

## Ej14.pdf



GeXx\_



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga

ZERO AZÚCAR  
**#ZERO  
PALABRAS**

DEMASIADO BUENO PARA  
EXPLICARLO CON PALABRAS

*Coca-Cola*  
Real Magic™

REAL MAGIC, COCA-COLA ZERO son marcas registradas de The Coca-Cola Company.



quieres la play quinta??

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



```
.include "inter.inc"
.text
/* Agrego vector interrupci n */
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, =0b0000100000000000000010000000000000
str r1, [r0, #GPFSEL0]
/* guia bits xx99988877766655544433222111000 */
ldr r1, =0b0000000000100000000000000000001001
str r1, [r0, #GPFSEL1]
ldr r1, =0b00000000001000000000000000001000000
str r1, [r0, #GPFSEL2]
/* 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

/* Repetir para siempre */
bucle: b bucle

/* Rutina de tratamiento de interrupci n IRQ */
irq_handler:
push {r0, r1, r2}
ldr r0, =GPBASE
ldr r1, =cuenta
/* Apago todos LEDs 10987654321098765432109876543210 */
ldr r2, =0b00001000010000100000111000000000
str r2, [r0, #GPCLR0]

ldr r2, [r1] @ Leo variable cuenta
subs r2, #1 @ Decremento
moveq r2, #25 @ Si es 0, volver a 25
```

participa  
aqu 



Ser  sorteada  
entre todos los  
usuarios  
estudiantes que  
el d a de la  
finalizaci n del  
concurso est n  
en el top de su  
comunidad

```

        str        r2, [r1], #-4                @ Escribo cuenta y dejo en r1,
cuenta-4(bitson) para que luego coja el primer led (x8 son dos
instrucciones)
        ldr        r2, [r1, +r2, LSL #3] @ Leo secuencia
        str        r2, [r0, #GPSET0]          @ Escribo secuencia en LEDs

/* Reseteo estado interrupci n de C1 */
        ldr        r0, =STBASE
        mov        r2, #0b0010
        str        r2, [r0, #STCS]
/* Programo siguiente interrupci n en 500ms */
        ldr        r2, [r0, #STCLO]
        ldr        r1, =500000                @ 2 Hz
        add        r2, r1
        str        r2, [r0, #STC1]
/* Recupero registros y salgo */
        pop        {r0, r1, r2}
        subs       pc, lr, #4

/* Rutina de tratamiento de interrupci n FIQ */
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
/* Pongo estado altavoz seg n variable bitson */
        mov        r10, #0b10000            @ GPIO 4 (altavoz)
        streq      r10, [r8, #GPSET0]
        strne      r10, [r8, #GPCLR0]
/* Leo cuenta y luego elemento correspondiente en secuen*/
        ldr        r10, [r9]                @leo cuenta
        ldr        r9, [r9, +r10, LSL #3] @elemento de la secuencia
/* Reseteo estado interrupci n de C3 */
        ldr        r8, =STBASE
        mov        r10, #0b1000
        str        r10, [r8, #STCS]
/* Programo retardo seg n valor le do en array */
        ldr        r10, [r8, #STCLO]
        add        r10, r9
        str        r10, [r8, #STC3]

/* Salgo de la FIQ */
        subs       pc, lr, #4

bitson:    .word    0                        @ Bit 0 = Estado del altavoz
cuenta:    .word    25                      @ Entre 1 y 25, LED a encender
secuen:    .word    0b10000000000000000000000000000000
            .word    1275 @ Sol
            .word    0b00000010000000000000000000000000
            .word    1136 @ La
/* guia bits 7654321098765432109876543210 */
            .word    0b00000000000010000000000000000000
            .word    1275 @ Sol
            .word    0b00000000000000000000100000000000
            .word    1012 @ Si

```

```

/* guia bits 7654321098765432109876543210 */
.word 0b00000000000000000000000000000000
.word 956 @ Do'
.word 0b00000000000000000000000000000000
.word 956 @ Do'
.word 0b10000000000000000000000000000000
.word 1515 @ Mi
.word 0b00000010000000000000000000000000
.word 1351 @ Fa#
.word 0b00000000000010000000000000000000
.word 1275 @ Sol
.word 0b00000000000000000000000000000000
.word 1012 @ Si
.word 0b00000000000000000000000000000000
.word 852 @ Re'
.word 0b00000000000000000000000000000000
.word 1706 @ Re
.word 0b10000000000000000000000000000000
.word 1706 @ Re
.word 0b00000010000000000000000000000000
.word 1275 @ Sol
.word 0b00000000000010000000000000000000
.word 1136 @ La
.word 0b00000000000000000000000000000000
.word 1706 @ Re
.word 0b00000000000000000000000000000000
.word 1515 @ Mi
.word 0b00000000000000000000000000000000
.word 1706 @ Re
.word 0b10000000000000000000000000000000
.word 1706 @ Re
.word 0b00000010000000000000000000000000
.word 1351 @ Fa#
.word 0b00000000000010000000000000000000
.word 1275 @ Sol
.word 0b00000000000000000000000000000000
.word 1706 @ Re
.word 0b00000000000000000000000000000000
.word 1515 @ Mi
.word 0b00000000000000000000000000000000
.word 1706 @ Re
.word 0b10000000000000000000000000000000
.word 1706 @ Re

```