

# Ejer14nuevo.pdf



GeXx\_



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga



deja de comprar café en la uni  
que tienes 2€ en el banco.

Yo te regalo un año de Kaiku

participa  
aquí



Todos los ganadores de cada categoría (usuario con más apuntes validados, usuario con más registros y el top Wuolah) se llevan un año de Kaiku Caffè Latte. Por el esfuerzo

**WUOLAH**



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

ventajas

PRO



Di adiós a la publi en los apuntes y en la web



Participa gratis en todos los sorteos



Descarga carpetas completas

estudia sin publi  
WUOLAH PRO

```
.include "inter.inc"
.text

ADDEXC    0x18,  irq_handler
ADDEXC    0x1c,  fiq_handler

@Inicializo la pila en modos FIQ, IRQ y SVC
mov        r0, #0b11010001    @ Modo FIQ,  FIQ&IRQ  desact
msr        cpsr_c,  r0
mov        sp, #0x4000
mov        r0, #0b11010010    @ Modo IRQ,  FIQ&IRQ  desact
msr        cpsr_c,  r0
mov        sp, #0x8000
mov        r0, #0b11010011    @ Modo SVC,  FIQ&IRQ  desact
msr        cpsr_c,  r0
mov        sp, #0x8000000

@Configuro GPIOs 4, 9, 10, 11, 17, 22 y 27 como salida */
ldr        r0, =GPBASE
ldr        r1, =0b00001000000000000000000000000000
str        r1, [r0, #GPFSEL0]
/* guia bits                xx999888777666555444333222111000 */
ldr        r1, =0b00000000000100000000000000000001001
str        r1, [r0, #GPFSEL1]
ldr        r1, =0b00000000000100000000000000000000000
str        r1, [r0, #GPFSEL2]
ldr        r6, =0 @registro que me muestra en què sentido van los leds,
con 0 hacia la derecha y con 1 hacia la izquierda

@ habilito los botones para las interrupciones
mov        r1, #0b000000000000000000000000000001100
str        r1, [r0, #GPFEN0]

@ habilitamos los pulsadores para que pueda ser interrumpido el programa
ldr        r0, =INTBASE

/* guia bits                10987654321098765432109876543210 */
mov        r1, #0b00000000000010000000000000000000000
str        r1, [r0, #INTENIRQ2] @ para pulsadores INTENIRQ2 bit 20

@Programo C1 y C3 para dentro de 2 microsegundos */
ldr        r0, =STBASE
ldr        r1, [r0, #STCLO]
add        r1, #2
str        r1, [r0, #STC1]
str        r1, [r0, #STC3]

@Habilito C1 para IRQ */
ldr        r0, =INTBASE
mov        r1, #0b0010
str        r1, [r0, #INTENIRQ1]

@Habilito C3 para FIQ */
mov        r1, #0b10000011
str        r1, [r0, #INTFIQCON]

@Habilito interrupciones globalmente */
mov        r0, #0b00010011    @ Modo SVC,  FIQ&IRQ  activo
msr        cpsr_c,  r0

bucle:    b          bucle
```

```

irq_handler:    push    {r0, r1, r2,r3,r4,r5}
                ldr     r0, =GPBASE
                @ metemos los dos pulsadores en distintos registros
                mov r5, #0b0000000000000000000000000000100 @pulsador 1

GPIO2          mov r3, #0b00000000000000000000000000001000 @pulsador 2

GPIO3

                ldr r4, [r0, #GPEDS0]
                @comparamos el pulsador GPIO2
                cmp r4, r5
                beq otrosent @si es el pulsador 2 el parpadeo va en
sentido contrario
                @comparamos el GPIO3
                cmp r4, r3 @ si es el pulsador 3 el parpadeo va normal y
seguimos con la secuencia
                bne reg

sent:           ldr r6, =0
                ldr     r1, =cuenta
@Apago todos LEDs    10987654321098765432109876543210
                ldr     r2, =0b000010000100001000001111000000000
                str     r2, [r0, #GPCLR0]
                ldr     r2, [r1]                    @ Leo variable cuenta
                subs    r2, #1                      @ Decremento
                moveq   r2, #25                     @ Si es 0, volver a 25
                str     r2, [r1], #-4               @ Escribo cuenta,
decremento en 4 para coger los leds
                ldr     r2, [r1, +r2, LSL #3] @ Leo secuencia
multiplico el contador por 8 y le sumo la direccion de secuen
                str     r2, [r0, #GPSET0] @ Escribo secuencia en LEDs
                b res

reg:            cmp r6, #1
                beq otrosent
                bne sent

```

```

otrosent:      ldr r6, =1
               ldr      r1, =cuenta
@Apago todos LEDs      10987654321098765432109876543210
               ldr      r2, =0b00001000010000100000111000000000
               str      r2, [r0, #GPCLR0]
               ldr      r2, [r1]                @ Leo variable cuenta
               add      r2, #1                  @ aumento
               cmp      r2, #26
               moveq     r2, #1                  @ Si es 26, vuelvo a 1
               str      r2, [r1], #-4          @ Escribo cuenta,
decremento en 4 para coger los leds
               ldr      r2, [r1, +r2, LSL #3] @ Leo secuencia
multiplico el contador por 8 y le sumo la direccion de secuen
               str      r2, [r0, #GPSET0] @ Escribo secuencia en LEDs

```

16



# LA ÚLTIMA

**AITANA OCAÑA · MIGUEL BERNARDEAU**  
· UNA HISTORIA SOBRE EMPEZAR ·

Temporada completa  
**2 de diciembre** solo en





Reservados todos los derechos.  
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

ventajas

PRO



Di adiós a la publi en los apuntes y en la web



Acumula tickets para los sorteos



Descarga carpetas completas

estudia sin publi  
WUOLAH PRO

```
.word 1706 @ Re
.word 0b10000000000000000000000000000000
.word 1706 @ Re
.word 0b00000100000000000000000000000000
.word 1351 @ Fa#
.word 0b00000000000100000000000000000000
.word 1275 @ Sol
.word 0b00000000000000000100000000000000
.word 1706 @ Re
.word 0b00000000000000000010000000000000
.word 1515 @ Mi
.word 0b00000000000000000000001000000000
.word 1706 @ Re
.word 0b10000000000000000000000000000000
.word 1706 @ Re
```

```
fiq_handler:    ldr    r8, =GPBASE
                ldr    r9, =bitson
@Hago sonar altavoz invirtiendo estado de bitson
                ldr    r10, [r9]
                eors    r10, #1
                str    r10, [r9], #4 @ le sumo 4 para acceder a la
cuenta
@Leo cuenta y luego elemento correspondiente en secuen
                ldr    r10, [r9] @leo cuenta
                ldr    r9, [r9, +r10, LSL #3] @desplazo la cuenta
(multiplico por 8) y le sumo la direccion de secuen
@Pongo estado altavoz seg'n variable bitson
                mov    r10, #0b10000 @ GPIO 4 (altavoz)
                streq   r10, [r8, #GPSET0]
                strne   r10, [r8, #GPCLR0]
@Reseteo estado interrupción de C3
                ldr    r8, =STBASE
                mov    r10, #0b1000
                str    r10, [r8, #STCS]
@Programo retardo seg'n valor leído en array
                ldr    r10, [r8, #STCLO]
                add    r10, r9
                str    r10, [r8, #STC3]
                subs   pc, lr, #4
```