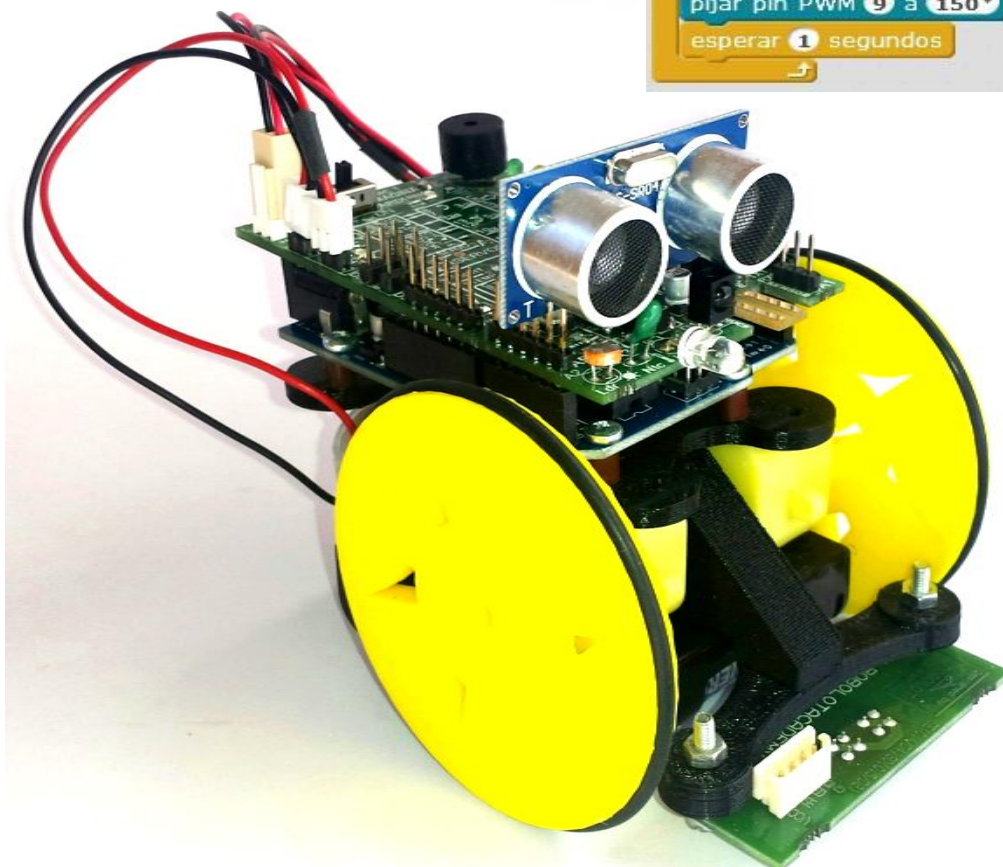


Actividades con mBlock, IDE y Arduino3dBot



```

al presionar [bandera]
por siempre
  set digital pin 7 output as HIGH
  set digital pin 8 output as LOW
  pijar pin PWM 9 a 0
  esperar 1 segundos
  pijar pin PWM 9 a 150
  esperar 1 segundos
  
```





ÍNDICE

Introducción 3

Material 3

Herramientas 4

¿Cómo funciona un robot? 4

mBlock 5

Actividades (Modo1: cable USB conectado)..... 6

A1.-Encender un LED 6

A2.-Encender un LED con el pulsador..... 8

A3.-Medir la luminosidad con LDR (sensor de luz)..... 9

A4.-Medir la temperatura con NTC (sensor de temperatura) 10

A5.-Medir la temperatura en grados Celsius..... 11

Actividades (Modo2: cable USB desconectado)..... 12

A6.-Generador de notas musicales con el zumbador 12

A7.-Controlar un motor CC (on, off, inversor de giro) 14

A8.-Control de dos motores CC..... 18

A9.-Robot rastreador..... 19

A10.-Robot explorador con ultrasonidos 21

A11.-Mide distancias con ultrasonidos (HC-SR04) 23

A12.-Control por infrarrojos (con IDE)..... 26

A13.-Control por Bluetooth 29

A14.-Control servomotor 31

A15.-Alarma intrusos (sensor PIR) 32

A16.-Sonda temperatura DS18B20 inox..... 34

A17.-Nunchuk 36

Anexo 1.-Montaje del robot Imagina3dBot 38

Anexo 2.-Posibilidades de la placa imagina Arduino 41

Anexo 3.-Programación de picaxe con mBlock 42

Anexo 4.-Esquema electrónico del Shield Imagina Arduino 45

Introducción

En este manual os proponemos una serie de actividades relacionadas con la robótica educativa utilizando el robot Arduino3dBot.

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

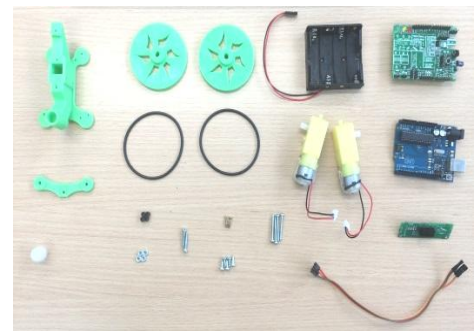
Aunque el robot Imagina Arduino se pueda programar en lenguaje C, en este manual se explica cómo hacerlo de manera gráfica con [mBlock](#), que es una versión modificada del Scratch 2.0. De todas formas siempre se tiene acceso a la conversión en C para ir aprendiendo este lenguaje.



Material

El kit robot Imagina3dBot básico ref. RBL0960 incluye:

- 1 chasis y 2 ruedas impresas en 3D.
- 2 motores cableados (Ref. bo1_120).
- 1 rueda loca y 2 juntas tóricas.
- 1 placa de control Arduino Uno rev3 compatible con cable USB.
- 1 placa driver de motores Shield Imagina Arduino (Ref. rbl0692_V2).
- 1 accesorio sensores de línea cny70 con cable y conectores (Ref. rbl0672-rast).
- 1 porta pilas plano 4xAA. (Pilas no incluidas).
- 1 juego de tornillos, tuercas, roscas y separadores.



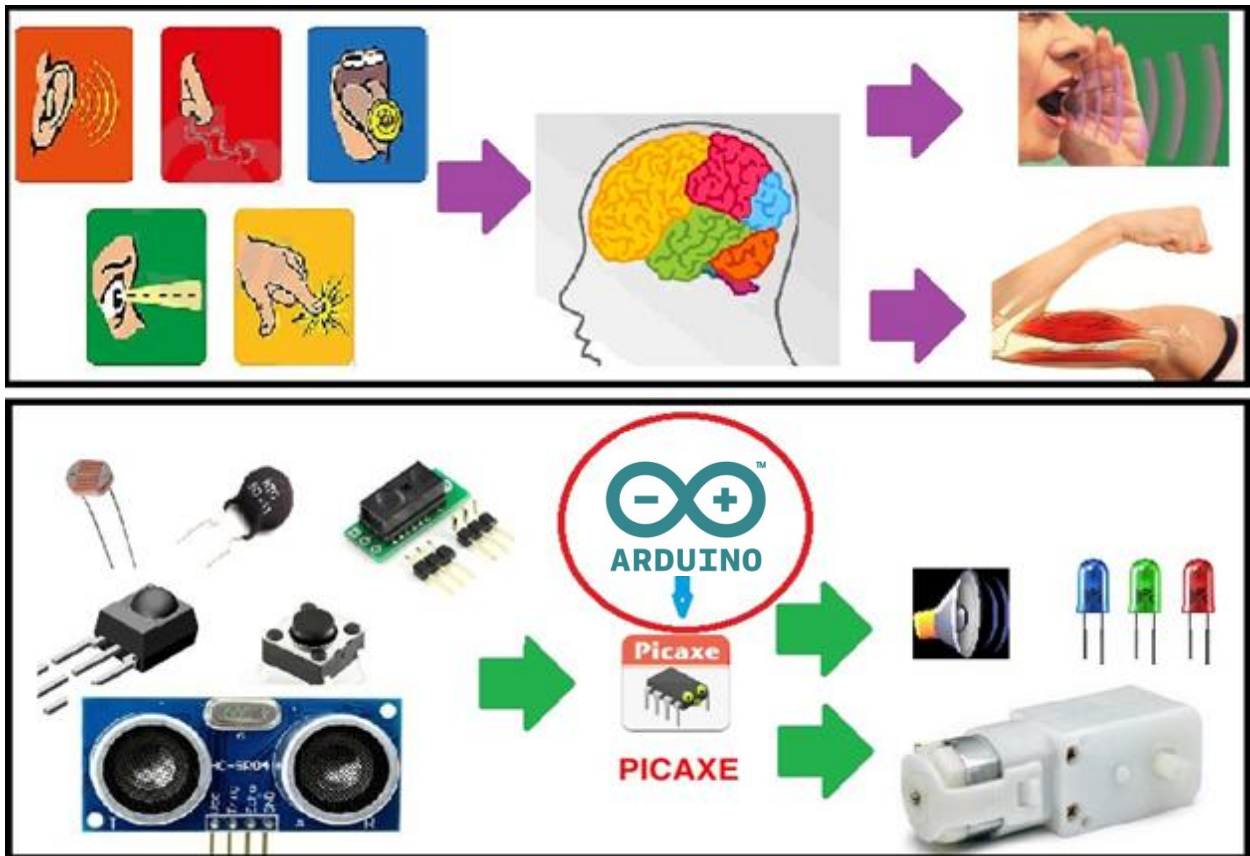
Herramientas

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



¿Cómo funciona un robot?

Los robots funcionan de una manera similar a nosotros. Nuestro cerebro recibe información de nuestros sentidos (oído, olfato, gusto, vista y tacto), analiza esta información y da estímulos a las cuerdas vocales para emitir sonido o da órdenes a los músculos para que estos se muevan. Los 5 sentidos serían las entradas de información y, la voz y los músculos serían las salidas, sensores y motores.



En el robot, un chip hace la función de cerebro. Este chip es un microcontrolador y tiene 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 tiene que hacer.



Entonces nosotros tenemos que decirle lo que queremos que haga, según las señales de los sensores. ¡A esto se llama programar!

mBlock

El "mBlock" es un entorno de programación desarrollado por el equipo Makeblock basado en Scratch. Está pensado para que niños y niñas aprendan a programar a partir de los 8 años.



Cada personaje (animación) tiene un programa asociado, donde las instrucciones tienen forma de rompecabezas. Esto permite aprender a programar muy rápido y de forma muy divertida, creando juegos, historias, animaciones, simulaciones, música, arte, mappings...

Los programas se activan al pulsar encima de la bandera verde i se paran al pulsar encima del disco rojo.  

MBlock añade algunos bloques originales relacionados con el hardware original de Scratch, con estos bloques, el usuario puede leer sensores, controlar motores e incluso controlar todo el robot.



Además de bloques para las funcionalidades básicas de microcontroladores, analógicas, digitales (lectura y escritura de estas) y salidas PWM. También hay bloques específicos, tales como sensores de ultrasonidos, sensores de temperatura, sensores de luz, driver de motor DC, driver de motor paso a paso, etc. Con estos bloques es simple para interactuar con muchos tipos de módulos electrónicos.

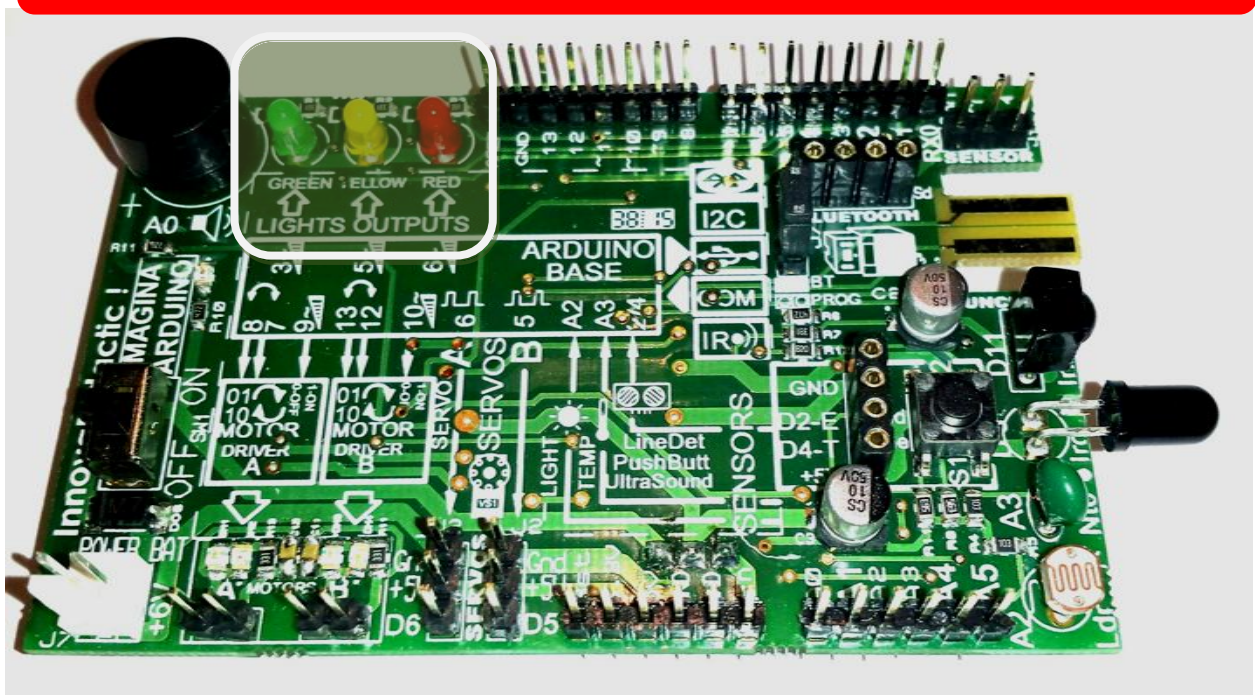
El mBlock es una modificación del programa Scratch 2.0 versión offline. Es una versión personalizada de Scratch (desarrollado por el Lifelong Kindergarten Group del MIT Media Lab.) Con mBlock aquellos que estén familiarizados con Scratch podrán, de manera sencilla, interactuar con módulos electrónicos de Arduino.

Actividades (modo1: cable USB conectado)

A continuación os proponemos una serie de actividades para aprender, paso a paso, cómo funcionan los sensores de vuestro robot y como darle "vida".

Estas actividades se proponen utilizando la placa permanentemente conectada al ordenador utilizando el cable de programación (en mode1).

A1.-Encender un LED.



En esta primera actividad aprendemos a encender el LED amarillo y apagarlo.

La placa de control Arduino Uno tiene 13 entradas/salidas digitales disponibles. Generalmente se llaman Dx, donde la x es un número comprendido entre 0 y 13.

La placa Imagina Android dispone de 3 LEDs que están conectados a las salidas de la siguiente manera: verde (D3), amarillo (D5) y rojo (D6).

Conectamos nuestro robot con el cable de programación habiendo instalado previamente los *drivers* de la placa Arduino Uno para que el ordenador la reconozca y le asigne un puerto COM.

En la categoría *Robots* aparecerá una *Extensión de Arduino*, con todas las instrucciones disponibles para programarlo, tal como:



Introduciremos el programa siguiente:



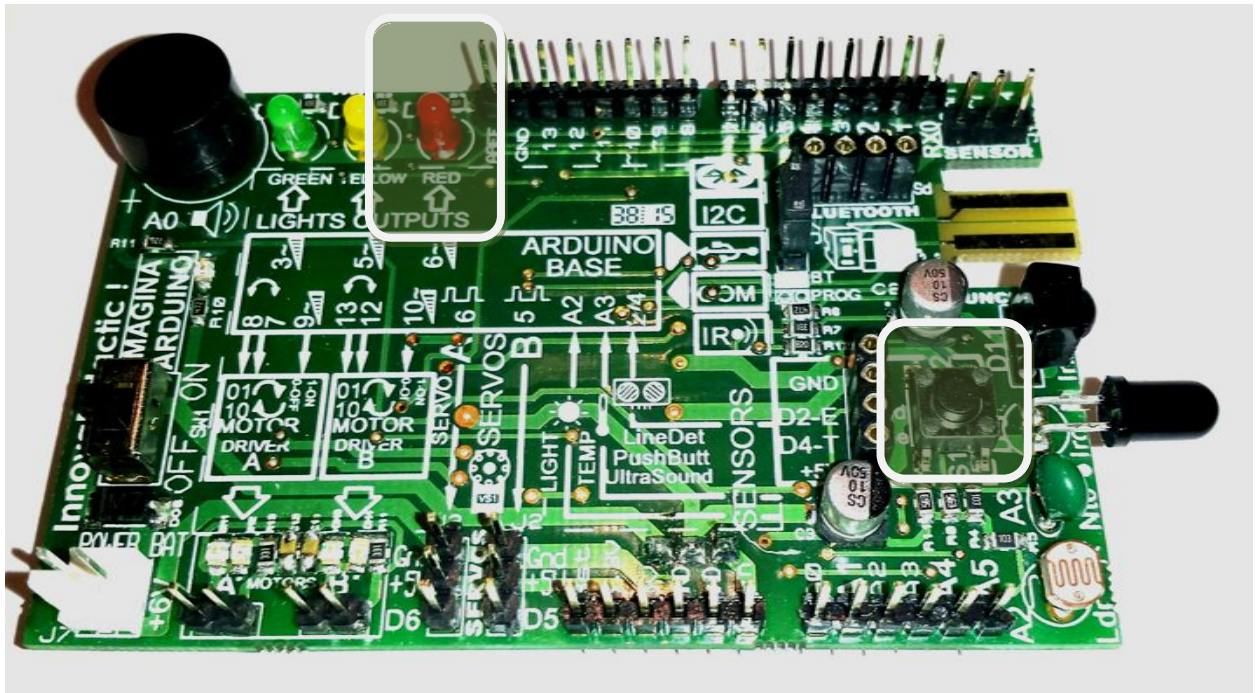
Comprobaremos el funcionamiento haciendo clic sobre la bandera verde.

¿Qué pasa si aumentamos el tiempo de espera? ¿Hace la intermitencia más rápida? ¿O es al revés?

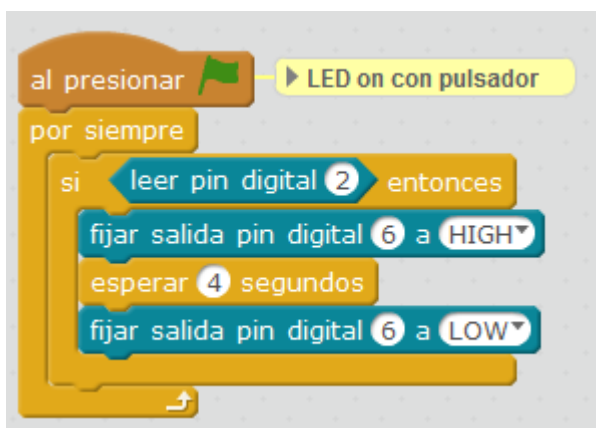
¿Cómo hay que modificar el programa para hacer que el LED verde esté 2 segundos encendido y 4 apagado?

A2.-Encender un LED con el pulsador.

En esta actividad queremos encender el LED rojo (D6) durante 4 segundos al presionar el pulsador (D2) de la placa.



Introduciremos el programa siguiente y pondremos comentarios a todas las instrucciones explicando brevemente su funcionalidad:



Comprobamos su funcionamiento.

¿Cómo modificaríamos el tiempo de funcionamiento a 2 segundos?

A3.-Medir la luminosidad con LDR

En esta actividad se encenderá el LED amarillo (D5) cuando oscurezca, como si de una sonda crepuscular se tratase. Para ello se utilizaremos la entrada analógica (A2) de la LDR (Resistor Dependiente de la Luz) y visualizaremos el valor del sensor en el escenario creando una variable a la que llamaremos luz.



Introduciremos el siguiente programa:



Con este programa hacemos que, si el valor de la luz es inferior a 450 encienda el LED amarillo (D5) durante dos segundos y si no,

lo apagará.

Comprobaremos el funcionamiento.

¿Cómo lo podríamos hacer para que se active el LED amarillo a partir de un nivel de luz más alto?

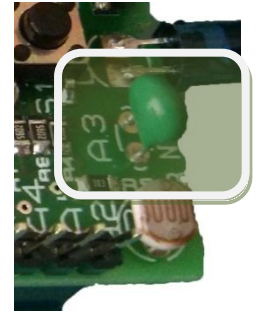
¿Sabrías hacer que, cambiase el color del vestido del panda cuando se enciende un LED?

A4.-Medir la temperatura con NTC

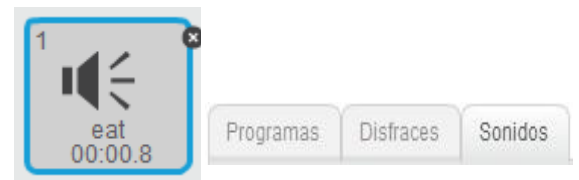
En esta práctica haremos un detector de incendios. Se tendrá que encender el LED rojo (D6) cuando se supere un determinado valor del sensor de temperatura conectado en (A3).

El sensor de temperatura que utilizamos es de tipo analógico y se llama NTC (Coeficiente de Temperatura Negativo). Este sensor se caracteriza por disminuir su resistencia a medida que la temperatura aumenta.

Para visualizar el valor del sensor en el escenario hay que crear una variable que se llame temperatura.



En esta ocasión haremos que el panda cambie de vestido cuando se supere el valor 40. Desde la pestaña de vestidos crearemos y pintaremos de color rojo el segundo vestido que se llama "Panda-b". También haremos que reproduzca un sonido "eat" o cualquier otro.



En la pestaña de Programas del Panda introduciremos el siguiente programa:

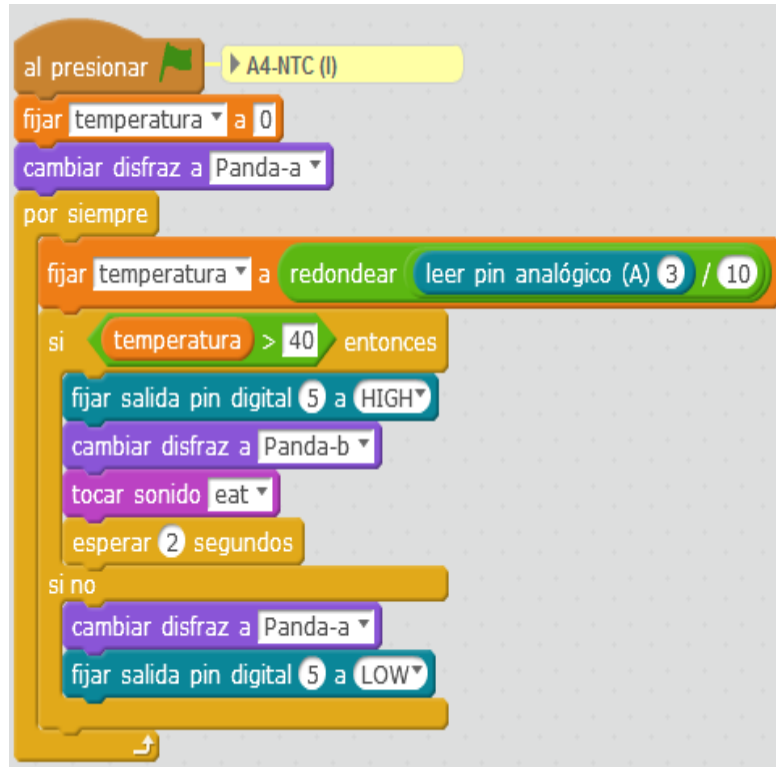
Crearemos una variable que vamos a llamar "temperatura".

Esta tomará el valor de A3 (sensor de temperatura dividido por 10 para escalar la medida y redondeamos para trabajar con valores enteros).

Cuando el valor de la temperatura sea superior 40, el panda cambiará de color y hará el sonido "eat".

Comprobaremos el funcionamiento.

¿Si queremos que el sistema reaccione a menor temperatura, que debemos cambiar?



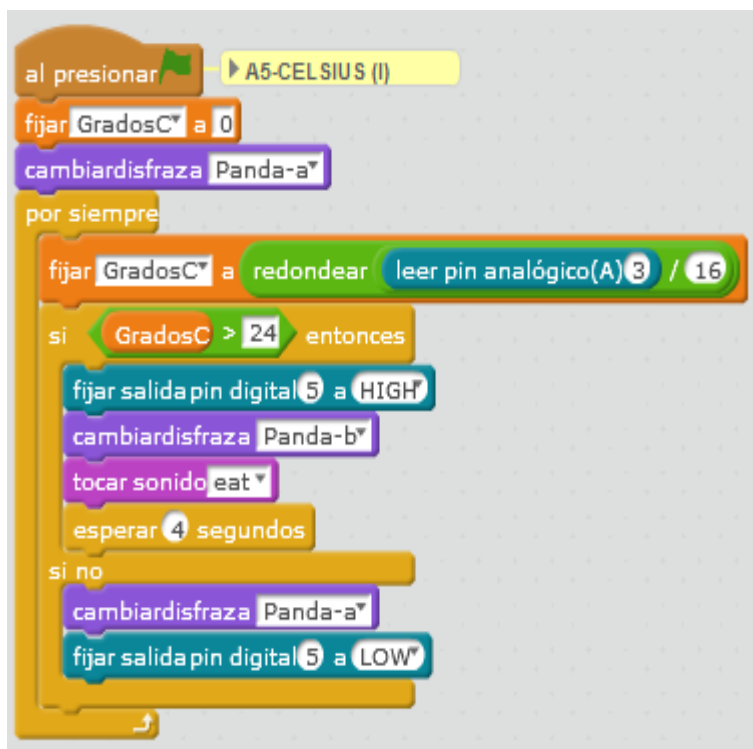
A5.-Medir la temperatura en grados Celsius.

Para hacer que la medida de temperatura se corresponda con los grados Celsius (°C) tendremos que crear una variable. Una variable es un espacio de la memoria donde se guardan valores entre 0 y 255.

Para hacerlo, deberemos ir a la categoría *Datos*, hacer clic en *Crear una Variable* y la llamaremos GradosC.



Introduciremos el programa siguiente:



***El hecho de dividir el valor del sensor A3 entre dieciséis es para convertir la medida de gados Celsius (°C).**

***La instrucción Redondear elimina los decimales de la conversión a grados.**

***Hay que tener en cuenta que este sensor no es lineal. Para lecturas más precisas habría que hacer una aproximación polinómica que no se tratará en este manual.**

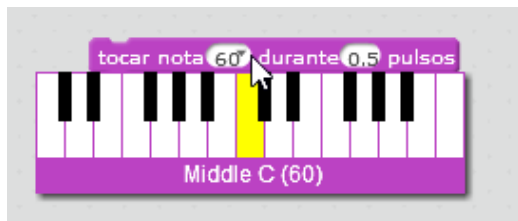
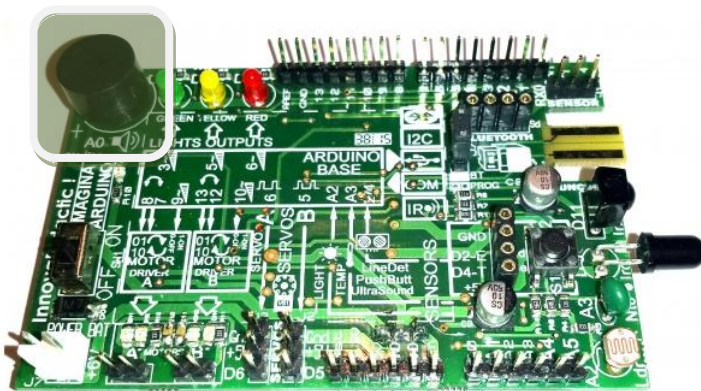
Comprobaremos su funcionamiento y añadiremos los comentarios a cada instrucción.

Actividades (modo2: cable USB desconectado)

Estas actividades se proponen con la placa desconectada del ordenador. El procedimiento está detallado en el anexo 3.

A6.-Generador de notas musicales con el zumbador

En esta actividad vamos a generar notas musicales con el zumbador que lleva la placa y que está conectado al pin A0, que también se llama D14 cuando se quiere utilizar como pin digital.



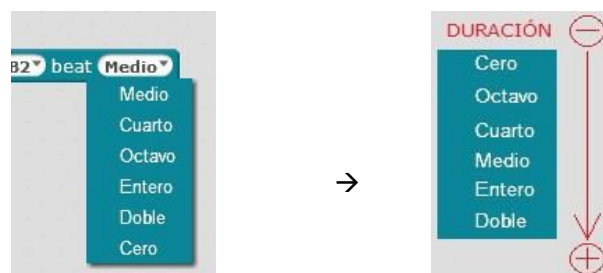
Para la selección de las notas os puede ser de utilidad la instrucción:

Así podemos relacionar el número de correspondencia con las notas musicales.

Para la programación vamos a usar la instrucción:



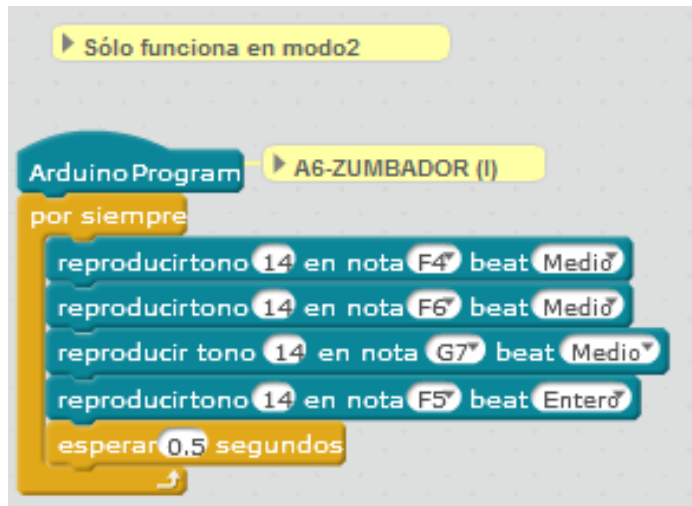
Pulsando sobre Medio, podremos seleccionar la duración del sonido tal como:



Pulsando sobre B2, podremos seleccionar el tono y la nota de nuestro sonido.

note	B2	beat
C2		
D2		
E2		
F2		
G2		
A2		
B2		
C3		
D3		
E3		
F3		
G3		
A3		
B3		
C4		
D4		
E4		
F4		
G4		
A4		
B4		
C5		
D5		

Experimentaremos con el siguiente programa:



***La opción de zumbador sólo funciona en modo2, es decir cambiando la bandera verde por el bloque *ArduinoProgram* y transmitiendo el programa a la placa Arduino.**

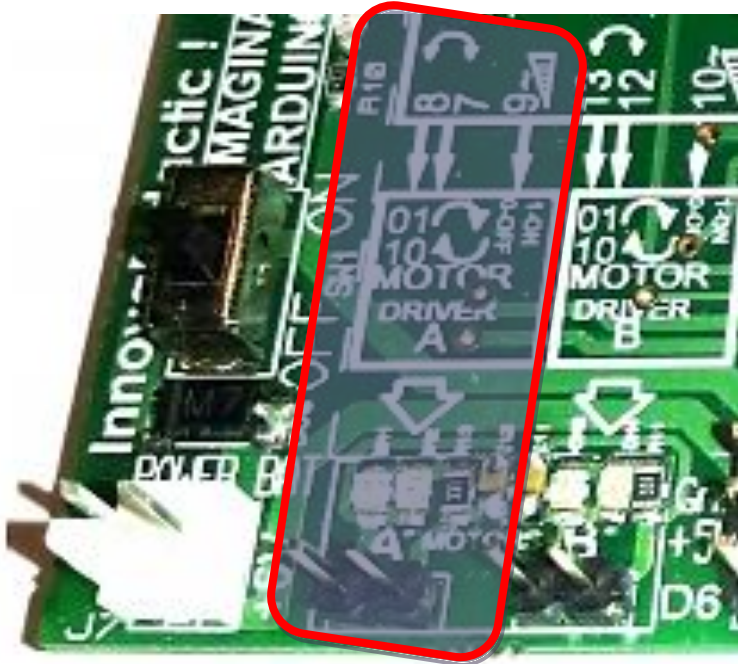
Para poder experimentar con otro tipo de sonidos, cambiaremos el valor de las notas y de las pulsaciones.

¿Qué pasa cuando subimos el valor de una nota? ¿El sonido es más agudo o más grave?

¿Y qué pasa cuando subimos el valor de la pulsación? ¿La duración del sonido es menor o mayor?

A7.-Controlar un motor CC.

Con la placa Shield Imagina Arduino podremos realizar el control de dos motores de corriente continua de hasta 2Amperios.



***Para realizar estas prácticas se requiere de tener las pilas conectadas, pues el puerto USB no da la suficiente corriente. Pueden coexistir las pilas con la alimentación USB.**

En esta práctica vamos a hacer funcionar el motor A, que es donde tenemos conectado el motor de la izquierda, en el sentido de la marcha.

Para hacer funcionar el motor A hacia adelante tenemos que activar la salida 7(HIGH), desactivar la 8(LOW) y activar el pin PWM 9 con un valor entre 25 y 255.

Con la modulación por ancho de pulso PWM de la salida 9, activamos el canal A del driver de motores L293DD y lo hacemos regulable en velocidad.

Los pines 7 y 8 sirven para seleccionar el sentido de giro.

Este procedimiento lo podemos hacer con los bloques tal como se muestra en la imagen adjunta.

A continuación se detalla en forma de tabla los posibles movimientos del motor A:

<p>CONTROL</p>	 <p>Adelante</p> <pre>fijar salida pin digital 7 a HIGH fijar salida pin digital 8 a LOW pizar pin PWM 9 a 100</pre>	 <p>Parado</p> <pre>fijar salida pin digital 7 a HIGH fijar salida pin digital 8 a LOW pizar pin PWM 9 a 0</pre>	 <p>Atrás</p> <pre>fijar salida pin digital 7 a LOW fijar salida pin digital 8 a HIGH pizar pin PWM 9 a 100</pre>
<p>MOTOR_A</p>	<p><i>Pin 7 (on)</i></p> <p><i>Pin 8 (off)</i></p> <p><i>Pin PWM 9 (25-255)</i></p>	<p><i>Pin 7 (off)</i></p> <p><i>Pin 8 (off)</i></p> <p><i>Pin PWM 9 (0)</i></p>	<p><i>Pin 7 (off)</i></p> <p><i>Pin 8 (on)</i></p> <p><i>Pin PWM 9 (25-255)</i></p>

Introduciremos este programa para que el motor A vaya hacia delante durante 1 segundo, se pare y repita el proceso.

<pre>al presionar [A7-MOTOR_A INTERMITENTE] por siempre fijar salida pin digital 7 a HIGH fijar salida pin digital 8 a LOW pizar pin PWM 9 a 100 esperar 1 segundos pizar pin PWM 9 a 0 esperar 1 segundos</pre>	<p>MOTOR_A</p>   <pre>pizar pin PWM 9 a 0</pre> <pre>pizar pin PWM 9 a 100</pre>
--	---

***Si el motor gira hacia atrás, tendremos que invertir la posición de los cables del motor A.**

Si queremos que gire hacia atrás durante un segundo, se pare y repita el proceso se haría así:

al presionar ▶ A7-MOTOR_A INTERMITENTE ATRÁS

por siempre

fijar salida pin digital 7 a LOW

fijar salida pin digital 8 a HIGH

pijar pin PWM 9 a 100

esperar 1 segundos

pijar pin PWM 9 a 0

esperar 1 segundos

MOTOR_A




pijar pin PWM 9 a 0

pijar pin PWM 9 a 100

Y si queremos que el motor A funcione una vez hacia delante y otra hacia atrás durante un segundo, pasando por paro (stop). La secuencia sería la siguiente:

 Adelante	 Parado	 Atrás	 Parado
fijar salida pin digital 7 a HIGH fijar salida pin digital 8 a LOW pijar pin PWM 9 a 100	fijar salida pin digital 7 a HIGH fijar salida pin digital 8 a LOW pijar pin PWM 9 a 0	fijar salida pin digital 7 a LOW fijar salida pin digital 8 a HIGH pijar pin PWM 9 a 100	fijar salida pin digital 7 a LOW fijar salida pin digital 8 a HIGH pijar pin PWM 9 a 0

Intentad hacer vosotros el programa y comprobad el funcionamiento.

***El motor_A se puede parar de dos formas: una poniendo PWM9 a cero y/o poniendo las salidas 7 y 8 a LOW.**

```

al presionar ▶ A7-MOTOR_A ADELANTE-PARO-ATRÁS-PARO
por siempre
    fijar salida pin digital 7 a HIGH ▶ ADELANTE
    fijar salida pin digital 8 a LOW
    pijar pin PWM 9 a 100
    esperar 1 segundos
    pijar pin PWM 9 a 0 ▶ PARO
    esperar 1 segundos
    fijar salida pin digital 7 a LOW ▶ ATRÁS
    fijar salida pin digital 8 a HIGH
    pijar pin PWM 9 a 100
    esperar 1 segundos
    pijar pin PWM 9 a 0 ▶ PARO
    esperar 1 segundos
    
```


Haremos lo mismo para el motor B. El motor B lleva asociadas las salidas del pin PWM 10, el pin 12 y el pin 13 tal como se puede ver en la tabla resumen siguiente:

CONTROL MOTORES	 Adelante	 Parado	 Atrás e
MOTOR_A (motor_izquierda)	<i>Pin 7 (on)</i> <i>Pin 8 (off)</i> <i>Pin PWM 9 (25-255)</i>	<i>Pin 7 (off)</i> <i>Pin 8 (off)</i> <i>Pin PWM 9 (0)</i>	<i>Pin 7 (off)</i> <i>Pin 8 (on)</i> <i>Pin PWM 9 (25-255)</i>
MOTOR_B (motor_derecha)	<i>Pin 12 (on)</i> <i>Pin 13 (off)</i> <i>Pin PWM 10 (25-255)</i>	<i>Pin 12 (off)</i> <i>Pin 13 (off)</i> <i>Pin PWM 10 (0)</i>	<i>Pin 12 (off)</i> <i>Pin 13 (on)</i> <i>Pin PWM 10 (25-255)</i>

***Si el motor B gira al revés tenéis que invertir su conector.**

```

al presionar bandera
  ▶ A7-MOTOR_B ADELANTE-PARO-ATRÁS-PARO
  por siempre
    fijar salida pin digital 12 a HIGH ▶ ADELANTE
    fijar salida pin digital 13 a LOW
    pjar pin PWM 10 a 100
    esperar 1 segundos
    pjar pin PWM 10 a 0 ▶ PARO
    esperar 1 segundos
    fijar salida pin digital 12 a LOW ▶ ATRÁS
    fijar salida pin digital 13 a HIGH
    pjar pin PWM 10 a 100
    esperar 1 segundos
    pjar pin PWM 10 a 0 ▶ PARO
    esperar 1 segundos
  
```

*Todas estas actividades se pueden transferir sustituyendo la instrucción de la bandera verde por:



Así el robot estará desvinculado del ordenador.

A8.-Control de dos motores CC.

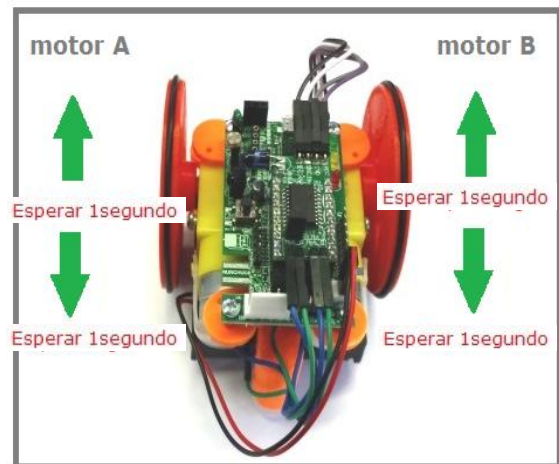
```

al presionar
por siempre
  fijar salida pin digital 7 a HIGH ▶ MOTOR_A ADELANTE
  fijar salida pin digital 8 a LOW
  pijar pin PWM 9 a 255
  fijar salida pin digital 12 a HIGH ▶ MOTOR_B ADELANTE
  fijar salida pin digital 13 a LOW
  pijar pin PWM 10 a 255
  esperar 1 segundos
  pijar pin PWM 9 a 0 ▶ PARO
  pijar pin PWM 10 a 0 ▶ PARO
  esperar 1 segundos
  fijar salida pin digital 7 a LOW ▶ MOTOR_A ATRÁS
  fijar salida pin digital 8 a HIGH
  pijar pin PWM 9 a 255
  fijar salida pin digital 12 a LOW ▶ MOTOR_B ATRÁS
  fijar salida pin digital 13 a HIGH
  pijar pin PWM 10 a 255
  esperar 1 segundos
  pijar pin PWM 9 a 0 ▶ PARO
  pijar pin PWM 10 a 0 ▶ PARO
  esperar 1 segundos
  
```

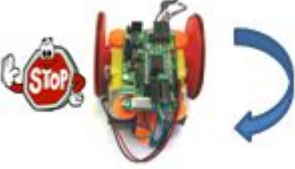
***Se requiere de la conexión de las pilas. En modo1 es normal que se los motores no se pongan en marcha simultáneamente.**

En esta actividad vamos hacer que el robot vaya hacia adelante y atrás durante un tiempo. Para hacerlo tendremos que hacer funcionar el motor A y el motor B a la vez.

Introduciremos el programa siguiente y comprobaremos su funcionamiento:



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

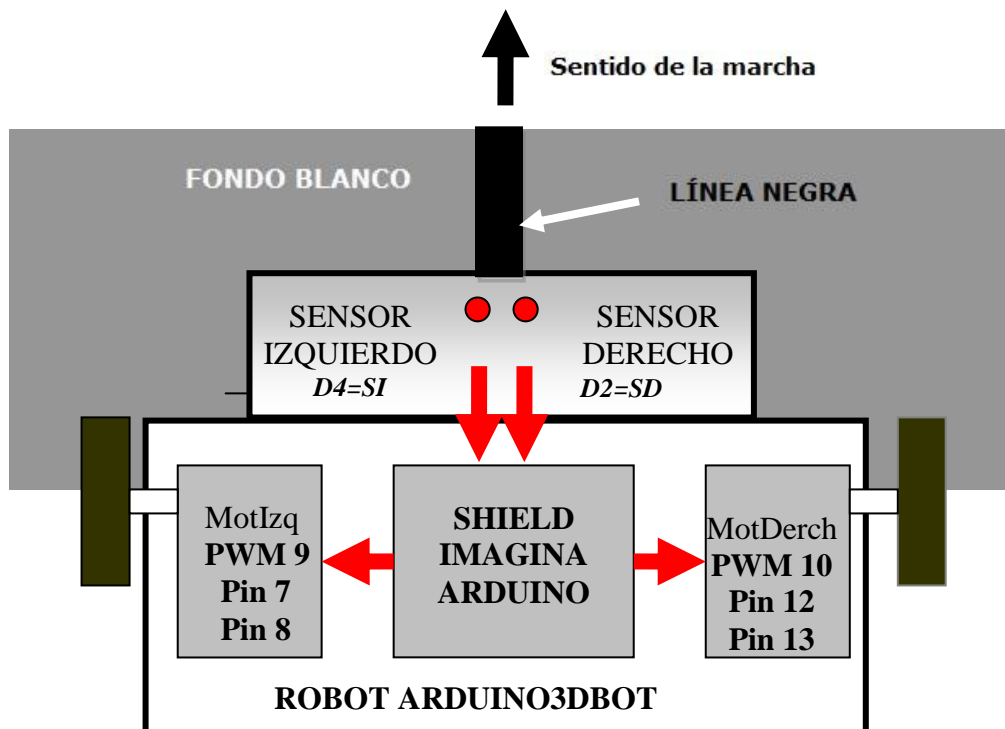
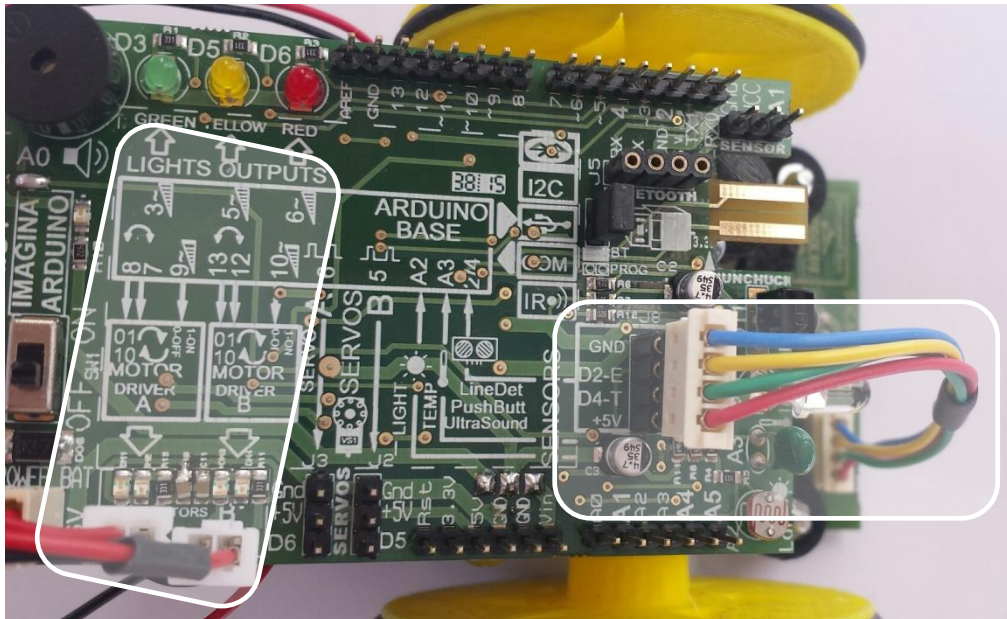
Pivotar a la izquierda :motor_A=inv /motor_B=on	Pivotar a la derecha: motor_A=on /motor_B=inv
	
Girar a la izquierda adelante: motor_A=off/motor_B=on	Girar a la derecha adelante: motor_A=on/motor_B=off
	
Girar a la izquierda atrás: motor_A=inv/motor_B=off	Girar a la derecha atrás: motor_A=off/motor_B=inv
	

A9.-Robot rastreador.

Cuando conectamos el accesorio ref. RBL0672-rast, podemos hacer que el robot se comporte como un rastreador de líneas. Esta placa va conectada con cables al Shield Imagina Arduino (consultar anexo1).



La disposición de los sensores y de los motores es la siguiente:



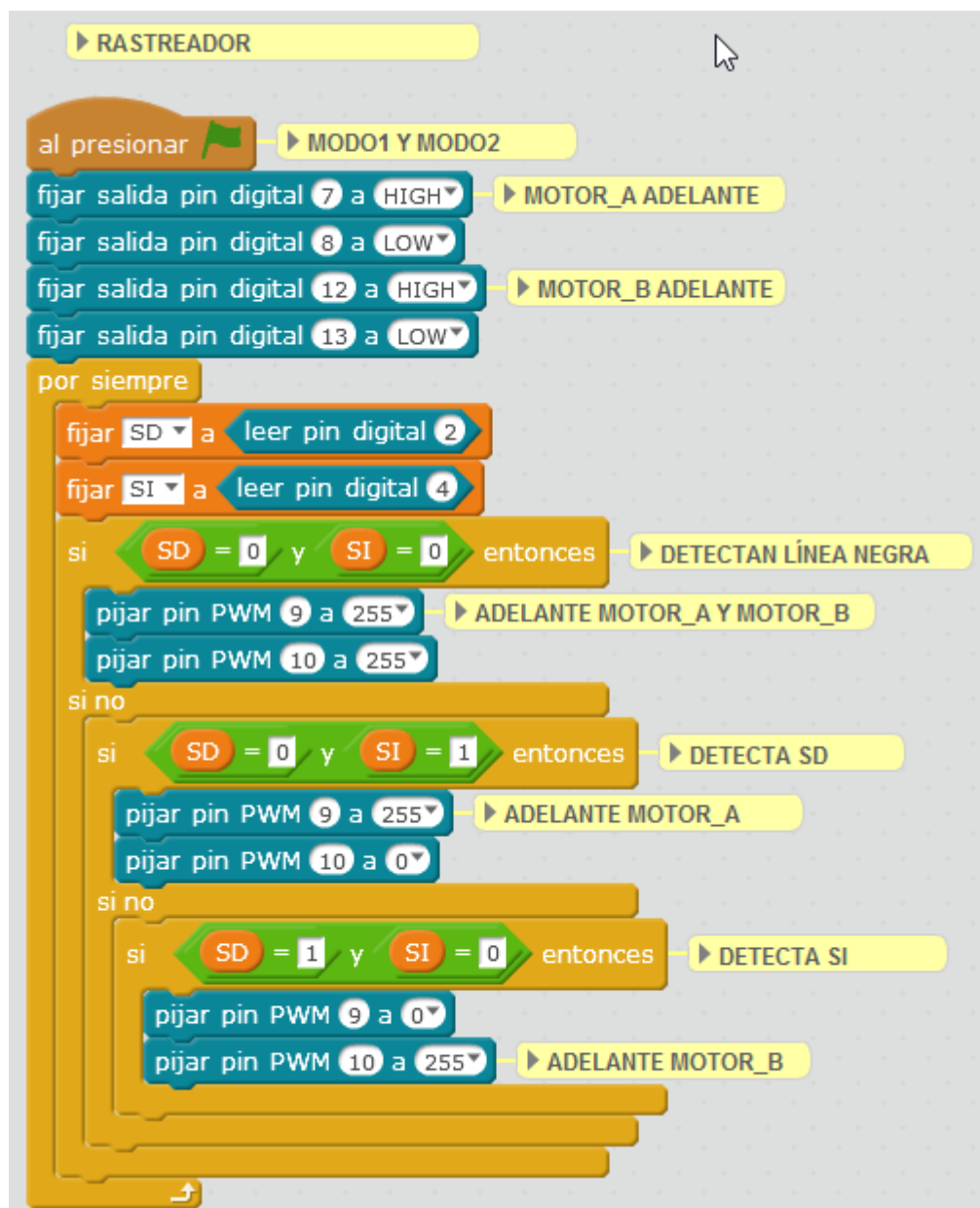
Las actuaciones del robot dependen de las detecciones de sus sensores.

1-Si los dos sensores del robot están encima de la línea negra, detectan un 0 lógico y el robot tiene que ir hacia delante, es decir tienen que funcionar los dos (motor_A=on) y (motor_B=on).

2-Si no detecta línea el sensor derecho (SD), paramos el de la izquierda (motor_A=off) y ponemos en marcha el de la derecha (motor_B=on).

3- Si no detecta línea el sensor izquierdo (SI), paramos el de la derecha (motor_B=off) y ponemos en marcha el de la izquierda (motor_A=on).

El programa sería este:



```

    RASTREADOR
    al presionar
        MODO1 Y MODO2
        fijar salida pin digital 7 a HIGH MOTOR_A ADELANTE
        fijar salida pin digital 8 a LOW
        fijar salida pin digital 12 a HIGH MOTOR_B ADELANTE
        fijar salida pin digital 13 a LOW
        por siempre
            fijar SD a leer pin digital 2
            fijar SI a leer pin digital 4
            si SD = 0 y SI = 0 entonces DETECTAN LÍNEA NEGRA
                pijar pin PWM 9 a 255 ADELANTE MOTOR_A Y MOTOR_B
                pijar pin PWM 10 a 255
            si no
                si SD = 0 y SI = 1 entonces DETECTA SD
                    pijar pin PWM 9 a 255 ADELANTE MOTOR_A
                    pijar pin PWM 10 a 0
                si no
                    si SD = 1 y SI = 0 entonces DETECTA SI
                        pijar pin PWM 9 a 0
                        pijar pin PWM 10 a 255 ADELANTE MOTOR_B
    
```

Comprobaremos el funcionamiento de los sensores de línea SD y SI, y luego transferiremos el programa al robot para que trabaje en modo2 (autónomo).

A10.-Robot explorador con ultrasonidos.

En esta actividad equiparemos el robot con un sensor de ultrasonido HC-SR04, ubicado en el conector de los sensores de línea y haremos que funcione autónomamente explorando su entorno sin chocar.

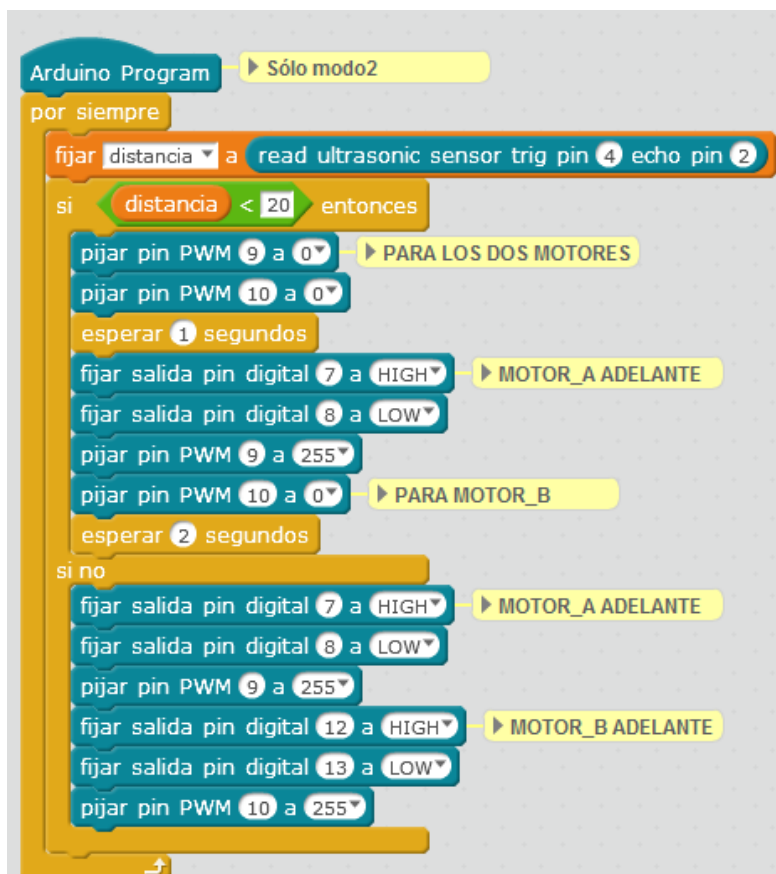
Para evitar el choque iremos realizando varias estrategias, de la más simple a la más complicada.

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.



```

Arduino Program - Sólo modo2
por siempre
  fijar distancia a read ultrasonic sensor trig pin 4 echo pin 2
  si distancia < 20 entonces
    pijar pin PWM 9 a 0 - PARA LOS DOS MOTORES
    pijar pin PWM 10 a 0
    esperar 1 segundos
  si no
    fijar salida pin digital 7 a HIGH - MOTOR_A ADELANTE
    fijar salida pin digital 8 a LOW
    pijar pin PWM 9 a 255
    fijar salida pin digital 12 a HIGH - MOTOR_B ADELANTE
    fijar salida pin digital 13 a LOW
    pijar pin PWM 10 a 255
  
```



```

Arduino Program - Sólo modo2
por siempre
  fijar distancia a read ultrasonic sensor trig pin 4 echo pin 2
  si distancia < 20 entonces
    pijar pin PWM 9 a 0 - PARA LOS DOS MOTORES
    pijar pin PWM 10 a 0
    esperar 1 segundos
    fijar salida pin digital 7 a HIGH - MOTOR_A ADELANTE
    fijar salida pin digital 8 a LOW
    pijar pin PWM 9 a 255
    pijar pin PWM 10 a 0 - PARA MOTOR_B
    esperar 2 segundos
  si no
    fijar salida pin digital 7 a HIGH - MOTOR_A ADELANTE
    fijar salida pin digital 8 a LOW
    pijar pin PWM 9 a 255
    fijar salida pin digital 12 a HIGH - MOTOR_B ADELANTE
    fijar salida pin digital 13 a LOW
    pijar pin PWM 10 a 255
  
```

La segunda estrategia consistirá en hacer que cuando se detenga gire a la derecha durante un tiempo determinado y luego siga adelante. Podemos experimentar con diferentes temporizaciones. Ahora está en 2 segundos.

Como tercera estrategia haremos que después de detenerse gire hasta que la distancia sea más grande de 30cm.

```

Arduino Program  ▶ Sólo modo2
por siempre
  fijar distancia a read ultrasonic sensor trig pin 4 echo pin 2
  si distancia < 20 entonces
    pijar pin PWM 9 a 0 ▶ PARA LOS DOS MOTORES
    pijar pin PWM 10 a 0
    esperar 1 segundos
    si distancia < 30 entonces
      fijar salida pin digital 7 a HIGH ▶ MOTOR_A ADELANTE
      fijar salida pin digital 8 a LOW
      pijar pin PWM 9 a 255
      pijar pin PWM 10 a 0 ▶ PARA MOTOR_B
      esperar 0.5 segundos
    si no
      fijar salida pin digital 7 a HIGH ▶ MOTOR_A ADELANTE
      fijar salida pin digital 8 a LOW
      pijar pin PWM 9 a 255
      fijar salida pin digital 12 a HIGH ▶ MOTOR_B ADELANTE
      fijar salida pin digital 13 a LOW
      pijar pin PWM 10 a 255
  
```

A11.-Mide distancias con ultrasonidos (HC-SR04).

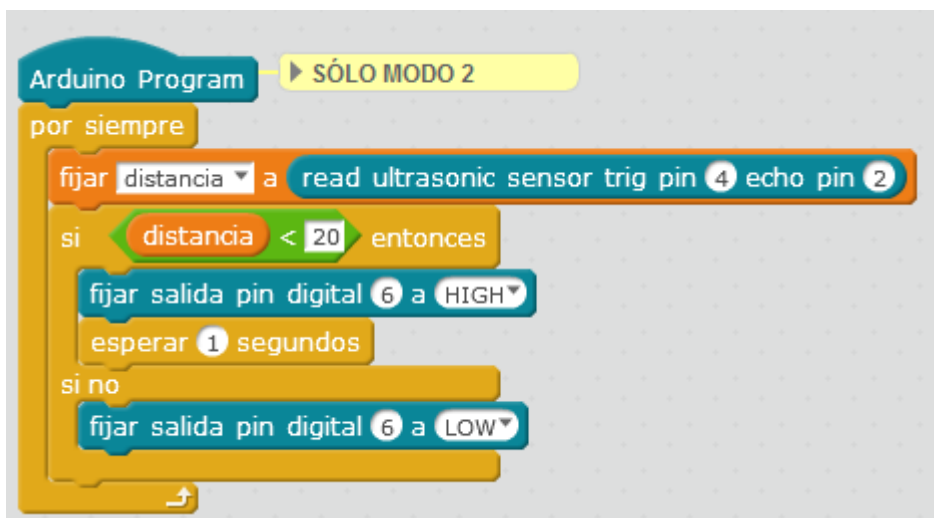
Si disponemos de un sensor de ultrasonido HC-SR04 podremos realizar medidas de distancia directamente en centímetros.

El funcionamiento de los sensores de ultrasonidos consiste en la emisión de un sonido no audible y medir el tiempo que tarda en rebotar. Así se puede calcular la distancia a la que se encuentra el objeto que está delante. El rango de medida es de 0 a 255cm.

Para realizar esta práctica conectaremos el sensor HC-SR04 en el lugar donde se conectan los sensores de línea. Hay que prestar atención al realizar la conexión, pues si se conecta al revés el sensor se estropea.

Lamentablemente en mBlock este recurso sólo funciona en modo2 (cable USB desconectado).

En primer lugar haremos un programa que cuando detecte un objeto a menos de 20cm encienda el LED rojo (D6).



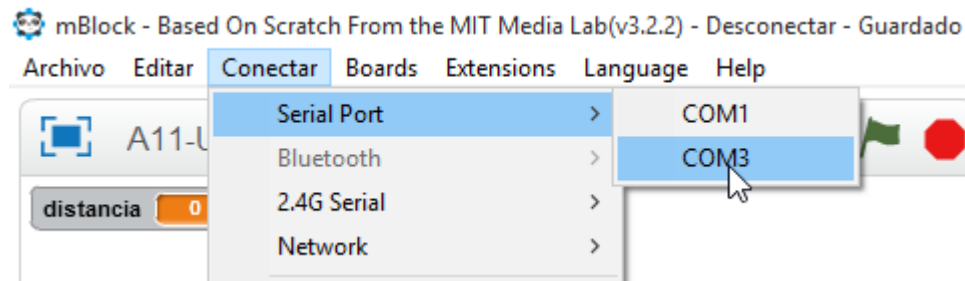
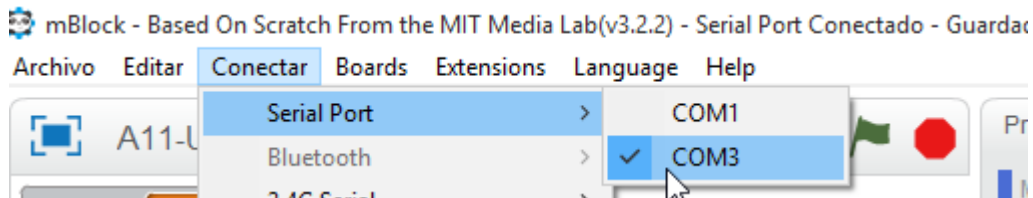
Lo cargaremos al Arduino Uno y comprobaremos su funcionamiento. Podéis probar con otras distancias.

En el ejemplo anterior nos damos cuenta que sería interesante poder visualizar la medida del sensor de alguna manera. Para ello proponemos un programa que la envíe por el puerto

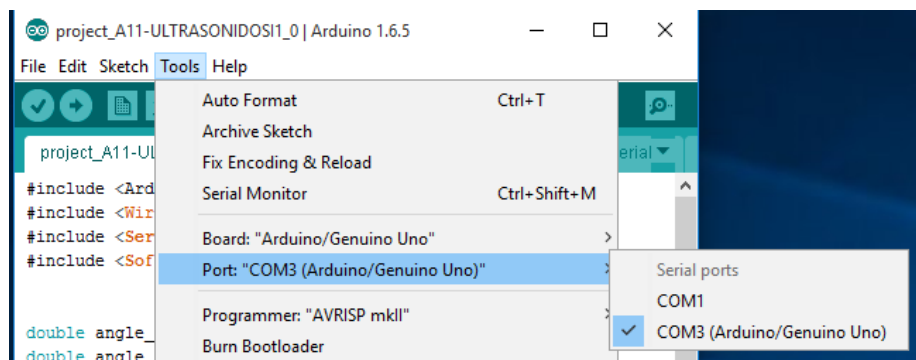
serie para poder visualizarla en cualquier terminal de comunicaciones. El programa sería el siguiente:

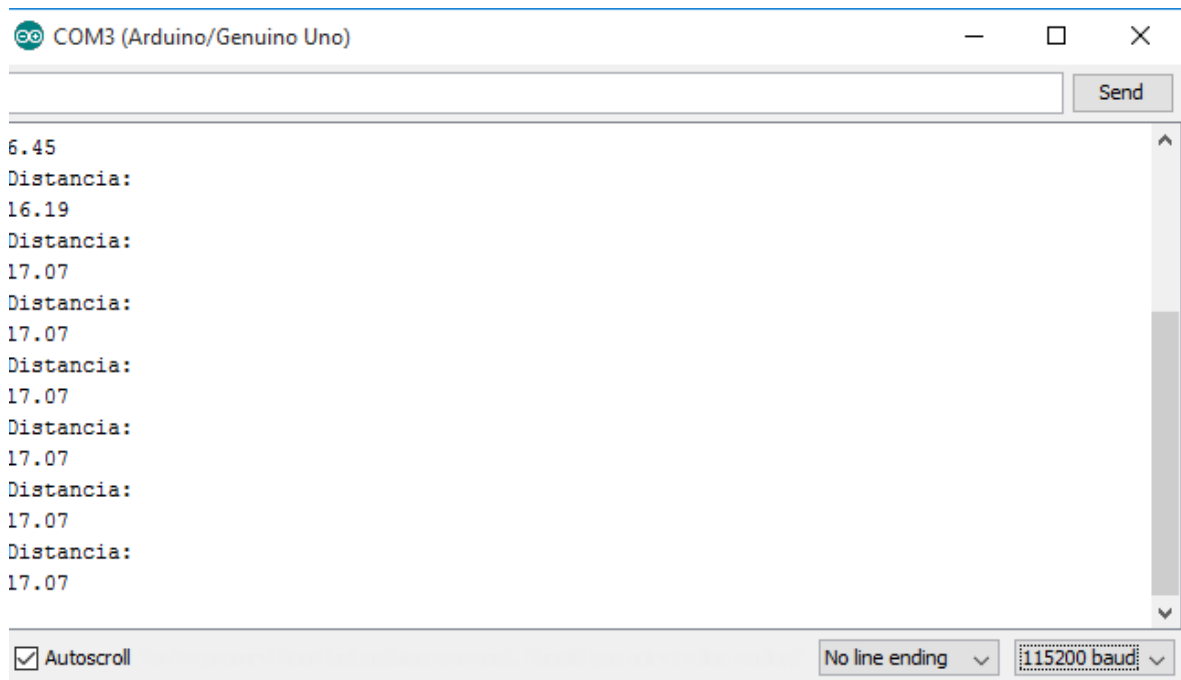


Lo transferimos al Arduino, desconectamos el puerto COM asignado en mBlock (desmarcar la tilde haciendo clic encima con el ratón).



Luego abrimos el entorno de programación IDE que dispone de un terminal. Seleccionamos la placa Arduino Uno, el puerto COM asignado y pulsamos CTRL+SHIFT+M para abrirlo.





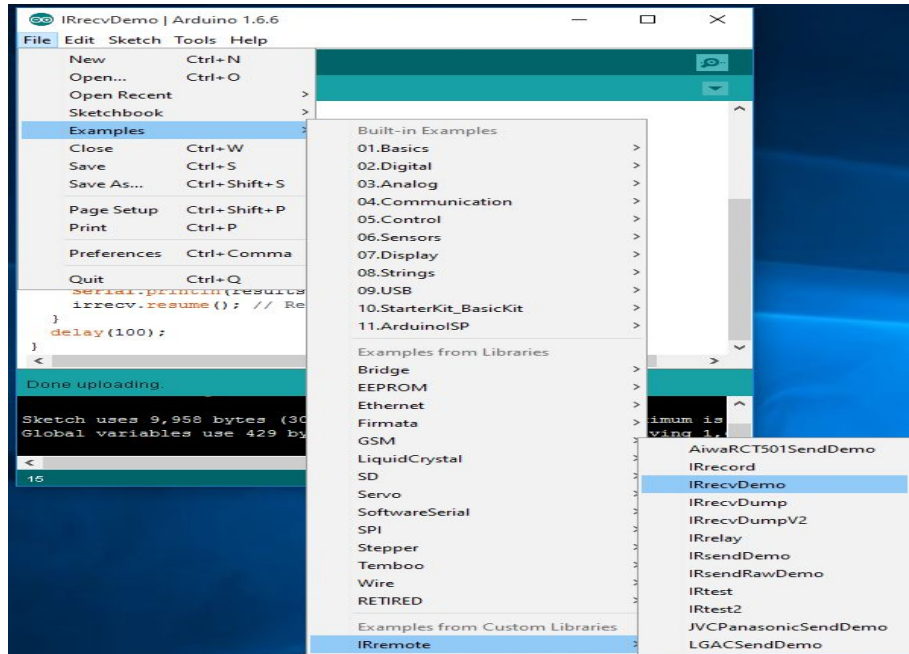
En esta ventana del terminal de comunicaciones se visualizan las medidas del ultrasonido.

***Es importante seleccionar correctamente la velocidad de comunicación a 115200baudios para que se visualicen correctamente los valores.**

A12.-Control por infrarrojos

En esta actividad aprenderemos a utilizar el mando de la televisión [bxl-rc001](#), configurado con el estándar Sony, para dar órdenes a nuestro robot. Lamentablemente todavía no es posible con mBlock. En esta ocasión lo vamos a realizar con [IDE](#).

Empezaremos por encender un LED y apagarlo con el mando. Cargaremos el programa IRrecvDemo desde `File /Examples/IRremote`



```

A12-INFRARROJOS_I | Arduino 1.6.6
File Edit Sketch Tools Help
[Icons]
A12-INFRARROJOS_I
/*
 * IRremote: IRrecvDemo - demonstrates receiving IR codes
 * Encendido del LED amarillo (D5) con la tecla número
 */

#include <IRremote.h>

int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
  pinMode(5, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value);
    irrecv.resume(); // Receive the next value
  }
  delay(100);
  if(((results.value)==(16))){
    digitalWrite(5,1); //Motor_A adelante
  }else{
    digitalWrite(5,0); //Apaga LED amarillo
  }
}
    
```

Luego añadiremos el código necesario para que quede tal como el siguiente:

Si abrimos el terminal de comunicaciones del IDE (CTRL+SHIFT+M) veremos que al pulsar la tecla 1 del mando (configurado como Sony) envía un 16. Con este valor encenderemos el LED amarillo D5 y con cualquier otra lo apagaremos.

Se reciben los códigos por la entrada D11 y se espera recibir el código 16.

Descarga del programa: [A12-INFRARROJOS\(I\)](#)

Una vez hemos aprendido a encender y apagar un LED vamos a realizar un programa para controlar el robot con el mando de la TV. Para ello vamos a utilizar las siguientes teclas, cuyos valores hemos obtenido con el programa anterior:

				
2832	144	2192	1680	3728



El programa para el control del robot con el mando de la TV seria tal como:

```

A12-INFRRARROJOS_IL_ | Arduino 1.6.6
File Edit Sketch Tools Help
A12-INFRRARROJOS_IL_
/*
 * IRremote: IRrecvDemo - demonstrates receiving IR codes with IRr
 * Control robot Arduino3dbot con mando TV.
 *
 */

#include <IRremote.h>

int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
}
    
```

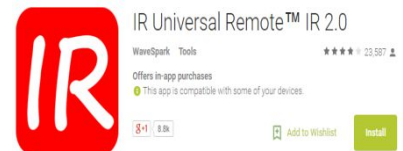
```

void loop() {
  if (irrecv.decode(&results) ) {
    //Serial.println(results.value); //Desmarcar para ver los codigos enviados
    irrecv.resume(); // Receive the next value
  }
  delay(100);
  if(((results.value)==(2832))){ //Adelante
    digitalWrite(7,1); //Motor_A adelante
    digitalWrite(8,0); //
    digitalWrite(9,1); //
    digitalWrite(12,1); //Motor_B adelante
    digitalWrite(13,0);
    digitalWrite(10,1);
  }
  if(((results.value)==(3728))){ //Paro
    digitalWrite(9,0); //Motor_A parado
    digitalWrite(10,0); //Motor_B parado
  }
  if(((results.value)==(1680))){ //Atrás
    digitalWrite(7,0); //Motor_A atrás
    digitalWrite(8,1); //
    digitalWrite(9,1); //
    digitalWrite(12,0); //Motor_B atrás
    digitalWrite(13,1);
    digitalWrite(10,1);
  }
  if(((results.value)==(144))){ //izquierda
    digitalWrite(9,0); //Motor_A parado
    digitalWrite(12,1); //Motor_B adelante
    digitalWrite(13,0);
    digitalWrite(10,1);
  }
  if(((results.value)==(2192))){ //derecha
    digitalWrite(10,0); //Motor_B parado
    digitalWrite(7,1); //Motor_A adelante
    digitalWrite(8,0);
    digitalWrite(9,1);
  }
}

```

***En el caso de que se quiera hacer funcionar otro robot con otro mando, para que no se interfieran deberemos usar otras teclas.**

Si disponemos de un móvil Android, con emisor de infrarrojos, podemos descargarnos [IR REMOTE 2.0](#) para controlar el robot. En *IR Databases* se debe seleccionar Sony y luego teclado numérico.



*Descarga programa: [A12-INFRRARROJOS\(II\)](#)

Comprobaremos su funcionamiento.

***Es imprescindible configurar el mando de televisión [bxi-rc001](#) con el estándar Sony:**

- pulsar SET y TV1 simult.
- (se activa el LED rojo)
- pulsar:0126
- pulsar:TV1

A13.-Control por Bluetooth

En esta actividad aprenderemos a controlar nuestro robot con un móvil Android a través de Bluetooth.

Colocaremos el módulo Bluetooth [JY-MCU-HC06](#), que se suministra configurado a 9600 baudios, respetando los pines de la serigrafía de la placa.



A nuestro robot le cargaremos el siguiente programa:

 A13-BLUETOOTH | Arduino 1.6.6

File Edit Sketch Tools Help

✓
↶
📄
⬆️
⬇️

A13-BLUETOOTH
MeSerial.cpp
MeSerial.h

```

#include <Arduino.h>
#include <Wire.h>
#include <Servo.h>
#include <SoftwareSerial.h>

#include "MeSerial.h"

double command;
MeSerial se;

void setup() {
  Serial.begin(9600);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  command = 0;
  Serial.println("Arduino3dbot");
  digitalWrite(9,1);
  digitalWrite(10,1);
}

```

```

void loop() {

  if(Serial.available() > (0)){
    command = Serial.read();
    if(((command)=='U')){ //Adelante
      digitalWrite(7,1);
      digitalWrite(8,0);
      digitalWrite(12,1);
      digitalWrite(13,0);
    }
    if(((command)=='7')){ //Paro
      digitalWrite(7,0);
      digitalWrite(8,0);
      digitalWrite(12,0);
      digitalWrite(13,0);
    }
    if(((command)=='8')){ //Atrás
      digitalWrite(7,0);
      digitalWrite(8,1);
      digitalWrite(12,0);
      digitalWrite(13,1);
    }
    if(((command)=='6')){ //Izquierda
      digitalWrite(7,0);
      digitalWrite(8,0);
      digitalWrite(12,1);
      digitalWrite(13,0);
    }
    if(((command)=='2')){ //Derecha
      digitalWrite(7,1);
      digitalWrite(8,0);
      digitalWrite(12,0);
      digitalWrite(13,0);
    }
  }
}

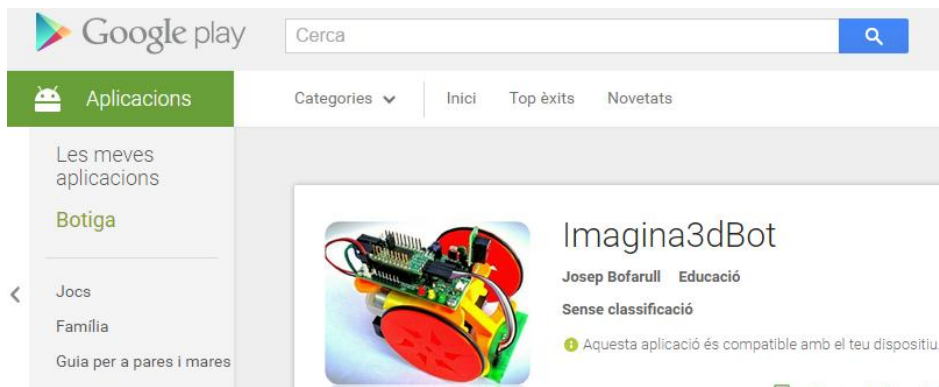
```

*Descarga programa: [A13-BLUETOOTH\(I\)](#)

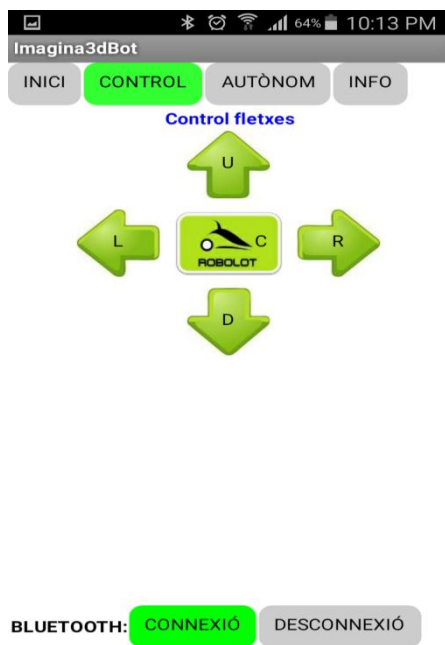
***Para transferir el programa hay que quitar el módulo Bluetooth o el jumper, ya que utiliza las mismas señales que el puerto COM (USB).**

***En el momento de realizar este manual las instrucciones de mBlock no funcionan correctamente para la lectura de datos vía serie.**

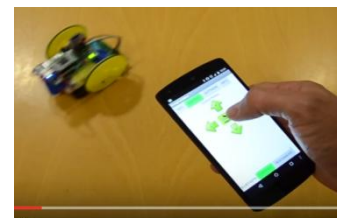
Y en el móvil instalaremos una App llamada [Imagina3dBot](#) que nos descargaremos de GooglePlay.



El robot se puede controlar con botones con forma de flechas o con el giroscopio del propio móvil Android. Para hacer la conexión solo se debe elegir CONNEXIÓ/Dirección Mac del módulo bluetooth del robot/ botón CONTROL y "Control fletxes (flechas)" o "Control orientació (orientación)":

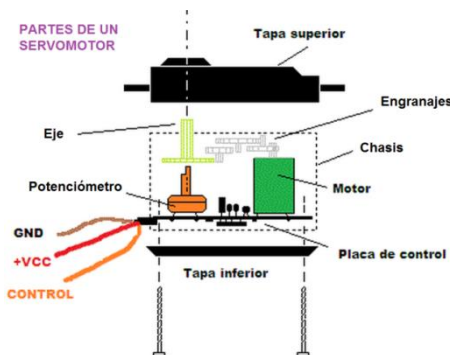


Video del control con botones tipo flechas:
<https://www.youtube.com/watch?v=SytjWNLB7Q>



Video del control con giroscopio:
https://www.youtube.com/watch?v=WJ_96BzCuQ

A14.-Control servomotor



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.

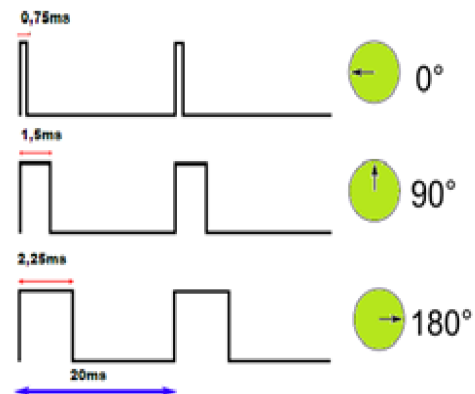


Los servomotores estándar sólo pueden girar 180°, aunque en el mercado podemos encontrar de 270° y de 360° (giro continuo).

La señal de control que se envía al servomotor son impulsos comprendidos entre 0,75ms y 2,25ms de duración que se repiten cada 20ms.

Con mBlock simplemente hay que indicar directamente los grados del ángulo que debe posicionarse el eje del servomotor.

IMPULSOS DE CONTROL POSICION EJE SERVOMOTOR



```

al presionar MOD01 Y MOD02
por siempre
  fijar ángulo del pin 6 del servo a 45°
  esperar 1 segundos
  fijar ángulo del pin 6 del servo a 135°
  esperar 1 segundos
  
```

Si conectamos un servomotor en D6 y introducimos el programa siguiente, podremos ver cómo va alternando la posición de su eje entre 45° y 135°.

Para mejorar el programa anterior os proponemos hacer que la posición del servomotor cambie según el valor de la luz que recibe el sensor LDR (Resistor Dependiente de la Luz) conectado en la entrada analógica A2.

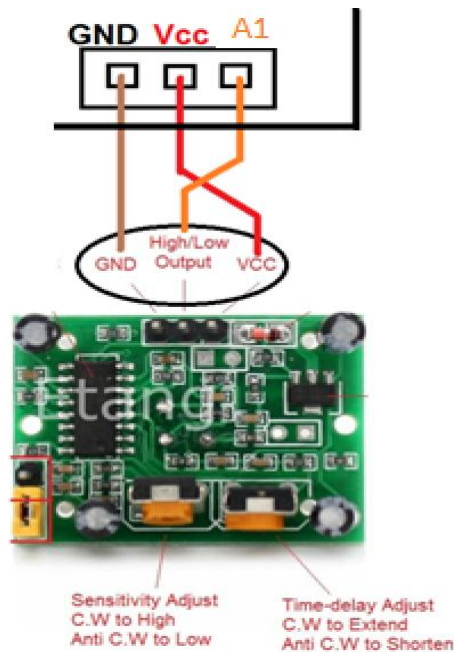
```

al presionar MOD01 Y MOD02
por siempre
  fijar luz a leer pin analógico (A) 2 / 6
  fijar ángulo del pin 6 del servo a luz
  esperar 0.1 segundos
  
```

Dividiremos por 6 el valor de la variable luz para escalarla, ya que las entradas analógicas varían entre 0-1023 y el servo admite 0-180.

A15.-Alarma intrusos (sensor PIR)

En esta actividad crearemos una alarma contra intrusos utilizando un sensor de movimiento PIR ref. [HC-SR501](#). Este sensor tiene un alcance máximo de unos 6m según el fabricante y una apertura de detección de 110°.



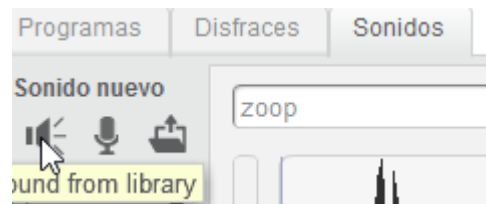
***Un sensor PIR detecta la radiación infrarroja que desprenden los seres vivos o cuerpos calientes.**

***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 i 60 segundos y es muy recomendable la ausencia de personas durante la calibración.**

***Se debe prestar atención al realizar la conexión del sensor, pues tiene dos pines intercambiados. Consultar esquema de la izquierda.**

Cuando el sensor detecta un intruso debe activarse un sonido durante 4 segundos y mostrar el objeto Panda en el escenario. Por defecto si no se activa la sirena, el panda debe estar oculto.

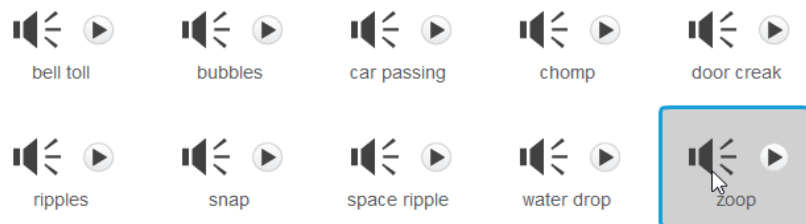
Desde la pestaña Sonidos hacemos clic en Efectos y seleccionamos zoop.



Biblioteca de sonidos

Categoría

- Todos
- Animal
- Efectos
- Electrónica
- Humanos
- Instrumentos
- Bucles de música
- Percusión
- Canto



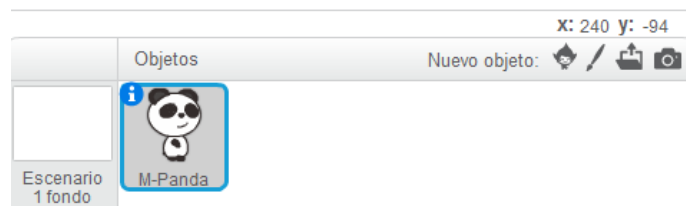
Desde la pestaña Programas introduciremos el programa siguiente:

```

al presionar bandera
  esconder
  por siempre
    fijar detector a leer pin digital 15
    si detector = 1 entonces
      A1=D15 SENSOR PIR
      mostrar
      tocar sonido zoop
      esperar 4 segundos
    si no
      esconder
  
```

***Seguramente tendremos que bajar el volumen de vuestro ordenador.**

El resultado será:



A16.-Sonda temperatura DS18B20 INOX

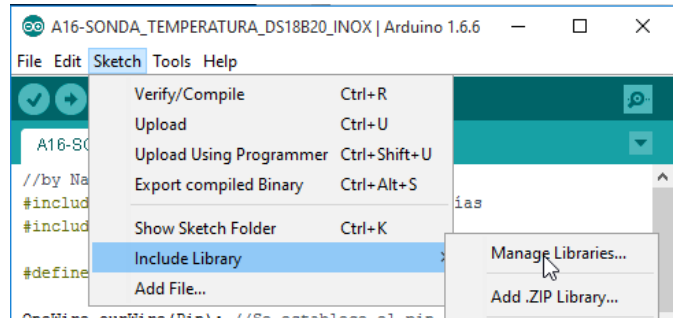


En esta actividad vamos a aprender a leer la temperatura con precisión usando una sonda DS18B20-inox que funciona con el protocolo OneWire. Los datos de la sonda los leeremos por el pin A2 (D15) y los enviaremos por el puerto serie.

Requiere instalar dos librerías: la Onewire y la DallasTemperature.

***Entre el Vcc y la salida de la sonda debe haber una resistencia de 4k7 ohmios.**

Luego introduciremos el programa siguiente:



```

A16-SONDA_TEMPERATURA_DS18B20_INOX | Arduino 1.6.6
File Edit Sketch Tools Help
A16-SONDA_TEMPERATURA_DS18B20_INOX
//LECTURA DE TEMPERATURA CON SONDA DS18B20 INOX

#include <OneWire.h> //Se importan las librerías
#include <DallasTemperature.h>

#define Pin 15 //Pin de entrada de datos del sensor A1=D15

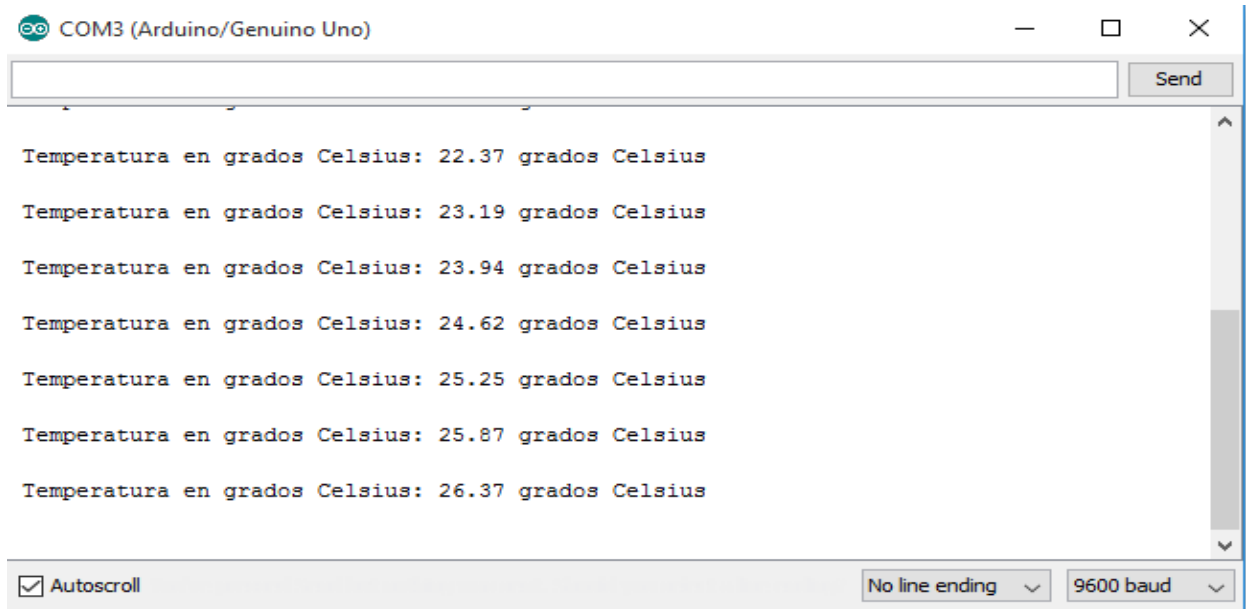
OneWire ourWire(Pin); //Se establece el pin declarado como bus para la comunicación One Wire
DallasTemperature sensors(&ourWire); //Se llama a la librería Dallas Temperature

void setup() { //Configuraciones
  Serial.begin(9600); //Configuración de la velocidad del puerto serie
  sensors.begin(); //Inicialización de los sensores
}

void loop() {
  sensors.requestTemperatures(); //Prepara el sensor para la lectura
  delay(500); //Tiempo entre lecturas
  Serial.print(" Temperatura en grados Celsius: "); // Sin cambio de línea
  Serial.print(sensors.getTempCByIndex(0)); //Lee e imprime la temperatura en grados Celsius
  Serial.println(" grados Celsius"); // Con cambio de línea
  Serial.println(); //Imprime línea en blanco
}
    
```

Descarga programa: [A16-DS18B20](#)

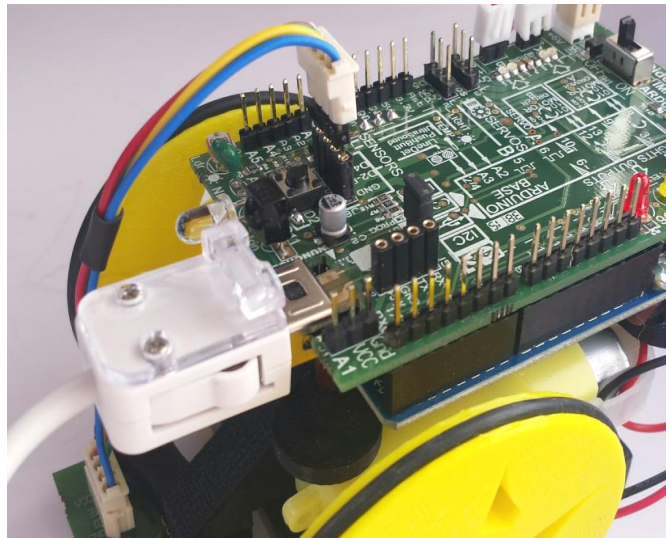
En el terminal del entorno de programación IDE (CTRL+SHIFT+M) se visualizarán los valores cada medio segundo tal como la imagen adjunta.



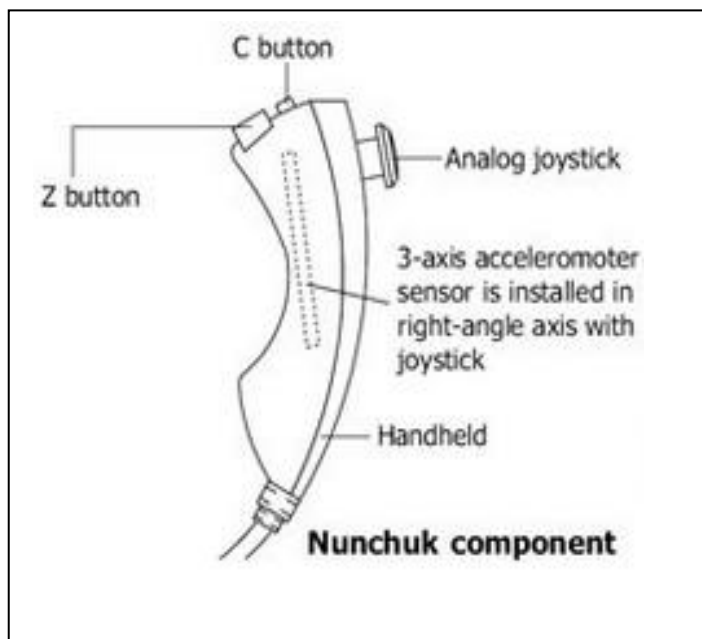
A17.- Nunchuk

En primer lugar hay que conectar la Nunchuk de la Wii a la placa tal como en la imagen:

***Atención: hay que vigilar la posición del conector, pues en caso de invertir la posición se puede estropear la Nunchuk.**



El mando Nunchuk de la Wii dispone de múltiples sensores integrados, tales como:



- 1 joystick de dos ejes, X y Y.
- 2 pulsadores Z i C.
- 1 acelerómetro de 3 ejes: X, Y y Z. *(Sólo algunos modelos).

Dado que el Shield Imagina Arduino permite la conexión de la Nunchuk realizaremos un programa para aprender a leer los datos de sus sensores y visualizarlos por pantalla.

```
A17_NUNCHUK_I_ | Arduino 1.0.5
File Edit Sketch Tools Help
[Icons: Checkmark, Arrow, File, Upload, Download]
A17_NUNCHUK_I_

/*
 *Envía los valores de la Nunchuk por el puerto serie
 *Para visualizarlos se requiere abrir el terminal de
 *comunicaciones del entorno IDE (CTRL+SHIFT+M)
 */

#include <Wire.h>
#include <ArduinoNunchuk.h>

#define BAUDRATE 9600

ArduinoNunchuk nunchuk = ArduinoNunchuk();

void setup()
{
  Serial.begin(BAUDRATE);
  nunchuk.init();
}







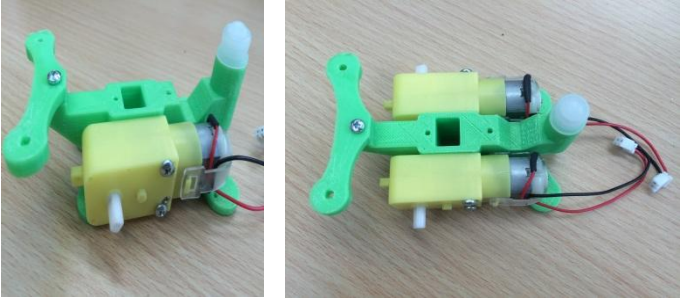

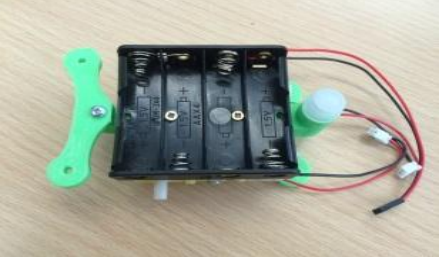
void loop()
{
  nunchuk.update();

  Serial.print(nunchuk.analogX, DEC);
  Serial.print(' ');
  Serial.print(nunchuk.analogY, DEC);
  Serial.print(' ');
  Serial.print(nunchuk.accelX, DEC);
  Serial.print(' ');
  Serial.print(nunchuk.accelY, DEC);
  Serial.print(' ');
  Serial.print(nunchuk.accelZ, DEC);
  Serial.print(' ');
  Serial.print(nunchuk.zButton, DEC);
  Serial.print(' ');
  Serial.println(nunchuk.cButton, DEC);
}
```

Descarga programa: [A17-NUNCHUK\(I\)](#)

Anexo 1.-Montaje del robot Imagina Arduino

A continuación detallamos el procedimiento de montaje del robot Imagina3dBot en 15 pasos.

<p>1.-Repasamos que tengamos todo el material especificado en la página 3.</p>	
<p>2.-Colocamos la rueda "loca" de la parte de detrás.</p> 	
<p>3.-Ponemos el soporte de delante y lo fijamos con el tornillo 2,9x9,5mm.</p> <p>*No hacer mucha fuerza.</p> 	
<p>4.-Fijamos el motor con los tornillos de 2,9x25mm.</p> <p>*No hacer mucha fuerza, ya que si comprimimos demasiado la caja reductora del motor este girará con dificultad.</p> 	
<p>5.-Fijamos el porta pilas con los dos tornillos de cabeza cónica dorados.</p> 	

6.-Roscamos los 4 tornillos de 2,9x9,5 mm, con los separadores.



7.-Fijamos la placa Arduino/Funduino Uno con los tornillos anteriores encima del chasis, con la ayuda de un destornillador estrella.

***No tenemos que hacer mucha fuerza.**

Luego ponemos el Shield Imagina Arduino de forma que coincidan las referencias de los pines.

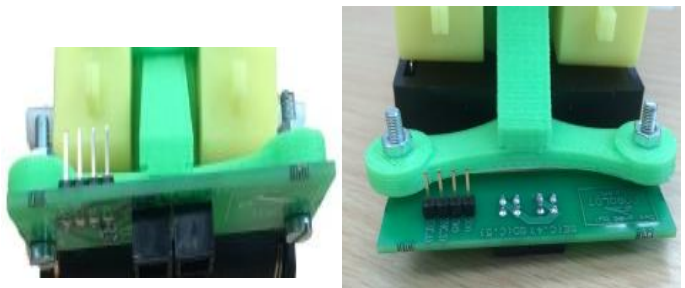
***Tomar como referencia RX (D0) de las dos placas.**



8.-Ponemos los tornillos M3x16mm con sus respectivas tuercas, como en la imagen.



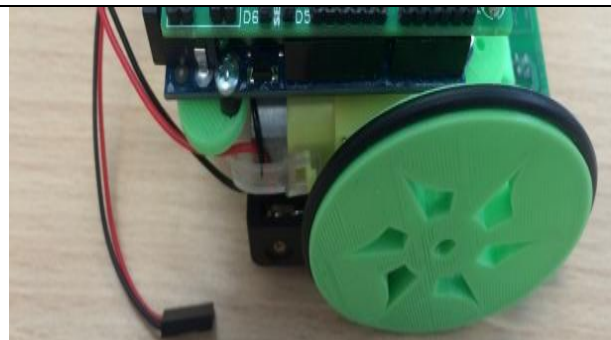
9.-Ponemos la placa de los sensores CNY70 y la sujetamos con dos tuercas M3.

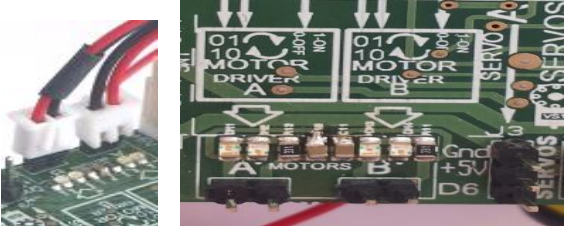


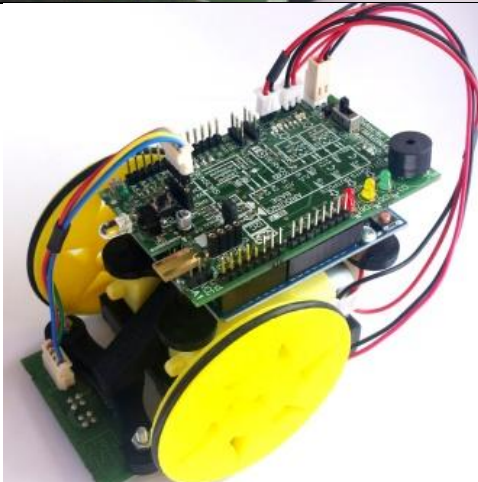


10.-Montamos las juntas a las ruedas.

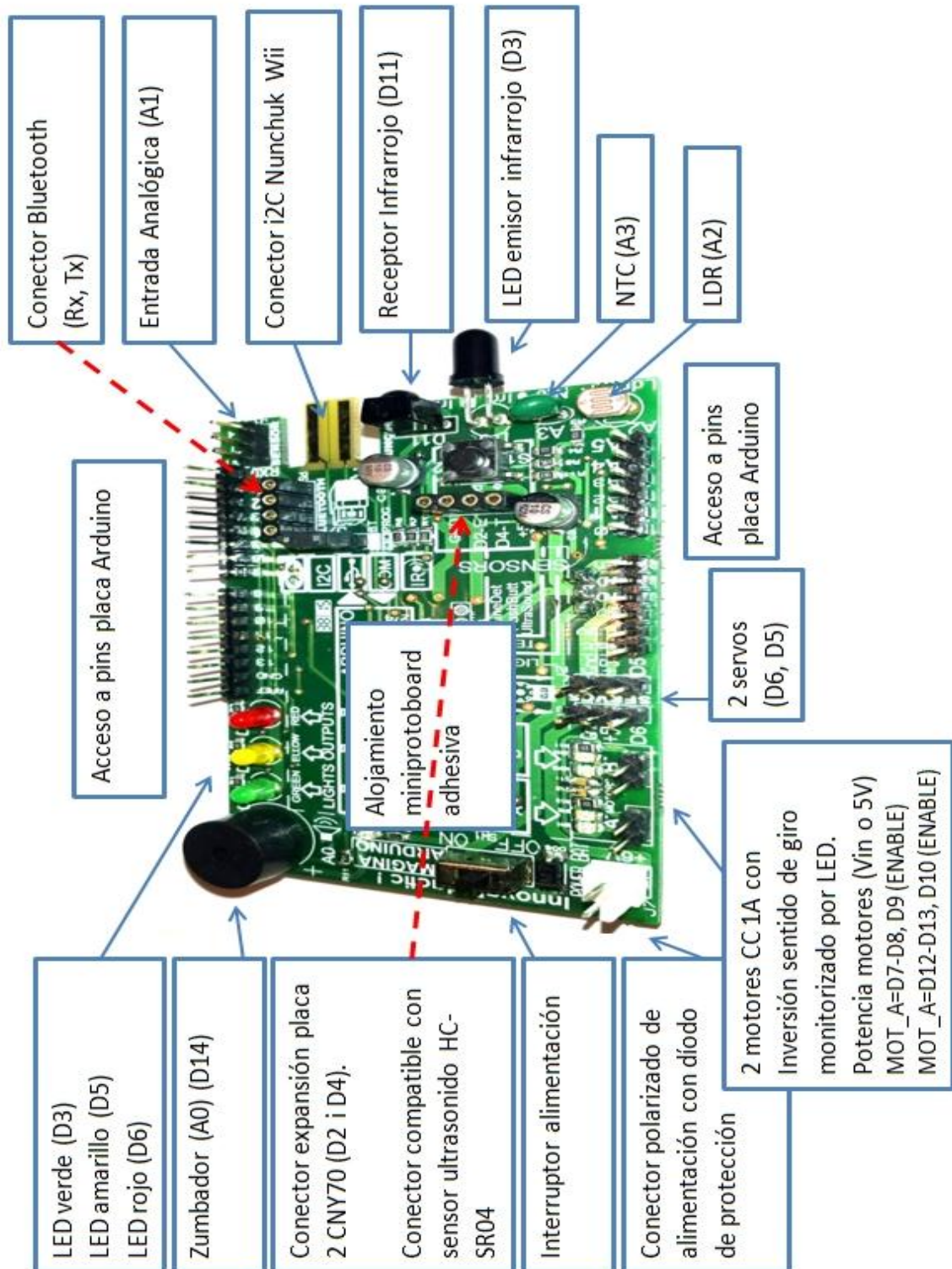


11.-Colocamos las ruedas a presión con la cara lisa al exterior.



<p>12.- Conectamos los motores.</p> <p>*El motor A es el de la izquierda y el motor B el de la derecha, en el sentido de la marcha.</p>	
<p>13.-Conectamos el porta pilas.</p> <p>*El cable rojo es el positivo (+ 5V) y el negro el negativo (- GND).</p> <p>*Esta placa no lleva ningún regulador de tensión, por lo que no se puede alimentar con más de 6V(4 pilas AA).</p>	
<p>14.-Conectamos los pins de la placa driver: VCC, D4 , D2 y GND, con los de la placa sensores que se llaman igual.</p> <p>*Los cables VCC(+5v) y GND de uno y otro extremo deben tener el mismo nombre.</p>	
<p>15.-Finalmente y tenemos al robot construido. Ya está listo para programarlo. Consultar el índice.</p>	

Anexo2-Posibilidades del Shield Imagina Arduino.



Anexo3-Programación arduino con mBlock.

- **Preparación**

1.-Descargar e instalar AdobeAIRInstaller.exe

<http://get.adobe.com/air/>

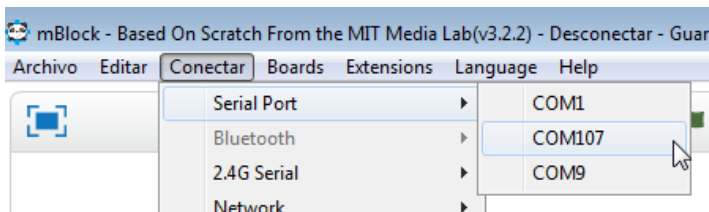
2.- Descargar mBlock e instalarlo.

<http://mblock.cc/download/>

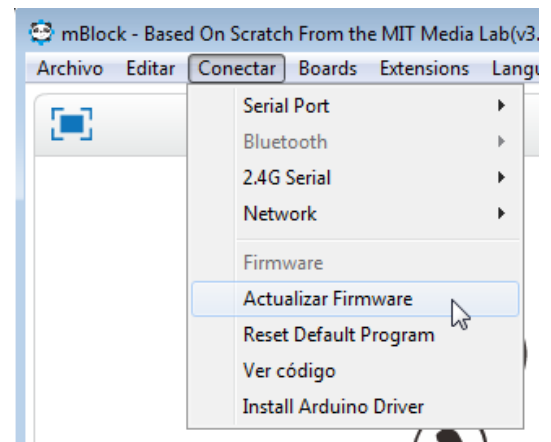
- **Utilización Modo 1 (con cable USB conectado al ordenador)**

1.- Ejecutar mBlock.

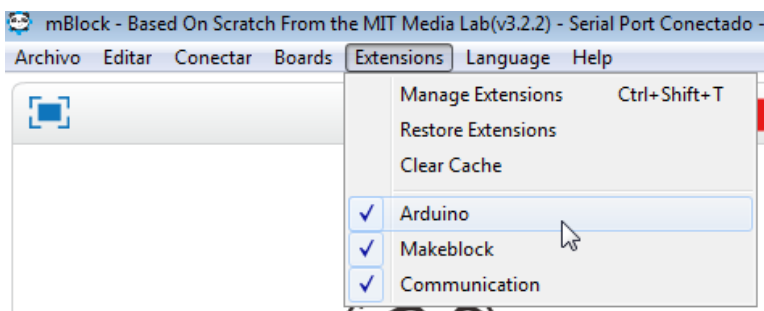
2.-Seleccionar el puerto COM que el Windows nos ha asignado (COM1, COM2, COM3...) al conectar la placa Arduino Uno.



3.-Seleccionar *Actualizar Firmware*. Esta acción carga un programa de comunicaciones en la placa Arduino Uno.



4.- Activamos la extensión de Arduino y la de Comunicaciones.



5.-Para probar el funcionamiento podemos introducir el siguiente programa.



Al pulsar la bandera verde la salida D5 que corresponde al LED amarillo hará una intermitencia cada segundo.

Para detener el programa hay que pulsar encima del botón de color rojo.

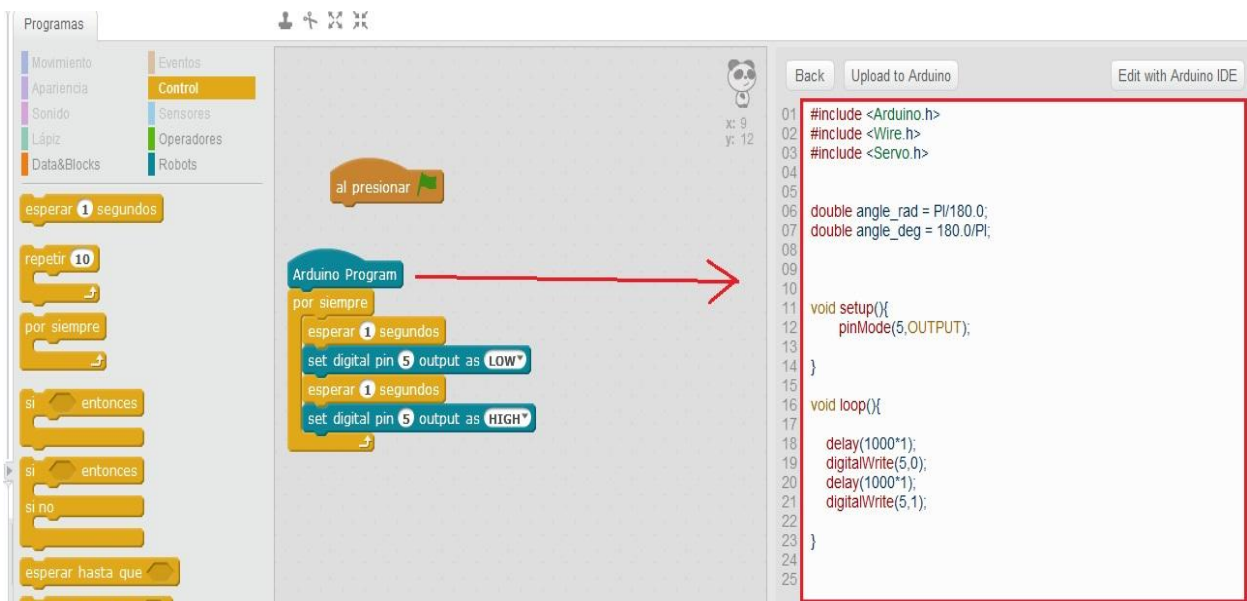


- **Utilización Modo 2 (sin cable USB y alimentando la placa)**

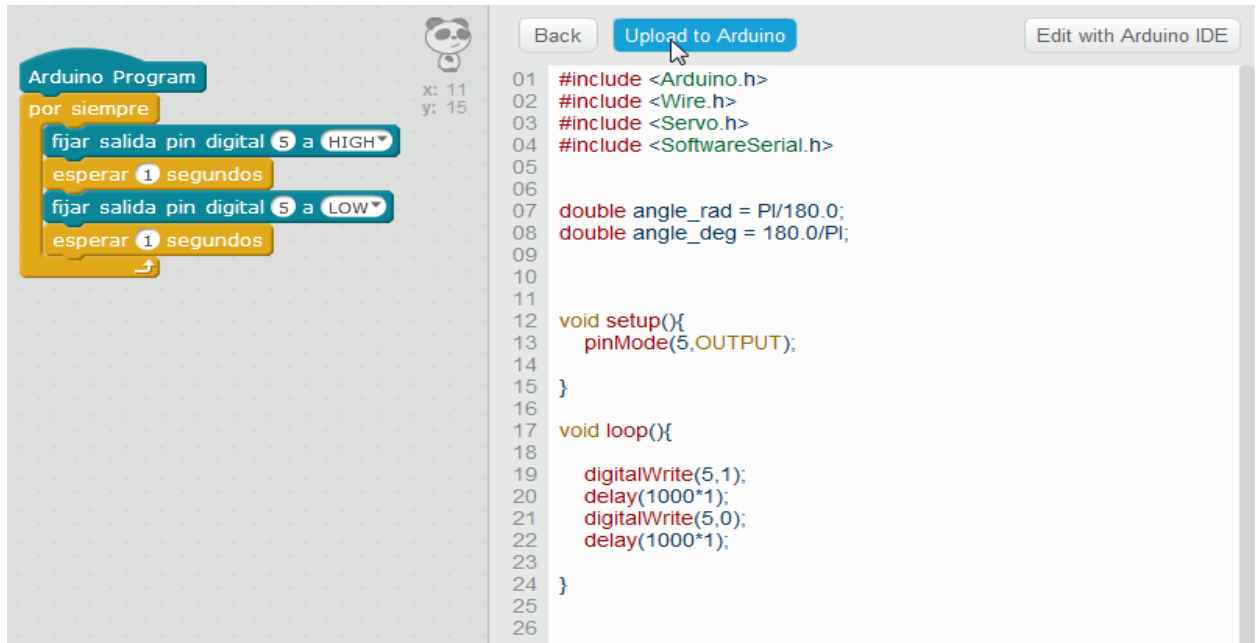
6.- Si ahora nos interesa pasar este programa a la placa Arduino Uno para trabajar desconectada de mBlocky del ordenador, tenemos que hacerlo sustituyendo la bandera por la instrucción Arduino.



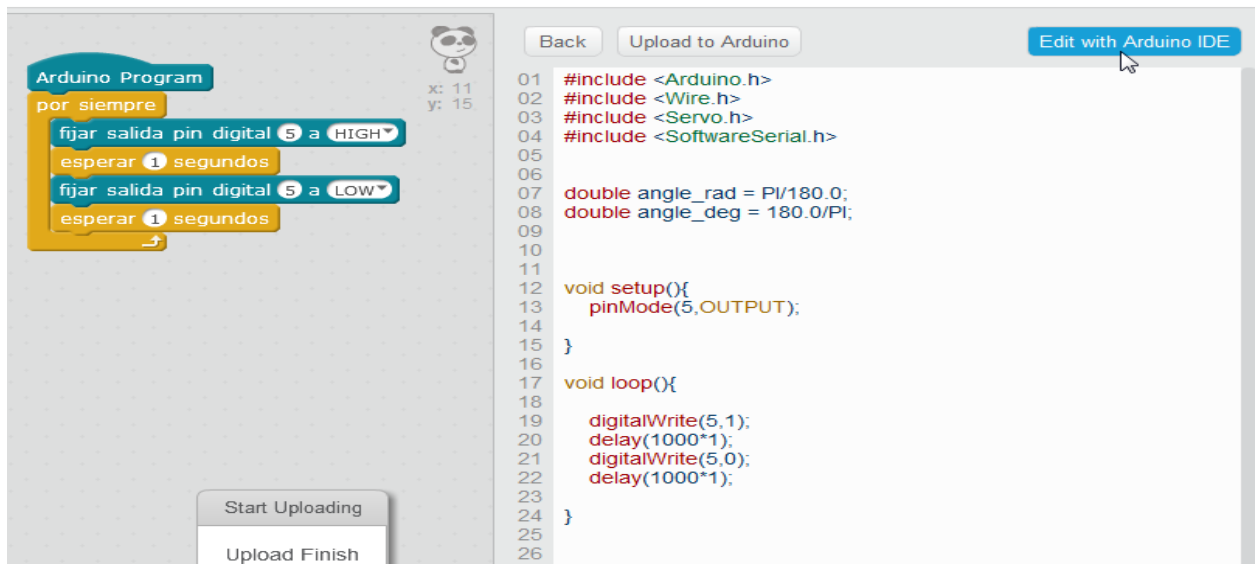
7.-Hacemos clic encima de "Arduino Program".



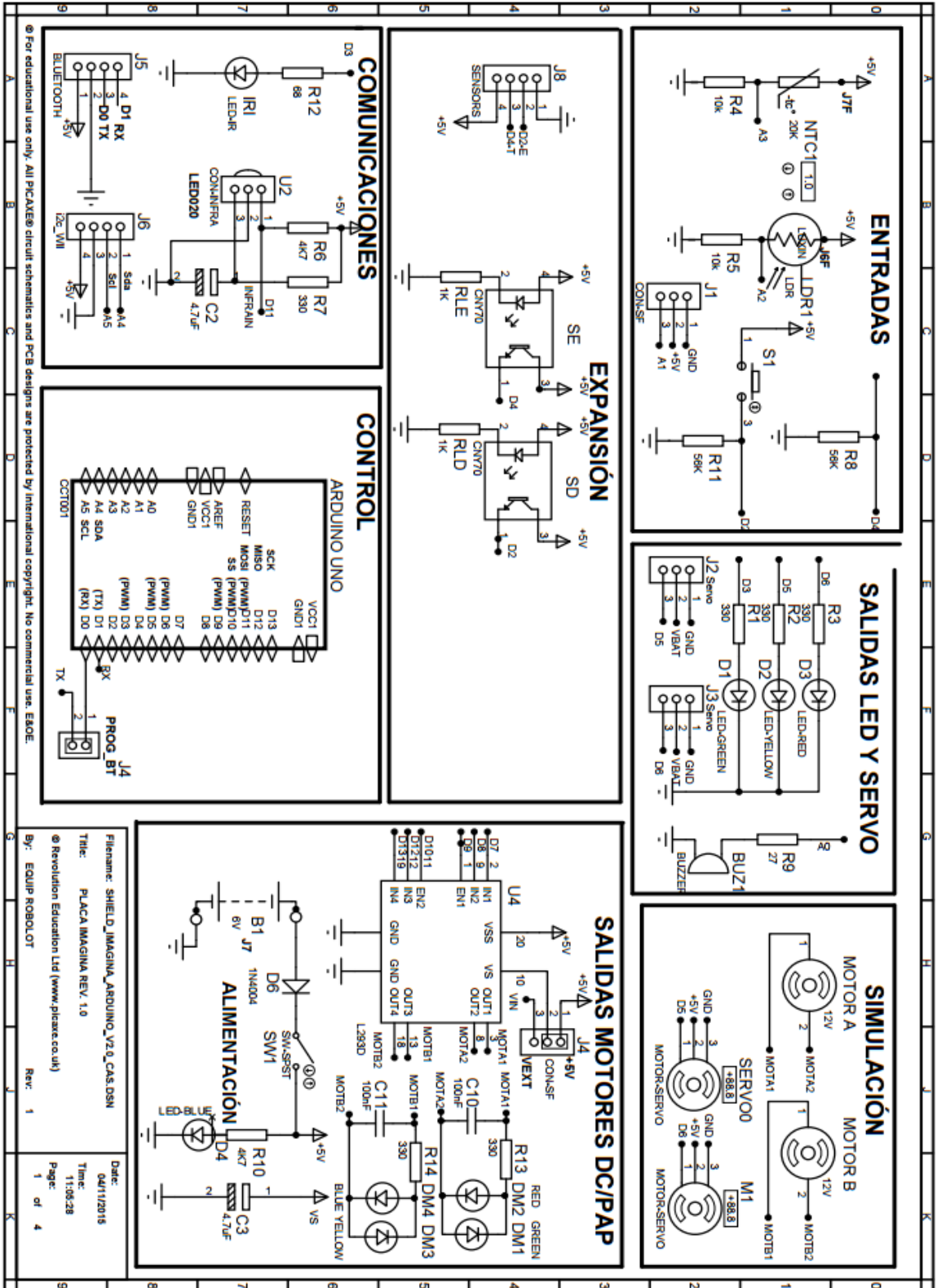
8.- Hacemos clic en "UPLOAD TO ARDUINO", esperamos unos segundos y ya tendremos el programa introducido en la placa Arduino. Entonces desconectamos el cable USB, alimentamos con pilas o un powerbank y funcionará autónomamente.



9.- También existe la posibilidad de editarlo para corregir y añadir código desde "EDIT WITH ARDUINO IDE".



Anexo 4.-Esquema electrónico de la placa Imagina.



© For educational use only. All PICAXE® circuit schematics and PCB designs are protected by international copyright. No commercial use. E&OE.

Filename: SHIELD_IMAGINA_ARDUINO_V2.0_CAS.DSN
 Title: PLACA IMAGINA REV. 1.0
 © Revolution Education Ltd (www.picaxe.co.uk)
 By: EQUIP ROBOLOTT
 Rev: 1
 Date: 04/11/2015
 Time: 11:56:28
 Page: 1 of 4