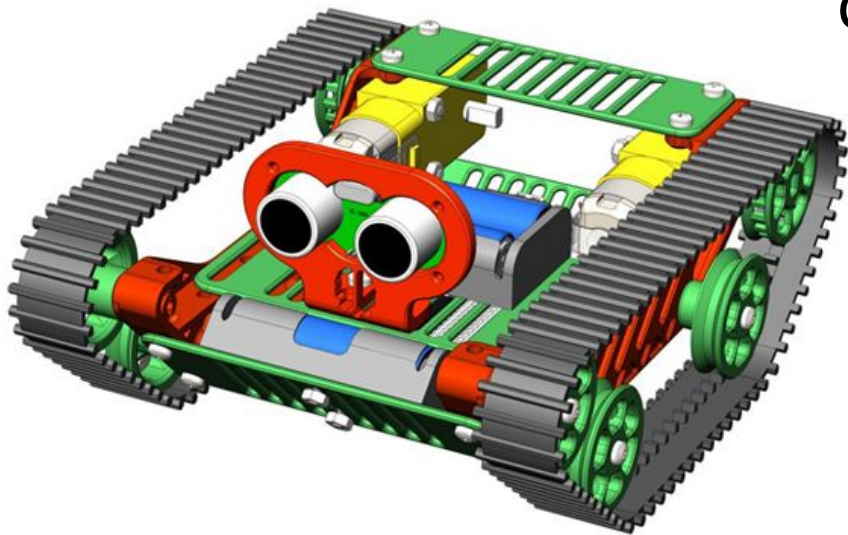


ROBOTON

образовательные роботы

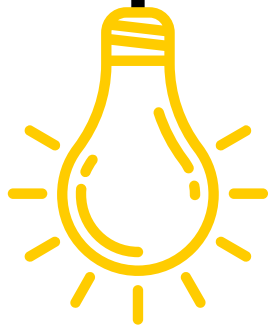


3 курс

2 занятие



Курс “Танк”



Привет!

Сегодня мы познакомимся с новым контроллером и датчиком, вспомним, как работает RobotON Studio, настроим и откалибруем моторы и сонар.



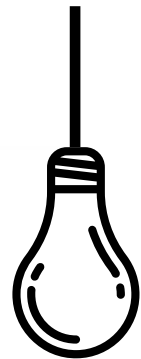


Вспоминаем Микроконтроллер

Микроконтроллер – или МК - это устройство, предназначенное для управления любым относительно сложным электрическим устройством. Обычно для управления такими микроконтроллерами пишется программа из текстового кода.

Наш робот будет управляться таким же микроконтроллером, как на рисунке – модуль esp8266.

Для того, чтобы ты мог управлять своим роботом, мы научим тебя программировать в специально созданной для этого программе – **RobotON Studio**.





Вспоминаем

Как устроен микроконтроллер

Сейчас нам важно изучить, за что отвечает каждый вход/выход (или ножки) МК.

1. Как подать питание на МК? Где «плюс»?

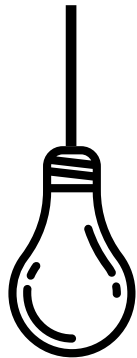
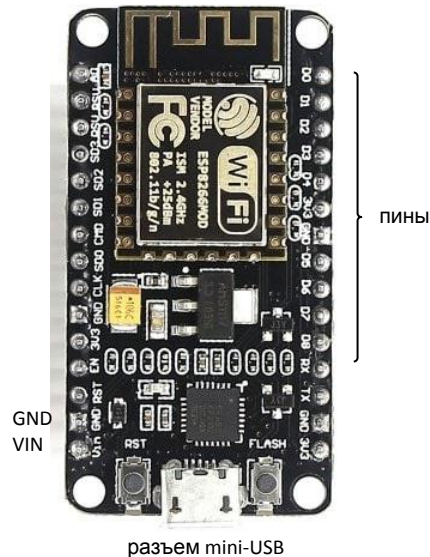
Для этого есть **разъем mini-USB** – для подключения платы к компьютеру и **VIN** для подачи напряжения напрямую от батарейки.

2. Где «минус»?

Минусовыми ножками или землёй являются 2 выхода с названием **GND**, которые располагаются с разных сторон на плате.

3. К чему подключать устройства нашего робота?

Посмотрим на плату и найдем на ней цифры от 1 до 13. Им соответствуют ножки, или как их обычно называют - пины цифровых входов/выходов. Мы будем использовать их чаще всего, чтобы управлять различными устройствами - от светодиодов до различных моторов.



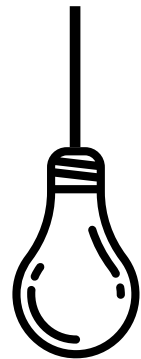
Вспоминаем

RobotON Studio

Для дальнейшей работы тебе необходимо зайти в нашу программу. Она уже установлена на твоём рабочем компьютере. Попроси преподавателя тебе помочь её открыть.

Открой программу и рядом презентацию.

Дальше мы пробежимся по основным блокам программы. Если ты только недавно закончил собирать других наших роботов - можешь на них не останавливаться, или вернуться к ним, если что-то подзабылось.





Вспоминаем

Раздел “Математика”

Внизу экрана можно создать свою переменную. Нужно выбрать тип (пока выбираем "число"), написать название и нажать "+". Тогда в программе создается соответствующий блок.

Подсказка: целое число будет синим цветом, дробное – коричневым, логическая переменная - зеленым.

Зачем нужна переменная?

Например, мы сделали управляемую машинку. На неё можно ставить колеса различного размера. В процессе обучения, вы поймете, что размер колеса сильно влияет на движение робота. Поэтому, чтобы при установке других колёс не изменять всю программу, можно в самом начале создать переменную "Размер колеса" и присвоить ей конкретное значение. Это позволит быстро менять размер колеса сразу во всей программе.

В данном разделе мы также видим 5 блоков, соответствующих основным математическим действиям: сложение, вычитание, умножение, деление и присваивание - блок, с помощью которого можно “присвоить” своей переменной какое-либо значение.

Движение

- Сложение
- Вычитание
- Умножение
- Деление

Датчики

Условия

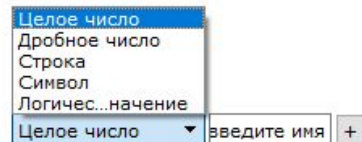
Управление

Математика

- Присвоение
- модуль
- кв. корень
- в степени
- синус
- косинус
- тангенс
- арксинус
- арккосинус
- арктангенс

Мои блоки

Внизу раздела расположено окно для создания переменных. Нужно выбрать тип переменной и ввести её название



Вспоминаем

Раздел “Движение”

Данный раздел содержит блоки, с помощью которых будет осуществляться управление моторами. Так как двигатели бывают разного типа, то и логика управления ими разная. Поэтому в данном разделе существует блок управления сервомотором, мотором постоянного тока и шаговым мотором.

Движение

Серво на угол

Мотор со скоростью

Датчики

Шаговый повернуть на со ск

Шаговики повернуть на

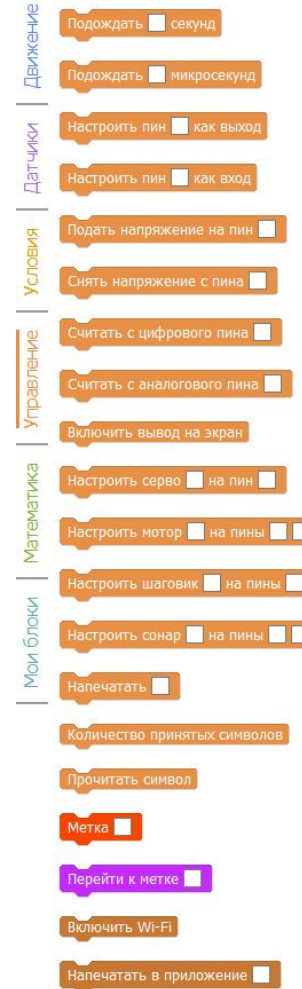


Вспоминаем

Раздел “Управление”

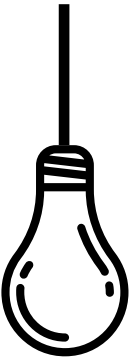
В данном разделе будет происходить предварительная настройка нашего устройства, подключенного к плате. Здесь мы настраиваем:

- ❖ ножки платы Arduino и действия с ними;
- ❖ сервоприводы, моторы, шаговики и т.д.;
- ❖ задержки в работе нашей программы;
- ❖ “метки” отмечающие места в программе, к которым можно вернуться, с помощью “Перейти к метке”;
- ❖ вывод на экран и печать - возможность наблюдать за результатом выполнения нашей программы.



The screenshot shows the Arduino IDE interface with a vertical sidebar on the left containing the following categories and their corresponding blocks:

- Движение**: Подождать секунд, Подождать микросекунд
- Датчики**: Настроить пин как выход, Настроить пин как вход
- Условия**: Подать напряжение на пин (orange), Снять напряжение с пина (orange)
- Управление**: Считать с цифрового пина (orange), Считать с аналогового пина (orange), Включить вывод на экран (orange)
- Математика**: Настроить серво на пин (orange), Настроить мотор на пины (orange)
- Мои блоки**: Настроить шаговик на пины (orange), Настроить сонар на пины (orange), Напечатать (orange), Количество принятых символов (orange), Прочитать символ (orange), Метка (orange), Перейти к метке (purple), Включить Wi-Fi (orange), Напечатать в приложение (orange)

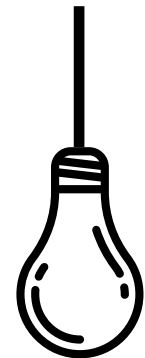


Вспоминаем

Вывод на экран

Очень часто необходимо посмотреть результат работы программы на компьютере, чтобы убедиться в корректности работы программы. Например, Вам нужно узнать, правильно ли меряет датчик расстояния до препятствия. Для этого нужно будет **“Включить вывод на экран”** соответствующим блоком, а затем **“Напечатать”** его.

Чтобы увидеть полученные значения, надо открыть терминал.

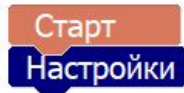


Вспоминаем



Как работать в RobotON Studio

САМОЕ ВАЖНОЕ: любая программа строится по этому принципу:



В рабочем поле **ВСЕГДА** должен быть 1 старт, подключенный к 1 определенной программе.

2. Как выполнять наши задания:

Подключи к твоему компьютеру МК и настрой его.

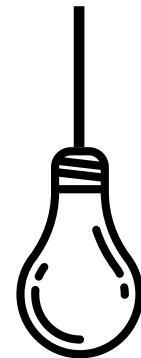
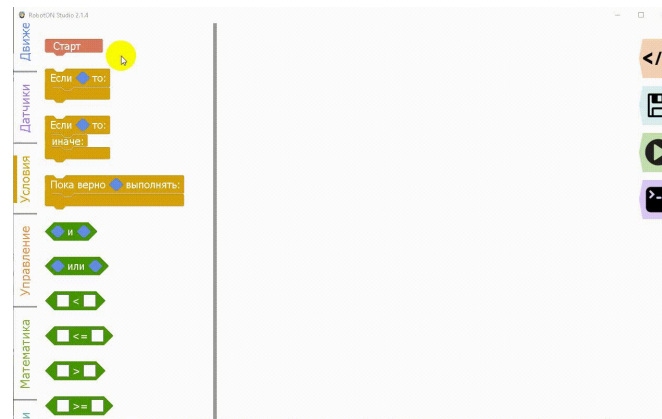
Повтори программу у себя на экране.

Нажми зеленую кнопку.

В появившемся окне нажми “загрузить”.

Когда программа загрузится, открой терминал и посмотри результат.

1. Как составить программу:



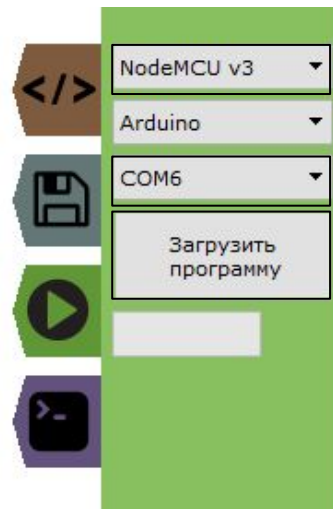


Вспоминаем

Как подключить плату к компьютеру

Для того, чтобы запрограммировать плату Arduino, нужно подключить ее к компьютеру через USB. После этого необходимо открыть RobotON Studio и нажать на вкладку с зеленым треугольником. В ней необходимо правильно выбрать тип подключенной платы (в нашем случае NodeMCU v3) и порт, к которому она подсоединена (гнездо USB). Номер COM-порта проще всего проверить, отсоединив плату: тот, что исчезнет - точно твой.

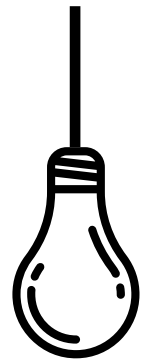
В дальнейшем мы будем загружать программы в МК, для этого нажми кнопку “Загрузить программу”.



Выбери NodeMCU v3

Выбери порт

Загрузи программу в МК





Напишем программу

Давай напишем самую простую программу, с которой начинается обучение всех программистов.

- ❖ Подключи плату к компьютеру, если еще не сделал это.
- ❖ Повтори программу у себя на экране.
- ❖ Нажми зеленую кнопку.
- ❖ В появившемся окне нажми **“загрузить”**.
- ❖ Когда программа загрузится, открой терминал и посмотри результат.

Старт

Включить вывод на экран

Напечатать Привет МИР!!!





Калибровка моторов

Вернемся к работе с нашим Танком.

Перед окончательной установкой схемы в корпус танка необходимо удостовериться, что она корректно работает.

Сперва проверим, правильно ли крутятся моторы. Для этого соберем программу и протестируем вращение колес.



```
Старт
Настроить мотор мотор_левый на пины 0 5
Настроить мотор мотор_правый на пины 2 4
Скорость вперед = 100
Мотор мотор_левый со скоростью Скорость вперед / 2 - 51
Мотор мотор_правый со скоростью Скорость вперед / 2 - 51
```



Обрати внимание! Тебе нужно создать переменную `скорость вперед` (блок “Математика”)

При таком коде моторы должны двигать колеса таким образом, чтобы они вращались в одну сторону и толкали Танк вперед. В случае, если одно из колес двигается не в ту сторону необходимо поменять местами контакты мотора в терминале шилда.

Задание. Поменяй значения в программе так, чтобы танк поехал назад.

Запоминаем новое

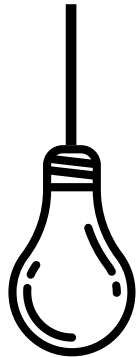
Знакомься: сонар



Датчики расстояния - или сонары - это устройства, посылают сигнал впереди себя. Когда сигнал наталкивается на препятствие, отражается от него и прилетает обратно, сонар это фиксирует и измеряет расстояние, которое успел пролететь сигнал.



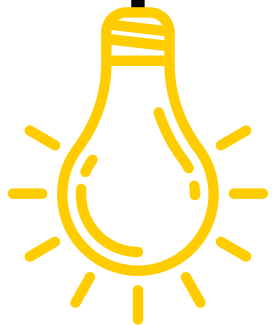
Благодаря сонару наш танк будет ориентироваться в пространстве и находить себе дорогу так, чтобы не наталкиваться на стены и препятствия.





Проверка сонара

Проверим, как работает наш сонар.



Сначала объясним нашему роботу, как у него подключен новый датчик:

Создадим новую переменную `расстояниеСм` и настроим её так, чтобы она была равна расстоянию до препятствия, которое считывает сонар. Добавим блок `Напечатать расстояниеСм`.

```
Старт
Включить вывод на экран
Настроить сонар датчик1 на пины 11 12
Пока верно истина выполнять:
  расстояниеСм = Считать расстояние датчик1
  Напечатать расстояниеСм
  Подождать 1 секунд
```

Запусти программу и поставь перед сонаром руку. А чтобы увидеть, как меняются значения на экране - приближай или отодвигай ладонь от датчика.



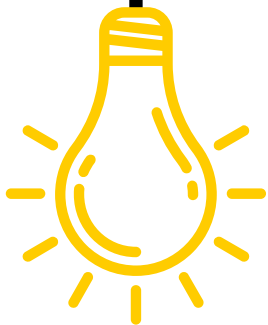
Проверка сонара

Теперь добавим к проверке сонара моторы.

Сначала объясним нашему роботу, как у него подключены оба мотора.

Далее возьмем цикл, поместим в него часть настроек моторов, который у нас уже есть.

И добавим условие, если расстояние меньше 10 см, то танк должен остановить моторы.



```
Старт
Настроить сонар датчик на пины 13 15
Настроить мотор мотор_левый на пины 0 5
Настроить мотор мотор_правый на пины 2 4
Скорость вперед = 100
```

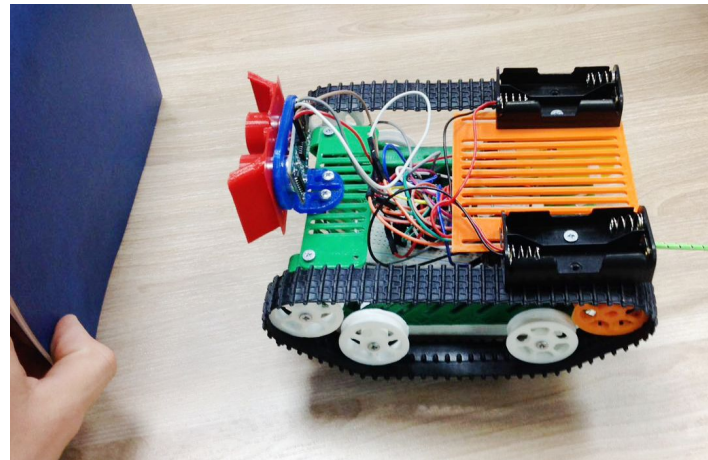
```
Пока верно истина выполнять:
  Мотор мотор_левый со скоростью Скорость вперед / 2 - 51
  Мотор мотор_правый со скоростью Скорость вперед / 2 - 51
  РасстояниеСм = Считать расстояние датчик
  Если РасстояниеСм < 10 то:
    Мотор мотор_левый со скоростью 0
    Мотор мотор_правый со скоростью 0
    Подождать 0.1 секунд
```




Результат



При запуске программы Танк должен ехать до тех пор, пока не увидит препятствие на расстоянии 10 см. Можешь попробовать и во время движения поставить перед ним тетрадь.





Поиск пути

Теперь, увидев препятствие, Танк остановится. А как заставить его изменить направление движения и искать новый путь?

Пересоберем нашу программу.
Обрати внимание на новую переменную

скорость назад

Теперь, если сонар не видит препятствие, он спокойно двигается вперед.

Если же на расстоянии 10 см сонар что-то обнаружит, он даст команду Танку отъехать назад.

```
Пока верно истина выполнять:  
  расстояние = Считать расстояние датчик  
Если расстояние > 10 то:  
  скорость вперед = 100  
  Мотор мотор_левый со скоростью скорость вперед / 2 - 51  
  Мотор мотор_правый со скоростью скорость вперед / 2 - 51  
  Подождать 0.1 секунд  
иначе:  
  скорость назад = -100  
  Мотор мотор_левый со скоростью скорость назад / 2 - 50  
  Мотор мотор_правый со скоростью скорость назад / 2 - 50  
  Подождать 0.1 секунд
```





Разворот



Теперь попробуем научить наш танк разворачиваться.

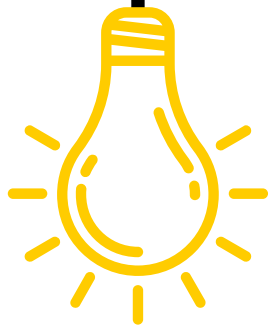
Разворот танка происходит тогда, когда одна из его гусеница крутится в одну сторону, а другая в другую, или же просто стоит.

Поэтому нам нужно заставить гусеницы двигаться одновременно вразнобой так, чтобы танк повернулся на примерно 180 градусов.





Разворот



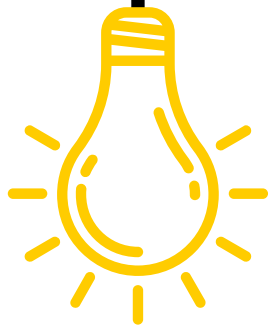
Нам предстоит собрать такую программу, которая бы заставила робота проехать прямо 5 секунд, развернуться, и поехать обратно. Можно зациклить этот процесс, чтобы танк ездил туда-обратно.

Для этого нам понадобятся предыдущие знания, а также осуществить подбор времени, которое потребуется танку для разворота.

```
Старт
Настроить мотор мотор_левый на пины 0 5
Настроить мотор мотор_правый на пины 2 4
Скорость = 100
Пока верно истина выполнять:
  Мотор мотор_левый со скоростью Скорость / 2 - 51
  Мотор мотор_правый со скоростью Скорость / 2 - 51
  Подождать 5 секунд
  Мотор мотор_левый со скоростью Скорость / 2 - 51
  Мотор мотор_правый со скоростью - Скорость / 2 - 51
  Подождать 2 секунд
```



Результат



Создав программу, мы довольно быстро подобрали время разворота танка на 180 (примерно) градусов. Попробуй подобрать идеальное время, указав его с точностью до сотой (например, 0.93 секунды).

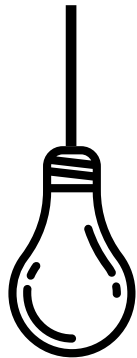
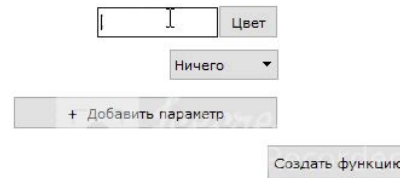


Вспоминаем

Функция

Функция - это блок в программе. В функцию ты можешь записать значения переменных, действия - всё, что захочешь. При дальнейшей работе с программой, если тебе понадобится аналогичная часть программы, ты сможешь скопировать ранее созданную функцию и вставить в нужное место, а не собирать её заново.

В RobotON Studio создание функций находится в разделе “Мои блоки”. Здесь ты можешь задать имя функции, её цвет и содержание - типы переменных.





Применение функций



Переходим к следующему этапу - применению функций. Соберем функцию для разворота, чтобы каждый раз не создавать её заново.

Замени функцией аналогичный кусок кода.

```
Разворот
Мотор мотор_левый со скоростью Скорость / 2 - 51
Мотор мотор_правый со скоростью - Скорость / 2 - 51
Подождать 2 секунд
```

```
Старт
Настроить мотор мотор_левый на пины 0 5
Настроить мотор мотор_правый на пины 2 4
Скорость = 100
Пока верно истина выполнять:
Мотор мотор_левый со скоростью Скорость / 2 - 51
Мотор мотор_правый со скоростью Скорость / 2 - 51
Подождать 5 секунд
Разворот
```



Результат



Видно что наш танк едет так, как мы и планировали. Те блоки, которые стоят в красной функции можно менять, тем самым изменяя код в основной программе. Попробуйте поставить различные значения времени и посмотрите на результат.





Друзья! Наше занятие подошло к концу!
Обязательно сохраните код, который вы написали сегодня.
До встречи!