UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

MINIPROYECTO PERIFÉRICO E INTERFACES HEXAVISION

AITOR VENTURA DELGADO

GRADO EN INGENIERÍA INFORMÁTICA

10 DE JUNIO DE 2020

ESCUELA DE INGENIERÍA INFORMÁTICA

Índice

OBJETIVOS 1
RECURSOS UTILIZADOS 1
REALIZACIÓN PRÁCTICA 2
CONCLUSIONES & COMENTARIOS FINALES 11

OBJETIVOS

Este proyecto se centra en la aplicación de conocimientos teóricos asociados al módulo 3 de la asignatura, relativos a los periféricos de entrada y salida de datos. Dados ya un programa sencillo, un "Turnomatic", el alumno deberá de hacer uso de los diferentes puertos de entrada salida de una placa Arduino para implementar diferentes funciones dadas por el profesorado. Con ello, con este proyecto lo que se pretende conseguir es que los estudiantes alcancen satisfactoriamente las siguientes capacidades:

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

Y, además, conseguir los siguientes objetivos:

- Conocer la estructura interna de un interfaz típico de entrada/salida a través del estudio y manejo de los puertos de una tarjeta Arduino.
- Poner en práctica los conocimientos básicos sobre las entradas/salidas paralelas en un sistema computador.
- Conocer y entender la funcionalidad de los diferentes pines de los puertos de E/S del microcontrolador, así como los múltiples aspectos relativos a la conexión con el hardware externo.
- Conocer los aspectos prácticos de funcionamiento de los diferentes componentes básicos a utilizar en la práctica.
- Entender la integración y conexión de componentes básicos para diseñar periféricos sencillos que cumplan con una funcionalidad determinada.
- Analizar, diseñar e implementar el software de control del periférico para su correcto funcionamiento en base a la funcionalidad especificada.
- Verificar y depurar la integración hardware/software para la aplicación hasta alcanzar un funcionamiento satisfactorio y fiable.

RECURSOS UTILIZADOS

Los recursos utilizados que han ayudado para el desarrollo y entendimiento del miniproyecto Hexavisión han sido:

- Software Arduino 1.8.33.0
- Documentación disponible en el Moodle de la asignatura.
- Referencias de páginas web (https://www.arduino.cc/).

REALIZACIÓN PRÁCTICA

El miniproyecto relacionado con la práctica 3 consiste en añadir una funcionalidad al Turnomatic de base, de nombre hexavision(), consistente en la visualización hexadecimal 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F en dígito de las unidades según la tecla o secuencia de teclas pulsadas. En el dígito de las unidades de millar aparece una "H" para indicar que el sistema está en la modalidad "hexavision". Esta funcionalidad se activa cuando se pulsa la secuencia de teclas #*, y se llama desde loop(). Para salir de la funcionalidad "hexavision", se ha de pulsar la secuencia de teclas *#.

Tecla	Display: Dígito millares	Display: Dígito unidades	Acción
#*	H	Apagado	Entrada en hexavision()
0	H	0	Visualiza dígitos
1	H	1	Visualiza dígitos
2	H	2	Visualiza dígitos
3	H	3	Visualiza dígitos
4	H	4	Visualiza dígitos
5	H	5	Visualiza dígitos
6	H	6	Visualiza dígitos
7	H	7	Visualiza dígitos
8	H	8	Visualiza dígitos
9	H	9	Visualiza dígitos
#0	H	A	Visualiza dígitos
#1	H	B	Visualiza dígitos
#2	H	C	Visualiza dígitos
#3	H	D	Visualiza dígitos
#4	H	E	Visualiza dígitos
#5	H	F	Visualiza dígitos
*#	apagado	Contador Turnomatic base	Salir de hexavision()

Para este proyecto, partiendo del código base del Turnomatic, he declarado un array que contiene todos los números para indicarle al PORTA qué leds encender. Una variable de tipo int que indicará la posición del array previamente declarado, y dos variables de checkeo que se pueden tratar de tipo booleano que nos avisan si se han pulsado las teclas '#' o '*'.

He creado la función **hexavision()**, cuyo funcionamiento principal es encender en el dígito de los millares una H, y luego encender en el dígito de las unidades el valor que se teclee. Para ello, hacemos uso del método *digitalWrite* para encender o apagar el pin seleccionado (unidades o millares) con el valor de PORTA.

En **loop()** he añadido el checkeo para poder entrar en el modo hexavision con una simple condición.

En el método **teclado()** es donde se encuentra el kit de la cuestión. Aquí es donde entra en funcionamiento las variables que entendemos como booleanas, donde se cambia el modo del Turnomatic para indicar que se quiere entrar en el modo hexavision, y donde la variable posición toma valores para indicarle a PORTA el valor que debe tomar. Para entrar al modo, se comprueba que se ha pulsado la secuencia de teclas #*, poniendo así modoTurnomatic a 0 y haciendo que en loop() se acceda a la función. Dependiendo de si se ha pulsado la tecla '#' previamente al tecleo de un número, la variable posición tomará un valor u otro para cumplir con las especificaciones del programa. De manera similar a entrar, para salir del programa se comprueba si

previamente se ha pulsado la tecla '*', y, si resulta ser así, cuando se pulsa '#' se vuelve al funcionamiento base del Turnomatic.

A continuación se muestra el código completo del programa.

```
// Declaración de variables

// Barrido del display-7seg y scanner del teclado (PORTL)
// Display (4 bits inferiores)
volatile int D4=  B11111110;  //PL0 D4-C1
volatile int D3=  B11111101;  //PL1 D3-C2
volatile int D2=  B11111011;  //PL2 D2-C3
volatile int D1=  B11110111;  //PL3 D1-nn

// Pulsadores pup, pdown, pleft, pright, penter y speaker(PORTC)
int pright = 30;  //PC7
int pdown = 31;   //PC6
int pleft = 32;   //PC5
int penter = 33;  //PC4
int pup = 34;     //PC3
                  //PC2
                  //PC1
int speaker =37;  //PC0

/*
// Teclado (4 bits superiores)
volatile int fila_R4=   B11100000;  //PL4 fila_R4
volatile int fila_R3=   B11010000;  //PL5 fila_R3
volatile int fila_R2=   B10110000;  //PL6 fila_R2
volatile int fila_R1=   B01110000;  //PL7 fila_R1

// Display de 7 segmentos (PORTA)
volatile int dp=29;  //PA7
volatile int g= 28;  //PA6
volatile int f= 27;  //PA5
volatile int e= 26;  //PA4
volatile int d= 25;  //PA3
volatile int c= 24;  //PA2
volatile int b= 23;  //PA1
volatile int a= 22;  //PA0
*/
```

```
// otras variables
volatile int  tecla_anterior=-1;
volatile int  tecla_actual=-1;
volatile boolean modoTurnomatic=1;
volatile int estado=0;
volatile int contador=0;
volatile int incremento=1;
volatile int unidades;
volatile int decenas;
volatile int val;

unsigned int fclk = 200; // 100 HZ, frecuencia del reloj de interrupción
unsigned int sound_base = 100;
unsigned int sound;
unsigned long duration = 200;

// código de 7-segmentos
byte tabla7seg[] = {63,06,91,79,102,109,125,39,127,103, 0,1, 2, 4, 8, 16, 32};
byte vsound[]={262,277,294,311,330,349,370,392,415,440};

/**
 * FUNCIONAMIENTO DE HEXAVISION
 */

// 63 = 3F, 91 = 5B, 79 = 4F, 66 = 102, 6D = 109, 7D = 125, 7F = 127, 67 = 103, 77 = 119,
// 7c = 124, 39 = 57, 9e = 158, 79 = 121, 71 = 113
byte hexarr[] = {63, 06, 91, 79, 102, 109, 125, 07, 127, 103, 119, 124, 57, 158, 121, 113};
int pos = 0;
// check 0 no se ha pulsado #, 1 cuando la acabo de pulsar
int check = 0;
// check2 0 no se ha pulsado *, 1 cuando la acabo de pulsar
int check2 = 0;
void hexavision(){
  if (pos == -1){
    PORTA = 0x76;
    digitalWrite(46, HIGH);
    delay(10);
    digitalWrite(46, LOW);
  } else {
    PORTA = 0x76;
    digitalWrite(46, HIGH);
    delay(10);
    digitalWrite(46, LOW);
    PORTA = hexarr[pos];
    digitalWrite(49, HIGH);
    delay(10);
    digitalWrite(49, LOW);
  }
}
```

```
void setup(){
  Serial.begin(9600);

  //PA7-PA0 (dpgfedcba) salidas --> display 7-seg
  DDRA = B11111111;

  // PC7-PC0 --> pulsadores entrada PC7-PC3 (entrada)) PC4-PC5 (free, entrada), PC0 speaker (salida)
  DDRC = B00000001;
  // activación pull-up en las líneas de entrada
  PORTC = B11111110;

  // PL7-PL0 --> barrido display PL4-PL0 (salida) y scanner teclado (PL7-PL4)
  DDRL = B00001111;
  // activación líneas de pull-up en las entradas de PORTL
  PORTL = B11111111;

  // Pin para conectar el altavoz
  pinMode(speaker,OUTPUT);

  // Inicialización de variables
  estado = 0;
  unidades = 0;
  decenas = 0;
  sound = sound_base;

  // Habilitación del TIMER1 para interrumpir cada 10ms (100Hz)
  // Funcionamiento normal... todo a cero
  // Disable interrupts
  cli();
  // modo normal de funcionamiento
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0; // cuenta inicial a cero
  // mode CTC
  TCCR1B |= (1<<WGM12);
  // prescaler  N = 1024
  TCCR1B |= (1<<CS12)|(1<<CS10);
  // fintr = fclk/(N*(OCR1A+1)) --> OCR1A = [fclk/(N*fintr)] - 1
  // para fintr = 100Hz --> OCR1A = [16*10^6/(1024*100)] -1 = 155,25 --> 155
  OCR1A = 155; // para 200 Hz  programar OCR1A = 77 (se ve mejor a 200 Hz!)

   // enable timer1 compare interrupt
   TIMSK1 |=(1<<OCIE1A);
   // habilitamos interrupciones
   sei();
 }


 void loop(){
   // Entrar al modo hexavision
   if (modoTurnomatic == 0) hexavision();
   // Programa principal de lectura de los pulsadores
   if (digitalRead(pup) == 0){
```

```
    // Incrementar
    incrementar();
    do
    {
      val = digitalRead(pup);
      if (digitalRead(pdown)==0) reset();
    } while (val == 0);
  }

    // decrementar
    if (digitalRead(pdown) == 0){
    // Decrementar
    decrementar();
    do
    {
      val = digitalRead(pdown);
      if (digitalRead(pup)==0) reset();
    } while (val==0);
  }
}
void incrementar()
{
      contador=decenas*10+unidades;
      contador=contador+incremento;
      if (contador >99)contador = contador-100;
      decenas = int(contador/10);
      unidades = contador % 10;
      beep();
}

void decrementar()
{
      contador=decenas*10+unidades;
      contador=contador-incremento;
      if (contador < 0)contador = contador+100;
      decenas = int(contador/10);
      unidades = contador % 10;
      beep();
}

void reset()
{
      unidades=0;
      decenas=0;
      beep();
}
```

```
// función de refresco de los displays
//void DisplayRefresh()
ISR(TIMER1_COMPA_vect)
{
  // display-off
  PORTL=PORTL|B00001111;
  switch(estado){
    case 0:

      // UNIDADES
      // Actualizamos portA con unidades
      PORTA = tabla7seg[unidades];
      // Activamos unidades en PORTL (D1D2D3D4)
      PORTL = D4;
      teclado(estado);
      estado=1;

      break;

    case 1:
      // DECENAS
     // Actualizamos portA con decenas
      PORTA = tabla7seg[decenas];
     // Activamos decenas en PORTL (D1D2D3D4)
      PORTL=D3;
      teclado(estado);
      estado=2;
      break;

    case 2:
      // solo barrido de la tercera columna del teclado (D1D2D3D4)
      PORTA = tabla7seg[10];
      PORTL=D2;
      teclado(estado);
      estado=0;
      break;

    case 3:
      // No hace falta. Necesario solo si el teclado tuviese una cuarta columna o si queremos usar el D1
      PORTL=D1;
      teclado(estado);
      estado=0;
      break;
  }
}

/**
 * Se añaden al método teclado funciones de hexavisión (cambios de la posición del array)
 */
void teclado(int estado){
      int teclau, tecla;
```

```
// Leemos teclado
teclau = (PINL & 0xF0)>>4; // leemos filas del teclado
if (teclau != 15) // hay pulsación y esperamos a que se suelte tecla
{
  while( ((PINL & 0xF0)>>4) !=15);
}


switch(estado){
 case 0:
     // Barrido de la primera columna del teclado
       switch (teclau) {
         case 7:
          if (modoTurnomatic == 0){
             if (check == 1) pos = 11;
             else pos = 1;
          }
          check = 0; check2 = 0;
          tecla = 1;
          break;
         case 11:
          if (modoTurnomatic == 0){
             if (check == 1) pos = 14;
             else pos = 4;
          }
          check = 0; check2 = 0;
          tecla = 4;
          break;
         case 13:
          if (modoTurnomatic == 0) pos = 7;
          check = 0; check2 = 0;
          tecla = 7;
          break;
         case 14:
          // Activar Hexavision, comprobar si hemos pulsado #
          if (check == 1){
             check = 0;
             modoTurnomatic = 0;
             pos = -1;
             // Si no, es que se quiere desactivar
           } else check2 = 1;
           // "*" 0x2A=42
           tecla=42;
           break;
          }
          break;
case 1:
```

```
        // barrido de la segunda columna del teclado
       switch (teclau) {
           case 7:
            if (modoTurnomatic == 0){
               if (check == 1) pos = 12;
               else pos = 2;
            }
            tecla = 2;
            break;
           case 11:
            if (modoTurnomatic == 0){
               if (check == 1) pos = 15;
               else pos = 5;
            }
            tecla = 5;
            break;
           case 13:
            if (modoTurnomatic == 0) pos = 8;
            tecla = 8;
            break;
           case 14:
            if (modoTurnomatic == 0) pos = 0;
            tecla = 0;
            break;
          }
         check = 0; check2 = 0;
         break;

case 2:
    // Barrido de la tercera columna
      switch (teclau) {
           case 7:
            if (modoTurnomatic == 0) pos = 3;
            tecla = 3;
            break;
           case 11:
            if (modoTurnomatic == 0) pos = 6;
            check = 0; check2 = 0;
           tecla = 6;
           break;
          case 13:
           if (modoTurnomatic == 0) pos = 9;
           check = 0; check2 = 0;
          tecla = 9;
          break;
```

```
              case 14:
                // Desactivar Hexavision, comprobamos si se ha pulsado *
                if (check2 == 1){
                  check2 = 0;
                  modoTurnomatic = 1;
                  // Si no, es que se quiere activar
                } else check = 1;
                // #
               tecla=0x23;
               break;
            }
          break;
  }
// si teclau != 15 es que se ha pulsado una tecla
    if(teclau !=15){
      tecla_anterior=tecla_actual;
      tecla_actual=tecla;
      //Serial.println(tecla);
    }


    if (modoTurnomatic == 1){
      // modo turnomatic
       if(tecla_anterior == '*' && (tecla_actual >=0 && tecla_actual<=9))
         { // cambiar frecuencia sonido
              sound = sound_base + tecla_actual * 400;
              if(tecla_actual == 0) sound=0;
              tecla_anterior=-1;
              tecla_actual=-1;
         }
       }
}


void beep(){

  //noTone(ptone);
  tone(speaker,sound,duration);

}
```

CONCLUSIONES & COMENTARIOS FINALES

Gracias a las tareas anteriores del miniproyecto ha sido mucho más llevadero y sencillo poder hacer la función, puesto que la mayoría de las actividades aquí propuestas las hicimos de una manera similar en las tareas. Por pesar de no tener una placa y no poder acceder a los laboratorios de la Universidad para poder llevar a cabo las pruebas y funcionamiento del programa, me es imposible otorgar el 100% de fiabilidad en decir que el programa funciona sin fallos, más allá de saber que no existen errores de compilación.

Creo que ante la grave situación y salto inminente a la docencia no presencial, se ha llevado bastante bien la adaptación a este mismo sistema de evaluación, y que las tareas respecto a la última práctica han sido variadas y desafiantes, ayudando al estudiantado conseguir el objetivo propuesto de obtener diferentes tipos de capacidades.