

## Aspectos de implementación para la práctica 1 (tareas no entregables)

Variables importantes a declarar:

Tabulación de los códigos de 7 segmentos para visualizar los diferentes caracteres:

**tabla\_7seg[] = {63, 06, 91, .....0x79, 0x71}; // cod. 7seg de: 0, 1 2 .....EF**

// Si hacemos **PORTA = tabla\_7seg[15]** colocaremos el código de siete segmentos de la 'F' en el puerto A. Si activamos el cátodo común de las unidades (0) entonces se visualizaría una F en el dígito de las unidades.

### Tarea 2: Transmisión serie.

Resumen: En el monitor serie (entorno Arduino) o en el Virtual Terminal (Proteus) debe aparecer un menú desde el cual se puede elegir la opción de parpadeo: unidades, decenas, etc...

Con la opción 6 del menú se selecciona el carácter a visualizar: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (código hexadecimal). El carácter inicial sería todos los segmentos encendidos. Una vez seleccionado el carácter, éste aparecerá en el display y luego con las otras opciones del menú se podrá hacer parpadear en las unidades, decenas, ....

Posible estructura del programa:

El menú lo podemos poner en setup():

```
// Envío del menu al monitor serie (virtual terminal)
Serial.println("1.- Parpadeo de unidades");
Serial.println("2.- Parpadeo de decenas");
Serial.println("3.- Parpadeo de centenas");
Serial.println("4.- Parpadeo de millares");
Serial.println("5.- Parpadeo secuencial de los digitos");
Serial.println("6.- Selección del carácter hexadecimal (0-F) que se visualiza en los dígitos");
Serial.println("Elija una opcion...");
```

En loop(), comprobamos si ha llegado algún carácter (opción de menú) por el canal serie provenientes del teclado del "Virtual Terminal". Si ha llegado, lo leemos y esa será la nueva opción (compruebe que la opción del menú está dentro del rango 1-6)

Variable "**displayChar**": carácter a visualizar: 0, 1, 2, 3, .... F

```
if(Serial.available()>0 ){
  opcion_old=opcion;
  opcion=Serial.read();
  opcion=opcion-0x30;
}

switch(opcion){

  case 1:
    // parpadeo de las unidades
    PORTA = tabla7seg[displayChar];
    for(i=0;i<=2;i++){
      PORTL = D4; // activamos dígito unidades
      delay(250);
      PORTL = D0; // desactivamos dígito unidades
      delay(250);
    }
    break;

  case 2:
    // parpadeo de las decenas
```

### Tarea 3: Contador de dos dígitos. Sincronización consulta de estado

El sistema lee los pulsadores constantemente para ver si se produce una pulsación (entrada =0). Cuando el pulsador está sin pulsar se lee un "1" por el pin, si se pulsa entonces entonces se lee un "0".

#### Antirrebote de pulsadores o teclas:

Cuando se pulsa un pulsador o una tecla se producen varias pulsaciones seguidas debido a que el contacto no es perfecto y se producen transiciones espurias hasta que finalmente se estabiliza en "0". De cada pulsación solo se debe tener en cuenta la primera ( $t=0$ ). Si cuando se explora de nuevo el pulsador se lee otro 0 entonces se mira cuán separado esta del primero (en tiempo). Si la diferencia de tiempo entre el primer 0 (el bueno) y el segundo 0 (posible ruido) es menos de 200 ms (parámetro ajustable, dependiente de la calidad del pulsador) entonces el segundo 0 no se tiene en cuenta.

Ejemplo de implementación para el pulsador PUP:

Declarar variables similares a:

```
time_now = 0L;           // importante que la variable sea de 32 bits para poder almacenar los milisegundos
time_old = 0L;

// exploración de pulsadores
// PUP
if (digitalRead(PUP) == 0){ // se ha pulsado PUP incremento del contador
    time_now = millis();
    if ((time_now - time_old) > 200) {contador = contador + incremento; time_old = time_now;}
    if(contador > cmax) contador = 0;
}
```

### Tarea 4: Visualización entrelazada y sincronizada por interrupción (INT3, pin18)

La parte de visualización o gestión del display se coloca en una rutina de servicio de interrupción (ISR) que se activa cada 5 ms. En cada interrupción se visualiza un dígito (uds, decs, cents o millares) del display de 7 segmentos.

Por tanto, cada 5 ms cambiamos de dígito (variable digit u otra)

digit = 0 --> se visualiza unidades

digit = 1 --> se visualiza decenas

digit = 2 --> se visualiza centenas

digit = 3 --> se visualiza millares

Cada cuando tiempo se refresca actualiza un dígito --> 4 dígitos x 5 ms/dígito = 20 ms

```

// GESTIÓN DEL DISPLAY EN UNA ISR()
// ISR para gestión del display: entrelazado, interrupción INT3, pin 18
// Esta función se ejecuta cada 5 ms (la INT3 interrumpe cada 5 ms)
ISR(INT3_vect){

    PORTL = D0FF;
    switch(digit){
        case 0:
            // visualización de UDS
            PORTA = tabla_7seg[contador % 10];
            PORTL = D4;
            teclado(digit);
            digit++;
            break;
        case 1:
            // visualización de DEC
            PORTA = tabla_7seg[int(contador/10) % 10];
            PORTL = D3;
            teclado(digit);
            digit++;
            break;
        case 2:
            PORTA = tabla_7seg[(int(contador/100)%10)];
            PORTL = D2;
            teclado(digit);
            digit = 0;
            break;
        case 3:
            PORTA = tabla_7seg[int(contador/1000)];
            PORTL = D1;
            digit = 0;
            break;
    }
} // ISR()

```