

COMUNICACION POR PROTOCOLO
RS-232

ASIGNATURA: REDES INDUSTRIALES

REPORTE DE PRACTICA
ANGEL ARTURO RAMIREZ SUAREZ

2 DE JUNIO DE 2013

Índice general

1. Introducción.	3
2. Bases teóricas.	4
2.1. Protocolo RS-232	4
2.2. MikroC for PIC32	5
3. Desarrollo de la practica.	7
3.1. Materiales.	7
3.2. Herramientas.	7
3.3. Desarrollo.	7
4. Conclusiones.	15
List of Figures	17
A. Programa de prueba de puerto UART por protocolo RS-232.	18
B. Localizar el puerto COM al cual pertenece el conector convertidor DB-9 a USB serial.	

Capítulo 1

Introducción.

El protocolo RS-232 es un protocolo de comunicación serial asíncrono, esto es, que no tiene un orden de envío de datos entre los dispositivos, por lo que se vuelve necesario el cuidar la sincronización del envío para evitar pérdidas de información o fallos en la comunicación.

Otra de sus características principales es ser un protocolo punto a punto, esto es, que solamente permite la comunicación de un dispositivo con respecto a otro empleando una terminal de comunicación determinada. No permite la creación de redes.

El protocolo puede trabajarse de manera asíncrona o síncrona y tipos de canal simplex, half duplex o full duplex. En un canal simplex los datos siempre viajarán en una dirección, por ejemplo desde DCE a DTE. En un canal half duplex, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección. En un canal full duplex, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de handshaking de la RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado. [1]

Capítulo 2

Bases teóricas.

2.1. Protocolo RS-232

Es un protocolo desarrollado para la comunicación serial de dispositivos sencillos, ampliamente utilizado debido a la facilidad de comunicación y las ventajas en coste que representa la comunicación serial. Hace uso de conectores de tipo DB-25, sin embargo, es común observar dispositivos con conectores de tipo DB-9, de 9 pines debido a su menor costo.

El interfaz eléctrico utiliza una conexión eléctrica asimétrica con circuitos no equilibrados, todos referenciados a tierra. Los estados lógicos son definidos por los siguientes niveles de voltaje 2.1:

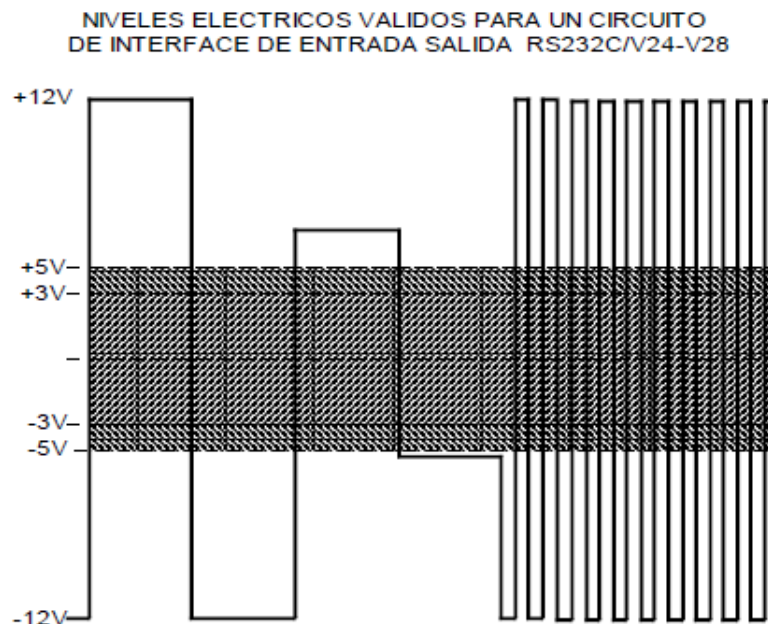


Figura 2.1: Especificaciones de voltaje del circuito RS-232. Tomada de [3]

Como puede observarse, el estándar considera uno lógico a todo valor de voltaje entre -5 y -12 volts, y un cero lógico a aquellos valores entre 5 y 12 volts positivos, con una zona de transición (zona de operación prohibida) de 5 a -5 volts.

La interfaz se utiliza a una razón de menos de 20Kbps para una distancia menor de 15m. En la práctica se pueden exceder estos límites utilizando cables de baja capacidad en entornos eléctricamente poco ruidosos.

El protocolo RS-232 normal en el conector tipo DB-9 utilizado en esta práctica es el siguiente 2.3:

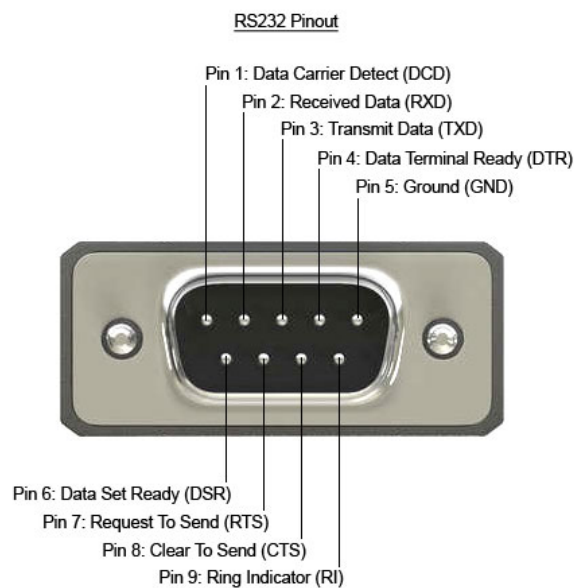


Figura 2.2: Conector del circuito RS-232 con indicación de cada uno de sus significados.

2.2. MikroC for PIC32



Figura 2.3: Logo de MikroC.

MikroC for PIC32 es un entorno de programación de sistemas embebidos, esto es, sistemas que no dependen de una computadora y que son dedicados

a realizar una tarea específica. Carecen de la versatilidad de una computadora y sus recursos son más limitados, sin embargo, son mucho más compactos y eficientes en realizar la tarea específica para la cual son programados.

Un modelo popular debido a la facilidad y amplios recursos que poseen es el microcontrolador de tipo PIC, creados por la empresa Mikroelektronika [4]. Al ser hardware propietario, no tienen la misma apertura que otros microcontroladores como el Arduino, sin embargo tienen la ventaja de tener soporte de la empresa y apoyo técnico.

De estos, la versión utilizada para la elaboración de esta práctica es el PIC32, versión de microcontroladores PIC caracterizados por altas prestaciones y una tasa de transferencia de datos de 32MBytes, lo cual los hace altamente potentes en sus labores de procesamiento y altamente eficientes, requiriendo únicamente 3.3 volts para operar.

El lenguaje utilizado por la empresa es conocido como MikroC, el cual es una versión muy similar al lenguaje C convencional, con algunas instrucciones específicas para los microcontroladores. Al ser de nivel medio su aprendizaje es mucho más sencillo que el uso de otros lenguajes como Ensamblador.

Capítulo 3

Desarrollo de la practica.

3.1. Materiales.

- Módulo RS-232 - Integrado MAX232.
- 2 Terminal block de 2 terminales.
- 4 Capacitores de 4.7 uF.
- Base para circuito integrado de 8 pines.
- Conector DB-9 hembra.
- Alambre para puente.

3.2. Herramientas.

- Cautín 30 Watts.
- Cable convertidor DB-9 serial macho a USB.
- Computadora portátil.
- Aplicación MikroC for PIC32.
- Aplicación HID-Bootloader de Mikroelektronika.
- Microcontrolador MINI-32 PIC32MX534F064H.

3.3. Desarrollo.

1. Para el desarrollo de la práctica se elaboró una aplicación para el microcontrolador MINI-32 que emplea el PIC32MX534F064H utilizando el software MikroC for PIC32, el cual es una versión elaborada específicamente para el PIC32. Dicha aplicación puede observarse en el Apéndice A.

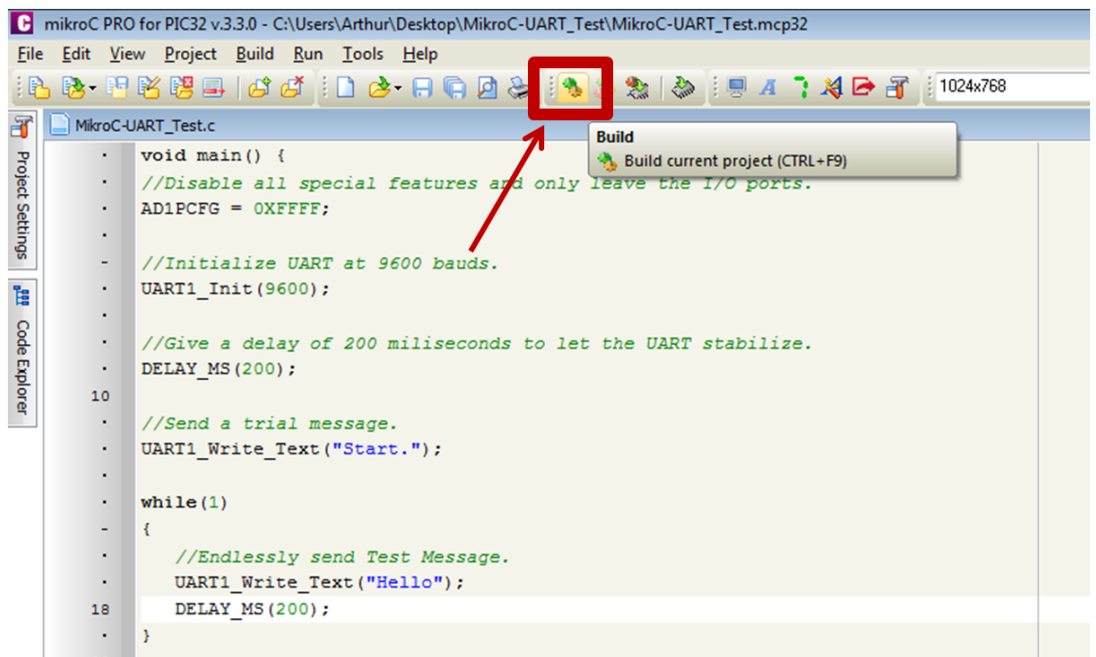


Figura 3.1: Compilación de programa de prueba UART en MikroC for PIC32.

2. Una vez elaborada la aplicación, se hace clic en el botón compilar y se generará un ejecutable .hex que deberá ser cargado en el microcontrolador MINI-32. En la figura 3.1 puede observarse la compilación del archivo.
3. Posteriormente debe cargarse en el microcontrolador, para lo cual se hace uso del HID-Bootloader que provee la compañía Mikroelektronika, creadora de estos dispositivos. En la figura 3.2 puede observarse la apertura del programador
4. Abierto el programador e inicializado el microcontrolador presionando el botón **RESET**, se selecciona el directorio en el cual se encuentra el archivo .hex producto de la compilación del programa. En la figura 3.3 puede observarse la conexión del microcontrolador una vez se ha hecho clic en el botón **RESET**, tras lo cual se tienen 5 segundos para hacer clic en el botón *Connect* (Conectar).
5. Se hace clic en *Load Hex* (Cargar .Hex) y se selecciona el archivo que se encuentra en la carpeta donde se compiló la aplicación, tal y como se observa en la figura 3.4.
6. Se hace clic en el botón *Upload* (Cargar) y se espera. Ésto cargará el archivo al microcontrolador que entonces realizará infinitamente las acciones programadas. En la figura 3.5 se observa el proceso de carga.
7. Posteriormente se procede a la elaboración del circuito MAX-232. Para ello se utilizó el esquemático que se ilustra a continuación en la figura 3.6 puede observarse la conexión del circuito MAX232 hacia un microcontrolador

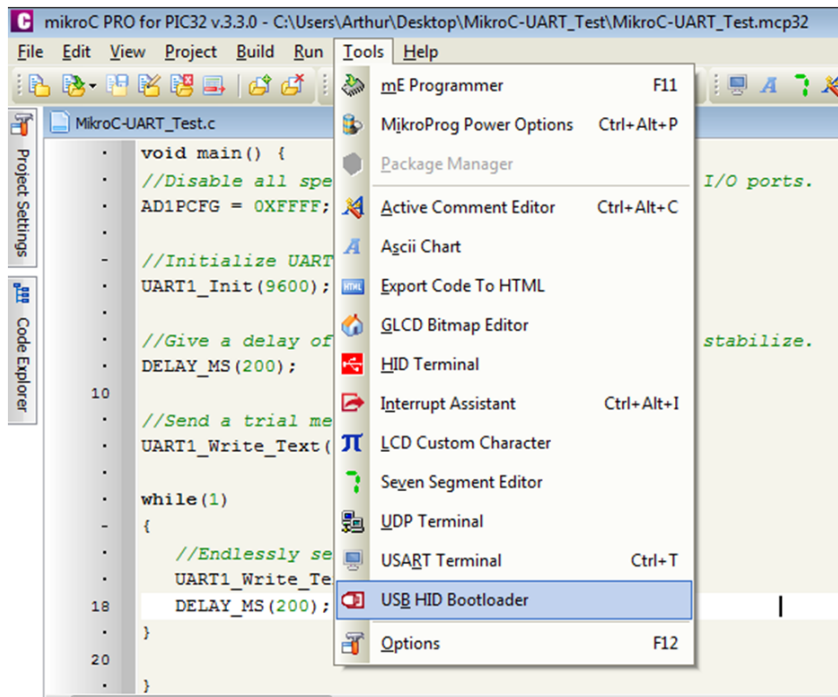


Figura 3.2: Apertura del programador HID-Bootloader.



Figura 3.3: Conexión al microcontrolador.

PIC32MX460F512L. La única diferencia consiste en la localización de los pines RX y TX del microcontrolador, que se detallan en el siguiente punto.

8. Las salidas del circuito MAX-232 se conectan a los pines D2 y D3 del

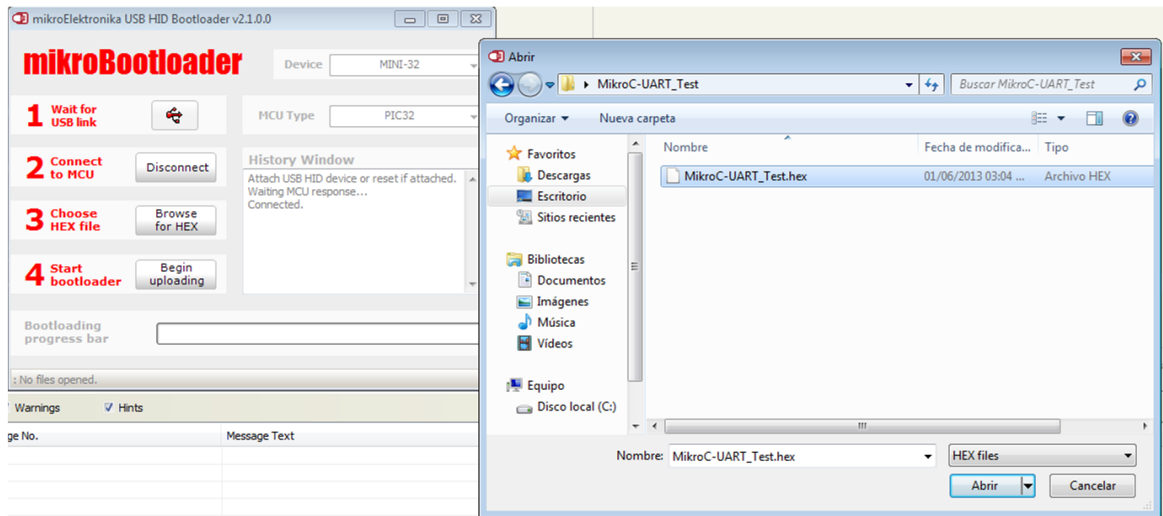


Figura 3.4: Selección del archivo .hex.

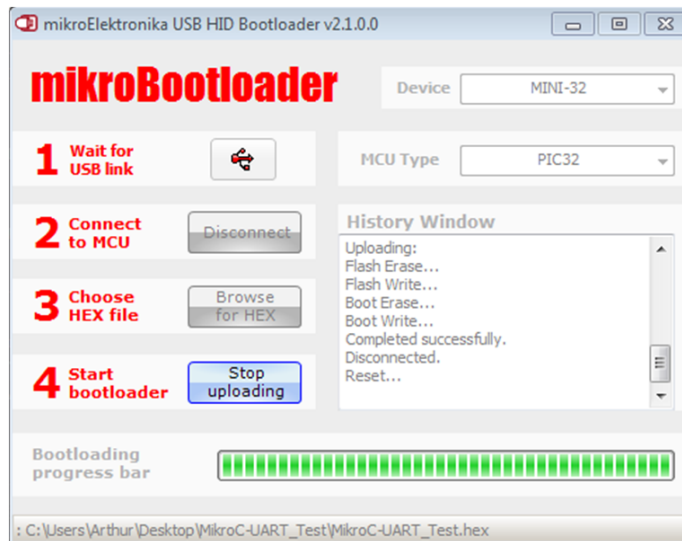


Figura 3.5: Carga del archivo .hex.

circuito tal y como se observa en la figura 3.7 para el PIC32MX534F064H.

9. Una vez conectado, se conecta el cable convertidor DB-9 a serial al conector en el circuito armado del MAX-232. En la figura 3.8 se observa dicha conexión.
10. Después es necesario probar la recepción adecuada del circuito, para lo cual se utiliza un programa llamado Hyperterminal, nativo de Windows y que permite supervisar la entrada y salida de una red. En este caso se abre una nueva sesión de Hyperterminal y se selecciona un nuevo nombre para la conexión. Se hace clic en OK tal y como se observa en la figura

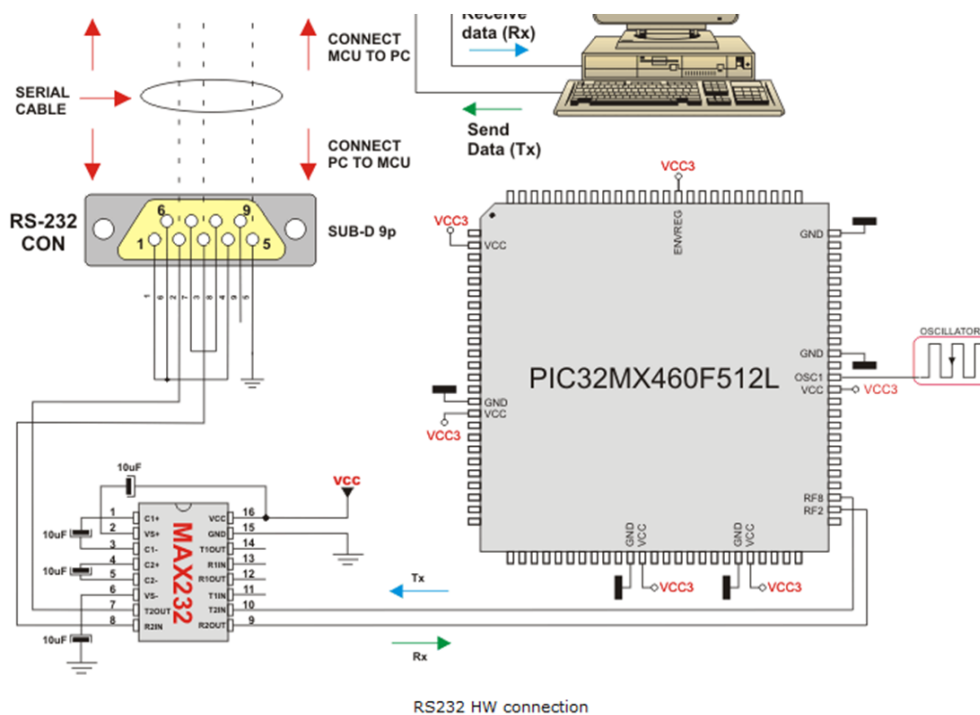


Figura 3.6: Esquema de conexión del circuito MAX-232, tomado de [2].

3.9.

11. Se selecciona el puerto COM que coincide con el que el equipo reconoció perteneciente al cable convertidor DB-9 a serial usb. En este caso el puerto reconocido es el COM7, tal y como se observa en la figura 3.10. Se hace clic en OK. En el apéndice B se indica cómo reconocer a qué puerto pertenece.
12. Se selecciona el nivel de baudios de operación del dispositivo. En este caso se elige 9600 que es el valor programado en la aplicación elaborada en MikroC for PIC32. En caso que se hubiera programado otro valor de baudios, se seleccionará en el menú desplegable tal y como se observa en la figura 3.11. Se hace clic en OK y se tendrá configurada la conexión.
13. Se alimenta el circuito MAX-232 con 5 volts de corriente directa y se observará que en la consola de la Hyperterminal se mostrará el mensaje programado en el microcontrolador PIC32MX534F064H. En la figura 3.12 se observa la conexión realizada en la cual se alimenta el circuito utilizando un Arduino Mega 2560 como fuente de voltaje constante de 5 volts CD.
14. Finalmente se observa en la consola el mensaje programado tal y como se ve en la figura 3.13.

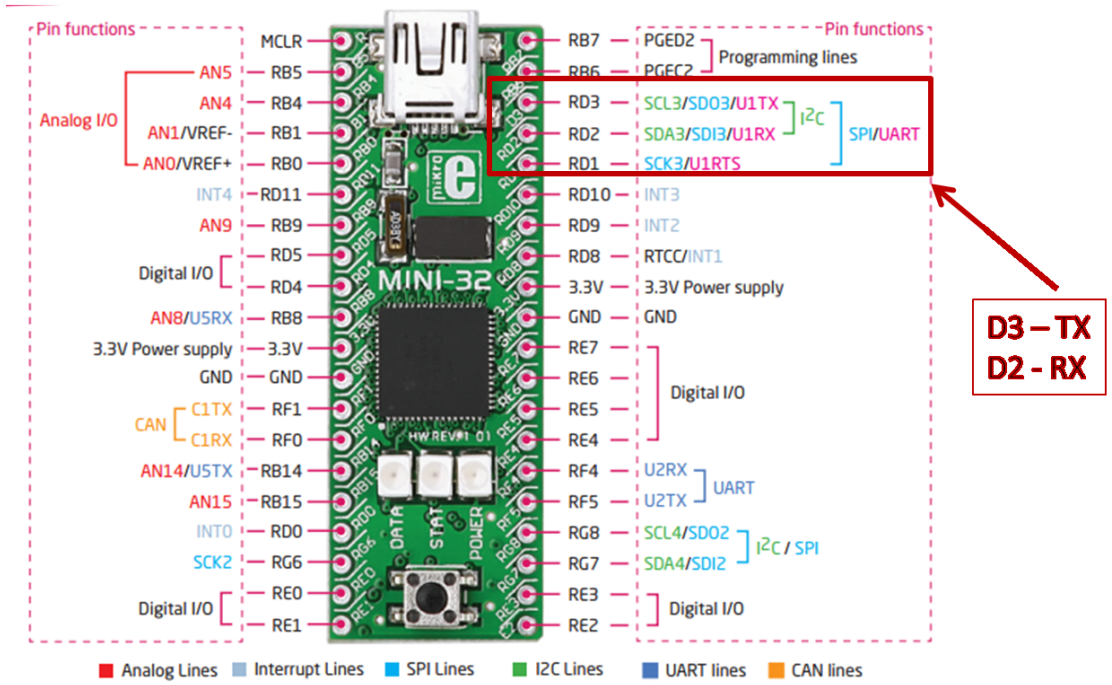


Figura 3.7: Pines del microcontrolador MINI-32, tomado del manual de [2].

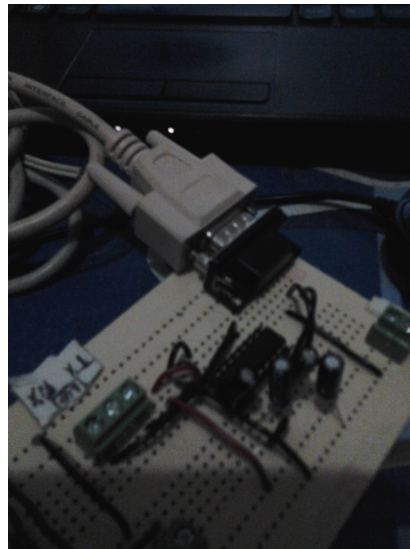


Figura 3.8: Circuito MAX-232 conectado a la computadora por medio del convertidor DB-9 a USB serial.

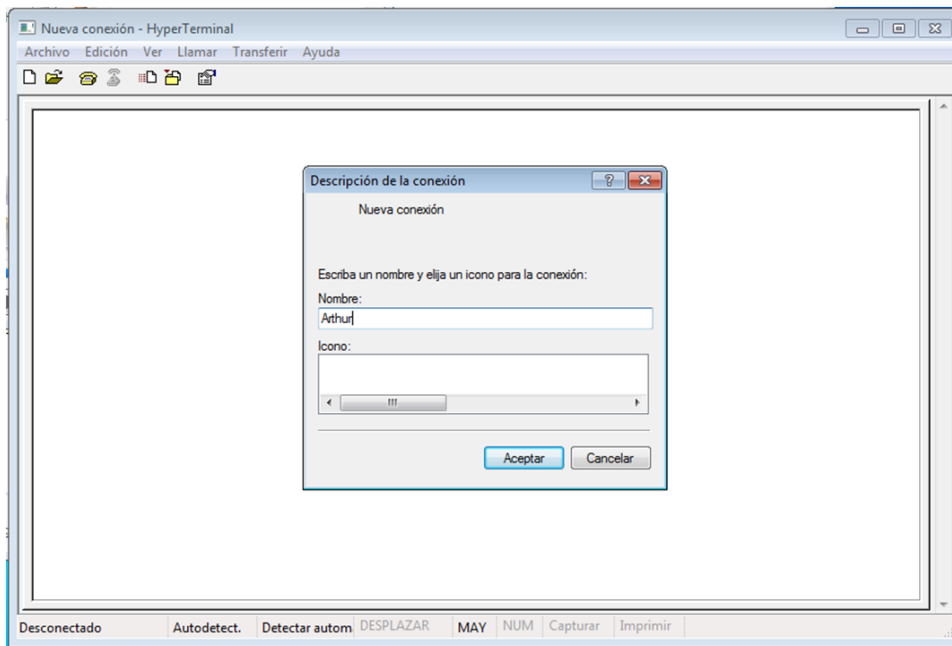


Figura 3.9: Inicialización de una nueva conexión en Hyperterminal.

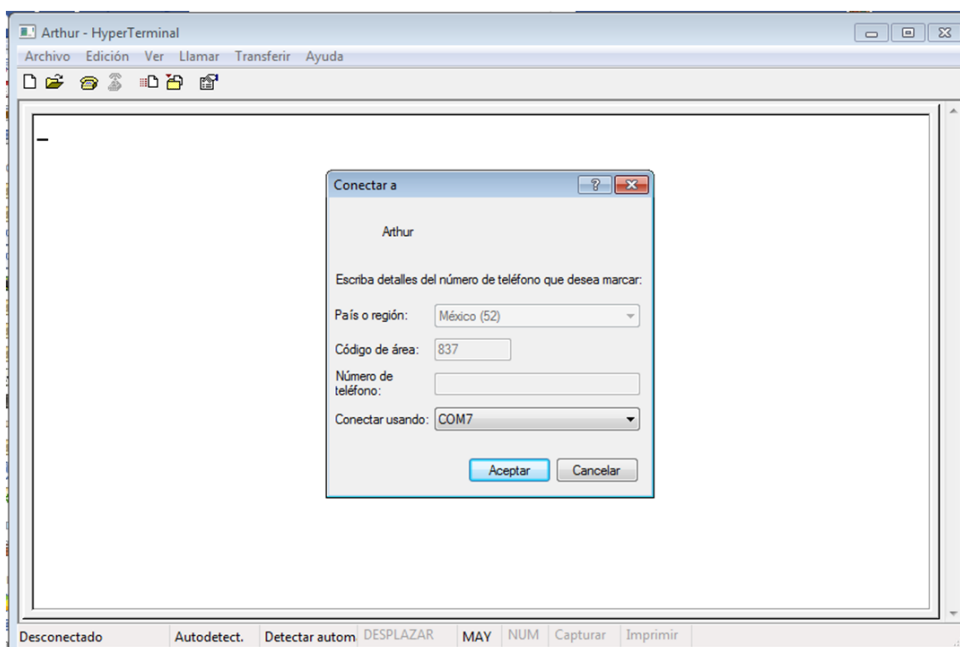


Figura 3.10: Selección del puerto COM.

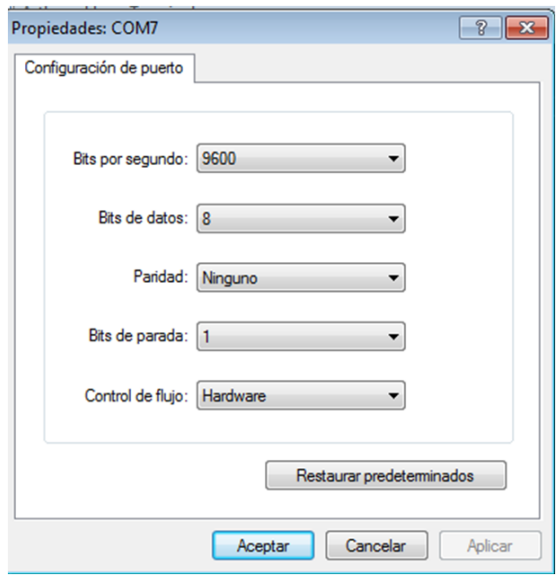


Figura 3.11: Selección del valor en baudios, en este caso de 9600 baudios.

Figura 3.12: Conexión del circuito RS-232 a la computadora, tras lo que se logrará la comunicación serial.

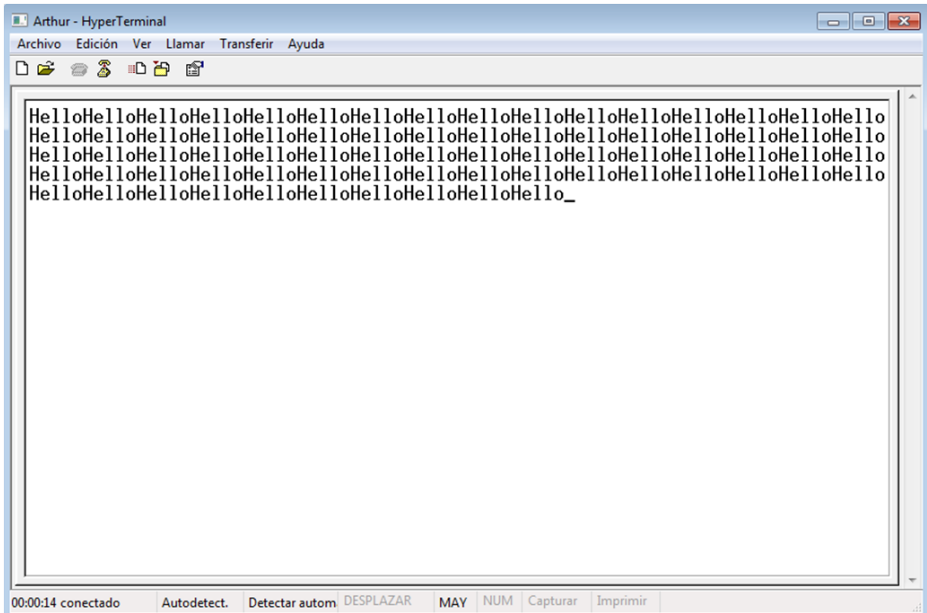


Figura 3.13: Conexión en operación.

Capítulo 4

Conclusiones.

Mediante la elaboración de esta práctica se obtuvieron conocimientos referentes a la elaboración del circuito para el protocolo de comunicación RS-232, el cual es un protocolo punto a punto altamente utilizado en la industria para la comunicación de dispositivos simples.

Bibliografía

- [1] Wikipedia: RS-232. Visto por última vez: 2013-02-06.
<http://es.wikipedia.org/wiki/RS-232>
- [2] MikroC for PIC32 Compiler Help.
- [3] ESPElectronicDesign - Protocolo RS-232C. Visto por última vez: 2013-02-06.
http://www.espelectronicdesign.com/Protocolos/protocolo_RS232.php
- [4] Mikroelektronika. Visto por última vez: 2013-02-06.
<http://www.mikroe.com/>

Índice de figuras

2.1. Especificaciones de voltaje del circuito RS-232. Tomada de [3] . . .	4
2.2. Conector del circuito RS-232 con indicación de cada uno de sus significados.	5
2.3. Conector del circuito RS-232 con indicación de cada uno de sus significados.	6
3.1. Compilación de programa de prueba UART en MikroC for PIC32.	8
3.2. Apertura del programador HID-Bootloader.	9
3.3. Conexión al microcontrolador.	9
3.4. Selección del archivo .hex.	10
3.5. Carga del archivo .hex.	10
3.6. Esquema de conexión del circuito MAX-232, tomado de [2]. . . .	11
3.7. Pines del microcontrolador MINI-32, tomado del manual de [2]. . .	12
3.8. Circuito MAX-232 conectado a la computadora por medio del convertidor DB-9 a USB serial.	12
3.9. Inicialización de una nueva conexión en Hyperterminal.	13
3.10. Selección del puerto COM.	13
3.11. Selección del valor en baudios, en este caso de 9600 baudios. . . .	14
3.12. Conexión del circuito RS-232 a la computadora, tras lo que se logrará la comunicación serial.	14
3.13. Conexión en operación.	14
B.1. Apertura de propiedades del sistema.	19
B.2. Apertura del administrador de dispositivos.	20
B.3. Puertos COM disponibles.	21

Apéndice A

Programa de prueba de puerto UART por protocolo RS-232.

```
void main() {
//Disable all special features and only leave the I/O ports.
ADIPCFG = 0XFFFF;

//Initialize UART at 9600 bauds.
UART1_Init(9600);

//Give a delay of 200 miliseconds to let the UART stabilize.
DELAY_MS(200);

//Send a trial message.
UART1_Write_Text(" Start. ");

while(1)
{
//Endlessly send Test Message.
UART1_Write_Text(" Hello ");
DELAY_MS(200);
}
}
```

Apéndice B

Localizar el puerto COM al cual pertenece el conector convertidor DB-9 a USB serial.

Para localizar el puerto utilizado por el convertidor DB-9 a USB serial, se requiere acceder al administrador de dispositivos, en el cual se observarán todos los dispositivos conectados a la computadora. Para ello se siguen los siguientes pasos:

- Se hace clic en Inicio, y clic derecho en MI PC / Equipo tal y como se observa en la figura B.1.

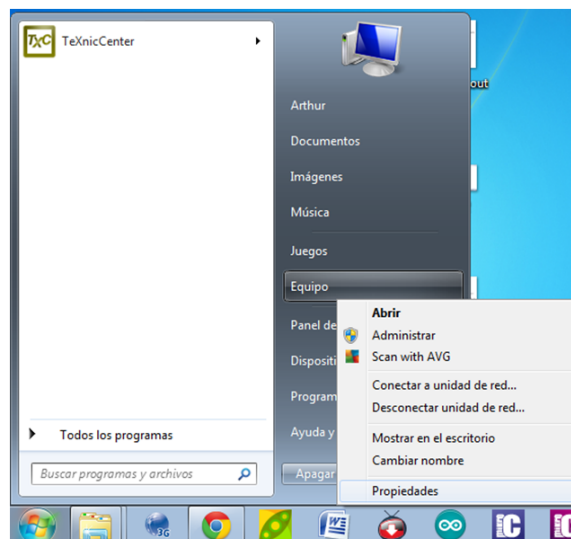


Figura B.1: Apertura de propiedades del sistema.

- Se selecciona el Administrador de dispositivos como se muestra en la figura B.2.

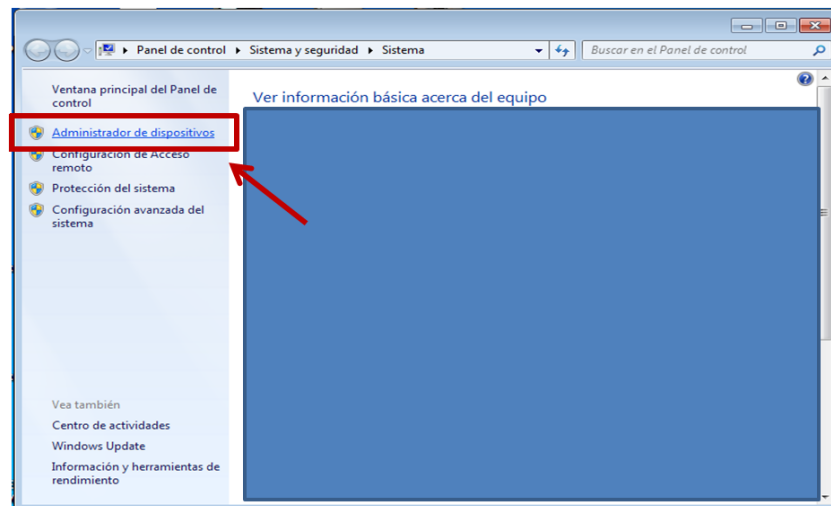


Figura B.2: Apertura del administrador de dispositivos.

- Finalmente, abierto el administrador de dispositivos, se hace clic en la pestaña de puertos COM y LPT y se busca el que indica Prolific to Serial COM Port como se muestra en la figura B.3.

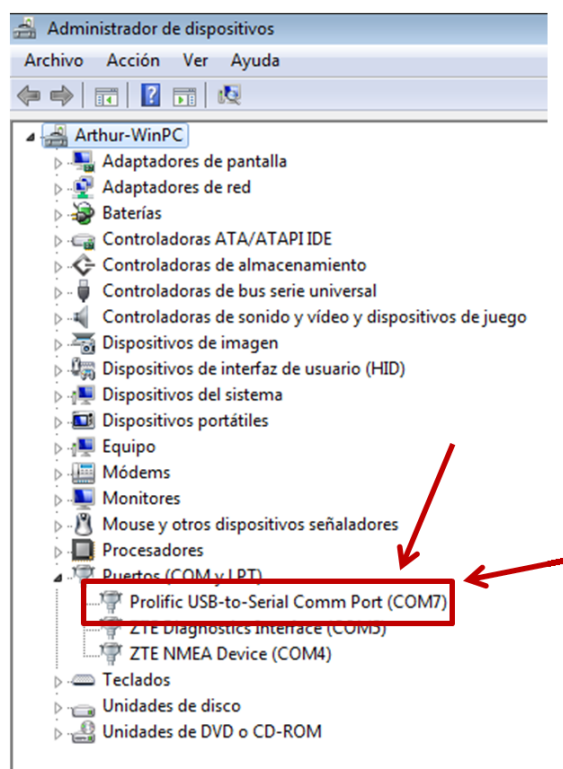


Figura B.3: Puertos COM disponibles.