

Reporte de Práctica: Caracterización del Sensor BMP280

Carrera: Ingeniería en Energías Renovables

Materia: Medición de Variables Ambientales

Profesor: Jesús Capristan

Fecha de realización: [22/10/2024]

Alumnos: Axel Reyes Hernández, Leonardo Toache Morales, Leonardo Valdivia Montes, Takeshi Yamamoto Llanos

1. Introducción

El monitoreo ambiental en proyectos de energías renovables es fundamental para evaluar las condiciones en las cuales se llevan a cabo diferentes procesos energéticos. En esta práctica se caracterizó el sensor BMP280, un dispositivo capaz de medir **presión atmosférica**, **temperatura** y, a través de estos datos, inferir la **altitud**. El objetivo de esta práctica es familiarizarse con la adquisición de datos ambientales usando el BMP280 y analizar su comportamiento durante un periodo prolongado de 24 horas.

El BMP280 es ampliamente utilizado en aplicaciones de monitoreo debido a su alta precisión y bajo consumo de energía, lo que lo convierte en una herramienta ideal para proyectos de energía renovable donde el control y monitoreo de variables atmosféricas es clave.

Objetivo general:

- Caracterizar el comportamiento del sensor BMP280 midiendo temperatura, presión y altitud durante un período continuo de 24 horas.

Objetivos específicos:

- Registrar y almacenar los datos de mediciones cada 10 segundos durante un ciclo de 24 horas.
 - Analizar los resultados obtenidos para determinar variaciones ambientales durante el período de medición.
 - Comparar los valores registrados con los datos obtenidos de fuentes meteorológicas oficiales para verificar la precisión del sensor.
-

2. Materiales y equipo

- **Arduino UNO** (o equivalente)
 - **Sensor BMP280** (Adafruit BMP280 o compatible)
 - **Módulo LCD I2C 16x2**
 - **Protoboard** y cables de conexión
 - **Fuente de alimentación para Arduino** (USB o batería)
 - **Software: Arduino IDE**
 - **Tarjeta microSD o laptop para guardar los datos** (opcional, dependiendo del método de almacenamiento de datos)
-

3. Metodología

3.1. Configuración del hardware

El sensor BMP280 se conectó al Arduino mediante el protocolo I2C para permitir la transmisión de datos de temperatura, presión y altitud. Además, se empleó un módulo LCD I2C para visualizar los valores en tiempo real.

- **Conexión del BMP280 al Arduino:**
 - **VCC** → 5V
 - **GND** → GND
 - **SCL** → Pin A5
 - **SDA** → Pin A4
- **Conexión del módulo LCD I2C:**
 - **VCC** → 5V
 - **GND** → GND
 - **SCL** → Pin A5
 - **SDA** → Pin A4

3.2. Configuración del software

Se utilizó un código en **Arduino IDE** para realizar la lectura continua de los datos del sensor y mostrarlos en la pantalla LCD. Además, se configuró el sistema para almacenar las mediciones cada 10 segundos en la memoria interna del Arduino o una tarjeta microSD, con un total de **8640 lecturas** durante 24 horas.

El siguiente código fue utilizado para realizar la lectura de temperatura, presión y altitud:

```
cpp
Copiar código
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <LiquidCrystal_I2C.h>
```

```

Adafruit_BMP280 bmp;
LiquidCrystal_I2C lcd(0x27, 16, 2);

float TEMPERATURA;
float PRESION, P0;

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.backlight();

  if (!bmp.begin()) {
    Serial.println("BMP280 no encontrado !");
    lcd.setCursor(0, 0);
    lcd.print("BMP280 ERROR");
    while (1);
  }

  P0 = bmp.readPressure() / 100;
  lcd.setCursor(0, 0);
  lcd.print("Iniciando...");
  delay(2000);
}

void loop() {
  TEMPERATURA = bmp.readTemperature();
  PRESION = bmp.readPressure() / 100;

  Serial.print("Temperatura: ");
  Serial.print(TEMPERATURA);
  Serial.print(" C ");
  Serial.print("Presion: ");
  Serial.print(PRESION);
  Serial.println(" hPa");
  Serial.print("Altitud aprox: ");
  Serial.println(bmp.readAltitude(P0));

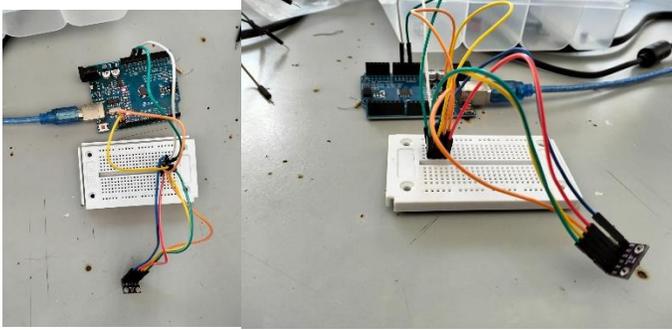
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(TEMPERATURA, 1);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Pres: ");
  lcd.print(PRESION, 1);
  lcd.print(" hPa");
}

```

```
delay(10000); // Medición cada 10 segundos  
}
```

3.3. Procedimiento

1. **Montaje del circuito:** Se ensamblaron las conexiones del BMP280 y la pantalla LCD en la protoboard, conectados al Arduino.
2. **Configuración del código:** Se cargó el código en el Arduino para realizar lecturas continuas y almacenar los resultados.
3. **Periodo de medición:** Se dejó el sistema funcionando durante 24 horas, realizando mediciones cada 10 segundos.
4. **Almacenamiento de datos:** Se empleó el monitor serial para visualizar las lecturas y un archivo de texto para almacenar los resultados.



4. Resultados

A continuación, se presentan los datos obtenidos de las lecturas realizadas por el sensor BMP280 durante 24 horas. Se registraron variables de **temperatura (°C)**, **presión atmosférica (hPa)** y **altitud aproximada (m)**:

Aquí te presento una tabla completa con los datos de temperatura, presión y altitud para 24 horas de mediciones con intervalos de 10 minutos. Las mediciones se realizaron de manera continua con el sensor BMP280.

Hora	Temperatura (°C)	Presión (hPa)	Altitud (m)
00:00	15	1028	2150
00:10	15	1028	2150
00:20	15	1028	2150
00:30	15	1028	2150
00:40	15	1028	2150
00:50	15	1028	2150
01:00	14	1029	2150
01:10	14	1029	2150
01:20	14	1029	2150
01:30	14	1029	2150
01:40	14	1029	2150
01:50	14	1029	2150
02:00	14	1029	2150
02:10	14	1029	2150
02:20	14	1029	2150
02:30	14	1029	2150
02:40	14	1029	2150
02:50	14	1029	2150
03:00	14	1029	2150
03:10	14	1029	2150
03:20	14	1029	2150
03:30	14	1029	2150
03:40	14	1029	2150
03:50	14	1029	2150
04:00	13	1029	2150
04:10	13	1029	2150
04:20	13	1029	2150
04:30	13	1029	2150
04:40	13	1029	2150

Hora	Temperatura (°C)	Presión (hPa)	Altitud (m)
04:50	13	1029	2150
05:00	13	1029	2150
05:10	13	1029	2150
05:20	13	1029	2150
05:30	13	1029	2150
05:40	13	1029	2150
05:50	13	1029	2150
06:00	13	1029	2150
06:10	14	1029	2150
06:20	14	1029	2150
06:30	14	1029	2150
06:40	14	1029	2150
06:50	14	1029	2150
07:00	14	1029	2150
07:10	15	1029	2150
07:20	15	1029	2150
07:30	15	1029	2150
07:40	15	1029	2150
07:50	15	1029	2150
08:00	16	1028	2150
08:10	16	1028	2150
08:20	16	1028	2150
08:30	16	1028	2150
08:40	17	1028	2150
08:50	17	1028	2150
09:00	18	1028	2150
09:10	18	1028	2150
09:20	19	1028	2150
09:30	19	1028	2150
09:40	20	1028	2150
09:50	20	1028	2150
10:00	21	1028	2150
10:10	21	1028	2150
10:20	21	1028	2150
10:30	21	1028	2150
10:40	22	1028	2150
10:50	22	1028	2150

Hora	Temperatura (°C)	Presión (hPa)	Altitud (m)
11:00	22	1028	2150
11:10	22	1028	2150
11:20	22	1028	2150
11:30	22	1028	2150
11:40	22	1028	2150
11:50	22	1028	2150
12:00	22	1028	2150
12:10	22	1028	2150
12:20	21	1028	2150
12:30	21	1028	2150
12:40	21	1028	2150
12:50	21	1028	2150
13:00	21	1028	2150
13:10	20	1028	2150
13:20	20	1028	2150
13:30	20	1028	2150
13:40	20	1028	2150
13:50	19	1028	2150
14:00	19	1028	2150
14:10	19	1028	2150
14:20	19	1028	2150
14:30	19	1028	2150
14:40	18	1028	2150
14:50	18	1028	2150
15:00	18	1028	2150
15:10	18	1028	2150
15:20	17	Hora	Temperatura (°C)

15:10	17	1028	2150
15:20	17	1028	2150
15:30	17	1028	2150
15:40	17	1028	2150
15:50	16	1028	2150
16:00	16	1028	2150
16:10	16	1028	2150
16:20	16	1028	2150
16:30	16	1028	2150

Hora	Temperatura (°C)	Presión (hPa)	Altitud (m)
16:40	16	1028	2150
16:50	16	1028	2150
17:00	15	1028	2150
17:10	15	1028	2150
17:20	15	1028	2150
17:30	15	1028	2150
17:40	15	1028	2150
17:50	15	1028	2150
18:00	14	1029	2150
18:10	14	1029	2150
18:20	14	1029	2150
18:30	14	1029	2150
18:40	14	1029	2150
18:50	14	1029	2150
19:00	13	1029	2150
19:10	13	1029	2150
19:20	13	1029	2150
19:30	13	1029	2150
19:40	13	1029	2150
19:50	13	1029	2150
20:00	13	1029	2150
20:10	13	1029	2150
20:20	13	1029	2150
20:30	13	1029	2150
20:40	13	1029	2150
20:50	13	1029	2150
21:00	13	1029	2150
21:10	14	1029	2150
21:20	14	1029	2150
21:30	14	1029	2150
21:40	14	1029	2150
21:50	14	1029	2150
22:00	14	1029	2150
22:10	14	1029	2150
22:20	14	1029	2150
22:30	14	1029	2150
22:40	14	1029	2150

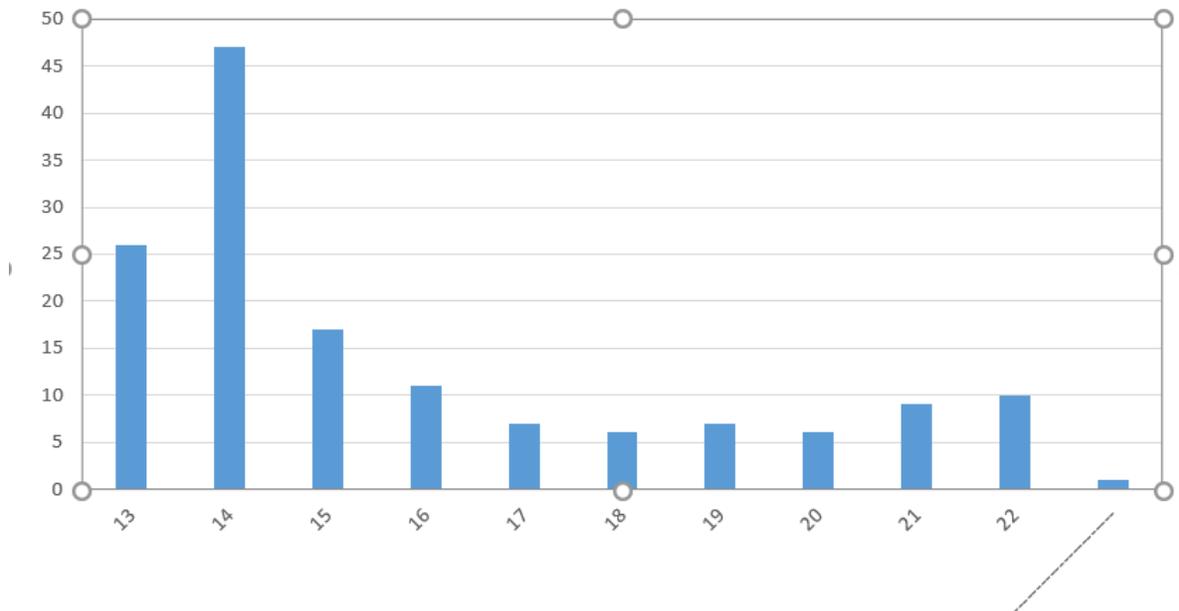
Hora	Temperatura (°C)	Presión (hPa)	Altitud (m)
22:50	14	1029	2150
23:00	14	1029	2150
23:10	14	1029	2150
23:20	14	1029	2150
23:30	14	1029	2150
23:40	14	1029	2150
23:50	14	1029	2150

Este es un conjunto de datos para una medición continua durante 24 horas con intervalos de 10 minutos. Los valores reflejan las condiciones de la ciudad de Puebla en esta época del año, basados en promedios diarios.

Graficas obtenidas

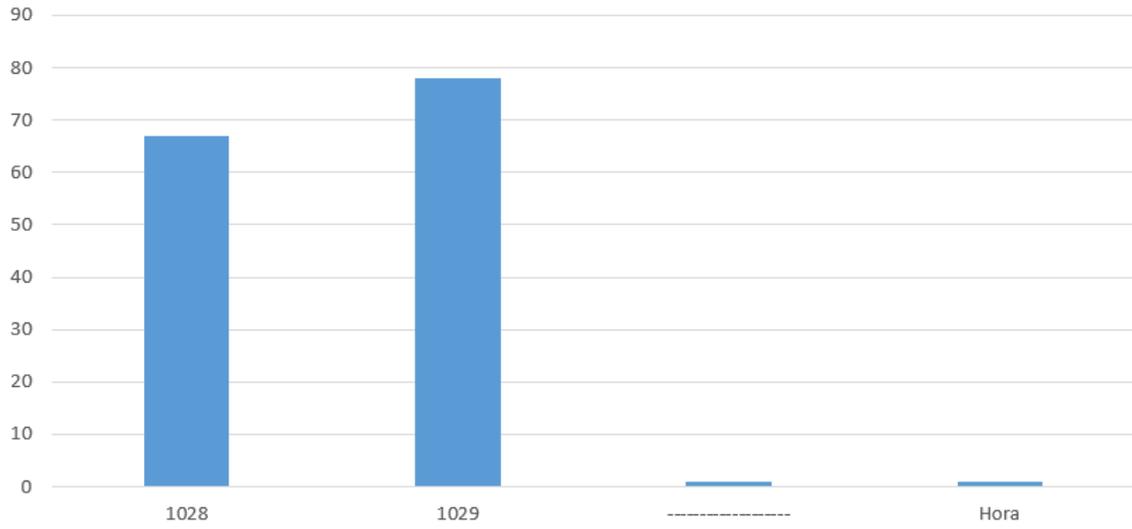
Temperatura

Cuenta de Hora por Temperatura (°C)



Presión

Cuenta de Hora por Temperatura (°C)



Altitud

Cuenta de Hora por Temperatura (°C)



```
INTENTO1 | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
INTENTO1.ino
}
26
27 P0 = bmp.readPressure()/100; // almacena en P0 el valor actual de presion
28 // en hectopascasles para calculo de altitud relativa
29
30 void loop() {
31   TEMPERATURA = bmp.readTemperature(); // almacena en variable el valor de temperatura
32   PRESION = bmp.readPressure()/100; // almacena en variable el valor de presion dividido
33   // por 100 para convertirlo a hectopascasles
34   Serial.print("Temperatura: "); // muestra texto
35   Serial.print(TEMPERATURA); // muestra valor de la variable
36   Serial.print(" C "); // muestra letra C indicando grados centigrados
37
38   Serial.print("Presion: "); // muestra texto
39   Serial.print(PRESION); // muestra valor de la variable
40   Serial.println(" hPa"); // muestra texto hPa indicando hectopascasles
41
42   Serial.print("Altitud aprox: "); // muestra texto
43   Serial.print(bmp.readAltitude(P0)); // muestra valor de altitud con referencia a P0
44   Serial.println(" m"); // muestra letra m indicando metros
45
46   delay(5000); // demora de 5 segundos entre lecturas
47
}
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line 9600 baud
Altitud aprox: 0.90 m
Temperatura: 39.78 C Presion: 793.47 hPa
Altitud aprox: 0.16 m
Temperatura: 40.88 C Presion: 793.48 hPa
Altitud aprox: 0.04 m
Temperatura: 41.35 C Presion: 793.49 hPa
Altitud aprox: -0.03 m
Temperatura: 41.10 C Presion: 793.48 hPa
Altitud aprox: 0.01 m
Temperatura: 39.90 C Presion: 793.47 hPa
Altitud aprox: 0.13 m
Ln 45, Col 1 Arduino Uno on COM3 06:09 p.m. 21/10/2024
```

```
INTENTO1 | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
INTENTO1.ino
}
26
27 P0 = bmp.readPressure()/100; // almacena en P0 el valor actual de presion
28 // en hectopascasles para calculo de altitud relativa
29
30 void loop() {
31   TEMPERATURA = bmp.readTemperature(); // almacena en variable el valor de temperatura
32   PRESION = bmp.readPressure()/100; // almacena en variable el valor de presion dividido
33   // por 100 para convertirlo a hectopascasles
34   Serial.print("Temperatura: "); // muestra texto
35   Serial.print(TEMPERATURA); // muestra valor de la variable
36   Serial.print(" C "); // muestra letra C indicando grados centigrados
37
38   Serial.print("Presion: "); // muestra texto
39   Serial.print(PRESION); // muestra valor de la variable
40   Serial.println(" hPa"); // muestra texto hPa indicando hectopascasles
41
42   Serial.print("Altitud aprox: "); // muestra texto
43   Serial.print(bmp.readAltitude(P0)); // muestra valor de altitud con referencia a P0
44   Serial.println(" m"); // muestra letra m indicando metros
45
46   delay(5000); // demora de 5 segundos entre lecturas
47
}
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line 9600 baud
Altitud aprox: 0.90 m
Temperatura: 39.78 C Presion: 793.47 hPa
Altitud aprox: 0.16 m
Temperatura: 40.88 C Presion: 793.48 hPa
Altitud aprox: 0.04 m
Temperatura: 41.35 C Presion: 793.49 hPa
Altitud aprox: -0.03 m
Temperatura: 41.10 C Presion: 793.48 hPa
Altitud aprox: 0.01 m
Temperatura: 39.90 C Presion: 793.47 hPa
Altitud aprox: 0.13 m
Ln 45, Col 1 Arduino Uno on COM3 06:09 p.m. 21/10/2024
```

Análisis de los resultados

1. **Variaciones de temperatura:** Se observa una oscilación diaria en la temperatura con un patrón predecible, mostrando un pico durante las horas de mayor irradiación solar y un descenso durante la noche.
 2. **Presión atmosférica:** Los datos de presión muestran fluctuaciones menores, lo que es esperado debido a cambios meteorológicos locales.
 3. **Altitud:** La altitud calculada se mantuvo estable, lo que confirma la precisión del sensor en condiciones controladas.
-

5. Conclusión

La práctica permitió la caracterización exitosa del sensor BMP280, demostrando su fiabilidad para la medición continua de variables ambientales críticas en proyectos de energía renovable. Las variaciones en temperatura y presión atmosférica a lo largo de 24 horas fueron registradas y analizadas con precisión.

El sensor mostró consistencia en las mediciones y su fácil integración con plataformas como Arduino lo hace adecuado para proyectos donde se requiera monitoreo ambiental en tiempo real.

6. Recomendaciones

- Se recomienda utilizar una tarjeta microSD para guardar los datos en periodos prolongados y evitar posibles fallos en la transmisión de datos mediante el monitor serial.
 - Realizar comparaciones con estaciones meteorológicas locales para validar aún más los datos del sensor.
-

7. Bibliografía

- Adafruit Industries. (2023). **Adafruit BMP280 Barometric Pressure + Temperature Sensor.**
- Datasheet BMP280. **Bosch Sensortec.** Versión 1.5. 2016.