

[Components](#) → [Sensor Component](#) → BME680 Temperature+Pressure+Humidity+Gas Sensor via BSEC

BME680

Temperature+Pressure+Humidity+Gas Sensor via BSEC

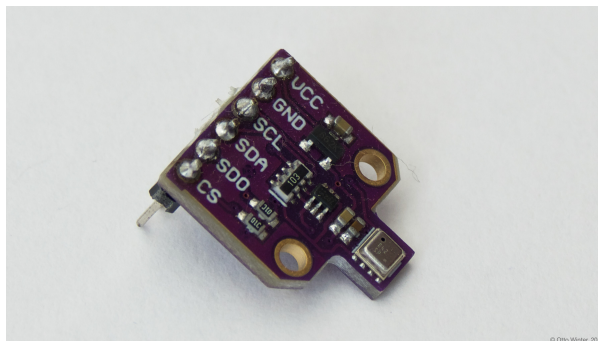
Component/Hub

The `bme680_bsec` sensor platform allows you to use your BME680 ([datasheet](#), [Adafruit](#)) temperature, pressure and humidity and gas sensors with ESPHome via the Bosch Sensortec Environmental Cluster (BSEC) software library. The use of Bosch's proprietary algorithms provide an Index for Air Quality (IAQ) measurement derived from the gas resistance sensor's response to specific Volatile Organic Compounds (VOC). The BSEC software also provides estimated values for CO₂ and Breath Volatile Organic Compounds (b-VOC) using a correlation between VOC and CO₂ in a human's exhaled breath.

Note:

The BSEC library is only available for use after accepting its software license agreement. By enabling this component, you are explicitly agreeing to the terms of the [BSEC license agreement](#). You must not distribute any compiled firmware binaries that include this component.

The [I²C](#) is required to be set up in your configuration for this sensor to work.



BME680 Temperature, Pressure, Humidity & Gas Sensor.

```
# Minimal example configuration with common sensors
i2c:

bme680_bsec:

sensor:
  - platform: bme680_bsec
    temperature:
      name: "BME680 Temperature"
    pressure:
      name: "BME680 Pressure"
    humidity:
      name: "BME680 Humidity"
    iaq:
      name: "BME680 IAQ"
      id: iaq
    co2_equivalent:
      name: "BME680 CO2 Equivalent"
    breath_voc_equivalent:
      name: "BME680 Breath VOC Equivalent"
```

```

text_sensor:
  - platform: bme680_bsec
    iaq_accuracy:
      name: "BME680 IAQ Accuracy"

  - platform: template
    name: "BME680 IAQ Classification"
    icon: "mdi:checkbox-marked-circle-outline"
    lambda: |-
      if ( int(id(iaq).state) <= 50) {
        return {"Excellent"};
      }
      else if (int(id(iaq).state) >= 51 && int(id(iaq).state) <= 100) {
        return {"Good"};
      }
      else if (int(id(iaq).state) >= 101 && int(id(iaq).state) <= 150) {
        return {"Lightly polluted"};
      }
      else if (int(id(iaq).state) >= 151 && int(id(iaq).state) <= 200) {
        return {"Moderately polluted"};
      }
      else if (int(id(iaq).state) >= 201 && int(id(iaq).state) <= 250) {
        return {"Heavily polluted"};
      }
      else if (int(id(iaq).state) >= 251 && int(id(iaq).state) <= 350) {
        return {"Severely polluted"};
      }
      else if (int(id(iaq).state) >= 351) {
        return {"Extremely polluted"};
      }
      else {
        return {"error"};
      }

```

Configuration variables:

- **address** (*Optional*, int): Manually specify the I2C address of the sensor. Defaults to 0x76. Another address can be 0x77.
- **temperature_offset** (*Optional*, float): Temperature offset if device is in enclosure and reads too high. This value is subtracted from the reading (e.g. if the sensor reads 5°C higher than expected, set this to 5) and also corrects the relative humidity readings. Defaults to 0.
- **iaq_mode** (*Optional*, string): IAQ calculation mode. Default is `static` for static applications (e.g. fixed indoor devices). Can be `mobile` for mobile applications (e.g. carry-on devices).
- **sample_rate** (*Optional*, string): Sample rate. Default is `lp` for low power consumption, sampling every 3 seconds. Can be `ulp` for ultra-low power, sampling every 5 minutes. This controls the sampling rate for gas-dependent sensors and will govern the interval at which the sensor heater is operated. By default, this rate will also be used for temperature, pressure, and humidity sensors but these can be overridden on a per-sensor level if required.
- **state_save_interval** (*Optional*, [Time](#)): The minimum interval at which to save calibrated BSEC algorithm state to flash so that calibration doesn't have to start from zero on device restart. Defaults to 6h.
- **id** (*Optional*, [ID](#)): Manually specify the ID used for code generation. Use this ID in the sensor section to refer to the correct BME680 if you have more than one device. This will also be used to refer to the calibrated BSEC algorithm state saved to flash.

Sensor

Configuration variables:

- **bme680_bsec_id** (*Optional*, [ID](#)): Sets the ID of the `bme680_bsec` component to refer to. Useful when working with multiple devices.
- **temperature** (*Optional*): The information for the temperature sensor.

- **name** (**Required**, string): The name for the temperature sensor.
- **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
- **sample_rate** (*Optional*, string): Optional sample rate override for this sensor. Can be `1p` for low power consumption, sampling every 3 seconds or `u1p` for ultra-low power, sampling every 5 minutes.
- All other options from [Sensor](#).
- **pressure** (*Optional*): The information for the pressure sensor.
 - **name** (**Required**, string): The name for the pressure sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - **sample_rate** (*Optional*, string): Optional sample rate override for this sensor. Can be `1p` for low power consumption, sampling every 3 seconds or `u1p` for ultra-low power, sampling every 5 minutes.
 - All other options from [Sensor](#).
- **humidity** (*Optional*): The information for the humidity sensor.
 - **name** (**Required**, string): The name for the humidity sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - **sample_rate** (*Optional*, string): Optional sample rate override for this sensor. Can be `1p` for low power consumption, sampling every 3 seconds or `u1p` for ultra-low power, sampling every 5 minutes.
 - All other options from [Sensor](#).
- **gas_resistance** (*Optional*): The information for the gas sensor.
 - **name** (**Required**, string): The name for the gas resistance sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - All other options from [Sensor](#).
- **iaq** (*Optional*): The information for the IAQ sensor.
 - **name** (**Required**, string): The name for the IAQ sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - All other options from [Sensor](#).
- **iaq_accuracy** (*Optional*): The information for the numeric IAQ accuracy sensor.
 - **name** (**Required**, string): The name for the IAQ accuracy sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - All other options from [Sensor](#).
- **co2_equivalent** (*Optional*): The information for the CO₂ equivalent sensor.
 - **name** (**Required**, string): The name for the CO₂ equivalent sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - All other options from [Sensor](#).
- **breath_voc_equivalent** (*Optional*): The information for the Breath VOC equivalent humidity sensor.
 - **name** (**Required**, string): The name for the Breath VOC equivalent sensor.
 - **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
 - All other options from [Sensor](#).

Text Sensor

Accuracy can be reported in text format.

Configuration variables:

- **bme680_bsec_id** (*Optional*, ID): Sets the ID of the bme680_bsec component to refer to. Useful when working with multiple devices.
- **iaq_accuracy** (*Optional*): The information for the IAQ accuracy sensor. Shows: Stabilizing, Uncertain, Calibrating, Calibrated.

- **name** (**Required**, string): The name for the IAQ accuracy sensor.
- **id** (*Optional*, ID): Set the ID of this sensor for use in lambdas.
- All other options from [TextSensor](#).



BME680 VOC Equivalent

0.5 ppm



BME680 CO2 Equivalent

504.8 ppm



BME680 Gas Resistance

835411 Ω



BME680 Humidity

65.1 %



BME680 Pressure

968.9 hPa



BME680 IAQ

26 IAQ



BME680 IAQ Accuracy

Uncertain



BME680 Temperature

19.1 °C

Advanced configuration

The following configuration shows all the available sensors and optional settings for the component. It also includes an example of filtering to guard against outliers, limit the number of updates sent to home assistant and reduce storage requirements in other systems such as influxdb used to store historical data.

For each sensor, all other options from [Sensor](#) and [TextSensor](#) are also available for filtering, automation and so on.

```
bme680_bsec:
  # id
  # -----
  # Identifier for this component, useful when working with multiple devices.
  # Must be unique, and can be used in the sensor sections to refer to the correct
  # Default: auto-computed
  id: bme680_internal

  # i2c address
  # -----
  # Common values are:
  # - 0x76
  # - 0x77
  # Default: 0x76
  address: 0x76

  # Temperature offset
  # -----
```

```

# Useful if device is in enclosure and reads too high
# For example, if it reads 5C too high, set this to 5
# This also corrects the relative humidity readings
# Default: 0
temperature_offset: 0

# IAQ calculation mode
# -----
# Available options:
# - static (for fixed position devices)
# - mobile (for on a person or other moveable devices)
# Default: static
iaq_mode: static

# Sample rate
# -----
# Available options:
# - lp (low power - samples every 3 seconds)
# - ulp (ultra-low power - samples every 5 minutes)
# Default: lp
sample_rate: ulp

# Interval at which to save BSEC state
# -----
# Default: 6h
state_save_interval: 6h

```

sensor:

```

- platform: bme680_bsec
  # ID of the bme680_bsec component to use for the next sensors.
  # Useful when working with multiple devices
  bme680_bsec_id: bme680_internal

```

```

temperature:
  # Temperature in °C
  name: "BME680 Temperature"
  sample_rate: lp
  filters:
    - median
pressure:
  # Pressure in hPa
  name: "BME680 Pressure"
  sample_rate: lp
  filters:
    - median
humidity:
  # Relative humidity %
  name: "BME680 Humidity"
  sample_rate: lp
  filters:
    - median
gas_resistance:
  # Gas resistance in Ω
  name: "BME680 Gas Resistance"
  filters:
    - median
iaq:
  # Indoor air quality value
  name: "BME680 IAQ"
  filters:
    - median
iaq_accuracy:
  # IAQ accuracy as a numeric value of 0, 1, 2, 3
  name: "BME680 Numeric IAQ Accuracy"
co2_equivalent:
  # CO2 equivalent estimate in ppm
  name: "BME680 CO2 Equivalent"
  filters:

```

```

    - median
  breath_voc_equivalent:
    # Volatile organic compounds equivalent estimate in ppm
    name: "BME680 Breath VOC Equivalent"
    filters:
      - median

text_sensor:
  - platform: bme680_bsec
    iaq_accuracy:
      # IAQ accuracy as a text value of Stabilizing, Uncertain, Calibrating, Calibrated
      name: "BME680 IAQ Accuracy"

```

Multiple sensors

The following configuration shows how to set up multiple BME680 devices. They can be configured to use the same I²C bus or to use different busses, but remember that the BME680 can only be set to operate on I²C address 0x76 or 0x77. There is no limit on the number of BME680 devices that can be connected.

```

# I2C bus for the BME680 devices
i2c:
  - id: "i2cbus_bme"
    sda: GPIO18
    scl: GPIO19
    scan: true

# BME680 devices using BSEC library
bme680_bsec:
  - id: bme680_internal
    i2c_id: "i2cbus_bme"
    address: 0x76
  - id: bme680_external
    i2c_id: "i2cbus_bme"
    address: 0x77

sensor:
  # Sensors for the internal BME680 device
  - platform: bme680_bsec
    bme680_bsec_id: bme680_internal
    temperature:
      name: "IN_Temp"
    pressure:
      name: "IN_Press"
    humidity:
      name: "IN_RH"
    iaq:
      name: "IN_IAQ"
    co2_equivalent:
      name: "IN_CO2eq"
    breath_voc_equivalent:
      name: "IN_VOCeq"

  # Sensors for the external BME680 device
  - platform: bme680_bsec
    bme680_bsec_id: bme680_external
    temperature:
      name: "OUT_Temperatura"
    pressure:
      name: "OUT_Pressione"
    humidity:
      name: "OUT_RH"
    iaq:
      name: "OUT_IAQ"
    co2_equivalent:
      name: "OUT_CO2eq"

```

```
breath_voc_equivalent:
  name: "OUT_VOCeq"

text_sensor:
  # Text sensor for the internal BME680 device
  - platform: bme680_bsec
    bme680_bsec_id: bme680_internal
    iaq_accuracy:
      name: "IN_IAQaccuracy"

  # Text sensor for the external BME680 device
  - platform: bme680_bsec
    bme680_bsec_id: bme680_external
    iaq_accuracy:
      name: "OUT_IAQaccuracy"
```

Index for Air Quality (IAQ) Measurement

The measurements are expressed with an index scale ranging from 0 to 500. The index itself is deduced from tests using ethanol gas, as well as important VOC in the exhaled breath of healthy humans. The VOC values themselves are derived from several publications on breath analysis studies. The BSEC software library defines the levels as follows:

IAQ Index	Air Quality
0 - 50	Excellent
51 - 100	Good
101 - 150	Lightly polluted
151 - 200	Moderately polluted
201 - 250	Heavily polluted
251 - 350	Severely polluted
> 351	Extremely polluted

The selected b-VOC gasses are as follows:

Compound	Molar fraction
Ethane	5 ppm
Isoprene	10 ppm
Ethanol	10 ppm
Acetone	50 ppm
Carbon Monoxide	15 ppm

IAQ Accuracy and Calibration

The BSEC software automatically calibrates automatically in the background to provide consistent IAQ performance. The calibration process considers the recent measurement history so that a value of 50 corresponds to a “typical good” level and a value of 200 to a “typical polluted” level. The IAQ Accuracy sensor will give one of the following values:

- **Stabilizing:** The device has just started, and the sensor is stabilizing (this typically lasts 5 minutes)
- **Uncertain:** The background history of BSEC is uncertain. This typically means the gas sensor data was too stable for BSEC to clearly define its reference.
- **Calibrating:** BSEC found new calibration data and is currently calibrating.
- **Calibrated:** BSEC calibrated successfully.

Every `state_save_interval`, or as soon thereafter as full calibration is reached, the current algorithm state is saved to flash so that the process does not have to start from zero on device restart.

See Also

- [Sensor Filters](#)
- [Absolute Humidity](#)
- [BME680 Temperature+Pressure+Humidity+Gas Sensor](#)
- [API Reference](#)
- [BME680 Datasheet](#)
- [BME680 VOC classification](#)
- [BSEC Arduino Library](#) by [Bosch Sensortec](#)
- [Bosch Sensortec Community](#)
- [Edit this page on GitHub](#)