

Trabajo Práctico EMbebidos

“Simon dice”

1.- Secuencia inicial indicando que el juego no está iniciado.

1000 1100 0100 0110 0010 0011 0001 y vuelve al principio (200ms por etapa)

2.- Si se está mostrando la secuencia inicial y se presiona el SW1 se permite elegir la cantidad de niveles a jugar (4 a 12). Cuando se presiona por primera vez se muestra el nivel configurado, este queda activo durante 2seg. Si se presiona mientras se muestra el nivel se incrementa en uno el nivel hasta llegar a 12. Cada vez que se presiona SW1 se deja el nivel mostrándose durante 2Seg como máximo.

3.- Si se está mostrando la secuencia inicial y se presiona SW0 por más de 1Seg y menos de 2Seg el juego comienza, después de la siguiente secuencia.

a.- Se muestra el nivel por 1Seg

b.- Se encienden todos los led durante 1Seg y se comienzan a apagar de derecha a izquierda

XXXX (Nivel 2Seg) -> 1111 (1Seg) -> 1110 (1000ms) -> 1100 (1000ms) -> 1000 (1000ms) -> 0000 (2Seg) y el juego comienza.

4.- Si se está en el juego y no se presiona ningún pulsador durante 3Seg o se presiona uno equivocado, se da por perdido el juego, en esta caso los leds se encienden todos juntos parpadeando cada 100ms durante 2Seg.

Si se completan los niveles, el juego se da por ganado, en este caso se encienden todos los leds parpadeando cada 500ms por 2Seg.

Ya sea que se haya perdido o ganado, después de mostrar la secuencia correspondiente se deben apagar todos los leds esperar 3Seg y volver a mostrar la secuencia inicial.

5.- Los leds se elegirán de manera aleatoria, debe ser un número entre 0 y 3, el tiempo de encendido de cada led que muestra la secuencia a memorizar será también aleatorio y estará comprendido entre 100...500 ms.

Se debe incluir la librería estándar de c: (<stdlib.h>)

La función para generar los Leds aleatorios es la siguiente:

randomLed = ((rand()%MAXLED+1)-1); // ((rand()%4+1)-1);

Como el 0 es muy esporádico, se genera hasta 4 y se resta 1. Hay que controlar que el valor no sea 0.

Para generar el tiempo utilizar la siguiente función:

randomInterval = (rand()%(MAXTIME+1)+BASETIME); // ((rand()%(500+1)+100);

Antes de utilizar la función rand(), se debe inicializar la semilla para generar los números aleatorios con la función srand(número), por ejemplo:

srand(miTimer.read_us());

6.- Consejos:

Se puede utilizar BusIn para los botones:

BusIn Nombre(PIN0,PIN1,PIN2...PIN15);

Se puede utilizar BusOut para los Leds

BusOut Nombre(PIN0,PIN1,PIN2...PIN15);

De esa forma se los maneja cómo un grupo y no separados, otorga ventajas para hacer operaciones bitwise.