

Decodificador de Código de Barras, utilizando el Microcontrolador AT90S2313

Resumen

En la actualidad los sistemas enfocados a la identificación de objetos o identidades, están diseñados para trabajar con lectores de cinta magnética o lectores de códigos de barras principalmente. El presente trabajo describe el diseño de un sistema lector de código de barras basado en un microcontrolador AT90S2313 de ATMEL y un lector de barras SR11 Data Logic, el cual puede ser adaptado para realizar funciones que impliquen identificación, tales como reconocimiento de productos, inventarios, reconocimiento de personal, etc.

Introducción

El concepto de codificación basándose en barras no es nuevo, desde la década de los 60's, donde los ordenadores todavía tenían como lotes de entrada de información las tarjetas perforadas, un grupo de ingenieros probaron de manera informal otro tipo de lote, soporándose en el empleo de los entonces raros sensores ópticos. En esos años el concepto no pudo concretarse comercialmente, considerando que años más tarde el empleo de material magnético abarcó el terreno dejado por los sensores ópticos. El paso de los años y el avance de la tecnología dio una nueva oportunidad a la codificación de barras, principalmente como sistema de preciación y serialización, es decir, asignar precios y número de inventarios.

A decir verdad, los códigos de barras pueden tener mil y una aplicaciones, desde sistemas de seguridad (como acceso a edificios, cerraduras, etc.) hasta un modesto lenguaje. De acuerdo a la aplicación se puede implementar un formato de codificación y es lógico pensar en ello: cada diseñador adecua el formato a sus propósitos. Como mero comentario, en Internet se tiene llegada a varios programas gestores de los códigos

de barras más comerciales, como el UPC-A, EAN-13, Code 39, FIM, PostNet, etc. Todos éstos formatos tienen el inconveniente de que el algoritmo de codificación no se coloca a disposición del público, es decir, a aquellos que intenten decodificarlo (incluyéndonos) les resulte muy difícil hacerlo. Pero la solución se mencionó de manera implícita: la elaboración de un código propio.

Justificación y planteamiento del sistema

Justificación

El primer punto a discutir fue: ¿Qué aplicación se tiene en mente? Realmente no se tiene un fin específico, se quiso diseñar un sistema que pudiera expandirse o adecuarse casi a cualquier sistema.

Por último punto, ¿Qué tan confiable es el sistema? El sistema por sí sólo tiene una confiabilidad del 100%, aplicado a un sistema específico, su confiabilidad irá creciendo de acuerdo al formato en el código de barras empleado, si se utiliza un formato muy complejo, la confiabilidad tenderá a aumentar, si el formato es muy sencillo, el crédito se reducirá por el hecho de que cualquiera podrá decodificarlo. En este caso se dio un enfoque educativo al proyecto, es decir, que sea fácil al usuario comprender la forma cómo funciona el sistema.

Bases y planteamiento del proyecto

Por las razones expuestas anteriormente, el formato más sencillo es a través de anchos de barra. Si se define una barra con un ancho específico para que represente un carácter (ya sea dígito o letra), se pueden definir otros anchos de barra para definir otros caracteres.

Físicamente, ¿qué se necesita para implementar el sistema? El actor principal es el lector de código de barras, seguido por un dispositivo capaz de procesar la señal que genere el anterior, este dispositivo es un microcontrolador.

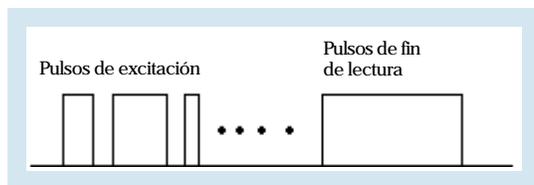
El papel del lector será el de reproducir eléctricamente el código de entrada para que de esa forma el microcontrolador pueda procesar la señal y realizar su decodificación. En este caso, el ancho de barras es el parámetro a medir por el microcontrolador, el cual le indicará el código que le corresponde a cada secuencia de barras.

Sin embargo, se depende de factores muy estrictos: el tiempo y constancia con la que el usuario pase la tarjeta por el lector. Si la tarjeta se pasa de manera uniforme y a velocidad constante, el sistema en general no deberá tener problemas para procesar la señal, pero si alguno de estos parámetros varía se tendrán problemas como códigos erróneos provocados por la forma en la que opera el lector de barras SR11, la forma en que se soluciona este problema se explicará a su tiempo.

Se propone hacer uso de los temporizadores que forman parte del Microcontrolador para realizar la lectura del lector de barras, de tal forma que sólo durante los estados altos de la señal generada por el lector se accione dicho temporizador.

Para la realización de este proyecto se hizo uso de un lector de barras de la compañía *Data Logic Optic Electronics*, seriado como SR 11. Consiste en un par emisor - receptor infrarrojos con un sencillo circuito interno que genera una señal equivalente al código de barras sentido, como se muestra en la figura 1.

Figura 1. Formas de señales generadas por el SR 11



El último pulso corresponde a un método propio del lector para indicar fin de lectura. Al visualizar la señal en un osciloscopio se observa que el pulso de fin de lectura abarca aproximadamente un 32% de la duración total de la señal. Esto significa un primer inconveniente: hay que programar al microcontrolador, de manera que desprecie el efecto de éste último pulso.

Cuando mencionamos el problema de los tiempos, implícitamente requerimos un microcontrolador que

ocupe el mínimo de tiempo para realizar sus instrucciones. Este es uno de los factores por el cual se eligió el microcontrolador AT90S2313 de ATMEL, ya que con un cristal de 4MHz es capaz de procesar instrucciones en 250 ns (en promedio).

La pregunta es ahora: ¿Cómo procesar la señal?. Para empezar, se define que el número de caracteres máximo para el sistema será de diez, estos incluyen sólo los dígitos primarios (0 a 9). Esto significa que el lector generará diez pulsos equivalentes a las diez barras más el pulso de final de lectura, como se propuso eliminar el pulso de final de lectura, la forma de ignorarlo es llevar un conteo de pulsos leídos, y cuando sea igual al número máximo de caracteres, colocar al microcontrolador en modo *stand by* durante un tiempo mayor a la duración del pulso de fin de lectura.

Pero el problema va más allá del pulso final, ¿cómo sabremos qué valor tiene cada barra? La solución es sencilla: implementar una barra de referencia. Si definimos a esta barra de un ancho tal que el resto de las barras sean un submúltiplo de la referencia, una simple operación aritmética bastará para decodificar cada barra. Aquí entran en juego los temporizadores, los cuales sólo se activan en los tiempos altos de los pulsos, los valores de la cuenta son equivalentes a los anchos de las barras. De esa forma, una división de la referencia entre cada una de las barras restantes basta para realizar la decodificación.

Otro aspecto que se debe considerar es: ¿Dónde guardar los valores? El AT90S2313 cuenta con un área de RAM estática, la cual puede ser accesada con una simple instrucción en muy poco tiempo, ya sea para lectura o escritura. Lo óptimo es que al momento de obtener los valores del conteo se almacenen simultáneamente en la RAM estática. Una vez almacenados todos los valores, se les puede recuperar para ser procesados.

Anteriormente se mencionó el deseo de hacer dinámico el número de barras, esto es, que se pueda modificar el número de caracteres por código durante el tiempo de ejecución. Para realizar lo anterior se hace uso de las dos interrupciones externas del microcontrolador, una para aumentar el número y otra para disminuirla. Existe un rango de una a diez barras.

La visualización de la información o el código leído es hecha mediante un display de cristal líquido.

Por último, ¿Cómo hacer flexible al sistema? Como se hizo notar, el diseño no tiene una aplicación especí-

fica, lo indicado es que cada código pueda ser enviado a un sistema mayor. La forma que se eligió para transmitir la información es mediante el protocolo de comunicación RS232. De esta forma el sistema sólo sirve como intérprete entre el lector de barras y un sistema que realiza una función específica que requiera decodificar códigos de barras.

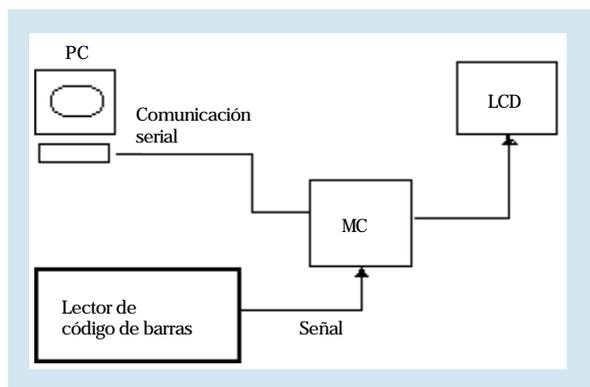


Figura 2. Diagrama a bloques del sistema

Diseño del sistema

En general el lector de barras proporciona un nivel alto (5 Volts) en la parte oscura de la tira de barras, y un nivel bajo (0 Volts) en la parte blanca, es decir, el lector lee las partes negras.

A una velocidad moderada se observó que el tiempo que tardaba el pulso en alto ante una barra de 5mm de ancho es de 23ms aproximadamente, como se muestra en la figura 3.

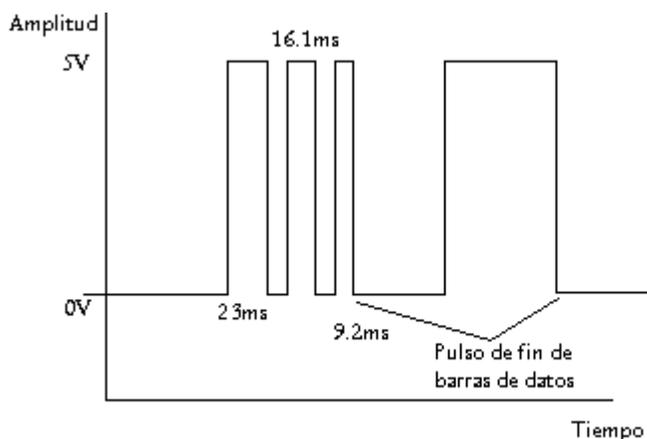


Figura 3. Señales obtenidas del lector de barras

Protocolo del código de barras.

El protocolo de las barras elaboradas consiste en una barra de ancho base (5mm) cada 5 números, el tiempo que tarde esta barra será la referencia del número 1, ya

que si se quiere colocar un uno en el código de barras se tendrá que poner una barra del mismo grosor, ya que el número estará dado por la siguiente fórmula.

$$n = \frac{W_{base}}{W_{dato}} \quad \text{Ecuación 1}$$

donde:

n = Valor numérico de la barra dato.

Wbase = Ancho base de la primer barra.

Wdato = Ancho de la barra dato.

Dato	Ancho de barra base (mm)	Ancho de barra dato (mm)
0	5	5.50
1		5.00
2		2.50
3		1.60
4		1.25
5		1.00
6		0.83
7		0.71
8		0.62
9		0.55

Tabla 1. Anchos recomendables para cada dato

Tomando un ancho base de 5mm, se tienen los anchos mostrados en la tabla 1, para cada número que se tenga que hacer.

La principal función de esta barra base de 5 mm es la de eliminar el efecto que tiene el hecho de pasar las barras por el lector con una velocidad que no sea constante, debido a esto se decidió colocar una barra base al inicio y otra después de 5 barras.

El algoritmo final del programa se muestra en la figura 4.



El algoritmo final del programa se muestra en la figura 4.

Cabe mencionar que durante la lectura de barras dato y barras base, la bandera de interrupción global está desactivada, ya que no se puede modificar la cantidad de datos a leer al mismo tiempo en que se está pasando la tarjeta por el lector y cuando se está calculando e imprimiendo los valores de las barras en el LCD.

Para calcular el valor numérico de las barras a partir de la barra base es necesario hacer una división de 16 bits, de la cual sólo se toma la parte entera del resultado, desechando el residuo, esto nos da un margen de certeza más amplio para los números pequeños (0, 1, 2, 3, 4, 5 y 6), sin embargo dicho margen se reduce para números grandes (7, 8 y 9).

La figura 5 exhibe el diagrama de conexiones del sistema.

El impreso del proyecto se muestra en la figura 6, el cual fue realizado con OrCAD.

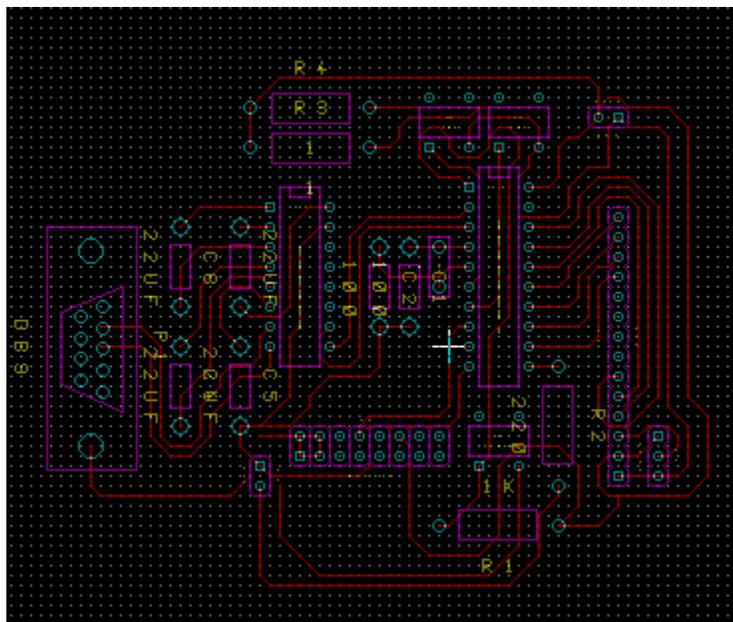


Figura 6. Impreso del proyecto

Conclusiones

Debido a la exactitud que demanda el funcionamiento de este sistema de lectura de barras, se elige el uso del microprocesador AT90S2313, debido a que con un temporizador de 16 bits y un cristal de 4MHz, se logra una cuenta total en aproximadamente 35 ms y dado que en la caracterización del lector se muestra que la duración promedio de la barra más gruesa es de 32 ms, esto hace que el sistema final que sea muy exacto, además que debido a la resolución se eliminan errores, así se pase el código por el lector de barras muy rápido o a una velocidad moderada.

El tiempo que se tarda el microcontrolador en obtener y guardar los datos en la ESTATICRAM del mismo se resiente un poco en la exactitud del sistema para los números grandes o barras pequeñas. Este tiempo es considerable ya que se trata de 24 datos (dos a la vez) en la memoria interna del microcontrolador.

No se recomiendan los números 7, 8 y 9, pues se pierde exactitud en el procesado de la señal. La razón estriba en que la diferencia del ancho de barra entre esos tres números es prácticamente indistinguible, se tendría que ampliar la barra base o de referencia, o bien, disminuir proporcionalmente el ancho de barras de los números citados, con el riesgo de que el lector no pueda reproducir fielmente la señal **T**

Respecto a la comunicación serial

Al momento de pasar la tarjeta por el lector, el sistema manda la cantidad de datos leídos y posteriormen-

te cada uno de los datos, cabe mencionar que la tarjeta no se puede pasar por el lector de manera inversa. La comunicación serial está configurada a 9600 baudios. Este punto refleja precisamente la flexibilidad del sistema. Como indicamos desde un principio, el proyecto no tiene una aplicación específica, por ello se decidió agregar la propiedad de tener disponibles los datos en formato serial, volviendo de esta manera al sistema muy versátil, además de esa manera el programador puede hacer el uso que considere pertinente a los datos enviados.

Elaboración de las barras.

El valor recomendado para la barra base es de 5 mm por lo que las barras de los datos serán más delgadas excepto el cero, que medirá 5.5mm aproximadamente, el espacio entre las barras puede ser de entre 2 y 3 mm, pero se recomienda que después de las barras bases se dejen 3 mm mínimo, ya que el programa tarda un poco en asimilar dichas barras como barras bases.

Referencias

- [1] ATMEL
1997 AVR enhanced risc microcontroller data book
ATMEL Company.
- [2] ATMEL
1997 AVR Development Tools User Guide ATMEL
Company.

Notas

- [3] AND
1996 Liquid Crystal Display Catalog 1996 AND Company.
[4] INTERSIL
1987 Component Data Catalog 1987 INTERSIL Company.

Enrique Guzmán Ramírez *

Arias Martínez Oswaldo **

Hernández Méndez Arturo **

Zambrano Nila Juan G. **

* *Profesor Investigador del Instituto de Electrónica y Computación, Universidad Tecnológica de la Mixteca*

** *Alumnos de la carrera de Ingeniería en Electrónica, Universidad Tecnológica de la Mixteca*