Blockly guide
code your own robot

- familiarize with the block programming environment.
- First simple projects, like programming a dance with Otto.
- learn the concept of sequential thinking and conditionals.
- make a complicated project, Otto must carry out a determined action.

install the software with the examples in: https://www.ottodiy.com/#blockly
```c
#include <US.h>
#include <Otto0.h>

Otto9 Otto;

int distance;
bool obstacleDetected = false;
#define PIN_Trigger 8 //ultrasound
#define PIN_Echo 9 //ultrasound
#define PIN_VL 2 //left leg, servo[0]
#define PIN_VR 3 //right leg, servo[1]
#define PIN_RL 4 //left foot, servo[2]
#define PIN_RR 5 //right foot, servo[3]
#define PIN_Trigger 8 //ultrasound
#define PIN_Echo 9 //ultrasound
#define PIN_Buzzer 13 //buzzer

void setup()
{
  Otto.init(PIN_VL, PIN_VR, PIN_RL, PIN_RR, true, A6,
}

void loop()
{
  if (Otto.getDistance() < 15) {
    Otto.playGesture(OttoConfused);
    for (int count=0; count<3; count++) {
      Otto.turn(3,1800,1); // RIGHT
    }
    Otto.walk(3,1000,1); // FORWARD
  }
}```
<table>
<thead>
<tr>
<th>Example</th>
<th>Level</th>
<th>Robot: Otto DIY</th>
<th>OPEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>★★★★☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legs calibration</td>
<td>★★★☆☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blink LED on board</td>
<td>★★★★☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buzzer beeps</td>
<td>★★★★☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dance</td>
<td>★★★☆☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid obstacles</td>
<td>★★★☆☆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servo control</td>
<td>★★★☆☆</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
you did it!
now let’s learn the basics
Check the settings according to level and language*

*after click on Accept Press F5 to reload.
Structure: Start or stop or create a loop in the program.
Time: used to delimit the program over time.
**Otto and Humanoid specific movements, sounds and gestures.**
Logic: create conditionals, repetitions and logic programming.
Math: make calculations or to insert numbers.
Function: create a procedure that can be repeated.
Variable: A variable is a named value that can be changed.

Sensor: interact with the environment with multiple sensors.
Motor: move a servo motor or activate a regular DC motor.
Display: turn on or off multiple types of LEDs or matrix or screen.
Audio: emit a sound (with an mp3 player or a buzzer-piezo).

Communication: with certain modules. Bluetooth or Serial.
Text: insert text into the program.
parts of a robot

- difference between sensors and actuators. (inputs vs outputs)
- where is the brain of the robot?
- what is a servomotor?
- can Otto talk?
- how does Otto see?
- other components & interactions.
every robot has basically 3 component groups:

- **sensors (INPUTS)** able to interpret information.
- **processor system** a “mini computer”
- **actuators (OUTPUTS)** produce the effect programmed.
we could say that the sensors are the robot’s senses, these transmit information to the processor that allows to alter the function performed by actuators.

in addition a robot will need a power source to function and a physical structure to support the elements that compose it and perform its functions.
how does a robot think?

- think (process) one thing at a time, he can't do two things at the same time.
- think and repeat actions very quickly, so much that sometimes we can't even see what he is doing and you have to tell him to wait a bit.
- he never does anything you haven't told him, you have to program everything, whatever you want he to do and give him the orders one at a time.
- normally, when a robot finishes its list of orders, it returns to start over, repeating your programming over and over again. (Loop)
what is a servomotor?
is an **actuator** that can rotate (usually between 0° and 180°). It is used to control the angular position, at neutral can rotate 90° to the left and 90° to the right. (forced beyond this and they will break)

Otto has 4 servo motors that collectively help the robot walk and dance.
servomotor includes 3 pins:

**VCC power pin** (typically red) connects to V (+5V)

**GND ground pin** (typically brown or black) connects to G (0V)

**PWM signal pin** (typically orange or white) receives the control signal, connects to the S of a determined pin number.
it comes with 3 screws:

2 for mounting (long & pointy)

1 for center horn short & flat)

and

3 horns/ arms “white keys”
we only need 1 per servo.
time to code again!
servos wiring

example servo control
(align center)
walk before run
2 walk examples

Otto.walk(1,1000,1);

what is the difference?
can you make Otto run?
dance time!
<table>
<thead>
<tr>
<th>Function</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otto.walk()</td>
<td>(2, 1000, 1)</td>
</tr>
<tr>
<td>Otto.tiptoeSwing()</td>
<td>(2, 1000, 25)</td>
</tr>
<tr>
<td>Otto.moonwalker()</td>
<td>(5, 1000, 40, 1)</td>
</tr>
<tr>
<td>Otto.moonwalker()</td>
<td>(5, 1000, 40, -1)</td>
</tr>
<tr>
<td>Otto.crusaito()</td>
<td>(5, 1000, 40, 1)</td>
</tr>
</tbody>
</table>
can Otto talk?
piezo buzzer is an actuator used to generate sound, beep or even make the melody of a song.

Otto can’t talk but he can create sounds related to his emotions similar to R2D2.

make sure to identify the positive mark and connect in the right pin
Otto.sing(S_superHappy);
what is a gesture?
movements + sounds to generate emotions or multiple moods.

```
Otto.playGesture(OttoLove);
Otto.playGesture(OttoSuperHappy);
Otto.playGesture(OttoSad);
Otto.playGesture(OttoSleeping);
Otto.playGesture(OttoConfused);
Otto.playGesture(OttoAngry);
Otto.playGesture(OttoMagic);
Otto.playGesture(OttoWave);
Otto.playGesture(OttoVictory);
Otto.playGesture(OttoFail);
Otto.playGesture(OttoFart);
```
how does Otto “see”?
ultrasonic sensor

is used to measure the distance to an object by using ultrasonic waves, includes 4 pins:

VCC (Power), GND (Ground)
TRIG receives the control signal
ECHO sends a signal (pulse)

by measuring the duration of pulse we can calculate the distance.
Otto can avoid obstacles!
what will you code?
other components & interactions

touch & sound sensors and LED matrix displays. (depends on the robot kit)


but for better performance and care of the environment, use rechargeable batteries. 
more detail info here: https://www.ottodiy.com/blog/power
troubleshooting & debugging

find and fix, perseverance is important.

not finding Otto connected in your PC?
install the CH340 driver to recognize USB device.

can not upload code?
check USB cable and that Bluetooth is disconnected

does Otto reset every now and then?
that is because of lack of power, discharged batteries.

are Otto legs and feet twisted?
check that you centered your servos before assembly for precise movements calibration is needed read this blog article: https://www.ottodiy.com/blog/calibration

ask for help in the community forum : https://wikifactory.com/+OttoDIY/forum
more robot projects you can make:

https://wikifactory.com/+OttoDIY/projects
join us!

become a part of this global robot movement a very friendly, creative and collaborative community.

BECOME AN OTTO BUILDER TODAY!

join us here: ottodiy.com/#join-us
create yours here: https://www.ottodiy.com/blog/clubs
1. build your own robot
2. code your own robot
3. learn to code
4. learn to really code
5. create your own dance for Otto
6. what is inside Otto?
7. play with sensors
8. design your own robot
9. make your own accessories
10. document your new Otto REMIX

ottoman.com
reach to us for more content