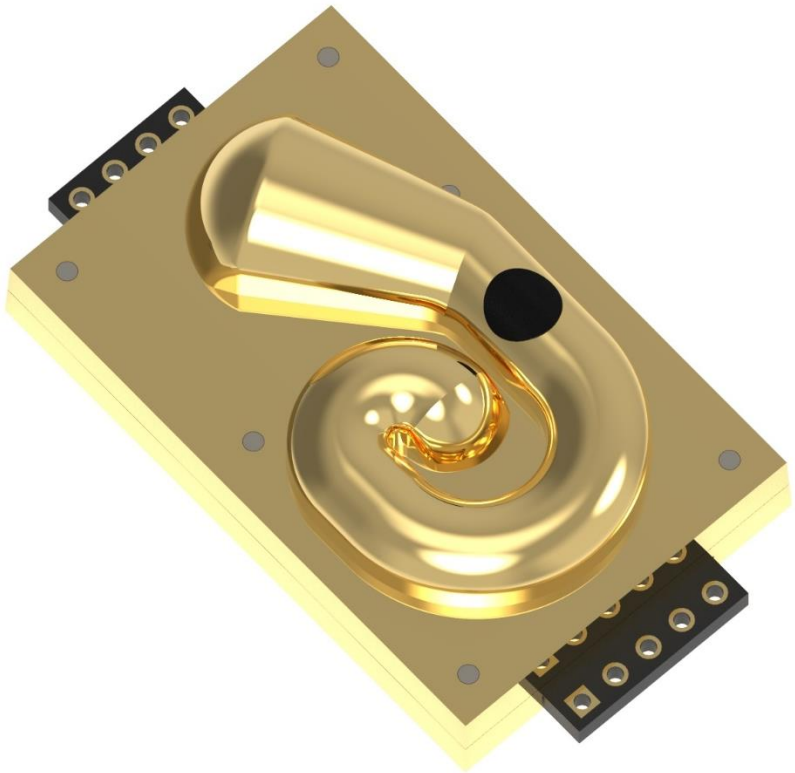
Tempus
Sensor for Lifecare Service

Datasheet

Rev. 1.2 / May 2023

TES0704

Dual Channel NDIR Refrigerant Sensor Module



TES0704- Dual Channel NDIR Refrigerant Sensor Module

Introduction

TES0704 is an ultra-small dual channel NDIR sensor module with a digital interface for measuring refrigerant concentration for R-32, R-290

The dual-channel NDIR refrigerant sensor module, TES0704, has excellent performance on C-H bond stretching overtone by using Tempus' high end NDIR sensors with unique module structure and fast speed.

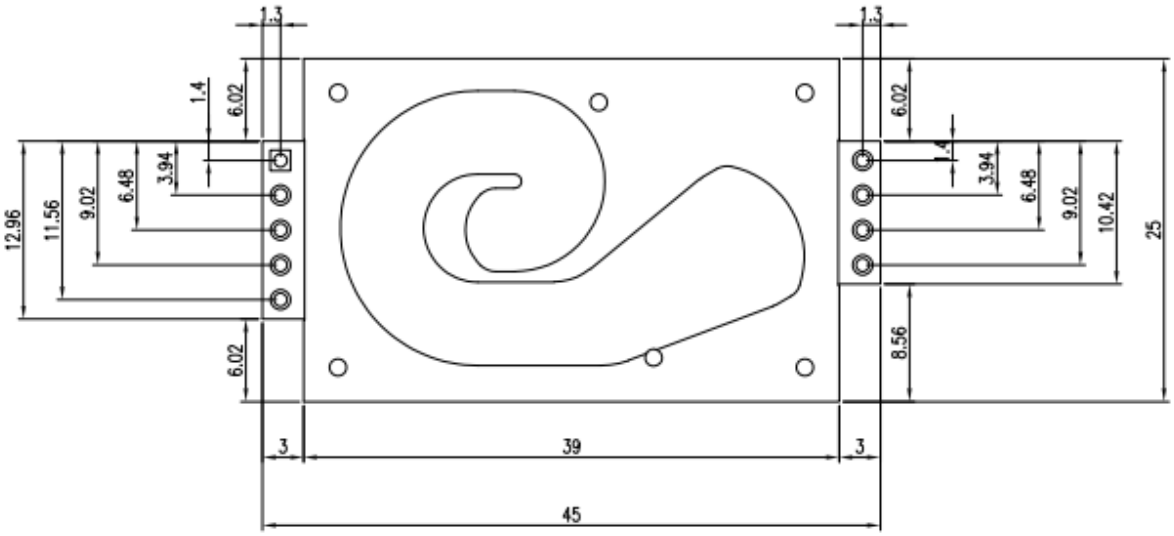
TES0704 is suitable for measuring a gas leak alarming, and it is an ideal solution for customers who want low concentration precision.

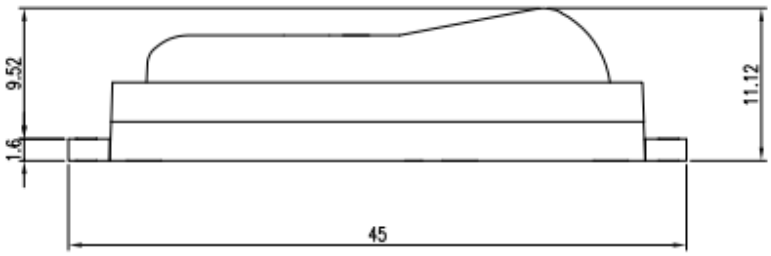
TES0704 is has high accuracy and stability with reasonable price. It is offered to a customer after being calibrated according to a customer's requested gas measurement. (R32/ R290)

Features

- Ultra-Small Size
- Low Power
- High Accuracy and Excellent stability
- Absolute measurement with Dual-channel NDIR sensor
- Pre-calibrated and ready-to-use
- Digital interface using RS-232
- ABC feature support

Dimension and Connector Pin Information





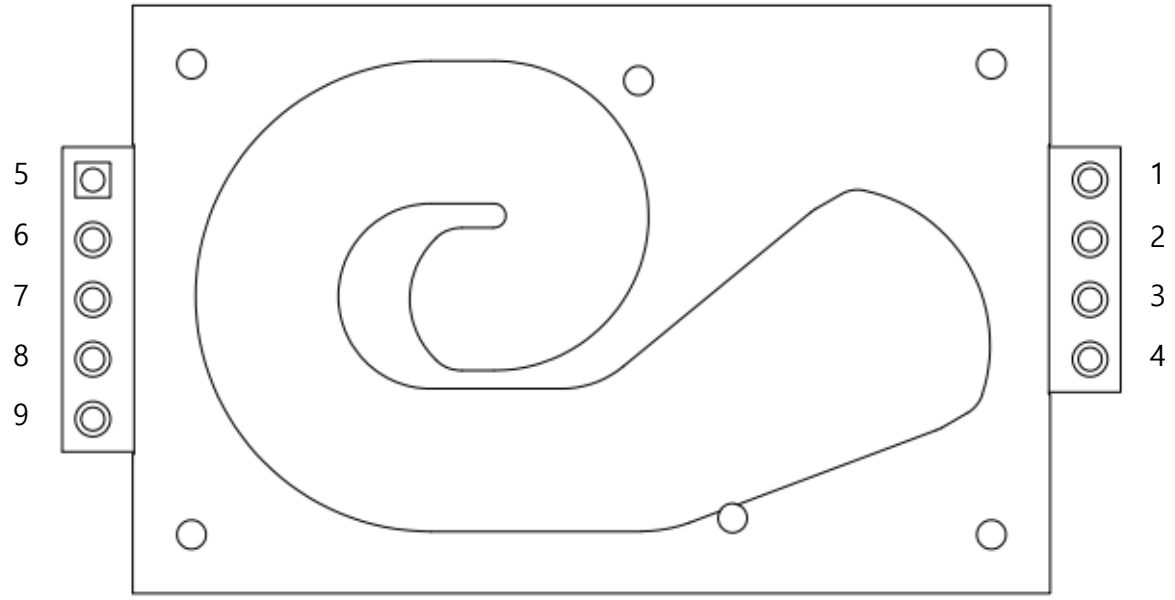
Dimension and Pin Assignment (mm, TOP View)

*** Module Total Dimension: 45mm (W) x 25mm (L) x 11.12mm (H)**

*** Specification is subject to change without notice.**

Pin Description

Pin No.	Pin Name	Description
1	+5V	+5V Input
2	GND	Ground
3	NC	No Connection
4	NC	No Connection
5	+3.3V	+3.3 Input
6	RX	RX: 3.3V(Typ.) CMOS Level Signal
7	TX	TX: 3.3V(Typ.) CMOS Level Signal
8	NC	No Connection
9	NC	No Connection



*** Pin configuration is subject to change without notice.**

Item		Specification
General	Product Name	Dual Channel NDIR Refrigerant sensor module
	Operation Technology	Non-dispersive Infrared(NDIR)
	Operating Temperature	-10℃ ~ 50℃ (Non-condensing)
	Operating Humidity	0 ~ 95% RH (Non-condensing)
	Operating Environment	Residential, Commercial spaces
	Storage Temperature	-20℃ ~ 80℃(Non-condensing)
Refrigerant Measurement	Sensing Method	Dual Channel NDIR (Non-dispersive Infrared)
	Measurement Range	R32: 0 to 5,000 ppm R290: 0 to 21,000ppm
	Accuracy	R32: ±(50ppm +3% of measured value) ±0.0050% at 0% R32 ±0.0200% at 0.5% R32
		R290: ±(50ppm +3% of measured value) ±0.0050% at 0% R290 ±0.0680% at 2.1% R290
	Warm-up Time	< 1 minute
	Response Time	< 3 minutes (diffusion)
	Sampling Interval	5 seconds
Electrical data	Power Input	5 VDC @5% (4.75Vdc ~ 5.25Vdc) Average current 33.0 mA@5V Peak current 550 mA@5V IR Lamp On 150 mA@5V IR Lamp Off 30 mA@5V
	Output connector	6 pins (Terminals not mounted)
Output interface	Digital Output	RS-232(UART)
Weight	Module Weight	7.2g

Calibration

TES0704 is pre-calibrated sensor, but it is very sensitive. Due to shock during the transportation or incorrect installation, TES0704 can be drifted to unrelated ambient concentrations. So TES0704 supports ABC Calibration (AC) functions so that users can easily calibrate at any time.

AC compensates the drift to 0ppm immediately. If used in a normal environment, it is recommended to use the AC function to compensate the sensor. If TES0704 is used in a space where C-H bonds always exists, turn off the ABC function.

For accuracy compensation, TES0704 should be supplied power at least 5 minutes in a well-ventilated and stable environment prior to calibration. Also, it should be done at room temperature.

User can manually calibrate the sensor to 0ppm via the "0x52 (AC)" command set. Please refer to the ABC_CAL.

Digital Interfaces

The TES0704 has RS-232 digital inteface. Users control the register map through digital inteface by reading and writing register values. RS-232 digital inteface is described in this section, and the command/response map will be introduced in the subsequent section.

1. UART Interface

TES0704 supports a RS-232 serial interface. Pin Rx is UART Rx (input to sensor) and Pin Tx is UART Tx (output from sensor). In details, UART conditions are :

- 9600 Baud rates
- No Parity Bit
- 1 Stop bit
- 8 Data bits
- No Flow control

TES0704 periodically pushes ppm information to the host via UART every 5 sec.

The host can get additional inforamtion such as version information and serial number through command/request data, and its message format is as follows.

1.1 UART Protocol

Format of the Message

UART Request Message Format

2 bytes	1 byte	1 byte	n byte	2 bytes
Sync	Command	Length	Data	Crc16

Type	*Size*	*Description*
Sync	2 bytes	Sync Data, 0xAA55
Command	1 byte	Command code. details in Command List
Length	1 byte	Data Size Field except CRC16
Data	n byte	Data to be transmitted
CRC16	2 bytes	Error check code

UART Response Message Format

2 bytes	1 byte	1 byte	n byte	2 bytes
Sync	Response	Length	Data	Crc16

Type	*Size*	*Description*
Sync	2 bytes	Sync Data, 0xBB66
Response	1 byte	Response code.
Length	1 byte	Data Size Field except CRC16
Data	n byte	Data to be reported
CRC16	2 bytes	Error check code

- Host must include the **Cyclical Redundancy Check fields (CRC16)** at the end of the message for error check.
- All command/response is hexadecimal.

1.2 Command/Response List

Name	***Code***	***R/W***	***Data Type***	***Description***
CMD_GET_VER	0x10	RO	-	Read Firmware Version Information
CMD_GET_SER	0x12	RO	-	Read Serial Number
CMD_GET_PPM	0x14	RO	-	Read PPM
CMD_GET_ABC_ON	0x20	RO	-	Read ABC mode on/off
CMD_SET_ABC_ON	0x22	WO	uint8_t	Write ABC mode on/off
CMD_GET_ABC_DUR	0x24	RO	-	Read ABC Period
CMD_SET_ABC_DUR	0x26	WO	uint16_t	Write ABC Period
CMD_RESET_PWR	0x1E	WO	-	Reset power supplied to TES0704

1.3 Examples of protocol

Read PPM

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x15	Response
0x02	Size
xx	Gas Concentration (LSB)
xx	Gas Concentration (MSB)
xx	CRC (LSB)
xx	CRC (MSB)

To calculate the ppm, do the following.

$$\mathbf{PPM = MSB\ (HEX\ to\ DEC) \times 256 + LSB\ (HEX\ to\ DEC)}$$

*** If the host wants to get inforamtion such as version information, serial number and alarm setting, the host should send 'stop ppm push' command first. Please refer to the STOP_PPM_PUSH.**

FIRMWARE VERSION (CMD_GET_VER)

This command will return the current firmware version number.
Version number is something such as v1.1.2

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x10	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x11	Response
0x03	Size
xx	Major
xx	Minor
xx	Build
xx	CRC (LSB)
xx	CRC (MSB)

Example 1 – request / response to read the Firmware Version

SERIAL NUMBER (CMD_GET_SER)

This command will return serial number of the sensor. Length of serial number is 8 bytes.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x12	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x13	Response
0x08	Size
xx	S/N Byte 0 (LSB)
…	…
…	…
xx	S/N Byte 7 (MSB)
xx	CRC (LSB)
xx	CRC (MSB)

Example 2 – request / response to read the Serial Number

ABC ON/OFF (CMD_GET_ABC_ON)

This command will report ABC ON state of TES0704. If value is 0, ABC feature will be turned off.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x20	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x21	Response
0x02	Size
0x00	
xx	ABC ON/OFF
xx	CRC (LSB)
xx	CRC (MSB)

Example 5 – request / response to ABC feature

~~ABC ON/OFF (CMD_SET_ABC_ON)~~

This command will enable/disable ABC feature of TES0704. If value is 0, ABC feature will be turned off.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x22	Command
0x02	Size
0x00	
xx	ABC ON/OFF
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x23	Response
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Example 6 – request / response to ABC feature

ABC DURATION (CMD_GET_ABC_DUR)

This command will report ABC period of TES0704.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x24	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x25	Response
0x02	Size
xx	ABC Duration (LSB)
xx	ABC Duration (MSB)
xx	CRC (LSB)
xx	CRC (MSB)

Example 7 – request / response to ABC Duration

~~ABC DURATION (CMD_SET_ABC_DUR)~~

This command will set ABC period of TES0704. By default, its value is "1" which means 1 days. Period starts from "0".

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x26	Command
0x02	Size
xx	ABC Duration (LSB)
xx	ABC Duration (MSB)
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x27	Response
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Example 8 – request / response to ABC Duration

~~POWER RESET (CMD_RESET_PWR)~~

This command will reset the power supplied to TES0704. It is not resetting the value of the register, but re-applying power.

Therefore, re-supplying power to the sensor does not change the ABC on/off setting, ABC duration, ABC offset, or alarm, and the value set by the user is loaded as it was.

For example, if the user sends a command to set the ABC duration to 1 day before re-supplying the power, the ABC duration remains as 1 day even if power is supplied again.

However, the time counter used to check whether the ABC duration has passed is stored in RAM, not flash. So when power is supplied again, the count is lost, and it starts again from 0.

For example, if the ABC duration is 7 days, and the sensor is powered on continuously for 5 days and then powered off, the ABC calibration will not be performed after 2 days after the next power is supplied, but it will be performed after 7days.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x1E	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Example 11 – request to reset the power supplied to TES0704

Response (UART)

No response. After sending this command, the power is reset immediately.

1.4 CRC API

```
uint16_t Calculate_CRC16 ( uint8_t *cmd , int cmd_length )
{
        uint16_t ret = 0xffff , polynomial = 0xa001 ;
        int shift = 0x0;
        for ( int i = cmd_length – 1 ;   i >= 0 ; i-- )   {
              unt16_t code = ( uint16_t )( cmd [ cmd_length -1 - i ]   & 0xff );
              ret = ret ^ code ;
              shift = 0x0;
              while ( shift <= 7 )   {
                    if ( ret & 0x1 ) {
                          ret = ret >> 1;
                          ret = ret ^ polynomial ;
                    } else {
                          ret = ret >> 1;
                    }
                    shift++;
             }
        }
        return ret;
}
```

2. ABC feature

TES0704 supports ABC (Automatic Background Calibration) feature.
TES0704 monitors the trends and min/max value of C-H ppm, and counts its timeframe.
Once its time count reaches at "ABC Period", ABC algorithm will apply and count its timeframe from zero again.
Its time counting is working while its power is supplying, so in case of its power is off and on, its time counting starts from zero again and counter will be reset.
So, if user wants to make ABC feature working, need to keep power on without off.

User can enable or disable ABC feature by "0x22" command set with given 1-byte option value.
By default, its ABC feature is "ON".
Please refer to the ABC_ONOFF.

Also, user can change its ABC duration by "0x26" command set with given duration value.
By default, its ABC duration is "1" and its range is "0 ~ 30".
Please refer to the ABC_DURATION.

3. Stop ppm push

By default, TES0704 pushes ppm value periodically to host every 5 sec., so host doesn't need any action to get ppm value.

If host doesn't need ppm push, that is, if host wants to get ppm whenever host requests, there is two ways to do this via CMD_GET_PPM, 0x14 or CMD_GET_VER, 0x10.

While TES0704 is pushing ppm periodically, if host sends CMD_GET_PPM or CMD_GET_VER, ppm pushing will stop and no more ppm data will be pushed.

After then, host can read ppm data by sending CMD_GET_PPM command.

If host wants to enable ppm pushing feature again, try to reset TES0704 module.

Example :

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x14	Command
0x00	Size
xx	CRC (LSB)
xx	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x15	Response
0x02	Size
xx	Gas Concentration (LSB)
xx	Gas Concentration (MSB)
xx	CRC (LSB)
xx	CRC (MSB)