ESCUELA DE INGENIERÍA INFORMÁTICA

Grado en Ingeniería Informática

PERIFÉRICOS E INTERFACES

PRÁCTICAS DE LABORATORIO 17/18

Módulo 4 – Práctica 4

Máquina de juegos

"Práctica básica y mejorada"

Ana Isabel Santana Medina

Grupo: 1.18.43

Periodo de realización: 10ª – 15ª semana

Índice

1. Competencias y objetivos básicos de la práctica

Los siguientes datos e información detallada corresponden a la cuarta práctica base de la asignatura Periféricos e Interfaces y la mejora de la misma.

Una breve descripción de esta es que se profundiza en el diseño de los interfaces de entrada/salida para lo que se propone utilizar el hardware diseñado en la práctica anterior pero cambiando el software para implementar una nueva funcionalidad del periférico orientada al ocio. Para ello, se propone diseñar y programar un juego que básicamente consistirá en adivinar un número de 00 a 99 en un tiempo dado.

Con la realización de esta práctica hemos alcanzado las siguientes competencias:

1. Capacidad para entender e interrelacionar los diferentes componentes hardware y software que integran un interfaz sencillo de un sistema basado en microcontrolador.
2. Capacidad para diseñar, implementar y verificar el correcto funcionamiento de dispositivos externos sencillos para ser conectados al interfaz paralelo.
3. Capacidad para el desarrollo de programas que permitan el control básico del dispositivo externo haciendo uso de un lenguaje de programación.
4. Capacidad para desarrollar programas que doten al sistema de una determinada funcionalidad.
5. Capacidad para aprender y aplicar nuevos conceptos de forma autónoma e interdisciplinar.
6. Capacidad para emplear la creatividad en la resolución de los problemas.

Gracias a la consecución de los siguientes objetivos:

1. Profundizar en el conocimiento de la estructura interna de un periférico sencillo dotándolo del software de control e interconexión con el usuario haciendo uso

de la plataforma Arduino.

2. Conocer las particularidades de las diferentes señales físicas que aparecen en los conectores de la placa a efectos de poder conectar dispositivos correctamente.
3. Conocer el funcionamiento de los componentes electrónicos básicos utilizados y saber integrarlos en el sistema para lograr la funcionalidad pretendida del periférico.
4. Saber diseñar el software de control del periférico básico propuesto para su correcto funcionamiento de acuerdo a la funcionalidad exigida incluido la comunicación con el usuario y con el sistema computador si fuese necesario.

2. Descripción y realización de la práctica y mejora

Hemos hecho uso de la placa Arduino, que es la encargada de controlar los periféricos, así como la placa del laboratorio en la que se encuentra integrada la misma:

- El Display 7 segmentos encargada de mostrar los números que correspondan
- 2 botones o pulsadores
- Un altavoz para reproducir frecuencias
- Un teclado matricial de 3x4
- Una pantalla de 16x4

Para esta práctica básicamente lo que he hecho ha sido reutilizar el código de la práctica 3 y editarlo para hacer el juego que se pide. La descripción del mismo es la siguiente: La máquina deberá generar un número aleatorio entre 00 y 99 y el usuario tendrá que adivinarlo tecleando números a través del teclado. El tiempo de cada partida es limitado, lo fijará el sistema en función del grado de dificultad del juego y se visualizará constantemente en el display por lo que siempre conoceremos el tiempo que queda para terminar la partida. Los números, de uno o dos dígitos, serán introducidos a través del teclado y terminarán, siempre, con la tecla "#" que hará de retorno de carro o fin de la entrada de datos. Si el usuario acierta, el sistema emitirá varios pitidos que creen en el usuario la sensación de acierto y mostrará en pantalla los puntos obtenidos. Si el

usuario falla se emitirá un pitido largo y ayudará al usuario informando de si el número secreto es mayor o menor al introducido.

Poco decir de la estrategia puedo, pues como es un código ampliamente reutilizado, las variables son las mismas, a excepción de algunas como la correspondiente a controlar la dificultad (mejora), el contador, etc. Además, he hecho uso de los puertos Serial y Serial3 para observar los datos y enviar datos al display LCD.

Para la mejora he realizado la número uno, propuesta por el profesorado, la cual se trataba de hacer uso del pdown para que el usuario pueda seleccionar el nivel de dificultad del juego: 0, 1 y 2. Podemos observar a través de esta tabla cómo he distribuido los tiempos y los niveles de dificultad con los correspondientes puntos; esta tabla y ejemplo fue propuesto por los profesores y se incluía en el enunciado de la práctica, no obstante, si quisiéramos cambiar la dificultad, así como el tiempo o los puntos con los que se recompensa sería muy sencillo modificarlo.

Nivel dificultad	Tiempo restante (seg)	Peso	Puntos
0	60	1	60
1	40	3	120
2	20	8	180

El código que se muestra a continuación contiene la práctica básica y la mejorada pues la comencé a editar toda en el mismo archivo y así resultó.

3. Código de la práctica básica y mejorada

```
/**
 *
 * Autor: Ana Isabel Santana Medina
 * Grupo: 43
 * Fecha: 20/05/18
 *
 */

//#############################################################################
 // Definición de variables del programa
//#############################################################################
```

```
 /**
  * Variables del display 7 Segmentos
  */

 const int unidades = 49;    // Señal D4-C1 (pin 49, PL0)
 const int decenas = 48;     // Señal D3-C2 (pin 48, PL1)
 const int centenas = 47;    // Señal D2-C3 (pin 47, PL2)
 const int millares = 46;    // Señal D1-nn (pin46, PL3)

 int unidad = 0;   // Número a mostrar en el 7S
 int decena = 0;   // Número a mostrar en el 7S
 int centena = 0;  // Número a mostrar en el 7S
 int millar = 0;   // Número a mostrar en el 7S

// Lista con los numeros que muestran los numero decimales en el display
 const int numeros[12] = {63,6,91,79,102,109,125,7,127,103,1,8};

 /**
  * Variables de los pulsadores y altavoz
  */

 const int pup = 34; // Pin botón arriba (PC3-in)
 const int pdown = 31; // Pin botón abajo (PC6-in)
 const int pright = 30; // Pin botón derecha (PC7-in)
 const int pleft = 32; // Pin botón izquierda (PC5-in)
 const int penter = 33; // Pin botón centro (enter) (PC4-in)

 const int speaker = 37; // Pin altavoz (PC0-out)

 /**
  * Variables auxiliares
  */

 int puntero = 0;
 int pausa = 1;
 int contador = 0;
 int aleatorio = 0;
 int numero[3] = {0,0};
 int pos = 1;
 int bandera = 0;
 int segundo = 0;
 int dificultad = 1;


//#############################################################################
```

```
// Comienzo del programa
//################################################################################################

 /**
  *  Preparación en la que activamos los puertos, sus correspondientes y las interrupciones.
  * Esta solo se realiza una vez.
  */

void setup(){

  Serial.begin(9600);
  Serial3.begin(9600);

//################################################################################################

  DDRA = B11111111; // Definimos el PORTA de salida (Bus de salida conectado al 7S)

  DDRC = B00000001; // Definimos el PORTC de entrada (pulsadores) salvo PC0 (salida frecuencia del altavoz)
  PORTC = B11111000; // Activamos el pull-up interno de las líneas entrada

  DDRL = B00001111; // Definimos el PORTL de entrada (teclado) menos los que son de salida (7S)
  PORTL = B11111111; // Activamos el pull-up interno de las líneas entrada PC7-PC3

//################################################################################################

  attachInterrupt(1, interruption, CHANGE);

  decena = 6;
  unidad = 0;
  reiniciar();
}

/**
 * Bucle único y principal, todo lo demás son métodos adicionales
 */

void loop() {
  while(pausa == 1){
    pausa = digitalRead(pup);
    if(pausa == 0){
      reiniciar();
      tone(3, 100);
      if(dificultad == 1){
```

```
      decena = 6;
      unidad = 0;
    }
    if(dificultad == 3){
      decena = 4;
      unidad = 0;
    }
    if(dificultad == 8){
      decena = 2;
      unidad = 0;
    }
    aleatorio = random(100);
  }
  if(digitalRead(pdown) == LOW && pausa == 1){
    char t = lecturaTeclado();
    int i = 0;
    while(i == 0){
      if(t == '0'){
        dificultad = 1;
        i = 1;
      }
      if (t == '1'){
        dificultad = 3;
        i = 1;
      }
      if (t == '2'){
        dificultad = 8;
        i = 1;
      }
      t = lecturaTeclado();
      controlPuntero();
    }
    Serial.print(dificultad);
  }
}

if (digitalRead(pup) == LOW && digitalRead(pdown) == LOW){
  segundo = 0;
  bandera = 0;
  apagado();
}
if (segundo == 1){
  decrementarUnidad();
  segundo = 0;
}
```

```
  if (bandera == 1){
    int resultado = comprobarNumero(tecla);
    delay(200);
    bandera = 0;
    if (resultado == 1){
      comprobarResultado();
    }
  }
}

/**
  * Manejador de interrupciones
 */

void interruption(){
  if (bandera == 0){
    contador++;
    if (contador == 100){
      segundo = 1;
      contador = 0;
    }
    if(puntero == 0){
      encenderDisplay(unidad, puntero);
    }else if(puntero == 1){
      encenderDisplay(decena, puntero);
    }else if(puntero == 2){
      PORTA = 0;              // Evitar el efecto fantasma
      PORTL = B11111011;
    }

    char tecla = lecturaTeclado();
    if (tecla != 'n'){    // No se ha pulsado nada
      bandera = 1;
    }

    if(puntero == 2){
      PORTL = B11111111;
    }

    puntero++;

    if(puntero == 3){
      puntero = 0;
    }
```

```
  }
}

/**
  * Funcion detecta donde se enciende el display
 */

void controlPuntero(){
  puntero++;
  if (puntero == 3){
    PORTL = B11111111;
    puntero = 0;
  }
  if (puntero == 0){
    PORTA = 0;
    PORTL = B11111110;
  }
  if (puntero == 1){
    PORTA = 0;
    PORTL = B11111101;
  }
  if (puntero == 2){
    PORTA = 0;
    PORTL = B11111011;
  }
}

/**
  * Funcion detecta que tecla hemos apretado
 */

char lecturaTeclado(){
  if(puntero == 0){
    if (digitalRead(42) == LOW){
      return '1';
    }else if (digitalRead(43) == LOW){
      return '4';
    }else if (digitalRead(44) == LOW){
      return '7';
    }else if (digitalRead(45) == LOW){
      return '*';
    }
  }else if(puntero == 1){
    if (digitalRead(42) == LOW){
      return '2';
```

```
    }else if (digitalRead(43) == LOW){
      return '5';
    }else if (digitalRead(44) == LOW){
      return '8';
    }else if (digitalRead(45) == LOW){
      return '0';
    }
  }else{
    if (digitalRead(42) == LOW){
      return '3';
    }else if (digitalRead(43) == LOW){
      return '6';
    }else if (digitalRead(44) == LOW){
      return '9';
    }else if (digitalRead(45) == LOW){
      return '#';
    }
  }
  return 'n';
}

/**
  * Funcion para manejar las decenas y unidades
 */

void encenderDisplay(int n, int p){
  if (p == 0){
    PORTL = B11111110;  // Enciende el primer display (unidades)
    PORTA = numeros[n];
  }else if(p == 1){
    PORTL = B11111101;  // Enceiende el segundo display (decenas)
    PORTA = numeros[n];
  }
}

/**
  * Funcion para mostrar el numero introducido en el display
 */

void numeroDisplay(int n){
  if (pos == 0){
    reiniciar();
    Serial3.write(" >> Numero introducido: ");
  }
  Serial3.print(n);
```

```
}

/**
  * Funcion para controlar el numero introducido
 */

int comprobarNumero(char tecla){
  if(pos == -1){
    if (tecla == '#'){
      return 1;
    } else if (tecla == '*'){
      asterisco(5000);
    } else {
      fallo();
    }
  }
  switch(tecla){
      case '0':
        numero[pos] = 0;
        pos--;
        break;
      case '1':
        numero[pos] = 1;
        pos--;
        break;
      case '2':
        numero[pos] = 2;
        pos--;
        break;
      case '3':
        numero[pos] = 3;
        pos--;
        break;
      case '4':
        numero[pos] = 4;
        pos--;
        break;
      case '5':
        numero[pos] = 5;
        pos--;
        break;
      case '6':
        numero[pos] = 6;
        pos--;
        break;
```

```
      case '7':
        numero[pos] = 7;
        pos--;
        break;
      case '8':
        numero[pos] = 8;
        pos--;
        break;
      case '9':
        numero[pos] = 9;
        pos--;
        break;
      case '#':
        return 1;
      case '*':
        asterisco(5000);
        reiniciar();
        return 0;
      default:
        break;
  }
  if (pos != 1){
    numeroDisplay(numero[pos+1]);
  }
  return 0;
}

/**
  * Función que decrementa las unidades
 */

void decrementarUnidad(){ // Disminuye unidad
  unidad--;
  if (unidad < 0 && decena > 0){
    decena--;
    unidad = 9;
  }else if (unidad < 0){
    unidad = 0;
    decena = 0;
    fallo();
    parpadeo();
    apagado();
  }
}
```

```
/**
  * Funcion para comprobar el resultado
 */

void comprobarResultado(){
  int resultado = 100;
  if (pos == -1){
    resultado = numero[1]* 10 + numero[0];
  } else {
    resultado = numero[1];
  }
  if (resultado == aleatorio){
    win();
    asterisco(200);
    delay(100);
    asterisco(1000);
    delay(100);
    asterisco(4000);
    parpadeo();
    apagado();
  } else {
    if (resultado > aleatorio){
      mostrarFallo(1);
      fallo();
    } else {
      mostrarFallo(-1);
      fallo();
    }
    pos = 1;
  }
}

/**
  * Funcion para mostrar al usuario que ha ganado y la puntuacion
 */

void win(){
  Serial3.write(0xFE);
  Serial3.write(0x45);
  Serial3.write(0x14);
  Serial3.write("¡Felicidades! YOU WIN");
  Serial3.write(0xFE);
  Serial3.write(0x45);
  Serial3.write(0x54);
  Serial3.write("Puntuacion: ");
```

```
  int resultado = (decena*10 + unidad)* dificultad;
  Serial3.print(resultado);
}

/**
  * Funcion para mostrar al usuario que ha fallado
 */

void mostrarFallo(int n){
  Serial3.write(0xFE);
  Serial3.write(0x45);
  Serial3.write(0x14);
  Serial3.write("Oh no... Has fallado");
  Serial3.write(0xFE);
  Serial3.write(0x45);
  Serial3.write(0x54);
  if(n == 1){
    Serial3.write("El numero que buscas es menor");
  }else{
    Serial3.write("El numero que buscas es mayor");
  }
  Serial3.write(0xFE);
  Serial3.write(0x45);
  Serial3.write(0x00);
}

void asterisco(int frec){
  noTone(3);
  tone(speaker, frec, 100);
  delay(100);
  noTone(speaker);
  tone(3, 100);
  pos = 1;
  numero[1] = 0;
  numero[0] = 0;
}

/**
  * Funcion para reiniciar el lcd
 */

void reiniciar(){
  Serial3.write(0xFE);
  Serial3.write(0x51);
  Serial3.write(0xFE);
```

```
  Serial3.write(0x46);
}

/**
  * Funcion para los fallos
 */

void fallo(){
  asterisco(4000);
  delay(100);
  asterisco(1000);
  delay(100);
  asterisco(100);
}

/**
  * Funcion para hacer que parpadee el display
 */

void parpadeo(){
  for(int i = 0; i < 40; i++){
    detachInterrupt(1);
    delay(50);
    attachInterrupt(1, interruption, CHANGE);
  }
  tone(3,100);
}

/**
  * Funcion para apagar el tercer display
 */

void apagado(){
  pausa = 1;
  noTone(3);
  delay(100);
  PORTL = B11111111;
}
```

4. Conclusiones

Esta práctica me ha sido grata de realizar puesto que se trataba de un juego, y aunque dio algunos problemas, pude solucionarlos al final. Me hubiera gustado implementar la

segunda mejora ya que me encantó hacer uso de la memoria, pero no disponía de suficiente tiempo.

5. Comentarios finales

Como comentarios finales, he de añadir que, en mi opinión, han resultado suficiente los recursos que nos han facilitado para la realización de la práctica, puesto que aparte de estos, teníamos la capacidad de acceder a internet. Pero he de añadir que al tener tantos exámenes no he podido dedicarle tanto tiempo como me gustaría.

6. Recursos empleados

Para la realización de esta práctica he hecho uso de los siguientes recursos:

- **Arduino IDE**
- **Enunciado de la práctica 2017**
- **Transparencias de teoría**
- **Transparencias de la práctica**
- **Página web de Arduino**
- **Documentación varia**