**3\_3\_Interrupciones**



**Programación De Sistemas Embebidos**

Mecatrónica 8°A

**Maestro**: Moran Garabito Carlos

**Alumno:**

* Eduardo Robles Vázquez

**3\_3\_Interrupciones**

**OBJETIVOS**: El alumno deberá realizar la programación de la tarjeta CY8CKIT-059 PSoC para que cumpla las siguientes condiciones:

1. Al presionar un botón prender el foco.
2. Al presionar nuevamente el botón apagar el foco y prender la LCD y los leds. Y así sucesivamente.
3. Los leds al estar activados se prenderán desde el led 1 hasta el led 10, luego se apagarán y empezarán de nuevo.
4. La LCD en la primera fila tendrá el estado de los leds si están prendidos o apagados mientras que en la segunda línea al estar encendidos se deberán encender las celdas en el mismo orden que los leds.

**MARCO TEÓRICO:**

**Interrupciones**

Un procesador embebido normalmente tiene que:

* Medir tiempos
* Generar eventos basados en tiempo
* Responder en tiempo real a eventos que ocurren en tiempos impredecibles

Un sistema embebido tiene, generalmente, que atender varias “tareas”, algunas de ellas periódicas, otras disparadas por eventos. Las interrupciones y los timers son herramientas clave para lograr un manejo efectivo del tiempo y los eventos en un sistema embebido.

En la programación estándar, en una PC, las interrupciones son un recurso del hardware que se usa a nivel del S.O., el kernel, los drivers, lo que llamamos “bajo nivel”. En un µC la programación es siempre de “bajo nivel”. Programamos directamente sobre el CPU, y no sobre la “máquina virtual” del S.O. Asimismo, en la programación normal de la PC, bajo el S. O., existen los threads. El equivalente en el uC son las interrupciones.

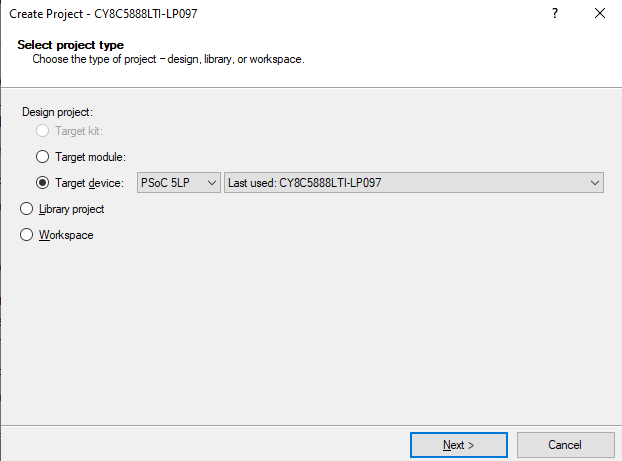
Las interrupciones son un recurso esencial de los sistemas embebidos. Básicamente, la interrupción es un mecanismo mediante el cual el CPU puede, ante cierto evento, suspender lo que está haciendo en ese momento y pasar a atender una rutina de alta prioridad. Una vez finalizada ésta, el CPU vuelve a su actividad anterior.

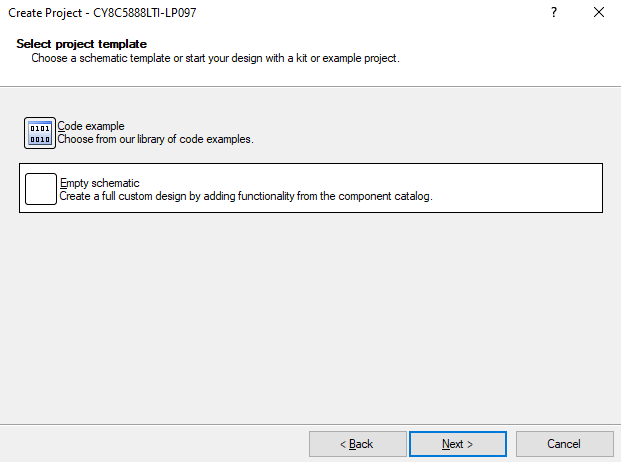
**MATERIALES:**

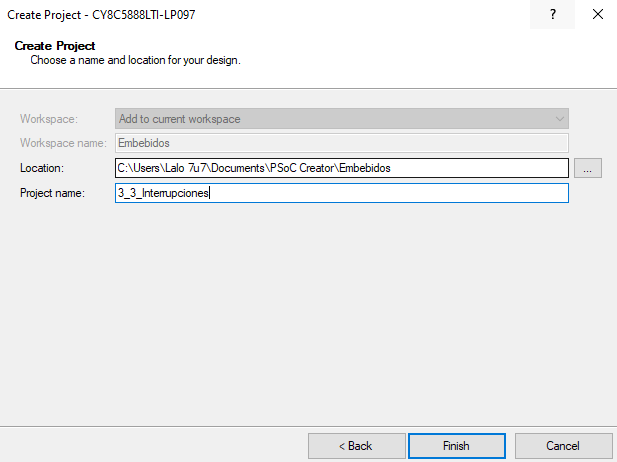
1. Computadora con software PSoC Creator.
2. Tarjeta CY8CKIT-059 PSoC
3. Potenciómetro
4. Resistencias
5. LCD
6. Foco
7. Relevador

**PROCEDIMIENTO:**

1. En el software PSoC Creator crear un nuevo proyecto con las características necesarias para programar la tarjeta CY8CKIT-059 PSoC.





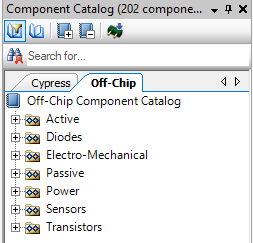


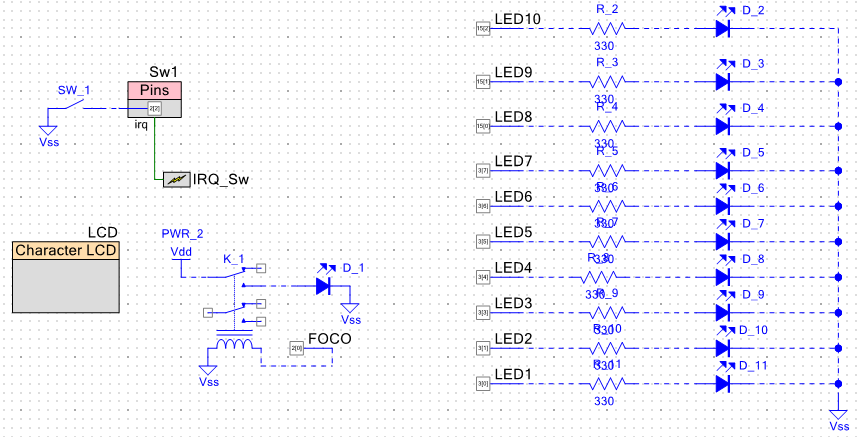
1. Insertar los componentes necesarios para su posterior uso. Una vez hecho esto se puedo construir el proyecto sin problema alguno.

* Primero ingresamos a TopDesign a través del menú que se encuentra en la izquierda.

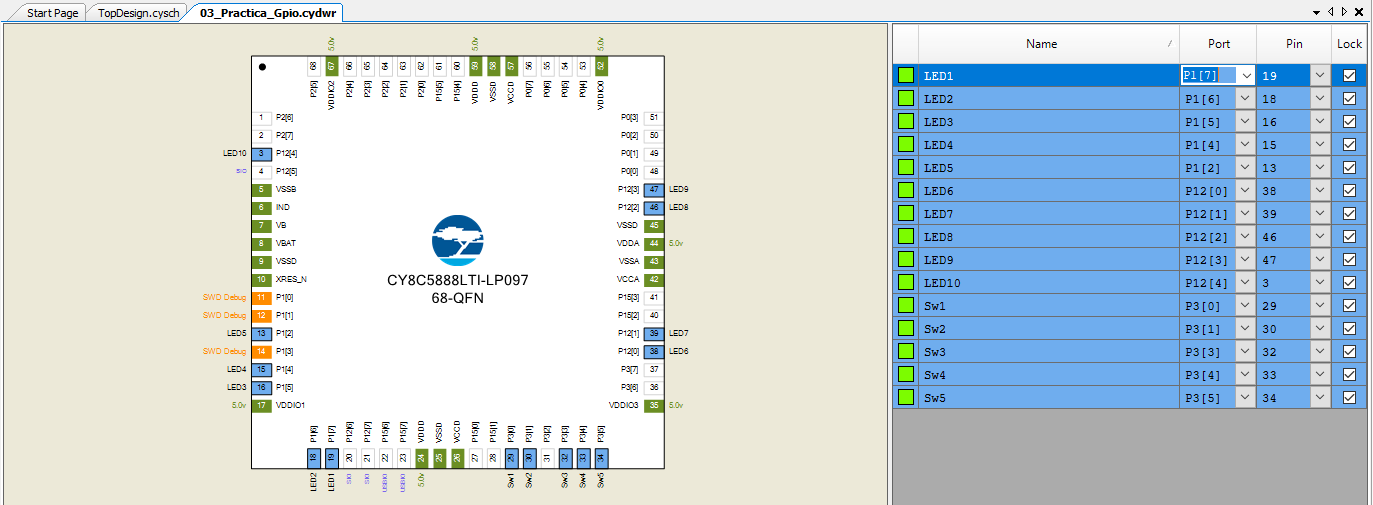
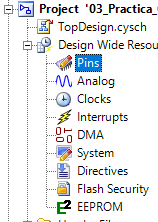


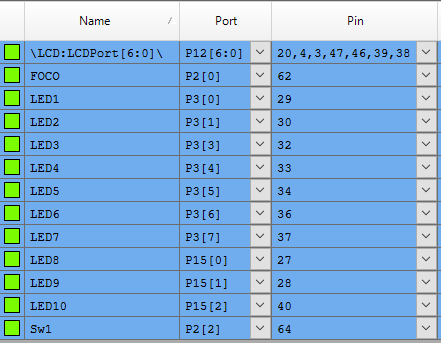
* En el menú que se encuentra a la derecha podremos escoger los componentes que necesitemos.



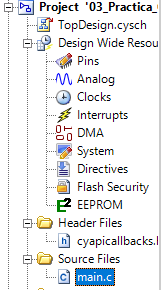


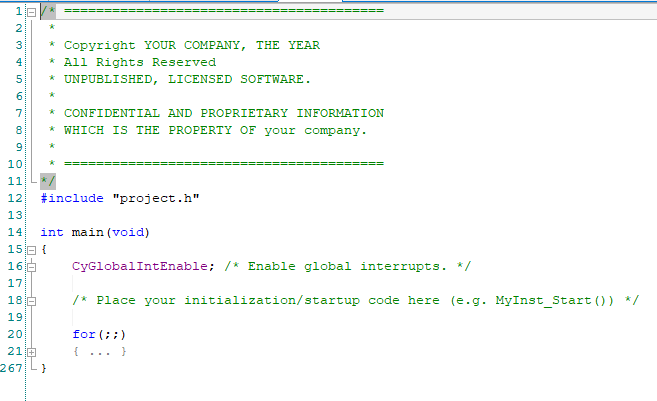
1. Definir los pines que usaremos para cada componente. Este lo haremos ingresando al menú dando clic en Pins.





1. Procedemos a realizar la programación ingresando al Main.c



* Cuando ingresamos al main se nos muestra la siguiente ventana donde escribiremos nuestro programa dentro del For.
* A continuación, se muestra un fragmento del código utilizado

#include "project.h"

CY\_ISR(Interrupcion){

FOCO\_Write(~FOCO\_Read());

Sw1\_ClearInterrupt();

}

int main(void)

{

void LCD\_BG();

CyGlobalIntEnable;

IRQ\_Sw\_StartEx(Interrupcion);

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);

}

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);

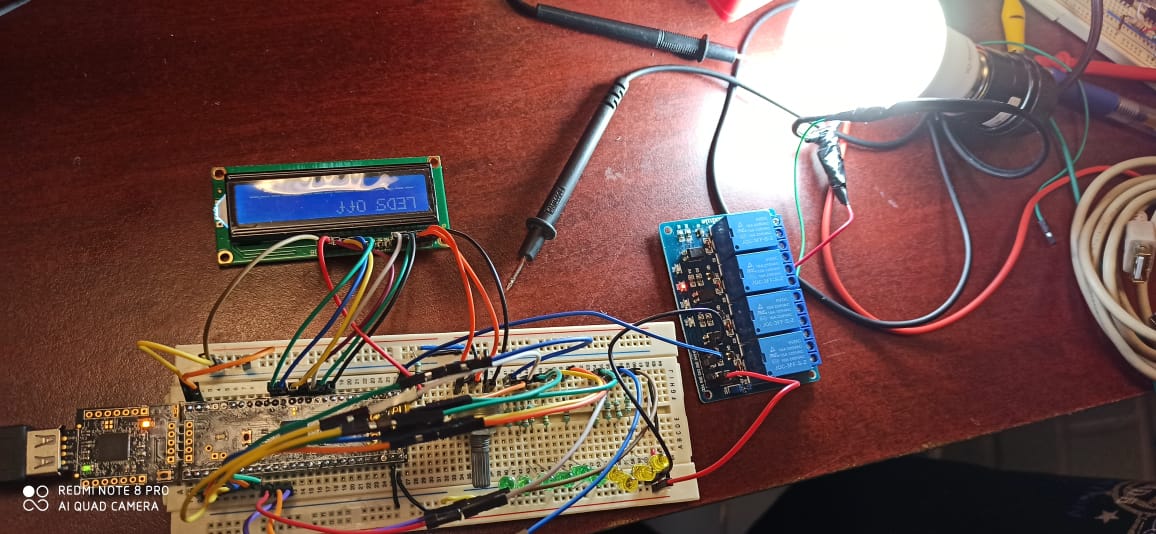
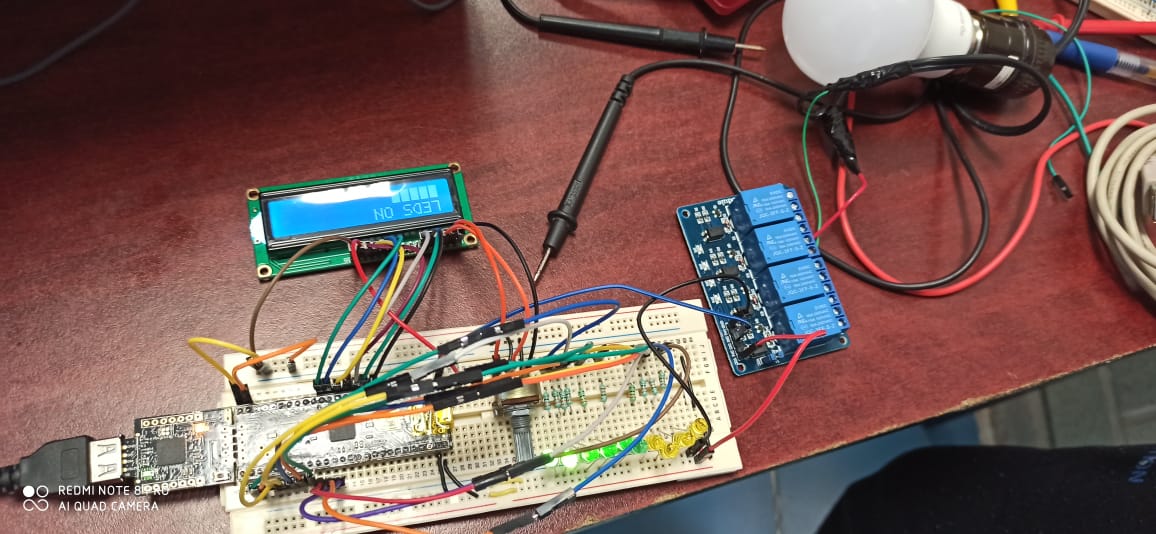
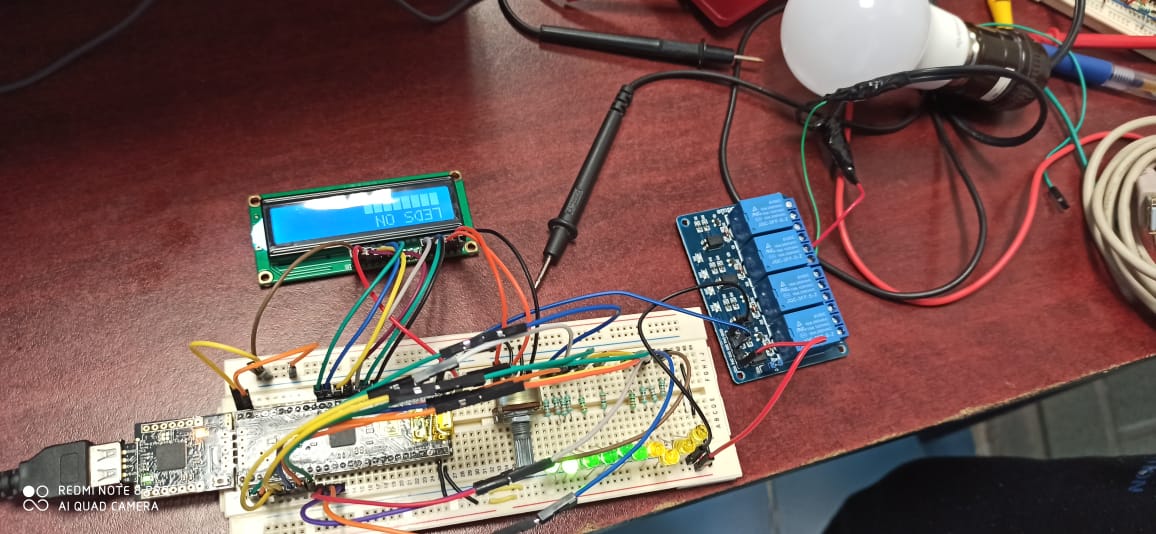
}

}

}

1. Programamos la tarjeta para proceder a ver los resultados.

**Resultados:**



**CONCLUSIÓN:**

Esta practica resulto ser mas complicada de lo que llegue a pensar puesto que las interrupciones son un tema en el cual aun no estoy muy preparado, pero a pesar de eso los conocimientos adquiridos anteriormente resultaron ser una gran fuente de apoyo para poder obtener el resultado final.