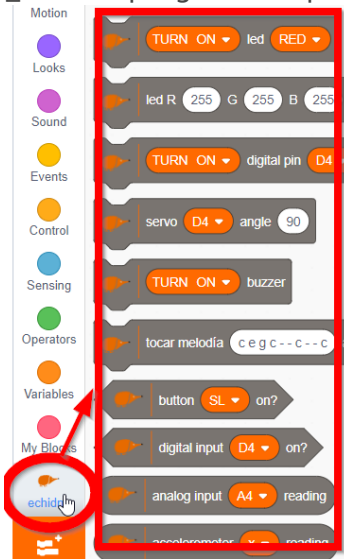


¿Qué lenguaje gráfico es mejor para Echidna?

EchidnaScratch

- Es un programa específico para Echidna, tiene ya **instrucciones especiales**



Esto es **UNA ENORME VENTAJA** pues permite que el sprite interactúa con Echidna **sin necesidad de usar mensajes y variables globales** como en mBlock

- **EchidnaScratch** es un programa que permite usar la placa **EN VIVO** esto nos permite interactuar con el ordenador
- **No** permite **CARGA** es decir, dependemos del ordenador, no podemos cargar el programa en el Echidna y que trabaje independientemente de PC.
- EchidnaScratch esta integrado con la **INTELIGENCIA ARTIFICIAL** LearningML
- La detección Makey Makey de los pines lo detecta igual que mBlock

mBlock

- Es un programa muy extendido multiplaca-
- Permite programar **EN VIVO** igual que EchidnaScratch
- Pero también permite **CARGAR EL PROGRAMA EN LA PLACA lo que permite crear proyectos independientes del ordenador**
- EchidnaBlack o Echidna Shield lo lee como si fuera un **Arduino UNO**
- Para la comunicación entre el Device Arduino = Echidna y los diferentes Sprites del programa hay que usar **MENSAJES o VARIABLES GLOBALES** lo que hace engorroso cualquier programa

- □ También permite la IA pero con extensiones, ver <https://libros.catedu.es/books/robotica-educativa-con-mbot/page/cognitive-services-servicios-cognitivos> pero no está tan integrado como EchidnaScratch
- □ Detecta bien los pines Makey Makey

Snap4Arduino

- □ También es un programa que permite usar EchidnaScratch **en vivo**
- □ Tiene **instrucciones especiales** para la placa
- □ No es tan amigable como los anteriores.

¿ POR CUAL NOS DECANTAMOS ?

- Si vas a trabajar en modo gráfico programas con o sin Makey Makey **ECHIDNASCRATCH**
- Si vas a trabajar con IA **ECHIDNASCRATCH**
- Si vas a trabajar con proyectos que no quieres que se ejecuten con el PC **mBlock**
-

EN RESUMEN, al menos que te interese trabajar con proyectos donde quieres CARGAR el programa en la placa, **RECOMENDAMOS ECHIDNA SCRATCH**

- **OJO si pasas de MBLOCK a ECHIDNASCRATCH hay que cargar Firmata** ver <https://libros.catedu.es/books/echidna/page/echidnascratch-no-detecta-echidna>
- **OJO si pasas de ECHIDNASCRATCH a MBLOCK hay que cargar el Firmware de mBlock** ver <https://libros.catedu.es/books/echidna/page/mblock-no-detecta-echidna>
- **Por lo tanto es mejor no ir cambiando de programa**

P: Por defecto cuando lo compro ¿Qué tiene cargado? ¿Firmata para usarlo con EchidnaScratch o Firmware para usarlo con mBlock?

R: Firmata para usarlo con EchidnaScratch

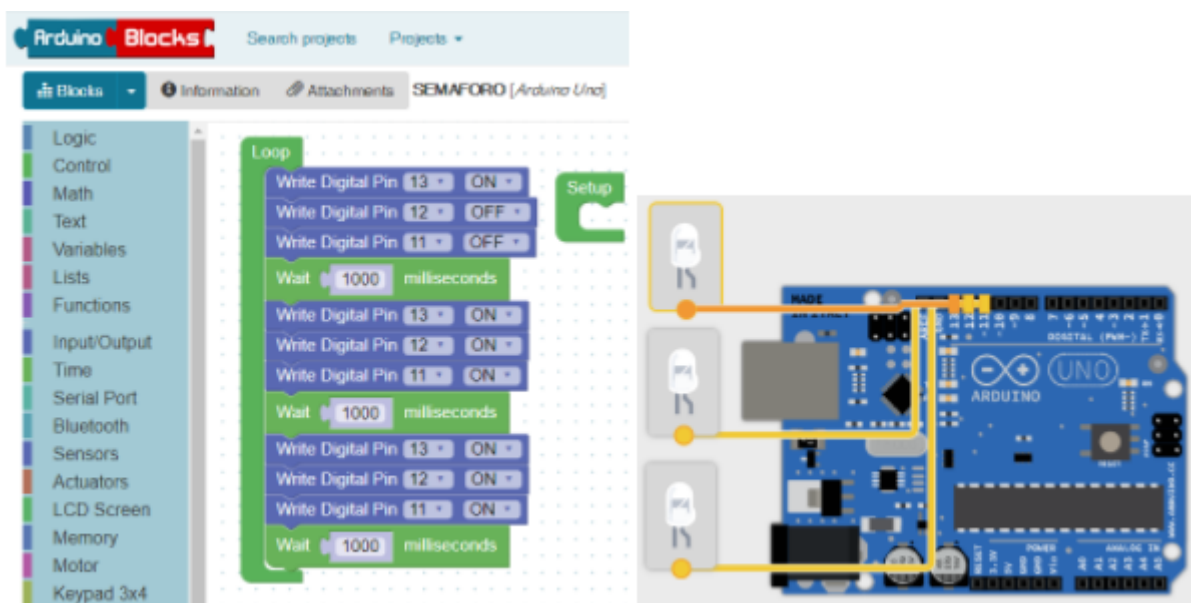
Otros programas gráficos

Existen otros programas gráficos, el ArduinoBlocks, Bitblock... pero no tienen la posibilidad de interaccionar con Sprites del ordenador, ni tienen el mismo entorno del referente Scratch, por lo que no se utilizarán en este curso, pero son posibilidades interesantes, os mostramos dos capturas de pantalla del mismo programa semáforo visto anteriormente, fíjate que aquí no hay *Sprite*:

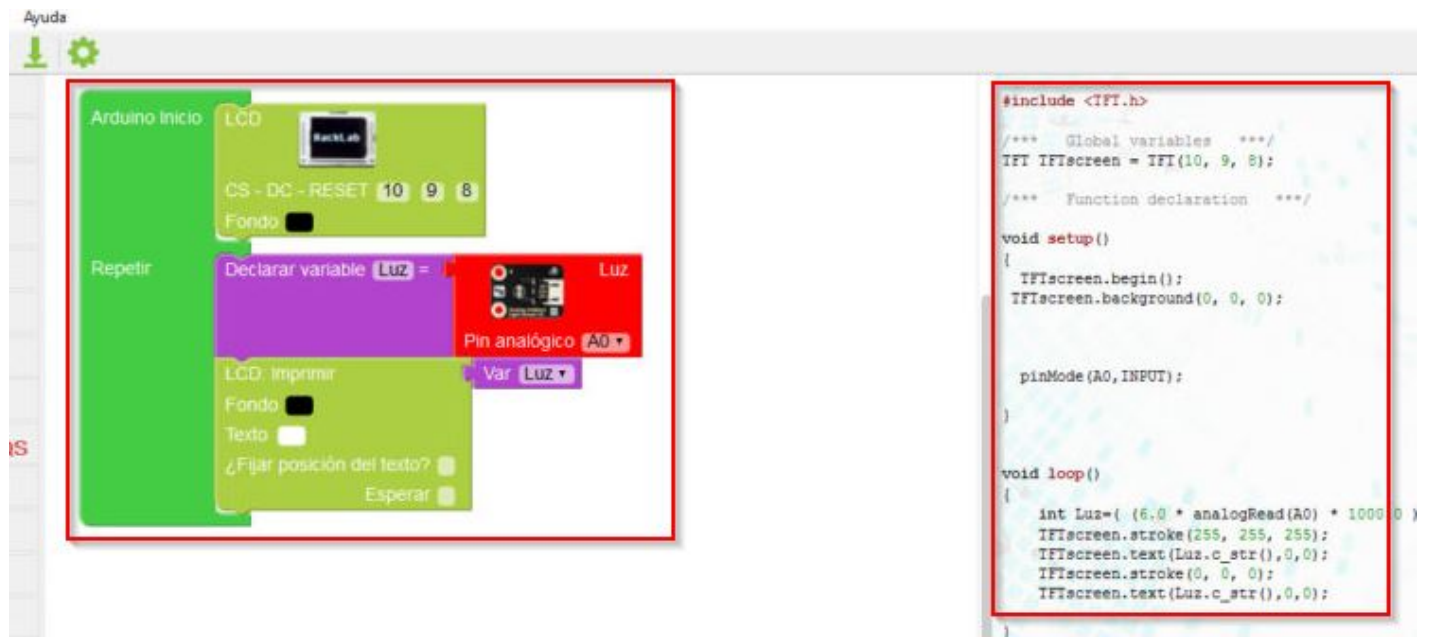
Con Bitbloq:



Con ArduinoBlocks:



Y un favorito nuestro el **VISUALINO** pues a la vez que haces programación gráfica, va enseñando el código a la derecha:



The image shows the Visualino IDE interface. On the left, a block-based program is visible, enclosed in a red box. It starts with an 'Arduino Inicio' block, followed by an 'LCD' block with pins 10, 9, and 8. Then, a 'Repetir' loop contains a 'Declarar variable Luz = 0' block, a 'Pin analógico A0' block, and an 'LCD imprimir' block. The 'LCD imprimir' block has a 'Fondo' dropdown set to black, a 'Texto' input field, and checkboxes for '¿Fijar posición del texto?' and 'Esperar'. On the right, the corresponding C++ code is shown, also enclosed in a red box. The code includes the <TFT.h> header, defines global variables for the TFT screen, and implements the setup and loop functions. The loop function reads the analog value from pin A0, calculates the light intensity (Luz), and prints it to the TFT screen.

```
#include <TFT.h>

/** Global variables */
TFT TFTscreen = TFT(10, 9, 8);

/** Function declaration */

void setup()
{
  TFTscreen.begin();
  TFTscreen.background(0, 0, 0);

  pinMode(A0, INPUT);
}

void loop()
{
  int Luz= (6.0 * analogRead(A0) * 1000.0);
  TFTscreen.stroke(255, 255, 255);
  TFTscreen.text(Luz.c_str(),0,0);
  TFTscreen.stroke(0, 0, 0);
  TFTscreen.text(Luz.c_str(),0,0);
}
```

Revision #30

Created 31 March 2022 08:48:42 by Equipo CATEDU

Updated 20 December 2024 15:00:45 by Javier Quintana