**1. Inclusión de Bibliotecas**

El programa incluye las siguientes bibliotecas:

* **SPI.h**: Requerida por Adafruit\_GFX, aunque no se usa directamente en el código.
* **Wire.h**: Para comunicación I2C, usada por la pantalla OLED.
* **Adafruit\_GFX.h**: Biblioteca gráfica para manejar la pantalla OLED.
* **Adafruit\_SSD1306.h**: Biblioteca específica para controlar la pantalla OLED SSD1306.
* **DHT.h**: Biblioteca para interactuar con el sensor DHT11.

**2. 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);

* Define el ancho (128 píxeles) y alto (64 píxeles) de la pantalla OLED.
* OLED\_RESET se establece en -1, indicando que no se usa un pin de reset.
* SCREEN\_ADDRESS (0x3C) es la dirección I2C de la pantalla OLED.
* Crea un objeto display para controlar la pantalla, pasando el ancho, alto, la interfaz I2C (&Wire) y el pin de reset.

**3. Definición del Bitmap**

static const unsigned char PROGMEM sunfounderIcon[] = {

// Aquí iría el array de bytes del icono

};

* Declara un array de bytes (sunfounderIcon) para un bitmap que podría mostrarse en la pantalla OLED.
* Nota: El array no está completo en el código proporcionado, por lo que no se utiliza en el programa actual.

**4. Configuración del Sensor DHT11**

#define DHTPIN 5

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

* DHTPIN define el pin digital 5 del Arduino para el sensor DHT11.
* DHTTYPE especifica que se usa un sensor DHT11.
* Crea un objeto dht para interactuar con el sensor, pasando el pin y el tipo de sensor.

**5. Función setup()**

Serial.begin(9600);

dht.begin();

* Inicia la comunicación serial a 9600 baudios para depuración.
* Prepara el sensor DHT11 para leer datos.

**Inicialización de la Pantalla OLED**

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

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

for (;;);

}

* Intenta inicializar la pantalla OLED con alimentación interna (SSD1306\_SWITCHCAPVCC) y la dirección I2C.
* Si falla, imprime un mensaje de error en el monitor serial y detiene el programa en un bucle infinito.

display.clearDisplay();

display.setTextSize(1.5);

display.setTextColor(WHITE);

display.setCursor(0, 28);

display.println("Inicializando...");

display.display();

delay(2000);

display.clearDisplay();

* Limpia el buffer de la pantalla OLED.
* Establece el tamaño del texto a 1.5 (escala personalizada).
* Define el color del texto como blanco (único color en pantallas monocromáticas).
* Coloca el cursor en la posición (0, 28) para centrar el texto verticalmente.
* Muestra el mensaje "Inicializando...".
* Actualiza la pantalla con display.display().
* Espera 2 segundos y limpia la pantalla nuevamente.

**6. Función loop()**

delay(2000);

float h = dht.readHumidity();

float t = dht.readTemperature();

* Espera 2 segundos entre lecturas para estabilizar el sensor.
* Lee la humedad (h) y la temperatura (t) del sensor DHT11.

**Verificación de Errores**

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

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

return;

}

* Verifica si las lecturas son inválidas (NaN, "Not a Number").
* Si hay un error, imprime un mensaje en el monitor serial y termina la iteración actual del loop().

**Impresión en el Monitor Serial**

Serial.print("Humedad: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperatura: ");

Serial.print(t);

Serial.println(" \*C");

* Muestra los datos en el monitor serial en el formato: Humedad: X % Temperatura: Y \*C.

**Mostrar Datos en la Pantalla OLED**

display.setTextSize(2);

display.setCursor(0, 28);

display.println("Temperatura:");

display.display();

display.setCursor(88, 28);

display.println(t);

display.display();

display.setCursor(0, 45);

display.println("Humedad:");

display.display();

display.setCursor(88, 45);

display.println(h);

display.display();

delay(2000);

display.clearDisplay();

* Establece el tamaño del texto a 2 para mayor legibilidad.
* Muestra "Temperatura:" en la posición (0, 28) y el valor de t en (88, 28).
* Muestra "Humedad:" en la posición (0, 45) y el valor de h en (88, 45).
* Actualiza la pantalla después de cada escritura con display.display().
* Espera 2 segundos y limpia la pantalla.

**Flujo General del Programa**

1. **Inicio**: Configura el puerto serial, el sensor DHT11 y la pantalla OLED, mostrando "Inicializando..." durante 2 segundos.
2. **Bucle continuo**:
   * Lee temperatura y humedad cada 2 segundos.
   * Si las lecturas son inválidas, muestra un error en el monitor serial y reinicia el ciclo.
   * Si las lecturas son válidas, muestra los datos en:
     + Monitor serial: Humedad: X % Temperatura: Y \*C.
     + Pantalla OLED: "Temperatura: Y" y "Humedad: X" en dos líneas.
   * Espera 2 segundos, limpia la pantalla y repite el proceso.

**Notas Adicionales**

* **Conexiones físicas**:
  + El sensor DHT11 debe estar conectado al pin 5 del Arduino.
  + La pantalla OLED debe estar conectada al bus I2C (pines SDA y SCL).
  + Verifica la dirección I2C (0x3C) con un escáner I2C si no funciona.
* **Bibliotecas necesarias**: Instala Adafruit\_GFX, Adafruit\_SSD1306 y DHT en el IDE de Arduino.
* **Errores comunes**:
  + Si el sensor no lee datos, verifica las conexiones y la resistencia pull-up (4.7kΩ o 10kΩ) entre el pin de datos y VCC.
  + Si la pantalla no muestra nada, confirma la dirección I2C y las conexiones.
* **Bitmap no utilizado**: El array sunfounderIcon está definido pero no se usa en el código actual.