

Práctica 1. Manejo básico del Connected Launchpad de Texas Instruments

1. Objetivo

En esta primera práctica de la parte digital, perseguimos varios objetivos:

- Conocer el sistema de desarrollo del microcontrolador estudiado
- Realizar algunos ejemplos sencillos de manejo del mismo
- Aprender a manejar la documentación de la librería DriverLib.

2. Material necesario

- Connected Launchpad de Texas Instruments con el microcontrolador TIVA TM4C1294NCPDT
- Manual de la librería de funciones DriverLib y datasheet del microcontrolador

3. Fundamento teórico

Para la realización de la práctica hará falta conocer el funcionamiento de los diferentes periféricos explicados en clase. Aparte de esto, se usarán las funciones de configuración y manejo de las interrupciones de los pines de entrada/salida. Para ello, habrá que buscar información de las siguientes funciones (presentes en gpio.c / gpio.h):

```
extern void GPIOIntTypeSet(uint32_t ui32Port, uint8_t ui8Pins, int32_t ui32IntType);
extern void GPIOIntEnable(uint32_t ui32Port, uint32_t ui32IntFlags);
extern uint32_t GPIOIntStatus(uint32_t ui32Port, bool bMasked);
extern void GPIOIntClear(uint32_t ui32Port, uint32_t ui32IntFlags);
extern void GPIOIntRegister(uint32_t ui32Port, void (*pfnIntHandler)(void));
```

Adicionalmente, se deberán usar las siguientes funciones generales de manejo de interrupciones (presentes en interrupt.c/interrupt.h):

```
extern bool IntMasterEnable(void);
extern bool IntMasterDisable(void);
extern void IntEnable(uint32_t ui32Interrupt);
extern void IntDisable(uint32_t ui32Interrupt);
```

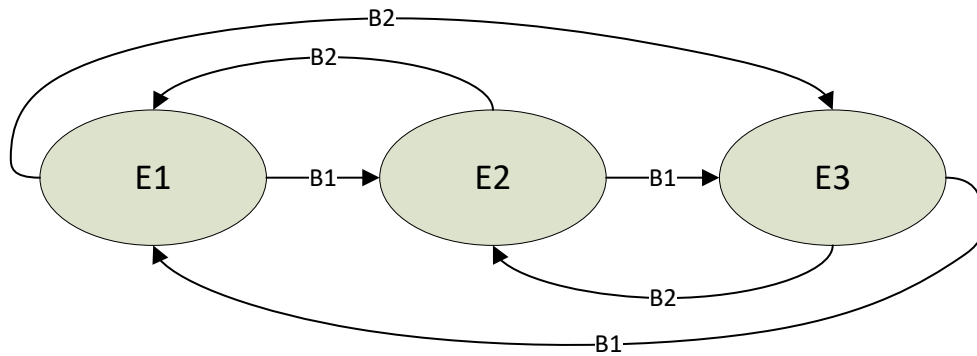
Dado que este curso vamos un poco retrasados en la teoría, deberemos explicar en la práctica el manejo de los pines y de las interrupciones.

4. Realización de la práctica

I. Primer ejemplo básico

Realizar un primer ejemplo básico, desde cero, que realice un programa en el cual se haga lo siguiente:

- Deberá recorrer los modos 1, 2 y 3 según se pulsen los botones B1 y B2, atendiendo al siguiente diagrama de modos de funcionamiento:



- En el modo 1, Empezará parpadeando los 4 leds de la placa, con un periodo de 1s y un duty cycle del 10% (0.1s encendido, 0.9s apagado).
- En el modo 2 realizará la siguiente secuencia: empezando por todos los leds apagados, los irá encendiendo, esperando 1s entre uno y otro, y cuando llegue al cuarto, esperará 3s antes de volver al principio.
- En el modo 3, la secuencia que se pretende de encendido y apagado será 1010-0101, con un intervalo de 500ms entre uno y otro

Para el desarrollo de este primer proyecto, se puede usar como base el ejemplo 2. Como no dio tiempo a hacerlo en clase, se comenzará por cargar el ejemplo y probar su funcionamiento, explicando el manejo del mismo.

II. Segundo Ejemplo: con interrupciones

En este segundo ejemplo, para evitar la pérdida de pulsaciones, se rehará el código usando interrupciones, para poder atender a los botones de manera inmediata. Básicamente, el código será el del proyecto anterior, al que habrá que añadir lo siguiente:

- Configurar (antes del bucle central) los pines de interrupciones
- Apuntar a la rutina de interrupción que se vaya a usar (registrarla)
- Habilitar la interrupción del puerto de entrada J
- Habilitar globalmente las interrupciones (no necesario)

Aparte, fuera de la función `main()` habrá que definir una función que será la que atienda a la interrupción, y que se debe definir como `void Funcion(void)`.

En la rutina de interrupción, se deberán hacer dos cosas:

- Comprobar qué botón se pulsó (con la función `GPIOIntStatus`, por ejemplo)
- Incrementar o decrementar el estado
- Borrar la interrupción pendiente (con la función `GPIOIntClear`)