**3\_3\_Interrupciones**



**Programación De Sistemas Embebidos**

Mecatrónica 8°A

**Maestro**: Moran Garabito Carlos

**Alumno:**

* Cesar Omar Alvarado Contreras.

**Interrupción**

**OBJETIVOS**: El alumno deberá realizar la programación de la tarjeta CY8CKIT-059 PSoC para hacer una interrupción por software y para una serie de led y prender un foco con interface de potencia.

**MARCO TEÓRICO:**

**IRQ**

Las IRQ son líneas que llegan al controlador de interrupciones, un componente de hardware dedicado a la gestión de las interrupciones, y que puede estar integrado en el procesador principal o ser un circuito separado conectado al mismo. El controlador de interrupciones debe ser capaz de habilitar o inhibir las líneas de interrupción y establecer prioridades entre las mismas. Cuando varias líneas de petición de interrupción se activan a la vez, el controlador de interrupciones utilizará estas prioridades para escoger la interrupción sobre la que informará al procesador principal. También puede darse el caso de que una rutina de tratamiento de interrupción sea interrumpida para realizar otra rutina de tratamiento de una interrupción de mayor prioridad a la que se estaba ejecutando; aunque hay interrupciones que no se pueden deshabilitar (conocidas como interrupciones no enmascarables o NMI).

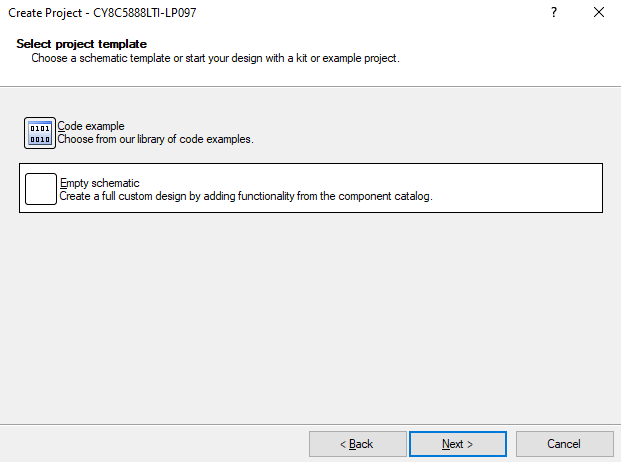
Un procesador principal que no tenga un controlador de interrupciones integrado, suele tener una única línea de interrupción llamada habitualmente INT. Esta línea es activada por el controlador de interrupciones cuando tiene una interrupción que servir. Al activarse esta línea, el procesador consulta los registros del controlador de interrupciones para averiguar cual IRQ hay que atender. A partir del número del IRQ busca en la tabla de vectores de interrupción la dirección de la rutina a la que debe llamar para atender la petición del dispositivo asociado a dicha IRQ.

**MATERIALES:**

1. Computadora con software PSoC Creator.
2. Tarjeta CY8CKIT-059 PSoC
3. LED´s 10
4. Resistencias 10
5. Foco y relevador con clavija.

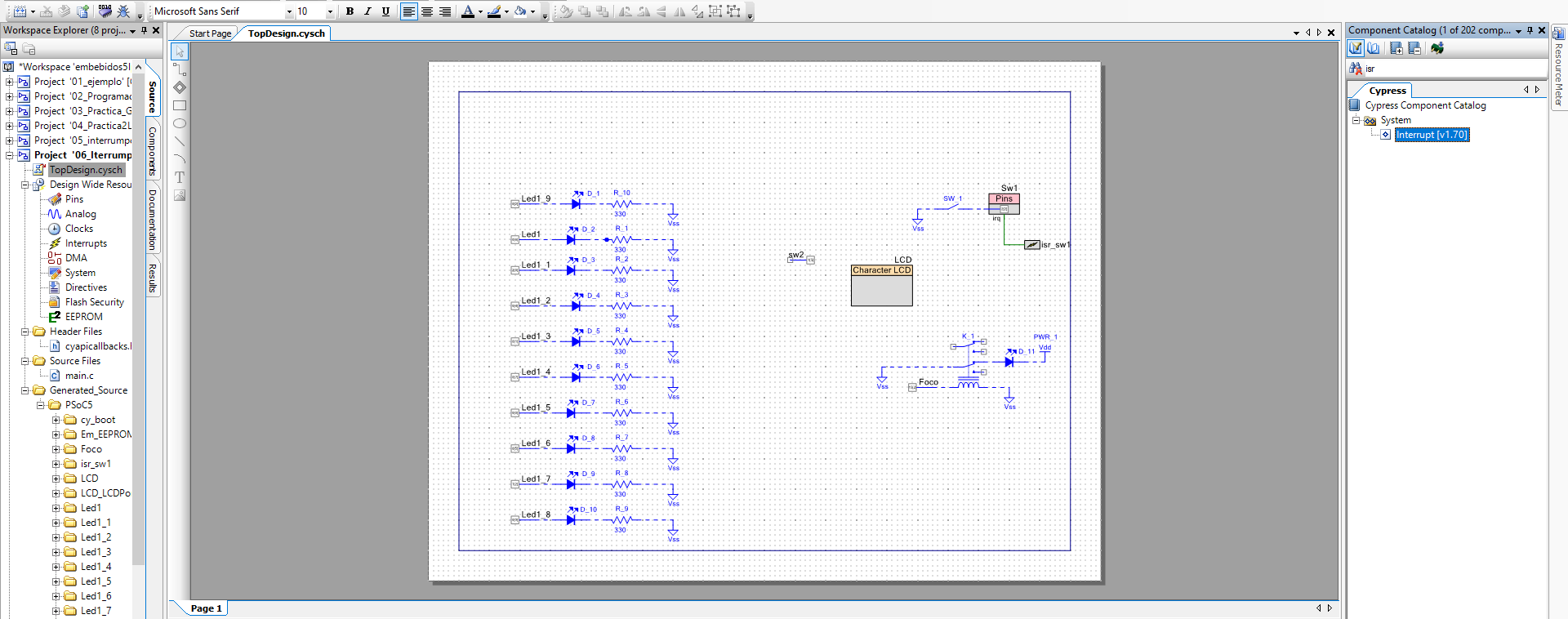
**PROCEDIMIENTO:**

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

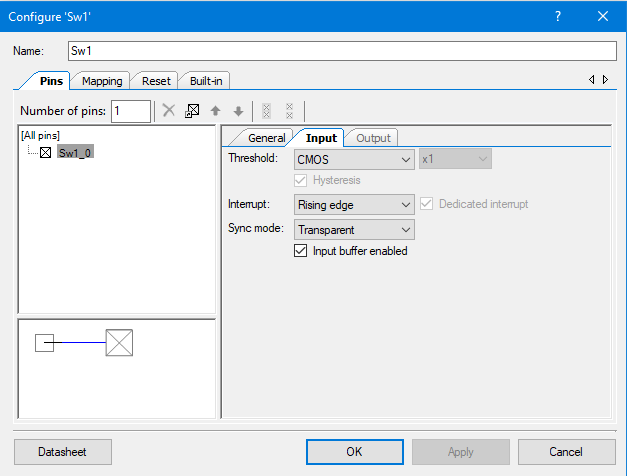


1. Insertar los componentes necesarios para su posterior uso. Una vez hecho esto se puedo construir el proyecto sin problema alguno.

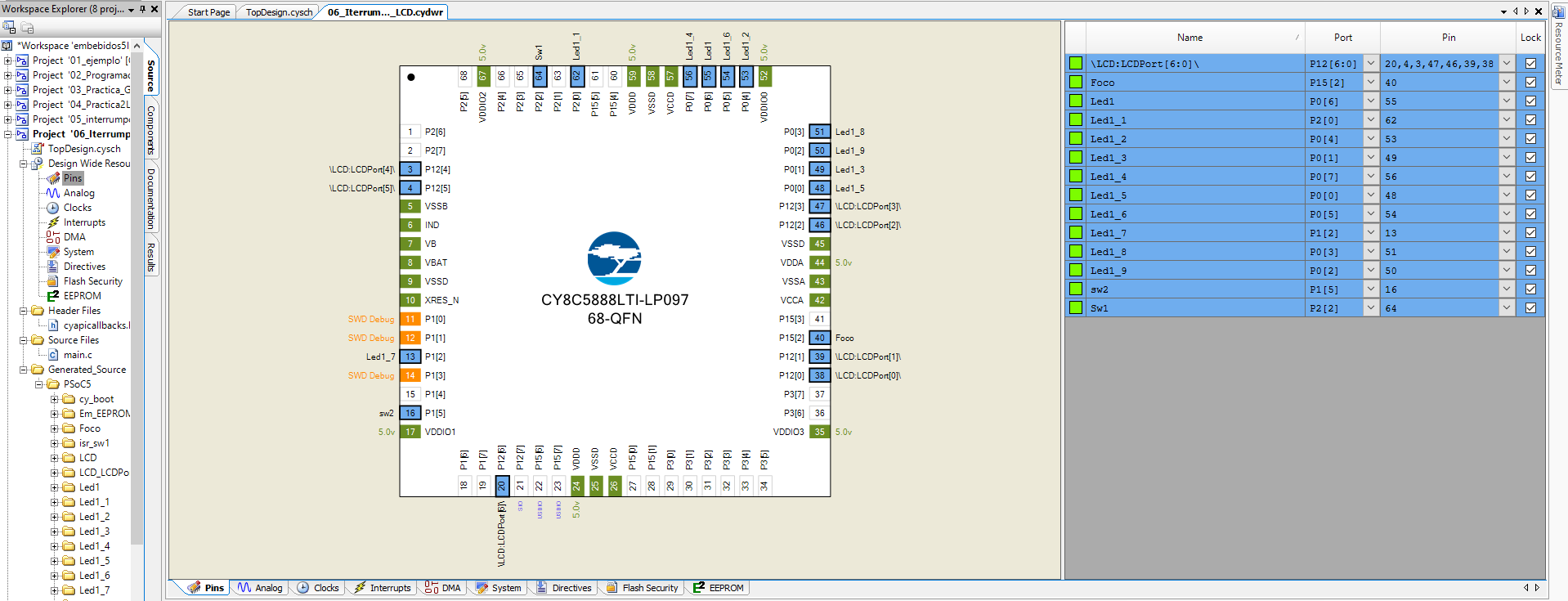
* Primero ingresamos a TopDesign a través del menú que se encuentra en la izquierda.
* En el menú que se encuentra a la derecha podremos escoger los componentes que necesitemos.



1. Configuraremos el Sw1 de la siguiente manera para poder conectar la interrupción como Interrupt.

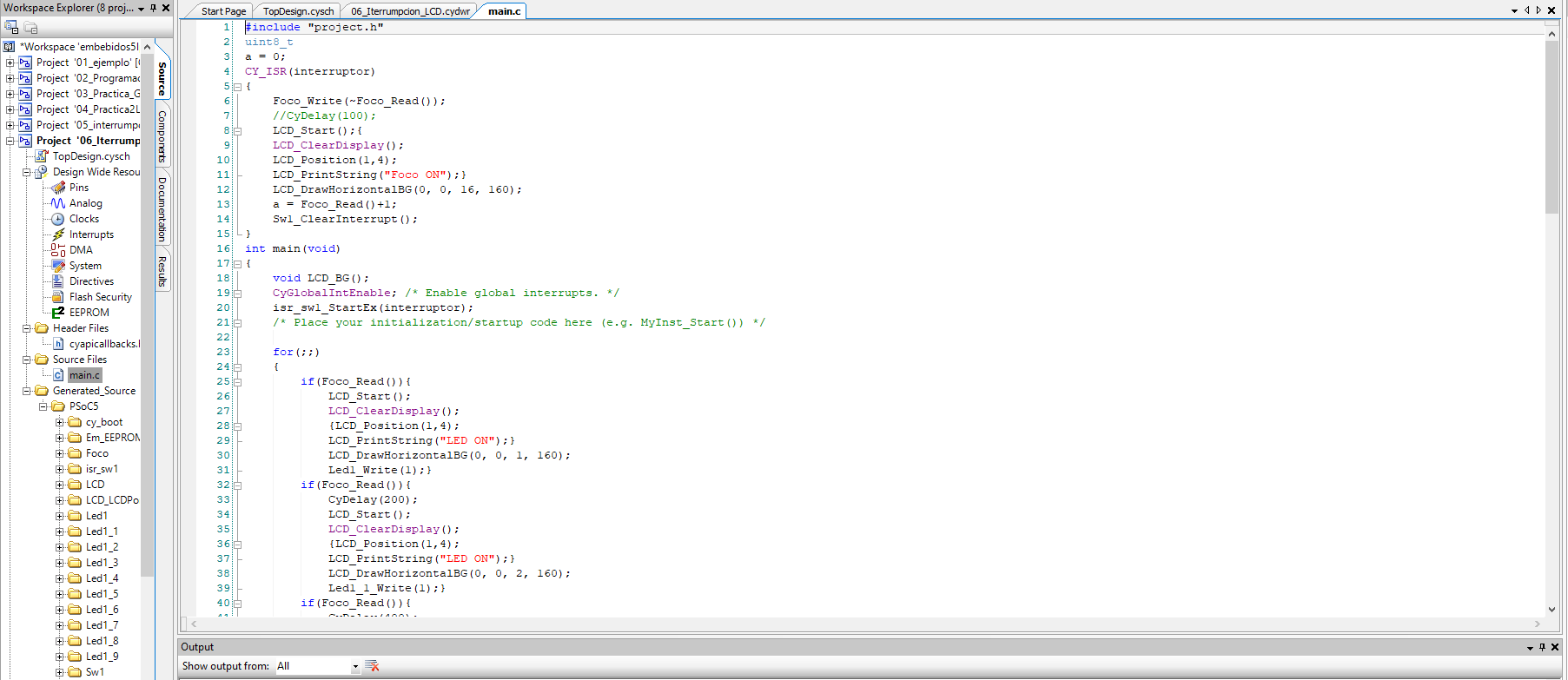


1. Definir los pines que usaremos para cada componente. Este lo haremos ingresando al menú dando clic en Pins.



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"

uint8\_t

a = 0;

CY\_ISR(interruptor)

{

Foco\_Write(~Foco\_Read());

//CyDelay(100);

LCD\_Start();{

LCD\_ClearDisplay();

LCD\_Position(1,4);

LCD\_PrintString("Foco ON");}

LCD\_DrawHorizontalBG(0, 0, 16, 160);

a = Foco\_Read()+1;

Sw1\_ClearInterrupt();

}

int main(void)

{

void LCD\_BG();

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

isr\_sw1\_StartEx(interruptor);

/\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

for(;;)

{

if(Foco\_Read()){

LCD\_Start();

LCD\_ClearDisplay();

{LCD\_Position(1,4);

LCD\_PrintString("LED ON");}

LCD\_DrawHorizontalBG(0, 0, 1, 160);

Led1\_Write(1);}

if(Foco\_Read()){

CyDelay(200);

LCD\_Start();

LCD\_ClearDisplay();

{LCD\_Position(1,4);

LCD\_PrintString("LED ON");}

LCD\_DrawHorizontalBG(0, 0, 2, 160);

Led1\_1\_Write(1);}

if(Foco\_Read()){

CyDelay(400);

LCD\_Start();

LCD\_ClearDisplay();

{LCD\_Position(1,4);

LCD\_PrintString("LED ON");}

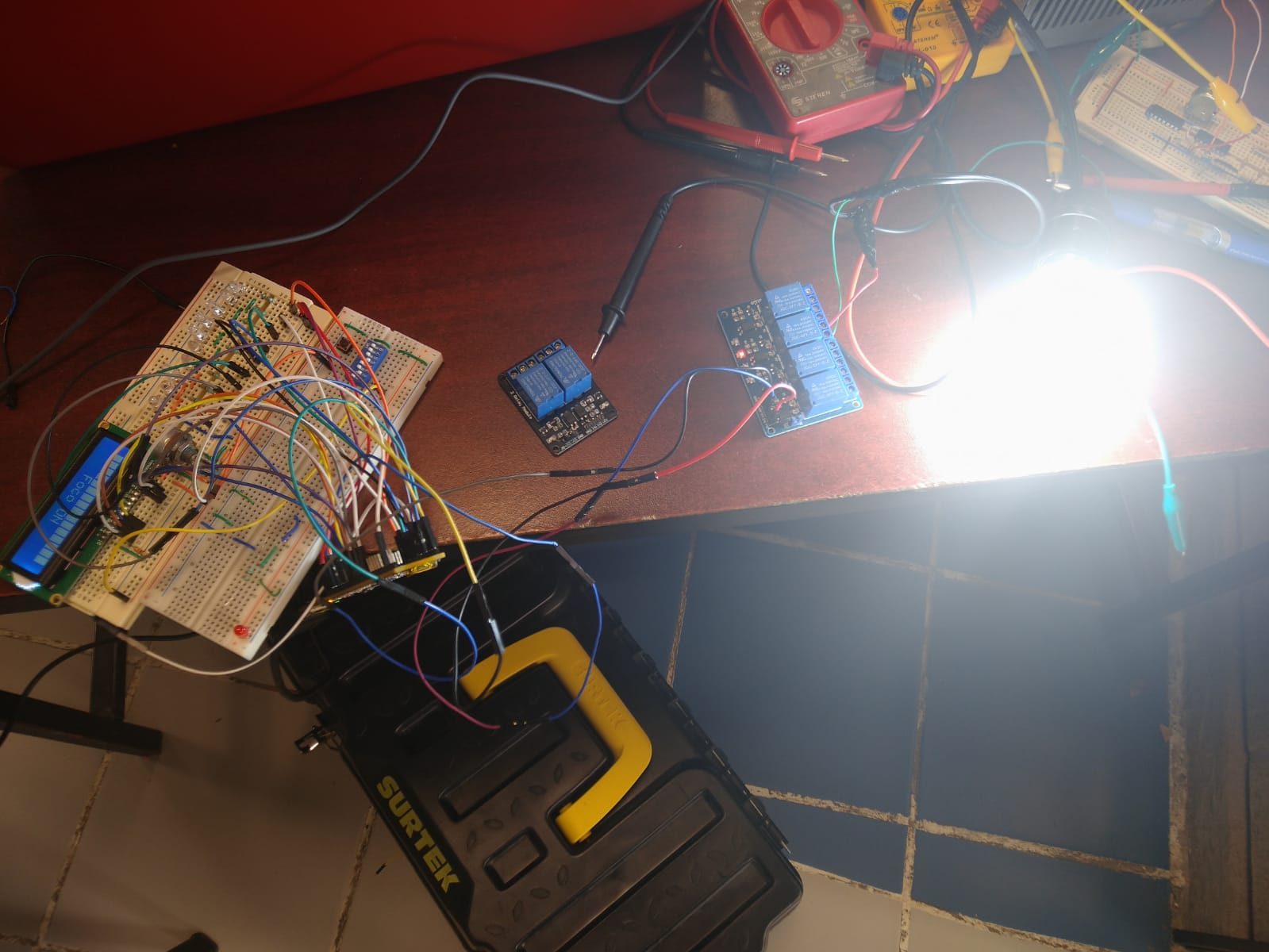
LCD\_DrawHorizontalBG(0, 0, 3, 160);

Led1\_2\_Write(1);}

1. Programamos la tarjeta para proceder a ver los resultados.
2. **Resultados:**

Una vez programada la tarjeta procedemos a conectar todo en su lugar como se muestra en el top TopDesign y lo probamos si todo va bien funcionara como debe.





**CONCLUSIÓN:**

El diseño de estar practica se desarrollo muy simple por la cuestión de que se resolvía por si sola con solo juntar una practica anterior y agregando la interrupción en lo cual se aprendió a usar este tipo de programa para parar el sistema y atender otra cosa que es prioridad