**Изучение датчиков и моторов**

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.

**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**

**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.

**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 значение: **"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 внутрь программного блока **"Цикл"**.