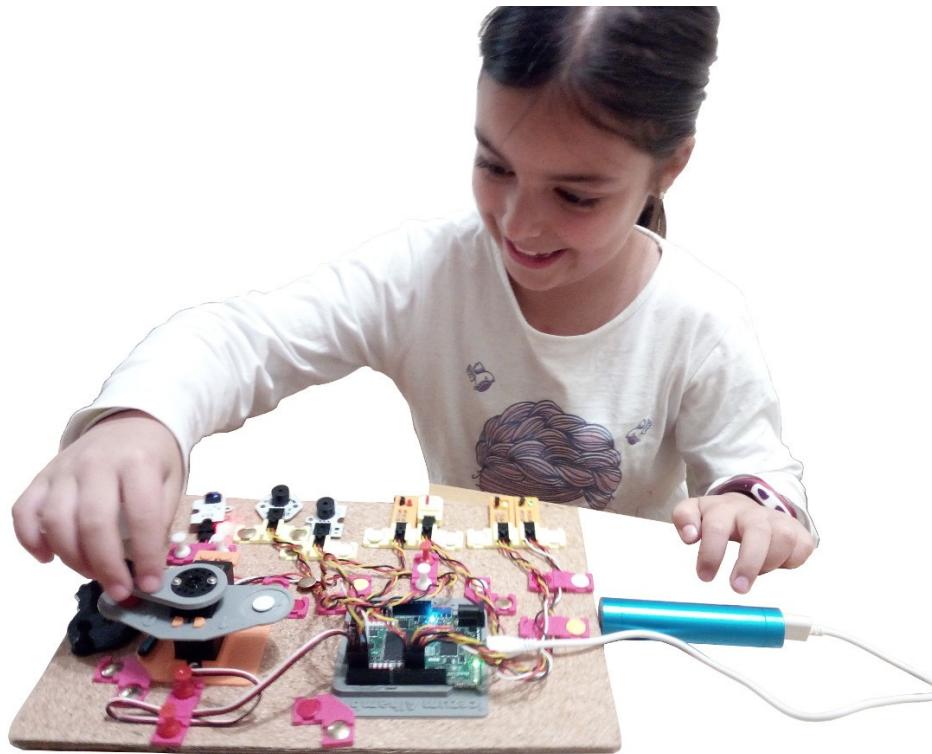




Electrónica digital para todos con FPGAs Libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>

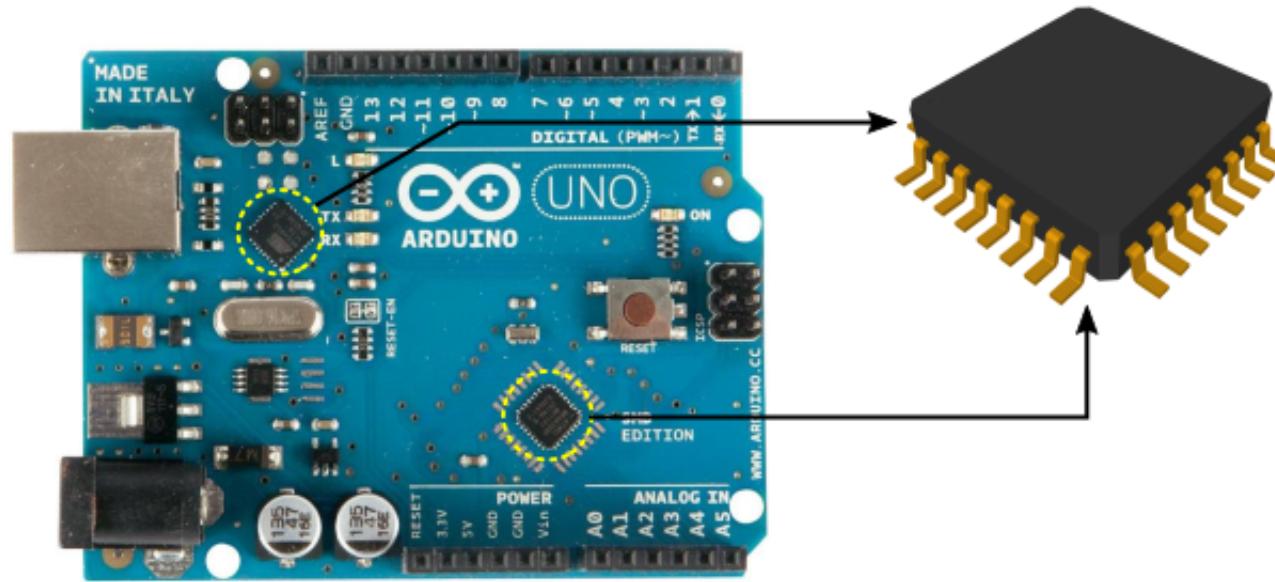
Contenido

Parte I: *Electrónica digital y FPGAs libres*

Parte II: *Demo. Electrónica digital accesible para NO técnicos*

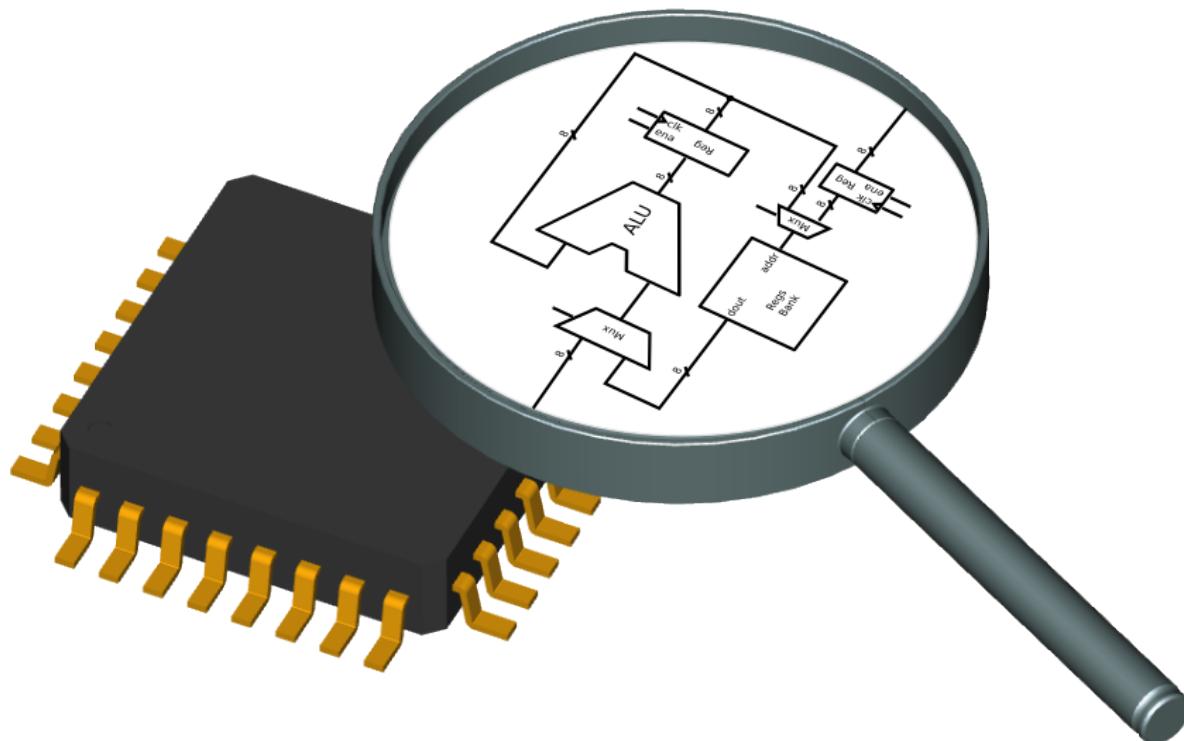
Parte III: *Demo. Electrónica digital accesible. Avanzado*

Chips digitales



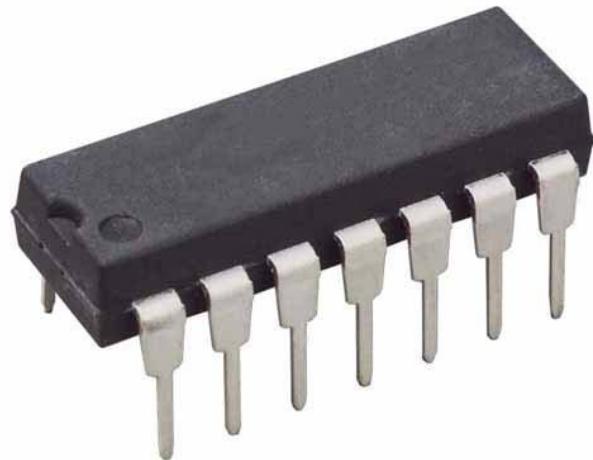
- El gran invento del siglo XX
- Están por todos lados
- Muy baratos
- Los compramos y los usamos

Electrónica digital

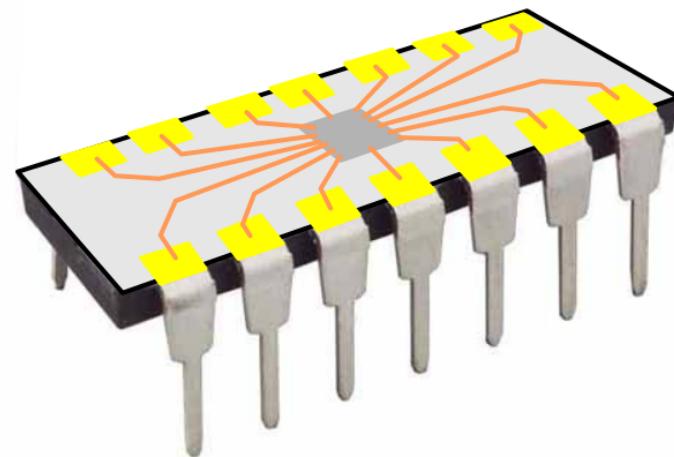


- Nivel de electrónica digital
- Información: Sólo 1s y 0s (Bits)
- Función: **Manipular, almacenar y transportar bits**

Viaje al interior de la electrónica (I)

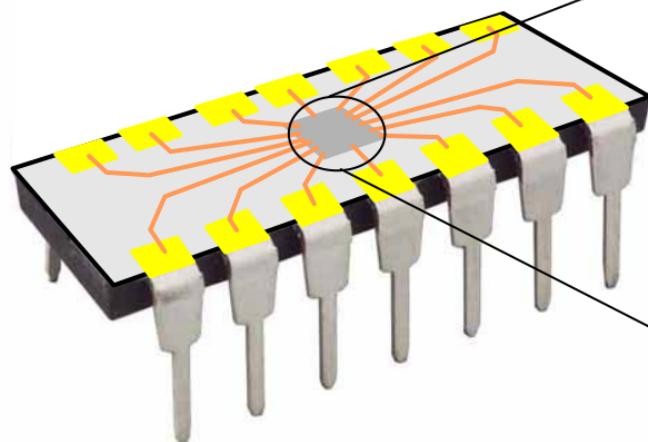


Microchip

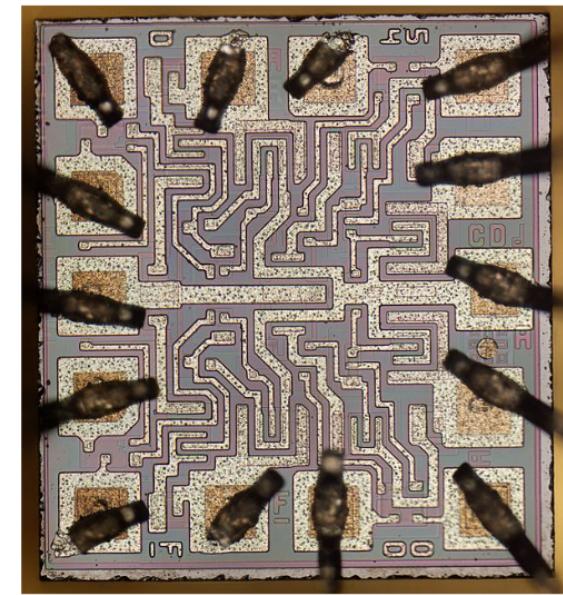


Microchip

Viaje al interior de la electrónica (II)

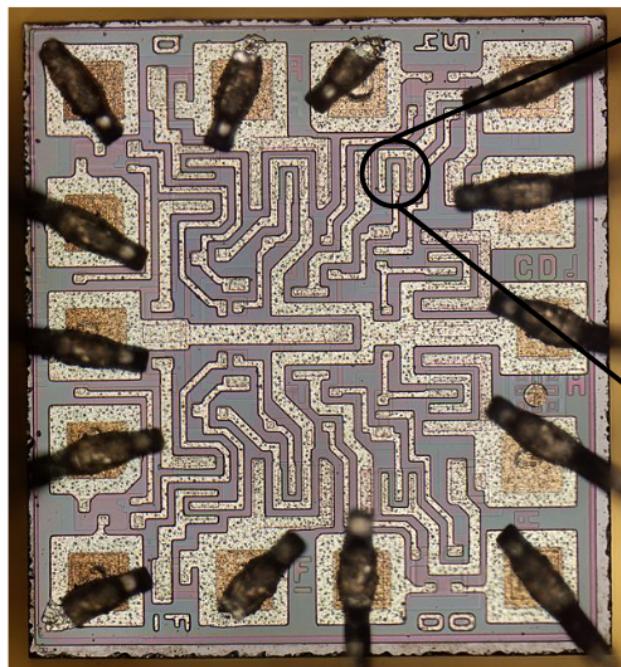


Microchip

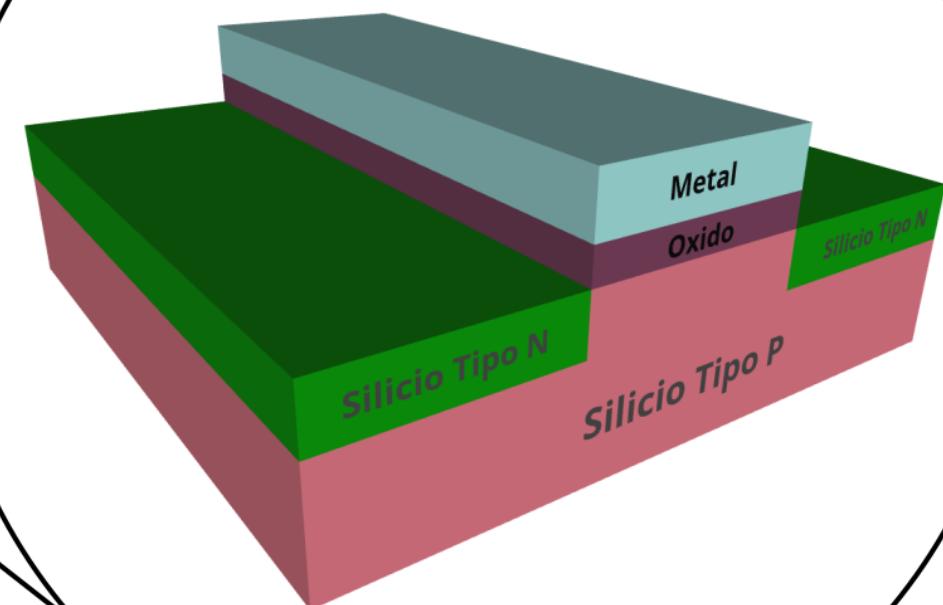


Dado

Viaje al interior de la electrónica (III)

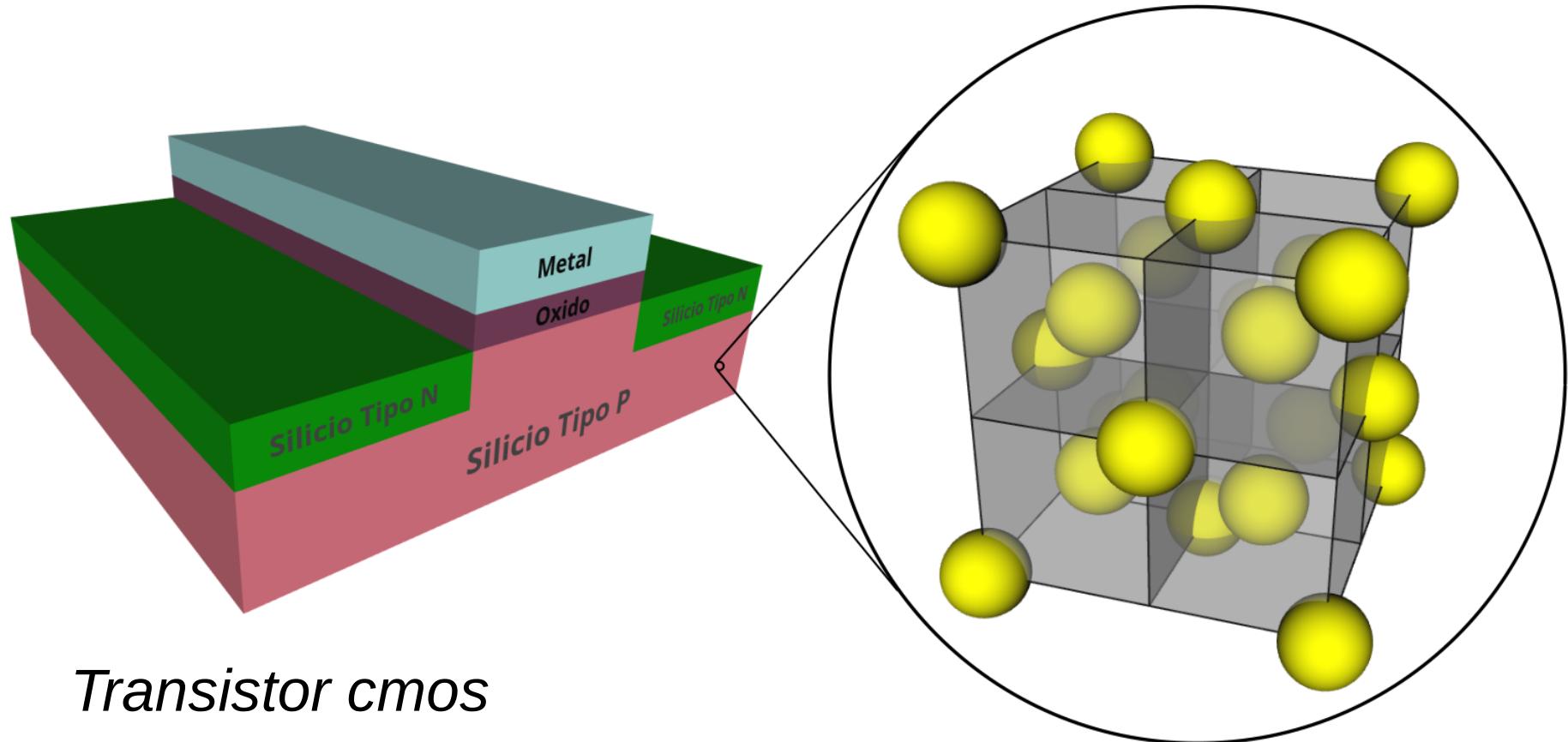


Dado



Transistor cmos

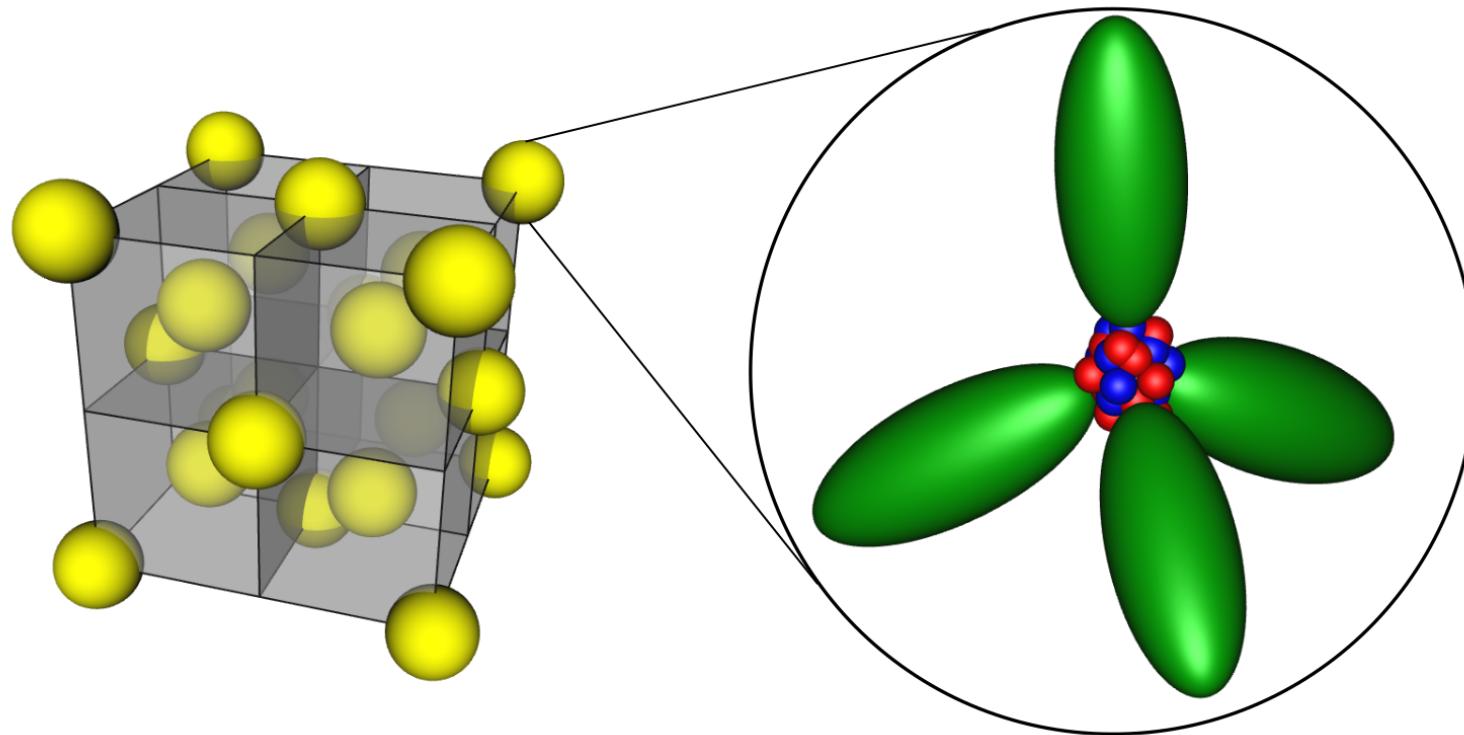
Viaje al interior de la electrónica (IV)



Transistor cmos

Cristal de silicio

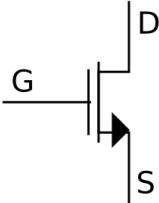
Viaje al interior de la electrónica (V)



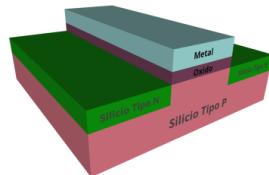
Cristal de silicio

Átomo de Silicio

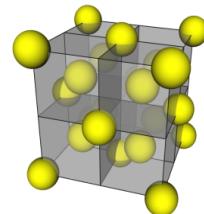
Agrupación en niveles



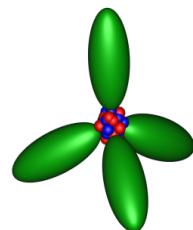
Nivel 4: Transistor



Nivel 3: Semiconductores



Nivel 2: Materiales



Nivel 1: Átomos

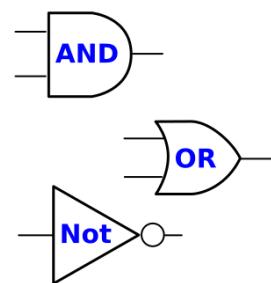
Agrupación en niveles (II)



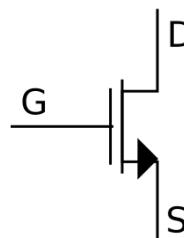
Nivel 7: Software



Nivel 6: Microprocesador

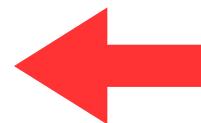
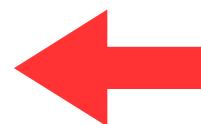
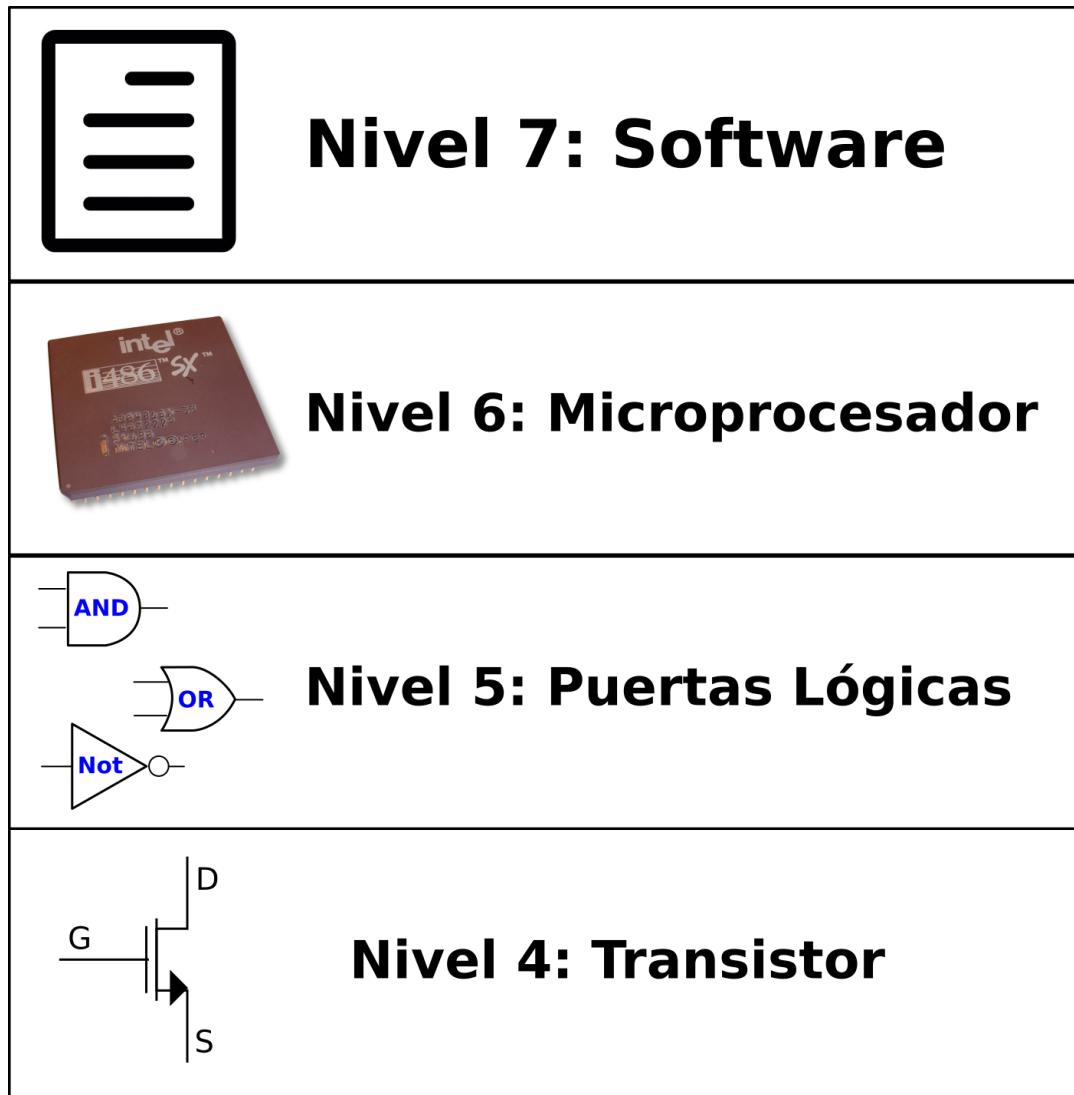


Nivel 5: Puertas Lógicas



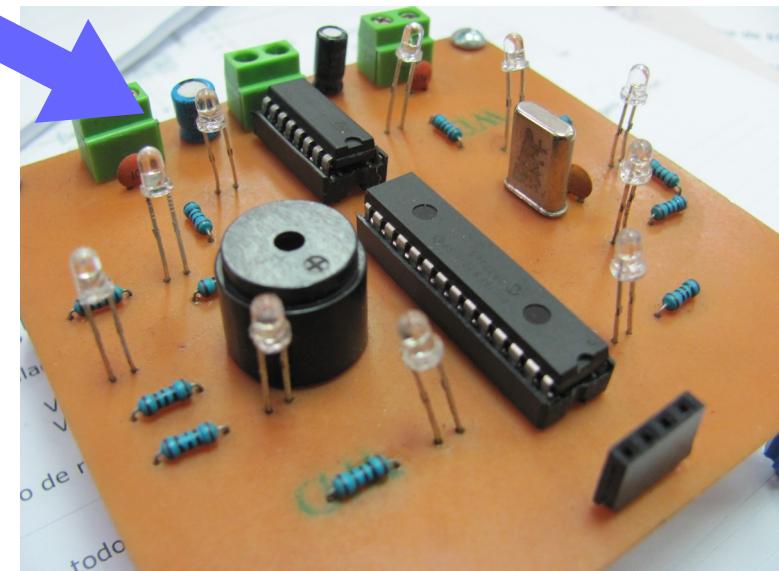
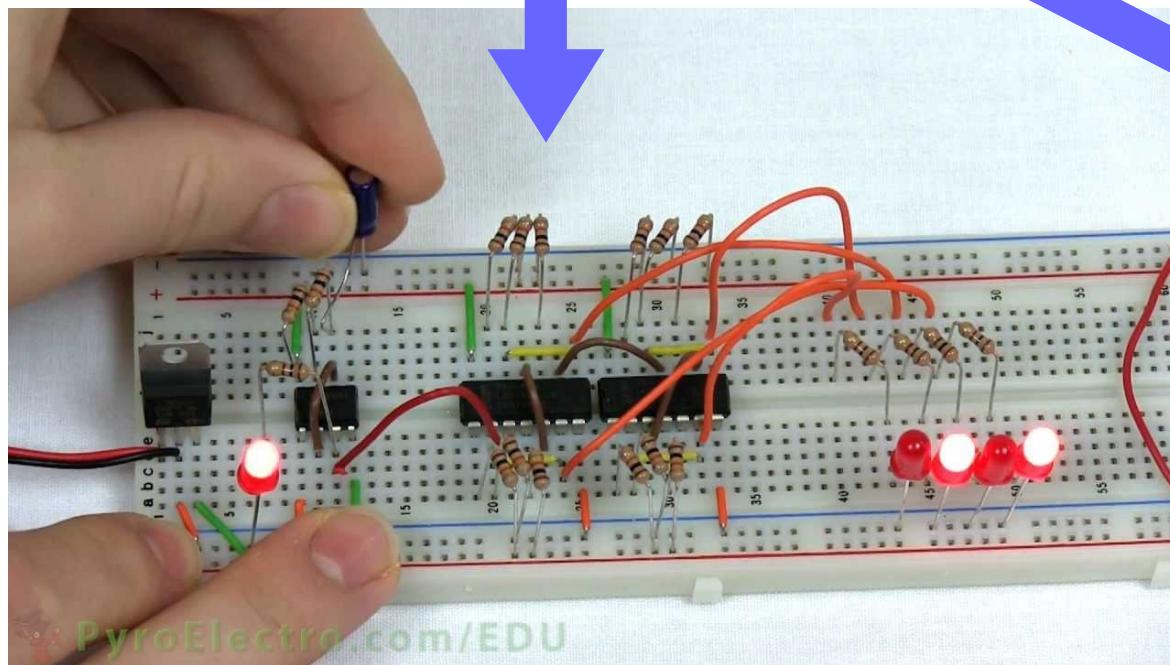
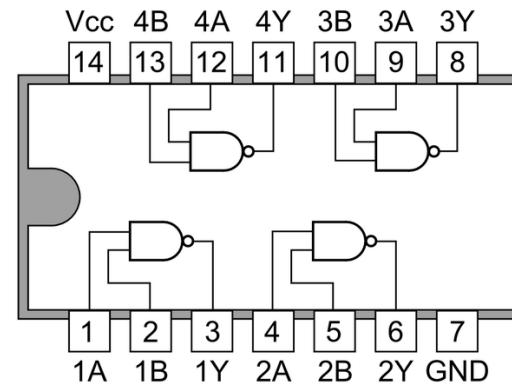
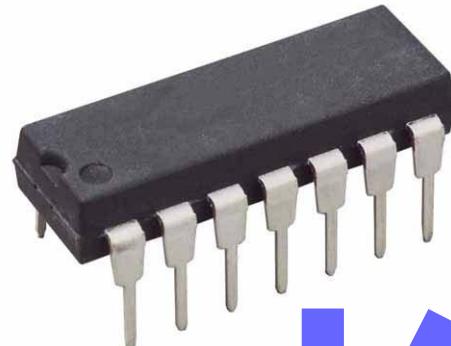
Nivel 4: Transistor

Arduino y Electrónica digital

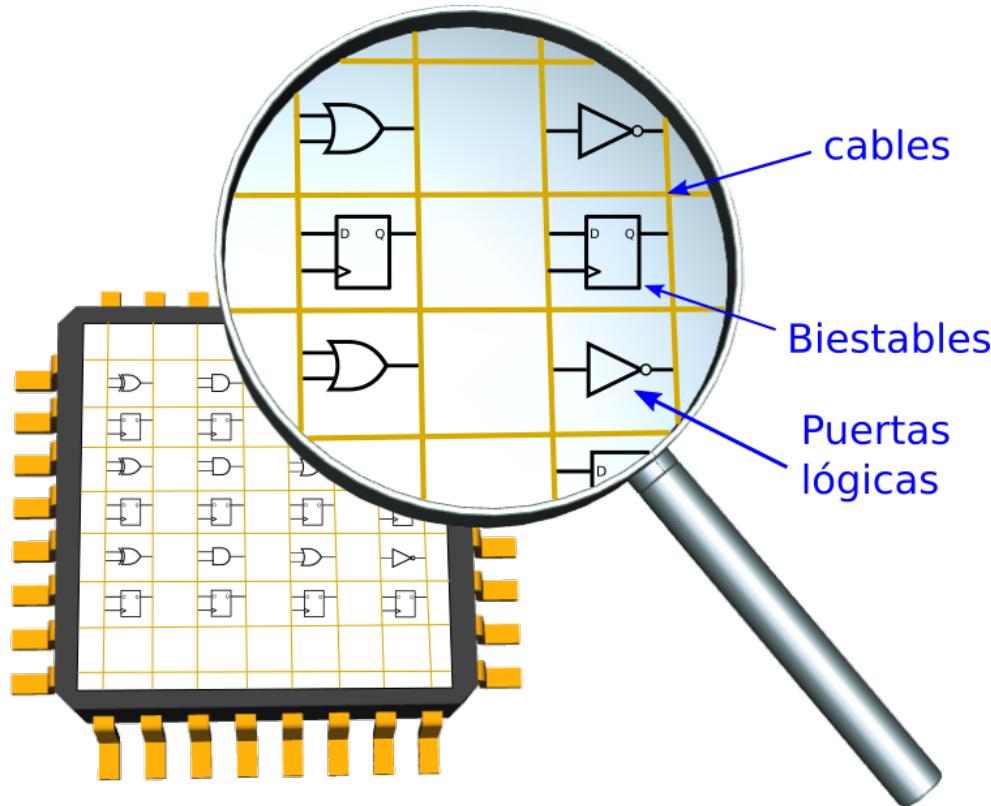


*Electrónica
digital*

¿Cómo se hacen los circuitos digitales?

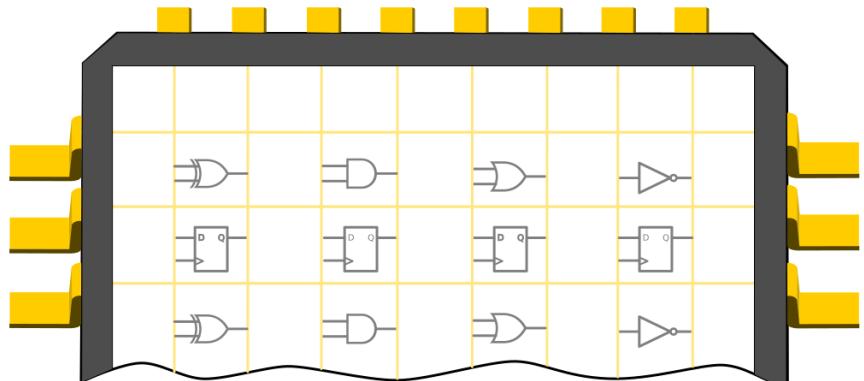


Tecnología FPGA

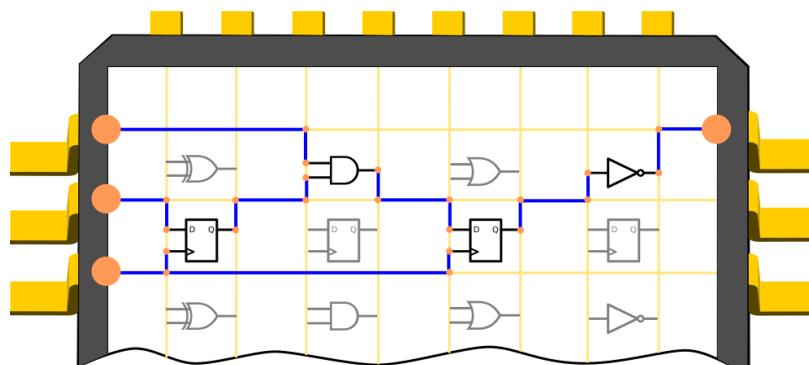


FPGA: Chip “en blanco” que contiene una matriz con los 3 componentes básicos: puertas lógicas, biestables y cables

Electrónica digital con FPGAs



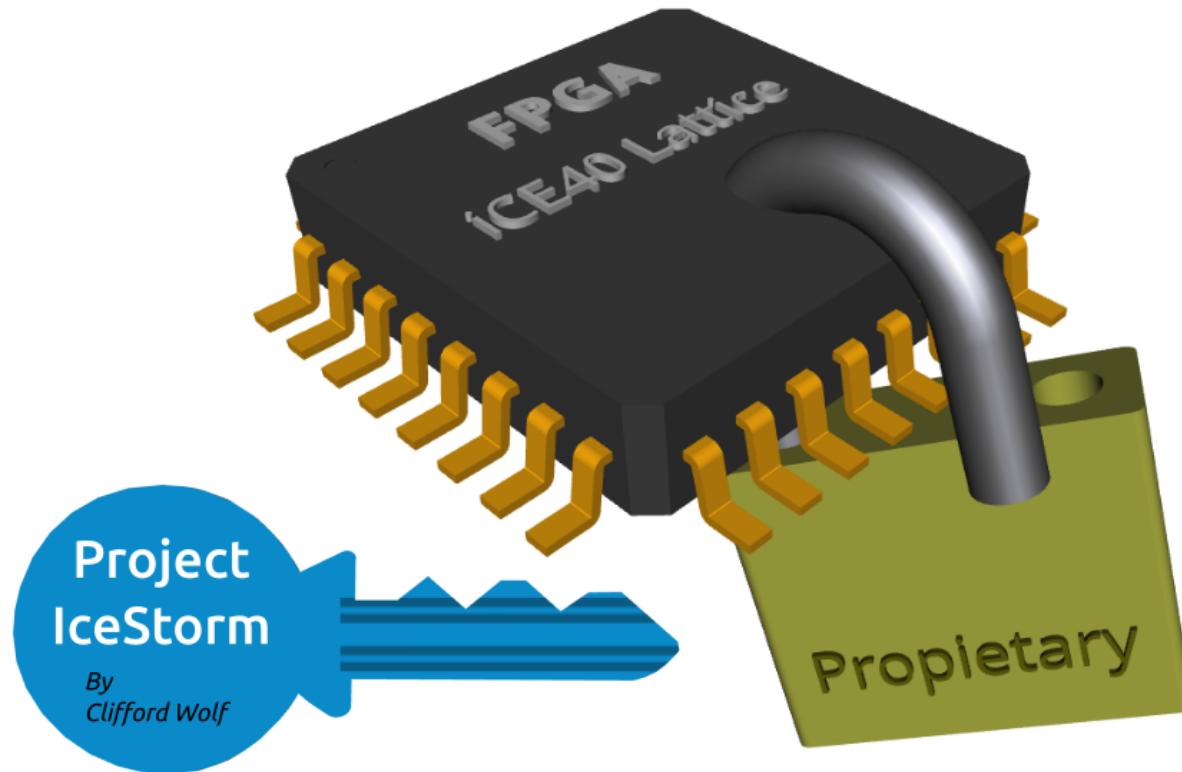
FPGA en Blanco



FPGA configurada

Circuito creado configurando las uniones entre los elementos básicos de la FPGA

FPGAs libres: El renacimiento



- Proyecto Icestorm (Mayo, 2015). **Clifford Wolf**
- La primera *toolchain* que permiten pasar de Verilog al bitstream usando sólo Herramientas libres

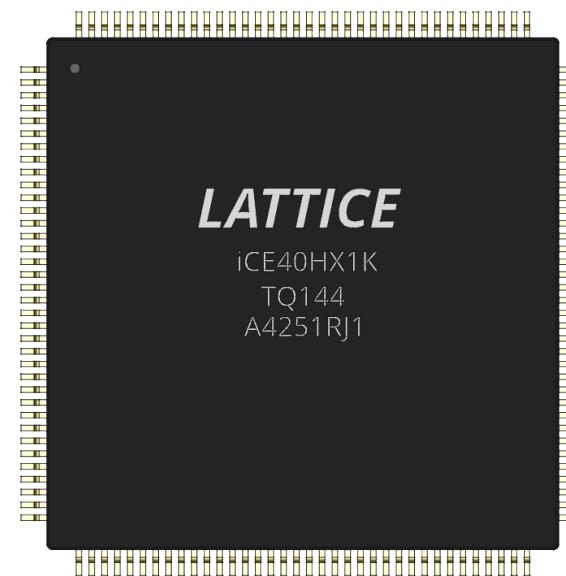
FPGAs libres

- Definición:

Denominamos **FPGAs libres** a aquellas FPGAs que disponen de una **toolchain totalmente libre**

- **FPGAs libres actualmente:**

- Familia **Lattice iCE40**
- Silego Greenpark
- Lenguaje Verilog (Maduro)
- Soporte VHDL (Muy alfa)



Comunidad FPGAwars



- Comunidad para **compartir conocimiento** relacionado con **FPGAs libres**
- Es el **clonewars** de las FPGAs, pero en modesto :-)
- Idioma: Castellano
- 380 miembros
- Cualquier pregunta / comentario / sugerencia → Correo a la lista :-)

<http://fpgawars.github.io/>

Contenido

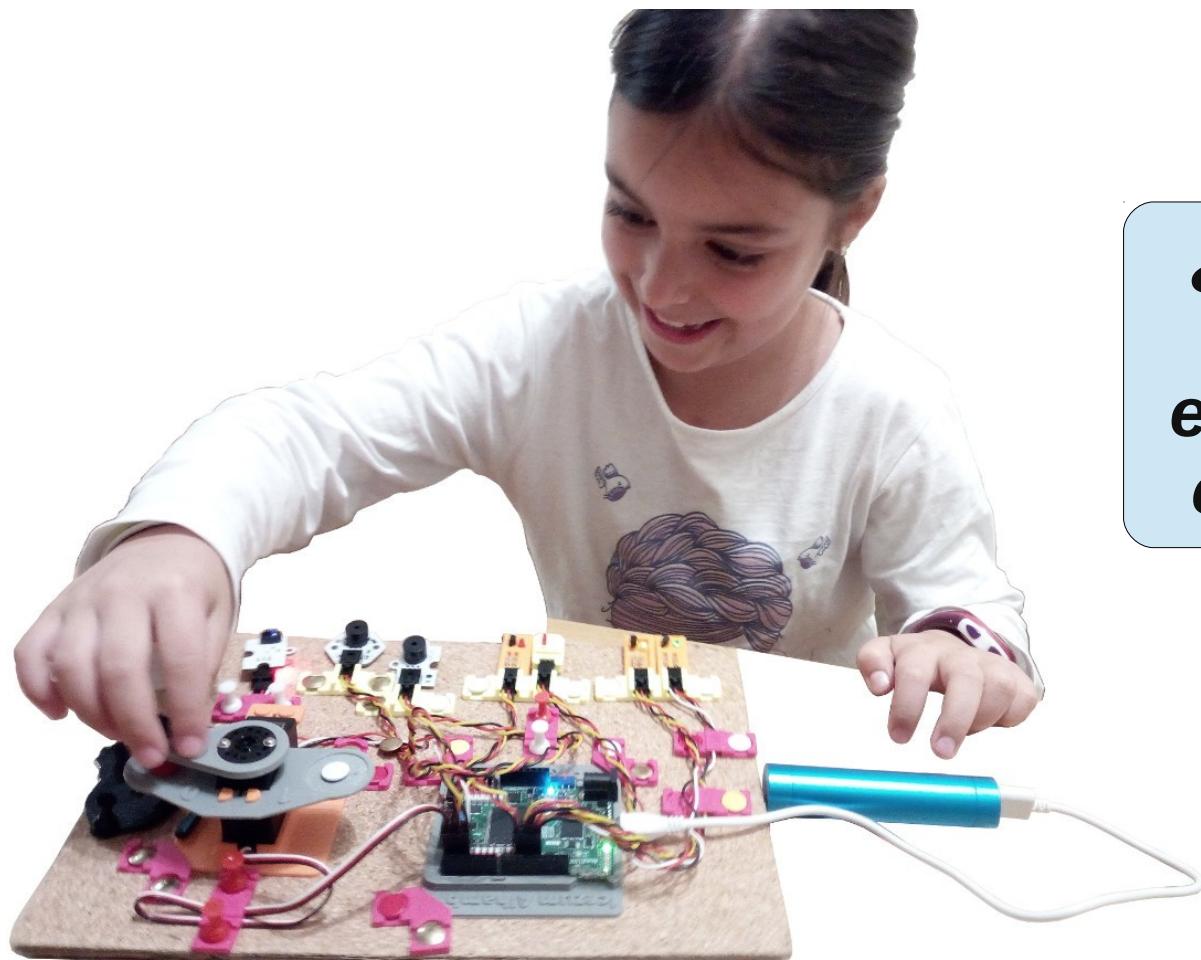
Parte I: Electrónica digital y FPGAs libres

Parte II: Demo
Electrónica digital accesible para NO técnicos

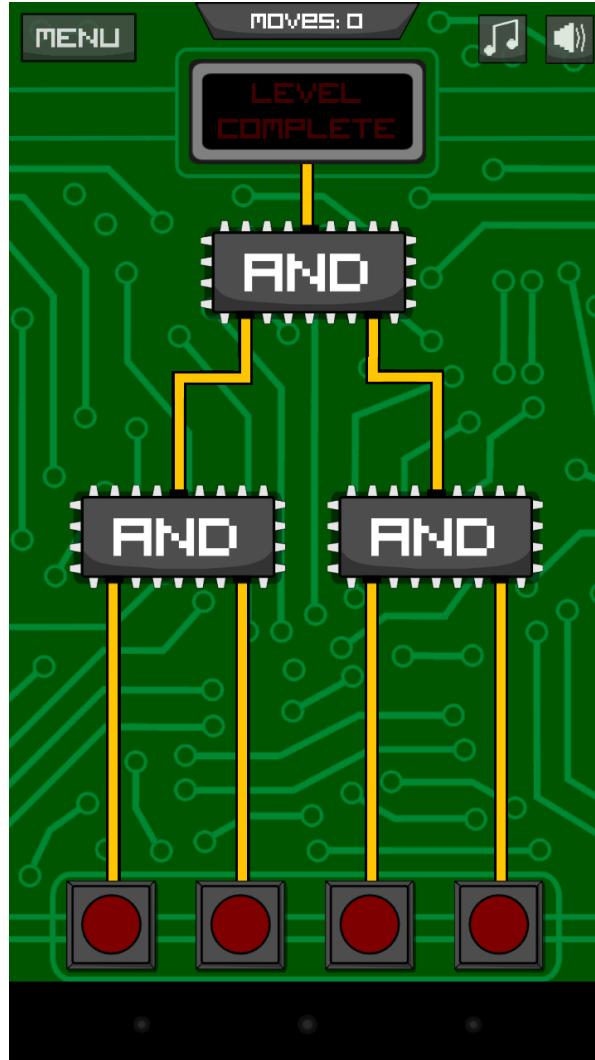
Parte III: Demo. Electrónica digital accesible.
Avanzado

Motivación

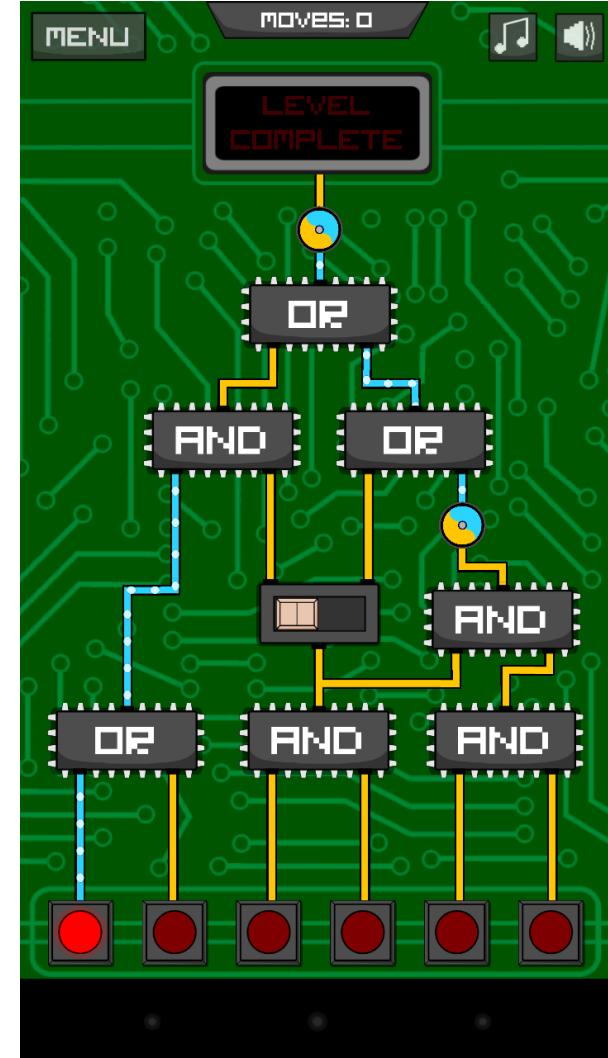
Electrónica digital accesible



*¿Cómo podrían los
niños y los no
electrónicos diseñar
circuitos digitales?*



APP:
Circuit
Scramble

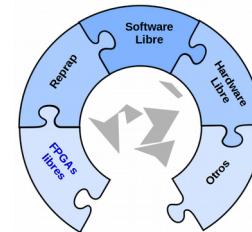
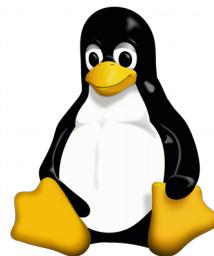
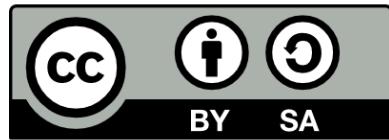


La electrónica digital es intuitiva y...
¡Divertida!

You're leaving the Privative sector...

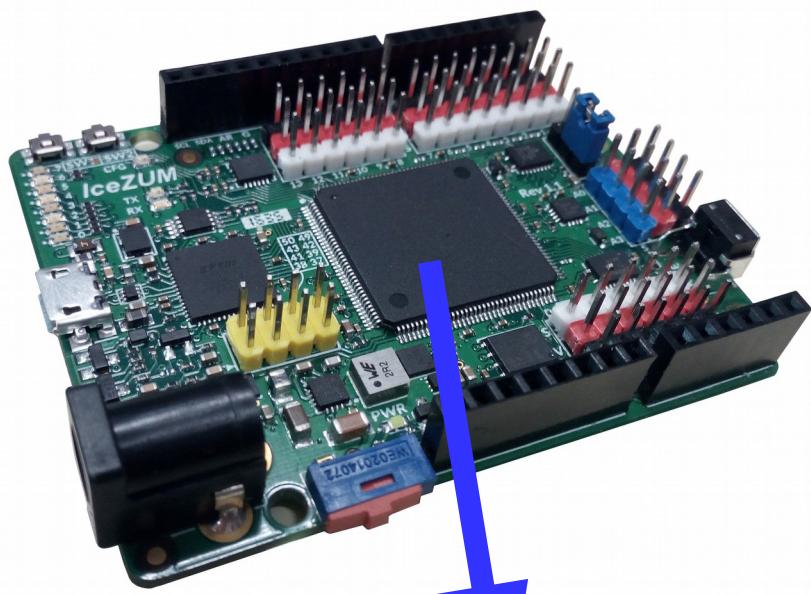


A partir de aquí: Sólo tecnologías libres



Icezum Alhambra v1.1

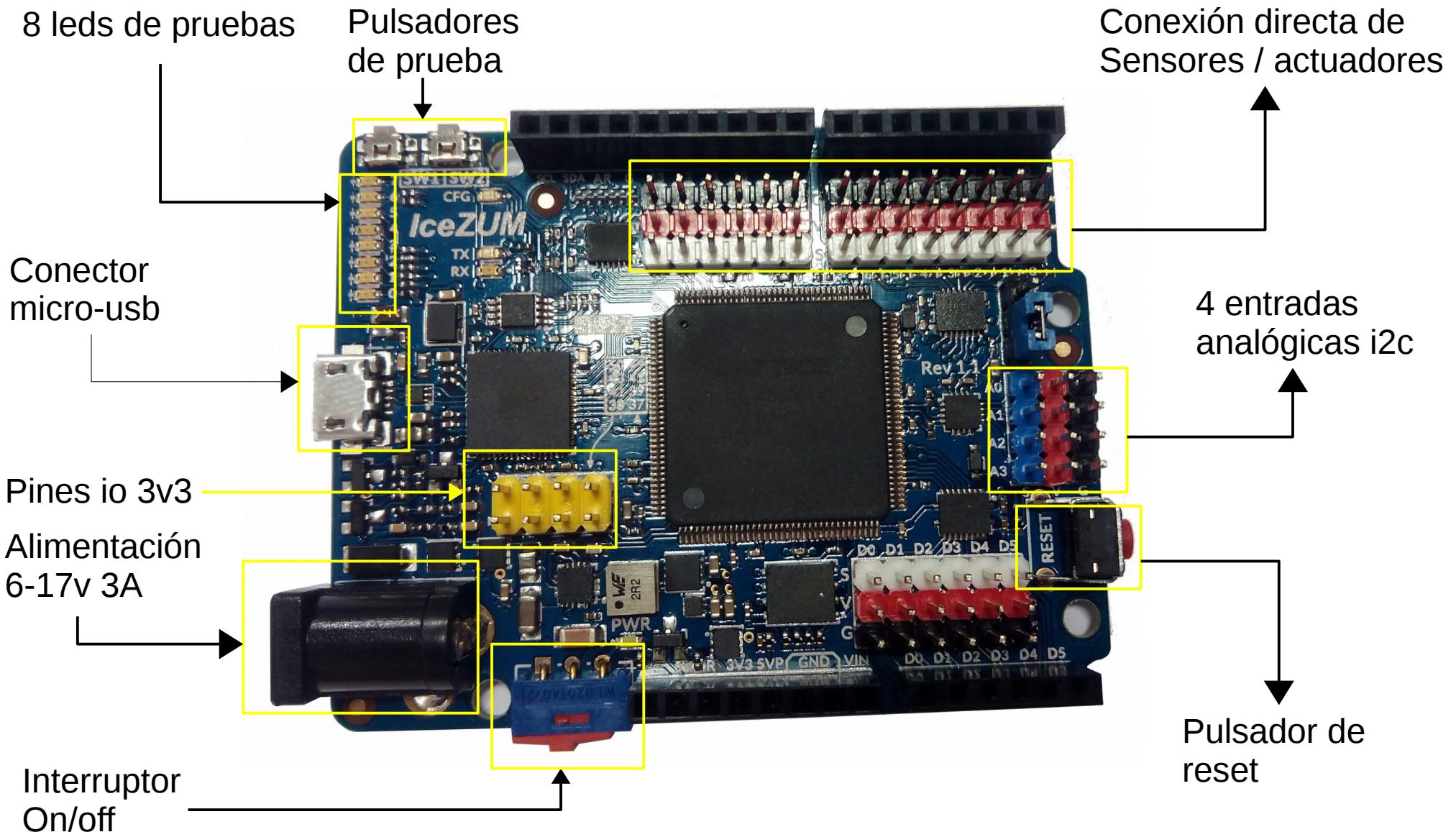
<https://github.com/FPGAwars/icezum/wiki>



FPGA Libre

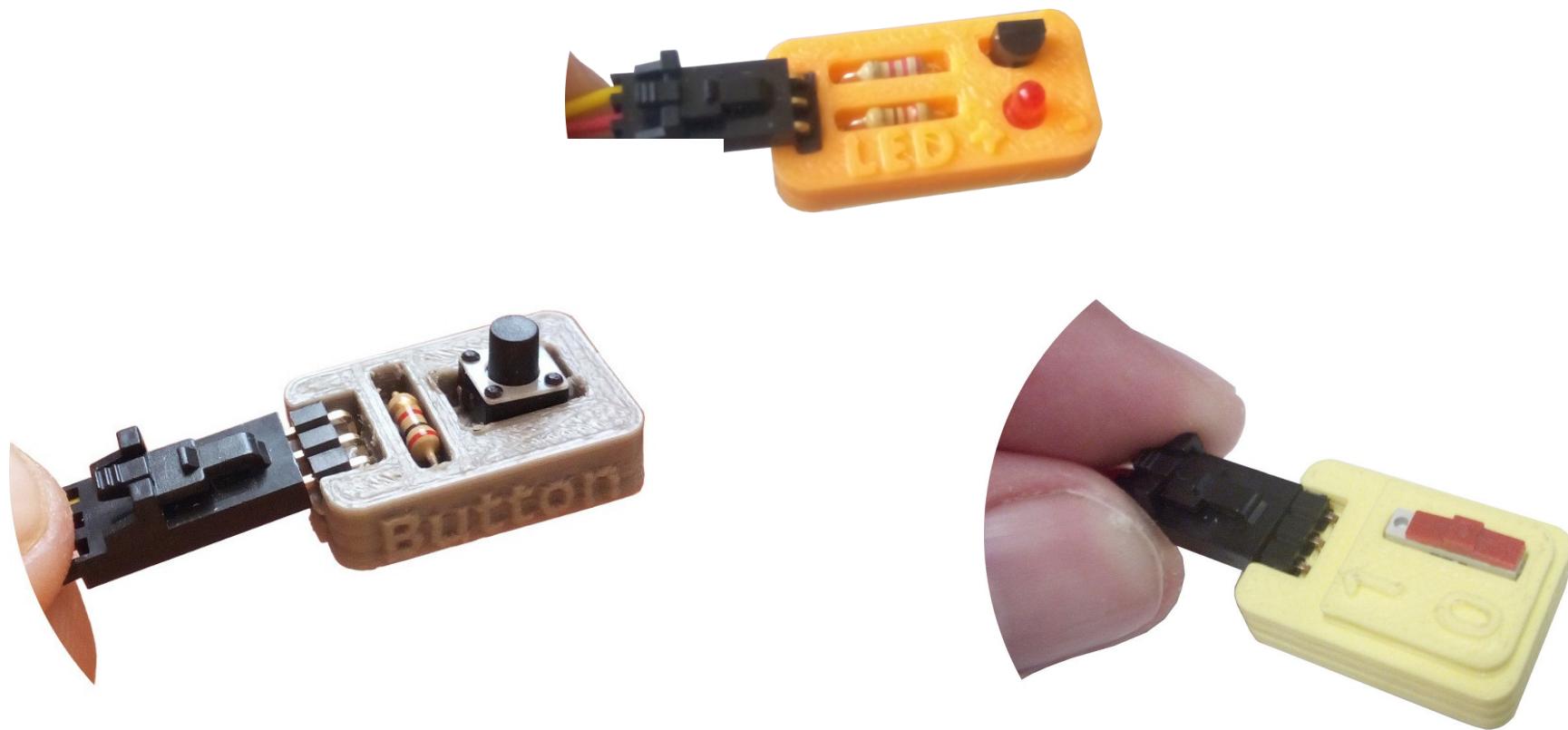
- Autor: **Eladio Delgado**
- Diseñada en Pinos del Valle (Granada)
- Arduino de las **FPGAs**
- Compatible Arduino
- Fácil conexión de circuitos externos/sensores/servos
- Reutilización de los shields de arduino
- 20 entradas/salidas de 5v
- 3A corriente de entrada
- Perfecta para hacer robots

Icezum Alhambra v1.1



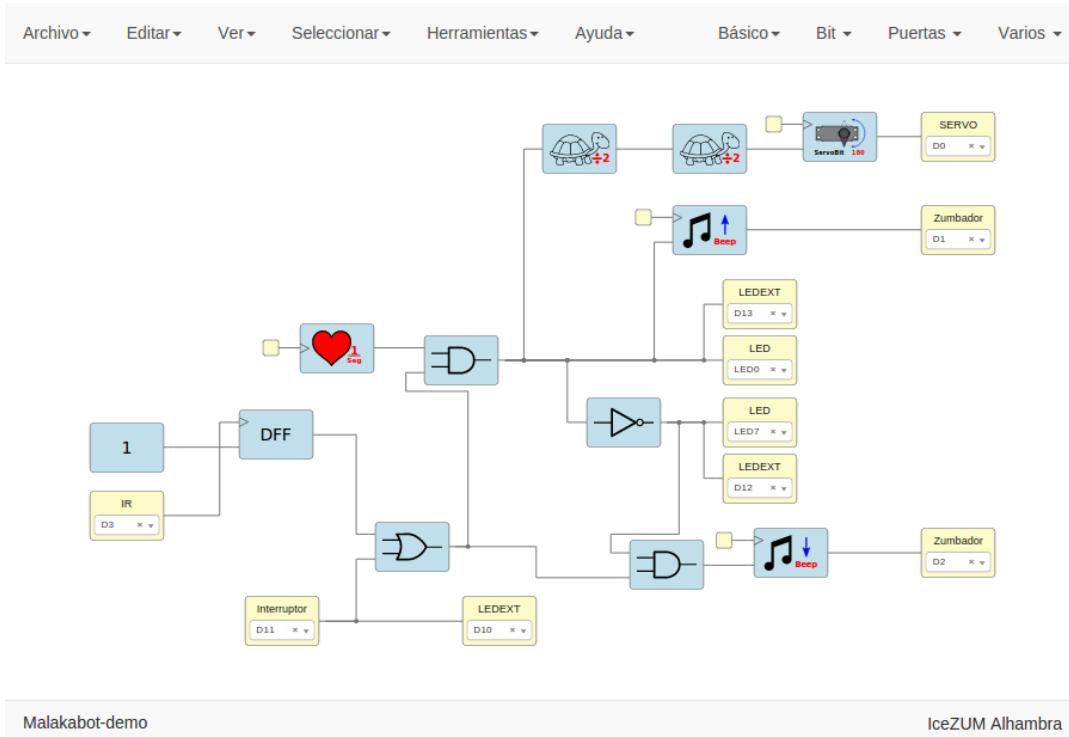
Periféricos

PCBprints: Mini-circuitos impresos en 3D





icesstudio



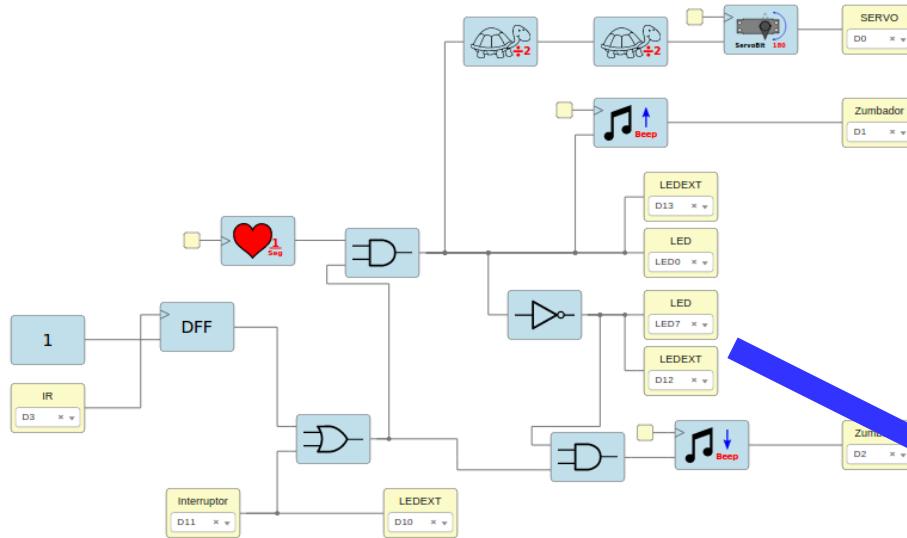
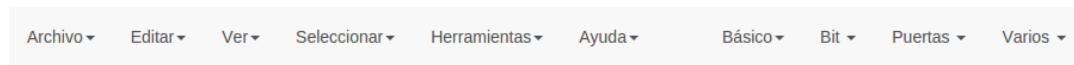
Malakabot-demo

IceZUM Alhambra

<https://github.com/FPGAwars/icesstudio>

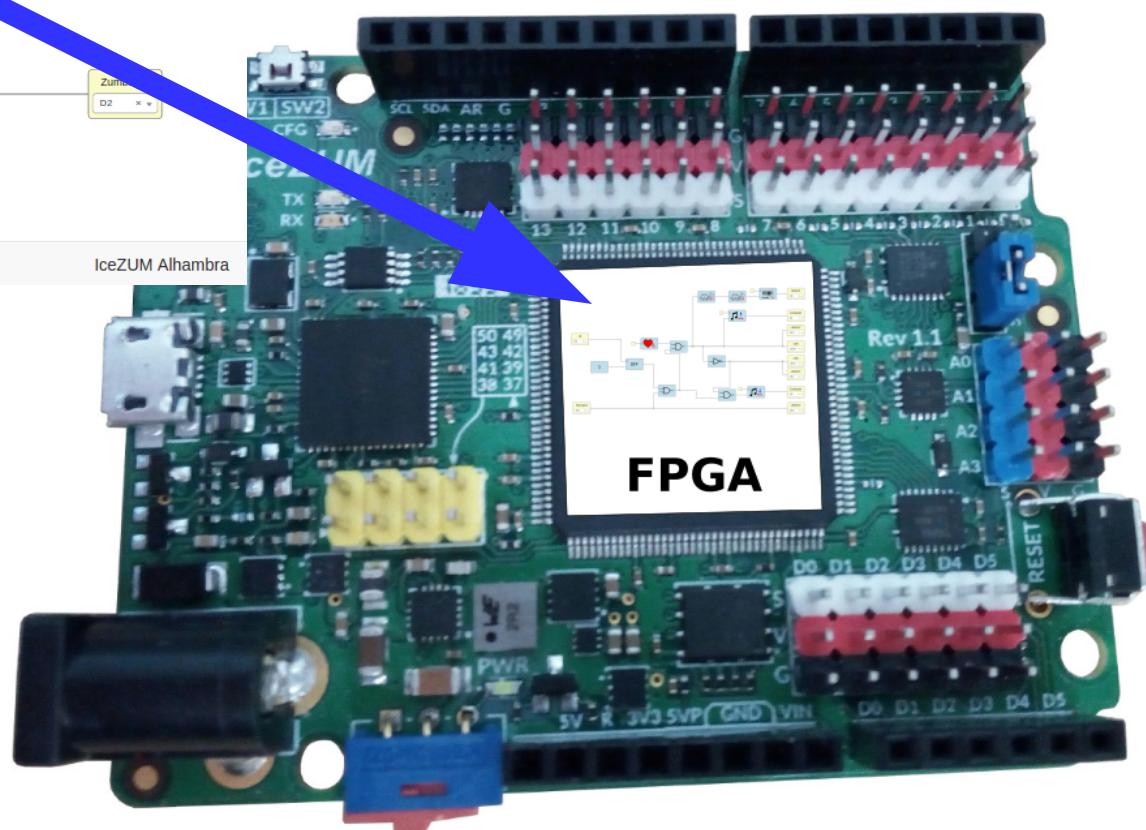
- Autor: **Jesús Arroyo**
- Electrónica digital para todos
- Herramienta visual
- Traduce a verilog

La magia de las FPGAs

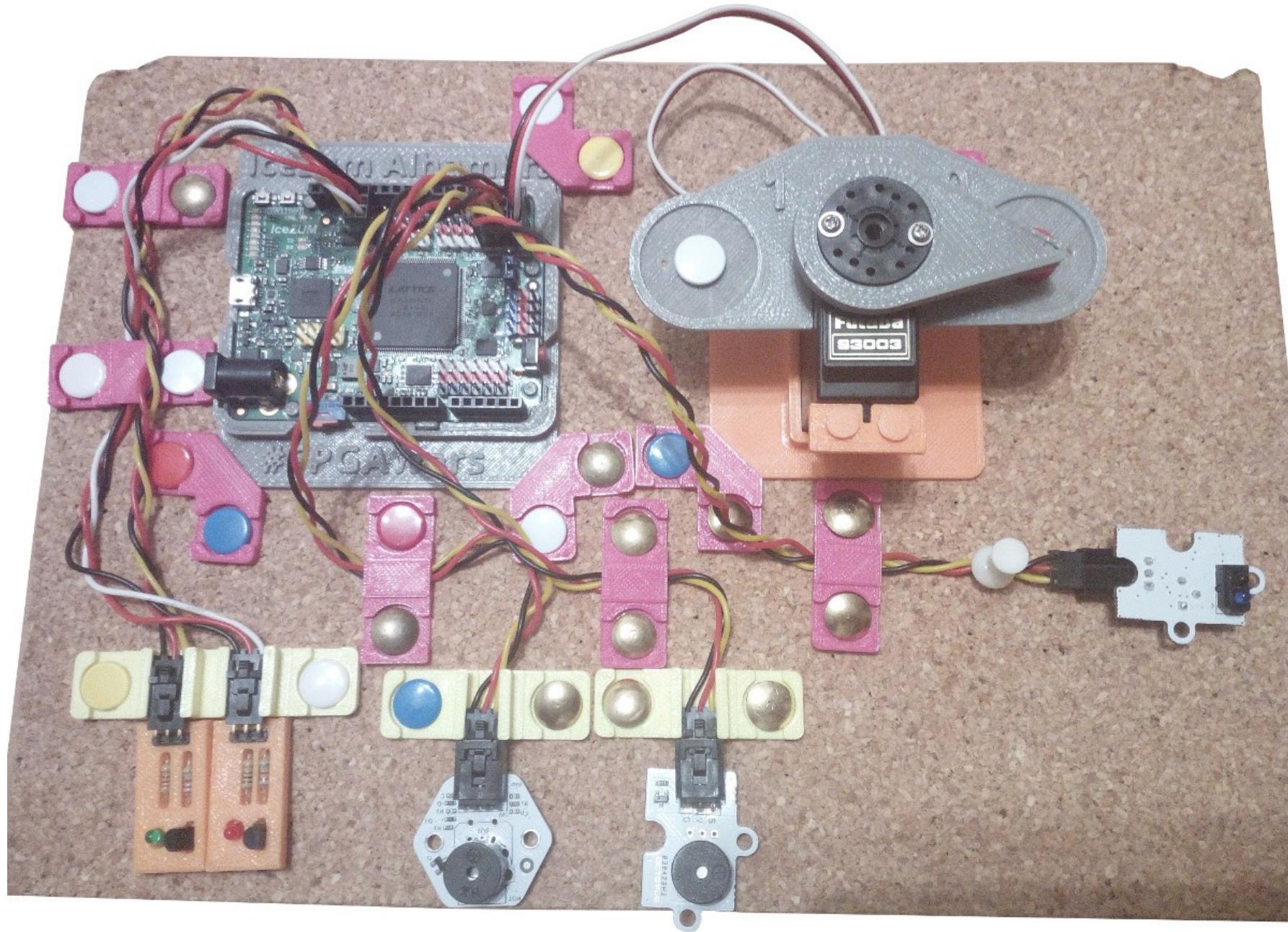


Malakabot-demo

IceZUM Alhambra



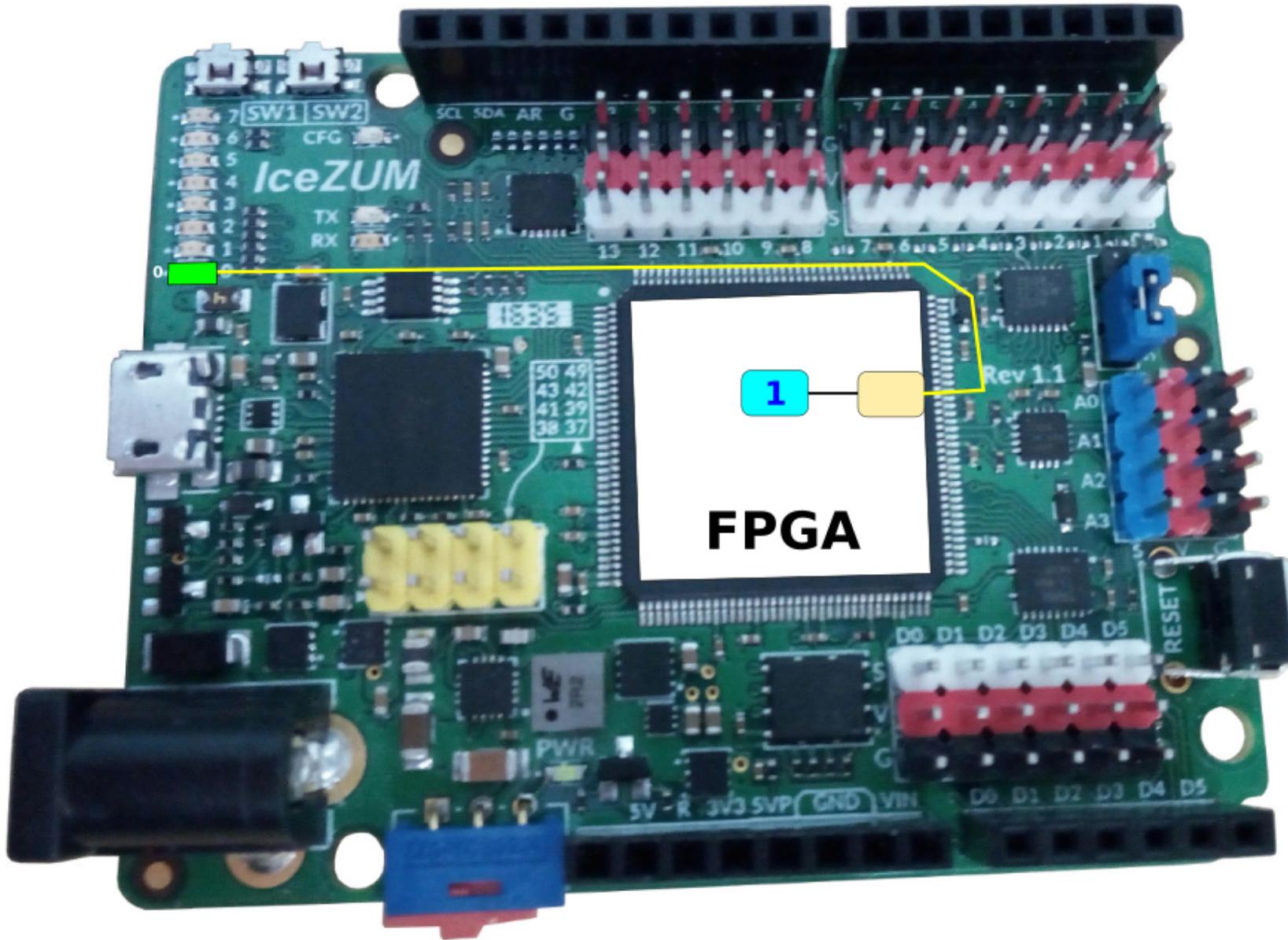
¡Empezamos!



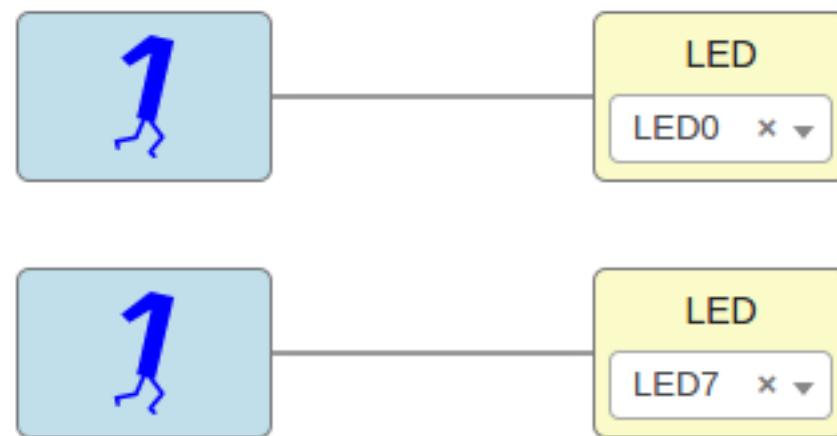
Ejemplo 1: Hola Mundo



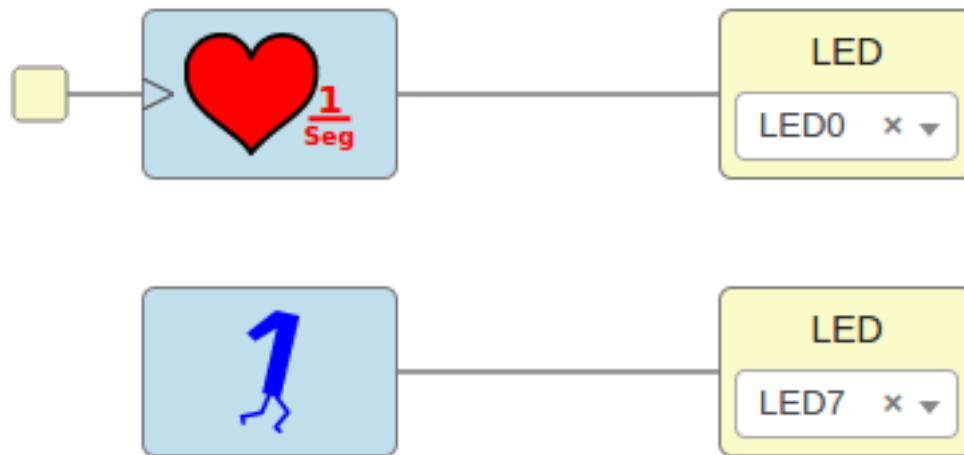
Hola mundo: Implementación física



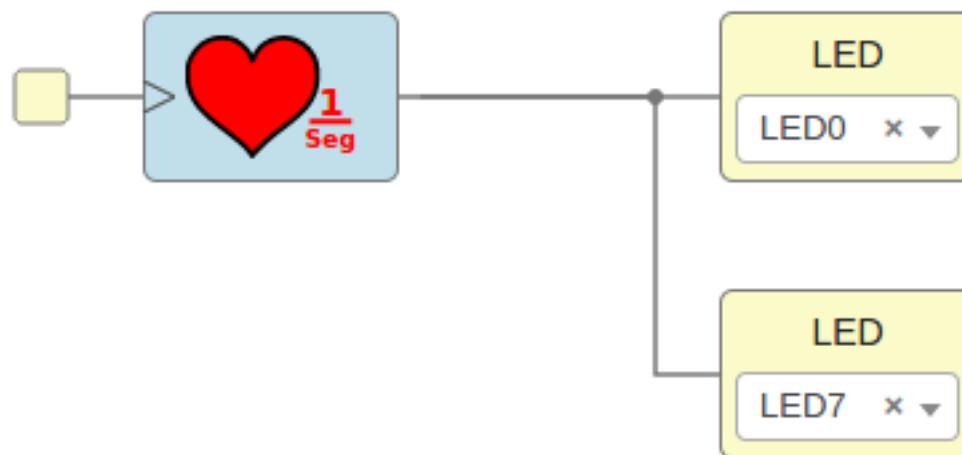
Ejemplo 2: Dos leds en paralelo



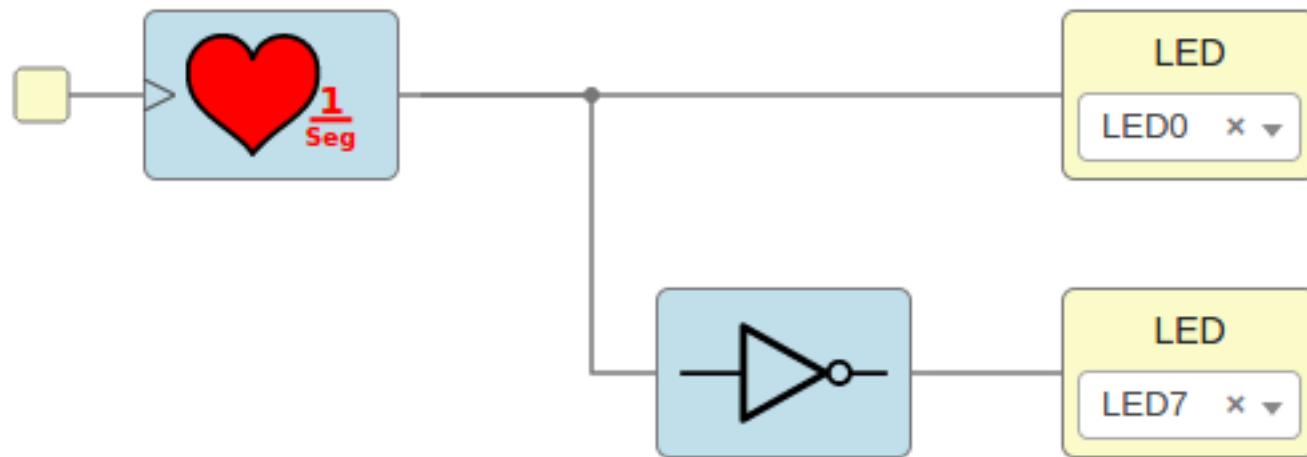
Ejemplo 3: Led pulsante



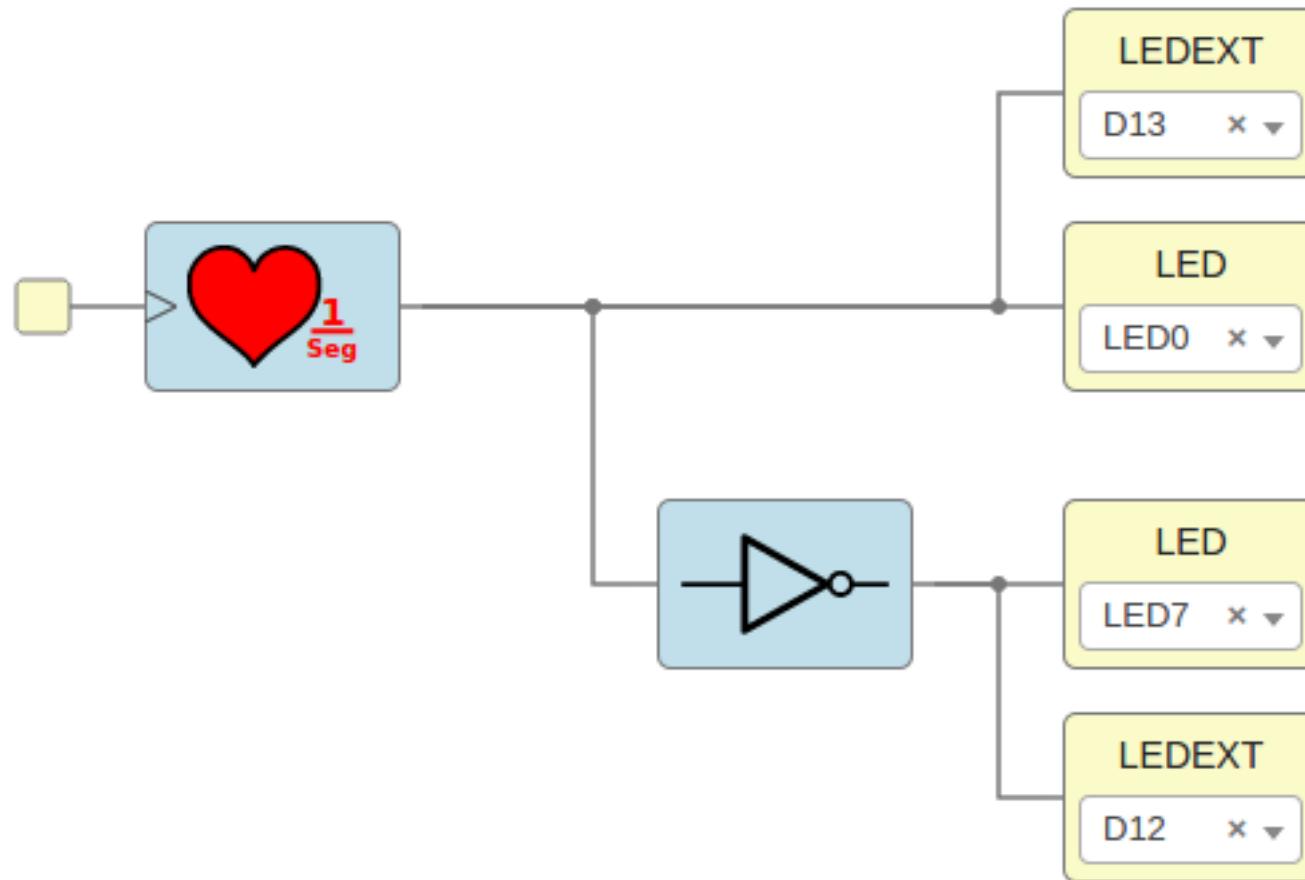
Ejemplo 4: Leds pulsantes Mismo ritmo



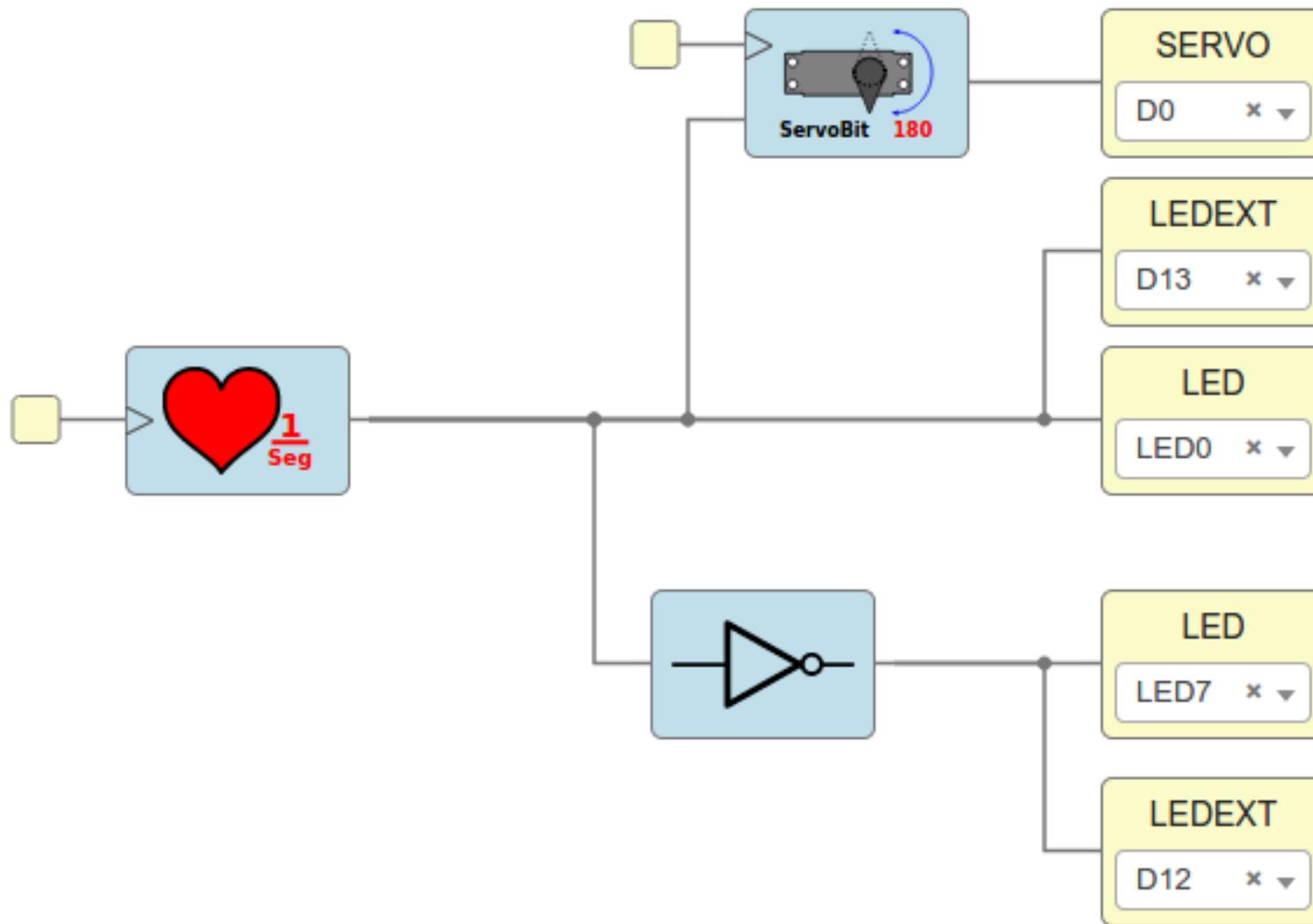
Ejemplo 5: Leds alternativos



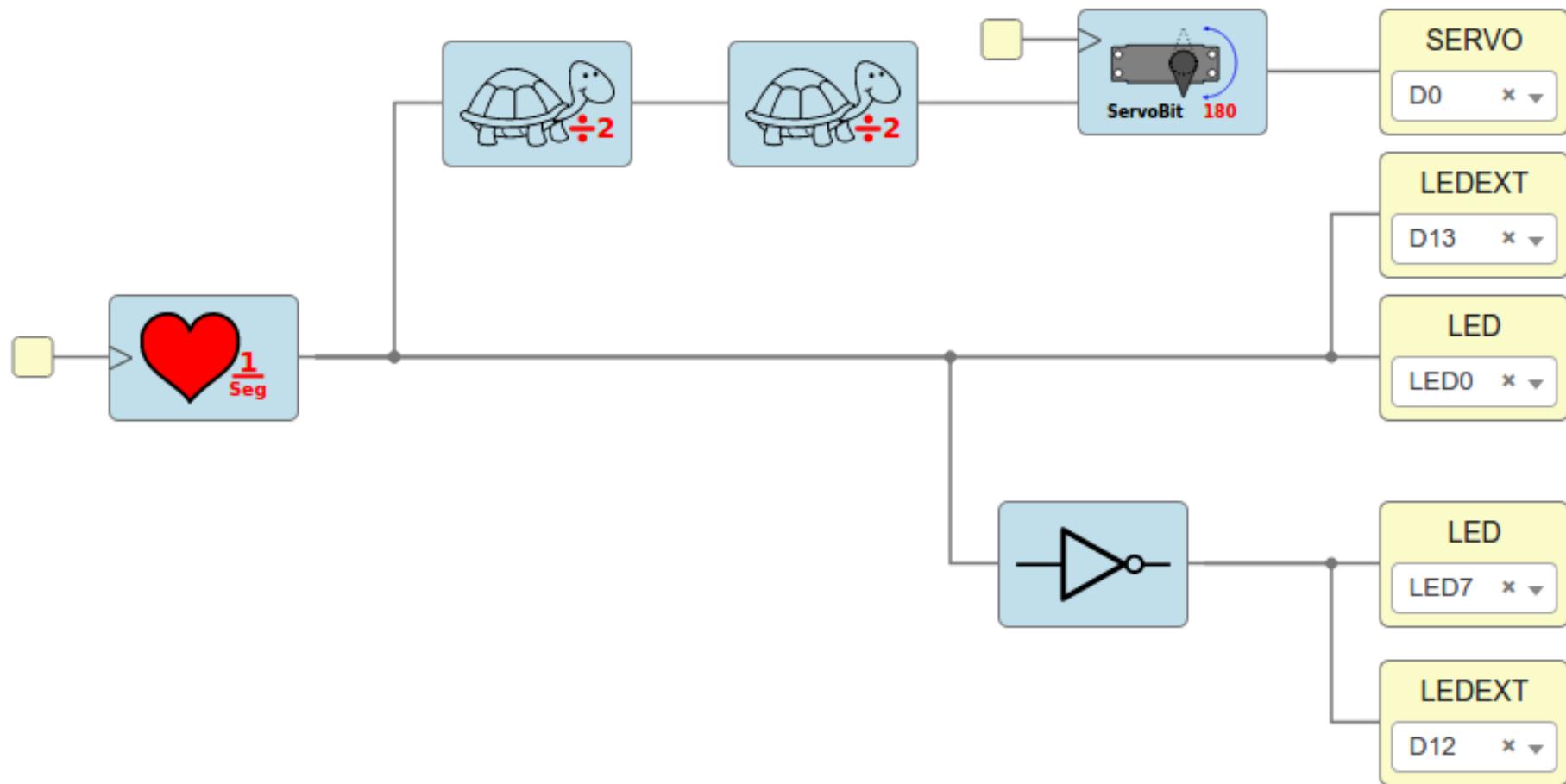
Ejemplo 6: Leds externos



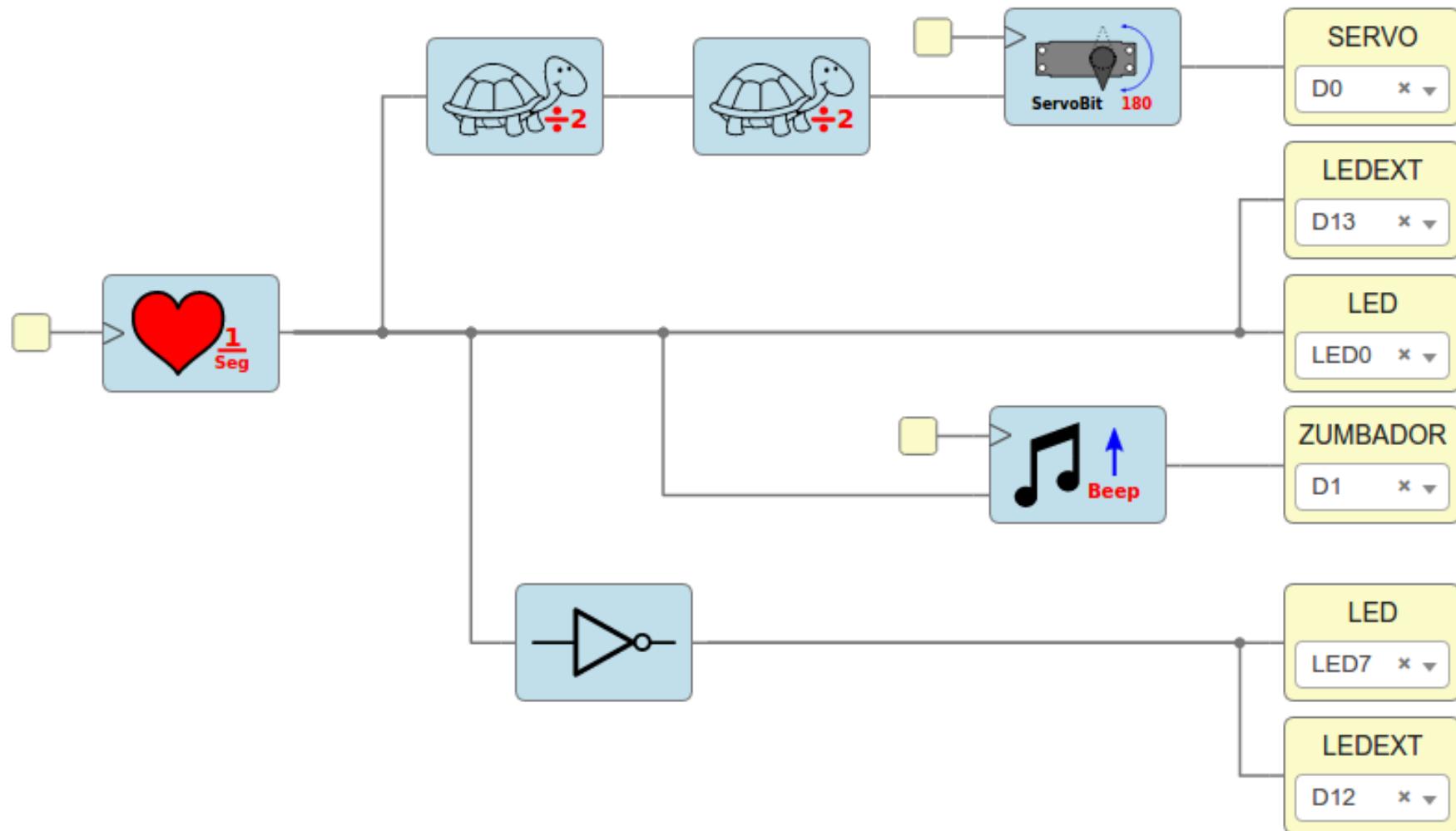
Ejemplo 7: Servo binario



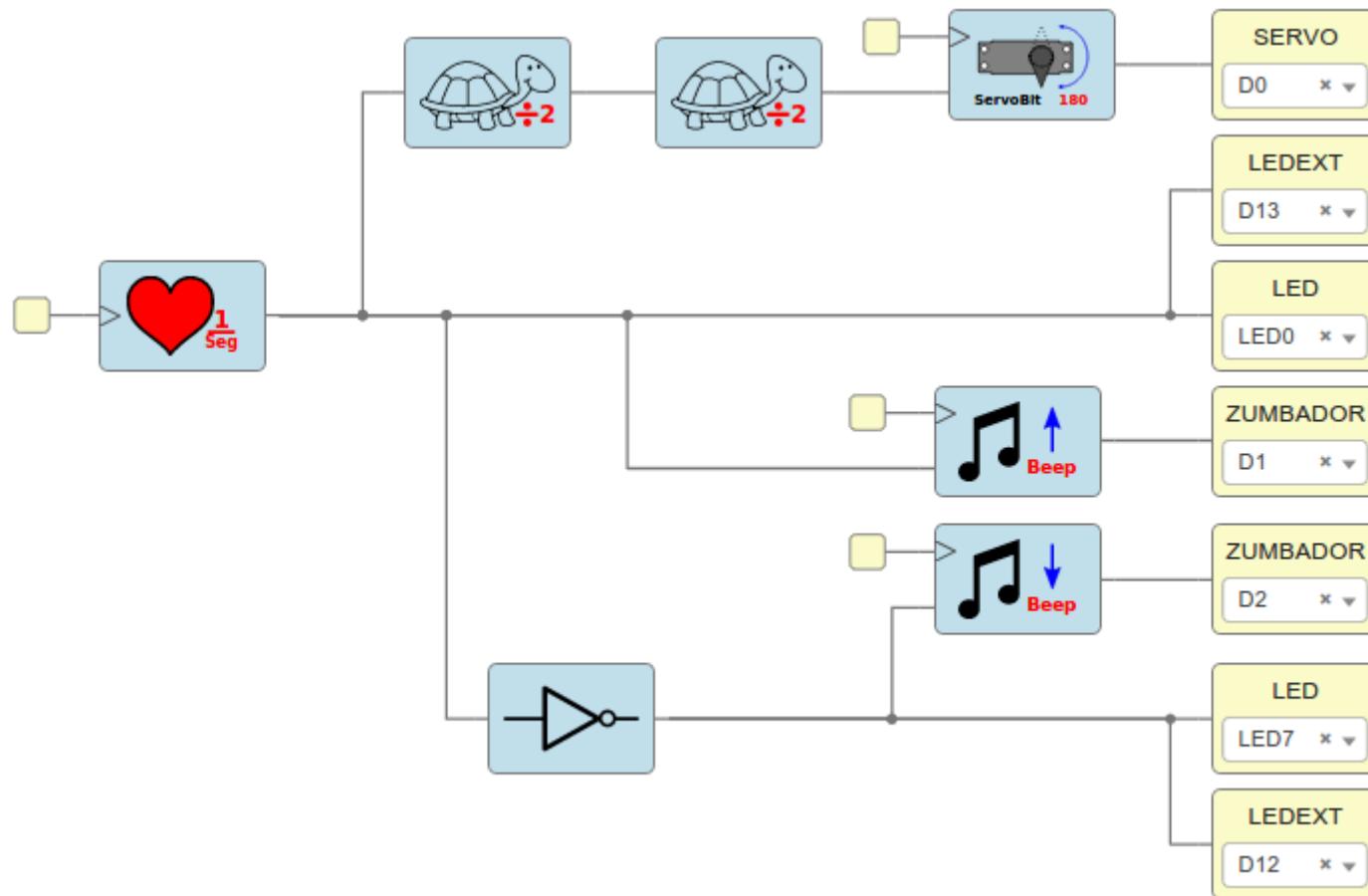
Ejemplo 8: Bajando el ritmo



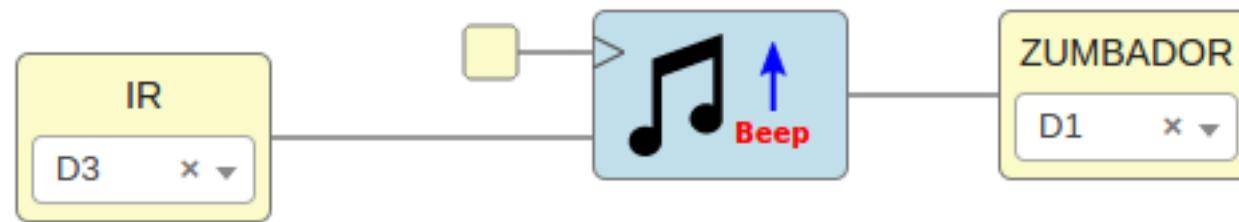
Ejemplo 9: Zumbador



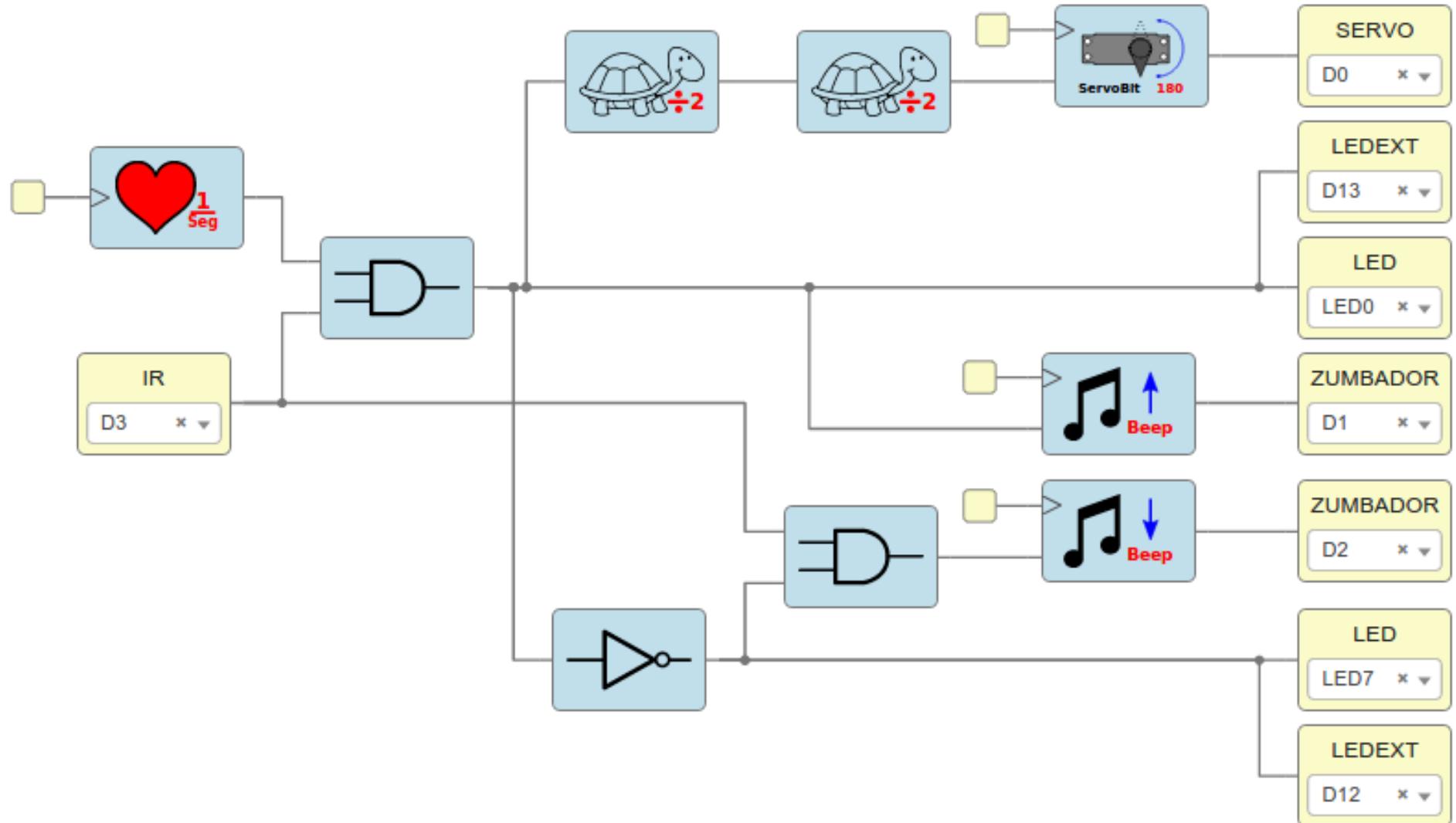
Ejemplo 10: Sirena



Ejemplo 11: Test IR



Ejemplo 12: Alarma v1.0



Contenido

Parte I: Electrónica digital y FPGAs libres

Parte II: Demo
Electrónica digital accesible para NO técnicos

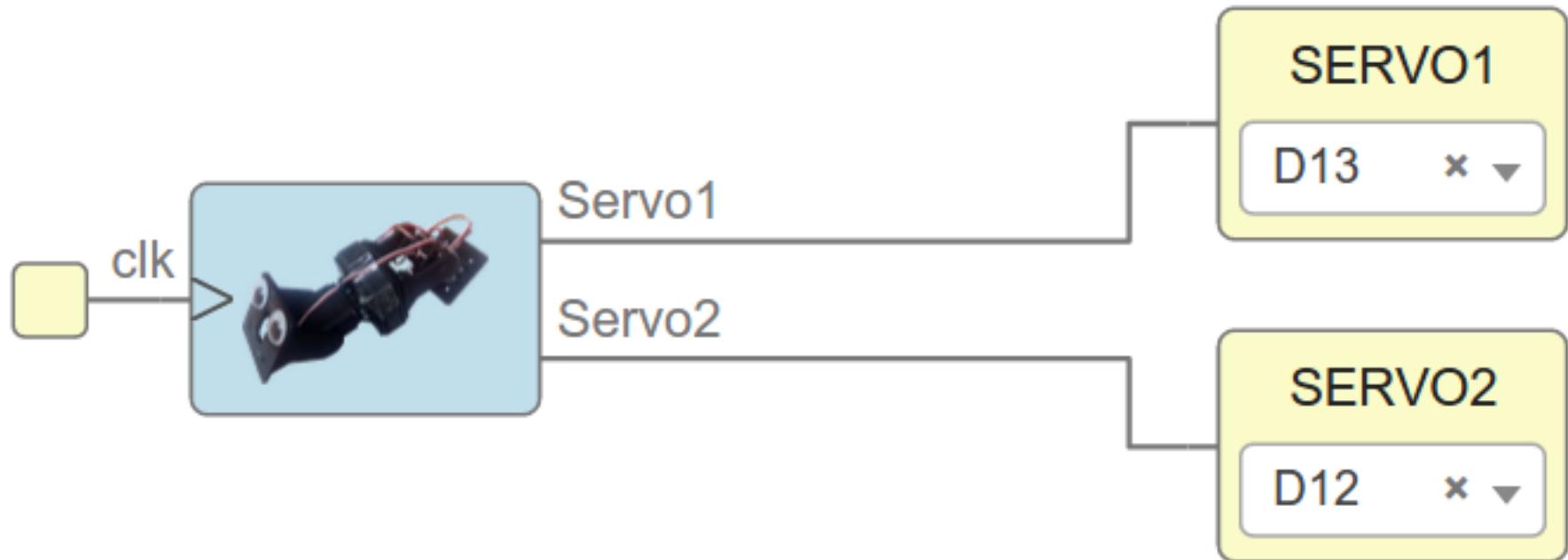
Parte III: Demo
Electrónica digital accesible. Avanzado

Larby: Robot modular

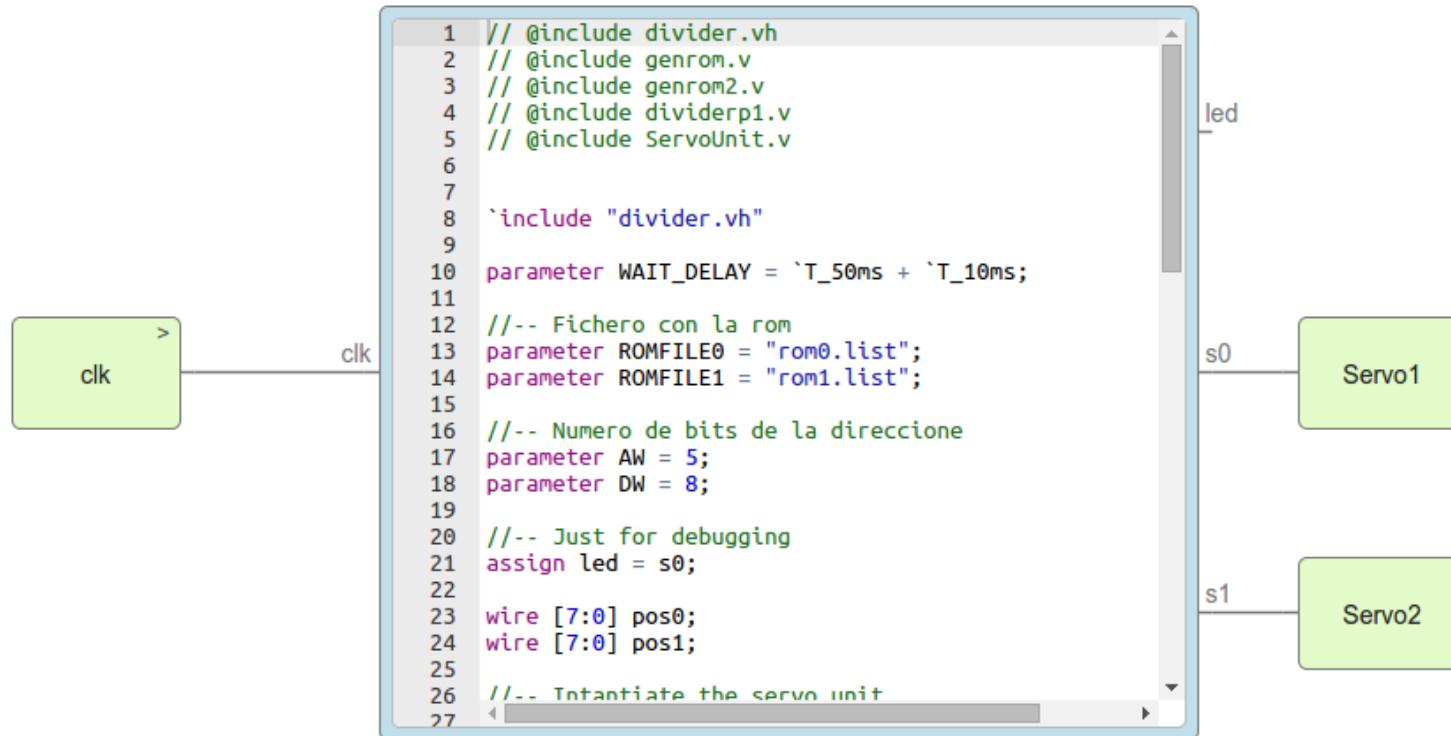


- Servos conectados directamente a Icezum Alhambra
- Configuración mínima pitch-pitch
- Módulo impresos en 3D

Circuito Larby en Icestudio (I)

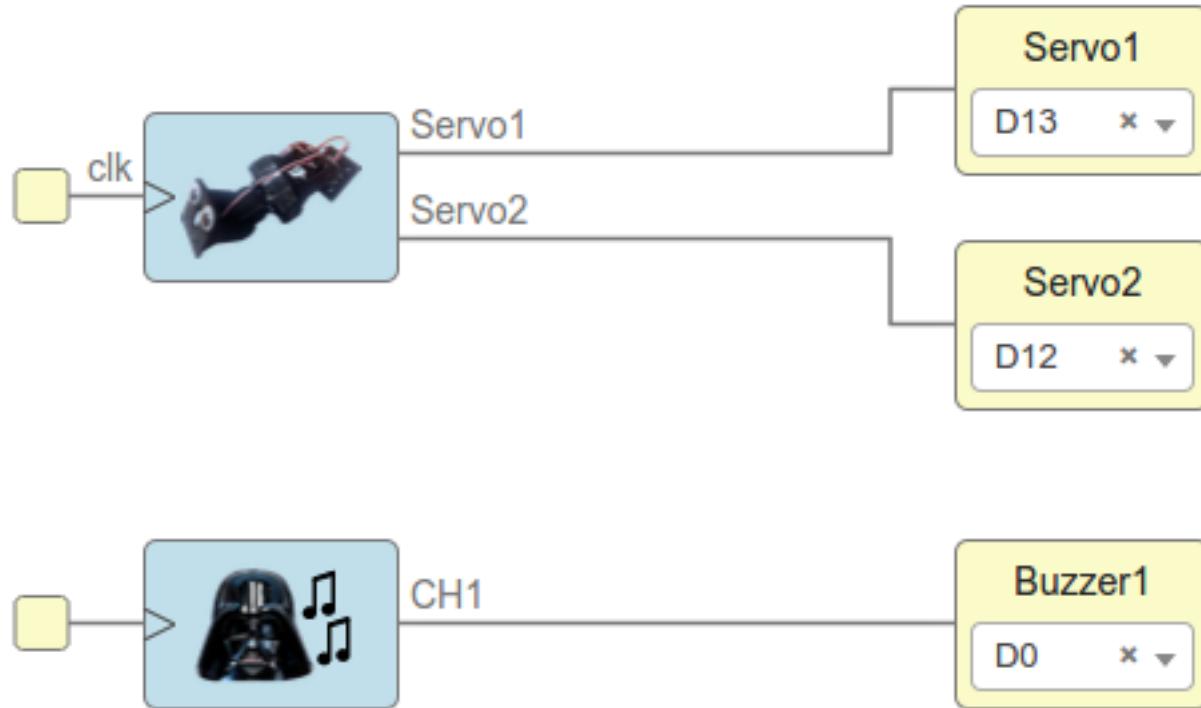


Circuito Larby en Icestudio (II)



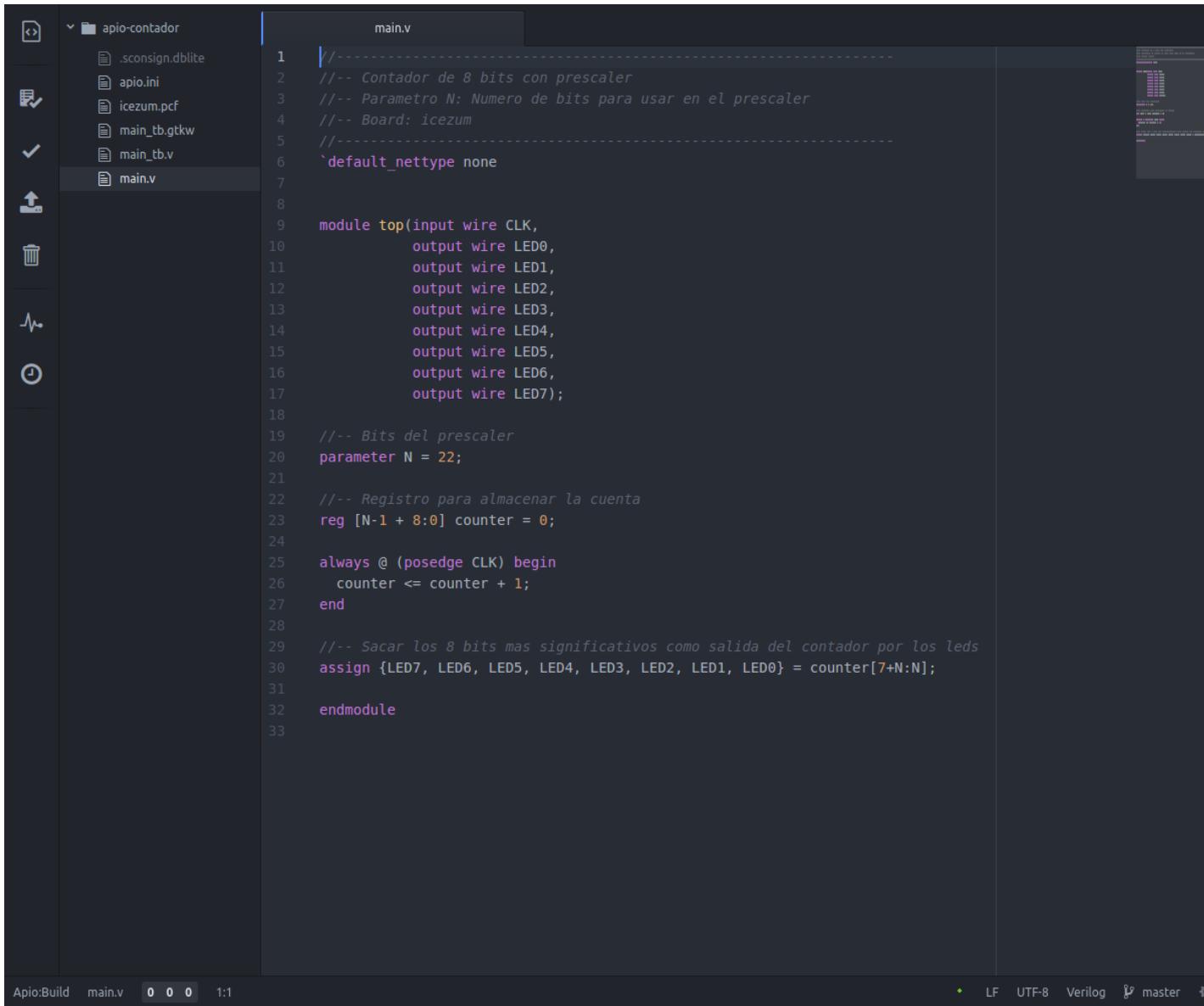
- Código Verilog

Circuito Larby en Icestudio (III)



- Añadamos música...
- El hardware funciona en paralelo...

Apio Ide (I)



The screenshot shows the Apio IDE interface. On the left is a file explorer with the following contents:

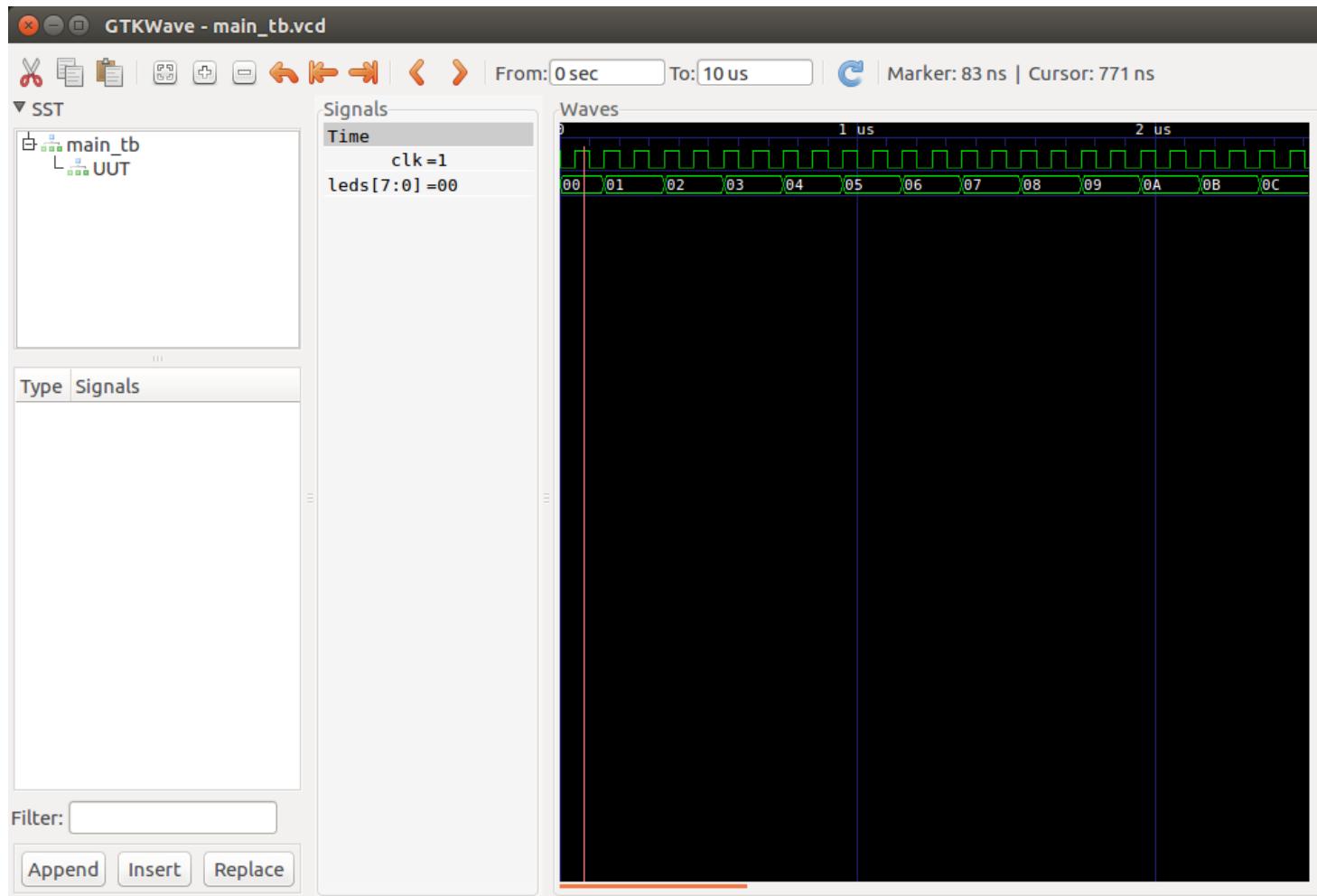
- apio-contador (selected folder)
 - .sconsign.dblite
 - apio.ini
 - icezum.pcf
 - main_tb.gtkw
 - main_tb.v
 - main.v

The main area displays the Verilog code for "main.v". The code defines a top module with 8 output wires (LED0 to LED7) and a parameter N set to 22. It includes an always block for incrementing a counter and an assign statement to map the counter value to the LED pins.

```
1 //-----
2 ///- Contador de 8 bits con prescaler
3 ///- Parametro N: Numero de bits para usar en el prescaler
4 ///- Board: icezum
5 //-----
6 `default_nettype none
7
8
9 module top(input wire CLK,
10             output wire LED0,
11             output wire LED1,
12             output wire LED2,
13             output wire LED3,
14             output wire LED4,
15             output wire LED5,
16             output wire LED6,
17             output wire LED7);
18
19 ///- Bits del prescaler
20 parameter N = 22;
21
22 ///- Registro para almacenar la cuenta
23 reg [N-1 : 8:0] counter = 0;
24
25 always @ (posedge CLK) begin
26     counter <= counter + 1;
27 end
28
29 ///- Sacar los 8 bits mas significativos como salida del contador por los leds
30 assign {LED7, LED6, LED5, LED4, LED3, LED2, LED1, LED0} = counter[7+N:N];
31
32 endmodule
33
```

At the bottom, the status bar shows "Apio:Build" and "main.v" along with build statistics: 0 0 0 and 1:1. To the right of the status bar are icons for LF, UTF-8, Verilog, master, and settings.

Apio Ide (II)



Simulación

Lattuino

https://github.com/INTI-CMNB/Lattuino_IP_Core



Lattuino_Counter | Arduino 1.8.2

File Edit Sketch Tools Help

Lattuino_Counter

```
// Lattuino Stick
// 4 bit counter

#define D1 14
#define D2 0
#define D3 1
#define D4 2
#define D5 3

#define DELAY 8

byte counter = 0;
int ledPin[] = {D4,D3,D2,D1};


```

Done uploading.

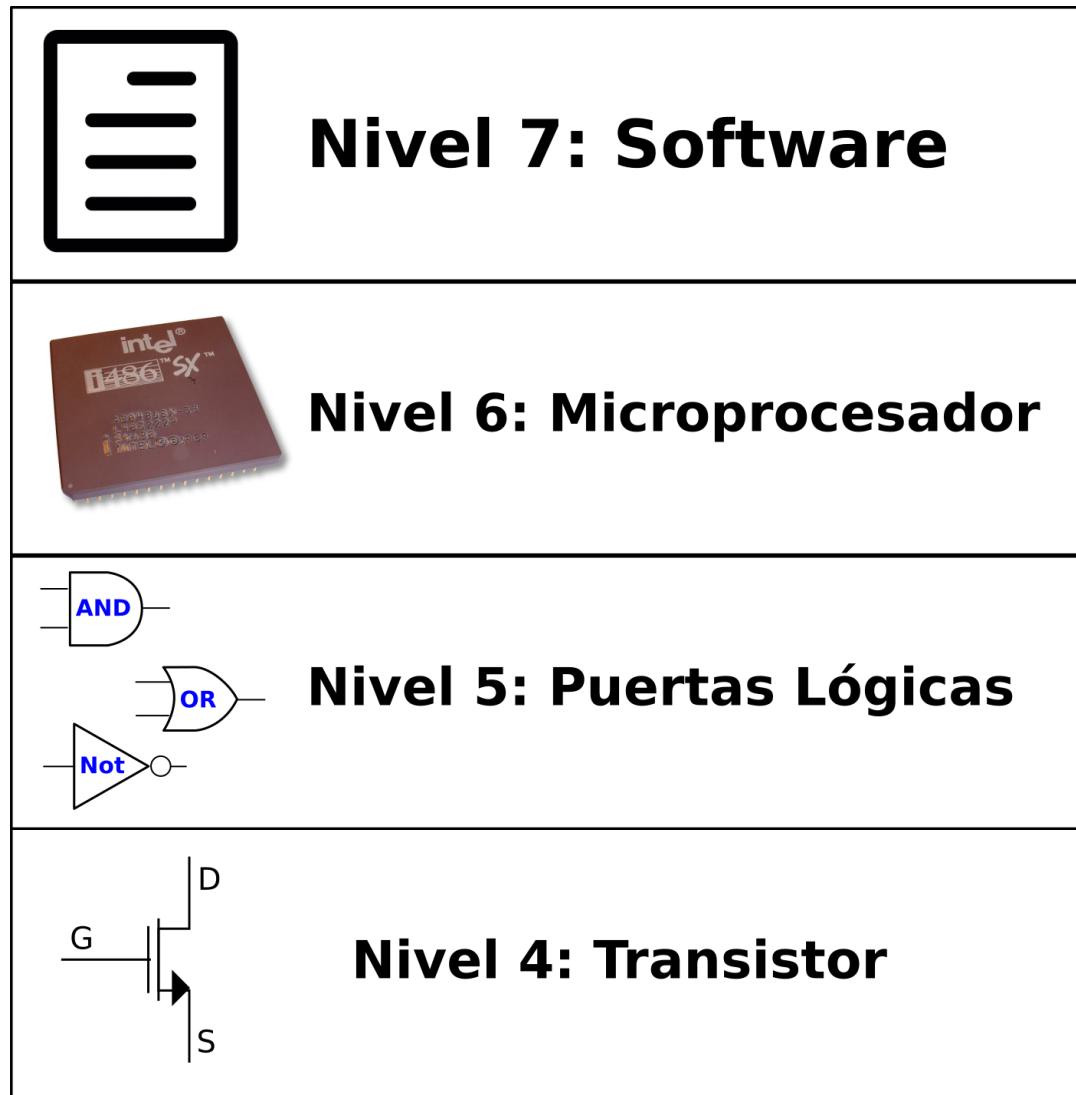
Sketch uses 496 bytes (35%) of program storage space. Maximum j
Global variables use 10 bytes (7%) of dynamic memory, leaving I

28 Lattuino Stick (2k) on /dev/ttyUSB1

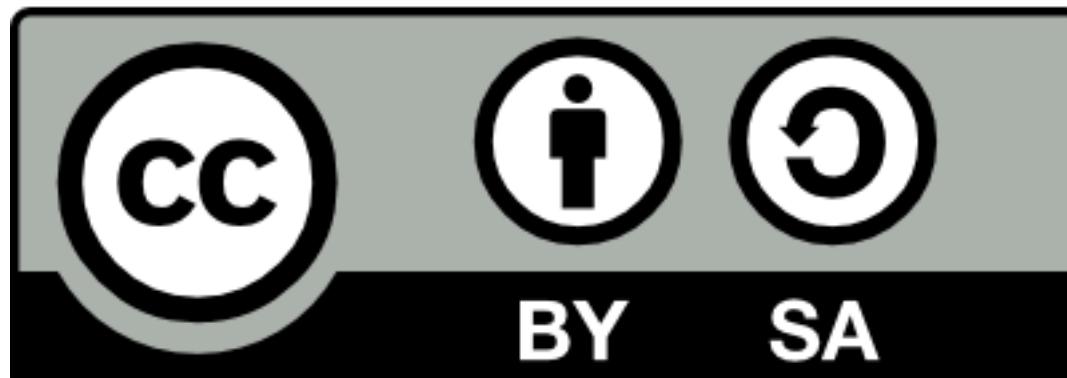
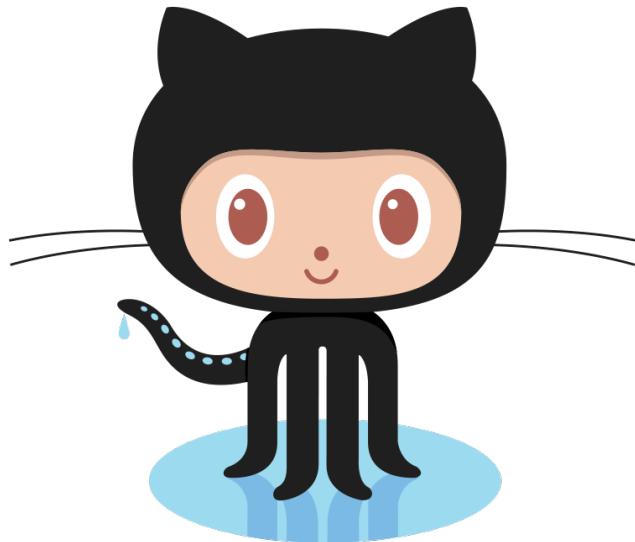
- Autor: **Salvador Tropea**
- Core de Arduino para FPGA
- Lattice Ice40 (1k, 4k, 8k)
- **VHDL**
- Herramientas privativas
- Migrando a **herramientas libres**



Lattuino (II)



¡Comparte con la comunidad!

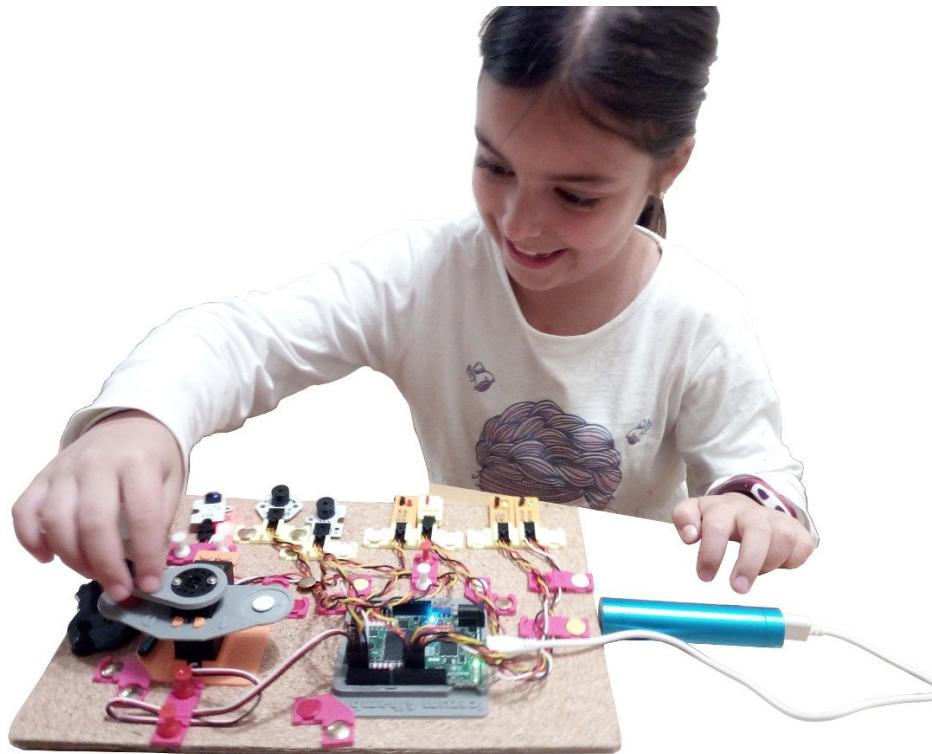


¡Que las FPGAs libres os acompañen!





Electrónica digital para todos con FPGAs Libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>