

Р • О • Б • О • Ф • И • Ш • К • И



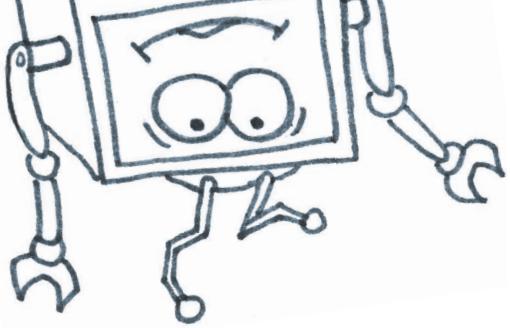
КОНСТРУИРУЕМ РОБОТОВ

на

LEGO[®] MINDSTORMS[®]
Education EV3

Сборник
проектов №2



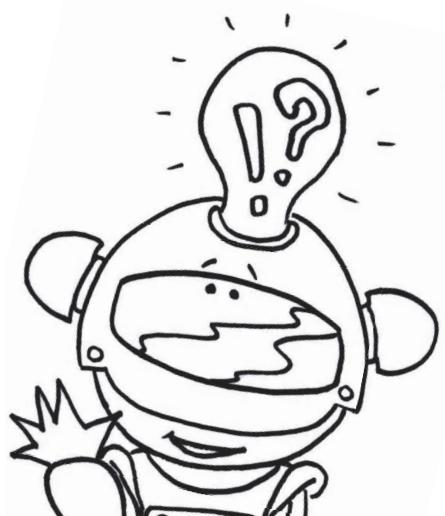


КОНСТРУИРУЕМ РОБОТОВ

на **LEGO[®]** MINDSTORMS[®]
Education EV3

Сборник
проектов № 2

Электронное
издание



Лаборатория знаний
Москва
2020

УДК 373.167
ББК 32.97
К65

Серия основана в 2016 г.

Ведущие редакторы серии
Т. Г. Хохлова, Ю. А. Серова

А в т о р ы:

Салахова А. А. (проект «Кормушка для питомца» в соавторстве);
Красных А. В. (проект «Кормушка для питомца» в соавторстве);
Тарапата В. В. (проект «Крутое пике» в соавторстве);
Рыжая Е. И. (проект «Крутое пике» в соавторстве);
Удалов В. В. (проект «Крутое пике» в соавторстве);
Валуев А. А. (проект «Который час?»);
Стерхова М. А. (проект «Секрет ткацкого станка»)

Конструируем роботов на LEGO® MINDSTORMS®
K65 Education EV3. Сборник проектов № 2 / сост. Ю. А. Серова.— Электрон. изд.— М. : Лаборатория знаний, 2020.— 285 с.— (РОБОФИШКИ).— Систем. требования: Adobe Reader XI ; экран 10".— Загл. с титул. экрана.— Текст : электронный.

ISBN 978-5-00101-880-3

Стать гениальным изобретателем легко! Серия книг «РОБОФИШКИ» поможет вам создавать роботов, учиться и играть вместе с ними.

В сборник № 2 вошли четыре проекта, с помощью которых вы сможете собрать роботов, выполняющих различные задачи: автоматическую кормушку для питомца, робота-авиасимулятора, роботизированные часы с кукушкой, а также ткацкий станок.

Для технического творчества в школе и дома, а также для занятий в робототехнических кружках.

УДК 373.167
ББК 32.97

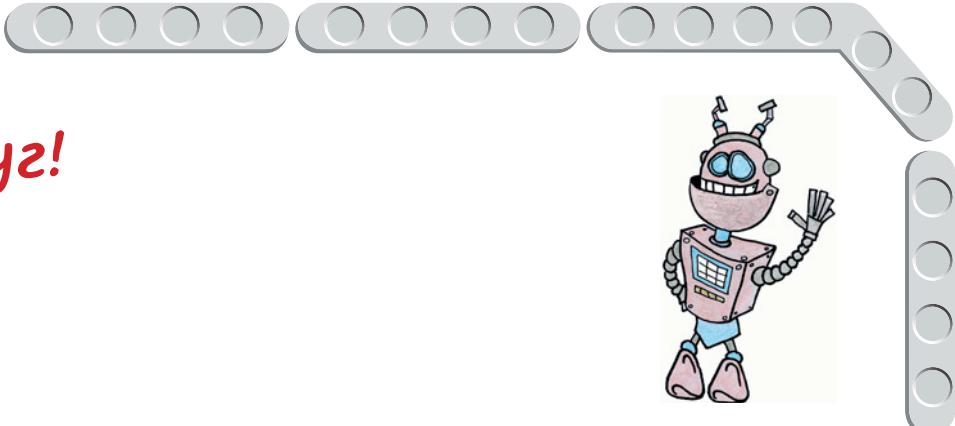
Деривативное издание на основе печатного аналога: Конструируем роботов на LEGO® MINDSTORMS® Education EV3. Сборник проектов № 2 / сост. Ю. А. Серова.— М. : Лаборатория знаний, 2020.— 282 с. : ил.— (РОБОФИШКИ).— ISBN 978-5-00101-243-6.

(6+)

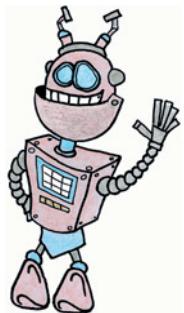
В соответствии со ст. 1299 и 1301 ГК РФ при устраниении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации

ISBN 978-5-00101-880-3

© Лаборатория знаний, 2020



Дорогой друг!



Как видно, ты уже совсем не новичок в LEGO, раз добрался до набора LEGO® MINDSTORMS® Education EV3 и, конечно, быстро собрал всё, что там предлагалось!

Что же делать теперь? Набор дорогой, выбрасывать жалко, а у младшего братика (если он есть) пока другие игрушки. Не расстраивайся! Мы тебе поможем.

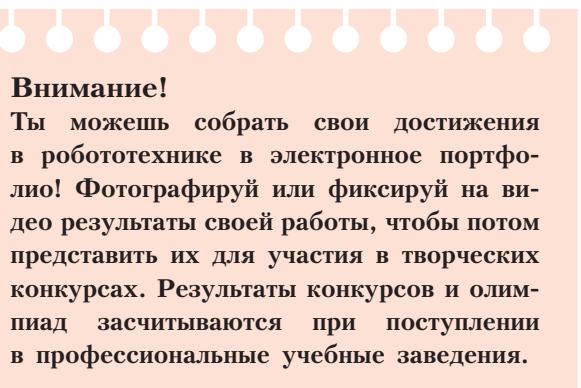
Из этого набора можно собрать ещё много интересных и полезных вещей. Например, ты можешь собрать:

- автоматическую кормушку для питомца;
- робота-авиасимулятора;
- робота, способного измерять время и имитировать работу аналоговых (стрелочных) часов, да ещё и добавишь к нему кукушку, оповещающую о каждом новом часе;
- ткацкий станок, на котором можно соткать фенечки, ленточки, одежду для кукол, эмблемы любых футбольных клубов.

Задумайся над этим!

Фактически за какой-то час работы ты сумеешь пройти многовековой путь изобретателей прошлого!

Почему в настоящее время такое стало возможно? Можно ли изобрести что-нибудь новое, не зная, какие машины и механизмы существовали в прошлом? Как интересней работать — одному или вместе с другом?



Внимание!

Ты можешь собрать свои достижения в робототехнике в электронное портфолио! Фотографируй или фиксируй на видео результаты своей работы, чтобы потом представить их для участия в творческих конкурсах. Результаты конкурсов и олимпиад засчитываются при поступлении в профессиональные учебные заведения.

Оборудование:

- Базовый набор LEGO® MINDSTORMS® Education EV3.
- Компьютер (минимальные системные требования): Windows XP, Vista, Windows 7, Windows 8 (за исключением METRO), Windows 10 (32/64 бит), оперативная память не менее 1 Гб, процессор – 1,6 ГГц (или быстрее), разрешение экрана – 1024 × 600, свободное место на диске – 5 Гб.
- Программное обеспечение LEGO® MINDSTORMS® Education EV3 (LME-EV3).

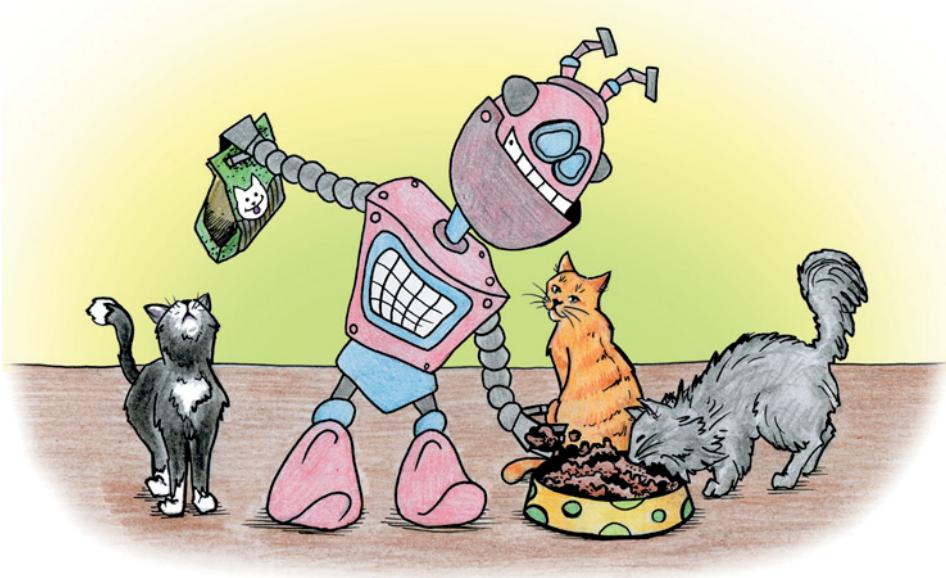
Обозначения

В тексте тебе встретятся обозначения, которые мы сейчас поясним на примерах.

1. Балка № 7 – это балка с семью отверстиями.
2. 3-модульный штифт – штифт, длина которого равна длине балки № 3.
3. Ось № 5 – ось, длина которой равна длине балки № 5.



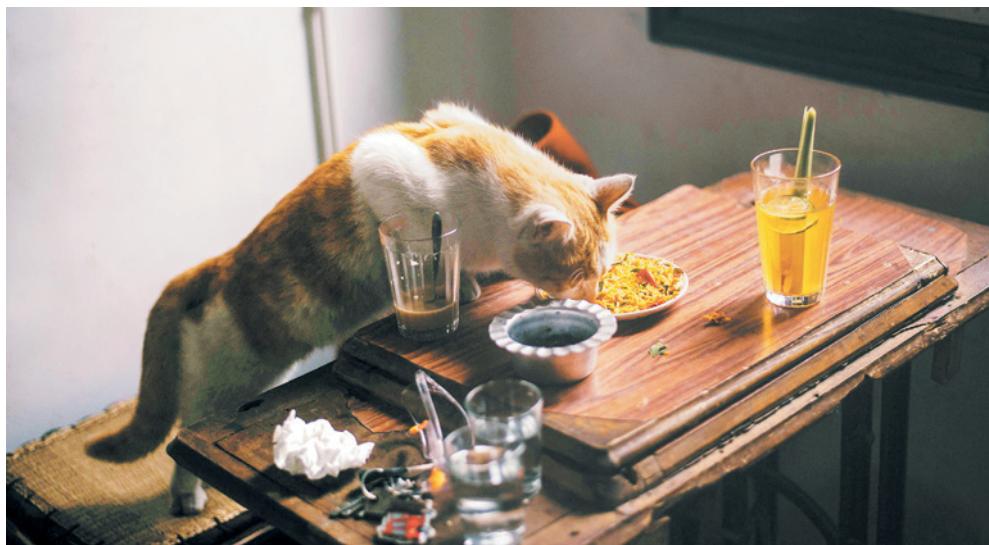
Кормушка для питомца



Сытый питомец — довольный питомец



Давай немного пофантазируем. Представь, что после школы ты пошёл в секции, а потом — как же иначе — пошёл погулять с друзьями. Дворовый футбол! Или хоккей? Нет разницы, но домой ты пришёл поздно. В прихожей — стащил с ног ботинки, добрался поскорее до комнаты и упал лицом в подушку.



Миау! Ми-и-и-а-а-у!

Знакомы эти звуки? Ты ничего не забыл? Как говорил известный французский писатель Антуан де Сент-Экзюпери, мы в ответе за тех, кого приручили. И твой любимый котик ничуть не виноват, что ты или твоя младшая сестрёнка забыли его покормить! Он же живой. Ему хочется есть. К сожалению, не все хозяева успевают вовремя следить за рационом своих любимых питомцев. Это обусловлено высокой занятостью в течение дня как взрослых, так и детей. Чем меньше животное, тем обычно быстрее протекает его обмен веществ. И если взрослый человек может потерпеть без обеда, то даже тебе уже трудно! Что говорить о бедном коте, который не может самостоятельно открыть холодильник и налить себе тарелку вчерашнего супа. Кстати, если питомец сам решит найти еду и стащит что-то со стола, это



может крайне негативно сказаться на его здоровье. Аналогичная ситуация касается собак.

Котятам и старым кошкам часто требуется соблюдать особый рацион, предусматривающий кормление по расписанию. Им важно получать все элементы для крепкого здоровья, но малыши не всегда могут точно определить, сколько они способны съесть. Ещё хуже дело обстоит у рыбок и жителей террариумов и экзотариумов: им нужно строго дозировать рацион.

Чтобы решить эту проблему, были придуманы автоматические кормушки для домашних питомцев. Они бывают различных типов. Чаще всего в них предусматривается выдача еды по таймеру (через определённое время), в более продвинутых допускается установка расписания в зависимости от часов реального времени.

На самом деле необходимость в автоматических кормушках возникла в первую очередь не для домашних любимцев, а для хозяйственных животных, например коров или кур. Кроме ферм, автоматические кормушки и поилки устанавливаются в заказниках и заповедниках. На подворьях с большим количеством скота автоматические кормушки являются средством безопасности! Они позволяют избежать попадания воды или влажного корма в сухой, закрывают доступ бактерий и грязи к нетронутому корму и дают ровно столько еды, сколько съест животное, что обеспечивает для него чистоту и комфорт. Для человека это тоже важно! В про-

Кстати!

Часы реального времени (**ЧРВ**) — это специальная электронная схема, которая состоит из автономного источника питания и учитывающего устройства. В отличие от таймера, который хранит время от запуска таймера или самого устройства, ЧРВ сохраняет информацию о проходящем времени всегда (естественно, в пределах возможностей его источника питания).



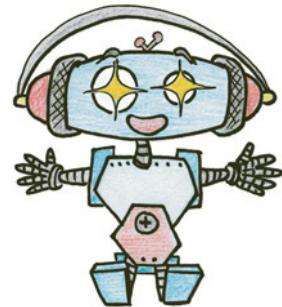
тивном случае он может тоже заразиться — не только от контакта с коровой, но и через молоко или мясо.

Животные быстро привыкают к использованию автоматических средств и приучаются нажимать на определённые кнопки или рычажки, зная, что за этим последует лакомство. Правда, некоторые начинают искать способы добраться и до других автоматов, предназначенных для людей, как кусуля на фотографии.

И всё же, преисполнившись важности момента, вспомни о своём любимом питомце! Мы предлагаем тебе самому собрать и запрограммировать автоматическую кормушку для питомца. Она будет выдавать корм через определённое время. Готов к великому делу на пользу кота?

Вперёд, зооинженер!

Этап 1. Устройство кормушки



Рассмотри автоматическую кормушку для питомца, собранную на основе набора LEGO® MINDSTORMS® Education EV3.

Попробуй выделить в ней рабочие детали — лоток для подачи корма, запирающий механизм, блок управления подачей. Куда кот должен нажать своей мягкой лапкой? Что тогда будет происходить? Почему конструкция не переворачивается даже с тяжёлым кормом и где находится её центр тяжести?

Обрати внимание, в конце проекта дан перечень всех деталей, которые потребуются тебе для сборки. Этот перечень поможет быстро найти то, что необходимо, и не ошибиться при конструировании.

Вот так выглядит автоматическая кормушка, которую ты соберёшь (рис. 1).

Кстати!

Несмотря на то что на платах компьютеров всегда присутствуют часы реального времени (именно благодаря им до входа в систему компьютер знает, какой сейчас год, месяц, день и даже время), ты будешь использовать обычный таймер.

Дело в том, что стандартная система, установленная на твой блок EV3, и сре-да программирования LME имеют доступ только к таймеру. На официальном сайте не указано, есть ли схема часов реаль-ного времени на основной плате блока, но опознать её без документации мы не смогли. Скорее всего, при установке аль-тернативных систем (например, *Debian*) на блок, EV3 использует время сети.

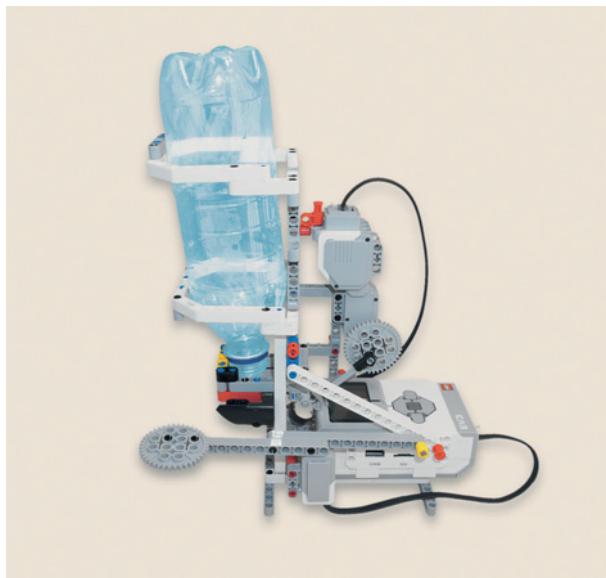
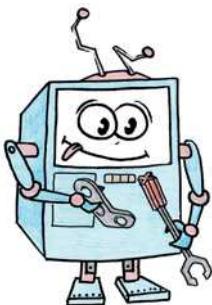
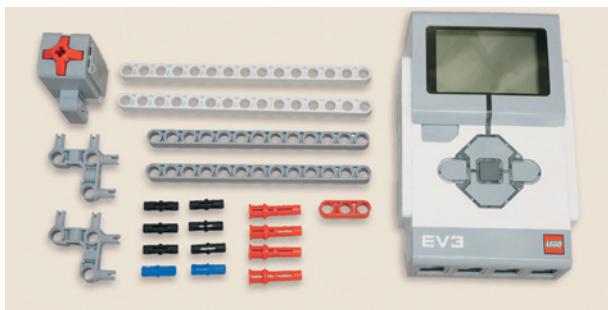


Рис. 1. Автоматическая кормушка

Этап 2. Сборка автоматической кормушки



ШАГ 1. СБОРКА ОСНОВАНИЯ КОРМУШКИ



Детали для сборки:

- балка № 15, белая, 2x;
- балка № 13, серая, 2x;



- балка № 3, красная, 1x;
- соединительный штифт, 2-модульный, чёрный, 6x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт со втулкой, 3-модульный, красный, 4x;
- угловой соединительный штифт, 3 × 3-модульный, серый, 2x;
- датчик касания, 1x;
- модуль EV3, 1x.



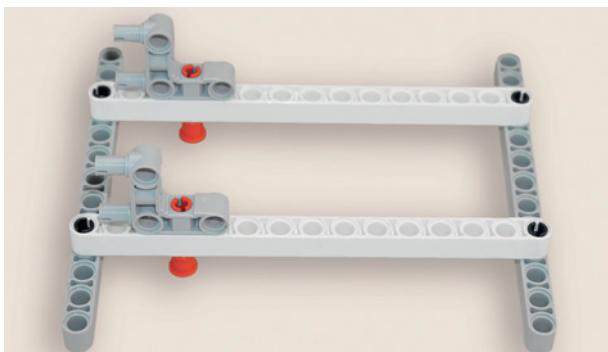
1. Возьми две белые балки № 15 и вставь снизу в первые и последние модули по одному чёрному штифту, а в третья модули вставь серые угловые соединительные штифты так, чтобы крепления были направлены влево.



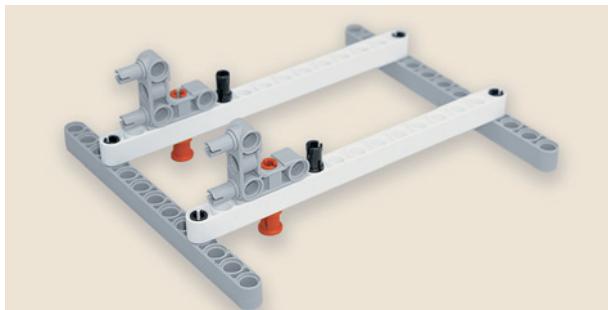
2. Вставь снизу по одному красному 2-модульному штифту с втулкой в четвёртые модули белых балок № 15.



3. Надень две серые балки № 13 на свободные концы чёрных штифтов четырьмями и десятыми модулями.



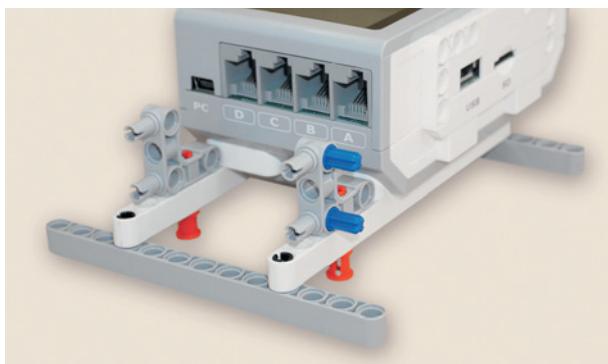
4. Возьми два чёрных штифта и вставь их в шестые модули белых балок № 15.

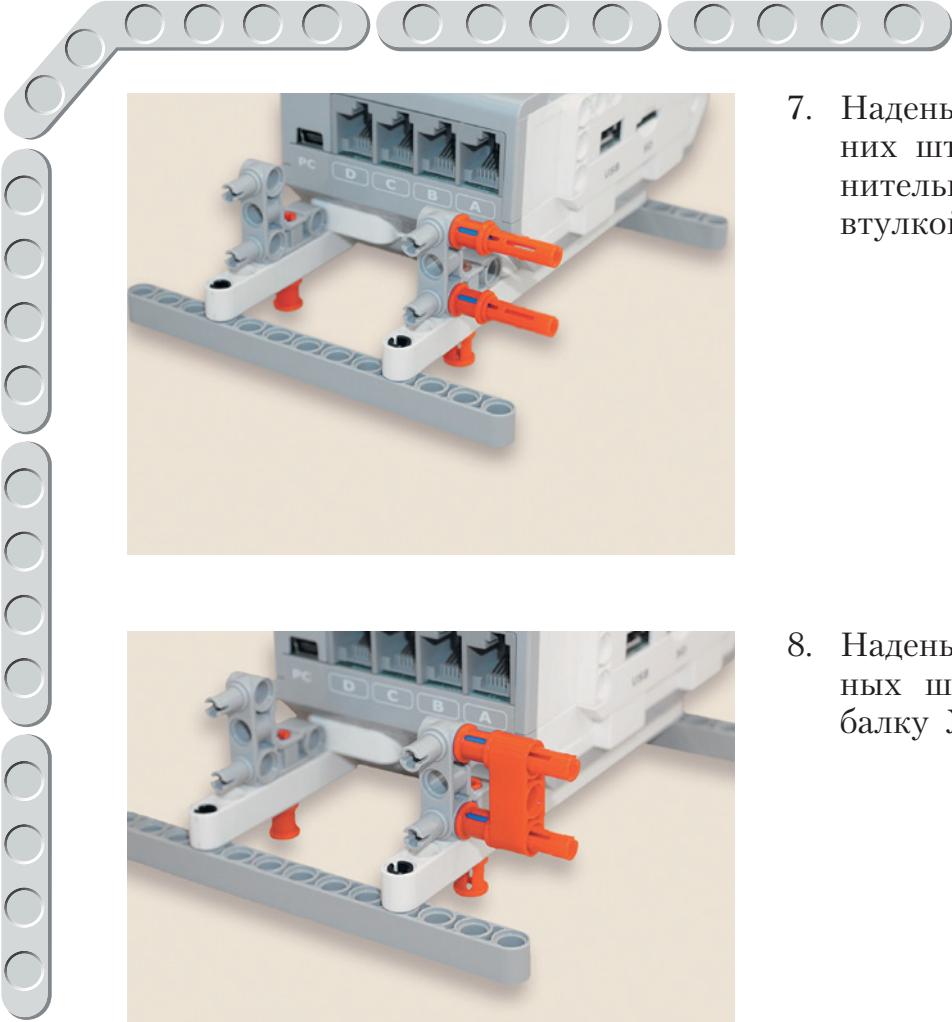


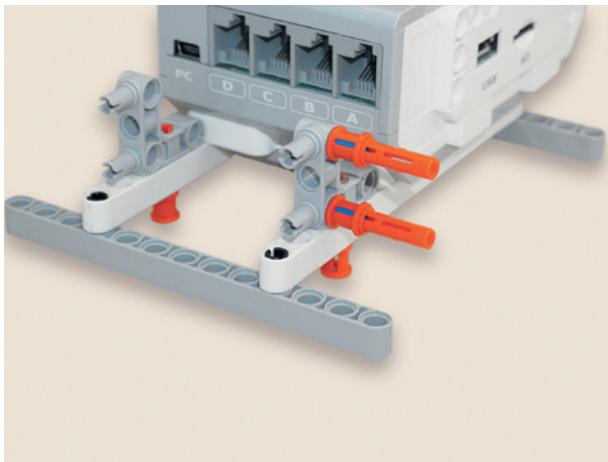
5. Надень на свободные концы чёрных штифтов нижними креплениями блок EV3 так, чтобы числовые разъёмы располагались справа.



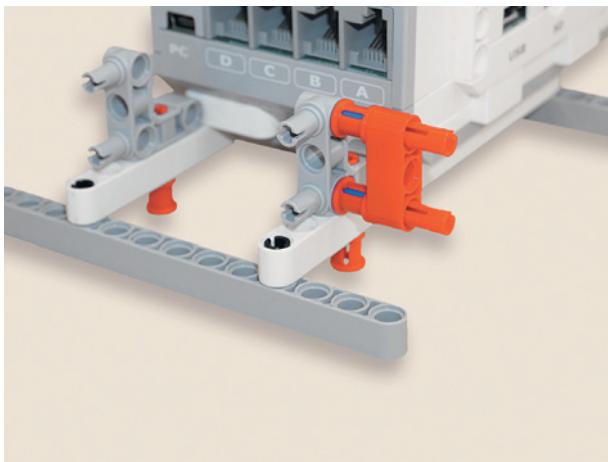
6. Поверни собранную конструкцию угловыми соединительными штифтами к себе. Вставь в вертикальные модули правого штифта два синих 2-модульных штифта с внешней стороны конструкции.



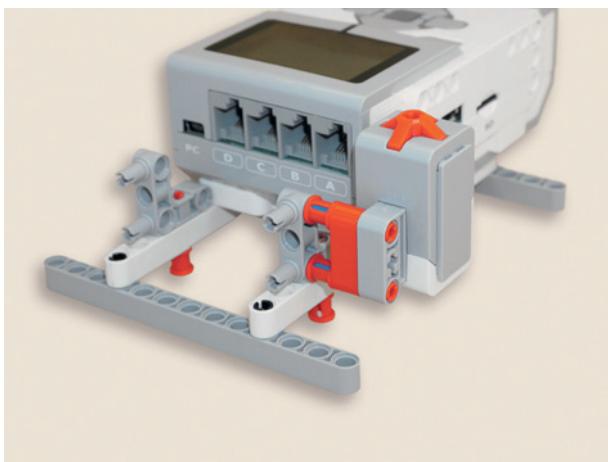
- 
7. Надень на свободные концы синих штифтов два красных соединительных 3-модульных штифта с втулкой.



8. Надень на свободные концы красных штифтов до упора красную балку № 3.



9. Затем надень на свободные концы красных штифтов датчик касания разъёмом вниз.



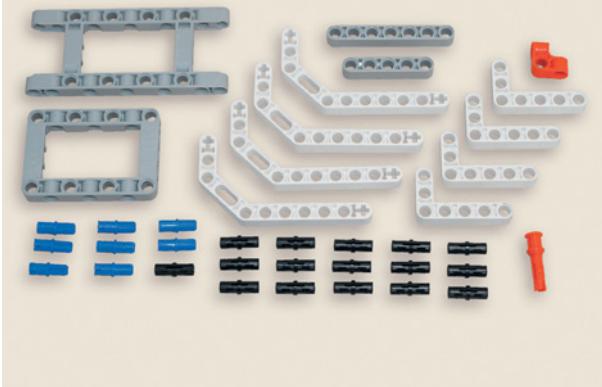
С основанием ты закончил. Переходи к креплению ёмкости для корма — бутылки.



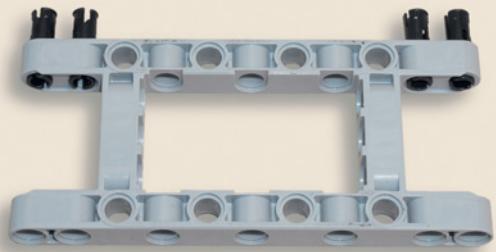
ШАГ 2. СБОРКА КРЕПЛЕНИЙ ЁМКОСТИ ДЛЯ КОРМА

Детали для сборки:

- рамка 5×11 , серая, 1x;
- рамка 5×7 , серая, 1x;
- балка № 7, серая, 1x;
- балка № 5, серая, 1x;
- двойная угловая балка, 3×7 -модульная, белая, 4x;
- прямоугольная балка 3×5 , белая, 4x;
- поперечная балка 2×1 , красная, 1x;
- соединительный штифт, 2-модульный, чёрный, 16x;
- соединительный штифт, 2-модульный, синий, 8x;
- соединительный штифт со втулкой, 3-модульный, красный, 1x.



1. Возьми серую рамку и вставь в крайние модули верхней длинной стороны по два чёрных 2-модульных штифта.



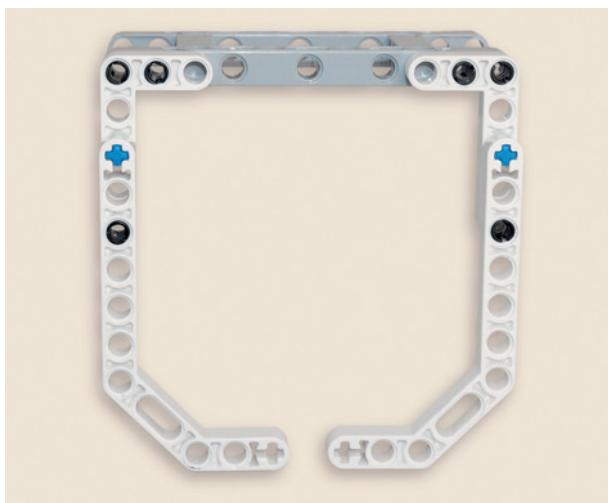
2. Поверни рамку чёрными штифтами к себе и надень на их свободные концы по одной белой угловой балке короткими сторонами, а длинными сторонами — вниз, как показано на рисунке.



3. Возьми два синих 2-модульных штифта и вставь их в длинные стороны белых угловых балок в третья модули сверху, а два чёрных штифта вставь в пятые модули.



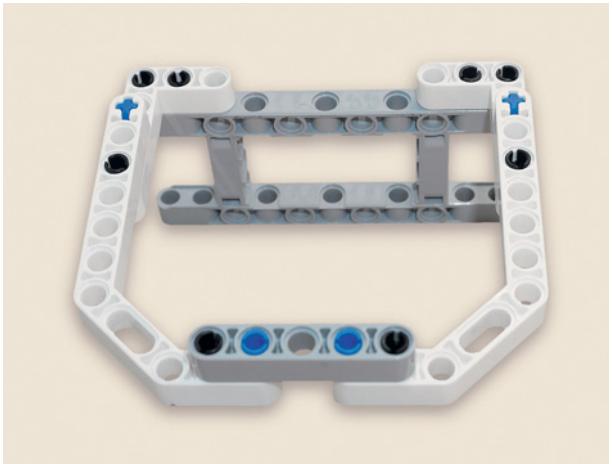
4. Надень на свободные концы синих и чёрных штифтов по одной белой двойной угловой балке первыми и третьими модулями длинных сторон сверху так, чтобы короткие стороны балок были направлены внутрь конструкции.



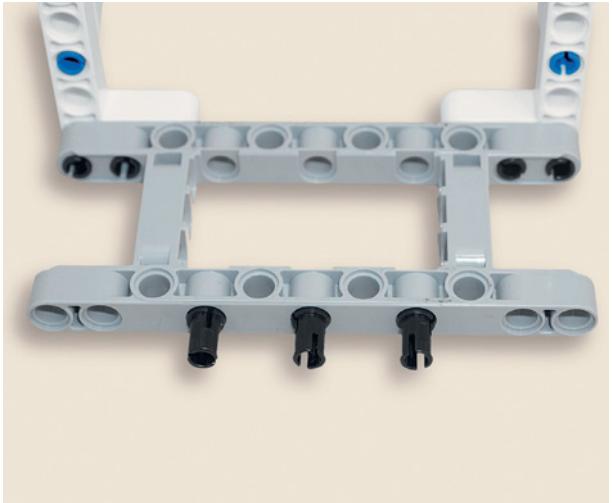
5. Теперь возьми два синих 2-модульных штифта и вставь их в крайние модули коротких сторон двойных угловых балок. Затем вставь два чёрных штифта во вторые модули.



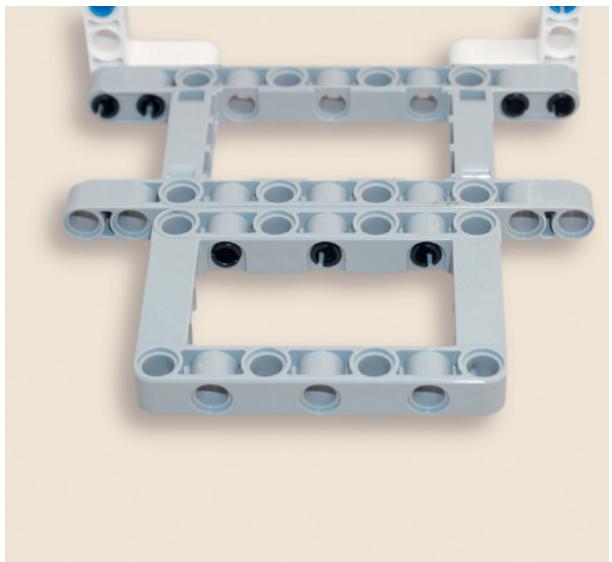
6. Надень на свободные концы синих и чёрных штифтов серую балку № 5 первым, вторым, четвёртым и пятым модулями.



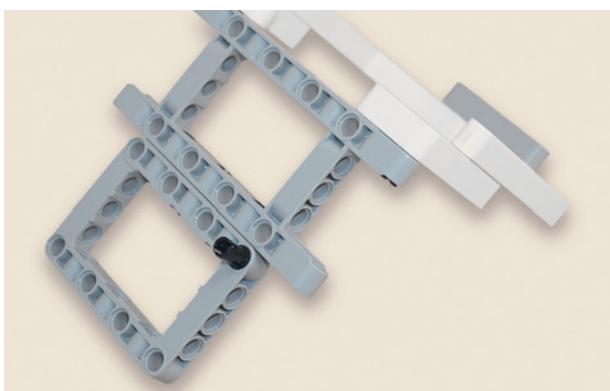
7. Поверни деталь рамкой к себе. Вставь три чёрных штифта в третий, четвёртый и пятый модули длинной стороны серой рамки.



8. Надень на чёрные штифты серую 5 × 7-модульную рамку длинной стороной вторым, четвёртым и шестым модулями.



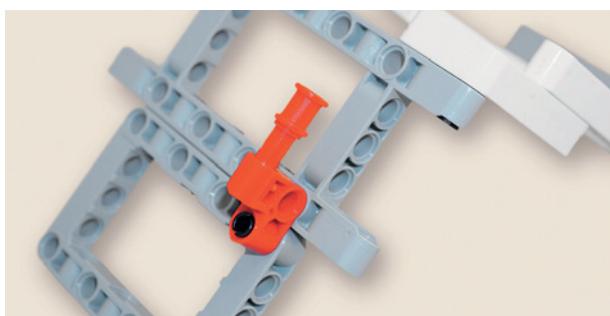
9. Теперь поверни подготовленную конструкцию из серых рамок так, чтобы белые балки располагались в сторону от тебя. Затем возьми один чёрный штифт и вставь его в седьмой модуль верхней длинной стороны рамки.



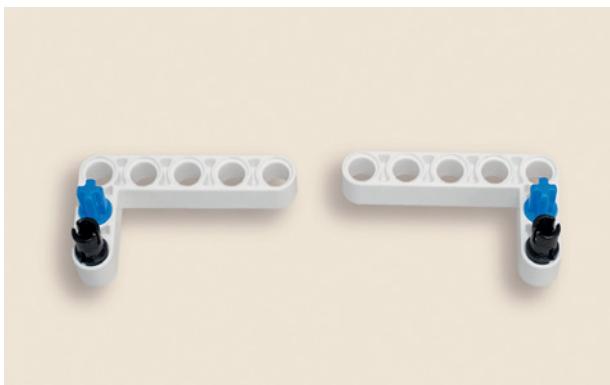
10. Надень на свободный конец чёрного штифта красную поперечную балку так, чтобы перпендикулярный модуль был наверху.



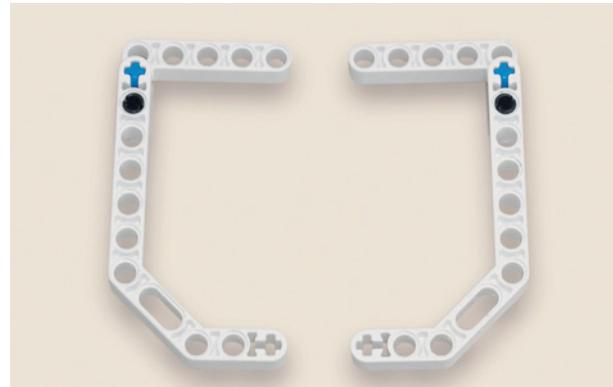
11. Вставь в верхнее крепление красной балки один 3-модульный красный штифт. Эта балка тебе понадобится для последующей сборки крепления большого мотора — привода дозатора корма (запирающего механизма).



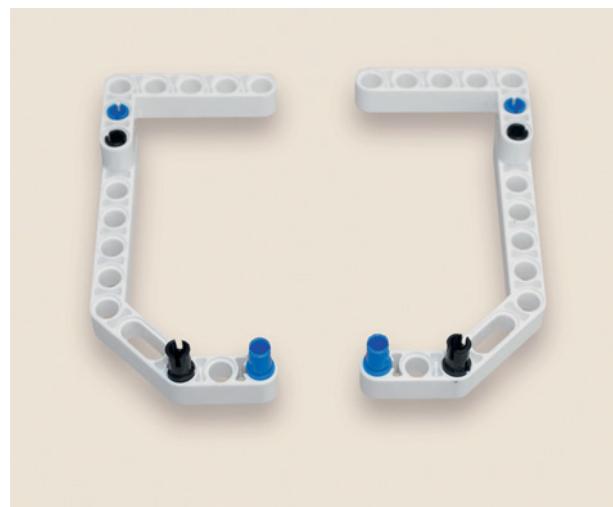
12. Подготовь ещё одну деталь для держателя ёмкости для корма. Для этого возьми две белые угловые балки и расположи их таким образом, чтобы короткие части балок оказались с разных сторон. Вставь в крайние модули коротких сторон балок по чёрному штифту, а во вторые модули по одному синему 2-модульному штифту.



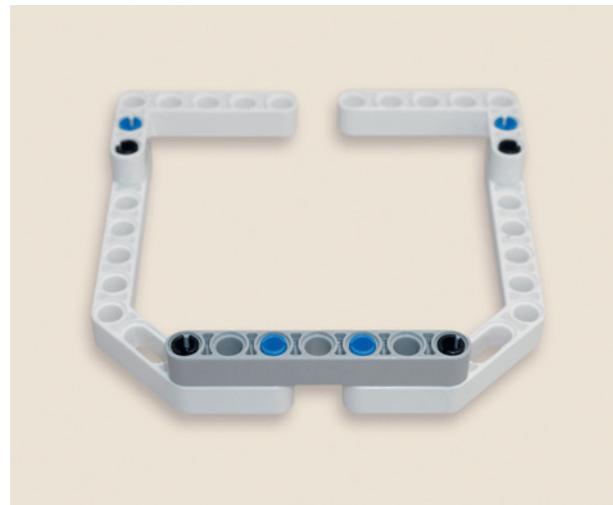
13. Надень на свободные концы синих и чёрных штифтов по одной двойной угловой балке первыми и вторыми модулями длинных сторон балок, чтобы короткие стороны балок были направлены внутрь конструкции.



14. Разверни конструкцию, как на рисунке. Вставь в крайние модули коротких сторон двойных угловых балок два синих 2-модульных штифта, затем установи по одному чёрному штифту слева в первый, а справа в третий модули коротких сторон угловых балок.

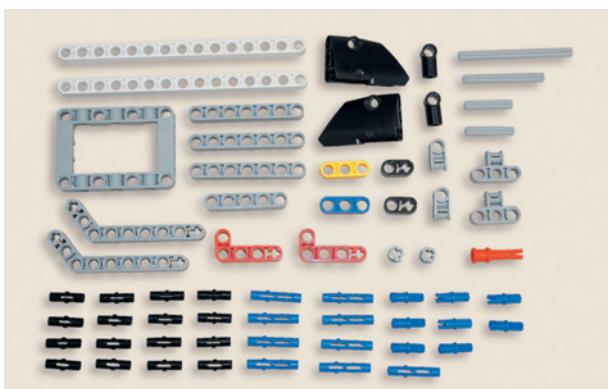


15. Надень на эти штифты серую балку № 7 первым, третьим, пятым и седьмым модулями.



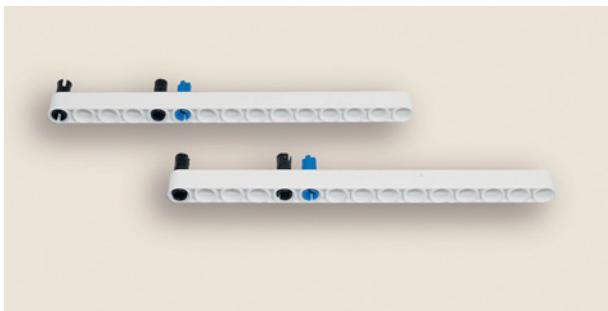
Ура! Теперь корм точно никуда не денется. Подобное крепление достаточно универсально. Перейдём к автоматике.

ШАГ 3. СБОРКА ПРИВОДА ДОЗАТОРА КОРМА



Детали для сборки:

- рамка 5×7 , серая, 1x;
- балка № 15, белая, 2x;
- балка № 7, серая, 3x;
- балка № 5, серая, 1x;
- балка № 3, жёлтая, 1x;
- балка № 3, синяя, 1x;
- балка с поперечным отверстием, 2-модульная, чёрная, 2x;
- балка угловая, 3×7 -модульная, серая, 2x;



- прямоугольная балка, 2×4 -модульная, красная, 2x;
- ось № 7, серая, 1x;
- ось № 5, серая, 1x;
- ось № 3, серая, 2x;
- поперечный блок, 3×2 -модульный, серый, 2x;
- поперечный блок, 2-модульный, серый, 2x;
- угловой блок, 1, 0° , чёрный, 2x;
- втулка, 1-модульная, серая, 2x;
- левая криволинейная панель 3×5 , чёрная, 1x;
- правая криволинейная панель 3×5 , чёрная, 1x;
- соединительный штифт, 2-модульный, чёрный, 16x;
- соединительный штифт, 2-модульный, синий, 8x;
- соединительный штифт, 3-модульный, синий, 9x;
- соединительный штифт со втулкой, 3-модульный, красный, 1x.

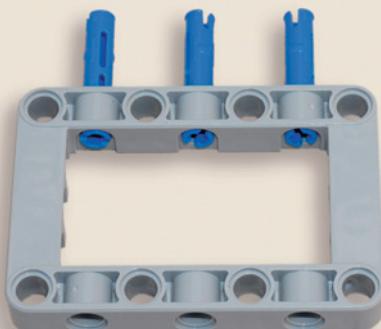
1. Собери дозатор для кормушки с приводом! Для этого возьми две белые балки № 15 и вставь в каждую по два чёрных штифта в первые и пятые модули, а после возьми два синих 2-модульных штифта и вставь их по одному в каждую балку в шестые модули.
2. Далее возьми четыре чёрных штифта и вставь их снизу по два в каждую балку в восьмые и десятые модули.



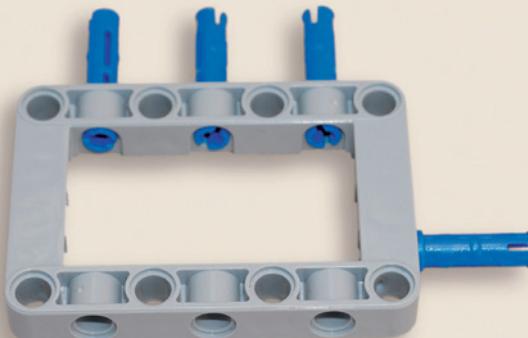
3. Надень на свободные концы этих штифтов по одной чёрной 3 × 5-модульной криволинейной панели первым и третьим модулями зеркально друг другу.



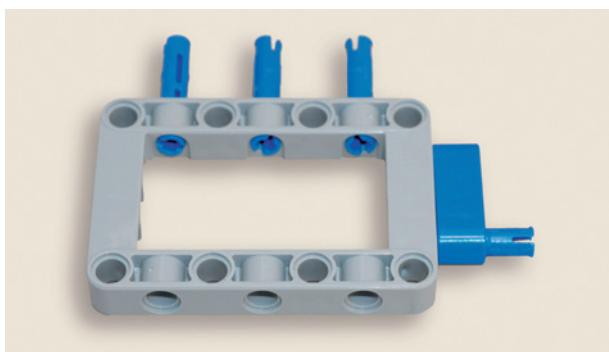
4. Вставь по длинной стороне серой рамки 5 × 7 сверху во второй, четвёртый и шестой модули по одному синему 3-модульному штифту короткими концами.



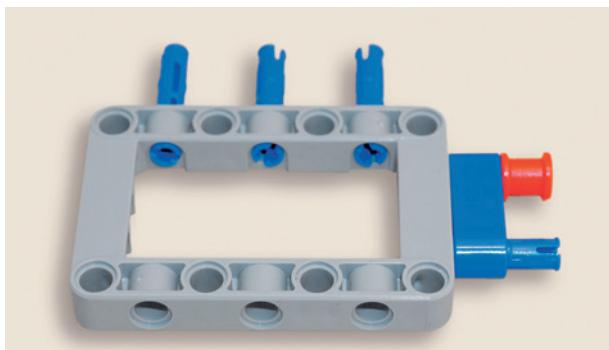
5. Вставь синий 3-модульный штифт коротким концом в правую короткую сторону серой рамки в четвёртый модуль.



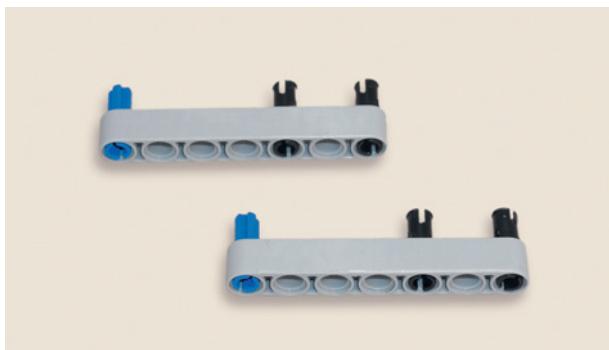
6. Надень синюю балку № 3 третьим модулем на синий штифт до упора.



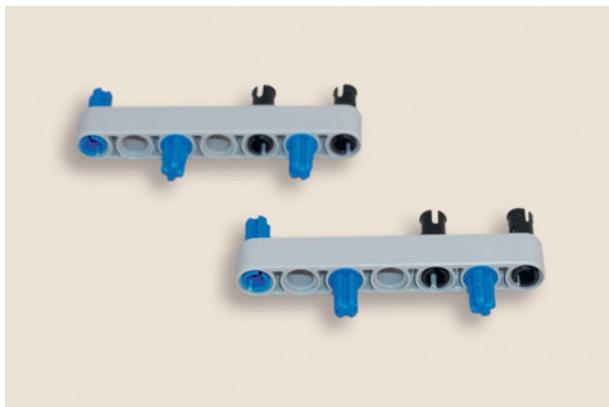
7. Вставь в первый модуль синей балки № 3 до упора один красный 3-модульный штифт.



8. Теперь собери рамку дозатора с затвором. Для этого возьми две серые балки № 7 и вставь по одному синему 2-модульному штифту в первые модули, а в пятые и седьмые модули — по одному чёрному штифту.



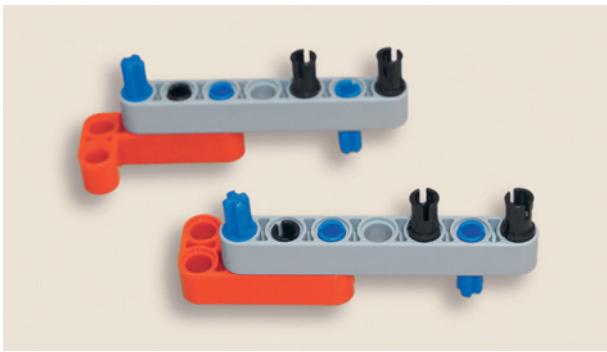
9. Вставь по два синих 2-модульных штифта снизу в третью и шестые модули серых балок № 7.



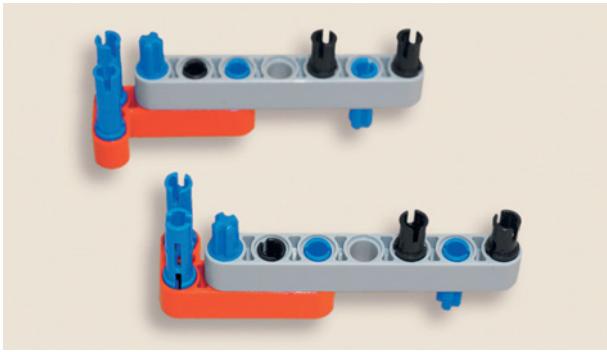
10. Вставь по одному чёрному штифту снизу в каждую балку во вторые модули.



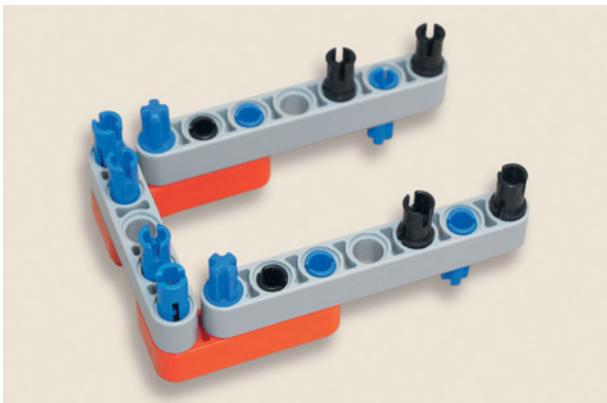
11. Надень снизу на свободные концы синих и чёрных штифтов по одной красной прямоугольной 2 × 4-модульной балке третьим и четвёртым модулями длинных сторон так, чтобы короткие стороны были направлены друг на друга.



12. Установи в короткие стороны красных балок по два синих 3-модульных штифта короткими концами вниз.



13. Надень на свободные концы синих 3-модульных штифтов серую балку № 5 первым, вторым, четвёртым и пятым модулями.



14. Надень на свободные концы синих 2-модульных штифтов, установленных в первых модулях серых балок № 7, и синих штифтов в первых модулях балки № 5 по одной 2-модульной чёрной балке с поперечным отверстием, как показано на рисунке.



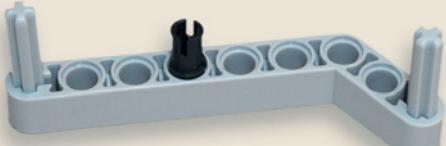
15. Надень на оставшиеся в серой балке № 5 синие штифты первым и третьим модулями желтую балку № 3.



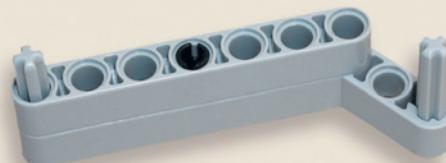
16. Теперь возьми два поперечных серых 3 × 2-модульных блока и надень их по одному в серые балки № 7 на свободные концы чёрных штифтов так, чтобы крестовые модули блоков были направлены наружу получившейся рамки.



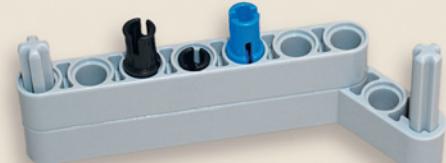
17. Собери затвор дозатора корма. Поверни угловую 3×7 -модульную балку так, чтобы короткая часть находилась справа снизу. В первый модуль балки вставь серую ось № 3, а в четвёртый модуль — чёрный штифт. Возьми ещё одну серую ось № 3 и вставь её в крайний модуль короткой стороны этой балки.



18. Надень на свободные концы оси и чёрного штифта первым и четвёртым модулями серую балку № 7.



19. Затем вставь в третий модуль серой балки № 7 чёрный штифт, а в пятый модуль серой балки № 7 установи длинным концом синий 3-модульный штифт.



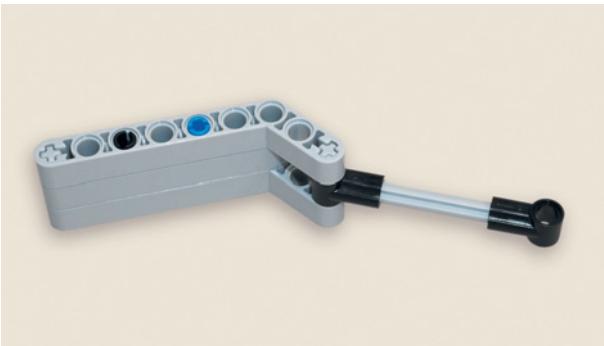
20. Настало время собрать тягу привода затвора кормушки. Для этого надень на серую ось № 5 с обоих концов по одному чёрному угловому блоку так, чтобы отверстия на блоках были направлены в одну сторону.



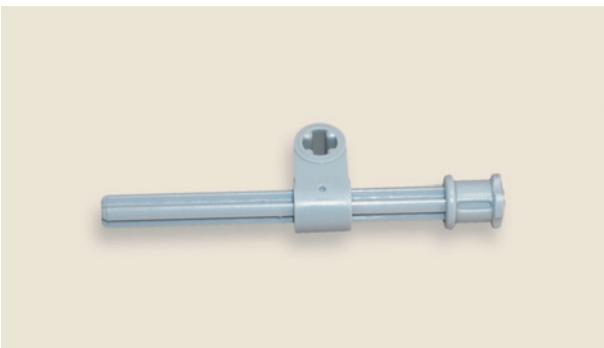
21. Надень свободным модулем связку на серую ось, установленную в крайнем модуле короткой стороны детали.



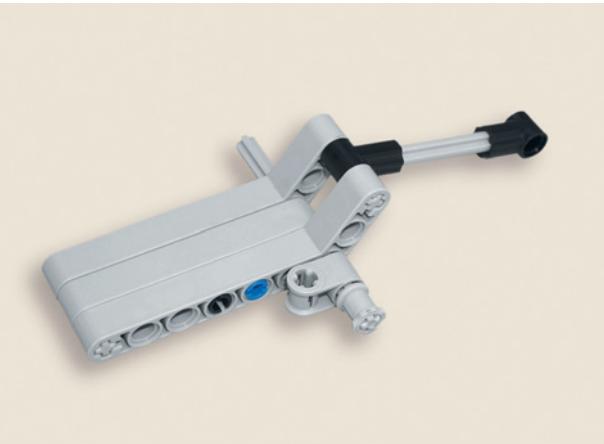
22. Надень сверху ещё одну серую угловую 3×7 -модульную балку, чтобы все модули совпали. Ура!



23. Теперь возьми серую ось № 7 и надень на неё серую 1-модульную втулку. Затем надень 3-модульный поперечный блок круглым модулем. Будь внимателен!



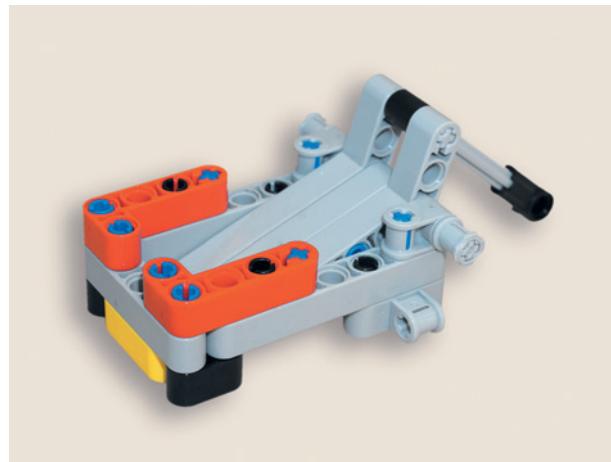
24. Продень ось с блоком в седьмой модуль длинной стороны 3×7 -модульной угловой балки.



25. Надень на оставшийся свободный конец серой оси ещё один серый 2-модульный поперечный блок и серую 1-модульную втулку, как ты уже делал ранее.

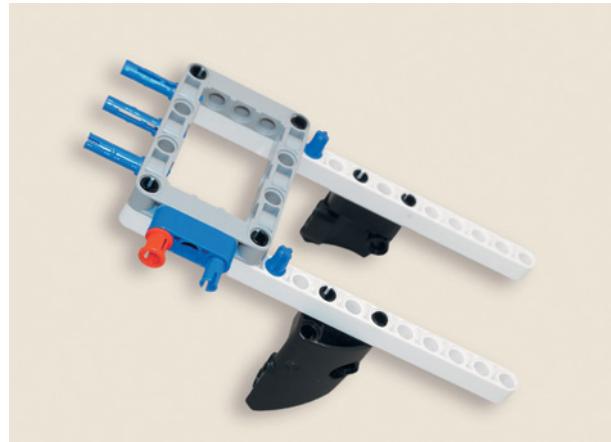


26. Собери воедино подготовленную рамку дозатора и затвор кормушки. Переверни рамку красными балками вверх, а на свободные концы синих 2-модульных штифтов установи крестообразные модули поперечных блоков так, чтобы ось дозатора располагалась справа.



Ты подготовил почти все детали для сборки дозатора! Осталось его собрать.

27. Для этого возьми две белые балки № 15 с дополнительными деталями на них и надень на свободные концы чёрных штифтов в первых и пятых модулях серую рамку держателя, как показано на рисунке.



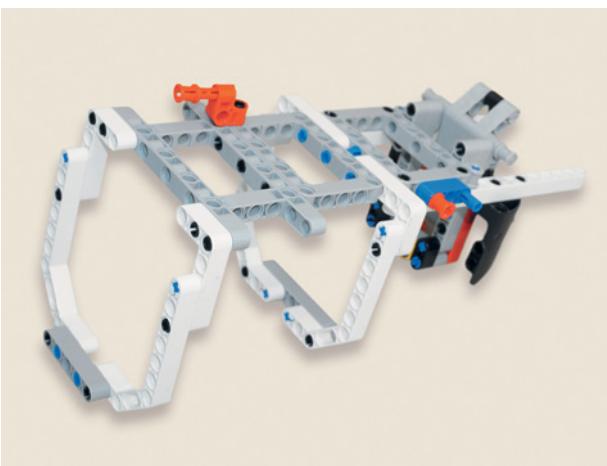
28. Надень рамку дозатора свободными крестообразными модулями по перечных блоков на синие штифты в белых балках № 15 справа от первой серой рамки так, чтобы рамка дозатора была направлена вниз. Руководствуясь рисунком.



29. Надень ранее подготовленный держатель ёмкости с кормом на свободные концы синих 3-модульных штифтов серой рамки слева четвёртым модулем белой угловой балки, а справа — вторым модулем. Для удобства можешь немного развернуть конструкцию.



30. Нарасти держатель! Возьми верхнюю часть держателя и надень серой рамкой на подготовленную конструкцию вторым, четвёртым и шестым модулями на свободные концы синих 3-модульных штифтов.



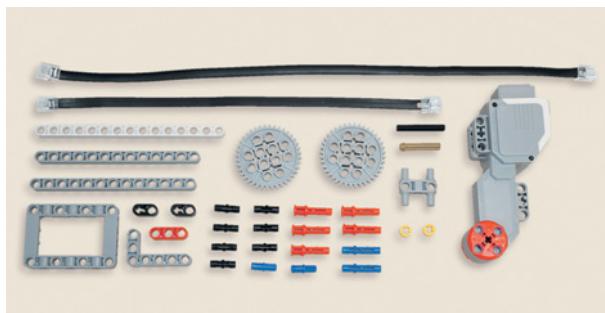
Ура! Теперь ёмкость для корма точно никуда не денется! Давай установим держатель и подключим привод дозатора к блоку EV3.

ШАГ 4. ФИНАЛЬНАЯ СБОРКА КОНСТРУКЦИИ КОРМУШКИ И ПОДКЛЮЧЕНИЕ ПРОВОДОВ

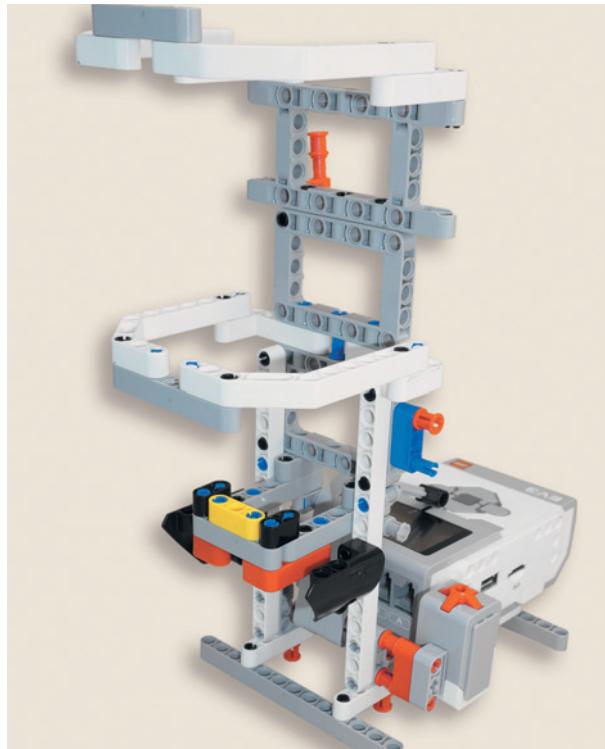
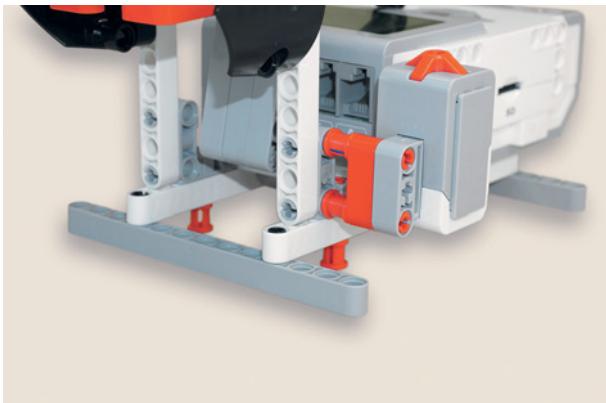
Детали для сборки:

- рамка 5×7 , серая, 1x;
- балка № 15, белая, 1x;
- балка № 13, серая, 2x;
- балка № 3, красная, 1x;
- балка с поперечным отверстием, 2-модульная, чёрная, 2x;
- балка угловая, 3 × 5-модульная, серая, 1x;
- ось № 4, чёрная, 1x;
- ось № 3 с шипом, темно-бежевая, 1x;
- большой мотор, 1x;
- двойной соединительный штифт, 3 × 3-модульный, серый, 1x;
- колесо зубчатое, 40 зубьев, серое, 2x;
- втулка, $\frac{1}{2}$ -модульная, жёлтая, 2x;
- соединительный штифт, 2-модульный, чёрный, 7x;

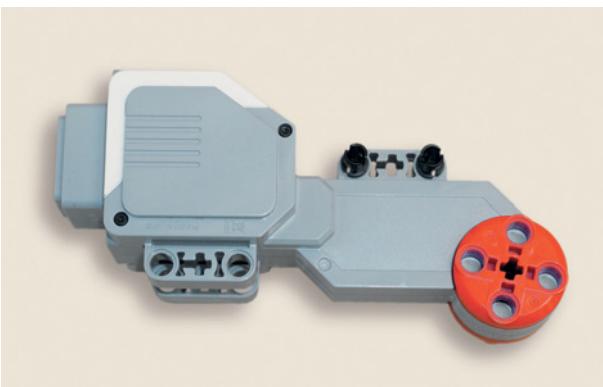
- ↗
- соединительный штифт, 2-модульный, синий, 2x;
 - соединительный штифт, 3-модульный, синий, 2x;
 - соединительный штифт со втулкой, 3-модульный, красный, 5x;
 - кабель, 35 см, чёрный, 1x;
 - кабель, 25 см, чёрный, 1x.



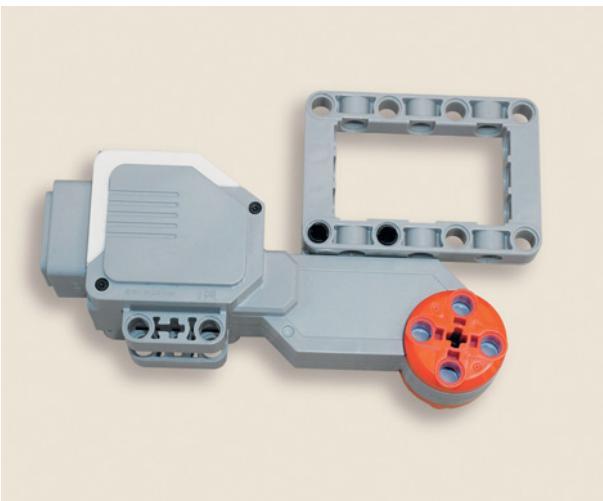
1. Возьми конструкцию держателя емкости для корма и присоедини её к собранному ранее основанию с блоком EV3. Надень концы белых балок № 15 на свободные штифты серых угловых соединительных штифтов первыми и третьими модулями.



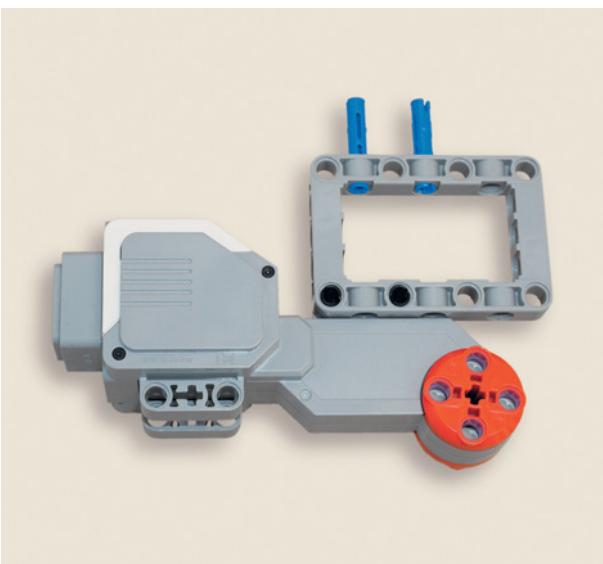
Теперь собери крепление большого мотора к дозатору корма.



2. Для этого возьми большой мотор и поверни его так, чтобы разъём мотора находился слева, а крепление возле привода находилось сверху справа. Вставь в крепление возле привода два чёрных штифта в первый и третий модули.

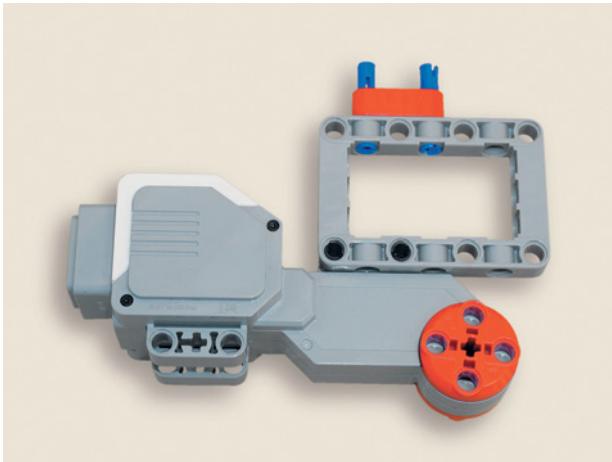


3. Надень серую 5×7 -модульную рамку первым и третьим модулями сверху на чёрные штифты.



4. Вставь два синих 3-модульных штифта короткими концами во второй и четвёртый модули серой рамки.

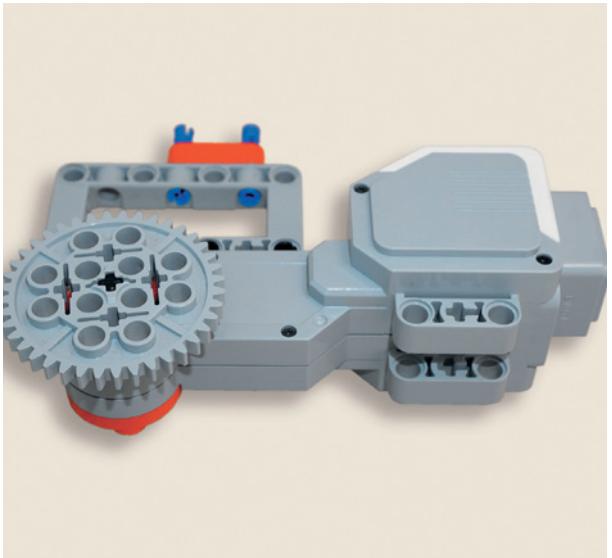
5. Надень на их концы красную балку № 3 первым и третьим модулями до упора.

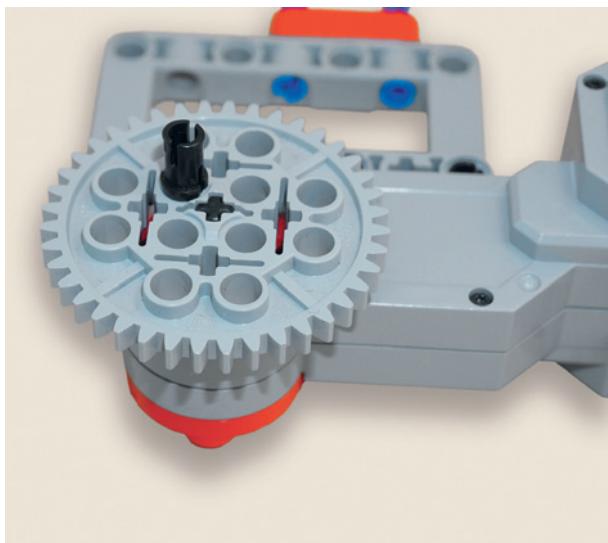


6. Поверни большой мотор так, чтобы разъём находился справа, а серая рамка — слева наверху. Вставь чёрную ось № 4 в привод большого мотора.

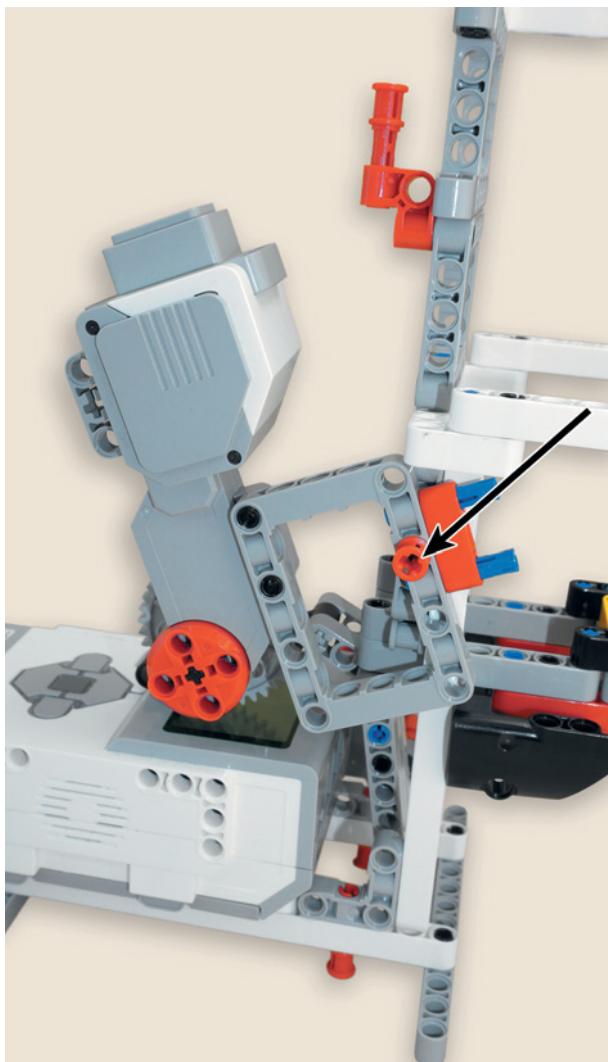


7. Надень на свободный конец оси зубчатое колесо на 40 зубьев.





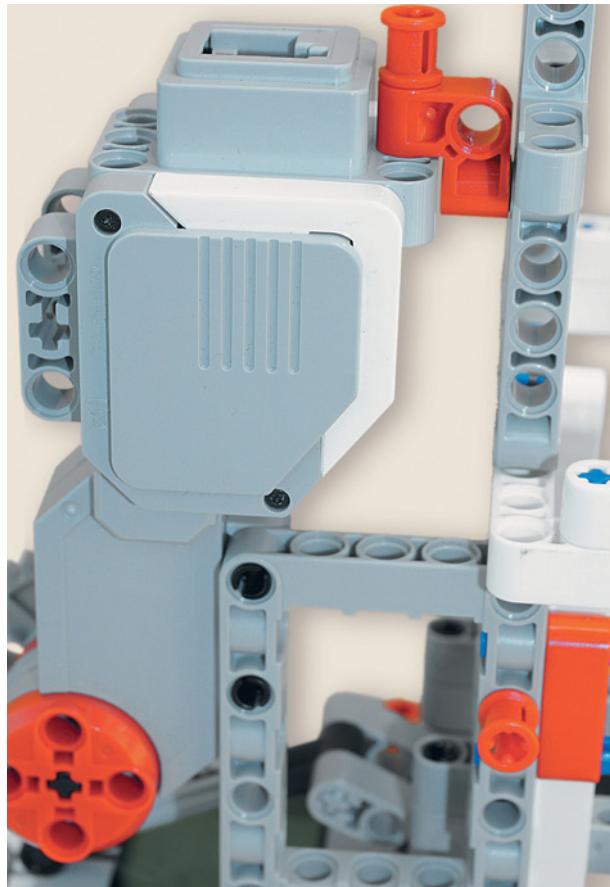
8. Вставь в первый модуль от центра на зубчатом колесе один чёрный 2-модульный штифт, как показано на рисунке.



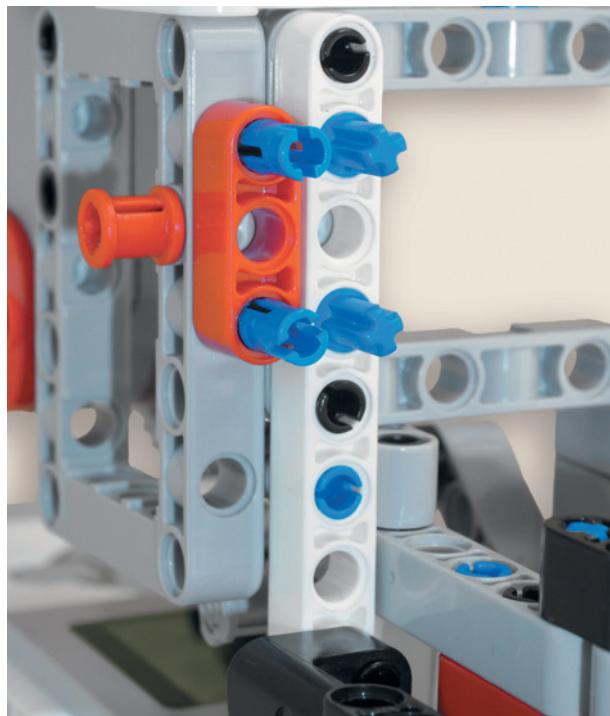
9. Поверни собранную конструкцию так, чтобы держатель для ёмкости находился справа, а блок EV3 – слева. Приложи к держателю большой мотор с серой рамкой: привод должен находиться снизу, а разъём большого мотора – сверху. Вставь в раму держателя насеквоздь в третий модуль длинной стороны серой рамки один красный 3-модульный штифт.



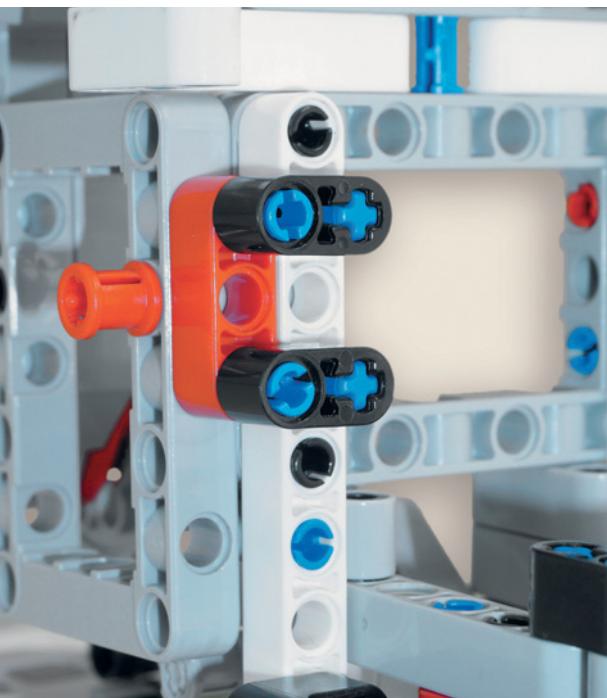
10. Зафиксируй сверху возле разъёма большого мотора конструкцию красным 3-модульным штифтом, установленным в красную угловую балку, — просто защёлкни крепление!



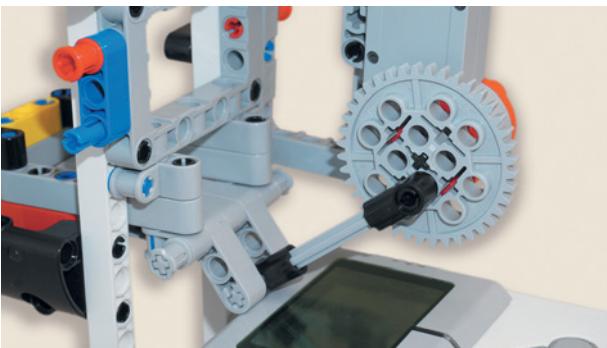
11. Теперь разверни собранную конструкцию дозатором к себе. Вставь напротив свободных концов синих штифтов красной балки № 3 по одному синему 2-модульному штифту в белые балки № 15 рамы дозатора.



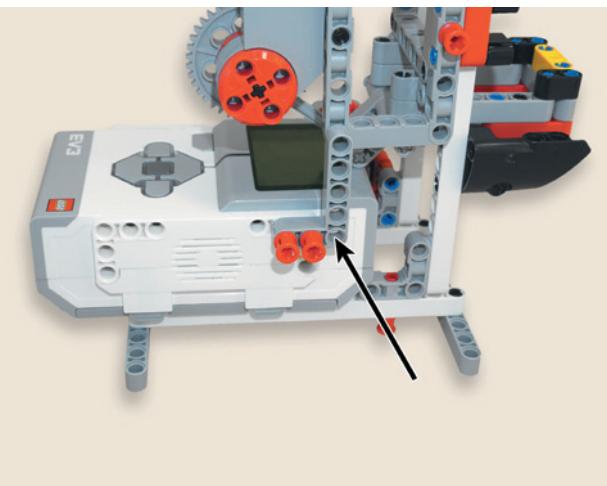
12. Надень на синие штифты в красной и белой балках по одной чёрной 2-модульной балке с поперечным отверстием, как показано на рисунке.



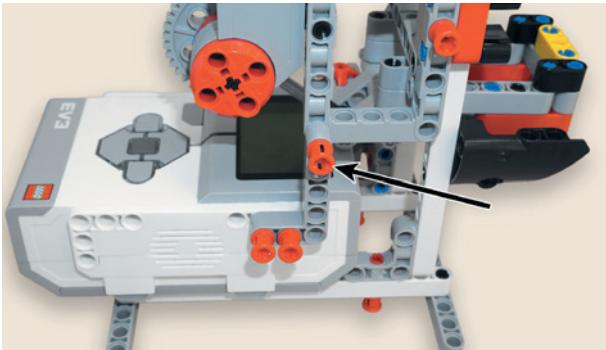
13. Надень на свободный конец серого штифта в зубчатом колесе чёрный блок тяги привода затвора. Осталось совсем немного!



14. Поверни кормушку так, чтобы модуль EV3 находился слева. Приложи к горизонтальному боковому креплению на блоке серую 3 × 5-модульную угловую балку так, чтобы длинная сторона балки была направлена вверх. Вставь в короткую часть балки два красных 3-модульных штифта в первый и второй модули насеквоздь в крепление блока EV3 во второй и третий модули соответственно.



15. Вставь ещё один красный 3-модульный штифт насквозь в первый модуль серой угловой балки, соединив с серой рамкой в её седьмом модуле.



16. Снова переверни конструкцию, но теперь модулем EV3 направо. Надень первым модулем белую балку № 15 на свободный конец синего штифта в синей балке № 3.

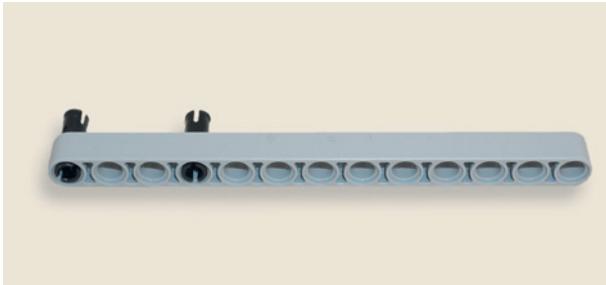


17. Приложи крайний модуль белой балки № 15 к боковому креплению на модуле EV3 и вставь красный 3-модульный штифт во второй модуль горизонтального крепления.

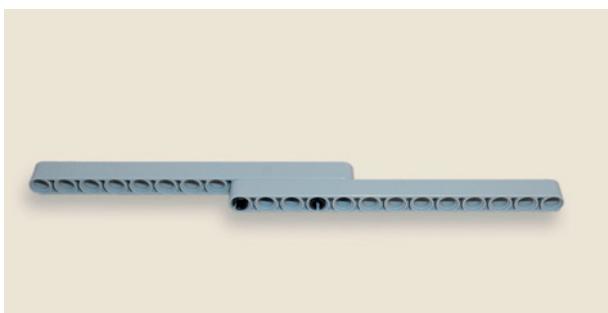


Питомцу будет неудобно нажимать маленькую кнопку. Собери педаль для выдачи корма по требованию.

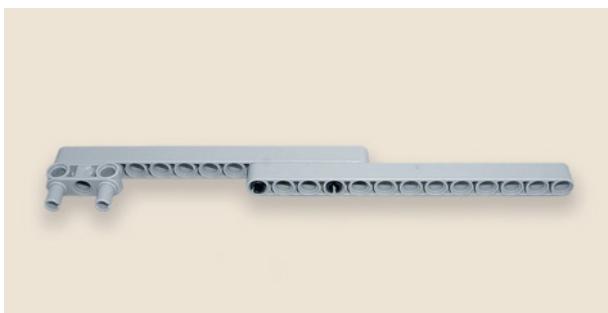
18. Для этого возьми серую балку № 13 и вставь в первый и четвёртый модули по одному чёрному штифту.



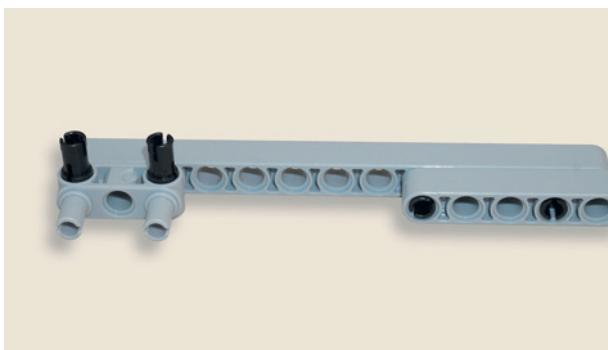
19. Надень на свободные концы чёрных штифтов серую балку № 13 девятым и двенадцатым модулями.



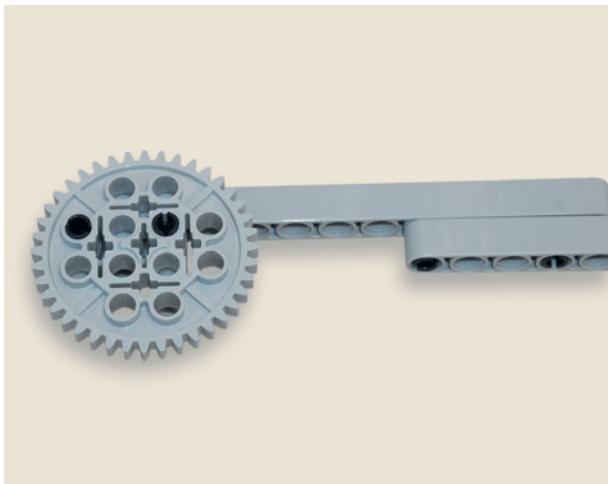
20. Вставь в первый и третий модули этой балки двойной соединительный 3 × 3-модульный штифт.



21. Вставь в свободные модули двойного соединительного 3 × 3-модульного штифта по одному чёрному штифту.



22. Надень на чёрные штифты третьим и пятым модулями зубчатое колесо на 40 зубьев.



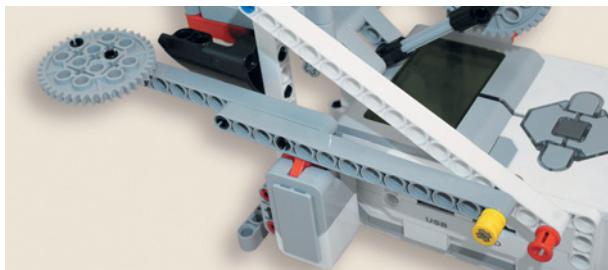
23. Возьми темно-бежевую ось № 3 с шипом и вставь её в белую балку № 15 в тринадцатый модуль. Упор оси должен располагаться со стороны блока EV3.



24. Надень на свободный конец оси подготовленную педаль крайним модулем серой балки № 13, как показано на рисунке.

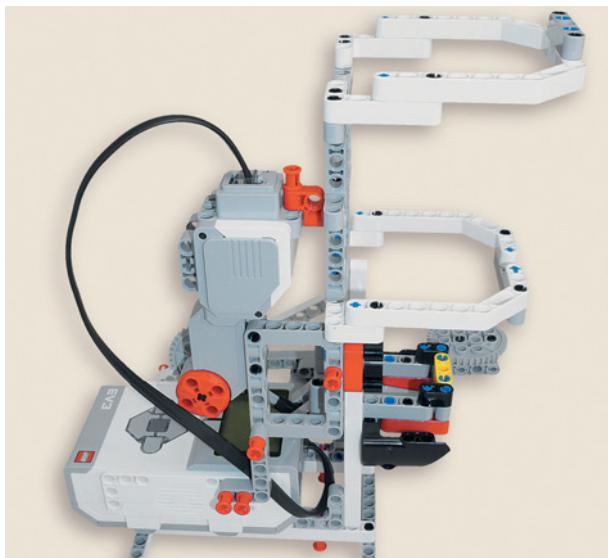


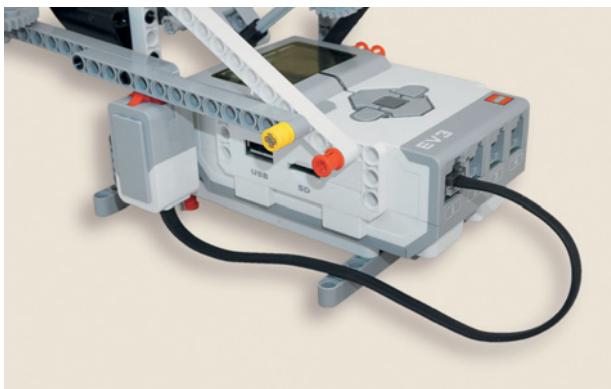
25. Зафиксируй серую балку с помощью двух жёлтых $\frac{1}{2}$ -модульных втулок.



Настало время подключать провода!

26. Возьми провод 35 мм и вставь один конец в порт С на блоке EV3, а второй конец в разъём большого мотора.

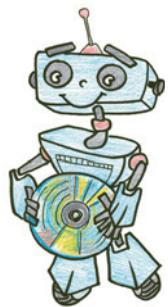




27. Теперь возьми провод 25 мм и вставь один конец в датчик касания, а второй в разъём № 1 на модуле EV3.

Поздравляем! Ты справился со сложной задачей. Давай скорее программируй, чтобы питомец точно не остался голодным!





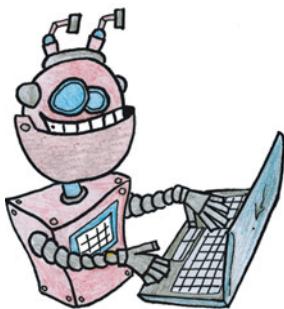
Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел Загрузки -> LEGO® MINDSTORMS® Education EV3, где ты сможешь скачать установочный файл LME-EV3.

Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

Этап 4. Создание программы для кормушки

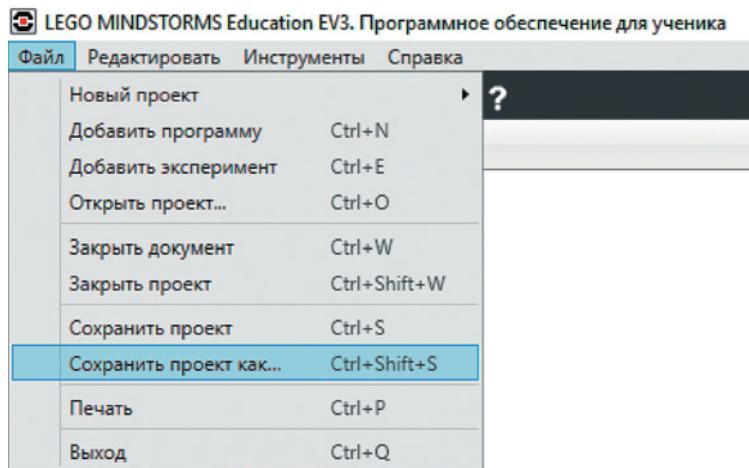


ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

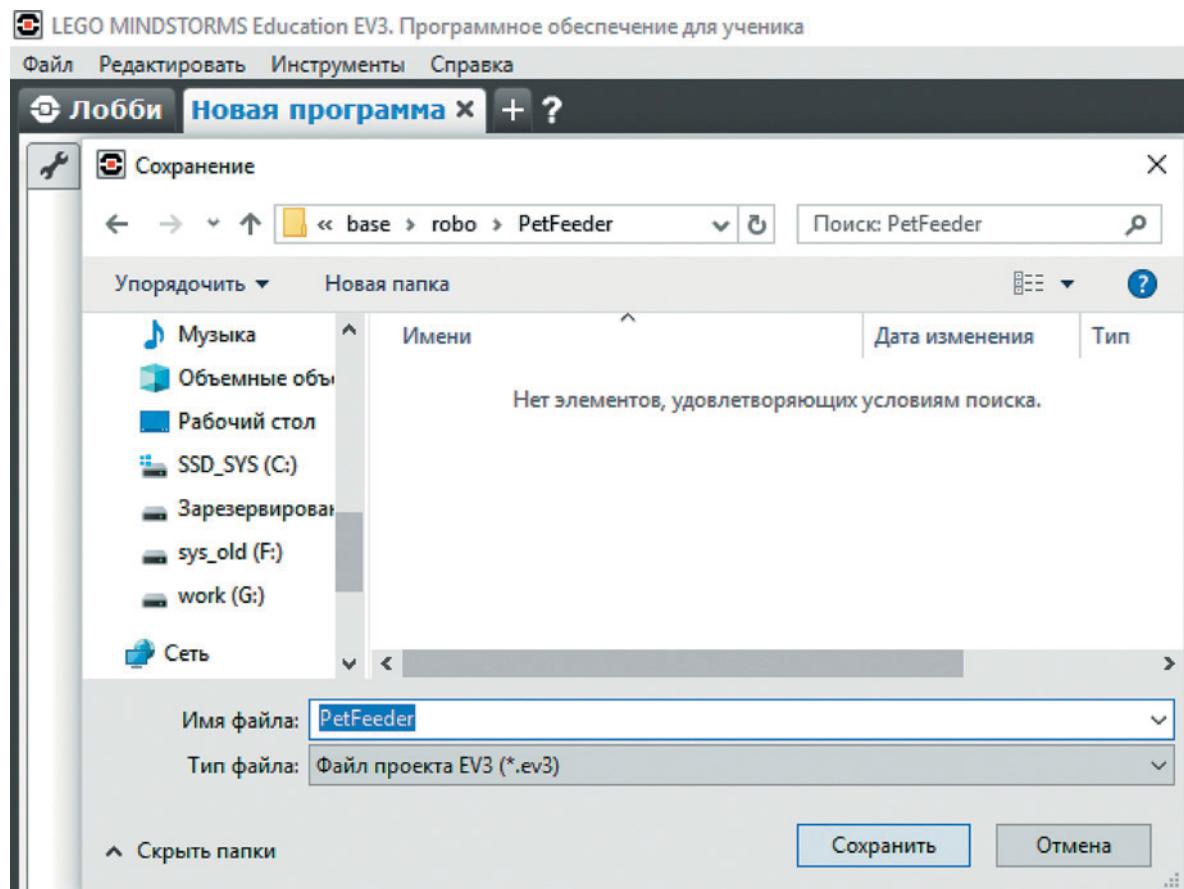
1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, то не нужно искать его в компьютере.



1. Для начала сохрани свой проект в памяти компьютера, чтобы не потерять его. Для этого нажми меню **Файл** в левом верхнем углу окна программы и выбери в контекстном меню пункт **Сохранить проект как...**
2. Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла: PetFeeder** (в переводе с английского — кормушка питомца) в нижней части окна и нажми **Сохранить**.



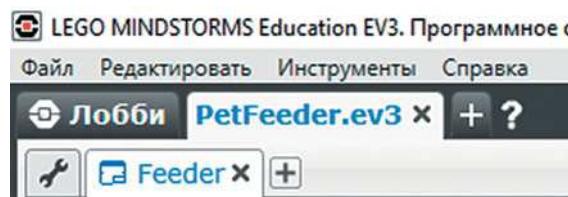
Именно так будет называться папка в памяти программируемого модуля EV3, где будут храниться все программы, которые ты напишешь в рамках этого проекта.

Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они пишут, например, каждое слово с большой буквы (как это сделал ты) или добавляют нижнее подчёркивание.

Осталось дать имя самой программе (в нашем проекте будет только одна).

3. Для этого дважды кликни по имени **Program** в левом верхнем углу поля программы и задай новое имя: *Feeder* (в переводе с английского — кормушка), далее нажми клавишу **Enter**.



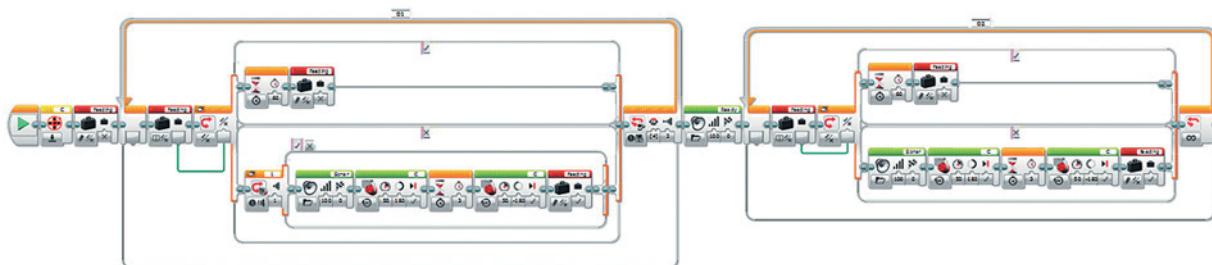
ЛОГИКА ПРОГРАММЫ

Для начала подумаем, как же должна работать твоя кормушка. Начальное положение, то есть стартовое, которое занимает автомат при запуске, должно быть строго зафиксировано — это одно из основных правил для автоматики и робототехники. Для кормушки это поднятый до упора запирающий механизм (положение «закрыто»), для чего требуется перевести мотор в нужное число градусов и сбросить энкодер — дать положению ярлык «старт» («нулевое положение»). Также необходимо установить в переменную-флаг «Кормление» значение «Нет».

Программа выполняется в бесконечном цикле, поскольку питомец может и должен есть не один раз в день. Внутри производится ожидание команды от блока управления — проверка состояния кнопки. Когда питомец нажимает кнопку, будучи не накормленным, воспроизводится короткий звуковой сигнал, и запирающий механизм переводится в положение «открыто». После трёх секунд механизм запирает ёмкость для корма.

В переменную «Кормление» заносится отметка, что питомец недавно поел. Если питомец поел, то производится ожидание нужного времени до следующего обеда, переменная «Кормление» сбрасывается.

Пока в переменной «Кормление» стоит положительный флаг, открытие запирающего механизма запрещено (следовательно, вся работа с кнопкой и механизмом должна быть помещена в отрицательную ветку условия). Добавим режим расписания. Пусть после нажатия на блоке кнопки «вверх» производится выход из ручного режима. Чтобы знать наверняка, что режим переключен, блок воспроизводит звук. Снова необходимо выполнять кормление и проверки в цикле. Если кормление не произведено (во флаге установлено положение «нет»), то высыпать порцию корма аналогично ручному режиму и поставить флаг о кормлении. Единственное, чем будет отличаться второй цикл, — это проверкой нажатия кнопки, точнее, отсутствием этого условия.



Автоматический режим можно реализовать внутри первого цикла, но это будет немного сложнее. Давай уже перейдём непосредственно к программированию. Готов?

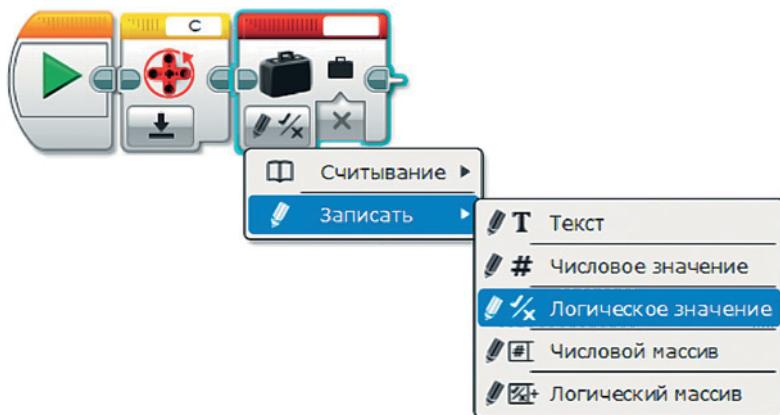
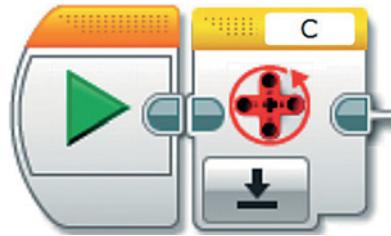
Внимание!

Количество секунд для поддержания положения «открыто» зависит от размера твоего питомца и корма. Если, допустим, ты положил в кормушку лакомство, то его порция должна быть намного меньше, чем порция основного корма.
Время между кормлениями также строго индивидуально!

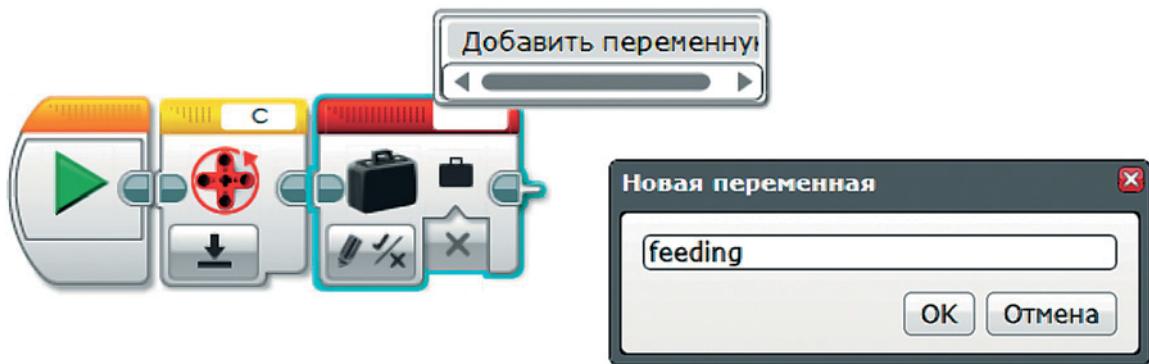
СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ КОРМУШКИ

Начинай составлять программу для автоматической кормушки в открывшемся окне проекта.

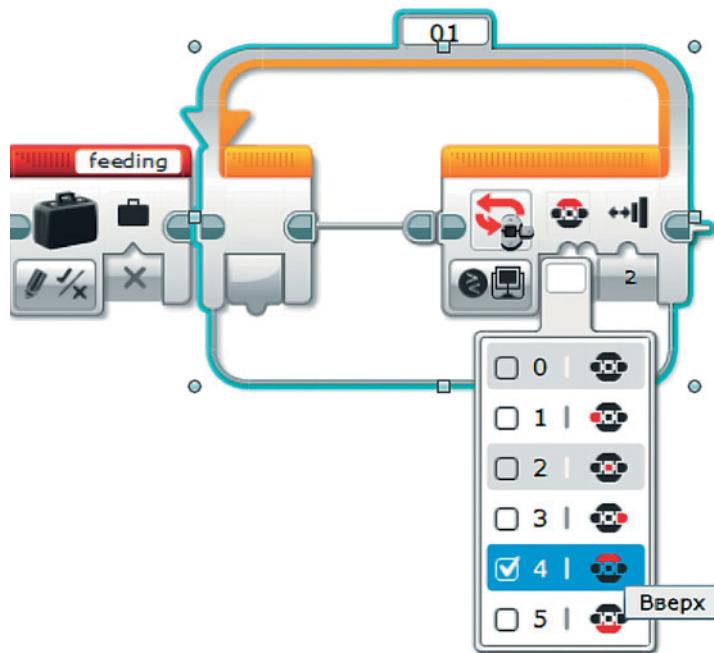
1. Начнём со сброса энкодера. Для этого сразу после блока старта вытащи жёлтый блок **Вращение мотора**. В настройках блока выбери **Сброс** и не забудь проверить, что наверху указан мотор на порте **C**.
2. Нужно выставить значение по умолчанию для флага. Вытащи красный блок **Переменная**, выбери в свойствах блока **Записать** и затем — **Логическое значение**.



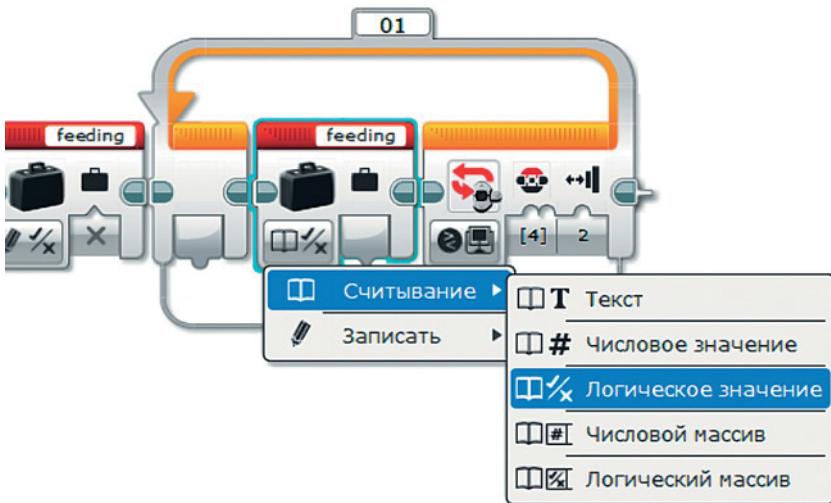
3. Место подготовлено, осталось создать саму переменную. Нажми на пустую область наверху блока и команду **Добавить переменную**. Дай ей имя *feeding* (в переводе с английского — кормление).



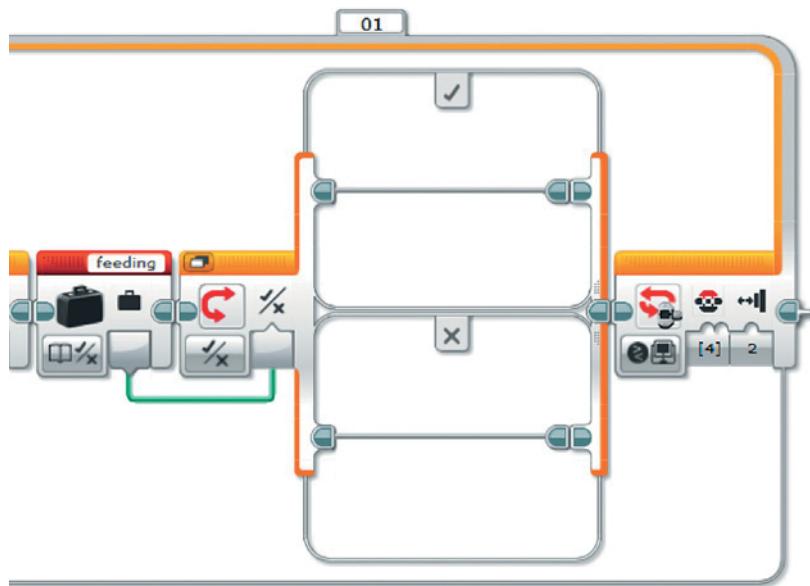
4. Теперь добавим область, где будет находиться ручной режим — цикл. Вытащи оранжевый блок **Цикл**. В качестве условия для выхода выбери **Кнопки управления модулем — Сравнение**. Среди идентификаторов поставь галочку только напротив четвёртого варианта, а в режиме касания установи **Щелчок**. Выход из цикла.



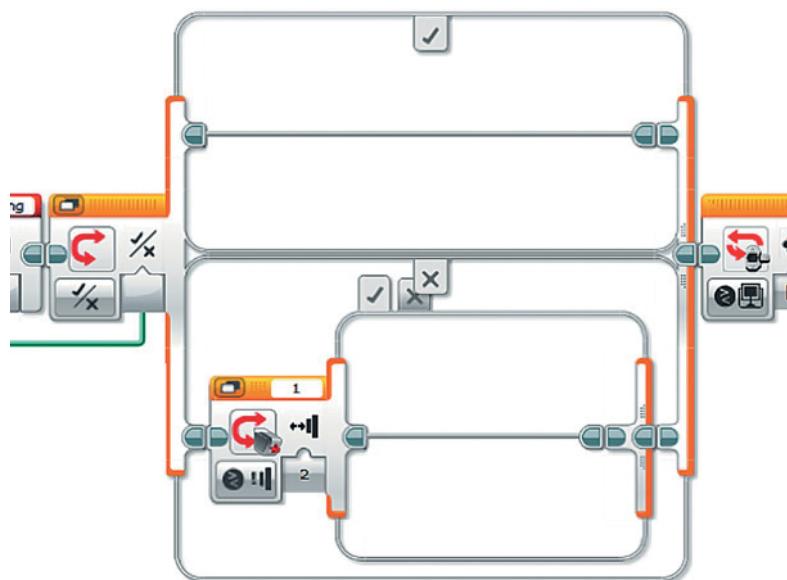
5. Теперь нужно реализовать выбор действия в зависимости от флага внутри переменной «Кормление». Для этого вытащи внутрь цикла красный блок **Переменная**, далее выбери **Считывание — Логическое значение**. Убедись, что справа наверху появилось имя *feeding*.



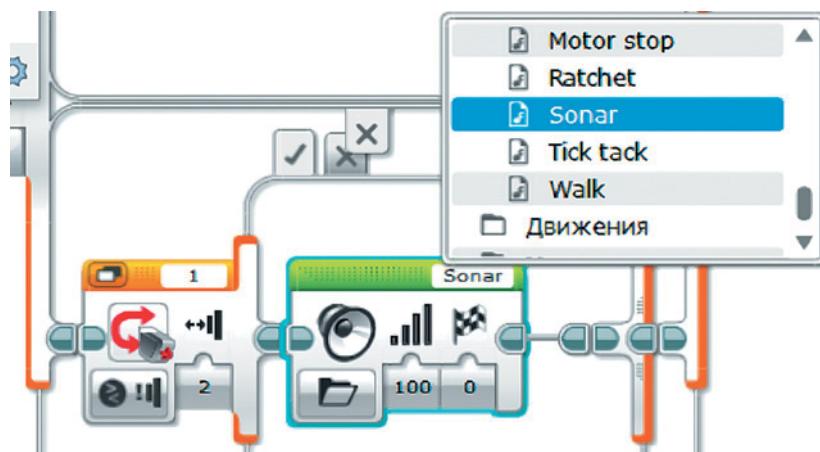
6. После переменной вытащи оранжевый блок **Переключатель**. Выбери в качестве условия **Логическое значение**. Соедини выход блока **Переменная** и вход **Переключателя**.



7. Отлично! Переключатель задаёт вопрос, был ли накормлен питомец. Давай начнём с отрицательной ветки, то есть когда питомец голоден. Тебе понадобится ещё один оранжевый блок **Переключатель**. Поскольку мы будем использовать его без отрицательной ветки, сразу же переключи его в режим вкладок, нажав на изображение окошечек в верхнем левом углу блока — **Переключиться на вид с вкладками**. В качестве условия для переключения должен быть **Датчик касания — Сравнение — Состояние**, а в качестве второго параметра — **Щелчок**. Убедись, что на блоке указан порт 1.



8. Начнём наполнять условие. Добавь звуковое оповещение, перетащив во вложенный переключатель зелёный блок **Звук**. Затем нажми на изображение папки наверху блока и выбери из библиотеки любой звук на свой вкус. Мы взяли **Механизмы — Sonar**.





Оповещение проведено, пора открывать запирающий механизм!

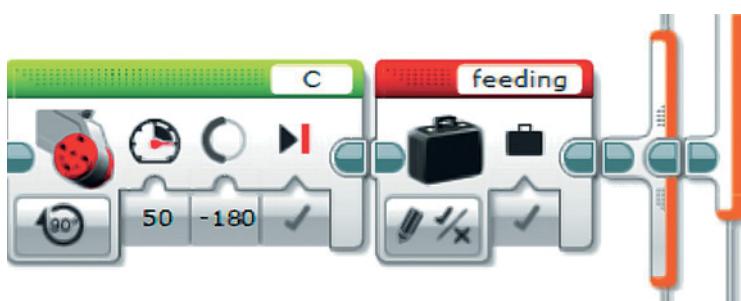
- Для этого вытащи зелёный блок **Большой мотор**. Выбери режим **Включить на количество градусов**. Выстави в качестве атрибутов мощность **50** и половину круга — **180** градусов. Убедись, что порт для мотора выбран верно и в окошке стоит **C**.



- Нужно немного подождать, чтобы высыпался корм. Используй оранжевый блок **Ожидание**. Мы уже упоминали, что время зависит от состава корма и самого питомца. Для теста мы поставили значение **3**. Сразу же добавь копию зелёного блока **Большой мотор**, но с отрицательным значением поворота в **-180** градусов.
Кстати, чтобы скопировать и вставить блок, выдели нужный блок мышкой и нажми классические сочетания клавиш **Ctrl+C** и **Ctrl+V**, а затем помести новый блок на нужное место.



- Здесь ты почти закончил! Осталось вытащить красный блок **Переменная**, выбрать **Записать логическое значение** и **Истина** в качестве аргумента. Проверь, что данные заносятся в *feeding*. Теперь кормушка знает, что питомец накормлен.



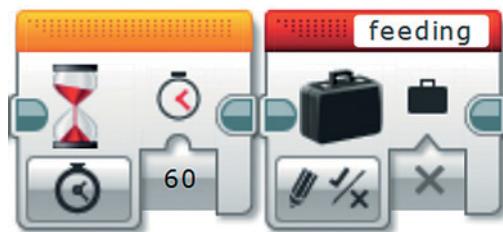
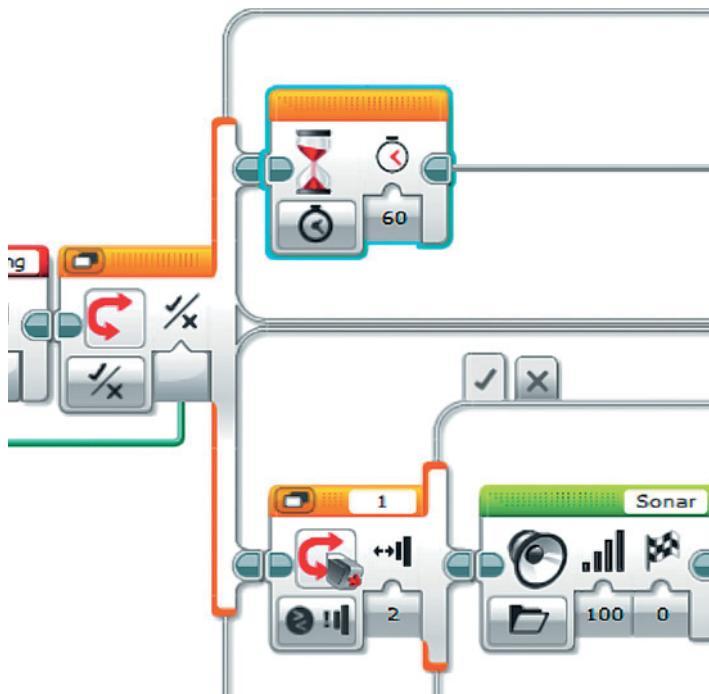
Ура! Питомец теперь сможет попросить еды у этой чудо-машины. Но давай оформим вторую ветку, положительную, что он не ел больше положенного.

12. В верхнюю ветку внешнего переключателя перетащи оранжевый блок **Ожидание**. В качестве времени используй подсчитанное значение в секундах с помощью формулы:

$$\text{Ожидание} = \text{часы} \cdot 60 \text{ минут} \cdot 60 \text{ секунд} = \text{часы} \cdot 3600$$

Например, если кота можно кормить раз в четыре часа (по словам ветеринара), то выстави значение $4 \cdot 3600 = 14\,400$ секунд.

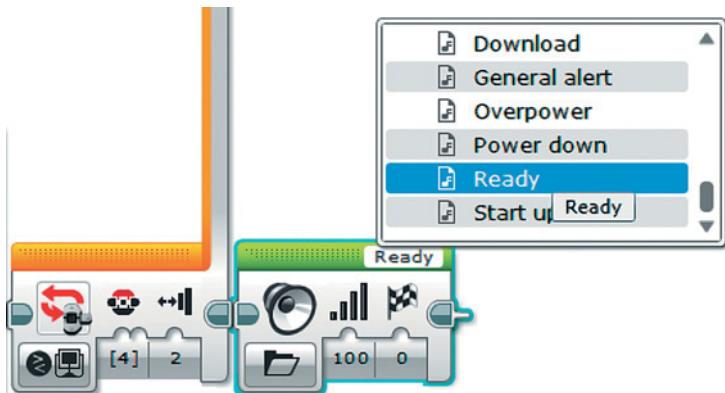
Для тестирования мы выставим минуту, то есть 60 секунд.



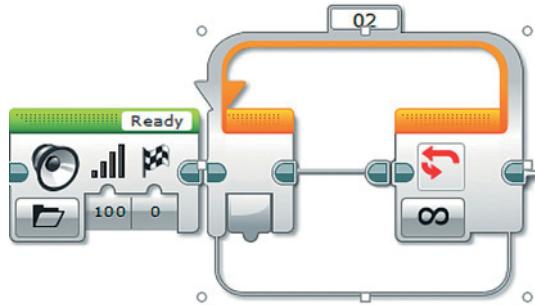
13. Когда пройдёт отведённое время, статус питомца надо сменить на «голодный». Для этого используй красный блок **Переменная** и режим **Запись – Логическое значение**. Убедись, что выбрана переменная *feeding*, и занеси значение **Ложь**.

Поздравляем! Ты закончил с ручным режимом!

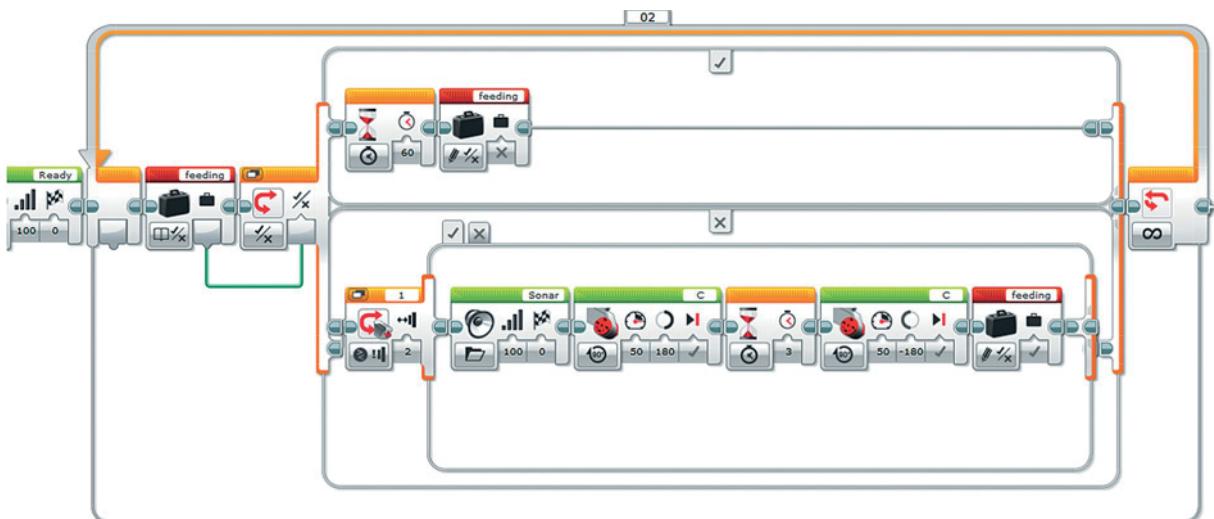
14. Теперь реализуй автоматический режим. После выхода из цикла добавь соответствующий сигнал для удобства использования кормушки. Помести зелёный блок **Звук** сразу после цикла. Мы использовали звук из папки **Система — Ready**.



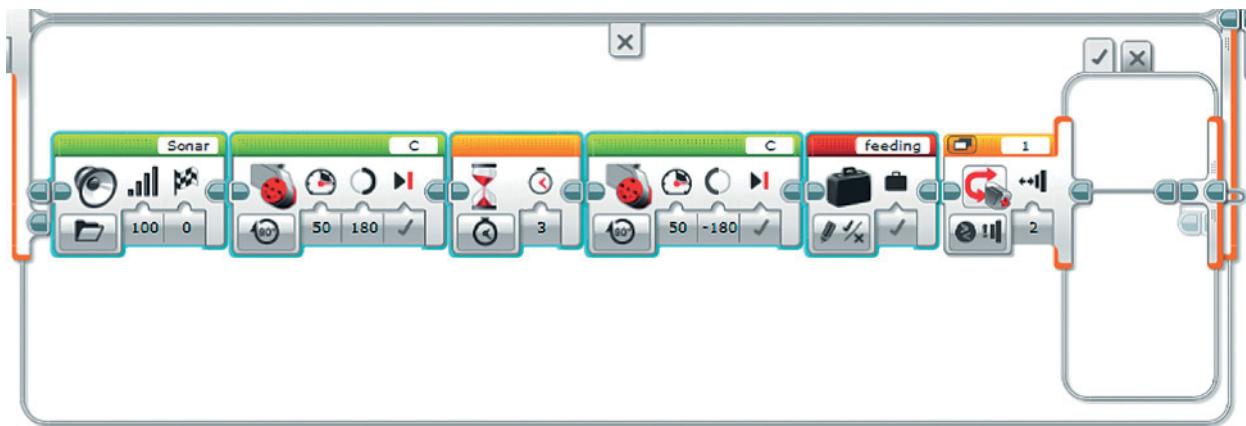
15. Перетащи на рабочую область оранжевый блок **Цикл** и оставь его без условия выхода — **бесконечным**. Переименуй цикл, чтобы не возникло ошибок. Для этого замени название в верхней части блока на **02**.



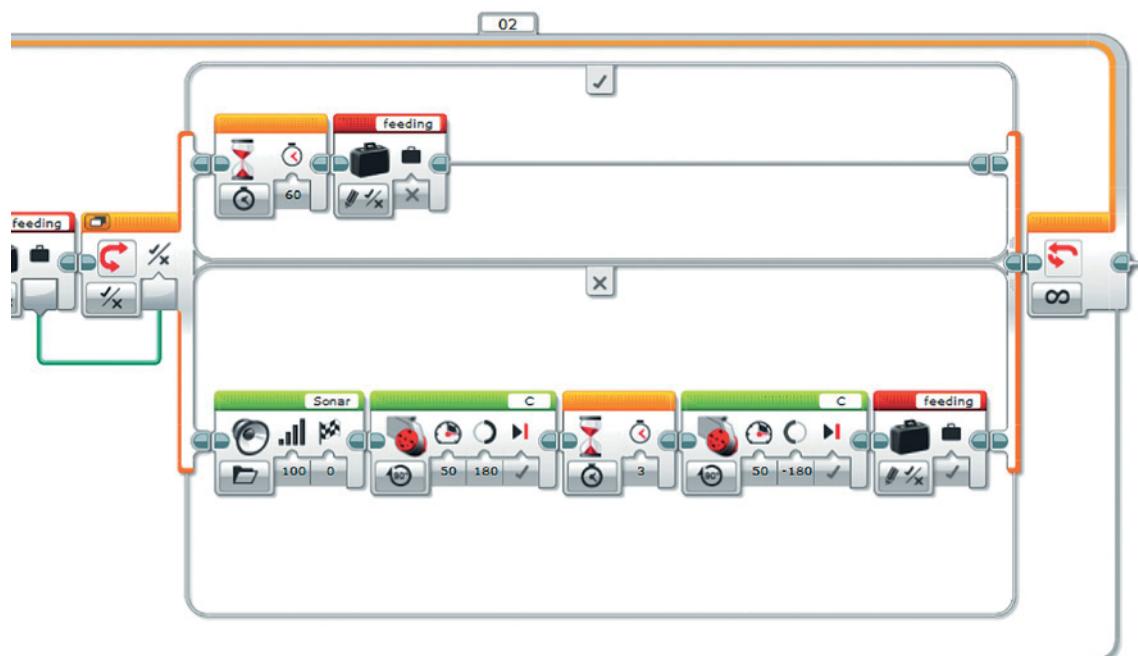
16. Скопируй содержимое цикла ручного режима. Для этого зажми клавишу **Ctrl** и нажми левой клавишей мыши на красный блок **Переменная** и внешний **Переключатель**. Нажми сочетание **Ctrl+C** и вставь с помощью **Ctrl+V** внутрь второго **Цикла**.



17. Убери лишнее. В автоматическом режиме кормление осуществляется без проверки условия на нажатие клавиши. Для этого уже знакомым тебе способом выдели вложенные команды **Переключателя** с условием нажатия кнопки, а затем поставь их перед ним.



18. Удали лишний блок **Переключатель**. Автоматический режим готов! Кстати, можешь установить в отрицательную ветку условия различные значения таймера для ручного и автоматического режимов в зависимости от потребностей питомца.



Ты закончил! Поздравляем! Теперь твой питомец точно не останется голодным, если, конечно, ты не забудешь заполнить ёмкость для корма.

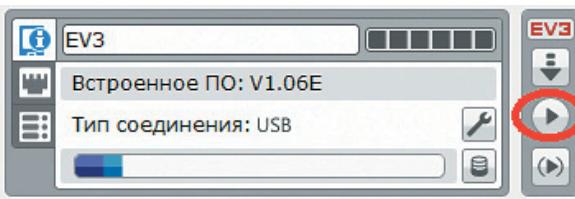
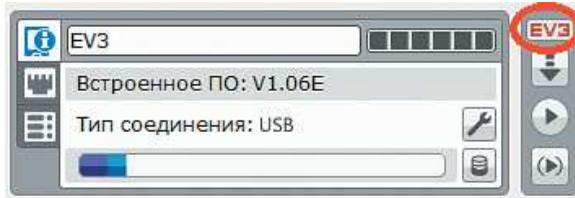


Этап 5. Загрузка программы и её тестирование

ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру, соединив порты PC на модуле и USB на компьютере. В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. Перед запуском программы обязательно убедись, что закрывающий механизм кормушки находится в положении «закрыто», а банк коры, наоборот, открыт. Корм должен без препятствий проходить через горлышко бутылки, но не должен просачиваться наружу при закрытом механизме. Подставь миску.
2. Запусти программу. Кормушка начнёт работать в ручном режиме подачи корма. Изобрази голодного котёнка (мяукать не обязательно) и нажми «лапкой» на кнопку. Запирающий механизм должен опуститься, выдать дозированную порцию и подняться обратно. Убедись, что корм попадает именно в подставленную миску, а не мимо.
3. Попробуй нажать на кнопку снова, пока не истекла минута. Кормушка не должна никак реагировать. В противном случае — где-то случился сбой.

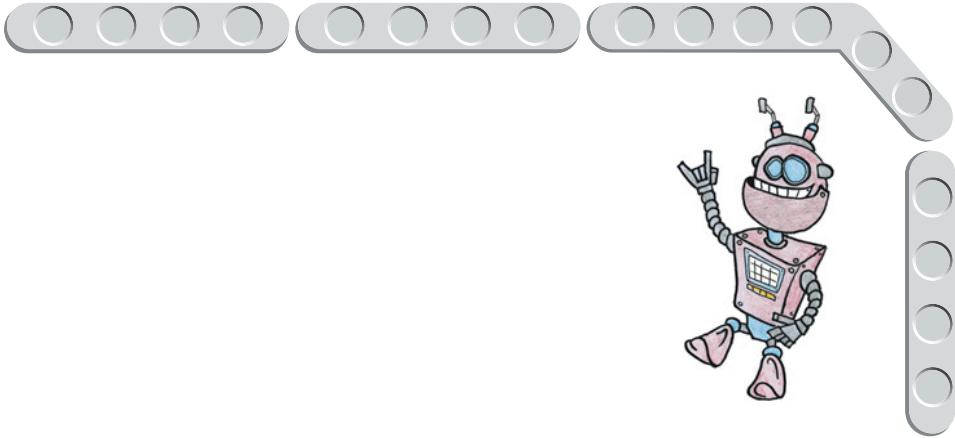
Через минуту попробуй снова нажать на кнопку — кормушка выдаст новую порцию.

Подожди истечения срока действия режима ожидания. Нажми на кнопку вверх на программируемом блоке. Должен прозвучать сигнал и высыпаться первая порция корма. Включится автоматический режим.

Подожди минуту и убедись, что через положенное время кормушка автоматически выдаст новую порцию корма.

Если что-то пошло не так, не расстраивайся! Это может быть ошибка компиляции или банальная невнимательность при подготовке кода и механизма. Проверь всё ещё раз и загрузи программу на программируемый блок. Не забудь снова её протестировать!





А теперь...



У тебя есть автоматизированная кормушка! Твой кот счастлив, котёнок не дерёт твою ногу когтями в тщетных попытках выпросить лакомство, твои носки не обслонялены ластящимся щенком! Прекрасное мгновение. И казалось бы, всё идеально в твоей конструкции, но... всегда найдутся «но», не правда ли? Что же делать, если ты вдруг забудешь поменять корм, добавить новый, ведь так приятно думать, что всё работает само? У нас есть решение!

В любой непонятной робототехнической ситуации — просто добавь датчик. На раме достаточно места, чтобы закрепить ещё один датчик. На этот раз тебе понадобится модуль распознавания цвета и света. Установи его так, чтобы «глазок» (но ты же знаешь, что это фотодиод!) смотрел сквозь бутылку.

Для начала убери этикетки и убедись, что пластик достаточно прозрачен, чтобы в нём было чётко видно границу корма. Закрепи датчик на уровне где-то на $\frac{1}{4}$ над основанием бутылки. Таким образом, когда уровень корма выше минимальной отметки, свет не будет проходить через бутылку, следовательно, датчик получит одно показание. Если же корма не будет, света через бутылку пройдёт больше. Элементарно, да?

Да-да, ты прав, это сработает только в условиях примерно одинаковой освещённости. Но ты же не оставляешь на вечер своего бедного котика одного в пустой квартире с выключенным светом? Надеемся, нет — ему и так одиноко без тебя!

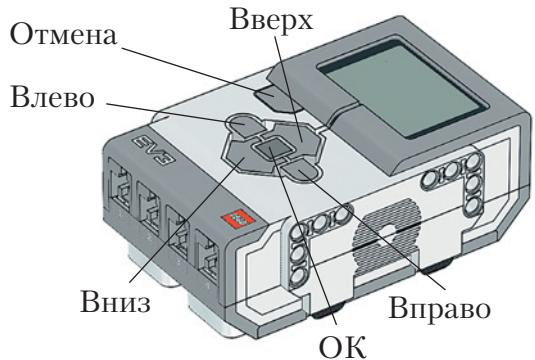
При достаточном освещении (слабая лампа и дневной свет из окна или его отсутствие) погрешности всё же измеримы. Составь таблицу. Помести кормушку и датчик и фиксируй показания. Выведи средние значения и самое тёмное с кормом/светлое без корма — они пригодятся тебе при настройке условий.

Время		С кормом	Без корма
1			
2			
3			
4			
Среднее значение			
Максимальное/минимальное значение			

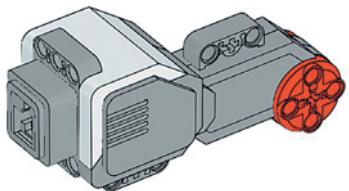
Возможности твои практически неограничены! Ты можешь создать кормушку, учитывающую расписание, как настоящие часы, однако для этого придётся постараться. Вперёд!

Перечень деталей

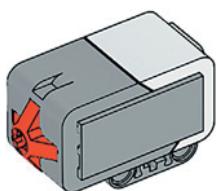
Модуль EV3, 1x



Большой мотор, 1x



Датчик касания, 1x

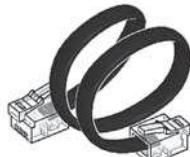


Кабель

Кабель, 35 см, 1x

Кабель, 25 см, 1x

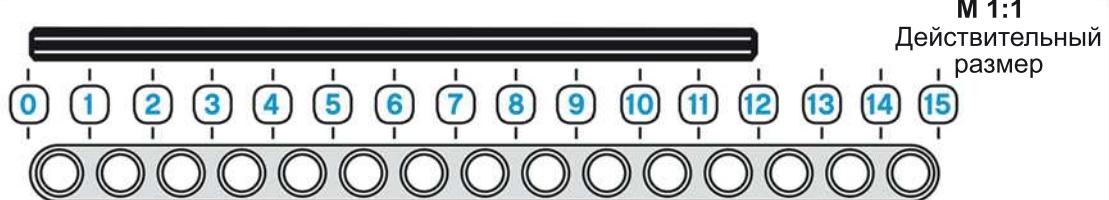
Кабель, 50 см, 1x



Левая криволинейная панель 5×11, чёрная, 1x



Правая криволинейная панель 5×11, чёрная, 1x



Зубчатые колёса

Зубчатое колесо,
40 зубьев, серое, 2x



Балки прямые

Балка с поперечным
отверстием, 2-модульная,
чёрная, 4x

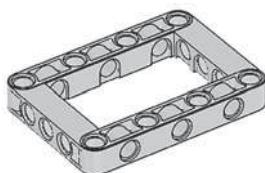


Балка № 3, жёлтая, 1x

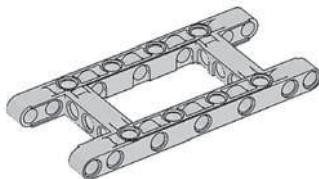


Рамы

Рамка 5×7, серая, 3x



Рамка 5×11, серая, 1x



Втулки

Втулка, $\frac{1}{2}$ -модульная,
жёлтая, 2x



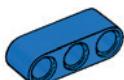
Втулка, 1-модульная,
серая, 2x



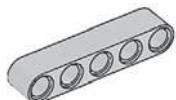
Балка № 3, красная, 2x



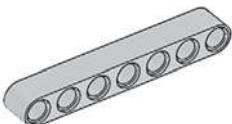
Балка № 3, синяя, 1x



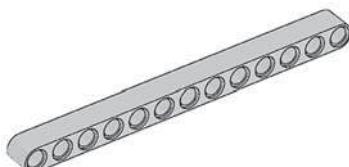
Балка № 5, серая, 2x



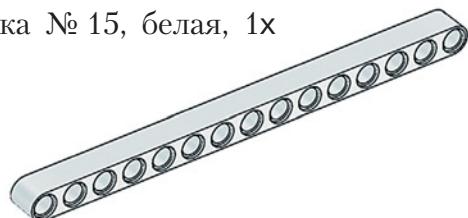
Балка № 7, серая, 4x



Балка № 13, серая, 4x



Балка № 15, белая, 1x

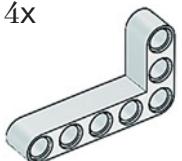


Балки угловые

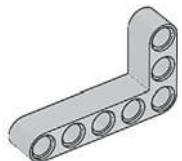
Прямоугольная балка,
 2×4 -модульная,
красная, 2x



Прямоугольная балка,
 3×5 -модульная, белая, 4x



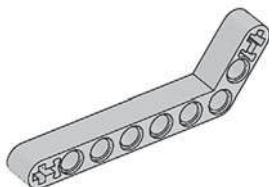
Угловая балка, 3×5 -модульная,
серая, 1x



Двойная угловая балка,
 3×7 -модульная, белая, 4x



Угловая балка, 3×7 -модульная,
серая, 2x



Балки поперечные

Поперечная балка 2×1 ,
красная, 1x



Штифты

Соединительный штифт,
2-модульный, чёрный, 45x



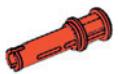
Соединительный штифт,
2-модульный, синий, 20x



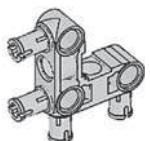
Соединительный штифт,
3-модульный, синий, 11x



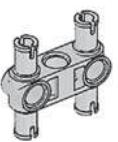
Соединительный штифт
со втулкой, 3-модульный,
красный, 14x



Угловой соединительный
штифт, 3×3-модульный,
серый, 2x



Двойной соединительный
штифт, 3×3-модульный,
серый, 1x



Оси

Ось, 3-модульная,
серая, 13x



Ось с шипом, 3-модульная,
тёмно-бежевая, 13x



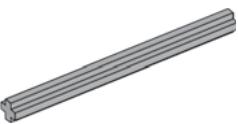
Ось, 4-модульная,
чёрная, 3x



Ось, 5-модульная,
серая, 1x

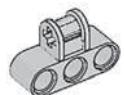


Ось, 7-модульная,
серая, 1x



Блоки

Поперечный блок,
3 × 2-модульный,
серый, 2x



Угловой блок, 1, 0°,
чёрный, 2x

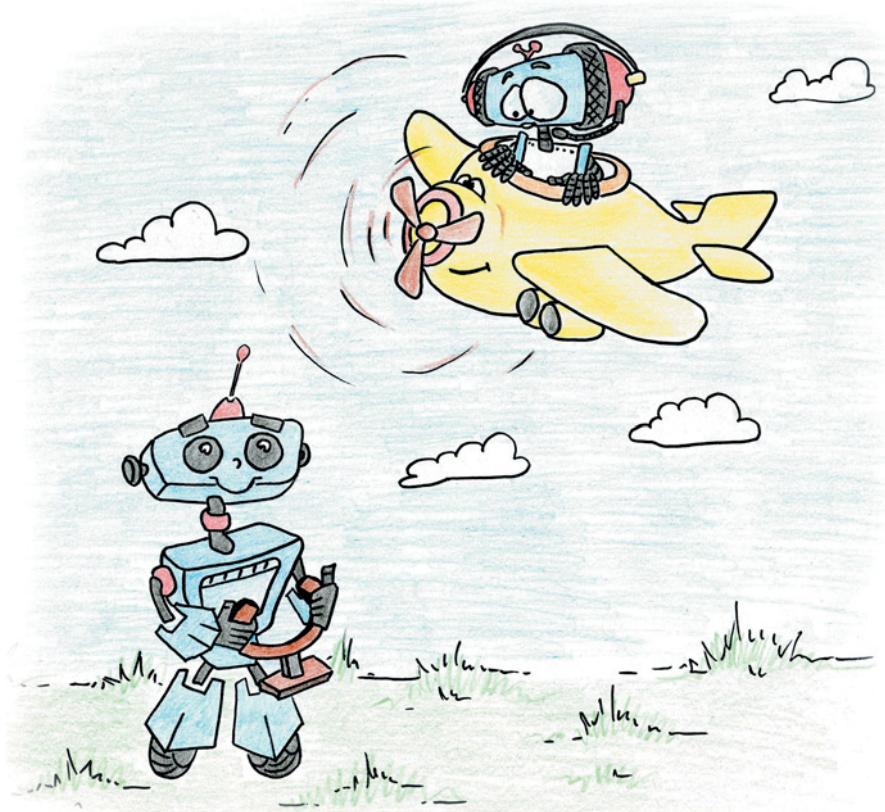


Поперечный блок,
2-модульный, серый, 2x



education

Крутое пике



Знакомимся с самолётом



Человек всегда хотел летать, как птица. Мечта сбылась — мы с лёгкостью преодолеваем огромные расстояния на самолёте. Может быть, ты мечтаешь управлять этой красивой серебристой птицей — стать пилотом?

Самолёт — аппарат для полётов в воздухе с помощью двигателей и неподвижных крыльев.

Главные части самолёта:

- *крыло* — помогает самолёту оторваться от земли;
- *хвостовое оперение* — поверхности, предназначенные для обеспечения устойчивости и управляемости;
- *фюзеляж* — «тело» самолета, на нём крепится всё остальное: крыло, оперение, шасси, двигатели; кроме того, в фюзеляже находится кабина пилотов, салон для пассажиров, грузы и оборудование;
- *шасси* — это система опор, необходимых для разбега самолёта при взлёте, пробега при посадке, а также передвижения и стоянки на земле; чаще всего шасси бывают колёсные;
- *силовая установка* — это двигатели;
- *системы бортового оборудования* — это различные приборы и устройства, которые позволяют выполнять полёты при любых погодных условиях. Бортовая электроника является наиболее «умным», сложным и дорогостоящим оборудованием и превосходит по стоимости все остальные части самолёта.

По своему назначению самолёты бывают:

- военные (истребители, перехватчики, бомбардировщики, ракетоносцы, авианосцы, разведчики и пр.);
- гражданские (пассажирские, транспортные, почтовые, сельскохозяйственные, санитарные, спортивные, учебно-тренировочные).

Также самолёты различаются по массе, типу и числу двигателей, по числу и расположению крыльев, по расположению оперения, по размерам и этажности фюзеляжа, по скорости полёта, по типу взлёта и посадки и пр.

Историю создания летательных аппаратов и воздухоплавания можно прочесть, перейдя по ссылкам:

- http://fern-flower.ru/articles/istoriya_vozniknoveniya_vozduhoplavaniya_aviacii_i_pokoreniya_kosmosa

- <http://www.phisiki.com/2012-02-28-10-51-54/95-istoriya-isobreteniya-samoleta>
- <http://www.thingshistory.com/kratkaya-istoriya-dvigatelya-i-samoleta/>

НАВИГАЦИОННЫЕ ПРИБОРЫ

В кабине пилотов на панели управления размещены различные рычаги, штурвалы и приборы для контроля полёта и оборудования самолёта (рис. 1). Это бортовые измерительные приборы. К приборам для контроля параметров полёта относится и искусственный горизонт — авиагоризонт.



Рис. 1. Кабина пилотов Airbus-A320

Авиагоризонт — это навигационный прибор для измерения и сообщения экипажу углов крена и тангажа, соответствующих положению самолёта относительно горизонтальной плоскости (то есть земли).

Крен — поворот самолёта вокруг его продольной оси (рис. 2 и 3).

Тангаж — угловое движение самолёта относительно его поперечной



Рис. 2. Оси самолёта

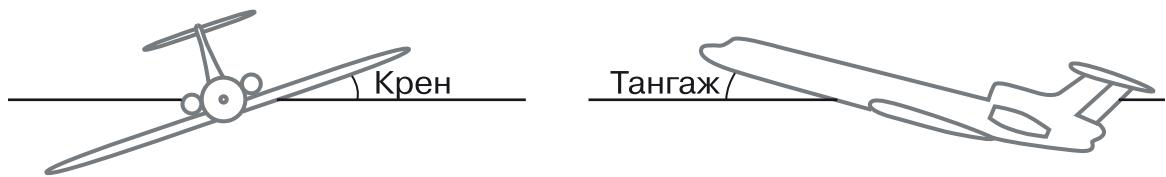


Рис. 3. Крен и тангаж самолёта. Взято с сайта по адресу <http://myleksii.ru/1-34279.html>

оси (см. рис. 2 и 3): при взлёте носовая часть поднимается вверх, хвостовая опускается вниз, при посадке наоборот.

Авиагоризонт (рис. 4) является *гироскопическим прибором*, или *гироскопом*. Он может измерять углы наклона тела, на котором оно установлено (в нашем случае — на самолёте).

Авиагоризонт необходим лётчику для контроля положения самолёта в пространстве: для обеспечения безопасности во время полёта, при выведении самолёта из сложного положения, при выполнении фигур пилотажа.

Гироскоп, размещённый внутри авиагоризонта, «отслеживает» изменения положения самолёта относительно его продольной и поперечной осей и отображает эти изменения на экране авиагоризонта в виде схематически представленных неба (синяя полусфера вверху), земли (тёмная полусфера внизу) и самолёта — U-образная конструкция в центре, с жёлтыми горизонтальными «крыльышками» (см. рис. 4).

Граница между небом и землёй на приборе есть линия *искусственного горизонта*. Если в обычном полёте (при хорошей видимости) обратить внимание на взаимное расположение *естественного горизонта* (впереди за лобовым стеклом) и *искусственного* (на приборе), то окажется, что они всегда параллельны.

Если «приподнять» нос самолёта подачей штурвала на себя, то искусственный горизонт, так же как и естественный, переместится вниз. Опускаем «нос» — и линии обоих горизонтов синхронно поползут вверх.

Вооружённые таким прибором пилоты всегда могут выбрать безопасное положение самолёта в пространстве, ориентируясь на показания авиагоризонта.

Авиагоризонты оснащены все современные самолёты как гражданской, так и военной авиации.

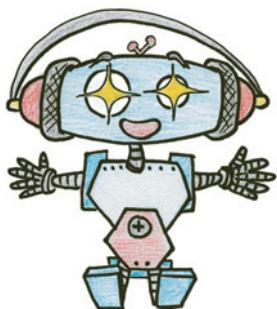


Рис. 4. Авиагоризонт

В нашем проекте мы будем собирать и управлять моделью штурвала пассажирского самолёта Airbus-A320. При управлении самолётом с целью обеспечения безопасности необходимо учитывать следующие параметры:

- скорость полета: 800–840 км/ч, или 222–233 м/с;
- задаваемые значения углов: тангажа — от –20 градусов до 22 градусов, крена — от –15 градусов до 15 градусов.

Этап 1. Устройство авиасимулятора



Рассмотри модель робота-авиасимулятора, собранную на основе набора LEGO® MINDSTORMS® Education EV3 (рис. 5).

Выдели на ней рабочие детали: модель штурвала; два мотора, отвечающие за углы крена и тангла; два датчика касания — газ и тормоз.

Обрати внимание, в конце проекта в таблице даны все детали, которые потребуются тебе для сборки. Эта таблица поможет быстро найти то, что необходимо, и не ошибиться при конструировании. А теперь давай соберём эту модель!

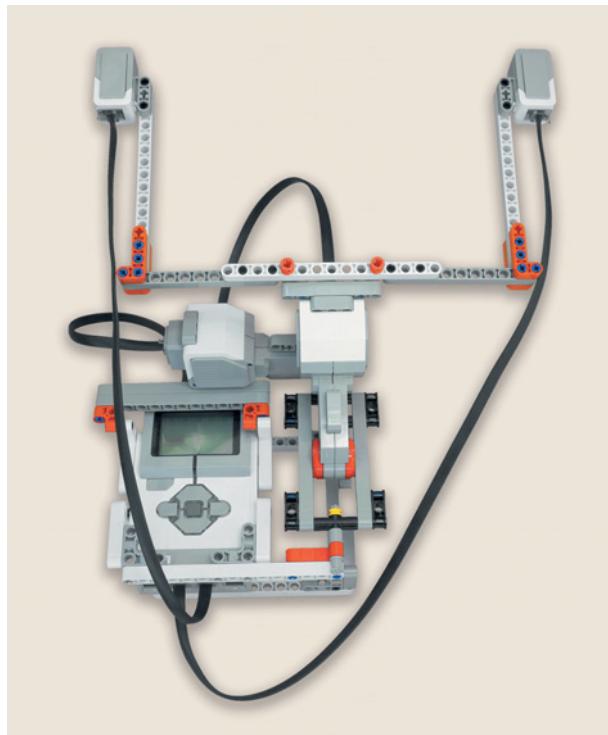
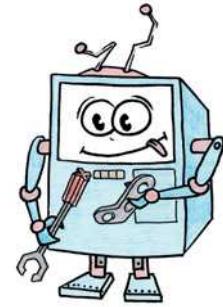


Рис. 5. Робот-авиасимулятор

Этап 2. Сборка авиасимулятора



ШАГ 1. СБОРКА КОНТУРА ЖЁСТКОСТИ

Детали для сборки:

- программируемый модуль EV3, 1x;
- прямоугольная балка 3×5 , белая, 4x;
- угловой соединительный штифт, 3 × 3-модульный, серый, 2x;
- соединительный штифт, 2-модульный, чёрный, 6x.



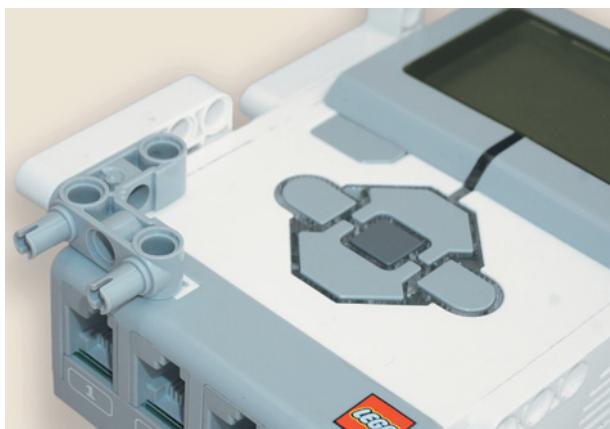
1. Вставь на левой боковой панели программируемого модуля чёрные штифты в первый, третий и четвёртый верхние модули.



2. Закрепи на двух левых штифтах прямоугольную балку, установи на правом штифте вторуюую прямоугольную балку, как показано на рисунке.



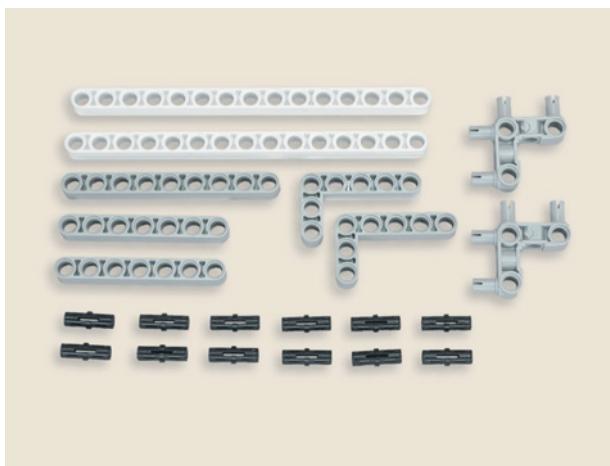
- Установи с обратной стороны этой балки угловой штифт в её третьем и пятом модулях.



- Перейди на правую боковую панель EV3 и зеркально собери то же самое.



ШАГ 2. СБОРКА ОСНОВАНИЯ РУЛЕВОГО МЕХАНИЗМА

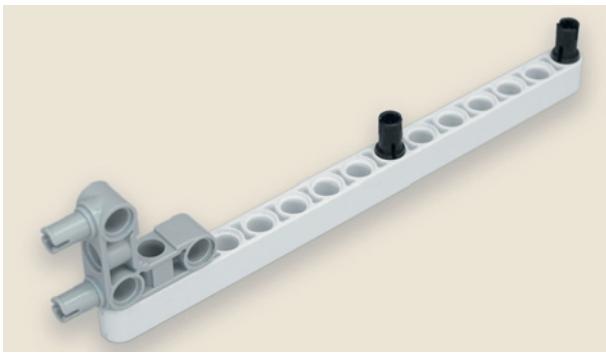


Детали для сборки:

- балка № 15, белая, 2x;
- балка № 9, серая, 1x;
- балка № 7, серая, 2x;
- прямоугольная балка 3 × 5, серая, 2x;
- угловой соединительный штифт, 3 × 3-модульный, 2x;
- соединительный штифт, 2-модульный, чёрный, 12x.

Основание состоит из двух одинаковых половинок. Сначала собери левую.

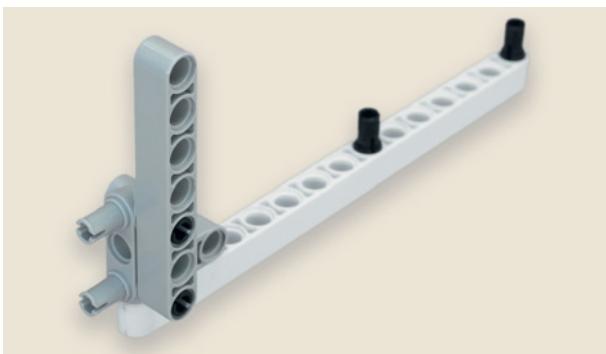
1. Вставь в девятый и пятнадцатый модули балки № 15 чёрные штифты. Установи во втором и четвёртом модулях угловой штифт, как показано на рисунке.



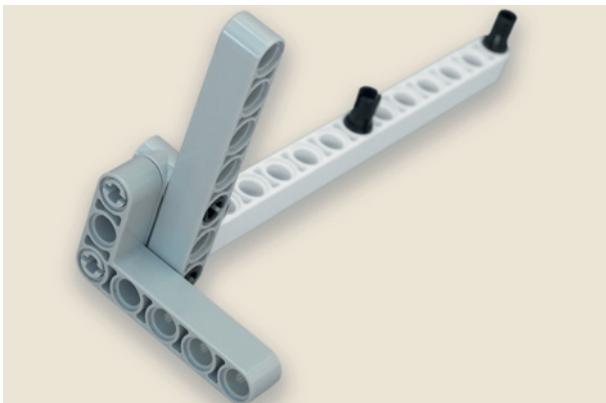
2. Вставь в его вертикальные модули чёрные штифты.



3. Закрепи на этих штифтах балку № 7.



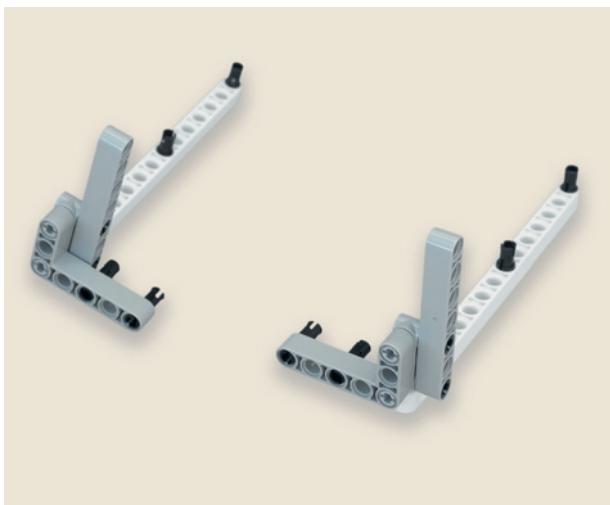
4. Установи на угловом штифте с торца прямогоугольную балку.



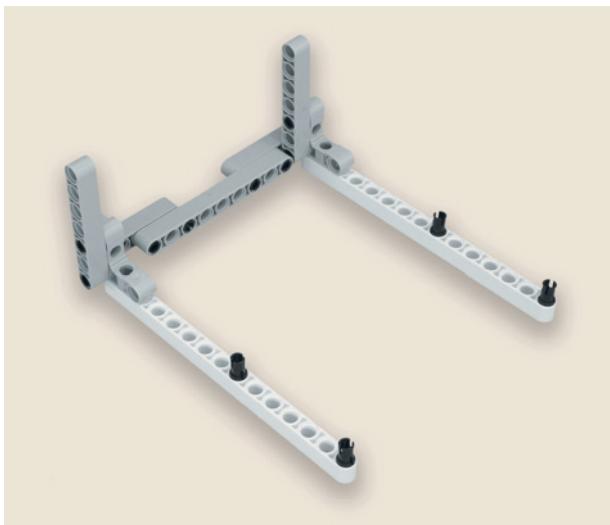
5. Вставь в первый и третий модули балки с внутренней стороны чёрные штифты.



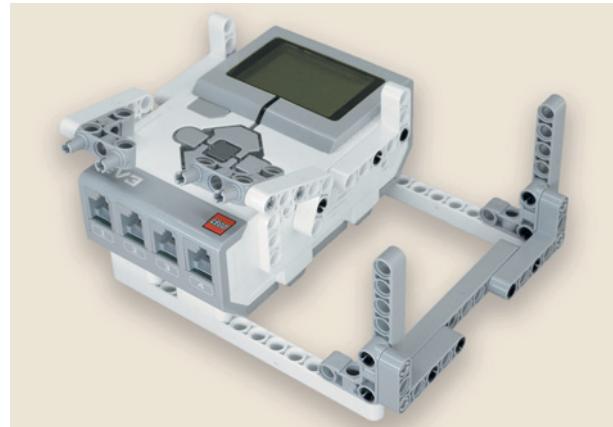
6. Собери зеркально правую половинку основания рулевого механизма, но балку № 7 установи с внешней стороны.



7. Соедини получившиеся половинки балкой № 9, как показано на рисунке.



8. Установи на четырёх свободных чёрных штифтах программируемый модуль.



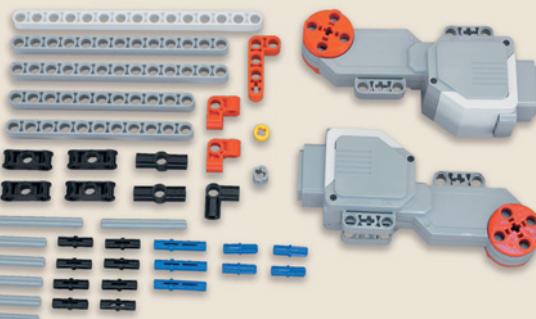
ШАГ 3. СБОРКА РУЛЕВОГО МЕХАНИЗМА

Детали для сборки:

- балка № 15, белая, 1x;
- балка № 13, серая, 2x;
- балка № 11, серая, 2x;
- прямоугольная балка 2 × 4, красная, 1x;
- поперечная балка 2 × 1, красная, 2x;
- втулка, $\frac{1}{2}$ -модульная, жёлтая, 1x;
- втулка, 1-модульная, серая, 1x;
- ось № 5, серая, 2x;
- ось № 3, серая, 5x;
- угловой блок 2, 180°, чёрный, 2x;
- угловой блок 6, 90°, чёрный, 1x;
- двойной поперечный блок, 3-модульный, чёрный, 4x;
- соединительный штифт, 3-модульный, синий, 3x;

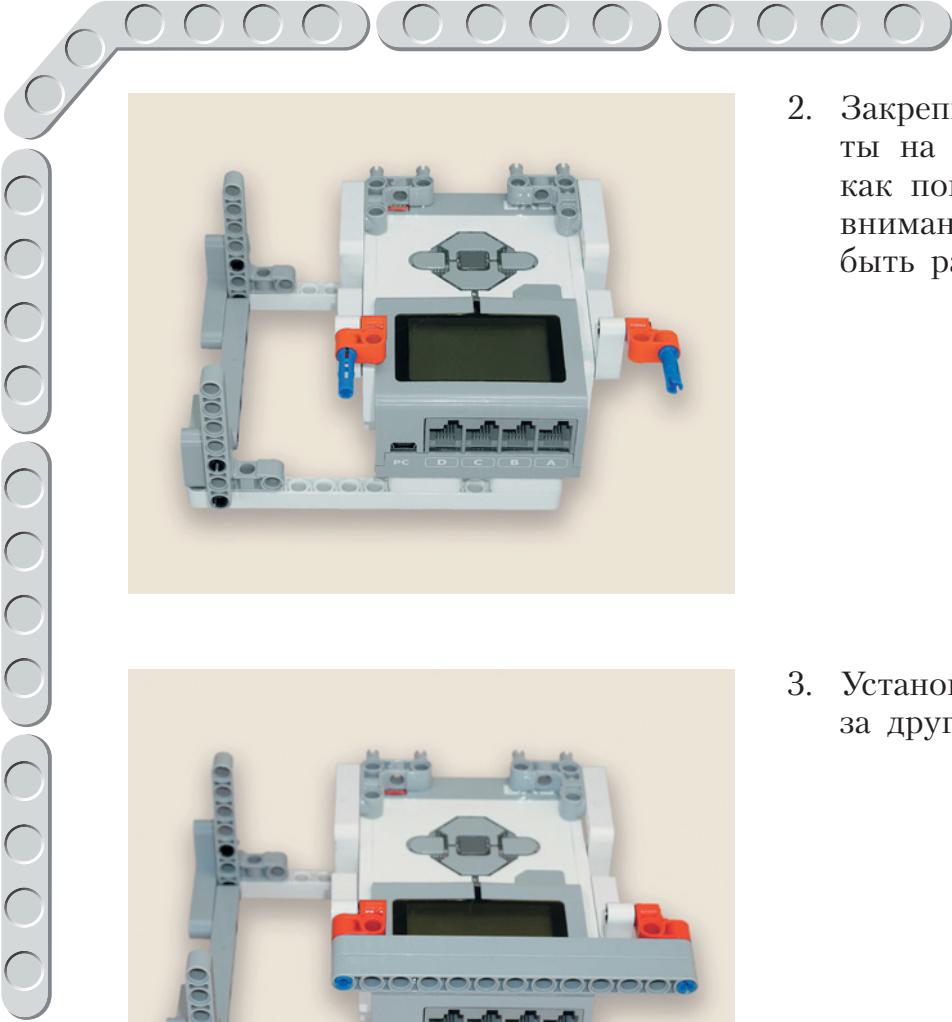


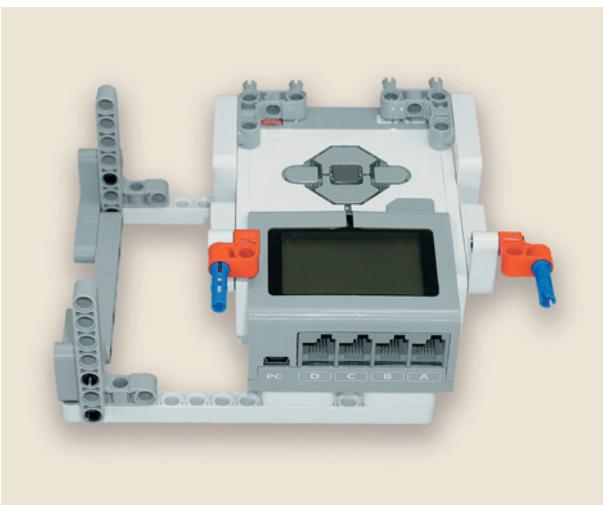
- соединительный штифт, 2-модульный, синий, 4x;
- соединительный штифт, 2-модульный, чёрный, 8x;
- большой мотор, 2x.



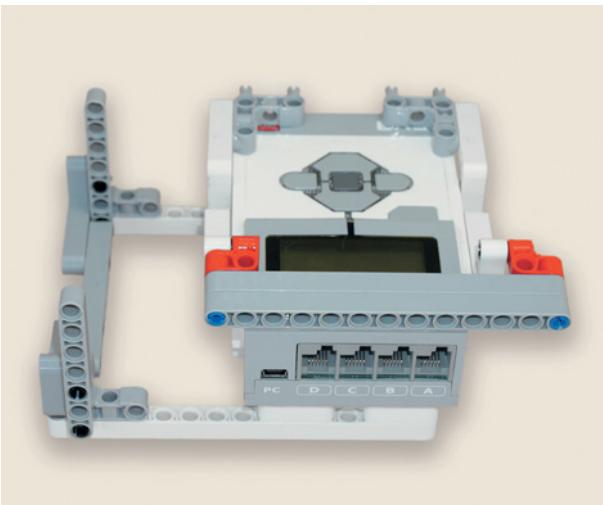
1. Вставь в красные поперечные балки 2 × 1 по одному чёрному 2-модульному и по одному синему 3-модульному штифту.



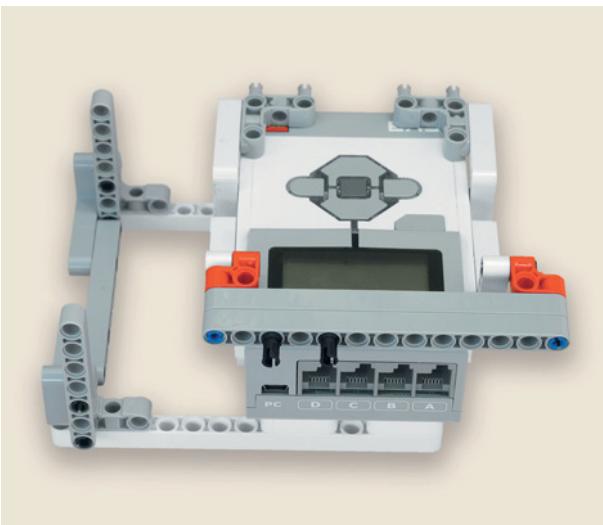
- 68
- 
2. Закрепи получившиеся элементы на балках контура жёсткости, как показано на рисунке. Обрати внимание на то, как они должны быть расположены!



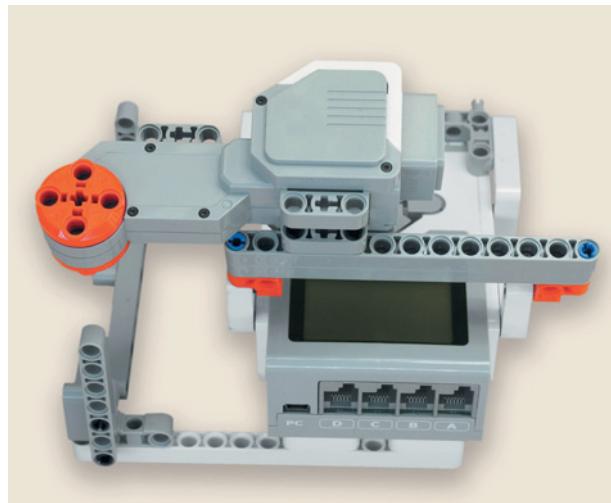
3. Установи на синих штифтах друг за другом две балки № 13.



4. Вставь в третий и пятый модули крайней балки № 13 чёрные штифты.



5. Установи на этих штифтах большой мотор. Этот мотор управляет углом крена самолёта. Мы будем называть его *горизонтальным*.



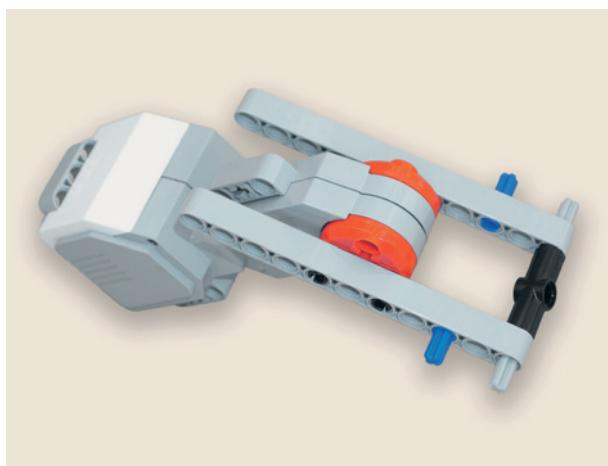
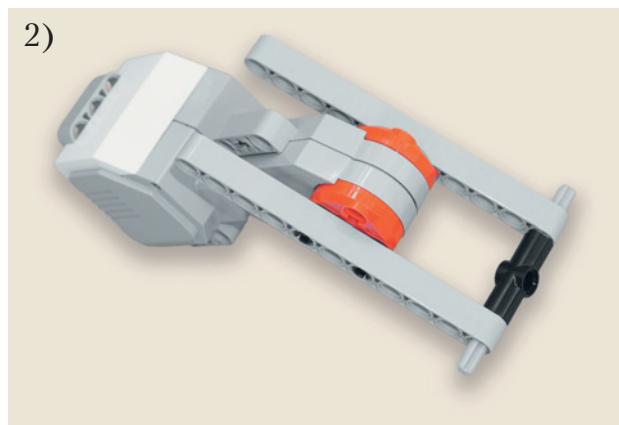
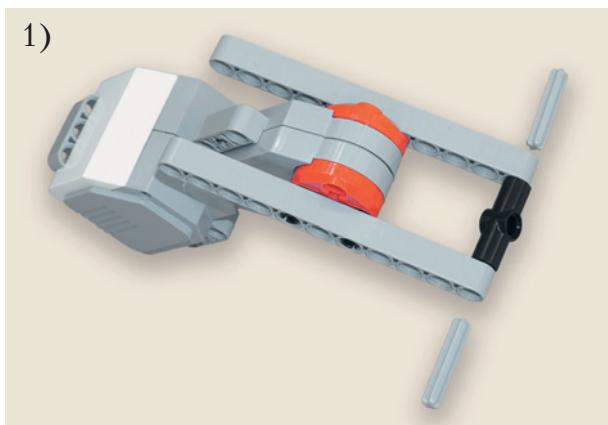
6. Возьми второй мотор. С обеих сторон привода в круглые отверстия вставь симметрично по два чёрных штифта. Этот мотор управляет углом тангажа самолёта. Мы будем называть его *вертикальным*.



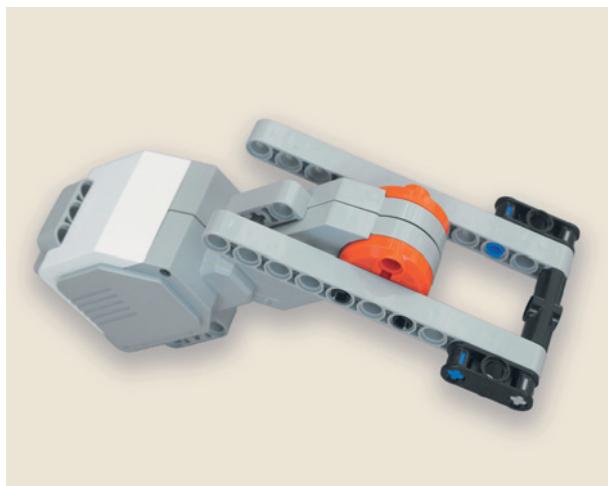
7. Закрепи на них балки № 11 пятым и седьмым модулями.



8. Вложи угловой блок 2 между концами балок так, чтобы его центральный модуль «смотрел» на мотор. Закреши с обеих сторон осями № 3.



9. Вставь в третий модуль балок № 11 по одному синему 2-модульному штифту.



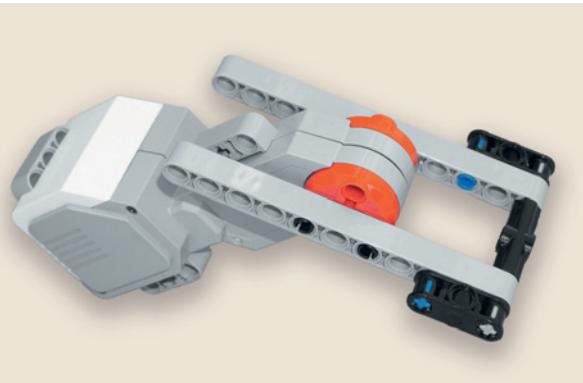
10. Закреши с обеих сторон на штифте и на выступающем конце оси по двойному поперечному блоку.



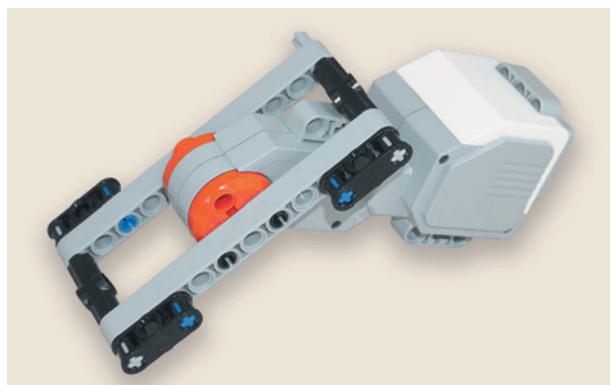
11. Переверни мотор «вверх дном» и прокрути рулевой механизм так, чтобы чёрный угловой блок 2 опустился на стол.

Внимание!

Для продолжения сборки мотор должен находиться в таком положении!



12. Собери на свободных концах балок № 11 такую же конструкцию, как внизу, только с левой стороны не ставь синий штифт и двойной поперечный блок.



13. Вставь в девятый модуль левой балки ось № 3 и придержи её рукой. Надень на внутренний конец оси угловой блок 6, как показано на рисунке.



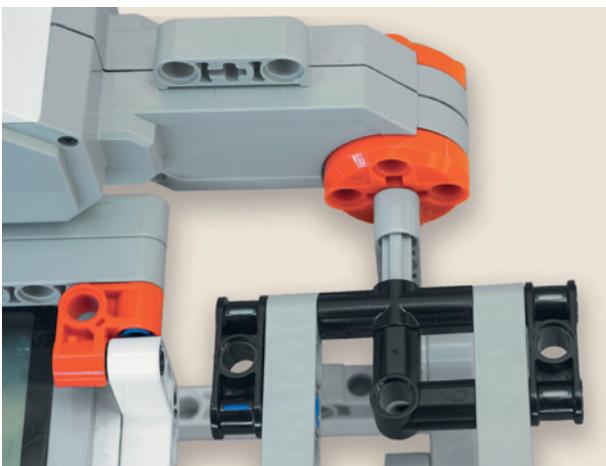
14. Надень на выступающие концы осей двойной поперечный блок.



15. Продень ось № 5 сверху так, чтобы она прошла в обе втулки.



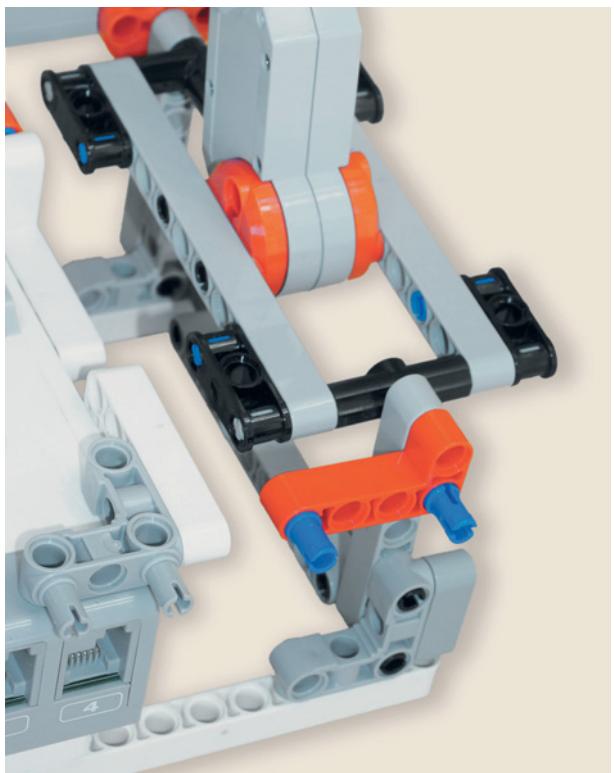
16. Присоедини получившуюся конструкцию к приводу горизонтального мотора. Для этого продень выступающий конец оси № 5 через верхний модуль задней балки основания рулевого механизма так, чтобы он вошёл в центральный крестообразный модуль привода.



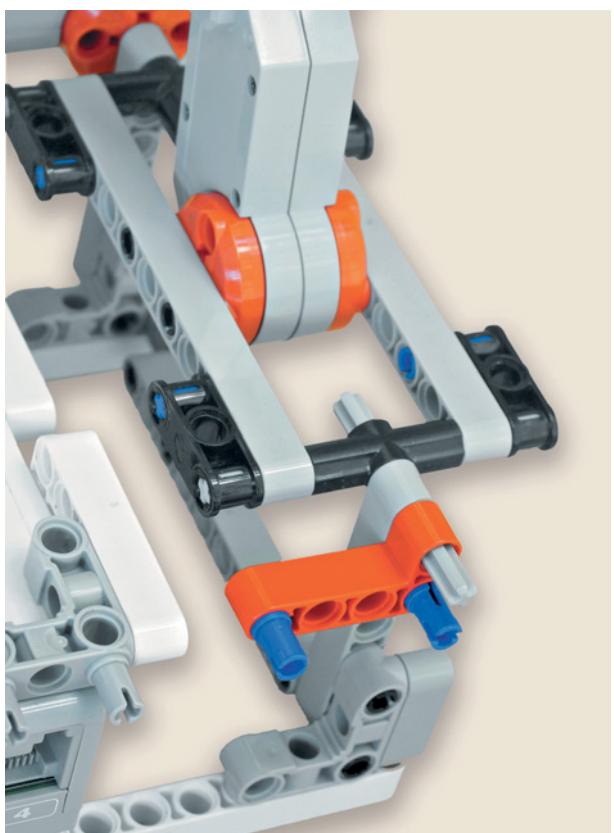
17. Возьми красную прямоугольную балку и вставь в её угловой модуль синий 3-модульный штифт длинным концом вперёд, а в крестообразный модуль — синий 2-модульный.



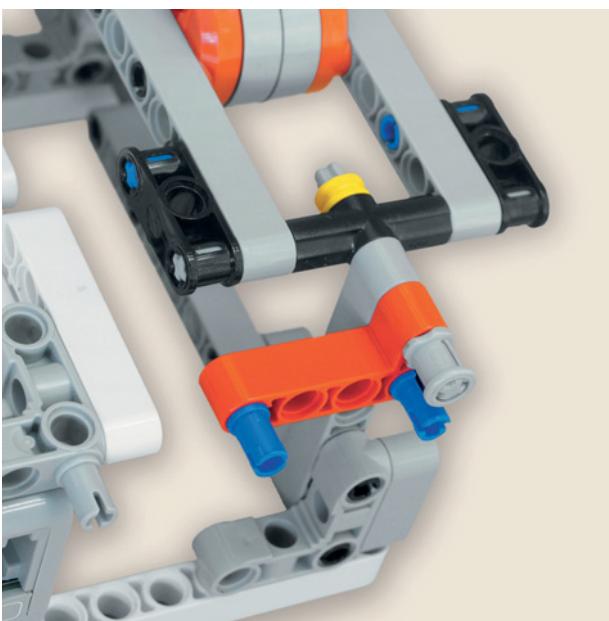
18. Установи получившийся элемент на передней балке основания во втором модуле сверху.



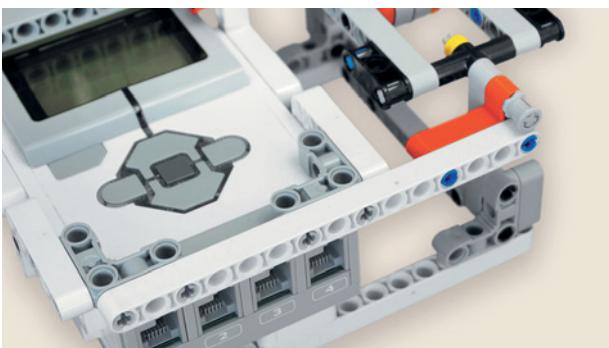
19. Продень ось № 5 через верхний модуль красной балки так, чтобы она прошла в центральный модуль чёрной втулки рулевого механизма.



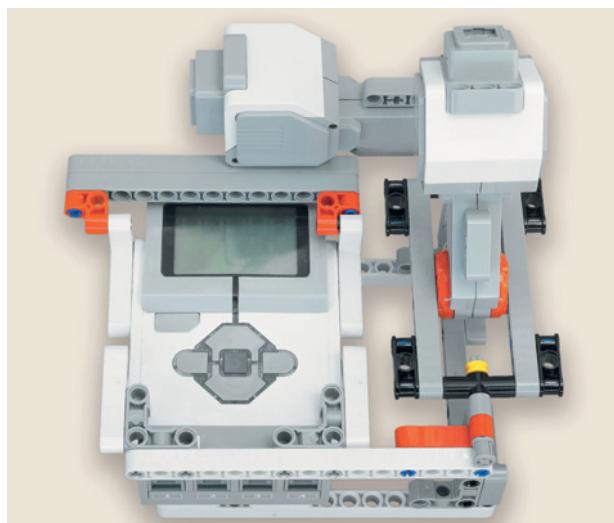
20. Зафиксируй эту ось жёлтой и се-
рой втулками.



21. На свободных штифтах установи
балку № 15.



Рулевой механизм готов!



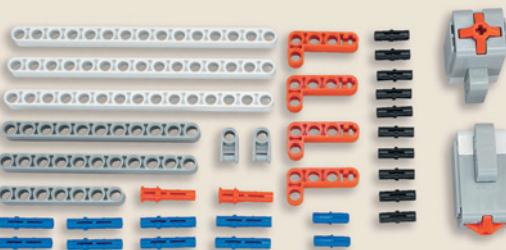
ШАГ 4. СБОРКА ШТУРВАЛА

Детали для сборки:

- балка № 15, белая, 3x;
- балка № 11, серая, 2x;
- балка № 7, серая, 1x;
- прямоугольная балка 2 × 4, красная, 4x;
- соединительный штифт с втулкой, 3-модульный, красный, 2x;
- поперечный блок, 2-модульный, серый, 2x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт, 2-модульный, чёрный, 10x;

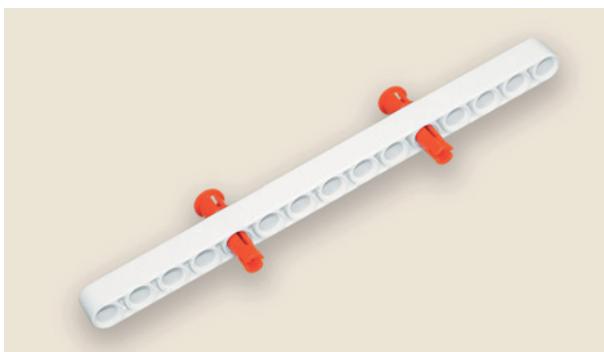


- соединительный штифт, 3-модульный, синий, 8x;
- датчик касания, 2x.

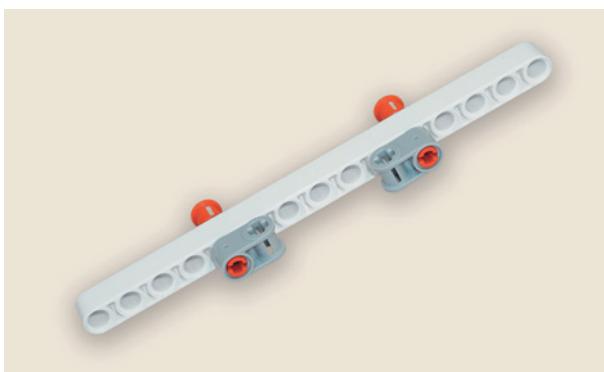


Штурвал состоит из трёх частей: центральной и двух одинаковых боковых. Начни с центральной части.

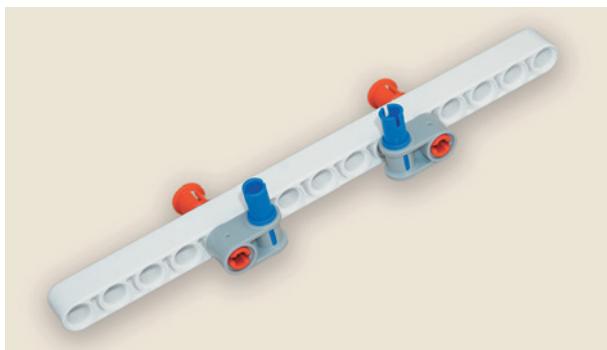
1. Вставь в пятый и одиннадцатый модули балки № 15 с обратной стороны по одному красному штифту с втулкой.



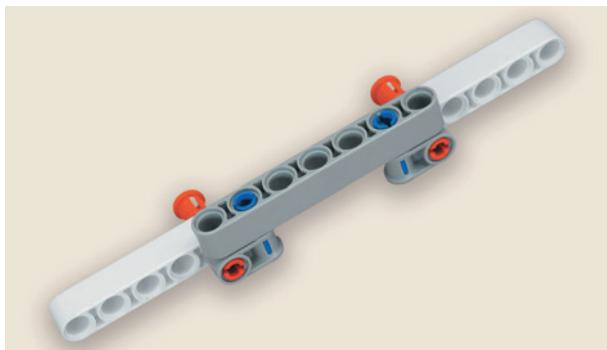
2. Надень на их концы серые поперечные блоки.



3. Вставь в крестообразные модули блоков синие 2-модульные штифты.

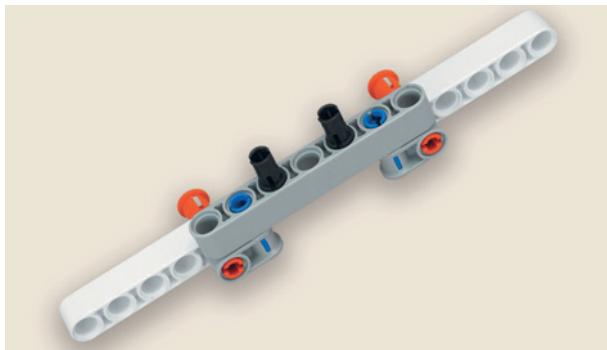


4. Закрепи балку № 7 на этих штифтах вторым и шестым модулями.



5. Вставь в третий и пятый модули этой балки по одному чёрному штифту.

Центральная часть штурвала авиасимулятора готова!

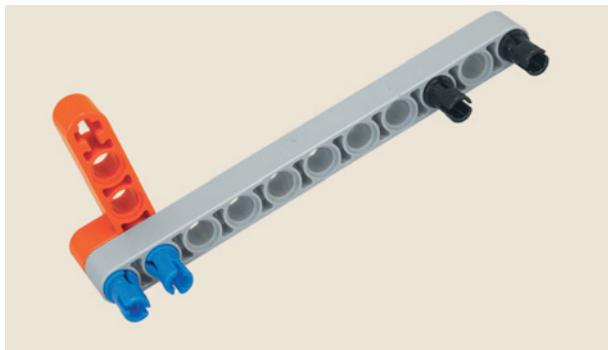


Теперь собери левую часть штурвала.

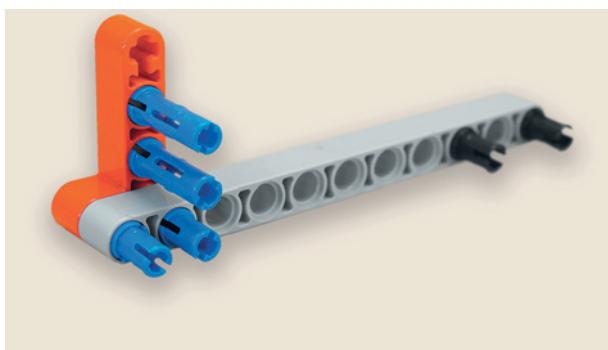
6. Вставь в первый и второй модули балки № 11 синие 3-модульные штифты длинными концами вперёд, а в девятый и одиннадцатый модули — чёрные штифты.



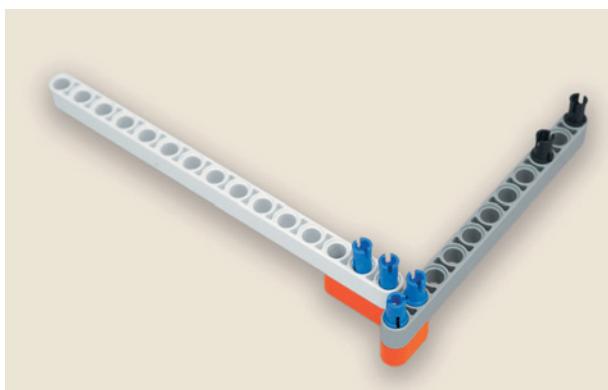
7. Установи на синих штифтах с обратной стороны красную прямогольную балку, как показано на рисунке.



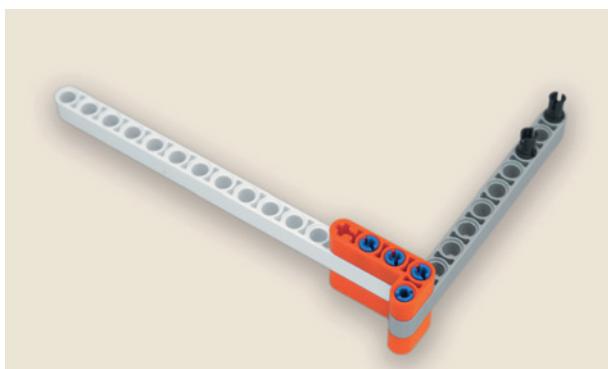
8. Вставь в её свободные модули два синих 3-модульных штифта короткими концами вперёд.



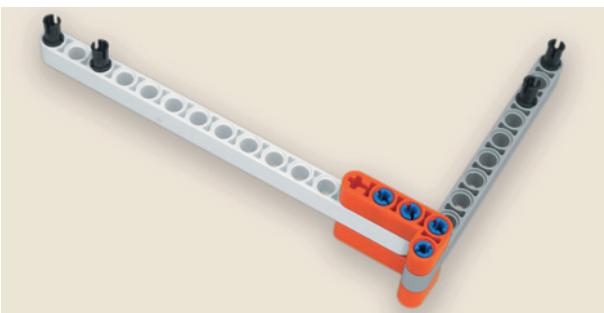
9. Закрепи на этих двух штифтах балку № 15.



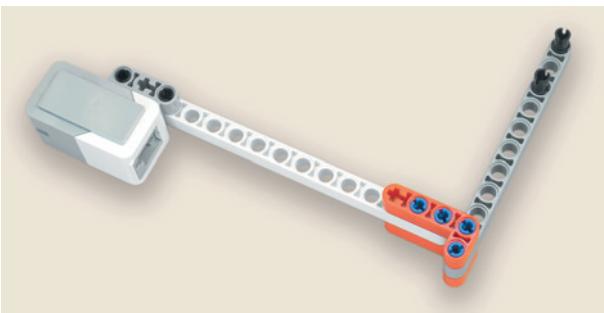
10. Надень вторую красную прямогольную балку на свободные концы синих штифтов.



11. Вставь в первый и третий модули балки № 15 чёрные штифты.

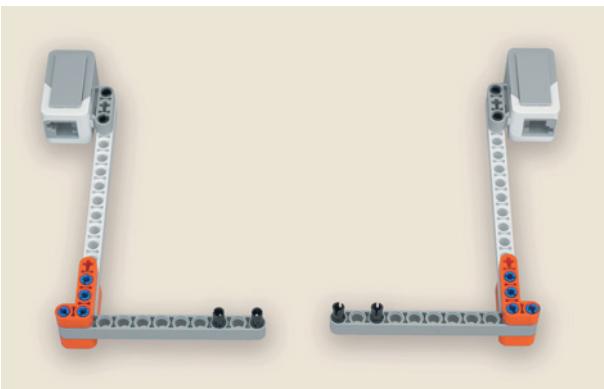


12. Закрепи на этих штифтах датчик касания.



Левая часть штурвала готова!

13. Правую часть штурвала собери зеркально.



14. Получившиеся три части штурвала соедини вместе, как показано на рисунке.

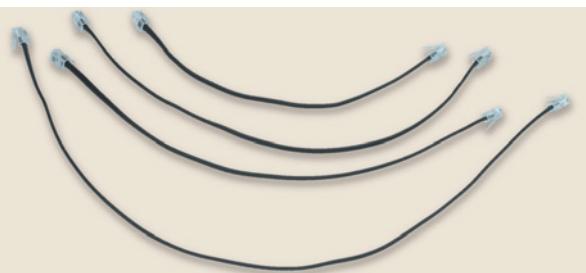


Сборка штурвала завершена!

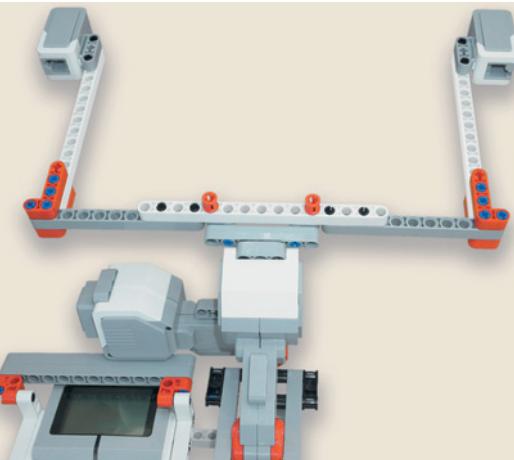
ШАГ 5. СОЕДИНЕНИЕ РУЛЕВОГО МЕХАНИЗМА СО ШТУРВАЛОМ

Детали для сборки:

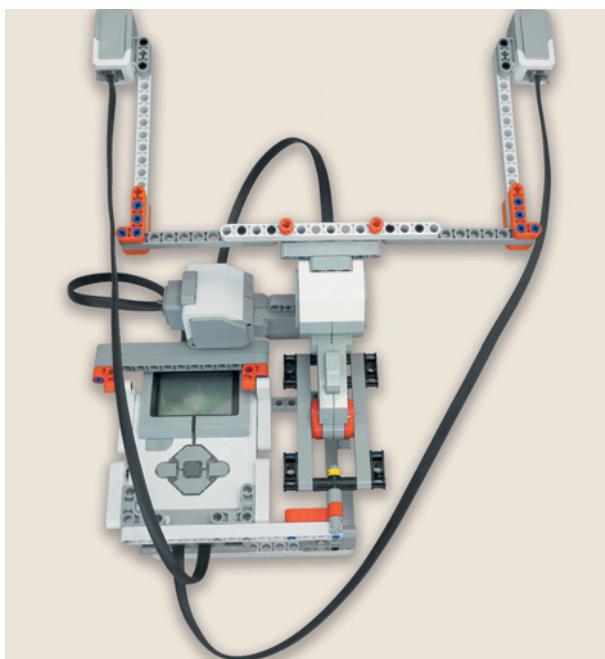
- кабель, 50 см, 1x;
- кабель, 35 см, 2x;
- кабель, 25 см, 1x.



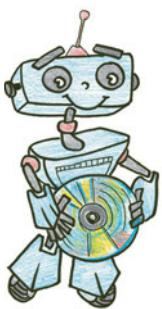
1. Закрепи штурвал на рулевом механизме, вставив выступающие чёрные штифты в модули на вертикальном (см. с. 69, пункты 5 и 6) моторе.



2. Подключи кабели:
 - горизонтальный мотор кабелем длиной 25 см к порту B;
 - вертикальный мотор кабелем длиной 35 см к порту C;
 - левый датчик касания кабелем длиной 35 см к порту 3;
 - правый датчик касания кабелем длиной 50 см к порту 2.



Молодец! Ты справился со сборкой авиасимулятора.



Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.



Этап 4. Создание программы для робота-авиасимулятора

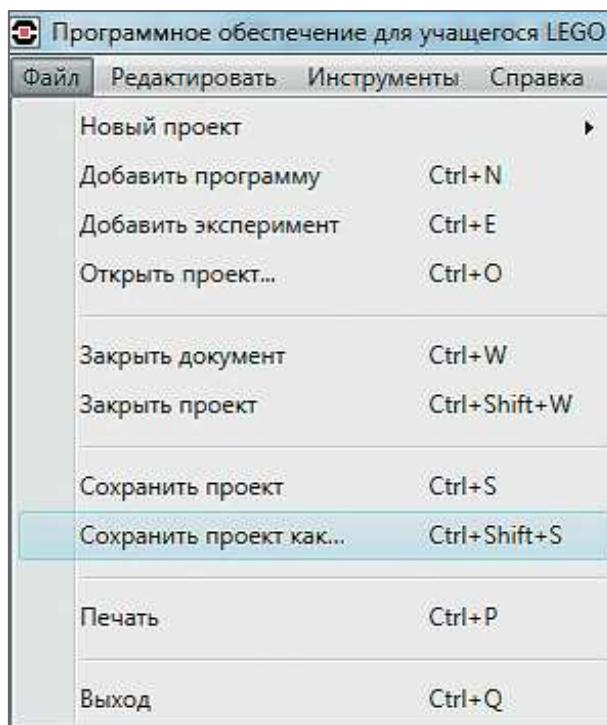
ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, не нужно искать, где он записан.

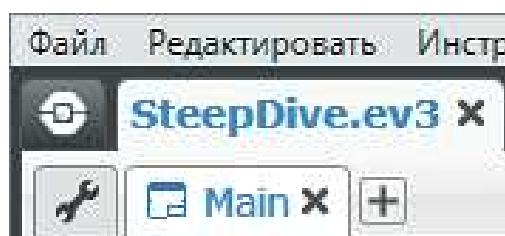
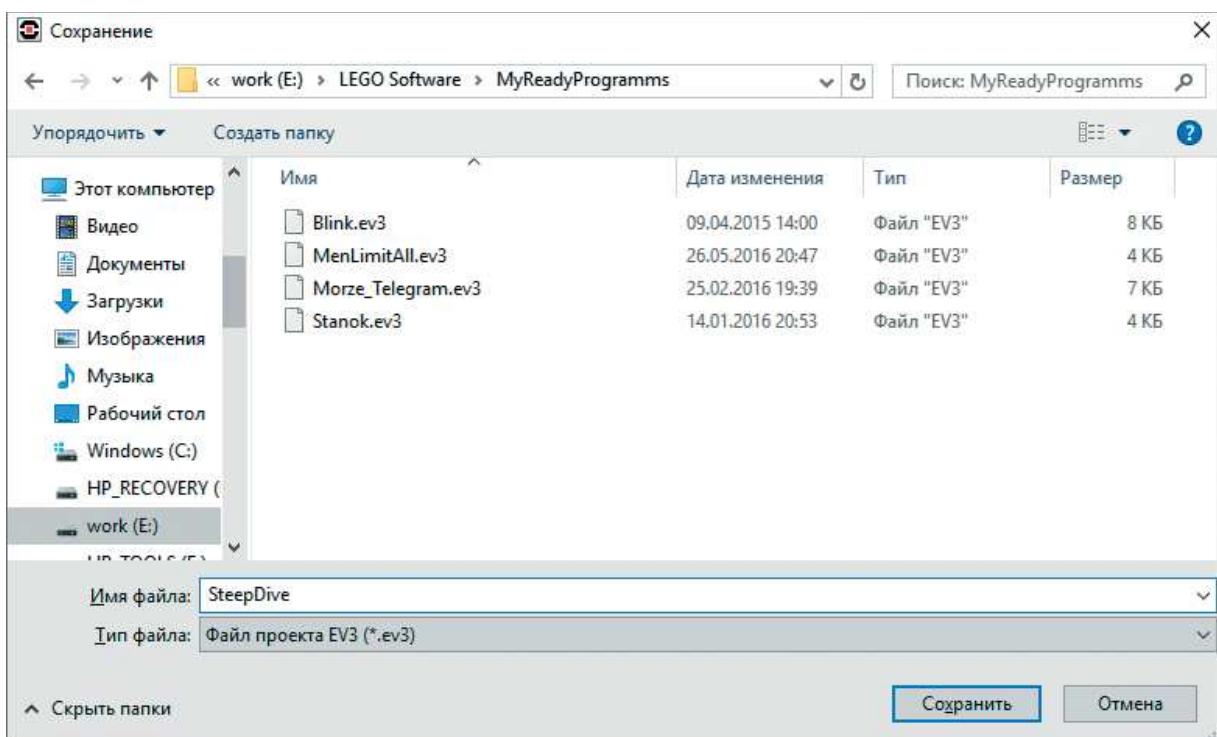
1. Для начала сохрани твой проект в памяти компьютера, чтобы избежать его потери. Для этого выбери меню **Файл** в левом верхнем углу окна программы и в контекстном меню выбери пункт **Сохранить проект как...**



Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они пишут, например, каждое слово с большой буквы (как это сделал и ты) или добавляют нижнее подчёркивание.

2. Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла**: *SteepDive* (в переводе с английского — крутой пике) в нижней части окна и нажми **Сохранить**. Именно так будет называться в памяти программируемого модуля EV3 папка для хранения программ, которые ты напишешь в рамках этого проекта.



3. Осталось дать имя самой программе (в нашем проекте их будет несколько). Для этого дважды кликни по имени *Program* в левом верхнем углу поля программы и задай новое имя: *Main* (в переводе с английского — главный, именно так программисты называют основную программу, если она в проекте не одна) и нажми клавишу **Enter**.

ЛОГИКА ПРОГРАММЫ

Так как мы создаём настоящий авиасимулятор, то программа получится довольно сложной и объёмной. Причём она будет содержать внутри себя другие, совершенно самостоятельные подпрограммы. Поэтому программирование всего устройства мы разобьём на семь частей и опишем логику для каждой из них в отдельности.

Часть 1. Исходные положения. Переменные и начальные параметры.

Часть 2. Двигатели — на старт! Увеличение и уменьшение мощности турбин.

Часть 3. Скорость, крен, тангаж — завернём крутой вираж! Показания спидометра и авиагоризонта.

Часть 4. Тревога! Тревога! Система сигнализации об опасном уровне тангажа.

Часть 5. Бесконечность — не предел! Работа одометра, расчёт проходимого расстояния.

Часть 6. Всё выше, и выше, и выше! Работа альтиметра, расчёт набранной высоты.

Часть 7. Последний рывок. Вывод на экран показаний всех приборов, завершение.

Кстати!

В этой программе будет множество интересных алгоритмических конструкций. Самой первой из них будет подпрограмма.

СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОТА-АВИАСИМУЛЯТОРА

В открывшемся окне проекта начинай составлять программу для робота-авиасимулятора.

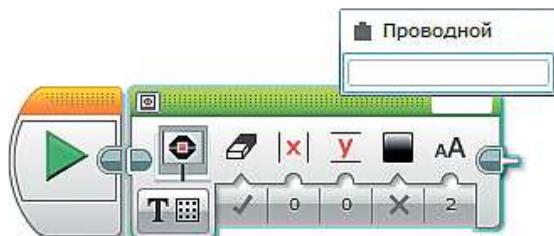
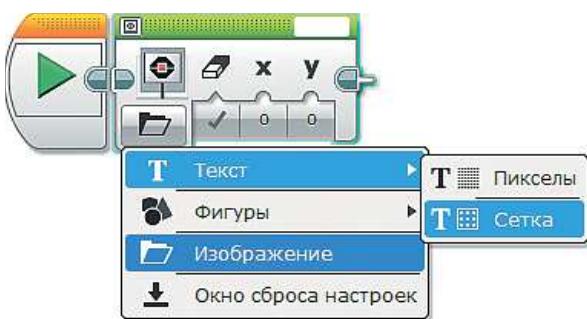
Часть 1. Исходные положения. Переменные и начальные параметры

Логика. В этой части ты инициализируешь в программе несколько переменных, которые понадобятся для ведения подсчётов. Кроме того, ты очистишь экран (ещё его называют «дисплей») программируемого модуля и подготовишь к работе параметр времени.

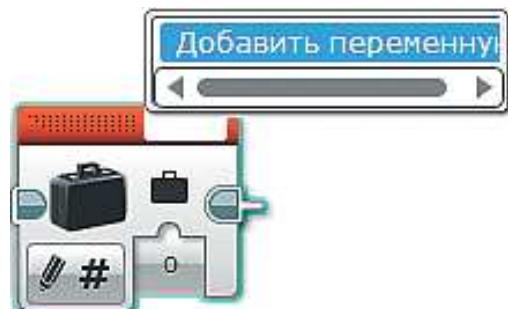
1. Для начала нужно очистить экран от всех надписей. Для этого добавь команду **Экран** (зелёный блок).



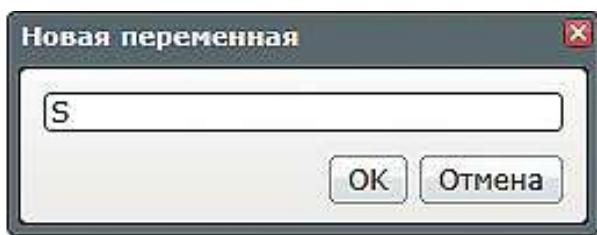
2. Выбери опцию **Текст** → **Сетка**.



3. Кликни мышью в правом верхнем углу, где написано слово MIND-STORMS, сотри эту надпись и нажми клавишу **Enter**. Именно так ты очистишь экран, по сути выведя на него «пустой» текст.



4. Теперь нужно объявить (то есть создать в программе) сразу четыре переменные. Добавь команду **Переменная** (красный блок) и в правом верхнем окошке выбери пункт **Добавить переменную**.



5. Введи в открывшемся окне с клавиатуры название первой переменной: **S** (от латинского *Spatium* — расстояние) — в этой переменной будет рассчитываться и храниться значение пройденного расстояния, то есть показания прибора **Одометр**. Нажми **OK**.

6. Оставь без изменений опцию **Записать числовое значение** и нуль на входе данной команды. Это означает, что ты создал переменную и **инициализировал** её, то есть записал начальное значение, равное нулю.

7. Остальные переменные добавь аналогичным способом, задавая им имена:

- **V** (от латинского *Velocitas* — скорость) — здесь будет рассчитываться и храниться значение текущей скорости самолёта, то есть показания прибора **Спидометр**;
- **T** (от латинского *Tempus* — время) — в этой переменной будет храниться значение времени, которое тратится на обработку одного измерения приборов (в нашем случае мы возьмём одну секунду);
- **H** (от английского *Height* — высота) — здесь будет храниться временное значение угла тангла-жа для расчёта высоты.

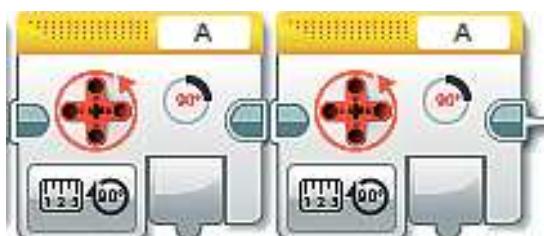
Внимание!

Эта переменная именована строчной (маленькой) буквой.

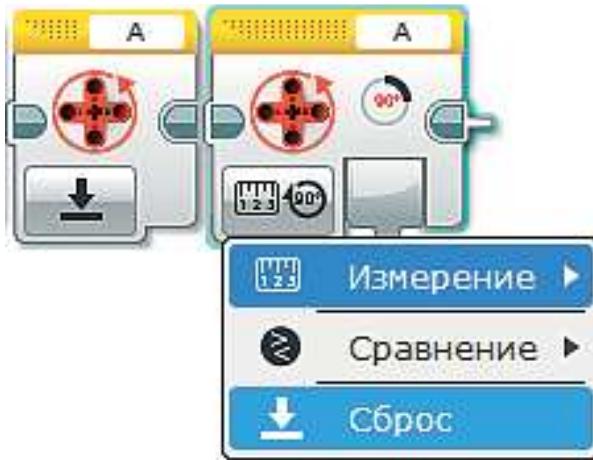
8. В итоге должна получиться такая цепочка команд.



9. Далее тебе нужно сбросить показания моторов, то есть обнулить значения энкодеров¹. Для этого добавь две команды **Вращение мотора** (жёлтый блок).

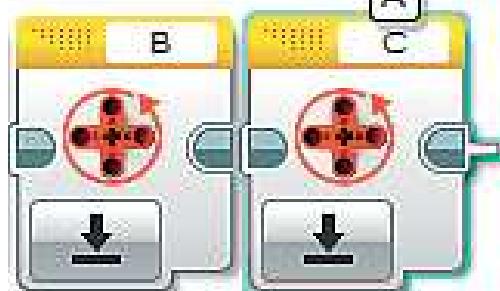


10. Выбери в обеих командах опцию **Сброс**.



¹ Энкодер (от английского *Encoder* — кодировщик) — устройство внутри мотора, позволяющее снимать различные показания: число совершённых оборотов или угол поворота привода.

11. Теперь осталось проверить и при необходимости задать порты моторов. При сборке они были подключены к портам **В** и **С**, поэтому в правых верхних окошках команд **Вращение мотора** должны быть заданы именно эти порты.

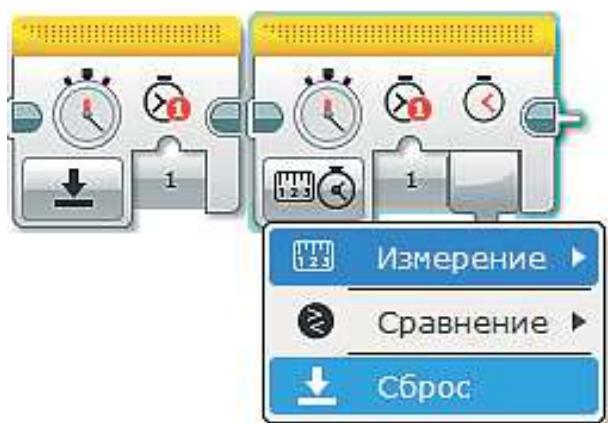


Последними командами перед началом основной программы будут команды таймера. Нам необходимо инициализировать два таймера, каждый из которых будет отвечать за частоту обновления показаний приборов на экране EV3, выполняющего роль приборной панели самолёта.

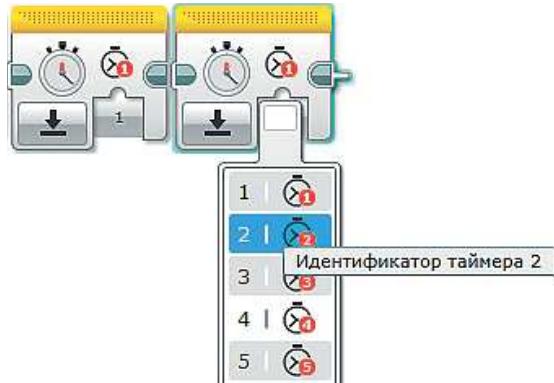
12. Добавь две команды **Таймер** (жёлтый блок).



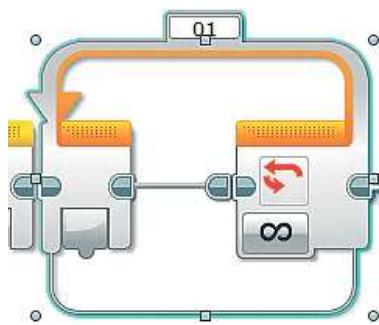
13. Выбери в каждой из них опцию **Сброс**.



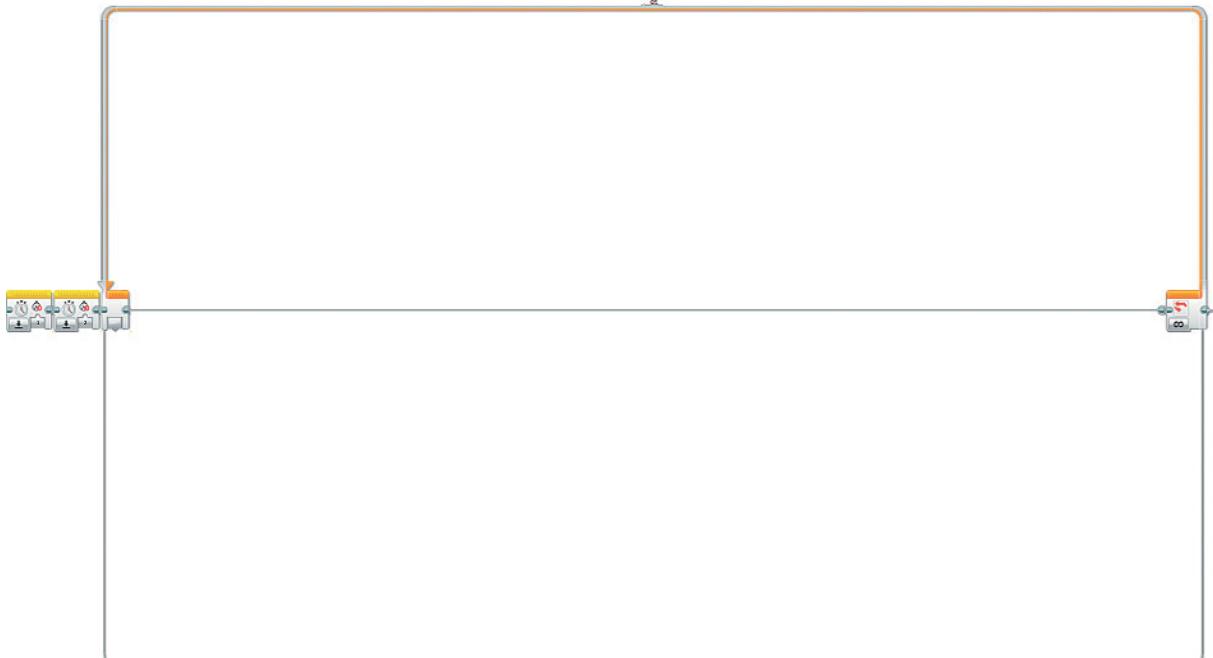
14. Оставь в первой команде **Идентификатор таймера 1**, а во второй выбери **2**. Первый таймер будет отвечать за показания одометра, а второй — за другие приборы.



15. Пришло время приступить к основной программе. Она будет заключена в большом бесконечном **Цикле** (оранжевый блок), который тебе и надо добавить.

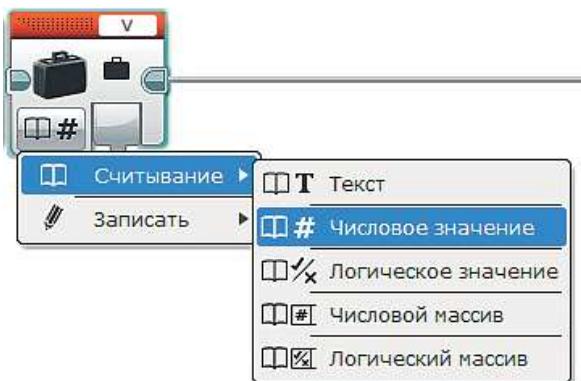


16. Программа внутри цикла будет очень большой, поэтому целесообразно его сразу расширить. Это можно сделать, потянув курсором мыши за кружочки и квадратики, окружающие цикл. В результате должно получиться примерно так:

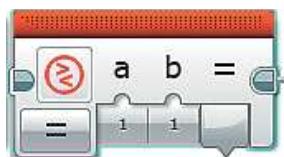




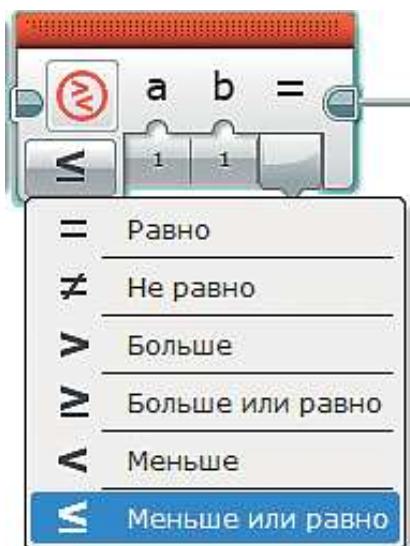
17. Далее все команды добавляются внутрь этого цикла, и первой из них будет команда **Переменная** (**красный блок**). В правом верхнем углу выбери для обработки переменную **Т**, опцию и входное значение оставь без изменений — каждый повтор цикла эта переменная будет обнуляться.



18. Следующей командой тоже будет **Переменная** (**красный блок**), в которой будет обрабатываться переменная **V**. Выбери в ней опцию **Считывание** → **Числовое значение**.

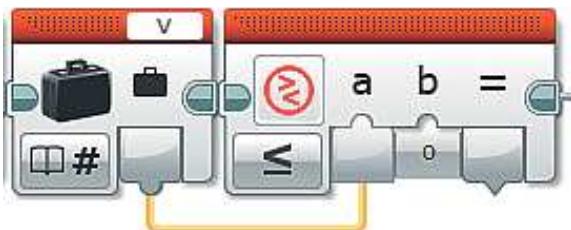


19. Теперь добавь команду **Сравнение** (**красный блок**).

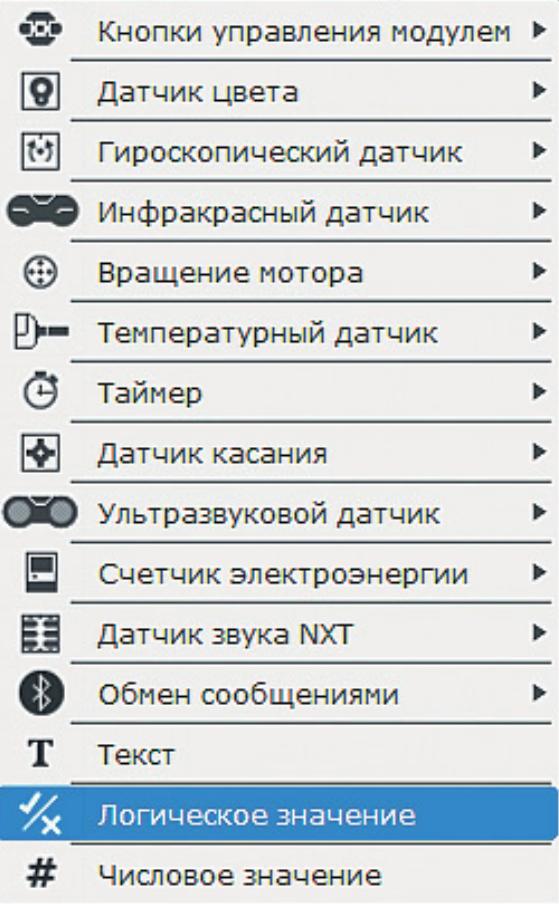
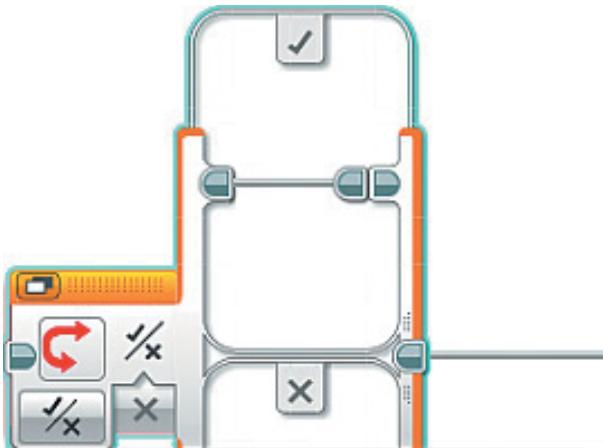


20. Выбери опцию **Меньше** или **равно**.

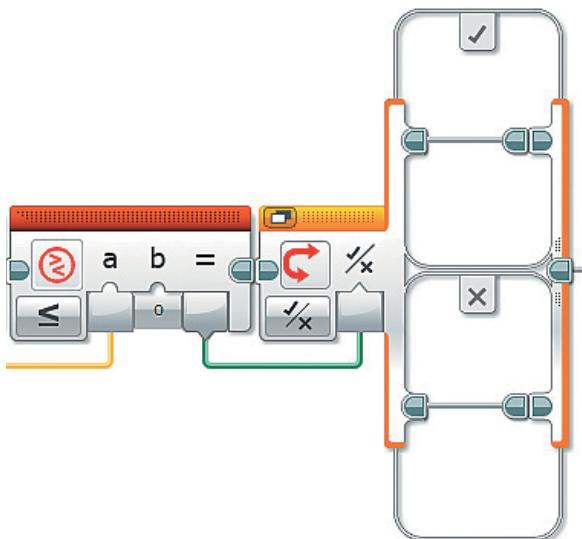
21. Соедини выход переменной **V** со входом **a** команды **Сравнение**. Введи во вход **b** с клавиатуры число **0**.



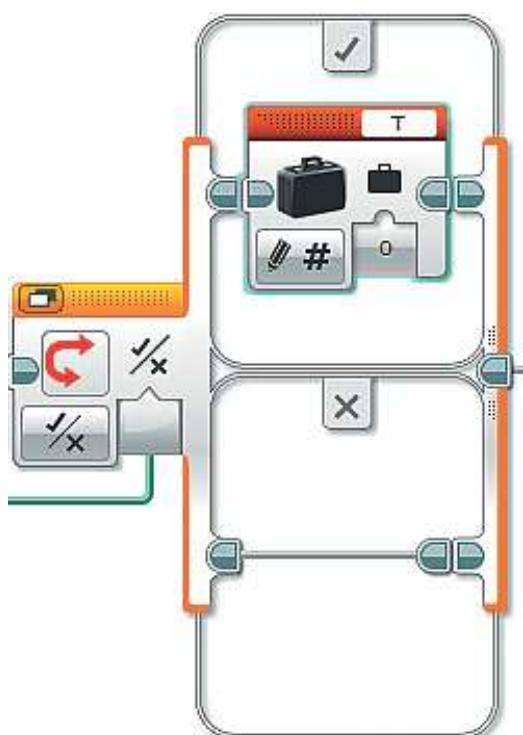
22. Добавь команду **Переключатель** (оранжевый блок) и выбери опцию **Логическое значение**.



23. Соедини выход « $=$ » команды **Сравнение** со входом **Переключателя**.



24. Последние три команды позволяют тебе проверить следующее условие: «*Если значение скорости меньше или равно нулю, то...*»; в случае истинности (выполнения) этого условия необходимо **обнулять** переменную **T**, а в случае ложности (невыполнения) устанавливать её равной **0,00027**. Разберёмся.



26. Добавь в нижнюю часть **Переключателя** (условие = ложь) такую же команду **Переменная** (красный блок) и выбери ту же переменную **T**, но входное значение будет другое: тебе нужно рассчитать пройденное расстояние. Оно вычисляется по формуле $S = V \cdot T$, где S — расстояние в километрах; V — скорость в км/ч; T — время в секундах. Так как значения S и V связаны с километрами и часами,



ми, значит, и значение T должно быть в часах. Поэтому переведём секунды в часы и заодно договоримся, что **частота обновления** показаний приборов на экране будет **1 раз в секунду**, или **1 Герц¹**. Тебе надо выполнить следующую цепочку преобразований:

Секунды → Минуты → Часы.

Ты знаешь, что секунды — это «маленькие» единицы измерения времени, в отличие от минут и часов. Чтобы узнать, какую часть составляет 1 секунда от 1 часа, посчитай, сколько же всего в нём секунд:

$$1 \text{ час} = 60 \text{ минут.}$$

$$1 \text{ мин} = 60 \text{ секунд.}$$

$$1 \text{ час} = 1 \cdot 60 \text{ мин} = 1 \cdot 60 \cdot 60 = 3600 \text{ секунд.}$$

В итоге: 1 час = 3600 секунд.

Теперь ты можешь сказать, что 1 секунда — это одна часть из 3600 частей, содержащихся в часе, то есть это обыкновенная дробь вида

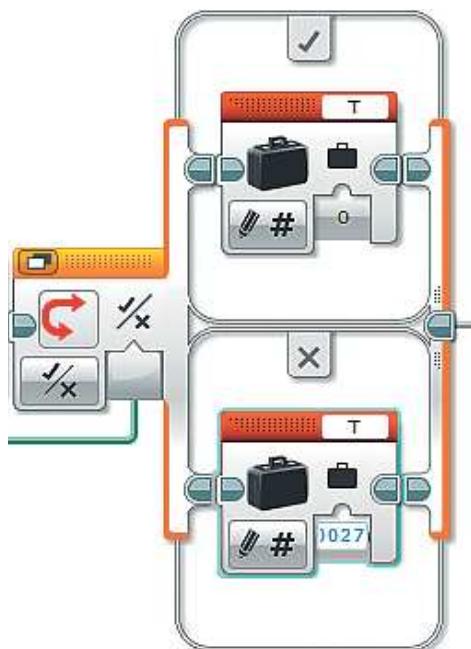
$$\frac{1}{3600}.$$

Черта дроби обозначает операцию деления. Разделим единицу на 3600:

$$1:3600 \approx 0,00027 \text{ часа.}$$

Вот так ты получил число для входного значения переменной **T**.

На этом первая часть программы окончена. В ней ты задал все необходимые первоначальные параметры и теперь готов «оживлять» свой авиасимулятор.

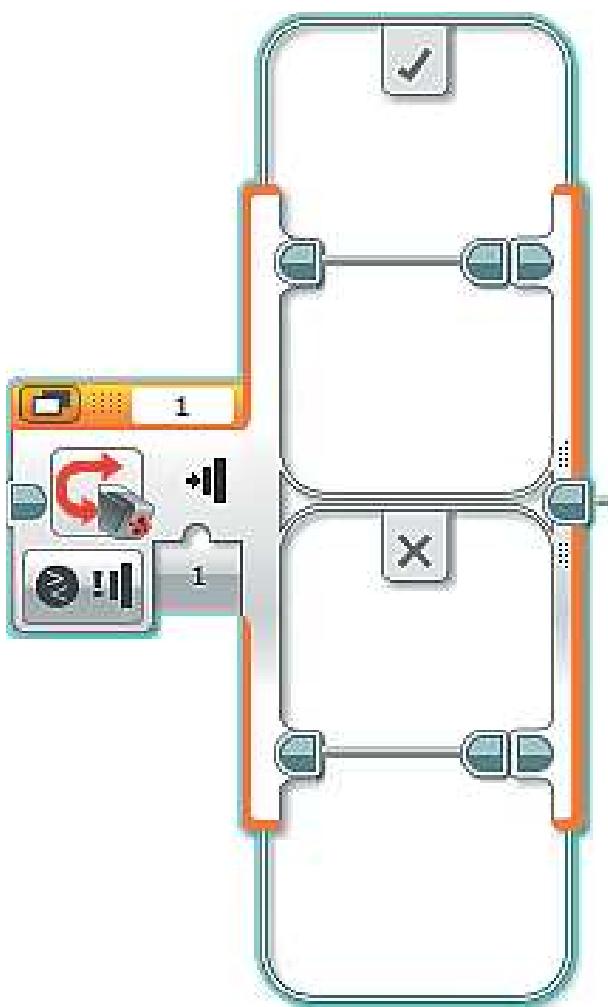


¹ Герц — единица частоты периодических процессов (например, колебаний) в Международной системе единиц (СИ).

Часть 2. Двигатели — на старт! Увеличение и уменьшение мощности турбин

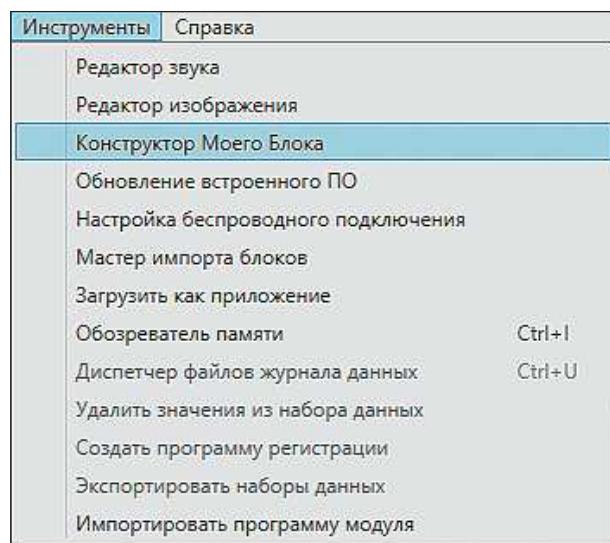
Логика. В этой части ты создаешь подпрограмму, которая будет отвечать за работу турбин самолёта и управление ими. Так как самолёт не может равномерно разгоняться на протяжении всего движения, в зависимости от набранной скорости он будет ускоряться или замедляться по-разному. Для увеличения мощности ты запрограммируешь правый датчик касания на штурвале, а для уменьшения — левый.

Подпрограмма необходима для компактности вида всей программы и удобной работы с разными её частями. Опытные программисты всегда выделяют из основной программы некоторые её части, которые занимаются отдельной обработкой однородных событий. В нашем случае — это обеспечение работы турбин.

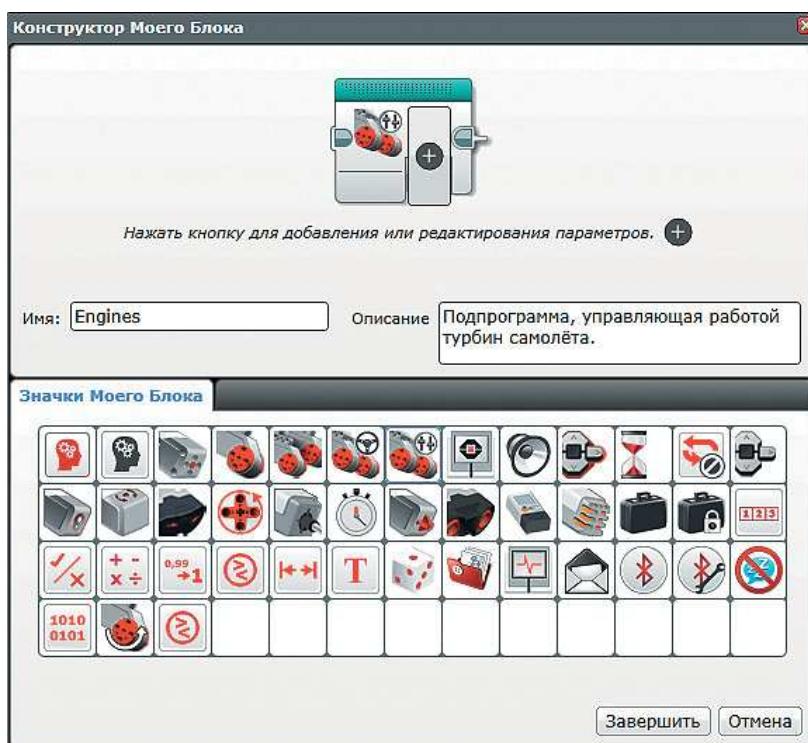


1. Перед тем как приступать к созданию подпрограммы, тебе необходимо сначала добавить хотя бы одну команду из неё. И у тебя это будет команда **Переключатель** (оранжевый блок).

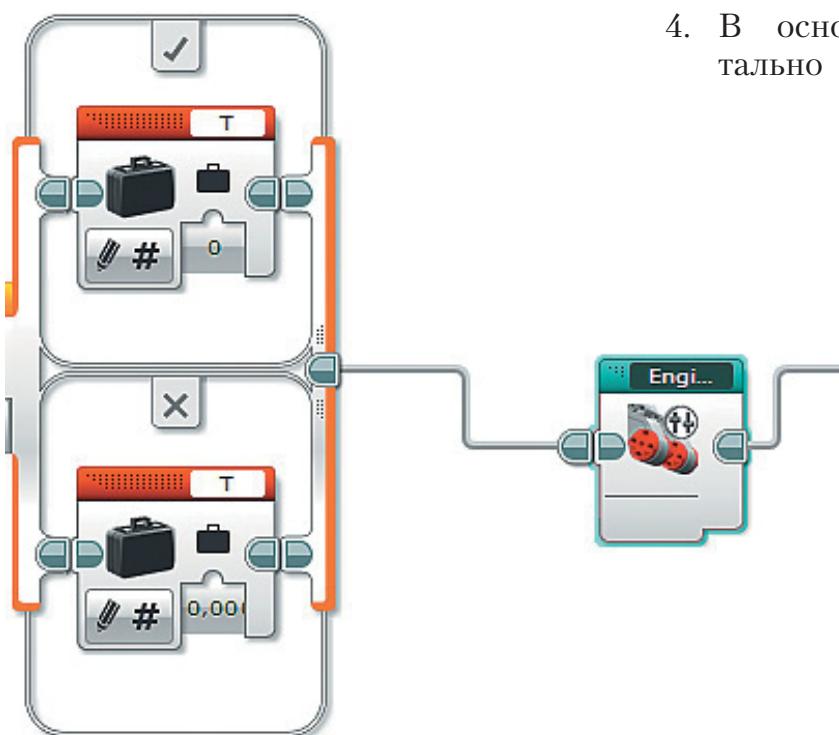
2. Когда ты добавляешь новую команду, она выделена голубым контуром, то есть активна. Если этого не произошло, кликни по выбранной команде мышью, и она активируется. Теперь помести эту команду в подпрограмму, то есть создай **свой блок**. Для этого в верхней части окна программы выбери меню **Инструменты** → → **Конструктор Моего Блока**.



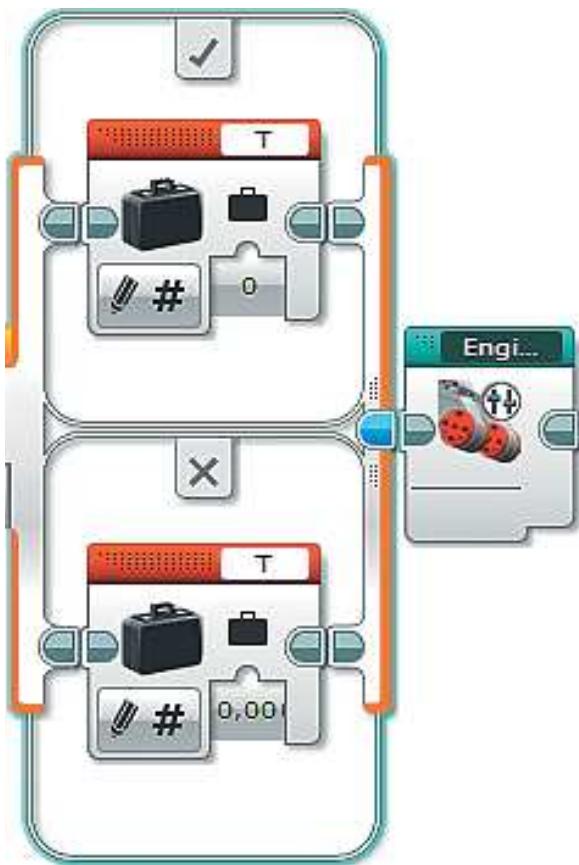
3. Задай в открывшемся окне имя подпрограммы, введи её краткое описание и выбери подходящую пиктограмму (маленькое изображение). Для турбин логично задать имя, например **Engines** (в переводе с английского — двигатели). Помни, чтобы легко ориентироваться в программах, всегда следует называть их сообразно тому, какую функцию они выполняют. Затем нажми кнопку **Завершить**. Краткое описание и пиктограмму можешь задать, как на рисунке или самостоятельно.



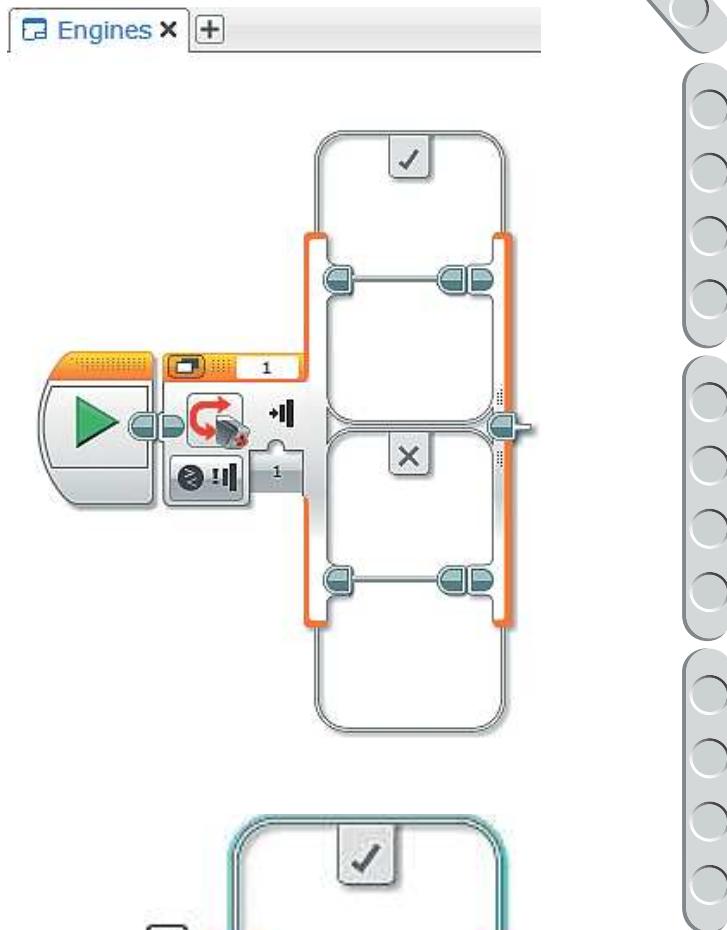
4. В основной программе моментально появится новая команда.



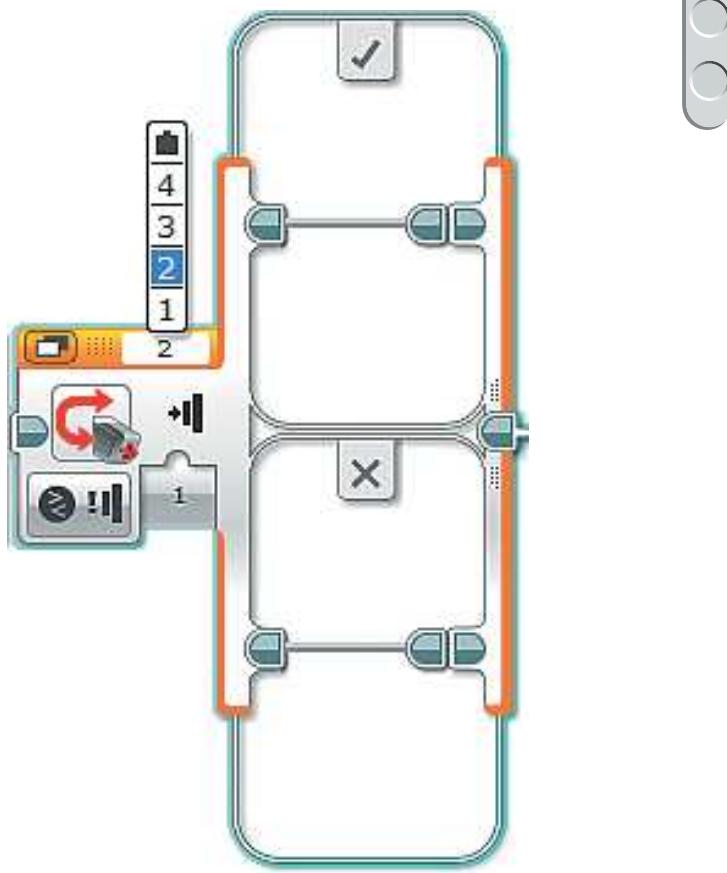
5. Для удобства работы можно убрать расстояние между этими командами, кликнув по выходной связке команды **Переключатель**.

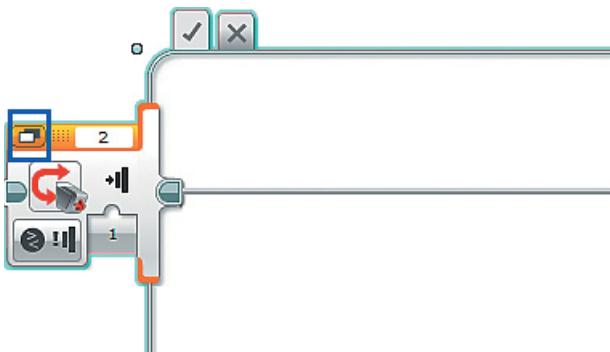


6. Дважды кликни по созданной команде **Engines**. На экране откроется новое рабочее поле для редактирования подпрограммы. Более того, сверху появится специальная вкладка с именем *Engines*.

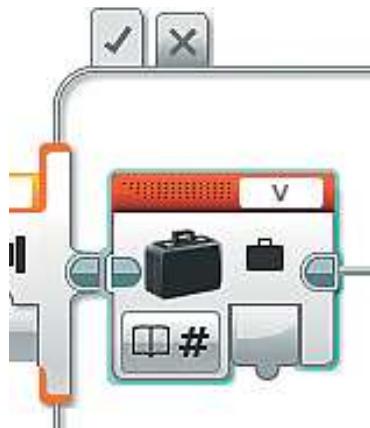


7. Теперь приступай непосредственно к самой подпрограмме. В команде **Переключатель** по умолчанию установлены нужная тебе опция **Датчик касания** → **Сравнение** → **Состояние** и проверка состояния датчика (нажат [1]). Проверь, чтобы в верхнем правом оконшке этой команды стоял номер порта, к которому подключён правый датчик касания. В нашем случае это порт № 2.





8. Для более удобной работы с **Переключателем** переведи его в **Вид с вкладками**, нажав на кнопку в левом верхнем углу этой команды. И сразу расширь эту команду, так как в ней будет содержаться всё остальное.



9. Сначала пропиши все команды для случая, когда условие выполнено, то есть датчик касания, подключённый к порту № 2, нажат. Добавь команду **Переменная** (красный блок) и выбери для обработки переменную **V**. Установи опцию **Считывание** → → **Числовое значение**.

10. Разберёмся, как же мы будем управлять работой турбин. Датчик касания, подключённый к порту № 2, то есть расположенный на штурвале справа, будет отвечать за увеличение мощности. Однако на экран будет выводиться скорость самолёта. Обратимся к инженерной хитрости и будем наращивать скорость в соответствии со следующей таблицей:

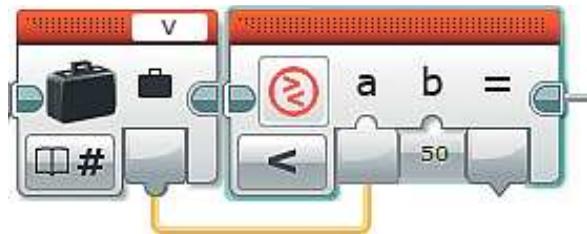
Скорость самолёта, км/ч	Соответствующее неравенство	Коэффициент прироста скорости, км/ч
Меньше 50	$V < 50$	0,05
От 50 до 100	$50 \leq V \leq 100$	0,1
От 100 до 200	$100 \leq V \leq 200$	0,3
От 200 до 300	$200 \leq V \leq 300$	0,5
От 300 до 600	$300 \leq V \leq 600$	0,7
От 600 до 850	$600 \leq V \leq 850$	1



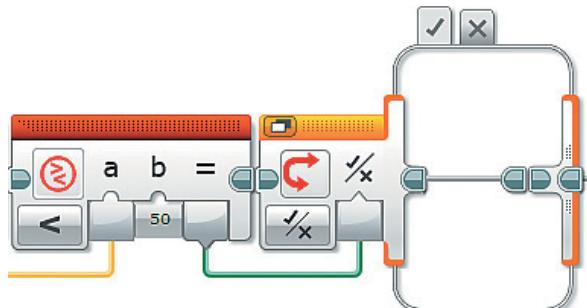
Пояснение. Подпрограмма будет считывать значение переменной **V**, отвечающей за скорость, и проверять, в каком диапазоне сейчас находится её значение. Чем выше скорость, тем «проще» самолёту её набирать; поэтому для разных диапазонов будет разный коэффициент прироста скорости (то есть на сколько будет увеличиваться значение этой переменной в каждый момент времени).

11. Добавь команду **Сравнение** (красный блок) и выбери опцию **Меньше**. Соедини выход команды **Переменная** со входом **a** команды **Сравнение**; введи во вход **b** с клавиатуры значение **50**.

Таким образом, ты обеспечил проверку самого первого диапазона скорости самолёта с помощью соответствующего неравенства.

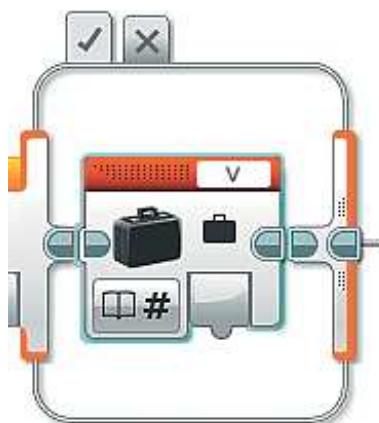


12. Результат этой проверки нужно обработать. Добавь команду **Переключатель** (оранжевый блок) и выбери опцию **Логическое значение**. Для удобства работы перейди к **Виду с вкладками**. Соедини выход «**=**» команды **Сравнение** с входом **Переключателя**.

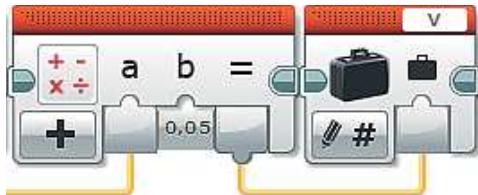


13. Ты пропишешь команды только для **Истины**, так как сейчас нас не интересует, что будет происходить в случае ложности, то есть если $V \geq 50$.

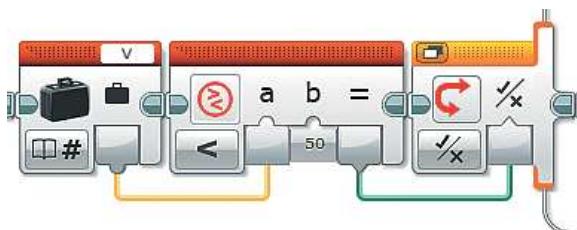
Здесь ты и обеспечишь прирост скорости. Для этого добавь команду **Переменная** (красный блок) и выбери для обработки переменную **V**, затем установи опцию **Считывание → Числовое значение**.



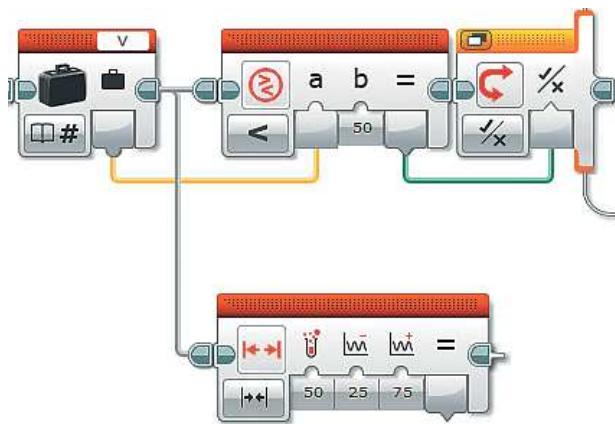
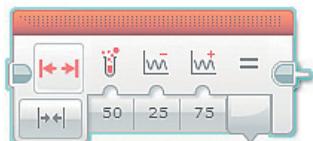
14. Добавь команду **Математика** (красный блок), опцию **Сложение** оставь без изменений. Соедини выход команды **Переменная** с входом **a** команды **Математика**, введи во вход **b** с клавиатуры коэффициент прироста скорости **0,05**.



15. Результат сложения перезапиши в переменную **V**. Для этого добавь команду **Переменная**, для обработки выбери переменную **V** и соедини выход «**=**» команды **Математика** с входом переменной **V**.

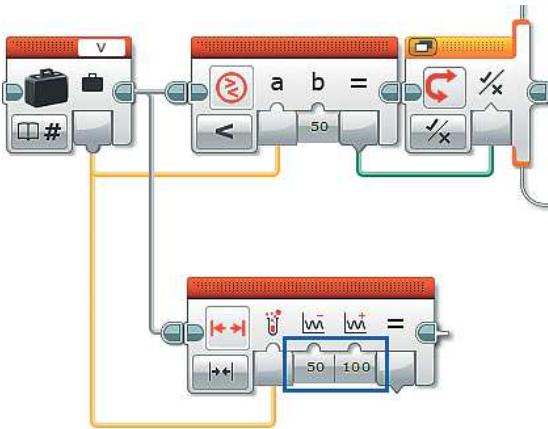


16. Теперь реализуем набор скорости в другом скоростном диапазоне. Для этого размести команду **Интервал** (красный блок) чуть ниже написанного фрагмента программы.

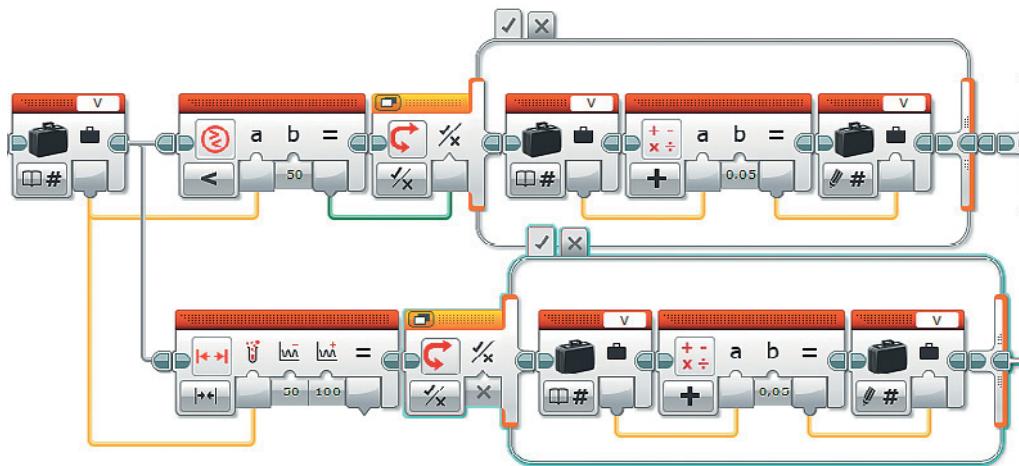


17. Заметь, что эта команда пока неактивна, так как она ни к чему не привязана. Нужно задать с клавиатуры момент времени, с которого она должна начать выполняться в программе. Мы проверяем величину переменной **V** сразу после считывания её значения. Поэтому соедини эту команду с командой **Переменная**. Для этого курсором мыши перетащи соединительный шлейф этой команды к соединительному порту команды **Интервал**.

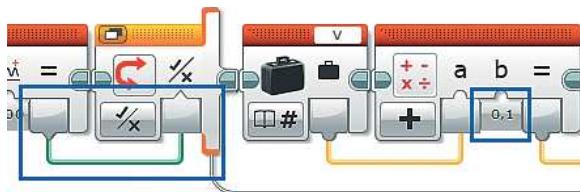
18. Выход команды **Переменная** соедини с входом **Тестовое значение** команды **Интервал**, а во входы **Нижняя граница** и **Верхняя граница** с клавиатуры запиши соответственно числа **50** и **100**.



19. Следующий фрагмент программы, регулирующий изменение скорости, будет точно таким же, как и в пунктах 12–15. Для ускорения процесса написания программы можно копировать одинаковые её части. Сейчас тебе нужно скопировать команду **Переключатель** с её внутренним наполнением и настройками. Для этого зажми клавишу **Ctrl** на клавиатуре и перетащи курсором мыши команду **Переключатель** вниз, подсоединив к команде **Интервал**.

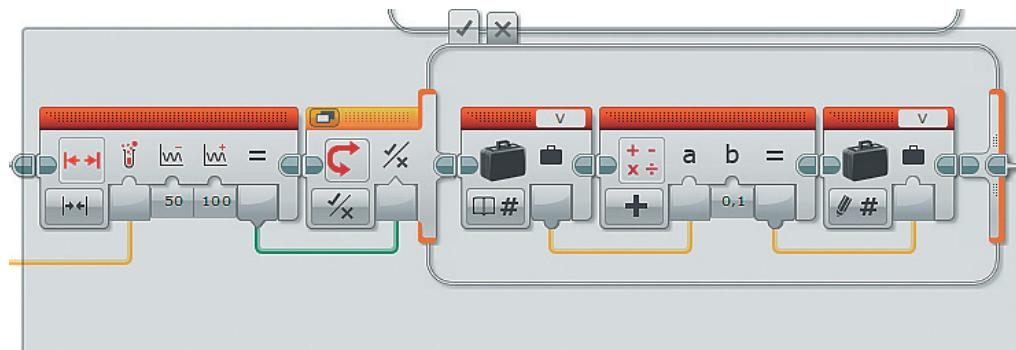


20. Соедини выход « \Rightarrow » команды **Интервал** с входом **Переключателя**. Единственное изменение, которое нужно внести в скопированный фрагмент, — это коэффициент прироста скорости: во вход **b** команды **Математика** запиши новое значение: **0,1**.



21. Осталось ещё четыре скоростных интервала. По сравнению с предыдущим фрагментом в них будут уже два отличия — скоростной интервал и коэффициент прироста скорости. Поэтому тебе нужно копировать только две команды, **Интервал** и **Переключатель**, и менять соответствующие значения.

Чтобы скопировать сразу две команды, **выдели** их. Это можно сделать с помощью мыши, растянув область выделения:

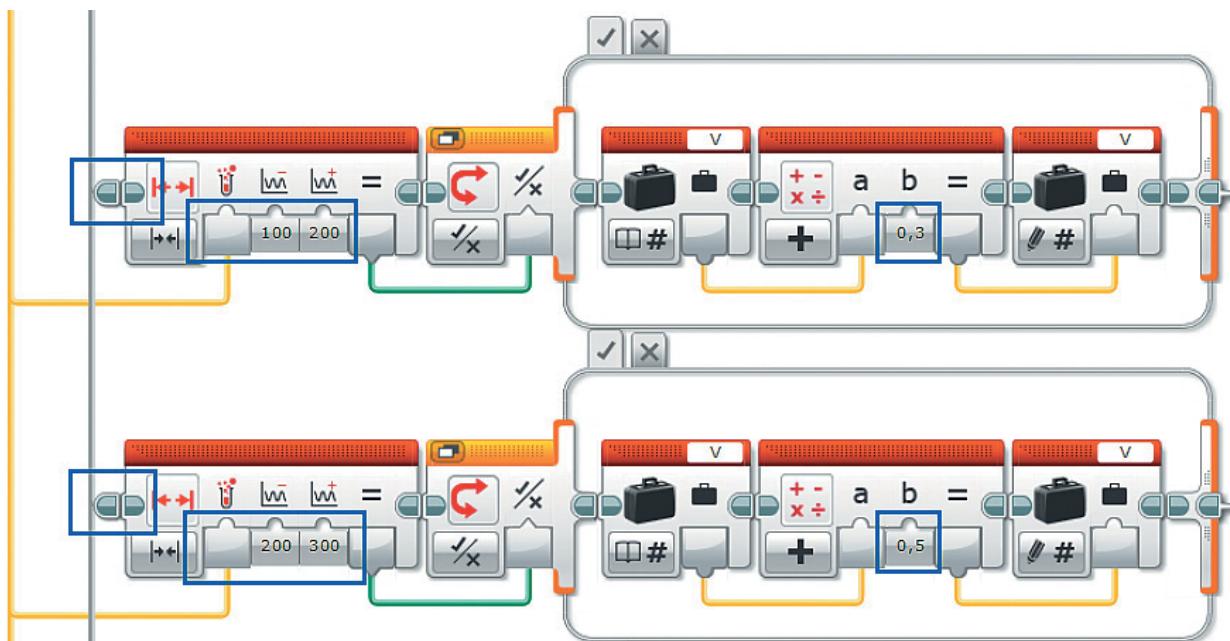


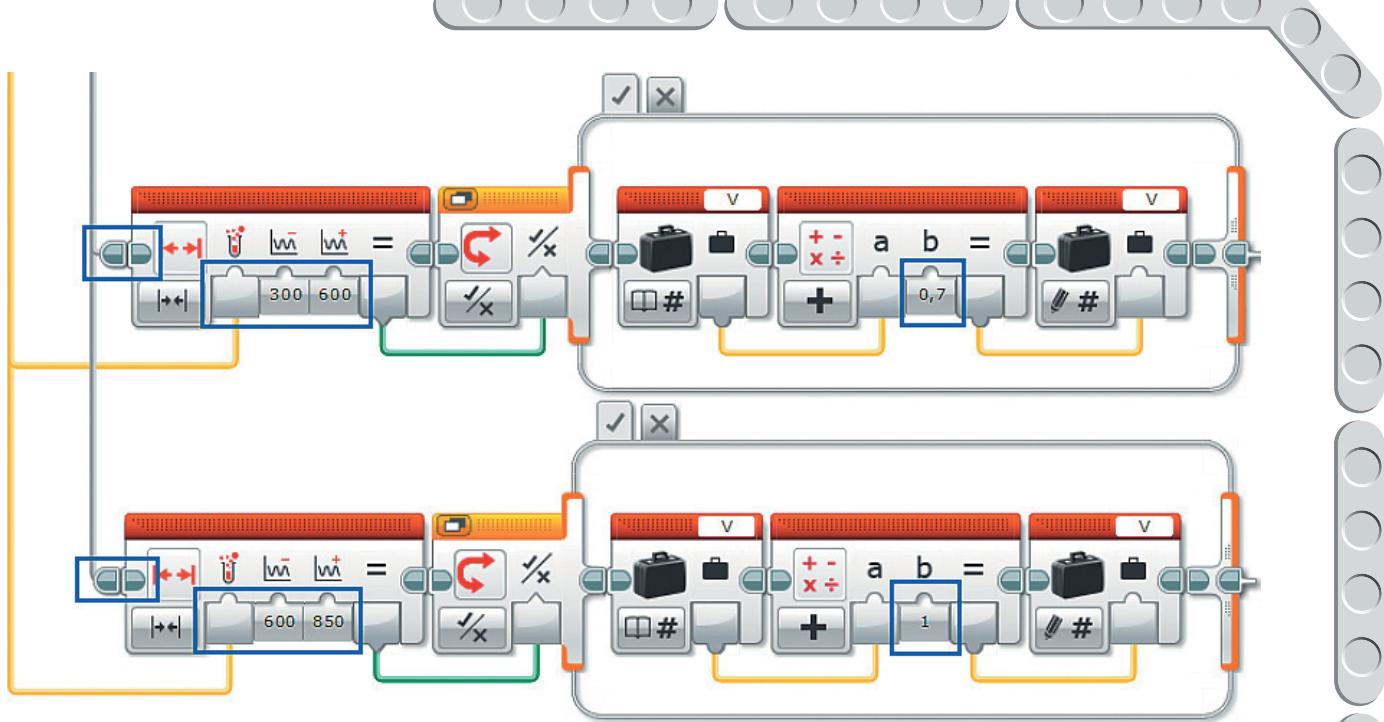
Внимание!

Не забывай соединять команду **Переменная** с командой **Интервал шлейфом**, а также их выход и вход.

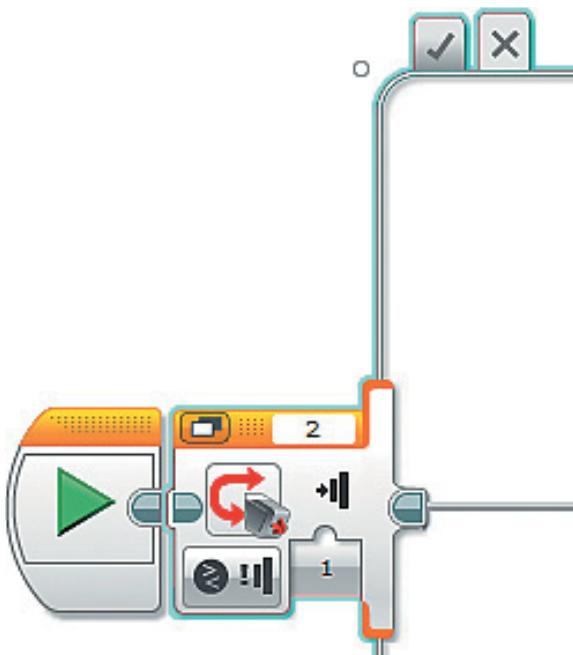
или, зажав клавишу **Ctrl**, кликать по тем командам, которые ты будешь копировать, — они сразу же будут подсвечиваться голубой рамкой.

22. Проверь себя, должны получиться следующие фрагменты:

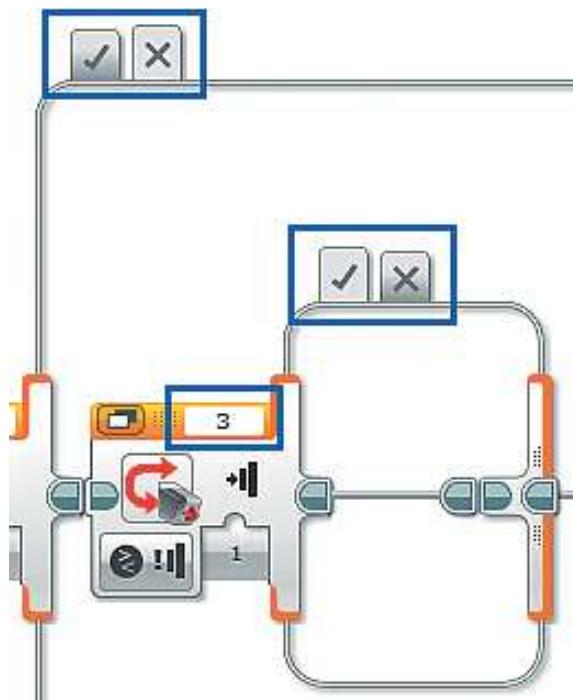




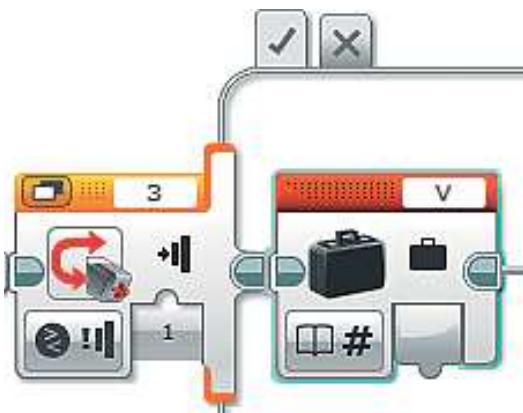
23. Вернись в начало подпрограммы. Ты прописал ту часть, в которой условие самого первого внешнего переключателя выполнено, то есть **правый датчик касания** был действительно нажат и мощность турбин необходимо увеличивать. Теперь тебе надо рассмотреть вариант, когда проверочное условие оказалось ложным, то есть правый датчик касания нажат не был. В этом случае появляется вторая возможность: а был ли при этом нажат **левый датчик касания**? Если да, то мощность турбин нужно **уменьшать**, то есть обеспечить замедление и торможение воздушного судна. Перейди во внешнем **Переключателе** на вкладку **действий при ложном условии**.



24. Добавь внутрь этого переключателя ещё одну команду **Переключатель** (оранжевый блок) и перейди к **Виду с вкладками**. Опции оставь без изменения. Проверь в правом верхнем окошке номер порта, к которому подключён левый датчик касания (в нашем проекте это порт № 3).

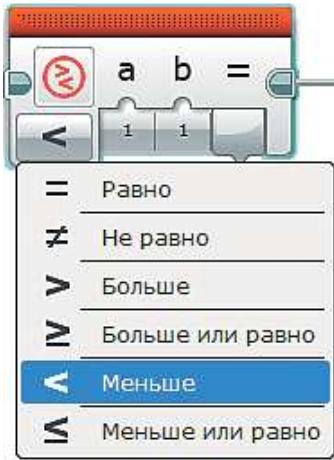


25. Сразу расширь рабочее поле внутреннего **Переключателя**, а затем напиши практически такой же фрагмент программы, как при увеличении мощности, только коэффициент прироста скорости станет **коэффициентом убывания** скорости, который нужно будет вычитать. Первой командой будет **Переменная** (красный блок). Для обработки выбери переменную **V** и опцию **Считывание → Числовое значение**.

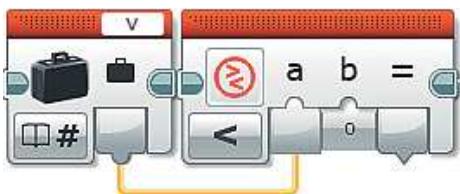


26. Перед началом работы по уменьшению мощности турбин нужно исключить ошибку ухода показаний спидометра **ниже нуля** (при увеличении мощности за это автоматически отвечала команда **Интервал**).

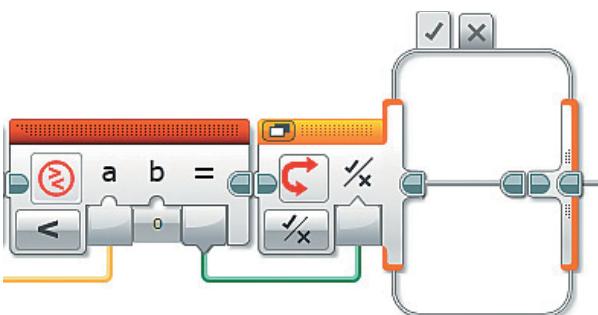
Для этого добавь команду **Сравнение** (красный блок) и выбери опцию **Меньше**.



27. Соедини выход команды **Переменная** с входом **a** команды **Сравнение**, введи во вход **b** с клавиатуры число **0**.

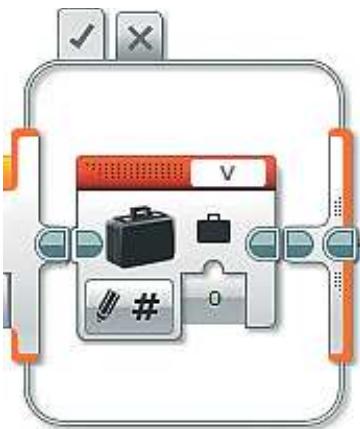


28. Теперь добавь команду **Переключатель** (оранжевый блок), перейди к **Виду с вкладками** и выбери опцию **Логическое значение**. Соедини выход «**=**» команды **Сравнение** с входом **Переключателя**.



29. Если скорость действительно попытается «убежать» ниже нуля (в отрицательные значения), «верни» её к нулю.

Для этого добавь внутрь этого **Переключателя** команду **Переменная** (красный блок) и выбери для обработки переменную **V**. По умолчанию в ней настроена запись нуля. Именно это тебе и нужно. Теперь ниже нуля скорость не опустится.



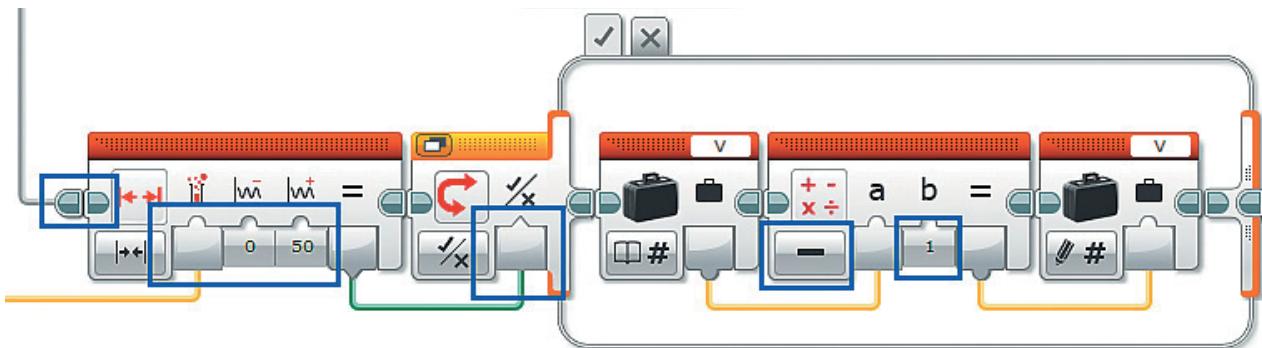
30. Далее пропиши фрагмент программы, регулирующий уменьшение скорости.

Для удобства воспользуйся таблицей:

Скорость самолёта, км/ч	Соответствующее неравенство	Коэффициент убывания скорости, км/ч
От 0 до 50	$0 \leq V \leq 50$	1
От 50 до 100	$50 \leq V \leq 100$	0,7
От 100 до 200	$100 \leq V \leq 200$	0,5
От 200 до 300	$200 \leq V \leq 300$	0,3
От 300 до 600	$300 \leq V \leq 600$	0,1
Свыше 600	$V > 600$	0,05

Пояснение. Чем ниже скорость, тем «проще» самолёту тормозить.

31. Реализация каждого из диапазонов, кроме последнего, практически ничем не отличается от наращивания скорости. Логика первого фрагмента такова: в диапазоне от 0 до 50 км/ч скорость должна уменьшаться на 1 км/ч в каждый момент времени. Тогда получится следующая цепочка команд:

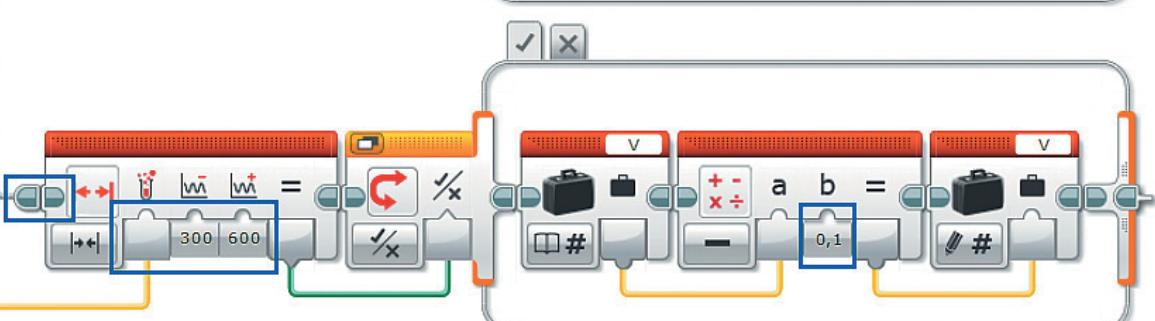
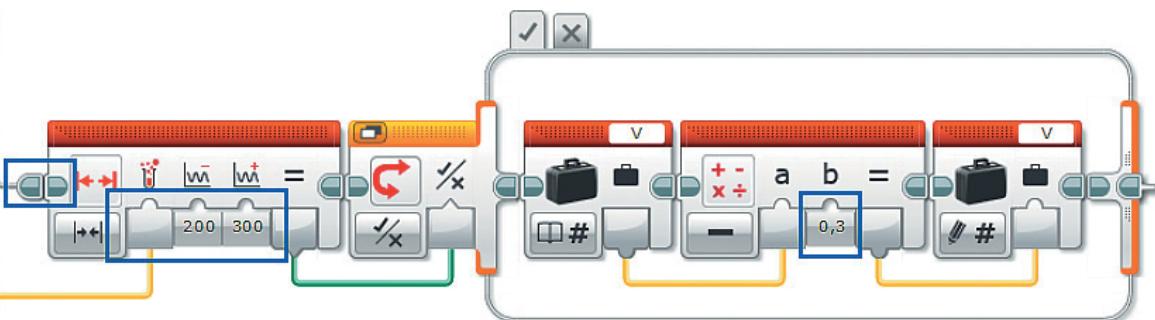
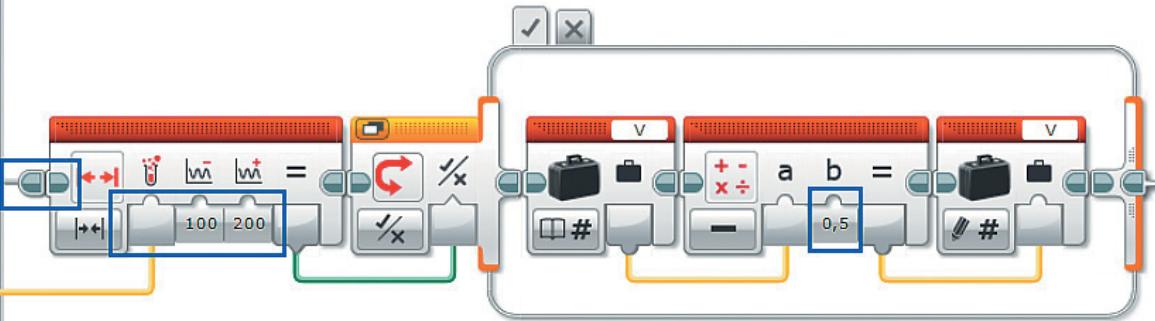
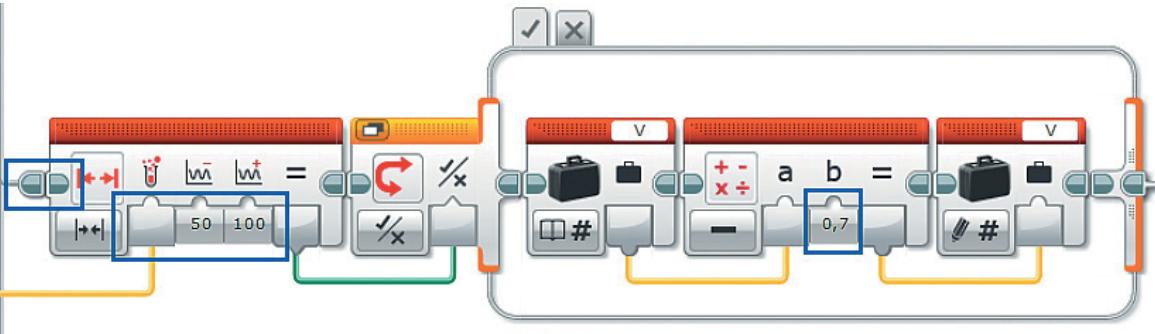


Внимание!

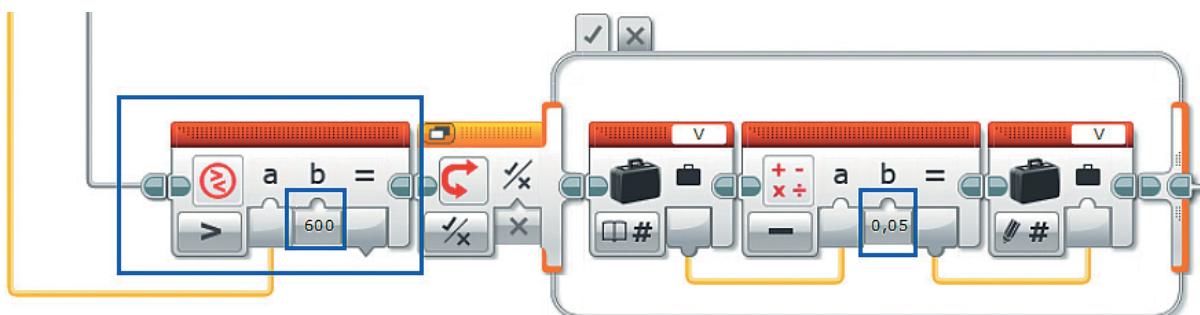
Не забывай соединять команду Переменная с командой Интервал шлейфом, а также их выход и вход.

32. Для следующих четырёх диапазонов скорости фрагменты будут аналогичные, поэтому их удобно копировать, изменяя границы диапазона и значение коэффициента убывания скорости.

Проверь себя. Должны получиться следующие фрагменты:



33. Последний фрагмент в этой подпрограмме отличается проверкой условия. Здесь значение скорости сравнивается только с одним числом, поэтому команда **Интервал** заменится на **Сравнение**:

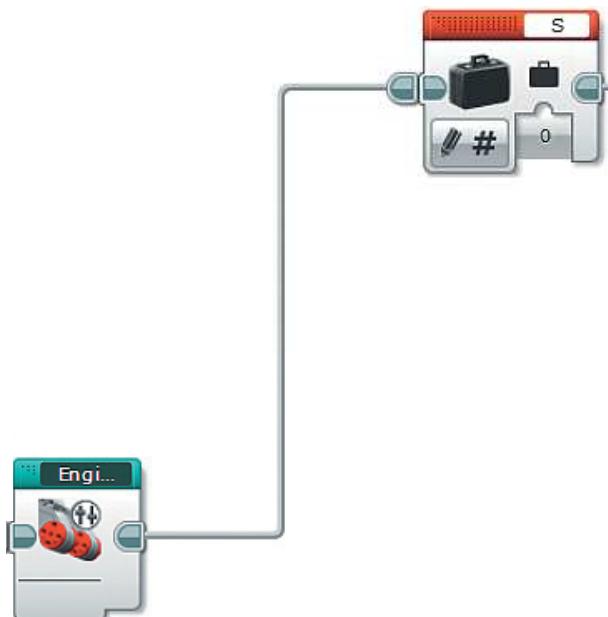


Поздравляем! Первая подпрограмма составлена, благодаря ей твой самолёт готов отправиться в путешествие!

Часть 3. Скорость, крен, тангаж — завернём крутой вираж! Показания спидометра и авиагоризонта

Логика. В третьей части программирования ты выведешь на приборную панель первые показания приборов, а именно скорость и углы крена и тангажа.

Для удобства вывода на экран и компактности программы воспользуйся возможностью составления текстовых строк из нескольких различных объектов: чисел, значений переменных и символов.

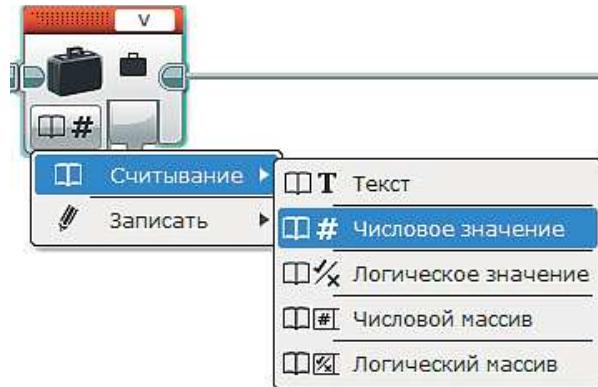


1. Вернись в основную программу **Main**. После подпрограммы **Engines** у тебя будут ещё три ветви, и сейчас ты пропишешь первую из них. Для удобства их лучше расположить друг под другом, поэтому первую команду **Переменная (красный блок)** добавь чуть правее и выше от **Engines**.

Внимание!

Не забудь соединить эти команды шлейфом.

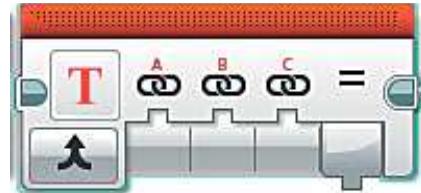
2. Выбери для обработки переменную **V** и опцию **Считывание** → → **Числовое значение**.



3. Далее добавь две команды **Вращение мотора** (жёлтый блок). Проверь, чтобы в правом верхнем окошке значились порты **B** и **C**.



4. Теперь добавь команду **Текст** (красный блок). С её помощью ты сможешь вывести на экран программируемого модуля сразу несколько различных значений.

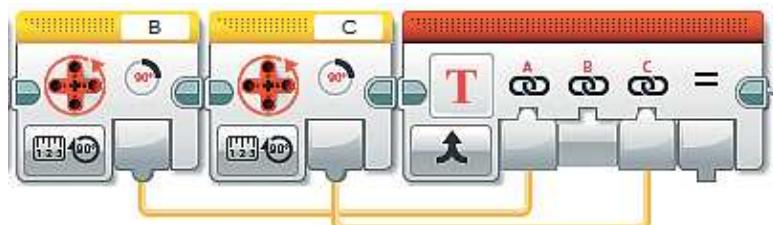


Пояснение. Эта команда позволяет сформировать для отображения на экранчике программируемого модуля текст, состоящий из трёх различных фрагментов или, как говорят программисты, из трёх строк.

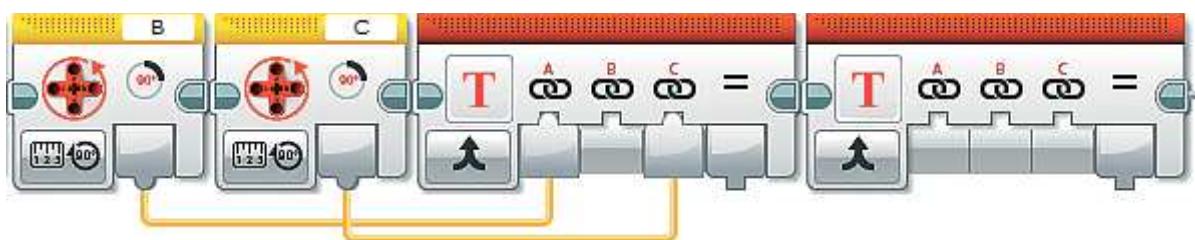
5. Мы зададим следующий порядок в тексте: <Угол тангажа><Пробел><Угол крена><Пробел><Скорость>.

Получилось пять условных объектов для вывода, значит, одной командой **Текст** тебе не обойтись.

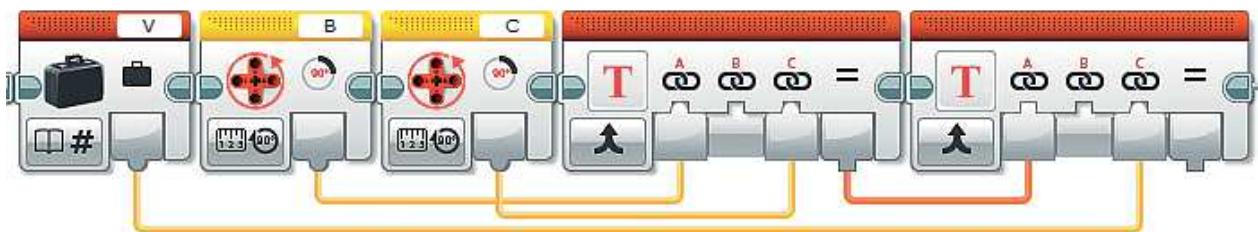
Для начала соедини выход первой команды **Вращение мотора** с входом **A**, во входе **B** с клавиатуры введи один пробел, выход второй команды **Вращение мотора** соедини со входом **C**.



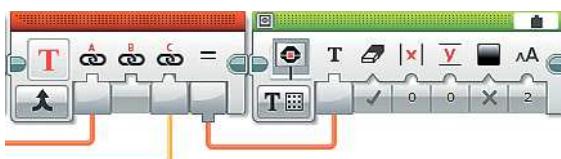
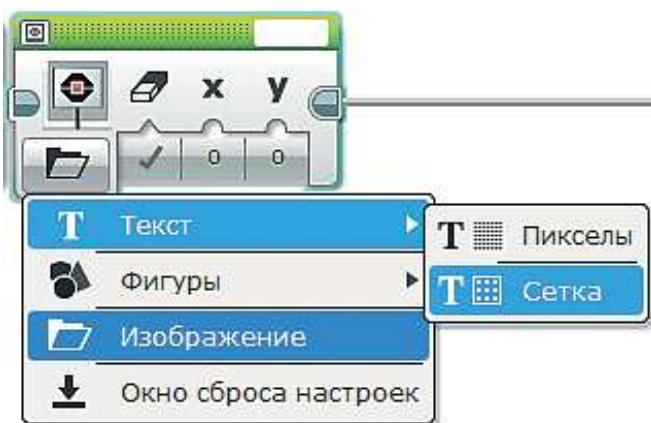
6. Добавь ещё одну команду **Текст** (красный блок).



7. Соедини выход «==» первой команды **Текст** с входом **A**, во вход **B** с клавиатуры введи один пробел, выход переменной **V** соедини со входом **C** второй команды **Текст**.



8. Осталось вывести результат этих слияний на экран EV3. Добавь команду **Экран** (зелёный блок) и выбери опцию **Текст → Сетка**.



9. Выбери в правом верхнем окошке пункт **Проводной**, а затем соедини выход «==» с входом **Текст**.

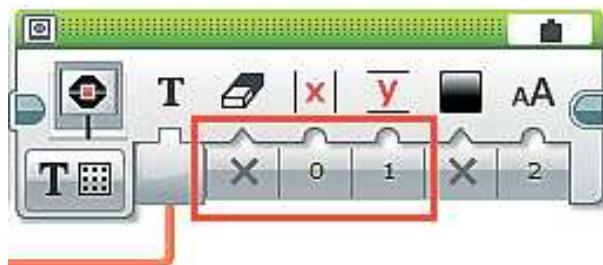
10. Осталось настроить параметры вывода на экран.

Очистить экран — ложность (тебе ведь не нужно, чтобы экран то и дело мерцал, очищаясь).

Введи с клавиатуры во входы **x** и **y** значения: **X = 0, Y = 1**.

В результате текст будет выводиться от левого края экрана, но чуть ниже первой строки.

Остальные параметры оставь **без изменений**.



Часть 4. Тревога! Тревога! Система сигнализации об опасном уровне тангажа

Логика. Ты подошёл к одному из очень важных этапов — созданию системы сигнализации опасного угла тангажа. Если бы в сложных технических устройствах, тем более таких, как система управления воздушным судном, не было подобных систем, людям было бы очень сложно замечать и ликвидировать опасные ситуации и неисправности.

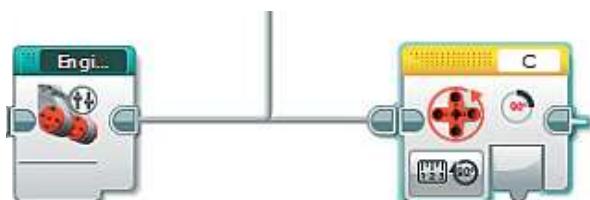
Наша система сигнализации будет подавать сигнал **Down** (в переводе с английского — вниз), если нос самолёта поднят слишком высоко, и сигнал **Up** (от английского — вверх), если нос слишком низко опущен.

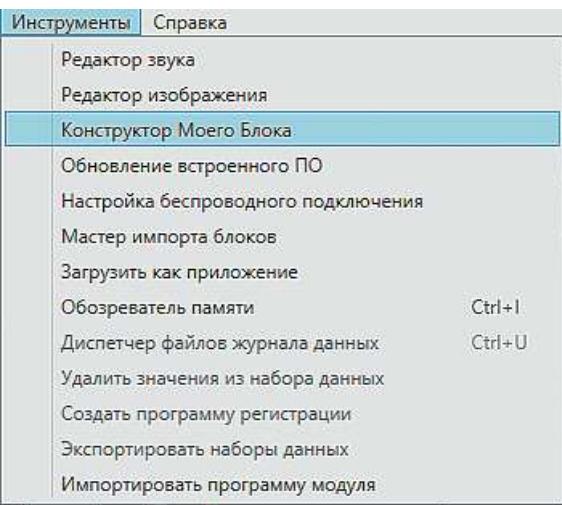
1. Чтобы узнать, действительно ли нос самолёта находится в опасной зоне, нужно считать показания мотора, который отвечает за тангаж. В нашем случае это мотор, подключенный к порту **C**.

Так как данная система представляет собой самостоятельную часть программы, естественно, ты создаешь её как подпрограмму.

Для этого добавь команду **Вращение мотора** (жёлтый блок), расположи её напротив команды **Engines** и соедини их с помощью шлейфа. Опцию оставь без изменений, в данном случае нам нужно знать, на сколько градусов был повернут мотор, то есть как сильно отклонён штурвал вперёд или назад.

Проверь, чтобы в правом верхнем окошке значился порт **C**.





2. Теперь приступай к составлению подпрограммы. Команда **Вращение мотора** должна быть выделена (подсвечен голубым контуром). В верхней части программы выбери меню **Инструменты** → **Конструктор Моего Блока**.

3. Задай в открывшемся окне имя подпрограммы, введи её краткое описание и выбери подходящую пиктограмму. Логично задать имя **Alarm** (в переводе с английского — тревога). Помни, чтобы легко ориентироваться в программах, следует называть их сообразно тому, какую функцию они выполняют. Затем нажми кнопку **Завершить**. Краткое описание и пиктограмму можешь задать, как на рисунке, или самостоятельно.



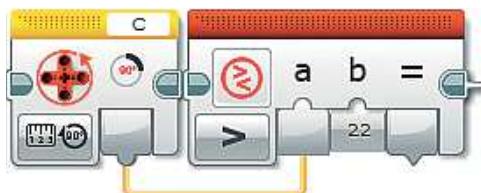
4. В основной программе **Main** появится новая команда.



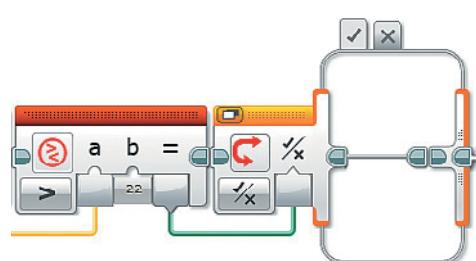
5. Дважды кликни по ней. На экране откроется новое рабочее поле для редактирования подпрограммы. Более того, сверху появится вкладка с именем *Alarm*.

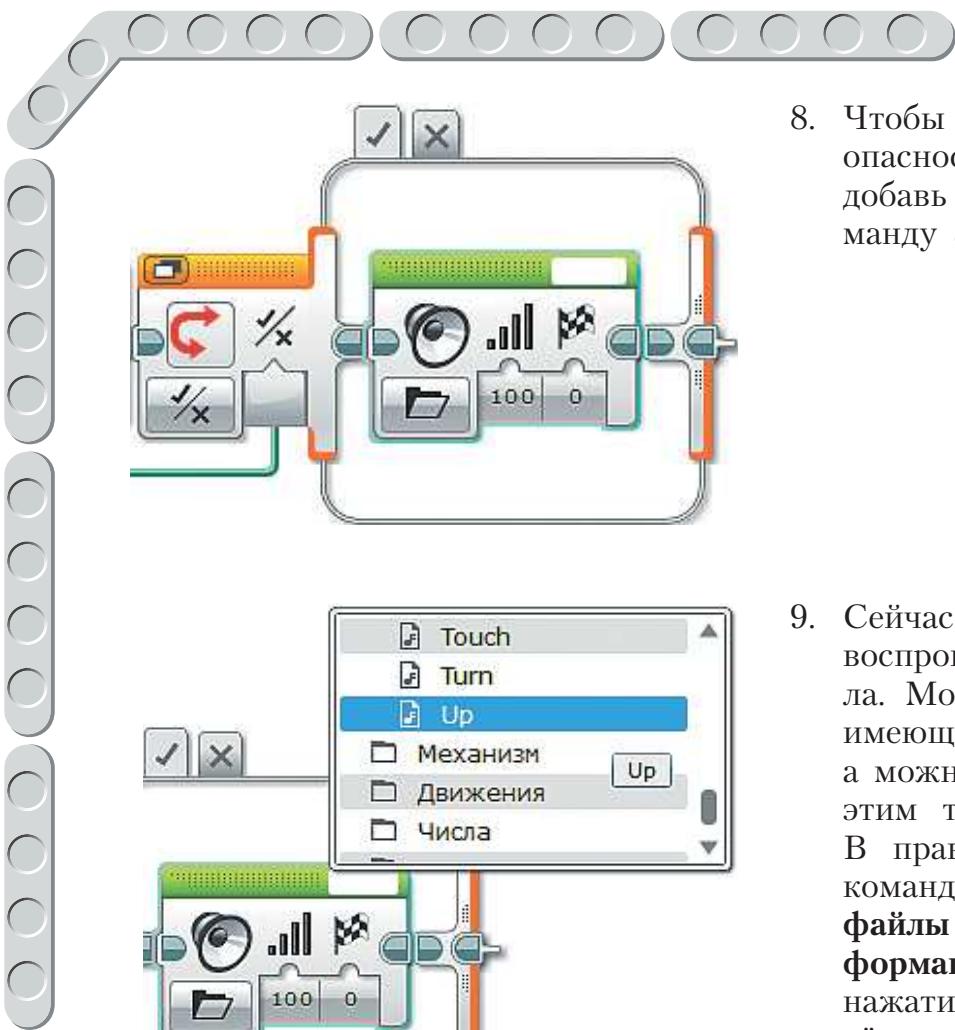


6. Шкала градусов, показывающая угол отклонения носа самолёта, может иметь как отрицательные, так и положительные значения. Для нашего расположения вертикального мотора подъём носа (штурвал на себя) соответствует перемещению по шкале в отрицательном направлении (отрицательный тангаж), а опускание носа (штурвал от себя) соответствует перемещению по шкале в положительном направлении — угол в градусах увеличивается (положительный тангаж). Отработаем для начала ситуацию с критическим уровнем опущенного носа самолёта, составляющим угол в 22 градуса. При таком уклоне самолёт переходит в пикирование. Добавь команду **Сравнение** (красный блок), выбери опцию **Больше** и соедини выход команды **Вращение мотора** и вход **a** команды **Сравнение**. Во вход **b** запиши число **22** — критическое значение.



7. Далее добавь команду **Переключатель** (оранжевый блок), перейди к **Виду с вкладками** и выбери опцию **Логическое значение**. Соедини выход «**=**» команды **Сравнение** с входом **Переключателя**.





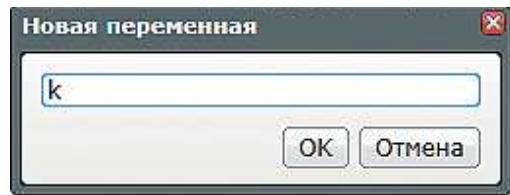
8. Чтобы просигнализировать об опасности и дать инструкцию **Up**, добавь внутрь **Переключателя** команду **Звук** (зелёный блок).



9. Сейчас эта команда настроена на воспроизведение звукового файла. Можно выбрать файл из уже имеющихся в библиотеке звуков, а можно добавить собственный (с этим ты познакомишься позже). В правом верхнем окошке этой команды выбери папку **Звуковые файлы LEGO**, в ней — папку **Информация** и затем файл **Up**. При нажатии на него он воспроизведётся.

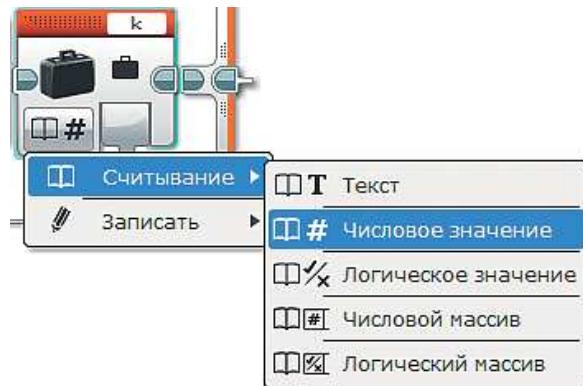
10. Ты понимаешь, что, если долго держать самолёт в таком состоянии, произойдет авиакатастрофа. Это тебе тоже надо учсть. Чтобы понять, что самолёт перешёл в сваливание и разбился, можно посчитать, сколько раз был подан сигнал предупреждения. Для этого добавь команду **Переменная** (красный блок) и выбери в правом верхнем окошке пункт **Добавить переменную**.

11. Введи в открывшемся окне имя **k** (эта переменная понадобится тебе для подсчёта количества поданных сигналов) и нажми **OK**.

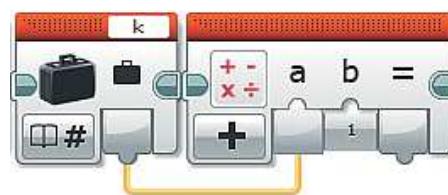


12. Переменная **k** будет увеличиваться на единицу после каждого поданного сигнала. Если сигнал не подаётся, то есть самолёт находится в нормальном положении, эта переменная будет заново обнуляться.

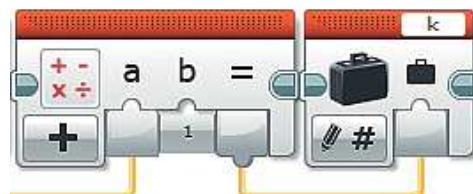
Для этого выбери опцию **Считывание** → **Числовое значение**.



13. Далее добавь команду **Математика** (красный блок). Соедини выход переменной **k** с входом **a** команды **Математика**. Остальные опции и параметры оставь без изменений.

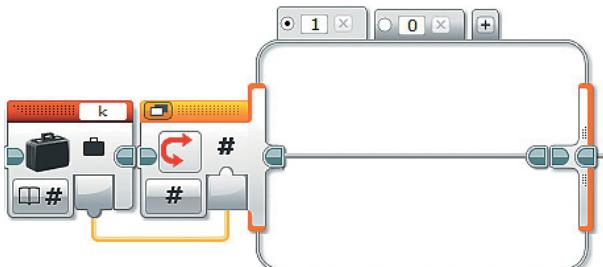


14. Добавь ещё одну команду **Переменная** (красный блок). Выбери для обработки переменную **k** и соедини выход «**=**» команды **Математика** с входом переменной **k**.

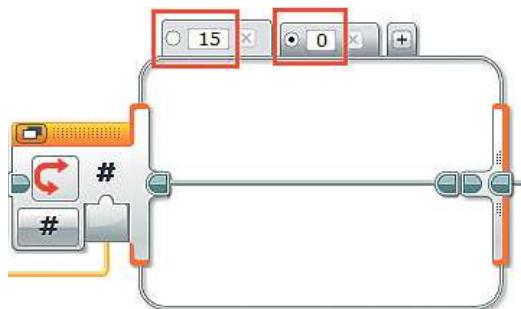


15. Давай договоримся, что если сигнал **Up** был подан **15** раз, то эта ситуация будет считаться крушением, после которого авиасимулятор прекращает свою работу. Для проверки добавь команду **Переменная** (красный блок), выбери для обработки переменную **k** и установи опцию **Считывание** → **Числовое значение**.





16. Добавь команду **Переключатель** (оранжевый блок), перейди к **Виду с вкладками** и выбери опцию **Числовое значение**. Затем соедини выход переменной **k** с входом **Переключателя**.

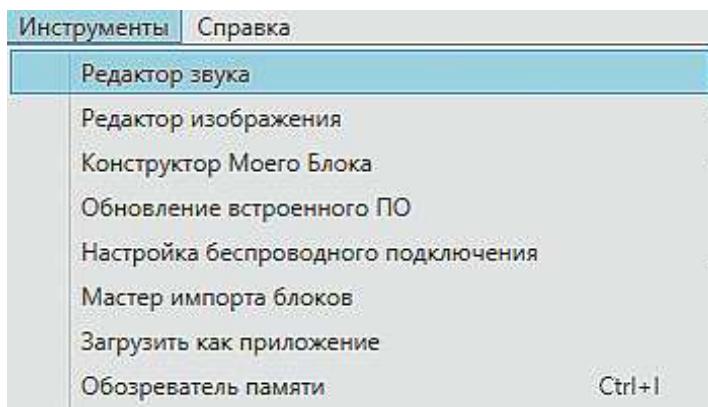


17. В верхней части **Переключателя** вместо единицы задай значение **15** — количество повторов сигнала, а второе значение оставь равным **нулю**, только переставь к нему точку выбора значения по умолчанию. Для этого кликни в кружок рядом с нулём.

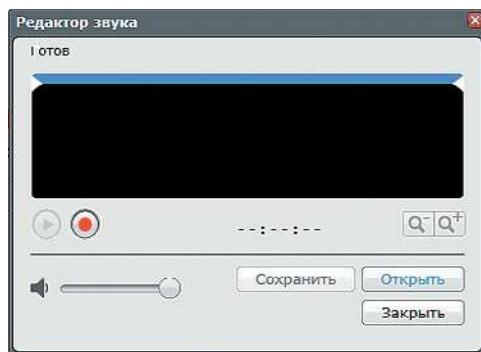
Пояснение. Данный **Переключатель** будет проверять, достигла ли переменная **k** значения, равного **15**. Для этого случая ты пропишешь определённые действия. Если же **k** будет иметь иное значение, то **Переключатель** будет считать его равным нулю и ничего не делать, так как для этой ситуации ты не будешь задавать ему действий на исполнение.

Важно! Переключатель **НЕ** обнуляет переменную **k**.

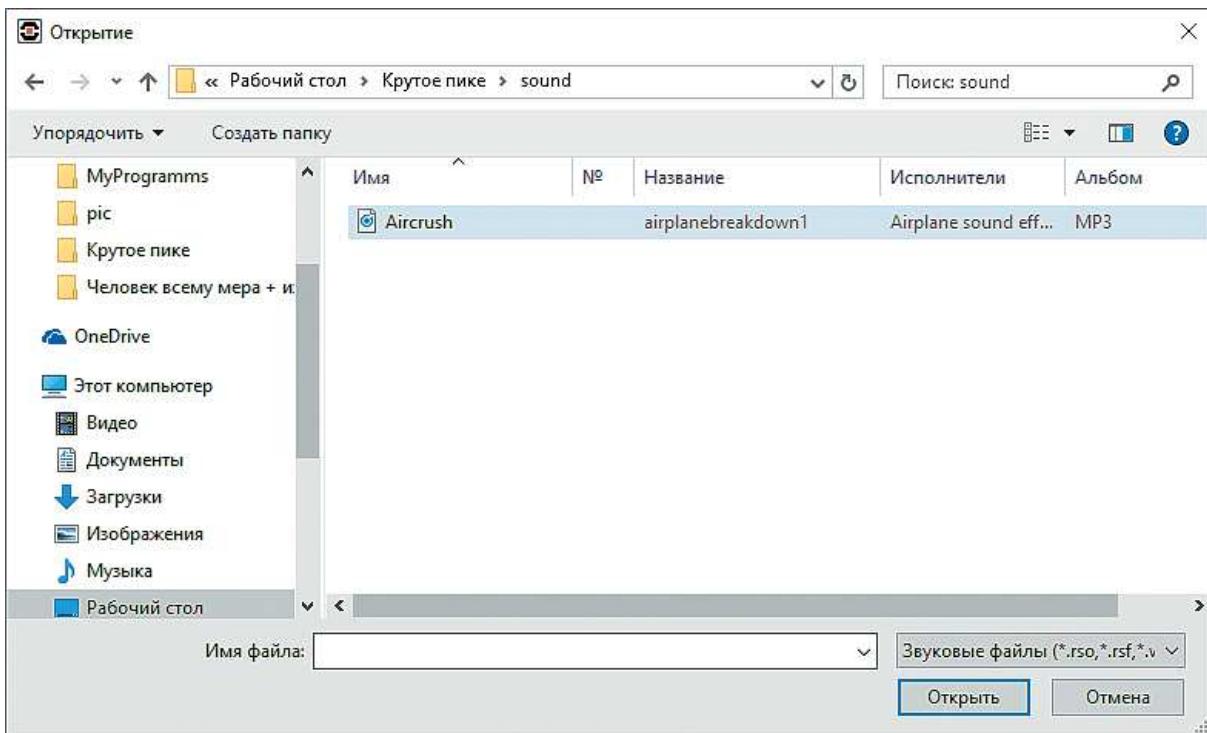
18. Теперь «научим» твой авиасимулятор воспроизводить звук крушения самолёта. Скачай его на сайте издательства по ссылке <http://pilotz.ru/files/10045/>, выбери файл **Aircrush.mp3** и сохрани в памяти компьютера. Затем выбери вверху экрана меню **Инструменты** → **Редактор звука**.



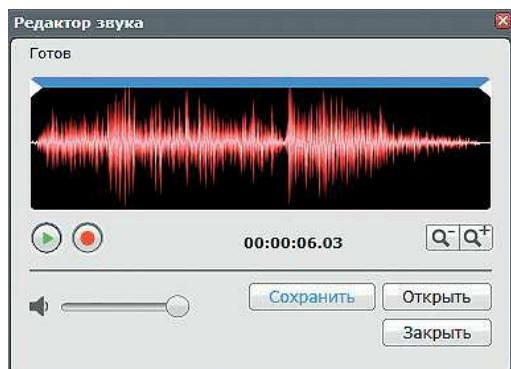
19. Нажми в открывшемся окне кнопку **Открыть**.

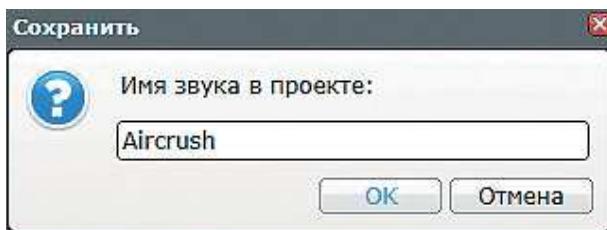


20. Найди в новом окне файл звука **Aircrush** (в переводе с английского — авиакатастрофа) там, где ты сохранил при скачивании, и кликни по нему дважды.

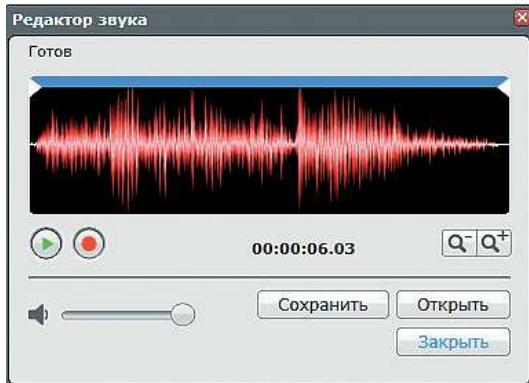


21. Файл сразу загрузится в редактор, где ты можешь его обработать. Данный файл не требует никаких дополнительных действий, поэтому ты можешь его прослушать, нажав клавишу **Play** (зелёный треугольник). Затем нажми кнопку **Сохранить**.

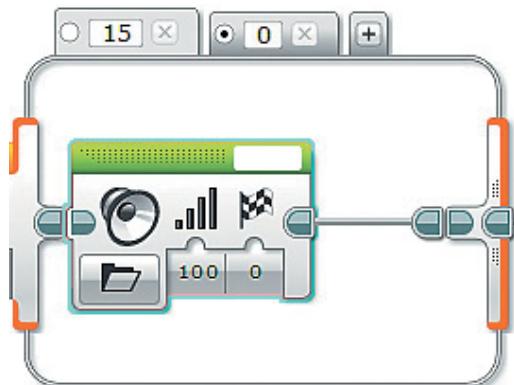




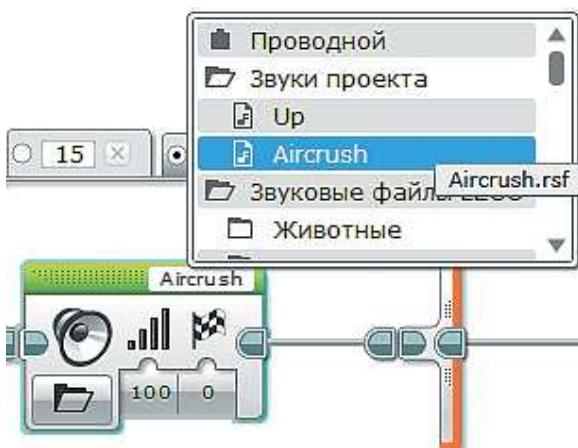
22. Введи в открывшемся окне имя этого звука, например тоже **Air-crush**, и нажми **OK**. По этому имени он будет идентифицироваться в твоём проекте.



23. Теперь нажми кнопку **Закрыть**.



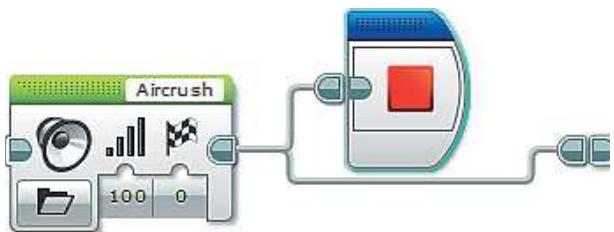
24. Перейди в **Переключателе** на вкладку с числом **15** и добавь туда команду **Звук** (зелёный блок).



25. Выбери в правом верхнем окошке этой команды папку **Звуки проекта** и в ней файл **Aircrush**.

26. После воспроизведения этого звука авиасимулятор должен прекратить свою работу, то есть программа должна полностью завершиться.

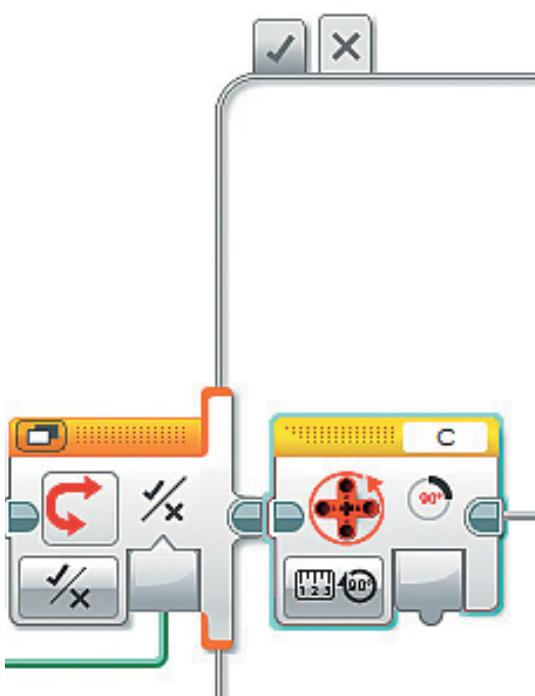
Для этого добавь команду **Остановить программу** (синий блок), только расположи её так, чтобы она не присоединилась к команде **Звук**. Её надо соединить с командой **Звук** при помощи шлейфа — это нужно, чтобы обеспечить возможность повтора тревоги в случае, когда самолёт выведен в нормальное положение после неё.



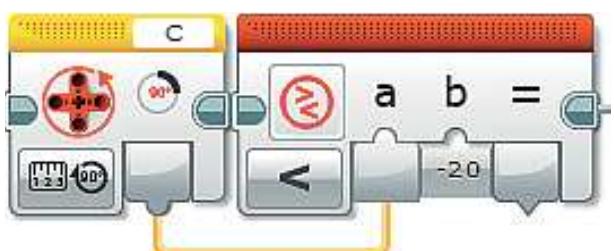
27. Итак, ты создал предупреждающую сигнализацию для случая, когда нос самолёта слишком опущен, то есть первый **Переключатель** в этой подпрограмме принял значение **Истина**.

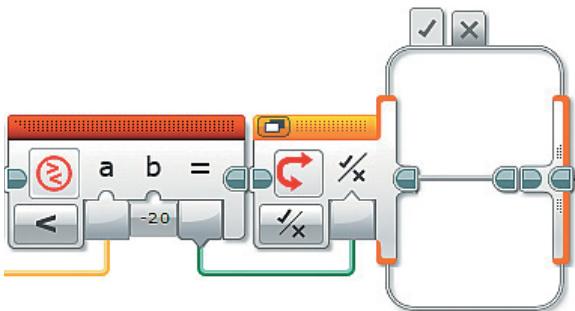
Теперь рассмотрим случай, когда нос слишком сильно поднят. Как мы договаривались ранее, критическим значением будем считать **20** градусов.

Перейди на вкладку **Ложь** внешнего **Переключателя** и добавь команду **Вращение мотора** (жёлтый блок). Проверь, что в верхнем правом окошке выбран порт **C**.



28. Добавь команду **Сравнение** (красный блок), выбери опцию **Меньше** и соедини выход команды **Вращение мотора** с входом **a** команды **Сравнение**. Введи во вход **b** с клавиатуры число **-20**.

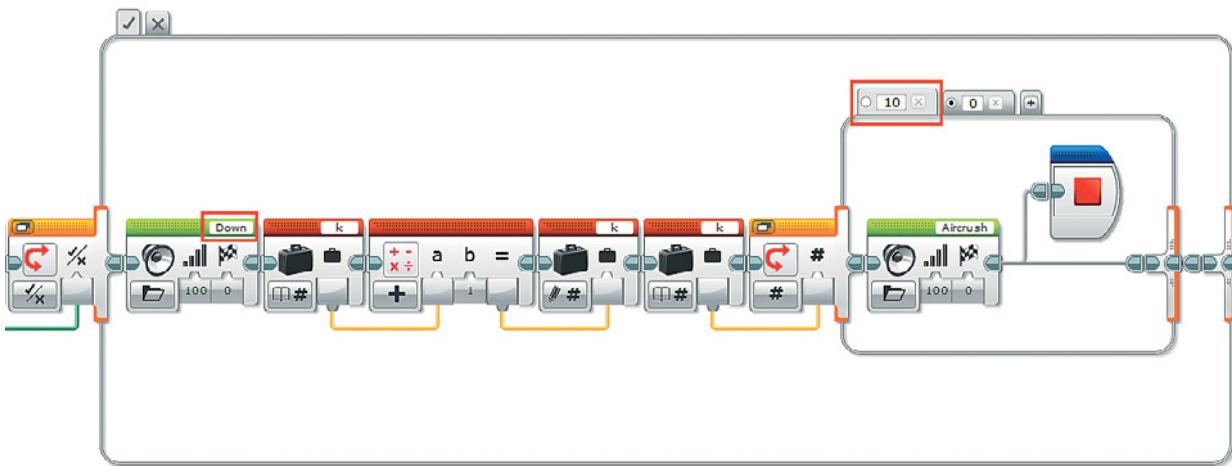




29. Далее добавь команду **Переключатель** (оранжевый блок), перейди к **Виду с вкладками**, выбери опцию **Логическое значение** и соедини выход `==>` команды **Сравнение** с входом **Переключателя**.

30. Напиши такой же фрагмент программы, как и для предыдущего случая, но с небольшими изменениями:

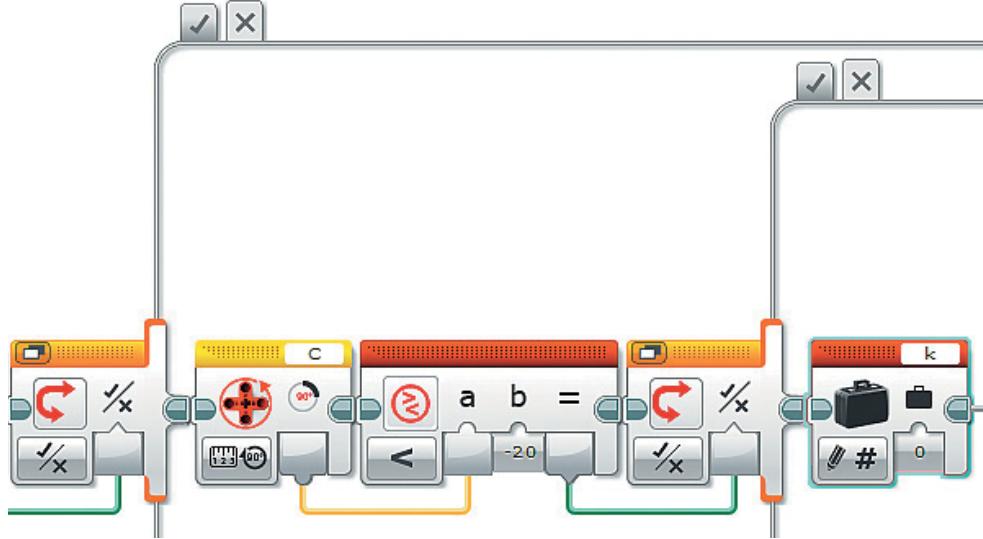
- звуком сигнализации будет **Down**; он находится в папке **Звуковые файлы LEGO → Информация → Down**;
- числовое значение для второго переключателя уже не 15, а **10**, так как слово **Down** звучит дольше, чем **Up**, значит, количество его повторов должно быть меньше.



Осталось рассмотреть третий случай, когда самолёт находится в нормальном состоянии или пилот выровнял его.

31. Перейди в подпрограмме во втором по счёту **Переключателе** на вкладку **Ложь** и добавь команду **Переменная** (красный блок). Для обработки выбери переменную **k**, остальные параметры оставь без изменений.

Пояснение. Если самолёт находится в нормальном состоянии или выведен в него, то счётчик сигналов бедствия должен быть обнулён. Именно это ты и реализовал.



Часть 5. Бесконечность — не предел! Работа одометра, расчёт пройденного расстояния

Логика. В этой части ты обеспечишь корректную работу прибора под названием **одометр** (от греческого ὁδός — дорога и μέτρον — мера), который будет показывать пройденный самолётом путь. Без такого прибора невозможно определить, сколько километров пролетел самолёт или пробежал по взлётно-посадочной полосе, а также оценить запасы топлива.

Ты уже, конечно, знаешь, что пройденное расстояние — это *скорость движения, умноженная на затраченное в пути время*. Однако эта формула (которую мы рассматривали в пункте 24 первой части) действительна только для какого-то конечного и конкретного значения времени. Одометр же должен рассчитывать путь, пройденный за какое-то время, и наращивать (то есть прибавлять) это значение к предыдущему, начиная с нуля.

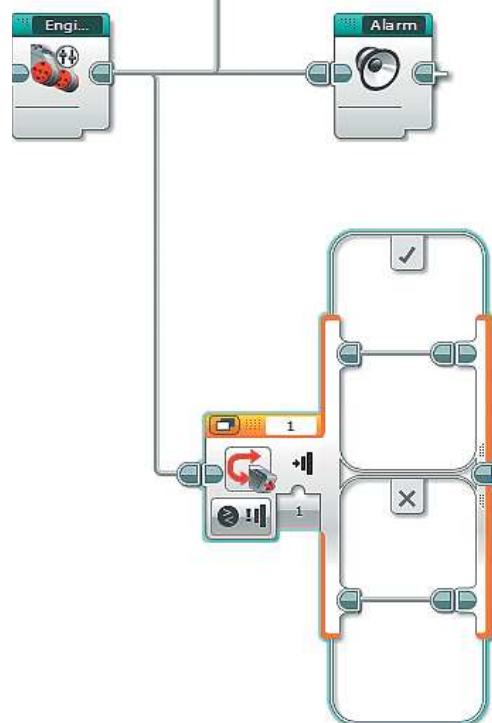
Давай выберем таким промежутком времени 1 секунду, то есть показания одометра на экране EV3 будут обновляться раз в секунду и расти в зависимости от скорости.

Пройденное расстояние S_2 на момент отсчёта определяется по формуле:

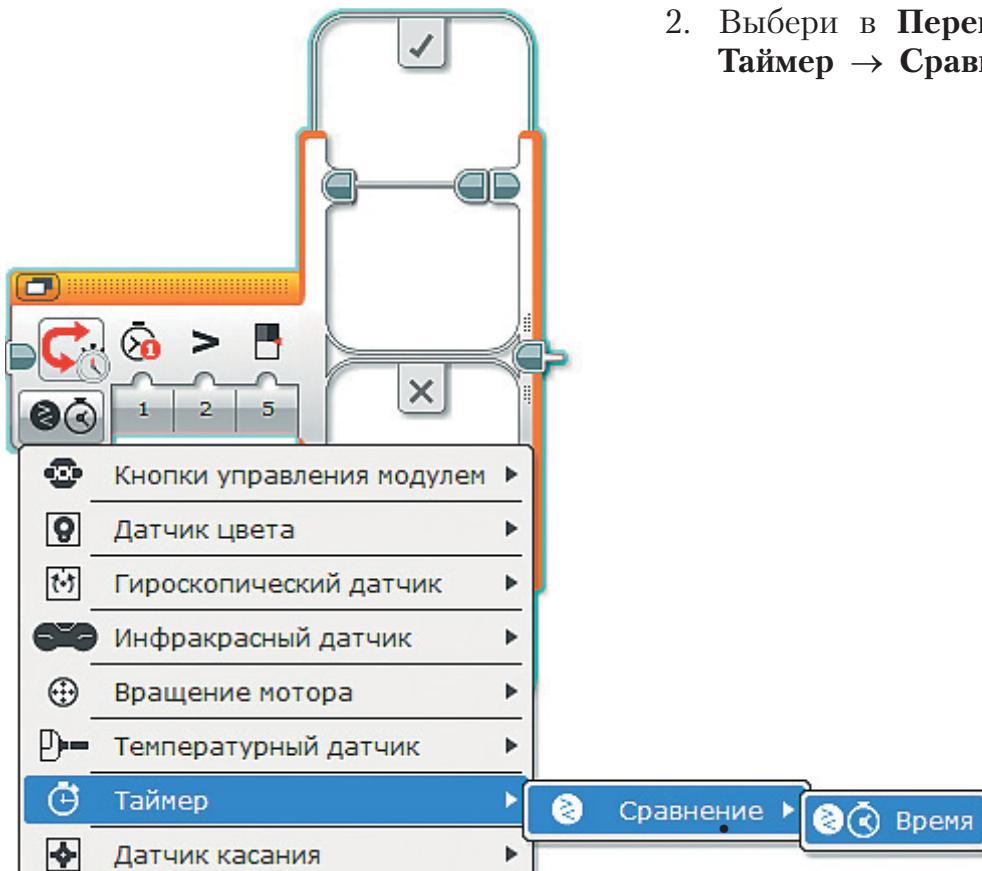
$$S_2 = S_1 + (V \cdot T),$$

где S_1 — значение расстояния, пройденного за предыдущее время; V — скорость в определённый момент времени; T — время обновления информации на экране, равное 0,00027 часа при скорости, не равной нулю (это значение ты рассчитывал тоже в пункте 24 первой части).

Приступай!



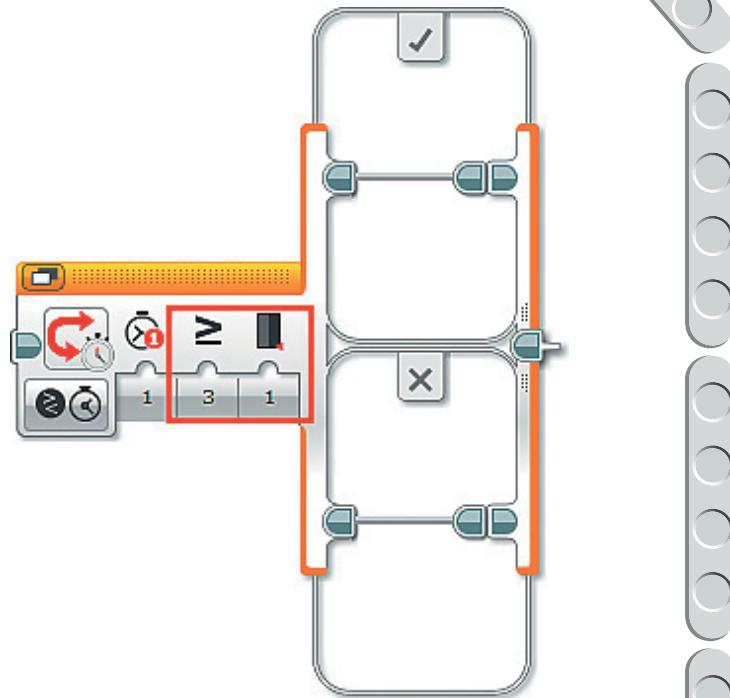
1. Вернись в основную программу **Main**. Тебе осталось построить последнюю, третью ветку в этой программе. Добавь команду **Переключатель** (оранжевый блок) и расположи её под командой **Alarm**, а затем соедини её с помощью шлейфа от команды **Engines**.



2. Выбери в **Переключателе** опцию **Таймер** → **Сравнение** → **Время**.

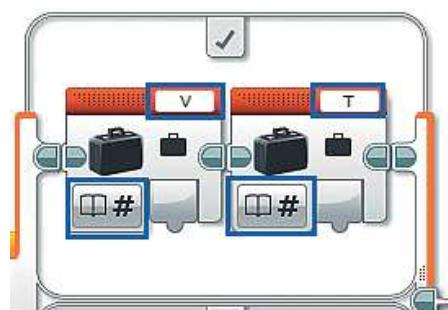
3. Этот **Переключатель** заработает сразу, как только в одном из **таймеров** (например, в первом) будет превышаться значение 1 секунда. Для этого настрой параметры **Переключателя**:

- **Тип сравнения** — больше или равно (3);
- **Пороговое значение** — 1 (введи с клавиатуры).



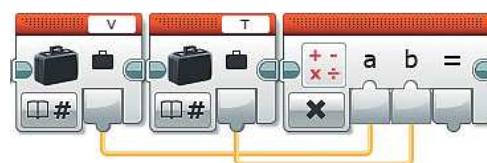
4. Нас **не интересует** интервал времени от 0 до 1 секунды, поэтому весь фрагмент программы для одометра будет внутри вкладки **Истина**. Реализуем формулу подсчёта пройденного расстояния:

$$S_2 = S_1 + (V \cdot T).$$



Добавь две команды **Переменная** (**красный блок**), выбери в них опцию **Считывание** → **Числовое значение**. В первой команде для обработки установи переменную **V**, а во второй — **T**.

5. Добавь команду **Математика** (**красный блок**) и выбери опцию **Умножить**. Соедини выход переменной **V** с входом **a**, выход переменной **T** с входом **b**.



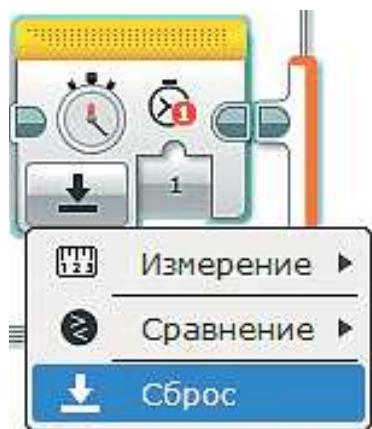
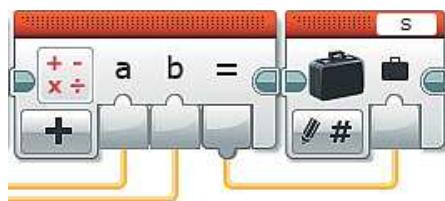
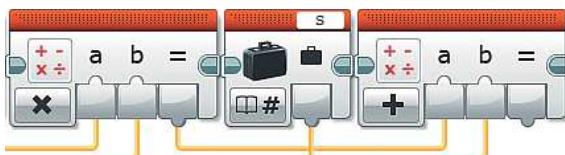
6. Теперь рассчитанное значение расстояния, пройденного за 1 секунду (0,00027 часа), нужно прибавить к предыдущему значению **S**.

Добавь команду **Переменная** (красный блок). Для обработки установи переменную **S** и выбери опцию **Считывание** → **Числовое значение**.

7. Добавь команду **Математика** (красный блок), опцию оставь без изменений, соедини выход « $=$ » предыдущей команды **Математика** с входом **a**, выход переменной **S** с входом **b**.

8. Осталось записать результаты подсчёта в переменную **S**. Добавь команду **Переменная** (красный блок) и выбери для обработки переменную **S**, затем соедини выход « $=$ » команды **Математика** с входом переменной **S**.

9. После этого остаётся только сбросить значение таймера № 1, используемого для одометра. Добавь команду **Таймер** (жёлтый блок), по умолчанию в ней выбран таймер № 1. Установи опцию **Сброс**.



Так держать! Теперь ты всегда будешь знать, какое расстояние преодолел твой самолёт!



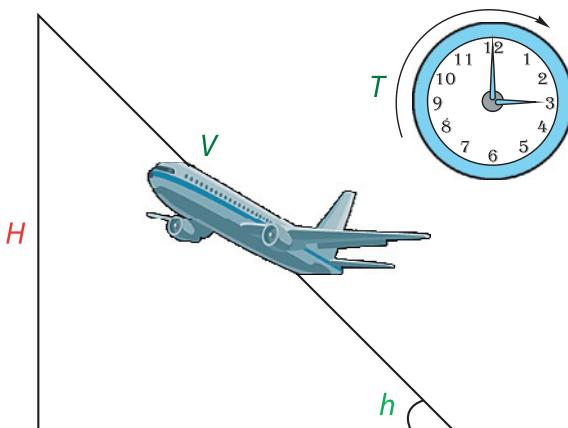
Часть 6. Всё выше, и выше, и выше! Работа альтиметра, расчёт набранной высоты

Логика. В этой части ты обеспечишь работу одного из наиболее сложных приборов — **альтиметра** (от латинского *altus* — высоко). Вообще для расчёта высоты нужны знания такого раздела математики, как тригонометрия, который изучают старшеклассники. Но учёные придумали методы преобразования различных сложных формул в более лёгкие и понятные. Воспользуемся ими.

Ты уже с уверенностью можешь сказать, какие параметры полёта тебе известны.

Для расчёта высоты тебе понадобятся:

- скорость V в расчётный момент времени;
- конкретный момент времени $T = 0,00027$ часа;
- угол тангажа (показания мотора, подключённого к порту **C**).



Как и для других приборов, ты будешь рассчитывать приращение высоты за определённый промежуток времени (1 секунда = 0,00027 часа) и прибавлять это значение к общему значению высоты.

Набранная высота H_2 на момент отсчёта определяется по формуле

$$H_2 = H_1 + \left(\frac{(h^2 + 3) \cdot h \cdot T \cdot V}{3} \cdot 1000 \right),$$

где H_1 — высота, набранная за предыдущее время;

h — угол тангажа, переведённый из градусной меры в радианную (дело в том, что градусы плохо пригодны для подобных «плоских» подсчётов, поэтому существует другая мера — радианы);

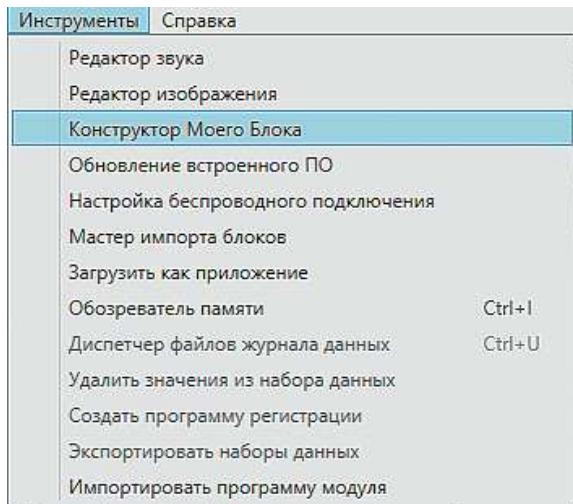
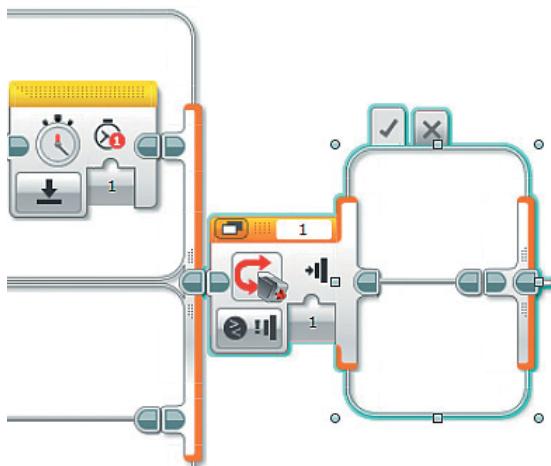
T — фиксированное время;

V — скорость в определённый момент времени.

Результат расчётов по этой формуле получится в километрах, а вы-
соту удобнее и правильнее представлять в метрах, поэтому в конце
формулы стоит множитель 1000.

Часть формулы в больших скобках и есть приращение высоты.

Такой сложный процесс целесообразно вынести в отдельную **подпрограмму**.

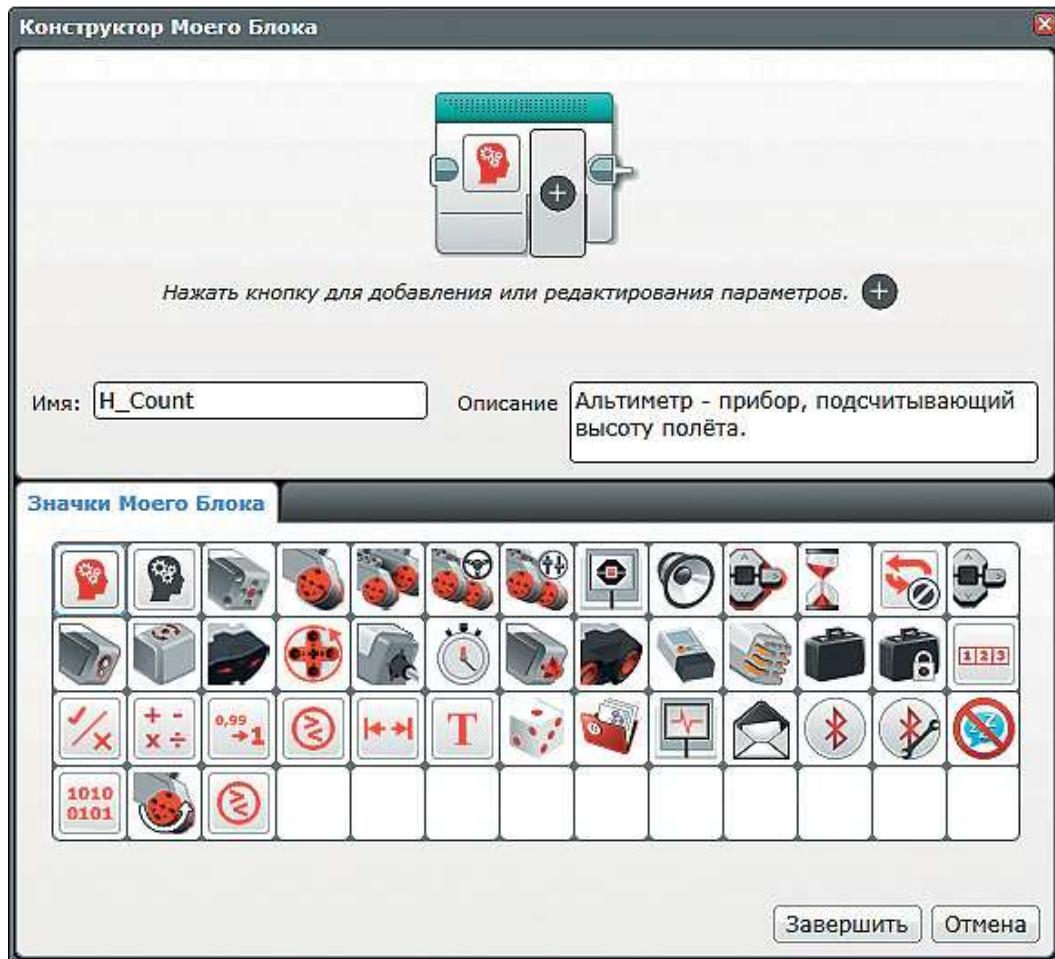


1. Продолжи в основной программе **Main** третью, нижнюю, ветку. Добавь после **Переключателя** ещё один **Переключатель** (оранжевый блок) и перейди к **Виду с вкладками**.

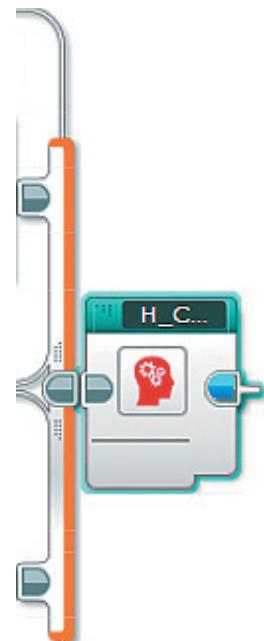
2. Теперь начинай создавать подпрограмму. Команда **Переключатель** должна быть выделена (подсве-
ченена голубым контуром). Выбери в верхней части программы меню **Инструменты** → **Конструктор
Моего Блока**.

3. Задай в открывшемся окне имя подпрограммы, введи её краткое описание и выбери подходящую пиктограмму.

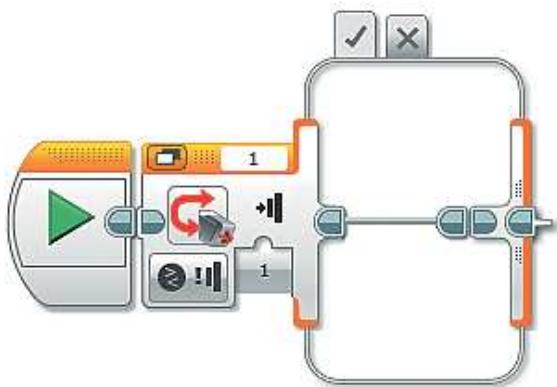
Имя логично задать, например, **H_Count** (от английского *height* –
высота; *count* – подсчёт). Затем нажми кнопку **Завершить**. Краткое описание и пиктограмму можешь задать, как на рисунке, или само-
стоятельно.



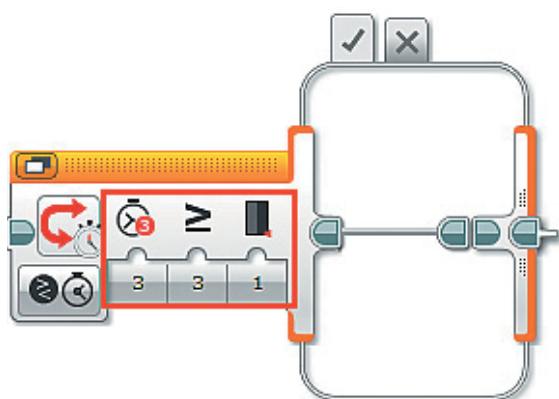
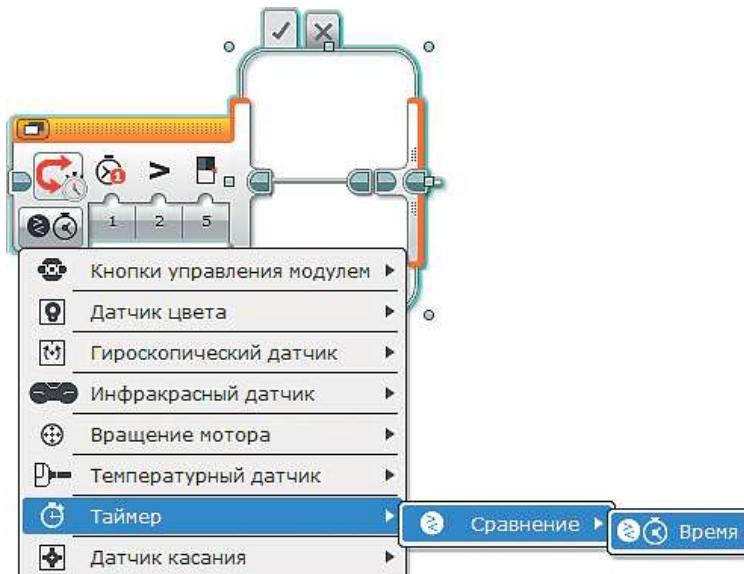
4. В основной программе появится новая команда **H_Count**. Именно в ней будет происходить подсчёт показаний альтиметра.



5. Дважды кликни по ней. На экране откроется новое рабочее поле для редактирования подпрограммы. Сверху появится вкладка с именем **H_Count**.



6. Выбери для **Переключателя** опцию **Таймер** → **Сравнение** → **Время**.



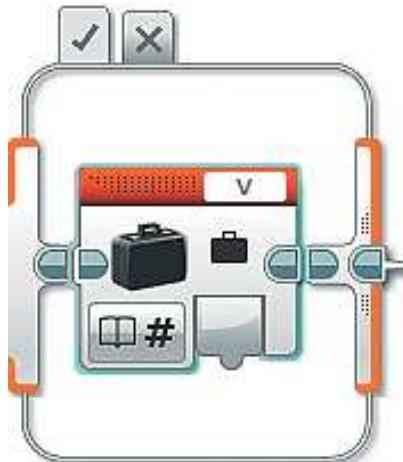
7. Настрой параметры:

- Идентификатор таймера** – 3;
- Тип сравнения** – больше или равно (3);
- Пороговое значение** – 1 (введи с клавиатуры).

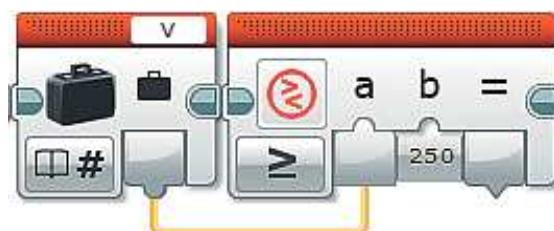
8. Такую конструкцию ты уже реализовывал в предыдущей части — она обеспечивает частоту обновления информации 1 раз в секунду. Поэтому в **Переключателе** нас будет интересовать только вкладка **Истина**.

Будет правильно обеспечить минимальные скорости взлёта и посадки. В среднем, в зависимости от модели самолёта, она составляет 250 км/ч, поэтому введём ограничение на возможность изменения высоты при скорости меньше 250 км/ч.

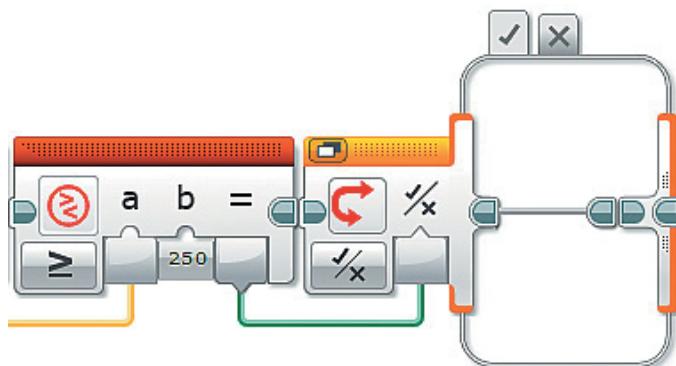
Добавь команду **Переменная** (**красный блок**), для обработки установи переменную **V**, выбери опцию **Считывание** → **Числовое значение**.

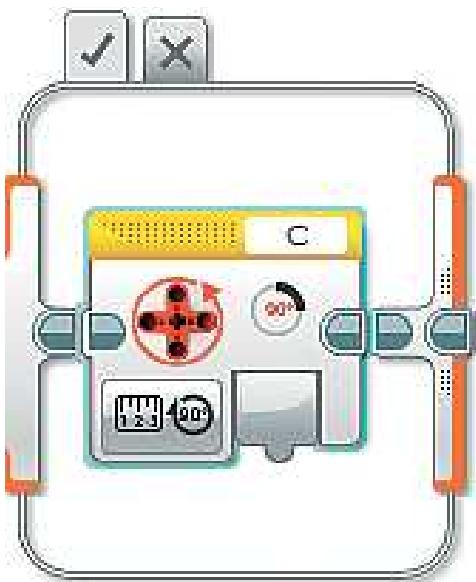


9. Далее добавь команду **Сравнение** (**красный блок**), выбери опцию **Больше или равно**, затем соедини выход переменной **V** с входом **a** команды **Сравнение**, а во вход **b** введи с клавиатуры число **250**.

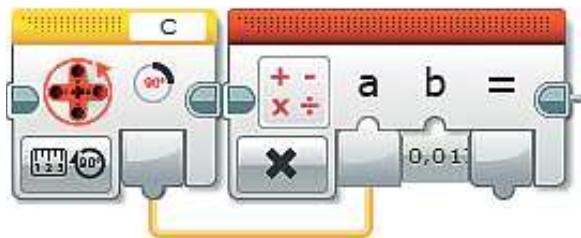


10. Теперь добавь команду **Переключатель** (**оранжевый блок**), перейди к **Виду с вкладками**, выбери опцию **Логическое значение** и соедини выход «**=**» команды **Сравнение** с входом **Переключателя**.



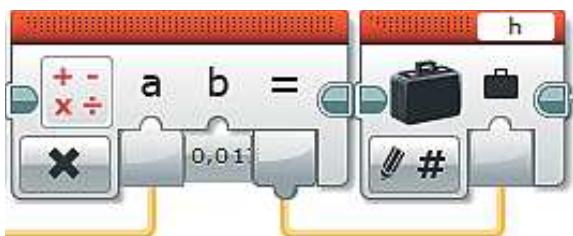


11. Для начала поработай с вкладкой **Истина**. В ней нужно реализовать формулу для расчёта высоты, поэтому приготовься к большому объёму математических действий. Добавь команду **Вращение мотора** (жёлтый блок). Проверь, чтобы в верхнем правом окошке значился порт **C**.



12. Далее добавь команду **Математика** (красный блок) и выбери опцию **Умножить**. Затем соедини выход команды **Вращение мотора** с входом **a**, во вход **b** с клавиатурой введи число **0,0175**.

Пояснение. Именно это действие позволяет перевести градусную меру угла в радианную.

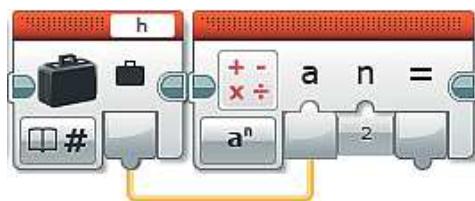


13. Добавь команду **Переменная** (красный блок). Для обработки выбери переменную **h** и соедини выход «**=**» команды **Математика** с входом переменной **h**.

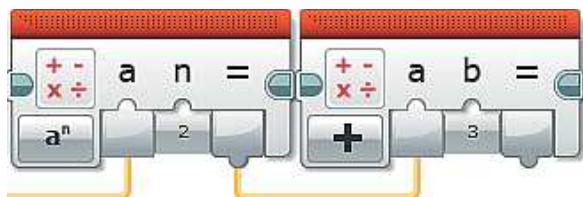


14. Согласно формуле, тебе нужно возвести полученное значение в квадрат. Добавь команду **Переменная** (красный блок), установи для обработки переменную **h**, выбери опцию **Считывание → Числовое значение**.

15. Добавь команду **Математика** (красный блок), выбери опцию **Показатель степени** и соедини выход переменной **h** с входом **a**, во вход **n** с клавиатуры введи **2**.



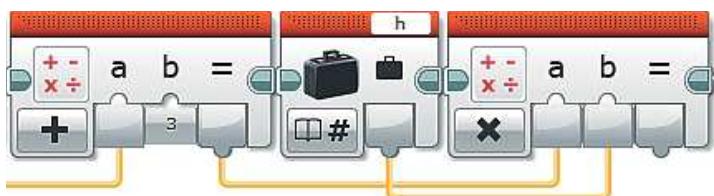
16. К этому значению нужно прибавить 3. Добавь команду **Математика** (красный блок) и соедини выход «**=**» предыдущей команды с входом **a** новой, во вход **b** введи с клавиатуры число **3**.



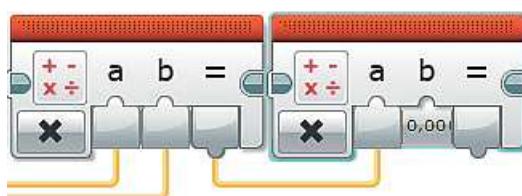
17. Полученный результат нужно умножить на **h**. Добавь команду **Переменная** (красный блок). Установи для обработки переменную **h** и выбери опцию **Считывание** → → **Числовое значение**.



18. Далее добавь команду **Математика** (красный блок) и выбери опцию **Умножить**. Соедини выход «**=**» предыдущей команды **Математика** с входом **a**, а выход переменной **h** с входом **b**.



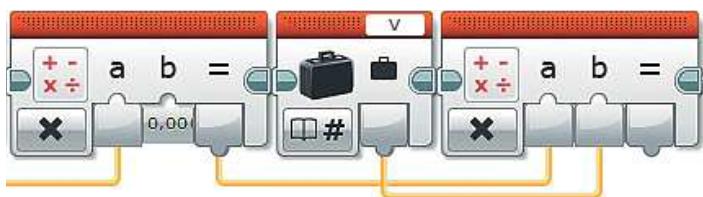
19. Теперь полученное значение умножим на фиксированное время. Добавь команду **Математика**, выбери опцию **Умножить** и соедини выход «**=**» предыдущей команды **Математика** с входом **a**, введи во вход **b** с клавиатуры число **0,00027**.



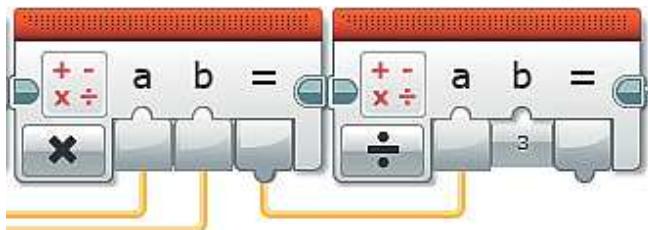


20. Согласно формуле, в числителе остаётся умножить результат на скорость V . Добавь команду **Переменная** (красный блок), установи для обработки переменную **V** и выбери опцию **Считывание** → → **Числовое значение**.

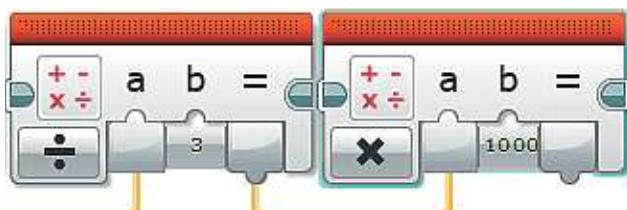
21. Добавь команду **Математика** (красный блок). Выбери опцию **Умножить**, а затем соедини выход $\ll=$ предыдущей команды **Математика** с входом **a**, выход переменной **V** с входом **b**.



22. Теперь результат всех вычислений нужно разделить на 3. Добавь команду **Математика** (красный блок). Выбери опцию **Разделить** и соедини выход $\ll=$ предыдущей команды **Математика** с входом **a**, введи во вход **b** с клавиатуры число **3**.

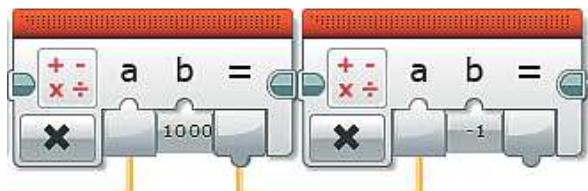


23. Для перевода километров в метры результат нужно умножить на 1000. Добавь команду **Математика** (красный блок). Выбери опцию **Умножить** и соедини выход $\ll=$ предыдущей команды **Математика** с входом **a**, введи во вход **b** с клавиатуры число **1000**.



24. Подсчёты практически завершены. Последний шаг, который тебе надо сделать, — скорректировать вычисление, умножив результат на -1 . Это связано, как ты помнишь, с расположением вертикального мотора. Таким образом ты произведёшь **инвертирование** (одна из математических хитростей) результатов подсчёта. Тогда значение высоты будет положительным.

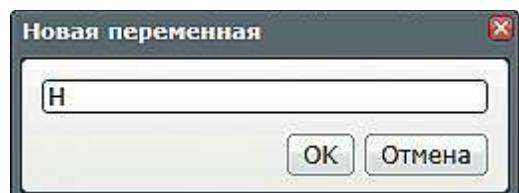
Добавь команду **Математика** (красный блок). Выбери опцию **Умножить**, затем соедини выход « $=$ » предыдущей команды **Математика** с входом **a**, а во вход **b** с клавиатуры введи -1 .



25. Рассчитанное мгновенное значение высоты нужно прибавить (то есть прирастить) к основной высоте. Для неё ещё не объявлена переменная, поэтому добавь команду **Переменная** (красный блок) и выбери в правом верхнем окошке пункт **Добавить переменную**.

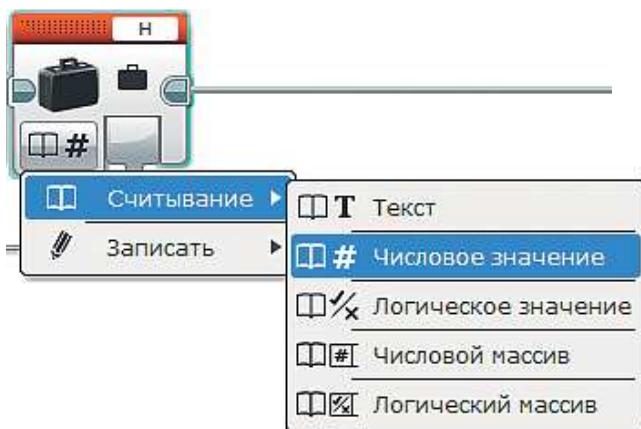


26. Введи в открывшемся окне с клавиатуры **H** и нажми **OK**.

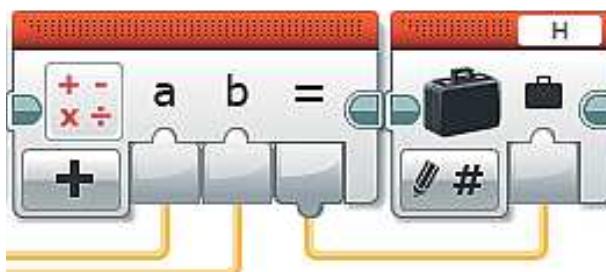
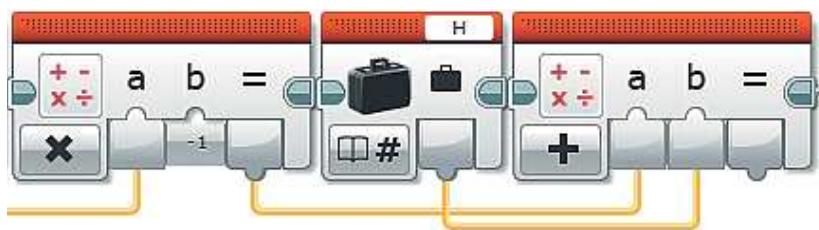


Пояснение. Прописные и строчные буквы для обозначения переменных — это разные буквы, и их можно и нужно использовать как отдельные объекты.

27. Выбери в команде **Переменная** опцию **Считывание** → **Числовое значение**.



28. Далее добавь команду **Математика** (красный блок) и соедини выход « $=$ » предыдущей команды **Математика** с входом **a**, выход переменной **H** с входом **b**.

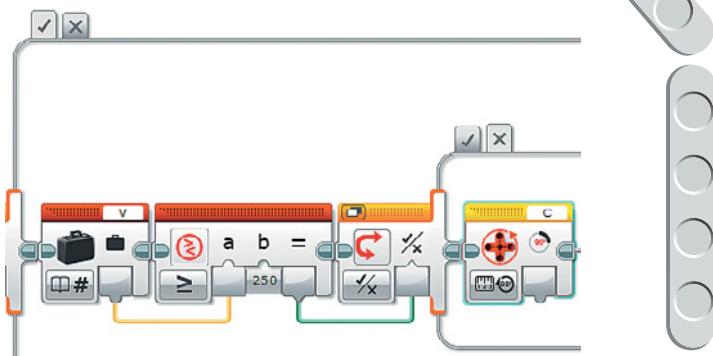


29. Запиши результат в блок **H**. Для этого добавь команду **Переменная** (красный блок), установи для обработки переменную **H** и соедини выход « $=$ » команды **Математика** с входом переменной **H**.

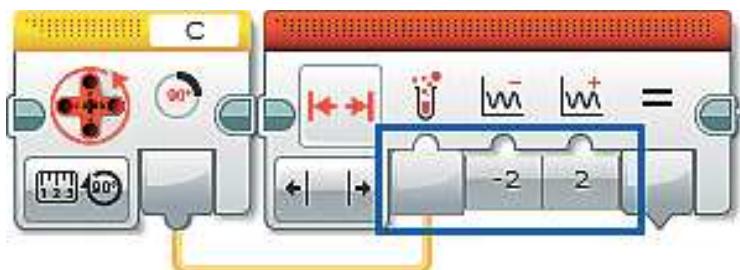


30. Осталось сбросить значение таймера. Добавь команду **Таймер** (жёлтый блок), выбери опцию **Сброс** и установи Идентификатор таймера: 3.

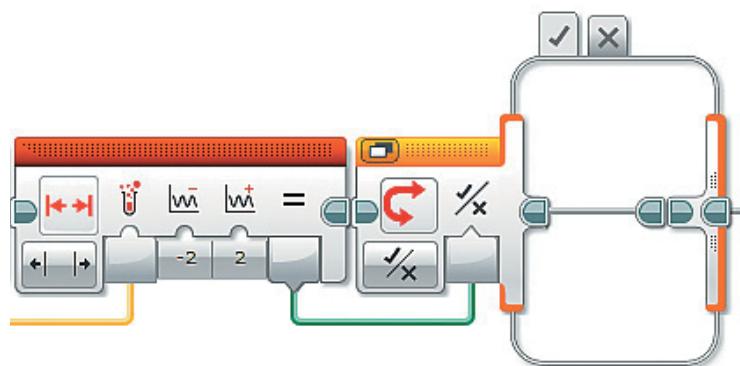
31. Вернёмся к случаю, когда условие «скорость превысила 250 км/ч» **не выполнено**. Перейди к вкладке **Ложь** (крестик наверху команды) внутреннего **Переключателя** и добавь туда команду **Вращение мотора** (жёлтый блок). Проверь, чтобы в верхнем правом окошке значился порт **C**.



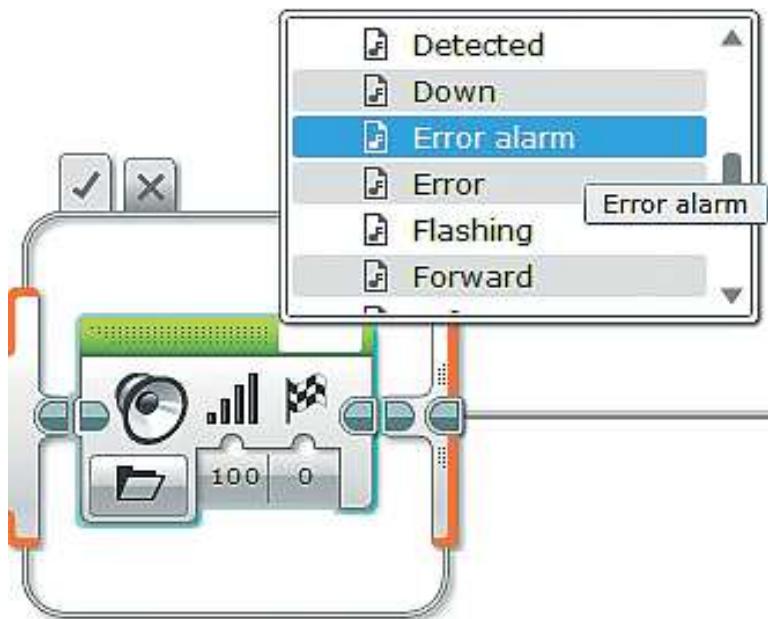
32. При разбеге самолёта штурвал может незначительно отклоняться, такой случай нужно обязательно предусмотреть. Пусть допустимым отклонением штурвала будет интервал от -2 до 2 градусов. Добавь команду **Интервал** (красный блок) и выбери опцию **За пределами**. Далее соедини выход команды **Вращение мотора** с входом **Тестовое значение**. Введи во входы **Нижняя граница** и **Верхняя граница** с клавиатуры соответственно -2 и 2 .



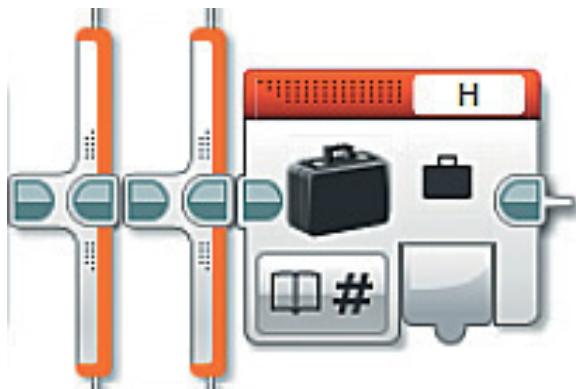
33. Теперь, если допустимое отклонение действительно превышено, необходимо подать сигнал тревоги. Добавь команду **Переключатель** (оранжевый блок). Перейди к **Виду с вкладками**, выбери опцию **Логическое значение** и соедини выход « $=$ » команды **Интервал** с входом **Переключателя**.



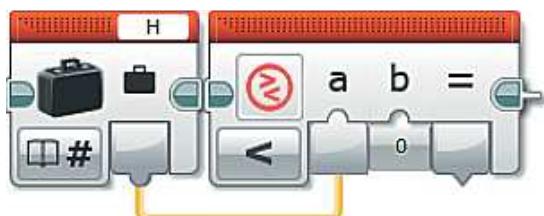
34. Тебя будет интересовать только вкладка **Истина**. Добавь в неё команду **Звук** (зелёный блок) и выбери в правом верхнем окошке в меню папку **Звуковые файлы LEGO**, в ней папку **Информация** и наконец файл **Error alarm**.



Осталось предусмотреть тот случай, когда значение высоты попытается «убежать» в отрицательную сторону (самолёт же не может улететь под землю!).

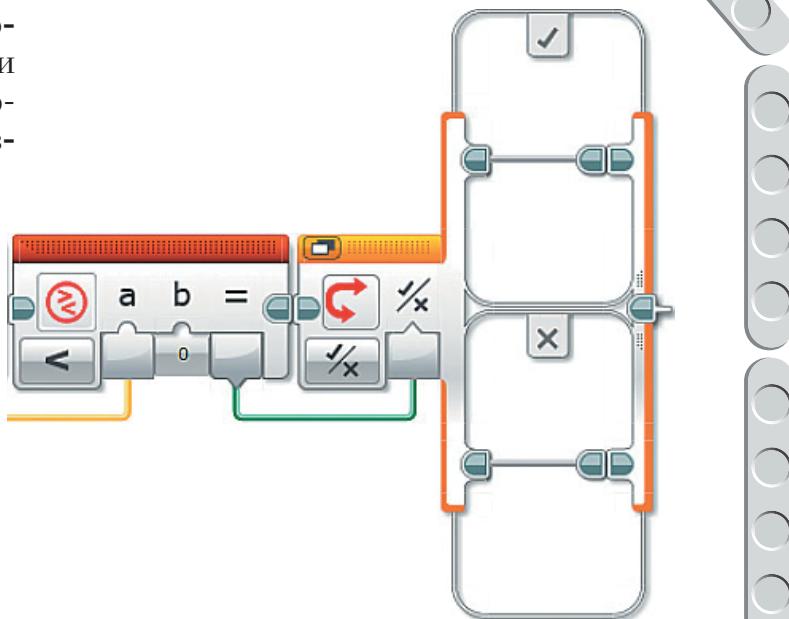


35. Добавь в конец этой подпрограммы после внешнего **Переключателя** команду **Переменная** (красный блок), установи для обработки переменную **H**, выбери опцию **Считывание** → **Числовое значение**.

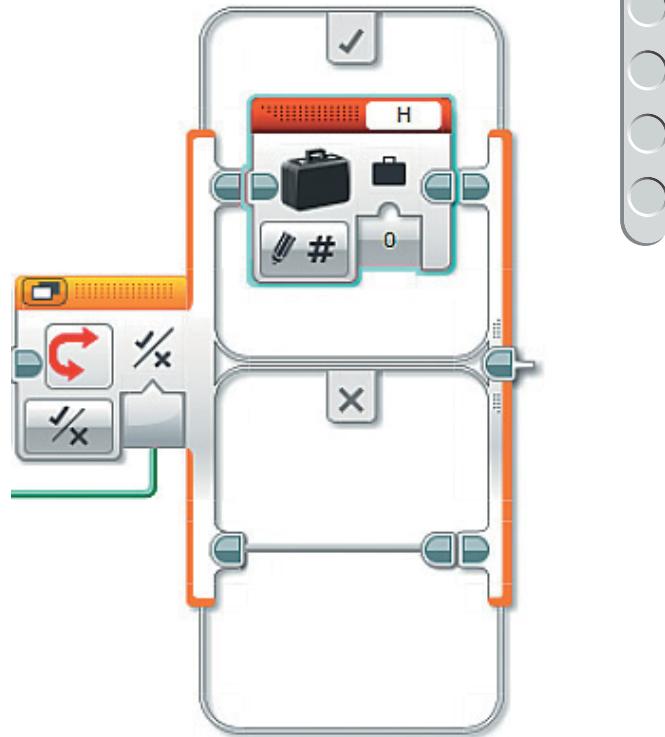


36. Далее добавь команду **Сравнение** (красный блок), выбери опцию **Меньше** и соедини выход переменной **H** с входом **a**, введи во вход **b** с клавиатуры число **0**.

37. Теперь добавь команду **Переключатель** (оранжевый блок), выбери опцию **Логическое значение** и соедини выход **«=»** команды **Сравнение** с входом **Переключателя**.



38. Добавь во вкладку **Истина** команду **Переменная** (красный блок) и установи для обработки переменную **H**. Опцию и входное значение менять не нужно.



Пояснение. К такой хитрости ты уже прибегал, когда настраивал подобный запрет для скорости — если она попытается «убежать» ниже нуля, то программа «вернёт» её значение к нулю.

Поздравляем! Самое сложное позади!

Часть 7. Последний рывок. Вывод на экран показаний всех приборов (завершение)

Ты большой молодец! Ты проделал огромный путь к тому, чтобы твой авиасимулятор заработал в полную силу! Самое сложное позади, осталось сделать последние штрихи и вывести на экран показания всех приборов.

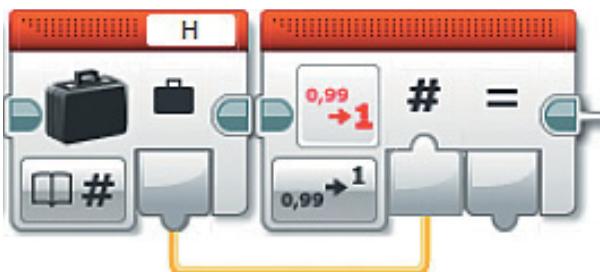
Некоторые показания приборов нужно будет округлить до целых значений, обеспечить корректный вывод показаний на экран EV3 с правильным расположением и задать частоту обновления информации на экране.

Вперёд!



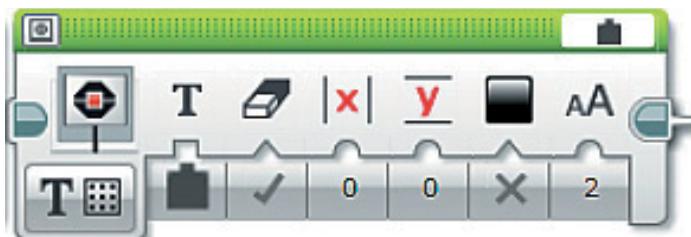
1. Вернись в основную программу **Main** и продолжи составление программы после команды **H_Count**. Добавь команду **Переменная** (красный блок), установи для обработки переменную **Н** и выбери опцию **Считывание** → → **Числовое значение**.

Так как формула для расчёта высоты довольно сложная, то и результат ты получишь далеко не целый. Удобнее будет его округлить.



2. Добавь команду **Округление** (красный блок) и соедини выход переменной **Н** с входом **Ввод** этой команды.

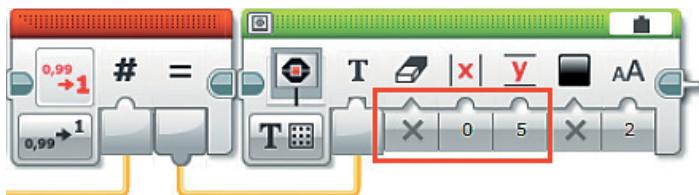
3. Выведи значение высоты на экран EV3. Для этого добавь команду **Экран** (зелёный блок), опцию **Текст** → **Сетка**, выбери в правом верхнем окошке пункт **Проводной**.



4. Теперь соедини выход «=» команды **Округление** с входом **Текст**.

Настрой параметры команды **Текст**:

- **Очистить экран** — ложь;
- **X = 0, Y = 5** (то есть значение будет выводиться у левого края экрана, но ниже первой строки, к середине).

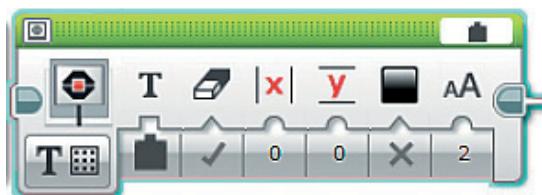


Осталось вывести на экран показания одометра.

5. Добавь команду **Переменная** (красный блок), установи для обработки переменную **S**, выбери опцию **Считывание → Числовое значение**.

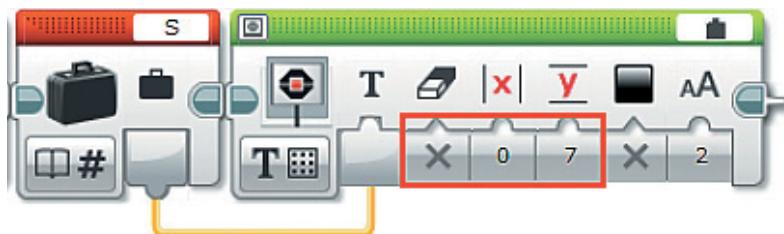


6. Добавь команду **Текст** (зелёный блок), выбери опцию **Текст → Сетка**.



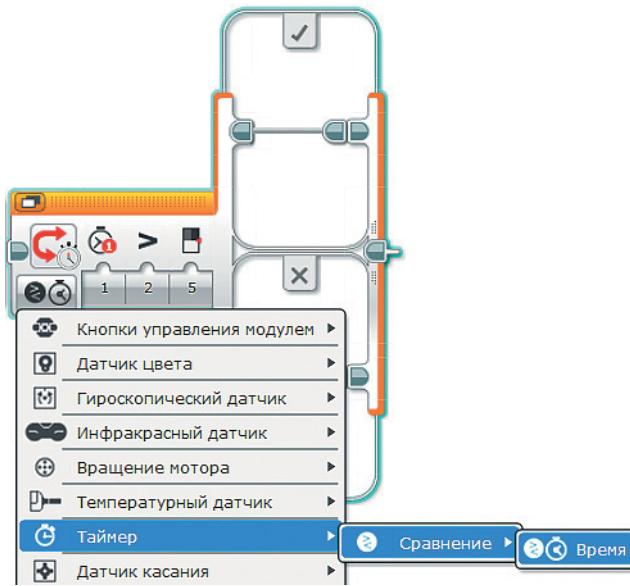
7. Соедини выход переменной **S** с входом **Текст**. Настрой параметры команды **Текст**:

- **Очистить экран** — ложь;
- **X = 0, Y = 7** (то есть значение будет выводиться у левого края экрана, но ниже первой строки, к середине).

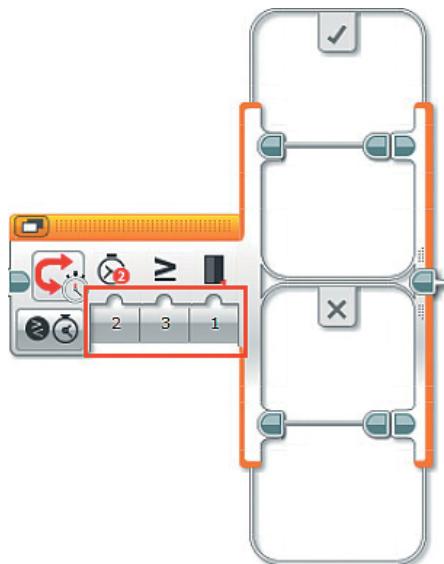


Последнее, что осталось сделать, — это то, о чём мы с тобой много говорили. Нужно обеспечить частоту обновления информации на экране EV3 1 раз в секунду. Сделать это очень просто: с помощью таймера очищать экран 1 раз в секунду.

- Добавь команду **Переключатель** (оранжевый блок) и выбери опцию **Таймер** → **Сравнение** → **Время**.



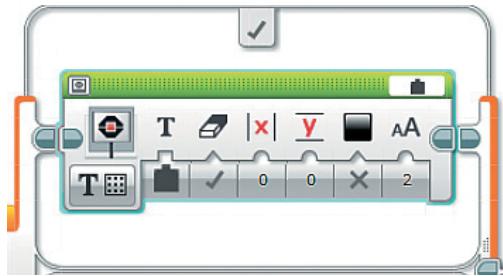
- Настрой параметры команды:
 - Идентификатор таймера** — 2;
 - Тип сравнения** — больше или равно (3);
 - Пороговое значение** — 1 (введи с клавиатуры).



10. Нас интересует только выполненное условие, поэтому во вкладку **Истина** добавь команду **Экран** (зелёный блок), выбери опцию **Текст → Сетка**, а в правом верхнем окошке — пункт **Проводной**. В этой команде никакие параметры настраивать не требуется. Здесь важно, что она ничего не выводит на экран, а параметр **Очистка экрана** — **Истина**.

Внимание!

Так как в твоём проекте содержится несколько программ, при запуске авиасимулятора ты должен загрузить в память EV3 и запустить главную из них — Main.



Теперь нужно сбросить показания таймера № 2.

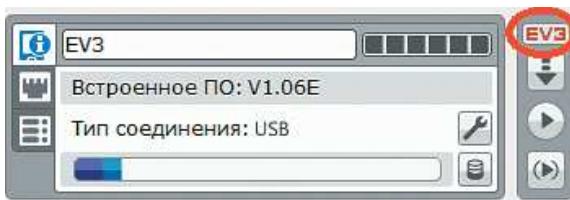
11. Добавь команду **Таймер** (жёлтый блок), выбери опцию **Сброс** и **Идентификатор таймера: 2**.



Этап 5. Загрузка программы и её тестирование

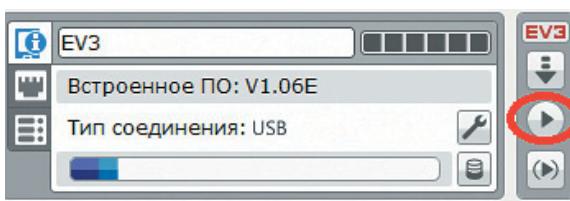


ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ



Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру, соединив порты PC на модуле и USB на компьютере. В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. Перед запуском программы штурвал должен находиться строго в вертикальном положении, то есть, если твой авиасимулятор стоит на столе, вертикальный мотор (управляет углом тангажа) должен быть перпендикулярен столу.
2. Нажми кнопку правого датчика касания — на экране EV3 в правом верхнем углу должно появиться нарастающее значение скорости. То же самое проделай с левым датчиком. Для него, наоборот, значение скорости начнёт уменьшаться. Убедись, что скорость не выходит из диапазона 0–850 км/ч.
3. Наклони штурвал на себя — в верхней части экрана показания угла тангажа должны измениться в отрицательную сторону. При наклоне штурвала от себя показания изменятся в положительную сторону. Проделай эти же действия для крена, наклонив штурвал вправо и затем влево.



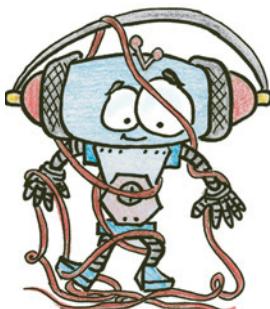
4. При скорости менее 250 км/ч и отклонении угла тангажа в ту или другую сторону более чем на 2 градуса должен подаваться запрещающий сигнал.
5. При скорости более 250 км/ч и превышении критических углов тангажа должны подаваться предупреждающие сигналы **Up** или **Down**.
6. После 15 сигналов **Up** или 10 сигналов **Down** авиасимулятор должен воспроизвести звук крушения **Aircrush** и завершить работу.
7. При скорости более 250 км/ч в зависимости от угла тангажа должны увеличиваться (штурвал на себя) или уменьшаться (штурвал от себя) показания альтиметра — значения высоты полёта.
8. При скорости более 0 км/ч в нижней части экрана EV3 должны изменяться показания одометра: чем выше скорость, тем быстрее растёт значение.

Если ты хочешь повторно запустить программу или продемонстрировать работу своего робота другу, то для его запуска компьютер тебе уже не нужен: включи программируемый блок, затем выбери папку **SteepDive** и в ней программу **Main**.

Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

Этап 6. От винта!



Теперь ты можешь позвать друзей, и вы вместе поиграете в «Экипаж самолёта»!

Роли можно распределить так.

Командир корабля, первый пилот — управляет самолётом при взлёте и дальнейшем полёте, ведёт переговоры с авиадиспетчером.

Второй пилот — управляет самолётом, производит посадку самолёта, ведёт переговоры с авиадиспетчером.

Авиадиспетчер — ведёт переговоры с пилотами, наблюдает за ходом полёта.

Сценарий, по которому вы можете сыграть:

- командир корабля запрашивает у авиадиспетчера подробную сводку о погодных условиях на заданном маршруте в пунктах взлёта и посадки;
- авиадиспетчер сообщает сводку о погодных условиях (температура воздуха, скорости ветра, солнечно/пасмурно/дождь/снег);
- командир корабля запрашивает у авиадиспетчера разрешение на взлёт;
- авиадиспетчер даёт разрешение на взлёт;
- командир информирует экипаж о готовности к взлёту, набирает скорость, плавно тянет штурвал на себя, взлетает, набирая высоту.

Второй пилот в это время следит за показаниями на приборной панели и информирует командира.

Затем:

- командир корабля и второй пилот меняются ролями;
- второй пилот запрашивает у авиадиспетчера разрешение на посадку;
- авиадиспетчер даёт разрешение на посадку, сообщает об атмосферном давлении, скорости и направлении ветра, температуре воздуха в пункте посадки;
- второй пилот сообщает о готовности к посадке, уменьшает скорость, отклоняет штурвал от себя, плавно снижаясь, а затем производит посадку при **безопасной** скорости 300–350 км/ч и угле тангажа не более 2 градусов;



- командир корабля следит за скоростью самолёта и информирует второго пилота о показаниях приборов;
 - авиадиспетчер наблюдает за посадкой самолёта, даёт корректизы.
- В завершение командир корабля сообщает авиадиспетчеру об успешном приземлении самолёта.

Это ещё не всё! Твой авиасимулятор может ещё больше приблизить тебя к романтике настоящего полёта!

Горизонт НЕ завален!

Найди в сети Интернет характеристики одного из популярных пассажирских самолётов (например, Sukhoi Superjet100 или Ту-204) и определи средний показатель допустимого угла крена самолёта — усовершенствуй свою программу так, чтобы авиасимулятор сигнализировал об опасности.

Захожу на второй круг!

Часто случается так, что самолёт не может зайти на посадку с первого раза, и экипажу приходится с помощью авиадиспетчера заходить на повторные круги. Для этого, несомненно, нужно знать, на какой угол выполнил поворот самолёт и каково его отклонение от первоначальной траектории.

Дополни авиасимулятор новым прибором, который будет показывать угол, на который отклонился самолёт. Для расчётов можешь воспользоваться следующей формулой:

$$Q = \frac{r \cdot 0,000046}{V},$$

где Q — угол, на который повернулся самолёт; r — угол крена (в градусах); V — текущая скорость воздушного судна (в км/ч).

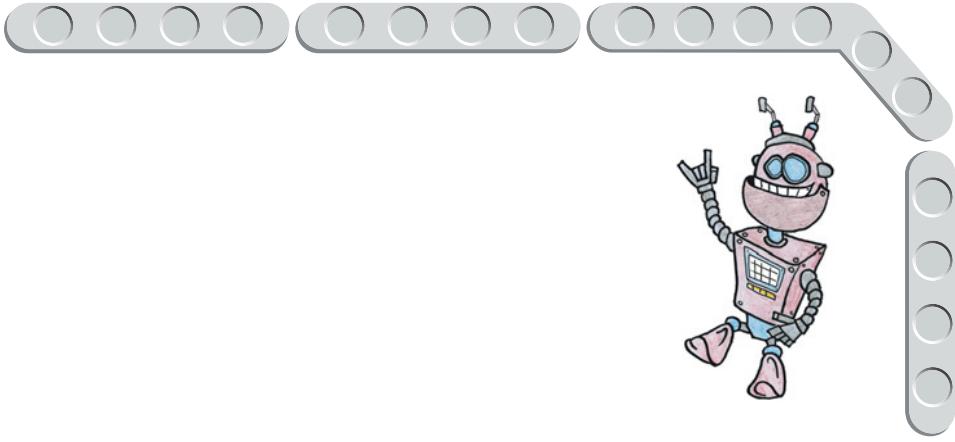
Когда самолёт поворачивает, у него появляется крен в ту или иную сторону. Прибор рассчитывает по формуле, на какой угол самолёт отклонился от прежней траектории, и этот угол нужно «запомнить». Но когда самолёт закончит поворот, значение крена станет равным нулю, и показания прибора тоже станут равными нулю. Вот здесь и нужен авиадиспетчер, который с помощью линейки и транспортира фиксирует на бумаге все значения угла поворота самолёта, то есть его новой траектории.

Внимание!

Расчёт этого значения необходимо делать с периодичностью 1 раз в секунду и каждый раз прибавлять его к предыдущему (суммировать общее значение).

Дозаправка требуется?

Придумай, как реализовать в программе учёт расхода топлива и упра- ждающую сигнализацию об его уровнях, особенно о критическом значе- нии. В нижней части дисплея EV3 ещё осталось место, поэтому можно написать подпрограмму **Fuel**, указывающую заполненность топливных баков (например, в процентах). В зависимости от развиваемой мощно- сти турбин и этапа полёта запасы топлива будут уменьшаться быстрее или медленнее.



А теперь...

И ГУЛ ТУРБИН МЫ СЛЫШИМ ВНОВЬ!

Сделай авиасимулятор более реалистичным — добавь в него звук работающих турбин! Для этого скачай с сайта издательства по ссылке <http://pilotlz.ru/files/10045/> три файла: **Low** (низкий), **Medium** (средний) и **High** (высокий), каждый из которых соответствует разным уровням мощности работы турбин.

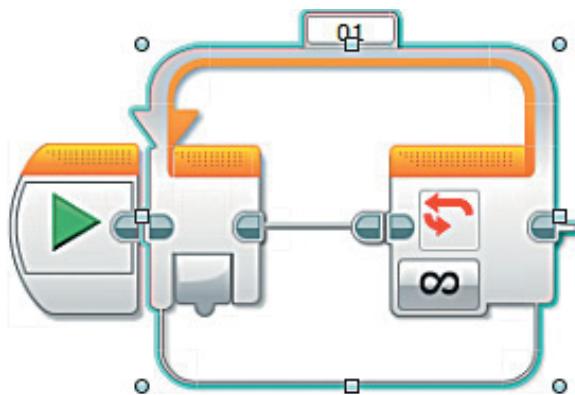
Для примера зададим звучание турбин на низкой мощности.

Снова открой свой проект **SteepDive** в LME-EV3 (**Файл → Открыть проект → Открыть**) и перейди к основной программе **Main**. Найди её начало, оно будет слева (это команда **Начало** с пиктограммой зелёного треугольника). На самом деле вся твоя программа и подпрограммы начинались именно с этой команды, и это не случайно! В программировании есть такой приём, как **параллельное исполнение** двух различных процессов в одной программе. И вот теперь параллельно с работой основной программы ты запустишь воспроизведение аудиоэффектов, имитирующих звук турбин.

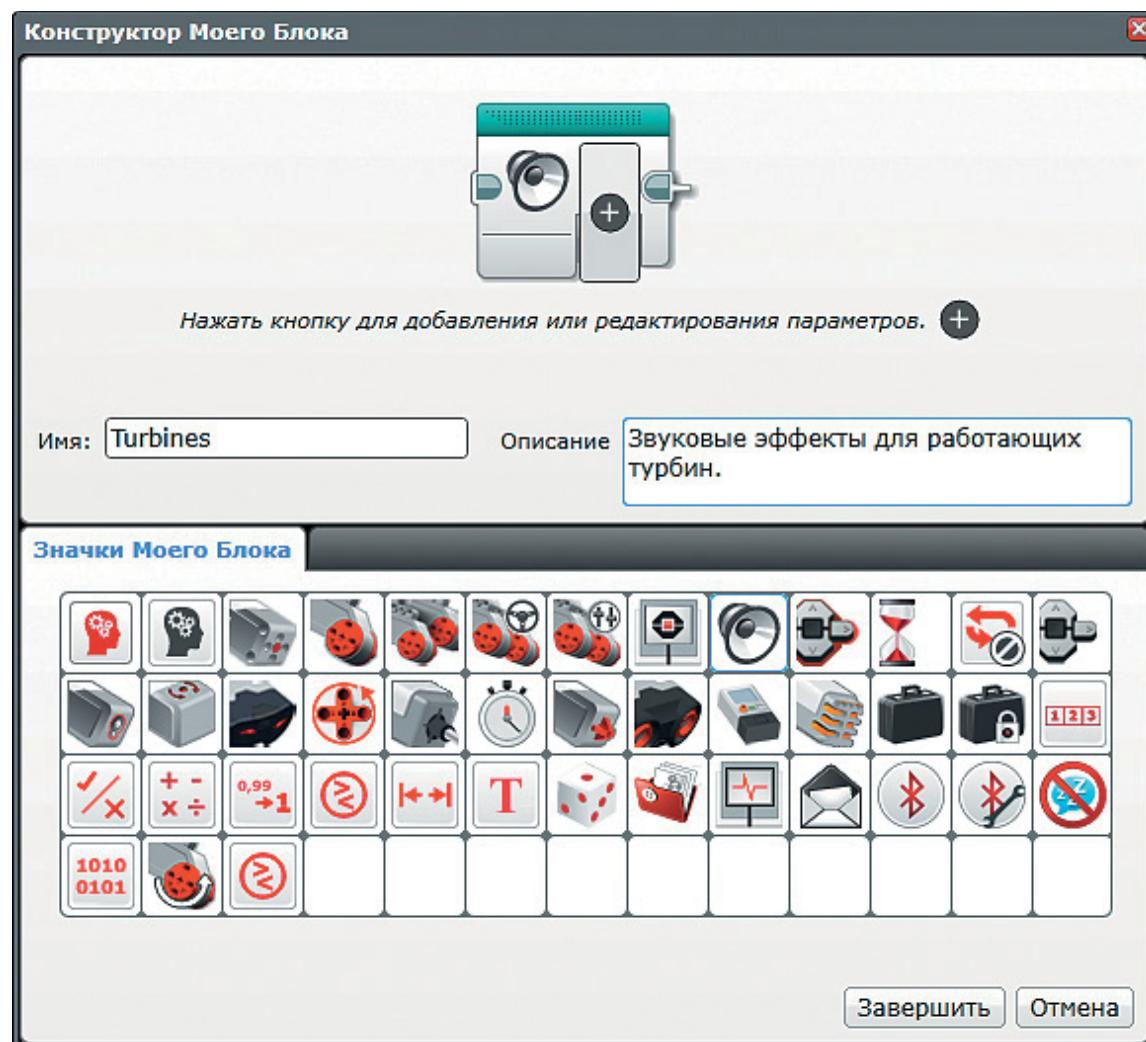
1. Добавь команду **Начало** (зелёный блок), как показано на рисунке.



2. Далее добавь команду **Цикл** (оранжевый блок).



3. Теперь создай подпрограмму, которая будет отвечать за аудиоэффект турбин. Ты уже знаешь, как это делать. Назвать её можно, например, **Turbines** (в переводе с английского — турбины).

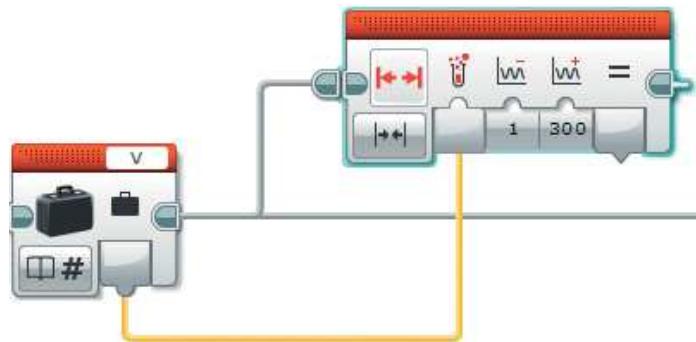


4. Кликни дважды по созданной команде **Turbines**. Учи, что звуковые эффекты должны соответствовать каждый своей скорости. Поэтому добавь внутрь цикла команду **Переменная** (красный блок), для обработки установи переменную **V** и выбери опцию **Считывание** → → **Числовое значение**.

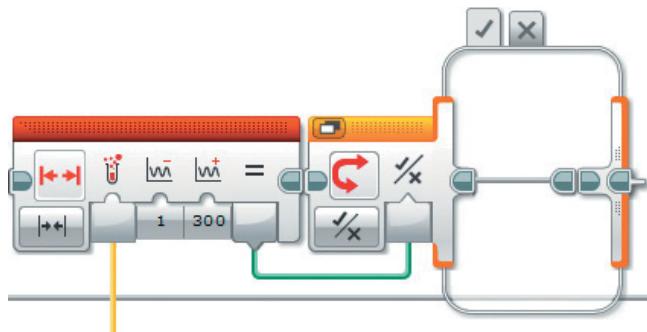


5. Теперь нужно создать похожую конструкцию, как ты делал для основной работы двигателей в подпрограмме **Engines**. Добавь команду **Интервал** (красный блок), соедини её с помощью шлейфа с командой **Переменная**, а выход переменной **V** соедини с входом **Тестовое значение**.

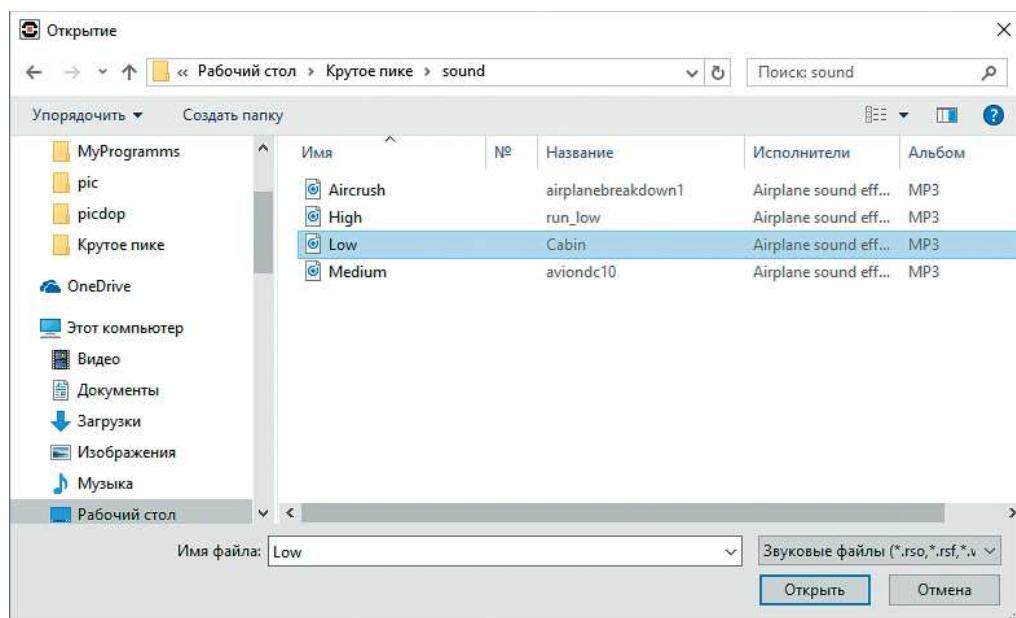
Введи во входы **Нижняя граница** и **Верхняя граница** с клавиатуры соответственно **1** и **300** — это будет диапазон скорости, при котором турбины будут издавать звук малой мощности.



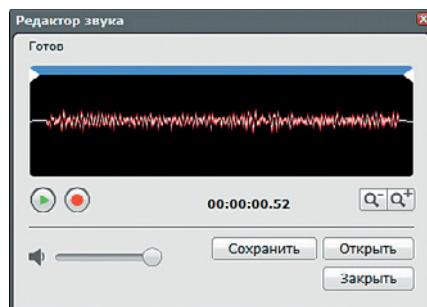
6. Добавь команду **Переключатель** (оранжевый блок), перейди к **Виду с вкладками**, выбери опцию **Логическое значение**, затем соедини выход «=» команды **Интервал** с входом **Переключателя**.



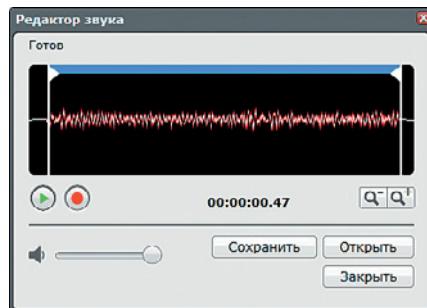
7. Внутри **Переключателя** понадобится всего лишь одна команда для воспроизведения звука. Однако сам звук ещё надо добавить в проект. Открой сверху меню **Инструменты** → **Редактор звука**. Нажми в открывшемся окне кнопку **Открыть**, найди в памяти компьютера сохранённый звуковой файл **Low** и кликни по нему дважды.



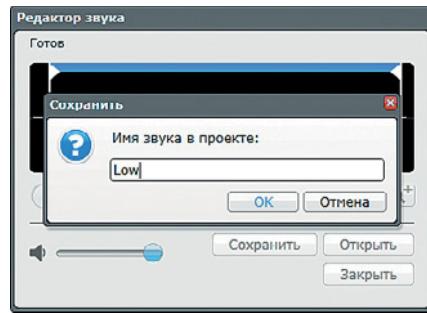
8. Звук откроется для предварительного редактирования.



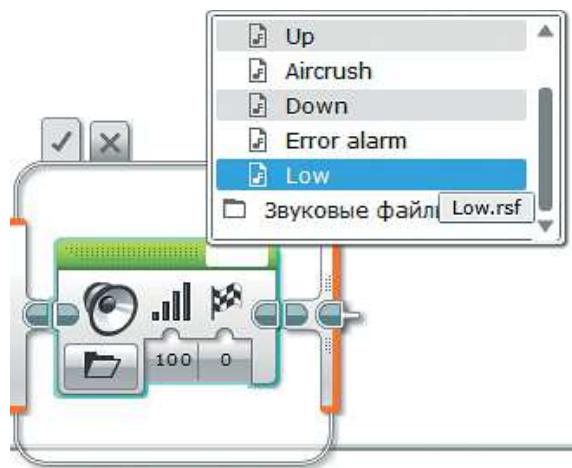
9. Ты можешь заметить, что у этого звука в начале и в конце есть тонкие линии — это тишина. Чтобы эффект звучал правильнее и непрерывно, тебе нужно **обрезать** тишину. Сделать это можно с помощью двух **бегунков сверху**: подвинь левый **бегунок вправо** до начала колебаний, а **правый — влево**.



10. Прослушай этот аудиоэффект, нажав кнопку с зеленым треугольничком, а затем нажми кнопку **Сохранить**. Задай в открывшемся окне имя, по которому данный звук будет идентифицироваться в твоём проекте, например, как и его оригинальное название, **Low**. И нажми **OK**, затем кнопку **Закрыть**.



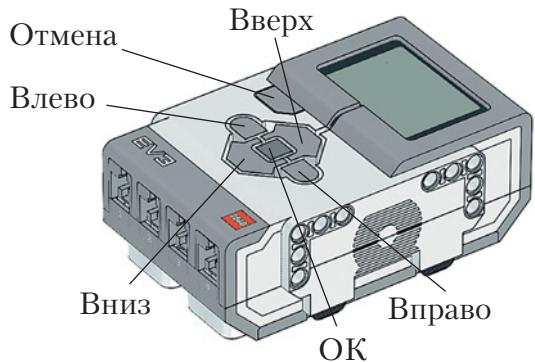
11. Добавь внутрь **Переключателя** команду **Звук** (зелёный блок), опции оставь без изменений. В правом верхнем окошке в папке **Звуки проекта** выбери файл **Low**. Аналогично, копируя и изменения команды **Интервал**, **Переключатель**, **Звук** и добавляя звуки с помощью **Редактора**, задай и остальные аудиоэффекты!



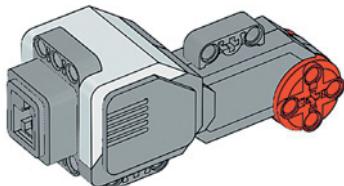
Как видишь, возможности твоего авиасимулятора могут расширяться и дальше! Попробуй реализовать новые горизонты твоей инженерной мысли!

Перечень деталей

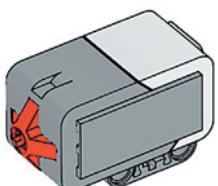
Модуль EV3, 1x



Большой мотор, 1x



Датчик касания, 1x

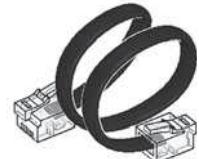


Кабель

Кабель, 25 см, 1x

Кабель, 35 см, 2x

Кабель, 50 см, 1x



USB-кабель, 1x

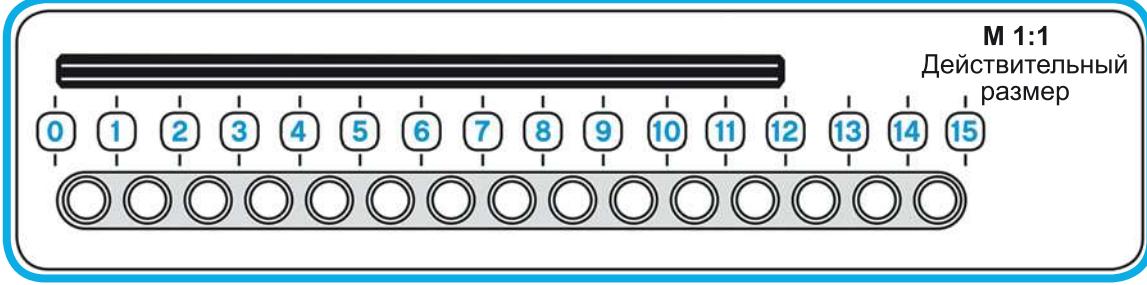


Оси

Ось № 3, серая, 5x

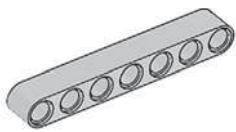


Ось № 5, серая, 2x

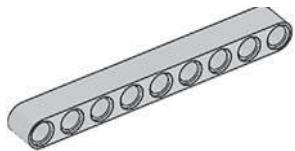


Балки прямые

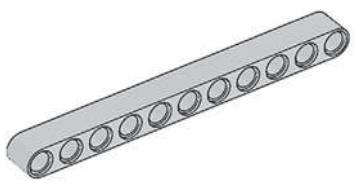
Балка № 7, серая, 3x



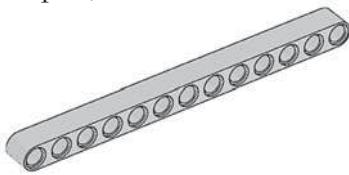
Балка № 9, серая, 1x



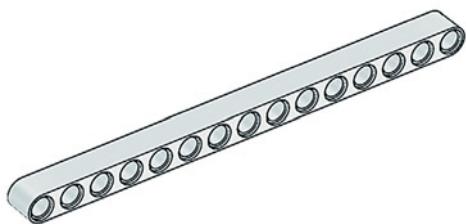
Балка № 11, серая, 4x



Балка № 13, серая, 2x

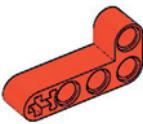


Балка № 15, белая, 6x

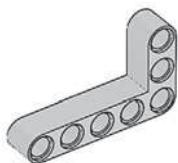


Балки угловые

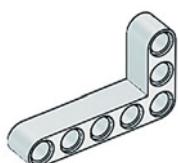
Прямоугольная балка 2 × 4,
красная, 5x



Прямоугольная балка 3 × 5,
серая, 2x



Прямоугольная балка 3 × 5,
белая, 4x



Поперечная балка 2 × 1,
красная, 2x



Блоки

Двойной поперечный блок,
3-модульный, чёрный, 4x



Поперечный блок,
2-модульный, серый, 2x



Штифты

Соединительный штифт,
2-модульный, чёрный, 36x



Соединительный штифт,
2-модульный, синий, 6x



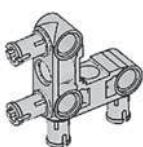
Соединительный штифт,
3-модульный, синий, 11x



Соединительный штифт
со втулкой, 3-модульный,
красный, 2x



Угловой соединительный
штифт, 3×3-модульный,
серый, 4x



Втулки

Втулка, $\frac{1}{2}$ -модульная, жёлтая,
1x



Втулка, серая, 1x



Втулка, 3-модульная, чёрная,
2x



Втулка, 3-модульная угловая,
чёрная, 1x



Который час



История развития часовогого мастерства



Технология измерения времени уходит корнями в глубокою древность. Время для нас является великой тайной Вселенной, которую мы постепенно разгадываем и учимся понимать. Именно поэтому так важно его исследовать.

Первые часы значительно отличались от тех, к которым ты привык. Чаще всего для измерения времени использовали некие предметы, которыми совершили определённые действия. Например, хозяйка знала, что яйцо будет сварено вкрутую, как только она дважды успеет наполнить кувшин водой, то есть эталоном измерения служил вполне конкретный кувшин.

Но эталоны у людей были разные, и меры различались. Примером ешё может служить деревянная лучина — тонкая длинная щепка. Время её горения составляет около 15 минут, как и тонкой свечки.

Узнать примерное время суток можно было, соорудив простейшие солнечные часы: воткнуть в землю палку или перо и нарисовать вокруг циферблата или вбить колышки. Такая палка называлась «гномон» (*γνωμόν*), что в переводе с греческого означает «указатель». Однако пользоваться такими часами можно было только в светлое время суток. И всё же эти часы подходили лишь для примерного измерения, но в быту этого вполне хватало.

Для научных измерений (в основном для астрономии) требовались более точные показания и отслеживание времени в течение дня или даже месяцев и лет. Изобретатели древности совершенствовали солнечные часы. Они покрывали циферблат множеством линий, показывающих соответствие направления тени и времени суток для всех месяцев.

Позднее было вычислено, что если располагать гномон параллельно земной оси, то конец тени будет двигаться вдоль циферблата по окружности с одинаковой скоростью и часовые деления окажутся **равными**. Такую конструкцию солнечных часов назвали экваториальными (рис. 1).



Рис. 1. Экваториальные часы



Интересно, что в нашем Северном полушарии с марта по сентябрь тень на экваториальных часах падает на циферблат, а с октября по февраль — под него. Не очень удобно, правда? Древние греки считали так же, поэтому сделали циферблат расположенным горизонтально, заменив гномон-палку на треугольник с острым углом, равным: $(90 - \phi)^\circ$, где ϕ — географическая широта местности. Например, для Афин она равна 38° , для Красноярска — 56° .

Когда мы говорим о старинных часах, помимо солнечных часов на ум всегда приходят **песочные** (рис. 2).

Они знакомы тебе по урокам физкультуры и кабинетам врачей. Песок для них подойдёт не любой. В XIII веке использовали следующий состав: смесь речного песка и мраморной пыли кипятили с лимонным соком и вином, снимали накипь, повторяя весь процесс десять раз! Только тогда удавалось добиться, чтобы песок не налипал на стенки и сыпался практически равномерно. Подобные часы обладали большой погрешностью, не годились для точных измерений и отсчитывали лишь небольшой отрезок времени.

Другим древним таймером были водяные часы (рис. 3), которые назывались «клепсидра» (κλεψυδρα), что в переводе с греческого — воровка воды.

Название отлично отображает принцип её действия. Клепсидра состояла из большого сосуда с отмеченными делениями и дырочками на дне. Когда вода вытекала, уровень её менялся и измерялся согласно отметкам. Отмерять равные промежутки времени требовалось на публичных выступлениях философов и спорщиков в судах, чтобы никто не говорил дольше, чем остальные.

Другая стихия тоже помогала измерять время. В Китае были широко распространены **огненные часы**. Они представляли собой горящие металлические палочки с привязанными на нитях шариками в определённых местах: когда фитиль догорал до нити, та пережигалась, а груз со звоном падал в глиняную посуду. Существовали также часы



Рис. 2. Песочные часы



Рис. 3. Клепсидра

с фитилями из смеси нескольких сортов дерева и благовоний, закрученными большой спиралью. Они могли гореть месяцами, не требуя обслуживания.

Однако сделать совершенно одинаковые фитили не получалось, поэтому часы не были точными. К тому же воду в клепсидре можно налить заново, а сгоревшие части восстановлению не подлежат. Также китайцы использовали свечи с нанесёнными уровнями для определения часа. Если в фитиль добавляли травы, то свеча каждый час издавала различный аромат, поэтому на часы для определения времени даже смотреть не требовалось.

Свечные часы использовались и в Европе. Они были схожи с китайскими, а вот применяемые наряду с ними **лампадные** отличались. В них наливалось определённое количество масла, сгорающее за отмеченный заранее промежуток времени.

Точное измерение времени связано именно с механическими часами. В греческой литературе с 300 года до нашей эры по 500 год нашей эры описываются механические устройства, позволяющие следить за перемещением небесных тел с помощью расчётов.

Не отставал и Китай, где примерно в 725 году инженер-астроном И Син с помощником Лян Линцзанем создали небесный глобус с изображением небесного экватора, созвездий и солнечной широты — астрономический инструмент с механическими часами на водном приводе. Каждые четверть часа автоматически звучал барабан, а час — колокол. Фигуры и указатели перемещались с помощью зубчатых передач, как в современных механических часах.

Изобретение современных механических часов во многих источниках приписывается Герберту д'Ориллаку (около 946–1003), позднее ставшему папой римским Сильвестром II. Его часы, построенные в Магдебурге (Германия), также показывали перемещения небесных тел. Именно благодаря Герберту в Европе стали использовать для записи арабские цифры — он заметил лёгкость операций с ними и начал вводить их в обиход взамен римских.

Тем не менее в XIII веке европейские башенные часы не имели ни циферблата, ни даже стрелок! Они оповещали местное население с помощью звона через определённые промежутки времени. В 1335 году в Милане были построены первые башенные часы со стрелкой, ход которых обеспечивался двухсоткилограммовой гирей на канате, врашающейся вал. Чтобы канат не разматывался резко, его сдерживали грузами. Однако точностью данные часы особо не отличались.

Первые механические часы на просторах средневековой Руси были созданы в 1374 году в городе Кафа (сейчас — Феодосия, Крым). В 1404 году башенные часы появились и у Московского Кремля на башне дворца великого князя Василия Дмитриевича, сына Дими-

трия Донского. Люди называли их «дивом мира», любуясь изображением Луны, Солнца и звёзд на шестиметровом циферблате (рис. 4).

Часы построил Лазарь Сербин, наделив их примечательной особенностью: двигалась не часовая стрелка, а сам циферблат. В движение механизм приводил подвешенный груз, заставляющий из-за силы тяжести вращать шестерни и двигать круг со звёздным небом.

Что тебе известно о маятниковых часах и их истории? В 1584 году Галилей, зайдя в собор в родном городе Пиза (Италия), обратил внимание, как качаются бронзовые лампы, подвешенные к потолку. Они имели разный размер и вес, но были подвешены на одинаковых цепях. Это вдохновило гения на идею маятника, регулирующего неравномерный ход строившихся в то время часов. В жизнь же изобретение воплотил в 1675 году Христиан Гюйгенс, проводивший собственные расчёты и придумавший конструкцию независимо от итальянца. Система приводилась в движение пружиной, равномерность хода же обеспечивалась маятником. Он через валок был сцеплен с анкером (дуги с двумя скошенными выступами), при качании маятника он входил в прорези секундного колеса и регулировал скорость его движения. Минутное и часовое колёса также были связаны с секундным с помощью промежуточных колёс. После касания анкером колеса валок возвращается в исходное положение и тем самым двигает маятник.

Ты, наверное, видел в старых фильмах людей, обслуживающих башенные часы. Эта профессия действительно была необходима и уважаема. Часы требовалось заводить каждый день! К концу суток они могли отстать на 10–15 минут, а ведь по главным (зачастую — единственным) башенным часам ориентировался весь город. Если в первый день «пропажу времени» никто не заметил бы, потому что минутная стрелка у механических часов появилась лишь около 1675 года, а секундная — в 1700 году, то через четыре дня потерялся бы целый час!

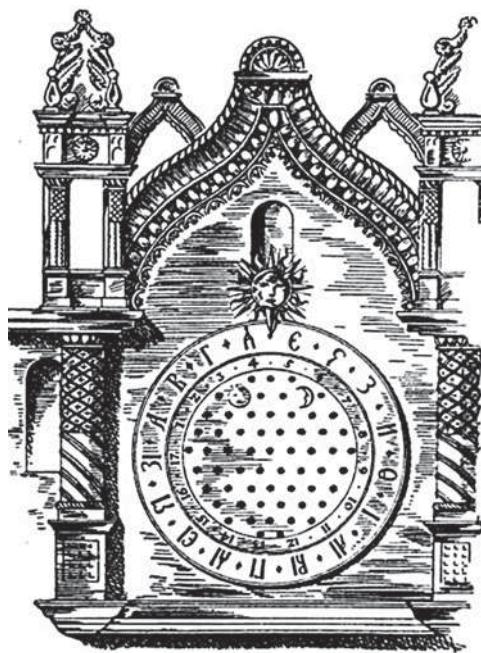


Рис. 4. Первые механические часы Московского Кремля

Кстати!

Первые башенные часы в Европе были построены в 1288 году в Вестминстере, Англия. А самые старейшие на сегодня в Европе башенные часы расположены в городе Гродно в Белоруссии.



Рис. 5. Первые карманные часы

Тем не менее часы были не только башенными. Уже в далёком 1500 году Петер Генлейн собрал первые карманные часы (рис. 5)!

Конечно, они мало были похожи по удобству на современные, но технология совершенствовалась. Например, в конце XVIII века часовщик Абрахам-Луи Бреге придумал современную **заводную головку**, которой к тому же можно было переводить стрелки. До этого карманные часы заводились специальным ключом.

Какие часы называют «дедушкиными»? Правильно, **часы с кукушкой!** Жил в XVIII веке в немецкой Баварии в городке Шёнвальде мастер Франц Антон Кеттер. Ему хотелось сделать часы

лучше, чем у остальных, подобные «живым». Например, утром они могли бы имитировать крик петуха. Однако задача оказалась слишком сложной — заставить механизм издавать четыре разных звука («Ку-ка-ре-ку!») у него никак не получалось. Задумавшись, мастер пришёл к выводу, что кукушка — тоже хорошая птица. Главное, лёгкая. С помощью миниатюрных кузнецких мехов Кеттер заставил механическую птицу издавать знаменитое: «Ку-ку!». Правда, история кончилась плохо. Первые зрители почему-то разбежались с криками: «Колдун!».

Примерно в это же время в России жил Иван Петрович Кулибин (1735–1818). Он преподнёс императрице Екатерине II невероятные по тонкости исполнения и красоте часы в форме гусиного яйца.

Карманные и наручные часы действуют по похожему принципу, что и маятниковые, но с одним существенным различием — вместо маятника равномерное вращение регулируется специальным металлическим стержнем (билянцем) с двумя лопatkами (взамен анкера).

Ход современных часов зависит от натяжения пружины. Если она раскручена почти полностью, ход часов замедляется. Чтобы избежать этого, часовую пружину делают длинной, используя только среднюю часть, где она почти всегда в одном положении. Другое решение — замена маятника на баланс. Балансом называется маленькое маховое колесо, попеременно вращающееся то вправо, то влево. В таких часах пружина накручивается на вал, получающий вращение через промежуточные колёса от заводной головки. Чтобы пружина не раскрутилась, добавляется деталь под названием «храповая собачка». Она стопорит барабанное колесо.

Только что ты прошёл через всю историю часовых механизмов от древности до современности! Было интересно? Ты уже догадался, что сделать чисто механические часы на твоём наборе LEGO®



MINDSTORMS® EV3 трудновато, поэтому ты соберёшь **аналоговые часы**. Что это такое? Давай разбираться!

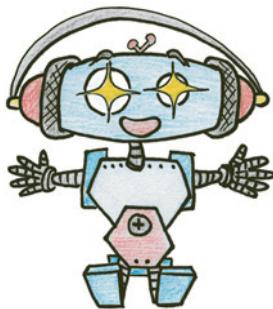
Аналоговыми называются часы, время на которых указано с помощью циферблата и стрелок. Как видишь, в определении не идёт речь о конкретном механизме. Аналоговые часы бывают механическими, кварцевыми и электронными. Хорошие механические часы имеют погрешность около 40 с, лучшие же кварцевые — 5 с за целый день.

Принцип действия **кварцевых часов** основан на способности кристалла кварца при прохождении через него тока выдавать импульсы с определённой частотой.

Электронные относятся к часам с повышенным классом точности, потому что люди научились контролировать импульсы в электрических схемах намного лучше.

Ты займёшься созданием именно электронных аналоговых часов — самых точных и, благодаря наличию стрелок, всё же «живых». Возможно, ты даже усовершенствуешь их конструкцию, как когда-то смог Абрахам-Луи Бреге.

Вперёд, инженер!



Этап 1. Изучение устройства роботизированных часов с кукушкой

Рассмотри модель роботизированных часов с кукушкой, собранную на основе набора LEGO® MINDSTORMS® Education EV3 (рис. 6).

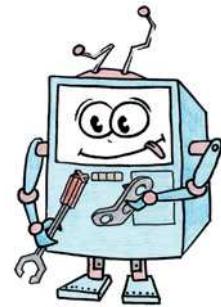
Попробуй выделить на ней рабочие детали: часовую и минутную стрелки; редуктор, находящийся позади стрелок и обеспечивающий передачу вращения среднего мотора через зубчатые колёса; дверцу «домика» кукушки; рамку циферблата.

Именно их тебе предстоит собрать.



Рис. 6. Роботизированные часы с кукушкой

Этап 2. Сборка роботизированных часов с кукушкой



Теперь попробуй собрать эту модель. Если что-то непонятно, обращайся за помощью к взрослым.

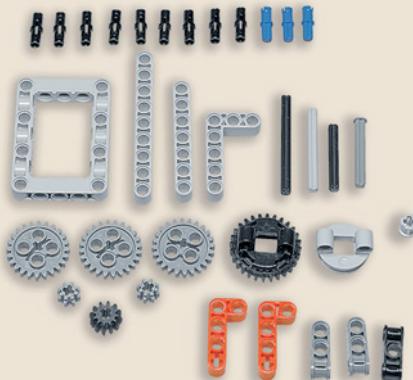
Обрати внимание, в конце проекта в таблице приведены все детали, которые потребуются тебе для сборки. Эта таблица поможет быстро найти необходимую деталь и не ошибиться при конструировании.

ШАГ 1. СБОРКА ПЕРВОЙ ЧАСТИ РЕДУКТОРА ЧАСОВОГО МЕХАНИЗМА

Детали для сборки:

- вращающаяся верхушка на 28 зубьев, чёрная, 1x;
- вращающееся днище на 28 зубьев, серое, 1x;
- рамка 5×7 , серая, 1x;
- ось № 6, чёрная, 1x;
- ось № 5, серая, 1x;
- ось № 4, чёрная, 1x;
- ось № 4 с фиксирующим элементом, серая, 1x;
- прямоугольная балка 3×5 , серая, 1x;
- прямоугольная балка 2×4 , красная, 2x;
- балка № 9, серая, 1x;
- балка № 7, серая, 1x;
- соединительный штифт, 2-модульный, чёрный, 9x;
- соединительный штифт, 2-модульный, синий, 3x;
- зубчатое колесо на 8 зубьев, серое, 2x;

- ☞
- зубчатое колесо на 12 зубьев, чёрное, 1x;
 - зубчатое колесо на 24 зуба, серое, 3x;
 - поперечный блок, 3-модульный, серый, 2x;
 - фиксатор, 3-модульный, чёрный, 1x;
 - втулка, серая, 1x.



1. Соедини чёрную вращающуюся верхушку с серым вращающимся днищем.



2. Вставь два чёрных штифта с внутренней короткой стороны рамки 5×7 .



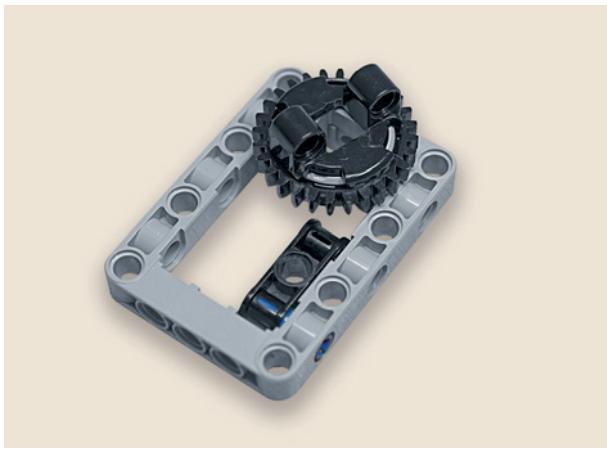
3. Закрепи на них серыми модулями собранный вращающийся элемент.



4. Возьми синий 2-модульный штифт и вставь его в крайний модуль 3-модульного фиксатора.



5. Закрепи получившуюся деталь на внутренней длинной стороне рамки.



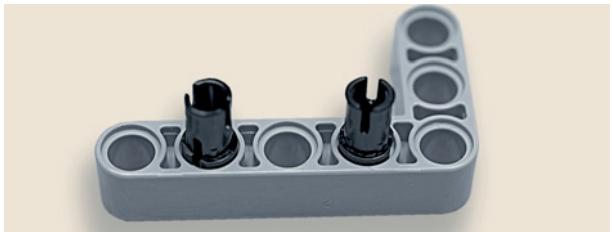
6. Переверни рамку чёрной вращающейся верхушкой вниз и вложи с внутренней стороны два попечерных 3-модульных блока напротив фиксатора.



7. Закреши вложенные модули осью № 5, вставив её в боковое среднее отверстие рамки.



8. Теперь возьми прямоугольную балку 3 × 5 и вставь в её второй и четвёртый модули по одному чёрному штифту с длинной стороны.



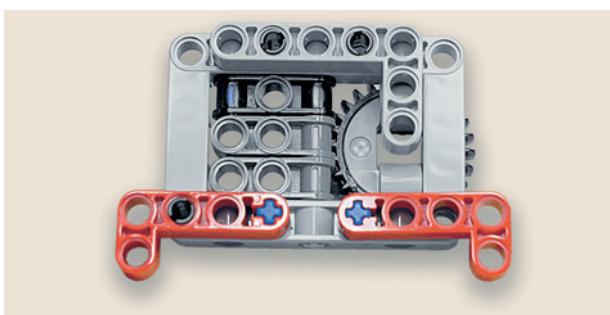
9. Закреши получившуюся деталь сверху на рамке во втором и в четвёртом модулях.



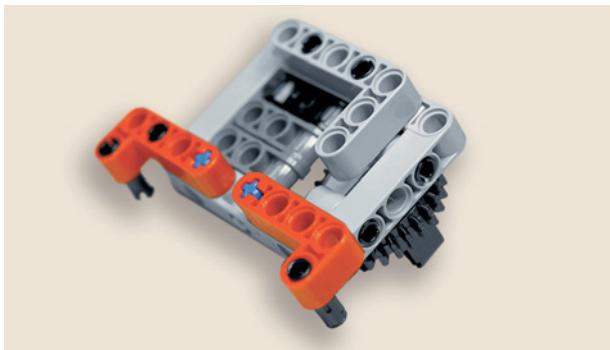
10. Возьми две красные прямоугольные балки и вставь в их крестовые модули по одному синему 2-модульному штифту. В левой балке в третьем модуле справа закрепи ещё и чёрный 2-модульный штифт.



11. Установи обе балки на рамке во втором и в четвёртом модулях.



12. Вставь в угловые модули красных балок снизу по одному чёрному штифту.



13. Закрепи на этих штифтах балку № 9.



14. Вставь сверху в крайние свободные модули балки № 9 по одному чёрному штифту.



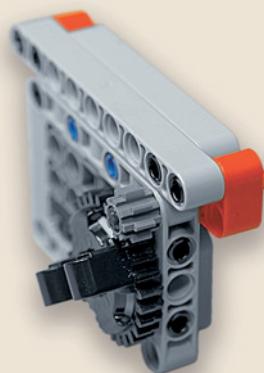
15. Установи на этих штифтах серую балку № 7.



16. Возьми чёрную ось № 4 и зубчатое колесо на 8 зубьев и соедини их вместе.



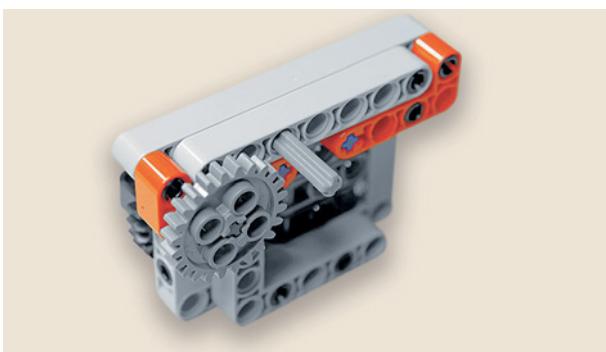
17. Продень готовую деталь в модуль серой рамки со стороны чёрной вращающейся верхушки, как показано на рисунке.



18. Надень на ось № 4 с обратной стороны зубчатое колесо на 24 зуба.



19. Продень ось № 4 с фиксатором в четвёртый модуль справа.



20. Надень на эту ось зубчатое колесо на 12 зубьев до упора.



21. Надень на выступающий конец оси № 4 зубчатое колесо на 24 зуба.



22. Продень ось № 6 до конца с левой стороны от зубчатого колеса в четвёртый модуль балки так, чтобы она не выступала снизу, и зафиксируй её серой втулкой.



23. Надень сверху на ось № 6 поверх фиксирующей втулки зубчатое колесо на 8 зубьев.



24. Надень поверх зубчатого колеса на 8 зубьев ещё одно зубчатое колесо, но на 24 зуба.



Одна половина редуктора готова! Можно приступать к сборке второй половины редуктора часового механизма.

ШАГ 2. СБОРКА ВТОРОЙ ЧАСТИ РЕДУКТОРА ЧАСОВОГО МЕХАНИЗМА

Детали для сборки:

- вращающаяся верхушка на 28 зубьев, чёрная, 1x;
- вращающееся днище на 28 зубьев, серое, 1x;
- рамка 5 × 7, серая, 1x;
- балка № 3, чёрная, 1x;
- фиксатор, 3-модульный, чёрный, 1x;
- соединительный штифт, 2-модульный, чёрный, 4x;
- соединительный штифт, 2-модульный, синий, 3x;
- соединительный штифт, 3-модульный, синий, 1x;
- ось № 5, серая, 1x;
- ось № 8, чёрная, 1x;



- поперечный блок, 3-модульный, серый, 2x;
- поперечный блок, 2-модульный, серый, 1x;
- зубчатое колесо на 8 зубьев, серое, 1x;
- зубчатое колесо на 12 зубьев, чёрное, 1x.



1. Соедини чёрную вращающуюся верхушку с серым вращающимся днищем.
2. Возьми рамку 5 × 7 и вставь два чёрных штифта с внутренней узкой стороны в первый и третий модули.

3. Закрепи на этих штифтах собранный вращающийся элемент.



4. Возьми 3-модульный фиксатор и вставь в его правый модуль синий 2-модульный штифт.



5. Закрепи получившуюся деталь на внутренней широкой стороне рамки справа.



6. Вложи внутрь рамки 5×7 два поперечных 3-модульных блока, чтобы они совпали с чёрной балкой, и закрепи всю конструкцию осью № 5, вставив её в среднее отверстие сбоку широкой стороны рамки.



7. Переверни деталь вверх вращающимся элементом. Вставь два чёрных штифта в первый и третий модули рамки слева.



8. Установи на этих штифтах балку № 3 чёрного цвета.



9. Вставь с правой стороны сбоку синий 2-модульный штифт.



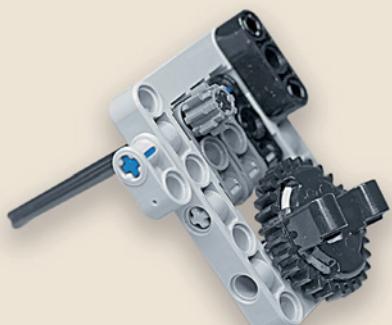
10. Присоедини к этому штифту серый поперечный 2-модульный блок.



11. Возьми ось № 8 и надень на неё серое зубчатое колесо на 8 зубьев.



12. Продень сквозь серый поперечный 3-модульный блок свободный конец чёрной оси № 8.



13. Надень снизу на ось № 8 чёрное зубчатое колесо на 12 зубьев.



14. Возьми синий 2-модульный штифт и вставь в балку № 3 чёрного цвета по центру.

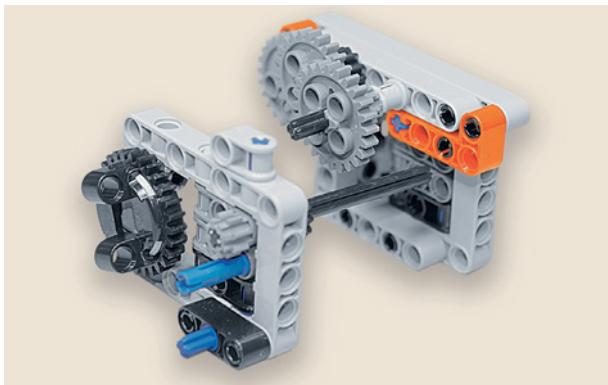


- 
15. Вставь короткой стороной синий 3-модульный штифт в свободный серый поперечный блок.



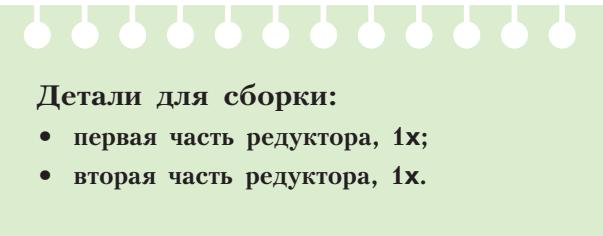
На этом сборка второй части редуктора часового механизма завершена. Теперь нужно соединить обе части редуктора.

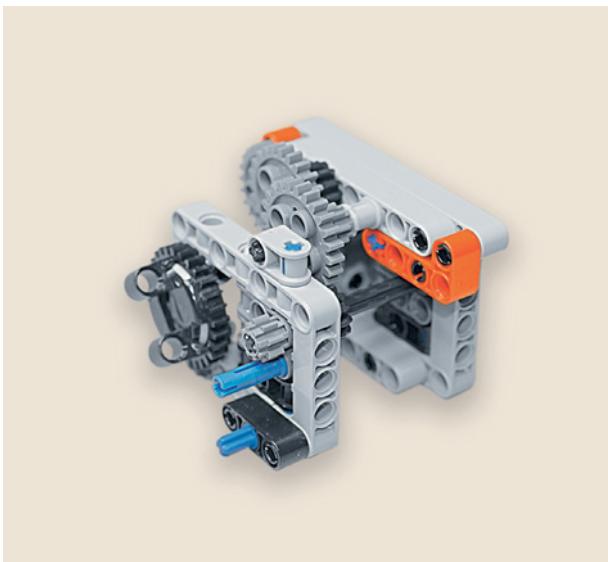
ШАГ 3. СБОРКА РЕДУКТОРА ЧАСОВОГО МЕХАНИЗМА



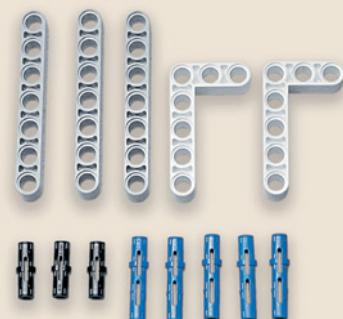
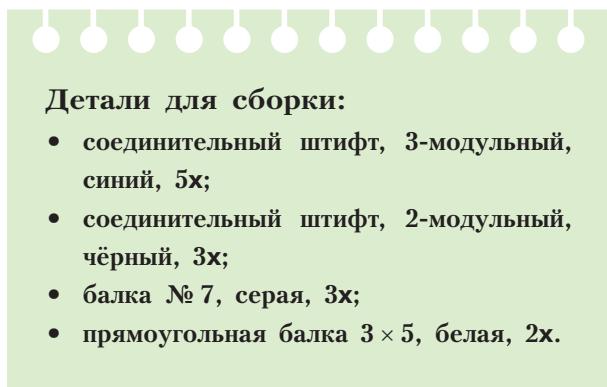
Детали для сборки:

- первая часть редуктора, 1x;
- вторая часть редуктора, 1x.

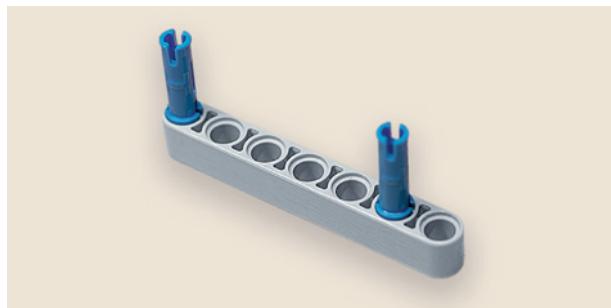
- 
1. Возьми две готовые части редуктора и соедини их, как показано на рисунке.



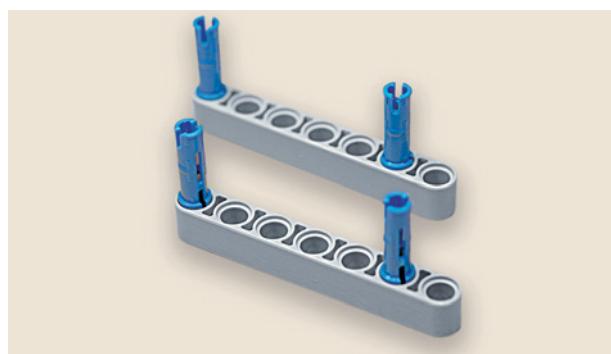
ШАГ 4. СБОРКА ОСНОВАНИЯ ЧАСОВОГО МЕХАНИЗМА



1. Возьми балку № 7 и вставь синие 3-модульные штифты в первое и шестое отверстия короткими концами.



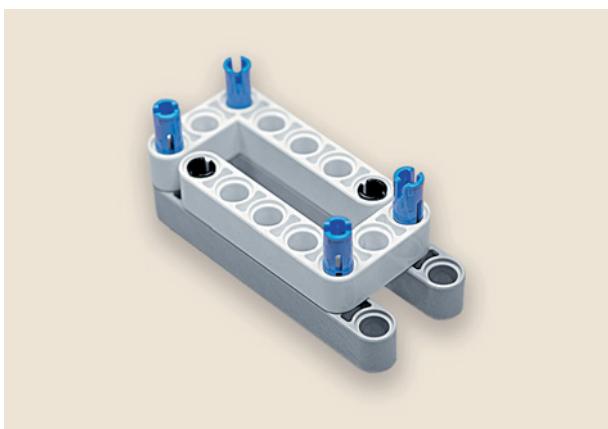
2. Повтори то же самое с ещё одной балкой.



3. Вставь в пятое отверстие верхней балки и во второе отверстие нижней по чёрному штифту.



4. После этого установи сверху две белые прямоугольные балки 3×5 .



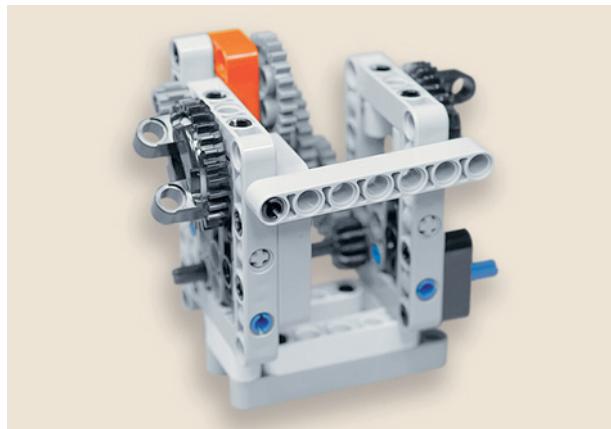
5. Теперь поставь часовой механизм на собранное основание.



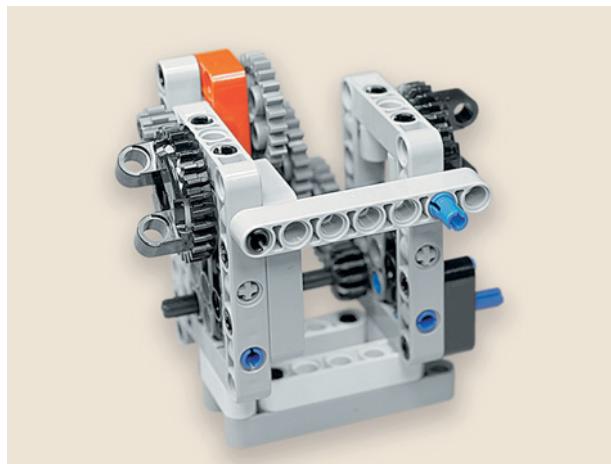
6. Поверни собранный механизм так, чтобы красные балки оказались слева. Вставь слева в свободный модуль рамки чёрный штифт.



7. Закрепи на этом штифте серую балку № 7.



8. Вставь в шестой модуль синий 3-модульный штифт длинным концом так, чтобы он прошёл и через рамку.



Часовой механизм теперь может стоять!

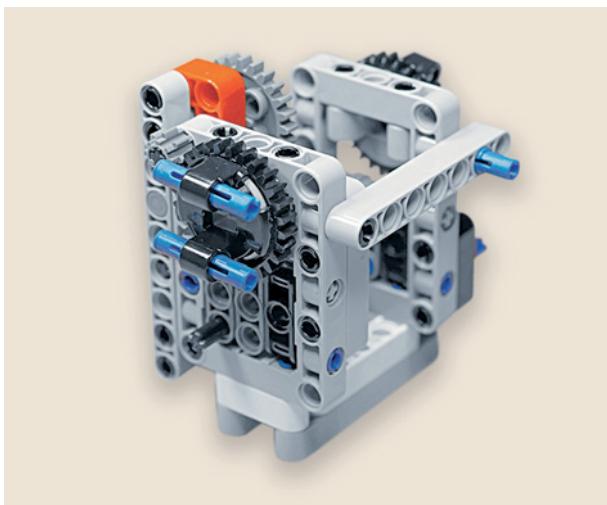
ШАГ 5. СБОРКА ЧАСОВОЙ СТРЕЛКИ

Детали для сборки:

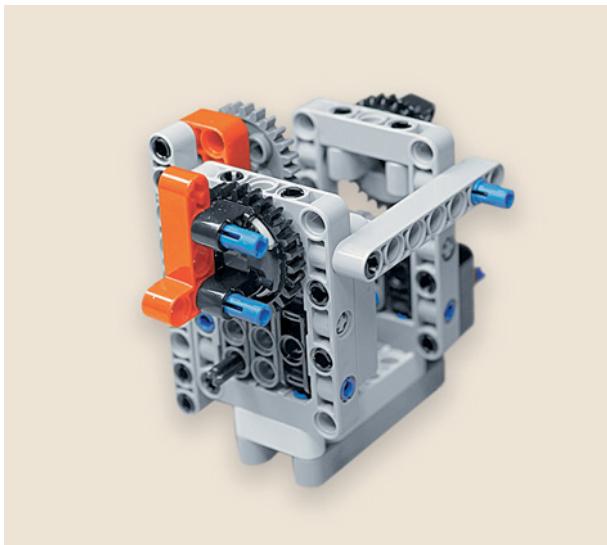
- соединительный штифт, 3-модульный, синий, 2x;
- соединительный штифт, 2-модульный, синий, 1x;
- балка № 3, жёлтая, 1x;
- прямоугольная балка 2 × 4, красная, 2x.



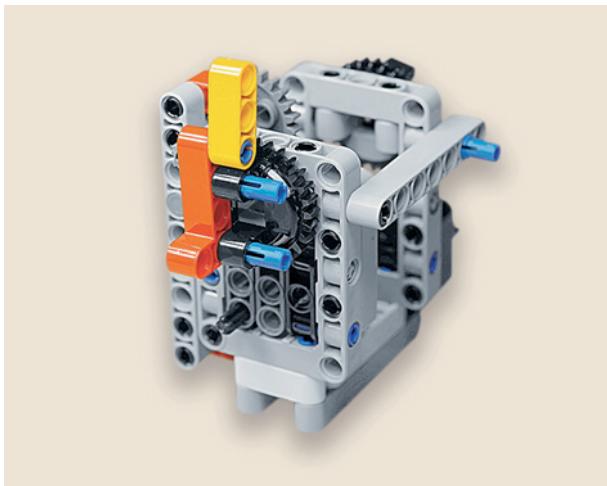
1. Возьми два синих 3-модульных штифта и вставь их слева в отверстия вращающегося элемента длинными концами вперёд.



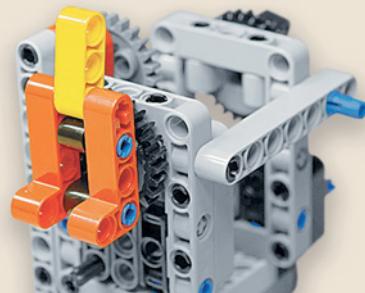
2. Установи на этих штифтах слева красную прямоугольную балку.



3. Вставь в её крестовой модуль 2-модульный штифт и закрепи на нём жёлтую балку.



4. Теперь зафиксируй всю конструкцию второй красной прямоугольной балкой с правой стороны.



Молодец! Теперь ты можешь узнать, который час, но впереди тебя ждёт уточнение по минутам!

ШАГ 6. СБОРКА МИНУТНОЙ СТРЕЛКИ

Детали для сборки:

- угловая втулка, 3-модульная, чёрная, 1x;
- втулка, 3-модульная, чёрная, 1x;
- втулка, 2-модульная, красная, 2x;
- ось № 10, чёрная, 1x;
- ось № 2, красная, 4x;
- соединительный штифт с втулкой, 3-модульный, красный, 1x.



1. Возьми 3-модульную угловую втулку и вставь ось № 10 в одно из её крестообразных отверстий.



2. Вставь красную ось № 2 в свободное крестообразное отверстие втулки.



3. Далее надень на эту ось красную 2-модульную втулку.



4. Вставь в эту втулку ещё одну красную ось № 2.



5. Надень на неё красную втулку.



6. Вставь сверху ещё одну красную ось № 2.



7. Надень на свободный конец оси № 10 чёрную 3-модульную втулку.



8. Вставь в неё красную ось № 2.



9. Надень на красную ось с другой стороны красный штифт.



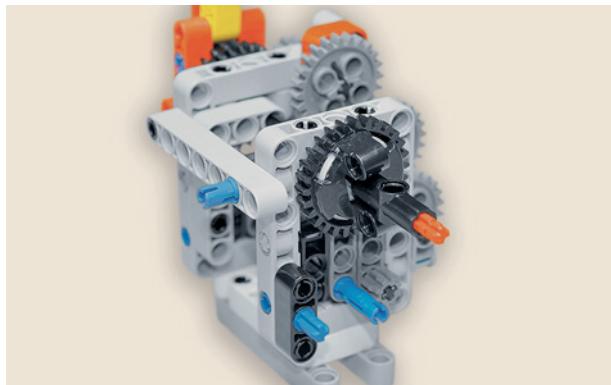
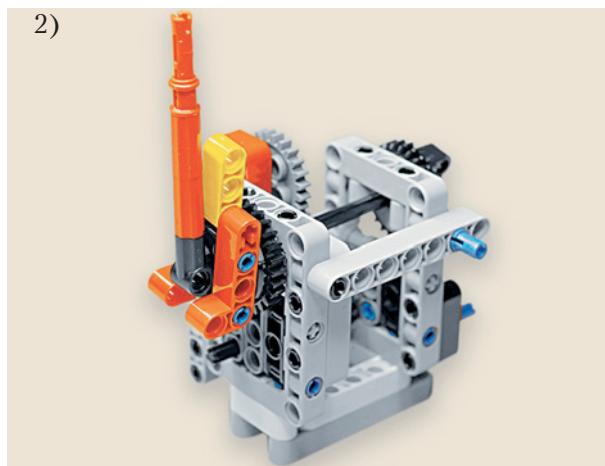
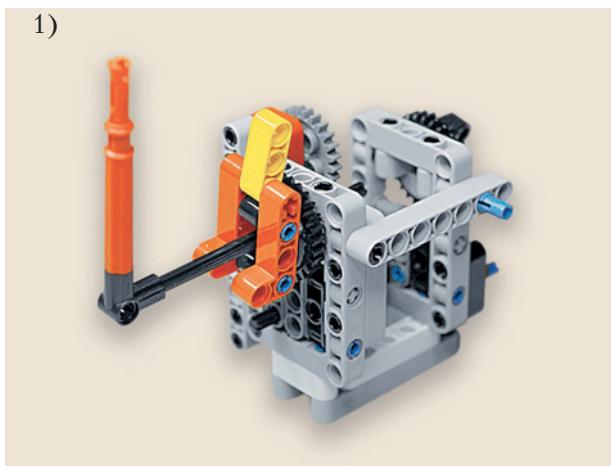
Минутная стрелка готова!

ШАГ 7. УСТАНОВКА МИНУТНОЙ СТРЕЛКИ

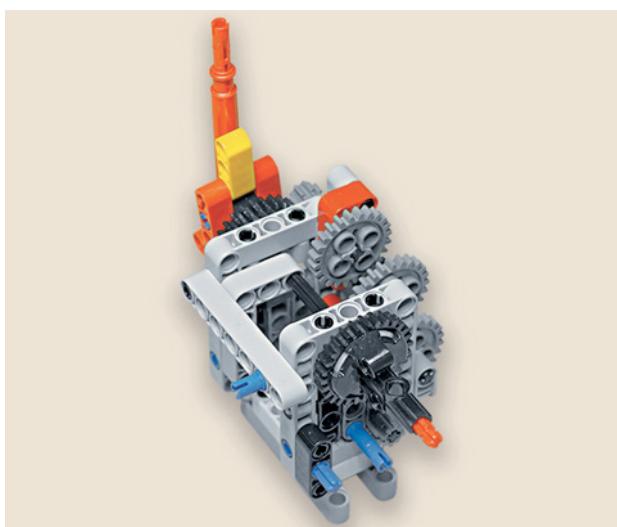
1. Теперь на время разъедини ось № 10 и 3-модульную чёрную втулку.



2. Поверни собранный механизм часовой стрелкой к себе. Возьми минутную стрелку и свободный конец оси № 10 чёрного цвета и пропусти её через центр вращающегося элемента насквозь.



3. Надень с противоположной стороны на свободный конец оси № 10 3-модульную чёрную втулку в крестообразное отверстие.



4. Для окончания сборки часового механизма тебе надо установить зубчатое колесо. Но перед этим поверни обе стрелки на 12 часов ровно.

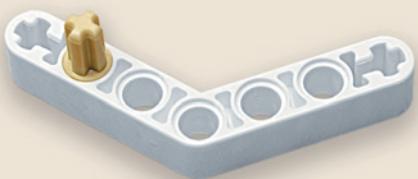
ШАГ 8. СБОРКА ЧАСОВОГО МЕХАНИЗМА

Детали для сборки:

- угловая балка 4×4 , белая, 1x;
- соединительный штифт, 2-модульный, бежевый, 1x;
- зубчатое колесо на 24 зуба, серое, 1x.



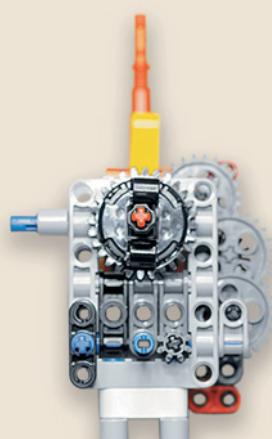
1. Возьми угловую балку 4×4 и вставь бежевый штифт во второе отверстие слева.



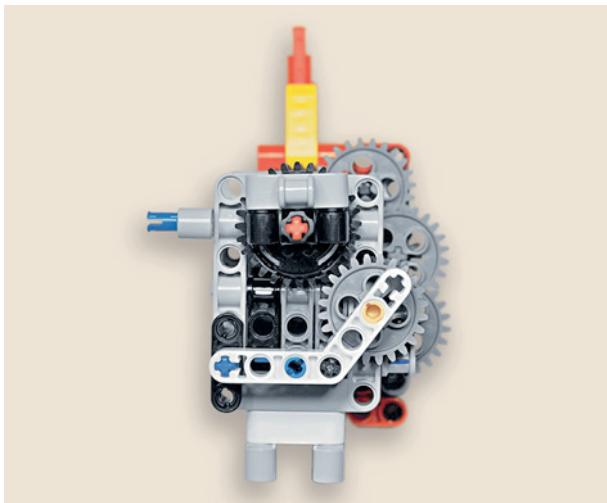
2. Надень сверху на него зубчатое колесо на 24 зуба.



3. Переверни часовой механизм так, чтобы стрелки смотрели **от тебя**.



- 
- 
- Придерживая обе стрелки в положении 12 часов ровно, закрепи угловую балку в сборе с зубчатым колесом, как показано на рисунке.



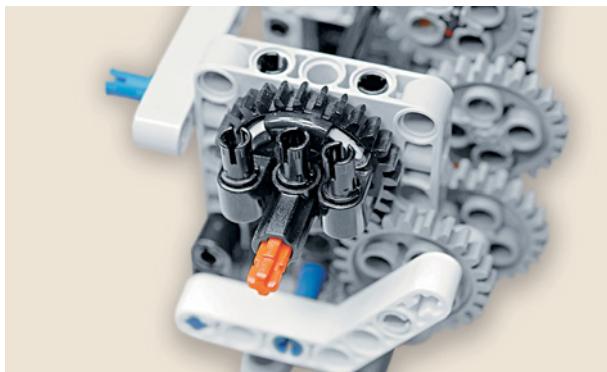
Часовой механизм готов!

ШАГ 9. СБОРКА МЕХАНИЗМА БОЯ ЧАСОВ



Детали для сборки:

- поперечный блок, 4-модульный, серый, 1x;
- соединительный штифт, 2-модульный, чёрный, 3x;
- ось № 4, чёрная, 1x;
- ось № 2, красная, 1x;
- труба, 2-модульная, серая, 1x;
- поперечный фиксатор, 2-модульный, чёрный (резиновый), 1x.



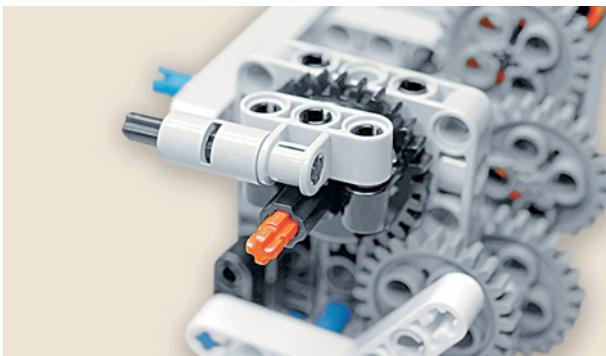
- Для того чтобы наши часы каждый час подавали звуковой сигнал и из них выпрыгивала кукушка, тебе надо собрать управляющий механизм. Для этого закрепи сверху на заднем вращающемся элементе и втулке минутной стрелки три чёрных 2-модульных штифта.



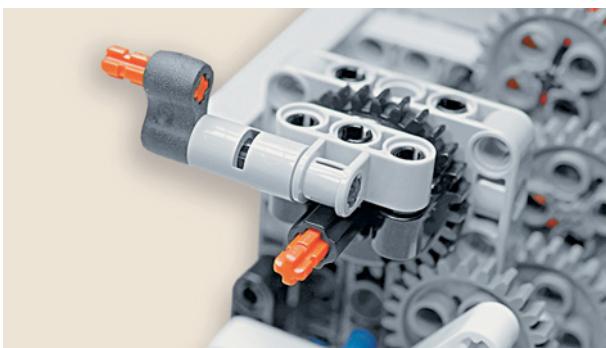
2. Далее надень на эти штифты по-перечный 4-модульный блок.



3. Вставь в отверстие слева на по-перечном 4-модульном блоке чёрную ось № 4 и на неё надень серую трубу.



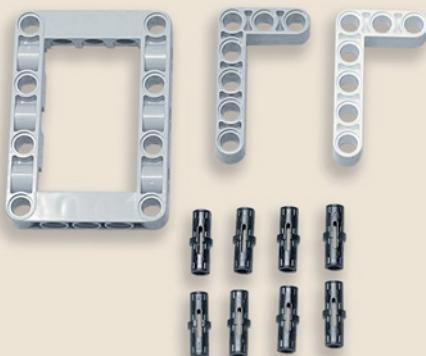
4. Закрепи на оси резиновый по-перечный фиксатор свободным отверстием вверх. Вставь в свободное крестообразное отверстие красную ось № 2.



ШАГ 10. СБОРКА КРЕПЛЕНИЯ ОСНОВАНИЯ ЧАСОВ

Детали для сборки:

- прямоугольная балка 3×5 , серая, 1x;
- прямоугольная балка 3×5 , белая, 1x;
- соединительный штифт, 2-модульный, чёрный, 8x;
- рамка 5×7 , серая, 1x.



1. Для начала переверни часовой механизм основанием вверх и стрелками к себе. Закрепи четыре чёрных штифта во втором и в четвёртом отверстиях длинной стороны серой и белой прямоугольных балок.



2. Закрепи их на основании.



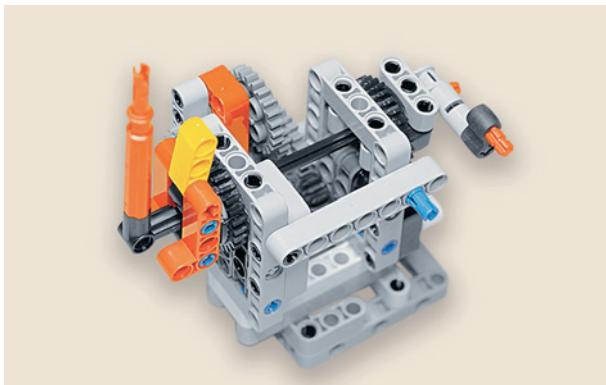
3. Далее возьми рамку 5×7 и закре-
пи по длинным сторонам в пер-
вом и третьем модулях чёрные
штифты.



4. Прикрепи рамку к собранным угловым балкам так, чтобы штифты попали в их угловые модули.



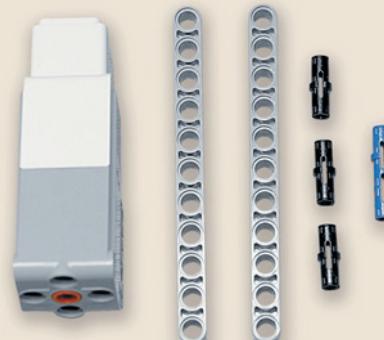
5. Теперь можно перевернуть механизм и поставить его на готовое основание.



ШАГ 11. СБОРКА И КРЕПЛЕНИЕ ЭЛЕКТРОПРИВОДА ЧАСОВОГО МЕХАНИЗМА

Детали для сборки:

- соединительный штифт, 2-модульный, чёрный, 3x;
- соединительный штифт, 3-модульный, синий, 1x;
- балка № 11, серая, 2x;
- средний мотор, 1x.



1. Возьми средний мотор и поверни его крестообразным отверстием к себе и вставь два чёрных соединительных штифта с правой стороны мотора.



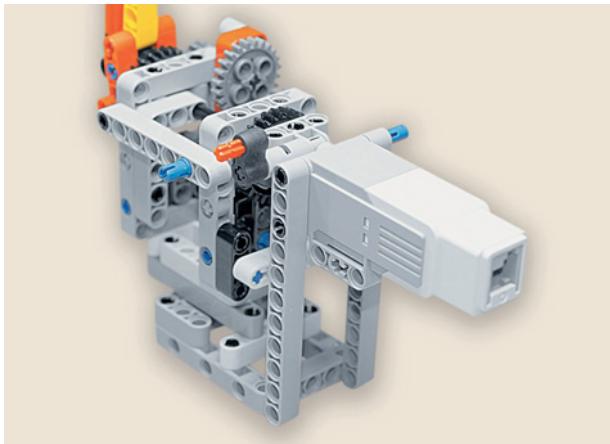
2. Далее вставь с левой стороны в верхнее отверстие сбоку синий 3-модульный штифт, а в нижнее отверстие чёрный соединительный штифт.



3. Теперь закрепи по обеим сторонам мотора балки № 11 в первый и третий модули.



4. Прикрепи готовую деталь к часовому механизму: надень её на свободный конец красной оси напротив стрелок.



Скоро твои часы пойдут!

ШАГ 12. СБОРКА КРЕПЛЕНИЯ ДЛЯ СРЕДНЕГО МОТОРА И ПРОГРАММИРУЕМОГО МОДУЛЯ EV3, ЧАСТЬ 1

Детали для сборки:

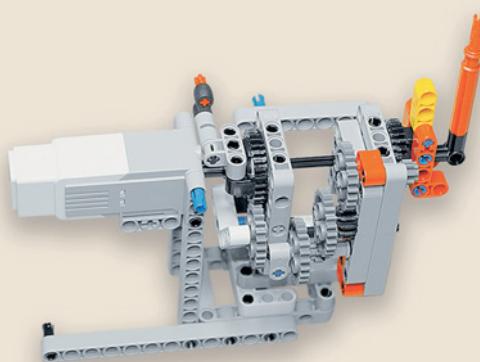
- соединительный штифт, 2-модульный, чёрный, 9x;
- балка № 13, серая, 2x;
- балка № 11, серая, 1x;
- балка № 9, серая, 1x;
- рамка 5 × 11, серая, 1x.



1. Возьми балку № 13 и закреши на ней четыре чёрных штифта в первом, восьмом, десятом и двенадцатом модулях.

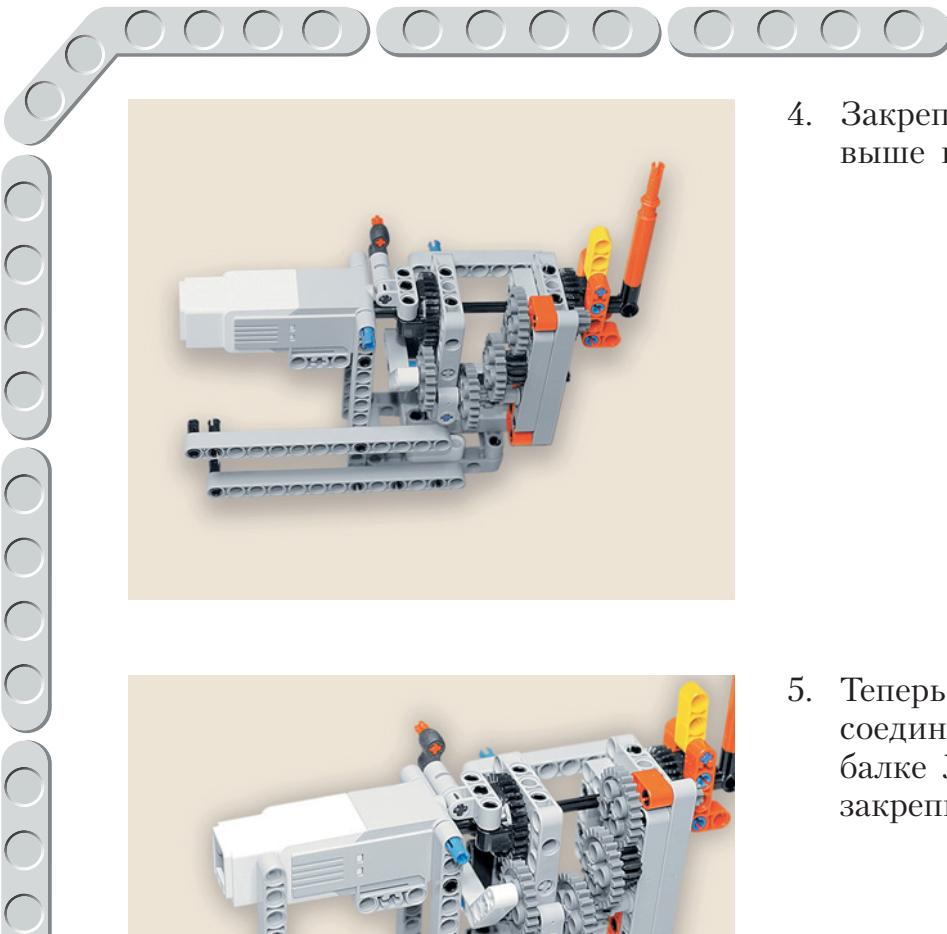


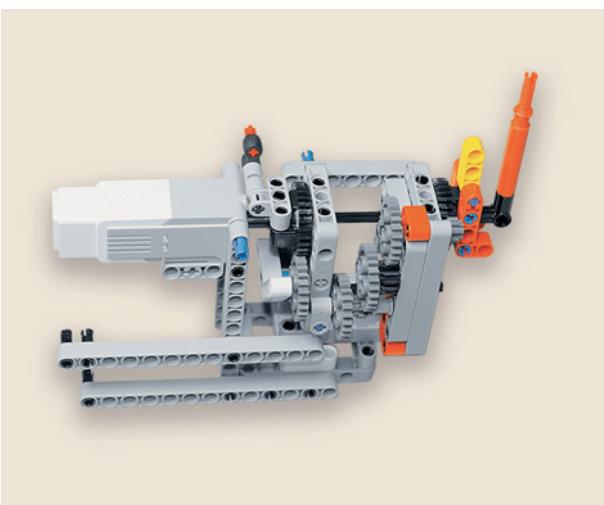
2. После этого вставь собранную балку с правой стороны часовового механизма так, как показано на рисунке. При этом стрелки на часовом механизме должны находиться с правой стороны.



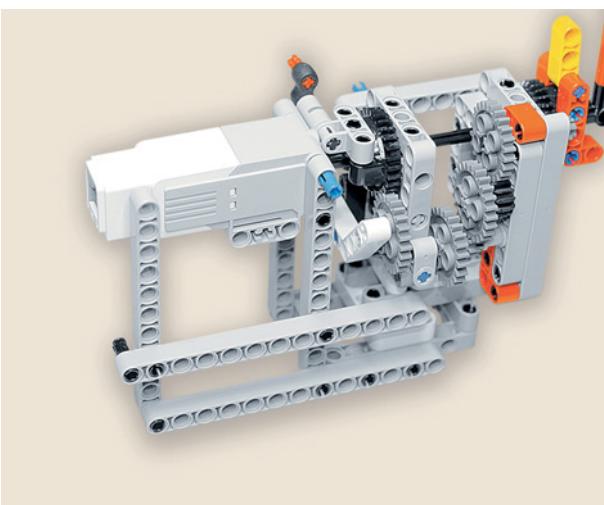
3. Теперь возьми вторую балку № 13 и закреши в её первом, втором и девятом модулях по одному чёрному штифту.



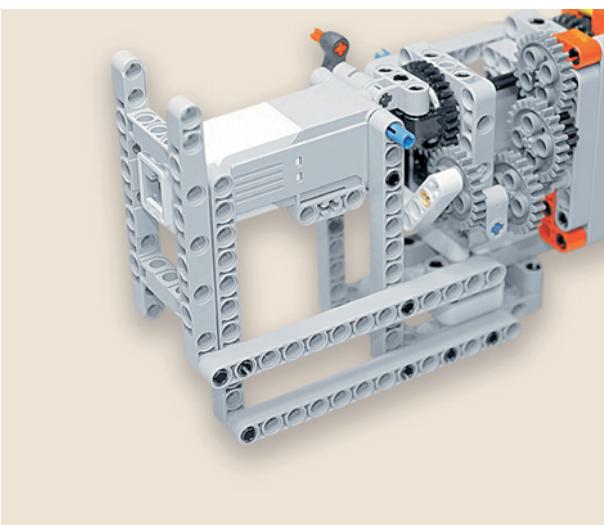
- 
4. Закрепи балку на три модуля выше предыдущей.



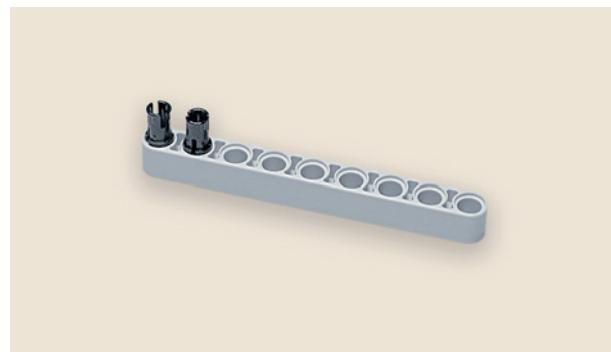
5. Теперь возьми балку № 11 и присоедини её вертикально к нижней балке № 13 нижним модулем, она закрепится и на верхней балке.



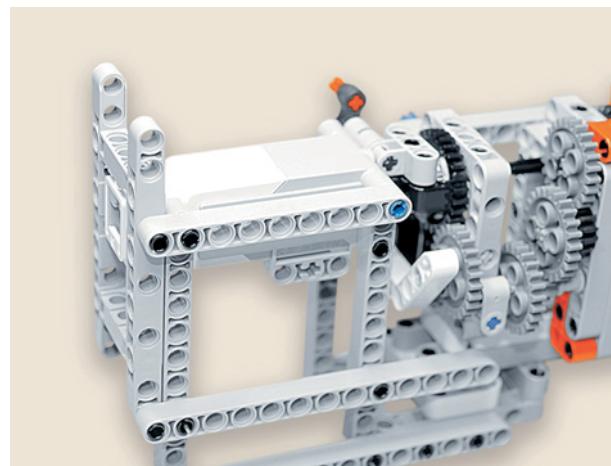
6. Теперь возьми рамку 5×11 и крайним модулем с длинной стороны присоедини ко второй балке в первый модуль.



7. Возьми балку № 9 и вставь в её первый и второй модули по одному чёрному штифту.



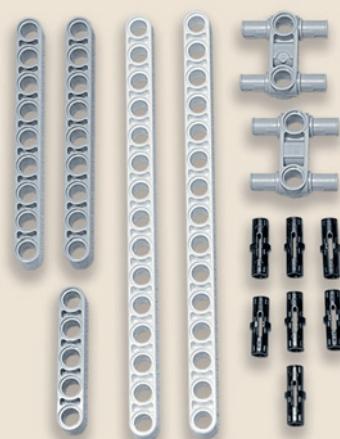
8. Теперь соедини балку № 9 с синим 3-модульным штифтом, находящимся на среднем моторе, а её чёрные штифты вставь в свободные модули слева.



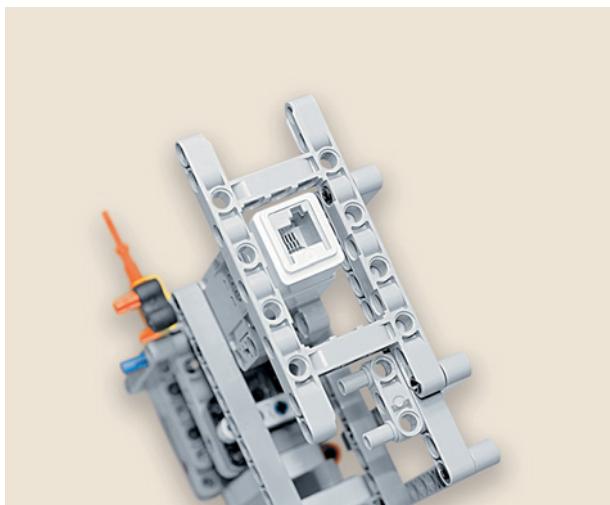
ШАГ 13. СБОРКА КРЕПЛЕНИЯ СРЕДНЕГО МОТОРА И ПРОГРАММИРУЕМОГО МОДУЛЯ EV3, ЧАСТЬ 2

Детали для сборки:

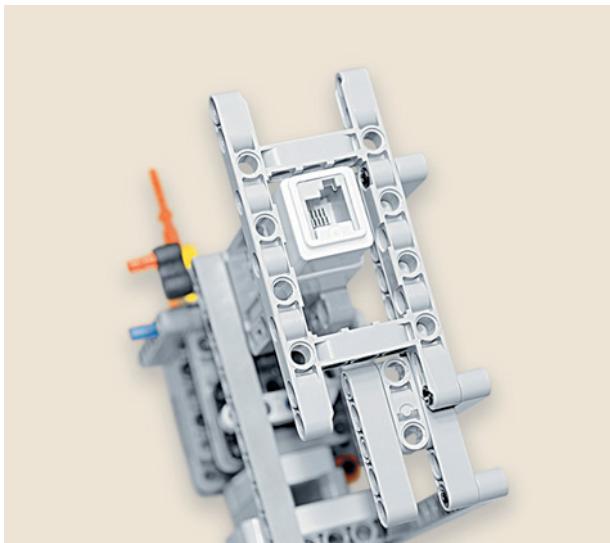
- балка № 15, белая, 2x;
- балка № 9, серая, 2x;
- балка № 5, серая, 1x;
- Н-образный соединительный штифт, серый, 2x;
- соединительный штифт, 2-модульный, чёрный, 7x.



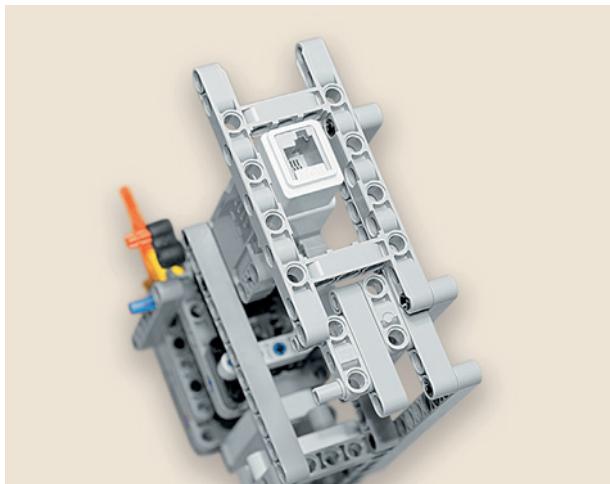
1. Теперь возьми собранный механизм и поверни его разъёмом на среднем моторе к себе. Возьми H-образный соединительный штифт и прикрепи его под рамкой 7×11 к серой балке № 11 в третий и пятый модули снизу.



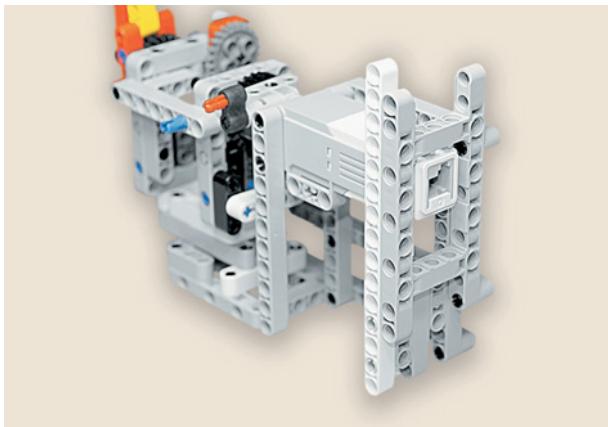
2. Возьми балку № 5 и присоедини её к H-образному штифту в первый и третий модули.



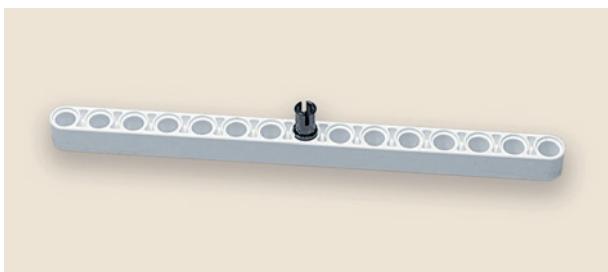
3. Затем возьми второй H-образный штифт и присоедини его к балке № 5, во второй и в четвёртый модули снизу.



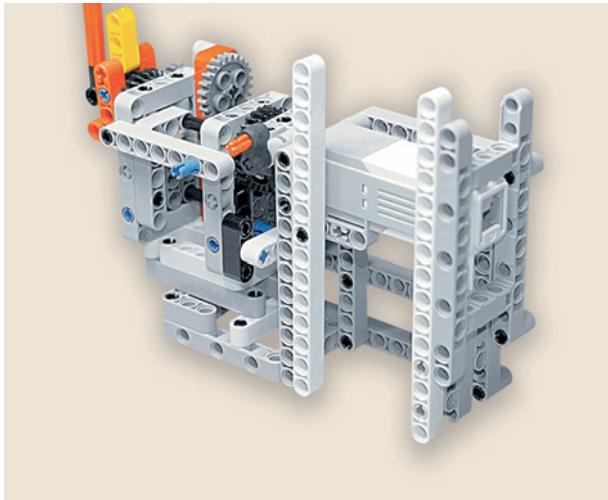
4. Теперь подготовь место для крепления программируемого модуля EV3. Возьми белую балку № 15 и соедини её с Н-образным штифтом во втором и в четвёртом модулях снизу.



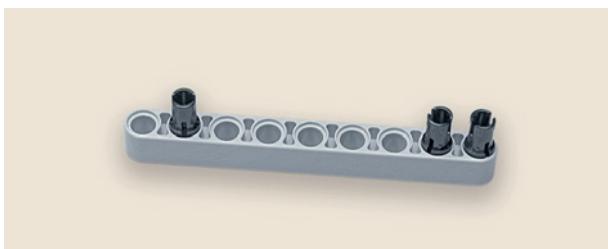
5. Возьми вторую балку № 15 и вставь в восьмой модуль чёрный штифт.



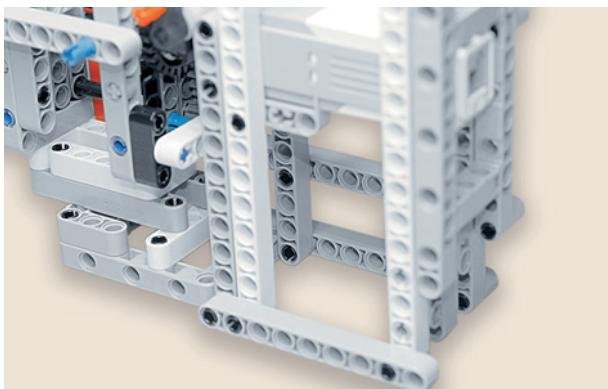
6. Теперь разверни часовой механизм так, чтобы стрелки находились слева. Возьми подготовленную балку № 15 и соедини её с креплением на среднем моторе снизу в левый модуль.



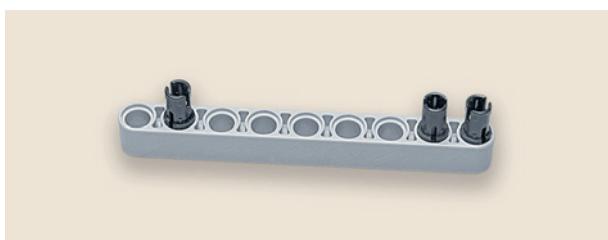
7. Далее возьми балку № 9 и вставь в её второй, восьмой и девятый модули по одному чёрному штифту.



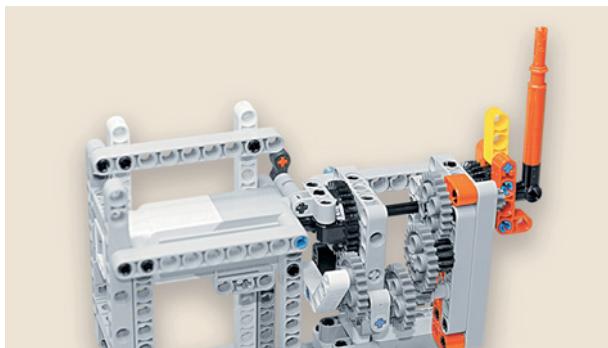
8. Присоедини подготовленную балку № 9 со штифтами к основанию часового механизма, как показано на рисунке.



9. Теперь возьми вторую балку № 9 и вставь в её второй, восьмой и девятый модули по одному чёрному штифту.

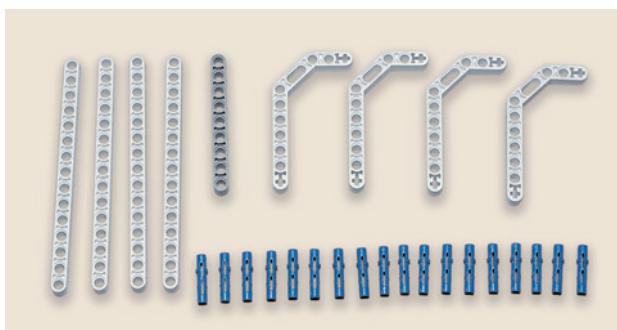


10. Поверни механизм так, чтобы стрелки были справа. Присоедини собранную балку № 9 к механизму над средним мотором к балкам № 15 во вторые модули так, чтобы соединительный штифт на серой балке № 9 совпал с отверстием на рамке 7 × 11 изнутри.



Крепление собрано! Молодец!

ШАГ 14. СБОРКА РАМКИ ДЛЯ ЦИФЕРБЛАТА ЧАСОВ

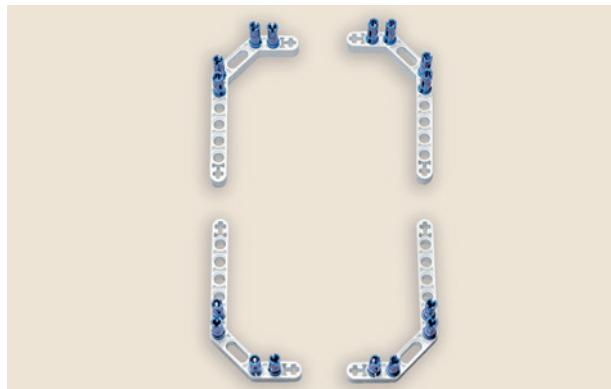


Детали для сборки:

- двойная угловая балка 3 × 7, белая, 4x;
- соединительный штифт, 3-модульный, синий, 18x;
- балка № 15, белая, 4x;
- балка № 9, серая, 1x.



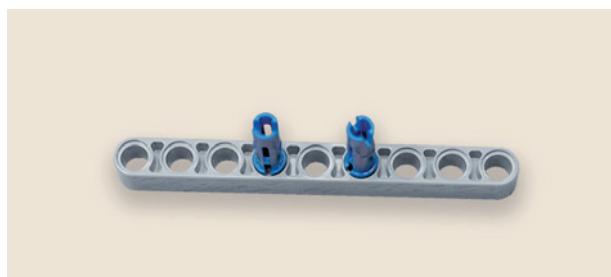
1. Возьми четыре двойные угловые балки 3×7 и вставь в каждую из них по четыре синих соединительных штифта короткими концами в крайние модули от овального отверстия.



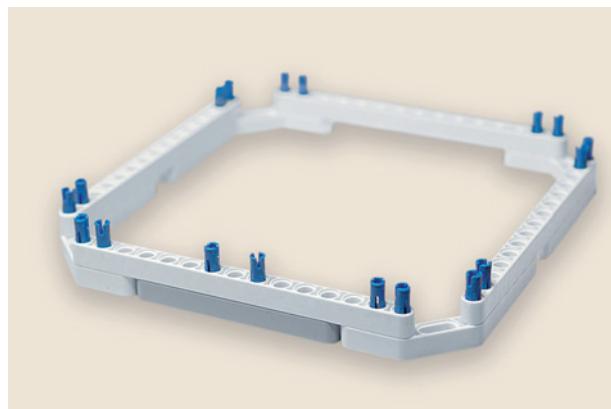
2. Теперь возьми четыре балки № 15 и соедини их крайними модулями со штифтами.



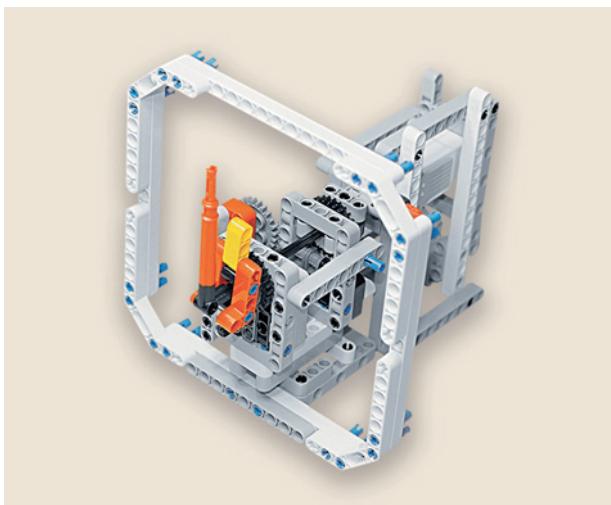
3. Далее возьми балку № 9 и вставь два синих 3-модульных штифта в четвёртый и шестой модули.



4. Соедини собранную балку с одной из сторон рамки снизу.



- Прикрепи собранную конструкцию к основанию часов механизма, как показано на рисунке.



Циферблат готов!

ШАГ 15. СБОРКА ДОМИКА ДЛЯ КУКУШКИ



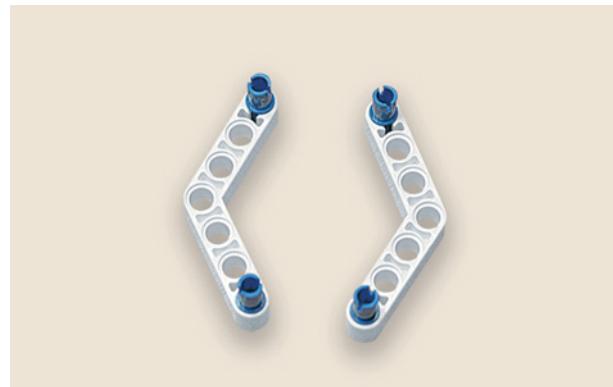
Детали для сборки:

- угловая балка 3×7 , серая, 2x;
- угловая балка 4×4 , белая, 2x;
- соединительный штифт, 2-модульный, синий, 8x.



- Возьми две серые угловые балки 3×7 и вставь в их крестообразные отверстия по одному синему штифту.

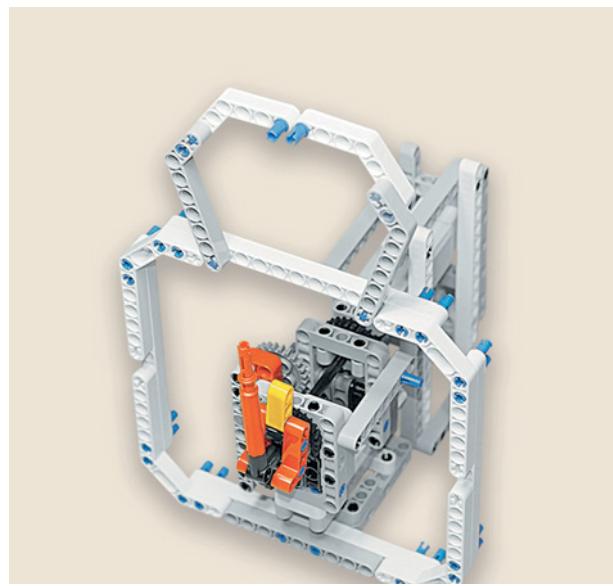
2. Теперь возьми две белые угловые балки 4×4 и тоже вставь в их крестообразные отверстия по одному синему штифту.



3. После этого соедини между собой серые и белые балки.



4. Прикрепи сверху к собранному циферблату свободными концами серых балок с синими штифтами домик для кукушки.



Домик для кукушки готов! Теперь надо собрать дверку.

ШАГ 16. СБОРКА ДВЕРКИ, ЕЁ КРЕПЛЕНИЯ И МЕХАНИЗМА ОТКРЫВАНИЯ



Детали для сборки:

- втулка, жёлтая, 2x;
- ось № 5, серая, 1x;
- поперечный блок, 2-модульный, серый, 2x;
- вилка поперечного блока 2×2, чёрная, 2x;
- соединительный штифт, 2-модульный, серый, 2x;
- соединительный штифт, 2-модульный, чёрный, 2x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт с втулкой, 3-модульный, красный, 2x;
- левая криволинейная панель 5×11, чёрная, 1x;
- правая криволинейная панель 5×11, чёрная, 1x.



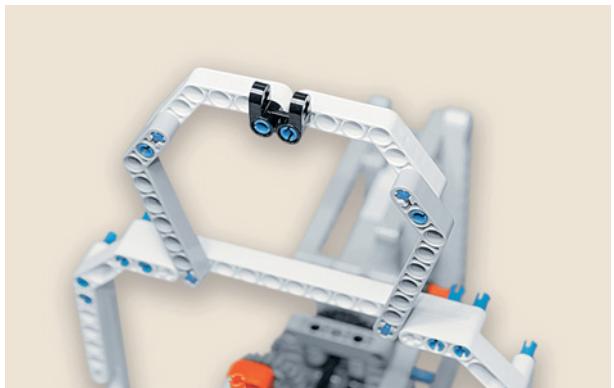
1. Для того чтобы собрать дверку, возьми левую и правую криволинейные панели и установи один серый и чёрный штифты в первый и третий модули сбоку одной из них.

2. Теперь соедини их вместе.



Дверка готова, но для того чтобы закрепить её на рамку для кукушки, тебе надо собрать механизм открывания и крепления.

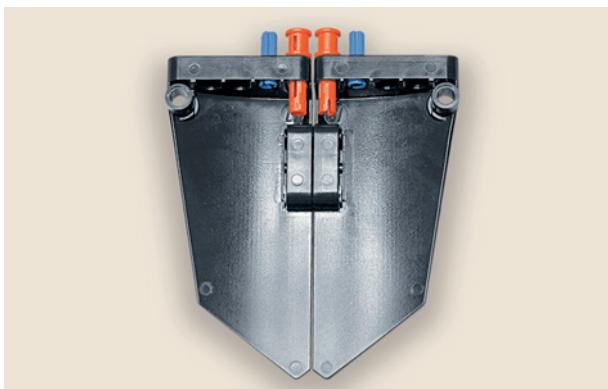
3. Для этого надень на рамке для кукушки сверху на свободные синие соединительные штифты вилку поперечного блока.



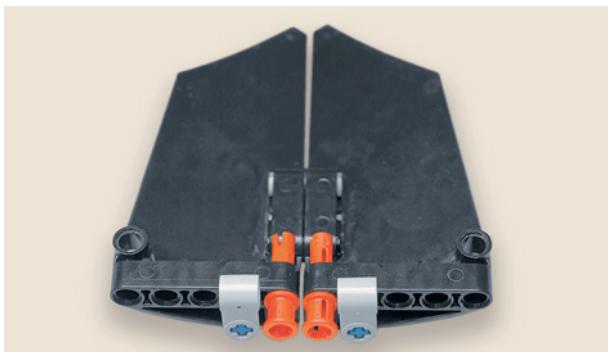
4. Далее сверху на дверке надо закрепить вторую часть механизма крепления. Для этого вставь в два центральных модуля собранной дверки два красных 3-модульных штифта.



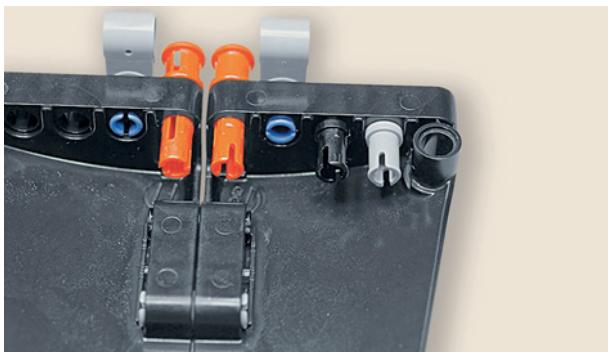
5. Теперь вставь два синих 2-модульных штифта по бокам от красных в соседние модули на дверке.



6. Надень на них два серых попечерных 2-модульных блока.



7. Теперь для управления открыванием дверки тебе необходимо изнутри установить вторую вилку по перечного блока. Для этого вставь один серый и один чёрный соединительные штифты в крайние модули дверки изнутри справа.



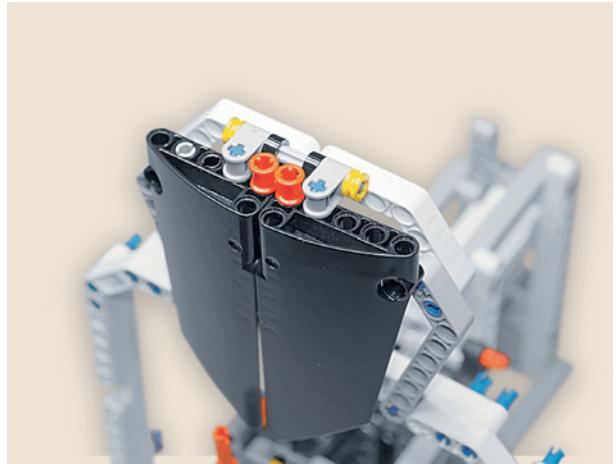
8. Надень на них вилку.



9. Теперь возьми ось № 5 и присоедини дверку к рамке домика кукушки.



10. Зафиксируй концы оси жёлтыми втулками.

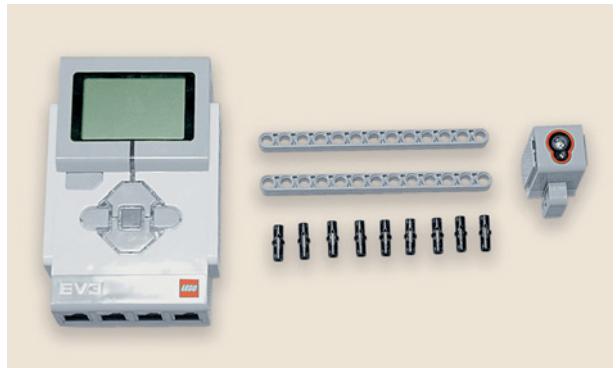


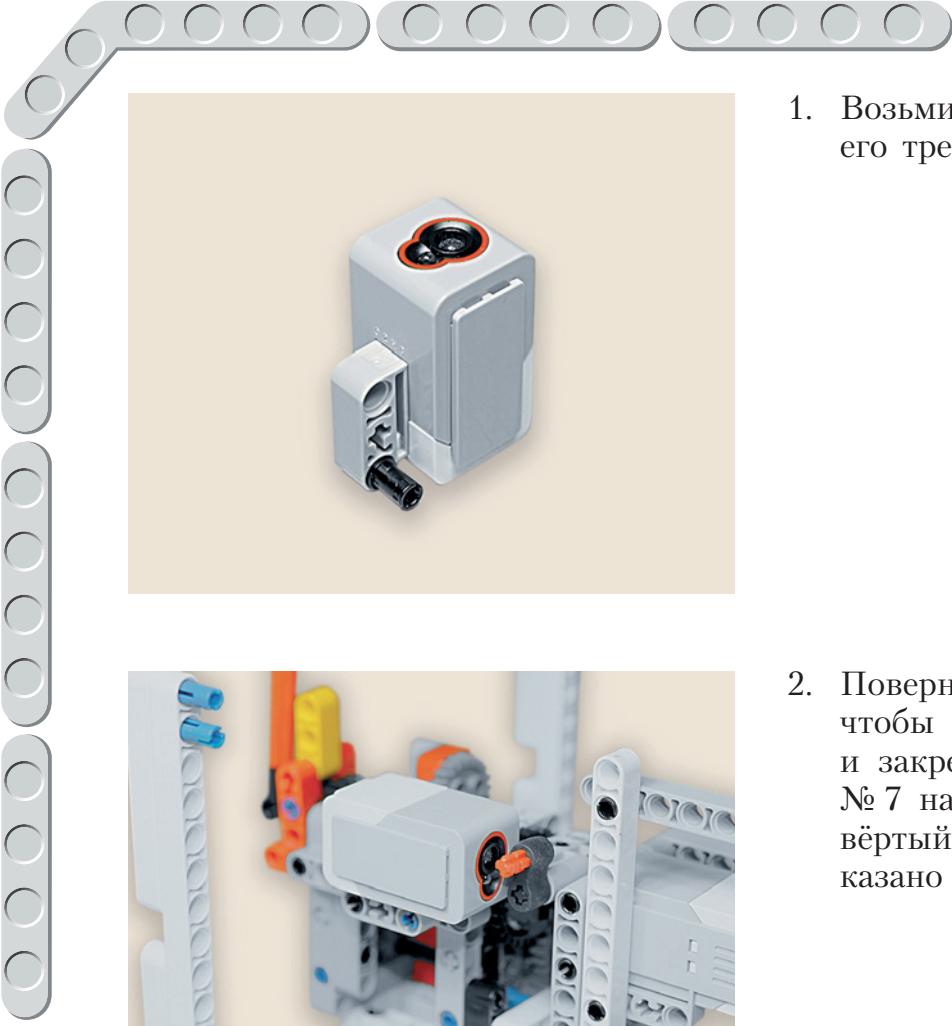
Ты повесил дверку на своё место, но для её автоматического открывания надо собрать управляющий механизм вместе с мотором. А перед этим тебе надо установить датчик цвета и сам модуль EV3.

ШАГ 17. УСТАНОВКА МОДУЛЯ EV3 И ДАТЧИКА ЦВЕТА

Детали для сборки:

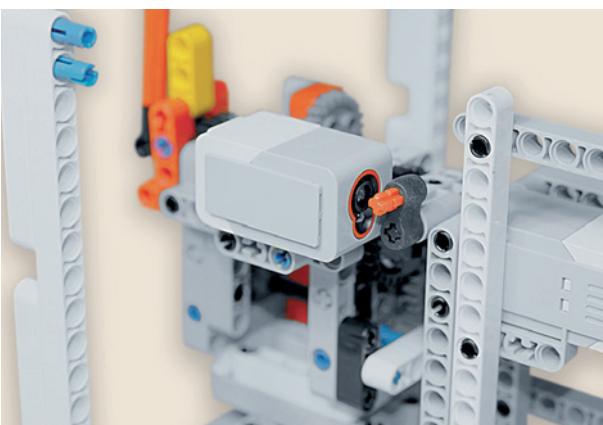
- соединительный штифт, 2-модульный, чёрный, 9x;
- балка № 13, серая, 2x;
- датчик цвета, 1x;
- программируемый модуль EV3, 1x.



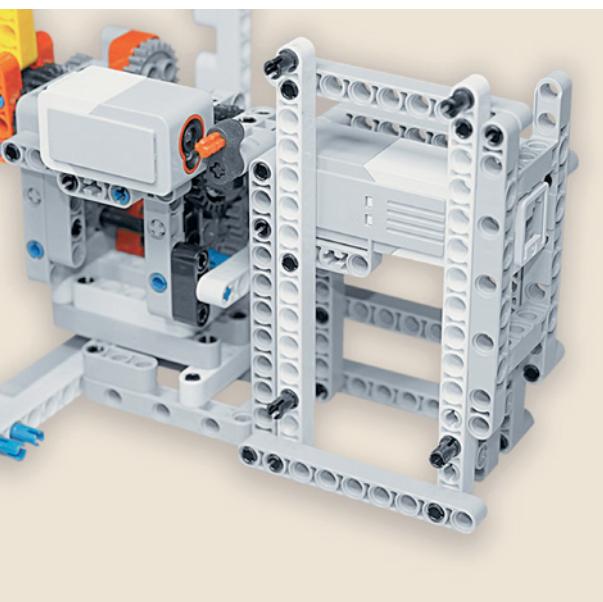
- 
1. Возьми датчик цвета и установи в его третий модуль чёрный штифт.



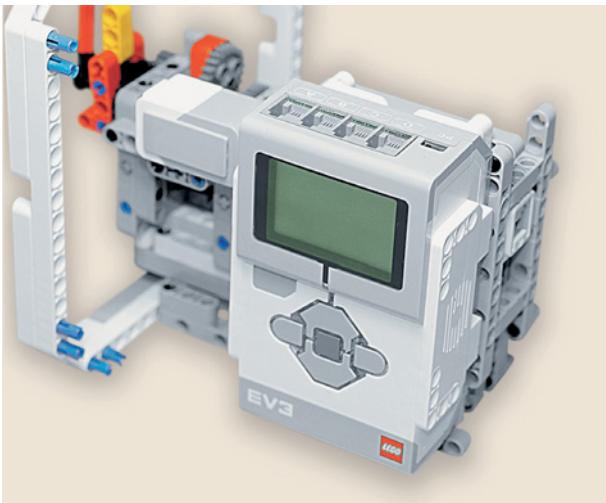
2. Поверни собранный механизм так, чтобы стрелки «смотрели» влево, и закрепи датчик цвета на балку № 7 на часовом механизме в четвёртый и шестой модули, как показано на рисунке.



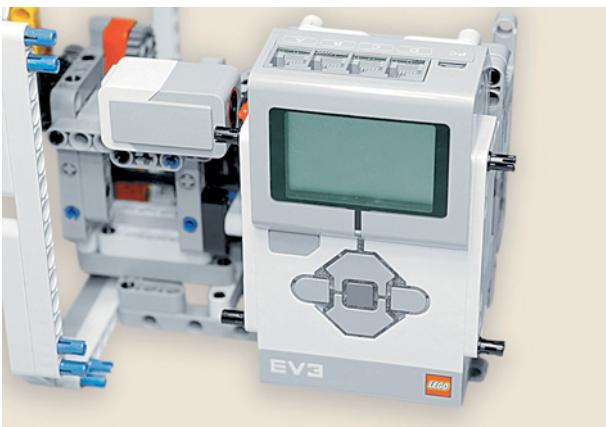
3. Далее тебе надо закрепить программируемый модуль. Для этого установи четыре чёрных штифта в белые балки № 15: два в первые модули и два в тринадцатые модули.



4. Теперь возьми модуль EV3 и установи его на балки № 15.



5. Для устойчивости модуля EV3 закрепи по бокам программируемого модуля по два штифта в верхний и нижний модули с каждой стороны.

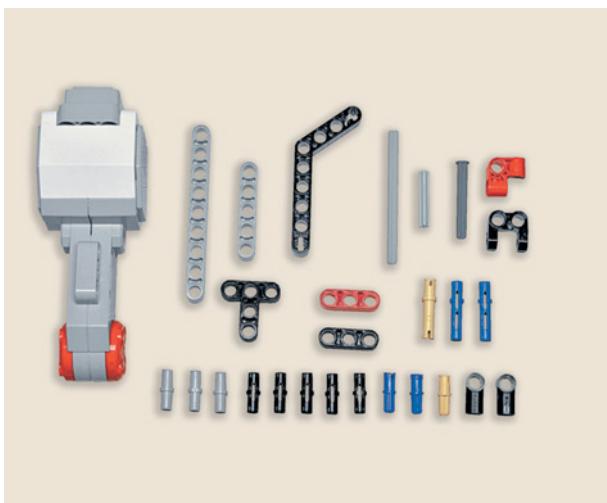


6. Закрепи на установленных штифтах балки № 13.



Осталось собрать механизм открывания дверки кукушки.

ШАГ 18. СБОРКА УПРАВЛЯЮЩЕГО МЕХАНИЗМА ДЛЯ КУКУШКИ



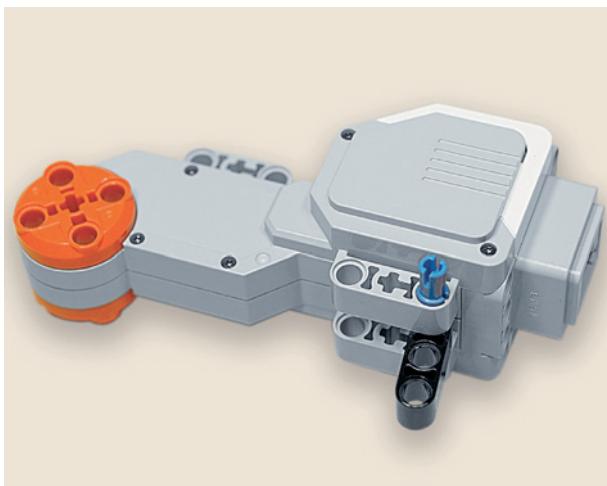
Детали для сборки:

- большой мотор, 1x;
- балка № 9, серая, 1x;
- балка № 5, серая, 1x;
- балка № 3, красная, 1x;
- балка № 3, чёрная, 1x;
- угловая балка 4 × 6, чёрная, 1x;



- поперечная балка 2 × 1, красная, 1x;
- вилка поперечного блока 2 × 2, чёрная, 1x;
- тавровая балка 3 × 3, чёрная, 1x;
- прямой блок № 1, чёрный, 2x;
- ось № 3, серая, 1x;
- ось № 7, серая, 1x;
- ось № 4 с фиксирующим элементом, серая, 1x;
- соединительный штифт, 2-модульный, чёрный, 5x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт, 2-модульный, серый, 3x;
- соединительный штифт, 3-модульный, синий, 2x;
- соединительный штифт, 3-модульный, бежевый, 1x;
- соединительный штифт, 2-модульный, бежевый, 1x.

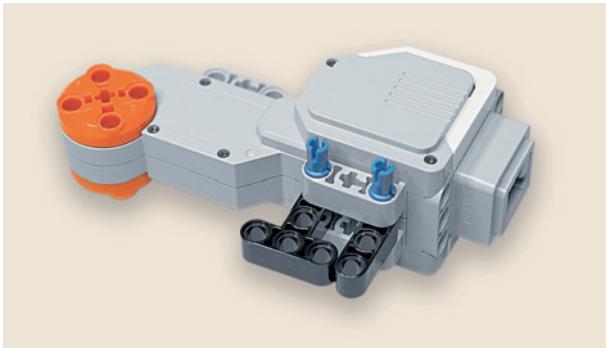
Для того чтобы автоматически открывалась дверка и выезжала кукушка, тебе необходимо собрать и установить привод механизма.



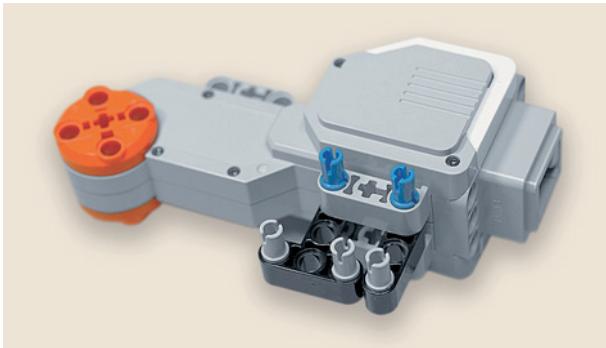
1. Возьми большой мотор, вложи между модулями под разъёмом чёрную балку № 3 напротив первого модуля и закреши её с помощью синего 3-модульного штифта, продев его длинной стороной сквозь балку.



2. То же самое сделай с тавровой балкой 3×3 , расположив её между третьими модулями.



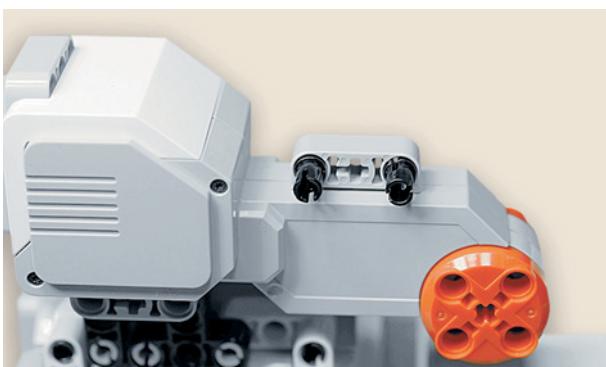
3. Вставь в установленных балках со стороны, где выступают синие штифты, три серых штифта.



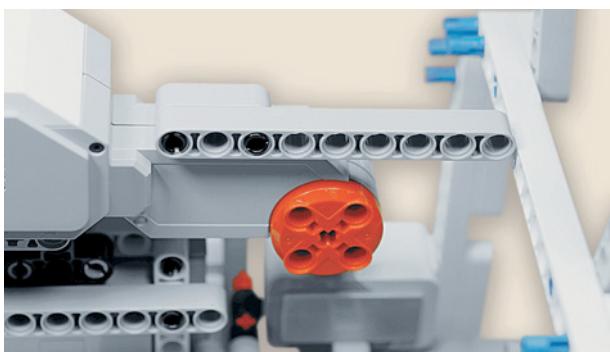
4. Установи мотор, закрепив его на программируемом модуле сзади.



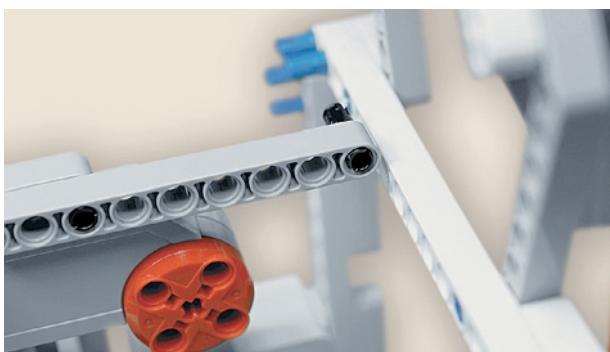
5. Далее установи на моторе сверху в первый и третий модули по чёрному штифту.



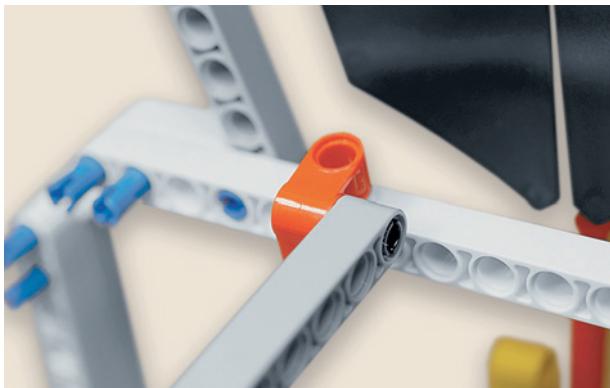
6. На них закреши балку № 9.



7. Вставь в девятый модуль на этой балке чёрный штифт.



8. Присоедини красную поперечную балку 2×1 , как показано на рисунке.



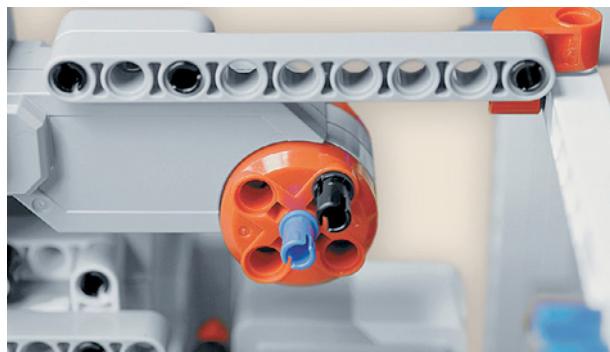
9. Вставь в красную поперечную балку с обратной стороны ещё один чёрный штифт, а затем прикрепи получившуюся конструкцию к рамке в шестой модуль балки № 15 сверху циферблата.



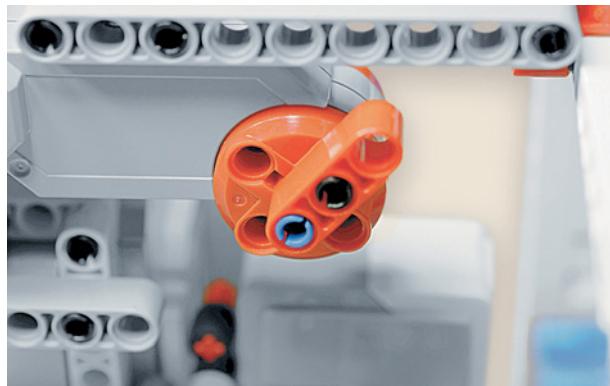
10. Теперь установи в крестообразное отверстие привода мотора справа синий 2-модульный штифт.



11. Вставь в отверстие рядом чёрный штифт.



12. Установи на эти штифты красную балку № 3.



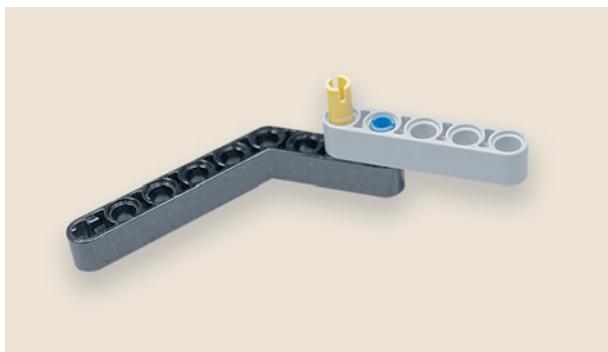
13. Вставь в оставшийся свободный модуль на балке № 3 бежевый 2-модульный штифт.



14. Далее возьми угловую чёрную балку 4×6 и вставь синий 2-модульный штифт в крестообразное отверстие с малой стороны, а рядом с ним — бежевый 3-модульный штифт короткой стороной.



15. Надень на эти штифты балку № 5 первым и вторым модулями.



16. Теперь возьми ось № 7 и по бокам надень прямые чёрные блоки так, чтобы они «смотрели» в одну сторону.



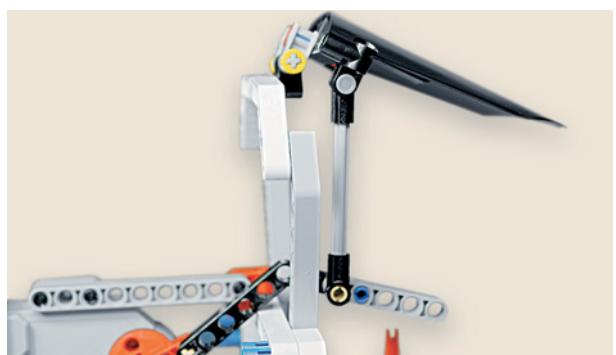
17. Скрепи получившиеся детали, надев чёрный блок на выступающий бежевый штифт.



18. Собранную конструкцию закреши осью № 4 с фиксирующим элементом на дверке, как показано на рисунке.



19. Закреши свободный конец угловой чёрной балки на бежевом штифте мотора.

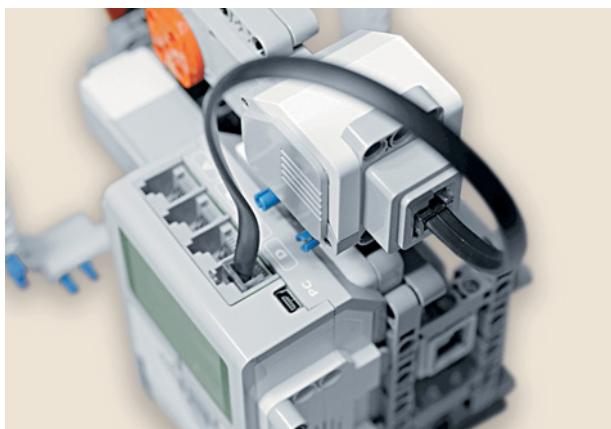


20. Чтобы механизм двигался плавно и не стопорился, возьми вилку поперечного блока и установи её в девятый модуль балки № 15 сверху циферблата — закреши её осью № 3, продев сквозь крестообразные отверстия.



ШАГ 19. ПОДКЛЮЧЕНИЕ МОТОРОВ И ДАТЧИКА

Подключи с помощью трёх кабелей 25 см:



1) большой мотор к порту D;

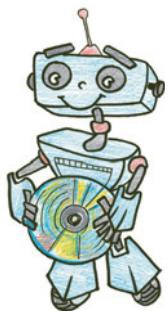


2) средний мотор к порту A;



3) датчик цвета к порту 4.





Этап 3. Установка программного обеспечения на компьютере

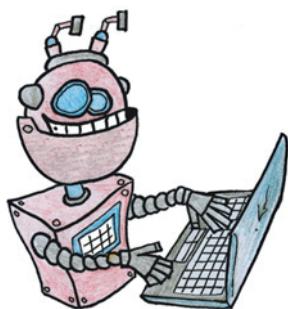
1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт [http://Education.
LEGO.com](http://Education.LEGO.com) и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

Этап 4.

Создание программы для роботизированных часов с кукушкой



ЛОГИКА ПРОГРАММЫ

Программа будет состоять из двух одновременно выполняющихся подпрограмм: одна будет отвечать за счёт времени и движение стрелок, а вторая контролировать наступление нового часа и поведение кукушки.

Кстати!

В данной программе будет несколько интересных алгоритмических конструкций и приёмов программирования. Один из них – это параллельное выполнение нескольких программ, а также использование разных видов циклов: бесконечный и с заданным числом повторений, которые образуют конструкцию вложенных циклов (один цикл в другом).

Стрелками управляет средний мотор, который подсоединен к конструкции из нескольких шестерёнок (редукторов) так, чтобы один оборот мотора соответствовал одному полному обороту минутной стрелки и $\frac{1}{12}$ оборота часовой. Тогда в программе средний мотор будет вращаться на 6 градусов – общепринятую единицу измерения углов на плоскости (почему именно на 6, мы разберёмся немного позже) и ждать, когда пройдёт 60 секунд, после чего сдвинется ещё на 6 градусов и так далее.

Кукушкой управляют датчик цвета и большой мотор. Как только перед датчиком цвета пройдёт красная ось механизма боя часов, в работу включится большой мотор, выдвигая кукушку из домика. Затем кукушка будет подавать звук столько раз, сколько часов показывают стрелки на циферблате, и после прятаться обратно в домик до следующего часа. Для правильной работы кукушки нам понадобится считать количество часов, которые мы будем записывать в переменную.

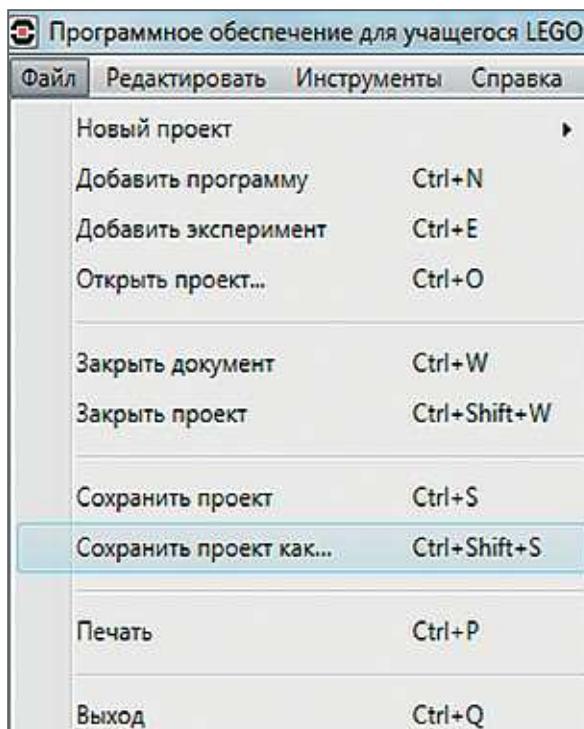
ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, не нужно искать, где он записан на компьютере.

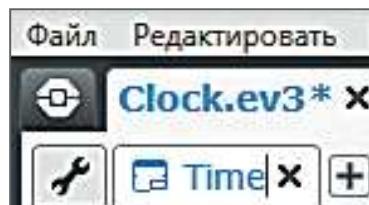
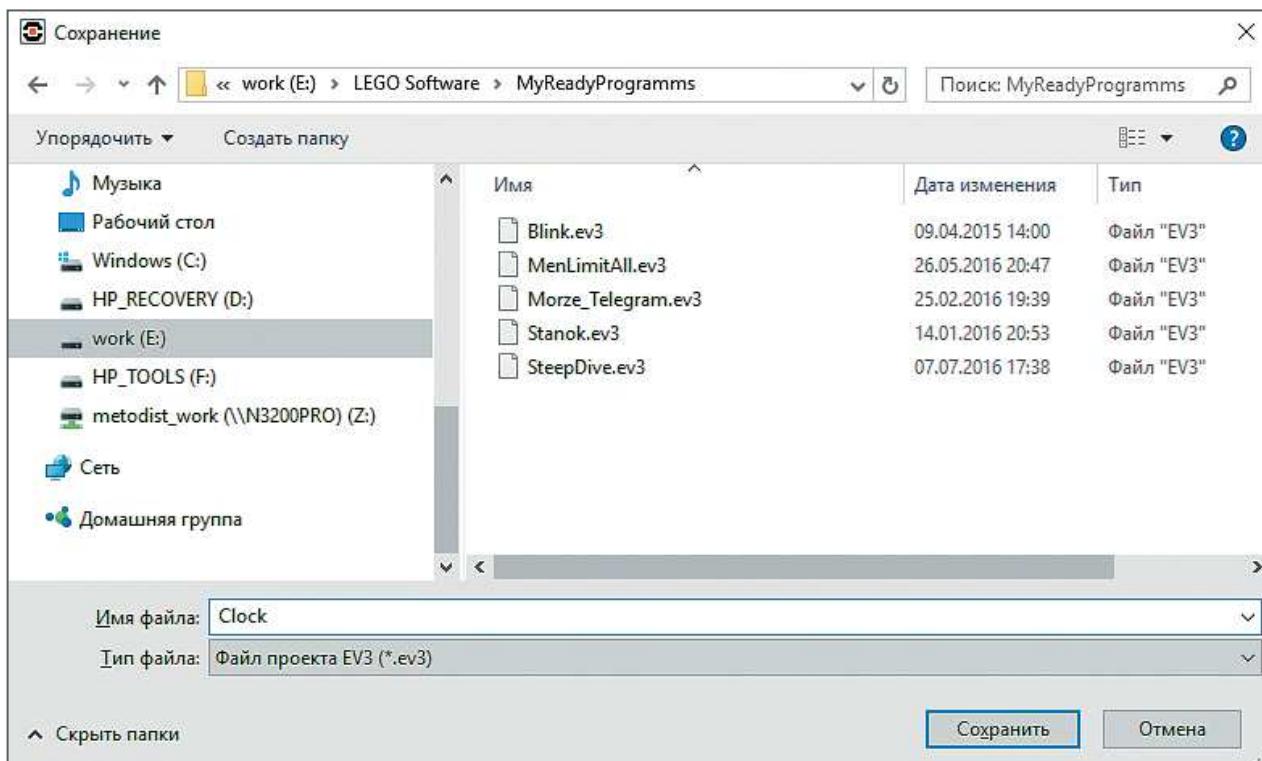
1. Для начала сохрани свой проект в памяти компьютера, чтобы избежать его потери. Для этого выбери меню **Файл** в левом верхнем углу окна программы и выбери в контекстном меню пункт **Сохранить проект как...**



2. В открывшемся окне выбери удобное расположение для всех файлов твоей программы, затем задай **Имя файла:** *Clock* (в переводе с английского — часы) в нижней части окна и нажми **Сохранить**. Именно так будет называться папка в памяти программируемого модуля EV3, в которой будут храниться все программы, которые ты напишешь в рамках этого или других проектов.

Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они пишут, например, каждое слово с большой буквы или используют нижнее подчёркивание.

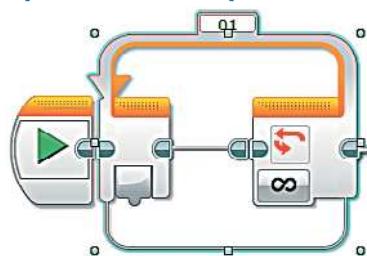


3. Осталось дать имя самой программе (в нашем проекте она будет одна). Для этого дважды кликни по имени **Program** в левом верхнем углу поля программы, задай новое имя: *Time* (в переводе с английского — время) и нажми клавишу **Enter**.

СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОТИЗИРОВАННЫХ ЧАСОВ С КУКУШКОЙ

В открывшемся окне проекта начинай составлять программу для роботизированных часов.

Программа 1. Время

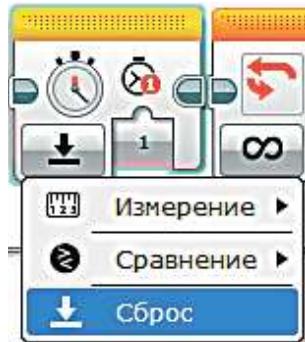


1. Для начала добавь бесконечный **Цикл** (оранжевый блок), так как счёт времени будет идти непрерывно. Вся остальная часть этой программы будет содержаться внутри этого цикла.

2. Далее добавь команду **Таймер** (жёлтый блок).



3. С помощью этого таймера часы будут понимать, что пора сдвинуть стрелки на одну минуту. В начале работы показания таймера необходимо обнулить, поэтому выбери для него опцию **Сброс**.



После того как таймер сброшен, необходимо вращать средний мотор и затем подождать 60 секунд. Давай разберёмся, почему мотор будет вращаться всего на 6 градусов.

Дело в том, что угол полного оборота любого объекта по некоторой окружности всегда равен 360 градусов.

Для часов полный оборот минутной стрелки по окружности циферблата равен одному часу или 60 минутам. Другими словами, циферблат можно условно разделить на 60 равных частей (на многих часах именно так и сделано).

Тогда рассчитаем, на сколько градусов должна отклониться минутная стрелка, чтобы показать одну минуту.

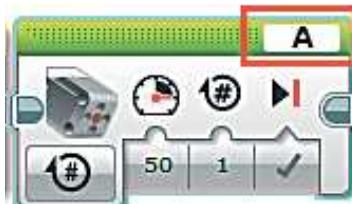
$$\frac{360 \text{ градусов}}{60 \text{ минут}} = \frac{6 \text{ градусов}}{1 \text{ минута}}$$

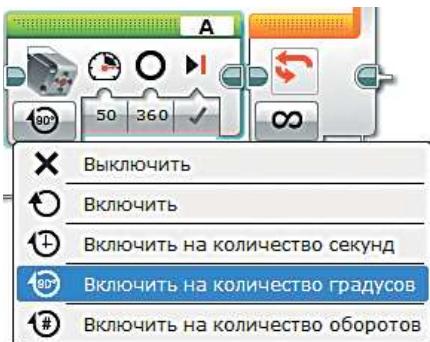
Таким образом, мы получаем соотношение, что 6 градусов укладывается в одной минуте. Соответственно, и вращаться мотор будет на 6 градусов один раз в минуту.

Продолжим написание программы.

4. Далее добавь команду **Средний мотор** (зелёный блок).

Проверь, что в правом верхнем окошке указан порт A!





5. Выбери опцию **Включить на количество градусов**.



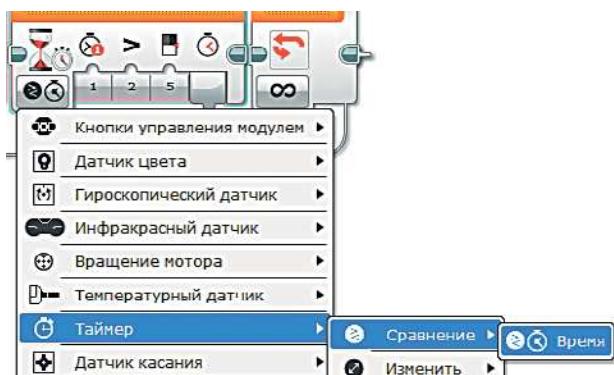
6. Задай параметр **Градусы** равным 6.



7. Далее добавь команду **Поддерживать в активном состоянии** (синий блок), чтобы таймер автоматического выключения EV3 сбрасывался каждую минуту и часы продолжали функционировать.



8. Последним шагом в этой программе будет задержка на 60 секунд. Для её установки добавь команду **Ожидание** (оранжевый блок).



9. Выбери опцию **Таймер → Сравнение → Время**.

10. Настрой параметры этой команды:

Тип сравнения: Больше или равно (3);

Пороговое значение: 60.



Кстати!

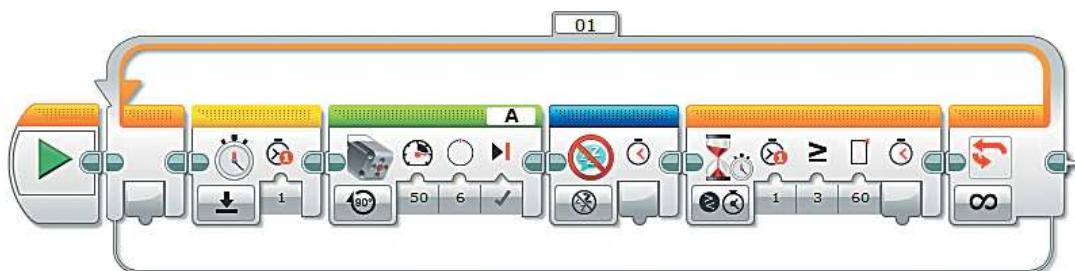
Мы могли бы сделать всё немного проще, без использования таймера. Для этого можно было бы просто выставить в команде Ожидание время задержки, равное 60. Но в связи с тем что до неё тоже есть команды, на исполнение которых тратится время (хоть и очень-очень маленькое), мы бы получали растущую с каждым разом погрешность. С помощью таймера её удается снизить практически до нуля.

Молодец! Ты запрограммировал работу часов! Теперь остаётся оживить кукушку!

Программа 2. Кукушка

Как мы и говорили, будут одновременно выполняться две программы. Поэтому сейчас ты добавишь ещё одно начало программы.

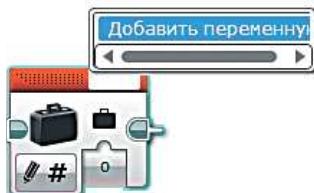
- Добавь команду **Начало** (оранжевый блок) чуть ниже первой программы.



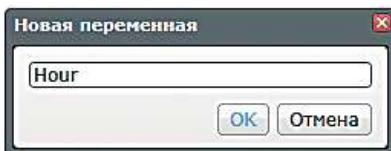
Вся остальная программа будет прописываться в этой ветви.



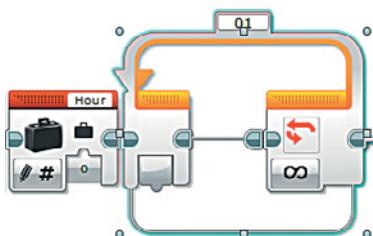
2. Как мы обсуждали в логике программы, нам понадобится считать количество часов, чтобы кукушка издавала звук нужное количество раз. Это значение мы будем хранить в переменной. Для этого добавь команду **Переменная (красный блок)**.



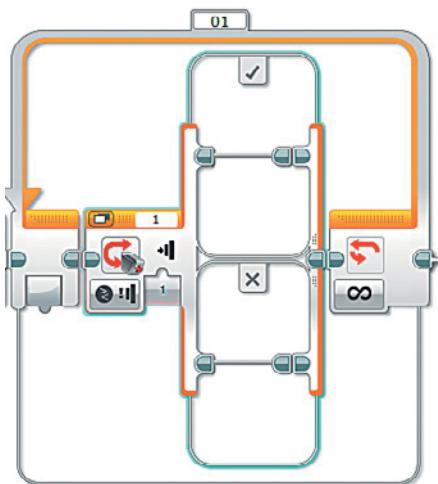
3. Выбери в правом верхнем углу пункт **Добавить переменную**.



4. Введи в открывшемся окне имя переменной **Hour** (в переводе с английского – час) и нажми **OK**.

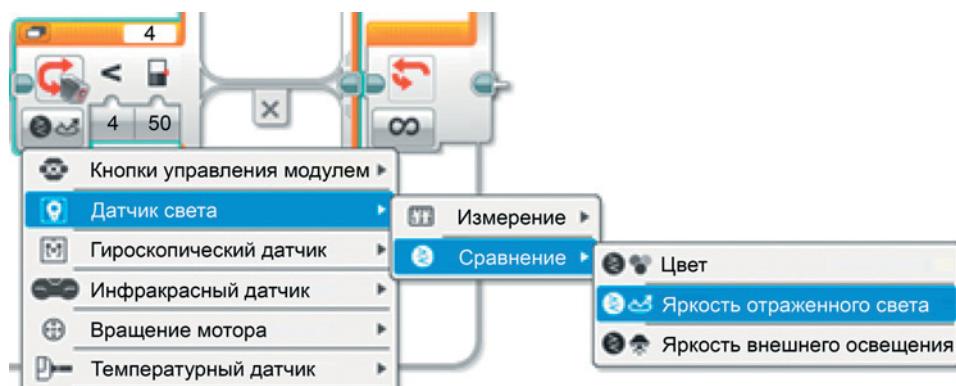


5. Далее добавь команду **Цикл (оранжевый блок)**. Остальная часть программы будет записана внутри этого цикла.



6. Теперь добавь команду **Переключатель (оранжевый блок)**.

7. В этом переключателе мы будем проверять, проходила ли мимо датчика цвета красная ось механизма боя часов. Поэтому в нём выбери опцию **Датчик цвета** → **Сравнение** → **Яркость отражённого света**.

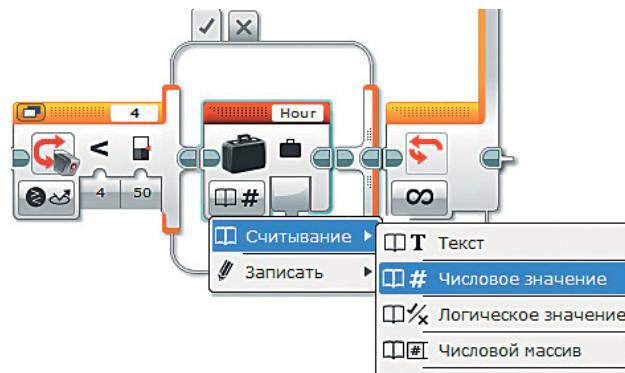


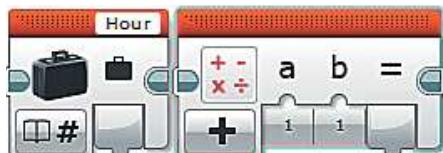
8. Для удобства работы с переключателем нажми на значок в левом верхнем углу **Переключиться на вид с вкладками**.



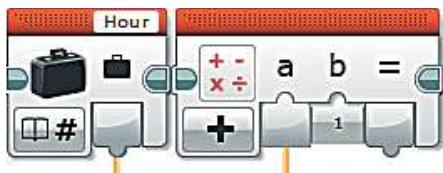
В данном переключателе мы будем проверять, понижалась ли яркость отражённого света, то есть появлялся ли какой-то объект перед датчиком цвета. Соответственно, нас будут интересовать только действия, когда проверяемое условие выполнено, то есть вкладка, помеченная «галочкой».

9. Изменять начальные параметры проверки условия (яркость меньше 50) мы не будем, этого достаточно. Добавь внутрь переключателя команду **Переменная (красный блок)** и выбери опцию **Считывание** → **Числовое значение**.



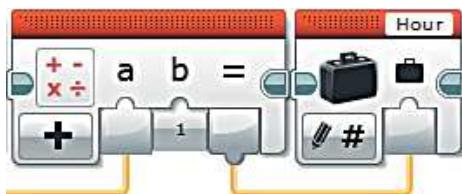


10. Далее добавь команду **Математика** (красный блок).



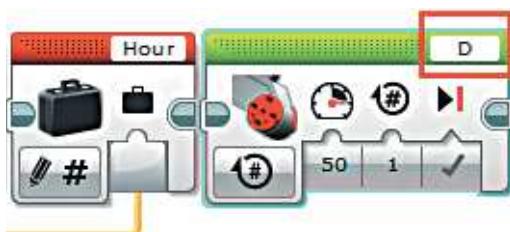
11. Соедини выход переменной **Hour** с входом **a** команды **Математика**, а во входе **b** оставь **1**.

Таким образом, как только прошёл один час, мы его посчитали, добавив единицу к переменной **Hour**.



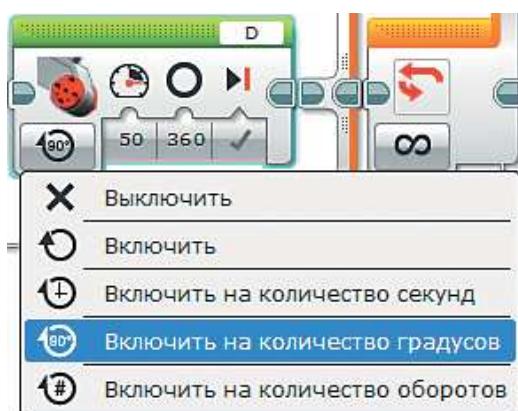
12. Добавь ещё одну команду **Переменная** (красный блок) и соедини выход «**=**» команды **Математика** с входом переменной **Hour**.

13. Теперь мы готовы «оживить» кукашку. Для начала она должна «вылезти» из своего домика. Для этого добавь команду **Большой мотор** (зелёный блок).



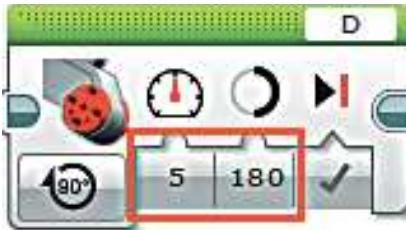
Внимание!

Проверь, чтобы в правом верхнем окошке значилась буква **D** – именно к этому порту ты подключил большой мотор.

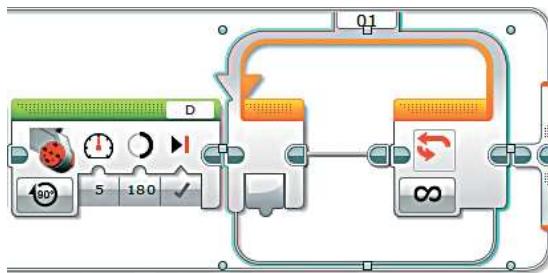


14. Далее выбери опцию **Включить на количество градусов**.

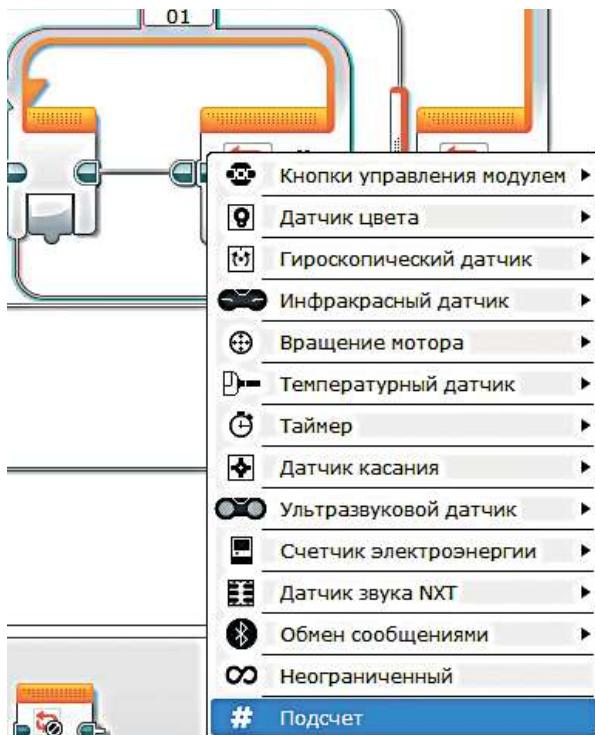
15. Теперь настрой параметры этой команды:
Мощность: 5;
Градусы: 180.



16. Теперь кукушке пора «куковать»!
Добавь команду **Цикл** (оранжевый блок).

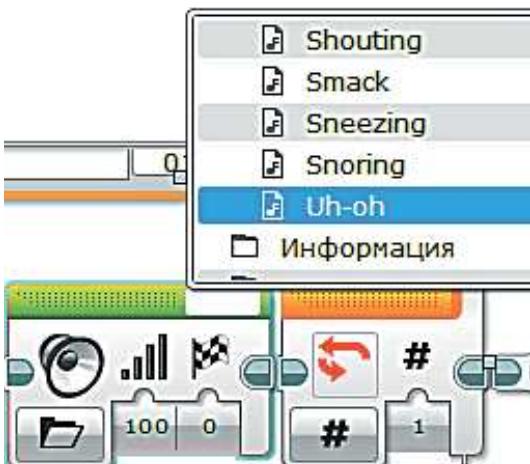


17. В отличие от уже используемых, этот цикл не должен быть бесконечным, наоборот, он должен работать столько раз, сколько записано в переменной **Hour**. Поэтому выбери для этого цикла опцию **Подсчёт**.

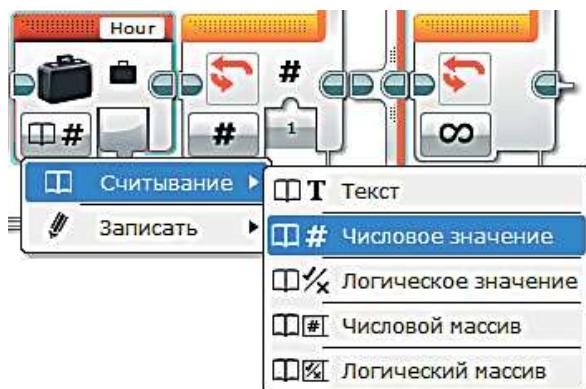


18. Теперь добавь внутрь этого цикла команду **Звук** (зелёный блок).

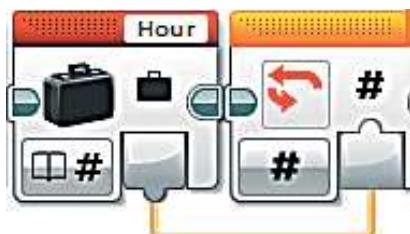




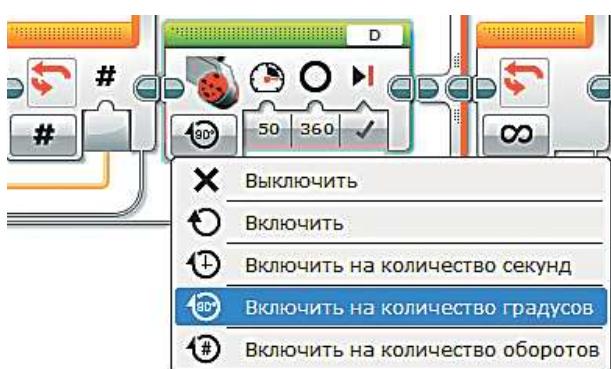
19. В правом верхнем окошке теперь надо выбрать звуковой файл, который будет имитировать «кукование». Например, **Звуковые файлы LEGO** → **Выражения** → → Uh-Oh.



20. В этом цикле осталось только настроить количество повторений. Для этого добавь команду **Переменная** (красный блок) и выбери опцию **Считывание** → **Числовое значение**.



21. Соедини выход переменной **Hour** с входом параметра повторов цикла.



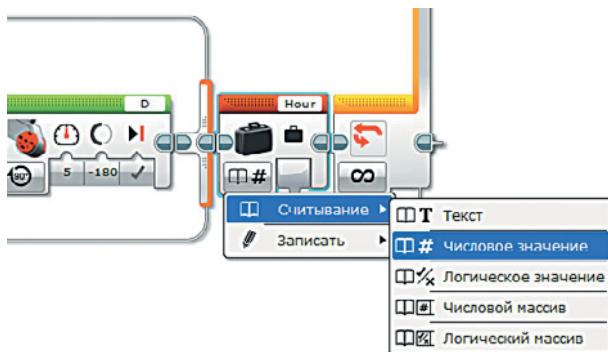
22. После того как цикл «кукования» отработал, кукушка должна спрятаться обратно. Для этого добавь ещё одну команду **Большой мотор** (зелёный блок) и выбери опцию **Включить на количество градусов**.

23. И настрой параметры команды на обратный ход.
Мощность: 5;
Градусы: -180.

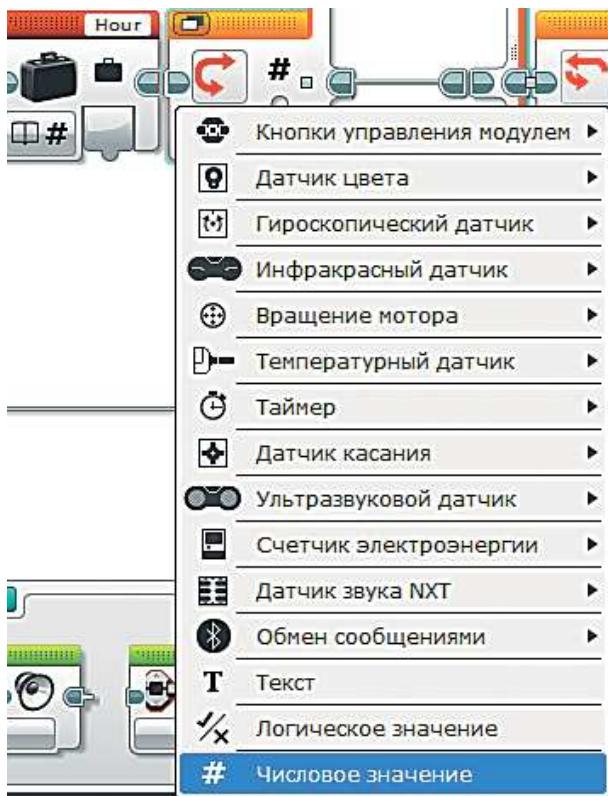


Казалось бы, на этом всё. Но нужно учитывать тот факт, что полный оборот часовой стрелки по циферблату насчитает 12 часов, после чего должен начать заново. В нашей программе пока что это не так — мы не обнуляли переменную по достижении ею значения 12. Сделаем это.

24. После переключателя добавь команду **Переменная** (красный блок) и выбери опцию **Считывание** → → **Числовое значение**.



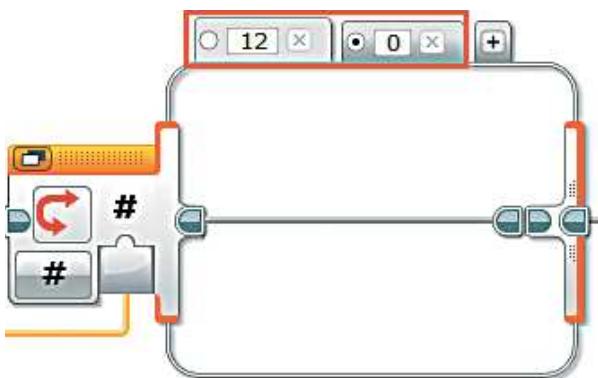
25. Далее добавь команду **Переключатель** (оранжевый блок), перейди к виду с вкладками и выбери опцию **Числовое значение**.



26. Соедини выход переменной **Hour** с входом переключателя.

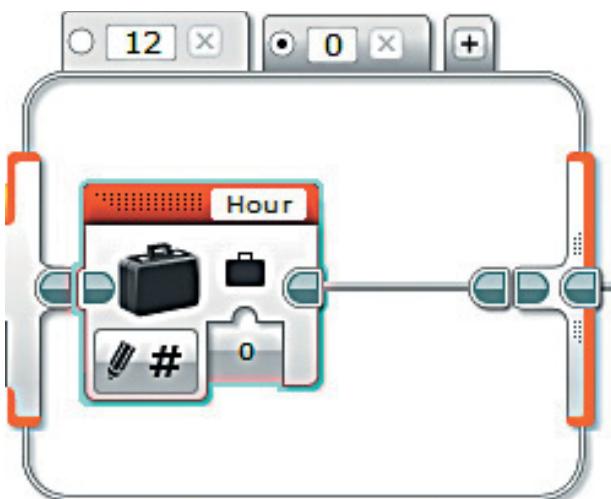


27. Сверху переключателя в первую вкладку введи с клавиатуры наше контрольное число **12**, во второй вкладке оставь **0** и поставь **точку** возле него — это будет указывать на вариант по умолчанию (то есть те действия, когда **Hour** **НЕ** равна **12**).



Внимание!

Для дальнейшего написания программы вкладка с числом **12** должна быть выделена. Именно для этого значения переменной **Hour** надо написать все действия.

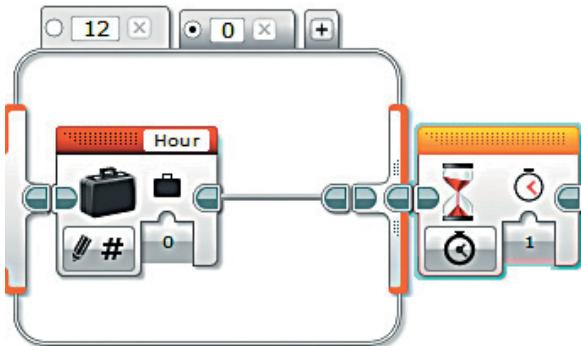


28. Добавь внутрь переключателя команду **Переменная** (красный блок).

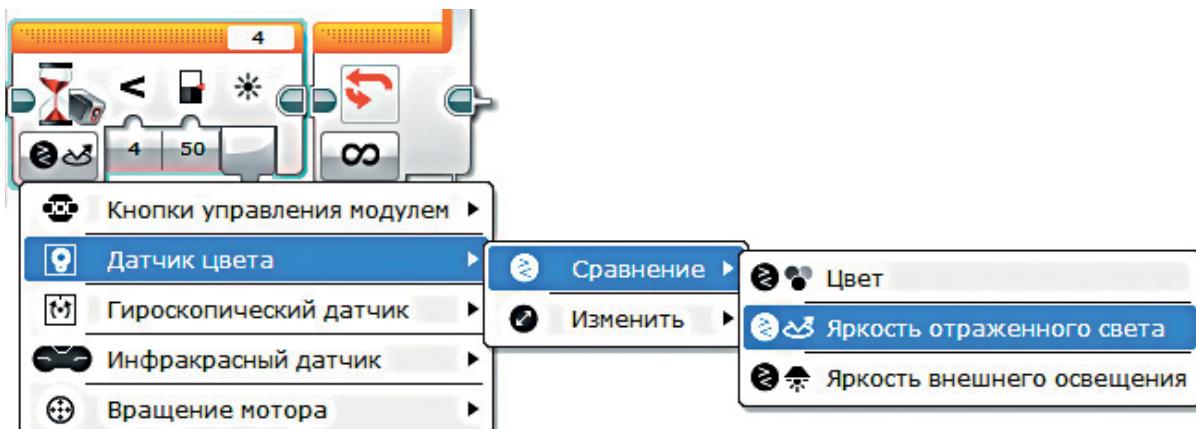
Её параметры изменять не нужно — эта команда обнулит переменную **Hour** по достижении ею значения **12**.

Остался последний шаг. Дело в том, что механизм боя часов не сразу «ходит» из поля зрения датчика цвета. Поэтому перед тем, как общий цикл пойдёт на повтор, необходимо добавить ожидание того, чтобы яркость отражённого света для датчика снова увеличилась.

29. После переключателя добавь команду **Ожидание** (оранжевый блок).



30. Выбери опцию **Датчик цвета** → **Сравнение** → **Яркость отражённого света**. Проверь, чтобы в правом верхнем окошке значилась цифра 4 — именно к этому порту ты подключил датчик цвета!



31. Настрой параметры этой команды.

Тип сравнения: Больше (2);

Пороговое значение: 70.

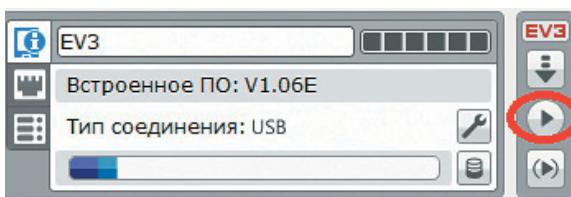
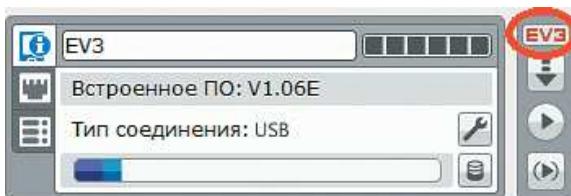


Поздравляем! Твоя программа готова к первому пуску!
Бой часов раздастся вскоре!

Этап 5. Загрузка программы и её тестирование



ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ



Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

ТЕСТИРОВАНИЕ ПРОГРАММЫ

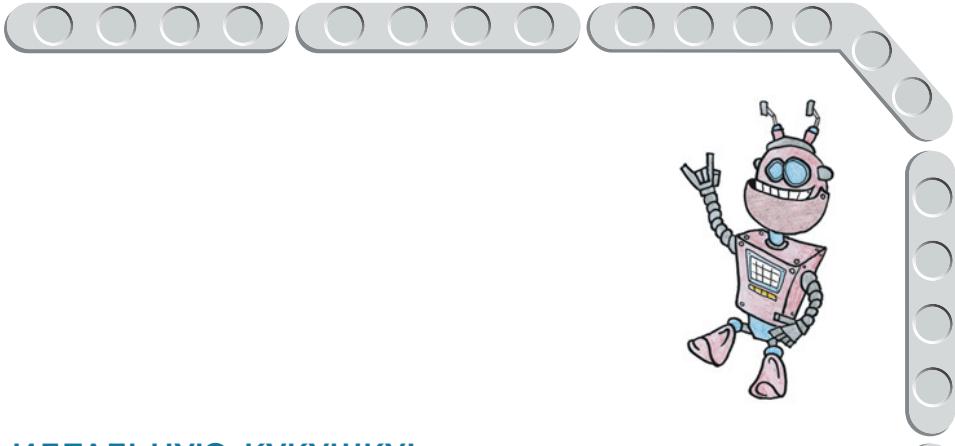
Внимание!

Наша программа – это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

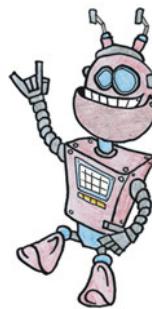
Если ты хочешь повторно запустить программу или продемонстрировать работу своего робота другу, то для его запуска компьютер тебе уже не нужен.

Включи программируемый блок, затем выбери папку *Clock* и в ней программу *Time*.

1. Перед запуском программы обязательно убедись, что обе стрелки, часовая и минутная, установлены в положение «12 часов ровно».
2. Стрелки часов должны начать двигаться один раз в минуту. Это можно будет заметить по движению минутной стрелки.
3. Один раз в час, при прохождении перед датчиком цвета механизма боя часов, из домика должна появляться кукушка и издавать звук столько раз, сколько часов показывают часы, а затем снова «прятаться» обратно в домик.



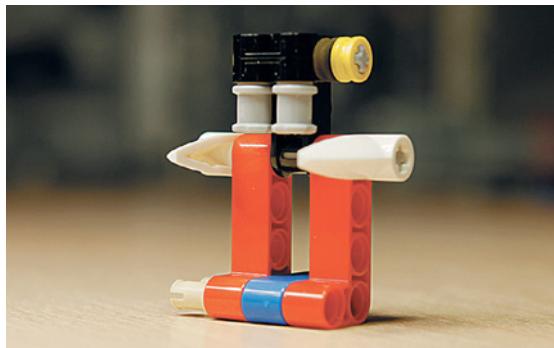
А теперь...



СОЗДАЙ СОБСТВЕННУЮ ИДЕАЛЬНУЮ КУКУШКУ!

Ты собрал часы с кукушкой, но самой кукушки, о которой мы так много говорили, — нет! Дело в том, что здесь полёт фантазии позволит тебе собрать свою уникальную и неповторимую кукушку!

Посмотри, что получилось у нас!



Мы решили собрать человечка из оставшихся деталей набора, который очень отдалённо напоминает нам знаменитого персонажа из космической саги «Звёздные войны» — злодея Дарта Вейдера!

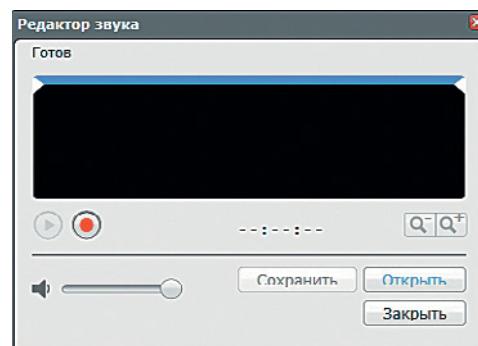
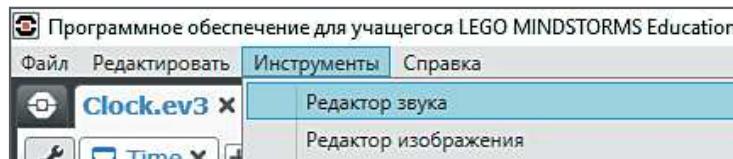
КУКУШКА, КУКУШКА, СКОЛЬКО МНЕ ОСТАЛОСЬ?

Помимо того, что сама кукушка может быть абсолютно любой, она ещё и может издавать именно тот звук, который ты захочешь!

Мы сделали так, чтобы наш Дарт Вейдер каждый час издавал свои знаменитые «вздохи». Для этого мы записали звук, похожий на его дыхание, и воспользовались редактором звука, встроенным в программное обеспечение EV3.

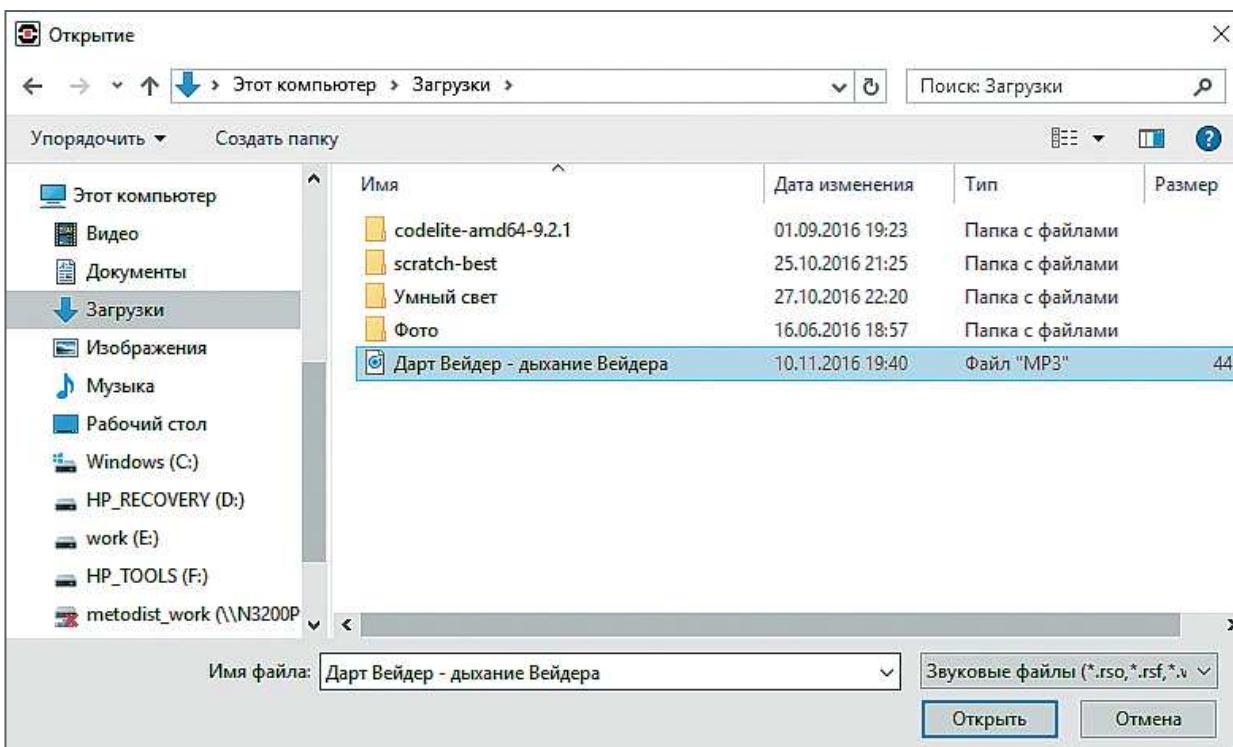
Снова открой свой проект **Clock** в LME-EV3 (**Файл → Открыть проект → Открыть**) и перейди к команде воспроизведения звука для кукушки. В ней ты чуть позже настроишь воспроизведение собственного звука.

- Скачай или запиши любой желаемый звук и загрузи его на компьютер.
- Открой сверху меню **Инструменты** → **Редактор звука**.

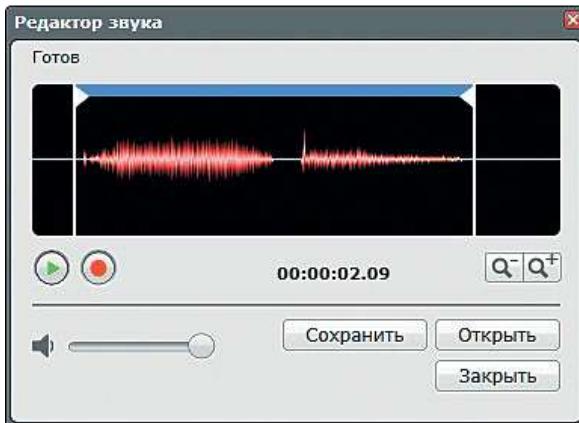


- Nажми в открывшемся окне кнопку **Открыть**.

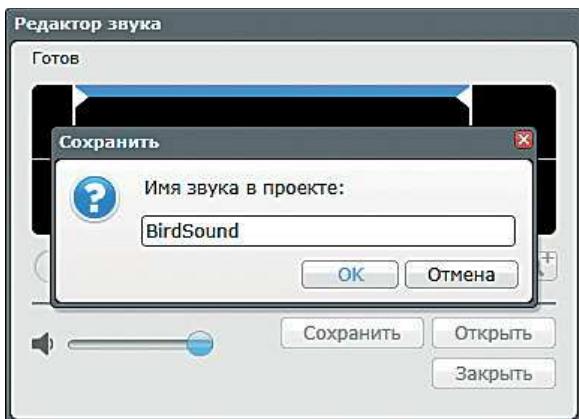
- Найди в памяти компьютера сохранённый звуковой файл и кликни по нему дважды.



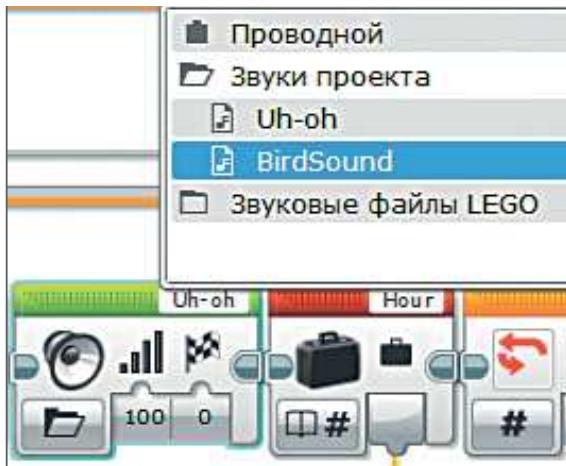
5. Звук откроется для предварительного редактирования. Ты можешь заметить, что у этого звука в начале и конце есть тонкие линии — это тишина. Чтобы он звучал правильно, тебе нужно **обрезать** тишину. Сделать это можно с помощью двух **бегунков сверху**: подвинь левый *бегунок вправо* до начала колебаний, а правый — влево.



6. Далее можешь прослушать этот звук, нажав кнопку с зелёным треугольничком, а затем нажми кнопку **Сохранить**. В открывшемся окне задай имя, по которому данный звук будет идентифицироваться в твоём проекте. В нашем примере мы назовём его **BirdSound** («Звук птицы», потому как у нас всё же «Часы с кукушкой»). Затем нажми кнопки **OK** и **Закрыть**.



7. Выбери теперь в программе в команде **Звук** (зелёный блок) в правом верхнем окошке в папке **Звуки проекта** файл **BirdSound**.

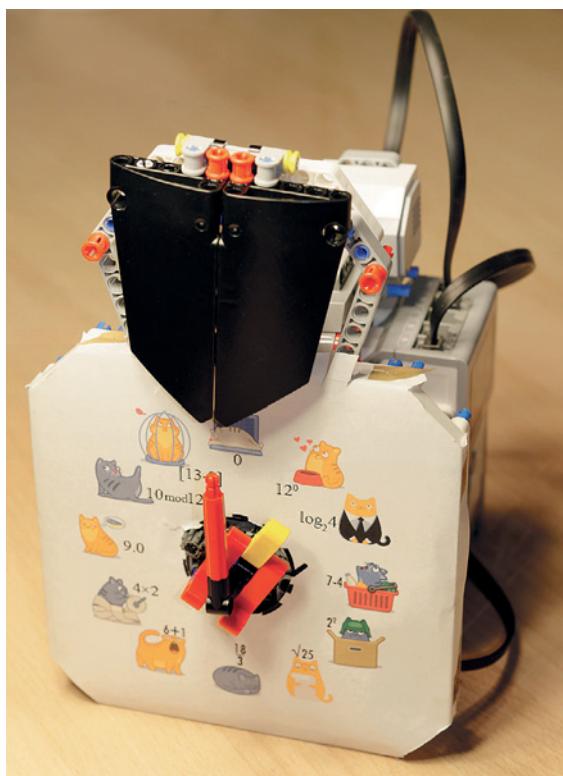


Готово! Наш Дарт Вейдер ожил!
Интересно, а что получилось у тебя?

ТАК КОТОРЫЙ ЧАС?

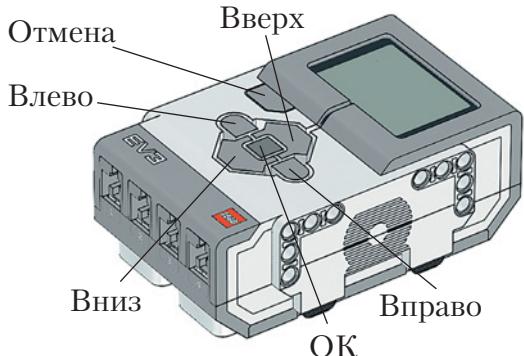
Совершенным произведением искусства может стать твой собственный циферблат часов! Ты можешь сделать его интересным, доселе невиданным, используя вместо стандартных цифровых обозначений, собственные знаки или шифры. Возможно, твои часы будут показывать время на Марсе?

Посмотри, какой циферблат получился у нас: каждый час мы закодировали математическим примером и сопроводили изображением котика, чтобы показать, чем он должен заниматься в определённое время.

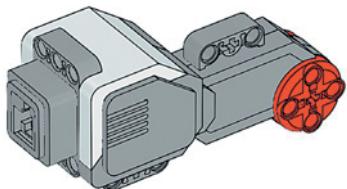


Перечень деталей

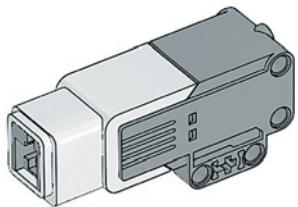
Модуль EV3, 1x



Большой мотор, 1x



Средний мотор, 1x

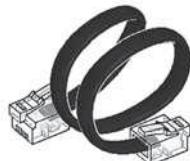


Датчик цвета, 1x



Кабель

Кабель, 25 см, 3x



USB-кабель, 1x



Зубчатые колёса

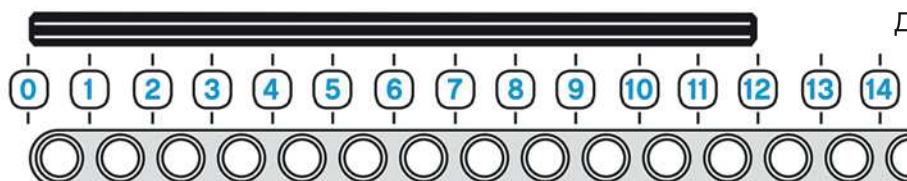
Зубчатое колесо, 8 зубьев,
тёмно-серое, 4x



Двойное коническое зубчатое
колесо, 12 зубьев,
чёрное, 1x



Зубчатое колесо, 24 зуба,
тёмно-серое, 2x

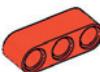


Балки прямые

Балка, 3-модульная,
чёрная, 3x



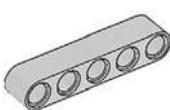
Балка, 3-модульная,
красная, 3x



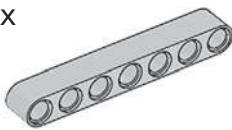
Балка, 3-модульная,
жёлтая, 2x



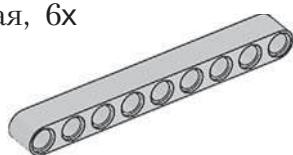
Балка, 5-модульная,
серая, 3x



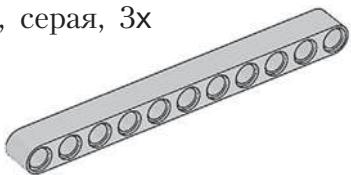
Балка № 7, серая, 4x



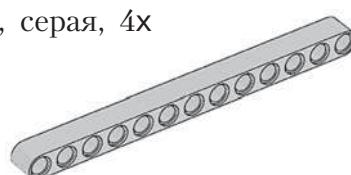
Балка № 9, серая, 6x



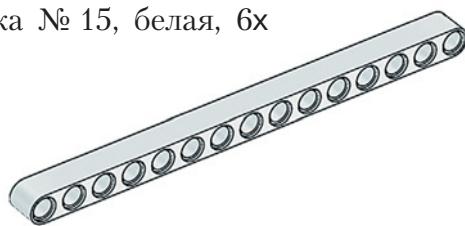
Балка № 11, серая, 3x



Балка № 13, серая, 4x

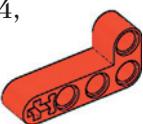


Балка № 15, белая, 6x

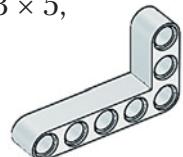


Балки угловые

Прямоугольная балка 2×4 ,
красная, 4x



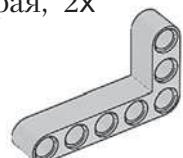
Прямоугольная балка 3×5 ,
белая, 3x



Угловая балка 4×4 ,
белая, 3x



Угловая балка 3×5 , серая, 2x



Угловая балка 4×6 ,
чёрная, 1x



Угловая балка 3×7 , серая, 2x



Двойная угловая балка
 3×7 , белая, 4x



Балки поперечные

Поперечная балка 2 × 1,
красная, 1x



Тавровая балка 3 × 3,
чёрная, 1x



Блоки

Двойной поперечный блок,
3-модульный, чёрный, 2x



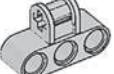
Поперечный блок,
2-модульный, серый, 3x



Поперечный блок, 3-мо-
дульный, серый, 4x



Поперечный блок, 4-мо-
дульный, серый, 1x



Прямой блок № 1, чёрный,
2x



Труба

Труба, 2-модульная,
серая, 1x



Штифты

Соединительный штифт,
2-модульный, чёрный, 62x



Соединительный штифт,
3-модульный, синий, 29x



Соединительный штифт
со втулкой, 3-модульный,
красный, 3x



Соединительный штифт,
2-модульный, серый, 7x



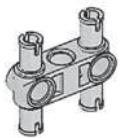
Соединительный штифт,
2-модульный, бежевый, 2x



Соединительный штифт,
3-модульный, бежевый, 1x



Н-образный соединительный
штифт, серый, 2x



Оси

Ось № 4 с фиксирующим элементом, серая, 2x



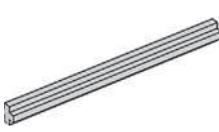
Ось № 10, чёрная, 1x



Ось № 8, чёрная, 1x



Ось № 7, серая, 1x



Ось № 6, чёрная, 1x



Ось № 5, серая, 3x



Ось № 4, чёрная, 2x



Ось № 3, серая, 1x



Ось № 2, красная, 5x



Втулки

Втулка, $\frac{1}{2}$ -модульная, жёлтая, 2x



Втулка, серая, 1x



Втулка, 3-модульная, чёрная, 1x



Угловая втулка, 3-модульная, чёрная, 1x



Втулка, 2-модульная, красная, 2x



Вилка поперечного блока 2×2 , чёрная, 3x



Вращающееся днище на 28 зубьев, серое, 2x



Вращающаяся верхушка на 28 зубьев, чёрная, 2x



Поперечный фиксатор, 2-модульный, чёрный (резиновый), 1x



Правая криволинейная панель
 5×11 , чёрная, 1x

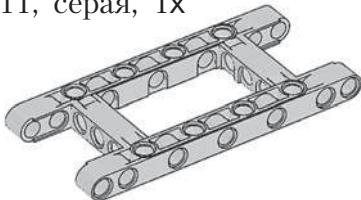


Левая криволинейная панель
 5×11 , чёрная, 1x

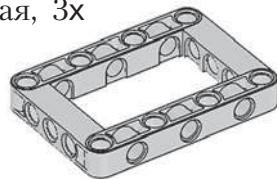


Рамки

Рамка 5×11 , серая, 1x

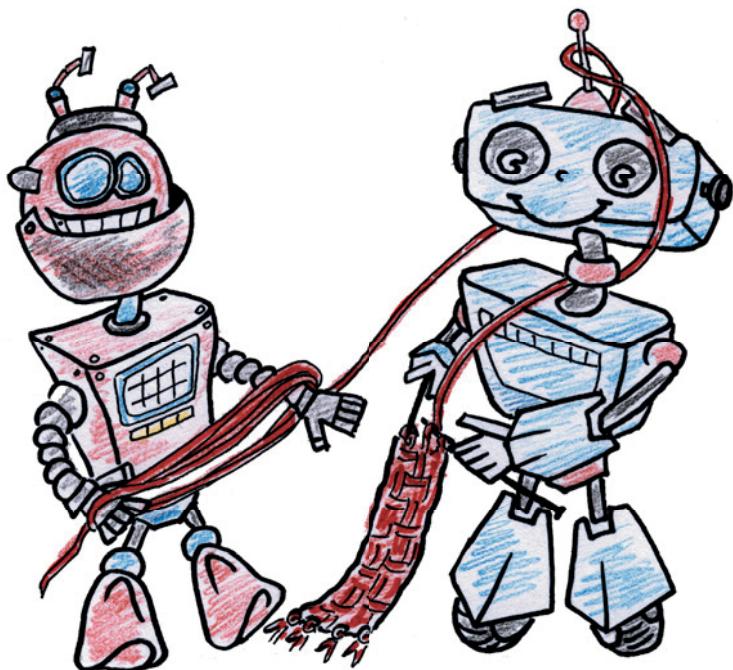


Рамка 5×7 , серая, 3x



education

Ткацкий станок



История ткачества



Ткани повсюду: из них шьют одежду, постельное бельё, тканями украшают своё жилище.

Ткачество — очень древнее ремесло. Сначала люди носили шкуры, но это было не совсем удобно. Шуба на севере зимой — подходящая одежда, но в жарких странах нужно что-то полегче.

Ткачеству предшествовало плетение: люди использовали траву, полоски кожи и жилы животных, чтобы защитить тело от палящего зноя, холода, дождя и ветра.

Шли годы, десятилетия, века, а ткачество развивалось и совершенствовалось. За 5–6 тысяч лет до н. э. появились первые ткацкие станки (рис. 1, взят с сайта www.liveinternet.ru). На одном из таких ткала Пенелопа, жена Одиссея.

Сначала станки были примитивными, но всё-таки очень облегчили монотонный труд ткача. По мере развития ткачество превратилось в промысел. Ткани стали изготавливать не только для себя, но и на продажу. У многих народов ткачество — это часть национальной культуры.

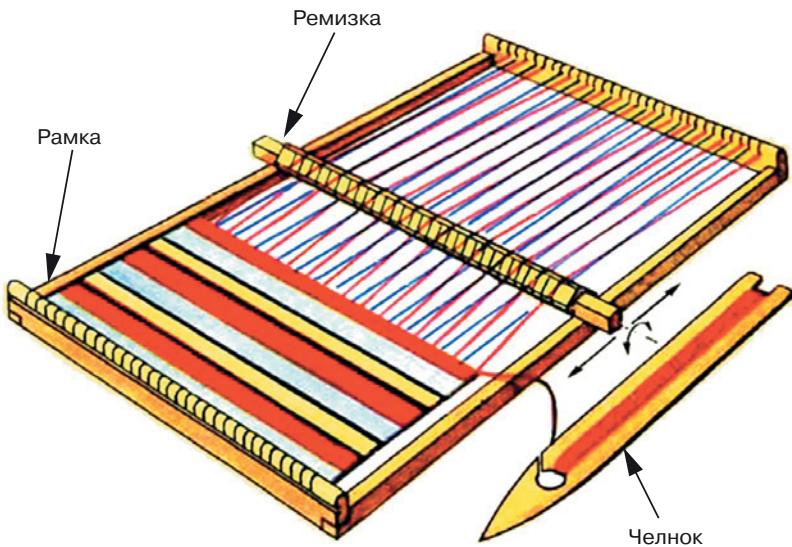


Рис. 1. Модель простейшего ткацкого станка



С древних времён на Руси существовало домашнее ткачество, которое играло важную роль в жизни крестьян. Каждая женщина в доме с малых лет умела ткать одежду, пояса, ленты, полотенца, скатерти, покрывала, занавески, половики и многое другое. Как сырьё использовали лён, коноплю, шерсть (козью или овечью). Для начала сырьё обрабатывали, потом отбеливали, красили и пряли. И только после этого приступали к трудоёмкому и требующему внимания процессу ткачества.

В XVIII веке в Англии появился первый механический ткацкий станок — это было одним из главных достижений человечества.

Принцип действия станка прост: нити основы делят гребёнкой на две части (можно через одну нить, можно через несколько) и пропускают через «глазки» на вертикальных планках (на рисунке эта часть обозначена как «Ремизки»). Ремизки поочерёдно поднимаются и опускаются, в результате нити расходятся. В образовавшийся зазор (зев) протаскивают нить утка с помощью челнока.

Затем с помощью бёрда (рис. 2, взят с сайта www.cross-kpk.ru) нить прибивают плотнее к уже продетым. После этого ремизки меняют местами (одну поднимают вверх, другую опускают вниз) — получается переплетение. Этот процесс повторяется много раз, пока не получится полотно нужной длины. Нити основы сматываются с катушки навоя, переплетаются с утком, и на катушку товарного валика наматывается уже готовая ткань.

Следующий шаг развития ткачества — автоматизация: в XVIII веке на станках устанавливают двигатель. Появляются ткацкие фабрики.

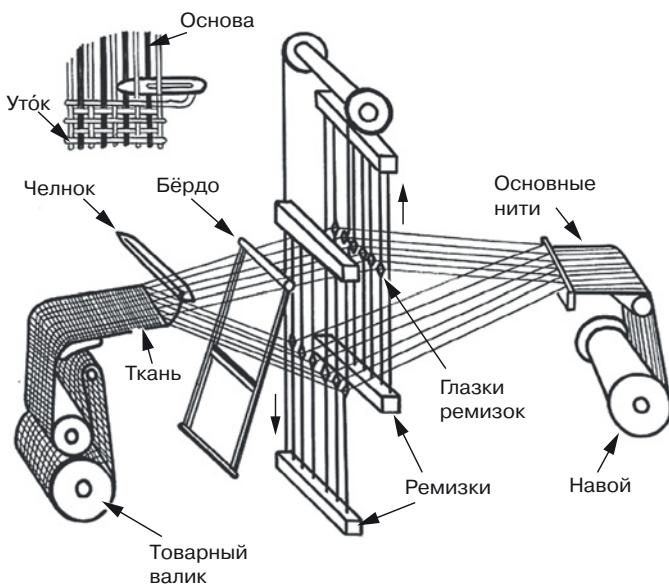


Рис. 2. Основные детали механического ткацкого станка

Внимание!

Для выполнения этого проекта потребуется ресурсный набор LEGO® MINDSTORMS® Education EV3, а также нитки для основы и утка типа «ирис» двух цветов.

Домотканые полотна начинают вытесняться фабричными тканями. Ткачество уходит в прошлое, а ему на смену приходит ткацкое производство. Но традиции домашнего ткачества сохранились до наших дней.

Способ изготовления ткани остался тем же, что и в древности. Ткань состоит из перпендикулярных нитей,

переплётённых между собой. Нити удерживают друг друга и образуют полотно, свойства которого будут зависеть от вида и качества нитей, способа их переплетения, от его плотности.

Нити, идущие вдоль ткани, называются **основа**, а нити, вплетённые поперёк основы, называются **уток**.

И ты, собрав станок, сможешь выткать на нём то, что тебе захочется.

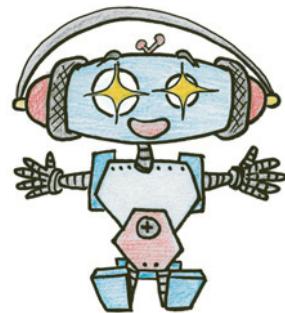
Как работает ручной ткацкий станок, можно посмотреть на видео, выбрав любую ссылку из предложенных¹.

Если ссылки окажутся неактивными, тебе будет нетрудно найти нужный материал в Интернете.

Современный ткацкий станок фактически является роботизированным комплексом, модель которого (вполне рабочую) мы предлагаем тебе построить и самому получить тканое полотно.

¹ http://www.youtube.com/watch?v=2TFEieeY_JM
<http://www.youtube.com/watch?v=ixDNo5fy7lk>
<http://www.youtube.com/watch?v=390vvnOUNG0>
<http://www.youtube.com/watch?v=jELlbFQwuCg>

Этап 1. Устройство автоматического ткацкого станка



Рассмотри модель ткацкого станка, собранную на основе набора LEGO® MINDSTORMS® Education EV3 (рис. 3).

Попробуй выделить на ней рабочие детали станка — ремизку, бёрдо, челнок.

Попробуй собрать эту модель.

Обрати внимание: в таблице в конце проекта даны все детали, которые потребуются тебе для сборки. Эта таблица поможет быстро найти то, что необходимо, и не ошибиться при конструировании.

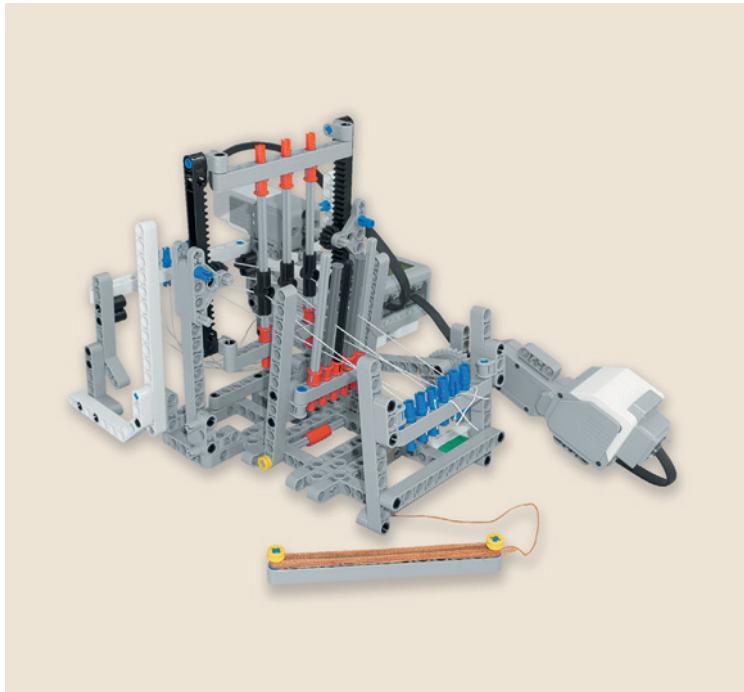
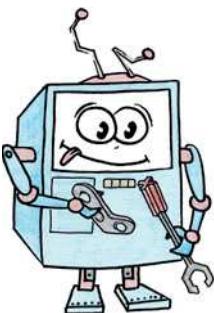


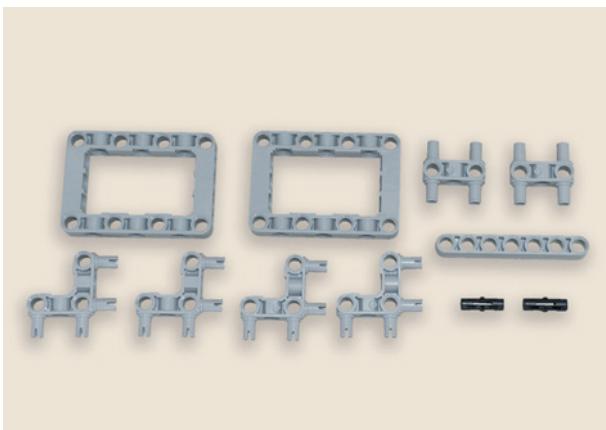
Рис. 3. Модель автоматического ткацкого станка



Этап 2. Сборка автоматического ткацкого станка



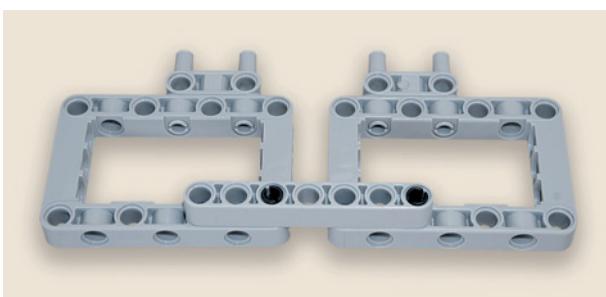
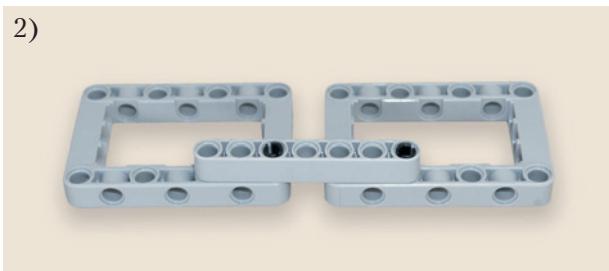
ШАГ 1. СБОРКА ОСНОВАНИЯ



Детали для сборки:

- рамка 5×7 , серая 2x;
- соединительный штифт, 2-модульный, чёрный, 2x;
- угловой соединительный штифт, 3 × 3-модульный, серый, 4x;
- Н-образный соединительный штифт, серый, 2x;
- балка № 7, серая, 1x.

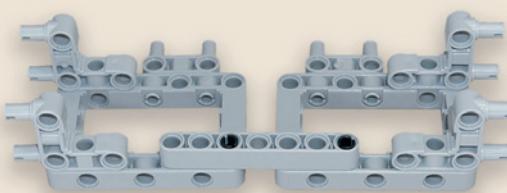
1. Вставь в рамки чёрные штифты, как показано на рисунке, и на них закреши балку третьим и седьмым модулями.



2. Вставь в боковые отверстия рамок два Н-образных штифта.

3. Установи по углам основания четыре угловых штифта.

Основание готово.



ШАГ 2. СБОРКА ПОДЪЁМНОГО МЕХАНИЗМА РЕМИЗКИ

Детали для сборки:

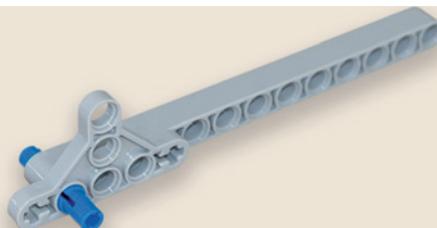
- ось № 3, серая, 3х;
- ось № 5, серая, 1х;
- балка № 13, серая, 2х;
- соединительный штифт, 3-модульный, синий, 2х;
- треугольный фиксатор 3×5 , серый, 4х;
- балка с зубцами № 13, чёрная, 2х;
- зубчатое колесо на 12 зубцов, чёрное, 2х.



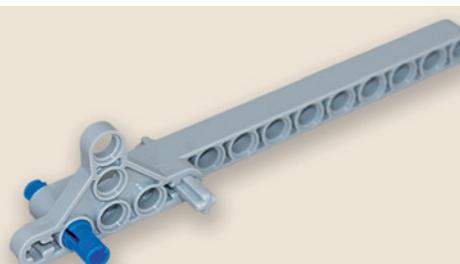
Ремизка состоит из основной части и подъёмного механизма.

Подъёмный механизм состоит из двух почти одинаковых частей. Сначала собери первую половину.

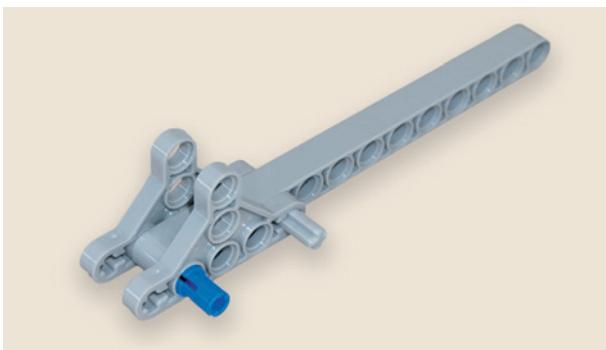
1. Приложи треугольный фиксатор спереди к балке и закрепи синим 3-модульным штифтом со стороны фиксатора (от себя).



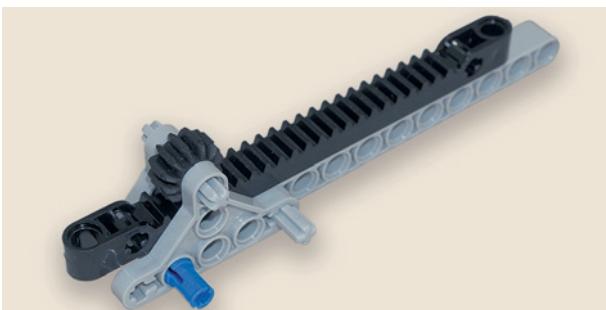
2. Продень ось № 3 в крестообразное отверстие фиксатора.



- Установи на ось и штифт второй треугольный фиксатор (сзади).

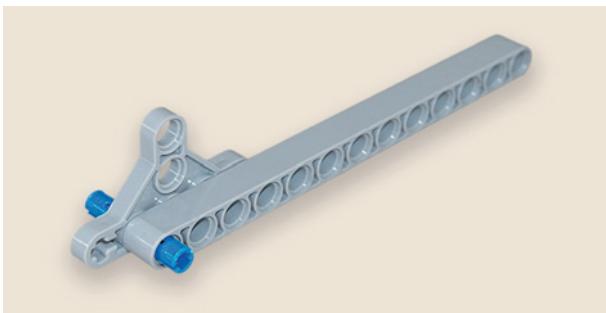


- Вложи между фиксаторами балку с зубцами и закреши её сверху зубчатым колесом и осью № 3.

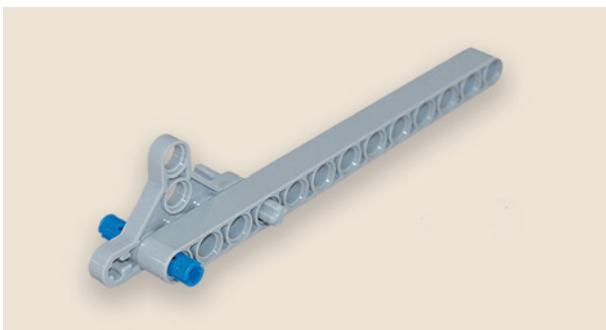


Теперь собери вторую часть подъёмного механизма. Для сборки потребуются те же детали, что и для первой половины, только добавится ещё ось № 5.

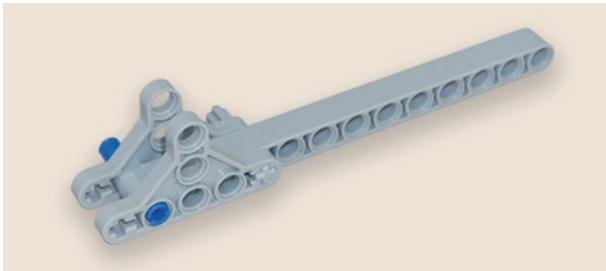
- Приложи треугольный фиксатор к балке сзади и закреши синим 3-модульным штифтом (к себе).



- Продень ось № 3 в крестообразное отверстие фиксатора.



7. Установи на ось и штифт второй треугольный фиксатор.



8. Вложи между фиксаторами балку с зубцами и закрепи её сверху зубчатым колесом и осью № 5.

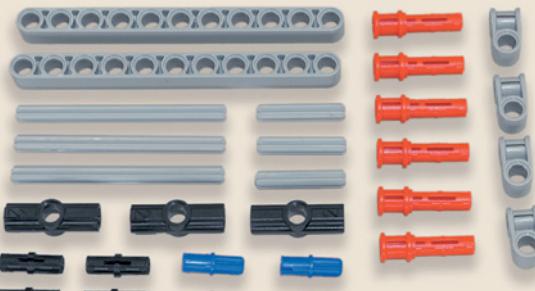


Подъёмный механизм готов.

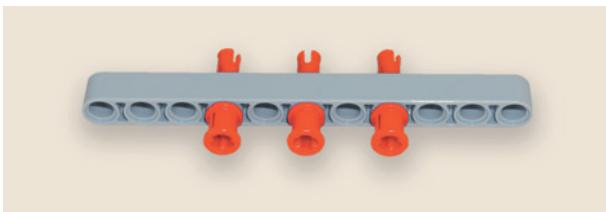
ШАГ. 3. СБОРКА ОСНОВНОЙ ЧАСТИ РЕМИЗКИ

Детали для сборки:

- ось № 3, серая, 3x;
- ось № 7, серая, 3x;
- балка № 11, серая, 2x;
- поперечный блок, 2-модульный, серый, 4x;
- соединительный штифт, 2-модульный, синий, 4x;
- соединительный штифт, 2-модульный, чёрный, 4x;
- соединительный штифт с втулкой, красный, 6x;
- угловой блок 2, 180°, чёрный, 3x.



1. Вставь в балку № 11 красные штифты в четвёртый, шестой и восьмой модули.



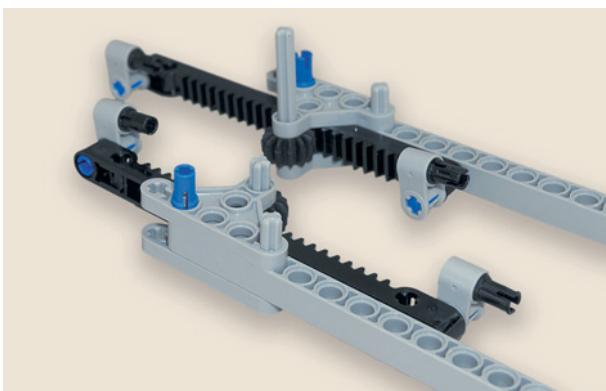
2. Закрепи по концам балок с зубцами синие штифты.



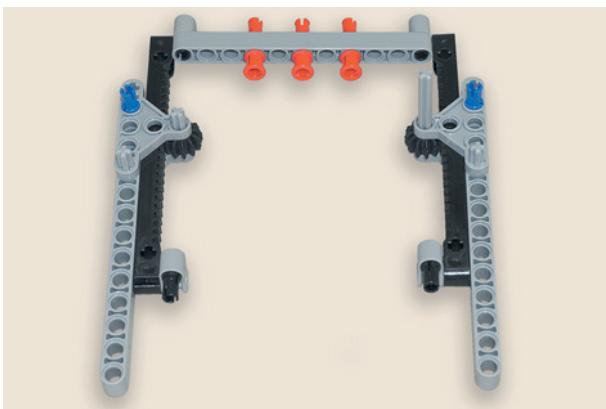
3. Вставь в поперечные блоки чёрные штифты.



4. Установи на синих штифтах балок с зубцами поперечные блоки.



5. Соедини половины подъёмного механизма балкой № 11 с красными штифтами.



На очереди — направляющие резинки.

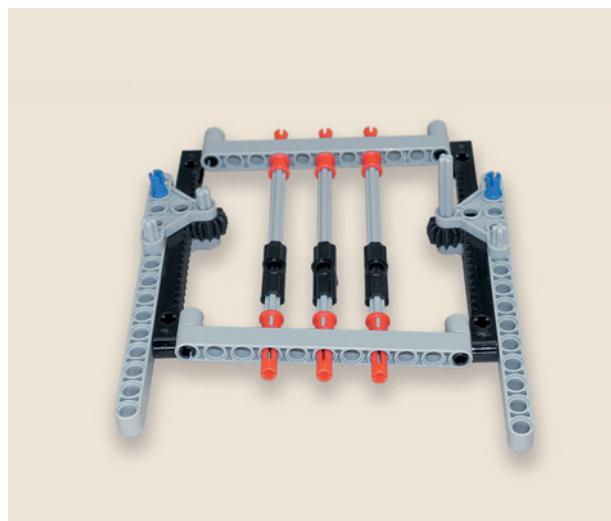
6. Сделай глазок ремизки из соединительной втулки с модулем, которую вставь между осью № 7 и осью № 3 (длинные оси направляющих — верх, короткие — низ).



7. Вставь в красные штифты направляющие ремизки.

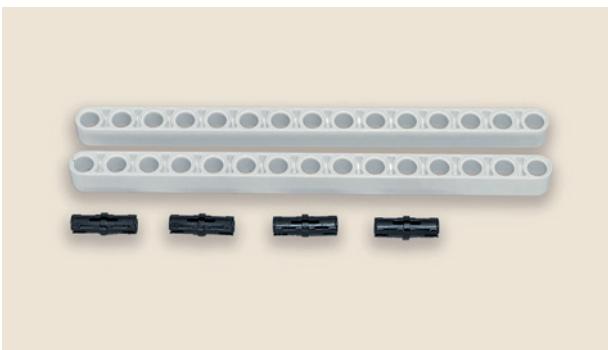


8. Закрепи направляющие ремизки в четвёртом, шестом и восьмом модулях второй балки № 11.



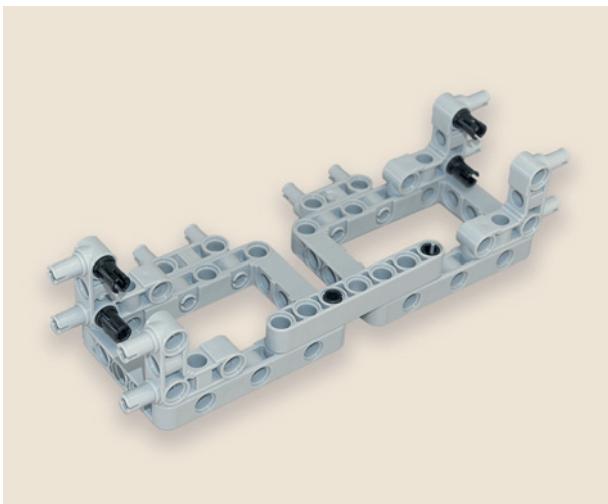
Ремизка готова.

ШАГ 4. СОЕДИНЕНИЕ ОСНОВАНИЯ С РЕМИЗКОЙ

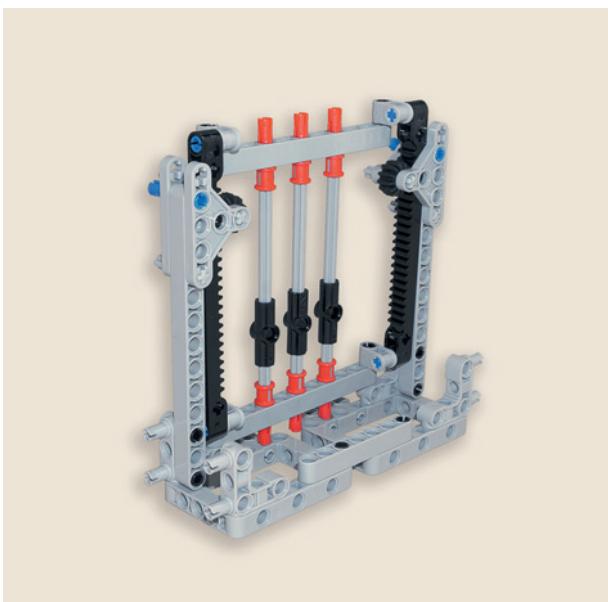


Детали для сборки:

- собранное основание;
- ремизка;
- балка № 15, белая, 2x;
- соединительный штифт, 2-модульный, чёрный, 4x.

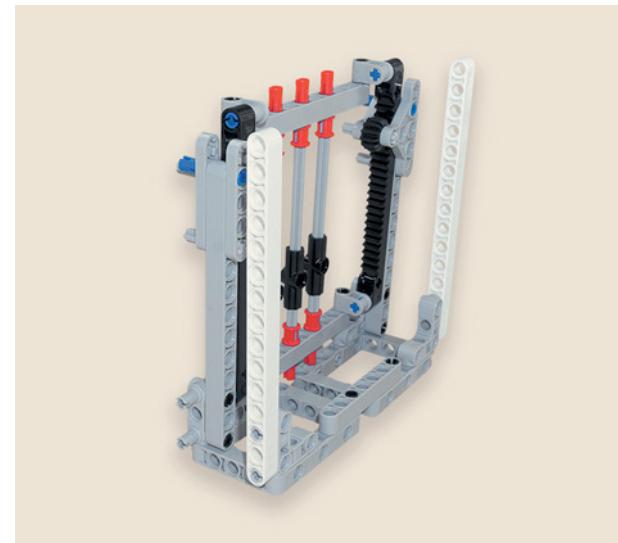


1. Вставь в угловые соединительные 3×3 -модульные штифты основания чёрные 2-модульные штифты.



2. Установи на них балки № 13 ремизки.

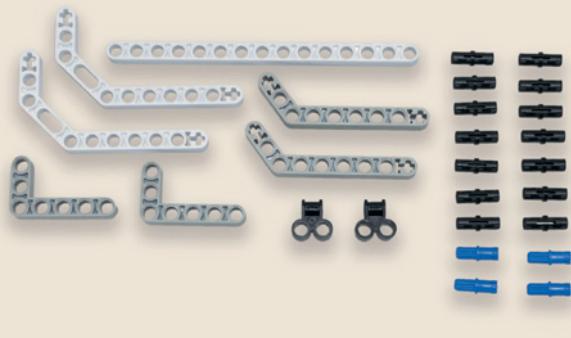
- Закрепи с другой стороны основания на угловых штифтах белые балки № 15.



ШАГ 5. СБОРКА РАМ ДЛЯ ЗАКРЕПЛЕНИЯ НИТЕЙ ОСНОВЫ

Детали для сборки:

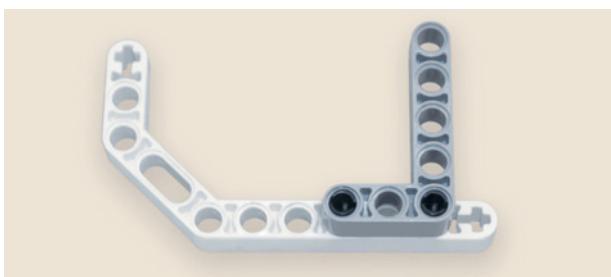
- балка № 15, белая, 1x;
- двойная угловая балка 3 × 7, белая, 2x;
- угловая балка 3 × 7, серая, 2x;
- прямоугольная балка 3 × 5, серая, 2x;
- поперечный блок, 3-модульный, чёрный, 2x;
- соединительный штифт, 2-модульный, чёрный, 14x;
- соединительный штифт, 2-модульный, синий, 4x.



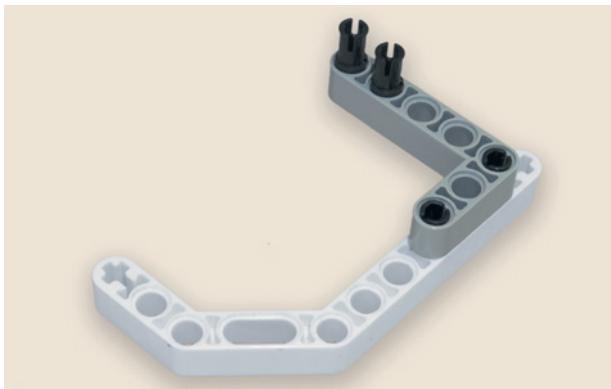
- Вставь чёрные штифты в белую двойную угловую балку.



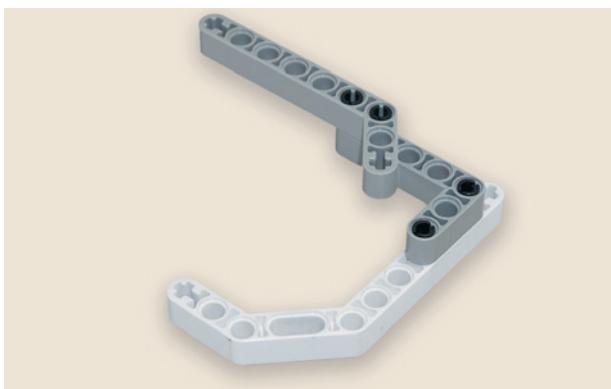
2. Соедини штифтами белую двойную балку с серой прямоугольной балкой 3×5 .



3. Вставь в серую прямоугольную балку два чёрных штифта.



4. А теперь надень на них серую угловую балку 3×7 .



5. Собери вторую такую же конструкцию, но в зеркальном отображении.

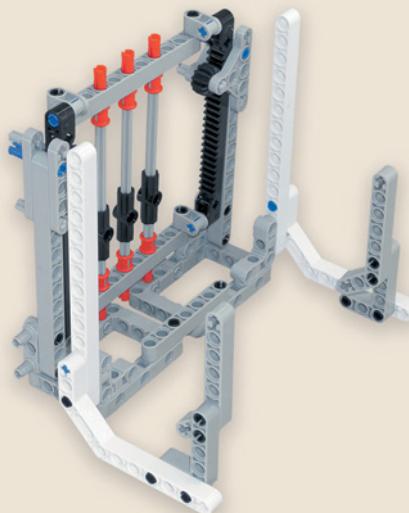


Теперь нужно подготовить элементы рамы к соединению с основанием станка.

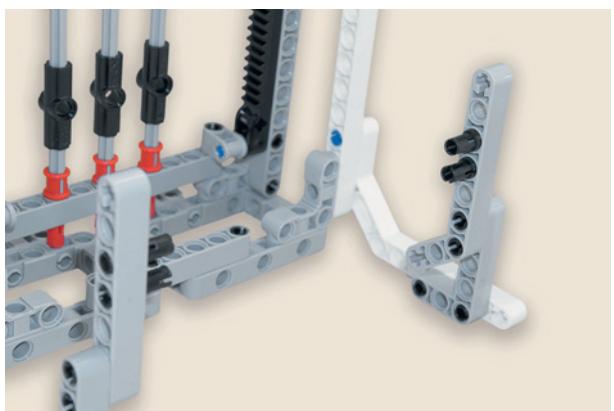
6. Вставь синие штифты в крестообразные модули в верхних концах белых двойных угловых балок 3×7 , вставь через один модуль чёрные штифты.



7. Закрепи двойные угловые балки получившихся элементов рамы на белых балках основания с тыльной стороны.



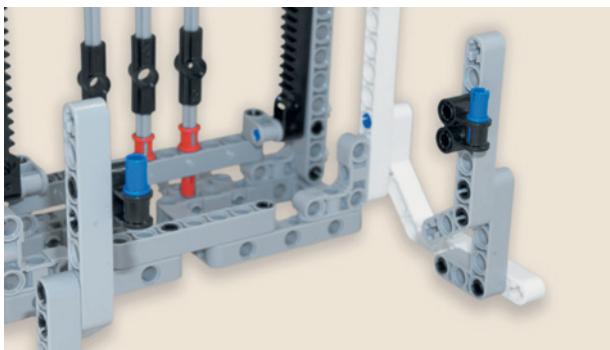
8. Закрепи в третьем и четвёртом модулях серых угловых балок чёрные штифты.



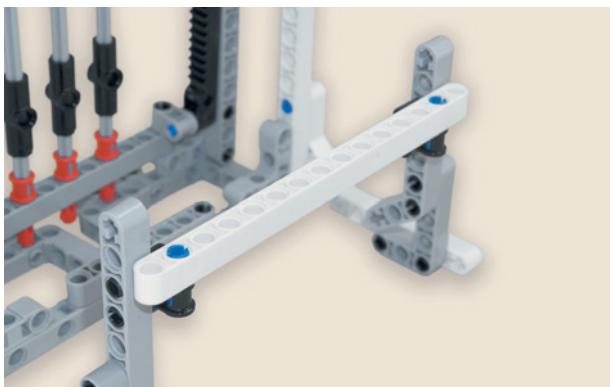
9. Вставь синие штифты в крестообразные модули поперечных блоков.



10. Закрепи поперечные блоки на серых угловых балках.



11. Поставь на синие штифты белую балку № 15 во втором и в четырнадцатом модулях.



Рама для закрепления нитей с тыльной стороны станка готова.

ШАГ 6. УСТАНОВКА СРЕДНЕГО МОТОРА



Детали для сборки:

- ремизка;
- средний мотор, 1x;
- соединительный штифт, 3-модульный, синий, 2x.

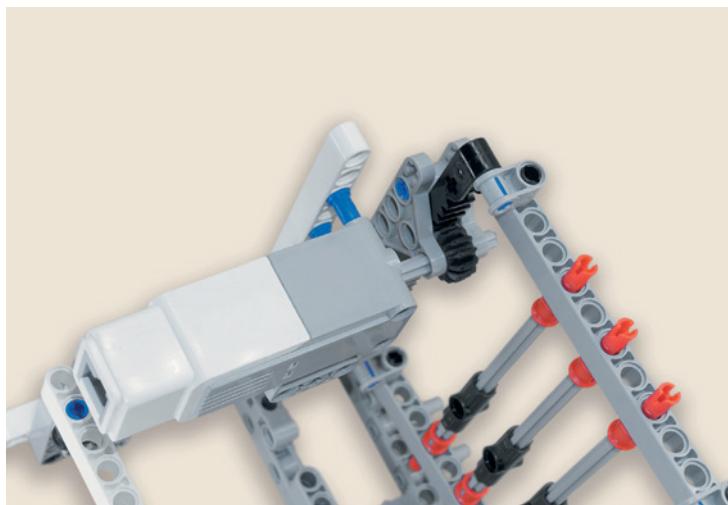


1. Протолкни ось № 5 в ремизке, чтобы её конец выступал на противоположной стороне. Ось должна выступать со стороны белой балки.

Установи на белой балке № 15 основания со стороны, где выступает серая ось ремизки, в третьем и пятом модулях сверху 3-модульные соединительные синие штифты.

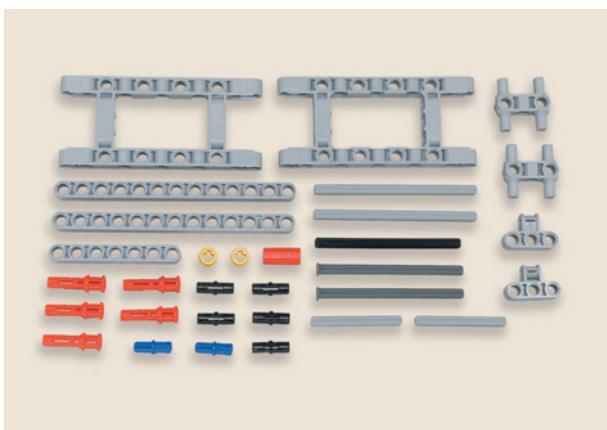


2. Установи на эти штифты и серую ось средний мотор.



Убедись, что ось надёжно закреплена в моторе! Средний мотор управляет движением ремизки вверх и вниз.

ШАГ 7. СБОРКА БЁРДА

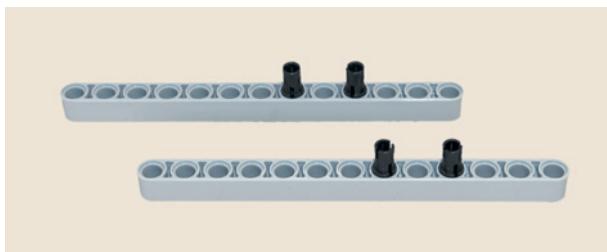
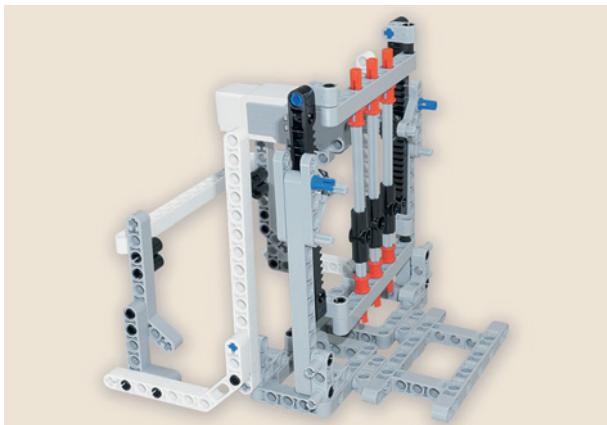


Детали для сборки:

- рамка 7×11 , серая, 2x;
- балка № 13, серая, 2x;
- балка № 7, серая, 1x;
- ось № 9, серая, 2x;



- ось № 8, с фиксирующим элементом, серая, 2x;
- ось № 8, чёрная, 1x;
- ось № 5, серая, 2x;
- поперечный блок, 4-модульный, серый, 2x;
- соединительный штифт с втулкой, 3-модульный, красный, 5x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт, 2-модульный, чёрный, 5x;
- Н-образный соединительный штифт, серый, 2x;
- втулка, $\frac{1}{2}$ -модульная, жёлтая, 2x;
- втулка-удлинитель оси, 2-модульная, красная, 1x.

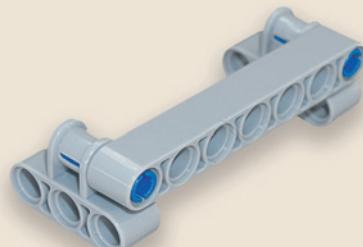


- Закрепи с лицевой стороны станка рамку 7×11 на Н-образных соединительных штифтах основания.
- Закрепи на серых балках № 13 в восьмом и десятом модулях чёрные штифты.

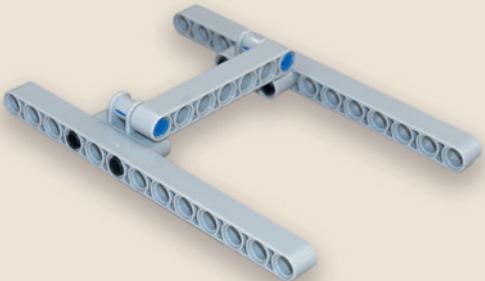
3. Вставь в серые 4-модульные поперечные блоки синие штифты с выступами.



4. Поставь на синие штифты крайними модулями серую балку № 7.

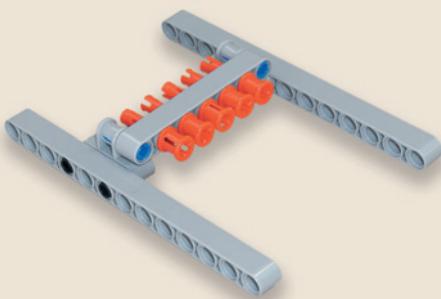


5. Соедини 4-модульные поперечные блоки с двумя балками № 13.

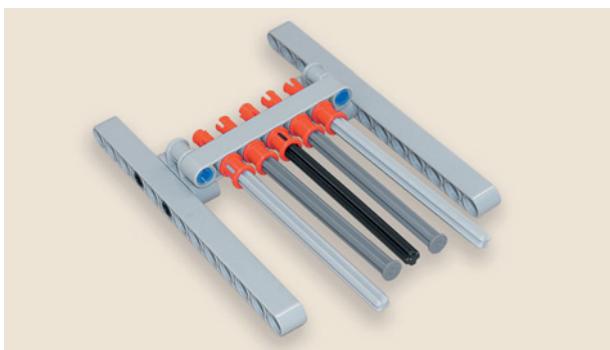


Теперь соберём прибивной механизм бёдра.

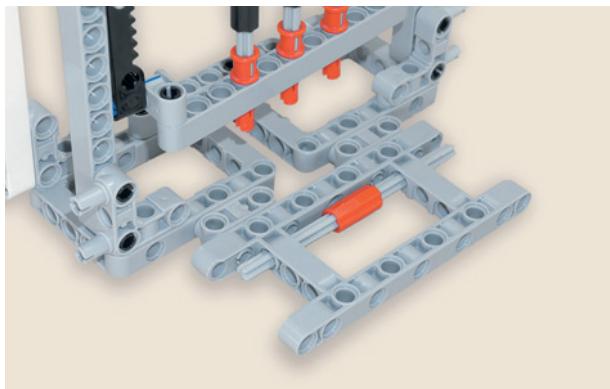
6. Вставь в балку № 7 красные штифты с втулками.



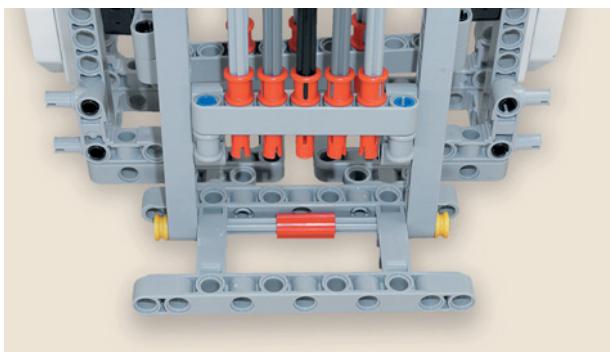
7. Вставь в крайние красные штифты серые оси № 9, в следующие штифты — оси № 8 с фиксирующим элементом, в средний штифт — чёрную ось № 8.



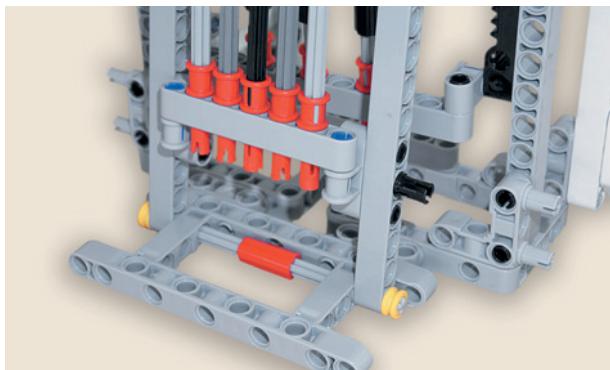
8. Продень в рамку 7×11 две оси № 5 и соедини их посередине красной втулкой.



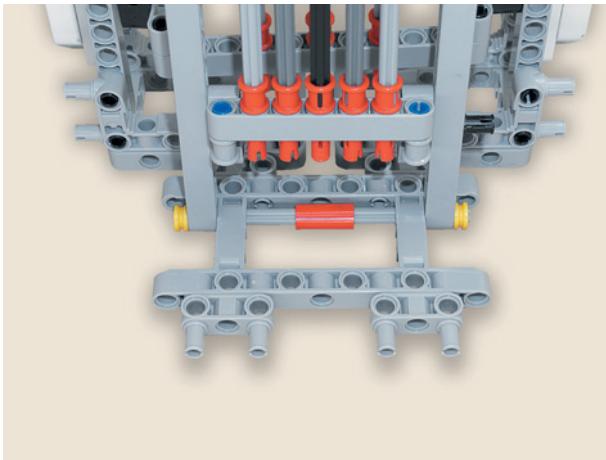
9. Надень балки № 13 пришивного механизма на оси № 5 (считая снизу) и закреши жёлтыми втулками.



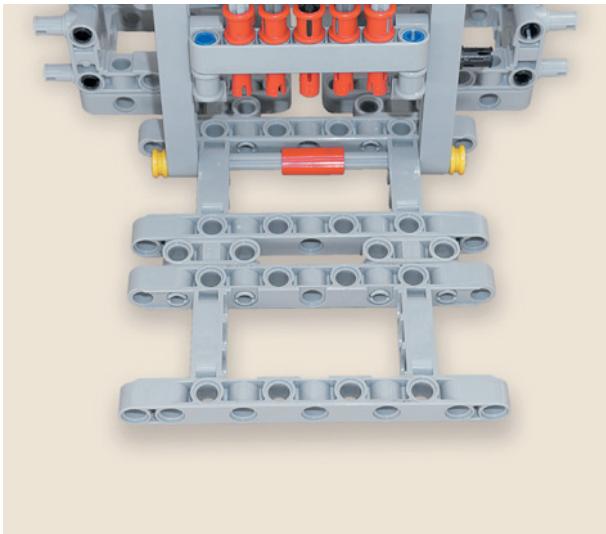
10. Установи с левой стороны пришивного механизма в балке № 13 в пятом модуле (считая снизу) чёрный штифт.



11. Установи с другой стороны рамки два Н-образных штифта.



12. Закрепи на этих штифтах рамку 7 × 11.

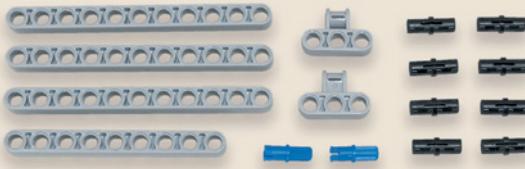


Бёрдо готово.

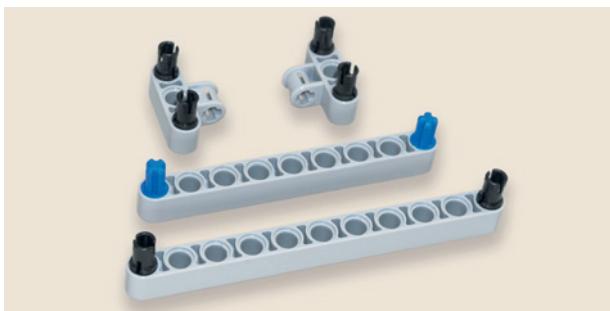
ШАГ 8. СБОРКА УСТРОЙСТВА КРЕПЛЕНИЯ НИТЕЙ ОСНОВЫ

Детали для сборки:

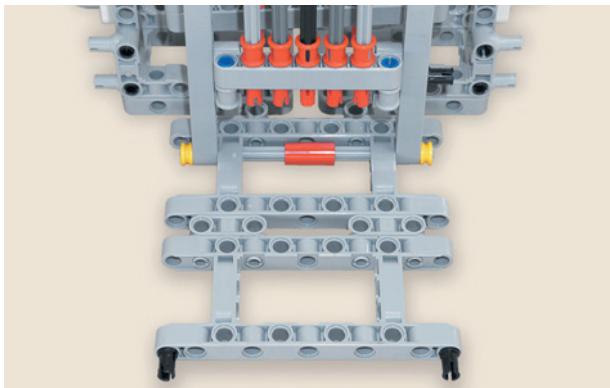
- балка № 11, серая, 3x;
- балка № 9, серая, 1x;
- поперечный блок, 4-модульный, серый, 2x;
- соединительный штифт, 2-модульный, чёрный, 8x;
- соединительный штифт, 2-модульный, синий, 2x.



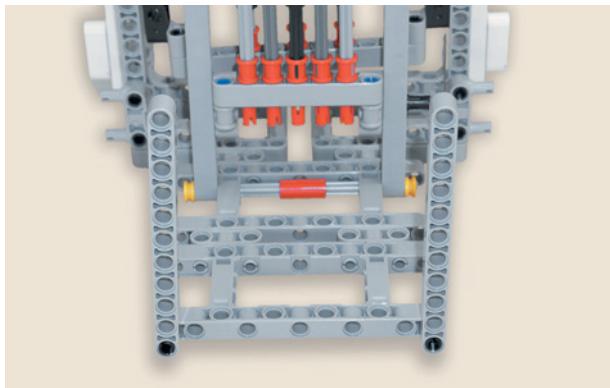
1. Установи чёрные штифты в балке № 11 и в 4-модульных поперечных блоках, синие – в балке № 9, как показано на рисунке.



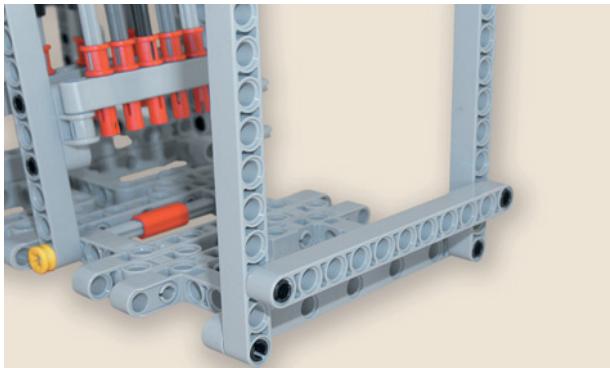
2. Вставь на свободной стороне рамки 7×11 в крайние модули чёрные штифты.



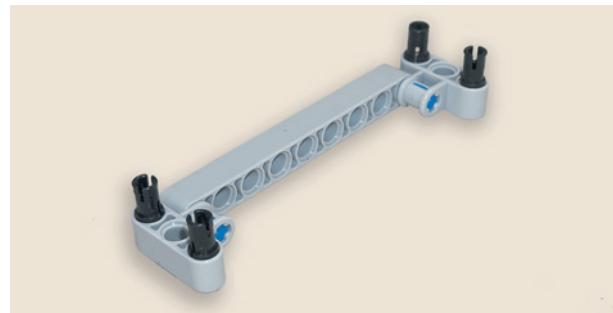
3. Закрепи на них серые балки № 11.



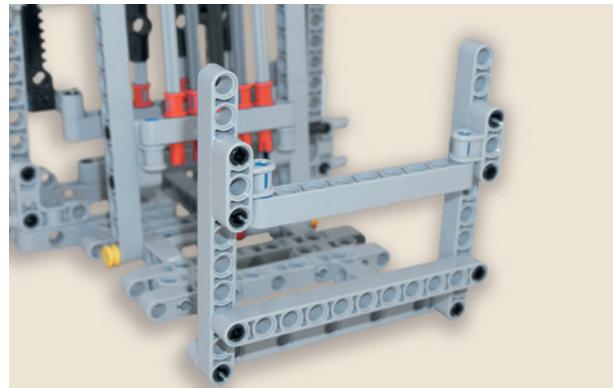
4. Вставь для обеспечения жёсткости конструкции в третий модуль снизу балок № 11 чёрные штифты и закрепи на них такую же балку.



5. Надень 4-модульные поперечные блоки на синие штифты балки № 9.



6. Закрепи собранный элемент в третьем и пятом (считая сверху) модулях балок № 11.



Устройство для крепления нитей основы с тыльной стороны станка готово. Чтобы прибивной механизм бё尔да заработал, необходимо собрать передаточный механизм.

ШАГ 9. СБОРКА ПЕРЕДАТОЧНОГО МЕХАНИЗМА БЁРДА

Детали для сборки:

- большой мотор, 1x;
- балка № 7, серая, 2x;
- двойной поперечный блок, 3-модульный, чёрный, 1x;
- ось № 5, серая, 1x;
- прямоугольная балка 3 × 5, белая, 1x;
- балка № 3, зелёная, 1x;
- зубчатое колесо на 40 зубьев, серое, 1x;
- соединительный штифт, 2-модульный, чёрный, 6x;
- соединительный штифт, 2-модульный, синий, 2x.



1. Вставь два чёрных штифта в шестой и седьмой модули балки и скрепи её с белой прямоугольной балкой.



2. Вставь синие штифты в крайние модули 3-модульного чёрного двойного поперечного блока.



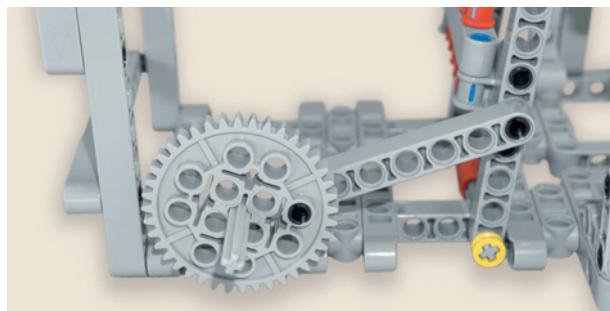
3. Установи чёрный поперечный блок на балке № 7 так, чтобы он оказался над белой балкой. Затем вставь в белую угловую балку два чёрных штифта и закрепи на них зелёную балку № 3; в её левый модуль вставь чёрный штифт.



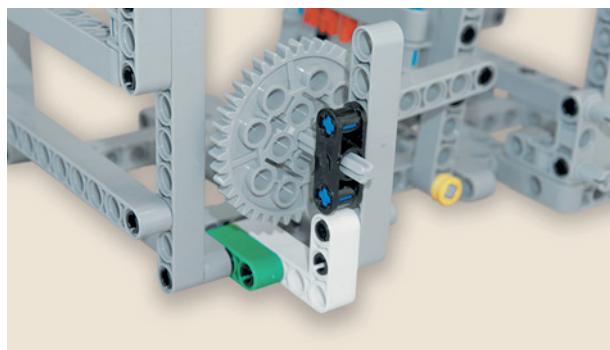
4. Прикрепи зубчатое колесо на 40 зубьев к балке № 7 чёрным штифтом.



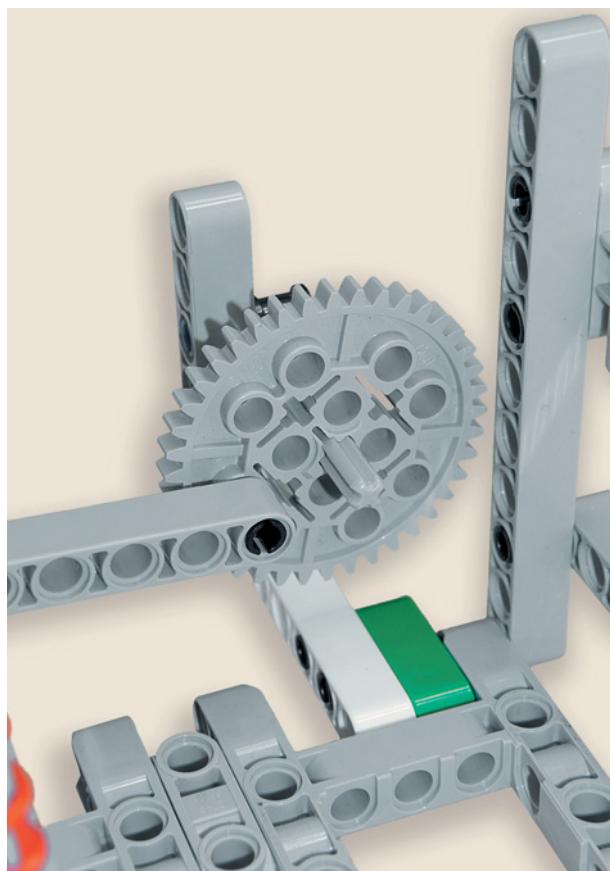
5. Надень зубчатое колесо на ось средним модулем. Прикрепи балку № 7 к пришивному механизму бёрда чёрным штифтом балки № 13.



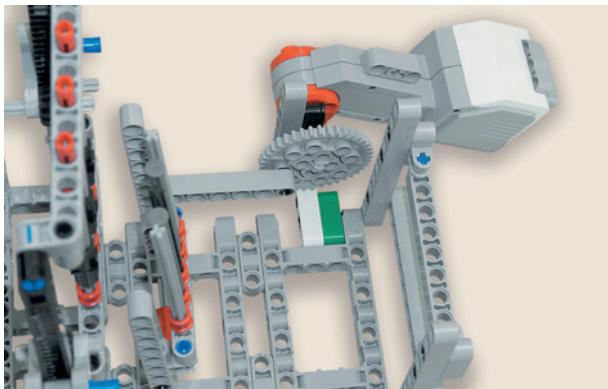
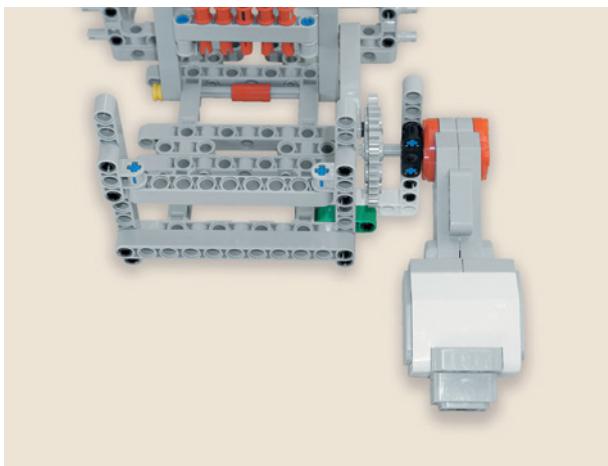
6. Пропусти ось через средний модуль 3-модульного поперечного блока элемента передаточного механизма.



7. Прикрепи передаточный механизм к раме основания чёрным штифтом зелёной балки.



8. Закрепи на оси большой мотор.



Внимание!

Посмотри, как должна быть расположена балка № 7, связывающая бёрдо и передаточный механизм. Необходимо точно воспроизвести это положение!

ШАГ 10. КРЕПЛЕНИЕ НИТЕЙ ОСНОВЫ

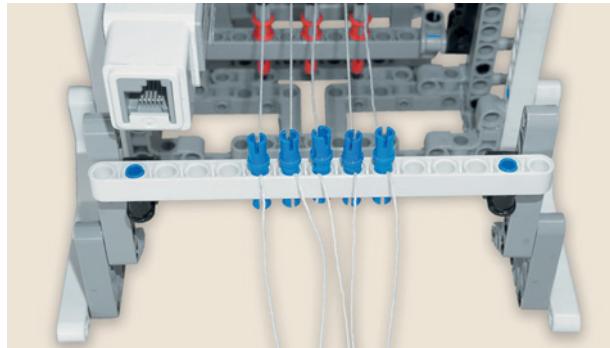


Детали для сборки:

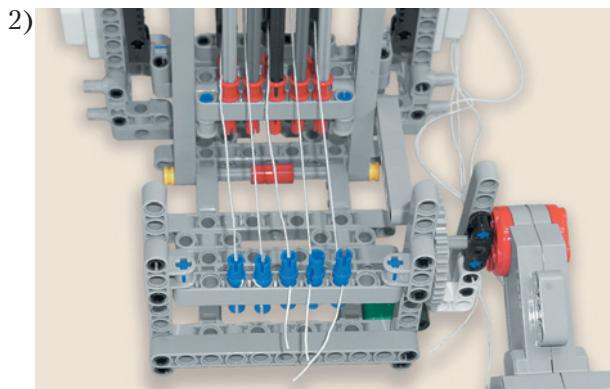
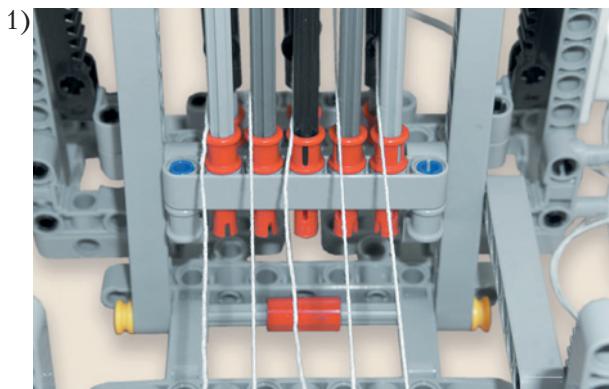
- нити основы одного цвета, 50 см, 5x;
- соединительный штифт, 3-модульный, синий, 10x.

Нити основы закрепляются в раме с лицевой и тыльной сторон синими 3-модульными соединительными штифтами.

- Закрепи синими штифтами нити основы с лицевой стороны в шестом, седьмом, восьмом, девятом и десятом модулях белой балки № 15. Первая, третья и пятая нити должны проходить через глазки ремизки, вторая и четвёртая — между ремизками.



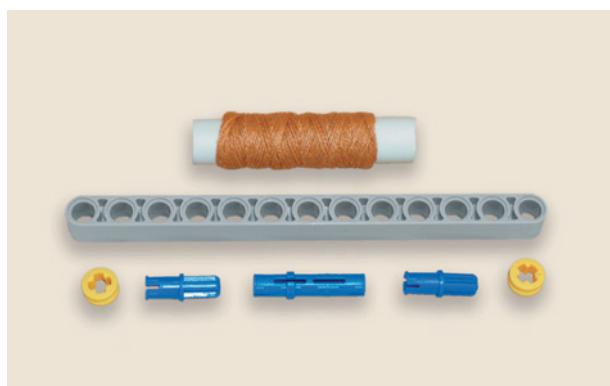
- Проведи нити основы через пришивной механизм бёरда, как на рисунке, и закрепи на тыльной стороне рамы синими штифтами в третьем, четвёртом, пятом, шестом и седьмом модулях балки № 9.



ШАГ 11. СБОРКА ЧЕЛНОКА

Детали для сборки:

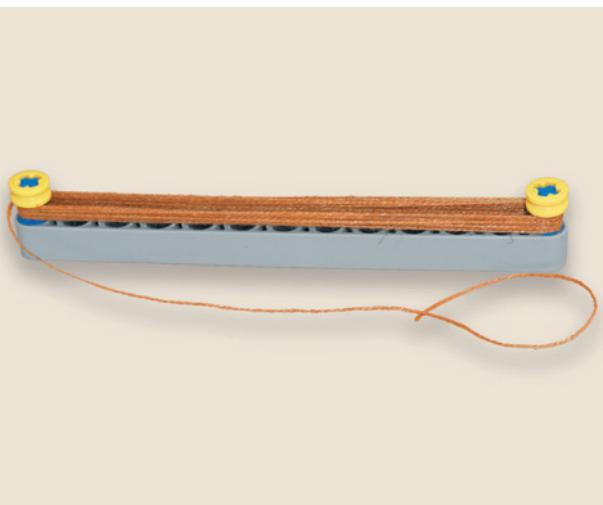
- балка № 13, серая, 1x;
- соединительный штифт, 2-модульный, синий, 2x;
- соединительный штифт, 3-модульный, синий, 1x;
- втулка, $\frac{1}{2}$ -модульная, жёлтая, 2x;
- нить утка, 1x.



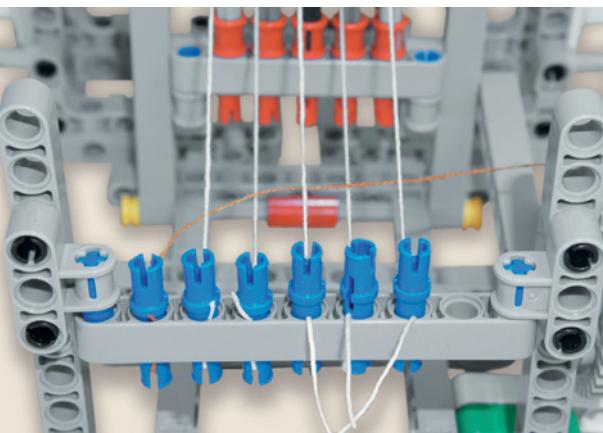
- Вставь синие штифты в крайние модули балки № 13 и надень сверху не до конца жёлтые втулки.



2. После того как челнок собран, намотай нить утка на синие штифты под жёлтые втулки.



3. Вставь в третий модуль серой балки синий 3-модульный штифт и закрепи на нём нить утка рядом с нитями основы, пропустив на тыльной стороне рамы через них челнок: одна нить основы сверху, другая снизу и т. д.



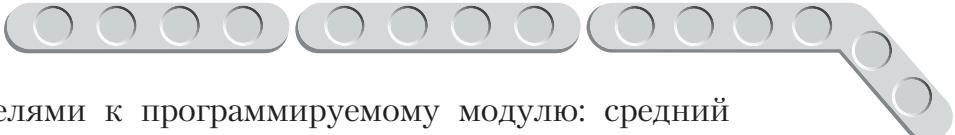
ШАГ 12. ПОДКЛЮЧЕНИЕ МОТОРОВ



Детали для сборки:

- программируемый модуль EV3, 1x;
- кабель, 25 см, 2x.





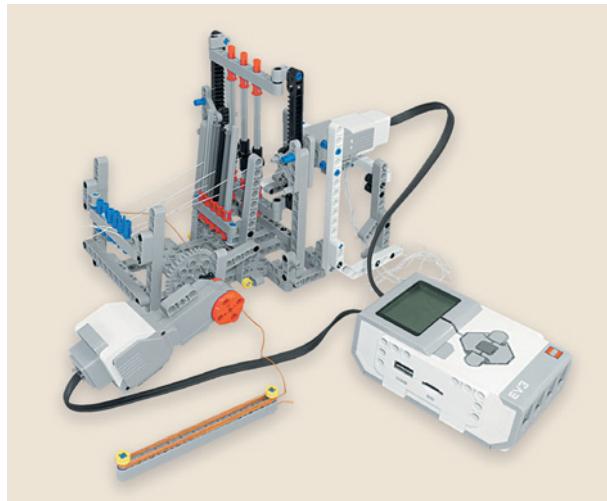
Подключи моторы кабелями к программируемому модулю: средний мотор — к порту A, большой мотор — к порту B.

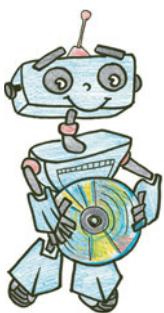


Внимание!

В начале работы ремизка должна находиться в нижнем положении, а пришивной механизм — стоять строго вертикально.

Нити основы должны иметь максимальное натяжение! Оно регулируется штифтами, удерживающими нити.





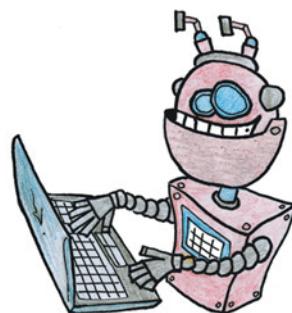
Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MIND-STORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

Этап 4. Создание программы для станка



ЛОГИКА ПРОГРАММЫ

Средний мотор, подключённый к порту А, будет поднимать ремизку. Станок будет ожидать нажатие кнопки **Вправо** на программируемом модуле, чтобы продолжить работу после прохода челнока. После нажатия кнопки прибивной механизм, приводимый в движение большим мотором (порт **В**), будет прибивать нить утка и возвращаться в начальное положение. После этого мотор А опускает ремизку и снова ожидает нажатие кнопки **Вправо** на программируемом модуле, чтобы продолжить работу после прохода челнока. Затем снова срабатывает прибивной механизм. Описанные действия будут повторяться бесконечно, пока человек, работающий со станком, не выключит исполнение программы.

ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

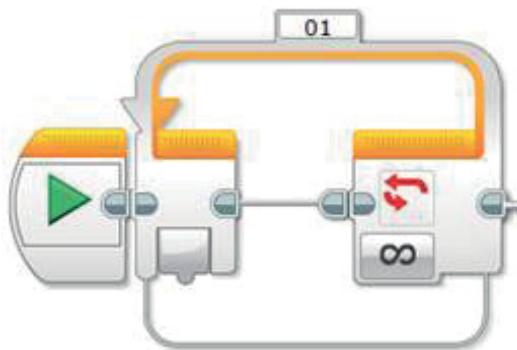
СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ СТАНКА

В открывшемся окне проекта начинай составлять программу для станка.

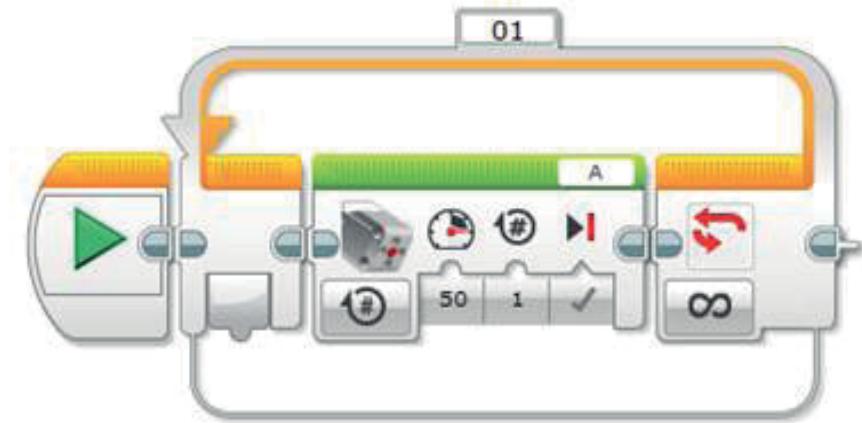
ОБЩИЙ ВИД ПРОГРАММЫ



1. Нужно начать с бесконечного цикла (оранжевый блок), так как тебе необходимо, чтобы работа станка не прекращалась автоматически. Вся твоя программа будет находиться внутри цикла.



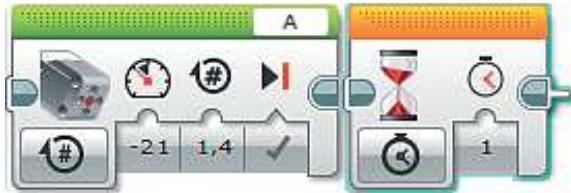
2. Первым делом надо поднять ремизку. За это отвечает средний мотор (зелёный блок).



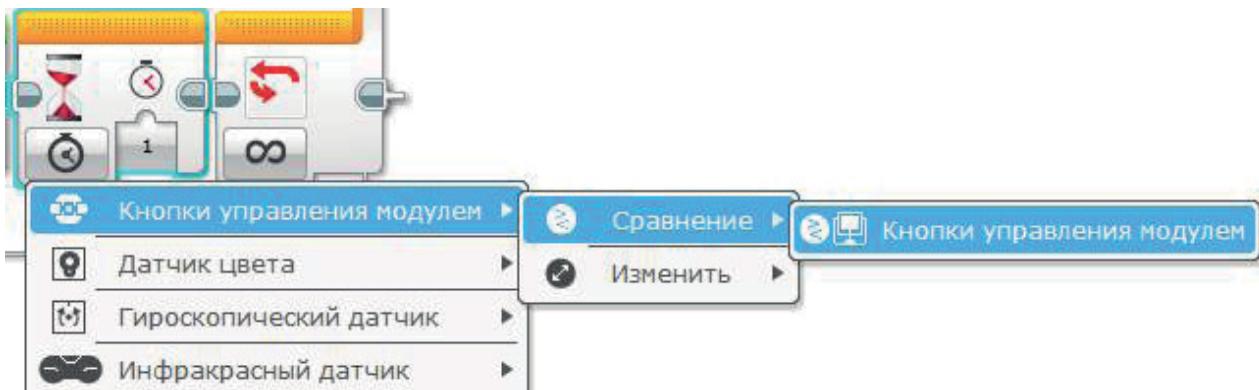
3. Настрой этот блок. По умолчанию он работает по количеству оборотов. Эту опцию не меняй. Мощность: **-21**. Количество оборотов: **1,4**. Также проверь, что мотор настроен на порт A.



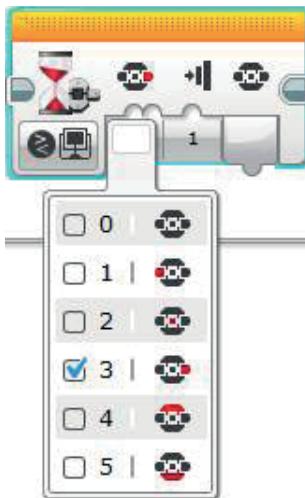
4. Далее добавь команду **Ожидание** (оранжевый блок). Станок будет ожидать провода членка, а затем нажатия кнопки на программируемом модуле.



5. Выбери в этом блоке опцию **Кнопки управления модулем** → **Сравнение** → **Кнопки управления модулем**.



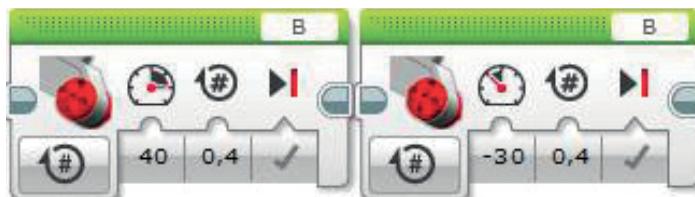
6. Настрой параметры блока. Оставь в параметре **Набор идентификаторов** кнопок модуля галочку только напротив **3**. Остальные настройки оставь без изменений.



7. Добавь команду **Большой мотор** (зелёный блок) и настрой её. Мощность: **40**. Количество оборотов: **0,4**. Проверь, что мотор настроен на порт **B**.



8. Большой мотор отвечает за работу пришивного механизма; значит, нужно вернуть его в начальное положение. Поэтому понадобится вторая команда **Большой мотор** (зелёный блок), но необходимо выставить мощность: **-30**. Количество оборотов: **0,4**. Проверь, что мотор настроен на порт **B**.



9. Затем надо опустить ремизку. Добавь команду **Средний мотор** (зелёный блок). Она будет такой же, как и первая, но нужно выставить мощность: **21**. Проверь, что мотор настроен на порт **A**.

10. Теперь станок должен снова ожидать провода челнока и нажатия кнопки на программируемом модуле. Повтори команду **Ожидание** (оранжевый блок) и настрой её точно так же, как и первую. Проверь подключение к порту **A**.



11. Для завершения одного цикла работы необходимо, чтобы ещё раз сработал пришивной механизм. Для этого добавь ещё две команды **Большой мотор** (зелёный блок) и настрой их точно так же, как и в пунктах 7 и 8. Проверь подключение к порту **B**.



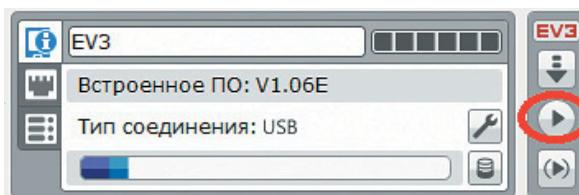
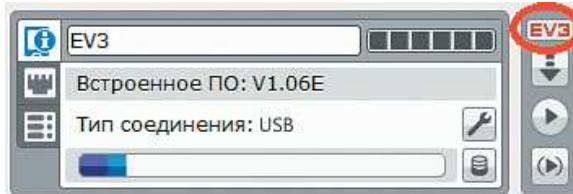
Этап 5. Загрузка программы и её тестирование



ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. После запуска программы должна подняться ремизка.
2. Нажми кнопку **Вправо**. Должно сработать бёрдо: оно наклонится и выпрямится. После этого ремизка опустится.
3. Нажми ещё раз кнопку **Вправо**. Бёрдо наклонится и выпрямится. После этого ремизка поднимется.

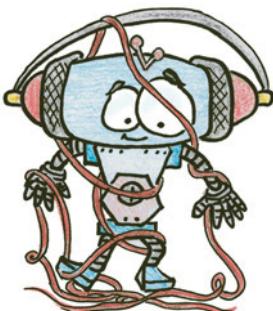
Если всё получилось, значит, тестирование прошло успешно. Молодец!

Если хотя бы один из тестов не получился, скорее всего были допущены ошибки при сборке или написании программы. Начни с проверки программы.

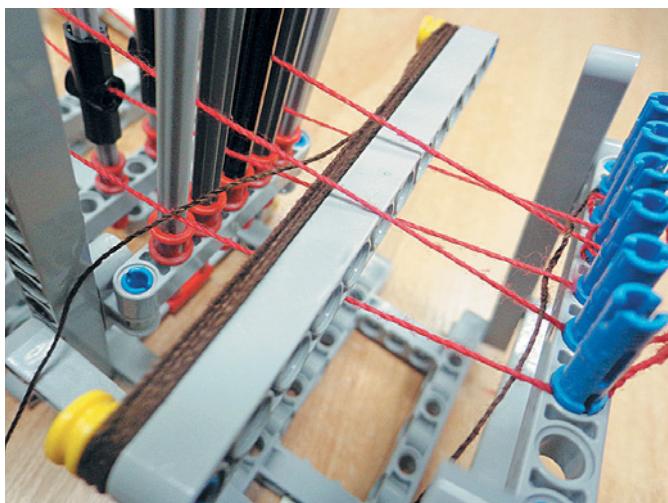
Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

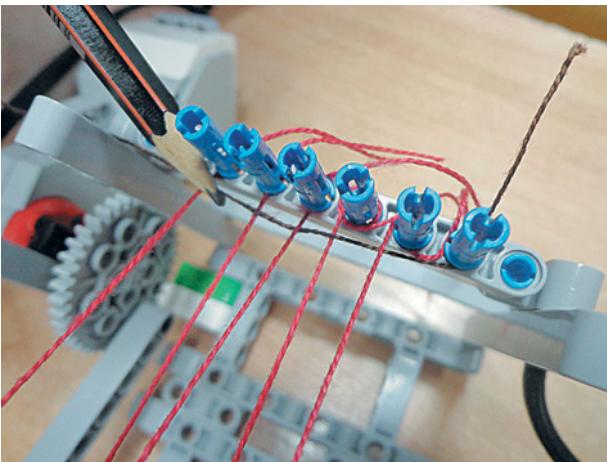
Этап 6. Создаём на станке тканое полотно



1. Включи станок (запусти программу), ремизка поднимется. Продень в образовавшийся зев нитей основы челнок¹.



2. Нажми кнопку **Вправо**; сработает бёрдо.



Внимание!

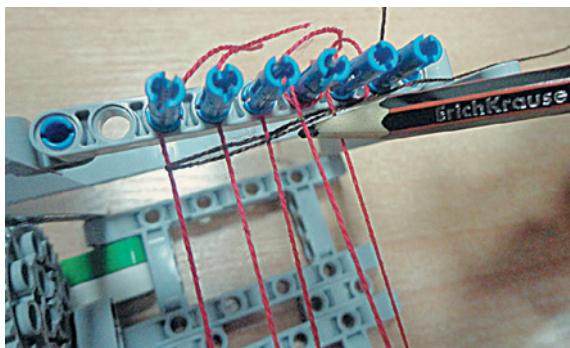
Вначале пришивной механизм может не доставать до начальных рядов нитей, поэтому пришивай их к краю будущего полотна вручную, например с помощью карандаша.

- Продень в образовавшийся зев нитей основы челнок в обратном направлении.



- И снова подровняй ткань.

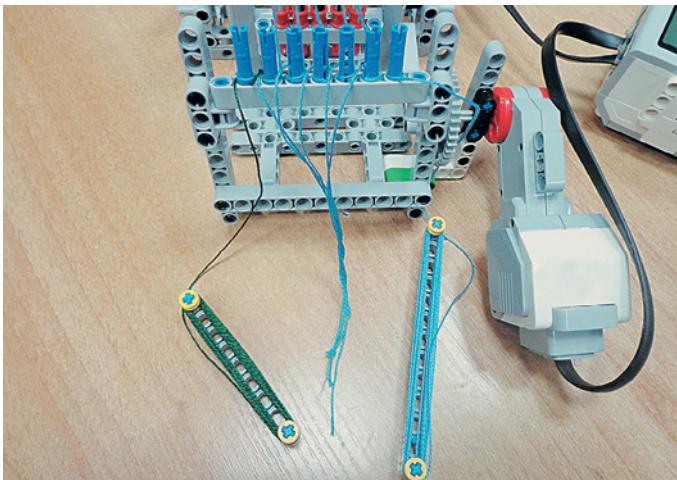
Повторяй эти действия и получишь самое настоящее тканое полотно!



Молодец! У тебя всё получилось! Ты умеешь создавать тканое полотно! Попробуй поэкспериментировать. Сделай ткань с цветным узором.

ПЕРЕКРЁСТНЫЙ УЗОР

- Собери ещё один челнок (он может быть другого размера) и намотай на него нить утка второго цвета. Закрепи вторую нить утка с противоположной стороны.

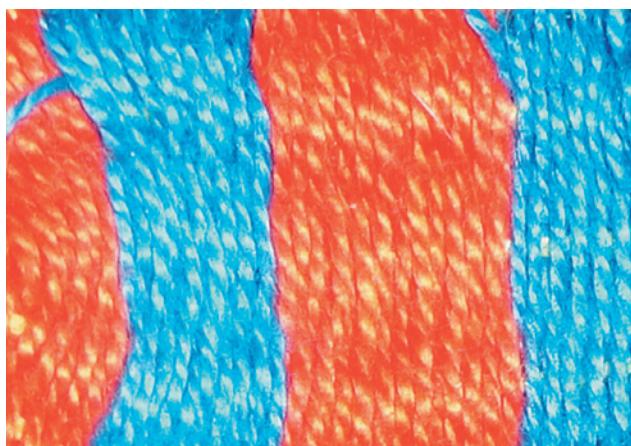


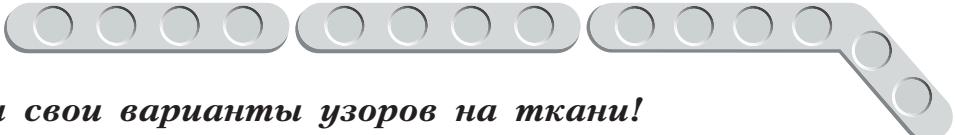
2. Начинай ткать. Продень первый челнок (на рисунке он слева, с зелёной нитью) через зев нитей основы, чтобы он оказался справа. Второй челнок (на рисунке он справа, с голубой нитью) продень через зев нитей основы, чтобы он оказался слева. Затем нажми кнопку **Вправо** на программируемом модуле; сработает бёрдо, затем ремизка опустится.
3. Продолжай «перебрасывать» челноки тем же способом и получишь перекрёстный узор.



ЧЕРЕДОВАННЫЙ УЗОР

1. Тебе снова понадобятся два челнока. Можно использовать другие цвета нити утка, например голубой и оранжевый. Закрепи обе нити так же, как в перекрёстном узоре.
 2. Начни работать только одним челноком и создай участок желаемой длины одного цвета, например оранжевого.
 3. Отложи в сторону первый челнок, возьми второй с другим цветом нити и работай им.
- Меняя челноки, ты получишь красивые полосочки.

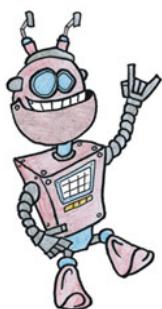




Подумай и предложи свои варианты узоров на ткани!

А чтобы станок работал быстрее, его можно усовершенствовать! Запрограммируй его так, чтобы ремизка опускалась и поднималась не по нажатию кнопки **Вправо**, а по времени. Для этого рассчитай, за сколько секунд ты успеваешь продеть челнок сквозь зев нитей основы, и выставь секунды в блоке **Ожидание**.

А теперь...



На своём ткацком станке ты можешь создавать уникальные украшения для себя и своих друзей!

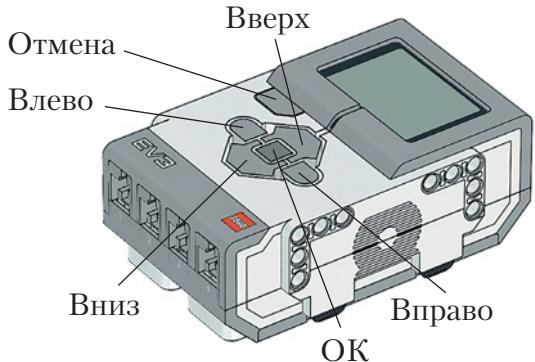
Например, вот такое прекрасное ожерелье:



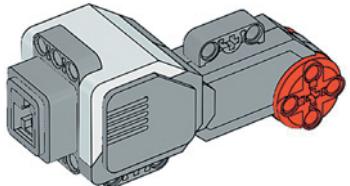
Подумай, какие полезные вещи ты сможешь сделать из кусочков ткани. Может, это будет закладка для книги или броши в подарок маме? Фантазируй — и у тебя всё получится!

Перечень деталей

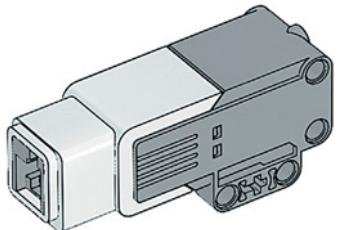
Модуль EV3, 1x



Большой мотор, 1x

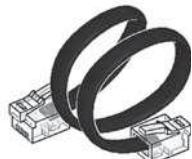


Средний мотор, 1x



Кабель

Кабель, 25 см, 2x



USB-кабель, 50 см, 1x

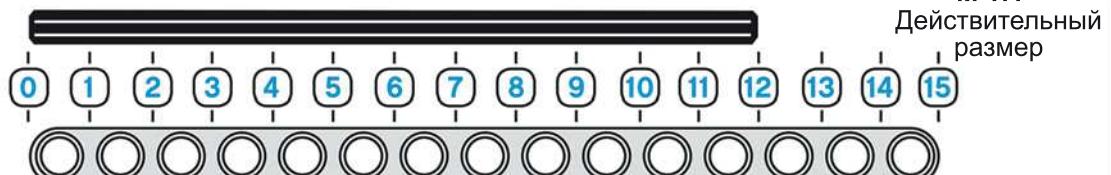


Зубчатые колёса

Зубчатое колесо на 40 зубьев, серое, 1x

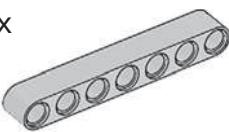


Зубчатое колесо на 12 зубьев, чёрное, 2x

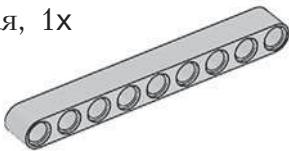


Балки прямые

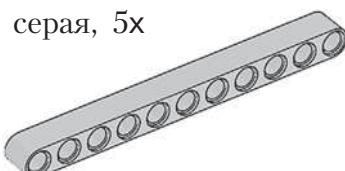
Балка № 7, серая, 4x



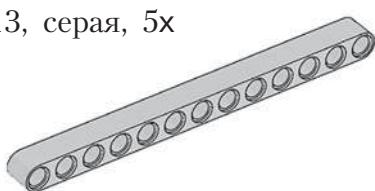
Балка № 9, серая, 1x



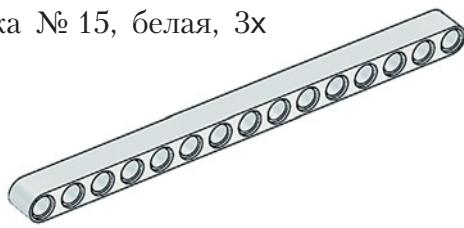
Балка № 11, серая, 5x



Балка № 13, серая, 5x



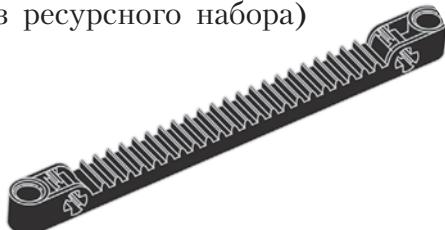
Балка № 15, белая, 3x



Балка № 3, зелёная, 1x

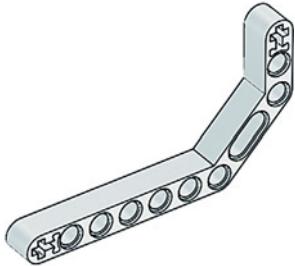


Балка с зубцами № 13, чёрная,
2x (из ресурсного набора)

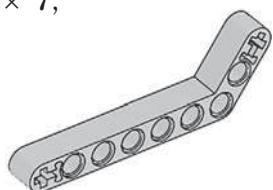


Балки угловые

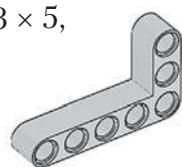
Двойная угловая балка 3 × 7,
белая, 2x



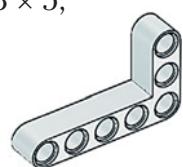
Угловая балка 3 × 7,
серая, 2x



Прямоугольная балка 3 × 5,
серая, 2x



Прямоугольная балка 3 × 5,
белая, 1x



Блоки

Поперечный блок,
2-модульный, серый, 4x



Поперечный блок,
4-модульный, серый, 4x



Поперечный блок,
3-модульный, чёрный, 3x



Штифты

Соединительный штифт,
2-модульный, чёрный, 43х



Соединительный штифт,
2-модульный, синий, 17х



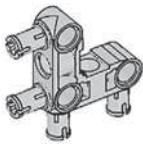
Соединительный штифт,
3-модульный, синий, 15х



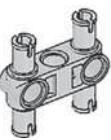
Соединительный штифт
со втулкой, 3-модульный,
красный, 11х



Угловой соединительный
штифт, 3×3-модульный,
серый, 4х



Н-образный соединительный
штифт, 3×1-модульный, 4х



Оси

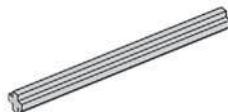
Ось № 3, серая, 6х



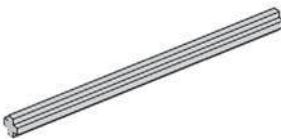
Ось № 5, серая, 4х



Ось № 7, серая, 3х



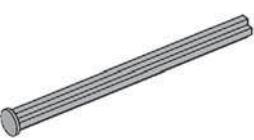
Ось № 9, серая, 2х



Ось № 8, чёрная, 1х



Ось № 8, с головкой, серая,
2х



Втулки

Втулка, $\frac{1}{2}$ -модульная, жёлтая, 4x



Втулка — удлинитель оси, 2-модульная, красная, 1x



Угловой блок 2, 180°, чёрный, 3x

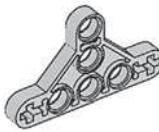


Блоки

Двойной поперечный блок, 3-модульный, чёрный, 1x

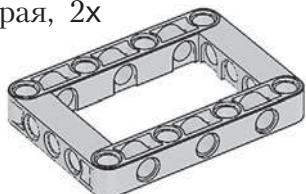


Треугольный фиксатор 3 × 5, серый, 4x

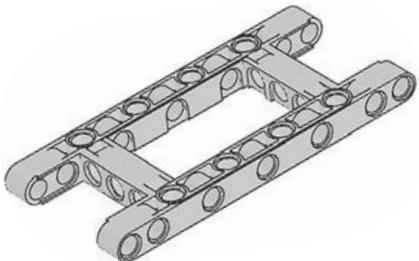


Рамки

Рамка 5 × 7, серая, 2x



Рамка 7 × 11, серая, 2x
(одна — из базового набора,
одна — из ресурсного)



education

Содержание

Дорогой друг!	3
КОРМУШКА ДЛЯ ПИТОМЦА	5
Сытый питомец — довольный питомец	6
Этап 1. Устройство кормушки	9
Этап 2. Сборка автоматической кормушки	10
Шаг 1. Сборка основания кормушки	10
Шаг 2. Сборка креплений ёмкости для корма	13
Шаг 3. Сборка привода дозатора корма	18
Шаг 4. Финальная сборка конструкции кормушки и подключение проводов	27
Этап 3. Установка программного обеспечения на компьютере	37
Этап 4. Создание программы для кормушки	38
Запуск программного обеспечения LME-EV3	38
Создание нового проекта в памяти EV3	38
Логика программы	40
Составление программы для кормушки	41
Этап 5. Загрузка программы и её тестирование	49
Загрузка программы в программируемый модуль	49
Тестирование программы	49
А теперь	51
Перечень деталей	53
КРУТОЕ ПИКЕ	57
Знакомимся с самолётом	58
Навигационные приборы	59
Этап 1. Устройство авиасимулятора	62
Этап 2. Сборка авиасимулятора	63
Шаг 1. Сборка контура жёсткости	63
Шаг 2. Сборка основания рулевого механизма	64
Шаг 3. Сборка рулевого механизма	67
Шаг 4. Сборка штурвала	75
Шаг 5. Соединение рулевого механизма со штурвалом	79

Этап 3. Установка программного обеспечения на компьютере	80
Этап 4. Создание программы для робота-авиасимулятора	81
Запуск программного обеспечения LME-EV3	81
Создание нового проекта в памяти EV3	81
Логика программы	83
Составление программы для робота-авиасимулятора	83
Этап 5. Загрузка программы и её тестирование	140
Загрузка программы в программируемый модуль	140
Тестирование программы	140
Этап 6. От винта!	142
А теперь	145
И гул турбин мы слышим вновь!	145
Перечень деталей	150
КОТОРЫЙ ЧАС?	153
История развития часовочного мастерства	154
Этап 1. Изучение устройства роботизированных часов с кукушкой	160
Этап 2. Сборка роботизированных часов с кукушкой	161
Шаг 1. Сборка первой части редуктора часовочного механизма	161
Шаг 2. Сборка второй части редуктора часовочного механизма	168
Шаг 3. Сборка редуктора часовочного механизма	172
Шаг 4. Сборка основания часовочного механизма	173
Шаг 5. Сборка часовой стрелки	175
Шаг 6. Сборка минутной стрелки	177
Шаг 7. Установка минутной стрелки	179
Шаг 8. Сборка часовочного механизма	181
Шаг 9. Сборка механизма боя часов	182
Шаг 10. Сборка крепления основания часов	183
Шаг 11. Сборка и крепление электропривода часовочного механизма	185
Шаг 12. Сборка крепления для среднего мотора и программируемого модуля EV3, часть 1	187
Шаг 13. Сборка крепления среднего мотора и программируемого модуля EV3, часть 2	189
Шаг 14. Сборка рамки для циферблата часов	192
Шаг 15. Сборка домика для кукушки	194
Шаг 16. Сборка дверки, её крепления и механизма открывания	196



Шаг 17. Установка модуля EV3 и датчика цвета	199
Шаг 18. Сборка управляющего механизма для кукушки	202
Шаг 19. Подключение моторов и датчика	208
Этап 3. Установка программного обеспечения на компьютере	209
Этап 4. Создание программы для роботизированных часов с кукушкой	210
Логика программы	210
Запуск программного обеспечения	210
Создание нового проекта в памяти EV3	211
Составление программы для роботизированных часов с кукушкой	212
Этап 5. Загрузка программы и её тестирование	224
Загрузка программы в программируемый модуль	224
Тестирование программы	224
А теперь	225
Создай собственную идеальную Кукушку!	225
Кукушка, кукушка, сколько мне осталось?	225
Так который час?	228
Перечень деталей	229
ТКАЦКИЙ СТАНОК	235
История ткачества	236
Этап 1. Устройство автоматического ткацкого станка	239
Этап 2. Сборка автоматического ткацкого станка	240
Шаг 1. Сборка основания	240
Шаг 2. Сборка подъёмного механизма ремизки	241
Шаг 3. Сборка основной части ремизки	243
Шаг 4. Соединение основания станка с ремизкой	246
Шаг 5. Сборка рам для закрепления нитей основы	247
Шаг 6. Установка среднего мотора	250
Шаг 7. Сборка бё尔да	252
Шаг 8. Сборка устройства крепления нитей основы	255
Шаг 9. Сборка передаточного механизма бёрда	257
Шаг 10. Крепление нитей основы	260
Шаг 11. Сборка челнока	261
Шаг 12. Подключение моторов	262
Этап 3. Установка программного обеспечения на компьютере	264
Этап 4. Создание программы для станка	265
Логика программы	265
Запуск программного обеспечения LME-EV3	265
Составление программы для станка	265
Общий вид программы	265

Этап 5. Загрузка программы и её тестирование	269
Загрузка программы в программируемый модуль	269
Тестирование программы.	269
Этап 6. Создаём на станке тканое полотно.	270
Перекрёстный узор	271
Чередованный узор	272
А теперь	274
Перечень деталей	275



Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"

Учебное электронное издание

Серия: «РОБОФИШКИ»

**КОНСТРУИРУЕМ РОБОТОВ НА LEGO® MINDSTORMS® EDUCATION EV3.
СБОРНИК ПРОЕКТОВ № 2**

Составитель Ю. А. Серова

Для детей среднего школьного возраста

Ведущий редактор Ю. А. Серова

Художники В. А. Прокудин, Я. В. Соловцова, И. Е. Марев, Ю. Н. Елисеев

Фотосъемка: И. А. Федягин

Технический редактор Т. Ю. Федорова

Корректор И. Н. Панкова

Компьютерная верстка: Е. Г. Ивлева

Подписано к использованию 27.02.20.

Формат 210×260 мм

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: info@pilotLZ.ru, <http://www.pilotLZ.ru>