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: 3

Total de horas: 2 Hrs.

Objetivos

El alumno deberá configurar puertos del microcontrolador como entradas y salidas, además de controlar el tiempo de activación para realizar un multiplexado que sea visible en displays.

Descripción de la práctica

El estudiante deberá realizar la programación e instrucciones necesaria para un microcontrolador, capaz de detectar una ubicación especifica en una matriz de "push buttons", esto con un solo puerto del microcontrolador. La selección que se realice, se mostrara en 2 displays de 7 segmentos, que serán conectados y controlados por un solo puerto del microcontrolador.

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).

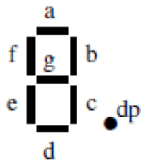


Figura 1. Los siete segmentos y dp de un display numérico.

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.

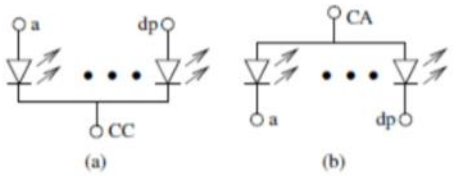
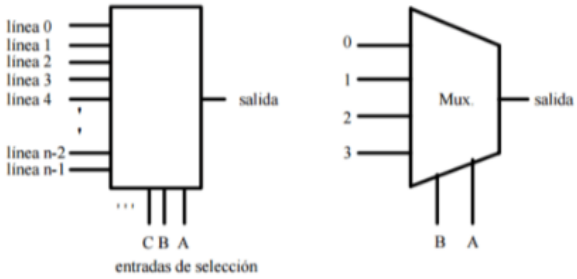


Figura 2. Display numérico con cátodo común (a) y ánodo común (b).

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.

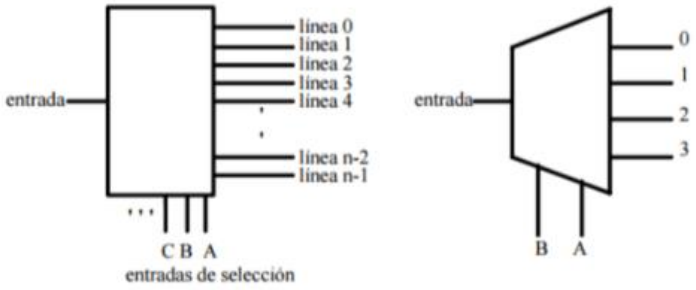


Figura 4. Representa, respectivamente, un demultiplexor genérico de n líneas y un multiplexor de 4 líneas.

El sistema en el que se basa el funcionamiento de una matriz de botones o teclado matricial es el multiplexado. Una matriz de teclado 4×4 tiene 8 pines divididos en 4 filas y 4 columnas. Cuando se pulsa un botón, un pin de la fila se cortocircuita con un pin de la columna.

Los botones de una columna se conectan por medio de una patita en serie, ya que está será la salida de la señal. La otra patita del botón se conectará a un mismo nodo con los otros botones, pero de la fila, ya que estos son los que recibirán la corriente.

Lo que se va a generar con este multiplexado es que se vaya energizando fila por fila, y si se aprieta un botón la señal que está en la fila se va a ir por la columna del botón que se presione lo que ayuda a saber cuál es el botón que se presiona, ya que al ir energizando fila por fila se sabe cuál botón de la columna es el que se aprieta.

EQUIPO Y MATERIALES

- Software para programación
- Software de simulación
- Consumibles electrónicos

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 (Puerto X y Y) quedará a consideración del alumno.
   b) El puerto X será 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.
   d) El puerto Y será configurado como entrada (4 pines) y salida (4 pines)
2. Desarrollar la simulación del circuito (de acuerdo al diagrama de bloques, Figura 5), tomando en cuenta los siguiente:
   a) La matriz de botones tendrá 16 botones que serán considerados como localidades o coordenadas (figura 6).
   b) Una vez que se encienda el microcontrolador los displays deberán marcar “00” y mantendrán este valor hasta que se presione un botón de la matriz.
   c) Cuando se oprima un botón de la matriz, el microcontrolador capturara la coordenada y la mostrara en los displays.

d) El primer display mostrará la columna en la que se encuentra el botón presionado y el segundo “el renglón”.
e) La selección del display a controlar se hará a través de dos pines de salida del microcontrolador

3. Realizar la implementación en físico de la simulación.

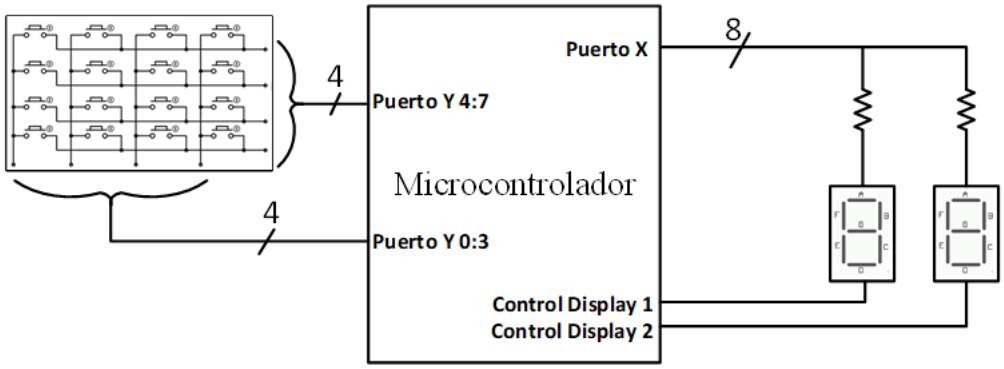


Figura 5. Diagrama de bloques para el circuito *multiplexor I_O.*

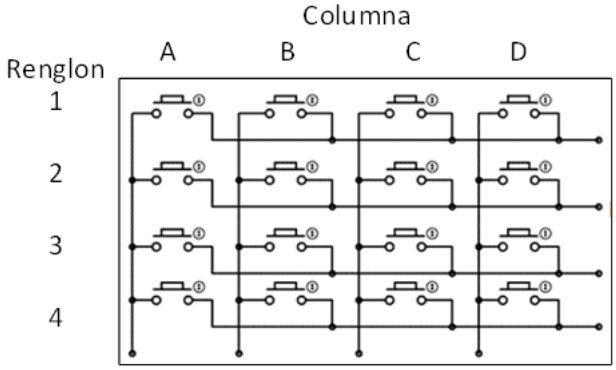
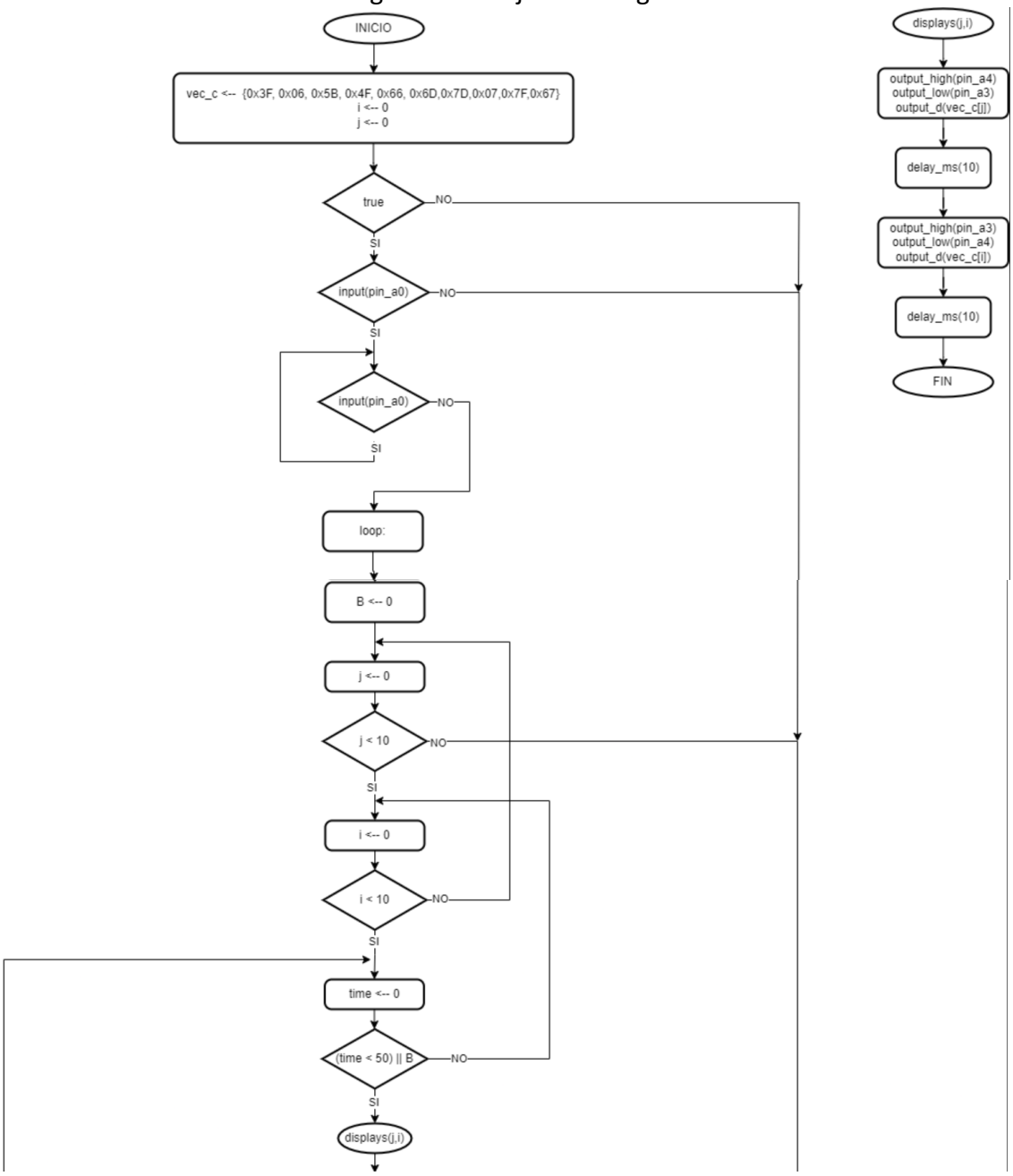
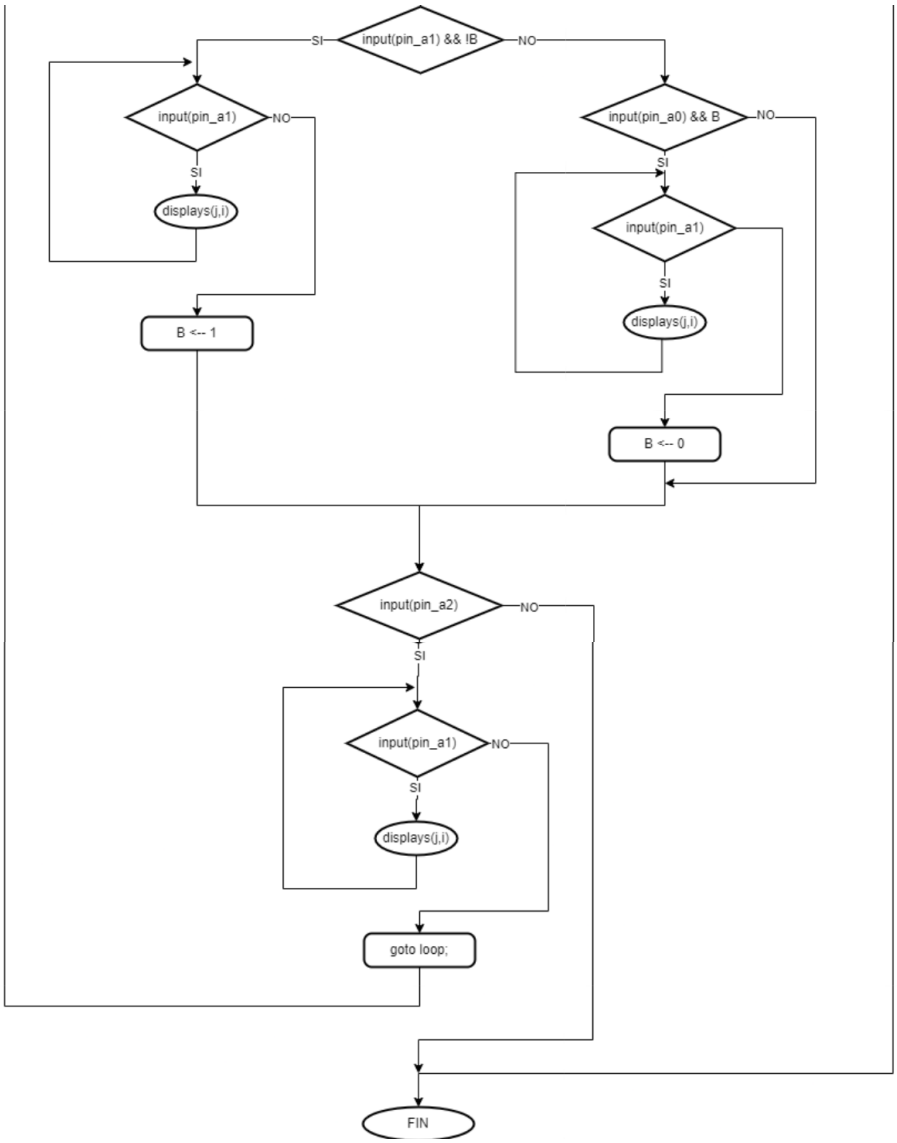
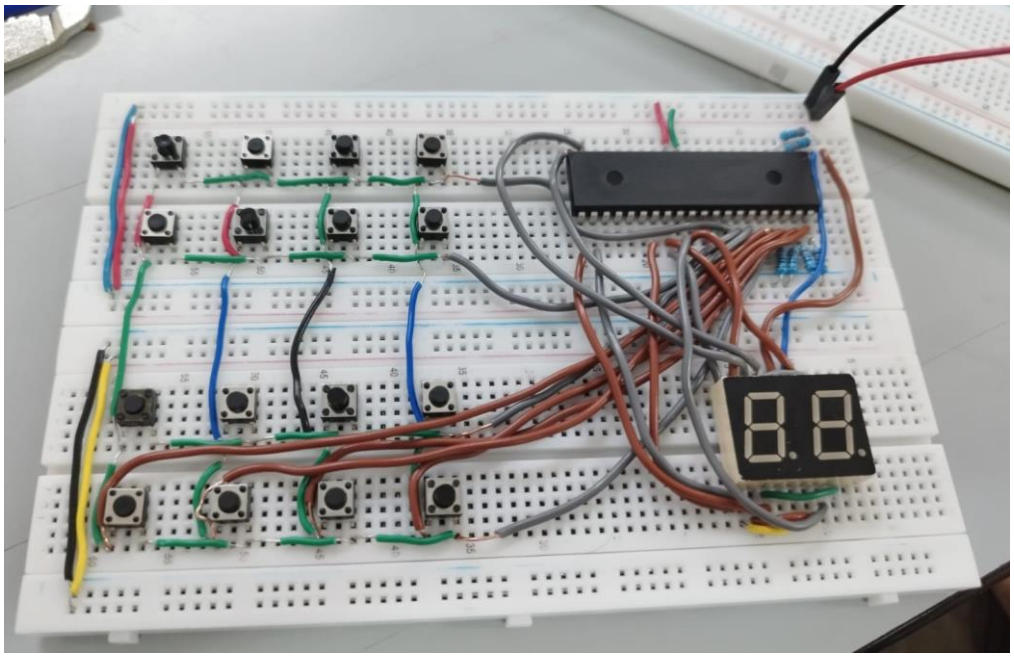


Figura 6. Distribución de coordenadas para la matriz de botones.

Diagrama de flujo del código.







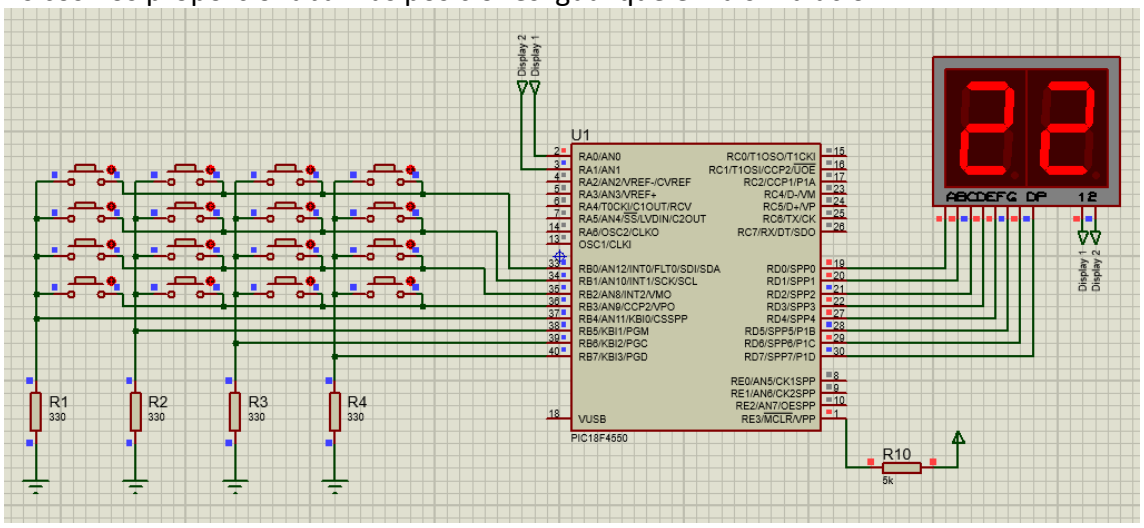
Circuito armando

Código explicado

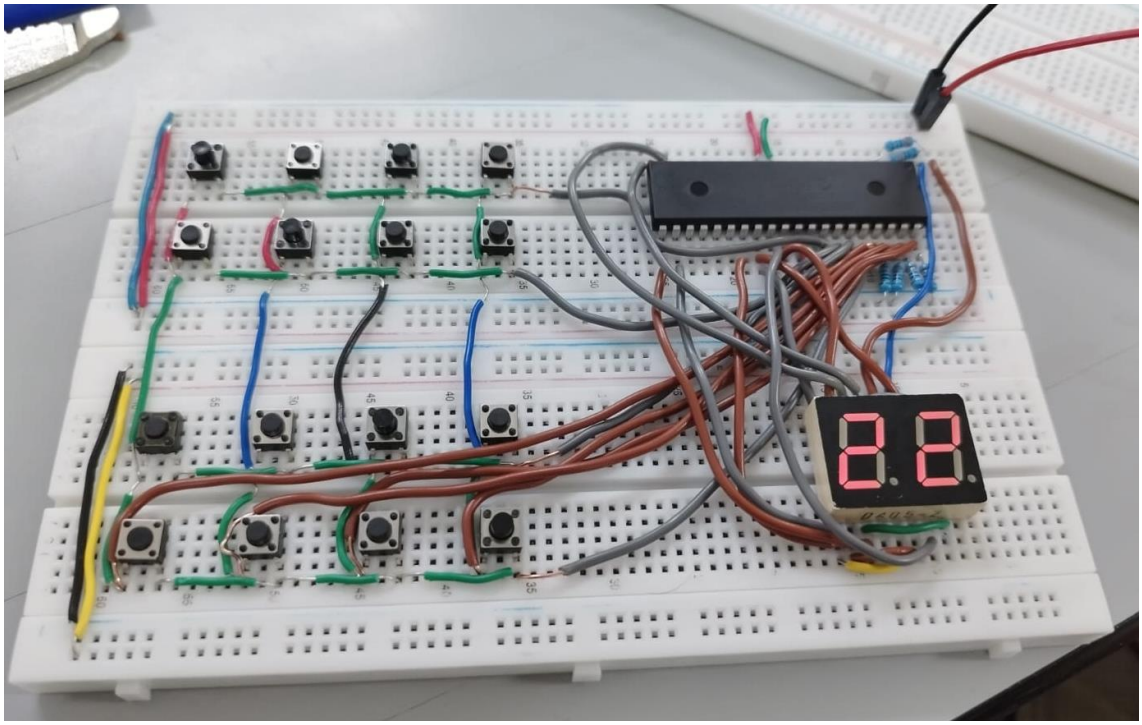
```c
#include <18f4550.h>   // Libreria del Microcontrolador
#fuses INTRC, NOWDT, NOPROTECT, NOLVP, CPUDIV1, PLL1, NOMCLR // Fusibles (Configuraciones del microcontrolador)
#use delay(clock = 8M)  // 4M es 4 millones. Establecemos la velocidad va a operar el micro

#define Display1 pin_a0
#define Display2 pin_a1

// Columna, Renglon. 4 de Entrada, 4 de Salida

void displays(int d, int u);
int CHECK(int *i);

int vec_c[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,0x7D,0x07,0x7F,0x67};

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

   while(true)
   {
      output_high(pin_b0);
      if(CHECK(&i)) // Si se encontro columna presionada. Guardamos la fila que se activo
         j = 1;
      output_low(pin_b0);

      output_high(pin_b1);
      if(CHECK(&i)) // Si se encontro columna presionada
         j = 2;
      output_low(pin_b1);

      output_high(pin_b2);
      if(CHECK(&i)) // Si se encontro columna presionada
         j = 3;
      output_low(pin_b2);

      output_high(pin_b3);
      if(CHECK(&i)) // Si se encontro columna presionada
         j = 4;
      output_low(pin_b3);

      displays(i,j);
   }
}

void displays(int d, int u) // Imprime las unidades y decentas en 20 ms
{
   output_high(Display2);
   output_low(Display1); // Prendemos D1
   output_d(vec_c[d]); // Mandamos decenas a D1
   delay_ms(10);
   output_high(Display1);
   output_low(Display2); // Prendemos D2
   output_d(vec_c[u]); // Mandamos unidades a D2
   delay_ms(10);

}

int CHECK(*i) // Checa si alguna columna se presiono, pone su valor en i y devulve 1. Sino no modifica i y devuelve 0
{
   if(input(pin_b4))
   {
      *i = 1;
      return 1;
   }
   else if(input(pin_b5))
   {
      *i = 2;
      return 1;
   }
   else if(input(pin_b6))
   {
      *i = 3;
      return 1;
   }
   else if(input(pin_b7))
   {
      *i = 4;
      return 1;
   }
   else
      return 0;
}
```

Resultados de la práctica

Comparando los resultados físicos y simulados, estos fueron iguales, ya que los botones físicos nos proporcionaban las posiciones igual que en la simulación.



Resultados simulados



Resultados físicos

Discusión

- ¿Cómo se resolvió el problema del multiplexado en el microcontrolador?
  Se resolvió poniendo un delay para controlar los tiempos en los que se energizaba cada fila y los tiempos que se mostraban los números en pantalla.
- Poner un caso en el que crean que esta práctica tendría uso en la industria.
  Un caso podría ser cuando hay un sistema automatizado que necesita cierta precisión en cuanto a cuando activarlo, pausarlo e iniciarlo

- ¿En qué casos sería mejor emplear un circuito específico para multiplexar?
  Cuando queremos utilizar menos pines, es decir, ser más eficientes al momento de realizar un proceso y obtener el mismo resultado

Conclusiones de la práctica

En esta práctica pudimos reforzar los conocimientos adquiridos en la práctica anterior, así como también aprendimos como funciona el multiplexado con los botones de 4 patitas en físico, ya que en la simulación solo tenían dos, al final resulto que se comportaban de la misma manera solo que se conectaban las patitas en diagonal. Al final todo resulto bien en físico así como en la simulación.

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.