

CUADERNO DE PRÁCTICAS

DISEÑO SISTEMAS ELECTRÓNICOS
2015-2016

José Antonio Saborido Martín

Cristóbal López Peñalver

Sofía Fernández Moreno



ugr

Universidad
de **Granada**

Índice

1. <u>Práctica 1</u>	3
2. <u>Práctica 2</u>	5
3. <u>Práctica 3</u>	8
4. <u>Práctica 4</u>	10

1. Práctica 1

• Iluminación de un LED mediante señales PWM

Implementar un sistema que haga parpadear dos LEDs. Uno mediante la propia señal PWM y otro mediante una interrupción excitada por la señal PWM, además, este último LED modificará su frecuencia de parpadeo en función de si uno de los botones está o no pulsado. Manteniéndose a 5Hz cuando el botón no esté pulsado, y disminuyendo a 0.5Hz cuando si lo esté.

En este ejercicio se va a crear un sistema de cronometraje que permitir completar unas tareas (encender un Led) cada milésima de segundo, cada centésima de segundo y cada segundo. El sistema temporal consiste en un Timer que alcanza su máximo y activa la interrupción cada milésima de segundo. Se debe de hacer uso de flags para notificar al bucle principal que ha pasado un determinado intervalo temporal.

El utilizar dichos flags nos permitirá usarla en otras partes del firmware para saber que tarea debemos ejecutar. Si tratamos de ejecutar una tarea completa dentro de una interrupción en lugar de usar flags se pueden dar problemas de desbordamiento de la interrupción o sincronismo.

A modo de resumen, el sistema electrónico a diseñar consistirá:

- Un reloj de sincronización del sistema a 100kHz.
- Un señal PWM de 1ms de periodo con el duty cycle elegido por alumno. Este duty cycle determinará la luminosidad del LED conectado a la salida del PWM.
- Un LED excitado por la señal PWM (PWM_Pin).
- Un LED excitado por la interrupción del PWM (LED_Pin).
- Un botón que modifique la frecuencia de parpadeo del LED (LED_Pin).

El esquemático queda así:

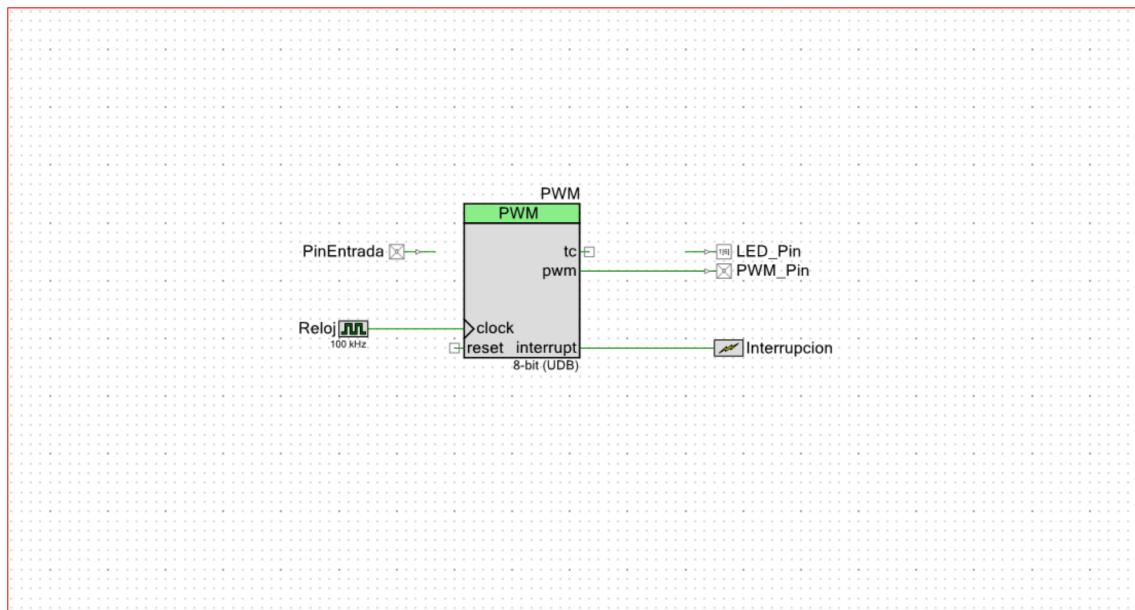


Ilustración 1 Esquemático Iluminación LED mediante señal PWM

El LED_Pin quedaría configurado de la siguiente manera:

- Coloque una salida digital Pin de componentes en el esquema del proyecto.
- Llamar el componente "LED_Pin" y deshabilitar la conexión de hardware.
- Habilitar el terminal externo para conectar a los componentes externos en el esquema.
- Asignar a un pin el correspondiente LED_Pin(P1[6])
- Conectar el pin a un LED.

El módulo PWM será utilizado para generar las interrupciones del LED mediante PWM_Pin.

Utilizamos un botón para marcar la frecuencia con la que se actualiza uno de los LEDs.

Limitaciones de diseño: El PWM por defecto comienza con una frecuencia de 10 ciclos. Y sólo podrá tomar valor 10, 100 y 1000. Cuando pulsemos el botón PinEntrada aumentará de 10 a 100, de 100 a 1000 o disminuirá de 1000 a 10. Por lo que será necesario la presencia de un contador para medir cuando encender nuestro LED.

2. Práctica 2

▪ Multiplicador de 2 bits

En esta práctica se busca implementar un multiplicador de dos bits con tres botones, uno para incrementar el valor, otro para decrementarlo y otro para seleccionar la variable que vamos a modificar o el resultado. En mi caso he colocado un * debajo de la variable que esté seleccionada, o se muestra directamente el resultado en caso de ser el seleccionado.

Para crear el circuito del multiplicador utilizaremos el programa LogiSim en el que introduciremos la tabla que aparece en el guion de prácticas.

Los elementos que utilizaremos son los siguientes:

- Pantalla LCD para mostrar los resultados
- 2 registros de control, uno para una variable y el otro para la otra
- Registro de estado para el resultado
- Circuito de puertas lógicas producido por LogiSim

El esquemático quedaría como sigue:

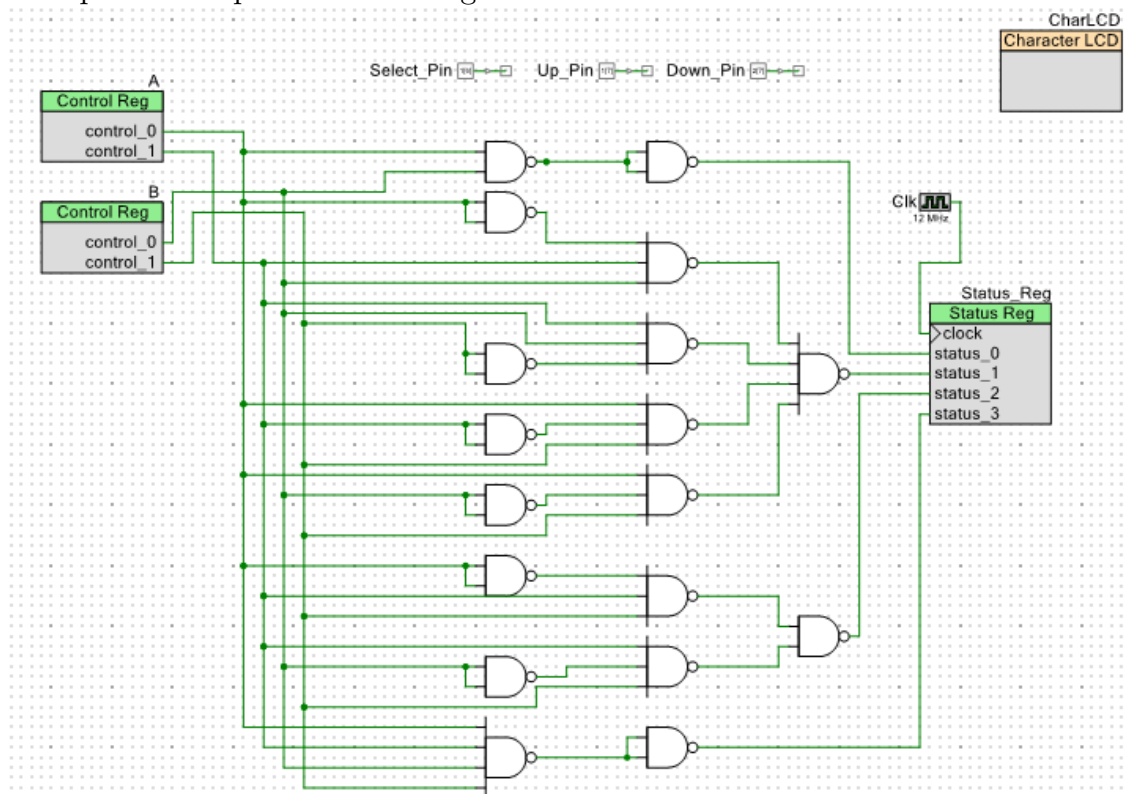


Ilustración 2 Esquemático Multiplicador 2 bits

Limitaciones del diseño: Las dos entradas son de 2 bits (4 valores distintos), por lo tanto el resultado será de 16 valores diferentes.

▪ Acertijo

En esta práctica se pide realizar el acertijo del Granjero-Zorro-Gallina-Maíz que tiene que cruzar el río. La idea consiste en que el zorro y la gallina no estén en el mismo lado sin el granjero, al igual que la gallina y el maíz. Para esto, he implementado una look-up-table para las soluciones en las que una salida es el juego finalizado por error (cuando vale 0) y la otra es el juego ganado (cuando vale 1). Aunque realmente el valor de salida no será 0 o 1, puesto que como son 2 salidas, el valor será 00, 01, 10 o 11. Cuando valga 00 el juego no continuará, con valor 01 el juego se habrá ganado, y si vale 10 se mostrará por pantalla el estado actual.

En la tabla considero el valor 0 de cada elemento a un lado del río, y el valor 1 al otro lado del río. Para ganar necesitamos tener todo “al otro lado del río” por lo que el valor de todos los elementos (Granjero, Gallina, Maíz y Zorro) debe ser 1.

Para realizar la práctica he utilizado los siguientes elementos:

- Módulo LCD
- Registro de Control
- Look-up-table
- Registro de estado
- 4 pines de entrada correspondientes a cada elemento del acertijo

El esquemático que queda sería el siguiente:

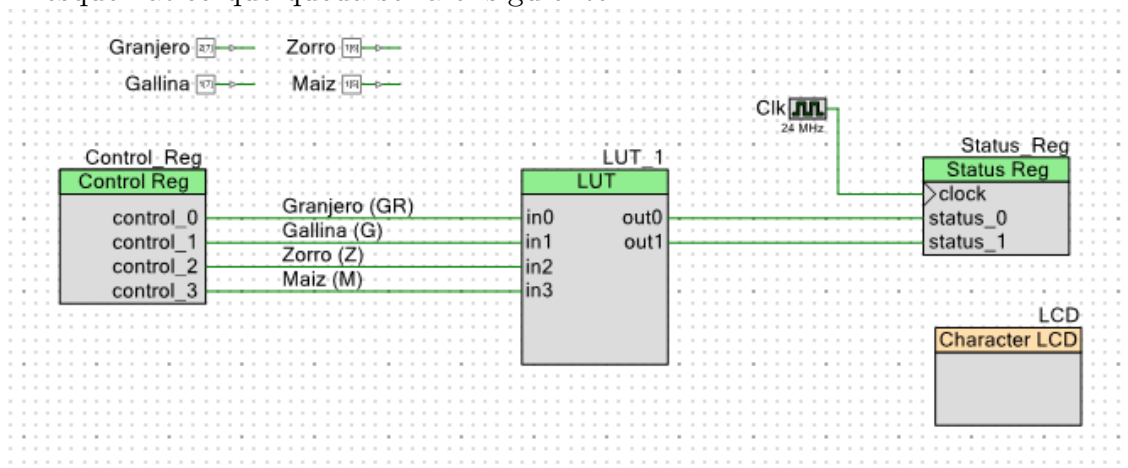


Ilustración 3 Esquemático Acertijo

Los elementos del esquemático quedan enlazados con los siguientes pines:

Alias	Name	Port	
	\LCD:LCDPort[6:0]\	P2[6:0]	▼
	Gallina	P1[7]	▼
	Granjero	P2[7]	▼
	Maiz	P1[5] JTAG:ntrst	▼
	Zorro	P1[6]	▼

Ilustración 4 Pines asignados

En todo momento se está comprobando si se pulsa uno de los botones correspondientes con los elementos. Si se pulsa se comprueba si da a un estado de error en el que no se puede seguir el juego, si es así el juego termina, se resetea la posición de los elementos y se reinicia el juego a los 5 segundos. Por el contrario, si se puede seguir en el juego se muestra por pantalla el estado actual y se sigue esperando a que se pulse otro botón.

Como ya se ha explicado anteriormente, el registro de estado marca cuando finaliza el juego correctamente o por error, por lo que hay que comprobar dicho estado continuamente y realizar la acción correspondiente.

Hay que tener en cuenta que el granjero puede moverse sólo de un lado al otro del río, es decir puede ir sin compañía. Sin embargo tanto el zorro, como la gallina o el maíz, deben cruzar con el granjero. Para que esto quede reflejado, cuando se pulsa el botón del granjero, sólo se actualizará el estado de dicho elemento. Para el resto de elementos cuando se pulse su botón, se actualizará su estado y el del granjero.

Limitaciones del diseño

Las limitaciones las pone el usuario. Él es el que decide la acción que va a realizar, sea correcta o errónea.

3. Práctica 3

En esta práctica se pide realizar el mismo problema de la práctica 1 pero utilizando un USB-UART. Nos comunicaremos con él a través del USB, pero mandaremos las órdenes con un HyperTerminal.

El esquemático queda tal que así:

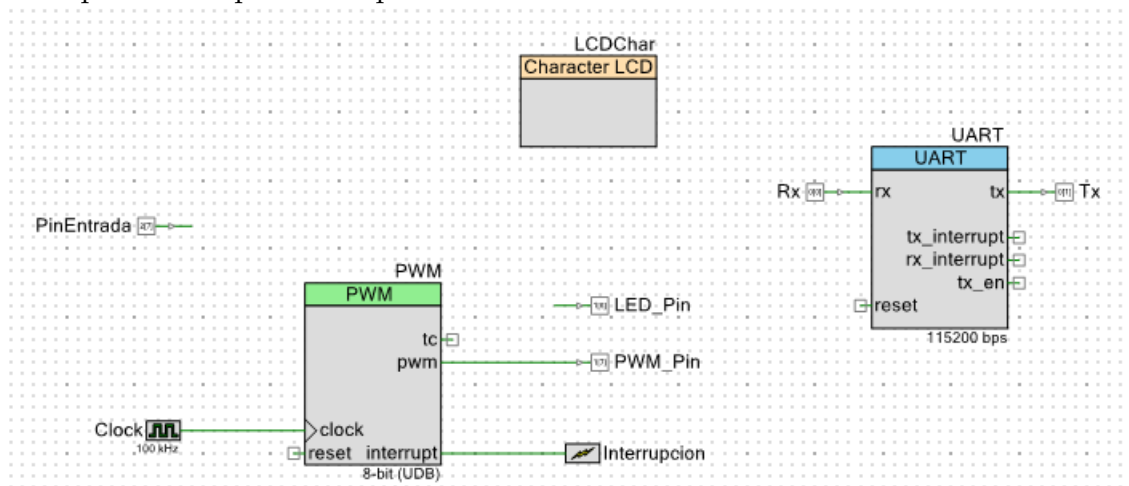


Ilustración 5 Esquemático

- El módulo UART, utilizado para la comunicación, está configurado tal y como indica la práctica:
 - 115200 Baudios
 - 8 bits de datos
 - Sin paridad
 - Sin control de flujo
 - Reloj interno
 - Interrupciones deshabilitadas
 - Buffers de 4 Bytes
 - Transmisión hardware para RS-485
- El módulo PWM, utilizado para generar las interrupciones del LED 'PWM_Pin'.
- Reloj de 100 kHz.
- Pin de entrada para el parpadeo del LED 'LED_Pin' (como en la otra práctica).

Funcionamiento

En la práctica 1 utilizábamos un botón para marcar la frecuencia con la que se actualizaba uno de los LEDs, LED_Pin en esta práctica. Este botón vamos a dejarlo funcionar igual.

Lo que cambiamos a través del HyperTerminal es el periodo del PWM con el que vamos a encender el LED correspondiente. Para ello se han designado las siguientes funciones:

- Encender LED PWM – A través de las teclas “e” ó “E”.
- Apagar LED PWM – A través de las teclas “a” ó “A”.
- Bajar periodo PWM 20% - A través de las teclas “b” ó “B”.
- Subir periodo PWM 20% - A través de las teclas “s” ó “S”.

Limitaciones del diseño

El PWM por defecto comienza con una frecuencia de 10 ciclos. Y sólo podrá tomar valor 10, 100 y 1000. Cuando pulsemos el botón PinEntrada aumentará de 10 a 100, de 100 a 1000 o disminuirá de 1000 a 10.

Diagrama de Estados

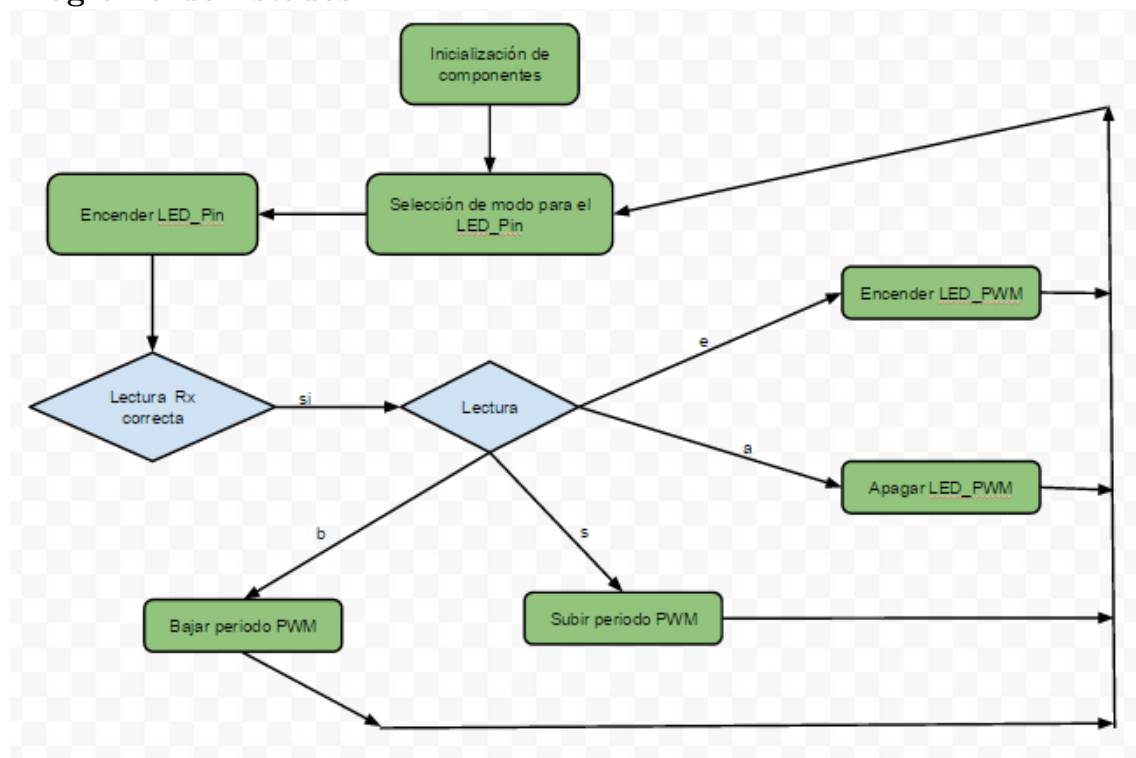


Ilustración 6 Diagrama de estados Práctica 3

4. Práctica 4

▪ Voltímetro

El objetivo de esta práctica es realizar un voltímetro digital para muestrear tensiones entre 0V y 3.3V. Para ello utilizamos los siguientes elementos:

- Dos pines para conectar la tensión, uno para el positivo y otro para el negativo
- Pantalla LCD para mostrar el valor obtenido
- Conversor de analógico digital (ADC)

El esquemático sería el siguiente:

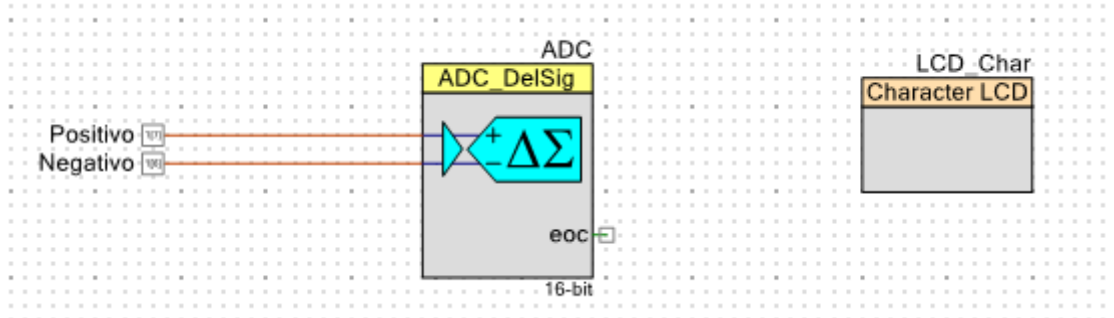


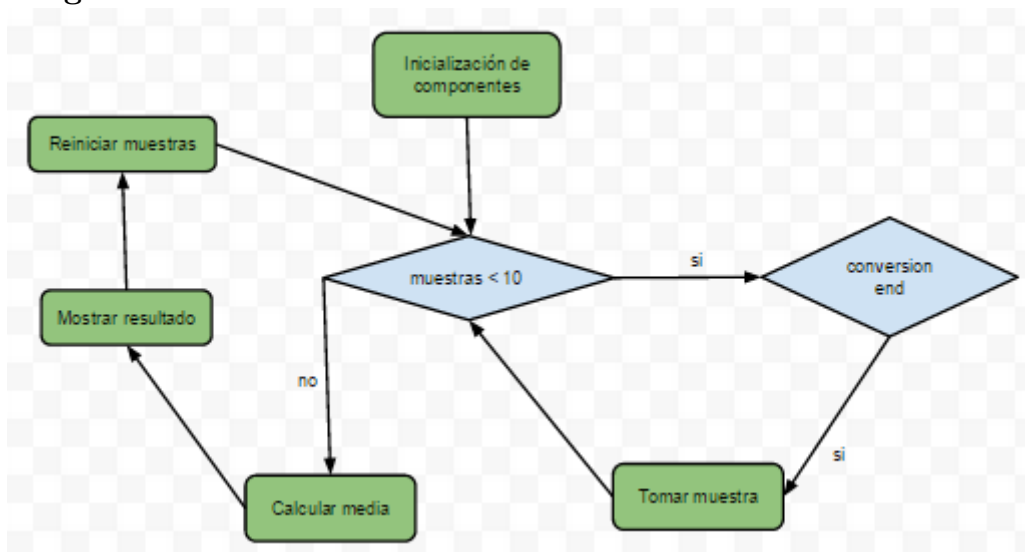
Ilustración 7 Esquemático Voltímetro

En el código se lee el valor del conversor en un entero sin signo de 16 bits, para después mostrarlo por la pantalla LCD con un retraso para q se lean bien los dígitos del valor del voltaje medido. Se expresan en miliVoltios, anteriormente convertida la muestra a dicha unidad.

Limitaciones del diseño

La limitación la pone la capacidad de entrada de voltaje que tiene el PSoC.

Diagrama de estados



▪ Generador de Funciones

Esta práctica busca que el PSoC genere una serie de señales diferentes, así como aumentar o disminuir la amplitud o la frecuencia de dicha onda. Para esto son necesarios los siguientes elementos:

- 2 convertidores de señal Digital a Analógico (WaveDAC8) para generar las 4 señales, puesto que cada convertidor es capaz de generar dos tipos de señales distintos
- Pantalla para mostrar la amplitud y frecuencia y modificarlas
- PWM para poder modificar la frecuencia de la señal de reloj
- Un pin de salida por el que sacar la señal generada
- Un módulo UART para conectar por Bluetooth y enviar desde ahí un carácter para generar un tipo de señal u otro

El esquemático sería el siguiente:

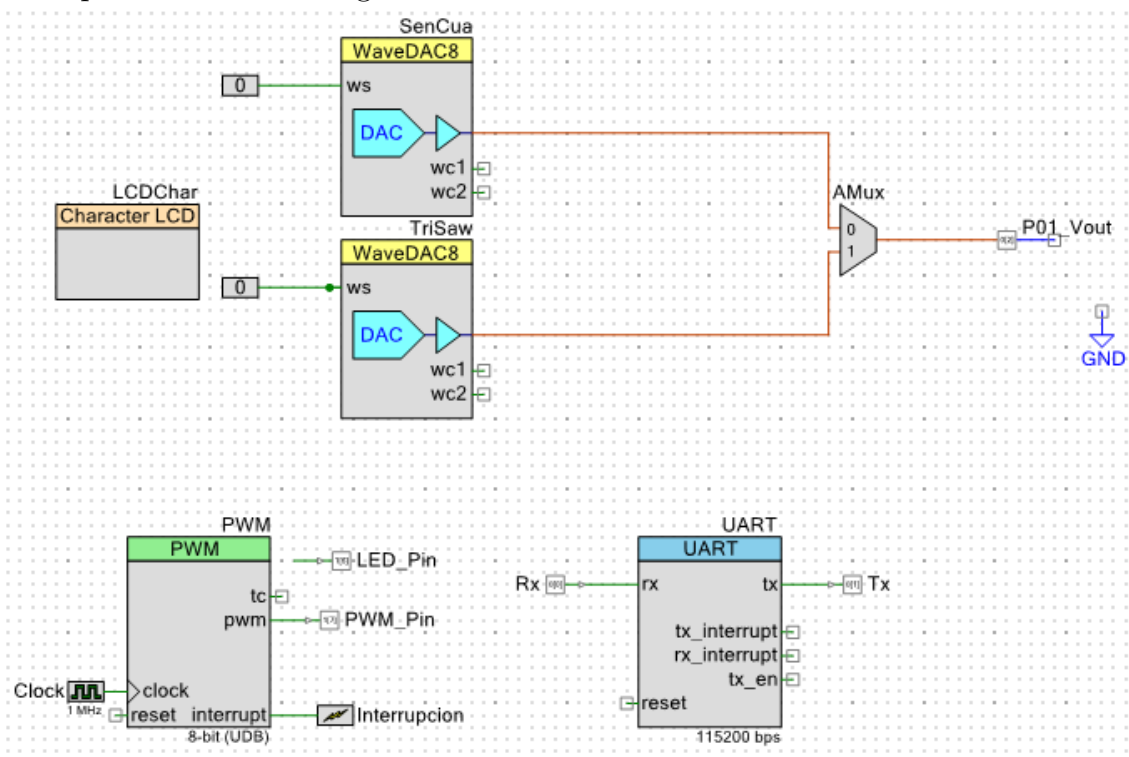


Ilustración 8 Esquemático Generador de funciones

Bibliografía

- [1] <http://www.cypress.com/>