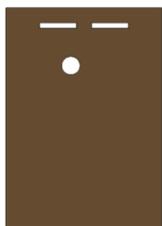




DYOR MinionBot

Materiales



1x Base



1x Frontal



1x Soporte



Accesorios: 2xBrazo,
2xBanana, 1xCorona



2x Servo motor
(rotación continua)



2x Rueda



1x Rueda loca



2x Servo motor



1x Matrix de LEDs 8x8



1x Sensor distancia



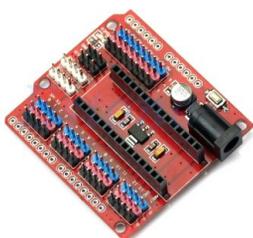
1x Zumbador sonido



1x Módulo Bluetooth



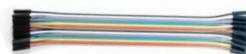
1x Seguíneas



1x Placa extensión



1x Arduino + Cable USB



2x10 Cables



1x Powerbank

Herramientas y materiales adicionales



Pistola encolar
y pegamento



Destornillador



Pintura (acrílica) y
pinceles



Electrónica

Arduino Nano v3.0

- Placa de desarrollo de hardware libre basada en el microcontrolador AtMega328 de bajo coste.
- Permite interaccionar con sistemas físicos gracias a su facilidad de programación, variedad en el manejo de entradas y salidas digitales y analógicas.
- Trabaja con señales de 5V de baja potencia.
- Capacidad de procesamiento limitada, pero más que suficiente para la gran mayoría de proyectos domésticos y educativos.
- Medición de todo tipo de señales eléctricas gracias a la gran variedad de sensores que son compatibles con esta electrónica.
- Control de motores: CC, servo motores, motores, paso a paso, etc...
- Manejo de dispositivos con buses de comunicación: I2C, SPI, 1-Wire, etc...
- Memoria 32KB para el programa, 2KB de SRAM y 1KB EEPROM.



Pines

Dispone de 22 pines que pueden ser usados como entradas y/o salidas (E/S), entradas analógicas o funcionalidades especiales según se muestra en la siguiente tabla, Los pines 0 y 1 están usados por el puerto serie (comunicación USB) y NO deberían ser usados para otro propósito.

PIN	Funcionalidad	PIN	Funcionalidad
0	Comunicación USB (RX)	11	E/S digital; PWM; SPI (MOSI)
1	Comunicación USB (TX)	12	E/S digital; SPI (MISO)
2	E/S digital; Interrupción 2	13	E/S digital; Led Integrado; SPI (SCK)
3	E/S digital; Interrupción 3; PWM	A0	Entrada analógica; E/S digital;
4	E/S digital	A1	Entrada analógica; E/S digital;
5	E/S digital; PWM	A2	Entrada analógica; E/S digital;
6	E/S digital; PWM	A3	Entrada analógica; E/S digital;
7	E/S digital;	A4	Entrada analógica; E/S digital; I2C (SDA)
8	E/S digital;	A5	Entrada analógica; E/S digital; I2C (SCL)
9	E/S digital; PWM	A6	Entrada analógica
10	E/S digital; PWM; SPI (SS)	A7	Entrada analógica

Programación

Facilino permite la programación por instrucciones de bloques. Algunas de sus instrucciones se activan mediante una licencia (<https://roboticafacil.es>). Debes usar el cable USB para programar Arduino Nano.

Descarga Facilino en: <https://roboticafacil.es/descargas/>

Drivers

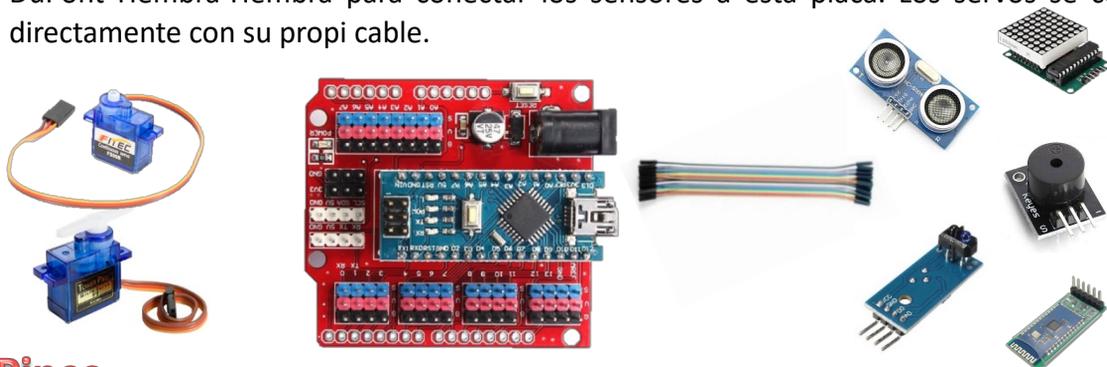
Los clones de Arduino utilizan el chip CH341G o similar. Windows 10 y Ubuntu lo reconocen como un puerto serie automáticamente. En Windows 7 macOS debéis configurarlo adecuadamente. **Más información en nuestras FAQ:** <https://roboticafacil.es/grupos/faq>



Electrónica

Placa Expansión Arduino Nano

La placa de expansión de Arduino Nano permite la fácil conexión de dispositivos al microcontrolador. Para ello, dispone de un conjunto de pines que permiten conectar la señal (S) asociada al pin de Arduino, la alimentación (VCC) y la masa (GND). Utiliza los cables DuPont Hembra-Hembra para conectar los sensores a esta placa. Los servos se conectan directamente con su propi cable.



Pines

Los pines pueden ser usados como entradas y/o salidas (E/S), entradas analógicas o funcionalidades especiales según se muestra en la siguiente tabla, Los pines D0 y D1 están usados por el puerto serie (comunicación USB) y NO deberían ser usados para otro propósito.

PIN	Funcionalidad	PIN	Funcionalidad
D0	Comunicación USB (RX)	D11	E/S digital; PWM; SPI (MOSI)
D1	Comunicación USB (TX)	D12	E/S digital; SPI (MISO)
D2	E/S digital: Interrupción 2	D13	E/S digital; Led Integrado; SPI (SCK)
D3	E/S digital; Interrupción 3; PWM	A0	Entrada analógica; E/S digital;
D4	E/S digital	A1	Entrada analógica; E/S digital;
D5	E/S digital; PWM	A2	Entrada analógica; E/S digital;
D6	E/S digital; PWM	A3	Entrada analógica; E/S digital;
D7	E/S digital;	A4	Entrada analógica; E/S digital; I2C (SDA)
D8	E/S digital;	A5	Entrada analógica; E/S digital; I2C (SCL)
D9	E/S digital; PWM	A6	Entrada analógica
D10	E/S digital; PWM; SPI (SS)	A7	Entrada analógica

Programación

Facilino permite la programación por instrucciones de bloques. Requiere de Arduino IDE para la correcta programación y algunas de sus instrucciones se activan mediante una licencia. Debes usar el cable USB para programar Arduino Nano.

Descarga Facilino en: <https://roboticafacil.es/descargas/>

Drivers

Los clones de Arduino utilizan el chip CH341G o similar. Windows 10 y Ubuntu lo reconocen como un puerto serie automáticamente. En Windows 7 y macOS debéis configurarlo adecuadamente. Más información en nuestras FAQ: <https://roboticafacil.es/grupos/faq>



Electrónica

Servos SG90 y FS90R

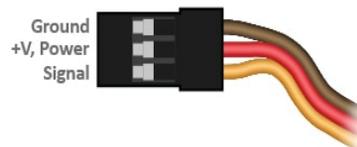
Existen dos tipos de servos, los de posición (SG90) y los de rotación continua (FS90R). Puedes fácilmente diferenciarlos por el nombre del modelo que viene indicado en la etiqueta. Ambos servos se pueden controlar con cualquiera de los pines digitales Arduino.



Servo de posición SG90



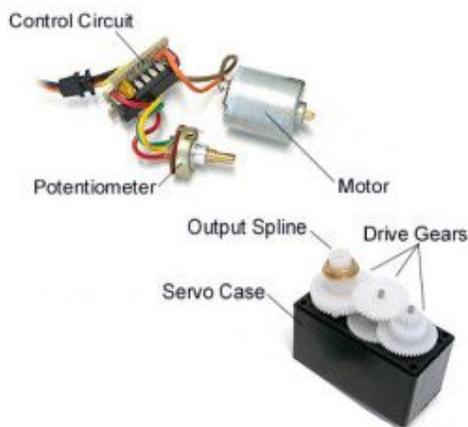
Servo de rotación continua FS90R



Conector de un servo

Servos de Posición

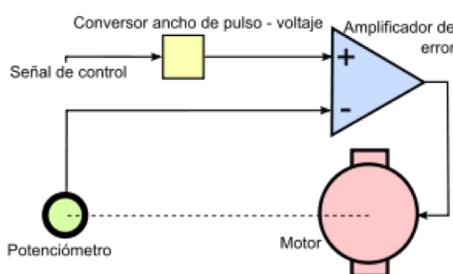
Los servos de posición pueden moverse entre 0° y 180°. Internamente disponen de un motor, un potenciómetro para medir la posición del eje, un circuito de control y unos engranajes que permiten aumentar el par del motor. No tratéis de forzar la posición de un servo a mano, dado que podríais dañar sus engranajes. El control interno de un servo es simple y no siempre alcanza la posición deseada, particularmente si tiene una "carga" grande que mover, aunque para la gran mayoría de aplicaciones esto no represente un problema.



Servos de Rotación Continua

Los servos de rotación continua permiten controlar la velocidad de giro de una rueda. Trabajan de forma similar a los servos de posición. La principal diferencia es que el valor del potenciómetro no depende de la posición del eje, si no que está fijo. Por tanto, la velocidad de giro es proporcional a la diferencia entre el voltaje deseado a partir de la señal de control y el valor del potenciómetro. Este valor se puede ajustar externamente para que a los 90° el servo esté quieto (suele ser lo habitual).

La velocidad real de un servo de rotación continua dependerá de factores como la carga del motor (peso del robot) y la tensión de alimentación (particularmente si se alimentan con pilas).

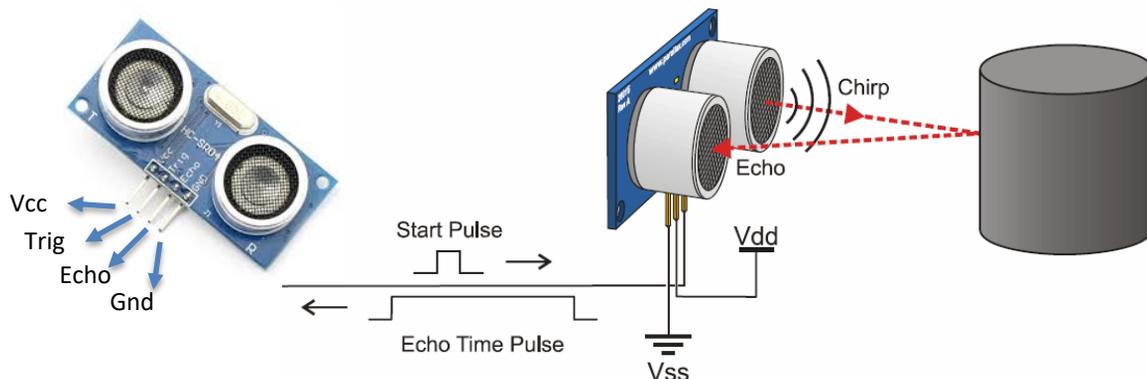




Electrónica

Sensor de Ultrasonidos HC-SR04

Es un sensor que mide distancia. El sensor dispone de un emisor que se activa con la señal Trig y un receptor que activa la señal Echo tras recibir la onda al rebotar con un objeto. La distancia al objeto es proporcional al tiempo que transcurre desde que se emite la onda hasta que se recibe*. Esta técnica de medición se conoce bajo el nombre de tiempo de vuelo.



El sensor HC-SR04 es un sensor poco fiable, ya que observaréis que en ocasiones proporciona mediciones incorrectas. Aún así, se considera un sensor que es indispensable en los robots de bajo coste, ya que nos permite reaccionar frente a los objetos del entorno para evitar colisiones.

A continuación os destacamos algunos de los problemas que podréis detectar:

- **Incertidumbre angular:** El sensor es capaz de medir objetos dentro de un ángulo de aproximadamente unos $\pm 30^\circ$. Esto hace que no sepamos con precisión en qué posición está el objeto, sólo sabremos que *delante* del sensor hay un objeto.
- **Reflexión especular:** Los objetos, al rebotar la onda, actúan como si fueran un espejo, es decir, que la onda rebota en la dirección opuesta. Esto hace que a partir de los 30° se muy difícil detectar objetos planos aún teniéndolos justo delante, p.e.: la pantalla del ordenador. En la práctica, la mayoría de los objetos no son totalmente planos, con lo que este efecto puede ser difícil de caracterizar si el objeto tiene rugosidades o contiene otros elementos internos.
- **Distancia máxima:** Objetos a partir de una distancia máxima de unos 400cm no pueden ser detectados.
- **Diafonía:** Cuando varios sensores emiten a la vez, sus ondas pueden cruzarse. Esto ocurre cuando los sensores se apuntan directamente el uno hacia el otro, pero también como consecuencia del rebote de la señal en los objetos. Esto también ocurre cuando un sensor dispara una onda antes de que la anterior haya vuelto, puede producirse el efecto de diafonía temporal. Es recomendable dejar como mínimo unos 50ms entre medición y medición del sensor de ultrasonidos para evitar este problema.

Estudad el comportamiento del sensor frente a diferentes materiales y objetos, ángulos de medición, etc..., para que podáis sacar vuestras propias conclusiones. Si disponéis de dos o más sensores, tratad de observar el efecto producido por la diafonía, para ver cómo afecta a las mediciones.

*La distancia es la velocidad del sonido en el aire multiplicada por la mitad del tiempo transcurrido.



Electrónica

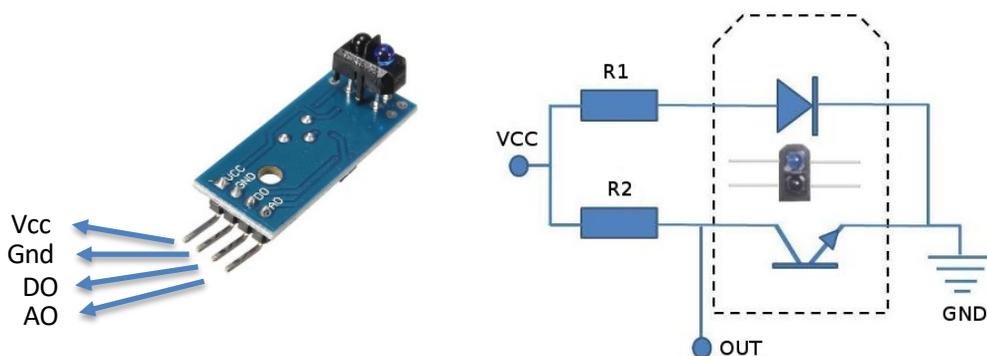
Módulo seguidor de líneas TCRT5000

El módulo seguidor de líneas TCRT5000 es en realidad un sensor de luz que puede utilizarse con varios propósitos:

1. Seguir líneas en el suelo (con un alto contraste entre la línea y el suelo).
2. Detectar una marca que delimita un área en la que el robot no debe salir.
3. Detectar el borde de una mesa.
4. Final de carrera óptico.

A continuación explicaremos brevemente su principio de funcionamiento y cómo usarlo en las aplicaciones anteriormente mencionadas. El sensor dispone de un emisor de luz infrarroja (LED) y un receptor de luz infrarroja (foto-transistor). La luz del LED rebota sobre un objeto (normalmente el suelo) y el foto-transistor capta la cantidad de luz devuelta.

Dependiendo de la distancia al objeto y del color del mismo, la cantidad de luz varía. Por tanto, pueden darse dos situaciones que permiten la detección de forma sencilla: que el color del objeto varíe o que la distancia del objeto varíe. El primer caso, es el principio de funcionamiento de las dos primeras aplicaciones, ya que normalmente el color del suelo cambiará en función de si estamos en la línea o fuera de ella (de blanco a negro o viceversa). El segundo caso es el principio de funcionamiento de las otras dos aplicaciones, ya que la ausencia de suelo (cuando estamos en el borde de una mesa) o la presencia de un objeto (como en los finales de carrera) modificarán la medición.



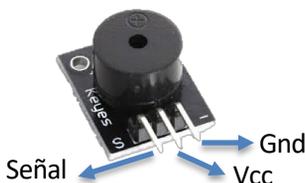
El módulo TCRT5000 dispone de dos tipos de salidas: digital (DO) y analógica (AO). La salida analógica es inversamente proporcional a la cantidad de luz recibida (cuanto más blanco sea el objeto o más próximo esté más pequeño será el valor medido). La salida digital se activará en función de un potenciómetro que incorpora el módulo TCRT5000. Este potenciómetro establece un nivel de comparación a partir del cual se activará o desactivará la salida digital. Lo recomendable es que para aplicaciones de seguimiento de línea, utilicemos la salida analógica, ya que nos proporcionará más información para poder ajustar el giro del robot para que no se salga de la línea. Mientras que, en el resto de aplicaciones anteriormente mencionadas, lo recomendable es usar la salida digital y ajustar convenientemente el potenciómetro para nuestra aplicación.



Electrónica

Zumbador de sonidos KY-006

El zumbador de sonidos (o buzzer en inglés) es un transductor capaz de convertir la energía eléctrica en sonidos (por lo general de alta frecuencia). El zumbador de sonido está basado en el principio de los materiales piezo-eléctricos de forma que al aplicar un voltaje éste emite un 'click' o sonido. Para generar notas musicales, se requiere de un oscilador o reloj que haga que vibre a la frecuencia deseada.



Existen unos zumbadores que son activos y que incorporan el oscilador de forma interna. A estos zumbadores, no se les puede modificar la frecuencia de oscilación y suelen usarse como alarmas y pitidos. El zumbador KY-006 es, por el contrario, un zumbador pasivo, lo que implica que debemos generar, mediante el pin de la señal la frecuencia de oscilación deseada. Arduino dispone de una instrucción para generar la vibración a la frecuencia (y duración) deseada.

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

A continuación incluimos las frecuencias de las notas musicales más comunes para que comprobéis el sonido que se genera*:

Nota	Frec. (Hz)	Nota	Frec. (Hz)
Do	261.63	Do# Reb	277.18
Re	293.66	Re# Mib	311.13
Mi	329.63		
Fa	349.23	Fa# Solb	369.99
Sol	392	Sol# Lab	415.3
La	440	La# Sib	466.16
Si	493.88	Do agudo	523.25

Melodías musicales

El zumbador de sonido se puede utilizar para reproducir sencillas melodías musicales (monotono). Para ello, debéis combinar notas musicales, pero además debéis de considerar su duración. Las redondas son las notas con la máxima duración; las blancas duran justo la mitad de una redonda; las negras duran la cuarta parte; las corcheas la octava parte y las semicorcheas una dieciseisava parte. Por tanto, debéis de determinar la duración de la redonda en función del ritmo que tenga vuestra melodía y calcular la duración para las otras notas musicales.

Entre dos notas musicales consecutivas es conveniente hacer una pequeña pausa, salvo que las notas estén unidas con el símbolo de ligadura. La pausa se puede hacer con una instrucción especial de Arduino que hará que el zumbador deje de vibrar.

<https://www.arduino.cc/reference/en/language/functions/advanced-io/notone/>

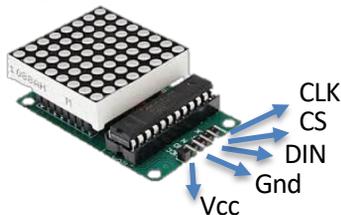
*Estas frecuencias cumplen con la fórmula de las notas musicales de la escala de Helmholtz $f_n = f_0 \cdot 2^{n/12}$, siendo f_0 una frecuencia fija de una nota, típicamente se usa $f_0 = 440$ Hz correspondiente al 'la'; y n es la distancia en semitonos con respecto a esa nota (positivo si es más aguda y negativo si es más grave). Por ejemplo, el 'do agudo' está a tres semitonos del 'la', con lo que de acuerdo a esta fórmula le corresponde la frecuencia $f_n = 440 \cdot 2^{3/12} = 523.25$.



Electrónica

Matrix de LEDs 8x8 max7219

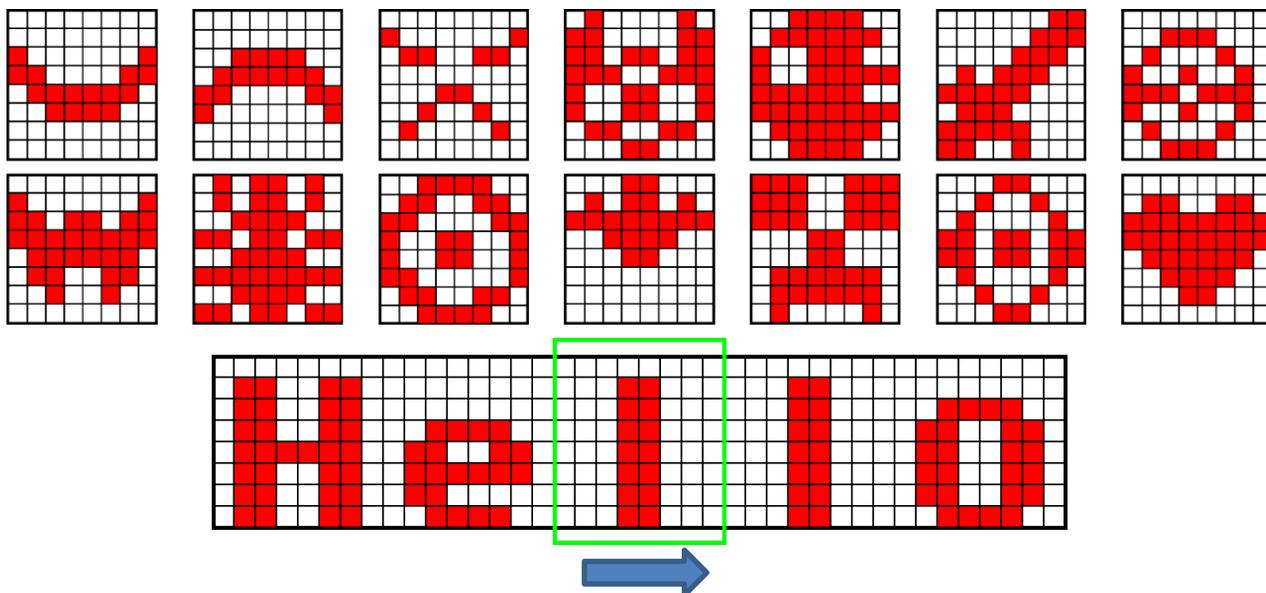
La matriz de LEDs max7219 permite mostrar un patrón de 8x8 LEDs en disposición matricial. Este dispositivo se utiliza típicamente para mostrar información de texto como un *display* digital.



El chip max7219 que se encarga de controlar el encendido y apagado de los LEDs funciona a través del protocolo de comunicación SPI (*Serial Peripheral Interface*). Se trata de un bus de comunicaciones muy utilizado en los microcontroladores cuando se quiere enviar información entre el microcontrolador y un dispositivo. Las señales necesarias para comunicar con el chip max7219 se pueden generar desde cualquiera de los pines de Arduino*. Concretamente, los datos se envían a través del pin DIN (Data Input), el chip recibirá los datos siempre que sea seleccionado con el pin EN (*Enable*), ya que de lo contrario ignorará la información. Para sincronizar adecuadamente el momento en el que los datos contienen información válida hace falta una señal de reloj CLK (*Clock*).

Expresiones, iconos, símbolos, texto...

Facilino permite expresar emociones y/o iconos, símbolos, texto o cualquier imagen personalizada. También pueden realizarse efectos dinámicos de forma que la imagen o texto a mostrar se desplace en una dirección (se conoce como flujo de texto o imagen). Así mismo, se tiene en consideración la orientación de la matriz, ya que ésta puede ser diferente en función del robot. A continuación mostramos algunas de las posibilidades que ofrece:



* Arduino utiliza los pines 10, 11, 12 y 13 para comunicar con dispositivos SPI, si bien, las instrucciones de Facilino permiten utilizar este dispositivo con tres pines digitales cualesquiera.

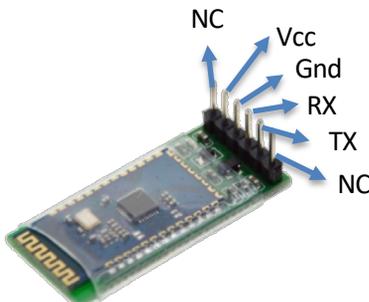


Electrónica

Bluetooth

El módulo Bluetooth SPP-C utiliza la especificación Bluetooth 2.0+EDR. La señal Bluetooth es una señal inalámbrica de 2.4GHz y cuyo rango de alcance es aproximadamente 10 metros. Este módulo está pensado para comunicar dispositivos inalámbricos como el móvil o la Tablet con Arduino.

Desde el punto de vista de Arduino, el dispositivo se comporta como un dispositivo serie universal (UART) que requiere de un pin de transmisión (TX) y otro de recepción (RX). El pin TX se utiliza para enviar información a Arduino, mientras que el pin RX se utiliza para recibir información de Arduino.



La comunicación es cruzada, quiere decir que el pin TX del dispositivo tiene que conectarse a cualquier pin de Arduino que configuremos como pin de recepción y viceversa. **No conectéis los pines directamente a los pines 0 y 1** que en la placa de expansión aparecen marcados como RX y TX, ya que estos pines se utilizan para la comunicación serie por USB. Facilino permite configurar la comunicación Bluetooth de forma que se debe indicar cuales son los pines de recepción y transmisión (desde el punto de vista de Arduino).

Emparejamiento

Debes emparejar el dispositivo Bluetooth. Proporciona alimentación al módulo a través de la placa de expansión, conectando los cables **Vcc** y **Gnd** a cualquier de los pines rojo y negro, respectivamente. En tu dispositivo móvil, debes buscar entre los dispositivos Bluetooth cercanos, el nombre por defecto que debe aparecer es **SPP-C**. Una vez seleccionado, debes introducir el pin que por defecto es **1234**.

Conexión

El robot DYOR dispone de una App en Play Store que puedes utilizar para el control remoto del robot. Descárgate la App con el código QR proporcionado.

Comprueba que cuando te conectas al módulo (pulsando sobre el botón de conexión con el símbolo Bluetooth) el LED del módulo Bluetooth deja de parpadear y cuando te desconectas vuelve a parpadear.

Hasta que no lo programes, el robot no se moverá, así que es inútil que aprietes el resto de iconos por el momento...



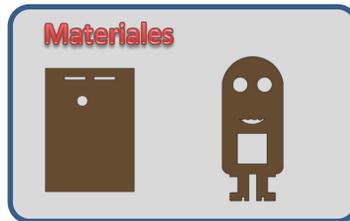
*Afortunadamente, ambos dispositivos se pueden comunicar directamente, aunque lo recomendable sería disponer de un adaptador de tensión de 5V a 3.3V para no dañar el módulo Bluetooth.



Instrucciones de montaje

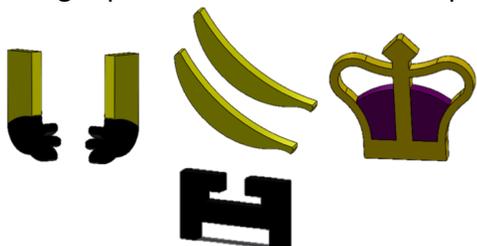
Paso 1

Pintar la base de amarillo y el frontal del robot con el aspecto de vuestro Minion preferido (ver imagen de referencia).



Paso 2

Pintar los accesorios: los brazos de amarillo con guantes negros, las bananas de color amarillo y la corona dorada y morada. El soporte para el módulo seguidor se pinta de negro por estar a la altura de los pies.



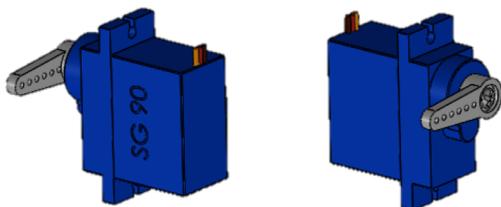
Paso 3

Colocar la goma de la rueda y atornillarla a los servos de rotación continua (en la etiqueta aparece el modelo del motor FS90R).



Paso 4

Atornilla las manillas simples de los servos de posición (SG90) en la posición que se indica en la figura (observa la posición del cable y la orientación de la manilla para cada uno de los servos).

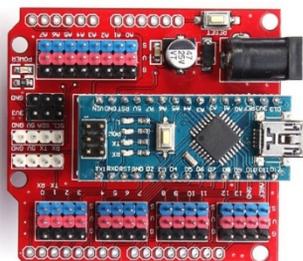




Instrucciones de montaje

Paso 5

Inserta el microcontrolador Arduino Nano v3.0 en la placa de expansión de forma que el conector queda como se muestra en la figura. En ocasiones, cuesta insertar el microcontrolador. Asegúrate que los pines están entrando de la forma correcta antes de aplicar una presión fuerte.

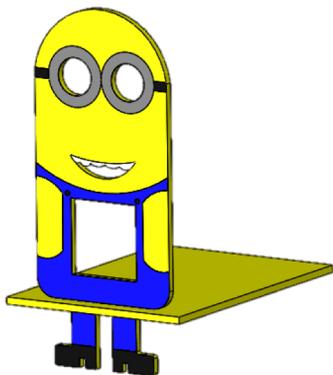


Materiales



Paso 6

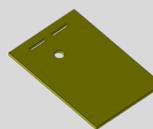
Pega el frontal del robot con la base. Para que el pegamento no sea demasiado visible, puedes aplicar el pegamento por la parte de detrás del frontal.



Herramientas

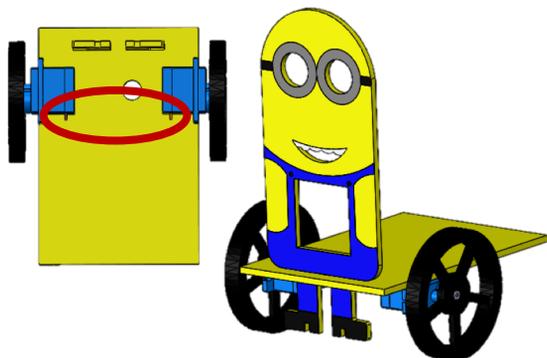


Materiales



Paso 7

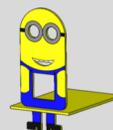
Ahora, dale la vuelta al robot y por la parte de debajo de la base, pega los servos de rotación continua de forma que los ejes estén alineados, es decir, a la misma altura y en la parte delantera del robot (observa la figura que están a la altura del agujero aprox.). El cable de los servos debe salir hacia la parte de detrás del robot.



Herramientas



Materiales



2x

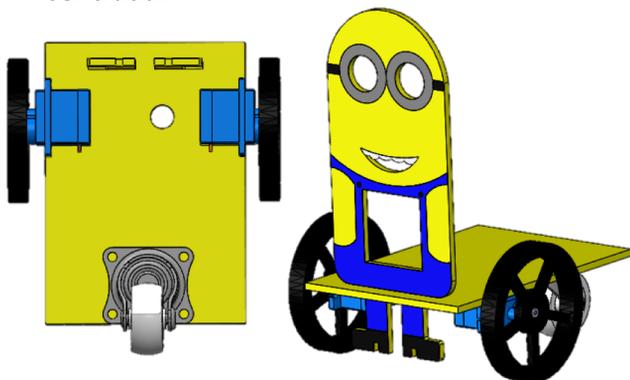




Instrucciones de montaje

Paso 8

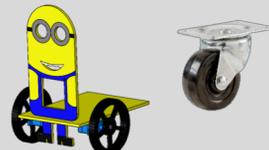
Pega la rueda loca en la parte en la parte trasera. La rueda debe estar lo más atrás posible y centrada.



Herramientas

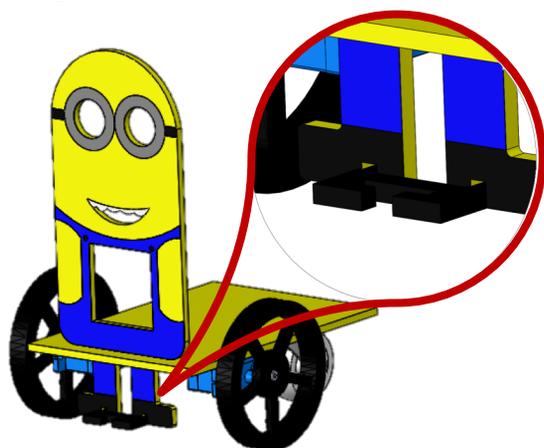


Materiales



Paso 9

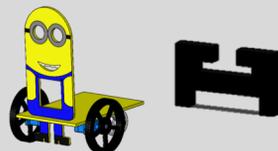
Pega el soporte a los pies del frontal del robot, procurando que forme un ángulo de 90°.



Herramientas

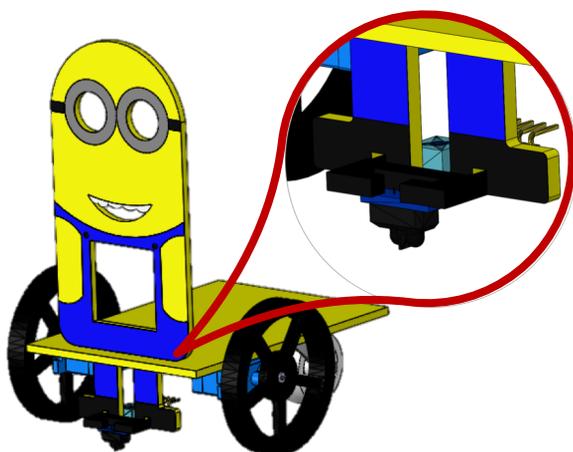


Materiales



Paso 10

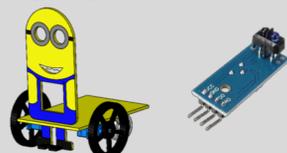
Ahora pega el sensor seguilíneas al soporte asegurándonos que está apuntando hacia el suelo a poca distancia (~5mm).



Herramientas



Materiales

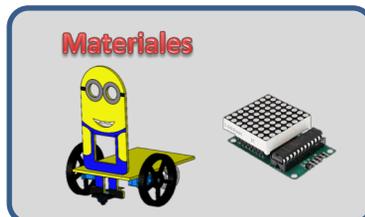
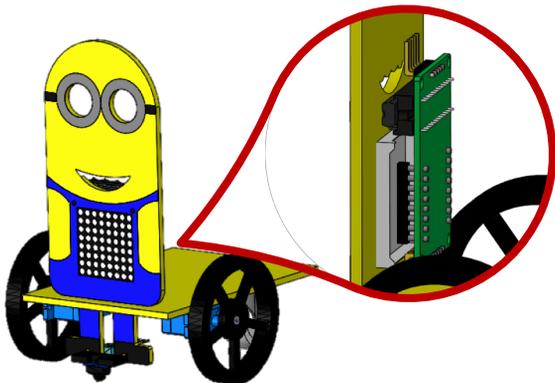




Instrucciones de montaje

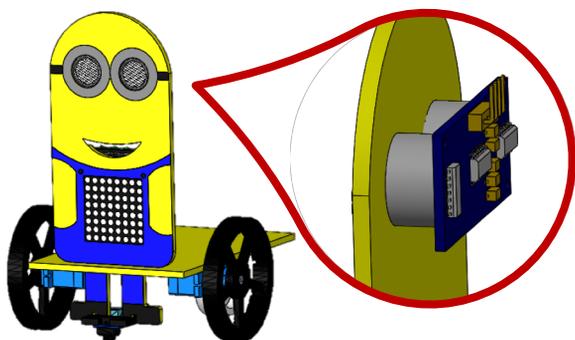
Paso 11

Pega la matriz de LEDs por la parte trasera del frontal. El conector con el chip "max7219" debe quedar por la parte superior. Lo habitual es que la matriz de LEDs no sobresalga por la parte delantera del frontal.



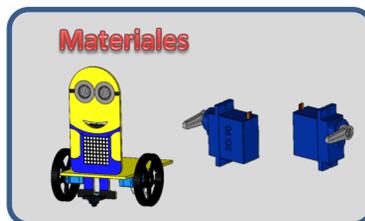
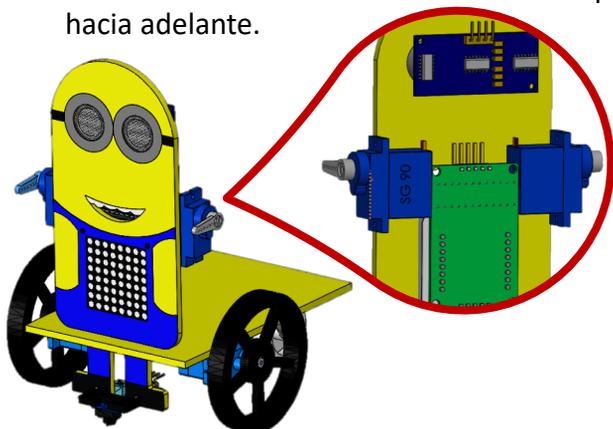
Paso 12

Pega el sensor de ultrasonidos por la parte trasera del frontal. El sensor debe quedar de forma que no sobresalga por la parte delantera y con los pines de conexión apuntando hacia arriba.



Paso 13

Ahora pega los servos de posición en la parte trasera del frontal a la misma altura como se muestra. Los cables de los servos salen por la parte superior y las manillas deben apuntar hacia adelante.

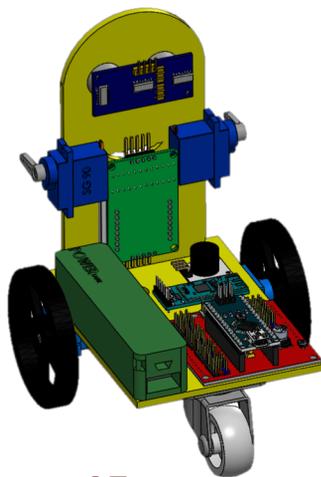




Instrucciones de montaje

Paso 14

Pega ahora el resto de la electrónica en la base del robot con la disposición que se muestra en la figura. El Powerbank tiene los conectores en la parte trasera del robot, así como la placa de expansión, que junto con el microcontrolador Arduino Nano deben tener los conectores en la parte trasera. El módulo Bluetooth y el zumbador de sonido pueden ir en cualquier lugar.



Herramientas



Materiales



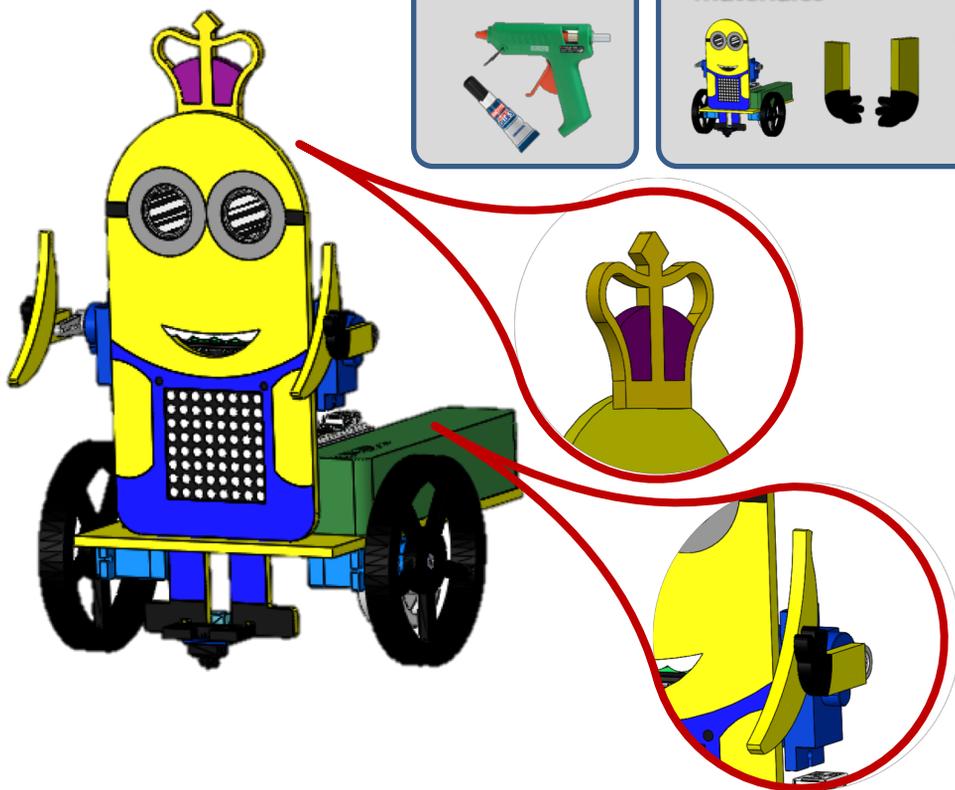
Paso 15

Pega los brazos a las manillas de los servos. Luego pega una banana en cada mano. Finalmente pega la corona en la parte superior del frontal (por la parte de detrás).

Herramientas



Materiales





Instrucciones de Conexión

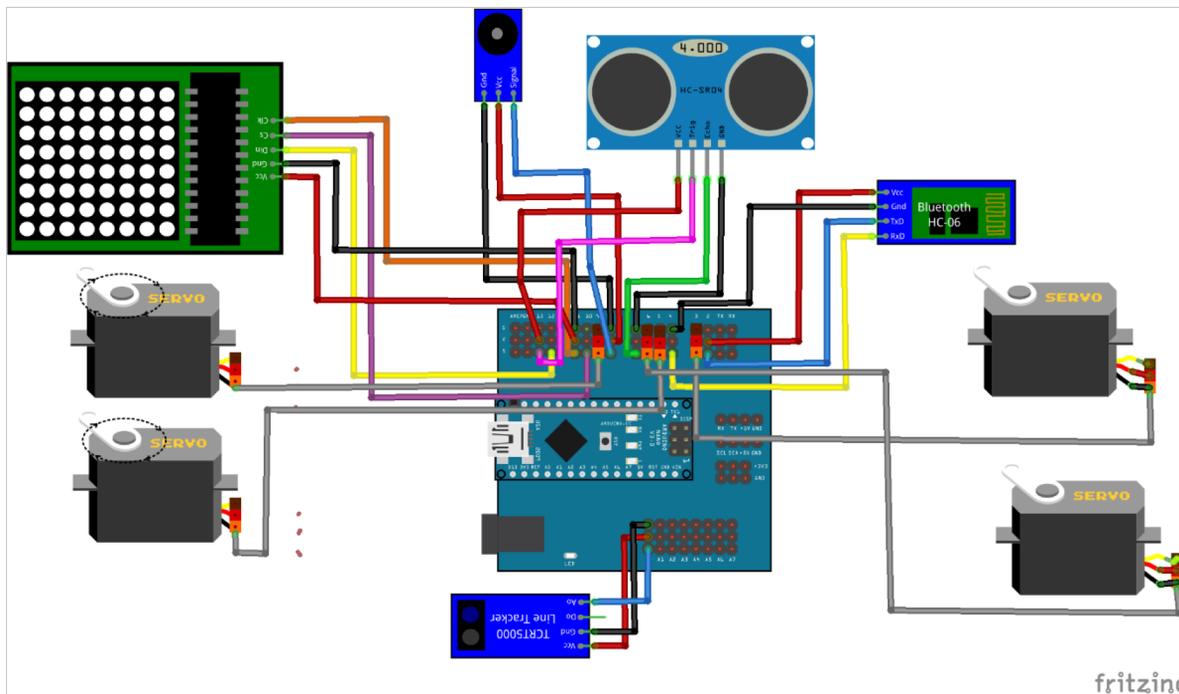
Tabla de conexión

La placa de expansión de Arduino Nano dispone de unos números indicando el número del pin.

Los pines **VCC** se conectan en la fila **roja**. Los pines **GND** se conectan en la fila **negra**.

A continuación os mostramos una propuesta de conexión que utilizaremos en los ejemplos posteriores.

Dispositivo	PIN	Observaciones	
Servo rotación (izquierdo)	9	El cable NARANJA del servo en el pin azul (señal).	
Servo rotación (derecho)	5	El cable NARANJA del servo en el pin azul (señal).	
Servo posición (izquierdo)	6	El cable NARANJA del servo en el pin azul (señal).	
Servo posición (derecho)	3	El cable NARANJA del servo en el pin azul (señal).	
Matrix de LEDs 8x8	10	CS	Los otros cables son Vcc y Gnd (conectar adecuadamente)
	11	CLK	
	12	DIN	
Zumbador (KY-006)	8	Los otros cables son Vcc y Gnd (conectar adecuadamente)	
Sensor ultrasonido SR-HC04	7	Pin ECHO.	Los otros cables son Vcc y Gnd (conectar adecuadamente)
	13	Pin TRIGGER.	
Bluetooth	2	RX (Arduino) TX (Bluetooth)	Los otros cables son Vcc y Gnd (conectar adecuadamente)
	4	TX (Arduino) RX (Bluetooth)	
Seguirlíneas TCRT5000	A0	A0	Los otros cables son Vcc y Gnd (conectar adecuadamente) . DO sin conectar.

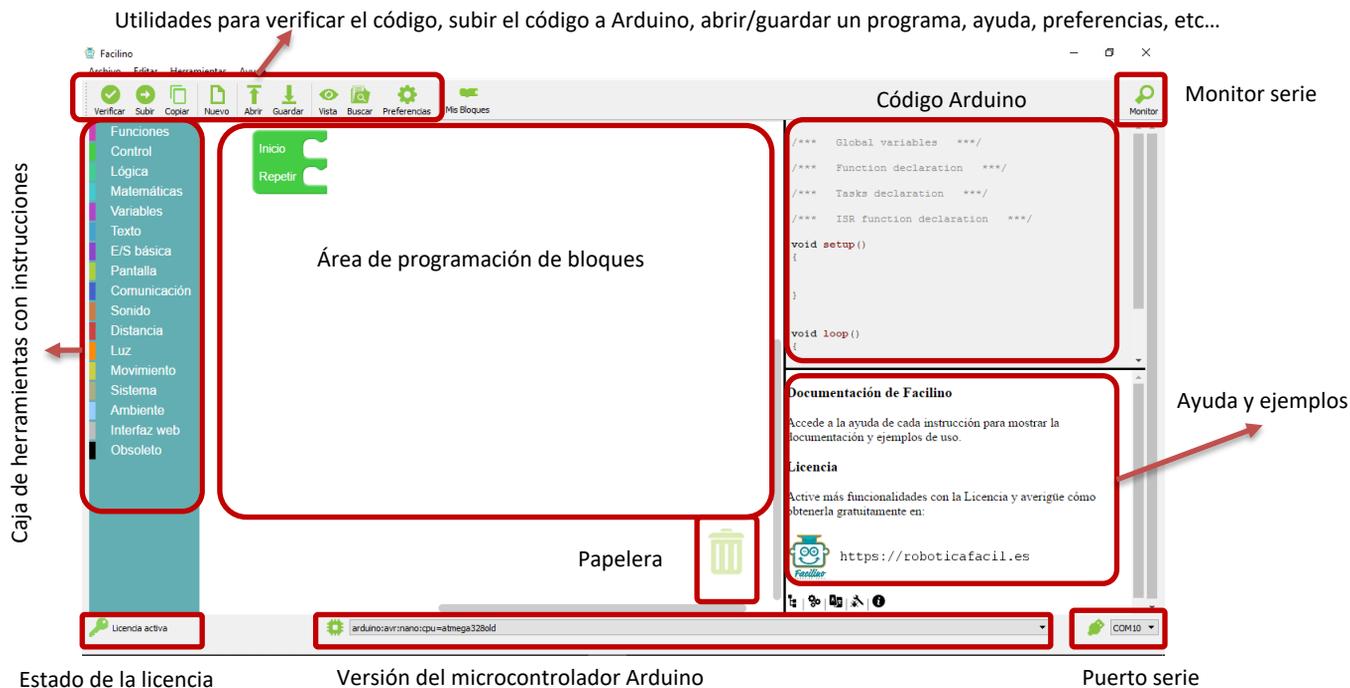




Facilino

PROGRAMACIÓN DE ARDUINO CON FACILINO

Facilino permite, mediante bloques, crear código para realizar determinadas funciones en Arduino como mover los motores del robot, detectar obstáculos, generar melodías, etc.



¿CÓMO ENVIAR UN PROGRAMA?

- A partir de la versión Facilino 1.4.0, aseguraos que la versión del microcontrolador es “arduino:avr:nano:cpu=atmega328old”, siempre y cuando la versión de Arduino IDE sea 1.8.x.
- El puerto serie dependerá de la configuración de vuestro ordenador, pero al conectar y desconectar el cable USB os cambiará la lista, con lo que fácilmente lo podréis averiguar.
- Al conectar el cable USB se encenderá la luz del LED POW de forma fija.
- Al verificar un código aparecerá una ventana que nos indicará si hay algún problema. Si todo es correcto debe indicar “Build finished” o “Finalizó”.
- Al subir un código, primero verificará que esté bien, es decir que no tiene ningún error de compilación. Después lo enviará a Arduino y las luces de los LEDs TX y RX parpadearán durante un breve instante de tiempo. Cuando haya finalizado la carga del programa, éste comenzará a ejecutarse nada más finalizar.

SI TENÉIS UN CABLE MAL CONECTADO, PUEDE QUE ESTÉIS CREANDO UN CORTOCIRCUITO QUE PODRÍA DAÑAR AL ROBOT. DESCONECTAD INMEDIATAMENTE SI DETECTÁIS ALGÚN PROBLEMA Y AVISAD AL PROFESOR/A.





Bloques de Facilino

FUNCIONES

variable: Entero x parámetros

func. (sin retorno) f1 ejecutar Define funciones y agrega los parámetros de entrada.

func. (con retorno) f2 ejecutar devuelve Entero

f1 Llama a las funciones desde la parte del código quieras.

f2

CONTROL DE FLUJO

Esperar [ms] 1000 Espera un tiempo en ms.

Espera a Valor1 Valor2 Espera condicional.

Inicio Repetir Ejecuta instrucciones al inicio (cuando arranca Arduino) o de forma repetitiva (en el bucle principal).

mientras Hacer Repite instrucciones hasta cumplir una cierta condición.

Hacer mientras

si ejecutar Ejecuta instrucciones si se cumple una cierta condición.

si

Contar con desde hasta ejecutar Repite instrucciones un número de veces determinado.

MATEMÁTICAS

0 Número (entero o decimal)

+ Operaciones aritméticas.

Mapea Valor entre [0-] Escala un valor.

Aleatorio entre y Genera un número aleatorio.

LÓGICA

= Compara valores.

y Operador lógico.

no Negación lógica.

verdadero Constante Booleana.

VARIABLES

Usa una variable existente.

Var mi_var

Crea una variable local.

Declara variable de tipo Texto =

Crea una variable global.

Declara variable GLOBAL de tipo Texto =

E/S BÁSICA

Pin digital D0 Selecciona un pin digital.

Pin analógico A0 Selecciona un pin analógico.

Lectura/Escritura de señales digitales.

LED en la placa estado Encendido

Lee el pin digital PIN

Escribe en el pin digital PIN estado

Lectura/Escritura de señales analógicas.

Lee el pin analógico Pin

Escribe en PIN digital el valor analógico

TEXTO

" " Cadena de texto. ' ' Carácter.

Crea texto con Crea cadenas de texto con texto, números, variables, etc...

A número Entero Convierte un texto a un número.

COMUNICACIÓN USB

Imprime texto por el puerto serie (almacena en buffer de salida).

Imprime por puerto serie

Imprime texto por el puerto serie (envía inmediatamente todo el buffer de salida).

Imprime por puerto serie con salto de línea

Dibuja datos por puerto serie Envía datos para ser graficados.



Bloques de Facilino

MOVIMIENTO

Mover base (servos)

Izquierda Pin digital D9

Derecha Pin digital D5

Velocidad % 50

Avanza

¿Desconectar después?

Movimientos simples del robot (avanzar, girar, retroceder o parar). Establece la velocidad de movimiento en porcentaje (de 0% a 100%).

Mover base (servos)

Izquierda Pin digital D9

Derecha Pin digital D5

Velocidad % 100

Lineal (-100~100) 50

Giro (-100~100) 15

¿Desconectar después?

Movimientos avanzados (velocidad lineal y angular). Establece los valores en porcentaje.

Pinza

Izquierda Pin digital D6

Derecha Pin digital D3

Abre pinzas

¿Desconectar después?

Abre o cierra los dedos de una pinza.

Brazos

Izquierda Pin digital D6

Derecha Pin digital D3

Levanta brazos

¿Desconectar después?

Levanta o baja los brazos del robot.

Si seleccionas 'desconectar después' los motores no consumirán corriente al cabo de un tiempo.

PANTALLA

Matriz 8x8

CS Pin digital D10

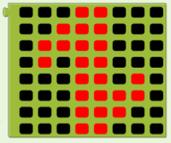
DIN Pin digital D12

CLK Pin digital D11

Vertical

Cabreado

Muestra una expresión en la matriz de LEDs. Necesitas indicar los pines, la orientación y la expresión a mostrar.



Expresión personalizable.

Matriz 8x8

CS Pin digital D10

DIN Pin digital D12

CLK Pin digital D11

Vertical

Esperar [ms] 100

¿Desplazamiento suave?

¿Tarea de fondo?

Hello

Creas texto animado o una secuencia de imágenes a mostrar en la matriz de LEDs. Puedes hacer un desplazamiento suave (de columna en columna) de los píxeles. Si seleccionas la tarea de fondo la animación se reproduce mientras ejecutas las siguientes instrucciones.

LUZ

Establece la cantidad de giro. La ganancia establece la sensibilidad al giro. **El sensor debe calibrarse.**

Seguidor de líneas

Luz

Sensor infrarrojo

PIN Pin analógico A0

Ganancia % 10

Calibra blanco 40

Calibra negro 730

Calibración del sensor a partir de valores que han sido previamente anotados a partir de lecturas del sensor (usar la principio del programa).

SONIDO

Creas una melodía musical. Si seleccionas la tarea de fondo reproduce la música mientras ejecuta las siguientes instrucciones.

Zumbador

PIN Pin digital D3

STAR WARS

¿Tarea de fondo?

Reproduce un sonido predefinido.

Zumbador

PIN Pin digital D3

Conexión



Creas tu propia partitura añadiendo notas musicales al pentagrama. **La partitura debe acabar con el fin de pentagrama.**

ULTRASONIDOS

Sensor de Ultrasonidos

Eco Pin digital D8

Disparo Pin digital D9

Devuelve la distancia al objeto más próximo detectado con el sensor de ultrasonidos. **La distancia se mide en centímetros.**

Detecta colisión

Eco Pin digital D8

Disparo Pin digital D9

Distancia 10

Colisión

Libre

Si un objeto está más próximo a la distancia indicada entonces se ejecutará el código que introduzcamos dentro del "Colisión". En caso contrario, se ejecutará el código que introduzcamos en "Libre".

COMUNICACIÓN BLUETOOTH

Define los pines a los que está conectado el módulo Bluetooth. **Esta instrucción debe usarse al inicio del programa.**

Define Bluetooth

Tasa de baudios 9600

RX Pin digital D2

TX Pin digital D4

Recibe comandos a través de Bluetooth. La instrucción es configurable de forma que podéis agregar tantos comandos queráis. Los comandos están predefinidos (ver tabla de comunicación Bluetooth más adelante).

Bluetooth Recibir Comando

comando Comando Avanza

Repetir?

hacer

Si seleccionáis 'Repetir?' la instrucción recordará el comando enviado cada vez que se evalúe de nuevo hasta que se reciba el comando de cancelación. Esto es útil para los comandos de evitación de obstáculos o seguir líneas.



Ejemplos

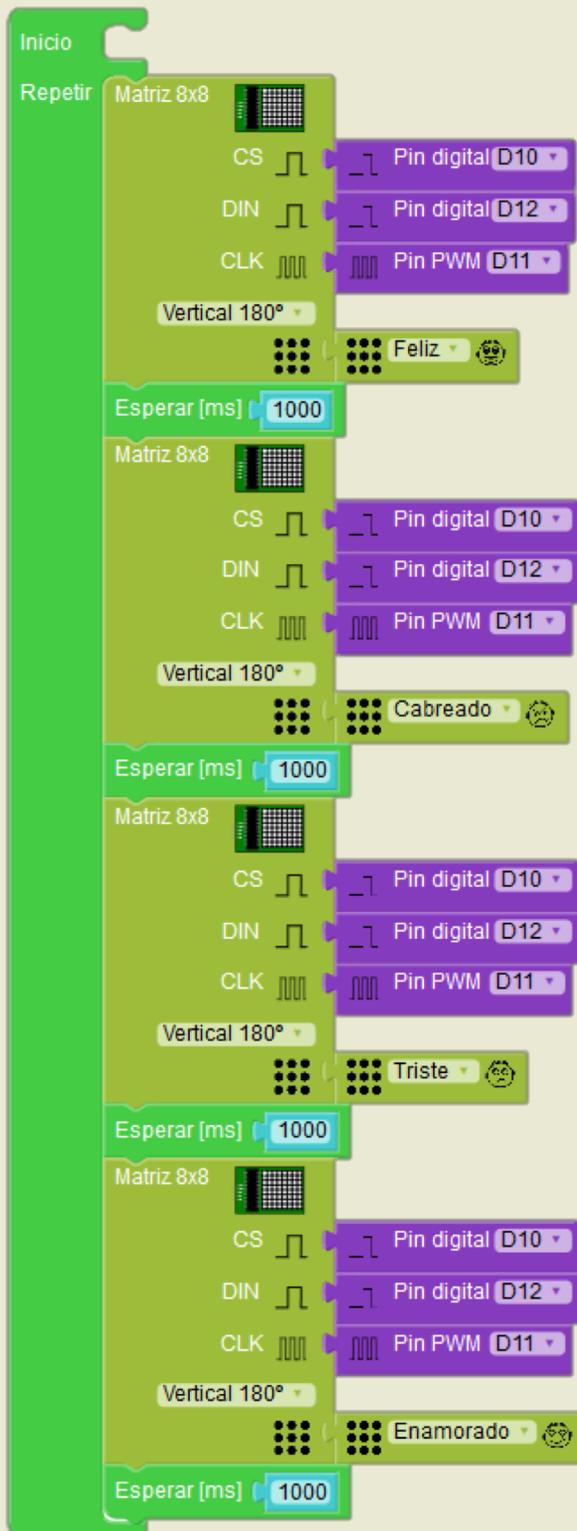
MOSTRAR EXPRESIONES PREDEFINIDAS

Objetivos

Mostrar expresiones predefinidas con la matriz de LEDs.

Comentarios

Por la orientación de la matriz de LEDs, debe seleccionarse la opción 'Vertical 180°'.





Ejemplos

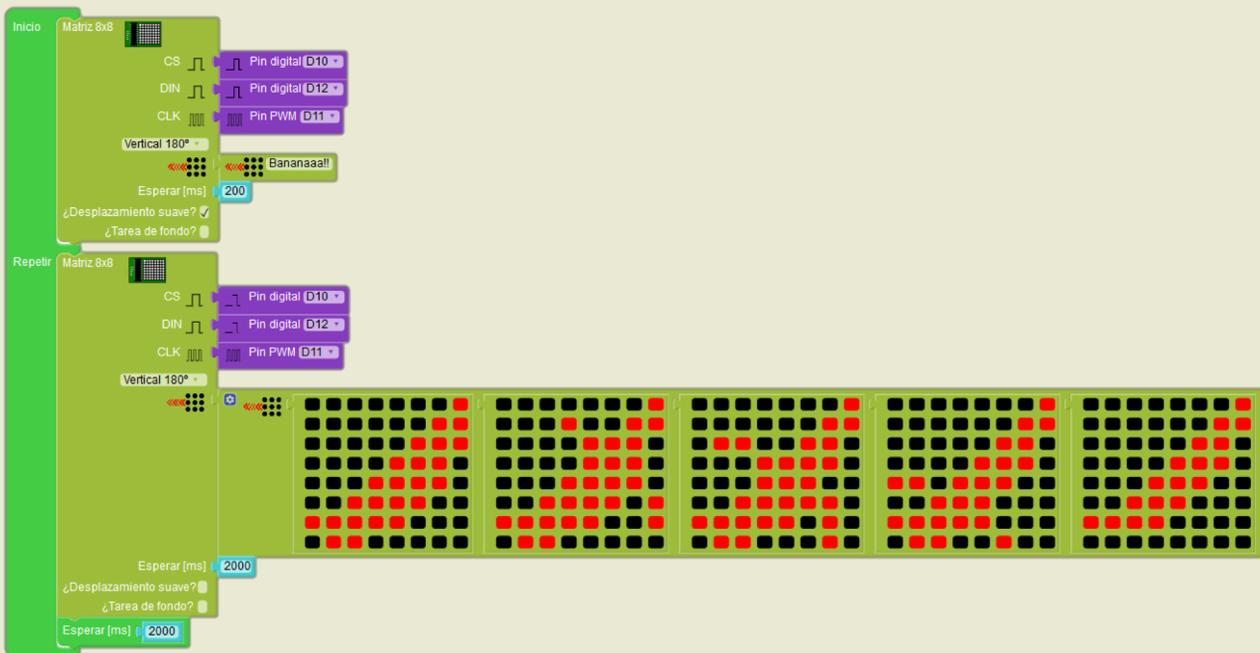
MOSTRAR TEXTOS Y ANIMACIONES PERSONALIZADAS

Objetivos

Mostrar textos y animaciones personalizadas con la matriz de LEDs.

Comentarios

Se muestra primero el texto 'Bananaaa!!'. Al estar la casilla de desplazamiento suave activada, el texto de desplazará de derecha a izquierda de forma suave. Por otro lado, la animación personalizada representa una banana pelándose (al no estar seleccionada la casilla anterior, se mostrarán las imágenes de forma individual transcurrido el tiempo indicado).





Ejemplos

REPRODUCIR SONIDOS PREDEFINIDOS Y MELODÍAS

Objetivos

Reproducir sonidos predefinidos y melodías personalizadas con el zumbador de sonidos.

Comentarios

Primero se muestra un par de ejemplos de sonidos predefinidos. Después se reproduce la primera estrofa de la canción 'Bad' de Michael Jackson que aparece en la película 'Mi Villano Favorito 3' (Depicable Me 3).

The code block starts with a green 'Inicio' (Start) tab. It contains three sequential blocks:

- Zumbador** (Speaker) block: Set to 'PIN' (square wave icon), 'Pin digital D3', and 'Sorpresa' (Surprise) sound.
- Esperar [ms]** (Wait) block: Set to 1000 ms.
- Zumbador** (Speaker) block: Set to 'PIN' (square wave icon), 'Pin digital D3', and 'Pedo 1' (Pedal 1) sound.
- Esperar [ms]** (Wait) block: Set to 1000 ms.
- Zumbador** (Speaker) block: Set to 'PIN' (square wave icon), 'Pin digital D3', and a custom melody. The melody is represented by a treble clef and a series of notes on a staff.

The code block ends with a '¿Tarea de fondo?' (Background task?) checkbox and a 'Repetir' (Repeat) tab.



Ejemplos

MOVER LA BASE DEL ROBOT

Objetivos

Mover la base del robot mediante primitivas de movimiento sencillas.

Comentarios

El robot se mueve hacia adelante y hacia atrás de forma repetitiva.





Ejemplos

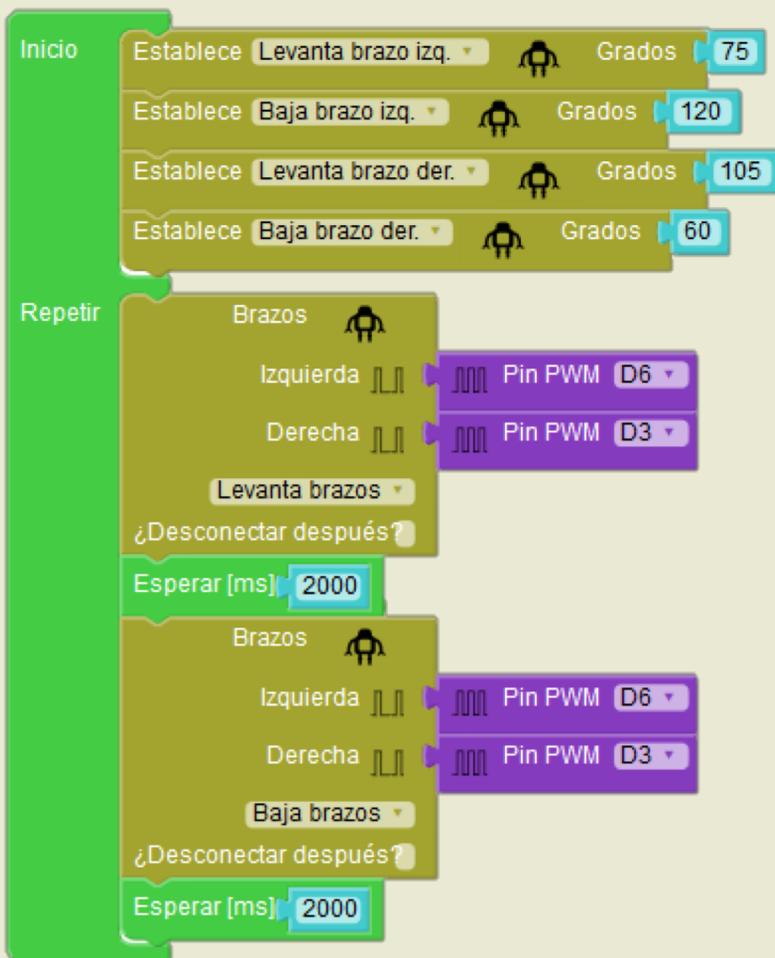
MOVER LOS BRAZOS DEL ROBOT

Objetivos

Mover los servos de posición de forma coordinada (son los brazos del robot).

Comentarios

Según cómo hayáis atornillado la manilla del servo a cada uno de los brazos, puede que los valores de calibración de la posición sean diferentes. Ajustad esos valores convenientemente para que el funcionamiento sea el esperado.





Ejemplos

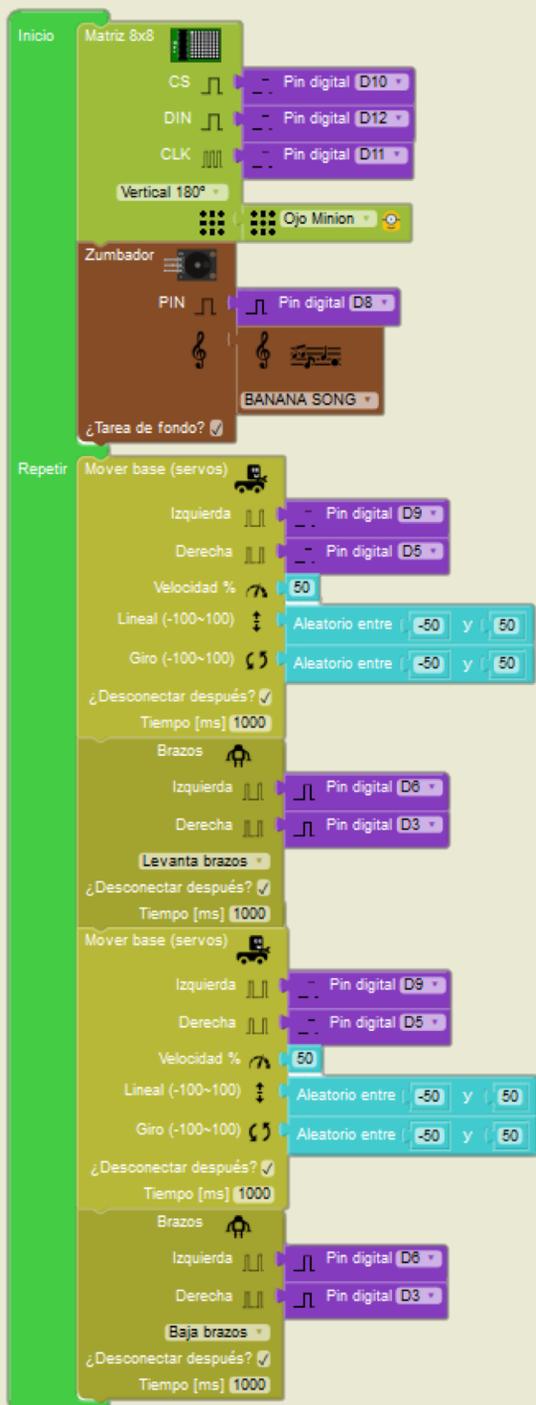
MOVER EL ROBOT A MISMO TIEMPO QUE SE REPRODUCE UNA CANCIÓN

Objetivos

Aprender a mover el robot con un comportamiento aleatorio. Reproducir una canción mientras se produce el movimiento y mostrar un ojo del Minion en la matriz de LEDs.

Comentarios

Para evitar consumir demasiada corriente los motores se desconectan al finalizar su movimiento. Al estar activada la tarea de fondo en la música, la canción se reproducirá mientras se realiza el movimiento, de lo contrario, el movimiento no comenzaría hasta que la canción haya finalizado. No se han incluido las instrucciones de calibración de la posición de los brazos por brevedad (ver ejemplo anterior).





Ejemplos

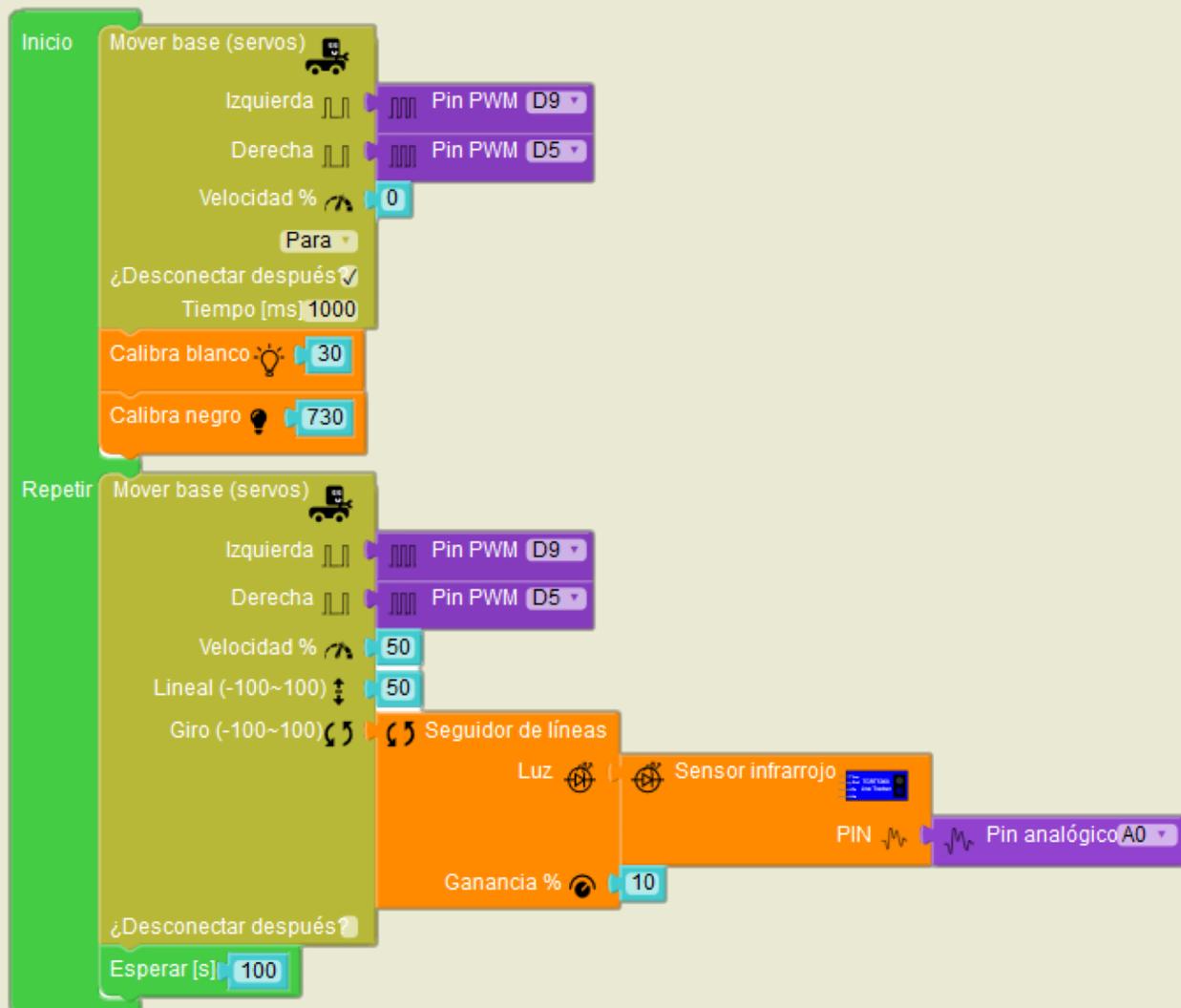
APLICACIÓN SEGUILÍNEAS

Objetivos

Aprender a seguir líneas utilizando el módulo seguilíneas TCRT5000.

Comentarios

Los valores de calibración de blanco y negro han sido medidos previamente (mostrando el valor por el puerto serie USB) y anotados para su posterior uso. Estos valores dependerán en gran medida de la posición del sensor con respecto al suelo y del color de la línea (contraste entre blanco y negro). El valor de ganancia debe ser ajustado para lograr el comportamiento deseado.





Ejemplos

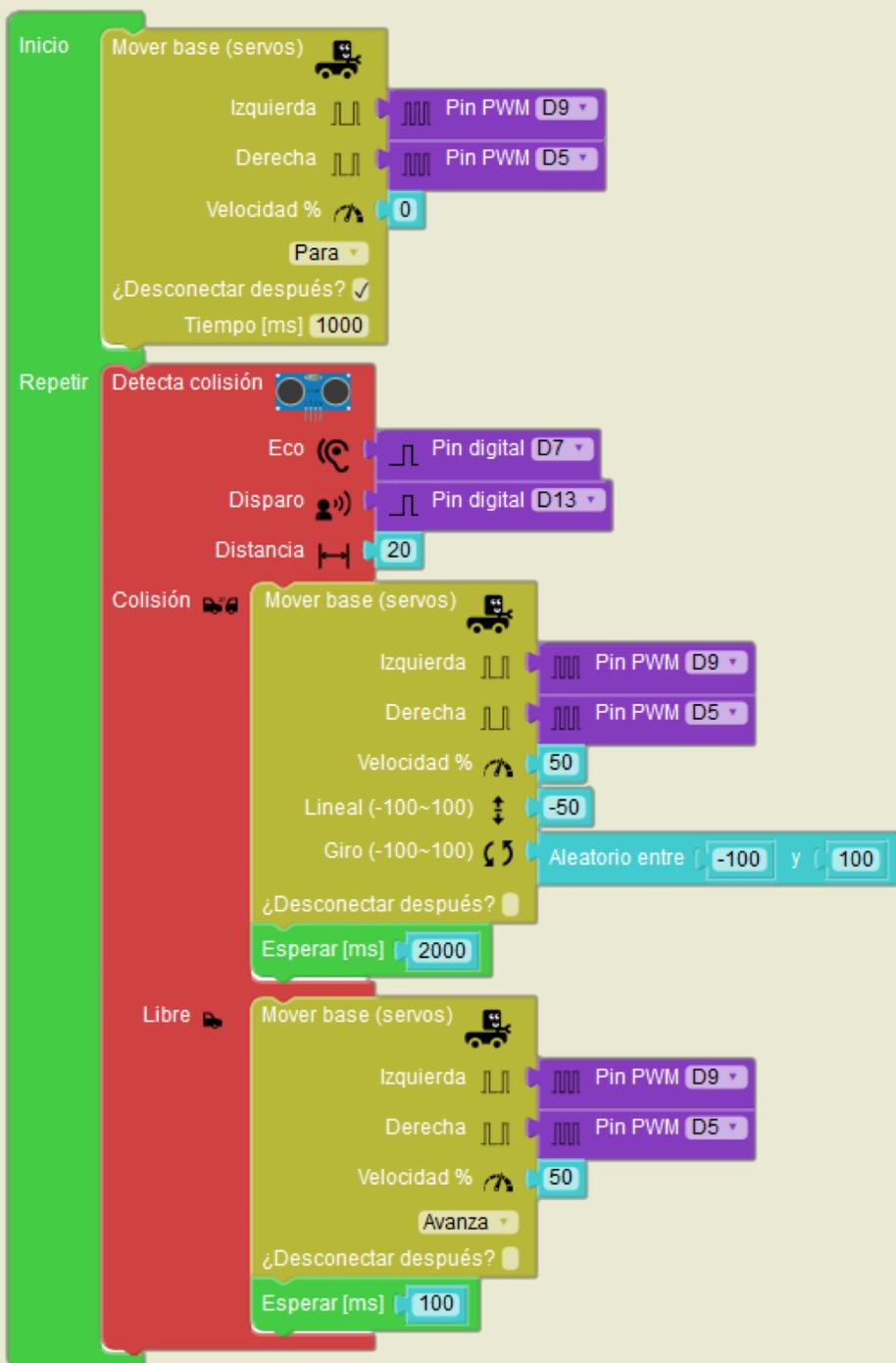
EVITACIÓN DE OBSTÁCULOS

Objetivos

Aprender evitar obstáculos a partir de la distancia de objeto más cercano obtenida con el sensor de ultrasonidos HC-SR04.

Comentarios

El robot avanza mientras no haya ningún obstáculo a menos de 20cm (supuestamente enfrente del robot). En esa situación está en un situación de 'Libre'. Si pasara a la situación de 'Colisión', en caso de detectar un obstáculo a menos de 20cm, entonces el robot gira aleatoriamente hacia la izquierda o derecha y espera un tiempo para que el giro sea lo suficientemente grande como para que el objeto ya no esté enfrente del robot.





Ejemplos

CONTROL REMOTO POR BLUETOOTH

Objetivos

Aprender a controlar remotamente el robot por Bluetooth.

Comentarios

Este código muestra cómo controlar el robot DYOR con la App de control remoto. Cada vez que se pulsa un botón de movimiento, se envía un comando que es recogido por la instrucción y ejecuta en función del caso el movimiento correspondiente. Completa el código usando la tabla de comandos adjunta integrando los ejemplos anteriores.

DYOR APP 



```

Inicio
  Define Bluetooth
  Tasa de baudios 9600
  RX Pin digital D2
  TX Pin digital D4
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  ¿Desconectar después?
  Repetir
  Bluetooth Recibir Comando
  comando Comando Para
  Repetir?
  hacer
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  ¿Desconectar después?
  comando Comando Avanza
  Repetir?
  hacer
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  Avanza
  ¿Desconectar después?
  comando Comando Retrocede
  Repetir?
  hacer
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  Retrocede
  ¿Desconectar después?
  comando Comando Gira derecha
  Repetir?
  hacer
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  Gira derecha
  ¿Desconectar después?
  comando Comando Gira izquierda
  Repetir?
  hacer
  Mover base (servos)
  Izquierda Pin digital D9
  Derecha Pin digital D5
  Velocidad % 50
  Gira izquierda
  ¿Desconectar después?
  
```

TABLA DE COMANDOS BLUETOOTH

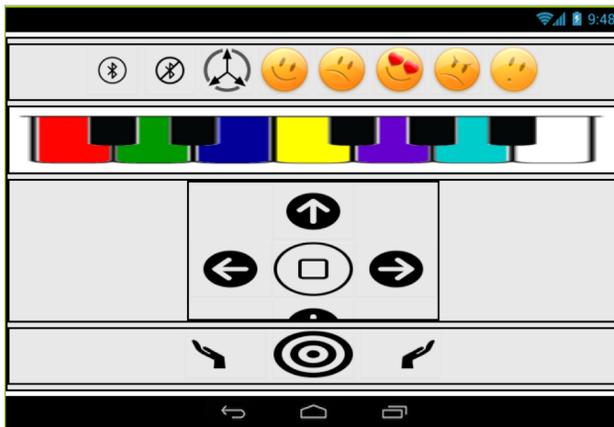
Comando		Comando	
Cancela repetición	0	Cabreado	1
Feliz	2	Triste	3
Enamorado	4	Sorprendido	5
Do	11	Re	12
Mi	13	Fa	14
Sol	15	La	16
Si	17	Avanza	21
Retrocede	22	Gira derecha	23
Gira Izquierda	24	Para	25
Cierra dedo Izq.	31	Abre dedo Izq.	32
Cierra dedo Der.	33	Cierra dedo Der.	34
Cierra pinzas	35	Abre pinzas	36
Evita obstáculos	41	Seguirlíneas	42
Negro	43	Blanco	44
Levanta Brazo Izq.	51	Baja Brazo Izq.	52
Levanta Brazo Der.	53	Baja Brazo Der.	54
Levanta Brazos	55	Baja Brazos	56



App Inventor 2

INTERFAZ DE USUARIO

Ahora vamos a crear una App para el control remoto del robot DYOR utilizando App Inventor2. Tu primer objetivo es acabar de completar la interfaz de usuario de una App que te proporcionamos para que tenga el aspecto que te mostramos.



Para ello debes:

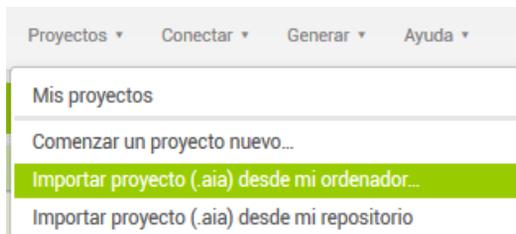
Descomprimir el siguiente fichero:

http://dyor.roboticafacil.es/ai2/aia/DYOR_vacio.zip

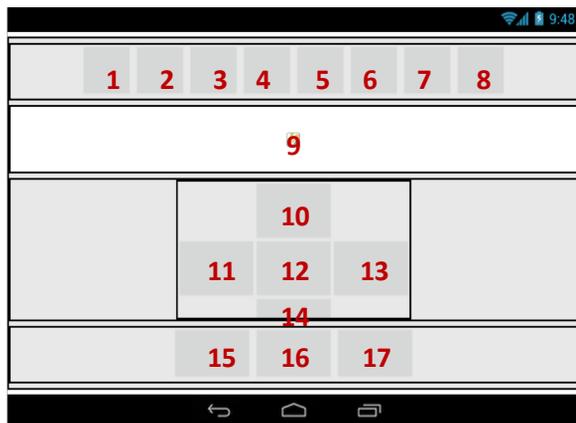
Iniciar sesión en App Inventor 2 (requiere una cuenta de Gmail) en:

<http://ai2.appinventor.mit.edu>

Descomprimir el fichero zip e importar el proyecto 'aia' desde el ordenador



En la App que te proporcionamos verás que hay un conjunto de componentes, pero ninguno de ellos tiene una imagen asociada. Debes subir las imágenes proporcionadas en el fichero zip al proyecto y asignar cada una de ellas según se indica en la tabla para que tenga el aspecto deseado. La columna UI de la tabla indica la posición del componente en la figura de la izquierda.



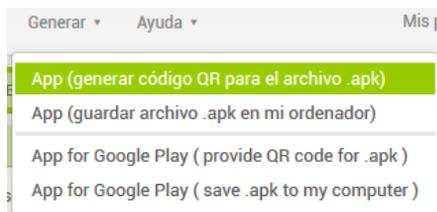
Componente	UI	Bluetooth	Componente	UI	Bluetooth
bluetooth_conn	1	-	bluetooth_disconn	2	-
imu_but	3	21-25	happy	4	2
sad	5	3	love	6	4
angry	7	1	stunned	8	5
Canvas1	9	11-17	up	10	21
left	11	24	stop	12	25
right	13	23	down	14	22
left_arm	15	51-52	right_arm	16	53-54
shoot	17	55-56			



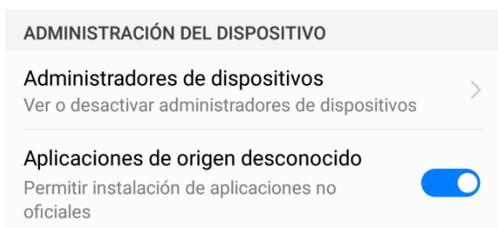
App Inventor 2

COMPILAR LA APP

Cada vez que quieras probar la App en tu dispositivo debes de generar un código QR con la App para poder instalártela.



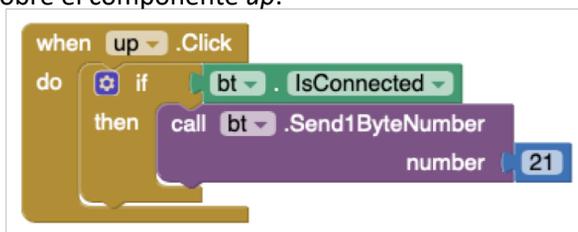
Utiliza un lector QR para descargar la App en tu móvil. Deberás permitir la instalación de aplicaciones de origen desconocido (dependiendo de la versión de Android estará en un lugar u otro, aquí se muestra un para la versión 7.0).



ENVIAR COMANDOS BLUETOOTH

De momento, en la App proporcionada, sólo funcionan los botones de conexión, desconexión y para avanzar. No es mucho, pero verás lo sencillo que es de completar.

En la vista de 'Blocks' tienes el código que ejecuta la App. Observa que se han incluido las instrucciones para el manejo de la conexión y desconexión del robot, así como un ejemplo del código que debes usar en el caso de hacer 'Click' sobre el componente *up*.



Ahora estás en disposición de rellenar parte del código. Debes, para cada componente del tipo botón, agregar un evento del tipo 'Click' y enviar el número asociado según se indica en la tabla de la página anterior (en la columna con el símbolo de bluetooth).

Por el momento los componentes *imu_but* y *Canvas1* no los utilizaremos.

En los componentes *left_arm*, *right_arm* y *shoot*, en vez de utilizar el evento 'Click', utilizaremos los eventos 'TouchDown' y 'TouchUp' para enviar el comando de levantar o bajar el brazo (el primer número de la columna corresponde a la acción de levantar, mientras que el segundo número corresponde a la acción de bajar).

Asegúrate que has completado los pasos para emparejar el dispositivo Bluetooth y que el robot dispone del código necesario para recibir los comandos por Bluetooth (como en el ejemplo de control remoto). En ese caso, ahora ya estás en disposición de enviar comandos Bluetooth y jugar con tu robot.

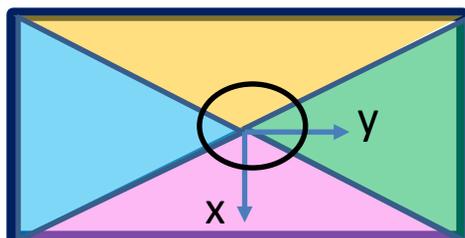


App Inventor 2

CONTROL REMOTO CON EL ACELERÓMETRO

El objetivo ahora es controlar el robot con el acelerómetro del dispositivo móvil. Esto es en realidad más sencillo de lo que parece, ya que debemos de usar las mediciones del acelerómetro para discriminar el movimiento deseado.

Si ponéis el móvil en horizontal apaisado, los ejes X e Y del acelerómetro están como se muestra en la figura:



Esto quiere decir que cuando inclinemos el móvil en una de las direcciones X o Y indicadas, se medirá una aceleración positiva correspondiente a una proporción de la gravedad. Cuando más inclinado esté el móvil, mayor será esta proporción.

Observa las regiones en las que hemos dividido la imagen anterior. La zona circular de en medio corresponde a la zona de parada, en la que si la magnitud de la gravedad en X e Y no supera una cierta cantidad, entonces quiere decir que el dispositivo móvil está prácticamente horizontal. Cuando lo inclinamos hacia adelante, aparece una componente X negativa fuerte, mientras que si lo inclinamos hacia detrás la componente X es claramente positiva. De forma similar valdría para las regiones de los lados, pero con la componente Y de la gravedad. El siguiente código discrimina estas regiones, complétalo para enviar el comando Bluetooth correspondiente (los códigos son los mismos que cuando apretábamos los botones):

```
when Clock1.Timer
do
  if get global imu
  then
    initialize local x to acc sensor.XAccel
    in initialize local y to acc sensor.YAccel
    in if square root (get x * get x + get y * get y) <= 3
    then Parar?
    else if (get y - get x >= 0 and get y + get x >= 0)
    then Derecha?
    else if (get y - get x <= 0 and get y + get x <= 0)
    then Izquierda?
    else if (get y - get x <= 0 and get y + get x >= 0)
    then Retroceder?
    else if (get y - get x >= 0 and get y + get x <= 0)
    then Avanzar?
    then
```

La variable 'global_imu' es una variable global que inicializaremos a **false**. La variable valdrá **true** cada vez que pulsemos sobre el botón *imu_but*. Cuando se pulse el botón *stop*, entonces volveremos el valor de esta variable a **false**.



App Inventor 2

TOCAR EL PIANO

El objetivo ahora es reproducir las notas musicales con el piano que hay dibujado en el componente Canvas1. Utilizaremos el evento 'Touched' de este componente para detectar que se ha tocado la imagen. La imagen del piano dispone de unos colores cuya paleta de colores coincide con la que utiliza App Inventor2.



Podemos obtener el color del píxel que se ha tocado y compararlo con los colores existentes de cada una de las teclas. El código que mostramos a continuación obtiene el color del píxel tocado sobre la imagen y lo compara con los colores de App Inventor2. Completar el código para que envíe los comandos Bluetooth de las notas musicales del Do al Si, según se indica en la tabla de comandos Bluetooth.

```
when Canvas1 .Touched
  x y touchedAnySprite
  do
    initialize local color to call Canvas1 .GetPixelColor
      x get x
      y get y
    in
      if get color = red
      then
      else if get color = green
      then
      else if get color = blue
      then
      else if get color = yellow
      then
      else if get color = purple
      then
      else if get color = cyan
      then
      else if get color = white
      then
```



SOLUCIONES



Si estás aquí es porque no has sido capaz de finalizar todas las actividades propuestas, lo cual lamentamos mucho.

Aún así, queremos proporcionarte las soluciones para el control remoto del robot, pero no te lo vamos a poner tan fácil. Debes decodificar la siguiente URL (las letras han sido modificadas):

`sggk://wbli.ilylgrxzuzxro.vh/zr2/zrz/WBLI_zinh.ark`

Pista

Sólo las letras de la a-z han sido codificadas (los números y los símbolos especiales como . : / _ no se han modificado. Se mantienen las mayúsculas y minúsculas.