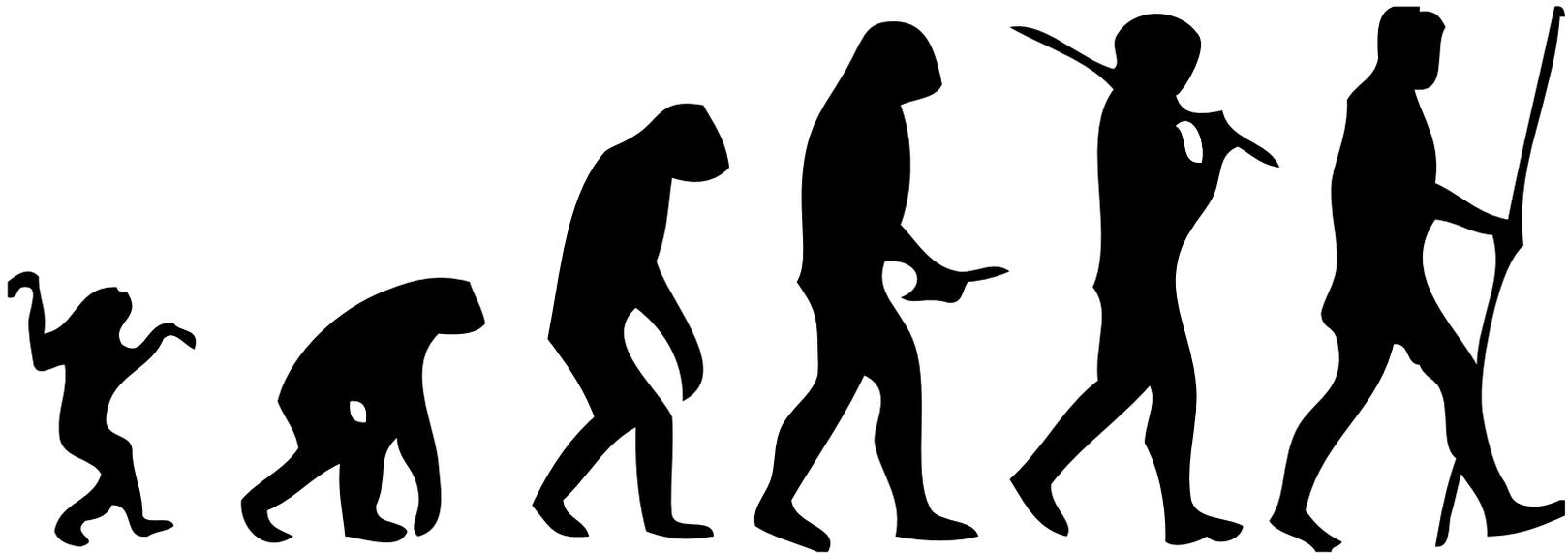




UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



## Presentación curso ARM Cortex-M



2013/04/12



[armcortexm.blogs.upv.es](http://armcortexm.blogs.upv.es)



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# Contenido

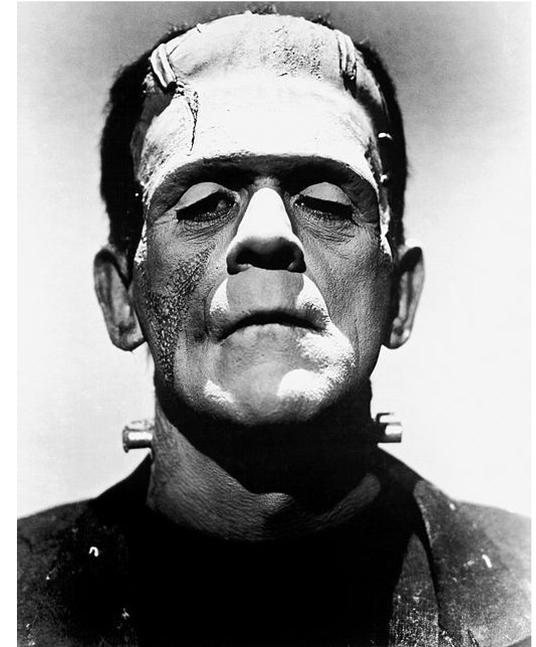
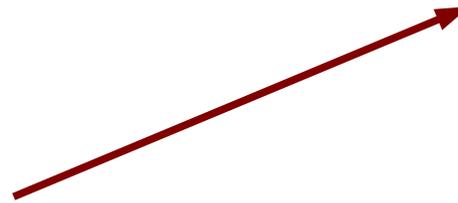
- ARM Ltd. y su modelo de licencia
- La familia ARM Cortex
- Los microcontroladores ARM Cortex-M
- ¿Me interesa Arduino o ARM Cortex-M?
- Por qué elegimos ARM-Cortex-M
- El curso: del paraíso al infierno
- Quiero aprender por mi cuenta
- Que hardware usaremos
- Deberes para la siguiente sesión



# ARM Ltd. y su modelo de licencia

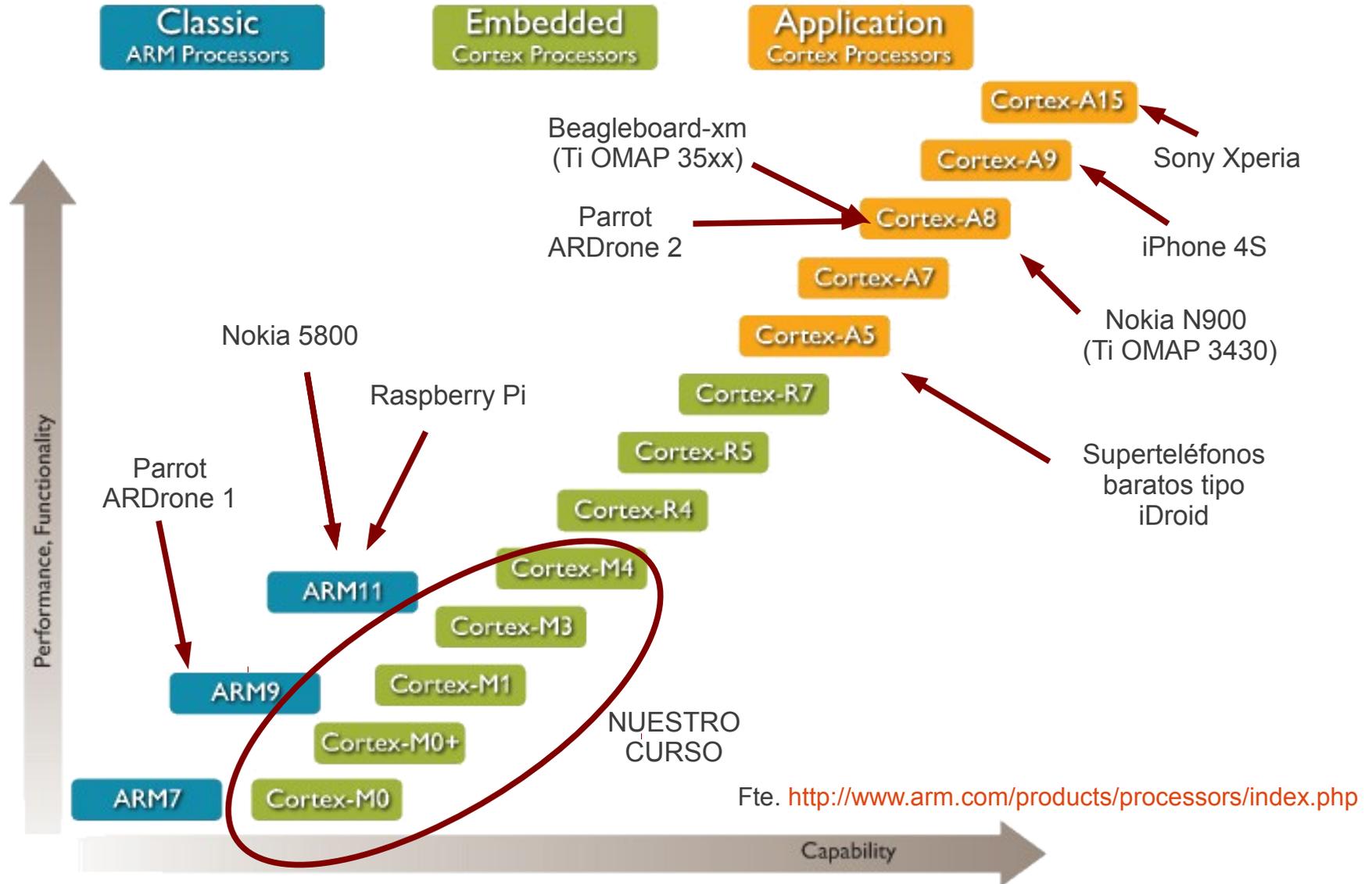
- Advanced **RISC** Machines Holdings Limited (**ARM**) diseña y licencia procesadores
- Los clientes adquieren estas licencias en forma de **Intellectual Property (IP) Verilog** junto con herramientas adicionales
- Los clientes tiene derecho adaptar y complementar los IP
- Las adaptaciones son fabricadas:
  - o por el mismo cliente
  - o encargadas a terceros (fables)
  - o son relicenciadas

un ARM



# La familia ARM Cortex

- Cortex-A de Application, -R de Real-time y -M de Microcontroller



# La familia ARM Cortex

- Cortex-A de Application
  - aplicaciones de alto rendimiento con sistemas operativos generales
  - tablets, e-books, móviles, smart-TV, ...
- Cortex-R de Real-time
  - sistemas de tiempo real-críticos y rendimiento equilibrado
  - robótica, impresoras, control de electrónico de motores, ...
- Cortex-M de Microcontroller
  - aplicaciones típicas de microcontroladores
  - lavadoras, microondas, mandos, nodos inalámbricos, ...
- TÍPICO FALLO DE LA GENTE
  - matar mosca a cañonazos



# La familia ARM Cortex-M

- Quizá esta tabla sea suficiente para entender las diferencias

ARM Cortex-M Instruction Sets <sup>[6][7]</sup>

ARM Cortex-M	Thumb	Thumb-2	Hardware Multiply	Hardware Divide	Saturated Math	DSP Extensions	Floating Point	ARM Architecture	Core Architecture
Cortex-M0 <sup>[1]</sup>	Most	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M <sup>[6]</sup>	Von Neumann
Cortex-M0+ <sup>[2]</sup>	Most	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M <sup>[6]</sup>	Von Neumann
Cortex-M1 <sup>[3]</sup>	Most	Subset	3 or 33 cycle	No	No	No	No	ARMv6-M <sup>[6]</sup>	Von Neumann
Cortex-M3 <sup>[4]</sup>	Entire	Entire	1 cycle	Yes	Yes	No	No	ARMv7-M <sup>[7]</sup>	Harvard
Cortex-M4 <sup>[5]</sup>	Entire	Entire	1 cycle	Yes	Yes	Yes	Optional	ARMv7E-M <sup>[7]</sup>	Harvard

Fte. [http://en.wikipedia.org/wiki/ARM\\_Cortex-M](http://en.wikipedia.org/wiki/ARM_Cortex-M)



# Te interesa Arduino o ARM Cortex-M

- Arduino Uno (Atmel AVR 8)
- ARM Cortex-M

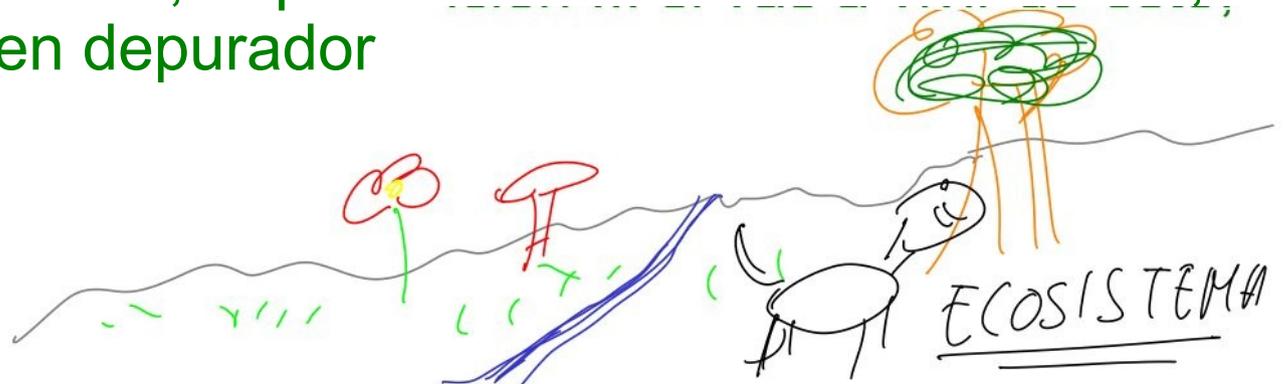


- ¿No sabes programar?
  - ¡Empieza con Arduino! (no sea que te estampes).
- ¿Tienes idea de PIC, 8051, HC-11, ... ?
  - ¡Pásate a ARM Cortex-M! (Incluido el Arduino Due)



# Por qué elegir ARM Cortex-M

- UNA arquitectura MUCHAS implementaciones
  - Cada fabricante añade su especialidad: automoción, aeroespacial, bajo consumo, FPGA, ...
- Muchas herramienta hardware y software. Comerciales y libres
  - Emuladores, compiladores, IDEs, bibliotecas, ...
- Comunidad muy activa: blogs, foros, proyectos (mbed, lpcpresso, mapple, ...)
- “Starter kits” prácticamente regalados
- Depuración, depuración, depuración ... si vas a vivir de esto, más te vale un buen depurador



# El curso: del paraíso al infierno

- Nuestro plan. Quizá cambie.

Sesión 1: "Getting started"	Sesión 2: Comprendiendo	Sesión 3: El fondo	Sesión 4: Lo típico	Sesión 5: Lo más
<p>Presentación. Modelo de licencia de ARM y razones de su éxito.</p> <p>"Getting started". Puesta a punto y prueba de un sistema completo incluyendo instalación y configuración del software, configuración de la placa de evaluación, prueba con un programa ejemplo.</p> <p>Entrada y salida digital. Subsistema de E/S mediante bibliotecas.</p>	<p>El estándar CMSIS y bibliotecas de dispositivos del fabricante.</p> <p>Entendiendo un proyecto software con ARM Cortex-M.</p> <p>Entrada/salida analógica. Nociones básicas de conversión A/D, D/A. Subsistema de E/S analógica.</p>	<p>Toolchain: descripción de sus componentes.</p> <p>Depuración: conceptos básicos.</p> <p>Buenas prácticas de C para microcontroladores. volatile, static, alineación de datos, mapa de memoria.</p>	<p>Sistema interrupciones.</p> <p>Contadores y temporizadores. Temporizadores. PWM.</p>	<p>Ejemplo de sistema operativo con FreeRTOS.</p> <p>Ejemplo de servicios TCP/IP y servidor web para desde a Internet.</p> <p>Criterios de selección de herramientas y micros.</p>
				



# Quiero aprender por mi cuenta

- (Y soy pobre) nuestras recomendaciones



LPCExpresso



STM32 discovery



# Qué hardware usaremos

- St STM32F4 Discovery
  - ARM Cortex-M4
  - 168 MHz
  - 2 MB Flash, 196 Kb RAM



# Deberes de este tema

- Echar un vistazo a las variantes ARM Cortex-M en
  - [http://en.wikipedia.org/wiki/ARM\\_Cortex-M](http://en.wikipedia.org/wiki/ARM_Cortex-M)
- Informarse sobre el kit STM32F4
  - <https://my.st.com/public/STe2ecomunities/mcu/default.aspx>
  - Google y preguntar sobre el kit y tus intereses
  - Youtube

