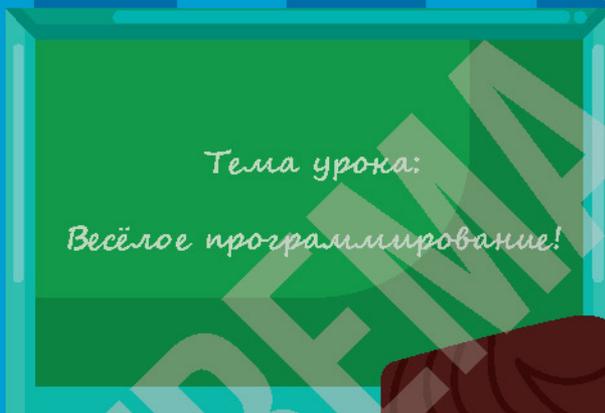


обучающий набор

КВАНТ

TREMA.RU
сделано в России



Содержание

Введение 4



Урок 0: Первое знакомство 6

Урок 1: «Умная» плата 30



Урок 2: Модуль «Светодиод» 52



Урок 3: Модуль «RGB-светодиод» 62

Урок 4: Контроль 66

Проект 1: Гирлянда 70



Урок 5: Модуль «Кнопка» 74

Урок 6: Переменные 78

Урок 7: Операторы 82



Урок 8: Модуль «Зуммер» 88

Проект 2: Светофор 90



Урок 9: Модуль «Датчик освещённости» 94



Урок 10: Сервопривод _____ 98

Проект 3: Шлагбаум _____ 102

Проект 4: Игра «Pong» _____ 108



Урок 11: Модуль «Датчик температуры» _____ 120

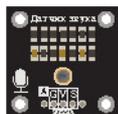


Урок 12: Модуль «Потенциометр» _____ 124

Проект 5: Игра «Ловим яблоки» _____ 130

Урок 13: Измерительные шкалы приборов _____ 140

Проект 6: Климат-контроль _____ 144



Урок 14: Модуль «Датчик звука» _____ 150

Урок 15: Разрешите обратиться? _____ 152

Проект 7: Аплодисментометр _____ 156

Проект 8: Сигнализация _____ 162



Урок 16: Модуль «Датчик расстояния» _____ 170

Урок 17: Автономность _____ 178

Проект 9: Умный дом _____ 184

Что дальше? _____ 194

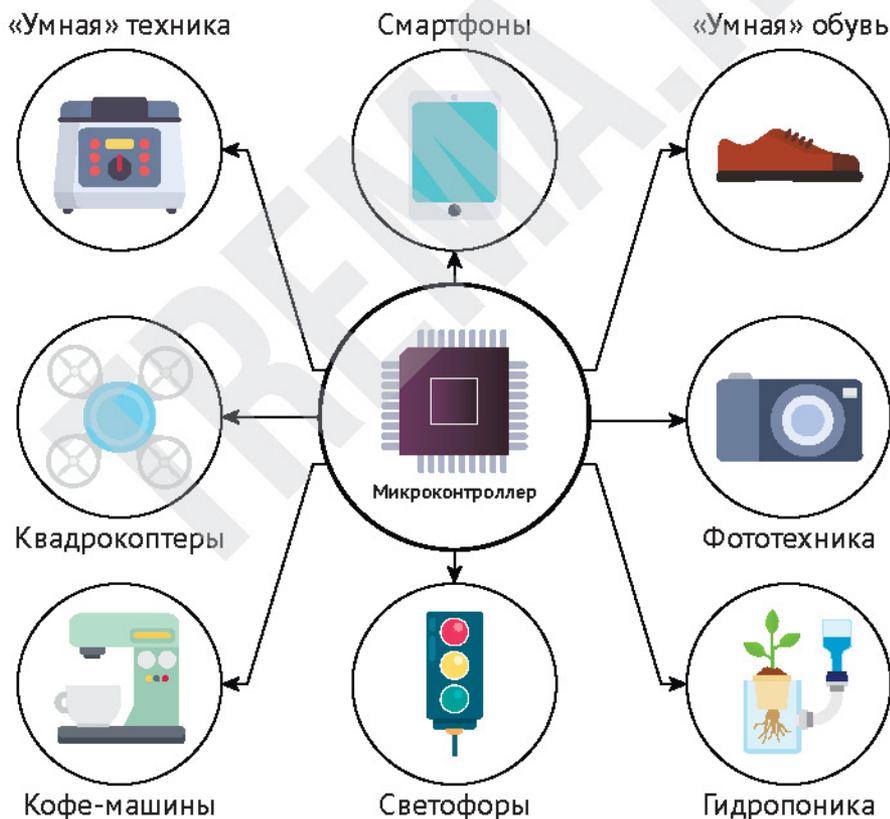
Введение

Привет! Меня зовут **Хаски**.
Я буду рядом на страницах
книги и помогу советом в
любом сложном задании!



Приветствуем тебя, дорогой друг!

В повседневной жизни человек всё чаще взаимодействует со сложными устройствами, использует их в «умных домах», медицинских приборах, а также в широком списке другого оборудования и систем. Всех их объединяет одно – они управляются **микроконтроллерами (МК)**.



Урок 0: Первое знакомство

Скоро ты создашь свою первую программу!



Приветствуем тебя, дорогой читатель! Буквально через несколько страниц перед тобой откроется богатый мир контроллеров и модулей, за которым последует и создание на их основе крутых проектов! Но прежде, чем перейти к «железу», стоит выбрать **среду разработки** (то есть программу для создания скриптов), которую ты будешь использовать далее.

0.1: Среда разработки

«Но ведь сред много, как из них выбрать нужную?» — спросишь ты. Ответ прост: для работы с набором лучше всего подойдёт среда mBlock, которая умеет «общаться» с микроконтроллерами.

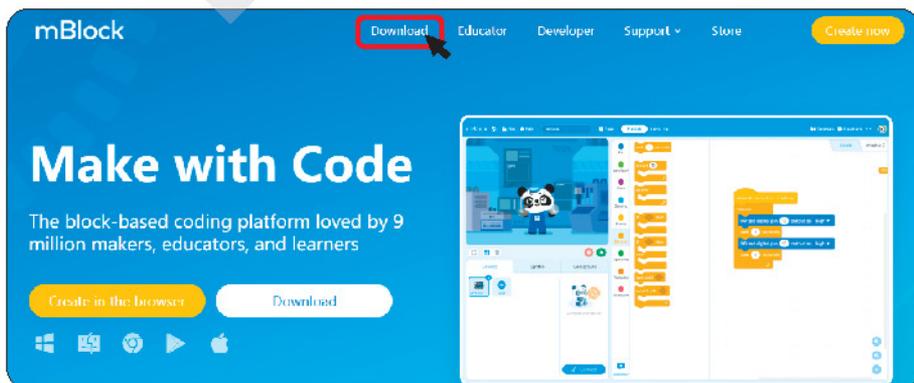
mBlock — **визуальная** среда программирования, где вместо текста используются графические **блоки**, при соединении которых и получаются программы (в mBlock они называются **скрипты** или **сценарии**).

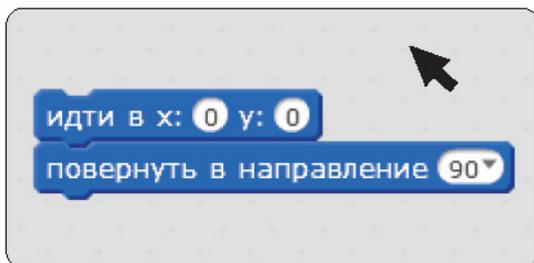
0.2: Установка mBlock

Чтобы создавать скрипты, надо установить среду программирования на компьютер (ПК). Для этого просто двигайся по картинкам-шкагам:

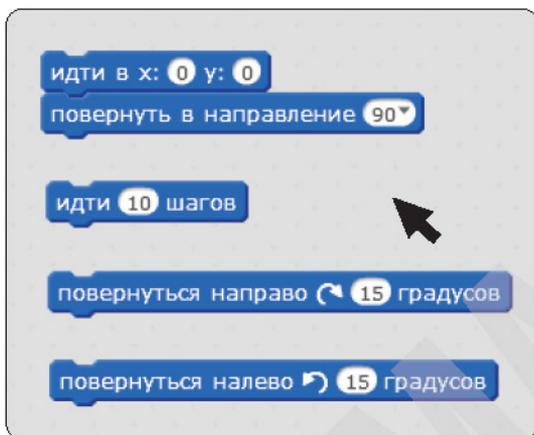
Шаг 1

Зайди на сайт <https://mblock.cc> и выбери пункт меню «Download».

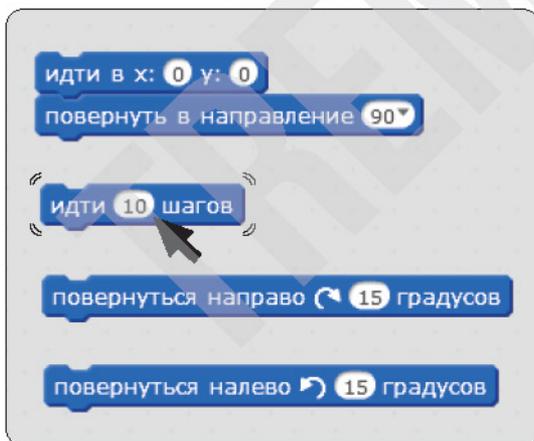




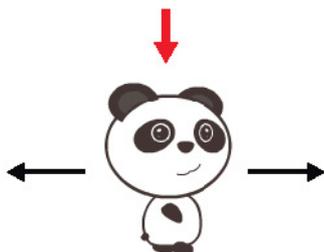
В результате у тебя должна получиться конструкция из блоков как на рисунке.

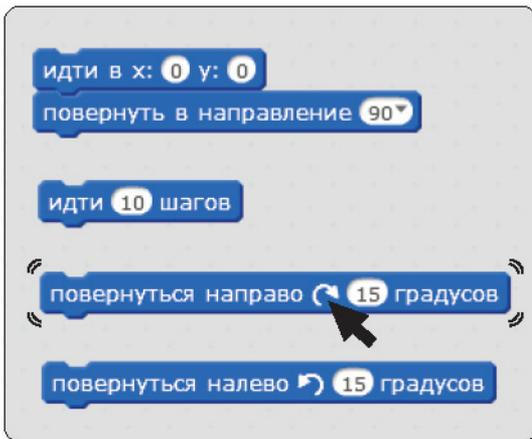


Повторив действия, как на стр.17, вынеси ещё в рабочую область блоки: «идти 10 шагов», «повернуться направо 15 градусов», «повернуться налево 15 градусов».



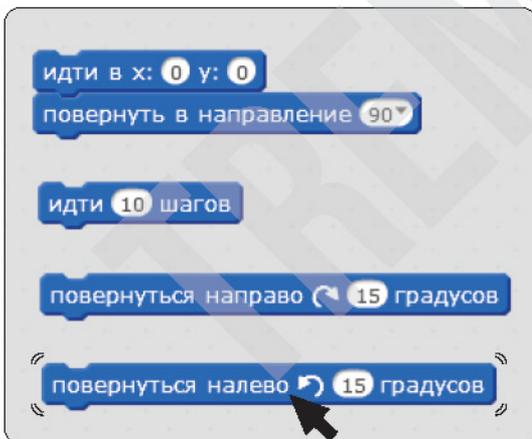
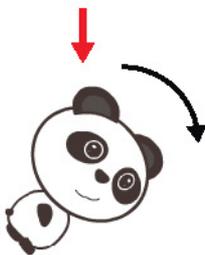
Наведи курсор на блок «идти 10 шагов» и щёлкни мышкой один раз. Обрати внимание на Панду в этот момент. Поэкспериментируй с движением Панды, изменяя число внутри блока. Для того, чтобы Панда двигалась в обратном направлении, добавь к числу знак «минус».





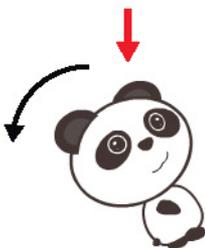
Теперь наведи курсор на блок «**повернуться направо 15 градусов**» и щёлкни мышкой один раз.

Заметь, как изменилось положение спрайта Панды. Поэкспериментируй с числом внутри блока, изменяя угол правого поворота.



Как и предыдущий, данный блок «**повернуться налево 15 градусов**» можно щёлкнуть мышкой.

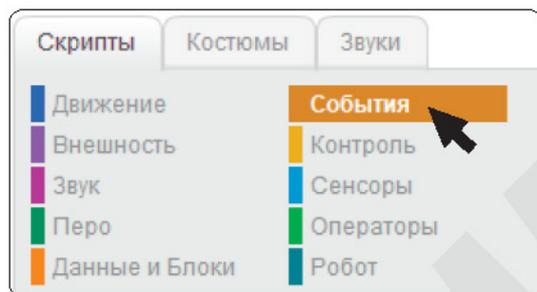
Поэкспериментируй с числом внутри блока, изменяя угол левого поворота.



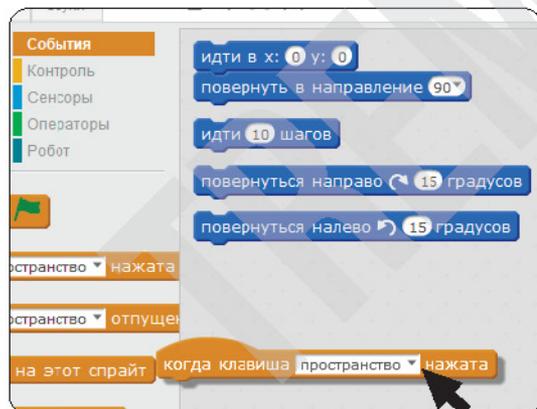
0.5: Управление работой скрипта с клавиатуры

Получившийся скрипт, безусловно, хорош. Однако, пользоваться им не так удобно, как хотелось бы. Для того, чтобы это исправить, добавим в скрипт возможность управления с клавиатуры, когда нажатие на ту или иную клавишу будет выполнять тот или иной набор действий.

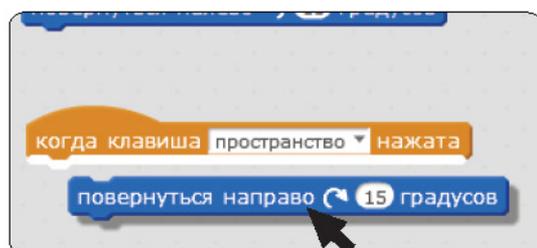
Для этого тебе пригодятся блоки группы «События», внутри которой ты найдёшь 6 блоков типа «шапка», с которых в дальнейшем будет начинаться большая часть твоих скриптов. Блоки данного типа предполагают присоединение к ним всех прочих блоков только с одной стороны (снизу), тем самым они всегда располагаются сверху скрипта. Но довольно теории!



Для начала перейди в группу «События» в верхней центральной части окна программы.



Внеси в рабочую область блок «когда клавиша _____ нажата» (который уже находится в рабочей области) так, чтобы он не соединился с другими блоками.



Перенеси блок «повернуться направо 15 градусов» так, чтобы он «прилип» к блоку «когда клавиша _____ нажата»

Модули, которые подключаются к ЦИФРОВЫМ выводам



Светодиод

Прибор, излучающий свет при прохождении через него электрического тока. Используется для освещения, создания электронных табло и т.д.



Зуммер

Устройство, способное излучать звук. Используется для оповещения, передачи звуковых сообщений и т.д.



RGB - светодиод

Отличается от просто светодиода тем, что имеет в корпусе сразу 3 светодиода (**Red** – красный, **Green** – зелёный, **Blue** – синий). Управляя яркостью каждого, можно получить разные цвета.



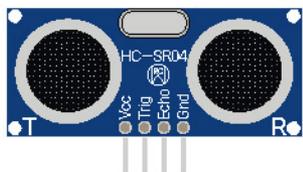
Кнопка

Устройство для замыкания/размыкания цепи. Используется для управления устройствами, ввода значений и т.д.



Сервомотор (сервопривод)

Способен поворачивать свой вал на указанный угол и удерживать его в этом положении. Используется в роботах, станках, поворотных механизмах и т.д.



Ультразвуковой датчик расстояния

Способен измерять расстояние до препятствий в пределах 4 метров. Используется при создании роботов, электронных линеек и т.д.

Модули, которые подключаются к АНАЛОГОВЫМ выводам



Датчик звука

С помощью встроенного в модуль микрофона фиксирует громкость звука. Используется при создании систем управления/охраны.



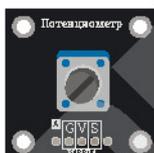
Датчик освещённости

Позволяет определить уровень освещения. Используется в системах автоматического управления светом, сигнализациях и т.д.



Датчик температуры

Датчик, измеряющий температуру окружающей среды в диапазоне от -40°C до $+125^{\circ}\text{C}$.



Потенциометр

Позволяет управлять яркостью, скоростью, громкостью и т.д., так как напряжение на выходе потенциометра напрямую зависит от угла поворота его ручки.

Модули, которые подключаются к ЛЮБЫМ выводам



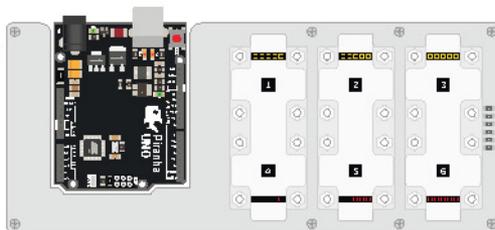
Расширитель выводов

С помощью этого модуля ты сможешь подключить к плате сервоприводы и другие модули или датчики, которые нельзя подключить к рабочей плате напрямую.

1.9: Работа с набором

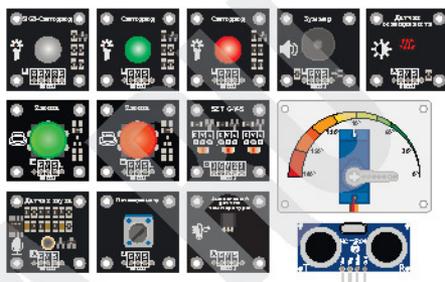
Рабочая плата (РП)

- Имеет в основе плату Piranha UNO;
- Управляет устройствами;
- Подключается к ПК через USB;
- Имеет колодки с выводами **GVS**:
G (GND) – земля; **V** (VCC) – 5 Вольт;
S (SIGNAL) – сигнал.



Модули, датчики и сервопривод

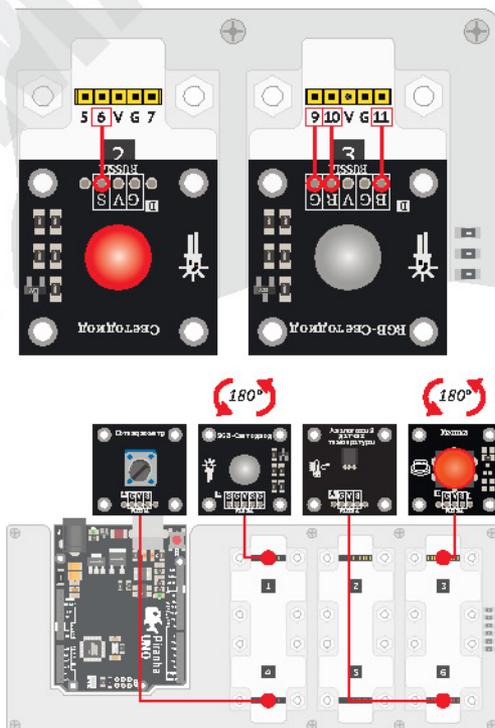
- Выполняют широкий круг задач;
- Подключаются к РП;
- Имеют выводы **GVS** («RGB-светодиод», «Расширитель выводов» и «Датчик расстояния» имеют больше выводов);
- Для удобной работы и большей устойчивости сервопривод установлен в специальный корпус.



Шаг 1

Подключение модулей

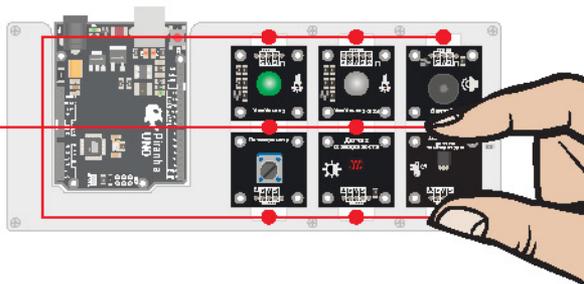
1. Выбери ячейку РП, выводы которой подходят нужному модулю (см. таблицы на [стр.38-39](#))
 2. Перед установкой модуля запомни **цифру**, которая указана около вывода, в который устанавливается модуль – её ты будешь указывать в **скриптах**.
 3. Установи модуль сверху на пластиковые направляющие и убедись, что контактные разъёмы модуля попали в колодку выводов РП.
 4. Надави на модуль до упора.
- Вот и всё, модуль установлен!



Шаг 2

Отключение модулей

1. С обеих сторон от каждого модуля есть специальные **пазы**, с помощью которых модули можно отключить от РП. Поддень модуль и движением «вверх» отключи модуль.



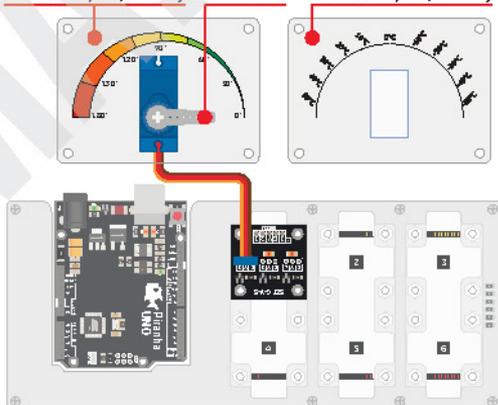
Шаг 3

Работа с сервоприводом

1. Установи «Расширитель выводов» в любую колодку РП.
2. Вставь разъём сервопривода в любую колодку «Расширителя выводов».
3. Чтобы сменить рисунок насадки, сними её, переверни и установи обратно на сервопривод. Качалку (пластиковая стрелка, показывающая угол поворота) **снимать не надо**.
4. Качалка уже снята? Не страшно! Установи качалку примерно так же, как она стояла, а чуть позже ты её откалибруешь.



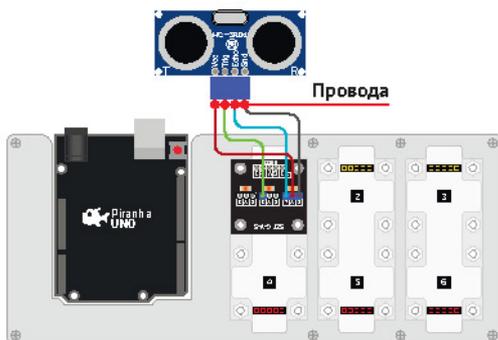
Насадка (сторона 1) Качалка Насадка (сторона 2)



Шаг 4

Работа с датчиком расстояния

1. Ножки «Датчика расстояния» подключаются к «Расширителю выводов» в следующем порядке:
 - **Vcc** к выводу **V** «Расширителя»
 - **Trig** к выводу **2** «Расширителя»
 - **Echo** к выводу **1** «Расширителя»
 - **Gnd** к выводу **G** «Расширителя»



Урок 2: Модуль «Светодиод»

Вот и пришло время знакомиться с возможностями модулей! И первыми в списке будут модули «Светодиод».

Краткое описание:

Модуль работает с **цифровыми выходами**, а используется для индикации, освещения и т.д. И, к слову, в конце урока ты создашь кое-что интересное!



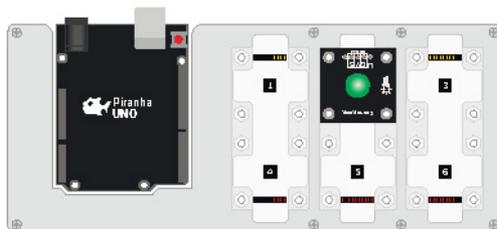
2.1: Работа с цифровым сигналом

Пример модуля из жизни – **фонарик**: нажимаешь на клавишу «Вкл» – свет включился, нажимаешь на клавишу «Выкл» - выключился. Вот и ты сейчас в скрипте соберёшь прототип фонарика, который управляется с клавиатуры!

Хотя тебе уже знакомо управление с клавиатуры (**стр. 22**), иногда надо нажатием одной клавиши запустить сразу несколько скриптов. Для этого есть блоки «**передать _**» и «**когда я получил _**», с помощью которых из одного скрипта запускают другой скрипт или сразу несколько скриптов. Сейчас мы сделаем простой наглядный пример работы этой функции, но ты можешь добавить в скрипт всё, что угодно, чтобы попрактиковаться!

Шаг 1

Установи модуль в ячейку **2**, как показано на рисунке. Внимательно посмотри, в какие **выводы** ячейки **2** будет подключен модуль. В данном случае за подачу управляющего сигнала отвечает **вывод (pin) 6** – его и укажем в скрипте!



Шаг 2

Создай из блоков скрипт:

когда я получил 1

установить цифровой pin 6 udgang как HIGH

когда клавиша стрелка вверх нажата

передать 1

когда я получил 0

установить цифровой pin 6 udgang как LOW

когда клавиша стрелка вниз нажата

передать 0

Шаг 3

Данный скрипт **не надо запускать**. Просто нажимай клавиши <Вверх> и <Вниз> на клавиатуре ПК, включая или выключая светодиод.

Чтобы передать сообщение, в блоке **передать** нажми на **черный треугольник**, выбери **«новое сообщение»** и в новом окне укажи текст сообщения для передачи, который в этом скрипте будет «0» или «1».



Помимо включения, можно управлять яркостью светодиода. Сделать это можно с помощью ШИМ-сигнала, о котором мы сейчас и поговорим.

2.2: Работа с ШИМ-сигналом

Чтобы освежить память, можешь вернуться на **стр. 35-36** и вспомнить, как же устроен ШИМ-сигнал.

Это важно знать!



ШИМ-сигнал может принимать значение **от 0 до 255** (всего **256** значений, так как **0** — это тоже значение).

В данной главе ты учишься работать с **выходным сигналом**, а о **входном сигнале** более подробно мы поговорим чуть позже.

В этом примере ты создашь аналог **клавишного диммера** — электронного устройства, используемого для регулировки яркости свечения лампочек или светодиодов, где каждая клавиша отвечает за уровень яркости.

Шаг 1

Модуль «Светодиод» остаётся в той же ячейке, в которой был установлен. Поэтому создай новый проект и переходи к созданию нового скрипта!

Шаг 2

Создай из блоков следующий скрипт:

когда клавиша нажата
установить PWM/ШИМ pin 6 udgang как

когда клавиша нажата
установить PWM/ШИМ pin 6 udgang как

когда клавиша нажата
установить PWM/ШИМ pin 6 udgang как

когда клавиша нажата
установить PWM/ШИМ pin 6 udgang как

когда клавиша нажата
установить PWM/ШИМ pin 6 udgang как

Огрехи перевода!
Данный термин
значит не что иное,
как клавиша
<Пробел>.



Ищи эти блоки в группе
Робот

В этом скрипте создаётся пять уровней яркости, в каждом из которых значение ШИМ-сигнала разное. При нажатии на одну из клавиш на светодиоде устанавливается соответствующее ей значение яркости. Сами значения можно представить в следующем виде:

Значение **255** = 100% яркости – светодиод горит максимально ярко;

Значение **191** = 75% яркости;

Значение **127** = 50% яркости – светодиод горит в половину мощности;

Значение **63** = 25% яркости;

Значение **0** = 0% яркости – светодиод погас.

В группе **Внешность** есть блоки:

спрятаться

показаться

Они отвечают за **отображение костюмов** в области анимации. Разместив их в рабочей области нужного спрайта, ты сможешь скрыть/отобразить его костюм.



Когда скрипт создан, жми на любую из кнопок, указанных в нём, и смотри, как будут меняться состояние модуля «Светодиод» и твой спрайт в области анимации.



Каверзные вопросы



- Какие виды сигналов ты знаешь? В чём их отличие друг от друга?
- Как ты думаешь, сигнал ШИМ – цифровой или аналоговый?
- Как, по твоему мнению, работает блок **передать** ?
- Определи опытным путём, имеется ли ограничение на количество передаваемых сообщений и, если да, то какое?
- В чём, на твой взгляд, основная функция блока **когда я получил** ?
- Проверь, есть ли ограничение на количество спрайтов, которые можно создать?
- Также опытным путём проверь, есть ли ограничение на количество костюмов у одного спрайта?
- Если ты не удалил спрайт Панды, а только скрыл его, то добавь блоки **спрятаться** и **показаться** в скрипт спрайта «M-Panda» так, чтобы при включении светодиода с любым уровнем яркости спрайт Панды был скрыт, а при выключении светодиода он снова стал виден.

Урок 3: Модуль «RGB-Светодиод»

В данном уроке мы поговорим про модуль, внешне очень похожий на модуль «Светодиод», но который устроен совершенно иначе внутри.

Краткое описание:

Модуль «**RGB-Светодиод**» может работать с **цифровыми выходами**, а в его корпусе стоит сразу три разных светодиода: **красный (Red)**, **зелёный (Green)**, **синий (Blue)**. Благодаря этому, управляя яркостью каждого из них, можно получить миллионы оттенков цветов!



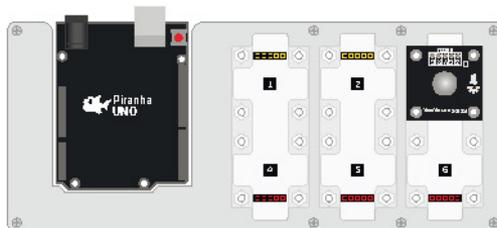
3.1: Работа с цифровым сигналом

Современные смартфоны имеют **индикацию событий**, когда светодиод в его корпусе горит **синим** при входящих сообщениях, **красным** при разряде батареи и **зелёным** при полном заряде. Сейчас ты воссоздашь работу такой индикации, заставив RGB-светодиод светиться разными цветами!

Шаг 1

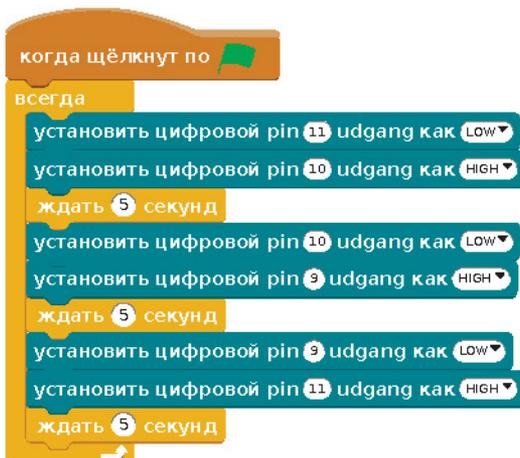
Для начала подключи модуль в ячейку **3**, как показано на рисунке.

Так как в корпусе этого модуля находится сразу **три светодиода**, то и отправлять **0** или **1** ты сможешь на каждый из них!



Шаг 2

Создай новый проект и в нём создай скрипт, как указано далее.



Ты можешь расположить **цвета** светодиода в том порядке, в каком тебе больше нравится. Также ты можешь изменить время ожидания, чтобы цвета менялись быстрее или медленнее.



После запуска скрипта светодиода внутри модуля будут включаться в следующем порядке:

- **красный** на пятую секунду;
 - **зелёный** на пятую секунду;
 - **синий** на пятую секунду
- и так по кругу бесконечно долго.

3.2: Работа с ШИМ-сигналом

Сейчас ты познакомишься с главной «изюминкой» модуля: при использовании **ШИМ-сигнала** можно получить **большее** количество цветов, чем при использовании **цифрового сигнала**: в этот раз скрипт выполнит смену восьми цветов, пять из которых получены за счёт смешивания **красного**, **зелёного** и **синего** цветов.

Шаг 1

Создай следующий скрипт:

когда щёлкнут по 

передать красный и ждать

передать зелёный и ждать

передать синий и ждать

передать жёлтый и ждать

передать фиолетовый и ждать

передать голубой и ждать

передать оранжевый и ждать

передать белый и ждать

передать выключить

остановить этот скрипт

когда я получил красный

установить PWM/ШИМ pin 9 udgang как 0

установить PWM/ШИМ pin 10 udgang как 255

установить PWM/ШИМ pin 11 udgang как 0

ждать 3 секунд

остановить этот скрипт

когда я получил зелёный

установить PWM/ШИМ pin 9 udgang как 255

установить PWM/ШИМ pin 10 udgang как 0

установить PWM/ШИМ pin 11 udgang как 0

ждать 3 секунд

остановить этот скрипт

когда я получил синий

установить PWM/ШИМ pin 9 udgang как 0

установить PWM/ШИМ pin 10 udgang как 0

установить PWM/ШИМ pin 11 udgang как 255

ждать 3 секунд

остановить этот скрипт

когда я получил жёлтый

установить PWM/ШИМ pin 9 udgang как 255

установить PWM/ШИМ pin 10 udgang как 255

установить PWM/ШИМ pin 11 udgang как 0

ждать 3 секунд

остановить этот скрипт

когда я получил фиолетовый

установить PWM/ШИМ pin 9 udgang как 0

установить PWM/ШИМ pin 10 udgang как 255

установить PWM/ШИМ pin 11 udgang как 255

ждать 3 секунд

остановить этот скрипт

когда я получил голубой

установить PWM/ШИМ pin 9 udgang как 255

установить PWM/ШИМ pin 10 udgang как 0

установить PWM/ШИМ pin 11 udgang как 255

ждать 3 секунд

остановить этот скрипт

когда я получил оранжевый

установить PWM/ШИМ pin 9 udgang как 127

установить PWM/ШИМ pin 10 udgang как 255

установить PWM/ШИМ pin 11 udgang как 0

ждать 3 секунд

остановить этот скрипт

когда я получил белый

установить PWM/ШИМ pin 9 udgang как 255

установить PWM/ШИМ pin 10 udgang как 255

установить PWM/ШИМ pin 11 udgang как 255

ждать 3 секунд

остановить этот скрипт



В этом скрипте ты впервые используешь блок

остановить этот скрипт

Он пригодится тебе тогда, когда надо выполнить определенный сценарий **только один раз**.



Каверзные вопросы



- Почему стало возможным получить из трех цветов семь?
- Как, по твоему мнению, работает блок `передать` и `ждать` ?
- Измени скрипт на **стр. 63** так, чтобы загорались цвета, которые ты получал с помощью ШИМ-сигнала. Все ли из них ты можешь получить? Почему?
- Определи опытным путём, какие ещё цвета можно получить, задавая различные значения ШИМ-сигналов для каждого из светодиодов модуля «RGB-светодиод»?

Урок 5: Модуль «Кнопка»

До этого момента тебе встречались модули, которые работали с **выходным сигналом**. Пришло время познакомиться с модулями, которые работают иначе.

Краткое описание:

Модуль «Кнопка» может управлять различными процессами: выключать и включать устройства, запускать выполнение скрипта и многое другое. Модуль работает с платой с помощью **цифрового сигнала**.



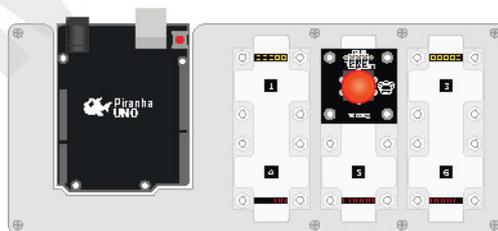
5.1: Работа с модулем

Шаг 1

Для начала подключи модуль в ячейку 2, как показано на рисунке.

Шаг 2

Создай следующий скрипт:

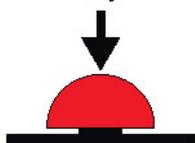


После запуска скрипта ты больше не видишь Панду в области анимации. Однако, при нажатии на кнопку она снова появится. Это связано с тем, что кнопка работает с цифровым сигналом, который имеет два значения (0 или 1): при нажатии сигнал равен 1, а при отпускании равен 0.

Кнопка модуля не нажата



Кнопка модуля нажата



5.2: Примеры работы модуля

В следующих нескольких примерах ты увидишь, какими ещё возможностями обладает данный модуль.

Шаг 1

Создай новый проект и скрипт, принцип работы которого будет таким: до тех пор, пока ты держишь «Кнопку» нажатой, светодиод, расположенный на плате Piranha UNO, горит, а как только ты её отпустишь, светодиод погаснет.



когда щёлкнут по 

всегда

если **читать цифровой pin 14 = 1** то

установить цифровой pin 3 udgang как HIGH

если **читать цифровой pin 16 = 1** то

установить цифровой pin 10 udgang как HIGH

если **читать цифровой pin 14 = 1** и **читать цифровой pin 16 = 1** то

установить цифровой pin 5 udgang как HIGH

установить цифровой pin 6 udgang как HIGH

установить цифровой pin 7 udgang как HIGH

если **читать цифровой pin 14 = 0** и **читать цифровой pin 16 = 0** то

установить цифровой pin 3 udgang как LOW

установить цифровой pin 10 udgang как LOW

установить цифровой pin 5 udgang как LOW

установить цифровой pin 6 udgang как LOW

установить цифровой pin 7 udgang как LOW

Скоро я подробно расскажу тебе о том, когда и зачем нужны эти зелёные блоки, а пока просто помни, что они находятся в группе **Операторы**.



Для того, чтобы создавать более сложные скрипты, предлагаем тебе подробнее познакомиться с группой блоков **Данные и Блоки**. О них и пойдёт речь в следующем уроке.



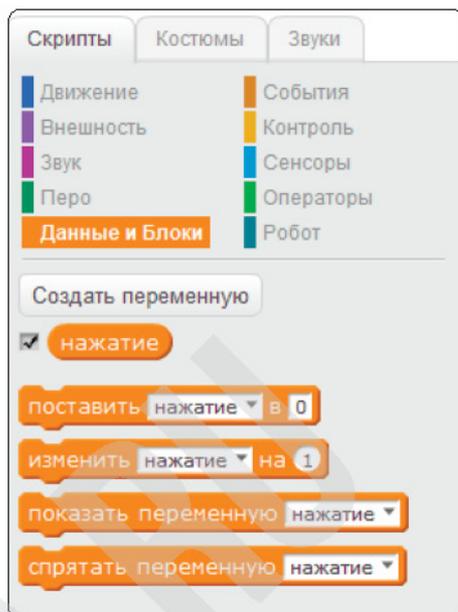
Каверзные вопросы



- С каким типом сигнала работает модуль «Кнопка»?
- Создай самостоятельно скрипт, в котором светодиоды будут загораться не при нажатии, а при отпускании кнопки.

Шаг 3

Переменная «**нажатие**» создана! В группе **Данные и Блоки** появились новые блоки. Давай разберёмся, как с ними работать.



6.2: Описание блоков

В группе «**Данные и блоки**» тебе пока понадобятся только блоки, которые появляются при создании переменной. Вот о них и поговорим:

поставить нажатие в 0

— делает переменную «**нажатие**» равной указанному значению (в данном случае сделать равной **0**). Значением может быть как **число**, так и **текст**.

изменить нажатие на 1

— меняет значение переменной «**нажатие**» на указанную величину (в данном случае — увеличить на 1). Величина может быть **положительным** или **отрицательным** числом.

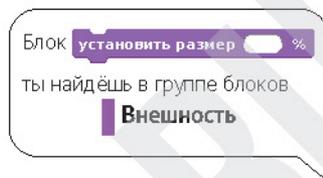
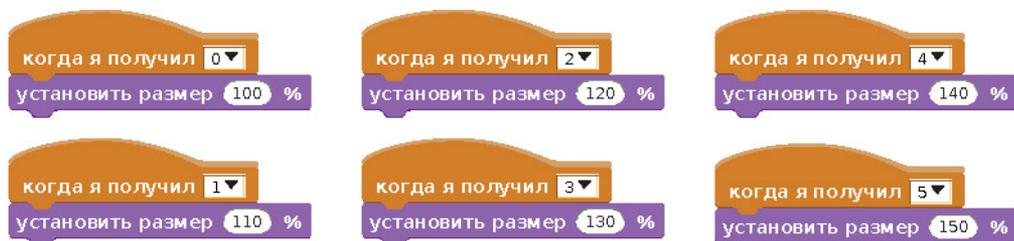
показать переменную нажатие

— выводит значение переменной «**нажатие**» в область анимации.

спрятать переменную нажатие

— скрывает значение переменной «**нажатие**» в области анимации.

Скрипт для спрайта «Квадрат»



Смело жми на флажок, чтобы запустить скрипт, а уже после этого переходи к управлению с помощью клавиш клавиатуры!



Каверзные вопросы



- Что такое «переменная» и зачем она нужна?
- Придумай и создай свои варианты использования переменных.
- Обрати внимание на имена переменных в скриптах. Как ты думаешь, зачем давать им такие имена?
- Создай свой собственный скрипт, который будет использовать больше одной переменной.

округлить — Округлить значение.

Блок используется в тех случаях, когда ты получаешь дробное значение, а нужна только целая часть.

Это важно знать!



Важное правило:

— Цифра после запятой **меньше 5** — число округляется в **меньшую** сторону.

Пример: **4.3** округляется до 4;

— Цифра после запятой **больше или равна 5** — число округляется в **большую** сторону.

Пример: **4.7** округляется до 5.

sqrt от 10 — Список математических функций.

Данный блок включает в себя 14 арифметических и тригонометрических операции. Опишем основные из них:

abs — возвращает модуль указанного числа (возвращает только число без учёта знака, стоящего перед ним);

sqrt — возвращает квадратный корень из указанного числа;

sin — возвращает значение синуса указанного числа (в радианах);

cos — возвращает значение косинуса указанного числа (в радианах);

10^ — возвращает число 10, возведённое в степень указанного числа.



Каверзные вопросы



- Какие типы операторов ты знаешь?
- Почему числа, которые компьютер генерирует случайным образом, называются «псевдослучайными»?
- Что значит «округлить значение числа» и по каким правилам это происходит?
- Что такое «остаток от деления»?

Урок 8: Модуль «Зуммер»

Может ли интересный проект обойтись без звукового сопровождения? Правильно, не может, поэтому знакомься — модуль «Зуммер»!

Краткое описание:

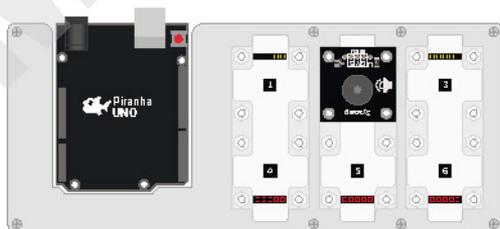
Модуль «Зуммер» имеет встроенный генератор, для включения которого на вывод модуля подаются цифровой сигнал. Зуммер используют в играх, сигнализациях или просто для подачи звуковых сигналов в своё удовольствие (и назло другим)!



8.1: Работа с модулем

Шаг 1

Для начала подключи модуль в ячейку 2, как показано на рисунке.



Шаг 2

Создай новый скрипт, который будет включать и выключать «Зуммер».



Надеюсь, тебе достаточно быстро стало понятно, что модуль исправен.



Как мы сказали в предисловии к данной главе, модуль из набора имеет встроенный генератор. Однако, есть и такие модели зуммеров, не имеющие генератора.

При работе с модулями **без генератора** ты не просто включаешь и выключаешь «Зуммер», а задаёшь тональность, с которой он будет работать. Благодаря этому, на таком модуле можно даже проигрывать мелодии! «А что такое задание тональности?» – спросишь ты.

«**Задание тональности**» – это, если очень грубо, **задание частоты, с которой модуль будет издавать звук при включении.**

Шаг 3

Модуль из набора не способен менять свою тональность, однако в виде эксперимента мы предлагаем тебе создать следующий скрипт, в котором будет постепенно меняться значение ШИМ-сигнала. Звук модуля будет меняться, имитируя изменение тональности «Зуммера» без генератора. Для начала создай переменную **«тональность»**, а затем уже весь скрипт:



Если ты всё же решишь создать устройство, с помощью которого можно будить друзей/родителей/соседей, то сразу хочу тебя предупредить:

блок **всегда** будет

как нельзя кстати (только я тебе об этом не говорил!).



В зависимости от проекта и того модуля, который будет у тебя в руках, в своих будущих самостоятельных проектах ты будешь пользоваться или первым, или вторым способом управления работой «Зуммера».



Каверзные вопросы

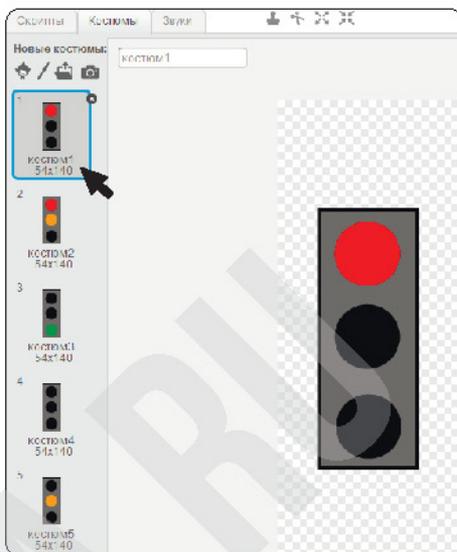


- Для чего нужен «Зуммер»?
- С какими видами сигналов он может работать?
- На что влияет подача на вывод модуля не цифрового, а ШИМ-сигнала?

2.3: Создание анимации

Этот блок будет с сюрпризом, но мы к этому вернёмся чуть позже. Для начала, создай новый спрайт «Светофор», у которого есть пять одинаковых костюмов, а затем раскрась каждый из них следующим образом:

- 1 костюм — горит только **красный**;
- 2 костюм — горит **красный** и **жёлтый**;
- 3 костюм — горит только **зелёный**;
- 4 костюм — не горит ни **один** свет;
- 5 костюм — горит только **жёлтый**;



Теперь пришло время создать для нового спрайта скрипт. И вот тут тебя ждёт сюрприз — в этот раз тебе придётся самостоятельно его создать, без наших подсказок (разве что Хаски даст несколько дельных советов). Что ж, в добрый путь!



Помнишь, как в проекте «Гирлянда» (стр. 70) шары на ёлке меняли свои цвета? Тут тот же принцип: в тот момент, когда должен загореться очередной свет на светофоре, ты просто меняешь костюм на тот, который подходит в данный момент. Вот и всё!

Все необходимые блоки, как и в прошлый раз, ты найдёшь в группе **Внешность**





Каверзные вопросы

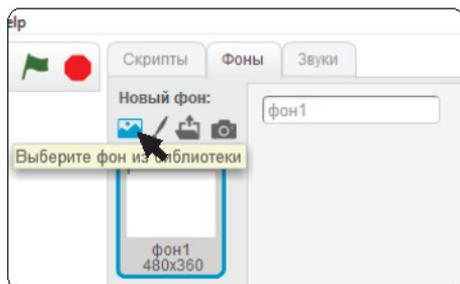


— Добавь в ячейку **4** модуль «Светодиод» **зелёного цвета**, а в ячейку **5** — модуль «Светодиод» **красного цвета**. Добавь в скрипт и в анимацию новые блоки так, чтобы работало **два светофора**: трёхцветный теперь будет аналогом настоящего светофора для автомобилей, а вот двухцветный будет для пешеходов. И не забудь, что теперь и «Зуммер» должен работать вместе с другим светофором!

TREMA.RU

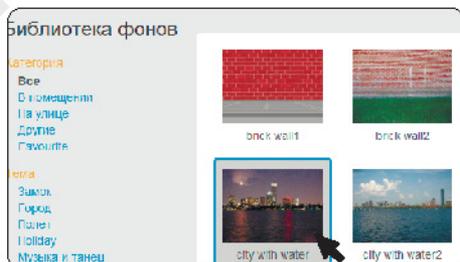
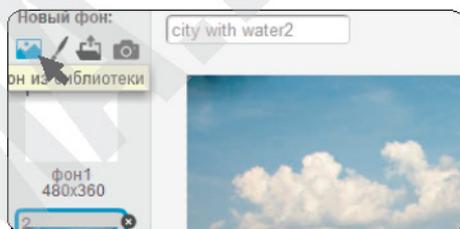
Шаг 1

В начале, надо добавить фон для области анимации. Для этого зайти в спрайт «Сцена», перейти на вкладку «Фоны» и нажми кнопку «Выберите фон из библиотеки».



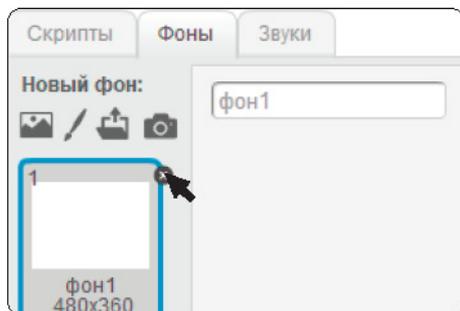
Шаг 2

Перед тобой откроется список всех имеющихся в программе фонов для сцены. Выберем таких два фона, чтобы на них была видна смена дня и ночи. Сначала выберем дневной фон. Запомни имя фона – в дальнейшем тебе это пригодится! После этого добавь ночной фон, повторив все те действия, что и с первым фоном.



Шаг 3

Чтобы тебя не сбивал с толку белый фон, его можно просто удалить. Для этого наведи курсор на фон и нажми на крестик в правом верхнем углу.



Шаг 4

Вернись во вкладку «Скрипты» спрайта «M-Panda» и измени свой скрипт:



когда щёлкнут по [флаг]

поставить [показания датчика] в [0]

всегда

поставить [показания датчика] в [читать аналоговый pin (A)]

если [показания датчика < 450] то

сказать [Ночь!]

перейти к фону [city with water]

иначе

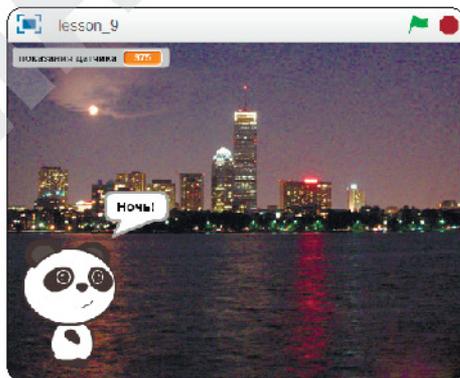
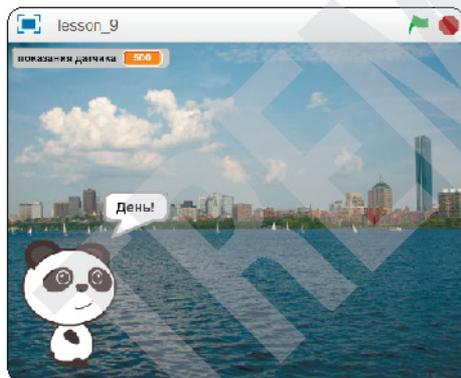
сказать [День!]

перейти к фону [city with water2]

Вполне возможно, что в этом поле тебе надо будет изменить значение на своё. Ты всегда можешь подобрать его опытным путём!



Если в твоём классе или комнате светло, то задёрни шторы, запусти свой скрипт, а после этого выключи в кабинете/комнате свет и смотри, что будет происходить в твоей виртуальной комнате при изменении освещённости!



Каверзные вопросы

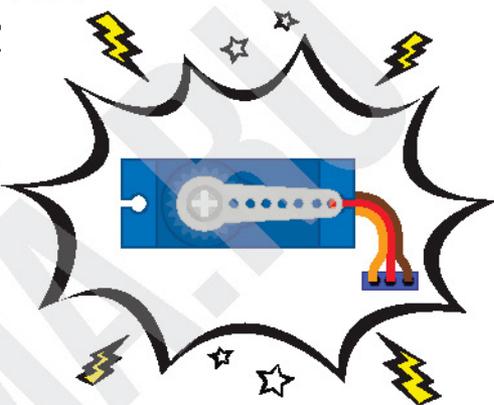


— С какими типами сигналов ты работаешь в этом скрипте и почему? Какие значения принимает тот или иной тип сигнала? Почему?

Урок 10: Сервопривод

Игрушки, роботы, станки и другие знакомые тебе вещи могут иметь в своей конструкции **сервопривод** — механизм с мотором, который можно повернуть на нужный угол и задать ему удерживать это положение.

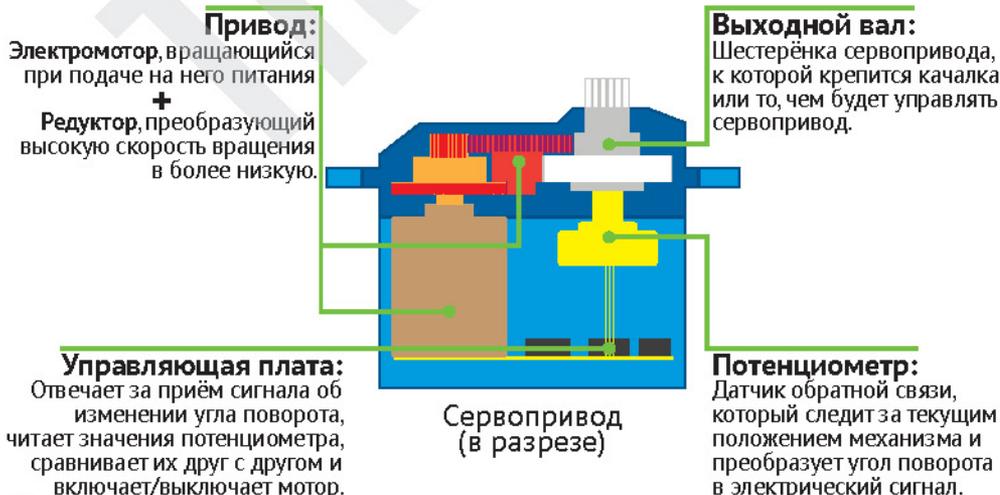
Есть разные сервоприводы, например, модели **постоянного вращения** (могут вращаться на 360° и более) или те, что имеют **предел угла поворота** (могут вращаться только на $45^\circ/90^\circ$ /и т.д.). В наборе идёт модель с пределами угла поворота **от 0° до 180°** .



10.1: Принцип работы

Принцип работы сервопривода:

- сервопривод получает на вход параметр, например, угол поворота вала;
- управляющая плата сравнивает это значение с текущим положением вала (оно определяется с помощью датчика обратной связи);
- на основе сравнения привод выполняет такое действие, чтобы текущее положение вала стало как можно ближе к полученному на вход параметру;



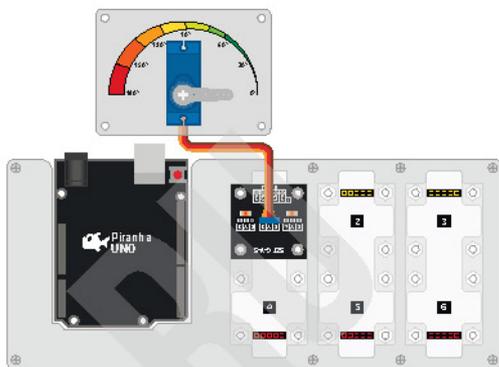
10.2: Установка и подключение

Сервопривод установлен в специальный корпус и имеет накладку со **шкалами**. Накладка съёмная, поэтому в будущем её можно будет менять или переворачивать.

Шаг 1

Подключи модули, как на рисунке:

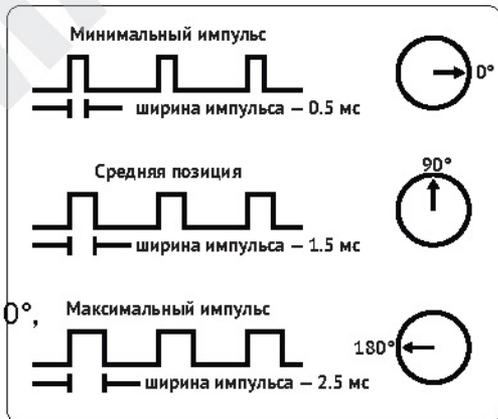
- «Расширитель» в ячейку **1**
- во **вторую колодку** «Расширителя» надо подключить сервопривод:
 - **коричневый провод** – G
 - **красный провод** – V
 - **оранжевый провод** – 2



10.3: Работа с приводом

Управление сервоприводом с РП происходит следующим образом:

- на сервопривод в течении 20 микросекунд подаётся сигнал, который большую часть времени имеет значение LOW;
- в начале каждого такого сигнала его значение становится HIGH на время от 0.5 до 2.5 микросекунд, где 0.5 мс – значит установить угол 0° , а 2.5 мс – установить угол 180° .
- промежуточное время между 0.5 и 2.5 микросекундами соответствует значениям угла от 0° до 180° .



Для того, чтобы не указывать ширину импульса и длительность всего сигнала каждый раз при работе с сервоприводом, в группе «Робот» есть специальный блок «установить серво pin __ как __ grader».

Создадим скрипт, после старта которого сервопривод совершит плавный поворот от 0° до 180°, а затем то же самое, но в обратном направлении. Скрипт подходит для **калибровки** сервопривода: просто проверь, чтобы угол поворота качалки (пластиковая стрелка) совпадал с углом накладки!

Шаг 1

Создай новый скрипт (не забудь создать переменную «**угол**») и запусти его:

```
когда щёлкнут по [флаг]
поставить угол в 0
всегда
  установить серво pin 3 как угол grader
  изменить угол на 1
  если угол = 180 то
    повторить 180 range
      изменить угол на -1
      установить серво pin 3 как угол grader
```



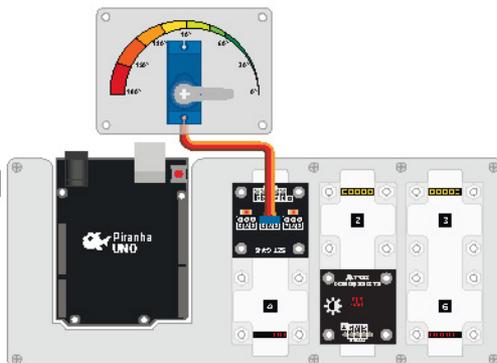
В данном случае я пошёл на маленькую хитрость, которая продиктована моей ленью :) В скрипте, для того, чтобы сервопривод поворачивал вал в обратном направлении, мы просто добавили в блок «повторить 180 раз» изменение угла на -1°, что позволяет вернуть качалку в начальное положение. Если же тебе хочется всё сделать «по-взрослому», то попробуй добавить блоки «передать ___» и «когда я получил ___», с помощью которых можно менять направление вращения!

Теперь создадим такой скрипт, в котором сервопривод будет менять угол поворота в зависимости от уровня освещённости: после запуска скрипта качалка сервопривода будет поворачиваться от 0° до 180°, и чем больше света падает на датчик, тем угол поворота больше и, наоборот, если ты очень плотно закроешь датчик рукой, то сервопривод вернётся в 0°.

Шаг 1

Добавь к уже установленному модулю ещё один:

- «Датчик освещённости» в ячейку 5



Шаг 2

Создай из блоков следующий скрипт (и не забудь про переменные «**угол**» и «**уровень освещённости**»):

Проект 3: Шлагбаум

Предлагаем тебе симитировать работу шлагбаума, который будет открываться по отпечатку пальца (задействуй своё воображение!). Да, сканер отпечатка пальца тоже придётся симитировать, зато радость от создания проекта точно будет настоящей!

Для проекта тебе понадобятся:

- Модуль «Датчик освещённости»;
- Модуль «Светодиод» **красного** цвета;
- Модуль «Светодиод» **зелёного** цвета;
- Модуль «Расширитель выводов»;
- Сервопривод.

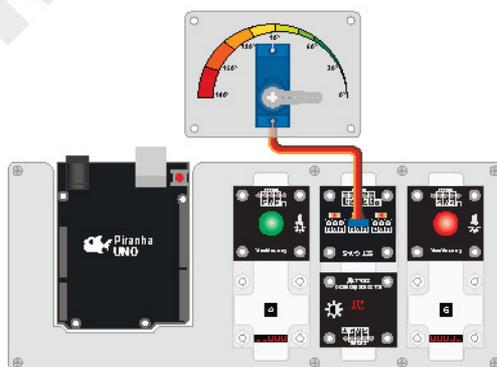
Можно будет организовать свою личную резиденцию с ограниченным доступом!



3.1: Подключение

Подключи модули в ячейки, как показано на рисунке справа:

- «Датчик освещённости» в ячейку **5**;
- **Зелёный** «Светодиод» в ячейку **1**;
- **Красный** «Светодиод» в ячейку **3**;
- «Расширитель» в ячейку **2**
- во **вторую колодку** «Расширителя» надо подключить Сервопривод



3.2: Создание скрипта

когда я получил | отпечаток получен
установить цифровой pin 8 udgang как HIGH
установить цифровой pin 10 udgang как LOW

когда я получил | ожидание следующего отпечатка
установить цифровой pin 8 udgang как LOW
установить цифровой pin 10 udgang как HIGH
поставить | шлагбаум открыт | в | 0

«А где же тут сканер отпечатка пальца?» – спросишь ты. Его работу как раз и «имитирует» модуль «Датчик освещённости».

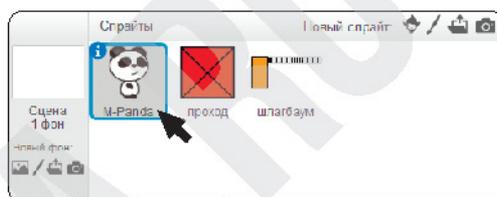
3.3: Создание анимации

Что ж, такой проект не может обойтись без красочной анимации!

Шаг 1

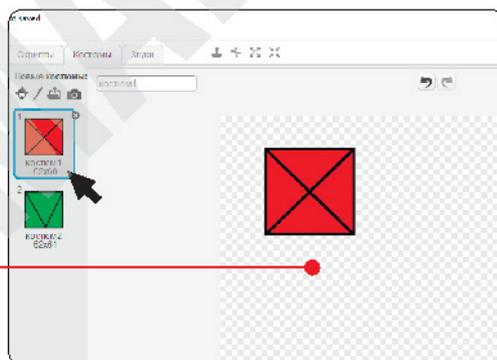
Создай следующие спрайты:

- «*Проход*»;
- «*Шлагбаум*».



Шаг 2

У спрайта «*Проход*» создай два костюма:
костюм 1 – проход запрещён;
костюм 2 – проход разрешён.

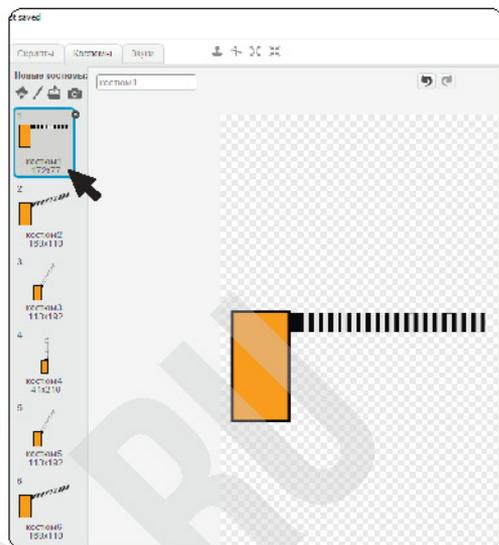


В данном случае никто не запрещает тебе проявить свою фантазию и изобразить любой рисунок, который будет отображать разрешение и запрет на проход!

Шаг 3

Для спрайта «Шлагбаум» создай шесть костюмов, как показано на рисунке:

- костюм 1 – Шлагбаум закрыт (0°);
- костюм 2 – Шлагбаум открыт на 30°;
- костюм 3 – Шлагбаум открыт на 60°;
- костюм 4 – Шлагбаум открыт на 90°;
- костюм 5 – Шлагбаум закрыт на 60°;
- костюм 6 – Шлагбаум закрыт на 30°.



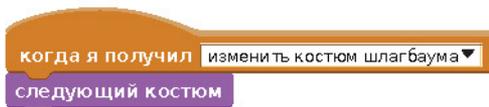
Возможно, тебе уже стало понятно, что костюмы 5 и 6 – это точные копии костюмов 3 и 2 соответственно. Поэтому достаточно скопировать рядом стоящие костюмы 2 и 3, после чего переместить и переименовать новые костюмы согласно их порядковому номеру. Вот и всё, все костюмы готовы!

Шаг 4

Скрипт для спрайта «Проход»



Скрипт для спрайта «Шлагбаум»





Каверзные вопросы



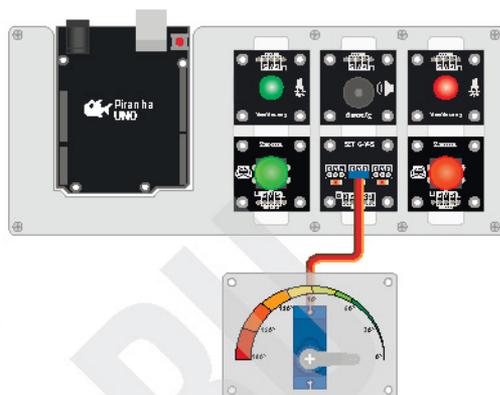
- Каким образом можно определить значение уровня освещённости, при котором скрипт будет работать?
- Можно ли уменьшить количество костюмов спрайта «*Шлагбаум*»? Если да, то как? Если нет, то почему?
- Измени скрипт спрайта «*M-Panda*» так, чтобы шлагбаум работал в инверсном режиме (то есть, чтобы он открывался самостоятельно при высоком уровне освещённости и закрывался при уменьшении уровня освещённости ниже заданного значения).

TREMA.RU

4.2: Подключение

Подключи модули в ячейки, как показано на рисунке справа:

- **Зелёный** «Светодиод» в ячейку **1**
- «Зуммер» в ячейку **2**
- **Красный** «Светодиод» в ячейку **3**
- **Зелёную** «Кнопку» в ячейку **4**
- «Расширитель» в ячейку **5**
- **Красную** «Кнопку» в ячейку **6**
- во вторую колодку «Расширителя»
надо подключить сервопривод:
 - **коричневый провод** – G
 - **красный провод** – V
 - **оранжевый провод** – 2



Скажем несколько слов о том, для чего будет использоваться каждый из модулей:

- **Зелёный** «Светодиод» будет кратковременно загораться каждый раз при касании мячиком платформы;
- **Красный** «Светодиод» будет загораться только при проигрыше;
- «Зуммер» будет кратковременно подавать звуковой сигнал при касании мяча платформы;
- **Зелёная** «Кнопка» будет отвечать за перемещение платформы влево;
- **Красная** «Кнопка» будет отвечать за перемещение платформы вправо;
- «Расширитель выводов» нужен для того, чтобы подключить к плате «Сервопривод», который будет показывать угол, на который отскочил мяч от платформы.

Шаг 1

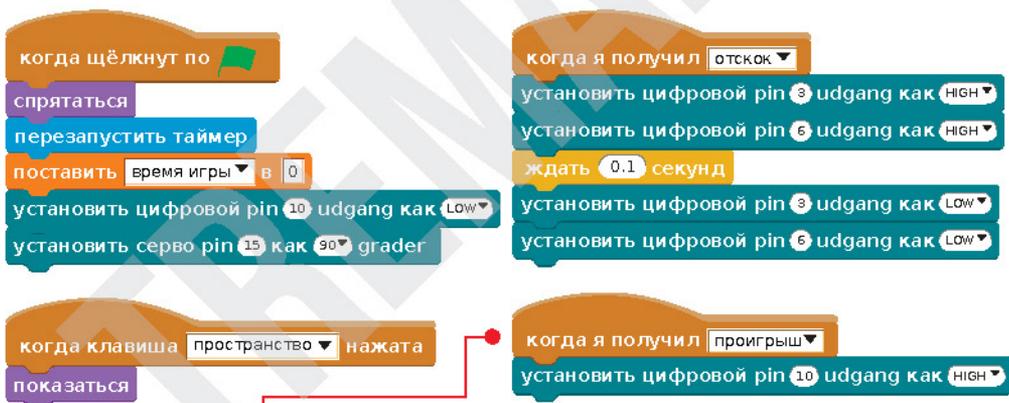
Для того, чтобы добавить отсчёт времени в скрипт, есть блок **таймер**, который находится в группе **Сенсоры**, а чтобы время можно было зафиксировать, понадобится переменная «**время игры**», куда будет сохраняться значение **таймера**.

Шаг 2

Настроим работу световой и звуковой индикации игры, то есть «Зуммера» и «Светодиодов»: при каждом отскоке мяча будет загораться **Зелёный** светодиод и кратковременно включаться «Зуммер». В случае проигрыша загорится **Красный** светодиод.

Ещё сделаем так, чтобы при запуске новой игры таймер и «Сервопривод» возвращались в начальное положение, а спрайт «*M-Panda*» был скрыт. При нажатии на клавишу <Пробел> спрайт снова вернётся в область анимации.

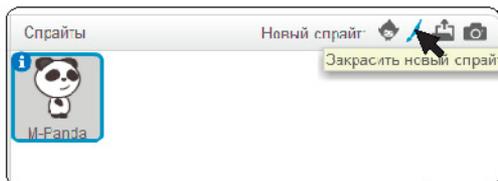
Скрипт для спрайта «*M-Panda*»:



В блоке **когда я получил** нажми на треугольник справа и выбери «**Новое сообщение**». В появившемся окне введи «**отскок**» и нажми «ОК». Повтори то же самое для команды «**проигрыш**».

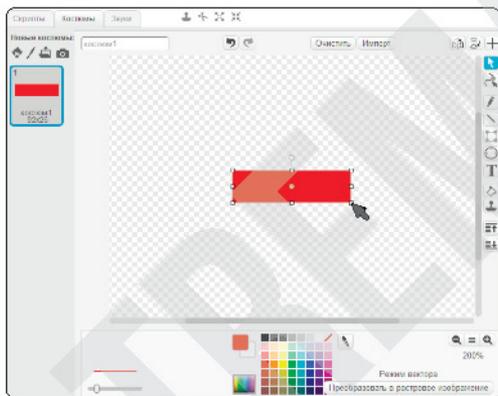
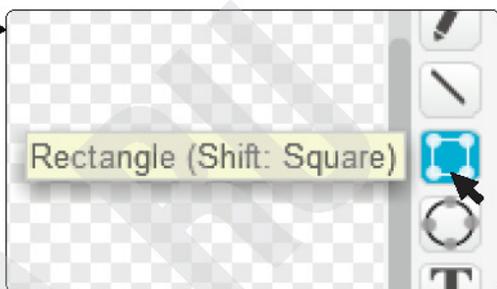
Шаг 4

Теперь создай спрайт «*платформа*», которым будешь отбивать мяч.



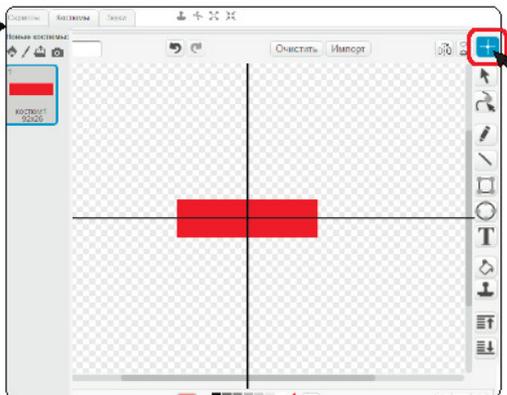
Нажми кнопку
«Закрасить новый спрайт»

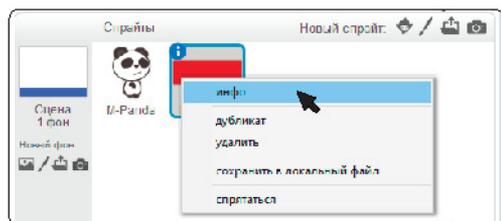
В панели инструментов справа
выбери инструмент
«**Rectangle**» (прямоугольник).



Нарисуй
прямоугольник
красного цвета.

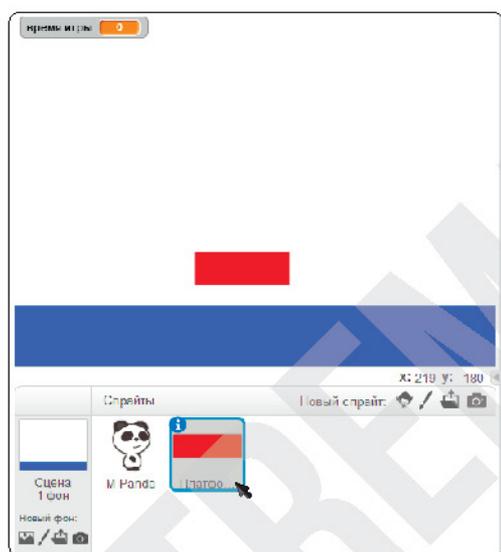
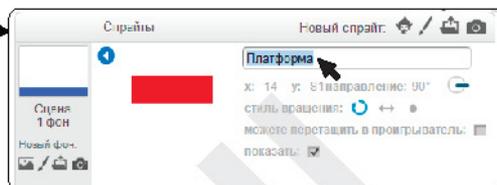
После этого выбери справа
«**Установите костюм**»
и проверь, что перекрестие
чёрных линий находится
в центре твоего
прямоугольника





Щёлкни правой кнопкой по
спрайту *«Sprite 1»*,
выбери строку *«инфо»*.

И измени имя
спрайта на
«Платформа»



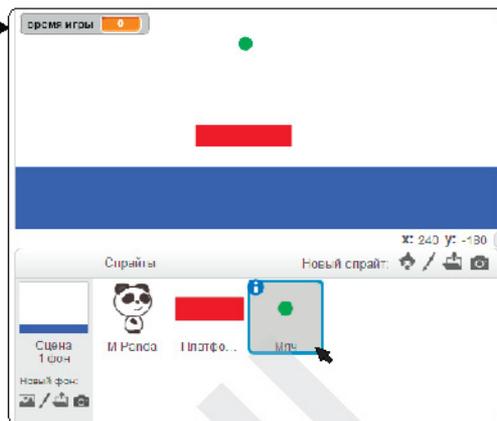
У тебя должен появиться
спрайт *«Платформа»* в
области анимации.

Шаг 5

Выбери спрайт *«Платформа»* и перейди во вкладку *«Скрипты»*. Сейчас ты создашь скрипт, который будет отвечать за движение *«Платформы»*:

- при нажатии на **Зелёную** кнопку *«Платформа»* будет сдвигаться влево;
- при нажатии на **Красную** кнопку *«Платформа»* будет сдвигаться вправо;
- каждый раз при старте *«Платформа»* должна возвращаться в начальное положение.

Не забудь сменить имя спрайта на «Мяч»

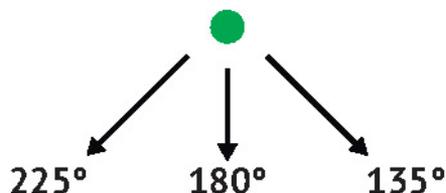


Шаг 7

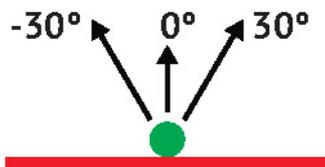
Пришло время разобраться с будущими настройками скрипта «Мяча»:

- при старте скрипта спрайт должен возвращаться в начальное положение и начинать движение по случайной траектории;
- когда мяч касается «Платформы», он отскакивает на определенный угол («*угол отскока*»), а сервопривод на этот же угол отклоняется от положения 90° ;
- если мяч коснулся «мёртвой зоны», то игра и таймер останавливаются;
- Блок **выдать случайное от** **до** способен выдавать числа из указанного диапазона в случайном порядке. В будущем скрипте он используется:

- в начале (**от 135° до 225°**).
Это будут границы углов, между которыми «Мяч» будет начинать движение на старте.



- в середине (**от -30° до 30°**).
Это будут границы углов отскока «Мяча» от «Платформы».

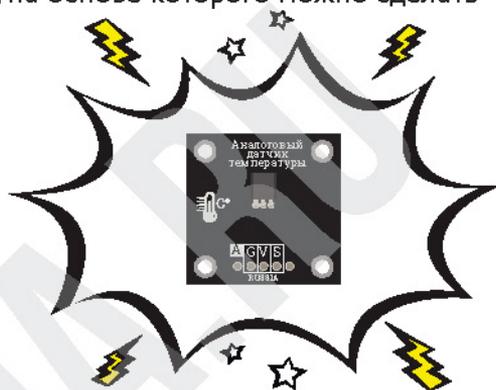


Урок 11: Модуль «Датчик температуры»

Хочешь определить температуру в помещении? На улице? Да где угодно! Сейчас ты познакомишься с модулем, на основе которого можно сделать свой собственный термометр!

Краткое описание:

Модуль «Датчик температуры» может точно измерять температуру в диапазоне от -50°C до $+125^{\circ}\text{C}$. Так как это аналоговый датчик, то и подключать его надо к аналоговому выводу.



11.1: Работа с модулем

Принцип работы модуля схож с «Датчиком освещённости»: при изменении температуры изменяется напряжение на выходе модуля.

«В таком случае у меня есть как минимум два вопроса!» – скажешь ты.

«Какие?» – спросим мы.

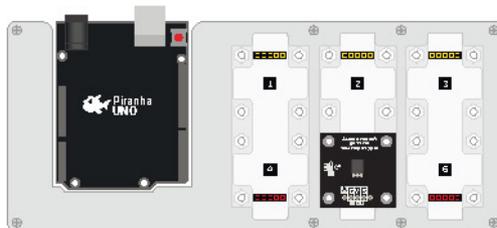
«А вот они:

- Как преобразовать входное аналоговое значение в градусы?
- Как отобразить на экране полученную температуру?»

Отличные вопросы. Давай разбираться сразу на практике!

Шаг 1

Для начала подключи модуль в ячейку **5**, как показано на рисунке.



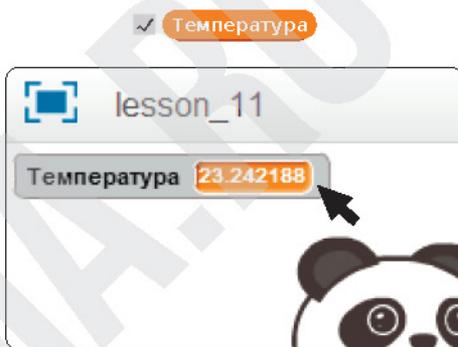
Шаг 2

В новом проекте создай переменную «*Температура*», где будут храниться значения с «Датчика температуры», и собери скрипт, используя общую формулу перевода напряжения в температуру:



Шаг 3

Ответ на второй вопрос тебе, скорее всего, уже известен: зайти в группу **Данные и Блоки** и поставить галочку возле переменной **Температура**. Теперь в области анимации будет отображаться **текущая температура** того места, где ты (или твой «Датчик температуры») сейчас находишься.



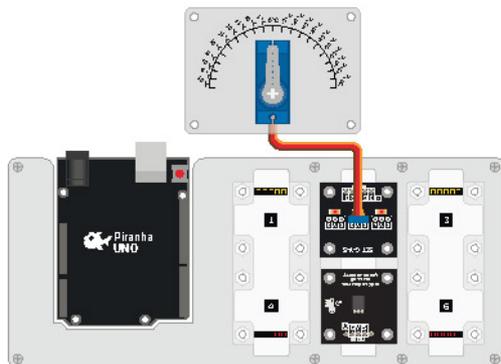
Помимо вывода переменной в область анимации, ты также можешь создать анимацию для отображения температуры в виде цифр или шкалы значений, либо воспользоваться внешней шкалой значений.

В этот раз мы воспользуемся готовой шкалой, подключив к рабочей плате сервопривод. Однако, прежде чем начинать, проверни накладку на корпусе сервопривода на ту сторону, где указана температура.

Шаг 1

Сперва надо добавить следующие модули:

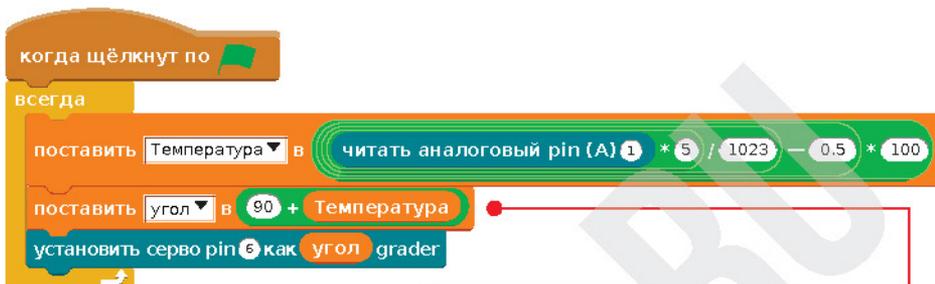
- «Расширитель» в ячейку **2**
- во **вторую колодку** «Расширителя» надо подключить сервопривод



Шаг 2

В скрипте мы сделаем так, чтобы качалка сервопривода при старте устанавливалась в положение, соответствующее 0°C на накладке, а уже из этого положения отклонялась на такой угол, который на шкале будет показывать полученное значение с «Датчика температуры».

Теперь можно переходить к скрипту.



Почему надо температуру прибавлять к углу? Это сделано для того, чтобы качалка сервопривода, в зависимости от температуры, отклонялась от своего центра **вправо или влево**. А так как **центром** температурной шкалы на накладке является угол в **90°**, то и прибавлять температуру надо к числу 90.



Вот теперь ты можешь наблюдать значение температуры не только на экране монитора, но и на температурной шкале наклейки, которая установлена на сервопривод!



Каверзные вопросы



- По какому принципу работает модуль «Датчик температуры»?
- Каким образом происходит перевод из аналогового значения с модуля в конечное значение температуры в градусы?
- Какие способы отображения температуры тебе известны?

Урок 12: Модуль «Потенциометр»

Если тебе приходилось крутить ручку громкости на аудиоколонках или стики на джойстиках игровых приставок или выставлять температуру на утюге, то тебе будут знакомы возможности данного модуля!

Краткое описание:

Модуль «Потенциометр» позволяет менять значение сигнала на выходе за счёт вращения рукоятки. Именно поэтому его часто используют для регулировки звука, яркости, скорости вращения и много другого. Так как у модуля аналоговый сигнал, то и подключать его надо к аналоговым выводам.

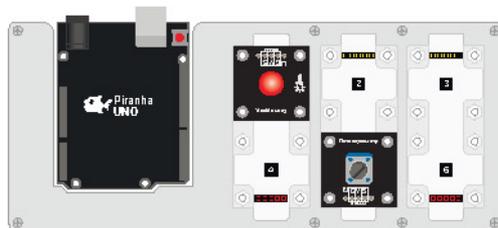


12.1: Работа с модулем

Шаг 1

Для начала подключи модули, как показано на рисунке:

- «Светодиод» в ячейку **1**;
- «Потенциометр» в ячейку **5**.



Шаг 2

Создадим скрипт, имитирующий работу **круглого диммера**: мы будем считывать значение положения ручки потенциометра и преобразовывать его в соответствующее ему значение ШИМ-сигнала. Таким образом, ты будешь управлять яркостью светодиода вращением ручки потенциометра.

К слову, в смартфонах, когда ты вручную настраиваешь яркость экрана, происходит тот же самый процесс, только без физического вращения ручки!

12.3: Создание скрипта и анимации

Шаг 1

Создай новый проект и составь следующий скетч:

Скрипт для объекта «*M-Panda*»

The image shows a Scratch script for an object named «M-Panda». The script consists of three event blocks, each followed by an action block:

- Event 1:** «когда щёлкнут по флажку» (when clicked on the flag) → «спрятаться» (hide) → «всегда» (always) → «поставить показания потенциометра в читать аналоговый pin (A)» (set potentiometer reading to read analog pin (A)).
- Event 2:** «когда клавиша стрелка вверх нажата» (when up arrow key is pressed) → «передать Движение по Y» (send Y movement).
- Event 3:** «когда клавиша стрелка вправо нажата» (when right arrow key is pressed) → «передать Движение по X» (send X movement).

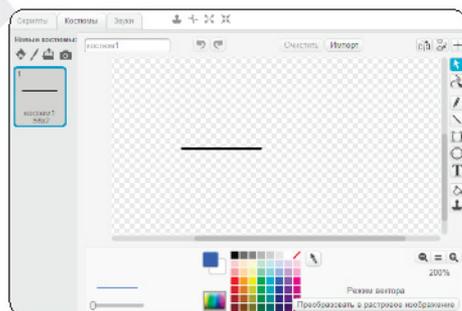
Below these are two more event blocks:

- «когда клавиша пространство нажата» (when space key is pressed) → «показаться» (show) → «остановить все» (stop all).

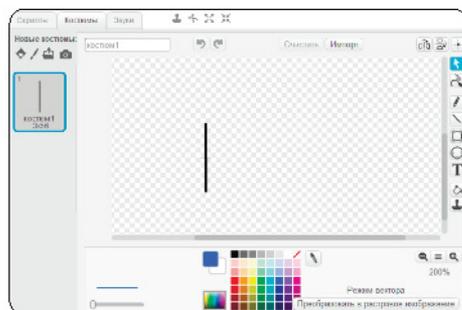
Шаг 2

Создай два новых спрайта:

– «*Горизонтальная линия*» – спрайт, который представляет из себя небольшую горизонтальную линию, расположенную в центре области анимации.



– «*Вертикальная линия*» – спрайт, который представляет из себя небольшую вертикальную линию, расположенную в центре области анимации.



Шаг 3

Создай для каждого спрайта скрипты, в которых положение спрайта будет зависеть от положения ручки потенциометра. И помни о том, что размеры оси X и оси Y разные!

Стоит также перед созданием скрипта объяснить некоторый нюанс, связанный с вычислением **отрицательных** координат в области анимации.

Допустим, тебе надо, повернув ручку потенциометра в крайнее правое положение, установить $Y = 180$. Если считать показания с потенциометра напрямую, то ты получишь число 1023. Значит, как мы делали ранее, можно добавить преобразующий коэффициент: $1023/180 = 5.68$ в скрипт при считывании значений с потенциометра. Вроде всё? Нет.

Что же теперь делать, если надо получить координату $Y = -180$? Если ты повернёшь ручку потенциометра в крайнее левое положение, то Y станет равен 0. Логично, что надо просто сделать: $0 - 180$ и всё. Но не всё так просто, ведь крайнее правое положение в этом случае будет равно: $1023/5.68 - 180 = 0$! Вот тут-то мы и воспользуемся одним нюансом: установим преобразующий коэффициент в 2 раза ниже, чтобы при вычитании из максимального значения получалось 180, а при вычитании из минимального значения получалось -180. Проверим:

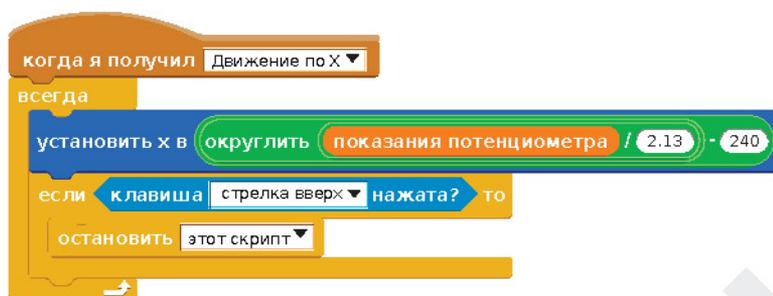
- крайнее левое положение ручки потенциометра: $0/2.84 - 180 = -180$
- крайнее правое положение ручки потенциометра: $1023/2.84 - 180 = 180$

Получилось! Вот теперь можно переходить к созданию скрипта!

Скрипт для спрайта «Горизонтальная линия»

Этот блок ты найдёшь в группе **Движение**

Если не можешь найти нужный тебе блок **клавиша стрелка вправо нажата?**, ищи его в группе блоков **Сенсоры**



После запуска скрипта выбери нужную тебе линию клавишами <Вверх> или <Вправо> и регулируй её положение с помощью вращения ручки «Потенциометра». Нажатие клавиши <Пробел> остановит весь скрипт.



Каверзные вопросы



- Для чего нужен модуль «Потенциометр»?
- Создай свой скрипт, в котором линии будут перемещаться по диагонали, то есть из одного угла в противоположный. Соответственно, спрайты тоже будут новые!

Проект 5: Игра «Ловим яблоки»

Модуль «Потенциометр» – очень полезная вещь, причём не только в серьёзных проектах, но и в проектах «для души». Именно таким проектом и будет игра, в которой главный герой бегает по экрану и ловит яблоки для того, чтобы сварить вкуснейший виртуальный компот.

Для проекта тебе понадобятся:

- Модуль «Потенциометр».

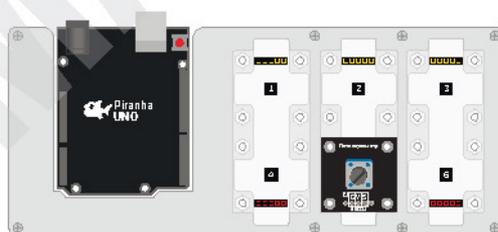
Ох, столько спелых яблок я видел только у дедушки в деревне. Какие же они были вкусные, аж слюнки потекли!



5.1: Подключение

Установи следующий модуль:

- «Потенциометр» в ячейку 5



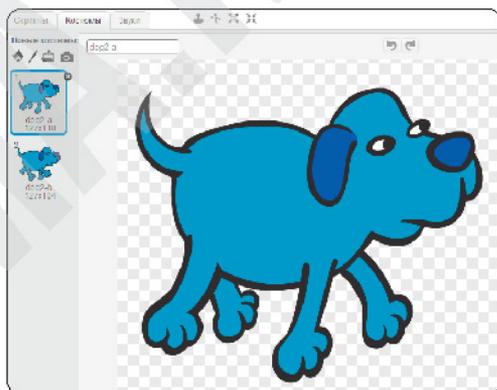
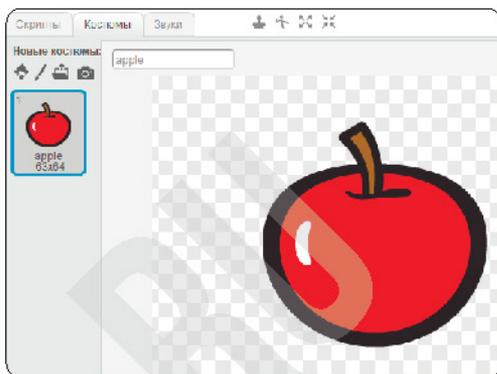
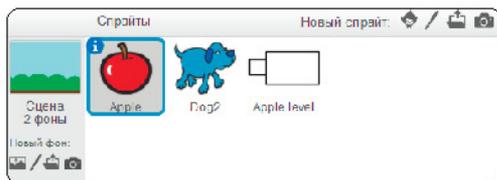
5.2: Создание анимации

Давай разобьём всю игру на части, каждая из которых в дальнейшем будет описана в скрипте:

- персонаж, которым ты управляешь, движется только по горизонтали и собирает яблоки, количество которых будет влиять на результат игры;
- яблоки движутся сверху вниз, с разной скоростью, а при падении на землю исчезают с экрана. Чтобы яблок на экране появлялось сразу много, мы будем использовать несколько новых блоков, но об этом чуть позже;
- есть индикация того, на какой стадии игры ты находишься.

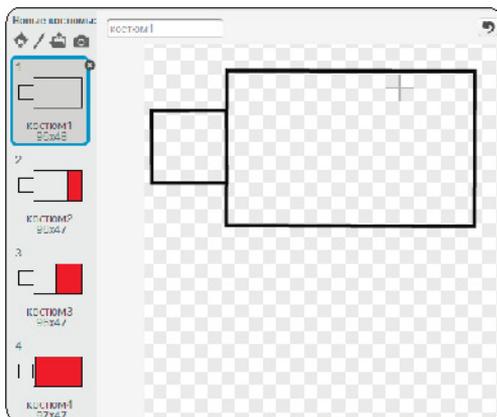
Шаг 1

Для начала необходимо добавить в проект 3 новых спрайта: «*Apple*», «*Dog2*» и «*Apple level*». Для первых двух в библиотеке спрайтов уже есть готовые костюмы, а вот для «*Apple level*» надо будет создать костюмы самостоятельно.



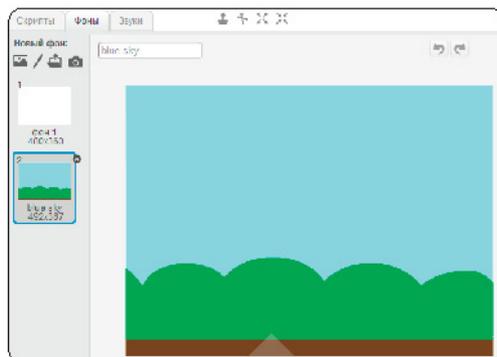
Шаг 2

Для создания спрайта «*Apple level*» нарисуй два прямоугольника, как показано на рисунке справа. Затем в большой прямоугольник в каждом костюме надо добавить прямоугольник красного цвета, который будет отображать прогресс прохождения игры.



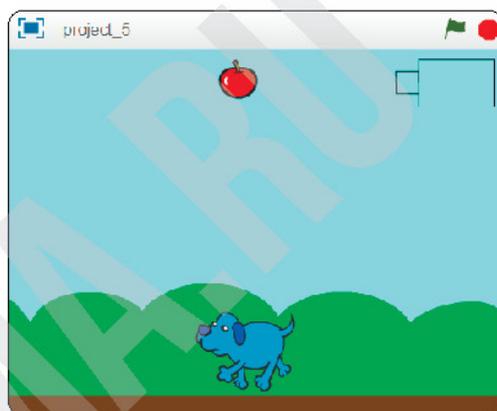
Шаг 3

Последнее, что стоит сделать, — это изменить фон «Сцены». Для это в библиотеке фонов выбери любой фон, который тебе больше всего нравится, и сделай его основным.



Шаг 4

В результате у тебя должно получиться нечто похожее на рисунок справа.

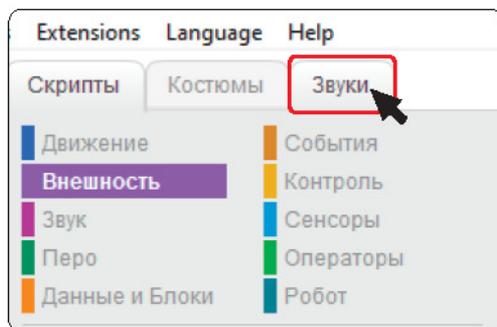


5.3: Немного о звуках

В программе mBlock можно добавлять в скрипты мелодии и звуки. Для этого есть группа блоков **Звук**, и сейчас ты научишься с ней работать!

Шаг 1

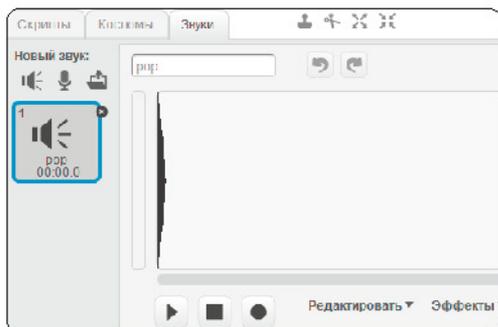
Чтобы добавить звуки в скрипт, сперва надо выбрать нужный спрайт и зайти в его вкладку «Звуки» в верхней части программы. В каждый спрайт звуки добавляются независимо друг от друга! В этом проекте надо будет добавить звуки в спрайты «Apple» и «Dog2».



Шаг 2

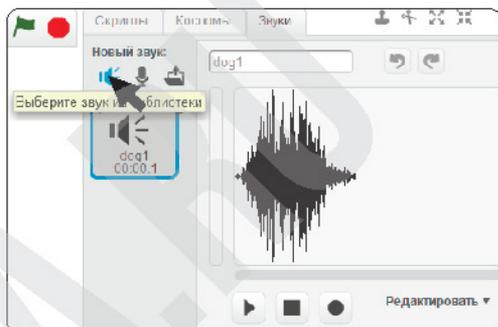
Сперва добавим звук в спрайт «*Apple*». Для этого зайти во вкладку «Звуки» этого спрайта и...

О, гляди, — нужный звук («*pop*») уже добавлен в спрайт по умолчанию! В таком случае переходим к спрайту «*Dog2*».



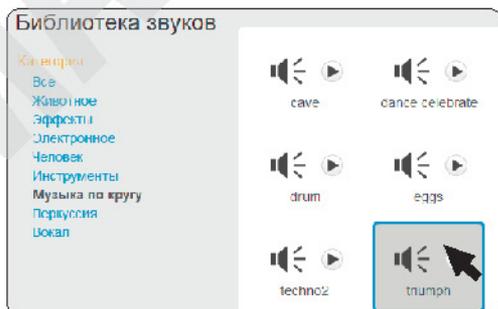
Шаг 3

Выбери спрайт «*Dog2*», зайти во вкладку «Звуки» и нажми кнопку «Выберите звук из библиотеки».



Шаг 4

Для этого проекта необходимо два звука: «*triumph*» и «*save*». Добавь их по очереди в список звуков твоего спрайта.

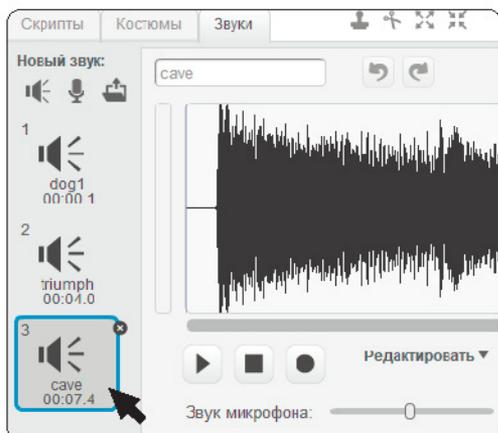


Шаг 5

Во вкладке «Звуки» теперь должно быть три звуковых дорожки.

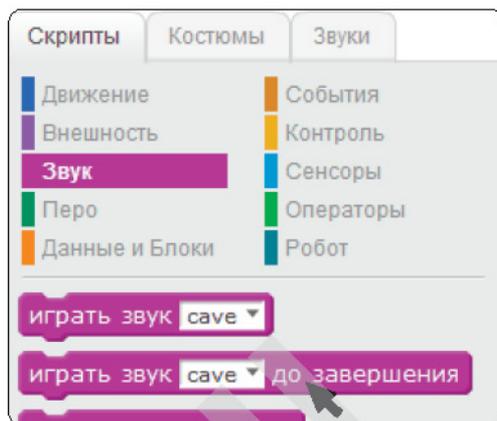
Их можно ещё раз прослушать.

Запомни названия нужных звуков.



Шаг 6

Теперь в скрипте можно указать звуки для проигрывания! Для этого надо зайти в группу блоков **Звук** и выбрать блок **играть звук cave** до завершения, внутри которого можно выбрать нужный тебе звук из добавленных ранее.



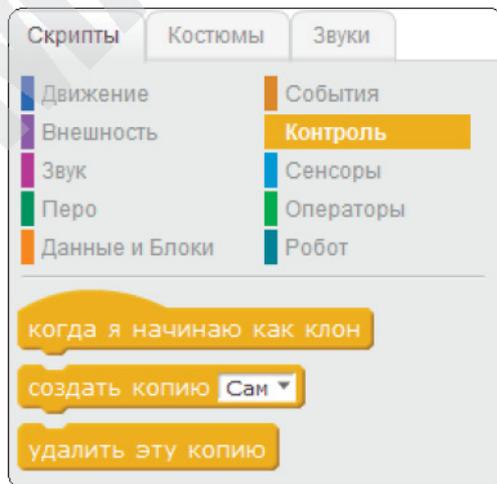
5.4: Создание клонов

Пожалуй, последнее, что тебе стоит узнать перед тем, как перейти к созданию скрипта, – это полезную во многих случаях возможность создавать **клоны** (или **копии**) спрайтов.

Наверняка, тебе уже попадались на глаза блоки с этим словом, которые ты найдёшь в группе «**Контроль**». С помощью блока «**создать копию**» на экране можно создать **до 300(!)** копий одного спрайта!

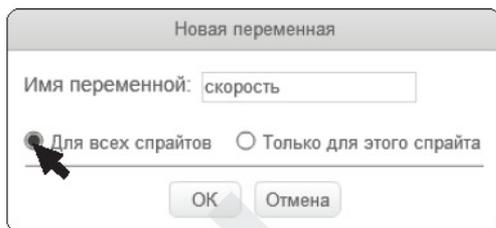
Важно знать, что созданные клоны будут пропадать каждый раз при остановке работы твоего скрипта, а в случае его запуска будут полностью повторять все действия оригинала. Если же нужно, чтобы клон выполнял отличное от оригинала действие, тебе понадобится блок «**когда я начинаю как клон**», в котором и указываются команды для клона.

Прежде чем переходить к следующей странице, создай новые переменные: «**X**», «**предыдущее значение X**», «**очки**».

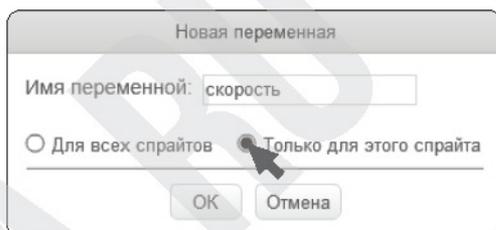


В рамках данного проекта стоит познакомиться с ещё одной особенностью переменных в среде mBlock, которая активно используется в работе клонов, — **локальные и глобальные переменные**.

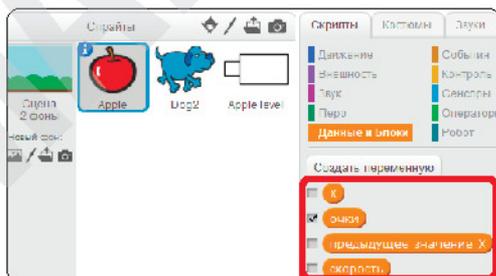
Глобальная переменная — та, к которой есть доступ (**чтение и запись**) у всех спрайтов (и у сцены) в проекте. По умолчанию все переменные, которые создаются в mBlock, являются глобальными.



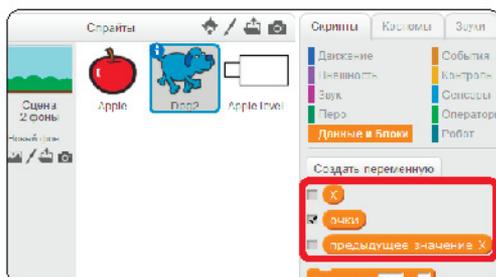
Локальная переменная — та, к которой доступ есть только у спрайта, в котором она создана («**Сцена**» не может иметь таких переменных). Этот тип переменных создаётся так же, как и предыдущий, но выбрать надо «**Только для этого спрайта**».



Ты спросишь: «А зачем мне это знать?» Затем, что сейчас ты создашь скрипт, при работе которого на поле будет выводиться сразу несколько одинаковых спрайтов «**Apple**», которые должны двигаться с разной скоростью.



В этом случае тебе и пригодится **локальная переменная**: когда ты создаёшь копию «**Apple**», то каждый клон создаёт себе свою собственную копию переменной «**скорость**», и в каждую из этих копий переменной записывается разное значение.



Если же выбрать **глобальную переменную**, то для каждого клона «**Apple**» надо будет создать свою переменную для значения скорости. Это легко сделать, когда яблок 3 или 10, а если яблок сотня, тогда это будет не так-то просто!

Ничего страшного, если сейчас тебе до конца не понятна разница между переменными. Стоит перейти к созданию самого скрипта и на практике разобраться, как это работает. Ну что ж, приступим к созданию скрипта!

5.5: Создание скрипта

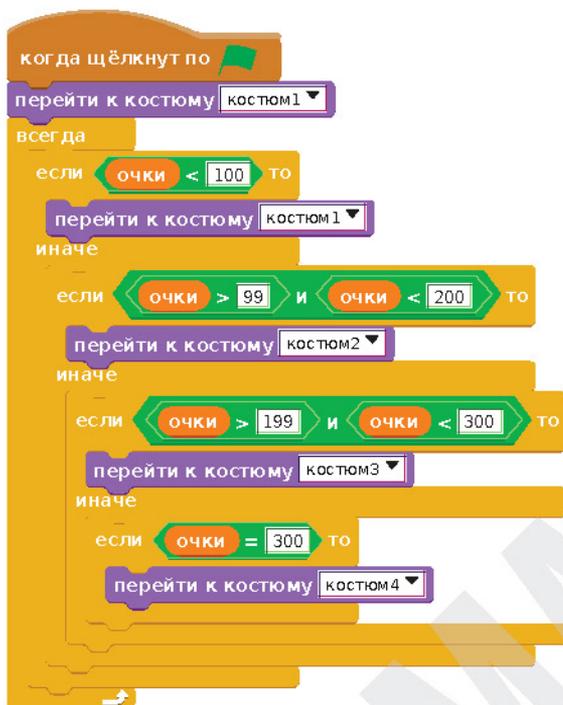
Для начала создадим скрипт для спрайта «Dog2»:

```
когда щёлкнут по флагу
сказать
установить направление вращения влево-вправо
поставить очки в 0
идти в x: 0 y: -113
всегда
поставить X в округлить читать аналоговый pin (A) 1 / 2.13 * 240
идти в x: X y: -113
если X > предыдущее значение X - 2 то
    повернуть в направление 90
    следующий костюм
если X < предыдущее значение X + 2 то
    повернуть в направление -90
    следующий костюм
если очки = -100 то
    сказать О, нет! Самые вкусные яблоки разбились!
    играть звук cave до завершения
    остановить все
иначе
    если очки = 300 то
        сказать Победа! Теперь яблок хватит надолго!
        играть звук triumph до завершения
        остановить все
поставить предыдущее значение X в X
```



Ты можешь самостоятельно задать те условия, при наступлении которых будет происходить выигрыш или проигрыш. Для этого достаточно просто изменить числа в этих блоках!

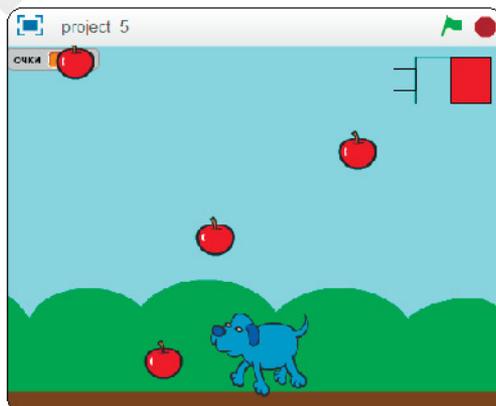
Теперь создадим скрипт для спрайта «Apple level»:



Ну вот и всё, можно переходить к игре! Давай посмотрим, что у тебя должно получиться в итоге:

Шаг 1

После запуска игры ты, управляя своим персонажем, ловишь падающие сверху яблоки. При этом в правом верхнем углу отображается твой прогресс.



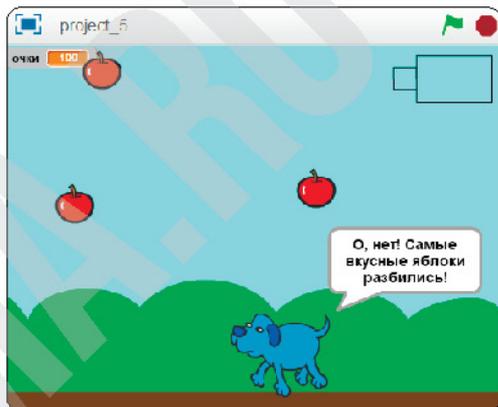
Шаг 2

В том случае, если количество пойманных яблок достигло верхней границы, то игра считается успешно пройденной, включается победная мелодия, и на экран выводится сообщение об этом.



Шаг 3

Если же было пропущено слишком много яблок, и количество очков достигло нижней границы значений, то игра считается проигранной, после чего будет включена соответствующая мелодия, и выведено текстовое сообщение на экран.



Каверзные вопросы



– Измени игру так, чтобы к простым яблокам добавились ещё и «плохие» яблоки, которые будут изменять скорость движения персонажа или после ловли определённого их количества игра также будет заканчиваться.

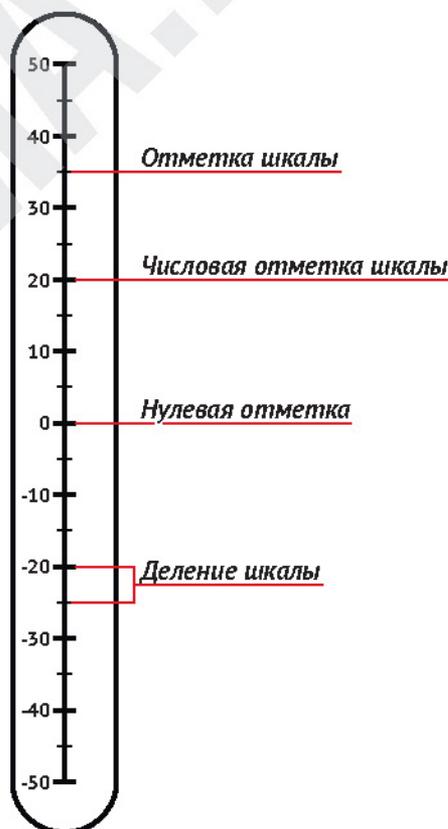
Урок 13: Измерительные шкалы приборов

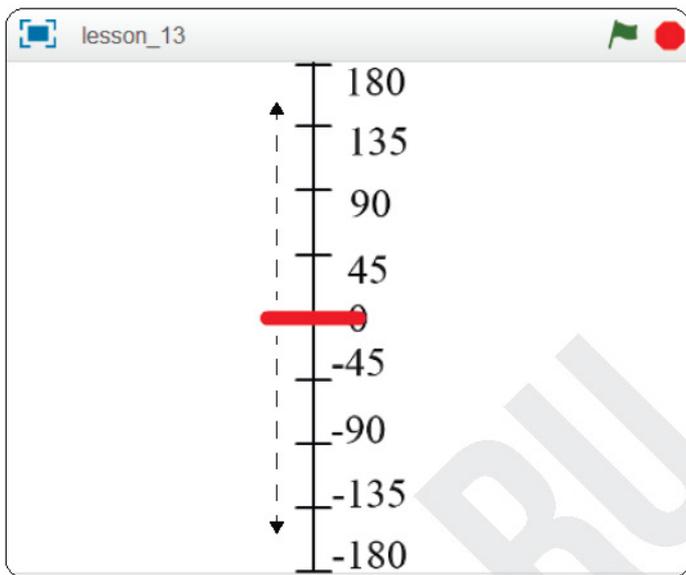
Ещё одной возможностью, которую даёт использование датчиков вместе с анимацией, является создание разных **систем отображения**, где реальные значения с датчиков соотносятся с их графическим отображением в области анимации. Такие системы называются **измерительными шкалами приборов**.

13.1: Структура

Шкалы могут иметь несколько вариантов исполнения: круглые, дугообразные, прямые, но все они имеют общие элементы:

- **Отметка шкалы:**
Знак на шкале, соответствующий определенному значению физической величины.
- **Числовая отметка шкалы:**
Отметка шкалы, с числовыми значениями.
- **Нулевая отметка:**
Отметка шкалы, соответствующая нулевому значению измеряемой величины.
- **Деление шкалы:**
Промежуток между двумя соседними отметками шкалы.
- **Цена деления шкалы:**
Разность значений двух соседних отметок шкалы.





Каверзные вопросы



- Какие элементы имеет шкала?
- Подумай, в каких приборах используются шкалы?
- Какой внешний вид может иметь шкала значений?
- Попробуй создать свою собственную шкалу, отличную от уже созданной тобой в этом уроке.

Проект 6: Климат-контроль

Многие автомобили, офисные здания, а так же «умные» дома всё чаще оснащают системой **климат-контроля** – сложной системой, способной в автоматическом режиме определять и менять температуру, влажность и много чего ещё, чтобы значения находились в установленных пределах. В этом проекте ты создашь прототип такой системы, которая будет определять температуру, сверять её с заданной границей, а при превышении – её выдавать предупреждающий звуковой сигнал.

Для проекта тебе понадобятся:

- Модуль «Потенциометр»;
- Модуль «Светодиод» **красного** цвета;
- Модуль «Светодиод» **зелёного** цвета;
- Модуль «Датчик температуры»;
- Модуль «Зуммер».

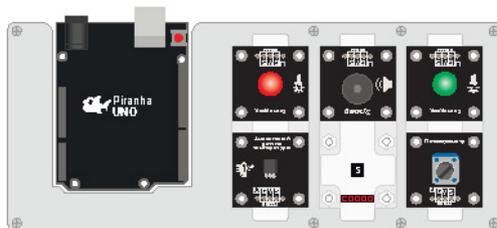
Создание комфортного микроклимата в помещении – важная и нужная задача!



6.1: Подключение

Установи следующие модули:

- **Красный** «Светодиод» в ячейку **1**
- «Зуммер» в ячейку **2**
- **Зелёный** «Светодиод» в ячейку **3**
- «Датчик температуры» в ячейку **4**
- «Потенциометр» в ячейку **6**



6.2: Создание скрипта

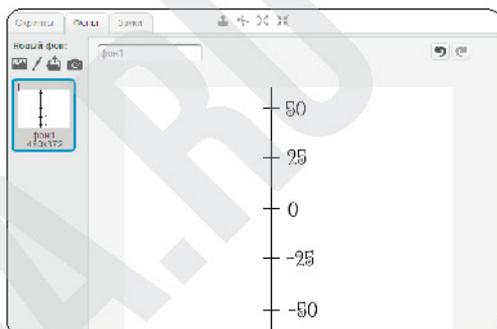
Перед созданием скрипта опишем его **принцип работы**:

- в области анимации есть шкала и два спрайта, отражающие **реальное** и **максимально допустимое** значение температуры;
 - вращением ручки потенциометра можно настроить максимально допустимый уровень температуры, двигая спрайт **«Предел»** вдоль шкалы;
 - если температура в пределах нормы, то горит **зелёный** «Светодиод», но если её значение превышает максимально допустимое, то система подаёт звуковой сигнал «Зуммером» и включает **красный** «Светодиод».
- Вот теперь можно переходить к созданию анимации, а затем и скрипта!

Шаг 1

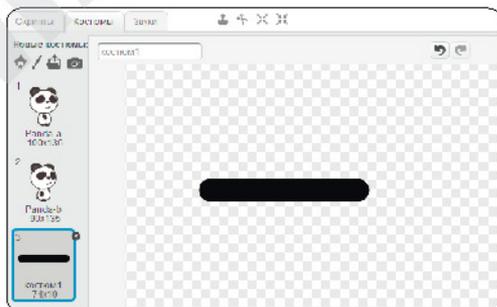
Взяв за основу анимацию из урока 13 (стр. 141), создай шкалу в «Сцене» следующим образом:

- раздели область анимации напополам и в центре установи 0;
- сверху и снизу шкалы поставь отметки 50 и -50 соответственно;
- расстояние от 50 до 0 тоже подели на пополам и поставь отметку 25;
- так же сделай и с нижней половиной, но отметка будет -25;



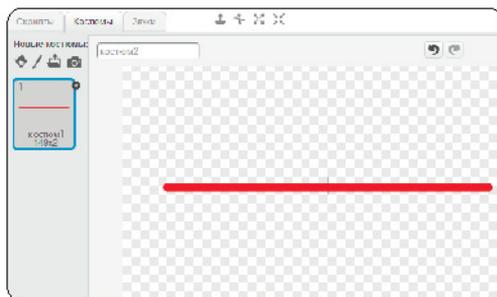
Шаг 2

Для спрайта «M-Panda» создай костюм в виде горизонтальной линии **чёрного** цвета и, расположив её по центру шкалы спрайта «Сцена», запомни координаты по оси X и по оси Y.



Шаг 3

Создай спрайт «Предел» в виде горизонтальной линии **красного** цвета и, расположив её по центру шкалы спрайта «Сцена», запомни координаты по оси X и по оси Y.



Скрипт для спрайта «M-Panda»

когда щёлкнут по 

идти в x: -26 y: 0

всегда

поставить Температура (датчик) в округлить $\text{читать аналоговый pin (A) } 0 * 5 / 1023 - 0.5 * 100$

поставить Предел (потенциометр) в округлить $\text{читать аналоговый pin (A) } 2 / 10.2 - 50$

поставить Температура (шкала) в Температура (датчик) * 3

поставить Предел (шкала) в Предел (потенциометр) * 3

установить y в Температура (шкала)

если Температура (датчик) > Предел (потенциометр) то

установить цифровой pin 3 udgang как HIGH

установить цифровой pin 6 udgang как HIGH

установить цифровой pin 10 udgang как LOW

иначе

установить цифровой pin 3 udgang как LOW

установить цифровой pin 6 udgang как LOW

установить цифровой pin 10 udgang как HIGH

Если будешь делать шкалу со своими пределами, то тебе придётся опытным путём подобрать свои коэффициенты для **второго** преобразования.



Скрипт для спрайта «Предел»

когда щёлкнут по 

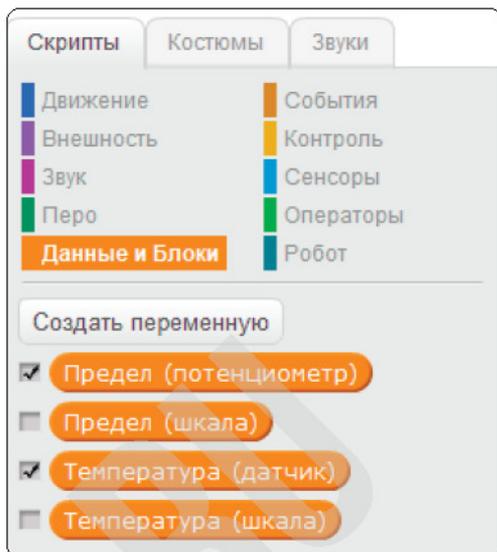
идти в x: -26 y: 0

всегда

установить y в Предел (шкала)

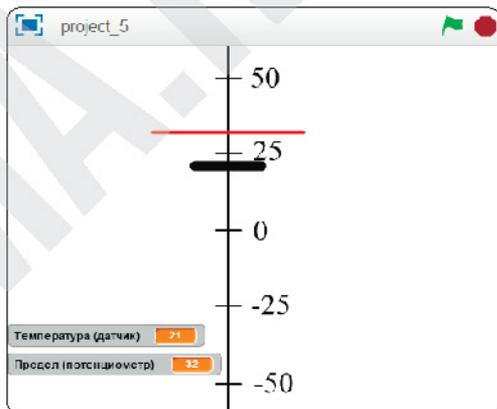
Шаг 6

В группе блоков **Данные и Блоки** поставь галочки напротив переменных **Температура (датчик)** и **Предел (потенциометр)**, чтобы рядом со шкалой в области анимации отображались значения реальной и максимально допустимой температур.



Шаг 7

Если всё сделано верно, то в области анимации ты получишь работающий прототип системы климат-контроля.



Каверзные вопросы



– Пользуясь полученными ранее знаниями (в этом уроке, а так же в уроках **12** и **13**), измени работу скрипта так, чтобы появилась возможность задавать не только верхний, но и нижний предел температуры.

– Используя «RGB-светодиод», создай новый прототип системы, чтобы, в зависимости от значения температуры, цвет «RGB-светодиода» менялся от **синего** («холодно») к **красному** («жарко»).

Проект 7: Аплодисментометр

Часто победителя соревнований выбирают с помощью аплодисментов, но всегда это вызывает споры: каждый считает, что громче всех хлопали именно его фавориту. Дабы унять споры, сегодня ты создашь устройство, которое наглядно и точно покажет, чьи аплодисменты громче всех!

Для проекта тебе понадобятся:

- Модуль «Светодиод» **красного** цвета;
- Модуль «Зуммер»;
- Модуль «Датчик звука»;
- Модуль «Расширитель выводов»;
- Сервопривод.

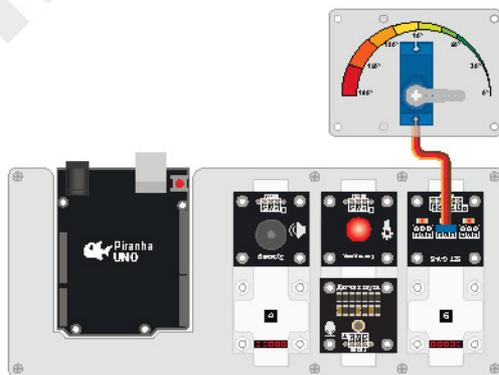
Хлопать надо так, как никогда раньше! В конечном счёте, ты же выберешь чемпиона!



7.1: Подключение

Подключи модули в ячейки, как показано на рисунке справа:

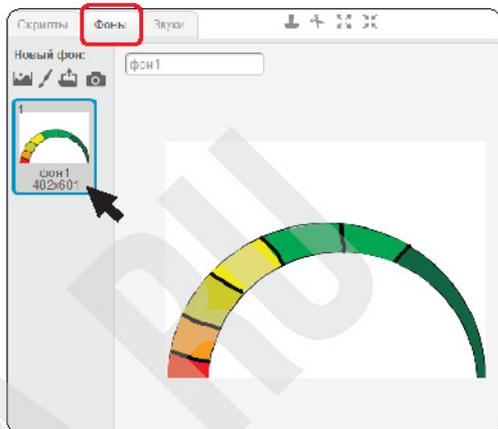
- «Зуммер» в ячейку **1**
- **Красный** «Светодиод» в ячейку **2**
- «Расширитель» в ячейку **3**
- Сервопривод во **вторую колодку** «Расширителя» (и установи для проекта правильную наклейку!)
- «Датчик звука» в ячейку **5**



7.2: Создание анимации

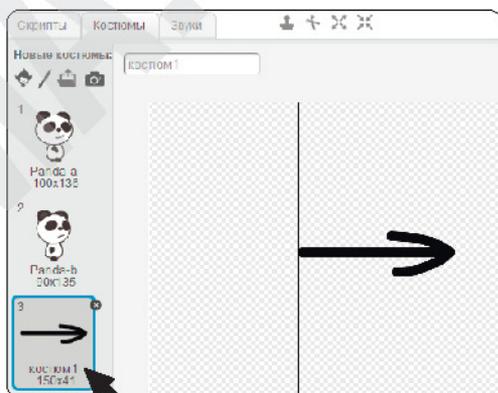
Шаг 1

Для будущего аплодисментметра в первую очередь надо создать точную шкалу. Для этого в списке спрайтов зайти в «Сцену» и во вкладке «Фоны» нарисуй рисунок, подобный тому, что изображён справа.



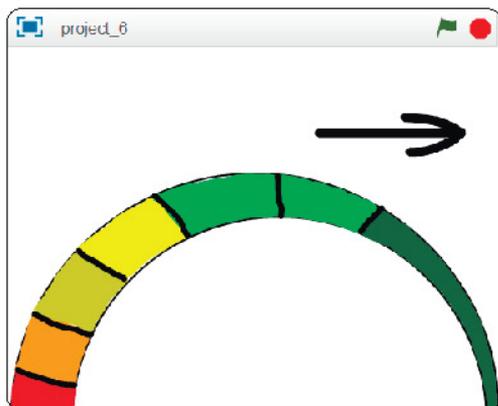
Шаг 2

Во вкладке «Костюмы» спрайта «M-Panda» нарисуй стрелку и установи центр костюма в её крайнем левом положении. Стрелка должна смотреть вправо.



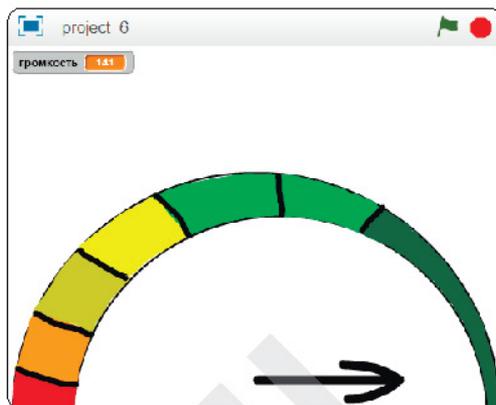
Шаг 3

В области анимации у тебя должно всё выглядеть примерно так же, как на рисунке справа. Не переживай, стрелку мы чуть позже установим на законное место, а сейчас — кое-что совершенно иное!



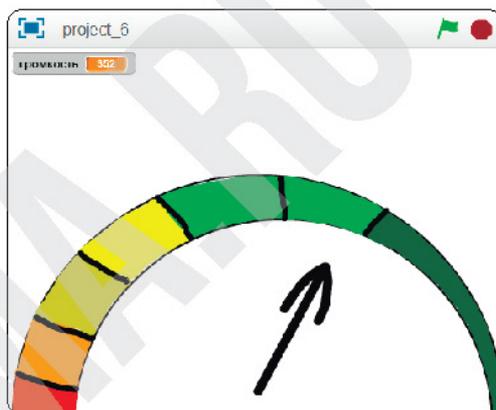
Шаг 3

После запуска скрипта спрайт «*M-Panda*», который имеет костюм в виде стрелки, будет отклоняться от начала отсчёта синхронно с качалкой сервопривода.



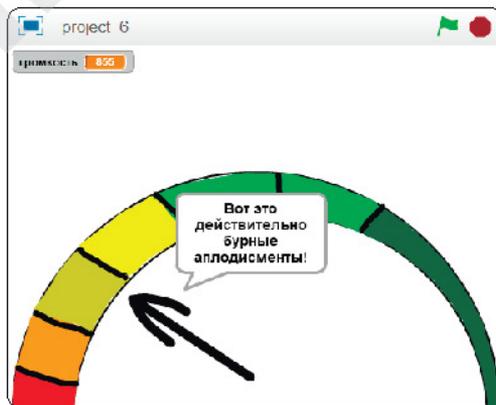
Шаг 4

До тех пор, пока громкость аплодисментов будет ниже границы, стрелка будет просто отклоняться от своего начального положения и затем возвращаться в него.



Шаг 5

Как только пороговое значение будет превышено, в области анимации будет воспроизводиться мелодия и выводиться текстовое сообщение.





Каверзные вопросы

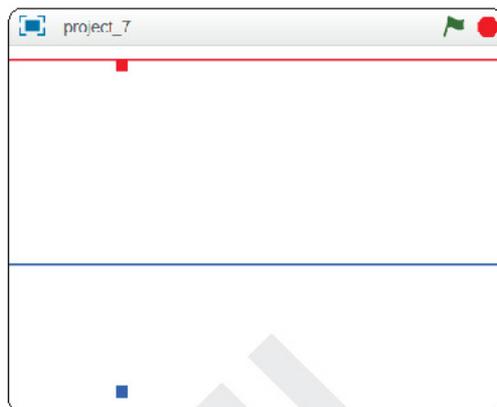


- Как ты думаешь, почему при переводе значений из переменной в значение угла поворота костюма у коэффициента перевода отрицательное значение? Что будет, если оно станет положительным? **Громкость**
- Какие блоки надо добавить, чтобы на экран выводилось только максимальное значение переменной **Громкость**?

TREMA.RU

Шаг 2

После того, как все спрайты и скрипты созданы, область анимации должна принять примерно тот же вид, что и наш рисунок справа (не забудь скрыть спрайт «*M-Panda*»!).



8.3: Создание скриптов

Прежде, чем создавать скрипты, опишем принципы их работы:

– принцип работы системы следующий: слева направо будут непрерывно двигаться спрайты «*Свет*» и «*Звук*», каждый из которых будет оставлять за собой нарисованный график изменения значений каждого из датчиков. Спрайты «*Предел освещённости*» и «*Предел громкости*» будут отображать максимально допустимые значения, которые можно будет менять с помощью вращения ручки «*Потенциометра*». При пересечении одного из первых двух спрайтов с любым из последних двух будут включаться «*Светодиод*» и «*Зуммер*», а при возврате значений в норму они будут автоматически выключаться.

– скрипт в спрайте «*M-Panda*» будет отвечать за логику работы системы. Для этого в нём будут считываться значения с «*Датчика звука*», «*Датчика освещённости*» и с «*Потенциометра*», а для хранения этих значений будут созданы переменные «*Уровень шума*», «*Освещённость*» и «*Потенциометр*» соответственно. В этом же скрипте будет создано управление каждым из предельных спрайтов, для переключения между которыми будут использоваться клавиши клавиатуры.

Шаг 1

Пришло время создать скрипты для каждого спрайта и заставить нашу систему работать! Также обрати внимание, что в скрипте используется большое количество переменных, которые тебе надо предварительно создать!

Скрипт для спрайта «Свет»

```
когда щёлкнут по флагу
изменить размер пера на 0.01
очистить
поднять перо
установить цвет пера
поставить Свет достиг края в 0
поставить Свет (X-координата) в -240
идти в х: Свет (X-координата) у: Освещённость
опустить перо
всегда
если Свет (X-координата) > 240 то
  если Свет достиг края = 0 то
    поставить Свет достиг края в 1
    поднять перо
    поставить Свет (X-координата) в -240
  иначе
    изменить Свет (X-координата) на 1
    идти в х: Свет (X-координата) у: Освещённость
    ждать 0.05 секунд
```

```
когда я получил опустить перо
очистить
опустить перо
```

При достижении пером правого края области анимации переменная **«Свет достиг края»** устанавливается в 1 и ждёт получения команды **«опустить перо»**. Ранее, в скрипте спрайта **«M-Panda»**, была введена проверка, что правого края достигли оба спрайта: **«Свет»** и **«Звук»**. Именно поэтому в скрипте спрайта **«Звук»** будет переменная

«Звук достиг края»,

которая при достижении спрайтом правого края будет устанавливаться в 1, после чего область анимации будет очищена, а обе переменные будут установлены в 0.



Скрипт для спрайта «Предел освещённости»

```
когда я получил Регулировка уровня света
идти в х: 0 у: Предел освещённости
всегда
поставить Предел освещённости в округлить Потенциометр / 5.68
установить у в Предел освещённости
если клавиша стрелка вниз нажата? то
  остановить этот скрипт
иначе
  если клавиша пространство нажата? то
    остановить этот скрипт
```

TREMA.RU

Как уже было сказано, все расчёты выполняет МК. И обычно это включает в себя получение времени, которое прошло между отправкой сигнала и его возвратом, и умножение этого времени на постоянное число, равное скорости звука в воздухе (~340 м/с). Однако, для того, чтобы не нагромождать скрипт каждый раз формулами, вычисляющими нужное значение, в mBlock существует специальный блок, который берёт всю работу по расчётам на себя. Его можно найти в группе блоков «Робот»:

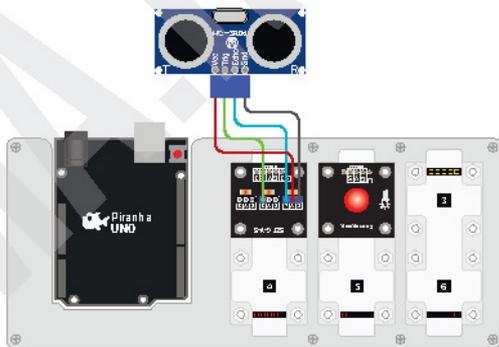
read ultrasonic sensor trig pin ③ echo pin ④

В данном блоке надо только указать те выводы, к которым подключен датчик, и всё, дальше блок выполнит всю необходимую работу самостоятельно и вернёт обратно значение расстояния в сантиметрах!

16.2: Работа с датчиком

Что ж, сначала подключи модули следующим образом:

- «Расширитель» в ячейку 1
- «Датчик расстояния» подключи следующим образом:
 - «Vcc» к выводу V «Расширителя»
 - «Trig» к выводу 2 «Расширителя»
 - «Echo» к выводу 1 «Расширителя»
 - «Gnd» к выводу G «Расширителя»
- «Светодиод» в ячейку 2

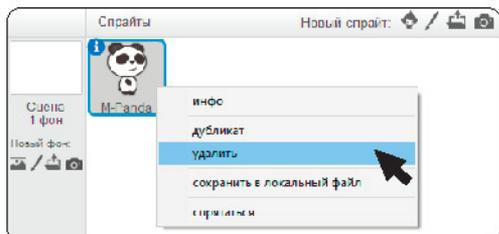


16.3: Создание анимации

В этот раз ты создашь скрипт, который будет работать по аналогии с системой **парк-троник**: «Датчик расстояния» поможет «припарковаться» так, чтобы не ударить соседскую машину, роль которой будет играть твоя рука или любое другое препятствие, а сам процесс будет в реальном времени отображаться в области анимации.

Шаг 1

Для начала тебе надо удалить спрайт «M-Panda». Для этого щёлкни правой кнопкой мыши по спрайту и выбери строку «удалить».



```

когда щёлкнут по [флаг]
всегда
  сказать [ ]
  поставить [Расстояние] в [read ultrasonic sensor trig pin 3 echo pin 4]
  установить размер [20 / Расстояние * 100 %]
  ждать [1] секунд
  если [Расстояние < 5] то
    СВЕТ [1]
    сказать [УДАР!]
    барабану [4] играть [0.25] тактов
  иначе
    сказать [ ]
    СВЕТ [0]
  
```



Помни, что присвоив переменной «состояние светодиода» значение 1/0 и передав его в блок «СВЕТ», ты устанавливаешь данное значение вместо параметра HIGH/LOW блока

установить цифровой pin 6 udgang как HIGH

Шаг 7

Запусти скрипт и проверь работу твоей системы – теперь всё должно работать как надо!



Каверзные вопросы



- Как работает «Ультразвуковой датчик расстояния»?
- Как создать свой собственный блок? Какие возможности он даёт?
- Добавь в скрипт опцию изменения звукового сопровождения в зависимости от расстояния до препятствия.

Урок 17: Автономность

Рано или поздно тебе станет тесно на твоём рабочем столе и захочется выйти за его рамки. В этот момент ты и столкнёшься с главной сложностью на пути к своему желанию – как обеспечить питание платы после того, как она будет отключена от ПК? Об этом мы сейчас и поговорим!

17.1: Про источники питания

Для того, чтобы твоё устройство было полностью мобильным, то есть работало без подключения к ПК, его необходимо оборудовать **источником питания**.

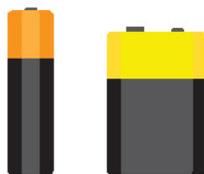
Источник питания используется для обеспечения электрическим питанием других устройств. Он может состоять из различных **элементов питания**.

Элементами питания могут быть **батарейки** или **аккумуляторы**.

Электрическая батарейка – элемент питания, который невозможно перезарядить.

Электрический аккумулятор – элемент питания, который с помощью зарядного устройства можно перезарядить.

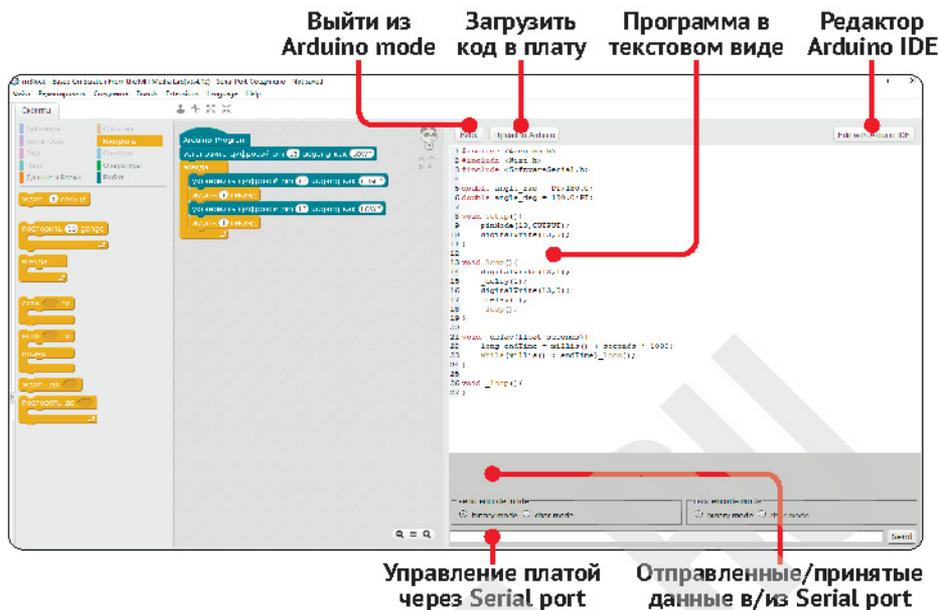
Важно помнить, что плата Arduino/Piranha питается от **напряжения 6-12В**. Это важно потому, что не все источники питания способны обеспечить такое напряжение. Чтобы было понятно, о чём речь, рассмотрим характеристики каждого из вышеперечисленных элементов питания.



Электрические батарейки.

Бывают разных типов, в зависимости от используемых в их конструкции материалов. Для создания источника питания желательно применять **солевые** или **щелочные** батарейки:

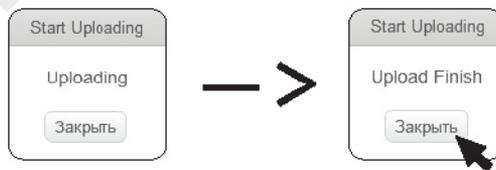
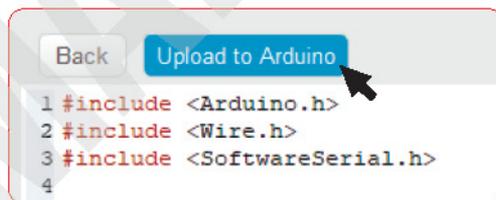
- 1 батарейка типоразмера АА/ААА имеет **напряжение 1.5В, ёмкость 550-3000мА*ч**;
- 1 батарейка типоразмера КРОНА имеет **напряжение 9В, ёмкость 500-650мА*ч**.



Шаг 3

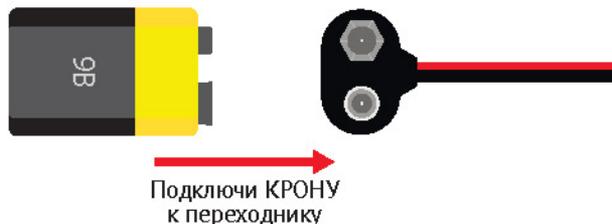
Для загрузки программы нажми «**Upload to Arduino**». После этого начнётся загрузка. Дождись, пока появится надпись «Upload Finish».

Если надпись не появилась или написано «Upload Failed», проверь свой скрипт, наличие подключения и какой выбран COM-порт (стр. 42). В крайнем случае перезагрузи mBlock.



Шаг 4

Теперь установи переходник на КРОНУ: соотнеси контакты на переходнике с контактами на КРОНЕ и вставь их до упора.



Шаг 5

Отключи плату от ПК и вытащи из платы USB-кабель. После этого возьми переходник с КРОНОЙ и вставь штекер в круглый разъем на плате (слева от USB-разъёма).



Шаг 6

Что ж, если теперь светодиод на плате Piranha (около 13-го вывода) каждую секунду загорается и гаснет, то всё сделано верно, и твоя плата работает в автономном режиме. Поздравляем!



P.S. если решишь вернуться в **интерактивный** режим работы, то после подключения платы к ПК и выбора COM-порта нажми меню «Соединить» > «Обновить прошивку», иначе скрипты с участием платы работать не будут!



Каверзные вопросы



- Какие источники питания тебе известны?
- Что такое «ёмкость» и «саморазряд»?
- В чём отличие автономного режима работы от интерактивного?
- Создай любой другой скрипт и запусти его сначала в интерактивном режиме, а потом в автономном (помни, что в автономном режиме **невозможно** выполнять несколько процессов одновременно!)

Проект 9: Умный дом

Для того, чтобы поздравить тебя с окончанием обучающего курса, мы предлагаем создать автономный «умный» дом с автоматическим управлением освещением, воротами, контролем температуры и подачей светового оповещения в случае отклонений от нормы.

Для проекта тебе понадобятся:

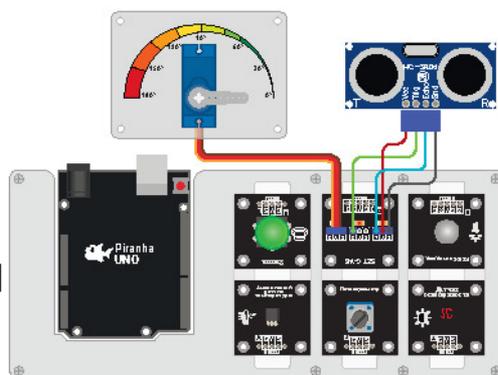
- Модуль «Кнопка»;
- Модуль «Расширитель выводов»;
- Модуль «RGB-Светодиод»;
- Модуль «Датчик температуры»;
- Модуль «Потенциометр»;
- Модуль «Датчик освещённости»;
- Модуль «Датчик расстояния»;
- Сервопривод.

9.1: Подключение

Шаг 1

Подключи модули в ячейки, как показано на рисунке справа:

- «Кнопку» в ячейку **1**
- «Расширитель» в ячейку **2**
- «RGB-светодиод» в ячейку **3**
- «Датчик температуры» в ячейку **4**
- «Потенциометр» в ячейку **5**
- «Датчик освещённости» в ячейку **6**
- Сервопривод в **третью колодку** «Расширителя выводов»
- «Датчик расстояния» в **первую и вторую колодку** «Расширителя выводов»



установить PREPARE_SETTINGS

поставить ZUMMER_PIN в 3

поставить SERVO_PIN в 5

поставить DISTANCE_SENSOR_TRIG_PIN в 6

поставить DISTANCE_SENSOR_ECHO_PIN в 7

поставить GREEN_PIN в 9

поставить RED_PIN в 10

поставить BLUE_PIN в 11

поставить SOUND_SENSOR_ANALOG_PIN в 0

поставить TEMPERATURE_SENSOR_ANALOG_PIN в 1

поставить LIGHTING_SENSOR_ANALOG_PIN в 2

поставить Angle в 0

установить серво pin SERVO_PIN как Angle grader

поставить NORMAL в 0

поставить TEMPERATURE_ALARM в 1

поставить SOUND_ALARM в 2

поставить LIGHTING_BORDER в 500

поставить TEMPERATURE_BORDER в 20

поставить SOUND_BORDER в 700

поставить DISTANCE_BORDER в 10

поставить WorkingResume в NORMAL

поставить RealDistance в read ultrasonic sensor trig pin DISTANCE_SENSOR_TRIG_PIN echo pin DISTANCE_SENSOR_ECHO_PIN

поставить RawTemperatureLevel в читать аналоговый pin (A) TEMPERATURE_SENSOR_ANALOG_PIN

поставить RealTemperatureLevel в округлить $\text{RawTemperatureLevel} * 5 / 1023 - 0.5 * 100$

поставить RealSoundLevel в читать аналоговый pin (A) SOUND_SENSOR_ANALOG_PIN

поставить RealLightingLevel в читать аналоговый pin (A) LIGHTING_SENSOR_ANALOG_PIN



Сначала ты присваиваешь новым переменным значения выводов, к которым подключены модули, тем самым они становятся константами. А потом передаёшь эти константы в блоки группы «Робот». Это сделано для того, чтобы в дальнейшем твой код было легко читать тем, кто видит его впервые.



Тут задаются режимы работы и границы значений для датчиков и сенсоров, от которых будет зависеть, какой сценарий выполняется. Сценариев, к слову, три:

- сценарий нормальной работы, когда **WorkingResume = NORMAL;**
- тревожный сценарий при критическом снижении температуры, когда **WorkingResume = TEMPERATURE_ALARM;**
- тревожный сценарий при критическом превышении уровня громкости, когда **WorkingResume = SOUND_ALARM;**



Последние пять блоков отвечают за опрос датчиков и сенсоров до того, как начнёт выполняться цикл в скрипте, чтобы избежать выполнения тревожных сценариев. Вариантов решения данной проблемы может быть два:

- самостоятельно задать переменным, которые хранят показания сенсоров и датчиков, те значения, которые будут удовлетворять условия режима «NORMAL»;
- до цикла опросить все сенсоры и датчики и записать значения в переменные.

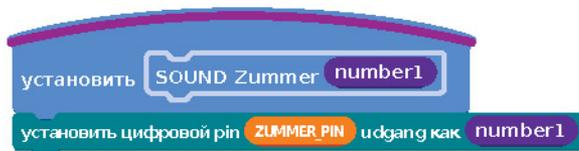
Мы выбрали второй вариант. В этом случае скрипт сразу после запуска работает с реальными значениями.

Блок «**PREPARE_SETTINGS**» является **процедурой**, и в неё ничего не передаётся. Мы обратимся к ней в самом начале скрипта, и она будет выполнена лишь один раз. Мы вернёмся к этому чуть позже, а сейчас просто запомни этот момент и переходи к следующей странице.

Шаг 3

Создай следующие новые блоки (стр. 174), каждый из которых будет отвечать за работу отдельных модулей и сценариев:

– блок «**SOUND**»;



Блок отвечает за работу зуммера.



– блок «**LIGHT**»;



Блок отвечает за работу RGB-светодиода.



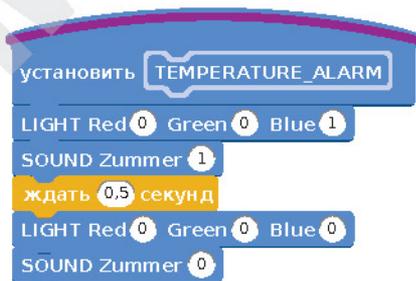
– блок «**SOUND_ALARM**»;



Блок отвечает за работу RGB-светодиода и Зуммера в случае, если громкость шума превысит заданное значение.



– блок «**TEMPERATURE_ALARM**»;



Блок отвечает за работу RGB-светодиода и Зуммера в случае, если температура опустится ниже заданного значения.



Arduino Program

PREPARE_SETTINGS

ждать 1 секунд

всегда

если таймер > 0.1 то

поставить RealDistance в read ultrasonic sensor trig pin DISTANCE_SENSOR_TRIG_PIN echo pin DISTANCE_SENSOR_ECHO_PIN

поставить RawTemperatureLevel в читать аналоговый pin (A) TEMPERATURE_SENSOR_ANALOG_PIN

поставить RealTemperatureLevel в округлить $\text{RawTemperatureLevel} * 5 / 1023 - 0.5 * 100$

поставить RealSoundLevel в читать аналоговый pin (A) SOUND_SENSOR_ANALOG_PIN

поставить RealLightingLevel в читать аналоговый pin (A) LIGHTING_SENSOR_ANALOG_PIN

перезапустить таймер

если WorkingResume = NORMAL то

если RealTemperatureLevel < TEMPERATURE_BORDER то

поставить WorkingResume в TEMPERATURE_ALARM

если RealSoundLevel > SOUND_BORDER то

поставить WorkingResume в SOUND_ALARM

если RealDistance < DISTANCE_BORDER то

SERVO_MOVING

если RealLightingLevel < LIGHTING_BORDER то

LIGHT Red 1 Green 1 Blue 1

иначе

LIGHT Red 0 Green 0 Blue 0

иначе

если WorkingResume = TEMPERATURE_ALARM то

если RealTemperatureLevel = TEMPERATURE_BORDER или RealTemperatureLevel > TEMPERATURE_BORDER то

поставить WorkingResume в NORMAL

TEMPERATURE_ALARM

иначе

если WorkingResume = SOUND_ALARM то

если RealSoundLevel = SOUND_BORDER или RealSoundLevel < SOUND_BORDER то

поставить WorkingResume в NORMAL

SOUND_ALARM



Шаг 7

Ну что ж, можно тебя поздравить! Плата должна работать как часы!



Каверзные вопросы



– Зачем в данном примере все имена переменных, функций и констант указаны латиницей?

– Опираясь на знания из урока 16 (стр. 170), а так же учитывая ответ на первый вопрос, внеси изменения в имена переменных, которые используются в созданных тобой блоках (number1, number2 и т.д.).

– На стадии «отладки» скрипта в интерактивном режиме добавь в проект анимацию «умного» дома, где будут шлагбаум, освещение, а также режимы сигнализации по шуму и температуре.

Что дальше?

Что ж, вот ты и в конце книги. Но это совершенно не означает, что набор тебе больше не понадобится! У тебя впереди ещё масса проектов, которые ты можешь создавать самостоятельно, или взяв их на нашем портале (<https://nabor.iarduino.ru>), посвященном этому набору и многим другим.

В тот момент, когда ты поймёшь, что исчерпал все возможности данного набора, ты можешь смело переходить к наборам, где используется текстовый редактор программ, а работать с платой можно либо напрямую, либо с использованием огромного количества шилдов и модулей!

Теперь ты знаешь о программировании и радиоэлектронике достаточно, чтобы не бояться отправиться во взрослое плавание, где тебя ждут платы, радиоэлементы, провода, паяльники и многое другое, поэтому мы не прощаемся с тобой, а говорим:

«До скорой встречи, дорогой друг!»

Надеюсь, мы скоро увидимся?



Для заметок

TRFEMAF.RU



МЫ В INSTAGRAM

@IARDUINO.RU | #IARDUINO



МЫ В ВКОНТАКТЕ

WWW.VK.COM/IARDUINO



МЫ В YOUTUBE

WWW.YOUTUBE.COM/IARDUINO

Другие увлекательные наборы ищи тут:

<https://iarduino.ru/shop/Nabor/>