**3\_5\_UART\_Tx\_Rx\_SPI\_I2C**



**Programación De Sistemas Embebidos**

Mecatrónica 8°A

**Maestro**: Moran Garabito Carlos

**Alumno:**

* Eduardo Robles Vázquez

**3\_5\_UART\_Tx\_Rx\_SPI\_I2C**

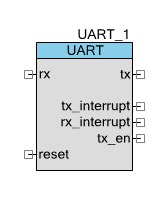
**OBJETIVOS**: El alumno deberá realizar la programación de la tarjeta CY8CKIT-059 PSoC para controlar un motor a pasos con ayudar de UART.

**MARCO TEÓRICO:**

**UART**

El UART proporciona comunicaciones asincrónicas comúnmente conocidas como RS232 o RS485. El componente UART se puede configurar para las versiones Full Duplex, Half Duplex, RX solamente o TX solamente. Todas las versiones proporcionan la misma funcionalidad básica. Sólo difieren en la cantidad de recursos utilizados.

Funciones



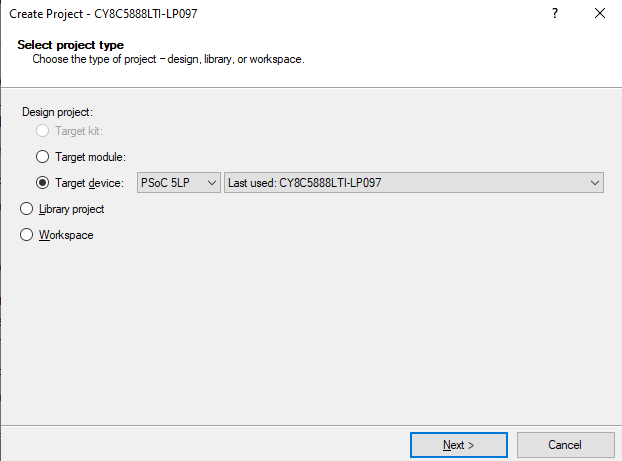
* Modo de dirección de 9 bits con detección de direcciones de hardware
* Tasas de baudios de 110 a 921600 bps o arbitrarias hasta 4 Mbps
* Búferes RX y TX de 4 a 65535
* Detección de errores de encuadre, paridad y sobrecarga
* Dúplex completo, medio dúplex, TX solamente, y hardware optimizado RX solamente
* Dos de cada tres votos por bit
* Generación y detección de señales de interrupción
* Sobre muestreo de 8x o 16x

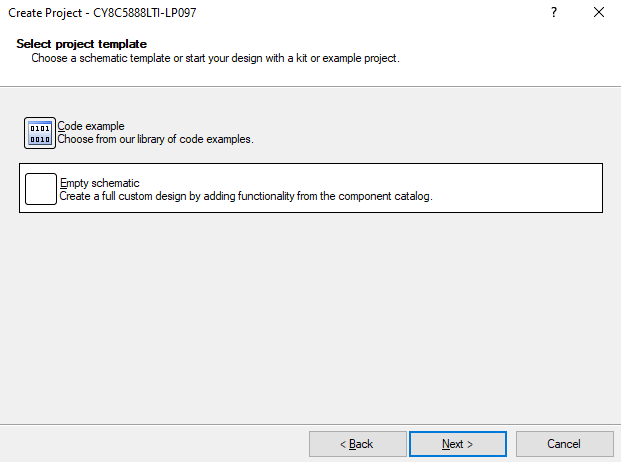
**MATERIALES:**

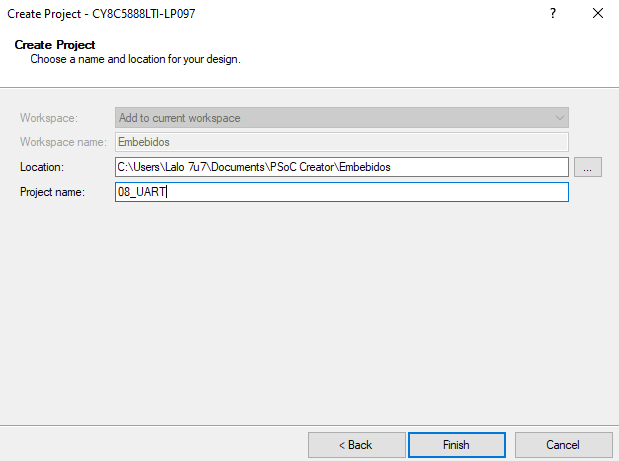
1. Computadora con software PSoC Creator.
2. Tarjeta CY8CKIT-059 PSoC
3. Software Putty
4. Motor a pasos y su driver.

**PROCEDIMIENTO:**

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





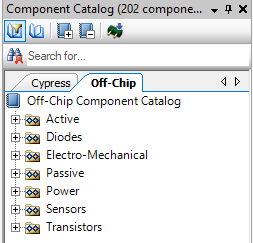


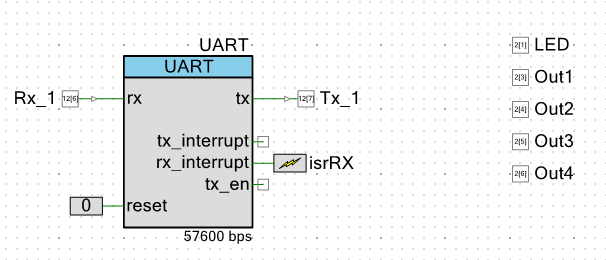
1. Insertar los componentes necesarios para su posterior uso. Una vez hecho esto se puedo construir el proyecto sin problema alguno.

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

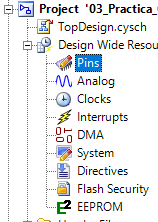
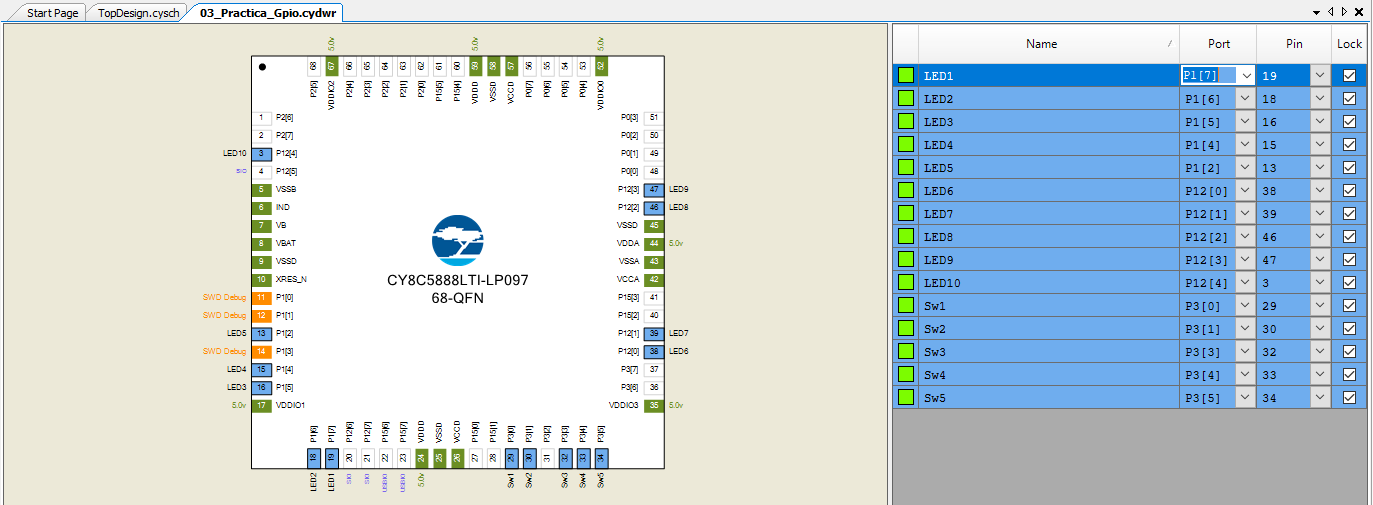


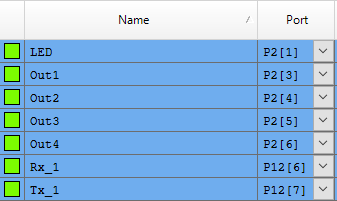
* En el menú que se encuentra a la derecha podremos escoger los componentes que necesitemos.



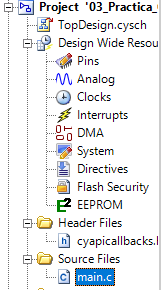


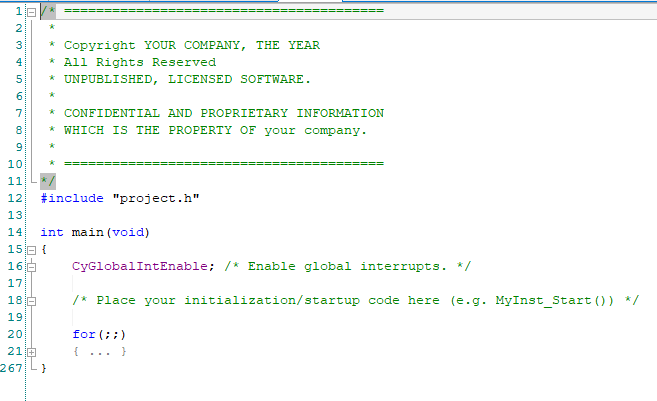
1. Definir los pines que usaremos para cada componente. Este lo haremos ingresando al menú dando clic en Pins.





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:

void horario(int grados) {

for (int x = 0; x<=grados \* AGrados; x++){

Out1\_Write(0);

Out2\_Write(0);

Out3\_Write(0);

Out4\_Write(1);

CyDelay(speed);

Out1\_Write(0);

Out2\_Write(0);

Out3\_Write(1);

Out4\_Write(0);

CyDelay(speed);

Out1\_Write(0);

Out2\_Write(1);

Out3\_Write(0);

Out4\_Write(0);

CyDelay(speed);

Out1\_Write(1);

Out2\_Write(0);

Out3\_Write(0);

Out4\_Write(0);

CyDelay(speed);

}

}

void antihorario(int grados){

for (int x = grados \* AGrados; x>=0; x--){

Out1\_Write(1);

Out2\_Write(0);

Out3\_Write(0);

Out4\_Write(0);

CyDelay(speed);

Out1\_Write(0);

Out2\_Write(1);

Out3\_Write(0);

Out4\_Write(0);

CyDelay(speed);

Out1\_Write(0);

Out2\_Write(0);

Out3\_Write(1);

Out4\_Write(0);

CyDelay(speed);

Out1\_Write(0);

Out2\_Write(0);

Out3\_Write(0);

Out4\_Write(1);

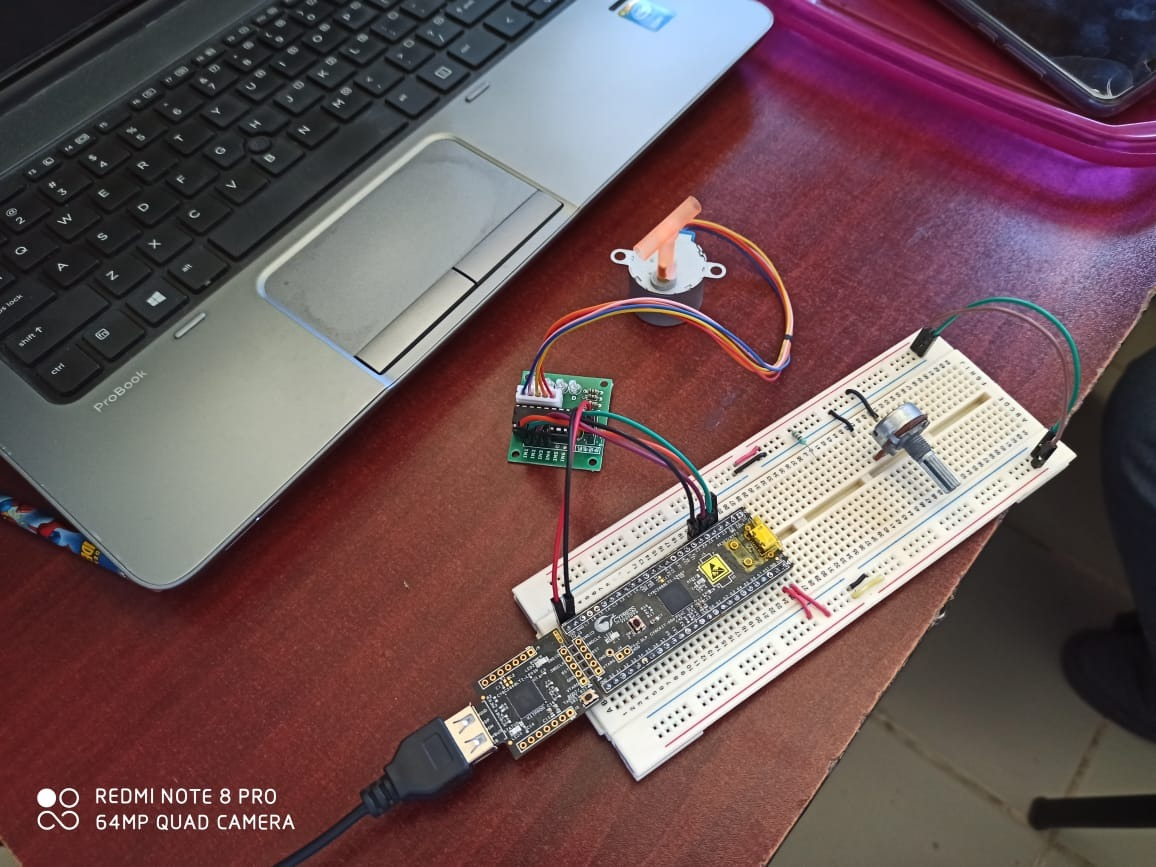
CyDelay(speed);

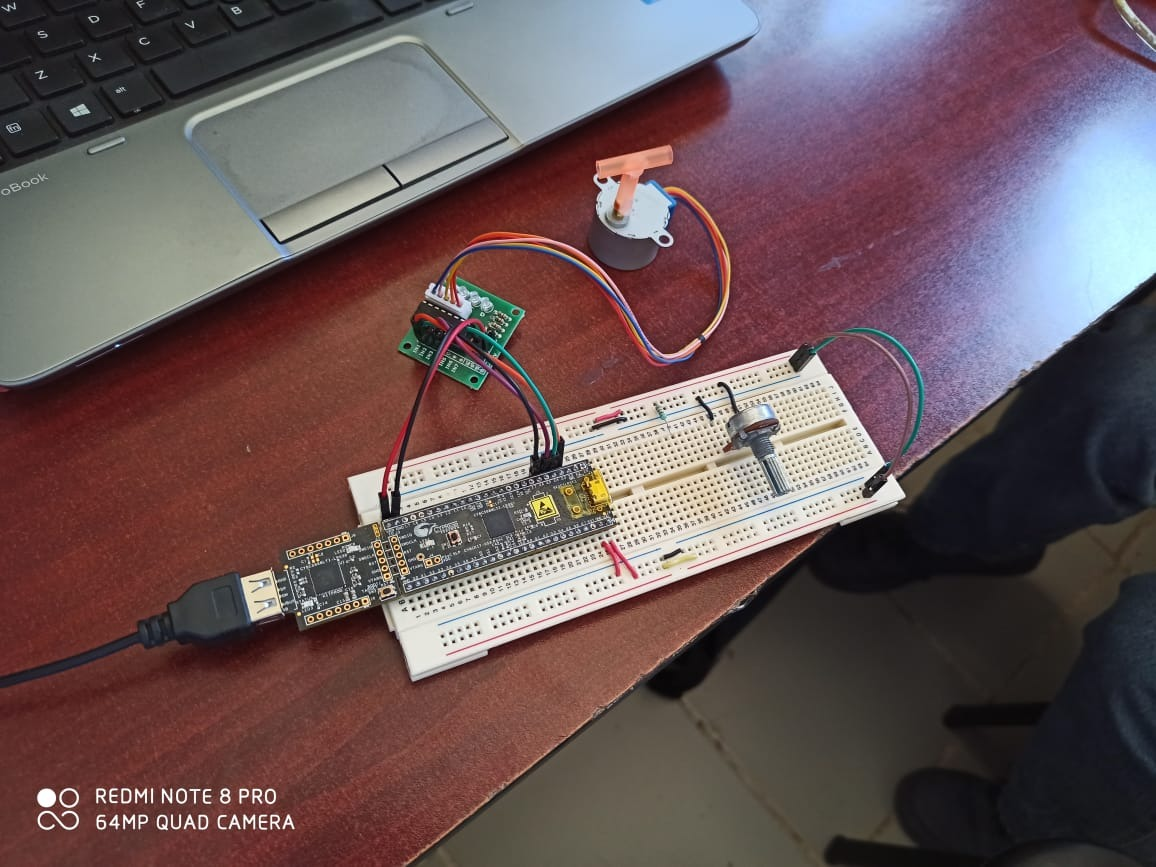
}

}

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

**Resultados:**





**CONCLUSIÓN:**

En esta práctica aprendimos acerca de la comunicación UART y a darle un uso especifico como lo es el controlar un motor a pasos lo cual es muy conveniente para nuestros proyectos anuales, ya que nuestro robot cuenta con múltiples motores a pasos. También la comunicación UART será empleada en el proyecto ya que nos ayudará a efectuar la comunicación de la tarjeta con ROS.