BME280-3.3

BME280-3.3 Digital Module User Manual

Temperature, Barometric Pressure Sensor Module for Arduino

1. Introduction

The BME280-3.3 is a high-precision digital sensor module designed for measuring temperature, barometric pressure, and humidity. This module is an advanced upgrade to previous generations like the BMP085/BMP180/BMP183, offering enhanced accuracy and versatility. It supports both I2C and SPI communication interfaces, making it suitable for a wide range of microcontroller projects, including those based on Arduino platforms.

With its ability to measure barometric pressure with ±1 hPa absolute accuracy and temperature with ±1.0°C accuracy, the BME280-3.3 can also function as an altimeter with ±1 meter accuracy, as pressure changes with altitude. This makes it ideal for various applications from environmental monitoring to navigation systems.

2. Product Overview

The BME280-3.3 module features a compact design with a 1x6 pin header for easy integration. Below is an image illustrating the module's pinout and general appearance.

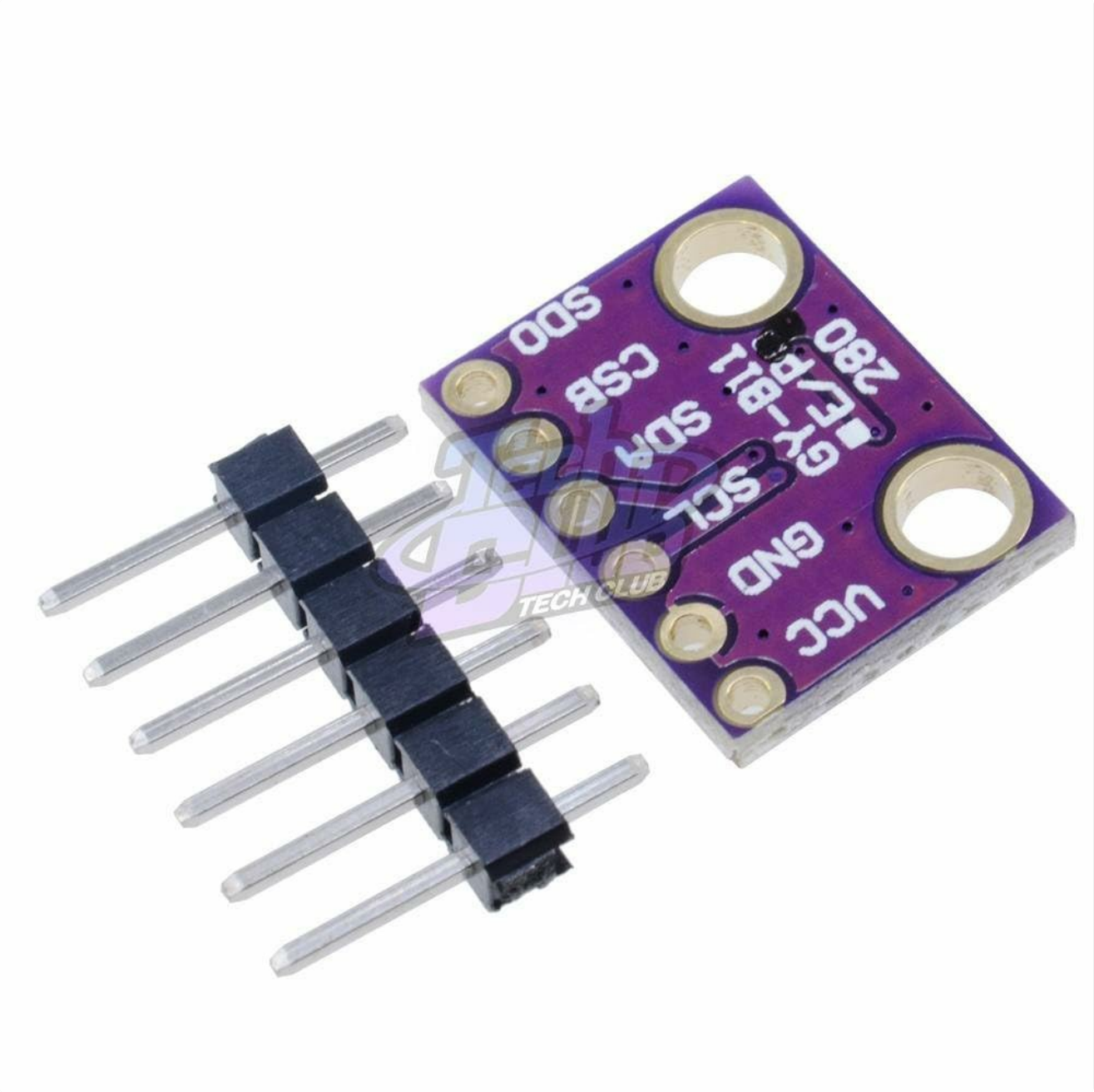

Figure 2.1: Top view of the BME280-3.3 module showing pin labels (VCC, GND, SCL, SDA, CSB, SDO).

The module's small form factor (1.52 cm x 1.20 cm) allows for integration into space-constrained projects. The pin header provides connections for power, ground, and communication lines.

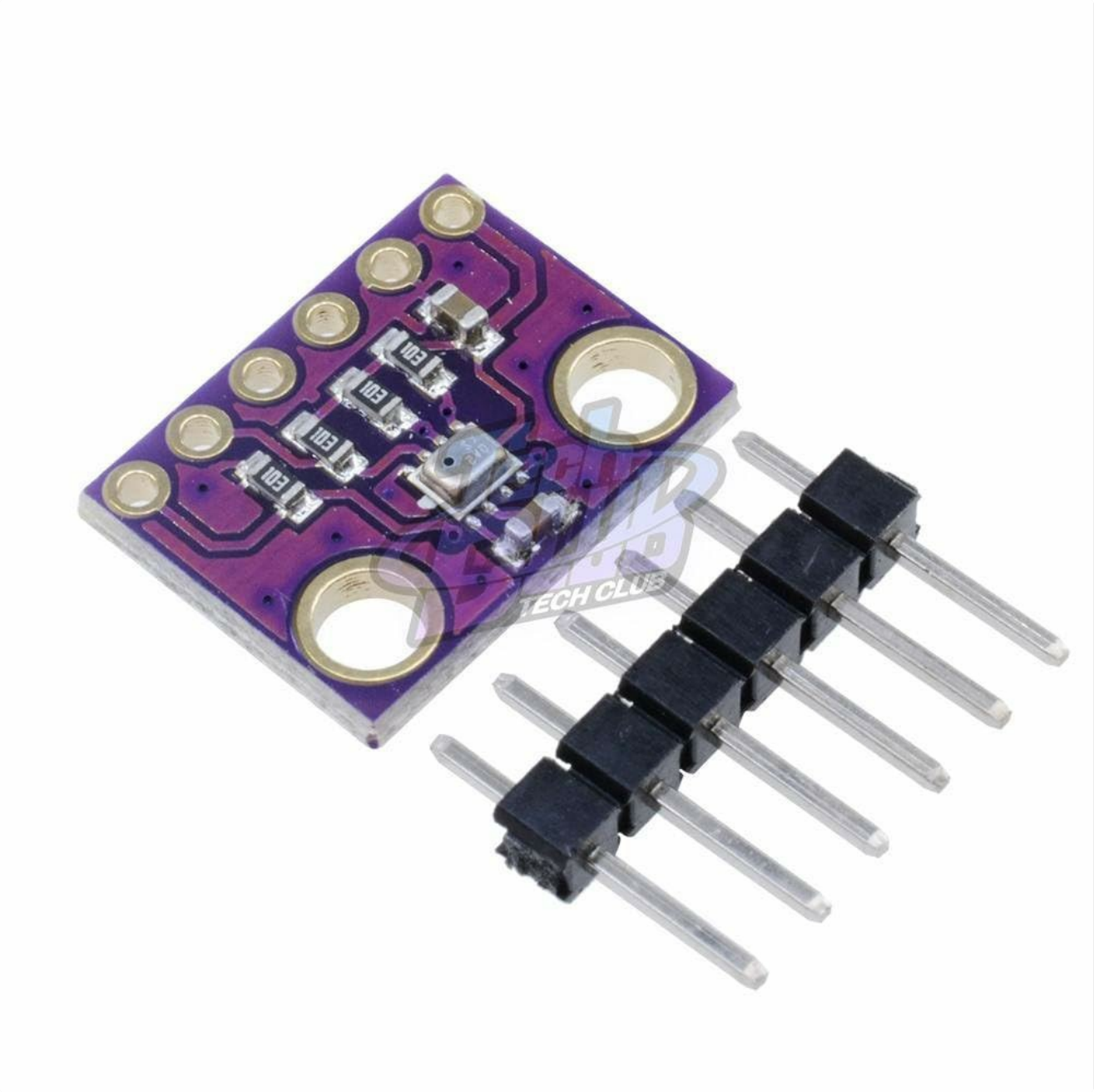

Figure 2.2: Angled view of the BME280-3.3 module with the 1x6 pin header attached, showing the compact size.

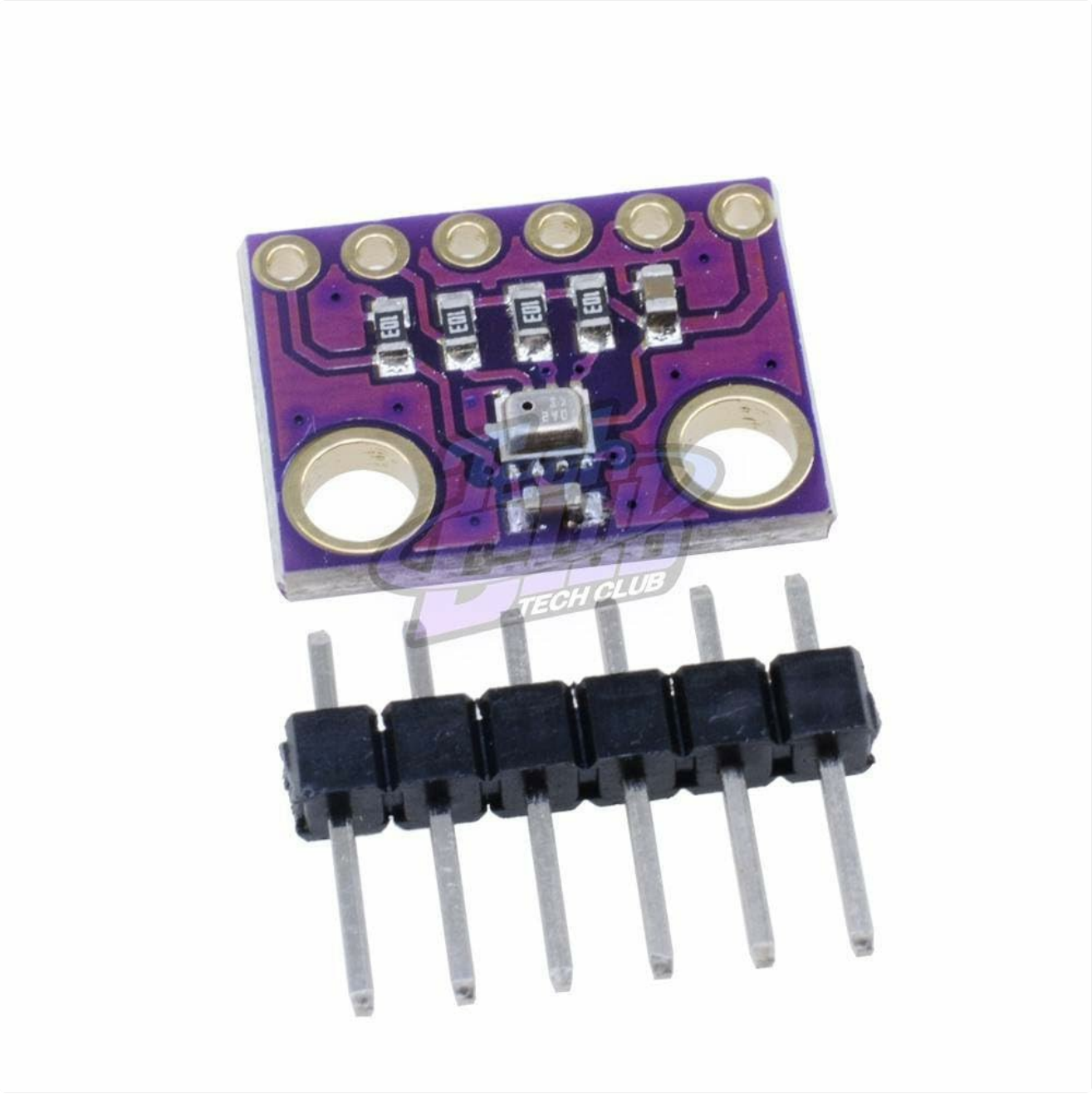

Figure 2.3: Top view of the BME280-3.3 module with the pin header detached, highlighting the solder points.

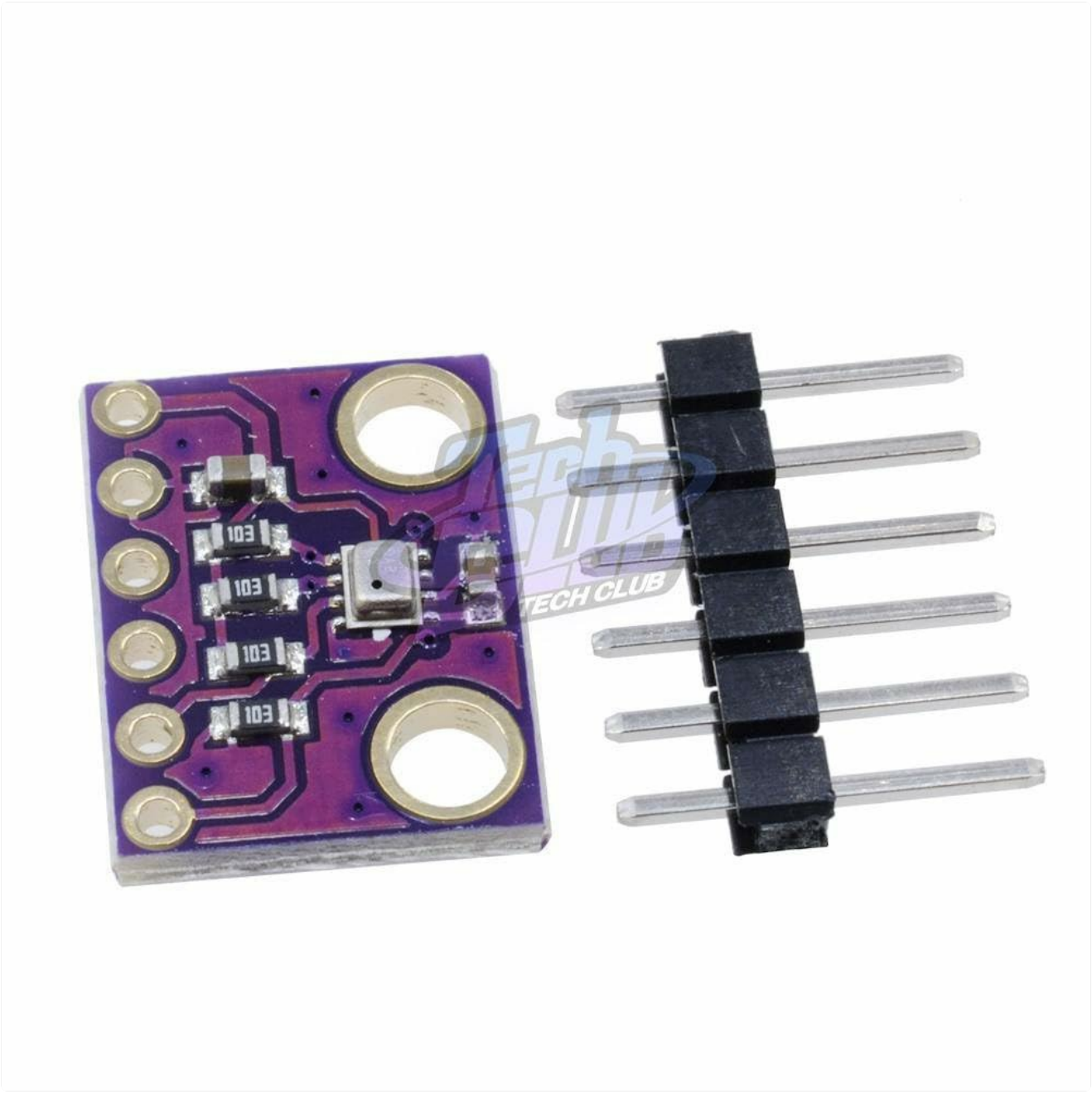

Figure 2.4: Side view of the BME280-3.3 module, showing its low profile.

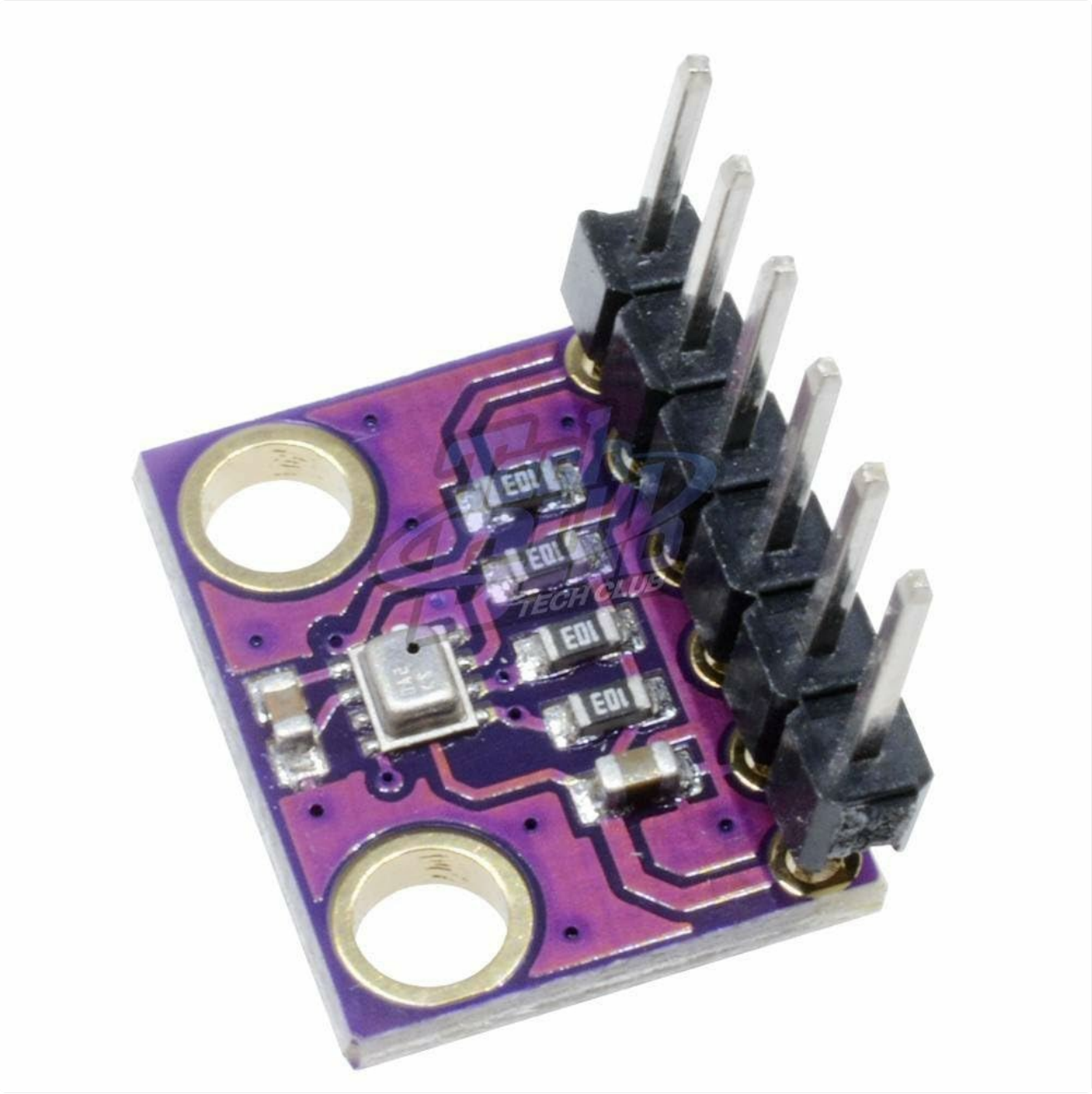

Figure 2.5: Bottom view of the BME280-3.3 module, showing the surface-mount components.

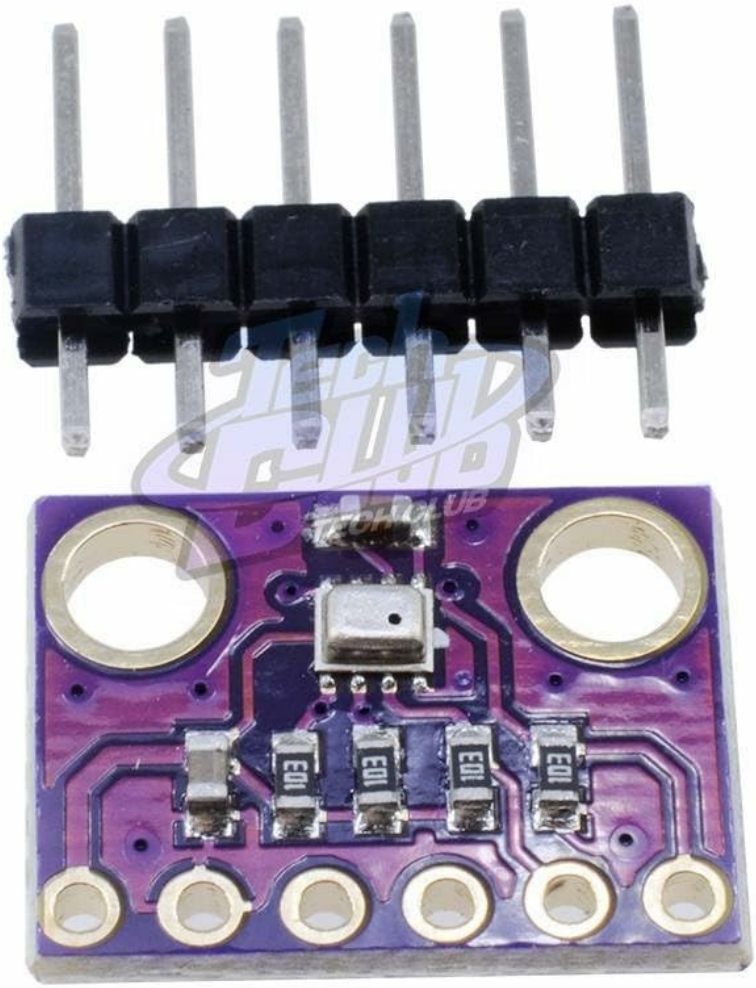

Figure 2.6: The BME280-3.3 module and its accompanying pin header shown separately.

3. Specifications

Feature	Description
Model	BME280-3.3 (compatible with BMP280)
Operating Voltage	3.3V
Sensor Type	Temperature, Barometric Pressure
Pressure Accuracy	±1 hPa (absolute)
Temperature Accuracy	±1.0°C
Altimeter Accuracy	±1 meter
Communication Interface	I2C, SPI
PCB Size (L x W)	1.52 cm x 1.20 cm (0.60" x 0.47")

4. Setup and Connection

This section outlines the basic steps for connecting the BME280-3.3 module to a microcontroller, such as an Arduino board. The module supports both I2C and SPI communication protocols.

4.1. Pinout Description

- **VCC:** Power supply input (3.3V).
- **GND:** Ground connection.
- **SCL:** I2C Clock Line / SPI Clock Line.
- **SDA:** I2C Data Line / SPI Master Out Slave In (MOSI).
- **CSB:** Chip Select for SPI mode. Connect to VCC for I2C mode.
- **SDO:** SPI Master In Slave Out (MISO). Connect to GND for I2C mode (default I2C address).

4.2. I2C Connection (Recommended for simplicity)

For I2C communication, connect the module to your Arduino board as follows:

- **VCC** → Arduino 3.3V
- **GND** → Arduino GND
- **SCL** → Arduino A5 (SCL on Uno/Nano), or SCL pin on other boards
- **SDA** → Arduino A4 (SDA on Uno/Nano), or SDA pin on other boards
- **CSB** → Arduino 3.3V (or VCC)
- **SDO** → Arduino GND

Note: Ensure your Arduino board can supply 3.3V. If using a 5V Arduino, use a logic level converter for SCL and SDA lines to prevent damage to the BME280-3.3 module.

4.3. SPI Connection

For SPI communication, connect the module to your Arduino board as follows:

- **VCC** → Arduino 3.3V
- **GND** → Arduino GND
- **SCL** → Arduino D13 (SCK on Uno/Nano)
- **SDA** → Arduino D11 (MOSI on Uno/Nano)
- **CSB** → Any digital pin (e.g., D10) for Chip Select
- **SDO** → Arduino D12 (MISO on Uno/Nano)

Note: When using SPI, the CSB pin must be actively controlled by the microcontroller.

5. Operating Principles and Applications

The BME280-3.3 module integrates a highly accurate MEMS sensor for environmental measurements. It uses internal compensation algorithms to provide precise temperature and pressure readings.

5.1. Data Acquisition

After proper connection, you will need to use a compatible library (e.g., Adafruit BME280 Library for Arduino) to communicate with the sensor. The library handles the low-level I2C or SPI communication, allowing you to easily read temperature, pressure, and altitude data.
Typical data readings include:

- Temperature in Celsius (°C) or Fahrenheit (°F)

- Barometric Pressure in Pascals (Pa), hectoPascals (hPa), or millibars (mbar)
- Altitude in meters or feet (calculated from pressure and sea-level pressure)

5.2. Typical Applications

The versatility and accuracy of the BME280-3.3 make it suitable for numerous applications:

- **Enhancement of GPS Navigation:** Improving time-to-first-fix, dead-reckoning, and slope detection.
- **Indoor Navigation:** Floor detection and elevator detection in multi-story buildings.
- **Outdoor Navigation, Leisure, and Sports:** Tracking altitude changes for hiking, cycling, or drone applications.
- **Weather Forecast:** Local weather station data collection.
- **Health Care Applications:** Such as spirometry (measuring lung function).
- **Vertical Velocity Indication:** Determining ascent/descent rates.
- **Target Devices:** Integration into mobile phones, tablet PCs, GPS devices, navigation systems, portable health care devices, home weather stations, flying toys, and watches.

6. Maintenance and Care

The BME280-3.3 module is a robust electronic component, but proper handling and care will ensure its longevity and accurate performance.

- **Environmental Protection:** Avoid exposing the module to extreme temperatures, high humidity, or direct water contact. While it measures environmental parameters, it is not designed for harsh outdoor conditions without proper enclosure.
- **Static Electricity:** Handle the module with care to prevent electrostatic discharge (ESD), which can damage sensitive electronic components. Use anti-static precautions when handling.
- **Physical Handling:** Avoid applying excessive force or bending the PCB. Do not touch the sensor element directly.
- **Cleaning:** If necessary, gently clean the PCB with a soft, dry brush or compressed air. Do not use liquid cleaners.
- **Power Supply:** Always ensure the correct operating voltage (3.3V) is supplied. Over-voltage can permanently damage the module.

7. Troubleshooting

If you encounter issues while using the BME280-3.3 module, consider the following troubleshooting steps:

- **No Readings / Sensor Not Detected:**
  - *Check Wiring:* Verify all connections (VCC, GND, SCL, SDA, CSB, SDO) are correct and secure according to your chosen communication protocol (I2C or SPI).
  - *Power Supply:* Ensure the module is receiving a stable 3.3V power supply.
  - *Logic Levels:* If using a 5V microcontroller, confirm you are using a logic level converter for the data lines.
  - *I2C Address:* For I2C, ensure the correct address is being used in your code. The default address is typically 0x76 or 0x77, depending on the SDO pin state (GND for 0x76, VCC for 0x77).
  - *Library Installation:* Confirm that the necessary sensor library (e.g., Adafruit BME280) is correctly installed in your IDE.
- **Inaccurate Readings:**
  - *Environmental Factors:* Ensure the sensor is not in direct sunlight, near heat sources, or in a draft, which can affect temperature readings.
  - *Self-Heating:* If the microcontroller or other components generate heat, this can affect the temperature reading of the BME280 if placed too close. Consider isolating the sensor.

- *Calibration:* While the BME280 is factory calibrated, minor offsets can be adjusted in software if precise calibration is required for specific applications.

- **Module Gets Hot:**
  - *Incorrect Voltage:* Immediately disconnect power. Supplying 5V to a 3.3V module can cause overheating and permanent damage.
  - *Short Circuit:* Check for any accidental short circuits on the PCB or connections.

8. Support

For further technical assistance or inquiries regarding the BME280-3.3 module, please refer to the manufacturer's documentation or community forums related to Arduino and sensor modules. Many online resources and communities provide extensive support for integrating and programming these types of sensors.
Note: This manual provides general guidance. Specific implementation details may vary based on your chosen microcontroller, development environment, and libraries.