжелые (до 10000 Н), сверхтяжелые (свыше 10000 Н);
- 6) по типу силового привода: электромеханический, пневматический, гидравлический, комбинированный;
- 7) по подвижности основания: мобильные, стационарные;
- 8) по виду программы: с жесткой программой, перепрограммируемые, адаптивные, с элементами искусственного интеллекта;
- 9) по характеру программирования: позиционное, контурное, комбинированное.

3.4. Принципиальное устройство промышленного робота.

Манипулятор промышленного робота по своему функциональному назначению должен обеспечивать движение выходного звена и закрепленного в нем объекта манипулирования в пространстве по заданной траектории и с заданной ориентацией. Для полного выполнения этого требования основной рычажный механизм манипулятора должен иметь не менее шести подвижностей, причем движение по каждой из них должно быть управляемым. Промышленный робот с шестью подвижностями является сложной автоматической системой. Эта система сложна как в изготовлении, так и в эксплуатации. Поэтому в реальных конструкциях промышленных роботов часто используются механизмы с числом подвижностей менее шести. Наиболее простые манипуляторы имеют три, реже две, подвижности. Такие манипуляторы значительно дешевле в изготовлении и эксплуатации, но предъявляют специфические требования к организации рабочей среды. Эти требования связаны с заданной ориентацией объектов манипулирования относительно механизма робота. Поэтому оборудование должно располагаться относительно такого робота с требуемой ориентацией.

Рассмотрим для примера структурную и функциональную схемы промышленного робота с трехподвижным манипулятором. Основной механизм руки манипулятора состоит из неподвижного звена 0 и трех подвижных звеньев 1, 2 и 3 (рис.12.1). Структурная схема механизма этого манипулятора соответствует цилиндрической системе координат. В этой системе звено 1 может вращаться относительно звена 0 (относительное угловое перемещение φ_{10}), звено 2 перемещается по вертикали относительно звена 1 (относительное линейное перемещение S_{21}) и звено 3 перемещается в горизонтальной плоскости относительно звена 2 (относительное линейное перемещение S_{32}). На конце звена 3 укреплено захватное устройство или схват, предназначенный для захвата и удержания объекта манипулирования при работе манипулятора. Звенья основного рычажного механизма манипулятора образуют между собой три одноподвижные кинематические пары (одну вращательную А и две поступательные В и С) и могут обеспечить перемещение объекта в пространстве без управления его ориентацией. Для выполнения каждого из трех относительных движений манипулятор должен быть оснащен приводами, которые состоят из двигателей с редуктором и системы датчиков обратной связи. Так как движение объекта осуществляется по заданному закону движения, то в системе должны быть устройства сохраняющие и задающие программу движения, которые назовем носителями программ. При управлении от ЭВМ такими устройствами могут быть дискеты, диски CD, магнитные ленты и др. Преобразование заданной программы движения в сигналы управления двигателями осуществляется системой управления. Эта система включает ЭВМ, с соответствующим программным обеспечением, цифроаналоговые преобразователи и усилители. Система управления, в соответствии с заданной программой, формирует и выдает на исполнительные устройства приводов (двигатели) управляющие

воздействия u_i . При необходимости она корректирует эти воздействия по сигналам x_i , которые поступают в нее с датчиков обратной связи. Функциональная схема промышленного робота приведена на рис. 12.2.

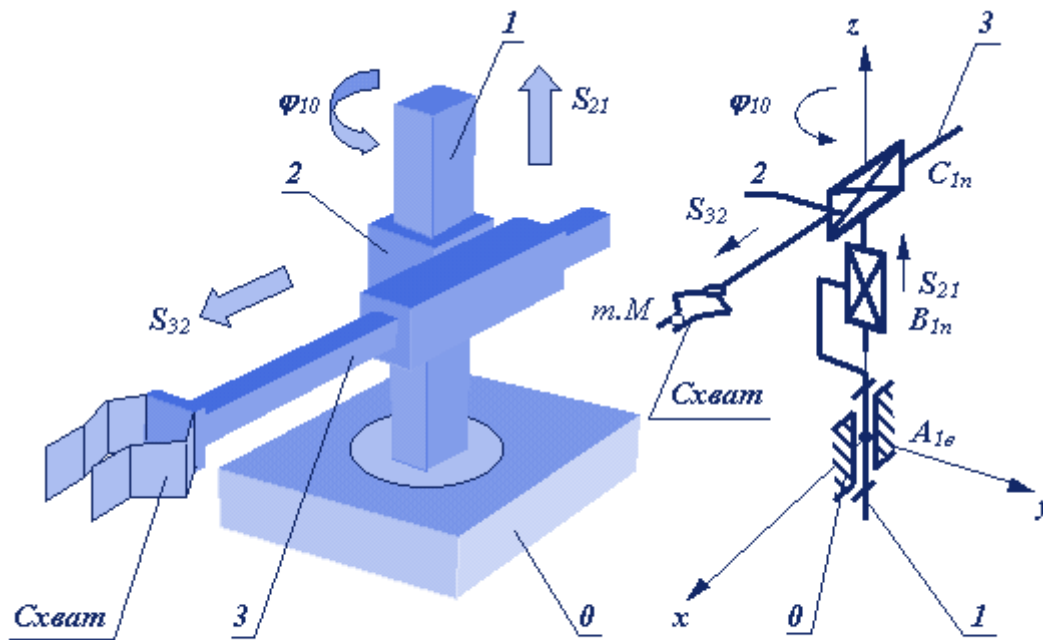


Рис. 12.1

3.5. Основные понятия и определения. Структура манипуляторов. Геометро-кинematicкие характеристики.

Формула строения - математическая запись структурной схемы манипулятора, содержащая информацию о числе его подвижностей, виде кинематических пар и их ориентации относительно осей базовой системы координат (системы, связанной с неподвижным звеном).

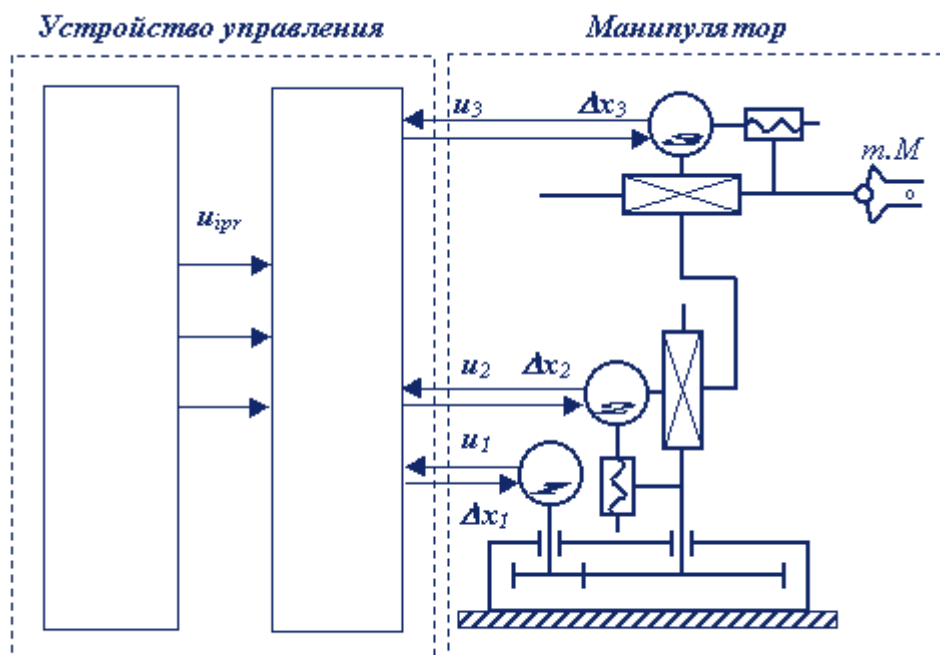


Рис. 12.2

Движения, которые обеспечиваются манипулятором делятся на:

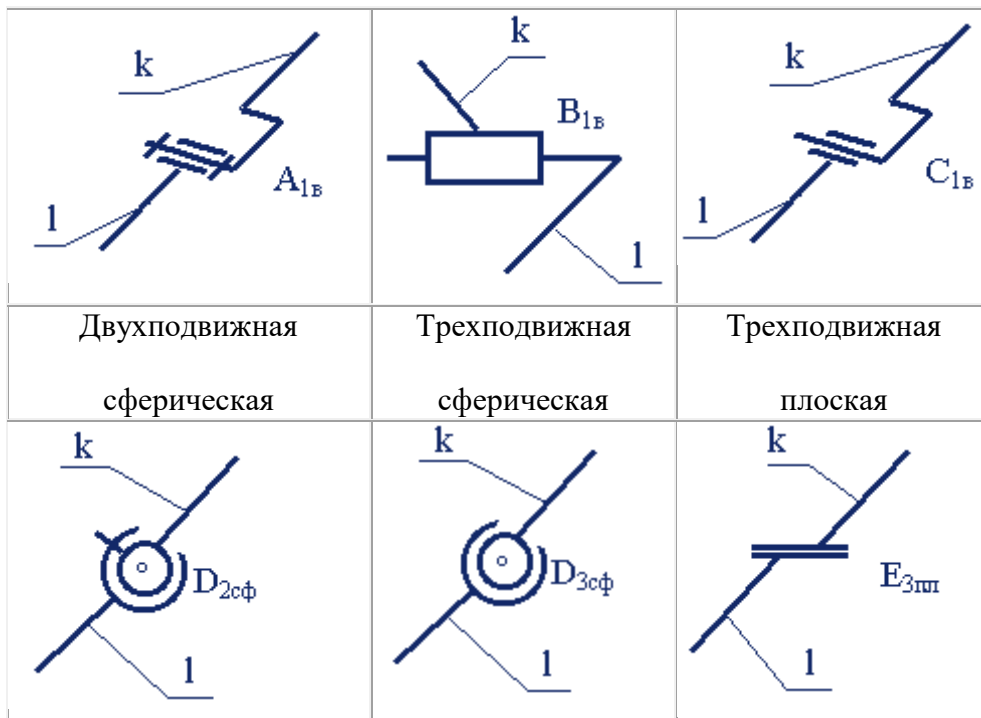
- *глобальные* (для роботов с подвижным основанием) - движения стойки манипулятора, которые существенно превышают размеры механизма;
- *региональные* (транспортные) - движения, обеспечиваемые первыми тремя звеньями манипулятора или его “рукой”, величина которых сопоставима с размерами механизма;
- *локальные* (ориентирующие) - движения, обеспечиваемые звеньями манипулятора, которые образуют его “кисть”, величина которых значительно меньше размеров механизма.

В соответствии с этой классификацией движений, в манипуляторе можно выделить два участка кинематической цепи с различными функциями: механизм руки и механизм кисти. Под “*рукой*” понимают ту часть манипулятора, которая обеспечивает перемещение центра схвата – точки M (региональные движения схвата); под “*кистью*” – те звенья и пары, которые обеспечивают ориентацию схвата (локальные движения схвата).

Структурная схема механизма – его графическое изображение на котором показаны стойка, подвижные звенья, кинематические пары и их взаимное расположение. Графическое изображение элементов схемы выполняется с учетом принятых условных обозначений. В таблице 12.1 приведены условные обозначения кинематических пар. Кинематической цепью называется система звеньев, образующих между собой кинематические пары. Цепь в которой каждое звено входит не более чем в две кинематические пары, называется простой. Незамкнутой называется такая кинематическая цепь, в которой есть звенья входящие только в одну кинематическую пару.

Таблица 12.1

Одноподвижная	Одноподвижная	Двухподвижная
вращательная	поступательная	цилиндрическая



Рассмотрим структурную схему антропоморфного манипулятора, то есть схему которая в первом приближении соответствует механизму руки человека (рис.12.3). Этот механизм состоит из трех подвижных звеньев и трех кинематических пар: двух трехподвижных сферических $A_{3сф}$ и $C_{3сф}$ и одной одноподвижной вращательной $B_{1в}$.

Кинематические пары манипулятора характеризуются: именем или обозначением КП - заглавная буква латинского алфавита (А,В,С и т.д.); звеньями, которые образуют пару (0/1,1/2 и т.п.); относительным движением звеньев в паре (для одноподвижных пар - вращательное, поступательное и винтовое); подвижностью КП (для низших пар от 1 до 3, для высших пар от 4 до 5); осью ориентации оси КП относительно осей базовой или локальной системы координат.

Рабочее пространство манипулятора - часть пространства, ограниченная поверхностями огибающими к множеству возможных положений его звеньев. *Зона обслуживания манипулятора* - часть пространства соответствующая множеству возможных положений центра схвата манипулятора. Зона обслуживания является важной характеристикой манипулятора. Она определяется структурой и системой координат руки манипулятора, а также конструктивными ограничениями наложенными относительно перемещения звеньев в КП.

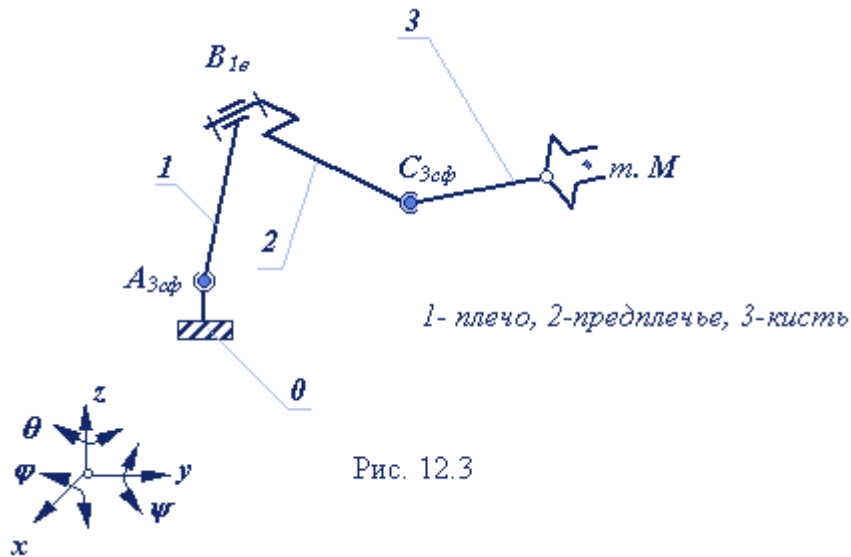


Рис. 12.3

Подвижность манипулятора W - число независимых обобщенных координат однозначно определяющее положение схвата в пространстве.

$$W = 6 \cdot n - \sum_{i=1}^5 (6 - i) \cdot p_i \quad (12.1)$$

или для незамкнутых кинематических цепей.

$$W = \sum_{i=1}^5 i \cdot p_i \quad (12.2)$$

Маневренность манипулятора M - подвижность манипулятора при зафиксированном (неподвижном) схвате

$$M = W - 6 \quad (12.3)$$

Возможность изменения ориентации схвата при размещении его центра в заданной точке зоны обслуживания характеризуется *углом сервиса* - телесным углом Ω , который может описать последнее звено манипулятора (звено на котором закреплен схват) при фиксации центра схвата в заданной точке зоны обслуживания

$$\Omega = f_C / l_{CM}^3 \quad (12.4)$$

где f_C - площадь сферической поверхности, описываемая точкой C звена 3,

l_{CM} - длина звена 3.

$$\text{Относительная величина } k_{\Omega} = \Omega / (4\pi \Omega_{\max}), \quad (12.5)$$

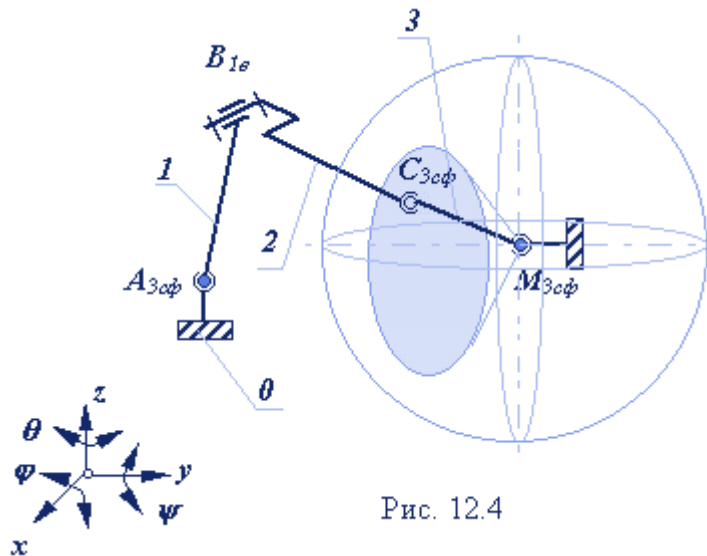
называется коэффициентом сервиса. Для манипулятора, изображенного на рис.12.3,

подвижность манипулятора $W = 6 \cdot 3 - (3 \cdot 2 - 5 \cdot 1) = 18 - 11 = 7$;

маневренность $M = 7 - 6 = 1$;

формула строения

$$W = [\square_{10} + \square_{10} + \square_{10}] + \square_{21} + [\square_{32} + \square_{32} + \square_{32}]. \quad (12.6)$$



Структура кинематической цепи манипулятора должна обеспечивать требуемое перемещение объекта в пространстве с заданной ориентацией. Для этого необходимо, чтобы хват манипулятора имел возможность выполнять движения минимум по шести координатам: трем линейным и трем угловым. Рассмотрим на объекте манипулирования точку M , которая совпадает с центром хвата. Положение объекта в неподвижной (базовой) системе координат $Ox_0y_0z_0$ определяется радиусом-вектором точки M и ориентацией единичного вектора с началом в этой точке. В математике положение точки в пространстве задается в одной из трех систем координат:

- прямоугольной декартовой с координатами x_M, y_M, z_M ;
- цилиндрической с координатами r_{sM}, φ_M, z_M ;
- сферической с координатами r_M, φ_M, θ_M .

Ориентация объекта в пространстве задается углами φ , θ и ψ , которые вектор ориентации образует с осями базовой системы координат. На рис. 12.5 дана схема шести подвижного манипулятора с вращательными кинематическими парами с координатами объекта манипулирования.

При структурном синтезе механизма манипулятора необходимо учитывать следующее:

- кинематические пары манипуляторов снабжаются приводами, включающими двигатели и тормозные устройства, поэтому в схемах манипуляторов обычно используются одноподвижные кинематические пары: вращательные или поступательные;

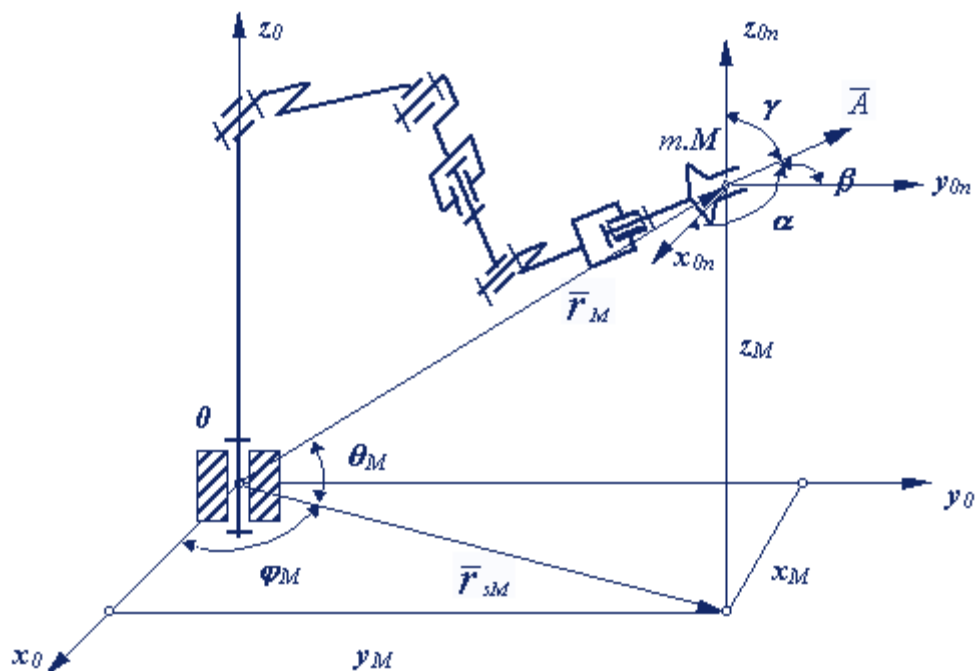


Рис 12.5

- необходимо обеспечить не только заданную подвижность свата манипулятора, но и такую ориентацию осей кинематических пар, которая обеспечивала необходимую форму зоны обслуживания, а также простоту и удобство программирования его движений;
- при выборе ориентации кинематических пар необходимо учитывать расположение приводов (на основании или на подвижных звеньях), а также способ уравнивания сил веса звеньев.

Перемещение схвата в пространстве можно обеспечить, если ориентировать оси первых трех кинематических пар по осям одной из осей координат. При этом выбор системы координат определяет тип руки манипулятора и вид его зоны обслуживания. По ГОСТ 25685-83 определены виды систем координат для руки манипулятора, которые приведены в таблице 12.2. Здесь даны примеры структурных схем механизмов соответствующие системам координат. Структурные схемы механизмов кисти, применяемые в манипуляторах, даны в таблице 12.3. Присоединяя к выходному звену руки тот или иной механизм кисти, можно получить большинство известных структурных схем манипуляторов, которые применяются в реальных промышленных роботах.

Структура манипулятора определяется и местом размещения приводов. Если приводы размещаются непосредственно в кинематических парах, то к массам подвижных звеньев манипулятора добавляются массы приводов. Суммарная нагрузка на приводы и их мощность увеличиваются, а отношение массы манипулятора к полезной нагрузке (максимальной массе объекта манипулирования) уменьшается. Поэтому при проектировании роботов приводы звеньев руки, как наиболее мощные и обладающие большей массой, стремятся разместить ближе к основанию робота. Для передачи движения от привода к звену используются дополнительные кинематические цепи. Рассмотрим схему руки манипулятора ПР фирмы ASEA (рис.12.6). К трехзвенному механизму с ангулярной системой координат добавлены:

- для привода звена 2 - простейший кулисный механизм, образованный звеньями 4,5 и 2;

- для привода звена 3 - цепь, состоящая из кулисного механизма (звенья 6,7 и 8) и шарнирного четырехзвенника (звенья 8,9,2 и 3).

Таким образом, в рычажном механизме можно выделить кинематическую цепь руки (звенья 1,2 и 3) и кинематические цепи приводов. Манипуляторы использующие принцип размещения приводов на основании имеют более сложные механизмы. Однако увеличение числа звеньев и кинематических пар компенсируется уменьшением масс и моментов инерции, подвижных звеньев манипулятора. Кроме того, замкнутые кинематические цепи повышают точность и жесткость механизма. В целом манипуляторы, использующие принципы комбинированного размещения приводов (часть приводов на основании, часть на подвижных звеньях), обладают лучшими энергетическими и динамическими характеристиками, а также более высокой точностью.

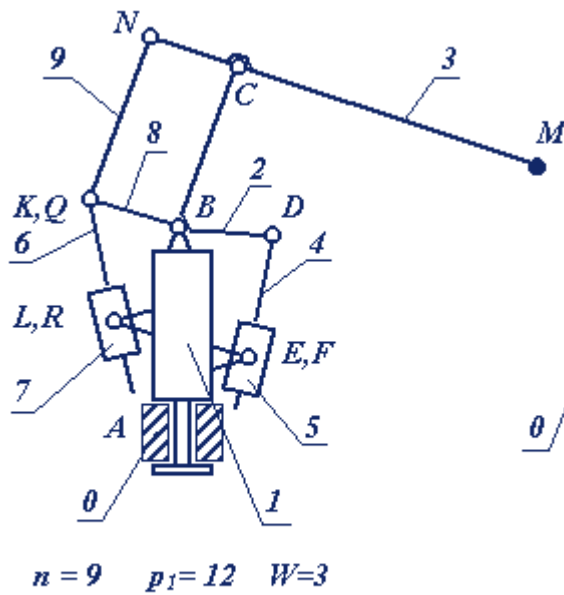


Рис.12.6

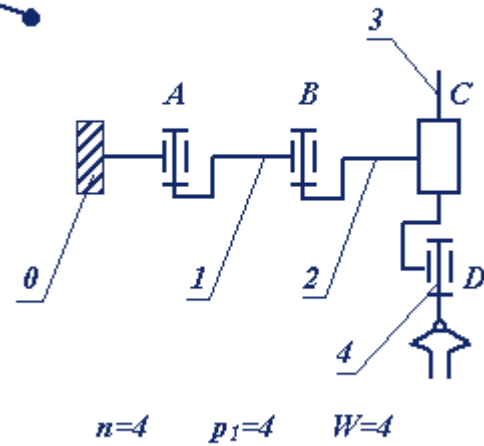
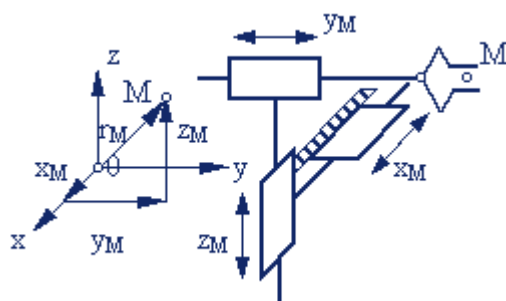


Рис.12.7

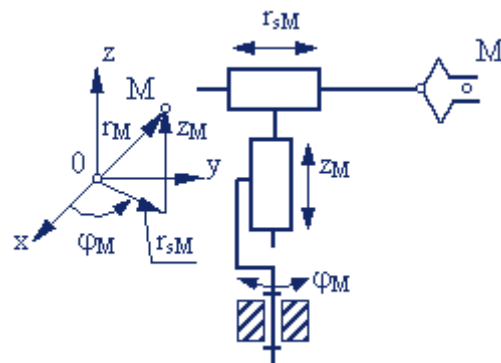
Системы координат “руки” манипулятора. Таблица 12.2

Прямоугольная (декартова)

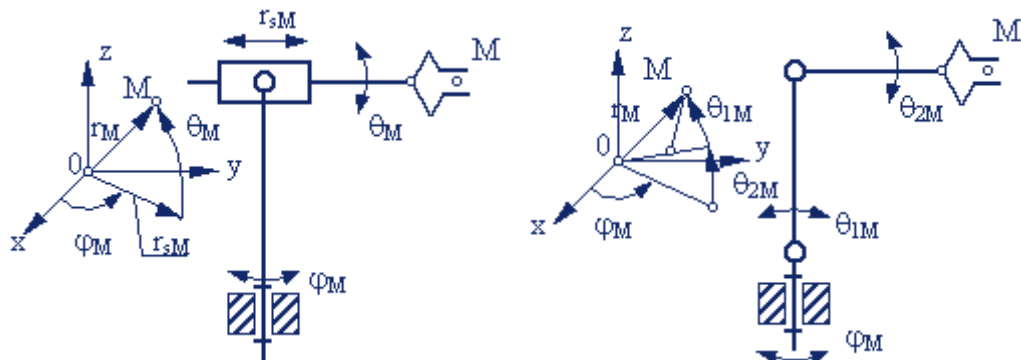


Сферическая

Цилиндрическая



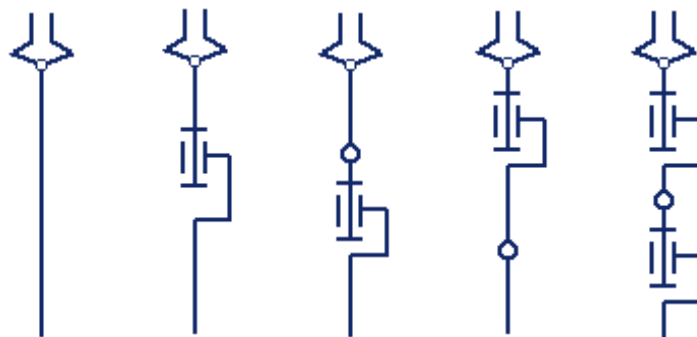
Угловая (ангулярная)



Другие

Таблица 12.3

Структурные схемы кисти манипулятора



В кинематических схемах рассмотренных манипуляторов веса звеньев вызывают дополнительную нагрузку на приводы. Фирма SKILAM разработала робот SANCIO (рис. 12.7) в котором веса приводов и звеньев воспринимаются кинематическими парами, а на момент двигателей влияют только через силы трения. Такая структурная схема механизма потребовала увеличения размеров кинематических пар, однако в целом был получен существенный выигрыш по энергетическим и динамическим показателям.

Важная особенность манипуляторов – изменение структуры механизма в процессе работы, о чем говорилось на лекции по структуре механизмов. В соответствии с циклограммой или программой работы робота, в некоторых кинематических парах включаются тормозные устройства. При этом два звена механизма жестко соединяются с друг другом, образуя одно звено. Из структурной схемы механизма исключается одна кинематическая пара и одно звено, число подвижностей схвата механизма уменьшается (обычно на единицу).

Изменяется структура механизма и в тех случаях, когда в процессе выполнения рабочих операций (на пример, при сборке или сварке) схват с объектом манипулирования соприкасается с окружающими предметами, образуя с ними кинематические пары. Кинематическая цепь механизма замыкается, а число подвижностей уменьшается. В этом случае в цепи могут возникать избыточные связи. Эти структурные особенности манипуляторов необходимо учитывать при программировании работы промышленного робота.

3.6. Пример структурного и кинематического анализа

В качестве примера рассмотрим структурный и кинематический анализ манипулятора робота *Unimate*. Структурная схема механизма изображена на рис.12.8.

Заполним таблицы лабораторного журнала. После заполнения таблицы сделаем вывод, который для данной схемы можно сформулировать так:

Манипулятор Unimate является шестиподвижным пространственным механизмом со сферической системой координат и разомкнутой кинематической цепью.

3.7 Порядок выполнения практической работы

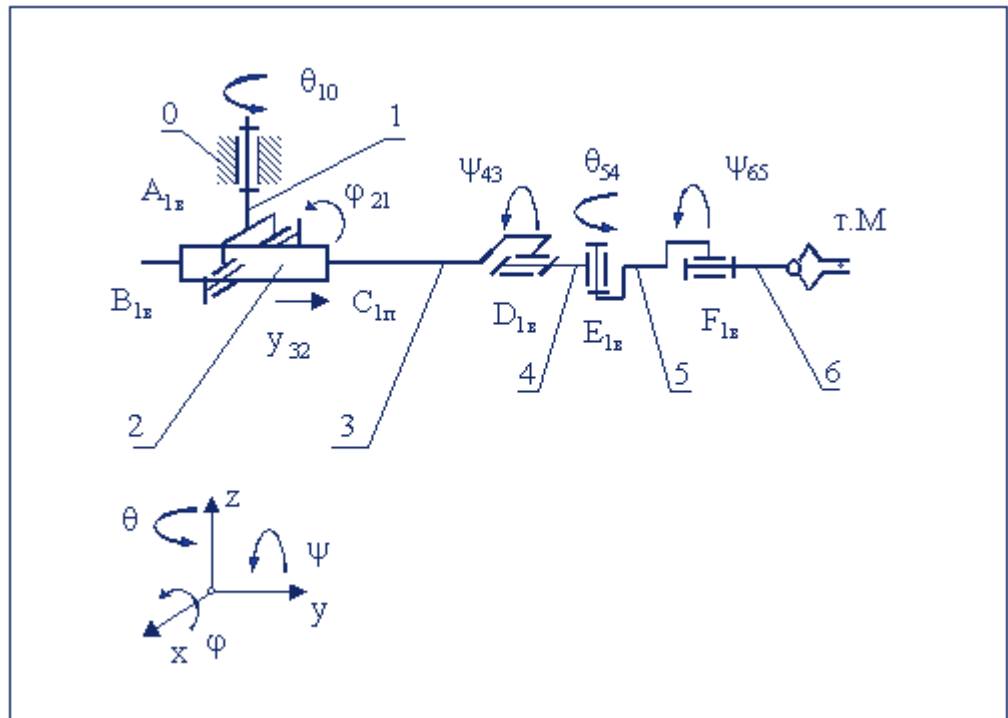
- 1) Включить промышленный робот или манипулятор и изучить относительное движение звеньев механизма. Начертить структурную схему манипулятора, нанести обозначения звеньев, кинематических пар, центра схвата манипулятора.
- 2) Определить основные параметры структурной схемы манипулятора и параметры его кинематических пар, заполнить соответствующие таблицы журнала.
- 3) Для выбранного промышленного робота или манипулятора по заданию преподавателя (задается формула строения манипулятора) выполнить структурный и кинематический анализ.

Сделать выводы по работе указав в них число подвижностей манипулятора, систему координат, вид кинематической цепи.

Формулы строения манипуляторов промышленных роботов

Таблица 12.4

№	Модель робота	Формула строения	№	Модель робота	Формула строения
1	Unimate	$W = 6 - 1 + 4 + 1 + 1 + 1 + 1$	9	Fanuc M-1	$W = 6 - 1 + z + y + 1 + 1 + 1$
2	Универсал	$W = 6 - 1 + x + 1 + 1 + 1 + 1$	10	Fanuc M-2	$W = z + 1 + x + 1 + 1 + 1$
3	Auto-Mate	$W = 6 - 1 + x + 1 + 1 + 1 + 1$	11	Toyota	$W = x + z + y + 1 + 1$
4	Prab	$W = 6 - 1 + x + 1 + 1 + 1 + 1$	12	Autohand	$W = z + y + x + 1 + 1$
5	PC-40	$W = x + 1 + 1 + 1 + z + y$	13	Kaufeldt	$W = z + 1 + x + 1 + 1$
6	PB- 231	$W = x + 1 + z + y + 1 + x$	14	IR-60	$W = 6 - 1 + 1 + 1 + 1 + 1 + 1$
7	Versatran	$W = y + 1 + z + x + 1 + 1$	15	Контур	$W = 6 - 1 + 1 + 1 + 1 + 1 + 1$
8	Matbac	$W = 6 - 1 + z + y + 1 + 1 + 1$	16	Asea	$W = 6 - 1 + 1 + 1 + 1 + 1$



манипулятора

Рис. 12.8

Основные параметры структурной схемы манипулятора

1	Формула строения	$W = \square_{10} + \square_{21} + y_{32} + \square_{43} + \square_{54} + \square_{65}$			
2	Подвижные звенья, образующие манипуляторе	“руку”	1	2	3
		В “кисть”	4	5	6
3	Система координат “руки” манипулятора	сферическая			
4	Число звеньев	общее	k	7	
		подвижных	n	6	
5	Число одноподвижных кинематических пар	вращательных	$p_{1в}$	5	
		поступательных	$p_{1п}$	1	
6	Число подвижностей в пространстве	$W_{пр}$	6		
7	Маневренность в пространстве	$M_{пр}$	0		
8	Число избыточных связей в механизме	q	0		

Кинематические пары манипулятора

Обозначение КП	A	B	C	D	E	F
Звенья, образующие КП	1/0	2/1	3/2	4/3	5/4	6/5
Относит. движение	вр.	вр.	пост.	вр.	вр.	вр.
Подвижность в КП	1	1	1	1	1	1
Ось ориентации КП	z	x	y	y	z	y

Практическое занятие №4

Знакомство с конструктором

Цель работы: получить практические навыки работы с конструктором *Lego mindstorms*

4.1. 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*, ты, 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.

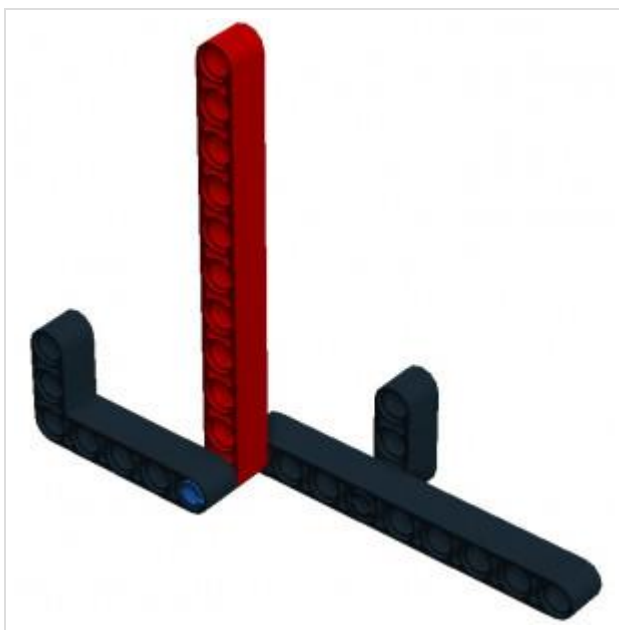


Рис. 3

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

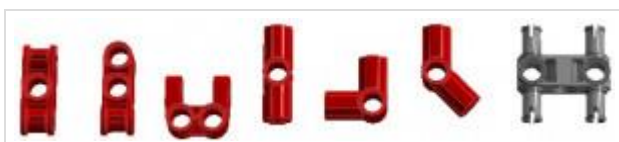


Рис. 4

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 обеспечивают 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.

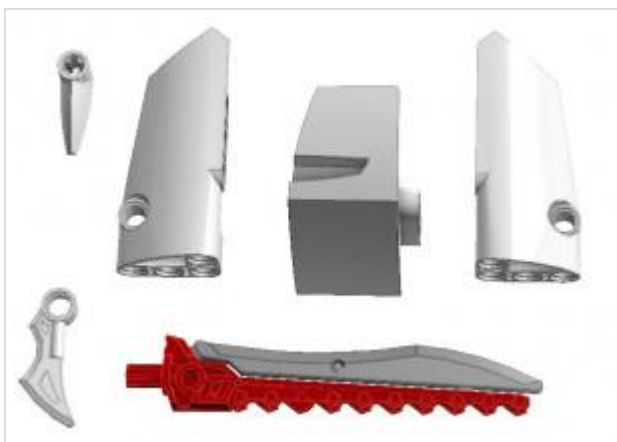


Рис. 7

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.

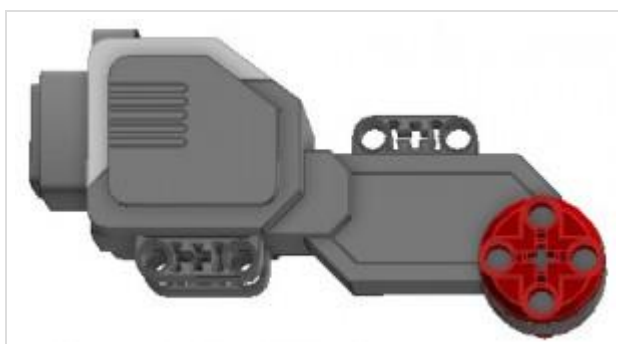


Рис. 8

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.

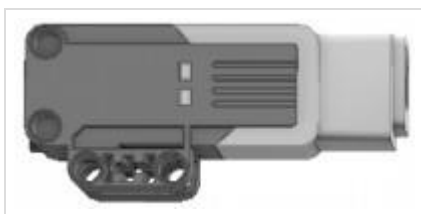


Рис. 9

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.

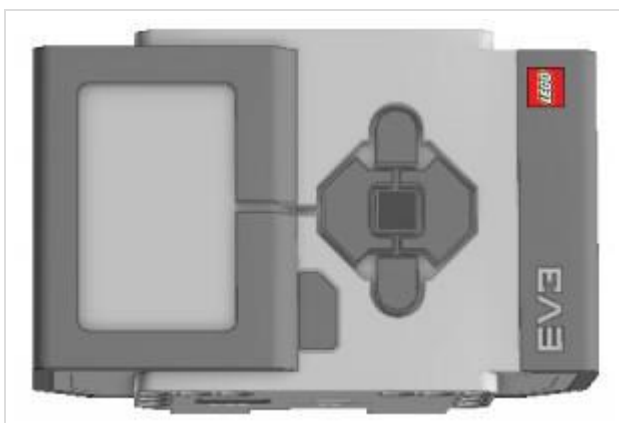


Рис. 11

4.2. 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.

4.3. 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.

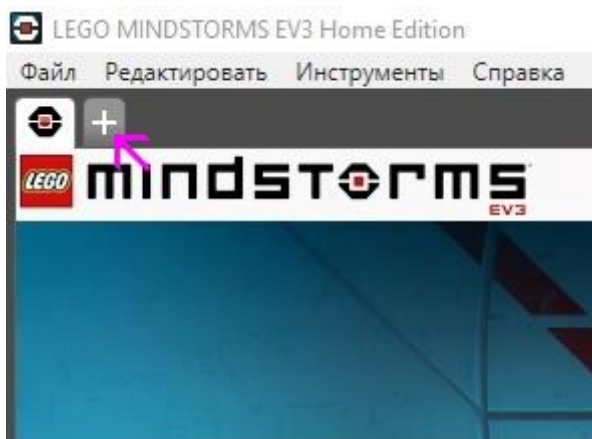


Рис. 12

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.

4.4. 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 кнопками контроллера EV3 (главного блока). Выберем блок **"Рулевое управление"** и перетащим его в область программирования (центральная область программы).

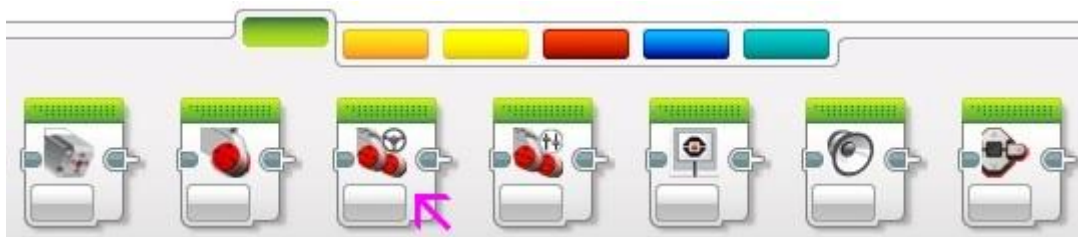


Рис. 14

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.

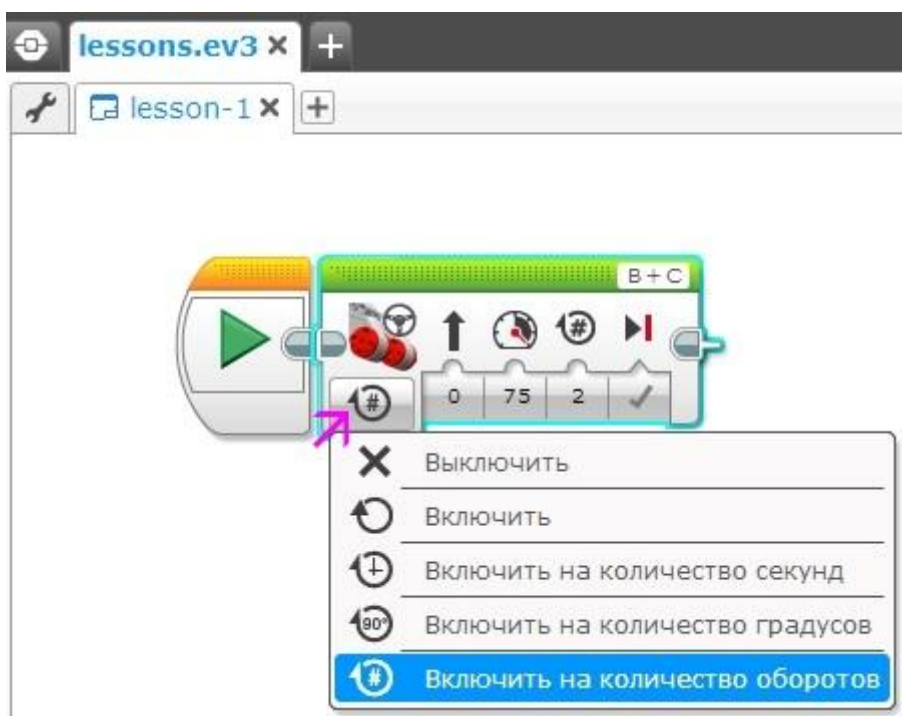


Рис. 15

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 провернуться.

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.

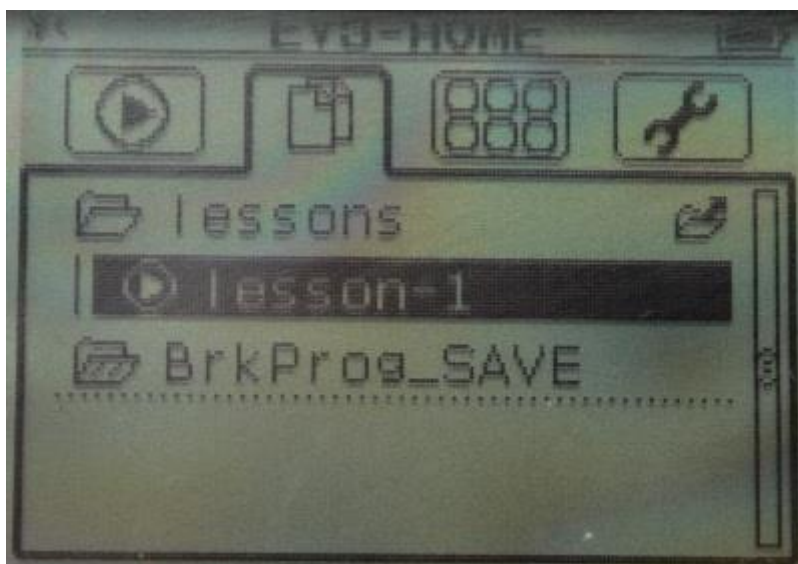


Рис. 17

Практическое занятие № 5.

Программирование движения робота

Цель работы: получить практические навыки программирования движения робота

Введение:

На втором занятии мы детальнее 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.

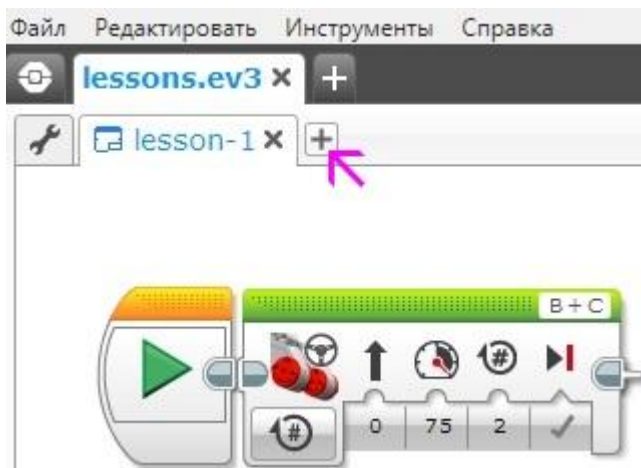


Рис. 1

5.1. 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

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

5.2. 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.

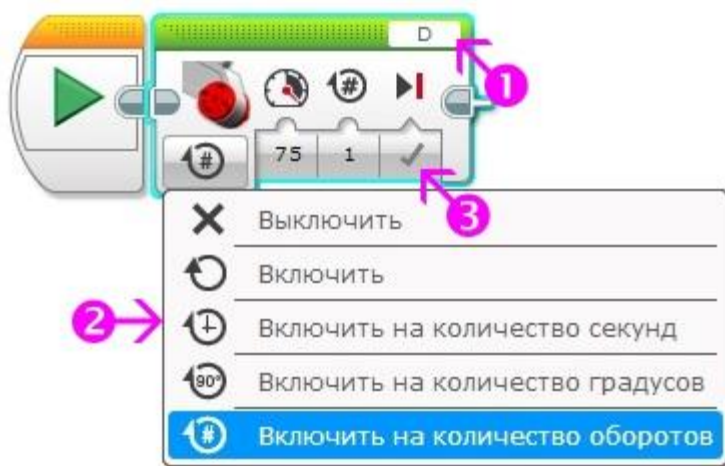


Рис. 3

Для правильной настройки блока управления большим мотором мы должны:

1. Выбрать порт, к которому подключен мотор (**А, В, С или D**) (Рис. 3 поз. 1)
2. Выбрать режим работы мотора (Рис. 3 поз. 2)
3. Настроить параметры выбранного режима (Рис. 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. 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 тормозить после выполнения команды, а если установлен в значение **"Двигаться накатом"**, 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. 4 поз. 1).

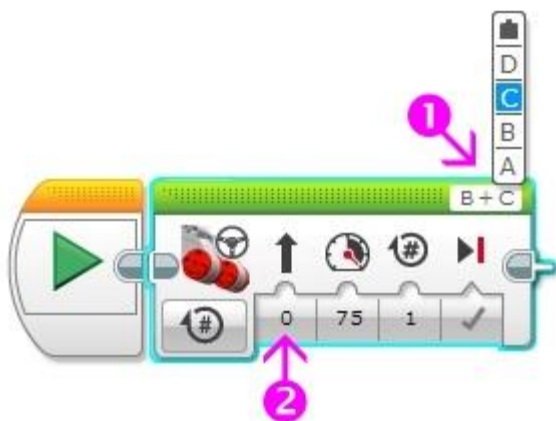


Рис. 4

Параметр "Рулевое управление" (Рис. 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. 5).



Рис. 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. 6).

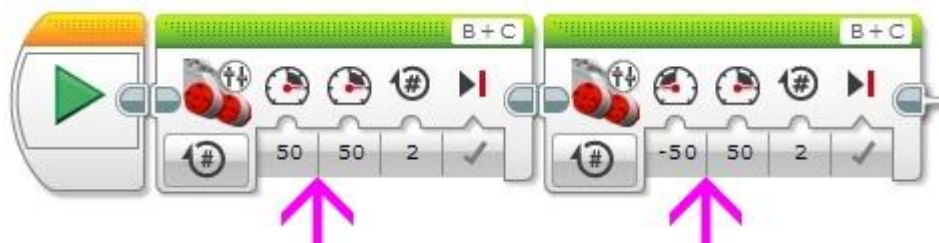


Рис. 6

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

5.3. 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. 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 провернуться валы левого и правого моторов нашего робота. На какой угол провернется робот вокруг своей оси - зависит от размера (диаметра) колес и расстояния между ними. На **Рис. 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.

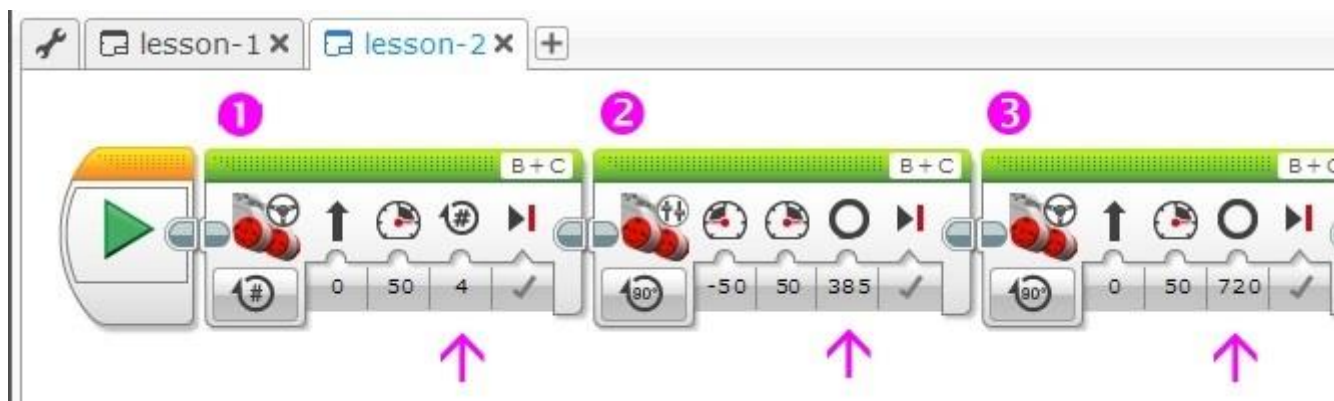


Рис. 7

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...

5.4. 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.

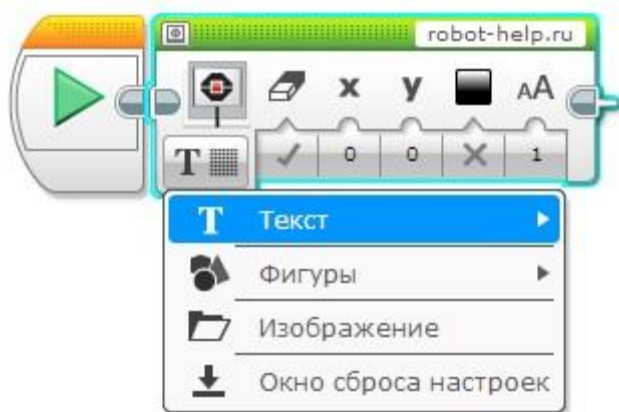


Рис. 8

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.

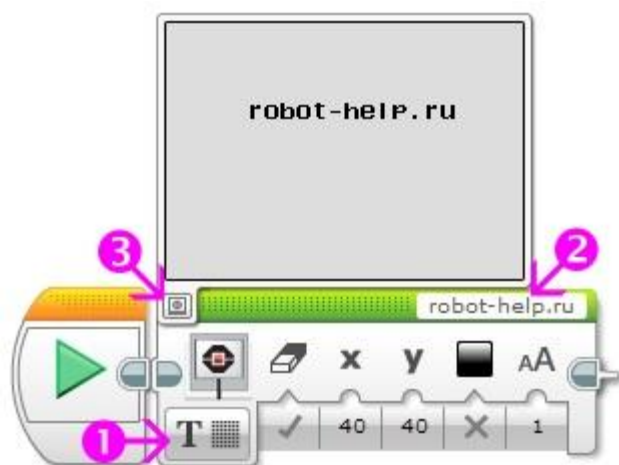


Рис. 9

Рассмотрим параметры программного блока "Экран" в режиме "Текст" (Рис. 9 поз.1). Строка, предназначенная для вывода на экран, вводится в специальное поле (Рис. 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 установите 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)

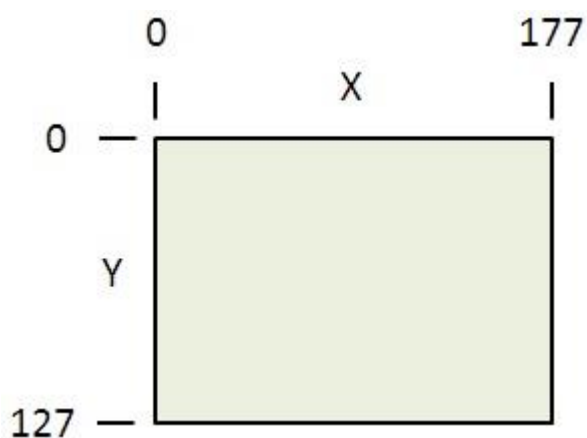


Рис. 10

Во время настройки программного блока "Экран" можно включить режим предварительного просмотра (Рис. 9 поз. 3) и визуально оценить результат настроек вывода информации.

В режиме "Фигуры" (Рис. 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. 11 поз. 2) отвечает за то, что будет отображен либо контур фигуры, либо внутренняя область фигуры будет заполнена цветом, заданным в параметре "Цвет" (Рис. 11 поз. 3).

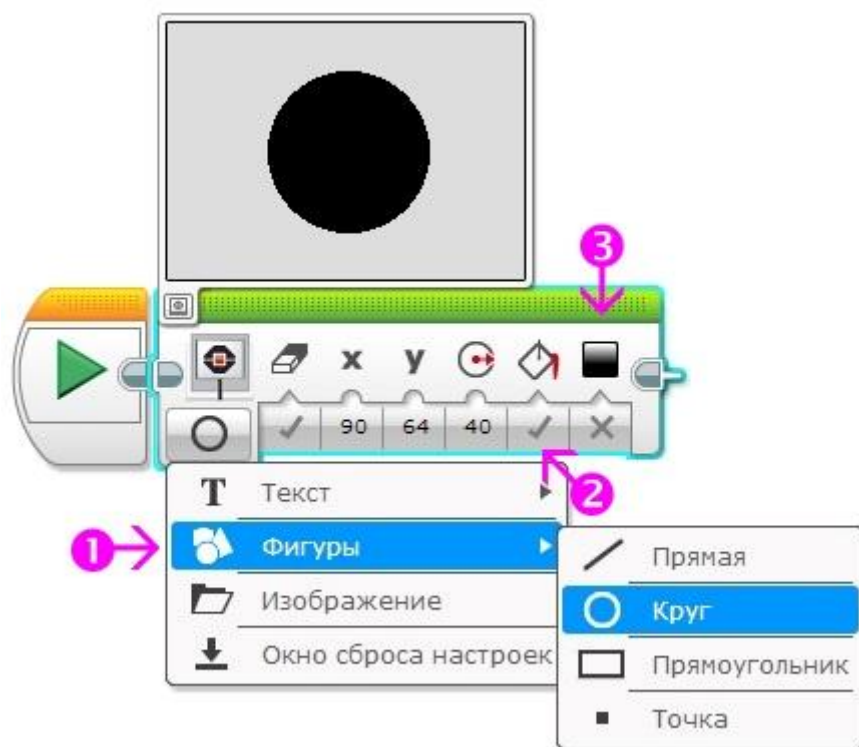


Рис. 11

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

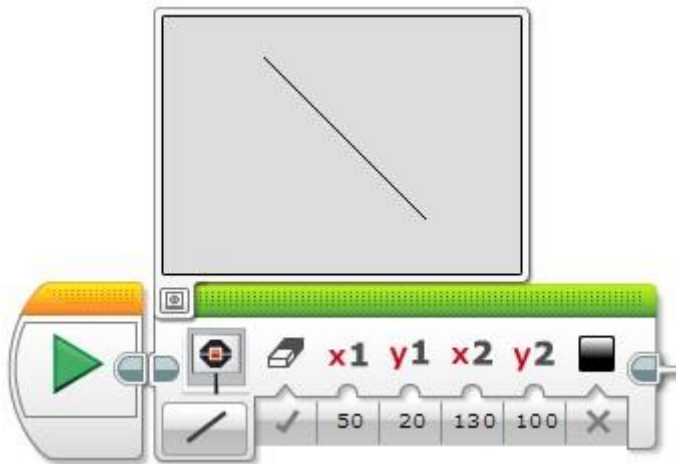


Рис. 12

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

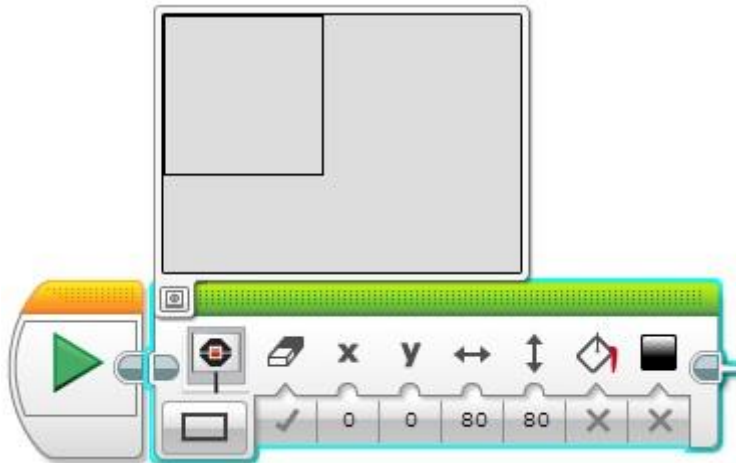


Рис. 13

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 изображения**"). Создавая своё изображение, вы можете также вывести на экран символы русского алфавита.

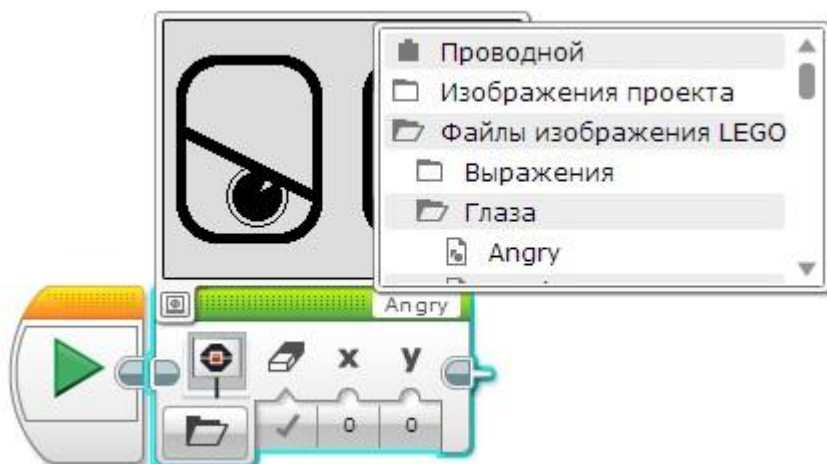


Рис. 14

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. 15). В этом режиме необходимо задать "Частоту" тона (Рис. 15 поз. 1), "Продолжительность" звучания в секундах (Рис. 15 поз. 2), а также громкость звучания (Рис. 15 поз. 3).

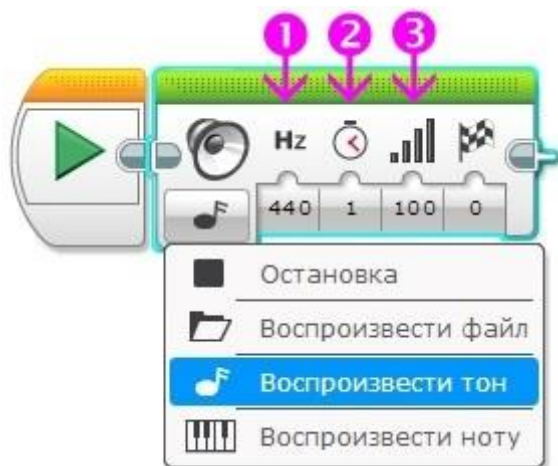


Рис. 15

В режиме "Воспроизвести ноту" вам вместо частоты тона необходимо выбрать ноту на виртуальной клавиатуре, а также установить длительность звучания и громкость (Рис. 16).



Рис. 16

В режиме **"Воспроизвести файл"** вы можете выбрать один из звуковых файлов из библиотеки (Рис. 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.

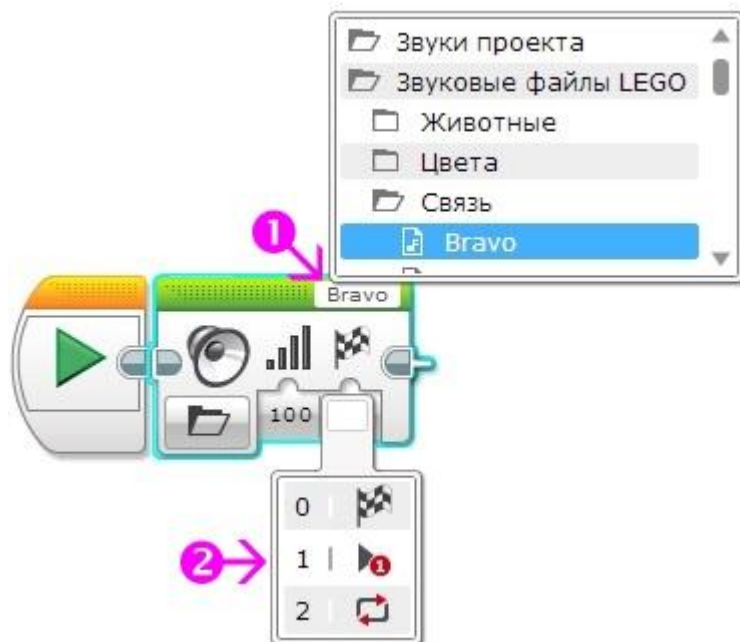


Рис. 17

Давайте отдельно рассмотрим параметр **"Тип воспроизведения"** (Рис. 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. 18 поз. 1). Параметр **"Цвет"** задает цветовое оформление индикации (Рис. 18 поз. 2). Параметр **"Импульсный"** отвечает за включение - отключение режима мерцания цветовой индикации (Рис. 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.

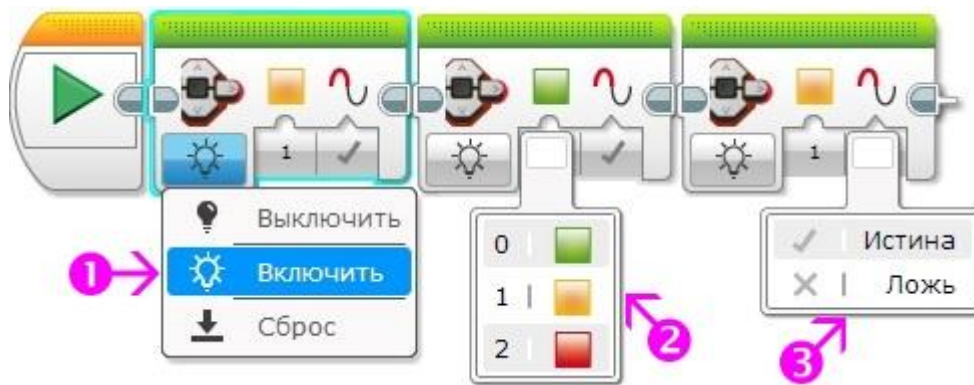


Рис. 18

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"**

3.Критерии оценивания практической работы

№ п/п	Критерии оценки	Макс.балл-5 баллов
1	Схема собрана верно и аккуратно. Схема легко читается.	3
2	При подключении провода проложены аккуратно.	1
3	Код программы не содержит ошибки и алгоритм работает в соответствии с условием задачи.	1
	ИТОГО	5

4.Список рекомендуемой литературы

Основная литература

1. Архипов, М. В. Промышленные роботы: управление манипуляционными роботами : учебное пособие для среднего профессионального образования / М. В. Архипов, М. В. Вартанов, Р. С. Мищенко. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2023. — 170 с. — (Профессиональное образование). — ISBN 978-5-534-13082-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru>
2. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для среднего профессионального образования / Л. А. Станкевич. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 495 с. — (Профессиональное образование). — ISBN 978-5-534-16241-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/>
3. Серебряков, А. С. Автоматика : учебник и практикум для среднего профессионального образования / А. С. Серебряков, Д. А. Семенов, Е. А. Чернов ; под общей редакцией А. С. Серебрякова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 476 с. — (Профессиональное образование). — ISBN 978-5-534-15853-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru>

Дополнительные источники

4. Сафиуллин, Р. К. Основы автоматки и автоматизация процессов : учебное пособие для среднего профессионального образования / Р. К. Сафиуллин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 146 с. — (Профессиональное образование). — ISBN 978-5-534-08256-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru>

Учебно-методические материалы:

- 1.Методические указания к практическим/лабораторным работам (Электронный ресурс)/ Коровин Ю.И., Горохов Д.В., – Москва: РГАУ-МСХА, 2021 – ЭБС –«РГАУ-МСХА»

Интернет – ресурсы

1. Электронно-библиотечная система РГАУ-МСХА им. К.А. Тимирязева (далее ЭБС) сайт www.library.timacad.ru
2. Научная электронная библиотека «КиберЛенинка» - <https://cyberleninka.ru/>
3. Сетевая электронная библиотека аграрных вузов - <https://e.lanbook.com/books>