

PRACTICA No 6

Objetivo general:

En esta práctica aprenderemos a usar el puerto paralelo realizando un circuito donde utilizaremos señales de salida del mismo además de aprender a realizar el cable que servirá de interfaz entre el puerto paralelo y nuestro protoboard.

Material y equipo:

- 2m de cable plano de 25 hilos o mas
- Conectores DB-25 hembra y macho
- 8 leds de luz visible
- 16 resistencias de 1K
- Pinzas de punta y de corte
- Multímetro
- Protoboard

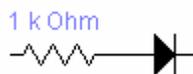
Desarrollo:

En primer lugar realizamos la construcción de nuestro cable que conectaría el puerto paralelo de la computadora con la protoboard.

La conexión se llevo mediante el uso del cable plano y los conectores DB-25 de presión tomando en cuenta que el cable plano que conseguimos era de 26 hilos procedimos a quitarle un hilo para que quedaran 25. Lo siguiente fue enlazar uno de los 2 DB-25 tomando en cuenta que el pin 1 del DB-25 debería corresponder con el hilo pintado de rojo y después conectamos el otro DB-25 siguiendo la misma instrucción.

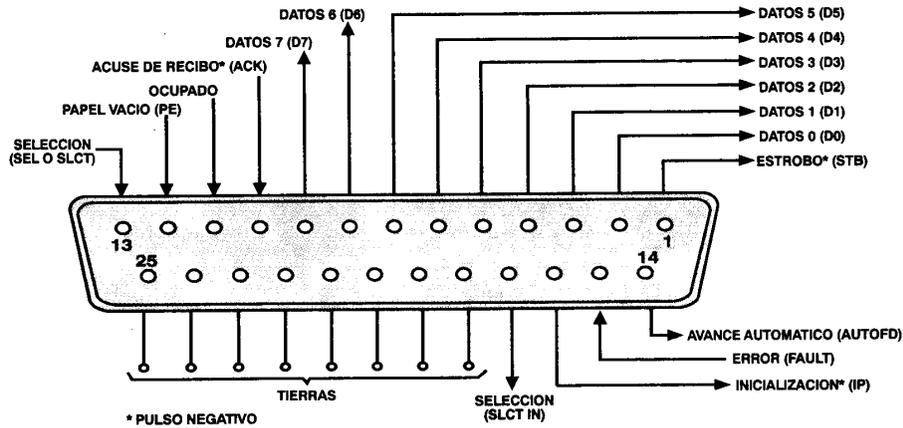
Ya aprisionado el cable plano dentro de los DB-25 procedimos a medir la continuidad entre cada uno de los pines DB-25 correspondientes así el pin1 del DB-25 macho debería tener continuidad con el DB-25 hebra en su pin 1 y solamente en ese y así sucesivamente con todos los demás pines. Al resultar que todos los pines tenían continuidad con los pines correctos podíamos decir que el cable estaba correcto.

Luego de esto procedimos a realizar el circuito de salida al que íbamos a conectar el puerto paralelo para esto seguimos el siguiente diagrama:



Siendo que la resistencia por el momento quedaría suelta y el ánodo de nuestro led iría al la línea azul reconociendo que este sería el negativo de nuestro circuito electrónico (el negativo del led se puede reconocer por que es la parte chata o es la banderilla mas grande). Siendo así pusimos 8 resistencias y ocho leds dentro de nuestro protoboard además de unir las líneas azules y las rojas mediante cables para que tuvieran continuidad entre si.

Después de haber hecho lo anterior aprendimos cuales son las terminales que maneja el puerto paralelo las cuales se ilustran en el siguiente diagrama:

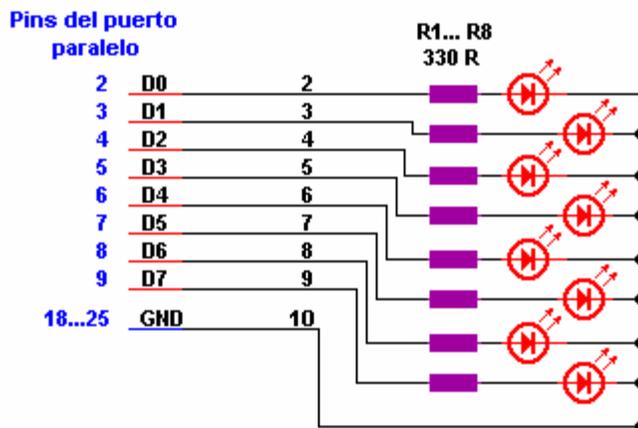


Siendo que del pin 2 al 9 podemos obtener datos el pin 2 es el menos significativo y el 9 el más significativo.

Del pin 18 al 25 tenemos que son tierras o negativos por lo que nuestra línea azul del protoboard ira conectada a todos o alguno de estos pines.

Ya que sabemos como esta configurado nuestro puerto paralelo debemos insertar cables que vayan del cable plano a nuestro protoboard para esto es muy útil un header o conector de cable plano a protoboard.

Después de esto realizamos la siguiente conexión:



Ya teniendo esto listo procedemos a la realización de los programas:

El primer programa que realizamos realiza una serie del 255 al 0 en forma descendente de 1 en 1 el cual va desplegando los resultados tanto en el puerto paralelo como en la pantalla:

El código fuente del programa quedo de la siguiente forma:

/*

FECHA: 31-NOV-06

PROGRAMA: 15

OBJETIVO: /*MEDIANTE EL USO DEL CICLO FOR REALIZAR

UNA SERIE DEL 255 AL 0 DESCENDENTE Y
QUE SE MUESTRE POR EL PUERTO PARALELO*/

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
main ()
{
int x,y,l,t,u=5,g=5,h=200;
clrscr ();
gotoxy (5,5);
printf ("Presiona una tecla para continuar");
gotoxy (5,7);
printf ("Serie\n");
textmode (C4350);
textcolor (RED);
textbackground (CYAN);
clrscr ();
for (l=1;l<=80;l++)
{
gotoxy (l,1);
printf ("*");
delay (g);
}
for (l=1;l<=49;l++)
{
gotoxy(80,l);
printf ("*");
delay (g);
}
for (l=80; l>=1;--l)
{
gotoxy (l,49);
printf ("*");
delay (g);
}
for (l=49;l>=1;--l)
{
gotoxy (1,l);
printf ("*");
delay (g);
}
printf ("\n\n");
```

```
for (x=255; x>=0;x--)
{
t=u*45;
if (x!=t || x==0)
{
outport (0x378,x);
printf ("*\tx es:%d\n", x);
```

```

    delay (h);
}
else
{
    u--;
    printf ("\n*\tPresiona una tecla para continuar\n");
    getch ();
    clrscr ();
    for (l=1;l<=80;l++)
    {
        gotoxy (l,1);
        printf ("*");
        delay (g);
    }
    for (l=1;l<=49;l++)
    {
        gotoxy(80,l);
        printf ("*");
        delay (g);
    }
    for (l=80; l>=1;--l)
    {
        gotoxy (l,49);
        printf ("*");
        delay (g);
    }
    for (l=49;l>=1;--l)
    {
        gotoxy (1,l);
        printf ("*");
        delay (g);
    }
    outport (0x378,x);
    printf ("\n*\tx es:%d\n", x);
    delay (20);
}
}
getch ();
}
/*CONCLUSIONES: EN ESTE PROGRAMA COMENZAMOS
A VERIFICAR COMO FUNCIONA EL PUERTO PARALELO
Y QUE EL COMANDO NECESARIO PARA ENVIAR SEÑALES
ES EL OUTPORT*/

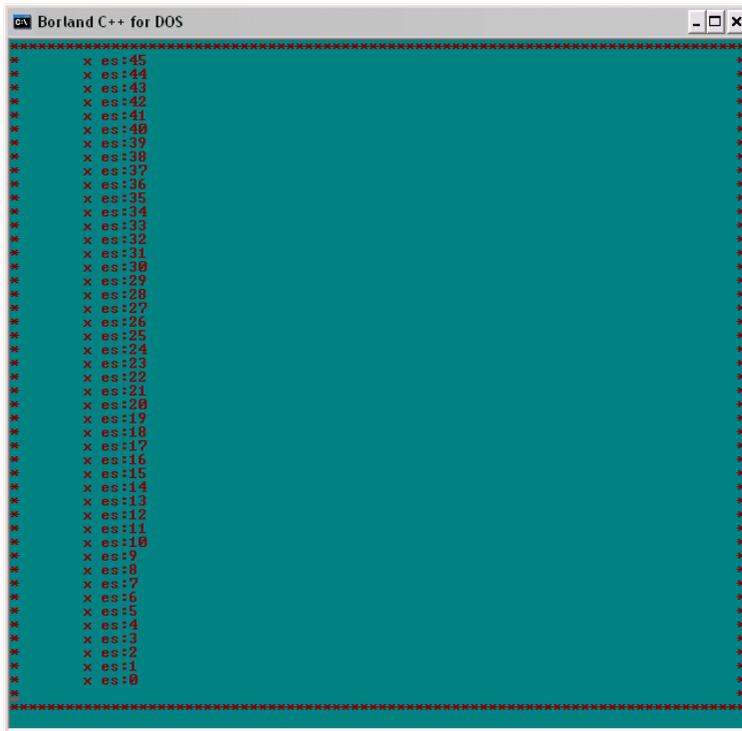
```

La corrida se vería de la siguiente forma (Se imprimieron solo tres pantallas de las 5:

```
Borland C++ for DOS
*****
x es:255
x es:254
x es:253
x es:252
x es:251
x es:250
x es:249
x es:248
x es:247
x es:246
x es:245
x es:244
x es:243
x es:242
x es:241
x es:240
x es:239
x es:238
x es:237
x es:236
x es:235
x es:234
x es:233
x es:232
x es:231
x es:230
x es:229
x es:228
x es:227
x es:226

Presiona una tecla para continuar
*****
```

```
Borland C++ for DOS
*****
x es:135
x es:134
x es:133
x es:132
x es:131
x es:130
x es:129
x es:128
x es:127
x es:126
x es:125
x es:124
x es:123
x es:122
x es:121
x es:120
x es:119
x es:118
x es:117
x es:116
x es:115
x es:114
x es:113
x es:112
x es:111
x es:110
x es:109
x es:108
x es:107
x es:106
x es:105
x es:104
x es:103
x es:102
x es:101
x es:100
x es:99
x es:98
x es:97
x es:96
x es:95
x es:94
*****
```



```

x es:45
x es:44
x es:43
x es:42
x es:41
x es:40
x es:39
x es:38
x es:37
x es:36
x es:35
x es:34
x es:33
x es:32
x es:31
x es:30
x es:29
x es:28
x es:27
x es:26
x es:25
x es:24
x es:23
x es:22
x es:21
x es:20
x es:19
x es:18
x es:17
x es:16
x es:15
x es:14
x es:13
x es:12
x es:11
x es:10
x es:9
x es:8
x es:7
x es:6
x es:5
x es:4
x es:3
x es:2
x es:1
x es:0

```

En esta lo que sucedía es que los leds iban prendiendo conforme a la cuanta en pantalla además de que cada vez quedaban encendidos menos leds de los mas significativos ya que la cuanta era regresiva pero el led 1 seguía haciendo su secuencia normal. Cabe destacar que todos los resultados fueron correctos en el código binario.

El segundo programa debe realizar una seria del 0 al 127 creciente de uno en uno por lo que usamos nuevamente el método anterior poniendo como comando de salida el outport (0x378, variable), en este caso la variable es una x que conforma un contador de un ciclo for.

El código fuente es el que sigue:

```
/* FECHA: 27-SEP-06
```

```
PROGRAMA: 16
```

```
OBJETIVO: /*REALIZAR UNA SERIE QUE DESPLIEGUE TANTO
EN PANTALLA COMO EN LEDS POR EL PUERTO PARALELO
UNA SERIE DEL 0 AL 128 DE FORMA ASCENDENTE CON EL
COMANDO OUTPORT */
```

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

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

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

```
main ()
```

```
{
```

```
int x,y,l,t,u=1,g=5,h=200;
```

```
clrscr ();
```

```
gotoxy (5,5);
```

```
printf ("Presiona una tecla para continuar");
```

```
gotoxy (5,7);
```

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

```
textmode (C4350);
```

```
textcolor (RED);
```

```

textbackground (CYAN);
clrscr ();
for (l=1;l<=80;l++)
{
gotoxy (l,1);
printf ("*");
delay (g);
}
for (l=1;l<=49;l++)
{
gotoxy(80,l);
printf ("*");
delay (g);
}
for (l=80; l>=1;--l)
{
gotoxy (l,49);
printf ("*");
delay (g);
}
for (l=49;l>=1;--l)
{
gotoxy (1,l);
printf ("*");
delay (g);
}
printf ("\n\n");
for (x=0; x<128;x++)
{
t=u*40;
if (x!=t)
{
outport (0x378,x);
printf ("*\tx es:%d\n", x);
delay (h);
}
}
else
{
u++;
printf ("\n*\tPresiona una tecla para continuar\n");
getch ();
clrscr ();
for (l=1;l<=80;l++)
{
gotoxy (l,1);
printf ("*");
delay (g);
}
for (l=1;l<=49;l++)
{
gotoxy(80,l);

```

```

printf ("*");
delay (g);
}
for (l=80;l>=1;--l)
{
gotoxy (l,49);
printf ("*");
delay (g);
}
for (l=49;l>=1;--l)
{
gotoxy (1,l);
printf ("*");
delay (g);
}
outport (0x378,x);
printf ("\n\n*tx es:%d\n", x);
delay (20);
}
}
printf ("\n\n*\tFIN");
getch ();
}
/*CONCLUSIONES: MANEJAMOS EL COMANDO OUTPORT Y VEMOS
QUE LAS SERIES DESPLEGADAS EN LOS LEDS SON NUMEROS
REPRESENTADOS EN FORMA BINARIA*/

```

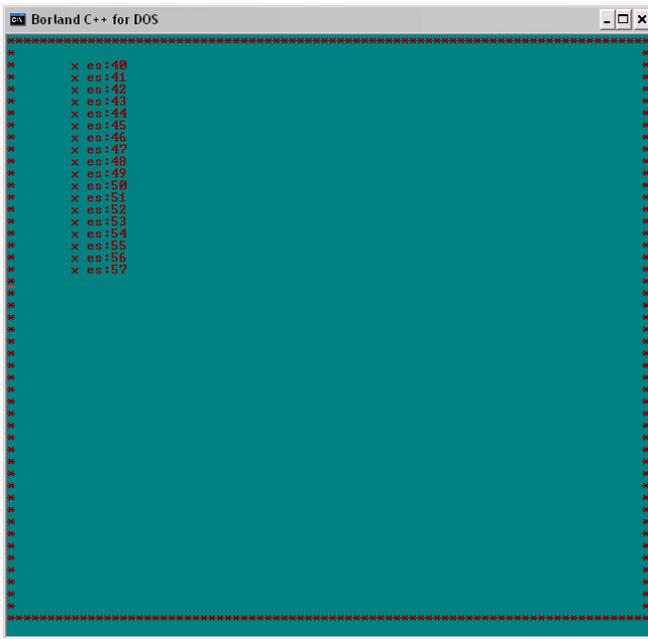
La corrida del programa se ve de la siguiente forma:

```

x es:0
x es:1
x es:2
x es:3
x es:4
x es:5
x es:6
x es:7
x es:8
x es:9
x es:10
x es:11
x es:12
x es:13
x es:14
x es:15
x es:16
x es:17
x es:18
x es:19
x es:20
x es:21
x es:22
x es:23
x es:24
x es:25
x es:26
x es:27
x es:28
x es:29
x es:30
x es:31
x es:32
x es:33
x es:34
x es:35
x es:36
x es:37
x es:38
x es:39

Presiona una tecla para continuar

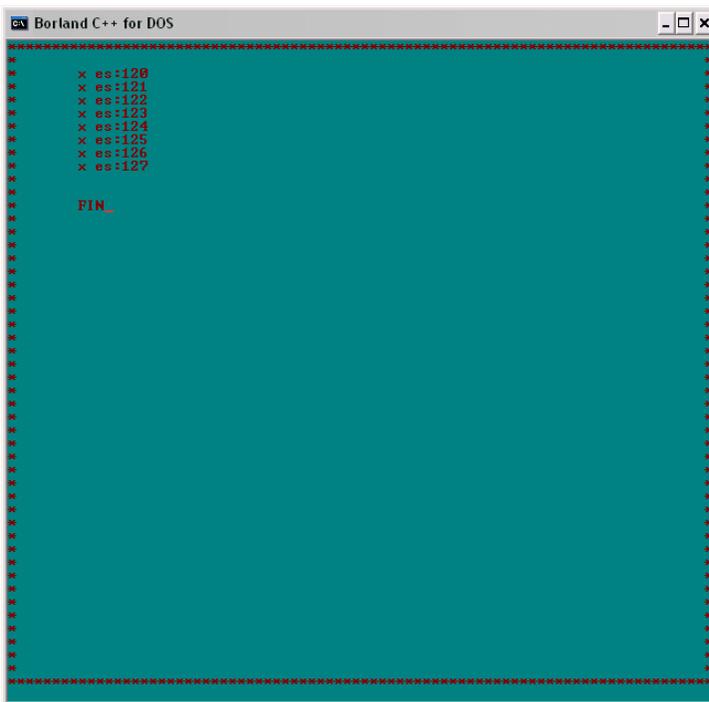
```



```

x es:40
x es:41
x es:42
x es:43
x es:44
x es:45
x es:46
x es:47
x es:48
x es:49
x es:50
x es:51
x es:52
x es:53
x es:54
x es:55
x es:56
x es:57

```



```

x es:120
x es:121
x es:122
x es:123
x es:124
x es:125
x es:126
x es:127

FIN_

```

Sucede lo mismo que en el caso anterior además como se pueden dar cuenta en la pantalla tres se puede ver que los valores van apareciendo poco a poco esto gracias a un delay que sirve para que nuestros datos se puedan notar mientras cambia de valores.

El tercer programa debe realizar una serie del 0 al 255 desplegando en los leds y la pantalla los múltiplos del numero dos o mejor dicho que la serie vaya ascendiendo de dos en dos. El código fuente del programa es el siguiente:

```

/*
FECHA: 27-SEP-06
PROGRAMA: 17
OBJETIVO: /*QUE EL ALUMNO COMPRENDA COMO
FUNCIONA EL PUERTO PARALELO REALIZANDO UNA
SERIE DEL 0 AL 255 CON NUMEROS PARES */
#include <stdio.h>
#include <conio.h>
#include <dos.h>
main ()
{
int x,y,l,t,u=1,g=5,h=200;
clrscr ();
gotoxy (5,5);
printf ("Presiona una tecla para continuar");
gotoxy (5,7);
printf ("Serie\n");
textmode (C4350);
textcolor (RED);
textbackground (CYAN);
clrscr ();
for (l=1;l<=80;l++)
{
gotoxy (l,1);
printf ("*");
delay (g);
}
for (l=1;l<=49;l++)
{
gotoxy(80,l);
printf ("*");
delay (g);
}
for (l=80; l>=1;--l)
{
gotoxy (l,49);
printf ("*");
delay (g);
}
for (l=49;l>=1;--l)
{
gotoxy (1,l);
printf ("*");
delay (g);
}
printf ("\n\n");
for (x=0; x<=255;x+=2)
{
t=u*80;
if (x!=t)
{

```

```

    outport (0x378,x);
    printf ("*\tx es:%d\n", x);
    delay (h);
}
else
{
    u++;
    printf ("\n*\tPresiona una tecla para continuar\n");
    getch ();
    clrscr ();
    for (l=1;l<=80;l++)
    {
        gotoxy (l,1);
        printf ("*");
        delay (g);
    }
    for (l=1;l<=49;l++)
    {
        gotoxy(80,l);
        printf ("*");
        delay (g);
    }
    for (l=80; l>=1;--l)
    {
        gotoxy (l,49);
        printf ("*");
        delay (g);
    }
    for (l=49;l>=1;--l)
    {
        gotoxy (1,l);
        printf ("*");
        delay (g);
    }
    outport (0x378,x);
    printf ("\n\n*\tx es:%d\n", x);
    delay (20);
}
}
printf ("\n\n*\tFIN");
getch ();
}
/*CONCLUSIONES: OBSERVAMOS QUE EL PUERTO
PARALELO ES CONGRUENTE CON NUESTRAS AFIRMACIONES
YA QUE ESTA VEZ AL SER TODOS LOS NUMEROS PARES NUNCA
SE ENCENDIO EL BIT MENOS SIGNIFICATIVO O UNO*/

```

La corrida se ve de la siguiente forma(de nuevo se imprimieron solo tres pantallas aunque aparecen todas en la corrida del programa):

```
Borland C++ for DOS
*****
x es:10
x es:12
x es:14
x es:16
x es:18
x es:10
x es:12
x es:14
x es:16
x es:18
x es:20
x es:22
x es:24
x es:26
x es:28
x es:30
x es:32
x es:34
x es:36
x es:38
x es:40
x es:42
x es:44
x es:46
x es:48
x es:50
x es:52
x es:54
x es:56
x es:58
x es:60
x es:62
x es:64
x es:66
x es:68
x es:70
x es:72
x es:74
x es:76
x es:78

Presiona una tecla para continuar
*****
```

```
Borland C++ for DOS
*****
x es:80
x es:82
x es:84
x es:86
x es:88
x es:90
x es:92
x es:94
x es:96
x es:98
x es:100
x es:102
x es:104
x es:106
x es:108
x es:110
x es:112
x es:114
x es:116
x es:118
x es:120
x es:122
x es:124
x es:126
x es:128
x es:130
*****
```

```
Borland C++ for DOS
*****
*
*      x es:240
*      x es:242
*      x es:244
*      x es:246
*      x es:248
*      x es:250
*      x es:252
*      x es:254
*
*      FIN
*
*****
```

Como se puede observar los tres programas siguieron la misma metodología un contador que va incrementando o decrementando hasta que su valor alcanza el deseado además de que los números se imprimen en pantalla y se sacan por el puerto paralelo solo cabe decir que para imprimir por pantallas se toma como referencia que el contador no rebase un valor cuando lo hace se toma como un caso especial y se borra la pantalla se pone nuevamente el margen y se incrementa un contador que multiplicado por un entero hará la función de comparar que la siguiente vez que el numero rebase un múltiplo del primero que se tomo en cuenta se repita el proceso de borrado e impresión del margen.

Conclusiones generales:

En esta práctica aprendimos a usar el puerto paralelo lo cual nos servirá mucho en la aplicación de este al control sistemas reales o fiscos aprendimos el uso del comando outport e inport dentro de la programación en lenguaje además de aprender a realizar el cable interfaz entre la computadora y el circuito electrónico.