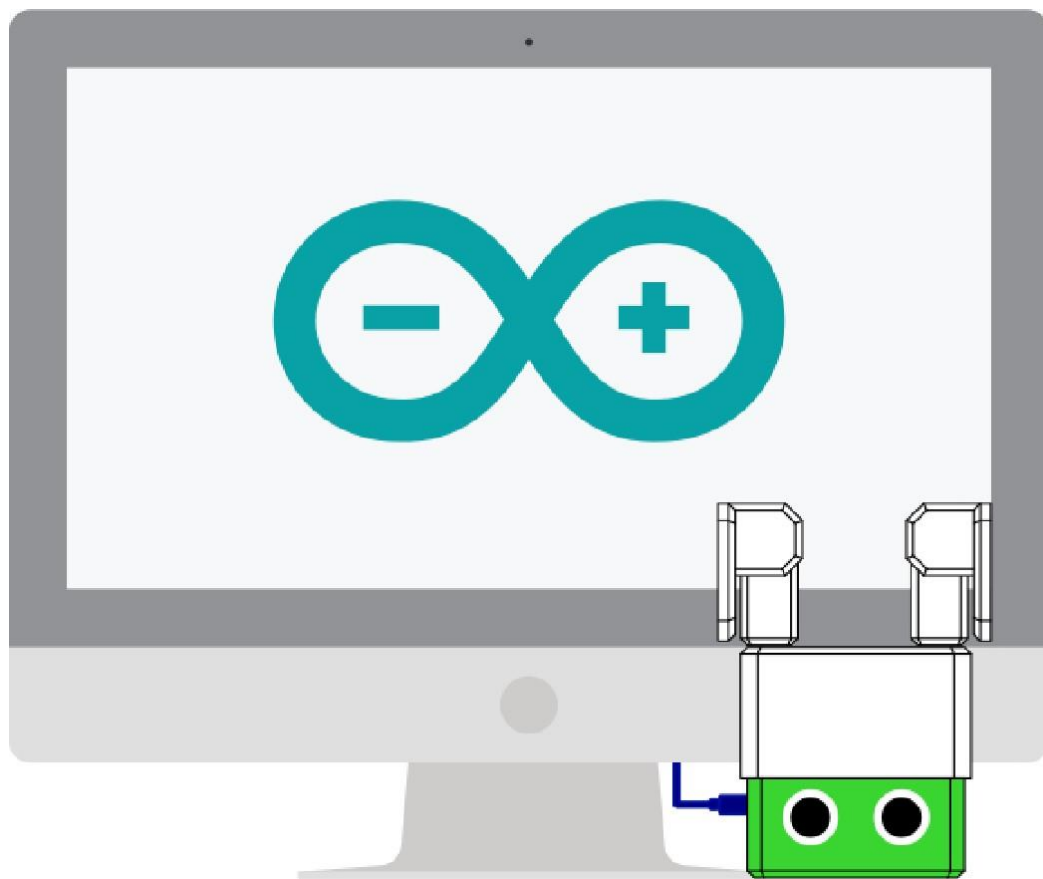




coding guide



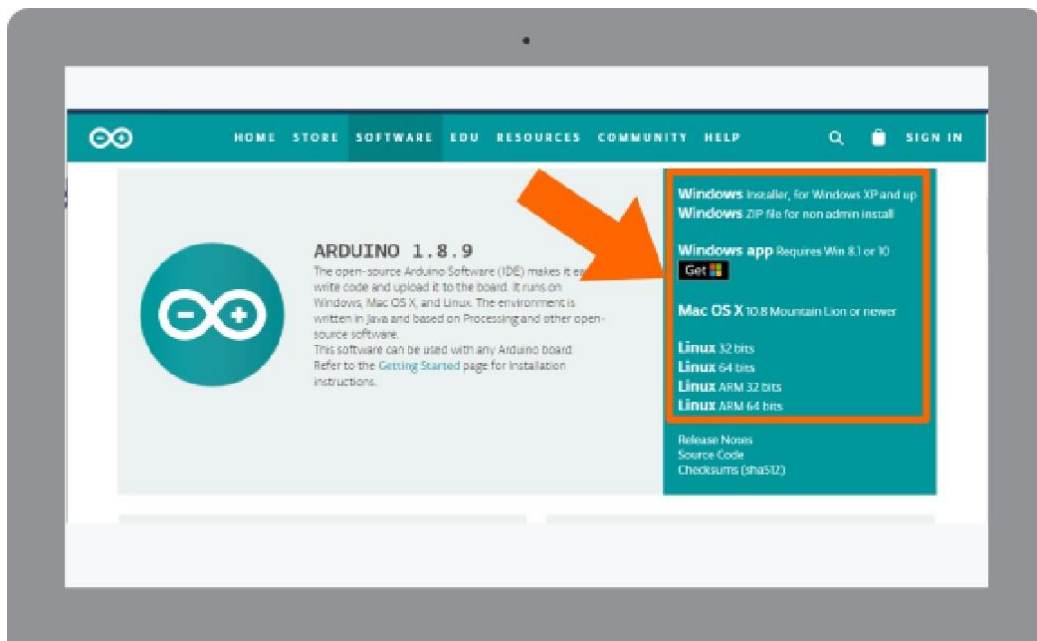
[L
i
n
k](#)

@ ottodiy.ru



1

- Перейдите на - <https://www.arduino.cc/>



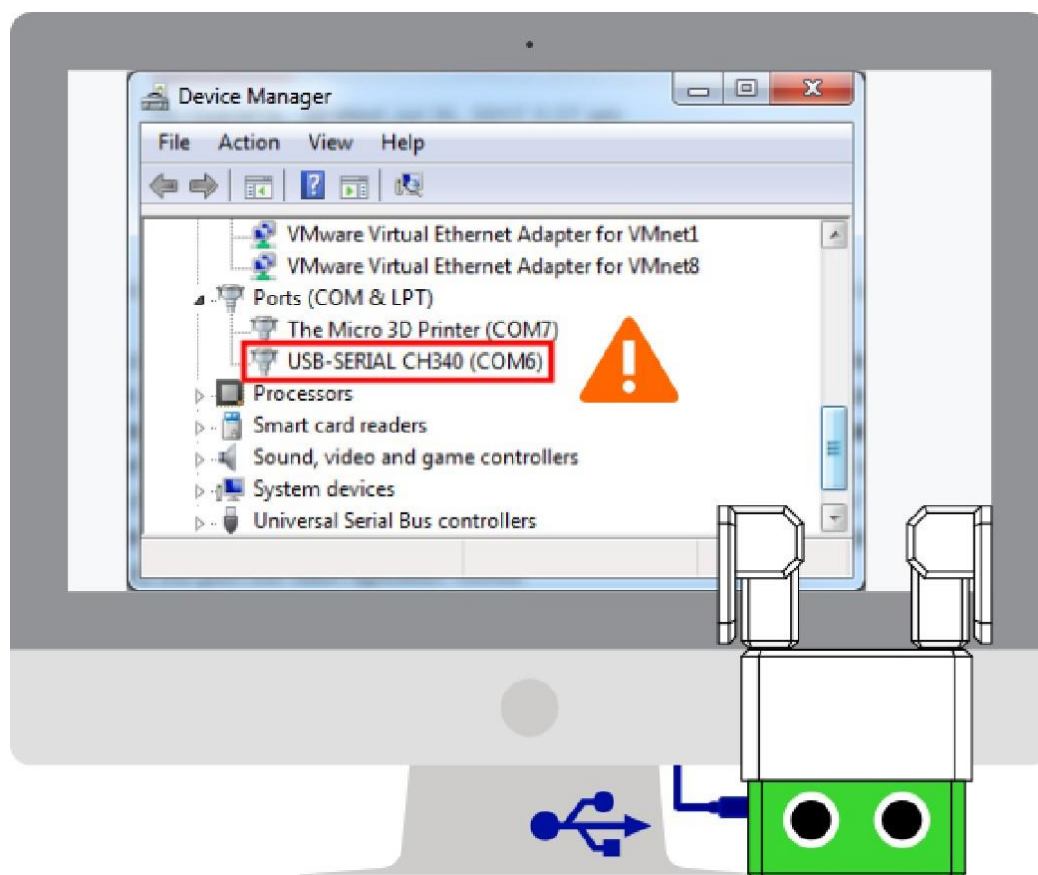
- Скачайте & установите Arduino IDE



2

- Перейдите по ссылке

<https://sparks.gogo.co.nz/ch340.html>

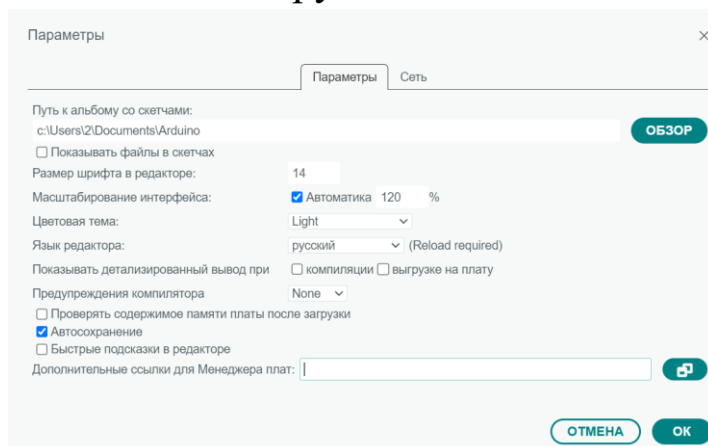




3

Зайдите в меню Файл->Параметры (File->Preferences)

- поставьте русский язык



- Впишите следующую строчку:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

в поле - **Дополнительные ссылки для Менеджера плат:**

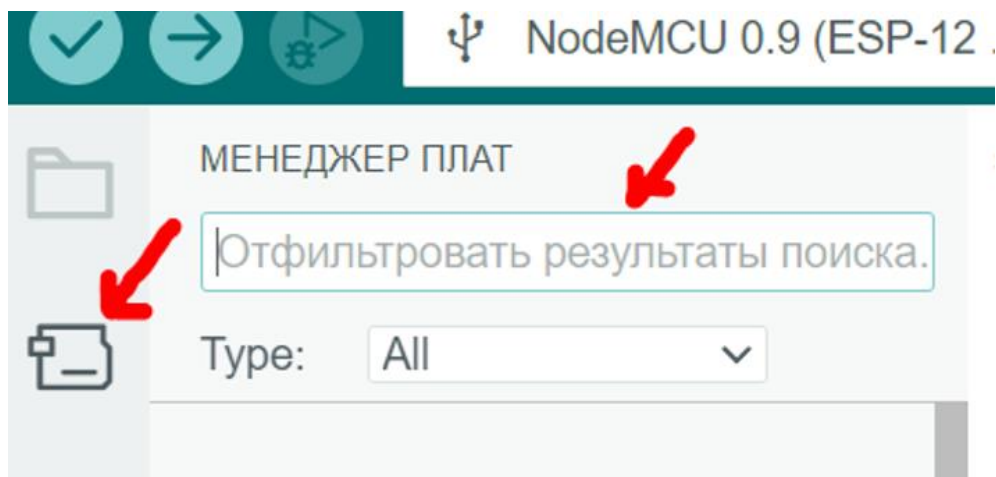


- Нажмите Ок – и Arduino IDE начнёт загрузку информации для нашей платы - podemcu.



4

Зайдите в Менеджер Плат (слева есть значок похожий на микроконтроллер)



- Поставьте курсор в поле Отфильтровать..
- И напишите слово – node
- Найдите в списке ESP8266



5

Файл Изменить Скetch Инструменты Справка

NodeMCU 0.9 (ESP-12 ...)

МЕНЕДЖЕР ПЛАТ

node

Type: All

esp8266 от **ESP8266** Community

Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, Lively Agrumino...

[Дополнительная информация](#)

3.1.2 **УСТАНОВКА**

```
1 void setup() {
2     // put your setup
3
4 }
5
6 void loop() {
7     // put your main c
8
9 }
10
```

Вывод

Нажмите Установка. Дождитесь результата как на картинке - ESP8266 установилась!

Type: All

esp8266 от ESP8266 Community

3.1.2 installed

Boards included in this package:
Arduino, ESPino (ESP-12 Module), ESPresso Lite 2.0, ITEAD Sonoff,...

[Дополнительная информация](#)

3.1.2 **УДАЛИТЬ**

Вывод

```
Configuring tool.
esp8266:xtensa-lx106-elf-gcc@3.1.0-gcc10.3-e5f9fec устан
Installing esp8266:mkspliffs@3.1.0-gcc10.3-e5f9fec
Configuring tool.
esp8266:mkspliffs@3.1.0-gcc10.3-e5f9fec установлен
Installing esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec
Configuring tool.
esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec установлен
Installing esp8266:python3@3.7.2-post1
Configuring tool.
esp8266:python3@3.7.2-post1 установлен
Installing platform esp8266:esp8266@3.1.2
Configuring platform.
Platform esp8266:esp8266@3.1.2 installed
```



6

Перейдите в Управление библиотеками:

Файл Изменить Скетч Инструменты Справка

УПРАВЛЕНИЕ БИБЛИОТЕКАМИ

ottodiy

Тип: All

Тема: All

OttoDIYLib от Otto DIY,
Camilo Parra Palacio

Otto DIY official Arduino Libraries.
OttoDIYLib contains all the main Otto
robot libraries to program Arduino...

[Дополнительная информация](#)

13.0.0

УСТАНОВКА

sketch_nov1a.ino

```
1 void setup(  
Вывод
```

```
Configuring tool.  
esp8266:xtensa-lx  
Installing esp826  
Configuring tool.  
esp8266:mkspiffs@  
Installing esp826  
Configuring tool.  
esp8266:mklittlef  
Installing esp826  
Configuring tool.
```

- Введите в поле поиска – ottodiy
- Жмём – УСТАНОВКА

Вывод

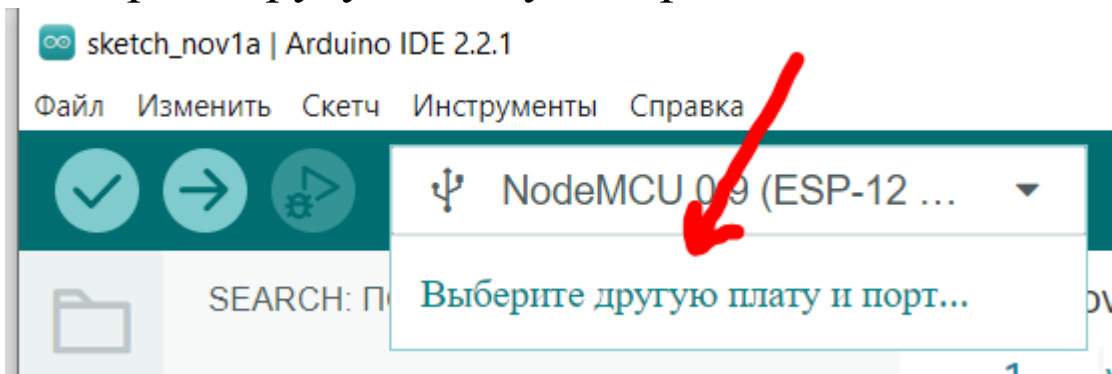
```
Downloading OttoDIYLib@13.0.0  
OttoDIYLib@13.0.0  
Installing OttoDIYLib@13.0.0  
Installed OttoDIYLib@13.0.0
```



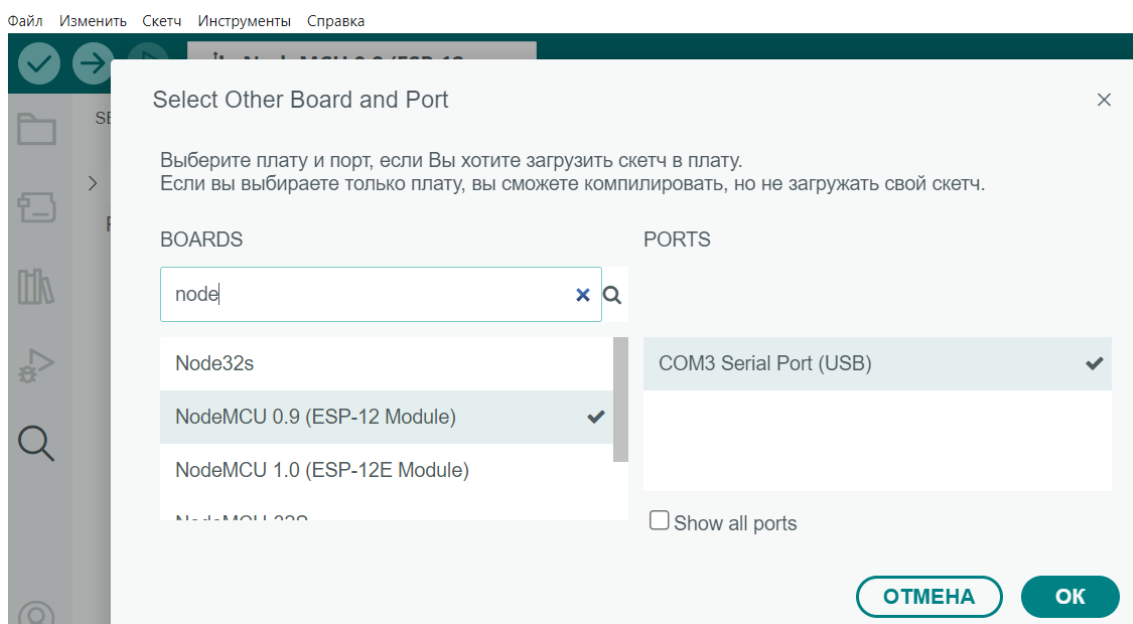
7

Выбор микроконтроллера

Щёлкаем по значку USB и чуть ниже появится – “Выбрать другую плату и порт” – нажимаем



В открывшемся окне, в строке поиска набираем node, выбираем как на картинке:

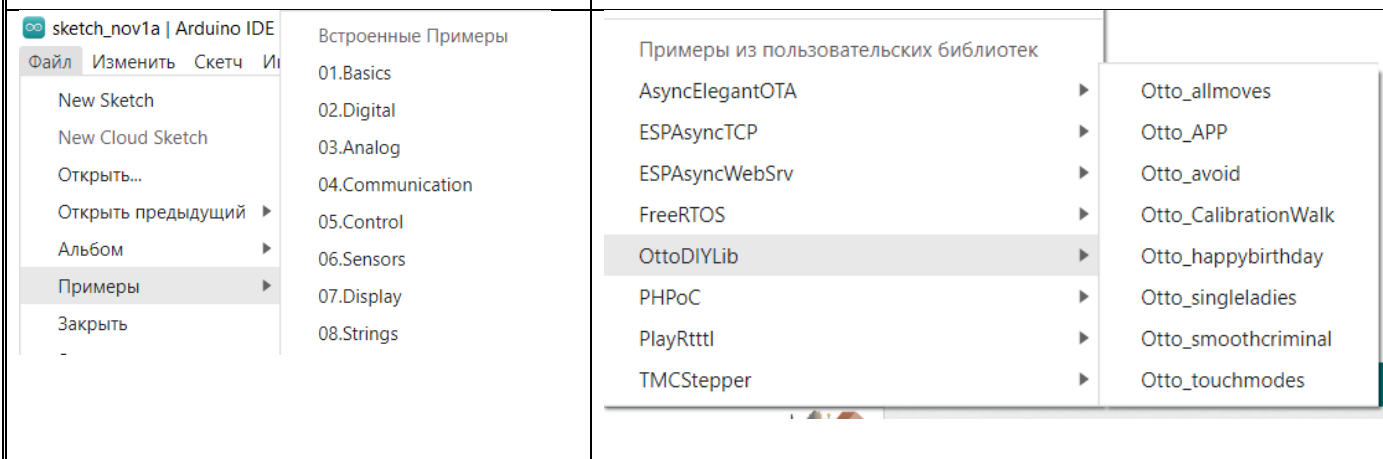


Как открыть примеры Отто? Есть 3 варианта:

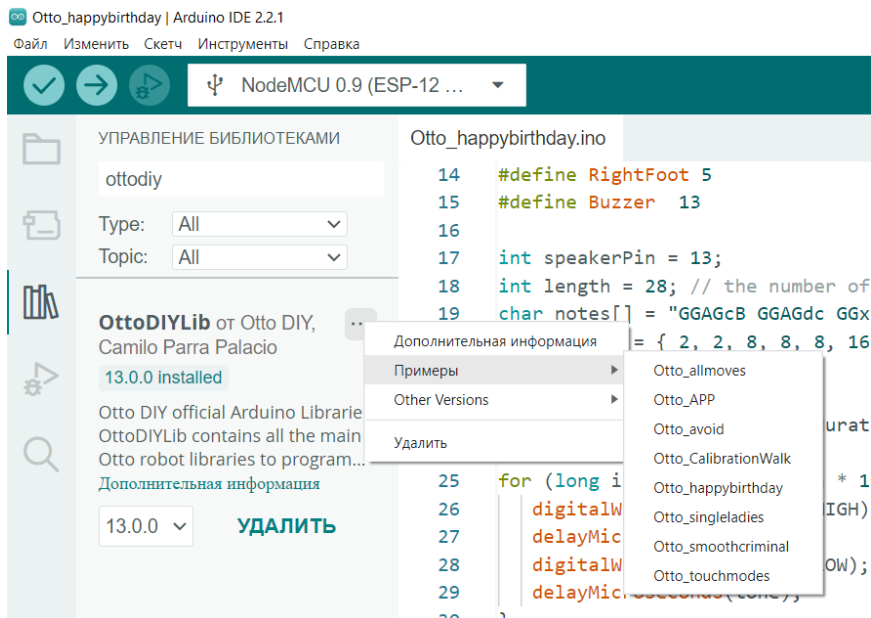
1. Через меню файл

Переходим в примеры

Находим OttoDIYLib и выбираем Otto_avoid



2. Через три точки в поиске библиотек



3. Открываем в файловой системе, по адресу
C:\Users\2\Documents\Arduino\libraries\OttoDIYLib\examples
где вместо цифры 2 – имя вашего пользователя

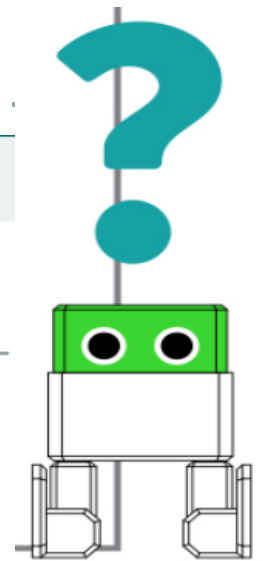
ottodiy.ru

Открываем - Otto_avoid.ino

// ---Проверяем расстановку цифр---

Otto_avoid.ino

```
7  #include <Otto.h>
8  Otto Otto; //This is Otto!
9  //-----
10 #define LeftLeg 2
11 #define RightLeg 0
12 #define LeftFoot 4
13 #define RightFoot 5
14 #define Buzzer 13
15 #define Trigger 12 // ultrasonic sensor trigger pin
16 #define Echo 14 // ultrasonic sensor echo pin
--
```



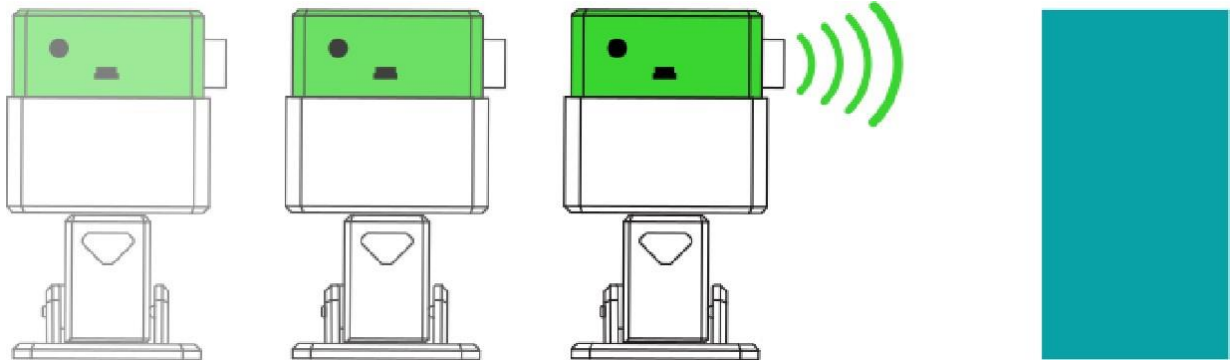
Код основного цикла выглядит следующим образом:

```
void loop() {
    if (ultrasound() <= 15) {
        Otto.sing(S_surprise);
        Otto.playGesture(OttoConfused);
        Otto.walk(2,1000,-1); // BACKWARD x2
        Otto.turn(3,1000,1); // LEFT x3
    }
    Otto.walk(1,1000,1); // FORWARD x1
}
```



8

Отто будет идти бесконечно, пока не обнаружит препятствия, которых нужно избегать



`Otto.sing(songName)`

В скобках `songName` – меняем на любые звуки из этой таблички:

<code>S_disconnection</code> (Отключение)	<code>S_mode2</code> (Режим 2)	<code>S_sad</code> (Грусть)
<code>S_buttonPushed</code> (Нажата кнопка)	<code>S_mode3</code> (Режим 3)	<code>S_confused</code> (Смущение)
<code>S_surprise</code> (Удивление)	<code>S_OhOoh</code> (Ой-ой)	<code>S_fart1</code> (Пук 1)
<code>S_OhOoh2</code> (Ой-ой 2)	<code>S_cuddly</code> (Ласково)	<code>S_fart2</code> (Газы2)



9

S_sleeping (Сон)

S_happy (Счастье)

S_superHappy
(Очень Рад)

Otto.sing(S_surprise);

↑
sing function ("звук для воспроизведения ")

Код программы:

```
#include <Otto.h>
Otto Otto;
const int buzzer = 13;
void setup() {
    Otto.init(0, 0, 0, 0, true, buzzer);
}

void loop() {
    delay(5*1000);
    Otto.sing(S_superHappy);
    delay(5*1000);
    Otto.sing(S_confused);
    delay(5*1000);
    Otto.sing(S_happy_short);
    delay(5*1000);
    Otto.sing(S_sad);
    delay(5*1000);
}
```

Вы можете создать новый звук для Otto с помощью функции `_tone`:

```
Otto._tone(10, 3, 1);
```

(`noteFrequency`, `noteDuration`, `silentDuration`)

У нас есть три параметра:

1. Первый - `noteFrequency` (10) представляет собой ноту, которую мы хотим сыграть.
2. Второй параметр `noteDuration` (3) указывает, как долго мы будем удерживать эту ноту.
3. Третий - `silentDuration` (1) определяет, как долго мы будем молчать после того, как сыграли ноту. Здесь 1 - это короткая пауза.

Otto.walk(2,1300,-1);

move function ("#steps, Time[ms], direction")

Функция "Walk" (Ходьба) - **Otto.walk**(steps, time, dir);

Здесь "**steps**" (шаги) указывает, сколько раз вы хотите повторить это движение.

"**Time**" (обозначается как T ниже) в миллисекундах определяет длительность движения. Более высокое значение "time" означает медленное движение, попробуйте значения между 500 и 3000 миллисекунд.

"**Dir**" (направление) - это направление: 1 для движения вперед и -1 для движения назад.

Пример: **Otto.walk**(2, 1000, 1);

В этом примере 2 - это количество шагов, 1000 - "TIME" в миллисекундах, и робот будет двигаться вперед.

"**h**" (высота) - размер движений, которые связаны с танцами

Таблица по другим функциям Отто с обозначениями и допустимыми диапазонами:

Функция	Описание	Диапазон steps	Диапазон T	Диапазон h	dir значения
Otto.turn (steps, T, dir);	Вращение	1-50	500-5000	-	1 или -1
Otto.bend (steps, T, dir);	Наклон	1-50	500-5000	-	1 или -1
Otto.shakeLeg (steps, T, dir);	Тряска ногой	1-50	500-5000	-	1 или -1
Otto.jump (steps, T);	Прыжок	1-50	500-5000	-	-
Otto.moonwalker (steps, T, h, dir);	Ходьба "лунным ходом"	1-50	500-5000	15-40	1 или -1
Otto.crusaito (steps, T, h, dir);	Танец Рыцарь	1-50	500-5000	20-50	1 или -1
Otto.flapping (steps, T, h, dir);	"Всплеск"	1-50	500-5000	10-30	1 или -1
Otto.swing (steps, T, h);	Качание	1-50	500-5000	0-50	-
Otto.tiptoeSwing (steps, T, h);	Качание на цыпочках	1-50	500-5000	0-50	-
Otto.jitter (steps, T, h);	Дрожание	1-50	500-5000	5-25	-
Otto.updown (steps, T, h);	Поднятие и опускание	1-50	500-5000	0-90	-
Otto.ascendingTurn (steps, T, h);	Плавный поворот вверх	1-50	500-5000	5-15	-

Otto.turn(steps, T, dir); (количество шагов, T в миллисекундах, 1 - поворот влево или -1 - поворот вправо)

Otto.bend(steps, T, dir); (количество шагов, T в миллисекундах, 1 - наклон влево или -1 - наклон вправо)

Otto.shakeLeg(steps, T, dir); (количество шагов, T в миллисекундах, 1 - трясется влево или -1 - трясется вправо)

Otto.jump(steps, T); (количество шагов вверх, T в миллисекундах) в этой функции отсутствует параметр "dir" (направление), поскольку Otto на самом деле не прыгает! ;Р

Танцы:

Аналогично движениям, но более весело! Вы можете регулировать новый параметр "h" (высота или размер движений), чтобы сделать танец более интересным.

Otto.moonwalker(steps, T, h, dir);

(количество шагов, T в миллисекундах, h – высота или размер шага, 1 - влево или -1 - вправо) **h**: вы можете попробовать изменить в диапазоне от 15 до 40

Пример: **Otto.moonwalker**(3, 1000, 25, 1);

Otto.crusaito(steps, T, h, dir); (dir: 1 - влево или -1 - вправо)

h: вы можете попробовать изменить в диапазоне от 20 до 50

Otto.flapping(steps, T, h, dir); (dir: 1 - вперед или -1 - назад)

h: вы можете попробовать изменить в диапазоне от 10 до 30

Otto.swing(steps, T, h); **Otto.tiptoeSwing**(steps, T, h);

h: от 0 до 50

Otto.jitter(steps, T, h); h: от 5 до 25

Otto.updown(steps, T, h); h: от 0 до 90

Otto.ascendingTurn(steps, T, h); h: от 5 до 15.



10

Otto.playGesture(OttoFretful);

playGesture function ("эмоции для выражения ")

Цель функции playGesture - выразить эмоции, сочетая звуки с движениями одновременно. Если у вас есть светодиодная матрица, вы также можете отображать эмоции в работе через его "рот"!

```
Otto.playGesture(gesture);
```

Эмоция	Описание	Пример команды	Перевод на русский
OttoHappy	Эмоция счастья	Otto.playGesture(OttoHappy);	Радость
OttoSuperHappy	Эмоция восторга	Otto.playGesture(OttoSuperHappy);	Восторг
OttoSad	Эмоция грусти	Otto.playGesture(OttoSad);	Грусть
OttoVictory	Эмоция победы	Otto.playGesture(OttoVictory);	Победа
OttoAngry	Эмоция злости	Otto.playGesture(OttoAngry);	Злость
OttoSleeping	Эмоция сна	Otto.playGesture(OttoSleeping);	Сон
OttoFretful	Эмоция беспокойства	Otto.playGesture(OttoFretful);	Беспокойство
OttoLove	Эмоция любви	Otto.playGesture(OttoLove);	Любовь
OttoConfused	Эмоция смущения	Otto.playGesture(OttoConfused);	Смущение
OttoFart	Эмоция "пук"	Otto.playGesture(OttoFart);	Пук
OttoWave	Эмоция прощания ("пока")	Otto.playGesture(OttoWave);	Прощание
OttoMagic	Эмоция "волшебства"	Otto.playGesture(OttoMagic);	Волшебство
OttoFail	Эмоция неудачи	Otto.playGesture(OttoFail);	Неудача