

# PRÁCTICA 09

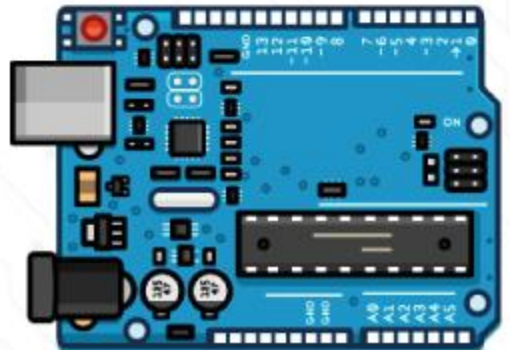
## Cronómetro

### OBJETIVO

Realizar un cronómetro que cuente décimas de segundo, segundos, minutos y horas

### + RESEÑA

*Proyecto para realizar un cronómetro con 3 botones y pantalla LCD 16X2.*



## **+ MATERIALES**

- 1 Placa de programación UNO (ARD-010)
- 1 Cable USB terminales A-B (USB-476)
- 1 Pantalla LCD (ARD-380)
- 1 Potenciómetro 10K (101-10K)
- 1 Resistencia de 220 Ohms (R220 1/2)
- 3 Micro *switch* de Push (AU-101L)
- 1 Tira Sencilla de pines (F36-S)
- 1 Paquete de cables tipo Dupont (ARD-310)
- 1 Protoboard (509-010)
- 1 Computadora con el entorno de desarrollo integrado (IDE) instalado

## + DESARROLLO

- 1 Coloca los tres micro switches en la protoboard. El botón 1 conéctalo al PIN 8 de la placa de programación, el botón 2 al PIN 9 y el 3 al PIN 10 y la otra terminal de cada micro switch envíala a GND.

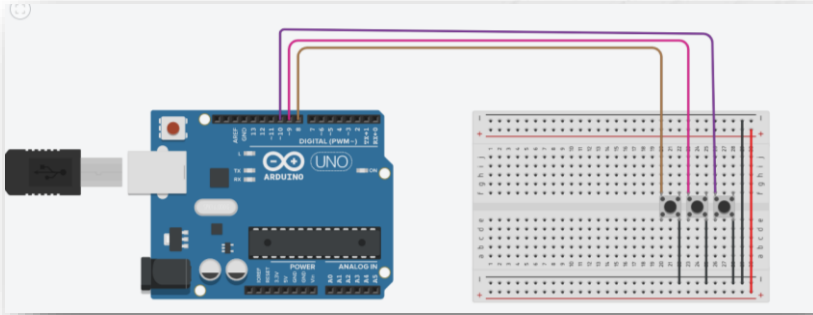


Imagen 1

- 2 Conecta las terminales Vcc y GND desde tu placa, hasta las líneas reservadas en la protoboard.

Conecta la pantalla LCD como se muestra, los pines que le soldaste podrán ser insertados en la protoboard o conectados con cables tipo Dupont hasta los pines de la placa de programación.

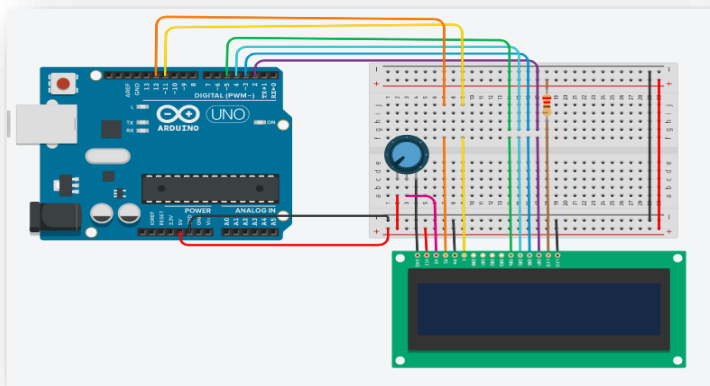


Imagen 2

## + CÓDIGO

- 3 **Abre un nuevo *sketch* dentro del entorno de desarrollo y coloca el siguiente programa:**

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
int inicio = 8;
int pausa = 9;
int reset = 10;
```

```
int horas = 0;
int minutos = 0;
int segundos = 0;
int decimas = 0;
int milisegundos = 0;
```

```
int a = 0;
int b = 0;
```

```
void setup()
{
  pinMode(inicio, INPUT_PULLUP);
  pinMode(pausa, INPUT_PULLUP);
  pinMode(reset, INPUT_PULLUP);
  lcd.begin(16, 2);
}
```

```
void loop()
{
  lcd.setCursor(12, 0);
  lcd.print("Time");
  lcd.setCursor(0, 0);
  lcd.print("h");
}
```

## + CÓDIGO

```
lcd.setCursor(3, 0);  
lcd.print("m");  
lcd.setCursor(6, 0);  
lcd.print("s");  
lcd.setCursor(9, 0);  
lcd.print("ds");  
  
inicio:  
if (digitalRead( inicio) == false )  
{  
    a = 1;  
    b = 0;  
    while (a == 1)  
    {  
        contador:  
        milisegundos = millis();  
        if (milisegundos % 100 == 0)  
        {  
            decimas++;  
            if (decimas == 10)  
            {  
                decimas = 0;  
                segundos++;  
            }  
            if (segundos == 60) {  
                segundos = 0;  
                minutos++;    }  
            if (minutos == 60) {  
                minutos = 0;  
                horas++;      }  
            }  
        }  
    }
```

## + CÓDIGO

```
lcd.setCursor(0, 1);
  if (horas < 10) {
    lcd.print("0");
  }
  lcd.print(horas);
  lcd.print(":");
  lcd.setCursor(3, 1);
  if (minutos < 10) {
    lcd.print("0");
  }
  lcd.print(minutos);
  lcd.print(":");
  lcd.setCursor(6, 1);
  if (segundos < 10) {
    lcd.print("0");
  }
  lcd.print(segundos);
  lcd.print(":");
  lcd.setCursor(9, 1);
  lcd.print(decimas);

  if (digitalRead(pausa) == false) {
    delay(200);
    a = 0;
    b = 1;
    while (b == 1) {
      if (digitalRead( inicio) == false) {
        goto contador;
      }
      if (digitalRead(reset) == false) {
        horas = 0;
        minutos = 0;
        segundos = 0;
        decimas = 0;
        lcd.clear();
        delay(200);
        goto inicio;
      }
    }
  }
  if (digitalRead(reset) == false)
  {
    a = 0;
    horas = 0;
    minutos = 0;
    segundos = 0;
    decimas = 0;
    lcd.clear();
    delay(200);
    goto inicio;
  }
}
```

## + DESARROLLO

- 4 Haz clic en el botón "Compilar". Podrás establecer un nombre para tu proyecto y validar que no existan errores de sintaxis.

COMPILAR



Imagen 3

# + DESARROLLO

5

Para ver el resultado, haz clic en el botón "Subir".

**SUBIR**

```
Archivo Editar Programa Herramientas Ayuda

cronometro_1

1 #include <LiquidCrystal>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 int inicio = 8;
5 int pausa = 9;
6 int reset = 10;
7
8 int horas = 0;
9 int minutos = 0;
10 int segundos = 0;
11 int decimas = 0;
12 int milisegundos = 0;
13
14 int a = 0;
15 int b = 0;
16
17 void setup()
18 {
19   pinMode(inicio, INPUT_PULLUP); |
20   pinMode(pausa, INPUT_PULLUP);
21   pinMode(reset, INPUT_PULLUP);
22   lcd.begin(16, 2);
23 }
24
25 void loop()
26 {
27   lcd.setCursor(12, 0);
28   lcd.print("Time");
```

Subido

Imagen 4



## **DESARROLLO**

**6**

Observarás en CERO, los valores para: horas, minutos, segundos y décimas.

Presionando el botón "inicio" el cronómetro se pondrá en funcionamiento.

El botón central, permite pausar el conteo y el botón de la derecha o reset, inicializa el conteo a CERO.

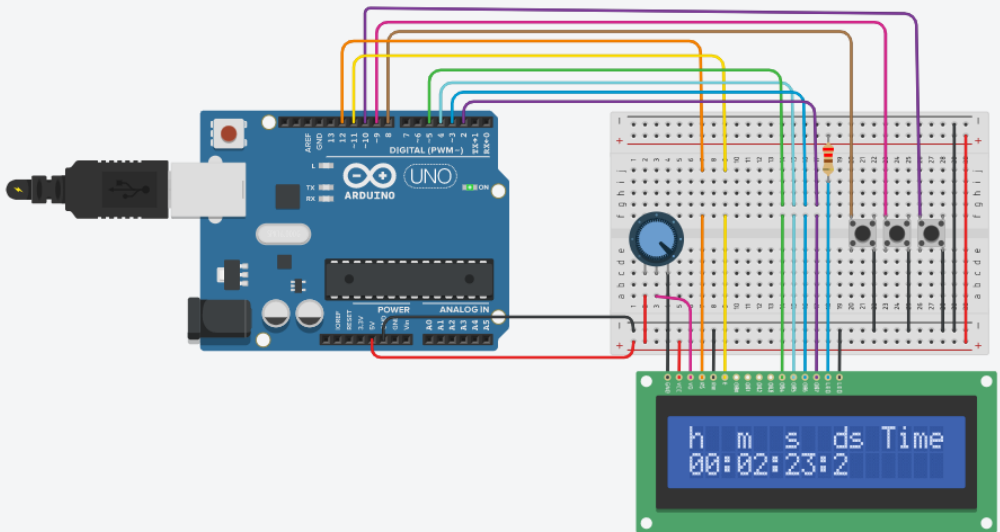


Imagen 5

¡Demos un vistazo al programa para ver cómo funciona!



## FUNCIONAMIENTO

- 1 El programa inicia con la declaración de la librería para el control de la pantalla LCD y los pines de la misma. Se nombran los pines que controlan los tres botones que permiten el inicio, pausa y reinicio del cronómetro. En esta sección también se crean las variables para llevar los diferentes conteos del tiempo transcurrido. Finalmente se crean unas variables que llevarán el estatus del conteo.

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 int inicio = 8;
5 int pausa = 9;
6 int reset = 10;
7
8 int horas = 0;
9 int minutos = 0;
10 int segundos = 0;
11 int decimas = 0;
12 int milisegundos = 0;
13
14 int a = 0;
15 int b = 0;
16
```

Imagen 6



## FUNCIONAMIENTO

2

En la sección `void setup()` se configuran los botones como entradas de alta impedancia, es decir, la única lectura que tomará como un cambio de su estado normal será la del cero absoluto o GND. También se inicializa la pantalla LCD como 16 caracteres y dos líneas.

```
16
17 void setup()
18 {
19     pinMode(inicio, INPUT_PULLUP);
20     pinMode(pausa, INPUT_PULLUP);
21     pinMode(reset, INPUT_PULLUP);
22     lcd.begin(16, 2);
23 }
24
```

Imagen 7

## + FUNCIONAMIENTO

3 Al inicio de la sección `void loop()`, se mandan a imprimir, en la primera línea, de la pantalla, las etiquetas para identificar cada sección del cronómetro.

Con ayuda del condicional IF, se especifica que solo cuando el botón "Inicio" sea pulsado, comenzará el conteo.

La variable "a" en conjunto con la estructura de control `while{ }`, permiten que mientras el valor de "a" sea 1, el cronómetro realice el conteo, determinando primero el número de decimales y a continuación los segundos.

```
25 void loop()
26 {
27     lcd.setCursor(12, 0);
28     lcd.print("Time");
29     lcd.setCursor(0, 0);
30     lcd.print("h");
31     lcd.setCursor(3, 0);
32     lcd.print("m");
33     lcd.setCursor(6, 0);
34     lcd.print("s");
35     lcd.setCursor(9, 0);
36     lcd.print("ds");
37
38     inicio:
39     if (digitalRead( inicio) == false )
40     {
41         a = 1;
42         b = 0;
43         while (a == 1)
44         {
45             contador:
46             milisegundos = millis();
47             if (milisegundos % 100 == 0)
48             {
49                 decimas++;
50                 if (decimas == 10)
51                 {
52                     decimas = 0;
```

Imagen 7



## FUNCIONAMIENTO

4

El programa continúa determinando los minutos y las horas. Para los dígitos con una posición, se prevé la colocación de un cero a la izquierda. Finalmente, son mostrados en la pantalla LCD los diferentes valores del conteo.

```
51 {  
52     decimas = 0;  
53     segundos++;  
54 }  
55 if (segundos == 60) {  
56     segundos = 0;  
57     minutos++;  
58 }  
59 if (minutos == 60) {  
60     minutos = 0;  
61     horas++;  
62 }  
63 lcd.setCursor(0, 1);  
64 if (horas < 10) {  
65     lcd.print("0");  
66 }  
67 lcd.print(horas);  
68 lcd.print(":");  
69 lcd.setCursor(3, 1);  
70 if (minutos < 10) {  
71     lcd.print("0");  
72 }  
73 lcd.print(minutos);  
74 lcd.print(":");  
75 lcd.setCursor(6, 1);  
76 if (segundos < 10) {  
77     lcd.print("0");  
78 }  
79 lcd.print(segundos);  
80 lcd.print(":");  
81 lcd.setCursor(9, 1);  
82 lcd.print(decimas);
```

Imagen 8

## + FUNCIONAMIENTO

- 5 Cuando el botón “pausa” es presionado, la variable “b” es la que tomará ahora el valor de UNO deteniendo el conteo que se muestra en la pantalla LCD.

```

79     if (digitalRead(pausa) == false) {
80         delay(200);
81         a = 0;
82         b = 1;
83         while (b == 1) {
84             if (digitalRead(inicio) == false) {
85                 goto contador;
86             }
87             if (digitalRead(reset) == false) {
88                 horas = 0;
89                 minutos = 0;
90                 segundos = 0;
91                 decimas = 0;
92                 lcd.clear();
93                 delay(200);
94                 goto inicio;
95             }
96         }
97     }

```

Imagen 9

- 6 Al presionar el botón “reset”, se igualan con cero todas las variables. Y la pantalla LCD se mostrará lista para iniciar un nuevo conteo.

```

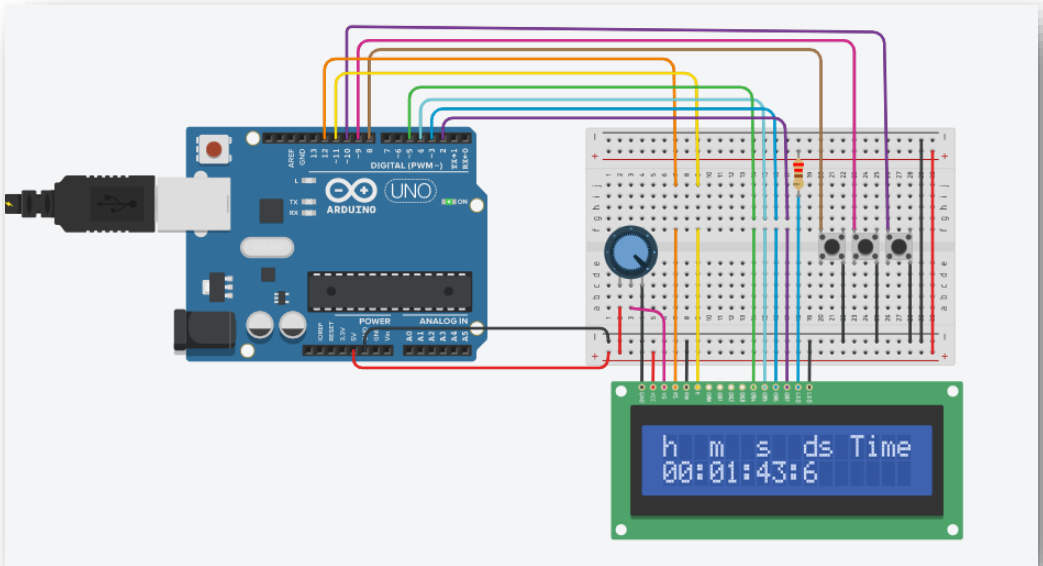
96     if (digitalRead(reset) == false)
97     {
98         a = 0;
99         horas = 0;
100        minutos = 0;
101        segundos = 0;
102        decimas = 0;
103        lcd.clear();
104        delay(200);
105        goto inicio;
106    }
107 }
108 }
109 }
110 }
111 }

```

Imagen 10

## **+ RESULTADO**

Inicia una y otra vez el conteo del cronómetro, y observa la exactitud que este muestra.



## **+ CONCLUSIONES**

En esta actividad:

Conjuntaste el uso del condicional IF y la estructura de control While.

Hiciste uso de la función `millis()`, la cual, lleva un conteo de tiempo en milisegundos de forma permanente siempre que la placa de programación esté energizada.