

PRÁCTICA DE ARDUINO - ESTACIÓN METEOROLÓGICA

NIVEL: 2

DURACIÓN: 2 horas

OBJETIVO: Los estudiantes aprenderán a crear una estación meteorológica con sensores múltiples

MATERIALES

- Arduino Uno (o similar)
- Sensor de temperatura y humedad: DHT11 o DHT22
- Sensor de presión atmosférica: BMP180 o BMP280
- Sensor de lluvia
- Pantalla LCD con modulo I2C PCF8574
- 2 Resistencias 10k
- 2 Switch
- Módulo Bluetooth HC-05 o HC-06 (opcional))
- Cables y protoboard
- Computadora con el software Arduino IDE instalado
- Proteus

1. DESARROLLO DE LA PRÁCTICA

1.1 SIMULACIÓN EN PROTEUS

1.1.1 MONTAJE

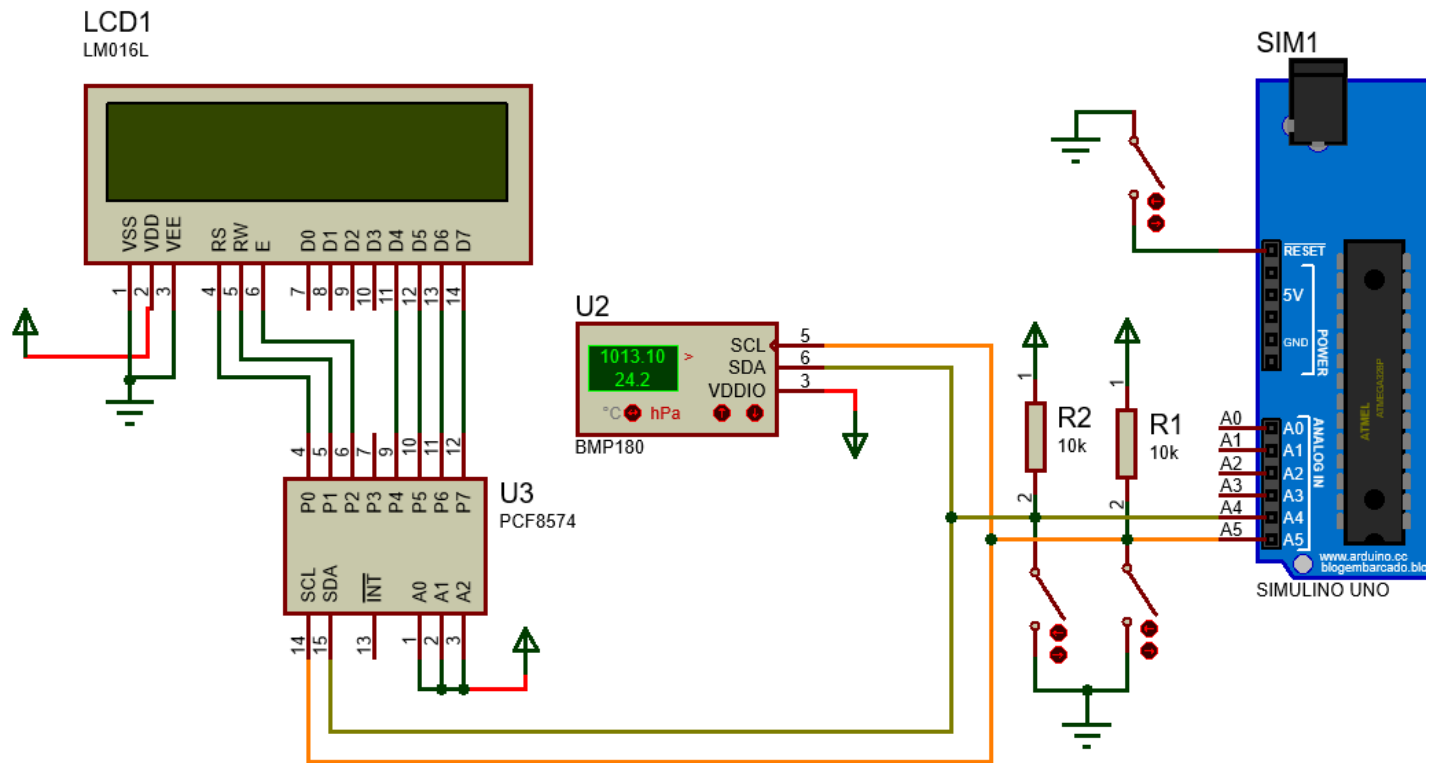


Figura 2. Montaje en pines Analógicos

2.2 PROGRAMACIÓN EN ARDUINO IDE

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085.h> // Para el sensor BMP180/BMP280
#include <DHT.h>
#include <LiquidCrystal_I2C.h> // Librería para pantalla LCD con I2C

#define DHTPIN 2 // Pin del sensor DHT
#define DHTTYPE DHT11 // Define si estás usando el DHT11
#define RAIN_PIN 3 // Pin del sensor de lluvia

DHT dht(DHTPIN, DHTTYPE); // Inicializa el sensor DHT
Adafruit_BMP085 bmp; // Inicializa el sensor BMP180/BMP280
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dirección I2C del LCD 16x2

void setup() {
    Serial.begin(9600); // Inicializa la comunicación serie
    lcd.init(); // Inicializa la pantalla LCD
    lcd.backlight(); // Enciende la retroiluminación

    dht.begin(); // Inicializa el sensor DHT
    if (!bmp.begin()) { // Inicializa el sensor BMP180
        Serial.println("Error BMP180!");
        while (1); // Detener si no se encuentra el sensor
    }

    pinMode(RAIN_PIN, INPUT); // Inicializa el pin del sensor de lluvia
}

void loop() {
    // Leer temperatura y humedad del DHT11
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    // Leer presión del BMP180
    float pressure = bmp.readPressure() / 100.0F; // Convertir a hPa

    // Leer estado del sensor de lluvia
    int rain = digitalRead(RAIN_PIN);
    String rainStatus = (rain == HIGH) ? "L1" : "NoL1"; // LL = Si hay lluvia, NLL = No hay lluvia

    // Mostrar los datos en el monitor serial
    Serial.print("T: "); Serial.print(t); Serial.print(" C "); // T = Temperatura, C = °C
    Serial.print("H: "); Serial.print(h); Serial.print(" % "); // H = Humedad
    Serial.print("P: "); Serial.print(pressure); Serial.println(" hPa"); // P = Presión
    Serial.println(rainStatus);

    // Mostrar los datos en la pantalla LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("T:"); lcd.print(t); lcd.print("C H:"); lcd.print(h); lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print("P:"); lcd.print(pressure); lcd.print("hPa");
    lcd.setCursor(0, 2);
    lcd.print(rainStatus);

    // Enviar los datos por Bluetooth
    Serial.print("Temp: "); Serial.print(t); Serial.print("C ");
    Serial.print("Hum: "); Serial.print(h); Serial.print("% ");
    Serial.print("Pressure: "); Serial.print(pressure); Serial.println("hPa");
    Serial.println(rainStatus);

    delay(2000); // Esperar 2 segundos entre lecturas
}
```

Explicación

- Wire.h: Se usa para la comunicación I2C entre el Arduino y los dispositivos como el BMP180 y la pantalla LCD.
- Adafruit_Sensor.h y Adafruit_BMP085.h: Son necesarias para el sensor de presión atmosférica BMP180 (o BMP280).
- DHT.h: Controla el sensor de temperatura y humedad DHT11.
- LiquidCrystal_I2C.h: Para manejar la pantalla LCD con comunicación I2C.
- DHTPIN: Define el pin digital donde está conectado el sensor DHT11 (pin 2 en este caso).
- DHTTYPE: Define el tipo de sensor DHT, en este caso DHT11.
- RAIN_PIN: Define el pin donde se conecta el sensor de lluvia (pin 3).
- dht: Inicializa el sensor DHT11 en el pin 2.
- bmp: Inicializa el objeto para manejar el sensor BMP180.
- lcd: Inicializa la pantalla LCD con la dirección I2C 0x27 (puede variar dependiendo del hardware).
- Serial.begin(9600): Configura la velocidad de la comunicación serie a 9600 baudios para enviar datos al monitor serie o a un módulo Bluetooth.
- lcd.init(): Inicializa la pantalla LCD.
- lcd.backlight(): Enciende la retroiluminación de la pantalla LCD.
- dht.begin(): Inicializa el sensor DHT11 para comenzar a leer temperatura y humedad.
- bmp.begin(): Inicializa el sensor de presión BMP180. Si falla, el código entra en un bucle infinito y envía un mensaje de error al monitor serie.
- pinMode(RAIN_PIN, INPUT): Configura el pin 3 como entrada, donde se conecta el sensor de lluvia.
- float h = dht.readHumidity();: Lee la humedad del sensor DHT11.
- float t = dht.readTemperature();: Lee la temperatura del sensor DHT11.
- float pressure = bmp.readPressure() / 100.0F;: Lee la presión del sensor BMP180 y la convierte a hectopascales (hPa).
- int rain = digitalRead(RAIN_PIN);: Lee el estado del sensor de lluvia. Si el valor leído es HIGH, significa que no hay lluvia. Si es LOW, significa que hay lluvia.
- String rainStatus = (rain == HIGH) ? "LI" : "NoLI";: Usa el operador ternario para asignar el estado de lluvia. "LI" significa que está lloviendo, y "NoLI" significa que no hay lluvia.
- lcd.clear(): Limpia la pantalla LCD antes de actualizar los valores.
- lcd.setCursor(0, 0): Coloca el cursor en la posición (columna 0, fila 0) para escribir los datos de temperatura y humedad en la primera fila.
- lcd.setCursor(0, 1): Coloca el cursor en la posición (columna 0, fila 1) para mostrar la presión y el estado de la lluvia en la segunda fila.

2. ANÁLISIS DE RESULTADOS

- ¿Qué parámetros mide la estación meteorológica y cómo se presentan?
 - Temperatura: El sensor DHT11 mide la temperatura y la muestra en grados Celsius tanto en el monitor serie como en la pantalla LCD.
 - Humedad: El mismo sensor DHT11 mide la humedad relativa y la muestra en porcentaje. Estos valores también se envían al monitor serie y la pantalla LCD.
 - Presión atmosférica: El sensor BMP180 mide la presión atmosférica, que se convierte a hectopascales (hPa) y se muestra en la pantalla LCD y en el monitor serie.
 - Lluvia: El sensor de lluvia detecta si hay agua en el ambiente, mostrando "LI" en caso de lluvia y "NoLI" en caso de no haberla. Esta información también se imprime en el monitor serie y en la pantalla.
- ¿Cómo se interpretan los datos proporcionados por los sensores?
 - Temperatura: Se puede verificar si las temperaturas leídas son consistentes con las condiciones reales del entorno.
 - Humedad: Se puede observar si los niveles de humedad están dentro de los rangos normales o si son altos/bajos.
 - Presión atmosférica: Este dato es útil para predecir cambios climáticos, ya que las bajas presiones suelen estar asociadas con mal tiempo.
 - Lluvia: El sensor proporciona una indicación simple sobre la presencia de lluvia, lo que puede ser útil para sistemas de alerta o monitoreo del clima en tiempo real.
- ¿Cuál es la frecuencia de actualización de los datos?

La frecuencia de actualización es de 2 segundos (definida por delay(2000)), lo que proporciona una lectura de datos cada 2 segundos. Esta frecuencia es suficiente para monitorear cambios climáticos en tiempo real sin sobrecargar el sistema.

- ¿Cómo interactúa el usuario con el sistema?

El usuario puede visualizar los datos tanto en la pantalla LCD como en el monitor serie del Arduino IDE. Además, si hay un módulo Bluetooth conectado, los datos también pueden ser enviados a una aplicación móvil o un dispositivo externo para su monitoreo remoto.

- ¿Cómo maneja el sistema los errores o fallas?

Si el sensor BMP180 no se inicializa correctamente, el programa envía un mensaje de error al monitor serie ("Error BMP180!") y detiene la ejecución del código (while (1);). Esto asegura que el sistema no funcione incorrectamente sin advertir al usuario sobre un mal funcionamiento del sensor de presión.

- ¿El sistema es escalable?

Sí, el sistema puede ampliarse para incluir otros sensores meteorológicos (como un sensor de velocidad del viento o radiación solar), ya que la estructura del código es modular. Además, el uso de comunicación I2C permite añadir más sensores sin muchos conflictos de pines.

- ¿Qué tan precisos son los sensores utilizados?

- DHT11: Tiene una precisión moderada para temperatura y humedad, pero no es tan preciso como el DHT22.
- BMP180: Es muy preciso para medir la presión atmosférica, con una resolución de hasta 0.01 hPa.
- Sensor de lluvia: Es un sensor simple que detecta si hay agua presente, pero no mide la cantidad de lluvia.

3. CONCLUSIONES

- a. La estación meteorológica diseñada es capaz de medir múltiples parámetros ambientales (temperatura, humedad, presión atmosférica y lluvia) y presentarlos de manera eficaz tanto en una pantalla LCD como en un monitor serie o por medio de Bluetooth. Esto permite al usuario tener acceso a datos climáticos en tiempo real.
- b. El sistema es modular, lo que significa que se pueden añadir o modificar sensores fácilmente, según las necesidades del usuario. Esto lo hace adecuado para proyectos futuros que puedan incluir más variables meteorológicas o integrar nuevas tecnologías como Internet de las Cosas (IoT).
- c. El hecho de que los datos se muestren en tiempo real y en diferentes interfaces (pantalla LCD, monitor serie y Bluetooth) ofrece flexibilidad. El uso de Bluetooth abre la posibilidad de integrar la estación con aplicaciones móviles para un monitoreo remoto, mejorando la utilidad del sistema para proyectos de campo o uso doméstico.
- d. El sensor DHT11 es menos preciso en comparación con otros modelos como el DHT22. Si se necesita mayor precisión en la medición de temperatura y humedad, se podría optar por un sensor más avanzado.

- e. El sensor de lluvia es una solución simple que solo detecta si hay o no agua, pero no mide la intensidad de la lluvia. En aplicaciones donde es necesario cuantificar la cantidad de precipitación, sería mejor optar por un pluviómetro o un sensor más sofisticado.
- f. Mejoras:
- Se podrían añadir sensores para medir la velocidad y dirección del viento, la radiación solar, y la cantidad exacta de lluvia para obtener una estación meteorológica más completa.
 - Dependiendo de las necesidades, la frecuencia de muestreo podría ajustarse para ahorrar energía o hacer el sistema más eficiente.
 - Sería útil registrar los datos climáticos a lo largo del tiempo en una tarjeta SD o enviarlos a un servidor remoto para análisis a largo plazo.