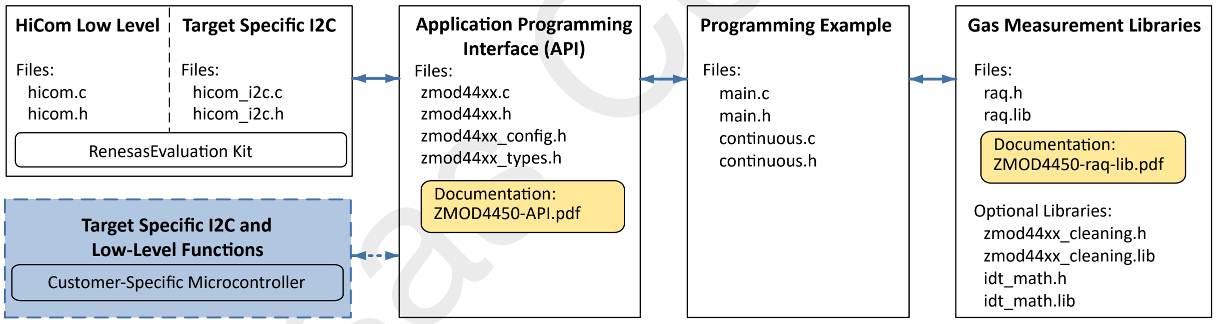
ZMOD4450 API Documentation

Contents

Chapter 1

ZMOD4450 Application Programming Interface Overview

This document refers to the IDT document *ZMOD4450 Programming Manual - Read Me*. The figure below shows an overview of the ZMOD4450 API, programming example and libraries. Custom microcontrollers can be used to establish I2C communication. Using the user's own microcontroller requires implementing the user's own target-specific I2C and low-level functions (highlighted in light blue). The following describes in detail the Application Programming Interface (API) of the ZMOD4450.



Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Module Documentation

5.1 Gas sensor IDs

Macros

- #define **ZMOD4450_PID** (0x7310)

5.1.1 Detailed Description

The gas sensor product IDs.

5.2 Error codes

Macros

- #define ERROR_INIT_OUT_OF_RANGE (1)
- #define ERROR_GAS_TIMEOUT (2)
- #define ERROR_I2C (3)
- #define ERROR_SENSOR_UNSUPPORTED (4)
- #define ERROR_CONFIG_MISSING (5)
- #define ERROR_SENSOR (6)
- #define ERROR_ACCESS_CONFLICT (7)
- #define ERROR_POR_EVENT (8)

5.2.1 Detailed Description

The gas sensor and API error codes.

5.2.2 Macro Definition Documentation

5.2.2.1 #define ERROR_ACCESS_CONFLICT (7)

AccessConflict.

5.2.2.2 #define ERROR_CONFIG_MISSING (5)

There is no pointer to a valid configuration.

5.2.2.3 #define ERROR_GAS_TIMEOUT (2)

The operation took too long.

5.2.2.4 #define ERROR_I2C (3)

Failure in i2c communication.

5.2.2.5 #define ERROR_INIT_OUT_OF_RANGE (1)

The initialize value is out of range.

5.2.2.6 #define ERROR_POR_EVENT (8)

POR_event.

5.2.2.7 #define ERROR_SENSOR (6)

Sensor malfunction.

5.2.2.8 #define ERROR_SENSOR_UNSUPPORTED (4)

Sensor is not supported with this firmware.

Chapter 6

Data Structure Documentation

6.1 zmod44xx_conf Struct Reference

Structure to hold the gas sensor module configuration.

```
#include <zmod44xx_types.h>
```

Data Fields

- uint8_t **start**
- zmod44xx_conf_str **h**
- zmod44xx_conf_str **d**
- zmod44xx_conf_str **m**
- zmod44xx_conf_str **s**
- zmod44xx_conf_str **r**

6.1.1 Detailed Description

Structure to hold the gas sensor module configuration.

The documentation for this struct was generated from the following file:

- zmod44xx_types.h

6.2 zmod44xx_conf_str Struct Reference

A single data set for the configuration.

```
#include <zmod44xx_types.h>
```

Data Fields

- uint8_t **addr**
- uint8_t **len**
- const uint8_t ∗ **data**

6.2.1 Detailed Description

A single data set for the configuration.

The documentation for this struct was generated from the following file:

- zmod44xx_types.h

6.3 zmod44xx_dev_t Struct Reference

Device structure ZMOD44xx.

```
#include <zmod44xx_types.h>
```

Data Fields

- uint8_t i2c_addr
- uint8_t config [6]
- uint8_t prod_data [5]
- uint16_t mox_lr
- uint16_t mox_er
- uint16_t pid
- zmod44xx_i2c_ptr_t read
- zmod44xx_i2c_ptr_t write
- zmod44xx_delay_ptr_p delay_ms
- const zmod44xx_conf ∗ init_conf
- const zmod44xx_conf ∗ meas_conf

6.3.1 Detailed Description

Device structure ZMOD44xx.

6.3.2 Field Documentation

6.3.2.1 uint8_t config[6]

configuration parameter set

6.3.2.2 zmod44xx_delay_ptr_p delay_ms

function pointer to delay function

6.3.2.3 uint8_t i2c_addr

i2c address of the sensor

6.3.2.4 const zmod44xx_conf∗ init_conf

pointer to the initialize configuration

6.3.2.5 const zmod44xx_conf∗ meas_conf

pointer to the measurement configuration

6.3.2.6 uint16_t mox_er

sensor specific parameter

6.3.2.7 uint16_t mox_lr

sensor specific parameter

6.3.2.8 uint16_t pid

product id of the sensor

6.3.2.9 uint8_t prod_data[5]

production data

6.3.2.10 zmod44xx_i2c_ptr_t read

function pointer to i2c read

6.3.2.11 zmod44xx_i2c_ptr_t write

function pointer to i2c write

The documentation for this struct was generated from the following file:

- zmod44xx_types.h

Chapter 7

File Documentation

7.1 zmod44xx.c File Reference

ZMOD44xx functions.

```
#include "zmod44xx.h"
#include "zmod44xx_config.h"
```

Functions

- int8_t zmod44xx_read_sensor_info (zmod44xx_dev_t ∗dev)
  Read sensor parameter.
- int8_t zmod44xx_init_sensor (zmod44xx_dev_t ∗dev)
  Initialize the sensor after power on.
- int8_t zmod44xx_init_measurement (zmod44xx_dev_t ∗dev)
  Initialize the sensor for zmod4450 measurement.
- int8_t zmod44xx_start_measurement (zmod44xx_dev_t ∗dev)
  Start the measurement.
- int8_t zmod44xx_read_status (zmod44xx_dev_t ∗dev, uint8_t ∗status)
  Read the status of the device.
- int8_t zmod44xx_read_rmox (zmod44xx_dev_t ∗dev, float ∗rmox)
  Read adc values from sensor and calculate RMOX.

7.1.1 Detailed Description

ZMOD44xx functions.

Author

IDT

7.1.2 Function Documentation

7.1.2.1 int8_t zmod44xx_init_measurement (zmod44xx_dev_t ∗ *dev*)

Initialize the sensor for zmod4450 measurement.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.1.2.2 int8_t zmod44xx_init_sensor (zmod44xx_dev_t ∗ *dev*)

Initialize the sensor after power on.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.1.2.3 int8_t zmod44xx_read_rmox (zmod44xx_dev_t ∗ *dev*, float ∗ *rmox*)

Read adc values from sensor and calculate RMOX.

Parameters

`in`	*dev*	pointer to the device
`in,out`	*rmox*	pointer to the resulting Rmox value

Returns

error code

Return values

0	success
!= 0	error

7.1.2.4 int8_t zmod44xx_read_sensor_info (zmod44xx_dev_t ∗ *dev*)

Read sensor parameter.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

Note

This function must be called once before running other sensor functions.

7.1.2.5 int8_t zmod44xx_read_status (zmod44xx_dev_t ∗ *dev,* uint8_t ∗ *status*)

Read the status of the device.

Parameters

`in`	*dev*	pointer to the device
`in,out`	*status*	pointer to the status variable

Returns

error code

Return values

0	success
!= 0	error

7.1.2.6 int8_t zmod44xx_start_measurement (zmod44xx_dev_t * *dev*)

Start the measurement.

Parameters

in	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.2 zmod44xx.h File Reference

ZMOD44xx functions.

```
#include "zmod44xx_types.h"
```

Macros

- #define **ZMOD4450_I2C_ADDRESS** (0x32)
- #define **ZMOD44XX_ADDR_PID** (0x00)
- #define **ZMOD44XX_ADDR_CONF** (0x20)
- #define **ZMOD44XX_ADDR_PROD_DATA** (0x26)
- #define **ZMOD44XX_ADDR_CMD** (0x93)
- #define **ZMOD44XX_ADDR_STATUS** (0x94)
- #define **ZMOD44XX_LEN_PID** (2)
- #define **ZMOD44XX_LEN_CONF** (6)
- #define **ZMOD44XX_LEN_PROD_DATA** (5)
- #define STATUS_SEQUENCER_RUNNING_MASK (0x80)
- #define STATUS_SLEEP_TIMER_ENABLED_MASK (0x40)
- #define STATUS_ALARM_MASK (0x20)
- #define STATUS_LAST_SEQ_STEP_MASK (0x1F)
- #define STATUS_POR_EVENT_MASK (0x80)
- #define STATUS_ACCESS_CONFLICT_MASK (0x40)

Functions

- int8_t zmod44xx_read_sensor_info (zmod44xx_dev_t ∗dev)

  Read sensor parameter.
- int8_t zmod44xx_init_sensor (zmod44xx_dev_t ∗dev)

  Initialize the sensor after power on.
- int8_t zmod44xx_init_measurement (zmod44xx_dev_t ∗dev)

  Initialize the sensor for zmod4450 measurement.
- int8_t zmod44xx_start_measurement (zmod44xx_dev_t ∗dev)

  Start the measurement.
- int8_t zmod44xx_read_status (zmod44xx_dev_t ∗dev, uint8_t ∗status)

  Read the status of the device.
- int8_t zmod44xx_read_rmox (zmod44xx_dev_t ∗dev, float ∗rmox)

  Read adc values from sensor and calculate RMOX.

7.2.1 Detailed Description

ZMOD44xx functions.

Author

IDT

7.2.2 Macro Definition Documentation

7.2.2.1 #define STATUS_ACCESS_CONFLICT_MASK (0x40)

AccessConflict

7.2.2.2 #define STATUS_ALARM_MASK (0x20)

Alarm

7.2.2.3 #define STATUS_LAST_SEQ_STEP_MASK (0x1F)

Last executed sequencer step

7.2.2.4 #define STATUS_POR_EVENT_MASK (0x80)

POR_event

7.2.2.5 #define STATUS_SEQUENCER_RUNNING_MASK (0x80)

Sequencer is running

7.2.2.6 #define STATUS_SLEEP_TIMER_ENABLED_MASK (0x40)

SleepTimer_enabled

7.2.3 Function Documentation

7.2.3.1 int8_t zmod44xx_init_measurement (zmod44xx_dev_t * *dev*)

Initialize the sensor for zmod4450 measurement.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.2.3.2 int8_t zmod44xx_init_sensor (zmod44xx_dev_t * *dev*)

Initialize the sensor after power on.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.2.3.3 int8_t zmod44xx_read_rmox (**zmod44xx_dev_t** ∗ *dev,* float ∗ *rmox*)

Read adc values from sensor and calculate RMOX.

Parameters

`in`	*dev*	pointer to the device
`in,out`	*rmox*	pointer to the resulting Rmox value

Returns

error code

Return values

0	success
!= 0	error

7.2.3.4 int8_t zmod44xx_read_sensor_info (**zmod44xx_dev_t** ∗ *dev*)

Read sensor parameter.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

Note

This function must be called once before running other sensor functions.

7.2.3.5 int8_t zmod44xx_read_status (**zmod44xx_dev_t** ∗ *dev,* uint8_t ∗ *status*)

Read the status of the device.

Parameters

`in`	*dev*	pointer to the device
`in,out`	*status*	pointer to the status variable

Returns

error code

Return values

0	success
!= 0	error

7.2.3.6 int8_t zmod44xx_start_measurement (zmod44xx_dev_t ∗ *dev*)

Start the measurement.

Parameters

`in`	*dev*	pointer to the device

Returns

error code

Return values

0	success
!= 0	error

7.3 zmod44xx_config.h File Reference

ZMOD44xx configuration.

```
#include <stdint.h>
#include "zmod44xx_types.h"
```

Variables

- const uint8_t **data_set_4450_continuous** [ ]

- const uint8_t **data_set_4450i** [ ]
- const zmod44xx_conf zmod4450_continuous

  ZMOD4450 configuration for continuous mode.
- const zmod44xx_conf zmod44xxi

  ZMOD44XX sensor initialization configuration.

7.3.1 Detailed Description

ZMOD44xx configuration.

Author

IDT

7.3.2 Variable Documentation

7.3.2.1 const uint8_t data_set_4450_continuous[ ]

Initial value:

```
= {0x20, 0x04, 0x40, 0x09,
                                          0x03,
                                          0x00, 0x00, 0x80, 0x08}
```

7.3.2.2 const uint8_t data_set_4450i[ ]

Initial value:

```
= {0x00, 0x28, 0xC3, 0xE3,
                                   0x00, 0x00, 0x80, 0x40}
```

7.3.2.3 const zmod44xx_conf zmod4450_continuous

Initial value:

```
= {
        .start = 0xC0,
        .h = {.addr = 0x40, .len = 2},
        .d = {.addr = 0x50, .len = 4, .data = &data_set_4450_continuous[0]},
        .m = {.addr = 0x60, .len = 1, .data = &data_set_4450_continuous[4]},
        .s = {.addr = 0x68, .len = 4, .data = &data_set_4450_continuous[5]},
        .r = {.addr = 0x99, .len = 2}

}
```

ZMOD4450 configuration for continuous mode.

7.3.2.4 const zmod44xx_conf zmod44xxi

Initial value:

```
= {
        .start = 0x80,
        .h = {.addr = 0x40, .len = 2},
        .d = {.addr = 0x50, .len = 2, .data = &data_set_4450i[0]},
        .m = {.addr = 0x60, .len = 2, .data = &data_set_4450i[2]},
        .s = {.addr = 0x68, .len = 4, .data = &data_set_4450i[4]},
        .r = {.addr = 0x97, .len = 4}

}
```

ZMOD44XX sensor initialization configuration.

7.4 zmod44xx_types.h File Reference

ZMOD44xx types.

```
#include <stdint.h>
```

Data Structures

- struct zmod44xx_conf_str

  A single data set for the configuration.
- struct zmod44xx_conf

  Structure to hold the gas sensor module configuration.
- struct zmod44xx_dev_t

  Device structure ZMOD44xx.

Macros

- #define **ZMOD4450_PID** (0x7310)
- #define ZMOD44XX_OK (0)
- #define ERROR_INIT_OUT_OF_RANGE (1)
- #define ERROR_GAS_TIMEOUT (2)
- #define ERROR_I2C (3)
- #define ERROR_SENSOR_UNSUPPORTED (4)
- #define ERROR_CONFIG_MISSING (5)
- #define ERROR_SENSOR (6)
- #define ERROR_ACCESS_CONFLICT (7)
- #define ERROR_POR_EVENT (8)

Typedefs

- typedef int8_t(∗ zmod44xx_i2c_ptr_t) (uint8_t addr, uint8_t reg_addr, const uint8_t ∗data, uint8_t len)

  function pointer type for i2c access
- typedef void(∗ zmod44xx_delay_ptr_p) (uint32_t ms)

  function pointer to hardware dependent delay function

7.4.1 Detailed Description

ZMOD44xx types.

Author

IDT

7.4.2 Macro Definition Documentation

7.4.2.1 #define ZMOD44XX_OK (0)

Return value if no fault has been found.

7.4.3 Typedef Documentation

7.4.3.1 typedef int8_t(∗ zmod44xx_i2c_ptr_t) (uint8_t addr, uint8_t reg_addr, const uint8_t ∗data, uint8_t len)

function pointer type for i2c access

Parameters

`in`	*addr*	7-bit I2C slave address of the ZMOD44xx
`in`	*reg_addr*	address of internal register to read/write
`in,out`	*data*	pointer to the read/write data value
`in`	*len*	number of bytes to read/write

Returns

error code

Return values

0	success
!= 0	error