

Planificación de las prácticas

Curso 2018

Descripción

Este documento describe la línea de aprendizaje que seguiremos según habilidades y dinámica del grupo y en las tres áreas de aprendizaje (programación, electrónica y mecánica).

En este grupo trabajamos con chavales que ya han cursado al menos un año de Robótica, al menos conocen Scratch y han montado algún robot sencillo.

Esta primera parte de la planificación cubre lo que haremos en el primer trimestre, luego según el nivel iremos diseñando como seguiremos.

Áreas de Aprendizaje

1. Programación/Software

Este año saltamos de entornos de programación básicos, Scratch a programación en C. Para ello trabajaremos con **ARDUBLOCK**, entorno de desarrollo que nos permite programar con bloques y ver la traducción en código C.

Si tenemos en la clase algún alumno que prefiera hacer más programación introduciremos Python.

El objetivo a lo largo del año conseguir asimilar los conceptos básicos de la programación:
Instrucciones de control, condicionales, operadores matemáticos, operadores lógicos y variables. Flujogramas.

2. Electrónica

Introducir los conceptos de electricidad sobre una base práctica. Montajes con luces y motores para comprender la continuidad, montajes serie y paralelo... usaremos motores, luces, cables, interruptores y pilas.

En principio trabajaremos estos conceptos de forma aislada sobre montajes en placas de ensayo... una vez dominados los conceptos introduciremos el control de estos circuitos con Arduino y la shield I/O.

Nos familiarizaremos con los componentes usando una técnica divertida, la ingeniería inversa de abrir y destripar cacharos.

Aprenderemos a soldar con estaño, para llegar a montar nuestra propia shield I/O.

3. Mecánica

Objetivo es tener a lo largo del año una idea la importancia de la mecánica y saber identificar las distintas estructuras que usamos a lo largo del día.

Mecánica elemental: palancas, poleas, fuerzas,

Sistemas de tracción: gomas, gravedad, motores, viento

Estructuras: brazos, plataformas elevadores, hélices...

Clasificación de elementos por medidas y formas.

Estudio de los distintos tipos de plataformas mecánicas usadas en robótica: móviles con dos ruedas, tres ruedas, giros con servos o por diferencia de velocidad de motores, insectos.

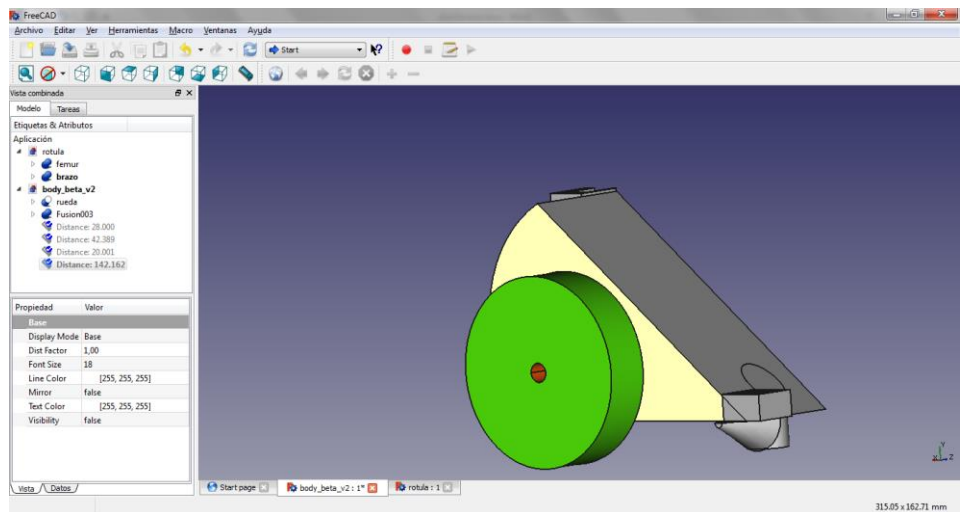
Aplicaremos estos conceptos durante la creación de nuestros robots, tendremos varios grupos trabajando con distintas estructuras para poder visualizar ventajas e inconvenientes de los distintos diseños.

4. Robótica

Construiremos nuestro bot, creando la estructura, instalando motores y sensores y finalmente programaremos su comportamiento.

Ofreceremos varias plataformas básicas para que los chavales elijan: triciclo, cuatro ruedas... Luego sobre esta base iremos personalizando los diseños al gusto de cada uno.

En diseño



Electrónica, conceptos de electricidad

Descripción

Como base para empezar a descubrir la electrónica tenemos que aprender unos conceptos básicos de electricidad.

Circuitos electricidad Básicos: símil h2o

| | | |
|--------------|--------------|----------------|
| Electricidad | Bombilla | |
| Interruptor | Motor | serie/paralelo |
| Baterías | | |
| Cables | conmutadores | |

Prácticas:

| | |
|--|---|
| Ctos básicos con cable y regletas | Interruptor. Conmutadas. Pulsadores. Serie y Paralelo |
| Puerta corredera garaje Barrera paso tren. Ascensor. | Haciendo circuitos con pila cables y conmutadores/interruptores |

Lógica Binaria:

Explicamos las puertas lógicas básicas y hacemos circuitos sencillos de puertas lógicas.

| | |
|-----------------|---|
| | |
| AND OR XOR NAND | Trabajamos con protoboards chip y leds. |

Juegos

Buscaminas con concepto AND.

Simulación:

Todos estos conceptos y prácticas se pueden simular con scratch / S4A

Scratch, aprendiendo a programar

Descripción

Línea de formación para aprender los fundamentos de la programación de forma lúdica y divertida. Trabajaremos por secciones los distintos conceptos, intentando captar las ideas de forma intuitiva.

Trabajar en paralelo la visión de Ardublock y de en 'C'

Instructions de control:

Comienzo de programa.
Por siempre (while)
Repetir 'n' (for)

Esperar hasta
Si (if) Si no (else)
Detener

Para trabajar estos conceptos haremos pequeños montajes usando la placa scratach I/O.

Operadores lógicos y matemáticos:

And
Or
Not
Suma
Resta

Multiplicar
Dividir
Redondear
Modulo
Azar

Variables y listas. Local vs Global, constantes y contadores.

Flujogramas.

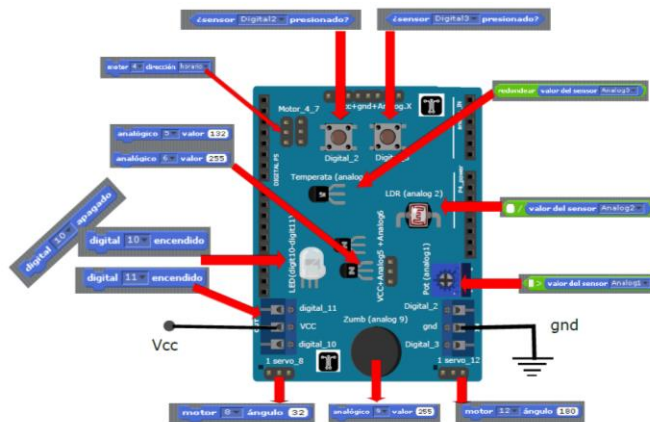
Funciones, mensajes.

Simuladores

Concepto de simulador y realidad. Por ejemplo, realizar una práctica con un pulsador simulado en pantalla que encienda una luz para luego hacer un pulsador real usando Arduino.

Más complejos hacer Simulador del tanque para luego en mecánica realizar la misma práctica.

Trabajo con S4A y Ardublock



Poco a poco iremos introduciendo algunos conceptos de electrónica y control con Ardublock. A medida que avancemos en los conceptos de electrónica trabajando estas sencillas prácticas.

Documentación para Ardublock:

<https://rhasl.files.wordpress.com/2014/09/ide-arduino-arddublock.pdf>

Software (ojo que hay versión por la red que no funcionan)

<http://www.carobot.cc/how-to/ardublock/>

Algunas ideas sobre temáticas de programación para empezar.

Programación de “amigos ciegos”, Juego de la programación: un niño en el centro de la clase guiado por las ordenes que le da un compañero para realizar una tarea (anda tres pasos, gira derecha, coge...)

Con el Lápiz de scratch dibujar figuras geométricas, ecuaciones...

Videojuegos

Gravedad

Fondos

Mandos

Lanzar proyectiles

Volantes

Marcianitos

Laberintos

Máquinas: ascensor, parquímetro.

Primeros pasos con Arduino//S4A//Ardublock

Para empezar a programar con Scratch y Arduino, tenemos que sincronizarnos con el taller de electrónica en el que montaremos y probaremos la shield de Arduino scratch_io. Luego tendremos que preparar la placa de Arduino cargando el firmware de s4a, que permite comunicar la placa con el ordenador.

Práctica #Cero: Juego del Frontón

Objetivo

Mediante el desarrollo de este videojuego, vamos a aprender programación con scratch y el manejo de un sensor analógico.

Duración de la práctica

Depende edad y conocimientos previos, pero normalmente 2 horas.

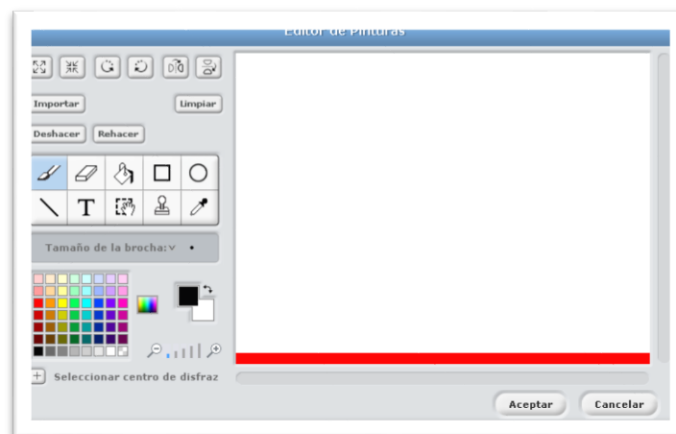
Conocimientos previos

Entendemos que esta práctica se desarrolla con alumnos que están dentro de un programa educativo de programación y robótica... Llegado este punto tiene una buena base de metodología de programación es Scratch. Aunque también se puede utilizar esta práctica dentro de las unidades de enseñanza de Scratch.

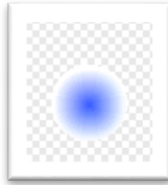
Algo de teoría sobre el entorno de trabajo

En este proyecto vamos a utilizar cuatro objetos:

El escenario: esta dibujado como un fondo blanco con una línea roja en la parte baja. Esta línea roja no sirve para saber cuando la raqueta no golpea la bola, y por lo tanto el jugador ha fallado un tanto.



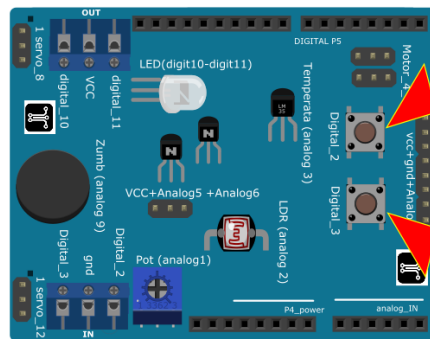
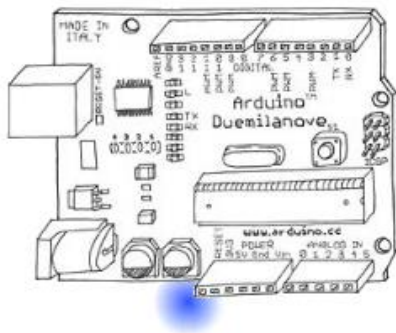
Bola: pintamos una bola redonda. Nada especial, nosotros hemos cogido un color azul.



La raqueta: al igual que la bola, esta dibujada como una raya ancha de color negro. Es este caso el color es importante, porque nos servirá en la programación para detectar cuando la raqueta golpea la bola.



La placa Arduino y la shield scratch_io: este objeto nos informara de cuando el jugador pulsa el botón derecho o el izquierdo de nuestra placa. Usaremos esta información de la pulsación para mover la raqueta con Scratch en la dirección apropiada.



Desarrollo guiado

Estudiaremos ahora como implementamos el código en Scratch para cada uno de los objetos que compone el programa.

En Scratch tenemos varios hilos de programa corriendo en paralelo, en nuestro caso, cuando pulsamos la banderita de comienzo, todos los programas de todos los objetos que tienen la

bandera  se empiezan a ejecutar.

a) Paso: Objeto Bola

El objeto bola tiene tres hilos de programa ejecutándose a la vez, uno controla cuando la raqueta falla, otro cuando la bola es golpeada por la raqueta y otro que hace que la bola rebote al tocar un borde del escenario. Vamos a estudiarlos:

Primer hilo de programa del objeto Bola:



Comienzo de esta parte del programa al pulsar “comenzar”,

Vemos la bandera verde de comienzo de programa en la esquina superior izquierda de la pantalla.



Estas dos instrucciones siguientes, hacen visible el objeto bola y lo colocan en la posición de la pantalla (x=13, y=157)

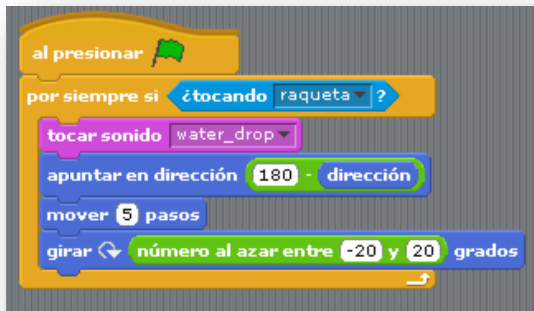


Esta instrucción espera a que la bola toque el color rojo. Si recordáis el color rojo es la línea de base que hemos colocado en el escenario. Si la bola toca la línea roja significa que la raqueta ha fallado y la partida se acaba.



Las siguientes tres instrucciones, hacen desaparecer la bola, ejecuta el sonido "game over" (que hemos grabado) previamente y detenemos el programa.

Segundo hilo de programa del objeto Bola:



Observamos que una vez pulsada la bandera de comienzo de programa, entramos en un bucle que se repite "por siempre si" se cumple una condición.

En este caso la condición es ¿tocando la raqueta?... ¿está tocando el objeto bola la raqueta?, si la respuesta es afirmativa entramos a ejecutar el código en el interior del bucle, si la respuesta es negativa no se ejecuta el código del interior y sigue esperando por siempre y analizando a la condición a la espera que la bola golpee la raqueta.



El código del interior del bucle, emite el sonido del golpeo de la pelota en la raqueta “tocar sonido”. Cambia la dirección de movimiento de la bola en sentido opuesto (restando 180º a la dirección de llegada). Movemos un poco la bola para que deje de

tocar el rojo y se libere el bucle. Finalmente añadimos un pequeño giro a la dirección de la bola para que no siempre siga el mismo recorrido y añadir un poco de diversión al juego.

Tercer hilo de programa del objeto Bola:

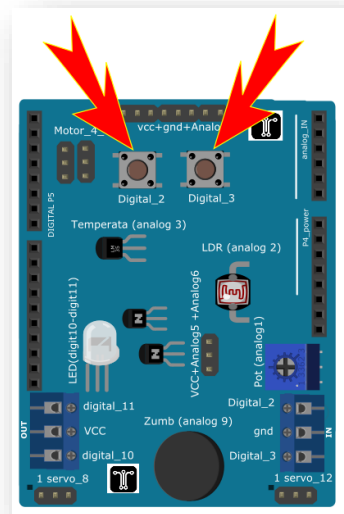
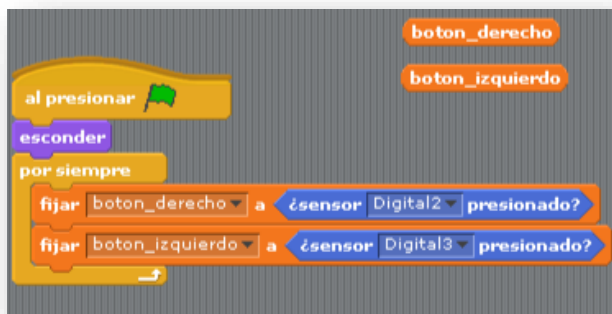


Este último hilo del programa que trabaja sobre el objeto bola y que se ejecuta en paralelo al pulsa la bandera.

Como vemos estamos dentro de un bucle infinito “por siempre”: el código que está dentro del bucle se ejecuta eternamente y sin esperar a ninguna condición. Parará

su ejecución cuando el programa se detenga de forma general. Esta parte del código hace que la bola rebote cuando toca un borde del escenario.

b) Paso: Objeto Arduino // scratch io Shield



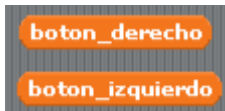
El objeto Arduino nos permite interactuar con el exterior y condicionar el funcionamiento de nuestro programa a las acciones que ocurran a nuestra placa. En este caso vamos a utilizar los pulsadores de la placa, con este objeto sabremos que botón está pulsado en la placa y usaremos esta información para mover la raqueta de derecha a izquierda.

Primero escondemos el dibujo de la placa en la pantalla, solo tenemos que ver la raqueta y la bola en la pantalla.

Luego en un bucle infinito miramos el estado de los pulsadores, para saber cuál está pulsado.



Sabemos que los pulsadores están conectados en las entradas etiquetadas como 'Digital2' y 'Digital3'. Esta condición es 'VERDADERA' cuando el pulsador está accionado y 'FALSA' cuando no está pulsado.



hemos creado dos variables "boton_derecho" y "botón_izquierdo" para almacenar el valor del pulsador 'VERDADERO' o 'FALSO', según esté pulsado o liberado el pulsador de la placa.

Usaremos estas variables para mover el objeto 'raqueta', por ahora solo actualizamos su valor.

c) Paso: Objeto Raqueta



El código que regula el comportamiento de la raqueta tiene el objetivo de mover la raqueta de izquierda a derecha según el pulsado de la placa que acciones el jugador.

Primero vemos que tenemos un bucle infinito "por siempre".



Dentro del bucle infinito tenemos dos instrucciones de control 'si condición', el código que tenemos dentro del 'si' solo se ejecuta si la condición es cierta (valor =1)



Este operador compara dos valores y da como resultado VERDADERO o TRUE o '1' si los dos valores son iguales. Da FALSO o FALSE o '0' si los dos valores son distintos.



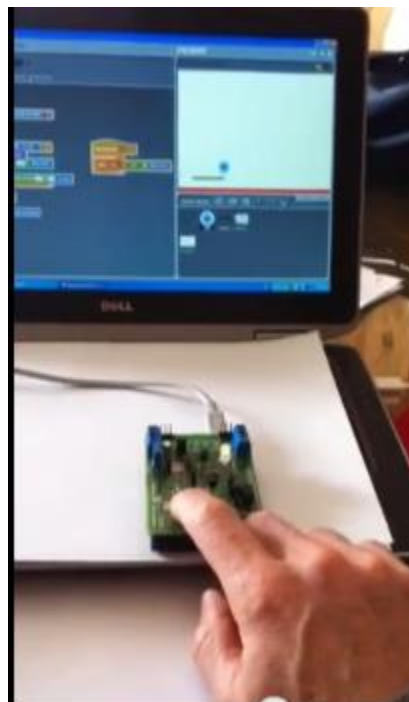
En nuestro código miramos si la variable 'botón_izquierdo' es VERDADERA o TRUE, lo que indica que el pulsador izquierdo está pulsado. Recordar que esta variable la actualiza el objeto 'Arduino' a valor VERDADERO cuando el pulsador está accionado.



La siguiente acción es mover un poco la raqueta siguiendo la instrucción que ha dado el jugador en el sentido que desea. En nuestro caso movemos 8 unidades, si queremos que la raqueta se mueva más rápido podemos aumentar este valor.

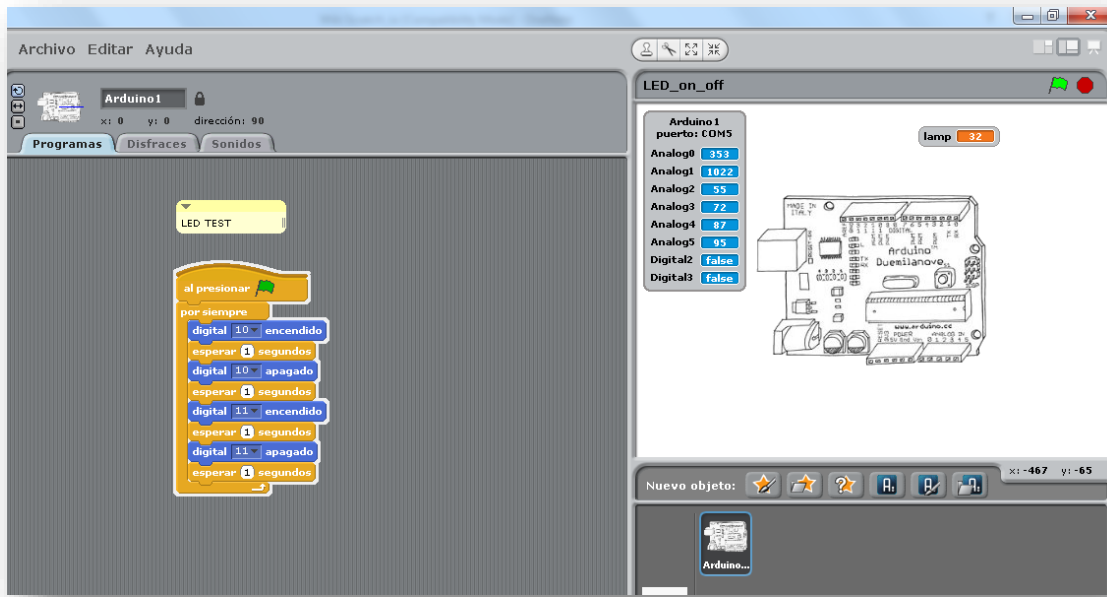
Resultado final:

<https://www.youtube.com/watch?v=w11je0dc468>



Práctica #1. Encendido Apagado de LED bicolor

Esta sencilla práctica enciende y apaga el diodo bicolor de la shield_IO. El código necesario es el que aparece en la pantalla.



En este link QR podéis ver un video de la práctica funcionando:

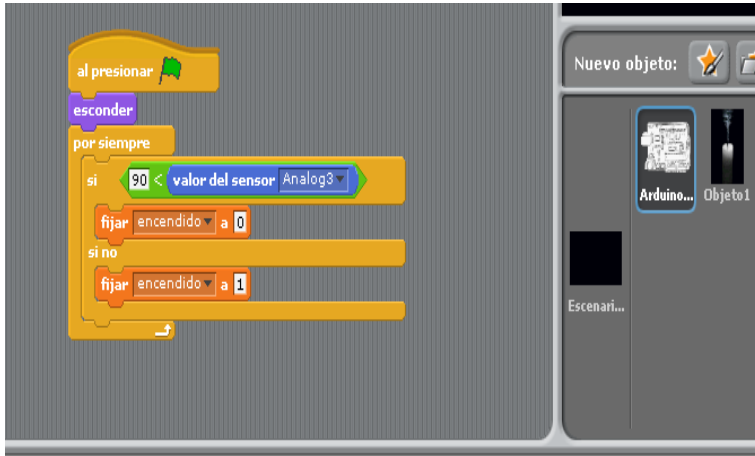
<https://www.youtube.com/watch?v=YVgjinm5oSs4>



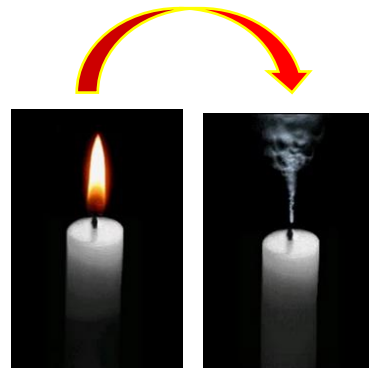
Práctica #2. El Dragón enciende velas

Esta curiosa práctica utiliza el sensor de temperatura soldado en la placa (LM35), para detectar la subida de temperatura generada por el aliento del chaval sobre la placa... este aumento de temperatura anima la imagen en pantalla para encender la vela.

Código del Objeto ARDUINO. Lee el valor del sensor de temperatura, conectado en la patilla “analog3” y lo compara con un valor fijo ‘90’ (salida del sensor para la temperatura ambiente). Cambiamos el valor de la variable ‘encendido’ de cero a uno cuando la temperatura sube (porque el chaval lanzar el aliento sobre el sensor, subiendo la temperatura de éste):



El código del Objeto1 ‘la vela’, toma el valor de la variable ‘encendido’ para cambiar el disfraz de la vela encendida a la vela apagada y viceversa:



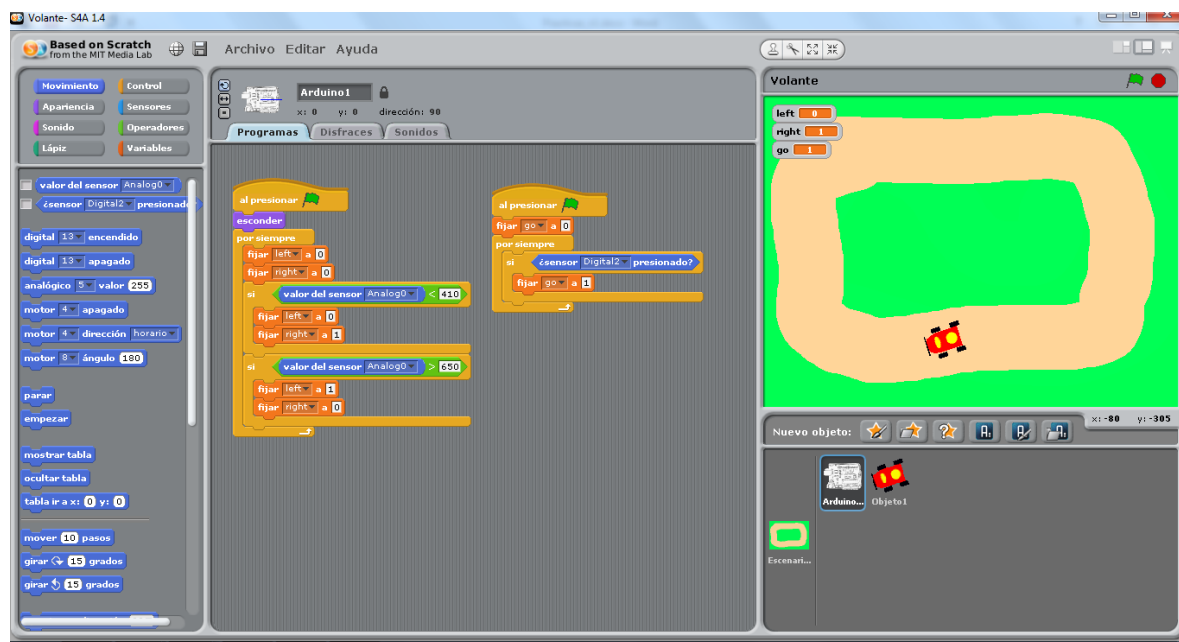
Aquí podeis ver el video de la práctica funcionando:

https://www.youtube.com/watch?v=OONQVJ_bflc



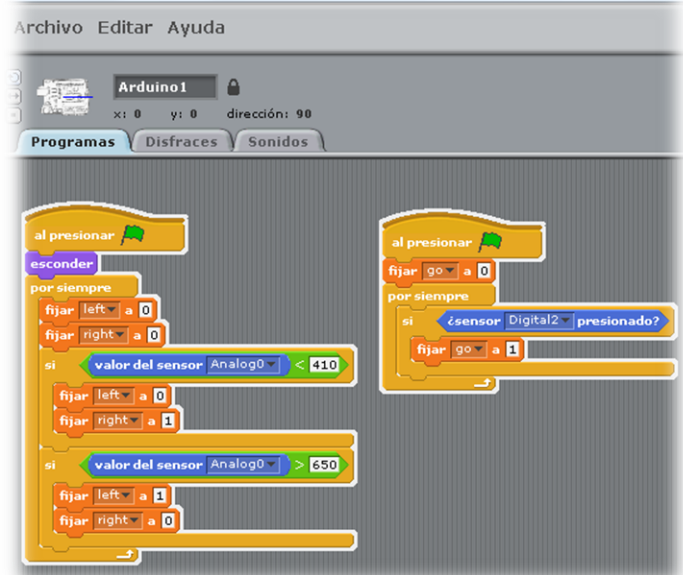
Práctica #3. Video Juego volante de mi coche.

En esta práctica usamos la entrada analógica y la resistencia variable para implementar un Joystick sencillo que nos permite controlar un simulador de coches. Practica sencilla pero muy interesante para comenzar con los videojuegos.



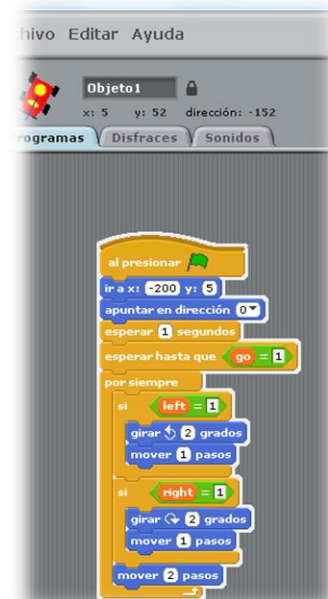
Código del arduino scratch IO

La placa del Arduino, hace dos tareas. Primero espera a que se pulse el pulsador 'Digital2' para empezar a mover hacia delante el cochecito. También analiza el giro del potenciómetro por la vía analógica y actualiza las variables 'left' y 'right' según giremos el volante a la izquierda o derecha. Si está en la posición central ni derecha, ni izquierda pero recto.



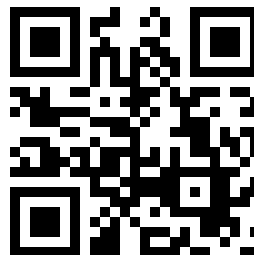
Código del objeto coche:

El programa que controla el movimiento del cochecito, primero lo coloca en la posición inicial y una vez pulsado el arranque 'go'... comienza a moverlo y girarlo según nos indique las variables 'left' y 'right'. Puede seguir recto o girar dos grados en uno u otro sentido de giro.



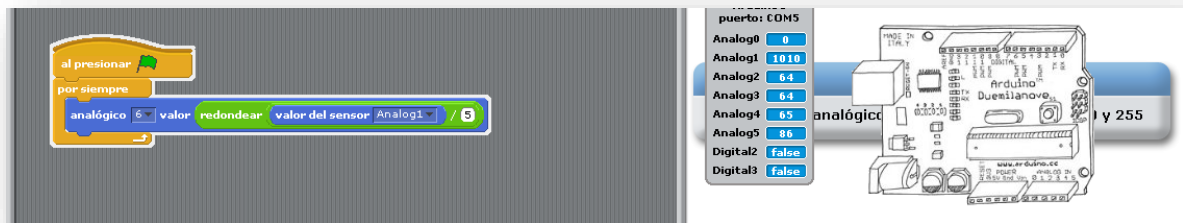
Puedes ver un video de la práctica:

<https://youtu.be/BLcEb11tfjM>



Práctica #4. Controlando la luminosidad de una bombilla con la claridad del día.

En esta práctica ...



El video.

<https://www.youtube.com/watch?v=MjXg0g8XwQs>

