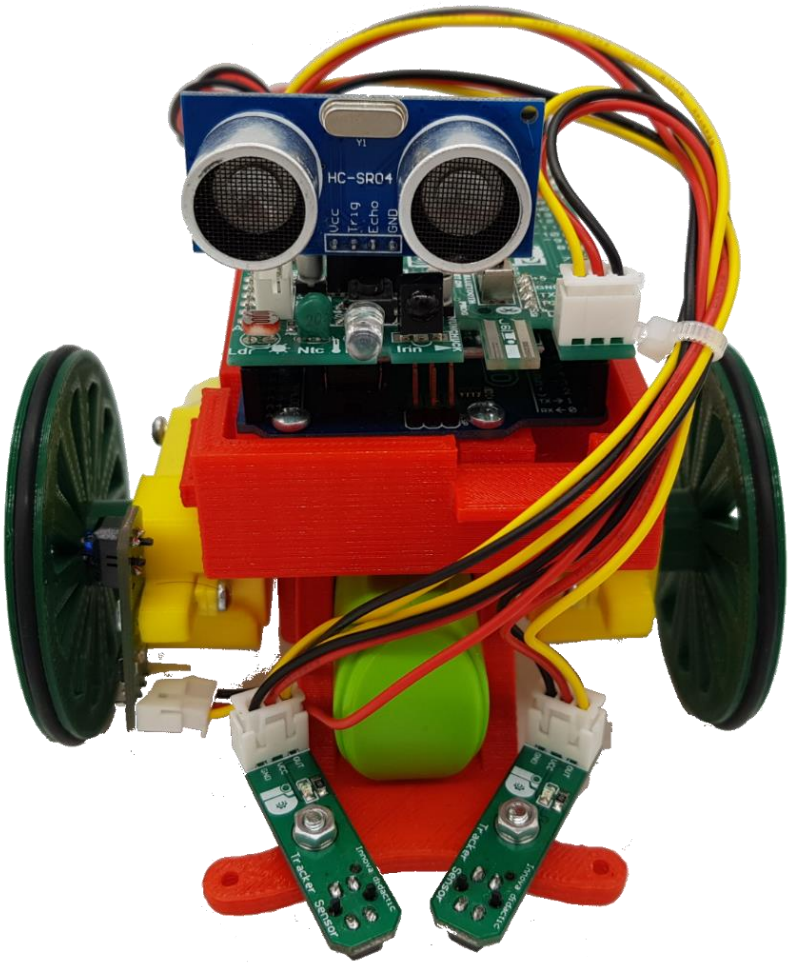
Manual de Actividades con ArduinoBlocks y el robot Imagina 3dBot Arduino (RBL0965)

Con la nueva Placa Imagina Arduino (Ref: RBL0692_v3)



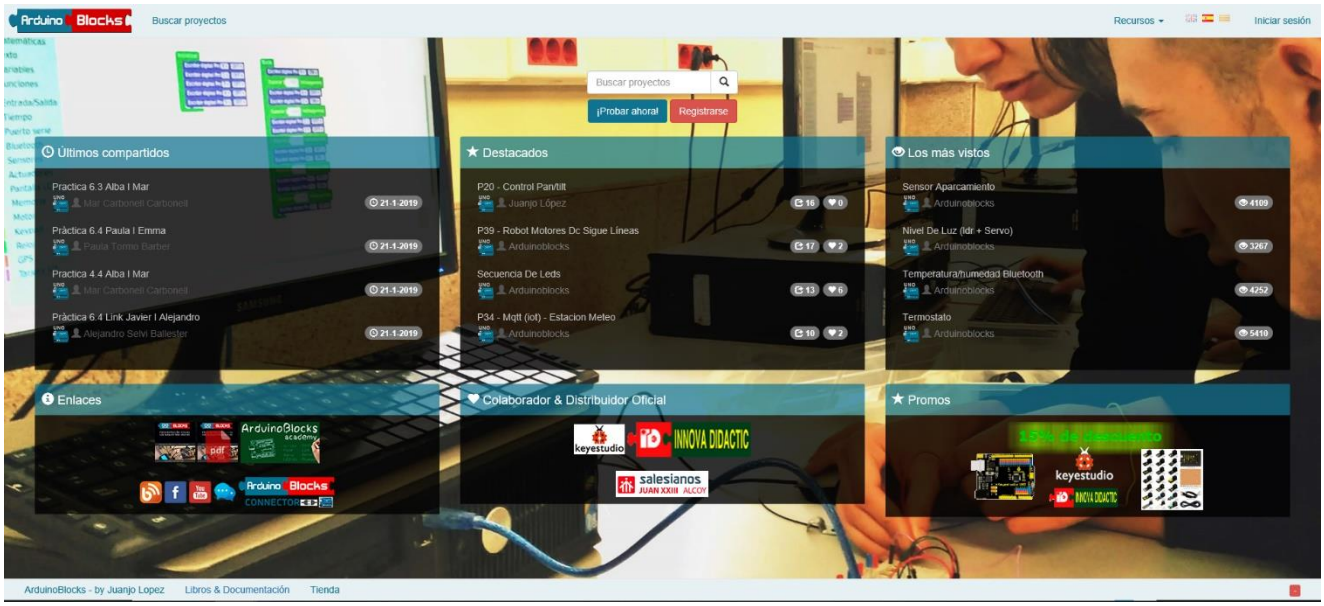
Índice

Introducción

En este manual os proponemos una serie de actividades relacionadas con la robótica educativa utilizando el robot Imagina 3dBot Arduino.

El objetivo de este manual es el de proporcionar unas actividades guiadas para aprender a programar de una manera entretenida y divertida.

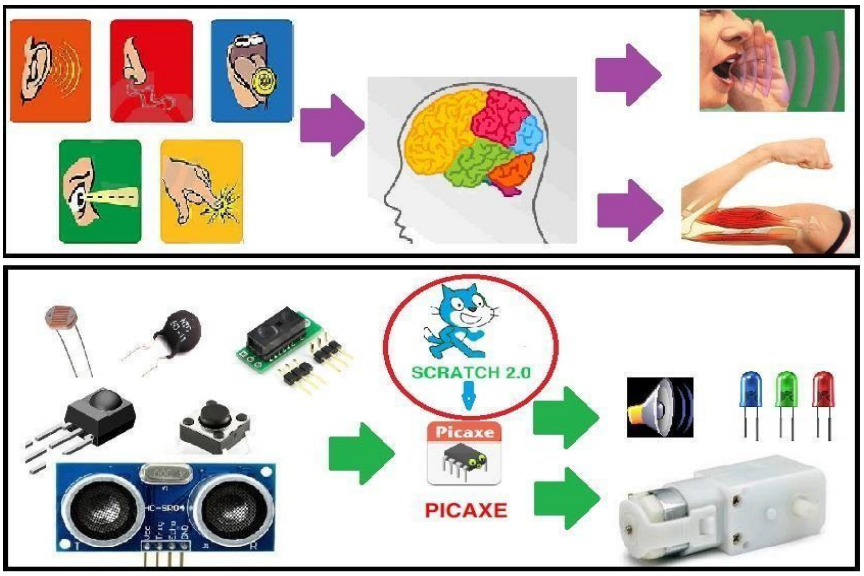
La placa Arduino Uno, placa en que está basado el robot Imagina 3dBot Arduino, se programa con un lenguaje de programación tipo texto que se llama “C++”, pero en este manual se explicará cómo hacerlo con ArduinoBlocks, un lenguaje de programación visual basado en bloques.



El portal es www.arduinoblocks.com.

¿Cómo funciona un robot?

Los robots funcionan de forma similar a nosotros. Cuando nuestro cerebro recibe información de los sentidos (oído, olfato, gusto, vista y tacto), analiza esta información y da estímulos a las cuerdas vocales para emitir sonido u órdenes a los músculos para que se muevan. Los 5 sentidos equivalen a entradas de información y, la voz y los músculos serían las salidas sonoras y motrices.



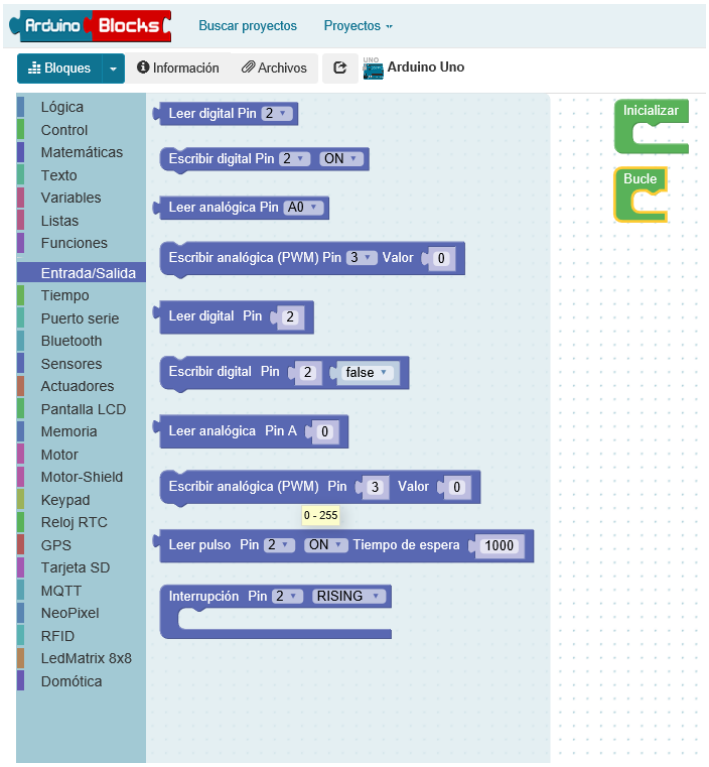
En el caso de un robot, un chip hace la función de cerebro. Este chip se llama microcontrolador y tiene unas entradas de información donde se conectan los sensores de luz (LDR), temperatura (NTC), sonido… y también tiene salidas, donde se conectan los motores, LEDs…

La diferencia principal es que, así como nuestro cerebro ya sabe lo que tiene que hacer porque lo ha aprendido a lo largo de nuestra vida a base de estímulos positivos y negativos, el robot tiene su memoria vacía, es decir, no sabe lo que debe hacer. Entonces nosotros tenemos que decirle como tiene que actuar en función de las señales que recibe de los sensores. ¡A esta acción se le llama programar!

ArduinoBlocks

ArduinoBlocks es un lenguaje de programación gráfico por “Bloques” creado por Juanjo López. Está pensado para que niños y niñas aprendan a programar con placas Arduino a partir de los XX años.

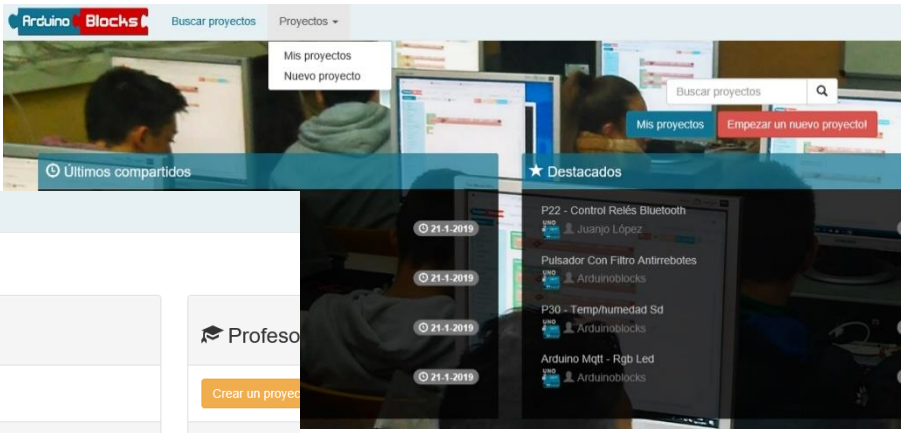
Los distintos bloques sirven para leer y escribir las distintas entradas y salidas de la placa, así como programar funciones lógicas, de control, etc.



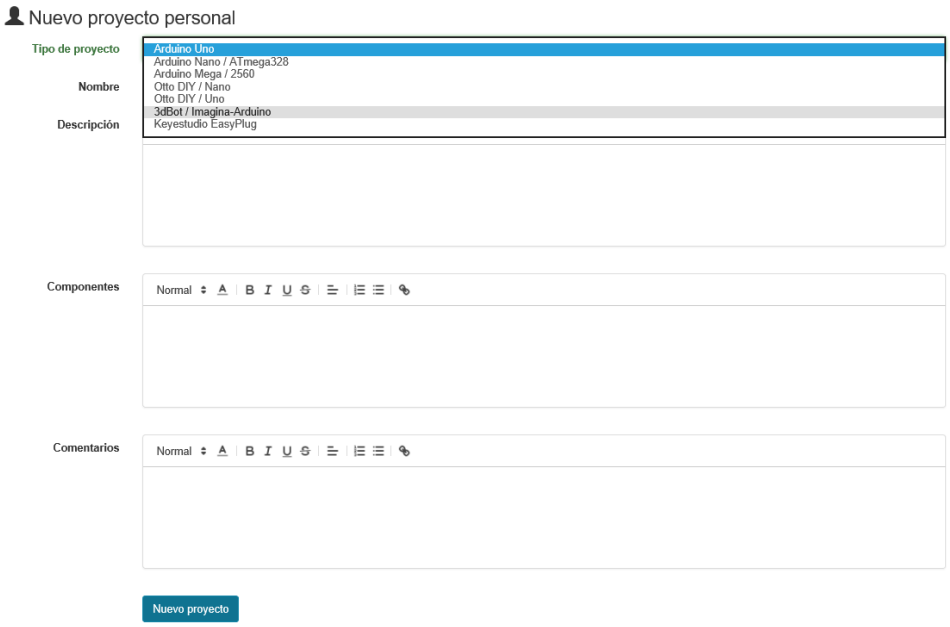
ArduinoBlocks y el robot Imagina 3dBot Arduino

En este manual usaremos un menú dedicado solo al Imagina 3dBot Arduino, con estos bloques conectaremos con las entradas y salidas de la placa Arduino Uno a través del Shield Imagina Arduino, que entre otras cosas permite adquirir valores de temperatura, luminosidad y controlar el printbot (robot impreso con tecnología 3D) llamado Imagina 3dBot.

Seleccionaremos “Nuevo proyecto” y “Iniciar un proyecto personal”.



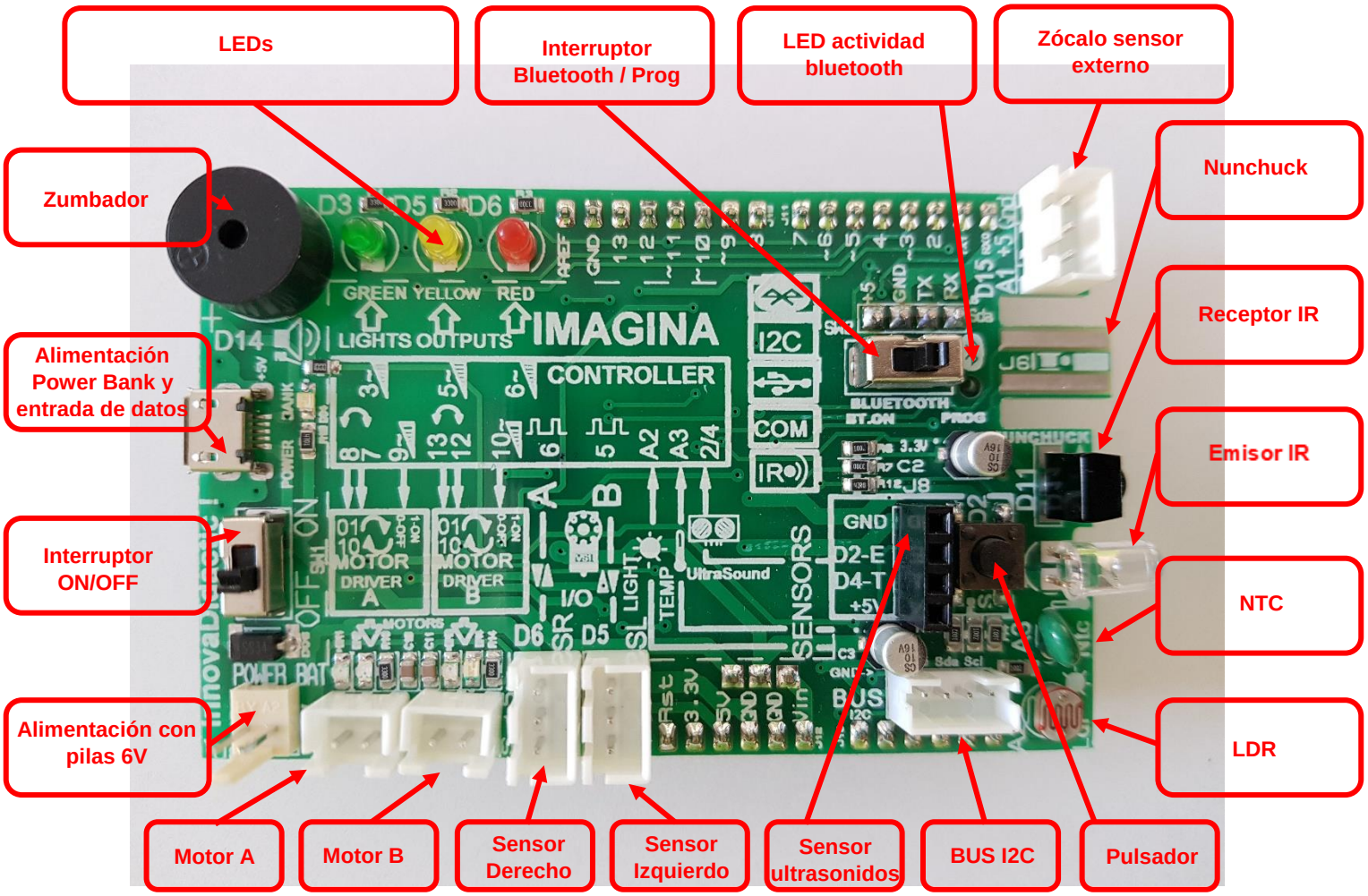
En el menú de “Tipo de proyecto” seleccionamos la opción “3dBot / Imagina-Arduino”.



Con el Imagina 3dBot Arduino trabajaremos de dos formas distintas:

En la primera parte de las prácticas, utilizaremos solo las placas Arduino, juntamente con el Shield Imagina Arduino, conectada al ordenador. Aprenderemos a utilizar las entradas y salidas, y los distintos sensores que podemos conectar.

En la siguiente fotografía se puede observar que disponemos de Entradas y Salidas para tomar datos del exterior a través de una gran variedad de sensores y actuar con distintos elementos según nos convenga.



Ubicación de componentes

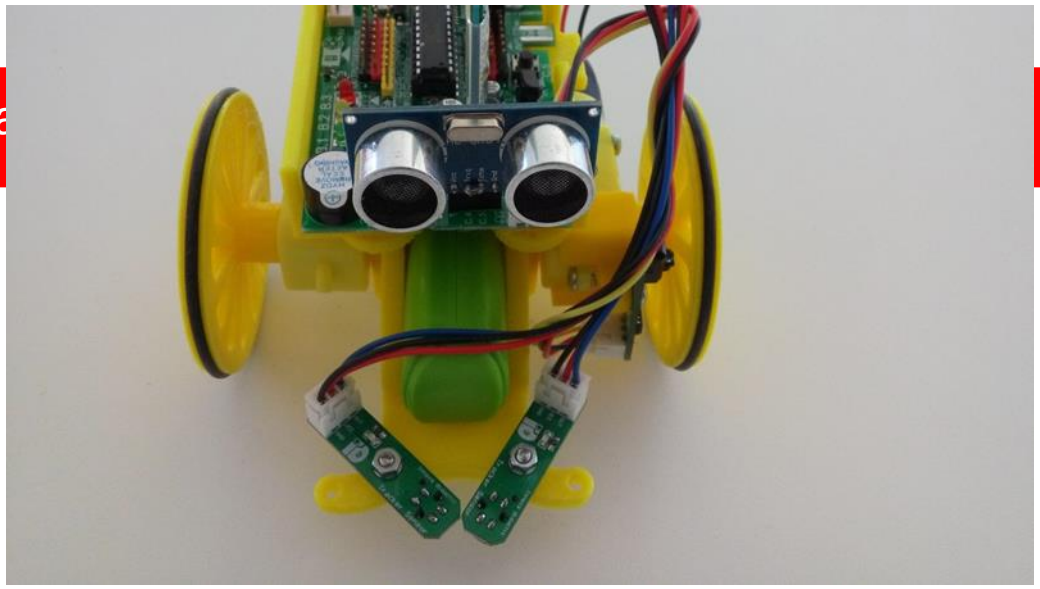
Además, tenemos pulsadores, leds y otros elementos ya incorporados para poder practicar sin necesidad de ningún montaje extra.

Tabla de entradas y salidas utilizadas

Entradas/Salidas Analógicas	Entradas / Salidas digitales
A0 Zumbador	D0 Bluetooth RX
A1 Sensor de línea fotoeléctrico, sensor PIR, servo o cualquier otro sensor	D1 Bluetooth TX
A2 Sensor de luz (resistencia LDR)	D2 Sensor Ultrasonidos Pulsador
A3 Sensor de temperatura (resistencia NTC)	D3 Diodo LED verde Emisor infrarrojo
A4	D4 Sensor Ultrasonidos
A5	D5 Diodo LED amarillo Sensor de línea Izquierdo
	D6 Diodo LED rojo Sensor de línea Derecho
	D7 Motor A
	D8 Motor A
	D9 Motor A
	D10 Motor B
BUS I2C / Nunchuk	D11 Receptor infrarrojo
	D12 Motor B
	D13 Motor B
	D14
	D15

En la segunda parte, con el robot totalmente montado; descubriremos como transferir el programa creado por nosotros al robot, para poder desconectar el cable de datos USB y que este actúe según las indicaciones que le hemos dado.

Materia



El Kit Imagina 3dBot Arduino básico incluye:

- 1 Chasis impreso en 3D compacto: 1 caja soporte para la placa, 2 soportes para sensores, 2 ruedas + 2 juntas tóricas y 1 bola de plástico

- 1 Placa de control Arduino Uno Rev3 con cable USB (Ref. A000066C).

- 1 Shield Imagina Arduino V3 Innova Didactic con Bluetooth (Ref. RBL0692_V3).

- 2 Motores con placa + cables (Ref. ID_MOTOR_PLACA).

- 3 Sensores infrarrojo siguelíneas (Ref. RBL030120).

- 3 Cables M-M de 20 cms. para conectar los sensores de línea (Ref. RBLSENSOR-PP).

- 1 Power-Bank 5V/2200 mAh (Ref. VL2200PB001GR).

- 1 Sensor de ultrasonidos de 4 pins (Ref. HC-SR04).

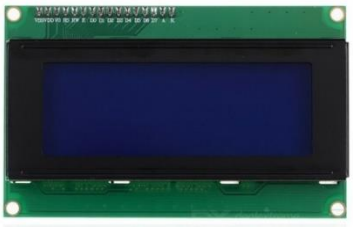
- 1 Mando para control remoto (Ref. KS9002).

- 1 Juego de tornillería compuesto por: DIN7981 2,9×25 (4 unidades) - DIN7981 2,9×6,5 (7 unidades) - DIN7985 M3x12 (2 unidades) - DIN934 M3 (2 unidades).

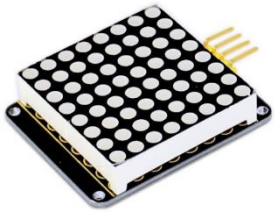
Material opcional

Pregunta a Innova Didactic para adquirir el material opcional.

- 1 Módulo pantalla LCD I2C LCD2004 (Ref. KS0062).



- 1 Matriz de LED 8x8 I2C HT16K33 (Ref. KS0064 o KS0396).



- 1 Sensor de presencia PIR (Ref. KS0052).

- 1 Servomotor (Ref. KS0194).

- Mando Nunchuk de la Wii (Ref. A3-12986).

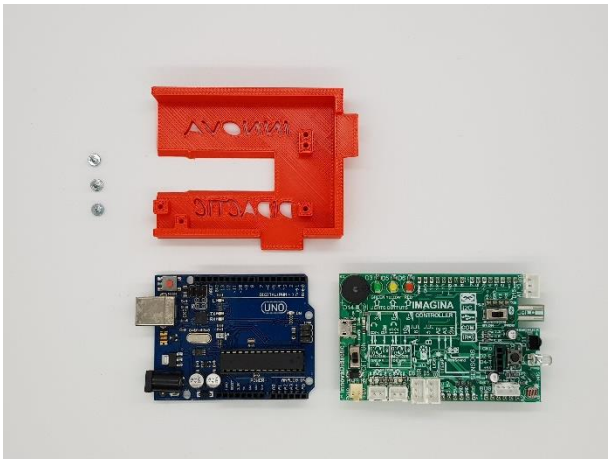
- Decenas de referencias para hacer volar la imaginación y crear cualquier idea!

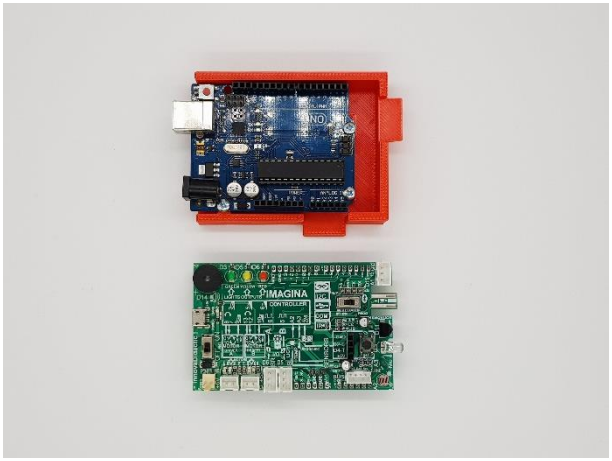
Herramientas

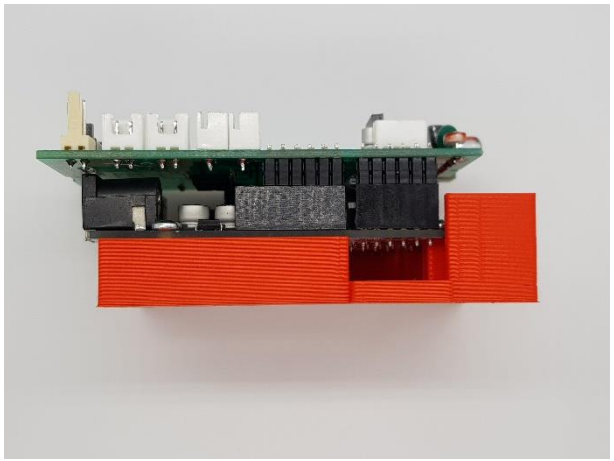
Para el montaje de este robot vamos a necesitar un destornillador estrella y una llave fija plana del número 6-7.

Preparativos: montaje de la placa

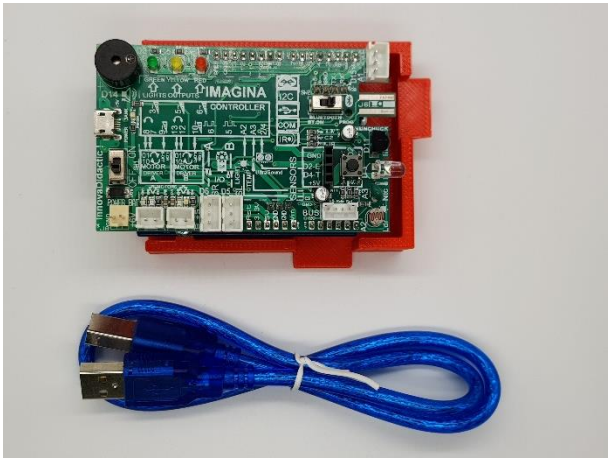
Para la primera parte de las prácticas, solo utilizaremos la placa o Shield Imagina Arduino. Montaremos la placa Arduino Uno Rev3 en la caja de soporte de la placa con 3 tornillos de estrella de 2,9x6,5mm, encima de esta, instalaremos el Shield Imagina Arduino y conectaremos la placa Arduino Uno Rev3 al ordenador con el cable USB.





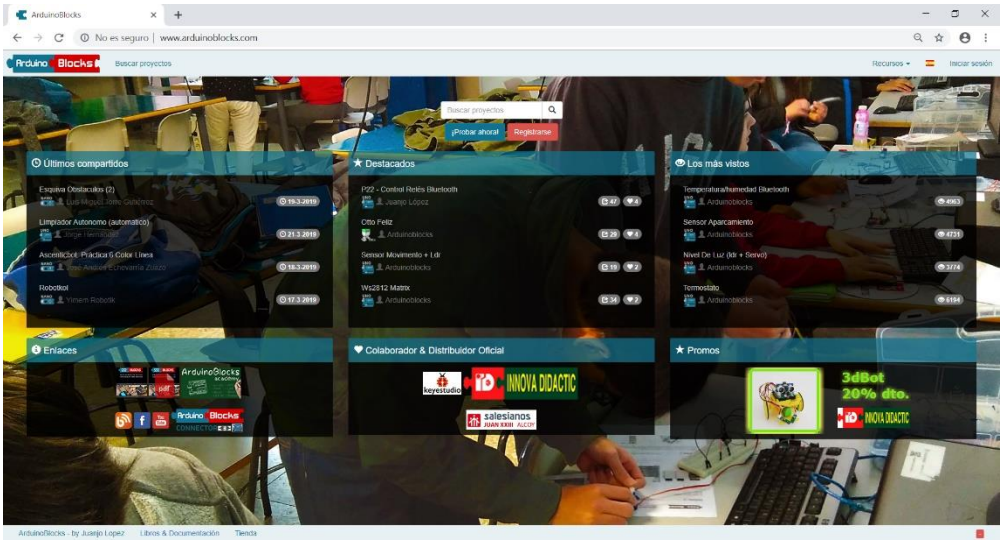


Solo con la placa podemos hacer cantidad de ejercicios para practicar e ir aprendiendo poco a poco.

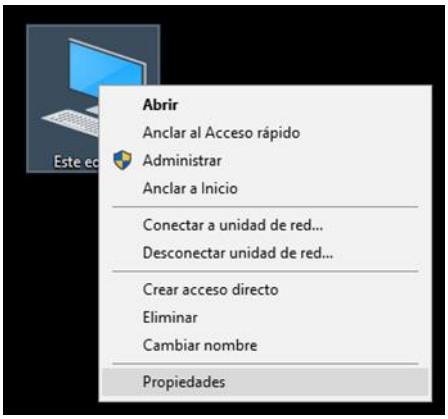


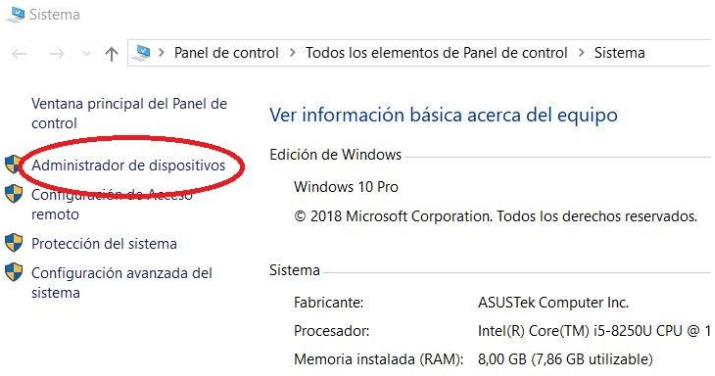
Preparativos: instalación de los drivers y programas

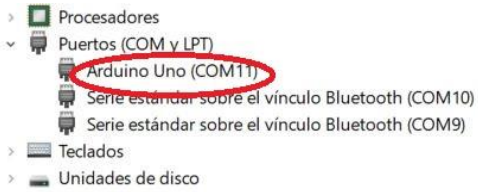
Para programar la placa Imagina Arduino a través de la placa Arduino Uno Rev3 con el programa online ArduinoBlocks, necesitaremos instalar un pequeño programa disponible en la sección de "Enlaces". Se trata de ArduinoBlocks Connector.



Una vez instalado podemos asegurarnos que se ha reconocido la placa, para esto con el botón derecho nos dirigiremos sobre el icono "Este equipo" y pulsaremos en "Propiedades". Nos dirigiremos a "Administrador de dispositivos" y observaremos si en "Puertos (COM y LPT)" aparece nuestra placa Arduino UNO con un (COM*XX*) disponible.

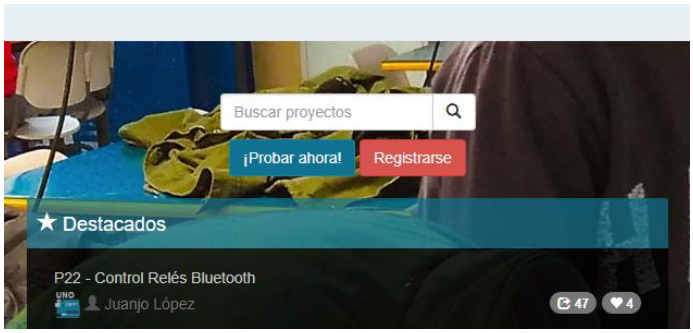




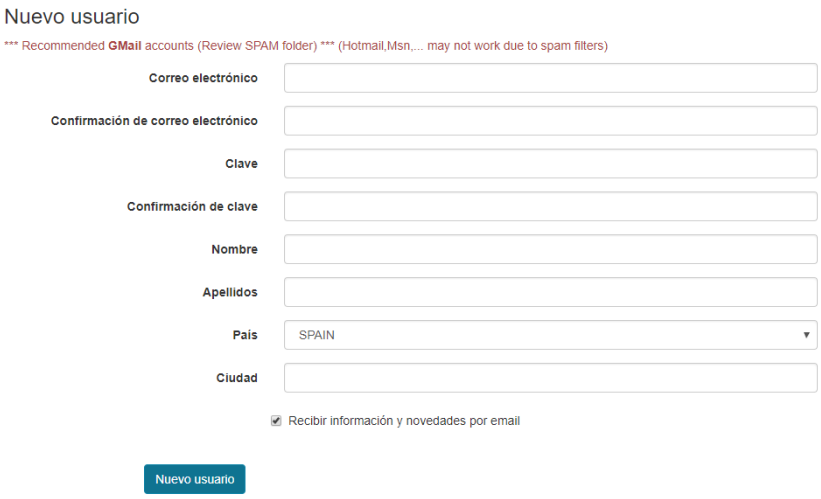


*En caso de que no nos reconociera la placa, como paso opcional podemos descargar e instalar el programa oficial Arduino IDE para que se instalen los drivers necesarios, ni tan si quiera es necesario ejecutarlo.

Programación Arduino con ArduinoBlocks

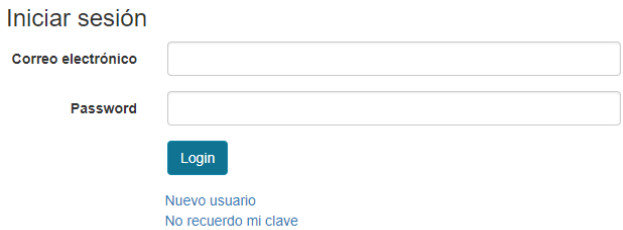


En el portal www.arduinoblocks.com, tan solo es necesario realizar un registro gratuito rellenando todos los campos y pulsando en “Nuevo usuario”.

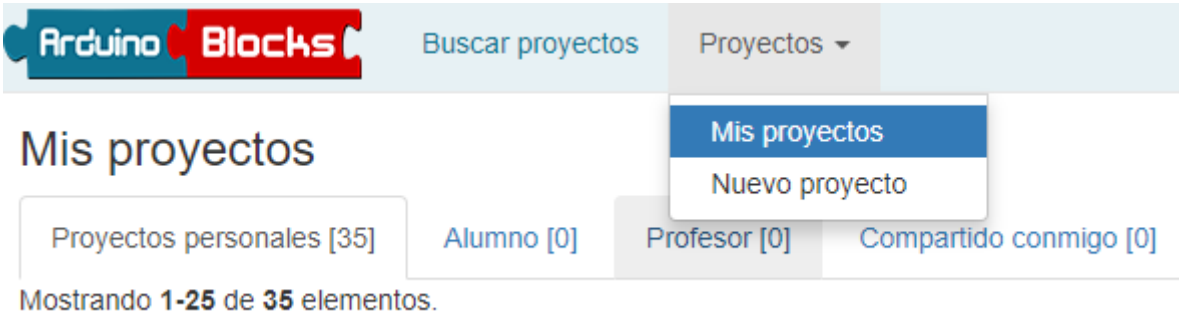


Será necesario validar la cuenta en el correo recibido en nuestra dirección de correo (sin no aparece en “Bandeja de entrada”, revisar la carpeta “Spam” o “Correo Basura”) y validar clicando sobre el link recibido.

Seguidamente ya podremos “Iniciar sesión”.

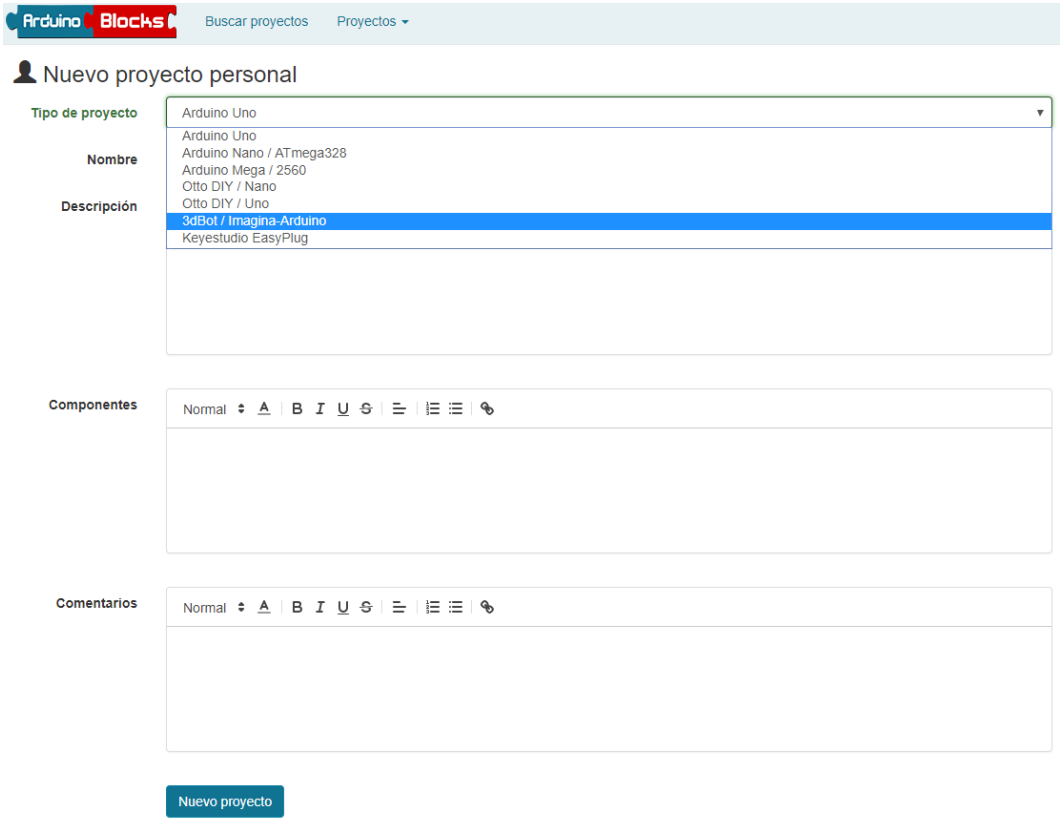


Para empezar a trabajar, nos dirigimos a “Proyectos” y seleccionamos “Nuevo proyecto”.

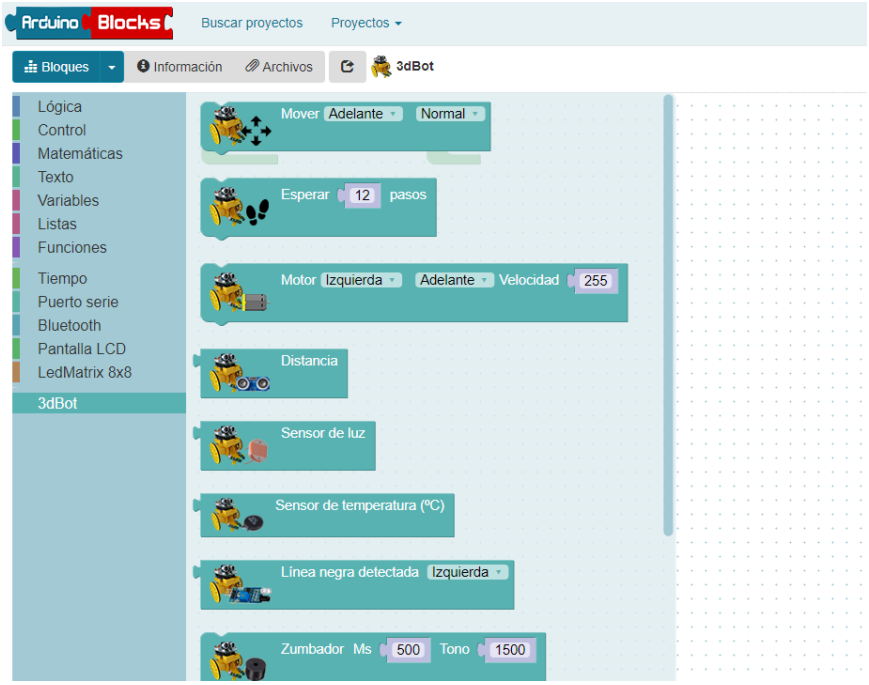


Escogemos “Proyecto personal”.

Y finalmente, ArduinoBlocks permite programar placas Arduino de varios tipos, y de realizar proyectos de todo tipo, pero para nuestro caso, en “Tipo de proyecto” seleccionamos 3dBot / Imagina-Arduino.



Esta opción nos presentará los menús necesarios para poder programar nuestro robot de forma fácil y sencilla con las funciones preparadas expresamente para el Imagina 3dBot Arduino.

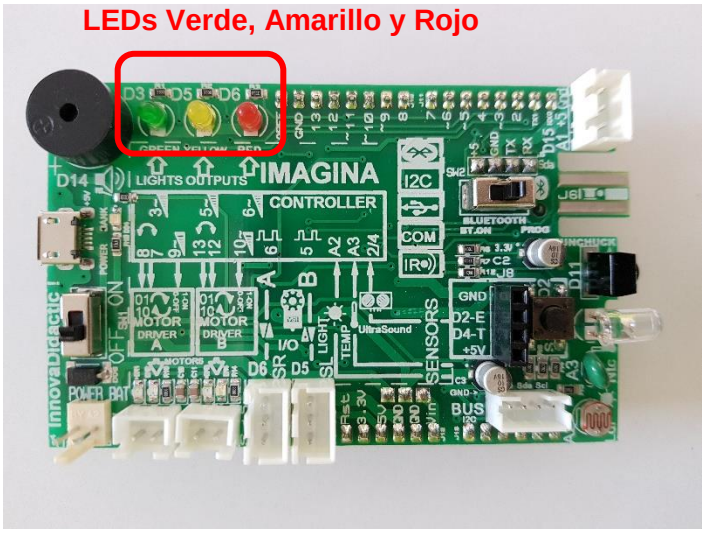


A continuación, vamos a aprender a usar las herramientas básicas para trabajar con nuestro robot.

Actividades con la placa Imagina Arduino

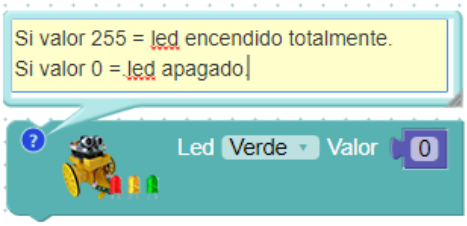
Os proponemos una serie de actividades para aprender, paso a paso, cómo funcionan los sensores de vuestro robot, incorporados en la placa, para más adelante montar el robot completo y darle "vida".

A01. – Encender un LED



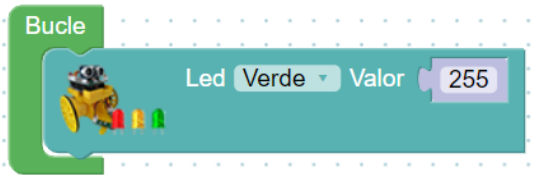
Activar un led como indicativo de que todo va bien:

En el apartado "3dBot" de ArduinoBlocks, encontramos el bloque "Led" que nos permite encender y apagar los leds verde, amarillo y rojo del robot.



El led se enciende totalmente introduciendo un 255 en la parte derecha del bloque. Para apagarlo basta con poner un 0. También se puede variar la intensidad de brillo. Para ello, cualquier valor entre 0 y 255 hará que el brillo del led aumente proporcionalmente hasta llegar a su máximo en 255.

A01_01 Encender mi primer led:



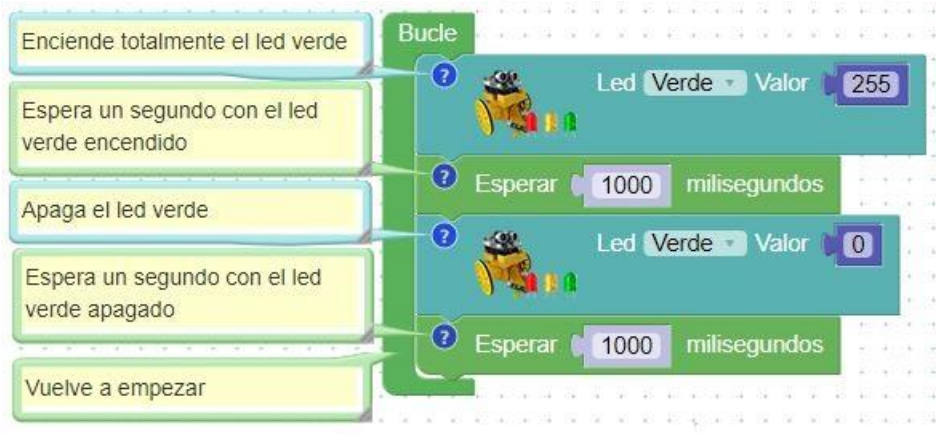
Tras cargar este programa, el led verde deberá permanecer encendido constantemente. Si es así, tienes bien conectada la placa y has conseguido cargar tu primer programa. ¡Seguimos!

A01_02 Juego de luces con el bloque “Esperar”.

El bloque “Esperar” lo podemos encontrar en el apartado “Tiempo”. Se corresponde con el famoso “Delay”. Lo que hacemos al insertar un bloque de “Esperar”, es dejar el robot en el estado en el que estaba justo antes de la espera, durante el tiempo que ésta dure.

Cuidado porque el tiempo usado en este bloque está en milisegundos, por lo que para esperar por ejemplo medio segundo, tendremos que introducir un 500.

Programa ejemplo:

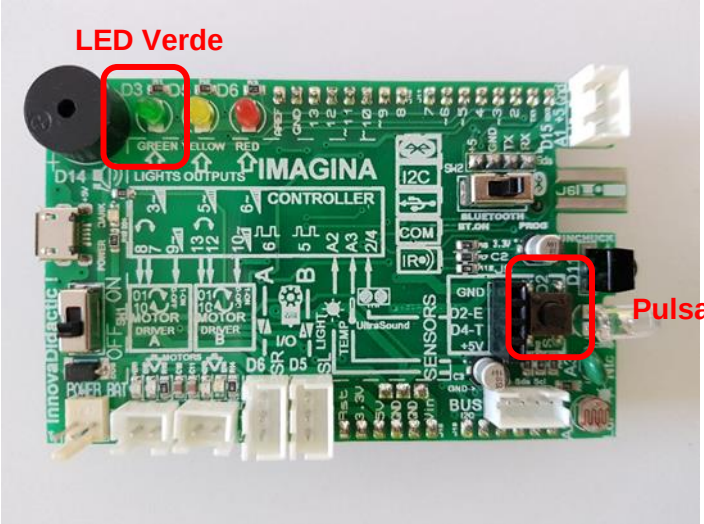


Con este programa conseguimos que el led verde parpadee con una frecuencia de 1 segundo de encendido y apagado. Puedes probar a cambiar tiempos e introducir más leds, así como a variar su intensidad de brillo.

El problema de la función “Esperar” es que deja la placa bloqueada en el estado anterior durante ese tiempo de espera, es decir, no recibe ni envía nuevas órdenes en ese tiempo. Es por esto que, en ciertos casos, es más conveniente utilizar el bloque “Ejecutar cada” que veremos en futuros apartados.

A02. – Encender un LED con el pulsador

En esta actividad queremos encender el LED verde al presionar el pulsador de la placa.

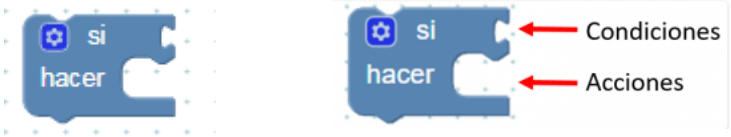


En este ejercicio ya empezamos a utilizar funciones del menú “Lógica” con las funciones de condición.

Condiciones “Si…hacer”.

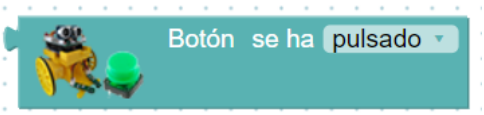
Se trata del famoso bucle *Si* (*if* en inglés) que es uno de los pilares de la programación, ya que permite evaluar estados y tomar decisiones en consecuencia.

Funciona como una oración condicional en español, si se cumple la condición incluida en su primer apartado, entonces se realiza la acción incluida en su segundo aparado. En caso contrario, no se hace nada.



En el apartado de condiciones se pueden introducir multitud de factores: estado de sensores, (analógicos o digitales), comparaciones, igualdades, operaciones matemáticas, etc.

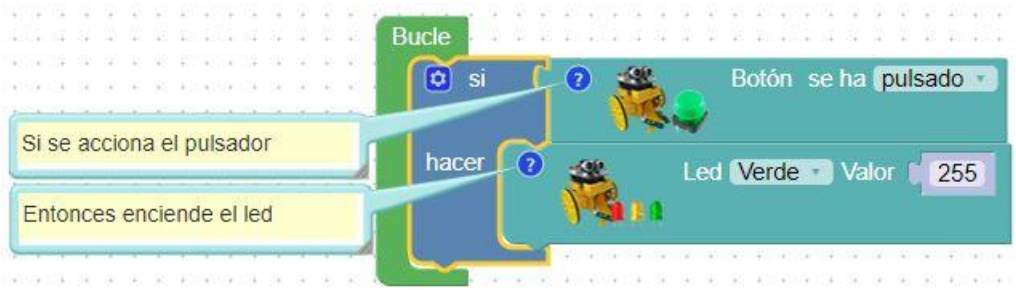
Para el programa ejemplo vamos a usar un nuevo bloque que controla el pulsador que incorpora la placa Imagina Arduino del Imagina 3dBot Arduino. Lo podemos encontrar en el apartado “3dBot”.



Con ese bloque crearemos la condición. Como acción a ejecutar si se cumple la condición, vamos a encender un led. Para ello volveremos a utilizar la función “Led”, cuyo bloque localizamos en la misma sección que el pulsador.

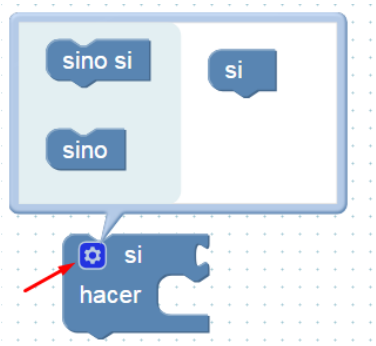
Programa A2_01:

En el siguiente programa hacemos que al pulsar el botón se encienda el led.

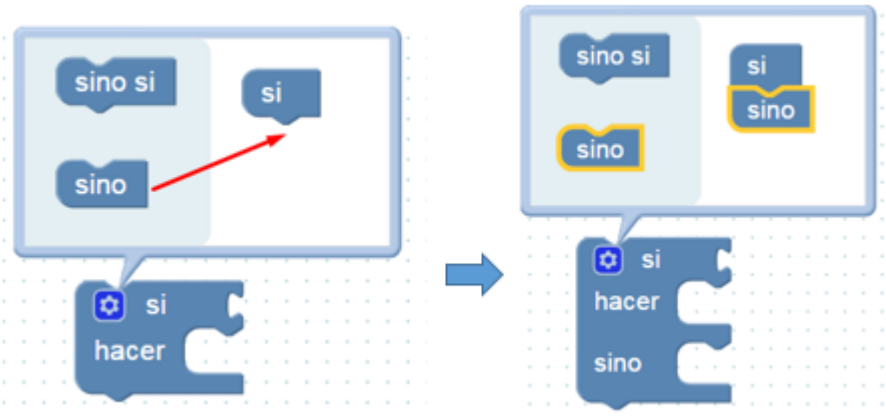


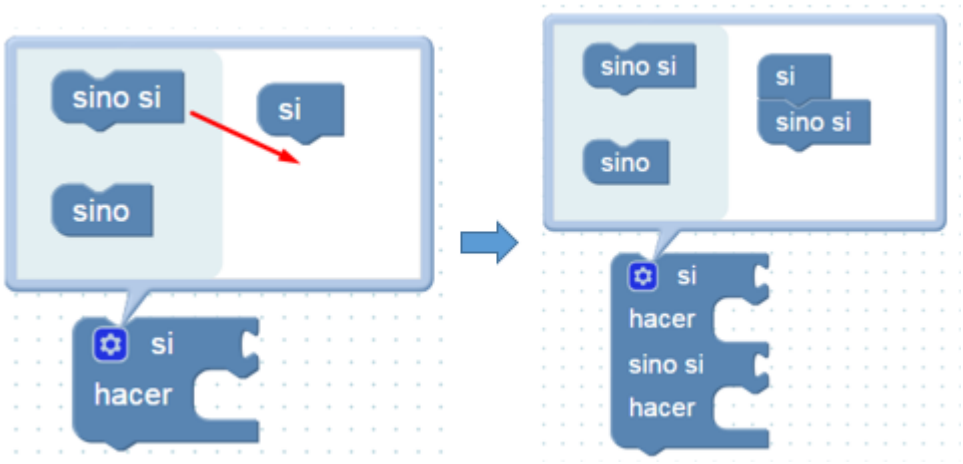
Ampliando el condicional.

Haciendo clic sobre el símbolo del engranaje señalado con la flecha roja en la imagen inferior, nos aparece un cuadro con funciones con las que podemos ampliar el condicional “Si”.



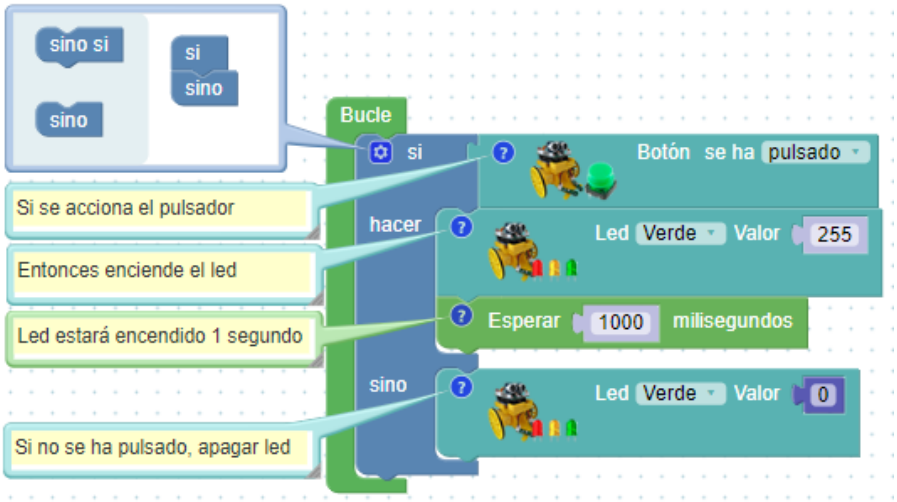
Hay dos opciones que se consiguen arrastrando los bloques como se aprecia en las siguientes imágenes:





Veremos ejemplos del uso de estas variantes a lo largo de diferentes programas en este documento. Pero siguiendo con el ejemplo del led, realizaremos un programa que al pulsar el botón se encenderá el led durante un segundo, si no se pulsa, se apagará.

A02_02 Ampliando el condicional.



Comprobad su funcionamiento.

A03. – Medir la luminosidad con LDR

Anteriormente hemos comentado que existe la función “Ejecutar cada”, aprovecharemos esta función junto con la función “Enviar” para ver los datos del sensor de luminosidad en la pantalla del ordenador.

Pero antes de hacer el ejercicio vamos a ver las distintas funciones poco a poco.

- Gestión de tiempos con el bloque “Ejecutar cada”.

Encontramos el bloque en la sección “Tiempo”. Este bloque ejecuta una vez cada *X* tiempo las órdenes que estén dentro de él. Cuidado porque NO detiene el programa en el estado anterior a él, como si hace el bloque “Esperar”.

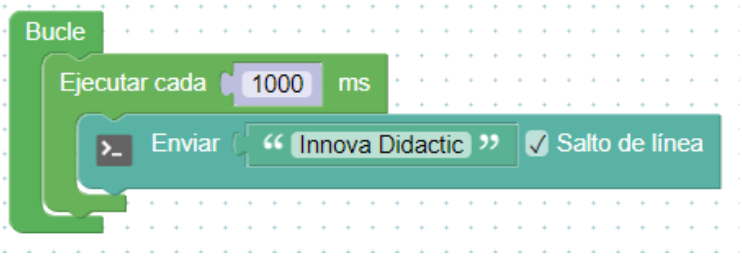
Junto con “Enviar”, es una función muy útil para imprimir datos en el ordenador. Vamos a verlo con un ejemplo en el siguiente apartado, en el que aprendemos a enviar datos desde el Imagina 3dBot Arduino al ordenador.

- Imprimir en el puerto serie (enviar datos al ordenador).

En la sección “Puerto serie” tenemos el bloque “Enviar”, mediante el que mandamos valores a la pantalla del ordenador.

Podemos enviar cualquier orden o dato que queramos solo con introducirlo en el espacio en blanco entre comillas del bloque.

Programa ejemplo:



Con este programa lo que hacemos es enviar a la pantalla del ordenador la palabra “Innova Didactic” una vez por segundo (1000 ms).

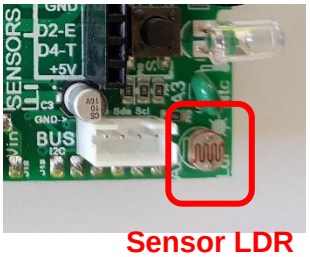
Tras cargar el programa, pulsando el botón “Consola” y a continuación “Conectar”, veremos en la pantalla del ordenador los valores enviados.

Hay que destacar que, si no usamos el bloque “Ejecutar cada”, enviamos al ordenador un dato cientos de veces por segundo, lo que satura la comunicación y bloquea el sistema. De ahí la importancia de este bloque.

También podríamos usar el bloque “Esperar” para enviar datos cada segundo, pero entonces, en los tiempos de espera, el robot no podría escuchar otras órdenes ni realizar otras acciones. Tranquilos, veremos todo esto más adelante con calma… ¡y con más ejemplos!

Ahora vamos a aplicar estos dos conceptos para leer los valores del sensor LDR incorporado en la placa.

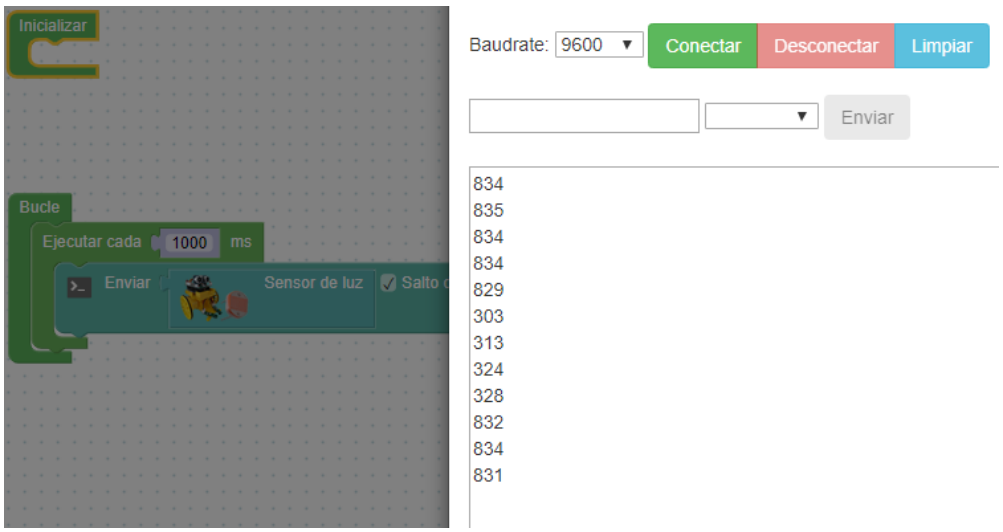
En esta actividad se pretende que cuando oscurezca se encienda el LED amarillo, como si de una sonda crepuscular se tratara.



Sensor LDR

Para hacerlo vamos a utilizar el bloque de la LDR (Resistencia Dependiente de la Luz) como dato para enviar a la pantalla del ordenador.

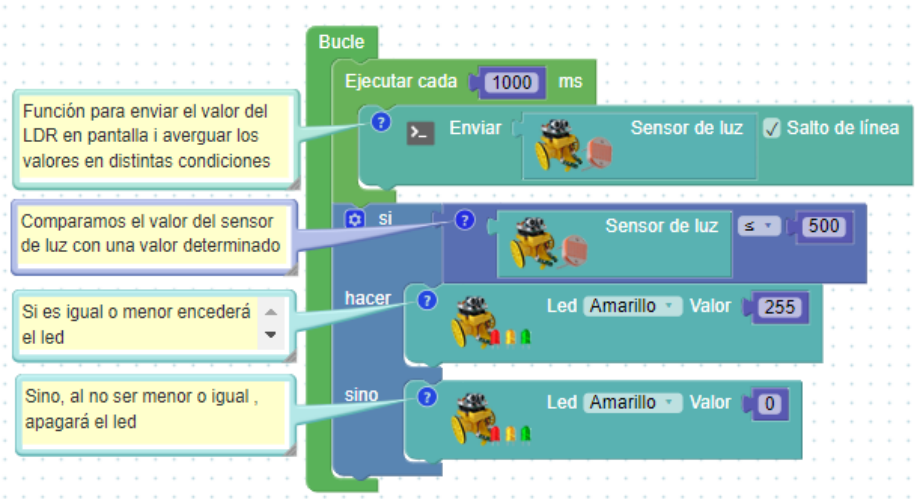
Introducimos el siguiente programa y abrimos la consola:



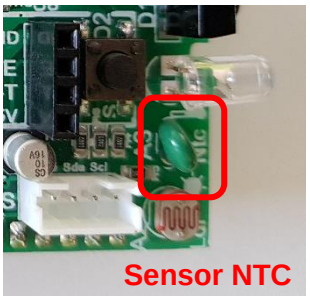
Comprobad su funcionamiento.

Vemos cómo cambia de valor si tapamos el sensor con el dedo y nos permite saber el valor con luz ambiental o a oscuras.

¿Cómo podríamos hacer para que se active el LED amarillo a partir de cierto nivel de luz?



A04. – Medir la temperatura con NTC



En esta práctica haremos un detector de incendios de forma que se deberá encender el LED Rojo cuando se supere un determinado valor del sensor de temperatura.

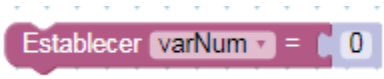
El sensor de temperatura que utilizamos es de tipo analógico y se llama NTC (Coeficiente de Temperatura Negativo).

Aprovecharemos para introducir el concepto de variables.

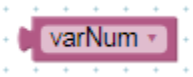
Las variables son elementos muy comunes en programación. Básicamente, crear una variable es darle un nombre a un dato o a una lectura. Por ejemplo, las mediciones de valores de temperatura las podemos guardar en una variable que se llame “Temperatura”. No es obligatorio su uso, pero nos permiten trabajar más cómodamente, además, como podemos personalizar su nombre, ayudan a clarificar el código y utilizar un lenguaje más natural.

Al trabajar con variables vamos a tener dos tipos de bloques principales:

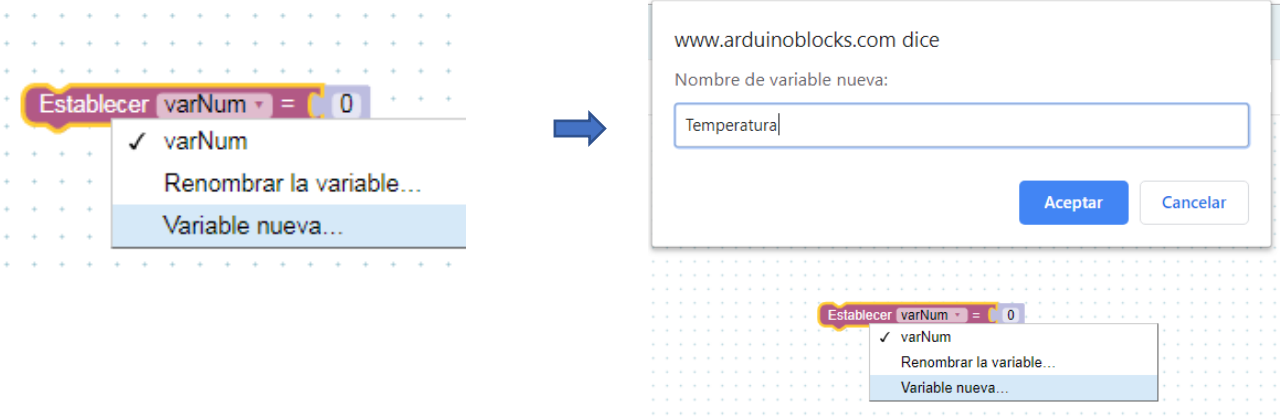
1. El bloque en el que le damos valor a la variable:



2. Y el bloque de la propia variable creada, para poder insertarla y combinarla con otros bloques:



También podemos personalizar el nombre de la variable, de la siguiente forma:

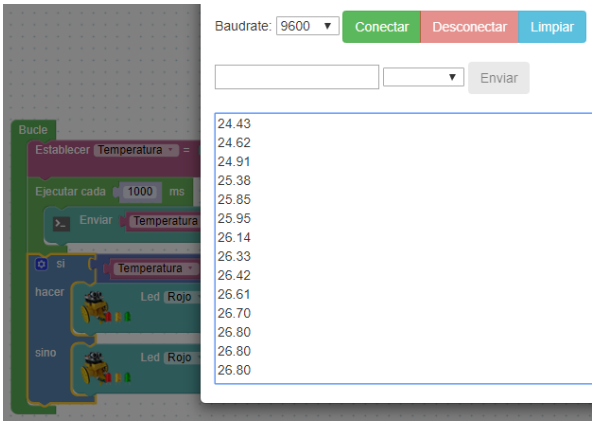


Una vez creada la nueva variable, podemos seleccionarla haciendo clic sobre el desplegable:

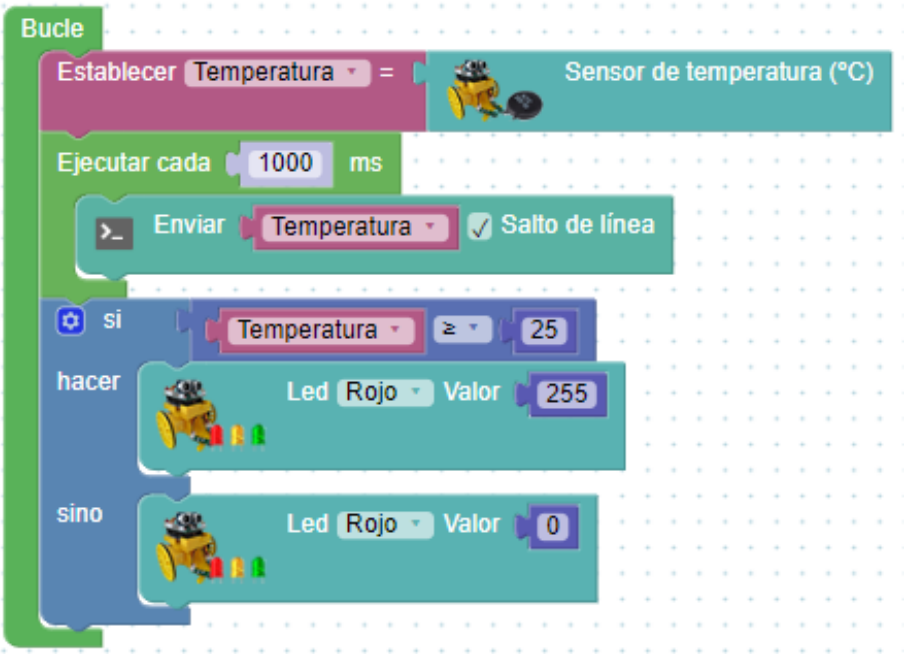
Ten en cuenta que las variables solo pueden estar formadas por una palabra. Si quieres incluir varias palabras, puedes usar el truco de separarlas con una barra baja "_", como en el ejemplo, Valor_Temperatura.

Usaremos el mismo procedimiento para descubrir el valor de la Temperatura como hemos hecho con el valor de Luminosidad.

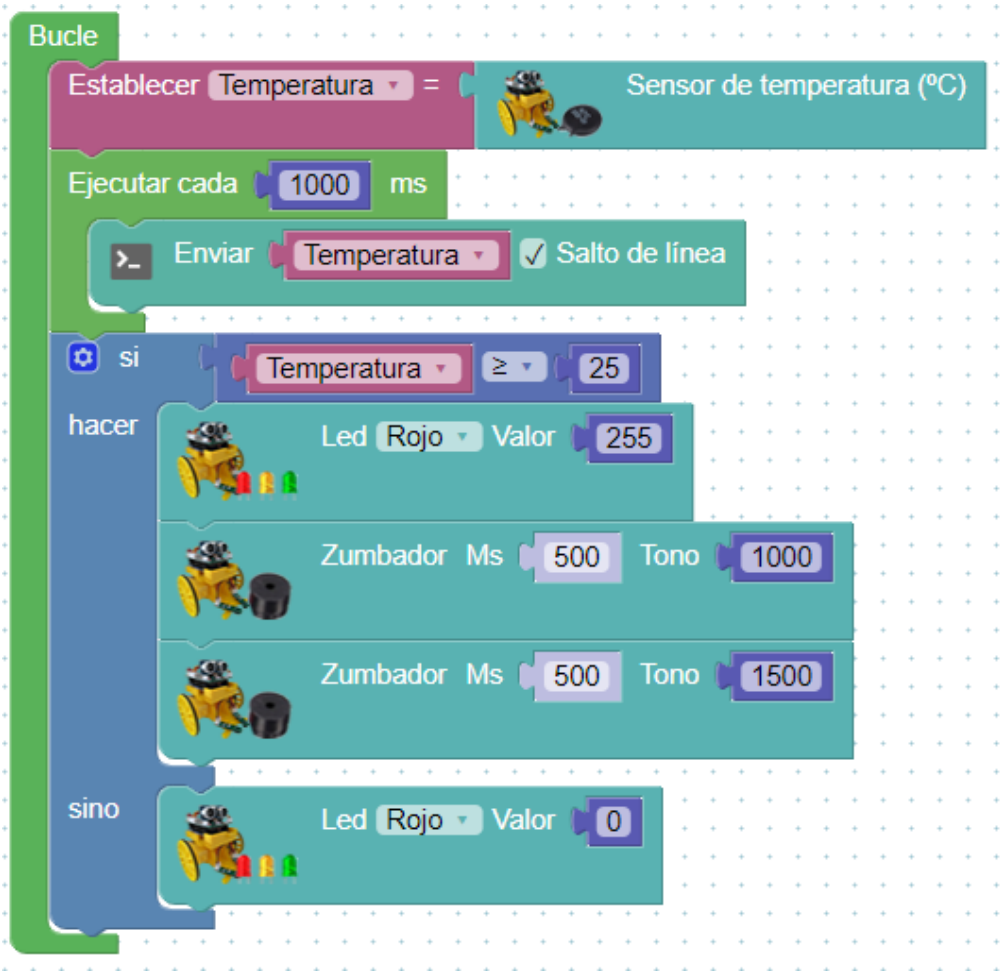
En el caso que se adjunta, vemos que el valor ambiental es sobre el 24 y cogiendo el sensor de temperatura con los dedos, sube más de 25.



Así que ya estamos listos para hacer nuestro detector de incendios, se activará el led cuando sobrepasemos un valor determinado.



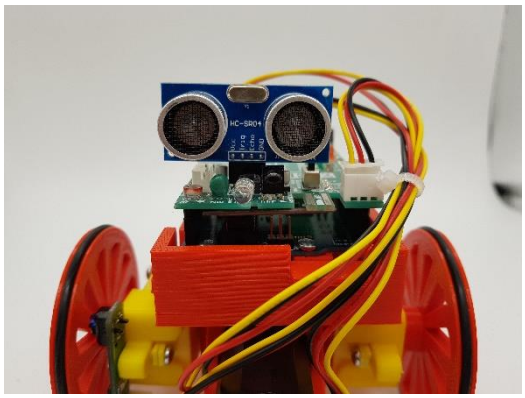
¿Añadimos una sirena?



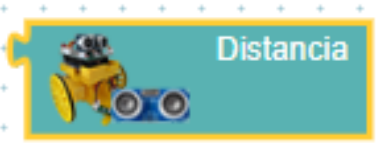
A05. – Mide distancias (sensor ultrasonidos HC-SR04)

Si disponemos de un sensor de ultrasonidos HC-SR04, podremos realizar medidas de distancia directamente con ArduinoBlocks.

El funcionamiento de los sensores de ultrasonidos consiste en emitir un sonido y medir el tiempo que tarda en rebotar con el objeto que tiene delante y volver. Así se puede calcular la distancia a la que se encuentra el objeto que está enfrente.



Para a realizar esta práctica lo conectaremos directamente al zócalo delantero tal como en las imágenes adjuntas:



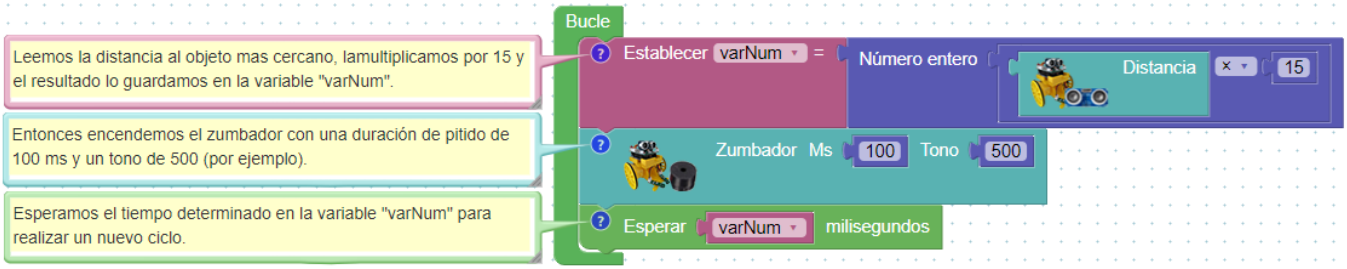
Con la instrucción “Distancia”, utilizando las variables y la opción de enviar datos a la pantalla del ordenador realizaremos un programa donde podremos jugar con la distancia.

A05_01 El programa seria el del ejemplo, y con la consola vamos a ver las lecturas.

A05_02 Ahora, vamos a hacer como si se tratara de un sensor de aparcamiento de un coche.

El robot va a emitir un pitido intermitente cuya frecuencia dependerá de la distancia a la que esté un objeto que se coloque delante. Cuanto más cerca esté, más veces por segundo pitará.

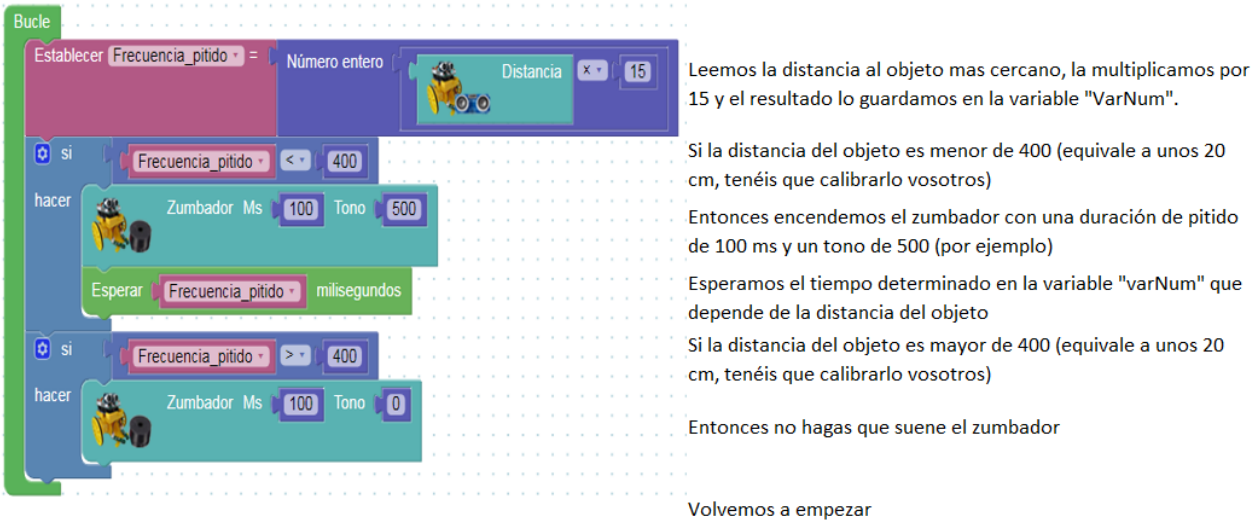
Hay que tener en cuenta que el tiempo que pasa entre pitido y pitido (que depende de la distancia) se indica en milisegundos, por eso, multiplicamos por un factor (en el ejemplo, 15) que hace que la frecuencia de pitido se adapte a nuestras necesidades. Podéis probar el sistema, variarlo, escalarlo y adaptarlo a vuestro gusto.



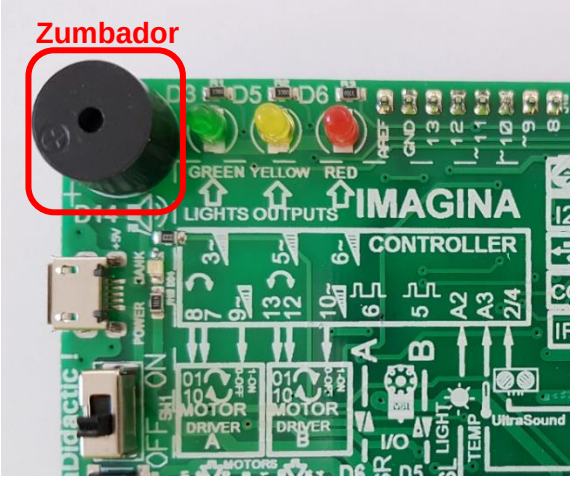
Vemos en el programa que la frecuencia con la que se realiza cada ciclo, que equivale a la frecuencia con la que damos cada pitido, depende directamente de la distancia del 3dBot al objeto.

A05_03: Sensor de distancias

En el programa anterior hay un inconveniente, y es que el robot siempre va a pitar, aunque el objeto que esté delante se encuentre a mucha distancia. Ahora vamos a poner una nueva condición para que este pitido solo comience cuando el objeto esté a una determinada distancia mínima. Para el ejemplo que se propone, se ha probado con 20cm, pero, igual que antes, podéis ajustarlo como queráis.

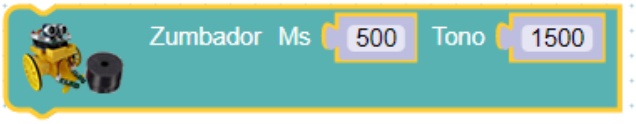


A06. – Generador de notas musicales con el zumbador



El zumbador es el pequeño “altavoz” situado a la esquina de la placa, al lado de los leds, y con él, podemos generar melodías musicales.

El bloque correspondiente es este:

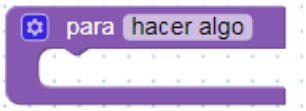


Vamos a crear un programa con la escala des de Do:

A07. – Funciones

Una función es simplemente un conjunto de instrucciones a las que damos un nombre, para no tener que repetirlas en diferentes partes del programa y para clarificar y ordenar el código. Al crear una función, se genera automáticamente un bloque de dicha función, que ya puede ser insertado en cualquier parte del programa.

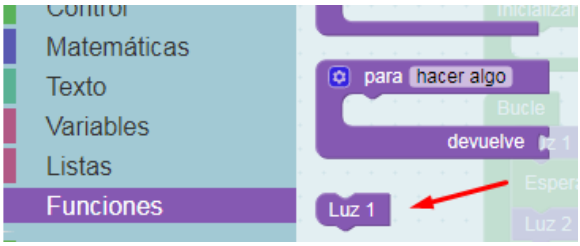
Para ello, tenemos que usar el bloque “para” que encontramos en el apartado “Funciones” de ArduinoBlocks.



Primero tenemos que darle un nombre a la función, e incluir dentro de la misma, el conjunto de instrucciones que queremos que realice el robot cada vez que la usemos.

Por ejemplo, podemos definir una función con el nombre “Luz 1” cuya finalidad sea encender el led verde y apagar el amarillo y el rojo.

Si ahora vamos a la sección “Funciones” del panel izquierdo de ArduinoBlocks, encontraremos ya la nueva función creada, disponible para ser seleccionada e insertada en cualquier parte del código.



Es importante destacar que las funciones no se definen dentro del “Bucle”. Se crean fuera y luego se insertan dentro en los momentos en los que queramos que se ejecuten.

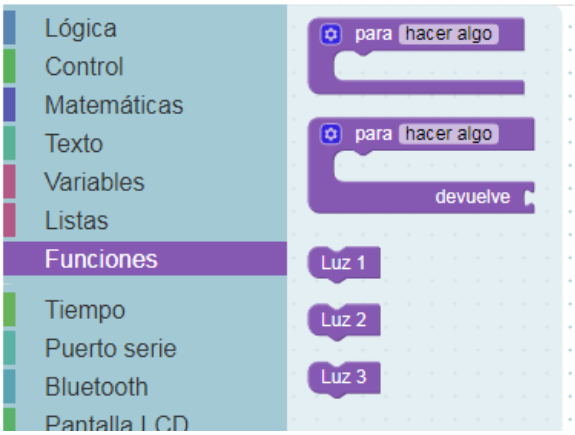
Cuando creamos un programa pequeño (de pocos bloques), no suele merecer la pena definir funciones, ya que no ahorra mucho tiempo. Pero cuando realizamos programas más extensos o con partes iguales que se repiten, es una buena práctica hacer uso de ellas, tanto por comodidad, como por claridad del programa realizado.

A07_01 Juegos de luces.

En el siguiente programa se definen tres funciones creando juegos de luces con los tres leds del 3dBot, y a continuación se realiza una secuenciación y repetición de estas.

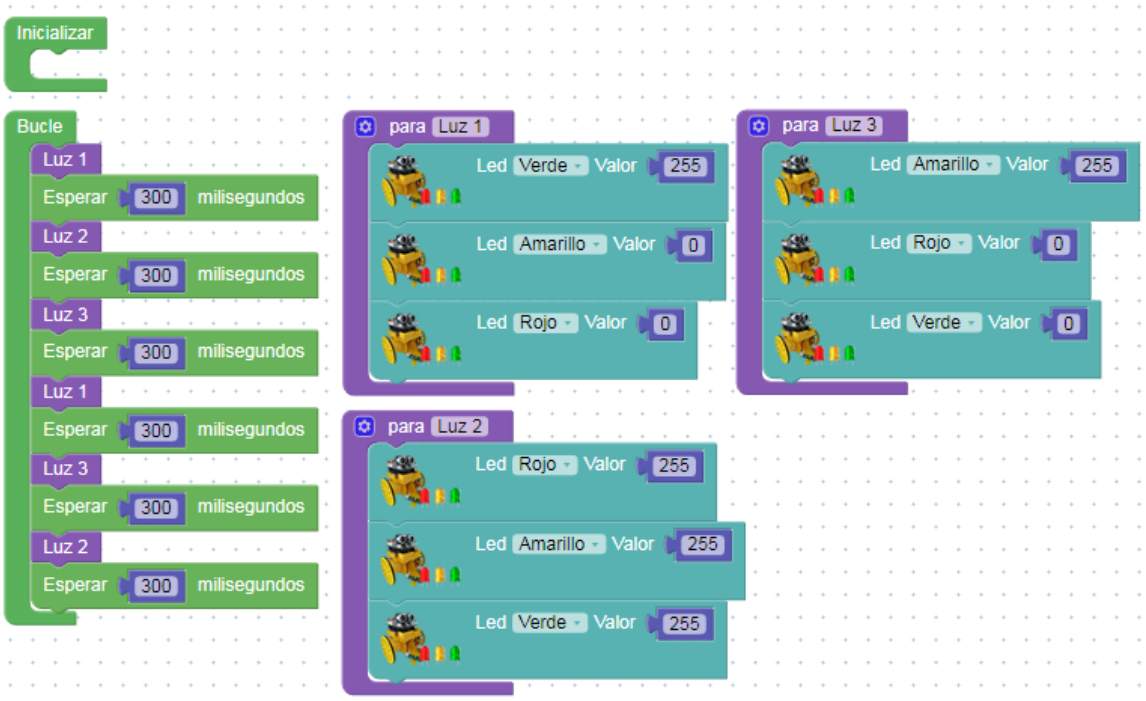
- Las funciones, “Luz 1”, “Luz 2” y “Luz 3”, son:

Tras crear las funciones, ya podemos encontrarlas en el a apartado “Funciones”, como bloques, para insertarlas en el “Bucle”:



Y la “Inicialización” y el “Bucle” quedan:

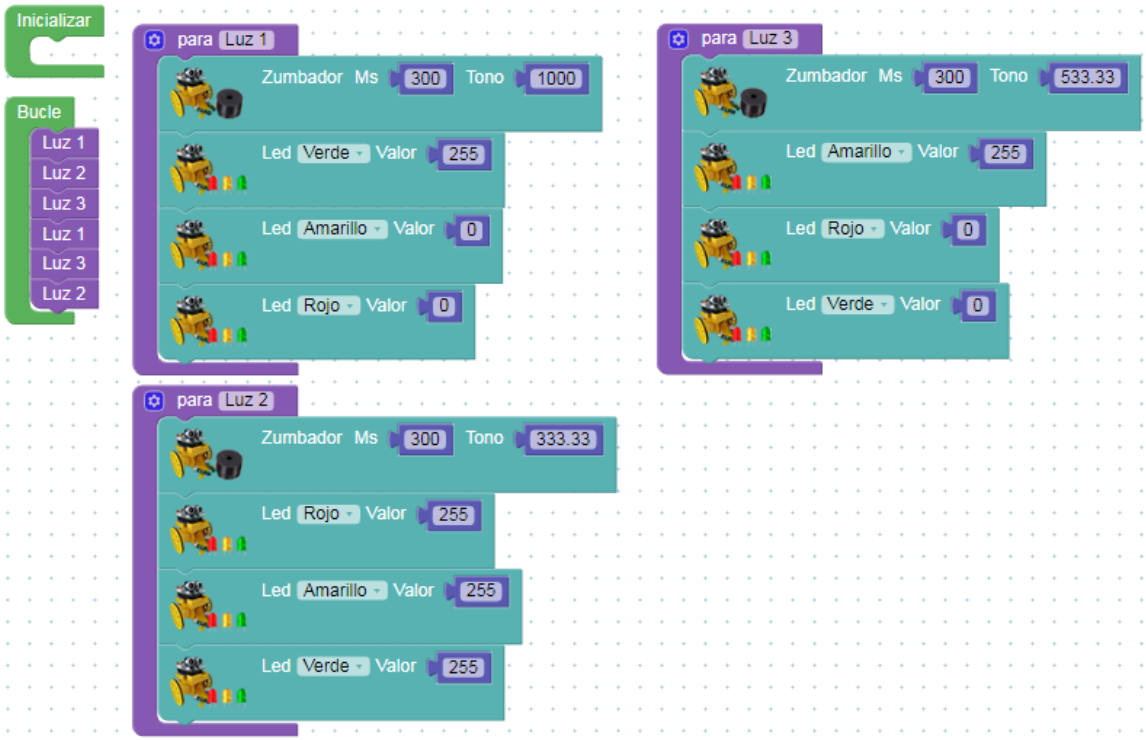
El programa completo queda de la siguiente forma:



Podéis probar a crear más funciones, insertarlas en diferente orden, cambiar los tiempos, etc.

A07_02 Juegos de luces con sonido.

Siguiendo el programa anterior, vamos a añadir unos “ruiditos” en las funciones, con el bloque “Zumbador”. Para que siga a la misma velocidad, ahora quitamos los tiempos de espera en el bucle, ya que el zumbador ya incluye ese tiempo de espera.



Igual que en el caso anterior, podéis probar a cambiar los tiempos, el tono, el orden de las funciones…

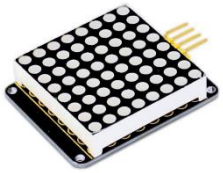
A08. – Dando expresión al robot

El siguiente material es opcional y no viene incluido en el Kit básico.

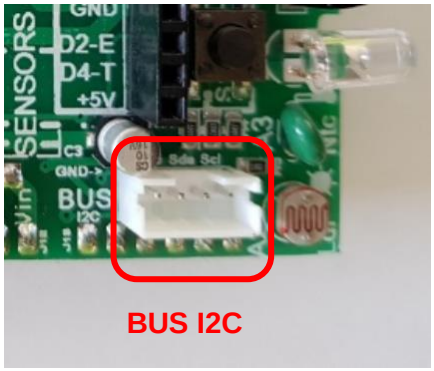
Tenemos muchos accesorios opcionales para poder conectar al robot. Hay un par de opciones muy interesantes que son una pantalla de texto y una matriz de leds:

Pantalla LCD 1602 I2C (Ref KS0061)

Matriz de Led 8x8 (Ref KS0064)

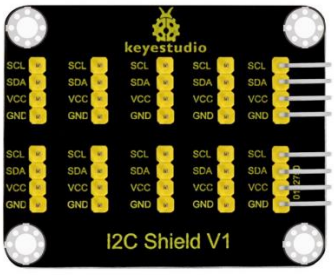


Estos elementos se conectan a través de un bus llamado I2C, el conector se encuentra al lado de los sensores de luz y temperatura.



I2C es un protocolo para poder conectar varios dispositivos de forma muy fácil con un par de cables para los datos y otros dos para alimentación y tierra o GND.

Existiendo, además, Hubs para poder conectar más de uno a la vez.



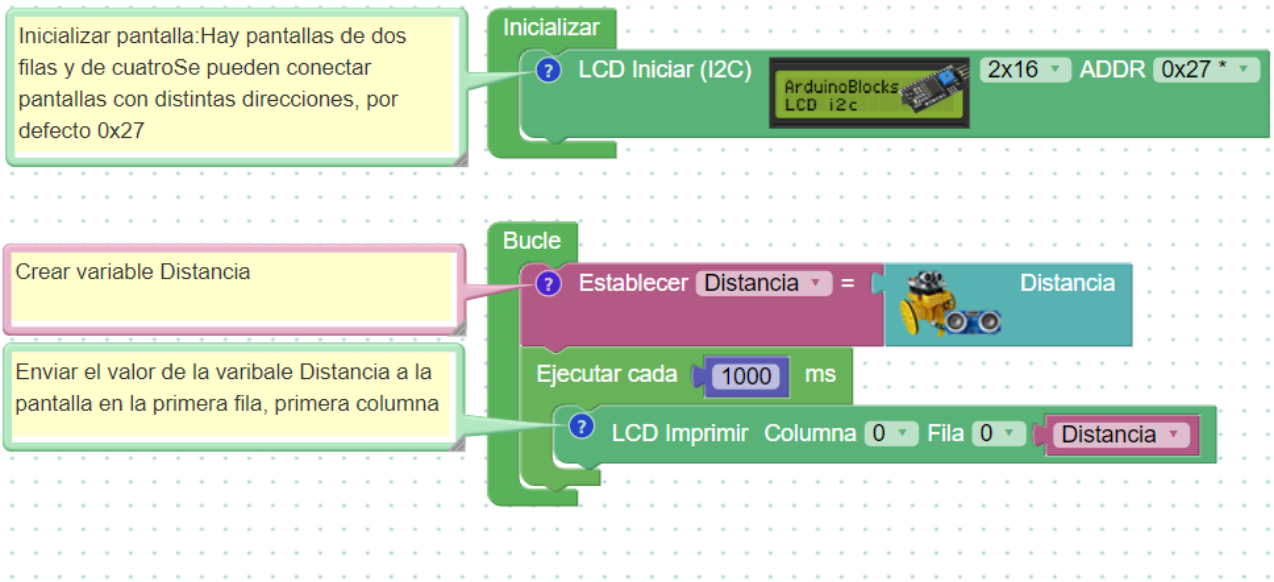
Módulo hub I2C (Ref KS0392)

A08_01 Pantalla

En la pantalla podemos enviar datos como los valores de los distintos sensores, temperatura, luminosidad y por ejemplo la distancia detectada por el sensor de “Distancia”.

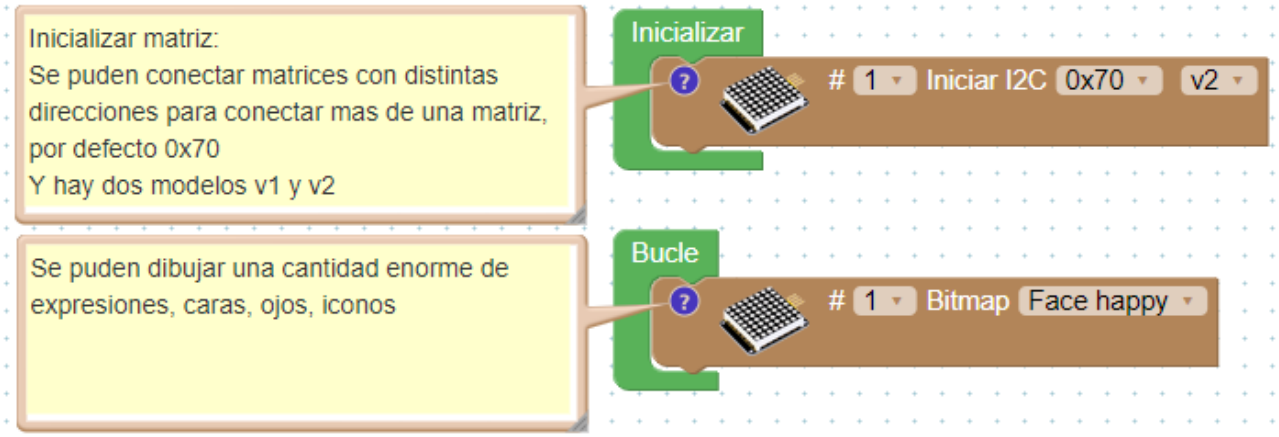
En el bloque “Pantalla LCD” encontramos la función “LCD Iniciar”, sirve para indicar que vamos a usar la pantalla, que hay que poner al bloque “Inicializar”, y la función “LCD Imprimir” que hay que poner el bloque “Bucle”.

Se puede “imprimir” los datos en la primera fila=0 o en la segunda fila=1 y empezar a escribir en la columna deseada des de la primera columna=0 hasta la última columna=15 o 19 (según modelo).



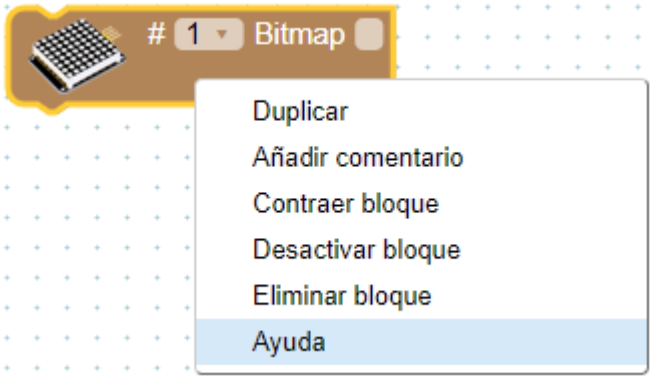
A08_01 Matriz led 8x8

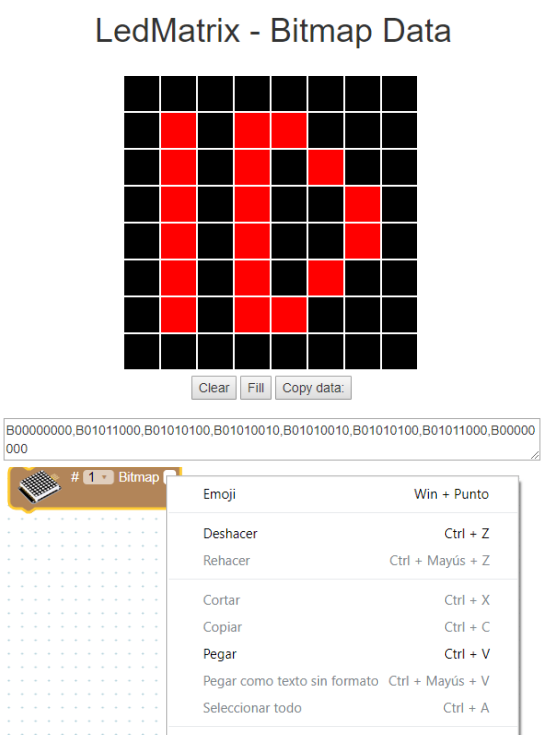
Con la matriz podemos hacer caras, iconos… hay cantidad de opciones:



Hay varios bloques para jugar, pero otro muy curioso es el bloque de “Bitmap”:

Seleccionando “Ayuda” con el botón derecho, se nos abre otra ventana/pestaña del navegador con la opción de dibujar lo que deseamos:





Una vez dibujado, clicamos en “Copy data:” y volviendo a la ventana de programación sobre el bloque “Bitmap”, en cuadro en blanco, le damos a “Pegar”.

A09. – Alarma de intrusos (sensor PIR)

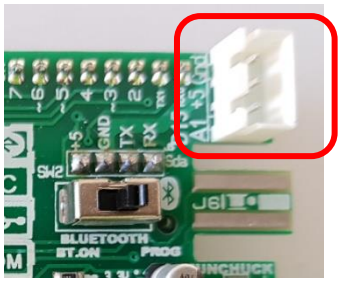
* El sensor PIR es un accesorio opcional no incluido en el Kit Básico.

En esta actividad crearemos una alarma contra intrusos utilizando un sensor infrarrojo pasivos PIR (Ref. KS0052) como sensor de movimiento. Este sensor tiene un alcance máximo de unos 7m según el fabricante y una apertura de detección de 100º.

Un sensor PIR detecta la radiación infrarroja que desprenden los seres vivos o cuerpos calientes.

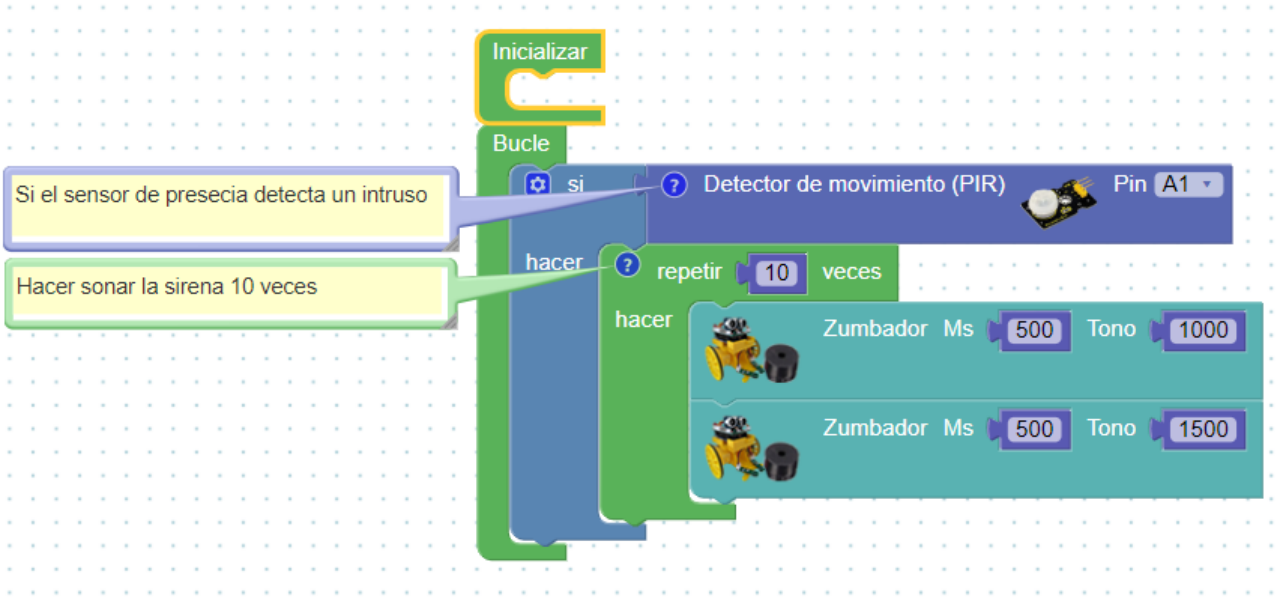
Antes de nada, vamos a conectar el sensor PIR en zócalo para sensores externos, utilizando uno de los cables de tres hilos, utiliza la entrada A1.

Zócalo sensor externo



Cuando damos alimentación a este sensor, se requiere de un tiempo para adaptarse a las condiciones donde se ha instalado. Durante este tiempo el sensor aprende a reconocer el estado de reposo (no detección). Este período de aprendizaje dura entre 10 y 60 segundos y es muy recomendable la ausencia de personas durante la calibración.

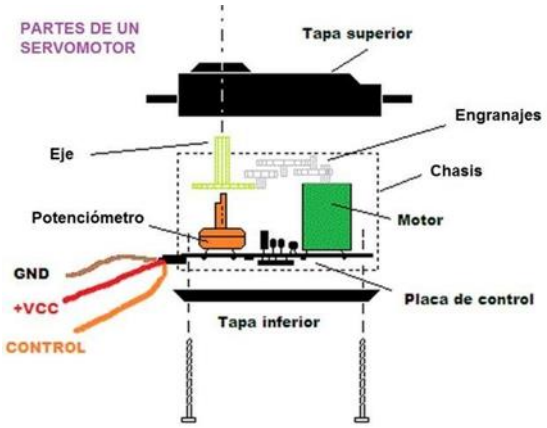
Cuando se detecta un intruso debe sonar la alarma 10 veces seguidas. La alarma constará de un sonido tipo sirena.



A10. – Controlar un servomotor

[* El servomotor es un accesorio opcional no incluido en el Kit Básico.]

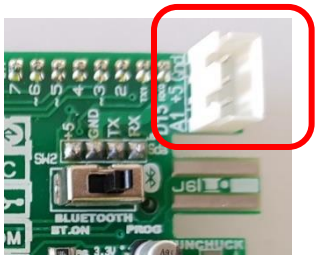
Un elemento muy utilizado en robótica y el mundo de Arduino es un servomotor, es un motor especial que puede posicionar su eje en un ángulo determinado y lo puede mantener en esta posición. Para funcionar sólo necesita alimentación GND, VCC (5voltios) y una señal de control.



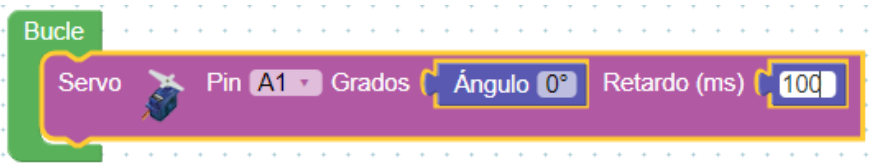
Los servomotores estándar sólo poden girar 180º, aunque en el mercado podemos encontrar de 270º y de 360º (giro continuo).

En el menú “Motor” encontramos un bloque que se llama “Servo”. Con el tenemos la posibilidad de controlar el servomotor, indicando los grados de rotación que queremos y el tiempo de retardo.

Zócalo sensor externo

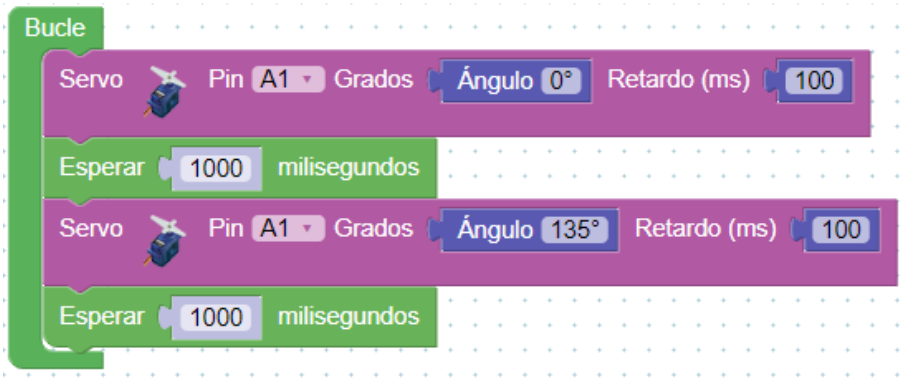


Por ejemplo, una buena idea es programar, por primera vez, el Ángulo a 0º para descubrir el punto de origen y a partir de aquí montar alguno de los accesorios que vienen con el servo para poder visualizar la rotación del eje. Vamos a volver a usar el conector “Zócalo sensor externo” que utiliza la salida A1.



Ahora ya podemos practicar con distintos grados, y observar donde apunta la “flecha” del eje.

Un servo motor nos puedes servir para accionar una pala para nuestro 3dBot Imagina Arduino, para mover un brazo de un robot humanoide, las posibilidades son muchas.



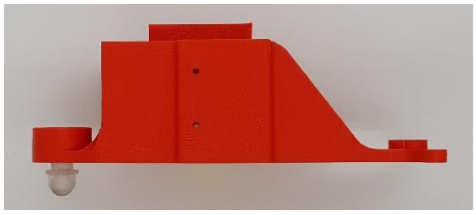
Actividades con el Imagina 3dBot Arduino completo

Estas actividades se proponen ya con el robot montado completamente, se adjuntan los pasos de montaje detalladamente, la programación no tiene ningún paso adicional, basta con subir el programa, desconectarlo del ordenador y conectar la batería.

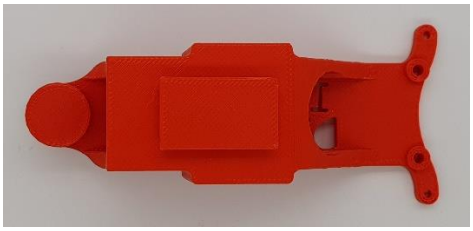
Preparativos: montaje completo del robot Imagina3dBot

1.-Comprobamos que el kit Imagina 3dBot Arduino contenga todo el material que se especifica a la página 10.

2.-Colocamos y verificamos que el chasis ya lleve puesta la rueda “loca” trasera.

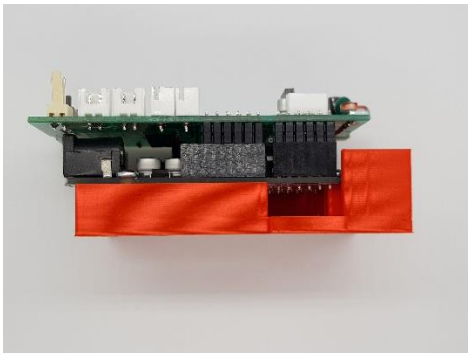


3.-Comprobamos la parte superior del chasis compacto, donde tenemos la pieza trapezoidal superior, donde encajara la caja de soporte de la placa.



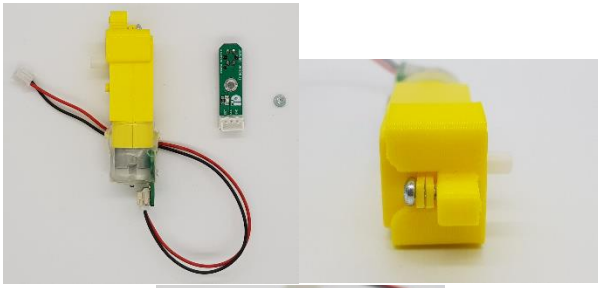
4.-Montamos la placa Arduino y el Shield Imagina Arduino a la caja de soporte de la placa con 3 tornillos de estrella de 2,9x6,5mm.

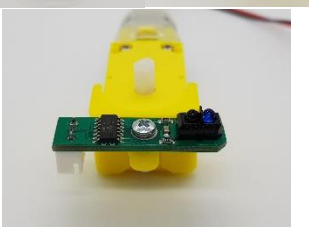
(Ya realizado para las primeras prácticas)



5.-Preparamos el motor izquierdo (motor A), que es el motor que lleva incorporado el sensor de línea fotoeléctrico que hace de “encoder”.

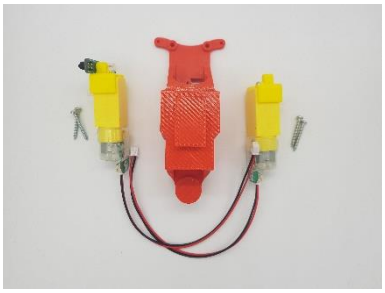
Hemos de poner el soporte para el sensor de línea tal y como se ve en la fotografía con un tornillo de 2,9x6,5mm.



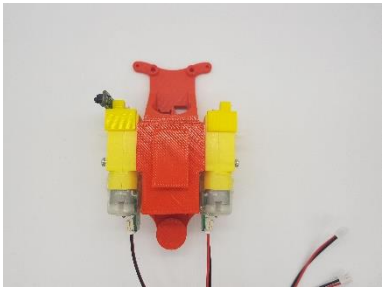


6.- Sujetamos al chasis los motores del robot apretándolos con dos tornillos de 2,9x25mm.

¡Importante! No hay que apretar en exceso los tornillos para no aplastar la caja reductora y los engranajes de los motores.

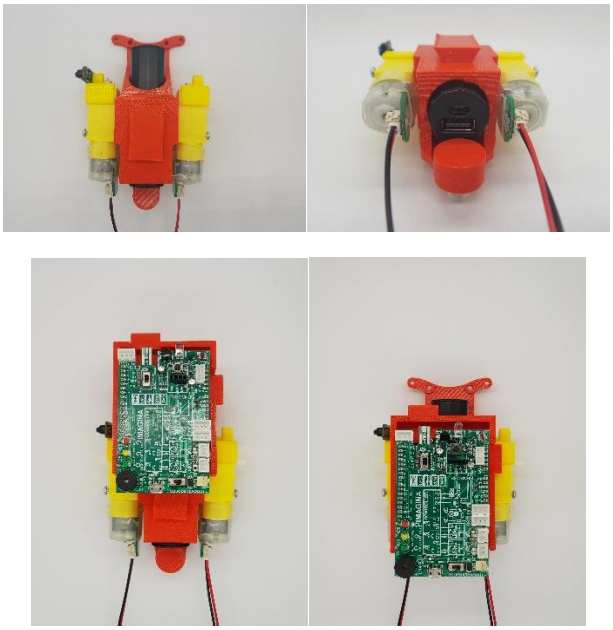


Hemos de tener en cuenta que el motor A lleva el encoder, es el que se coloca a la izquierda.

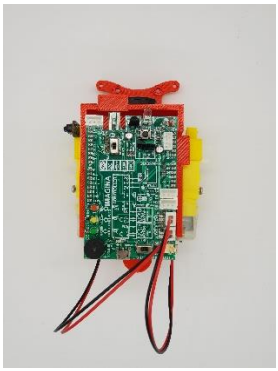


7.- Insertamos la batería Power Bank, en la parte inferior del chasis con el conector USB hacia la parte trasera.

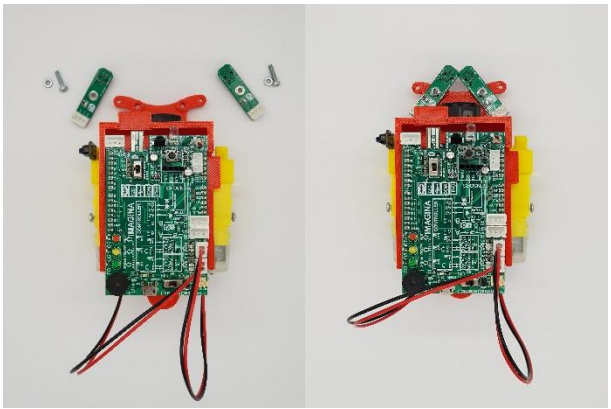
Luego insertamos la caja de soporte de la placa Imagina Arduino haciéndola encajar con la pieza trapezoidal de la parte superior del chasis.



8.- Conectamos el motor izquierdo o motor A y el motor derecho o motor B a la placa.

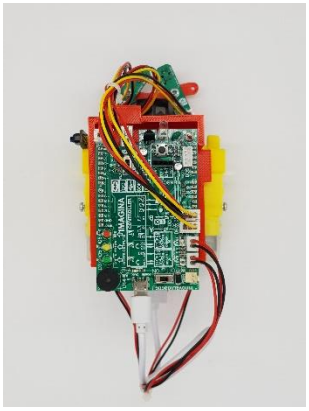


9.- Colocamos los sensores de línea fotoeléctricos, sujetándolos con un tornillo de M3x12 y una tuerca de M3.

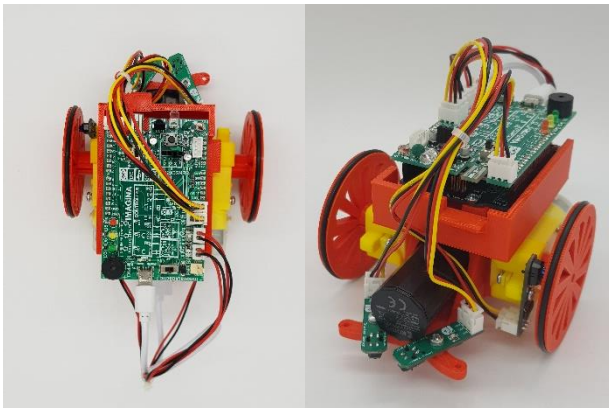


10.- Cuando ya los tenemos montados, conectamos el sensor fotoeléctrico izquierdo a la entrada D5 o SL, y el sensor fotoeléctrico derecho a la entrada D6 o SR, con sus respectivos cables.

También conectamos a la entrada D15 el sensor de línea fotoeléctrico de la rueda izquierda (encoder).

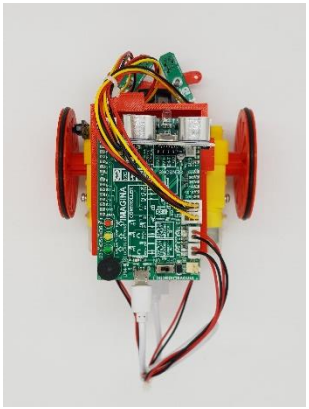


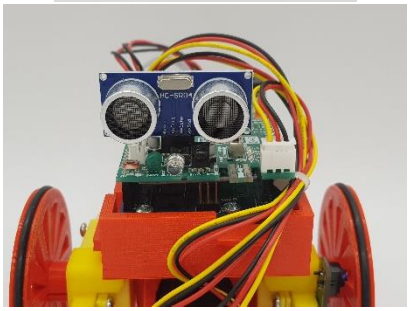
11.- Colocamos las ruedas al robot.

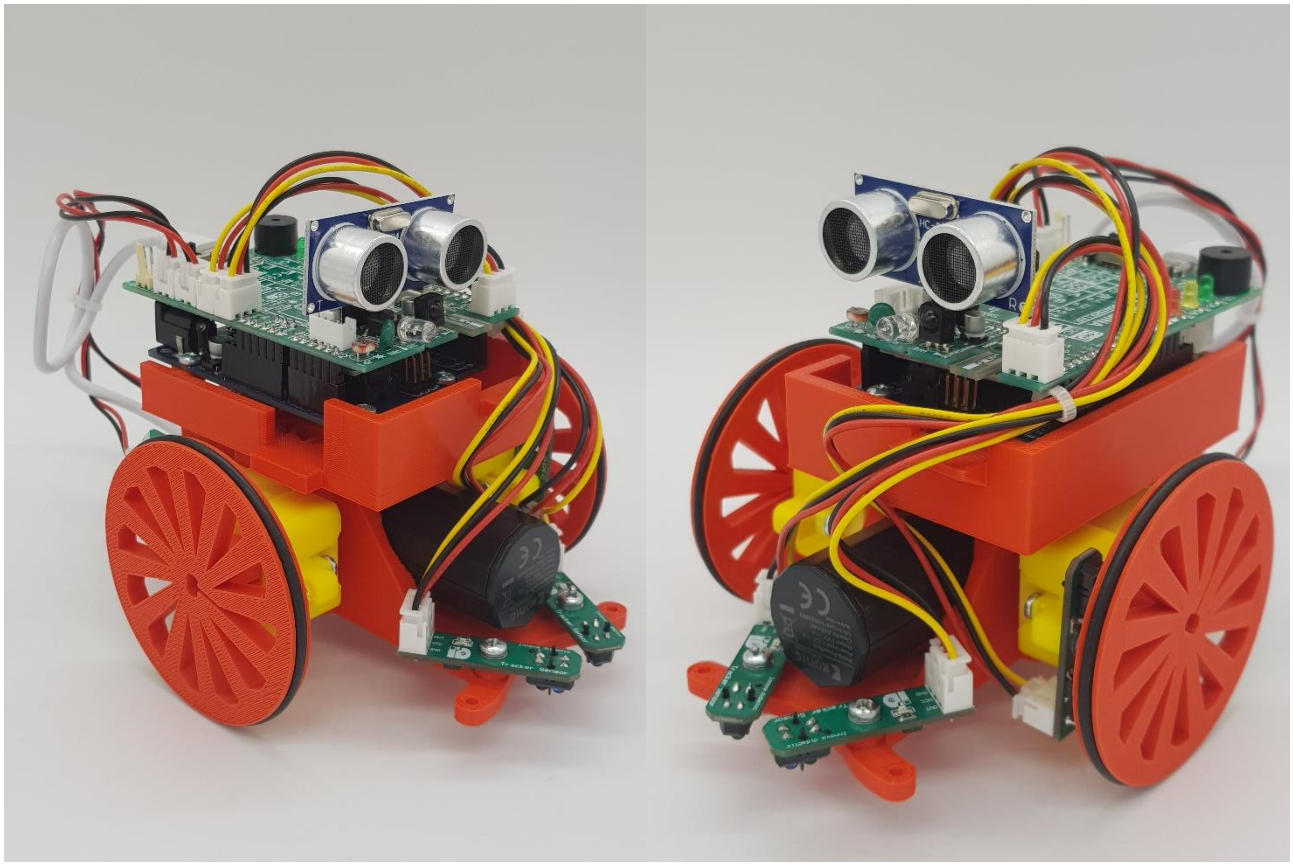


12.- Conectamos el sensor de ultrasonidos a la placa Imagina Arduino en su correspondiente zócalo, en la parte delantera de la placa.

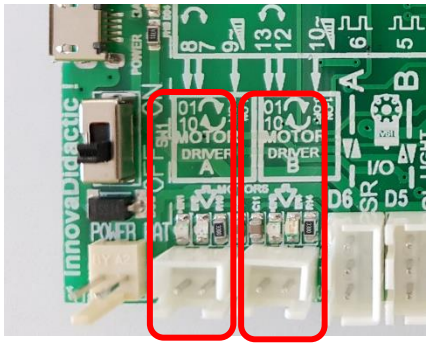
Ya utilizado para el ejercicio de distancias.



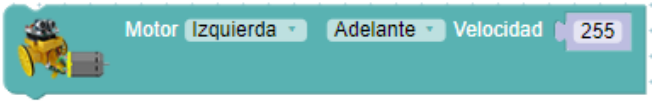




A11. – Controlar un motor CC

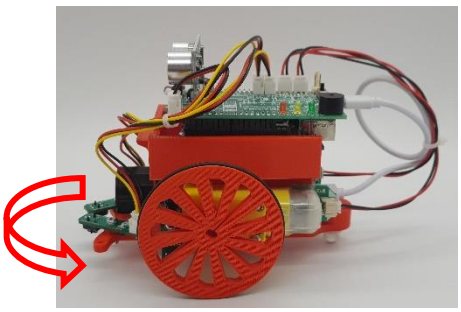


La placa Imagina Arduino ya lleva incorporado un “driver” o controlador de motores de CC, con él podemos realizar el control de dos motores de corriente continua (CC) de hasta 2A (Amperios).



Seguramente habréis observado que existen algunas instrucciones específicas para controlar el motor A o izquierdo, como las de la imagen adjunta. Cambiando la selección del motor, también sirve para controlar las mismas acciones para el motor B o derecho.

En el montaje del robot veíamos que, para conseguir un correcto funcionamiento con los bloques de ArduinoBlocks, hay que unir el motor izquierdo con el conector del motor A y el derecho con el conector B. Si los intercambiamos, provocaremos que el robot se mueva en sentido contrario al programado.



Realizar esta práctica nos servirá para saber si hemos montado y conectado correctamente los motores. Vamos a aprender a hacer funcionar el motor A, que es donde está conectado el motor de la izquierda, lo haremos mover hacia adelante en el sentido de la marcha o en el sentido contrario a las agujas del reloj.

En total podemos hacer funcionar cualquiera de los dos motores, en cualquiera de los sentidos, parar e incluso podemos ajustar la velocidad de rotación.

* En “Velocidad”, valores muy pequeños no son capaces de poner en marcha el motor.

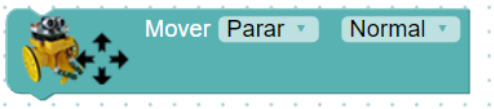
Si introducimos este programa observaremos que el motor A va hacia adelante a una velocidad rápida durante 2 segundos, luego a una velocidad más lenta otros 2 segundos y finalmente estará parado y repite el proceso.

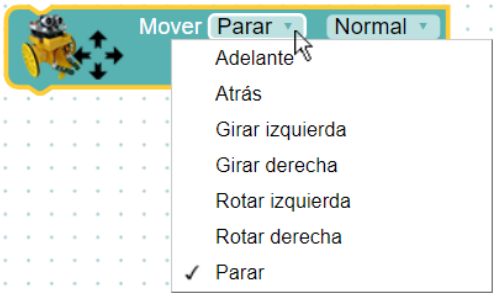
Hasta aquí hemos visto que podemos controlar los motores de forma individual, pero para hacer funcionar el Imagina 3dBot Arduino existe una función aún más fácil que se llama “Mover”.

Ahora ya estamos listos para sacarnos el carné básico de “Entrenador de robots”.

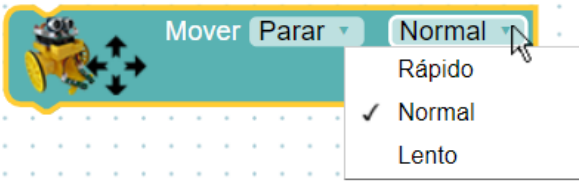
A12. – Controlar dos motores CC

Con un solo bloque, “Mover”, que encontramos en el apartado “3dbot”, podemos dar las ordenes de avanzar, retroceder, girar, rotar sobre sí mismo y parar. ¡Así de sencillo!





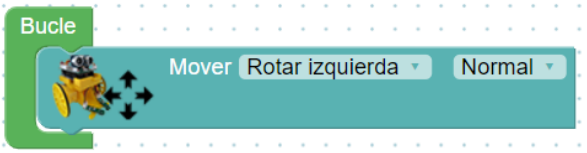
De la misma forma, se puede seleccionar la velocidad de entre tres posibles opciones:



Vamos a realizar una serie de ejercicios para practicar todo lo que hemos aprendido hasta ahora y mezclar los temas de funciones, variables,…

A12_01: ¡A bailar!

El siguiente programa hace que el robot rote sobre si mismo constantemente:

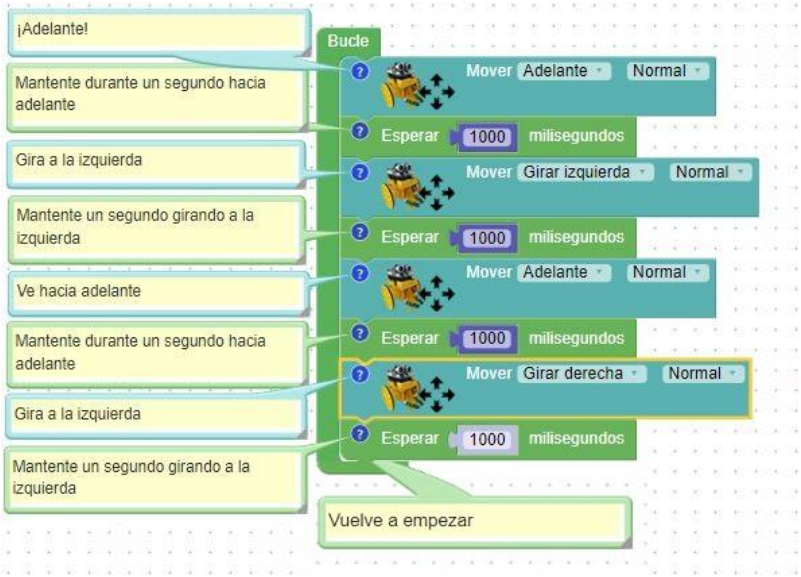


A12_02 Creando secuencias de movimientos.

Vamos a empezar a estirar las piernas y las ruedas… con unos movimientos sencillos de calentamiento. Poco a poco haremos que estas secuencias dependan de estímulos externos y condiciones más elaboradas.

El robot va a realizar los siguientes movimientos:

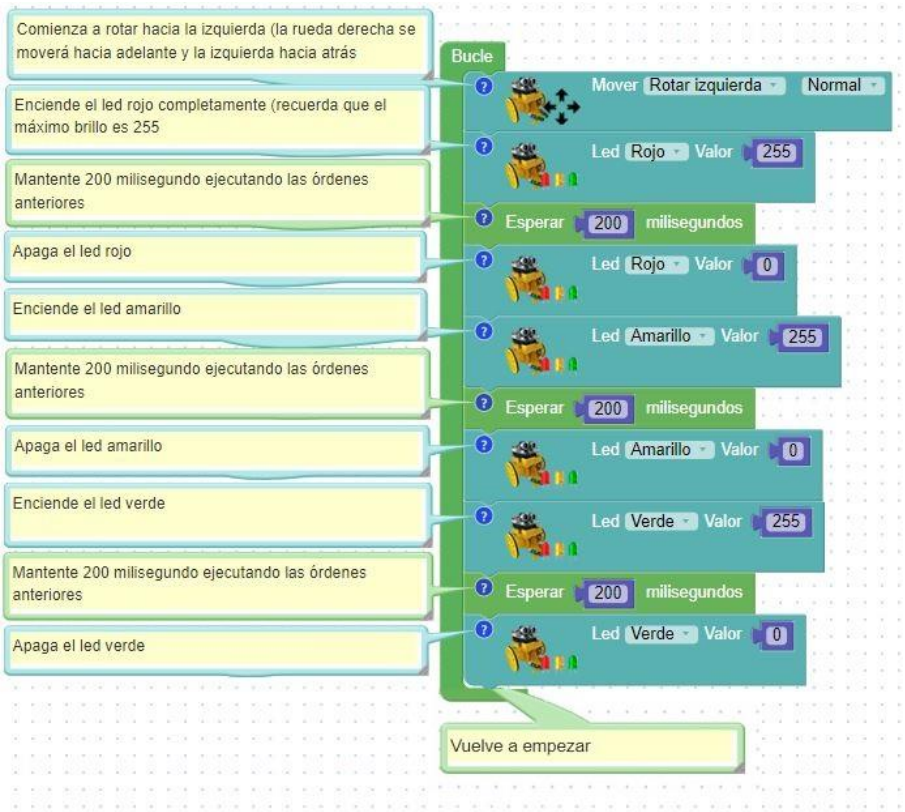
1. Ir en línea recta durante un segundo
2. Girar a la izquierda durante 1s
3. Volver a ir en línea recta otro segundo
4. Girar a la izquierda 1s



A12_03 Giro y control de los leds rojo, amarillo y verde: Semáforo loco.

Empieza el espectáculo: Leds, cámaras y… ¡Acción!

Ahora nuestro robot va a rotar todo el tiempo a la izquierda, mientras sus tres leds se encienden y apagan de forma secuencial. ¡Recuerda que el ritmo desenfrenado lo marcas tú!

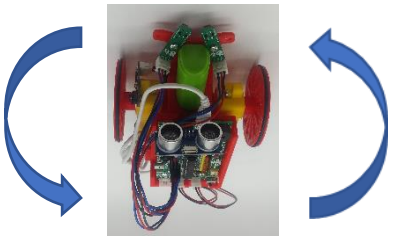


Ten en cuenta que el robot va a estar rotando a la izquierda todo el tiempo, independientemente de lo que hagan los leds, ya que nunca le ordenamos que pare.

A12_04 Para poder conocer todos los sentidos de giro posibles que puede hacer el robot, os proponemos que realicéis los programas de los movimientos siguientes.

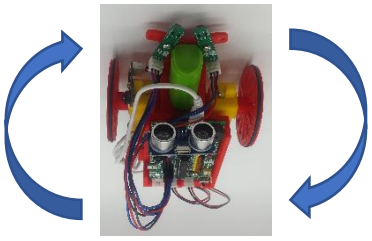
Pivota a la izquierda:

Motor A atrás / Motor B adelante



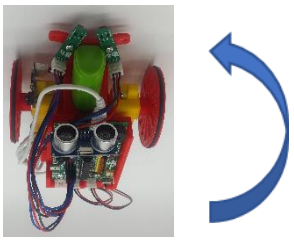
Pivota a la derecha:

Motor A adelante / Motor B atrás



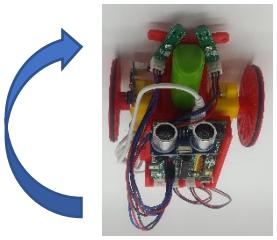
Gira a la izquierda adelante:

Motor A parado / Motor B adelante



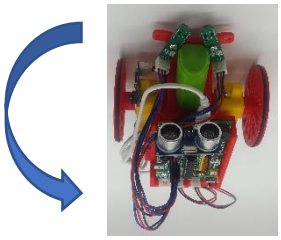
Gira a la derecha adelante:

Motor A adelente / Motor B parado



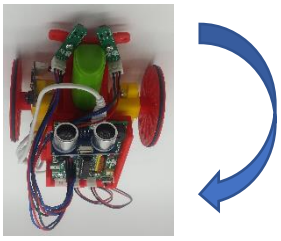
Gira a la derecha atrás:

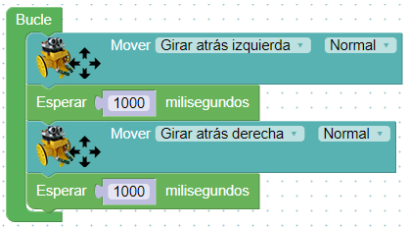
Motor A parado / Motor B atrás



Gira a la izquierda atrás:

Motor A atrás / Motor B parado





A12_05 Señalizando la marcha atrás.

Este programa tiene dos partes fundamentales:

- Avance hacia delante con el piloto verde encendido.
- Marcha atrás con el piloto rojo encendido y un pitido intermitente de señalización de maniobra “peligrosa”.

Vamos a presentar una versión del programa en la que creamos dos funciones, y otra, en la que se realiza el programa completo directamente en el bucle (sin funciones). La diferencia en este caso no es significativa, ya que no es un programa extenso ni se realizan muchas repeticiones.

Por cierto, cuidado porque lo peligroso puede acabar siendo el pitido… para los oídos.

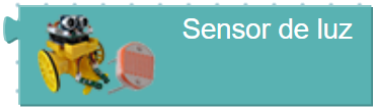
Programa con funciones

Se definen dos funciones: “ir hacia delante” e “ir hacia atrás”. El programa completo es el siguiente:

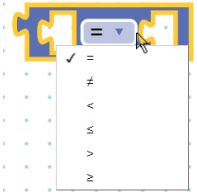
Programa sin funciones

A12_06 En busca de la luz.

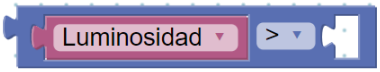
Vamos a jugar un poco con los niveles de luz. Imagina 3dBot Arduino tiene un sensor de luz (LDR) que detecta las variaciones de iluminación y las traduce en señales de tensión. En el apartado “3dBot” se encuentra el bloque que se encarga de programar este sensor.



Además, en apartado “Lógica” encontramos un bloque que nos permite realizar comparaciones que NO dependen de operaciones matemáticas directas, es decir, comparaciones de estados, valores, etc. Haciendo clic sobre la flecha dibujada en el bloque, se abre un desplegable con diferentes opciones de comparación:



Vamos a combinar este último bloque con funciones ya aprendidas, para comparar un valor de iluminación fijado por nosotros, con el valor de iluminación que esté leyendo el robot. En este caso, vamos a guardar los valores de iluminación leídos en la variable “Luminosidad”.

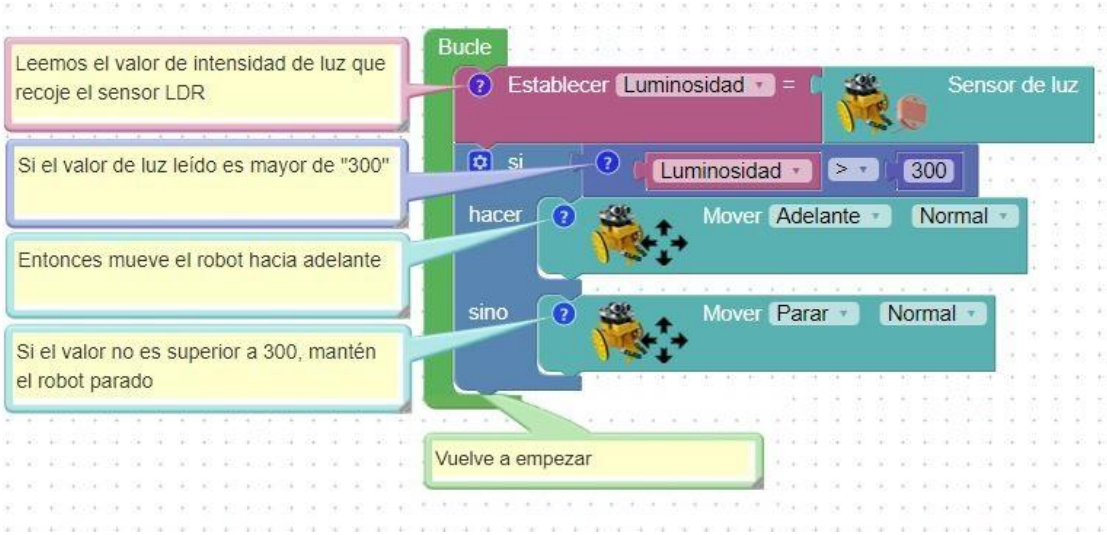


Ahora, sacamos el bloque de números del apartado “Matemáticas”, para realizar la comparación, y lo incluimos en el bloque general:

Si hay luz, ¡avanza!

En este caso vamos a hacer que, si el valor de luz detectado es superior a una cifra (en este ejemplo 300) el robot avance en línea recta. Si el valor es menor, se mantenga parado.

Podéis usarlo, por ejemplo, para hacer que se mueva solo cuando se le enfoque con la linterna del teléfono móvil en una habitación con poca luz.



Si tras cargar el programa el robot se mueve todo el tiempo (aunque no se le enfoque con la linterna), querrá decir que el nivel de luz de la sala es mayor que ese 300 incluido en el programa. En ese caso, simplemente habrá que aumentarlo hasta conseguir el resultado deseado.

Por otro lado, solo con cambiar el operando de la comparación, conseguiremos el efecto contrario, que el robot avance cuando haya poca luz y se pare cuando el nivel suba:

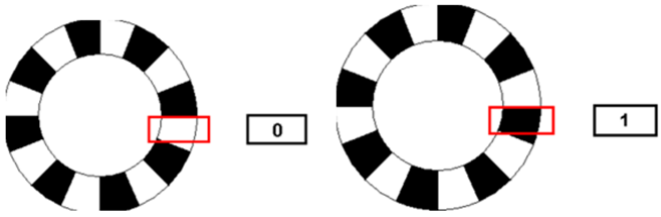
A13. – Control del robot con un encoder (codificador rotatorio)

¿Qué es un encoder?

En pocas palabras, un encoder es un dispositivo de detección que proporciona una respuesta. Los encoders convierten el movimiento en una señal eléctrica que puede ser leída por algún tipo de dispositivo de control en un sistema de control de movimiento, como por ejemplo nuestro robot. Nos permitirá saber cuándo avanza y retrocede nuestro robot.

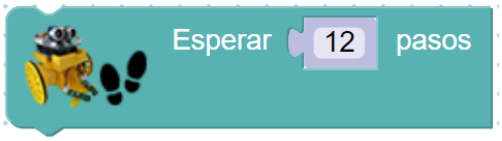
Hemos visto en el montaje del robot que en la rueda izquierda hay un sensor óptico con el fin de actuar como un encoder. La función del encoder es convertir en señales eléctricas el giro de la rueda, para así saber cuánto ha girado en cada momento.

Como se aprecia en la siguiente figura, el sensor óptico se activa cuando detecta un radio de nuestra rueda delante de él y se desactiva cuando no hay nada (espacio entre radios).



Teniendo en cuenta que la rueda tiene 12 radios, cuando hayamos contado 12 pulsos querrá decir que la rueda ha dado una vuelta completa.

Podemos utilizar esta funcionalidad para controlar distancias y giros en el 3dBot. Lo hacemos con el bloque “Esperar__Pasos”, que se encuentra en el apartado 3dBot. A cada pulso recibido se le llama “paso”. Siguiendo el ejemplo de antes, una vuelta de rueda completa serán 12 “pasos”.



Se puede comprobar experimentalmente que, aproximadamente, para realizar un giro de 90 grados hacia la derecha, la rueda izquierda tiene que moverse 4 pasos (pasan 4 radios de la rueda por delante del sensor óptico). De esta forma le podemos decir a nuestro robot que gire 90 grados, 180 grados, etc.

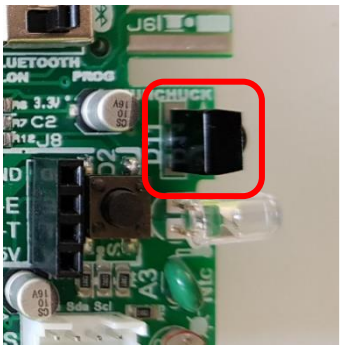
Recorrido programado contando “pasos”.

El programa funciona de forma similar a los juegos con led usando la función “Esperar”. Aquí, el robot mantendrá la última dirección programada, hasta que se dé el número de pasos descrito tras esta.

A14. – Control del robot con infrarrojos

Control con mando a distancia.

La placa Imagina Arduino lleva incorporado un receptor infrarrojo, se encuentra al lado de los sensores de temperatura y luz.



Receptor infrarojo

Pasamos a controlar el robot en tiempo real. Vamos a utilizar inicialmente el mando de Keyestudio, que dispone de un bloque específico en ArduinoBlocks, lo que hace que la programación sea muy rápida y sencilla.

Posteriormente aprenderemos a utilizar cualquier mando, incluido el de la tele, que siempre está muy a mano.

Y finalmente, haremos un programa un poco más avanzado en el que, no solo cambiaremos la dirección del robot, sino también su velocidad.

A14_01: Control de la dirección.

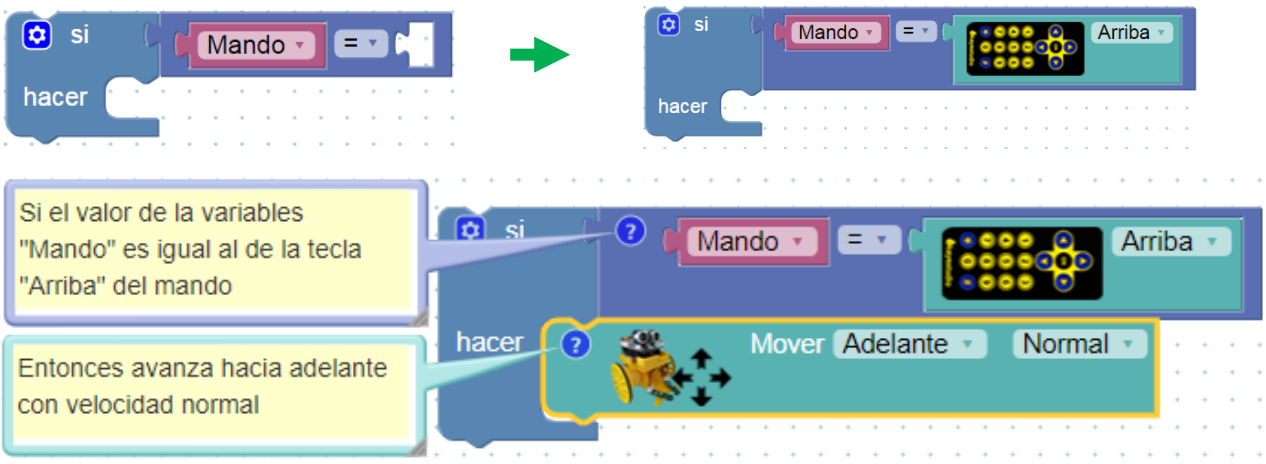
Para este ejemplo vamos a utilizar el mando de infrarrojos (IR) de Keyestudio. Concretamente usaremos las teclas de dirección para mover al robot en las cuatro direcciones. Además, con el botón central de “OK”, haremos que pare.

Para ello, necesitamos dos nuevos bloques:

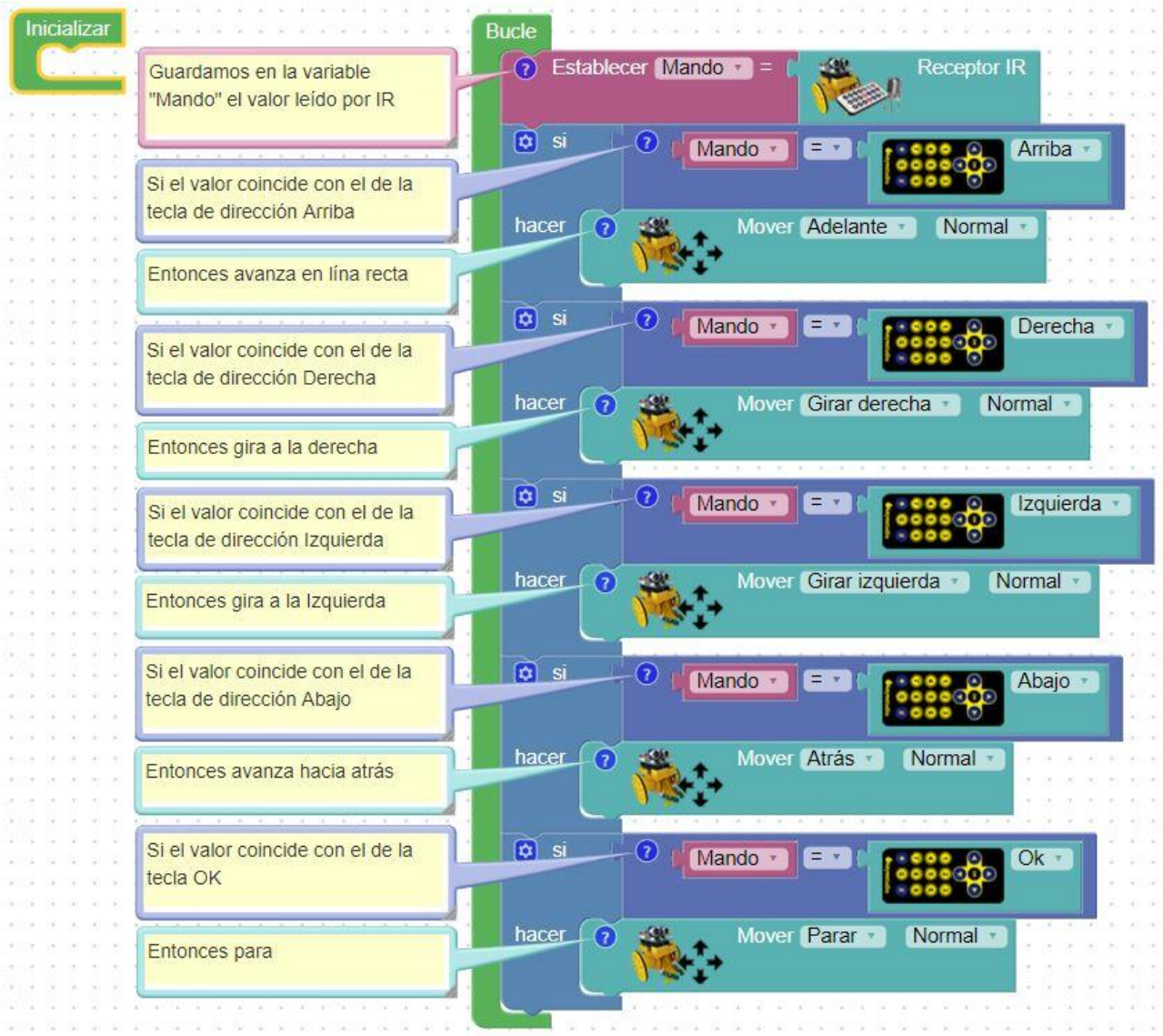
- En el apartado “3dBot” tenemos el bloque “Receptor IR” que lee los datos procedentes de un mando con infrarrojos:

- En el apartado “3dBot” encontramos el bloque del mando de Keyestudio, en el que, pulsando en el desplegable, podemos seleccionar cualquiera de sus teclas:

Usando el bloque comparación, ubicado en la sección “Lógica” (y explicado en el apartado A02), compararemos el dato leído por el sensor IR con el dato seleccionado por nosotros para una función determinada. Con un condicional “Si”, decidiremos qué acción tomar.

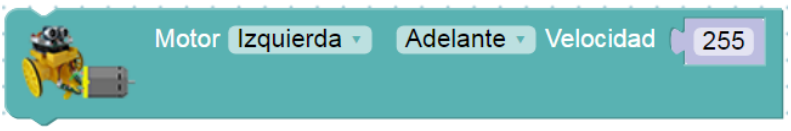


En la variable “Mando” guardamos los valores leídos por el receptor infrarrojos, para así compararlos con los valores correspondientes al mando. El programa resultante lo podemos ver a continuación:



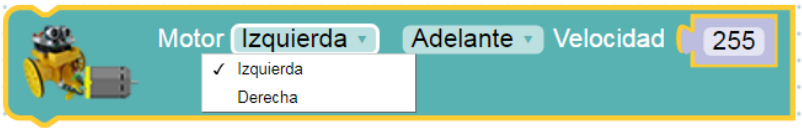
A14_02: Control de dirección y velocidad con el mando de Keyestudio.

En el apartado “3dBot” encontramos el bloque con el que controlar la dirección de cada motor y su velocidad.



La velocidad del motor se controla con el dato numérico que hay en la parte derecha del bloque. Su velocidad aumenta proporcionalmente al número incluido, siendo 0 motor parado, y 255 motor a su máxima velocidad. Ahí podemos incluir una variable para que ese valor de velocidad provenga de una lectura del mando IR, por ejemplo.

En el desplegable al lado de la palabra “Motor”, seleccionamos qué motor queremos controlar:



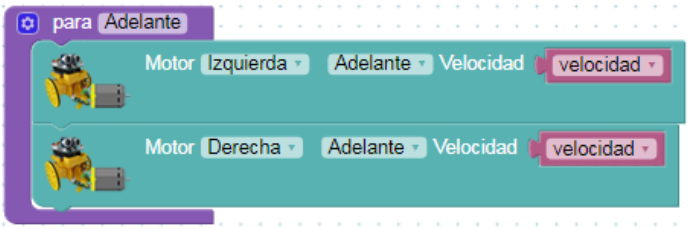
Y en el desplegable siguiente, elegimos el sentido de giro:

Vamos a utilizar también un nuevo tipo de variable, las variables de texto. Las encontramos también en el apartado “Variables” de ArduinoBlocks. En ellas podemos guardar palabras, como: “Adelante”, “Derecha” o “Izquierda”.

Una vez conocidos los nuevos elementos, vamos a realizar el programa. Para ello, se van a definir las variables que representan los movimientos que podrá efectuar el robot. Como decíamos, la velocidad la tomaremos de una variable “Velocidad” que crearemos en el bucle del programa.

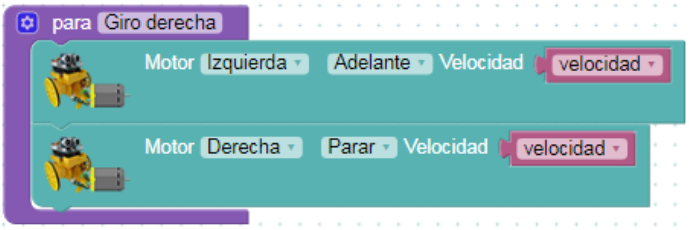
Funciones:

- “Adelante”: Con los dos motores girando en el sentido de avance, hacemos que se mueva en línea recta:



- “Giro izquierda”: Lo conseguimos avanzando con la rueda derecha y manteniendo la izquierda apagada:

- "Giro derecha": Lo conseguimos avanzando con la rueda izquierda y manteniendo la derecha apagada:



- "Parar": Lo hacemos seleccionando en ambos motores la opción "Parar".

Bucle:

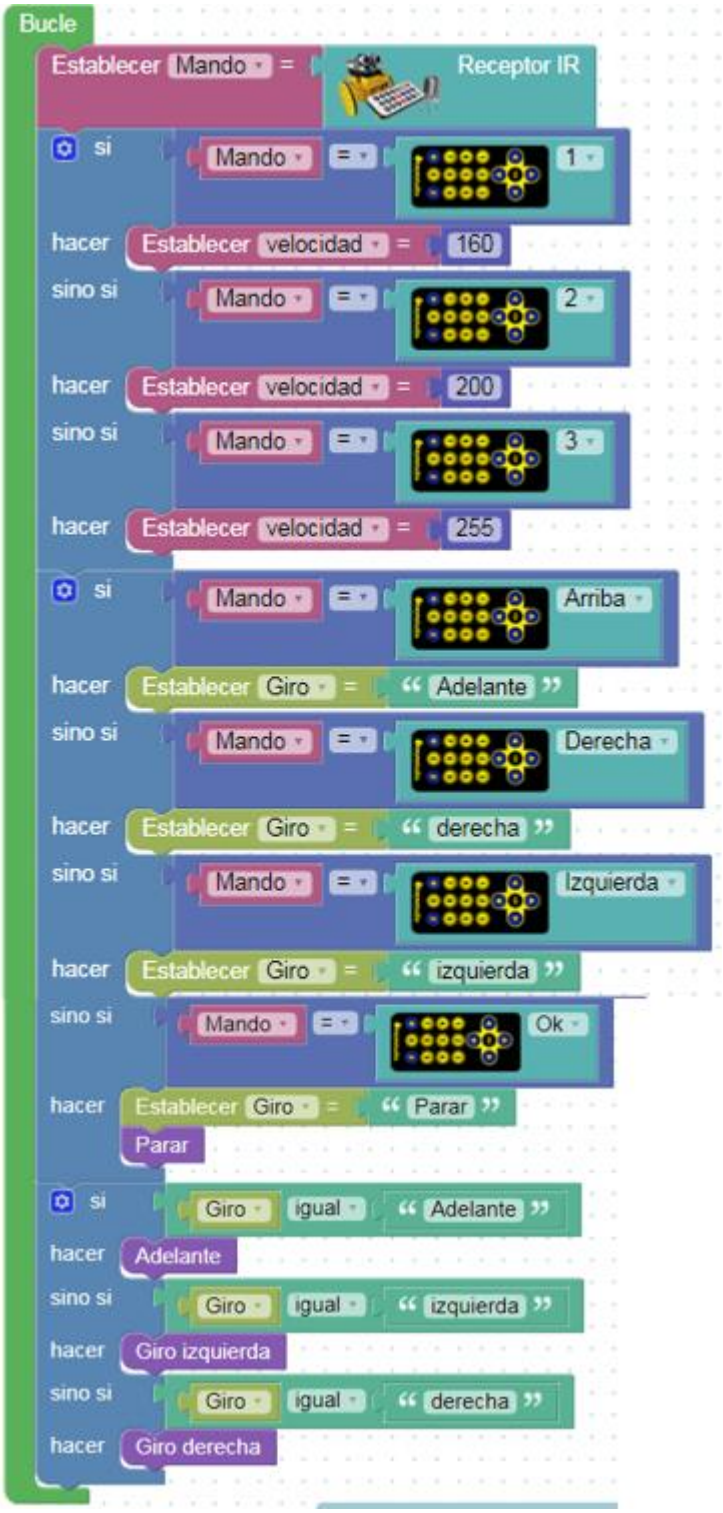
Primero leemos los datos recibidos por el receptor IR y los guardamos en la variable "Mando". A continuación, diferenciamos los datos recibidos en dos tipos:

- Datos para el cambio de velocidad.
- Datos para el cambio de dirección.

Los datos para el cambio de velocidad los guardamos en la variable numérica "Velocidad" y los datos para el cambio de sentido de giro los guardamos en la variable de texto "Giro".

Como hemos visto en la definición de las funciones, la variable "Velocidad" la insertamos en los bloques de movimiento de los motores, dentro de la definición de las funciones.

Finalmente, establecemos las condiciones para que, dependiendo del valor de la variable "Giro", vaya en la dirección deseada.


Bucle
Establecer Mando = Receptor IR
si Mando = 1
hacer Establecer velocidad = 160
sino si Mando = 2
hacer Establecer velocidad = 200
sino si Mando = 3
hacer Establecer velocidad = 255
si Mando = Arriba
hacer Establecer Giro = " Adelante "
sino si Mando = Derecha
hacer Establecer Giro = " derecha "
sino si Mando = Izquierda
hacer Establecer Giro = " izquierda "
sino si Mando = Ok
hacer Establecer Giro = " Parar "
Parar
si Giro igual " Adelante "
hacer Adelante
sino si Giro igual " izquierda "
hacer Giro izquierda
sino si Giro igual " derecha "
hacer Giro derecha

A14_03: Lectura de los códigos de las teclas desde un mando IR.

Primero tenemos que ver qué número nos manda cada botón del mando que vayamos a usar. Lo que vamos a hacer es presionar el botón del mando, leer el dato enviado e imprimirlo en la pantalla del ordenador. De ahí ya podremos copiarlo para nuestro futuro programa.

Simplemente hay que recordar que, para visualizar datos en pantalla, hay que realizar el envío de datos al ordenador cada cierto tiempo (no constantemente) para no saturar la comunicación. Veamos cómo queda el programa:

Dependiendo de qué botón se pulse, aparecerá un valor u otro. Es importante tener en cuenta que, si no accionamos ningún pulsador, lo habitual será recibir un 0. Obtendréis algo así, tras conectaros a la “Consola”:

A continuación, anotamos a qué pulsador corresponde cada valor recibido, para así añadirlo posteriormente a la programación. Para nuestro ejemplo, queda la siguiente tabla:

Número leído	Tecla	Función que voy a asignar en 3dBot
551486208	Flecha arriba	Adelante
551510144	Flecha derecha	Giro derecha
551542784	Flecha izquierda	Giro izquierda
551494336	Tecla OK	Parar

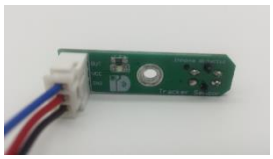
A14_04: Control del robot con un mando IR cualquiera.

Usando los códigos leídos en el programa anterior, vamos a hacer un sistema de control de nuestro robot por mando IR. El programa es exactamente igual que el del ejercicio anterior, solo que ahora comparamos con estos nuevos códigos.

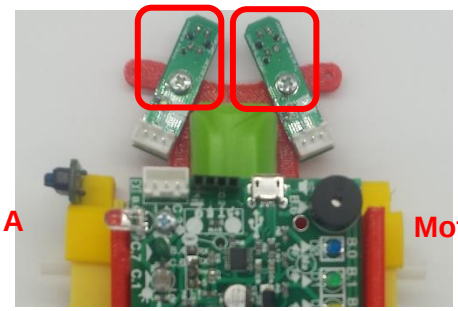
En este caso vamos a guardar las lecturas del sensor IR en una variable que hemos llamado “Lectura_Mando”. Da igual el nombre, podéis poner el que queráis.

Se le pueden añadir más funciones, como la de movimiento hacia atrás, que en este caso no se ha programado, rotaciones, luces, etc.

A15. – Robot seguidor de líneas

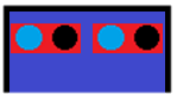


Podemos hacer que el robot Imagina 3dBot Arduino se comporte como un seguidor de líneas utilizando los sensores de línea fotoeléctricos. Para ello nos ayudaremos de las señales del sensor de línea izquierdo (SI), conectado a SL, (Sensor Left) y el sensor de línea derecho (SD), conectado SR (Sensor Right). Estos sensores ópticos permiten distinguir entre superficies de colores claros y superficies de colores oscuros.

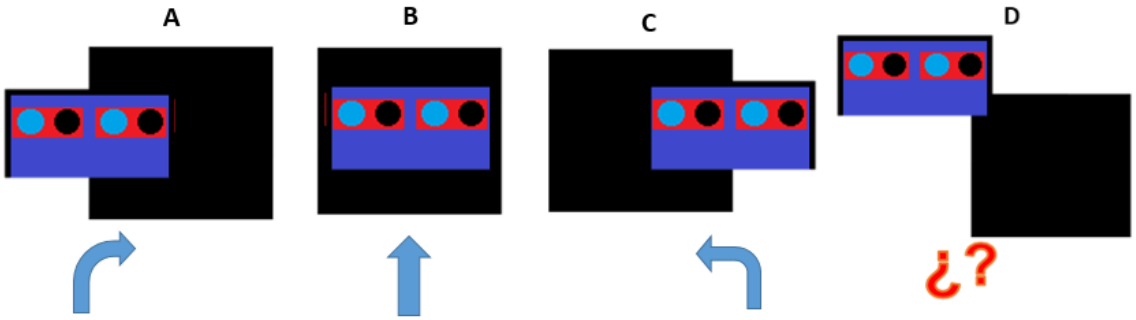


La disposición de los sensores y de los motores es:

Para entender el programa, vamos a representar los dos sensores ópticos con la siguiente figura:



Al ponerlos sobre una línea negra, se pueden dar cuatro casos:



En el caso A de la imagen superior, el robot se ha salido hacia la izquierda, por tanto, habrá que hacer que gire a la derecha hasta que ambos sensores se vuelvan a encontrar en línea negra. En el caso B, el robot debe avanzar en línea recta porque va bien, y en el caso C, deberá girar hacia la izquierda para corregir su rumbo, ya que se ha salido a la derecha. Con las curvas, el funcionamiento es el mismo.

Además, hay que añadir el caso en que el robot se sale totalmente de la línea negra (D), momento en que ambos sensores están en superficie blanca.

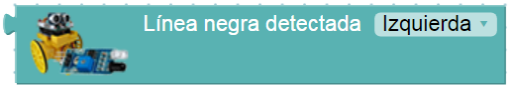
A15_01 Robot seguidor de línea. Versión 01

Para empezar, vamos a hacer un programa que actúe frente a los casos A, B y C. Para el caso D, en el que el robot pierde totalmente la referencia de la línea, vamos a hacer simplemente que se pare.

Lo primero que hacemos al iniciar el programa, es guardar el estado de cada sensor de línea en una variable booleana, que son aquellas que solo pueden tomar como valores 0 y 1. Esto quiere decir que, o se detecta línea negra, o se detecta línea blanca ¡Aquí no hay medias tintas!

Las variables booleanas las encontramos, al igual que las demás, en el apartado “Variables” de ArduinoBlocks.

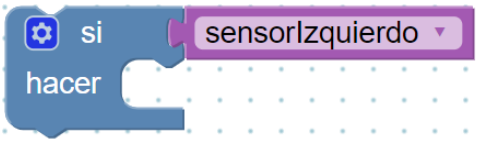
También vamos a usar el bloque correspondiente a los sensores ópticos “Línea negra detectada”. Este bloque nos permite diferenciar entre una superficie negra y una blanca, y lo podemos encontrar en el apartado “3dBot”.



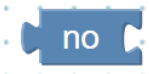
*Es importante tener en cuenta que este bloque indica señal de activación cuando se encuentra sobre línea negra.

Así, al guardar las lecturas del sensor de línea en una variable booleana, cuando el sensor esté sobre línea negra, esa variable valdrá 1. Sobre línea blanca, valdrá 0.

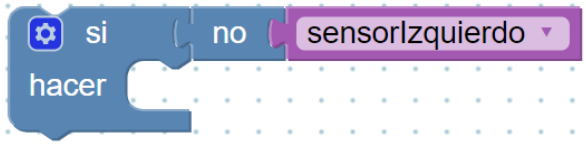
La combinación de estas variables con los condicionales “*Si*”, funciona de forma directa, sin tener que usar bloques de comparación. En la siguiente imagen vemos una condición que expresa directamente que, si la variable “sensorIzquierdo” vale 1, entonces que se ejecute la acción…



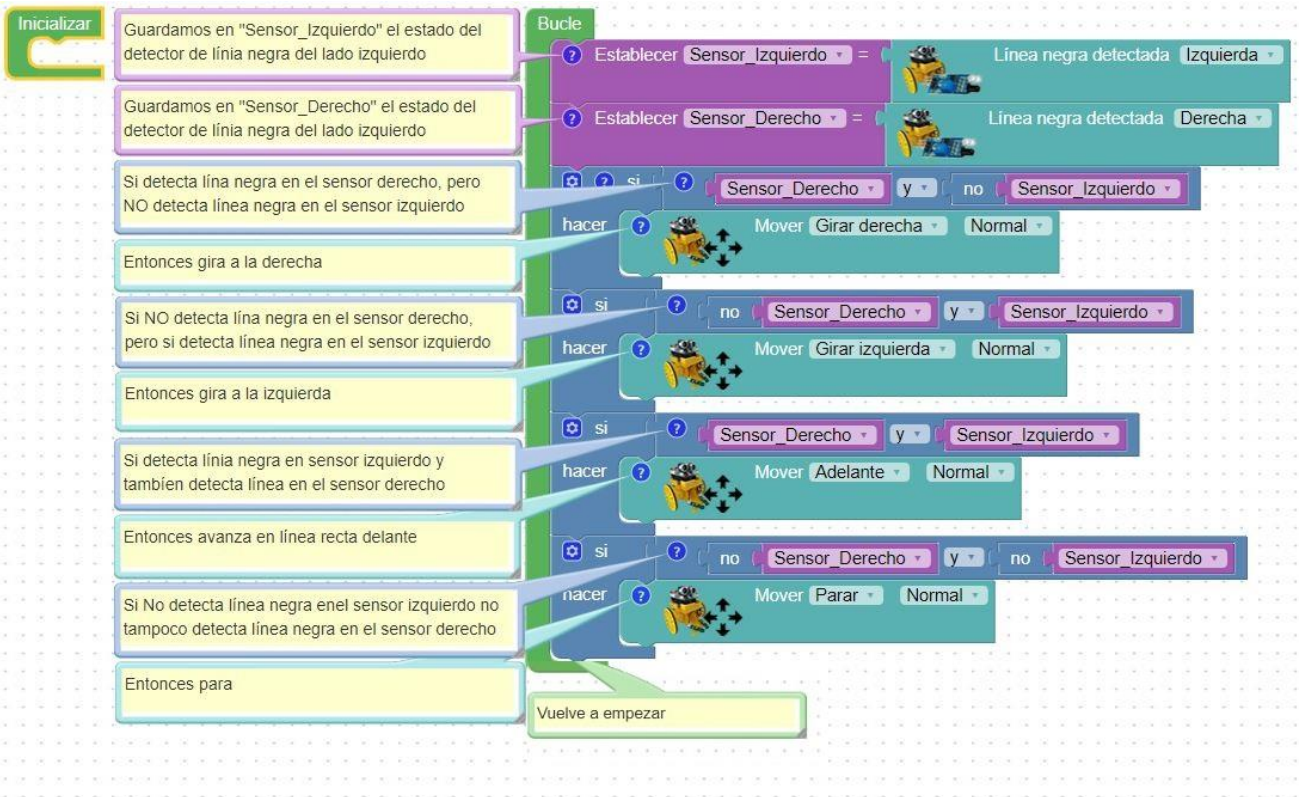
Para completar el proceso, hay que señalar también el uso del bloque “no”, situado en apartado “Lógica” de ArduinoBlocks, que lo que hace es negar lo que se coloque detrás de él.



Por lo tanto, el siguiente condicional indica que, si el valor de “sensorIzquierdo” NO es 1, es decir, es 0, entonces que se ejecute la acción… Esta condición, que es el caso contrario al anterior, se cumplirá cuando el sensor esté sobre línea blanca.



El programa queda de la siguiente manera:

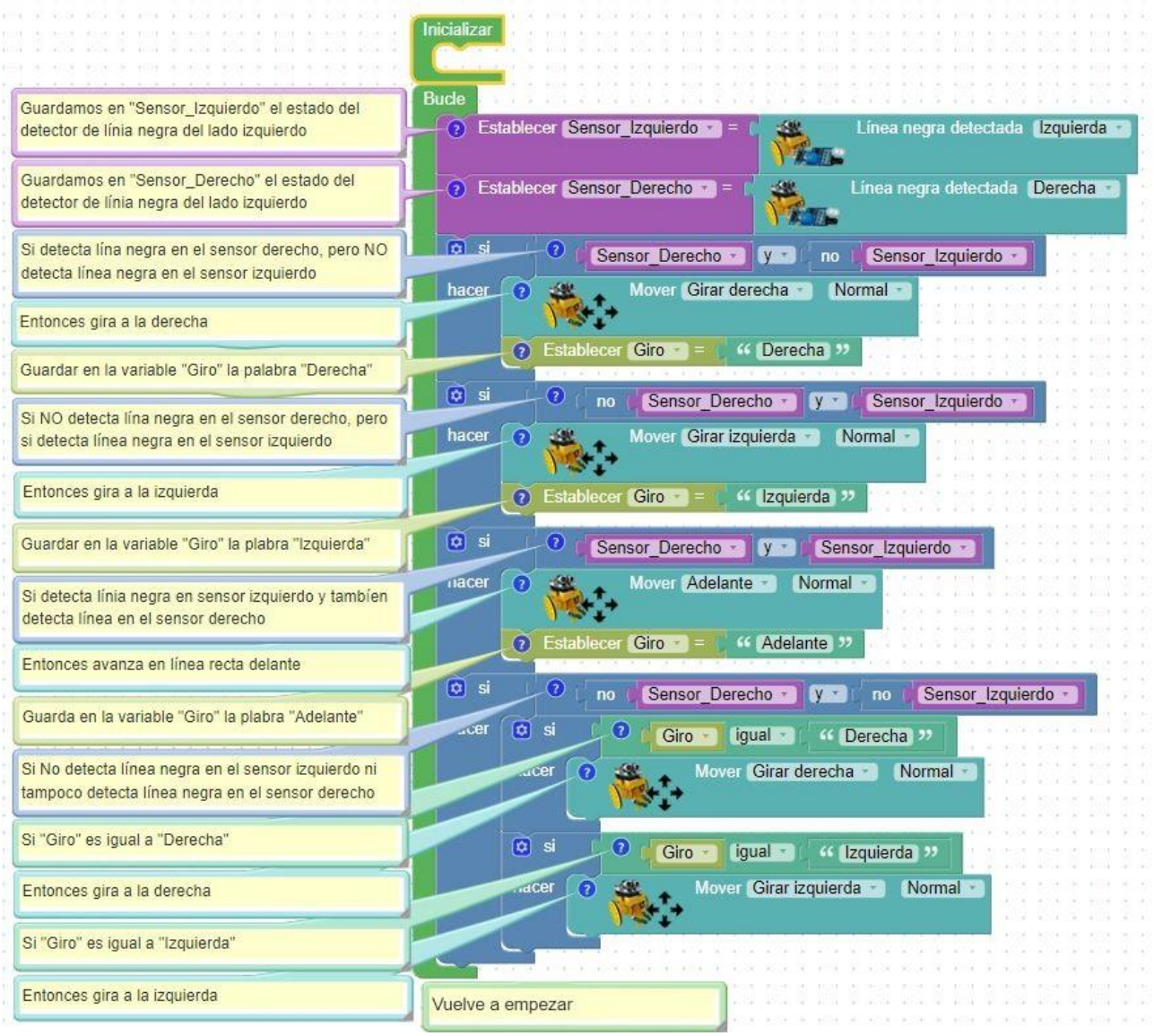


Es importante destacar que, dependiendo de la velocidad del robot y de lo cerradas que sean las curvas, es posible que en alguna de ellas el robot se pierda y se pare. Esto es porque, como decíamos, aún no le hemos dicho que hacer si ambos sensores detectan línea blanca a la vez. ¡Vamos a solucionarlo!

A015_02 Robot seguidor de línea Versión 02:

Si ahora se pierde, es decir, si se encuentran los dos sensores en línea blanca, comenzará a girar en el último sentido que lo ha hecho justo antes de perderse. Para ello, cada vez que ejecute un movimiento, tiene que memorizar que lo ha ejecutado.

Vamos a utilizar variables de texto para memorizar cuál ha sido el último movimiento realizado.



A16. – Control del robot por bluetooth

En esta actividad aprenderemos a controlar nuestro robot con un móvil Android a través del módulo bluetooth.

Bluetooth integrado

Como hemos visto antes del montaje de la placa Imagina Arduino encima de la placa Arduino Uno, el módulo bluetooth se encuentra en la parte inferior, podemos observar su funcionamiento a través del LED de actividad que esta justo debajo del símbolo “Bluetooth”.

*Los módulos bluetooth ya vienen configurados por defecto a 57600 Bauds.

A parte del módulo bluetooth instalado en el robot, para poder controlarlo con el móvil y trabajar con él, vemos en la siguiente imagen, que tenemos un interruptor que tiene la función de controlar si la placa Imagina Arduino del 3dBot se comunica con el ordenador o con el módulo bluetooth.

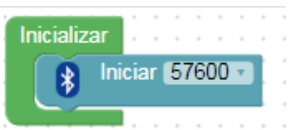
LED actividad e interruptor

Esto es fundamental, ya que para cargar el programa desde ArduinoBlocks, tendremos que poner el interruptor en la posición “PROG”, y tras esto, para comunicarnos con el robot mediante bluetooth, habrá que poner el interruptor en la posición “BT.ON”.

Bloque “Inicializar”:

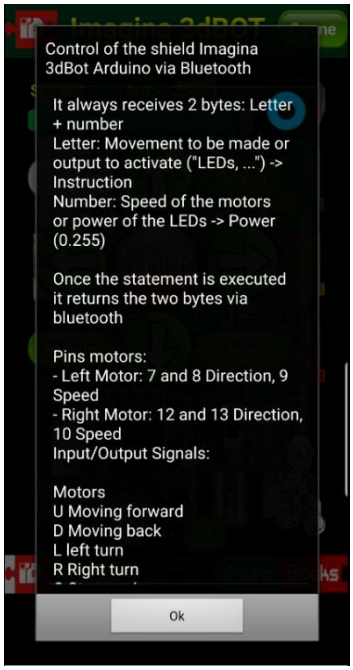
Lo primero que debemos hacer al empezar la programación, es fijarnos en el número que trae impreso el módulo, porque esa es la velocidad de transmisión de datos a la que está configurado.

Este dato es el único que tenemos que poner en la inicialización de parámetros. Encontramos el bloque en el conjunto de instrucciones “Bluetooth”:



Para la gestión de las órdenes en el móvil, disponemos de la App “Imagina 3dBot” en Google Play, una aplicación hecha expresamente para nuestro robot.

Para decidir qué letras o números son los que hacen avanzar o girar, tenemos que consultar el apartado de información de la App. Teniendo en cuenta dicha información, vamos a hacer un programa que lea los datos recibidos por bluetooth y ejecute un movimiento en función de la letra recibida.



Además de los bloques ya conocidos, vamos a leer los datos procedentes del bluetooth con el bloque “Recibir byte”, también ubicado en el apartado “Bluetooth”.

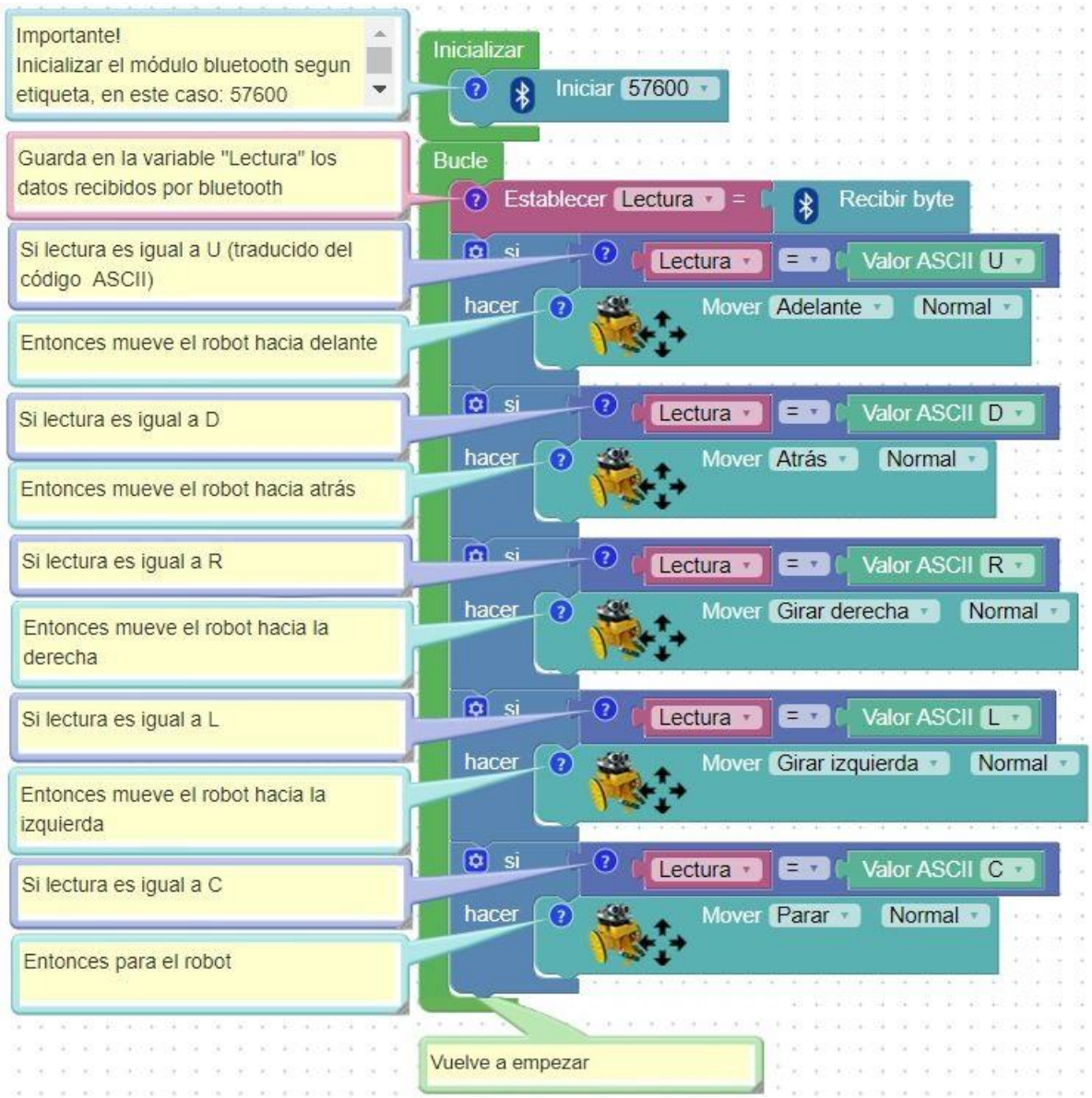
Trabajar con lecturas en bytes nos obliga a leer los caracteres en código ASCII. Este código es una forma de codificar en números, los caracteres y símbolos del lenguaje. Tranquilos porque ArduinoBlocks dispone de un bloque para hacer la traducción inmediata, por lo que no tendremos que hacer ningún paso extra.

Si la App indica que cuando pulsas la flecha de movimiento hacia delante, envía por bluetooth una *U*, por ejemplo, nosotros tendremos que seleccionar una *U* en la programación en ArduinoBlocks, mediante el siguiente bloque:

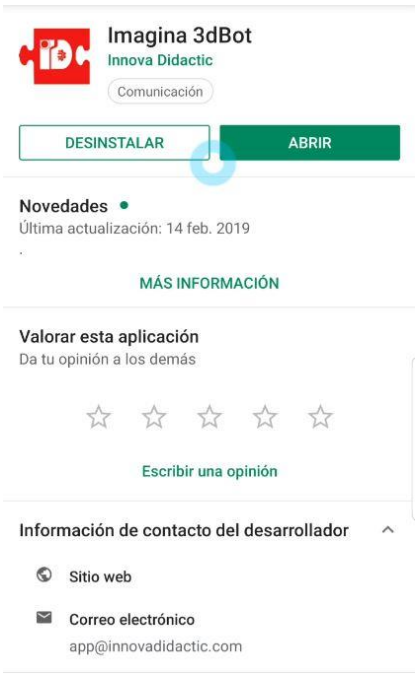
El bloque para traducir los datos leídos en bytes (*Valor ASCII)*, se encuentra al final del menú “Texto”.

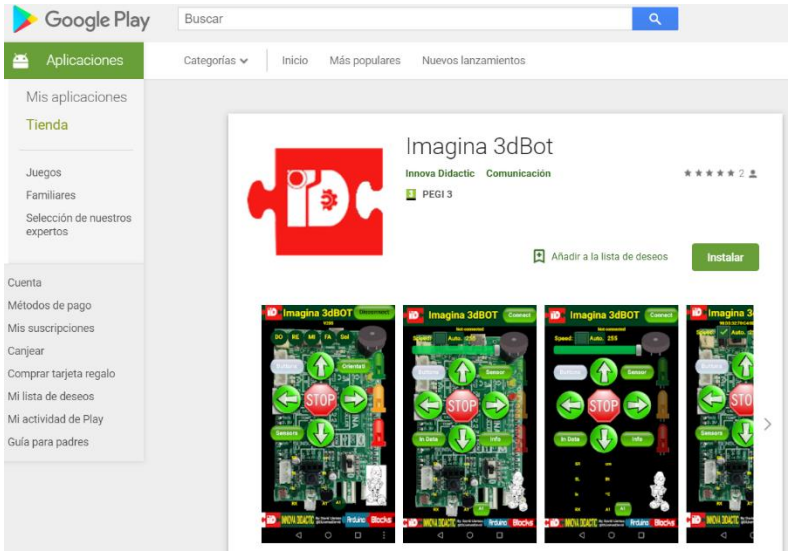
A16: Control del robot con el móvil.

Para la programación en ArduinoBlocks queda de la siguiente manera:

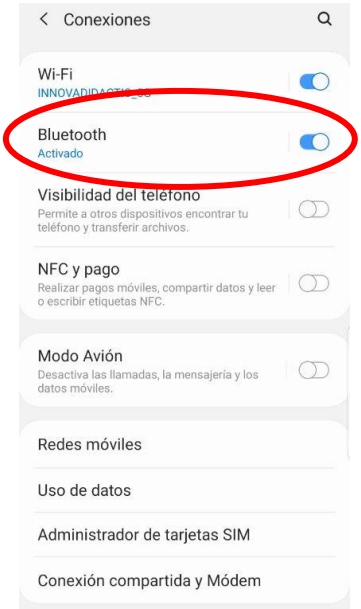


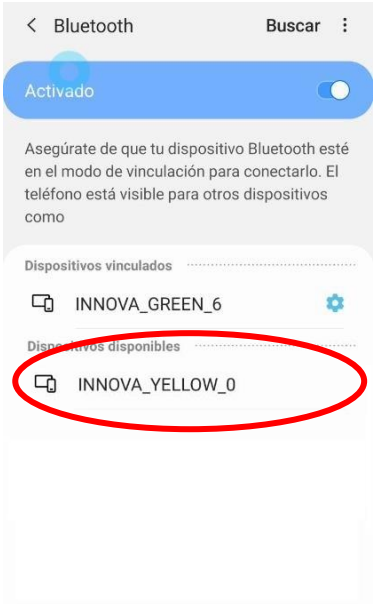
Y en el móvil instalaremos la App llamada Imagina 3dBot que nos descargaremos de GooglePlay.

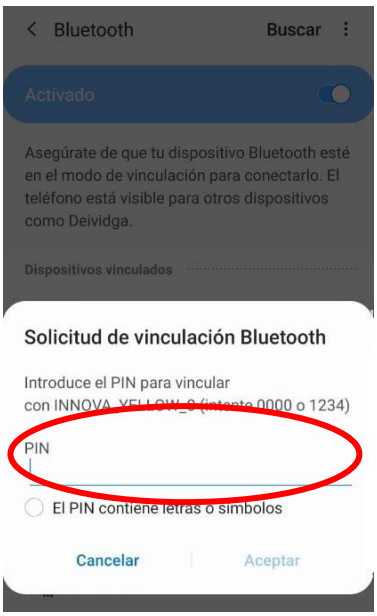


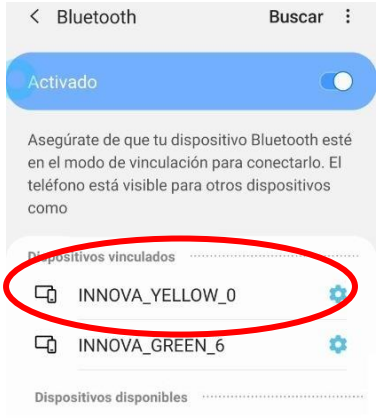


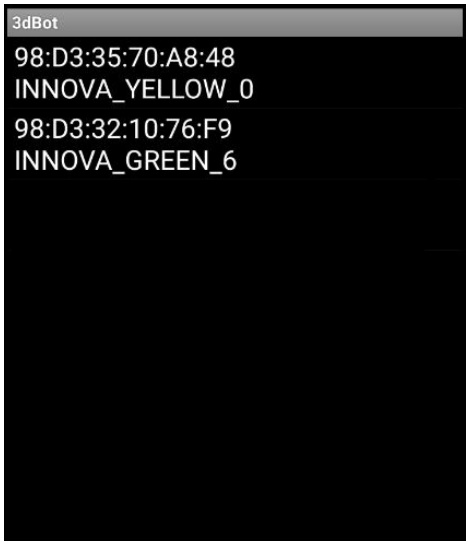
Para vincular el móvil con nuestro robot, activaremos la función bluetooth de nuestro teléfono. Cuando nos aparezcan los distintos “Dispositivos disponibles”, seleccionaremos nuestro robot según la etiqueta del módulo ubicado debajo de la placa, por ejemplo, INNOVA_YELLOW_0 e introducimos la contraseña 1234. Una vez aceptada ya debería aparecer el robot en “Dispositivos vinculados”.











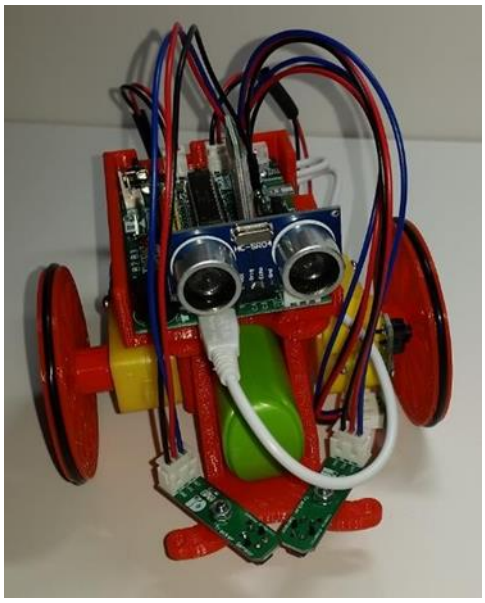
Ahora repetimos la operación de conexión en la App Imagina 3dBot, solo se debe pulsar sobre el botón “Connect” situado al lado del título de la aplicación, y volver a seleccionar el nombre de nuestro robot, “INNOVA_YELLOW_0”:

Video del control con botones tipo flechas: https://www.youtube.com/watch?v=SytjjWNLB7Q

Video del control con giroscopio: https://www.youtube.com/watch?v=WJ96BzCuQ

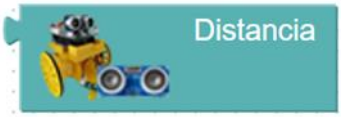
A17. – Robot explorador autónomo

En esta actividad equiparemos de “ojos” al robot con el sensor de ultrasonido HC-SR04 y haremos que funcione autónomamente explorando su entorno sin chocar. Para evitar el choque iremos haciendo varias estrategias, de la más simple a la más complicada.

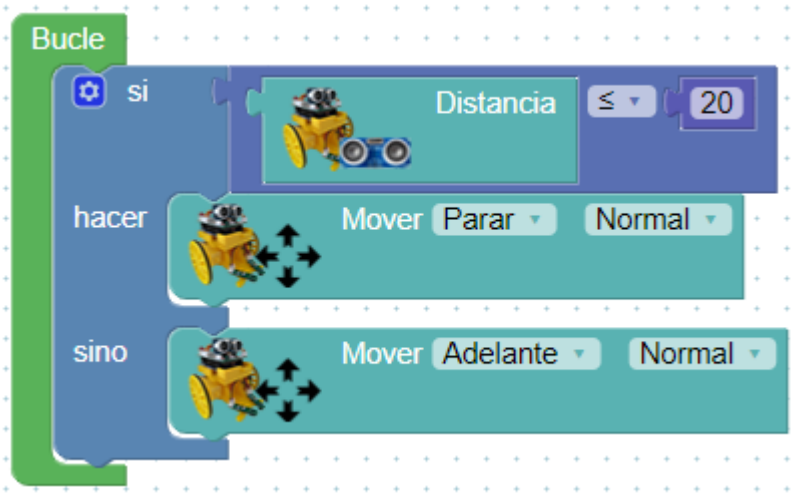


Primero haremos un programa para que el robot siempre vaya hacia adelante y que cuando encuentre un objeto, a menos de 20cm, se pare. Y si el objeto desaparece debe volver a avanzar.

El bloque se encuentra en el apartado “3dBot” de ArduinoBlocks:

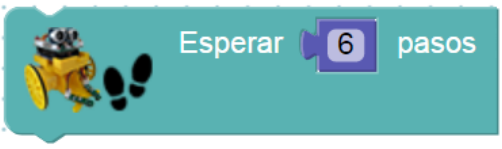


A17_01 Prueba el siguiente programa:



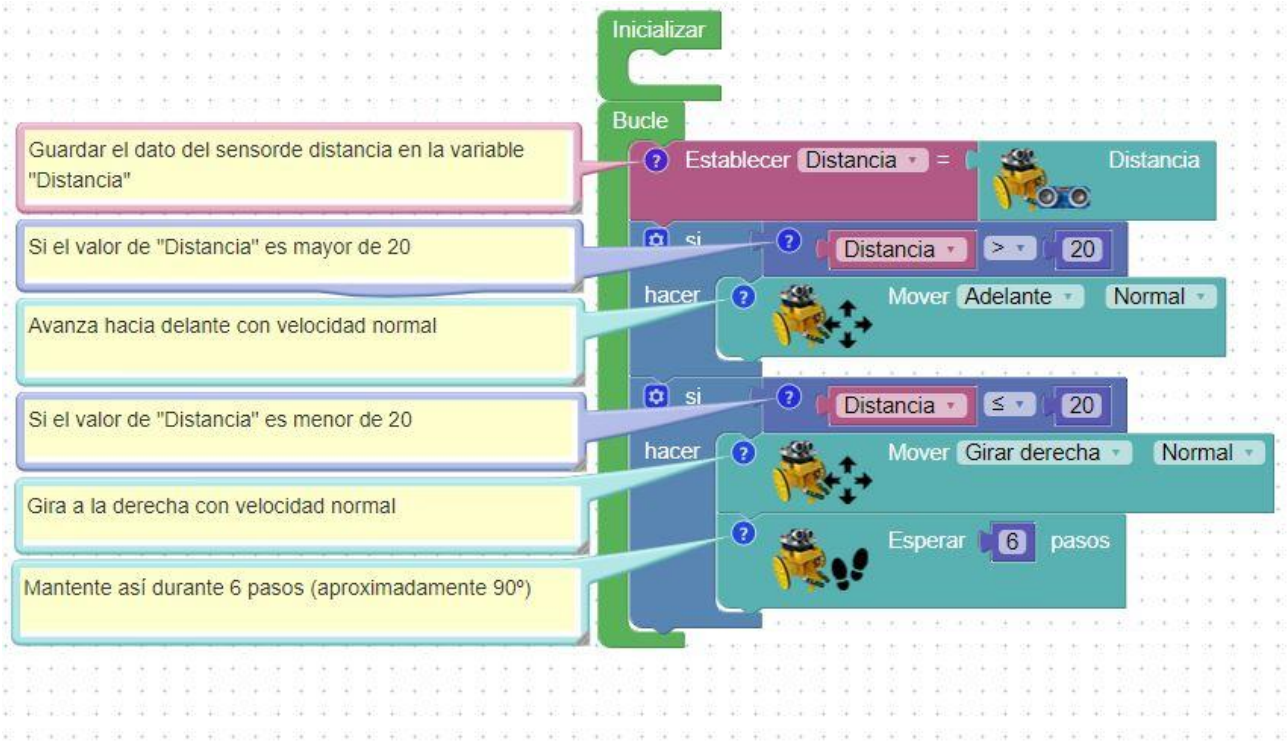
A17_02 La segunda estrategia consistirá en hacer que cuando se pare, gire a la derecha 90 grados y siga adelante. Para realizarlo va a ser especialmente importante utilizar el encoder que hemos explicado en la actividad A13.

Hemos visto también en la actividad A13 que, para girar 90 grados a la derecha, basta con girar a la derecha durante 6 pasos (aproximadamente).

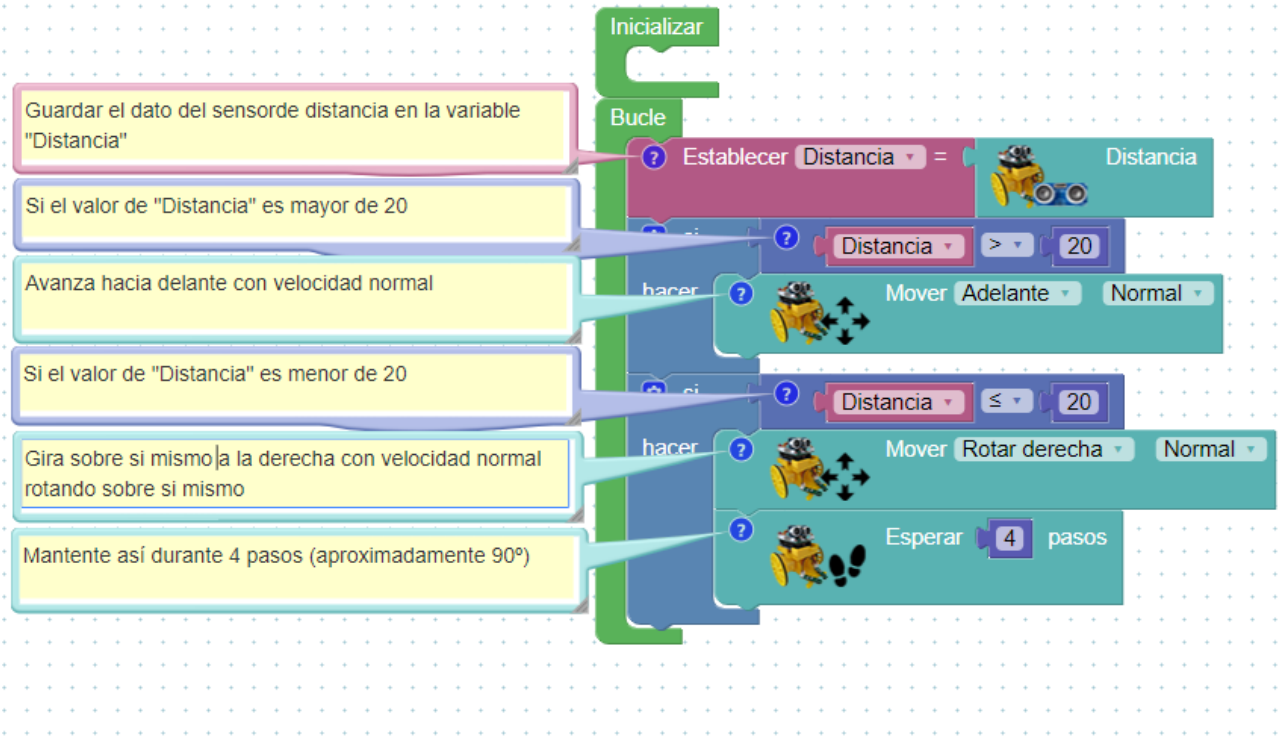


En lugar de la función “girar”, que sólo mueve una rueda, se puede utilizar la función “rotar”, en la que giran ambas ruedas en sentido contrario. Vamos a ver los dos ejemplos.

- **Programa utilizando el giro:**



- **Programa utilizando la rotación**



Si comparamos ambos ejemplos, vemos que el número de pasos necesario para rotar 90 grados no es el mismo que para girar 90 grados. Podéis probar con distintos valores para ver los resultados y así marear un poco al 3dBot.

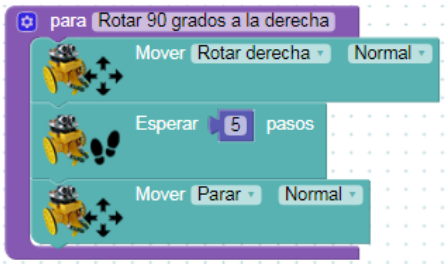
También es muy importante tener en cuenta que solo disponemos de encoder en la rueda izquierda y que en los giros a izquierdas esta no se mueve, por tanto, en ese giro no podremos controlar el ángulo girado. Como en las rotaciones si se mueven ambas ruedas, ahí si podemos medir el ángulo girado a izquierdas.

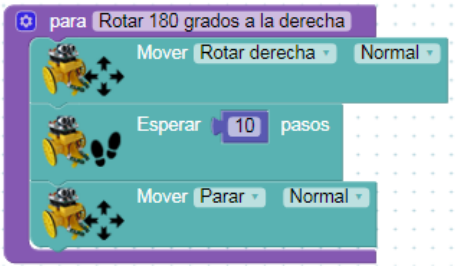
Por último, se puede cambiar la distancia mínima a la que detecta un obstáculo.

A17_03: Robot explorador autónomo en laberinto.

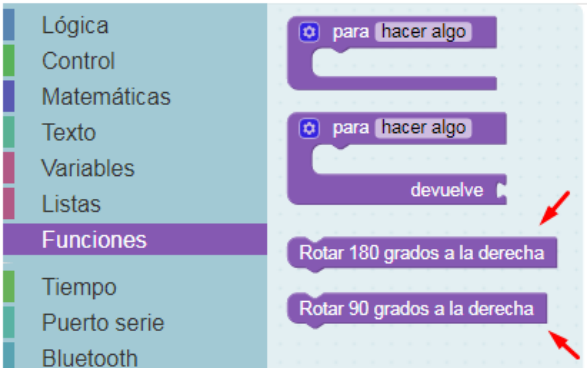
En el programa anterior, el robot solo puede esquivar obstáculos en una dirección, lo que provoca que cuando la única dirección posible es la izquierda, el robot se pierde o vuelve por donde ha venido.

Para evitarlo, vamos a utilizar dos rotaciones diferentes, una de 90 grados y otra de 180 grados. Estos dos movimientos los vamos a definir, por comodidad, en las siguientes funciones:





Tras esto, si vamos al apartado “Funciones”, veremos las nuevas funciones creadas y listas para insertar en cualquier parte de nuestro programa:



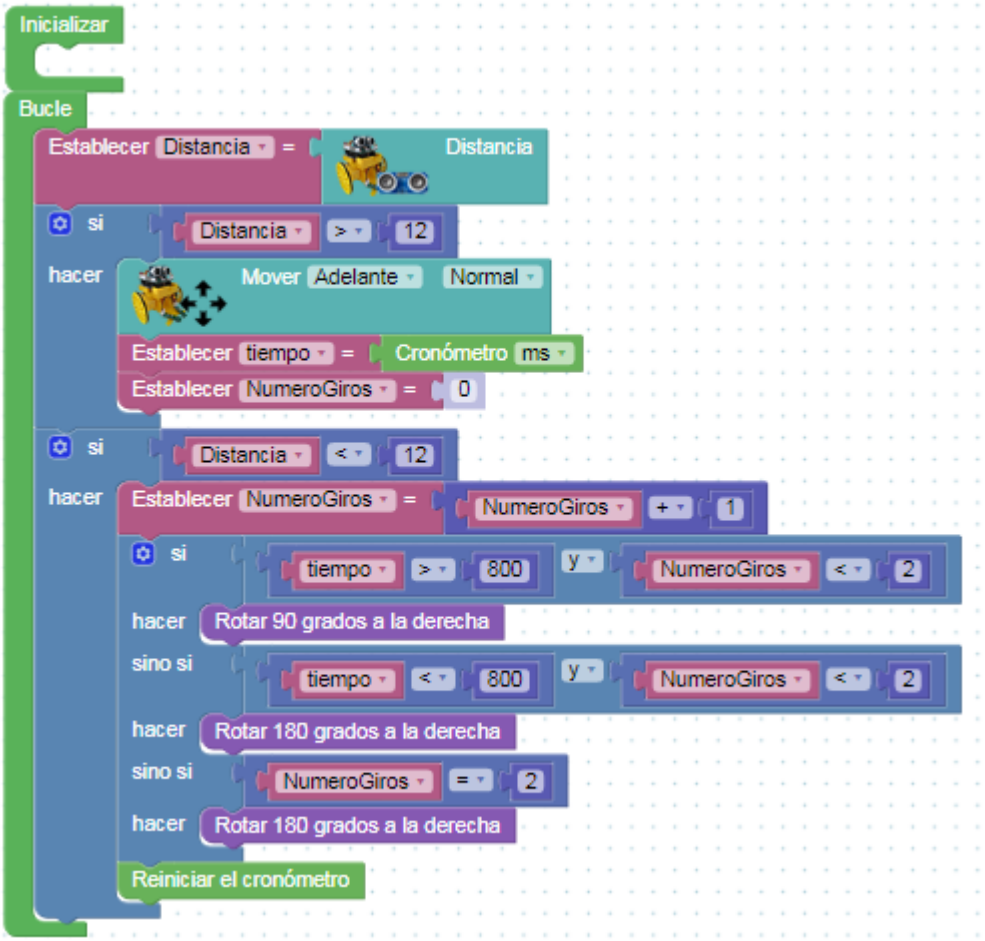
Además, vamos a introducir nuevas condiciones que le den a nuestro robot más herramientas a la hora de enfrentarse a obstáculos.

Como antes el robot solo giraba a la derecha, esto provocaba que, cuando la única dirección posible era a la izquierda, el robot se perdía o volvía por donde había venido.

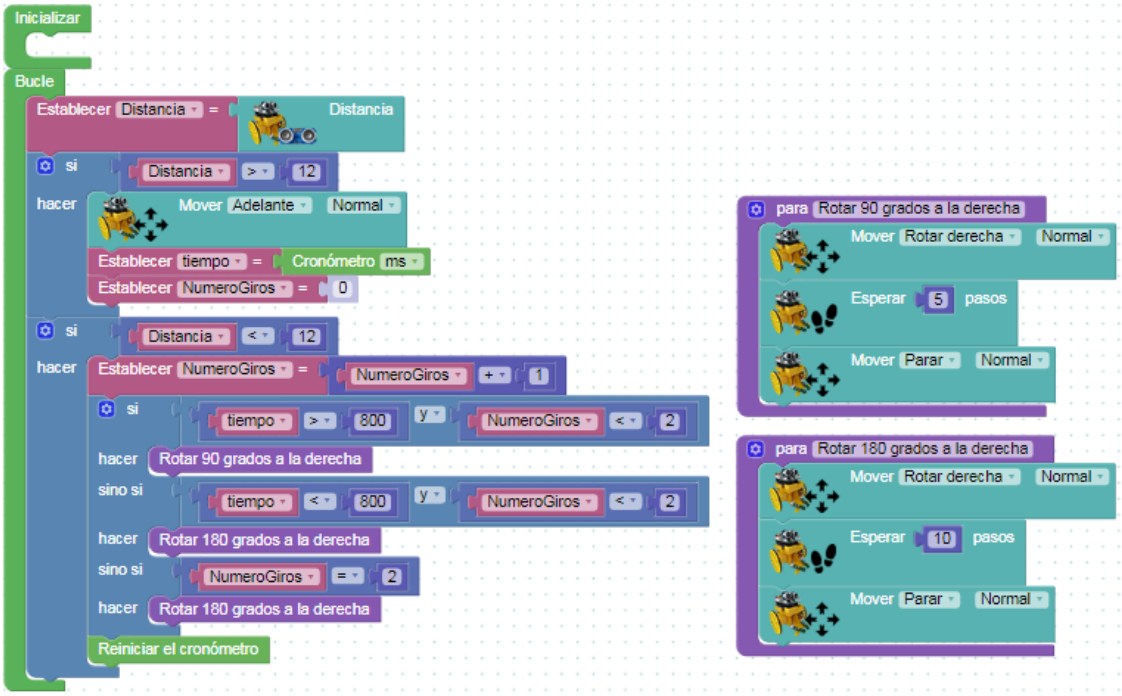
Insertar imagen superior del 3dbot en un laberinto con giro a la izquierda

Para evitarlo, vamos a decirle que si gira a la derecha e inmediatamente vuelve a encontrar otro obstáculo y va a volver a girar a la derecha, querrá decir que realmente tenía que haber girado hacia la izquierda (es lo más probable). Teniendo en cuenta que acaba de realizar un giro de 90 grados en la dirección equivocada, le diremos que gire 180 grados para ir en la dirección opuesta.

Veremos en el programa que, para hacer esta condición explicada más efectiva, el robot va a tomar la decisión descrita, no solo cuando haya interpretado dos giros **inmediatamente seguidos** a la derecha, sino también cuando los haya interpretado en menos de 800 ms, aunque en ese tiempo sí haya avanzado brevemente en línea recta.



Recordad dejar en la misma pantalla las funciones antes definidas. La definición de estas funciones no tiene que ir dentro de bucle. En total queda algo así:


Inicializar
Bucle
Establecer Distancia = Distancia
si Distancia > 12
hacer Mover Adelante Normal
Establecer tiempo = Cronómetro ms
Establecer NumeroGiros = 0
si Distancia < 12
hacer Establecer NumeroGiros = NumeroGiros + 1
si tiempo > 800 y NumeroGiros < 2
hacer Rotar 90 grados a la derecha
sino si tiempo < 800 y NumeroGiros < 2
hacer Rotar 180 grados a la derecha
sino si NumeroGiros = 2
hacer Rotar 180 grados a la derecha
Reiniciar el cronómetro
para Rotar 90 grados a la derecha
Mover Rotar derecha Normal
Esperar 5 pasos
Mover Parar Normal
para Rotar 180 grados a la derecha
Mover Rotar derecha Normal
Esperar 10 pasos
Mover Parar Normal

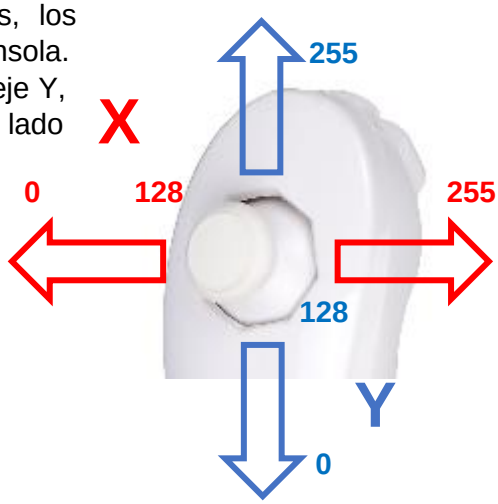
A18. – Control del robot con Nunchuk (Wii)

* El Nunchuk es un accesorio opcional no incluido en el Kit Básico.

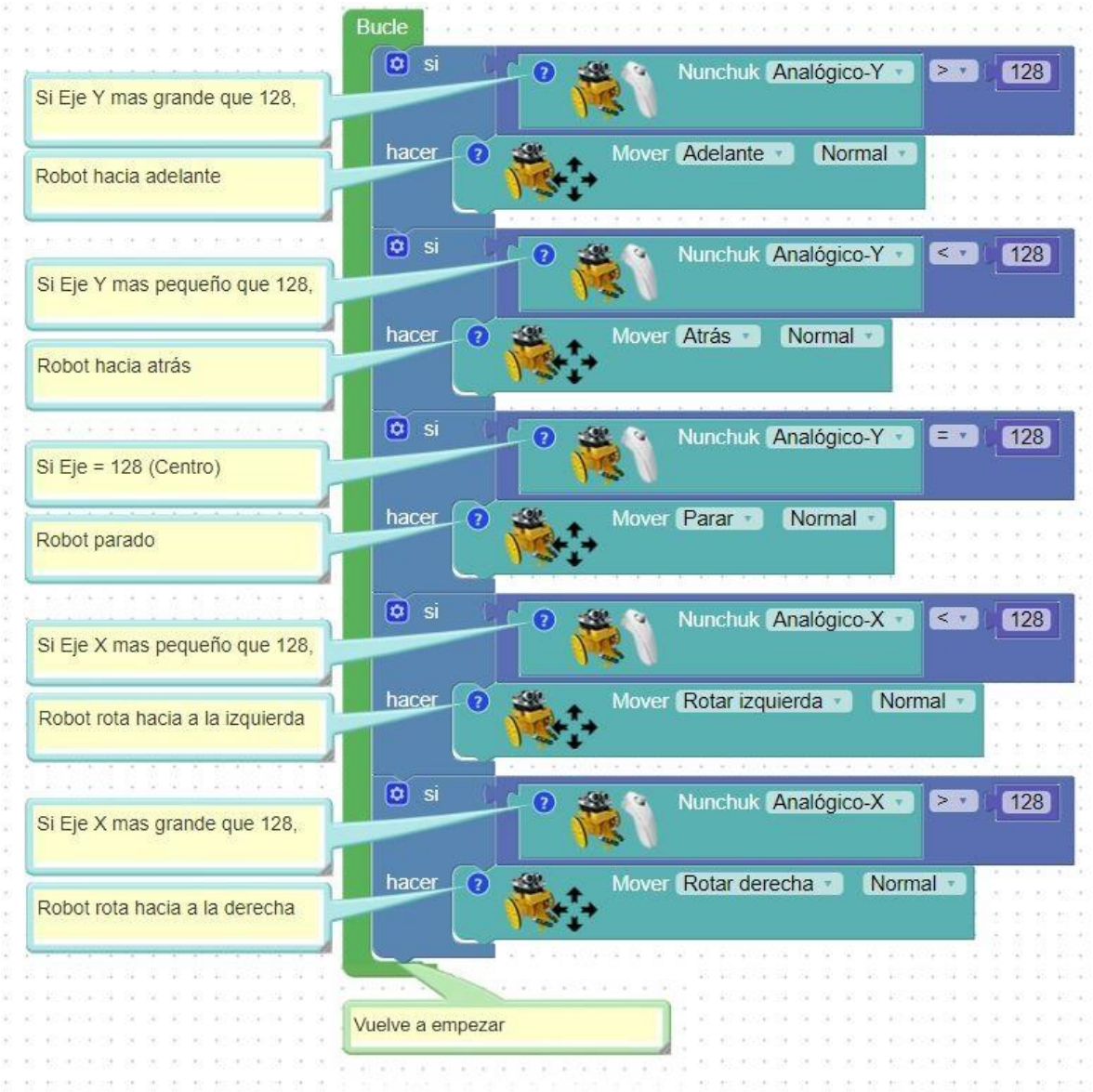
El mando Nunchuck nos servirá para poder controlar nuestro 3dBot con su joystick, se conecta a la placa con el siguiente conector:

Nunchuck

Para usar el Joystick tenemos que saber los valores, los podemos leer como hemos visto anteriormente en la consola. Como referencia saber que tanto en el eje X como en el eje Y, el valor en el punto central es 128, hacia un lado 0 y al lado contrario es 255.

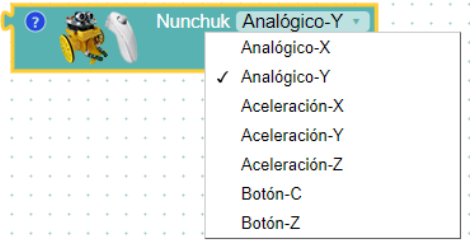


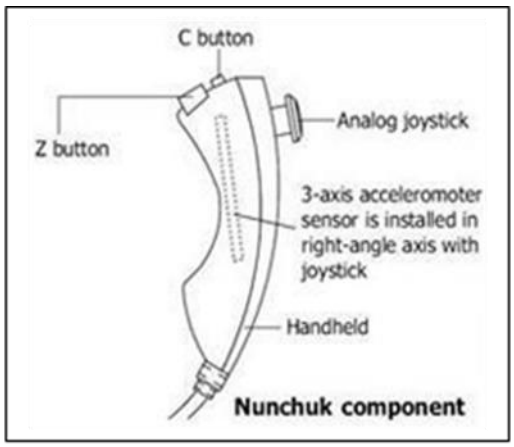
Vamos a hacer un programa básico para poder controlar el robot hacia adelante y atrás, y girar sobre sí mismo:



El mando Nunchuk, a parte del Joystick, dispone de dos botones C y Z, más un a acelerómetro de 3 ejes. Utilizando la función “Enviar” podemos descubrir los distintos valores de los distintos actuadores.

¿Te atreves a preparar un programa para poder controlar el robot solo con el movimiento de la mano?





Bonus track.audio

Las notas musicales son, simplemente, sonidos con una frecuencia determinada. Por tanto, conociendo la frecuencia correspondiente a cada nota, y configurando los tiempos adecuados de cada una (junto con los silencios) podemos programar una melodía.

Frecuencia de las notas musicales en Hertzios

		0	1	2	3	4	5	6	7	8
n=1	do		32,7	65,41	130,81	261,63	523,26	1046,5	2093	4186
n=2	do#		34,65	69,3	138,59	277,18	554,37	1108,7	2217,5	4434,9
n=3	re		36,71	73,42	146,83	293,66	587,33	1174,7	2349,3	4698,6
n=4	re#		38,89	77,78	155,56	311,13	622,25	1244,5	2489	4978
n=5	mi		41,2	82,41	164,81	329,63	659,26	1318,5	2637	5274
n=6	fa	21,826	43,65	87,31	174,61	349,23	698,46	1396,9	2793,8	5587,7
n=7	fa#	23,125	46,25	92,5	185	369,99	739,99	1480	2960	5919,9
n=8	sol	24,5	49	98	196	392	783,99	1568	3136	6271,9
n=9	sol#	25,96	51,91	103,83	207,65	415,3	830,61	1661,2	3322,4	
n=10	la	27,5	55	110	220	440	880	1760	3520	
n=11	la#	29,14	58,27	116,54	233,08	466	932,33	1864,7	3729,3	
n=12	si	30,87	61,74	123,47	246,94	493,88	987,77	1975,5	3951,1	

Bonus track: “Marcha imperial de Star Wars”.

Las notas de la parte programada de la canción son:

- Función *Primera parte*: La, La, La, Fa, Do, La, Fa, Do La
- Función *Segunda parte:* Mi, Mi, MI, Fa, Do, Lab, Fa, Do, La

Se ha realizado con un pulso de 600 ms.

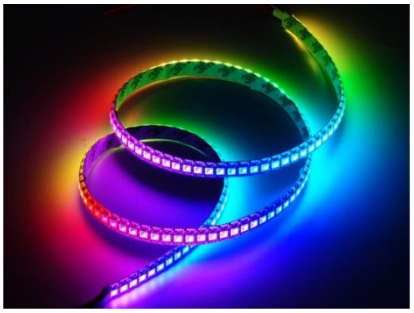
A este programa se le pueden añadir movimientos del robot, para que “baile” al ritmo de su propia música.

Bonus track.iluminación

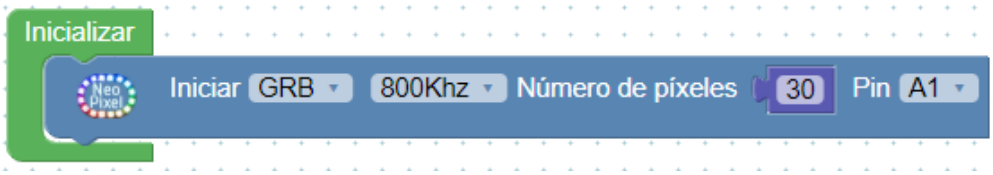
[* La Tira de leds Neopixel es un accesorio opcional no incluido en el Kit Básico.]

Las tiras de leds Neopixel son unas tiras compuestas por varios leds RGB que son direccionables individualmente. Esto quiere decir que podemos encender el led que deseemos con el color que queramos.

En realidad, se pueden encontrar en forma de tiras, aros, matrices…



En el menú “NeoPixel” tenemos el bloque para inicializar la tira o elemento, hay que ponerlo dentro del bucle “Inicializar”.

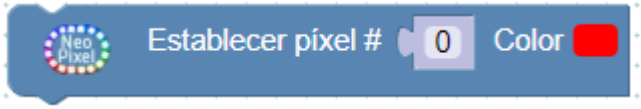


Y las distintas funciones para poder crear juegos de luces:

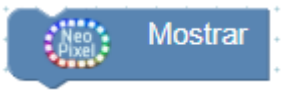
Dejar la tira, aro, matriz con todos los leds apagados

Encender el led seleccionado en el primer valor, con el color según valores R G B.

Parecido al anterior, pero esta vez podemos seleccionar el color directamente.



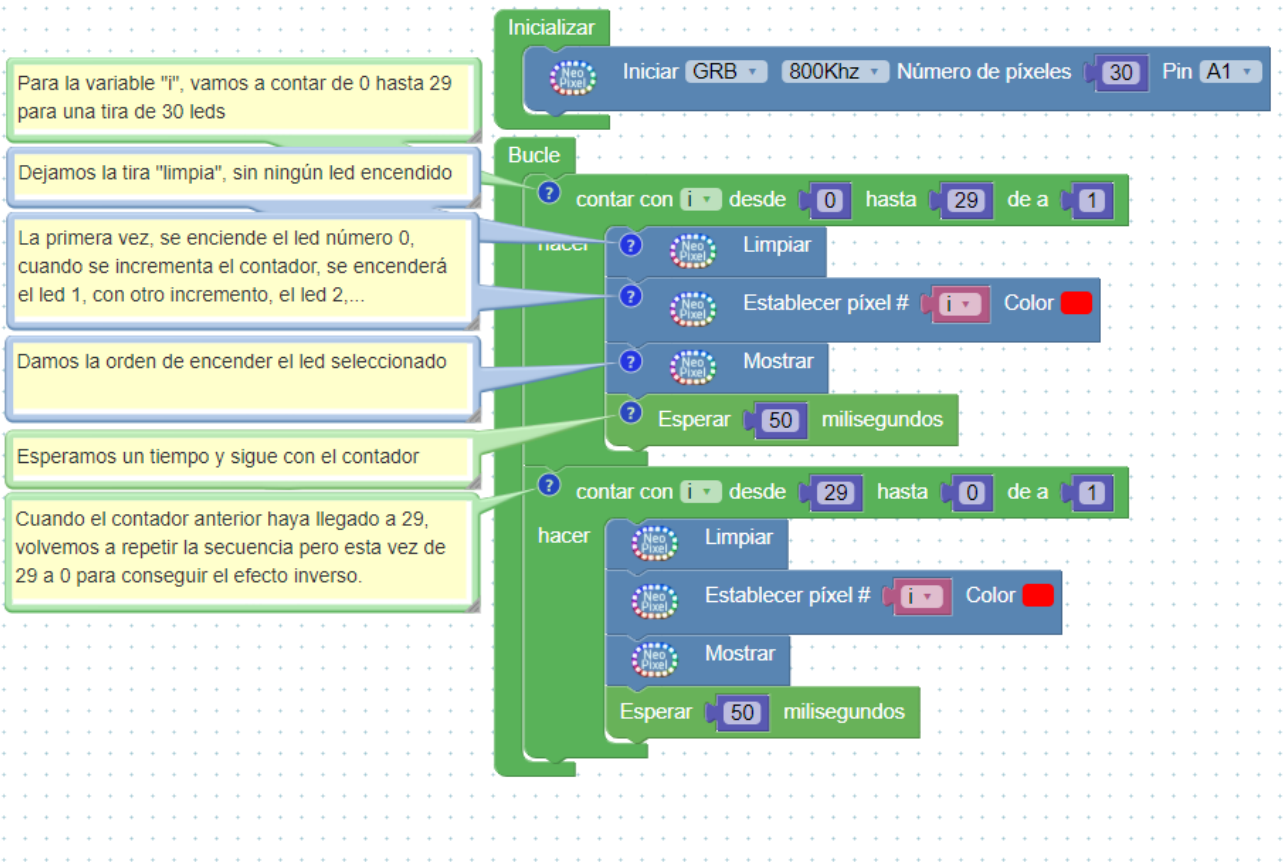
Bloque para hacer que lo programado se muestre en el elemento.



Aparte de estos, que son los esenciales, hay algún otro para matrices, crear dibujos… pero ahora vamos a lo interesante i a practicar con los comentados:

Bonus track: “El coche fantástico”.

Utilizando una tira de leds Neopixel vamos a crear la mítica luz del coche fantástico.



La función “contar con” se encuentra en el menú de “Control”. Con ella lo que vamos a hacer es incrementar el valor de una variable, en este caso “i” un número determinado de veces. Se ejecuta esta función hasta que llega al valor final, en el ejemplo del coche fantástico, hasta 29, luego el programa sigue con la siguiente instrucción.

Lo que conseguimos es que mientras el contador va incrementando el valor de la variable “i”, esta se usa para encender un determinado led. Dentro de la función contador hacemos tres acciones, dejamos la tira limpia, seleccionamos el led y lo hacemos mostrar en la tira, cada vez con un valor de “i” más alto.

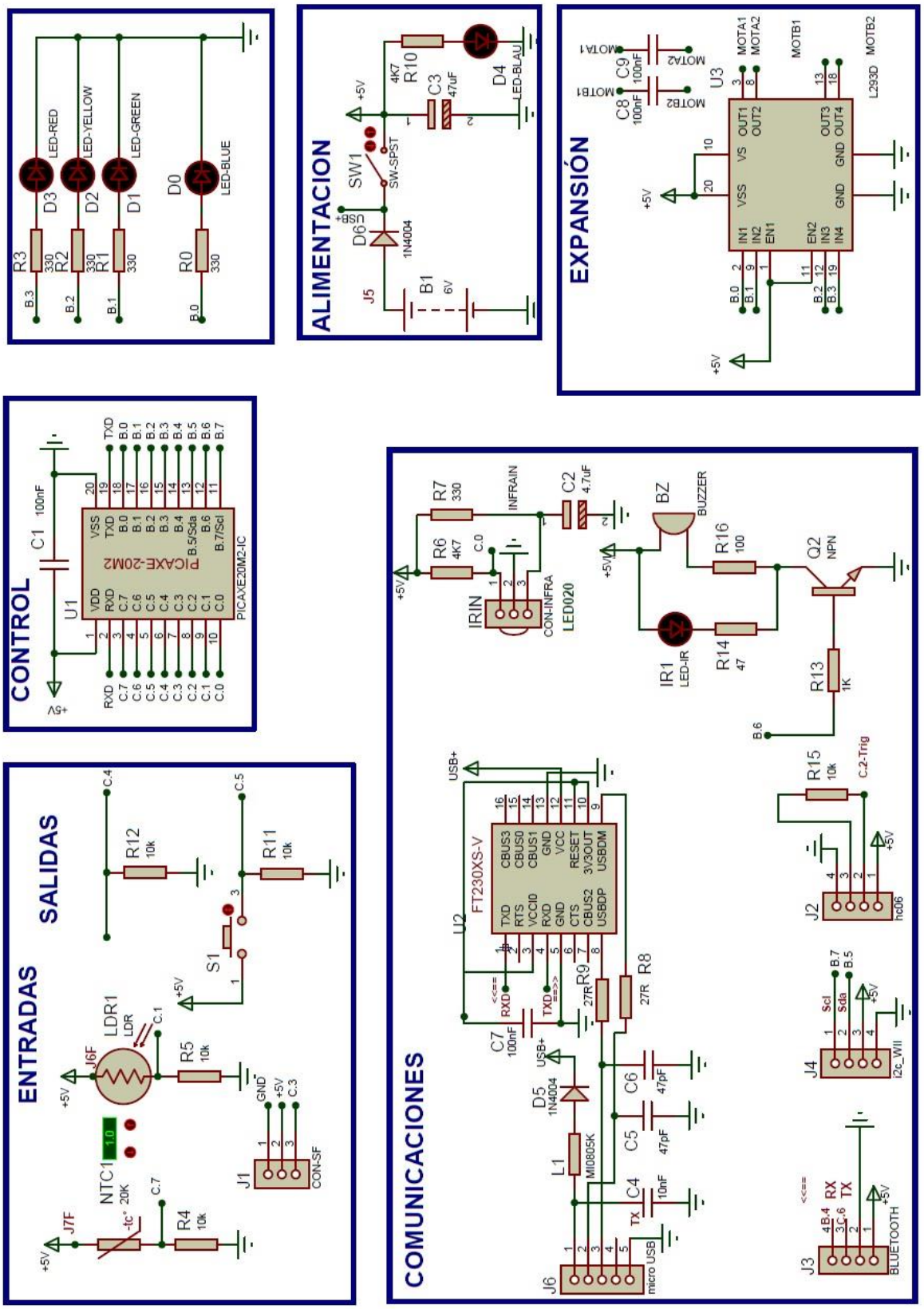
Luego repetimos decreciendo el contador.

¡Hasta aquí el tutorial para el robot Imagina 3dBot Arduino!

Lo que hemos visto es solo una pincelada del mundo Arduino. Hay infinidad de proyectos para crear con la gran cantidad de sensores y actuadores que existen. Consulta los Kits para primaria, secundaria y profesional disponibles en Innova Didactic.

Material Arduino

Anexo. – Esquema electrónico de la placa Imagina



Anexo. – Esquema electrónico de la placa Imagina