El paso del tiempo (II)

por Aurelio Gallardo

8 de Febrero de 2013

Construyendo un cronómetro con ARDUINO...

Índice General

Apartado I

Introducción

Como comentamos en la primera parte, dedicada al diseño y construcción de un reloj digital, con pocas modificaciones de hardware, podemos construir un cronómetro a partir del montaje del mismo. Se conserva el set de 4 displays de 7 segmentos, y los tres botones, y se modifica el programa casi al completo.

Ambos dispositivos, reloj y cronómetro, se fundamentan en dos ideas distintas de medir el tiempo. El reloj se construye para ser cíclico, para llevar una cuenta que se repite de forma indefinida; una cuenta externa a él, independiente (en principio) de él mismo. Si el reloj se estropease, esa cuenta del tiempo seguiría: sería posible continuarla en otro reloj. El cronómetro, sin embargo, lleva una cuenta que tiene principio y fin. No es cíclico. Empieza en un punto y acaba en otro.

Apartado II

Montaje

El montaje varía muy poco respecto del anterior. Tan sólo dos modificaciones:

1. Retirada del buzzer en el pin 1.
2. En su lugar, conectamos el pin 1 de ARDUINO a la patilla 7 del set de 4 displays. En dicha patilla encontramos el punto decimal, que puede situarse en cualquiera de los 4 displays.

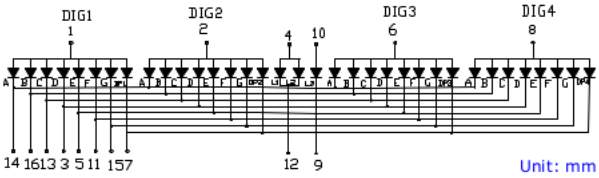

Figura 1: Patillaje del set de 4 displays

Apartado III

Explicación del código

NOTA: El código que se repite respecto al reloj digital no se vuelve a explicar.

Variables

```
int outcoma = 1;

// defino RESTART con el codigo ensamblador que hace un RESET
```

```
#define RESTART asm("jmp 0x0000");
.
.
.
// entradas para modificar la hora y minuto
int start = 0; // 1 --> empieza , 0--> para
int reset = 0; // 0 --> continua, 1 --> hace un reset
int mode = 0; // 0 --> modo segundos + decimas  1 --> modo minutos + segundos

// variables de entrada analogica
int instart = 0;
int inreset = 1;
int inmode = 2;

// valores a pasar al display, y calculos
int h = 0;
int m = 0;
int hora = 0;
int minutos = 0;
int segundos = 0;
int mediosegundo = 0;
int valorcoma = 2; // display2, o sea el tercero
.
.
.
//contador general de tiempo
unsigned long tiempo1=0;
unsigned long tiempo2=0;
unsigned long tiempo3=0;
.
.
.
```

- En primer lugar defino el pin de salida 1 como outcoma, ya que será la variable que guarde la información de la coma digital.
- Defino como RESTART a la llamada en ensamblador que reinicia el programa. Es la mejor manera de hacer un RESET o puesta a cero programable. Más información en la siguiente discusión de los foros de ARDUINO: http://arduino.cc/forum/index.php?topic=50803.5;wap2
- Las variables start, reset y mode tienen las siguientes funciones:
  - start: comienza el cronómetro. Si se vuelve a pulsar, para (stop). Al volverlo a poner en marcha se reanuda la cuenta.
  - reset: puesta a cero del cronómetro.
  - mode: son posibles 4 visualizaciones:
    - ⇒ Segundos y décimas (mode=0). Cuenta hasta 999,9 segundos. Punto decimal en el tercer dígito o display número 2.
    - ⇒ Segundos (mode=1): cuenta hasta 9999 segundos. Punto decimal en el cuarto dígito o display número 3.
    - ⇒ Minutos y segundos (mode=2): cuenta hasta 99' 59". Punto decimal en el segundo dígito o display número 1.
    - ⇒ Horas y minutos (mode=3): cuenta hasta 99 horas y 59'. Dos puntos centrales (muestra segundos).
- Cambio el nombre de las variables que detectan las pulsaciones de las entradas analógicas: instart, inreset e inmode.

- Variables h y m: valores respectivamente de los dos primeros dígitos del display (h) , y de los otros dos (m).
- Variables hora, minutos, segundos...: controlan los segundos-minutos-horas que trasncurren.
- Variable mediosegundo: controla el parpadeo de los puntos centrales.
- Variable valorcoma: indica en qué display hay que activar el punto decimal.
- Variables tiempo1, tiempo2, tiempo3: variables que llevan la cuenta del tiempo.

Rutina "mandasenal"

```
void mandasenal(int numdisplay, int valor) {
.
.
.
// ********************************
// b) activo un solo display cada vez
// ********************************
for (int i=0;i<=3;i++) { // recorrido de 0 hasta 3

if (numdisplay==valorcoma) {
  digitalWrite(outcoma,LOW);
} else {
  digitalWrite(outcoma,HIGH);
}

// elige el nodo
if (i==numdisplay) {
digitalWrite(anodo[i],HIGH); // activo por
} else {
digitalWrite(anodo[i],LOW); // desactivo por
} // fin del if
} // fin del for
}
```

Esta rutina prácticamente se deja como está. El único cambio radica en el ***if*** que detecta si el numdisplay es el mismo que valorcoma, para poder activar o no el punto decimal.

Rutina "valordisplay"

El cambio añadido a esta rutina es simple: las variables h y m han pasado a ser variables globales y no locales de la función, por lo que no se pasan como parámetros a la misma. El código de la rutina, sin embargo, no ha cambiado.

Otros cambios

- Se retira la funcion calculamuestratiempo
- Se añade en la rutina ***setup*** la orden **pinMode(outcoma, OUTPUT)**;

Rutina principal

Cálculo del tiempo transcurrido

```
tiempo1=millis(); // milisegundostranscurridos al empezar el bucle

if (start==1) {
  tiempo3=(tiempo1–tiempo2)/100; // decimas de segundo
} else {
  tiempo2=tiempo1–(tiempo3*100);
}
```

1. La variable tiempo1 almacena la salida de la función millis(). No cambia respecto al reloj digital.
2. La variable tiempo3 almacena el tiempo en el que el botón start se activa como “1”. Esta variable es la base de tiempo para cálculos posteriores.
3. La variable tiempo2 almacena el tiempo en el que tenemos estado de parada (start -> ”0”).

Por lo tanto, al tener activado “start”, se cuenta el tiempo que ha transcurrido menos el que se haya parado. Si “start” está desactivado, ocurre lo contrario: lo que se cuenta es el tiempo de parada. El desborde de la variable tiempo1 (a los 50 días) nunca ocurre, ya que el cronómetro, como máximo, alcanza las 99 horas y 59 minutos (algo más de 4 días).

Cómo se realizan cálculos con el paso del tiempo

```
// calculo y mostrar
if (mode==0) {
  h = tiempo3 /100; // centenas – decenas segundos
  m = tiempo3 %100; // unidades y decimas de segundo
  valorcoma = 2; // tercer display
  if (tiempo3>9999) { // caso que sobrepase la cantidad
     mode=2; // cambia al modo 2.
  }
} else if (mode==1) {
  // segundos sin decimas
  segundos = tiempo3 / 10; // calculo los segundos transcurridos, sin decimas
  h = segundos /100 ;
  m = segundos %100;
  valorcoma = 3; // la coma en el ultimo digito
  if (segundos>9999) { // caso que sobrepase la cantidad
     mode=3; // cambia al modo 3.
  }
} else if (mode==2) {
  // minutos – segundos
  segundos = tiempo3 / 10; // calculo los segundos transcurridos
  minutos = segundos / 60;
  segundos = segundos %60;
  h = minutos;
  m = segundos;
  valorcoma=1; // la coma en la segunda posicion
  if (segundos>5999) { // caso que sobrepase la cantidad
     mode=3; // cambia al modo 3.
  }
} else if (mode==3) {
```

```
  // horas – minutos
  segundos = tiempo3 / 10; // calculo los segundos transcurridos
  mediosegundo = segundos %2;
  minutos = segundos / 60;
  hora = minutos / 60 ;
  // hora = hora %24 NO, no es un reloj
  minutos = minutos %60;
  h = hora;
 m = minutos;
  digitalWrite(outmediosegundo, mediosegundo); // la coma en la segunda posicion,
     parpadeante
  valorcoma = 9; // este valor no existe, y no pondr como ninguna
  if (hora*3600+minutos*60>359940) {
    RESTART;
  }
}

// anula el parpadeo de leds centrales en caso de mode distinto de 3
if (mode!=3) {
  digitalWrite(outmediosegundo, HIGH);
}

// anula el parpadeo de leds centrales en caso de mode distinto de 3
if (mode!=3) {
  digitalWrite(outmediosegundo, HIGH);
}

  valordisplay(LOW,LOW);
  cualactivo = (cualactivo+1)*(cualactivo<3);
  mandasenal(cualactivo,displays[cualactivo]);
```

Depende del valor que tome la variable mode:

1. Modo décimas de segundo (mode=0): en este caso, tiempo3 ya viene dado en décimas de segundo. Divido entre 100 y se lo aplico a "h". El módulo de dividir entre 100 a "m", y pasaré esos valores a la variable displays cuando llame a valordisplay. Como en el caso del reloj, tengo que elegir cíclicamente un display a mostrar, con la variable cualactivo y mostrarla con mandasenal.

   a) mode=0 se desborda a los 999,9 segundos (o tiempo3=9999 décimas de segundo) por lo que fuerzo el cambio de modo a modo minutos - segundos.

   b) la coma de los decimales se aplica al tercer dígito con valorcoma = 2.

2. Modo segundos (mode=1): convierto tiempo3 en segundos y aplico lo mismo que el caso anterior a la variable "segundos".

   a) mode = 1 se desborda a los 9999 segundos, y fuerzo el cambio de modo a horas - minutos.

   b) la coma de los decimales se aplica al cuarto dígito con valorcoma = 3.

3. Modo minutos - segundos (mode=2): segundos se convierten en minutos - segundos. "h" pasa a ser los minutos y "m" los segundos.

   a) mode = 2 se desborda a los 99' 59" (o sea, los 5999 segundos). Se fuerza el cambio a horas - minutos.

   b) La coma decimal se aplica al segundo dígito.

4. Modo horas - minutos (mode=3): convertimos a horas y minutos, calculamos los medios segundos y activamos los leds centrales del display.

a) Al desbordarse (a las 99 horas y 59', los 359940") fuerzo el reinicio. Creo que es un límite más que aceptable para un cronómetro. Por eso nunca desbordaré la variable tiempo1, capaz de alcanzar los 50 días aproximadamente.

b) La coma decimal toma un valor imposible; así no aparece en ninguno de los dígitos.

Por último, solo indicar que los leds centrales se desconectan si no estamos en el último modo, el modo horas - minutos.

NOTA: es necesario investigar aún más detenidamente las rutinas que detectan el desbordamiento del conteo en cada modo y el salto a otro.

Activación de los botones

```
// 1) comprueba start
if (analogRead(instart)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(instart)>500) {
   start=!start; // cambio la condicion de start
}
}

// 2) comprueba reset
if (analogRead(inreset)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(inreset)>500) {
   start=0;
   RESTART; // hago un RESET en caliente
}
}

// 3) comprueba modo
if (analogRead(inmode)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(inmode)>500) {
   mode=(mode+1)*(mode<3);
}
}
```

1. En el primero, simplemente se complementa el valor de "start", pasando de 0 a 1 alternativamente. Esto parará la cuenta o no.
2. En el segundo paro la cuenta y hago un reinicio del programa, mediante la llamada RESTART.
3. Simplemente voy modificando la variable modo haciendo pasar todos los valores entre 0 y 3.

En todas las llamadas, hay un retraso y segunda lectura del botón que evita lecturas espurias (ver primera parte: reloj digital) y asegura que ha sido pulsado de forma correcta. Esto produce un leve parpadeo del display. Es quizás un efecto no deseado, pero disminuir el retraso implica que los botones no tengan lecturas correctas.

delay(5)

```
delay(5);
```

Sin esta sentencia final, hay efectos parásitos en la iluminación de los leds de los displays y, aunque se puede ver la cuenta del tiempo, la lectura del display no es cómoda. No estoy muy seguro del por qué sucede. Supongo que un cambio tan rápido (en el modo décimas de segundo) de los ánodos implica que haya algo de tensión residual en ellos, lo suficiente como para que se produzca una iluminación tenue en los segmentos de los displays que deberían aparecer como apagados. O bien puede haber algún efecto capacitivo asociado a los leds en el display.

Lo cierto es que, manteniendo cada ánodo encendido al menos 5 ms, consigo una iluminación correcta.

Apartado IV

Fotografías

El vídeo del cronómetro en funcionamiento puesdes encontralo en http://youtu.be/Ac8foOlST3k

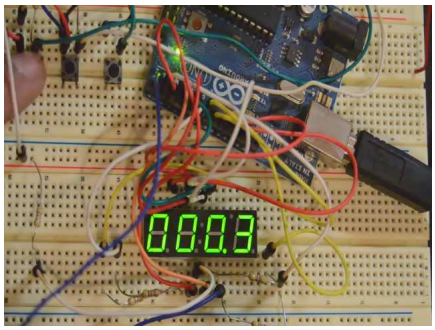

Figura 2: Puesta en marcha

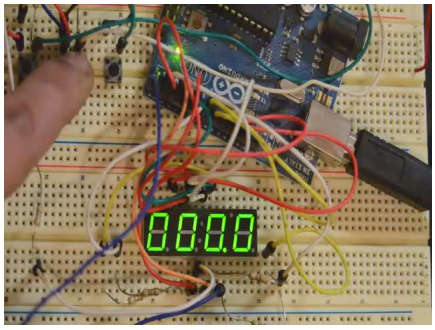

Figura 3: Reset

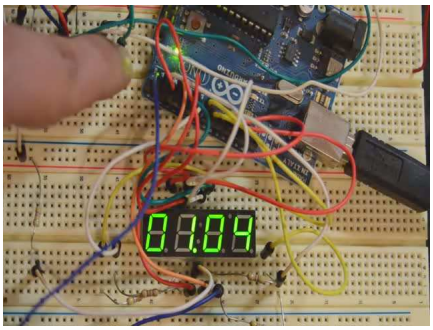
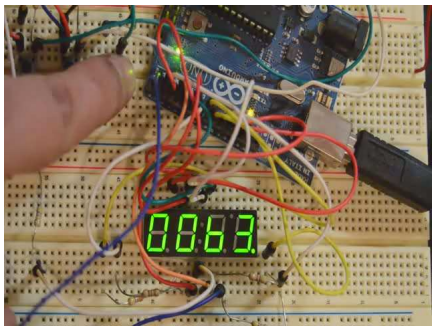
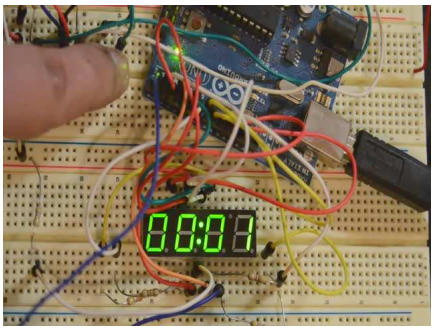

Figura 4: Modos de funcionamiento

Apartado V

Anexo I: código del programa "cronómetro"

```
// siete salidas digitales a cada tramo del display
int outa = 11;
int outb = 13;
int outc = 2;
int outd = 3;
int oute = 4;
int outf = 5;
int outg = 6;
int outmediosegundo = 12;
int outcoma = 1;

// defino RESTART con el codigo ensamblador que hace un RESET
#define RESTART asm("jmp 0x0000");

// a los anodos de los displays. Actuan por logica inversa
int anodo[4]={7,8,9,10};
// entradas para modificar la hora y minuto
int start = 0; // 1 --> empieza , 0--> para
int reset = 0; // 0 --> continua, 1 --> hace un reset
int mode = 0; // 0 --> modo segundos + decimas  1 --> modo minutos + segundos

// variables de entrada analogica
int instart = 0;
int inreset = 1;
int inmode = 2;

// valores a pasar al display, y calculos
int h = 0;
int m = 0;
int hora = 0;
int minutos = 0;
int segundos = 0;
int mediosegundo = 0;
int valorcoma = 2; // display2

// variables de los displays
unsigned long displays[4]; // variable para cada display
int cualactivo = 0 ; //variable para guardar cual se activa

//contador general de tiempo
unsigned long tiempo1=0;
unsigned long tiempo2=0;
unsigned long tiempo3=0;
int retrasoboton = 100; //retraso del botn al pulsar

// contador de proposito general
int i=0;

/* ====================== */
/* Funciones y subrutinas */
/* ====================== */
```

```
// ************************************************
// Funcion que pone un valor en un display a la vez
// ************************************************

void mandasenal(int numdisplay, int valor) {
// ****************************************
// a) activa cada tramo del display segun valor
// ****************************************
  // recibe el numero de display y el valor que tiene que entregar
switch (valor) {
case 0:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,LOW);
digitalWrite(oute,LOW);
digitalWrite(outf,LOW);
digitalWrite(outg,HIGH);
break;
break;
case 1:
digitalWrite(outa,HIGH);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,HIGH);
digitalWrite(oute,HIGH);
digitalWrite(outf,HIGH);
digitalWrite(outg,HIGH);
break;
case 2:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,HIGH);
digitalWrite(outd,LOW);
digitalWrite(oute,LOW);
digitalWrite(outf,HIGH);
digitalWrite(outg,LOW);
break;
case 3:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,LOW);
digitalWrite(oute,HIGH);
digitalWrite(outf,HIGH);
digitalWrite(outg,LOW);
break;
case 4:
digitalWrite(outa,HIGH);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,HIGH);
digitalWrite(oute,HIGH);
digitalWrite(outf,LOW);
digitalWrite(outg,LOW);
```

```
break;
case 5:
digitalWrite(outa,LOW);
digitalWrite(outb,HIGH);
digitalWrite(outc,LOW);
digitalWrite(outd,LOW);
digitalWrite(oute,HIGH);
digitalWrite(outf,LOW);
digitalWrite(outg,LOW);
break;
case 6:
digitalWrite(outa,HIGH);
digitalWrite(outb,HIGH);
digitalWrite(outc,LOW);
digitalWrite(outd,LOW);
digitalWrite(oute,LOW);
digitalWrite(outf,LOW);
digitalWrite(outg,LOW);
break;
case 7:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,HIGH);
digitalWrite(oute,HIGH);
digitalWrite(outf,HIGH);
digitalWrite(outg,HIGH);
break;
case 8:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,LOW);
digitalWrite(oute,LOW);
digitalWrite(outf,LOW);
digitalWrite(outg,LOW);
break;
case 9:
digitalWrite(outa,LOW);
digitalWrite(outb,LOW);
digitalWrite(outc,LOW);
digitalWrite(outd,HIGH);
digitalWrite(oute,HIGH);
digitalWrite(outf,LOW);
digitalWrite(outg,LOW);
break;
case 10:
digitalWrite(outa,HIGH);
digitalWrite(outb,HIGH);
digitalWrite(outc,HIGH);
digitalWrite(outd,HIGH);
digitalWrite(oute,HIGH);
digitalWrite(outf,HIGH);
digitalWrite(outg,LOW);
break;
}
```

```
// ********************************
// b) activo un solo display cada vez
// ********************************
for (int i=0;i<=3;i++) { // recorrido de 0 hasta 3

if (numdisplay==valorcoma) {
  digitalWrite(outcoma,LOW);
} else {
  digitalWrite(outcoma,HIGH);
}

// elige el nodo
if (i==numdisplay) {
digitalWrite(anodo[i],HIGH); // activo por
} else {
digitalWrite(anodo[i],LOW); // desactivo por
} // fin del if
} // fin del for
}

// ********************************
// displays
// ********************************
void valordisplay(int rayah, int rayam) {
  if (rayah==LOW) {
   displays[0]= h/10;
   displays[1]= h %10; // resto de la division entre 10
  } else {
   displays[0]= 10;
   displays[1]= 10; // guion
  }
  if (rayam==LOW) {
   displays[2]= m / 10;
   displays[3]= m %10; // resto de la division entre 10
  } else {
    displays[2]= 10;
    displays[3]= 10; // guion
  }
}

/* ===================================== */
/* Bucles principales del programa: setup */
/* ===================================== */
void setup () { // inicializo
pinMode(outa, OUTPUT);
pinMode(outb, OUTPUT);
pinMode(outc, OUTPUT);
pinMode(outd, OUTPUT);
pinMode(oute, OUTPUT);
pinMode(outf, OUTPUT);
pinMode(outg, OUTPUT);
pinMode(outcoma, OUTPUT);
pinMode(outmediosegundo, OUTPUT);
pinMode(anodo[0], OUTPUT);
pinMode(anodo[1], OUTPUT);
pinMode(anodo[2], OUTPUT);
```

```
pinMode(anodo[3], OUTPUT);
// Serial.begin(115200);
}

/* ===================================== */
/* Bucles principales del programa: loop  */
/* ===================================== */

void loop () { // bucle

tiempo1=millis(); // milisegundostranscurridos al empezar el bucle

if (start==1) {
  tiempo3=(tiempo1-tiempo2)/100; // decimas de segundo
} else {
  tiempo2=tiempo1-(tiempo3*100);
}

// calculo y mostrar
if (mode==0) {
  h = tiempo3 /100; // centenas - decenas segundos
  m = tiempo3 %100; // unidades y decimas de segundo
  valorcoma = 2; // tercer display
  if (tiempo3>9999) { // caso que sobrepase la cantidad
     mode=2; // cambia al modo 2.
  }
} else if (mode==1) {
  // segundos sin decimas
  segundos = tiempo3 / 10; // calculo los segundos transcurridos, sin decimas
  h = segundos /100 ;
  m = segundos %100;
  valorcoma = 3; // la coma en el ultimo digito
  if (segundos>9999) { // caso que sobrepase la cantidad
     mode=3; // cambia al modo 3.
  }
} else if (mode==2) {
  // minutos - segundos
  segundos = tiempo3 / 10; // calculo los segundos transcurridos
  minutos = segundos / 60;
  segundos = segundos %60;
  h = minutos;
  m = segundos;
  valorcoma=1; // la coma en la segunda posicion
  if (segundos>5999) { // caso que sobrepase la cantidad
     mode=3; // cambia al modo 3.
  }
} else if (mode==3) {
  // horas - minutos
  segundos = tiempo3 / 10; // calculo los segundos transcurridos
  mediosegundo = segundos %2;
  minutos = segundos / 60;
  hora = minutos / 60 ;
  // hora = hora %24 NO, no es un reloj
  minutos = minutos %60;
  h = hora;
  m = minutos;
```

```
    digitalWrite(outmediosegundo, mediosegundo); // la coma en la segunda posicion,
        parpadeante
    valorcoma = 9; // este valor no existe, y no pondr como ninguna
    if (hora*3600+minutos*60>359940) {
      RESTART;
    }
}

// anula el parpadeo de leds centrales en caso de mode distinto de 3
if (mode!=3) {
    digitalWrite(outmediosegundo, HIGH);
}

    valordisplay(LOW,LOW);
    cualactivo = (cualactivo+1)*(cualactivo<3);
    mandasenal(cualactivo,displays[cualactivo]);


// 1) comprueba start
if (analogRead(instart)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(instart)>500) {
    start=!start; // cambio la condicion de start
}
}

// 2) comprueba reset
if (analogRead(inreset)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(inreset)>500) {
    start=0;
    RESTART; // hago un RESET en caliente
}
}

// 3) comprueba modo
if (analogRead(inmode)>500) { //compara
delay(retrasoboton); // el retraso hace dar un pequeño salto al display.
// ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener
    que ser mas
if (analogRead(inmode)>500) {
    mode=(mode+1)*(mode<3);
}
}

delay(5);

}
// Fin del programa
```