Как было отмечено ранее, в основе самой платформы программного обеспечения WeDo 2.0 лежит графический язык программирования «G», заимствованный из среды программирования LabView (см. [«WeDo 2.0 Первые шаги»](http://xn--d1acamalered3bf4b5g.xn--p1ai/%d1%81%d1%80%d0%b5%d0%b4%d0%b0-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%bc%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f-lego-wedo-2-0-%d0%bf%d0%b5%d1%80%d0%b2%d1%8b%d0%b5-%d1%88%d0%b0%d0%b3/)). Но если в оригинале блоки соединяются между собой проводниками (на манер проводов в физическом мире), то для данной среды был выбран более простой подход. Блоки соединяются между собой по принципу «вагончиков» в составе поезда – друг за другом, а расширители блоков имеют пазлообразный вид и даже ребенку интуитивно понятно, что и куда нужно подсоединить.

Следующая идея разработчика тоже помогает «новичку» освоиться за самый короткий период времени. Это разделение программных блоков по цветовой палитре:

* Блоки управления мотором и индикатором смартхаба– зеленая палитра.
* Блоки работы с экраном, звуками и математикой – красная палитра.
* Блоки управления программой (запуск, ожидание, цикл) – желтая палитра.
* Блоки работы с датчиками – оранжевая палитра.
* Блоки расширения – синяя палитра.

Давайте разберём подробно, какие программные блоки можно встретить в каждой палитре.

**Блоки управления мотором и индикатором смартхаба**

Все блоки палитры имеют визуальную подсказку – на них нарисован мотор или смартхаб – поэтому сразу понятно каким элементом мы будем управлять, добавив этот блок в программу.



Первый блок с символом, похожим на спидометр, задает мощность (скорость вращения) двигателя. На практике чаще всего используют оба понятия, как взаимозаменяемые. Дети любят мыслить большими категориями и задают значения мощности в десятки тысяч, но это не имеет смысла, поскольку программное ограничение установлено на отметке «10», и все значения превышающие эту отметку воспринимаются именно как «10».

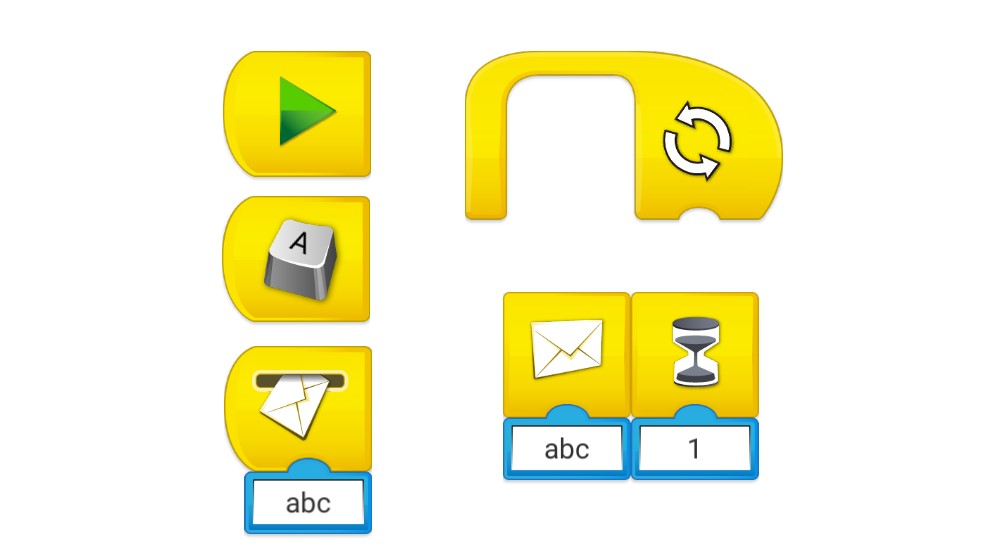
Блок с песочными часами задает время работы мотора. Единица измерения времени – секунда.

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

Блок с крестиком отвечает за остановку мотора. К слову, остановить мотором можно еще несколькими способами: установив в нужном месте алгоритма блок мощности с уставкой «0» или остановить программу целиком.

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

**Блоки управления программой (запуск, ожидание, цикл)**



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

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

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

Ещё один блок, который может быть стартовым – это блок «Получение сообщения» (работает в связке с блоком «Отправка сообщения»). Данный блок используется для перехода из одной ветки алгоритма в другую при достижении заданных параметров. Например, в основном алгоритме у вас выполняется программа, в которой содержится блок отправки сообщения «Stop». Вы устанавливается в рамках подпрограммы блок получения сообщения с аргументом «Stop» – и выполняете требуемую последовательность действий параллельно с выполнением основного кода. Этот блок достаточно часто применяется при опросе датчиков в режиме реального времени – под конкретное значение датчика пишется своя подпрограмма с аргументом, соответствующим этим числовым значениям.

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

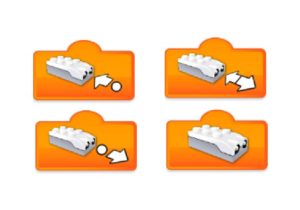
Последний в нашем обзоре блоков управления, но в то же время один их самых важных при написании программ – блок «Ожидание». По умолчанию это таймер, который останавливает выполнение программы на время, заданное в блоке расширения (отчёт ведётся в секундах). Расширить функционал блока можно, подключив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

**Блоки работы с датчиками**

Одно из основных отличий образовательного конструктора Lego WeDo 2.0 от обычного конструктора – это наличие датчиков, позволяющих роботам взаимодействовать с окружающим миром. Появление препятствий, удаление объектов, изменение наклона плоскости или управление джойстиком – все эти события внешнего мира нужно уметь понимать на программном уровне. Для этого в среде программирования WeDo 2.0 предусмотрены блоки расширения, которые считывают информацию с датчиков.

Датчик расстояния может работать в трех режимах:

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



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

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

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

* наклон носом вверх (к себе)
* наклон носом вниз (от себя)
* наклон влево
* наклон вправо
* отсутствие наклона (датчик расположен горизонтально) «0»
* наклон в любую сторону (режим «тряска»)

