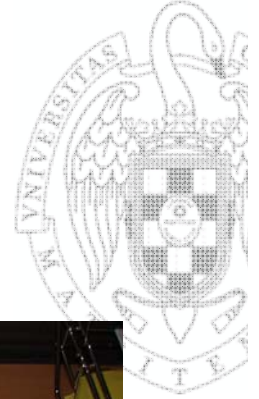


Fundamentos de Computadores

2º Cuatrimestre
2012-2013

PRÁCTICA 5: Sistema de Entrada/Salida
Puesto de laboratorio

Puesto de Laboratorio

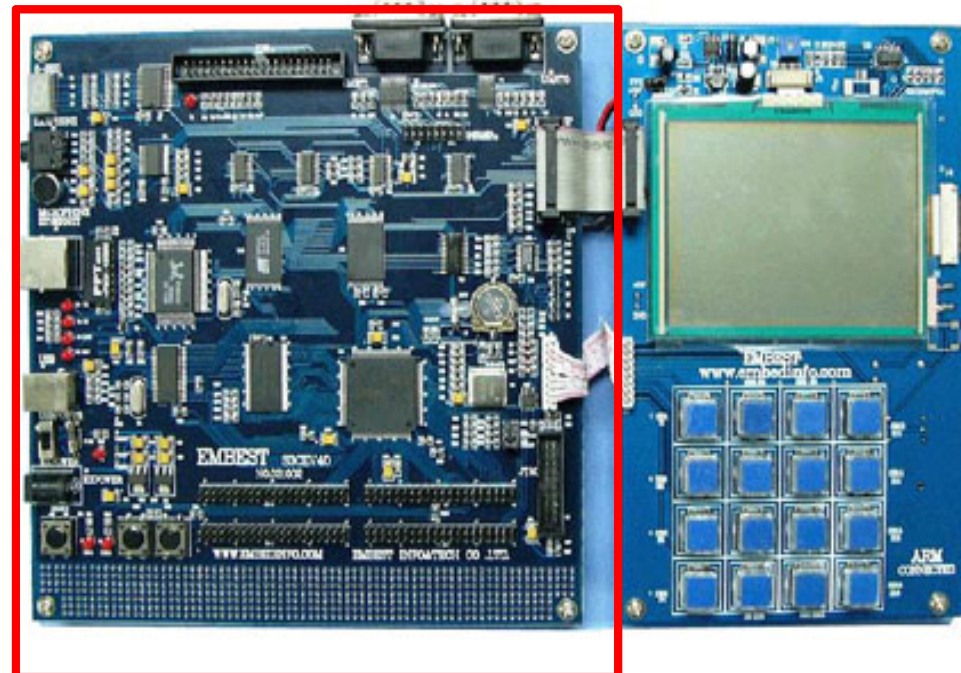


- Kit ARM (maletín)
 - Placa Embest S3CEV40
 - LCD + Touchpad + Teclado
 - Embest UNetICE
 - Cables de conexión
- Software
 - Embest IDE Pro 2004
 - Programador de flash

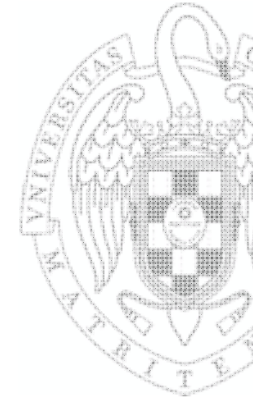


Placa Embest S3CEV40

- Basada en un sistema en chip (SoC) de Samsung, el S3C44B0X, que incorpora un procesador ARM7TDMI



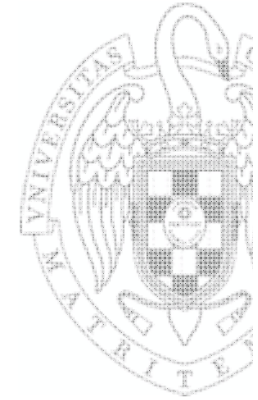
Placa Embest S3CEV40



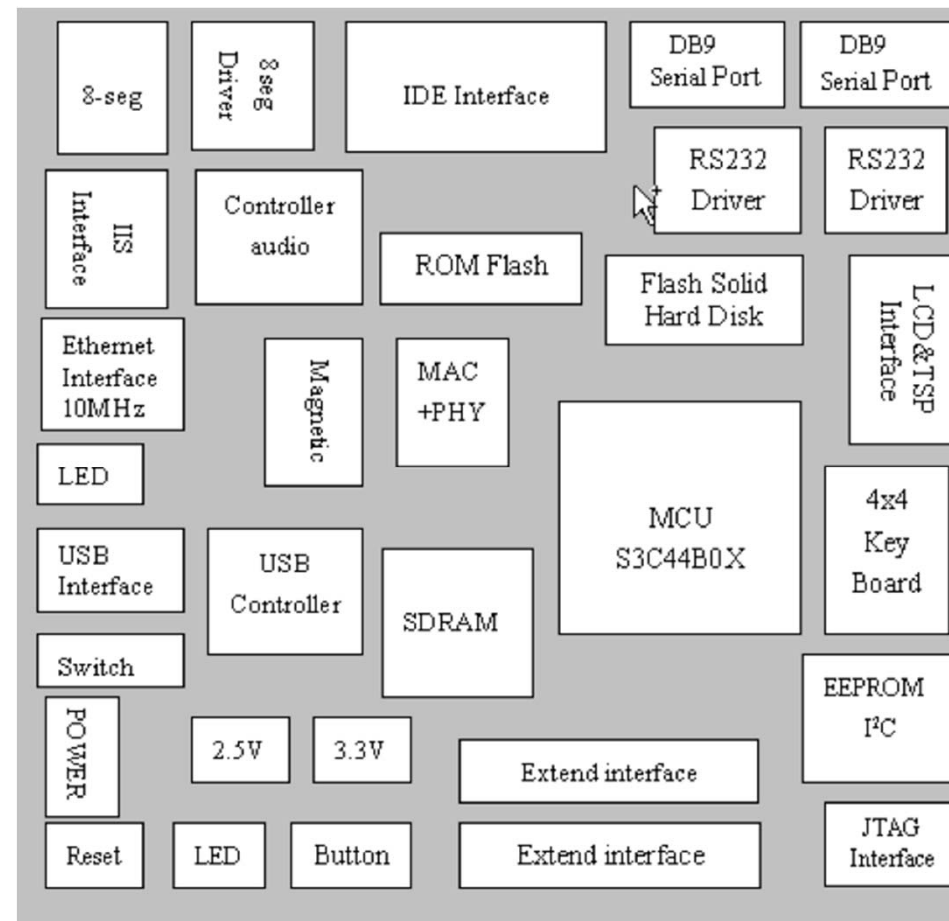
■ Componentes

- Memoria flash de 1M x 16 bits
- SDRAM de 4 x 1M x 16 bits
- EEPROM de 4Kbit (IIC)
- 2 puertos serie (simple y RS-232)
- 2 botones de interrupción
- 2 LEDs
- Interfaz para disco duro IDE
- Conector JTAG
- Conector USB
- Ethernet de 10 Mb/s
- LED de 8 segmentos
- Entrada de micrófono
- Salida de altavoces
- CODEC Audio/Voz
- Memoria NAND flash *de* 16MB

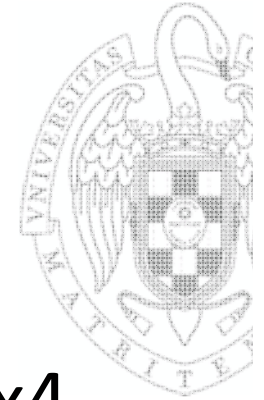
Placa Embest S3CEV40



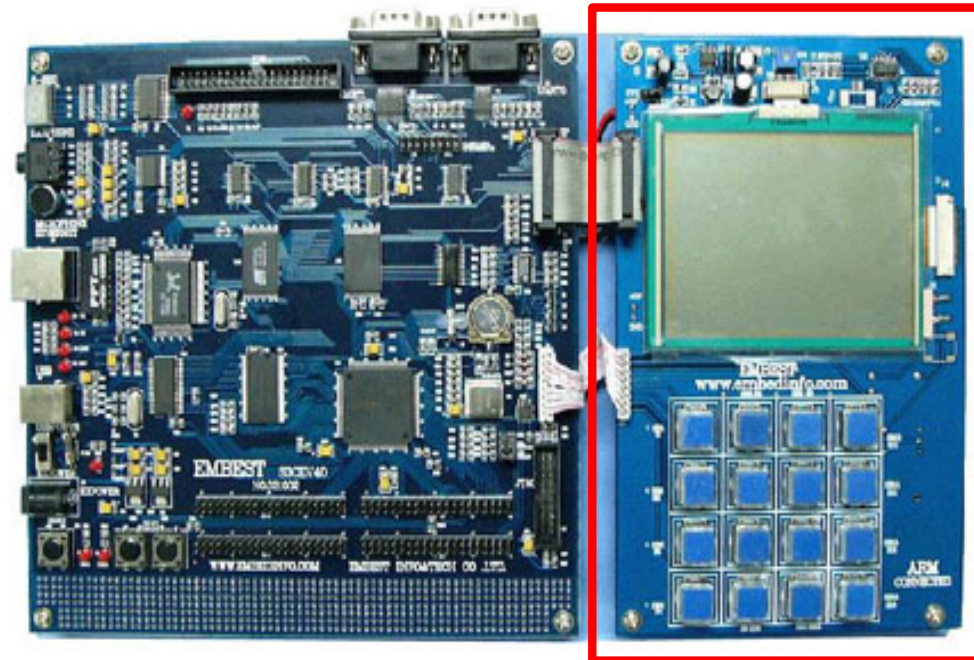
■ Esquema



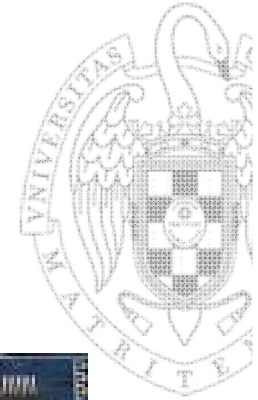
Placa de Expansión



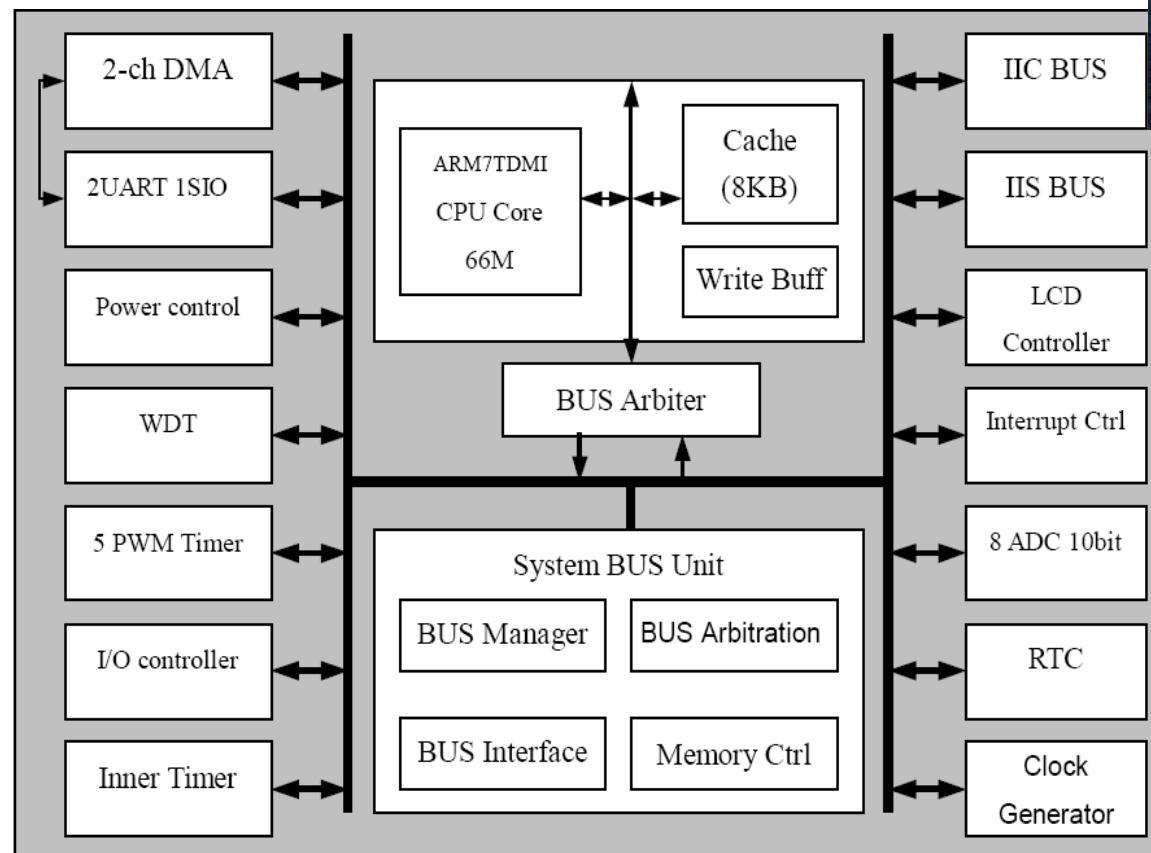
- Añade LCD, Touchpad y teclado matricial 4x4

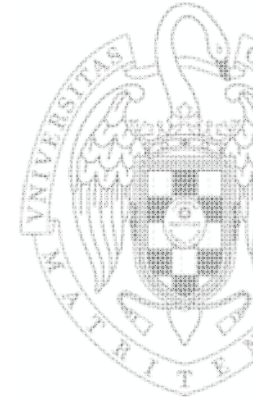


SoC Samsung S3C44B0X



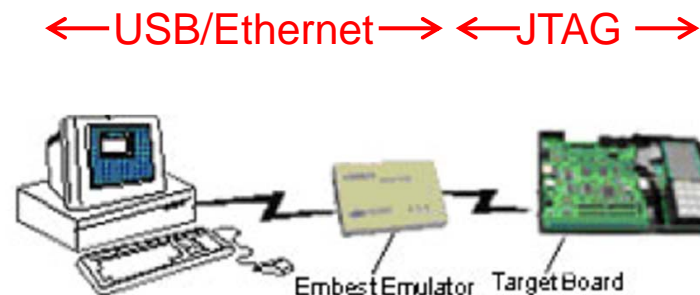
■ Esquema





Embest UNetICE

- Se conoce también como emulador JTAG
- Permite depurar el código sobre la propia placa de desarrollo
- La comunicación con el procesador ARM se realiza mediante el interfaz JTAG (IEEE 1149.1) que es un estándar para depuración en circuito

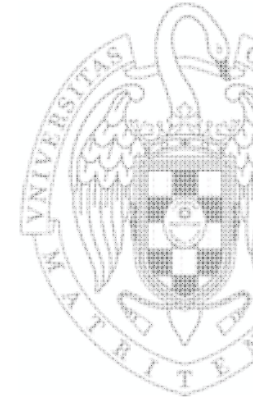


Sistema de E/S del ARM7TDMI



- E/S localizada en memoria
 - Los controladores de los dispositivos forman parte del espacio de direcciones
- 1 línea externa de RESET
- 2 líneas de interrupción (IRQ y FIQ)
 - No las usaremos
- Identificación de dispositivos mediante encuesta

¿Qué periféricos usaremos?



■ LEDs

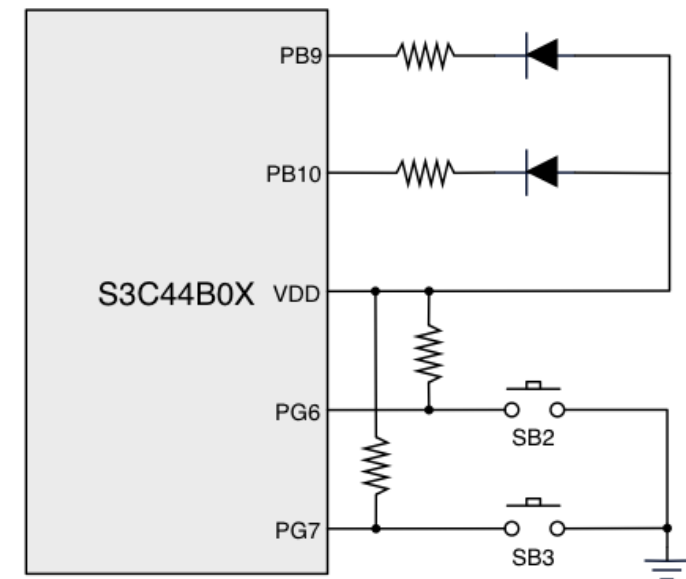
- Existen 2 leds
- Accesibles a través del puerto B de los pines multifunción de E/S (GPIO)

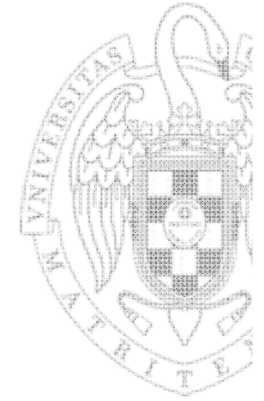
■ Pulsadores

- Existen 2 botones (pulsadores)
- Accesibles a través del puerto G de los pines multifunción de E/S (GPIO)

■ Display 8-segmentos

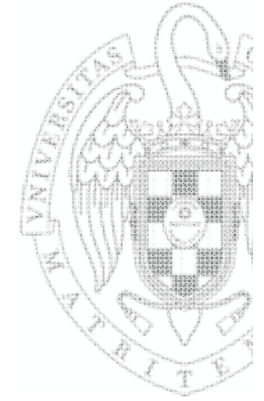
- 7 LEDs para formar cualquier dígito hexadecimal
- 1 LED para el punto decimal
- Si el LED está a '0' → luz encendida





Uso de LEDs

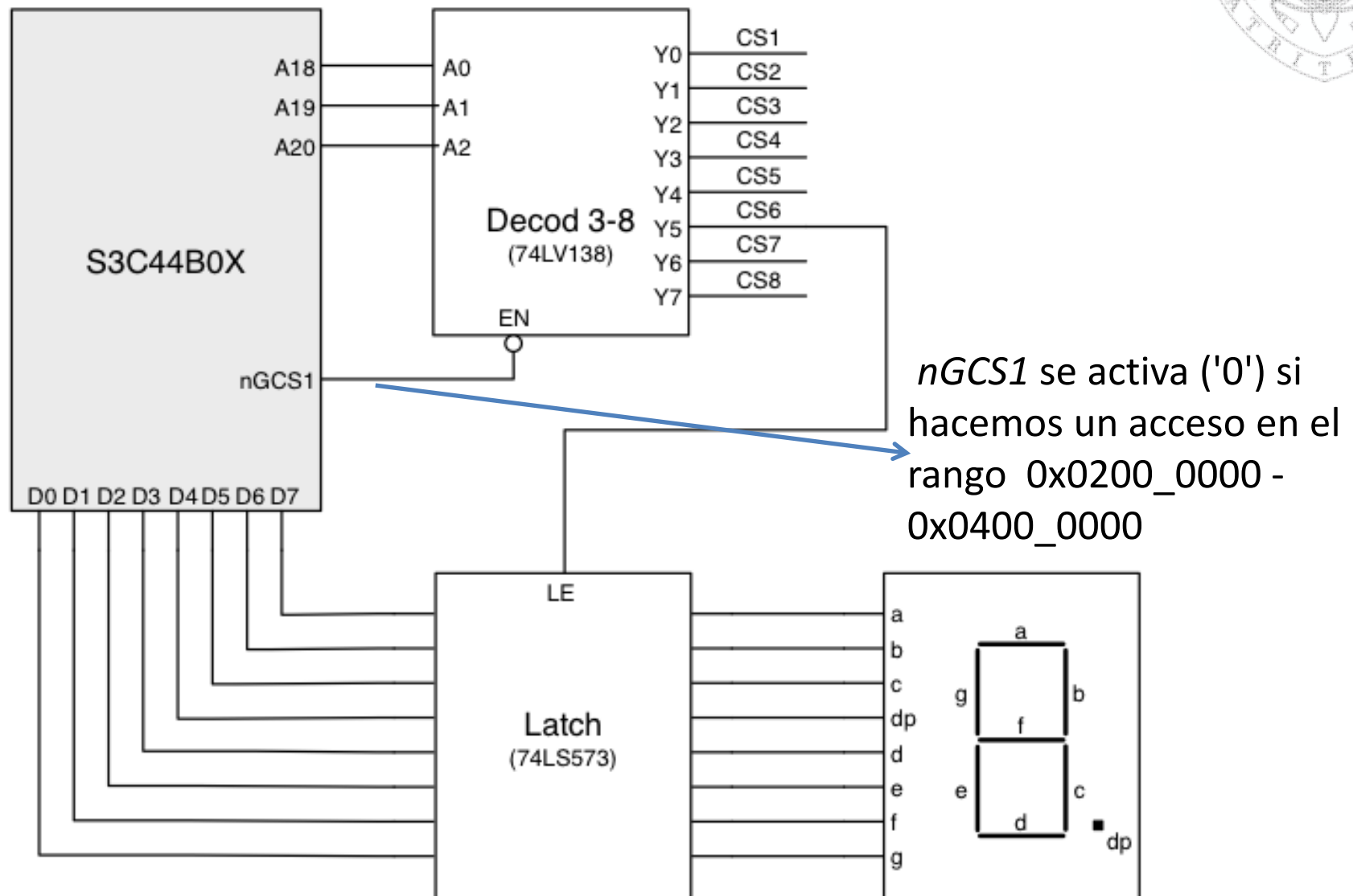
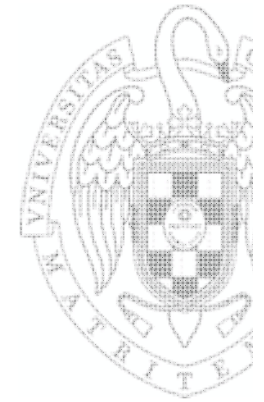
- Conectado los pines 9 y 10 del puerto B
 - Registro de control de 11 bits (PCONB) en dirección 0x01D20008
 - Registro de datos de 11 bits (PDATB) en dirección 0x01D2000C
- Operación:
 1. Configurar pines 9 y 10 como salida (una vez)
 - Escribir un '0' en los bits 9 y 10 de PCONB
 - Simplificación: escribir todo '0' en PCONB
 2. Escribir en PDATB para encender/apagar. Ejemplo:
 - Si bit 9 de PDATB =0 → LED 1 encendido
 - Si bit 10 de PDATB=1 → LED 2 apagado

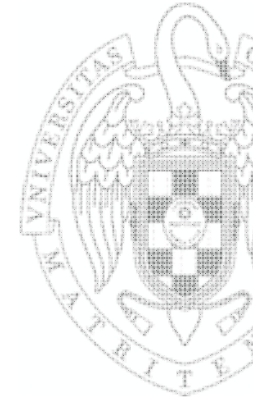


Uso de pulsadores

- Conectado los pines 6 y 7 del puerto G
 - Registro de control de 16 bits (PCONG) en dirección 0x01D20040
 - 2º registro de control de 8 bits PUPG en dirección 0x01D20048
 - Registro de datos de 8 bits (PDATG) en dirección 0x01D20044
- Operación:
 1. Configurar pines 6 y 7 como entrada (una vez)
 - Escribir '0000' en los bits 15:12 de PCONG
 - Escribir todo '0' en PUPG
 2. Leer de PDATG para comprobar si pulsado. Ejemplo:
 - Si bit 6 = 0 → botón 1 pulsado
 - Si bit 7 = 1 → botón 2 NO pulsado

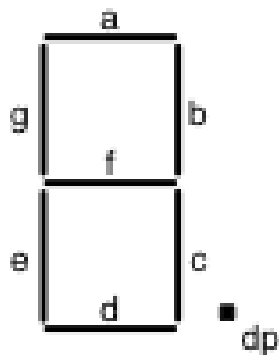
Uso display 8-segmentos





Uso display 8-segmentos

- No existe registro de control (siempre configurados como salida)
- Registro de datos (8 bits) en dirección 0x2140000
 - Cada segmento tiene asociado 1 bit del registro
 - led a -> bit de mayor peso. Orden *a b c dp d e f g*
 - Si $\text{bit}_0=0 \rightarrow$ el segmento *g* se enciende



- ¿Qué ocurre si escribimos 0x18?
- ¿Qué hay que escribir para conseguir una C?