

Informe primera práctica: EXTINT

Grupo SDMI 22 E

David Guerra

Roman Valls

Daniel Clemente

Se nos pide detectar una interrupción externa por el pin EXTINT de la placa. Para ello necesitamos montar un circuito externo que la active mediante un pulsador. La RSI incrementa un contador que se visualiza mediante los 8 leds de la placa.

Hemos usado la EXTINT (p2.2) y la INT07.

Montaje del circuito externo:

Descripción de conexionado:

Tanto la alimentación como la toma de tierra (GND) han sido tomadas de JP1 (Vcc patillas 1, 2, 39, 40 y GND patillas 19, 20, 21, 22). El bloque formado por condensador y pulsador ha sido montado en paralelo, uno de los extremos de este bloque está conectado a tierra, y el otro a la entrada de un trigger (patilla 1 del integrado 40106) y a una resistencia que a su vez tiene el otro extremo conectado a Vcc. La salida de la puerta en histéresis (patilla 2 del integrado 40106) está conectada al puerto 2.2 del JP1 (patilla 28).

Funcionamiento del circuito antirebotes:

Mientras el interruptor no esta conectado, el condensador C se carga a Vcc, esto provoca que a la entrada de la puerta en histéresis haya un "1" lógico, y como el trigger tiene la salida negada no tendremos ninguna petición de interrupción; si en cambio se pulsa el interruptor, el condensador C se descargará rápidamente a través del cortocircuito provocado por el interruptor, entonces la entrada del trigger habrá quedado a "0" lógico, y al estar la salida negada esto provocará una petición de interrupción.

En realidad la petición de interrupción no se genera cuando hay un "0" en la entrada del trigger, sino cuando la tensión en la entrada del trigger supera una tensión de referencia, según el manual del 40106 para una tensión de alimentación de 5 V. La entrada necesaria para

activar a "1" la salida del trigger debe ser superior a 2V (mínimo) o 3,5V (máximo), esto se hace para evitar que las oscilaciones (rebotes) provocados al apretar el pulsador no generen "falsas interrupciones".

Parte software:

stirq_ex.a96:

Este fichero ensamblador define las direcciones de algunos registros y del VI. Nuestra RSI de la INT07 está en la dirección D0E0h tal como se indica en el documento entorn2.pdf (pág 9).

Además se define la RSI que salva y restaura el contexto (PUSHA (...) POPA) y llama a la función en C.

```
EXTINT:
    PUSHA
    LCALL    RSIExtInt
    POPA
    RET
```

```
cseg at 0d0e0h
    br EXTINT
```

```
END
```

irq_ext.c:

RSIExtInt() incrementa en uno el contador. Deshabilitamos las interrupciones y las volvemos a habilitar posteriormente para ejecutar conta++ de forma atómica.

```
void RSIExtInt()
{
    asm{di;}
    conta++;
    asm{ei;}
}
```

Por el pin2.2 pueden llegar dos interrupciones, la INT07 y la INT13. En el main activamos la INT07 y desactivamos la INT13 porque no queremos que estén presentes ambas. Con ioc1 seleccionamos el pin 2.2 de la placa. Mediante un bucle infinito, mandamos a los 8 leds el valor de la variable.

```
void main()
{
    unsigned char delay;
```

```

conta=0xAA;
asm {di;}
wsr=0;                                //seleccionamos la ventana
int_mask=int_mask|0x80;               //activamos la interrupcion (EXTINT)
int_mask1=int_mask1&0xDF;             // Desactivamos INT13 (EXT_PIN=0)

wsr=0;
ioc1=ioc1 & 0xFD;                     //EXT_INT=0 para elegir p2.2


asm {ei;}

while(1)
{
    wsr=0;
    ioport1=conta;
}
}

```