INFORME LAB. ELECTRÓNICA II

SOFTWARE PARA ENVIÓ Y RECEPCIÓN DE HARDWARE

CRISTIAN STEVEN CHAVARRO RICO

SERGIO ALEXANDER FLOREZ GALEANO

PRESENTADO

RAMIRO ANDRES BARRIOS VALENCIA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERIA DE SISTEMAS Y COMPUTACION

LABORATORIO ELECTRONICA 2

PEREIRA, JULIO 2012

Tabla de contenido

[Introduccion 3](#_Toc327177169)

[Desarrollo de la Práctica 4](#_Toc327177170)

[Elementos Utilizados para el desarrollo: 4](#_Toc327177171)

[Diseño del Hardware 5](#_Toc327177172)

[Conclusiones 6](#_Toc327177173)

[Bibliografía 7](#_Toc327177174)

# INTRODUCCION

Para manejar un buen procesamiento de una cantidad grande de datos se necesitaban un medio el cual las almacenara y así poderlos administrar. Gracias a este pensamiento surgieron ideas brillantes la cuales se desarrollaron hasta llegar a la Memoria RAM; esta nos permitía grabar y acceder a los datos de forma rápida y así poder controlar su uso de estos. Para ele presente informe trata el tema manejo de memorias en la FPGA, las cuales serán cruciales para el desarrollo final del curso. El paso de datos a los servomotores, se leerán de 2 memorias RAM, estas contendrán 256 datos, con valores aleatorios de 0 a 255, los cuales serán determinaran el ángulo de giro de los servomotores en un momento dado.

# Desarrollo de la Práctica

Para desarrollar la interfaz de usuario se usara el lenguaje Orientado a objetos JAVA, el cual nos permite modelar de forma más fácil el mundo real. El objetivo del programa es generar una cantidad *n* de datos aleatorios los cuales serán enviados a una tarjeta de desarrollo (FPGA), con una UART descrita en ella, posteriormente guardarlos en las memorias RAM de la misma, y posteriormente mostrar el dato guardado en las memorias en los siete segmentos de la FPGA.

Para fines prácticos usaremos los 8 leds para mostrar la posición de la memoria RAM y los push buttons para avanzar o retroceder en las memorias.

# Elementos Utilizados para el desarrollo:

**Librería:** La librería escogida para realizar la conexión serial Se llama Giovynet, es una librería creada hacer observación, medición y control de sistemas electrónicos usando Tecnologías de Java. De esta librería se usaron los paquetes de ***SerialPort, Parameters, Com.***

* ***SerialPort:*** Esta clase nos permite crear una conexión a el puerto serial para el envió de datos.
* ***Parameters:*** Esta clase determina los parámetros que se vana usar en la comunicación serial.
* ***Com:*** Esta clase representa el puerto serial, al cual se le asignan los parámetros deseados.

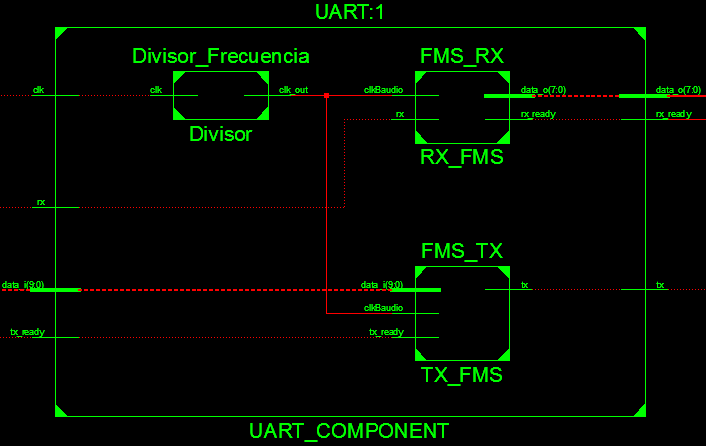
Esta librería se escogió por su fácil manejo y el rápido entendimiento de su de las clases.

**Código Fuente:** Adjunto al Documento. Serial.java

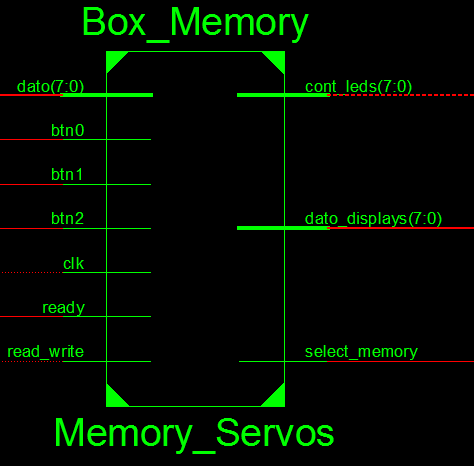
**Inconvenientes:** Algunos inconvenientes que ocurrieron al realizar este modulo fue la limitación de la librería 32 bits, ya que el equipo personal solo tenias S.O. E 64 bits.

# Diseño del Hardware

Para la implementación de las memorias RAM, que contendrán los 256 datos de los de los servomotores, usaremos el diseño inicial del UART, como se muestra a continuación.



Por medio de esta se recibirán los datos por el puerto serial y se guardaran en 2 memorias RAM contenidas en el siguiente modulo.



Este modulo se encargara de recibir el dato(dato(7:0)) enviado por la UART y almacenarlo en las RAM, al activarse la señal del rx\_ready, el modulo además contendrá un comportamiento adecuado para leer y guardar los datos.

Por medio de un switch(read\_write) se activara el modo lectura de memorias cuando la señal se encuentre en 1, y el modo Guardar en memorias cuando la señal se encuentre en 0.

Switch(read\_write)=0 GUARDAR DATOS EN MEMORIAS

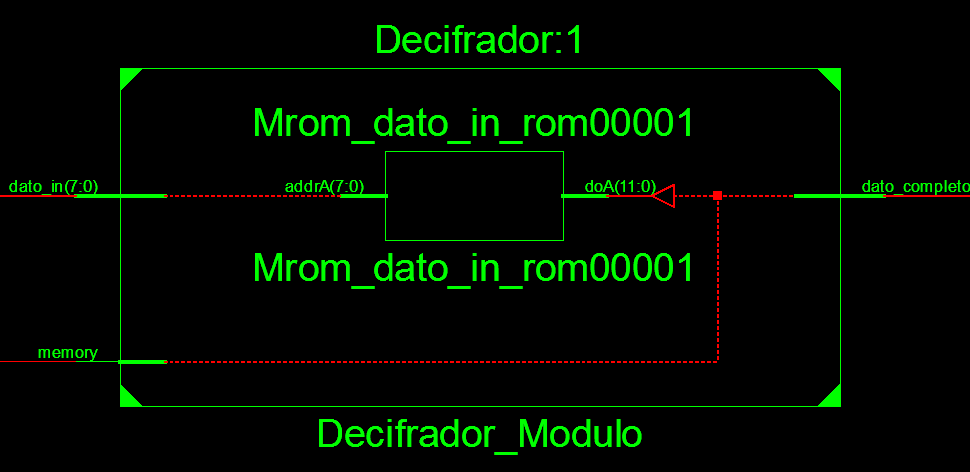
En este modo no se mostrara nada en los siete segmentos, solo ceros, y cada que se reciba una señal 1 en el ready del RX se guardara un dato en las memorias de la FPGA.

Switch(read\_write)=1 LEER DATOS DE MEMORIAS

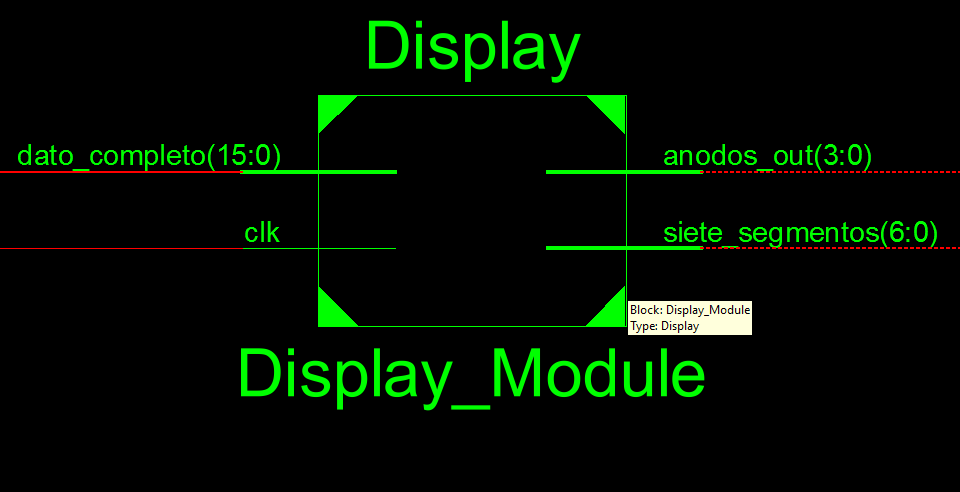
En este modo se mostrara en los siete segmentos, el dato almacenado en la memoria RAM en la posición dada por los registros que almacenan los contadores que especifican la posición en las memorias RAM.

Para la correcta codificación del dato de los logic\_vector a los siete segmentos, se usan 2 modulos.

1. DESCIFRADOR (Modulo con memorias): En este modulo existen 3 memorias ROM, cada una con 256 posiciones de memorias las cuales especifican las unidades, decenas y centenas del dato, con el fin de facilitar la codificación del dato a los siete segmentos.

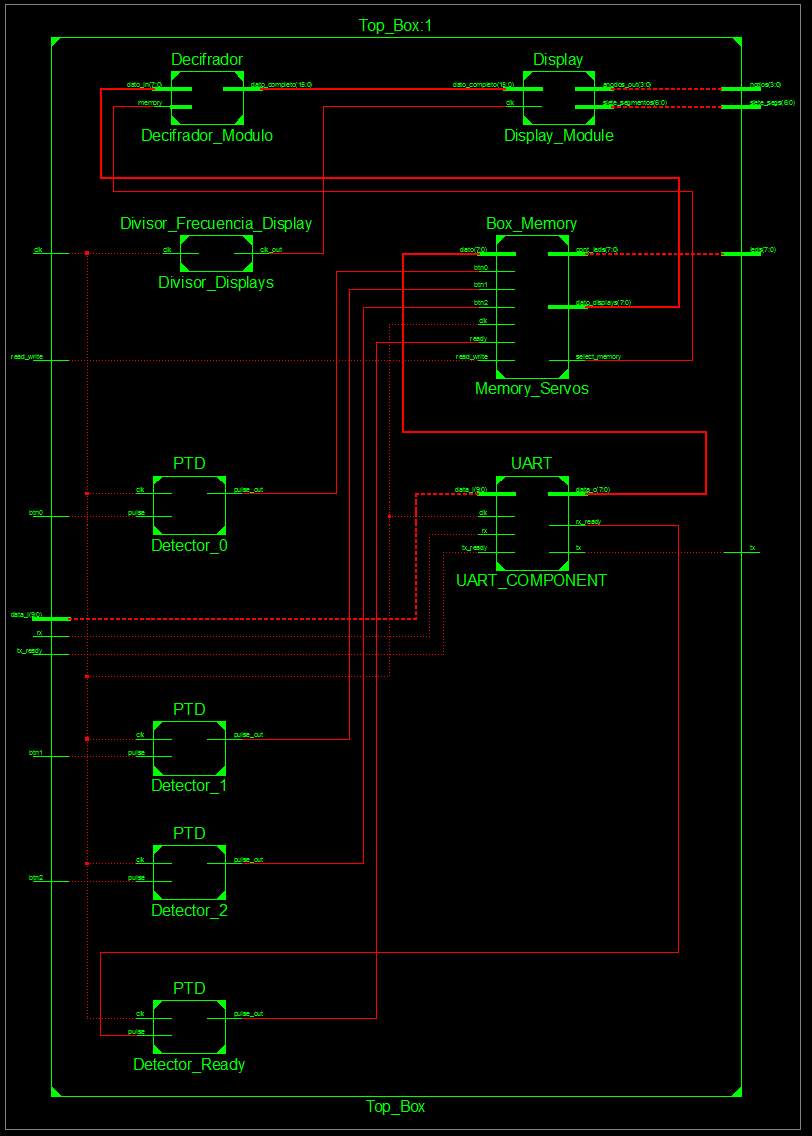


1. DISPLAYS (Modulo que convierte el dato a siete sementos): Como su nombre lo indica se encarga de convertir el dato a su correspondiente codificación para ser mostrado en los siete segmentos.



La combinación de todos estos elementos hacen posible el correcto funcionamiento del enviado de datos desde el software, y guardado de datos en las memorias.

El hardware también incluye detectores de flancos para los botones, y un divisor de frecuencia para la frecuencia de refresco de los siete segmentos.



# Conclusiones

El manejo de memorias RAM para el almacenamiento de los datos nos permitirá tener una organización de envío del dato a los servomotores.

# Bibliografía

Giovanynet.com – Java Electronic Solutions

<<http://www.giovynet.com/>>

LAS PATAS DEL CHIP. Héctor Eduardo Ugarte Rojas. Envió y recepción de datos a través del puerto serial de la pc - Aplicación: conexión entre 2 pcs, y pruebas haciendo uso de java

<<http://es.scribd.com/doc/49480537/45559031-ENVIO-Y-RECEPCION-DE-DATOS-A-TRAVES-DEL-PUERTO-SERIAL-DE-LA-PC-APLICACION-CONEXION-ENTRE-2-PCS-Y-PRUEBAS-HACIENDO-USO-DE-JAVA>>

Giovanynet Artículos. Manejo de Puertos Seriales RS-232 con Java para Windows

<<https://sites.google.com/site/articulosgiovynet/manejo-de-puertos-seriales-rs-232-con-java--para-windows>>