



Parcial 1 diseño digital

Jhonatan Alexander Gomez Gamboa

2023-03-06

1. Introducción experimental.

La creación e implementación del siguiente diseño digital tiene como función la implementación del protocolo de comunicación serial UART (Universal Asynchronous Receiver/Transmitter) el cual se utilizará para comunicar una computadora de escritorio con el modulo UART-RS232 implementado en un FPGA (Field Programming Gate Array) el cual tendra dos modos de transmisión utilizado ocho switches o al retransmitir la salida de la memoria, ademas de tener la capacidad de recibir como cadena hasta 6 bytes, almacenarlos en la memoria, mostrarlos y retransmitirlos, con la posibilidad de cambiar sus características de comunicación, los display 7 segmentos permitira ver estas características en tiempo real.

2. Ambiente experimental.

Quartus (Quartus Primer 17.1 Lite Edition)

FPGA DE10-Lite 10M50DAF484C7G

Modulo usb-serial communication.

Computadora (con serial monitor program)

#3. Programa

A continuación se muestra el módulo top el cual fue el diseño montado en el FPGA, donde podemos ver los diferentes bloques que lo conforman en la figura 1, explicaremos mas adelante el funcionamiento de cada uno de los módulos por separado. La intención del diseño es implementar comunicación serial diseñando un módulo UART RS232.

Table 1: Entradas y salidas del modulo TOP DUT (UART)

Señal	Dirección	Tamaño	Descripción
<i>clk_top</i>	in	1	Señal de reloj interna del FPGA de 50MHz
<i>rst_top</i>	in	1	Señal que permite inicializar todos los registros
<i>data_insw</i>	in	8	Datos de 8 switches(1 byte) caracter a transmitir
<i>ena_top</i>	in	1	Señal de habilitación para realizar acciones
<i>sel_top</i>	in	2	Señal de entrada de 2 switches para cada acción
<i>rxin_top</i>	in	1	Señal de entrada al módulo desde la pc (serial)
<i>txout_top</i>	out	1	Señal de salida del módulo top a la pc (serial)
<i>HEX0_top</i>	out	8	señal de salida del módulo al 7 segmentos 0
<i>HEX1_top</i>	out	8	señal de salida del módulo al 7 segmentos 1
<i>HEX2_top</i>	out	8	señal de salida del módulo al 7 segmentos 2

Señal	Dirección	Tamaño	Descripción
<i>HEX3_top</i>	out	8	señal de salida del módulo al 7 segmentos 3
<i>HEX4_top</i>	out	8	señal de salida del módulo al 7 segmentos 4
<i>HEX5_top</i>	out	8	señal de salida del módulo al 7 segmentos 5

Este módulo mostrado en la figura 2 se trata de un multiplexor que nos permite realizar varias acciones utilizando unicamente dos switches y un botón para hacer efectivos estos cambios, esto con la finalidad de ahorrar en el número de entradas ya que en la mayoría de los casos se utilizan contadores para que con un mismo botón podamos cambiar los parametros de la misma característica.

Utilizando el sel en 0 modificamos el baudrate, en 1 podemos cambiar del modo de transmisión si bien podemos hacerlo desde la memoria para reenviar todos los datos guardados tambien podemos hacerlo con los datos de entrada utilizando los switches.

Table 2: Entradas y salidas del modulo muxtwofour

Señal	Dirección	Tamaño	Descripción
<i>ena</i>	in	1	entrada que permite realizar los cambios al oprimirlo
<i>sel</i>	in	2	dependiendo de las combinaciones permite cambiar valores
<i>selbaud</i>	out	1	permite habilidad el cambio de baudage
<i>selmodtx</i>	out	2	permite cambiar el modo de transmisión
<i>seltxff</i>	out	3	permite transmitir la salida de la memoria
<i>seltx</i>	out	4	permite transmitir las entradas de los switches

El módulo mostrado en la figura 3 se trata del generador del baudrate, este al recibir una referencia en sus entradas podemos tener a la salida los respectivos baudrates de interes como lo son 9600 al tener una referencia de 324, 57600 al tener una referencia de 35, y 115200 al tener una referencia de 23, y sacar estas frecuencia de transmisión en la salida.

Table 3: Entradas y salidas del modulo baudgenerator

Señal	Dirección	Tamaño	Descripción
<i>clk</i>	in	1	señal de reloj de 50MHz del FPGA
<i>refer</i>	in	9	señal de referencia para cambiar el baudrate
<i>rst</i>	in	1	señal para inicializar los registros del módulo
<i>tickbd</i>	out	1	señal de salida del baudrate

El módulo mostrado en la figura 4 nos muestra la arquitectura del selector de baudage el cual utiliza una misma entrada para aumentar un contador interno que dependiendo el valor de este registro nos data una de las respectivas referencias. Utiliza una señal de 3Hz para hacer cambios de manera medida y lenta.

Table 4: Entradas y salidas del modulo baudselect

Señal	Dirección	Tamaño	Descripción
<i>rst</i>	in	1	Señal para inicializar los registros
<i>selbaud</i>	in	1	señal de activación para cambiar el baudage
<i>ticksel</i>	in	1	señal de entrada de 3Hz para seleccionar

Señal	Dirección	Tamaño	Descripción
<i>refer</i>	out	9	Señal de referencia la cual se conectara al generador

En el diagrama de la figura 5 podemos ver la arquitectua del modulo prescaler, el cual es un contador que nos permite tener ticks de habilitación para diferentes tareas al variar el valor maximo del contador, en este caso se utilizo para obtener 3Hz utilizando la fórmula siguiente

$$NUM = \frac{Frecuencia_{fpga}}{Frecuencia_{interes}}$$

este número nos dara el numero a contar y para obtener el número de bits necesario utilizamos la formula:

$$\log_2(NUM)$$

Table 5: Entradas y salidas del modulo prescaler

Señal	Dirección	Tamaño	Descripción
<i>rst</i>	in	1	Señal para inicializar los registros del módulo
<i>fastclock</i>	in	1	señal de reloj de entrda de 50MHz del FPGA
<i>slowclock</i>	out	1	señal de salida a menor frecuencia que la entrada

El módulo mostrado en la figura 6 se trata del receptor, el cual fue hecho utilizando una maquina de estados finita, ya que según el protocolo de UART, necesita la trama tener una forma espfecia la cual nos permitira recuperar el dato.

Table 6: Entradas y salidas del modulo rx

Señal	Dirección	Tamaño	Descripción
<i>bdtick</i>	in	1	Señal que tendra el baudage necesario del protocolo
<i>rx_in</i>	in	1	señal que entrara de la computadora para comunicación
<i>rx_rst</i>	in	1	señal que permite inicializar los registros
<i>rx_done</i>	out	1	bandera de salida al recibir una trama con exito
<i>rx_out</i>	out	8	Señal de salida de la trama recibida en paralelo

El módulo mostrado en la figura 7 se trata de la arquitectura del control de escritura utilizado, el cual se realizo utilizando una maquina de estados de Moore, la cual tiene una retroalimentación con la bandera rxdone para poder guardar el dato en la memoria de una manera optima.

Table 7: Entradas y salidas del modulo controlwrite

Señal	Dirección	Tamaño	Descripción
<i>clk</i>	in	1	Señal de reloj la cual sera la del baudrate
<i>datain</i>	in	8	señal de entrada que llega del modulo rx
<i>rst</i>	in	1	señal que permite inicializar los registros
<i>rxdone</i>	in	1	bandera cuando una trama fue capturada con exito

Señal	Dirección	Tamaño	Descripción
<i>addr</i>	out	3	señal utilizada como puntero para la escritura en memoria
<i>dataout</i>	out	8	señal de salida dirigida a la memoria
<i>we</i>	out	1	señal de habilitación de escritura

El módulo mostrado en la figura 8 se trata del módulo onepulse el cual se utiliza como mecanismo para evitar oprimir un botón para realizar una acción y tener varios eventos cuando solo se busca uno solo, hacemos uso de la frecuencia de 3Hz para asegurar esto.

Table 8: Entradas y salidas del modulo onepulse

Señal	Dirección	Tamaño	Descripción
<i>ena</i>	in	1	Señal que representa una logica dada por un botón
<i>tickbd</i>	in	1	señal del baudrate que sale del generador
<i>ticksel</i>	in	1	señal de 3Hz utilizada para obtener un unico alto
<i>flag</i>	out	1	salida logica que dura un ciclo de tickbd para logica

El módulo mostrado en la figura 9 se trata de la memoria RAM de dual port utilizada, se uso la plantilla proporcionada por el entorno de desarrollo Quartus, donde tenemos dos relojes tanto de escritura como de lectura que serán los del FPGA 50MHz.

Table 9: Entradas y salidas del modulo SDP_DCRAM

Señal	Dirección	Tamaño	Descripción
<i>data</i>	in	8	señal de los datos a guardar en la memoria
<i>read_addr</i>	in	3	puntero de dirección para lectura
<i>read_clock</i>	in	1	señal de reloj de 50MHz
<i>we</i>	in	1	señal de habilitación de escritura
<i>write_addr</i>	in	3	puntero de escritura de dirección de escritura
<i>write_clock</i>	in	1	señal de reloj de escritura de 50MHz
<i>q</i>	out	8	datos de salida de la memoria

En la figura 10 podemos ver la arquitectura del módulo control de lectura el cual fue hecho utilizando una maquina de estados moore.

Table 10: Entradas y salidas del modulo contolread

Señal	Dirección	Tamaño	Descripción
<i>rst</i>	in	1	señal que inicializa los registros
<i>tickbd</i>	in	1	señal del baudrate
<i>ticksend</i>	in	1	salida del modulo onepulse para transmitir
<i>txdone</i>	in	1	señal de entrada cuando una trama se envió con éxito
<i>addr</i>	out	3	Señal puntero para las direcciones de lectura
<i>txena</i>	out	1	señal que permite la transmisión con condiciones

El diseño mostrado en la figura 11 se trata del módulo transmisor el cual también fue realizado utilizando una maquina de estados moore, lo cual nos permitira asegurar el correcto diseño de la trama a transitar, este

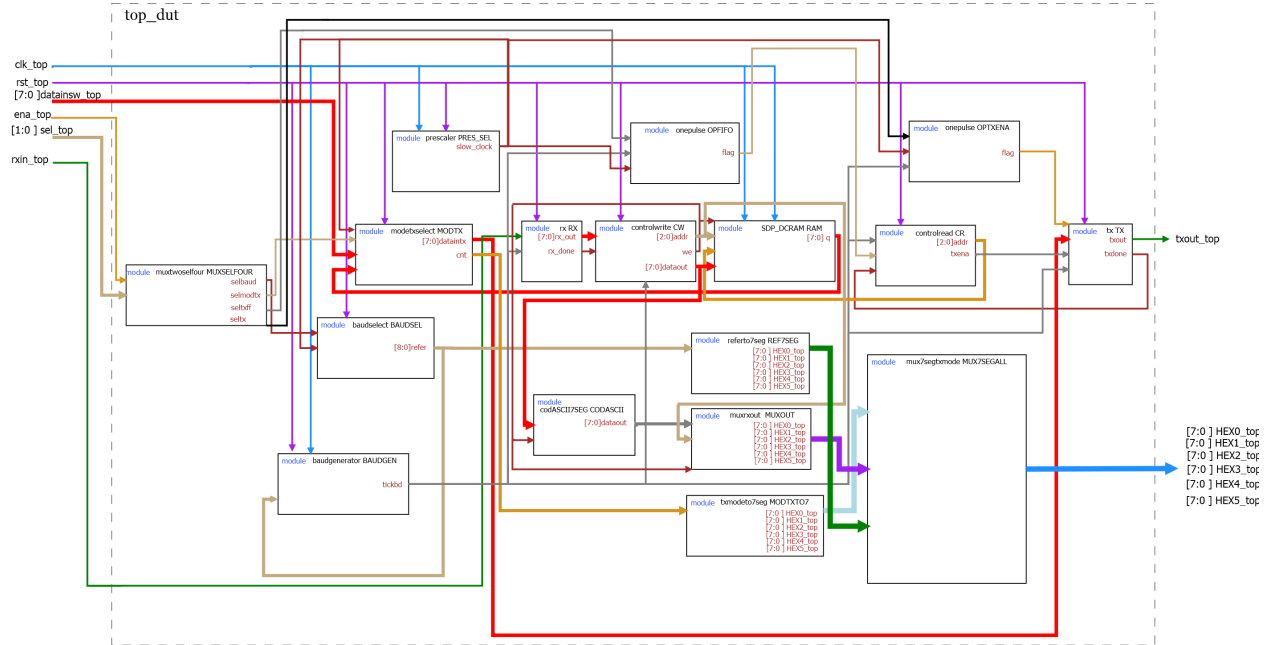


Figure 1: *Diagrama de bloques que representa la construcción del módulo top con sus respectivas entradas, salidas e interconexiones*

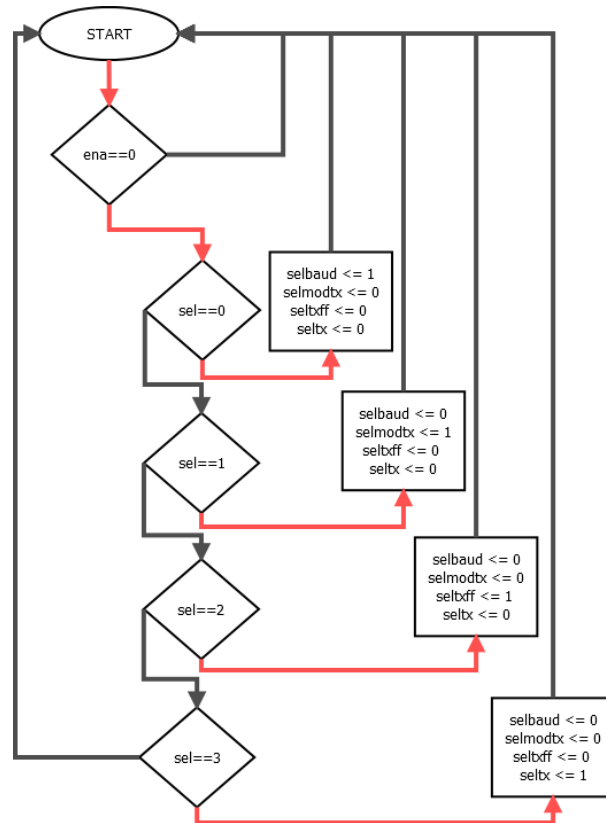


Figure 2: *Diagrama de flujo (mux2sel4) para la creación del módulo multiplexor que permite realizar diferentes acciones del diseño utilizando dos switches y un botón de habilitación de la acción*

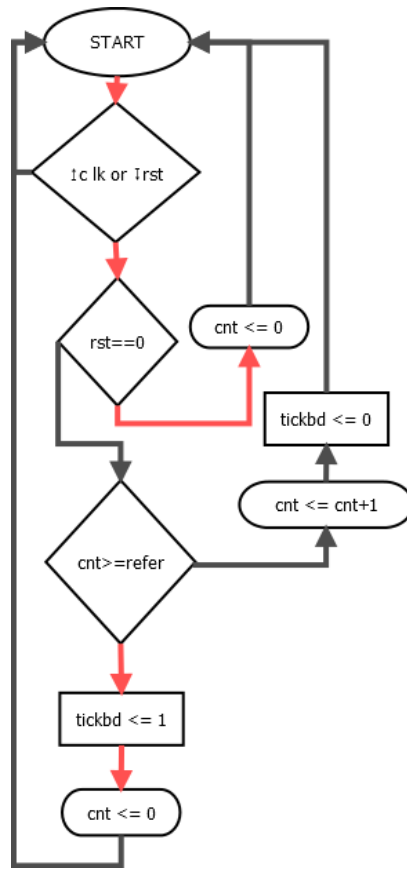


Figure 3: *Diagrama de flujo (baudgenerator) del módulo generador del baudrate el cual recibirá una referencia como entrada y en base a la misma tendrá de salida los diferentes baudages (9600, 57600, 115200)*

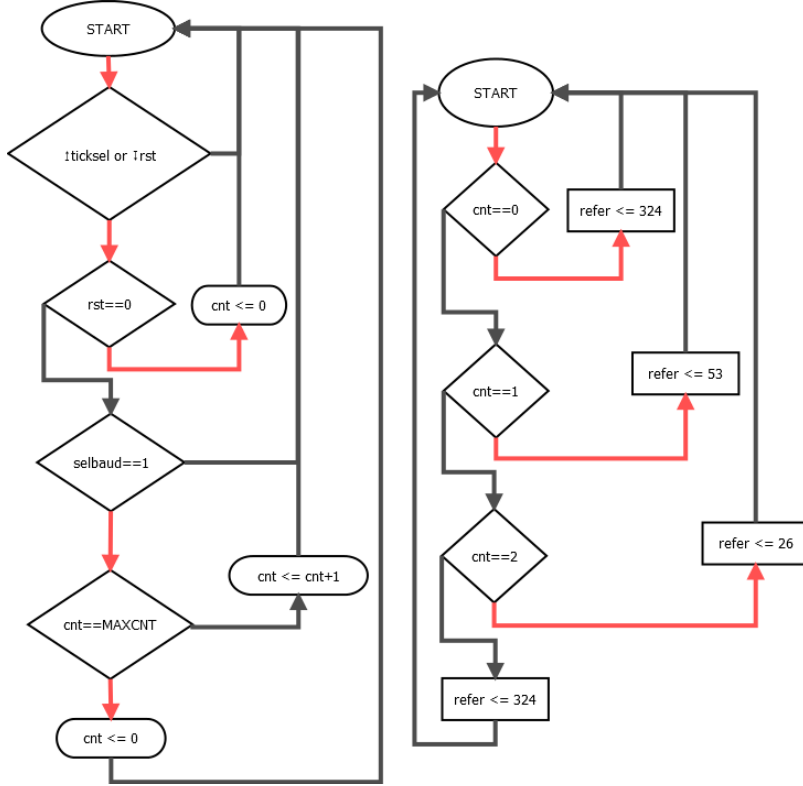


Figure 4: Diagrama de flujo (baudselect) del módulo seleccionador de baudage, el cual sacara una referencia la cual dependera del valor actual de la variable contadora (cnt)

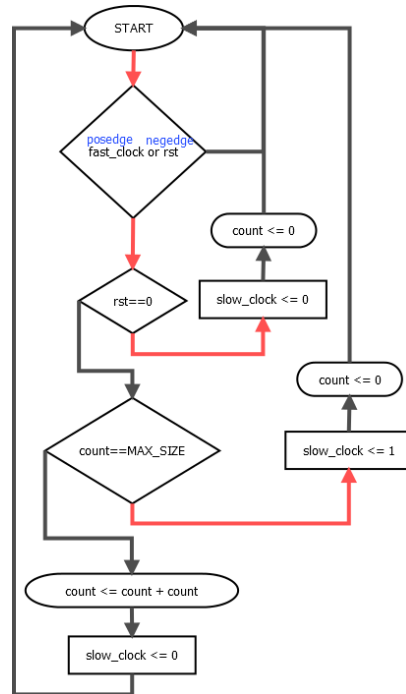


Figure 5: Diagrama de flujo (prescaler) del módulo prescaler que se encarga de dar como salida un tick con una frecuencia menor a la del reloj interno del FPGA, parametrizable por lo cual al variar MAX_COUNT podremos tener diferentes frecuencias

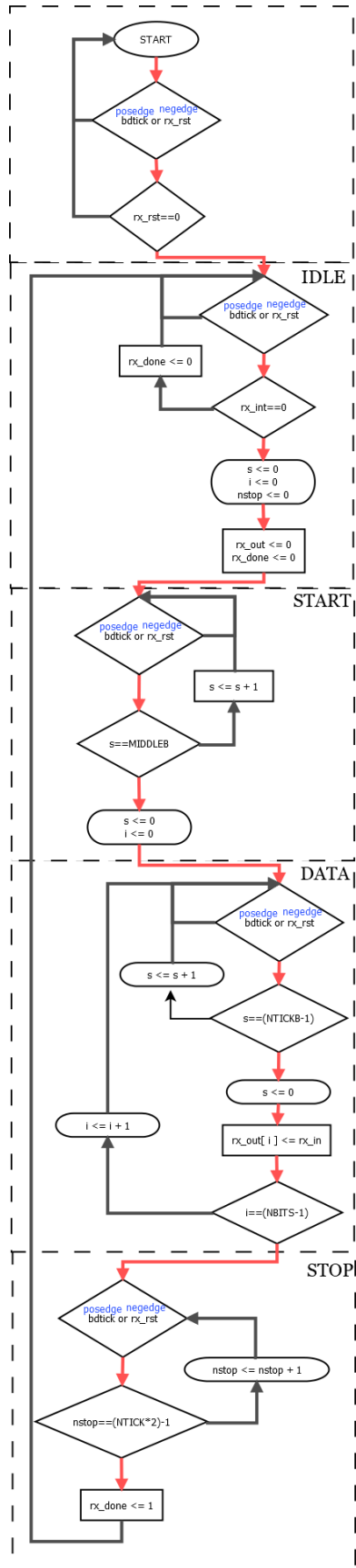


Figure 6: Diagrama de flujo (rx) de la máquina de estados del receptor (UART) el cual recibira los datos provenientes de la computadora, el módulo es parametrizable por lo que se podra modificar el número de bits de datos de la trama asi como el número de bits del stop

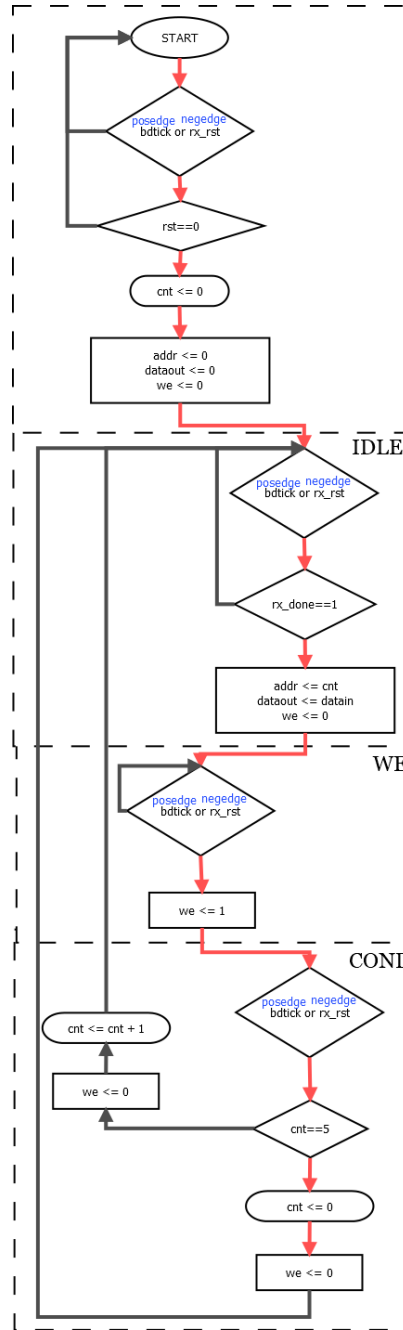


Figure 7: *Diagrama de flujo (controlwrite) de la máquina de estados del control de escritura el cual permitira guardar los datos en orden en la memoria uno detras de otro*

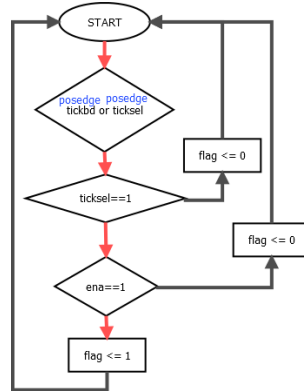


Figure 8: *Diagrama de flujo (onepulse) del módulo utilizado para poder evitar varios flancos de alto al utilizar un botón (medio mecanico) como entrada*

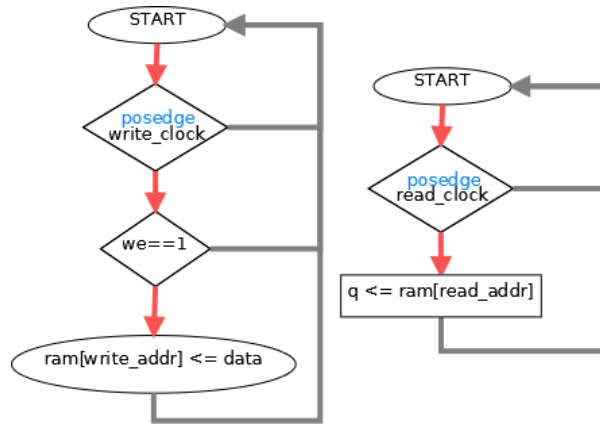


Figure 9: *Diagrama de flujo (SDP_DCRAM) del módulo de memoria RAM donde se guardarán los datos recibidos de la computadora*

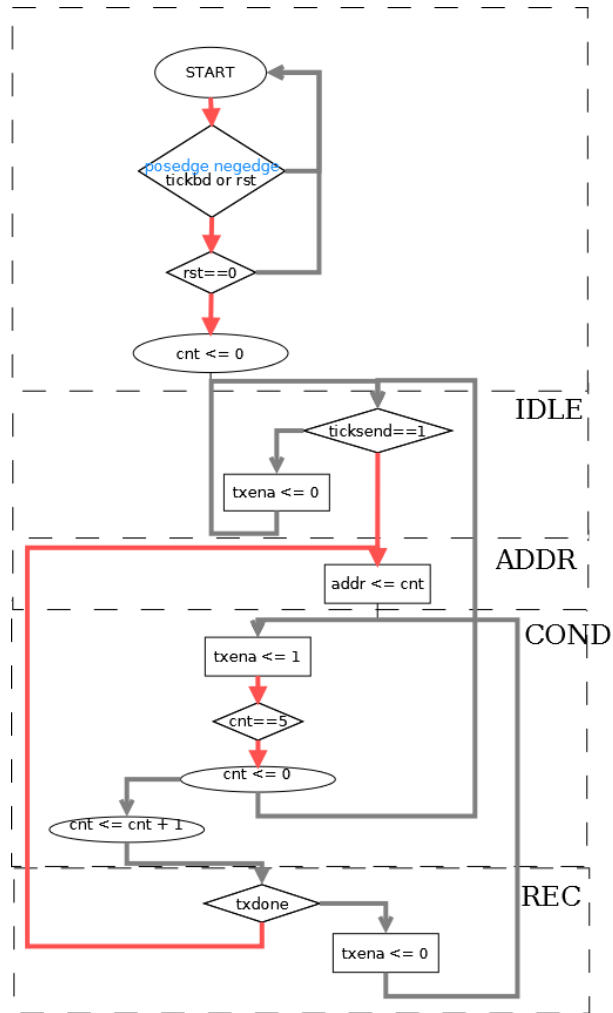


Figure 10: Diagrama de flujo (controlread) que representa la máquina de estados para el módulo que controlara la lectura de la memoria para poder sacar todos los datos guardados al oprimir un botón

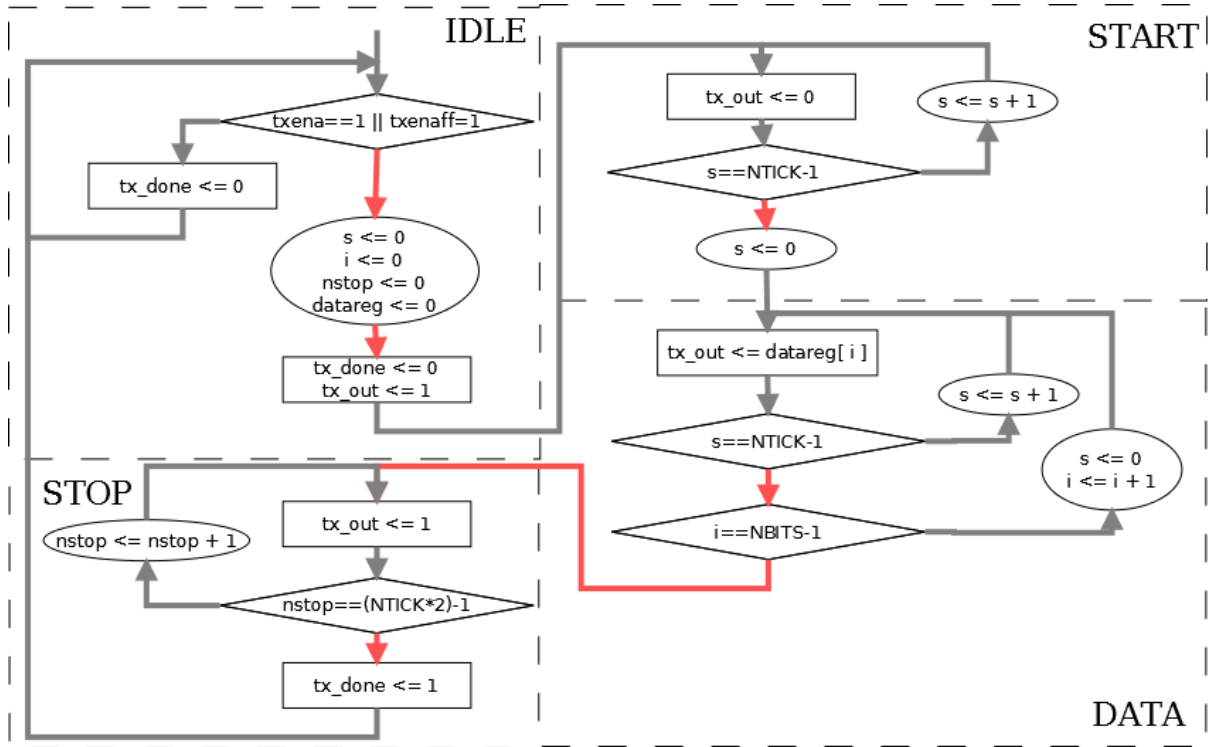
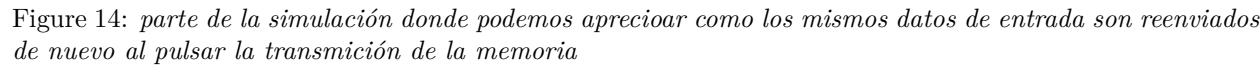
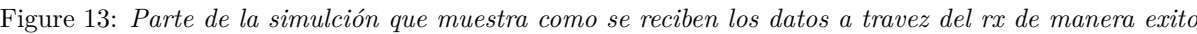


Figure 11: Diagrama de flujo (tx) que representa la máquina de estados del transmisor del protocolo UART, el cual utiliza como reloj el baudage seleccionado

modulo para poder transmitir tiene dos forma, recibiendo un estimulo del controlador de lectura o al cambiar el tipo de transmisión y utilizar los switches.

Table 11: Entradas y salidas del modulo tx

Señal	Dirección	Tamaño	Descripción
<i>bdtick</i>	in	1	Entrada de la señal de baudrate
<i>data_in</i>	in	8	señal de los datos a transmitir
<i>tx_ena</i>	in	1	Señal de habilitación mecanica para los switches
<i>tx_rst</i>	in	1	Señal para inicializar los registros
<i>txff_ena</i>	in1	1	Señal para habilitación de transmisión de memoria
<i>tx_done</i>	out	1	señal de que la trama se envio con exito
<i>tx_out</i>	out	1	salida que se conecta con la computadora



5. Resultados experimentales.

Como podremos ver en las imágenes siguientes la correcta implementación del módulo, tanto para configurar sus características figuras 15 a la 19.

En la figura 20 a la 21 podemos ver como la transmisión se lleva a cabo de manera correcta. transmitiendo un 0 utilizando el modo de transmisión de los switches.

En la figuras 22 y 23 podemos ver como al recibir 6 bytes de información o 6 caracteres estos se muestran en los displays y pueden ser retransmitidos de vuelta a la computadora mostrando un correcto funcionamiento.

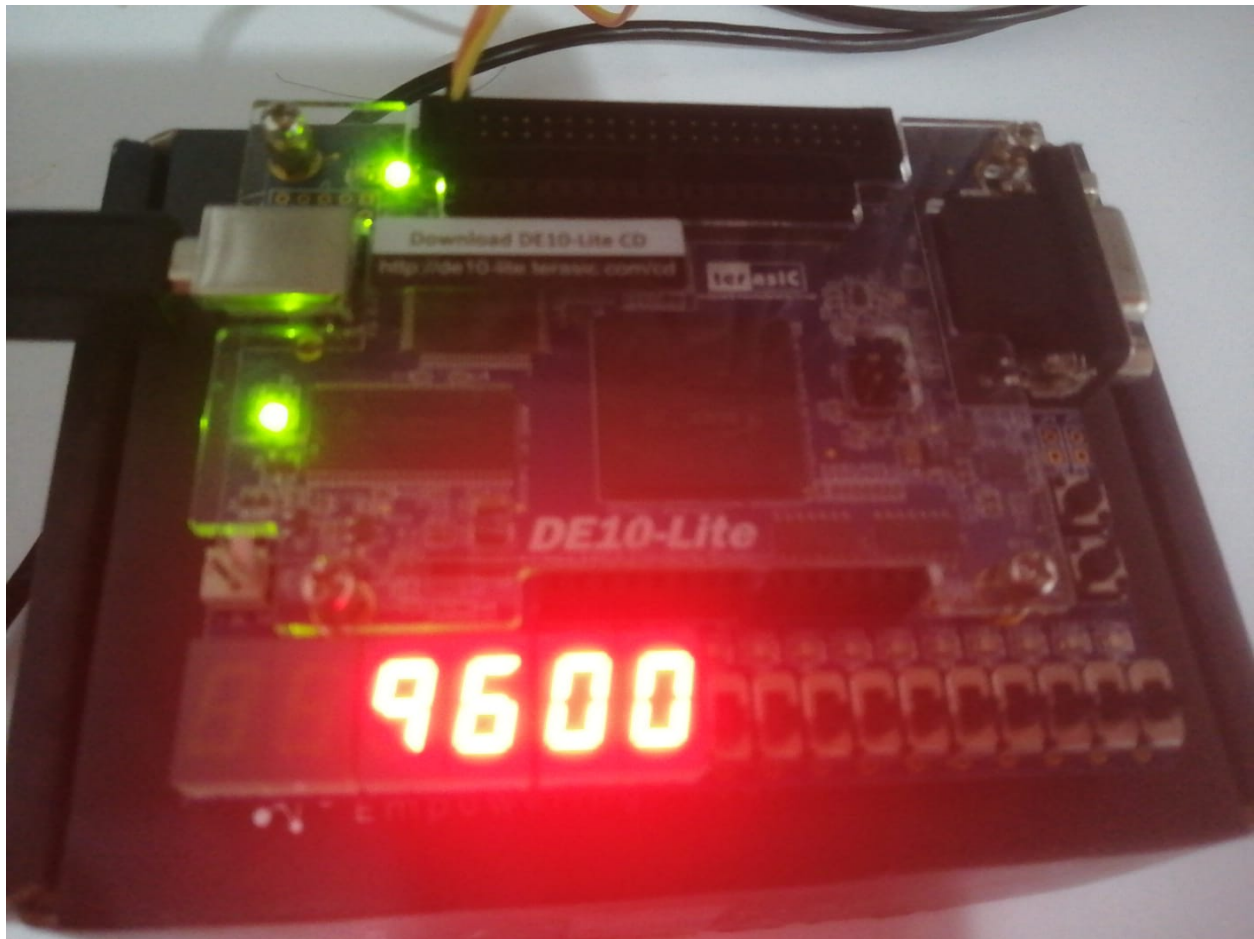


Figure 15: *Configuración de baudrate en 9600*

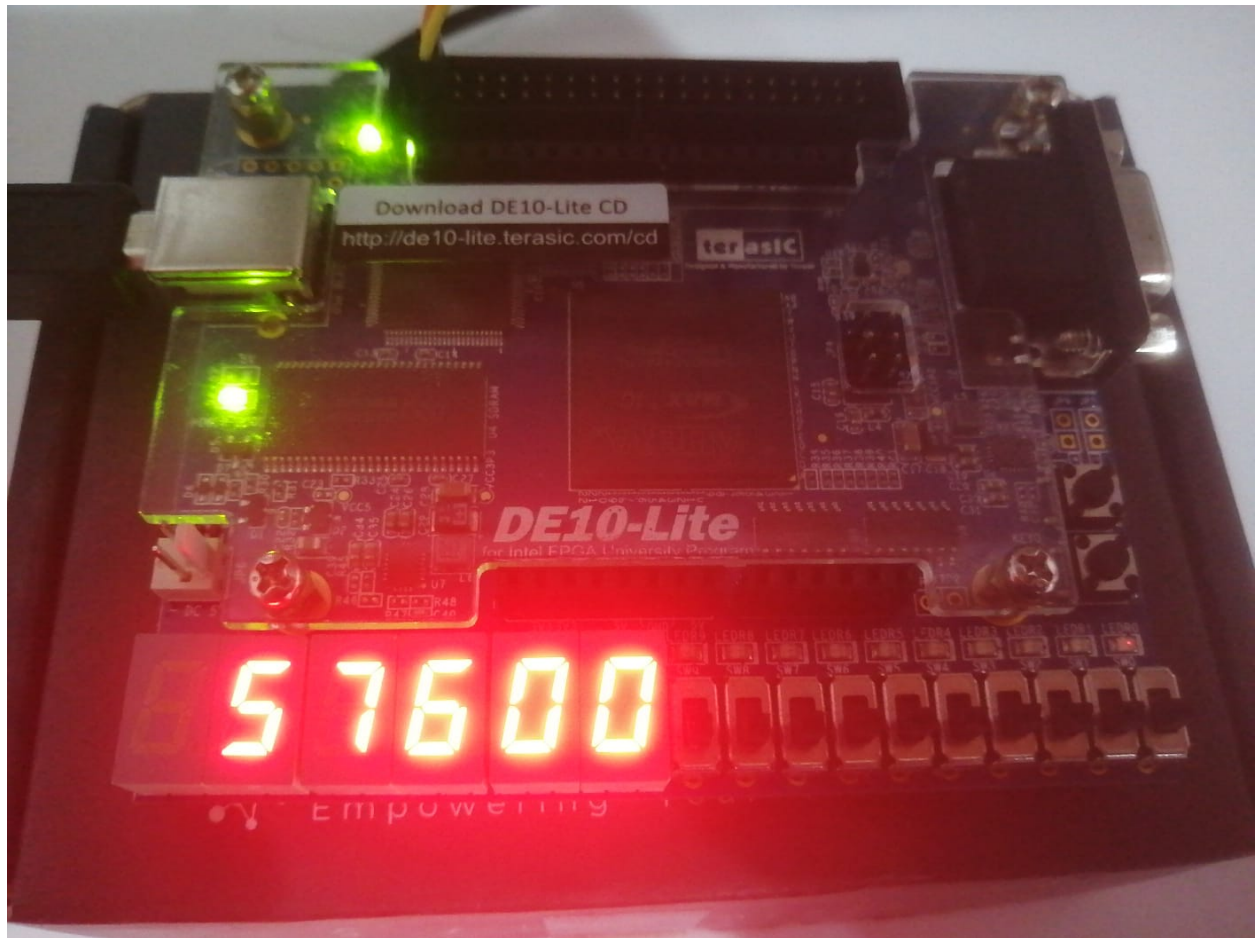


Figure 16: *Configuración de baudrate en 57600*

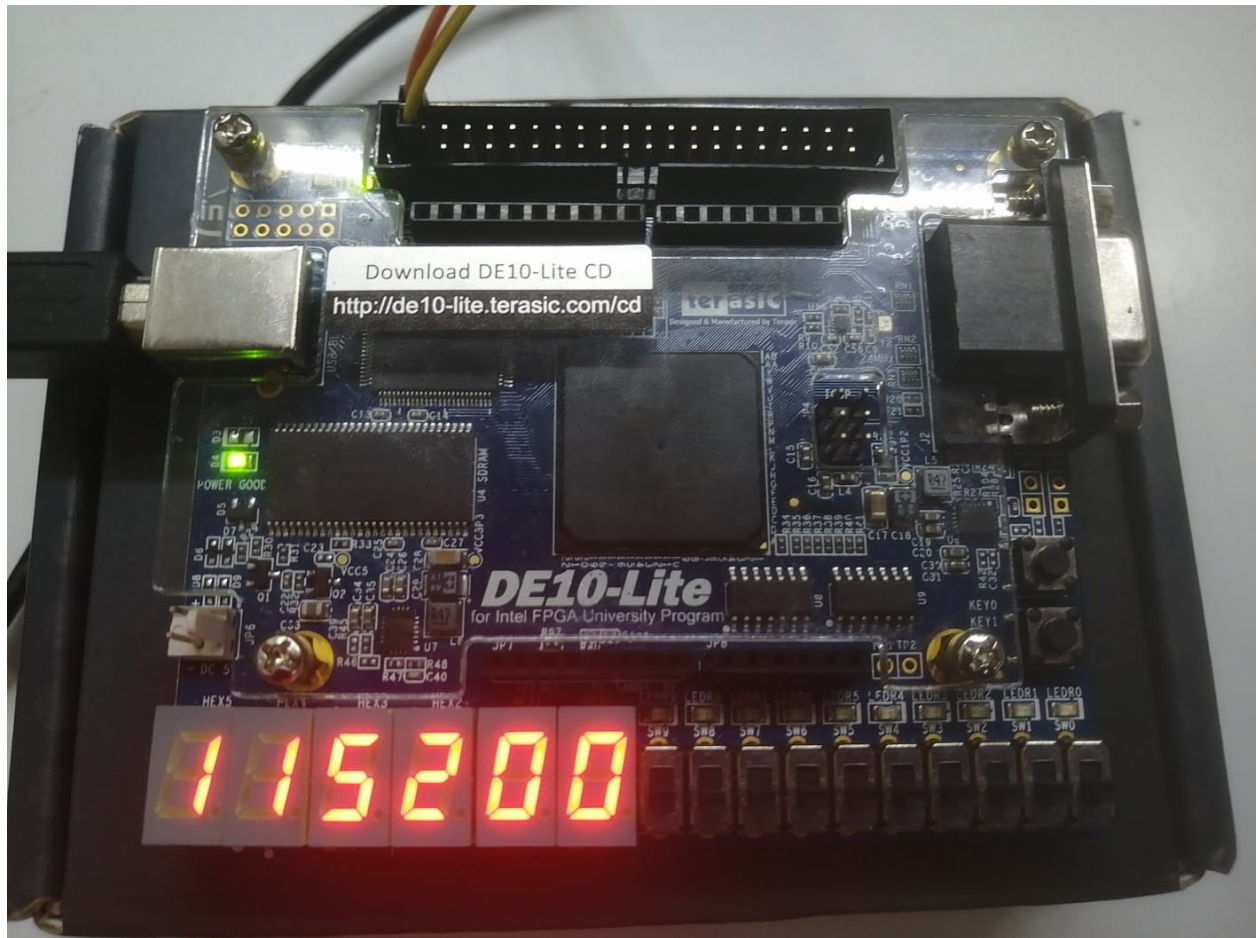


Figure 17: Configuración de baudrate en 115200



Figure 18: Configuración del modo de transmisión utilizando los datos guardados en la memoria para reenviar por el módulo transmisor

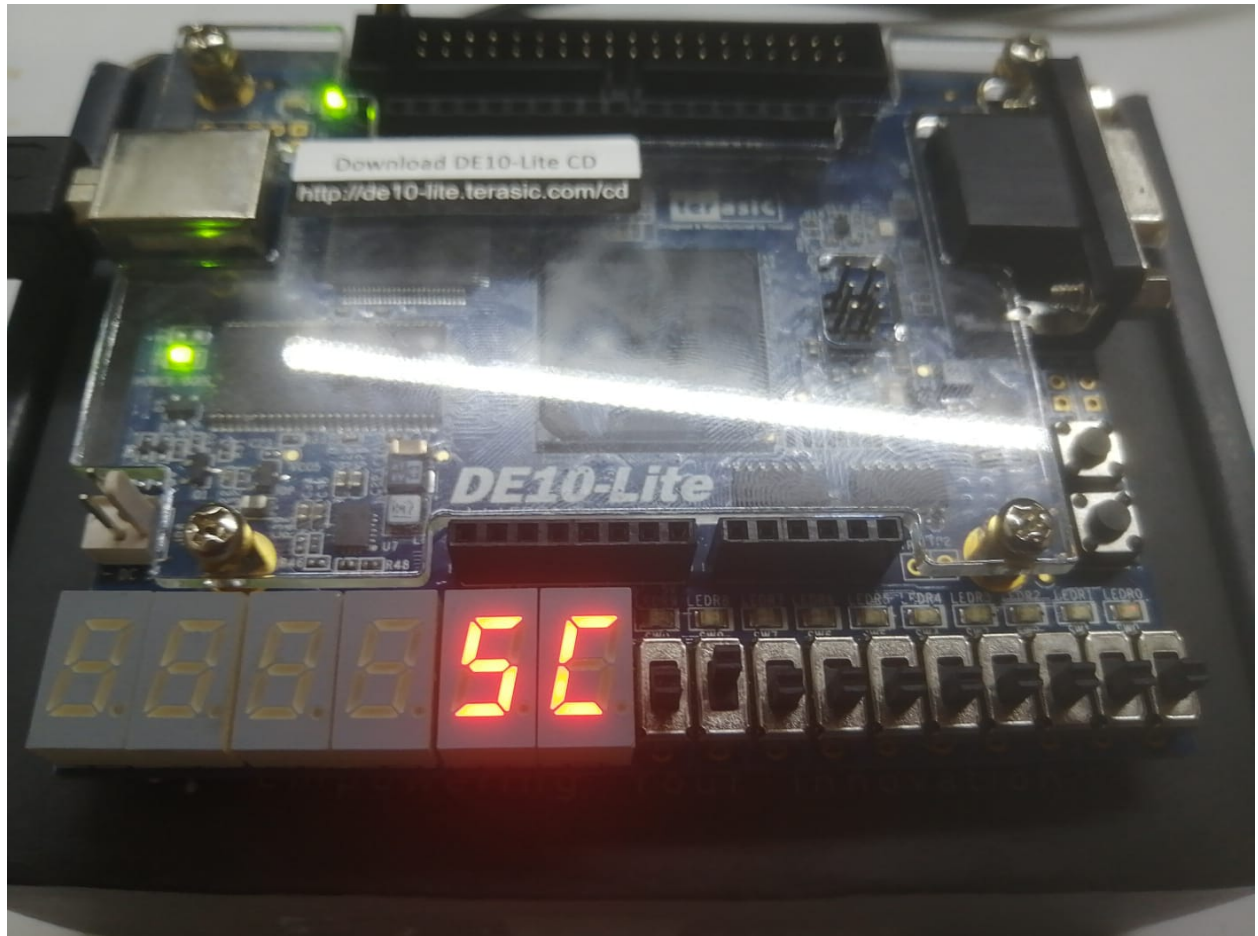


Figure 19: Configuración del modo de transmisión utilizando los datos de los 8 switches de entrada para reenviar por el módulo transmisor

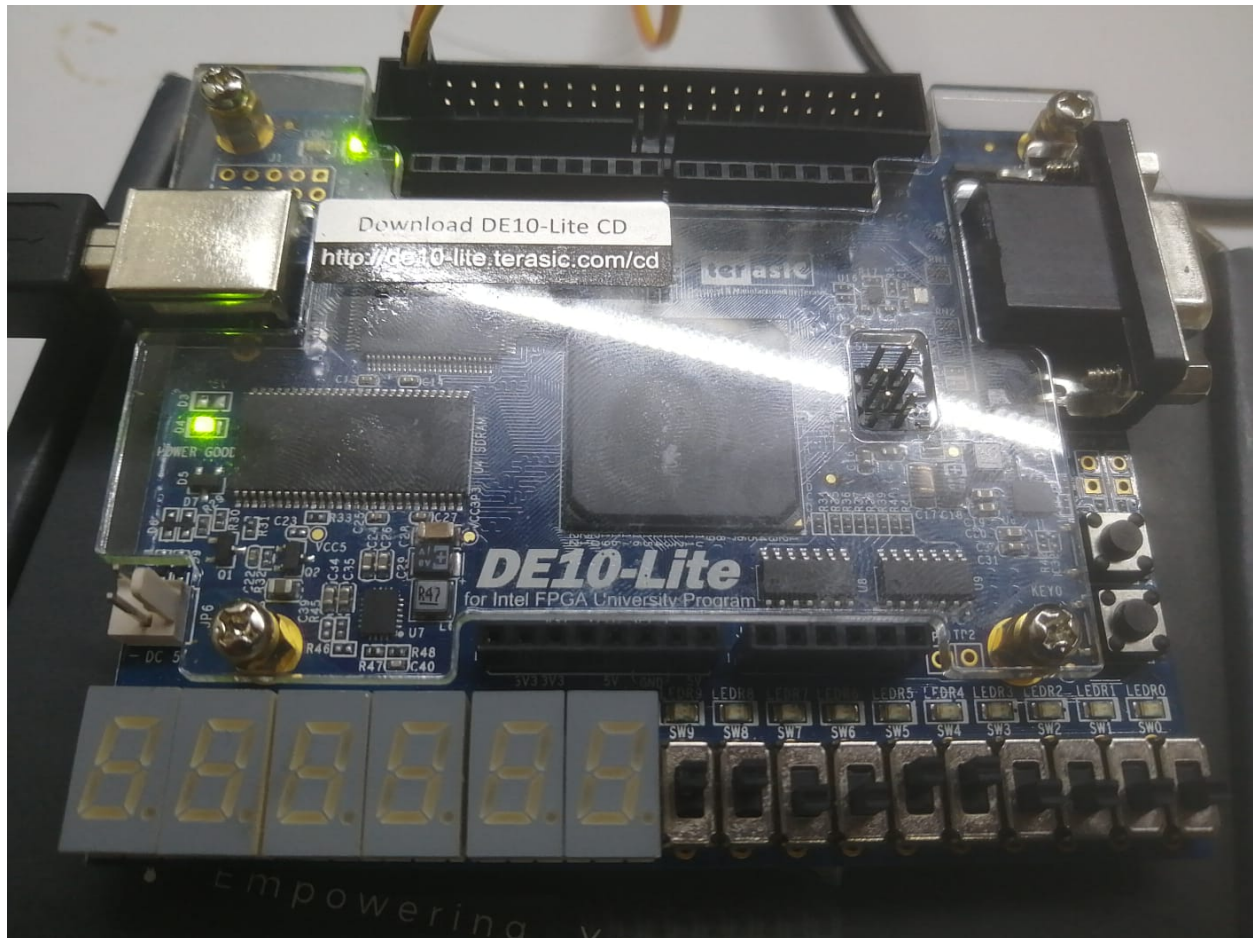


Figure 20: configuración en código ascii para poder renviar un cero

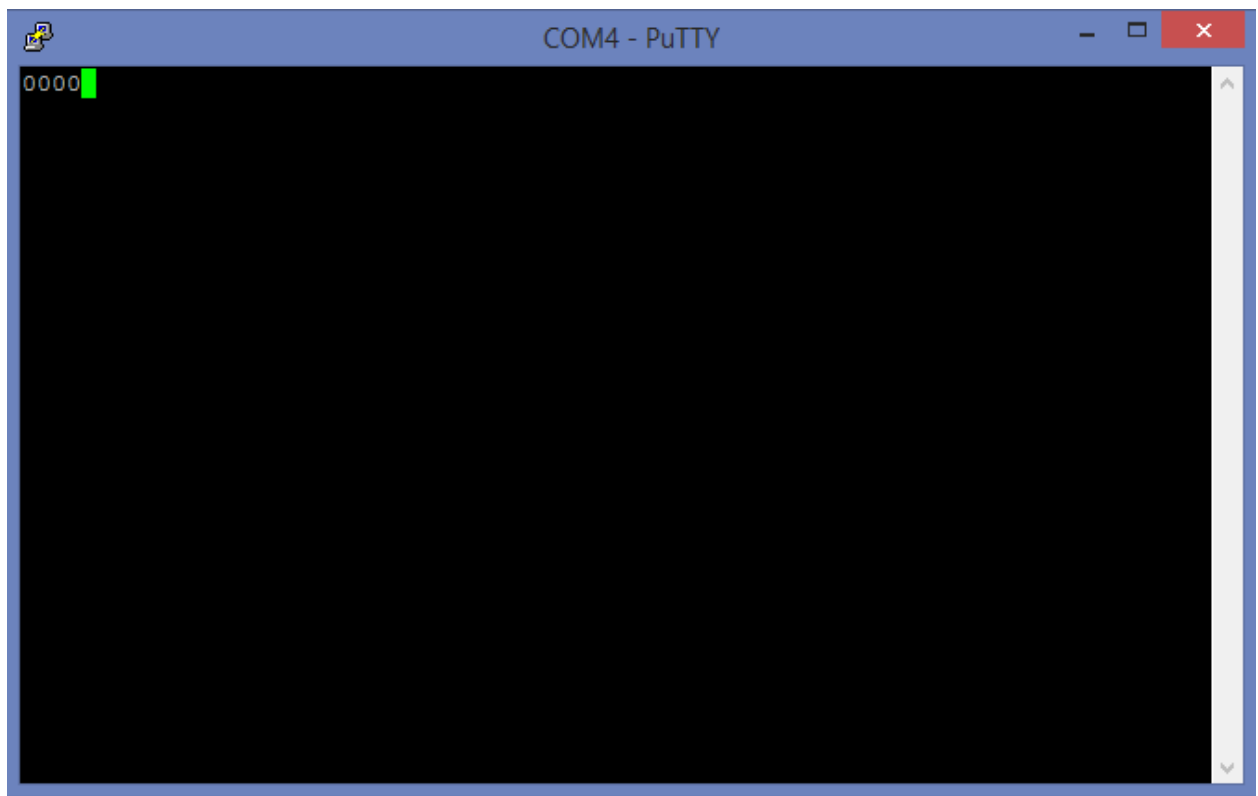


Figure 21: *terminar de comunicación donde recibimos el caracter mandado*

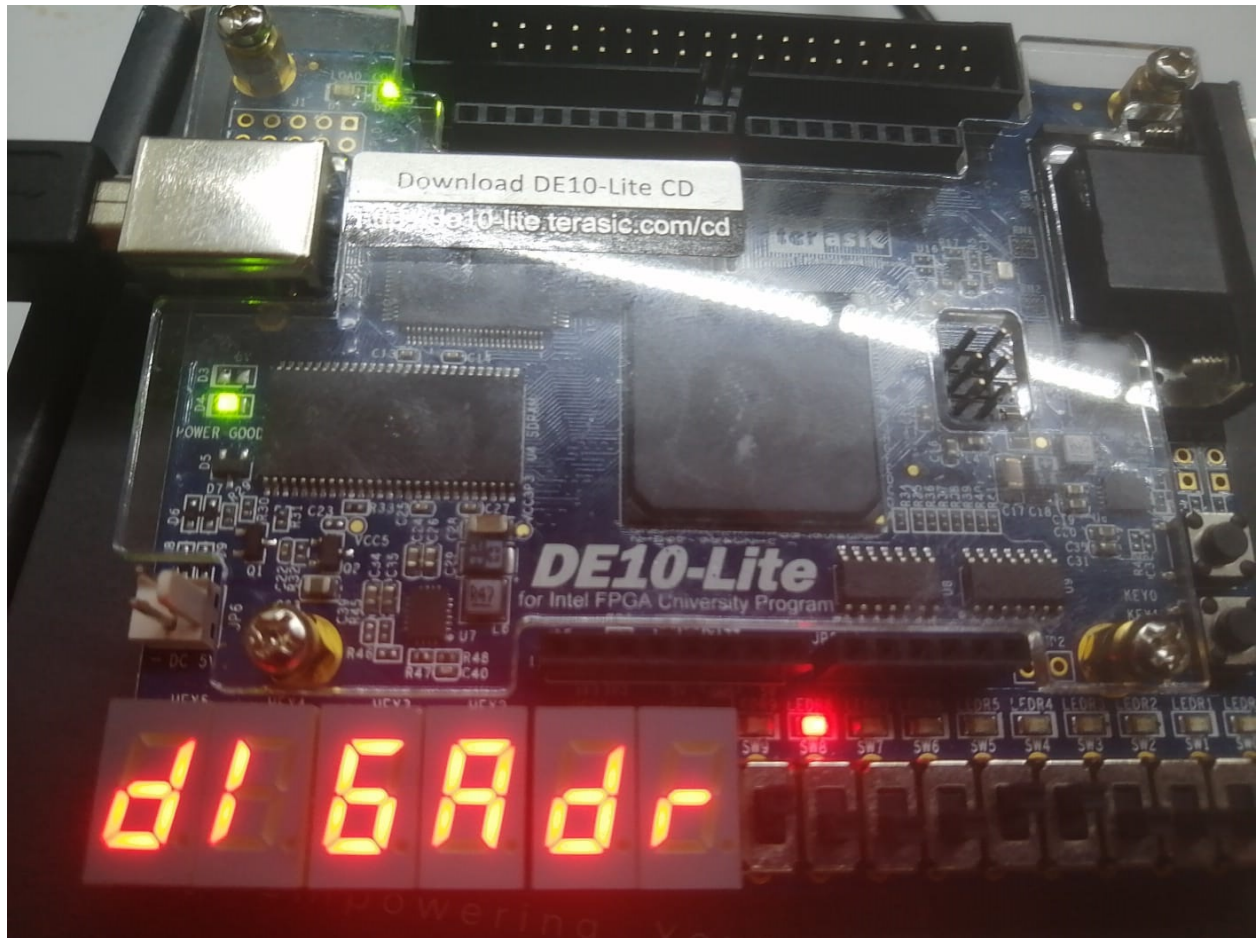


Figure 22: *recibiendo datos desde la terminar del monitor serial y guardada en la memoria*



Figure 23: *despues de pulsar el boton de renvio de los datos de la memoria*