

Programa tu propio robot

- Configuración e instalación del software
- Familiarización con el entorno de programación con bloques
- Primeros proyectos simples, como: programar un baile con Otto.
- Aprende los conceptos de pensamiento secuencial y condicionales.
- Haz un proyecto más complicado, interactúa con sensores, usa programación avanzada.

¿Leíste esta guía?



Cargar un programa, existen 3 formas...



Aplicación Independiente Otto Blockly: Instalación en el PC, funciona sin conexión a internet, fácil de usar, programación & carga por USB (No necesita IDE de Arduino))





Aplicación Web en línea, desde un navegador pero es necesario instala una extensión o Arduino





IDE de Arduino: Trabaja sin conexión a internet, compatible con todos los sistemas operativos, es una programación más avanzada





Otto Blockly Aplicación Independiente Windows

- Puede ser **instalada localmente** y por lo tanto funciona sin internet.
- Puede cargar un código a través de USB.
- Muchos bloques para controlar todo tipo de placas Arduino, sensores, luces RGB, pantallas, audio, cámaras, motores y más.
- Puede mostrar una vista previa y editar el código en texto.
- No es necesario instalar librerías.
- Tiene también una <u>versión en línea</u>.



- <u>Necesita conexión a internet</u>
- No puede cargar un programa pero puede interactuar con el robot Otto via BLE (Bluetooth) de forma instantánea.
- Control funciones básicas con AI.
- Simulador de escenarios para juegos.
- Solo necesita pre-cargar un código, una sola vez para la programación en vivo.

Ir a la página

Descargar Otto Blockly clic aquí (

*algunas veces windows defender o un antivirus puede decirte que el software no es seguro, procede de todas formas a guardarlo y abrirlo.



Instalar



Instalar el Driver USB

Algunos ordenadores **reconocerán a Otto como un dispositivo USB automáticamente cuando lo conectes,** pero otros no.

Si tu ordenador no reconoce el dispositivo USB necesitas **descargar e instalar el driver USB**:

USB serie CH340, descargalo aquí

<u>Ésta guía muestra el paso a paso para todos los</u> <u>sistemas operativos</u> o

FT232 USB UART descargalo aquí

<u>Esta guía muestra el paso a paso</u>



Interfaz de usuario de Otto Blockly



Prueba de inicio rápido



Hola Constructor de Otto Estoy de vuelta, listo para ayudarte

Ctto Blockly				
Level: 1 2 3 GB English 🔻 🔁 🖿 🖻 💿				
Structure				
V 🗑 Otto				
■Examples				
😔 Level 1 🗊 Servo centering Otto DIY Starter				
💮 Level 1 📢 Buzzer Otto DIY Starter				
😔 Level 1 📫 Melody Otto DIY Starter				
😔 Level 1 👹 Walk Otto DIY Starter				
👴 Level 1 👹 Legs calibration Otto DIY Starter				
🛞 Level 1 🗑 Dance Otto DIY Starter				
😔 Level 1 👹 Avoid obstacles Otto DIY Starter				

Abramos un ejemplo haciendo click en el botón del sombrero.

2 Selecciona el ejemplo "Dance"



Si eres aventurero escoge un ejemplo diferente como "Walk"

📕 Arduino Nano 🔻 🛛 🔫	A Select USB port
Setup	Select USB port COM7
Configure Leg left 2 right	3 Foot left (4
 Loop repeat time 	
Move forward Speed in	ormal 🔹
Move [] backward Speed	normal 🔹

Conecta tu robot y asegúrate que el puerto COM# esté seleccionado en el menú desplegable de puerto USB

*Si no aparece nada, entonces <u>el driver USB no</u> <u>fue correctamente</u> <u>instalado en tu ordenador.</u> Debes reiniciar tu ordenador e intentarlo nuevamente.





Error: Command failed: arduino-cli.exe upload --port COM7 --fgbn arduino:avr:nano sketch/sketch.ino avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x55 avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 2 of 10: not in sync: resp=0x55 avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 3 of 10: not in sync: resp=0x55 avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 4 of 10: not in sync: resp=0x55 avrdude: stk500_recv(): programmer is not responding avrdude: stk500_getsync() attempt 5 of 10: not in sync: resp=0x55 avrdude: stk500_recv(): programmer is not

responding avrdude: stk500_getsyncl resp=0x55 avrdude: stk500_recv(): avrdude: stk500_getsync() attempt avrdude: stk500_recv(): programme stk500_getsync() attempt 8 of 10: r stk500_recv(): programmer is not responding __vrdude: stk500_getsync() attempt 9 of 10: not in sync: resp=0+55 aurducker stk500_recv(): programmer is not responding avri 10: not in sync: resp=0x55 Error du error: exit status 1

Oops! no te preocupes, podemos arreglar eso

> tsync() attempt 10 of

loading error: uploading

ormal •

ormal •

Si aparece este error, eso significa que la placa Arduino Nano* no es correcta, entonces intenta seleccionando Arduino Nano (old bootloader) en su lugar y haz click en cargar nuevamente.

1	Arduino Nano	•
	C / C++	
	Arduino Nano	
7	Arduino Nano (old bootloader)	

*Las placas Nano vienen con un "bootloader" diferente, por lo tanto usan diferentes configuraciones para cargar un código, pero las dos funcionan exactamente igual en los robots Otto.

Ardui	no Nano (old bootloa	ider) 🔻 🛛 🕶 CC	ом7 🔹	ı 🔗 🔁
🗘 Setu	p			
Uploading	: 🗹 OK code uploade	d	×	00
1	Setup			
	Configure Le	g left 2 right	3 Foot left 4	right 5

Solamente lleva unos pocos segundos en cargar el código correctamente cuando la placa correcta está seleccionada.

Esto significa que tu código fue cargado a Otto! y tu robot debería estar bailando en este momento. ¡Si, lo hiciste, felicidades! hasta luego



Problemas comunes

¿Otto no parece estar conectado? instala el **driver USB CH340 or FTDI** para ver el puerto COM.

¿No puedes cargar el código? Placa equivocada seleccionada **(intenta Arduino Nano old bootloader o Arduino Nano) o revisa el cable USB**

¿Otto se reinicia cada cierto tiempo? Baja calidad de las baterías o malas conexiones.

¿Las piernas y pies de Otto están torcidos? Asegúrate que centraste y alineaste los servos al montarlo.



Aplicación web Otto Blockly (Alternativa para Mac)

\leftrightarrow \rightarrow $ extbf{C}$ https:/	/ottoschool.com/blockly/
Otto Blockly	
Level: 1 2 3 GB Englis	sh 🔺 🚱 🎓 🖹 🖻 🖺 🔯 🧿 😋 📱 Arduino Nano 🛛 👻 🚭
Structure	
🕨 🗑 Otto	
🔿 Time	
🔚 Input / Output	
Logic Control	
🔢 Math Operators	

Otto Blockly también es una herramienta web. <u>Abre este link en un navegador de</u> <u>Chrome</u>

Puedes usarlo desde cualquier sistema operativo incluyendo Linux y Chromebooks.

Pero primero necesitas hacer una configuración inicial siguiendo estos pasos

El dispositivo USB será reconocido por el navegador Chrome y puedes cargar tu código ¡directamente desde el navegador!



Usar el código de Arduino

Si tienes problemas configurando la carga desde la web, siempre puedes descargar el código o copiar/pegar y utilizarlo como código común de Arduino.

Haz click en el botón de guardar, eso descargará de manera rápida el archivo .ino

Esto es también una forma de **convertir bloques en código de texto.** Eso puedes guardarlo y compartirlo con otros proyectos de Arduino.

Abrir el código generado en Arduino

Puedes abrir directamente el archivo o abrir el IDE de Arduino y pegar el código allí. <u>(Sigue la guía de configuración para Arduino en este link)</u>

00	sketch_aug03a Arduino 1.8.13
🗸 💿 🛅 🛃 Verify	2
sketch_aug03a §	
<pre>#include <0tto9.h> 0tto9 0tto;</pre>	
<pre>#define PIN_YL 2 // left leg, servo[0] #define PIN_RR 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_YL, PIN_YR, PIN_RL, PIN_RR, true, A6, PIN_Buzzer, PIN_Trigger, } void loop() { Otto.walk(1,1000,1); // FORWARD Otto.moonwalker(1, 1000, 25, 1); }</pre>	PIN_Echo);
<pre>#define PIN_YL 2 // left leg, servo[0] #define PIN_RR 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_Echo 9 // ultrasound #define PIN_Buzzer 13 //buzzer void setup() { Otto.init(PIN_YL, PIN_YR, PIN_RL, PIN_RR, true, A6, PIN_Buzzer, PIN_Trigger, } void loop() { Otto.walk(1,1000,1); // FORWARD Otto.moonwalker(1, 1000, 25, 1); }</pre>	PIN_Echo);

Revisa la configuración (igual que en Otto Blockly)

- 1) Selecciona herramientas en la barra de menú ("Tools")
- 2) En la opción Placa ("Board") selecciona "Arduino Nano"
- 3) En la opción Procesador ("Processor") selecciona "Atmega328P" (selecciona "Old bootloader" si te da error)
- 4) En la opción Puerto ("Port") selecciona el COM USB donde está conectado tu Otto

ormat %T Sketch oding & Reload	sketch_aug03a Arduino 1.8.13				
Sketch oding & Reload					
ouning or noroda					ø
e Libraries 企業I					
fonitor					
Plotter 🗘 🏦 L					
1 / WiFiNINA Firmware Updater					
"Arduino Nano"	Boards Manager				
sor: "ATmega328P"	Arduino Vún				
dev/cu.wchusbserial1420" ► ard Info	Arduino Tun Arduino Uno Arduino Duemilanove or Diecimila				
mmer: "AVRISP mkll"	Arduino Nano Arduino Mega or Mega 2560 Arduino Mega ADK				
	Ionitor 수 원서 Iotter 수 원L 1 / WiFININA Firmware Updater "Arduino Nano" Sor: "ATmega328P" Sev/cu.wchusbserial1420" ard Info nmer: "AVRISP mkll" potloader	tonitor ☆ ೫€M lotter ☆ ೫€L 1 / WiFININA Firmware Updater Boards Manager *Arduino Nano" > bev/cu.wchusbserial1420" > ard Info Arduino Vún ard Info > nmer: "AVRISP mkll" > votloader >	tonitor ☆ ೫M lotter ☆ ೫L 1 / WiFiNINA Firmware Updater Boards Manager *Arduino Nano" > bev/cu.wchusbserial1420" > ard Info Arduino Yún ard Info > nmer: "AVRISP mkl!" > votloader >	Monitor ☆ 第KM Iotter ☆ 第L 1 / WiFININA Firmware Updater Boards Manager *Arduino Nano" > Arduino Yún Arduino Yún dev/cu.wchusbserial1420" > ard linfo Arduino Yún nmer: "AVRISP mkli" > votloader >	Monitor ☆ 第M Iotter ☆ 第L 1 / WiFININA Firmware Updater Boards Manager *Arduino Nano" > Arduino Yún Arduino Yún Arduino Vuno Arduino Uno ard Info Arduino Duemilanove or Diecimila • Arduino Maga or Mega 2560 Arduino Mega ADK

Carga el código a Otto

Una vez verificado, puedes cargar el código a Otto.

Por último, selecciona la flecha que apunta hacia la derecha para cargar.



Básicamente acabamos de crear un código en C++ para Arduino sin escribir una sola línea de código. ¡Tu Otto es ahora un máquina de caminar y bailar!

Ahora, vamos a aprender lo básico

Niveles & idioma









<u>Ayúdanos a</u> <u>mejorar y agregar</u> <u>nuevos idiomas</u> <u>aquí</u>



		Nivel 1 Barra de herramientas para Principiantes con todos los				
	Structure	novimientos principales de Otto				
Þ	🗑 Otto	Estructura: Setup y Loop, corchetes principales para el programa.				
Ī.	🗇 Time	Otto: Movimientos, sonidos y gestos. Sensores: Ultrasonido, tácto, sonido y giro.				
	📰 Input / Output	Ojos LED: Matriz I2C 16x8 .				
i.	Logic Control	Boca LED: Matriz SPI 8x8.				
1	🔢 Math Operators	Tiempo: Usado para delimitar el programa en el tiempo.				
İ.	E Functions	Lógica: Crea condicionales, repeticiones y lógica de programación.				
	🔀 Variables	Funciones: crea un proceso que puede repetirse.				
		Variables: Un valor con nombre, que puede cambiar.				



Sensing

Displays

MuVision

Audio Sound

Communication

Mouse & Keys

Motor

LED

Nivel 2 Barra de herramientas avanzada con bloques para proyectos Arduino sin fin.

Sentidos: Todo tipo de sensores que pueden agregarse a Otto o para crear otros proyectos de Arduino,
Motor: Controla todo tipo de servos, motores paso a paso y motores dc.
LED: RGB, Neopixeles y matriz de LED.
Pantallas: LCD, OLED y otras pantallas.
Audio: emite un sonido con un buzzer o reproduce canciones mp3
Comunicación: con módulos. Bluetooth, IrDa o Serial.
Teclados y Mouse: (solo para Arduino Leonardo y Micro)
MuVision: Grabación de video o reconocimiento facial más avanzado.



Setup / Loop

Primero haz click en Estructura en la parte derecha de la barra de menú

Comenzaremos con el bloque Setup / Loop

Loop abre y cierra una secuencia de funciones y las sigue repitiendo por siempre.

Si quieres reproducir tu código sólo una vez, solo utiliza la sección de **Setup**.

La sección de setup también sirve para inicializar componentes.

Arrastrar y soltar bloques

Desde la barra de menú selecciona Otto, arrastra de la lista de bloques siempre el bloque de configuración en el espacio de setup y cualquier otro bloque en el espacio de loop



Camina antes de

correr

Caminar en bloques Caminar en código C++ Otto.walk(1,1000,1); Setup Configure Leg left 2 right 3 Foot left [4 right •• 😂 Loop Move forward Speed normal

¿Puedes hacer correr a Otto?

🖸 Loop			
Move	↑ forward 🔹	Speed	very fast 🔹
			normal slow very slow
		1	fast very fast
			way to fast

Otto.walk(1,500,1);



Conexión de los servos

Ejemplo en Blockly para controlar cada motor





∞ Loop
rotate pin (2) angle [0°-180°] (90)
rotate pin (3) angle [0°-180°] (90) 180°
rotate pin (4 angle [0°-180°] (90)
rotate pin (5 angle [0°-180°] (90 180°
wait 200 milliseconds
rotate pin (4) angle [0°-180°] (2 0)
💽 wait 📢 200 milliseconds 🔽

Agrega o duplica un bloque más de servo y espera de tiempo

¿Qué sucede ahora?

¡Hora de bailar!



¡Haz tu propio baile! Arrastra y suelta más bloques de Otto.

Otto.walk(2,1000,1); función de mover ("#pasos, Tiempo[ms], dirección")

Otto.tiptoeSwing(2, 1000, 25); Otto.moonwalker(5, 1000, 40, 10; • Otto.moonwalker(5, 1000, 40, -1); Otto.crusaito(5, 1000, 40, 1); (pasos,Tiempo[ms],TamañodelMovimento,dir ección);

¿Otto puede hablar?



Piezo Buzzer

Es un actuador utilizado para generar sonido, pitar o incluso hacer melodías de una canción.

Otto no puede hablar pero puede crear sonidos relacionados a sus emociones, similar a R2D2.

Asegúrate de identificar la marca positiva y conectarlo en el pin correcto.

Es de tipo pasivo.





Comparado con otros componentes solo tiene dos cables/pines, uno se conecta a GND y el que tiene el signo + en la parte superior, lo conectamos al pin de señal número 13.

Puedes ver que comparado con las demás partes que usamos antes esta no necesita V.

Sound		happ	oy1	•
· · · · · · · · · · · · · · · · · · ·	happy1			
	happy3			
	sad confused			
	Cuddly			
	Oh			
	Surprise			

Bloques de sonido

Tienes 19 sonidos diferentes predefinidos relacionados a emociones con emojis. En código Arduino esto se traduce así:

Otto.sing(**S_superHappy**); función de cantar ("sonido a reproducir")

(S_surprise); (S_OhOoh); (S_OhOoh2); (S_cuddly); (S_sleeping); (S_happy); (S_happy_short); (S_sad); (S_confused); (S_buttonPushed); (S_fart1)...

¿Qué es un gesto?



Otto.playGesture(OttoLove); reproducir función de Gestos ("emoción/estado de ánimo")

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(OttoFail);

¿Cómo puede "ver" Otto?



Sensor de ultrasonido

Se usa para medir la distancia hasta un objeto utilizando ondas de ultrasonido, tiene 4 pines:

VCC (Energía), GND (Tierra) TRIG recibe la señal de control ECHO envía una señal (pulso)

Al medir la duración del pulso, podemos calcular la distancia.

¡Otto puede evitar obstáculos!



Ejemplo de esquivar



Setup Home USB Serial 9600
C Loop
Serial Print on same line 🕴 🎸 Ultrasonic sensor reading: 😕
Serial Print on same line Distance [cm]
Serial Print on same line 🕻 🎸 cm 🥺
Serial Print on new line

Medir distancia

- 1. Veamos un ejemplo para medir distancia.
- 2. Haz click en Cargar
- 3. Luego abre el monitor serial haciendo click en la esquina superior derecha "ícono de búsqueda".

Debes mantener a Otto conectado al USB

Monitor Serial

En la ventana de Monitor Serial haz click en Iniciar ("Start")

Desde aquí puedes ver la distancia medida por el sensor, expresada en centímetros.

Pruébalo colocando tu mano más cerca o más lejo de Otto.

Puedes usar una regla o una cinta métrica para ver cómo es de preciso.



Juega con otros componentes

Puedes agregar brazos, todo tipo de sensores, matriz LED, pantallas OLED y mucho más, **¡el límite está en tu imaginación!**





Otto Scratch AI (Artificial intelligence en línea)







Otto Scratch Al

Comparado con Blockly, no es necesario instalarlo.

Previamente, carga un código de firmware específico para poder trabajar o crear juegos en 2D (prueba las plantillas)



Hola constructor de Otto Estoy de regreso, listo para ayudarte.



Configurar Scratch

Instantáneamente programa y controla Otto vía Bluetooth desde Otto Scratch

- 1. <u>Carga este código en tu Otto</u>
- Instala el plugin de Scratch link y actívalo (debería aparecer en tu barra de trabajo)
- 3. Enciende el Bluetooth de tu computadora



A

Carga el código utilizando el IDE de Arduino



Configura Scratch

4. Abre Otto Scratch Al

5. Haz click en las extensiones en el ícono de abajo a la izquierda y selecciona **Otto Plus**



¡Hey! soy yo en el ícono

Extensiones de Otto Scratch Al



Face Detection Face detection with Tensorflow Requires Collaboration with DEEPRUPTIVE 3



Pose Detection Pose detection with Tensorflow Requires Collaboration with 9 DEEPRUPTIVE







00

Otto Eyes

Requires

Control your Otto Eyes



Collaboration with

DEEPRUPTIVE



Otto Humanoid Control your Otto Humanoid Requires

Collaboration with DEEPRUPTIVE



Build your Chatbot.

Requires

0

Collaboration with DEEPRUPTIVE



Configuración de Scratch

6. Encuentra tu dispositivo y conéctalo; al hacer click en cualquier bloque de Otto, verás a Otto moverse.

7. Mantén el módulo Bluetooth conectado a tu robot Otto de forma correcta.

Pin BLE TX va al pin 11 y RX al pin 12, Gnd a G, Vcc a V



Todavía necesitas darme energía, entonces ¿USB o baterías?



Configuración de Scratch

Si tuviste éxito al hacer click en culaquier bloque, el Otto Real en tus manos deberá reaccionar inmediatamente vía Bluetooth



¡Juega con todos los bloques!



Inteligencia Artificial

Ahora intenta el ejemplo "Hazme sentir" que hace que cuando le hables a tu computadora, Otto reaccione en respuesta a lo que dijiste, demostrando una emoción.



¡Hazme feliz por favor!, adiós

¿Qué vas a hacer / programar / diseñar?

¡Comparte tus creaciones en nuestra comunidad!



builders.ottodiy.com