Este código utiliza un Arduino para leer datos de temperatura y humedad de un sensor DHT11 y mostrarlos en una pantalla OLED (SSD1306) y una pantalla LCD (I2C). A continuación, te explico el paso a paso explícito de cómo funciona el código:

1. **Inclusión de bibliotecas:**

#include <Wire.h>

#include <SPI.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <LiquidCrystal\_I2C.h>

* **Wire.h**: Biblioteca para comunicación I2C, usada por la pantalla OLED y LCD.
* **SPI.h**: Biblioteca para comunicación SPI (no se usa directamente en este código, pero es requerida por Adafruit\_GFX).
* **Adafruit\_GFX.h**: Biblioteca gráfica para manejar la pantalla OLED.
* **Adafruit\_SSD1306.h**: Biblioteca específica para controlar la pantalla OLED SSD1306.
* **LiquidCrystal\_I2C.h**: Biblioteca para controlar la pantalla LCD con interfaz I2C.
* **DHT.h**: Biblioteca para interactuar con el sensor DHT11.

1. **Configuración de la pantalla OLED:**

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1

#define SCREEN\_ADDRESS 0x3C

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

* Se definen las constantes para el ancho (128 píxeles) y alto (64 píxeles) de la pantalla OLED.
* OLED\_RESET se establece en -1 porque no se usa un pin de reset para la pantalla.
* SCREEN\_ADDRESS (0x3C) es la dirección I2C de la pantalla OLED.
* Se crea un objeto display de la clase Adafruit\_SSD1306 para controlar la pantalla OLED, pasando el ancho, alto, la interfaz I2C (&Wire) y el pin de reset.

1. **Configuración de la pantalla LCD:**

LiquidCrystal\_I2C lcd(0x27, 16, 2);

* Se crea un objeto lcd de la clase LiquidCrystal\_I2C para controlar la pantalla LCD.
* 0x27 es la dirección I2C de la pantalla LCD.
* 16 y 2 indican que la pantalla tiene 16 columnas y 2 filas.

1. **Configuración del sensor DHT11:**

#define DHTPIN 5

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

* DHTPIN define el pin digital 5 del Arduino al que está conectado el sensor DHT11.
* DHTTYPE especifica que se usa un sensor DHT11.
* Se crea un objeto dht de la clase DHT para interactuar con el sensor, pasando el pin y el tipo de sensor.

1. **Función setup():**

void setup() {

Serial.begin(9600);

dht.begin();

* Inicialización del puerto serial: Serial.begin(9600) inicia la comunicación serial a 9600 baudios para depuración.
* Inicialización del sensor DHT: dht.begin() prepara el sensor DHT11 para comenzar a leer datos.
* **Inicialización de la pantalla OLED**

if (!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {

Serial.println(F("Error al iniciar el OLED"));

for (;;);

}

* display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS) intenta inicializar la pantalla OLED con alimentación interna (SSD1306\_SWITCHCAPVCC) y la dirección I2C.
* Si falla (devuelve false), se imprime un mensaje de error en el monitor serial y el programa se detiene en un bucle infinito (for(;;)).

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 0);

display.println("Iniciando...");

display.display();

delay(2000);

display.clearDisplay();

* display.clearDisplay(): Limpia el buffer de la pantalla OLED.
* display.setTextSize(1): Establece el tamaño del texto a 1 (normal).
* display.setTextColor(WHITE): Define el color del texto como blanco (único color en pantallas monocromáticas).
* display.setCursor(0, 0): Coloca el cursor en la esquina superior izquierda.
* display.println("Iniciando..."): Escribe el mensaje "Iniciando...".
* display.display(): Muestra el contenido del buffer en la pantalla.
* delay(2000): Pausa de 2 segundos.
* display.clearDisplay(): Limpia la pantalla nuevamente.
* **Inicialización de la pantalla LCD**

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Iniciando...");

delay(2000);

lcd.clear();

* lcd.init(): Inicializa la pantalla LCD.
* lcd.backlight(): Enciende la retroiluminación de la pantalla.
* lcd.setCursor(0, 0): Coloca el cursor en la primera fila, primera columna.
* lcd.print("Iniciando..."): Muestra el mensaje "Iniciando...".
* delay(2000): Pausa de 2 segundos.
* lcd.clear(): Limpia la pantalla.

1. Función loop()

float h = dht.readHumidity();

float t = dht.readTemperature();

* Lee la humedad (h) y la temperatura (t) del sensor DHT11 usando dht.readHumidity() y dht.readTemperature().
* **Verificación de errores en la lectura**

if (isnan(h) || isnan(t)) {

Serial.println("Error al leer los datos del sensor DHT11");

display.clearDisplay();

display.setCursor(0, 0);

display.println("Error al leer");

display.println("datos del sensor");

display.display();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Error!... ");

delay(2000);

return;

}

* isnan(h) || isnan(t): Comprueba si las lecturas de humedad o temperatura son inválidas (NaN, "Not a Number").
* Si hay un error:
* Imprime un mensaje de error en el monitor serial.
* Muestra "Error al leer datos del sensor" en la pantalla OLED (dos líneas).
* Muestra "Error!..." en la pantalla LCD.
* Espera 2 segundos (delay(2000)).
* return: Termina la iteración actual del loop() y comienza una nueva.
* **Impresión de datos válidos**

Serial.print("Humedad: ");

Serial.print(h);

Serial.print("% | Temperatura: ");

Serial.print(t);

Serial.println("C");

* Si las lecturas son válidas, imprime la humedad y temperatura en el monitor serial con el formato: Humedad: X% | Temperatura: YC.
* **Mostrar datos en la pantalla OLED**

display.clearDisplay();

display.setTextSize(1);

display.setCursor(0, 20);

display.print("Temperatura: ");

display.print(t);

display.print(" C");

display.setCursor(0, 40);

display.print("Humedad: ");

display.print(h);

display.print(" %");

display.display();

* Limpia la pantalla OLED.
* Establece el tamaño de texto a 1.
* En la posición (0, 20) (20 píxeles desde la parte superior), muestra "Temperatura: X C".
* En la posición (0, 40), muestra "Humedad: Y %".
* Actualiza la pantalla con display.display().
* **Mostrar datos en la pantalla LCD**

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(t);

lcd.print("C");

lcd.setCursor(0, 1);

lcd.print("Humedad: ");

lcd.print(h);

lcd.print("%");

* Limpia la pantalla LCD.
* En la primera fila (0, 0), muestra "Temp: X C".
* En la segunda fila (0, 1), muestra "Humedad: Y %".
* Pausa y repetición

delay(2000);

* Pausa de 2 segundos antes de la siguiente iteración del loop(), dando tiempo para leer los datos en las pantallas.

**Flujo general del programa**

1. **Inicio**: Configura el puerto serial, el sensor DHT11, la pantalla OLED y la pantalla LCD, mostrando un mensaje inicial ("Iniciando...") en ambas pantallas.
2. **Bucle continuo**:

* Lee la temperatura y humedad del sensor DHT11.
* Si las lecturas son inválidas, muestra mensajes de error en el monitor serial, OLED y LCD, y espera 2 segundos antes de intentar de nuevo.
* Si las lecturas son válidas, muestra los datos en:
* Monitor serial: "Humedad: X% | Temperatura: YC".
* Pantalla OLED: "Temperatura: X C" y "Humedad: Y %" en diferentes líneas.
* Pantalla LCD: "Temp: X C" en la primera fila y "Humedad: Y %" en la segunda.
* Espera 2 segundos y repite el proceso.

**Notas adicionales**

* **Conexiones físicas**:
  + El sensor DHT11 debe estar conectado al pin 5 del Arduino.
  + La pantalla OLED y LCD deben estar conectadas al bus I2C (pines SDA y SCL del Arduino).
  + Verifica las direcciones I2C (0x3C para OLED, 0x27 para LCD) usando un escáner I2C si no funcionan.
* **Bibliotecas necesarias**: Asegúrate de instalar las bibliotecas Adafruit\_GFX, Adafruit\_SSD1306, LiquidCrystal\_I2C, y DHT en el IDE de Arduino.
* **Errores comunes**:
  + Si el sensor DHT11 no lee datos, verifica las conexiones y la resistencia pull-up (normalmente 4.7kΩ o 10kΩ) entre el pin de datos y VCC.
  + Si las pantallas no muestran nada, confirma las direcciones I2C y las conexiones.