



**PROYECTO - FINAL:**

**CONTROL DEL MOVIMIENTO VEHICULAR  
POR RADIOFRECUENCIA Y  
COMPUTADORA**

CURSO : LENGUAJE DE PROGRAMACION I

ALUMNOS :

- MOISÉS FLORES ALCÁNTARA - CODIGO:

- JAIR CORONEL CANTARO - CODIGO :

TURNO : NOCHE CICLO : 2008-I

HORARIO : LUN 6.30pm

PROFESOR : ING. JUAN CHAVEZ OCAMPO

Lima – Junio 2008

## **INTRODUCCION**

Tal como ya indicáramos en el primer y segundo avance de este proyecto el objetivo de este proyecto radica en realizar aplicación directa de un lenguaje de programación hacia y con el volcado y estudio en la electrónica

Así producto de estos avances hemos realizado el presente trabajo la cual tuvimos que compartir de manera grupal al ir encargando las tareas programadas de inicio .

Esto es, destacamos a Coronel el desarrollo inicial y del borrador del software para el control en lenguaje Turbo C++ indicado por el profesor, el control del puerto paralelo la realizaría Flores, quien tomaría el desarrollo de la interfaz del puerto paralelo hacia fuera, es decir de la habilitación del móvil en sí, así pues, lo obtendríamos agenciándonos en base de algún modelo de carro tipo juguete, el cual para disminuir costos modificaríamos y adaptaríamos una interfaz para controlarlo mediante el computador.

Así luego, habiendo desarrollado progresivamente este trabajo, solo hemos realizado un desarrollo piloto y básico pero que cumple y combina las áreas requeridas por el curso a cabalidad.

Esperando sus comentarios, también les compartimos este trabajo en la web con la finalidad que el alumnado realice mayores investigaciones al respecto.  
( [www.geocities.com/mfloresa](http://www.geocities.com/mfloresa) )

**Los Autores**

# **PROYECTO FINAL : CONTROL DEL MOVIMIENTO VEHICULAR POR RADIOFRECUENCIA Y COMPUTADORA**

## **OBJETIVO**

Presentar informe de la experiencia realizada y habitar al alumno sobre el manejo de los conceptos de programación digital, y control del puerto paralelo de un computador. Asimismo presentar sus aplicaciones sobre los conceptos relativos al uso en la electrónica

Así en esta experiencia tocamos puntos sobre:

- Programación en lenguaje Visual C++
- Conceptos básicos de análisis de circuitos digitales, para la realización de los circuitos
- Control remoto por ondas de radiofrecuencia
- Servomecanismos (motores y reles )

## **DESCRIPCION DE PROYECTO**

Como indicamos, nuestro proyecto consiste en aprovechar entonces las características del manejo de radiofrecuencia y su sistema de control de radiofrecuencia de un carrito de juguete, interactuándola con un sistema mínimo de enlace de una interface doble creada y modificada entre la pc y el móvil para controlar un coche y minimizar efectos de costos, establecida por comandos de dirección inmediatos relativos a un mando manual de orden y de dirección comandado por teclado.

## **CARACTERÍSTICAS**

\* El sistema completo está conformado por:

- Un carro a batería de 9V quien posee un sistema de recepción de radiofrecuencia , El carro a control remoto consta de dos motores, uno para adelante y atrás y otro para derecha e izquierda. Así mismo contiene internamente un diseño inicial de fábrica el cual no hemos modificado nada, salvo el suministro de pilas para su movimiento independiente
- una tarjetería mixta interface, conformada por la adaptación del control de mando del carrito apegado a un conjunto en placa universal consistente en un diseño particular creado de transistores, relays y otros dispositivos como leds que sirven para su señalamiento y monitoreo
- un computador conteniendo un programa en lenguaje turbo C , cuya corrida controla los movimientos del móvil mediante el teclado

- La interface del control a distancia del carro es a base de switches, cada motor dispone de tres contactos, uno común y otros dos que especificarán la dirección del motor, esto significa que para activar un motor hacia una dirección hay que juntar la terminal común con la terminal correspondiente y para que vaya en la dirección contraria hay que juntar el común con la otra terminal.

## **MATERIALES Y EQUIPO :**

### SOFTWARE :

- Sistema operativo Windows Microsoft 98/2000/XP/Vista
- Lenguaje de programación Turbo C++ 3.0 de Borland, con librerías adecuadas
- 2 archivos driver de manejo previo de puertos (driverpuerto.exe y parmor.exe)

### HARDWARE:

01 Computador o Laptop PC compatible, Pentium III, IV o Dual Pro

01 cable centronics para uso del puerto paralelo

01 multímetro

01 soldador y soldadura

01 circuito impresor universal

01 Fuente de Poder o batería de 9v DC

04 resistencias : 8 x 10k , 4 x 2k

04 diodos leds rojos

04 transistores BC548

04 relays de 5v-9v

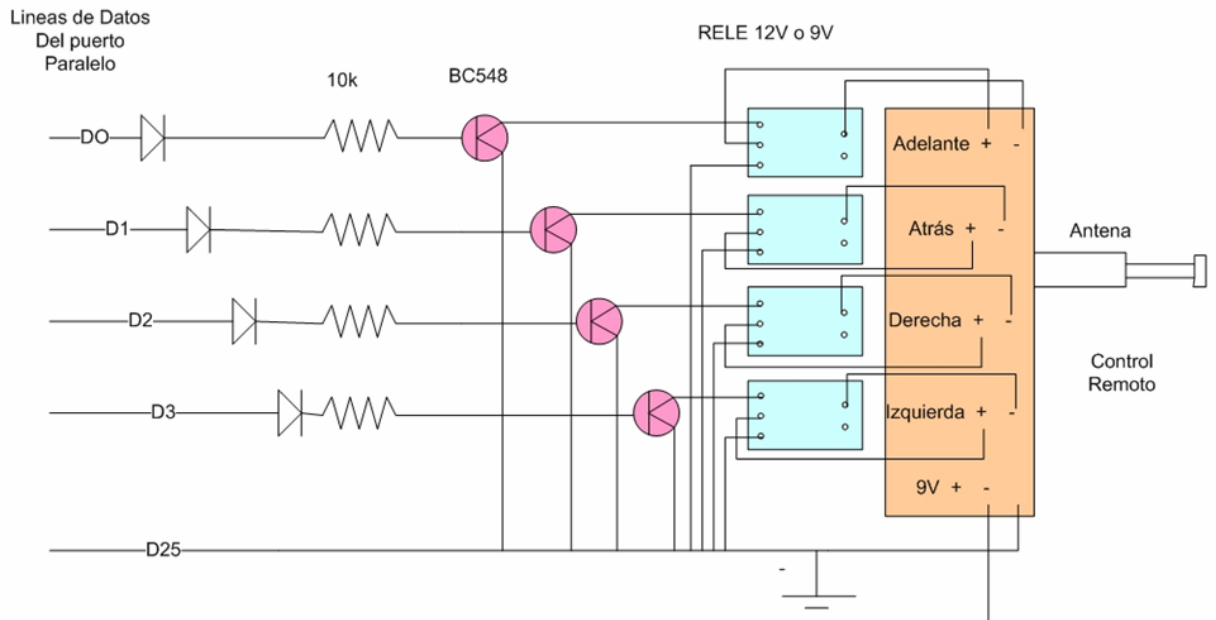
01 circuito de control de un carrito juguete a control remoto

01 interface controladora de RF de mando para el anterior

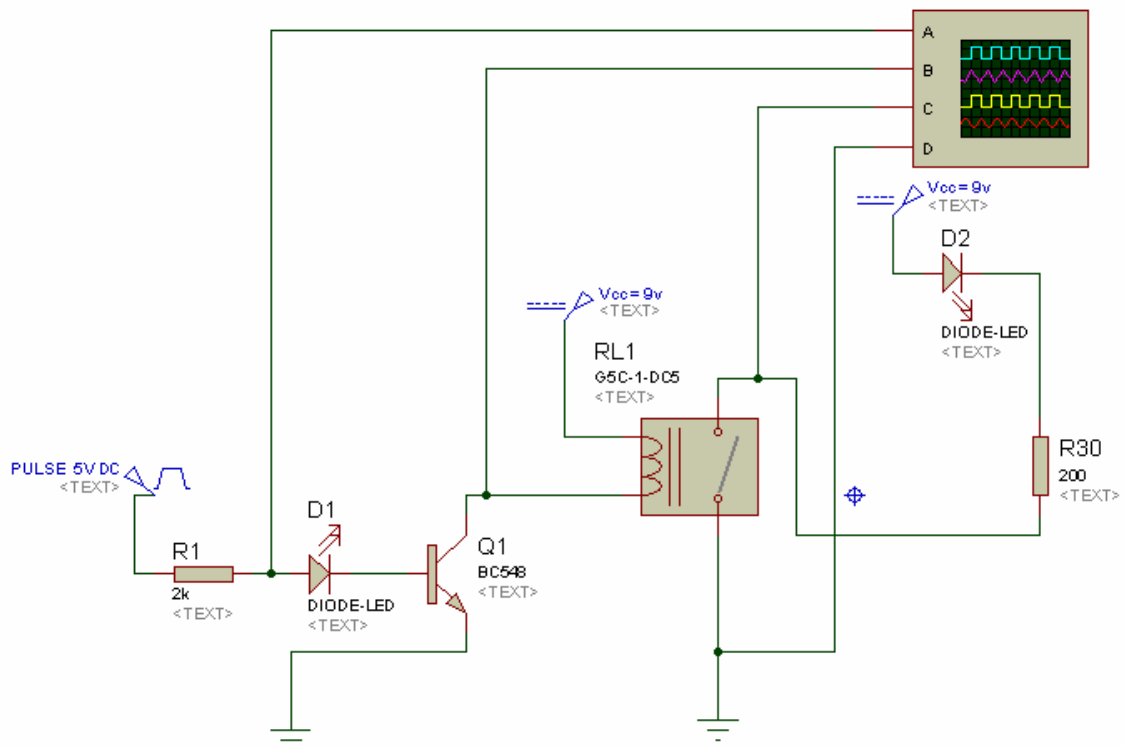
Cables de conexión

## EXPERIENCIA

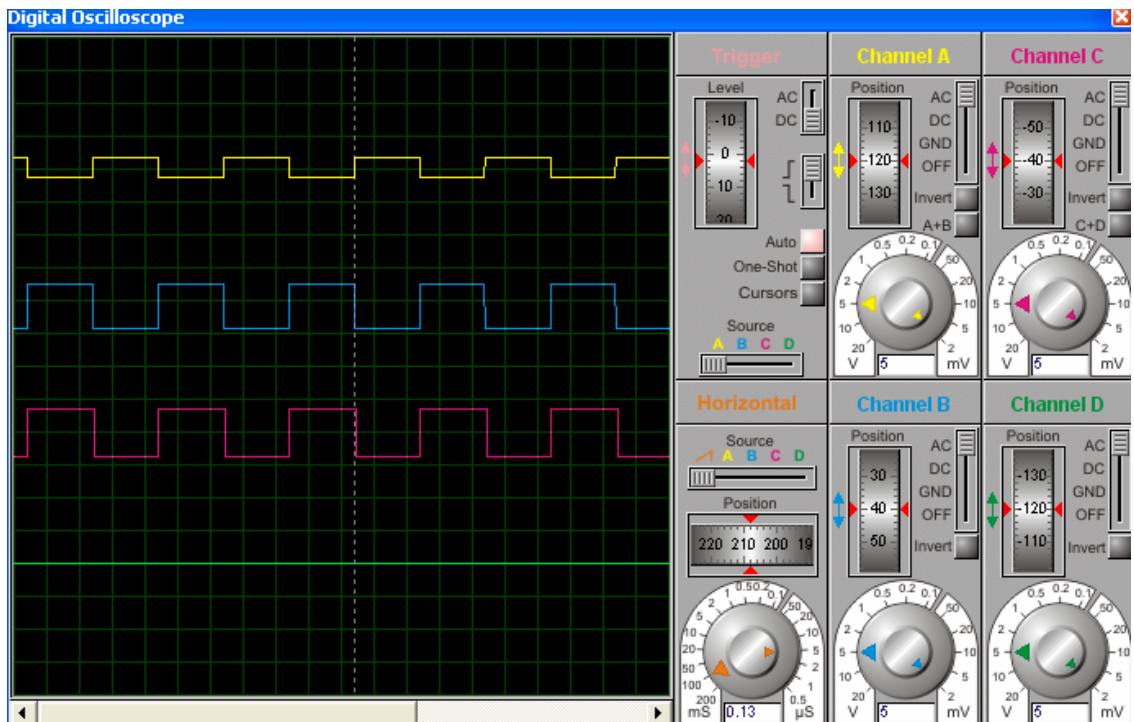
- En un principio nuestro circuito base salida de la interface –puerto paralelo– fue el siguiente:



- Se ubico los dispositivos y puntos tal como indica el diseño anterior en el protoboard, y posteriormente se implemento en un circuito impreso, diseñado o universal, pero se hizo cambios con esta consideración por cada transistor:



- Asimismo y virtualmente, se observo que la salida de los bits en el puerto paralelo, y en la interface, mediante el programa Proteus, salia de sta manera, por lo que se procedio a confirmar y conformar el circuito inicial :



## **EXPLICACION**

Explicando el esquema se observa que:

- Se alimento siempre con un transformador (de 9V y 3VA) para después de rectificad y filtrada se estabiliza en 5V con el regulador 7805. Para el caso hemos usado una bateria de 9v que nos fue mas facil adquirir y por la moviliad el control la hace mas practica .
- Posteriormente al ir desarrollando el esquema circuital, se observa que la corriente de mandos en el primario de cada relay es el punto basico para el control de los movimientos.
- Antes que nada se inidca que para el mismo hemos desacoplado y adaptado el control remoto de carrito con mando a distancia, en el cual el mando como se observa, tiene 4 contactores o switches al cual luego adaptamos y ahora van acoplados mediante los relays. Estos relays se activan solo si se le llega señal a la base del computador.
- En el computador usamos el puerto paralelo de 8 bits, los cuales solo trabajan con ceros y unos, es decir genera señales 0 y 5volt. Asi indicamos que se han utilizado solo 4 codigos (decimales) para jugar con los 4 movimientos del vehiculo. Se han usado 4 bits y por tanto 4 transistores y relays para su control.

- Por otra parte hemos dispuesto leds de control para saber la activacion y activar la base de los transistores que son del tipo emisor a tierra y tipo npn.
- El cuadro de salidas

D6	D5	D4	D3	D2	D1	D0	TECLA
				R			72 - UP
			R				80- DW
						R	75- IZQ
					R		62 - DER
			ABAJO	ARRIBA	DEREC	IZQUIER	

## **OBSERVACIONES**

1. El primer circuito se tuvo que probar primeramente en un protoboard para verificar la funcionalidad del esquema propuesto. Aquí es donde se opto mejor por utilizar mejor una bateria de 9 voltios a usar en cambio un transformador ya que se requiere por la portabilidad. Para los casos reales si sera necesario utilizar un transfo por que sera para periodos prolongados.
2. Por otra parte cuando se adquirieron los dispositivos se compraron en cantidad precisa pero cuando se malograron por razones diversas se tuvo que reemplazar con otros dispositivos mas o menos aproximados. En la mayoria de casos hay algunos valores que no son criticos y que pueden funcionar pero en otros como los relays o circuitos integrados se tiene que optar por los diseñados y no colocar similares. La funcionalidad debe ser la misma.
3. Las modificaciones se tuvieron que hacer en base a algunos diseños que hemos adquirido de informacion bajada en internet. Es decir si no habia diseños completos y muchos menos hasta reservados, hemos tenido que adaptar e improvisar con los materiales que teniamos a disposicion y economizando costos
4. En este proyecto se esta optando por usar un conjunto de leds que de lectura tipo barra para observar el control y activacion. Donde el encendido de leds indica uno digital o 5voltios, y que el relay esta activo y por tanto producir conductividad o cierre de circuito para el control de emando y movimiento del carro. Asi se pudo tambien controlar el movimiento solo con 4 bits o leds, como aquí, pero no olvidemos que a mas leds o relays dispuestos daria mas presicion por interpolacion.
5. Para mayor proteccion del puerto paralelo se podrian haber usado optoacopladores. Aquí usamos solo 4 relays con leds para el mejor monitoreo de las salidas. Mejor aun podria haberse realizado mediante

variables continuas como potenciometros pero el diseño variaria y resultaria mas complicado y costosa.

## **COSTOS**

El costo de este proyecto esta estimado según lo siguiente:

- Materiales	S/ 100.00
- carrito juguete	30.00
- bateria, pilas, etc	10.00
- dispositivos, reles, cableria	30.00
- Servicio Diseño/Instalacion	70.00
- Proforma comercial	US\$ 100.00

## **CONCLUSIONES**

1. Con esta experiencia hemos podido contrastar y **comprobar** todos los conceptos aprendidos en clase y que hemos realizado aquí, en el caso del transistor se verifico su estado de trabajo en solo corte y saturación, asi mismo en el caso de uso de los relays, de ondas de radiofrecuencia, para el manejo por control a distancia
2. De haber habido mas tiempo se hubiese para podido tambien contrastar conceptos y cambios con el software Proteus y mejorar asi el proyecto pero la implementacion nomas demoro en vista que este trabajo se hizo de manera no regular.
3. El sistema hace posible operar al carro mediante la utilización de las flechas del teclado de una computadora, previsto asi por su simplicidad y costo.
4. El sistema implementado aquí es de control por teclado y radiofrecuencia es de manera no automaica sino manual.
5. Se deja abierta la posibilidad de implementar aquí un control de movimiento de tipo automatizado o automatico, es decir, se ` puede adicionar un modulo al lenguaje tal que por teclado se ingrese y almacene previamente un ruteo previo de movil, tal que se pre establezca una trayecorial. Por razones de tiempo y costos, aquí el manejo es solo manual.
6. Asi mismo se deja la opcion de adicionarle hardware y via lenguaje de programación mayores características como son los sensores de deteccion de objetos, y por que no algunos conceptos de microprocesadores y

robotica. Por lenguaje se daría margen a que todo el sistema sea capaz de habilitar una memoria mayor para el movimiento de ruta y de ser capaz de guardar operaciones de tiempo superiores a la hora de trayectoria, limitados nada más por la capacidad de memoria, costos y tiempo.

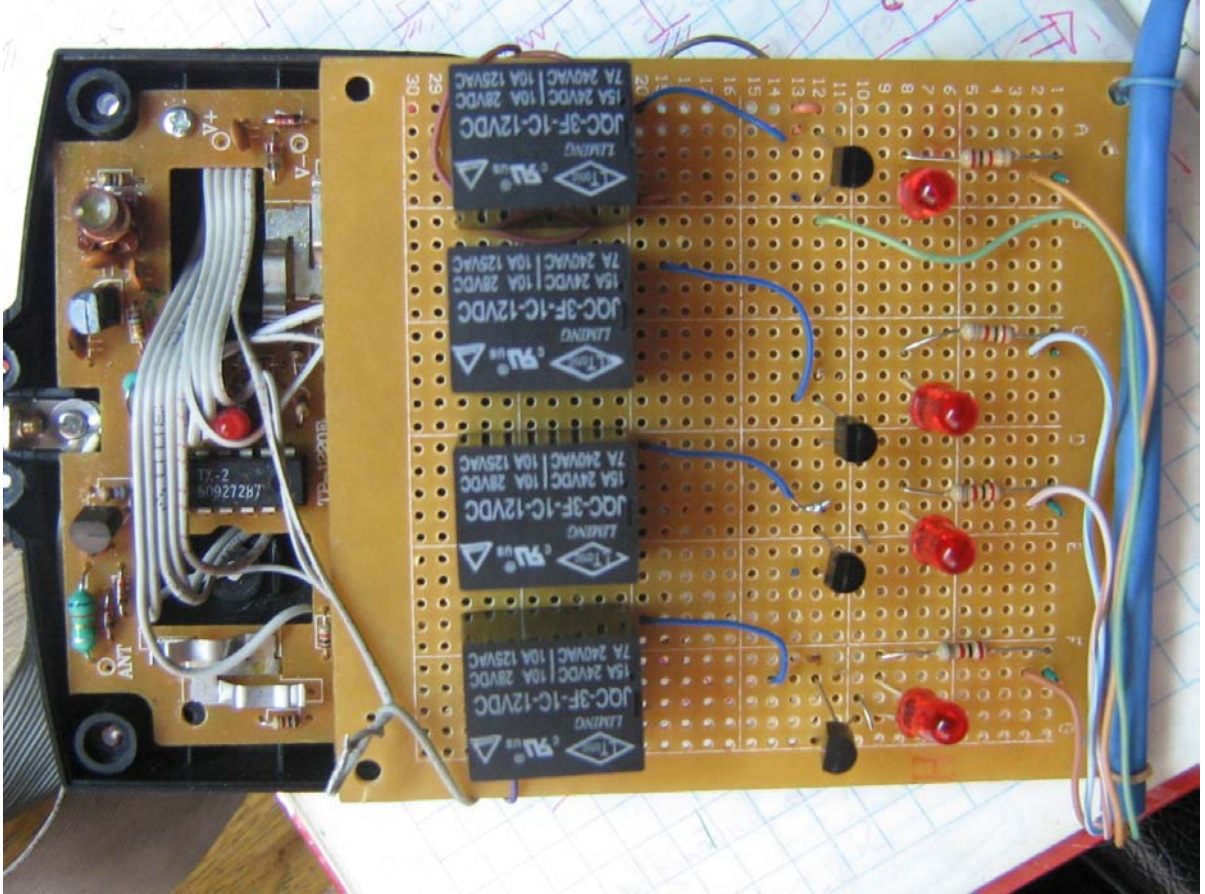
## **BIBLIOGRAFIA & WEBS**

- [www.forosdeelectronica.com/about1275.html](http://www.forosdeelectronica.com/about1275.html) - 54k -
  - [www.electronica2000.com/colaboraciones/colabora4.htm](http://www.electronica2000.com/colaboraciones/colabora4.htm)
  - BRAGA, Newton Cómo usar el Osciloscopio. En *Saber Electrónica*, Vol3, Nº 11, 1991.
  - *Enciclopedia Microsoft® Encarta® 98* © 1993-1997 Microsoft Corporation.
  - TEKTRONIX, En "*Electrónica fácil*" Nº 10, Junio de 1990. Medellín: Divulgación Técnica Electrónica.
  - [http://es.wikipedia.org/wiki/circuito\\_rc](http://es.wikipedia.org/wiki/circuito_rc)
- Apuntes de clase.



Moises Flores A.  
[www.geocities.com/mfloresa](http://www.geocities.com/mfloresa)

# FOTOGRAFÍAS





# **ANEXOS**

## CODIGO LENGUAJE C++

/\* ....

Programa hace movimiento vehicular mediante radiofrecuencia y teclado de pc.

Para WIN98/XP+TURBO C++3.0. En XP:

ejecutar antes: driverspuerto y parmon.

Efectos visualizables en interface hardware,

modo DOS a pantalla completa.

\*/

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
#include <dos.h>
```

```
int i,tecla,resultado,x,y,z;
```

```
int control[8];
```

```
void uno();
```

```
void cero();
```

```
void vista();
```

```
void reset();
```

### **void main ()**

```
{
```

```
reset();
```

```
tecla=0;
```

```
resultado=0;
```

```
x=0;
```

```
y=0;
```

```
z=0;
```

```
clrscr(); // IMPRIME MENU
```

```
printf("\nUTP-PERU / LP1 / 2008-1 ");
```

```
printf("\n");
```

```
printf("\nPROGRAMA PARA MOVER VEHICULO
```

```
POR RF Y PC ");
```

```
printf("\n");
```

```
printf("\n-MOISES FLORES ALCANTARA
```

```
www.geocities.com/mfloresa");
```

```
printf("\n-JAIR CORONEL CANTARO \n");
```

```
printf("\n");
```

```
printf("***** MENU *****
```

```
***** \n");
```

```
printf("\nPULSE SOLO TECLAS DE FLECHAS
```

```
PARA MOVIMIENTO:");
```

```
printf("\n FLECHAS O FN : NOT");
```

```
printf("\n F9 : CONTADOR EN
```

```
INTERFACE");
```

```
printf("\n F10 : SALIR");
```

```
outportb(0x378,0x00);
```

```
inicio:
```

```
vista();
```

```
do{
```

```
printf("\n PRESIONE UNA OPCION");
```

```
}while(!kbhit);
```

```
tecla=getch();
```

```
switch(tecla){
```

```
case 72: // codigo tecla: F1 = 59, .  
72 = flecha up , DO
```

```
z=0;
```

```
if (control[0]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 80: // F2 = 60, 80 = flecha  
down , D1
```

```
z=1;
```

```
if (control[1]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 75: // F3 = 61 , 75= flecha  
izq, D2
```

```
z=2;
```

```
if (control[2]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 62: // F4 = 62, 77 flecha  
der , D3
```

```
z=3;
```

```
if (control[3]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 77: // F5 = 62, 77 flecha  
der , D4
```

```
z=4;
```

```
if (control[4]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 64: // F6 → D5
```

```
z=5;
```

```
if (control[5]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 65: // F7 → D6
```

```
z=6;
```

```
if (control[6]==0) cero();
```

```
else uno();
```

```
break;
```

```
case 66: // F8 → D7
```

```
z=7;
```

```

    if (control[7]==0) cero();
    else uno();
    break;
case 67: // F9 → sin uso
    resultado=inportb(0x378);
    resultado++;
    outportb(0x378,resultado);
    break;
case 68: // F10 = fin
    goto fin;
case 69:
    reset();
    break;
}
goto inicio;
fin:
clrscr();
}

```

#### **void cero(){**

```

x=inportb(0x378);
control[z]=1;
switch(z){
    case 0:
        y=1;
        resultado=x|y;
        outportb(0x378,resultado);
        break;
    case 1:
        y=2;
        resultado=x|y;
        outportb(0x378,resultado);
        break;
    case 2:
        y=4;
        resultado=x|y;
        outportb(0x378,resultado);
        break;
    case 3:
        y=8;
        resultado=x|y;
        outportb(0x378,resultado);
        break;
    case 4:
        y=16;
        resultado=x|y;
        outportb(0x378,resultado);
        break;
    case 5:
        y=32;

```

```

        resultado=x|y;
        outportb(0x378,resultado);
        break;
case 6 :
    y=64;
    resultado=x|y;
    outportb(0x378,resultado);
    break;
case 7:
    y=128;
    resultado=x|y;
    outportb(0x378,resultado);
    break;
}
}

```

#### **void uno(){**

```

x=inportb(0x378);
control[z]=0;
switch(z){
    case 0:
        y=254;
        resultado=x&y;
        outportb(0x378,resultado);
        break;
    case 1:
        y=253;
        resultado=x&y;
        outportb(0x378,resultado);
        break;
    case 2:
        y=251;
        resultado=x&y;
        outportb(0x378,resultado);
        break;
    case 3:
        y=247;
        resultado=x&y;
        outportb(0x378,resultado);
        break;
    case 4:
        y=239;
        resultado=x&y;
        outportb(0x378,resultado);
        break;
    case 5:
        y=223;
        resultado=x&y;
        outportb(0x378,resultado);
        break;

```

```

case 6:
    y=191;
    resultado=x&y;
    outportb(0x378,resultado);
    break;
case 7:
    y=127;
    resultado=x&y;
    outportb(0x378,resultado);
    break;
}

vista();
}

void vista() {
    gotoxy(28,15);
    printf("         ade atr izq f4  der f6  f7  f8");
    gotoxy(28,16);
    printf("         D0 D1 D2 D3 D4 D5 D6
D7");
    gotoxy(28,17);
    printf("DOS parmor : ");
    for(i=0;i<=7;i++){
        printf(" %d ",control[i]);
    }
}

void reset()
{
    for(i=0;i<=7;i++)
        control[i]=0;
}

```

#### OBSERVACIONES:

La función **kbhit()** lo que hace es que devuelve 1 si el usuario ha pulsado una tecla y 0 en caso contrario. Cuando el usuario pulsa una tecla, el valor pulsado se guarda en un buffer que tiene el teclado, la función kbhit lo que hace es ver si ese buffer está vacío entonces devuelve 0 porque no hay tecla pulsada y 1 si el buffer contiene datos.

Ahora bien para poder rescatar los datos del buffer de teclado se tiene que utilizar la función **getch** o **getche** que interrumpe el programa, esto es así y no lo puedes hacer sin interrumpir, pero si utilizas bien la función getch y kbhit lo que puedes conseguir es que esa interrupción siempre tenga éxito, es decir que cuando se ejecute la interrupción siempre halla datos en el buffer y la función getch coga los datos, esto dura un instante de tiempo muy pequeño con lo que al usuario le da la impresión de que el programa no se ha interrumpido. O

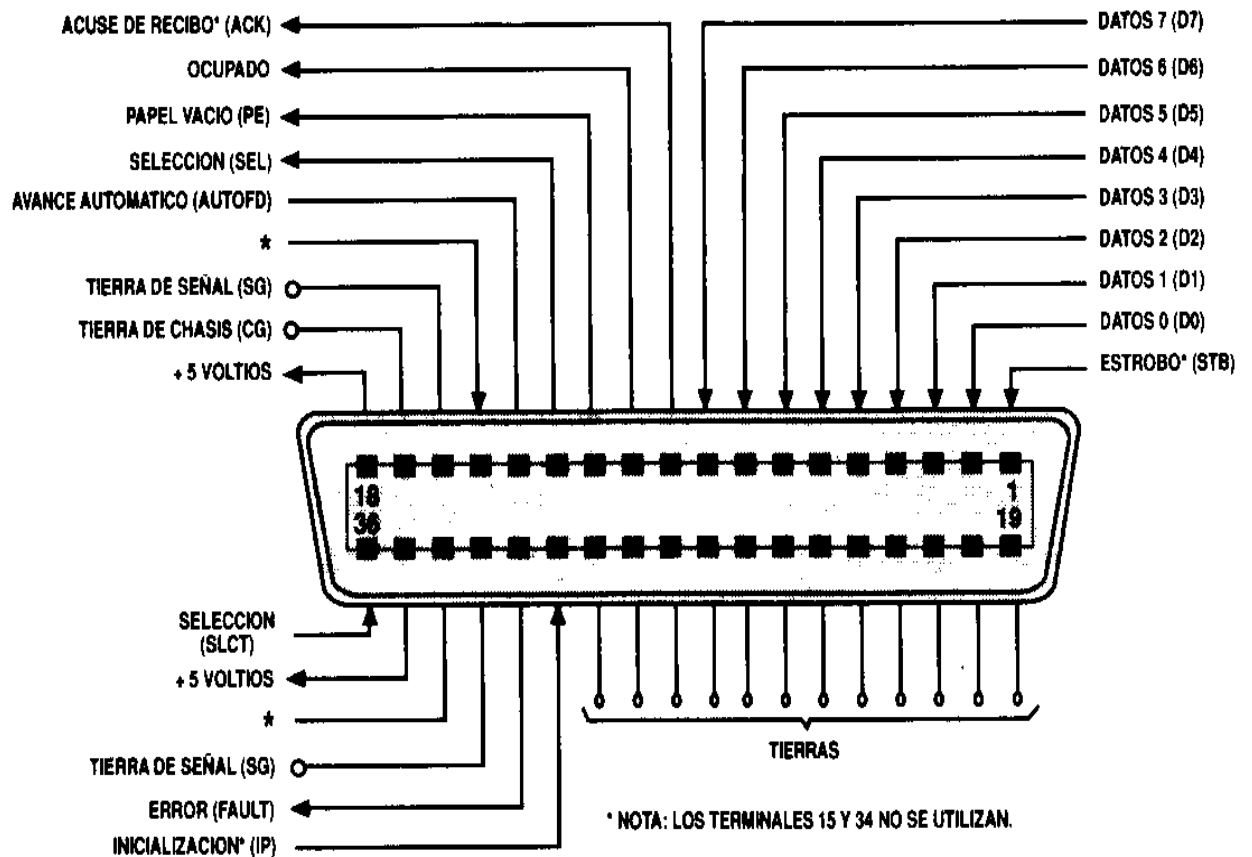
## CONCEPTOS PUERTO PARALELO

### *El extremo del periférico*

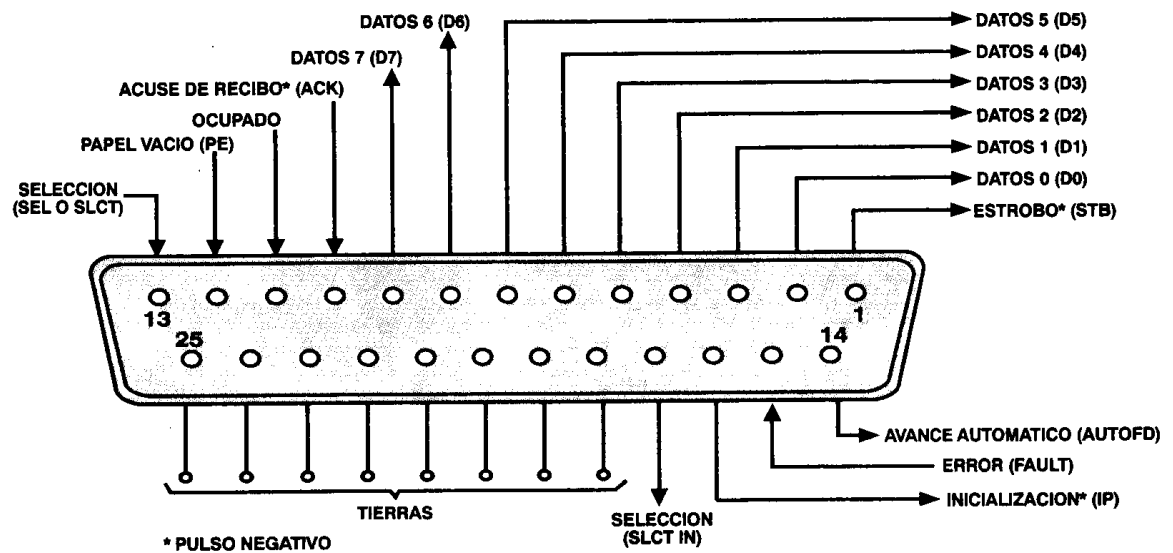
Si bien hemos tratado casi todas las señales que es probable que Ud. encuentre, hay aun algunas cosas que debe saber acerca de lo que hallara en el extremo del periférico del cable.

La informacion adicional le ayudara a construir sus propios cables para periféricos ya construidos.

Un conector hembra de 36 conductores es la terminacion mas adecuada que se encuentra en los periféricos paralelos. Las funciones convencionales de cada terminal de ese conector se muestran en la fig. 3.

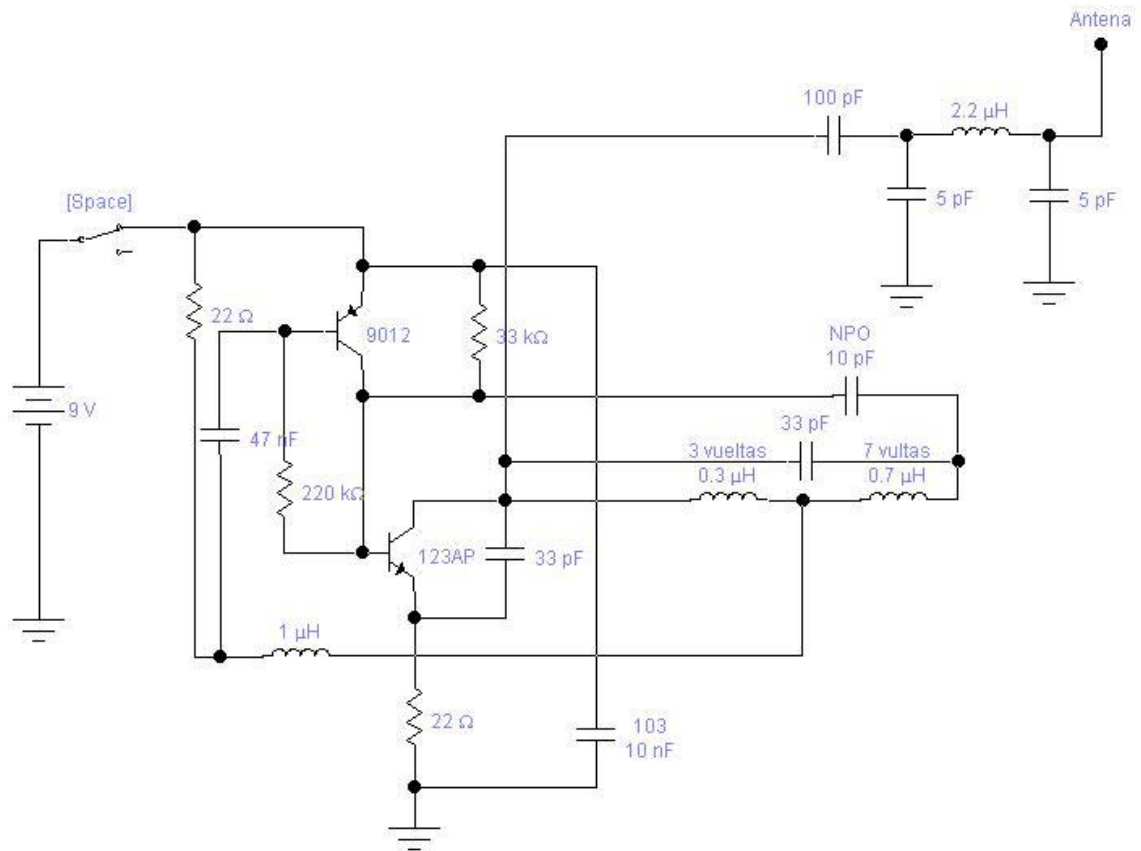


**FIGURA 3:** Esta es la asignación de terminales de un periférico compatible con Centronics. Las funciones son las mismas que las del conector DB-25 ilustrado en la fig. 2, pero hay unas cuantas tierras más y dos terminales de alimentación de 5 voltios.

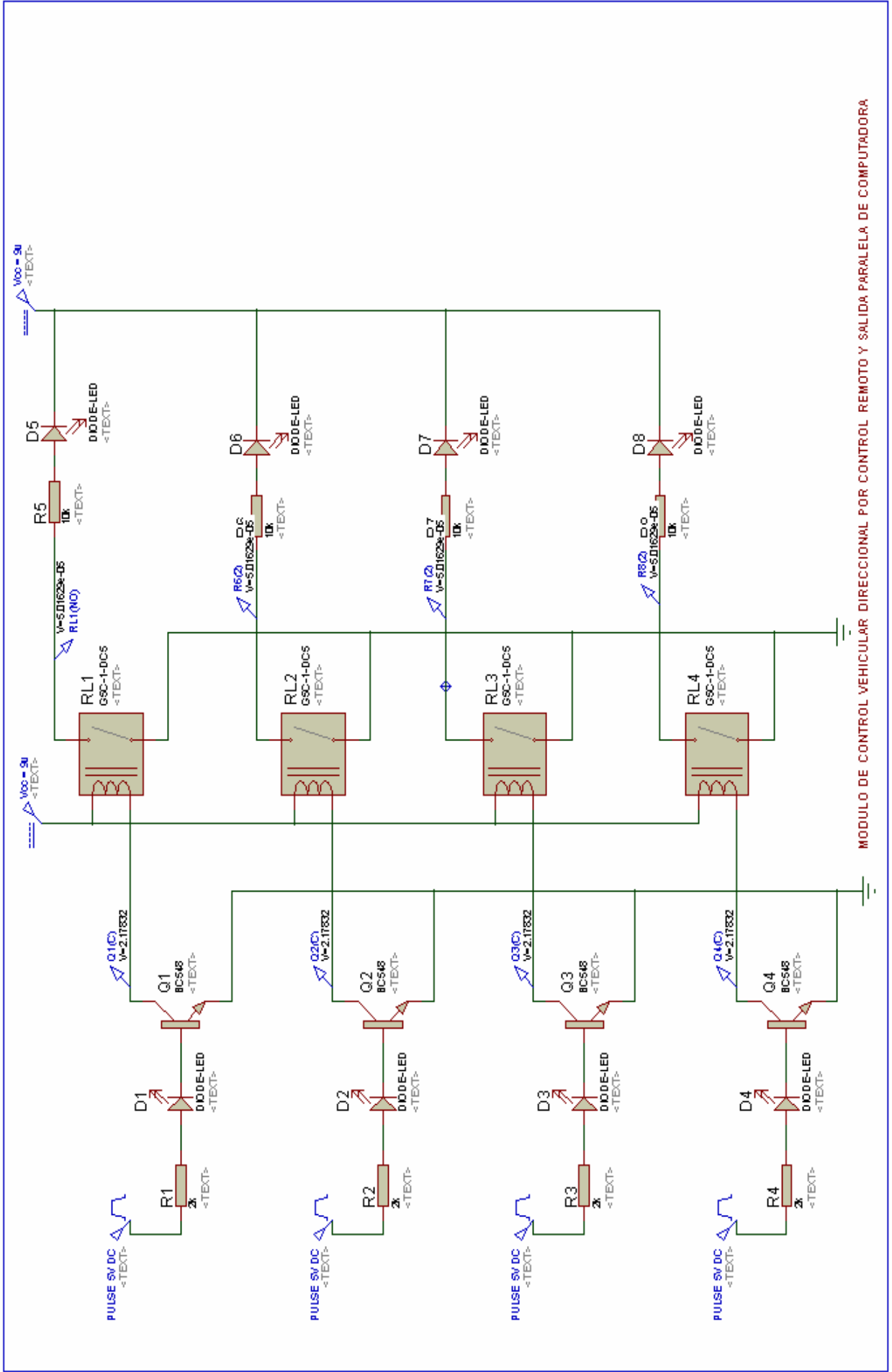


**FIGURA 1:** En la mayoría de las computadoras IBM compatibles (y algunas no compatibles) el puerto paralelo es como éste. Las flechas que apuntan hacia afuera del conector DB-25 son salidas, las que apuntan hacia adentro, entradas. Los terminales restantes, señalados con círculos, son tierras.

## TRANSMISOR







MODULO DE CONTROL VEHICULAR DIRECCIONAL POR CONTROL REMOTO Y SALIDA PARALELA DE COMPUTADORA