04 PRÁCTICA FINAL

NO. 04

Contador binario con pantalla LCD

SESIÓN	DURACIÓN	UNIDADES	PLATAFORMA
4 de 4	4 horas	7 y 8	Arduino Uno

Introducción

En la Práctica 2 el **74LS74** contaba con flip-flops: cada pulso de reloj sumaba uno y los LEDs mostraban el número en binario. Hoy el mismo trabajo lo hace un **microcontrolador** con un programa. Cada vez que se presiona el botón **SUMAR** la cuenta sube en uno; cuatro LEDs la muestran en binario y la pantalla LCD la escribe en decimal y en binario. Un puente llamado **BORRAR** la regresa a cero, como la entrada `CLR` del flip-flop.

El circuito se arma en tres etapas: primero los LEDs, luego las entradas y al final la pantalla. Cada etapa se prueba antes de pasar a la siguiente.

NOTA

Los grupos que trabajan con el **PICAXE-14M2** tienen su propia guía. El circuito hace exactamente lo mismo; solo cambian el microcontrolador y la forma de cargar el programa.

OBJETIVOS

- Conectar salidas (LEDs) y entradas (botón y puente) a un microcontrolador.
- Leer un número de 4 bits en binario y pasarlo a decimal.
- Usar una resistencia de 10 kΩ para que una entrada no quede al aire.
- Mostrar datos en una pantalla LCD por el bus I^2C.
- Cambiar el comportamiento del circuito modificando el programa.

MATERIALES Y EQUIPO

- **Arduino Uno** con su cable USB y una computadora con el Arduino IDE y la biblioteca `LiquidCrystal_I2C`.

- Pantalla LCD 16×2 con módulo I^2C y cuatro cables *jumper* macho-hembra.
- 4 LEDs y 4 resistencias de 330 Ω.
- 1 botón pulsador y 2 resistencias de 10 kΩ.
- Protoboard, cables *jumper* y el UT33B+.

PLANIFICACIÓN DE LA SESIÓN

Parte	Actividad	Tiempo
—	Apertura: contar en binario con 4 bits	20 min
A	Alimentación, LEDs y carga del programa	55 min
B	Las entradas: SUMAR y BORRAR	40 min
C	La pantalla LCD	45 min
D	Retos: cambiar el programa	35 min
—	Evaluación en Socrative, desarmado y cierre	25 min
	Total	**220 min + 20 min**

Fundamento teórico

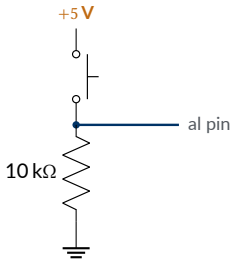
Contar con 4 bits

Cada LED es un bit. De derecha a izquierda valen 1, 2, 4 y 8: el número es la suma de los LEDs encendidos. Con 4 bits se cuenta de 0 a 15. Al sumar uno a 15 (1111) el resultado necesitaría un quinto bit, así que la cuenta **regresa a 0**, igual que un contador de flip-flops.

bit 3	bit 2	bit 1	bit 0	Valor
8	4	2	1	Peso de cada LED
0	1	1	0	$4 + 2 = 6$
1	0	1	1	$8 + 2 + 1 = 11$

Una entrada nunca queda al aire

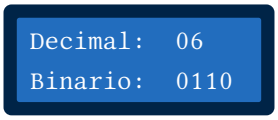
El botón une el pin con +5 V. Cuando está suelto, la resistencia de 10 kΩ lleva el pin a GND. Sin ella la entrada queda **flotando** y la cuenta avanza sola, como pasó con las entradas TTL al aire en la Práctica 1.



- **Suelto:** el pin lee `LOW` (0).
- **Presionado:** el pin lee `HIGH` (1). Por la resistencia pasan apenas $5\,\text{V}/10\,\text{k}\Omega = 0.5\,\text{mA}$.
- **El rebote:** al cerrarse, el contacto vibra unos milisegundos y el pin cambia varias veces. El programa espera 20 ms y, además, no vuelve a contar hasta que se suelta el botón.

La pantalla por I²C

El módulo soldado detrás de la LCD recibe los datos por solo dos cables: `SDA` (datos) al pin `A4` y `SCL` (reloj) al pin `A5`. Su dirección suele ser `0x27`; algunos módulos usan `0x3F`. Lo que se verá al final es esto:



El circuito

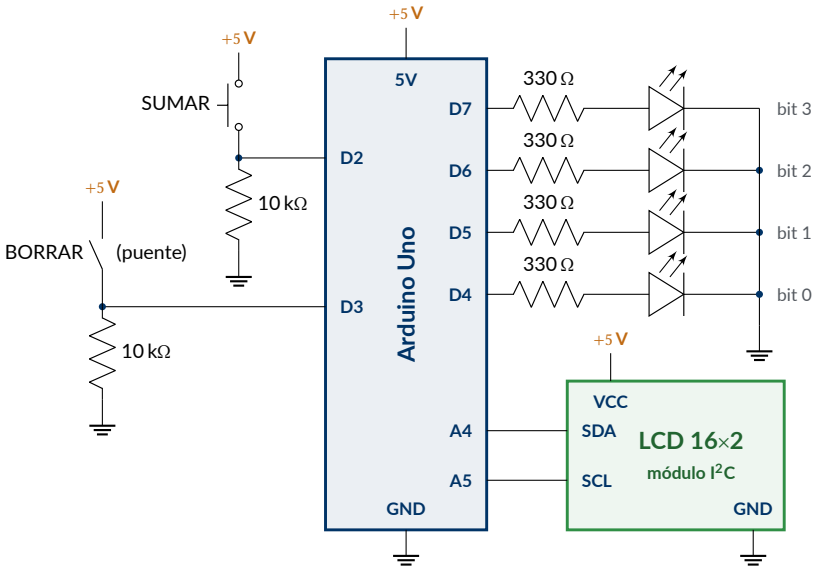


Figura 04.1. Esquema completo. Los cuatro LEDs tienen su cátodo (pata corta) hacia `GND`. El puente BORRAR es un cable que se pone y se quita.

Tabla 04.1. Qué va conectado a cada pin del Arduino

Pin	Se conecta a	Parte
5V	Riel + de abajo del protoboard	A
GND (dos)	Riel − de arriba y riel − de abajo	A
D7 a D4	Cada uno a una resistencia de 330 Ω y su LED: D7 es el bit 3 y D4 el bit 0	A
D2	Botón SUMAR (el otro lado a +5 V) y 10 kΩ a GND	B
D3	Puente BORRAR (el otro lado a +5 V) y 10 kΩ a GND	B
A4 / A5	SDA / SCL de la LCD; su VCC y GND a los rieles de abajo	C

El circuito armado

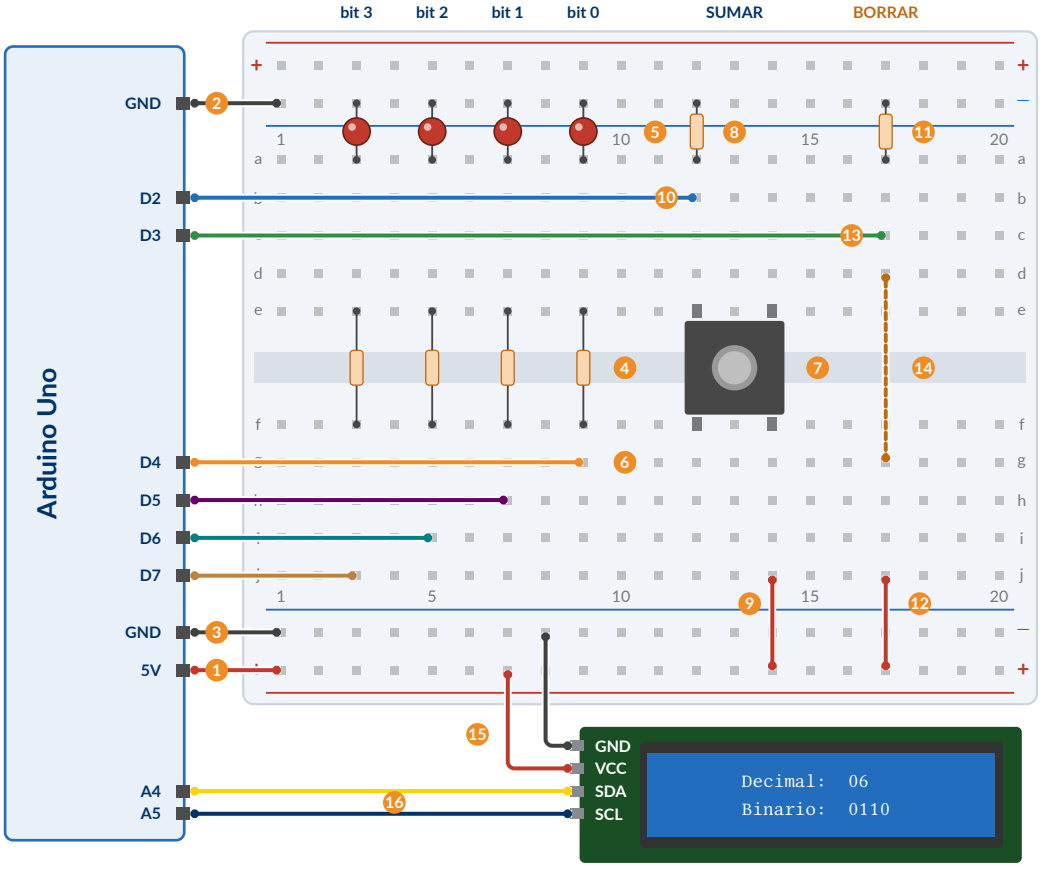


Figura 04.2. El circuito armado. Los números naranjas remiten a la lista de armado. Las letras (a a j) son las filas y los números, las columnas.

Tabla 04.2. Lista de armado

N.º	Qué se coloca y dónde	Parte
1	Cable rojo: 5V del Arduino al riel + de abajo	A
2	Cable negro: GND del Arduino al riel − de arriba	A
3	Cable negro: otro GND del Arduino al riel − de abajo	A
4	Cuatro resistencias de 330 Ω cruzando la ranura: de f3 a e3, f5–e5, f7–e7 y f9–e9	A
5	Cuatro LEDs: pata larga en a3, a5, a7 y a9; pata corta en el riel − de arriba	A
6	Cables: D7 a j3 (bit 3), D6 a i5 (bit 2), D5 a h7 (bit 1) y D4 a g9 (bit 0)	A
7	Botón SUMAR cruzando la ranura, con las patas en e12, e14, f12 y f14	B
8	Resistencia de 10 kΩ: de a12 al riel − de arriba	B
9	Cable rojo: de j14 al riel + de abajo	B
10	Cable: D2 a b12	B
11	Resistencia de 10 kΩ: de a17 al riel − de arriba	B
12	Cable rojo: de j17 al riel + de abajo	B
13	Cable: D3 a c17	B
14	Puente BORRAR: un cable de d17 a g17. Se pone para borrar y se quita	B
15	LCD: GND al riel − de abajo y VCC al riel + de abajo	C
16	LCD: SDA a A4 y SCL a A5, directo al Arduino	C

RECOMENDACIÓN

El botón usa dos patas **en diagonal**: e12 (arriba) y f14 (abajo). Dos patas en diagonal nunca están unidas por dentro, así que el circuito funciona aunque el botón se coloque girado.

ADVERTENCIA

Desconecte el USB antes de mover cualquier cable. Si su protoboard tiene los rieles partidos a la mitad, una las dos mitades con un cable.

Procedimiento

Parte A. Alimentación, LEDs y carga del programa

55 min

1. Con el USB desconectado, coloque los elementos 1 a 6 de la lista de armado.
2. Conecte el USB y mida con el UT33B+ entre los rieles + y − de abajo.
3. En el Arduino IDE abra `PF_Contador_LCD.ino` (su código completo está al final de esta guía), elija `Herramientas > Placa > Arduino Uno` y el puerto, y pulse **Subir**.
4. Al terminar la carga, la luz debe caminar una vez del bit 0 al bit 3. Después los LEDs se apagan: la cuenta está en 0. Pulse el botón **RESET** del Arduino para repetir la prueba.

Tabla 04.3. Primera prueba

Comprobación	Resultado
Tensión entre los rieles de abajo	
El programa se subió sin errores (sí / no)	
Orden en que encienden los LEDs al arrancar	
Al terminar la prueba, ¿cuántos LEDs quedan encendidos?	

NOTA

Si un LED nunca enciende, probablemente está al revés: la pata larga va en la fila a y la corta en el riel azul. Si no enciende ninguno, revise el cable negro del elemento 2.

Parte B. Las entradas: SUMAR y BORRAR

40 min

1. Desconecte el USB y coloque los elementos 7 a 14. El puente BORRAR *no* se deja puesto.
2. Conecte el USB. Presione SUMAR una vez por renglón de la tabla y anote los LEDs en el orden bit 3, bit 2, bit 1, bit 0 (encendido = 1).
3. Al final, ponga y quite el puente BORRAR.

Tabla 04.4. El contador en los LEDs

Pulsaciones desde 0	LEDs (bit 3 a bit 0)	En decimal
1		
2		
3		
4		
7		
8		
10		
15		
16		
Después de poner el puente BORRAR		

NOTA

Si la cuenta avanza sola o salta de dos en dos, revise la resistencia de 10 kΩ del elemento 8: sin ella la entrada `D2` queda al aire.

Parte C. La pantalla LCD

45 min

1. Desconecte el USB y conecte la LCD (elementos 15 y 16).
2. Conecte el USB. La pantalla muestra el saludo mientras la luz camina y después las dos líneas `Decimal:` y `Binario:`.
3. Si la luz de fondo enciende pero no se ven letras, gire con un desarmador el potenciómetro azul del módulo (contraste).
4. Lleve la cuenta a cada número de la tabla y anote lo que dice la LCD.

Tabla 04.5. La LCD y los LEDs

Lleve la cuenta a	LCD: Decimal	LCD: Binario	¿Coincide con los LEDs?
5			
9			
12			
14			

Tabla 04.6. El rebote

Prueba	La LCD marca
Ponga el puente BORRAR, quítelo y presione SUMAR 10 veces, lo más rápido que pueda	

Parte D. Retos: cambiar el programa

35 min

Cada reto cambia una o dos líneas. Guarde el programa con otro nombre antes de empezar.

1. **Contador de década.** Haga que la cuenta vaya de 0 a 9 y después regrese a 0, como el 74LS90. Pista: busque el número 15 en `loop()`.
2. **Contar hacia abajo.** Haga que BORRAR *reste* uno en lugar de ir a 0, y que debajo de 0 siga el 15, como el 74LS193.

Si algo no funciona

Síntoma	Revise
El IDE no encuentra el puerto	Otro cable USB (algunos solo cargan); el controlador `CH340` si la placa es compatible
Un LED nunca enciende	Su polaridad y que la resistencia cruce la ranura en la misma columna
La cuenta avanza sola	La resistencia de 10 kΩ de a12 al riel −
SUMAR no hace nada	El cable rojo de j14; el botón debe usar patas en diagonal
Los LEDs muestran otro número	Cables de `D4` a `D7` cruzados: `D7` es el bit 3
LCD encendida, sin letras	El contraste: potenciómetro azul del módulo
LCD con cuadros en la primera línea	La dirección: cambie `0x27` por `0x3F` y vuelva a subir
LCD apagada	`VCC` y `GND` del módulo; `SDA` y `SCL` cruzados

Preguntas de cierre

1. ¿Por qué después de 15 la cuenta regresa a 0?

2. ¿Qué pasaría si se quita la resistencia de 10 kΩ del botón SUMAR?

3. ¿Qué hace el programa para que un toque del botón cuente una sola vez?

4. ¿Cuántos cables necesita la LCD y qué lleva cada uno?

El programa completo

</> PF_CONTADOR_LCD.INO

```
1  // Universidad Rural de Guatemala - Electronica Digital FE731
```

```
// Practica final, version Arduino: contador binario con pantalla LCD
// SUMAR suma 1 y BORRAR regresa a 0. La cuenta, de 0 a 15, se ve en
// cuatro LEDs y en la LCD (I2C), en decimal y en binario.

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int SUMAR  = 2;    // boton
const int BORRAR = 3;    // puente a +5 V
const int LED0 = 4;      // bit 0 (derecha)
const int LED1 = 5;      // bit 1
const int LED2 = 6;      // bit 2
const int LED3 = 7;      // bit 3 (izquierda)

LiquidCrystal_I2C lcd(0x27, 16, 2);   // si solo salen cuadros, use 0x3F

int cuenta = 0;

void ponerLEDs() {
  digitalWrite(LED0, bitRead(cuenta, 0));
  digitalWrite(LED1, bitRead(cuenta, 1));
  digitalWrite(LED2, bitRead(cuenta, 2));
  digitalWrite(LED3, bitRead(cuenta, 3));
}

void mostrar() {
  ponerLEDs();

  lcd.setCursor(9, 0);          // Decimal:  07
  if (cuenta < 10) {
    lcd.print("0");
  }
  lcd.print(cuenta);

  lcd.setCursor(9, 1);          // Binario:  0111
  lcd.print(bitRead(cuenta, 3));
  lcd.print(bitRead(cuenta, 2));
  lcd.print(bitRead(cuenta, 1));
  lcd.print(bitRead(cuenta, 0));
}

void setup() {
  pinMode(SUMAR, INPUT);
  pinMode(BORRAR, INPUT);
  pinMode(LED0, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("URURAL  FE731");
  lcd.setCursor(0, 1);
```

```
  lcd.print("Contador binario");

  // Prueba de los LEDs: la luz camina del bit 0 al bit 3
  for (cuenta = 1; cuenta <= 8; cuenta = cuenta * 2) {
    ponerLEDs();
    delay(400);
  }

  cuenta = 0;
  lcd.clear();
  lcd.print("Decimal:");
  lcd.setCursor(0, 1);
  lcd.print("Binario:");
  mostrar();
}

void loop() {
  if (digitalRead(SUMAR) == HIGH) {
    cuenta = cuenta + 1;
    if (cuenta > 15) {
      cuenta = 0;
    }
    mostrar();
    delay(20);                                  // deja pasar el rebote
    while (digitalRead(SUMAR) == HIGH) { }      // espera a que se suelte
    delay(20);
  }

  if (digitalRead(BORRAR) == HIGH) {
    cuenta = 0;
    mostrar();
    delay(20);
    while (digitalRead(BORRAR) == HIGH) { }
    delay(20);
  }
}
```

Parte del programa	Qué hace
`ponerLEDs()`	Enciende cada LED con un bit de la cuenta: `bitRead(cuenta, 0)` es el bit 0
`mostrar()`	Actualiza los LEDs y escribe la cuenta en la LCD, en decimal y en binario
`setup()`	Prepara los pines y la LCD, saluda y hace caminar la luz
`loop()`	Revisa SUMAR y BORRAR una y otra vez; espera el rebote y que se suelte el botón

EVALUACIÓN EN LÍNEA (SOCRATIVE)

Al terminar la práctica se aplicará una evaluación breve en **Socrative**. Las preguntas combinan la teoría de la clase magistral en línea con lo que se acaba de hacer en el laboratorio.

El instructor indicará el código de la sala. Anótelo aquí, junto con cualquier indicación del día.

Código de sala / enlace

Espacio para la evaluación

Reservado para Socrative: el instructor colocará aquí el acceso o las indicaciones