



3_3_INTERRUPTIONES

Ingeniería mecatrónica

Programación de sistemas de embebidos

Ivan Alejandro Pasillas Gonzalez

Marco teórico

En el contexto de la informática, una interrupción (del inglés interrupt request, en español «petición de interrupción») es una señal recibida por el procesador de una computadora, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación. Una interrupción es una suspensión temporal de la ejecución de un proceso, para pasar a ejecutar una subrutina de servicio de interrupción, la cual, por lo general, no forma parte del programa, sino que pertenece al sistema operativo o al BIOS. Una vez finalizada dicha subrutina, se reanuda la ejecución del programa. Las interrupciones son generadas por los dispositivos periféricos habilitando una señal del CPU (llamada IRQ del inglés "interrupt request") para solicitar atención del mismo. Por ejemplo, cuando un disco duro completa una lectura solicita atención al igual que cada vez que se presiona una tecla o se mueve el ratón. La primera técnica que se empleó para esto fue el polling, que consistía en que el propio procesador se encargara de sondear los dispositivos periféricos cada cierto tiempo para averiguar si tenía pendiente alguna comunicación para él. Este método presentaba el inconveniente de ser muy ineficiente, ya que el procesador consumía constantemente tiempo y recursos en realizar estas instrucciones de sondeo. El mecanismo de interrupciones fue la solución que permitió al procesador desentenderse de esta problemática, y delegar en el dispositivo periférico la responsabilidad de comunicarse con él cuando lo necesitara. El procesador, en este caso, no sondea a ningún dispositivo, sino que queda a la espera de que estos le avisen (le "interrumpan") cuando tengan algo que comunicarle (ya sea un evento, una transferencia de información, una condición de error, etc).

Introducción:

En esta práctica se hará un circuito interrumpido con la tarjeta psoc el cual constara de la siguiente manera, en su fase normal hará una secuencia de 10 leds iluminando consecutivamente repitiendo el patrón una y otra vez al igual que el lcd mostrará con recuadros los leds que se iluminan y en otra fase se presionara un botón el cual interrumpirá la secuencia y con un relevador haremos que encienda un foco de 110v al soltar el botón la secuencia seguirá en donde se quedo

(por problemas con el teléfono perdí varias imágenes de practicas pero mostrare las imágenes de como quedo estructurado el circuito y su código)

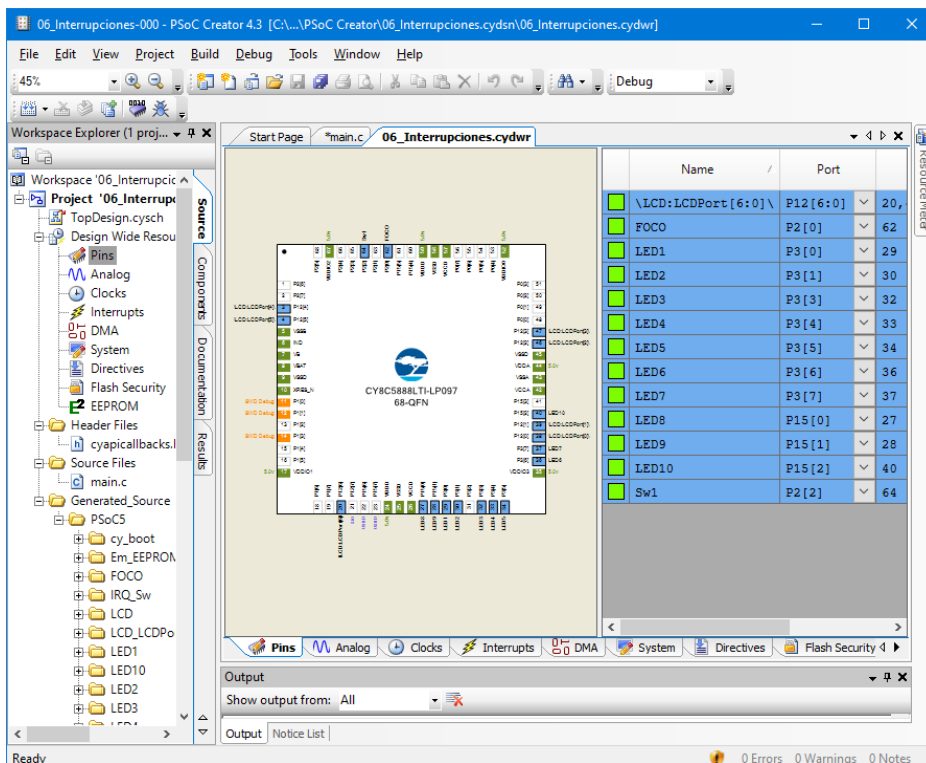


figura 1(distribución de pines)

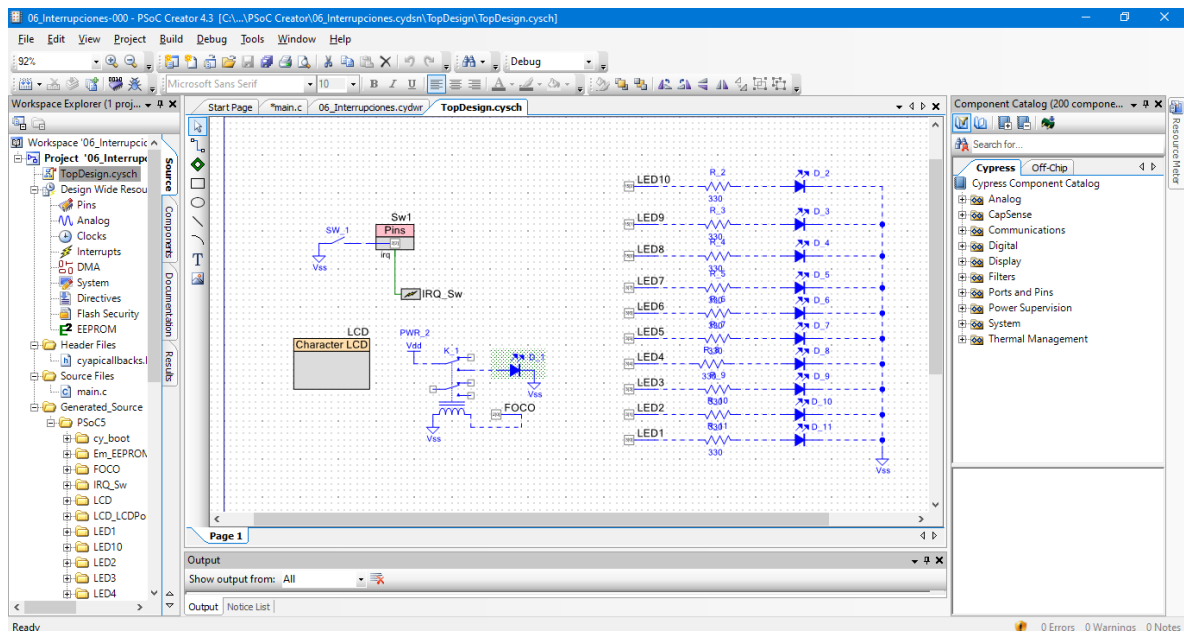


Figura 2: (top desing)

Código de programación:

```
#include "project.h"
```

```
CY_ISR(Interrupcion){
```

```
    FOCO_Write(~FOCO_Read());
```

```
    Sw1_ClearInterrupt();
```

```
}
```

```
int main(void)
```

```
{
```

```
    void LCD_BG();
```

```
    CyGlobalIntEnable; /* Enable global interrupts. */
```

```
IRQ_Sw_StartEx(Interrupcion);
```

```
/* Place your initialization/startup code here (e.g. MyInst_Start()) */
```

```
LCD_Start();
```

```
LCD_ClearDisplay();
```

```
LCD_Position(0,5);
```

```
for(;;)
```

```
{
```

```
    if(FOCO_Read())
```

```
    {
```

```
        LCD_ClearDisplay();
```

```
        LCD_Position(0,0);
```

```
        LCD_PrintString("LEDS ON");
```

```
        LCD_Position(1,0);
```

```
        LCD_DrawHorizontalBG( 1, 0, 1, 160);
```

```
        LED1_Write(1);
```

```
        CyDelay(500);
```

```
        if(FOCO_Read())
```

```
        {
```

```
            LCD_ClearDisplay();
```

```
            LCD_Position(0,0);
```

```
            LCD_PrintString("LEDS ON");
```

```
LCD_Position(1,0);

LCD_DrawHorizontalBG( 1, 0, 2, 160);

LED2_Write(1);

CyDelay(500);

}else

{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);

    CyDelay(500);

}
```

```
if(FOCO_Read())
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS ON");

    LCD_Position(1,0);

    LCD_DrawHorizontalBG( 1, 0, 3, 160);

    LED3_Write(1);

    CyDelay(500);
}else
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);
```

```

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);

    CyDelay(500);
}

if(FOCO_Read())
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS ON");

    LCD_Position(1,0);

    LCD_DrawHorizontalBG( 1, 0, 4, 160);

    LED4_Write(1);

    CyDelay(500);

}else
{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

```



```
    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);

    CyDelay(500);
}

if(FOCO_Read())
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS ON");

    LCD_Position(1,0);

    LCD_DrawHorizontalBG( 1, 0, 5, 160);

    LED5_Write(1);

    CyDelay(500);

}else
{

    LCD_ClearDisplay();

    LCD_Position(0,0);
```

```
LCD_PrintString("LEDS Off");

LCD_Position(1,0);

LCD_PrintString("-----");

LED1_Write(0);

LED2_Write(0);

LED3_Write(0);

LED4_Write(0);

LED5_Write(0);

LED6_Write(0);

LED7_Write(0);

LED8_Write(0);

LED9_Write(0);

LED10_Write(0);

CyDelay(500);

}

if(FOCO_Read())

{

LCD_ClearDisplay();

LCD_Position(0,0);

LCD_PrintString("LEDS ON");

LCD_Position(1,0);

LCD_DrawHorizontalBG( 1, 0, 6, 160);

LED6_Write(1);
```

```
CyDelay(500);}

else

{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);

    CyDelay(500);

}

if(FOCO_Read())

{

    LCD_ClearDisplay();
```

```
LCD_Position(0,0);

LCD_PrintString("LEDS ON");

LCD_Position(1,0);

LCD_DrawHorizontalBG( 1, 0, 7, 160);

LED7_Write(1);

CyDelay(500);}

else

{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);
```

```

        CyDelay(500);
    }

    if(FOCO_Read())
    {
        LCD_ClearDisplay();

        LCD_Position(0,0);

        LCD_PrintString("LEDS ON");

        LCD_Position(1,0);

        LCD_DrawHorizontalBG( 1, 0, 8, 160);

        LED8_Write(1);

        CyDelay(500);}

    else

    {

        LCD_ClearDisplay();

        LCD_Position(0,0);

        LCD_PrintString("LEDS Off");

        LCD_Position(1,0);

        LCD_PrintString("-----");

        LED1_Write(0);

        LED2_Write(0);

        LED3_Write(0);

        LED4_Write(0);

        LED5_Write(0);

```

```

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);

    LED9_Write(0);

    LED10_Write(0);

    CyDelay(500);
}

if(FOCO_Read())
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS ON");

    LCD_Position(1,0);

    LCD_DrawHorizontalBG( 1, 0, 9, 160);

    LED9_Write(1);

    CyDelay(500);}

else
{
    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");
}

```

```
    LED1_Write(0);  
    LED2_Write(0);  
    LED3_Write(0);  
    LED4_Write(0);  
    LED5_Write(0);  
    LED6_Write(0);  
    LED7_Write(0);  
    LED8_Write(0);  
    LED9_Write(0);  
    LED10_Write(0);  
    CyDelay(500);  
}  
  
if(FOCO_Read())  
{  
    LCD_ClearDisplay();  
    LCD_Position(0,0);  
    LCD_PrintString("LEDS ON");  
    LCD_Position(1,0);  
    LCD_DrawHorizontalBG( 1, 0, 10, 160);  
    LED10_Write(1);  
    CyDelay(500);}  
  
else  
{
```

```
LCD_ClearDisplay();

LCD_Position(0,0);

LCD_PrintString("LEDS Off");

LCD_Position(1,0);

LCD_PrintString("-----");

LED1_Write(0);

LED2_Write(0);

LED3_Write(0);

LED4_Write(0);

LED5_Write(0);

LED6_Write(0);

LED7_Write(0);

LED8_Write(0);

LED9_Write(0);

LED10_Write(0);

CyDelay(500);

}

if(FOCO_Read())

{

LCD_ClearDisplay();

LCD_Position(0,0);

LCD_PrintString("LEDS ON");

LCD_Position(1,0);
```



```
LCD_PrintString("");

LED1_Write(0);

LED2_Write(0);

LED3_Write(0);

LED4_Write(0);

LED5_Write(0);

LED6_Write(0);

LED7_Write(0);

LED8_Write(0);

LED9_Write(0);

LED10_Write(0);

CyDelay(500);}

else

{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);
```

```
        LED5_Write(0);

        LED6_Write(0);

        LED7_Write(0);

        LED8_Write(0);

        LED9_Write(0);

        LED10_Write(0);

        CyDelay(500);

    }

}

else

{

    LCD_ClearDisplay();

    LCD_Position(0,0);

    LCD_PrintString("LEDS Off");

    LCD_Position(1,0);

    LCD_PrintString("-----");

    LED1_Write(0);

    LED2_Write(0);

    LED3_Write(0);

    LED4_Write(0);

    LED5_Write(0);

    LED6_Write(0);

    LED7_Write(0);

    LED8_Write(0);
```

```
        LED9_Write(0);

        LED10_Write(0);

        CyDelay(500);

    }

}

/* [] END OF FILE */
```

Conclusión:

En la práctica se observó como una interrupción en el caso de esta detiene la secuencia, pero no la reinicia, esto es como lo que pasa en las computadoras cuando se mueven archivos y se interrumpe la acción, se detiene mas no se reinicia y se puede reanudar desde el punto donde se quedó, es una función importante para realizar procesos complejos