

# Environment click

PID: MIKROE-2467

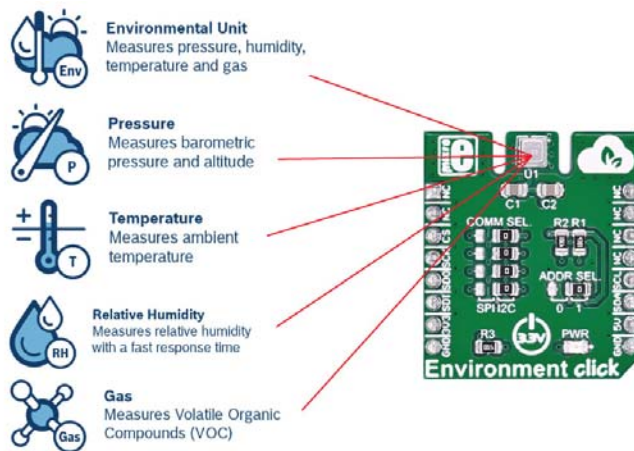


**Environment click** measures temperature, relative humidity, pressure and VOC (Volatile Organic compounds gases). The click carries the BME680 environmental sensor from Bosch. Environment click is designed to run on a 3.3V power supply. It communicates with the target microcontroller over SPI or I2C interface.

You can use it to test your indoor air quality, to control HVAC (heating, ventilation, and air conditioning) systems, in a weather station, sports applications and more.

## BME680 sensor features

The BME680 is as combined digital gas, humidity, pressure, and temperature sensor based on proven sensing principles.



The **humidity sensor** provides an extremely fast response time for fast context awareness applications and high overall accuracy over a wide temperature range. The **pressure sensor** is an absolute barometric pressure sensor with extremely high accuracy and resolution.

The integrated **temperature sensor** has been optimized for lowest noise and highest resolution. Its output is used for temperature compensation of the pressure and humidity sensors and can also be used for estimation of the ambient temperature.


The **gas sensor** within the BME680 can detect a broad range of gases to measure indoor air quality for personal well being. Gases that can be detected by the BME680 include Volatile Organic Compounds (VOC) from paints (such as formaldehyde), lacquers, paint strippers, cleaning supplies, furnishings, office equipment, glues, adhesives, and alcohol.

### Specifications

Type	Enviromental combo
Applications	Indoor air quality measurement, personalized weather station, home automation control, measuring ambient temperature, etc.
On-board modules	BME680 integrated environmental sensor
Key Features	A digital 4-in-1 sensor with gas, humidity, pressure and temperature measurement
Interface	I2C,SPI
Input Voltage	3.3V
Click board size	S (28.6 x 25.4 mm)

## Pinout diagram

This table shows how the pinout on **Environment click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
	NC	1	AN	PWM	16	NC	
	NC	2	RST	INT	15	NC	
Chip select	<b>CS</b>	3	CS	TX	14	NC	
SPI clock	<b>SCK</b>	4	SCK	RX	13	NC	
Master in slave out	<b>MISO</b>	5	MISO	SCL	12	<b>SCL</b>	I2C clock
Master out slave in	<b>MOSI</b>	6	MOSI	SDA	11	<b>SDA</b>	I2C data
Power supply	<b>+3.3V</b>	7	3.3V	5V	10	NC	
Ground	<b>GND</b>	8	GND	GND	9	<b>GND</b>	Ground

## Jumpers and settings

Designator	Name	Default Position	Default Option	Description
JP1	COMM. SEL.	Right	I2C	Selecting communication with MCU between SPI and I2C
JP2	ADDR. SEL.	Right	1	I2C address selection
JP3	COMM. SEL.	Right	I2C	Selecting communication with MCU between SPI and I2C
JP4	COMM. SEL.	Right	I2C	Selecting communication with MCU between SPI and I2C
JP5	COMM. SEL.	Right	I2C	Selecting communication with MCU between SPI and I2C

## Programming

Code examples for Environment click, written for MikroElektronika hardware and compilers are available on Libstock.

### *VOC library*

The Libstock page for the Environment click offers a library for temperature, humidity, and pressure readings. To get the library for air quality (VOC) measurements, visit the **Bosch website** and download the VOC library file for mikroC (libalgobsec.emcl).

The run-in phase for VOC measuring is approximately 5 minutes.

**Note:** Once you go to the Bosch website, you will need to read and agree with the terms of the software license agreement, before you can download the VOC library.

For more information about the Environment click library, see our Libstock page.

### *Code snippet*

The following code reads the procedure on the I2C bus (PIC32MX).

```
01 int8_t bus_read(uint8_t dev_addr, uint8_t reg_addr, uint8_t *reg_data_ptr,
uint8_t data_len)
02 {
03     uint8_t *ptr = reg_data_ptr;
04
05     I2C2_Start();
06     I2C2_Write( 0x77 << 1 );
07     I2C2_Write( reg_addr );
08     I2C2_Start();
09     I2C2_Write( ( 0x77 << 1 ) | 0x01 );
10     while( data_len-- > 1 )
11         *( ptr++ ) = I2C2_Read( _I2C_ACK );
12     *ptr = I2C2_Read( _I2C_NACK );
13     I2C2_Stop();
14     return 0;
15 }
```