

## PRÁCTICA 1 EC

### Apartado 1: GPIO como salida

prog\_gpio\_11 (booleano salida) {(salida: falso = 0 = entrada, verdadero = 1 = salida)

```

    dir_logica = dirLogica (13, dir_base, 0x100)
    entrar en modo supervisor
    acceder a registro de dir_logica
    si (salida = 0) {
        poner entrada en el registro
    }
    si (salida = 1) {
        poner salida en el registro (borrar los unos de esa zona y poner el código de
        salida)
    }
    salir de modo supervisor
}
```

salida	R0 (parámetro), R4
dir_base	R1
dir_logica	R3
valor registro (GPSEL1)	R0
máscara (0x38 para borrado de bits, 0x8 para programar salida, 0 para programar entrada)	R1

valor\_gpio\_11 (booleano corriente) {(corriente: falso = 0 = 0V, verdadero = 1 = 3.3V)

```

    dir_logica = acceso desde la dirección guardada
    entrar en modo supervisor
    si (corriente = 0) {
        acceder a registro de poner 0
        poner 0 en el registro (eliminar los unos de esa zona y escribir uno en el registro)
    }
    si (corriente = 1) {
        acceder a registro de poner 1
        poner 1 en el registro (eliminar los unos de esa zona y escribir uno en el registro)
    }
    salir de modo supervisor
}
```

corriente	R0 (parámetro), R4
dirección guardada	R1
dir_logica	R3
registro (GPCLR0, GPSET0)	R2
valor registro	R0
máscara (0x800)	R1

```

test_driver_gpio_11 {
    prog_gpio_11 (salida: 1)
    encendido = 0
    para (i desde 0 hasta 10) {
        si (encendido = 0) {
            encendido = 1
        }
        sino {
            encendido = 0
        }
        valor_gpio_11 (encendido)
        perder_tiempo (1 seg)
    }
}

```

salida	R0
encendido	R4, R0 (parámetro)
i	R5
perder_tiempo (instrucciones)	R1

perder\_tiempo (1 seg): bucle decrementado de 2000 instrucciones

dirLogica: SWI 0x68 (devuelve en r3)

entrar en modo supervisor: SWI 0x16

salir de modo supervisor: SWI 0x7C

## **Apartado 2: GPIO como entrada**

prog\_gpio\_2 (booleano salida) {(salida: falso = 0 = entrada, verdadero = 1 = salida)}

```

    dir_logica = dirLogica (13, dir_base, 0x100)
    entrar en modo supervisor
    acceder al valor del registro de selección
    si (salida = 0) {
        guardar el valor de entrada en el registro
    }
    si (salida = 1) {
        guardar el valor de salida en el registro
    }
    salir de modo supervisor
}

```

salida	R0 (parámetro), R4
dir_base	R1
dir_logica	R3
valor registro (GPSEL0)	R0
máscara (0x1C0 para borrado de bits, 0x40 para programar salida, 0 para programar entrada)	R1

progEv\_gpio\_2 (booleano descendente) {(descendente: falso = 0 = ascendente, verdadero = 1 = descendente)

```

    dir_logica = accedo desde la dirección guardada
    entrar en modo supervisor
    si (descendente = 0) {
        acceder a registro de detección de evento ascendente
    }
    si (descendente = 1) {
        acceder a registro de detección de evento descendente
    }
    activar la detección del evento (poner un uno en el registro)
    eliminar la detección del evento contrario
    borrar si se ha detectado algún evento (de otras cargas)
    guardar el nuevo valor del registro
    salir del modo supervisor
}

```

descendente	R0 (parámetro), R4
dirección guardada	R0
dir_logica	R3
registro (GPREN0, GPFEN0)	R2. R4
valor registro	R0
máscara (0x4 para borrado y poner bits)	R1

leeEv\_gpio\_2 {

```

    dir_logica = accedo desde la dirección guardada
    entrar en modo supervisor
    coger el registro de detección de eventos
    salir del modo supervisor
    comprobar si tiene el valor que buscamos (tiene un uno en el registro)
    si (tiene evento) {
        devolver 1
    }
    sino {
        devolver 0
    }
}

```

dirección guardada	R0
dir_logica	R3
valor registro	R0
Devolver	R0

clearEv\_gpio\_2 {

```

    dir_logica = accedo desde la dirección guardada
    perder_tiempo (250ms)
    entrar en modo supervisor
    acceder a registro de detección de eventos
    desactivar el evento (poner un uno en el registro)
    salir del modo supervisor

```

}

dirección guardada	R0
dir_logica	R3
perder_tiempo (instrucciones)	R0
valor registro	R0

perder\_tiempo (250 ms): bucle decrementado de 200 instrucciones

test\_driver\_gpio\_2 {

```

    prog_gpio_11 (salida: 1)
    prog_gpio_2 (entrada: 0)
    progEv_gpio_2 (descendente: 0)
    encendido = 0
    para (i desde 0 hasta 5) {
        evento = 0
        mientras (evento = 0) {
            evento = leeEv_gpio_2
        }
        clearEv_gpio_2
        si (encendido = 0) {
            encendido = 1
        }
        sino {
            encendido = 0
        }
        valor_gpio_11 (encendido)
    }

```

}

Parámetros (salida, entrada, descendente, encendido)	R0
evento	R0
encendido	R4
i	R5

### **Preguntas sobre la práctica**

#### **1. Describe los pasos necesarios para programar como salida un pin del dispositivo GPIO**

Para programar un pin como salida hay que encontrar la dirección lógica de los registros del dispositivo GPIO, luego ver que pin se quiere cambiar para saber que registro GPFSEL vamos a usar (0: pines del 0 al 9, 1: pines del 10 al 19, 2: pines del 20 al 29...) y escoger el offset del registro correcto. Se entra al modo supervisor para poder acceder al registro a partir de la dirección lógica base y el offset del registro. Se seleccionarán los bits que hay que cambiar sin variar el resto, para ello se combinarán los bits (con las instrucciones bic y orr) y en esos bits se pondrá 001 que es el código para poner un pin como salida, se vuelve a escribir en el registro que estaba con la misma dirección y se sale de modo supervisor.

#### **2. ¿Qué ocurre si se intenta dar valor al pin GPIO-11 y se ha olvidado realizar la inicialización?**

El pin estará programado como entrada por lo que como para dar un valor de salida, los registros son independientes del registro de entrada y el dispositivo no estará programado como salida, el pin no recibirá ninguna corriente.

#### **3. ¿Se puede producir un evento en un terminal inicializado como salida? En caso afirmativo, indicar la forma de hacerlo.**

Si el pin está programado como salida no se pueden producir eventos, ya que los eventos sólo se pueden producir de nivel y de flanco ascendente o descendente en la entrada.

#### **4. Para la práctica que has realizado, ¿qué consideras más adecuado, utilizar la detección de flanco ascendente o la de descendente? Justifica la respuesta.**

Creo que es mejor utilizar flanco ascendente porque el pulsador deja pasar un 0 cuando se pulsa y está a 1 en reposo, por lo que interesa coger el valor cuando se suelta, es decir, cuando el pulso asciende, para que no tome de nuevo valor cuando se estabiliza.