1. **DIAGRAMA DE BLOQUES**

**BOTONERA 1 (IZQ)**

**BOTONERA 2 (DER)**

**BOTONERA 3   
(CAMBIO)**

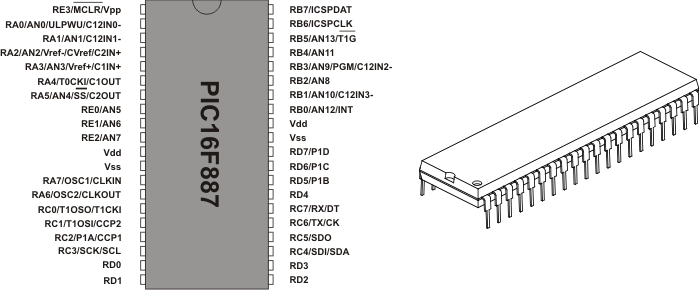
**BOTONERA 4 (Velocidad)**

**TERMINAL**

**VIRTUAL**

**(PROTEUS)**

**VISUALIZACIÓN MEDIANTE LA GLCD**



1. **DIAGRAMA DE FLUJO FUNCIONAL DEL PROGRAMA PRINCIPAL Y DE LAS SUBRUTINAS**

**SUBRUTINAS DEL JUEGO: CARGAR MENU**

**Subrutina del juego – Caen las piezas**

1. **DESCRIPCIÓN DEL ALGORITMO O ESTRATEGIA UTILIZADA**

Al iniciar el programa se declaran todas las variables a utilizar, se configura la Glcd en los puertos D, C y E, luego dentro del main() se inicializan los puertos respectivos como entradas o salidas, se activan las debidas interrupciones, se inicializa el módulo bluetooth, se grafica el título del juego y se procede a mostrar el menú del juego con ayuda de un while(bien==1).

En la parte del sonido, se colocó como salida analógica al puerto RA2 inicializando en el TRISA. La melodía que se programó es Yellow house.

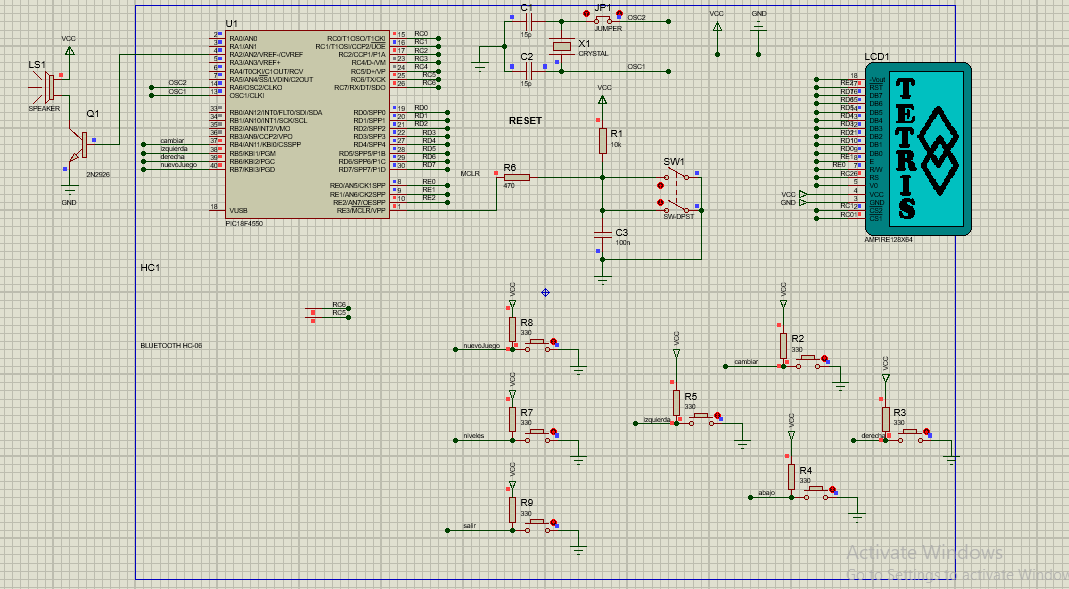
El menú consiste en 3 opciones mediante las cuales el usuario tiene la libertad de iterar entre ellas, cada opción tiene relacionado por una interrupción en RB7 que varía entre 0 y 1, según se aplaste la botonera que entra en el juego.

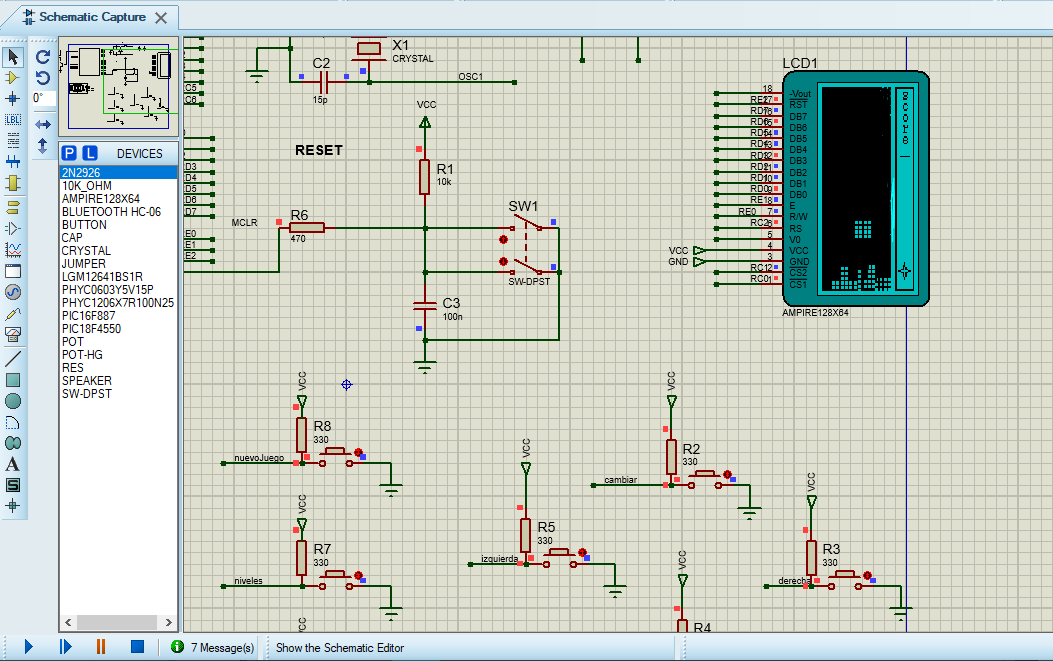
Al ingresar en el juego con la variable indicadora entre bien==1 (será cierto hasta que finalice el juego) las piezas caerán consecutivamente una tras otra. Ahora para evitar generar números aleatorios para las piezas, ya que no hay forma de usar srand(time(NULL)) en MikroC, se decidió utilizar un switch- case para que la variable “contador” para piezas intercalara en ese menú. Las piezas caerán de forma indefinidas hasta que complete el tamaño de la matriz del GLCD. Las variables “x” e “y” ya están ajustadas con las posiciones del final de la pieza dada, es decir, cuadrado en x=117 es el asentamiento para aquella pieza.

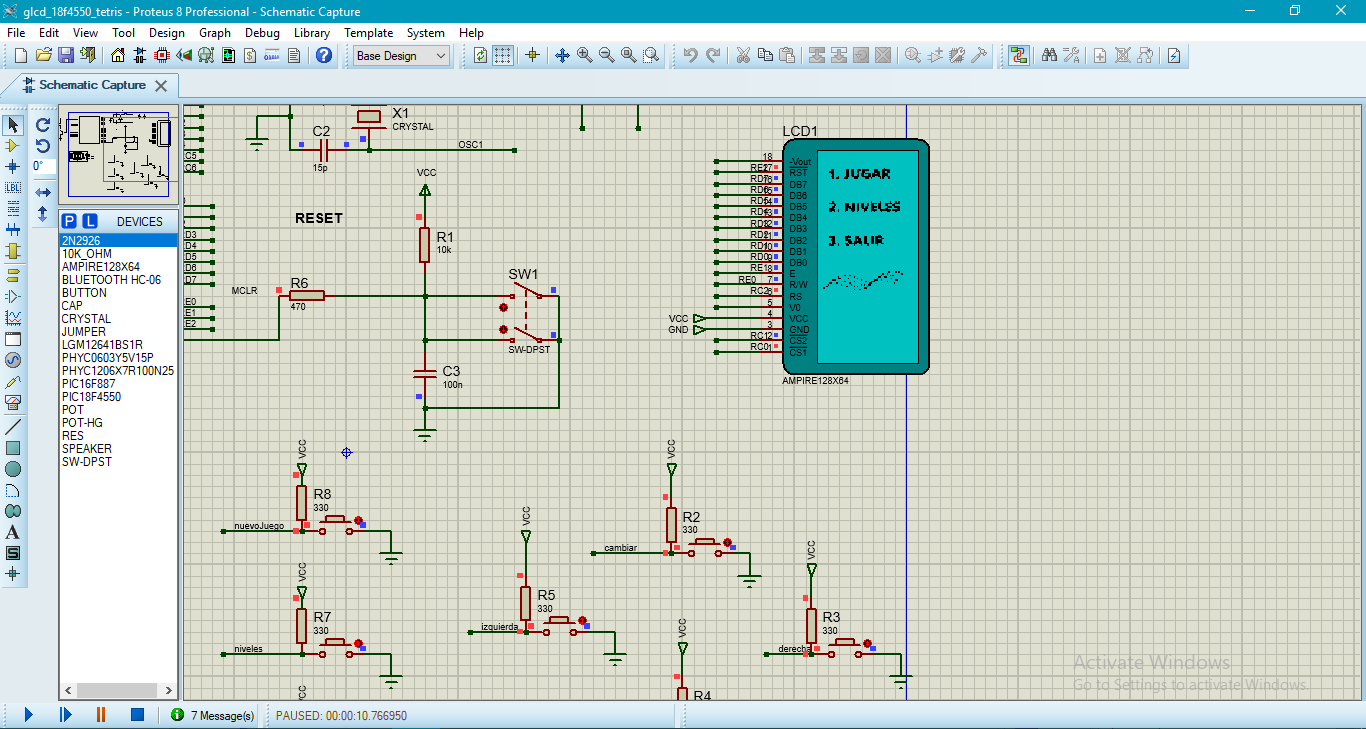
Para la parte de girar pieza, se hace una traslación de x y y, es decir un cambio entre las coordenadas x e y.

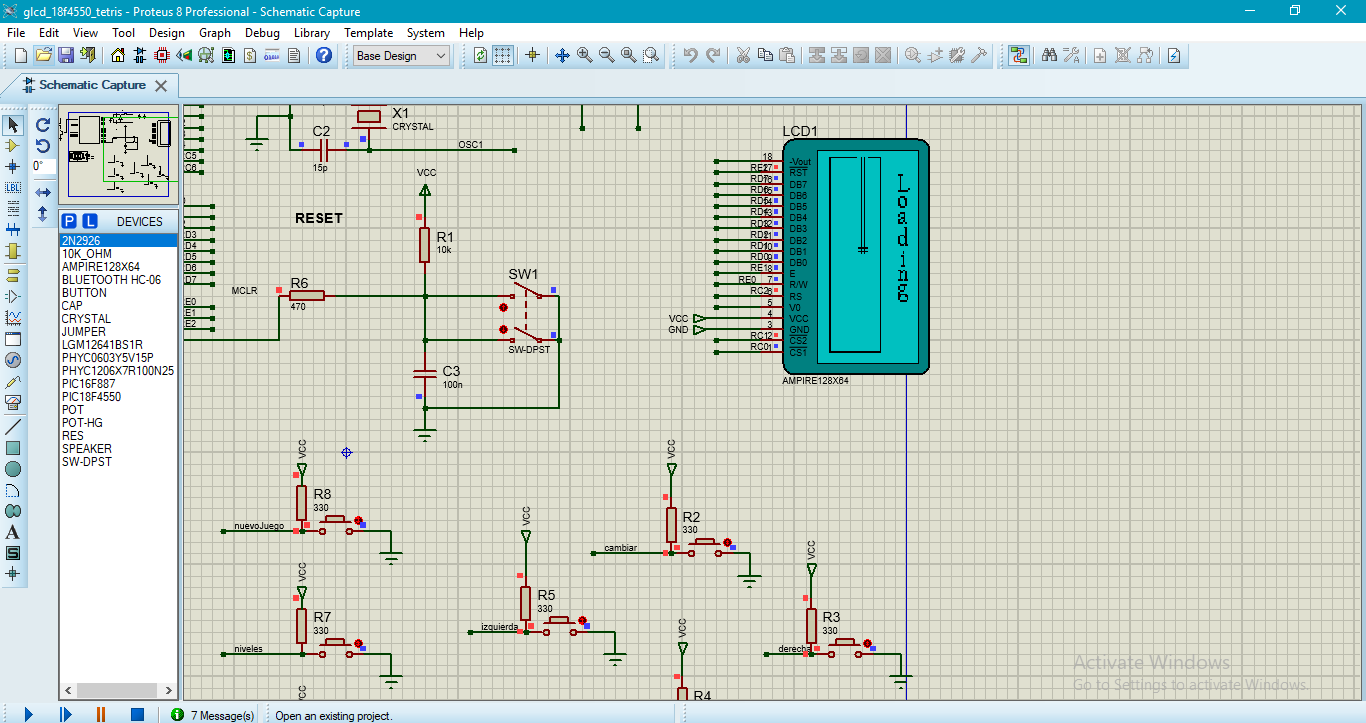
Uno pierde el juego cuando se llena el contador general, que representa la cantidad de piezas que puede llenar la matriz GLCD.

1. **LISTADO DEL PROGRAMA FUENTE EN LENGUAJE C CON COMENTARIOS EN LAS LÍNEAS DE CÓDIGO QUE CONSIDERE FUNDAMENTALES**
2. **RECOMENDACIONES**
3. **COPIA IMPRESA DEL CIRCUITO ARMADO EN PROTEUS PARA LA SIMULACIÓN EN EL MOMENTO DE SU EJECUCIÓN**









1. **CONCLUSIONES**

Fue una excelente dedición trabajar con el pic 18f4550 ya que viene más memoria RAM que el pic 16f887, ya que el juego de por si consume mucha memoria por las imágenes.

Se desarrollaron destrezas en la parte de algoritmos que permitieron solucionar cualquier problema presentado durante la elaboración del programa.

Una buena práctica de programación es el uso de “funciones definidas” ya que no hace falta escribir cuantas veces sea necesario aquel código, sino que tan solo con mencionar el nombre de la función y mandando los parámetros correspondientes del programa.

1. **RECOMENDACIONES**

Tener cuidado con la configuración de los pines RC4 y RC5 del PIC18F4550 ya que esos pines se usan para la transmisión y recepción de datos, no es aconsejable usarlos para el Glcd ya que setear baudios a otro pin distinto es más complicado.

Revisar el datasheet de la GLCD y el dispositivo, cuestiones como una mala soldada en los espadines causan graves problemas en la salida del programa.

Tener cuidado al momento de adquirir la GLCD, ya que no todas tienen el controlador que le permiten un exitoso funcionamiento con el PIC 18f4550.