PRÁCTICA 8
ENTRADA/SALIDA I

Objetivos:

- Aplicar los conceptos de entrada/salida.

Desarrollo / Comentario:

La arquitectura RISCV utiliza E/S mapeada en memoria, en la que se emplean varias direcciones de memoria para referirse a los puertos de los dispositivos periféricos.

En Ripes la dirección base de la E/S mapeada en memoria es 0xf0000000. Dispone de varios dispositivos de E/S. Matrices de LED, Switches y D-Pad.

Para empezar crearemos un Switch 0 haciendo click con el ratón en Switches y una Matriz de LED 0 haciendo click con el ratón en LED Matrix . El Switch 0 se corresponde con un dispositivo de entrada y sólo se podrá leer de él su valor. La Matriz de Leds se corresponden con direcciones de entrada/salida y se pueden mostrar colores en formato RGB (Tabla de códigos de colores RGB (rapidtables.org)). También se podrán leer valores de la Matriz de LEDs.

El siguiente trozo de programa muestra cómo escribir un LED en rojo en la matriz de LED

```
li a0, LED_MATRIX_0_BASE
li s0, 0xff0000
sw s0, 0(a0)
```

a) Se pide realizar un programa que lea de siwtch 0 y escriba tantos Leds en rojo de la matriz como el número leído del siwtch 0.

```
.text
        li a0, LED_MATRIX_0_BASE      #Cargamos el tamaño de la matriz
        li a1, SWITCHES_0_BASE        #Cargamos el valor del Switch0 (2^n)
        lw s1, 0(a1)                  #Guardamos el valor en s1
        li s0, 0xff0000               #Cargamos el rojo

bucle:
        beq s1,zero,final             #Si s1 = 0 es que en el contador no queda más
        sw s0, 0(a0)                  #Elegimos el color rojo
        addi a0,a0,4                  #Pasamos al siguiente LED (dir. memoria de 4 en 4)
        addi s1,s1,-1                 #Quitamos del contador un LED que pintar
        j bucle                       #Saltamos al bucle para comprobar

final:                                #fin
    li a7,10
    ecall
```

b) Se pide realizar un programa que lea de siwtch 0 y escriba alternativamente en rojo, verde y azul tantos Leds en matriz como el número leído del siwtch 0.

```
.text
li a0, LED_MATRIX_0_BASE        #Cargamos el tamaño de la matriz
li a1, SWITCHES_0_BASE          #Cargamos el valor que tenga Switch0
lw s1, 0(a1)                    #Guardamos este numero (contador)
li s0, 0xff0000                 #Cogemos el rojo
li s2, 0x00ff00                 #Cogemos el verde
li s3, 0x0000ff                 #Cogemos el azul

bucle:
beq s1,zero,final               #Si el contador está a 0, se caba
sw s0, 0(a0)                    #Elegimos el rojo
addi a0,a0,4                    #Pintamos el siguiente de rojo
addi s1,s1,-1                   #Quitamos una del contador
beq s1,zero,final               #Si aún queda por poner, sigue, sino, out
sw s2, 0(a0)                    #Elegimos el verde
addi a0,a0,4                    #Pintamos el siguiente de verde
addi s1,s1,-1                   #Quitamos uno del contador
beq s1,zero,final               #Si aún queda por poner, sigue, sino, out
sw s3, 0(a0)                    #Elegimos el azul
addi a0,a0,4                    #Pintamos el siguiente de azul
addi s1,s1,-1                   #Quitamos uno del contador
beq s1,zero,final               #Si aún queda por poner, sigue, sino, out
j bucle                         #Salto a repetir la lógica del enunciado

                                #En este código si quisiésemos que se
                                #pusiesen los colores las mismas veces
                                #no tendríamos que quitar 1 cada vez
                                #que pinte sino 1 al final

final:                          #fin
li a7,10
ecall
```

Negro	# 000000
Blanco	#FFFFFF
Rojo	# FF0000
Lima	# 00FF00
Azul	# 0000FF
Amarillo	# FFFF00
Cian / Aqua	# 00FFFF
Magenta / Fucsia	# FF00FF
Plata	# C0C0C0
gris	# 808080