

Control de Motor por Computadora

Daisy Kang¹, Gabriela Bobadilla²

¹Becaria del PDTA categoría iniciación científica

¹daisykang@gmail.com

Universidad Nacional del Este - Facultad Politécnica

²gabrielabobadilla@yahoo.es

Resumen. El propósito de esta investigación fue controlar un determinado dispositivo físico a través de una computadora aprovechando las posibilidades y ventajas que ésta ofrece. En un primer momento se investigó acerca de los puertos paralelos y el lenguaje de programación C, específicamente las librerías que permiten controlar el puerto paralelo; luego se desarrolló un programa de control en *software* libre, utilizando el *IDE Anjuta*, que envíe información de la computadora al dispositivo físico, en este caso un motor de corriente directa; y, por último, se realizó el circuito eléctrico de control del motor. Los resultados obtenidos fueron satisfactorios en cuanto al programa de control del motor de corriente directa y al funcionamiento del circuito eléctrico interfaz que controla al motor.

Palabras claves: **puerto paralelo, control de motor, lenguaje C.**

Abstract. The purpose of this investigation is to control a given physical device by using a computer, and the possibilities and advantages it offers. At first the parallel port and the C programming language were studied; specifically the libraries that allow us to control the parallel port. After that, a controlling program that could send signals from the computer to a physical device, in this case a direct current motor was developed using open software. For that purpose Anjuta IDE was employed. Finally the electric circuit that would control the motor was constructed. Satisfactory results were obtained when testing the program and the electric circuit interface used to control the motor.

Keywords: **parallel port, control the motor, C programming language.**

1. Introducción

Desde sus orígenes, el ser humano ha estado buscando la forma de facilitar sus tareas utilizando herramientas que le permitan realizar actividades que ocuparían un tiempo muy largo o serían imposibles de realizar sin el ingenio humano.

En la actualidad, el uso de computadoras para controlar dispositivos es una realidad, y lo que hasta hace unos años parecía imposible hoy es un hecho. Hay una gran cantidad de dispositivos que se pueden controlar a través de las computadoras, como por ejemplo: la alarma de la casa, el encendido de una lámpara, etc.; tareas sencillas que a veces se tornan tediosas por ser rutinarias, y que podrían ser realizadas sin mucho esfuerzo por computadoras, que, correctamente programadas, ahorrarían mucho esfuerzo y tiempo.

La idea de poder controlar dispositivos a través de una computadora motivó esta interesante

investigación, y, con el afán de conocer y utilizar estas ventajas, se construyó un programa de control de un motor de corriente directa en *software* libre, utilizando el *IDE Anjuta*, y un circuito interfaz que controle el motor de corriente directa.

3. Objetivos

3.1. Construir un circuito interfaz que controle un motor de corriente directa.

3.2. Desarrollar un software que envíe información de la *PC* a la interfaz de control, utilizando *Software* libre.

3. Materiales

3.1. Puertos Paralelos

El puerto paralelo de la computadora fue el elegido, pues es ideal como herramienta de control de motores, *relés*, *LED's*, etc.

La característica principal de este puerto es que los *bits*¹⁰ de datos viajan juntos enviando ocho bits, o sea, un *byte* completo o más a la vez. Es decir, se implementa un cable o una vía física para cada *bit* de datos formando un *bus*¹¹.

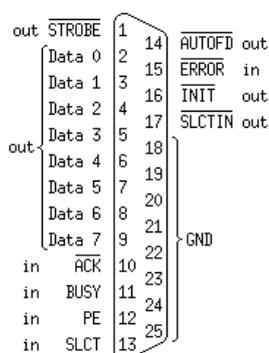
El tipo de puerto paralelo utilizado fue el puerto de impresora, pues es el puerto paralelo más conocido (que cumple la norma *IEEE*¹² 1284, también denominado tipo *Centronics*) que destaca por su sencillez y transmite 8 bits a una velocidad de 2,5 Mbps¹³ [1]

3.1.1. Funcionamiento

El mismo posee un *bus* de datos de 8 bits (*Pin*¹⁴ 2 a 9) y muchas señales de control, algunas de salida y otras de entrada, que también pueden ser usadas fácilmente.

En reglas generales, la dirección hexadecimal del puerto *LPT*¹⁵ 1 es igual a *0x378* (888 en decimal) y *0x278* (632 en decimal) para el *LPT*2.

El puerto paralelo de una computadora posee un conector de salida del tipo *DB25*¹⁶ hembra cuyo diagrama y señales utilizadas podemos ver en la siguiente figura:



¹⁰ Bit: (*Binary digit*) dígito del sistema binario, puede tomar uno de 2 valores diferentes (0 ó 1) [2]

¹¹ Bus: Transporte en inglés. Conjunto de conexiones físicas por donde se comunican los componentes de la computadora. [3]

¹² IEEE: (*The Institute of Electrical and Electronics Engineers*) Organización compuesta de ingenieros, científicos y estudiantes. Es conocida por desarrollar estándares de computación y electrónica.[5]

¹³ Mbps: Mega bits por segundo. Equivale a un millón de bits transferidos por segundo. [6]

¹⁴ Pin: Objeto cilíndrico, pequeño y delgado, de metal utilizado para conectar componentes.

¹⁵ LPT: (*Line Print Terminal*) Es el nombre que reciben en entorno DOS los puertos paralelos. [7]

¹⁶ DB25: Nombre para conectores con formato en D, usados para el intercambio de datos. El numero 25 indica la cantidad de pines que posee el conector. [8]

Si deseamos escribir un dato en el *bus* de salida de datos, debemos escribir el *byte* correspondiente en la dirección hexadecimal *0X378* (888 en decimal) cuando trabajamos con el *LPT*1 y *0x278* (632 en decimal) cuando trabajamos con el *LPT*2. Los distintos *pins* (*bits*) de salida correspondientes al *bus* de datos no pueden ser escritos en forma independiente, por lo que, siempre que se desee modificar uno, se deberán escribir los ocho *bits* nuevamente.

Para leer el estado de los *pins* de entrada (10, 12, 13 y 15) se debe realizar una lectura a la dirección hexadecimal *0x379* (889 en decimal) si trabajamos con el *LPT*1 o bien leer la dirección *0x279* (633 en decimal) si trabajamos con el *LPT*2. La lectura será devuelta en un *byte* en donde el *bit* 6 corresponde al *pin* 10, el *bit* 5 corresponde al *pin* 12, el *bit* 4 corresponde al *pin* 13 y el *bit* 3 corresponde al *pin* 15. [4]

3.2 Lenguaje C

Para la generación del programa de control se decidió utilizar el lenguaje de programación *C*, porque es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente.

En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

Es un lenguaje estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos. Hay que destacar que el *C* es un lenguaje portable, ya que permite utilizar el mismo código en diferentes equipos y sistemas informáticos: el lenguaje es independiente de la arquitectura de cualquier máquina en particular. [9]

3.2.1 Las librerías de Lenguaje C para Linux que fueron utilizadas fueron:

3.2.1.1 ncurses.h

La librería *ncurses.h* permitió programar en modo texto para *Linux*, es decir, capturar las teclas presionadas, etc.

3.2.1.2 sys/io.h

Es una librería utilizada para el manejo de Entrada/Salida a bajo nivel. Para utilizar las funciones de manejo de Entrada/Salida se debe tener en cuenta que se requiere otorgar permiso a un rango de direcciones a las cuales se tiene que acceder; estos permisos se obtienen a través de la función *ioperm* y las señales son enviadas a través de la instrucción *outb*. [10]

2.3 Sistema Operativo Linux

El Sistema Operativo elegido fue *Linux*, que posee la gran ventaja de ser un *software* gratuito que se distribuye bajo la Licencia Pública General de *GNU*¹⁷ o *GPL*¹⁸, además de ser robusto, estable y rápido.

Es usado ampliamente en servidores y supercomputadores, y cuenta con el respaldo de importantes corporaciones. [11]

3.4 Anjuta

Fue utilizado el entorno integrado de desarrollo (*IDE*) para programar en *C* y *C++* en sistemas *GNU/Linux Anjuta*, trabaja con *GTK*¹⁹ y en el escritorio *GNOME*²⁰, el cual era utilizado en este caso, además ofrece un gran número de características avanzadas de programación que facilitan en gran medida las tareas de programación como: un administrador de proyectos, asistentes, plantillas, depurador interactivo y un poderoso editor que verifica y resalta la sintaxis escrita, y aparte de eso, posee la ventaja de ser un software libre, liberado bajo la licencia GPL. [14]

9. Métodos

En un primer momento se investigó acerca de los puertos de comunicación de una computadora; en particular, el puerto paralelo, el cual es uno de los puertos más utilizados para controlar todo tipo de circuitos electrónicos; a continuación se estudiaron los sistemas de

transferencia de datos del puerto paralelo; es decir, la metodología de transferencia y recepción de la información en una computadora.

Luego se estudiaron las distintas funciones de cada uno de los *pin*s de un conector de puerto paralelo estándar y su configuración; también, las direcciones de Entrada/Salida asociadas con el puerto paralelo de la *PC* y las direcciones base asignadas por el *BIOS*.

Con relación al *software*, se averiguó el manejo de las funciones implementadas en Lenguaje *C*, para trabajar con puertos paralelos y su estructura de programación.

Con el fin de conocer mejor el funcionamiento del puerto se realizó primeramente un *software* en Lenguaje *C*, utilizando el *IDE turbo C* Figura 1, y la librería *dos.h* la cual permite el envío de señales desde la computadora al puerto paralelo a través de la instrucción:

outport (direcc_puerto, valor);

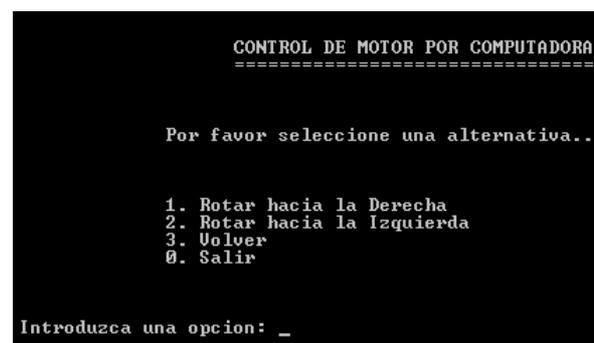


Figura 1: Interfaz de usuario en el IDE turbo C

Para comprobar el funcionamiento del puerto paralelo se construyó una pequeña estructura electrónica, compuesta de un conector de puerto paralelo macho, resistores y *LED*'s Figura 2.

¹⁷ GNU: (*GNU is not Unix*). Conjunto de programas desarrollados por la fundación *Free Software Foundation* (Fundación por el *software* libre) [12]

¹⁸ GPL: (*General Public License*) Licencia Pública General

¹⁹ GTK: (*GIMP tool Kit*) Librería grafica que se utiliza para la interfaz de usuario.

²⁰ GNOME: (*GNU Network Object Model Enviroment*) entorno de escritorio para sistemas operativos del tipo UNIX [13]

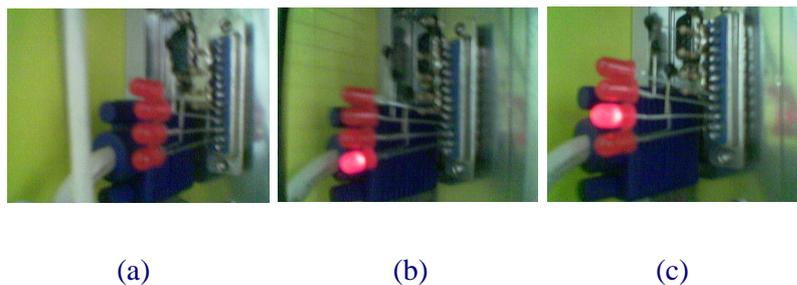


Figura 2. (a) Dispositivo electrónico luego del envío del valor 0, (b) Dispositivo electrónico luego del envío del valor 1, (c) Dispositivo electrónico luego del envío del valor 2.

Con la prueba se pudo obtener un resultado visual de las señales que fueron enviadas desde la computadora al dispositivo, de esta manera se pudo comprobar el correcto funcionamiento del programa de control y del Hardware utilizado.

El siguiente paso fue realizar un software que fuese capaz de controlar un motor a través del puerto paralelo, utilizando el Sistema Operativo Linux, fue utilizada la distribución Ubuntu 6.06 LTS - la versión Dapper Drake - publicada en junio de 2006 Figura 3 (a), y el IDE Anjuta Figura 3 (b).



Figura 3. (a) Sistema Operativo Ubuntu, (b) IDE Anjuta.

Se procedió entonces a conseguir la información suficiente para realizarlo. Se estudiaron las librerías de Lenguaje C que serían necesarias para la realización del software; y se utilizaron las librerías ncurses.h y sys/io.h. Con la librería ncurses.h se pudo hacer interactuar al software

con el usuario y con la librería sys/io.h se le dio al software los permisos necesarios para trabajar con el puerto paralelo, el permiso es otorgado o retirado con la instrucción:
ioperm(direcc_puerto, rango, activar);

La instrucción que permitió enviar señales al puerto paralelo fue la: outb(valor,direcc_puerto);

El software de control realizado, Figura 4, permite el control del movimiento de rotación de un motor de corriente directa a través de una computadora, siendo los estados posibles:

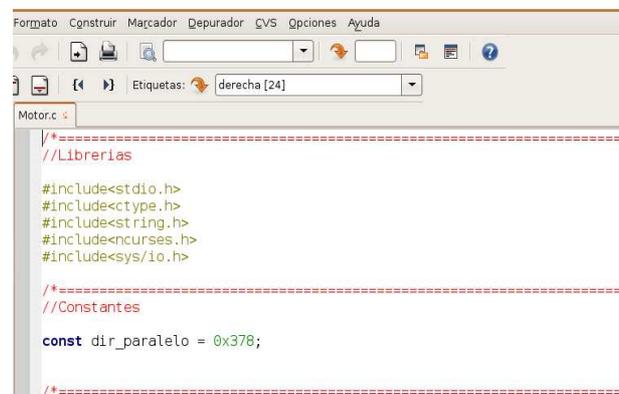


Figura 4. Programa de control desarrollado en el IDE Anjuta.

Desactivado Figura 5 (a), Rotar hacia la derecha Figura 5 (b), y Rotar hacia la izquierda Figura 5 (c). Dando como salida el valor “cero” para desactivarlo, el valor “uno” para rotarlo hacia la derecha y el “dos” para rotarlo hacia la izquierda

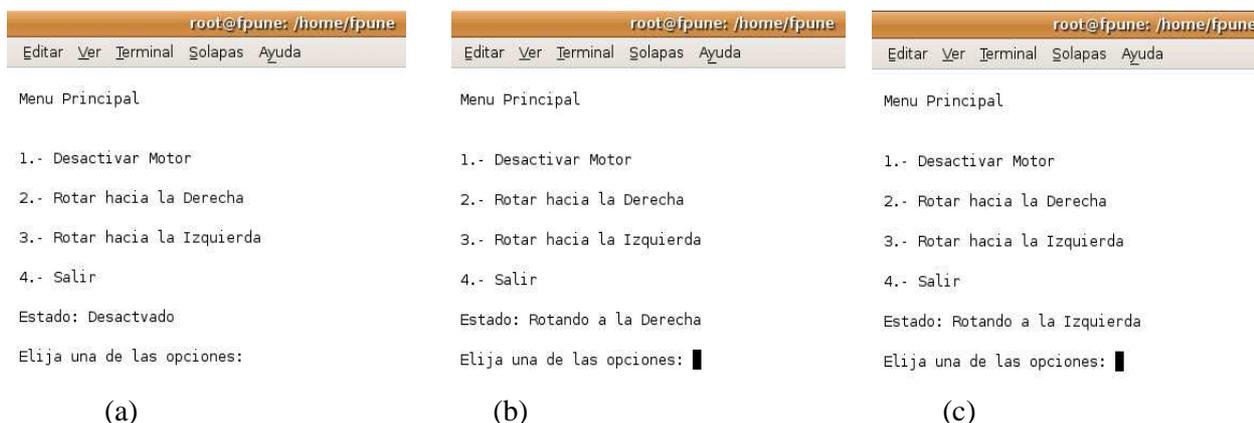


Figura 5. (a) Interfaz de usuario cuando el motor está desactivado, (b) Interfaz de usuario cuando el motor está rotando hacia la derecha, (c) Interfaz de usuario cuando el motor está rotando hacia la izquierda.

Una vez terminado el programa de control en el Sistema Linux, y habiendo comprobado su correcto funcionamiento, se procedió a realizar una estructura compuesta por una placa de conexiones, optoacopladores, un motor de corriente directa y transistores Figura 6 de acuerdo a la interfase principal del circuito ²¹ Figura 7.

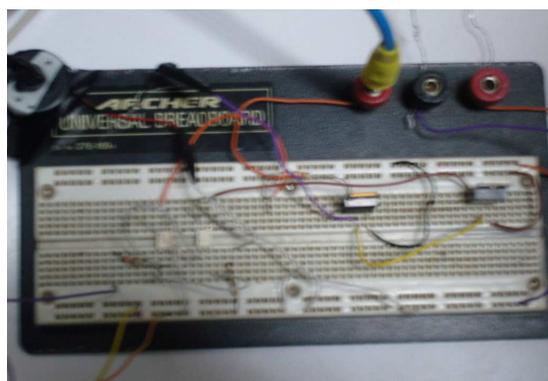


Figura 6. Circuito de control del motor de Corriente Directa

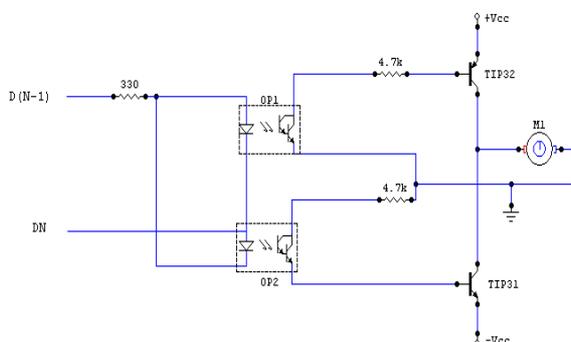


Figura 7. Interface principal del circuito.

5. Conclusión

²¹ Interfase principal del circuito creado en base a la revista Saber Electrónica año 19 / 2005 / N° 218

Las ventajas que se pueden obtener a través del control de dispositivos por computadora son muchas, y la cantidad de recursos que se utilizan en la elaboración es mínima, en comparación con los beneficios que se pueden obtener; pues, puede simplificar algunas tareas fácilmente realizables por una computadora con ahorro de tiempo para el usuario.

El proyecto realizado es un ejemplo de lo que se puede llegar a hacer con una computadora en el control de dispositivos físicos; con él se pudo obtener varios procedimientos para el control de un dispositivo en *software* libre y concluir las ventajas que brinda este recurso.

5. Referencias

- [1] Wikipedia. La enciclopedia libre. Puerto Paralelo [27-03-2008]
http://es.wikipedia.org/wiki/Puerto_paralelo
- [2] Wikipedia. La enciclopedia libre. Bit [01-04-2008]
es.wikipedia.org/wiki/Bit
- [3] Kioskea. Knowledge kiosk [01-04-2008]
<http://es.kioskea.net/pc/bus.php3>
- [4] Todorobot. Usando el puerto paralelo de una PC [01-04-2008]
<http://www.todorobot.com.ar/proyectos/paralelo/paralelo.htm>
- [5] Conectividad. [02-04-2008]
conectividad.com.ar/faqredes.htm
- [6] Sitios España. DICCIONARIO TECNOLOGIA - TERMINOS MAS COMUNES [02-04-2008]

www.sitiosespana.com/diccionarios/TECNOLOGIA/m.htm

[7] Lycos [03-04-2008]
usuarios.lycos.es/Resve/diccioninform.htm

[8] Casadomo [03-04-2008]
www.casadomo.com/diccionario.aspx

[9] EL LENGUAJE "C". FUNDAMENTOS DEL LENGUAJE "C". [07-05-2008]
<http://dgenp.unam.mx/planteles/p4/p4lenguajec/1int-c.htm>

[10] Tomeu Capó i Capó. Programación del Puerto Paralelo con Linux I [11-05-2008]

<http://bulma.net/impresion.phtml?nIdNoticia=1150>

[11] Ciberaula. Ventajas e inconvenientes de Linux [11-05-2008]

http://linux.ciberaula.com/articulo/ventajas_inconvenientes_linux/

[12] Chenico [12-05-2008]
<http://www.chenico.com/glosariogh.htm>

[13] Xonos [12-05-2008]
<http://xonos.wordpress.com/2007/12/23/una-alternativa-de-ubuntu-xubuntu/>

[14] Wikipedia. La enciclopedia libre. Anjuta [12-05-2008]
<http://es.wikipedia.org/wiki/Anjuta>

Haro Martínez, Bernardo. "Control de Motores por Computadora". Saber Electrónica Edición Argentina Año 19/2005/ N° 218 Pág. 60-62