

```
/* Programa PastilleroV10.ino
   Controla la operación del pastillero.
   Contiene, además de las funciones setup() y loop(),
   las funciones ConversionHminMs(), ConversionMsHmin(),
   ActivaMotor(), Alarma(), Despliegue(),
   DespliegueHoraToma() e Interruptor(). */

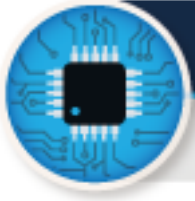
// Variables Alarma
int pinBuzzer = 15;
int pinLed1 = 16;
int pinLed2 = 17;
int pinBoton = 2;
boolean banderaAlarma = HIGH;
int rep;
int numRep = 10;
int tiempoEnc = 600;
int tiempoApag = 400;
int tiempoEspera = 9900;

// Variables ActivaMotor
int pinIntLamina = 5;
int pinIN1 = 7;
int pinIN2 = 8;
int pinPWM = 9;
boolean estadoIntLamina;
int valorPWM = 230;
long tiempoActivacion[5];
int toma = 0;

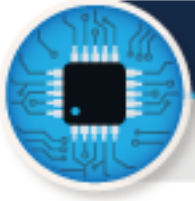
// Variables Despliegue
long tiempoMs;
long tiempoBase;
long tiempoActual;
long hora;
long minuto;
long horaActual;
long minActual;
int tiempoRet = 19000;

// Se incluye el código de la biblioteca del LCD
#include <LiquidCrystal.h>

// Inicializa la biblioteca con el número de los pines
LiquidCrystal lcd(4, 6, 11, 12, 13, 14);
```



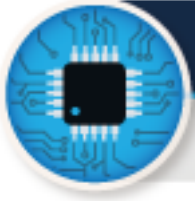
```
// Función setup
void setup()
{
  // Hora de reinicio
  hora = 8;
  minuto = 39;
  // Cálculo tiempo base
  ConversionHminMs();
  tiempoBase = tiempoMs;
  // Configuración del número de columnas y renglones del LCD
  lcd.begin(16,2);
  // Configuración terminales buzzer, led y botón alarma
  pinMode(pinBuzzer, OUTPUT);
  pinMode(pinLed1, OUTPUT);
  pinMode(pinLed2, OUTPUT);
  pinMode(pinBoton, INPUT);
  attachInterrupt(0, Interruptor, RISING);
  // Configuración terminales del puente H y del interruptor
  pinMode(pinIN1, OUTPUT);
  pinMode(pinIN2, OUTPUT);
  pinMode(pinIntLamina, INPUT);
  // Cálculo tiempos de activación del motor
  // Hora de la primera toma
  hora = 9;
  minuto = 00;
  ConversionHminMs();
  tiempoActivacion[0] = tiempoMs;
  // Hora de la segunda toma
  hora = 12;
  minuto = 30;
  ConversionHminMs();
  tiempoActivacion[1] = tiempoMs;
  // Hora de la tercera toma
  hora = 15;
  minuto = 00;
  ConversionHminMs();
  tiempoActivacion[2] = tiempoMs;
  // Hora de la cuarta toma
  hora = 18;
  minuto = 45;
  ConversionHminMs();
}
```



```
    tiempoActivacion[3] = tiempoMs;
// En caso de que haya más tomas, agregarlas a continuación:

// Hora final
hora = 23;
minuto = 59;
ConversionHminMs();
tiempoActivacion[4] = tiempoMs;
}

// Función loop
void loop()
{
// Envío señales iniciales al puente H: motor detenido
digitalWrite(pinIN1, 0);
digitalWrite(pinIN2, 0);
analogWrite(pinPWM, valorPWM);
Despliegue();
delay(tiempoRet);
/* Mientras el tiempoActual sea menor que el tiempoActivacion[toma]
+ tiempoBase, sólo despliega la hora */
do
{
    Despliegue();
    delay(tiempoRet);
    tiempoActual = millis() + tiempoBase;
} while(tiempoActual < tiempoActivacion[toma]);
/* Cuando el tiempoActual es mayor que el tiempoActivacion[toma]
se sale de la estructura do-while, activa el motor, despliega la
hora de toma y enciende la alarma */
ActivaMotor();
banderaAlarma = HIGH;
do
{
// Despliegue de la hora de toma
DespliegueHoraToma();
Alarma();
} while(banderaAlarma == HIGH);
/* La alarma se apagará cuando se oprima el botón de apagado, el
cual hará que la variable banderaAlarma tome el valor LOW, de
manera que se saldrá de la estructura do-while anterior */
// Borra lcd
lcd.clear();
toma++;
}
```

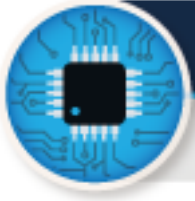


```
}

/* Función ConversionHminMs
   Convierte el tiempo en horas y minutos a milisegundos */
void ConversionHminMs()
{
    minuto = minuto + hora*60;
    tiempoMs = minuto*60000;
}

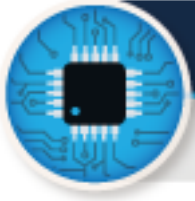
/* Función ConversionMsHmin
   Convierte el tiempo en milisegundos a horas y minutos */
void ConversionMsHmin()
{
    horaActual = tiempoActual/60/60000;
    minActual = tiempoActual/60000 - horaActual*60;
}

/* Función Despliegue
   Despliega en un LCD la hora, con formato hora:minutos 24 h
   * Pin RS del LCD al pin 4
   * Pin E del LCD al pin 6
   * Pin D4 del LCD al pin 11
   * Pin D5 del LCD al pin 12
   * Pin D6 del LCD al pin 13
   * Pin D7 del LCD al pin 14
   * Pin R/W del LCD a tierra */
void Despliegue()
{
    // Calcula el tiempoActual con respecto al tiempoBase
    tiempoActual = millis() + tiempoBase;
    // Conversión del tiempoActual a horas:minutos
    ConversionMsHmin();
    // Se inicializa el LCD
    lcd.begin(16,2);
    // Se coloca el cursor en el origen
    lcd.setCursor(0,0);
    // Limpia el primer renglón
    lcd.print("          ");
    // Se coloca el cursor en el origen
    lcd.setCursor(0,0);
    // Escribe el letrero Hora
    lcd.print("Hora ");
    // Escribe horas:minutos
```



```
lcd.print(horaActual);  
lcd.print(":");  
lcd.print(minActual);  
}
```

```
/* Función Activa Motor  
   Controla el motor de CD del pastillero, con base en  
   la señal de un interruptor de lámina activa alta */  
void ActivaMotor()  
{  
  do  
  {  
    digitalWrite(pinIN1, 1);  
    estadoIntLamina = digitalRead(pinIntLamina);  
  } while(estadoIntLamina == HIGH);  
  delay(200);  
  do  
  {  
    digitalWrite(pinIN1, 1);  
    estadoIntLamina = digitalRead(pinIntLamina);  
  } while(estadoIntLamina == LOW);  
  digitalWrite(pinIN1, 0);  
}  
  
/* Función Alarma  
   Hace sonar intermitentemente un buzzer, y al mismo tiempo  
   prende un led intermitentemente, hasta que se oprime el  
   botón de apagado de la alarma */  
void Alarma()  
{  
  for (rep = 0; rep < numRep; rep++)  
  {  
    digitalWrite(pinBuzzer, HIGH);  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, HIGH);  
    delay(tiempoEnc);  
    digitalWrite(pinBuzzer, LOW);  
  }  
}
```



```
        digitalWrite(pinLed1, LOW);
        digitalWrite(pinLed2, LOW);
        delay(tiempoApag);
    }
    delay(tiempoEspera);
}

/* Función Interruptor
   Rutina de servicio de interrupción */
void Interruptor()
{
    banderaAlarma = LOW;
}

/* Función DespliegueHoraToma
   Despliega en el segundo renglón del LCD la hora de la toma de
   las pastillas, indicando también a qué toma corresponde */
void DespliegueHoraToma()
{
    Despliegue();
    // Calcula el tiempo de la toma con respecto al tiempoBase
    tiempoActual = tiempoActivacion[toma];
    // Conversión del tiempoActual a horas:minutos
    ConversionMsHmin();
    // Se coloca el cursor en el origen del segundo renglón
    lcd.setCursor(0,1);
    // Limpia el segundo renglón
    lcd.print(" ");
    // Se coloca el cursor en el origen
    lcd.setCursor(0,1);
    // Escribe el número de la toma
    lcd.print(toma+1);
    // Escribe el letrero "a toma "
    lcd.print("a toma ");
    // Escribe horas:minutos
    lcd.print(horaActual);
    lcd.print(":");
    lcd.print(minActual);
}
```