

RECEPTOR BLUETOOTH CON HDL (VHDL Y VERILOG)

Receptor Bluetooth con HDL (VHDL Y VERILOG)

*Dr. Juan Antonio Jaramillo Gómez
jantonioj@yahoo.com, jjaramillo@ipn.mx*

*M. en C., Mirna Salmerón Guzmán
yupisg2@yahoo.com.mx*

*Dr. Brahim El Filali
braelf@hotmail.com*

*UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA
Y TECNOLOGÍAS AVANZADAS
INSTITUTO POLITÉCNICO NACIONAL*

Resumen

Este artículo presenta la forma de enviar datos de una App comercial de un teléfono celular a un receptor BlueTooth conectado a una tarjeta de desarrollo con un FPGA. Los datos transmitidos de la aplicación están codificados en ASCII, y pueden estar fijos o bien ser programados por el usuario, si la aplicación lo permite. Los lenguajes descriptivos de hardware (HDL) en que se hicieron los programas son VHDL y Verilog, utilizado el programa ISE de Xilinx y probados en la tarjeta Nexys 2.

Abstract

This article presents the way to send data from a commercial App of a cell phone to a BlueTooth receiver connected to a development board with an FPGA. The data transmitted from the application is encoded in ASCII, and can be fixed or programmed by the user, if the application allows it. The hardware description languages (HDL) in which the programs were made are VHDL and Verilog, used the ISE software of Xilinx and tested on the Nexys 2 board.

Introducción

La comunicación con distintos dispositivos para la medición o control de un sistema a través de elementos de comunicación inalámbricos (RF, IR, US, BlueTooth, NFC, etc.) permiten implementar aplicaciones de telefonía, reproducción de audio, telemando o telecontrol y telemetría.

BlueTooth

BlueTooth o BT es una especificación tecnológica para transmisión de voz y datos entre distintos dispositivos a una frecuencia de 2.4GHz. Existen equipos o dispositivos electrónicos

como celulares, tabletas, computadoras, vehículos, etc., que tienen incluida esta tecnología, pero cuando no se tiene disponible, se puede agregar un módulo comercial (ver figura 1) a un sistema, proyecto o una simple práctica de laboratorio.

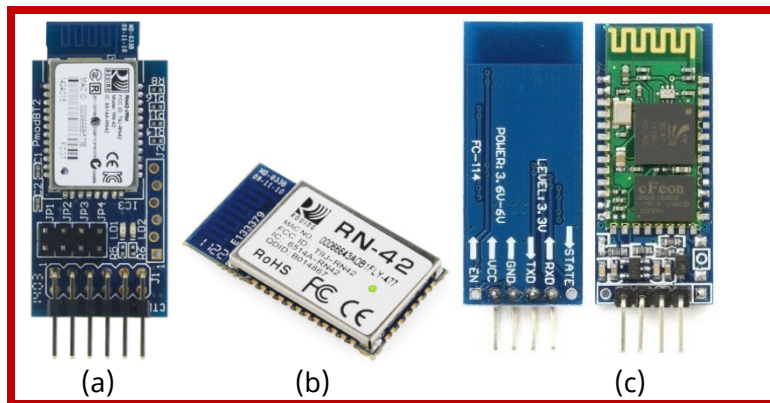


Figura 1. Módulos de BT comerciales: (a) PmodBT2 de digilent, (b) RN-42, (c) HC-06.

Apps

Existen dos posibilidades de tener una App en el teléfono:

- aplicaciones comerciales disponibles para Android y para iOS, tanto las gratuitas como las de paga
- aplicaciones hechas a la medida en un programa como el App inventor.

Dentro de las aplicaciones comerciales para BT algunas están dedicadas a dispositivos reproductores de música, otros a dispositivos para la comunicación con un auricular o con un vehículo; las de interés para éste artículo son las de controladores. Algunas aplicaciones de Android de este último tipo son las listadas a continuación y las mostradas en la figura 2:

- BlueTooth Serial Controller
- BT Controller
- Ultimate Bluetooth Control
- Bluetooth Device Controller
- Bluetooth Controler 8 Lamp
- Control Bluetooth Programable
- Joystick Arduino Bluetooth

- Bluetooth terminal, entre muchas otras más

En este artículo, se utilizaron las dos primeras; se configuran para mandar las letras, números y símbolos, tanto solos como combinados. Cuando se envían solos se reciben en formato ASCII.

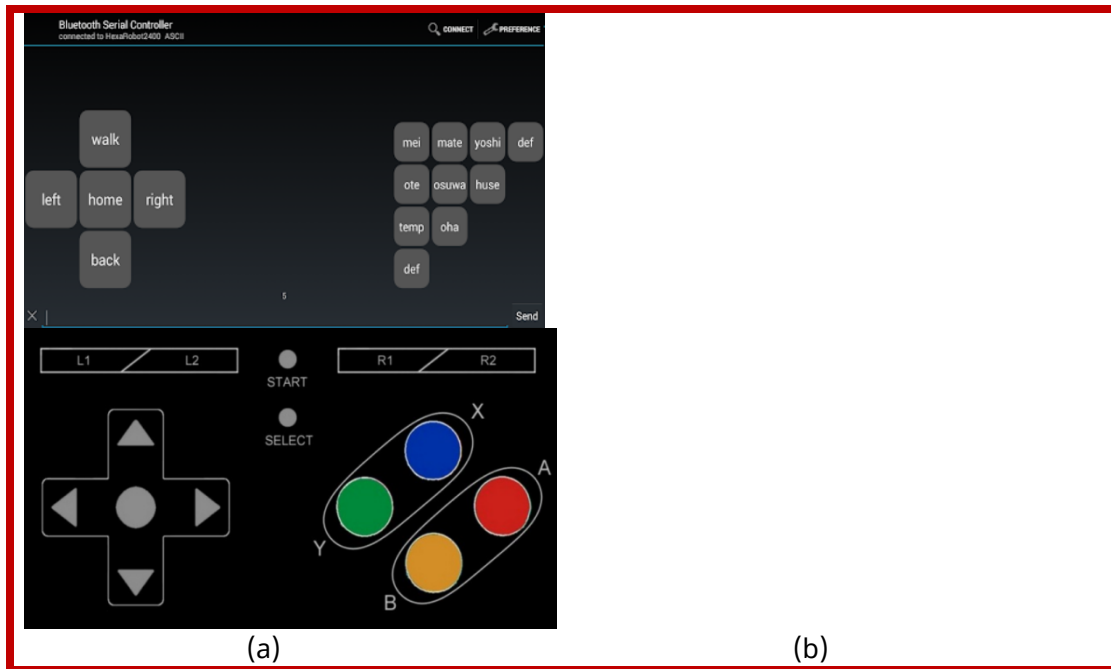


Figura 2. Apps de Android: (a) Bluetooth serial controller, (b) BT controller.

UART (Universal Asynchronous Receiver Transmitter)

UART (*Universal Asynchronous Receiver-Transmitter*) significa Transmisor-Receptor Asíncrono Universal, es un hardware dedicado a controlar los puertos o dispositivos serie y a controlar el medio (alámbrico-inalámbrico), el flujo o velocidad (2400, 9600, ... baudios) y la forma del envío (protocolo de comunicación), como se muestra en seguida:

- Alámbricos: RS-232, RS-422 y RS-485
- Inalámbricos: fibra óptica, infrarrojo y Bluetooth
- Modulados (con o sin cables): módems, radio frecuencia (RF)

En el protocolo RS232 alámbrico se trabaja con un conector DB9 (figura 3(a)), en donde los niveles de voltaje están entre +3 a +15V para un '0' y -3 a -15V para un '1', lo cual se pasa a TTL con un MAX232. Su esquema de comunicación se muestra en la figura 3(b). En el inalámbrico se utilizan los módulos de RF comerciales o un módulo BT.

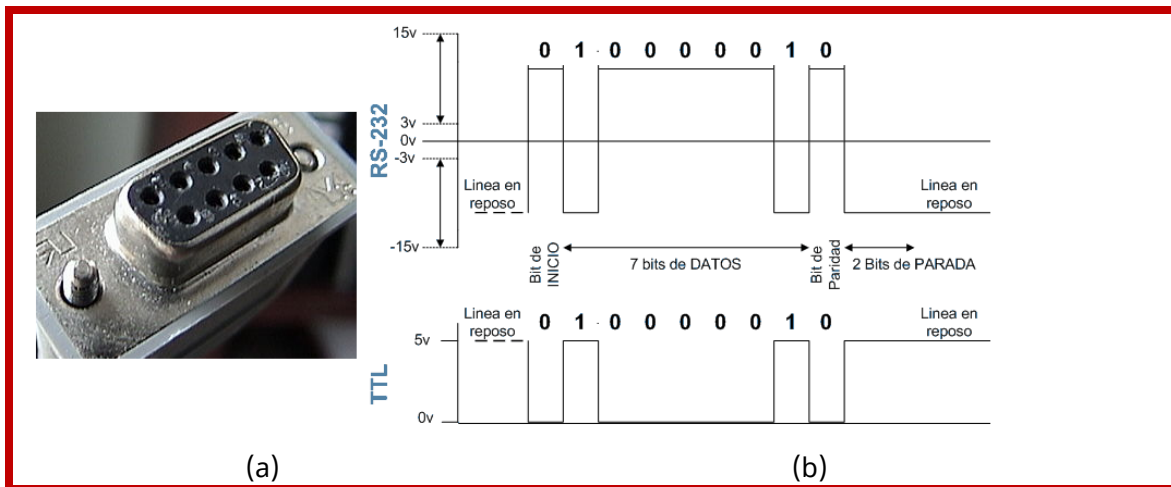


Figura 3. (a) conector DB9, (b) diagrama de tiempos para la comunicación.

La velocidad está estandarizada con la norma RS 232C en baudios (bits por segundo) como se ve a continuación, aunque también hay velocidades no estandarizadas.

- 75
- 1200
- 110
- 2400
- 150
- 4800
- 300
- 9600
- 600
- 19200

Desarrollo

UART & RS232

Para la implementación de un circuito UART que se comunique con el módulo BT HC-06 usando el protocolo RS232, que se inicie con bit de *start* y termine con bit de *stop*, se establecen los siguientes parámetros: bus de datos de 8 bits, un bit de *stop* y sin bit de paridad, como se muestra en las figuras 4 y 5. Otra forma alternativa se muestra en la figura 6, es con bit de paridad y con uno o dos bits de *stop*.

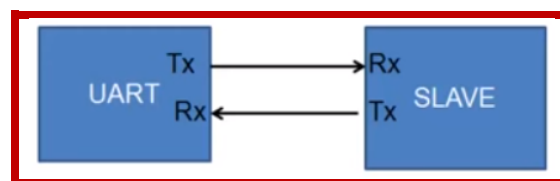


Figura 4. Circuito UART.

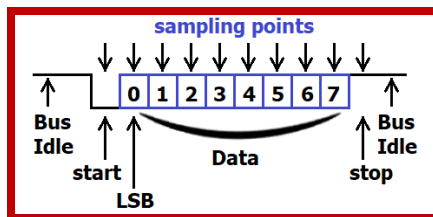


Figura 5. Parámetros para la comunicación con protocolo RS232.

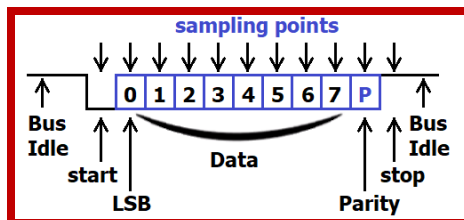


Figura 6. Protocolo RS232 con bit de paridad.

Posteriormente se elige la velocidad de transmisión, que para este caso es de 9600, y a partir de esta se obtiene el periodo de la señal ($T=1/9600=104.16\mu s$, por bit transmitido), con el que se diseña un contador para alcanzar la mitad del periodo ($T/2=52.08\mu s$) a partir del reloj de la tarjeta Nexys 2 que trabaja a 50MHz ($T_{nexys} = 1/50MHz = 20ns$). Realizando el cálculo de $52.08\mu s/20ns$ da un resultado de 2604.16 veces que cabe $T/2$ en el periodo de la tarjeta T_{nexys} , por lo que se necesita un contador de 2604 pulsos para establecer el alto y 2604 veces para el pulso bajo. Si se deseara cambiar de velocidad, la tabla 1 muestra los cálculos a otras velocidades de transmisión y el tamaño del contador.

Tabla 1. Datos importantes obtenidos de la velocidad en baudios.

Baudios [bits/seg]	Periodo T[ms]	T/2 [ms]	Contador @T/2 @50MHz	Baudios [bits/seg]	Periodo T[us]	T/2 [us]	Contador @T/2 @50MHz
75	13.33	6.67	333,333.3	1200	833.3	416.6	20,833.33
110	9.09	4.55	227,272.7	2400	416.6	208.3	10,416.66
150	6.67	3.33	166,666.6	4800	208.3	104.16	5,208.33
300	3.33	1.67	83,333.3	9600	104.16	52.08	2,604.16
600	1.67	0.833	41,666.6	19200	52.08	26.04	1,302.08

FSM del receptor BT con el módulo HC-06

Para la recepción de datos del módulo BT se implementa una máquina de estados finitos tipo Mealy de solo dos estados, mostrada en la figura 7, la cual funciona con las señales **Rx** (terminal D0 del HC-06) que detecta el primer nivel bajo para dar inicio a la recepción, y la señal **done** que indica cuando terminó de recibir los 8 bits del dato de Rx.

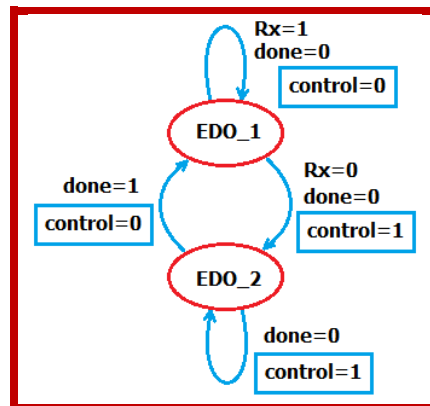


Figura 7. Máquina Mealy de dos estados.

El primer estado EDO_1 espera a que el receptor tenga el primer bit con cero que es el bit de *start* del protocolo RS232 para pasar al segundo estado EDO_2. En el EDO_2 se realizan varios procesos de la recepción, el primero y más importante es la generación de la señal de captura (*capture*) a partir del contador "c2" que tiene 3 veces la velocidad de transmisión. Con la señal de captura se guardan los datos de Rx con un registro de corrimiento, se activa un contador i que cuenta de 0 a 8 (el bit de *start* y los 8 bits de datos), y por último la actualización de la máquina de estados, como se muestra en la figura 8.

A partir de esta información se presentan los códigos de VHDL, Verilog y el archivo de restricciones de usuario UCF en archivo anexo [ReceptorBTCódigosHC06.pdf](#).

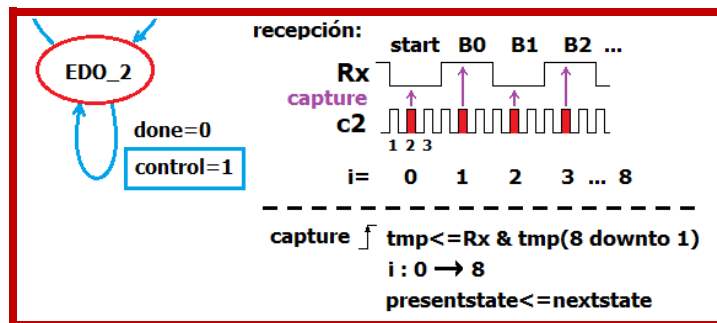


Figura 8. Actualización de la máquina de estados.

Para mostrar el funcionamiento de los códigos implementados, primero se realiza el siguiente paso:

Conexión o enlace con el módulo HC-06

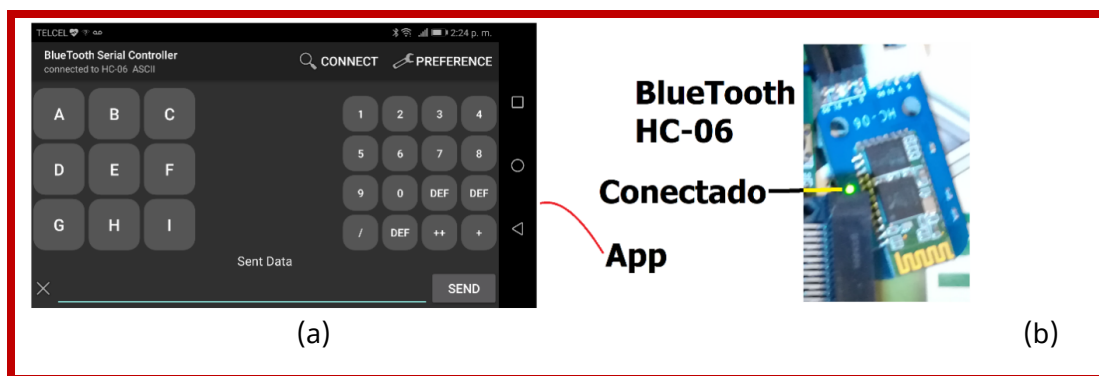
Para conectar el teléfono con el módulo BT se realizan los siguientes puntos:

- Activar BT en el teléfono y buscar dispositivos (tener energizado el HC-06)
- Seleccionar y conectar con el módulo (HC-06)
- Introducir clave si se solicita (i.e. 0000 o 1234)

Posteriormente después de haber instalado una aplicación en el celular

- Abrir la App y conectar al dispositivo (puede solicitar activar BT en caso necesario)
- Configurar las teclas para el envío de datos, enviar y probar la recepción de datos

En la figura 9 se muestran las fotos de las pruebas realizadas



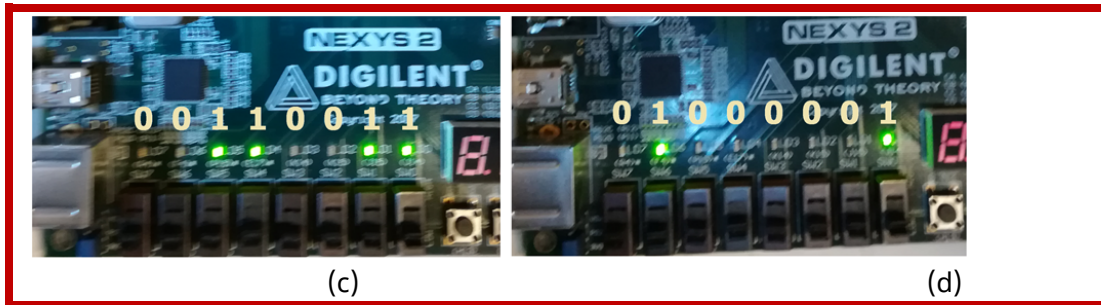


Figura 9. (a) App BlueTooth Serial Controller, (b) módulo BlueTooth HC-06, (c) Nexys2 recibiendo 00110011=33=3 en ASCII, (d) 01000001=41=A en ASCII.

Conclusiones

Se presentó información del manejo de una aplicación de celular para BlueTooth que envía datos ASCII a un receptor BT a través de la implementación de dos códigos en HDL y un archivo de restricciones de usuario (VHDL y Verilog y el UCF). Los datos fueron recibidos en el FPGA y visualizados en leds. El potencial de los códigos presentados está en las aplicaciones de telemando, telemetría o simplemente comunicación con otros dispositivos. Estos códigos permitirán a los alumnos continuar con la realización de proyectos con el dispositivo BT.

Referencias y Recursos electrónicos

Definición de BlueTooth, recuperado en mayo 2018, disponible en:
<https://definicion.de/bluetooth/>

Carlos A. Hernández Gutierrez, FPGA & Verilog. Receptor (Receiver) RS-232 HC05 bluetooth - Hackeando Tec., recuperado en mayo 2018, disponible en:
<https://www.youtube.com/watch?v=2d6jz-DrXiU>

Carlos A. Hernández Gutierrez, FPGA comunicación RS-232 con el modulo Bluetooth HC05 usando Verilog, recuperado en mayo 2018, disponible en:
<https://www.youtube.com/watch?v=tpajglbrd-A>

Protocolo serie RS232, recuperado en mayo 2018, disponible en:
http://docente.uco.mx/diegov/public_html/prac_uno_serie.htm.

App BlueTooth Serial Controller, bajada de Google Play, mayo 2018.