**3\_1\_GPIO**



**Programación De Sistemas Embebidos**

Mecatrónica 8°A

**Maestro**: Moran Garabito Carlos

**Alumno:**

* Eduardo Robles Vázquez

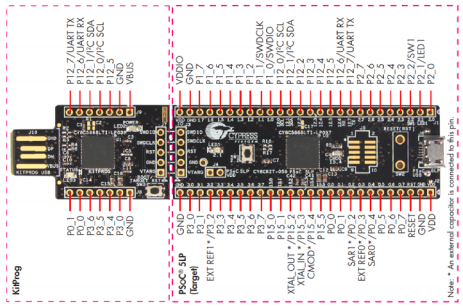
**GPIO**

**OBJETIVOS**: El alumno deberá realizar la programación de la tarjeta CY8CKIT-059 PSoC para cumplir con las sucesiones de leds impuestas por el profesor.

**MARCO TEÓRICO:**

**PSoC 5LP**

PSoC 5LP es el SoC programable más integrado de la industria, que combina periféricos analógicos y digitales de alta precisión y programables con una CPU ARM ® Cortex ® -M3 en un solo chip. Procese las señales del sensor con el coprocesador DFB de hardware de 24 bits, descargue las tareas tradicionales de la CPU a los bloques digitales universales basados ​​en CPLD y aumente el rendimiento del sistema con el controlador DMA de periférico a periférico. Integre los extremos frontales analógicos personalizados de 20 bits de alta precisión con los bloques analógicos programables que incluyen opamps, PGA, filtros, comparadores, ADC SAR y Delta-Sigma y la mejor solución de detección táctil CapSense de la industria.



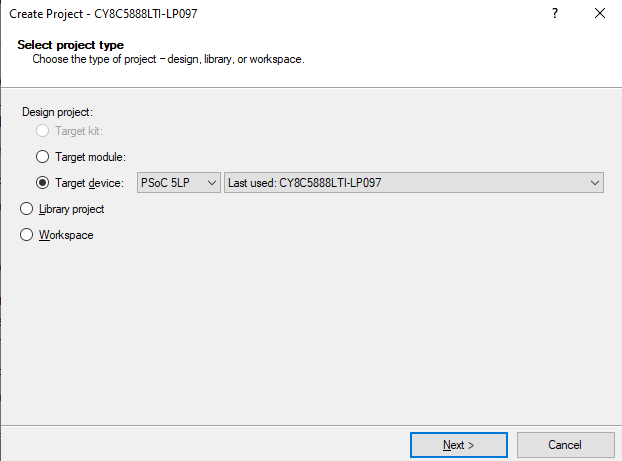
Las I / O del dispositivo PSoC 5LP en un formato compatible con la placa de pruebas. Cuenta con un encabezado micro-USB para crear prototipos con conectividad Full Speed ​​USB 2.0. También está diseñado con un conveniente factor de forma snappable, que permite a los usuarios separar el conector USB con el programador y depurador KitProg de la placa de destino para usarlos de forma independiente.

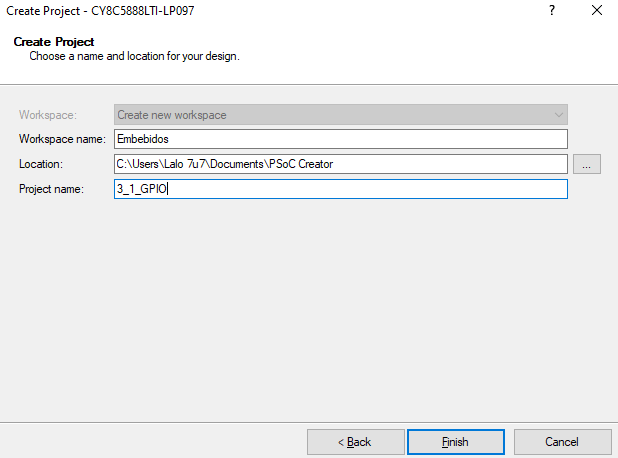
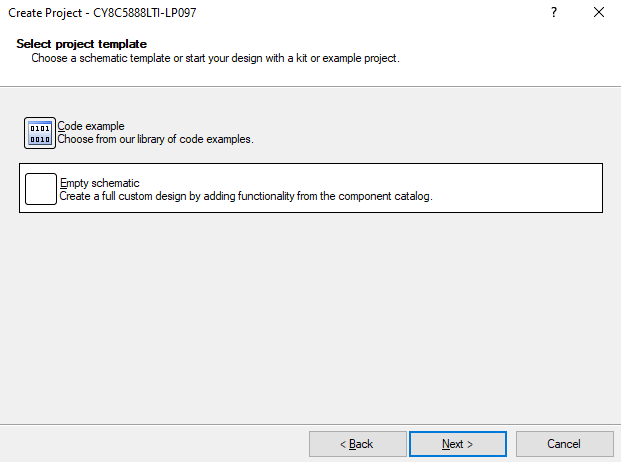
**MATERIALES:**

1. Computadora con software PSoC Creator.
2. Tarjeta CY8CKIT-059 PSoC
3. Leds
4. Resistencias

**PROCEDIMIENTO:**

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



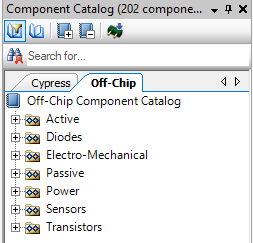


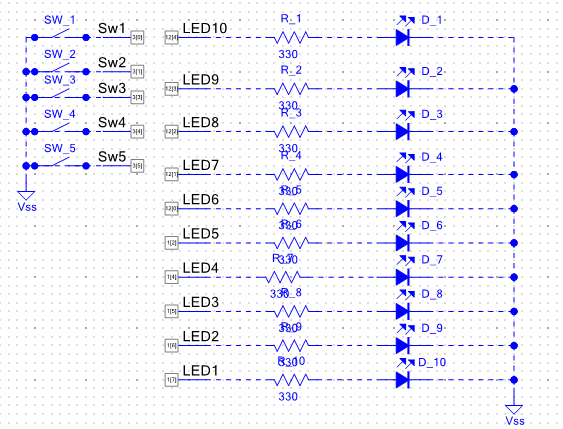
1. Insertar los componentes necesarios para su posterior uso. Una vez hecho esto se puedo construir el proyecto sin problema alguno.

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

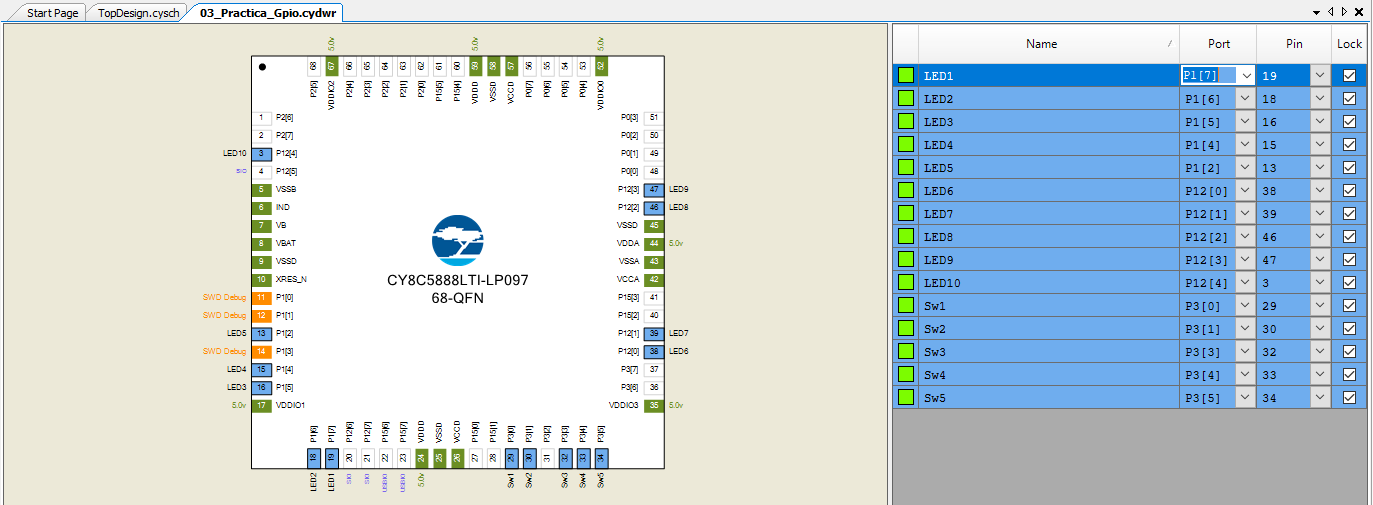
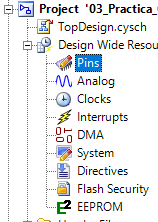


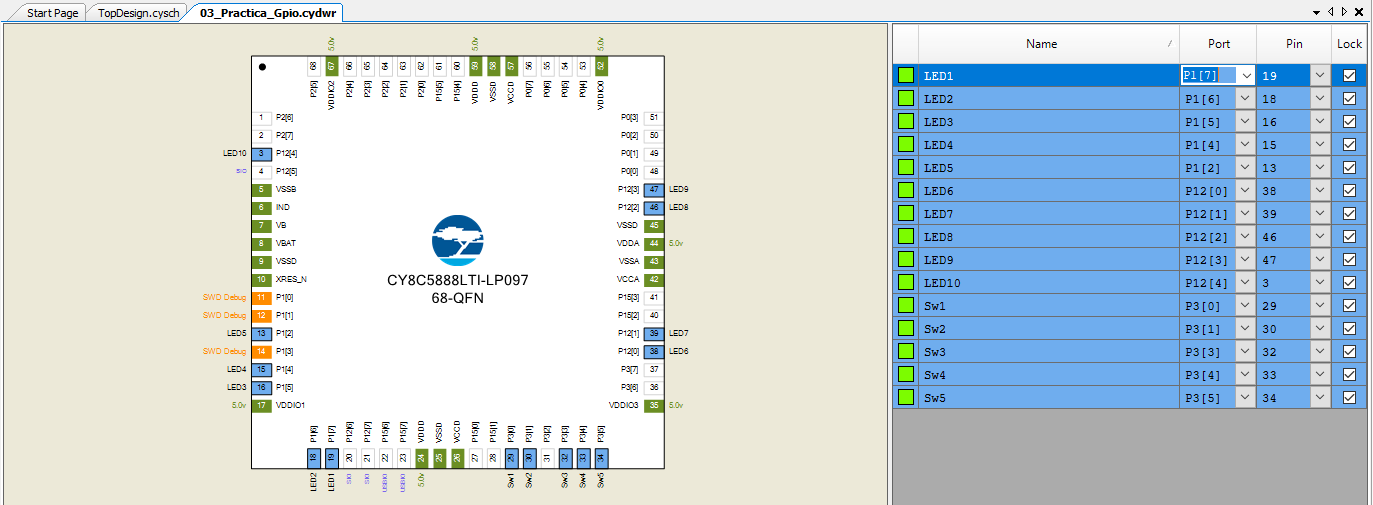
* En el menú que se encuentra a la derecha podremos escoger los componentes que necesitemos.



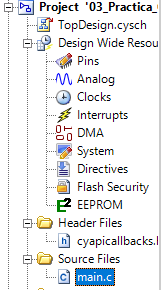


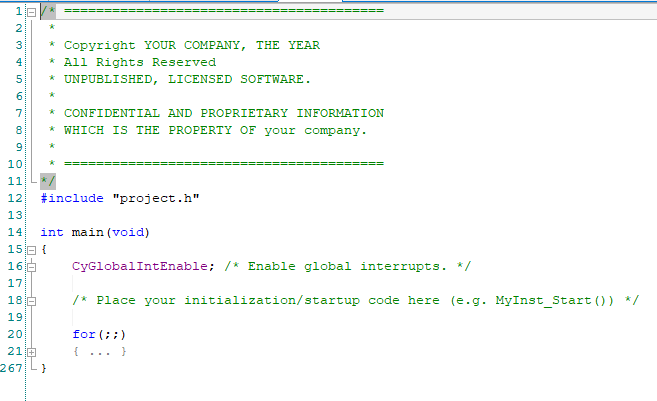
1. Definir los pines que usaremos para cada componente. Este lo haremos ingresando al menú dando clic en Pins.





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.

int main(void)

{

CyGlobalIntEnable;

for(;;)

{

switch(Sw1\_Read()){

case 1:{

LED1\_Write(1);

CyDelay(500);

LED1\_Write(0);

CyDelay(500);

LED10\_Write(1);

CyDelay(500);

LED10\_Write(0);

CyDelay(500);

LED2\_Write(1);

CyDelay(500);

LED2\_Write(0);

CyDelay(500);

LED9\_Write(1);

CyDelay(500);

LED9\_Write(0);

CyDelay(500);

LED3\_Write(1);

CyDelay(500);

LED3\_Write(0);

CyDelay(500);

LED8\_Write(1);

CyDelay(500);

LED8\_Write(0);

CyDelay(500);

LED4\_Write(1);

CyDelay(500);

LED4\_Write(0);

CyDelay(500);

LED7\_Write(1);

CyDelay(500);

LED7\_Write(0);

CyDelay(500);

LED5\_Write(1);

CyDelay(500);

LED5\_Write(0);

CyDelay(500);

LED6\_Write(1);

CyDelay(500);

LED6\_Write(0);

CyDelay(500);

}

break;

}

switch(Sw2\_Read()){

case 1:{

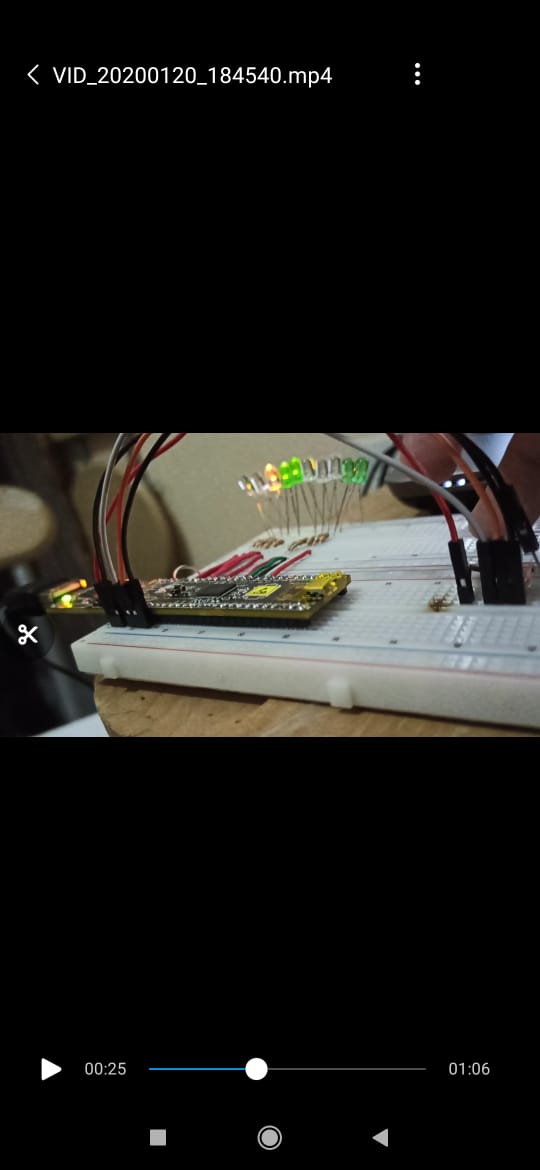
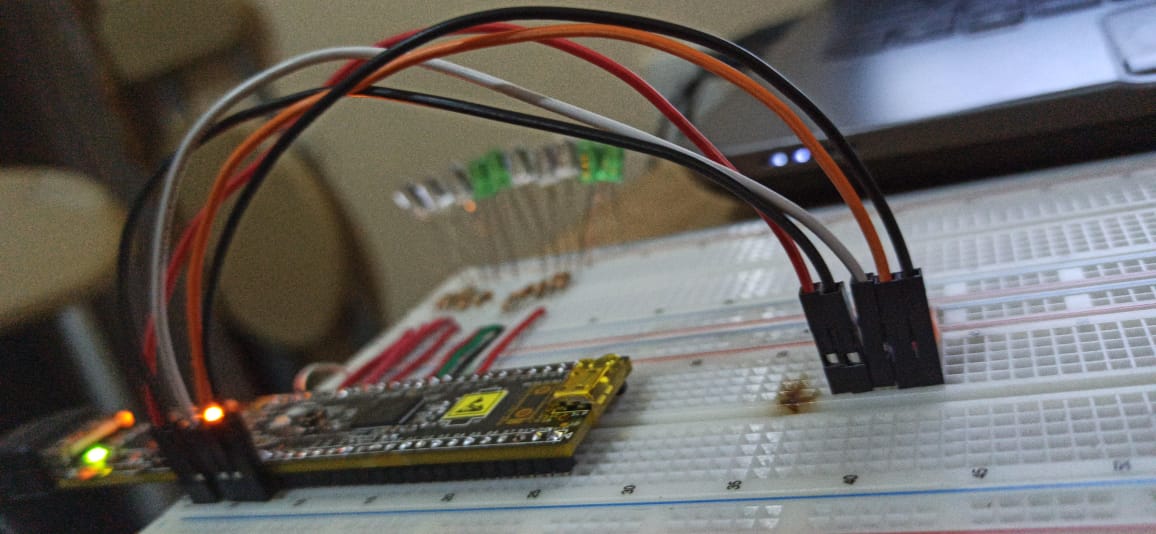
break;

}

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

**Resultados:**

Una vez programada la tarjeta al seleccionar alguno de los 5 diferentes switch podemos acceder a las diferentes sucesiones de leds con lo cual concluimos de manera satisfactoria la práctica.



**CONCLUSIÓN:**

En esta primera practica aprendimos a utilizar los GPIO de la tarjeta de una forma visual y a su vez aprendimos a programar y conocer características físicas de nuestra tarjeta para futuros usos. Es importante llevar a cabo este tipo de prácticas para que lo aprendido teóricamente se pueda reforzar en algo real y tangible.