lamaPLC: Arduino Modul: BME680

The program read BME-680 sensor to serial monitor.

Install the Adafruit BME680 Library (this will also prompt you to install the Adafruit Unified Sensor and Adafruit BME680 Library, which you need)

```
/*
 * BME680 Sensor - Arduino Uno/Nano
 * Connection: I2C
 * Data: Temperature, Humidity, Pressure, Gas (Air Quality)
 *
 * Required Libraries (Arduino Library Manager):
 *   - "Adafruit BME680 Library" by Adafruit
 *   - "Adafruit Unified Sensor" by Adafruit
 *
 * Wiring (I2C):
 *   BME680  -->  Arduino Uno/Nano
 *   VCC     -->  3.3V (or 5V depending on module)
 *   GND     -->  GND
 *   SDA     -->  A4
 *   SCL     -->  A5
 */

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME680.h>

// Sea level pressure for altitude calculation (adjust to your location in
hPa!)
#define SEA_LEVEL_PRESSURE_HPA 1013.25

// Create BME680 object (I2C address: 0x76 or 0x77)
Adafruit_BME680 bme;

void setup() {
  Serial.begin(9600);
  while (!Serial);  // Wait until Serial is ready

  Serial.println(F("============================="));
  Serial.println(F("  BME680 Sensor - Arduino   "));
  Serial.println(F("============================="));

  // Initialize sensor
  if (!bme.begin(0x76)) {
    Serial.println(F("ERROR: BME680 not found!"));
    Serial.println(F("Please check your wiring."));
    Serial.println(F("Trying address 0x77 ..."));
```

```
    if (!bme.begin(0x77)) {
      Serial.println(F("ERROR: BME680 not found on 0x77 either!"));
      while (1); // Halt program
    }
  }

  Serial.println(F("BME680 found successfully!"));

  // Configure sensor settings
  bme.setTemperatureOversampling(BME680_OS_8X);
  bme.setHumidityOversampling(BME680_OS_2X);
  bme.setPressureOversampling(BME680_OS_4X);
  bme.setIIRFilterSize(BME680_FILTER_SIZE_3);
  bme.setGasHeater(320, 150); // Heater temperature: 320°C, Heating time:
150ms

  Serial.println(F("Sensor configured. Starting measurements...\n"));
  delay(2000);
}

void loop() {
  // Start measurement and wait for result
  if (!bme.performReading()) {
    Serial.println(F("ERROR: Reading failed!"));
    delay(2000);
    return;
  }

  // Print sensor values
  Serial.println(F("----------------------------"));

  // Temperature
  Serial.print(F("Temperature:      "));
  Serial.print(bme.temperature, 1);
  Serial.println(F(" °C"));

  // Humidity
  Serial.print(F("Humidity:         "));
  Serial.print(bme.humidity, 1);
  Serial.println(F(" %"));

  // Pressure
  Serial.print(F("Pressure:         "));
  Serial.print(bme.pressure / 100.0, 2);
  Serial.println(F(" hPa"));

  // Altitude above sea level (calculated)
  Serial.print(F("Altitude (approx): "));
  Serial.print(bme.readAltitude(SEA_LEVEL_PRESSURE_HPA), 1);
  Serial.println(F(" m"));
```

```
  // Gas resistance (air quality)
  Serial.print(F("Gas Resistance:   "));
  Serial.print(bme.gas_resistance / 1000.0, 2);
  Serial.println(F(" kΩ"));

  // Air quality as simple rating
  Serial.print(F("Air Quality:      "));
  Serial.println(getAirQuality(bme.gas_resistance));

  Serial.println();

  // Wait 3 seconds until next measurement
  delay(3000);
}

// Simple air quality rating based on gas resistance
String getAirQuality(uint32_t gasResistance) {
  uint32_t kOhm = gasResistance / 1000;

  if (kOhm >= 300) {
    return F("Very Good");
  } else if (kOhm >= 150) {
    return F("Good");
  } else if (kOhm >= 50) {
    return F("Moderate");
  } else if (kOhm >= 10) {
    return F("Poor");
  } else {
    return F("Very Poor");
  }
}
```

code!, c, 2026, Arduino, BME680, sensor, i2c, comunication