Nombre de la asignatura: Laboratorio de Programación de Avanzada

Maestro en Ciencias: Moisés Agustín Martínez Hernández

Nombre de la práctica: Manejo de entradas y salidas digitales - Multiplexado

Integrantes:

- Zuñiga Fragoso Diego Joel
- Manríquez Navarro Daniela del Carmen

Número de práctica: 2

Total de horas: 2 Hrs.

Objetivos

El alumno podrá configurar un puerto del microcontrolador como señal de salida y usarlo para controlar dos displays de 7 segmentos al mismo tiempo, y de esta manera optimizar su uso. La comprensión de la teoría del multiplexor y el demultiplexor ayudará al alumno a conectar, seleccionar y controlar más de un componente con un solo puerto del microcontrolador, de este modo podrá simplificar el cableado y presentación de la práctica.

Descripción de la práctica

Mostrar un conteo ascendente 0 – 99 a través de dos Displays de 7 segmentos, este iniciará al presionar un botón “Inicio”. El conteo puede ser interrumpido en cualquier momento con un botón “Pausa” y continuar con el botón de “Inicio”, del mismo modo se podrá reiniciar el conteo desde 0 presionando un botón “Reinicio”. Se usará un solo puerto de salida para controlar ambos Displays. Para lograr esto se hará uso de un demultiplexor para alternar la selección entre un display y otro.

Marco teórico

El display numérico consiste en siete LEDs rectangulares los cuales están arreglados para formar un 8. Adicionalmente, la mayoría de los displays también tienen un punto, resultando en ocho LEDs los cuales están acomodados como se muestra en la figura 1. Los LEDs están etiquetados como a – g y dot point (dp).

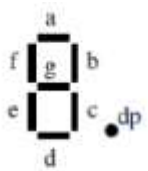


Figura 1. Los siete segmentos y dp de un display numérico.

Un diplay numérico tiene 8+1 pines: ocho pines están conectados a los ánodos o cátodos de todos los LEDs, el noveno pin es común para todos ánodos o cátodos, figura 2.

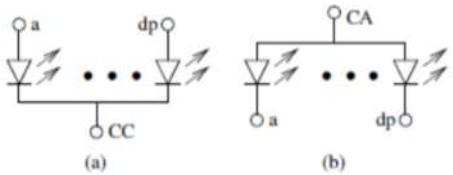
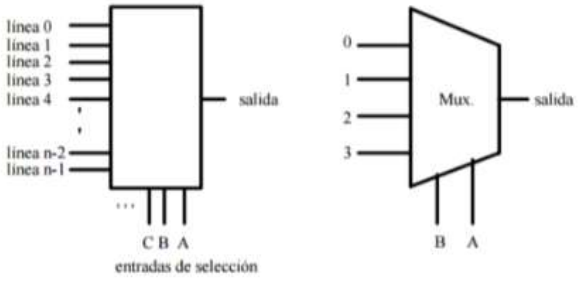


Figura 2. Display numérico con cátodo común (a) y ánodo común (b).

Para activar el display, el pin común debe conectarse a GND (cátodo común CC) o a VCC (ánodo común CA). Entonces, los LEDs pueden ser encendidos individualment conectando los pines correspondientes a un novel de voltaje apropiado. Al igual que los LEDs simples, un display numérico debe ser protegido contra sobrecorrientes.

La función multiplexado consiste en seleccionar una de entre varias fuentes de datos para enviarlos por una misma línea; un multiplexor realiza el multiplexado de **n** líneas de entrada a una sola línea de salida.

Un multiplexor es un subsistema digital que selecciona una de entre **n** fuentes de datos, comunicándola con la línea de salida del mismo; dispone de **n** líneas de entrada, **1** línea de salida y un conjunto de **m** líneas de control o selección, tal que $\mathbf{n = 2^m}$ (o bien, $\mathbf{2^m \geq n}$). En cada momento el vector de entrada en las líneas de selección determina (por su número) la línea de entrada que queda “conectada” a la salida. Este tipo de actuación se denomina también muestreo, pues selecciona de entre diversas líneas de entrada la que interesa en cada caso: el multiplexor “muestrea” la línea cuyo número binario es establecido sobre las entradas de control del mismo, como se muestra en la Figura 3.



Figuras 3. Representa, respectivamente, un multiplexor genérico de n líneas y un multiplexor de 4 líneas.

Los demultiplexores realizan una función contraria a la del multiplexor: reciben los datos por una sola entrada y los dirigen a una de entre **n** líneas de salida, seleccionables numéricamente por **m** líneas de control, Figura 4; en cada momento el dato presente en su entrada aparece en la salida cuyo número binario coincide con el establecido en las entradas de control y el resto de líneas de salida permanecen a 0.

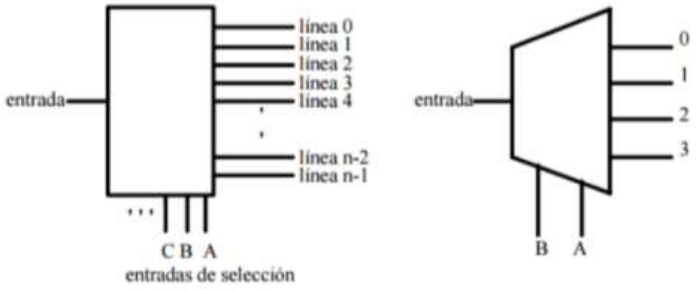


Figura 4. Representa, respectivamente, un demultiplexor genérico de n líneas y un multiplexor de 4 líneas.

EQUIPO Y MATERIALES

- Software para programación
- Software de simulación

DESARROLLO DE LA PRÁCTICA

1. Escribir el programa para el microcontrolador de la Figura 5, considerando:
   a) La selección de los pines como señal de entrada así como el puerto de salida quedará a consideración del alumno.
   b) El puerto configurado como salida, enviará la señal a ambos displays.
   c) Configurar dos pines del microcontrolador, los cuales, harán la función de selector de la salida, en este caso, el display que recibirá la señal del puerto.
2. Desarrollar la simulación del circuito (de acuerdo al diagrama de bloques, Figura 5), tomando en cuenta los siguiente:
   a) Se tendrán tres botones que servirán como selección de “Inicio”, “Pausa” y “Reinicio” del contador.
   b) El contador será mostrado con la ayuda de dos displays de siete segmentos, los cuales estarán conectados a un solo puerto del microcontrolador.
   c) El primer display mostrará decenas y el segundo las unidades del contador.
   d) La selección del display a controlar se hará a través de dos pines de salida del microcontrolador.
3. Realizar la conexión del circuito y programar el microcontrolador.

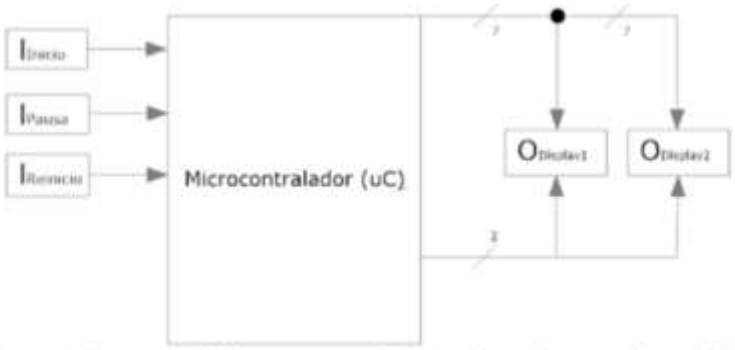


Figura 5. Diagrama de bloques para el circuito *Contador ascendente 0-99.*

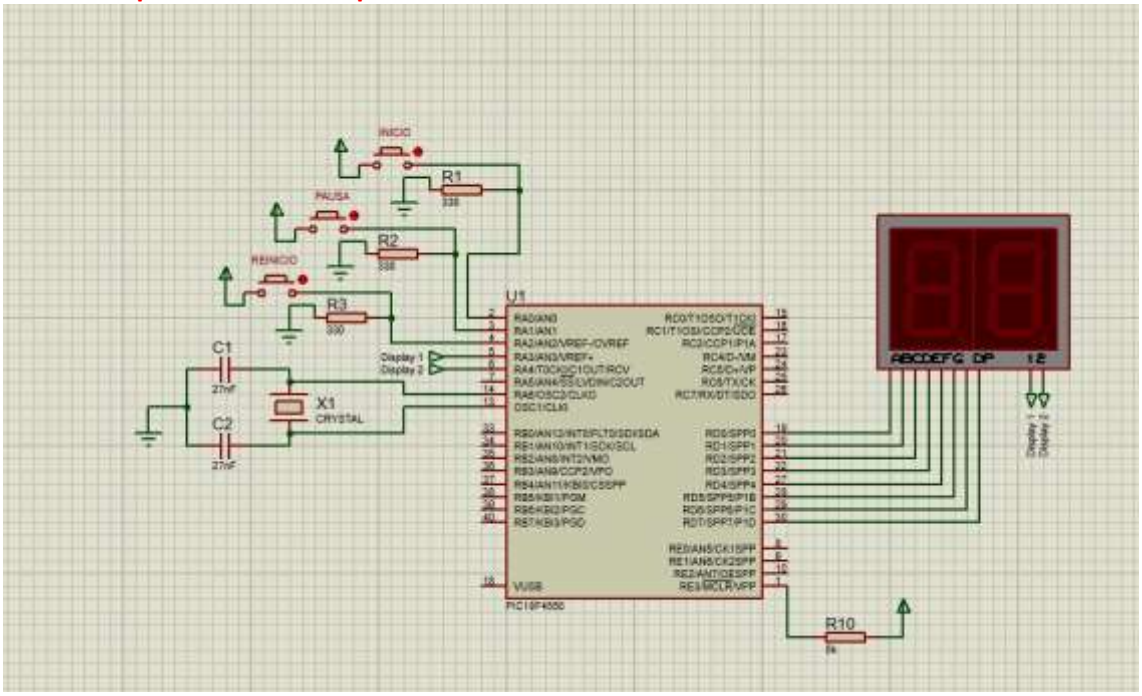
Resultados de la práctica

Descripción del código realizado

```
#include <18f4550.h>   // Libreria del Microcontrolador
#fuses INTRC, NOWDT, NOPROTECT, NOLVP, CPUDIV1, PLL1, NOMCLR // Fusibles (Configuraciones del microcontrolador)
#use delay(clock = 8M)

// DEFINIMOS LAS VARIABLES

#define Display1 pin_a3
#define Display2 pin_a4
#define Inicio pin_a0
#define Pausa pin_a1
#define Reinicio pin_a2

void displays(int d, int u);

// Vector para impresion de numeros en display tipo catodo
int vec_c[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,0x7D,0x07,0x7F,0x67};

void main()
{
   int i=0,j=0, time = 0, B = 0;

   while(true)
   {
      if(input(Inicio)) // Si se presiona el pin de inicio
      {
         while(input(Inicio)); // Se detiene el codigo para evitar errores

         // Etiqueta para hacer el reinicio
         loop:
         B = 0; // Bandera para saber si se ha pausado

         // CICLO DE IMPRESION DE NUMEROS. j = Decenas  i = Unidades
         for(j=0;j<10;j++)
            for(i=0;i<10;i++)
            {
               for(time = 0; (time < 50) || B; time++ ) // Se imprime 50 veces en el display (1 seg). Si esta pausado se queda ahi
               {
                  displays(j,i);

                  if(input(Pausa) && !B) // Permite pausar con el boton "PAUSA" si no se ha pausado antes
                  {
                     while(input(Pausa))
                        displays(j,i);

                     B = 1;
                  }
                  else if(input(Inicio) && B) // Permite reanudar con el boton "INICIO" si se ha pausado antes
                  {
                     while(input(Inicio))
                        displays(j,i);

                     B = 0;
                  }

                  if(input(Reinicio)) // Si se presiona el boton "INICIO"
                  {
                     while(input(Reinicio))
                        displays(j,i);

                     goto loop; // Vuelve a la etiqueta que redefine las variables
                  }
               }
            }
      }
   }
}

void displays(int d, int u) // Realiza el multiplexado de los displays en 20 ms
{
   output_high(Display2); // Apagamos D2
   output_low(Display1); // Prendemos D1
   output_d(vec_c[d]); // Mandamos decenas a D1
   delay_ms(10);
   output_high(Display1); // Apagamos D1
   output_low(Display2); // Prendemos D2
   output_d(vec_c[u]); // Mandamos unidades a D2
   delay_ms(10);

}
```

Poner el botón de encendido y reinicio fue sencillo, pero poner el botón de pausa si nos forzó a analizar línea por línea lo que hacía el programa, debido a que no podíamos dejar de imprimir en el display por el mero funcionamiento del multiplexado, por lo que decidimos aprovechar el ciclo que ya estaba ahí (Para imprimir cada segundo), y no permitir que el programa avance de unidad mediante una bandera.

Circuito implementado en la práctica



Conclusiones de la práctica

Joel Zuñiga:

Durante la realización de la práctica, pude comprender mejor como funcionan los pines del microcontrolador, porque se manda a tierra casi todos los componentes, y sobre todo la técnica de multiplexado para utilizar 2 displays usando los mismos pines que si fuera uno solo.

Daniela Manríquez:

En esta práctica pude reforzar los conocimientos adquiridos en la práctica anterior, así como también aprendí como funciona el multiplexado con los displays, aunque al final usamos uno ya multiplexado para más comodidad.

La utilización del multiplexado es una técnica muy eficiente y aprovecha la capacidad del microcontrolador de trabajar tan rápido, siendo prácticamente imperceptible para nosotros el notar cuando se apagan los displays. Aprendimos más de las capacidades y limitaciones del microcontrolador en esta práctica y se cumplió el objetivo.

Bibliografía

Grindling G. & Weiss B. (2007). Introduction to Microcontrollers. Vienna University of Technology.

Pollán T. (2007). Bloques codificadores y distribuidores. En Electrónica Digital. Zaragoza, España