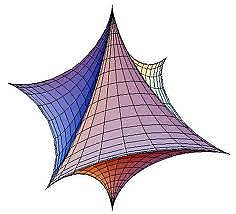


**3\_5\_UART\_Tx\_Rx\_SPI\_I2C**



.

Alumno: Fonseca Camarena Jonathan

Ingeniería Mecatrónica 8-A

Matricula 17311397

Tlajomulco de Zúñiga, Jalisco 25020/2019

MAESTRO: Morán Garabito Carlos Enrique

Marco teórico

La Psoc se conecta a nuestro ordenador a través del puerto USB, pero el puerto USB se debe conectar al microcontrolador a través del puerto serie, por ello debemos entender como están relacionados el puerto USB y el puerto serie.

En una Psoc usamos el puerto USB para dos funciones: cargar nuestro programa ya compilado en la memoria flash y conectarnos al puerto Serie (UART) predefinido en cada Psoc para comunicarnos durante la ejecución del programa. Ambas cosas se pueden hacer sin la necesidad del puerto USB, pero dada la facilidad de uso y que todos los ordenadores disponen de un puerto USB, nos facilita mucho hacer estas dos operaciones.

El puerto serie conectado al USB lo usamos como puerto de consola o puerto de debug.

UART

UART, son las siglas en inglés de Universal Asynchronous Receiver-Transmitter, en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

Un UART dual, o DUART, combina dos UART en un solo chip. Existe un dispositivo electrónico encargado de generar la UART en cada puerto serie. La mayoría de las computadoras modernas utilizan el chip UART 16550, que soporta velocidades de transmisión de hasta 921,6 Kbps (Kilobits por segundo). Las funciones principales de chip UART son: manejar las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

Procedimiento

Las funciones principales de chip UART son de manejar las interrupciones de los dispositivos conectados al puerto serie y de convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

El UART normalmente no genera directamente o recibe las señales externas entre los diferentes módulos del equipo. Usualmente se usan dispositivos de interfaz separados para convertir las señales de nivel lógico del UART hacia y desde los niveles de señalización externos.

Objetivo.

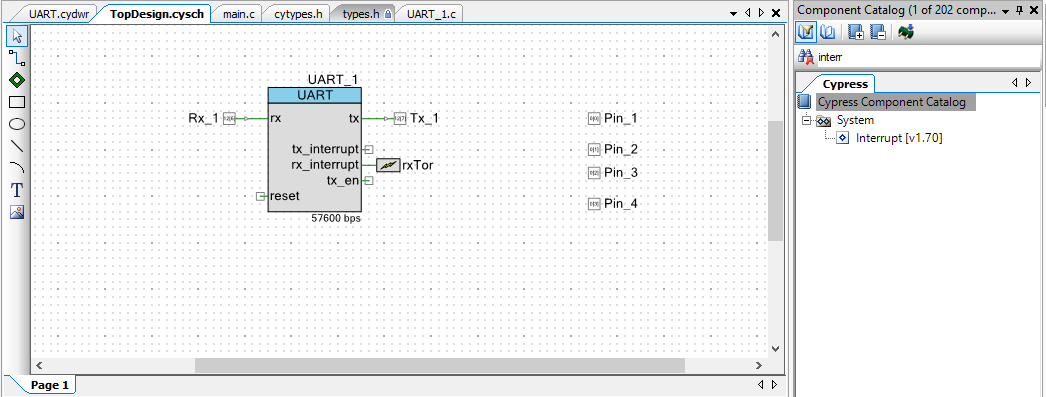
Lo que se realizara es controlar por medio de una terminal que se encuentra en nuestra computadora personal, ordenar al a tarjeta Psoc que gire los grados deseados del motor a pasos.

Los requisitos es que pongamos los grados deseados y mediante un PUSH botón en forma de enter, ejecutemos los ángulos antes ingresados.

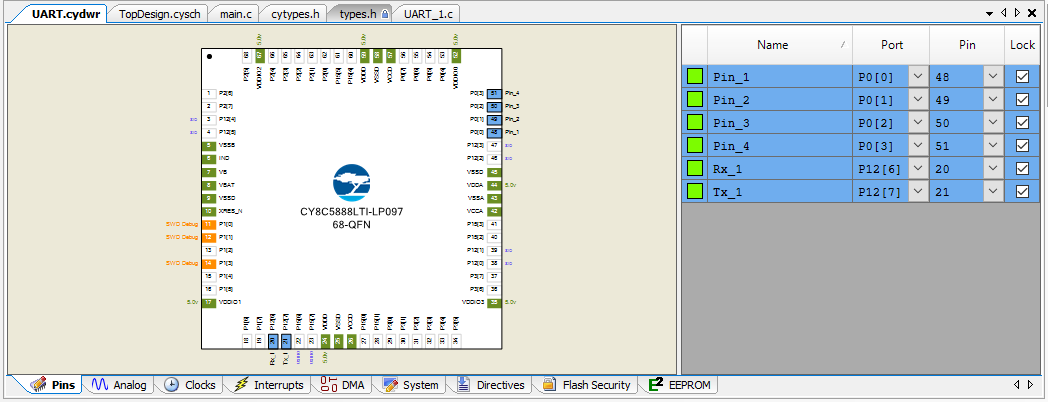
En la terminal deberá estar nuestro nombre y una instrucción que nos diga los ángulos.

De la misma forma debemos poner un tope que nosotros queramos y ángulos positivos y ángulos negativos.

1.En el primer paso vamos a crear nuestro diagrama en el TopDesing en nuestro workspace. En el cual lo primero es buscar el componente cypress UART\_1, en el cual tiene dos conectores simbólicos, uno es el Rx el cual es el que recibe la información y el Tx el que la trasmite de la terminal hacia la tarjeta, medien ante una interrupción para que el motor se mueva en el momento deseado.

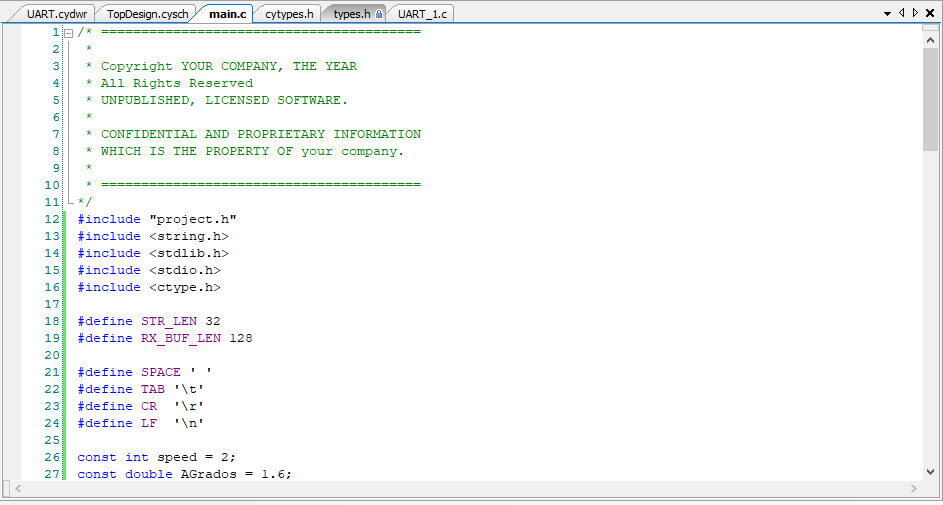


2.En este paso lo que prosigue es declarar los pocos pines que se necesitan para esta práctica, físicamente son solo cuatro pines, de los cuales son los que van a controlar la rapidez y la dirección del motor.

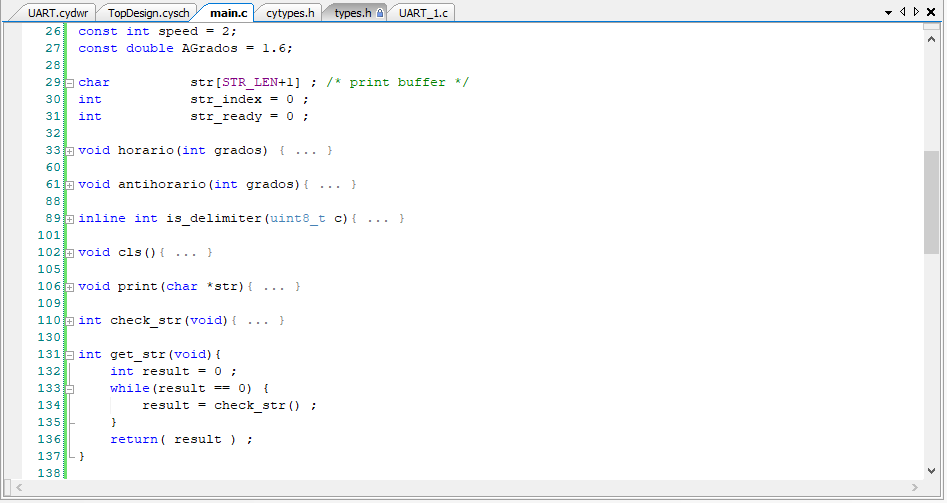


3.Ya que tenemos nuestro diagrama de diseño y declaro nuestros pines, el siguiente paso es empezar a crear nuestro código. Es importante buscar en las librerías de programa buscar los comandos correctos para poder hacer funcionar el motor.

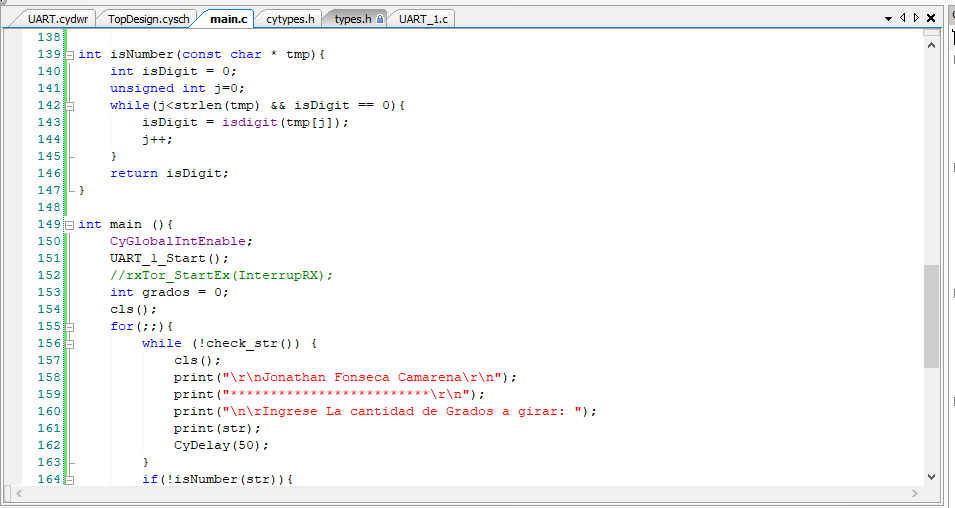
Como se muestra abajo, se llamaran seis librerías para poder ejecutar correctamente el código.



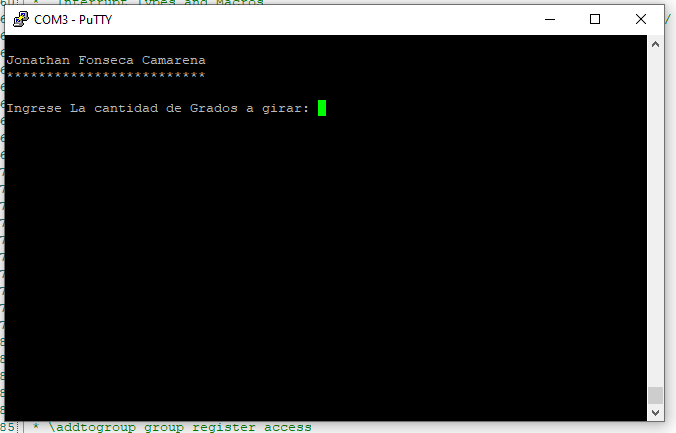
Se definirán los términos y caracteres que utilizaremos en nuestra terminal, es importante que algunas variables que utilicemos sean variables de números y otras variables movibles, esto quieres decir que dependiendo de la función, pueden tomar valores diferentes.



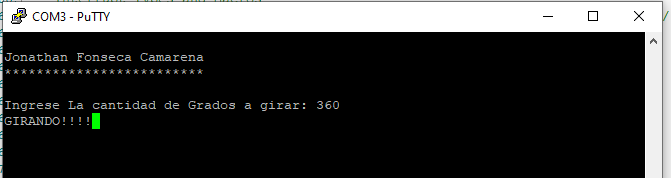
Proseguimos con el código utilizando varios void, por que no lo hacemos infinito, solo funcionara cuando nosotros lo ordenemos en la terminal.



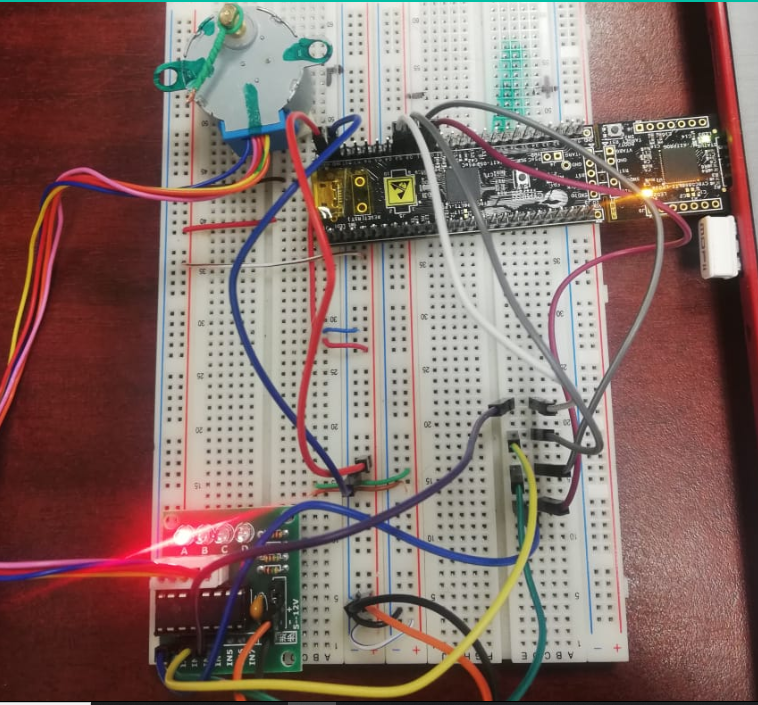
Como podemos observar aquí, es cuando ejecutamos el código “UART\_1\_Start” empezando la conexión. Es importante conectarlos en el mismo COM, en mi caso mi puerto se llama COM3, y los bit por segundo son de 57600, si por algo no concuerda esto con el programa terminal, mandara información errónea y no funcionara.



Al comprobar que la terminal se imprime correctamente, pues solo es cuestión de poner los primeros ángulos para ver si el motor que previamente está conectado correctamente funciona.



Agregamos los 360 y presionamos enter y podemos observar como los LEDS del Driver encienden al mismo tiempo que gira el motor.

Se prosigue a probar con mas ángulos:

Conclusiones:

Para nuestros proyectos futuros esta práctica nos dio las herramientas para tener comunicación de dos dispositivos, mediante varios estándares de comunicación, que determinara cual usar dependiendo del proyecto que estemos trabajando, estos factores son distancia, velocidad de comunicación etc., en mi caso fue 57900 bits por segundo. El código fue complicado, ya que es difícil llamar a una terminal para interrumpa y controle el programa de la tarjeta en tiempo real. Pero ya con esta práctica tenemos las bases para controlar la velocidad y ángulos de nuestros motores.