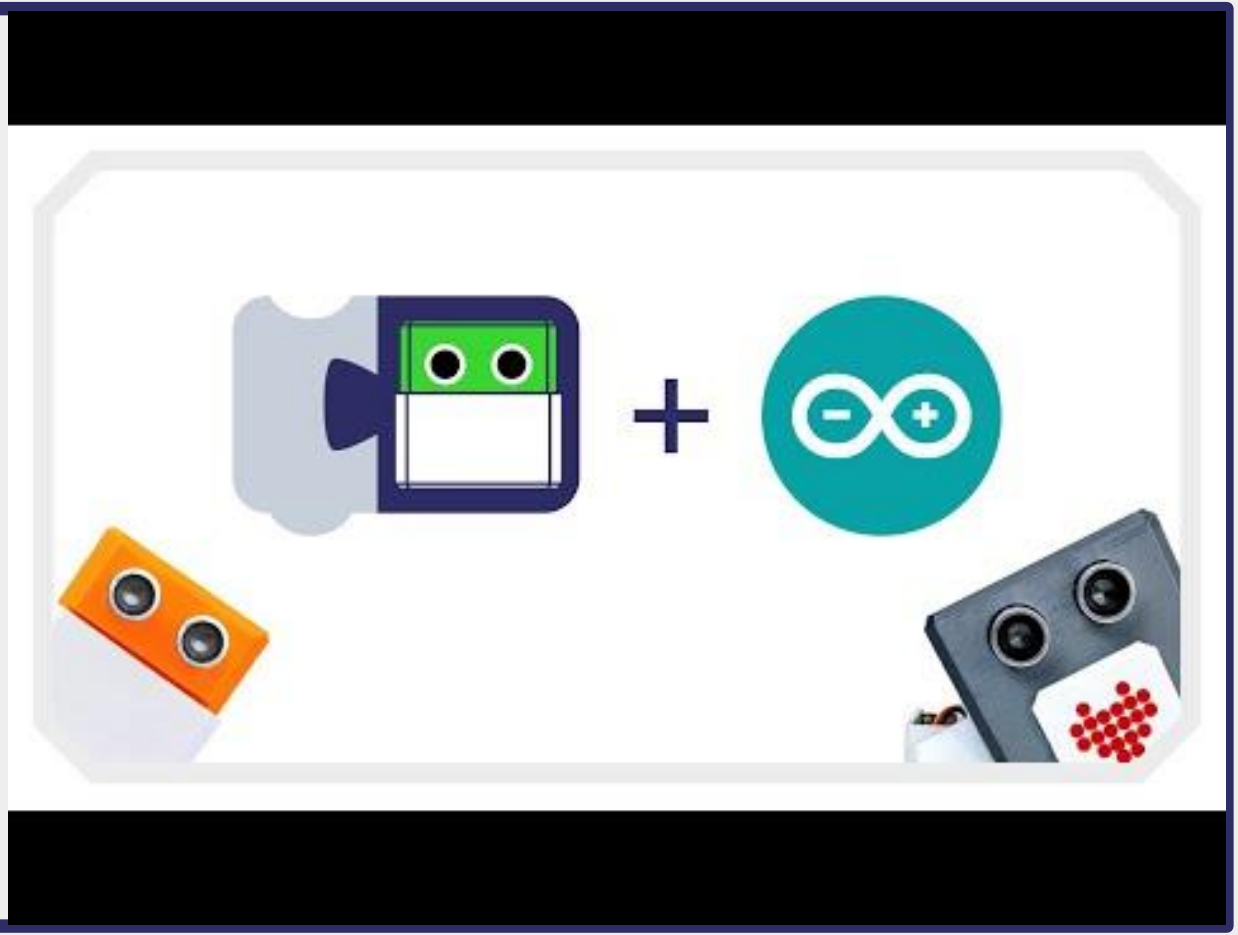
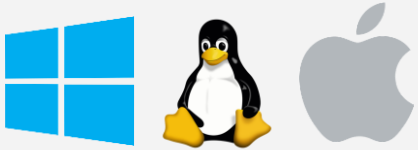
Otto Blockly



Для загрузки кода, есть **2** способа...

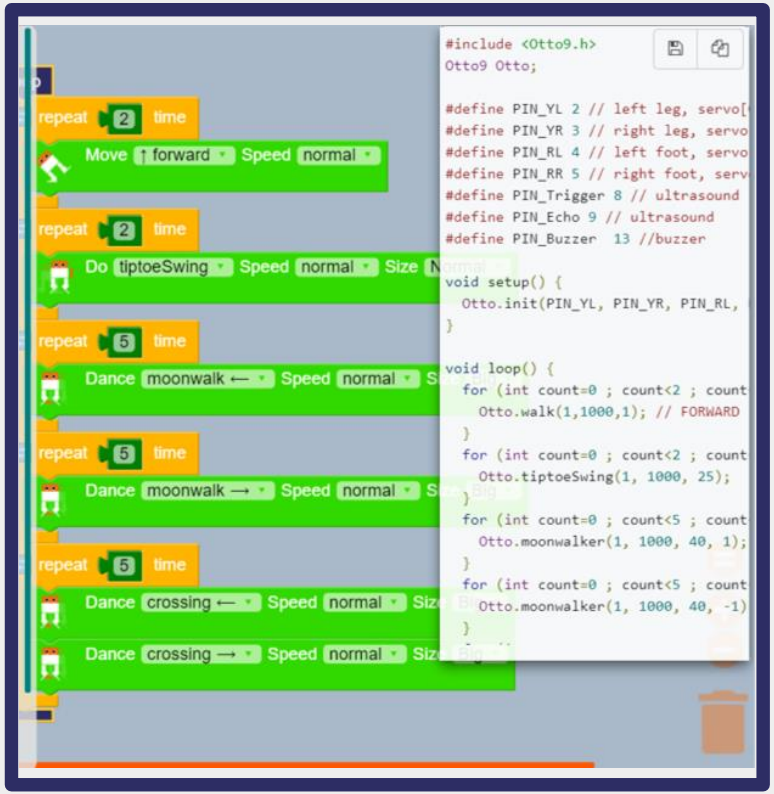
Otto Blockly Standalone: работает в автономном режиме, теперь совместим со всеми ОС, прост в использовании, все в одном кодировании и загрузке через **USB** (нет необходимости в библиотеках **Arduino**)



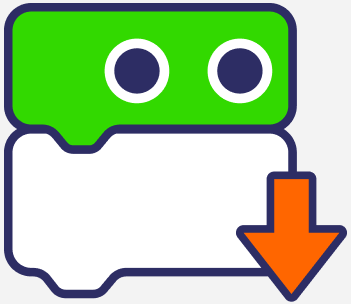
Arduino IDE: Работает в автономном режиме, совместим со всеми ОС, требует дополнительной установки библиотек и опыта программирования на **C++**

Программируйте своего собственного робота

1. Установка и настройка программного обеспечения.
2. Ознакомление с средой блочного программирования.
3. Первые простые проекты, например, программирование танца с Отто.
4. Разбираемся, как действовать последовательно и применять условия.
5. Подготовка более сложных проектов, включая взаимодействие с датчиками и использование более продвинутых методов программирования.



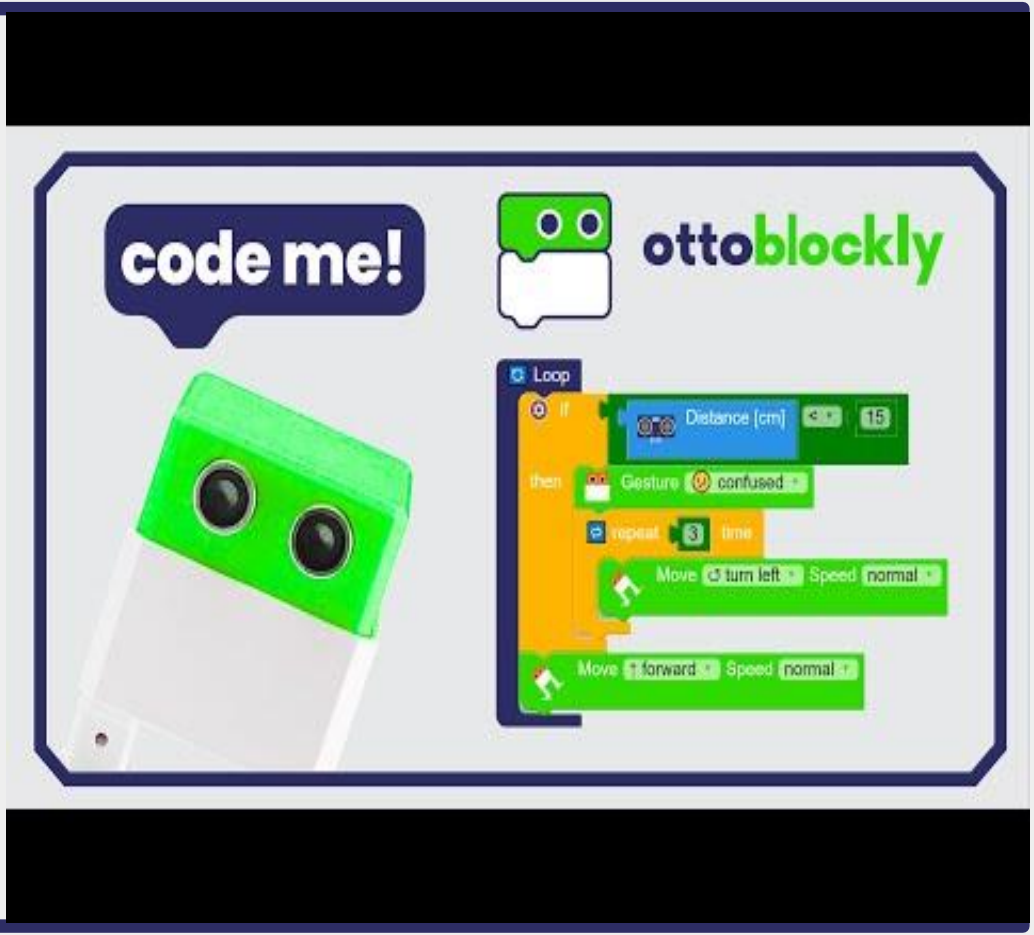
Загружаем **Otto Blockly**



ottoblockly

** Ваш антивирус может сказать вам, что программное обеспечение небезопасно, но это не так 😀, просто продолжайте*

Установите

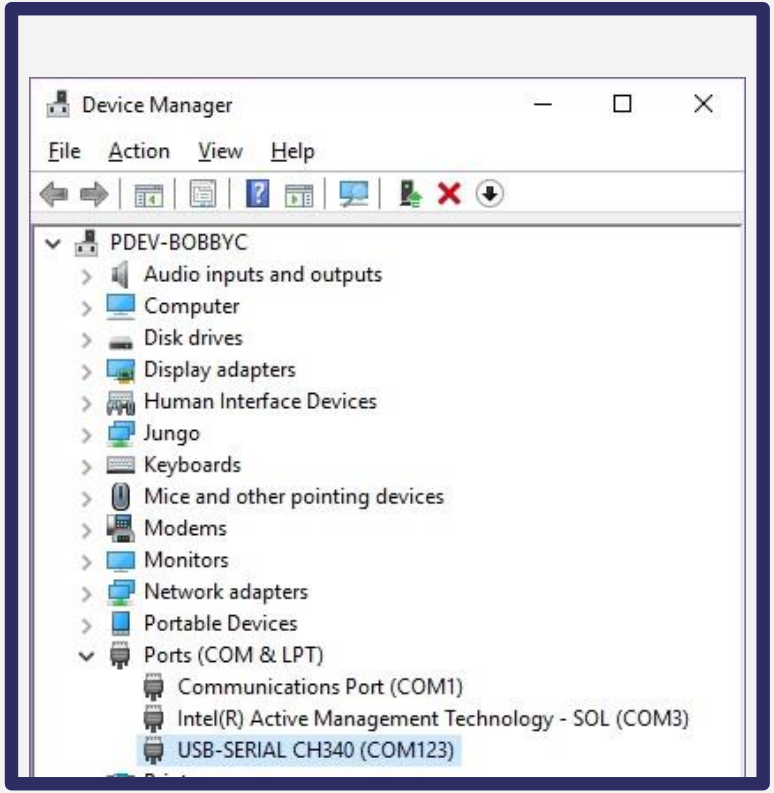

code me!
ottoblockly
Loop
if
Distance [cm]
15
then
Gesture
confused
repeat
3
time
Move
turn left
Speed
normal
Move
forward
Speed
normal

Установка USB-драйвера

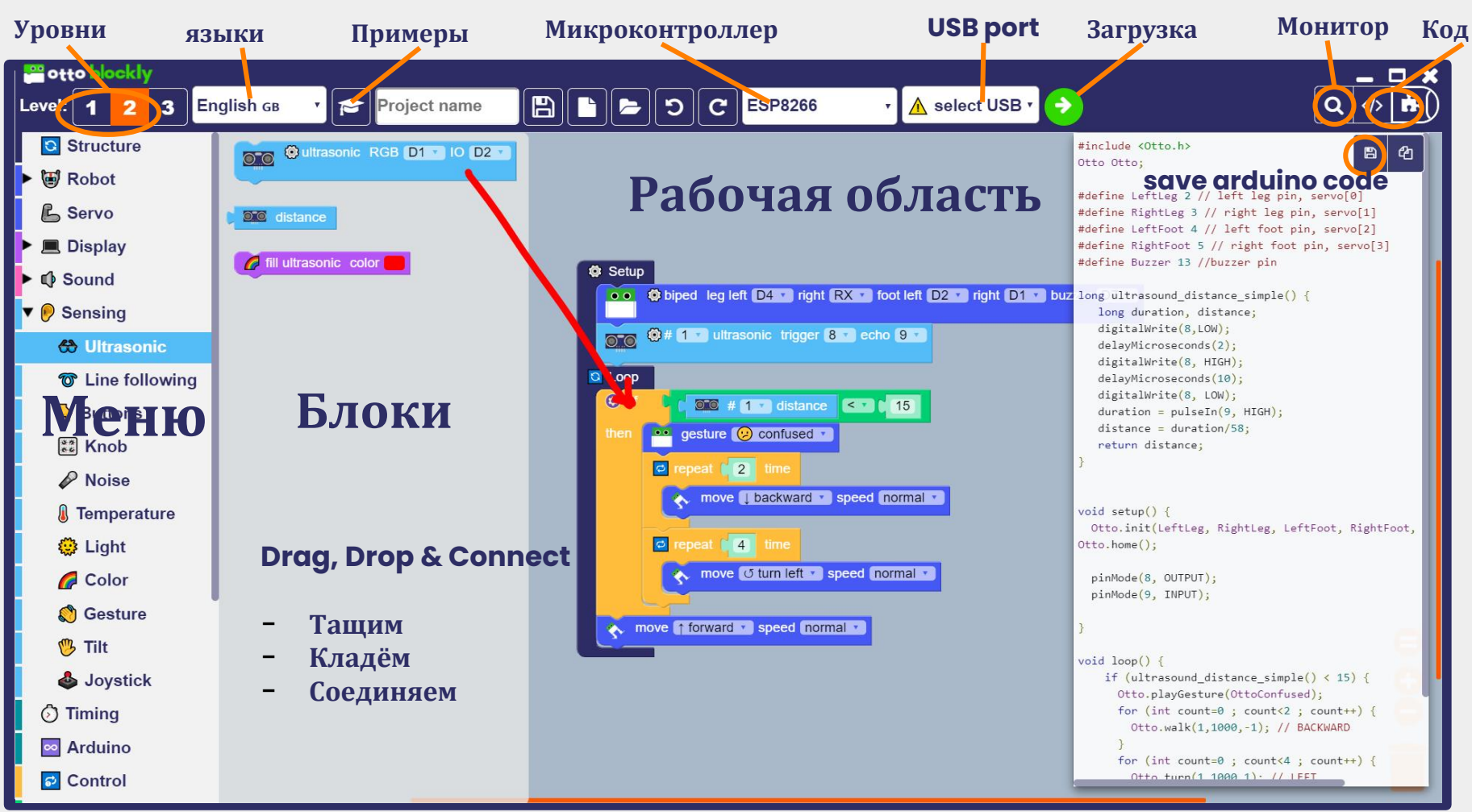
Некоторые компьютеры автоматически распознают **Otto** как **USB**-устройство при подключении, но некоторые этого не делают. Если компьютер не распознал **USB**-устройство, необходимо скачать и установить драйвер **USB**:

[USB serial CH340, download it here]

[This guide covers all OS step by step]

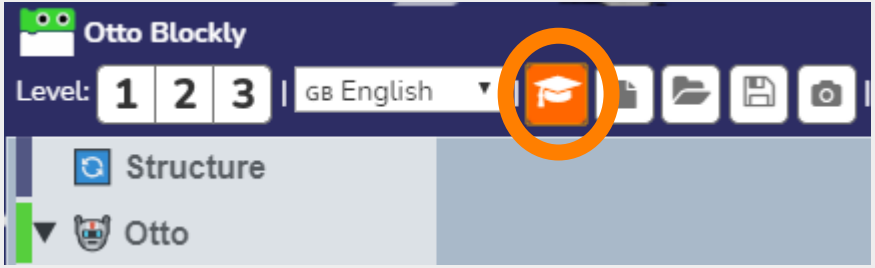


Пользовательский интерфейс **Otto Blockly**

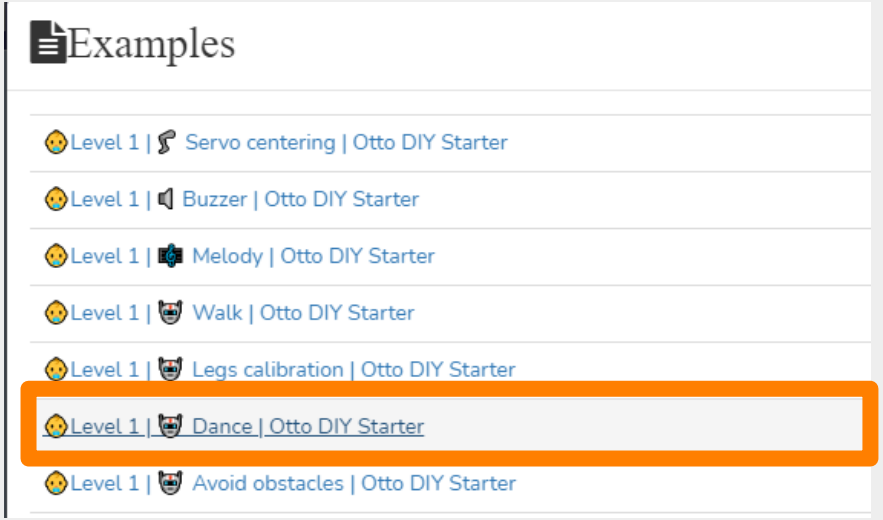

Уровни
языки
Примеры
Микроконтроллер
USB port
Загрузка
Монитор
Код
Level 1 2 3
English GB
Project name
ESP8266
select USB
Structure
Robot
Servo
Display
Sound
Sensing
Ultrasonic
Line following
Buttons
Knob
Noise
Temperature
Light
Color
Gesture
Tilt
Joystick
Timing
Arduino
Control
Меню
ultrasonic RGB D1 IO D2
distance
fill ultrasonic color
Блоки
Drag, Drop & Connect
- Тащим
- Кладём
- Соединяем
Рабочая область
Setup
biped leg left D4 right RX foot left D2 right D1
1 ultrasonic trigger 8 echo 9
Loop
1 distance < 15
then
gesture confused
repeat 2 time
move backward speed normal
repeat 4 time
move turn left speed normal
move forward speed normal
save arduino code
#include <Otto.h>
Otto Otto;
#define LeftLeg 2 // left leg pin, servo[0]
#define RightLeg 3 // right leg pin, servo[1]
#define LeftFoot 4 // left foot pin, servo[2]
#define RightFoot 5 // right foot pin, servo[3]
#define Buzzer 13 //buzzer pin
long ultrasound_distance_simple() {
long duration, distance;
digitalWrite(8,LOW);
delayMicroseconds(2);
digitalWrite(8, HIGH);
delayMicroseconds(10);
digitalWrite(8, LOW);
duration = pulseIn(9, HIGH);
distance = duration/58;
return distance;
}
void setup() {
Otto.init(LeftLeg, RightLeg, LeftFoot, RightFoot,
Otto.home();
pinMode(8, OUTPUT);
pinMode(9, INPUT);
}
void loop() {
if (ultrasound_distance_simple() < 15) {
Otto.playGesture(OttoConfused);
for (int count=0 ; count<2 ; count++) {
Otto.walk(1,1000,-1); // BACKWARD
}
for (int count=0 ; count<4 ; count++) {

Тест быстрого старта

Hi Otto Builder
Я вернулся, готов помочь вам.

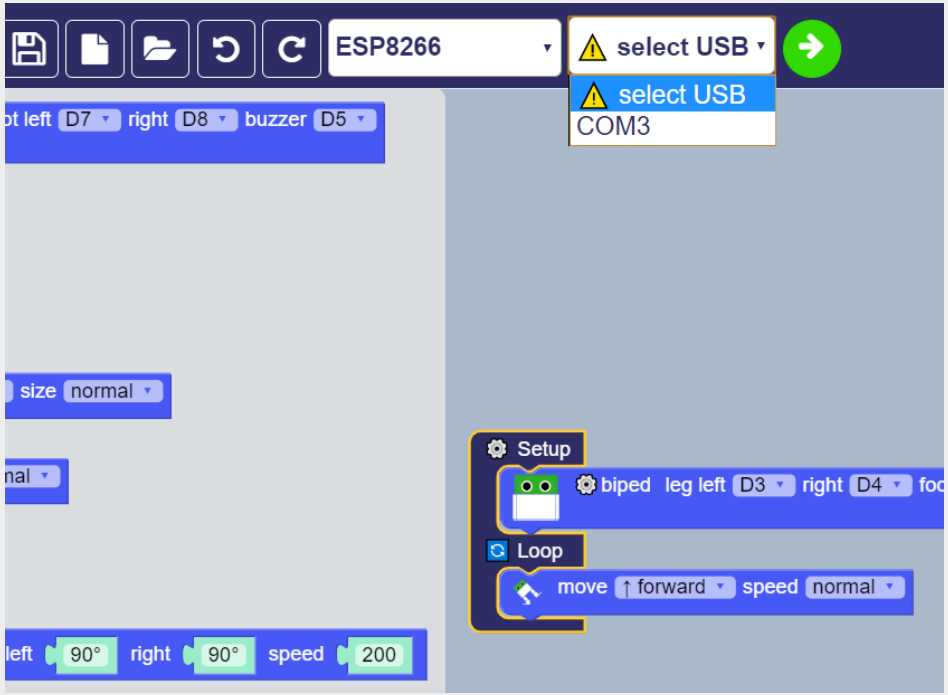


1 Давайте откроем пример, нажав кнопку шляпы.



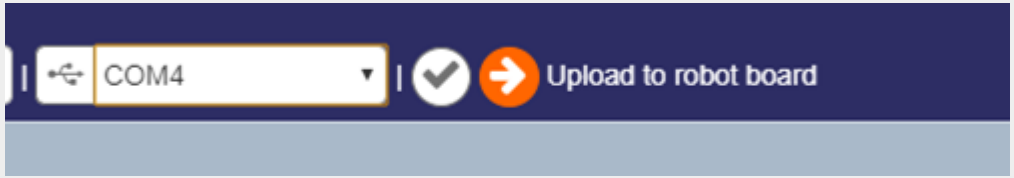
2 Выберите пример **«Biped dance»**.

Если вы любите приключения, выберите другой пример

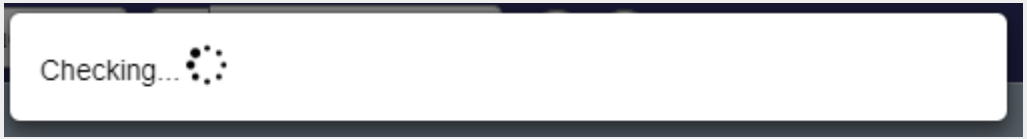


3 **Подключите робота и убедитесь, что в выпадающем меню USB – порта выбран Com port**

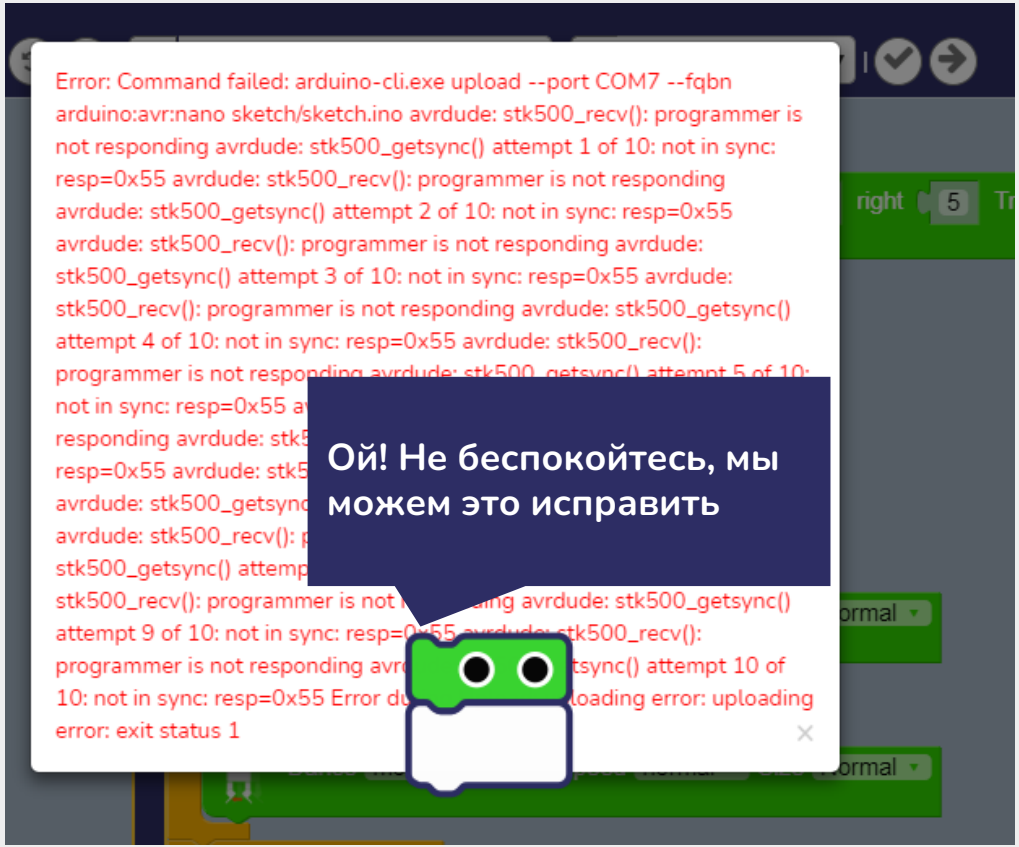
* Если ничего не появляется, значит, USB-драйвер не был правильно установлен в вашем компьютере, необходимо перезагрузить компьютер и повторить попытку.



4 **Нажмите кнопку загрузки.**



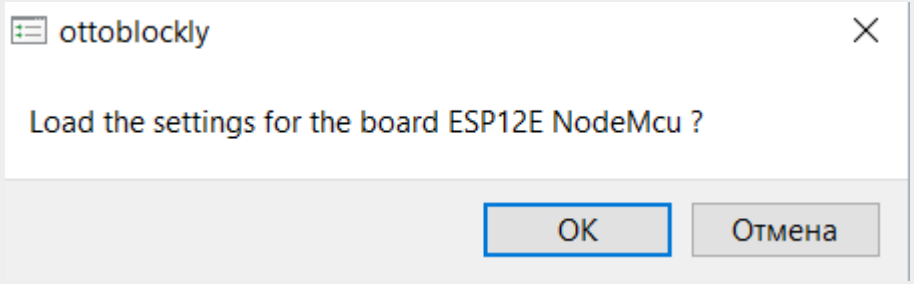
Процесс загрузки может занять около минуты.



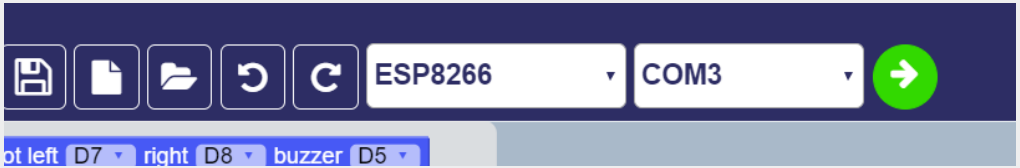
4 Иногда, при использовании блочного программирования и попытке загрузить программу на порт, появляется ошибка в красном цвете в **Otto Blockly**.

Возможное решение:

- убедитесь, что ваш робот подключен к компьютеру
- выберите правильный порт в настройках программы
- Выбран **ESP8266**



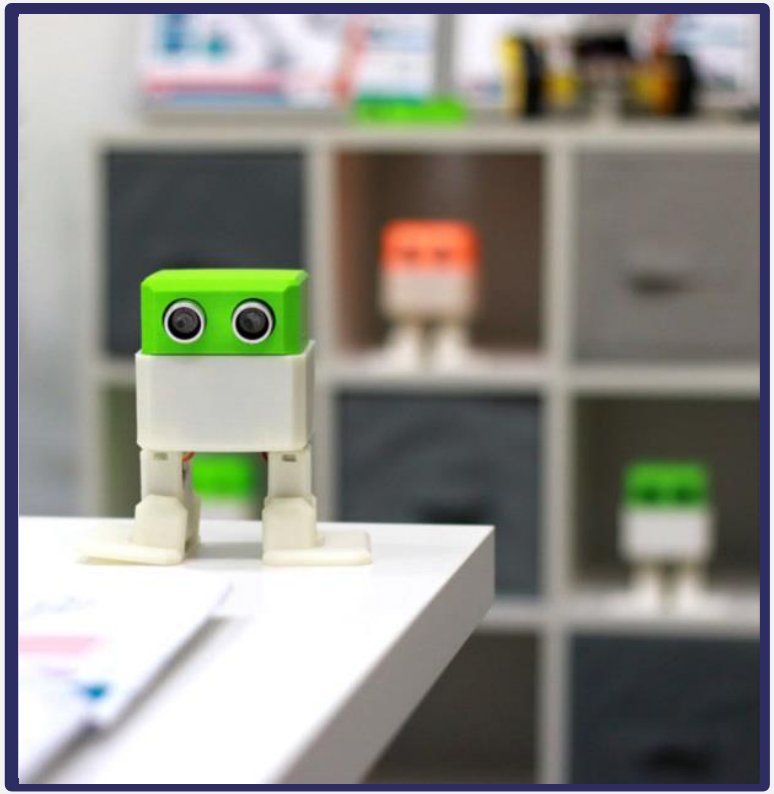
5 **Успешная загрузка кода, когда выбрана правильная плата – ESP8266 или по другому – ESP12E.**



Это означает, что ваш код был загружен в Otto! И теперь ваш робот должен танцевать.
Ура, вы это сделали, поздравляю! Увидимся

Быстрый старт резюме

1. Установить **Otto Blockly**
2. Открыть пример
3. Подключите свой **Otto**
4. Выберите **ESP8266**
5. Выберите **USB COM#**
6. Нажмите «Загрузить»
7. И немного подождите



Распространенные проблемы:

1. Отто не подключается**?**

- Установите драйвер **USB CH340,** чтобы увидеть его в **COM-**порту**.**

2. Не можете загрузить код**?**

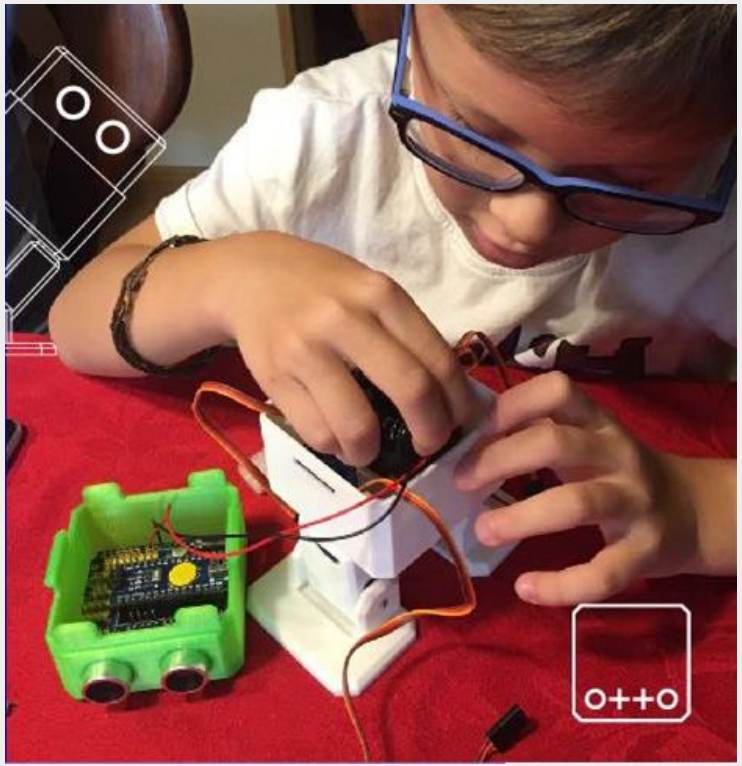
- Выбрана неправильная плата **(**попробуйте выбрать **ESP8266)**

3. Сбрасывает ли **Otto** время от времени**?**

-Низкое качество **USB** разъёма или плохое соединение проводом**.**

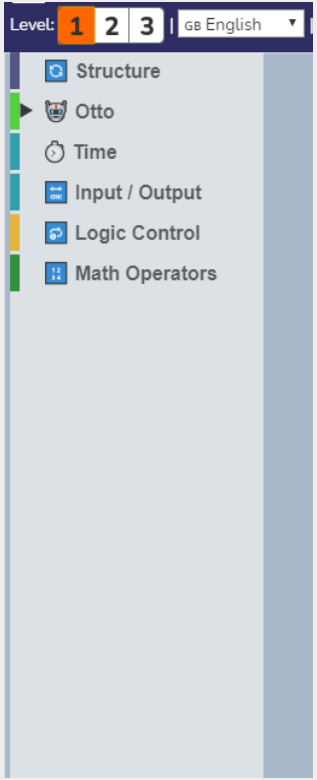
4. У Отто вывернуты ноги и ступни**?**

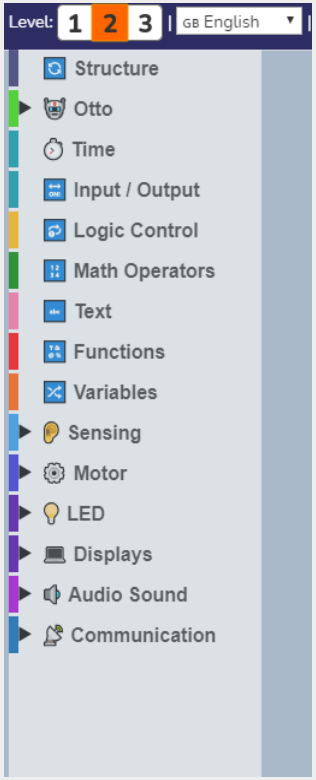
- Убедитесь**,** что вы отцентрировали и выровняли сервоприводы при сборке**.**

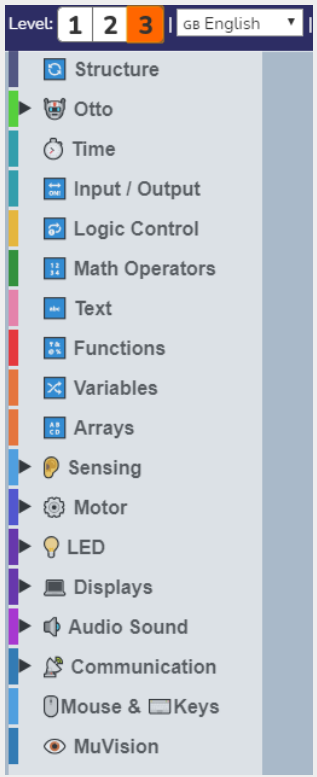


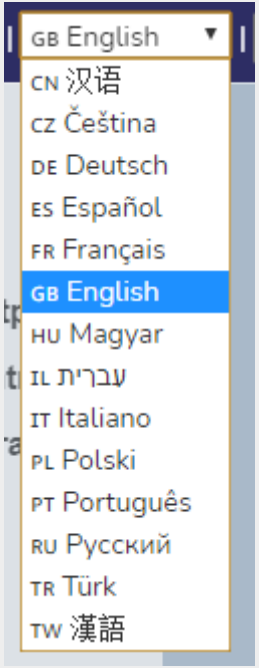
Теперь давайте изучим основы

Уровни и язык



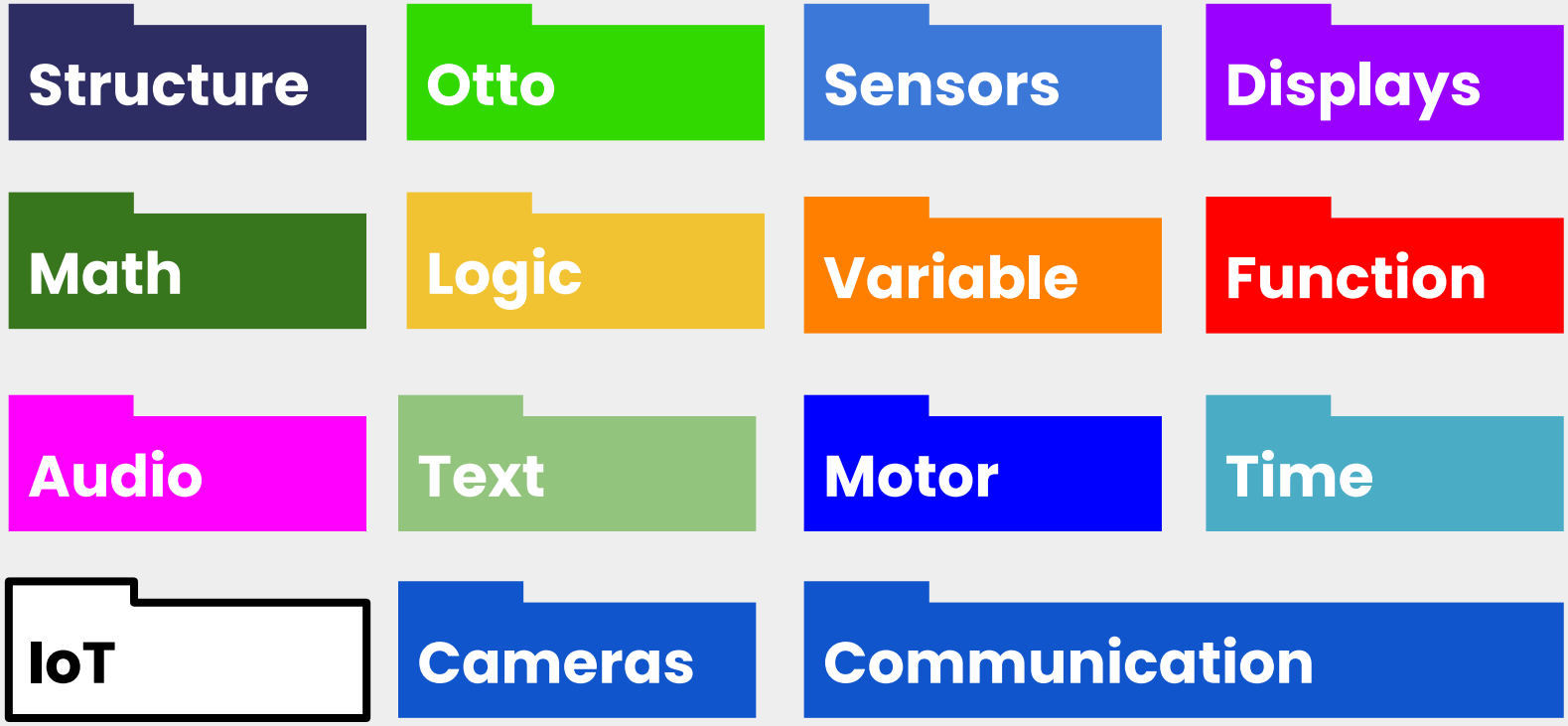


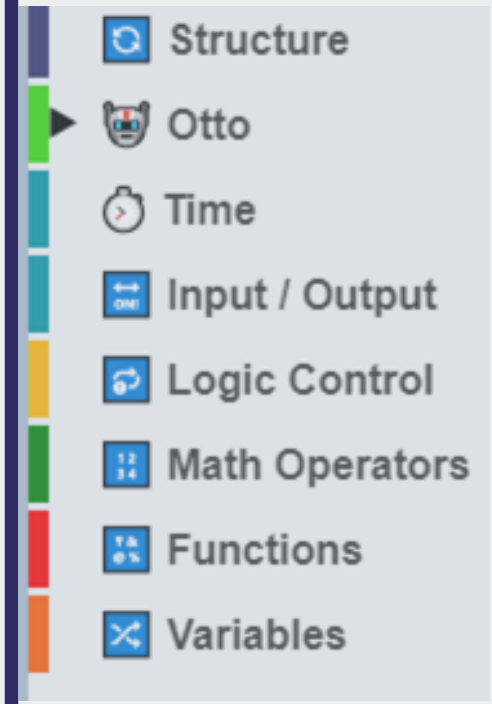




Помогите нам улучшить и добавить новые языки здесь

Категории в меню (Toolbar)

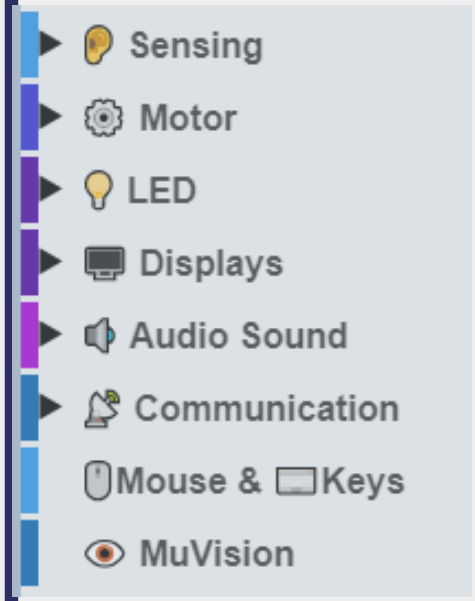




Level 1 (Уровень 1)

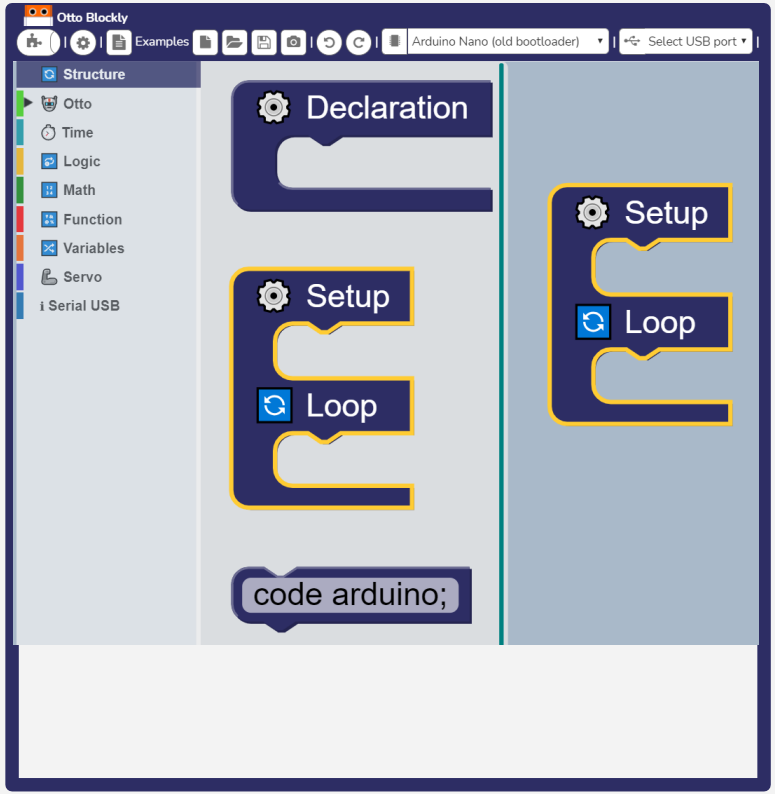
Меню для начинающих со всеми основными движениями Отто

- **Structure** (Структура): Настройка и основы программирования.
- **Otto** (Отто): Движения, звуки и жесты.
- **Sensors** (Датчики): Работа с ультразвуковыми, сенсорными, звуковыми и гироскопическими датчиками.
- **Time** (Время): Использование времени в программах.
- **Input/Output** (Ввод/вывод): Управление портами **Arduino GPIO**.
- **Logic** (Логика): Создание условий, циклов и логического программирования.
- **Math** (Математика): Выполнение математических операций и вставка чисел.
- **Functions** (Функции): Создание повторяющихся процедур.
- **Variables** (Переменная): Работа с именованными значениями, которые можно изменять.



Уровни 2 и 3 включают в себя более продвинутые инструменты для программирования Arduino и более широкий выбор возможностей:

- **Sensing** (Датчики): Возможность интеграции различных датчиков в проекты Отто или в создание других проектов с использованием Arduino.
- **Motor** (Мотор): Управление всеми видами сервоприводов, шаговых двигателей и постоянных моторов.
- **LED** (Светодиоды): Возможность управления RGB-светодиодами, Neopixels и светодиодными матрицами.
- **Displays** (Дисплеи): Использование дисплеев, таких как ЖК-дисплеи, OLED и другие виды экранов.
- **Audio** (Звук): Воспроизведение звука с помощью звуковых излучателей или воспроизведение MP3-файлов.
- **Communication** (Связь): Взаимодействие с различными модулями, такими как Bluetooth, IrDa или последовательная связь.
- **Keyboard and Mouse** (Клавиатура и мышь): Доступно только для Arduino Leonardo и Micro, это позволяет управлять Отто с клавиатуры и мыши.
- **MuVision** (MuVision): Запись видео или более сложное взаимодействие с распознаванием изображений.

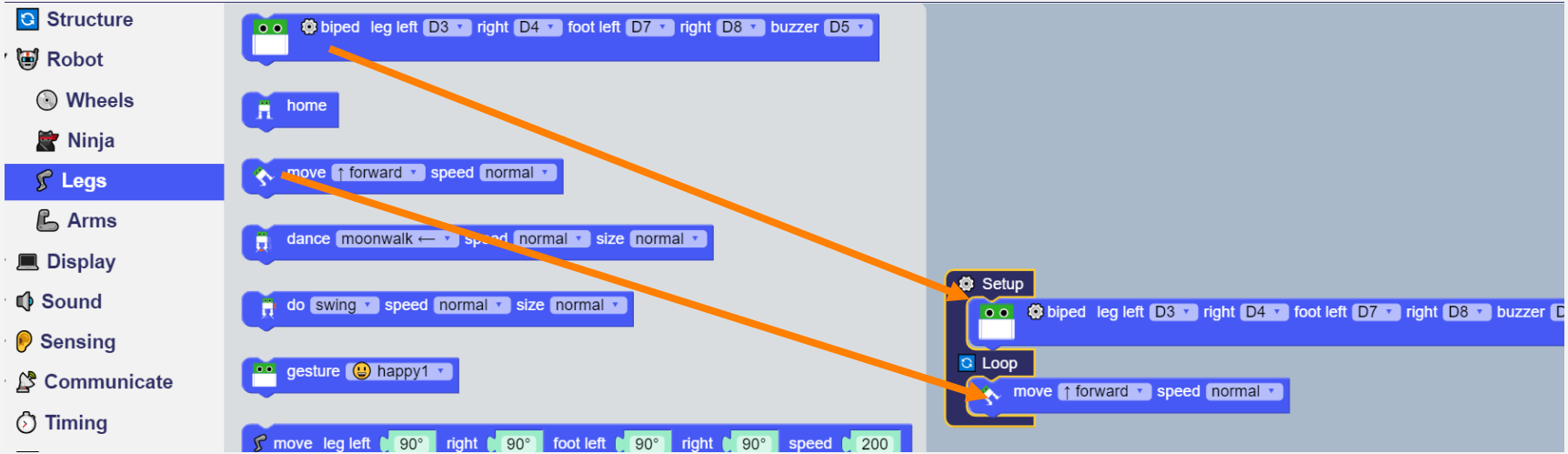


Setup / Loop

Важно всегда использовать этот блок. Сначала перейдите в раздел "Structure" на левой боковой панели. Для начала выберите блок "Setup / Loop" и кликните на него. Блок "Loop" открывает и завершает последовательность функций, которая будет повторяться бесконечно. Если вы хотите, чтобы ваш код выполнялся только один раз, используйте блок "Setup". Этот блок также используется для инициализации компонентов и настройки пинов.

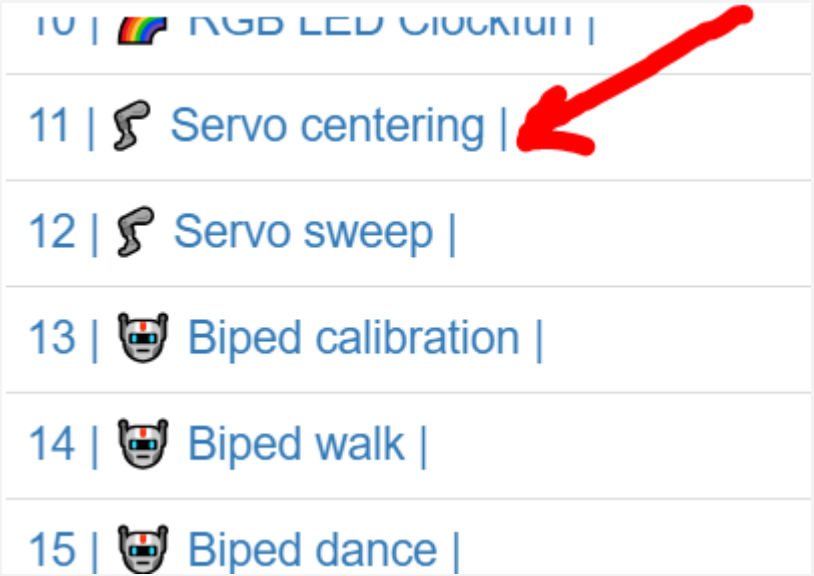
Перетаскивание блоков

Из панели инструментов выберите "**Otto**", перетащите блок конфигурации из списка в блок "**Setup**" и другие блоки в структуру "**Loop**". Пространство внутри **Setup** и **Loop** англичане называют Брекеты.

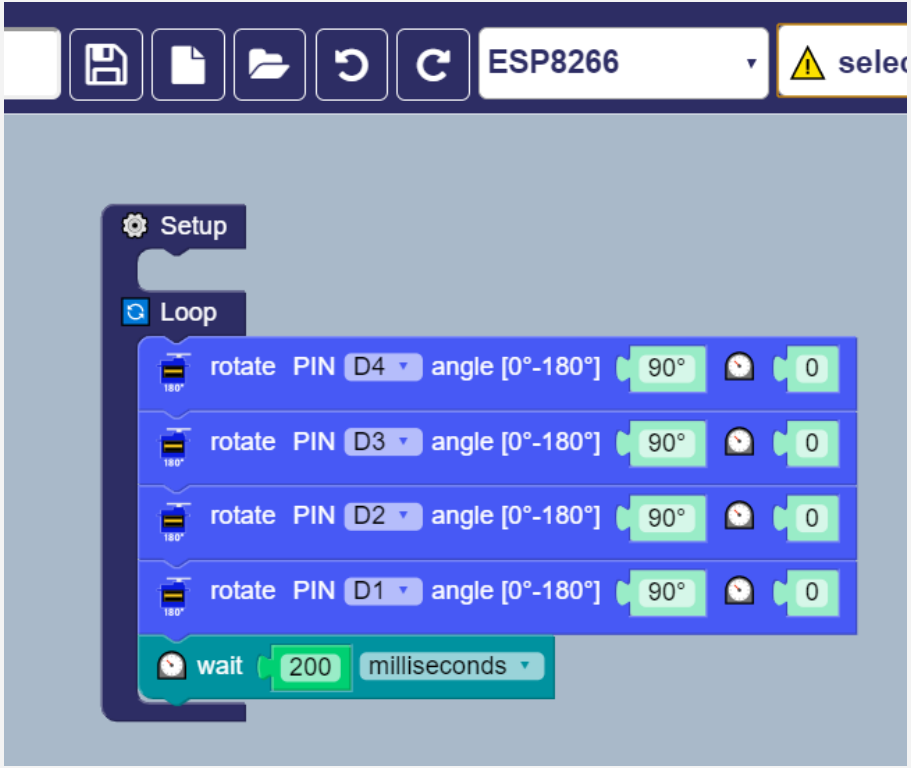


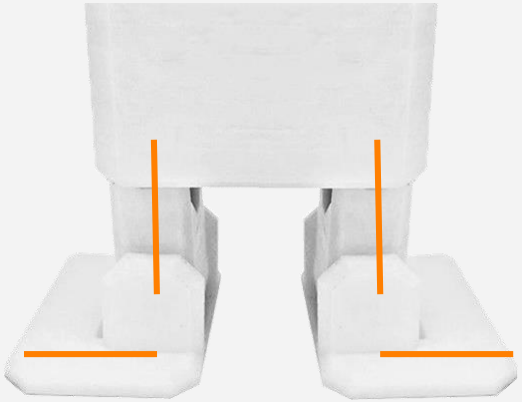
Servo centering example

- Центровка Сервоприводов



Upload code – Загрузите на робота этот скетч

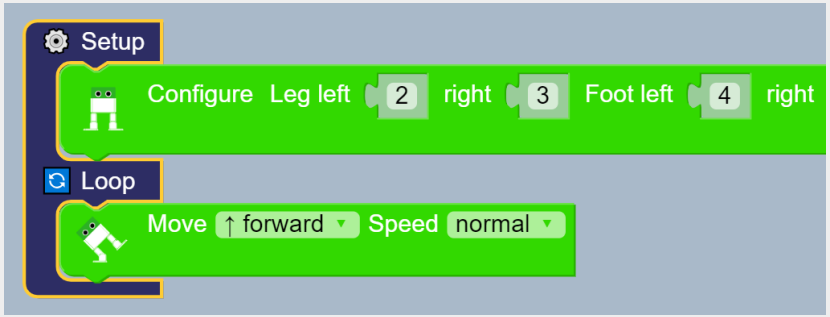




После загрузки кода ноги робота должны быть выровнены и находиться в центре, насколько это возможно. Если это не так, вам, возможно, придется пересобрать сервоприводы под правильным углом.

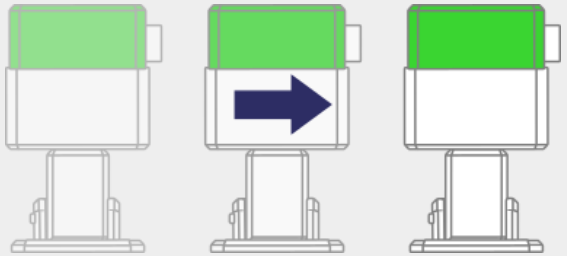
Ходьба перед бегом

Move – блок ходьбы

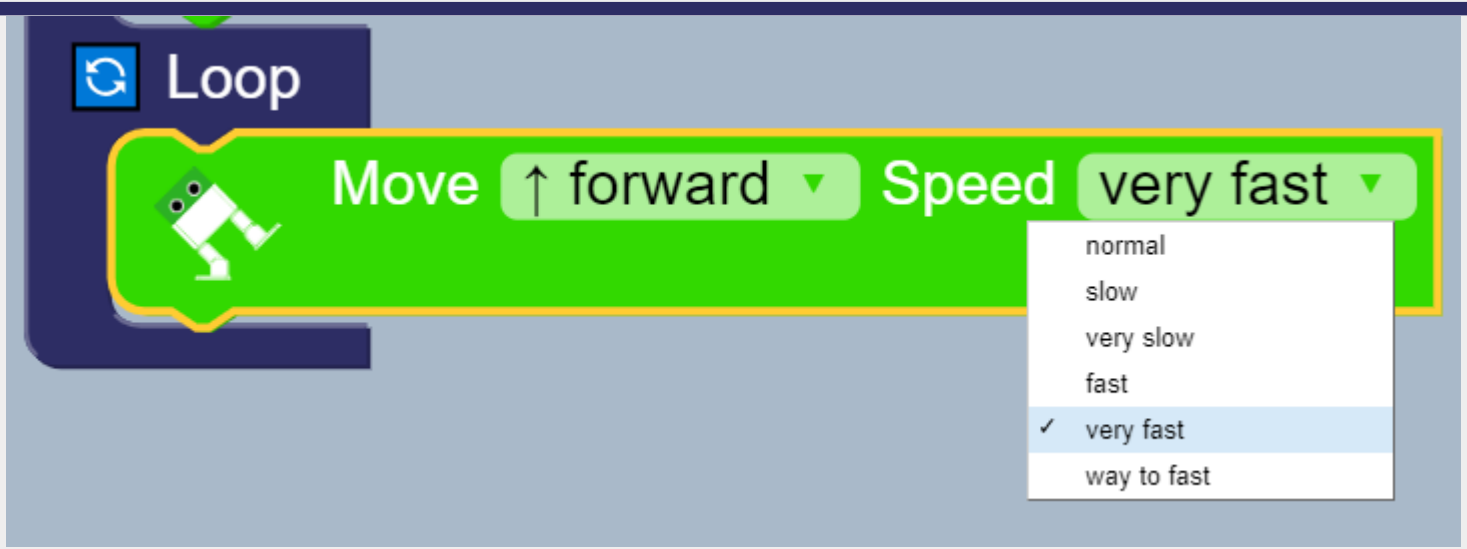


А в **C++** коде:

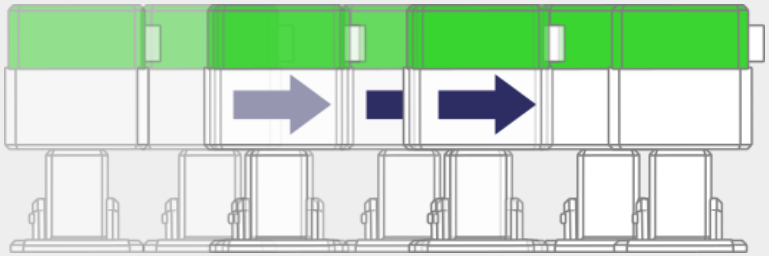
```
Otto.walk(1,1000,1);
```



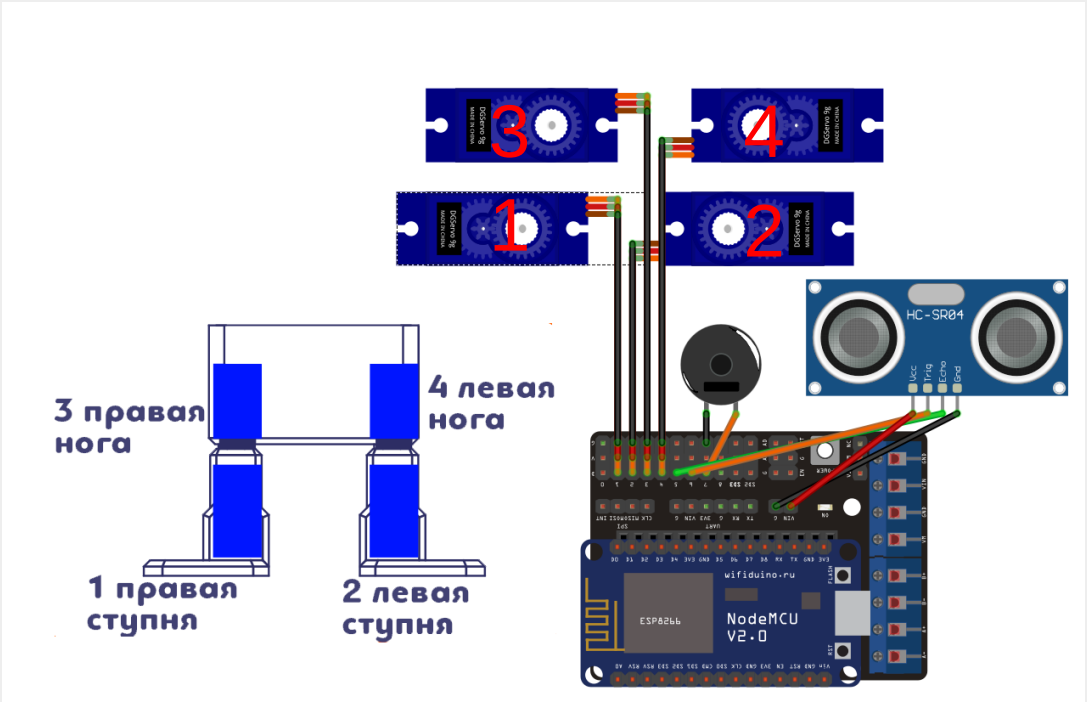
Сможете ли вы заставить Отто бежать?



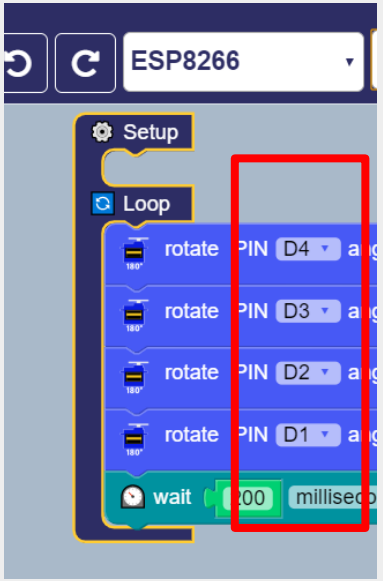
Otto.walk(1,500,1);



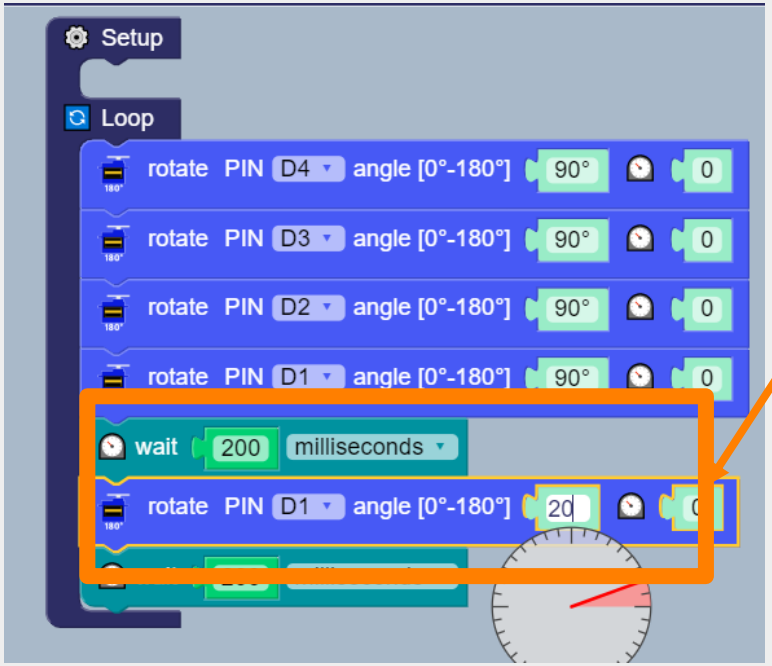
Подключение сервоприводов



Пример в Blockly для управления каждым двигателем



Добавьте ещё один блок **Servo** и **Time delay**



Нажмите дублировать(dublicate) на блоке вращение (rotate)

1. **С помощью этого блока пошевелите левой ступнёй**
2. **Затем покрутите правой ступнёй**
3. **Затем вращайте левой и правой ногами**

На этом этапе вы должны понимать:

- **какой пин на моторшилде отвечает за какую часть какой ноги**
- **Как работает цикл и задержки в программе**