

Ej12.pdf



GeXx_



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática Universidad de Málaga



te imaginas

No pagar ni primera
ni segunda matrícula??



participa

Si consigues subir más apuntes que tus compañeros te regalamos una matrícula valorada en 1000€

WUOLAH

quieres la play quinta??

(no digo el numerito porque ya nos conocemos, don comedia)

.include "inter.inc"

.text



```
garticiga
agui
```





será sorteada
entre todos los
usuarios
estudiantes que
el día de la
finalización del
concurso estén
en el top de su
comunidad

```
/* Agrego vectores de interrupciûn */
    ADDEXC 0x18, irq_handler
/* Inicializo la pila en modos IRQ y SVC */
    mov
           r0, #0b11010010
                           @ Modo IRQ, FIQ&IRQ desact
    msr
            cpsr c, r0
            sp, #0x8000
    mov
            r0, #0b11010011
                            @ Modo SVC, FIQ&IRQ desact
            cpsr c, r0
    msr
            sp, \overline{\#}0x8000000
    /* Configuro GPIOs 9 y 10 como salida */
           r0, =GPBASE
    ldr
             ldr rl.
    str r1, [r0, #GPFSEL0]
    /* guia bits xx999888777666555444333222111000 */
            ldr r1.
    str r1, [r0, #GPFSEL1]
/* Enciendo LEDs 10987654321098765432109876543210 */
            =0b000000000000000000001100000000
    ldr r1,
    str r1, [r0, #GPSET0]
   Habilito pines GPIO 2 y 3 (botones) para interrupciones */
         r1, #0b00000000000000000000000000001100
    mov
           r1, [r0, #GPFEN0]
    str
  Habilito interrupciones, local y globalmente */
    ldr
            r0, =INTBASE
            mov
/* guia bits
                      10987654321098765432109876543210 */
            r1, [r0, #INTENIRQ2]
    str
    mov
            r0, #0b01010011
                            @ Modo SVC, IRQ activo
            cpsr_c, r0
    msr
    /* Repetir para siempre */
bucle: b
               bucle
    /* Rutina de tratamiento de interrupciÛn */
irq_handler:
            {r0, r1}
    push
            r0, =GPBASE
    ldr
    /* Apago los dos LEDs rojos
                                54321098765432109876543210 */
           r1, #0b00000000000000000000011000000000
    mov
            r1, [r0, #GPCLR0]
    str
    /* Consulto si se ha pulsado el botûn GPIO2 */
    ldr
          r1, [r0, #GPEDS0]
            ands
    /* SÌ: Activo GPIO 9; No: Activo GPIO 10 */
    r1, [r0, #GPSET0]
    str
    /* Desactivo los dos flags GPIO pendientes de atenciÛn
```



