

LamaPLC: AHT20 / BMP280 Modul

This compact module combines an AHT20 air temperature and humidity sensor with a BMP280 air pressure sensor to provide a comprehensive view of the air conditions in your environment. The AHT20 and BMP280 are widely used and popular due to their accuracy and small size. Communication is easily handled via the **I²C** interface with a microcontroller of your choice, thanks to the operating voltage of 2-5V.



Technical Details

- Operating voltage: DC 2..5V
- Sensors: AHT20, BMP280
- Dimensions: 16 x 16.5 x 3 mm
- Weight: 4 g
- *I²C Addresses*
 - BMP280: 0x77
 - AHT20: 0x38
- *Air pressure*
 - Measurement range: 300 to 1100 hPa
 - Resolution: 0.18 Pa
 - Accuracy: ± 1 Pa
- *Humidity*
 - Resolution: 0.024 %RH
 - Tolerance: ± 2 %RH
 - Reproducibility: ± 0.1 %RH
 - Response time: 8 seconds
 - Measurement range: 0-100 %RH
 - Long-term drift: ± 0.5 %RH/year
- *Temperature*
 - Resolution: 0.01°C
 - Tolerance: ± 0.3 °C
 - Reproducibility: ± 0.1 °C
 - Response time: minimum 5 seconds, maximum 30 seconds
 - Measurement range: -40°C to 85°C
 - Long-term drift: ± 0.04 °C/year

Arduino

Arduino wiring

- SCL: A5
- GND: GND
- SDA: A4
- Vdd: 5V

Arduino code

For some reason, the Adafruit BMP280 library doesn't work reliably. It eventually freezes so severely that it only functions when the Arduino is powered off. I tried different parameter settings, but it didn't make a difference. It seems that the [BMx280MI](#) library written by Gregor Christandl works better.

```
// lamaPLC AHT20+BMP280 module test
// 2025.10.08. free code

#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_AHTX0.h>
#include <BMx280I2C.h>

#define I2C_ADDRESS 0x77

Adafruit_AHTX0 aht;

BMx280I2C bmx280(I2C_ADDRESS);

void(* resetFunc) (void) = 0; //declare reset function @ address 0

void setup() {
  Serial.begin(9600);
  Wire.begin();
  Serial.println("AHT20+BMP280 module test");

  if (!aht.begin()) {
    Serial.println("Could not find a valid AHT20 sensor, check wiring!");
    while (1);
  }

  if (!bmx280.begin()) {
    Serial.println("Could not find a valid BMP280 sensor, check wiring!");
    while (1);
  }

  bmx280.resetToDefaults();

  //set an oversampling setting for pressure and temperature measurements.
  bmx280.writeOversamplingPressure(BMx280MI::OSRS_P_x16);
  bmx280.writeOversamplingTemperature(BMx280MI::OSRS_T_x16);
}
```

```
Serial.println("AHT20 / BMP280 module ok");
}

void loop() {
  Serial.println("check -----");

  //start AHT20 measurement
  sensors_event_t humidity, temp;
  aht.getEvent(&humidity, &temp);
  Serial.print("AHT 20 Temperature: "); Serial.print(temp.temperature);
  Serial.println(" degrees C");
  Serial.print("AHT 20 Humidity: ");
  Serial.print(humidity.relative_humidity); Serial.println("% rH");

  //start BMP280 measurement
  if (!bmx280.measure())
  {
    Serial.println("could not start measurement, is a measurement
already running?");
    return;
  }

  //wait for the measurement to finish
  do
  {
    delay(100);
  } while (!bmx280.hasValue());

  Serial.print("BMP280 Pressure: "); Serial.print(bmx280.getPressure());
  Serial.println(" Pa");
  Serial.print("BMP280 Pressure (64 bit): ");
  Serial.print(bmx280.getPressure64()); Serial.println(" Pa");
  Serial.print("BMP280 Temperature: ");
  Serial.print(bmx280.getTemperature()); Serial.println(" degrees C");

  Serial.println();
  delay(5000);
}
```

code in github: <https://github.com/lamaPLC/AHT20-BMP280-Modul-Arduino-code>

Output on the serial monitor:

```
check -----
AHT 20 Temperature: 22.12 degrees C
AHT 20 Humidity: 62.55% rH
BMP280 Pressure: 99128.00 hPa
BMP280 Pressure (64 bit): 99125.84 hPa
```

BMP280 Temperature: 23.17 degrees C

Example codes

[Arduino - OLED SH1106 with AHT20/BMP280 Sensor](#)

[BMP280](#), [AHT20](#), [Adafruit](#), [temperature](#), [humidity](#), [pressure](#), [sensor](#), [arduino](#), [code](#)

This page has been accessed for: Today: 7, Until now: 134

From:

<http://lamaplc.com/> - **lamaPLC**

Permanent link:

http://lamaplc.com/doku.php?id=sensor:aht20_bmp280&rev=1770488836

Last update: **2026/02/07 18:27**

