

Practica 4:

Interrupciones

Hecho por Vladislav Kravchenko

Durante la realización de la practica 4, he trabajado con interrupciones, para realizar diversas modificaciones a la practica 3. Para su correcto funcionamiento he modificado la practica 3, eliminando varias funciones (void muestrabin_o_bcd (); void isp () ;), dentro de ellas la que realizaba la temporización de la practica 3, la función **delay ()**. Esta función la he sustituido por la función **SysTick_Handler ();** la cual he configurado de tal forma que nos interrumpa cada 1ms.

```
SysTick->LOAD=1000000; //1ms
NVIC_SetPriority(SysTick_IRQn,0);
SysTick->VAL= 0;
SysTick->CTRL=0x7;
```

Pero antes de todo esto, lo primero que he realizado es la configuracion de los pines, para definir como entrada los nuevos botones de la placa LPC1768 que se utilizan en esta practica. Despues de esto, con el registro LPC_PINCON->PINSELX, he configurado las tres interrupciones que se nos piden y las he configurado, con los registros que se presentan en el guion de la practica 4. Estos registros son:

- NVIC_EnableIRQ(IRQn_Type IRQn);
- NVIC_SetPriorityGrouping(uint32_t PriorityGroup);
- NVIC_SetPriority(IRQn_Type IRQn,uint32_t priority);
- EXTMODE, ETPOLAR y EXTINT

Para la correcta representacion de los numero primos, con el tiempo asignado al SysTick, he implementado en el codigo del **SysTick_Handler ();** contadores. Con ellos represento la temporizacion prevista de 125ms para los LEDS de la placa LD2 y LD3; y 400ms para el display de 7 segmentos. Toda la representacion del numero primo dura 2 segundos, como en la practica anterior.

En esta practica 4 he partido de la practica anterior, y he añadido dos funciones adicionales para calcular y representar los numeros primos. Para calcular los numeros primos tengo la función **calculaprimo ();**. A esta funcion le he pasado el codigo que tenia en la practica 3, en el main. Para realizar el calculo de numeros primos de arriba - abajo o viceversa he necesitado duplicar este codigo, separados en diferentes for, que tienen diferentes limites de funcionamiento, del minimo al maximo, y del maximo al minimo. Dentro de esta función, cuando se obtiene el número primo, el siguiente paso es representar ese número. Para eso he creado una funcion que hace esta tarea, con el nombre de **representa ();**.

En esta función **representa ();** lo primero que se mira es si el número que se va a representar lo va a hacer en binario o BCD, esto es mirar el valor de la variable asociada a la interrupcion que realiza el cambio de representacion entre binario y BCD, la interrupcion EINT1.

En esta practica se nos pide añadir la funcionalidad a los botones de la Placa MiniDK2. En ella tenemos los botones ISP, KEY1, KEY2 y RESET.

- El pulsador ISP, asociado a la ISR EINT0, debe realizar la funcion de cambiar entre la representacion en binario y BCD. Para eso he escrito el siguiente codigo:

```
void EINT0_IRQHandler(){
    LPC_SC->EXTINT|=0x00000001; //limpiar el flag
    binbcd^=1;
}
```

- El pulsador KEY1, asociado a la ISR EINT1, debe realizar la función de coger el límite inferior cuando se produzca un flanco de bajada y el límite superior cuando se produzca un flanco de subida. Para ello he escrito el siguiente código:

```
void EINT1_IRQHandler() {
    LPC_SC->EXTINT|=(1<<1); //limpiar el flag

    if((LPC_SC->EXTPOLAR &(1<<1))==0) {
        minimo=leedato();
        LPC_SC->EXTPOLAR |=(1<<1); //flanco de bajada = numero minimo
    }
    else{
        maximo=leedato();
        LPC_SC->EXTPOLAR &=~(1<<1); // flanco de subida = numero maximo
    }
}
```

- El pulsador KEY2, asociado a la ISR EINT2, debe realizar la función de cambiar el sentido de la representación de los números primos, en orden ascendente o descendente. Para ello he escrito el siguiente código:

```
void EINT2_IRQHandler() {
    LPC_SC->EXTINT|=0x00000004; //limpiar el flag
    arr_abaj^=1;
}
```

Durante la realización de la práctica he tenido un problema el cual consiste en que había descargado de la Blackboard la práctica 4, y trabajaba con ella desde el escritorio de mi ordenador. Y en los últimos días antes de entregar la práctica me di cuenta de que mi práctica 4, no era la misma que la práctica 4 subida en la BlackBoard. Le adjunto un pantallazo de la práctica 4 errónea:

- Cuando se produzca un flanco de bajada en el pin 12 del conector P3, (o se pulse SW3, pulsador marcado con **KEY1**) se conmutará la salida en el conector P2 entre binario y BCD (si en el momento del flanco se encuentra en binario, se pasará a BCD y viceversa).
- Cuando se produzca un flanco de bajada en el pin 11 del conector P3 se cogerá el valor mínimo desde el que se debe realizar la búsqueda de números primos (que se encontrara en los pines descritos en la práctica anterior), y cuando se produzca un flanco de subida en dicho pin, el valor que se obtenga en los pines será considerado como el valor máximo de la búsqueda. Se garantiza en las especificaciones que nunca habrá dos flancos en un tiempo inferior a 50 mseg. y que siempre, tras un reset o una pulsación de SW2 (**ISP**), se cogerá en primer lugar el valor mínimo y, posteriormente, el valor máximo, en el caso de que se modifique el valor por defecto (65535).
- Cuando se produzca un flanco de bajada pin 10 del conector P3 (pulsador **KEY2**), se cambiará el orden en el que se calculan los números primos. Es decir, si en un momento determinado se están calculando y mostrando los números primos desde el 2 hasta el 65535 en orden ascendente, tras el flanco de bajada se calcularán en orden descendente (del 65535 al 2).
- La temporización se deberá realizar mediante la programación del Timer SYSTICK y no mediante la función delay() que se utilizaba en la práctica anterior.

La cual se distingue de la práctica 4 subida en la BlackBoard.