|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | **Revisado por:** | **Autorizado por:** | **Vigente desde:** |
|  |  | **Dr. Francisco Javier Solorio Ordaz** | **2 de agosto de 2019** |



Índice de prácticas

Práctica 1: Tema 1 x

Práctica 2: Tema 2 xx

Práctica 3: Tema 3 xx

Práctica 4: Interfaz Software-Hardware xx

Práctica 5: Tema 5 xx

**Práctica #4**

**Interfaz Software Hardware**

***1. Seguridad en la ejecución***

|  |  |  |
| --- | --- | --- |
|  | **Peligro o Fuente de energía** | **Riesgo asociado** |
| 1 | Tensión alterna | Electrocución |

***2. Objetivos de aprendizaje***

OBJETIVO GENERAL: Entender los protocolos existentes para comunicar una computadora (software) con elementos mecánicos (hardware).

OBJETIVOS ESPECÍFICOS:

* Entender qué es la comunicación serie y un protocolo.
* Realizar la comunicación entre la computadora y un microcontrolador.

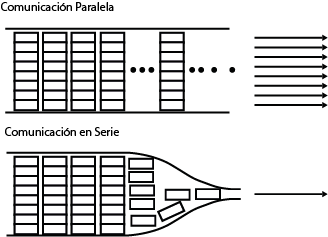
***3. Introducción***

La interfaz software-hardware es de amplio uso en la vida académica y profesional para un ingeniero en mecatrónica. Una de las principales actividades que realiza es la comunicación entre el software y microcontroladores con la intención de tener salidas mecánicas controladas, o bien para obtener información del exterior con el uso de sensores.

La manera más sencilla de comunicar una computadora con un microcontrolador es mediante el uso de la comunicación serial. La comunicación serial es un proceso por el cual se puede enviar un bit a la vez. Tiene como ventajas un uso menor de cables que una proceso en paralelo. En éste caso, con dos líneas se puede enviar y recibir información. En contraparte, la comunicación serie es más lenta que la paralela.

**Proceso en Serie y en Paralelo**

Se puede pensar en un cuello de botella. La información que se requiere enviar es “apilada”. En un puerto paralelo cada bit de información se envía mediante un cable. En el caso de un puerto serie, la información se “desapila” y se envía cada bit por separado. La imagen siguiente muestra ésta analogía.



*Figura 1.  Diferencia entre la comunicación serie y paralela.*

De ésta manera, es sencillo entender porqué se requieren más cables de comunicación en un proceso paralelo y por qué sólo se usan dos en un proceso serie. En el caso de las computadoras modernas los fabricantes han optado por el uso de puerto serie debido a que es más sencillo de manufacturar así como de enviar y recibir información debido a la menor cantidad de cables usados. El más conocido es el puerto USB (Universal Serial Bus). Sin embargo existen más protocolos de comunicación. A continuación se detallan los protocolos más utilizados:

* Ethernet
* I²C
* RS-232
* RS-485
* SPI
* Universal Serial Bus

Ahora que se ha introducido el término “***protocolo***”, se debe entender primero a qué se refiere ésto. Un protocolo es un conjunto de reglas y normas que están obligadas a cumplir todas la máquinas con la finalidad de mantener un orden y facilitar la comunicación entre las mismas. Sin éste conjunto de reglas, la comunicación sería caótica y por tanto imposible. Algunas de las características que norman los protocolo son:

* Detección de la conexión física
* Handshaking
* Cómo iniciar y finalizar un mensaje.
* Procedimientos en el formateo de un mensaje
* Detección de pérdida de información

Para comprender de una manera más práctica se hará uso de una analogía. Los seres humanos tienen un proceso de comunicación de tipo serie. Al hablar y escuchar los hacemos palabra por palabra en un orden específico. La primer palabra hablada, es la primer palabra escuchada. Por esta razón, el proceso de comunicación es en serie. Sin embargo, se puede hacer uso de distintos idiomas, como francés, español, alemán, chino entre otros. Cada idioma, tiene un conjunto de reglas y normas que se deben seguir para comunicarse correctamente. De éste modo, el idioma es una analogía al protocolo. Todo los protocolos enlistados anteriormente usan un proceso de comunicación de tipo serie. Cada uno de ellos tiene un conjunto de normas y reglas que los hace diferente. Ésto implica que los dispositivos que se van a comunicar deben de poder hacer uso de estos protocolos.

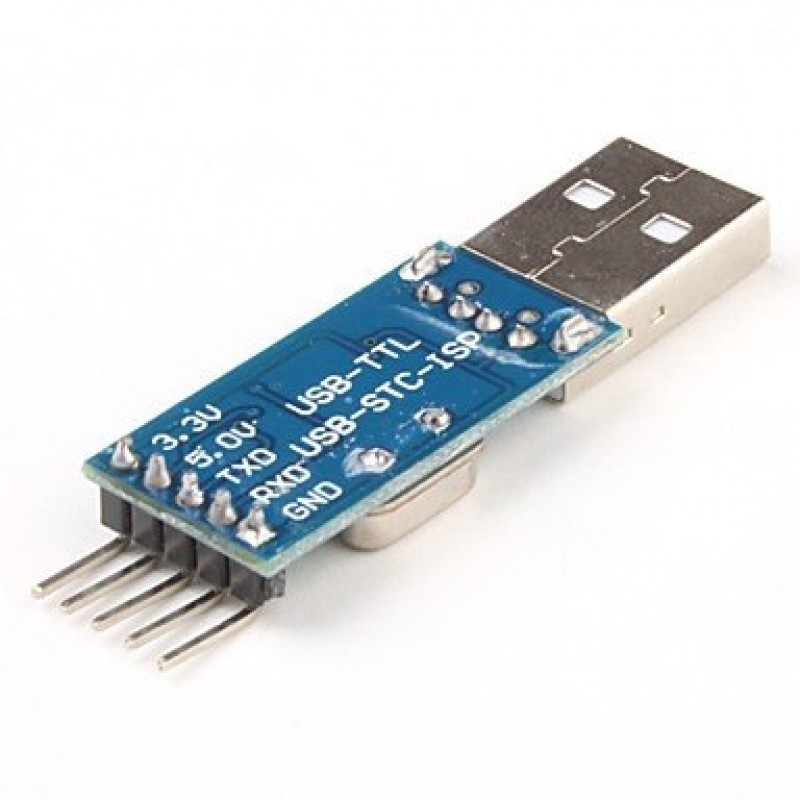
**Convertidor USB a RS-232**

En el caso de un microcontrolador debemos de comunicarnos usando el protocolo RS-232. Por ésta razón es necesario un convertidor de usb a RS232. en ocasiones éste convertidor también es conocido como un convertidor USB a Serial TTL. De éste modo podemos ver la siguiente imágen que ayudará a entender éste proceso.



*Figura 2.  Uso de un convertidor USB a RS232.*

En al figura 2 podemos observar los pines usados en el protocolo USB. Éstos p9nes son D+ y D-. Ambos se utilizan para enviar y recibir información, pero nunca al mismo tiempo, es decir, en algún momento, envían información y en otro reciben. Ambos pines desempeñan esta función. A ésto se le conoce como half Duplex. Quiere decir, que o se recibe información o se envía, pero nunca al mismo tiempo. En el caso de los pines entre el convertidor y el microcontrolador encontramos RX (recepción de datos) y TX (transmisión de datos). Cada pin desempeña una función diferente; el pin TX sólo envía y el RX sólo recibe. De éste modo es necesario conectar RX con TX y viceversa. A éste tipo de conexión se le conoce como conexión cruzada y la dirección de la información es de un sólo sentido. Tal cual como se muestra en las flechas. En el caso de algunas tarjetas de desarrollo como “Arduino”, éste convertidor ya está implementado en la tarjeta misma, por lo que no es necesario conectar un hardware extra. Si se desea hacer uso de un microcontrolador sin la tarjeta, se debe usar un convertidor como el CH340 o FTDI232.

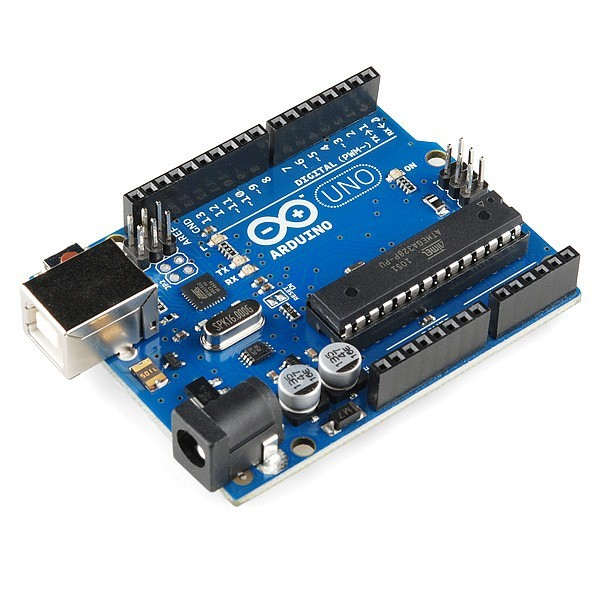


*Figura 3.  Imagen de un convertidor USB a RS232 (también conocido como Serial TTL).*

**Comunicación entre la Computadora y un Microcontrolador**

Para realizar la comunicación entre la computadora y un microcontrolador debemos hacer us

***4. Material y equipo***

***5. Desarrollo***

**Actividad 1*.* Comunicación Computadora a Microcontrolador**

El alumno realizará una práctica donde comunicará la computadora con una tarjeta de desarrollo o microcontrolador sin necesidad de salidas mecánicas.

**Actividad 2*.* Interfaz Software Hardware**

El alumno obtendrá información del entorno utilizando un sensor o entrada electrónica (LDR, optoreflectivo, pushbutton) y generar una salida mecánica o eléctrico (motor, servomotor, LED) haciendo uso de comandos desde la computadora.

***6. Bibliografía***

* CEBALLOS SIERRA, Francisco Javier. **Microsoft C#. Curso de programación**. México, Alfaomega, 2007
* DEITEL, Harvey y Deitel, PAUL. **C# Cómo programar**. España, Pearson, 2007
* LÓPEZ ROMÁN, Leobardo. **Metodología de la programación orientada a objetos**. México, Alfaomega, 2007
* LAAKMANM MCDOWELL, Gayle. **Cracking the Coding Interview: 189 Programming Questions and Solutions**. Palo Alto, CA., CareerCup, 2016.
* BELL Douglas y Parr Mike. **C# para estudiantes**. México, Pearson, 2010.