

EDUARDO MORA GONZÁLEZ
PRACTICA 2
SISTEMAS EMPOTRADOS Y UBICUOS

Ejercicio 1

```
int main(void) {  
  
    /* Write your code here */  
    unsigned int val_sw2 = 0x00000000;  
    unsigned int val_sw3 = 0x00000000;  
  
    //Habilitar el reloj para el puerto GPIO.  
    SIM_SCGC5 |= SIM_SCGC5_PORTA_MASK;  
    SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK;  
    SIM_SCGC5 |= SIM_SCGC5_PORTC_MASK; /  
  
    //Definir en el registro PORTx_PCRn la función GPIO.  
    PORTA_PCR4 = 0x100;  
    PORTB_PCR21 = 0x100;  
    PORTC_PCR6 = 0x100;  
  
    ////Definir en el registro PORTx_PCRn la función GPIO.  
    GPIOA_PDDR |= (0 << 4);  
    GPIOB_PDDR |= (1 << 21);  
    GPIOC_PDDR |= (0 << 6);  
  
    // Modificar los valores de entrada y salida de los registros.  
    GPIOB_PSOR = (1 << 21); /*Apago la luz*/  
    for (;;) {  
        val_sw2 = GPIOC_PDIR & (1 << 6);  
        val_sw3 = GPIOA_PDIR & (1 << 4);  
  
        if (val_sw2 == 0x00000000)  
            GPIOB_PCOR = (1 << 21); /*Enciendo la luz*/  
        if (val_sw3 == 0x00000000)  
            GPIOB_PSOR = (1 << 21); /*Apago la luz*/  
    }  
  
    /* Never leave main */  
    return 0;  
}
```

Como se puede ver en la imagen, se ha dividido el código en 4 bloques el significado de cada bloque es el siguiente:

- 1) Se habilita el control del reloj en los puertos A, B y C.
- 2) Por cada puerto abierto asigno un registro correspondiente a los pulsadores y al led.
- 3) Establezco los distintos bits del puerto como salida.
- 4) Hago las operaciones pertinentes según si se pulsa un switch u otro para encender el led.

Ejercicio 2

```
for (;;) {  
  
    if (BitBoton2_GetVal() == 0)  
        BitLed_PutVal(0);  
  
    if (BitBoton3_GetVal() == 0)  
        BitLed_PutVal(1);  
  
}
```

En este ejercicio hemos definido los dos botones y el led, y a través de los eventos detectamos si ha sido pulsado o no, en el caso de que se pulse el botón 2 se enciende el led, si se pulsa el botón 3 se apaga.

Ejercicio 3

Este ejercicio tiene un fichero Events.c donde el código es:

```
int counter = 0;  
void Pulsador_OnInterrupt(void) {  
    counter++;  
  
    if (counter > 4)  
        counter = 0;  
}  
  
/* END Events */  
  
#ifdef __cplusplus  
} /* extern "C" */  
#endif
```

Definimos una variable compartida para saber el numero de veces que se pulsa el pulsado, como solo nos interesa encender 4 led, cuando el contador llegue a un numero mayor lo reiniciamos a 0.

En el fichero main.c el código es:

```
for (;;) {  
  
    switch (counter) {  
    case 0:  
        BitLed1_PutVal(1);  
        BitLed2_PutVal(1);  
        BitLed3_PutVal(1);  
        BitLed4_PutVal(1);  
        break;  
    case 1:  
        BitLed1_PutVal(0);  
        break;  
    case 2:  
        BitLed2_PutVal(0);  
        break;  
    case 3:  
        BitLed3_PutVal(0);  
        break;  
    case 4:  
        BitLed4_PutVal(0);  
        break;  
    }  
  
}
```

En este caso, obtenemos el valor de la variable compartida del Events.c y depende del valor que tenga encendemos el led o los apagamos todos (cuando se reinicia la variable global a cero).

Ejercicio 4

Este ejercicio tiene un fichero Events.c donde el código es:

```
int opcion;  
void Pulsador2_OnInterrupt(void) {  
    opcion = 2;  
}
```

1

```
void Pulsador1_OnInterrupt(void) {  
    opcion = 1;  
}
```

2

```
int segundos;  
void Timer1_OnInterrupt(void) {  
  
    segundos++;  
    if (segundos % 10 == 0)  
        segundos = 0;  
}
```

3

El código tiene 3 bloques distintos:

- 1) En este bloque se define que se ha pulsado el segundo pulsador por eso en la variable compartida opción se le asigna el valor a 2.
- 2) Igual que en el anterior, pero en este caso se le asigna el valor 1.
- 3) Este bloque es el encargado de definir un contador del tiempo hasta el número 10, cuando pasan 10 segundos se reinicia, por eso la variable compartida se incrementa cada segundo.

En el fichero main.c el código es:

```
for (;;) {  
    if (opcion == 2) {  
        Timer1_OnInterrupt;  
  
        if (segundos <= 5) {  
            VERDE_PutVal(1);  
            ROJO_PutVal(0);  
        }  
        if (segundos <= 6)  
            VERDE_PutVal(0);  
  
        if (segundos > 6)  
            ROJO_PutVal(1);  
    }  
  
    if (opcion == 1) {  
        ROJO_PutVal(1);  
        VERDE_PutVal(1);  
        WAIT_Waitms(500);  
        ROJO_PutVal(0);  
        VERDE_PutVal(0);  
        WAIT_Waitms(500);  
    }  
}
```

1

2

El código lo podemos dividir en dos bloques, dependiendo del pulsador que se presione:

- 1) Este bloque tiene relacion con el 3º bloque del codigo de Events.c, por eso dependiendo del valor que tenga en cada momento la variable segundos se realiza una opcion u otra (bloque verde).
- 2) Este bloque se usa Wait (bloque verde) donde nos va a permitir la intermitencia de encender y apagar pasado un tiempo.

Ejercicio 5

Este ejercicio tiene un fichero Events.c donde el código es:

```
int counter = 1;
int PotValores = 0;
int Maximo = 65535;

void Pulsador2_OnInterrupt(void) {

    if (counter >= 4) {
        PotValores = 0;
        counter = 1;
    } else {
        PotValores = (Maximo / 3) * counter;
        counter++;
    }

}

void Pulsador1_OnInterrupt(void) {

    if (counter <= 0) {
        counter = 1;
        PotValores = 0;
    } else {
        counter--;
        PotValores = (Maximo / 3) * counter;
    }

}
```

El código tiene 2 bloques distintos (dependiendo del pulsador que se presione):

- 1) En este bloque se obtiene el valor de la intensidad actual y se incrementa, en el caso de que llegue al máximo, se apaga.
- 2) Igual que el anterior, pero esta vez se calcula el valor para decrementar la intensidad.

En el fichero main.c el código es:

```
for (;;) {

    PWM1_SetRatio16(PotValores);

}
```

Con la variable compartida de Events.c se asigna el valor de intensidad que va a tener el Led.

Ejercicio 6

Este ejercicio tiene un fichero Events.c donde el código es:

```
int Pulso = 0;

void Pulsador1_OnInterrupt(void) {
    Pulso = Pulso + 1;
}
```

Cuando se pulsa se incrementa la variable compartida.

En el fichero main.c el código es:

```
unsigned int tiempo_1;
unsigned int tiempo_2;

for (;;) {
    if (Pulso == 1) {
        printf("\nINICIO DEL CONTADOR\n");
        Cont_Reset();
        Cont_GetTimeMS(&tiempo_1);
    }
    if (Pulso == 2) {
        Cont_GetTimeMS(&tiempo_2);
        printf("\nEL TIEMPO ENTRE LAS PULSACIONES HA SIDO: %d \n",
            (tiempo_2 - tiempo_1));
        Pulso = 0;
    }
}
```

El código tiene 2 bloques distintos (dependiendo del valor de la variable compartida Pulso):

- 1) En este bloque se reinicia el contador y se empieza a contar.
- 2) En este bloque se obtiene el valor actual y se muestra por pantalla el tiempo transcurrido entre el primer tiempo y el segundo. Finalmente se reinicia la variable compartida pulso a 0.