

1 – Microprocesadores: Introducción

Objetivos

Microprocesadores y microcontroladores

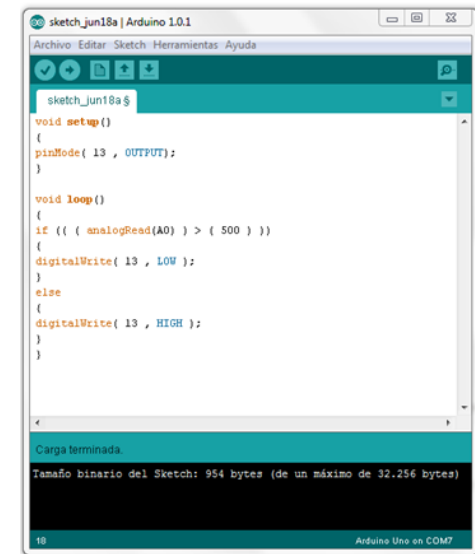
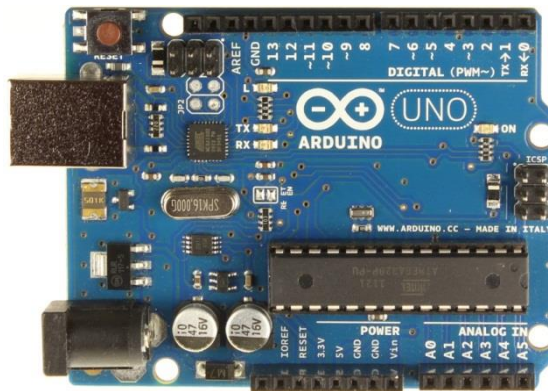
Sistemas objetivo en la asignatura

Organización docente

Objetivos

- En esta parte de la asignatura vamos a estudiar qué es un *sistema microprocesador* (μP) y un microcontrolador (μC)
- Veremos lo que es un sistema embebido y su diferencia con un sistema microprocesador de propósito general (un ordenador, por ejemplo)
- Aprenderemos cómo funciona un microprocesador en un sistema embebido, cómo interactúa con su entorno, y la influencia que esto tiene en cómo se programa
- Aplicaremos los conocimientos a un sistema real (trabajo)
 - Programación en C de un pequeño sistema

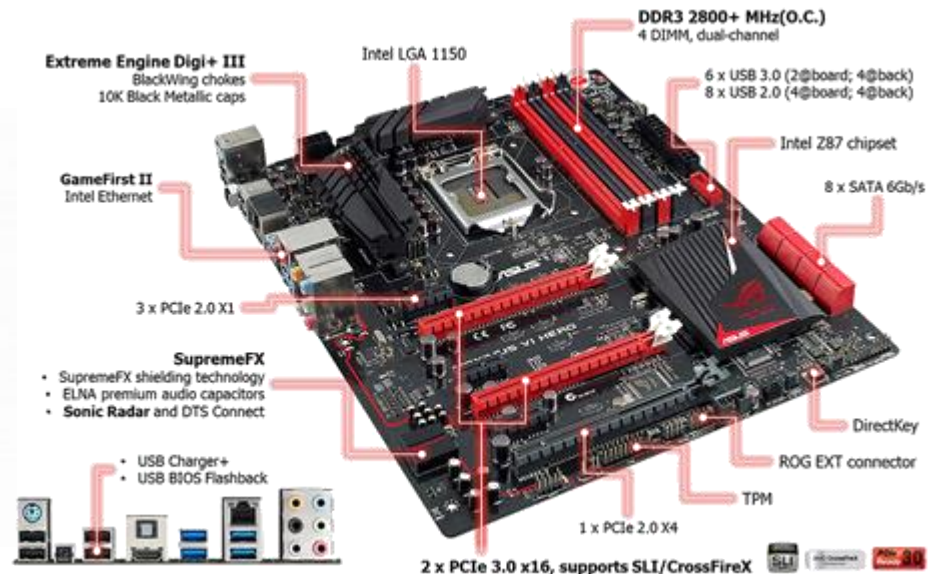
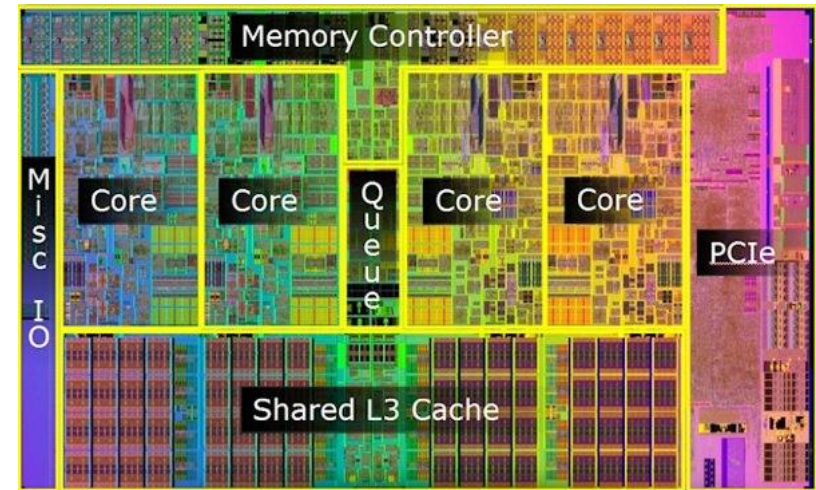
Para poder aplicar de forma práctica los conceptos estudiados, hemos particularizado los ejemplos y contenidos teniendo en cuenta que usaremos una plataforma de prototipado denominada Arduino. Esta plataforma, junto con una placa desarrollada ad-hoc en la ETSII, nos servirá a su vez para realizar el trabajo y las prácticas de esta parte de la asignatura.



Microprocesador (uP) vs Microcontrolador (uC o MCU)

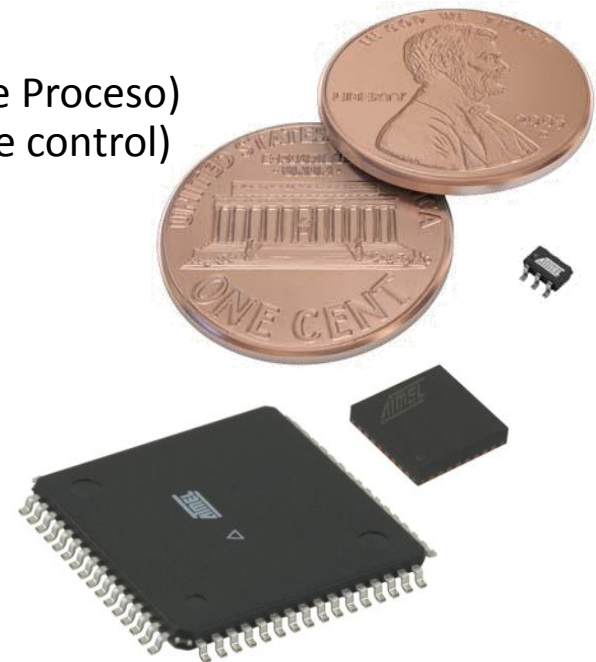
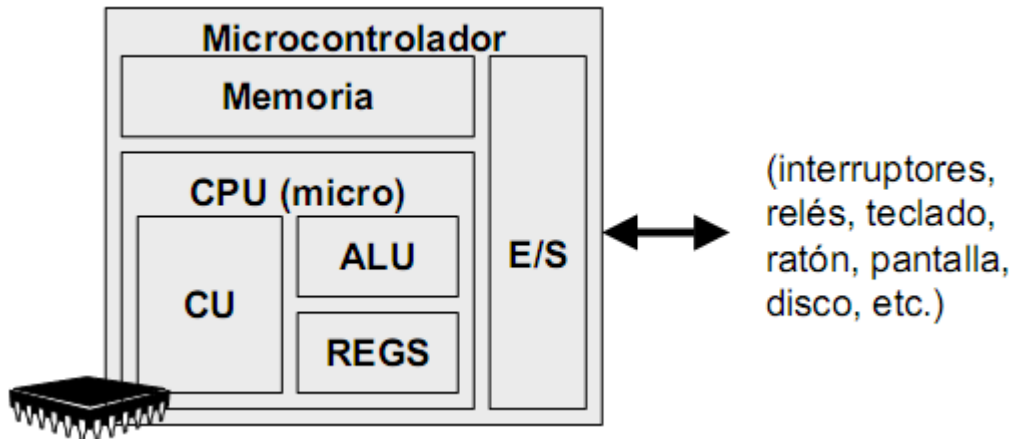
Intel Core 2 Duo architecture

- En nuestro ordenador hay, por ejemplo
 - Un **microprocesador** Intel i7 que tiene cuatro núcleos, cada uno de los cuales tiene...
 - Unidad de Control
 - Unidad Aritmético Lógica y FPU
 - Banco de Registros(veremos estos conceptos más adelante)
 - **Memoria** (los 4, 8 o 16 GB que tengamos) . Son circuitos (chips) adicionales al micro.
 - Lógica de interfaz con los **dispositivos de entrada/salida** (teclado, pantalla, ratón, discos, red, etc). También son circuitos adicionales.



El microcontrolador (μC o MCU)

- Un **microcontrolador** es
 - Un circuito digital integrado (todo un sistema en un chip)
 - Funcionalidad programable vía software
 - Utilizado generalmente para aplicaciones de bajas prestaciones (mucho más pequeñas que un ordenador/tablet/smartphone)
 - Dispone de
 - Memoria (para programa y datos)
 - Dispositivos de entrada/salida (I/O, ó E/S)
 - CPU (**C**entral **P**rocessing **U**nit – Unidad Central de Proceso) (que a su vez, contiene registros, ALU y unidad de control)



microcontrolador
Micro Controller Unit

¿Por qué usamos MCUs? Ventajas e inconvenientes

- Son la **mejor alternativa** frente al diseño digital “ad-hoc” si no hay problemas de prestaciones (fundamentalmente consumo y velocidad de ejecución)
- **Ventajas frente a los diseños digitales “ad-hoc”:**
 - Mayor simplicidad y flexibilidad (modificaciones tras fabricación del sistema)
 - Menor tiempo de desarrollo (del orden de 10 veces menos)
 - Menor coste para la misma funcionalidad, ya que son circuitos «standard» (para tiradas pequeñas < 100.000 -1.000.000 uds/año)
- **Inconvenientes:**
 - Menor velocidad de ejecución
 - Mayor consumo para la misma funcionalidad
 - Mayor coste para grandes tiradas
- **Ejemplos:**
 - Control de un electrodoméstico (uP/uC)
 - Decodificador de video en tiempo real (FPGA/Asic)

¿Dónde hay microcontroladores?



Driver information
Dashboard display
Car navigation

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers

Powertrain
Engine control system
Transmission
Fuel injection
Fuel pump

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers
Power MOS

Body electricals
Integrated body control module
CAN gateway
Light controls

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers
Power MOS
IPD



Chassis
Power steering
Braking
Vehicle dynamic management

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers
Power MOS



Security
Immobilizer
Car alarm

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers

Safety
Airbags
Seat occupation detection
Pedestrian protection
Night vision
Tire air pressure detection

Semiconductors used
8-bit microcontrollers
32-bit microcontrollers



Un coche puede disponer de entre 50 y 150 MCUs
(casi cualquier funcionalidad está realizada mediante una MCU)

http://www.real-programmer.com/interesting_things/IEEE%20SpectrumThisCarRunsOnCode.pdf

http://am.renesas.com/edge_ol/special/02/index.jsp

Algunos datos: Tipos de uP/uC

■ Prestaciones

- Gama baja: 4, 8, 16 bits. Dedicados fundamentalmente a tareas de control (electrodomésticos, cabinas telefónicas, algunos periféricos de ordenadores, etc.). Generalmente son μC .
- Gama media: 16, 32 bits. Tareas de control con cierto grado de procesamiento (control en automóvil, teléfonos móviles, PDA, etc.). Suelen ser μC o μP + periféricos integrados, y memoria externa.
- Gama alta: 32, 64 y 128 bits. Fundamentalmente procesamiento (ordenadores, videoconsolas, etc.) Hay muchos uC, pero también son μP + circuitería periférica + memoria.

■ Tecnología

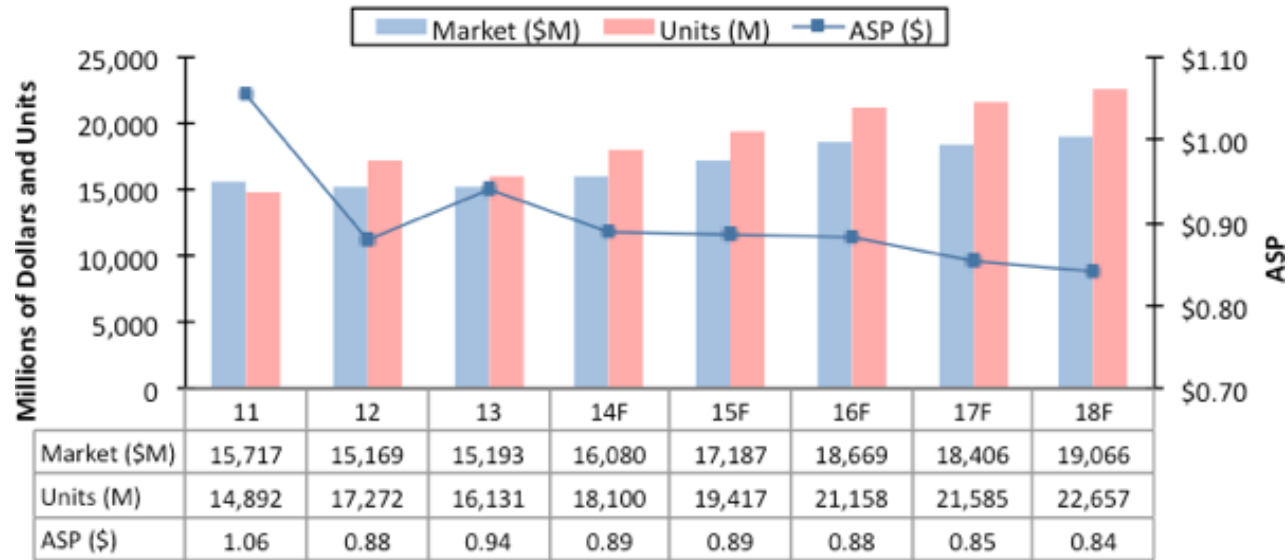
- Alimentación: 5 V, 3.3 V, 2.5 V, 1.8 V
- Consumo: μW – decenas de W
- Frecuencia: de kHz a GHz

■ Precio

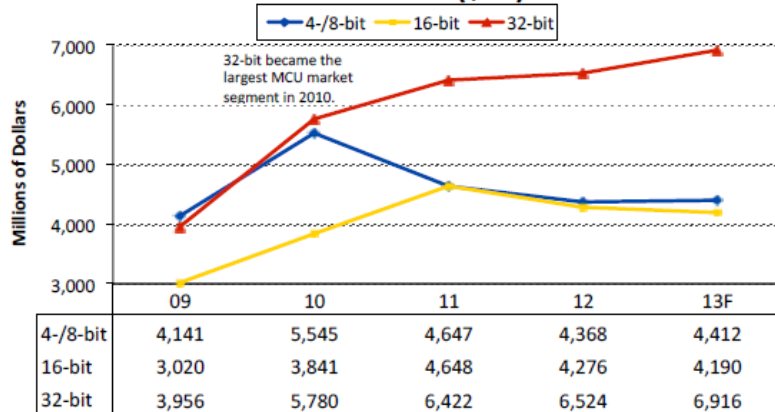
- Microcontroladores sencillos: 0.5€-20€
- Microprocesadores gama alta: 100€-1000€

Algunos datos: mercado de microcontroladores

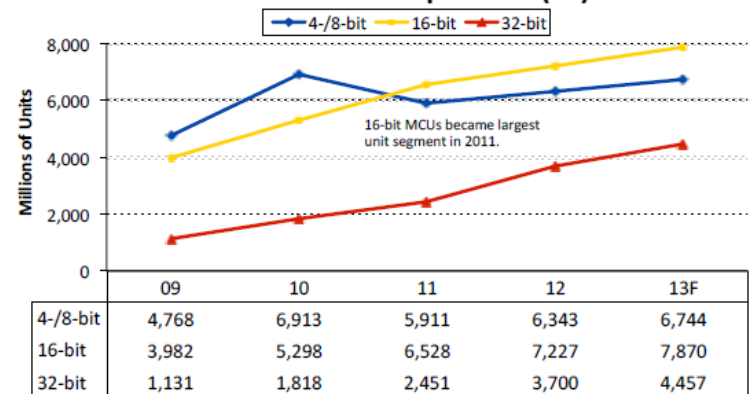
MCU Market History and Forecast



MCU Sales (\$M)



MCU Unit Shipments (M)

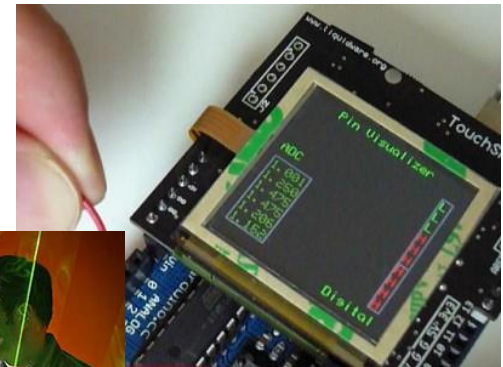
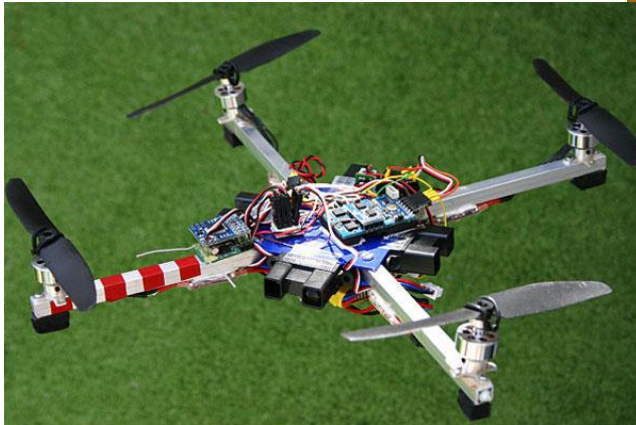


Source: IC Insights

<http://www.icinsights.com/data/articles/documents/541.pdf>

Sistemas objetivo en la asignatura

- Nos vamos a centrar en el diseño de sistemas relativamente sencillos
 - Diseñaremos con μC de bajas prestaciones
 - Fabricante $\rightarrow$ Atmel
 - Arquitectura AVR
 - Programaremos en C (pero teniendo en cuenta su relación con el lenguaje ensamblador, cercano al lenguaje máquina)
- En cualquier caso, los microprocesadores de «bajas prestaciones» tienen capacidades suficientes para muchísimas aplicaciones
- Ejemplos:



Sistemas objetivo en la asignatura

- ¿Qué va a hacer nuestro sistema?
 - Recibir señales de sensores analógicos y digitales
 - Controlar dispositivos externos (luces, altavoces, *displays*...)
 - Comunicarse con un PC
- ¿Qué se espera de vosotros?
 - Que sepáis lo que es un micro (microcontrolador o microprocesador)
 - Que entendáis cómo se ejecuta un programa en un micro
 - Que conozcáis los principales procedimientos de interacción entre los elementos físicos (hardware) y el programa (software/firmware) que los controla
 - Que programéis un sistema real
- Lo que se aprende aquí se usa constantemente en la industria

Organización docente: programación y evaluación

■ Programación temporal:

- Tema 1: Introducción (1h)
- Tema 2: Sistemas embebidos (2h)
- Tema 3: Arquitectura Interna de la CPU (2h)
- Tema 4: Dispositivos de E/S (2h)
- Tema 5: Métodos de E/S (2h)
- Tema 6: Interrupciones (3h)

■ Práctica obligatoria (2h) - Semana 30 nov. al 4 dic.

- Manejo del Arduino
- Ejercicios con E/S

■ PEC voluntaria - 18 de diciembre

- 70% de la calificación de la evaluación continua (CECMic)

■ Trabajo voluntario en equipos de 2 personas:

- 30% de la calificación de la evaluación continua (CECMic)
- Fecha límite: 17 de diciembre 23:59h (no prorrogable)

Organización docente: Requisitos

- ¿De programación? saber programar en C (tipos de datos y sentencias básicas)
 - Ejemplo: ¿Entiendes la sintaxis de este programa?

```
#define LED_PIN 9

// fades on and off a LED
int main() {
    while(true) {
        int x = 1;
        for (int i = 0; i > -1; i = i + x) {
            analogWrite(LED_PIN, i); // analogWrite(LED, x) sets LED intensity to x (0-255)
            if (i == 255) x = -1;
            delay(10); // delay(N) waits for N milliseconds
        }
    }
}
```

- Qué es un byte, un bit, cuánto ocupa un entero, qué valores se pueden codificar en un byte, operadores aritméticos, lógicos y relacionales...
- ¿De electrónica digital? Multiplexores, registros, puertas, y operaciones lógicas
- ¿De electrotecnia? Divisor resistivo y poco más

Bibliografía

- **Microcontrollers and microcomputers : principles of software and hardware engineering**
Cady, Frederick M.
 - Introducción (tema 1): Chapter 1
 - Arquitectura interna (tema 3): Chapter 2
 - Dispositivos de entrada salida (tema 4): Chapter 7, 10
 - Interrupciones (tema 6): Chapter 8
- **General**
 - The microprocessor : a biography
Malone, Michael S.
 - Microprocesadores : diseño práctico de sistemas
Angulo Usategui, José María
- **De Arduino**
 - <http://arduino.cc/en/Reference/HomePage>
 - Arduino programming notebook, Brian W. Evans
- **Enlaces recomendados en las transparencias**