RRH47000

NDIR CO_2 Sensor

Description

The RRH47000 sensor module uses non-dispersive infrared (NDIR) technology to accurately measure CO_2 in a compact package size while ensuring competitive pricing.

An integrated relative humidity and temperature sensor enables calibration and compensation of the NDIR sensor for precise operation in different environments. The integrated sensor signal conditioning and an MCU allow algorithms to use the correlation between all built-in sensing elements to improve the accuracy of each measurement.

CO_2 level is a key parameter for indoor air quality. The sensor module enables active, real-time monitoring of CO_2 along with humidity and temperature, allowing automated demand-controlled ventilation and environment control for better air quality, energy efficiency, and wellbeing of occupants.

RRH47000 is configurable with selectable UART or I^2C interface, with a 40.8 × 19.7 × 9.1 mm package.

Features

- Best-in-class NDIR CO_2 sensor technology
- Integrated temperature and humidity sensor
- CO_2 measurement range: 0 ppm to 5000 ppm
- CO_2 Accuracy: typical ±75ppm
- Current consumption: < 50mA at 1s sample time
- I^2C and UART interface
- Package: 40.8 × 19.7 × 9.1 mm pin type
- Long term stability and long lifetime >15 years
- Supply Voltage: 4.5V to 5.5V
- Compliant with RESET®, California Title 24, UBA and WELL Building Standard™

Applications

- HVAC / industrial automation / building control
- Air quality monitors
- Home appliances / air purifiers
- IoT devices
- Agriculture / greenhouses

Figure 1. Sensor Module

Contents

Figures

Tables

1. Overview

1.1 Functional Overview

This CO_2 sensor uses a non-dispersive infrared (NDIR) detection method. The sensor consists of four main components:

1. **Infrared Source:** Emits a beam of infrared light.
2. **Sample Chamber**: A chamber where the gas sample is introduced.
3. **Optical Filter:** An IR filter positioned in front of the detector, allowing only specific wavelengths of infrared light to pass through. This ensures that only the light absorbed by CO_2 reaches the detector.
4. **Infrared Detector:** Measures the intensity of the infrared light that passes through the filter and sample chamber

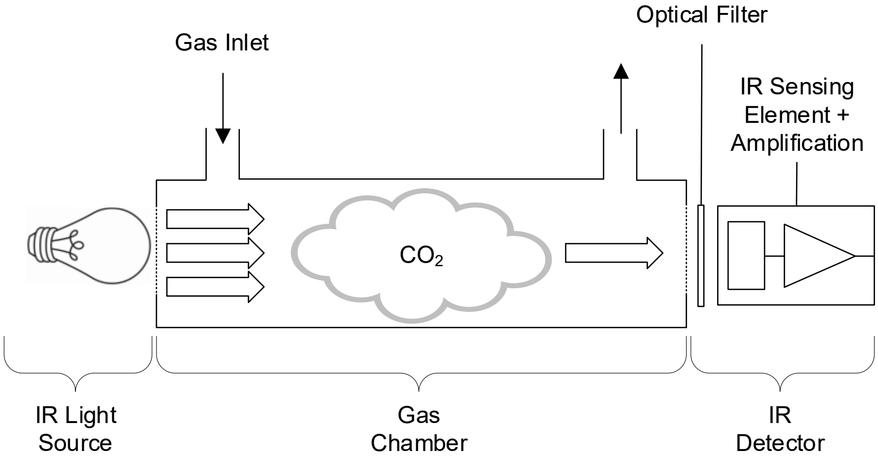


Figure 2. Functional Overview of RRH47000

Inside the sensor, an infrared light source emits a specific wavelength, in this case 4.26µm, that is absorbed by CO_2 molecules. A filter blocks all other wavelengths of light. The remaining light passes through the gas chamber and the sensor measures the intensity of light that reaches it. As CO_2 concentration increases, more light is absorbed, and the sensor output signal changes accordingly. This relationship between light absorption and gas concentration is described by the Lambert-Beer Law.

The Lambert-Beer Law, also known as Beer's Law, quantifies the relationship between the concentration of a gas and the absorption of light. The law is mathematically expressed as:

$$A = \varepsilon \cdot c \cdot l$$

where:

- (A) is the absorbance (no units)
- (ε) is the molar absorptivity or extinction coefficient ($L \cdot mol^{-1} \cdot cm^{-1}$)
- (c) is the concentration of the absorbing species in the sample ($mol \cdot L^{-1}$)
- (l) is the path length through which the light passes in the sample (cm)

This law illustrates that absorbance is directly proportional to the concentration of the absorbing species and the path length the light travels through the sample.

1.2 Block Diagram

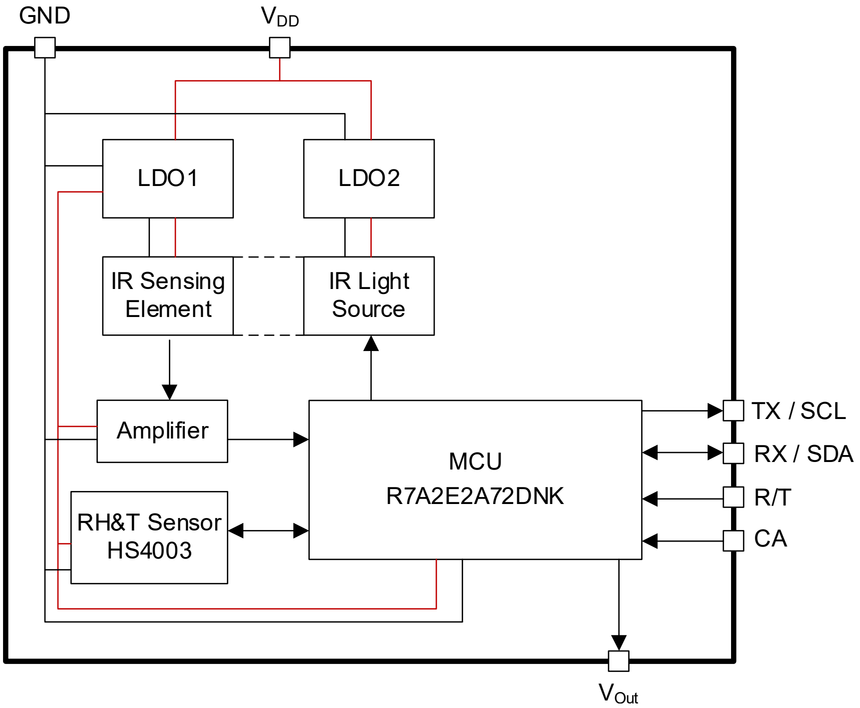


Figure 3. Functional Block Diagram of RRH47000

2. Pin Information

2.1 Pin Assignments

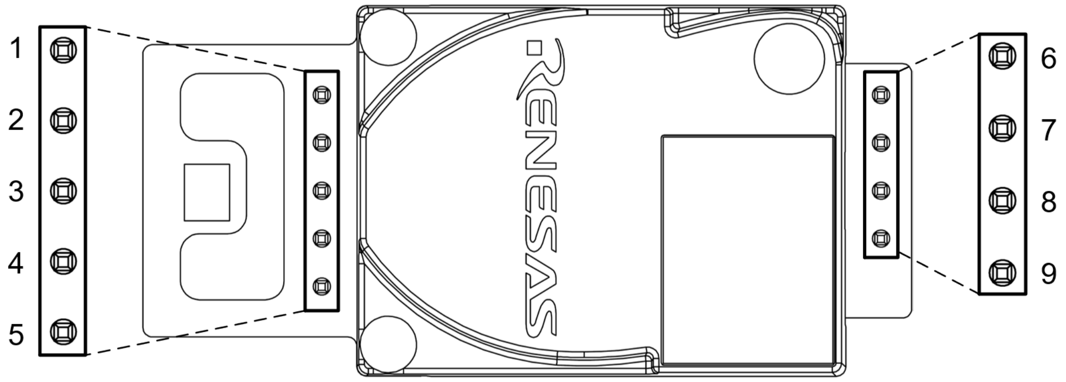


Figure 4. Pin Assignments – Top View

2.2 Pin Descriptions

Table 1. Pin Descriptions

Pin Number	Pin Name	Description
1	V_{Out}	Power output (+3.3V/100mA).
2	RX	UART-RX (Receiving).
	SDA	I²C data (slave address 0x31).
3	TX	UART-TX (Sending).
	SCL	I²C clock.
4	R/T	Output mode selection, high level or floating for UART, low level for I^2C.
5	CA	Manual Correction.
6	V_{DD}	Power supply input (+4.5V to +5.5V).
7	GND	Ground.
8	DNC	Do not connect. Leave floating.
9	DNC	Do not connect. Leave floating.

3. Specifications

3.1 Absolute Maximum Ratings

Caution: Do not operate at or near the maximum ratings listed for extended periods of time. Exposure to such conditions can adversely impact product reliability and result in failures not covered by warranty.

Table 2. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Unit
Supply Voltage	3.7	6	V
Storage Temperature Range	-40	85	°C

3.2 Recommended Operating Conditions

Table 3. Electrical Specifications

Parameter	Condition	Minimum	Typical	Maximum	Unit
Operating Supply Voltage	-	4.5	5	5.5	V
Operating/Storage Humidity Range	Non-condensing	0	-	95	%RH
Operating Temperature Range	-	-10	-	50	°C
Storage Temperature Range	-	-30	-	70	°C
Sleep Current	-	-	5	-	mA
Average Current	1s sample time	-	10	-	mA
Peak Current [1]	1s sample time	-	-	220	mA
Sampling Interval	-	-	1	-	s
Preheating time	-	30	-	-	s
Startup Time for Communication	UART	0.5	-	-	s
	I²C	3	-	-	s
Signal Output Voltage Level	UART/I²C	-	3.3	-	V

1. Describes the peak current when emitter is switched on.

4. Sensor Module Specifications

4.1 CO_2 Sensor Specifications

This sensor module is fully compliant with several well-established standards, ensuring its reliability and accuracy in various applications. It meets the requirements set by UBA[1] (Umweltbundesamt), WELL™ Building Standard[2], RESET® Air Standard[3], and California Title 24[4]. This allows the sensors to be used in residential, commercial, and industrial environments.

Table 4. CO_2 Sensor Specifications

Parameter	Conditions	Minimum	Typical	Maximum	Unit
CO_2 Measurement range, default [1]	Fast ABC turned on	400	-	5000	ppm
CO_2 Measurement range, extended	Fast ABC turned off	0	-	5000	ppm
Operating Temperature Range	-	-10	-	50	°C
Accuracy [2] [3]	0 to 5000 ppm, -10°C~50°C and 0-85%RH	-	±75	± (30+ 3% of reading)	ppm
Accuracy Drift	Without regular ABC	-	3	-	ppm/day
Life span	-	15	-	-	years
Response time	$T_{90\%}$	-	80	-	s

1. Sensor is designed to measure in the range 0~5000ppm; nevertheless, exposure to concentrations below 400ppm may result in incorrect operation of ABC algorithm and should be avoided for model with auto baseline correction (ABC) ON.
2. In normal IAQ applications, accuracy is defined after minimum three (3) ABC (automatic baseline correction) periods of continuous operation with ABC on.
3. Specification is referenced to certified calibration mixtures. Uncertainty of calibration gas mixtures (vendor dependent, typical ±2%) have to be added to the specified accuracy for absolute measurement.

4.2 Humidity and Temperature Sensor Specifications

Table 5. Humidity and Temperature Sensor Specifications, T_A = +25°C, V_{DD} = 1.71V to 3.6V

Parameter	Condition	Minimum	Typical	Maximum	Unit
Humidity Sensor					
Range	-	0	-	100	%RH
Accuracy [1]	20% to 80%RH	-	±3.5	-	%RH
Resolution	14-bit	-	0.04	-	%RH
Hysteresis	-	-	-	±1.0	%RH

[1] Umweltbundesamt, Gesundheitliche Bewertung von Kohlendioxid in der Innenraumluft, (Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz, 2008).
[2] WELL v2 pilot, Q4 2022.
[3] RESET Air Standard v2.0 Grade B, 2018.
[4] California Building Energy Efficiency Standards for Residential and Nonresidential Buildings, 2022.

Parameter	Condition	Minimum	Typical	Maximum	Unit
Non-linearity from Response Curve	20% to 80% RH	-	±0.15	±0.25	%RH
Long-Term Stability	-	-	±0.1	-	%RH/Yr
Response Time Constant [2] ($τ_H$)	20% to 80% RH Still Air	3.0	4.0	6.0	s
Temperature Sensor					
Range	-	-40	-	125	°C
Accuracy	-	-	±0.5	-	°C
Resolution	14-bit	-	0.01	-	°C
Response Time Constant [3] ($τ_T$)	-	-	>2.0	-	s
Long-Term Stability	-	-	-	0.03	°C/Yr

1. Monotonic increases from 20 to 80% RH after sensor has been stabilized at 50% RH.
2. Initial value to 63% of total variation. Response time depends on system airflow.
3. Initial value to 63% of total variation. Response time depends on system thermal mass and air flow.

5. Sensor Placement

To ensure a proper airflow to the CO_2 sensing path, a minimum distance of 1.5mm between the area of the waterproof filter and the other components must be considered. Not keeping the minimum distance will result in a slower sensor response time to environmental changes.

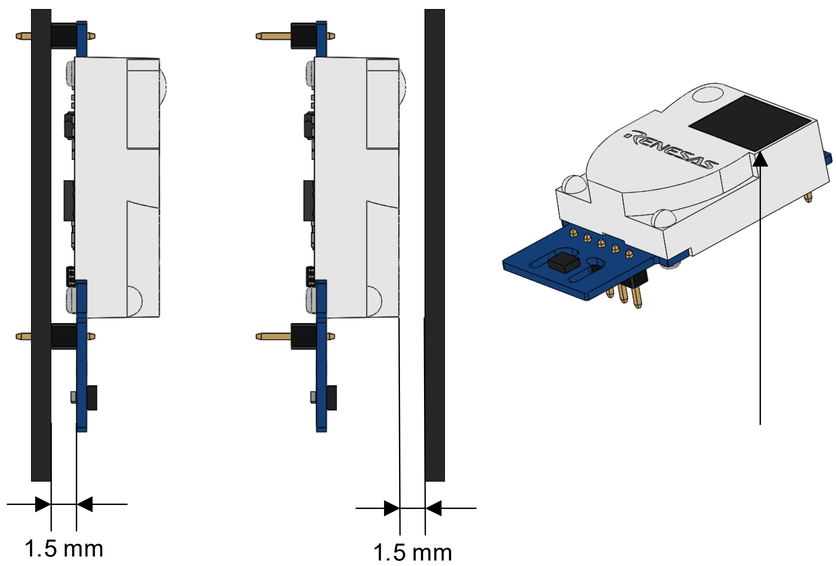


Figure 5. Sensor Placement Recommendation

6. Sensor Correction

6.1 Automatic Baseline Correction (ABC)

To ensure optimal performance, handle the CO_2 sensor with care during transport and installation. Rough handling, sensor aging, and drastic environmental changes may cause a reduction in accuracy.

The sensor incorporates an auto-baseline correction (ABC) feature to automatically compensate for drift. This feature operates as follows:

- **Initial Baseline Correction**: After the sensor is powered on, it continuously monitors the CO_2 levels for 24 hours. The lowest measured concentration during this period is established as the new baseline.
- **Periodic Baseline Correction**: Following the initial 24-hour period, the sensor automatically performs baseline correction every 7 days. The time interval for the periodic baseline correction can be adjusted.

For the ABC feature to function correctly, the sensor must be exposed to fresh air (approximately 400ppm CO_2) during both the initial 24-hous period and subsequent 7-day correction cycles. Ensure the sensor's locations allows for such exposure.

6.2 Manual Baseline Correction

To quickly restore sensor accuracy after installation or some special event, perform a manual baseline correction. To do this, place the sensor in an environment with a stable CO_2 concentration of approximately fresh, clean air (400 ppm). Connect the CA pin of the sensor to GND for at least 3 seconds. The total duration of the manual baseline correction is determined by the distance in ppm between the current ppm level and the set baseline level. The sensor requires 1 additional second for every 25 ppm.

Example calculation for a 400ppm baseline and 500ppm current CO_2 reading:

3s (trigger) + 1s (calc/storage) + 4s ((500ppm – 400ppm) / 25ppm) = 8s (total)

Manual correction can also be performed by sending a command to the sensor. Refer to the communication protocol documentation (7.1.4.7and 7.2.5.4) for details on how to send this command.

7. Functional Description

7.1 UART Interface

7.1.1 Typical Application Circuit for UART Interface

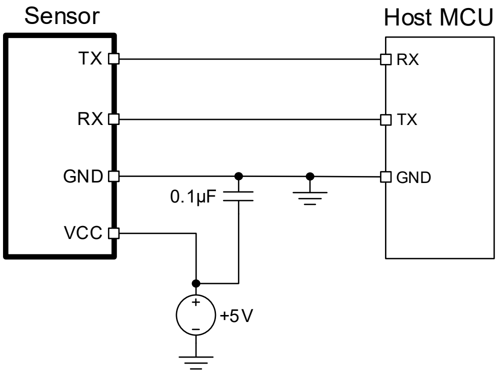


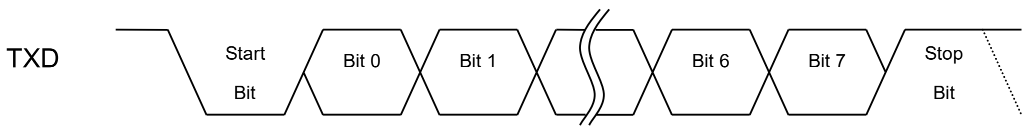
Figure 6. Application Circuit for UART Communication

7.1.2 UART Transmission Characteristics

Each data frame consists of a start bit, data, and stop bit. Data is sent asynchronously within each data frame. The data is all hexadecimal data (for example: “46” for decimal [70]). After power-on, the sensor needs at least 0.5s to start communication via UART.

[x x] is for single-byte data (unsigned, 0-255); for double data, high byte is in front of low byte.

Parameter	Value
Baud Rate	9600 bps
Data Bite Size	8-bit
Stop Bit Size	1-bit
Parity	No
Flow Control	No



7.1.3 Basic Data Transmission Format UART

Command Request

Start	Length	Command	Data	…	Data	Checksum
0x11	0xXX	0xXX	0xXX		0xXX	0xXX

Command Response

Start	Length	Command	Data	…	Data	Checksum
0x16	0xXX	0xXX	0xXX		0xXX	0xXX

Figure 7. Data Transmission Format UART Communication

Table 6. UART Protocol Format Description

Protocol Format	Description
Start	Master to slave is fixed to [0x11], slave to master is fixed to [0x16].
Length	Length of frame bytes = data length +1 (including Command + Data).
Command	See Table 7.
Data	Data of writing or reading, length is not fixed.
Check Sum	See Checksum Calculation.

7.1.4 UART Command Description

7.1.4.1 Overview of UART Commands

Table 7. Overview of Available UART Commands

Code	Action	Request Bytes	Response Bytes
0x0B	Set Measurement Range.	5	4
	Get Measurement Range.	4	5
0x31	Get Measurement Results.	4	12
0x10	Set ABC Parameters.	10	4
0x0F	Get ABC Parameters.	4	10
0x03	Set CO_2 Baseline Manually.	6	4
0x1F	Get Serial Number.	4	20
0x1E	Get Firmware Version Number.	4	15

7.1.4.2 Set Measurement Range

Read the current measurement range setting.

	Value	Bytes [Hex]	Description
Command	0x0B	-	-
Request bytes	5	[0x11] [0x02] [0x0B] [D1] [CS]	-
Response bytes	4	[0x16] [0x02] [0x0B] [DE]	[D1]: 0 = 0 – 5000 ppm 1 = 400 – 5000 ppm

7.1.4.3 Get Measurement Range

Read the current measurement range setting.

	Value	Bytes [Hex]	Description
Command	0x0B	-	-
Request bytes	4	[0x11] [0x01] [0x0B] [0xE3]	-
Response bytes	5	[0x16] [0x02] [0x0B] [D1] [CS]	[D1]: 0 = 0 – 5000 ppm 1 = 400 – 5000 ppm

7.1.4.4 Get Measurement Results

Read all measurement results of the RRH47000. Returns CO_2 (ppm), temperature (°C), and humidity (%RH) bytes. For more information on how to convert these bytes into the respective measure, see section 7.3.

	Value	Bytes [Hex]	Description
Command	0x31	-	-
Request bytes	4	[0x11] [0x01] [0x31] [0xBD]	-
Response bytes	12	[0x16] [0x09] [0x31] [D1] … [D8] [CS]	[D1] and [D2]: CO_2 Results [D3] and [D4]: Reserved [D5] and [D6]: Temperature Results [D7] and [D8]: Humidity Results

7.1.4.5 Set ABC parameters

This function is used to set the automatic baseline correction (ABC) parameters. The ABC function is crucial for maintaining the accuracy and reliability of the sensor's long-term CO_2 measurements. By periodically recalibrating the sensor to a baseline CO_2 concentration, typically representing fresh outdoor air, the ABC function compensates for sensor drift and ensures consistent performance.

A fast automatic baseline correction (fast ABC) occurs when the sensor undergoes a correction process outside the set correction period (default 24 h/7 days). This correction is typically triggered during periods of exceptionally low CO_2 concentrations that are unrealistic for fresh air (below 400ppm). The fast ABC ensures that the sensor quickly adapts to new baseline CO_2 concentrations, thereby maintaining its accuracy and reliability in dynamic conditions.

The CO_2 baseline value is used to set the lowest expected CO_2 concentrations to what the ABC references. This feature can be utilized to ensure accuracy in environments where fresh air is higher or lower than 400ppm, ensuring accuracy in different environments.

	Value	Bytes [Hex]	Description
Command	0x10	-	-
Request bytes	10	[0x11] [0x07] [0x10] [D1] … [D6] [CS]	[D1]: Reserved, default 0x64 [D2]: Enable/disable auto correction (0: enable all (default), 1: enable only ABC, 2: enable only fast ABC, 3: disable all) [D3]: Correction cycle period (1-30 days, default is 7 days) [D4]: High byte CO_2 baseline value, default 0x01 (400) [D5]: Low byte CO_2 baseline value , default 0x90 (400) [D6]: Reserved, default 0x64
Response bytes	4	[0x16] [0x01] [0x10] [0xD9]	-

7.1.4.5.1 Disable all ABC

The automatic baseline correction functions are enabled by default. To disable this function, set [2] = 3.

Table 8. Example Request to Disable “Automatic Baseline Correction”

	Value	Bytes [Hex]
Request bytes	10	[0x11] [0x07] [0x10] [0x64] [0x03] [0x07] [0x01] [0x90] [0x64] [0x76]
Response bytes	4	[0x16] [0x01] [0x10] [0xD9]

7.1.4.5.2 Enable ABC and Set Correction Cycle

To turn the automatic baseline correction (ABC) function back on after disabling it, set [D2] to 0. Additionally, set the correction cycle period to 7 days.

Table 9. Example Request to Open ABC and Set Correction Cycle

	Value	Bytes [Hex]
Request bytes	10	[0x11] [0x07] [0x10] [0x64] [0x00] [0x07] [0x01] [0x90] [0x64] [0x78]
Response bytes	4	[0x16] [0x01] [0x10] [0xD9]

7.1.4.5.3 Change the Correction Cycle Period

The correction cycle is 7 days by default. For example, if want to change the correction cycle to 10 days, the user should set the [D3] = 0A.

Table 10. Example Request to Change Correction Cycle

	Value	Bytes [Hex]
Request bytes	10	[0x11] [0x07] [0x10] [0x64] [0x00] [0x0A] [0x01] [0x90] [0x64] [0x75]
Response bytes	4	[0x16] [0x01] [0x10] [0xD9]

7.1.4.6 Get ABC Parameters

Read the current ABC parameter set.

	Value	Bytes [Hex]	Description
Command	0x0F	-	-
Request bytes	4	[0x11] [0x01] [0x0F] [0xDF]	-
Response bytes	10	[0x16] [0x07] [0x0F] [D1] … [D6] [CS]	[D1]: Reserved, default 0x64 [D2]: Enabled/disabled auto correction (0: enabled all (default), 1: enabled only ABC, 2: enabled only fast ABC, 3: disabled all) [D3]: Correction cycle period (1-30 days optional, default is 7 days) [D4]: High byte CO_2 baseline value, default 0x01 (400) [D5]: Low byte CO_2 baseline value , [D6]: Reserved, default 0x64

7.1.4.7 Set CO_2 Baseline Manually

Correct current CO_2 reading to a target concentration. Use the following equation to calculate D1 and D2. The correction range is 400 to 1500 ppm. Convert the decimal values to hexadecimal before writing them to the sensor.

$$CO_2 = [D1] \cdot 256 + [D2]$$

Before correction, make sure that the CO_2 concentration in the current environment is in the correction target range. Let the RRH47000 stabilize in the environment for at least 2 minutes before beginning the correction.

	Value	Bytes [Hex]	Description
Command	0x03	-	-
Request bytes	6	[0x11] [0x03] [0x03] [D1] [D2] [CS]	[D1] and [D2]: CO_2 concentration in ppm
Response bytes	4	[0x16] [0x01] [0x03] [0xE6]	-

Example:

For a CO_2 concentration of 600ppm, the bytes D1 and D2 need to represent this value in hexadecimal form. The value 600 in decimal is 0x258 in hexadecimal. Therefore, D1 would be 0x02 and D2 would be 0x58.

7.1.4.8 Get Serial Number

Returns the serial number of the sensor. The serial number received needs to be converted to ASCII.

	Value	Bytes [Hex]	Description
Command	0x1F	-	-
Request bytes	4	[0x11] [0x01] [0x1F] [0xCF]	-
Response bytes	20	[0x16] [0x11] [0x1F] [D1] … [D16] [CS]	[D1] to [D16]: Serial number as hexadecimal

Example:

Serial number in hex: 35 31 39 32 34 31 30 44 30 31 31 31 30 38 36 39

Corresponding serial number in ASCII: 5192410D01110869

7.1.4.9 Get Firmware Version Number

Returns the firmware version of the RRH47000. The received firmware version number needs to be converted to ASCII.

	Value	Bytes [Hex]	Description
Command	0x1E	-	-
Request bytes	4	[0x11] [0x01] [0x1E] [0xD0]	-
Response bytes	15	[0x16] [0x0C] [0x1E] [D1] … [D11] [CS]	[D1] to [D11]: Firmware version number as hexadecimal

Example:

Serial number in hex: 56 35 2E 30 35 5F 36 42 52 30 32

Corresponding serial number in ASCII: V5.05_6BR02

Note: ASCII code 20 represents a blank space.

7.2 I²C Interface

This protocol follows the standard I²C timing sequence, with a clock frequency ranging from 10kHz to 100kHz. Communication utilizes the big-endian format, where the most significant bit is transmitted first. After power-on, the sensor requires a minimum of 3 seconds to initiate communication via I²C.

7.2.1 Typical Application Circuit for I²C Interface

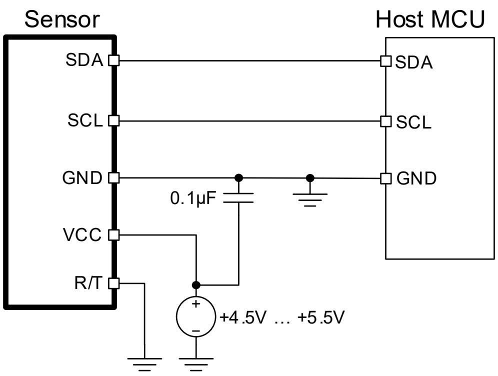


Figure 8. Application Circuit for I²C Communication

7.2.2 I²C Transmission Characteristics

Table 11. I²C Transmission Characteristics

Slave Address	0x31
Clock Frequency	Typical 100kHz
General Call Address	Unsupported
Logic Voltage Level High	> 2.64V (0.8 * 3.3V)
Logic Voltage Level Low	< 0.66V (0.2 * 3.3V)

7.2.3 I²C Features and Timing

Table 12. I²C Timing Parameters

Parameter	Symbol	Minimum	Typical	Maximum	Unit
SCL clock frequency	f_{SCL}	10	-	100	kHz
Hold time START condition	$t_{HD.STA}$	-	0.6	-	µs
Setup time START condition	$t_{SU.STA}$	-	0.6	-	µs
Data setup time	$t_{SU.DAT}$	-	250	-	ns
Data hold time	$t_{HD.DAT}$	-	0	-	µs
Setup time for STOP condition	$t_{SU.STO}$	-	4	-	µs

Note: SCL clock frequency is generated by the master device with the range 10kHz to 100kHz.

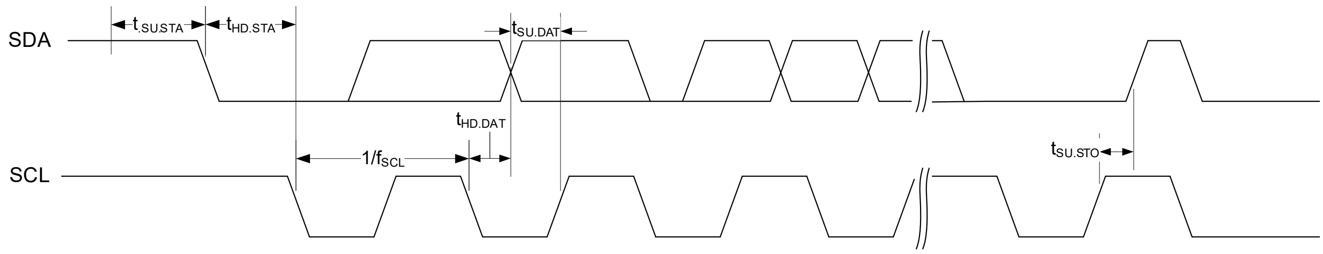


Figure 9. I²C Timing Diagram

If an I/O port is used to simulate an I²C master device, it is suggested to reserve a period before and after ACK signal (such as 100µs), after sending every byte (8 bit) to leave enough time for the MCU to process the data. Within requirements of speed, it is recommended to lower the reading speed as much as possible.

Using a delay of 300ms between request and response is recommended. Always verify the checksum to ensure data integrity.

7.2.4 Basic Data Transmission Format I²C

Figure 10. Data Transmission Format I²C Communication

7.2.5 I²C Command Description

7.2.5.1 Overview of I²C Commands

The slave address is 0x31; the data command of the slave device is listed in the following table.

Table 13. Overview of Available I²C Commands

Code	Action	Request bytes	Response Bytes
0x31	Get Measurement Results.	1	10
0x10	Set ABC Parameters.	7	8
0x03	Set CO_2 Baseline Manually.	3	4
0x1F	Get Serial Number.	1	18
0x1E	Get Firmware Version Number.	1	13

Every response includes a request command as a first byte and the checksum as last byte.

7.2.5.2 Get Measurement Results

Returns the measurement results for CO_2, temperature, and relative humidity. For more information on how to calculate the output values, see section 7.3.

	Value	Bytes [Hex]	Description
Command	0x31	-	-
Request bytes	1	[0x31]	-
Response bytes	10	[0x31] [D1] … [D8] [CS]	[D1] and [D2]: CO_2 Results [D3] and [D4]: Reserved [D5] and [D6]: Temperature Results [D7] and [D8]: Humidity Results

7.2.5.3 Set ABC Parameters

This function is used to set the automatic baseline correction (ABC) parameters. The ABC function is crucial for maintaining the accuracy and reliability of the sensor's long-term CO_2 measurements. By periodically recalibrating the sensor to a baseline CO_2 concentration, typically representing fresh outdoor air, the ABC function compensates for sensor drift and ensures consistent performance.

A fast automatic baseline correction (fast ABC) occurs when the sensor undergoes a correction process outside the set correction period (default 24 h/7 days). This correction is typically triggered during periods of exceptionally low CO_2 concentrations that are unrealistic for fresh air (below 400ppm). The fast ABC ensures that the sensor quickly adapts to new baseline CO_2 concentrations, thereby maintaining its accuracy and reliability in dynamic conditions.

The CO_2 baseline value is used to set the lowest expected CO_2 concentrations to what the ABC references. This feature can be utilized to ensure accuracy in environments where fresh air is higher or lower than 400ppm, ensuring accuracy in different environments.

	Value	Bytes [Hex]	Description
Command	0x10	-	-
Request bytes	7	[0x10] [D1] … [D6]	[D1]: Reserved, default 0x64 [D2]: Enable/disable auto correction (0: enable all (default), 1: enable only ABC, 2: enable only fast ABC, 3: disable all) [D3]: Correction cycle period (1-30 days, default is 7 days) [D4]: High byte CO_2 baseline value, default 0x01 (400) [D5]: Low byte CO_2 baseline value , [D6]: Reserved, default 0x64
Response bytes	8	[0x10] [D1] … [D6] [CS]	[D1]: Reserved, default 0x64 [D2]: Enabled/disabled auto correction (0: enabled all (default), 1: enabled only ABC, 2: enabled only fast ABC, 3: disabled all) [D3]: Correction cycle period (1-30 days optional, default is 7 days) [D4]: High byte CO_2 baseline value, default 0x01 (400) [D5]: Low byte CO_2 baseline value , [D6]: Reserved, default 0x64

7.2.5.4 Set CO_2 Baseline Manually

Correct current CO_2 reading to a target concentration. Use the following equation to calculate D1 and D2. The correction range is 400 to 1500 ppm. Convert the decimal values to hexadecimal before writing them to the sensor.

$$CO_2 = [D1] \cdot 256 + [D2]$$

Before correction, make sure that the CO_2 concentration in the current environment is in the correction target range. Let the RRH47000 stabilize in the environment for at least 2 minutes before beginning the correction.

	Value	Bytes [Hex]	Description
Command	0x03	-	-
Request bytes	3	[0x3] [D1] [D2]	[D1] and [D2]: CO_2 concentration in ppm
Response bytes	4	[0x03] [D1] [D2] [CS]	[D1] and [D2]: CO_2 concentration in ppm

Example:

For a CO_2 concentration of 600 ppm, the bytes D1 and D2 need to represent this value in hexadecimal form. The value 600 in decimal is 0x258 in hexadecimal. Therefore, D1 would be 0x02 and D2 would be 0x58.

7.2.5.5 Get Serial Number

Returns the serial number of the sensor. The serial number received needs to be converted to ASCII.

Note: Use a 160ms delay between making the request and reading the response.

	Value	Bytes [Hex]	Description
Command	0x1F	-	-
Request bytes	1	[0x1F]	-
Response bytes	18	[0x1F] [D1] to [D16] [CS]	[D1] to [D16]: Serial number in hexadecimal

Example:

Serial number in hex: 35 31 39 32 34 31 30 44 30 31 31 31 30 38 36 39

Corresponding serial number in ASCII: 5192410D01110869

7.2.5.6 Get Firmware Version Number

Returns the firmware version of the RRH47000. The received firmware version number needs to be converted to ASCII.

	Value	Bytes [Hex]	Description
Command	0x1E	-	-
Request bytes	1	[0x1E]	-
Response bytes	13	[0x1E] [D1] [D11] [CS]	[D1] to [D11]: Firmware version number as hexadecimal

Example:

Serial number in hex: 56 35 2E 30 35 5F 36 42 52 30 32

Corresponding serial number in ASCII: V5.05_6BR02

Note: ASCII code 20 represents a blank space.

7.3 Calculating CO_2, Temperature, and Relative Humidity Output

The following table contains all the necessary equations to calculate CO_2 levels, temperature, and relative humidity from the sensor output bytes. To calculate the measurement results of the RRH47000, the hexadecimal values must be transformed into decimal first. Use the decimal values in the equations of Table 14 to calculate the corresponding measurement outputs.

Table 14. Equations to Calculate the Measurement Outputs

	Condition	Equation	Unit
CO_2	-	$CO2 = [D1] \cdot 256 + [D2]$	ppm
Temperature	$\{[D5] \cdot 256 + [D6]\} < 32768$	$T = \frac{[D5] \cdot 256 + [D6]}{100}$	°C
	$\{[D5] \cdot 256 + [D6]\} \geq 32768$	$T = \frac{[D5] \cdot 256 + [D6] - 65536}{100}$	
Relative Humidity	-	$RH = \frac{[D7] \cdot 256 + [D8]}{100}$	%RH

The following is an example of how to calculate output for the RRH47000.

Table 15. Calculation Example for Measurement Outputs

	CO_2		Ignore		Temperature		Rel. Humidity	
Data Byte	1	2	3	4	5	6	7	8
Hexadecimal	0x02	0x29	0x00	0xF7	0x0B	0x72	0x12	0xAD
Decimal	2	41	Ignore		11	114	18	173
Equation [1]	$02 \cdot 256 + 41$				$\frac{11 \cdot 256 + 114}{100}$		$\frac{18 \cdot 256 + 173}{100}$	
Result	553ppm				29.3°C		47.81%RH	

1. The equations can be found in Table 14.

7.4 Checksum Calculation

To calculate the checksum, use the following equation:

$$CS = 256 - \sum Bytes \ \% \ 256$$

Table 16 provides an example of a checksum calculation. The bytes and hexadecimal values have been taken from Table 9.

Table 16. Example Checksum Calculation for UART

Byte	1	2	3	4	5	6	7	8	9
Byte Description	Start	Length	Command	Data	Data	Data	Data	Data	Data
Hexadecimal Value	0x11	0x07	0x10	0x64	0x00	0x07	0x01	0x90	0x64
Decimal Value	17	7	16	100	0	7	1	144	100
Equation	$CS = 256 - (17 + 7 + 16 + 100 + 0 + 7 + 1 + 144 + 100)\%256$								
Result	$CS = 256 - 392\%256 = 256 - 136 = 120$ = **0x78**								

Table 17. Example Checksum Calculation for I²C

Byte	1	2	3	4	5	6	7	8	9
Byte Description	Command	Data	Data	Data	Data	Data	Data	Data	Data
Hexadecimal Value	0x31	0x02	0x29	0x00	0xF7	0x0B	0x72	0x12	0xAD
Decimal Value	49	2	41	0	247	11	114	18	173
Equation	$CS = 256 - (49 + 2 + 41 + 0 + 247 + 11 + 114 + 18 + 173)\%256$								
Result	$CS = 256 - 655\%256 = 256 - 143 = 113$ = **0x71**								

8. Storage and Handling

Recommendation: Once the sensors are removed from their original packaging, store them in metal-in antistatic bags.

Avoid using polyethylene antistatic bags as they may affect the RHT sensors accuracy.

The nominal storage conditions are 10°C to 50°C and humidity levels within 20% to 60%RH. If stored outside of these conditions for extended periods of time, the RHT sensor readings may exhibit an offset. The RHT sensor can be reconditioned and returned to its calibration state by applying the following recondition procedure:

1. Bake at a temperature of 85°C with a humidity < 10% RH for 24 hours.
2. Rehydrate the sensor at a humidity of 75% RH and a temperature between 20°C to 30°C for 12 to 14 hours.

9. Marking Diagram

Figure 11. Sticker Label RRH47000

Code Component	Description	Example
Product Code (Char 1 – 3)	Code representing the product	‘519’
Date Code (Char 4 – 7)	Production date in the format YYWW$ (where $ = location code)	‘2410’
Location Code (Char 8)	Code indicating the site of production	‘D’
Lot# in Production Week (Char 9 – 10)	Number indicating the lot produced within the production week (ranges from 01 to 99)	‘01’
Internal Traceability Number (Char 11 – 16)	Internal tracking number for traceability	‘110869’

10. Quality and Reliability

The RRH47000 series is available as a qualified product for consumer and industrial market applications. All data specified parameters are guaranteed if not stated otherwise.

11. Package Outline Drawings

The package outline drawings are located at the end of this document and are accessible from the Renesas website. The package information is the most current data available and is subject to change without revision of this document.

12. Ordering Information

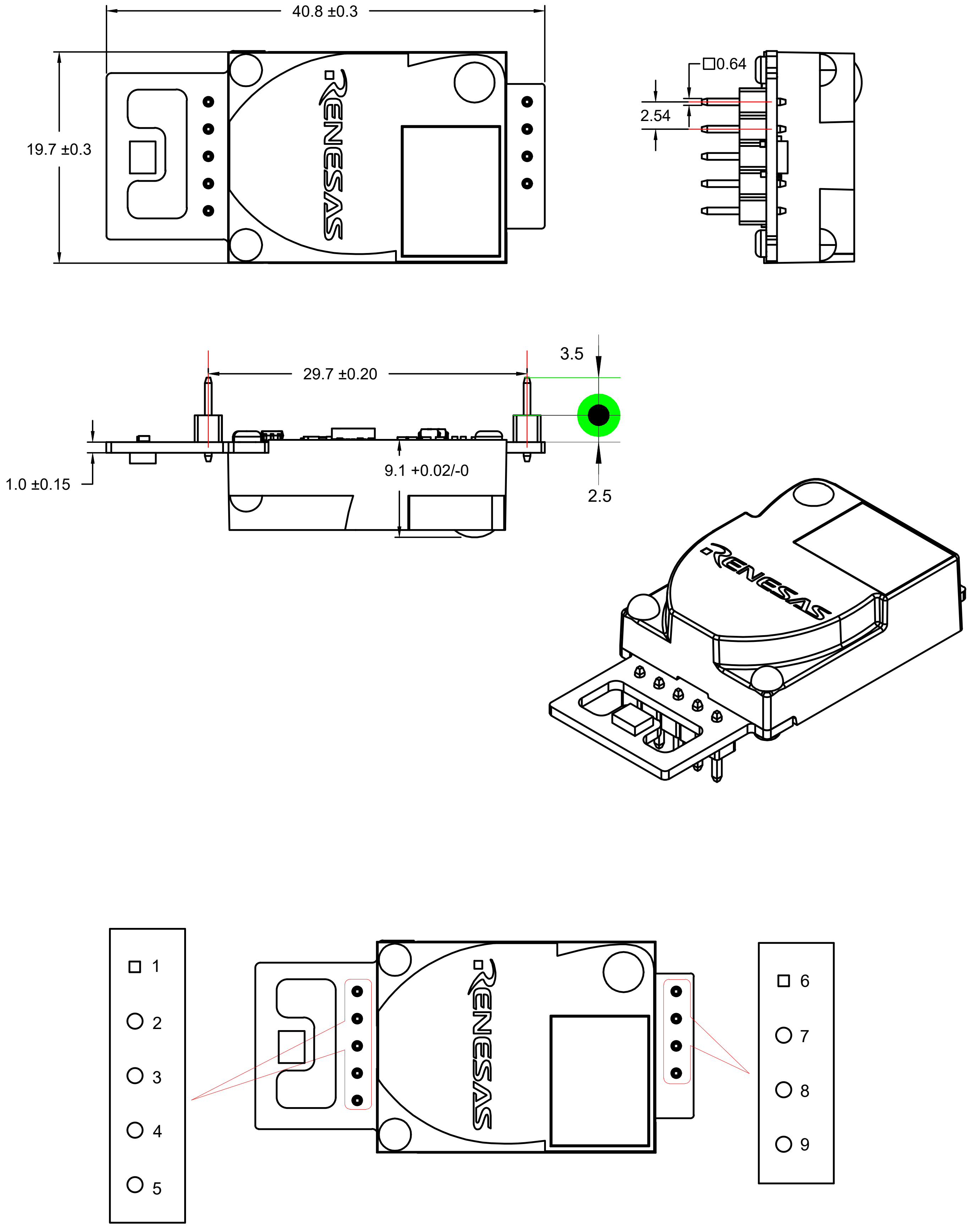
Part Number	Package Description	Description	MOQ (pcs)
RRH47000-A1V	40.8 × 19.7 × 9.1 mm MN0009AA Module	NDIR CO_2 sensor module with RHT sensor	540
RRH47000-EVK	RRH47000 Evaluation Kit including the RRH47000 Sensor Board, ESCom Communication Board (USB Interface), and Micro-USB Cable. (The Environmental Sensors Evaluation Software is available for download free of charge on the RRH47000-EVK product page.)		
RRH47000-EVK-DB	RRH47000 Sensor Board as spare part for RRH47000 Evaluation Kit.		

13. Revision History

Revision	Date	Description
1.00	Nov 11, 2024	Initial release.

Package Outline Drawing

Package Code: MN0009AA
9-Module 40.8 x 19.7 x 9.1 mm, 2.54mm Pitch
PSC-5046-01, Revision: 02, Date Created: Dec 3, 2024



Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.