

# Software libre Embebido

Víctor Suárez García - Akademy- ES 2020

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the bottom half of the slide.

# ¿Quién Soy?

## **Víctor Suárez García**

Ingeniero informático por la Universidad de Almería; miembro activo de Hacklab Almería y persona que siempre está tratando de aprender y enseñar nuevas tecnologías. Además de un apasionado de los juegos y consolas retro.



# ¿De qué vamos a hablar?

- Qué es el software embebido o empotrado.
- ¿Microprocesadores, Microcontroladores, FPGA?
- Dispositivos que podemos encontrar.
- El software libre y herramientas que podemos usar.

# ¿Que es el software embebido?

Un sistema embebido o empotrado (integrado, incrustado) es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real.



# Sistema embebido o empotrado

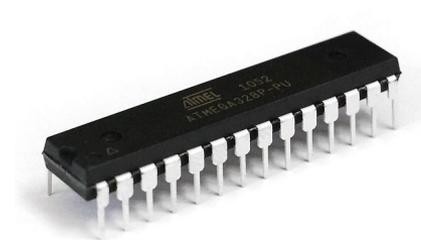
## Microprocesadores

Un microprocesador es un circuito integrado, que permite ejecutar programas, a través de un sistema Operativo.



## Microcontroladores

Es un circuito integrado programable que permite ejecutar órdenes grabadas en su memoria; no necesita un sistema operativo como tal.



## FPGA

Una matriz de puertas lógicas programables o FPGA es un dispositivo programable que permite interconectar distintos bloques creando un nuevo hardware.



# Microprocesador

Un microprocesador, es un circuito integrado que permite ejecutar programas a través de un sistema operativo (normalmente); estos programas, pueden ser ejecutados a través de una memoria externa (no tienen por que estar dentro del propio chip), y necesitan una memoria RAM externa para funcionar.

Existen soluciones de microprocesadores con memoria y toda la funcionalidad dentro del propio chip; normalmente se conocen como SoC (System on a Chip). Ej BCM2835



# DSP

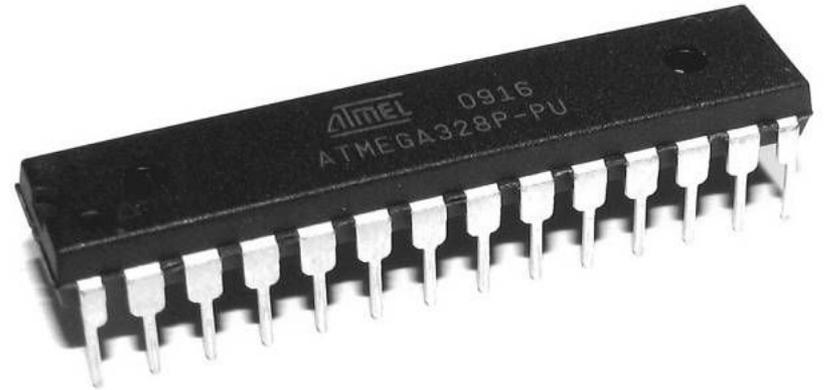
Un DSP o Procesador de señales digitales, es un sistema basado en un microprocesador que contiene un hardware y software optimizados para aplicaciones que requieran operaciones de muy alta velocidad; es útil para procesador y representar operaciones en tiempo real. Normalmente utiliza un conversor de señales Analógicas a Digital DAC/ADC.



# Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar órdenes que se han grabado en su memoria, Normalmente se compone de 3 partes; unidad central de procesamiento, memoria y Entrada/salida.

Una de las principales diferencias que tiene un microcontrolador, es que no necesita un sistema operativo como tal (aunque puede tener algún controlador de E/S, memoria,etc...).



# FPGA

Una FPGA, es una matriz de puertas lógicas, que permite interconectar distintos bloques para crear un nuevo circuito integrado. Normalmente se utiliza para crear estos nuevos circuitos de forma optimizada. Normalmente, tiene un software asociado para poder realizar esta interconexión.



# FPGA

## Symbiflow

Conjunto de herramientas libres (toolchain) para desarrollar en distintas FPGAs (Xilinx, Lattice Ice40, Quicklogic, etc...).

Es libre y tiene muchos recursos.

<https://symbiflow.github.io/>

# SymbiFlow

# FPGA Wars

Conjunto de proyectos Open Source, que permite aprender a utilizar las distintas FPGAs (también libres), de forma sencilla con muy buenos tutoriales.



# NES-MD



# Crear Nuestro S.O. embebido

Para utilizar los microprocesadores, necesitamos un Sistema Operativo; por lo tanto necesitaremos uno acorde al procesador que tengamos.

En este aspecto podemos encontrar dos proyectos que nos ayuden.

- YoctoProject
- BuildRoot

yocto .  
PROJECT



# Crear nuestro S.O embebido

Un ejemplo que podemos ver, es usar un Compute Module de Raspberry Pi, para crear una "Retro Consola de videojuegos"

Ejemplo la Retro-CM3

<https://www.instructables.com/Retro-CM3-a-Powerful-RetroPie-Handled-GAME-Console/>



# Java

Existen implementaciones de la JVM en hardware; es decir, un chip que interpreta el Java Bytecode para poder ejecutar programas; no necesita un Sistema Operativo como tal.

Existen distintas implementaciones; obviamente no todas son Software Libre.



# Arduino

Arduino es una plataforma de prototipado de código abierto.

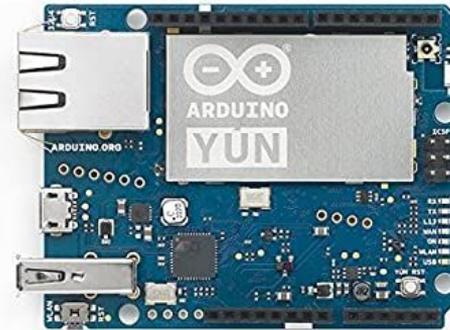
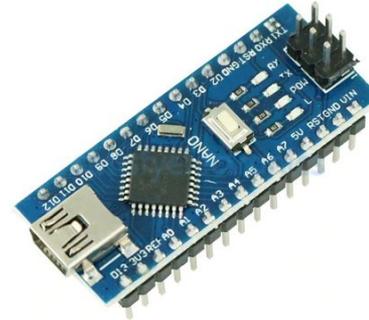
- Hardware libre.
- Herramientas de prototipado libres (IDE, compiladores, etc...).
- Recursos y documentación.



# Arduino

```
Archivo Editar Programa Herramientas Ayuda
sketch_mar19a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

9 Arduino Yún en COM9



# Python

Proyectos Open Source relacionados con el lenguaje Python para Sistemas Embebidos:

- PlatformIO
- MicroPython
- ESPHome



# PlatformIO

Conjunto de herramientas que permiten desarrollar soluciones para sistemas empotrados. Utiliza python para poder llamar a las distintas herramientas de compilación, depuración, testing,etc...

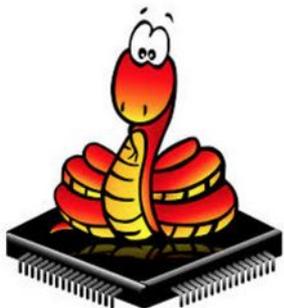
Además de que tiene una gran comunidad y muchos recursos.



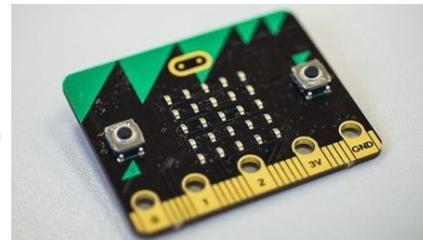
# MicroPython

Implementación ligera del intérprete y librerías para microcontroladores.

- ESP8266
- ESP32
- BBC MicroBit
- etc...



```
1 from machine import Pin
2 from time import sleep
3 #El Pin del led es el 2 que corresponde al GPIO2
4
5 ledpin=2
6 pin = Pin(ledpin, Pin.OUT)
7 while True:
8     pin.value(1)
9     sleep(1)
10    pin.value(0)
11    sleep(1)
```



# ESPHome

ESPHome es un proyecto openSource, que permite a través de una serie de ficheros de configuración, generar y programar un microcontrolador ya preparado. Utiliza PlatformIO.



<https://esphome.io>



Quick search

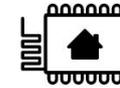
 Go

Table of Contents

ESPHome

- Guides
- Devices
- Core Components
- Sensor Components
- Binary Sensor Components
- Output Components
- Light Components
- Switch Components
- Fan Components
- Display Components
- Cover Components
- Text Sensor Components
- Climate Components
- Misc Components
- Additional Custom Components
- Cookbook

ESPHome



# ESPHome

ESPHome is a system to control your ESP8266/ESP32 by simple yet powerful configuration files and control them remotely through Home Automation systems.

[GitHub](#) [Discord](#) [Twitter](#) [Donate](#)

Guides

		
<a href="#">Getting Started through Command Line</a>	<a href="#">Getting Started with Home Assistant Add-on</a>	<a href="#">FAQ and Tips</a>
		
<a href="#">Migrating from Tasmota</a>	<a href="#">Automations</a>	<a href="#">Contributing</a>
		
<a href="#">Changelog</a>	<a href="#">Supporters</a>	<a href="#">DIY Examples</a>
		

# GO

Utilizando el lenguaje Go, podemos crear soluciones para dispositivos empotrados. Go tiene librerías y herramientas para programar en sistemas empotrados, además, existen distintos proyectos para ayudar a esto

- TyniGo

<https://tinygo.org/>



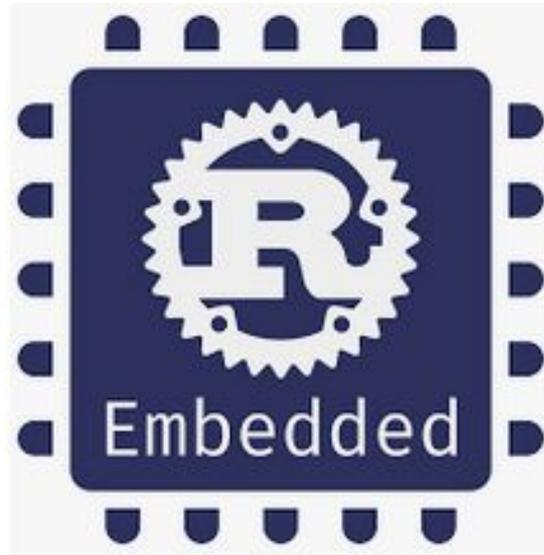
# Rust

Rust, permite desarrollar para dispositivos embebidos, de manera que podemos utilizar tanto librerías realizadas en C/C++, como las desarrolladas con el propio Rust; además de la gestión de memoria que nos provee Rust.

<https://www.rust-lang.org/what/embedded>

<https://github.com/rust-embedded>

<https://dev.to/creativcoder/how-to-run-rust-on-arduino-uno-40c0>



# Referencias

- <https://es.wikipedia.org/wiki/Microprocesador>
- <https://es.wikipedia.org/wiki/Microcontrolador>
- [https://es.wikipedia.org/wiki/Procesador\\_de\\_se%C3%B1ales\\_digitales](https://es.wikipedia.org/wiki/Procesador_de_se%C3%B1ales_digitales)
- <https://github.com/SymbiFlow>
- <http://fpgawars.github.io/>
- <https://joc.systems/en/>
- <https://www.yoctoproject.org/>
- <https://buildroot.org/>
- <https://www.arduino.cc/>
- <https://platformio.org/>
- <https://esphome.io/>
- <https://tinygo.org/>
- <https://www.rust-lang.org/what/embedded>