AS7050 ChipLib

API Documentation
revision v1.2.0

Generated by Doxygen

Contents

1 AS7050 ChipLib

1.1 Introduction

This document provides an overview on how the vital signs sensor family AS7050 can be controlled through the ChipLib (Chip Library) and the corresponding OSAL (operating system abstraction layer).

The ChipLib's API allows for usage in several different fields of application. It is programmed in standard C language.

Key features:

- Standardized measurement routine via callback handler
- Independent of hardware and platform
- Tested code source by ams

Limitations:

- Sensor calibration topics are not part of this library.
- The library provides only raw values.

The documentation is split into three sections:

- ChipLib: This describes how an application should handle the main library.
- OSAL: This section describes the adaptation of the ChipLib to other platforms.
- Module: Interface description of both APIs: ChipLib and OSAL

1.2 Overview

The ChipLib can directly be called by the user application and can be used without adaption. Hardware and platform dependencies must be specified separately in the OSAL. The OSAL interface is kept simple to keep integration efforts for the customers to a minimum. More information on the API functions can be found here: OSAL Functions. Details on how to implement application specific OSAL are described in OSAL

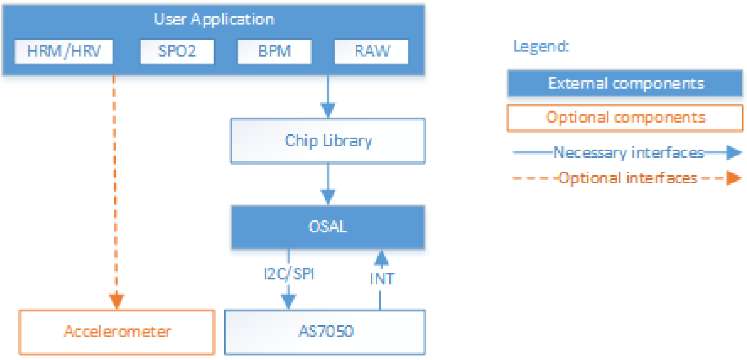

Figure 1 Structure Overview

- Accelerometer: Separate component which is necessary to support HRM under motion
- AS7050: Real sensor device
- OSAL: Provides interface to Chiplab for hardware adaption
- Chip Library: Control logic for the sensor
- User Application: Custom applications to calculate vital signs data like HRM, SPO2 etc.

1.3 Acronyms and Abbreviations

1.3.1 Acronyms

API = Application Programming Interface
BPM = Blood Pressure Monitor
HRM = Heart rate monitor
I2C = Inter-Integrated Circuit
OSAL = Operating System Abstraction Layer
SPI = Serial Peripheral Interface
SPO2 = Oxygen Saturation

1.3.2 Abbreviations

ChipLib = Chip Library for higher level communication with the sensor

2 ChipLib

This page describes the work with the ChipLib in more detail.

2.1 Block diagram

This block diagram shows you all the provided functions of the library. Three groups of functions are available:

- initialize/shutdown: starts and stops the work with the ChipLib
- configuration: allows the user to configure the ChipLib.
- measurement: this functions handle the real measurement

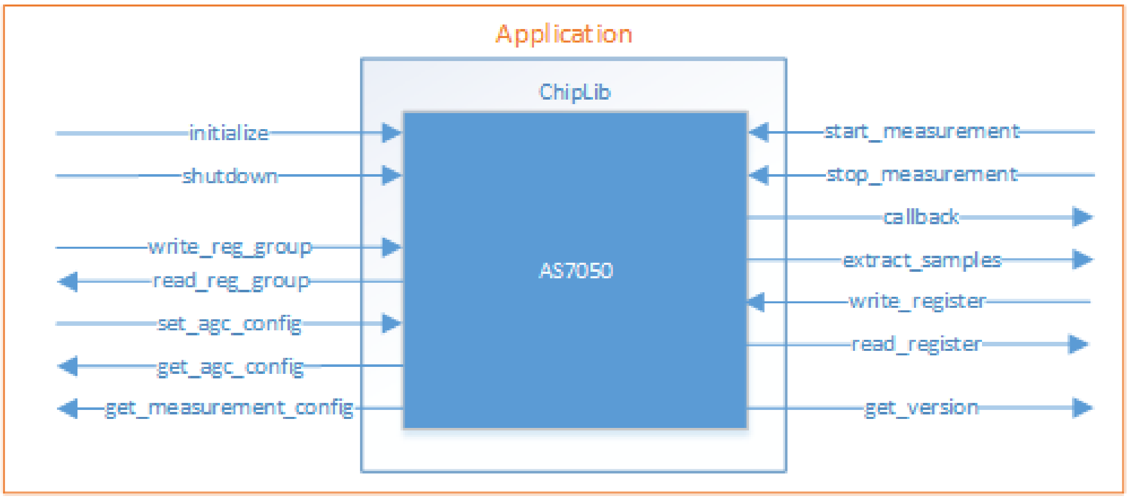

Figure 2 ChipLib API

The complete functions and their arguments will be described in section ChipLib Functions.

3 OSAL

The following pages describe the OSAL (operating system abstraction layer) of the ChipLib. This is used to encapsulate platform and operating system specific function to a separate layer, where everybody can customize this interface to his requirements.

This interface is implemented as easy as possible. No complex data types are used and dependencies between functions are small as possible. An overview of the OSAL is shown in the next section.

3.1 Block diagram

The following figure describes the dependencies to the ChipLib. Inside the OSAL, functions like I2C/SPI and Interrupt handling must be implemented.

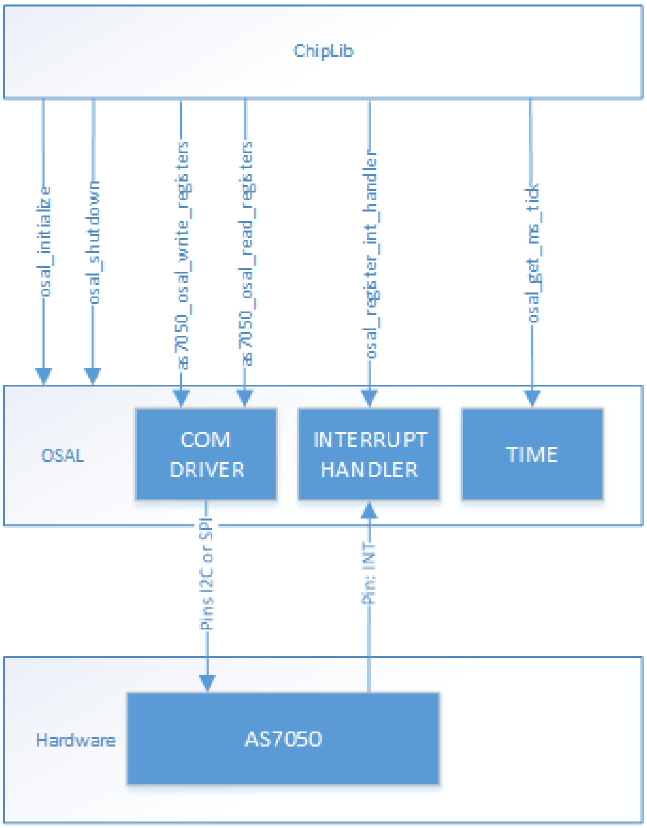

Figure 3 Overview OSAL

Function blocks:

- COM DRIVER: Is used to communicate with the sensor. The definition of the transfer function allows I2C or SPI.
- INTERRUPT HANDLER: This component handles the interrupt pin and fires an event on falling edge.
- TIME: This module is used to provide a time tick in milliseconds.

3.2 Function dependencies ChipLib - OSAL

Follwing table describes, which OSAL function is called by which ChipLib function:

Table 1 Function dependencies ChipLib - OSAL

OSAL functions	ChipLib functions
as7050_osal_initialize	as7050_initialize
as7050_osal_shutdown	as7050_shutdown
as7050_osal_write_registers, as7050_osal_read_registers	as7050_initialize, as7050_shutdown, as7050_write_register, as7050_read_register, as7050_start_measurement, as7050_stop_measurement, as7050_get_measurement_config, as7050_set_reg_group, as7050_get_reg_group

OSAL functions	ChipLib functions
as7050_osal_register_int_handler	as7050_initialize
as7050_osal_get_ms_tick	as7050_callback_t

3.3 Interfaces

The complete functions and their arguments will be described in section OSAL Functions.

4 Module Index

4.1 Modules

Here is a list of all modules:

5 Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

6 Module Documentation

6.1 ChipLib Functions

This is the chip library for ams vital signs chip AS7050.

Functions

- err_code_t CHIPLIB_DECLDIR as7050_initialize (const as7050_callback_t p_callback, const void ∗p_cb_←param, const char ∗p_interface_descr)

  Initializes the library and the device.
- err_code_t CHIPLIB_DECLDIR as7050_shutdown (void)

  Stops all internal actions and power down the device.
- err_code_t CHIPLIB_DECLDIR as7050_set_reg_group (enum as7050_reg_group_ids id, const uint8_t ∗p←_data, uint8_t size)

  Write a register group.
- err_code_t CHIPLIB_DECLDIR as7050_get_reg_group (enum as7050_reg_group_ids id, uint8_t ∗p_data, uint8_t ∗p_size)

  Read a register group.
- err_code_t CHIPLIB_DECLDIR as7050_write_register (uint8_t reg_addr, uint8_t reg_val)

  Write register.
- err_code_t CHIPLIB_DECLDIR as7050_read_register (uint8_t reg_addr, uint8_t ∗p_reg_val)

  Read register.
- err_code_t CHIPLIB_DECLDIR as7050_get_measurement_config (as7050_meas_config_t ∗p_meas_←config)

  Read the actual measurement configuration, which was set by the register groups.
- err_code_t CHIPLIB_DECLDIR as7050_set_agc_config (const as7050_agc_config_t ∗p_agc_config)

  Sets the configuration for auto-gain-control (AGC)
- err_code_t CHIPLIB_DECLDIR as7050_get_agc_config (as7050_agc_config_t ∗p_agc_config)

  Gets the configuration for auto-gain-control (AGC)
- err_code_t CHIPLIB_DECLDIR as7050_extract_samples (as7050_channel_flags_t chan, const uint8_t ∗p←_fifo_data, uint16_t fifo_data_num, uint32_t ∗p_chan_data, uint16_t ∗p_chan_data_num)

  Extract the requested samples from FIFO data, which was received by the as7050_callback_t.
- err_code_t CHIPLIB_DECLDIR as7050_start_measurement (void)

  Starts a measurement.
- err_code_t CHIPLIB_DECLDIR as7050_stop_measurement (void)

  Stops a measurement.
- err_code_t CHIPLIB_DECLDIR as7050_get_version (as7050_version_t ∗p_version)

  Requests the version information.
- err_code_t CHIPLIB_DECLDIR as7050_calculate_dac_reference_value (dac_ref_control_t control, uint16←_t average_cnt, dac_ref_status_t ∗p_status, uint32_t ∗p_dac_ref)

  Calculates the DAC reference value which is used for GSR measurement.

6.1.1 Detailed Description

This is the chip library for ams vital signs chip AS7050.

6.1.2 Function Documentation

6.1.2.1 as7050_initialize()

```
err_code_t CHIPLIB_DECLDIR as7050_initialize (
        const as7050_callback_t p_callback,
        const void * p_cb_param,
        const char * p_interface_descr )
```

Initializes the library and the device.

Following tasks will be done here:

- Initialize hardware abstraction layer.
- Reset chip.

Note

This function must be called at first, otherwise all other functions return with error code.

Parameters

in	*p_callback*	Pointer to the callback function, see as7050_callback_t
in	*p_cb_param*	Optional pointer to an application parameter, which will be transmitted with every callback.
in	*p_interface_descr*	Chiplib forwards this interface description to as7050_osal_initialize.

Return values

ERR_SUCCESS	Function returns without error.
ERR_ARGUMENT	An argument is invalid.
ERR_IDENTIFICATION	The specified sensor was not found.
ERR_DATA_TRANSFER	Communication error to sensor.

6.1.2.2 as7050_shutdown()

```
err_code_t CHIPLIB_DECLDIR as7050_shutdown (
        void )
```

Stops all internal actions and power down the device.

Following tasks will be done here:

- Stops measurement, if running.
- Power down the sensor device.
- Shutdown the hardware abstraction layer.
- Block calling of all other functions, but initialize.

Return values

ERR_SUCCESS	Function returns without error.
ERR_DATA_TRANSFER	Communication error to sensor.

6.1.2.3 as7050_set_reg_group()

```
err_code_t CHIPLIB_DECLDIR as7050_set_reg_group (
        enum as7050_reg_group_ids id,
        const uint8_t * p_data,
        uint8_t size )
```

Write a register group.

This function configures the sensor directly via register formated structure.

Parameters

in	*id*	Identification number of an item, see as7050_reg_group_ids.
in	*p_data*	Pointer to the register data, like as7050_config_led_t
in	*size*	Sets the size of register data in bytes.

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_ARGUMENT	Register group id is not supported.
ERR_POINTER	Detected NULL pointer for data
ERR_SIZE	Size of the data buffer is wrong
ERR_DATA_TRANSFER	Communication error to sensor.

6.1.2.4 as7050_get_reg_group()

```
err_code_t CHIPLIB_DECLDIR as7050_get_reg_group (
        enum as7050_reg_group_ids id,
        uint8_t * p_data,
        uint8_t * p_size )
```

Read a register group.

Reads the actual register data into a register group structure.

Parameters

in	*id*	Identification number of an item, see as7050_reg_group_ids.
out	*p_data*	Pointer, where the data of the register group can be saved.
in,out	*p_size*	IN: Maximum buffer size, OUT: Size in byte of the register group data The maximum size is defined in AS7050_MAX_GROUP_SIZE

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_ARGUMENT	Register group id is not supported.
ERR_POINTER	Detected NULL pointer for data
ERR_SIZE	Size of the data buffer is wrong
ERR_DATA_TRANSFER	Communication error to sensor

6.1.2.5 as7050_write_register()

```
err_code_t CHIPLIB_DECLDIR as7050_write_register (
        uint8_t reg_addr,
        uint8_t reg_val )
```

Write register.

This function sets the value of a single I2C-register.

Parameters

in	*reg_addr*	Register address of the sensor
in	*reg_val*	New register value, which shall be written to the register address

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_DATA_TRANSFER	Communication error to sensor

6.1.2.6 as7050_read_register()

```
err_code_t CHIPLIB_DECLDIR as7050_read_register (
        uint8_t reg_addr,
        uint8_t * p_reg_val )
```

Read register.

This function gets the value of a single I2C-register.

Parameters

in	*reg_addr*	Register address of the sensor
out	*p_reg_val*	Actual register value, which was read from the register address

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_POINTER	Detected NULL pointer for data
ERR_DATA_TRANSFER	Communication error to sensor

6.1.2.7 as7050_get_measurement_config()

```
err_code_t CHIPLIB_DECLDIR as7050_get_measurement_←
config (
            as7050_meas_config_t * p_meas_config )
```

Read the actual measurement configuration, which was set by the register groups.

Note

> The measurement configuration can be changed after set of new register blocks or writting to single registers. Therefore, read it back after finished configuration to get the actual values.

Parameters

out	*p_meas_config*	Actual measurement configuration, See as7050_meas_config_t

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_POINTER	Detected NULL pointer for data
ERR_DATA_TRANSFER	Communication error to sensor

6.1.2.8 as7050_set_agc_config()

```
err_code_t CHIPLIB_DECLDIR as7050_set_agc_config (
            const as7050_agc_config_t * p_agc_config )
```

Sets the configuration for auto-gain-control (AGC)

Parameters

in	*p_agc_config*	Pointer to the AGC-configuration structure, see as7050_agc_config_t

Return values

ERR_SUCCESS	Function returns without error.

Return values

ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_POINTER	Detected NULL pointer for data

6.1.2.9 as7050_get_agc_config()

```
err_code_t CHIPLIB_DECLDIR as7050_get_agc_config (
        as7050_agc_config_t * p_agc_config )
```

Gets the configuration for auto-gain-control (AGC)

Parameters

out	*p_agc_config*	Pointer to the AGC-configuration structure, see as7050_agc_config_t

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_POINTER	Detected NULL pointer for data

6.1.2.10 as7050_extract_samples()

```
err_code_t CHIPLIB_DECLDIR as7050_extract_samples (
        as7050_channel_flags_t chan,
        const uint8_t * p_fifo_data,
        uint16_t fifo_data_num,
        uint32_t * p_chan_data,
        uint16_t * p_chan_data_num )
```

Extract the requested samples from FIFO data, which was received by the as7050_callback_t.

Parameters

in	*chan*	Requested channel. See as7050_channel_flags_t.
in	*p_fifo_data*	FIFO data
in	*fifo_data_num*	Number of FIFO elements
out	*p_chan_data*	Pointer to buffer, where the channel data can be saved
in,out	*p_chan_data_num*	Input: Number of provided chan_data elements, Output: Number of used chan_data elements

Return values

ERR_SUCCESS	Function returns without error.

Return values

ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_ARGUMENT	adc_map or req_adc is wrong
ERR_POINTER	Detected NULL pointer for data
ERR_CONFIG	FIFO is not configured

6.1.2.11 as7050_start_measurement()

```
err_code_t CHIPLIB_DECLDIR as7050_start_measurement (
        void  )
```

Starts a measurement.

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_DATA_TRANSFER	Communication error to sensor.
ERR_CONFIG	PPG or ECG are not enabled, AOC and AGC are enabled in parallel

6.1.2.12 as7050_stop_measurement()

```
err_code_t CHIPLIB_DECLDIR as7050_stop_measurement (
        void  )
```

Stops a measurement.

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked, call as7050_initialize at first.
ERR_DATA_TRANSFER	Communication error to sensor.

6.1.2.13 as7050_get_version()

```
err_code_t CHIPLIB_DECLDIR as7050_get_version (
        as7050_version_t * p_version )
```

Requests the version information.

Parameters

out	*p_version*	Pointer to memory, where the version information can be saved. See as7050_version_t

Return values

ERR_SUCCESS	Function returns without error.
ERR_POINTER	Detected NULL pointer for data

6.1.2.14 as7050_calculate_dac_reference_value()

```
err_code_t CHIPLIB_DECLDIR as7050_calculate_↩
dac_reference_value (
            dac_ref_control_t control,
            uint16_t average_cnt,
            dac_ref_status_t * p_status,
            uint32_t * p_dac_ref )
```

Calculates the DAC reference value which is used for GSR measurement.

Attention

> This function is only needed for GSR measurement and uses the Chip Library in a special mode because multiple measurements with different configurations are performed.

Internally, DAC 0 and 1 are measured multiple times and are used to calculate the DAC reference value. For both DAC 0 and 1 a separate measurement with a different configuration is performed. Depending on sample rate and the number of samples to be measured per DAC, this calculation can take more than one second. This function does not block while the calculation is in progress, but needs to be called repeatedly until DAC_REF_STATUS_RUNNING is no longer returned via the p_status argument.

Requirements:

- ECG channel must be activated. PPG channels must be disabled.
- GSR registers must be configured.

Parameters

in	*control*	Control byte. See ::dac_ref_controls
in	*average_cnt*	Minimum number of measured samples per DAC
out	*p_status*	Pointer to variable to which the current calculation status is written. See dac_ref_status
out	*p_dac_ref*	Pointer to variable to which the resulting DAC reference value is written.

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Access to the library is blocked.
ERR_ARGUMENT	Provided input parameters are invalid.
ERR_POINTER	NULL pointer detected.
ERR_DATA_TRANSFER	Communication error to sensor.

6.2 OSAL Functions

This is the abstraction layer for the chip library.

Typedefs

- typedef err_code_t(∗ as7050_osal_interrupt_t) (void)
  Callback function, which will be called if a new interrupt notification is available.

Functions

- err_code_t as7050_osal_initialize (const char ∗p_interface_desc)
  Initialization of the hardware abstraction layer.
- err_code_t as7050_osal_shutdown (void)
  Shutdown of the hardware abstraction layer.
- err_code_t as7050_osal_write_registers (uint8_t address, uint8_t number, uint8_t ∗p_values)
  Set register values inside the chip.
- err_code_t as7050_osal_read_registers (uint8_t address, uint8_t number, uint8_t ∗p_values)
  Get register values from the chip.
- err_code_t as7050_osal_register_int_handler (as7050_osal_interrupt_t callback_function)
  Registers and unregisters an interrupt handler.
- err_code_t as7050_osal_get_ms_tick (uint32_t ∗p_tick_ms)
  Requests an actual time tick (milliseconds) of the system.

6.2.1 Detailed Description

This is the abstraction layer for the chip library.

The functions has dependencies to the operatation system. So the function must be implemented application specific. Following function groups must be implemented:

- initialize/shutdown
- I2C-transfers
- Interrupt-handler

6.2.2 Typedef Documentation

6.2.2.1 as7050_osal_interrupt_t `typedef err_code_t(∗ as7050_osal_interrupt_t) (void)`

Callback function, which will be called if a new interrupt notification is available.

This callback type will be registered via the function as7050_osal_register_int_handler.

6.2.3 Function Documentation

6.2.3.1 as7050_osal_initialize()

```
err_code_t as7050_osal_initialize (
        const char * p_interface_desc )
```

Initialization of the hardware abstraction layer.

- Initialization of global parameters.
- Activates the ENABLE-pin
- Open the interface to the sensor.

Note

This function must be called at first!

Parameters

in	*p_interface_desc*	Can be used to transfer special initialization data like interface description.

Return values

ERR_SUCCESS	Function returns without error.
ERR_ARGUMENT	Argument-content is not supported
ERR_COM_INTERFACE	The interface to the sensor is faulty.

6.2.3.2 as7050_osal_shutdown()

```
err_code_t as7050_osal_shutdown (
        void )
```

Shutdown of the hardware abstraction layer.

- Closes the interface to the sensor
- Deactivates the ENABLE-Pin

Note

This function must be called for cleanup

Return values

ERR_SUCCESS	Function returns without error.
ERR_COM_INTERFACE	The interface to the sensor is faulty

6.2.3.3 as7050_osal_write_registers()

```
err_code_t as7050_osal_write_registers (
        uint8_t address,
        uint8_t number,
        uint8_t * p_values )
```

Set register values inside the chip.

Parameters

in	*address*	Register address of the chip
in	*number*	Number of register values
in	*p_values*	Pointer to register values

Return values

ERR_SUCCESS	Function returns without error.
ERR_POINTER	If buffer size is unequal 0 and buffer address is zero.
ERR_DATA_TRANSFER	Data transfer error, like bus error or timeout.
ERR_PERMISSION	Library was not initialized by as7050_osal_initialize

6.2.3.4 as7050_osal_read_registers()

```
err_code_t as7050_osal_read_registers (
        uint8_t address,
        uint8_t number,
        uint8_t * p_values )
```

Get register values from the chip.

Parameters

in	*address*	Register address of the chip
in	*number*	Number of register values
out	*p_values*	Pointer to memory, where register values can be saved

Return values

ERR_SUCCESS	Function returns without error.

Return values

ERR_POINTER	If buffer size is unequal 0 and buffer address is zero.
ERR_DATA_TRANSFER	Data transfer error, like bus error or timeout.
ERR_PERMISSION	Library was not initialized by as7050_osal_initialize

6.2.3.5 as7050_osal_register_int_handler()

```
err_code_t as7050_osal_register_int_handler (
        as7050_osal_interrupt_t callback_function )
```

Registers and unregisters an interrupt handler.

If the function pointer is null, then the old callback will be unregistered.

Parameters

in	*callback_function*	Pointer to callback function or NULL

Return values

ERR_SUCCESS	Function returns without error.
ERR_PERMISSION	Library was not initialized by as7050_osal_initialize

6.2.3.6 as7050_osal_get_ms_tick()

```
err_code_t as7050_osal_get_ms_tick (
        uint32_t * p_tick_ms )
```

Requests an actual time tick (milliseconds) of the system.

Parameters

out	*p_tick_ms*	Pointer to memory, where the actual time tick in milliseconds can be saved

Return values

ERR_SUCCESS	Function returns without error.
ERR_POINTER	If the argument is zero.
ERR_PERMISSION	Library was not initialized by as7050_osal_initialize

6.3 Error Codes

Typedefs

- typedef enum error_codes err_code_t

Enumerations

- enum error_codes {
ERR_SUCCESS = 0,
ERR_PERMISSION = 1,
ERR_MESSAGE = 2,
ERR_MESSAGE_SIZE = 3,
ERR_POINTER = 4,
ERR_ACCESS = 5,
ERR_ARGUMENT = 6,
ERR_SIZE = 7,
ERR_NOT_SUPPORTED = 8,
ERR_TIMEOUT = 9,
ERR_CHECKSUM = 10,
ERR_OVERFLOW = 11,
ERR_EVENT = 12,
ERR_INTERRUPT = 13,
ERR_TIMER_ACCESS = 14,
ERR_LED_ACCESS = 15,
ERR_TEMP_SENSOR_ACCESS = 16,
ERR_DATA_TRANSFER = 17,
ERR_FIFO = 18,
ERR_OVER_TEMP = 19,
ERR_IDENTIFICATION = 20,
ERR_COM_INTERFACE = 21,
ERR_SYNCHRONISATION = 22,
ERR_PROTOCOL = 23,
ERR_MEMORY = 24,
ERR_THREAD = 25,
ERR_SPI = 26,
ERR_DAC_ACCESS = 27,
ERR_I2C = 28,
ERR_NO_DATA = 29,
ERR_SYSTEM_CONFIG = 30,
ERR_USB_ACCESS = 31,
ERR_ADC_ACCESS = 32,
ERR_SENSOR_CONFIG = 33,
ERR_SATURATION = 34,
ERR_MUTEX = 35,
ERR_ACCELEROMETER = 36,
ERR_CONFIG = 37,
ERR_BLE = 38 }

6.3.1 Detailed Description

Generic error codes used by ams libraries.

6.3.2 Typedef Documentation

6.3.2.1 err_code_t `typedef enum error_codes err_code_t`

This definition will be used for function return values.

6.3.3 Enumeration Type Documentation

6.3.3.1 error_codes `enum error_codes`

Values represent the error codes.

Enumerator

ERR_SUCCESS	Normal return code if everything was successful executed.
ERR_PERMISSION	Operation not permitted
ERR_MESSAGE	Message is invalid. For example: • Message type is not supported • incorrect crc • ...
ERR_MESSAGE_SIZE	Message has the wrong size.
ERR_POINTER	Pointer is invalid. Can be a NULL Pointer or point to a wrong memory area.
ERR_ACCESS	Access denied
ERR_ARGUMENT	Invalid argument
ERR_SIZE	Argument size is too long or too short.
ERR_NOT_SUPPORTED	Function is not supported/implemented.
ERR_TIMEOUT	Got timeout while waiting for answer.
ERR_CHECKSUM	Checksum comparision failed.
ERR_OVERFLOW	Data overflow detected.
ERR_EVENT	Error to get or set an event. For example: • event queue is full or empty • receive an unexpected event • ...
ERR_INTERRUPT	Error to get or set an interrupt. For example a interrupt resource is not available.
ERR_TIMER_ACCESS	Error while accessing timer periphery.
ERR_LED_ACCESS	Error while accessing LED periphery.

Enumerator

ERR_TEMP_SENSOR_ACCESS	Error while accessing temperature sensor.
ERR_DATA_TRANSFER	Communication error
ERR_FIFO	Faulty FIFO handling
ERR_OVER_TEMP	Overtemperature detected.
ERR_IDENTIFICATION	Sensor identification failed.
ERR_COM_INTERFACE	Generic communication interface error. For example: • communication interface is not available • error during open or close an communication interface • ...
ERR_SYNCHRONISATION	Synchronisation error, e.g. on protocol
ERR_PROTOCOL	Generic protocol error
ERR_MEMORY	Memory allocation error
ERR_THREAD	Thread can not created.
ERR_SPI	Error while accessing SPI periphery
ERR_DAC_ACCESS	Error while accessing DAC periphery.
ERR_I2C	Error while accessing I2C periphery.
ERR_NO_DATA	No data available.
ERR_SYSTEM_CONFIG	Error during system configuration. When a system resource is not available or generates an error for example.
ERR_USB_ACCESS	USB error
ERR_ADC_ACCESS	Error while accessing ADC periphery.
ERR_SENSOR_CONFIG	Error during sensor configuration.
ERR_SATURATION	Saturation detected
ERR_MUTEX	Error while mutex handling
ERR_ACCELEROMETER	Error while reading accelerometer data
ERR_CONFIG	Software component is not fully or correctly configured
ERR_BLE	Error while executing BLE stack function

6.4 Definitions

Description of the used data types.

Data Structures

- struct as7050_agc_config_t
- struct as7050_agc_status_t
- struct as7050_version
- union as7050_config_aoc_t
- union as7050_config_led_t
- union as7050_config_pd_t
- union as7050_config_ppg_t
- union as7050_config_sinc_t
- union as7050_config_seq_t
- union as7050_config_ref_t
- union as7050_config_gpio_t
- union as7050_config_ctrl_t
- union as7050_config_standby_t
- union as7050_config_fifo_t
- union as7050_config_ecg_t
- union as7050_config_afe_t
- union as7050_config_amp_t
- union as7050_config_tia_t
- union as7050_config_iir_t
- struct as7050_meas_config_t

Macros

- #define AS7050_MAX_GROUP_SIZE 128
- #define AS7050_CHANNEL_FLAG_GSR_OFFSET 12

Typedefs

- typedef enum as7050_reg_group_ids as7050_reg_group_ids_t
- typedef uint8_t dac_ref_control_t
- typedef uint8_t dac_ref_status_t
- typedef struct as7050_version as7050_version_t
- typedef void(∗ as7050_callback_t) (err_code_t error, uint8_t ∗p_data, uint16_t data_num, as7050_agc_status_t ∗p_agc_status, void ∗p_cb_param)

  Callback function, which transfers the measurement results to the application.

Enumerations

- enum as7050_reg_group_ids {
  AS7050_REG_GROUP_ID_CTRL = 0,
  AS7050_REG_GROUP_ID_GPIO = 1,
  AS7050_REG_GROUP_ID_STANDBY = 2,
  AS7050_REG_GROUP_ID_IIR = 3,
  AS7050_REG_GROUP_ID_REF = 4,
  AS7050_REG_GROUP_ID_AOC,
  AS7050_REG_GROUP_ID_PPG = 6,
  AS7050_REG_GROUP_ID_ECG = 7,
  AS7050_REG_GROUP_ID_AMP = 8,
  AS7050_REG_GROUP_ID_TIA,
  AS7050_REG_GROUP_ID_AFE,
  AS7050_REG_GROUP_ID_SINC = 11,
  AS7050_REG_GROUP_ID_LED = 12,
  AS7050_REG_GROUP_ID_PD = 13,
  AS7050_REG_GROUP_ID_FIFO = 14,
  AS7050_REG_GROUP_ID_SEQ = 15,
  AS7050_REG_GROUP_ID_NUM = 16 }
- enum as7050_channel_flags_t {
  AS7050_CHANNEL_FLAG_NONE = 0x0000,
  AS7050_CHANNEL_FLAG_PPG_1 = 0x0001,
  AS7050_CHANNEL_FLAG_PPG_2 = 0x0002,
  AS7050_CHANNEL_FLAG_PPG_3 = 0x0004,
  AS7050_CHANNEL_FLAG_PPG_4 = 0x0008,
  AS7050_CHANNEL_FLAG_PPG_5 = 0x0010,
  AS7050_CHANNEL_FLAG_PPG_6 = 0x0020,
  AS7050_CHANNEL_FLAG_PPG_7 = 0x0040,
  AS7050_CHANNEL_FLAG_PPG_8 = 0x0080,
  AS7050_CHANNEL_FLAG_ECG = 0x0100,
  AS7050_CHANNEL_FLAG_STATUS = 0x0200,
  AS7050_CHANNEL_FLAG_GSR = 0xF000 }
- enum FIFO_DATA_MARKERS {
  FIFO_DATA_MARKER_PPG_2_8 = 0x00,
  FIFO_DATA_MARKER_ECG = 0x01,
  FIFO_DATA_MARKER_PPG_1 = 0x02,
  FIFO_DATA_MARKER_STATUS = 0x03 }
- enum dac_ref_control {
  DAC_REF_CONTROL_CALC = 0,
  DAC_REF_CONTROL_ABORT }
- enum dac_ref_status {
  DAC_REF_STATUS_RUNNING = 0,
  DAC_REF_STATUS_ABORTED,
  DAC_REF_STATUS_FINISHED }
- enum as7050_reg_addresses {
  AS7050_REGADDR_GPIO1_CFG = 0x10,
  AS7050_REGADDR_GPIO2_CFG = 0x11,
  AS7050_REGADDR_INT_CFG = 0x12,
  AS7050_REGADDR_IO_CFGA = 0x13,
  AS7050_REGADDR_IO_CFGB = 0x14,
  AS7050_REGADDR_GPIO1_CFGB = 0x15,
  AS7050_REGADDR_GPIO2_CFGB = 0x16,
  AS7050_REGADDR_INT_CFGB = 0x17,

```
AS7050_REGADDR_PD_OFFSET_CFG = 0x18,
AS7050_REGADDR_PPG_MOD_CFGA = 0x19,
AS7050_REGADDR_PPG_MOD_CFGB = 0x1a,
AS7050_REGADDR_PPG_MOD_CFGC = 0x1b,
AS7050_REGADDR_PPG_MOD_CFGD = 0x1c,
AS7050_REGADDR_PPG_MOD_CFGE = 0x1d,
AS7050_REGADDR_TIA_CFGA = 0x1e,
AS7050_REGADDR_TIA_CFGB = 0x1f,
AS7050_REGADDR_ECG_AMP_CFGA = 0x20,
AS7050_REGADDR_ECG_AMP_CFGB = 0x21,
AS7050_REGADDR_ECG_AMP_CFGC = 0x22,
AS7050_REGADDR_PDSEL_CFG = 0x23,
AS7050_REGADDR_ECG_SOURCE = 0x24,
AS7050_REGADDR_ECG_MOD_CFGA = 0x25,
AS7050_REGADDR_TIA_CFGC = 0x26,
AS7050_REGADDR_LOWVDS_WAIT = 0x27,
AS7050_REGADDR_LED1_ICTRL = 0x28,
AS7050_REGADDR_LED2_ICTRL = 0x29,
AS7050_REGADDR_LED3_ICTRL = 0x2a,
AS7050_REGADDR_LED4_ICTRL = 0x2b,
AS7050_REGADDR_LED5_ICTRL = 0x2c,
AS7050_REGADDR_LED6_ICTRL = 0x2d,
AS7050_REGADDR_LED7_ICTRL = 0x2e,
AS7050_REGADDR_LED8_ICTRL = 0x2f,
AS7050_REGADDR_REF_CFGA = 0x30,
AS7050_REGADDR_REF_CFGB = 0x31,
AS7050_REGADDR_AFE_DAC0L = 0x32,
AS7050_REGADDR_AFE_DAC1L = 0x33,
AS7050_REGADDR_AFE_DACH = 0x34,
AS7050_REGADDR_AFE_CFGA = 0x35,
AS7050_REGADDR_AFE_CFGB = 0x36,
AS7050_REGADDR_CONTROL = 0x37,
AS7050_REGADDR_CGB_CFG = 0x38,
AS7050_REGADDR_ECG_AMP_CFGE = 0x39,
AS7050_REGADDR_AFE_GSR = 0x3a,
AS7050_REGADDR_SEQ_SAMPLE = 0x40,
AS7050_REGADDR_SEQ_PPGA = 0x41,
AS7050_REGADDR_SEQ_PPGB = 0x42,
AS7050_REGADDR_PD_PPG1 = 0x43,
AS7050_REGADDR_PD_PPG2 = 0x44,
AS7050_REGADDR_PD_PPG3 = 0x45,
AS7050_REGADDR_PD_PPG4 = 0x46,
AS7050_REGADDR_PD_PPG5 = 0x47,
AS7050_REGADDR_PD_PPG6 = 0x48,
AS7050_REGADDR_PD_PPG7 = 0x49,
AS7050_REGADDR_PD_PPG8 = 0x4a,
AS7050_REGADDR_PD_TIA = 0x4b,
AS7050_REGADDR_LED_INIT = 0x4c,
AS7050_REGADDR_LED_PPG1 = 0x4d,
AS7050_REGADDR_LED_PPG2 = 0x4e,
AS7050_REGADDR_LED_PPG3 = 0x4f,
AS7050_REGADDR_LED_PPG4 = 0x50,
AS7050_REGADDR_LED_PPG5 = 0x51,
AS7050_REGADDR_LED_PPG6 = 0x52,
AS7050_REGADDR_LED_PPG7 = 0x53,
AS7050_REGADDR_LED_PPG8 = 0x54,
AS7050_REGADDR_LED_TIA = 0x55,
AS7050_REGADDR_LED_MODE = 0x56,
AS7050_REGADDR_SEQ_COUNT = 0x57,
AS7050_REGADDR_SEQ_MODE = 0x59,
AS7050_REGADDR_SEQ_START = 0x5a,
AS7050_REGADDR_SINC_PPG_CFGA = 0x5b,
AS7050_REGADDR_SINC_PPG_CFGB = 0x5c,
AS7050_REGADDR_SINC_PPG_CFGC = 0x5d,
AS7050_REGADDR_SINC_ECG_CFGA = 0x5e,
AS7050_REGADDR_SINC_ECG_CFGB = 0x5f,
AS7050_REGADDR_SINC_ECG_CFGC = 0x60,
AS7050_REGADDR_IIR_CFG = 0x61,
AS7050_REGADDR_IIR_COEFF_ADDR = 0x62,
AS7050_REGADDR_IIR_COEFF_DATA = 0x63,
AS7050_REGADDR_OVS_CFG = 0x64,
AS7050_REGADDR_AOC_IOS_PPG1 = 0x65,
AS7050_REGADDR_AOC_IOS_PPG2 = 0x66,
AS7050_REGADDR_AOC_IOS_PPG3 = 0x67,
AS7050_REGADDR_AOC_IOS_PPG4 = 0x68,
AS7050_REGADDR_AOC_IOS_PPG5 = 0x69,
AS7050_REGADDR_AOC_IOS_PPG6 = 0x6a,
AS7050_REGADDR_AOC_IOS_PPG7 = 0x6b,
AS7050_REGADDR_AOC_IOS_PPG8 = 0x6c,
AS7050_REGADDR_AOC_PPG_THH = 0x6d,
AS7050_REGADDR_AOC_PPG_THL = 0x6e,
AS7050_REGADDR_AOC_PPG_CFG = 0x6f,
AS7050_REGADDR_AOC_IOS_ECG = 0x70,
AS7050_REGADDR_AOC_ECG_THH = 0x71,
AS7050_REGADDR_AOC_ECG_THL = 0x72,
AS7050_REGADDR_AOC_ECG_CFG = 0x73,
AS7050_REGADDR_AOC_IOS_LEDOFF = 0x74,
AS7050_REGADDR_FIFO_CTRL = 0x75,
AS7050_REGADDR_FIFO_THRESHOLD = 0x76,
AS7050_REGADDR_FIFO_LEVEL0 = 0x77,
AS7050_REGADDR_FIFO_LEVEL1 = 0x78,
AS7050_REGADDR_STATUS = 0x80,
AS7050_REGADDR_STATUS_CGBA = 0x81,
AS7050_REGADDR_STATUS_CGBB = 0x82,
AS7050_REGADDR_STATUS_MOD = 0x83,
AS7050_REGADDR_STATUS_LED = 0x84,
AS7050_REGADDR_IRQ_ENABLE = 0x88,
AS7050_REGADDR_GPIO_IO = 0x8f,
AS7050_REGADDR_REVISION = 0x90,
AS7050_REGADDR_STANDBY_CFGA = 0xb0,
AS7050_REGADDR_STANDBY_CFGB = 0xb1,
AS7050_REGADDR_PPG_BYTEL = 0xf4,
AS7050_REGADDR_PPG_BYTEM = 0xf5,
AS7050_REGADDR_PPG_BYTEH = 0xf6,
AS7050_REGADDR_ECG_BYTEL = 0xf8,
AS7050_REGADDR_ECG_BYTEM = 0xf9,
AS7050_REGADDR_ECG_BYTEH = 0xfa,
AS7050_REGADDR_FIFOL = 0xfc,
```

AS7050_REGADDR_FIFOM = 0xfd,
AS7050_REGADDR_FIFOH = 0xfe,
AS7050_REGADDR_BYTE0 = 0xff }

- enum as7050_channel_t {
AS7050_CHANNEL_DISABLED = 0,
AS7050_CHANNEL_PPG_1 = 1,
AS7050_CHANNEL_PPG_2 = 2,
AS7050_CHANNEL_PPG_3 = 3,
AS7050_CHANNEL_PPG_4 = 4,
AS7050_CHANNEL_PPG_5 = 5,
AS7050_CHANNEL_PPG_6 = 6,
AS7050_CHANNEL_PPG_7 = 7,
AS7050_CHANNEL_PPG_8 = 8,
AS7050_CHANNEL_TIA = 9 }
- enum as7050_agc_change_states_t {
AS7050_AGC_STATE_UNCHANGED = 0,
AS7050_AGC_STATE_INCREASED = 1,
AS7050_AGC_STATE_DECREASED = 2,
AS7050_AGC_STATE_ATMIN = 3,
AS7050_AGC_STATE_ATMAX = 4,
AS7050_AGC_STATE_UNKNOWN = 5 }
- enum as7050_agc_mode_t {
AS7050_AGC_MODE_DISABLED = 0,
AS7050_AGC_MODE_PPG_ONE_CHANNEL = 1,
AS7050_AGC_MODE_PPG_TWO_CHANNEL = 2,
AS7050_AGC_MODE_MAX_NUM = 4 }
- enum as7050_channel_group_t {
AS7050_CHANNEL_GROUP_A = 0,
AS7050_CHANNEL_GROUP_B = 1,
AS7050_NUM_CHANNEL_GROUP = 2 }

6.4.1 Detailed Description

Description of the used data types.

These are the type definitions used by AS7050 chip library.

6.4.2 Macro Definition Documentation

6.4.2.1 AS7050_MAX_GROUP_SIZE `#define AS7050_MAX_GROUP_SIZE 128`

maximum space for payload of register groups

6.4.2.2 AS7050_CHANNEL_FLAG_GSR_OFFSET `#define AS7050_CHANNEL_FLAG_GSR_OFFSET 12`

Bit position of GSR data inside as7050_channel_flags_t

6.4.3 Typedef Documentation

6.4.3.1 as7050_reg_group_ids_t `typedef enum as7050_reg_group_ids as7050_reg_group_ids_t`

Definition of the register group IDs

6.4.3.2 dac_ref_control_t `typedef uint8_t dac_ref_control_t`

Type for dac_ref_control.

6.4.3.3 dac_ref_status_t `typedef uint8_t dac_ref_status_t`

Type for dac_ref_status.

6.4.3.4 as7050_version_t `typedef struct as7050_version as7050_version_t`

Version information of the library

6.4.3.5 as7050_callback_t `typedef void(* as7050_callback_t) (err_code_t error, uint8_t *p_data, uint16_t data_num, as7050_agc_status_t *p_agc_status, void *p_cb_param)`

Callback function, which transfers the measurement results to the application.

This callback type will be registered via the function as7050_initialize. During the measurement, this function transfers the cyclic results.

Parameters

in	*error*	Default ERR_SUCCESS, otherwise an error is occurred during measurement and the measurement. stops. See error_codes
in	*p_data*	Pointer to the measurement data, the content depends on configuration.
in	*data_num*	Number of 16bit data
in	*p_agc_status*	Pointer to AGC status structure
in	*p_cb_param*	Application parameter which was defined during call of as7050_initialize.

6.4.4 Enumeration Type Documentation

6.4.4.1 as7050_reg_group_ids `enum as7050_reg_group_ids`

Definition of the register group IDs

Enumerator

AS7050_REG_GROUP_ID_CTRL	Group-ID of all registers of group 'Control'. See as7050_config_ctrl_t
AS7050_REG_GROUP_ID_GPIO	Group-ID of all registers of group 'GPIO'. See as7050_config_gpio_t
AS7050_REG_GROUP_ID_STANDBY	Group-ID of all registers of group 'Standby'. See as7050_config_standby_t
AS7050_REG_GROUP_ID_IIR	Group-ID of all registers of group 'IIR filter'. See as7050_config_iir_t
AS7050_REG_GROUP_ID_REF	Group-ID of all registers of group 'Reference'. See as7050_config_ref_t
AS7050_REG_GROUP_ID_AOC	Group-ID of all registers of group 'Automatic Offset Control'. See as7050_config_aoc_t
AS7050_REG_GROUP_ID_PPG	Group-ID of all registers of group 'PPG'. See as7050_config_ppg_t
AS7050_REG_GROUP_ID_ECG	Group-ID of all registers of group 'ECG'. See as7050_config_ecg_t
AS7050_REG_GROUP_ID_AMP	Group-ID of all registers of group 'amplifier'. See as7050_config_amp_t
AS7050_REG_GROUP_ID_TIA	Group-ID of all registers of group 'Transimpedance amplifier'. See as7050_config_tia_t
AS7050_REG_GROUP_ID_AFE	Group-ID of all registers of group 'Analog front end'. See as7050_config_afe_t
AS7050_REG_GROUP_ID_SINC	Group-ID of all registers of group 'Sinc filter'. See as7050_config_sinc_t
AS7050_REG_GROUP_ID_LED	Group-ID of all registers of group 'LED'. See as7050_config_led_t
AS7050_REG_GROUP_ID_PD	Group-ID of all registers of group 'Photodiodes'. See as7050_config_pd_t
AS7050_REG_GROUP_ID_FIFO	Group-ID of all registers of group 'FIFO'. See as7050_config_fifo_t
AS7050_REG_GROUP_ID_SEQ	Group-ID of all registers of group 'Sequencer'. See as7050_config_seq_t
AS7050_REG_GROUP_ID_NUM	Maximum number of supported register groups

6.4.4.2 as7050_channel_flags_t `enum as7050_channel_flags_t`

Channel flags

Enumerator

AS7050_CHANNEL_FLAG_NONE	No channel is selected
AS7050_CHANNEL_FLAG_PPG_1	Flag for PPG 1 channel
AS7050_CHANNEL_FLAG_PPG_2	Flag for PPG 2 channel
AS7050_CHANNEL_FLAG_PPG_3	Flag for PPG 3 channel
AS7050_CHANNEL_FLAG_PPG_4	Flag for PPG 4 channel
AS7050_CHANNEL_FLAG_PPG_5	Flag for PPG 5 channel
AS7050_CHANNEL_FLAG_PPG_6	Flag for PPG 6 channel
AS7050_CHANNEL_FLAG_PPG_7	Flag for PPG 7 channel
AS7050_CHANNEL_FLAG_PPG_8	Flag for PPG 8 channel

Enumerator

AS7050_CHANNEL_FLAG_ECG	Flag for ECG channel
AS7050_CHANNEL_FLAG_STATUS	Flag for status channel
AS7050_CHANNEL_FLAG_GSR	When performing GSR measurement, DAC 0 and DAC 1 are measured on the ECG channel. The chip alternates between DAC 0 and DAC 1 automatically. The bits masked by this bitmask contain the number of samples that are taken per DAC before the chip alternates to the other DAC.

6.4.4.3 FIFO_DATA_MARKERS `enum FIFO_DATA_MARKERS`

First two bits of the measurement data, which describes the type of the sample

Enumerator

FIFO_DATA_MARKER_PPG_2↩ _8	PPG sub samples 2 - 8
FIFO_DATA_MARKER_ECG	ECG sample
FIFO_DATA_MARKER_PPG_1	First PPG sample
FIFO_DATA_MARKER_STATUS	Status marker for AOC information

6.4.4.4 dac_ref_control `enum dac_ref_control`

Control states for DAC reference calculation

Enumerator

DAC_REF_CONTROL_CALC	Start or continue the DAC reference calculation
DAC_REF_CONTROL_ABORT	Abort the DAC reference calculation

6.4.4.5 dac_ref_status `enum dac_ref_status`

Status for DAC reference calculation

Enumerator

DAC_REF_STATUS_RUNNING	DAC reference calculation is running
DAC_REF_STATUS_ABORTED	DAC reference calculation was aborted
DAC_REF_STATUS_FINISHED	DAC reference calculation is finished

6.4.4.6 as7050_reg_addresses `enum as7050_reg_addresses`

Register definition of AS7050

Enumerator

AS7050_REGADDR_GPIO1_CFG	Register GPIO1_CFG
AS7050_REGADDR_GPIO2_CFG	Register GPIO2_CFG
AS7050_REGADDR_INT_CFG	Register INT_CFG
AS7050_REGADDR_IO_CFGA	Register IO_CFGA
AS7050_REGADDR_IO_CFGB	Register IO_CFGB
AS7050_REGADDR_GPIO1_CFGB	Register GPIO1_CFGB
AS7050_REGADDR_GPIO2_CFGB	Register GPIO2_CFGB
AS7050_REGADDR_INT_CFGB	Register INT_CFGB
AS7050_REGADDR_PD_OFFSET_CFG	Register PD_OFFSET_CFG
AS7050_REGADDR_PPG_MOD_CFGA	Register PPG_MOD_CFGA
AS7050_REGADDR_PPG_MOD_CFGB	Register PPG_MOD_CFGB
AS7050_REGADDR_PPG_MOD_CFGC	Register PPG_MOD_CFGC
AS7050_REGADDR_PPG_MOD_CFGD	Register PPG_MOD_CFGD
AS7050_REGADDR_PPG_MOD_CFGE	Register PPG_MOD_CFGE
AS7050_REGADDR_TIA_CFGA	Register TIA_CFGA
AS7050_REGADDR_TIA_CFGB	Register TIA_CFGB
AS7050_REGADDR_ECG_AMP_CFGA	Register ECG_AMP_CFGA
AS7050_REGADDR_ECG_AMP_CFGB	Register ECG_AMP_CFGB
AS7050_REGADDR_ECG_AMP_CFGC	Register ECG_AMP_CFGC
AS7050_REGADDR_PDSEL_CFG	Register PDSEL_CFG
AS7050_REGADDR_ECG_SOURCE	Register ECG_SOURCE
AS7050_REGADDR_ECG_MOD_CFGA	Register ECG_MOD_CFGA
AS7050_REGADDR_TIA_CFGC	Register TIA_CFGC
AS7050_REGADDR_LOWVDS_WAIT	Register LOWVDS_WAIT
AS7050_REGADDR_LED1_ICTRL	Register LED1_ICTRL
AS7050_REGADDR_LED2_ICTRL	Register LED2_ICTRL
AS7050_REGADDR_LED3_ICTRL	Register LED3_ICTRL
AS7050_REGADDR_LED4_ICTRL	Register LED4_ICTRL
AS7050_REGADDR_LED5_ICTRL	Register LED5_ICTRL
AS7050_REGADDR_LED6_ICTRL	Register LED6_ICTRL
AS7050_REGADDR_LED7_ICTRL	Register LED7_ICTRL
AS7050_REGADDR_LED8_ICTRL	Register LED8_ICTRL
AS7050_REGADDR_REF_CFGA	Register REF_CFGA
AS7050_REGADDR_REF_CFGB	Register REF_CFGB
AS7050_REGADDR_AFE_DAC0L	Register AFE_DAC0L
AS7050_REGADDR_AFE_DAC1L	Register AFE_DAC1L
AS7050_REGADDR_AFE_DACH	Register AFE_DACH

Enumerator

AS7050_REGADDR_AFE_CFGA	Register AFE_CFGA
AS7050_REGADDR_AFE_CFGB	Register AFE_CFGB
AS7050_REGADDR_CONTROL	Register CONTROL
AS7050_REGADDR_CGB_CFG	Register CGB_CFG
AS7050_REGADDR_ECG_AMP_CFGE	Register ECG_AMP_CFGE
AS7050_REGADDR_AFE_GSR	Register AFE_GSR
AS7050_REGADDR_SEQ_SAMPLE	Register SEQ_SAMPLE
AS7050_REGADDR_SEQ_PPGA	Register SEQ_PPGA
AS7050_REGADDR_SEQ_PPGB	Register SEQ_PPGB
AS7050_REGADDR_PD_PPG1	Register PD_PPG1
AS7050_REGADDR_PD_PPG2	Register PD_PPG2
AS7050_REGADDR_PD_PPG3	Register PD_PPG3
AS7050_REGADDR_PD_PPG4	Register PD_PPG4
AS7050_REGADDR_PD_PPG5	Register PD_PPG5
AS7050_REGADDR_PD_PPG6	Register PD_PPG6
AS7050_REGADDR_PD_PPG7	Register PD_PPG7
AS7050_REGADDR_PD_PPG8	Register PD_PPG8
AS7050_REGADDR_PD_TIA	Register PD_TIA
AS7050_REGADDR_LED_INIT	Register LED_INIT
AS7050_REGADDR_LED_PPG1	Register LED_PPG1
AS7050_REGADDR_LED_PPG2	Register LED_PPG2
AS7050_REGADDR_LED_PPG3	Register LED_PPG3
AS7050_REGADDR_LED_PPG4	Register LED_PPG4
AS7050_REGADDR_LED_PPG5	Register LED_PPG5
AS7050_REGADDR_LED_PPG6	Register LED_PPG6
AS7050_REGADDR_LED_PPG7	Register LED_PPG7
AS7050_REGADDR_LED_PPG8	Register LED_PPG8
AS7050_REGADDR_LED_TIA	Register LED_TIA
AS7050_REGADDR_LED_MODE	Register LED_MODE
AS7050_REGADDR_SEQ_COUNT	Register SEQ_COUNT
AS7050_REGADDR_SEQ_MODE	Register SEQ_MODE
AS7050_REGADDR_SEQ_START	Register SEQ_START
AS7050_REGADDR_SINC_PPG_CFGA	Register SINC_PPG_CFGA
AS7050_REGADDR_SINC_PPG_CFGB	Register SINC_PPG_CFGB
AS7050_REGADDR_SINC_PPG_CFGC	Register SINC_PPG_CFGC
AS7050_REGADDR_SINC_ECG_CFGA	Register SINC_ECG_CFGA
AS7050_REGADDR_SINC_ECG_CFGB	Register SINC_ECG_CFGB
AS7050_REGADDR_SINC_ECG_CFGC	Register SINC_ECG_CFGC
AS7050_REGADDR_IIR_CFG	Register IIR_CFG
AS7050_REGADDR_IIR_COEFF_ADDR	Register IIR_COEFF_ADDR
AS7050_REGADDR_IIR_COEFF_DATA	Register IIR_COEFF_DATA
AS7050_REGADDR_OVS_CFG	Register OVS_CFG
AS7050_REGADDR_AOC_IOS_PPG1	Register AOC_IOS_PPG1

Enumerator

AS7050_REGADDR_AOC_IOS_PPG2	Register AOC_IOS_PPG2
AS7050_REGADDR_AOC_IOS_PPG3	Register AOC_IOS_PPG3
AS7050_REGADDR_AOC_IOS_PPG4	Register AOC_IOS_PPG4
AS7050_REGADDR_AOC_IOS_PPG5	Register AOC_IOS_PPG5
AS7050_REGADDR_AOC_IOS_PPG6	Register AOC_IOS_PPG6
AS7050_REGADDR_AOC_IOS_PPG7	Register AOC_IOS_PPG7
AS7050_REGADDR_AOC_IOS_PPG8	Register AOC_IOS_PPG8
AS7050_REGADDR_AOC_PPG_THH	Register AOC_PPG_THH
AS7050_REGADDR_AOC_PPG_THL	Register AOC_PPG_THL
AS7050_REGADDR_AOC_PPG_CFG	Register AOC_PPG_CFG
AS7050_REGADDR_AOC_IOS_ECG	Register AOC_IOS_ECG
AS7050_REGADDR_AOC_ECG_THH	Register AOC_ECG_THH
AS7050_REGADDR_AOC_ECG_THL	Register AOC_ECG_THL
AS7050_REGADDR_AOC_ECG_CFG	Register AOC_ECG_CFG
AS7050_REGADDR_AOC_IOS_LEDOFF	Register AOC_IOS_LEDOFF
AS7050_REGADDR_FIFO_CTRL	Register FIFO_CTRL
AS7050_REGADDR_FIFO_THRESHOLD	Register FIFO_THRESHOLD
AS7050_REGADDR_FIFO_LEVEL0	Register FIFO_LEVEL0
AS7050_REGADDR_FIFO_LEVEL1	Register FIFO_LEVEL1
AS7050_REGADDR_STATUS	Register STATUS
AS7050_REGADDR_STATUS_CGBA	Register STATUS_CGBA
AS7050_REGADDR_STATUS_CGBB	Register STATUS_CGBB
AS7050_REGADDR_STATUS_MOD	Register STATUS_MOD
AS7050_REGADDR_STATUS_LED	Register STATUS_LED
AS7050_REGADDR_IRQ_ENABLE	Register IRQ_ENABLE
AS7050_REGADDR_GPIO_IO	Register GPIO_IO
AS7050_REGADDR_REVISION	Register REVISION
AS7050_REGADDR_STANDBY_CFGA	Register STANDBY_CFGA
AS7050_REGADDR_STANDBY_CFGB	Register STANDBY_CFGB
AS7050_REGADDR_PPG_BYTEL	Register PPG_BYTEL
AS7050_REGADDR_PPG_BYTEM	Register PPG_BYTEM
AS7050_REGADDR_PPG_BYTEH	Register PPG_BYTEH
AS7050_REGADDR_ECG_BYTEL	Register ECG_BYTEL
AS7050_REGADDR_ECG_BYTEM	Register ECG_BYTEM
AS7050_REGADDR_ECG_BYTEH	Register ECG_BYTEH
AS7050_REGADDR_FIFOL	Register FIFOL
AS7050_REGADDR_FIFOM	Register FIFOM
AS7050_REGADDR_FIFOH	Register FIFOH
AS7050_REGADDR_BYTE0	Register BYTE0

6.4.4.7 as7050_channel_t `enum as7050_channel_t`

AGC channel selection

6.4.4.8 as7050_agc_change_states_t `enum as7050_agc_change_states_t`

The possible values of the iLed_change field of the STATE_AGC structure.

Enumerator

AS7050_AGC_STATE_UNCHANGED	AGC state was not changed.
AS7050_AGC_STATE_INCREASED	AGC state was increased.
AS7050_AGC_STATE_DECREASED	AGC state was decreased.
AS7050_AGC_STATE_ATMIN	AGC needed a decrease, but was not possible as is is at the minumum already.
AS7050_AGC_STATE_ATMAX	AGC needed an increase, but was not possible as is is at the maximum already.
AS7050_AGC_STATE_UNKNOWN	No check on the AGC state was done, as the algorithm has not run yet.

6.4.4.9 as7050_agc_mode_t `enum as7050_agc_mode_t`

AGC algorithm modes

Enumerator

AS7050_AGC_MODE_DISABLED	AGC-Algorithm is disables
AS7050_AGC_MODE_PPG_ONE_CHANNEL	AGC-Algorithm is configured for HRM measurement
AS7050_AGC_MODE_PPG_TWO_CHANNEL	AGC-Algorithm is configured for SPO2 measurement
AS7050_AGC_MODE_MAX_NUM	only the first three modes are suported!

6.4.4.10 as7050_channel_group_t `enum as7050_channel_group_t`

AGC channel groups to support more than one LED configuration

7 Data Structure Documentation

7.1 as7050_config_afe_t::afe_regs Struct Reference

Data Fields

- uint8_t afe_dac0l
- uint8_t afe_dac1l
- uint8_t afe_dach
- uint8_t afe_cfga
- uint8_t afe_cfgb
- uint8_t afe_gsr

7.1.1 Detailed Description

Register content for AFE configuration

7.1.2 Field Documentation

7.1.2.1 afe_dac0l `uint8_t as7050_config_afe_t::afe_regs::afe_dac0l`

Content of register AS7050_REGADDR_AFE_DAC0L

7.1.2.2 afe_dac1l `uint8_t as7050_config_afe_t::afe_regs::afe_dac1l`

Content of register AS7050_REGADDR_AFE_DAC1L

7.1.2.3 afe_dach `uint8_t as7050_config_afe_t::afe_regs::afe_dach`

Content of register AS7050_REGADDR_AFE_DACH

7.1.2.4 afe_cfga `uint8_t as7050_config_afe_t::afe_regs::afe_cfga`

Content of register AS7050_REGADDR_AFE_CFGA

7.1.2.5 afe_cfgb `uint8_t as7050_config_afe_t::afe_regs::afe_cfgb`

Content of register AS7050_REGADDR_AFE_CFGB

7.1.2.6 afe_gsr `uint8_t as7050_config_afe_t::afe_regs::afe_gsr`

Content of register AS7050_REGADDR_AFE_GSR

7.2 as7050_config_amp_t::amp_regs Struct Reference

Data Fields

- uint8_t ecg_amp_cfga
- uint8_t ecg_amp_cfgb
- uint8_t ecg_amp_cfgc
- uint8_t ecg_amp_cfge

7.2.1 Detailed Description

Register content for amplifer configuration

7.2.2 Field Documentation

7.2.2.1 ecg_amp_cfga `uint8_t as7050_config_amp_t::amp_regs::ecg_amp_cfga`

Content of register AS7050_REGADDR_ECG_AMP_CFGA

7.2.2.2 ecg_amp_cfgb `uint8_t as7050_config_amp_t::amp_regs::ecg_amp_cfgb`

Content of register AS7050_REGADDR_ECG_AMP_CFGB

7.2.2.3 ecg_amp_cfgc `uint8_t as7050_config_amp_t::amp_regs::ecg_amp_cfgc`

Content of register AS7050_REGADDR_ECG_AMP_CFGC

7.2.2.4 ecg_amp_cfge `uint8_t as7050_config_amp_t::amp_regs::ecg_amp_cfge`

Content of register AS7050_REGADDR_ECG_AMP_CFGE

7.3 as7050_config_aoc_t::aoc_regs Struct Reference

Data Fields

- uint8_t aoc_ios_ppg1
- uint8_t aoc_ios_ppg2
- uint8_t aoc_ios_ppg3
- uint8_t aoc_ios_ppg4
- uint8_t aoc_ios_ppg5
- uint8_t aoc_ios_ppg6
- uint8_t aoc_ios_ppg7
- uint8_t aoc_ios_ppg8
- uint8_t aoc_ppg_thh
- uint8_t aoc_ppg_thl
- uint8_t aoc_ppg_cfg
- uint8_t aoc_ios_ecg
- uint8_t aoc_ecg_thh
- uint8_t aoc_ecg_thl
- uint8_t aoc_ecg_cfg
- uint8_t aoc_ios_ledoff

7.3.1 Detailed Description

Register content for AOC configuration

7.3.2 Field Documentation

7.3.2.1 aoc_ios_ppg1 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg1`

Content of register AS7050_REGADDR_AOC_IOS_PPG1

7.3.2.2 aoc_ios_ppg2 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg2`

Content of register AS7050_REGADDR_AOC_IOS_PPG2

7.3.2.3 aoc_ios_ppg3 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg3`

Content of register AS7050_REGADDR_AOC_IOS_PPG3

7.3.2.4 aoc_ios_ppg4 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg4`

Content of register AS7050_REGADDR_AOC_IOS_PPG4

7.3.2.5 aoc_ios_ppg5 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg5`

Content of register AS7050_REGADDR_AOC_IOS_PPG5

7.3.2.6 aoc_ios_ppg6 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg6`

Content of register AS7050_REGADDR_AOC_IOS_PPG6

7.3.2.7 aoc_ios_ppg7 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg7`

Content of register AS7050_REGADDR_AOC_IOS_PPG7

7.3.2.8 aoc_ios_ppg8 `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ppg8`

Content of register AS7050_REGADDR_AOC_IOS_PPG8

7.3.2.9 aoc_ppg_thh `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ppg_thh`

Content of register AS7050_REGADDR_AOC_PPG_THH

7.3.2.10 aoc_ppg_thl `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ppg_thl`

Content of register AS7050_REGADDR_AOC_PPG_THL

7.3.2.11 aoc_ppg_cfg `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ppg_cfg`

Content of register AS7050_REGADDR_AOC_PPG_CFG

7.3.2.12 aoc_ios_ecg `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ecg`

Content of register AS7050_REGADDR_AOC_IOS_ECG

7.3.2.13 aoc_ecg_thh `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ecg_thh`

Content of register AS7050_REGADDR_AOC_ECG_THH

7.3.2.14 aoc_ecg_thl `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ecg_thl`

Content of register AS7050_REGADDR_AOC_ECG_THL

7.3.2.15 aoc_ecg_cfg `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ecg_cfg`

Content of register AS7050_REGADDR_AOC_ECG_CFG

7.3.2.16 aoc_ios_ledoff `uint8_t as7050_config_aoc_t::aoc_regs::aoc_ios_ledoff`

Content of register AS7050_REGADDR_AOC_IOS_LEDOFF

7.4 as7050_agc_config_t Struct Reference

Data Fields

- uint8_t mode
- uint8_t res1
- uint8_t channel [AS7050_NUM_CHANNEL_GROUP]
- uint8_t current_min [AS7050_NUM_CHANNEL_GROUP]
- uint8_t current_max [AS7050_NUM_CHANNEL_GROUP]
- int32_t threshold_min
- int32_t threshold_max
- uint16_t sample_cnt
- uint16_t reset_interval

7.4.1 Detailed Description

This is the external configuration structure for the AGC algorithm

7.4.2 Field Documentation

7.4.2.1 mode `uint8_t as7050_agc_config_t::mode`

Definition which algorithms will be used. See as7050_agc_mode_t

7.4.2.2 res1 `uint8_t as7050_agc_config_t::res1`

unused parameter, only for alignment purposes

7.4.2.3 channel `uint8_t as7050_agc_config_t::channel[AS7050_NUM_CHANNEL_GROUP]`

as7050_channel_t

7.4.2.4 current_min `uint8_t as7050_agc_config_t::current_min[AS7050_NUM_CHANNEL_GROUP]`

The minimum allowed LED group current

7.4.2.5 current_max `uint8_t as7050_agc_config_t::current_max[AS7050_NUM_CHANNEL_GROUP]`

The maximum allowed LED group current

7.4.2.6 threshold_min `int32_t as7050_agc_config_t::threshold_min`

Minimum threshold of the controlled signal in ADC counts

7.4.2.7 threshold_max `int32_t as7050_agc_config_t::threshold_max`

Maximum threshold of the controlled signal in ADC counts

7.4.2.8 sample_cnt `uint16_t as7050_agc_config_t::sample_cnt`

Number of samples to average for calculating the signal mean

7.4.2.9 reset_interval `uint16_t as7050_agc_config_t::reset_interval`

Interval in milliseconds to reset temporary AGC parameters

7.5 as7050_agc_status_t Struct Reference

Data Fields

- uint8_t led_change [AS7050_NUM_CHANNEL_GROUP]
- uint8_t led_current [AS7050_NUM_CHANNEL_GROUP]
- uint8_t pd_offset_change [AS7050_NUM_CHANNEL_GROUP]
- uint8_t pd_offset_current [AS7050_NUM_CHANNEL_GROUP]

7.5.1 Detailed Description

This is the external status structure for the AGC algorithm

7.5.2 Field Documentation

7.5.2.1 led_change `uint8_t as7050_agc_status_t::led_change[AS7050_NUM_CHANNEL_GROUP]`

Gives an information if the AGC changed the LED current. See as7050_agc_change_states_t

7.5.2.2 led_current `uint8_t as7050_agc_status_t::led_current[AS7050_NUM_CHANNEL_GROUP]`

Actual configured current of the LED group

7.5.2.3 pd_offset_change `uint8_t as7050_agc_status_t::pd_offset_change[AS7050_NUM_CHANNEL_GR↩ OUP]`

Gives an information if the AGC changed the PD offset during LED-ON. See as7050_agc_change_states_t

7.5.2.4 pd_offset_current `uint8_t as7050_agc_status_t::pd_offset_current[AS7050_NUM_CHANNEL_G↩ ROUP]`

PD offset value

7.6 as7050_config_afe_t Union Reference

Data Structures

- struct afe_regs

Data Fields

- struct as7050_config_afe_t::afe_regs reg_vals
- uint8_t reg_buffer [sizeof(struct afe_regs)]

7.6.1 Detailed Description

Register group for configuration of the Analog Front End (AFE)

7.6.2 Field Documentation

7.6.2.1 reg_vals `struct as7050_config_afe_t::afe_regs as7050_config_afe_t::reg_vals`

Register content for AFE configuration

7.6.2.2 reg_buffer `uint8_t as7050_config_afe_t::reg_buffer[sizeof(struct afe_regs)]`

Register content for AFE configuration

7.7 as7050_config_amp_t Union Reference

Data Structures

- struct amp_regs

Data Fields

- struct as7050_config_amp_t::amp_regs reg_vals
- uint8_t reg_buffer [sizeof(struct amp_regs)]

7.7.1 Detailed Description

Register group for configuration of the amplifer

7.7.2 Field Documentation

7.7.2.1 reg_vals `struct as7050_config_amp_t::amp_regs as7050_config_amp_t::reg_vals`

Register content for amplifer configuration

7.7.2.2 reg_buffer `uint8_t as7050_config_amp_t::reg_buffer[sizeof(struct amp_regs)]`

Register content for amplifer configuration

7.8 as7050_config_aoc_t Union Reference

Data Structures

- struct aoc_regs

Data Fields

- struct as7050_config_aoc_t::aoc_regs reg_vals
- uint8_t reg_buffer [sizeof(struct aoc_regs)]

7.8.1 Detailed Description

Register group for configuration of Automatic Offset Control (AOC)

7.8.2 Field Documentation

7.8.2.1 reg_vals `struct as7050_config_aoc_t::aoc_regs as7050_config_aoc_t::reg_vals`

Register content for AOC configuration

7.8.2.2 reg_buffer `uint8_t as7050_config_aoc_t::reg_buffer[sizeof(struct aoc_regs)]`

Register content for AOC configuration

7.9 as7050_config_ctrl_t Union Reference

Data Structures

- struct ctrl_regs

Data Fields

- struct as7050_config_ctrl_t::ctrl_regs reg_vals
- uint8_t reg_buffer [sizeof(struct ctrl_regs)]

7.9.1 Detailed Description

Register group for configuration of the CONTROL register

7.9.2 Field Documentation

7.9.2.1 reg_vals `struct as7050_config_ctrl_t::ctrl_regs as7050_config_ctrl_t::reg_vals`

Register content for CONTROL register

7.9.2.2 reg_buffer `uint8_t as7050_config_ctrl_t::reg_buffer[sizeof(struct ctrl_regs)]`

Register content for CONTROL register

7.10 as7050_config_ecg_t Union Reference

Data Structures

- struct ecg_regs

Data Fields

- struct as7050_config_ecg_t::ecg_regs reg_vals
- uint8_t reg_buffer [sizeof(struct ecg_regs)]

7.10.1 Detailed Description

Register group for configuration of the ECG channel

7.10.2 Field Documentation

7.10.2.1 reg_vals `struct as7050_config_ecg_t::ecg_regs as7050_config_ecg_t::reg_vals`

Register content for ECG configuration

7.10.2.2 reg_buffer `uint8_t as7050_config_ecg_t::reg_buffer[sizeof(struct ecg_regs)]`

Register content for ECG configuration

7.11 as7050_config_fifo_t Union Reference

Data Structures

- struct fifo_regs

Data Fields

- struct as7050_config_fifo_t::fifo_regs reg_vals
- uint8_t reg_buffer [sizeof(struct fifo_regs)]

7.11.1 Detailed Description

Register group for configuration of the FIFO handling

7.11.2 Field Documentation

7.11.2.1 reg_vals `struct as7050_config_fifo_t::fifo_regs as7050_config_fifo_t::reg_vals`

Register content for FIFO configuration

7.11.2.2 reg_buffer `uint8_t as7050_config_fifo_t::reg_buffer[sizeof(struct fifo_regs)]`

Register content for FIFO configuration

7.12 as7050_config_gpio_t Union Reference

Data Structures

- struct gpio_regs

Data Fields

- struct as7050_config_gpio_t::gpio_regs reg_vals
- uint8_t reg_buffer [sizeof(struct gpio_regs)]

7.12.1 Detailed Description

Register group for configuration of the GPIOs

7.12.2 Field Documentation

7.12.2.1 reg_vals `struct as7050_config_gpio_t::gpio_regs as7050_config_gpio_t::reg_vals`

Register content for GPIO configuration

7.12.2.2 reg_buffer `uint8_t as7050_config_gpio_t::reg_buffer[sizeof(struct gpio_regs)]`

Register content for GPIO configuration

7.13 as7050_config_iir_t Union Reference

Data Structures

- struct iir_regs

Data Fields

- struct as7050_config_iir_t::iir_regs reg_vals
- uint8_t reg_buffer [sizeof(struct iir_regs)]

7.13.1 Detailed Description

Register group for configuration of the Infinite Impulse Response filter (IIR-filter)

7.13.2 Field Documentation

7.13.2.1 reg_vals `struct as7050_config_iir_t::iir_regs as7050_config_iir_t::reg_vals`

Register content for IIR configuration

7.13.2.2 reg_buffer `uint8_t as7050_config_iir_t::reg_buffer[sizeof(struct iir_regs)]`

Register content for IIR configuration

7.14 as7050_config_led_t Union Reference

Data Structures

- struct led_regs

Data Fields

- struct as7050_config_led_t::led_regs reg_vals
- uint8_t reg_buffer [sizeof(struct led_regs)]

7.14.1 Detailed Description

Register group for LED configuration

7.14.2 Field Documentation

7.14.2.1 reg_vals `struct as7050_config_led_t::led_regs as7050_config_led_t::reg_vals`

Register content for LED configuration

7.14.2.2 reg_buffer `uint8_t as7050_config_led_t::reg_buffer[sizeof(struct led_regs)]`

Register content for LED configuration

7.15 as7050_config_pd_t Union Reference

Data Structures

- struct pd_regs

Data Fields

- struct as7050_config_pd_t::pd_regs reg_vals
- uint8_t reg_buffer [sizeof(struct pd_regs)]

7.15.1 Detailed Description

Register group for configuration of the photodiodes

7.15.2 Field Documentation

7.15.2.1 reg_vals `struct as7050_config_pd_t::pd_regs as7050_config_pd_t::reg_vals`

Register content for photodiodes configuration

7.15.2.2 reg_buffer `uint8_t as7050_config_pd_t::reg_buffer[sizeof(struct pd_regs)]`

Register content for photodiodes configuration

7.16 as7050_config_ppg_t Union Reference

Data Structures

- struct ppg_regs

Data Fields

- struct as7050_config_ppg_t::ppg_regs reg_vals
- uint8_t reg_buffer [sizeof(struct ppg_regs)]

7.16.1 Detailed Description

Register group for configuration of the PPG channels

7.16.2 Field Documentation

7.16.2.1 reg_vals `struct as7050_config_ppg_t::ppg_regs as7050_config_ppg_t::reg_vals`

Register content for PPG configuration

7.16.2.2 reg_buffer `uint8_t as7050_config_ppg_t::reg_buffer[sizeof(struct ppg_regs)]`

Register content for PPG configuration

7.17 as7050_config_ref_t Union Reference

Data Structures

- struct ref_regs

Data Fields

- struct as7050_config_ref_t::ref_regs reg_vals
- uint8_t reg_buffer [sizeof(struct ref_regs)]

7.17.1 Detailed Description

Register group for configuration of the reference registers

7.17.2 Field Documentation

7.17.2.1 reg_vals `struct as7050_config_ref_t::ref_regs as7050_config_ref_t::reg_vals`

Register content for REF-register configuration

7.17.2.2 reg_buffer `uint8_t as7050_config_ref_t::reg_buffer[sizeof(struct ref_regs)]`

Register content for REF-register configuration

7.18 as7050_config_seq_t Union Reference

Data Structures

- struct seq_regs

Data Fields

- struct as7050_config_seq_t::seq_regs reg_vals
- uint8_t reg_buffer [sizeof(struct seq_regs)]

7.18.1 Detailed Description

Register group for configuration of the sequencer

7.18.2 Field Documentation

7.18.2.1 reg_vals `struct as7050_config_seq_t::seq_regs as7050_config_seq_t::reg_vals`

Register content for sequencer configuration

7.18.2.2 reg_buffer `uint8_t as7050_config_seq_t::reg_buffer[sizeof(struct seq_regs)]`

Register content for sequencer configuration

7.19 as7050_config_sinc_t Union Reference

Data Structures

- struct sinc_regs

Data Fields

- struct as7050_config_sinc_t::sinc_regs reg_vals
- uint8_t reg_buffer [sizeof(struct sinc_regs)]

7.19.1 Detailed Description

Register group for configuration of the SINC filter

7.19.2 Field Documentation

7.19.2.1 reg_vals `struct as7050_config_sinc_t::sinc_regs as7050_config_sinc_t::reg_vals`

Register content for SINC filter configuration

7.19.2.2 reg_buffer `uint8_t as7050_config_sinc_t::reg_buffer[sizeof(struct sinc_regs)]`

Register content for SINC filter configuration

7.20 as7050_config_standby_t Union Reference

Data Structures

- struct standby_regs

Data Fields

- struct as7050_config_standby_t::standby_regs reg_vals
- uint8_t reg_buffer [sizeof(struct standby_regs)]

7.20.1 Detailed Description

Register group for configuration for STANDBY mode

7.20.2 Field Documentation

7.20.2.1 reg_vals `struct as7050_config_standby_t::standby_regs as7050_config_standby_t::reg_↩ vals`

Register content for STANDBY configuration

7.20.2.2 reg_buffer `uint8_t as7050_config_standby_t::reg_buffer[sizeof(struct standby_regs)]`

Register content for STANDBY configuration

7.21 as7050_config_tia_t Union Reference

Data Structures

- struct tia_regs

Data Fields

- struct as7050_config_tia_t::tia_regs reg_vals
- uint8_t reg_buffer [sizeof(struct tia_regs)]

7.21.1 Detailed Description

Register group for configuration of the Transimpedance Amplifer (TIA)

7.21.2 Field Documentation

7.21.2.1 reg_vals `struct as7050_config_tia_t::tia_regs as7050_config_tia_t::reg_vals`

Register content for TIA configuration

7.21.2.2 reg_buffer `uint8_t as7050_config_tia_t::reg_buffer[sizeof(struct tia_regs)]`

Register content for TIA configuration

7.22 as7050_meas_config_t Struct Reference

Data Fields

- uint32_t ppg_sample_period_us
- uint32_t ecg_sample_period_us
- uint32_t max_adc_count
- uint16_t fifo_map
- uint16_t fifo_threshold
- uint8_t sample_size
- uint8_t max_adc_bit_width
- uint16_t reserved

7.22.1 Detailed Description

Measurement settings, which can be readout after register configuration

7.22.2 Field Documentation

7.22.2.1 ppg_sample_period_us `uint32_t as7050_meas_config_t::ppg_sample_period_us`

Sample period of PPG signals in microseconds

7.22.2.2 ecg_sample_period_us `uint32_t as7050_meas_config_t::ecg_sample_period_us`

Sample period of ECG signals in microseconds

7.22.2.3 max_adc_count `uint32_t as7050_meas_config_t::max_adc_count`

Maximum possible ADC count for the current configuration

7.22.2.4 fifo_map `uint16_t as7050_meas_config_t::fifo_map`

Definition which channels are mapped inside FIFO. (More than one flag can be set) See as7050_channel_flags_t

7.22.2.5 fifo_threshold `uint16_t as7050_meas_config_t::fifo_threshold`

FIFO threshold, when data shall be read

7.22.2.6 sample_size `uint8_t as7050_meas_config_t::sample_size`

3 or 4 bytes for every sample

7.22.2.7 max_adc_bit_width `uint8_t as7050_meas_config_t::max_adc_bit_width`

Maximum bit width of the ADC count for the current configuration

7.22.2.8 reserved `uint16_t as7050_meas_config_t::reserved`

only for alignment, not used. Always set to 0

7.23 as7050_version Struct Reference

Data Fields

- uint8_t major
- uint8_t minor
- uint8_t patch

7.23.1 Detailed Description

Version information of the library

7.23.2 Field Documentation

7.23.2.1 major `uint8_t as7050_version::major`

Major version position

7.23.2.2 minor `uint8_t as7050_version::minor`

Minor version position

7.23.2.3 patch `uint8_t as7050_version::patch`

Patch version position

7.24 as7050_config_ctrl_t::ctrl_regs Struct Reference

Data Fields

- uint8_t control

7.24.1 Detailed Description

Register content for CONTROL register

7.24.2 Field Documentation

7.24.2.1 control `uint8_t as7050_config_ctrl_t::ctrl_regs::control`

Content of register AS7050_REGADDR_CONTROL

7.25 as7050_config_ecg_t::ecg_regs Struct Reference

Data Fields

- uint8_t ecg_source
- uint8_t ecg_mod_cfga

7.25.1 Detailed Description

Register content for ECG configuration

7.25.2 Field Documentation

7.25.2.1 ecg_source `uint8_t as7050_config_ecg_t::ecg_regs::ecg_source`

Content of register AS7050_REGADDR_ECG_SOURCE

7.25.2.2 ecg_mod_cfga `uint8_t as7050_config_ecg_t::ecg_regs::ecg_mod_cfga`

Content of register AS7050_REGADDR_ECG_MOD_CFGA

7.26 as7050_config_fifo_t::fifo_regs Struct Reference

Data Fields

- uint8_t fifo_ctrl
- uint8_t fifo_threshold

7.26.1 Detailed Description

Register content for FIFO configuration

7.26.2 Field Documentation

7.26.2.1 fifo_ctrl `uint8_t as7050_config_fifo_t::fifo_regs::fifo_ctrl`

Content of register AS7050_REGADDR_FIFO_CTRL

7.26.2.2 fifo_threshold `uint8_t as7050_config_fifo_t::fifo_regs::fifo_threshold`

Content of register AS7050_REGADDR_FIFO_THRESHOLD

7.27 as7050_config_gpio_t::gpio_regs Struct Reference

Data Fields

- uint8_t gpio1_cfg
- uint8_t gpio2_cfg
- uint8_t gpio1_cfgb
- uint8_t gpio2_cfgb
- uint8_t gpio_io

7.27.1 Detailed Description

Register content for GPIO configuration

7.27.2 Field Documentation

7.27.2.1 gpio1_cfg `uint8_t as7050_config_gpio_t::gpio_regs::gpio1_cfg`

Content of register AS7050_REGADDR_GPIO1_CFG

7.27.2.2 gpio2_cfg `uint8_t as7050_config_gpio_t::gpio_regs::gpio2_cfg`

Content of register AS7050_REGADDR_GPIO2_CFG

7.27.2.3 gpio1_cfgb `uint8_t as7050_config_gpio_t::gpio_regs::gpio1_cfgb`

Content of register AS7050_REGADDR_GPIO1_CFGB

7.27.2.4 gpio2_cfgb `uint8_t as7050_config_gpio_t::gpio_regs::gpio2_cfgb`

Content of register AS7050_REGADDR_GPIO2_CFGB

7.27.2.5 gpio_io `uint8_t as7050_config_gpio_t::gpio_regs::gpio_io`

Content of register AS7050_REGADDR_GPIO_IO

7.28 as7050_config_iir_t::iir_regs Struct Reference

Data Fields

- uint8_t iir_cfg
- int16_t iir_coeff_data_sos [12][5]

7.28.1 Detailed Description

Register content for IIR configuration

7.28.2 Field Documentation

7.28.2.1 iir_cfg `uint8_t as7050_config_iir_t::iir_regs::iir_cfg`

Content of register AS7050_REGADDR_IIR_CFG

7.28.2.2 iir_coeff_data_sos `int16_t as7050_config_iir_t::iir_regs::iir_coeff_data_sos[12][5]`

RAM area where IIR-coefficients will be saved

7.29 as7050_config_led_t::led_regs Struct Reference

Data Fields

- uint8_t lowvds_wait
- uint8_t led1_ictrl
- uint8_t led2_ictrl
- uint8_t led3_ictrl
- uint8_t led4_ictrl
- uint8_t led5_ictrl
- uint8_t led6_ictrl
- uint8_t led7_ictrl
- uint8_t led8_ictrl
- uint8_t led_init
- uint8_t led_ppg1
- uint8_t led_ppg2
- uint8_t led_ppg3
- uint8_t led_ppg4
- uint8_t led_ppg5
- uint8_t led_ppg6
- uint8_t led_ppg7
- uint8_t led_ppg8
- uint8_t led_tia
- uint8_t led_mode

7.29.1 Detailed Description

Register content for LED configuration

7.29.2 Field Documentation

7.29.2.1 lowvds_wait `uint8_t as7050_config_led_t::led_regs::lowvds_wait`

Content of register AS7050_REGADDR_LOWVDS_WAIT

7.29.2.2 led1_ictrl `uint8_t as7050_config_led_t::led_regs::led1_ictrl`

Content of register AS7050_REGADDR_LED1_ICTRL

7.29.2.3 led2_ictrl `uint8_t as7050_config_led_t::led_regs::led2_ictrl`

Content of register AS7050_REGADDR_LED2_ICTRL

7.29.2.4 led3_ictrl `uint8_t as7050_config_led_t::led_regs::led3_ictrl`

Content of register AS7050_REGADDR_LED3_ICTRL

7.29.2.5 led4_ictrl `uint8_t as7050_config_led_t::led_regs::led4_ictrl`

Content of register AS7050_REGADDR_LED4_ICTRL

7.29.2.6 led5_ictrl `uint8_t as7050_config_led_t::led_regs::led5_ictrl`

Content of register AS7050_REGADDR_LED5_ICTRL

7.29.2.7 led6_ictrl `uint8_t as7050_config_led_t::led_regs::led6_ictrl`

Content of register AS7050_REGADDR_LED6_ICTRL

7.29.2.8 led7_ictrl `uint8_t as7050_config_led_t::led_regs::led7_ictrl`

Content of register AS7050_REGADDR_LED7_ICTRL

7.29.2.9 led8_ictrl `uint8_t as7050_config_led_t::led_regs::led8_ictrl`

Content of register AS7050_REGADDR_LED8_ICTRL

7.29.2.10 led_init `uint8_t as7050_config_led_t::led_regs::led_init`

Content of register AS7050_REGADDR_LED_INIT

7.29.2.11 led_ppg1 `uint8_t as7050_config_led_t::led_regs::led_ppg1`

Content of register AS7050_REGADDR_LED_PPG1

7.29.2.12 led_ppg2 `uint8_t as7050_config_led_t::led_regs::led_ppg2`

Content of register AS7050_REGADDR_LED_PPG2

7.29.2.13 led_ppg3 `uint8_t as7050_config_led_t::led_regs::led_ppg3`

Content of register AS7050_REGADDR_LED_PPG3

7.29.2.14 led_ppg4 `uint8_t as7050_config_led_t::led_regs::led_ppg4`

Content of register AS7050_REGADDR_LED_PPG4

7.29.2.15 led_ppg5 `uint8_t as7050_config_led_t::led_regs::led_ppg5`

Content of register AS7050_REGADDR_LED_PPG5

7.29.2.16 led_ppg6 `uint8_t as7050_config_led_t::led_regs::led_ppg6`

Content of register AS7050_REGADDR_LED_PPG6

7.29.2.17 led_ppg7 `uint8_t as7050_config_led_t::led_regs::led_ppg7`

Content of register AS7050_REGADDR_LED_PPG7

7.29.2.18 led_ppg8 `uint8_t as7050_config_led_t::led_regs::led_ppg8`

Content of register AS7050_REGADDR_LED_PPG8

7.29.2.19 led_tia `uint8_t as7050_config_led_t::led_regs::led_tia`

Content of register AS7050_REGADDR_LED_TIA

7.29.2.20 led_mode `uint8_t as7050_config_led_t::led_regs::led_mode`

Content of register AS7050_REGADDR_LED_MODE

7.30 as7050_config_pd_t::pd_regs Struct Reference

Data Fields

- uint8_t pdsel_cfg
- uint8_t pd_ppg1
- uint8_t pd_ppg2
- uint8_t pd_ppg3
- uint8_t pd_ppg4
- uint8_t pd_ppg5
- uint8_t pd_ppg6
- uint8_t pd_ppg7
- uint8_t pd_ppg8
- uint8_t pd_tia

7.30.1 Detailed Description

Register content for photodiodes configuration

7.30.2 Field Documentation

7.30.2.1 pdsel_cfg `uint8_t as7050_config_pd_t::pd_regs::pdsel_cfg`

Content of register AS7050_REGADDR_PDSEL_CFG

7.30.2.2 pd_ppg1 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg1`

Content of register AS7050_REGADDR_PD_PPG1

7.30.2.3 pd_ppg2 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg2`

Content of register AS7050_REGADDR_PD_PPG2

7.30.2.4 pd_ppg3 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg3`

Content of register AS7050_REGADDR_PD_PPG3

7.30.2.5 pd_ppg4 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg4`

Content of register AS7050_REGADDR_PD_PPG4

7.30.2.6 pd_ppg5 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg5`

Content of register AS7050_REGADDR_PD_PPG5

7.30.2.7 pd_ppg6 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg6`

Content of register AS7050_REGADDR_PD_PPG6

7.30.2.8 pd_ppg7 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg7`

Content of register AS7050_REGADDR_PD_PPG7

7.30.2.9 pd_ppg8 `uint8_t as7050_config_pd_t::pd_regs::pd_ppg8`

Content of register AS7050_REGADDR_PD_PPG8

7.30.2.10 pd_tia `uint8_t as7050_config_pd_t::pd_regs::pd_tia`

Content of register AS7050_REGADDR_PD_TIA

7.31 as7050_config_ppg_t::ppg_regs Struct Reference

Data Fields

- uint8_t ppg_mod_cfga
- uint8_t ppg_mod_cfgb
- uint8_t ppg_mod_cfgc
- uint8_t ppg_mod_cfgd
- uint8_t ppg_mod_cfge

7.31.1 Detailed Description

Register content for PPG configuration

7.31.2 Field Documentation

7.31.2.1 ppg_mod_cfga `uint8_t as7050_config_ppg_t::ppg_regs::ppg_mod_cfga`

Content of register AS7050_REGADDR_PPG_MOD_CFGA

7.31.2.2 ppg_mod_cfgb `uint8_t as7050_config_ppg_t::ppg_regs::ppg_mod_cfgb`

Content of register AS7050_REGADDR_PPG_MOD_CFGB

7.31.2.3 ppg_mod_cfgc `uint8_t as7050_config_ppg_t::ppg_regs::ppg_mod_cfgc`

Content of register AS7050_REGADDR_PPG_MOD_CFGC

7.31.2.4 ppg_mod_cfgd `uint8_t as7050_config_ppg_t::ppg_regs::ppg_mod_cfgd`

Content of register AS7050_REGADDR_PPG_MOD_CFGD

7.31.2.5 ppg_mod_cfge `uint8_t as7050_config_ppg_t::ppg_regs::ppg_mod_cfge`

Content of register AS7050_REGADDR_PPG_MOD_CFGE

7.32 as7050_config_ref_t::ref_regs Struct Reference

Data Fields

- uint8_t ref_cfga
- uint8_t ref_cfgb

7.32.1 Detailed Description

Register content for REF-register configuration

7.32.2 Field Documentation

7.32.2.1 ref_cfga `uint8_t as7050_config_ref_t::ref_regs::ref_cfga`

Content of register AS7050_REGADDR_REF_CFGA

7.32.2.2 ref_cfgb `uint8_t as7050_config_ref_t::ref_regs::ref_cfgb`

Content of register AS7050_REGADDR_REF_CFGB

7.33 as7050_config_seq_t::seq_regs Struct Reference

Data Fields

- uint8_t cgb_cfg
- uint8_t seq_sample
- uint8_t seq_ppga
- uint8_t seq_ppgb
- uint8_t seq_mode

7.33.1 Detailed Description

Register content for sequencer configuration

7.33.2 Field Documentation

7.33.2.1 cgb_cfg `uint8_t as7050_config_seq_t::seq_regs::cgb_cfg`

Content of register AS7050_REGADDR_CGB_CFG

7.33.2.2 seq_sample `uint8_t as7050_config_seq_t::seq_regs::seq_sample`

Content of register AS7050_REGADDR_SEQ_SAMPLE

7.33.2.3 seq_ppga `uint8_t as7050_config_seq_t::seq_regs::seq_ppga`

Content of register AS7050_REGADDR_SEQ_PPGA

7.33.2.4 seq_ppgb `uint8_t as7050_config_seq_t::seq_regs::seq_ppgb`

Content of register AS7050_REGADDR_SEQ_PPGB

7.33.2.5 seq_mode `uint8_t as7050_config_seq_t::seq_regs::seq_mode`

Content of register AS7050_REGADDR_SEQ_MODE

7.34 as7050_config_sinc_t::sinc_regs Struct Reference

Data Fields

- uint8_t sinc_ppg_cfga
- uint8_t sinc_ppg_cfgb
- uint8_t sinc_ppg_cfgc
- uint8_t sinc_ecg_cfga
- uint8_t sinc_ecg_cfgb
- uint8_t sinc_ecg_cfgc
- uint8_t ovs_cfg

7.34.1 Detailed Description

Register content for SINC filter configuration

7.34.2 Field Documentation

7.34.2.1 sinc_ppg_cfga `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ppg_cfga`

Content of register AS7050_REGADDR_SINC_PPG_CFGA

7.34.2.2 sinc_ppg_cfgb `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ppg_cfgb`

Content of register AS7050_REGADDR_SINC_PPG_CFGB

7.34.2.3 sinc_ppg_cfgc `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ppg_cfgc`

Content of register AS7050_REGADDR_SINC_PPG_CFGC

7.34.2.4 sinc_ecg_cfga `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ecg_cfga`

Content of register AS7050_REGADDR_SINC_ECG_CFGA

7.34.2.5 sinc_ecg_cfgb `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ecg_cfgb`

Content of register AS7050_REGADDR_SINC_ECG_CFGB

7.34.2.6 sinc_ecg_cfgc `uint8_t as7050_config_sinc_t::sinc_regs::sinc_ecg_cfgc`

Content of register AS7050_REGADDR_SINC_ECG_CFGC

7.34.2.7 ovs_cfg `uint8_t as7050_config_sinc_t::sinc_regs::ovs_cfg`

Content of register AS7050_REGADDR_OVS_CFG

7.35 as7050_config_standby_t::standby_regs Struct Reference

Data Fields

- uint8_t standby_cfga
- uint8_t standby_cfgb

7.35.1 Detailed Description

Register content for STANDBY configuration

7.35.2 Field Documentation

7.35.2.1 standby_cfga `uint8_t as7050_config_standby_t::standby_regs::standby_cfga`

Content of register AS7050_REGADDR_STANDBY_CFGA

7.35.2.2 standby_cfgb `uint8_t as7050_config_standby_t::standby_regs::standby_cfgb`

Content of register AS7050_REGADDR_STANDBY_CFGB

7.36 as7050_config_tia_t::tia_regs Struct Reference

Data Fields

- uint8_t pd_offset_cfg
- uint8_t tia_cfga
- uint8_t tia_cfgb
- uint8_t tia_cfgc

7.36.1 Detailed Description

Register content for TIA configuration

7.36.2 Field Documentation

7.36.2.1 pd_offset_cfg `uint8_t as7050_config_tia_t::tia_regs::pd_offset_cfg`

Content of register AS7050_REGADDR_PD_OFFSET_CFG

7.36.2.2 tia_cfga `uint8_t as7050_config_tia_t::tia_regs::tia_cfga`

Content of register AS7050_REGADDR_TIA_CFGA

7.36.2.3 tia_cfgb `uint8_t as7050_config_tia_t::tia_regs::tia_cfgb`

Content of register AS7050_REGADDR_TIA_CFGB

7.36.2.4 tia_cfgc `uint8_t as7050_config_tia_t::tia_regs::tia_cfgc`

Content of register AS7050_REGADDR_TIA_CFGC

Index