

# Dispositivos y plataformas para IoT

Ramón Alcarria  
Garrido

Tomás Robles  
Valladares

Miguel Ángel  
Manso Callejo

Borja Bordel  
Sánchez



**POLITÉCNICA**

**Introducción a la Internet de las Cosas**  
**Departamento de Ingeniería de Sistemas Telemáticos (UPM)**

# PROGRAMA

- Microcontroladores
- System on chip
- Ordenadores de placa simple
- Sistemas propietarios

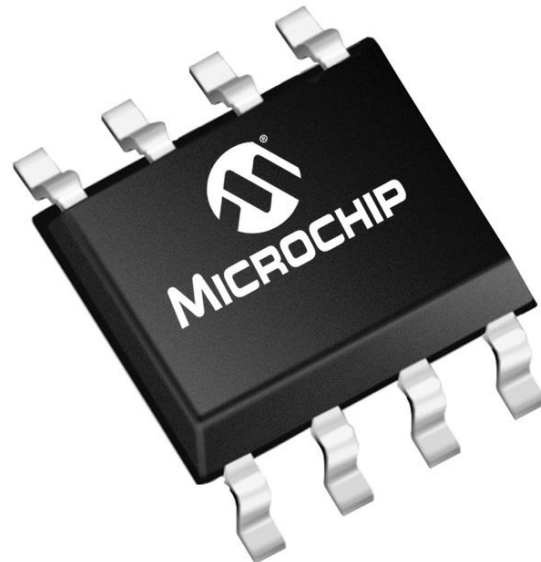


# MICROCONTROLADORES

- Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.
  - Se suele abreviar como  $\mu$ C, UC o MCU
- Está compuesto de varios bloques funcionales, cada uno de los cuales cumplen una tarea específica

# MICROCONTROLADORES

- Básicamente
  - Unidad central de procesamiento
  - Memoria
  - Periféricos de entrada/salida



# MICROCONTROLADORES

- Los microcontroladores exigen la programación de su memoria EEPROM mediante código máquina que puede escribirse de diversas maneras
  - Lenguaje de bajo nivel
  - Código ensamblador propietario

```
.file "hola.c"
.section .rodata
.LC0:
.string "%d"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl $3, %edx
movl $2, %eax
addl %eax, %edx
;; movl $.LC0, %eax
;; movl %edx, %esi
```

# MICROCONTROLADORES

- Existen muchos fabricantes de microcontroladores en la actualidad
  - Texas Instruments
  - Microchip
  - Motorola
  - Intel
  - ...



**MOTOROLA**



**MICROCHIP**



**TEXAS  
INSTRUMENTS**

# MICROCONTROLADORES

- Estos dispositivos, en general, requieren importantes conocimientos acerca del funcionamiento de un computador a muy bajo nivel
  - Registros, acumuladores, longitudes de palabra, espacio de memoria, etc.



# MICROCONTROLADORES

- Un microcontrolador puede seguir diversas arquitecturas
  - Von Neuman
  - Harvard
- La gestión de interrupciones y otros problemas de muy bajo nivel requieren diferente aproximación

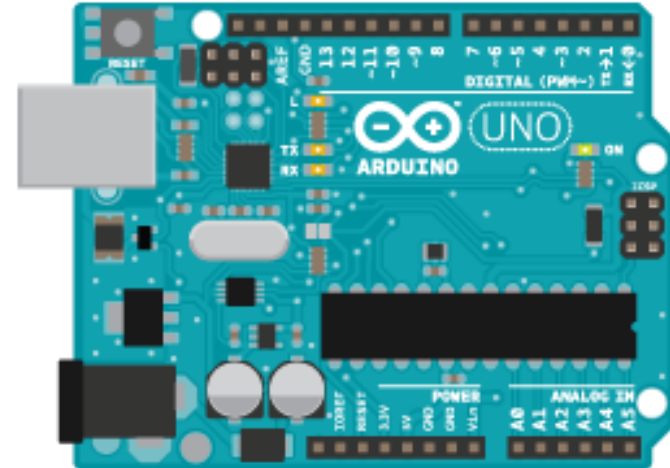


# MICROCONTROLADORES

- Ocultar estos detalles para permitir un rápido desarrollo de sistemas y dispositivos, especialmente por personas que solo tenga un conocimiento medio de esta tecnología comenzó a cobrar interés en la década de los 2000

# MICROCONTROLADORES

- El proyecto Arduino comenzó en 2005 como un intento de crear microprocesadores baratos que pudieran ser usados por estudiantes

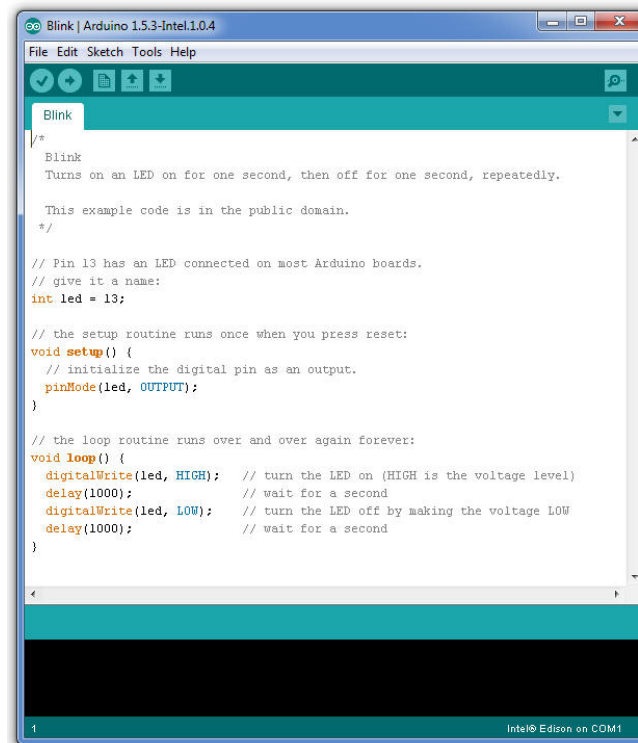


# MICROCONTROLADORES

- Para la producción en serie de la primera versión se tomó en cuenta que el coste no fuera mayor de 30 euros, que fuera ensamblado en una placa de color azul, debía ser Plug and Play y que trabajara con todas las plataformas informáticas tales como MacOSX, Windows y GNU/Linux.

# MICROCONTROLADORES

- Arduino se programa mediante su propio entorno de programación y compilación



```
Blink | Arduino 1.5.3-Intel.1.0.4
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

# MICROCONTROLADORES

- Estructura básica de un programa

```
Blink

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

**SETUP**

Se ejecuta una vez al comienzo

**LOOP**

Se ejecuta de forma infinita

# MICROCONTROLADORES

- Existen placas Arduino, placas de expansión, kits de desarrollo, etc.
- Toda la documentación sobre las operaciones nativa y la instrucciones disponibles se puede consultar en su página web
  - <https://www.arduino.cc/en/Reference/HomePage>

# MICROCONTROLADORES

- Pinout de Arduino Nano

No. PIN	Nombre	Dirección	Descripción
1-2, 5-16	D0(TR1), D1 (RX00) D2-D13	Entrada/Salida	Entradas Digitales
3, 28	Reset	Entrada	Reseteo (Activo en Bajo)
4, 29	GND	PWD	Tierra
17	3V3	Salida	Salida +3.3V (desde FTDI)
18	AREF	Entrada	Referencia ADC
19-26	A0-A7	Entrada	Entradas Analógicas
27	+5V	Entrada o Salida	Salida de 5V (del regulador) Entrada 5V (alimentación externa)
30	VIN	PWR	Alimentación de Voltaje

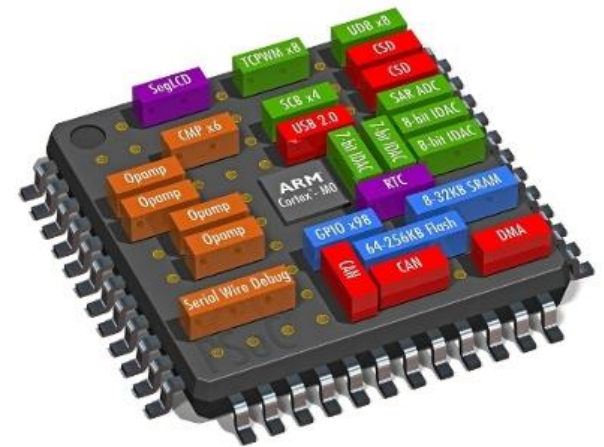
# SYSTEM ON CHIP

- Los System on Chip, abreviados SoC, se refieren a la integración de todos los módulos de un sistema electrónico mediante técnicas de fabricación microelectrónica
- Es común para sistemas de transmisión integrados con el microcontrolador que los gestiona



# SYSTEM ON CHIP

- Texas Instruments es actualmente el fabricante más conocido de estos circuitos
  - Tiene familias para sistemas basados en Bluetooth, ZigBee, etc.
- Normalmente su programación es compleja

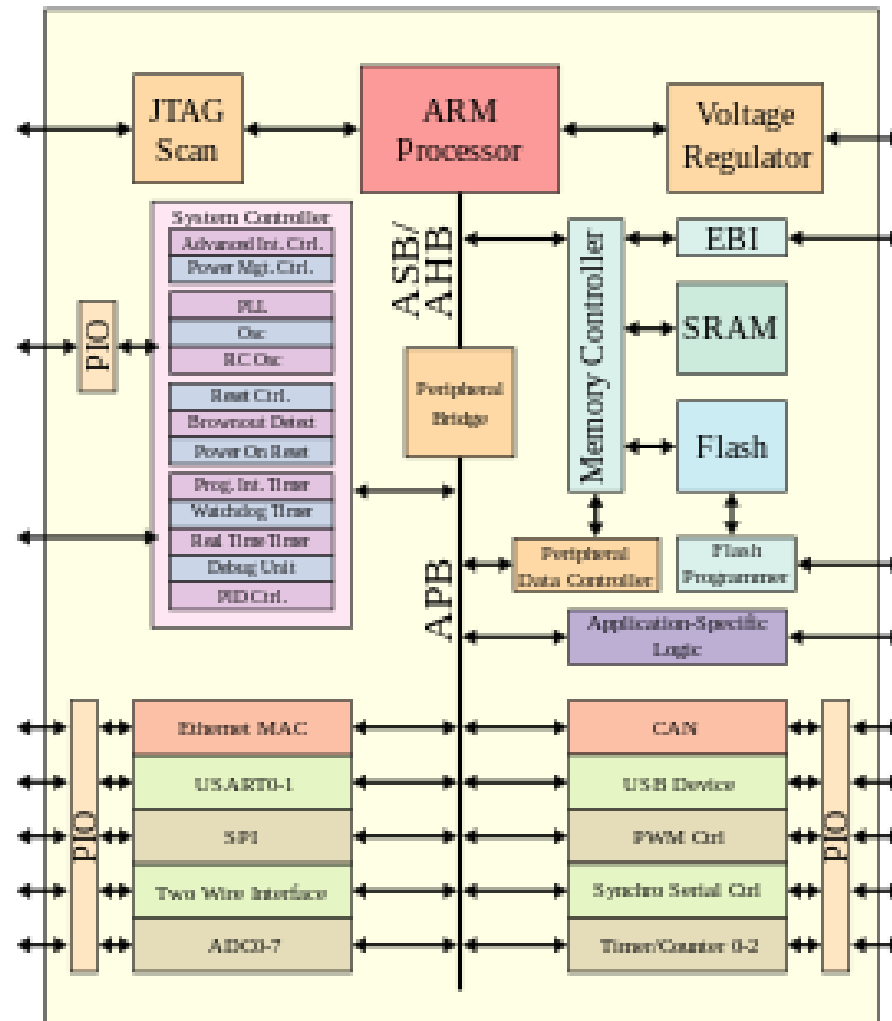


# SYSTEM ON CHIP

- Normalmente, existen entornos de programación propietarios para cada fabricante de SoC
- Lo más habitual es que se tenga que emplear lenguaje ensamblador
- Su uso y diseño no es sencillo

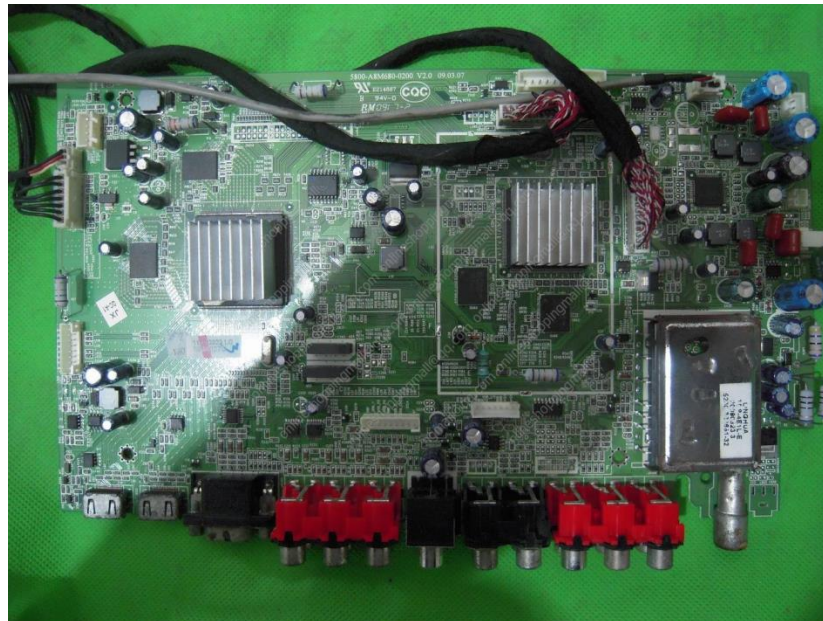
# SYSTEM ON CHIP

- SoC típico



# ORDENADORES DE PLACA SIMPLE

- Los ordenadores de placa simple (SBC, por sus siglas en inglés) es un completo ordenador funcional construido en una única placa impresa



# ORDENADORES DE PLACA SIMPLE

- Al contrario que los PC, los SBC no tienen ranuras para expandir sus prestaciones
- Suelen emplear varias familias de microprocesadores de 8 o 16 bits y memoria RAM estática
- Algunos están especialmente diseñados para ser servidores

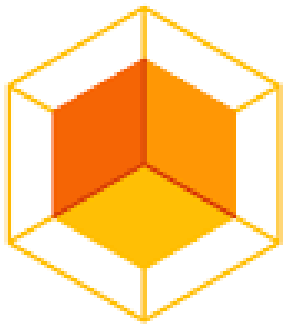
# ORDENADORES DE PLACA SIMPLE

- Los SBC son muy usados en la creación de primeros prototipos
- Al contrario que los MCU y los SoC, los SBC incluyen un sistema operativo, por lo que para trabajar con ellos se emplean técnicas de programación de sistemas

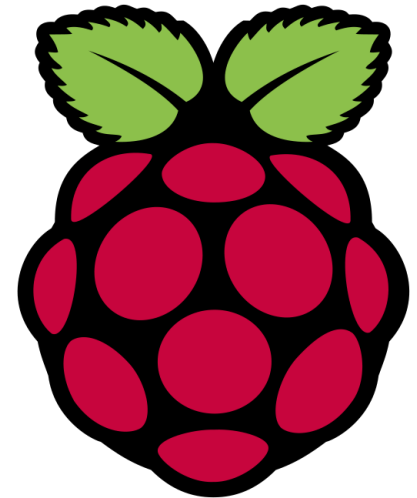


# ORDENADORES DE PLACA SIMPLE

- Actualmente hay varios fabricantes de SBC, aunque son dos los más utilizados
  - Raspberry Pi
  - Samsung Artik



Samsung  
**ARTIK**™



# ORDENADORES DE PLACA SIMPLE

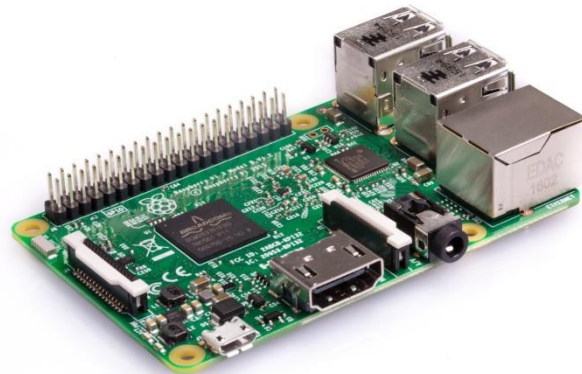
- Artik ofrece un conjunto de módulos para el desarrollo de IoT, entre los que se encuentra ARTIK™ 530





# ORDENADORES DE PLACA SIMPLE

- Los sistemas Raspberry están centrados en el aprendizaje a nivel de primaria
- La fundación Raspberry da soporte para las descargas de las distribuciones para arquitectura ARM



# ORDENADORES DE PLACA SIMPLE

- Sistemas operativos:
  - Raspbian (derivada de Debian)
  - RISC OS 5
  - Arch Linux ARM (derivado de Arch Linux)
  - Pidora (derivado de Fedora);
- Promueve principalmente el aprendizaje del lenguaje de programación Python



# SISTEMAS PROPIETARIOS

- Existen, también, sistemas comerciales (propietarios) pensados para desarrollar plataformas de IoT



# SISTEMAS PROPIETARIOS

- Thingspeak
  - <https://thingspeak.com/>
- Carriots
  - <https://www.carriots.com/>
- Adafruit IO
  - <https://io.adafruit.com/>
- Sentilo
  - <http://www.sentilo.io/wordpress/>

# SISTEMAS PROPIETARIOS

- Devicehive (open source)
  - <http://devicehive.com/>
  - <https://github.com/devicehive>
- Smart Cities as a Service
  - [http://www.iotsens.com/solutions\\_en/smart-city/](http://www.iotsens.com/solutions_en/smart-city/)
- Pubnub
  - <https://www.pubnub.com/>



# SISTEMAS PROPIETARIOS

- Thingworx
  - <http://www.thingworx.com/>
- Temboo
  - <https://temboo.com/>
- Thethings
  - <https://thethings.io/>
- Thinger
  - <https://thinger.io/>

# SISTEMAS PROPIETARIOS

- Ubidots
  - <http://ubidots.com/>
- Onion Cloud
  - <https://onion.io/cloud>
- IBM Bluemix
  - <http://www.ibm.com/cloud-computing/bluemix/>



IBM **Bluemix**<sup>™</sup>



# SISTEMAS PROPIETARIOS

- B-scada
  - <http://www.votplatform.com/>
- Amazon
  - <http://aws.amazon.com/es/iot/>



# SISTEMAS PROPIETARIOS

- En España Telefónica lanzó en el 2014 la plataforma Thinking Things
- Es una solución sencilla basada en módulos tipo Lego<sup>®</sup> apilables con conectividad 2G y plug and play para que los usuarios puedan desarrollar sus propias soluciones inteligentes sin necesidad de saber programación o instalar infraestructura adicional

# SISTEMAS PROPIETARIOS

- Thinking Things tiene web propia
  - <https://iot.telefonica.com/thinking-things>

