

```
#include <16F877a.h> /*Se incluye la librería del
dispositivo*/
#fuses XT,PUT,NOWDT /*Aqui se configura tales como el
perro guardian y, el tipo de reloj*/
#use delay (clock=4M) /*Configuración de la velocidad del
reloj*/
```

```
#byte port_a=0x05 /**/
#byte port_b=0x06 /**/
#byte port_c=0x07
#byte port_d=0x08
#byte port_e=0x09
```

```
#byte entrada=0xff
#byte salida=0x00
#byte apagado=0x00
#byte encendido=0xff
```

```
//Variables Glovales
int horas=0; //Variables para contar
int minutos=0;
int segundos=0;
```

```
int uMinutos=0; //Variables para visualizar
int dMinutos=0;
int uHoras=0;
int dHoras=0;
```

```
//Funciones Auxiliares
void demoraryPintar(int limite); //Cuenta en
segundos y pinta
void contar(); //Reloj
void cronometro(); //Cronómetro
int Escribirdisplay(int digito);//
void actualizar(); //Actualiza Cronometro
void modificar(); //En que minuto u hora empieza.
int separarU(int numero);
int separarD(int numero);
```

```
//Función Principal
void main(){
set_tris_a(entrada);
set_tris_b(salida);
set_tris_c(salida);
set_tris_d(salida);
set_tris_e(salida);
```

```
port_b=(apagado);
port_c=(apagado);
port_d=(apagado);
port_e=(apagado);
```

```
delay_ms(1000);
```

```
while(true){ //Unicio de ciclo repetitivo
```

```
/*switch(port_a){
case 0: //Cómo reloj
```

```
demoraryPintar(20);
contar();
Actualizar();
break;
```

```
case 1: //cómo cronometro regresivo.
demoraryPintar(20);
cronometro();
Actualizar();
if(minutos==0&&horas==0){
while(port_a==1){
port_d=encendido;
}
port_d=apagado;
}
break;
```

```
case 2:
modificar();
actualizar();
demoraryPintar(1);
break;
default:
break;
```

```
}*/
```

```
if(input(pin_a0)){
demoraryPintar(20);
contar();
Actualizar();
```

```
else{
demoraryPintar(20);
cronometro();
Actualizar();
```

```
void modificar(){
if(input(pin_a2)==1){
minutos++;
}
else if(input(pin_a3==1)){
horas++;
}
}
```

```
void actualizar(){
uMinutos=Escribirdisplay(separarU(minutos));
dMinutos=Escribirdisplay(separarD(minutos));
uHoras=127-Escribirdisplay(separarU(horas));
dHoras=127-Escribirdisplay(separarD(horas));
}
```

```
void contar(){
segundos=segundos+1;
if(segundos==60){
segundos=0;
```

```

        minutos=minutos+1;
        if(minutos==60){
            minutos=0;
            horas=horas+1;
            if(horas==24){
                horas=0;
            }
        }
    }

void cronometro(){ //minutos=horas
segundos=minutos;
    minutos=minutos+1;
    if(minutos==60){
        minutos=0;
        horas=horas+1;
        if(horas==24){
            horas=0;
        }
    }
}

void demoraryPintar(int limite){ //1S = 1000mS =
20*2*25mS
    int i;
    for(i=0; i<limite; i++){
        port_b=uMinutos;
        port_c=dMinutos;
        output_bit(pin_e0, 0); //activa salida de minutos
        delay_ms(25);
        port_b=uHoras;
        port_c=dHoras;
        output_bit(pin_e0, 1); //Activa las horas
        delay_ms(25);
    }
    output_toggle(pin_e1); // salida en segundos
}

int Escribirdisplay(int selector){ //retorna datos de decimal a
display 7 segmentos.
    switch(selector){
        case 0:
            return 126;
        case 1:
            return 48;
        case 2:
            return 109;
        case 3:
            return 121;
        case 4:
            return 51;
        case 5:
            return 91;
        case 6:
            return 95;
        case 7:
            return 112;

```

```

        case 8:
            return 127;
        case 9:
            return 123;
        default:
            break;
    }
}

int separarU(int numero){ //Módulo"residuo"
    return numero%10;
}

int separarD(int numero){ //Cociente
    return numero/10;
}

```

---

```

#include <16F877a.h> /*Se incluye la librería del
dispositivo*/
#fuses XT,PUT,NOWDT /*Aqui se configura tales como el
perro guardian y, el tipo de reloj*/
#use delay (clock=4M) /*Configuración de la velocidad del
reloj*/

```

```

#byte port_b=0x06 /**/
#byte port_a=0x05 /**/
#byte port_c=0x07 /**/
#byte port_d=0x08 /**/
#byte port_e=0x09 /**/

```

```

#byte entrada=0xff
#byte salida=0x00
#byte apagado=0x00

```

```

//funciones auxiliares
void demoraS(int segundos){ //Se ingresa un valor en
en segundos y retorna la demora
    int entero=segundos;
    int i;
    for(i=0; i<entero; i++){
        delay_ms(1000);
    }
}

```

```

void MoverIzquierda(){ //Enciende los leds hacia la
izquierda

```

```

    port_b=0b00000001;
    demoraS(1);

```

```

    port_b=0b00000010;
    demoraS(1);

```

```

    port_b=0b00000100;
    demoraS(1);

```

```

    port_b=0b00001000;

```

```

    demoraS(1);

    port_b=0b00010000;
    demoraS(1);

    port_b=0b00100000;
    demoraS(1);

    port_b=0b01000000;
    demoraS(1);
}

void MoverDerecha(){ //Enciende los leds hacia la
derecha

    port_b=0b10000000;
    demoraS(1);

    port_b=0b01000000;
    demoraS(1);

    port_b=0b00100000;
    demoraS(1);

    port_b=0b00010000;
    demoraS(1);

    port_b=0b00001000;
    demoraS(1);

    port_b=0b00000100;
    demoraS(1);

    port_b=0b00000010;
    demoraS(1);
}

int main(){
    set_tris_b(salida); //define el puerto B
    como salida

    port_b=apagado;
    while(true){ //inicia ciclo repetitivo
        MoverIzquierda();
        MoverDerecha();
    }
    return 0;
}

#include <16F628a.h> /*Se incluye la librería del
dispositivo*/
#fuses XT,PUT,NOWDT /*Aqui se configura tales como el
perro guardian y, el tipo de reloj*/
#use delay (clock=4M) /*Configuración de la velocidad del
reloj*/

```

```

#byte port_a=0x05 /**/
#byte port_b=0x06 /**/

#byte entrada=0xff
#byte salida=0x00
#byte apagado=0x00

void demoraS(int segundos){ //Se ingresa un valor en
en segundos y retorna la demora
    int entero=segundos;
    int i;
    for(i=0; i<entero; i++){
        delay_ms(1000);
    }
}

void MoverDerecha(){ //Enciende los leds hacia la
derecha

    port_b=0b00000000; //0
    demoraS(1);

    port_b=0b10000000; //1
    demoraS(1);

    port_b=0b01000000; //2
    demoraS(1);

    port_b=0b00100000; //3
    demoraS(1);

    port_b=0b00010000; //4
    demoraS(1);

    port_b=0b00001000; //5
    demoraS(1);

    port_b=0b00000100; //6
    demoraS(1);

    port_b=0b00000010; //7
    demoraS(1);
}

int main(){
    set_tris_b(salida); //define el puerto B
    como salida

    port_b=apagado;
    while(true){ //inicia ciclo repetitivo
        MoverDerecha();
        port_b=0b11111111; //Motobomba encendida
        demoraS(10);
    }
    return 0;
}

```