IMPLEMENTACION DE RS-232

SERGIO ALEXANDER FLOREZ GALEANO

CRISTIAN STEVEN CHAVARRO RICO

PRESENTADO A:

RAMIRO ANDRES BARRIOS VALENCIA



UNIVERSIDAD TECNOLOGICA DE PEREIRA

INGENIERIA DE SISTEMAS Y COMPUTACION

LABORATORIO ELECTRONICA II

PEREIRA, MAYO DE 2012

**INFORME DE RS232**

**Introducción**

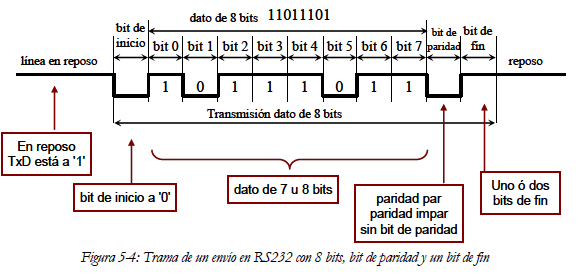
En la comunicación siempre debe existir alguien que transmite y alguien que recibe, llamados respectivamente transmisor y receptor. En la transferencia de datos entre computadores se puede hacer síncronamente o síncronamente, para cumplir el objetivo de esta práctica se implementara la comunicación asíncrona que permite realizar la transferencia de información sin la necesidad de un ciclo continuo de reloj.

**Marco Teórico**

**RS232**

En esta práctica se diseñará un transmisor/receptor serie asíncrono (Universal Asynchronous Receiver-Transmitter) que siga la norma RS232. Este módulo nos permitirá comunicar nuestra placa con el puerto serie del PC. Aunque en la norma se definen más señales, sólo son imprescindibles tres señales: la línea de transmisión de datos (TxD), la de recepción (RxD), y la línea de masa (GND).

Existen distintas velocidades de transmisión, que se definen en bits por segundo (bps) o baudios (921600, 460800, 230400, 115200, 57600, 38400, 19200, 9600, 4800, …). También se puede variar el número de bits del dato que se envía, así como el envío de un bit de paridad y el número de bits de fin.

La línea serie permanece a nivel alto ('1') mientras no se envían datos. Cuando el transmisor va a empezar la transmisión, lo hace enviando un bit de inicio que está a '0'. Posteriormente se envían consecutivamente los bits del dato empezando por el menos significativo. Después del último bit de dato se envía el bit de paridad en caso de que se haya especificado. Por último, se cierra la trama con uno o dos bits de fin con valor '1'. En la figura 5-4 se muestra el cronograma de un envío RS232 con 8 datos, un bit de paridad (par) y un bit de fin.

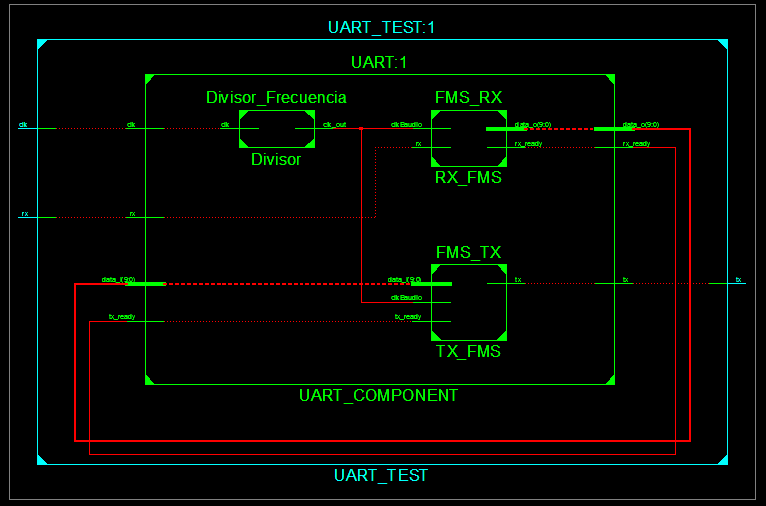
**Procedimiento**

El dato de entrada será de 8 bits, con un total de 10 bits con bit de inicio y bit de parada. No se tendrá en cuenta un bit de paridad.

Lo primero que se tuvo en cuenta fue como se iba a implementar los dos métodos de transferencia de RX y TX, para eso se crearon dos maquinas de estado. En la TX cada vez que ocurre un cambio en tx\_ready a 1 el empieza a pasar el dato\_i de entrada bit por bit. Cuando lo transfiere tendrá siempre salida 1 el cual lo mantiene a la UART en estado IDLE.

En la RX su trabajo es invertido; el empieza a recibir el dato rx, durante este tiempo en que no se envía nada se encuentra en estado IDLE; cuando se empieza a enviar empieza a recibir cada bit del dato y lo empieza a unir. Al final cuando halla unido todo el dato en uno solo envía una señal de ’1’ en el rx\_ready y empieza a recibir el dato.

Después se creo el divisor de frecuencia el cual va a transmitir a 434 baudios.



**Dificultades**

Durante la realización se encontraron algunos problemas al implementar los Maquinas de estados, como la transferencia del dato y la recepción, con el divisor de frecuencia es la frecuencia que se va a usar para a transferencia.

**Conclusiones**

Por medio de la implementación una UART podemos crear un puente de transferencia entre dos terminales y así intercambiar información entre estos.

La transferencia de forma serial es una forma muy útil y eficiente para transmitir datos por sus bajo porcentaje de error en la transferencia.

**Bibliografía**

cselectrobomba.googlecode.com/files/Serial\_RS232.pdf

sopa.dis.ulpgc.es/ii-dso/lecminix/manejado/rs232/rs232.pdf

http://www.doranscales.com/connectivity-bulletins/all-about-RS232.pdf