

Datasheet (Preliminary)

Rev. 1.4 / June 10, 2022

TES0903

Dual Channels NDIR CO2 Sensor Module

TES0903- Dual Channels NDIR CO₂ Sensor Module

Introduction

TES0903 is an ultra-small dual channels NDIR sensor module with digital interfaces for CO₂ concentration measurement.

The dual-channel NDIR CO₂ sensor module, TES0903, offers excellent performance that overcomes the drift of a light source and CO₂ over time by using Tempus' faster response NDIR sensors with its unique module structure.

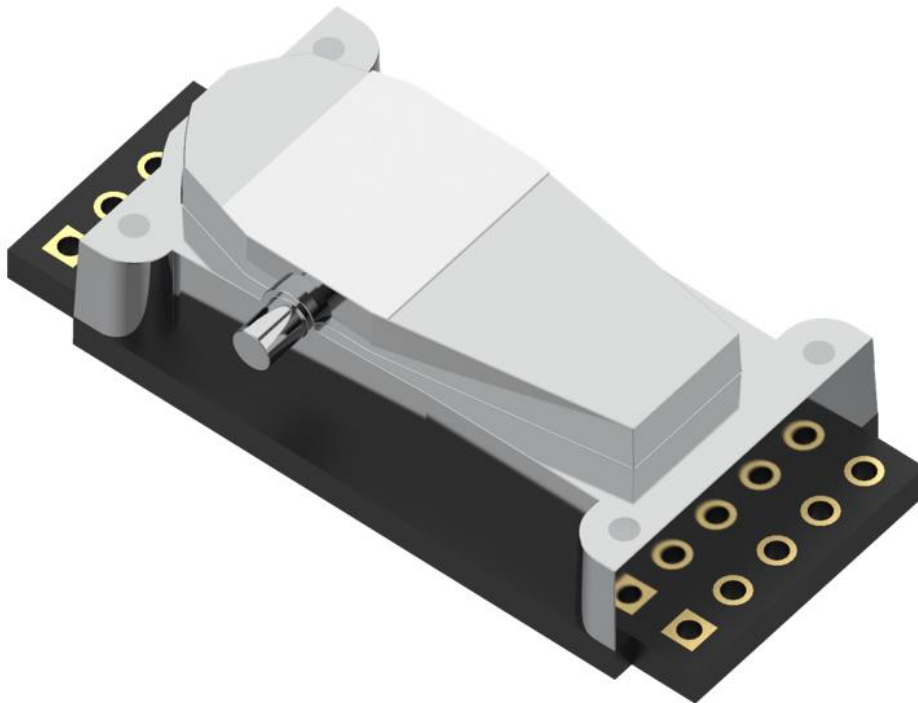
TES0903 series are an ideal solution for customers seeking simple integration into user's product. TES0903 is with performance of high accuracy and stability at a reasonable price. It is the individually pre-calibrated module and possible to measure CO₂ concentration up to 5,000ppm. (Other maximum concentration is available upon request).

Features

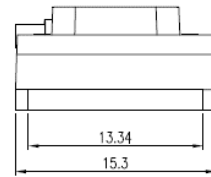
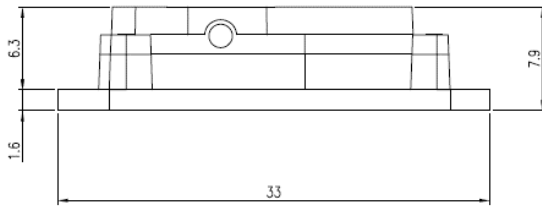
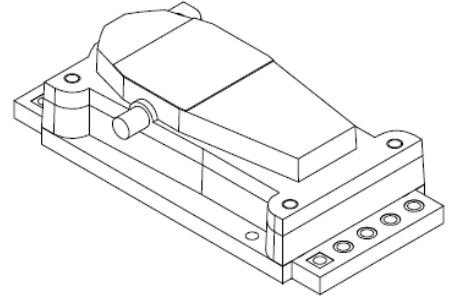
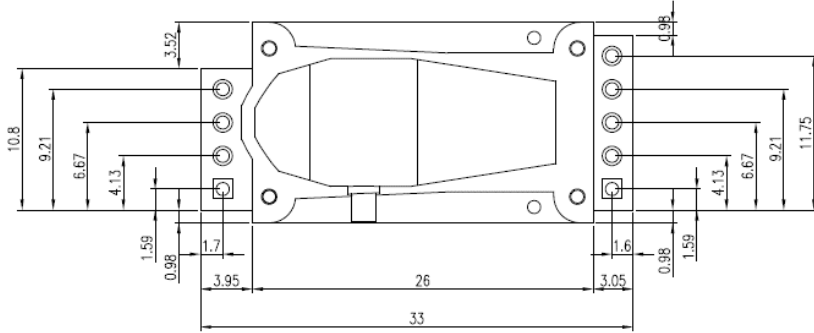
- Ultra-Small Size
- Low Power
- High Accuracy and Excellent stability
- Absolute measurement with Dual-channel NDIR sensor
- Pre-calibrated and ready-to-use
- Digital interface using RS232, I²C and PWM
- Digital output for CO₂ level warning. (ex: Warning When >2000ppm)
- ABC feature support
- Manual Calibration support (room temperature, room pressure)

Applications

- HVAC (Heating, Ventilation, Air-conditioning)
- Air conditioner
- Environment management system
- Indoor air quality (IAQ) control
- Automatic air ventilation system
- Smart farm



Dimension and Connector Pin Information



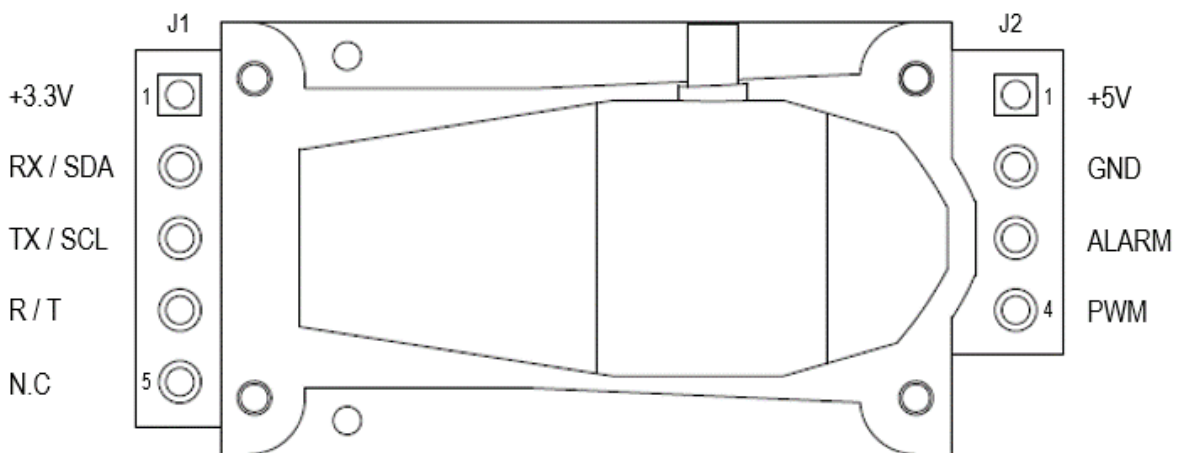
Dimension and Pin Assignment (mm, TOP View)

- * Module Total Dimension: 33.0mm (W) x 15.3mm (L) x 7.9mm (H)
- * Specification is subject to change without notice.

Pin Description

J1		
Pin No.	Pin Name	Description
1	+3.3V	Power supply output (+3.3V / 100mA)
2	RX/SDA	UART-RX / I ² C data, compatible with 3.3V communication
3	TX/SCL	UART-TX / I ² C clock, compatible with 3.3V communication
4	R/T	Output mode exchange High level or floating is UART communication mode Low level is I ² C communication mode
5	N.C	-

J2		
Pin No.	Pin Name	Description
1	+5V	Power supply input (+5V)
2	GND	Power supply input (GND)
3	ALARM	ALARM Signal
4	PWM	PWM Output: 3.3V(Typ.) CMOS Level Signal



* Pin configuration is subject to change without notice.

Specifications

*Digital interface is subject to change without notice.

Item		Specification
General	Product Name	Dual Channel NDIR CO2 sensor module
	Operation Technology	Non-dispersive Infrared(NDIR)
	Operating Temperature	-10°C ~ 50°C (Non-condensing)
	Operating Humidity	0 ~ 95% RH (Non-condensing)
	Operating Environment	Residential, Commercial spaces
	Storage Temperature	-20°C ~ 80°C(Non-condensing)
CO2 Measurement	Sensing Method	Dual Channel NDIR (Non-dispersive Infrared)
	Measurement Range	400 to 5,000 ppm
	Accuracy	±(50ppm +3% of measured value)
	Warm-up Time	< 2 minutes
	Response Time	< 30 seconds (diffusion)
	Sampling Interval	2 seconds
Electrical data	Power Input	5 VDC @5% (4.75Vdc ~ 5.25Vdc) Average current 25.0 mA@5V Peak current 246 mA@5V IR Lamp On 120 mA@5V IR Lamp Off 10 mA@5V
	Output connector	J1(5 pins), J2(4 pins) (Terminals not mounted)
Output interface	Digital Output	RS232, I ² C, PWM, Alarm CMOS level output
Weight	Module Weight	5.1g

Digital Interfaces

The TES0903 has several digital interfaces such as RS232, I²C, PWM, and alarm indication. Users control the register map through digital interfaces by reading and writing register values. This section describes each digital interface and the command/response map will be introduced in the next section.

1. UART Interface

TES0903 support a RS232 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

Host can get more informations like PPM, version info, serial number, alarm setting via command/request data, and supports two types of protocols.

Its message format is like as below

1.1 UART Protocol (Type-1)

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
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
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.

1.1.1 Command/Response List

Name	Code	Description
CMD_GET_VER	0x10	Read Firmware Version Information
CMD_GET_SER	0x12	Read Serial Number
CMD_GET_PPM	0x14	Read PPM
CMD_GET_ALARM	0x16	Read Alarm Level
CMD_SET_ALARM	0x18	Write Alarm Level
CMD_RESET_PWR	0x1E	Power Reset
CMD_GET_ABC_ON	0x20	Read ABC mode on/off
CMD_SET_ABC_ON	0x22	Write ABC mode on/off
CMD_GET_ABC_DUR	0x24	Read ABC Period
CMD_SET_ABC_DUR	0x26	Write ABC Period
CMD_SET_MANUAL_CAL	0x4C	Calibrate TES0903 to target concentration

READ PPM (CMD_GET_PPM)

This command will return the measured CO₂ result.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x14	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x15	Response
0x02	Length
DATA[0]	Gas Concentration (LSB)
DATA[1]	Gas Concentration (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

To calculate the ppm, do the following.

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

FIRMWARE VERSION (CMD_GET_VER)

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

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x10	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x11	Response
0x03	Length
DATA[0]	Major
DATA[1]	Minor
DATA[2]	Build
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

SERIAL NUMBER (CMD_GET_SER)

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

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x12	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x13	Response
0x08	Length
DATA[0]	S/N Byte 0 (LSB)
...	...
...	...
DATA[7]	S/N Byte 7 (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

ALARM READ (CMD_GET_ALARM)

This command will report Alarm PPM Value in Register.
By default, its value is 2,000.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x16	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x17	Response
0x02	Length
DATA[0]	PPM (LSB)
DATA[1]	PPM (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

ALARM WRITE (CMD_SET_ALARM)

This command will write Alarm Value in Alarm register. If value in Alarm register is 0, Alarm Function will be turned off.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x18	Command
0x02	Length
DATA[0]	Alarm PPM (LSB)
DATA[1]	Alarm PPM (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x19	Response
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

POWER RESET (CMD_RESET_PWR)

This command will reset the power supplied to TES0903. 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 7 days.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x1E	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

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

ABC ON/OFF (CMD_GET_ABC_ON)

This command will report ABC ON state of TES0903. If value is 0, ABC feature will be turned off.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x20	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x21	Response
0x02	Length
0x00	
DATA[0]	ABC ON/OFF
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

ABC ON/OFF (CMD_SET_ABC_ON)

This command will enable/disable ABC feature of TES0903. If value is 0, ABC feature will be turned off.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x22	Command
0x02	Length
0x00	
DATA[0]	ABC ON/OFF
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x23	Response
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

ABC DURATION (CMD_GET_ABC_DUR)

This command will report ABC period of TES0903.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x24	Command
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x25	Response
0x02	Length
DATA[0]	ABC Duration (LSB)
DATA[1]	ABC Duration (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

ABC DURATION (CMD_SET_ABC_DUR)

This command will set ABC period of TES0903. By default, its value is "7" which means 7 days. Period starts from "0".

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x26	Command
0x02	Length
DATA[0]	ABC Duration (LSB)
DATA[1]	ABC Duration (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x27	Response
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

MANUAL CALIBRATION (CMD_SET_MANUAL_CAL)

This command will calibrate TES0903 to 0 ~ 5000 ppm^{*(1)}.

At this moment, the environment temperature should be set to room temperature and room pressure.

Request (UART)

0xAA	Sync (MSB)
0x55	Sync (LSB)
0x4C	Command
0x00	Length
DATA[0]	PPM (LSB)
DATA[1]	PPM (MSB)
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

Response (UART)

0xBB	Sync (MSB)
0x66	Sync (LSB)
0x4D	Response
0x00	Length
CRC[0]	CRC (LSB)
CRC[1]	CRC (MSB)

*(1) When performing manual calibration below 400ppm, please disable ABC function and perform manual calibration.

1.1.2 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-- ) {
        uint16_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;
}
```

1.2 UART Protocol (Type-2)

Format of the Message

UART Request Message Format

1 byte	1 byte	1 byte	n byte	1 byte
HEAD	LEN	CMD	DATA1...DATAn	Check Sum

Type	Size	Description
HEAD	1 byte	Head Data, 0x11
LEN	1 byte	Data length + 1 (including CMD + DATA)
CMD	1 byte	Command
DATA	n byte	Data to be transmitted
Check Sum	1 byte	Error check code = $256 - (\text{HEAD} + \text{LEN} + \text{CMD} + \text{DATA}) \% 256$

UART Response Message Format

1 byte	1 byte	1 byte	n byte	1 byte
HEAD	LEN	CMD	DATA1...DATAn	Check Sum

Type	Size	Description
HEAD	1 byte	Head Data, 0x16
LEN	1 byte	Data length + 1 (including CMD + DATA)
CMD	1 byte	Command
DATA	n byte	Data to be transmitted
Check Sum	1 byte	Error check code = $256 - (\text{HEAD} + \text{LEN} + \text{CMD} + \text{DATA}) \% 256$

1.2.1 Command/Response List

Name	Code	Description
CMD_GET_PPM	0x01	Read PPM
CMD_SET_ABC	0x10	Set ABC ON/OFF and Parameter
CMD_SET_MANUAL_CAL	0x03	Calibrate concentration value of CO ₂
CMD_GET_SERIAL	0x1F	Read TES0903 Serial number
CMD_GET_VER	0x1E	Read Firmware Version Information

READ PPM (CMD_GET_PPM)

This command will return the measured CO₂ result.

Request (UART)

0x11	HEAD
0x01	LEN
0x01	CMD
0xED	Check Sum

Response (UART)

0x16	HEAD
0x05	LEN
0x01	CMD
DATA[0]	PPM (MSB)
DATA[1]	PPM (LSB)
DATA[2]	RESERVED
DATA[3]	RESERVED
CS[0]	Check Sum

To calculate the ppm, do the following.

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

SET ABC ON/OFF AND PARAMETER (CMD_SET_ABC)

This command will enable/disable ABC feature and sets parameters.

Request (UART)

0x11	HEAD
0x07	LEN
0x10	CMD
DATA[0]	RESERVED, Default 100 (0x64)
DATA[1]	ABC ON/OFF (0 : ON, 2 : OFF)
DATA[2]	ABC Duration
DATA[3]	ABC PPM (MSB)
DATA[4]	ABC PPM (LSB)
DATA[5]	RESERVED, Default 100 (0x64)
CS[0]	Check Sum

Response (UART)

0x16	HEAD
0x01	LEN
0x10	CMD
0xD9	Check Sum

To calculate the ABC PPM, do the following.

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

MANUAL CALIBRATION (CMD_SET_MANUAL_CAL)

This command will calibrate TES0903 to 0 ~ 5000ppm^{*(1)}.

At this moment, the environment temperature should be set to room temperature and room pressure.

Request (UART)

0x11	HEAD
0x03	LEN
0x03	CMD
DATA[0]	PPM (MSB)
DATA[1]	PPM (LSB)
CS[0]	Check Sum

Response (UART)

0x16	HEAD
0x01	LEN
0x03	CMD
0xE6	Check Sum

To calculate the PPM, do the following.

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

*(1) When performing manual calibration below 400ppm, please disable ABC function and perform manual calibration.

SERIAL NUMBER (CMD_GET_SER)

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

Request (UART)

0x11	HEAD
0x01	LEN
0x1F	CMD
0xCF	Check Sum

Response (UART)

0x16	HEAD
0x0B	LEN
0x1F	CMD
DATA[0]	RESERVED, Default 0x00
DATA[1]	S/N MSB (MSB), S/N MSB size is 4 bytes
...	...
DATA[4]	S/N MSB (LSB)
DATA[5]	RESERVED, Default 0x00
DATA[6]	S/N LSB (MSB), S/N LSB size is 4 bytes
...	...
DATA[9]	S/N LSB (LSB)
CS[0]	Check Sum

FIRMWARE VERSION (CMD_GET_VER)

This command will return the current firmware version number.
Version number is something such as TP00.01.02

Request (UART)

0x11	HEAD
0x01	LEN
0x1E	CMD
0xD0	Check Sum

Response (UART)

0x16	HEAD
0x0B	LEN
0x1E	CMD
DATA[0]	RESERVED, Default 'T' (0x54)
DATA[1]	RESERVED, Default 'P' (0x50)
DATA[2]	Major-1 (ASCII)
DATA[3]	Major-2 (ASCII)
DATA[4]	Default '.' (0x2E)
DATA[5]	Minor-1 (ASCII)
DATA[6]	Minor-2 (ASCII)
DATA[7]	Default '.' (0x2E)
DATA[8]	Build-1 (ASCII)
DATA[9]	Build-2 (ASCII)
CS[0]	Check Sum

Note : TP00.01.02

	Major-1	Major-2		Minor-1	Minor-2		Build-1	Build-2
TP	0x30	0x30	0x2E	0x30	0x31	0x2E	0x30	0x32

2. I²C Interface

2.1. I²C Overview

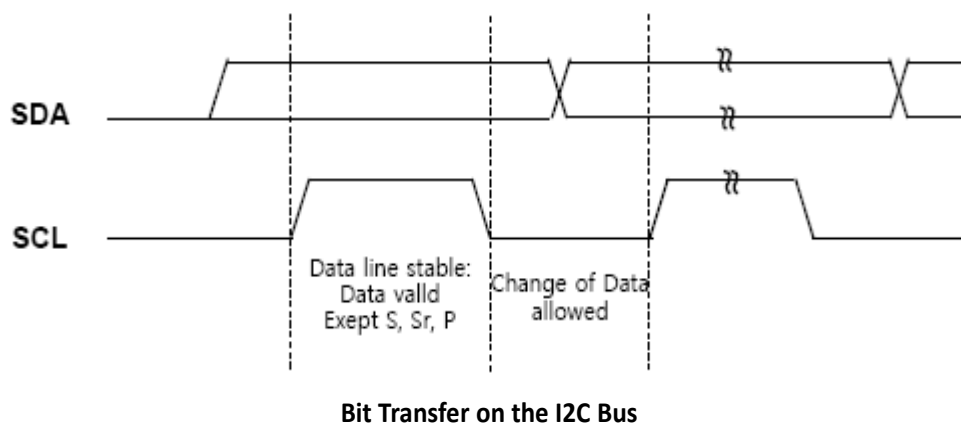
The I²C is one of industrial standard serial communication protocols, and which uses 2bus lines Serial Data Line (SDA) and Serial Clock Line (SCL) to exchange data. Because both SDA and SCL lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

- Compatible with I²C interface
- Up to 400KHz data transfer speed
- Support two 7-bit slave address
- Slave operation only

2.2. I²C Bit Transfer

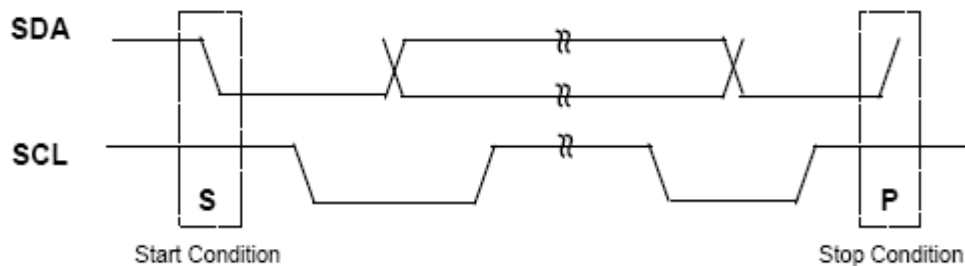
The data on the SDA line must be stable during High period of the clock, SCL. The High or Low state of the data line can only change when the clock signal on the SCL line is Low.

The exceptions are Start(S), repeated Start(Sr) and Stop(P) condition where data line changes when clock line is high.



2.3. Start / Repeated Start / Stop

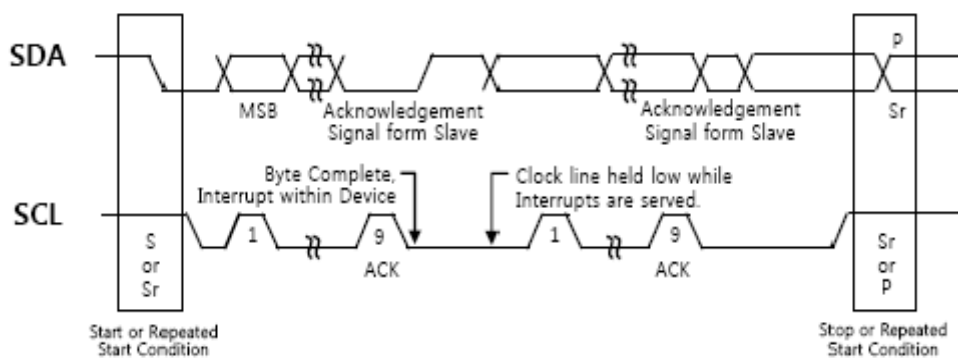
One master can issue a Start(S) condition to notice other devices connected to the SCL, SDA lines that it will use the bus. A Stop(P) condition is generated by the master to release the bus lines so that other devices can use it. A high to low transition on the SDA line while SCL is high defines a Start(S) condition. A low to high transition on the SDA line while SCL is high defines a Stop(P) condition. Start and Stop conditions are always generated by a master. The bus is considered to be busy after Start condition. The bus is considered to be free again after Stop condition, the bus is busy between Start and Stop condition. If a repeated Start condition (Sr) is generated instead of Stop condition, the bus stays busy. So, the Start and repeated Start conditions are functionally identical.



Start and Stop Condition

2.4. Data Transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCL low to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

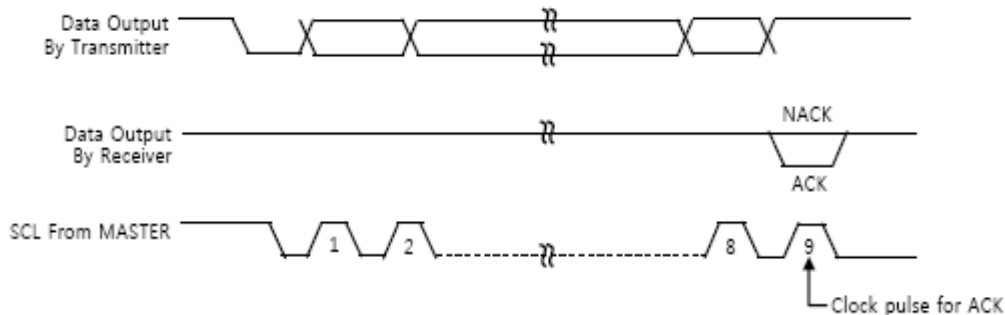


Stop or Repeated Start Condition

2.5. Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA line(High) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable Low during the High period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because its performing some real time function, the data line must be left High by the slave. and also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDA line (Data Packet). The master can then generate either a Stop condition to abort the transfer, or a repeated Start condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a Stop or repeated Start condition.



Acknowledge on the I2C Bus

2.6. Operation

The I²C is byte-oriented serial protocol and data transfer between master and this slave device is initiated by a start condition(S) from master. After start condition, the master sends 7-bit slave address and 1-bit read-write control bit. We call these 8-bit data address packet. The next bytes followed by address packet are all data packet unless another start condition is detected before a stop condition. The 2nd byte sent from master after address packet with write direction is interpreted as base register or memory address byte. And this base address is incremented only when master transmits more than 2 bytes after start condition because the 2nd byte is register address field.

2.6.1. Write Protocol (Master Transmitter)

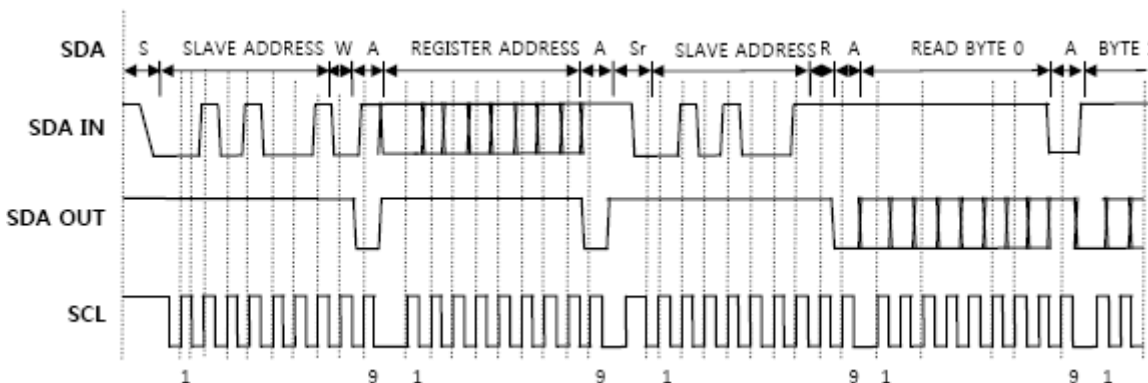
The master transmits a start condition(S), slave address and Write bit. If the high 7-bits of address packet equal to the device’s slave address, the TES0903 acknowledges by pulling down the SDA line at the 9th SCL clock period. After address packet and acknowledge bit, the master transmits data which is used for base address accessing internal memory or register of the device. The master transmits a number of data to be written and the slave always acknowledges for every data received. To finish transfer the master sends a stop condition regardless of the acknowledgement. The destination address for incoming data byte increments automatically by one data packet. For example, if master transmits 5 data bytes including a base address(=register address in the following figure) byte and the base address is configured as 00H, the internal address is defined as 00H for 1st data byte, 01H for 2nd data byte, 02H for 3rd data byte and 03H for 4th data byte. This applies to Read Protocol also.



I2C Write Protocol

2.6.2. Read Protocol (Master Receiver)

The master transmits a start condition(S), slave address and Write bit. If the high 7-bits of address packet equal to the device’s slave address, the TES0903 acknowledges by pulling down the SDA line at the 9th SCL clock period. After address packet and acknowledge bit, the master transmits a data which is used for base address accessing internal memory or register of the device. To initiate read operations, the master sends repeated start condition and slave address with Read bit. After this address packet, the master reads data bytes until it does not acknowledge. Note that to send a stop condition after receiving last data byte, the master must generate a NACK (not acknowledging) on the last data byte received. Like Write Protocol, the read address increases by 1 after every read byte. Note that the transfer direction changes in this protocol.



I2C Read Protocol

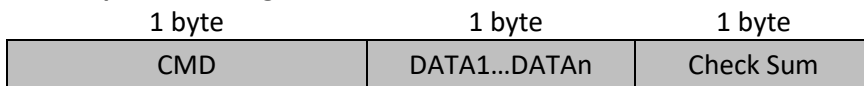
Format of the Message

I²C Request Message Format



Type	Size	Description
CMD	1 byte	Command
DATA	n byte	Data to be transmitted

I²C Response Message Format



Type	Size	Description
CMD	1 byte	Command
DATA	n byte	Data to be transmitted
Check Sum	1 byte	Error check code = 256 - (HEAD + LEN + CMD + DATA) % 256

2.7. I²C Command List

The slave address is 0x31, the data command of the slave device is as below.

Name	Code	Description
CMD_GET_PPM	0x01	Read PPM
CMD_SET_ABC	0x10	Set ABC ON/OFF and Parameter
CMD_SET_MANUAL_CAL	0x03	Calibrate concentration value of CO ₂
CMD_GET_SERIAL	0x1F	Read TES0903 Serial number
CMD_GET_VER	0x1E	Read Firmware Version Information

2.8. I²C Check Sum

Data check bit = -(CMD + DATA1 + + DATAn)

Only keep the lowest byte.

READ PPM (CMD_GET_PPM)

This command will return the measured CO₂ result.

Request (I²C)

0x01	CMD
------	-----

Response (I²C)

0x01	HEAD
DATA[0]	PPM (MSB)
DATA[1]	PPM (LSB)
DATA[2]	STATUS
CS[0]	Check Sum

STATUS : 0x00 – Preheating, 0x01 – Normal Operation

To calculate the ppm, do the following.

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

SET ABC ON/OFF AND PARAMETER (CMD_SET_ABC)

This command will enable/disable ABC feature and sets parameters.

Request (I²C)

0x10	CMD
DATA[0]	RESERVED, Default 100 (0x64)
DATA[1]	ABC ON/OFF (0 : ON, 2 : OFF)
DATA[2]	ABC Duration
DATA[3]	ABC PPM (MSB)
DATA[4]	ABC PPM (LSB)
DATA[5]	RESERVED, Default 100 (0x64)

Response (I²C)

0x10	CMD
DATA[0]	RESERVED, Default 100 (0x64)
DATA[1]	ABC ON/OFF (0 : ON, 2 : OFF)
DATA[2]	ABC Duration
DATA[3]	ABC PPM (MSB)
DATA[4]	ABC PPM (LSB)
DATA[5]	RESERVED, Default 100 (0x64)
CS[0]	Check Sum

To calculate the ABC PPM, do the following.

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

MANUAL CALIBRATION (CMD_SET_MANUAL_CAL)

This command will calibrate TES0903 to 0 ~ 5000ppm^{*(1)}.

At this moment, the environment temperature should be set to room temperature and room pressure.

Request (I²C)

0x03	CMD
DATA[0]	PPM (MSB)
DATA[1]	PPM (LSB)

Response (I²C)

0x03	CMD
DATA[0]	PPM (MSB)
DATA[1]	PPM (LSB)
CS[0]	Check Sum

To calculate the PPM, do the following.

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

*(1) When performing manual calibration below 400ppm, please disable ABC function and perform manual calibration.

SERIAL NUMBER (CMD_GET_SER)

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

Request (I²C)

0x1F	CMD
------	-----

Response (I²C)

0x1F	CMD
DATA[0]	RESERVED, Default 0x00
DATA[1]	S/N MSB (MSB), S/N MSB size is 4 bytes
...	...
DATA[4]	S/N MSB (LSB)
DATA[5]	RESERVED, Default 0x00
DATA[6]	S/N LSB (MSB), S/N LSB size is 4 bytes
...	...
DATA[9]	S/N LSB (LSB)
CS[0]	Check Sum

FIRMWARE VERSION (CMD_GET_VER)

This command will return the current firmware version number.
Version number is something such as TP00.01.02

Request (I²C)

0x1E	CMD
------	-----

Response (I²C)

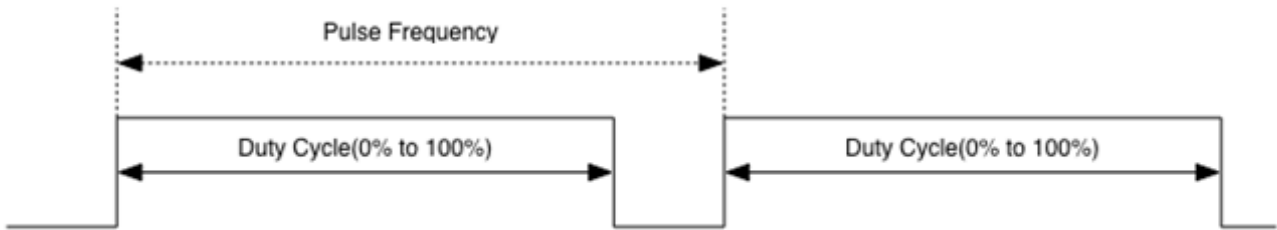
0x1E	CMD
DATA[0]	RESERVED, Default 'T' (0x54)
DATA[1]	RESERVED, Default 'P' (0x50)
DATA[2]	Major-1 (ASCII)
DATA[3]	Major-2 (ASCII)
DATA[4]	Default '.' (0x2E)
DATA[5]	Minor-1 (ASCII)
DATA[6]	Minor-2 (ASCII)
DATA[7]	Default '.' (0x2E)
DATA[8]	Build-1 (ASCII)
DATA[9]	Build-2 (ASCII)
CS[0]	Check Sum

Note : TP00.01.02

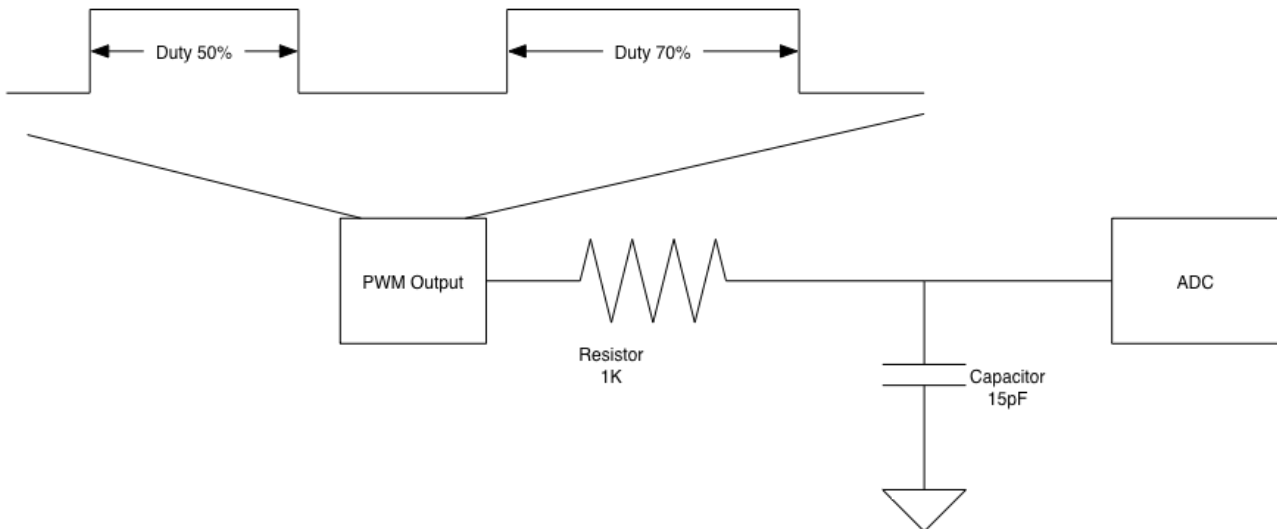
	Major-1	Major-2		Minor-1	Minor-2		Build-1	Build-2
TP	0x30	0x30	0x2E	0x30	0x31	0x2E	0x30	0x32

3. PWM Operation

PWM(Pulse Width Modulation) signal is a digital signal which is equivalent to analog output in power.



A general ADC is able to read PWM output through a LPF. The circuit shows generic first order low pass filter for converting PWM to analog signal to read it through an analog to digital converter.

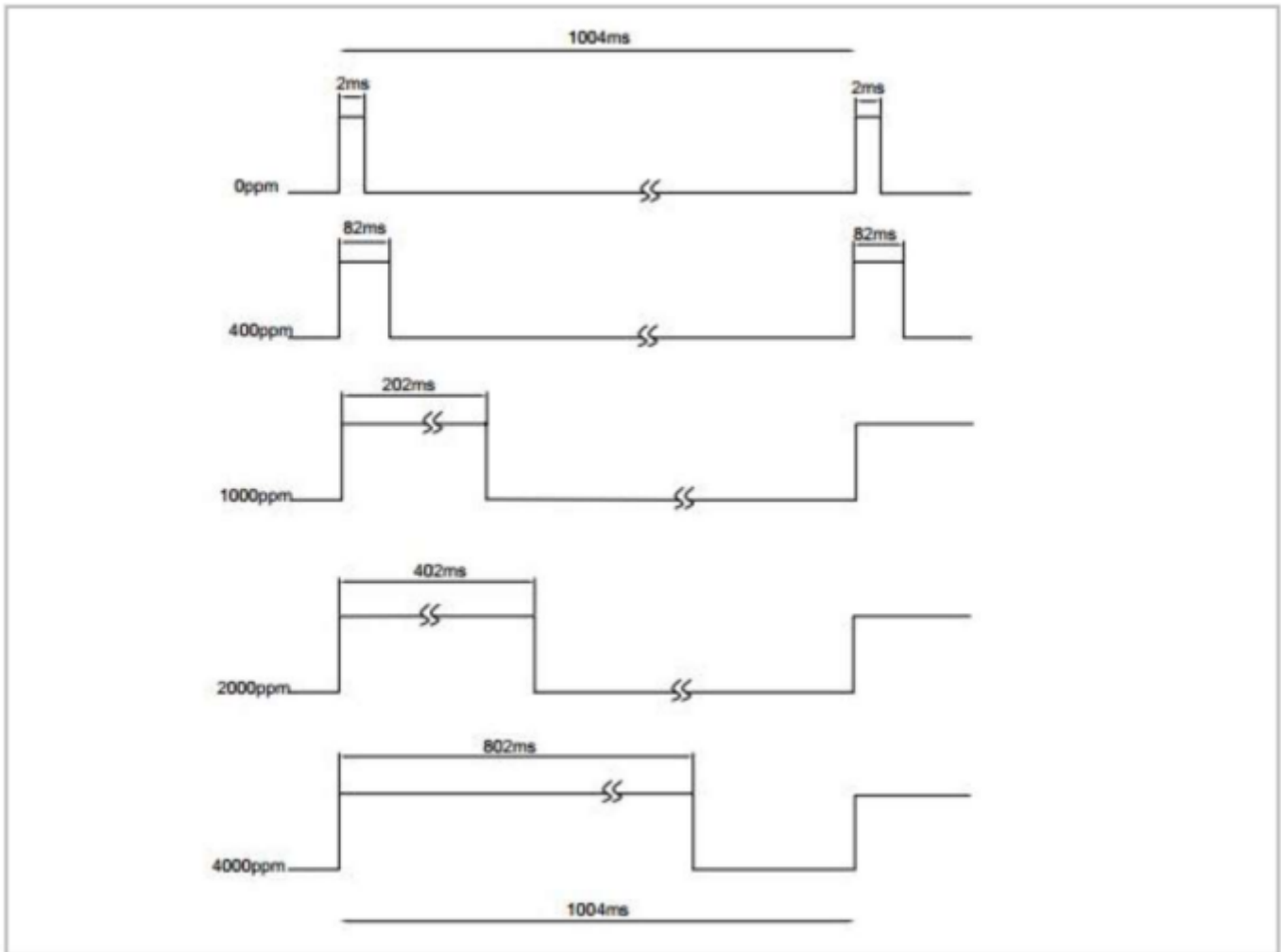


Simplified Low Pass Filter for PWM signal input to an ADC

TES0903 PWM output will be proportional to a 400 to 5000 ppm.
 The user can measure the duration of PWM pulse.

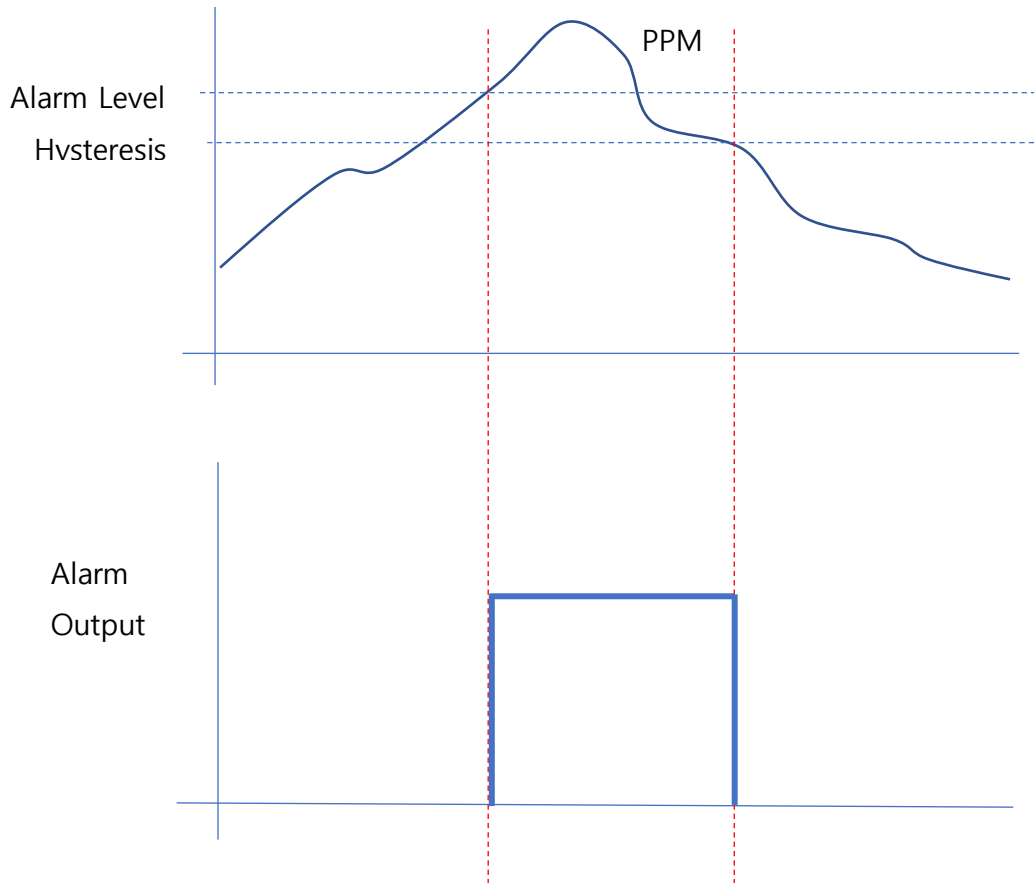
TOTAL PERIOD	1004.0 ms
OUTPUT range	400 to 5000 ppm ^{*(1)}
CO2 Level	$C_{ppm} = (PPM/5) + 2ms$

*(1) 400 ppm is the minimum value when ABC is on.



ALARM Operation

An alarm signal output is controlled by setting register addresses. This register stores CO2 level to be alarmed and its default level is 2,000 ppm. Host can change its ALARM level via command and can enabled/disable of function. It has hysteresis area when there is alarmed.



4. ABC feature

TES0903 supports ABC (Automatic Background Calibration) feature.

TES0903 monitors the trends and min/max value of CO2 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 "7" and its range is "1 ~ 30".

Please refer to the [ABC_DURATION](#).

5. Manual Calibration feature

TES0903 supports Manual Calibration (MC).

Due to shock during the transportation or incorrect installation, TES0903 can drift to unrelated ambient concentrations.

Manual Calibration compensates the drift to 400ppm immediately.

For accuracy compensation, TES0903 should be supplied power at least 3 minutes in a well-ventilated and stable environment prior to calibration. Also, it should be done at room temperature.

MC updates calibration parameters permanently unlike ABC.

If the user wants to calibrate the sensor without using ABC, MC is the best way.

User can manually calibrate the sensor to 400ppm via the "0x4C" command set.

Please refer to the [MANUAL_CAL](#).