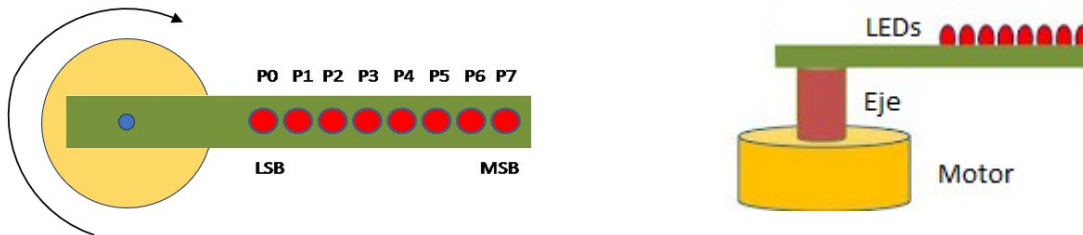


Los displays de LEDs giratorios (o de hélice) son un ingenioso dispositivo que utiliza la persistencia del ojo para “engañar” a nuestro cerebro, en forma similar a como lo hacen las pantallas de los televisores y monitores, a los efectos de desplegar textos simples.

En este caso se dispone de 8 LEDs colocados en un brazo (o hélice) que gira a una gran velocidad (3.000 rpm = revoluciones por minuto) respecto a un eje adosado a uno de sus extremos.



Prendiendo y apagando apropiadamente cada uno de los LEDs es posible “dibujar” letras en el círculo barrido por el brazo al girar.

Para ello se utilizan los dibujos matriciales de 8x5 puntos de los distintos caracteres, dejando una columna apagada entre dos caracteres consecutivos. Por ejemplo la letra A se dibuja:

MSB 7									
6									
5									
4									
3									
2									
1									
LSB 0									

El efecto logrado es el que se ve en la siguiente foto:



La empresa **PropelLED** nos ha encomendado la programación del controlador del display, responsable de permitir el ingreso del texto a desplegar y de encender y apagar apropiadamente los LEDs para lograr mostrar el texto ingresado, que tendrá un largo de 16 caracteres.

Se dispone de los siguientes elementos:

- un sensor del giro del motor que genera una interrupción por cada vuelta completa que da el brazo con los LEDs, invocando a la rutina de interrupción **giro()**.
- un puerto de E/S de solo escritura, de 8 bits, en la dirección **LEDS** que controla el encendido de los LEDs del brazo (el bit correspondiente en 1 significa LED encendido, si está en 0 el LED estará apagado).
- un timer, re-cargable, que genera una interrupción invocando a la rutina **timer()**, luego que pasa la

cantidad de tiempo que se escribe en el puerto de E/S, de solo escritura, de 16 bits, en la dirección **INTERVALO**. El valor de tiempo se expresa en microsegundos y se cuenta desde la última vez que se escribe un valor en el puerto, es decir que si se escribe un nuevo valor antes que se haya generado la interrupción del anterior, el tiempo empieza a contar nuevamente.

- un teclado que genera una interrupción por cada tecla digitada, invocando a la rutina **teclado()**, dejando en el puerto de E/S de solo lectura, de 8 bits, en la dirección **CHARACTER**, el código ASCII del carácter ingresado.

### Se pide:

Escribir en un lenguaje de alto nivel, preferentemente C, todas las rutinas para controlar el funcionamiento del display giratorio, teniendo en cuenta que:

- se debe siempre mostrar los últimos 16 caracteres ingresados. Entre el último carácter y el primero se dejarán cinco columnas apagadas en lugar de una.
- el motor gira a 3.000 rpm nominales, pero como es de baja calidad suele variar su velocidad. Por ello es necesario compensar ese efecto adecuando el intervalo del timer de modo de no se recorte el último carácter del texto (si el motor gira más lento) o que el espaciado entre el último y el primero sea más grande que el que especificado (si gira más rápido).
- se supone que se dispone de un arreglo bidimensional de bytes de nombre **matriz8x5[ascii][columna]** precargado con los dibujos de los caracteres ASCII en matrices de 8x5 puntos. Cada elemento del arreglo es una columna de la matriz de puntos.

**Solución:**

Vamos a usar el timer para generar una cadencia de interrupciones tal que nos permita en cada interrupción activar los LEDs que correspondan con una columna del carácter correspondiente. Pensemos que debemos desplegar todas las columnas correspondientes a las letras del texto y las columnas apagadas en cada vuelta de la hélice. El número total de columnas a considerar es:  $15 \times (5 + 1) + 1 \times (5 + 5) = 100$  columnas de los 16 caracteres con su respectivo espaciado, incluyendo el caso especial del último carácter (son 15 con 5 columnas de dibujo y una apagada y 1 con 5 de dibujo y 5 apagadas).

Algunas cuentas. Si gira a 3000 rpm, entonces da 3000 vueltas en 60 segundos, o sea una vuelta dura  $60 / 3000 = 1 / 50 = 20$  ms (milisegundos). En ese tiempo se deben desplegar las 100 columnas. Es decir se deben desplegar 100 columnas cada 20 ms, o sea que hay 0,2 ms (200 microsegundos) entre columnas.

Por tanto podemos programar que el timer interrumpa cada 200 us (es decir su período sea 200 us) y en cada interrupción dibujar una columna del texto. Ese tiempo entre columnas de 200 us se requiere ajustar (ya que puede variar por la mala calidad del motor). Para hacerlo vamos a contar cuantas columnas podemos dibujar en un giro y si son menos de 100 vamos a disminuir el valor del período (para que haya más interrupciones) y si son más vamos a aumentarlo.

Por su parte en la rutina de interrupción del teclado vamos a mover el texto un lugar a la izquierda (hacia las posiciones más bajas) y vamos a ingresar el nuevo carácter en la última posición.

```
char texto[16];
short periodo, columna;

void interrupt teclado() {
    for (short i = 1; i < 16; i++) texto[i - 1] = texto[i];
    texto[15] = in(CARACTER);
}

void interrupt timer() {
    if ((columna < 96) && (columna % 6 != 5)) {
        if out(LEDS, matriz8x5[texto[columna / 6]][columna % 6]);
    }
    else {
        out(LEDS, 0);
    }
    columna ++
}

void interrupt giro() {
    if (columna > 100) periodo++;
    if (columna < 100) periodo--;
    out(INTERVALO, periodo);
    columna = 0;
}
```

```
void main() {
    // instalo rutinas de interrupciones
    for (short i = 0; i < 16; i++) texto[i] = ' ';
    periodo = 200;
    out(INTERVALO, periodo);
    columna = 0;
    enable();
    while(TRUE);
}
```