



**UNIVERSIDAD POLITÉCNICA
DE LA ZONA METROPOLITANA DE GUADALAJARA**

PROGRAMACIÓN DE SISTEMAS EMBEBIDOS

SAMUEL CALEB MARTÍNEZ HERNÁNDEZ

8-A ING. MECATRÓNICA

PRACTICA 1_3_1_GPIO

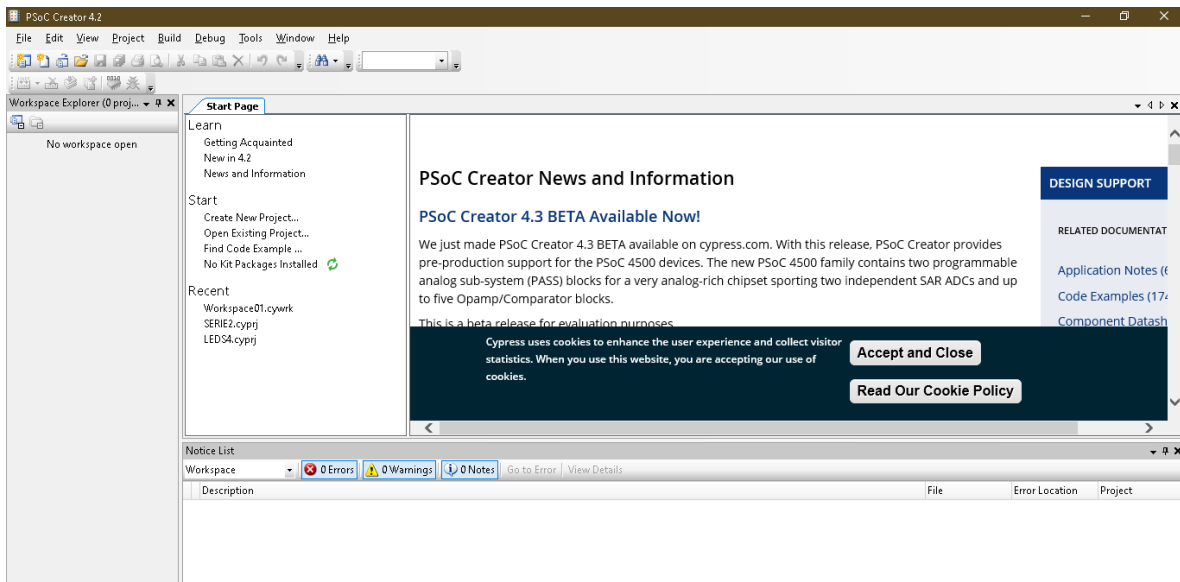
INTRODUCCIÓN

En esta ocasión se utilizaron diversos aparatos para la elaboración de la practica 1, entre ellos el microprocesador PSoC 5LP CY8CKIT-059 de Cypress. Este dispositivo será de vital utilidad dado que en ella se alojará la programación que deseemos.



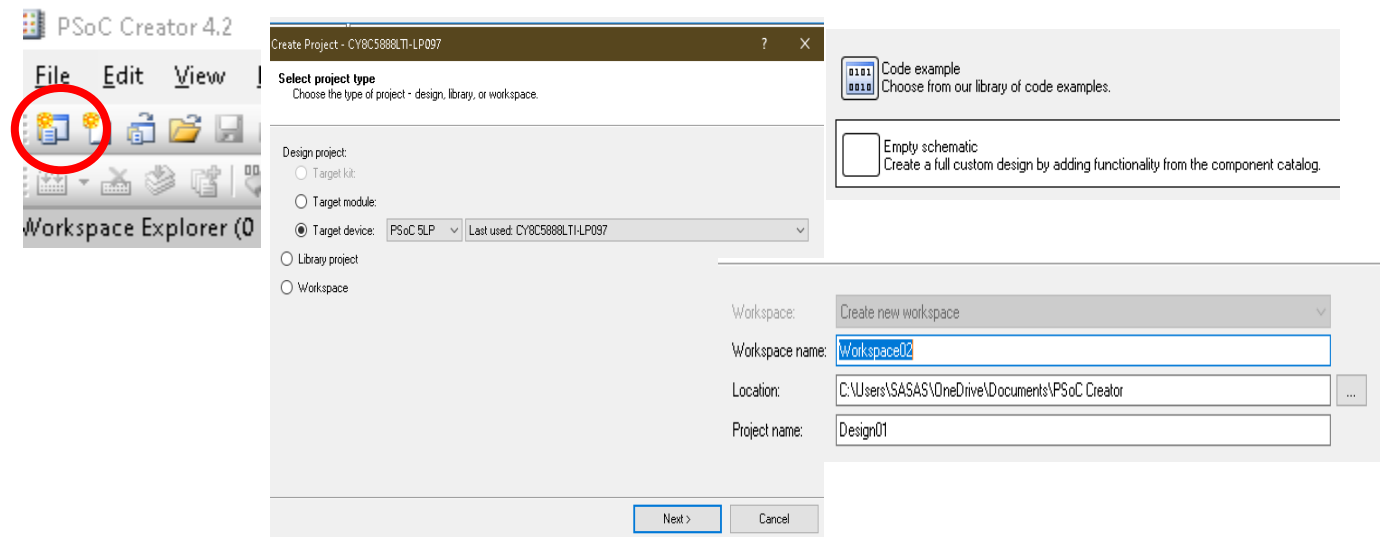
Sin embargo, para que esté listo para la práctica se requirió soldarle todos los pines para que fuera muchísimo más fácil el conectarle cables.

Para la programación de este dispositivo se utilizó el programa PSoC creator, y el programmer creator (cuya labor es actualizar el PSoC creator).

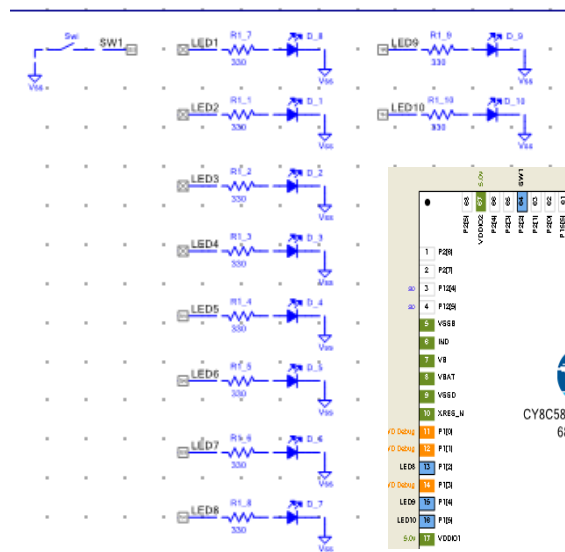


DESARROLLO

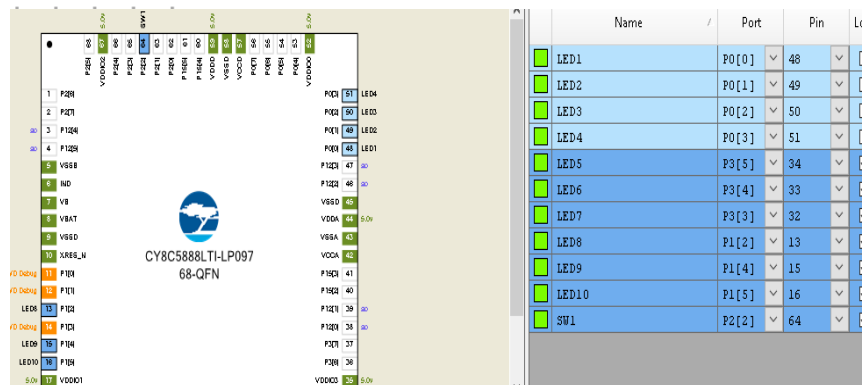
Lo primero que se realizó, fue el crear un nuevo proyecto dentro del programa de PSoC Creator, especificar el modelo de la PSoC, establecer un directorio el cual será tu espacio de trabajo “workspace”, además del nombre del proyecto y generar un esquemático vacío.



Una vez fue creado el proyecto, se creo el diagrama de circuito, posteriormente se compilo para verificar que no hubiese ninguna clase de error, se utilizaron 10 leds, 10 resistencias de 330 Ohms y un Switch (propio de la PSoC) en total, con sus respectivas tierras.

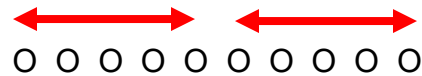


Los pines fueron establecidos por la misma compilación, salvo el switch, el cual tenia que ser necesariamente P2[2].



Cuando se compiló, el PSoC Creator nos genero un archivo tipo “**main.c**” el cual será nuestra herramienta de programación, con la cual le daremos la tarea de generarnos diversas tareas como las siguientes:

Serie 1:



Del primer LED pasa al segundo, después al tercero y llega hasta el 5, respectivamente lo hace el LED 6 hasta llegar al 10 y ambas partes se regresan hasta quedar nuevamente en el LED 1 y el LED 6.

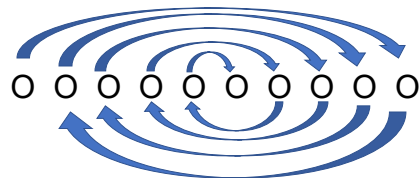
Código

```

12 #include "project.h"
13
14 int main(void)
15 {
16     CyGlobalIntEnable; /* Enable global interrupts. */
17
18     /* Place your initialization/startup code here (e.g. MyInst_Start()) */
19
20     for(;;)
21     {
22         if(SW1_Read())
23         {LED1_Write(1), LED10_Write(1);
24          CyDelay(800);
25          LED1_Write(1),LED10_Write(1), LED2_Write(1), LED9_Write(1);
26          CyDelay(800);
27          LED2_Write(1), LED9_Write(1), LED3_Write(1), LED8_Write(1),LED1_Write(1),LED10_Write(1);
28          CyDelay(800);
29          LED3_Write(1), LED8_Write(1), LED4_Write(1), LED7_Write(1),LED2_Write(1), LED9_Write(1),LED1_Write(0),
30          CyDelay(800);
31          LED3_Write(1), LED8_Write(1),LED4_Write(1), LED7_Write(1), LED5_Write(1), LED6_Write(1), LED2_Write(0)
32          CyDelay(800);
33          LED3_Write(0), LED8_Write(0),LED4_Write(1), LED7_Write(1), LED5_Write(1), LED6_Write(1);
34          CyDelay(800);
35          LED4_Write(0), LED7_Write(0), LED5_Write(1), LED6_Write(1);
36          CyDelay(800);
37          LED5_Write(0), LED6_Write(0);
38          CyDelay(800);
39         } else {
40             LED1_Write(0);
41             CyDelay(600);
42             break;
43         }
44     }
45 }
46
47 /* [] END OF FILE */
48

```

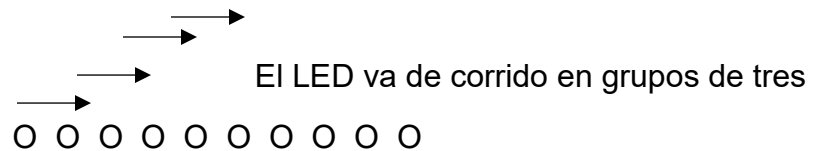
Serie 2:



Código

```
19
20     for(;;)
21     {
22         if(SW1_Read())
23         {LED1_Write(1);
24         CyDelay(800);
25         LED1_Write(0), LED10_Write(1);
26         CyDelay(800);
27         LED2_Write(1), LED10_Write(0);
28         CyDelay(800);
29         LED9_Write(1), LED2_Write(0);
30         CyDelay(800);
31         LED3_Write(1), LED9_Write(0);
32         CyDelay(800);
33         LED3_Write(0), LED8_Write(1);
34         CyDelay(800);
35         LED8_Write(0), LED4_Write(1);
36         CyDelay(800);
37         LED4_Write(0), LED7_Write(1);
38         CyDelay(800);
39         LED5_Write(1), LED7_Write(0);
40         CyDelay(800);
41         LED6_Write(1), LED5_Write(0);
42         CyDelay(800);
43         LED1_Write(1), LED6_Write(0);
44     } else {
45         LED1_Write(0);
46         CyDelay(600);
47     }
```

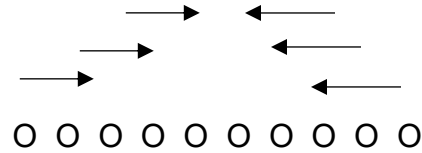
Serie 3:



Código

```
20     for(;;)
21     {
22         if(SW1_Read())
23         {LED1_Write(1),LED2_Write(1),LED3_Write(1);
24         CyDelay(800);
25         LED1_Write(0), LED2_Write(1),LED3_Write(1),LED4_Write(1);
26         CyDelay(800);
27         LED2_Write(0),LED3_Write(1),LED4_Write(1), LED5_Write(1);
28         CyDelay(800);
29         LED3_Write(0),LED4_Write(1), LED5_Write(1), LED6_Write(1);
30         CyDelay(800);
31         LED4_Write(0), LED5_Write(1), LED6_Write(1), LED7_Write(1);
32         CyDelay(800);
33         LED5_Write(0), LED6_Write(1), LED7_Write(1), LED8_Write(1);
34         CyDelay(800);
35         LED6_Write(0), LED7_Write(1), LED8_Write(1), LED9_Write(1);
36         CyDelay(800);
37         LED7_Write(0), LED8_Write(1), LED9_Write(1), LED10_Write(1);
38         CyDelay(800);
39         LED8_Write(0), LED9_Write(1), LED10_Write(1), LED1_Write(1);
40         CyDelay(800);
41         LED9_Write(0), LED10_Write(1), LED1_Write(1), LED2_Write(1);
42         CyDelay(800);
43         LED10_Write(0), LED1_Write(1), LED2_Write(1), LED3_Write(1);
44     } else {
45         LED1_Write(0);
46         CyDelay(600);
47         break;
48     }
```

Serie 4:

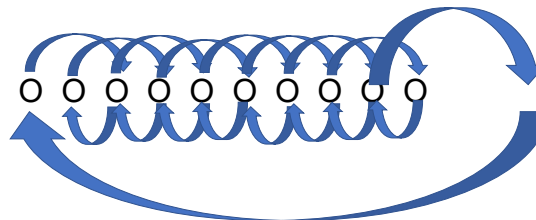


Los LED van en grupo de 3 hasta encontrarse el 5 con el 6 y desaparecer en fila respectivamente.

Código

```
16 CyGlobalIntEnable; /* Enable global interrupts. */
17
18 /* Place your initialization/startup code here (e.g. MyInst_Start()) */
19
20 for(;;)
21 {
22     if(SW1_Read())
23     {
24         LED1_Write(1), LED10_Write(1);
25         CyDelay(800);
26         LED1_Write(1), LED10_Write(1), LED2_Write(1), LED9_Write(1);
27         CyDelay(800);
28         LED2_Write(1), LED9_Write(1), LED3_Write(1), LED8_Write(1), LED1_Write(1), LED10_Write(1);
29         CyDelay(800);
30         LED3_Write(1), LED8_Write(1), LED4_Write(1), LED7_Write(1), LED2_Write(1), LED9_Write(1), LED1_Write(0), LED10_Write(0);
31         CyDelay(800);
32         LED3_Write(1), LED8_Write(1), LED4_Write(1), LED7_Write(1), LED5_Write(1), LED6_Write(1), LED2_Write(0), LED9_Write(0);
33         CyDelay(800);
34         LED3_Write(0), LED8_Write(0), LED4_Write(1), LED7_Write(1), LED5_Write(1), LED6_Write(1);
35         CyDelay(800);
36         LED4_Write(0), LED7_Write(0), LED5_Write(1), LED6_Write(1);
37         CyDelay(800);
38         LED5_Write(0), LED6_Write(0);
39         CyDelay(800);
40     } else {
41         LED1_Write(0);
42         CyDelay(600);
43         break;
44     }
45 }
```

Serie 5:



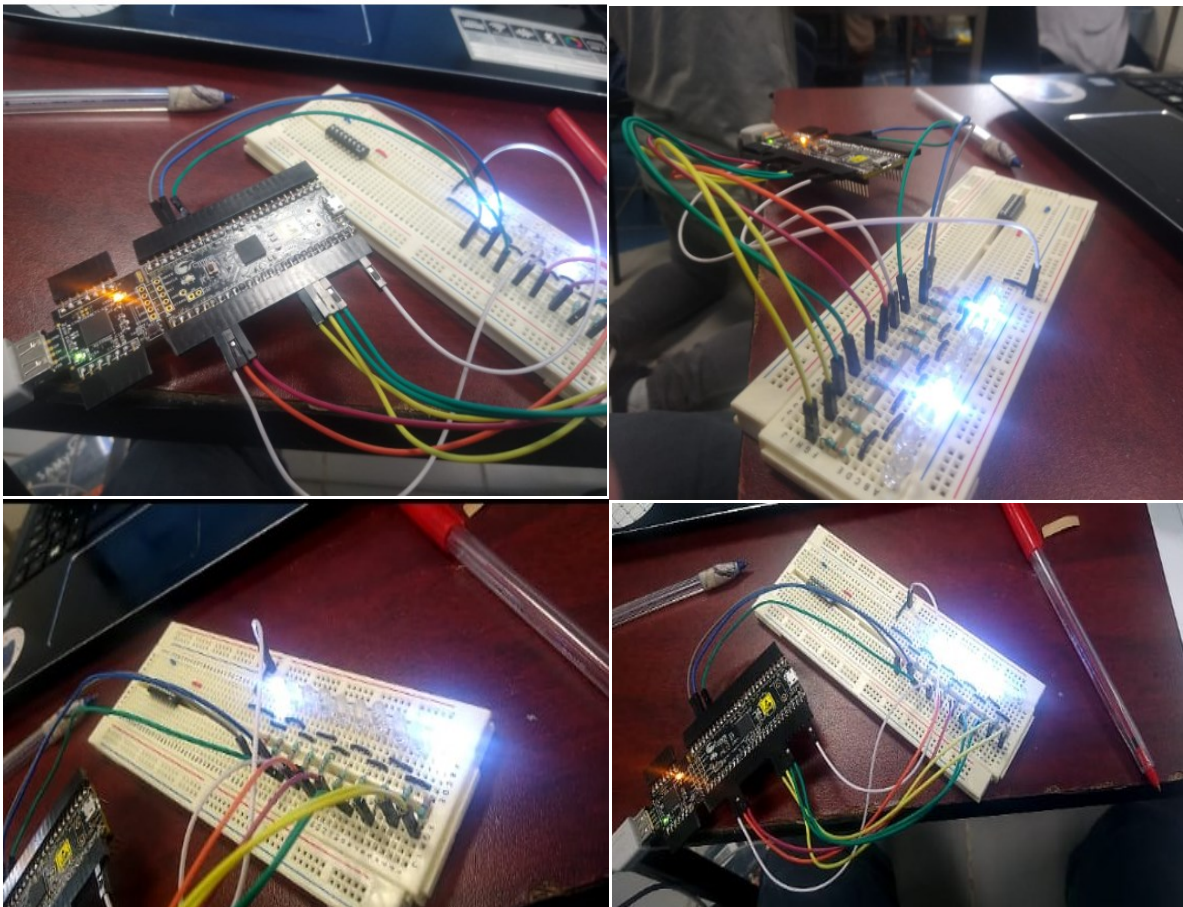
El LED avanza dos y regresa uno.

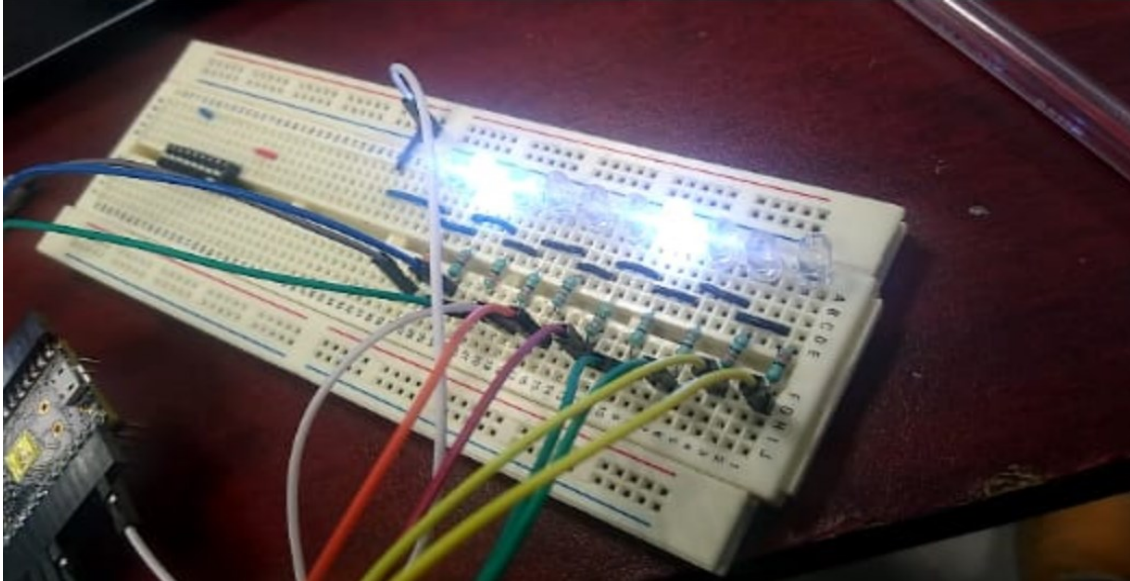
Código

```
for(;;)
{
    if(SW1_Read())
    {
        LED1_Write(1), LED10_Write(1);
        CyDelay(800);
        LED1_Write(0), LED10_Write(0), LED3_Write(1), LED8_Write(1);
        CyDelay(800);
        LED2_Write(1), LED3_Write(0), LED8_Write(0), LED9_Write(1);
        CyDelay(800);
        LED4_Write(1), LED2_Write(0), LED9_Write(0), LED7_Write(1);
        CyDelay(800);
        LED3_Write(1), LED4_Write(0), LED8_Write(1), LED7_Write(0);
        CyDelay(800);
        LED3_Write(0), LED5_Write(1), LED8_Write(0), LED6_Write(1);
        CyDelay(800);
        LED5_Write(0), LED4_Write(1), LED7_Write(1), LED6_Write(0);
        CyDelay(800);
        LED4_Write(0), LED1_Write(1), LED7_Write(0), LED10_Write(1);
        CyDelay(800);
    } else {
        LED1_Write(0);
        CyDelay(600);
        break;
    }
}
```

Todas las series de esta practica cuentan con el mismo TopDesing y establecimiento de pins.

FOTOS DE LA PRACTICA





CONCLUSIÓN

Es imposible el no reconocer que este dispositivo se siente muy familiar comparándolo con la raspberry (dispositivo usado en cuatrimestres pasados) en cuanto a las funciones que se le han requerido se refiere. Podría decirse que esta actividad sirve muy bien como introducción para darse a la idea de que la PSoC cuenta con un sinfín de posibilidades.