

Mini Robot controlado por una interfaz

Br. Abraham Salgado García, Br. Angelous Albenis Palacios Fajardo,
Br. Kevin Ariel Rivera.

Facultad de Ciencias e Ingenierías.¹

Recibido 3 de mayo 2013-Aprobado 28 de octubre 2013.

INTRODUCCIÓN

Una computadora cuenta con diversos dispositivos de entrada y salida, llamados periféricos tales como: puerto paralelo, serie, USB, joystick, etc. A estos periféricos podemos conectarle sensores, actuadores y programar para tomar decisiones lógicas. En la cual se usara un puerto paralelo. Los puertos de comunicación de la PC son de particular interés para el estudioso de la electrónica ya que le permiten utilizar una computadora personal para controlar todo tipo circuitos electrónicos, utilizados principalmente en actividades de automatización de procesos, adquisición de datos, tareas repetitivas y otras actividades que demandan precisión. Un puerto paralelo es una interfaz entre una computadora y un periférico cuya principal característica es que los bits de datos viajan juntos, enviando un paquete de byte a la vez. Es, decir se implementa un cable o una vía física para cada bit de datos formando un bus. En este proyecto, se encuentran tres etapas; la del puerto paralelo, el módulo de transistores con sus respectivos releí, y los dispositivos que van conectados a los releí, los dispositivos que se van a utilizar son motores. La elaboración de estas etapas, proporcionan el control de un Mini Robot que tiene velocidad hacia adelante, hacia atrás y giros de derecha e izquierda.

PRESENTACIÓN DEL PROYECTO

En éste artículo nos concentraremos en transferencias de ocho bits ya que ésta es la configuración del puerto paralelo de una PC. Un típico sistema de comunicación en paralelo puede ser de una dirección (*unidireccional*) o de dos direcciones (*bidireccional*). El más simple mecanismo utilizado en un puerto paralelo de una PC es de tipo unidireccional y es el que analizaremos en primer lugar. Distinguimos dos elementos: la *parte transmisora* y la *parte receptora*. La parte transmisora coloca la información en las *líneas de datos* e informa a la parte receptora que la información (los datos) está disponible; entonces la parte receptora lee la información en las líneas de datos e informa a la parte transmisora que ha tomado la información (los datos).

Observé que ambas partes sincronizan su respectivo acceso a las líneas de datos, la parte receptora no leerá las líneas de datos hasta que la parte transmisora se lo indique en tanto que la parte transmisora no colocará nueva información en las líneas de datos hasta que la parte receptora remueva la información y le indique a la parte transmisora que ya ha tomado los datos, a ésta coordinación de operaciones se le llama *acuerdo ó entendimiento*. A la coordinación de operaciones entre la parte transmisora y la parte receptora se le llama *handshaking*, que en español es el acto con el cual dos partes manifiestan estar de acuerdo, es decir, se dan un apretón de manos.

El handshaking

Para implementar el *handshaking* se requieren dos líneas adicionales. La línea de estroboscopia es la que utiliza la parte transmisora para indicarle a la parte receptora la disponibilidad de información. La línea de admisión es la que utiliza la parte receptora para indicarle a la parte transmisora que ha tomado la

¹ Trabajo dirigido por los maestros Bismarck Santana y Álvaro Segovia

Mini Robot controlado por interfaz

información (los datos) y que está lista para recibir más datos. El puerto paralelo provee de una tercera línea de handshaking llamada en inglés *busy* (*ocupado*), ésta la puede utilizar la parte receptora para indicarle a la parte transmisora que está ocupada y por lo tanto la parte transmisora no debe intentar colocar nueva información en las líneas de datos. Una típica sesión de transmisión de datos se parece a lo siguiente:

Parte transmisora:

- La parte transmisora checa la línea *busy* para ver si la parte receptora está ocupada. Si la línea *busy* está activa, la parte transmisora espera en un bucle hasta que la línea *busy* esté inactiva.
- La parte transmisora coloca la información en las líneas de datos.
- La parte transmisora activa la línea de *strobe*.
- La parte transmisora espera en un bucle hasta que la línea *acknowledge* esté activa.
- La parte transmisora inactiva la línea de *strobe*.
- La parte transmisora espera en un bucle hasta que la línea *acknowledge* esté inactiva.
- La parte transmisora repite los pasos anteriores por cada byte a ser transmitido.

Parte receptora:

- La parte receptora inactiva la línea *busy* (asumiendo que está lista para recibir información).
- La parte receptora espera en un bucle hasta que la línea *strobe* esté activa.
- La parte receptora lee la información de las líneas de datos (y si es necesario, procesa los datos).
- La parte receptora activa la línea *acknowledge*.
- La parte receptora espera en un bucle hasta que esté inactiva la línea de *strobe*.
- La parte receptora inactiva la línea *acknowledge*.
- La parte receptora repite los pasos anteriores por cada byte que debe recibir.

Se debe ser muy cuidadoso al seguir éstos pasos, tanto la parte transmisora como la receptora coordinan sus acciones de tal manera que la parte transmisora no intentará colocar varios bytes en las líneas de datos, en tanto que la parte receptora no debe leer más datos que los que le envíe la parte transmisora, un byte a la vez.

El hardware del puerto paralelo

El puerto paralelo de una típica PC utiliza un conector hembra de tipo D de 25 patitas (DB-25 S), éste es el caso más común, sin embargo es conveniente mencionar los tres tipos de conectores definidos por el estándar IEEE 1284, el primero, llamado 1284 tipo A es un conector hembra de 25 patitas de tipo D, es decir, el que mencionamos al principio. El orden de las patitas del conector es éste (ver imagen 1). El segundo conector se llama 1284 tipo B que es un conector de 36 patitas de tipo Centronics y lo encontramos en la mayoría de las impresoras; el tercero se denomina 1284 tipo C, se trata de un conector similar al 1284 tipo B pero más pequeño, además se dice que tiene mejores propiedades eléctricas y mecánicas, éste conector es el recomendado para nuevos diseños. Observe que el puerto paralelo tiene 12 líneas de salida (8 líneas de datos, *strobe*, *autofeed*, *init*, y *select input*) y 5 de entrada (*acknowledge*, *busy*, *falta de papel*, *select* y *error*).

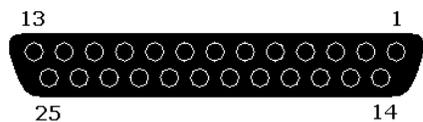


Imagen 1.

Interfaz para el puerto paralelo

Actualmente, la mayoría de los puertos instalados en las computadoras son de tipo multimodal configurable a través del BIOS de la máquina. Eléctricamente, el puerto paralelo entrega señales TTL y

como tal, teóricamente, se le puede conectar cualquier dispositivo que cumpla con los niveles de voltaje específicos de la lógica TTL, sin embargo el hardware del puerto paralelo está muy limitado en cuanto a su capacidad de manejo de corriente, por ésta razón se debe ser muy cuidadoso con el manejo de las señales del puerto, un corto circuito puede dañar permanentemente la tarjeta madre de la PC. Para disminuir lo más posible el riesgo de daños al puerto utilizamos un circuito integrado 74LS244 como etapa separadora y al mismo tiempo mejoramos la capacidad de manejo de corriente, de esta forma podemos conectar una serie de diodos emisores de luz (LED) que nos indiquen la actividad en las líneas de datos del puerto paralelo. El circuito se detalla en el siguiente diagrama (imagen 2 y 3):

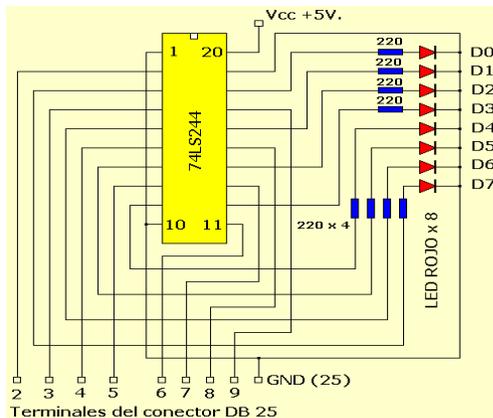


Imagen 2

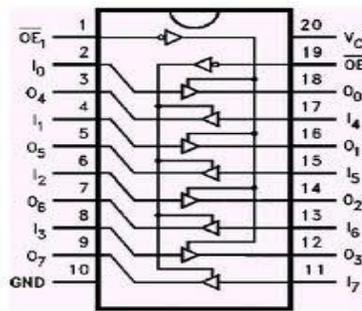


图 13 74LS244 内部结构

Imagen 3

Por cada línea de entrada que tomamos directamente del puerto paralelo existe una etapa amplificadora-separadora dentro del circuito integrado 74LS244 que nos permite trabajar con una tasa de entrega de corriente suficiente para desplegar en los diodos emisores de luz la información escrita en las líneas de datos del puerto. Además es posible habilitar o deshabilitar el despliegue del

nibble de orden inferior ó superior del byte escrito en el puerto. Colocando en un nivel lógico alto la patita 1 del CI 74LS244 inhabilitamos el despliegue del nibble de orden bajo, un nivel lógico alto en la patita 19 evita el despliegue del nibble de orden alto. Para comodidad, conecto las patitas 1 y 19 permanentemente a tierra de tal manera que sea posible visualizar la actividad en los diodos emisores de luz (LED). En el diagrama se especifican con números las correspondientes patitas del conector DB-25. Obviamente se requiere de una fuente regulada de 5 Voltios para operar éste circuito, además los siguientes materiales:

- 1 Circuito Integrado TTL 74LS244.
- 8 Diodos Emisores de Luz.
- 8 Resistencias de 220 Ohm, 1/2 Watt.
- 1 Cable y conector para el puerto paralelo.

Naturalmente lo más recomendable es probar el correcto funcionamiento del circuito antes de conectarlo al puerto paralelo de la PC. Ensamble el circuito, preferentemente en un circuito impreso, y conéctelo a una fuente regulada de 5 Voltios, conecte temporalmente un extremo de una resistencia de 10,000 Ohm a una línea de entrada, el resto de las líneas de entrada conéctelas a tierra. El otro extremo de la resistencia conéctelo directamente al borne positivo de la fuente de alimentación para inducir una señal TTL alta, el respectivo LED debe encender. Con un trozo de alambre conectado a Tierra, toque temporalmente el extremo de la resistencia que está conectado a la línea de entrada para inducir una señal TTL de lógica baja, el LED se debe apagar. Repita ésta operación para cada una de las ocho líneas de entrada. Una vez que ha verificado el correcto funcionamiento del circuito está listo para conectarlo al puerto paralelo de la PC.

En primer lugar apague la computadora y el circuito. Conecte el cable al puerto paralelo asegurándose que el conector esté firme en su sitio. Encienda el circuito y por último encienda la computadora, por regla general, el circuito de restablecimiento de la computadora coloca inicialmente en las líneas de datos del

Mini Robot controlado por interfaz

puerto paralelo un valor igual a 0x0h, por lo tanto todos los diodos deben estar apagados una vez que la computadora ha terminado su proceso de arranque, sin embargo, si algún diodo permanece encendido esto no indica una condición de falla, es responsabilidad del software que Usted escriba para controlar el puerto inicializarlo con un valor adecuado antes de realizar cualquier otra operación.

Escribiendo datos al puerto paralelo

Con ocho bits podemos escribir en el puerto un total de 256 valores diferentes, cada uno de éstos representa un byte de información y cada byte puede representar una acción concreta que nosotros podemos definir de acuerdo a nuestras necesidades. En éste artículo el objetivo es entender cómo trabajar con el puerto paralelo, por lo tanto hagamos un programa que nos permita escribir un número cualquiera entre 0 y 255 de tal manera que sea posible visualizar el valor en formato binario. En primer lugar consulte la documentación de su compilador para verificar la correcta sintaxis de la función que nos sirve para escribir en el puerto, en el caso específico de Symantec C++, dicha función es `outp()`. Ésta función requiere dos parámetros, el primero de tipo *unsigned int* que especifica la dirección del puerto paralelo, y el segundo de tipo *char* que especifica el valor a escribir en las líneas de datos de puerto.

Se aprecia la facilidad de manejo de la función, aunque diferentes compiladores dan a sus respectivas funciones nombres diferentes, la mecánica es la misma, se requieren dos parámetros, la dirección del puerto y el valor a escribir en el puerto. El programa determina el número de puertos paralelos instalados en la computadora y almacena sus respectivas direcciones en el arreglo llamado `direccion`, en caso de encontrarse una dirección válida además se establecen en 1 el valor del arreglo llamado `disponible` simplemente como una medida de seguridad adicional. Como resultado de estas acciones el programa despliega un menú basado en los puertos encontrados, en este punto Usted puede seleccionar, en caso de que disponga de más de un puerto, el puerto en donde está conectado el circuito mostrado en el diagrama de arriba.

El dispositivo está formado por tres etapas, la primera está formada por un cable para el puerto paralelo conectado de la computadora hacia la interfaz, esta interfaz es parte de la primera etapa y como se mencionó anteriormente recibe la información. La segunda etapa es el módulo de transistores con sus respectivos releí, donde hay cuatro transistores y cuatro releí; y la tercera etapa del dispositivo son los motores. A continuación se muestra el diagrama donde se presentan las dos etapas del dispositivo (ver imagen 4):

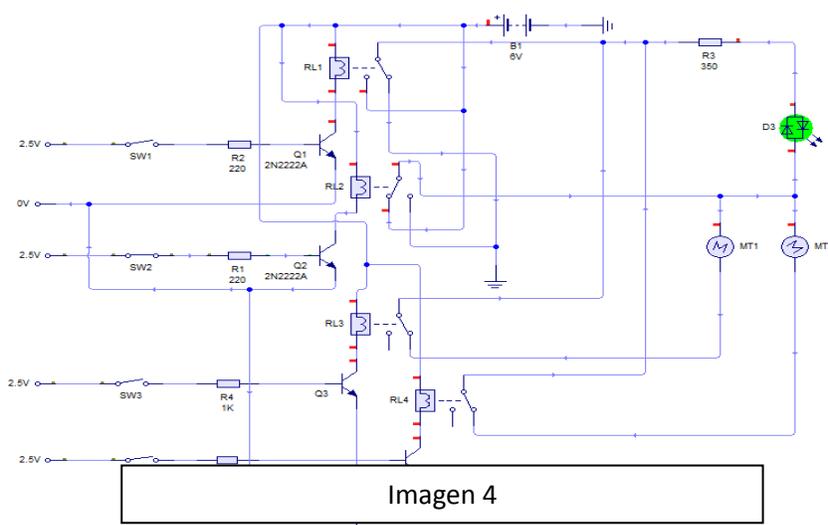


Imagen 4

incluye de 2.5 v.; la interfaz es alimentada con 5 v de CD.

Bueno en la primera etapa, la función es de transmitir datos en donde hay un transmisor y un receptor. Cuando recibe los datos la interfaz se encienden los diodos led indicando así que la información ha sido transmitida y hay 8 salidas de datos del circuito periférico, en este proyecto se utilizan 4 de estas salidas, incluyendo el tierra, para que sean conectadas cada una a un transistor, y a como se muestra en el diagrama cada transistor recibe un pulso que

En la segunda etapa; cuando los transistores reciben 2.5 v en la base de este, interfiriendo una resistencia y el tierra en el emisor, se activa el transistor y como es utilizado como un interruptor se activa el relé, en donde el relé se alimenta con 6 v de CD, la bobina de este componente va conectada del colector de transistor a la terminal positiva de la fuente; que el normalmente cerrado pasa a normalmente abierto del relé. En este primer diagrama llega el primer pulso a el primer transistor y se activa el relé, el cual se encienden los dos motores para direccionar hacia adelante el Mini Robot, donde se enciende un diodo bicolor y los otros transistores están inactivos y no sucede nada, a diferencia del primero. En el segundo pulso; por lógica se activa el segundo transistor, proporcionando que el relé pase el estado normalmente abierto a normalmente cerrado y así encendiendo los motores, direccionando el Mini Robot hacia atrás e indicando el diodo bicolor el estado encendido.

Ahora con el primero y tercer pulso, se activa el motor número uno avanzando hacia adelante, con el diodo bicolor activo. Y en el caso de proporcionar el pulso uno, pulso tres y pulso cuatro se desactiva el motor y el diodo bicolor, también con el pulso uno y pulso cuatro, se activa el motor numero dos con activación del diodo bicolor. Para el pulso dos y tres, se energiza el motor número uno y gira con dirección hacia atrás y si es para el caso del pulso tres, cuatro y dos, energizados sencillamente deshabilitan el giro del motor, bien se energiza el motor número dos y la iluminación del diodo emisor de luz, direccionándose hacia adelante. Montados en el Livewire. Este motor se alimenta con una fuente de 12 v de CD. Estos circuitos, muestran que se utiliza solo el relé segundo y tercero, ya que solo es necesario de utilizar solo esa parte del circuito para hacer girar el motor en direcciones opuestas.

CONCLUSIONES

El dispositivo conformado por las tres etapas; el puerto paralelo con la interfaz, el módulo de transistores y relé, y por los motores; todo este paquete tiene un solo enlace que está en función de trabajar de acuerdo a los datos que se establezcan en el inicio del programa que se construya con la computadora. Sin lugar a duda cuando se analizó cada parte del circuito, nos lleva esta información a varias remodelaciones que se pueden hacer de la segunda etapa del dispositivo, sin ningún problema; ya que la interfaz tiene varias salidas para ser utilizadas, y haciendo el mejor arreglo posible de los circuitos, se llegaría a modular una gran cantidad de estos al alcance de nuestra exigencia.

Y si observamos lo importante de este dispositivo, es el de hacer muchas operaciones desde la computadora o también de un pequeño sistema que este programado para hacer la tarea. Este sistema nos enseña que en una computadora el trabajo es igual a diferencia que la cantidad de arreglos en ella es mucho mayor; estos datos son interesantes, porque se logra entender con más claridad la tecnología que trabaja en función de transmitir y recibir información, claro no abarcando todo en sí.

BIBLIOGRAFÍA

T. L. Floyd. "Fundamentos de Sistemas Digitales". Capítulo 15. Editorial Prentice Hall.
Componentes y materiales electrónicos- Manuel García Gomes.
Instrumentación electrónica moderna- William Cooper.