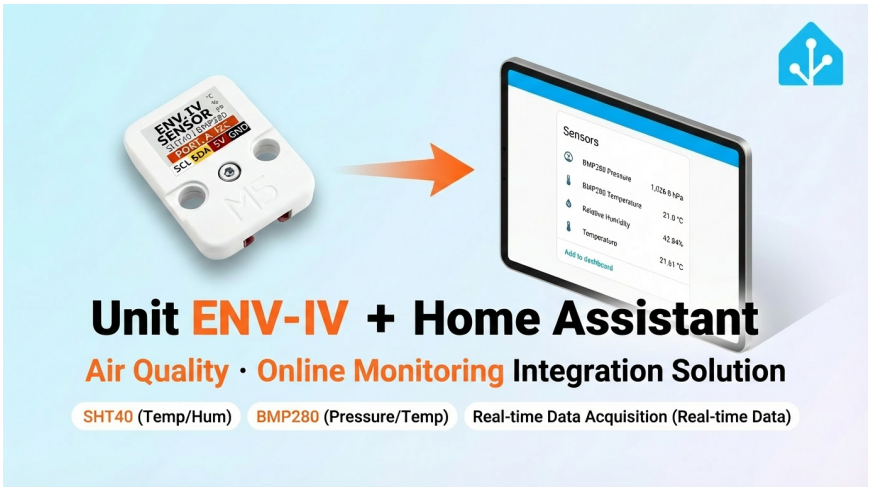
Unit ENV-IV Sensor Home Assistant Integration

This chapter introduces the configuration methods and practical steps for integrating the Unit ENV-IV sensor platform into Home Assistant.



Preparation

1. Confirm that the sensors used support ESPHome integration:
    1. BMP280 Temperature and Pressure Sensor: BMP280 datasheet
    2. SHT40 Temperature and Humidity Sensor: SHT40 datasheet
2. Refer to the official ESPHome documentation for the latest configuration instructions:
    1. BMP280 Temperature and Pressure Sensor ESPHome Configuration
    2. SHT4X Temperature and Humidity Sensor ESPHome Configuration

Note

Unit ENV-IV is a standalone sensor platform and requires an additional main controller (such as Atom series, Stamp series, Stick series, Core/Basic series, etc.) to be integrated into Home Assistant.

Learn

1. The I²C component needs to be enabled in the ESPHome configuration:

```
# Example configuration entry for ESP32
i2c:
  sda: GPIOXX
  scl: GPIOXX
  scan: true
```

The GPIO pins here will vary depending on the main controller used. For example, using Atom Lite as the main controller:

```yaml
# I2C Bus on Grove Port (HY2.0-4P)
i2c:
  sda: GPIO26
  scl: GPIO32
```

Unit ENV-IV configuration example:

yaml

```yaml
sensor:
  - platform: sht4x
    temperature:
      id: sht40_temp
      name: "Temperature"
    humidity:
      id: sht40_humi
      name: "Relative Humidity"
    address: 0x44

  - platform: bmp280_i2c
    temperature:
      name: "BMP280 Temperature"
      id: bmp280_temp
      oversampling: 16x
    pressure:
      name: "BMP280 Pressure"
      id: bmp_pressure
    address: 0x76
```

Similarly, you can calculate values such as altitude, absolute humidity, and dew point based on empirical formulas (calculation results are for reference only):

- Altitude Calculation
- Absolute Humidity Calculation
- Dew Point Calculation

yaml

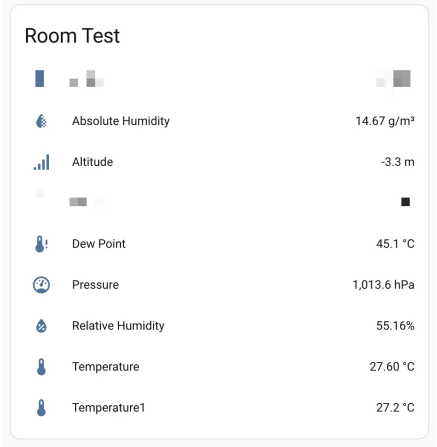
```yaml
sensor:
  ...
  # add the following under the previous sensor declarations
  - platform: template
    name: "Altitude"
    lambda: |-
      const float STANDARD_SEA_LEVEL_PRESSURE = 1013.25; //in hPa, see note
      return ((id(bmp280_temp).state + 273.15) / 0.0065) *
        (powf((STANDARD_SEA_LEVEL_PRESSURE / id(bmp_pressure).state), 0.190234) - 1); // in meter
    update_interval: 15s
    icon: 'mdi:signal'
    unit_of_measurement: 'm'

  - platform: absolute_humidity
    name: "Absolute Humidity"
    temperature: sht40_temp
    humidity: sht40_humi

  - platform: template
    name: "Dew Point"
    lambda: |-
      return (243.5*(log(id(sht40_humi).state/100)+((17.67*id(sht40_humi).state)/
      (243.5+id(sht40_temp).state)))/(17.67-log(id(sht40_humi).state/100)-
      ((17.67*id(sht40_temp).state)/(243.5+id(sht40_temp).state))));
    unit_of_measurement: °C
    icon: 'mdi:thermometer-alert'
```

Note

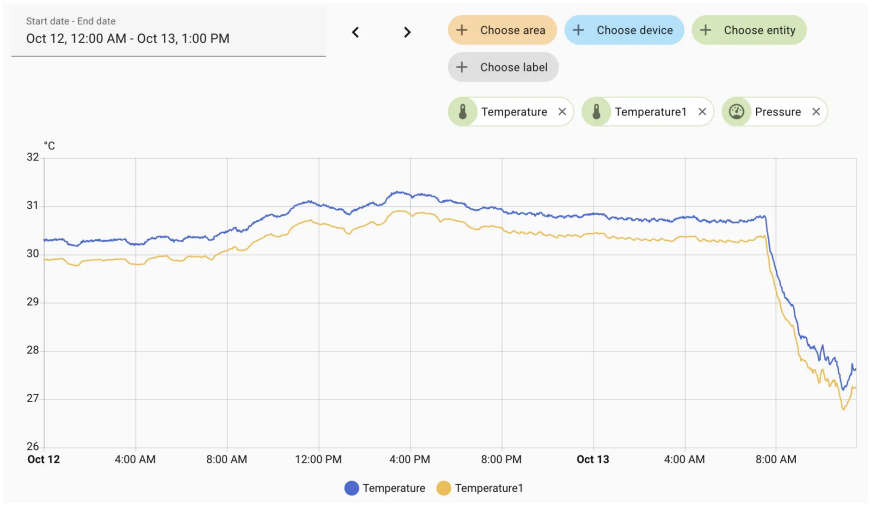
To accurately calculate altitude using the BMP280 pressure sensor, you must first obtain the sea level standard pressure for your location and the current day. This can be achieved by replacing the global constant `STANDARD_SEA_LEVEL_PRESSURE`, for example, by fetching the value in real-time from the internet or via the MQTT protocol from a fixed sensor.

Getting Started

1. After adding it to the Dashboard, you can view the sensor data in Home Assistant.



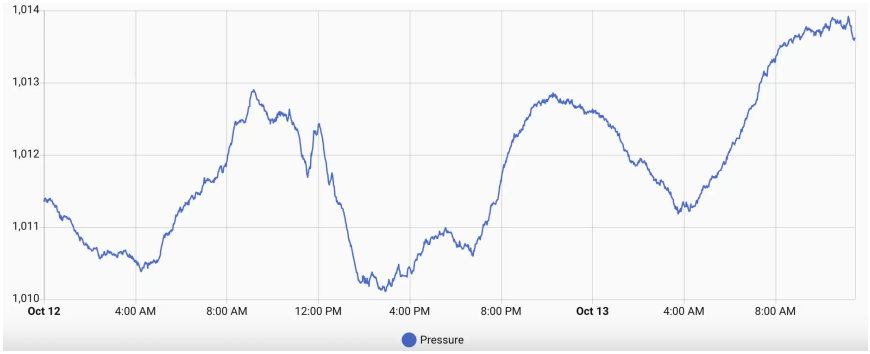
2. Monitor temperature change curves.



Note

Since the BMP280 also has a temperature sensor, there will be two temperature curves here.

3. Monitor atmospheric pressure changes.



Video