

PRÁCTICA DE ARDUINO - (I2C) DISPLAY LCD CON CONTADOR

NIVEL: **1**

DURACIÓN: 2 horas

OBJETIVO: Los estudiantes aprenderán a usar una pantalla LCD 16x2 para mostrar un número que se incrementa automáticamente cada segundo.

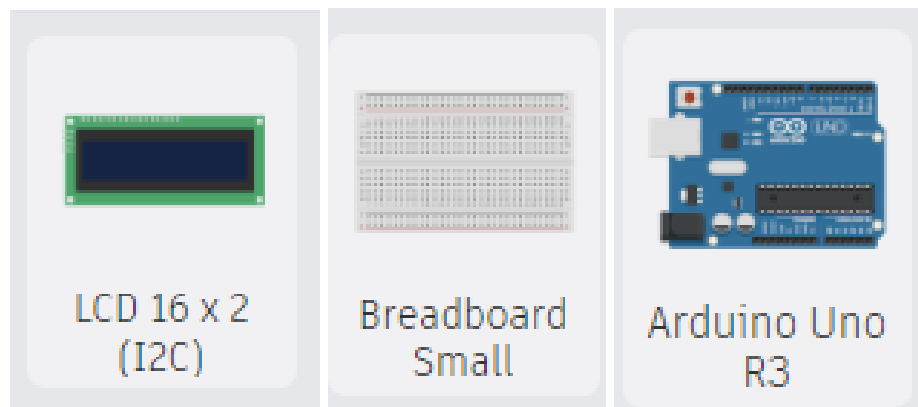
MATERIALES

- Placa Arduino Uno (o similar)
- Pantalla LCD 16x2
- Potenciómetro (si el LCD no tiene interfaz I2C, para ajustar el contraste)
- Módulo I2C
- Cables de conexión
- Protoboard
- Computadora con el software Arduino IDE instalado
- [ThinkerCad](#)

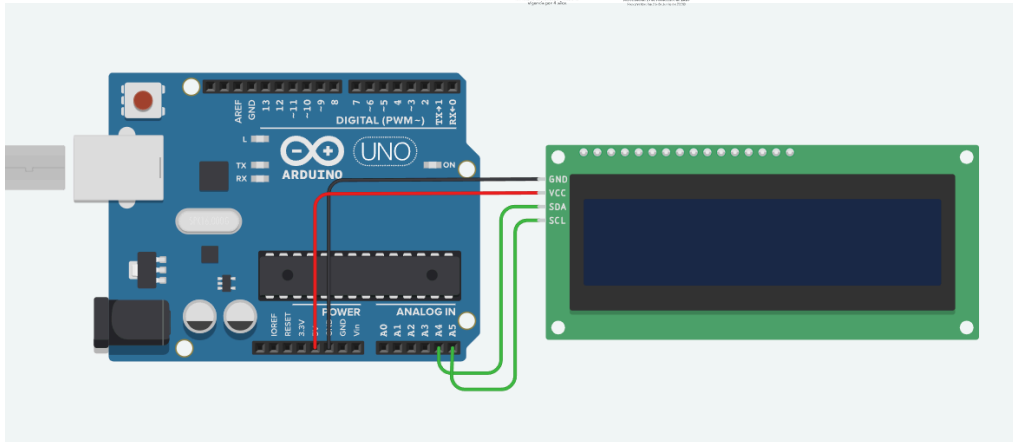
1. DESARROLLO DE LA PRÁCTICA

1.1 SIMULACIÓN EN THINKERCAD

1.1.1 COMPONENTES



1.1.2 MONTAJE



2.2 PROGRAMACIÓN EN ARDUINO IDE

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

// Crear un objeto LCD con la dirección I2C 0x27 (puede variar)
LiquidCrystal_I2C lcd(0x27, 16, 2);

int contador = 0; // Variable para el contador

void setup() {
    // Inicializar la pantalla LCD
    lcd.init();
    lcd.backlight(); // Encender la luz de fondo del LCD

    // Mostrar mensaje inicial
    lcd.setCursor(0, 0); // Colocar el cursor en la primera fila
    lcd.print("Contador:");
}

void loop() {
    // Mostrar el valor del contador en la segunda fila
    lcd.setCursor(0, 1); // Colocar el cursor en la segunda fila
    lcd.print(contador); // Mostrar el valor del contador
    lcd.print("  "); // Limpiar cualquier valor anterior

    contador++; // Incrementar el contador
    delay(1000); // Esperar 1 segundo
}
```

Explicación

- `#include <Wire.h>` y `#include <LiquidCrystal_I2C.h>`: Estas bibliotecas permiten la comunicación a través del protocolo I2C con la pantalla LCD.
- `LiquidCrystal_I2C lcd(0x27, 16, 2);`: Aquí creamos un objeto `lcd` con la dirección I2C (que generalmente es 0x27, aunque puede variar). El objeto está configurado para una pantalla de 16 columnas y 2 filas.
- `lcd.init();`: Inicializa la pantalla LCD.
- `lcd.backlight();`: Enciende la luz de fondo del LCD para que los caracteres sean visibles.
- `lcd.setCursor(0, 0);`: Coloca el cursor en la primera fila y primera columna.
- `lcd.print("Contador:");`: Muestra el texto "Contador:" en la pantalla.
- En cada iteración del `loop()`, el contador se muestra en la segunda fila (`lcd.setCursor(0, 1)`), y se incrementa cada segundo con `contador++` y `delay(1000)`.
- El código utiliza `lcd.print(" ")` después del contador para asegurarse de que los números anteriores se borren cuando el contador cambia de dígitos, evitando que queden caracteres no deseados en la pantalla.

2. ANÁLISIS DE RESULTADOS

- ¿Cómo simplifica el uso del módulo I2C la conexión de la pantalla LCD?

El uso del módulo I2C reduce significativamente la cantidad de pines utilizados en el Arduino. En lugar de usar 6 o más pines para conectar el LCD, solo necesitamos dos pines para la comunicación: SDA (A4) y SCL (A5). Esto libera pines para otros componentes, facilitando la creación de proyectos más complejos y ordenados.

- ¿Cómo se ajusta el contraste de la pantalla LCD con el módulo I2C?

Cuando se usa el módulo I2C, el contraste de la pantalla LCD se ajusta a través de un pequeño potenciómetro en la parte posterior del módulo I2C. Esto permite modificar el contraste de manera física sin necesidad de conectar un potenciómetro separado como se haría con una pantalla sin I2C.

- ¿Qué ventajas ofrece el módulo I2C para proyectos más complejos?

El módulo I2C permite manejar varias pantallas o dispositivos con solo 2 pines de comunicación, lo que es especialmente útil en proyectos complejos que requieren conectar múltiples sensores, pantallas u otros componentes. Además, el protocolo I2C es más eficiente y permite controlar varios dispositivos en el mismo bus I2C, utilizando diferentes direcciones de cada dispositivo.

- ¿Qué ocurre si la dirección I2C no es la correcta en el código?

Cada pantalla LCD con I2C tiene una dirección única que puede variar (aunque 0x27 es la más común). Si la dirección en el código no coincide con la del módulo I2C, la pantalla no funcionará correctamente o no mostrará nada. Se puede usar un escáner I2C para encontrar la dirección correcta .

- ¿Cómo podríamos modificar el proyecto para que el contador comience en un número específico?

Se puede modificar el valor inicial de la variable contador en el código para que comience desde cualquier número . Por ejemplo, que el contador comience en 100, simplemente cambia:

```
int contador = 100;
```

Esto hará que el contador comience a incrementarse desde ese número en lugar de 0.

- ¿Cómo modificar el código para que no sea necesario el modulo I2C?

```
#include <LiquidCrystal.h>
```

```
// Definir los pines de la LCD
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
int contador = 0; // Variable para el contador
```

```
void setup() {
```

```
  // Inicializar la pantalla LCD
```

```
  lcd.begin(16, 2);
```

```
  // Mostrar mensaje inicial
```

```
  lcd.setCursor(0, 0); // Colocar el cursor en la primera fila
```

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

```
}
```

```
void loop() {
```

```
  // Mostrar el valor del contador en la segunda fila
```

```
  lcd.setCursor(0, 1); // Colocar el cursor en la segunda fila
```

```
  lcd.print(contador); // Mostrar el valor del contador
```

```
  lcd.print("   "); // Limpiar cualquier valor anterior (para evitar números extra)
```

```
  contador++; // Incrementar el contador
```

```
  delay(1000); // Esperar 1 segundo
```

```
}
```

3. CONCLUSIONES

- A. Usar un módulo I2C con la pantalla LCD simplifica la conexión y reduce la cantidad de pines necesarios, lo que facilita el desarrollo de proyectos más grandes y complejos. EL I2C permite conectar varios dispositivos con pocos pines.
- B. El proyecto enseña a mostrar datos en tiempo real en una pantalla LCD, lo que es útil en proyectos que requieren monitorear y mostrar información, como contadores, temporizadores, o la lectura de sensores.
- C. La interacción con pantallas LCD permite al usuario visualizar datos de forma clara y ordenada. Se pueden mostrar múltiples variables o información detallada en diferentes filas y columnas de la pantalla.
- D. Al usar la comunicación I2C, liberamos pines en el Arduino, lo que nos permite agregar más componentes y sensores sin preocuparnos por quedarnos sin pines.
- E. El proyecto puede expandirse añadiendo más funciones, como controles adicionales para restablecer el contador, cambiar la velocidad del conteo, o agregar entradas de usuario como botones o sensores para interactuar con la pantalla y el sistema.